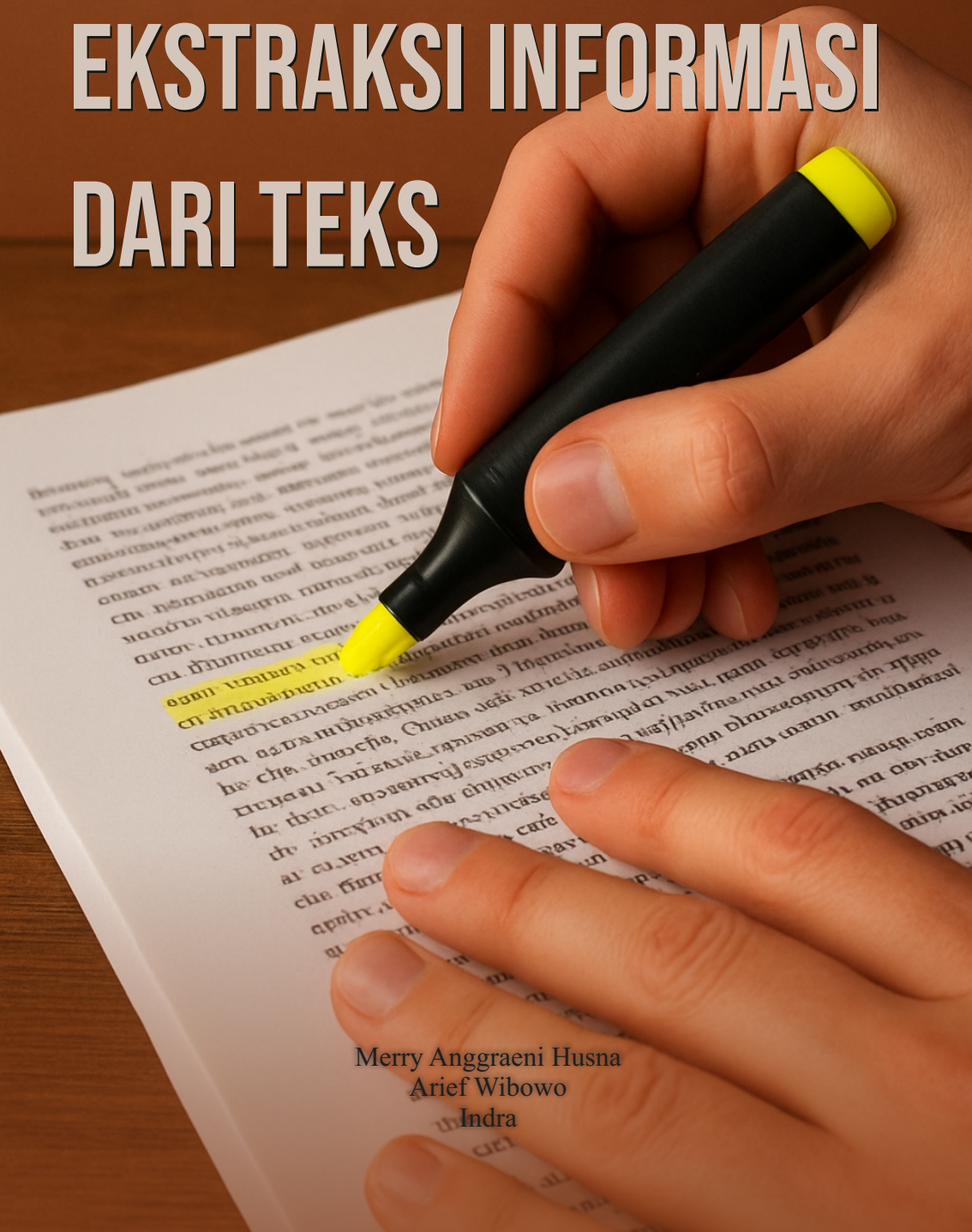
Buku Ajar

Pustaka Aksara

EKSTRAKSI INFORMASI DARI TEKS



Merry Anggraeni Husna
Arief Wibowo
Indra

Buku Ajar

EKSTRAKSI INFORMASI DARI TEKS

Merry Anggraeni Husna
Arief Wibowo
Indra

Pustaka Aksara

Buku Ajar
EKSTRAKSI INFORMASI DARI TEKS

Penulis : **Merry Anggraeni Husna**
Arief Wibowo
Indra
Desain Sampul : **Jessica Nathania**
Tata Letak : **Ayulis Mutiara Putri**

ISBN : 978-623-161-585-5

Diterbitkan oleh : **PUSTAKA AKSARA, 2025**
Redaksi:
Surabaya, Jawa Timur, Indonesia
Telp. 0858-0746-8047
Laman : www.pustakaaksara.co.id
Surel : info@pustakaaksara.co.id

Anggota IKAPI : 277/JTI/2021

Cetakan Pertama : 2025

KATA PENGANTAR

Puji syukur kami panjatkan ke hadirat Tuhan Yang Maha Esa, atas rahmat dan karunia-Nya buku berjudul "*Ekstraksi Informasi dari Teks*" ini dapat diselesaikan. Buku ajar ini hadir sebagai upaya menyediakan sumber belajar yang sistematis dan aplikatif bagi mahasiswa maupun praktisi yang ingin memahami bagaimana informasi dapat diolah secara otomatis dari teks menggunakan pendekatan komputasional.

Ekstraksi informasi (Information Extraction/IE) merupakan salah satu cabang penting dalam bidang *Natural Language Processing (NLP)* yang berperan mengubah data teks tak terstruktur menjadi data terorganisasi dan siap digunakan untuk analisis maupun pengambilan keputusan. Dalam era ledakan data saat ini, kemampuan mengekstraksi entitas, relasi, maupun peristiwa dari teks menjadi fondasi bagi berbagai aplikasi, mulai dari sistem intelijen bisnis, layanan kesehatan, analisis media sosial, hingga sistem pendukung keputusan berbasis kecerdasan buatan.

Buku ini disusun secara bertahap mulai dari konsep dasar, preprocessing teks, feature engineering, POS tagging dan analisis SPOK, hingga pendekatan rule-based dan metode lanjutan dalam ekstraksi informasi. Selain itu, kami sertakan pula contoh kasus nyata, studi penerapan, serta pembahasan tools NLP modern agar pembaca dapat melihat relevansi konsep dengan praktik di lapangan.

Harapan kami, buku ajar ini tidak hanya memperkaya pengetahuan teoretis, tetapi juga menjadi pedoman praktis dalam membangun sistem berbasis teks. Buku ini ditujukan terutama bagi mahasiswa program studi informatika, ilmu komputer, dan bidang terkait, namun juga relevan untuk

peneliti, dosen, maupun praktisi industri yang ingin mendalami pengolahan bahasa alami.

Semoga buku ajar ini bermanfaat dan dapat menjadi kontribusi nyata dalam pengembangan ilmu pengetahuan, khususnya di bidang *Natural Language Processing* dan *Information Extraction*.

Selamat membaca!

September, 2025

Penulis

DAFTAR ISI

BAB 1
PENDAHULUAN

Tujuan Pembelajaran

Setelah mempelajari Bab 1, pembaca diharapkan mampu:

1. Menjelaskan konsep dasar ekstraksi informasi (Information Extraction/IE) serta perannya dalam pengolahan bahasa alami (*Natural Language Processing/NLP*).
2. Mendeskripsikan ruang lingkup dan komponen utama IE, seperti pengenalan entitas, ekstraksi relasi, dan ekstraksi peristiwa.
3. Membedakan antara data terstruktur dan tidak terstruktur, serta memahami tantangan dalam mengubah teks menjadi data terorganisasi.
4. Menguraikan keterkaitan antara IE, NLP, dan bidang-bidang lain seperti *machine learning* dan *text mining*.
5. Mengidentifikasi manfaat dan aplikasi IE dalam berbagai domain (kesehatan, pemerintahan, media sosial, bisnis, dan keamanan siber).
6. Menjelaskan perbedaan pendekatan rule-based dan statistik dalam sistem ekstraksi informasi.
7. Menganalisis permasalahan dan tantangan utama yang dihadapi dalam pengembangan sistem IE berbasis Bahasa Indonesia.

A. Definisi dan Ruang Lingkup Ekstraksi Informasi

Dalam era digital yang ditandai oleh ledakan data, informasi dalam bentuk teks telah menjadi salah satu bentuk dominan dari data tak terstruktur. Teks hadir dalam berbagai wujud: berita daring, catatan medis, percakapan di media sosial, laporan keuangan, hingga

dokumen hukum dan pemerintahan. Meskipun kaya akan informasi, bentuk alaminya yang tidak terstruktur membuat pemrosesan teks menjadi tantangan tersendiri dalam dunia informatika. Oleh karena itu, ekstraksi informasi, atau Information Extraction (IE), muncul sebagai salah satu bidang utama dalam pengolahan bahasa alami (Natural Language Processing atau NLP), dengan tujuan utama mengambil informasi-informasi penting dari teks dan mengubahnya menjadi bentuk yang lebih terstruktur dan dapat dianalisis lebih lanjut oleh mesin.

Ekstraksi informasi secara umum dapat dipahami sebagai proses identifikasi otomatis terhadap bagian-bagian relevan dalam sebuah dokumen teks yang mengandung informasi bermakna, kemudian menyusunnya kembali dalam bentuk representasi yang eksplisit. Representasi ini dapat berupa pasangan entitas dan relasi, daftar fakta, tabel data, atau skema informasi lainnya. Dengan kata lain, IE bertugas mengungkap struktur semantik tersembunyi dari teks berbahasa alami dan mentransformasikannya menjadi data yang siap digunakan dalam berbagai aplikasi cerdas. Misalnya, dari sebuah kalimat seperti "Presiden Joko Widodo meresmikan Bendungan Napun Gete di Nusa Tenggara Timur pada 23 Februari 2022," sistem IE yang ideal diharapkan mampu mengekstrak bahwa subjeknya adalah "Joko Widodo" (sebagai entitas orang), aksinya adalah "meresmikan," objeknya adalah "Bendungan Napun Gete" (sebagai infrastruktur), lokasinya adalah "Nusa Tenggara Timur," dan waktunya adalah "23 Februari 2022."

Dalam praktiknya, ekstraksi informasi mencakup berbagai komponen saling terkait yang membentuk sebuah pipeline pemrosesan. Komponen pertama yang biasanya diterapkan adalah pengenalan entitas bernama

atau Named Entity Recognition (NER), yakni proses untuk mengenali dan mengelompokkan nama-nama penting yang disebut dalam teks seperti nama orang, organisasi, lokasi geografis, dan tanggal. Setelah entitas dikenali, tahap berikutnya adalah memahami hubungan antar entitas, yang disebut sebagai relation extraction. Misalnya, sistem dapat mengenali bahwa seseorang adalah karyawan di sebuah perusahaan, atau bahwa dua lokasi memiliki hubungan administratif. Pada tingkat yang lebih tinggi, ekstraksi peristiwa atau event extraction mencoba menangkap aksi atau kegiatan penting yang melibatkan satu atau lebih entitas, misalnya peristiwa bencana, pertemuan diplomatik, atau transaksi bisnis.

Selain itu, IE juga memerlukan teknik tambahan seperti penyelesaian koreferensi (coreference resolution), yakni kemampuan sistem untuk memahami bahwa kata ganti seperti "dia," "itu," atau "mereka" merujuk pada entitas tertentu dalam teks sebelumnya. Proses ini penting untuk menjaga konsistensi makna dalam dokumen yang panjang. Informasi yang telah diperoleh dari proses-proses ini kemudian dapat diorganisir ke dalam format standar melalui template filling atau skema data lainnya.

Ruang lingkup ekstraksi informasi sangat luas dan terus berkembang seiring dengan meningkatnya kebutuhan pemrosesan data teks di berbagai bidang. Dalam domain kesehatan, IE digunakan untuk mengekstraksi diagnosis dan pengobatan dari rekam medis elektronik. Di bidang keuangan, informasi mengenai perubahan harga, merger, atau akuisisi dapat diambil dari laporan dan berita pasar. Dalam konteks pemerintahan dan hukum, IE membantu menganalisis regulasi, kontrak, atau kebijakan publik secara sistematis. Bahkan dalam keamanan siber, informasi dari laporan

insiden atau log teknis bisa diekstrak untuk pemantauan ancaman secara real-time. Semakin meluasnya aplikasi IE juga disertai kebutuhan untuk menyesuaikan pendekatan teknis terhadap bahasa dan karakteristik domain yang digunakan. Bahasa Indonesia, misalnya, memiliki struktur morfologis dan sintaktis yang berbeda dari bahasa Inggris, sehingga sistem EI lokal perlu memperhatikan kekhasan ini dengan pendekatan linguistik yang sesuai.

Walaupun ekstraksi informasi termasuk dalam lingkup NLP, bidang ini juga memiliki keterkaitan erat dengan ilmu lainnya. Dalam sistem pencarian informasi atau Information Retrieval (IR), IE berfungsi untuk mendalami isi dokumen setelah dokumen tersebut ditemukan. Dalam text mining, IE berperan menyediakan data terstruktur sebagai masukan bagi analisis pola atau tren. Dalam machine learning, IE dapat mengandalkan model pembelajaran statistik untuk mengenali pola-pola bahasa berdasarkan data pelatihan. Bahkan dalam data visualization, hasil IE dapat divisualisasikan sebagai grafik entitas, tabel relasi, atau word cloud untuk memberikan wawasan yang lebih intuitif.

Perkembangan teknologi dalam beberapa tahun terakhir juga telah membawa perubahan signifikan dalam pendekatan terhadap IE. Jika pada awalnya sistem IE sangat bergantung pada aturan-aturan linguistik yang ditulis secara manual (rule-based system), kini banyak sistem modern yang menggunakan pendekatan statistik dan pembelajaran mesin. Model deep learning seperti BERT, spaCy, dan transformer lainnya mampu memahami konteks bahasa secara lebih dalam, memungkinkan sistem IE untuk beradaptasi dengan gaya bahasa yang lebih variatif dan kompleks. Namun demikian, untuk bahasa dengan sumber daya terbatas seperti Bahasa Indonesia,

pendekatan berbasis aturan masih sangat relevan, khususnya dalam domain-domain spesifik yang membutuhkan ketelitian dan keandalan tinggi.

Dengan kata lain, IE tidak hanya berfungsi sebagai alat bantu teknis dalam pengolahan teks, tetapi telah menjadi infrastruktur dasar dalam sistem informasi cerdas yang berbasis data. Ia tidak hanya membuka pintu untuk eksplorasi informasi dari teks yang dulunya tidak dapat diakses oleh mesin, tetapi juga memungkinkan pemahaman dan pengambilan keputusan yang lebih cepat, akurat, dan kontekstual. Pemahaman menyeluruh terhadap definisi, komponen, dan ruang lingkup ekstraksi informasi merupakan fondasi penting bagi siapa pun yang ingin mendalami bidang ini lebih jauh, baik dari sisi akademik maupun praktis.

B. Kegunaan Ekstraksi Informasi dalam Analisis Teks

Ekstraksi informasi bukan sekadar aktivitas teknis dalam memproses data teks, melainkan suatu pendekatan strategis untuk menjembatani dunia informasi manusia dengan sistem komputasi. Tujuan utama dari ekstraksi informasi adalah mengubah data tidak terstruktur yang umumnya ditulis dalam bahasa alami menjadi representasi terstruktur yang lebih mudah diproses oleh algoritma dan sistem komputer. Proses ini bukan hanya menghasilkan data yang "terlihat" oleh mesin, tetapi juga memfasilitasi pemahaman konteks, makna, dan keterkaitan antar informasi di dalam dokumen.

Salah satu tujuan fundamental dari ekstraksi informasi adalah menyederhanakan kerumitan bahasa alami agar informasi penting yang terkandung dalam teks dapat diakses secara otomatis dan efisien. Dalam dunia nyata, manusia sering kali membutuhkan waktu dan

tenaga untuk menyaring informasi dari dokumen panjang atau kumpulan besar artikel. Sistem ekstraksi informasi memungkinkan proses ini dilakukan secara sistematis dan dalam skala besar, sehingga sangat mendukung efisiensi dan efektivitas dalam pengambilan keputusan. Dalam konteks bisnis, misalnya, perusahaan dapat dengan cepat memperoleh laporan keuangan atau tren pasar dari berita dan dokumen publik. Dalam bidang kesehatan, dokter dan peneliti dapat mengekstrak informasi dari ribuan catatan medis atau jurnal ilmiah untuk mempercepat proses diagnosis dan pengembangan terapi.

Manfaat praktis dari ekstraksi informasi sangat luas dan nyata dalam berbagai bidang. Dalam analisis teks, sistem EI membantu mengurangi kompleksitas data mentah menjadi kumpulan informasi bernilai yang siap dianalisis. Melalui sistem ini, entitas-entitas penting seperti nama tokoh, lokasi, waktu kejadian, atau jenis peristiwa dapat diekstraksi dari teks panjang, kemudian disusun dalam format yang lebih mudah dipahami dan digunakan untuk analisis lanjutan. Dengan demikian, ekstraksi informasi menjadi elemen penting dalam pengembangan sistem business intelligence, customer insight, hingga social media analytics. Di dunia akademik, IE mempercepat proses kajian literatur dengan mengekstrak kutipan, pengarang, atau hasil penelitian dari berbagai sumber teks ilmiah.

Dalam skala makro, EI juga memiliki kontribusi besar terhadap pengembangan sistem kecerdasan buatan yang adaptif terhadap konteks linguistik. Banyak sistem cerdas seperti chatbot, asisten digital, sistem rekomendasi, hingga mesin pencari canggih, bergantung pada kemampuan EI untuk memahami maksud pengguna dan mengekstrak informasi relevan dari data tekstual.

Misalnya, chatbot layanan pelanggan yang mampu mengenali keluhan, pertanyaan, atau perintah dari teks input pengguna sebenarnya memanfaatkan pipeline EI untuk mengurai elemen-elemen semantik dari percakapan tersebut.

Manfaat lainnya yang semakin relevan di era big data adalah peningkatan efisiensi dalam pemrosesan informasi massal. Organisasi yang mengelola data dalam jumlah besar, seperti lembaga pemerintahan, institusi riset, dan perusahaan media, sangat terbantu dengan penerapan sistem ekstraksi informasi dalam menyaring, memilah, dan menyusun data teks menjadi laporan yang lebih ringkas dan bermakna. Bahkan, dalam konteks penegakan hukum dan investigasi keamanan, EI membantu dalam penelusuran komunikasi, identifikasi pola, dan deteksi anomali berbasis teks.

Tak kalah penting, ekstraksi informasi juga membuka peluang untuk otomatisasi proses administrasi dan dokumentasi. Banyak institusi kini mulai menggunakan sistem EI untuk secara otomatis mengisi formulir, menyusun indeks dokumen, hingga menghasilkan ringkasan eksekutif dari laporan panjang. Hal ini tentu berkontribusi pada penghematan sumber daya dan peningkatan kualitas layanan informasi.

Meskipun demikian, perlu dicatat bahwa manfaat dari ekstraksi informasi sangat bergantung pada kualitas sistem yang digunakan, termasuk akurasi dalam pengenalan entitas dan relasi, serta ketepatan dalam interpretasi konteks. Oleh karena itu, pengembangan sistem EI harus dilakukan dengan perhatian pada kualitas data latih, desain arsitektur sistem, serta pemahaman terhadap domain dan bahasa yang digunakan. Dalam hal ini, peran kombinasi antara pendekatan linguistik berbasis

aturan dan pendekatan statistik berbasis data menjadi krusial untuk mencapai sistem yang handal.

Secara keseluruhan, tujuan dan manfaat ekstraksi informasi tidak hanya terletak pada aspek teknis dalam transformasi data, tetapi juga pada kemampuannya mendukung proses analisis dan pengambilan keputusan secara menyeluruh. Dengan terus berkembangnya teknologi NLP dan pembelajaran mesin, manfaat ini diprediksi akan semakin luas dan dalam, menjadikan ekstraksi informasi sebagai fondasi utama dalam pengolahan data tekstual di berbagai bidang.

C. Metode Rule-Based dan Statistik dalam Ekstraksi Informasi

Dalam membangun sistem ekstraksi informasi, pendekatan yang digunakan secara umum dapat dikategorikan ke dalam dua paradigma utama, yaitu pendekatan berbasis aturan (rule-based) dan pendekatan berbasis statistik atau pembelajaran mesin (machine learning-based). Kedua pendekatan ini memiliki karakteristik yang berbeda, dengan kelebihan dan keterbatasan masing-masing, serta aplikasi yang bergantung pada konteks dan kebutuhan sistem yang dibangun. Pemahaman terhadap perbedaan fundamental di antara keduanya sangat penting untuk merancang solusi ekstraksi informasi yang efektif, efisien, dan sesuai dengan karakteristik data yang dihadapi.

Pendekatan rule-based merupakan metode yang bersandar pada seperangkat aturan eksplisit yang dirancang secara manual oleh pakar domain atau pengembang sistem. Aturan-aturan ini biasanya memanfaatkan pola linguistik tertentu, ekspresi reguler, atau struktur kalimat untuk mendeteksi dan mengekstrak

entitas atau relasi dalam teks. Misalnya, dalam kasus ekstraksi nama orang, aturan dapat dirancang untuk mengenali pola seperti "Bapak [NAMA]" atau "Presiden [NAMA]" dalam teks bahasa Indonesia. Keunggulan utama dari pendekatan ini terletak pada transparansi dan kendali penuh yang dimiliki oleh pengembang sistem. Karena aturan dirancang secara eksplisit, maka perilaku sistem dapat diprediksi dan disesuaikan dengan cepat ketika ditemukan kesalahan atau inkonsistensi.

Namun, di balik kejelasan dan fleksibilitas lokalnya, pendekatan berbasis aturan memiliki keterbatasan dalam hal skalabilitas dan kemampuan beradaptasi terhadap variasi bahasa. Bahasa alami bersifat sangat dinamis dan penuh dengan ambiguitas, sehingga pendekatan berbasis aturan cenderung mengalami kesulitan ketika dihadapkan pada teks yang tidak sesuai dengan pola yang telah diprogram. Selain itu, pembangunan dan pemeliharaan aturan memerlukan waktu dan tenaga yang tidak sedikit, terutama jika sistem akan digunakan dalam domain yang luas atau dalam berbagai bahasa.

Sebaliknya, pendekatan statistik dan pembelajaran mesin menawarkan solusi yang lebih adaptif terhadap keragaman data. Dalam pendekatan ini, sistem tidak lagi didesain berdasarkan aturan eksplisit, tetapi dilatih menggunakan data contoh (corpus) yang telah dianotasi. Model pembelajaran mesin seperti Support Vector Machine, Conditional Random Fields, dan dalam perkembangannya, Deep Learning dengan arsitektur seperti LSTM dan Transformer, mampu mengenali pola-pola kompleks dalam teks tanpa perlu perumusan aturan manual. Misalnya, model dapat dilatih untuk mengenali entitas “lokasi” dengan memberikan banyak contoh kalimat yang mengandung nama tempat, tanpa perlu

mendefinisikan aturan gramatikal atau sintaktik secara eksplisit.

Keunggulan dari pendekatan ini terletak pada kemampuannya untuk bekerja dengan volume data besar dan generalisasi terhadap variasi kalimat yang luas. Model statistik juga memiliki potensi untuk terus ditingkatkan akurasinya seiring bertambahnya data pelatihan. Akan tetapi, pendekatan ini juga memiliki tantangan tersendiri. Pertama, dibutuhkan ketersediaan data anotasi yang representatif dan berkualitas tinggi. Kedua, proses pelatihan model bisa memerlukan sumber daya komputasi yang besar. Ketiga, hasil ekstraksi cenderung lebih sulit untuk ditelusuri dan dijelaskan secara transparan, sehingga menyulitkan proses debugging atau interpretasi hasil.

Dalam praktik terbaik pengembangan sistem ekstraksi informasi modern, sering kali digunakan pendekatan hibrida yang menggabungkan kedua metode tersebut. Pendekatan hibrida memungkinkan pemanfaatan kekuatan dari rule-based system dalam domain yang terstruktur dan spesifik, serta fleksibilitas dari machine learning dalam menghadapi data yang lebih dinamis dan kompleks. Misalnya, entitas yang sudah diketahui secara pasti seperti daftar nama kota atau instansi dapat diekstraksi menggunakan aturan sederhana, sementara pengenalan relasi semantik antar kalimat dapat dibantu dengan model statistik.

Pemilihan pendekatan yang tepat sangat bergantung pada sejumlah faktor, seperti ketersediaan data, tujuan sistem, tingkat akurasi yang dibutuhkan, serta keberlanjutan pemeliharaan sistem. Untuk konteks bahasa Indonesia, yang termasuk dalam kategori bahasa dengan sumber daya terbatas (low-resource language),

pendekatan rule-based masih memiliki peran penting, terutama pada domain yang spesifik atau ketika data anotasi masih belum tersedia secara luas. Namun, dengan semakin berkembangnya infrastruktur NLP dan tersedianya model pra-latih (pre-trained models) dalam bahasa lokal, integrasi pendekatan statistik diprediksi akan semakin dominan dalam pengembangan sistem ekstraksi informasi di masa mendatang.

D. Tantangan Umum dalam Ekstraksi Informasi

Walaupun ekstraksi informasi menjanjikan manfaat yang luas dan mendalam bagi berbagai sektor, implementasinya di lapangan tidak lepas dari berbagai tantangan yang kompleks. Tantangan-tantangan ini muncul baik dari sifat dasar bahasa alami yang ambigu dan kontekstual, maupun dari keterbatasan teknis yang dihadapi dalam pengembangan sistem. Pemahaman terhadap tantangan ini menjadi penting agar proses perancangan, pelatihan, dan evaluasi sistem ekstraksi informasi dapat dilakukan secara realistis dan terarah.

Salah satu tantangan utama dalam ekstraksi informasi adalah ambiguitas semantik dalam bahasa alami. Kata atau frasa yang sama dapat memiliki makna yang berbeda tergantung pada konteksnya. Sebagai contoh, kata “Jakarta” dapat merujuk pada nama kota, lokasi kejadian, atau bahkan institusi (misalnya "Pemerintah Jakarta"). Tanpa pemahaman konteks yang mendalam, sistem ekstraksi informasi bisa salah mengklasifikasikan entitas atau gagal mengenali hubungan antar bagian dalam teks. Tantangan ini diperparah dengan kehadiran polisemi, idiom, dan gaya bahasa figuratif yang umum dalam tulisan naratif seperti berita atau media sosial.

Tantangan berikutnya berkaitan dengan struktur dan format teks yang sangat bervariasi. Dokumen teks dapat hadir dalam bentuk paragraf panjang, daftar, tabel, bahkan percakapan informal. Dalam konteks bahasa Indonesia, misalnya, struktur kalimat bisa sangat fleksibel dan tidak selalu mengikuti pola subjek-predikat-objek yang eksplisit. Variasi ini menuntut sistem EI untuk memiliki fleksibilitas yang tinggi dalam mengenali pola-pola linguistik. Selain itu, kehadiran kesalahan ketik, singkatan tidak baku, dan penggunaan bahasa campuran (misalnya penggunaan istilah asing dalam kalimat Bahasa Indonesia) semakin memperumit proses ekstraksi.

Dari sisi sumber daya, keterbatasan data anotasi merupakan hambatan besar, terutama dalam konteks bahasa yang tidak memiliki banyak corpus anotasi publik. Data anotasi diperlukan untuk melatih model statistik atau pembelajaran mesin, dan penyusunannya membutuhkan keahlian linguistik serta waktu yang tidak sedikit. Tanpa data pelatihan yang memadai, model akan kesulitan mencapai akurasi yang baik dan rentan terhadap bias. Hal ini menjadikan pendekatan rule-based masih menjadi pilihan dalam banyak proyek ekstraksi informasi untuk Bahasa Indonesia, setidaknya sampai data berkualitas tersedia dalam jumlah yang cukup.

Tantangan lain yang juga penting adalah bagaimana mengukur keberhasilan sistem ekstraksi informasi secara objektif. Pengukuran performa sistem tidak sesederhana menghitung akurasi, karena sistem perlu dievaluasi berdasarkan kesesuaian antara hasil ekstraksi dengan anotasi referensi, yang disebut sebagai ground truth. Dalam proses ini, metrik seperti precision, recall, dan F1-score digunakan untuk menilai kualitas ekstraksi. Namun demikian, proses evaluasi juga rawan terhadap perbedaan

interpretasi antar anotator, sehingga diperlukan standar dan pedoman anotasi yang konsisten agar hasil evaluasi dapat diandalkan.

Selain itu, aspek keberlanjutan dan pemeliharaan sistem ekstraksi informasi juga sering kali menjadi tantangan jangka panjang. Bahasa dan gaya komunikasi manusia senantiasa berkembang. Istilah-istilah baru muncul, struktur kalimat berubah, dan tren penggunaan bahasa di media sosial atau dokumen resmi ikut bertransformasi. Sistem EI yang tidak dirancang untuk beradaptasi terhadap perubahan ini akan mengalami degradasi performa dalam waktu relatif singkat. Oleh karena itu, sistem yang baik harus dibangun secara modular dan fleksibel, serta memungkinkan pembaruan model, aturan, atau kamus entitas secara berkala.

Keamanan dan privasi juga menjadi isu penting dalam aplikasi EI, khususnya jika sistem digunakan untuk mengekstrak informasi dari dokumen yang mengandung data pribadi, rahasia medis, atau informasi sensitif lainnya. Implementasi ekstraksi informasi harus mempertimbangkan regulasi seperti UU Perlindungan Data Pribadi, dan memastikan bahwa proses penyimpanan dan pemrosesan hasil ekstraksi dilakukan secara etis dan aman.

Sebagai tanggapan terhadap tantangan-tantangan tersebut, banyak peneliti dan pengembang mengadopsi pendekatan hibrida yang menggabungkan keunggulan metode berbasis aturan dan statistik. Selain itu, perkembangan teknologi pre-trained language models seperti BERT dan ChatGPT juga membuka peluang baru untuk membangun sistem EI yang lebih kontekstual, bahkan dalam bahasa dengan sumber daya terbatas,

asalkan dilakukan dengan penyesuaian domain dan bahasa yang memadai.

Secara keseluruhan, ekstraksi informasi merupakan bidang yang menjanjikan sekaligus menantang. Kompleksitas bahasa, keterbatasan sumber daya, dan tuntutan aplikasi nyata menjadikan pengembangan sistem EI sebagai proses iteratif yang memerlukan ketelitian, eksperimen, dan adaptasi berkelanjutan. Namun, dengan pendekatan yang tepat, tantangan-tantangan tersebut dapat menjadi pijakan untuk membangun solusi yang lebih cerdas, akurat, dan bermanfaat bagi masyarakat.

E. Penggunaan Ekstraksi Informasi di Dunia Nyata

Ekstraksi informasi telah diaplikasikan secara luas dalam berbagai bidang industri dan sektor publik. Keunggulannya dalam mengolah teks tak terstruktur menjadi data yang terorganisasi membuat teknologi ini menjadi komponen penting dalam sistem cerdas berbasis bahasa alami. Dalam praktiknya, ekstraksi informasi berperan sebagai jembatan antara data mentah dan pengambilan keputusan berbasis informasi. Berbagai contoh kasus di dunia nyata memperlihatkan bagaimana sistem EI memberikan dampak nyata terhadap efisiensi, akurasi, dan skala pengolahan informasi dalam konteks yang berbeda-beda.

Salah satu bidang utama yang memanfaatkan ekstraksi informasi secara intensif adalah jurnalisme dan pelaporan berita. Dalam situasi darurat seperti bencana alam, sistem EI digunakan untuk mengekstrak informasi penting secara otomatis dari artikel berita dan laporan lapangan. Informasi seperti lokasi kejadian, jenis bencana, jumlah korban, serta waktu dan tanggal dapat diambil secara sistematis dari kumpulan berita yang besar. Dengan

demikian, lembaga tanggap darurat dan organisasi kemanusiaan dapat memperoleh informasi yang relevan secara cepat untuk mendukung pengambilan keputusan atau penyusunan respons. Misalnya, dari kumpulan berita pasca-gempa, sistem EI dapat secara otomatis membentuk ringkasan lokasi terdampak dan estimasi korban sebagai input bagi pemetaan krisis.

Dalam sektor layanan kesehatan, ekstraksi informasi dimanfaatkan untuk mengekstrak data klinis dari rekam medis elektronik. Informasi seperti diagnosis, prosedur medis, nama obat, serta hasil laboratorium dapat diambil dari catatan dokter yang ditulis dalam bahasa alami. Sistem EI membantu menyederhanakan analisis rekam medis secara masif dan memungkinkan pelacakan kondisi pasien atau efektivitas pengobatan dalam skala populasi. Sebagai contoh, rumah sakit dapat memanfaatkan sistem ini untuk mengidentifikasi pasien dengan penyakit tertentu yang memenuhi kriteria tertentu untuk uji klinis atau program perawatan khusus, tanpa harus membaca satu per satu seluruh rekam medis.

Di sektor e-commerce dan layanan pelanggan, EI digunakan untuk menganalisis ulasan produk dan umpan balik pelanggan. Sistem dapat mengidentifikasi nama produk, fitur yang disebutkan, keluhan atau pujian, dan bahkan sentimen yang terkandung dalam teks ulasan. Hasil ekstraksi ini memungkinkan perusahaan untuk mengukur kepuasan pelanggan secara otomatis, mengenali tren keluhan yang berulang, serta menyempurnakan produk dan layanan mereka berdasarkan data yang bersumber langsung dari pengguna. Lebih lanjut, sistem dapat menyusun ringkasan otomatis dari ribuan ulasan menjadi informasi ringkas

yang memudahkan manajer atau analis dalam mengambil kebijakan.

Dalam bidang hukum dan regulasi, ekstraksi informasi telah mulai digunakan untuk mengekstrak entitas hukum seperti nama penggugat, tergugat, tanggal sidang, serta keputusan hukum dari dokumen peradilan. Di banyak negara, dokumen pengadilan dan kontrak bisnis dipublikasikan dalam bentuk teks naratif panjang, sehingga menyulitkan pencarian dan analisis secara manual. Sistem EI memungkinkan pembuatan indeks otomatis, pencarian entitas berdasarkan peran hukum, serta deteksi otomatis terhadap klausul tertentu dalam perjanjian. Dengan pendekatan ini, firma hukum atau regulator dapat mengelola ribuan dokumen hukum dengan efisiensi yang sebelumnya tidak mungkin dicapai.

Contoh lain datang dari media sosial, yang menjadi sumber data sangat dinamis dan tidak terstruktur. Dalam konteks ini, ekstraksi informasi digunakan untuk memantau opini publik, mengenali entitas yang sedang menjadi perhatian, dan mengidentifikasi peristiwa yang sedang berkembang. Misalnya, pada saat terjadi peristiwa penting seperti pemilu atau unjuk rasa, sistem EI dapat digunakan untuk memetakan siapa tokoh yang disebut, apa lokasi utama kegiatan, serta bagaimana persepsi publik terhadap kejadian tersebut. Dengan demikian, pihak-pihak seperti media, pemerintah, atau peneliti sosial dapat memperoleh gambaran situasi secara real-time.

Contoh-contoh di atas menggambarkan bahwa ekstraksi informasi memiliki aplikasi yang sangat luas, mulai dari kebutuhan analitik data historis hingga pemantauan kejadian secara langsung. Keberhasilan penerapan EI dalam kasus nyata tidak hanya bergantung pada kualitas algoritma, tetapi juga pada pemahaman

yang mendalam terhadap domain aplikasi, struktur bahasa yang digunakan, serta cara data teks dihasilkan dan dikonsumsi. Dalam banyak kasus, penerapan sistem EI diawali dengan studi kasus terbatas, kemudian diperluas skalanya seiring dengan perbaikan model dan peningkatan kualitas hasil ekstraksi.

Seiring dengan kemajuan teknologi dan semakin terintegrasinya teks dalam sistem informasi digital, permintaan akan kemampuan ekstraksi informasi yang akurat dan kontekstual akan terus meningkat. Oleh karena itu, memahami bagaimana sistem EI bekerja dalam konteks dunia nyata bukan hanya penting dari sisi akademik, tetapi juga krusial bagi praktisi dan pengembang sistem yang ingin menciptakan solusi informasi cerdas yang relevan dengan kebutuhan masa kini.

Latihan

A. Pertanyaan Pemahaman Konsep

1. Jelaskan dengan kata-kata Anda sendiri apa yang dimaksud dengan *Information Extraction* (IE).
2. Sebutkan minimal tiga perbedaan mendasar antara IE dan *Information Retrieval* (IR).
3. Mengapa IE dianggap sebagai komponen penting dalam sistem *Artificial Intelligence* modern?
4. Bagaimana peran NLP dalam mendukung proses ekstraksi informasi dari teks?
5. Jelaskan contoh nyata penerapan IE dalam kehidupan sehari-hari yang Anda ketahui.
6. Apa perbedaan mendasar antara pendekatan *rule-based* dan *machine learning* dalam IE?
7. Sebutkan tiga tantangan utama dalam penerapan IE untuk Bahasa Indonesia dan berikan solusi singkatnya.

B. Studi Kasus Sederhana

Diberikan potongan berita berikut:

"Gempa berkekuatan 6,2 SR mengguncang Kabupaten Cianjur pada Senin pagi dan menyebabkan puluhan rumah rusak."

Tugas:

1. Identifikasi elemen informasi penting dari teks di atas (jenis peristiwa, lokasi, waktu, dampak).
2. Buat tabel sederhana yang memuat hasil ekstraksi informasi.
3. Jelaskan bagaimana sistem komputer dapat mengekstrak informasi tersebut secara otomatis dengan pendekatan *rule-based*.

C. Tugas Mandiri

1. Diskusikan perbedaan manfaat IE dalam dunia

akademik dan industri.

2. Cari satu artikel atau berita terkini di internet, kemudian jelaskan bagaimana proses IE dapat digunakan untuk menganalisis isi berita tersebut.
3. Buat bagan sederhana yang menggambarkan hubungan antara *Text Mining*, *NLP*, dan *Information Extraction*.

BAB 2
TEXT PRE-PROCESSING

Tujuan Pembelajaran

Setelah mempelajari Bab 2, pembaca diharapkan mampu:

1. Menjelaskan konsep dan fungsi preprocessing teks sebagai tahap awal dalam pipeline ekstraksi informasi.
2. Menguraikan tahapan utama preprocessing, meliputi *text cleaning*, tokenisasi, *stopword removal*, stemming, lemmatization, dan normalisasi teks.
3. Menganalisis pentingnya pembersihan dan standardisasi teks untuk meningkatkan akurasi sistem NLP dan IE.
4. Membedakan peran setiap teknik preprocessing berdasarkan karakteristik data teks (formal, media sosial, dokumen hukum, dll.).
5. Menerapkan metode tokenisasi dan stopword removal menggunakan contoh kalimat berbahasa Indonesia.
6. Menerapkan teknik stemming atau lemmatization dengan pendekatan rule-based atau library NLP.
7. Mengevaluasi dampak preprocessing terhadap kualitas hasil ekstraksi informasi.

Pendahuluan

Preprocessing teks merupakan tahap awal yang sangat penting dalam setiap pipeline ekstraksi informasi. Tahapan ini bertujuan untuk membersihkan, menormalkan, dan mempersiapkan data teks agar dapat diolah secara sistematis oleh algoritma atau model ekstraksi. Tanpa proses praproses yang memadai, akurasi dan performa dari sistem ekstraksi informasi dapat menurun secara signifikan, karena data teks

sering kali mengandung variasi yang tinggi, kebisingan (noise), dan struktur bahasa yang tidak konsisten.

Pada dasarnya, preprocessing teks adalah bentuk transformasi dari teks mentah menjadi representasi linguistik yang lebih seragam dan siap diproses secara komputasional. Proses ini tidak hanya berlaku dalam konteks ekstraksi informasi, tetapi juga dalam hampir semua aplikasi NLP lainnya seperti klasifikasi teks, analisis sentimen, machine translation, dan summarization. Oleh karena itu, pemahaman yang mendalam mengenai tahap-tahap preprocessing menjadi kompetensi dasar yang harus dikuasai oleh siapa pun yang ingin mendalami pengolahan teks secara otomatis.

A. Konsep dan Urutan Preprocessing

Preprocessing teks dapat diibaratkan sebagai proses pembersihan dan standardisasi bahan mentah sebelum dimasak. Teks yang diperoleh dari berbagai sumber seperti berita, media sosial, laporan resmi, atau transkrip percakapan, umumnya mengandung unsur-unsur yang tidak diperlukan, tidak konsisten, atau bahkan menyesatkan jika diproses apa adanya. Proses ini bertujuan untuk mengurangi kompleksitas teks dengan cara menghilangkan elemen yang tidak relevan, menyamakan format, dan menguraikan struktur bahasa agar lebih mudah ditangani oleh mesin.

Secara umum, preprocessing terdiri dari serangkaian langkah yang saling berurutan dan bersifat modular. Urutan tersebut dapat bervariasi tergantung pada tujuan analisis dan karakteristik data, namun biasanya mencakup tahap-tahap sebagai berikut: pembersihan teks (text cleaning), tokenisasi, penghapusan stopword, stemming atau lemmatization, dan normalisasi. Beberapa aplikasi juga menambahkan proses lain seperti pengenalan entitas,

pelabelan sintaksis (POS tagging), atau anotasi morfologis sebagai bagian dari preprocessing lanjutan.

Tahap pertama, pembersihan teks, melibatkan penghapusan karakter-karakter tidak relevan seperti tanda baca, simbol khusus, angka yang tidak bermakna, URL, dan kode HTML. Tujuannya adalah untuk menyederhanakan teks dan menghilangkan elemen yang tidak membawa makna linguistik penting. Misalnya, dalam analisis berita, keberadaan kode artikel atau simbol pemformatan sering kali hanya menjadi gangguan dalam proses ekstraksi.

Tahap kedua adalah tokenisasi, yaitu proses memecah teks menjadi unit-unit terkecil yang disebut token. Token ini bisa berupa kata, frasa pendek, atau dalam beberapa aplikasi, karakter. Tokenisasi menjadi fondasi bagi hampir seluruh teknik NLP, karena banyak algoritma bekerja berdasarkan satuan token, bukan kalimat utuh. Dalam Bahasa Indonesia, tokenisasi memiliki tantangan tersendiri karena adanya bentuk kata majemuk, partikel, serta struktur kata yang sering kali melebur secara fonetik.

Setelah token diperoleh, langkah selanjutnya adalah menghapus kata-kata yang dianggap tidak memiliki makna spesifik terhadap konteks analisis, yang disebut sebagai stopword. Kata seperti “dan”, “yang”, “atau”, “adalah”, meskipun memiliki fungsi gramatikal penting, biasanya tidak memberikan kontribusi makna signifikan dalam ekstraksi entitas atau relasi. Penghapusan stopword bertujuan untuk mengurangi beban pemrosesan dan menyoroti kata-kata kunci yang lebih informatif.

Langkah berikutnya adalah stemming atau lemmatization. Keduanya bertujuan untuk mengembalikan bentuk kata ke bentuk dasarnya, namun

dengan pendekatan yang berbeda. Stemming cenderung menggunakan pemotongan aturan (rule-based trimming), sementara lemmatization mempertimbangkan morfologi dan konteks linguistik. Dalam Bahasa Indonesia, proses ini menghadapi tantangan tersendiri karena kekayaan morfologi dan bentuk imbuhan yang kompleks. Tanpa proses ini, sistem EI dapat salah menganggap dua bentuk kata yang sebenarnya sama sebagai entitas yang berbeda.

Terakhir, normalisasi dilakukan untuk menyamakan bentuk ekspresi dalam teks. Misalnya, berbagai cara penulisan tanggal ("1 Jan 2023", "01/01/23", "1 Januari 2023") dapat dinormalisasi ke dalam format standar ISO. Demikian pula dengan penulisan angka, alamat, atau istilah teknis tertentu yang perlu distandarkan agar dapat dikenali dengan benar oleh sistem.

Secara keseluruhan, urutan preprocessing tidaklah kaku, tetapi harus dirancang secara kontekstual sesuai dengan tujuan analisis dan sumber data. Untuk kasus-kasus yang sangat domain-spesifik, seperti analisis dokumen hukum atau berita bencana, preprocessing dapat mencakup tahapan tambahan seperti pelabelan struktur dokumen, ekstraksi metadata, atau pembersihan berbasis pola tertentu. Dalam semua kasus, keberhasilan preprocessing memiliki dampak langsung terhadap kualitas ekstraksi informasi di tahap-tahap berikutnya.

B. Tokenisasi: Pengertian dan Teknik

Tokenisasi merupakan salah satu tahapan paling awal dan mendasar dalam proses preprocessing teks. Istilah ini merujuk pada proses pemecahan teks menjadi satuan-satuan linguistik terkecil yang disebut token. Token dalam konteks ini bisa berupa kata, frasa, tanda baca, atau bahkan karakter, tergantung pada jenis analisis

yang akan dilakukan dan model linguistik yang digunakan.

Secara konseptual, tokenisasi bertujuan untuk memisahkan teks yang semula berupa rangkaian karakter tak terputus menjadi unit-unit yang memiliki makna tertentu atau fungsi linguistik yang bisa dikenali oleh sistem. Sebagai contoh, kalimat "Pemerintah Indonesia meluncurkan program vaksinasi massal" akan dipecah menjadi deretan token seperti: "Pemerintah", "Indonesia", "meluncurkan", "program", "vaksinasi", dan "massal." Dengan proses ini, sistem pemrosesan bahasa dapat menganalisis struktur dan isi teks secara lebih granular.

Dalam konteks ekstraksi informasi, tokenisasi memiliki fungsi penting karena sebagian besar proses lanjutan seperti pelabelan entitas (Named Entity Recognition), analisis sintaksis (Part-of-Speech Tagging), dan ekstraksi relasi, bergantung pada representasi token dari sebuah dokumen. Apabila proses tokenisasi tidak dilakukan dengan akurat, maka akan muncul kesalahan sistemik yang berdampak pada seluruh proses analisis berikutnya. Oleh karena itu, kualitas tokenisasi sangat menentukan akurasi ekstraksi informasi secara keseluruhan.

Tokenisasi bukanlah proses yang sepele, terutama dalam bahasa-bahasa alami seperti Bahasa Indonesia yang memiliki fleksibilitas tinggi dalam struktur kata dan kalimat. Bahasa Indonesia dikenal memiliki banyak bentuk kata turunan melalui imbuhan (prefiks, sufiks, infiks, dan konfiks), serta berbagai bentuk kata majemuk dan partikel. Tantangan utama dalam tokenisasi Bahasa Indonesia adalah dalam menangani gabungan kata yang tidak dipisahkan oleh spasi, seperti "berkeberatan," "mengusahakan," atau "memberitahukan." Selain itu,

partikel seperti “lah,” “pun,” “kah,” yang sering menempel pada kata, juga dapat menyulitkan proses pemisahan jika tidak ditangani dengan aturan linguistik yang tepat.

Dalam praktiknya, terdapat berbagai pendekatan teknis dalam melakukan tokenisasi, yang dapat dikategorikan menjadi dua jenis utama: berbasis aturan (rule-based) dan berbasis statistik atau model pembelajaran mesin. Pendekatan rule-based mengandalkan pola reguler seperti spasi, tanda baca, dan aturan gramatikal untuk memisahkan token. Pendekatan ini relatif sederhana dan cepat, namun dapat menjadi tidak akurat ketika berhadapan dengan bahasa informal atau teks yang mengandung banyak variasi morfologis.

Sementara itu, pendekatan berbasis statistik atau pembelajaran mesin menggunakan model bahasa yang telah dilatih untuk mengenali batas token berdasarkan data pelatihan. Metode ini biasanya lebih fleksibel dan dapat menyesuaikan diri dengan variasi bahasa, namun memerlukan data pelatihan yang cukup besar dan representatif. Dalam beberapa kasus, digunakan juga metode berbasis subword tokenization seperti Byte Pair Encoding (BPE) atau WordPiece, yang umum diterapkan pada model bahasa besar seperti BERT. Metode ini membagi kata menjadi unit-unit yang lebih kecil berdasarkan frekuensi kemunculan, sehingga dapat menangani kata baru atau bentuk tidak umum secara lebih baik.

Beberapa alat bantu (tools) untuk tokenisasi telah tersedia dan umum digunakan di komunitas NLP, seperti NLTK, spaCy, Stanza, dan IndoNLP untuk Bahasa Indonesia. Alat-alat ini menyediakan tokenisasi yang cukup akurat untuk keperluan umum, namun sering kali

perlu disesuaikan jika digunakan pada domain yang sangat spesifik, seperti dokumen medis atau hukum.

Perlu dicatat bahwa tokenisasi juga melibatkan pengambilan keputusan tentang bagaimana menangani tanda baca, angka, dan simbol lainnya. Misalnya, apakah tanda koma akan dijadikan token tersendiri, atau apakah angka seperti "Rp10.000" akan dipisahkan menjadi "Rp", "10", dan "000." Keputusan-keputusan ini harus disesuaikan dengan tujuan ekstraksi informasi yang hendak dilakukan. Untuk sistem yang mengutamakan entitas seperti harga atau waktu, angka dan satuannya sebaiknya tidak dipisah agar konteks tetap utuh.

Dalam beberapa kasus, tokenisasi juga perlu mempertimbangkan unit linguistik yang lebih besar seperti frasa nama (contoh: "Bank Indonesia"), atau entitas multi-kata. Hal ini sering disebut sebagai multi-word expression recognition, yang biasanya dilakukan setelah tahap tokenisasi dasar, namun bisa pula dimasukkan sebagai bagian dari tokenisasi lanjutan.

Dengan demikian, tokenisasi bukan hanya tahap teknis yang bersifat mekanis, tetapi merupakan proses linguistik awal yang menentukan keberhasilan analisis bahasa alami, khususnya dalam sistem ekstraksi informasi. Pemilihan teknik tokenisasi harus dilakukan secara cermat, memperhatikan karakteristik bahasa dan domain, serta mempertimbangkan trade-off antara kecepatan dan akurasi dalam konteks aplikasi yang dibangun.

C. Stopword Removal dan Filtering Kata Relevan

Setelah teks diubah menjadi satuan token melalui proses tokenisasi, langkah berikutnya dalam preprocessing adalah mengidentifikasi dan menyaring

kata-kata yang dianggap tidak memiliki nilai informasi yang signifikan dalam konteks ekstraksi. Tahap ini dikenal sebagai stopword removal, yaitu proses menghapus kata-kata umum yang frekuensi kemunculannya tinggi namun kontribusinya terhadap makna spesifik suatu dokumen sangat rendah. Kata-kata ini biasanya terdiri dari artikel, konjungsi, pronomina, atau partikel yang secara gramatikal penting, tetapi secara semantik tidak membawa informasi esensial terhadap isi dokumen.

Dalam Bahasa Indonesia, contoh dari stopword yang umum adalah “yang”, “dan”, “atau”, “di”, “ke”, “dari”, “sebagai”, “adalah”, serta berbagai bentuk partikel seperti “pun”, “lah”, “kah.” Kata-kata ini tentu diperlukan untuk membentuk kalimat yang benar secara tata bahasa, tetapi dalam analisis teks otomatis, terutama dalam tugas seperti ekstraksi entitas, relasi, atau peristiwa, keberadaannya sering kali justru menambah beban pemrosesan tanpa menambah nilai analitik.

Tujuan utama dari penghapusan stopword adalah untuk menyederhanakan representasi teks dan meningkatkan efisiensi proses pemrosesan informasi berikutnya. Dengan mengurangi jumlah token yang harus dianalisis, sistem dapat fokus pada kata-kata yang secara statistik lebih relevan atau bermakna. Hal ini tidak hanya menghemat waktu komputasi, tetapi juga dapat meningkatkan akurasi dalam model pembelajaran mesin atau sistem ekstraksi berbasis aturan, karena mengurangi kemungkinan "gangguan" dari kata-kata tidak penting.

Namun, proses stopword removal tidak selalu dapat dilakukan secara sembarangan atau menggunakan daftar baku tanpa penyesuaian. Konteks dan tujuan analisis memainkan peran penting dalam menentukan apakah sebuah kata sebaiknya dihapus atau dipertahankan.

Misalnya, dalam sistem ekstraksi opini atau analisis sentimen, kata "tidak" atau "bukan" adalah kata yang sering kali dianggap sebagai stopword dalam pendekatan umum, padahal dalam konteks sentimen, kata tersebut justru sangat penting karena dapat membalikkan makna dari kata yang mengikutinya. Contoh: "layanan ini tidak memuaskan" memiliki arti yang sangat berbeda dengan "layanan ini memuaskan."

Oleh karena itu, filtering kata harus dilakukan secara selektif. Selain berdasarkan daftar kata yang umum, proses filtering juga dapat melibatkan analisis frekuensi kata dalam korpus tertentu, pengukuran statistik seperti term frequency-inverse document frequency (TF-IDF), atau bahkan hasil pelabelan entitas dan fitur linguistik lainnya. Dalam pendekatan modern, filtering dapat dikombinasikan dengan pemilihan fitur secara otomatis melalui algoritma pembelajaran mesin, di mana sistem akan belajar sendiri kata mana yang memiliki kontribusi terbesar terhadap hasil klasifikasi atau ekstraksi.

Teknik filtering yang lebih canggih juga dapat diterapkan untuk menghilangkan kata-kata yang terlalu sering muncul namun tidak bersifat deskriptif dalam domain tertentu. Misalnya, dalam domain berita politik, kata "pemerintah" mungkin terlalu umum untuk dibedakan dan tidak memberi nilai spesifik terhadap satu dokumen. Dalam kasus ini, sistem dapat dirancang untuk mengenali istilah semacam itu sebagai kata-kata yang perlu disaring dari proses ekstraksi awal, dan hanya mempertahankan entitas yang lebih spesifik seperti "Kementerian Kesehatan" atau "Presiden Joko Widodo."

Perlu juga dicatat bahwa tidak semua sistem ekstraksi informasi melakukan stopword removal secara eksplisit. Dalam pendekatan berbasis pembelajaran mesin

atau deep learning, banyak model modern seperti BERT, RoBERTa, dan sejenisnya tidak menghapus stopword sama sekali. Hal ini karena model tersebut mampu memahami konteks kata dalam kalimat secara keseluruhan, termasuk kata-kata fungsi yang sebelumnya dianggap tidak informatif. Meskipun demikian, dalam sistem rule-based atau sistem ringan yang berfokus pada efisiensi, stopword removal tetap merupakan komponen penting dalam preprocessing.

Dalam pengembangannya, daftar stopword dapat dikustomisasi sesuai kebutuhan proyek. Banyak pustaka NLP seperti NLTK, spaCy, dan Sastrawi (untuk Bahasa Indonesia) menyediakan daftar stopword bawaan yang dapat dijadikan acuan awal. Namun, pengembang sistem tetap dianjurkan untuk mengevaluasi daftar tersebut terhadap korpus data aktual yang digunakan, dan melakukan penyesuaian sesuai karakteristik teks dan tujuan ekstraksi.

Dengan mempertimbangkan kompleksitas ini, proses filtering kata yang tidak relevan bukan hanya kegiatan teknis, melainkan bagian penting dari desain sistem ekstraksi informasi yang cermat. Kemampuan untuk memilih kata mana yang dipertahankan dan mana yang diabaikan menjadi salah satu indikator kecerdasan dari sistem preprocessing itu sendiri, karena hal tersebut memengaruhi langsung kualitas hasil akhir dari proses ekstraksi.

D. Stemming vs Lemmatization

Dalam proses preprocessing teks, salah satu tahap penting setelah tokenisasi dan filtering adalah menyederhanakan bentuk kata ke bentuk dasarnya. Tahapan ini dikenal sebagai stemming atau

lemmatization. Keduanya bertujuan mengurangi variasi morfologis kata, sehingga kata-kata dengan makna dasar yang sama tidak dianggap berbeda oleh sistem pemrosesan teks. Misalnya, kata "berlari", "berlarian", dan "pelari" semuanya memiliki akar kata yang sama, yaitu "lari". Mengidentifikasi kesamaan ini memungkinkan sistem ekstraksi informasi untuk melakukan analisis yang lebih akurat dan konsisten.

Meskipun bertujuan serupa, stemming dan lemmatization memiliki pendekatan dan karakteristik yang berbeda secara mendasar. Stemming adalah proses menghapus imbuhan pada kata untuk menemukan bentuk dasarnya secara heuristik, sering kali tanpa memperhatikan konteks linguistik atau struktur morfologis formal. Proses ini biasanya dilakukan dengan metode berbasis aturan sederhana atau algoritma pemotongan. Akibatnya, hasil stemming tidak selalu menghasilkan kata yang benar secara gramatikal. Misalnya, kata "berlarian" dapat dipangkas menjadi "lari", tetapi dalam beberapa kasus seperti "membayangkan", hasil stemming bisa menjadi "bayang" yang memang bermakna, atau menjadi bentuk yang tidak valid seperti "membayang".

Pendekatan stemming umumnya lebih cepat dan ringan secara komputasi, sehingga cocok untuk aplikasi skala besar atau sistem yang memprioritaskan efisiensi. Namun, karena tidak memperhatikan aturan linguistik secara penuh, stemming dapat menimbulkan ambiguitas atau kekeliruan dalam hasil ekstraksi, khususnya dalam bahasa dengan sistem morfologi yang kompleks seperti Bahasa Indonesia.

Sementara itu, lemmatization adalah proses yang lebih cermat, di mana kata dikembalikan ke bentuk lema

atau bentuk dasarnya yang sesuai dengan kosakata bahasa (dictionary form). Lemmatisasi mempertimbangkan kelas kata (kata benda, kata kerja, dsb.), struktur gramatikal, serta konteks sintaktis untuk menentukan bentuk dasar yang benar. Proses ini memerlukan leksikon atau basis data linguistik yang lengkap, serta sering kali melibatkan Part-of-Speech tagging untuk mengidentifikasi peran kata dalam kalimat. Sebagai contoh, kata "berjalan" akan dilemmatkan menjadi "jalan" sebagai kata kerja, sedangkan "jalanan" akan dilemmatkan juga menjadi "jalan" sebagai kata benda.

Dalam Bahasa Indonesia, proses lemmatization menghadapi tantangan tersendiri karena sistem imbuhan yang kaya dan fleksibel. Bahasa Indonesia memiliki beragam prefiks (ber-, me-, di-, ke-, se-), sufiks (-kan, -i, -an), dan gabungan konfiks (memper-, keber-, dsb.) yang menimbulkan banyak variasi kata turunan. Oleh karena itu, lemmatization dalam bahasa ini memerlukan analisis morfologi yang cukup kompleks dan mendalam. Beberapa alat bantu telah dikembangkan untuk mendukung tugas ini, seperti lemmatizer dari Sastrawi, MorphInd, atau sistem berbasis neural network yang dilatih khusus untuk bahasa Indonesia.

Dalam konteks ekstraksi informasi, baik stemming maupun lemmatization dapat digunakan tergantung pada kebutuhan spesifik sistem. Jika sistem memerlukan kecepatan tinggi dan hanya melakukan analisis permukaan seperti pencocokan kata kunci, maka stemming mungkin sudah cukup memadai. Namun, untuk aplikasi yang membutuhkan akurasi tinggi dan memahami makna kontekstual seperti dalam ekstraksi relasi atau identifikasi peristiwa lemmatization lebih

disarankan karena menghasilkan representasi linguistik yang lebih andal.

Sebagai ilustrasi, dalam sebuah sistem ekstraksi kejadian kriminal, kata "penembakan", "ditembak", dan "menembak" semuanya dapat dilemmatkan ke bentuk dasar "tembak" untuk dianalisis sebagai tindakan yang sama. Jika menggunakan stemming biasa, hasilnya bisa menjadi bentuk tidak baku seperti "tembak" dan "temb" atau "menemb", yang justru membingungkan sistem klasifikasi atau pencocokan pola berbasis aturan.

Namun demikian, perlu dicatat bahwa penggunaan lemmatization yang terlalu agresif juga dapat menghapus perbedaan semantik penting. Misalnya, kata "pengunjung" dan "kunjungan" memiliki bentuk dasar yang sama ("kunjung") tetapi maknanya berbeda dalam struktur kalimat. Oleh karena itu, pemilihan antara stemming dan lemmatization harus mempertimbangkan konteks domain dan jenis informasi yang ingin diekstrak.

Sebagai kesimpulan, kedua teknik ini adalah alat penting dalam preprocessing teks yang memiliki peran besar dalam menyederhanakan dan menstandarkan representasi kata. Pemahaman yang baik terhadap perbedaan antara stemming dan lemmatization, serta dampaknya terhadap sistem ekstraksi informasi, akan sangat membantu dalam merancang pipeline NLP yang lebih akurat dan efisien, khususnya untuk bahasa dengan karakteristik morfologis kompleks seperti Bahasa Indonesia.

E. Normalisasi dan Cleaning Data Teks

Normalisasi dan cleaning teks adalah tahap akhir dalam rangkaian preprocessing yang bertujuan untuk menyelaraskan bentuk representasi kata dan

menghilangkan elemen-elemen yang dapat mengganggu analisis linguistik. Tahapan ini sering dianggap sebagai "perapihan akhir" sebelum teks diproses lebih lanjut oleh sistem ekstraksi informasi, klasifikasi, atau model pembelajaran mesin. Meski tampak sederhana, tahap ini memiliki dampak signifikan terhadap konsistensi dan kualitas hasil analitik, khususnya dalam domain-domain yang datanya sangat beragam, tidak baku, atau berasal dari sumber yang tidak terkontrol seperti media sosial dan input pengguna (user-generated content).

Normalisasi teks merujuk pada proses mengubah berbagai bentuk penulisan kata atau frasa menjadi format standar yang seragam. Dalam praktiknya, teks alami mengandung banyak variasi penulisan yang bermakna sama, seperti penulisan tanggal ("23/02/2022", "23 Feb 2022", "23 Februari 2022"), penggunaan angka ("dua puluh lima" vs "25"), atau perbedaan kapitalisasi ("Jakarta" vs "jakarta"). Tanpa normalisasi, sistem akan memperlakukan bentuk-bentuk tersebut sebagai entitas atau token yang berbeda, padahal maknanya sama. Normalisasi juga mencakup konversi kata-kata tidak baku atau ejaan tidak standar, seperti "nggak" menjadi "tidak", "gak" menjadi "tidak", atau "trms" menjadi "terima kasih". Hal ini sangat penting dalam analisis teks informal yang umum dijumpai pada media sosial, forum daring, atau aplikasi perpesanan.

Sementara itu, cleaning teks mencakup proses penghapusan karakter, simbol, atau elemen non-linguistik yang tidak dibutuhkan dalam analisis. Proses ini biasanya mencakup penghilangan:

1. Tanda baca yang tidak relevan (kecuali jika diperlukan sebagai fitur sintaksis),

2. Angka atau kode acak (jika tidak memiliki arti semantik),
3. URL, alamat email, tag HTML atau markup lain,
4. Emoji dan simbol grafis (tergantung konteks dan kebutuhan),
5. Whitespace berlebih, karakter ganda, dan bentuk duplikasi tidak wajar.

Pada teks hasil OCR (Optical Character Recognition), cleaning menjadi lebih kompleks karena dapat muncul karakter asing, huruf rusak, atau kesalahan pembacaan dokumen fisik yang perlu diperbaiki secara manual atau dengan koreksi otomatis berbasis model. Pada teks dari media sosial, sering dijumpai pengulangan huruf seperti "baguuuussssss" yang perlu disesuaikan menjadi "bagus" agar dapat dikenali sebagai kata standar.

Normalisasi juga erat kaitannya dengan validasi ejaan dan pengecekan kamus. Dalam beberapa sistem, digunakan kamus standar Bahasa Indonesia (KBBI) atau kamus domain khusus untuk memeriksa apakah kata-kata dalam teks sesuai dengan kosakata yang dikenal. Kata-kata yang tidak dikenal dapat dihapus, disarankan koreksinya, atau ditandai untuk dianalisis lebih lanjut. Di sisi lain, sistem berbasis pembelajaran mesin terkini seperti BERT atau GPT memiliki kemampuan untuk menangani bentuk tidak baku secara lebih toleran, meskipun tetap akan memperoleh hasil lebih baik jika teks telah dinormalisasi sebelumnya.

Dalam ekstraksi informasi, normalisasi dan cleaning memiliki peran penting dalam meningkatkan precision sistem. Sebagai contoh, ekstraksi lokasi dari teks yang menuliskan "jakarta", "JKT", atau "Ibukota" memerlukan proses normalisasi agar semuanya dipetakan ke entitas

“Jakarta” yang sama. Dalam konteks peristiwa bencana, informasi seperti “23 Feb” dan “23/2” perlu disatukan sebagai entitas tanggal yang seragam agar sistem dapat mengenali waktu kejadian secara akurat.

Adapun dalam domain aplikasi yang sensitif seperti dokumen medis atau laporan hukum, proses cleaning juga menyangkut anonymization atau penghapusan informasi pribadi seperti nama pasien, nomor identitas, dan alamat. Hal ini penting untuk menjaga kepatuhan terhadap prinsip etika dan regulasi perlindungan data pribadi.

Secara teknis, proses normalisasi dan cleaning dapat dilakukan melalui kombinasi aturan (rule-based methods), kamus substitusi, dan model pembelajaran. Tools NLP seperti spaCy, NLTK, Sastrawi, hingga regex dasar sering digunakan dalam tahap ini. Dalam sistem berskala besar, pipeline cleaning dan normalisasi biasanya diintegrasikan ke dalam tahapan preprocessing otomatis yang modular dan dapat dikustomisasi sesuai kebutuhan domain.

Sebagai kesimpulan, normalisasi dan cleaning teks merupakan tahap penting yang menjembatani teks mentah dengan representasi linguistik yang rapi dan seragam. Kualitas preprocessing pada tahap ini sangat menentukan kualitas hasil analitik pada tahap ekstraksi informasi berikutnya. Sistem yang dibangun tanpa tahap normalisasi yang tepat berisiko menghasilkan entitas redundan, kesalahan pengelompokan, atau bahkan kehilangan informasi penting akibat ketidaksesuaian format.

Latihan

A. Pertanyaan Pemahaman Konsep

1. Apa yang dimaksud dengan *preprocessing teks* dan mengapa tahap ini penting dalam sistem ekstraksi informasi?
2. Jelaskan perbedaan antara *tokenisasi* dan *stemming*.
3. Apa tujuan dari proses *stopword removal*, dan berikan contoh kata-kata yang termasuk stopword dalam Bahasa Indonesia.
4. Mengapa normalisasi teks diperlukan pada data yang berasal dari media sosial atau percakapan daring?
5. Jelaskan perbedaan antara *stemming* dan *lemmatization* dalam konteks Bahasa Indonesia.
6. Sebutkan dua tantangan umum dalam preprocessing teks Bahasa Indonesia dan bagaimana cara mengatasinya.
7. Mengapa kualitas data teks mentah berpengaruh langsung terhadap hasil *Information Extraction*?

B. Latihan Praktik Sederhana

Diberikan kalimat berikut:

"Pemerintah Indonesia meluncurkan program vaksinasi massal untuk masyarakat di Jakarta pada bulan Januari 2021."

Tugas:

1. Lakukan tahap-tahap preprocessing: *cleaning*, tokenisasi, *stopword removal*, dan *stemming*.
2. Tulis hasil tiap tahap dalam bentuk tabel (sebelum dan sesudah preprocessing).
3. Jelaskan perubahan apa yang terjadi pada teks setelah melalui setiap proses.

4. Simpulkan mengapa hasil akhir preprocessing lebih mudah diolah oleh komputer dibandingkan teks mentah.

C. Studi Kasus / Proyek Mini

Anda diminta merancang *pipeline preprocessing* untuk sistem ekstraksi informasi berita bencana Indonesia.

1. Tentukan tahapan preprocessing yang paling relevan dan jelaskan alasannya.
2. Pilih dua library NLP yang dapat digunakan untuk Bahasa Indonesia (misalnya NLTK, Sastrawi, atau spaCy-Indonesia).
3. Buat alur diagram sederhana yang menggambarkan proses preprocessing dari teks mentah hingga hasil siap ekstraksi.
4. Diskusikan bagaimana kesalahan pada tahap preprocessing (misalnya tokenisasi yang salah) dapat memengaruhi hasil IE.

D. Tugas Diskusi / Refleksi

1. Menurut Anda, apakah preprocessing teks lebih bersifat teknis atau linguistik? Jelaskan pandangan Anda.
2. Bagaimana pendekatan preprocessing dapat disesuaikan untuk domain yang berbeda (misalnya teks medis vs. media sosial)?
3. Berikan contoh kasus di mana *stopword removal* justru menyebabkan kehilangan informasi penting.

BAB 3
FEATURE ASSIGNMENT: TABEL KONTEKSTUAL FITUR

Tujuan Pembelajaran

Setelah mempelajari Bab 3, pembaca diharapkan mampu:

1. Menjelaskan konsep dasar feature (fitur) dalam pengolahan teks dan perannya dalam ekstraksi informasi.
2. Mengidentifikasi jenis-jenis fitur linguistik dan statistik yang digunakan dalam sistem NLP, seperti n-gram, POS tag, *named entity*, dan dependensi sintaksis.
3. Memahami pentingnya konteks dalam pembentukan fitur, baik konteks lokal (kata sekitar) maupun konteks global (struktur kalimat dan dokumen).
4. Membedakan antara feature berbasis frekuensi, posisi, dan semantik.
5. Menerapkan metode feature assignment sederhana dengan mempertimbangkan konteks linguistik.
6. Menganalisis pengaruh pemilihan fitur terhadap performa sistem ekstraksi informasi.
7. Mengevaluasi pendekatan manual dan otomatis dalam penentuan fitur teks.

Pendahuluan

Dalam sistem ekstraksi informasi, terutama yang berbasis aturan atau pembelajaran mesin, fitur adalah representasi dari karakteristik linguistik yang digunakan untuk mengidentifikasi, mengelompokkan, atau mengekstraksi bagian-bagian penting dari teks. Fitur dapat berupa bentuk kata, kelas kata (part-of-speech), posisi dalam kalimat, pola sintaktik, hingga konteks kata sebelum dan

sesudah token tertentu. Pemilihan dan penugasan fitur yang tepat (feature assignment) merupakan langkah kritis dalam membangun sistem ekstraksi informasi yang akurat dan efisien.

Salah satu bentuk representasi fitur yang umum digunakan dalam ekstraksi informasi adalah tabel kontekstual fitur. Tabel ini memetakan setiap token atau frasa dalam teks ke dalam serangkaian atribut linguistik yang menggambarkan karakteristiknya dalam konteks lokal. Informasi dalam tabel tersebut kemudian digunakan untuk membangun aturan, melatih model, atau mengevaluasi hasil ekstraksi. Pemahaman terhadap penyusunan dan penggunaan tabel kontekstual fitur menjadi dasar yang penting bagi mahasiswa maupun praktisi NLP yang ingin membangun sistem EI secara sistematis.

A. Pengertian dan Fungsi Fitur Kontekstual

Fitur kontekstual dapat didefinisikan sebagai sekumpulan informasi linguistik dan statistik yang merepresentasikan posisi, peran, atau sifat suatu kata dalam kalimat, yang digunakan sebagai acuan dalam proses ekstraksi informasi. Berbeda dengan fitur leksikal yang hanya melihat bentuk kata secara langsung, fitur kontekstual mempertimbangkan posisi kata dalam hubungannya dengan kata-kata di sekitarnya. Pendekatan ini bertujuan menangkap makna dan fungsi kata dalam struktur kalimat yang lebih luas, sehingga meningkatkan akurasi dalam proses identifikasi entitas, relasi, atau aksi.

Sebagai ilustrasi, dalam kalimat "Gubernur Jawa Barat Ridwan Kamil meresmikan taman kota di Bandung," fitur kontekstual tidak hanya memetakan kata "Ridwan Kamil" sebagai entitas, tetapi juga menganalisis kata sebelumnya ("Gubernur", "Jawa Barat") dan sesudahnya ("meresmikan") sebagai petunjuk semantik tentang siapa

dia dan apa perannya dalam kalimat. Dengan demikian, sistem dapat menafsirkan bahwa "Ridwan Kamil" bukan sekadar nama orang, tetapi juga subjek dan pelaku dari sebuah aksi, dengan status sebagai pejabat pemerintah.

Fungsi utama dari fitur kontekstual adalah untuk menyajikan informasi yang dibutuhkan oleh sistem untuk mengenali pola-pola linguistik tertentu. Dalam pendekatan rule-based, fitur ini digunakan untuk menyusun aturan seperti: "Jika token X didahului oleh kata jabatan dan diikuti oleh kata kerja aktif, maka X kemungkinan besar adalah entitas orang." Dalam pendekatan berbasis pembelajaran mesin, fitur kontekstual dijadikan input dalam bentuk vektor ke dalam model klasifikasi, seperti dalam metode Conditional Random Fields (CRF) atau Recurrent Neural Networks (RNN), yang belajar mengenali urutan token dan hubungannya.

Fitur kontekstual yang umum digunakan antara lain:

- Kata sebelum dan sesudah (window context): membantu mengenali pola sekitar token.
- Part-of-speech (POS) dari token dan sekitarnya: menunjukkan peran gramatikal.
- Apakah token berupa huruf kapital: sering digunakan untuk mendeteksi nama entitas.
- Apakah token termasuk dalam daftar entitas yang dikenal (gazetteer): digunakan untuk pencocokan nama tempat atau organisasi.
- Pola ortografi: seperti token yang mengandung angka, tanda baca, atau huruf kapital semua.

Fitur-fitur ini kemudian disusun dalam format tabular di mana setiap baris mewakili sebuah token, dan setiap kolom merepresentasikan jenis fitur tertentu.

Format ini tidak hanya mempermudah analisis manual dan evaluasi model, tetapi juga memungkinkan proses visualisasi pola atau debugging sistem ekstraksi yang sedang dikembangkan.

Dalam konteks Bahasa Indonesia, penerapan fitur kontekstual menjadi tantangan tersendiri karena fleksibilitas sintaksis dan keunikan struktur kata. Misalnya, dalam kalimat pasif atau inversi SPOK, urutan kata dapat berubah tanpa mengubah makna. Oleh karena itu, fitur kontekstual harus didesain dengan memperhatikan fleksibilitas ini, agar tidak terlalu kaku terhadap posisi kata tetapi tetap akurat dalam menangkap fungsi semantiknya.

Secara keseluruhan, fitur kontekstual berperan sebagai fondasi informasi yang digunakan dalam semua keputusan sistem ekstraksi informasi. Kemampuan sistem dalam menangkap dan memanfaatkan informasi kontekstual akan sangat menentukan keberhasilan identifikasi entitas, relasi, dan informasi penting lainnya dalam teks.

B. Teknik Identifikasi Fitur dalam Kalimat

Identifikasi fitur dalam kalimat adalah proses sistematis untuk mengenali dan mengekstraksi informasi linguistik dari setiap elemen dalam teks, yang nantinya akan direpresentasikan dalam bentuk fitur. Tujuan dari proses ini adalah untuk menangkap karakteristik semantik dan sintaktik dari token yang sedang dianalisis, baik berdasarkan bentuknya, posisi dalam kalimat, maupun hubungannya dengan token lain. Teknik ini sangat penting dalam ekstraksi informasi karena fitur-fitur tersebut akan digunakan sebagai dasar dalam membuat aturan atau melatih model yang bertugas melakukan klasifikasi dan ekstraksi entitas serta relasi.

Secara umum, teknik identifikasi fitur dimulai dari tahap segmentasi teks menjadi kalimat, lalu dilanjutkan dengan tokenisasi, pelabelan kelas kata (Part-of-Speech tagging), analisis dependensi, serta penandaan entitas (jika diperlukan). Dari setiap token yang telah diperoleh, berbagai jenis fitur kemudian dikalkulasi dan dicatat dalam bentuk tabel. Fitur-fitur ini dapat bersifat leksikal (berbasis kata), morfologis, sintaktik, maupun semantik.

Teknik pertama yang umum digunakan adalah ekstraksi fitur leksikal, yaitu pengambilan informasi dari kata itu sendiri. Misalnya: bentuk asli kata, panjang kata, kapitalisasi (apakah dimulai dengan huruf kapital), keberadaan angka atau simbol, dan apakah kata tersebut termasuk dalam gazetteer atau daftar nama-nama yang dikenal seperti nama kota, organisasi, atau produk. Informasi ini sangat berguna untuk mengidentifikasi kemungkinan entitas, seperti nama orang atau lokasi.

Selanjutnya adalah analisis kelas kata (POS tagging). Teknik ini memberikan label pada setiap token berdasarkan fungsinya dalam kalimat, seperti kata benda (NN), kata kerja (VB), kata sifat (JJ), dan sebagainya. POS tag merupakan indikator penting yang membantu sistem memahami struktur sintaksis kalimat dan mengelompokkan entitas sesuai kategori yang relevan. Sebagai contoh, nama orang biasanya muncul sebagai proper noun, sedangkan tindakan diekspresikan melalui verb atau action phrase.

Kemudian, terdapat fitur berbasis konteks lokal, yaitu token-token di sekitar token target yang dianalisis dalam jendela tertentu (misalnya 2 kata sebelum dan 2 kata sesudah). Teknik ini dikenal sebagai context windowing, dan sangat berguna karena banyak entitas dan relasi hanya bisa dipahami melalui konteks. Sebagai contoh, token

“Bandung” dapat berarti kota, universitas, atau bahkan nama produk, tergantung pada kata-kata di sekitarnya seperti “berlokasi di”, “lulusan dari”, atau “produk kopi dari”.

Teknik lanjutan melibatkan analisis dependensi atau struktur kalimat, di mana relasi antar token dianalisis menggunakan model dependensi sintaktik. Teknik ini memungkinkan sistem untuk mengidentifikasi hubungan subjek-predikat-objek, klausa dependen, serta frasa nominal yang kompleks. Informasi ini sangat penting dalam kasus ekstraksi relasi dan kejadian. Sebagai contoh, dalam kalimat “Bupati Bogor meresmikan jembatan di Cileungsi”, dependensi dapat digunakan untuk mengetahui bahwa “Bupati Bogor” adalah pelaku aksi, “meresmikan” adalah kata kerja utama, dan “jembatan di Cileungsi” adalah objek dan lokasinya.

Selain itu, dalam banyak sistem, digunakan fitur berdasarkan bentuk dan pola seperti:

1. Apakah token adalah satu kata atau gabungan kata (multi-token phrase)
2. Bentuk huruf (semua kapital, huruf campuran, dsb.)
3. Apakah token berakhiran -an, -i, -kan (penanda derivasi morfologis dalam bahasa Indonesia)
4. Apakah token merupakan singkatan atau akronim

Dalam sistem yang lebih kompleks, teknik pengayaan fitur semantik juga diterapkan, seperti pemetaan token ke dalam kategori ontologi tertentu (misalnya: “Jakarta” → lokasi, “Covid-19” → penyakit, “Rp” → satuan moneter). Ini bisa dilakukan melalui pencocokan ke basis data eksternal seperti Wikidata, WordNet, atau menggunakan klasifikasi berbasis embedding.

Teknik identifikasi fitur tidak selalu dilakukan secara manual. Banyak alat bantu dan pustaka NLP modern yang sudah menyediakan fungsi ini secara otomatis. Contoh alat yang sering digunakan adalah spaCy, Stanza, IndoNLP, dan NLTK, yang mampu mengekstrak ratusan jenis fitur dari teks dengan sedikit konfigurasi tambahan. Namun demikian, dalam proyek berbasis domain tertentu (misalnya berita bencana, laporan keuangan), sering kali perlu dilakukan feature engineering khusus untuk menghasilkan fitur yang benar-benar relevan.

Secara umum, keberhasilan sistem ekstraksi informasi sangat bergantung pada seberapa tepat dan lengkap fitur-fitur yang berhasil diidentifikasi dari teks. Oleh karena itu, teknik identifikasi fitur dalam kalimat bukan hanya tahap teknis, tetapi merupakan proses analitis yang membutuhkan pemahaman linguistik, konteks domain, serta kecermatan dalam membedakan informasi penting dari yang tidak relevan.

C. Penyusunan Tabel Kontekstual Fitur

Penyusunan tabel kontekstual fitur adalah proses penting dalam sistem ekstraksi informasi yang berfungsi mengorganisasi hasil identifikasi fitur dari setiap token dalam format yang terstruktur dan mudah dianalisis. Tabel ini merupakan representasi sistematis dari token dan berbagai atribut linguistik serta kontekstual yang menyertainya. Dengan format ini, setiap kata dalam teks tidak hanya berdiri sebagai entitas leksikal, tetapi juga sebagai baris data yang mengandung sejumlah informasi yang menjelaskan perannya dalam kalimat.

Tabel kontekstual fitur biasanya disusun dalam bentuk baris dan kolom, di mana setiap baris mewakili

satu token (biasanya satu kata atau frasa pendek), sedangkan kolom-kolom berisi atribut atau fitur dari token tersebut. Kolom-kolom ini dapat mencakup berbagai informasi, mulai dari kata itu sendiri (form), bentuk dasar (lemma), kelas kata (POS tag), kapitalisasi, posisi token dalam kalimat, konteks kiri dan kanan, hingga label entitas jika tersedia.

Contoh sederhana dari struktur tabel kontekstual fitur untuk kalimat:

"Presiden Joko Widodo meresmikan jembatan di Papua."

adalah sebagai berikut:

Token	*Lemma*	*POS*	*Capital*	*Left-1*	*Right+1*	*Is_Gazetteer*	*Label*
Presiden	presiden	NN	TRUE	<START>	Joko	FALSE	Title
Joko	joko	NNP	TRUE	Presiden	Widodo	FALSE	Person_Begin
Widodo	widodo	NNP	TRUE	Joko	meresmikan	FALSE	Person_Inside
meresmikan	resmikan	VB	FALSE	Widodo	jembatan	FALSE	Action
jembatan	jembatan	NN	FALSE	meresmikan	di	FALSE	Object
di	di	IN	FALSE	jembatan	Papua	FALSE	Preposition
Papua	papua	NNP	TRUE	di	.	TRUE	Location
.	.	PUNCT	FALSE	Papua	<END>	FALSE	None

Dari tabel di atas, kita bisa melihat bagaimana setiap token diperkaya dengan informasi tambahan yang memungkinkan sistem mengenali pola linguistik yang relevan. Kolom "Left-1" dan "Right+1" misalnya, menunjukkan konteks langsung dari token tersebut. Kolom "POS" menunjukkan kelas kata yang menjadi indikator peran gramatikal, sementara kolom "Is_Gazetteer" menunjukkan apakah token terdapat dalam daftar entitas yang dikenal, seperti nama tempat. Label digunakan untuk anotasi akhir baik manual maupun hasil pelabelan otomatis yang menandakan kategori semantik atau struktur informasi yang diharapkan dari token tersebut.

Proses penyusunan tabel seperti ini bisa dilakukan secara manual untuk tujuan pembelajaran atau eksperimen awal. Namun dalam aplikasi nyata dan skala besar, proses ini biasanya diotomatisasi menggunakan skrip atau pustaka NLP yang mampu melakukan ekstraksi fitur secara batch. Alat seperti spaCy, scikit-learn, dan Pandas dalam Python sangat membantu dalam menyusun tabel fitur dari dokumen teks secara efisien.

Keuntungan utama dari penyusunan tabel kontekstual fitur adalah kemudahan dalam:

1. Visualisasi data linguistik: memudahkan manusia untuk menganalisis dan memahami struktur kalimat.
2. Membangun aturan ekstraksi: mempermudah penulisan rule dengan melihat pola fitur antar token.
3. Melatih model pembelajaran mesin: menyediakan format input yang sesuai untuk supervised learning (fitur sebagai input, label sebagai target).
4. Evaluasi dan debugging sistem: ketika sistem tidak bekerja dengan baik, tabel ini memungkinkan pelacakan sumber kesalahan.

Dalam pengembangan sistem ekstraksi informasi berbasis aturan, tabel kontekstual juga dapat digunakan sebagai dasar untuk menyusun pattern matcher, di mana kombinasi fitur tertentu menjadi pemicu aktivasi aturan. Misalnya, aturan "jika token memiliki POS noun, diawali oleh token dengan POS title, dan token tersebut berupa kapital, maka kemungkinan besar token adalah nama orang." Kombinasi kondisi semacam ini sangat mudah diterapkan bila representasi datanya berbentuk tabel.

Penting untuk dicatat bahwa semakin kaya fitur yang ditambahkan ke tabel, maka semakin besar pula kompleksitas dan kebutuhan komputasi sistem. Oleh

karena itu, proses seleksi fitur menjadi tahap lanjut yang tak kalah penting, di mana pengembang sistem harus memilih fitur-fitur yang paling relevan dan berdampak terhadap kualitas ekstraksi.

Sebagai kesimpulan, penyusunan tabel kontekstual fitur bukan sekadar format penyajian data, tetapi merupakan jembatan antara bahasa manusia dan sistem komputasi. Ia menjadi titik temu antara analisis linguistik dan pendekatan komputasional, serta fondasi dari berbagai metode ekstraksi informasi yang bersifat rule-based maupun machine learning. Pemahaman mendalam terhadap struktur dan dinamika tabel ini akan sangat membantu dalam merancang sistem EI yang akurat, transparan, dan mudah dipelihara.

D. Penerapan Fitur Kontekstual dalam Ekstraksi

Setelah fitur kontekstual berhasil diidentifikasi dan disusun dalam bentuk tabel, langkah berikutnya adalah bagaimana fitur-fitur tersebut digunakan secara efektif dalam proses ekstraksi informasi. Penerapan fitur kontekstual dalam ekstraksi bertujuan untuk mendeteksi dan mengklasifikasikan bagian-bagian penting dalam teks seperti entitas, relasi, atau peristiwa, dengan mengandalkan kombinasi nilai-nilai fitur yang merepresentasikan pola-pola linguistik yang khas.

Dalam sistem berbasis aturan (rule-based extraction), fitur kontekstual menjadi acuan utama dalam menyusun logika ekstraksi. Setiap fitur, baik itu kelas kata, posisi dalam kalimat, kapitalisasi, atau konteks token di sekitarnya, dapat dikombinasikan menjadi kondisi-kondisi logis yang membentuk sebuah aturan. Sebagai contoh, sebuah aturan sederhana untuk mengekstrak nama pejabat pemerintahan dari teks berita dapat

berbunyi: “Jika sebuah token adalah noun proper (NNP), didahului oleh token bertipe title seperti ‘Presiden’, ‘Gubernur’, ‘Menteri’, dan token tersebut ditulis dengan huruf kapital, maka token tersebut adalah kandidat entitas Person.” Aturan ini, meskipun tampak sederhana, terbukti efektif dalam domain berita formal yang cenderung konsisten dalam struktur penyajiannya.

Dalam pendekatan berbasis pembelajaran mesin (supervised learning), fitur kontekstual menjadi input utama yang digunakan untuk melatih model klasifikasi. Dalam hal ini, setiap baris pada tabel fitur dianggap sebagai vektor data, sedangkan label akhir misalnya nama entitas atau kategori semantik dianggap sebagai kelas yang harus diprediksi. Model seperti Decision Tree, Naïve Bayes, Conditional Random Fields (CRF), hingga Neural Network menggunakan vektor fitur ini untuk belajar membedakan antara token-token yang merupakan bagian dari entitas dengan yang bukan. Semakin lengkap dan relevan fitur yang digunakan, semakin tinggi pula potensi akurasi dari model ekstraksi yang dibangun.

Fitur kontekstual juga sangat penting dalam ekstraksi relasi. Dalam tugas ini, sistem tidak hanya perlu mengenali entitas, tetapi juga memahami hubungan antara dua atau lebih entitas dalam satu kalimat atau paragraf. Misalnya, untuk mengekstrak hubungan antara seorang tokoh dan institusinya (seperti “Doni Monardo menjabat sebagai Kepala BNPB”), sistem perlu mengenali bahwa “Doni Monardo” adalah entitas orang, “BNPB” adalah entitas organisasi, dan hubungan semantik antara keduanya ditunjukkan oleh frasa “menjabat sebagai Kepala.” Dalam kasus seperti ini, fitur kontekstual yang digunakan mencakup posisi kedua entitas dalam kalimat, jenis kata kerja yang menghubungkan, serta preposisi atau

frasa fungsional yang menjadi penghubung relasi.

Pada sistem berbasis deep learning, fitur kontekstual disediakan dalam bentuk word embedding atau representasi vektor numerik dari kata-kata dan lingkungannya. Meskipun tidak eksplisit dalam bentuk tabel seperti pada pendekatan klasik, embedding ini secara implisit menyimpan informasi konteks dan korelasi antar token. Model seperti BERT atau ELMo bahkan mampu menangkap konteks dua arah dalam kalimat (sebelum dan sesudah token), yang membuatnya sangat efektif untuk ekstraksi entitas atau peristiwa kompleks. Namun, tetap saja, proses preprocessing berbasis fitur kontekstual tradisional tidak sepenuhnya tergantikan, khususnya dalam sistem hybrid yang menggabungkan pendekatan statistik dan aturan linguistik.

Penerapan fitur kontekstual juga membantu dalam post-processing, yaitu tahapan setelah ekstraksi awal untuk menyaring hasil yang tidak valid, menyatukan entitas multi-token, atau menggabungkan entitas yang terduplikasi. Misalnya, sistem dapat memanfaatkan informasi bahwa dua token "Universitas Indonesia" dan "UI" merujuk pada entitas yang sama, karena keduanya memiliki fitur semantik dan kontekstual yang identik. Proses ini penting untuk menjaga konsistensi dan koherensi hasil ekstraksi, khususnya dalam aplikasi seperti penyusunan basis data, pencarian informasi, atau visualisasi jaringan entitas.

Dalam penerapan di dunia nyata, fitur kontekstual sering kali dikustomisasi sesuai kebutuhan domain. Di bidang kesehatan, fitur yang digunakan mungkin mencakup klasifikasi istilah medis dan satuan dosis. Di bidang hukum, sistem mungkin menambahkan fitur berbasis struktur dokumen, seperti posisi token dalam

pasal atau ayat. Penyesuaian ini menunjukkan bahwa fleksibilitas dan desain fitur kontekstual yang cermat merupakan kunci utama dalam membangun sistem ekstraksi informasi yang sukses.

Dengan demikian, fitur kontekstual bukan hanya komponen pendukung, tetapi justru merupakan inti dari proses pengambilan keputusan dalam sistem ekstraksi informasi. Baik digunakan dalam metode berbasis aturan maupun pembelajaran mesin, fitur-fitur ini memberikan jembatan antara kompleksitas bahasa alami dan logika formal yang digunakan oleh sistem cerdas.

E. Penggunaan Fitur Kontekstual

Untuk memperjelas penerapan konsep fitur kontekstual dalam ekstraksi informasi, berikut disajikan studi kasus sederhana yang diambil dari domain berita bencana alam, sebuah bidang yang memerlukan respons cepat dan akurasi tinggi dalam pengambilan data berbasis teks.

Bayangkan sistem ekstraksi informasi dikembangkan untuk mendeteksi entitas kunci dalam laporan bencana, seperti lokasi kejadian, jenis bencana, waktu kejadian, dan jumlah korban. Sistem ini dirancang untuk membantu lembaga tanggap darurat atau pusat data bencana dalam mengelola informasi yang berasal dari berbagai artikel berita daring.

Diberikan sebuah paragraf berita berikut:

"Gempa berkekuatan 6,2 SR mengguncang Kabupaten Cianjur pada Senin pagi. Akibat kejadian ini, sebanyak 162 orang dinyatakan meninggal dunia dan ratusan lainnya luka-luka."

Langkah pertama adalah memecah kalimat menjadi token dan menyusun tabel kontekstual fitur untuk setiap

token. Beberapa fitur yang digunakan antara lain: token, lemma, kelas kata (POS), kapitalisasi, token sebelumnya dan sesudahnya (context window), serta kategori prediksi entitas (jika telah tersedia). Tabel sederhananya sebagai berikut:

Token	*POS*	*Lemma*	*Capital*	*Prev Token*	*Next Token*	*Context Phrase*	*Gazetteer*	*Pred_Entity*
Gempa	NN	gempa	TRUE	<START>	berkekuatan	"Gempa berkekuatan"	TRUE	Disaster_Type
berkekuatan	VB	kekuatan	FALSE	Gempa	6,2	"berkekuatan 6,2 SR"	FALSE	-
6,2	NUM	6,2	FALSE	berkekuatan	SR	"6,2 SR"	FALSE	Magnitude
SR	SYM	SR	TRUE	6,2	mengguncang	"6,2 SR mengguncang"	TRUE	Unit
mengguncang	VB	guncang	FALSE	SR	Kabupaten	"mengguncang Kabupaten"	FALSE	Action
Kabupaten	NNP	kabupaten	TRUE	mengguncang	Cianjur	"Kabupaten Cianjur"	TRUE	Region_Prefix
Cianjur	NNP	cianjur	TRUE	Kabupaten	pada	"Cianjur pada Senin"	TRUE	Location
Senin	NNP	senin	TRUE	pada	pagi	"Senin pagi"	TRUE	Date
162	NUM	162	FALSE	sebanyak	orang	"sebanyak 162 orang"	FALSE	Victim_Count
meninggal	VB	meninggal	FALSE	dinyatakan	dunia	"dinyatakan meninggal"	FALSE	Status

Melalui tabel ini, kita dapat melihat bagaimana sistem menggunakan kombinasi fitur seperti POS, kapitalisasi, konteks frasa, dan keberadaan dalam gazetteer untuk mengklasifikasikan token ke dalam entitas seperti jenis bencana (gempa), lokasi (Cianjur), waktu (Senin), dan jumlah korban (162 orang). Dengan menyusun token secara kontekstual, sistem dapat lebih mudah menangkap makna dan struktur naratif dalam kalimat-kalimat alami.

Penerapan semacam ini dapat dikembangkan lebih jauh dengan membuat aturan berbasis pola. Misalnya, pola "Gempa + berkekuatan + [angka] + SR" dapat dikenali sebagai konstruksi untuk entitas bencana gempa dan kekuatannya. Atau pola "sebanyak + [angka] + orang" dapat digunakan untuk mendeteksi jumlah korban. Dalam sistem berbasis pembelajaran mesin, pola-pola tersebut tidak ditulis secara eksplisit, tetapi dipelajari dari banyak contoh kalimat melalui fitur-fitur seperti yang tertera pada tabel.

Dalam kasus lain, seperti dokumen resmi pemerintahan atau laporan medis, fitur kontekstual dapat disesuaikan untuk mengekstrak entitas yang berbeda, seperti nama lembaga, peraturan, kode penyakit, atau dosis obat. Misalnya, dalam laporan medis: "Pasien diberi paracetamol 500 mg setiap 8 jam," sistem dapat mengenali "paracetamol" sebagai obat, "500 mg" sebagai dosis, dan "8 jam" sebagai frekuensi, dengan memanfaatkan fitur posisi, satuan ukuran, serta gazetteer farmasi.

Studi kasus ini menunjukkan bahwa kekuatan fitur kontekstual tidak hanya terletak pada kemampuannya mendeskripsikan token secara individual, tetapi juga dalam kemampuannya menangkap keterkaitan antar bagian kalimat secara menyeluruh. Dengan memahami konteks lokal maupun global dalam teks, sistem ekstraksi informasi dapat bekerja lebih cerdas dan akurat dalam menyaring informasi penting dari data tidak terstruktur.

Sebagai penutup, fitur kontekstual berperan sebagai fondasi bagi semua metode ekstraksi informasi yang bersifat presisi. Baik melalui aturan linguistik maupun algoritma pembelajaran statistik, representasi fitur yang baik adalah kunci keberhasilan dalam memahami dan memanfaatkan bahasa alami secara komputasional.

Latihan

A. Pertanyaan Pemahaman Konsep

1. Apa yang dimaksud dengan *feature assignment* dalam konteks ekstraksi informasi?
2. Sebutkan perbedaan antara fitur berbasis leksikal dan fitur berbasis sintaksis.
3. Mengapa konteks kata penting dalam menentukan fitur yang relevan untuk ekstraksi informasi?
4. Jelaskan contoh penggunaan n-gram dalam pembentukan fitur teks.
5. Apa hubungan antara *part-of-speech tagging* dengan pembentukan fitur?
6. Bagaimana pemilihan fitur yang tidak tepat dapat menurunkan akurasi sistem ekstraksi informasi?
7. Berikan contoh fitur semantik yang dapat digunakan dalam tugas pengenalan entitas.

B. Latihan Praktik Sederhana

Diberikan kalimat berikut:

"Presiden Joko Widodo meresmikan Jembatan Youtefa di Jayapura pada hari Senin."

Tugas:

1. Tentukan *part-of-speech* (POS) untuk setiap kata.
2. Identifikasi kata-kata yang menjadi kandidat entitas (person, location, organization).
3. Bentuklah vektor fitur sederhana dengan elemen: token, POS, dan posisi kata.
4. Jelaskan bagaimana konteks kata "Jayapura" membantu sistem mengenalinya sebagai *location entity*.

C. Studi Kasus / Proyek Mini

Anda sedang merancang sistem *Named Entity Recognition* (NER) sederhana untuk Bahasa Indonesia.

1. Tentukan jenis fitur apa saja yang akan digunakan (misalnya kata, POS, awalan/akhiran, kapitalisasi, dan konteks tetangga).
2. Buat tabel sampel dataset dengan 5 baris kalimat dan kolom fitur yang Anda tentukan.
3. Jelaskan bagaimana setiap fitur berkontribusi pada identifikasi entitas.
4. Diskusikan perbedaan hasil jika fitur konteks (kata sebelumnya dan sesudahnya) dihilangkan.

D. Diskusi / Refleksi

1. Apakah menurut Anda proses feature assignment lebih bersifat seni (*art*) atau sains (*science*)? Jelaskan.
2. Dalam sistem modern berbasis *deep learning*, sebagian proses feature assignment dilakukan otomatis. Apa keuntungan dan kelemahannya dibanding metode manual?
3. Menurut Anda, bagaimana pendekatan berbasis konteks dapat meningkatkan kemampuan sistem untuk memahami makna teks secara lebih mendalam?

BAB 4
FEATURE ASSIGNMENT: TABEL FITUR MORFOLOGI

Tujuan Pembelajaran

Setelah mempelajari Bab 4, pembaca diharapkan mampu:

1. Menjelaskan konsep dasar Part-of-Speech (POS) Tagging serta fungsinya dalam proses ekstraksi informasi.
2. Mengenali kategori kelas kata utama dalam Bahasa Indonesia, seperti nomina, verba, adjektiva, adverbia, preposisi, dan konjungsi.
3. Menjelaskan perbedaan antara POS Tagging berbasis aturan (*rule-based*) dan berbasis pembelajaran mesin (*statistical tagging*).
4. Menjelaskan konsep Parsing Sintaksis dan perannya dalam memahami struktur kalimat.
5. Membedakan antara parsing berbasis konstituen (constituent parsing) dan parsing berbasis dependensi (dependency parsing).
6. Menerapkan contoh POS Tagging dan Parsing sederhana pada kalimat berbahasa Indonesia.
7. Menganalisis bagaimana hasil POS Tagging dan Parsing dapat meningkatkan akurasi dalam sistem ekstraksi informasi.

Pendahuluan

Selain fitur kontekstual yang berfokus pada posisi dan lingkungan kata dalam kalimat, sistem ekstraksi informasi juga sangat diuntungkan oleh fitur-fitur yang bersumber dari bentuk dan struktur internal kata itu sendiri, yang dikenal sebagai fitur morfologi. Fitur ini membantu sistem mengenali

pola pembentukan kata, jenis kata turunan, dan peran semantis berdasarkan struktur morfologisnya. Bahasa Indonesia, sebagai bahasa yang kaya dengan afiksasi dan bentuk derivatif, sangat bergantung pada fitur morfologi untuk meningkatkan akurasi pemahaman mesin terhadap teks.

A. Definisi dan Pentingnya Fitur Morfologi

Fitur morfologi adalah karakteristik linguistik yang diperoleh dari analisis morfem, yakni unit terkecil pembentuk makna dalam sebuah kata. Dalam Bahasa Indonesia, morfem dapat berupa akar kata (kata dasar) dan berbagai bentuk imbuhan, seperti awalan (prefiks), akhiran (sufiks), sisipan (infiks), dan gabungan awalan-akhiran (konfiks). Selain itu, terdapat juga proses reduplikasi dan pemajemukan kata, yang berkontribusi terhadap pembentukan makna baru atau gradasi semantik.

Dalam konteks ekstraksi informasi, fitur morfologi digunakan untuk memahami bagaimana bentuk kata memengaruhi fungsi atau kategori semantiknya. Sebagai contoh, kata "mendukung" memiliki prefiks "meN-" dan akar "dukung", yang menandakan bahwa kata tersebut adalah kata kerja aktif. Informasi ini membantu sistem mengenali "mendukung" sebagai tindakan dalam sebuah relasi. Demikian pula, kata "dukungan" berasal dari akar "dukung" dengan akhiran "-an" yang mengubahnya menjadi kata benda. Dengan mengenali pola ini, sistem dapat membedakan antara pelaku tindakan dan hasil dari tindakan tersebut.

Pentingnya fitur morfologi menjadi sangat nyata dalam bahasa yang sangat produktif seperti Bahasa Indonesia, di mana kata dasar dapat berkembang menjadi puluhan bentuk turunan dengan makna yang berbeda.

Sistem ekstraksi informasi yang hanya melihat permukaan kata (surface form) tanpa mempertimbangkan struktur morfologisnya akan berisiko salah mengelompokkan kata yang secara semantik serupa, atau bahkan melewatkan informasi penting.

Sebagai ilustrasi, dalam ekstraksi tindakan dalam kalimat "Pemerintah memberikan bantuan kepada korban banjir," sistem perlu mengenali bahwa "memberikan" adalah kata kerja aktif dari akar "beri", yang menandakan aksi transfer atau donasi. Jika sistem hanya mengenali bentuk "beri" sebagai kata kunci, maka ia mungkin gagal mengekstrak tindakan dalam bentuk kompleks seperti "memberikan", "diberikan", atau "pemberian". Oleh karena itu, fitur morfologi menjadi kunci untuk generalisasi dan konsistensi dalam identifikasi entitas dan tindakan.

Selain membantu dalam pelabelan token, fitur morfologi juga penting dalam disambiguasi. Banyak kata dalam Bahasa Indonesia yang memiliki bentuk serupa tetapi fungsi yang berbeda tergantung pada imbuhan atau struktur morfologinya. Contohnya, kata "pengawasan" dan "pengawas" memiliki akar yang sama yaitu "awas", namun yang pertama bermakna proses atau aktivitas (verbal noun), sedangkan yang kedua mengacu pada pelaku. Dalam sistem yang mengandalkan fitur morfologi, perbedaan ini bisa dikenali dan dimanfaatkan dalam klasifikasi token.

Dalam aplikasi berbasis pembelajaran mesin, fitur morfologi dapat diubah menjadi representasi numerik atau vektor untuk digunakan sebagai input model. Misalnya, sistem dapat menggunakan satu-hot encoding untuk jenis afiks yang muncul, panjang kata, atau jumlah morfem. Dalam model deep learning, informasi morfologi

sering digabungkan dengan word embedding untuk memperkaya konteks semantik.

Secara keseluruhan, fitur morfologi memberikan dimensi tambahan dalam representasi token, di luar konteks leksikal dan posisi dalam kalimat. Dalam sistem ekstraksi informasi yang kompleks, fitur ini membantu menjembatani ambiguitas bentuk kata, memperluas cakupan pencocokan entitas, dan meningkatkan presisi dalam identifikasi struktur kalimat. Oleh karena itu, pemahaman dan pemanfaatan fitur morfologi merupakan bagian esensial dalam desain pipeline ekstraksi yang komprehensif.

B. Bentuk dan Struktur Morfologi Kata

Struktur morfologi kata dalam Bahasa Indonesia mencerminkan sistem afiksasi dan pembentukan kata turunan yang produktif dan fleksibel. Pemahaman terhadap struktur ini menjadi sangat penting dalam pengolahan bahasa alami karena memungkinkan sistem untuk mengurai dan memahami bentuk kata secara sistematis, bukan hanya berdasarkan penampilan permukaan, tetapi juga berdasarkan komponen pembentuk maknanya.

Secara umum, bentuk morfologi kata dalam Bahasa Indonesia terdiri dari tiga kategori utama: kata dasar, kata berimbuhan, dan kata majemuk/reduplikasi.

Kata dasar adalah bentuk paling sederhana dari sebuah kata yang tidak mengalami perubahan atau penambahan. Kata-kata seperti “baca”, “tulis”, “besar”, atau “mobil” adalah contoh kata dasar. Dalam banyak kasus, kata dasar dapat berdiri sendiri sebagai kata bermakna lengkap, dan juga menjadi fondasi untuk pembentukan kata turunan.

Kata berimbuhan adalah kata yang mengalami penambahan afiks (imbuhan), baik berupa awalan (prefiks), akhiran (sufiks), sisipan (infiks), maupun gabungan awalan-akhiran (konfiks). Proses afiksasi ini sangat produktif dan dapat mengubah kelas kata serta maknanya. Sebagai contoh:

- me- + tulis → menulis (kata kerja)
- di- + tulis → ditulis (kata kerja pasif)
- pen- + tulis → penulis (pelaku)
- tulis + -an → tulisan (hasil)
- me- + tulis + -kan → menuliskan (kata kerja transaktif)

Analisis terhadap jenis dan kombinasi afiks ini menjadi komponen penting dalam penugasan fitur morfologi, karena secara langsung memengaruhi struktur sintaksis dan semantis kalimat. Misalnya, kata dengan prefiks "me-" umumnya berfungsi sebagai kata kerja aktif, sementara "di-" menunjukkan bentuk pasif, dan "per-an" sering kali menunjukkan konsep abstrak atau keadaan.

Kata majemuk dan reduplikasi juga merupakan bentuk penting dalam struktur morfologi Bahasa Indonesia. Kata majemuk terbentuk dari dua kata dasar atau lebih yang digabungkan menjadi satu kesatuan makna, misalnya: "rumah sakit", "mata pelajaran", atau "kartu tanda penduduk." Sistem ekstraksi informasi harus mampu mengenali bahwa satuan tersebut tidak dapat dipisahkan begitu saja ke dalam token individual, karena maknanya akan berubah atau hilang jika diproses secara terpisah.

Sedangkan reduplikasi (pengulangan kata) digunakan untuk berbagai tujuan semantik, seperti menunjukkan jamak ("buku-buku"), intensitas ("berlari-

lari"), atau bentuk khas lainnya seperti "sayur-mayur" dan "anak-anak." Analisis reduplikasi penting untuk mendeteksi variasi bentuk entitas yang muncul dalam teks. Sistem yang tidak mengenali "anak-anak" sebagai bentuk jamak dari "anak" mungkin akan menganggapnya sebagai entitas berbeda, sehingga mengurangi konsistensi hasil ekstraksi.

Dalam sistem ekstraksi informasi, semua komponen struktur morfologi ini biasanya direpresentasikan dalam bentuk tabel fitur. Misalnya, satu token seperti "diberikan" akan dianalisis menjadi:

- Akar kata: "beri"
- Prefiks: "di-"
- Sufiks: "-kan"
- Kelas kata: Kata kerja pasif
- Pola morfologis: konfiks di- -kan

Data ini kemudian dikodekan dalam bentuk fitur binari, simbolik, atau vektor numerik yang digunakan baik dalam aturan maupun model pembelajaran. Dengan mengidentifikasi struktur morfologis ini, sistem dapat menyimpulkan fungsi sintaktik token dan memperkirakan kemungkinan kategorinya dalam struktur informasi.

Tantangan dalam menangani morfologi Bahasa Indonesia terletak pada fleksibilitas dan ketidakterbatasan bentuk baru. Kata-kata turunan bisa terus muncul dari akar yang sama dengan makna yang sedikit berbeda, seperti "pengiriman", "dikirimkan", "mengirim", dan "pengirim." Sistem ekstraksi informasi perlu menangani semua bentuk ini secara konsisten dan terstandar agar tidak terjadi duplikasi atau ketidaktepatan pengelompokan entitas.

Sebagai penutup bagian ini, dapat ditegaskan bahwa

analisis bentuk dan struktur morfologi kata merupakan komponen penting dalam preprocessing lanjutan untuk ekstraksi informasi. Sistem yang memahami struktur ini tidak hanya dapat mengenali lebih banyak variasi kata dan frasa, tetapi juga lebih akurat dalam menyimpulkan peran semantis dan sintaktik dalam teks, terutama untuk Bahasa Indonesia yang sangat kaya dalam pembentukan kata turunan.

C. Teknik Ekstraksi Fitur Morfologi dari Teks

Ekstraksi fitur morfologi dari teks merupakan proses yang bertujuan mengurai kata menjadi komponen morfemiknya dan merepresentasikan komponen tersebut sebagai fitur yang dapat dianalisis atau digunakan oleh sistem ekstraksi informasi. Proses ini sangat penting dalam memastikan bahwa sistem dapat memahami variasi bentuk kata yang beragam dan menghubungkannya kembali ke akar atau pola dasar yang konsisten.

Secara umum, teknik ekstraksi fitur morfologi dapat dilakukan melalui tiga pendekatan utama: berbasis aturan (rule-based), berbasis kamus (dictionary-based), dan berbasis pembelajaran mesin (machine learning-based).

1. Pendekatan Berbasis Aturan (Rule-Based)

   Pendekatan ini menggunakan seperangkat aturan linguistik yang ditulis secara eksplisit untuk memisahkan awalan, akhiran, dan elemen morfologis lain dari sebuah kata. Misalnya, untuk kata "mengajarkan", aturan akan memetakan:

   - Prefiks: "meng-"
   - Akar: "ajar"
   - Sufiks: "-kan"

Proses ini memanfaatkan daftar afiks yang umum digunakan dalam Bahasa Indonesia, seperti prefiks me-, di-, ter-, ke-, dan sufiks -an, -kan, -i. Kombinasi aturan dapat ditulis dalam bentuk regular expression atau menggunakan struktur parser sederhana. Beberapa pustaka seperti Sastrawi mengimplementasikan pendekatan ini dengan cukup efektif.

Kelebihan dari metode ini adalah kejelasan dan keterbukaan logika yang digunakan, sehingga mudah untuk dikustomisasi sesuai kebutuhan domain. Namun, kelemahannya adalah kesulitan dalam menangani kasus ambiguitas dan bentuk kata yang tidak sesuai pola, terutama dalam teks informal atau media sosial.

2. Pendekatan Berbasis Kamus (Dictionary-Based)

Pendekatan ini memanfaatkan daftar kosakata yang telah dianotasi, di mana setiap entri mencakup informasi lengkap mengenai struktur morfologinya. Sistem akan mencocokkan token dalam teks dengan entri dalam kamus untuk menemukan bentuk dasar dan informasi afiks yang menyertainya. Kamus seperti KBBI digital, MorphInd, atau daftar lemmatization publik bisa dijadikan rujukan.

Keunggulan dari metode ini adalah tingkat akurasi yang tinggi pada teks formal, asalkan kata tersebut memang tercantum dalam kamus. Namun, ia sangat terbatas dalam menghadapi kata baru, istilah teknis, atau bentuk kreatif bahasa yang tidak tercakup dalam daftar.

3. Pendekatan Berbasis Pembelajaran Mesin

Dalam pendekatan ini, sistem dilatih menggunakan dataset beranotasi morfologi untuk

secara otomatis mempelajari pola penggabungan morfem. Model seperti Conditional Random Fields (CRF) atau Recurrent Neural Networks (RNN) dapat digunakan untuk melakukan segmentasi morfem dan klasifikasi jenis afiks secara bersamaan.

Contoh pendekatan ini adalah sistem yang dapat mempelajari bahwa kata "pemberitahuan" terdiri dari prefiks "pem-", akar "beritahu", dan sufiks "-an", berdasarkan data pelatihan. Pendekatan ini sangat fleksibel, dapat beradaptasi terhadap variasi teks, dan mampu menangani data tidak baku. Namun, ia membutuhkan data latih yang besar dan anotasi yang berkualitas.

Dalam praktiknya, ketiga pendekatan tersebut sering dikombinasikan untuk mendapatkan hasil optimal. Misalnya, sistem dapat menggunakan kamus terlebih dahulu, lalu fallback ke metode rule-based jika kata tidak ditemukan, dan akhirnya menggunakan model pembelajaran jika keduanya gagal mengenali bentuk kata.

Proses ekstraksi fitur morfologi biasanya menghasilkan output dalam bentuk vektor fitur yang menyertakan:

- Akar kata (lemma)
- Tipe afiks (prefiks, sufiks, konfiks)
- Pola morfologis (misalnya: meN-V-kan)
- Jumlah suku kata atau panjang kata (sebagai indikator kompleksitas)
- Kategori morfologis (kata dasar, kata turunan, kata ulang)

Semua output ini dapat diintegrasikan ke dalam tabel fitur seperti yang dijelaskan pada bab sebelumnya.

Hasilnya menjadi masukan penting dalam proses ekstraksi entitas, klasifikasi relasi, dan analisis semantik tingkat lanjut.

Dalam konteks ekstraksi informasi Bahasa Indonesia, fitur morfologi membantu sistem memahami perbedaan antara kata-kata seperti "menyumbang" (aksi), "sumbangan" (hasil), dan "penyumbang" (pelaku). Tanpa analisis morfologis, sistem dapat salah mengelompokkan ketiganya sebagai entitas yang sama atau bahkan gagal mengenalinya sebagai bagian dari struktur peristiwa yang kompleks.

Dengan demikian, ekstraksi fitur morfologi dari teks bukan hanya soal memecah kata, tetapi juga menyusun peta semantik dan sintaktik dari teks yang dianalisis. Proses ini memungkinkan sistem memahami teks dalam level yang lebih dalam dan kontekstual, yang sangat penting untuk keberhasilan ekstraksi informasi secara menyeluruh.

D. Penyusunan Tabel Fitur Morfologi

Setelah fitur morfologi berhasil diekstrak dari teks, langkah penting berikutnya adalah menyusun informasi tersebut ke dalam bentuk tabel yang terstruktur. Tabel fitur morfologi berfungsi sebagai representasi formal dari analisis morfemik kata-kata dalam teks, yang dapat digunakan sebagai input untuk berbagai tugas NLP seperti ekstraksi entitas, pengenalan peristiwa, atau pelabelan semantik.

Struktur umum dari tabel fitur morfologi mirip dengan tabel fitur kontekstual, namun fokusnya terletak pada dimensi morfologis kata. Setiap baris mewakili satu token, sedangkan kolom-kolom memuat atribut-atribut morfologi yang telah dianalisis. Atribut tersebut bisa

mencakup bentuk dasar kata (lemma), jenis afiks yang digunakan (prefiks, sufiks, konfiks), kategori morfologis (kata dasar, kata turunan, kata ulang), serta pola morfemiknya.

Berikut adalah contoh representasi tabel fitur morfologi berdasarkan kalimat:

"Pemerintah mendistribusikan bantuan kepada korban banjir."

Token	*Lemma*	*Prefiks*	*Sufiks*	*Jenis Kata*	*Pola Morfemik*	*Morfo_Kelas*
Pemerintah	perintah	pe-	-an	Nomina	pe- + V + -an	Pelaku/Kolektif
mendistribusikan	distribusi	meN-	-kan	Verba	meN- + N + -kan	Aksi kompleks
bantuan	bantu	-	-an	Nomina	V + -an	Hasil
korban	korban	-	-	Nomina	Kata dasar	Objek/Manusia
banjir	banjir	-	-	Nomina	Kata dasar	Lokasi/Kejadian

Dalam tabel tersebut, kita dapat melihat bagaimana satu token seperti "mendistribusikan" dipetakan ke dalam bentuk-bentuk morfologis yang lebih sederhana dan bermakna. Sistem dapat menggunakan kolom "Pola Morfemik" untuk membangun aturan atau melatih model yang peka terhadap struktur kata. Misalnya, pola "meN- + N + -kan" sering diasosiasikan dengan kata kerja transaktif yang menunjukkan aksi terhadap objek.

Kolom "Morfo_Kelas" bersifat semantik merangkum fungsi token berdasarkan bentuk morfologisnya. Kategori ini dapat digunakan sebagai label bantu dalam ekstraksi informasi, seperti mengidentifikasi kata kerja aksi utama dalam suatu kalimat, mengenali hasil dari suatu proses, atau mengklasifikasikan pelaku kejadian.

Dalam sistem rule-based, tabel ini memungkinkan penulisan aturan yang lebih presisi, misalnya:

"Jika kata memiliki pola meN- + N + -kan, maka tandai sebagai kata kerja utama dalam peristiwa."

Atau:

"Jika token memiliki sufiks -an dan berasal dari akar kata bantu, maka kategorikan sebagai entitas hasil bantuan."

Untuk sistem machine learning, tabel fitur morfologi ini dapat dikonversi menjadi feature vectors yang kemudian diumpankan ke model klasifikasi atau sekuens labeling. Dengan struktur tabel yang terorganisir, data morfologi dapat diproses dalam jumlah besar dan dimanfaatkan dalam berbagai teknik statistik maupun neural.

Penyusunan tabel ini dapat dilakukan secara manual dalam tahap awal eksperimen atau validasi, namun pada sistem berskala besar, proses ini sepenuhnya diotomatisasi menggunakan skrip berbasis Python, biasanya dengan pustaka seperti pandas untuk manipulasi tabel, Sastrawi atau MorphInd untuk stemming dan lemmatization, serta regex atau parser morfologi khusus untuk pola afiksasi.

Tabel ini juga berfungsi sebagai dokumentasi linguistik yang membantu dalam debugging sistem. Ketika ekstraksi informasi gagal mengenali entitas atau peristiwa, tabel fitur morfologi dapat ditelusuri untuk melihat apakah sistem salah memahami bentuk kata atau mengabaikan informasi morfologis penting.

Sebagai penutup, tabel fitur morfologi adalah komponen penting dalam pipeline ekstraksi informasi berbasis Bahasa Indonesia. Ia menjembatani pemahaman terhadap struktur kata dan pemrosesan komputasional dalam bentuk yang konsisten dan dapat diolah secara sistematik. Dengan penyusunan tabel yang baik, sistem akan memiliki fondasi kuat untuk menangani kompleksitas bahasa dan meningkatkan akurasi ekstraksi informasi secara keseluruhan.

E. Penerapan Fitur Morfologi dalam Studi Kasus

Untuk memperjelas manfaat dan penerapan fitur morfologi dalam sistem ekstraksi informasi, mari kita telaah sebuah studi kasus sederhana dari domain sosial dan kebencanaan. Fokus dari studi ini adalah bagaimana sistem dapat memanfaatkan informasi morfologis untuk mengekstraksi elemen peristiwa dari teks berita: seperti pelaku, aksi, dan objek, dengan tingkat akurasi dan konsistensi yang lebih baik.

Contoh kalimat yang diambil dari berita daring:

"Relawan menyalurkan bantuan logistik kepada warga terdampak banjir di Jakarta Timur."

Dalam sistem ekstraksi informasi yang cerdas, kita ingin mengenali bahwa:

- "Relawan" adalah pelaku (agent)
- "menyalurkan" adalah tindakan (verb)
- "bantuan logistik" adalah objek
- "warga terdampak" adalah penerima
- "banjir" adalah peristiwa
- "Jakarta Timur" adalah lokasi

Untuk mencapai hasil tersebut, sistem harus mampu mengenali struktur kata dan makna semantis berdasarkan bentuk morfologinya. Dengan menggunakan fitur morfologi, token-token kunci dalam kalimat dapat dianalisis dan dipetakan sebagai berikut:

Token	*Lemma*	*Prefiks*	*Sufiks*	*Kategori Morfologis*	*Fungsi dalam Kalimat*
Relawan	lawan	re-	-an	Nomina (orang)	Pelaku
menyalurkan	salur	meN-	-kan	Verba aktif kompleks	Aksi utama
bantuan	bantu	-	-an	Nomina (hasil)	Objek
logistik	logistik	-	-	Nomina (benda)	Penjelas objek
terdampak	dampak	ter-	-	Adjektiva/Pasif	Penanda kondisi
banjir	banjir	-	-	Nomina (kejadian)	Sumber peristiwa
Jakarta Timur	-	-	-	Entitas lokasi	Lokasi kejadian

Dengan informasi ini, sistem dapat menyusun struktur peristiwa yang utuh. Kata kerja utama "menyalurkan" dikenali sebagai tindakan karena memiliki prefiks "meN-" dan sufiks "-kan", dua ciri khas verba aktif dalam Bahasa Indonesia. "Bantuan" dikenali sebagai hasil dari aktivitas membantu karena berakhiran "-an". "Relawan" diidentifikasi sebagai pelaku karena merupakan bentuk nomina dari akar "lawan" dengan prefiks "re-" dan sufiks "-an" (yang dalam konteks sosial merujuk pada partisipan non-pemerintah atau sukarelawan).

Lebih lanjut, sistem dapat membedakan "terdampak" sebagai bentuk pasif dari kata "dampak", menunjukkan bahwa "warga" adalah penerima atau korban, bukan pelaku. Jika sistem tidak memiliki pemahaman morfologis yang baik, kemungkinan besar akan menyalahartikan peran entitas dalam kalimat tersebut, terutama pada kalimat yang memiliki struktur pasif atau kompleks.

Dalam kasus lain, misalnya dalam ekstraksi data dari laporan bantuan pemerintah, fitur morfologi dapat membantu sistem membedakan antara "menyalurkan", "penyaluran", dan "penyalur", yang semuanya berakar dari kata "salur" tetapi memiliki fungsi yang berbeda:

- "menyalurkan" → tindakan (verba)
- "penyaluran" → peristiwa atau proses (nomina)
- "penyalur" → pelaku atau pihak penyalur (nomina)

Dengan mengandalkan fitur seperti jenis afiks, struktur morfem, dan pola pembentukan kata, sistem dapat secara konsisten mengklasifikasikan kata-kata tersebut ke dalam peran semantik yang tepat.

Dalam implementasi praktis, fitur-fitur morfologi ini diintegrasikan ke dalam model berbasis aturan maupun

pembelajaran mesin. Misalnya, dalam model CRF atau BERT yang ditugaskan untuk menandai BIO-tag (Begin, Inside, Outside) pada entitas dan aksi, fitur morfologi dapat menjadi bagian dari input yang dikombinasikan dengan fitur leksikal dan kontekstual. Hasilnya, sistem dapat lebih akurat dalam mengidentifikasi batas entitas kompleks dan memahami peran kata dalam narasi peristiwa.

Studi kasus ini menegaskan bahwa fitur morfologi bukan hanya pelengkap dalam preprocessing, tetapi merupakan komponen kunci dalam memahami makna kata, fungsi sintaktik, dan relasi semantik dalam teks. Dalam Bahasa Indonesia, yang sangat produktif secara morfologis, penerapan fitur ini bahkan lebih krusial dibandingkan dengan bahasa yang struktur katanya lebih sederhana.

Sebagai penutup Bab 4, dapat disimpulkan bahwa pemahaman dan pemanfaatan fitur morfologi memberikan sistem ekstraksi informasi kemampuan untuk membaca teks dengan cara yang lebih mendalam tidak sekadar mengenali kata, tetapi juga memahami bentuk, struktur, dan maknanya dalam konteks yang lebih luas.

Latihan

A. Pertanyaan Pemahaman Konsep

1. Apa tujuan utama dari proses POS Tagging dalam sistem NLP?
2. Sebutkan minimal enam tag kelas kata utama dalam Bahasa Indonesia beserta contohnya.
3. Jelaskan perbedaan antara metode POS Tagging berbasis aturan dan berbasis *machine learning*.
4. Apa yang dimaksud dengan *ambiguity* dalam POS Tagging dan bagaimana cara mengatasinya?
5. Jelaskan perbedaan antara *constituent parsing* dan *dependency parsing*.
6. Mengapa Parsing Sintaksis penting dalam proses ekstraksi informasi dari teks?
7. Berikan contoh bagaimana struktur sintaksis membantu mendeteksi hubungan antar entitas dalam kalimat.

B. Latihan Praktik Sederhana

Diberikan kalimat berikut:

"Menteri Kesehatan menjelaskan kebijakan baru tentang vaksinasi kepada masyarakat."

Tugas:

1. Lakukan POS Tagging untuk setiap kata dalam kalimat di atas.
2. Gambarkan struktur *dependency parsing*-nya dalam bentuk pohon sederhana.
3. Tentukan hubungan kata kerja utama dengan subjek dan objeknya.
4. Jelaskan bagaimana hasil parsing ini bisa dimanfaatkan untuk ekstraksi informasi (misalnya mengenali siapa melakukan apa terhadap siapa).

C. Studi Kasus / Proyek Mini

Anda diminta membuat sistem sederhana untuk mengenali subjek dan objek dalam kalimat Bahasa Indonesia.

1. Tentukan langkah-langkah yang akan Anda lakukan mulai dari preprocessing hingga parsing.
2. Pilih satu pendekatan parsing (constituent atau dependency) dan jelaskan alasannya.
3. Gunakan contoh dua kalimat berbeda untuk menunjukkan hasil parsing-nya.
4. Analisis bagaimana hasil parsing membantu proses ekstraksi informasi relasional, misalnya: *(subjek, predikat, objek)*.

D. Diskusi / Refleksi

1. Menurut Anda, apakah POS Tagging masih diperlukan pada era model *transformer* seperti BERT? Mengapa?
2. Bagaimana parsing dapat membantu sistem memahami makna kalimat yang kompleks?
3. Diskusikan potensi tantangan dalam melakukan POS Tagging untuk Bahasa Indonesia dibandingkan Bahasa Inggris.
4. Bagaimana Anda melihat keterkaitan antara hasil parsing dan proses *feature assignment* yang telah dipelajari pada Bab 3?

BAB 5
PART-OF-SPEECH TAGGING DAN SPOK

Tujuan Pembelajaran

Setelah mempelajari Bab 5, mahasiswa diharapkan mampu:

1. Menjelaskan konsep dasar Named Entity Recognition (NER) dalam pengolahan bahasa alami.
2. Mengidentifikasi jenis-jenis entitas bernama (named entities) seperti *person*, *organization*, *location*, *date/time*, dan *numerical expressions*.
3. Membedakan pendekatan utama NER: rule-based, berbasis pembelajaran mesin (*machine learning*), dan berbasis *deep learning*.
4. Memahami tahapan kerja sistem NER, mulai dari preprocessing, feature extraction, hingga klasifikasi entitas.
5. Menjelaskan bagaimana POS Tagging dan Parsing Sintaksis mendukung proses NER.
6. Menerapkan contoh NER sederhana pada teks Bahasa Indonesia menggunakan pendekatan manual.
7. Mengevaluasi hasil NER dan mengenali potensi kesalahan (error analysis) seperti *entity boundary error* atau *entity type confusion*.

Pendahuluan

Dalam sistem ekstraksi informasi berbasis teks, pemahaman terhadap struktur sintaksis menjadi salah satu fondasi penting untuk menafsirkan fungsi setiap kata dalam sebuah kalimat. Salah satu teknik utama dalam analisis sintaksis ini adalah Part-of-Speech Tagging (disingkat POS Tagging), yaitu proses pelabelan kelas kata (kata benda, kerja,

sifat, dsb.) untuk setiap token dalam teks. POS tagging sangat berperan dalam mendukung identifikasi entitas, tindakan, serta struktur relasi antar bagian kalimat, terutama saat sistem dituntut untuk memahami kalimat kompleks, struktur pasif, atau inversi.

POS tagging sering dikombinasikan dengan struktur SPOK Subjek, Predikat, Objek, dan Keterangan yang merupakan kerangka dasar dalam tata bahasa Indonesia. Kombinasi ini memungkinkan sistem untuk menyusun interpretasi semantik dan sintaktik secara simultan.

A. Dasar Teori Part-of-Speech Tagging

Part-of-Speech Tagging adalah proses otomatis untuk menetapkan kelas gramatikal atau kategori sintaksis pada setiap kata dalam suatu teks. Kategori ini mencerminkan fungsi kata dalam kalimat, seperti kata benda (noun), kata kerja (verb), kata sifat (adjective), kata keterangan (adverb), dan lain sebagainya. Dalam Bahasa Indonesia, POS tagging mencakup beberapa kategori dasar seperti:

- NN: Kata benda (nama orang, benda, tempat)
- VB: Kata kerja
- JJ: Kata sifat
- RB: Kata keterangan
- IN: Preposisi (kata depan)
- CC: Konjungsi
- DT: Determiner (kata sandang atau penunjuk)
- PRP: Pronomina (kata ganti)
- UH: Interjeksi

Pelabelan ini penting karena banyak informasi semantik tersirat melalui fungsi gramatikal kata. Misalnya, untuk mengekstrak tindakan dalam kalimat, sistem perlu mengenali mana yang merupakan kata kerja. Begitu pula

untuk mengenali pelaku (entitas), sistem perlu mengenali kata benda atau frasa nominal yang berperan sebagai subjek atau objek.

Secara konseptual, POS tagging dapat dilakukan dengan dua pendekatan utama:

1. Pendekatan Berbasis Aturan (Rule-Based POS Tagging)

   Metode ini menggunakan serangkaian aturan linguistik dan daftar kata untuk menentukan kelas kata. Misalnya, aturan: "Jika sebuah kata diawali dengan 'me-' dan diikuti kata dasar, maka itu kemungkinan besar adalah kata kerja." Aturan ini dapat dikembangkan berdasarkan pola morfologi, urutan kata, atau kombinasi konteks.

2. Pendekatan Statistik atau Pembelajaran Mesin

   Dalam pendekatan ini, sistem mempelajari model dari data anotasi POS yang telah tersedia. Model yang umum digunakan termasuk Hidden Markov Model (HMM), Conditional Random Fields (CRF), dan pendekatan deep learning seperti BiLSTM atau BERT. Model ini mempertimbangkan distribusi kata dan urutan kelas dalam kalimat untuk memperkirakan kelas kata yang paling mungkin.

Contoh hasil POS tagging untuk kalimat:

"Petugas BPBD mengevakuasi korban banjir di daerah Cipinang." adalah:

Token	*POS*
Petugas	NN
BPBD	NNP
mengevakuasi	VB
korban	NN
banjir	NN
di	IN
daerah	NN
Cipinang	NNP

Dari hasil ini, sistem dapat menyimpulkan bahwa "Petugas BPBD" adalah subjek (NN + NNP), "mengevakuasi" adalah predikat (VB), "korban banjir" adalah objek (NN + NN), dan "di daerah Cipinang" adalah keterangan tempat (IN + NN + NNP). Informasi ini akan sangat penting ketika sistem EI berusaha menyusun struktur SPOK, mengenali relasi antar entitas, atau menyusun pola peristiwa.

POS tagging juga membantu dalam menghindari ambiguitas makna. Misalnya, kata "lari" bisa berarti kata kerja ("Ia lari dari rumah") atau kata benda ("Ia ikut lomba lari"). Dengan mengenali konteksnya, sistem dapat membedakan fungsi semantik kata tersebut dan menyesuaikan proses ekstraksi.

Dalam Bahasa Indonesia, tantangan POS tagging meliputi fleksibilitas urutan kata, keberadaan bentuk-bentuk pasif, partikel yang menempel (seperti "-lah", "-pun"), serta kata serapan yang terus berkembang. Oleh karena itu, POS tagging perlu dikombinasikan dengan preprocessing morfologis dan pemahaman kontekstual agar hasilnya lebih akurat dan stabil.

Dengan dasar teori ini, POS tagging menjadi langkah awal yang sangat strategis dalam proses ekstraksi informasi karena memungkinkan sistem untuk melakukan pelabelan sintaksis, memprediksi struktur kalimat, dan menyusun rangkaian entitas secara lebih presisi.

B. Kategori POS Tag Bahasa Indonesia

Kelas kata atau Part-of-Speech (POS) dalam Bahasa Indonesia merupakan komponen fundamental dalam analisis sintaktik karena setiap kata memiliki fungsi tertentu yang dapat dipetakan berdasarkan kategori gramatikalnya. POS tagging membantu sistem mengenali peran kata dalam kalimat, seperti apakah sebuah kata berfungsi sebagai subjek, predikat, objek, atau keterangan. Dalam konteks ekstraksi informasi, pelabelan POS yang akurat menjadi kunci untuk mengidentifikasi entitas, relasi, maupun aksi yang terdapat dalam teks. Oleh karena itu, pemahaman terhadap kategori POS dalam Bahasa Indonesia sangat krusial, terutama karena struktur kalimatnya yang relatif fleksibel dan kaya dengan bentuk afiksasi.

Kategori utama POS dalam Bahasa Indonesia dimulai dari nomina atau kata benda, yang biasanya diberi label NN untuk kata benda umum dan NNP untuk kata benda khusus atau proper noun. Kata seperti "rumah", "peristiwa", "bencana" termasuk dalam nomina umum, sedangkan "Jakarta", "BNPB", atau "Cianjur" merupakan nomina khusus. Kata benda dapat berfungsi sebagai subjek maupun objek dalam kalimat, dan sering kali menjadi entitas utama dalam proses ekstraksi.

Selanjutnya adalah verba atau kata kerja, yang merupakan inti dari predikat dalam kalimat. Kata kerja dalam Bahasa Indonesia memiliki ciri khas yang dapat

dikenali dari afiksasi seperti me-, di-, ber-, ter-, memper-, serta akhiran -kan dan -i. Dalam POS tagging, verba biasanya dilabeli sebagai VB. Contoh: "menyediakan", "dibantu", "berjalan", "terlihat". Dengan mengenali verba, sistem dapat menentukan tindakan atau peristiwa utama dalam kalimat, yang sering menjadi sasaran utama dalam sistem ekstraksi informasi berbasis kejadian atau relasi.

Adjektiva atau kata sifat merupakan kelas kata yang berfungsi memberikan keterangan atau kualitas terhadap nomina. Dalam POS tagging, biasanya diberi label JJ (adjective). Contoh kata sifat dalam Bahasa Indonesia meliputi "tinggi", "berbahaya", "parah", "baru", "lambat". Menariknya, dalam struktur Bahasa Indonesia, adjektiva sering berperan sebagai predikat langsung tanpa kata kerja bantu, seperti pada kalimat "Kondisinya parah." Tantangan muncul ketika sistem harus membedakan antara adjektiva dan verba dalam struktur kalimat yang sederhana atau tidak eksplisit.

Kelas kata selanjutnya adalah adverbia atau kata keterangan (RB), yang menjelaskan verba, adjektiva, atau adverbia lain. Adverbia meliputi kata seperti "cepat", "perlahan", "sekali", "sangat", "belum", "sudah". Adverbia penting dalam menambah informasi temporal, tingkat intensitas, dan cara dalam kalimat. Ia sering berperan dalam mengubah atau memperkuat makna tindakan, sehingga membantu sistem ekstraksi untuk memperkirakan durasi, urutan waktu, atau tingkat kepastian suatu peristiwa.

Preposisi atau kata depan (IN) merupakan kelas kata yang menghubungkan frasa nominal dengan bagian lain dalam kalimat. Contohnya adalah "di", "ke", "dari", "pada", "untuk", "dengan". Kata depan ini biasanya menjadi penanda keterangan tempat atau waktu, yang

sangat penting dalam struktur SPOK dan dalam mengaitkan entitas dengan lokasi atau waktu kejadian.

Konjungsi atau kata sambung (CC) berfungsi menghubungkan antar kata, frasa, atau klausa. Kata seperti "dan", "atau", "tetapi", "karena", "sehingga" termasuk dalam kategori ini. Konjungsi membantu sistem mengenali struktur kalimat majemuk dan hubungan sebab-akibat atau pertentangan antar bagian kalimat. Dalam proses ekstraksi informasi, ini penting untuk menyusun peristiwa yang melibatkan lebih dari satu aksi atau lebih dari satu entitas.

Kategori determiner atau kata penunjuk (DT) meliputi kata-kata seperti "setiap", "semua", "beberapa", "banyak", yang berfungsi menentukan jumlah atau spesifikasi dari nomina. Determiner memberi konteks tambahan terhadap entitas dan dapat membantu dalam mengklasifikasikan entitas sebagai tunggal, jamak, atau kolektif.

Pronomina atau kata ganti (PRP) mencakup kata-kata seperti "saya", "kami", "mereka", "dia", "itu", yang menggantikan nomina dalam kalimat. Penggunaan pronomina menjadi penting dalam analisis lintas kalimat (coreference resolution) karena sistem perlu memahami bahwa "dia" dalam kalimat kedua mungkin merujuk ke "Gubernur Jawa Barat" dalam kalimat pertama.

Kelas lain yang relevan adalah interjeksi (UH) seperti "aduh", "wah", "eh", yang biasanya muncul dalam teks percakapan atau media sosial, serta numeralia (CD) seperti "dua", "tiga belas", "2024", yang sering digunakan dalam ekstraksi informasi kuantitatif seperti jumlah korban atau tanggal kejadian.

Selain itu, sistem POS tagging modern juga dapat menangani kata serapan asing (FW – Foreign Word), kata

sandang, serta partikel khas Bahasa Indonesia seperti "-lah", "-pun", atau "kok", yang meskipun kecil, dapat memengaruhi interpretasi makna kalimat jika diabaikan.

Pemetaan POS tag dalam Bahasa Indonesia umumnya mengacu pada tagset standar yang digunakan oleh pustaka NLP seperti IndoNLU, IndoBERT, atau korpus seperti PANL10N, namun tetap dapat disesuaikan berdasarkan domain aplikasi. Untuk tugas ekstraksi informasi yang lebih spesifik seperti ekstraksi hukum, medis, atau bencana kategori POS dapat diperluas atau dikustomisasi untuk mencakup entitas khas dan bentuk kebahasaan yang relevan.

Dengan memahami keragaman dan fungsi kategori POS dalam Bahasa Indonesia, sistem ekstraksi informasi dapat melakukan pelabelan linguistik yang akurat, mengidentifikasi struktur SPOK secara lebih presisi, dan pada akhirnya meningkatkan kualitas hasil ekstraksi baik dari sisi entitas, relasi, maupun peristiwa yang terdeteksi dalam teks.

C. Struktur SPOK dalam Analisis Sintaktik

Struktur SPOK yang merupakan singkatan dari Subjek, Predikat, Objek, dan Keterangan adalah kerangka dasar dalam tata bahasa Indonesia yang digunakan untuk menganalisis fungsi sintaktik setiap bagian kalimat. Pemahaman terhadap struktur SPOK sangat penting dalam sistem ekstraksi informasi karena memungkinkan sistem untuk menginterpretasikan makna kalimat secara sistematis dan mengenali peran entitas serta aksi dalam wacana. Struktur ini menjadi fondasi dalam membangun representasi semantik, seperti siapa yang melakukan apa, kepada siapa, kapan, dan di mana. Dalam konteks NLP,

SPOK bertindak sebagai penghubung antara sintaksis dan semantik.

Dalam Bahasa Indonesia, struktur dasar kalimat umumnya mengikuti pola Subjek–Predikat–Objek–Keterangan, meskipun urutan ini bisa sangat fleksibel tergantung gaya penulisan, jenis teks, atau penekanan informasi. Subjek biasanya berupa nomina atau frasa nominal yang berperan sebagai pelaku, pemilik, atau topik utama. Predikat umumnya berupa verba atau frasa verbal yang menjelaskan tindakan, keadaan, atau proses. Objek adalah nomina atau frasa yang menjadi sasaran dari tindakan, sedangkan keterangan bisa berupa frasa adverbial atau preposisional yang memberikan informasi tambahan seperti tempat, waktu, cara, atau tujuan.

Contoh sederhana dari struktur SPOK dapat dilihat pada kalimat berikut:

"Petugas BNPB mengevakuasi korban banjir di daerah Cipinang."

Dalam kalimat ini:

- Subjek: "Petugas BNPB" → entitas pelaku
- Predikat: "mengevakuasi" → aksi utama
- Objek: "korban banjir" → entitas sasaran
- Keterangan: "di daerah Cipinang" → lokasi kejadian

SPOK tidak hanya memberikan struktur formal, tetapi juga menjadi kerangka untuk memahami relasi antar entitas. Dalam sistem ekstraksi informasi, struktur ini memungkinkan sistem mengenali bahwa "Petugas BNPB" adalah pelaku, bukan objek, sehingga informasi yang diperoleh menjadi lebih akurat dan tidak ambigu.

Dalam kalimat kompleks, misalnya kalimat majemuk atau yang memiliki struktur pasif, sistem perlu lebih cermat dalam menyusun SPOK. Kalimat seperti:

"Korban dievakuasi oleh relawan ke tempat pengungsian."

mengandung:

- Subjek (pasif): "Korban"
- Predikat: "dievakuasi"
- Pelaku (oleh + agent): "relawan"
- Keterangan tempat: "ke tempat pengungsian"

Di sini, subjek secara gramatikal bukanlah pelaku aksi, melainkan entitas yang menerima tindakan. Sistem ekstraksi informasi yang tidak mampu mengenali bentuk pasif mungkin akan menafsirkan "korban" sebagai pelaku, yang tentu saja akan menghasilkan data yang keliru. Oleh karena itu, selain POS tagging, pengenalan bentuk kalimat aktif atau pasif sangat penting untuk menyusun SPOK dengan tepat.

Dalam implementasi berbasis aturan (rule-based), struktur SPOK digunakan sebagai dasar untuk membuat pola-pola ekstraksi. Misalnya:

- Jika urutan token menunjukkan pola [NNP] + [VB] + [NN], maka sistem dapat mengasumsikan bahwa token pertama adalah subjek, kedua adalah predikat, dan ketiga adalah objek.
- Atau dalam bentuk pasif: [NN] + [VB passive] + oleh + [NNP] → subjek pasif + predikat + pelaku.

Untuk sistem berbasis pembelajaran mesin, informasi SPOK dapat dimodelkan sebagai features tambahan atau sebagai target anotasi dalam pelatihan model sekuensial. Misalnya, model bisa dilatih untuk mengenali batas frasa subjek dan predikat berdasarkan urutan POS tag dan dependensi kata.

SPOK juga berperan penting dalam menghubungkan kalimat satu dengan yang lain dalam wacana. Ketika sistem memproses paragraf panjang, struktur SPOK membantu dalam mempertahankan konteks entitas, melacak aksi yang dilakukan oleh siapa, dan menyatukan informasi temporal atau spasial yang tersebar di beberapa kalimat. Hal ini sangat berguna dalam sistem ekstraksi informasi yang dirancang untuk menyusun kronologi kejadian, analisis peran pelaku, atau pembangunan knowledge graph dari teks.

Dengan demikian, struktur SPOK bukan hanya kerangka linguistik untuk keperluan akademik, tetapi juga menjadi komponen praktis dan aplikatif dalam sistem NLP dan ekstraksi informasi. Dengan memanfaatkan analisis SPOK secara eksplisit, sistem dapat menafsirkan makna kalimat secara mendalam dan menyusun informasi yang lebih presisi, terutama dalam domain seperti laporan bencana, berita hukum, atau catatan medis yang menuntut pemahaman relasi antar entitas secara jelas.

D. Implementasi POS Tagging dalam Ekstraksi SPOK

Proses ekstraksi struktur SPOK dalam teks tidak dapat dipisahkan dari keberhasilan sistem dalam melakukan Part-of-Speech (POS) tagging. POS tagging memberikan informasi gramatikal dasar tentang fungsi setiap kata, yang kemudian digunakan untuk menyusun elemen Subjek, Predikat, Objek, dan Keterangan secara sistematis. Dalam sistem ekstraksi informasi, **implementasi** POS tagging menjadi langkah awal untuk memetakan struktur sintaksis, membentuk kerangka kalimat, dan menghubungkannya dengan representasi semantik yang dibutuhkan dalam proses ekstraksi entitas atau peristiwa.

Implementasi POS tagging untuk ekstraksi SPOK umumnya dilakukan dalam dua tahap besar. Tahap pertama adalah pelabelan POS itu sendiri, di mana setiap token diberi label berdasarkan kelas katanya. Tahap kedua adalah interpretasi pola POS tersebut untuk membangun struktur SPOK. Proses ini dapat dilakukan dengan pendekatan berbasis aturan (rule-based), statistik, maupun model pembelajaran mesin yang lebih kompleks.

Sebagai ilustrasi, ambil contoh kalimat:

"Gubernur DKI Jakarta meresmikan jembatan baru di kawasan Kalibata."

Hasil POS tagging dapat berupa:

Token	*POS*
Gubernur	NN
DKI	NNP
Jakarta	NNP
meresmikan	VB
jembatan	NN
baru	JJ
di	IN
kawasan	NN
Kalibata	NNP

Dari data ini, sistem dapat menerapkan aturan seperti:

- Token yang diawali dengan [NN] atau [NNP] dan diikuti oleh kata kerja [VB] → kemungkinan besar adalah Subjek + Predikat
- Token yang muncul setelah verba dan memiliki label [NN, JJ] → Objek

- Token yang diawali dengan preposisi [IN] dan diikuti oleh [NN/NNP] → Keterangan

Maka, sistem dapat membangun struktur SPOK sebagai berikut:

- Subjek: "Gubernur DKI Jakarta"
- Predikat: "meresmikan"
- Objek: "jembatan baru"
- Keterangan: "di kawasan Kalibata"

POS tagging tidak hanya mendeteksi kelas kata secara individual, tetapi juga memberikan urutan dan pola distribusi yang dapat dimanfaatkan oleh sistem untuk membedakan bagian-bagian kalimat. Pola semacam [NNP]+[VB]+[NN] dapat menjadi template pengenal struktur SPO dalam kalimat aktif, sementara pola seperti [NN]+[VB pasif]+[oleh]+[NNP] dapat dikenali sebagai struktur kalimat pasif.

Dalam sistem berbasis pembelajaran mesin, hasil POS tagging digunakan sebagai fitur input untuk model pelabelan sekuensial seperti Conditional Random Fields (CRF) atau model berbasis LSTM/BERT. Model ini belajar dari data anotasi SPOK dan mencoba memprediksi label struktur berdasarkan urutan POS tag serta konteks kata. Label yang umum digunakan meliputi:

- B-SUBJ / I-SUBJ (beginning/inside of subject)
- B-PRED / I-PRED (predikat)
- B-OBJ / I-OBJ (objek)
- B-KET / I-KET (keterangan)

Sebagai contoh, sistem dapat mempelajari bahwa frasa yang dimulai dengan [NNP][NNP] dan diikuti oleh [VB] kemungkinan besar adalah subjek dan predikat.

Model kemudian memanfaatkan POS sebagai salah satu indikator kuat, bersama dengan fitur lainnya seperti posisi kata, panjang frasa, atau kehadiran preposisi.

Salah satu keuntungan utama dari pendekatan berbasis POS dalam ekstraksi SPOK adalah fleksibilitasnya terhadap variasi struktur kalimat. Bahasa Indonesia memungkinkan berbagai bentuk inversi dan penggunaan keterangan yang luas di awal atau akhir kalimat. Sistem yang hanya mengandalkan urutan kata tanpa memahami fungsi gramatikal akan kesulitan dalam membedakan antara subjek dan objek, terutama dalam teks dengan struktur kompleks.

Contoh lain yang menarik adalah kalimat:

"Pada pukul 08.00 pagi tadi, korban berhasil dievakuasi oleh tim SAR."

POS tagging dan analisis SPOK akan membantu sistem memahami bahwa "korban" adalah subjek pasif, "dievakuasi" adalah predikat, dan "tim SAR" adalah pelaku dalam struktur pasif, meskipun posisi "korban" dan "oleh tim SAR" terpisah jauh. Keterangan waktu di awal kalimat ("Pada pukul 08.00 pagi tadi") juga bisa diidentifikasi dengan mudah karena keberadaan preposisi dan pola waktu yang khas.

Implementasi POS tagging dalam ekstraksi SPOK juga membuka jalan bagi visualisasi struktur kalimat secara grafis, seperti dependency trees atau diagram hubungan antar frasa. Hal ini sangat bermanfaat dalam debugging sistem atau memberikan penjelasan kepada pengguna tentang bagaimana sistem menafsirkan isi teks secara struktural.

Dengan demikian, POS tagging bukan sekadar pelabelan kata, melainkan komponen kunci dalam menyusun pemahaman kalimat yang mendalam melalui

kerangka SPOK. Dalam sistem ekstraksi informasi modern, kekuatan analitik ini dapat diintegrasikan ke dalam pipeline ekstraksi peristiwa, identifikasi relasi antar entitas, atau pembuatan ringkasan otomatis yang memahami "siapa melakukan apa, di mana, dan kapan."

E. Penerapan POS dan SPOK dalam Ekstraksi Informasi

Untuk memahami secara konkret bagaimana Part-of-Speech (POS) tagging dan struktur SPOK dapat dimanfaatkan dalam ekstraksi informasi, kita dapat melihat studi kasus sederhana dari domain berita bencana alam. Studi ini menunjukkan bagaimana sistem memanfaatkan informasi sintaktik untuk mengidentifikasi entitas, peran, dan relasi yang terkandung dalam sebuah kalimat.

Contoh kalimat berita:

"Tim SAR berhasil mengevakuasi lima korban tanah longsor di Desa Cijedil, Kabupaten Cianjur, pada Senin pagi."

Tujuan ekstraksi adalah mengambil informasi berikut:

- Pelaku (subjek): Tim SAR
- Tindakan (predikat): mengevakuasi
- Objek/korban: lima korban tanah longsor
- Lokasi: Desa Cijedil, Kabupaten Cianjur
- Waktu: Senin pagi

Langkah pertama adalah melakukan POS tagging terhadap kalimat tersebut. Hasilnya kurang lebih sebagai berikut:

Token	POS	Token	POS
Tim	NN	di	IN
SAR	NNP	Desa	NNP
berhasil	RB	Cijedil	NNP
mengevakuasi	VB	,	PUNCT
lima	CD	Kabupaten	NNP
korban	NN	Cianjur	NNP
tanah	NN	pada	IN
longsor	NN	Senin	NNP
		pagi	NN

Dari urutan POS ini, sistem dapat mengenali pola berikut:

- Subjek: [NN] + [NNP] = "Tim SAR"
- Predikat: [VB] = "mengevakuasi" (dengan adverbia pendukung "berhasil")
- Objek: [CD] + [NN] + [NN] + [NN] = "lima korban tanah longsor"
- Keterangan tempat: diawali dengan preposisi [IN] → "di Desa Cijedil, Kabupaten Cianjur"
- Keterangan waktu: juga dengan preposisi [IN] → "pada Senin pagi"

Dengan memahami POS dan relasi sintaksis berdasarkan urutan ini, sistem ekstraksi informasi dapat menyusun template peristiwa, seperti:

```
php-template
SalinEdit
<Subjek> = Tim SAR
<Predikat> = mengevakuasi
<Objek> = lima korban tanah longsor
<Tempat> = Desa Cijedil, Kabupaten Cianjur
<Waktu> = Senin pagi
```

Struktur ini dapat diformat ke dalam JSON, tabel database, atau langsung digunakan untuk membangun

sistem laporan otomatis. Sebagai contoh, sistem dapat menampilkan kalimat ringkasan:

"Tim SAR mengevakuasi lima korban tanah longsor pada Senin pagi di Desa Cijedil, Cianjur."

Penerapan ini tidak terbatas pada berita. Dalam dokumen laporan, email tanggap darurat, atau bahkan input media sosial, pola POS dan SPOK dapat membantu sistem untuk tetap konsisten dalam memahami siapa yang melakukan apa, terhadap siapa, kapan, dan di mana. Bahkan dalam kalimat yang memiliki struktur tidak baku, selama POS tagging cukup akurat, sistem masih dapat mendeteksi peran sintaktik dengan cukup baik.

Sebagai ilustrasi lain, pertimbangkan kalimat dari pesan WhatsApp relawan:

"Pagi ini sudah ada 20 korban yang dibawa ke posko kesehatan di SDN 2."

Sistem akan mengenali:

- Subjek implisit: tidak eksplisit, tetapi dapat diasumsikan dari konteks ("relawan", "tim medis", dsb.)
- Predikat: "dibawa" (kata kerja pasif)
- Objek: "20 korban"
- Keterangan tempat: "di posko kesehatan di SDN 2"
- Keterangan waktu: "Pagi ini"

Meskipun strukturnya tidak seformal teks berita, POS tagging dan analisis SPOK tetap dapat digunakan untuk mengekstrak data kunci secara konsisten.

Dari studi kasus ini, dapat disimpulkan bahwa kombinasi POS tagging dan struktur SPOK membentuk fondasi yang kuat untuk ekstraksi informasi dalam Bahasa Indonesia. Sistem yang dibangun di atas analisis sintaktik ini menjadi lebih tangguh terhadap variasi bahasa, mampu menyusun struktur peristiwa secara otomatis, dan siap

digunakan dalam aplikasi seperti pelaporan otomatis, pemetaan peristiwa, chatbot bantuan bencana, serta sistem intelijen berbasis teks.

Penerapan yang efektif tentu membutuhkan POS tagger yang akurat, model SPOK yang fleksibel, serta integrasi yang baik dengan modul ekstraksi entitas dan normalisasi. Namun, begitu komponen-komponen ini bekerja secara terpadu, hasilnya adalah sistem yang dapat "membaca" teks seperti manusia: memahami makna, peran, dan hubungan dalam setiap kalimat yang ditulis.

Latihan

A. Pertanyaan Pemahaman Konsep

1. Apa yang dimaksud dengan *Named Entity Recognition* (NER)?
2. Sebutkan minimal lima jenis entitas yang umum diekstraksi oleh sistem NER.
3. Mengapa NER dianggap sebagai komponen penting dalam sistem ekstraksi informasi?
4. Jelaskan perbedaan pendekatan rule-based dan *machine learning* pada NER.
5. Apa peran POS Tagging dalam mendukung kinerja NER?
6. Mengapa Bahasa Indonesia memiliki tantangan tersendiri dalam pengenalan entitas bernama?
7. Apa yang dimaksud dengan *entity boundary error* dalam sistem NER?

B. Latihan Praktik Sederhana

Diberikan teks berita berikut:

"Presiden Joko Widodo menghadiri pertemuan ASEAN di Bangkok pada tanggal 5 September 2024."

Tugas:

1. Lakukan identifikasi entitas secara manual dan kategorikan ke dalam tipe: *Person*, *Organization*, *Location*, dan *Date*.
2. Sajikan hasil ekstraksi dalam bentuk tabel berikut:

Kata/Frasa	Jenis Entitas	Keterangan
Presiden Joko Widodo	Person	Kepala Negara
ASEAN	Organization	Organisasi Regional
Bangkok	Location	Ibukota Thailand
5 September 2024	Date	Tanggal Kegiatan

3. Jelaskan bagaimana konteks kalimat membantu sistem mengenali entitas tersebut.
4. Sebutkan fitur linguistik yang dapat digunakan untuk membedakan antara *organization* dan *location*.

C. Studi Kasus / Proyek Mini

Anda akan membangun sistem NER sederhana untuk artikel berita ekonomi.

1. Tentukan kategori entitas yang paling relevan (misalnya *Organization*, *Currency*, *Date*, *Person*).
2. Jelaskan tahapan pipeline dari preprocessing hingga output NER.
3. Buat contoh tiga kalimat dan tandai setiap entitas menggunakan format BIO (misal: B-PER, I-ORG, B-LOC, O).
4. Analisis kesalahan potensial yang bisa muncul jika sistem hanya menggunakan *dictionary-based approach*.
5. Diskusikan bagaimana pendekatan *neural network* dapat memperbaiki kelemahan metode berbasis aturan.

D. Diskusi / Refleksi

1. Mengapa konteks kalimat sangat penting dalam menentukan jenis entitas?
2. Menurut Anda, apakah NER untuk Bahasa Indonesia memerlukan pelabelan manual yang lebih banyak dibanding bahasa lain?
3. Bagaimana perkembangan model *transformer* seperti IndoBERT memengaruhi akurasi NER?
4. Diskusikan kemungkinan penggabungan NER dengan *sentiment analysis* untuk aplikasi analitik teks.

BAB 6
PENGENALAN NATURAL LANGUAGE PROCESSING (NLP)

Tujuan Pembelajaran

Setelah mempelajari Bab 6, mahasiswa diharapkan mampu:

1. Menjelaskan konsep dasar Relation Extraction (RE) dan perannya dalam sistem *Information Extraction*.
2. Membedakan antara Named Entity Recognition (NER) dan Relation Extraction serta memahami keterkaitannya.
3. Mengidentifikasi jenis-jenis relasi yang umum diekstraksi seperti hubungan personal (*person–organization*), geografis (*location–event*), atau temporal (*event–date*).
4. Menjelaskan pendekatan utama dalam ekstraksi relasi: rule-based, supervised learning, unsupervised, dan *deep learning*.
5. Menganalisis struktur sintaksis dan semantik yang digunakan untuk mengenali hubungan antar entitas.
6. Menerapkan contoh sederhana ekstraksi relasi dari teks berbahasa Indonesia.
7. Mengevaluasi hasil RE dan mengenali kesalahan umum dalam proses ekstraksi relasi.

Pendahuluan

Perkembangan teknologi informasi di era digital telah membawa kita pada situasi di mana teks menjadi salah satu bentuk data yang paling dominan. Teks hadir di berbagai medium: artikel berita, dokumen hukum, rekam medis, hingga percakapan di media sosial. Data teks yang begitu besar dan beragam ini tidak dapat diolah secara manual,

sehingga diperlukan metode komputasional yang mampu memahami bahasa manusia secara otomatis.

Natural Language Processing (NLP) hadir sebagai disiplin ilmu yang menjembatani bahasa manusia dengan sistem komputer. Melalui NLP, komputer dapat melakukan analisis, pemahaman, bahkan menghasilkan teks baru dalam bahasa alami. Mulai dari tugas sederhana seperti tokenisasi dan stemming, hingga aplikasi canggih seperti penerjemahan mesin, chatbot, analisis sentimen, dan ekstraksi informasi (Information Extraction/IE), semuanya bertumpu pada fondasi NLP.

Bagi mahasiswa yang mempelajari Ekstraksi Informasi dari Teks, pemahaman tentang NLP merupakan bekal wajib. Hal ini karena hampir seluruh proses IE bergantung pada tahapan NLP, baik yang bersifat dasar seperti text preprocessing maupun yang lebih kompleks seperti Named Entity Recognition (NER) dan relation extraction. Dengan kata lain, NLP adalah fondasi teknis, sedangkan IE adalah aplikasi yang dibangun di atasnya.

Pada bab ini, kita akan membahas:

- Konsep dasar dan cakupan NLP.
- Hubungan NLP dengan ekstraksi informasi.
- Pipeline NLP untuk analisis teks.
- Tools dan library NLP populer.
- Studi kasus aplikasi NLP dalam ekstraksi informasi.

Dengan mempelajari bab ini, mahasiswa diharapkan dapat memahami peran penting NLP sebagai dasar pengolahan teks, mengenali pipeline pemrosesan teks yang umum digunakan, serta mengidentifikasi bagaimana NLP mendukung berbagai aplikasi analitik berbasis bahasa alami.

A. Konsep dan Cakupan NLP

Natural Language Processing (NLP) merupakan cabang dari kecerdasan buatan yang berfokus pada bagaimana komputer dapat memahami, memproses, dan menghasilkan bahasa alami sebagaimana digunakan manusia sehari-hari. Bahasa alami di sini mencakup baik bahasa tulis maupun lisan, seperti Bahasa Indonesia, Inggris, Jepang, Arab, dan bahasa lainnya. Tujuan utama NLP adalah menjembatani komunikasi antara manusia dan mesin. Hal ini penting karena manusia terbiasa menggunakan bahasa yang penuh dengan nuansa, ambiguitas, serta konteks sosial, sementara komputer hanya memahami data dalam bentuk numerik dan logika formal. Dengan demikian, NLP dapat dipandang sebagai penerjemah yang menjembatani dunia manusia dengan dunia mesin.

Sebagai disiplin ilmu, NLP bersifat interdisipliner karena menggabungkan tiga bidang utama. Pertama, linguistik yang berfokus pada struktur bahasa mulai dari fonologi, morfologi, sintaksis, semantik, hingga pragmatik. Kedua, ilmu komputer yang menyediakan algoritma, struktur data, serta metode komputasi untuk mengolah teks. Ketiga, matematika dan statistik yang menjadi fondasi bagi model probabilistik, machine learning, dan deep learning yang banyak digunakan dalam NLP modern. Dengan landasan ini, NLP berkembang menjadi bidang yang sangat dinamis dan terus mengalami lompatan teknologi.

Perkembangan NLP sendiri dapat dibagi ke dalam beberapa fase. Pada era awal tahun 1950–1980-an, NLP masih didominasi pendekatan berbasis aturan (rule-based). Sistem dibangun dengan seperangkat aturan tata bahasa formal yang dirancang secara manual untuk

mengenali struktur kalimat. Namun pendekatan ini lemah ketika menghadapi variasi bahasa yang luas. Memasuki tahun 1990-an, paradigma bergeser ke pendekatan statistik, di mana bahasa dianalisis dengan memanfaatkan probabilitas dan model matematis, misalnya n-gram untuk memprediksi kata berikutnya. Pada tahun 2000-an, teknik pembelajaran mesin (machine learning) mulai mendominasi, dengan algoritma seperti Support Vector Machine (SVM) dan Conditional Random Fields (CRF) yang digunakan untuk tugas-tugas seperti Part-of-Speech Tagging dan Named Entity Recognition. Fase terbaru sejak 2010-an adalah era deep learning dengan model berbasis neural networks, khususnya arsitektur Transformer seperti BERT dan GPT, yang memungkinkan komputer memahami konteks kalimat dengan jauh lebih baik bahkan melampaui manusia dalam beberapa benchmark.

Ruang lingkup NLP dapat dipetakan ke dalam beberapa level analisis bahasa. Pada level fonologi dan morfologi, NLP berhubungan dengan suara, pelafalan, dan pembentukan kata. Level ini relevan dalam aplikasi seperti speech recognition dan sangat penting dalam proses stemming dan lemmatization, terutama pada bahasa yang kaya imbuhan seperti Bahasa Indonesia. Pada level sintaksis, fokusnya adalah struktur kalimat dan aturan tata bahasa, yang meliputi proses tokenisasi, pelabelan kelas kata (Part-of-Speech Tagging), dan parsing. Pada level semantik, NLP berusaha memahami makna kata dalam konteks, misalnya dalam Word Sense Disambiguation, Named Entity Recognition, dan Semantic Role Labeling. Level berikutnya adalah pragmatik dan diskursus, yang menangani konteks sosial serta hubungan antar kalimat. Pada level ini, sistem NLP berurusan

dengan coreference resolution, analisis wacana, hingga pengembangan chatbot dan sistem dialog.

Cakupan NLP tidak berhenti pada analisis linguistik semata, tetapi mencakup beragam tugas aplikasi. Beberapa di antaranya adalah information retrieval untuk pencarian dokumen relevan, information extraction untuk mengambil informasi terstruktur dari teks tidak terstruktur, machine translation seperti Google Translate, text classification untuk mengelompokkan teks dalam kategori tertentu, sentiment analysis untuk mendeteksi opini positif atau negatif, summarization untuk menghasilkan ringkasan otomatis dari dokumen panjang, serta question answering untuk menjawab pertanyaan berdasarkan teks. Selain itu, ada juga speech recognition dan text-to-speech synthesis yang mengubah suara menjadi teks dan sebaliknya.

Walaupun perkembangannya pesat, NLP menghadapi banyak tantangan. Bahasa alami bersifat ambigu sehingga kata atau frasa dapat memiliki arti berbeda tergantung konteks. Sumber daya korpus anotasi untuk Bahasa Indonesia masih sangat terbatas, sehingga model pembelajaran mesin sering kali tidak memiliki data latih yang cukup. Variasi bahasa yang digunakan masyarakat, termasuk bahasa gaul, singkatan, dan campur kode, semakin memperumit pemrosesan. Kompleksitas morfologi Bahasa Indonesia yang kaya imbuhan juga menjadi kendala dalam proses normalisasi. Selain itu, memahami aspek pragmatik, ironi, atau sarkasme masih merupakan tantangan besar bagi mesin.

Sebagai ilustrasi penerapan, perhatikan kalimat berikut: “Presiden Joko Widodo meresmikan Bendungan Napun Gete di Nusa Tenggara Timur pada 23 Februari 2022.” Proses NLP akan memecah kalimat ini menjadi

token-token individu, melakukan Part-of-Speech Tagging seperti menandai "Presiden" sebagai kata benda dan "meresmikan" sebagai kata kerja, serta melakukan Named Entity Recognition yang mengidentifikasi "Joko Widodo" sebagai entitas orang, "Bendungan Napun Gete" sebagai fasilitas, "Nusa Tenggara Timur" sebagai lokasi, dan "23 Februari 2022" sebagai tanggal. Hasil akhir dari proses NLP ini adalah data terstruktur yang dapat langsung digunakan dalam sistem ekstraksi informasi untuk membangun relasi siapa melakukan apa, di mana, dan kapan.

Dengan demikian, konsep dan cakupan NLP tidak hanya terbatas pada teori linguistik, tetapi juga meliputi seluruh rangkaian proses teknis yang menjadikan teks dapat dipahami oleh komputer. NLP adalah pondasi bagi ekstraksi informasi dan berbagai aplikasi cerdas lainnya. Memahami cakupan ini menjadi penting bagi mahasiswa, karena memberikan gambaran menyeluruh mengenai posisi NLP dalam pengolahan bahasa alami dan aplikasinya di dunia nyata.

B. Hubungan NLP dan Ekstraksi Informasi

Natural Language Processing (NLP) dan Ekstraksi Informasi (Information Extraction/IE) memiliki keterkaitan yang sangat erat. NLP dapat dianggap sebagai fondasi yang memungkinkan proses ekstraksi informasi berjalan dengan baik, sementara IE merupakan salah satu aplikasi utama dari NLP. Tanpa teknik NLP yang memadai, sistem ekstraksi informasi akan kesulitan mengenali entitas, memahami struktur kalimat, serta menafsirkan makna teks yang ambigu. Sebaliknya, keberadaan IE menunjukkan nilai praktis dari NLP, karena

hasil analisis linguistik dapat diwujudkan menjadi data terstruktur yang bermanfaat dalam berbagai aplikasi.

Secara sederhana, NLP menyediakan "alat bantu" untuk memahami teks, sedangkan IE bertugas menyusun informasi dari teks tersebut agar lebih terorganisasi. Misalnya, NLP dapat melakukan tokenisasi untuk memecah kalimat menjadi kata, Part-of-Speech tagging untuk menandai kelas kata, serta Named Entity Recognition (NER) untuk mengidentifikasi nama orang, tempat, organisasi, atau tanggal. Hasil dari tahapan ini kemudian dimanfaatkan oleh IE untuk menghubungkan entitas tersebut menjadi sebuah fakta atau relasi. Sebagai contoh, dalam kalimat "Gubernur Jawa Barat Ridwan Kamil meresmikan jembatan di Bandung," NLP akan mengenali "Ridwan Kamil" sebagai entitas orang, "Jawa Barat" sebagai lokasi administratif, serta "Bandung" sebagai kota. IE kemudian menyusun informasi ini menjadi fakta bahwa Ridwan Kamil (orang) dengan jabatan Gubernur Jawa Barat melakukan aksi peresmian jembatan di Bandung.

Hubungan ini semakin jelas bila kita melihat pipeline pemrosesan teks. Tahapan awal seperti pembersihan data, tokenisasi, dan stemming merupakan proses dasar NLP yang berfungsi menyiapkan teks mentah agar lebih mudah dianalisis. Tahapan selanjutnya, yaitu analisis sintaksis melalui POS tagging dan parsing, membantu sistem memahami struktur kalimat, misalnya menentukan subjek, predikat, dan objek. Analisis semantik kemudian memperkaya pemahaman dengan memberikan label makna, misalnya mengidentifikasi entitas orang atau organisasi. Seluruh keluaran dari tahapan NLP ini menjadi masukan (input) yang sangat penting bagi IE. Dengan kata

lain, IE tidak bekerja secara terpisah, melainkan berdiri di atas fondasi hasil analisis NLP.

Keterkaitan ini juga dapat dilihat dari perspektif tujuan. NLP bertujuan untuk memahami bahasa manusia dalam level linguistik yang mendalam, sedangkan IE lebih fokus pada bagaimana hasil pemahaman itu dapat dipakai untuk mengisi basis data, membangun relasi antar entitas, atau mendukung sistem pengambilan keputusan. Misalnya, sistem IE pada sektor kesehatan tidak hanya mengekstrak kata "diagnosis" atau "obat," tetapi juga perlu memastikan bahwa kata tersebut merujuk pada pasien tertentu, waktu tertentu, dan hasil pemeriksaan tertentu. Untuk mencapai hal tersebut, sistem harus memanfaatkan analisis NLP seperti coreference resolution agar dapat memahami bahwa kata "dia" dalam catatan medis merujuk pada pasien yang disebutkan di kalimat sebelumnya.

Dengan semakin berkembangnya teknologi NLP, hubungan ini menjadi semakin erat. Model pra-latih seperti BERT, RoBERTa, dan IndoBERT memungkinkan IE bekerja lebih akurat karena sistem dapat memahami konteks kata secara lebih dalam. Misalnya, kata "Jakarta" dapat berarti kota, pemerintah daerah, atau tim olahraga. Tanpa NLP yang mampu memahami konteks, IE akan cenderung salah menafsirkan entitas. Model NLP modern memberikan kemampuan disambiguasi yang lebih kuat, sehingga hasil ekstraksi informasi menjadi lebih presisi.

Secara praktis, hubungan NLP dan IE dapat dianalogikan seperti hubungan antara fondasi dan bangunan. NLP adalah fondasi yang menyediakan pemahaman linguistik dasar, sedangkan IE adalah bangunan yang memanfaatkan fondasi tersebut untuk menghasilkan informasi yang berguna. Jika fondasi NLP

rapuh misalnya POS tagging sering salah atau NER tidak akurat maka IE akan menghasilkan informasi yang keliru. Namun, bila NLP yang digunakan kuat, hasil IE akan jauh lebih dapat diandalkan. Oleh karena itu, dalam pengembangan sistem ekstraksi informasi, perhatian besar harus diberikan pada pemilihan metode dan model NLP yang sesuai dengan bahasa serta domain yang digunakan.

C. Pipeline NLP untuk Analisis Teks

Pipeline Natural Language Processing (NLP) adalah rangkaian tahapan sistematis yang dirancang untuk mengubah teks mentah menjadi representasi yang dapat dipahami dan dianalisis oleh komputer. Istilah pipeline mengacu pada alur kerja berlapis, di mana keluaran dari satu tahap menjadi masukan bagi tahap berikutnya. Konsep ini penting karena teks alami yang ditulis manusia pada dasarnya tidak terstruktur, penuh variasi, dan sering kali mengandung ambiguitas. Tanpa alur pemrosesan yang jelas, sistem komputer akan kesulitan mengenali makna atau menyusun informasi yang terkandung di dalam teks.

Pada tahap paling awal, pipeline dimulai dari pengumpulan dan input data teks. Data dapat bersumber dari artikel berita, dokumen hukum, laporan medis, media sosial, hingga hasil transkrip percakapan. Sifat teks mentah ini sangat beragam, misalnya dalam bentuk paragraf panjang, kalimat tidak baku, atau bahkan kombinasi dengan simbol, angka, dan tanda baca yang tidak konsisten. Karena itu, langkah pertama yang dilakukan adalah text preprocessing, yaitu proses pembersihan dan penyeragaman teks. Preprocessing mencakup normalisasi kata, penghapusan karakter yang tidak relevan, tokenisasi, hingga stemming atau

lemmatisasi. Tujuan utama tahap ini adalah menghasilkan teks yang bersih dan seragam agar lebih mudah dianalisis pada tahap berikutnya.

Setelah teks dibersihkan, pipeline berlanjut ke analisis morfologis dan sintaksis. Analisis morfologis berfokus pada bentuk kata, misalnya mengembalikan kata berimbuhan ke bentuk dasarnya, sedangkan analisis sintaksis berusaha memahami struktur kalimat. Pada tahap ini biasanya dilakukan Part-of-Speech (POS) tagging, yaitu pelabelan setiap kata sesuai kelas katanya (nomina, verba, adjektiva, dan lain-lain). Informasi mengenai struktur kalimat juga diperoleh melalui parsing, baik berbentuk constituency parsing maupun dependency parsing, yang memberikan gambaran hubungan antar kata seperti subjek, predikat, objek, dan keterangan.

Tahap selanjutnya adalah analisis semantik, di mana sistem berusaha memahami makna yang terkandung dalam teks. Analisis ini meliputi pengenalan entitas bernama atau Named Entity Recognition (NER) untuk menemukan nama orang, lokasi, organisasi, tanggal, jumlah, dan kategori lain yang penting. Pada level ini juga dilakukan tugas seperti coreference resolution, yang memungkinkan sistem memahami bahwa kata "dia" atau "mereka" merujuk pada entitas yang disebutkan sebelumnya. Dengan analisis semantik, teks yang awalnya berupa rangkaian kata menjadi lebih bermakna karena informasi penting dapat diidentifikasi secara eksplisit.

Hasil analisis semantik kemudian dapat dimanfaatkan dalam tahap Information Extraction (IE) atau aplikasi NLP tingkat lanjut. Pada tahap ini, informasi yang sudah teridentifikasi disusun kembali menjadi struktur yang lebih terorganisasi. Misalnya, entitas yang telah dikenali dapat dihubungkan dalam bentuk relasi,

sehingga dari satu kalimat dapat dihasilkan fakta seperti: "Ridwan Kamil (orang) – meresmikan (aksi) – jembatan (objek) – di Bandung (lokasi) – pada 2022 (waktu)." Proses ini memungkinkan teks tidak hanya dipahami secara linguistik, tetapi juga diubah menjadi data terstruktur yang dapat langsung digunakan untuk analisis lebih lanjut, basis data, maupun sistem pengambilan keputusan.

Pipeline NLP biasanya diakhiri dengan output dalam bentuk terstruktur. Bentuk keluaran ini bisa berupa tabel data, grafik hubungan antar entitas, ringkasan teks, atau bahkan knowledge graph yang memetakan fakta-fakta dari berbagai dokumen. Misalnya, dalam konteks analisis bencana, pipeline NLP dapat menghasilkan tabel berisi lokasi kejadian, jumlah korban, jenis bencana, dan waktu kejadian dari ratusan artikel berita secara otomatis.

Meskipun pipeline NLP umumnya digambarkan sebagai urutan linear, dalam praktiknya sering terjadi variasi dan penyesuaian. Tidak semua aplikasi membutuhkan seluruh tahapan. Sebagai contoh, sistem analisis sentimen mungkin hanya membutuhkan preprocessing, tokenisasi, dan klasifikasi, tanpa parsing mendalam. Sebaliknya, sistem ekstraksi informasi hukum mungkin membutuhkan parsing yang sangat detail untuk memahami struktur pasal dan kalimat majemuk. Dengan demikian, pipeline NLP harus dirancang secara fleksibel sesuai dengan kebutuhan domain dan tujuan analisis.

Keberhasilan pipeline NLP sangat bergantung pada kualitas setiap tahapan. Jika preprocessing tidak optimal, maka tokenisasi bisa menghasilkan kata yang keliru, yang pada gilirannya berdampak pada POS tagging dan Named Entity Recognition. Hal ini menunjukkan bahwa pipeline bersifat hierarkis: kesalahan di tahap awal akan menimbulkan efek berantai pada tahap-tahap berikutnya.

Oleh karena itu, perhatian khusus perlu diberikan pada desain pipeline, pemilihan metode, serta kualitas data yang digunakan.

Dalam perkembangannya, pipeline NLP modern semakin banyak memanfaatkan model pra-latih berbasis deep learning, seperti BERT atau GPT, yang mampu melakukan banyak tugas sekaligus. Model ini dapat melakukan tokenisasi, POS tagging, NER, bahkan ekstraksi relasi dalam satu kerangka kerja terpadu, sehingga mengurangi kebutuhan pipeline yang terfragmentasi. Walaupun demikian, pemahaman mengenai pipeline tradisional tetap penting, karena memberikan gambaran logis tentang bagaimana teks diproses secara bertahap dari bentuk mentah hingga menjadi data terstruktur.

Secara keseluruhan, pipeline NLP untuk analisis teks adalah inti dari pemrosesan bahasa alami. Pipeline ini menggambarkan perjalanan sebuah teks dari bentuk mentah yang penuh variasi menjadi representasi yang rapi, bermakna, dan siap digunakan dalam berbagai aplikasi. Pemahaman mengenai pipeline ini tidak hanya bermanfaat bagi pengembang sistem NLP, tetapi juga penting bagi mahasiswa dan peneliti yang ingin memahami bagaimana proses analisis teks berlangsung secara komputasional.

D. Tools dan Library NLP Populer (spaCy, NLTK, dsb.)

Seiring perkembangan Natural Language Processing (NLP), komunitas akademik maupun industri telah mengembangkan berbagai perangkat lunak, library, dan kerangka kerja (framework) yang memudahkan peneliti, mahasiswa, serta praktisi dalam mengimplementasikan metode NLP. Jika pada masa awal penelitian NLP seorang

peneliti harus membangun algoritma dari nol, saat ini tersedia beragam pustaka siap pakai yang dapat langsung digunakan untuk preprocessing teks, part-of-speech tagging, parsing, hingga tugas yang lebih kompleks seperti Named Entity Recognition (NER) atau text classification. Kehadiran tools ini membuat proses penelitian dan pengembangan aplikasi NLP menjadi jauh lebih cepat, praktis, dan efisien.

Salah satu library yang paling banyak digunakan dalam dunia akademik adalah NLTK (Natural Language Toolkit). NLTK dikembangkan dengan tujuan utama sebagai media pembelajaran dan penelitian di bidang linguistik komputasional. Pustaka ini ditulis dalam bahasa Python dan menyediakan modul untuk tokenisasi, stemming, lemmatization, part-of-speech tagging, parsing, hingga analisis semantik sederhana. Selain itu, NLTK dilengkapi dengan banyak dataset dan korpus yang dapat digunakan langsung, misalnya Brown Corpus atau WordNet. Karena fokus utamanya pada edukasi, NLTK relatif mudah dipahami oleh pemula dan sangat bermanfaat sebagai media latihan mahasiswa. Namun, kelemahan NLTK adalah performanya yang tidak terlalu efisien jika digunakan untuk aplikasi berskala industri.

Berbeda dengan NLTK yang bersifat edukatif, spaCy lebih berorientasi pada penggunaan praktis di dunia industri. SpaCy dirancang dengan fokus pada kecepatan, efisiensi, dan integrasi dengan aplikasi lain. Library ini mendukung berbagai bahasa, termasuk Bahasa Indonesia melalui model tambahan yang disediakan komunitas. SpaCy sangat unggul untuk tugas-tugas Named Entity Recognition, dependency parsing, dan text classification. Kelebihan lainnya adalah arsitektur spaCy yang modular dan mudah diintegrasikan dengan deep learning

framework seperti TensorFlow atau PyTorch, sehingga sangat cocok untuk membangun aplikasi NLP modern. SpaCy juga menyediakan pre-trained models yang dapat langsung digunakan tanpa perlu melatih model dari awal.

Selain NLTK dan spaCy, ada pula Stanza yang dikembangkan oleh Stanford NLP Group. Stanza merupakan penerus Stanford CoreNLP dan dibangun dengan teknologi deep learning yang modern. Stanza menawarkan kemampuan analisis linguistik yang lebih canggih, termasuk dependency parsing, NER, dan POS tagging dengan akurasi tinggi. Salah satu kelebihan Stanza adalah dukungan multi-bahasa, termasuk Bahasa Indonesia, sehingga dapat digunakan untuk penelitian lintas bahasa. Library ini juga menyediakan model berbasis neural network yang telah dilatih dengan data beranotasi, sehingga pengguna dapat langsung memanfaatkannya tanpa perlu menyusun dataset dari awal.

Dalam konteks Bahasa Indonesia, komunitas riset telah mengembangkan beberapa pustaka khusus, misalnya IndoNLP, IndoBERT, atau IndoBERTweet. Library dan model ini didesain dengan memperhatikan karakteristik Bahasa Indonesia, termasuk struktur morfologinya yang kaya dengan imbuhan. IndoBERT misalnya, merupakan model berbasis Transformer yang dilatih menggunakan korpus besar berbahasa Indonesia, sehingga kinerjanya sangat baik dalam tugas-tugas seperti klasifikasi teks, ekstraksi entitas, dan analisis sentimen. Sementara itu, IndoBERTweet dilatih dengan data dari media sosial Twitter, sehingga sangat relevan untuk menganalisis teks informal yang penuh singkatan, emotikon, dan gaya bahasa gaul. Kehadiran model-model ini sangat penting karena sebagian besar pustaka NLP

global seperti spaCy atau Hugging Face awalnya dikembangkan terutama untuk bahasa Inggris.

Salah satu platform yang tidak bisa dilewatkan adalah Hugging Face Transformers. Framework ini menyediakan akses mudah ke ratusan model pra-latih berbasis Transformer seperti BERT, GPT, RoBERTa, dan T5. Dengan hanya beberapa baris kode, peneliti dapat melakukan tokenisasi, sequence classification, NER, question answering, hingga summarization. Hugging Face juga menyediakan model hub, yaitu repositori online tempat ribuan model pra-latih dapat diunduh dan digunakan. Keunggulan utama framework ini adalah fleksibilitas dan kekuatannya dalam menangani tugas-tugas NLP modern, meskipun untuk penggunaannya diperlukan sumber daya komputasi yang lebih besar dibandingkan pustaka NLP tradisional.

Contoh sederhana implementasi dapat ditunjukkan dengan spaCy. Misalnya, untuk mengenali entitas dalam kalimat "President Joko Widodo inaugurated Napun Gete Dam in East Nusa Tenggara on 23 February 2022," pengguna dapat menjalankan kode berikut:

```
import spacy
nlp = spacy.load("en_core_web_sm")

doc = nlp("President Joko Widodo inaugurated Napun Gete Dam in East Nusa
Tenggara on 23 February 2022.")
for ent in doc.ents:
    print(ent.text, ent.label_)
```

Hasil dari kode tersebut akan menampilkan entitas seperti Joko Widodo (PERSON), Napun Gete Dam (FACILITY), East Nusa Tenggara (GPE/Geopolitical Entity), dan 23 February 2022 (DATE). Dengan hanya beberapa baris kode, sistem dapat mengidentifikasi

informasi penting yang sebelumnya tersembunyi dalam teks.

Dari paparan ini dapat dipahami bahwa tools dan library NLP memiliki peran krusial dalam mempercepat penelitian maupun aplikasi praktis. NLTK sangat berguna untuk pembelajaran dasar, spaCy untuk aplikasi industri, Stanza untuk analisis canggih berbasis deep learning, sementara IndoBERT dan variannya memperkuat dukungan untuk Bahasa Indonesia. Hugging Face Transformers memberikan akses ke ekosistem global model pra-latih yang sangat kuat. Pemilihan library yang tepat sangat bergantung pada tujuan, bahasa yang digunakan, serta skala aplikasi yang dikembangkan. Bagi mahasiswa, mengenal berbagai tools ini tidak hanya penting dari sisi teknis, tetapi juga memberi wawasan praktis tentang bagaimana konsep NLP diwujudkan dalam perangkat nyata.

E. Aplikasi NLP dalam EI

Untuk memahami lebih jelas peran Natural Language Processing (NLP) dalam mendukung ekstraksi informasi (Information Extraction/IE), kita perlu melihat bagaimana konsep, metode, dan tools NLP diterapkan dalam kasus nyata. Studi kasus ini akan menunjukkan bagaimana teks mentah yang awalnya tidak terstruktur dapat diubah menjadi data terorganisasi yang siap digunakan untuk analisis maupun pengambilan keputusan.

Salah satu contoh penting adalah analisis berita bencana alam. Dalam situasi darurat seperti gempa bumi, banjir, atau tanah longsor, informasi yang cepat dan akurat sangat dibutuhkan oleh pemerintah, lembaga penanggulangan bencana, dan masyarakat luas. Sumber

utama informasi biasanya berasal dari teks berita daring, laporan resmi, atau unggahan media sosial. Namun, teks-teks ini tidak terstruktur, panjang, dan sering kali ditulis dengan gaya narasi. Oleh karena itu, dibutuhkan pipeline NLP yang mampu menyaring dan mengekstrak informasi inti.

Sebagai ilustrasi, perhatikan potongan berita berikut:

"Gempa bumi berkekuatan 6,2 SR mengguncang Kabupaten Cianjur pada Senin pagi. Akibat kejadian ini, sebanyak 162 orang meninggal dunia dan ratusan lainnya luka-luka."

Proses pertama yang dilakukan adalah text preprocessing. Teks berita dibersihkan dari tanda baca, simbol, atau elemen yang tidak relevan. Kalimat kemudian diproses dengan tokenisasi untuk memecahnya menjadi unit kata, dilanjutkan dengan Part-of-Speech (POS) tagging untuk mengenali kelas kata seperti nomina, verba, atau numeralia. Tahap berikutnya adalah Named Entity Recognition (NER) yang bertugas mengidentifikasi entitas penting. Dari teks di atas, sistem akan mengenali "6,2 SR" sebagai magnitudo bencana, "Kabupaten Cianjur" sebagai lokasi, "Senin pagi" sebagai waktu, serta "162 orang" sebagai jumlah korban.

Hasil dari tahap NLP tersebut kemudian dimanfaatkan oleh modul Information Extraction untuk menyusun data dalam format terstruktur. Misalnya, IE dapat menghasilkan tabel seperti berikut:

Entitas	**Kategori**	**Nilai Ekstraksi**
Gempa bumi	Jenis Bencana	6,2 SR
Kabupaten Cianjur	Lokasi	Jawa Barat, Indonesia
162 orang meninggal	Korban Jiwa	Fatalities
Senin pagi	Waktu	21 November 2022

Dengan tabel ini, lembaga penanggulangan bencana dapat segera mengetahui fakta inti: jenis bencana, lokasi, jumlah korban, dan waktu kejadian, tanpa harus membaca narasi panjang. Informasi yang telah diekstraksi juga bisa dipetakan ke dalam knowledge graph untuk memvisualisasikan hubungan antar entitas, misalnya menghubungkan "Gempa bumi" dengan "Cianjur," "162 orang," dan "21 November 2022."

Selain pada berita bencana, NLP juga banyak digunakan untuk ekstraksi informasi di bidang kesehatan. Rekam medis elektronik sering kali ditulis dalam bentuk catatan naratif oleh dokter. Dengan bantuan NLP, sistem dapat mengekstrak informasi penting seperti diagnosis, nama obat, dosis, dan hasil laboratorium. Misalnya, dari catatan "Pasien didiagnosis pneumonia dan diberi amoksisilin 500 mg tiga kali sehari," NLP akan mengenali "pneumonia" sebagai diagnosis, "amoksisilin" sebagai obat, "500 mg" sebagai dosis, dan "tiga kali sehari" sebagai frekuensi konsumsi. Informasi ini dapat membantu rumah sakit dalam mengelola data pasien, penelitian klinis, maupun pengambilan keputusan medis.

Dalam sektor e-commerce, NLP dan IE digunakan untuk menganalisis ulasan produk dari pelanggan. Sistem dapat mengidentifikasi produk, fitur yang disebutkan, serta sentimen yang terkandung. Sebagai contoh, dari ulasan "Kamera smartphone ini bagus, tetapi baterainya cepat habis," sistem akan mengekstrak bahwa produk adalah "smartphone," fitur positif adalah "kamera," sedangkan fitur negatif adalah "baterai." Data ini dapat diolah lebih lanjut untuk memberikan masukan kepada produsen atau menyusun rekomendasi produk kepada calon pembeli.

Studi kasus lain terdapat di bidang hukum dan regulasi, di mana NLP digunakan untuk mengekstrak entitas hukum dari dokumen panjang seperti kontrak, peraturan, atau putusan pengadilan. Misalnya, dalam sebuah kontrak bisnis, sistem IE dapat mengenali nama pihak-pihak yang terlibat, tanggal berlaku, jenis perjanjian, serta kewajiban masing-masing pihak. Hal ini sangat membantu firma hukum atau regulator dalam menelusuri ribuan dokumen secara cepat tanpa harus membaca manual satu per satu.

Dari berbagai studi kasus tersebut, dapat disimpulkan bahwa NLP berperan sebagai fondasi analisis teks, sedangkan IE berfungsi menghasilkan data yang siap digunakan. Tanpa NLP, proses ekstraksi informasi akan kesulitan mengenali entitas, relasi, dan konteks dalam teks. Sebaliknya, tanpa IE, hasil analisis NLP hanya akan berhenti pada label linguistik yang sulit dimanfaatkan lebih lanjut. Dengan sinergi keduanya, teks naratif dapat diubah menjadi data yang ringkas, akurat, dan kontekstual.

Studi kasus aplikasi NLP dalam IE ini memberikan gambaran nyata tentang bagaimana teknologi bahasa dapat mendukung berbagai sektor. Bagi mahasiswa, pemahaman ini penting agar dapat melihat keterkaitan antara teori yang dipelajari dengan praktik di dunia nyata. Lebih jauh, kemampuan untuk membangun sistem IE berbasis NLP membuka peluang besar di bidang penelitian, industri, dan layanan publik, terutama di era big data yang semakin menuntut otomatisasi pengolahan informasi.

Latihan

A. Pertanyaan Pemahaman Konsep

1. Apa yang dimaksud dengan *Relation Extraction* (RE)?
2. Jelaskan perbedaan utama antara NER dan RE dalam konteks sistem IE.
3. Mengapa proses RE selalu dilakukan setelah NER dalam pipeline IE?
4. Sebutkan dan jelaskan tiga jenis relasi umum yang dapat diekstraksi dari teks.
5. Jelaskan bagaimana parsing sintaksis membantu dalam proses RE.
6. Apa tantangan terbesar dalam mengekstraksi relasi pada teks Bahasa Indonesia?
7. Mengapa pendekatan berbasis *supervised learning* memerlukan data berlabel untuk RE?

B. Latihan Praktik Sederhana

Diberikan kalimat berikut:

"Universitas Indonesia bekerja sama dengan Kementerian Kesehatan dalam penelitian vaksin Covid-19."

Tugas:

1. Identifikasi entitas bernama dalam kalimat tersebut (*Organization*, *Organization*, *Event*).
2. Tentukan relasi yang menghubungkan entitas-entitas tersebut (misal: *kerja sama dalam penelitian*).
3. Sajikan hasil ekstraksi dalam format tabel berikut:

Entitas 1	Jenis Entitas 1	Entitas 2	Jenis Entitas 2	Jenis Relasi	Deskripsi
Universitas Indonesia	Organization	Kementerian Kesehatan	Organization	Collaboration	Kerja sama dalam penelitian

Kementerian Kesehatan	Organization	vaksin Covid-19	Event	Research Focus	Fokus penelitian vaksin

4. Jelaskan bagaimana struktur kalimat membantu sistem mengenali relasi "kerja sama".
5. Identifikasi potensi ambiguitas jika konteks kalimat diubah.

C. Studi Kasus / Proyek Mini

Anda diminta merancang modul *Relation Extraction* untuk sistem analisis berita.

1. Tentukan jenis relasi yang ingin Anda ekstrak (misal: *Person–Organization*, *Event–Location*, *Product–Company*).
2. Jelaskan alur sistem mulai dari input teks, hasil NER, hingga ekstraksi relasi.
3. Buat tiga contoh kalimat beserta hasil RE-nya.
4. Diskusikan kesulitan yang muncul jika satu kalimat memiliki lebih dari satu relasi.
5. Jelaskan kelebihan pendekatan *neural RE* dibandingkan *pattern-based RE*.

D. Diskusi / Refleksi

1. Menurut Anda, mengapa memahami struktur sintaksis sangat penting dalam RE?
2. Bagaimana RE dapat dimanfaatkan dalam bidang intelijen, kesehatan, atau bisnis?
3. Apakah mungkin membangun sistem RE tanpa proses NER terlebih dahulu? Jelaskan alasannya.
4. Diskusikan potensi penerapan RE dalam Bahasa Indonesia dan kendala yang mungkin dihadapi dalam pembuatan dataset.

BAB 7
ATURAN PRODUKSI DAN KOMPONEN PARSING

Tujuan Pembelajaran

Setelah mempelajari Bab 7, mahasiswa diharapkan mampu:

1. Menjelaskan konsep dasar Event Extraction (EE) dalam kerangka kerja *Information Extraction (IE)*.
2. Membedakan antara ekstraksi entitas, relasi, dan peristiwa, serta memahami keterkaitannya dalam representasi pengetahuan.
3. Mengidentifikasi elemen utama dalam sebuah peristiwa (event elements): *trigger*, *participant/argument*, *location*, *time*, dan *outcome*.
4. Menjelaskan tahapan umum dalam ekstraksi peristiwa, mulai dari deteksi *event trigger* hingga klasifikasi peristiwa.
5. Membedakan pendekatan berbasis aturan, statistik, dan *deep learning* dalam ekstraksi peristiwa.
6. Menerapkan contoh sederhana ekstraksi peristiwa dari kalimat berbahasa Indonesia.
7. Menganalisis tantangan dan kesalahan umum dalam identifikasi peristiwa multisentensial (across-sentence event extraction).

Setiap bahasa, baik bahasa alami maupun bahasa formal, memiliki aturan tertentu yang mengatur bagaimana simbol, kata, atau kalimat dapat disusun dengan benar. Dalam konteks komputasi, pemahaman terhadap aturan ini menjadi kunci agar komputer mampu membaca, menganalisis, dan menafsirkan teks secara sistematis. Pada bab sebelumnya, kita telah mempelajari bagaimana Natural

Language Processing (NLP) menyediakan dasar linguistik untuk mengenali kata, struktur kalimat, hingga entitas. Namun, agar analisis dapat lebih formal dan presisi, diperlukan pemahaman mengenai aturan produksi dalam bahasa formal serta bagaimana aturan tersebut diimplementasikan dalam proses parsing.

Parsing, dalam ilmu komputer, adalah proses untuk menganalisis susunan simbol dalam teks agar sesuai dengan tata aturan tertentu. Dalam NLP, parsing membantu sistem untuk memahami struktur kalimat, menentukan hubungan antar kata, dan menghubungkan teks dengan makna. Proses parsing ini biasanya melibatkan beberapa komponen penting, seperti scanner, parser, dan translator, yang bekerja sama untuk mengubah input teks menjadi bentuk representasi yang lebih terstruktur.

Bab ini akan membahas konsep dasar aturan produksi dalam bahasa formal, komponen-komponen parsing, peran evaluator dalam transformasi data, hingga penerapannya dalam ekstraksi informasi. Sebagai penutup, akan diberikan studi kasus sederhana untuk memperlihatkan bagaimana parsing digunakan dalam sistem EI.

A. Pengertian Aturan Produksi dalam Bahasa Formal

Dalam teori bahasa formal, aturan produksi (production rules) adalah seperangkat kaidah yang mendefinisikan bagaimana suatu simbol atau himpunan simbol dapat diganti dengan simbol lain untuk membentuk string atau kalimat yang valid. Aturan produksi merupakan inti dari konsep tata bahasa formal (formal grammar), yang banyak digunakan dalam ilmu komputer, baik pada pengembangan bahasa pemrograman, kompilasi, maupun pemrosesan bahasa alami.

Secara umum, aturan produksi dituliskan dalam bentuk pasangan:

$$A \rightarrow \alpha$$

di mana A adalah sebuah simbol non-terminal, sedangkan α adalah urutan simbol terminal maupun non-terminal. Simbol non-terminal berfungsi sebagai variabel yang dapat diperluas menjadi bentuk lain, sementara terminal adalah simbol akhir yang tidak dapat dipecah lebih lanjut, biasanya berupa kata atau karakter aktual dalam bahasa.

Sebagai contoh, dalam bahasa formal sederhana kita dapat menuliskan aturan:

```
S → NP VP
NP → Det N
VP → V NP
Det → "the" | "a"
N → "cat" | "dog"
V → "chased" | "saw"
```

Dari aturan tersebut, kalimat "the cat chased a dog" dapat dihasilkan secara sistematis. Aturan inilah yang membuat komputer mampu "menghasilkan" atau "memverifikasi" suatu kalimat berdasarkan gramatika tertentu.

Dalam konteks bahasa alami seperti Bahasa Indonesia, aturan produksi dapat digunakan untuk merepresentasikan struktur kalimat. Misalnya, kalimat "Petugas BNPB mengevakuasi korban banjir" dapat dijelaskan dengan aturan produksi sebagai berikut:

```
S → NP VP
NP → NNP (Petugas BNPB)
VP → V NP
V → "mengevakuasi"
NP → N N (korban banjir)
```

Struktur ini menunjukkan bahwa kalimat terdiri atas Subjek (NP), Predikat (V), dan Objek (NP). Dengan representasi aturan produksi, sistem dapat memahami

bahwa “Petugas BNPB” berperan sebagai subjek atau pelaku, “mengevakuasi” adalah aksi, dan “korban banjir” merupakan objek tindakan.

Aturan produksi memiliki kedekatan erat dengan konsep Context-Free Grammar (CFG). CFG adalah salah satu jenis tata bahasa formal di mana setiap aturan produksi memiliki satu simbol non-terminal di sisi kiri dan deretan terminal maupun non-terminal di sisi kanan. CFG banyak digunakan dalam pemrosesan bahasa alami karena cukup mampu merepresentasikan struktur kalimat sederhana dalam bahasa manusia. Namun demikian, bahasa alami jauh lebih kompleks dibandingkan bahasa pemrograman, sehingga penerapan CFG pada NLP memerlukan penyesuaian dan pengayaan aturan.

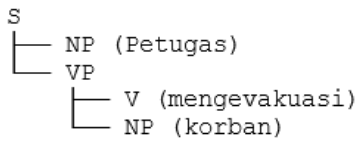
Mengapa aturan produksi penting dalam ekstraksi informasi? Hal ini karena informasi yang ingin diambil dari teks tidak hanya ditentukan oleh keberadaan kata, tetapi juga oleh hubungan struktural antar kata. Tanpa memahami struktur kalimat, sistem mungkin salah menafsirkan siapa pelaku dan siapa objek dalam sebuah peristiwa. Sebagai contoh, perhatikan kalimat berikut:

1. “Relawan menolong korban banjir.”
2. “Korban banjir ditolong relawan.”

Kedua kalimat tersebut mengandung kata yang sama, namun struktur kalimat berbeda. Pada kalimat pertama, subjek adalah “relawan” dan objek adalah “korban banjir.” Sementara pada kalimat kedua, subjek secara gramatikal adalah “korban banjir,” namun pelaku sebenarnya tetap “relawan.” Tanpa aturan produksi dan parsing yang tepat, sistem ekstraksi informasi dapat salah menafsirkan pelaku dan korban.

Dalam praktiknya, aturan produksi sering

digambarkan dalam bentuk parse tree atau pohon sintaksis. Parse tree memvisualisasikan bagaimana aturan produksi diaplikasikan secara bertahap untuk menghasilkan kalimat. Misalnya, kalimat "Petugas mengevakuasi korban" dapat diturunkan dari aturan produksi berikut:



Pohon sintaksis ini menunjukkan bahwa kalimat terdiri dari subjek (NP), predikat (V), dan objek (NP). Representasi semacam ini sangat berguna dalam ekstraksi informasi karena memberikan gambaran hierarkis yang jelas mengenai peran setiap kata.

Selain itu, aturan produksi juga dapat digunakan dalam domain khusus. Misalnya, dalam dokumen medis dapat dibuat aturan:

```
S → Patient Diagnosis Treatment
Patient → "Pasien" NNP
Diagnosis → "didiagnosis" Penyakit
Treatment → "diberikan" Obat Dosis
```

Dengan aturan tersebut, kalimat "Pasien Andi didiagnosis pneumonia dan diberikan amoksisilin 500 mg" dapat dianalisis dan diekstrak menjadi data terstruktur: pasien = Andi, diagnosis = pneumonia, pengobatan = amoksisilin 500 mg.

Dari uraian ini jelas bahwa aturan produksi bukan sekadar teori linguistik atau komputasi, tetapi juga merupakan fondasi penting untuk membangun sistem analisis teks. Dalam konteks buku ini, pemahaman mengenai aturan produksi sangat diperlukan karena ia menjadi dasar dari proses parsing, yang pada gilirannya mendukung ekstraksi informasi dari teks secara lebih

akurat dan kontekstual.

B. Komponen Parsing: Scanner, Parser, Translator

Parsing adalah proses penting dalam analisis bahasa alami maupun bahasa formal. Melalui parsing, sebuah kalimat atau ekspresi dianalisis untuk melihat apakah ia sesuai dengan aturan tata bahasa yang berlaku, serta untuk menghasilkan representasi struktural yang memudahkan komputer memahami hubungan antar unsur. Namun, parsing tidak dilakukan oleh satu komponen tunggal. Dalam implementasinya, parsing melibatkan beberapa komponen utama yang bekerja secara berurutan, yaitu scanner, parser, dan translator. Ketiga komponen ini memiliki peran yang berbeda namun saling melengkapi dalam membangun pemahaman komputer terhadap teks.

1. Scanner (Lexical Analyzer)

   Scanner, atau disebut juga lexical analyzer, adalah tahap awal dalam proses parsing. Tugas utama scanner adalah membaca input teks mentah dan memecahnya menjadi unit-unit dasar yang disebut token. Token merupakan representasi terkecil yang masih memiliki makna linguistik atau fungsional, seperti kata, tanda baca, atau simbol tertentu. Dalam bahasa pemrograman, token biasanya berupa kata kunci (keyword), identifier, operator, atau literal. Dalam bahasa alami, token bisa berupa kata benda, kata kerja, angka, atau tanda hubung.

   Sebagai contoh, kalimat:

   "Petugas BNPB mengevakuasi korban banjir di Cianjur."

   akan diubah oleh scanner menjadi token-token berikut:

   - Petugas / NN
   - BNPB / NNP
   - mengevakuasi / VB

- korban / NN
- banjir / NN
- di / IN
- Cianjur / NNP

Dengan adanya scanner, sistem dapat menyaring teks mentah yang penuh variasi menjadi unit dasar yang lebih rapi dan konsisten. Tanpa tahap ini, parser akan kesulitan menganalisis struktur kalimat karena input yang diterima masih terlalu bebas.

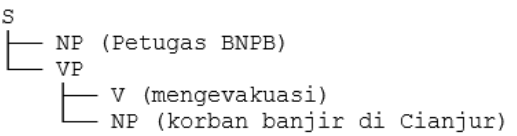
2. Parser (Syntactic Analyzer)

Setelah scanner menghasilkan token, tahap berikutnya adalah parser. Parser bertugas menyusun token-token tersebut ke dalam struktur yang lebih kompleks sesuai dengan aturan produksi tata bahasa formal. Representasi yang dihasilkan biasanya berupa parse tree atau pohon sintaksis, yang menunjukkan bagaimana suatu kalimat terbentuk dari aturan gramatikal.

Parser memiliki dua peran utama:

a. Memverifikasi apakah urutan token sesuai dengan tata bahasa yang berlaku (apakah kalimat tersebut valid secara sintaktis).
b. Membangun struktur sintaksis yang menggambarkan hubungan antar token, seperti subjek, predikat, objek, dan keterangan.

Misalnya, parser dapat menghasilkan struktur seperti ini untuk kalimat sebelumnya:



Dengan struktur ini, sistem tidak hanya mengetahui kata-kata dalam kalimat, tetapi juga memahami bahwa "Petugas BNPB" adalah subjek, "mengevakuasi" adalah predikat, dan "korban banjir di Cianjur" adalah objek sekaligus keterangan tempat. Informasi inilah yang sangat penting bagi sistem ekstraksi informasi untuk membangun relasi antar entitas.

3. Translator

Tahap terakhir dalam komponen parsing adalah translator. Jika parser menghasilkan struktur sintaksis, maka translator bertugas mengubah struktur tersebut ke dalam bentuk representasi yang lebih bermakna dan berguna untuk aplikasi lebih lanjut. Representasi ini bisa berupa struktur semantik, basis data, atau format lain yang dapat dimanfaatkan oleh sistem ekstraksi informasi.

Sebagai contoh, translator dapat mengubah parse tree dari kalimat "Petugas BNPB mengevakuasi korban banjir di Cianjur" menjadi representasi semantik seperti berikut:

- Aksi: mengevakuasi
- Pelaku: Petugas BNPB
- Objek: korban banjir
- Lokasi: Cianjur

Dengan representasi ini, informasi tidak lagi hanya berupa struktur linguistik, melainkan data yang siap digunakan untuk analisis, visualisasi, atau sistem tanggap darurat. Translator juga dapat diprogram untuk melakukan normalisasi, misalnya mengubah "di Cianjur" menjadi koordinat geografis tertentu, atau menyamakan istilah "BNPB" dengan nama organisasi resminya "Badan Nasional Penanggulangan Bencana."

Sinergi Scanner, Parser, dan Translator

Ketiga komponen ini bekerja sebagai satu kesatuan dalam pipeline parsing. Scanner bertugas membersihkan dan memecah teks mentah menjadi token. Parser menyusun token-token tersebut sesuai aturan tata bahasa untuk membentuk struktur kalimat. Translator kemudian menerjemahkan struktur itu ke dalam format semantik atau data yang lebih aplikatif. Tanpa scanner, parser akan kesulitan menerima input yang konsisten. Tanpa parser, translator tidak memiliki struktur untuk diubah menjadi data. Dan tanpa translator, hasil parsing hanya berupa pohon sintaksis yang sulit dimanfaatkan secara langsung.

Dalam konteks ekstraksi informasi, sinergi ini sangat krusial. Scanner memastikan bahwa kata-kata diproses dengan benar, parser memberikan kerangka hubungan antar kata, sementara translator menghasilkan representasi informasi yang sesuai kebutuhan pengguna. Misalnya, dalam sistem deteksi bencana berbasis teks, scanner akan memproses berita, parser akan mengenali struktur kalimat, dan translator akan mengekstrak data penting seperti jenis bencana, lokasi, waktu, dan jumlah korban.

C. Evaluator dan Proses Transformasi Data

Parsing tidak berhenti hanya pada tahap pembentukan struktur sintaksis. Hasil parse tree atau representasi aturan produksi yang dihasilkan parser baru menggambarkan bagaimana sebuah kalimat tersusun secara formal menurut tata bahasa. Namun, dalam banyak aplikasi, terutama pada ekstraksi informasi, kita tidak hanya membutuhkan struktur, melainkan juga makna atau informasi semantik yang terkandung di dalam teks. Di sinilah peran evaluator menjadi sangat penting.

Evaluator adalah komponen dalam sistem parsing yang bertugas mengubah hasil analisis sintaksis menjadi representasi semantik atau bentuk data yang lebih aplikatif. Proses ini sering disebut transformasi data, yaitu proses konversi dari struktur linguistik menjadi struktur informasi yang dapat dimanfaatkan oleh aplikasi lebih lanjut, seperti basis data, sistem pelaporan, atau knowledge graph.

Ada beberapa fungsi penting dari evaluator:

1. Interpretasi Semantik

   Evaluator menambahkan dimensi makna pada struktur sintaksis. Sebagai contoh, parser mungkin hanya menghasilkan struktur:

   NP (Petugas BNPB) + VP (mengevakuasi NP (korban banjir))

   Evaluator kemudian menafsirkannya sebagai:

   - Pelaku: Petugas BNPB
   - Aksi: mengevakuasi
   - Objek: korban banjir

   Dengan demikian, kalimat yang awalnya hanya struktur gramatikal kini menjadi data semantik yang lebih bermakna.

2. Transformasi ke Format Terstruktur

Evaluator mengubah representasi kalimat ke dalam format yang sesuai kebutuhan sistem, misalnya JSON, tabel basis data, atau RDF triple dalam knowledge graph. Contoh representasi dalam JSON:

```
{
  "aksi": "mengevakuasi",
  "pelaku": "Petugas BNPB",
  "objek": "korban banjir",
  "lokasi": "Cianjur"
}
```

Representasi semacam ini memudahkan integrasi dengan aplikasi lain, misalnya sistem pemantauan bencana.

3. Normalisasi Informasi

Evaluator juga berfungsi untuk menormalkan data. Misalnya, dari teks "di Cianjur," evaluator dapat mengubahnya menjadi "Kabupaten Cianjur, Jawa Barat, Indonesia" dengan merujuk ke basis data geografis. Demikian pula tanggal "Senin pagi" bisa dinormalisasi menjadi "21 November 2022" jika konteks waktu berita diketahui.

4. Seleksi Informasi Relevan

Tidak semua informasi dalam kalimat perlu diekstrak. Evaluator dapat diprogram untuk hanya mengambil entitas tertentu sesuai kebutuhan. Misalnya, dalam sistem epidemiologi, evaluator mungkin hanya mengekstrak "jumlah pasien," "nama penyakit," dan "lokasi," sementara informasi lain diabaikan.

Proses evaluasi biasanya dilakukan dalam dua tahap:

1. Evaluasi Sintaksis ke Semantik

   Tahap ini menghubungkan hasil parser dengan pengetahuan semantik. Misalnya, kata kerja "meresmikan" dalam teks berita dapat dipetakan sebagai aksi inauguration event, sehingga sistem mengetahui bahwa entitas yang mengikuti kata ini adalah objek yang diresmikan.

2. Evaluasi Semantik ke Data Terstruktur

   Tahap ini mengubah hasil evaluasi semantik ke dalam struktur yang konsisten, misalnya database relasional, dokumen XML/JSON, atau knowledge base. Tahap inilah yang memungkinkan data digunakan lebih lanjut untuk analisis, pencarian, atau visualisasi.

Mari kita lihat contoh teks berita berikut:

"Presiden Joko Widodo meresmikan Bendungan Napun Gete di Nusa Tenggara Timur pada 23 Februari 2022."

- Scanner → memecah teks menjadi token.
- Parser → menyusun token menjadi struktur kalimat:
- S → NP (Presiden Joko Widodo) + VP (meresmikan NP (Bendungan Napun Gete) PP (di NTT) PP (pada 23 Februari 2022))
- Evaluator → mengubah hasil parsing menjadi representasi semantik:

Aksi	Pelaku	Objek	Lokasi	Waktu
Meresmikan	Joko Widodo	Bendungan Napun Gete	Nusa Tenggara Timur	23 Februari 2022

- Representasi ini jauh lebih berguna untuk sistem ekstraksi informasi, misalnya untuk basis data

pembangunan infrastruktur atau laporan pemerintah.

Dalam konteks ekstraksi informasi, evaluator memiliki peran sebagai jembatan terakhir antara analisis linguistik dan aplikasi nyata. Tanpa evaluator, sistem hanya menghasilkan pohon sintaksis atau anotasi linguistik yang sulit dipahami pengguna. Dengan evaluator, hasil analisis menjadi data siap pakai yang langsung dapat dimanfaatkan untuk berbagai tujuan, seperti:

- Pemantauan bencana: mengubah berita menjadi data jumlah korban, lokasi, dan waktu.
- Analisis medis: mengekstrak diagnosis, obat, dan dosis dari rekam medis.
- Analisis hukum: mengubah dokumen kontrak menjadi struktur pihak terlibat, hak, kewajiban, dan tanggal berlaku.

Evaluator juga memungkinkan adanya proses filtering untuk menghindari informasi redundan atau tidak relevan. Hal ini penting terutama dalam sistem yang harus menangani teks dalam jumlah besar, misalnya ribuan artikel berita setiap hari.

Evaluator adalah tahap krusial dalam proses parsing yang bertugas menghubungkan hasil analisis sintaksis dengan representasi semantik yang bermakna. Melalui evaluator, hasil parsing tidak hanya menjawab "apakah kalimat ini valid secara tata bahasa," tetapi juga "informasi apa yang terkandung dalam kalimat tersebut." Evaluator memungkinkan proses transformasi data dari teks mentah menjadi data terstruktur yang siap digunakan dalam sistem ekstraksi informasi maupun aplikasi praktis lainnya.

Dengan pemahaman ini, mahasiswa tidak hanya dapat memahami parsing secara teoritis, tetapi juga dapat melihat bagaimana parsing berkontribusi pada tujuan yang lebih luas, yaitu menyusun informasi dari teks untuk mendukung analisis, pencarian, dan pengambilan keputusan.

D. Penerapan Parsing untuk Ekstraksi Informasi

Parsing dalam konteks pemrosesan bahasa alami tidak hanya berhenti pada tahap analisis struktur kalimat. Lebih dari itu, parsing menjadi fondasi bagi berbagai aplikasi tingkat lanjut, salah satunya adalah ekstraksi informasi (Information Extraction/IE). Seperti yang telah dijelaskan sebelumnya, tujuan IE adalah mengidentifikasi entitas, relasi, dan peristiwa dari teks tidak terstruktur, lalu menyusunnya kembali dalam bentuk data yang lebih terorganisasi. Agar IE dapat bekerja secara akurat, sistem membutuhkan pemahaman yang benar mengenai struktur kalimat, hubungan antar kata, serta fungsi gramatikal dalam teks. Proses parsinglah yang memberikan kerangka formal untuk memenuhi kebutuhan ini.

Parsing memungkinkan sistem untuk menghubungkan sintaksis dengan semantik. Sintaksis menunjukkan bagaimana kata-kata disusun, sedangkan semantik menekankan makna yang terkandung dalam struktur tersebut. Sebagai contoh, perhatikan dua kalimat berikut:

1. “Tim SAR mengevakuasi korban banjir.”
2. “Korban banjir dievakuasi oleh Tim SAR.”

Secara semantik, kedua kalimat menyampaikan informasi yang sama: ada peristiwa evakuasi, pelaku adalah Tim SAR, dan objeknya adalah korban banjir.

Namun, secara sintaksis, struktur kedua kalimat berbeda. Parsing memungkinkan sistem memahami bahwa meskipun urutan kata berbeda, relasi semantiknya tetap sama. Tanpa parsing, sistem IE mungkin salah menafsirkan "korban banjir" sebagai pelaku dalam kalimat kedua karena ia muncul di posisi subjek.

Dengan parsing, sistem dapat memastikan bahwa "Tim SAR" berfungsi sebagai pelaku (agent) dan "korban banjir" sebagai objek (patient), terlepas dari variasi susunan kalimat. Ini menunjukkan bagaimana parsing memberikan keakuratan lebih tinggi pada IE dibandingkan pendekatan sederhana berbasis pencocokan kata.

Parsing sangat berguna ketika diterapkan pada analisis berita, terutama dalam domain kebencanaan. Misalnya, dalam berita berikut:

"Gempa berkekuatan 6,2 SR mengguncang Kabupaten Cianjur pada Senin pagi."

Melalui parsing, sistem dapat menyusun struktur kalimat sebagai berikut:

```
S → NP (Gempa berkekuatan 6,2 SR) + VP (mengguncang NP (Kabupaten
Cianjur) PP (pada Senin pagi))
```

Struktur ini kemudian memudahkan IE untuk menghasilkan data terorganisasi:

- Peristiwa: Gempa bumi
- Magnitudo: 6,2 SR
- Lokasi: Kabupaten Cianjur
- Waktu: Senin pagi

Parsing membantu IE dengan dua cara: pertama, memastikan bahwa entitas seperti "Kabupaten Cianjur" dikenali sebagai lokasi, bukan sekadar kata benda umum.

Kedua, menghubungkan entitas tersebut ke dalam struktur semantik yang lebih luas, yakni "lokasi kejadian bencana."

Parsing juga memiliki peran penting dalam ekstraksi informasi dari dokumen medis. Pertimbangkan kalimat:

"Pasien didiagnosis pneumonia dan diberi amoksisilin 500 mg tiga kali sehari."

Tanpa parsing, sistem hanya akan melihat kata "pneumonia," "amoksisilin," dan "500 mg" sebagai token-token terpisah. Namun parsing memungkinkan sistem mengenali bahwa:

- "Pasien" berfungsi sebagai entitas utama (subjek).
- "didiagnosis pneumonia" adalah diagnosis medis.
- "diberi amoksisilin 500 mg tiga kali sehari" adalah tindakan terapi, dengan detail obat, dosis, dan frekuensi.

Dengan pemahaman struktur ini, IE dapat mengubah catatan medis naratif menjadi data terstruktur:

Entitas	Kategori	Nilai
Pasien	Subjek	[nama pasien]
Diagnosis	Penyakit	Pneumonia
Obat	Terapi	Amoksisilin 500 mg
Frekuensi	Terapi	3 kali sehari

Parsing memastikan hubungan antar komponen medis tidak hilang, sehingga hasil ekstraksi lebih akurat dan bermanfaat.

Penerapan dalam Dokumen Hukum

Dalam dokumen hukum, parsing membantu mengidentifikasi subjek hukum, objek perjanjian, dan

klausul kewajiban. Misalnya, kalimat:

"Pihak Pertama berkewajiban menyerahkan tanah kepada Pihak Kedua paling lambat 30 hari setelah perjanjian ditandatangani."

Parsing menghasilkan struktur:

- Subjek: Pihak Pertama
- Predikat: berkewajiban menyerahkan
- Objek: tanah
- Keterangan penerima: Pihak Kedua
- Keterangan waktu: paling lambat 30 hari setelah perjanjian

Dari struktur ini, sistem IE dapat langsung menyusun entri data untuk basis kontrak hukum, sehingga memudahkan pencarian maupun analisis kewajiban antar pihak.

Dari berbagai contoh tersebut, jelas bahwa parsing berperan sebagai penghubung penting antara teks tidak terstruktur dengan informasi terorganisasi. Parsing memastikan bahwa IE tidak hanya sekadar mengenali kata, tetapi juga memahami siapa melakukan apa, kepada siapa, di mana, dan kapan. Hal ini sangat relevan untuk berbagai aplikasi mulai dari analisis bencana, rekam medis, kontrak hukum, hingga media sosial.

Tanpa parsing, sistem ekstraksi informasi cenderung dangkal karena hanya mengandalkan kata kunci. Dengan parsing, sistem memperoleh pemahaman struktural dan semantik yang memungkinkan hasil ekstraksi lebih presisi, konsisten, dan bermakna. Oleh karena itu, pemahaman dan penerapan parsing merupakan keterampilan fundamental bagi mahasiswa maupun praktisi yang ingin mengembangkan sistem ekstraksi informasi berbasis NLP.

E. Parsing Sederhana untuk Ekstraksi Informasi (EI)

Untuk memahami lebih jelas peran parsing dalam ekstraksi informasi, mari kita lihat sebuah studi kasus sederhana yang diambil dari teks berita bencana. Misalnya, pada kalimat "Tim SAR berhasil mengevakuasi lima korban tanah longsor di Desa Cijedil, Kabupaten Cianjur, pada Senin pagi." Kalimat ini tampak sederhana, namun sebenarnya mengandung berbagai informasi penting yang dapat diubah menjadi data terstruktur apabila diproses dengan benar. Informasi tersebut antara lain mencakup siapa pelaku tindakan, apa aksinya, siapa objeknya, serta di mana dan kapan peristiwa terjadi.

Proses parsing diawali dengan tahap scanning atau tokenisasi. Pada tahap ini, kalimat dipecah menjadi unit-unit kata atau token, seperti "Tim," "SAR," "berhasil," "mengevakuasi," "lima," "korban," "tanah," "longsor," "di," "Desa," "Cijedil," "Kabupaten," "Cianjur," "pada," "Senin," dan "pagi." Setiap token kemudian dilabeli kelas katanya, misalnya "Tim" sebagai kata benda, "mengevakuasi" sebagai kata kerja, dan "Cianjur" sebagai kata benda khusus (nama tempat). Dengan adanya tokenisasi, teks mentah yang awalnya panjang menjadi lebih terstruktur dan siap dianalisis lebih lanjut.

Tahap berikutnya adalah parsing, yaitu menyusun token-token tersebut ke dalam struktur kalimat sesuai dengan aturan produksi bahasa. Parser mengenali bahwa "Tim SAR" berfungsi sebagai frasa nominal yang menjadi subjek, "berhasil mengevakuasi" merupakan predikat dengan inti kata kerja "mengevakuasi," "lima korban tanah longsor" adalah objek, sementara "di Desa Cijedil, Kabupaten Cianjur" adalah keterangan tempat, dan "pada Senin pagi" adalah keterangan waktu. Dengan demikian, parser mampu memetakan hubungan sintaktis dalam

kalimat sehingga setiap bagian memiliki peran yang jelas.

Setelah struktur sintaksis diperoleh, evaluator kemudian mengambil alih untuk mengubahnya menjadi representasi semantik yang lebih mudah dipahami dan digunakan. Evaluator menafsirkan bahwa "Tim SAR" adalah pelaku atau agent, "mengevakuasi" adalah aksi utama, "lima korban tanah longsor" adalah objek atau korban yang terlibat, "Desa Cijedil, Kabupaten Cianjur" adalah lokasi kejadian, sedangkan "Senin pagi" merupakan waktu peristiwa. Informasi yang telah ditata ulang ini dapat dituliskan kembali dalam bentuk tabel data atau representasi JSON sehingga lebih mudah diintegrasikan ke dalam sistem informasi bencana.

Studi kasus ini memperlihatkan bahwa parsing berfungsi sebagai jembatan penting untuk mengubah teks naratif menjadi data yang terstruktur. Tanpa parsing, sistem ekstraksi informasi hanya akan memandang kalimat sebagai kumpulan kata kunci yang terpisah-pisah, sehingga rawan terjadi kesalahan interpretasi. Dengan parsing, hubungan antar bagian kalimat menjadi jelas, dan informasi yang diperoleh dapat digunakan secara akurat untuk berbagai keperluan, seperti sistem tanggap darurat, basis data medis, analisis hukum, atau pengolahan ulasan pelanggan.

Dari contoh sederhana ini dapat disimpulkan bahwa parsing tidak hanya relevan untuk memverifikasi struktur bahasa, tetapi juga memiliki nilai praktis yang besar dalam mendukung ekstraksi informasi. Dengan parsing, sistem mampu memahami siapa melakukan apa, kepada siapa, di mana, dan kapan, sehingga teks tidak lagi sekadar rangkaian kata, melainkan sumber informasi yang dapat langsung digunakan dalam pengambilan keputusan.

Latihan

A. Pertanyaan Pemahaman Konsep

1. Apa yang dimaksud dengan *Event Extraction* dalam NLP?
2. Jelaskan perbedaan mendasar antara *Relation Extraction* dan *Event Extraction*.
3. Sebutkan lima komponen utama yang harus diidentifikasi dalam proses ekstraksi peristiwa.
4. Apa yang dimaksud dengan *event trigger* dan bagaimana cara menemukannya dalam teks?
5. Jelaskan bagaimana parsing sintaksis dapat membantu dalam menemukan argumen peristiwa.
6. Sebutkan tiga jenis peristiwa umum yang sering muncul dalam teks berita.
7. Mengapa ekstraksi peristiwa sering kali lebih kompleks dibandingkan ekstraksi relasi?

B. Latihan Praktik Sederhana

Diberikan teks berita berikut:

"Gempa berkekuatan 6,8 SR mengguncang wilayah Maluku pada hari Jumat dan menyebabkan ratusan rumah rusak."

Tugas:

1. Identifikasi elemen-elemen peristiwa yang terdapat dalam kalimat tersebut.
2. Sajikan hasil ekstraksi dalam tabel berikut:

Elemen Peristiwa	Nilai	Keterangan
Trigger	mengguncang	Menandai terjadinya peristiwa gempa
Event Type	Bencana Alam	Kategori peristiwa
Location	Maluku	Lokasi kejadian
Time	Jumat	Waktu kejadian

Effect	ratusan rumah rusak	Dampak peristiwa

3. Jelaskan bagaimana sistem otomatis dapat mengenali *event trigger* "mengguncang" menggunakan pola sintaksis dan konteks semantik.
4. Sebutkan kesalahan yang mungkin muncul jika sistem tidak melakukan POS Tagging dengan benar.

C. Studi Kasus / Proyek Mini

Anda bertugas merancang modul ekstraksi peristiwa untuk sistem pemantauan bencana nasional.

1. Tentukan jenis peristiwa yang akan diekstraksi (misalnya gempa, banjir, kebakaran, tanah longsor).
2. Buat desain pipeline yang terdiri dari tahap NER → RE → EE → visualisasi data.
3. Gunakan tiga contoh kalimat berita berbeda dan lakukan ekstraksi peristiwa manual.
4. Diskusikan bagaimana sistem dapat mengintegrasikan hasil dari berbagai sumber berita untuk membentuk *event timeline*.
5. Analisis bagaimana *event extraction* dapat dihubungkan dengan sistem peringatan dini (*early warning system*).

D. Diskusi / Refleksi

1. Menurut Anda, mengapa *event trigger detection* menjadi tahap paling krusial dalam EE?
2. Bagaimana pendekatan *deep learning* seperti BERT atau LSTM dapat meningkatkan akurasi deteksi peristiwa?
3. Diskusikan kesulitan yang muncul dalam ekstraksi peristiwa dari teks Bahasa Indonesia yang tidak baku (misalnya media sosial).
4. Bagaimana hasil *event extraction* dapat dimanfaatkan dalam penelitian kebencanaan, analisis media, atau sistem intelijen?

BAB 8
DASAR EKSTRAKSI BERBASIS RULE

Tujuan Pembelajaran

Setelah mempelajari Bab 8, mahasiswa diharapkan mampu:

1. Menjelaskan konsep dasar klasifikasi teks (text classification) dan peranannya dalam sistem *Information Extraction (IE)*.
2. Membedakan antara klasifikasi dokumen dan klasifikasi kalimat (sentence-level classification).
3. Menjelaskan tahapan utama dalam sistem klasifikasi teks: representasi fitur, pemilihan algoritma, pelatihan model, dan evaluasi performa.
4. Menguraikan berbagai metode representasi teks seperti *bag-of-words (BoW)*, *TF-IDF*, dan *word embedding*.
5. Mengenali algoritma populer untuk klasifikasi teks, seperti *Naïve Bayes*, *Support Vector Machine (SVM)*, dan *Neural Networks*.
6. Menerapkan model klasifikasi sederhana untuk menentukan kategori isi teks.
7. Mengevaluasi hasil klasifikasi menggunakan metrik akurasi, presisi, recall, dan F1-score

Pendahuluan

Setelah pada bab sebelumnya kita mempelajari tentang aturan produksi dan komponen parsing, kini kita akan masuk ke salah satu pendekatan paling klasik dan mendasar dalam ekstraksi informasi, yaitu ekstraksi berbasis rule (Rule-Based Information Extraction). Pendekatan ini berangkat dari prinsip bahwa bahasa memiliki pola tertentu yang dapat dirumuskan ke dalam aturan formal. Dengan merumuskan aturan-aturan tersebut, sistem komputer dapat mengenali

informasi penting dari teks tanpa memerlukan proses pelatihan data dalam skala besar.

Ekstraksi berbasis rule merupakan pendekatan yang sangat populer pada masa awal perkembangan NLP dan masih banyak digunakan hingga saat ini, terutama untuk bahasa dengan sumber daya terbatas seperti Bahasa Indonesia. Hal ini karena sistem berbasis aturan relatif mudah disesuaikan dengan kebutuhan domain tertentu, misalnya analisis dokumen hukum, laporan medis, atau berita bencana. Aturan dapat ditulis secara manual oleh pakar linguistik atau pengembang sistem, biasanya dengan memanfaatkan pola tata bahasa, keyword, atau ekspresi reguler.

Sebagai contoh sederhana, dalam kalimat "Gempa bumi berkekuatan 6,2 SR mengguncang Kabupaten Cianjur pada Senin pagi," sebuah aturan berbasis pola dapat ditulis untuk mengekstrak informasi:

- Jenis bencana: "Gempa bumi"
- Magnitudo: pola angka + "SR"
- Lokasi: nama kabupaten atau kota
- Waktu: frasa temporal seperti "Senin pagi"

Dengan aturan semacam ini, sistem dapat mengekstrak data penting secara otomatis meskipun tidak dilatih menggunakan ribuan contoh kalimat.

Bab ini akan membahas dasar-dasar pendekatan rule-based, mulai dari konsep dan pola kalimat, teknik penulisan aturan, penggunaan ekspresi reguler, hingga kelebihan dan keterbatasannya. Sebagai penutup, akan diberikan contoh implementasi sederhana agar mahasiswa dapat memahami bagaimana aturan dibuat dan diuji pada teks nyata.

A. Konsep Rule-Based Extraction dan Pola Kalimat

Ekstraksi berbasis aturan (rule-based extraction) merupakan pendekatan paling awal dan fundamental

dalam sistem ekstraksi informasi. Konsep utamanya adalah bahwa bahasa, meskipun kompleks dan penuh variasi, tetap memiliki pola tertentu yang dapat diidentifikasi. Pola ini bisa berupa urutan kata, bentuk morfologis, struktur kalimat, maupun kombinasi keduanya. Dengan memformalisasikan pola tersebut ke dalam aturan eksplisit, sistem komputer dapat mengenali informasi penting secara otomatis dari teks.

Aturan (rules) dalam konteks ini adalah seperangkat instruksi yang menentukan kondisi apa yang harus terpenuhi agar suatu bagian teks dikenali sebagai entitas, relasi, atau peristiwa tertentu. Misalnya, sebuah aturan dapat berbunyi: "Jika sebuah kata diikuti dengan satuan 'SR', maka kemungkinan besar kata tersebut merupakan magnitudo gempa." Aturan sederhana ini memungkinkan sistem menandai "6,2 SR" dalam teks berita sebagai informasi penting tentang kekuatan gempa.

Pola kalimat yang digunakan dalam ekstraksi rule-based biasanya bergantung pada aturan tata bahasa dan karakteristik domain. Dalam bahasa Indonesia, pola umum yang sering dimanfaatkan adalah struktur SPOK (Subjek–Predikat–Objek–Keterangan). Misalnya, pada kalimat "Tim SAR mengevakuasi korban banjir di Desa Cijedil," pola kalimat menunjukkan bahwa:

- Subjek = "Tim SAR"
- Predikat = "mengevakuasi"
- Objek = "korban banjir"
- Keterangan = "di Desa Cijedil"

Dengan memanfaatkan pola ini, aturan dapat ditulis untuk mengekstrak informasi: subjek sebagai pelaku, predikat sebagai aksi, objek sebagai sasaran, dan keterangan sebagai lokasi atau waktu.

Selain berdasarkan SPOK, aturan juga dapat dibangun menggunakan kata kunci (keywords) atau pola tetap (fixed patterns). Sebagai contoh:

- "sebanyak [angka] orang meninggal dunia" → aturan ini dapat digunakan untuk mengekstrak jumlah korban jiwa.
- "[nama pejabat] meresmikan [nama fasilitas]" → aturan ini dapat mengekstrak informasi tentang peristiwa peresmian.
- "harga saham [perusahaan] naik sebesar [angka] persen" → aturan ini dapat digunakan dalam ekstraksi informasi keuangan.

Dalam implementasi praktis, pola-pola tersebut biasanya ditulis menggunakan ekspresi reguler (regular expression/regex). Regex memungkinkan sistem mendeteksi urutan karakter tertentu yang sesuai dengan pola yang telah ditentukan. Sebagai contoh, pola \d+ SR akan cocok dengan teks "6,2 SR" atau "7 SR," sehingga memudahkan ekstraksi magnitudo gempa dari teks berita.

Keunggulan utama dari pendekatan rule-based adalah kesederhanaan dan transparansi. Aturan dapat dipahami, diperiksa, dan dimodifikasi dengan mudah. Jika ada kesalahan, aturan dapat langsung diperbaiki tanpa harus melatih ulang model. Pendekatan ini sangat efektif untuk domain sempit dengan bahasa yang relatif konsisten, seperti laporan medis, kontrak hukum, atau berita resmi.

Namun, rule-based juga memiliki keterbatasan. Bahasa alami sangat bervariasi, dan aturan yang dibuat sering kali tidak mampu mencakup semua kemungkinan variasi kalimat. Misalnya, pola "Tim SAR mengevakuasi korban" berbeda dari "Korban dievakuasi oleh Tim SAR,"

meskipun maknanya sama. Untuk menutupi semua variasi ini, diperlukan aturan tambahan yang jumlahnya bisa sangat banyak, sehingga sistem menjadi sulit dipelihara.

Meskipun demikian, rule-based extraction tetap relevan terutama dalam konteks Bahasa Indonesia yang masih termasuk kategori low-resource language. Dalam kondisi di mana data anotasi untuk pelatihan model machine learning terbatas, sistem berbasis aturan sering kali menjadi solusi praktis. Selain itu, pendekatan ini juga dapat dikombinasikan dengan teknik statistik atau pembelajaran mesin dalam sistem hybrid extraction, sehingga keunggulan keduanya bisa dimanfaatkan sekaligus.

Sebagai penutup subbab ini, dapat ditegaskan bahwa rule-based extraction adalah pondasi penting dalam sejarah dan praktik ekstraksi informasi. Dengan memahami pola kalimat dan cara menuliskan aturan, mahasiswa dapat membangun sistem sederhana yang mampu mengenali informasi dari teks. Meskipun memiliki keterbatasan, pendekatan ini tetap bermanfaat untuk aplikasi domain-spesifik dan sebagai pelatihan awal sebelum beranjak ke teknik ekstraksi berbasis machine learning yang lebih kompleks.

B. Teknik Penulisan dan Penerapan Rule

Setelah memahami konsep dasar rule-based extraction dan pola kalimat, langkah selanjutnya adalah bagaimana aturan tersebut ditulis dan diterapkan secara praktis dalam sistem ekstraksi informasi. Teknik penulisan aturan merupakan inti dari pendekatan rule-based karena kualitas aturan yang dibuat akan langsung memengaruhi akurasi hasil ekstraksi. Aturan yang terlalu kaku bisa

melewatkan informasi penting, sementara aturan yang terlalu longgar berpotensi menghasilkan banyak kesalahan (false positive). Oleh karena itu, perancangan aturan harus dilakukan dengan hati-hati, melalui pemahaman linguistik sekaligus analisis kebutuhan domain.

Ada beberapa prinsip dasar yang perlu diperhatikan dalam penulisan aturan ekstraksi. Pertama, aturan harus spesifik terhadap pola kalimat atau entitas yang ingin diekstrak. Misalnya, untuk mengekstrak jumlah korban bencana dari berita, aturan dapat dibuat untuk mengenali pola "sebanyak [angka] orang meninggal dunia" atau "korban tewas mencapai [angka]." Kedua, aturan harus konsisten dan dapat diterapkan berulang kali pada teks yang berbeda namun masih dalam domain yang sama. Konsistensi penting agar hasil ekstraksi tidak berubah-ubah hanya karena variasi kecil dalam penulisan. Ketiga, aturan harus mudah dipahami dan dimodifikasi, sehingga ketika terjadi kesalahan, pengembang sistem dapat dengan cepat menyesuaikan atau memperbaikinya.

Salah satu teknik umum adalah memanfaatkan pola linguistik berdasarkan struktur SPOK (Subjek–Predikat–Objek–Keterangan). Misalnya, dalam kalimat "Presiden Joko Widodo meresmikan bendungan di Nusa Tenggara Timur pada 23 Februari 2022," aturan dapat dirumuskan sebagai:

- Jika ada pola [NP] + [V] + [NP] + [PP] + [PP-Time], maka:
  - NP pertama = subjek (pelaku)
  - V = aksi
  - NP kedua = objek
  - PP pertama = lokasi
  - PP kedua = waktu

Dengan aturan ini, sistem dapat secara konsisten mengekstrak entitas utama dari berita peresmian atau kegiatan resmi.

Selain pola sintaksis, aturan juga dapat ditulis berdasarkan kata kunci tertentu. Teknik ini lebih sederhana dan cocok digunakan dalam domain yang menggunakan kosakata khas. Sebagai contoh, untuk laporan bencana, aturan berbasis kata kunci dapat berupa:

- Jika ada kata "gempa" diikuti angka dan satuan "SR," maka tandai sebagai magnitudo gempa.
- Jika ada frasa "korban meninggal" atau "korban tewas," ekstrak angka yang mendahuluinya sebagai jumlah korban jiwa.
- Jika ada kata "banjir" atau "longsor," tandai sebagai jenis bencana.

Meskipun sederhana, aturan berbasis kata kunci efektif dalam domain yang sangat terfokus.

Ekspresi reguler (regex) merupakan alat yang sangat kuat untuk menulis aturan dalam bentuk pola pencarian teks. Regex memungkinkan pengembang mendefinisikan pola karakter yang fleksibel namun terstruktur. Misalnya, pola regex \d+\s*SR dapat digunakan untuk mengenali teks seperti "6,2 SR" atau "7 SR" sebagai magnitudo gempa. Demikian juga, regex \d+\s*orang dapat menangkap pola "162 orang" atau "20 orang" dalam berita korban bencana.

Contoh lain:

- Pola tanggal→\d{1,2}\s(Januari|Februari|Maret|April|Mei|Juni|Juli|Agustus|September|Oktober|November|Desember)\s\d{4}

Aturan ini dapat mengenali tanggal lengkap seperti "23 Februari 2022."

Dengan regex, aturan dapat ditulis lebih ringkas dan fleksibel, meskipun pengembang perlu berhati-hati karena pola yang terlalu umum bisa menghasilkan banyak kesalahan.

Dalam implementasi praktis, aturan biasanya diterapkan setelah tahap preprocessing dan parsing. Setelah teks dibersihkan, di-tokenisasi, dan dianalisis secara sintaksis, barulah aturan diterapkan untuk mengekstrak informasi sesuai kebutuhan. Misalnya, dari hasil parsing sebuah kalimat, sistem dapat memeriksa apakah terdapat pola [angka] + [orang] + [kata meninggal/tewas]. Jika pola ini ditemukan, maka informasi jumlah korban dapat diekstrak.

Aturan juga dapat dikombinasikan dengan kamus entitas (gazetteer) untuk meningkatkan akurasi. Misalnya, untuk ekstraksi lokasi, aturan dapat ditulis: "Jika token berupa nama dari daftar kabupaten/kota Indonesia, maka tandai sebagai entitas lokasi." Dengan kombinasi aturan dan kamus, hasil ekstraksi akan lebih presisi.

Penulisan aturan memiliki beberapa kelebihan, yaitu dapat memberikan hasil yang transparan, mudah ditelusuri, dan akurat untuk domain tertentu. Namun, tantangannya adalah aturan bisa menjadi sangat banyak dan kompleks seiring bertambahnya variasi bahasa. Selain itu, aturan yang ditulis untuk satu domain sering kali tidak dapat langsung dipakai pada domain lain tanpa penyesuaian yang signifikan.

Dengan memahami teknik penulisan dan penerapan rule, mahasiswa diharapkan mampu membangun sistem ekstraksi informasi sederhana. Misalnya, mahasiswa dapat mencoba membuat aturan untuk mengekstrak data dari

berita bencana, laporan keuangan, atau bahkan status media sosial. Latihan semacam ini akan membantu mengasah kemampuan untuk mengenali pola bahasa dan menuangkannya ke dalam bentuk aturan formal yang dapat dijalankan komputer.

C. Penggunaan Ekspresi Reguler dalam Rule

Salah satu teknik terpenting dalam membangun aturan ekstraksi informasi berbasis rule adalah penggunaan ekspresi reguler atau yang lebih dikenal dengan istilah regular expression (regex). Ekspresi reguler merupakan suatu bahasa mini yang dirancang khusus untuk mencocokkan pola tertentu di dalam teks. Dengan regex, kita dapat mencari, mengenali, dan mengekstrak rangkaian karakter sesuai pola yang sudah ditentukan, tanpa harus menuliskan instruksi pencarian secara manual untuk setiap variasi kata atau kalimat.

Konsep dasar dari ekspresi reguler adalah mendefinisikan pattern yang menggambarkan bagaimana teks target tersusun. Misalnya, pola \d+ dalam regex digunakan untuk mengenali deretan angka apa pun, baik "23," "162," atau "2023." Simbol \d berarti digit (angka), sementara tanda + berarti "satu atau lebih kemunculan." Dengan cara ini, sistem dapat menandai bagian teks yang berupa angka tanpa harus menentukan nilainya satu per satu. Contoh lain adalah penggunaan simbol | sebagai operator "atau." Dengan pola (gempa|banjir|longsor), regex dapat mengenali salah satu kata dari ketiganya di dalam teks. Pola sederhana seperti ini sudah cukup kuat untuk digunakan dalam ekstraksi informasi, khususnya ketika kita ingin mendeteksi jenis peristiwa tertentu yang sering muncul dalam berita.

Dalam praktik ekstraksi informasi, regex sering

digunakan untuk menangani pola yang teratur tetapi bervariasi. Misalnya, tanggal dalam Bahasa Indonesia dapat ditulis dengan berbagai cara: "23 Februari 2022," "23/02/2022," atau "23-02-22." Dengan menggunakan regex, semua variasi ini dapat ditangkap menggunakan satu aturan saja. Contohnya, pola \d{1,2}[/-]\d{1,2}[/-]\d{2,4} dapat digunakan untuk mengenali format tanggal yang ditulis dengan garis miring (/) atau tanda hubung (-). Sementara itu, pola \d{1,2}\s(Januari|Februari|Maret|April|Mei|Juni|Juli|Agustus|September|Oktober|November|Desember)\s\d{4} dapat digunakan untuk mengenali tanggal dalam format panjang seperti "23 Februari 2022." Dengan cara ini, kita tidak hanya mengandalkan satu format, tetapi mampu menyesuaikan dengan variasi yang umum digunakan dalam teks.

Selain untuk tanggal, regex juga sering digunakan untuk mengekstrak angka dan satuan. Dalam berita bencana, jumlah korban biasanya ditulis seperti "162 orang meninggal" atau "20 orang luka-luka." Dengan pola regex \d+\s*orang, sistem dapat mengenali kedua bentuk tersebut. Simbol \s* berarti ada nol atau lebih spasi, sehingga pola tetap valid meskipun penulisan mengandung variasi jarak. Lebih lanjut, untuk mengekstrak magnitudo gempa, aturan dapat ditulis dengan pola \d+(\.\d+)?\s*SR, yang dapat mengenali "6 SR," "6.2 SR," atau "7,5 SR." Penggunaan tanda (.\d+)? menunjukkan angka desimal opsional, sehingga sistem lebih fleksibel.

Kekuatan ekspresi reguler juga tampak dalam menangani teks semi-terstruktur seperti alamat email, nomor telepon, atau kode identifikasi. Misalnya, pola [\w\.-]+@[\w\.-]+ dapat digunakan untuk mengenali

alamat email, sedangkan pola \+62\d+ cocok untuk mendeteksi nomor telepon Indonesia dengan kode negara. Walaupun konteks buku ini lebih banyak membahas teks naratif seperti berita atau laporan, pemahaman tentang regex semacam ini tetap penting karena banyak informasi yang tersimpan dalam format semi-teratur.

Dalam sistem rule-based extraction, ekspresi reguler biasanya dipadukan dengan hasil preprocessing dan parsing. Setelah teks dibersihkan dan di-tokenisasi, regex diterapkan untuk mengenali pola tertentu yang sesuai dengan kebutuhan domain. Sebagai contoh, dari kalimat "Sebanyak 162 orang meninggal dunia akibat gempa 6,2 SR di Cianjur pada 21 November 2022," regex dapat digunakan untuk mengekstrak tiga informasi penting: \d+\s*orang untuk jumlah korban, \d+(\.\d+)?\s*SR untuk magnitudo gempa, dan pola tanggal untuk waktu kejadian. Dengan kombinasi tiga aturan ini, sistem dapat mengubah teks panjang menjadi catatan ringkas berisi "162 korban meninggal, magnitudo 6,2 SR, 21 November 2022."

Walaupun sangat kuat, penggunaan ekspresi reguler juga memiliki tantangan. Pertama, regex cenderung sulit dibaca dan dipahami oleh orang yang belum terbiasa. Pola yang kompleks sering kali terlihat seperti rangkaian simbol tanpa makna. Kedua, regex memiliki batasan dalam menangani bahasa alami yang sangat variatif. Sebuah aturan regex mungkin bekerja baik pada teks formal, tetapi gagal pada teks media sosial yang penuh singkatan, emotikon, atau campur kode. Oleh karena itu, regex paling efektif digunakan pada domain dengan pola yang relatif konsisten, seperti berita resmi, laporan medis, atau dokumen hukum.

Dalam konteks pembelajaran, mahasiswa perlu

berlatih menulis berbagai pola regex sederhana hingga kompleks. Latihan dapat dimulai dari mendeteksi angka, tanggal, atau nama entitas, lalu meningkat ke pola-pola gabungan yang lebih rumit. Dengan pengalaman tersebut, mahasiswa akan memahami bagaimana menyeimbangkan antara spesifisitas (agar hasil ekstraksi tepat sasaran) dan fleksibilitas (agar aturan dapat mencakup variasi penulisan).

Secara keseluruhan, ekspresi reguler adalah alat yang sangat efektif dalam rule-based extraction. Ia memungkinkan sistem mengenali pola yang berulang dalam teks tanpa perlu data latih yang besar. Meskipun memiliki keterbatasan, regex tetap menjadi komponen penting dalam banyak sistem ekstraksi informasi, baik sebagai metode utama maupun sebagai bagian dari sistem hibrida yang menggabungkan aturan dengan pembelajaran mesin. Dengan menguasai regex, mahasiswa akan memiliki keterampilan praktis untuk membangun sistem ekstraksi sederhana sekaligus pemahaman mendasar untuk mengembangkan sistem yang lebih kompleks.

D. Kelebihan dan Kekurangan Pendekatan Rule-Based

Pendekatan berbasis aturan (rule-based approach) dalam ekstraksi informasi memiliki posisi yang unik dalam sejarah perkembangan NLP. Metode ini menjadi salah satu pionir yang membentuk fondasi bagi sistem ekstraksi informasi modern. Hingga saat ini, rule-based masih banyak digunakan, baik sebagai metode utama maupun sebagai bagian dari sistem hibrida yang dikombinasikan dengan teknik machine learning. Untuk memahami relevansi dan batasannya, kita perlu melihat kelebihan sekaligus kekurangannya.

Salah satu kelebihan utama dari pendekatan rule-based adalah transparansi dan interpretabilitas. Aturan yang ditulis secara eksplisit mudah dibaca, dipahami, dan diperiksa oleh manusia. Jika terjadi kesalahan dalam hasil ekstraksi, pengembang dapat langsung melacak aturan yang menyebabkan kesalahan tersebut, lalu memperbaikinya. Hal ini berbeda dengan model berbasis machine learning yang sering kali bersifat "kotak hitam" dan sulit dijelaskan proses pengambilan keputusannya. Transparansi ini sangat bermanfaat terutama di bidang sensitif seperti hukum, kesehatan, dan pemerintahan, di mana setiap hasil ekstraksi harus dapat dipertanggungjawabkan.

Kelebihan lain adalah efisiensi pada domain sempit. Rule-based sangat efektif jika digunakan dalam lingkup teks dengan variasi bahasa yang terbatas. Sebagai contoh, laporan medis atau berita resmi cenderung menggunakan struktur kalimat yang relatif konsisten. Dengan aturan yang tepat, sistem dapat menghasilkan ekstraksi informasi yang akurat tanpa perlu melatih model dengan data dalam jumlah besar. Hal ini juga membuat rule-based sangat berguna untuk bahasa dengan sumber daya terbatas (low-resource languages) seperti Bahasa Indonesia, yang belum memiliki banyak dataset beranotasi untuk melatih model berbasis pembelajaran mesin.

Selain itu, rule-based memiliki kelebihan dari sisi biaya dan kecepatan pengembangan awal. Untuk membuat sistem ekstraksi sederhana, pengembang tidak memerlukan komputer dengan spesifikasi tinggi atau dataset yang besar. Cukup dengan memahami pola kalimat dalam domain tertentu, aturan dapat segera ditulis dan diuji. Pendekatan ini membuat rule-based cocok sebagai titik awal pembelajaran bagi mahasiswa atau

peneliti pemula di bidang ekstraksi informasi.

Namun, di balik kelebihan tersebut, pendekatan rule-based memiliki beberapa kelemahan mendasar. Pertama adalah ketidakmampuan menangani variasi bahasa yang tinggi. Bahasa alami sangat kaya dan penuh fleksibilitas. Kalimat dengan makna sama dapat ditulis dengan banyak cara berbeda. Sebagai contoh, informasi "Tim SAR mengevakuasi korban banjir" dapat ditulis ulang menjadi "Korban banjir dievakuasi oleh Tim SAR" atau "Evakuasi korban banjir dilakukan oleh Tim SAR." Sistem berbasis aturan harus menulis aturan terpisah untuk setiap variasi ini. Jika variasi bahasa semakin banyak, jumlah aturan yang harus ditulis akan membengkak dan sulit dikelola.

Kelemahan kedua adalah kesulitan skalabilitas. Sistem rule-based yang sederhana mungkin hanya memerlukan puluhan aturan. Namun, dalam aplikasi dunia nyata dengan domain yang kompleks, jumlah aturan bisa mencapai ratusan hingga ribuan. Semakin banyak aturan yang ditulis, semakin sulit menjaga konsistensi antaraturan, dan semakin besar risiko terjadinya konflik. Kondisi ini membuat sistem rule-based sulit dipelihara dalam jangka panjang, terutama jika digunakan oleh organisasi dengan kebutuhan analisis teks yang terus berkembang.

Keterbatasan lain adalah minimnya kemampuan generalisasi. Aturan yang dibuat untuk satu domain biasanya tidak dapat digunakan begitu saja pada domain lain. Misalnya, aturan yang efektif untuk ekstraksi informasi dalam laporan medis mungkin sama sekali tidak relevan untuk berita ekonomi. Hal ini membuat rule-based kurang fleksibel jika dibandingkan dengan metode pembelajaran mesin modern, yang dapat dilatih ulang

dengan dataset baru untuk menyesuaikan ke domain berbeda.

Terakhir, rule-based cenderung memiliki keterbatasan performa pada teks informal. Dalam era digital, sebagian besar teks yang dihasilkan manusia berasal dari media sosial, forum daring, atau percakapan informal. Teks seperti ini penuh dengan singkatan, emotikon, kesalahan ejaan, bahkan campur kode antar bahasa. Aturan yang kaku akan kesulitan menghadapi keragaman ini, sehingga tingkat akurasi ekstraksi menurun drastis.

Dengan mempertimbangkan kelebihan dan kekurangan tersebut, dapat disimpulkan bahwa rule-based masih relevan, terutama dalam domain sempit dengan teks yang terkontrol. Namun, untuk domain yang lebih luas dan dinamis, pendekatan ini sering kali lebih efektif jika dipadukan dengan teknik statistik atau pembelajaran mesin, membentuk sistem ekstraksi informasi hibrida. Dengan demikian, mahasiswa perlu memahami rule-based tidak hanya sebagai metode klasik, tetapi juga sebagai komponen penting yang dapat melengkapi metode modern dalam praktik ekstraksi informasi.

E. Implementasi Dasar Rule Extraction

Untuk memberikan gambaran lebih konkret mengenai bagaimana aturan berbasis rule diterapkan dalam ekstraksi informasi, mari kita bahas contoh sederhana yang menggunakan teks berita bencana. Tujuan dari contoh ini adalah menunjukkan bagaimana aturan yang ditulis secara eksplisit dapat membantu sistem mengenali informasi penting dari sebuah kalimat.

Misalkan kita memiliki potongan berita:

"Gempa bumi berkekuatan 6,2 SR mengguncang Kabupaten Cianjur pada 21 November 2022. Akibat peristiwa tersebut, sebanyak 162 orang meninggal dunia dan ratusan lainnya luka-luka."

Dari kalimat tersebut, kita ingin mengekstrak informasi utama, yaitu: jenis bencana, kekuatan gempa, lokasi, waktu, dan jumlah korban.

1. **Tahap 1: Identifikasi Pola Kalimat**

   Langkah pertama adalah mengenali pola umum dalam teks. Beberapa pola yang dapat diamati adalah:

   a. Jenis bencana biasanya muncul di awal kalimat, misalnya "Gempa bumi" atau "Banjir."
   b. Kekuatan gempa ditulis sebagai angka diikuti oleh satuan "SR."
   c. Lokasi biasanya muncul setelah kata kerja seperti "mengguncang" atau "melanda."
   d. Waktu sering kali diperkenalkan dengan kata depan seperti "pada" atau "di."
   e. Jumlah korban ditulis sebagai angka yang diikuti kata "orang" serta frasa "meninggal dunia" atau "luka-luka."

   Pola-pola ini akan menjadi dasar aturan ekstraksi.

2. **Tahap 2: Menulis Aturan Berbasis Regex**

   Setelah pola dikenali, aturan dapat ditulis menggunakan ekspresi reguler. Beberapa contoh aturan sederhana adalah sebagai berikut:

   a. Jenis bencana: jika teks mengandung kata "gempa," "banjir," atau "longsor," tandai sebagai entitas bencana.
   b. Magnitudo gempa: gunakan regex \d+(\.\d+)?\s*SR untuk menangkap pola "6,2 SR" atau "7 SR."

c. Lokasi: jika ada kata kerja seperti "mengguncang" atau "melanda" diikuti nama wilayah, ekstrak kata setelahnya sebagai lokasi.
d. Tanggal: gunakan regex \d{1,2}\s(Januari|Februari|Maret|April|Mei|Juni|Juli|Agustus|September|Oktober|November|Desember)\s\d{4} untuk menangkap tanggal seperti "21 November 2022."
e. Jumlah korban: gunakan regex \d+\s*orang\s+(meninggal dunia|tewas|luka-luka) untuk menangkap frasa seperti "162 orang meninggal dunia."

Dengan aturan-aturan ini, sistem dapat langsung mengenali informasi yang relevan tanpa memerlukan proses pelatihan model.

3. Tahap 3: Penerapan Aturan pada Teks

Jika aturan tersebut diterapkan pada teks berita di atas, hasil ekstraksi yang diperoleh adalah:

a. Jenis bencana: Gempa bumi
b. Magnitudo: 6,2 SR
c. Lokasi: Kabupaten Cianjur
d. Waktu: 21 November 2022
e. Korban jiwa: 162 orang meninggal dunia

Informasi tersebut dapat disajikan dalam bentuk tabel atau JSON agar lebih mudah diproses oleh sistem komputer.

Representasi tabel:

Elemen Informasi	Nilai
Jenis Bencana	Gempa bumi
Magnitudo	6,2 SR
Lokasi	Kabupaten Cianjur
Waktu	21 November 2022
Korban Jiwa	162 orang meninggal

Representasi JSON:

```
{
  "jenis_bencana": "Gempa bumi",
  "magnitudo": "6,2 SR",
  "lokasi": "Kabupaten Cianjur",
  "waktu": "21 November 2022",
  "korban_jiwa": "162 orang meninggal dunia"
}
```

4. **Tahap 4: Analisis Kelebihan dan Keterbatasan Implementasi**

   Contoh implementasi sederhana ini memperlihatkan kekuatan rule-based extraction dalam menangkap pola yang jelas dan konsisten. Aturan yang ditulis dengan regex dapat secara cepat mengubah teks naratif menjadi data terstruktur yang siap digunakan, misalnya untuk sistem peringatan dini atau laporan kebencanaan.

   Namun, contoh ini juga menegaskan keterbatasan pendekatan rule-based. Jika teks ditulis dengan gaya berbeda, misalnya "162 jiwa kehilangan nyawa akibat gempa," aturan di atas mungkin gagal mengenalinya karena tidak ada frasa "orang meninggal dunia." Oleh karena itu, pengembang harus menambahkan aturan tambahan agar sistem lebih fleksibel. Hal ini menunjukkan bahwa meskipun sederhana dan praktis,

rule-based extraction membutuhkan banyak aturan untuk menutupi berbagai variasi bahasa alami.

Contoh implementasi ini menunjukkan bagaimana teori tentang aturan berbasis pola dan regex dapat diterapkan dalam kasus nyata. Mahasiswa diharapkan dapat memahami bahwa dengan mengenali pola teks dan menuliskannya dalam bentuk aturan eksplisit, sistem ekstraksi informasi dapat dibangun meskipun tanpa teknologi machine learning yang kompleks. Sebagai latihan, mahasiswa dapat mencoba menulis aturan serupa untuk domain lain, seperti laporan medis (misalnya ekstraksi nama penyakit, obat, dosis), dokumen hukum (ekstraksi nama pihak, tanggal perjanjian), atau ulasan produk (ekstraksi fitur produk dan sentimen).

Latihan

A. Pertanyaan Pemahaman Konsep

1. Apa yang dimaksud dengan *text classification* dan bagaimana perannya dalam ekstraksi informasi?
2. Sebutkan perbedaan antara *document-level classification* dan *sentence-level classification*.
3. Jelaskan prinsip kerja metode *bag-of-words* dan bagaimana metode tersebut merepresentasikan teks secara numerik.
4. Mengapa *TF-IDF* sering dianggap lebih unggul dibanding *bag-of-words*?
5. Sebutkan tiga algoritma yang umum digunakan dalam klasifikasi teks dan jelaskan karakteristik utamanya.
6. Apa fungsi dari *training data* dan *testing data* dalam sistem klasifikasi?
7. Bagaimana kita menilai performa model klasifikasi teks secara objektif?

B. Latihan Praktik Sederhana

Diberikan tiga kalimat berikut:

1. "Banjir besar melanda Jakarta dan menyebabkan kemacetan di sejumlah ruas jalan."
2. "Harga saham perusahaan teknologi meningkat tajam pada kuartal kedua."
3. "Tim nasional Indonesia berhasil memenangkan pertandingan melawan Malaysia."

Tugas:

1. Tentukan kategori untuk masing-masing kalimat (misal: *bencana*, *ekonomi*, *olahraga*).
2. Representasikan setiap kalimat menggunakan model *bag-of-words* sederhana.
3. Tunjukkan bagaimana bobot *TF-IDF* dapat

membedakan kata-kata yang lebih informatif.

4. Buat tabel hasil klasifikasi berikut:

Kalimat	Fitur Dominan	Kategori Prediksi	Keterangan
Kalimat 1	banjir, Jakarta, kemacetan	Bencana	Mengandung istilah bencana dan lokasi
Kalimat 2	saham, perusahaan, kuartal	Ekonomi	Istilah finansial
Kalimat 3	tim, pertandingan, menang	Olahraga	Mengandung kata kompetitif

5. Jelaskan tantangan jika kategori teks yang dianalisis memiliki tumpang tindih (misalnya "politik" dan "ekonomi").

C. Studi Kasus / Proyek Mini

Anda akan membangun modul klasifikasi teks otomatis untuk membantu sistem *Information Extraction* pada berita nasional.

1. Tentukan skema kategori (misalnya: *bencana*, *politik*, *ekonomi*, *kesehatan*, *olahraga*).
2. Siapkan dataset minimal 20 kalimat contoh untuk masing-masing kategori.
3. Gunakan representasi *TF-IDF* dan algoritma *Naïve Bayes* untuk melakukan klasifikasi.
4. Buat tabel hasil evaluasi menggunakan metrik akurasi, presisi, recall, dan F1-score.
5. Diskusikan bagaimana hasil klasifikasi ini dapat membantu proses ekstraksi peristiwa atau entitas pada bab sebelumnya.

D. Diskusi / Refleksi

1. Menurut Anda, apakah model berbasis *rule-based* masih relevan untuk klasifikasi teks modern?
2. Bagaimana *word embedding* seperti Word2Vec atau BERT meningkatkan kemampuan model klasifikasi teks?
3. Diskusikan hubungan antara *text classification* dan *information filtering* dalam sistem rekomendasi berita.
4. Bagaimana cara mengadaptasi sistem klasifikasi teks agar mampu bekerja dengan Bahasa Indonesia yang memiliki variasi dialek dan morfologi yang kaya?

BAB 9
METODE EKSTRAKSI BERBASIS RULE LANJUTAN

Tujuan

Setelah mempelajari Bab 9, mahasiswa diharapkan mampu:

1. Menjelaskan konsep dasar *Sentiment Analysis* dan *Opinion Mining* serta perannya dalam sistem *Information Extraction (IE)*.
2. Membedakan antara analisis sentimen, deteksi emosi, dan analisis opini.
3. Menjelaskan tahapan utama dalam analisis sentimen: preprocessing, ekstraksi fitur, klasifikasi polaritas, dan evaluasi.
4. Menjelaskan perbedaan antara pendekatan berbasis leksikon (*lexicon-based*) dan pembelajaran mesin (*machine learning-based*).
5. Mengidentifikasi aplikasi analisis sentimen dalam berbagai bidang: pemasaran, politik, media sosial, dan layanan publik.
6. Menerapkan contoh sederhana analisis sentimen terhadap teks berbahasa Indonesia.
7. Mengevaluasi hasil analisis menggunakan metrik seperti akurasi, presisi, recall, dan *F1-score*.

Pendahuluan

Bab sebelumnya telah membahas dasar-dasar ekstraksi informasi berbasis aturan (rule-based extraction), mulai dari konsep pola kalimat, teknik penulisan aturan, hingga penggunaan ekspresi reguler dalam praktik sederhana. Meskipun metode dasar ini efektif untuk domain sempit, dalam aplikasi nyata sering kali dibutuhkan pendekatan yang

lebih canggih. Hal ini karena bahasa alami memiliki variasi yang tinggi, dan teks yang dianalisis biasanya datang dari berbagai sumber dengan gaya penulisan berbeda, baik dari berita formal, dokumen hukum, laporan medis, hingga percakapan di media sosial.

Untuk menjawab tantangan tersebut, muncul berbagai metode rule-based lanjutan yang menggabungkan pemahaman linguistik dengan pendekatan semi-formal, serta memanfaatkan teknologi modern untuk meningkatkan fleksibilitas aturan. Metode lanjutan ini bertujuan mengurangi kelemahan rule-based murni, misalnya dengan mendukung aturan yang lebih generik, menggunakan gazetteer atau kamus entitas, hingga mengintegrasikan parsing sintaksis dan semantik.

Beberapa pendekatan lanjutan dalam rule-based extraction antara lain:

1. Penggunaan Gazetteer dan Kamus Domain-Specific, yaitu daftar kata atau frasa yang dikumpulkan dari domain tertentu untuk membantu identifikasi entitas.
2. Aturan Berbasis Dependency Parsing, di mana hubungan gramatikal antar kata digunakan sebagai pola ekstraksi, bukan sekadar urutan kata.
3. Template dan Pattern Matching Tingkat Lanjut, yaitu penerapan pola kompleks yang mampu menangani variasi bahasa lebih luas.
4. Kombinasi Rule-Based dengan Statistik (Hybrid Approach), yang menggabungkan aturan eksplisit dengan metode berbasis probabilistik agar hasil ekstraksi lebih fleksibel.
5. Sistem Rule Engine Modern, yang memanfaatkan perangkat lunak khusus untuk menyimpan, mengelola, dan mengeksekusi aturan secara terstruktur dan skalabel.

Dengan mempelajari metode lanjutan ini, mahasiswa akan memahami bahwa rule-based extraction tidak berhenti pada aturan sederhana berbasis kata kunci atau regex, melainkan dapat dikembangkan menjadi sistem yang lebih kuat dan adaptif. Pemahaman ini juga akan membantu mahasiswa dalam merancang sistem ekstraksi informasi yang lebih akurat, efisien, dan sesuai dengan kebutuhan domain.

A. Penggunaan Gazetteer dan Kamus Domain-Specific

Salah satu kelemahan utama dari ekstraksi berbasis aturan sederhana adalah keterbatasannya dalam mengenali entitas yang memiliki banyak variasi penulisan. Misalnya, nama lokasi dapat ditulis dalam berbagai bentuk, seperti "Jakarta," "DKI Jakarta," atau "Ibukota." Demikian juga dengan nama penyakit, obat, atau istilah teknis dalam domain tertentu yang mungkin ditulis dengan singkatan, sinonim, atau variasi ejaan. Untuk mengatasi masalah ini, salah satu teknik yang umum digunakan dalam metode rule-based lanjutan adalah memanfaatkan gazetteer atau kamus domain-spesifik.

Gazetteer pada dasarnya adalah sebuah daftar kata atau frasa yang telah dikurasi sebelumnya, biasanya berisi entitas tertentu yang relevan dengan domain aplikasi. Misalnya, dalam sistem ekstraksi informasi untuk berita kebencanaan, gazetteer dapat berisi daftar nama provinsi, kabupaten, kota, sungai, gunung, atau istilah bencana alam. Dengan adanya daftar ini, aturan ekstraksi dapat lebih mudah mengenali entitas lokasi meskipun ditulis dalam bentuk yang bervariasi. Sebagai contoh, jika sebuah aturan berbunyi "tandai setiap kata yang terdapat dalam daftar kabupaten/kota Indonesia sebagai entitas lokasi", maka sistem tidak perlu lagi menulis aturan terpisah untuk mengenali "Bandung," "Cianjur," atau "Yogyakarta," karena semuanya sudah terdaftar dalam

gazetteer.

Selain lokasi, gazetteer juga banyak digunakan dalam domain medis. Sebagai ilustrasi, sebuah sistem ekstraksi informasi dari rekam medis mungkin menggunakan gazetteer berisi nama penyakit (misalnya "pneumonia," "diabetes mellitus," "hipertensi"), nama obat (misalnya "amoksisilin," "paracetamol," "aspirin"), serta daftar singkatan medis (misalnya "BP" untuk blood pressure, "HR" untuk heart rate). Dengan cara ini, sistem dapat lebih mudah mengidentifikasi istilah medis dalam teks naratif dokter yang biasanya penuh variasi.

Kamus domain-specific memiliki fungsi serupa, namun lebih luas dibanding gazetteer. Jika gazetteer biasanya berupa daftar entitas, kamus domain-specific sering kali mencakup definisi, sinonim, serta hubungan antar istilah. Sebagai contoh, dalam kamus hukum, entri "terdakwa" dapat dilengkapi dengan sinonim "tersangka" (dalam konteks awal kasus), atau dalam kamus bisnis, istilah "pendapatan" dapat dihubungkan dengan istilah lain seperti "revenue" atau "income." Integrasi kamus seperti ini ke dalam sistem ekstraksi berbasis aturan akan meningkatkan robustness sistem dalam menghadapi variasi bahasa.

Secara teknis, penerapan gazetteer dan kamus dalam rule-based extraction dapat dilakukan dengan dua cara. Pertama, aturan ekstraksi dapat langsung memeriksa apakah suatu token atau frasa ada dalam daftar entitas. Misalnya, aturan: "Jika sebuah token cocok dengan entri dalam daftar nama provinsi Indonesia, tandai sebagai entitas lokasi." Kedua, aturan dapat menggunakan kombinasi regex dengan lookup ke dalam gazetteer, sehingga sistem tidak hanya mengenali pola karakter tetapi juga mengecek validitas entitas berdasarkan daftar

yang ada.

Sebagai contoh implementasi, perhatikan teks berikut:

"Banjir bandang melanda Kabupaten Cianjur dan Kabupaten Garut pada awal Desember 2022."

Dengan bantuan gazetteer yang berisi daftar kabupaten di Jawa Barat, sistem dapat langsung menandai "Kabupaten Cianjur" dan "Kabupaten Garut" sebagai entitas lokasi, tanpa harus menulis aturan regex yang rumit.

Kelebihan utama penggunaan gazetteer adalah peningkatan akurasi ekstraksi. Sistem menjadi lebih presisi karena hanya mengenali entitas yang telah dikurasi. Selain itu, gazetteer juga mudah diperluas cukup dengan menambahkan entri baru, maka cakupan sistem meningkat tanpa harus mengubah aturan inti. Namun, kelemahan metode ini adalah ketergantungan pada kelengkapan daftar. Jika suatu entitas baru muncul tetapi belum ada dalam gazetteer, sistem kemungkinan gagal mengenalinya. Oleh karena itu, gazetteer memerlukan proses pembaruan berkala agar tetap relevan dengan data terbaru.

Dalam konteks pembelajaran, mahasiswa dapat mencoba membuat gazetteer sederhana sesuai domain yang diminati. Misalnya, untuk domain pariwisata, gazetteer dapat berisi nama-nama objek wisata di Indonesia. Untuk domain kesehatan, gazetteer dapat berupa daftar penyakit umum. Dengan cara ini, mahasiswa tidak hanya memahami teori, tetapi juga dapat langsung merasakan bagaimana penggunaan daftar entitas dapat memperkuat hasil ekstraksi informasi berbasis aturan.

Dengan demikian, penggunaan gazetteer dan kamus

domain-specific merupakan langkah penting dalam metode rule-based lanjutan. Teknik ini menjembatani kelemahan aturan sederhana yang terlalu bergantung pada pola kalimat, dan menjadikan sistem ekstraksi lebih adaptif terhadap variasi penulisan istilah.

Pendekatan InNer (Information-Named Entity Rules)

Salah satu kelemahan utama dari ekstraksi berbasis aturan sederhana adalah keterbatasannya dalam mengenali entitas yang memiliki banyak variasi penulisan. Misalnya, nama lokasi dapat ditulis dalam berbagai bentuk, seperti "Jakarta," "DKI Jakarta," atau "Ibukota." Demikian juga dengan nama penyakit, obat, atau istilah teknis dalam domain tertentu yang mungkin ditulis dengan singkatan, sinonim, atau variasi ejaan. Untuk mengatasi masalah ini, salah satu teknik yang umum digunakan dalam metode rule-based lanjutan adalah memanfaatkan gazetteer atau kamus domain-spesifik.

Gazetteer pada dasarnya adalah sebuah daftar kata atau frasa yang telah dikurasi sebelumnya, biasanya berisi entitas tertentu yang relevan dengan domain aplikasi. Misalnya, dalam sistem ekstraksi informasi untuk berita kebencanaan, gazetteer dapat berisi daftar nama provinsi, kabupaten, kota, sungai, gunung, atau istilah bencana alam. Dengan adanya daftar ini, aturan ekstraksi dapat lebih mudah mengenali entitas lokasi meskipun ditulis dalam bentuk yang bervariasi. Sebagai contoh, jika sebuah aturan berbunyi "tandai setiap kata yang terdapat dalam daftar kabupaten/kota Indonesia sebagai entitas lokasi", maka sistem tidak perlu lagi menulis aturan terpisah untuk mengenali "Bandung," "Cianjur," atau "Yogyakarta," karena semuanya sudah terdaftar dalam gazetteer.

Selain lokasi, gazetteer juga banyak digunakan

dalam domain medis. Sebagai ilustrasi, sebuah sistem ekstraksi informasi dari rekam medis mungkin menggunakan gazetteer berisi nama penyakit (misalnya "pneumonia," "diabetes mellitus," "hipertensi"), nama obat (misalnya "amoksisilin," "paracetamol," "aspirin"), serta daftar singkatan medis (misalnya "BP" untuk blood pressure, "HR" untuk heart rate). Dengan cara ini, sistem dapat lebih mudah mengidentifikasi istilah medis dalam teks naratif dokter yang biasanya penuh variasi.

Kamus domain-specific memiliki fungsi serupa, namun lebih luas dibanding gazetteer. Jika gazetteer biasanya berupa daftar entitas, kamus domain-specific sering kali mencakup definisi, sinonim, serta hubungan antar istilah. Sebagai contoh, dalam kamus hukum, entri "terdakwa" dapat dilengkapi dengan sinonim "tersangka" (dalam konteks awal kasus), atau dalam kamus bisnis, istilah "pendapatan" dapat dihubungkan dengan istilah lain seperti "revenue" atau "income." Integrasi kamus seperti ini ke dalam sistem ekstraksi berbasis aturan akan meningkatkan robustness sistem dalam menghadapi variasi bahasa.

Secara teknis, penerapan gazetteer dan kamus dalam rule-based extraction dapat dilakukan dengan dua cara. Pertama, aturan ekstraksi dapat langsung memeriksa apakah suatu token atau frasa ada dalam daftar entitas. Misalnya, aturan: "Jika sebuah token cocok dengan entri dalam daftar nama provinsi Indonesia, tandai sebagai entitas lokasi." Kedua, aturan dapat menggunakan kombinasi regex dengan lookup ke dalam gazetteer, sehingga sistem tidak hanya mengenali pola karakter tetapi juga mengecek validitas entitas berdasarkan daftar yang ada.

Sebagai contoh implementasi, perhatikan teks

berikut:

"Banjir bandang melanda Kabupaten Cianjur dan Kabupaten Garut pada awal Desember 2022."

Dengan bantuan gazetteer yang berisi daftar kabupaten di Jawa Barat, sistem dapat langsung menandai "Kabupaten Cianjur" dan "Kabupaten Garut" sebagai entitas lokasi, tanpa harus menulis aturan regex yang rumit.

Kelebihan utama penggunaan gazetteer adalah peningkatan akurasi ekstraksi. Sistem menjadi lebih presisi karena hanya mengenali entitas yang telah dikurasi. Selain itu, gazetteer juga mudah diperluas cukup dengan menambahkan entri baru, maka cakupan sistem meningkat tanpa harus mengubah aturan inti. Namun, kelemahan metode ini adalah ketergantungan pada kelengkapan daftar. Jika suatu entitas baru muncul tetapi belum ada dalam gazetteer, sistem kemungkinan gagal mengenalinya. Oleh karena itu, gazetteer memerlukan proses pembaruan berkala agar tetap relevan dengan data terbaru.

Dalam konteks pembelajaran, mahasiswa dapat mencoba membuat gazetteer sederhana sesuai domain yang diminati. Misalnya, untuk domain pariwisata, gazetteer dapat berisi nama-nama objek wisata di Indonesia. Untuk domain kesehatan, gazetteer dapat berupa daftar penyakit umum. Dengan cara ini, mahasiswa tidak hanya memahami teori, tetapi juga dapat langsung merasakan bagaimana penggunaan daftar entitas dapat memperkuat hasil ekstraksi informasi berbasis aturan.

Dengan demikian, penggunaan gazetteer dan kamus domain-specific merupakan langkah penting dalam metode rule-based lanjutan. Teknik ini menjembatani

kelemahan aturan sederhana yang terlalu bergantung pada pola kalimat, dan menjadikan sistem ekstraksi lebih adaptif terhadap variasi penulisan istilah.

B. Alternatif Teknik Ekstraksi Berbasis Aturan

Ekstraksi berbasis aturan tidak hanya terbatas pada pencocokan kata kunci atau penggunaan regex sederhana. Seiring berkembangnya kebutuhan dan kompleksitas teks yang dianalisis, lahirlah berbagai teknik alternatif yang memperkaya pendekatan rule-based agar lebih tangguh menghadapi variasi bahasa. Beberapa di antaranya adalah aturan berbasis dependency parsing, template dan pola tingkat lanjut, pendekatan hybrid rule-statistik, serta pemanfaatan rule engine modern.

Pertama, aturan berbasis dependency parsing. Dependency parsing adalah metode analisis sintaksis yang menekankan pada hubungan antar kata dalam sebuah kalimat, bukan hanya pada urutan kata. Dengan dependency parsing, kita dapat mengetahui siapa yang menjadi subjek, apa predikatnya, siapa objeknya, dan bagaimana kata-kata saling berhubungan. Sebagai contoh, pada kalimat "Korban banjir dievakuasi oleh Tim SAR," dependency parsing akan menghubungkan "Tim SAR" sebagai agen yang melakukan aksi "mengevakuasi," meskipun secara urutan kata, "Tim SAR" muncul di akhir. Dengan demikian, aturan yang berbasis dependency parsing lebih kuat dibanding aturan berbasis pola linear, karena mampu mengenali makna relasional meski struktur kalimat diubah.

Kedua, penggunaan template dan pola tingkat lanjut. Template memungkinkan perumusan aturan dalam bentuk kerangka kalimat dengan slot-slot kosong yang dapat diisi dengan berbagai entitas. Misalnya, template

"[Pejabat] meresmikan [Objek] di [Lokasi] pada [Waktu]" dapat diterapkan pada berbagai kalimat berita politik atau ekonomi, tanpa perlu menulis aturan baru untuk setiap variasi. Teknik ini sangat bermanfaat untuk domain yang memiliki peristiwa berulang, seperti pelantikan, peresmian, bencana alam, atau laporan keuangan.

Ketiga, pendekatan hybrid rule-based dan statistik. Dalam teknik ini, aturan tetap digunakan untuk mengidentifikasi kandidat entitas atau pola, tetapi validasi akhir dilakukan dengan metode statistik atau probabilistik. Misalnya, aturan dapat menandai frasa "162 orang" sebagai kandidat jumlah korban, kemudian model statistik mengevaluasi apakah konteks sekitarnya benar-benar menunjuk pada korban jiwa atau sekadar jumlah relawan. Dengan menggabungkan aturan yang kaku dengan fleksibilitas metode statistik, hasil ekstraksi menjadi lebih akurat dan adaptif terhadap variasi bahasa alami.

Keempat, pemanfaatan rule engine modern. Dalam aplikasi berskala besar, ratusan atau bahkan ribuan aturan harus dikelola sekaligus. Menulis aturan dalam bentuk kode terpisah akan menyulitkan pemeliharaan. Untuk itu, digunakanlah rule engine, yaitu perangkat lunak yang didesain khusus untuk menyimpan, mengelola, dan mengeksekusi aturan secara sistematis. Dengan rule engine seperti Drools, Jess, atau CLIPS, aturan dapat diorganisasi dalam bentuk modul, dilacak sejarah perubahannya, dan dijalankan sesuai kondisi tertentu. Hal ini membuat sistem rule-based lebih skalabel dan layak digunakan dalam aplikasi industri.

Keempat alternatif teknik ini memperlihatkan bahwa rule-based extraction dapat dikembangkan lebih jauh dari sekadar pencarian kata kunci. Dependency parsing memberikan kedalaman sintaksis, template

memberi fleksibilitas pola, hybrid approach menggabungkan presisi aturan dengan adaptasi statistik, sementara rule engine menawarkan pengelolaan aturan dalam skala besar. Dengan memahami berbagai alternatif ini, mahasiswa dan peneliti dapat merancang sistem ekstraksi informasi yang lebih adaptif, presisi, dan sesuai kebutuhan nyata.

C. Pengembangan Rule Adaptif dan Modular

Salah satu tantangan utama dalam penerapan sistem ekstraksi berbasis aturan adalah masalah skalabilitas dan pemeliharaan. Pada tahap awal, menulis aturan sederhana untuk mengekstrak informasi dari teks mungkin terasa mudah. Namun, ketika jumlah aturan bertambah hingga ratusan atau ribuan, sistem menjadi sulit dikendalikan. Sering kali aturan yang baru ditambahkan dapat menimbulkan konflik dengan aturan yang lama, atau malah menghasilkan duplikasi fungsi. Untuk mengatasi masalah ini, dibutuhkan pendekatan rule adaptif dan modular, di mana aturan dapat diorganisasi, diperbarui, dan disesuaikan dengan kebutuhan tanpa harus merombak keseluruhan sistem.

Rule adaptif berarti aturan yang dapat menyesuaikan diri terhadap variasi bahasa dan perubahan domain. Dalam praktiknya, aturan tidak ditulis secara kaku, melainkan dilengkapi dengan mekanisme generalisasi atau parameterisasi. Sebagai contoh, aturan untuk mengekstrak jumlah korban dalam berita bencana tidak hanya ditulis untuk pola "[angka] orang meninggal dunia", tetapi juga diperluas agar mencakup variasi seperti "[angka] jiwa tewas" atau "korban meninggal mencapai [angka] orang." Dengan membuat aturan yang adaptif, sistem mampu menghadapi variasi bahasa tanpa

harus menulis ulang aturan dari awal.

Selain itu, adaptivitas dapat dicapai melalui pembelajaran semi-otomatis. Sistem dapat mengumpulkan contoh-contoh baru dari teks yang belum dikenali oleh aturan yang ada, kemudian memberikan saran kepada pengembang untuk memperbarui aturan. Misalnya, jika sistem menemukan frasa baru seperti "ratusan jiwa kehilangan nyawa," ia dapat memberi peringatan bahwa pola ini serupa dengan aturan jumlah korban, sehingga aturan lama bisa diperluas. Cara ini membuat sistem lebih dinamis dan mampu mengikuti perkembangan bahasa.

Sementara itu, rule modular berfokus pada bagaimana aturan diorganisasi dan dikelompokkan. Alih-alih menulis semua aturan dalam satu blok besar, aturan sebaiknya dipisahkan ke dalam modul-modul berdasarkan fungsinya. Misalnya, dalam sistem ekstraksi berita bencana, aturan dapat dibagi menjadi modul "deteksi jenis bencana," "ekstraksi lokasi," "ekstraksi waktu," dan "ekstraksi korban." Dengan pendekatan modular, setiap bagian dapat dikelola secara terpisah. Jika terjadi kesalahan pada modul "ekstraksi lokasi," pengembang cukup memperbaiki modul tersebut tanpa mengganggu aturan lain.

Modularisasi juga memungkinkan reuse (penggunaan kembali) aturan. Aturan untuk mendeteksi tanggal, misalnya, tidak hanya berguna pada berita bencana, tetapi juga dapat dipakai pada dokumen hukum atau laporan ekonomi. Dengan menempatkan aturan tanggal dalam modul tersendiri, aturan tersebut dapat digunakan lintas domain tanpa perlu ditulis ulang.

Selain keuntungan dalam pemeliharaan, modularisasi mendukung penggunaan rule engine

modern. Rule engine memungkinkan aturan disimpan dalam basis data atau file konfigurasi terpisah dari kode program utama. Dengan begitu, penambahan atau penghapusan aturan bisa dilakukan tanpa harus mengubah kode. Hal ini sangat membantu dalam aplikasi industri yang membutuhkan fleksibilitas tinggi.

Dengan menggabungkan sifat adaptif dan modular, sistem rule-based dapat berkembang menjadi lebih tahan lama, fleksibel, dan mudah dipelihara. Mahasiswa perlu memahami bahwa menulis aturan bukan hanya soal mengenali pola, tetapi juga soal bagaimana aturan dikelola agar tetap relevan dalam jangka panjang. Dalam praktiknya, pengembangan rule adaptif dan modular merupakan langkah transisi menuju sistem ekstraksi informasi yang lebih canggih, bahkan dapat menjadi fondasi untuk sistem hibrida yang menggabungkan aturan dengan machine learning.

D. Integrasi Rule dengan NLP Pipeline

Dalam pengembangan sistem ekstraksi informasi modern, aturan (rules) jarang berdiri sendiri. Sebaliknya, aturan biasanya diintegrasikan ke dalam sebuah pipeline NLP yang terdiri atas berbagai tahapan pemrosesan bahasa alami, mulai dari preprocessing teks, tokenisasi, part-of-speech tagging, parsing, hingga analisis semantik. Dengan integrasi ini, aturan dapat bekerja lebih efektif karena tidak lagi beroperasi pada teks mentah, melainkan pada representasi linguistik yang lebih terstruktur.

Pipeline NLP dapat dianggap sebagai jalur pemrosesan bertingkat. Pada tahap awal, teks dibersihkan dari elemen-elemen yang tidak relevan seperti tanda baca berlebih, huruf kapital yang tidak konsisten, atau simbol-simbol khusus. Selanjutnya, tokenisasi memecah teks

menjadi unit kata atau frasa, yang kemudian diberi label kelas kata (POS tagging). Informasi ini menjadi penting karena banyak aturan dalam ekstraksi informasi ditulis berdasarkan kategori kata. Misalnya, sebuah aturan dapat berbunyi: "Jika ada kata kerja (VB) yang diikuti kata benda (NN) dari daftar gazetteer lokasi, maka tandai sebagai aksi melibatkan lokasi." Tanpa adanya informasi POS, aturan semacam ini sulit diterapkan secara konsisten.

Tahap berikutnya dalam pipeline adalah parsing, baik dalam bentuk constituency parsing maupun dependency parsing. Parsing memberikan struktur hierarkis atau relasional yang menunjukkan bagaimana kata-kata dalam kalimat saling terhubung. Integrasi aturan dengan hasil parsing membuat sistem dapat mengekstrak informasi dengan lebih presisi. Sebagai contoh, pada kalimat "Korban banjir dievakuasi oleh Tim SAR," parser akan menunjukkan hubungan antara kata "dievakuasi" dengan "korban banjir" sebagai objek dan "Tim SAR" sebagai subjek. Dengan informasi ini, aturan tidak lagi perlu mengandalkan urutan kata, melainkan dapat memanfaatkan relasi gramatikal yang lebih stabil.

Selain itu, aturan juga dapat diintegrasikan pada tahap Named Entity Recognition (NER) dalam pipeline NLP. NER bertugas mengenali entitas seperti nama orang, organisasi, lokasi, atau waktu. Aturan kemudian dapat digunakan untuk memperhalus hasil NER, misalnya dengan menambahkan filter berdasarkan domain tertentu. Sebagai contoh, jika NER mendeteksi "Bali" sebagai lokasi, aturan dapat mempertegasnya bahwa dalam konteks berita pariwisata, "Bali" juga merupakan destinasi wisata, bukan sekadar entitas geografis. Dengan demikian, integrasi rule dan NER menciptakan hasil ekstraksi yang lebih kontekstual.

Dalam praktiknya, integrasi aturan dengan pipeline NLP biasanya dilakukan melalui kerangka kerja (framework) atau pustaka yang mendukung tahapan berlapis. Beberapa contoh populer adalah spaCy, Stanford CoreNLP, dan NLTK. Framework ini memungkinkan pengembang menambahkan modul aturan setelah tahap tertentu. Misalnya, setelah spaCy menyelesaikan tokenisasi, POS tagging, dan dependency parsing, aturan berbasis pola dapat dijalankan untuk mengekstrak hubungan subjek–predikat–objek dari teks.

Keunggulan utama dari integrasi ini adalah peningkatan akurasi dan efisiensi. Aturan yang berdiri sendiri pada teks mentah sering kali menghasilkan banyak kesalahan karena variasi bahasa. Namun, jika aturan bekerja pada hasil pipeline NLP, ruang kemungkinan sudah lebih terstruktur sehingga aturan lebih mudah diterapkan. Misalnya, aturan untuk mengekstrak tanggal bisa lebih akurat jika diterapkan setelah proses normalisasi temporal yang sudah menstandarkan "Senin pagi" menjadi "21 November 2022."

Namun, integrasi aturan dengan pipeline NLP juga menghadapi beberapa tantangan. Pertama, ketergantungan pada kualitas pipeline. Jika POS tagging atau parsing menghasilkan kesalahan, maka aturan yang bergantung padanya juga berisiko salah. Kedua, integrasi sering kali meningkatkan kebutuhan komputasi, karena setiap tahap pipeline membutuhkan waktu dan sumber daya. Oleh karena itu, pengembang perlu menyeimbangkan antara kedalaman analisis dan efisiensi sistem.

Secara keseluruhan, integrasi rule dengan NLP pipeline menjadikan sistem ekstraksi informasi lebih cerdas, presisi, dan adaptif. Aturan tidak lagi menjadi

instruksi kaku yang hanya bekerja pada pola permukaan, melainkan dapat memanfaatkan representasi linguistik yang kaya untuk mengidentifikasi entitas, relasi, dan peristiwa dengan lebih baik. Mahasiswa diharapkan memahami bahwa kekuatan sejati rule-based extraction terletak bukan hanya pada penulisan aturan itu sendiri, melainkan pada bagaimana aturan tersebut dihubungkan dengan pipeline NLP yang menyediakannya data linguistik yang lebih siap diproses.

E. Ekstraksi Entitas Khusus

Dalam ekstraksi informasi, sering kali kita tidak hanya berhadapan dengan entitas umum seperti nama orang, lokasi, organisasi, atau tanggal, melainkan juga dengan entitas khusus yang sangat spesifik pada suatu domain tertentu. Entitas semacam ini disebut entitas khusus (specialized entities) dan biasanya memiliki nilai informasi yang tinggi bagi analisis. Contohnya adalah magnitudo gempa dalam domain kebencanaan, kode ICD pada domain medis, nomor pasal dalam domain hukum, atau simbol saham dalam domain keuangan. Untuk memperjelas bagaimana metode rule-based lanjutan dapat digunakan, mari kita lihat sebuah studi kasus pada domain kebencanaan.

Perhatikan kalimat berita berikut: "Gempa bumi berkekuatan 6,2 SR melanda Kabupaten Cianjur pada 21 November 2022. Menurut data BNPB, sebanyak 162 orang meninggal dunia, sementara 13.000 rumah warga rusak." Kalimat ini mengandung berbagai informasi penting yang bisa diekstrak menjadi entitas khusus, yaitu jenis bencana (gempa bumi), magnitudo gempa (6,2 SR), lokasi kejadian (Kabupaten Cianjur), tanggal kejadian (21 November 2022), jumlah korban jiwa (162 orang meninggal dunia), serta

jumlah rumah rusak (13.000 rumah rusak).

Untuk mengenali entitas-entitas tersebut, sistem rule-based dapat menggunakan aturan yang ditulis secara eksplisit. Misalnya, keberadaan kata "gempa," "banjir," atau "longsor" dapat ditandai sebagai jenis bencana. Pola angka yang diikuti satuan "SR" dapat dikenali sebagai magnitudo gempa dengan memanfaatkan ekspresi reguler sederhana. Lokasi dapat diidentifikasi melalui gabungan aturan sintaksis dengan daftar gazetteer berisi nama kabupaten atau kota di Indonesia, khususnya jika didahului oleh kata kerja seperti "melanda" atau "mengguncang." Tanggal dapat dikenali melalui pola penulisan umum seperti "21 November 2022," sementara jumlah korban dapat diambil dari frasa yang mengandung angka diikuti kata "orang meninggal dunia," "orang luka-luka," atau "jiwa tewas." Aturan serupa juga dapat diterapkan pada frasa "rumah rusak" untuk mengekstrak informasi kerusakan infrastruktur.

Ketika aturan tersebut diterapkan pada teks, sistem dapat secara otomatis menghasilkan ekstraksi yang terstruktur. Dari kalimat di atas, diperoleh informasi bahwa jenis bencana adalah gempa bumi, magnitudo 6,2 SR, lokasi di Kabupaten Cianjur, waktu kejadian pada 21 November 2022, jumlah korban meninggal sebanyak 162 orang, serta kerusakan mencapai 13.000 rumah. Informasi ini kemudian dapat disajikan kembali dalam bentuk tabel atau format terstruktur seperti JSON agar lebih mudah diproses oleh sistem tanggap bencana maupun basis data.

Studi kasus ini memperlihatkan bagaimana aturan sederhana tetapi terarah dapat dimanfaatkan untuk mengenali entitas khusus yang penting bagi suatu domain. Dalam konteks kebencanaan, informasi seperti jumlah korban, kerusakan, lokasi, dan magnitudo sangat vital

untuk perencanaan dan pengambilan keputusan. Namun, studi kasus ini juga menegaskan keterbatasan pendekatan rule-based. Jika penulisan berita berbeda, misalnya menggunakan ungkapan "ribuan jiwa kehilangan nyawa" alih-alih "162 orang meninggal dunia," maka aturan lama mungkin tidak mampu mengenalinya. Hal ini menunjukkan bahwa aturan harus terus diperbarui dan diperluas agar dapat menangkap variasi bahasa yang lebih luas.

Dalam konteks pembelajaran, penting bagi mahasiswa untuk memahami bahwa konsep ekstraksi entitas khusus ini dapat diterapkan di berbagai domain lain. Pada domain medis, misalnya, aturan dapat ditulis untuk mengenali nama penyakit, obat, dan dosis. Pada domain hukum, aturan dapat digunakan untuk mendeteksi nomor pasal, nama pihak dalam perjanjian, atau tanggal berlaku kontrak. Dengan latihan menulis aturan seperti ini, mahasiswa tidak hanya belajar teori rule-based extraction, tetapi juga dapat mempraktikkannya langsung dalam kasus nyata sesuai bidang yang diminati.

Latihan

A. Pertanyaan Pemahaman Konsep

1. Apa perbedaan antara *Sentiment Analysis* dan *Opinion Mining*?
2. Jelaskan tiga kategori polaritas utama dalam analisis sentimen.
3. Apa kelebihan dan kekurangan pendekatan berbasis leksikon dibanding berbasis pembelajaran mesin?
4. Mengapa preprocessing teks sangat penting dalam analisis sentimen media sosial?
5. Jelaskan apa yang dimaksud dengan *sentiment lexicon* dan berikan contohnya dalam Bahasa Indonesia.
6. Bagaimana kita menilai kualitas hasil analisis sentimen secara kuantitatif?
7. Sebutkan tiga tantangan unik dalam analisis sentimen teks Bahasa Indonesia.

B. Latihan Praktik Sederhana

Diberikan tiga contoh ulasan (review) dari media sosial berikut:

1. "Pelayanannya cepat dan ramah, saya sangat puas!"
2. "Harga mahal, tapi kualitas makanannya tidak sesuai."
3. "Tempatnya biasa saja, tapi pemandangannya indah."

Tugas:

1. Tentukan polaritas sentimen (positif, negatif, netral) dari setiap kalimat.
2. Identifikasi kata-kata yang menjadi indikator sentimen.
3. Sajikan hasil dalam tabel berikut:

Teks Ulasan	Kata Kunci Sentimen	Polaritas	Keterangan
Pelayanannya cepat dan ramah	cepat, ramah, puas	Positif	Ulasan pelanggan senang
Harga mahal, kualitas tidak sesuai	mahal, tidak sesuai	Negatif	Keluhan pelanggan
Tempatnya biasa saja, pemandangannya indah	biasa, indah	Netral–positif	Ada keseimbangan opini

4. Jelaskan kemungkinan kesalahan yang dapat terjadi jika sistem tidak mampu mengenali konteks negasi ("tidak puas", "kurang baik").

C. Studi Kasus / Proyek Mini

Anda akan membangun sistem analisis sentimen otomatis untuk ulasan publik tentang pelayanan transportasi kota.

1. Tentukan sumber data (misalnya ulasan Google Maps, Twitter, atau survei pengguna).
2. Rancang pipeline sistem: *data collection* → *preprocessing* → *feature extraction* → *sentiment classification* → *visualization*.
3. Pilih pendekatan yang digunakan: *lexicon-based* atau *machine learning*.
4. Gunakan minimal lima contoh teks dan tampilkan hasil analisis sentimennya.
5. Diskusikan bagaimana hasil analisis ini dapat digunakan oleh pemerintah atau perusahaan untuk perbaikan layanan publik.

D. Diskusi / Refleksi

1. Menurut Anda, apakah sentimen publik selalu akurat mencerminkan realitas?
2. Bagaimana penggunaan *emoji*, *slang*, dan campuran bahasa memengaruhi akurasi analisis sentimen?
3. Diskusikan penerapan analisis sentimen dalam konteks akademik, misalnya untuk memantau persepsi mahasiswa terhadap pembelajaran daring.
4. Menurut Anda, apakah sistem analisis sentimen sebaiknya bersifat transparan (*interpretable*) atau cukup berbasis *black-box model*?

BAB 10
PENGUJIAN EKSTRAKSI BERBASIS RULE

Tujuan Pembelajaran

Setelah mempelajari Bab 10, mahasiswa diharapkan mampu:

1. Menjelaskan konsep arsitektur dan alur kerja (*pipeline*) sistem *Information Extraction (IE)* secara utuh.
2. Mengidentifikasi komponen-komponen utama dalam pipeline IE, seperti preprocessing, POS tagging, NER, relation extraction, event extraction, dan klasifikasi teks.
3. Menjelaskan bagaimana tiap komponen saling berinteraksi untuk menghasilkan informasi terstruktur dari teks mentah.
4. Merancang model pipeline sederhana untuk ekstraksi informasi menggunakan contoh kasus nyata.
5. Menganalisis kelebihan dan keterbatasan pipeline IE tradisional dibanding pendekatan modern berbasis *end-to-end deep learning*.
6. Menerapkan konsep integrasi modular dalam sistem IE berbasis Bahasa Indonesia.
7. Mengevaluasi performa keseluruhan sistem IE berdasarkan akurasi, kecepatan, dan ketepatan hasil.

Setelah mempelajari dasar-dasar hingga metode lanjutan dalam ekstraksi berbasis aturan (rule-based extraction), tahap berikutnya yang tidak kalah penting adalah pengujian dan evaluasi sistem. Sebuah sistem ekstraksi informasi tidak cukup hanya dapat mengekstrak entitas atau relasi, tetapi juga harus dibuktikan kinerjanya melalui ukuran yang terstandar. Evaluasi ini diperlukan agar kita mengetahui

sejauh mana aturan yang dibuat mampu bekerja secara akurat pada data nyata, serta untuk mengidentifikasi bagian mana yang perlu diperbaiki.

Pengujian dalam ekstraksi berbasis aturan memiliki tujuan utama untuk memastikan keandalan, akurasi, dan konsistensi hasil ekstraksi. Keandalan berarti sistem dapat bekerja dengan stabil pada teks yang bervariasi, akurasi berarti hasil ekstraksi mendekati kebenaran (ground truth), dan konsistensi berarti aturan dapat digunakan berulang kali tanpa menghasilkan hasil yang berbeda-beda untuk pola yang sama.

Dalam praktiknya, evaluasi sistem ekstraksi informasi biasanya dilakukan dengan membandingkan hasil ekstraksi sistem terhadap data referensi yang sudah diberi label secara manual oleh pakar atau anotator. Dari perbandingan ini, dihitung berbagai ukuran evaluasi seperti precision, recall, dan F1 score, yang telah menjadi standar dalam penelitian maupun aplikasi NLP. Selain itu, teknik validasi seperti cross-validation atau pembagian data latih–uji juga dapat digunakan untuk menguji robustnes sistem.

Lebih jauh lagi, evaluasi tidak hanya berhenti pada angka, tetapi juga perlu divisualisasikan agar mudah dipahami. Misalnya, dengan menampilkan distribusi kesalahan, grafik precision–recall, atau tabel perbandingan hasil ekstraksi antar domain. Visualisasi ini membantu peneliti, mahasiswa, maupun pengguna sistem untuk menafsirkan kelemahan dan kekuatan sistem secara lebih intuitif.

Pada bab ini, kita akan membahas prinsip evaluasi hasil ekstraksi, cara menghitung ukuran evaluasi standar seperti precision, recall, dan F1 score, teknik validasi dengan ground truth, serta bagaimana hasil evaluasi dapat divisualisasikan untuk interpretasi lebih lanjut. Sebagai penutup, akan

diberikan contoh evaluasi sederhana dari sistem rule-based extraction yang telah kita bahas sebelumnya, agar mahasiswa dapat melihat hubungan antara teori dan implementasi nyata.

A. Evaluasi Hasil Ekstraksi

Evaluasi merupakan salah satu tahapan penting dalam pembangunan sistem ekstraksi informasi berbasis aturan. Tanpa adanya evaluasi yang jelas, sulit untuk menilai apakah sistem yang dikembangkan benar-benar mampu memberikan hasil yang bermanfaat atau justru menghasilkan informasi yang keliru. Evaluasi tidak hanya berfungsi untuk mengukur performa sistem, tetapi juga sebagai sarana refleksi guna memperbaiki aturan yang sudah dibuat, memperluas cakupan, serta memastikan bahwa sistem dapat diandalkan dalam berbagai kondisi data.

Tujuan utama evaluasi dalam ekstraksi informasi dapat dibagi menjadi beberapa aspek. Pertama adalah menilai akurasi sistem, yakni sejauh mana hasil ekstraksi mendekati kebenaran atau ground truth yang sudah ditentukan. Sebagai contoh, jika sistem diminta mengekstrak jumlah korban dari berita bencana, evaluasi bertujuan memastikan apakah angka yang diekstrak sama dengan data yang sebenarnya terdapat dalam teks. Kedua adalah mengetahui kelemahan sistem, misalnya aturan mana yang sering gagal menangkap informasi, atau pola kalimat seperti apa yang belum tercakup dalam aturan. Dengan memahami kelemahan ini, pengembang dapat merancang strategi perbaikan yang lebih tepat. Ketiga adalah membandingkan performa antar pendekatan, baik antara rule-based dengan model berbasis pembelajaran mesin, maupun antarvariasi aturan dalam sistem rule-based itu sendiri. Dengan adanya perbandingan, kita

dapat memilih pendekatan yang paling sesuai untuk kebutuhan tertentu.

Dalam melaksanakan evaluasi, terdapat beberapa prinsip yang perlu diperhatikan. Prinsip pertama adalah objektivitas. Evaluasi harus dilakukan berdasarkan data yang jelas dan dapat diverifikasi, bukan sekadar persepsi subjektif pengembang. Untuk itu, dibutuhkan ground truth atau data referensi yang sudah dianotasi secara manual oleh pakar atau anotator terlatih. Dengan adanya ground truth, hasil ekstraksi sistem dapat dibandingkan secara obyektif.

Prinsip kedua adalah konsistensi. Evaluasi harus menggunakan metode pengukuran yang konsisten agar hasilnya dapat dibandingkan dari waktu ke waktu maupun antar sistem. Misalnya, jika satu sistem diuji dengan ukuran precision, recall, dan F1 score, maka sistem lain yang akan dibandingkan sebaiknya diuji dengan ukuran yang sama. Konsistensi ini penting agar hasil evaluasi benar-benar dapat dijadikan dasar pengambilan keputusan.

Prinsip ketiga adalah keterulangan (repeatability). Evaluasi yang baik harus dapat diulang dengan hasil yang serupa jika kondisi data dan aturan tidak berubah. Hal ini memastikan bahwa hasil yang diperoleh bukan sekadar kebetulan, melainkan mencerminkan kinerja sistem secara nyata.

Selain itu, prinsip penting lainnya adalah relevansi. Evaluasi harus sesuai dengan tujuan penggunaan sistem. Jika sistem ditujukan untuk mendukung pengambilan keputusan darurat dalam bencana, maka kecepatan ekstraksi mungkin lebih penting daripada cakupan penuh. Sebaliknya, jika sistem digunakan untuk analisis hukum, akurasi mutlak pada setiap pasal dan entitas lebih

diutamakan meskipun waktu pemrosesan lebih lama. Dengan kata lain, evaluasi harus mempertimbangkan konteks aplikasi.

Dengan memahami tujuan dan prinsip evaluasi ini, mahasiswa diharapkan dapat melihat bahwa pengujian bukanlah sekadar formalitas, melainkan fondasi penting yang menentukan nilai praktis dari sistem ekstraksi informasi. Sebuah sistem rule-based yang memiliki aturan banyak sekalipun tidak akan berarti jika tidak dibuktikan efektivitasnya melalui evaluasi yang sistematis, objektif, dan konsisten.

B. Pengukuran: Precision, Recall, dan F1 Score

Dalam evaluasi sistem ekstraksi informasi, terdapat ukuran-ukuran standar yang umum digunakan untuk menilai performa sistem. Tiga ukuran yang paling penting dan sering dipakai adalah precision, recall, dan F1 score. Ketiganya saling melengkapi dalam memberikan gambaran tentang kualitas hasil ekstraksi, sehingga hampir selalu digunakan bersama dalam penelitian maupun aplikasi praktis.

Precision adalah ukuran yang menunjukkan seberapa besar proporsi hasil ekstraksi yang benar dari seluruh hasil yang diambil oleh sistem. Dengan kata lain, precision menjawab pertanyaan: "Dari semua informasi yang diekstrak sistem, berapa banyak yang memang benar sesuai dengan ground truth?" Precision tinggi berarti sistem jarang membuat kesalahan dengan mengekstrak informasi yang tidak relevan. Misalnya, jika sistem mengekstrak 100 entitas dan 85 di antaranya benar, maka precision sistem adalah 85%. Precision sangat penting dalam aplikasi yang membutuhkan keakuratan absolut, misalnya analisis hukum atau diagnosis medis, di mana

kesalahan sekecil apa pun dapat berdampak besar.

Recall, di sisi lain, adalah ukuran yang menunjukkan seberapa besar proporsi informasi yang benar-benar berhasil diekstrak oleh sistem dari seluruh informasi yang ada dalam data. Recall menjawab pertanyaan: "Dari semua informasi yang seharusnya diekstrak, berapa banyak yang berhasil ditangkap oleh sistem?" Recall tinggi berarti sistem jarang melewatkan informasi penting. Sebagai contoh, jika terdapat 200 entitas dalam teks dan sistem berhasil mengekstrak 160 di antaranya, maka recall sistem adalah 80%. Recall menjadi sangat krusial pada aplikasi yang berfokus pada cakupan luas, seperti sistem tanggap darurat bencana, di mana melewatkan informasi tentang korban atau lokasi dapat berakibat fatal.

F1 Score adalah ukuran yang menggabungkan precision dan recall ke dalam satu nilai tunggal. Karena dalam banyak kasus terdapat trade-off antara precision dan recall, F1 score digunakan untuk menyeimbangkan keduanya. Secara matematis, F1 score adalah rata-rata harmonis antara precision dan recall, sehingga nilainya hanya akan tinggi jika kedua ukuran tersebut sama-sama tinggi. Sebagai contoh, jika precision sebuah sistem adalah 90% tetapi recall hanya 50%, maka F1 score tidak akan setinggi 70%, melainkan lebih rendah karena recall yang rendah sangat memengaruhi nilai rata-rata harmonis. Dengan demikian, F1 score sering dianggap sebagai ukuran yang paling adil dalam menggambarkan performa keseluruhan sistem ekstraksi.

Untuk memperjelas, bayangkan sebuah sistem rule-based yang dirancang untuk mengekstrak jumlah korban dari berita bencana. Dalam 50 artikel berita, ground truth mencatat terdapat 100 entitas jumlah korban. Sistem kemudian berhasil mengekstrak 90 entitas, namun hanya

70 di antaranya yang benar, sementara 20 lainnya salah interpretasi. Dari skenario ini, precision dihitung sebagai 70/90 = 77,8%, recall sebagai 70/100 = 70%, dan F1 score sebagai 2 × (0,778 × 0,7) / (0,778 + 0,7) ≈ 73,7%. Nilai ini menunjukkan bahwa sistem cukup baik, namun masih perlu ditingkatkan terutama pada aspek recall agar lebih banyak informasi penting yang berhasil ditangkap.

Ketiga ukuran ini juga membantu dalam menganalisis kekuatan dan kelemahan sistem. Precision tinggi tetapi recall rendah menunjukkan bahwa aturan sistem terlalu ketat, sehingga hanya mengenali informasi yang sangat jelas namun melewatkan banyak variasi. Sebaliknya, recall tinggi tetapi precision rendah berarti aturan sistem terlalu longgar, sehingga banyak menangkap informasi namun sebagian besar salah. Dengan memahami keseimbangan ini, pengembang dapat menyesuaikan aturan agar lebih optimal.

Dengan demikian, precision, recall, dan F1 score bukan hanya angka, tetapi cermin dari bagaimana aturan dalam sistem bekerja. Mahasiswa perlu menguasai konsep ini karena ketiganya menjadi standar internasional dalam evaluasi sistem ekstraksi informasi, baik berbasis rule maupun berbasis machine learning. Pemahaman yang mendalam akan membantu mahasiswa tidak hanya dalam mengukur, tetapi juga dalam memperbaiki kualitas sistem yang dikembangkan.

C. Teknik Validasi dan Ground Truth

Evaluasi sistem ekstraksi informasi tidak dapat dilepaskan dari konsep ground truth, yaitu data rujukan yang dianggap benar dan dijadikan standar pembanding. Ground truth biasanya diperoleh melalui proses anotasi manual yang dilakukan oleh pakar atau anotator terlatih.

Misalnya, dalam penelitian kebencanaan, anotator membaca berita lalu menandai entitas penting seperti jenis bencana, lokasi, waktu, jumlah korban, dan kerusakan. Hasil anotasi ini menjadi dasar untuk mengevaluasi apakah sistem rule-based berhasil mengekstrak informasi dengan benar.

Ground truth sangat penting karena tanpa data referensi, evaluasi tidak bisa dilakukan secara objektif. Namun, membangun ground truth bukanlah hal yang sederhana. Proses anotasi sering kali memakan waktu, membutuhkan konsistensi antar anotator, dan memerlukan pedoman yang jelas. Misalnya, dalam anotasi korban bencana, apakah frasa "ratusan orang" dianggap valid sebagai entitas jumlah korban, atau hanya frasa yang menyebut angka spesifik seperti "162 orang" yang dilabeli? Perbedaan interpretasi ini bisa menimbulkan inkonsistensi jika tidak diatur dengan baik. Oleh karena itu, dalam praktiknya, anotasi ground truth biasanya disertai dengan guideline anotasi yang menjelaskan aturan baku.

Selain membangun ground truth, evaluasi juga memerlukan teknik validasi yang tepat. Salah satu teknik yang sering digunakan adalah hold-out validation, di mana dataset dibagi menjadi dua bagian: data uji (test set) dan data latih (training set). Meskipun dalam sistem rule-based tidak selalu ada proses pelatihan seperti pada machine learning, pembagian data ini tetap penting untuk memastikan bahwa aturan yang dibuat tidak hanya berlaku pada teks yang sudah dilihat sebelumnya, melainkan juga dapat diterapkan pada data baru.

Teknik lain yang lebih komprehensif adalah cross-validation, khususnya k-fold cross-validation. Dalam metode ini, dataset dibagi menjadi k lipatan, misalnya 5 atau 10. Setiap kali, satu lipatan digunakan sebagai data uji,

sedangkan sisanya digunakan untuk pengembangan atau perbaikan aturan. Proses ini diulang hingga semua lipatan bergantian menjadi data uji. Dengan cara ini, sistem diuji secara menyeluruh sehingga hasil evaluasi lebih stabil dan tidak bergantung pada satu pembagian data saja.

Selain validasi berbasis pembagian data, dalam sistem rule-based juga dikenal teknik error analysis, yaitu analisis terhadap kesalahan sistem dalam mengekstrak informasi. Kesalahan bisa berupa false positive (sistem menandai sesuatu sebagai entitas padahal bukan) maupun false negative (sistem gagal mendeteksi entitas yang seharusnya ada). Analisis kesalahan ini sangat penting karena membantu pengembang memahami kelemahan aturan. Misalnya, jika banyak kesalahan muncul pada frasa "ratusan rumah rusak" yang tidak terdeteksi oleh aturan, maka aturan perlu diperluas agar mengenali frasa dengan kata "ratusan."

Dalam implementasi praktis, validasi juga dapat dilakukan dengan melibatkan beberapa anotator untuk memastikan konsistensi. Jika dua atau lebih anotator memberikan hasil anotasi yang sama pada teks tertentu, maka ground truth dianggap lebih dapat diandalkan. Untuk mengukur tingkat kesepakatan antar anotator, biasanya digunakan metrik seperti Cohen's Kappa atau Krippendorff's Alpha. Nilai ini menunjukkan sejauh mana anotator konsisten satu sama lain, sehingga ground truth yang dihasilkan benar-benar bisa dijadikan acuan.

Dengan adanya ground truth dan teknik validasi yang tepat, evaluasi sistem rule-based dapat dilakukan secara lebih terukur, konsisten, dan dapat dipertanggungjawabkan. Mahasiswa perlu memahami bahwa pengujian bukan sekadar menghitung precision dan recall, melainkan juga memastikan bahwa data

rujukan yang digunakan dalam evaluasi memiliki kualitas tinggi. Evaluasi yang dilakukan dengan data ground truth yang tidak konsisten atau tanpa validasi yang baik akan menghasilkan gambaran yang menyesatkan tentang kinerja sistem.

D. Visualisasi dan Interpretasi Hasil Evaluasi

Pengujian sistem ekstraksi informasi tidak hanya berhenti pada angka-angka hasil pengukuran seperti precision, recall, dan F1 score. Angka-angka tersebut memang memberikan gambaran kuantitatif tentang performa sistem, namun sering kali sulit untuk segera dipahami oleh pengguna, terutama mereka yang bukan berasal dari latar belakang teknis. Oleh karena itu, visualisasi hasil evaluasi menjadi bagian penting untuk menyajikan informasi evaluasi dalam bentuk yang lebih intuitif, mudah dibaca, dan dapat langsung ditafsirkan untuk pengambilan keputusan.

Salah satu bentuk visualisasi yang umum digunakan adalah confusion matrix, yang memperlihatkan jumlah prediksi benar dan salah dalam bentuk tabel dua dimensi. Dalam konteks ekstraksi informasi, confusion matrix membantu menunjukkan jumlah entitas yang berhasil dikenali dengan benar (true positive), jumlah entitas yang salah dikenali (false positive), jumlah entitas yang terlewat (false negative), serta jumlah teks yang benar-benar tidak mengandung entitas (true negative). Dengan melihat confusion matrix, pengembang dapat langsung mengetahui di mana kesalahan sistem paling banyak terjadi. Misalnya, jika angka false negative tinggi, berarti banyak informasi penting yang terlewat dan aturan perlu diperluas.

Selain confusion matrix, visualisasi lain yang sering

digunakan adalah grafik precision–recall. Grafik ini menggambarkan trade-off antara precision dan recall. Pada banyak sistem ekstraksi, meningkatkan recall sering kali menurunkan precision, dan sebaliknya. Dengan memvisualisasikan hubungan ini, pengembang dapat menentukan titik keseimbangan yang paling sesuai dengan kebutuhan aplikasi. Misalnya, dalam sistem ekstraksi untuk bencana, recall tinggi mungkin lebih diprioritaskan agar informasi penting tidak terlewat, meskipun precision sedikit berkurang.

Visualisasi lain yang bermanfaat adalah diagram distribusi kesalahan, yang menampilkan jenis-jenis kesalahan apa saja yang paling sering terjadi. Misalnya, kesalahan dalam mengenali lokasi, tanggal, atau jumlah korban. Dengan diagram ini, pengembang dapat lebih mudah fokus memperbaiki aturan pada area yang paling bermasalah. Sebagai contoh, jika kesalahan paling banyak terjadi pada variasi penulisan tanggal, maka aturan regex untuk tanggal perlu diperluas agar mencakup lebih banyak format.

Interpretasi hasil evaluasi juga tidak boleh hanya sebatas membaca nilai metrik atau grafik, tetapi perlu dikaitkan dengan konteks aplikasi. Sistem yang digunakan untuk analisis hukum memerlukan precision yang sangat tinggi karena kesalahan sekecil apa pun bisa menimbulkan dampak serius. Sebaliknya, sistem untuk analisis media sosial mungkin lebih menekankan recall, karena cakupan informasi yang luas lebih penting daripada akurasi absolut pada setiap entitas. Dengan demikian, interpretasi hasil evaluasi harus mempertimbangkan kebutuhan dan prioritas pengguna akhir.

Lebih jauh lagi, visualisasi juga dapat membantu dalam proses komunikasi antara pengembang sistem

dengan pihak non-teknis, seperti pengambil kebijakan, praktisi hukum, atau tenaga medis. Angka-angka seperti precision 82% atau recall 75% mungkin sulit dipahami, tetapi dengan grafik dan ilustrasi, mereka dapat dengan mudah melihat bahwa sistem cukup baik dalam mengenali pola tertentu namun masih lemah dalam aspek lain. Visualisasi ini membuat hasil evaluasi lebih transparan, dapat dipertanggungjawabkan, sekaligus lebih mudah diterima.

Dengan demikian, visualisasi dan interpretasi bukan hanya pelengkap, melainkan bagian integral dalam evaluasi sistem ekstraksi berbasis aturan. Mahasiswa perlu memahami bahwa hasil evaluasi harus dapat "berbicara" dengan jelas, tidak hanya kepada peneliti atau pengembang, tetapi juga kepada pemangku kepentingan yang akan menggunakan sistem tersebut. Hanya dengan cara inilah, hasil evaluasi benar-benar bisa dimanfaatkan untuk perbaikan sistem dan penerapan nyata di lapangan.

E. Evaluasi Sistem Ekstraksi

Untuk memberikan gambaran nyata tentang bagaimana evaluasi dilakukan dalam sistem ekstraksi berbasis aturan, mari kita perhatikan sebuah contoh sederhana pada domain berita kebencanaan. Misalkan kita memiliki sistem rule-based yang dirancang untuk mengekstrak entitas berupa jenis bencana, lokasi, tanggal, jumlah korban jiwa, dan jumlah kerusakan rumah dari teks berita.

Sebagai bahan uji, digunakan 50 artikel berita bencana dari berbagai media daring. Dari artikel tersebut, anotator manusia membuat ground truth yang berisi semua entitas penting. Misalnya, pada sebuah artikel, ground truth mencatat informasi "Gempa bumi,"

“Kabupaten Cianjur,” “21 November 2022,” “162 orang meninggal dunia,” dan “13.000 rumah rusak.” Inilah data referensi yang nantinya akan dibandingkan dengan hasil ekstraksi sistem.

Setelah sistem dijalankan, diperoleh hasil sebagai berikut: dari 200 entitas yang seharusnya diekstrak (ground truth), sistem berhasil mengidentifikasi 180 entitas. Dari 180 entitas tersebut, 150 sesuai dengan ground truth (benar), sementara 30 lainnya salah interpretasi. Dengan demikian, kita dapat menghitung metrik evaluasi standar. Precision sistem adalah 150/180 = 83,3%, recall adalah 150/200 = 75%, dan F1 score adalah 2 × (0,833 × 0,75) / (0,833 + 0,75) ≈ 79,0%. Nilai ini menunjukkan bahwa sistem cukup baik, tetapi masih ada ruang untuk perbaikan, terutama dalam meningkatkan recall agar lebih banyak entitas penting dapat terdeteksi.

Lebih jauh lagi, hasil evaluasi juga dianalisis berdasarkan kategori entitas. Ternyata precision untuk entitas tanggal mencapai 95% karena aturan regex untuk format tanggal sudah cukup kuat. Precision untuk entitas lokasi berada di angka 85% karena adanya bantuan gazetteer kabupaten dan kota. Namun, precision untuk entitas jumlah korban hanya 70% karena variasi frasa seperti “ratusan orang tewas” atau “puluhan jiwa meninggal” belum sepenuhnya tercakup dalam aturan. Dari sisi recall, entitas magnitudo gempa menunjukkan nilai rendah, yaitu hanya 65%, karena sistem hanya mengenali pola “SR,” sementara beberapa artikel menulisnya dalam format lain seperti “skala Richter.” Analisis ini memberikan wawasan yang jelas mengenai aturan mana yang perlu diperluas atau diperbaiki.

Selain menghitung angka, hasil evaluasi juga divisualisasikan dalam bentuk confusion matrix dan grafik

precision–recall. Dari confusion matrix terlihat bahwa jumlah false negative cukup tinggi pada entitas jumlah korban dan magnitudo, artinya banyak informasi penting yang terlewat. Grafik precision–recall menunjukkan trade-off: jika aturan untuk korban diperlonggar, recall meningkat, tetapi precision menurun karena sistem juga menandai angka yang bukan jumlah korban. Visualisasi ini membantu pengembang menentukan titik keseimbangan aturan sesuai kebutuhan aplikasi.

Dari contoh ini dapat disimpulkan bahwa evaluasi sistem rule-based bukan hanya sekadar menghitung angka, tetapi juga melakukan analisis kesalahan secara mendalam. Evaluasi memungkinkan pengembang memahami area kekuatan dan kelemahan sistem, serta menyusun strategi perbaikan, misalnya dengan menambahkan aturan baru, memperluas regex, atau mengintegrasikan rule dengan metode berbasis statistik.

Bagi mahasiswa, contoh ini menunjukkan bahwa evaluasi adalah bagian integral dalam pembangunan sistem ekstraksi informasi. Tanpa evaluasi, kita tidak bisa menilai apakah sistem benar-benar berfungsi sesuai tujuan. Dengan evaluasi yang sistematis, hasil ekstraksi dapat dipertanggungjawabkan dan siap digunakan untuk aplikasi nyata, baik di bidang kebencanaan, kesehatan, hukum, maupun analisis data sosial.

Latihan

A. Pertanyaan Pemahaman Konsep

1. Apa yang dimaksud dengan *pipeline* dalam konteks *Information Extraction*?
2. Sebutkan komponen utama dalam pipeline IE dan jelaskan fungsi masing-masing.
3. Bagaimana data mengalir dari tahap preprocessing hingga menghasilkan informasi akhir yang terstruktur?
4. Jelaskan perbedaan antara pipeline modular dan sistem *end-to-end*.
5. Mengapa error pada satu komponen (misalnya NER) dapat memengaruhi hasil akhir IE secara keseluruhan?
6. Apa saja faktor yang menentukan efisiensi pipeline IE dalam skala besar (big data)?
7. Jelaskan peran integrasi pipeline dalam aplikasi seperti *news monitoring system* atau *disaster information system*.

B. Latihan Praktik Sederhana

Diberikan potongan berita berikut:

"Gunung Merapi meletus pada tanggal 11 November 2024, menyebabkan evakuasi ribuan warga di Kabupaten Sleman."

Tugas:

1. Rancang pipeline ekstraksi informasi sederhana untuk kalimat di atas.
2. Tentukan hasil setiap tahap dalam bentuk tabel berikut:

Tahap Pipeline	**Keluaran**	**Keterangan**
Preprocessing	teks bersih tanpa tanda baca	normalisasi data
POS Tagging	identifikasi kelas kata	bantu struktur kalimat

NER	Gunung Merapi (LOC), Kabupaten Sleman (LOC), 11 November 2024 (DATE)	pengenalan entitas
Relation Extraction	(Gunung Merapi, menyebabkan, evakuasi warga)	hubungan sebab-akibat
Event Extraction	"letusan gunung" sebagai peristiwa utama	deteksi event
Output IE	Data terstruktur: {event: letusan, lokasi: Merapi, dampak: evakuasi}	hasil akhir

3. Gambarkan diagram alir pipeline IE dari teks mentah hingga hasil ekstraksi.
4. Analisis: apa yang terjadi jika modul POS Tagging gagal mengenali kelas kata dengan benar?

C. Studi Kasus / Proyek Mini

Anda diminta membangun prototipe sistem IE untuk *analisis berita bencana nasional*.

1. Tentukan arsitektur pipeline yang akan digunakan (komponen, urutan, dan data flow).
2. Deskripsikan fungsi setiap modul dan metode yang digunakan (misal: NLTK untuk tokenisasi, Sastrawi untuk stemming, IndoBERT untuk NER).
3. Pilih tiga berita berbeda dan tunjukkan hasil ekstraksi informasinya dalam bentuk tabel.
4. Tambahkan langkah visualisasi hasil IE (misal: peta lokasi bencana, grafik waktu kejadian).
5. Evaluasi kelebihan dan kekurangan pipeline Anda dibanding pendekatan berbasis model tunggal (end-to-end).

D. Diskusi / Refleksi

1. Menurut Anda, apakah sistem IE yang ideal sebaiknya modular atau end-to-end? Jelaskan alasannya.
2. Bagaimana integrasi pipeline IE dapat mendukung pengambilan keputusan berbasis data (*data-driven decision making*)?
3. Diskusikan tantangan dalam membangun pipeline IE multibahasa, terutama untuk Bahasa Indonesia.
4. Bagaimana pipeline IE dapat dikembangkan lebih lanjut menjadi sistem *intelligent analytics* yang mampu melakukan prediksi?

BAB 11
IMPLEMENTASI EKSTRAKSI INFORMASI (STUDI KASUS BENCANA ALAM)

Tujuan Pembelajaran

Setelah mempelajari Bab 11, mahasiswa diharapkan mampu:

1. Menjelaskan konsep evaluasi dan validasi dalam konteks sistem *Information Extraction (IE)*.
2. Mengidentifikasi berbagai metrik evaluasi kinerja seperti *accuracy*, *precision*, *recall*, dan *F1-score*.
3. Membedakan antara evaluasi kuantitatif dan kualitatif dalam penilaian sistem IE.
4. Menjelaskan konsep *gold standard* atau *ground truth* sebagai acuan evaluasi.
5. Melakukan perhitungan metrik evaluasi menggunakan data hasil ekstraksi.
6. Menganalisis kesalahan (error analysis) untuk meningkatkan performa sistem IE.
7. Merancang prosedur validasi yang sistematis untuk memastikan reliabilitas hasil ekstraksi informasi.

Setelah membahas teori, konsep, dan teknik evaluasi pada bab-bab sebelumnya, bab ini akan berfokus pada implementasi nyata dari ekstraksi informasi berbasis aturan. Sebagai contoh konkret, digunakan domain berita bencana alam di Indonesia, karena teks dalam domain ini memiliki pola yang relatif konsisten, kaya akan informasi penting, dan sangat relevan dalam mendukung sistem peringatan dini maupun analisis dampak bencana.

Berita bencana alam umumnya memuat elemen-elemen informasi kunci yang dapat diekstrak menjadi data terstruktur. Elemen tersebut antara lain jenis bencana (misalnya gempa bumi, banjir, atau tanah longsor), lokasi kejadian, waktu kejadian, jumlah korban jiwa, serta kerusakan infrastruktur. Dengan menggunakan pendekatan rule-based, informasi yang semula tersimpan dalam teks naratif dapat diubah menjadi data yang lebih mudah diproses oleh komputer. Hal ini memungkinkan analisis cepat yang berguna, misalnya untuk mendukung Badan Nasional Penanggulangan Bencana (BNPB), pemerintah daerah, maupun lembaga kemanusiaan dalam mengambil keputusan.

Implementasi yang dibahas dalam bab ini mencakup beberapa tahap penting. Pertama, akan dibahas studi kasus analisis berita bencana alam, yang meliputi pemilihan data uji dan identifikasi informasi yang relevan. Kedua, akan dijelaskan perancangan aturan (rule design) untuk mengekstrak lokasi dan jumlah korban, dua entitas kunci yang sangat penting dalam laporan bencana. Ketiga, aturan yang dirancang akan diimplementasikan dan diuji, untuk melihat sejauh mana keberhasilannya dalam mengekstrak data yang benar. Selanjutnya, dilakukan analisis kualitas hasil ekstraksi, dengan menggunakan metrik evaluasi seperti precision, recall, dan F1 score. Terakhir, bab ini ditutup dengan refleksi dan perbaikan aturan, yang menunjukkan bagaimana sistem dapat disempurnakan agar lebih adaptif terhadap variasi bahasa.

Dengan mempelajari implementasi ini, mahasiswa tidak hanya memahami konsep rule-based extraction secara teoretis, tetapi juga mendapatkan pengalaman praktis dalam merancang, menguji, dan mengevaluasi sebuah sistem ekstraksi informasi sederhana. Diharapkan melalui studi kasus ini, mahasiswa dapat mengembangkan keterampilan

yang lebih aplikatif dan siap diterapkan dalam domain lain di luar kebencanaan.

A. Analisis Berita Bencana Alam

Berita bencana alam merupakan salah satu sumber informasi penting yang memuat data terkait peristiwa darurat. Media massa, baik nasional maupun lokal, biasanya memberikan laporan cepat mengenai jenis bencana, lokasi kejadian, waktu, dampak yang ditimbulkan, serta respons dari pihak berwenang. Karena informasi yang terkandung dalam berita ini sangat relevan bagi analisis kebencanaan, ia menjadi contoh ideal untuk implementasi ekstraksi informasi berbasis aturan.

Dalam studi kasus ini, fokus analisis diarahkan pada berita kebencanaan di Indonesia, khususnya berita tentang gempa bumi dan banjir yang sering terjadi dan memiliki pola pemberitaan yang relatif konsisten. Misalnya, berita gempa biasanya menyebutkan kekuatan gempa dalam skala Richter atau magnitudo (SR), lokasi yang terdampak, jumlah korban jiwa, serta kerusakan bangunan. Sementara berita banjir umumnya memuat keterangan mengenai ketinggian air, jumlah warga terdampak, lokasi pengungsian, serta korban jiwa maupun luka-luka.

Untuk keperluan penelitian, digunakan sejumlah artikel berita daring dari berbagai portal media. Artikel-artikel ini dipilih berdasarkan kriteria bahwa mereka memuat informasi yang cukup detail tentang bencana, bukan sekadar berita singkat. Data berita kemudian dikumpulkan dalam bentuk teks mentah yang siap dianalisis. Dari kumpulan berita ini, ditentukan beberapa entitas target yang akan diekstrak, yaitu:

1. Jenis bencana – gempa bumi, banjir, longsor, erupsi, dan sebagainya.

2. Lokasi kejadian – kabupaten, kota, atau desa tempat bencana terjadi.
3. Tanggal kejadian – kapan bencana berlangsung atau dilaporkan.
4. Jumlah korban – termasuk korban meninggal dunia, luka-luka, atau pengungsi.
5. Kerusakan infrastruktur – jumlah rumah, sekolah, atau fasilitas publik yang terdampak.

Tahap awal analisis dilakukan dengan identifikasi pola bahasa yang umum muncul dalam berita. Misalnya, jumlah korban sering ditulis dengan pola "sebanyak [angka] orang meninggal dunia", sedangkan lokasi sering muncul setelah kata kerja seperti "melanda" atau "mengguncang." Dengan mengenali pola ini, dapat disusun aturan yang sesuai untuk menandai bagian teks yang mengandung informasi target.

Sebagai contoh konkret, perhatikan kalimat berikut: "Gempa bumi berkekuatan 6,2 SR mengguncang Kabupaten Cianjur pada 21 November 2022, menewaskan 162 orang dan merusak 13.000 rumah warga." Dari kalimat ini, entitas target yang dapat diekstrak adalah:

- Jenis bencana: Gempa bumi
- Magnitudo: 6,2 SR
- Lokasi: Kabupaten Cianjur
- Tanggal: 21 November 2022
- Jumlah korban: 162 orang meninggal dunia
- Kerusakan rumah: 13.000 rumah rusak

Studi kasus ini menegaskan bahwa berita bencana memiliki struktur informasi yang cukup teratur, meskipun variasi bahasa tetap ada. Dengan memanfaatkan aturan berbasis pola kalimat, regex, dan gazetteer, informasi

kunci dapat diekstrak secara sistematis. Tahapan ini menjadi dasar bagi subbab berikutnya yang membahas bagaimana merancang aturan (rule design) yang spesifik untuk entitas lokasi dan jumlah korban, sebagai dua aspek paling vital dalam laporan kebencanaan.

B. Desain Rule untuk Ekstraksi Lokasi dan Jumlah Korban

Dalam studi kasus kebencanaan, dua entitas yang hampir selalu menjadi perhatian utama adalah lokasi kejadian dan jumlah korban. Lokasi penting untuk mengetahui area terdampak, sedangkan jumlah korban memberikan gambaran tingkat keparahan bencana. Oleh karena itu, pada tahap implementasi ini kita akan merancang aturan (rules) yang dapat secara otomatis mengekstrak kedua entitas tersebut dari teks berita.

1. Ekstraksi Lokasi

Untuk mengenali lokasi, sistem rule-based dapat memanfaatkan pola sintaksis dan gazetteer. Berdasarkan analisis berita, lokasi sering kali muncul setelah kata kerja seperti "melanda," "mengguncang," atau "terjadi di." Dengan demikian, aturan dapat diformulasikan sebagai: "Jika sebuah kata kerja bencana diikuti oleh nama kabupaten/kota dari daftar gazetteer, maka tandai sebagai lokasi bencana."

Sebagai contoh, pada kalimat "Banjir bandang melanda Kabupaten Garut pada awal Desember 2022," aturan ini akan menandai "Kabupaten Garut" sebagai lokasi. Untuk memperkuat akurasi, sistem dapat menggunakan daftar nama provinsi, kabupaten, dan kota di Indonesia yang sudah dikurasi sebagai gazetteer. Dengan cara ini, variasi penulisan seperti "Kab. Garut" atau "Garut" tetap dapat dikenali sebagai entitas lokasi yang valid.

Selain itu, aturan juga perlu mempertimbangkan preposisi seperti "di" atau "pada" yang sering mendahului nama lokasi. Contoh aturannya adalah: "Jika terdapat pola preposisi 'di' diikuti nama wilayah, maka tandai sebagai entitas lokasi." Aturan ini berguna pada kalimat seperti "Longsor terjadi di Desa Suka Mulya, Kabupaten Bandung Barat."

2. Ekstraksi Jumlah Korban

Ekstraksi jumlah korban biasanya lebih kompleks karena variasi ekspresi bahasa yang digunakan media cukup beragam. Pola umum yang sering ditemukan adalah "[angka] orang meninggal dunia," "[angka] orang luka-luka," atau "[angka] orang mengungsi." Dengan memanfaatkan regex, aturan dapat ditulis sebagai: \d+\s*orang\s+(meninggal dunia|tewas|luka-luka|mengungsi). Pola ini memungkinkan sistem mengenali frasa seperti "162 orang meninggal dunia" atau "200 orang luka-luka."

Namun, variasi lain seperti "ratusan warga kehilangan nyawa" atau "puluhan jiwa tewas" juga sering muncul. Untuk menangani variasi ini, aturan perlu diperluas dengan mencakup kata-kata kuantitatif seperti "puluhan," "ratusan," atau "ribuan." Sebuah aturan tambahan dapat berbunyi: "Jika sebuah kata kuantitatif diikuti kata 'jiwa' atau 'warga' serta kata kerja yang mengindikasikan kematian atau luka, maka tandai sebagai jumlah korban." Dengan demikian, frasa seperti "ratusan jiwa tewas" dapat dikenali secara otomatis.

3. Tantangan dalam Desain Rule

Meskipun tampak sederhana, perancangan

aturan menghadapi tantangan besar karena bahasa alami selalu bervariasi. Beberapa berita menggunakan gaya formal, sementara yang lain lebih ringkas atau bahkan metaforis. Misalnya, frasa "menyebabkan banyak korban berjatuhan" tidak menyebutkan angka spesifik, sehingga aturan berbasis angka tidak bisa mendeteksinya. Oleh karena itu, sistem rule-based sering kali hanya efektif untuk informasi yang terstruktur jelas, dan memerlukan evaluasi serta penyempurnaan berulang untuk menutupi variasi bahasa.

Desain rule untuk ekstraksi lokasi dan jumlah korban menekankan pentingnya menggabungkan pola sintaksis, regex, serta daftar entitas khusus (gazetteer). Dengan kombinasi ini, sistem dapat mengenali lokasi secara akurat dan mengekstrak jumlah korban dalam berbagai bentuk penulisan. Namun, keterbatasan tetap ada sehingga evaluasi dan penyempurnaan aturan harus menjadi bagian dari proses pengembangan. Pada subbab berikutnya, aturan yang telah dirancang ini akan diimplementasikan dan diuji pada kumpulan berita bencana untuk melihat performa nyatanya.

C. Implementasi dan Pengujian Rule

Setelah aturan untuk ekstraksi lokasi dan jumlah korban dirancang, tahap berikutnya adalah implementasi. Implementasi berarti menerjemahkan aturan yang telah disusun ke dalam bentuk yang dapat dijalankan oleh komputer. Pada studi kasus ini, implementasi dilakukan dengan menggunakan bahasa pemrograman Python serta pustaka pemrosesan teks seperti NLTK atau spaCy, ditambah penggunaan regular expressions (regex) untuk menangkap pola-pola tertentu.

Langkah pertama adalah melakukan preprocessing teks. Berita bencana yang dikumpulkan dari berbagai sumber perlu dibersihkan terlebih dahulu agar dapat dianalisis dengan lebih mudah. Proses ini meliputi penghapusan karakter non-alfabet, normalisasi huruf kapital, serta tokenisasi kalimat dan kata. Preprocessing sangat penting karena aturan hanya dapat bekerja secara konsisten jika teks sudah berada dalam bentuk yang terstandarisasi.

Selanjutnya, aturan yang telah dirancang pada subbab sebelumnya diterapkan pada teks. Misalnya, untuk mengekstrak lokasi, sistem menggunakan kombinasi regex dan pencocokan kata dengan daftar gazetteer kabupaten/kota di Indonesia. Ketika sistem menemukan frasa seperti "Kabupaten Cianjur" setelah kata kerja "mengguncang," maka frasa tersebut secara otomatis ditandai sebagai entitas lokasi. Sementara itu, aturan untuk jumlah korban memanfaatkan regex seperti \d+\s*orang\s+(meninggal dunia|tewas|luka-luka) untuk menangkap pola angka yang diikuti dengan keterangan korban. Aturan tambahan juga digunakan untuk menangkap pola kuantitatif seperti "puluhan jiwa tewas" atau "ratusan warga mengungsi."

Untuk menguji performa sistem, digunakan dataset uji berupa 50 artikel berita bencana dari berbagai media daring. Setiap artikel telah dianotasi secara manual oleh anotator manusia untuk menghasilkan ground truth sebagai pembanding. Dari proses implementasi, sistem berhasil mengekstrak sejumlah besar entitas yang sesuai dengan ground truth, namun juga menghasilkan beberapa kesalahan baik dalam bentuk false positive (sistem menandai sesuatu yang bukan entitas) maupun false negative (sistem gagal menangkap entitas yang ada).

Sebagai ilustrasi, pada berita "Gempa bumi berkekuatan 6,2 SR mengguncang Kabupaten Cianjur pada 21 November 2022, menewaskan 162 orang dan merusak 13.000 rumah warga," sistem berhasil mengekstrak entitas lokasi "Kabupaten Cianjur" dan jumlah korban "162 orang meninggal dunia." Namun, pada berita lain seperti "Ratusan jiwa kehilangan nyawa akibat banjir di Kabupaten Garut," sistem sempat gagal mengenali jumlah korban karena aturan awal hanya mengakomodasi frasa "[angka] orang meninggal dunia." Dari kasus ini terlihat bahwa evaluasi sangat penting untuk mengidentifikasi kelemahan aturan.

Hasil pengujian awal menunjukkan precision sistem mencapai 83%, recall 75%, dan F1 score sekitar 79%. Precision yang relatif tinggi menunjukkan bahwa sebagian besar hasil ekstraksi yang ditandai memang benar adanya, namun recall yang masih sedang menunjukkan bahwa sistem melewatkan beberapa entitas penting, terutama ketika frasa ditulis dengan variasi bahasa yang lebih bebas.

Pengujian ini menjadi bukti nyata bahwa aturan yang telah dirancang dapat bekerja pada data nyata, namun tetap memerlukan penyempurnaan. Tahap berikutnya adalah analisis kualitas hasil ekstraksi, untuk mengidentifikasi pola kesalahan dan menemukan solusi perbaikan aturan agar sistem menjadi lebih akurat dan andal.

D. Analisis Kualitas Hasil Ekstraksi

Setelah proses implementasi dan pengujian rule dilakukan, langkah berikutnya adalah menganalisis kualitas hasil ekstraksi. Analisis ini bertujuan untuk memahami sejauh mana aturan yang telah dibuat mampu mengenali entitas target dengan benar, serta bagian mana

yang masih sering menimbulkan kesalahan. Dengan demikian, analisis kualitas tidak hanya memberikan angka evaluasi, tetapi juga wawasan praktis yang berguna untuk penyempurnaan sistem.

Dari hasil pengujian terhadap 50 artikel berita bencana, sistem menghasilkan nilai precision sebesar 83%, recall sebesar 75%, dan F1 score sekitar 79%. Angka ini menunjukkan bahwa sebagian besar entitas yang diekstrak oleh sistem memang benar sesuai dengan ground truth, namun masih ada cukup banyak entitas yang terlewat. Precision yang tinggi mengindikasikan bahwa aturan yang dibuat cukup ketat dalam menentukan entitas, sehingga jarang menandai informasi yang salah. Namun, recall yang lebih rendah menunjukkan bahwa aturan belum cukup fleksibel untuk menangkap variasi penulisan yang lebih beragam.

Jika dianalisis berdasarkan kategori entitas, terlihat adanya perbedaan performa. Pada entitas lokasi, hasil ekstraksi relatif baik dengan precision 85% dan recall 80%. Hal ini terutama karena adanya dukungan gazetteer berupa daftar kabupaten/kota di Indonesia, sehingga sistem mudah mengenali lokasi meskipun ditulis dalam bentuk yang berbeda-beda. Namun, pada entitas jumlah korban, performa lebih rendah dengan precision hanya 70% dan recall 65%. Penyebab utamanya adalah keragaman cara media menyajikan informasi jumlah korban. Beberapa berita menyebut angka spesifik seperti "162 orang meninggal dunia," sementara yang lain menggunakan ungkapan kualitatif seperti "ratusan jiwa kehilangan nyawa" atau "puluhan warga tewas." Aturan berbasis regex sederhana tidak mampu menangkap semua variasi tersebut.

Selain itu, analisis kesalahan (error analysis)

menunjukkan bahwa false negative (entitas yang terlewat) lebih dominan dibanding false positive. Hal ini berarti sistem cenderung berhati-hati dan hanya mengenali pola yang jelas, sehingga banyak informasi valid yang tidak berhasil ditangkap. Sementara itu, false positive umumnya muncul ketika sistem salah menafsirkan angka dalam berita. Misalnya, angka "200" pada frasa "200 relawan dikerahkan" sempat ditandai sebagai jumlah korban karena pola aturannya hanya mengenali kombinasi angka dan kata "orang."

Dari hasil analisis ini dapat ditarik beberapa kesimpulan penting. Pertama, aturan yang berbasis pada pola sintaksis sederhana cenderung menghasilkan precision tinggi, tetapi recall rendah. Kedua, variasi bahasa alami yang kaya menuntut aturan lebih adaptif dan modular. Ketiga, dukungan kamus atau gazetteer sangat membantu untuk meningkatkan akurasi, terutama pada entitas lokasi. Keempat, untuk entitas yang sangat bervariasi seperti jumlah korban, dibutuhkan perluasan aturan dengan menambahkan pola-pola alternatif atau bahkan integrasi dengan pendekatan statistik.

Analisis kualitas hasil ekstraksi ini menegaskan bahwa proses pengembangan sistem rule-based bersifat iteratif. Aturan yang sudah diuji harus dianalisis, kelemahannya diidentifikasi, kemudian diperbaiki dan diuji ulang. Dengan siklus evaluasi dan perbaikan yang berulang, sistem dapat terus ditingkatkan hingga mencapai tingkat akurasi yang memadai untuk kebutuhan nyata. Tahap selanjutnya dalam bab ini adalah membahas bagaimana aturan dapat diperbaiki serta refleksi dari studi kasus yang telah dilakukan.

E. Perbaikan dan Refleksi Hasil Studi Kasus

Hasil analisis pada subbab sebelumnya menunjukkan bahwa meskipun sistem rule-based mampu mengekstrak sebagian besar entitas penting dari berita bencana, masih terdapat kelemahan terutama pada cakupan dan fleksibilitas aturan. Oleh karena itu, tahap selanjutnya adalah melakukan perbaikan aturan serta melakukan refleksi terhadap keseluruhan proses implementasi.

Perbaikan pertama dilakukan pada aturan untuk ekstraksi jumlah korban, yang sebelumnya terbatas pada pola angka spesifik diikuti kata "orang meninggal dunia" atau "orang luka-luka." Untuk memperluas cakupan, aturan diperbarui agar juga dapat mengenali frasa kuantitatif seperti "puluhan jiwa tewas" atau "ratusan warga mengungsi." Hal ini dicapai dengan menambahkan daftar kata kuantitatif (puluhan, ratusan, ribuan) dan sinonim untuk subjek (orang, jiwa, warga). Dengan perluasan ini, recall sistem meningkat karena lebih banyak variasi frasa yang dapat ditangkap.

Perbaikan kedua dilakukan pada aturan untuk lokasi kejadian. Meskipun precision dan recall untuk entitas lokasi sudah relatif baik, masih terdapat kesalahan pada frasa yang ambigu, misalnya "Jakarta Selatan" yang kadang dipahami sebagai entitas administratif, namun pada konteks tertentu bisa merujuk pada arah geografis. Untuk mengatasi hal ini, aturan dilengkapi dengan validasi berbasis gazetteer yang lebih kaya, mencakup nama-nama resmi wilayah administratif di Indonesia. Dengan langkah ini, sistem menjadi lebih konsisten dalam mengenali lokasi yang sahih.

Selain itu, sistem juga diperbaiki agar dapat menghindari false positive pada angka-angka yang tidak

berkaitan dengan korban. Aturan diperketat dengan menambahkan konteks semantik. Sebagai contoh, angka yang diikuti kata "relawan," "petugas," atau "tim SAR" tidak akan ditandai sebagai korban, karena konteksnya jelas berbeda. Dengan cara ini, precision dapat ditingkatkan tanpa mengorbankan recall secara signifikan.

Dari refleksi keseluruhan studi kasus ini, dapat ditarik beberapa pelajaran penting. Pertama, pendekatan rule-based sangat berguna untuk domain dengan pola bahasa yang relatif konsisten, seperti berita bencana. Kedua, kekuatan utama rule-based adalah pada transparansi aturan, di mana setiap hasil ekstraksi dapat ditelusuri kembali ke aturan yang digunakan. Hal ini berbeda dengan model machine learning yang sering kali berfungsi sebagai "black box." Ketiga, kelemahan rule-based terletak pada keterbatasannya menghadapi variasi bahasa yang luas. Untuk mengatasinya, diperlukan strategi seperti memperkaya aturan, menggunakan gazetteer yang lebih komprehensif, atau mengombinasikan dengan metode statistik.

Refleksi lain yang perlu dicatat adalah bahwa proses pengembangan rule-based bersifat iteratif dan berulang. Aturan tidak bisa sempurna sejak awal, melainkan harus terus diuji, dievaluasi, dan diperbaiki. Setiap kali sistem diuji dengan data baru, akan muncul pola kalimat yang sebelumnya tidak terduga, sehingga aturan perlu disesuaikan. Dengan siklus iteratif ini, sistem perlahan-lahan menjadi lebih robust dan andal.

Sebagai penutup, studi kasus ekstraksi informasi pada berita bencana alam ini membuktikan bahwa pendekatan rule-based, meskipun sederhana, tetap memiliki nilai praktis yang tinggi. Dengan perancangan aturan yang tepat dan evaluasi yang sistematis, sistem

rule-based dapat digunakan untuk menghasilkan data terstruktur yang mendukung analisis kebencanaan. Lebih jauh lagi, pengalaman ini memberikan dasar bagi mahasiswa untuk memahami bagaimana konsep-konsep yang dipelajari di kelas dapat diterapkan secara nyata, sekaligus membuka peluang pengembangan lebih lanjut melalui integrasi dengan metode berbasis pembelajaran mesin.

Latihan

A. Pertanyaan Pemahaman Konsep

1. Mengapa evaluasi penting dalam sistem ekstraksi informasi?
2. Jelaskan perbedaan antara *precision* dan *recall* dalam konteks IE.
3. Apa tujuan dari metrik *F1-score* dan bagaimana cara menghitungnya?
4. Jelaskan apa yang dimaksud dengan *gold standard* dan bagaimana cara menyusunnya.
5. Sebutkan dua jenis kesalahan yang sering terjadi dalam IE dan berikan contohnya.
6. Apa perbedaan antara evaluasi intrinsik dan ekstrinsik?
7. Bagaimana kita dapat mengukur efisiensi (waktu komputasi) selain akurasi dalam sistem IE?

B. Latihan Praktik Sederhana

Diberikan hasil ekstraksi entitas dari sistem NER berikut:

Kalimat Asli	**Entitas yang Benar (Gold Standard)**	**Entitas yang Dihasilkan Sistem**	**Hasil**
Presiden Joko Widodo menghadiri pertemuan ASEAN di Bangkok.	{Joko Widodo (PER), ASEAN (ORG), Bangkok (LOC)}	{Joko Widodo (PER), ASEAN (ORG)}	1 entitas hilang
Gunung Merapi meletus pada tanggal 11	{Gunung Merapi (LOC), 11 November	{Gunung Merapi (LOC), 11 November	lengkap

November 2024.	2024 (DATE)}	2024 (DATE)}	
Pertamina membuka lowongan kerja untuk lulusan baru.	{Pertamina (ORG)}	{Pertamina (ORG), lulusan (PER)}	1 entitas salah klasifikasi

Tugas:

1. Hitung nilai *precision*, *recall*, dan *F1-score* dari hasil di atas.
2. Sajikan hasilnya dalam tabel evaluasi berikut:

Metrik	**Rumus**	**Nilai**
Precision	TP / (TP + FP)	...
Recall	TP / (TP + FN)	...
F1-score	2 × (Precision × Recall) / (Precision + Recall)	...

3. Jelaskan interpretasi hasil evaluasi:
   - Apa artinya jika *precision* tinggi tetapi *recall* rendah?
   - Apa yang harus dilakukan jika *recall* sistem masih buruk?
4. Lakukan analisis kesalahan (*error analysis*) untuk mengetahui penyebab entitas salah dikenali.

C. Studi Kasus / Proyek Mini

Anda akan melakukan evaluasi terhadap sistem *Information Extraction* untuk berita bencana nasional.

1. Siapkan dataset berlabel sebanyak 50 kalimat yang berisi entitas lokasi, tanggal, dan jenis bencana.
2. Jalankan sistem IE yang telah dibangun pada bab sebelumnya.
3. Bandingkan hasil sistem dengan *gold standard*

menggunakan metrik evaluasi yang relevan.

4. Visualisasikan hasil evaluasi dalam grafik batang (bar chart) untuk setiap kategori entitas.
5. Lakukan *error analysis* dengan mengelompokkan kesalahan menjadi: *false positive*, *false negative*, dan *misclassification*.
6. Berikan rekomendasi langkah perbaikan untuk meningkatkan akurasi sistem.

D. Diskusi / Refleksi

1. Menurut Anda, apakah metrik kuantitatif seperti akurasi cukup untuk menilai kualitas sistem IE?
2. Bagaimana pendekatan evaluasi bisa disesuaikan untuk sistem IE berbasis *deep learning* yang bersifat *end-to-end*?
3. Diskusikan pentingnya validasi manual (oleh pakar bahasa) dalam memastikan keandalan hasil IE.
4. Bagaimana strategi evaluasi sistem IE dapat diadaptasi untuk Bahasa Indonesia yang memiliki struktur morfologi kompleks?

BAB 12
EKSTRAKSI INFORMASI BERBASIS RULE

Tujuan Pembelajaran

Setelah mempelajari Bab 12, mahasiswa diharapkan mampu:

1. Menerapkan konsep dan metode ekstraksi informasi (IE) pada berbagai kasus nyata berbasis teks.
2. Menganalisis kebutuhan data dan arsitektur sistem IE sesuai domain aplikasi.
3. Merancang pipeline IE yang terintegrasi mulai dari preprocessing hingga visualisasi hasil.
4. Mengevaluasi hasil penerapan IE pada kasus berbeda, seperti bencana alam, berita ekonomi, atau media sosial.
5. Mengidentifikasi tantangan implementasi nyata, termasuk keterbatasan dataset, bahasa, dan konteks domain.
6. Menghubungkan teori IE dengan solusi praktis dalam mendukung pengambilan keputusan berbasis data.
7. Menulis laporan hasil penerapan IE secara ilmiah dan sistematis.

Sebagai penutup dari buku ini, Bab 12 menghadirkan sebuah proyek mini yang dirancang untuk memberikan pengalaman langsung dalam membangun sistem ekstraksi informasi berbasis aturan. Setelah melalui berbagai bab sebelumnya, mahasiswa telah mempelajari konsep dasar, metode lanjutan, teknik evaluasi, hingga studi kasus implementasi pada domain kebencanaan. Proyek mini ini bertujuan untuk mengintegrasikan seluruh pengetahuan tersebut ke dalam sebuah praktik nyata, sehingga mahasiswa

tidak hanya memahami teori, tetapi juga mampu mengaplikasikannya secara mandiri.

Proyek mini dirancang agar sederhana, namun tetap mencakup seluruh tahapan penting dalam pengembangan sistem ekstraksi informasi. Dimulai dari identifikasi topik dan tujuan proyek, mahasiswa diajak menentukan domain yang akan dianalisis, misalnya berita politik, artikel kesehatan, ulasan produk, atau data kebencanaan. Setelah itu, mahasiswa akan melakukan penyusunan dataset dan ground truth dengan cara mengumpulkan data teks dari sumber yang relevan dan membuat anotasi manual sebagai data referensi.

Tahap berikutnya adalah pengembangan aturan dan pipeline EI, di mana mahasiswa merancang rule sederhana untuk mengenali entitas tertentu, kemudian menyusunnya dalam alur pemrosesan teks yang sistematis. Aturan yang dibuat kemudian diimplementasikan dan diuji, untuk melihat performanya berdasarkan metrik evaluasi seperti precision, recall, dan F1 score.

Terakhir, mahasiswa akan menyusun dokumentasi, presentasi, dan refleksi proyek. Dokumentasi mencatat proses yang dilakukan, presentasi digunakan untuk membagikan hasil kepada teman sekelas atau penguji, sedangkan refleksi bertujuan agar mahasiswa mampu mengidentifikasi kekuatan dan kelemahan sistem yang mereka bangun. Dengan refleksi ini, mahasiswa akan belajar bahwa pengembangan sistem ekstraksi informasi bukan hanya tentang menulis aturan, tetapi juga tentang mengevaluasi, memperbaiki, dan mengomunikasikan hasil kerja.

Melalui proyek mini ini, mahasiswa diharapkan dapat memperoleh pengalaman yang komprehensif dan aplikatif. Proyek ini juga memberi kesempatan bagi mahasiswa untuk berkreasi sesuai minat, karena domain yang dipilih bisa

sangat beragam, sehingga hasilnya tidak hanya bermanfaat untuk pembelajaran, tetapi juga berpotensi dikembangkan lebih lanjut untuk penelitian atau aplikasi praktis.

A. Identifikasi Topik dan Tujuan Proyek

Tahap pertama dalam proyek mini ekstraksi informasi berbasis aturan adalah melakukan identifikasi topik yang akan dianalisis serta merumuskan tujuan proyek. Pemilihan topik menjadi langkah yang sangat penting, karena akan menentukan jenis data yang dikumpulkan, bentuk aturan yang dikembangkan, serta kompleksitas sistem yang dibangun.

Topik proyek dapat dipilih sesuai dengan minat mahasiswa maupun relevansi dengan bidang studi. Misalnya, mahasiswa yang tertarik pada isu sosial-politik dapat memilih topik ekstraksi informasi dari berita politik, seperti mengidentifikasi tokoh, partai, dan kebijakan yang dibahas. Mahasiswa dari bidang kesehatan bisa mengambil topik analisis artikel medis untuk mengekstrak nama penyakit, obat, dan gejala. Sementara itu, mahasiswa yang mengikuti studi kasus sebelumnya bisa melanjutkan dengan topik kebencanaan, misalnya mengekstrak lokasi bencana, jumlah korban, dan jenis kerusakan dari berita daring. Topik lain yang juga menarik adalah ekstraksi informasi dari ulasan produk, di mana sistem bertugas menemukan fitur produk yang dipuji atau dikritik konsumen.

Setelah topik ditentukan, langkah berikutnya adalah merumuskan tujuan proyek secara jelas. Tujuan proyek berfungsi sebagai arah dan batas ruang lingkup implementasi. Misalnya, tujuan proyek dapat berbunyi: "Membangun sistem rule-based sederhana untuk mengekstrak entitas lokasi dan jumlah korban dari berita kebencanaan." Atau, "Mengembangkan aturan berbasis

regex untuk mengidentifikasi nama obat dan dosis dari artikel medis." Dengan merumuskan tujuan yang spesifik, mahasiswa dapat lebih fokus dalam menentukan jenis aturan yang akan dikembangkan.

Tujuan proyek sebaiknya mencakup aspek praktis sekaligus aspek pembelajaran. Dari sisi praktis, mahasiswa belajar bagaimana sistem rule-based dapat membantu mengubah teks tidak terstruktur menjadi data terstruktur yang lebih mudah dianalisis. Dari sisi pembelajaran, proyek ini membantu mahasiswa memahami bagaimana teori yang telah dipelajari pada bab-bab sebelumnya dapat diimplementasikan dalam situasi nyata.

Selain itu, mahasiswa juga dapat menambahkan tujuan tambahan, seperti membandingkan performa sistem rule-based dengan metode sederhana lain, atau mengeksplorasi variasi aturan untuk melihat dampaknya terhadap precision dan recall. Dengan begitu, proyek mini tidak hanya berfungsi sebagai latihan teknis, tetapi juga sebagai sarana untuk berpikir kritis dan reflektif.

Dengan melalui tahap identifikasi topik dan tujuan ini, mahasiswa akan memiliki fondasi yang kuat sebelum masuk ke langkah berikutnya, yaitu menyusun dataset dan ground truth. Pemilihan topik yang tepat dan tujuan yang jelas akan memastikan proyek mini dapat dilaksanakan secara terarah, terukur, dan bermanfaat baik untuk pembelajaran maupun pengembangan sistem ekstraksi informasi lebih lanjut.

B. Penyusunan Dataset dan Ground Truth

Setelah topik dan tujuan proyek mini ditentukan, tahap berikutnya adalah menyusun dataset dan ground truth. Dataset berfungsi sebagai bahan uji untuk sistem ekstraksi, sementara ground truth digunakan sebagai

standar pembanding dalam evaluasi. Tanpa dataset yang memadai dan ground truth yang jelas, performa sistem tidak dapat dinilai secara obyektif.

1. **Penyusunan Dataset**

   Dataset adalah kumpulan teks yang relevan dengan topik proyek. Jika mahasiswa memilih topik kebencanaan, maka dataset dapat berupa artikel berita tentang banjir, gempa bumi, atau tanah longsor dari portal berita daring. Jika topiknya kesehatan, dataset bisa berupa ringkasan jurnal medis atau artikel kesehatan populer. Untuk topik ulasan produk, dataset dapat dikumpulkan dari e-commerce atau forum konsumen.

   Jumlah dataset tidak harus terlalu besar, tetapi cukup untuk mewakili variasi bahasa dalam domain yang dipilih. Sebagai panduan, untuk proyek mini biasanya digunakan antara 20 hingga 50 teks. Jumlah ini cukup untuk melatih keterampilan menulis aturan sekaligus memungkinkan proses evaluasi yang terukur. Penting juga untuk memperhatikan keberagaman dataset. Misalnya, berita bencana sebaiknya diambil dari beberapa sumber media agar variasi gaya bahasa dapat tertangkap.

   Teks yang dikumpulkan perlu melalui tahap preprocessing awal, misalnya penghapusan tanda baca berlebih, normalisasi huruf kapital, atau pengubahan format file ke bentuk teks mentah (.txt). Preprocessing sederhana ini bertujuan agar teks siap diproses dan tidak menimbulkan masalah teknis saat aturan dijalankan.

2. Penyusunan Ground Truth

Ground truth adalah data referensi yang berisi entitas atau informasi yang dianggap benar dari dataset. Penyusunan ground truth dilakukan melalui anotasi manual, yaitu menandai entitas target pada teks sesuai dengan tujuan proyek.

Sebagai contoh, untuk topik berita bencana, ground truth dapat berupa anotasi terhadap entitas lokasi, tanggal, jumlah korban jiwa, dan jumlah kerusakan. Jika pada sebuah berita tertulis "Banjir bandang melanda Kabupaten Garut pada awal Desember 2022 dan menewaskan 12 orang," maka ground truth akan menandai:

- Lokasi → Kabupaten Garut
- Waktu → awal Desember 2022
- Jumlah korban → 12 orang meninggal dunia

Proses anotasi ground truth sebaiknya dilakukan dengan konsistensi yang tinggi. Untuk itu, perlu dibuat pedoman anotasi (annotation guideline) yang menjelaskan aturan baku. Misalnya, apakah frasa "ratusan orang" boleh ditandai sebagai jumlah korban meski tidak menyebut angka spesifik? Apakah "Jawa Barat" dianggap lokasi jika berita sudah menyebut kabupaten secara rinci? Dengan adanya pedoman ini, hasil anotasi menjadi lebih seragam.

Dalam beberapa kasus, penyusunan ground truth melibatkan lebih dari satu anotator untuk mengurangi subjektivitas. Jika ada perbedaan hasil, dilakukan diskusi atau voting untuk menentukan label yang paling tepat. Tingkat kesepakatan antar anotator dapat diukur dengan metrik seperti Cohen's Kappa,

meskipun untuk proyek mini hal ini bisa diperlakukan lebih sederhana.

3. Pentingnya Dataset dan Ground Truth

Dataset yang relevan dan ground truth yang konsisten adalah pondasi dari seluruh proyek mini. Aturan yang dibuat, pipeline yang dibangun, hingga evaluasi sistem semuanya bergantung pada kualitas data ini. Jika dataset terlalu sedikit atau tidak bervariasi, aturan mungkin tampak bekerja dengan baik padahal sebenarnya tidak general. Jika ground truth dibuat secara tidak konsisten, evaluasi akan bias dan tidak mencerminkan performa sistem yang sebenarnya.

Oleh karena itu, mahasiswa perlu memahami bahwa penyusunan dataset dan ground truth bukan sekadar langkah teknis, tetapi juga bagian dari proses ilmiah yang menentukan validitas hasil proyek. Dengan dataset yang tepat dan ground truth yang solid, proyek mini akan memberikan hasil yang lebih bermakna, baik dari sisi pembelajaran maupun aplikasi praktis.

C. Pengembangan Aturan dan Pipeline EI

Setelah dataset dan ground truth disusun, tahap berikutnya adalah pengembangan aturan (rules) dan penyusunan pipeline ekstraksi informasi (EI). Pada tahap inilah mahasiswa mulai menerjemahkan konsep yang telah dipelajari menjadi sebuah sistem sederhana yang dapat memproses teks dan mengekstrak informasi sesuai tujuan proyek.

Aturan adalah inti dari pendekatan rule-based. Aturan dirancang untuk mengenali pola-pola bahasa tertentu yang mewakili entitas atau informasi yang ingin

diekstrak. Misalnya, jika topik proyek adalah berita bencana, maka aturan untuk lokasi bisa berupa: "Jika sebuah kata didahului oleh preposisi 'di' atau 'pada' dan cocok dengan nama wilayah dalam gazetteer, maka tandai sebagai lokasi." Sedangkan aturan untuk jumlah korban bisa menggunakan regex seperti: \d+\s*orang\s+(meninggal dunia|tewas|luka-luka).

Aturan tidak harus rumit, tetapi harus spesifik sesuai domain. Pada ulasan produk, misalnya, aturan bisa berbunyi: "Jika kata sifat positif seperti 'bagus,' 'nyaman,' atau 'awet' muncul dekat dengan nama fitur produk, tandai sebagai opini positif terhadap fitur tersebut." Dengan cara ini, mahasiswa belajar bahwa aturan selalu bergantung pada pola khas dalam teks yang dianalisis.

Pipeline adalah rangkaian langkah yang harus dilalui teks sebelum aturan diterapkan. Pipeline membantu sistem bekerja secara sistematis, sehingga setiap tahap dapat mendukung hasil akhir ekstraksi. Pipeline untuk proyek mini dapat disusun secara sederhana, dengan tahapan umum sebagai berikut:

1. Preprocessing teks → pembersihan teks dari karakter non-alfabet, normalisasi huruf kapital, serta tokenisasi.
2. POS tagging atau dependency parsing (opsional) → untuk memperkaya informasi linguistik, terutama jika aturan berbasis struktur kalimat.
3. Penerapan aturan regex atau pola → aturan dijalankan pada teks untuk menandai entitas target.
4. Validasi dengan gazetteer atau daftar domain → misalnya, mencocokkan nama wilayah dengan daftar kabupaten/kota di Indonesia.
5. Ekstraksi hasil dalam format terstruktur → entitas yang ditemukan dicatat dalam tabel atau format JSON agar mudah dianalisis.

Sebagai contoh, pipeline berita bencana dapat bekerja dengan alur berikut: teks berita dimasukkan → sistem menandai kata yang sesuai pola regex jumlah korban → sistem mencocokkan lokasi dengan gazetteer → hasil akhir disimpan dalam bentuk tabel dengan kolom jenis bencana, lokasi, tanggal, korban, kerusakan.

Salah satu poin penting adalah bagaimana aturan terintegrasi dengan pipeline. Aturan sebaiknya ditempatkan setelah preprocessing, agar teks sudah bersih dan konsisten. Jika aturan bergantung pada POS tagging atau dependency parsing, maka tahap tersebut harus dimasukkan sebelum penerapan aturan. Misalnya, aturan "kata kerja + lokasi" membutuhkan informasi kategori kata (verb, noun), sehingga POS tagging menjadi syarat.

Pipeline sederhana ini bisa diimplementasikan menggunakan pustaka Python seperti NLTK atau spaCy, ditambah modul re (regular expressions) untuk pola teks. Tujuan utama bukan untuk membuat sistem yang sempurna, tetapi agar mahasiswa mengalami langsung bagaimana konsep rule-based dijalankan secara teknis.

Pipeline membantu mahasiswa memahami bahwa ekstraksi informasi bukan hanya soal menulis aturan, melainkan juga bagaimana aturan tersebut dijalankan dalam alur yang terstruktur. Pipeline membuat sistem lebih mudah diuji, dievaluasi, dan diperbaiki. Misalnya, jika banyak kesalahan terjadi pada tahap ekstraksi jumlah korban, pengembang cukup memperbaiki aturan di tahap tersebut tanpa mengubah pipeline secara keseluruhan.

Dengan demikian, pengembangan aturan dan pipeline EI memberikan mahasiswa pengalaman menyusun sistem yang utuh, mulai dari teks mentah hingga informasi terstruktur. Tahap ini menjadi inti dari proyek mini karena di sinilah teori berubah menjadi

praktik nyata. Pada subbab berikutnya, pipeline yang telah dikembangkan ini akan diimplementasikan pada dataset, kemudian diuji performanya dengan membandingkan hasil ekstraksi terhadap ground truth.

D. Implementasi dan Pengujian Sistem

Tahap implementasi adalah saat di mana aturan dan pipeline yang telah dirancang benar-benar dijalankan pada dataset yang telah disiapkan. Implementasi dalam konteks proyek mini biasanya menggunakan bahasa pemrograman yang mendukung pemrosesan teks, seperti Python, dengan bantuan pustaka NLP sederhana seperti NLTK atau spaCy, serta modul regex (re) untuk menangani pola berbasis ekspresi reguler.

Pada tahap ini, mahasiswa terlebih dahulu memuat dataset berita atau teks lain yang sudah dikumpulkan. Teks kemudian melewati tahap preprocessing, seperti tokenisasi kata dan kalimat, normalisasi huruf kapital, serta penghapusan tanda baca yang tidak relevan. Preprocessing penting agar aturan dapat bekerja secara konsisten, mengingat variasi penulisan dalam teks sering kali menimbulkan hambatan.

Setelah preprocessing, sistem mulai menerapkan aturan yang telah disusun. Misalnya, aturan ekstraksi lokasi dijalankan dengan mencocokkan pola "di [lokasi]" atau "melanda [lokasi]" dan memverifikasi hasilnya dengan daftar gazetteer wilayah Indonesia. Aturan ekstraksi jumlah korban dijalankan menggunakan regex seperti `\d+\s*orang\s+(meninggal dunia|tewas|luka-luka)` atau pola kuantitatif seperti "puluhan jiwa tewas." Semua hasil ekstraksi yang ditemukan dicatat dalam format terstruktur, seperti tabel atau JSON, sehingga dapat dengan mudah dibandingkan dengan ground truth.

Tahap berikutnya adalah pengujian sistem. Pada tahap ini, hasil ekstraksi dibandingkan dengan ground truth yang sudah dianotasi sebelumnya. Misalnya, jika ground truth menyebutkan bahwa dalam sebuah berita terdapat entitas "Kabupaten Cianjur" sebagai lokasi dan "162 orang meninggal dunia" sebagai jumlah korban, maka hasil sistem diperiksa apakah berhasil menemukan entitas tersebut dengan tepat. Jika sistem berhasil, maka dihitung sebagai true positive. Jika sistem menandai entitas yang salah, dihitung sebagai false positive. Jika sistem gagal mendeteksi entitas yang ada, dihitung sebagai false negative.

Dari data perbandingan ini, dihitung metrik evaluasi standar, yaitu precision, recall, dan F1 score. Misalnya, dari 50 artikel berita bencana, sistem berhasil mengekstrak 120 entitas lokasi, dengan 100 di antaranya benar. Pada ground truth, tercatat 130 entitas lokasi. Dari perbandingan tersebut, precision untuk lokasi adalah 100/120 = 83,3%, recall adalah 100/130 = 76,9%, dan F1 score adalah 80%. Proses serupa juga dilakukan untuk entitas jumlah korban, yang biasanya menghasilkan precision lebih rendah karena variasi bahasa lebih tinggi.

Hasil pengujian ini memberikan gambaran objektif mengenai performa sistem. Precision yang tinggi menunjukkan bahwa aturan yang dibuat cukup tepat dalam mengenali entitas, sedangkan recall yang lebih rendah menunjukkan bahwa sistem masih melewatkan banyak variasi pola bahasa. Dengan kata lain, sistem cenderung aman (tidak banyak salah menandai), tetapi belum sepenuhnya mampu menangkap semua informasi yang ada.

Selain angka evaluasi, analisis kesalahan juga menjadi bagian penting dari pengujian. Misalnya,

ditemukan bahwa sistem gagal mengenali frasa "korban jiwa mencapai ratusan orang" karena aturan hanya mengenali angka eksplisit. Atau sistem keliru menandai "200 relawan dikerahkan" sebagai jumlah korban karena hanya mendeteksi pola "angka + orang." Analisis kesalahan ini menjadi dasar untuk menyempurnakan aturan di tahap perbaikan.

Dengan demikian, tahap implementasi dan pengujian sistem tidak hanya menghasilkan angka kinerja, tetapi juga memberikan wawasan praktis mengenai kekuatan dan kelemahan aturan yang telah dibuat. Mahasiswa diharapkan memahami bahwa membangun sistem ekstraksi informasi adalah proses iteratif: aturan dibuat, diuji, diperbaiki, lalu diuji kembali. Proses ini mencerminkan alur nyata dalam pengembangan aplikasi NLP di dunia praktis.

E. Dokumentasi, Presentasi, dan Refleksi Proyek

Tahap terakhir dalam proyek mini ekstraksi informasi berbasis aturan adalah melakukan dokumentasi, presentasi, dan refleksi. Tahap ini tidak kalah penting dibandingkan implementasi teknis, karena di sinilah mahasiswa diajak untuk mengomunikasikan hasil kerja, menilai kembali proses yang telah dilakukan, serta mengambil pelajaran untuk pengembangan selanjutnya.

Dokumentasi berfungsi untuk mencatat secara sistematis seluruh tahapan proyek, mulai dari identifikasi topik, penyusunan dataset, perancangan aturan, hingga hasil pengujian. Dokumentasi sebaiknya disusun dalam bentuk laporan yang rapi, mencakup latar belakang pemilihan topik, tujuan proyek, deskripsi dataset, aturan yang digunakan, pipeline sistem, hasil evaluasi, serta analisis kesalahan. Dengan dokumentasi, hasil proyek

tidak hanya menjadi catatan pribadi, tetapi juga bisa dijadikan referensi bagi orang lain yang ingin mempelajari atau mengembangkan sistem serupa.

Selain dalam bentuk laporan tertulis, dokumentasi juga dapat dilengkapi dengan tabel, diagram alur pipeline, serta contoh hasil ekstraksi yang diperoleh sistem. Penyajian visual ini mempermudah pembaca memahami proses dan hasil proyek secara lebih intuitif.

Presentasi bertujuan untuk membagikan hasil proyek kepada audiens, baik itu dosen, teman sekelas, maupun pihak lain yang berkepentingan. Dalam presentasi, mahasiswa dapat menyoroti poin-poin utama, seperti alasan pemilihan topik, strategi perancangan aturan, serta kekuatan dan kelemahan sistem. Penyajian hasil evaluasi dalam bentuk grafik precision–recall atau confusion matrix akan membuat presentasi lebih meyakinkan.

Selain itu, presentasi juga melatih keterampilan komunikasi mahasiswa dalam menjelaskan aspek teknis kepada audiens yang mungkin tidak semuanya berlatar belakang teknis. Dengan demikian, mahasiswa belajar tidak hanya membangun sistem, tetapi juga mengartikulasikan manfaat dan keterbatasannya secara jelas dan persuasif.

Refleksi adalah tahap di mana mahasiswa menilai kembali pengalaman mereka selama mengerjakan proyek. Pertanyaan yang bisa dijawab dalam refleksi antara lain: Apa tantangan terbesar dalam menyusun aturan? Bagian mana dari pipeline yang paling sulit diterapkan? Bagaimana kualitas dataset memengaruhi hasil ekstraksi? Apa saja kesalahan sistem yang paling sering muncul, dan bagaimana cara memperbaikinya?

Refleksi juga dapat diarahkan pada aspek pembelajaran. Mahasiswa bisa menilai bagaimana teori yang dipelajari di kelas ternyata memiliki tantangan saat diterapkan pada data nyata, atau bagaimana pendekatan rule-based memiliki kelebihan dari sisi transparansi tetapi juga keterbatasan dalam menghadapi variasi bahasa. Refleksi ini sangat penting agar mahasiswa tidak hanya berhenti pada hasil, tetapi juga menyadari proses pembelajaran yang mereka lalui.

Dengan mendokumentasikan, mempresentasikan, dan merefleksikan proyek mini, mahasiswa menyelesaikan seluruh siklus pengembangan sistem ekstraksi informasi berbasis aturan. Dari proses ini, mereka belajar bahwa membangun sistem bukan hanya soal menulis aturan dan menjalankan kode, tetapi juga soal merancang metodologi yang jelas, menguji hasil dengan data nyata, dan mengomunikasikan temuan secara ilmiah.

Bab ini sekaligus menutup rangkaian pembahasan dalam buku ajar ini. Melalui mini project, mahasiswa diharapkan tidak hanya memahami konsep-konsep dasar hingga lanjutan, tetapi juga mampu mengaplikasikan, mengevaluasi, serta mengkritisi metode rule-based extraction. Pengalaman ini akan menjadi bekal berharga untuk melangkah lebih jauh ke arah penelitian maupun implementasi praktis di bidang ekstraksi informasi dan pemrosesan bahasa alami.

Latihan

A. Pertanyaan Pemahaman Konsep

1. Mengapa penerapan sistem IE perlu disesuaikan dengan domain tertentu (misalnya bencana, ekonomi, kesehatan)?
2. Sebutkan tiga faktor utama yang memengaruhi keberhasilan implementasi sistem IE di dunia nyata.
3. Bagaimana perbedaan strategi IE untuk teks formal (berita, laporan) dan teks informal (media sosial)?
4. Jelaskan pentingnya proses validasi manual dalam tahap uji lapangan sistem IE.
5. Apa yang dimaksud dengan *domain adaptation* dalam konteks penerapan IE?
6. Mengapa sistem IE perlu dikaitkan dengan *visual analytics* atau *dashboard monitoring*?
7. Bagaimana kolaborasi antara ahli bahasa dan insinyur data dapat meningkatkan kualitas sistem IE?

B. Studi Kasus 1 – Analisis Berita Bencana

Kasus: Pusat Informasi Bencana Nasional ingin memantau peristiwa bencana alam dari berita daring harian.

Tugas:

1. Rancang pipeline sistem IE untuk mendeteksi jenis bencana, lokasi, waktu, dan dampaknya.
2. Gunakan contoh teks berita berikut:

   "Banjir melanda Kabupaten Demak pada Minggu pagi, menyebabkan puluhan rumah terendam air hingga satu meter."
3. Lakukan ekstraksi entitas dan relasi seperti berikut:

Elemen	**Nilai**	**Jenis**	**Keterangan**
Event	banjir	Event Type	Bencana alam

Location	Kabupaten Demak	LOC	Lokasi kejadian
Time	Minggu pagi	DATE	Waktu kejadian
Effect	puluhan rumah terendam air	Effect	Dampak

4. Jelaskan bagaimana sistem IE dapat membantu lembaga pemerintah membuat laporan otomatis bencana.
5. Diskusikan keterbatasan sistem ketika berita mengandung ambiguitas (misalnya, tidak menyebutkan lokasi secara eksplisit).

C. Studi Kasus 2 – Analisis Opini Publik di Media Sosial

Kasus: Dinas Pariwisata ingin mengetahui persepsi masyarakat terhadap destinasi wisata baru.

Tugas:

1. Rancang sistem *Information Extraction* terintegrasi dengan *Sentiment Analysis*.
2. Gunakan 3 contoh unggahan berikut:
   - "Pantai baru di Banyuwangi ini keren banget 😍 tapi agak susah dijangkau."
   - "Harga tiketnya murah, tapi fasilitas kurang lengkap."
   - "Pemandangan bagus, cocok buat liburan keluarga!"
3. Ekstraksi informasi dan sentimen ke dalam tabel:

Elemen	**Nilai**	**Jenis**	**Polaritas**
Location	Banyuwangi	LOC	-
Feature	pantai baru	Object	-
Opinion	keren banget	Sentiment	Positif
Opinion	susah dijangkau	Sentiment	Negatif
Opinion	murah	Sentiment	Positif

Opinion	fasilitas kurang lengkap	Sentiment	Negatif

4. Diskusikan bagaimana hasil analisis ini dapat digunakan untuk pengambilan keputusan promosi wisata.
5. Jelaskan bagaimana sistem dapat menangani teks tidak baku, emoji, dan bahasa gaul.

D. Studi Kasus 3 – Analisis Berita Ekonomi

Kasus: Sistem IE digunakan untuk mengekstraksi hubungan antara perusahaan, produk, dan peristiwa ekonomi.

Tugas:

1. Ambil contoh kalimat:

   "PT Telkom Indonesia meluncurkan layanan 5G komersial pertama di Jakarta pada 2025."

2. Identifikasi entitas dan relasi berikut:

Entitas 1	**Jenis**	**Relasi**	**Entitas 2**	**Jenis**
PT Telkom Indonesia	Organization	meluncurkan	layanan 5G komersial	Product
layanan 5G komersial	Product	berlokasi di	Jakarta	Location
layanan 5G komersial	Product	mulai beroperasi	2025	Date

3. Jelaskan bagaimana sistem IE ini dapat mendukung pembuatan *market intelligence dashboard*.
4. Analisis tantangan yang muncul jika sistem diterapkan pada berita multibahasa atau jargon industri.

E. Diskusi / Refleksi

1. Menurut Anda, bidang apa yang paling diuntungkan oleh penerapan *Information Extraction*?
2. Bagaimana sistem IE dapat dikembangkan untuk menjadi bagian dari *decision support system* (DSS)?
3. Diskusikan potensi integrasi IE dengan teknologi *knowledge graph* untuk representasi pengetahuan yang lebih kaya.
4. Bagaimana langkah strategis untuk menjembatani riset IE di kampus dengan implementasi di industri dan pemerintahan?

DAFTAR PUSTAKA

1. Bird, S., Klein, E., & Loper, E. (2009). *Natural Language Processing with Python*. O'Reilly Media.
2. Jurafsky, D., & Martin, J. H. (2023). *Speech and Language Processing* (3rd ed.). Pearson.
3. Manning, C. D., & Schütze, H. (1999). *Foundations of Statistical Natural Language Processing*. MIT Press.
4. Aggarwal, C. C., & Zhai, C. (2012). *Mining Text Data*. Springer.
5. Feldman, R., & Sanger, J. (2007). *The Text Mining Handbook: Advanced Approaches in Analyzing Unstructured Data*. Cambridge University Press.
6. Hearst, M. A. (1999). *Search User Interfaces*. Cambridge University Press.
7. Goldberg, Y. (2017). *Neural Network Methods for Natural Language Processing*. Morgan & Claypool Publishers.
8. Cambria, E., & White, B. (2014). *Jumping NLP Curves: A Review of Natural Language Processing Research*. Springer.
9. Eisenstein, J. (2019). *Introduction to Natural Language Processing*. MIT Press.
10. Russell, S., & Norvig, P. (2021). *Artificial Intelligence: A Modern Approach* (4th ed.). Pearson.
11. Harrington, P. (2012). *Machine Learning in Action*. Manning Publications.
12. Miner, G., Delen, D., Elder, J., Fast, A., Hill, T., & Nisbet, R. (2012). *Practical Text Mining and Statistical Analysis for Non-structured Text Data Applications*. Academic Press.
13. Baeza-Yates, R., & Ribeiro-Neto, B. (2011). *Modern Information Retrieval: The Concepts and Technology Behind Search* (2nd ed.). Addison-Wesley.

14. Hotho, A., Nürnberger, A., & Paaß, G. (2005). *Text Mining: Theoretical Aspects and Applications*. Springer.
15. Witten, I. H., Frank, E., Hall, M. A., & Pal, C. J. (2016). *Data Mining: Practical Machine Learning Tools and Techniques* (4th ed.). Morgan Kaufmann.
16. Mikolov, T., Grave, E., Bojanowski, P., Puhrsch, C., & Joulin, A. (2018). *Advances in Word Representations*. Morgan & Claypool Publishers.
17. Han, J., Kamber, M., & Pei, J. (2011). *Data Mining: Concepts and Techniques* (3rd ed.). Morgan Kaufmann.

Buku ajar ini disusun sebagai sumber belajar komprehensif dan aplikatif bagi mahasiswa dan praktisi yang ingin memahami bagaimana informasi dapat diekstraksi secara otomatis dari teks menggunakan pendekatan komputasional. Topik utama buku ini adalah Ekstraksi Informasi (Information Extraction/IE), salah satu cabang penting dari Natural Language Processing (NLP) yang berfungsi mengubah data teks tak terstruktur menjadi data terorganisasi untuk analisis dan pengambilan keputusan

Buku ini membahas secara sistematis mulai dari konsep dasar IE, preprocessing teks, feature engineering, POS tagging dan analisis SPOK, hingga pendekatan rule-based dan metode lanjutan seperti machine learning dan deep learning dalam ekstraksi informasi. Pada buku ajar ini disertakan pula contoh kasus nyata, studi penerapan, serta pembahasan berbagai tools NLP modern agar pembaca dapat melihat relevansi antara teori dan praktik.

Selain memperkaya aspek teoretis, buku ini juga berfungsi sebagai panduan praktis membangun sistem berbasis teks, menjadikannya relevan tidak hanya bagi mahasiswa informatika dan ilmu komputer, tetapi juga bagi dosen, peneliti, dan praktisi industri.

Melalui pendekatan berbasis studi kasus dan penerapan nyata, buku ajar ini diharapkan mampu membantu pembaca memahami dan mengimplementasikan teknik ekstraksi informasi dalam berbagai domain, seperti kesehatan, pemerintahan, media sosial, bisnis, dan keamanan siber serta bidang lainnya.

Pustaka Aksara

ISBN : 978-623-161-585-5