

# Bab 10: Pengembangan Sistem Komputer Berbasis Pemrograman

---

## 10.1 Konsep Dasar Pengembangan Sistem Komputer di Era Kecerdasan Buatan

Pengembangan sistem komputer era konvergensi antara AI dan kehidupan sehari-hari tidak hanya berfokus pada perangkat keras dan perangkat lunak secara terpisah, tetapi pada integrasi menyeluruh antara komputasi, kecerdasan buatan (AI), keamanan, serta konektivitas digital. Sistem komputer semakin diarahkan untuk adaptif, cerdas, aman, dan berkelanjutan.

Sistem komputer berkembang ke arah arsitektur terdistribusi dan komputasi hybrid. Arah Pengembangan Edge Computing : Proses data dilakukan dekat sumber perangkat (IoT), mengurangi latensi. *Cloud-Native Architecture*: Sistem dirancang untuk berjalan di cloud: scalable, fleksibel. *Heterogeneous Computing*: Kombinasi CPU, GPU, TPU untuk AI dan aplikasi berat. Diseluruh pengembangan tujuan utamanya adalah sistem lebih cepat merespons kebutuhan real-time.

Secara umum, pengembangan sistem komputer mengikuti siklus hidup pengembangan sistem atau *System Development Life Cycle* (SDLC). Siklus ini mencakup lima tahapan utama yang saling berkesinambungan, yakni analisis kebutuhan, perancangan sistem, implementasi, pengujian, dan pemeliharaan.

Tahap analisis kebutuhan berfokus pada upaya mengidentifikasi masalah dan kebutuhan pengguna melalui berbagai metode seperti pengumpulan data, observasi, serta wawancara. Tujuan dari tahap ini adalah memastikan bahwa sistem yang dikembangkan sesuai dengan kebutuhan dan tujuan organisasi. Setelah kebutuhan sistem ditetapkan, tahap perancangan sistem atau *system design* dilakukan dengan merancang arsitektur, struktur data, antarmuka pengguna, dan alur logika program. Pada tahap ini pula ditentukan rancangan teknis yang menjadi pedoman dalam implementasi agar sistem dapat berjalan efektif dan efisien.

Tahap implementasi kemudian dilakukan dengan menerjemahkan rancangan sistem ke dalam kode program menggunakan bahasa pemrograman tertentu, seperti *Python*, *Java*, atau *C#*. Proses implementasi membutuhkan ketelitian tinggi dalam menulis kode agar sistem berfungsi sesuai rancangan tanpa kesalahan logika. Setelah itu, tahap pengujian atau *testing* dilakukan untuk menemukan dan memperbaiki kesalahan yang mungkin muncul. Pengujian ini memastikan bahwa sistem beroperasi sesuai dengan kebutuhan pengguna dan bebas dari bug yang dapat mengganggu fungsionalitas.

Tahap terakhir adalah pemeliharaan, yaitu kegiatan yang dilakukan setelah sistem diimplementasikan dan digunakan. Pemeliharaan mencakup pembaruan perangkat lunak, perbaikan kesalahan, serta penyesuaian terhadap perubahan kebutuhan pengguna atau lingkungan teknologi. Pendekatan SDLC dinilai efektif karena menyediakan kerangka kerja yang sistematis dan terukur, sehingga mampu menjaga konsistensi dan kualitas dalam proses pengembangan sistem (Sommerville, 2020).

### **10.1.2 Pendekatan dan Metodologi Pengembangan**

Pengembangan sistem komputer saat ini tidak hanya berfokus pada pembuatan perangkat lunak, tetapi juga integrasi dengan AI, keamanan, *cloud computing*, dan otomatisasi proses. Karena itu diperlukan pendekatan dan metodologi yang adaptif, kolaboratif, dan berorientasi nilai bisnis.

Dalam praktiknya, pengembangan sistem komputer dapat dilakukan dengan berbagai metodologi yang disesuaikan dengan karakteristik proyek dan kebutuhan organisasi. Pendekatan pengembangan sistem modern mencakup : pendekatan iteratif & *incremental*, pendekatan berpusat pada pengguna (*User-Centered Design*), Pendekatan Berbasis Data (*Data-Driven Development*), Pendekatan *Cloud-Native* dan Pendekatan *Secure by Design*. Berikut metodologi yang umum dipakai di era saat ini : *Agile*, *Scrum*, *MLOps*, *Design Thinking*, *Rapid Application Development (RAD)* dan *Spiral Model*.

Model *Agile* menekankan fleksibilitas dan kolaborasi antara anggota tim dengan menggunakan iterasi singkat yang disebut

*sprints*. Melalui pendekatan ini, pengembang dapat menyesuaikan sistem secara cepat terhadap perubahan kebutuhan pengguna tanpa harus menunggu seluruh tahap selesai.

Selain itu, *DevOps* merupakan pendekatan modern yang menggabungkan proses pengembangan (*development*) dan operasional (*operations*). Tujuannya adalah mempercepat siklus rilis perangkat lunak serta meningkatkan kualitas sistem melalui otomatisasi, kolaborasi lintas tim, dan penerapan prinsip kerja berkelanjutan. Penelitian menunjukkan bahwa penerapan prinsip *Agile* dan *DevOps* dapat meningkatkan efisiensi tim, mempercepat waktu pengiriman perangkat lunak, serta memperkuat ketahanan sistem terhadap kesalahan operasional (Ahmad et al., 2021).

### **10.1.3 Aspek Kualitas dan Pemeliharaan Sistem**

Kualitas sistem komputer tidak hanya bergantung pada keberhasilan fungsi teknisnya, tetapi juga pada sejumlah aspek penting seperti keandalan (*reliability*), keamanan (*security*), kemudahan penggunaan (*usability*), dan kemampuan adaptasi (*scalability*). Oleh karena itu, pengembangan sistem harus memperhatikan standar kualitas perangkat lunak yang telah ditetapkan secara internasional, seperti *ISO/IEC 25010*. Standar ini mencakup karakteristik utama sistem yang meliputi efisiensi kinerja, kompatibilitas, serta kemudahan pemeliharaan agar sistem dapat digunakan secara berkelanjutan dan terpercaya.

Tahap pemeliharaan memiliki peran yang sangat penting dalam menjamin keberlanjutan operasional sistem setelah implementasi. Kegiatan ini meliputi pembaruan perangkat lunak,

perbaikan bug, serta penambahan fitur untuk menyesuaikan sistem dengan kebutuhan pengguna dan perkembangan teknologi. Dalam era digital saat ini, praktik *continuous integration* dan *continuous delivery* (CI/CD) menjadi strategi penting dalam menjaga stabilitas sistem. Melalui pendekatan ini, pembaruan sistem dapat dilakukan secara otomatis, cepat, dan teruji dengan baik sehingga tidak mengganggu layanan utama yang sedang berjalan.

Secara keseluruhan, pengembangan sistem komputer berbasis pemrograman bukan hanya sekadar proses teknis, melainkan juga kegiatan manajerial yang memerlukan koordinasi, perencanaan, dan evaluasi yang matang. Dengan menerapkan metodologi pengembangan yang tepat serta memperhatikan aspek kualitas dan keberlanjutan, sistem komputer dapat berfungsi secara efektif untuk mendukung kinerja organisasi dan menjawab tantangan masyarakat di era digital yang terus berkembang.

## **10.2 Tahapan Pengembangan Sistem**

### **10.2.1 Analisis Kebutuhan**

Tahap pertama dalam pengembangan sistem adalah analisis kebutuhan (*requirement analysis*). Pada tahap ini, pengembang berupaya memahami permasalahan yang dihadapi pengguna serta kebutuhan fungsional dan nonfungsional dari sistem yang akan dibangun. Analisis kebutuhan melibatkan proses pengumpulan informasi melalui wawancara, observasi, survei, dan studi dokumentasi. Tujuannya adalah menghasilkan spesifikasi kebutuhan

yang jelas, lengkap, dan dapat dipahami oleh seluruh pihak yang terlibat.

Dalam praktik modern, pendekatan *user-centered design* sering diterapkan untuk memastikan bahwa sistem yang dikembangkan benar-benar menjawab kebutuhan pengguna. Pendekatan ini menekankan keterlibatan aktif pengguna sejak tahap awal, sehingga kesesuaian antara produk akhir dan ekspektasi pengguna dapat terjamin. Menurut penelitian terkini, keterlibatan pengguna secara berkelanjutan dalam analisis kebutuhan berkontribusi signifikan terhadap keberhasilan proyek teknologi informasi karena mengurangi risiko miskomunikasi dan kesalahan spesifikasi (Alamri & Alharbi, 2023).

### **10.2.2 Perancangan Sistem**

Tahap perancangan sistem (*system design*) berfungsi sebagai jembatan antara hasil analisis kebutuhan dan implementasi teknis. Perancangan meliputi pembuatan model arsitektur sistem, desain basis data, serta antarmuka pengguna (*user interface*). Pada tahap ini, pengembang menentukan bagaimana komponen perangkat lunak saling berinteraksi untuk memenuhi kebutuhan yang telah diidentifikasi.

Desain yang baik harus memperhatikan prinsip modularitas, skalabilitas, dan keamanan. Selain itu, penggunaan diagram seperti *Unified Modeling Language* (UML) membantu memvisualisasikan struktur dan alur proses sistem secara sistematis. Tren terkini menunjukkan bahwa banyak organisasi kini mengadopsi *model-driven engineering* untuk mempercepat transisi dari desain ke

implementasi, karena pendekatan tersebut mampu menghasilkan kode dasar secara otomatis dari model konseptual (Khan & Ahmad, 2022).

### **10.2.3 Implementasi**

Tahap implementasi merupakan proses mengubah desain menjadi kode program nyata. Pengembang menulis kode menggunakan bahasa pemrograman yang sesuai dengan kebutuhan sistem, seperti Python, Java, atau C#. Tahap ini memerlukan penerapan prinsip rekayasa perangkat lunak yang baik, seperti penggunaan *version control*, pengujian unit, dan dokumentasi kode.

Dalam pendekatan modern seperti *Agile Development*, implementasi tidak dilakukan sekaligus, melainkan secara iteratif melalui siklus pengembangan yang singkat (*sprint*). Pendekatan ini memungkinkan tim untuk meninjau hasil secara berkala dan melakukan perbaikan cepat berdasarkan umpan balik pengguna. Dengan cara ini, sistem yang dikembangkan menjadi lebih adaptif terhadap perubahan kebutuhan dan lingkungan teknologi yang dinamis.

### **10.2.4 Pengujian dan *Debugging***

Setelah implementasi, tahap berikutnya adalah pengujian (*testing*) dan *debugging*. Tujuan utamanya adalah memastikan bahwa sistem bekerja sesuai spesifikasi dan bebas dari kesalahan (*error*). Pengujian dilakukan melalui beberapa pendekatan, seperti pengujian unit (*unit testing*), integrasi (*integration testing*), dan pengujian penerimaan pengguna (*user acceptance testing*).

*Debugging* adalah proses sistematis untuk menemukan dan memperbaiki kesalahan yang muncul selama pengujian. Penggunaan alat bantu seperti *automated testing tools* dan *continuous integration systems* kini menjadi standar dalam praktik rekayasa perangkat lunak modern. Melalui penerapan metodologi pengujian yang komprehensif, kualitas sistem dapat ditingkatkan sekaligus menurunkan biaya pemeliharaan di tahap berikutnya.

### **10.2.5 Pemeliharaan Sistem**

Tahap terakhir adalah pemeliharaan (*maintenance*), yaitu proses memastikan bahwa sistem tetap berfungsi optimal setelah diterapkan. Pemeliharaan mencakup kegiatan memperbaiki *bug*, meningkatkan performa, menyesuaikan sistem terhadap perubahan kebutuhan pengguna, serta melakukan pembaruan terhadap teknologi yang digunakan.

Dalam konteks pengembangan jangka panjang, pemeliharaan juga mencakup evaluasi berkala terhadap efektivitas sistem. Sistem yang baik harus memiliki fleksibilitas untuk beradaptasi dengan perubahan organisasi, regulasi, maupun perkembangan teknologi informasi. Oleh karena itu, dokumentasi dan manajemen versi yang rapi menjadi komponen penting agar pemeliharaan dapat dilakukan secara efisien dan berkelanjutan.

## 10.3 Bahasa Pemrograman dan Alat Pengembangan

### 10.3.1 Perkembangan Bahasa Pemrograman dalam Sistem Komputer

Bahasa pemrograman merupakan elemen fundamental dalam sistem komputer, berfungsi sebagai medium komunikasi antara manusia dan mesin. Melalui bahasa ini, pengembang dapat menuliskan instruksi logis yang akan diterjemahkan menjadi operasi yang dapat dijalankan oleh prosesor. Perkembangan bahasa pemrograman telah mengalami evolusi signifikan dari generasi awal berbasis *machine code* hingga hadirnya bahasa tingkat tinggi yang lebih mudah dipahami, efisien, serta adaptif terhadap berbagai kebutuhan teknologi modern.

Dalam konteks pengembangan sistem masa kini, pemilihan bahasa pemrograman tidak hanya didasarkan pada sintaksis atau efisiensi komputasi, tetapi juga pada kompatibilitas lintas platform, keamanan, dukungan komunitas, serta ketersediaan pustaka (*library*) dan kerangka kerja (*framework*). Bahasa seperti *Python*, *Java*, *C/C++*, dan *JavaScript* menonjol karena memiliki ekosistem luas, dokumentasi lengkap, dan dukungan komunitas global yang kuat. Keempat bahasa tersebut menjadi fondasi utama dalam berbagai bidang, mulai dari pengembangan aplikasi web, sistem enterprise, hingga kecerdasan buatan (*artificial intelligence*) dan sistem tertanam (*embedded systems*) (Hassan et al., 2023).

### 10.3.2 Karakteristik dan Kelebihan Bahasa Pemrograman Populer

*Python* merupakan salah satu bahasa pemrograman yang paling fleksibel dan mudah dipelajari. Bahasa ini dirancang dengan sintaksis sederhana yang menyerupai bahasa alami, sehingga ramah bagi pemula sekaligus efisien bagi pengembang profesional. *Python* mendukung paradigma pemrograman berorientasi objek, prosedural, maupun fungsional, dan memiliki pustaka luas seperti *NumPy*, *TensorFlow*, serta *Django* yang menjadikannya relevan untuk analisis data, *machine learning*, maupun pengembangan web.

Sementara itu, *Java* menempati posisi penting dalam dunia industri karena kemampuannya untuk dijalankan di berbagai platform melalui konsep “*write once, run anywhere*”. Dengan dukungan *Java Virtual Machine* (JVM), program berbasis *Java* dapat berjalan secara konsisten di berbagai sistem operasi. Bahasa ini banyak digunakan dalam pengembangan sistem *enterprise*, aplikasi *Android*, serta sistem keuangan berskala besar yang memerlukan keamanan dan stabilitas tinggi.

Bahasa *C* dan *C++* tetap menjadi pilihan utama untuk pengembangan perangkat lunak tingkat sistem, termasuk sistem operasi, *firmware*, dan aplikasi dengan kebutuhan efisiensi tinggi. *C* memberikan kendali langsung terhadap manajemen memori, sedangkan *C++* memperluas fungsionalitasnya dengan pendekatan berorientasi objek. Keduanya menawarkan performa tinggi yang tetap relevan di tengah dominasi bahasa modern karena mampu memberikan kontrol penuh terhadap perangkat keras.

Adapun *JavaScript* menjadi bahasa dominan dalam pengembangan web modern. Awalnya digunakan untuk menambah interaktivitas halaman web di sisi klien, kini *JavaScript* telah berkembang menjadi bahasa yang serbaguna berkat hadirnya *Node.js* yang memungkinkan eksekusi di sisi server. Dengan dukungan berbagai kerangka kerja seperti *React*, *Angular*, dan *Vue.js*, *JavaScript* menjadi fondasi utama bagi pengembangan *single-page application* (SPA) dan platform web interaktif.

### **10.3.3 Alat Pengembangan: Integrated Development Environment (IDE)**

Selain penguasaan bahasa pemrograman, pengembang juga membutuhkan alat bantu yang efisien untuk mendukung proses penulisan dan pemeliharaan kode. *Integrated Development Environment* (IDE) merupakan perangkat lunak yang mengintegrasikan berbagai fungsi penting seperti editor kode, *compiler* atau *interpreter*, *debugger*, serta sistem manajemen proyek dalam satu kesatuan antarmuka. Keberadaan IDE membantu pengembang mempercepat proses *coding*, mengurangi kesalahan, dan meningkatkan kualitas perangkat lunak.

Beberapa IDE populer yang banyak digunakan saat ini meliputi *Visual Studio Code* (VS Code), *IntelliJ IDEA*, dan *Eclipse*. VS Code dikembangkan oleh Microsoft dan dikenal karena kemampuannya mendukung berbagai bahasa pemrograman melalui sistem ekstensi yang fleksibel. *IntelliJ IDEA* menjadi pilihan utama bagi pengembang *Java* karena fitur otomatisasi pengkodean dan integrasi yang kuat dengan sistem kontrol versi seperti *Git*.

Sementara itu, *Eclipse* tetap populer di kalangan akademisi dan industri karena bersifat *open source* serta mendukung berbagai plugin untuk bahasa dan platform yang berbeda. Dengan memanfaatkan IDE, pengembang dapat melakukan *debugging*, pengujian kode, serta pengelolaan proyek secara efisien dalam satu lingkungan kerja terpadu.

#### **10.3.4 Tantangan dan Arah Perkembangan ke Depan**

Walaupun berbagai bahasa dan alat pengembangan telah mencapai tingkat kematangan tinggi, dunia pemrograman tetap menghadapi tantangan baru seiring percepatan inovasi digital. Isu keamanan siber, interoperabilitas lintas platform, serta efisiensi energi dalam pemrosesan menjadi perhatian utama pengembang perangkat lunak. Selain itu, tren *low-code development* dan pemrograman berbantuan kecerdasan buatan mulai mengubah paradigma pengembangan tradisional.

Kemunculan teknologi seperti *GitHub Copilot* dan *AI-based code assistant* menghadirkan peluang baru dalam meningkatkan produktivitas, tetapi sekaligus menuntut pengembang untuk memiliki keterampilan berpikir kritis dan adaptif terhadap alat generatif tersebut (Zhou & Xu, 2021). Dengan demikian, pembelajaran bahasa pemrograman dan pemanfaatan IDE modern bukan hanya berorientasi pada keterampilan teknis, tetapi juga pada kesiapan menghadapi transformasi digital yang berkelanjutan di masa depan.

## **10.4 Penerapan Sistem dalam Dunia Nyata**

### **10.4.1 Konsep Umum Penerapan Sistem Komputer**

Sistem komputer berbasis pemrograman telah menjadi komponen penting dalam hampir seluruh aspek kehidupan modern. Melalui kombinasi perangkat keras (*hardware*) dan perangkat lunak (*software*), sistem ini berfungsi mengotomatisasi proses, mengolah data, serta mendukung pengambilan keputusan di berbagai sektor. Penerapan sistem komputer tidak hanya terbatas pada aktivitas bisnis atau industri, tetapi juga meluas ke bidang pendidikan, kesehatan, pemerintahan, dan layanan publik.

Setiap sistem komputer dirancang untuk memenuhi kebutuhan tertentu berdasarkan analisis pengguna, tujuan operasional, serta ketersediaan sumber daya teknologi. Dalam konteks ini, pendekatan *user-centered design* menjadi penting agar sistem yang dikembangkan benar-benar sesuai dengan kebutuhan pengguna. Kegagalan memahami kebutuhan tersebut sering kali menjadi penyebab utama rendahnya tingkat adopsi dan efektivitas suatu sistem. Oleh karena itu, proses rekayasa perangkat lunak (*software engineering*) harus melibatkan pengguna sejak tahap perancangan hingga evaluasi sistem (Pressman & Maxim, 2020).

### **10.4.2 Sistem Informasi Akademik dan Keuangan**

Salah satu penerapan sistem komputer yang paling umum terdapat pada sektor pendidikan melalui sistem informasi akademik. Sistem ini berfungsi mengelola data mahasiswa, kurikulum, jadwal perkuliahan, serta hasil belajar. Keunggulan utamanya terletak pada

kemampuan mengintegrasikan seluruh proses akademik secara real time sehingga memudahkan koordinasi antara dosen, mahasiswa, dan administrasi. Selain itu, penerapan sistem ini juga mendukung efisiensi operasional institusi pendidikan serta meningkatkan transparansi layanan akademik.

Di sektor keuangan dan perbankan, sistem komputer memegang peran strategis dalam mengelola transaksi dan data nasabah. Aplikasi keuangan modern mengintegrasikan teknologi *application programming interface* (API) dan *machine learning* untuk mendeteksi anomali transaksi serta meningkatkan keamanan siber. Keandalan sistem menjadi kunci keberhasilan dalam menjaga kepercayaan pengguna dan stabilitas lembaga keuangan. Oleh karena itu, setiap sistem harus memenuhi standar keamanan yang ketat, termasuk enkripsi data, autentikasi berlapis, dan *real-time monitoring* terhadap ancaman siber. Penelitian terkini menunjukkan bahwa kualitas arsitektur sistem dan keamanan algoritmik menjadi faktor penentu keberhasilan sistem perbankan digital (Rahman et al., 2022).

#### **10.4.3 Sistem Kendali Industri dan Aplikasi Mobile**

Dalam dunia industri, sistem komputer berperan penting dalam sistem kendali otomatis (*industrial control systems*). Melalui pemrograman berbasis logika seperti *Programmable Logic Controller* (PLC), sistem ini mengatur proses produksi, memantau kondisi mesin, serta memastikan efisiensi energi. Penerapan teknologi *Internet of Things* (IoT) semakin memperluas kemampuan sistem kendali dengan menghadirkan integrasi antara mesin dan

jaringan data secara waktu nyata. Hal ini memungkinkan pengawasan jarak jauh, prediksi kerusakan, serta peningkatan produktivitas melalui analisis berbasis data besar (*big data analytics*).

Sementara itu, aplikasi mobile dan web menjadi wujud paling kasatmata dari penerapan sistem komputer di kehidupan sehari-hari. Melalui antarmuka yang interaktif, sistem ini memfasilitasi berbagai aktivitas mulai dari komunikasi, perdagangan, pendidikan, hingga layanan kesehatan. Penerapan prinsip *responsive design* dan arsitektur *cloud computing* memungkinkan sistem untuk diakses dari berbagai perangkat tanpa kehilangan kinerja maupun keamanan. Keberhasilan aplikasi mobile ditentukan oleh kemudahan penggunaan (*usability*), kecepatan respons, serta kemampuan sistem menyesuaikan dengan kebutuhan pengguna yang dinamis.

#### **10.4.4 Sistem Keamanan, Pengawasan, dan Kualitas Kode**

Aspek keamanan menjadi elemen fundamental dalam setiap penerapan sistem komputer. Sistem keamanan modern menggabungkan teknologi pengenalan wajah (*facial recognition*), biometrik, dan analisis perilaku pengguna untuk meningkatkan tingkat deteksi ancaman. Di lingkungan industri maupun publik, sistem pengawasan berbasis komputer berfungsi menjaga keselamatan, mengurangi risiko kebocoran informasi, dan mendukung pengambilan keputusan cepat melalui pemantauan visual serta data terintegrasi.

Namun, keberhasilan sistem tidak hanya ditentukan oleh teknologi yang digunakan, melainkan juga oleh kualitas kode yang membangunnya. Kode yang baik harus efisien, mudah dipelihara, aman, dan sesuai standar pengembangan perangkat lunak. Kesalahan kecil dalam logika atau struktur kode dapat berdampak signifikan terhadap keandalan sistem. Oleh karena itu, proses *code review*, pengujian unit (*unit testing*), dan evaluasi berkelanjutan merupakan bagian esensial dalam memastikan stabilitas serta keberlanjutan sistem di dunia nyata.

## 10.5 Latihan Soal

1. Sebutkan dan jelaskan lima tahapan pengembangan sistem komputer!
2. Apa perbedaan mendasar antara bahasa pemrograman Python dan C++?
3. Mengapa penting melakukan pengujian dan debugging pada sistem?
4. Berikan contoh sistem berbasis pemrograman di bidang pendidikan!
5. Apa saja peran IDE dalam pengembangan sistem perangkat lunak?