



BERITA ACARA SIDANG PENDADARAN TUGAS AKHIR

S/UBL/FTI/0246/I/26

Pada hari ini, Senin 26 Januari 2026 telah dilaksanakan Ujian Sidang Pendadaran Tugas Akhir sebagai berikut:

Judul: SISTEM SMART CLASS BERBASIS INTERNET OF THINGS MENGGUNAKAN ESP32 DEVKIT V1 BERBASIS WEB

Nama : Rahmat Hidayat Nasution
NIM : 2113500025
Dosen Pembimbing : Ir. Gatot Purwanto, M.M.

Berdasarkan penilaian pada Demo Alat, Presentasi, Penulisan, Penguasaan Materi, Penguasaan Program maka Mahasiswa tersebut di atas dinyatakan:

LULUS

dengan nilai angka : **76** huruf : **B+**

Mahasiswa tersebut di atas wajib menyerahkan hasil perbaikan tulisan Tugas Akhir dalam bentuk terjilid sesuai dengan Panduan Perbaikan Tugas Akhir, selambat-lambatnya Senin 09 Februari 2026.

Panitia Penguji:

- 1 Ketua Irawan, S.Kom., M.Kom.
- 2 Anggota Riri Irawati, S.Kom., M.Kom.
- 3 Moderator Ir. Gatot Purwanto, M.M.

Keterangan:

Nilai Huruf: A:85-100 A-:80-84,99 B+:75-79,99 B:70-74,99 B-:65-69,99 C:60-64,99 D:40-59,99 E-:0-39,99



UNIVERSITAS BUDI LUHUR
FAKULTAS TEKNOLOGI INFORMASI

Kartu Bimbingan Tugas Akhir

NIM: 2113500025

Nama: Rahmat Hidayat Nasution

Pembimbing: Ir. Gatot Purwanto, M.M.

No.	Tanggal	Materi
1	29-09-2025	KONSULTASI ALAT
2	02-10-2025	JUDUL TA
3	02-01-2026	BAB 1
4	05-01-2026	BAB 2
5	09-01-2026	BAB 3
6	13-01-2026	BAB 4
7	15-01-2026	BAB 4
8	19-01-2026	BAB 5



LEMBAR PENGESAHAN

Nama : Rahmat Hidayat Nasution
Nomor Induk Mahasiswa : 2113500025
Program Studi : Sistem Komputer
Bidang Peminatan : Robotics
Jenjang Studi : Strata 1
Judul : SISTEM SMART CLASS BERBASIS INTERNET OF THINGS
MENGUNAKAN ESP32 DEVKIT V1 BERBASIS WEB



Laporan Tugas Akhir ini telah disetujui, disahkan dan direkam secara elektronik sehingga tidak memerlukan tanda tangan tim penguji.

Jakarta, Senin 26 Januari 2026

Tim Penguji:

Ketua : Irawan, S.Kom., M.Kom.
Anggota : Riri Irawati, S.Kom., M.Kom.
Pembimbing : Ir. Gatot Purwanto, M.M.
Ketua Program Studi : Rizky Pradana, S.Kom., M.Kom.

**SISTEM *SMART CLASS* BERBASIS *INTERNET OF THINGS (IOT)*
MENGUNAKAN *ESP32 DEVKIT V1* BERBASIS *WEB***

TUGAS AKHIR



**Oleh:
Rahmat Hidayat Nasution
NIM: 2113500025**

**PROGRAM STUDI SISTEM KOMPUTER
FAKULTAS TEKNOLOGI INFORMASI
UNIVERSITAS BUDI LUHUR**

**JAKARTA
2026**

PENGESAHAN TUGAS AKHIR



PROGRAM STUDI SISTEM KOMPUTER
FAKULTAS TEKNOLOGI INFORMASI
UNIVERSITAS BUDI LUHUR

LEMBAR PENGESAHAN

Nama : Rahmat Hidayat Nasution
Nomor Induk Mahasiswa : 2113500025
Program Studi : Sistem Komputer
Bidang Peminatan : Robotics
Jenjang Studi : Strata1
Judul : SISTEM SMART CLASS BERBASIS INTERNET OF THINGS
MENGUNAKAN ESP32 DEVKIT V1 BERBASIS WEB



Laporan Tugas Akhir ini telah disetujui, disahkan dan direkam secara elektronik sehingga tidak memerlukan tanda tangan tim penguji.

Jakarta, Senin 26 Januari 2026

Tim Penguji:

Ketua : Irawan, S.Kom., M.Kom.
Anggota : Riri Irawati, S.Kom.,
M.Kom.
Pembimbing : Ir. Gatot Purwanto, M.M.
Ketua Program Studi : Rizky Pradana, S.Kom.,
M.Kom.

ABSTRAK

Perkembangan teknologi *Internet of Things (IoT)* memungkinkan penerapan sistem otomasi dan monitoring pada berbagai bidang, termasuk lingkungan pendidikan. Permasalahan yang sering ditemukan pada ruang kelas konvensional adalah pengelolaan fasilitas yang masih dilakukan secara manual, seperti pengendalian kipas dan sistem keamanan akses pintu, sehingga kurang efisien dan sulit dipantau secara terpusat. Penelitian ini bertujuan untuk merancang dan membangun *prototype* sistem *Smart Class* berbasis *Internet of Things* yang mampu melakukan monitoring dan pengendalian fasilitas ruang kelas secara otomatis maupun manual melalui antarmuka berbasis *web*. Sistem ini menggunakan mikrokontroler *ESP32 DevKit V1* sebagai pusat kendali yang terhubung dengan sensor *DHT22* untuk memantau suhu dan kelembapan ruangan, *keypad* sebagai sistem keamanan akses pintu, serta *Liquid Crystal Display (LCD)* sebagai media penampil informasi kondisi kelas. Sistem dilengkapi dengan aktuator berupa motor *servo* untuk mekanisme buka dan tutup pintu serta modul *relay* yang digunakan untuk mengendalikan kipas ruang kelas. Komunikasi data antara perangkat dan *server* dilakukan melalui jaringan *WiFi* menggunakan protokol *Hypertext Transfer Protocol (HTTP)* dengan metode *GET* dan *POST*, sehingga data dapat dimonitor dan perangkat dapat dikendalikan secara jarak jauh melalui *dashboard web*. Hasil pengujian menunjukkan bahwa sistem *Smart Class* yang dirancang mampu bekerja dengan baik sesuai dengan fungsinya. Sistem dapat memantau kondisi suhu dan kelembapan ruang kelas secara *real-time*, mengendalikan kipas dan akses pintu baik pada mode otomatis maupun manual, serta meningkatkan efisiensi pengelolaan ruang kelas. Dengan demikian, penelitian ini menghasilkan sebuah *prototype Smart Class* berbasis *Internet of Things* yang dapat digunakan sebagai referensi awal dalam pengembangan sistem ruang kelas cerdas di lingkungan pendidikan.

Kata kunci: *Smart Class, Internet of Things, ESP32, Otomasi Kelas, Monitoring Ruang Kelas*

DAFTAR ISI

PENGESAHAN TUGAS AKHIR.....	i
PERNYATAAN TIDAK PLAGIAT.....	ii
ABSTRAK	iii
KATA PENGANTAR.....	iv
DAFTAR TABEL	v
DAFTAR GAMBAR	vi
DAFTAR SIMBOL	vii
DAFTAR ISI.....	viii
BAB I PENDAHULUAN.....	1
1.1 Latar Belakang.....	1
1.2 Rumusan Masalah.....	2
1.3 Batasan Masalah.....	2
1.4 Tujuan Penelitian.....	3
1.5 Manfaat Penelitian.....	3
1.6 Sistematika Penulisan.....	3
BAB II LANDASAN TEOR.....	5
2.1 Studi Literatur.....	5
2.2 Smart Class.....	8
2.3 Internet Of Things (IoT).....	8
2.4 Arduino IDE.....	9
2.5 ESP32 DevKit V1.....	10
2.6 Sensor DHT 22.....	11
2.7 Sensor LDR (<i>Light Dependent Resistor</i>).....	12
2.8 Keypad 4x4.....	14
2.9 Servo.....	15
2.10 Fan.....	16
2.11 Relay Optocoupler 2 channel.....	17
2.12 LED.....	18
2.13 Liquid Crystal Display I2C 20x4.....	19
2.14 Web Hosting.....	20
BAB III METODOLOGI PENELITIAN.....	21
3.1 Data Penelitian.....	21
3.2 Rancangan Diagram Blok.....	22
3.3 Rangkaian Perangkat Keras.....	23
3.3.1 Rangkaian Keseluruhan.....	23
3.3.2 Rangkaian keypad 4x4.....	25
3.3.3 Rangkaian Sensor DHT 22.....	26
3.3.4 Rangkaian LDR.....	27
3.3.5 Rangkaian Servo.....	28
3.3.6 Rangkaian LED.....	29
3.3.7 Rangkaian Liquid Crystal Display 20x4 I2C.....	30
3.4 Rancangan Metodologi Pengembangan Software.....	31
3.4.1 Langkah Alur Metodologi.....	31
3.4.2 Formulasi Matematika.....	32
3.5 Flowchart.....	35
3.6 Rancangan Layar.....	36

3.6.1	Layar Dashboard.....	37
3.6.2	Layar Monitoring.....	37
3.6.3	Layar Kontrol.....	39
3.6.4	Layar Pengaturan.....	39
3.7	Rancangan Desain 3D.....	39
3.8	Rancangan Pengujian.....	40
BAB IV IMPLEMENTASI DAN EVALUASI PROGRAM.....		43
4.1	Cara Kerja Sistem.....	43
4.2	Hasil Pengujian Perangkat Keras.....	44
4.2.1	Pengujian keypad 4x4.....	45
4.2.2	Pengujian Sensor DHT 22.....	46
4.2.3	Pengujian Sensor LDR.....	46
4.2.4	Pengujian Motor Servo.....	47
4.2.5	Pengujian Relay dan Fan.....	48
4.2.6	Pengujian Liquid Crystal Display 20x4 I2C.....	48
4.2.7	Pengujian Komunikasi API (GET/POST).....	49
4.3	Analisis Hasil Pengujian.....	49
4.4	Kelebihan Sistem.....	51
4.5	Kekurangan Sistem.....	51
4.6	Foto Sistem.....	52
BAB V PENUTUP.....		54
5.1	Kesimpulan.....	54
5.2	Saran.....	55
DAFTAR PUSTAKA.....		56
LAMPIRAN.....		59

BAB I PENDAHULUAN

1.1 Latar Belakang

Perkembangan teknologi informasi yang semakin pesat telah memberikan dampak besar terhadap berbagai bidang kehidupan, termasuk bidang pendidikan. Salah satu penerapan teknologi yang mulai banyak dikembangkan adalah konsep Smart Class, yaitu ruang kelas yang memanfaatkan teknologi untuk membantu pengelolaan fasilitas dan menciptakan lingkungan belajar yang lebih efektif dan efisien. Namun, pada kenyataannya masih banyak ruang kelas yang pengoperasian fasilitasnya dilakukan secara manual, seperti pengendalian lampu, kipas, dan akses pintu, sehingga kurang praktis dan sulit dipantau secara terpusat.

Seiring dengan berkembangnya teknologi Internet of Things (IoT), berbagai perangkat elektronik kini dapat saling terhubung dan berkomunikasi melalui jaringan internet. Teknologi ini memungkinkan proses pemantauan dan pengendalian perangkat dilakukan secara real-time melalui sistem berbasis web. Beberapa penelitian terdahulu juga telah menunjukkan pentingnya penerapan IoT dalam ruang kelas. Misalnya, penelitian mengenai sistem pemantauan ruang kelas berbasis IoT menunjukkan bahwa integrasi sensor dan antarmuka web mampu memantau kondisi lingkungan dan mengoptimalkan penggunaan energi secara otomatis dengan notifikasi apabila parameter lingkungan melebihi batas tertentu, sehingga mendukung pengelolaan fasilitas ruang kelas yang lebih efektif (Latief et al., 2025). Penelitian lain mengenai penerapan konsep Smart Class di Sekolah Tinggi Teknologi Wastukencana Purwakarta menunjukkan bahwa penggunaan IoT mampu menghubungkan berbagai perangkat elektronik ruang kelas agar dapat dikontrol melalui internet sehingga proses pembelajaran menjadi lebih nyaman dan efisien (Putri Ida Sunaryathy Samad, 2024). Selain itu, pengembangan sistem monitoring ruang kelas berbasis IoT juga telah diterapkan dengan kemampuan pengoperasian fasilitas elektronik dan pemantauan kondisi ruang secara real-time melalui antarmuka web sehingga dapat meminimalkan kerusakan fasilitas dan memaksimalkan efisiensi penggunaannya (Riyanto et al., 2021).

Berdasarkan kondisi tersebut, diperlukan sebuah sistem Smart Class yang mampu mengintegrasikan fungsi monitoring dan pengendalian fasilitas ruang kelas dalam satu kesatuan. Pada penelitian ini dirancang sebuah prototype sistem Smart Class berbasis Internet of Things (IoT) dengan menggunakan mikrokontroler ESP32 DevKit V1 sebagai pengendali utama. Sistem ini memanfaatkan perangkat input berupa sensor DHT22 untuk memantau suhu dan kelembapan ruangan serta keypad sebagai sistem pengaman akses pintu. Selain itu, sistem juga menggunakan Liquid Crystal Display (LCD) I2C sebagai media tampilan informasi kondisi ruang kelas secara langsung.

Perangkat output pada sistem ini berupa motor servo yang digunakan untuk mekanisme buka dan tutup pintu serta modul relay yang digunakan untuk mengendalikan kipas ruang kelas. Seluruh data yang dihasilkan oleh sensor diproses oleh ESP32 dan dikirimkan ke database, kemudian ditampilkan melalui antarmuka berbasis web sehingga dapat dipantau dan dikendalikan oleh pengguna secara jarak jauh.

Sistem ini juga dirancang dengan dua mode pengoperasian, yaitu mode otomatis dan mode manual. Pada mode otomatis, sistem bekerja berdasarkan input dari sensor,

sedangkan pada mode manual pengguna dapat mengendalikan perangkat secara langsung melalui web.

Melalui penelitian ini, diharapkan prototype sistem Smart Class berbasis Internet of Things (IoT) yang dirancang dapat menjadi solusi awal dalam pengelolaan fasilitas ruang kelas secara cerdas dan terintegrasi. Sistem ini diharapkan mampu meningkatkan efisiensi pengelolaan ruang kelas, memberikan kemudahan dalam pemantauan dan pengendalian perangkat, serta menjadi referensi bagi pengembangan sistem Smart Class yang lebih kompleks dan implementatif di lingkungan pendidikan pada masa mendatang.

1.2 Rumusan Masalah

Berdasarkan latar belakang yang telah dijelaskan sebelumnya, maka dapat dirumuskan beberapa permasalahan yang akan dibahas dalam penelitian ini sebagai berikut:

1. Bagaimana merancang dan membangun *prototype* sistem *Smart Class* berbasis *Internet of Things (IoT)* yang mampu memantau dan mengendalikan fasilitas ruang kelas secara terintegrasi?
2. Bagaimana pemanfaatan mikrokontroler *ESP32 DevKit V1* dalam mengolah data sensor dan mengendalikan perangkat ruang kelas melalui sistem berbasis *web*?
3. Bagaimana penerapan mode otomatis dan mode manual pada sistem *Smart Class* berbasis *IoT* dalam pengelolaan fasilitas ruang kelas?

1.3 Batasan Masalah

Agar pembahasan dalam penelitian ini lebih terfokus dan sesuai dengan kemampuan peralatan yang digunakan, maka batasan masalah dalam penelitian ini adalah sebagai berikut:

1. Sistem yang dirancang merupakan *prototype Smart Class* berbasis *IoT* dan belum diimplementasikan secara penuh pada ruang kelas sebenarnya.
2. Perangkat yang dikendalikan dan dipantau dalam sistem ini dibatasi pada beberapa fasilitas ruang kelas, seperti pemantauan kondisi lingkungan dan pengendalian perangkat elektronik melalui relay dan servo.
3. Sistem monitoring dan pengendalian dilakukan melalui antarmuka *web* dengan komunikasi data menggunakan jaringan internet.

1.4 Tujuan Penelitian

Tujuan dari penelitian ini disusun berdasarkan permasalahan yang telah dirumuskan dan kontribusi yang diharapkan dari pembuatan Tugas Akhir, yaitu sebagai berikut:

1. Merancang dan membangun *prototype* sistem *Smart Class* berbasis *Internet of Things (IoT)* menggunakan mikrokontroler *ESP32 DevKit V1*.
2. Mengimplementasikan sistem monitoring dan pengendalian fasilitas ruang kelas secara otomatis dan manual melalui antarmuka *web*.
3. Menguji kinerja *prototype* sistem *Smart Class* berbasis *IoT* dalam memantau dan mengendalikan fasilitas ruang kelas secara real-time.

1.5 Manfaat Penelitian

Manfaat dari penelitian ini diharapkan dapat dirasakan oleh beberapa pihak sebagai berikut:

1. Bagi Institusi Pendidikan, sebagai referensi awal dalam pengembangan sistem *Smart Class* untuk pengelolaan fasilitas ruang kelas yang lebih efisien dan terintegrasi.
2. Bagi Mahasiswa dan Peneliti, sebagai bahan pembelajaran dan referensi dalam pengembangan sistem berbasis *Internet of Things (IoT)* dan *web*.
3. Bagi Pengembang Sistem, sebagai acuan dalam merancang dan mengimplementasikan sistem *monitoring* dan pengendalian ruang berbasis *IoT* yang dapat dikembangkan lebih lanjut.

1.6 Sistematika Penulisan

Sistematika penulisan tugas akhir ini disusun untuk memberikan gambaran secara umum mengenai isi dari setiap bab yang terdapat dalam laporan. Adapun sistematika penulisan tugas akhir ini adalah sebagai berikut:

Bab I Pendahuluan	Bab ini berisi latar belakang penelitian, rumusan masalah, batasan masalah, tujuan penelitian, manfaat penelitian, serta sistematika penulisan yang menjelaskan susunan penulisan laporan tugas akhir secara keseluruhan.
Bab II Tinjauan Pustaka	Bab ini membahas teori-teori pendukung yang berkaitan dengan penelitian, meliputi konsep Internet of Things (IoT), mikrokontroler ESP32, sensor dan aktuator yang digunakan, serta penelitian terdahulu yang relevan sebagai dasar perancangan sistem.
Bab III Metodologi Penelitian	Bab ini menjelaskan tahapan dan metode penelitian yang digunakan dalam perancangan dan pembuatan prototype Sistem Smart Class berbasis Internet of Things. Pembahasan meliputi perancangan sistem, diagram blok, perancangan perangkat keras dan perangkat lunak, rancangan pengujian, serta rancangan layar sistem.
Bab IV Implementasi dan Evaluasi Program	Bab ini membahas hasil implementasi sistem serta evaluasi kinerja berdasarkan pengujian yang telah dilakukan. Pembahasan meliputi cara kerja sistem, hasil pengujian setiap

komponen, analisis hasil pengujian, kelebihan dan kekurangan sistem, serta dokumentasi foto sistem.

Bab V Penutup

Bab ini berisi kesimpulan yang diperoleh dari hasil penelitian dan pembahasan yang telah dilakukan, serta saran yang dapat digunakan sebagai pengembangan sistem pada penelitian selanjutnya.

BAB II

LANDASAN TEORI

2.1 Smart Class

Smart Class merupakan konsep ruang kelas yang memanfaatkan teknologi informasi dan komunikasi untuk mendukung proses pembelajaran yang lebih efektif, efisien, dan terintegrasi. Konsep ini bertujuan untuk mengoptimalkan pengelolaan fasilitas ruang kelas serta menciptakan lingkungan belajar yang nyaman melalui pemanfaatan sistem otomatis dan digital.

Menurut beberapa penelitian, Smart Class tidak hanya berfokus pada media pembelajaran digital, tetapi juga pada pengelolaan fasilitas ruang kelas seperti pencahayaan, sirkulasi udara, keamanan, dan pemantauan kondisi lingkungan. Penerapan teknologi Internet of Things pada Smart Class memungkinkan perangkat-perangkat tersebut saling terhubung dan dapat dipantau serta dikendalikan secara real-time.

Penelitian yang dilakukan oleh (Latief et al., 2025) menyatakan bahwa penerapan Smart Class berbasis IoT mampu meningkatkan efisiensi pengelolaan ruang kelas serta memberikan kemudahan bagi pengguna dalam memantau kondisi ruang melalui sistem berbasis web. Dengan adanya sistem ini, proses pengambilan keputusan terkait pengelolaan ruang kelas dapat dilakukan secara lebih cepat dan tepat.

Dengan demikian, Smart Class dapat dipahami sebagai sistem ruang kelas cerdas yang mengintegrasikan teknologi IoT, sistem informasi, dan perangkat elektronik untuk mendukung aktivitas pembelajaran dan pengelolaan fasilitas ruang kelas secara terpusat.

2.2 Internet Of Things (IoT)

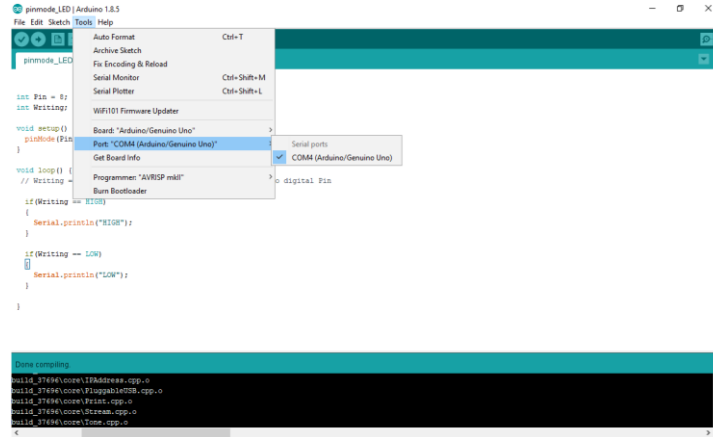
Internet of Things (IoT) merupakan konsep teknologi yang memungkinkan berbagai perangkat fisik untuk saling terhubung dan bertukar data melalui jaringan internet. Perangkat-perangkat tersebut dapat berupa sensor, aktuator, dan sistem kendali yang bekerja secara otomatis maupun terintegrasi dengan sistem informasi berbasis web.

IoT bekerja dengan menggabungkan perangkat keras, jaringan komunikasi, dan aplikasi perangkat lunak. Data yang diperoleh dari sensor akan dikirimkan ke server atau database untuk diproses dan ditampilkan kepada pengguna melalui antarmuka tertentu. Konsep ini banyak diterapkan pada sistem monitoring ruangan, smart building, dan Smart Classroom.

Penelitian oleh (Riyanto et al., 2021) menjelaskan bahwa penerapan IoT dalam sistem monitoring ruangan memungkinkan pengumpulan data secara real-time serta memberikan kemudahan dalam pengendalian perangkat dari jarak jauh. Hal ini menjadikan IoT sebagai teknologi yang tepat untuk diterapkan pada sistem Smart Class yang membutuhkan monitoring dan kontrol fasilitas ruang kelas secara terintegrasi.

Dengan penerapan IoT, sistem Smart Class dapat bekerja secara otomatis berdasarkan data sensor maupun dikendalikan secara manual melalui sistem berbasis web, sehingga meningkatkan fleksibilitas dan efisiensi pengelolaan ruang kelas.

2.3 Arduino IDE



Gambar 2. 1 Software Arduino IDE

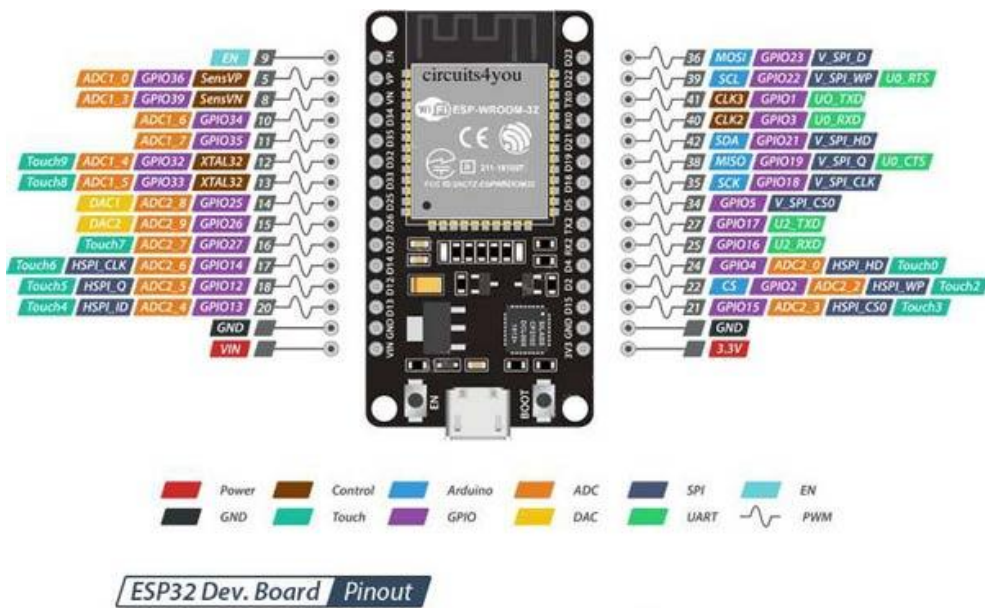
<https://www.theengineeringprojects.com/2018/10/introduction-to-arduino-ide.html>

Arduino IDE (*Integrated Development Environment*) merupakan perangkat lunak yang digunakan untuk menulis, mengompilasi, dan mengunggah program ke papan mikrokontroler berbasis Arduino maupun papan pengembangan lain yang kompatibel, termasuk ESP32 DevKit V1. Arduino IDE menyediakan antarmuka yang sederhana dan mudah digunakan, sehingga banyak dimanfaatkan dalam pengembangan sistem Internet of Things (IoT) baik untuk keperluan pembelajaran maupun penelitian.

Arduino IDE menggunakan bahasa pemrograman berbasis C/C++ yang telah disederhanakan melalui berbagai pustaka (*library*) bawaan maupun tambahan. Di dalam Arduino IDE tersedia fitur text editor untuk penulisan kode program, compiler untuk menerjemahkan kode ke dalam bahasa mesin, serta serial monitor yang digunakan untuk melihat hasil komunikasi data antara mikrokontroler dan komputer. Fitur-fitur tersebut memudahkan pengguna dalam melakukan pengujian dan pemantauan data yang dihasilkan oleh sensor secara real-time.

Arduino IDE juga mendukung pengembangan sistem berbasis ESP32 melalui penambahan board manager yang disediakan oleh Espressif Systems. Dengan dukungan tersebut, Arduino IDE dapat digunakan untuk mengelola pin input dan output, membaca data sensor, mengendalikan aktuator, serta mengatur komunikasi data melalui jaringan Wi-Fi. Kemampuan ini menjadikan Arduino IDE sebagai perangkat lunak utama dalam pengembangan prototype Smart Class berbasis IoT karena mendukung proses pemrograman, pengujian, dan integrasi sistem secara menyeluruh.

2.4 ESP32 DevKit V1



Gambar 2. 2 ESP32 Devkit V1

<https://www.ardutech.com/mengenal-esp32-development-kit-untuk-iot-internet-of-things/>

ESP32 DevKit V1 merupakan papan pengembangan mikrokontroler yang menggunakan chip ESP32 buatan *Espressif Systems*. Papan ini dirancang untuk mendukung pengembangan sistem berbasis *Internet of Things (IoT)* karena telah dilengkapi dengan modul Wi-Fi dan Bluetooth yang terintegrasi di dalam satu chip. Dengan adanya fitur tersebut, ESP32 DevKit V1 mampu melakukan komunikasi data secara nirkabel tanpa memerlukan modul tambahan.

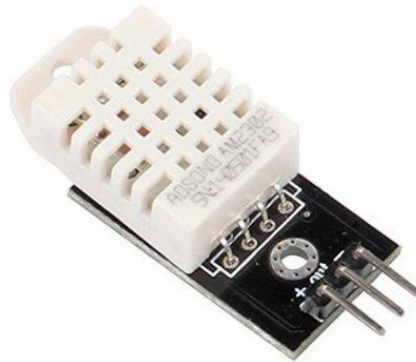
ESP32 DevKit V1 memiliki prosesor dual-core Tensilica LX6 dengan kecepatan hingga 240 MHz, sehingga mampu mengolah data dari sensor dan mengendalikan aktuator secara cepat dan stabil. Papan ini juga menyediakan sejumlah pin input dan output digital maupun analog yang dapat digunakan untuk menghubungkan berbagai perangkat eksternal seperti sensor suhu, sensor gas, relay, servo motor, dan perangkat lainnya. Selain itu, ESP32 mendukung berbagai antarmuka komunikasi, antara lain UART, SPI, I2C, PWM, dan ADC, yang memberikan fleksibilitas tinggi dalam perancangan sistem.

Untuk catu daya, ESP32 DevKit V1 dapat dioperasikan menggunakan sumber tegangan 5 V melalui koneksi USB atau tegangan 3,3 V melalui pin daya yang tersedia. Konsumsi daya yang relatif rendah menjadikan papan ini cocok digunakan pada sistem monitoring dan kontrol yang bekerja secara terus-menerus. Dengan spesifikasi dan fitur yang dimilikinya, *ESP32 DevKit V1* sangat sesuai digunakan sebagai pengendali utama dalam perancangan *prototype Smart Class* berbasis IoT yang membutuhkan fungsi monitoring, kontrol perangkat, serta integrasi dengan sistem berbasis *web (Espressif Systems, 2023)*.

Tabel 2. 1 Spesifikasi ESP32 DevKit V1

No	Spesifikasi	Keterangan
1	Chip Mikrokontroler	ESP32
2	Prosesor	Dual-Core Tensilica LX6
3	Kecepatan Clock	Hingga 240 MHz
4	Tegangan Operasi	3,3 V
5	Tegangan Input	5 V (USB)
6	Memori Flash	4 MB
7	RAM	±520 KB
8	Konektivitas	Wi-Fi 802.11 b/g/n, Bluetooth
9	Pin GPIO	±30 pin
10	Antarmuka	UART, SPI, I2C, PWM, ADC

2.5 Sensor DHT 22



Gambar 2. 3 Sensor DHT 22

<https://jagorobotik.com/blog/2025/06/28/pengenalan-modul-dht22-sensor-suhu-dan-kelembaban-andal-untuk-proyek-elektronikmu/>

Sensor DHT22 merupakan sensor digital yang digunakan untuk mengukur suhu dan kelembaban udara dalam suatu ruangan. Sensor ini dirancang untuk memberikan hasil pengukuran yang cukup akurat dan stabil, sehingga banyak digunakan pada sistem monitoring lingkungan berbasis Internet of Things. DHT22 mengirimkan data hasil pengukuran dalam bentuk sinyal digital melalui satu jalur komunikasi, sehingga memudahkan proses pembacaan data oleh mikrokontroler seperti ESP32.

Sensor DHT22 mampu melakukan pengukuran suhu dan kelembaban secara periodik dan menghasilkan data yang dapat digunakan untuk memantau kondisi lingkungan ruang kelas. Informasi suhu dan kelembaban tersebut sangat penting dalam menjaga kenyamanan ruang belajar serta mendukung pengambilan keputusan dalam pengendalian perangkat seperti kipas atau ventilasi. Sensor suhu dan kelembaban banyak diterapkan pada sistem monitoring ruangan berbasis IoT karena mampu memberikan data kondisi lingkungan secara real-time dan mudah diintegrasikan dengan sistem berbasis *web* (Ridla & Rahman, 2024).

Tabel 2. 2 Spesifikasi Sensor DHT 22

No	Spesifikasi	Keterangan
1	Jenis Sensor	Digital Temperature & Humidity
2	Tegangan Operasi	3,3 – 6 V
3	Arus Operasi	±2,5 mA
4	Rentang Suhu	-40°C sampai 80°C
5	Akurasi Suhu	±0,5°C
6	Rentang Kelembaban	0 – 100% RH
7	Akurasi Kelembaban	±2% RH
8	Resolusi	0,1°C / 0,1% RH
9	Antarmuka	Digital (Single Wire)

2.6 Sensor LDR (*Light Dependent Resistor*)



Gambar 2. 4 Light Dependent Resistor

<https://nusabot.id/blog/menggunakan-input-ldr/>

Sensor LDR (*Light Dependent Resistor*) merupakan salah satu jenis sensor cahaya yang nilai resistansinya berubah sesuai dengan intensitas cahaya yang diterimanya. Sensor ini bekerja berdasarkan prinsip fotokonduktivitas, di mana resistansi LDR akan menurun ketika intensitas cahaya meningkat dan akan meningkat ketika kondisi cahaya berkurang atau gelap.

LDR umumnya terbuat dari bahan semikonduktor seperti cadmium sulfide (CdS) yang memiliki karakteristik sensitif terhadap cahaya tampak. Ketika cahaya mengenai permukaan LDR, elektron akan berpindah sehingga hambatan listrik menjadi lebih kecil. Sebaliknya, pada kondisi gelap, hambatan LDR menjadi besar karena minimnya energi cahaya yang diterima.

Dalam sistem berbasis *Internet of Things (IoT)*, sensor LDR banyak digunakan sebagai perangkat input untuk mendeteksi kondisi pencahayaan lingkungan secara otomatis. Sensor ini sering diaplikasikan pada sistem smart lighting, monitoring intensitas cahaya ruangan, serta pengendalian lampu otomatis berdasarkan kondisi terang dan gelap.

Pada sistem *Smart Class* yang dikembangkan dalam penelitian ini, sensor LDR digunakan sebagai perangkat input untuk mendeteksi intensitas cahaya di dalam ruang kelas. Data intensitas cahaya yang diperoleh dari sensor LDR diproses oleh mikrokontroler

ESP32 DevKit V1 sebagai dasar pengendalian LED putih yang berfungsi sebagai lampu atau indikator ruang kelas.

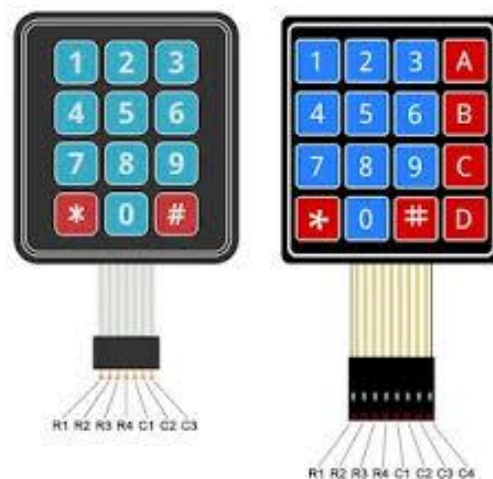
Pada kondisi terang (siang hari), sistem akan mengaktifkan LED, sedangkan pada kondisi gelap (malam hari), sistem akan mematikan LED. Penerapan sensor LDR ini bertujuan untuk mendukung otomatisasi pencahayaan ruang kelas serta meningkatkan efisiensi penggunaan energi listrik.

Penggunaan sensor LDR sebagai pendeteksi cahaya dinilai efektif karena memiliki rangkaian yang sederhana, biaya implementasi yang rendah, serta mudah diintegrasikan dengan mikrokontroler ESP32 dan sistem monitoring berbasis web (Suryadi & Fadli, 2021).

Tabel 2. 3 Spesifikasi Sensor LDR

No	Parameter	Keterangan
1	Jenis Sensor	Light Dependent Resistor (LDR)
2	Prinsip Kerja	Perubahan resistansi terhadap cahaya
3	Tegangan Operasi	3,3 – 5 V
4	Output	Analog
5	Rentang Sensitivitas	Cahaya rendah hingga tinggi
6	Antarmuka	Analog (ADC ESP32)
7	Aplikasi	Smart lighting, monitoring cahaya
8	Fungsi pada Sistem	Deteksi intensitas cahaya ruangan
9	Mode Operasi	Otomatis & Manual
10	Integrasi	ESP32 DevKit V1

2.7 Keypad 4x4



Gambar 2. 5 Keypad 4x4

<https://www.aksesoriskomputerlampung.com/2019/07/keypad-membran-4x4-arduino.html>

Keypad 4×4 merupakan perangkat input yang terdiri dari 16 tombol yang disusun dalam matriks 4 baris dan 4 kolom. Setiap tombol mewakili kombinasi tertentu antara baris dan kolom, sehingga memungkinkan sistem membaca banyak input dengan jumlah pin input/output yang lebih efisien. Keypad jenis ini umum digunakan pada sistem tertanam karena sederhana, mudah diimplementasikan, serta memiliki tingkat keandalan yang baik sebagai media masukan data.

Prinsip kerja keypad 4×4 dilakukan dengan metode pemindaian (scanning) baris dan kolom oleh mikrokontroler. Ketika sebuah tombol ditekan, akan terbentuk koneksi antara satu baris dan satu kolom tertentu yang kemudian dikenali sebagai input oleh sistem. Metode ini memungkinkan identifikasi tombol secara akurat tanpa memerlukan rangkaian tambahan yang kompleks.

Dalam sistem Smart Class yang dirancang, keypad 4×4 digunakan sebagai sistem autentikasi untuk pengamanan akses pintu ruang kelas. Pengguna diwajibkan memasukkan kode akses tertentu sebelum sistem memberikan perintah kepada aktuator untuk membuka pintu. Penerapan keypad sebagai kontrol akses bertujuan untuk meningkatkan keamanan ruang kelas dan membatasi akses hanya kepada pengguna yang berwenang.

Penggunaan keypad sebagai sistem pengaman akses telah banyak diterapkan pada sistem keamanan ruangan berbasis mikrokontroler dan Internet of Things. Keypad dinilai efektif sebagai media autentikasi karena mudah diintegrasikan dengan sistem kontrol serta dapat dikombinasikan dengan pengelolaan akses berbasis web (Fauziman & Mukhaiyar, 2023).

Tabel 2. 4 Spesifikasi Keypad

No	Parameter	Keterangan
1	Jenis Perangkat	Keypad matriks
2	Jumlah Tombol	16 tombol
3	Konfigurasi	4 baris × 4 kolom
4	Tegangan Operasi	3,3 – 5 V
5	Metode Pembacaan	Scanning baris dan kolom
6	Antarmuka	Digital
7	Jumlah Pin	8 pin (4 baris, 4 kolom)
8	Material Tombol	Membran / tombol tekan
9	Fungsi Utama	Input kode atau perintah
10	Aplikasi Umum	Sistem keamanan dan kontrol akses

2.8 Servo



Gambar 2. 6 Servo

<https://www.edukasielatronika.com/2020/12/motor-servo-sg90.html>

Motor servo merupakan aktuator yang digunakan untuk menghasilkan gerakan mekanik dengan sudut tertentu secara presisi. Motor servo banyak digunakan pada sistem otomasi dan Internet of Things karena mampu mengontrol posisi secara akurat berdasarkan sinyal kendali dari mikrokontroler. Penerapan motor servo pada sistem otomasi pintu dan mekanisme gerak telah banyak digunakan pada sistem berbasis mikrokontroler dan IoT (Samsugi et al., 2020).

Motor servo pada sistem ini dikendalikan menggunakan sinyal Pulse Width Modulation (PWM) dari ESP32 untuk menentukan sudut putaran poros. Dengan kemampuan pengaturan sudut yang presisi, motor servo digunakan untuk membuka dan menutup pintu atau ventilasi secara otomatis maupun manual melalui sistem berbasis web.

Tabel 2. 5 Spesifikasi Servo

Parameter	Keterangan
Tegangan Kerja	4,8 – 6 V
Sudut Putar	0° – 180°
Sinyal Kendali	PWM
Tipe Gerak	Presisi sudut
Aplikasi	Otomasi pintu & mekanisme

2.9 Fan



Gambar 2. 7 Fan

<https://h5.lazada.co.id/products/kipas-dc-12v-4x4-cm-cooling-fan-12v-dc-kipas-mini-pendingin-4-x-4cm-i829118165.html>

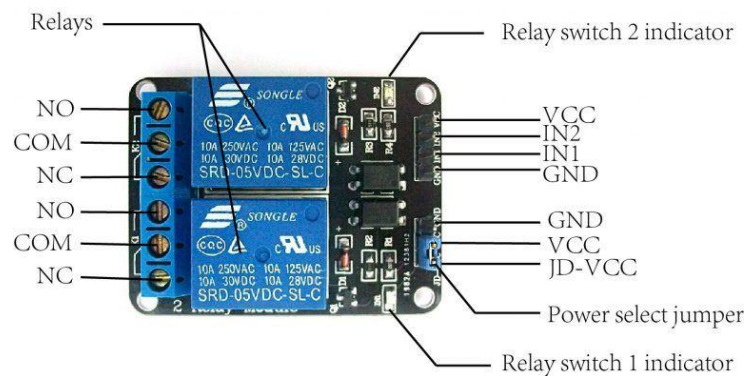
Fan atau kipas angin merupakan perangkat output yang digunakan untuk membantu sirkulasi udara dan menjaga kenyamanan suhu ruangan. Penggunaan kipas angin pada sistem monitoring lingkungan berbasis IoT umum diterapkan untuk meningkatkan kualitas udara dan kenyamanan ruang (Budiyanto et al., 2020).

Pada sistem Smart Class, fan dikendalikan melalui relay berdasarkan data suhu dan kualitas udara dari sensor. Fan dapat bekerja secara otomatis maupun manual melalui antarmuka web untuk menjaga kondisi ruang kelas tetap nyaman.

Tabel 2. 6 Spesifikasi Fan

Parameter	Keterangan
Tegangan Kerja	12 DC
Kendali	Relay
Fungsi	Sirkulasi udara
Mode Operasi	Otomatis & Manual
Aplikasi	Ruang kelas

2.10 Relay Optocoupler 2 channel



Gambar 2. 8 Relay Optocoupler 2 Channel

<https://1ohm.in/product/2-channel-5v-relay-module-for-arduino-with-optocoupler-2/>

Relay optocoupler 2 channel merupakan modul saklar elektronik yang dilengkapi dua kanal dan isolasi optocoupler untuk memisahkan rangkaian kontrol dan rangkaian daya. Relay jenis ini banyak digunakan pada sistem otomasi berbasis IoT untuk meningkatkan keamanan pengendalian perangkat listrik (N Somayasa et al., 2024).

Relay ini memungkinkan mikrokontroler mengendalikan dua perangkat listrik secara terpisah dengan tegangan kontrol rendah. Isolasi optocoupler berfungsi melindungi ESP32 dari gangguan listrik yang berasal dari beban.

Tabel 2. 7 Spesifikasi Relay Optocoupler 2 Channel

Parameter	Keterangan
Jumlah Kanal	2 Channel
Tegangan Kontrol	3,3 – 5 V
Tegangan Beban	Hingga 250 VAC
Arus Maksimum	10 A
Isolasi	Optocoupler

2.11 LED



Gambar 2. 9 LED

https://soldered.com/learn/led-light-emitting-diode-explained/?srsltid=AfmBOorb_nOXm67IhLiKW8gw3m1nPeYfhdobGNMhYfBUoCP_E5ccVc0

Light Emitting Diode (LED) merupakan perangkat output yang berfungsi sebagai indikator visual pada suatu sistem. LED banyak digunakan pada sistem tertanam dan IoT karena konsumsi daya yang rendah dan respon yang cepat (Kalbuana & Kurnianto, 2024).

Pada sistem Smart Class, LED digunakan sebagai indikator status perangkat yang dapat dikendalikan melalui sistem berbasis web. LED memberikan informasi visual terkait kondisi aktif atau nonaktif suatu perangkat.

Tabel 2. 8 Spesifikasi LED

Parameter	Keterangan
Jumlah Kanal	2 Channel
Tegangan Kontrol	3,3 – 5 V
Tegangan Beban	Hingga 250 VAC
Arus Maksimum	10 A
Isolasi	Optocoupler

2.12 Liquid Crystal Display I2C 20x4



Gambar 2. 10 Liquid Crystal Display I2C

https://www.sunfounder.com/products/i2c-lcd2004-module?srsId=AfmBOoqs1_eZLHG9p0SxR9ZcwaCK3uuHkYC0fQ3thK9cuqLeqIJpExE

Liquid Crystal Display (LCD) I2C merupakan perangkat output yang digunakan untuk menampilkan informasi dalam bentuk teks dengan antarmuka komunikasi I2C. LCD I2C banyak digunakan pada sistem IoT karena efisiensi penggunaan pin dan kemudahan integrasi (Susilania, 2025).

LCD I2C digunakan untuk menampilkan informasi kondisi ruang kelas secara langsung tanpa perlu mengakses web, seperti suhu ruangan, kualitas udara, dan status perangkat.

Tabel 2. 9 Spesifikasi Liquid Crystal Display I2C

Parameter	Keterangan
Ukuran	20 × 4
Antarmuka	I2C
Tegangan Kerja	5 V
Fungsi	Tampilan informasi
Jumlah Pin	4 pin

2.13 Web Hosting



Gambar 2. 11 Web Hosting

<https://www.codepolitan.com/blog/ingin-bikin-website-sendiri-ini-dia-rahasia-dibaliknya-ipk88n/>

Web hosting merupakan layanan yang menyediakan sumber daya server untuk menyimpan, mengelola, dan menjalankan aplikasi berbasis web agar dapat diakses melalui jaringan internet. Web hosting bekerja bersama dengan domain, di mana domain berfungsi sebagai alamat unik yang digunakan pengguna untuk mengakses sistem. Dalam konteks sistem berbasis Internet of Things, web hosting berperan sebagai pusat pengelolaan data dan antarmuka pengguna yang menghubungkan perangkat IoT dengan pengguna akhir melalui jaringan internet. Penggunaan web hosting dan domain memungkinkan sistem monitoring dan kontrol dapat diakses secara jarak jauh tanpa keterbatasan lokasi (Ontowirjo Y.Q Fauzah & Poekoel C. Vecky, 2018)

Pada sistem Smart Class yang dirancang, web hosting digunakan untuk menampung aplikasi web yang berfungsi menampilkan data hasil pembacaan sensor, menyimpan data ke dalam database, serta menyediakan fitur kontrol perangkat. Melalui domain yang terhubung dengan web hosting, pengguna dapat mengakses sistem untuk memantau kondisi ruang kelas secara real-time, mengatur mode pengoperasian sistem, serta mengendalikan perangkat output seperti lampu, kipas, dan pintu. Implementasi web hosting sebagai media monitoring dan kontrol pada sistem IoT menjadikan sistem lebih terpusat, mudah diakses, dan fleksibel untuk dikembangkan sesuai kebutuhan di masa mendatang.

2.14 Studi Literatur

Melakukan studi literatur merupakan proses mengumpulkan dan mempelajari referensi ilmiah yang relevan dengan topik penelitian. Studi literatur dilakukan untuk memperoleh pemahaman mendalam mengenai konsep, teori, maupun penelitian terdahulu yang berkaitan dengan sistem *Smart Class* berbasis *Internet of Things*. Referensi yang digunakan pada penelitian ini merupakan jurnal ilmiah terbit dalam lima tahun terakhir dan memiliki keterkaitan dengan perancangan sistem monitoring, kontrol perangkat, sensor, serta aplikasi web berbasis IoT. Adapun tabel studi literatur yang digunakan adalah sebagai berikut:

Tabel 2. 10 Studi Literatur

No	Kolom	Isi
1	Judul	Penerapan Teknologi Internet of Things (IoT) untuk Monitoring Kualitas Udara dalam Ruangan
	Penulis dan tahun terbit	(Aswaldi, 2025)
	Penerbit / Jurnal	Journal of Computer Science and Information Technology
	Deskripsi	Penelitian merancang sistem IoT untuk memonitor kualitas udara dalam ruangan secara real-time, termasuk suhu, kelembapan, CO ₂ , dan partikel debu (PM2.5) yang dikirim dan ditampilkan melalui platform IoT.
2	Judul	Desain dan Implementasi Smart Classroom Berbasis ESP32 dengan Monitoring Berbasis Web
	Penulis dan tahun terbit	(Sholeha et al., 2025)
	Penerbit / Jurnal	Impression: Jurnal Teknologi dan Informasi
	Deskripsi	Penelitian ini merancang dan mengimplementasikan media pembelajaran dan monitoring IoT menggunakan mikrokontroler ESP32 sebagai basis sistem yang terhubung ke internet. Sistem ini menggunakan ESP32 untuk mengambil data sensor, mengirimkannya secara real-time ke server web, dan mengendalikan fungsi perangkat melalui tampilan web sehingga pengguna (mis. guru atau admin) dapat memantau serta mengatur perangkat jarak jauh lewat web browser. Hasil uji validasi menunjukkan sistem ini layak dan efektif dalam pembelajaran interaktif serta monitoring berbasis web, memperlihatkan bahwa ESP32 DevKit V1 mampu mengelola sensor dan antarmuka web secara real-time
3	Judul	Peningkatan Akurasi Sensor Suhu dan Kelembaban DHT11 dengan Kalibrasi Suhu Berbasis IoT pada Platform Thingspeak
	Penulis dan tahun terbit	(Tiyas et al., 2025)
	Penerbit / Jurnal	Jurnal Pendidikan dan Teknologi Indonesia

	Deskripsi	Fokus pada peningkatan akurasi pembacaan sensor suhu dan kelembapan DHT11 pada sistem IoT berbasis platform Thingspeak, meningkatkan keandalan pengukuran lingkungan.
4	Judul	Sistem Smart Class Berbasis Internet Of Things dengan Metode Prototype
	Penulis dan tahun terbit	(Pramudita & Setyawan, 2022)
	Penerbit / Jurnal	SMARTICS Journal
	Deskripsi	Penelitian ini membahas perancangan prototype Smart Class berbasis IoT yang memungkinkan monitoring dan kontrol perangkat elektronik pada ruang kelas (lampu, AC, proyektor) secara otomatis menggunakan NodeMCU dan aplikasi Blynk sebagai antarmuka web/mobile.
5	Judul	Pengumpulan Data Suhu dan Kelembapan Ruangan Menggunakan Embedded System Berbasis IoT
	Penulis dan tahun terbit	(Made et al., 2023)
	Penerbit / Jurnal	INSANtek
	Deskripsi	Sistem IoT yang dibuat untuk mengumpulkan data suhu & kelembapan ruangan secara otomatis dan konsisten, mendukung pengambilan keputusan pada sistem pendingin ruangan.
6	Judul	Rancang Bangun Sistem Monitoring Lingkungan dengan Sistem IoT dan Logika Fuzzy
	Penulis dan tahun terbit	(Yunardi et al., 2023)
	Penerbit / Jurnal	Cyberspace: Jurnal Pendidikan Teknologi Informasi
	Deskripsi	Merancang sistem web monitoring kualitas udara dalam ruangan dengan metode logika fuzzy untuk interpretasi data suhu, kelembapan, dan VOC real-time dari sensor.
7	Judul	Sistem Smart Class Berbasis Internet of Things (IoT)
	Penulis dan tahun terbit	(Sari et al., 2025)
	Penerbit / Jurnal	Sudo Jurnal Teknik Informatika

	Deskripsi	Penelitian ini merancang dan mengimplementasikan sistem <i>Smart Class</i> berbasis IoT untuk meningkatkan efisiensi absensi dan keamanan di ruang kelas melalui teknologi IoT yang terhubung dengan platform Blynk.cloud dan Telegram untuk monitoring real-time.
8	Judul	Rancang Bangun Sistem Monitoring Kualitas Udara di Laboratorium / Ruang Komputer
	Penulis dan tahun terbit	(Putri et al., 2025)
	Penerbit / Jurnal	Jurnal Informatika dan Teknik Elektro Terapan
	Deskripsi	Sistem IoT untuk monitoring suhu, kelembapan, dan kualitas udara di ruang lab komputer dengan NodeMCU, data dikirim ke aplikasi Blynk & Telegram untuk notifikasi kondisi abnormal.
9	Judul	Sistem Monitoring Suhu Ruangan Server Berbasis IoT (HTTP)
	Penulis dan tahun terbit	(Nur & Sari, 2021)
	Penerbit / Jurnal	Infosecure
	Deskripsi	Monitoring suhu & kelembapan ruang server menggunakan IoT dengan protokol HTTP, menampilkan data real-time di web & notifikasi otomatis.
10	Judul	Sistem Monitoring Kualitas Udara dan Pengaturan Suhu Berbasis IoT
	Penulis dan tahun terbit	(Irdyanti et al., 2025)
	Penerbit / Jurnal	Jurnal Ampere
	Deskripsi	Meneliti sistem monitoring kualitas udara berbasis IoT yang mampu mendeteksi suhu, kelembapan, CO, serta rekomendasi pengaturan otomatis untuk mempertahankan kondisi ideal.

BAB III

METODOLOGI PENELITIAN

Dalam merancang prototype pada tugas akhir ini, diperlukan metode penelitian yang terstruktur agar proses pengembangan sistem *Smart Class* berbasis *Internet of Things* dapat dilakukan secara sistematis dan sesuai dengan tujuan penelitian. Perancangan dilakukan melalui beberapa tahapan, mulai dari pengumpulan data, analisis kebutuhan komponen, perancangan rangkaian perangkat keras, perancangan perangkat lunak pada mikrokontroler, hingga integrasi dengan aplikasi web sebagai media monitoring dan kontrol. Metodologi ini disusun berdasarkan teori yang telah dibahas pada Bab II dan disesuaikan dengan fungsi setiap komponen yang digunakan, sehingga prototype yang dihasilkan mampu bekerja secara optimal, terintegrasi, dan mendukung implementasi sistem *Smart Class* yang efektif.

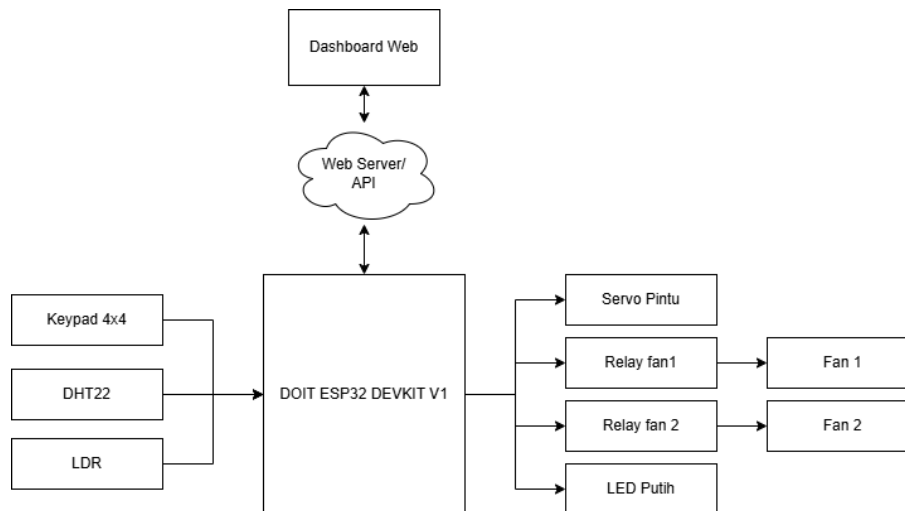
3.1 Data Penelitian

Data penelitian pada tugas akhir ini diperoleh melalui proses pengumpulan dan pengujian data menggunakan perangkat input yang terintegrasi pada sistem *Smart Class* berbasis *Internet of Things (IoT)*. Data utama yang digunakan meliputi kondisi lingkungan ruang kelas berupa suhu, kelembapan udara, serta intensitas cahaya ruangan. Data suhu dan kelembapan diperoleh dari sensor *DHT22* yang digunakan untuk memantau kondisi lingkungan ruang kelas secara berkala, sedangkan data intensitas cahaya diperoleh dari sensor *Light Dependent Resistor (LDR)*.

Selain data lingkungan, data terkait autentikasi akses pintu diperoleh melalui *keypad* 4×4 sebagai perangkat input yang digunakan oleh pengguna. Seluruh data yang diperoleh dari sensor dan perangkat input tersebut diolah oleh mikrokontroler *ESP32 DevKit V1* sebagai dasar dalam pengambilan keputusan sistem, baik pada *mode otomatis* maupun *mode manual*.

Data hasil pembacaan sensor serta status perangkat kemudian dikirimkan ke server melalui koneksi *WiFi* untuk disimpan ke dalam *database* dan ditampilkan pada antarmuka berbasis web secara *real-time*. Dengan demikian, data penelitian yang digunakan tidak hanya mendukung proses monitoring kondisi lingkungan ruang kelas, tetapi juga berperan penting dalam pengembangan, pengujian, dan evaluasi fungsionalitas sistem *Smart Class* yang dirancang pada tugas akhir ini.

3.2 Rancangan Diagram Blok Rangkaian



Gambar 3. 1 Diagram Blok Alat

Gambar 3.1 merupakan diagram blok dari sistem *Smart Class* berbasis *Internet of Things (IoT)* yang dirancang pada tugas akhir ini. Diagram ini menunjukkan alur kerja sistem secara keseluruhan, mulai dari perangkat input, mikrokontroler sebagai pusat kendali, hingga perangkat output serta integrasi dengan *web server*.

Pada bagian kiri diagram terdapat perangkat input yang digunakan dalam sistem, yaitu *keypad* 4×4 sebagai alat autentikasi akses pintu, sensor *DHT22* untuk membaca suhu dan kelembapan ruang kelas, serta sensor *Light Dependent Resistor (LDR)* untuk mendeteksi intensitas cahaya ruangan. Data yang diperoleh dari perangkat input tersebut dikirimkan ke mikrokontroler untuk diproses sesuai dengan logika sistem yang telah dirancang.

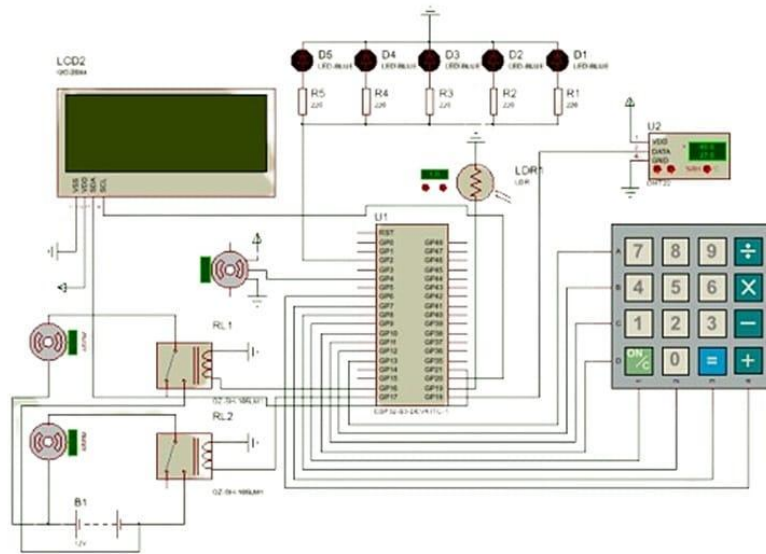
Pada bagian tengah diagram terdapat modul *DOIT ESP32 DevKit V1* yang berperan sebagai pusat pengolahan data dan pengendalian sistem. Mikrokontroler *ESP32* menerima data dari sensor dan *keypad*, menjalankan logika sistem baik pada *mode otomatis* maupun *mode manual*, serta mengendalikan perangkat output sesuai dengan kondisi yang ditentukan. Selain itu, *ESP32* juga berkomunikasi dengan *Web Server* atau *Application Programming Interface (API)* melalui koneksi *WiFi*. *Web server* berfungsi sebagai penghubung antara *ESP32* dan *dashboard web*, sehingga data hasil pembacaan sensor dapat dikirim ke server dan ditampilkan secara *real-time*, serta perintah dari pengguna dapat dikirim kembali ke *ESP32*.

Pada bagian kanan diagram terdapat perangkat output yang dikendalikan oleh *ESP32*, yaitu motor *servo* yang digunakan untuk mekanisme buka dan tutup pintu ruang kelas, modul *relay* yang berfungsi untuk mengendalikan kipas (*Fan 1* dan *Fan 2*), serta *LED* putih yang digunakan sebagai lampu atau indikator ruang kelas. Selain itu, sistem juga menggunakan modul *buck converter (LM2596)* yang berfungsi menurunkan tegangan dari *Power Supply Unit (PSU)* agar sesuai dengan kebutuhan tegangan *ESP32* dan perangkat lainnya.

Secara keseluruhan, diagram blok ini menggambarkan integrasi antara perangkat input, proses pengolahan data oleh mikrokontroler, komunikasi berbasis *web service*, serta perangkat output dalam sistem *Smart Class* yang dirancang.

3.3 Rancangan Gambar Rangkaian Keseluruhan

3.3.1 Rangkaian Keseluruhan



Gambar 3. 2 Rangkaian Keseluruhan Alat

Gambar 3.2 merupakan rangkaian keseluruhan sistem *Smart Class* berbasis *Internet of Things (IoT)* yang dirancang pada tugas akhir ini. Pada rangkaian tersebut, seluruh perangkat input dan aktuator dihubungkan ke mikrokontroler *DOIT ESP32 DevKit V1* yang berfungsi sebagai pusat kendali utama sistem.

Perangkat input pada sistem ini terdiri dari *keypad* 4×4 yang digunakan untuk autentikasi akses pintu, sensor *DHT22* yang berfungsi untuk membaca suhu dan kelembapan ruang kelas, serta sensor *Light Dependent Resistor (LDR)* yang digunakan untuk mendeteksi intensitas cahaya ruangan. Data yang diperoleh dari seluruh perangkat input tersebut dikirimkan ke *ESP32* untuk diproses sesuai dengan logika sistem yang telah dirancang.

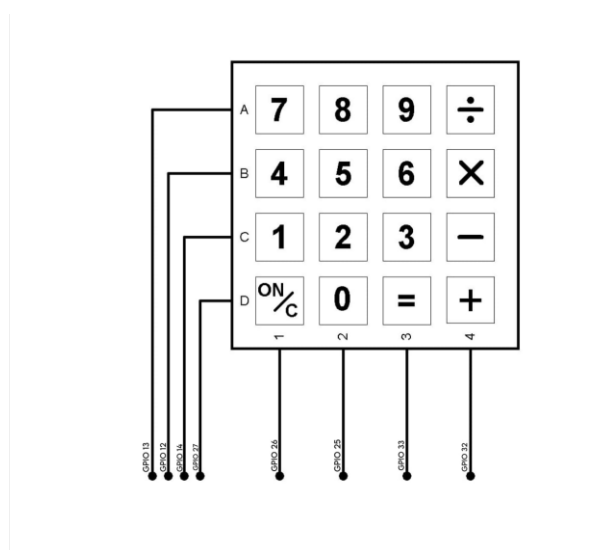
Perangkat output pada rangkaian ini meliputi motor *servo* yang digunakan sebagai mekanisme buka dan tutup pintu ruang kelas, dua unit kipas (*Fan 1* dan *Fan 2*) yang dikendalikan melalui modul *relay 2-channel*, serta *LED* putih yang berfungsi sebagai lampu atau indikator ruang kelas. *LED* dikendalikan oleh sistem berdasarkan nilai intensitas cahaya yang diperoleh dari sensor *LDR* pada mode otomatis. Selain itu, sistem juga dilengkapi dengan *Liquid Crystal Display (LCD) I2C 20×4* yang digunakan untuk menampilkan informasi kondisi ruang kelas, seperti suhu, kelembapan, intensitas cahaya, dan status sistem secara langsung.

Untuk catu daya, sistem menggunakan sumber tegangan 9 V yang dialirkan ke modul *relay* dan kipas, sedangkan mikrokontroler *ESP32* mendapatkan suplai daya melalui jalur 5 V yang telah distabilkan menggunakan modul *buck converter (LM2596)*. Seluruh rangkaian dirancang agar setiap komponen dapat bekerja secara terintegrasi, sehingga mendukung proses monitoring dan pengendalian fasilitas ruang kelas melalui *dashboard web* secara jarak jauh.

Tabel 3. 1 Rangkaian Keseluruhan Alat

No	Komponen	Pin Komponen	Pin ESP32 DevKit V1
1	Keypad 4×4	R1	GPIO 13
		R2	GPIO 12
		R3	GPIO 14
		R4	GPIO 27
		C1	GPIO 26
		C2	GPIO 25
		C3	GPIO 33
		C4	GPIO 32
2	DHT22	DATA	GPIO 18
3	LDR	A/O	GPIO 15
4	Servo Pintu	Signal	GPIO 4
5	Relay Fan 1	IN	GPIO 16
6	Relay Fan 2	IN	GPIO 17
7	LED	Anoda	GPIO 3
9	LCD I2C 20×4	SDA	GPIO 21
		SCL	GPIO 22
		VCC	5V
		GND	GND
10	Power Supply	PSU 9V → Relay/Kipas	—
		Buck Converter → 5V	5V ESP32
		GND	Semua komponen

3.3.2 Rangkaian keypad 4x4



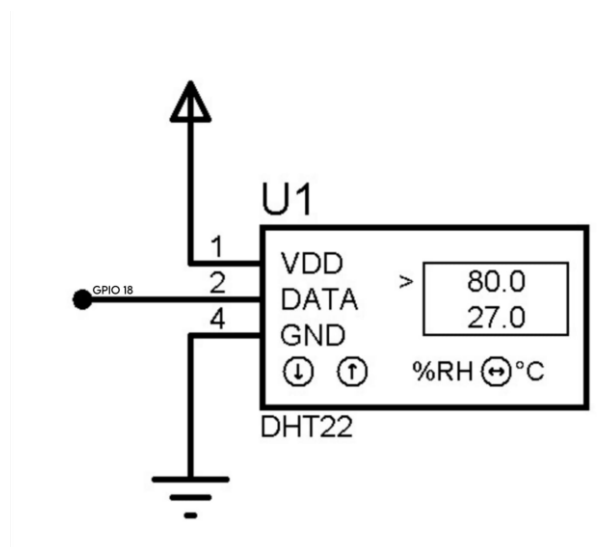
Gambar 3. 3 Rangkaian Keypad 4x4

Pada Gambar 3.3 menunjukkan rangkaian keypad 4×4 yang terhubung ke ESP32 DevKit V1. Pada rangkaian ini, empat pin baris keypad (R1–R4) dihubungkan ke pin GPIO 13, 12, 14, dan 27, sedangkan empat pin kolom (C1–C4) dihubungkan ke GPIO 26, 25, 33, dan 32. Konfigurasi ini memungkinkan ESP32 membaca kombinasi tombol yang ditekan pengguna dan digunakan sebagai input PIN untuk sistem akses pintu.

Tabel 3. 2 Rangkaian Keypad 4x4

No	Pin Keypad	Pin ESP32
1	R1	GPIO 13
2	R2	GPIO 12
3	R3	GPIO 14
4	R4	GPIO 27
5	C1	GPIO 26
6	C2	GPIO 25
7	C3	GPIO 33
8	C4	GPIO 32

3.3.3 Rangkaian Sensor DHT 22



Gambar 3. 4 Rangkaian Sensor DHT 22

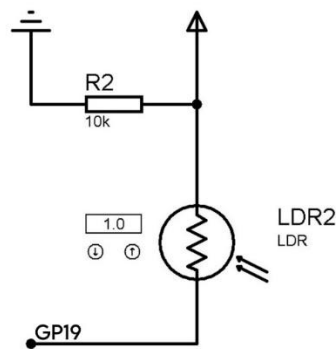
Pada Gambar 3.4 menunjukkan rangkaian sensor DHT22 yang terhubung ke ESP32. Sensor ini menggunakan pin data yang terhubung ke GPIO 18. DHT22 menerima suplai daya dari pin 5V dan GND. Dengan konfigurasi ini sensor dapat mengirimkan data suhu dan kelembapan ke ESP32 secara digital.

Tabel 3. 3 Rangkaian Sensor DHT 22

No	Pin DHT22	Pin ESP32
1	VCC	5V

2	DATA	GPIO 18
3	GND	GND

3.3.4 Rangkaian LDR



Gambar 3. 5 Rangkaian LDR

Pada Gambar 3.5 menunjukkan rangkaian sensor Light Dependent Resistor (LDR) yang terhubung ke mikrokontroler ESP32 DevKit V1. Sensor LDR digunakan untuk mendeteksi intensitas cahaya di dalam ruang kelas sebagai dasar pengendalian LED putih pada sistem Smart Class.

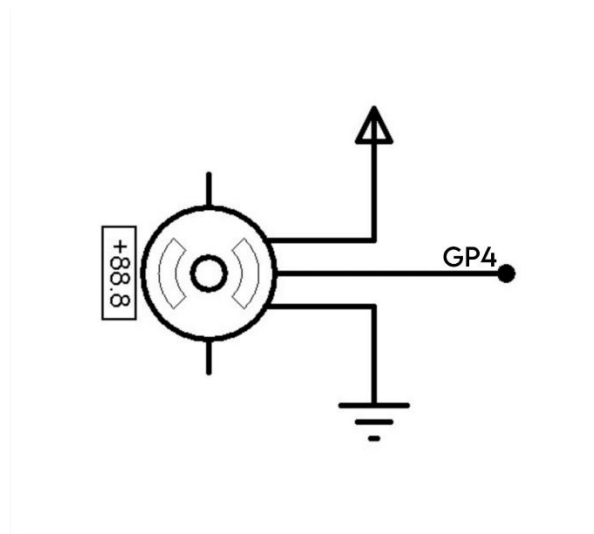
Rangkaian sensor LDR disusun menggunakan konfigurasi pembagi tegangan (voltage divider) sehingga menghasilkan sinyal analog yang berubah sesuai dengan intensitas cahaya lingkungan. Output rangkaian LDR dihubungkan ke salah satu pin Analog to Digital Converter (ADC) pada ESP32, yaitu GPIO 15. Sensor LDR mendapatkan suplai tegangan dari pin 3,3 V dan GND pada ESP32.

Nilai tegangan analog yang dihasilkan oleh sensor LDR akan dibaca oleh ESP32 dan digunakan sebagai parameter pengendalian LED putih secara otomatis. Pada kondisi cahaya terang (siang hari), sistem akan mengaktifkan LED, sedangkan pada kondisi cahaya rendah atau gelap (malam hari), LED akan dimatikan sesuai dengan logika sistem yang telah dirancang.

Tabel 3. 4 Rangkaian Sensor LDR

No	Pin LDR	Pin ESP32
1	VCC	3.3 V
2	Output	GPIO 15
3	GND	GND

3.3.5 Rangkaian Servo



Gambar 3. 6 Rangkaian Servo

Gambar 3.6 merupakan rangkaian motor servo yang digunakan pada sistem Smart Class sebagai aktuator untuk mekanisme buka dan tutup pintu ruang kelas. Motor servo ini terhubung langsung ke mikrokontroler ESP32 DevKit V1 dan mendapatkan suplai daya dari jalur tegangan 5 V.

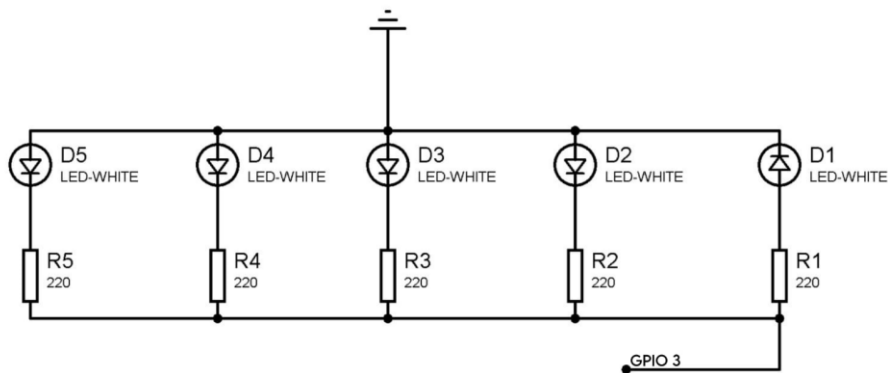
Pada rangkaian tersebut, pin sinyal motor servo dihubungkan ke salah satu pin General Purpose Input Output (GPIO) pada ESP32, yaitu GPIO 4. Motor servo dikendalikan menggunakan sinyal Pulse Width Modulation (PWM) yang dihasilkan oleh ESP32 untuk mengatur sudut putaran sesuai dengan perintah sistem.

Motor servo berfungsi untuk membuka dan menutup akses pintu ruang kelas secara otomatis berdasarkan input dari keypad maupun perintah yang diberikan melalui dashboard web. Dengan konfigurasi ini, motor servo dapat bekerja secara presisi dan responsif sesuai dengan logika pengendalian yang diterapkan pada sistem Smart Class.

Tabel 3. 5 Rangkaian Servo

No	Komponen	Pin Servo	Pin ESP32
1	Servo Pintu	Signal	GPIO 4
2	VCC	5V	5V
3	GND	GND	GND

3.3.6 Rangkaian LED



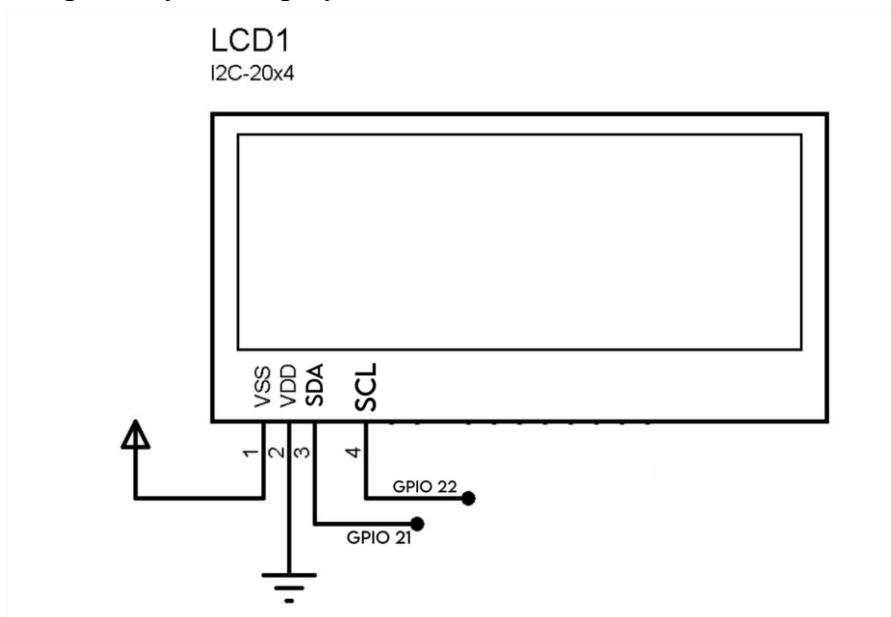
Gambar 3. 7 Rangkaian LED

Gambar 3. 7 merupakan rangkaian LED yang digunakan sebagai indikator pada sistem Smart Class. LED berfungsi sebagai lampu ruangan yang dapat menyala berdasarkan intensitas cahaya atau perintah manual dari web. Pada rangkaian ini, LED terhubung ke pin GPIO 3 pada ESP32. LED dihubungkan ke ground melalui resistor untuk menjaga kestabilan arus.

Tabel 3. 6 Rangkaian LED

No	Komponen	Pin LED	Pin ESP32
1	LED	Anoda	GPIO 3
		Katoda	GND

3.3.7 Rangkaian Liquid Crystal Display 20x4 I2C



Gambar 3. 8 Rangkaian Liquid Crystal Display 20x4 I2C

Pada Gambar 3.8 menunjukkan rangkaian LCD I2C 20×4 yang terhubung ke ESP32 melalui jalur komunikasi I2C. Pin SDA LCD dihubungkan ke GPIO 21 dan pin SCL dihubungkan ke GPIO 22. LCD mendapat suplai daya 5V dan ground. Dengan konfigurasi ini, LCD dapat menampilkan informasi suhu, kondisi cuaca, status kipas, serta status pintu secara real-time.

Tabel 3. 7 Rangkaian Liquid Crystal Display 20x4 I2C

No	Pin LCD	Pin ESP32	Fungsi
1	SDA	GPIO 21	Transfer data I2C
2	SCL	GPIO 22	Clock I2C
3	VCC	5V	Suplai daya LCD
4	GND	GND	Ground

3.4 Rancangan Metodologi Pengembangan *Software*

Metodologi pengembangan software yang digunakan pada penelitian ini bertujuan untuk menggambarkan alur kerja sistem Smart Class secara menyeluruh, mulai dari proses penerimaan input hingga menghasilkan output berupa pengendalian perangkat dan informasi monitoring. Metodologi ini dirancang agar sistem dapat berjalan secara terstruktur, mudah dipahami, serta mampu mendukung proses otomatisasi dan kontrol manual yang diterapkan pada sistem.

Pengembangan perangkat lunak dilakukan dengan pendekatan alur proses (*process flow-based*), di mana setiap tahapan saling terhubung dan berjalan secara berulang (*looping*) untuk memastikan sistem dapat bekerja secara realtime dan berkelanjutan.

3.4.1 Langkah Alur Metodologi

Secara umum, alur metodologi pengembangan *software* pada sistem *Smart Class* dapat dijelaskan sebagai berikut:

1. Inisialisasi Sistem

Proses dimulai ketika sistem pertama kali diaktifkan. Mikrokontroler *ESP32 DevKit V1* melakukan inisialisasi terhadap seluruh komponen perangkat keras yang digunakan, meliputi sensor *DHT22*, sensor *Light Dependent Resistor (LDR)*, keypad 4×4, motor *servo*, modul *relay*, *LED*, serta *Liquid Crystal Display (LCD)*. Selain itu, sistem juga melakukan konfigurasi dan koneksi jaringan *WiFi* agar dapat terhubung dengan server.

2. Penerimaan Input

Input pada sistem diperoleh dari dua sumber utama, yaitu:

1. **Input sensor**, berupa data suhu dan kelembapan ruang kelas yang diperoleh dari sensor *DHT22* serta data intensitas cahaya yang diperoleh dari sensor *LDR*.
2. **Input pengguna**, yang berasal dari *dashboard web* dalam bentuk pemilihan mode operasi (*auto* atau *manual*) serta perintah pengendalian perangkat.

3. Pengambilan Data dari Server
Setelah terhubung ke jaringan, *ESP32* secara berkala melakukan permintaan data ke server menggunakan metode *HTTP GET*. Data yang diambil meliputi mode sistem dan status aktuator pada mode manual, seperti pengaturan kipas, lampu (*LED*), dan pintu.
4. Proses pengolahan data
Pada tahap ini, sistem melakukan pengolahan data berdasarkan mode operasi yang aktif:
 - a. Mode otomatis, sistem mengambil keputusan berdasarkan nilai sensor yang diperoleh.
 - b. Mode manual, sistem menonaktifkan logika sensor dan hanya menjalankan perintah yang diterima dari server.
5. Eksekusi Output
Hasil pengolahan data digunakan untuk mengendalikan perangkat output, yaitu mengatur posisi motor *servo* sebagai mekanisme buka dan tutup pintu ruang kelas serta menghidupkan atau mematikan kipas melalui modul *relay* sesuai dengan kondisi sistem.
6. Pengiriman data ke server
Setelah sistem dijalankan, *ESP32* mengirimkan data sensor dan status perangkat ke server menggunakan metode *HTTP POST*. Data ini disimpan ke dalam database dan ditampilkan secara realtime pada dashboard web.
7. Monitoring dan Evaluasi
Seluruh proses yang berjalan ditampilkan pada Serial Monitor sebagai media monitoring dan evaluasi sistem. Informasi yang ditampilkan meliputi mode sistem, nilai sensor, serta status aktuator untuk memastikan sistem bekerja sesuai dengan perancangan.

Seluruh proses yang berjalan ditampilkan pada Serial Monitor sebagai media monitoring dan evaluasi sistem. Informasi yang ditampilkan meliputi mode sistem, nilai sensor, serta status aktuator untuk memastikan sistem bekerja sesuai dengan perancangan.

3.4.2 Formulasi Matematika

Formulasi matematika pada sistem Smart Class digunakan sebagai dasar dalam proses pengolahan data sensor dan pengambilan keputusan sistem, khususnya pada mode otomatis. Perhitungan yang diterapkan bersifat sederhana, namun cukup untuk mendukung proses monitoring dan pengendalian perangkat secara akurat dan realtime.

1. Formulasi Pengukuran Suhu dan Kelembapan (Sensor DHT22)

Sensor DHT22 menghasilkan data suhu dan kelembapan dalam bentuk nilai digital yang langsung dibaca oleh mikrokontroler ESP32. Nilai suhu dan kelembapan yang terbaca dinyatakan sebagai:

$$T = T_{sensor}$$

$$H = H_{sensor}$$

dengan:

a. T = suhu ruangan dalam derajat Celcius ($^{\circ}C$)

b. H = kelembapan udara dalam persen (%)

Nilai suhu dan kelembapan ini digunakan sebagai parameter utama dalam pengendalian kipas (fan). Logika pengambilan keputusan ditentukan dengan persamaan kondisi berikut:

$$Fan1 = \begin{cases} 1, & \text{jika } T > 30^{\circ}C \text{ atau } H > 80\% \\ 0, & \text{jika } T \leq 30^{\circ}C \text{ dan } H \leq 80\% \end{cases}$$

Gambar 3. 9 Rumus Fan 1

Artinya, kipas akan menyala secara otomatis apabila suhu atau kelembapan melebihi batas yang telah ditentukan, dan akan mati ketika kondisi ruangan kembali normal.

2. Formulasi Pengukuran Intensitas Cahaya (Sensor LDR)

Sensor *Light Dependent Resistor* (LDR) digunakan untuk mendeteksi intensitas cahaya di dalam ruang kelas. Sensor LDR menghasilkan nilai analog yang berubah sesuai dengan kondisi cahaya lingkungan, di mana nilai resistansi LDR akan menurun ketika intensitas cahaya meningkat dan meningkat ketika intensitas cahaya menurun. Nilai analog dari sensor LDR dibaca oleh *Analog to Digital Converter* (ADC) pada mikrokontroler ESP32 dan dinyatakan sebagai berikut:

$$C = C_{sensor}$$

Nilai intensitas cahaya ini digunakan sebagai parameter dalam pengendalian LED putih secara otomatis. Logika pengambilan keputusan ditentukan dengan menggunakan nilai ambang batas (threshold) sebagai berikut:

$$LED = \begin{cases} 1, & \text{jika } C > C_{threshold} \\ 0, & \text{jika } C \leq C_{threshold} \end{cases}$$

Gambar 3. 10 Rumus Pengendalian LED Berdasarkan Sensor LDR

3. Formulasi Mode Operasi Sistem (Manual dan Otomatis)

Mode operasi sistem dinyatakan dengan variabel:

$$M = \begin{cases} \textit{Auto}, & \text{jika } M = 1 \\ \textit{Manual}, & \text{jika } M = 0 \end{cases}$$

Gambar 3. 11 Rumus Motor dc

Pada mode otomatis, seluruh aktuator dikendalikan berdasarkan hasil perhitungan sensor dan formulasi matematika yang telah dijelaskan sebelumnya. Sedangkan pada mode manual, status perangkat ditentukan langsung oleh perintah pengguna melalui dashboard web tanpa memperhatikan nilai sensor.

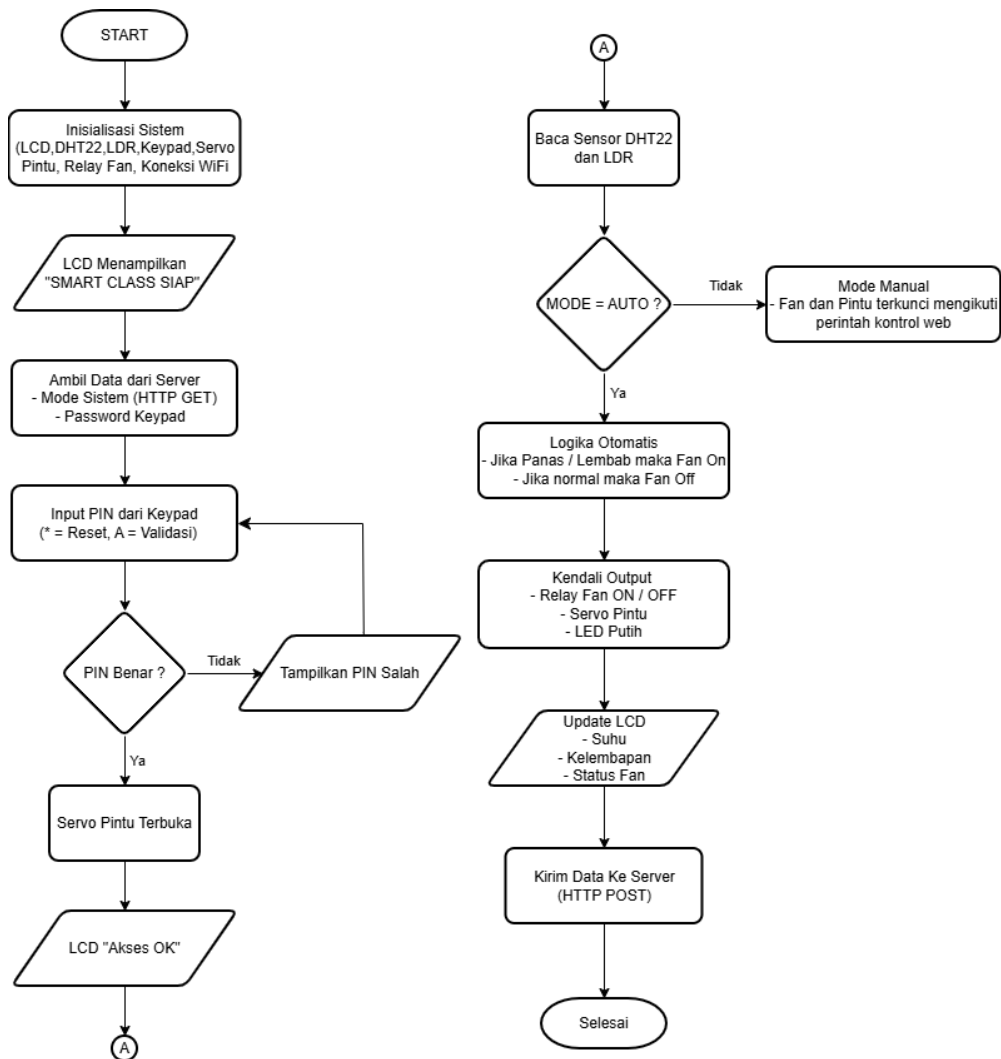
4. Formulasi Validasi PIN Keypad

Validasi PIN dilakukan dengan membandingkan input pengguna dengan PIN yang tersimpan di server. Proses validasi dinyatakan sebagai:

$$\textit{Akses} = \begin{cases} \textit{Diterima}, & \text{jika } PIN_{input} = PIN_{server} \\ \textit{Ditolak}, & \text{jika } PIN_{input} \neq PIN_{server} \end{cases}$$

Jika PIN benar, servo pintu akan bergerak membuka. Sebaliknya, jika PIN salah sebanyak tiga kali berturut-turut, sistem akan mengaktifkan buzzer dan indikator alarm.

3.5 Flowchart Sistem



Gambar 3. 12 Flowchart Sistem

Flowchart di atas menggambarkan alur kerja keseluruhan dari sistem *Smart Class* yang dikembangkan pada tugas akhir ini. Proses dimulai ketika mikrokontroler *ESP32 DevKit V1* diaktifkan dan melakukan inisialisasi seluruh perangkat yang digunakan, meliputi *Liquid Crystal Display (LCD)*, sensor *DHT22*, sensor *Light Dependent Resistor (LDR)*, *keypad 4x4*, motor *servo* pintu, modul *relay* kipas, *LED* putih, serta konfigurasi koneksi jaringan *WiFi* menggunakan *WiFi Manager*. Setelah seluruh perangkat berhasil diinisialisasi, *LCD* menampilkan informasi “SMART CLASS SIAP” sebagai tanda bahwa sistem siap digunakan.

Selanjutnya, *ESP32* mengambil data awal dari server menggunakan metode *HTTP GET*, yang meliputi mode operasi sistem (*auto* atau *manual*) serta *password* untuk autentikasi *keypad*. Proses ini bertujuan untuk menjaga sinkronisasi antara perangkat keras dan *dashboard* web.

Setelah data diperoleh, pengguna melakukan proses autentikasi dengan memasukkan PIN melalui *keypad*. Sistem kemudian melakukan validasi PIN dengan mencocokkan input pengguna dengan data *password* yang diperoleh dari server. Apabila PIN yang dimasukkan benar, sistem akan membuka pintu menggunakan motor *servo* dan *LCD* menampilkan pesan “Akses OK”. Sebaliknya, jika PIN yang dimasukkan tidak sesuai,

sistem akan menampilkan pesan kesalahan dan kembali meminta pengguna untuk memasukkan PIN yang benar.

Setelah proses autentikasi berhasil, sistem melanjutkan dengan melakukan pembacaan sensor secara berkala. Sensor *DHT22* digunakan untuk membaca nilai suhu dan kelembapan ruang kelas, sedangkan sensor *LDR* digunakan untuk mendeteksi intensitas cahaya ruangan. Data hasil pembacaan sensor ini menjadi dasar dalam pengambilan keputusan sistem.

Sistem kemudian melakukan pengecekan mode operasi. Apabila sistem berada pada *mode otomatis*, *ESP32* akan menjalankan logika otomatis untuk mengendalikan perangkat. Kipas akan dinyalakan ketika suhu atau kelembapan melebihi batas yang telah ditentukan dan dimatikan ketika kondisi kembali normal. Selain itu, *LED* putih akan dikendalikan berdasarkan nilai intensitas cahaya dari sensor *LDR*, di mana *LED* akan menyala pada kondisi cahaya terang dan mati pada kondisi cahaya rendah atau gelap.

Apabila sistem berada pada *mode manual*, maka seluruh pengendalian perangkat, termasuk kipas dan pintu, mengikuti perintah yang diberikan melalui *dashboard* web tanpa mempertimbangkan nilai sensor.

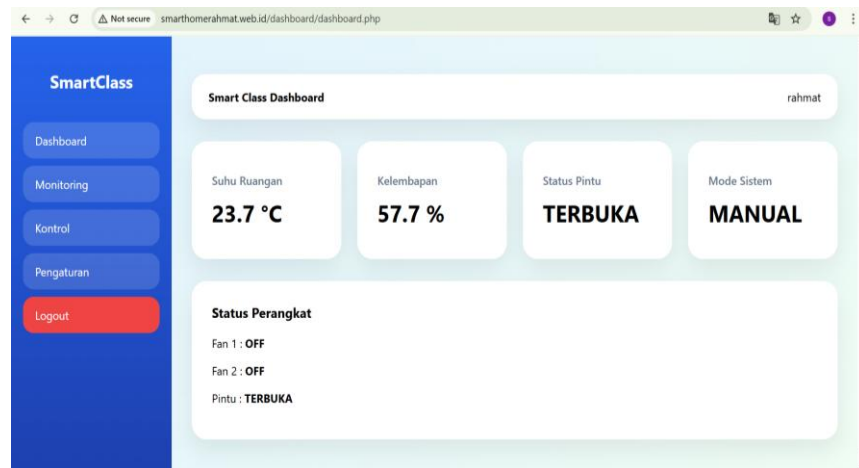
Setelah logika sistem dijalankan, *ESP32* mengendalikan perangkat output yang terdiri dari *relay* kipas, motor *servo* pintu, dan *LED* putih. Sistem kemudian memperbarui tampilan *LCD* dengan informasi terbaru yang meliputi suhu, kelembapan, serta status kipas. Pada interval waktu tertentu, *ESP32* juga mengirimkan data sensor dan status perangkat ke server menggunakan metode *HTTP POST* agar dapat disimpan ke dalam *database* dan ditampilkan secara *real-time* pada *dashboard* web.

Seluruh proses tersebut berjalan secara berulang (*loop*), sehingga sistem *Smart Class* dapat melakukan monitoring dan pengendalian fasilitas ruang kelas secara terus-menerus dan berkelanjutan sesuai dengan perancangan sistem.

3.6 Rancangan Layar

Rancangan layar sistem *Smart Class* berbasis web dibuat untuk memudahkan pengguna dalam melakukan monitoring, pengendalian, dan pengaturan sistem secara terpusat. Antarmuka sistem dirancang dengan tampilan yang sederhana dan mudah dipahami agar pengguna dapat mengakses informasi dan melakukan kontrol sistem secara *real-time* melalui web browser.

3.6.1 Layar Dashboard

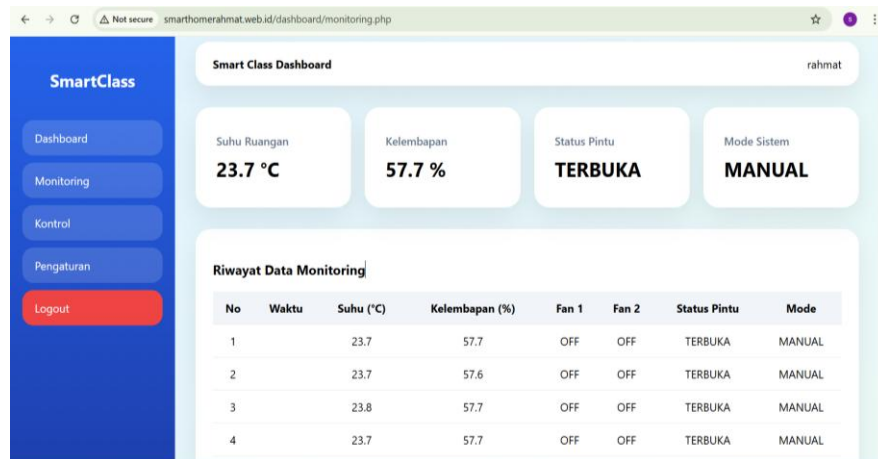


Gambar 3. 13 Layar Dashboard

Gambar 3.19 menunjukkan rancangan layar Dashboard pada sistem Smart Class berbasis web. Halaman ini merupakan halaman utama yang menampilkan ringkasan kondisi ruang kelas secara real-time.

Informasi yang ditampilkan pada halaman Dashboard meliputi suhu ruangan, kelembapan udara, kualitas udara, dan kondisi cuaca. Selain itu, halaman ini juga menampilkan status perangkat seperti kipas, lampu, dan pintu. Dengan adanya halaman Dashboard, pengguna dapat memantau kondisi sistem secara cepat tanpa perlu berpindah ke halaman lain.

3.6.2 Layar Monitoring

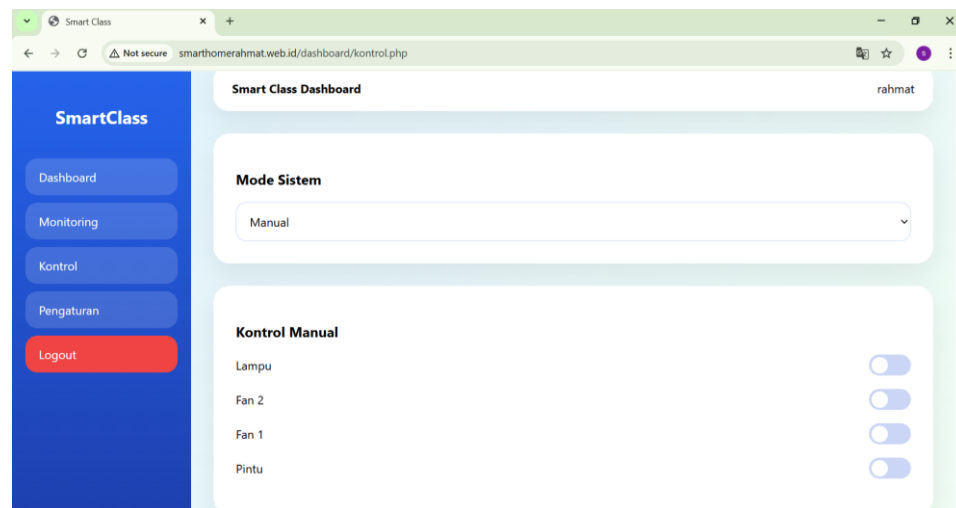


Gambar 3. 14 Layar Monitoring

Gambar 3.20 menunjukkan rancangan layar Monitoring yang digunakan untuk menampilkan data sensor secara detail. Pada halaman ini ditampilkan nilai suhu, kelembapan, kualitas udara, kondisi cuaca, serta tabel riwayat data sensor yang tersimpan pada database.

Halaman Monitoring berfungsi untuk membantu pengguna memantau perubahan kondisi ruang kelas berdasarkan data historis yang diperoleh secara berkala.

3.6.3 Layar Kontrol

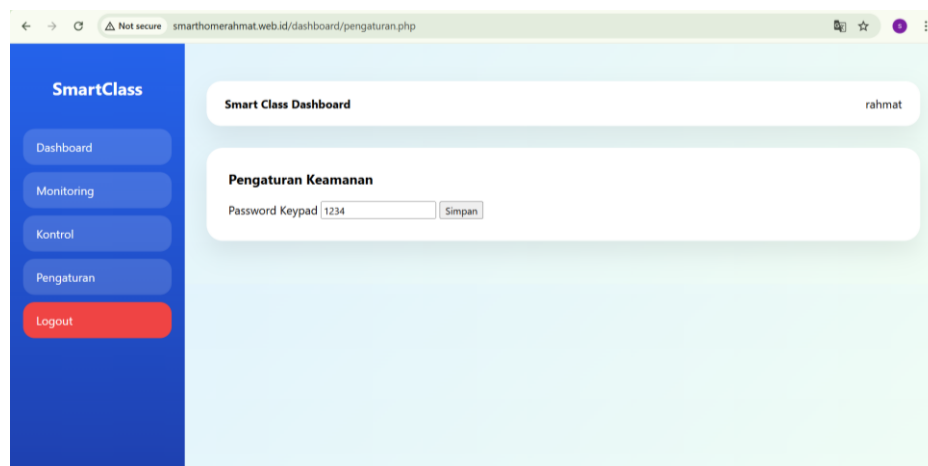


Gambar 3. 15 Layar Kontrol

Gambar 3.21 menunjukkan rancangan layar Kontrol yang digunakan untuk mengatur mode operasi sistem. Pengguna dapat memilih mode otomatis atau mode manual sesuai dengan kebutuhan.

Pada mode manual, pengguna dapat mengendalikan perangkat seperti kipas, lampu, dan pintu melalui tombol kontrol yang tersedia. Halaman Kontrol berfungsi sebagai media pengendalian perangkat secara langsung melalui sistem berbasis web.

3.6.4 Layar Pengaturan



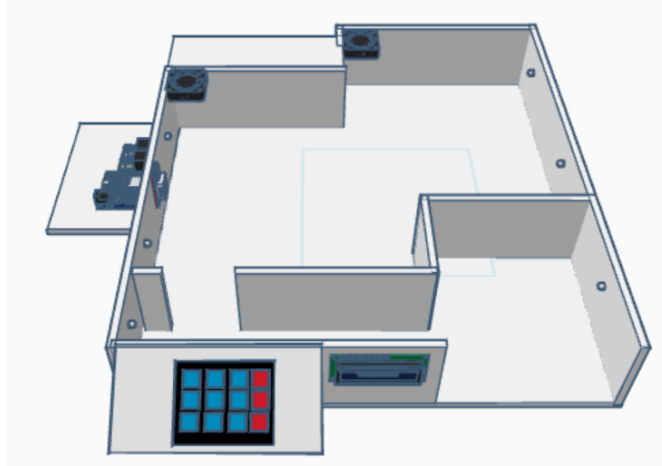
Gambar 3. 16 Layar Pengaturan

Gambar 3.22 menunjukkan rancangan layar Pengaturan yang digunakan untuk mengatur aspek keamanan sistem. Pada halaman ini pengguna dapat melakukan pengaturan password keypad yang digunakan sebagai sistem autentikasi akses pintu.

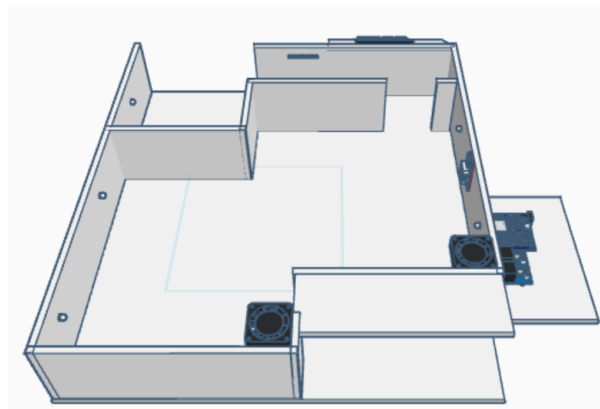
Halaman Pengaturan berfungsi untuk meningkatkan keamanan sistem Smart Class dan membatasi akses hanya kepada pengguna yang berwenang.

3.7 Rancangan Desain 3D

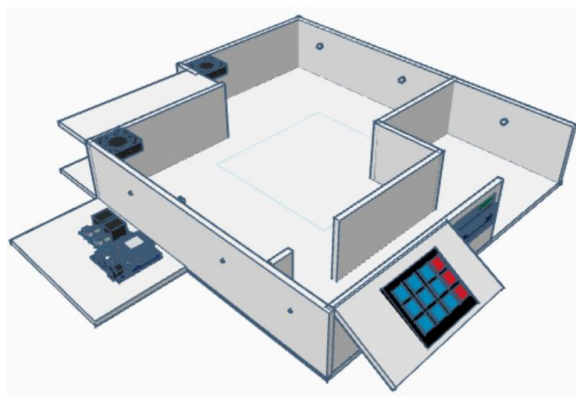
Berikut adalah gambar rancangan desain 3D dari sistem Smart Class yang menampilkan tampak atas, tampak samping kiri, tampak samping kanan, dan tampak belakang sebagai visualisasi tata letak komponen serta posisi instalasi perangkat pada prototipe.



Gambar 3. 17 Tampak Depan



Gambar 3. 18 Tampak Belakang



Gambar 3. 19 Tampak Samping

3.8 Rancangan Pengujian

Dalam proses pembuatan prototipe sistem *Smart Class*, dilakukan penyusunan rancangan pengujian sebelum sistem diimplementasikan secara menyeluruh. Rancangan pengujian ini bertujuan untuk memastikan bahwa setiap perangkat input dan aktuator yang digunakan dapat berfungsi sesuai dengan logika sistem yang telah dirancang.

Rancangan pengujian mencakup pengujian pembacaan sensor *DHT22* untuk mengetahui kemampuan sistem dalam memonitor suhu dan kelembapan ruang kelas, pengujian *keypad* 4×4 sebagai media autentikasi akses pintu, serta pengujian motor *servo* sebagai aktuator pembuka dan penutup pintu. Selain itu, dilakukan juga pengujian modul *relay* dalam mengendalikan kipas baik pada mode otomatis maupun mode manual, serta pengujian *LCD* sebagai media tampilan informasi kondisi sistem.

Setiap komponen diuji secara terpisah untuk memastikan fungsinya berjalan dengan baik, kemudian diuji kembali secara terintegrasi sebagai satu kesatuan sistem. Pengujian terintegrasi dilakukan untuk memastikan bahwa mikrokontroler *ESP32 DevKit V1* mampu memproses data sensor secara *real-time* dan mengeksekusi perintah pengendalian perangkat sesuai dengan mode operasi yang dipilih, baik *mode otomatis* maupun *mode manual* melalui *dashboard web*. Dengan adanya rancangan pengujian ini, diharapkan sistem *Smart Class* dapat berjalan secara stabil, responsif, dan sesuai dengan kebutuhan penggunaan di lingkungan ruang kelas.

Tabel 3. 8 Rancangan Pengujian

No	Komponen	Tujuan Pengujian	Metode Pengujian	Hasil yang Diharapkan
1	Keypad 4×4	Menguji proses input PIN pengguna	Menekan kombinasi PIN benar / salah	PIN benar → pintu terbuka; PIN salah → counter bertambah; 3× salah → alarm aktif
2	Sensor DHT22	Menguji pembacaan suhu & kelembapan	Mengarahkan sensor pada lingkungan berbeda (panas/lembap)	Nilai suhu & kelembapan berubah sesuai kondisi ruangan
3	Sensor LDR	Menguji pendeteksian intensitas cahaya	Mengubah kondisi cahaya (terang/gelap) di sekitar sensor	Sistem mendeteksi perubahan cahaya dan mengontrol LED secara otomatis sesuai kondisi
4	Servo Pintu	Menguji mekanisme gerak aktuator	Memberikan perintah buka/tutup	Servo bergerak pada sudut yang sesuai (0°/90°)

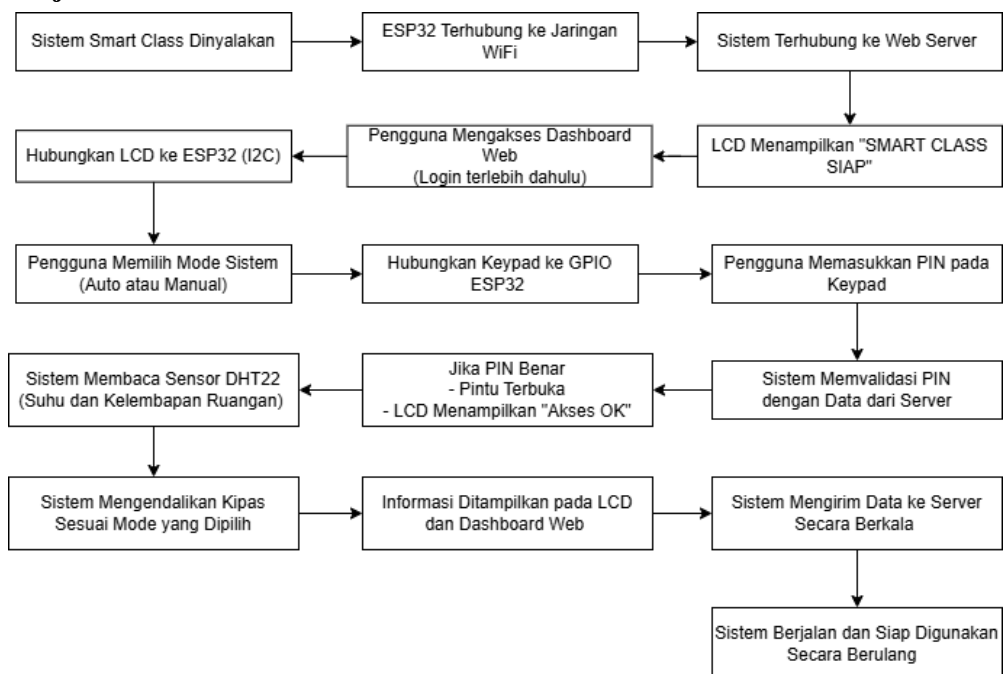
5	Relay (Kipas 1 & 2)	Menguji kendali beban kipas	Mengaktifkan logika otomatis & manual	Relay <i>ON/OFF</i> sesuai perintah; kipas menyala stabil
6	LED Putih	Menguji LED	Mengaktifkan kondisi otomatis	LED menyala sesuai kondisi sistem
7	LCD I2C 20×4	Menguji tampilan proses sistem	Menyalakan sistem dan mengamati update LCD	LCD menampilkan suhu, cuaca, status perangkat, dan pesan PIN
8	Koneksi API / Web Server	Menguji komunikasi GET & POST	Mengambil status perangkat & mengirim data sensor	Data berhasil terkirim dan dashboard menampilkan perubahan secara realtime

BAB IV

IMPLEMENTASI DAN EVALUASI PROGRAM

Setelah tahap perancangan yang telah dibahas pada BAB III, maka pada BAB IV ini dilakukan proses implementasi dan pengujian terhadap sistem *Smart Class* yang telah dirancang. Bab ini berfokus pada bagaimana setiap komponen diintegrasikan menjadi satu kesatuan sistem, cara kerja sistem saat dijalankan, serta hasil pengujian yang diperoleh dari masing-masing fungsi yang telah diuji. Melalui pembahasan pada bab ini, dapat diketahui apakah seluruh perancangan yang telah dilakukan sebelumnya dapat berjalan secara optimal dan sesuai dengan tujuan penelitian.

4.1 Cara Kerja Sistem



Gambar 4. 1 Diagram Blok Cara Kerja

Alur kerja sistem *Smart Class* dimulai ketika sistem pertama kali dinyalakan. Pada tahap awal, mikrokontroler *ESP32 DevKit V1* diaktifkan dan melakukan koneksi ke jaringan *WiFi* yang tersedia. Setelah koneksi jaringan berhasil, sistem terhubung dengan *web server* sebagai media pertukaran data antara perangkat keras dan *dashboard web*.

Setelah sistem terhubung dengan *web server*, *Liquid Crystal Display (LCD)* akan menampilkan informasi awal berupa pesan “SMART CLASS SIAP” sebagai indikasi bahwa sistem telah aktif dan siap digunakan. Pengguna kemudian dapat mengakses *dashboard web* melalui perangkat yang terhubung ke jaringan internet dengan melakukan proses *login* terlebih dahulu.

Pengguna selanjutnya memilih mode sistem yang akan digunakan, yaitu *mode otomatis* atau *mode manual*, melalui *dashboard web*. Mode yang dipilih akan dikirimkan ke *ESP32* dan digunakan sebagai acuan dalam proses pengendalian sistem.

Untuk mengakses ruang kelas, pengguna melakukan proses autentikasi dengan memasukkan PIN melalui *keypad* yang terhubung ke *ESP32* melalui pin *General Purpose Input Output* (GPIO). Data PIN yang dimasukkan oleh pengguna kemudian divalidasi oleh sistem dengan membandingkannya dengan data PIN yang tersimpan di *web server*. Apabila PIN yang dimasukkan benar, sistem akan membuka pintu menggunakan motor *servo* dan *LCD* akan menampilkan pesan “Akses OK”.

Setelah proses autentikasi berhasil, sistem mulai melakukan pembacaan sensor secara berkala. Sensor *DHT22* digunakan untuk membaca nilai suhu dan kelembapan ruang kelas, sedangkan sensor *Light Dependent Resistor* (*LDR*) digunakan untuk mendeteksi intensitas cahaya ruangan. Data dari kedua sensor tersebut diproses oleh *ESP32* sebagai dasar pengambilan keputusan sistem.

Berdasarkan nilai intensitas cahaya yang diperoleh dari sensor *LDR*, sistem secara otomatis mengendalikan *LED* putih sebagai lampu ruang kelas. Pada kondisi cahaya ruangan terang, *LED* akan dimatikan, sedangkan pada kondisi cahaya rendah atau gelap, *LED* akan menyala sesuai dengan logika sistem yang telah dirancang.

Selain pengendalian *LED*, data suhu dan kelembapan dari sensor *DHT22* digunakan untuk mengendalikan kipas sesuai dengan mode sistem yang dipilih. Pada *mode otomatis*, kipas akan menyala ketika suhu atau kelembapan melebihi batas yang ditentukan, sedangkan pada *mode manual* pengendalian kipas mengikuti perintah dari *dashboard web*.

Seluruh informasi hasil pembacaan sensor, termasuk suhu, kelembapan, dan intensitas cahaya, serta status perangkat seperti kipas, *LED*, dan pintu, ditampilkan pada *LCD* dan *dashboard web* secara *real-time*. Selain itu, sistem secara berkala mengirimkan data sensor dan status perangkat ke *web server* untuk disimpan ke dalam *database*.

Seluruh proses tersebut berjalan secara berulang (*loop*), sehingga sistem *Smart Class* selalu berada dalam kondisi siap digunakan untuk melakukan monitoring dan pengendalian fasilitas ruang kelas secara berkelanjutan.

4.2 Hasil Pengujian Perangkat Keras

Pengujian perangkat keras dilakukan untuk memastikan bahwa seluruh komponen yang digunakan dalam prototype sistem *Smart Class* berbasis *Internet of Things* (*IoT*) dapat berfungsi dengan baik dan sesuai dengan perancangan yang telah dilakukan. Pengujian ini bertujuan untuk mengetahui kinerja masing-masing perangkat keras dalam mendukung sistem secara keseluruhan.

Setiap pengujian dilakukan secara terpisah pada masing-masing komponen utama yang digunakan, meliputi *keypad* 4×4 sebagai perangkat input autentikasi, sensor *DHT22* sebagai pembaca suhu dan kelembapan ruangan, motor *servo* sebagai penggerak pintu, modul *relay* untuk mengendalikan kipas, serta *Liquid Crystal Display* (*LCD*) sebagai media tampilan informasi. Pengujian dilakukan dengan cara memberikan input tertentu pada setiap komponen dan mengamati respons output yang dihasilkan oleh sistem.

Hasil dari setiap pengujian dianalisis untuk memastikan bahwa perangkat keras dapat bekerja sesuai dengan fungsi yang diharapkan dan terintegrasi dengan baik dengan mikrokontroler *ESP32*. Dengan adanya pengujian perangkat keras ini, dapat diketahui tingkat keberhasilan integrasi antar komponen serta kesiapan sistem *Smart Class* untuk dioperasikan, baik dalam mode otomatis maupun mode manual melalui sistem berbasis web.

4.2.1 Pengujian keypad 4x4

Pengujian keypad 4×4 dilakukan untuk memastikan bahwa setiap tombol pada keypad dapat terdeteksi dengan benar oleh mikrokontroler ESP32 serta mampu menjalankan fungsi autentikasi akses pintu. Proses pengujian dilakukan dengan menekan tombol angka secara berurutan untuk membentuk PIN, kemudian menekan tombol konfirmasi untuk melakukan verifikasi.

Tabel 4. 1 Pengujian Keypad

No	Input PIN	Respon LCD	Status Sistem
1	PIN benar	PIN BENAR	Pintu Terbuka
2	PIN salah	PIN SALAH	Akses Ditolak
3	Tombol reset	PIN direset	Standby

Berdasarkan hasil pengujian, keypad 4×4 pada Tabel 4.1 mampu membaca setiap input tombol dengan baik tanpa adanya kesalahan pembacaan. Sistem berhasil membandingkan input PIN dengan data password yang diperoleh dari server. Ketika PIN yang dimasukkan benar, sistem memberikan respon berupa tampilan pesan keberhasilan pada LCD dan membuka pintu melalui motor servo. Sebaliknya, apabila PIN salah, sistem menolak akses dan menampilkan peringatan. Hal ini menunjukkan bahwa mekanisme autentikasi berbasis keypad berjalan sesuai dengan fungsi keamanan yang dirancang.

4.2.2 Pengujian Sensor DHT 22

Pengujian sensor DHT22 dilakukan untuk mengukur tingkat akurasi sensor dalam membaca suhu dan kelembapan ruangan. Data pengujian diambil secara real-time dan disimpan ke dalam tabel sensor_data pada database.

Tabel 4. 2 Pengujian Sensor DHT 22

No	Waktu	Suhu (°C)	Kelembapan (%)	Keterangan
1	11:41:24	23.3	87.2	Berhasil
2	11:41:25	23.4	83.5	Berhasil
3	11:41:38	23.5	83.3	Berhasil
4	11:41:44	23.5	83.2	Berhasil
5	11:41:56	23.5	82.6	Berhasil

Hasil pengujian menunjukkan bahwa sensor DHT22 mampu membaca suhu dan kelembapan secara stabil dengan perubahan nilai yang relatif kecil antar waktu pengukuran. Data sensor berhasil dikirim ke server dan tersimpan di database tanpa adanya kehilangan data. Informasi suhu dan kelembapan ini selanjutnya digunakan oleh sistem sebagai dasar pengambilan keputusan dalam mode otomatis, seperti pengendalian kipas ruangan.

4.2.3 Pengujian Sensor LDR

Pengujian sensor *Light Dependent Resistor* (LDR) dilakukan untuk mengetahui kemampuan sensor dalam mendeteksi perubahan intensitas cahaya di dalam ruang kelas serta memastikan sensor dapat bekerja sesuai dengan perancangan sistem *Smart Class*. Pengujian dilakukan dengan mengamati nilai pembacaan sensor LDR pada kondisi cahaya yang berbeda, yaitu kondisi terang dan kondisi gelap.

Data hasil pengujian sensor LDR diperoleh secara *real-time* melalui mikrokontroler *ESP32 DevKit V1* dan disimpan ke dalam tabel *sensor_data* pada *database*. Nilai intensitas cahaya yang diperoleh kemudian digunakan sebagai dasar pengendalian *LED* putih secara otomatis pada sistem.

Tabel 4. 3 Pengujian Sensor LDR

No	Waktu	Nilai Cahaya (ADC)	Kondisi Cahaya	Status LED	Keterangan
1	11:45:12	2850	Terang	ON	Berhasil
2	11:45:25	2785	Terang	ON	Berhasil
3	11:45:39	1920	Redup	OFF	Berhasil
4	11:45:52	1650	Gelap	OFF	Berhasil

Hasil pengujian menunjukkan bahwa sensor LDR mampu mendeteksi perubahan intensitas cahaya dengan baik. Pada kondisi cahaya terang, nilai ADC yang dihasilkan sensor berada di atas nilai ambang batas sehingga *LED* putih menyala secara otomatis. Sebaliknya, pada kondisi cahaya redup atau gelap, nilai ADC berada di bawah ambang batas dan *LED* akan mati.

Data intensitas cahaya yang diperoleh dari sensor LDR juga berhasil dikirim ke server dan disimpan ke dalam *database* tanpa mengalami kehilangan data. Berdasarkan hasil pengujian tersebut, dapat disimpulkan bahwa sensor LDR berfungsi dengan baik sebagai perangkat input untuk pengendalian pencahayaan ruang kelas secara otomatis pada sistem *Smart Class*.

4.2.4 Pengujian Motor Servo

Pengujian motor servo dilakukan untuk memastikan aktuator dapat bergerak sesuai perintah sistem.

Tabel 4. 4 Pengujian Motor Servo

No	Servo	Parameter Uji	Nilai Sistem	Status Servo	Keterangan
1	Servo Pintu	PIN Benar	door_state = 1	Terbuka	Berhasil
2	Servo Pintu	PIN Salah	door_state = 0	Tertutup	Berhasil

3	Servo Pintu	Mode Auto	mode = auto	Tertutup	Berhasil
---	-------------	-----------	-------------	----------	----------

Pengujian motor servo dilakukan untuk memastikan aktuator mampu bergerak sesuai dengan perintah sistem. Servo pintu diuji berdasarkan input PIN melalui keypad serta status `door_state` yang tersimpan pada sistem. Ketika PIN dimasukkan dengan benar, servo pintu bergerak membuka, sedangkan ketika PIN salah atau sistem berada pada kondisi terkunci, servo pintu tetap tertutup.

4.2.5 Pengujian Relay dan Fan

Pengujian relay dan fan dilakukan untuk memastikan modul relay mampu mengendalikan kipas sesuai dengan perintah sistem. Pengujian dilakukan dengan mengaktifkan dan menonaktifkan kipas baik pada mode otomatis maupun mode manual melalui dashboard web.

Tabel 4. 5 Pengujian Relay dan Fan

No	Mode Operasi	Kondisi Sensor	Status Fan	Keterangan
1	Otomatis	Suhu Normal	Mati	Berhasil
2	Otomatis	Suhu Tinggi	Menyala	Berhasil
3	Manual	–	Menyala	Berhasil
4	Manual	–	Mati	Berhasil

Pada mode otomatis, relay bekerja berdasarkan data suhu, kelembapan, dan kualitas udara yang dibaca oleh sensor. Kipas menyala ketika nilai sensor melebihi ambang batas yang telah ditentukan dan mati saat kondisi kembali normal. Pada mode manual, kipas dapat dikendalikan langsung melalui web tanpa memperhatikan nilai sensor. Hasil pengujian menunjukkan bahwa relay dan fan dapat bekerja dengan baik sesuai perintah sistem.

4.2.6 Pengujian Liquid Crystal Display 20x4 I2C

Pengujian LCD I2C 20x4 dilakukan untuk memastikan tampilan informasi dapat ditampilkan dengan benar dan stabil. Informasi yang ditampilkan meliputi suhu ruangan, kondisi cuaca, status kipas, serta status pintu.

Tabel 4. 6 Pengujian LCD

No	Informasi Ditampilkan	Hasil Tampilan	Keterangan	No
1	Suhu dan Kelembapan	Sesuai	Berhasil	1
2	Status Fan	Sesuai	Berhasil	2
3	Status Pintu	Sesuai	Berhasil	3

Hasil pengujian menunjukkan bahwa LCD mampu menampilkan seluruh informasi dengan jelas dan sesuai dengan data yang diproses oleh ESP32. Perubahan nilai sensor dan status perangkat dapat langsung diperbarui pada LCD secara real-time. Dengan demikian, LCD I2C 20x4 dinyatakan berfungsi dengan baik sebagai media informasi lokal pada sistem Smart Class.

4.2.7 Pengujian Komunikasi API (GET/POST)

Pengujian komunikasi API dilakukan untuk memastikan proses pertukaran data antara ESP32 dan web server berjalan dengan baik. Metode HTTP GET digunakan untuk mengambil data pengaturan sistem seperti mode operasi dan status perangkat, sedangkan metode HTTP POST digunakan untuk mengirimkan data sensor dan status aktuator ke server.

Tabel 4. 7 Pengujian Komunikasi API (GET/POST)

No	Metode API	Data Dikirim/Diterima	Status	Keterangan
1	GET	Mode Sistem	Berhasil	Data diterima
2	GET	Status Perangkat	Berhasil	Data diterima
3	POST	Data Sensor	Berhasil	Data tersimpan
4	POST	Status Aktuator	Berhasil	Data tersimpan

Hasil pengujian menunjukkan bahwa ESP32 berhasil melakukan komunikasi dua arah dengan server. Data yang dikirim menggunakan metode POST dapat tersimpan di database dan ditampilkan pada dashboard web, sedangkan data yang diambil menggunakan metode GET dapat diterima dan dijalankan oleh sistem dengan benar. Berdasarkan hasil tersebut, komunikasi API pada sistem Smart Class dinyatakan berjalan dengan baik.

4.3 Hasil Pengujian Perangkat Lunak

Berdasarkan seluruh rangkaian pengujian yang telah dilakukan pada sistem Smart Class berbasis *Internet of Things (IoT)* menggunakan *ESP32 DevKit V1*, dapat dianalisis bahwa perangkat lunak yang dikembangkan mampu bekerja sesuai dengan perancangan sistem. Pengujian difokuskan pada kinerja program, proses pembacaan data sensor, pengolahan logika kendali, komunikasi dengan server, serta integrasi dengan dashboard web.

Hasil pengujian menunjukkan bahwa program pada ESP32 mampu membaca data dari sensor DHT22 secara konsisten dan stabil. Nilai suhu dan kelembapan yang diterima diproses oleh sistem sebagai parameter dalam pengambilan keputusan otomatis. Logika program berjalan sesuai dengan algoritma yang telah dirancang, di mana ketika nilai suhu atau kelembapan melebihi batas ambang yang ditentukan, sistem secara otomatis

mengaktifkan kipas melalui relay. Sebaliknya, ketika nilai kembali normal, sistem mematikan kipas secara otomatis. Hal ini menunjukkan bahwa proses pengolahan data sensor dan eksekusi logika kendali berjalan dengan baik tanpa error.

Pengujian pembacaan sensor Light Dependent Resistor (LDR) juga menunjukkan bahwa perangkat lunak mampu mengolah nilai analog menjadi parameter kendali pencahayaan. Program secara otomatis membandingkan nilai intensitas cahaya dengan batas threshold yang telah ditentukan. Berdasarkan hasil pengujian, sistem mampu mengaktifkan atau menonaktifkan LED sesuai kondisi cahaya ruangan. Proses ini berlangsung secara real-time dan tidak ditemukan kegagalan dalam eksekusi logika program.

Pada pengujian fitur autentikasi keypad 4×4, perangkat lunak berhasil membaca input PIN dari pengguna dan membandingkannya dengan data yang tersimpan pada server. Proses validasi dilakukan melalui komunikasi data berbasis HTTP. Ketika PIN yang dimasukkan sesuai, sistem menjalankan perintah pembukaan pintu serta menampilkan informasi akses pada LCD. Sebaliknya, ketika PIN tidak sesuai, sistem menolak akses dan kembali ke kondisi awal. Hasil ini menunjukkan bahwa algoritma autentikasi dan pengelolaan data akses berjalan sesuai dengan rancangan.

Pengujian kontrol perangkat melalui dashboard web menunjukkan bahwa sistem mampu menerima perintah dari antarmuka web menggunakan metode HTTP POST dan HTTP GET. Perintah yang dikirimkan dari dashboard berhasil diterima dan diproses oleh ESP32 tanpa kehilangan data. Status perangkat yang diperbarui juga berhasil dikirim kembali ke server dan ditampilkan secara real-time pada dashboard. Hal ini membuktikan bahwa komunikasi antara perangkat IoT dan sistem berbasis web berjalan stabil dan responsif.

Selain itu, pengujian mode otomatis dan manual menunjukkan bahwa perangkat lunak mampu mengatur prioritas logika kendali dengan baik. Pada mode otomatis, sistem bekerja berdasarkan pembacaan sensor dan algoritma yang telah ditentukan. Pada mode manual, kontrol sistem dialihkan sepenuhnya ke pengguna melalui dashboard web. Proses pergantian mode berjalan tanpa konflik logika maupun gangguan pada sistem.

Selama proses pengujian berulang, program yang berjalan pada ESP32 tidak mengalami crash, hang, maupun restart mendadak. Loop program berjalan stabil dengan waktu respon rata-rata satu hingga dua detik tergantung kondisi jaringan internet. Tidak ditemukan error komunikasi yang signifikan selama pengujian berlangsung.

Secara keseluruhan, hasil pengujian perangkat lunak menunjukkan bahwa sistem Smart Class berbasis IoT telah mampu menjalankan fungsi monitoring, pengendalian otomatis, autentikasi akses, serta komunikasi data berbasis web sesuai dengan tujuan penelitian. Integrasi antara logika program pada ESP32 dan sistem web service berjalan dengan baik, sehingga perangkat lunak dapat dinyatakan berfungsi secara optimal dan siap diimplementasikan pada lingkungan ruang kelas.

4.4 Kelebihan Sistem

Sistem *Smart Class* yang dirancang memiliki beberapa kelebihan yang mendukung penerapannya dalam lingkungan ruang kelas. Salah satu kelebihan utama sistem ini adalah kemampuannya dalam mengintegrasikan perangkat input dan aktuator ke dalam satu sistem kendali yang terpusat menggunakan mikrokontroler *ESP32*. Integrasi ini

memungkinkan proses monitoring dan pengendalian perangkat dapat dilakukan secara efisien dan terkoordinasi.

Sistem mampu melakukan monitoring kondisi ruang kelas berupa suhu dan kelembapan secara *real-time* serta mengendalikan perangkat kipas baik secara otomatis maupun manual melalui *dashboard web*. Pada mode otomatis, sistem dapat menyesuaikan kondisi lingkungan ruang kelas berdasarkan parameter sensor, sedangkan pada mode manual pengguna dapat melakukan pengendalian perangkat secara langsung melalui antarmuka web. Hal ini memberikan fleksibilitas dalam pengoperasian sistem sesuai dengan kebutuhan pengguna.

Penggunaan mikrokontroler *ESP32* sebagai pusat kendali memungkinkan sistem terhubung dengan jaringan internet sehingga proses pertukaran data dengan *server* dapat dilakukan secara *real-time*. Selain itu, penerapan sistem keamanan akses pintu menggunakan *keypad* memberikan nilai tambah dalam mengontrol akses masuk ke ruang kelas. Sistem juga dirancang dalam bentuk *prototype*, sehingga memudahkan pengembangan dan penyesuaian fitur di masa mendatang sesuai dengan kebutuhan pengguna.

4.5 Kekurangan Sistem

Meskipun sistem *Smart Class* yang dirancang memiliki berbagai kelebihan, sistem ini juga memiliki beberapa keterbatasan yang perlu diperhatikan. Salah satu kekurangan sistem adalah keterbatasan parameter monitoring lingkungan, di mana sistem hanya memantau suhu dan kelembapan ruang kelas. Kondisi lingkungan lain yang juga dapat mempengaruhi kenyamanan ruang kelas belum diakomodasi dalam sistem ini.

Ketergantungan sistem terhadap koneksi jaringan internet juga menjadi salah satu keterbatasan. Apabila koneksi *WiFi* mengalami gangguan atau tidak tersedia, maka proses komunikasi antara *ESP32* dan *server* tidak dapat berjalan secara optimal. Kondisi ini dapat mempengaruhi fungsi monitoring dan pengendalian perangkat melalui *dashboard web*, khususnya pada mode manual.

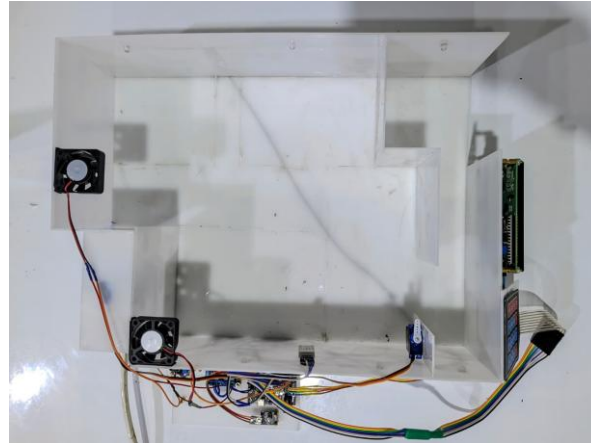
Selain itu, sistem keamanan yang digunakan masih terbatas pada autentikasi PIN melalui *keypad*. Meskipun metode ini cukup efektif untuk mengontrol akses pintu, tingkat keamanan sistem masih dapat ditingkatkan dengan menambahkan metode autentikasi lain agar lebih aman dan fleksibel. Sistem juga masih berupa *prototype*, sehingga belum sepenuhnya diuji untuk penggunaan jangka panjang dan kondisi lingkungan yang beragam.

Keterbatasan lain dari sistem ini adalah belum adanya mekanisme pencatatan kesalahan (*error handling*) dan notifikasi secara otomatis apabila terjadi gangguan pada perangkat keras atau jaringan. Oleh karena itu, pemantauan kondisi sistem masih memerlukan pengawasan dari pengguna melalui *dashboard web*.

4.6 Foto Sistem

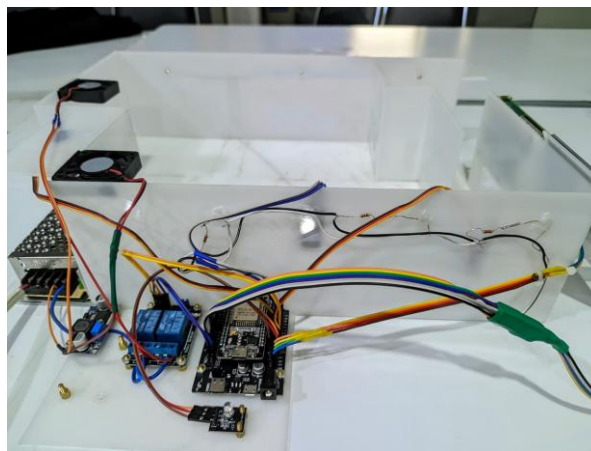
Bagian ini menampilkan dokumentasi berupa foto *prototype* Sistem *Smart Class* berbasis Internet of Things yang telah dirancang dan diimplementasikan. Foto sistem digunakan untuk menunjukkan bentuk fisik *prototype* serta susunan komponen perangkat keras yang digunakan dalam sistem.

Melalui dokumentasi ini, dapat dilihat integrasi antara mikrokontroler ESP32, sensor, dan aktuator yang terpasang pada prototype. Foto sistem juga berfungsi sebagai bukti implementasi perangkat keras yang mendukung proses pengujian dan evaluasi sistem Smart Class yang telah dilakukan.



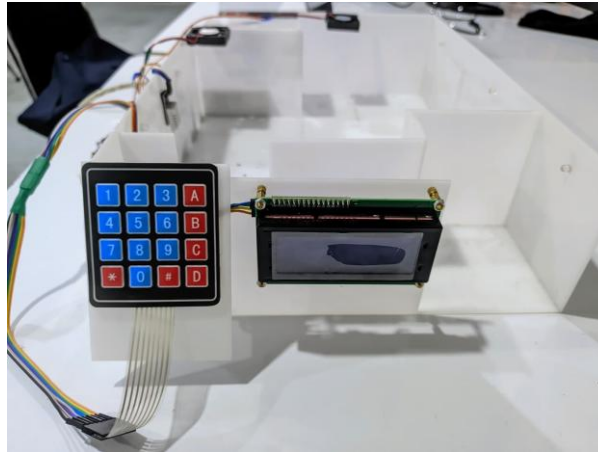
Gambar 4. 2 Tampak atas

Gambar 4.2 menunjukkan prototype Smart Class berbasis IoT yang memperlihatkan susunan komponen di dalam box, meliputi kipas sebagai simulasi sirkulasi udara, rangkaian kendali berbasis ESP32 DevKit V1, modul relay, serta jalur pengkabelan yang terintegrasi.



Gambar 4. 3 Tampak samping

Gambar 4.3 menunjukkan prototype Smart Class dengan penempatan mikrokontroler ESP32 DevKit V1, modul relay, dan pengkabelan antar komponen sebagai bagian dari sistem kendali dan monitoring.



Gambar 4. 4 Tampak Depan

Gambar 4.4 menunjukkan prototype Smart Class yang menampilkan modul keypad sebagai sistem keamanan akses pintu dan modul LCD sebagai media penampil informasi kondisi sistem.

BAB V

PENUTUP

5.1 Kesimpulan

Berdasarkan seluruh tahapan yang telah dilalui, mulai dari perancangan sistem, proses implementasi, hingga pengujian yang dilakukan, dapat disimpulkan bahwa sistem Smart Class berbasis Internet of Things (IoT) yang dikembangkan pada penelitian ini telah berhasil menjawab rumusan masalah yang diajukan.

Proses perancangan dan pembangunan prototype Smart Class menunjukkan bahwa berbagai perangkat keras dapat diintegrasikan ke dalam satu sistem yang terpusat dan saling terhubung dengan baik. Sistem mampu melakukan monitoring kondisi ruang kelas, yang meliputi suhu, kelembapan udara, dan intensitas cahaya ruangan. Sensor DHT22 digunakan untuk memperoleh data suhu dan kelembapan, sedangkan sensor Light Dependent Resistor (LDR) digunakan untuk mendeteksi tingkat pencahayaan ruangan. Selain fungsi monitoring, sistem juga mampu mengendalikan fasilitas ruang kelas seperti kipas, lampu, dan pintu secara terkoordinasi. Hasil pengujian menunjukkan bahwa seluruh komponen dapat berfungsi sesuai dengan perancangan dan bekerja secara terintegrasi dalam satu sistem.

Penggunaan mikrokontroler ESP32 DevKit V1 sebagai pusat kendali terbukti efektif dalam mengolah data sensor dan mengendalikan perangkat ruang kelas. ESP32 mampu melakukan komunikasi dua arah dengan server melalui metode HTTP GET dan HTTP POST, sehingga data sensor dan status perangkat dapat dikirim, disimpan ke dalam database, serta ditampilkan secara real-time melalui dashboard web. Hal ini menunjukkan bahwa sistem monitoring dan pengendalian berbasis web dapat diimplementasikan dengan baik pada sistem Smart Class ini.

Penerapan dua mode pengoperasian, yaitu mode otomatis dan mode manual, juga dapat berjalan sesuai dengan tujuan perancangan. Pada mode otomatis, sistem mampu mengambil keputusan berdasarkan nilai sensor, seperti menyalakan kipas ketika suhu atau kelembapan melebihi batas yang telah ditentukan serta mengendalikan lampu berdasarkan intensitas cahaya ruangan. Lampu akan menyala ketika kondisi ruangan gelap dan akan mati ketika kondisi ruangan terang. Sementara itu, pada mode manual, pengguna diberikan kebebasan untuk mengendalikan perangkat secara langsung melalui dashboard web tanpa bergantung pada nilai sensor. Keberadaan dua mode ini memberikan fleksibilitas dalam pengelolaan fasilitas ruang kelas sesuai dengan kebutuhan pengguna.

Dari sisi keamanan, sistem akses pintu yang menggunakan keypad sebagai media input PIN dapat berfungsi dengan baik dalam membatasi akses masuk ke ruang kelas. Proses autentikasi berjalan sesuai dengan logika sistem dan terintegrasi dengan pengendalian motor servo sebagai mekanisme buka dan tutup pintu.

Secara keseluruhan, sistem Smart Class berbasis IoT yang dikembangkan mampu meningkatkan efisiensi dalam monitoring dan pengendalian fasilitas ruang kelas. Prototype ini menunjukkan bahwa penerapan teknologi Internet of Things dapat digunakan sebagai solusi terintegrasi dalam pengelolaan ruang kelas, serta dapat dijadikan dasar untuk pengembangan sistem Smart Class yang lebih lanjut dan berskala lebih luas.

5.2 Saran

Berdasarkan hasil penelitian yang telah dilakukan, terdapat beberapa saran yang dapat diberikan untuk pengembangan sistem ke depannya, yaitu:

1. Sistem dapat dikembangkan dengan menambahkan mekanisme penyimpanan data lokal sebagai cadangan apabila koneksi internet mengalami gangguan.
2. Pengujian sistem dapat dilakukan secara langsung pada ruang kelas sebenarnya untuk memperoleh hasil yang lebih akurat.
3. Antarmuka dashboard web dapat dikembangkan agar lebih informatif dan mudah digunakan.
4. Penambahan sensor atau fitur lain dapat dilakukan untuk meningkatkan fungsi dan keandalan sistem Smart Class.

DAFTAR PUSTAKA

- Aswaldi, H. (2025). Penerapan Teknologi Internet of Things (IoT) untuk Monitoring Kualitas Udara dalam Ruangan Article history. *Journal of Computer Science and Information Technology*, 1(2). <https://doi.org/10.70716/jocsit.v1i2.225>
- Budiyanto, A., Pramudita, G. B., Adinandra, S., Studi, P., & Elektro, T. (2020). Kontrol Relay dan Kecepatan Kipas Angin Direct Current (DC) dengan Sensor Suhu LM35 Berbasis Internet of Things (IoT). *Jurnal Ilmiah Elektroteknika*, 19, 43–54.
- Espressif Systems. (2023). *ESP32 Series Datasheet Version 5.2 2.4 GHz Wi-Fi + Bluetooth® + Bluetooth LE SoC Including*. www.espressif.com
- Fauziman, H., & Mukhaiyar, R. (2023). Rancang Bangun Sistem Keamanan Pintu Menggunakan Fingerprint Berbasis Internet Of Things (IoT). *JTEIN: Jurnal Teknik Elektro Indonesia*, 4(2). <https://doi.org/10.24036/jtein.v4i2.438>
- Irdayanti, Y., Alrasyid, J., & Azzahra, A. (2025). *Sistem Monitoring dan Pengaturan Kualitas Udara Dalam Ruangan Berbasis Internet of Things*. 10(1). <https://doi.org/10.31851/ampere>
- Kalbuana, N., & Kurnianto, B. (2024). Desain Sistem Deteksi Asap Berbasis Sensor Mikrokontroler Sebagai Upaya Pencegahan Kebakaran. *MALCOM: Indonesian Journal of Machine Learning and Computer Science*, 4(1), 266–272. <https://doi.org/10.57152/malcom.v4i1.1158>
- Latief, M. F., Irmansyah, I., & Rosyidi, L. (2025). Sistem Pemantauan Ruang Kelas berbasis Internet of Things (IoT) untuk Proses Pendidikan yang Efektif. *Digital Transformation Technology*, 4(2), 1278–1284. <https://doi.org/10.47709/digitech.v4i2.5485>
- Made, N., Astari, A. J., Ardiyasa, W., Made, I., Gautama, B., Susiriyanti, P., Teknologi, I., Bisnis, D., & Bali, S. (2023). Pengumpulan Data Suhu dan Kelembapan Ruangan Menggunakan Embedded System Berbasis IoT. In *Sains Teknik Elektro* (Vol. 4, Number 2). <http://jurnal.bsi.ac.id/index.php/insantek47>
- N Somayasa, D. M., Nursalam, M. N., Taslimah, I. I., Sutiari, D. K., & Muhammad Abidin, dan Z. (2024). PROTOTYPE PENGONTROLAN NYALA DAN PADAMNYA LAMPU BERBASIS IOT (INTERNET OF THINGS). *Jurnal Nasional Hasil Penelitian Bidang Multidisiplin*, 1, 1–14. <https://doi.org/10.30598/jnbcxxxxxxxxxxxxxx>
- Nur, A., & Sari, E. (2021). SMART MONITORING SYSTEM SUHU RUANGAN SERVER BERBASIS IOT MENGGUNAKAN METODE PROTOKOL KOMUNIKASI HTTP PADA STMIK MARDIRA INDONESIA BANDUNG. In *Jurnal* (Vol. 2, Number 1).
- Ontowirjo Y.Q Fauzah, & Poekoel C. Vecky. (2018). Implementasi Internet of Things Pada Sistem Monitoring Suhu dan Kelembaban Pada Ruangan Pengering Berbasis Web. *Jurnal Teknik Elektro Dan Komputer*, 7, 331–338. www.cec-unsrat.com.
- Pramudita, R., & Setyawan, K. (2022). Sistem Smart Class Berbasis Internet Of Things Dengan Menggunakan Metode Prototype. *SMARTICS Journal*, 8(1). <https://doi.org/10.21067/smartics.v8i1.7209>
- Putri Ida Sunaryathy Samad. (2024). Rancang Bangun Alat Monitoring Ruangan Berbasis IoT Dan MQTT. *JETC, Volume 9, Nomor 1*.

- Putri, J. W. D., Firdaus, J. R., Khaerunnisa, L. S. K., & Sobur, S. (2025). SISTEM MONITORING SUHU, KELEMBAPAN DAN GAS BERBASIS IOT PADA LAB KOMPUTER. *Jurnal Informatika Dan Teknik Elektro Terapan*, 13(3S1). <https://doi.org/10.23960/jitet.v13i3S1.8175>
- Ridla, M. A., & Rahman, M. F. (2024). Perancangan Prototype Monitoring Suhu Berbasis Internet Of Things (IoT). *JUSIFOR: Jurnal Sistem Informasi Dan Informatika*, 3(1), 72–79. <https://doi.org/10.33379/jusifor.v3i1.4367>
- Riyanto, J., Nurlaila, F., Haerudin, H., & Jarastino, B. T. (2021). Rancang Bangun Sistem Monitoring Ruang Kelas Berbasis Internet of Things pada Universitas Pamulang. *Jurnal Informatika Universitas Pamulang*, 5(4), 483. <https://doi.org/10.32493/informatika.v5i4.7018>
- Samsugi, S., Mardiyansyah, Z., & Nurkholis, A. (2020). SISTEM PENGONTROL IRIGASI OTOMATIS MENGGUNAKAN MIKROKONTROLER ARDUINO UNO. In *JTST* (Vol. 01, Number 01).
- Sari, I. P., Apdilah, D., & Guntur, S. (2025). Sistem Smart Class Berbasis Internet of Things (IoT). *Sudo Jurnal Teknik Informatika*, 4(1), 33–39. <https://doi.org/10.56211/sudo.v4i1.778>
- Sholeha, D., Sri Rezeki, N., Sinaga, J., & Tarigan, K. (2025). Impression: Jurnal Teknologi dan Informasi Penerapan Media Pembelajaran Sistem Monitoring Energi Listrik Berbasis Internet of Things (IoT) Menggunakan ESP32. In *Jurnal Teknologi dan Informas* (Vol. 4, Number 3).
- Susilania. (2025). Perancangan Sistem Monitoring Berbasis Iot Untuk Peningkatan Kepatuhan Aturan Kecepatan Dalam Berkendara di Area Tambang. *Jurnal Ilmu Komputer*, 1, 145–158.
- Tiyas, A. W., Erwanto, D., & Yanuartanti, I. (2025). Peningkatan Akurasi Sensor Suhu dan Kelembaban DHT11 dengan Kalibrasi Suhu Berbasis IoT pada Platform Thingspeak. *Jurnal Pendidikan Dan Teknologi Indonesia*, 5(3), 625–633. <https://doi.org/10.52436/1.jpti.709>
- Yunardi, D. H., Misbullah, A., & Gemilang, D. G. (2023). RANCANG BANGUN SISTEM WEB MONITORING KUALITAS UDARA DI DALAM RUANGAN MENGGUNAKAN LOGIKA FUZZY. *Jurnal Pendidikan Teknologi Informasi*, 7(1), 25–34.

<https://www.theengineeringprojects.com/2018/10/introduction-to-arduino-ide.html>

<https://www.ardutech.com/mengenal-esp32-development-kit-untuk-iot-internet-of-things/>

<https://jagorobotik.com/blog/2025/06/28/pengenalan-modul-dht22-sensor-suhu-dan-kelembaban-andal-untuk-proyek-elektronikmu/>

<https://nusabot.id/blog/menggunakan-input-ldr/>

<https://www.aksesoriskomputerlampung.com/2019/07/keypad-membran-4x4-arduino.html>

<https://www.edukasielektronika.com/2020/12/motor-servo-sg90.html>

<https://h5.lazada.co.id/products/kipas-dc-12v-4x4-cm-cooling-fan-12v-dc-kipas-mini-pendingin-4-x-4cm-i829118165.html>

<https://1ohm.in/product/2-channel-5v-relay-module-for-arduino-with-optocoupler-2/>

https://soldered.com/learn/led-light-emitting-diode-explained/?srsltid=AfmBOorb_nOXm67IhLiKW8gw3m1nPeYfhdobGNMhYfBUoCPE5ccVc0

https://www.sunfounder.com/products/i2c-lcd2004-module?srsltid=AfmBOoqs1_eZLHGip0SxR9ZcwaCK3uuHkYC0fQ3thK9cuqLeqIjpExE

<https://www.codepolitan.com/blog/ingin-bikin-website-sendiri-ini-dia-rahasia-di-baliknya-ipk88n/>

**SISTEM *SMART CLASS* BERBASIS *INTERNET OF THINGS*
(*IOT*) MENGGUNAKAN *ESP32 DEVKIT V1* BERBASIS *WEB***

TUGAS AKHIR



Oleh:
Rahmat Hidayat Nasution
NIM: 2113500025

**PROGRAM STUDI SISTEM KOMPUTER
FAKULTAS TEKNOLOGI INFORMASI
UNIVERSITAS BUDI LUHUR**

**JAKARTA
2026**

**SISTEM *SMART CLASS* BERBASIS *INTERNET OF THINGS*
(*IOT*) MENGGUNAKAN *ESP32 DEVKIT V1* BERBASIS *WEB***

**Diajukan untuk memenuhi salah satu persyaratan dalam memperoleh gelar
Sarjana Komputer (S.Kom)**



**Oleh:
Rahmat Hidayat Nasution
NIM: 2113500025**

**PROGRAM STUDI SISTEM KOMPUTER
FAKULTAS TEKNOLOGI INFORMASI
UNIVERSITAS BUDI LUHUR**

**JAKARTA
2026**

PENGESAHAN TUGAS AKHIR



PROGRAM STUDI SISTEM KOMPUTER
FAKULTAS TEKNOLOGI INFORMASI
UNIVERSITAS BUDI LUHUR

LEMBAR PENGESAHAN

Nama : Rahmat Hidayat Nasution
Nomor Induk Mahasiswa : 2113500025
Program Studi : Sistem Komputer
Bidang Peminatan : Robotics
Jenjang Studi : Stratal
Judul : SISTEM SMART CLASS BERBASIS INTERNET OF THINGS
MENGUNAKAN ESP32 DEVKIT V1 BERBASIS WEB



Laporan Tugas Akhir ini telah disetujui, disahkan dan direkam secara elektronik sehingga tidak memerlukan tanda tangan tim penguji.

Jakarta, Senin 26

Januari 2026 Tim

Penguji:

Ketua : Irawan, S.Kom., M.Kom.
Anggota : Riri Irawati, S.Kom.,
M.Kom.
Pembimbing : Ir. Gatot Purwanto, M.M.
Ketua Program Studi : Rizky Pradana, S.Kom.,
M.Kom.

PERNYATAAN TIDAK PLAGIAT

SURAT PERNYATAAN TIDAK PLAGIAT DAN PERSETUJUAN PUBLIKASI

Saya yang bertanda tangan dibawah ini:

Nama : Rahmat Hidayat Nasution
NIM : 2113500025
Program Studi : Sistem Komputer
Bidang Peminatan : Robotik
Jenjang Studi : Strata 1
Fakultas : Fakultas Teknologi Informasi

Menyatakan bahwa TUGAS AKHIR YANG berjudul :

**PERANCANGAN DAN PEMBUATAN PROTOTYPE SISTEM SMART CLASS
BERBASIS INTERNET OF THINGS (IOT) MENGGUNAKAN ESP32 DEVKIT V1
BERBASIS WEB**

Merupakan:

1. Karya tulis saya sebagai laporan tugas akhir yang asli dan belum pernah diajukan untuk mendapatkan gelar akademik apapun, baik di Universitas Budi Luhur maupun di perguruan tinggi lainnya.
2. Karya tulis ini bukan saduran/terjemahan, dan murni gagasan, rumusan dan pelaksanaan penelitian/implementasi saya sendiri, tanpa bantuan pihak lain, kecuali arahan pembimbing akademik dan pembimbing di organisasi tempat riset.
3. Dalam karya tulis ini tidak terdapat karya atau pendapat yang telah ditulis atau dipublikasikan orang lain, kecuali secara tertulis dengan dicantumkan sebagai acuan dalam naskah dengan disebutkan nama pengarang dan dicantumkan dalam daftar pustaka.
4. Saya menyerahkan hak milik atas karya tulis ini kepada Universitas Budi Luhur, dan oleh karenanya Universitas Budi Luhur berhak melakukan pengelolaan atas karya tulis ini sesuai dengan norma hukum dan etika yang berlaku.

Pernyataan ini saya buat dengan sesungguhnya dan apabila di kemudian hari terdapat penyimpangan dan ketidakbenaran dalam pernyataan ini, maka saya bersedia menerima sanksi akademik berupa pencabutan gelar yang telah diperoleh berdasarkan karya tulis ini, serta sanksi lainnya sesuai dengan norma di Universitas Budi Luhur dan Undang-Undang yang berlaku.

Jakarta, 24 Januari
2026



Rahmat Hidayat Nasution

ABSTRAK

Perkembangan teknologi *Internet of Things (IoT)* memungkinkan penerapan sistem otomasi dan monitoring pada berbagai bidang, termasuk lingkungan pendidikan. Permasalahan yang sering ditemukan pada ruang kelas konvensional adalah pengelolaan fasilitas yang masih dilakukan secara manual, seperti pengendalian kipas dan sistem keamanan akses pintu, sehingga kurang efisien dan sulit dipantau secara terpusat. Penelitian ini bertujuan untuk merancang dan membangun *prototype* sistem *Smart Class* berbasis *Internet of Things* yang mampu melakukan monitoring dan pengendalian fasilitas ruang kelas secara otomatis maupun manual melalui antarmuka berbasis *web*. Sistem ini menggunakan mikrokontroler *ESP32 DevKit V1* sebagai pusat kendali yang terhubung dengan sensor *DHT22* untuk memantau suhu dan kelembapan ruangan, *keypad* sebagai sistem keamanan akses pintu, serta *Liquid Crystal Display (LCD)* sebagai media penampil informasi kondisi kelas. Sistem dilengkapi dengan aktuator berupa motor *servo* untuk mekanisme buka dan tutup pintu serta modul *relay* yang digunakan untuk mengendalikan kipas ruang kelas. Komunikasi data antara perangkat dan *server* dilakukan melalui jaringan *WiFi* menggunakan protokol *Hypertext Transfer Protocol (HTTP)* dengan metode *GET* dan *POST*, sehingga data dapat dimonitor dan perangkat dapat dikendalikan secara jarak jauh melalui *dashboard web*. Hasil pengujian menunjukkan bahwa sistem *Smart Class* yang dirancang mampu bekerja dengan baik sesuai dengan fungsinya. Sistem dapat memantau kondisi suhu dan kelembapan ruang kelas secara *real-time*, mengendalikan kipas dan akses pintu baik pada mode otomatis maupun manual, serta meningkatkan efisiensi pengelolaan ruang kelas. Dengan demikian, penelitian ini menghasilkan sebuah *prototype Smart Class* berbasis *Internet of Things* yang dapat digunakan sebagai referensi awal dalam pengembangan sistem ruang kelas cerdas di lingkungan pendidikan.

Kata kunci: *Smart Class, Internet of Things, ESP32, Otomasi Kelas, Monitoring Ruang Kelas*

KATA PENGANTAR

Puji dan syukur penulis panjatkan ke hadirat Tuhan Yang Maha Esa atas berkat dan karunia-Nya, sehingga penulis dapat menyelesaikan penyusunan skripsi ini dengan judul “**Sistem Smart Class Berbasis Internet of Things (IoT) Menggunakan ESP32 DevKit V1 Berbasis Web**”. Skripsi ini disusun sebagai salah satu syarat untuk memperoleh gelar Sarjana Komputer pada Program Studi Sistem Komputer, Fakultas Teknologi Informasi, Universitas Budi Luhur.

Dalam penyusunan skripsi ini, penulis menyadari bahwa keberhasilan tidak terlepas dari bantuan, bimbingan, dan dukungan berbagai pihak. Oleh karena itu, pada kesempatan ini penulis ingin menyampaikan penghargaan dan ucapan terima kasih yang sebesar-besarnya kepada:

1. Tuhan Yang Maha Esa atas rahmat dan karunia-Nya.
2. Orang tua dan segenap keluarga yang senantiasa memberikan doa, dukungan, dan motivasi.
3. Prof. Dr. Agus Setyo Budi, M.Sc., selaku Rektor Universitas Budi Luhur.
4. Dr. Achmad Solichin, S.Kom., M.Kom., selaku Dekan Fakultas Teknologi Informasi.
5. Rizky Pradana, S.Kom., M.Kom., selaku Ketua Program Studi Sistem Komputer.
6. Ir. Gatot Purwanto, M.M., selaku Dosen Pembimbing yang telah memberikan arahan, bimbingan, serta masukan yang sangat berharga dalam penyusunan tugas akhir ini.
7. Yani Prabowo, S.Kom, M.Si., selaku Dosen Pembimbing Akademik yang telah membimbing penulis selama masa perkuliahan.
8. Rekan-rekan mahasiswa yang telah memberikan dukungan, motivasi, serta diskusi selama proses penyusunan skripsi ini.

Penulis menyadari bahwa skripsi ini masih memiliki keterbatasan. Oleh karena itu, kritik dan saran yang membangun sangat diharapkan demi penyempurnaan karya ini. Semoga skripsi ini dapat memberikan manfaat bagi pembaca dan pengembangan ilmu pengetahuan di bidang teknologi informasi.

Jakarta, 19 Januari 2026

Rahmat Hidayat Nasution

DAFTAR TABEL




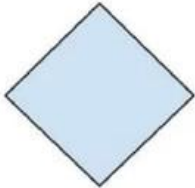
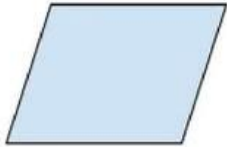
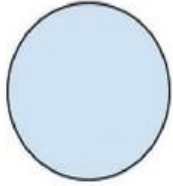
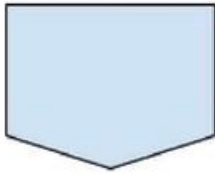

Tabel 2. 1 Studi Literatur	18
Tabel 2. 2 Spesifikasi ESP32 DevKit V1.....	8
Tabel 2. 3 Spesifikasi Sensor DHT 22	9
Tabel 2. 4 Spesifikasi Sensor LDR	10
Tabel 2. 5 Spesifikasi Keypad.....	12
Tabel 2. 6 Spesifikasi Servo.....	13
Tabel 2. 7 Spesifikasi Fan	13
Tabel 2. 8 Spesifikasi Relay Optocoupler 2 Channel	14
Tabel 2. 9 Spesifikasi LED	15
Tabel 2. 10 Spesifikasi <i>Liquid Crystal Display</i> I2C	16
Tabel 3. 1 Rangkaian Keseluruhan Alat.....	24
Tabel 3. 2 Rangkaian Keypad 4x4	25
Tabel 3. 3 Rangkaian Sensor DHT 22	26
Tabel 3. 4 Rangkaian Sensor LDR.....	27
Tabel 3. 5 Rangkaian Servo	28
Tabel 3. 6 Rangkaian LED.....	29
Tabel 3. 7 Rangkaian <i>Liquid Crystal Display</i> 20x4 I2C	30
Tabel 3. 8 Rancangan Pengujian	41
Tabel 4. 1 Pengujian <i>Keypad</i>	45
Tabel 4. 2 Pengujian Sensor DHT 22.....	46
Tabel 4. 3 Pengujian Sensor LDR.....	47
Tabel 4. 4 Pengujian Motor Servo	47
Tabel 4. 5 Pengujian Relay dan Fan	48
Tabel 4. 6 Pengujian LCD.....	48
Tabel 4. 7 Pengujian Komunikasi API (GET/POST).....	49

DAFTAR GAMBAR

Gambar 2. 1 <i>Software Arduino IDE</i>	6
Gambar 2. 2 ESP32 Devkit V1	7
Gambar 2. 3 Sensor DHT 22.....	8
Gambar 2. 4 <i>Light Dependent Resistor</i>	9
Gambar 2. 5 Keypad 4x4	11
Gambar 2. 6 Servo	12
Gambar 2. 7 Fan.....	13
Gambar 2. 8 Relay Optocoupler 2 Channel	14
Gambar 2. 9 LED.....	15
Gambar 2. 10 <i>Liquid Crystal Display I2C</i>	16
Gambar 2. 11 <i>Web Hosting</i>	17
Gambar 3. 1 Diagram Blok Alat	22
Gambar 3. 2 Rangkaian Keseluruhan Alat.....	23
Gambar 3. 3 Rangkaian Keypad 4x4	25
Gambar 3. 4 Rangkaian Sensor DHT 22	26
Gambar 3. 5 Rangkaian LDR	27
Gambar 3. 6 Rangkaian Servo	28
Gambar 3. 7 Rangkaian LED.....	29
Gambar 3. 8 Rangkaian <i>Liquid Crystal Display 20x4 I2C</i>	30
Gambar 3. 9 Rumus Fan 1	33
Gambar 3. 10 Rumus Pengendalian LED Berdasarkan Sensor LDR.....	34
Gambar 3. 11 Rumus Motor dc.....	34
Gambar 3. 12 Flowchart Sistem	35
Gambar 3. 13 Layar <i>Dashboard</i>	37
Gambar 3. 14 Layar Monitoring.....	37
Gambar 3. 15 Layar Kontrol.....	38
Gambar 3. 16 Layar Pengaturan	39
Gambar 3. 17 Tampak Depan	39
Gambar 3. 18 Tampak Belakang.....	40
Gambar 3. 19 Tampak Samping.....	40
Gambar 4. 1 Diagram Blok Cara Kerja	43
Gambar 4. 2 Tampak atas.....	52
Gambar 4. 3 Tampak samping	53
Gambar 4. 4 Tampak Depan	53

DAFTAR SIMBOL

Simbol Flowchart

No.	Simbol Flowchart	Nama	Arti Simbol Flowchart
1		<i>Terminator</i>	Awal atau akhir konsep (prosedur)
2		<i>Process</i>	Proses operasional
3		<i>Document</i>	Dokumen atau laporan berupa <i>print out</i>
4		<i>Decision</i>	Keputusan atau sub-point. Garis yang terhubung dengan bentuk <i>decision</i> merujuk pada situasi-situasi yang berbeda sesuai dengan keputusan yang digambarkan
5		Data	Input dan Output (Contohnya, Input: feedback dari pelanggan. Output: desain produk baru)
6		<i>On-Page Reference/Connector</i>	Penghubung alur dalam halaman yang sama
7		<i>Off-Page Reference/Off-Page Connector</i>	Penghubung alur dalam halaman yang berbeda
8		<i>Flow</i>	Arah alur dalam konsep (prosedur)

DAFTAR ISI

PENGESAHAN TUGAS AKHIR.....	i
PERNYATAAN TIDAK PLAGIAT	ii
ABSTRAK.....	iii
KATA PENGANTAR.....	iv
DAFTAR TABEL.....	v
DAFTAR GAMBAR.....	vi
DAFTAR SIMBOL	vii
DAFTAR ISI.....	viii
BAB I PENDAHULUAN.....	1
1.1 Latar Belakang.....	1
1.2 Rumusan Masalah.....	2
1.3 Batasan Masalah	2
1.4 Tujuan Penelitian	3
1.5 Manfaat Penelitian	3
1.6 Sistematika Penulisan	3
BAB II LANDASAN TEORI	5
2.1 Studi Literatur	18
2.2 Smart Class	5
2.3 Internet Of Things (IoT)	5
2.4 Arduino IDE.....	6
2.5 ESP32 DevKit V1	7
2.6 Sensor DHT 22	8
2.7 Sensor LDR (<i>Light Dependent Resistor</i>).....	9
2.8 Keypad 4x4	11
2.9 Servo	12
2.10 Fan	13
2.11 Relay Optocoupler 2 channel.....	14
2.12 LED.....	15
2.13 Liquid Crystal Display I2C 20x4.....	16
2.14 Web Hosting.....	17
BAB III METODOLOGI PENELITIAN	21
3.1 Data Penelitian.....	21
3.2 Rancangan Diagram Blok.....	22
3.3 Rangkaian Perangkat Keras	23
3.3.1 Rangkaian Keseluruhan	23
3.3.2 Rangkaian keypad 4x4.....	25
3.3.3 Rangkaian Sensor DHT 22	26
3.3.4 Rangkaian LDR	27
3.3.5 Rangkaian Servo	28
3.3.6 Rangkaian LED.....	29
3.3.7 Rangkaian Liquid Crystal Display 20x4 I2C.....	30
3.4 Rancangan Metodologi Pengembangan Software	30
3.4.1 Langkah Alur Metodologi.....	31

3.4.2 Formulasi Matematika	32
3.5 Flowchart	35
3.6 Rancangan Layar	36
3.6.1 Layar Dashboard.....	37
3.6.2 Layar Monitoring	37
3.6.3 Layar Kontrol.....	38
3.6.4 Layar Pengaturan	39
3.7 Rancangan Desain 3D.....	39
3.8 Rancangan Pengujian.....	40
BAB IV IMPLEMENTASI DAN EVALUASI PROGRAM	43
4.1 Cara Kerja Sistem	43
4.2 Hasil Pengujian Perangkat Keras.....	44
4.2.1 Pengujian keypad 4x4	45
4.2.2 Pengujian Sensor DHT 22	46
4.2.3 Pengujian Sensor LDR.....	46
4.2.4 Pengujian Motor Servo	47
4.2.5 Pengujian Relay dan Fan	48
4.2.6 Pengujian Liquid Crystal Display 20x4 I2C	48
4.2.7 Pengujian Komunikasi API (GET/POST)	49
4.3 Analisis Hasil Pengujian	Error! Bookmark not defined.
4.4 Kelebihan Sistem	51
4.5 Kekurangan Sistem	51
4.6 Foto Sistem	52
BAB V PENUTUP.....	54
5.1 Kesimpulan	54
5.2 Saran	55
DAFTAR PUSTAKA	56
LAMPIRAN	59

BAB I

PENDAHULUAN

1.1 Latar Belakang

Perkembangan teknologi informasi yang semakin pesat telah memberikan dampak besar terhadap berbagai bidang kehidupan, termasuk bidang pendidikan. Salah satu penerapan teknologi yang mulai banyak dikembangkan adalah konsep Smart Class, yaitu ruang kelas yang memanfaatkan teknologi untuk membantu pengelolaan fasilitas dan menciptakan lingkungan belajar yang lebih efektif dan efisien. Namun, pada kenyataannya masih banyak ruang kelas yang pengoperasian fasilitasnya dilakukan secara manual, seperti pengendalian lampu, kipas, dan akses pintu, sehingga kurang praktis dan sulit dipantau secara terpusat.

Seiring dengan berkembangnya teknologi Internet of Things (IoT), berbagai perangkat elektronik kini dapat saling terhubung dan berkomunikasi melalui jaringan internet. Teknologi ini memungkinkan proses pemantauan dan pengendalian perangkat dilakukan secara real-time melalui sistem berbasis web. Beberapa penelitian terdahulu juga telah menunjukkan pentingnya penerapan IoT dalam ruang kelas. Misalnya, penelitian mengenai sistem pemantauan ruang kelas berbasis IoT menunjukkan bahwa integrasi sensor dan antarmuka web mampu memantau kondisi lingkungan dan mengoptimalkan penggunaan energi secara otomatis dengan notifikasi apabila parameter lingkungan melebihi batas tertentu, sehingga mendukung pengelolaan fasilitas ruang kelas yang lebih efektif (Latief et al., 2025). Penelitian lain mengenai penerapan konsep Smart Class di Sekolah Tinggi Teknologi Wastukencana Purwakarta menunjukkan bahwa penggunaan IoT mampu menghubungkan berbagai perangkat elektronik ruang kelas agar dapat dikontrol melalui internet sehingga proses pembelajaran menjadi lebih nyaman dan efisien (Putri Ida Sunaryathy Samad, 2024). Selain itu, pengembangan sistem monitoring ruang kelas berbasis IoT juga telah diterapkan dengan kemampuan pengoperasian fasilitas elektronik dan pemantauan kondisi ruang secara real-time melalui antarmuka web sehingga dapat meminimalkan kerusakan fasilitas dan memaksimalkan efisiensi penggunaannya (Riyanto et al., 2021).

Berdasarkan kondisi tersebut, diperlukan sebuah sistem Smart Class yang mampu mengintegrasikan fungsi monitoring dan pengendalian fasilitas ruang kelas dalam satu kesatuan. Pada penelitian ini dirancang sebuah prototype sistem Smart Class berbasis Internet of Things (IoT) dengan menggunakan mikrokontroler ESP32 DevKit V1 sebagai pengendali utama. Sistem ini memanfaatkan perangkat input berupa sensor DHT22 untuk memantau suhu dan kelembapan ruangan serta keypad sebagai sistem pengaman akses pintu. Selain itu, sistem juga menggunakan Liquid Crystal

Display (LCD) I2C sebagai media tampilan informasi kondisi ruang kelas secara langsung.

Perangkat output pada sistem ini berupa motor servo yang digunakan untuk mekanisme buka dan tutup pintu serta modul relay yang digunakan untuk mengendalikan kipas ruang kelas. Seluruh data yang dihasilkan oleh sensor diproses oleh ESP32 dan dikirimkan ke database, kemudian ditampilkan melalui antarmuka berbasis web sehingga dapat dipantau dan dikendalikan oleh pengguna secara jarak jauh.

Sistem ini juga dirancang dengan dua mode pengoperasian, yaitu mode otomatis dan mode manual. Pada mode otomatis, sistem bekerja berdasarkan input dari sensor, sedangkan pada mode manual pengguna dapat mengendalikan perangkat secara langsung melalui web.

Melalui penelitian ini, diharapkan prototype sistem Smart Class berbasis Internet of Things (IoT) yang dirancang dapat menjadi solusi awal dalam pengelolaan fasilitas ruang kelas secara cerdas dan terintegrasi. Sistem ini diharapkan mampu meningkatkan efisiensi pengelolaan ruang kelas, memberikan kemudahan dalam pemantauan dan pengendalian perangkat, serta menjadi referensi bagi pengembangan sistem Smart Class yang lebih kompleks dan implementatif di lingkungan pendidikan pada masa mendatang.

1.2 Rumusan Masalah

Berdasarkan latar belakang yang telah dijelaskan sebelumnya, maka dapat dirumuskan beberapa permasalahan yang akan dibahas dalam penelitian ini sebagai berikut:

1. Bagaimana merancang dan membangun *prototype* sistem *Smart Class* berbasis *Internet of Things (IoT)* yang mampu memantau dan mengendalikan fasilitas ruang kelas secara terintegrasi?
2. Bagaimana pemanfaatan mikrokontroler *ESP32 DevKit V1* dalam mengolah data sensor dan mengendalikan perangkat ruang kelas melalui sistem berbasis *web*?
3. Bagaimana penerapan mode otomatis dan mode manual pada sistem *Smart Class* berbasis *IoT* dalam pengelolaan fasilitas ruang kelas?

1.3 Batasan Masalah

Agar pembahasan dalam penelitian ini lebih terfokus dan sesuai dengan kemampuan peralatan yang digunakan, maka batasan masalah dalam penelitian ini adalah sebagai berikut:

1. Sistem yang dirancang merupakan *prototype Smart Class* berbasis *IoT* dan belum diimplementasikan secara penuh pada ruang kelas sebenarnya.

2. Perangkat yang dikendalikan dan dipantau dalam sistem ini dibatasi pada beberapa fasilitas ruang kelas, seperti pemantauan kondisi lingkungan dan pengendalian perangkat elektronik melalui relay dan servo.
3. Sistem monitoring dan pengendalian dilakukan melalui antarmuka *web* dengan komunikasi data menggunakan jaringan internet.

1.4 Tujuan Penelitian

Tujuan dari penelitian ini disusun berdasarkan permasalahan yang telah dirumuskan dan kontribusi yang diharapkan dari pembuatan Tugas Akhir, yaitu sebagai berikut:

1. Merancang dan membangun *prototype* sistem *Smart Class* berbasis *Internet of Things (IoT)* menggunakan mikrokontroler *ESP32 DevKit V1*.
2. Mengimplementasikan sistem monitoring dan pengendalian fasilitas ruang kelas secara otomatis dan manual melalui antarmuka *web*.
3. Menguji kinerja *prototype* sistem *Smart Class* berbasis *IoT* dalam memantau dan mengendalikan fasilitas ruang kelas secara real-time.

1.5 Manfaat Penelitian

Manfaat dari penelitian ini diharapkan dapat dirasakan oleh beberapa pihak sebagai berikut:

1. Bagi Institusi Pendidikan, sebagai referensi awal dalam pengembangan sistem *Smart Class* untuk pengelolaan fasilitas ruang kelas yang lebih efisien dan terintegrasi.
2. Bagi Mahasiswa dan Peneliti, sebagai bahan pembelajaran dan referensi dalam pengembangan sistem berbasis *Internet of Things (IoT)* dan *web*.
3. Bagi Pengembang Sistem, sebagai acuan dalam merancang dan mengimplementasikan sistem *monitoring* dan pengendalian ruang berbasis *IoT* yang dapat dikembangkan lebih lanjut.

1.6 Sistematika Penulisan

Sistematika penulisan tugas akhir ini disusun untuk memberikan gambaran secara umum mengenai isi dari setiap bab yang terdapat dalam laporan. Adapun sistematika penulisan tugas akhir ini adalah sebagai berikut:

Bab I Pendahuluan

Bab ini berisi latar belakang penelitian, rumusan masalah, batasan masalah, tujuan penelitian, manfaat penelitian, serta sistematika penulisan yang menjelaskan susunan penulisan laporan tugas akhir secara keseluruhan.

Bab II Tinjauan Pustaka	Bab ini membahas teori-teori pendukung yang berkaitan dengan penelitian, meliputi konsep Internet of Things (IoT), mikrokontroler ESP32, sensor dan aktuator yang digunakan, serta penelitian terdahulu yang relevan sebagai dasar perancangan sistem.
Bab III Metodologi Penelitian	Bab ini menjelaskan tahapan dan metode penelitian yang digunakan dalam perancangan dan pembuatan prototype Sistem Smart Class berbasis Internet of Things. Pembahasan meliputi perancangan sistem, diagram blok, perancangan perangkat keras dan perangkat lunak, rancangan pengujian, serta rancangan layar sistem.
Bab IV Implementasi dan Evaluasi Program	Bab ini membahas hasil implementasi sistem serta evaluasi kinerja berdasarkan pengujian yang telah dilakukan. Pembahasan meliputi cara kerja sistem, hasil pengujian setiap komponen, analisis hasil pengujian, kelebihan dan kekurangan sistem, serta dokumentasi foto sistem.
Bab V Penutup	Bab ini berisi kesimpulan yang diperoleh dari hasil penelitian dan pembahasan yang telah dilakukan, serta saran yang dapat digunakan sebagai pengembangan sistem pada penelitian selanjutnya.

BAB II

LANDASAN TEORI

2.1 Smart Class

Smart Class merupakan konsep ruang kelas yang memanfaatkan teknologi informasi dan komunikasi untuk mendukung proses pembelajaran yang lebih efektif, efisien, dan terintegrasi. Konsep ini bertujuan untuk mengoptimalkan pengelolaan fasilitas ruang kelas serta menciptakan lingkungan belajar yang nyaman melalui pemanfaatan sistem otomatis dan digital.

Menurut beberapa penelitian, Smart Class tidak hanya berfokus pada media pembelajaran digital, tetapi juga pada pengelolaan fasilitas ruang kelas seperti pencahayaan, sirkulasi udara, keamanan, dan pemantauan kondisi lingkungan. Penerapan teknologi Internet of Things pada Smart Class memungkinkan perangkat-perangkat tersebut saling terhubung dan dapat dilantau serta dikendalikan secara real-time.

Penelitian yang dilakukan oleh (Latief et al., 2025) menyatakan bahwa penerapan Smart Class berbasis IoT mampu meningkatkan efisiensi pengelolaan ruang kelas serta memberikan kemudahan bagi pengguna dalam memantau kondisi ruang melalui sistem berbasis web. Dengan adanya sistem ini, proses pengambilan keputusan terkait pengelolaan ruang kelas dapat dilakukan secara lebih cepat dan tepat.

Dengan demikian, Smart Class dapat dipahami sebagai sistem ruang kelas cerdas yang mengintegrasikan teknologi IoT, sistem informasi, dan perangkat elektronik untuk mendukung aktivitas pembelajaran dan pengelolaan fasilitas ruang kelas secara terpusat.

2.2 Internet Of Things (IoT)

Internet of Things (IoT) merupakan konsep teknologi yang memungkinkan berbagai perangkat fisik untuk saling terhubung dan bertukar data melalui jaringan internet. Perangkat-perangkat tersebut dapat berupa sensor, aktuator, dan sistem kendali yang bekerja secara otomatis maupun terintegrasi dengan sistem informasi berbasis web.

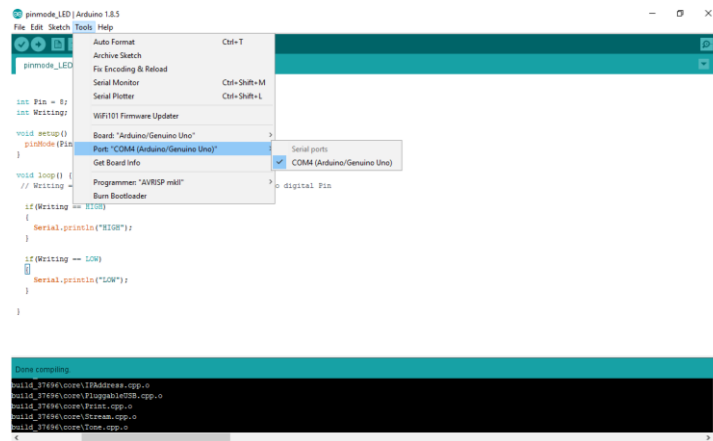
IoT bekerja dengan menggabungkan perangkat keras, jaringan komunikasi, dan aplikasi perangkat lunak. Data yang diperoleh dari sensor akan dikirimkan ke server atau database untuk diproses dan ditampilkan kepada pengguna melalui antarmuka tertentu. Konsep ini banyak diterapkan pada sistem monitoring ruangan, smart building, dan Smart Classroom.

Penelitian oleh (Riyanto et al., 2021) menjelaskan bahwa penerapan IoT dalam sistem monitoring ruangan memungkinkan pengumpulan data secara

real-time serta memberikan kemudahan dalam pengendalian perangkat dari jarak jauh. Hal ini menjadikan IoT sebagai teknologi yang tepat untuk diterapkan pada sistem Smart Class yang membutuhkan monitoring dan kontrol fasilitas ruang kelas secara terintegrasi.

Dengan penerapan IoT, sistem Smart Class dapat bekerja secara otomatis berdasarkan data sensor maupun dikendalikan secara manual melalui sistem berbasis web, sehingga meningkatkan fleksibilitas dan efisiensi pengelolaan ruang kelas.

2.3 Arduino IDE



Gambar 2. 1 Software Arduino IDE

<https://www.theengineeringprojects.com/2018/10/introduction-to-arduino-ide.html>

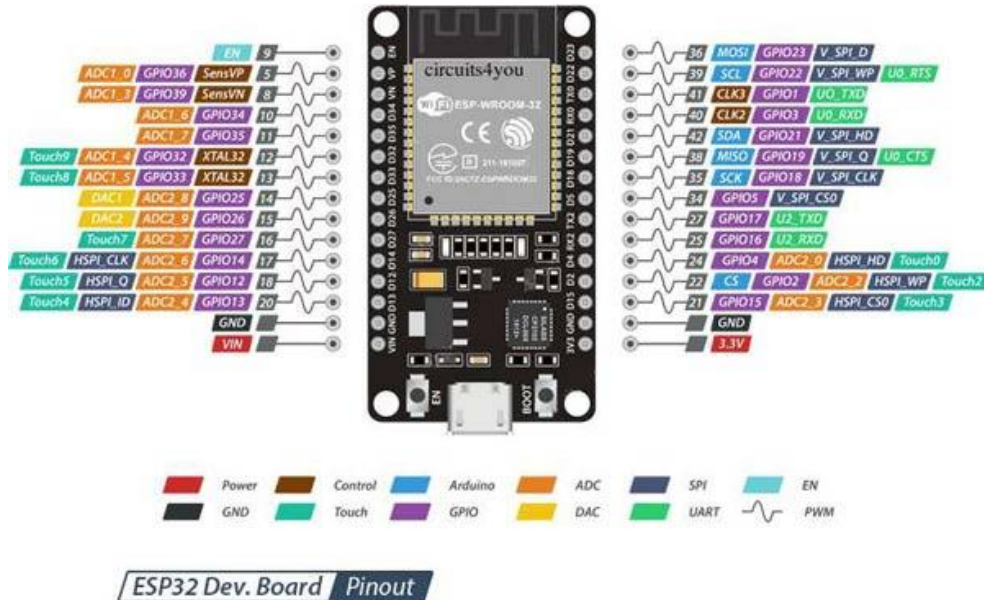
Arduino IDE (*Integrated Development Environment*) merupakan perangkat lunak yang digunakan untuk menulis, mengompilasi, dan mengunggah program ke papan mikrokontroler berbasis Arduino maupun papan pengembangan lain yang kompatibel, termasuk ESP32 DevKit V1. Arduino IDE menyediakan antarmuka yang sederhana dan mudah digunakan, sehingga banyak dimanfaatkan dalam pengembangan sistem Internet of Things (IoT) baik untuk keperluan pembelajaran maupun penelitian.

Arduino IDE menggunakan bahasa pemrograman berbasis C/C++ yang telah disederhanakan melalui berbagai pustaka (*library*) bawaan maupun tambahan. Di dalam Arduino IDE tersedia fitur text editor untuk penulisan kode program, compiler untuk menerjemahkan kode ke dalam bahasa mesin, serta serial monitor yang digunakan untuk melihat hasil komunikasi data antara mikrokontroler dan komputer. Fitur-fitur tersebut memudahkan pengguna dalam melakukan pengujian dan pemantauan data yang dihasilkan oleh sensor secara real-time.

Arduino IDE juga mendukung pengembangan sistem berbasis ESP32 melalui penambahan board manager yang disediakan oleh Espressif Systems. Dengan dukungan tersebut, Arduino IDE dapat digunakan untuk mengelola pin

input dan output, membaca data sensor, mengendalikan aktuator, serta mengatur komunikasi data melalui jaringan Wi-Fi. Kemampuan ini menjadikan Arduino IDE sebagai perangkat lunak utama dalam pengembangan prototype Smart Class berbasis IoT karena mendukung proses pemrograman, pengujian, dan integrasi sistem secara menyeluruh.

2.4 ESP32 DevKit V1



Gambar 2. 2 ESP32 Devkit V1

<https://www.ardutech.com/mengenal-esp32-development-kit-untuk-iot-internet-of-things/>

ESP32 DevKit V1 merupakan papan pengembangan mikrokontroler yang menggunakan chip ESP32 buatan *Espressif Systems*. Papan ini dirancang untuk mendukung pengembangan sistem berbasis *Internet of Things (IoT)* karena telah dilengkapi dengan modul Wi-Fi dan Bluetooth yang terintegrasi di dalam satu chip. Dengan adanya fitur tersebut, ESP32 DevKit V1 mampu melakukan komunikasi data secara nirkabel tanpa memerlukan modul tambahan.

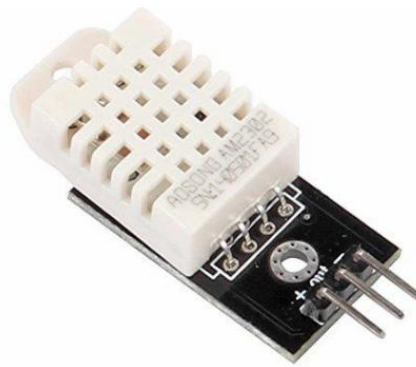
ESP32 DevKit V1 memiliki prosesor dual-core Tensilica LX6 dengan kecepatan hingga 240 MHz, sehingga mampu mengolah data dari sensor dan mengendalikan aktuator secara cepat dan stabil. Papan ini juga menyediakan sejumlah pin input dan output digital maupun analog yang dapat digunakan untuk menghubungkan berbagai perangkat eksternal seperti sensor suhu, sensor gas, relay, servo motor, dan perangkat lainnya. Selain itu, ESP32 mendukung berbagai antarmuka komunikasi, antara lain UART, SPI, I2C, PWM, dan ADC, yang memberikan fleksibilitas tinggi dalam perancangan sistem.

Untuk catu daya, ESP32 DevKit V1 dapat dioperasikan menggunakan sumber tegangan 5 V melalui koneksi USB atau tegangan 3,3 V melalui pin daya yang tersedia. Konsumsi daya yang relatif rendah menjadikan papan ini cocok digunakan pada sistem monitoring dan kontrol yang bekerja secara terus-menerus. Dengan spesifikasi dan fitur yang dimilikinya, *ESP32 DevKit V1* sangat sesuai digunakan sebagai pengendali utama dalam perancangan *prototype Smart Class* berbasis IoT yang membutuhkan fungsi monitoring, kontrol perangkat, serta integrasi dengan sistem berbasis *web* (*Espressif Systems, 2023*).

Tabel 2. 1 Spesifikasi ESP32 DevKit V1

No	Spesifikasi	Keterangan
1	Chip Mikrokontroler	ESP32
2	Prosesor	Dual-Core Tensilica LX6
3	Kecepatan Clock	Hingga 240 MHz
4	Tegangan Operasi	3,3 V
5	Tegangan Input	5 V (USB)
6	Memori Flash	4 MB
7	RAM	±520 KB
8	Konektivitas	Wi-Fi 802.11 b/g/n, Bluetooth
9	Pin GPIO	±30 pin
10	Antarmuka	UART, SPI, I2C, PWM, ADC

2.5 Sensor DHT 22



Gambar 2. 3 Sensor DHT 22

<https://jagorobotik.com/blog/2025/06/28/pengenalan-modul-dht22-sensor-suhu-dan-kelembaban-andal-untuk-proyek-elektronikmu/>

Sensor DHT22 merupakan sensor digital yang digunakan untuk mengukur suhu dan kelembaban udara dalam suatu ruangan. Sensor ini dirancang untuk memberikan hasil pengukuran yang cukup akurat dan stabil, sehingga banyak digunakan pada sistem monitoring lingkungan berbasis Internet of Things. DHT22 mengirimkan data hasil pengukuran dalam bentuk

sinyal digital melalui satu jalur komunikasi, sehingga memudahkan proses pembacaan data oleh mikrokontroler seperti ESP32.

Sensor DHT22 mampu melakukan pengukuran suhu dan kelembaban secara periodik dan menghasilkan data yang dapat digunakan untuk memantau kondisi lingkungan ruang kelas. Informasi suhu dan kelembaban tersebut sangat penting dalam menjaga kenyamanan ruang belajar serta mendukung pengambilan keputusan dalam pengendalian perangkat seperti kipas atau ventilasi. Sensor suhu dan kelembaban banyak diterapkan pada sistem monitoring ruangan berbasis IoT karena mampu memberikan data kondisi lingkungan secara real-time dan mudah diintegrasikan dengan sistem berbasis *web* (Ridla & Rahman, 2024).

Tabel 2. 2 Spesifikasi Sensor DHT 22

No	Spesifikasi	Keterangan
1	Jenis Sensor	Digital Temperature & Humidity
2	Tegangan Operasi	3,3 – 6 V
3	Arus Operasi	±2,5 mA
4	Rentang Suhu	-40°C sampai 80°C
5	Akurasi Suhu	±0,5°C
6	Rentang Kelembaban	0 – 100% RH
7	Akurasi Kelembaban	±2% RH
8	Resolusi	0,1°C / 0,1% RH
9	Antarmuka	Digital (Single Wire)

2.6 Sensor LDR (*Light Dependent Resistor*)



Gambar 2. 4 Light Dependent Resistor

<https://nusabot.id/blog/menggunakan-input-ldr/>

Sensor LDR (*Light Dependent Resistor*) merupakan salah satu jenis sensor cahaya yang nilai resistansinya berubah sesuai dengan intensitas cahaya yang diterimanya. Sensor ini bekerja berdasarkan prinsip fotokonduktivitas, di mana resistansi LDR akan menurun ketika intensitas cahaya meningkat dan akan meningkat ketika kondisi cahaya berkurang atau gelap.

LDR umumnya terbuat dari bahan semikonduktor seperti cadmium sulfide (CdS) yang memiliki karakteristik sensitif terhadap cahaya tampak. Ketika cahaya mengenai permukaan LDR, elektron akan berpindah sehingga hambatan listrik menjadi lebih kecil. Sebaliknya, pada kondisi gelap, hambatan LDR menjadi besar karena minimnya energi cahaya yang diterima.

Dalam sistem berbasis Internet of Things (IoT), sensor LDR banyak digunakan sebagai perangkat input untuk mendeteksi kondisi pencahayaan lingkungan secara otomatis. Sensor ini sering diaplikasikan pada sistem smart lighting, monitoring intensitas cahaya ruangan, serta pengendalian lampu otomatis berdasarkan kondisi terang dan gelap.

Pada sistem *Smart Class* yang dikembangkan dalam penelitian ini, sensor LDR digunakan sebagai perangkat input untuk mendeteksi intensitas cahaya di dalam ruang kelas. Data intensitas cahaya yang diperoleh dari sensor LDR diproses oleh mikrokontroler *ESP32 DevKit V1* sebagai dasar pengendalian *LED* putih yang berfungsi sebagai lampu atau indikator ruang kelas.

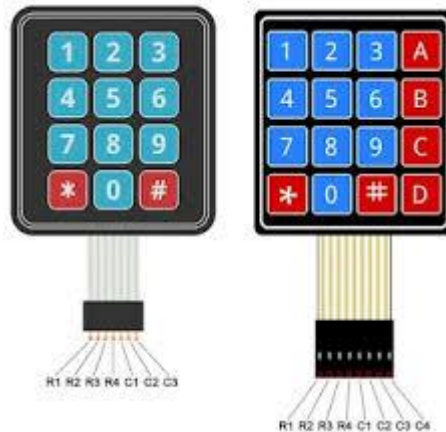
Pada kondisi terang (siang hari), sistem akan mengaktifkan *LED*, sedangkan pada kondisi gelap (malam hari), sistem akan mematikan *LED*. Penerapan sensor LDR ini bertujuan untuk mendukung otomatisasi pencahayaan ruang kelas serta meningkatkan efisiensi penggunaan energi listrik.

Penggunaan sensor LDR sebagai pendeteksi cahaya dinilai efektif karena memiliki rangkaian yang sederhana, biaya implementasi yang rendah, serta mudah diintegrasikan dengan mikrokontroler ESP32 dan sistem monitoring berbasis web (Suryadi & Fadli, 2021).

Tabel 2. 3 Spesifikasi Sensor LDR

No	Parameter	Keterangan
1	Jenis Sensor	Light Dependent Resistor (LDR)
2	Prinsip Kerja	Perubahan resistansi terhadap cahaya
3	Tegangan Operasi	3,3 – 5 V
4	Output	Analog
5	Rentang Sensitivitas	Cahaya rendah hingga tinggi
6	Antarmuka	Analog (ADC ESP32)
7	Aplikasi	Smart lighting, monitoring cahaya
8	Fungsi pada Sistem	Deteksi intensitas cahaya ruangan
9	Mode Operasi	Otomatis & Manual
10	Integrasi	ESP32 DevKit V1

2.7 Keypad 4x4



Gambar 2. 5 Keypad 4x4

<https://www.aksesoriskomputerlampung.com/2019/07/keypad-membran-4x4-arduino.html>

Keypad 4×4 merupakan perangkat input yang terdiri dari 16 tombol yang disusun dalam matriks 4 baris dan 4 kolom. Setiap tombol mewakili kombinasi tertentu antara baris dan kolom, sehingga memungkinkan sistem membaca banyak input dengan jumlah pin input/output yang lebih efisien. Keypad jenis ini umum digunakan pada sistem tertanam karena sederhana, mudah diimplementasikan, serta memiliki tingkat keandalan yang baik sebagai media masukan data.

Prinsip kerja keypad 4×4 dilakukan dengan metode pemindaian (scanning) baris dan kolom oleh mikrokontroler. Ketika sebuah tombol ditekan, akan terbentuk koneksi antara satu baris dan satu kolom tertentu yang kemudian dikenali sebagai input oleh sistem. Metode ini memungkinkan identifikasi tombol secara akurat tanpa memerlukan rangkaian tambahan yang kompleks.

Dalam sistem Smart Class yang dirancang, keypad 4×4 digunakan sebagai sistem autentikasi untuk pengamanan akses pintu ruang kelas. Pengguna diwajibkan memasukkan kode akses tertentu sebelum sistem memberikan perintah kepada aktuator untuk membuka pintu. Penerapan keypad sebagai kontrol akses bertujuan untuk meningkatkan keamanan ruang kelas dan membatasi akses hanya kepada pengguna yang berwenang.

Penggunaan keypad sebagai sistem pengaman akses telah banyak diterapkan pada sistem keamanan ruangan berbasis mikrokontroler dan Internet of Things. Keypad dinilai efektif sebagai media autentikasi karena mudah diintegrasikan dengan sistem kontrol serta dapat dikombinasikan dengan pengelolaan akses berbasis web (Fauziman & Mukhaiyar, 2023).

Tabel 2. 4 Spesifikasi Keypad

No	Parameter	Keterangan
1	Jenis Perangkat	Keypad matriks
2	Jumlah Tombol	16 tombol
3	Konfigurasi	4 baris × 4 kolom
4	Tegangan Operasi	3,3 – 5 V
5	Metode Pembacaan	Scanning baris dan kolom
6	Antarmuka	Digital
7	Jumlah Pin	8 pin (4 baris, 4 kolom)
8	Material Tombol	Membran / tombol tekan
9	Fungsi Utama	Input kode atau perintah
10	Aplikasi Umum	Sistem keamanan dan kontrol akses

2.8 Servo



Gambar 2. 6 Servo

<https://www.edukasielatronika.com/2020/12/motor-servo-sg90.html>

Motor servo merupakan aktuator yang digunakan untuk menghasilkan gerakan mekanik dengan sudut tertentu secara presisi. Motor servo banyak digunakan pada sistem otomasi dan Internet of Things karena mampu mengontrol posisi secara akurat berdasarkan sinyal kendali dari mikrokontroler. Penerapan motor servo pada sistem otomasi pintu dan mekanisme gerak telah banyak digunakan pada sistem berbasis mikrokontroler dan IoT (Samsugi et al., 2020).

Motor servo pada sistem ini dikendalikan menggunakan sinyal Pulse Width Modulation (PWM) dari ESP32 untuk menentukan sudut putaran poros. Dengan kemampuan pengaturan sudut yang presisi, motor servo digunakan untuk membuka dan menutup pintu atau ventilasi secara otomatis maupun manual melalui sistem berbasis web.

Tabel 2. 5 Spesifikasi Servo

Parameter	Keterangan
Tegangan Kerja	4,8 – 6 V
Sudut Putar	0° – 180°
Sinyal Kendali	PWM
Tipe Gerak	Presisi sudut
Aplikasi	Otomasi pintu & mekanisme

2.9 Fan



Gambar 2. 7 Fan

<https://h5.lazada.co.id/products/kipas-dc-12v-4x4-cm-cooling-fan-12v-dc-kipas-mini-pendingin-4-x-4cm-i829118165.html>

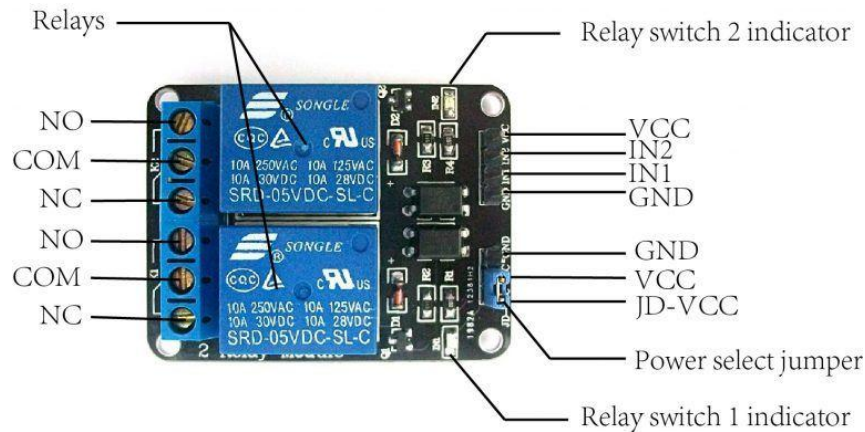
Fan atau kipas angin merupakan perangkat output yang digunakan untuk membantu sirkulasi udara dan menjaga kenyamanan suhu ruangan. Penggunaan kipas angin pada sistem monitoring lingkungan berbasis IoT umum diterapkan untuk meningkatkan kualitas udara dan kenyamanan ruang (Budiyanto et al., 2020).

Pada sistem Smart Class, fan dikendalikan melalui relay berdasarkan data suhu dan kualitas udara dari sensor. Fan dapat bekerja secara otomatis maupun manual melalui antarmuka web untuk menjaga kondisi ruang kelas tetap nyaman.

Tabel 2. 6 Spesifikasi Fan

Parameter	Keterangan
Tegangan Kerja	12 DC
Kendali	Relay
Fungsi	Sirkulasi udara
Mode Operasi	Otomatis & Manual
Aplikasi	Ruang kelas

2.10 Relay Optocoupler 2 channel



Gambar 2. 8 Relay Optocoupler 2 Channel

<https://1ohm.in/product/2-channel-5v-relay-module-for-arduino-with-optocoupler-2/>

Relay optocoupler 2 channel merupakan modul saklar elektronik yang dilengkapi dua kanal dan isolasi optocoupler untuk memisahkan rangkaian kontrol dan rangkaian daya. Relay jenis ini banyak digunakan pada sistem otomasi berbasis IoT untuk meningkatkan keamanan pengendalian perangkat listrik (N Somayasa et al., 2024).

Relay ini memungkinkan mikrokontroler mengendalikan dua perangkat listrik secara terpisah dengan tegangan kontrol rendah. Isolasi optocoupler berfungsi melindungi ESP32 dari gangguan listrik yang berasal dari beban.

Tabel 2. 7 Spesifikasi Relay Optocoupler 2 Channel

Parameter	Keterangan
Jumlah Kanal	2 Channel
Tegangan Kontrol	3,3 – 5 V
Tegangan Beban	Hingga 250 VAC
Arus Maksimum	10 A
Isolasi	Optocoupler

2.11 LED



Gambar 2. 9 LED

https://soldered.com/learn/led-light-emitting-diode-explained/?srsltid=AfmBOorb_nOXm67IhLiKW8gw3m1nPeYfhdobGNMhYfBUoCPE5ccVc0

Light Emitting Diode (LED) merupakan perangkat output yang berfungsi sebagai indikator visual pada suatu sistem. LED banyak digunakan pada sistem tertanam dan IoT karena konsumsi daya yang rendah dan respon yang cepat (Kalbuana & Kurnianto, 2024).

Pada sistem Smart Class, LED digunakan sebagai indikator status perangkat yang dapat dikendalikan melalui sistem berbasis web. LED memberikan informasi visual terkait kondisi aktif atau nonaktif suatu perangkat.

Tabel 2. 8 Spesifikasi LED

Parameter	Keterangan
Jumlah Kanal	2 Channel
Tegangan Kontrol	3,3 – 5 V
Tegangan Beban	Hingga 250 VAC
Arus Maksimum	10 A
Isolasi	Optocoupler

2.12 Liquid Crystal Display I2C 20x4



Gambar 2. 10 Liquid Crystal Display I2C

https://www.sunfounder.com/products/i2c-lcd2004-module?srsId=AfmBOoqs1_eZLHGj9p0SxR9ZcwaCK3uuHkYC0fQ3thK9cuqLeqIjExE

Liquid Crystal Display (LCD) I2C merupakan perangkat output yang digunakan untuk menampilkan informasi dalam bentuk teks dengan antarmuka komunikasi I2C. LCD I2C banyak digunakan pada sistem IoT karena efisiensi penggunaan pin dan kemudahan integrasi (Susilania, 2025).

LCD I2C digunakan untuk menampilkan informasi kondisi ruang kelas secara langsung tanpa perlu mengakses web, seperti suhu ruangan, kualitas udara, dan status perangkat.

Tabel 2. 9 Spesifikasi Liquid Crystal Display I2C

Parameter	Keterangan
Ukuran	20 × 4
Antarmuka	I2C
Tegangan Kerja	5 V
Fungsi	Tampilan informasi
Jumlah Pin	4 pin

2.13 Web Hosting



Gambar 2. 11 Web Hosting

<https://www.codepolitan.com/blog/ingin-bikin-website-sendiri-ini-dia-rahasia-di-baliknya-ipk88n/>

Web hosting merupakan layanan yang menyediakan sumber daya server untuk menyimpan, mengelola, dan menjalankan aplikasi berbasis web agar dapat diakses melalui jaringan internet. Web hosting bekerja bersama dengan domain, di mana domain berfungsi sebagai alamat unik yang digunakan pengguna untuk mengakses sistem. Dalam konteks sistem berbasis Internet of Things, web hosting berperan sebagai pusat pengelolaan data dan antarmuka pengguna yang menghubungkan perangkat IoT dengan pengguna akhir melalui jaringan internet. Penggunaan web hosting dan domain memungkinkan sistem monitoring dan kontrol dapat diakses secara jarak jauh tanpa keterbatasan lokasi (Ontowirjo Y.Q Fauzah & Poekoel C. Vecky, 2018)

Pada sistem Smart Class yang dirancang, web hosting digunakan untuk menampung aplikasi web yang berfungsi menampilkan data hasil pembacaan sensor, menyimpan data ke dalam database, serta menyediakan fitur kontrol perangkat. Melalui domain yang terhubung dengan web hosting, pengguna dapat mengakses sistem untuk memantau kondisi ruang kelas secara real-time, mengatur mode pengoperasian sistem, serta mengendalikan perangkat output seperti lampu, kipas, dan pintu. Implementasi web hosting sebagai media monitoring dan kontrol pada sistem IoT menjadikan sistem lebih terpusat, mudah diakses, dan fleksibel untuk dikembangkan sesuai kebutuhan di masa mendatang.

2.14 Studi Literatur

Melakukan studi literatur merupakan proses mengumpulkan dan mempelajari referensi ilmiah yang relevan dengan topik penelitian. Studi literatur dilakukan untuk memperoleh pemahaman mendalam mengenai konsep, teori, maupun penelitian terdahulu yang berkaitan dengan sistem *Smart Class* berbasis *Internet of Things*. Referensi yang digunakan pada penelitian ini merupakan jurnal ilmiah terbit dalam lima tahun terakhir dan memiliki keterkaitan dengan perancangan sistem monitoring, kontrol perangkat, sensor, serta aplikasi web berbasis IoT. Adapun tabel studi literatur yang digunakan adalah sebagai berikut:

Tabel 2. 10 Studi Literatur

No	Kolom	Isi
1	Judul	Penerapan Teknologi Internet of Things (IoT) untuk Monitoring Kualitas Udara dalam Ruangan
	Penulis dan tahun terbit	(Aswaldi, 2025)
	Penerbit / Jurnal	Journal of Computer Science and Information Technology
	Deskripsi	Penelitian merancang sistem IoT untuk memonitor kualitas udara dalam ruangan secara real-time, termasuk suhu, kelembapan, CO ₂ , dan partikel debu (PM2.5) yang dikirim dan ditampilkan melalui platform IoT.
2	Judul	Desain dan Implementasi Smart Classroom Berbasis ESP32 dengan Monitoring Berbasis Web
	Penulis dan tahun terbit	(Sholeha et al., 2025)
	Penerbit / Jurnal	Impression: Jurnal Teknologi dan Informasi
	Deskripsi	Penelitian ini merancang dan mengimplementasikan media pembelajaran dan monitoring IoT menggunakan mikrokontroler ESP32 sebagai basis sistem yang terhubung ke internet. Sistem ini menggunakan ESP32 untuk mengambil data sensor, mengirimkannya secara real-time ke server web, dan mengendalikan fungsi perangkat melalui tampilan web sehingga pengguna (mis. guru atau admin) dapat memantau serta mengatur perangkat jarak jauh lewat web browser. Hasil uji validasi menunjukkan sistem ini layak dan efektif dalam pembelajaran interaktif serta monitoring berbasis web, memperlihatkan bahwa ESP32 DevKit V1 mampu mengelola sensor dan antarmuka web secara real-time

3	Judul	Peningkatan Akurasi Sensor Suhu dan Kelembaban DHT11 dengan Kalibrasi Suhu Berbasis IoT pada Platform Thingspeak
	Penulis dan tahun terbit	(Tiyas et al., 2025)
	Penerbit / Jurnal	Jurnal Pendidikan dan Teknologi Indonesia
	Deskripsi	Fokus pada peningkatan akurasi pembacaan sensor suhu dan kelembapan DHT11 pada sistem IoT berbasis platform Thingspeak, meningkatkan keandalan pengukuran lingkungan.
4	Judul	Sistem Smart Class Berbasis Internet Of Things dengan Metode Prototype
	Penulis dan tahun terbit	(Pramudita & Setyawan, 2022)
	Penerbit / Jurnal	SMARTICS Journal
	Deskripsi	Penelitian ini membahas perancangan prototype Smart Class berbasis IoT yang memungkinkan monitoring dan kontrol perangkat elektronik pada ruang kelas (lampu, AC, proyektor) secara otomatis menggunakan NodeMCU dan aplikasi Blynk sebagai antarmuka web/mobile.
5	Judul	Pengumpulan Data Suhu dan Kelembapan Ruangan Menggunakan Embedded System Berbasis IoT
	Penulis dan tahun terbit	(Made et al., 2023)
	Penerbit / Jurnal	INSANtek
	Deskripsi	Sistem IoT yang dibuat untuk mengumpulkan data suhu & kelembapan ruangan secara otomatis dan konsisten, mendukung pengambilan keputusan pada sistem pendingin ruangan.
6	Judul	Rancang Bangun Sistem Monitoring Lingkungan dengan Sistem IoT dan Logika Fuzzy
	Penulis dan tahun terbit	(Yunardi et al., 2023)
	Penerbit / Jurnal	Cyberspace: Jurnal Pendidikan Teknologi Informasi
	Deskripsi	Merancang sistem web monitoring kualitas udara dalam ruangan dengan metode logika fuzzy untuk interpretasi data suhu, kelembapan, dan VOC real-time dari sensor.

7	Judul	Sistem Smart Class Berbasis Internet of Things (IoT)
	Penulis dan tahun terbit	(Sari et al., 2025)
	Penerbit / Jurnal	Sudo Jurnal Teknik Informatika
	Deskripsi	Penelitian ini merancang dan mengimplementasikan sistem <i>Smart Class</i> berbasis IoT untuk meningkatkan efisiensi absensi dan keamanan di ruang kelas melalui teknologi IoT yang terhubung dengan platform Blynk.cloud dan Telegram untuk monitoring real-time.
8	Judul	Rancang Bangun Sistem Monitoring Kualitas Udara di Laboratorium / Ruang Komputer
	Penulis dan tahun terbit	(Putri et al., 2025)
	Penerbit / Jurnal	Jurnal Informatika dan Teknik Elektro Terapan
	Deskripsi	Sistem IoT untuk monitoring suhu, kelembapan, dan kualitas udara di ruang lab komputer dengan NodeMCU, data dikirim ke aplikasi Blynk & Telegram untuk notifikasi kondisi abnormal.
9	Judul	Sistem Monitoring Suhu Ruangan Server Berbasis IoT (HTTP)
	Penulis dan tahun terbit	(Nur & Sari, 2021)
	Penerbit / Jurnal	Infosecure
	Deskripsi	Monitoring suhu & kelembapan ruang server menggunakan IoT dengan protokol HTTP, menampilkan data real-time di web & notifikasi otomatis.
10	Judul	Sistem Monitoring Kualitas Udara dan Pengaturan Suhu Berbasis IoT
	Penulis dan tahun terbit	(Irdyanti et al., 2025)
	Penerbit / Jurnal	Jurnal Ampere
	Deskripsi	Meneliti sistem monitoring kualitas udara berbasis IoT yang mampu mendeteksi suhu, kelembapan, CO, serta rekomendasi pengaturan otomatis untuk mempertahankan kondisi ideal.

BAB III METODOLOGI PENELITIAN

Dalam merancang prototype pada tugas akhir ini, diperlukan metode penelitian yang terstruktur agar proses pengembangan sistem *Smart Class* berbasis *Internet of Things* dapat dilakukan secara sistematis dan sesuai dengan tujuan penelitian. Perancangan dilakukan melalui beberapa tahapan, mulai dari pengumpulan data, analisis kebutuhan komponen, perancangan rangkaian perangkat keras, perancangan perangkat lunak pada mikrokontroler, hingga integrasi dengan aplikasi web sebagai media monitoring dan kontrol. Metodologi ini disusun berdasarkan teori yang telah dibahas pada Bab II dan disesuaikan dengan fungsi setiap komponen yang digunakan, sehingga prototype yang dihasilkan mampu bekerja secara optimal, terintegrasi, dan mendukung implementasi sistem *Smart Class* yang efektif.

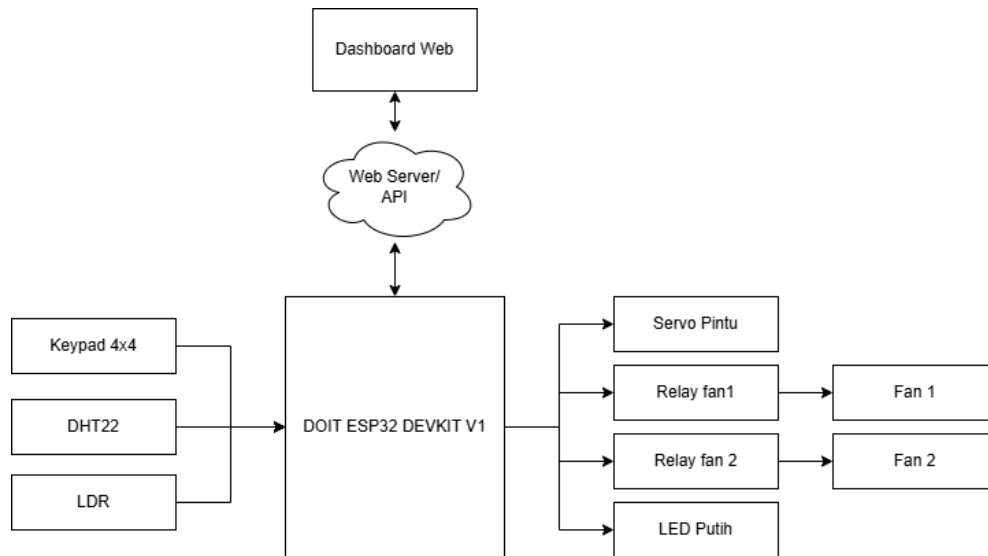
3.1 Data Penelitian

Data penelitian pada tugas akhir ini diperoleh melalui proses pengumpulan dan pengujian data menggunakan perangkat input yang terintegrasi pada sistem *Smart Class* berbasis *Internet of Things (IoT)*. Data utama yang digunakan meliputi kondisi lingkungan ruang kelas berupa suhu, kelembapan udara, serta intensitas cahaya ruangan. Data suhu dan kelembapan diperoleh dari sensor *DHT22* yang digunakan untuk memantau kondisi lingkungan ruang kelas secara berkala, sedangkan data intensitas cahaya diperoleh dari sensor *Light Dependent Resistor (LDR)*.

Selain data lingkungan, data terkait autentikasi akses pintu diperoleh melalui *keypad 4×4* sebagai perangkat input yang digunakan oleh pengguna. Seluruh data yang diperoleh dari sensor dan perangkat input tersebut diolah oleh mikrokontroler *ESP32 DevKit V1* sebagai dasar dalam pengambilan keputusan sistem, baik pada *mode otomatis* maupun *mode manual*.

Data hasil pembacaan sensor serta status perangkat kemudian dikirimkan ke server melalui koneksi *WiFi* untuk disimpan ke dalam *database* dan ditampilkan pada antarmuka berbasis web secara *real-time*. Dengan demikian, data penelitian yang digunakan tidak hanya mendukung proses monitoring kondisi lingkungan ruang kelas, tetapi juga berperan penting dalam pengembangan, pengujian, dan evaluasi fungsionalitas sistem *Smart Class* yang dirancang pada tugas akhir ini.

3.2 Rancangan Diagram Blok Rangkaian



Gambar 3. 1 Diagram Blok Alat

Gambar 3.1 merupakan diagram blok dari sistem *Smart Class* berbasis *Internet of Things (IoT)* yang dirancang pada tugas akhir ini. Diagram ini menunjukkan alur kerja sistem secara keseluruhan, mulai dari perangkat input, mikrokontroler sebagai pusat kendali, hingga perangkat output serta integrasi dengan *web server*.

Pada bagian kiri diagram terdapat perangkat input yang digunakan dalam sistem, yaitu *keypad 4×4* sebagai alat autentikasi akses pintu, sensor *DHT22* untuk membaca suhu dan kelembapan ruang kelas, serta sensor *Light Dependent Resistor (LDR)* untuk mendeteksi intensitas cahaya ruangan. Data yang diperoleh dari perangkat input tersebut dikirimkan ke mikrokontroler untuk diproses sesuai dengan logika sistem yang telah dirancang.

Pada bagian tengah diagram terdapat modul *DOIT ESP32 DevKit V1* yang berperan sebagai pusat pengolahan data dan pengendalian sistem. Mikrokontroler *ESP32* menerima data dari sensor dan *keypad*, menjalankan logika sistem baik pada *mode otomatis* maupun *mode manual*, serta mengendalikan perangkat output sesuai dengan kondisi yang ditentukan. Selain itu, *ESP32* juga berkomunikasi dengan *Web Server* atau *Application Programming Interface (API)* melalui koneksi *WiFi*. *Web server* berfungsi sebagai penghubung antara *ESP32* dan *dashboard web*, sehingga data hasil pembacaan sensor dapat dikirim ke server dan ditampilkan secara *real-time*, serta perintah dari pengguna dapat dikirim kembali ke *ESP32*.

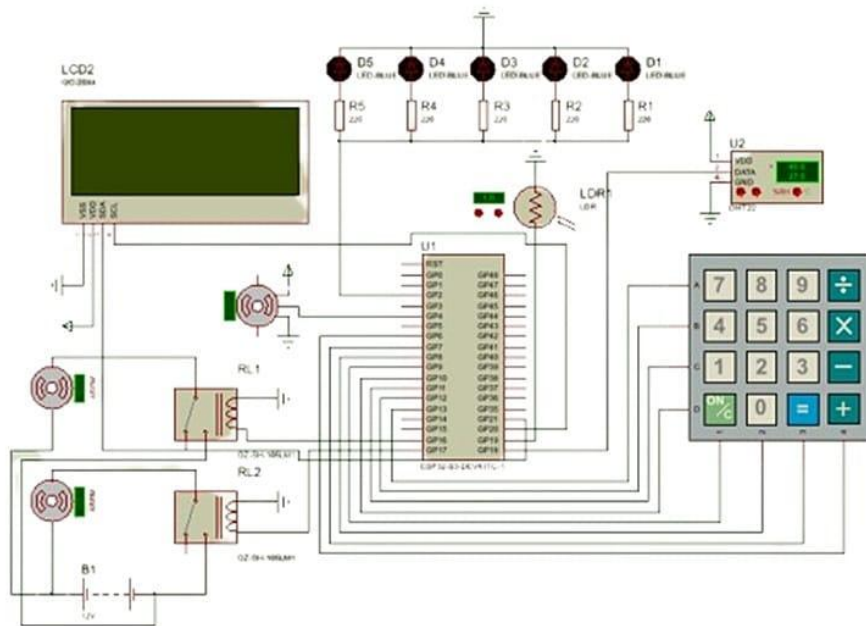
Pada bagian kanan diagram terdapat perangkat output yang dikendalikan oleh *ESP32*, yaitu motor *servo* yang digunakan untuk mekanisme buka dan tutup pintu ruang kelas, modul *relay* yang berfungsi untuk mengendalikan kipas (*Fan 1* dan *Fan 2*), serta *LED* putih yang digunakan sebagai lampu atau

indikator ruang kelas. Selain itu, sistem juga menggunakan modul *buck converter* (*LM2596*) yang berfungsi menurunkan tegangan dari *Power Supply Unit* (*PSU*) agar sesuai dengan kebutuhan tegangan *ESP32* dan perangkat lainnya.

Secara keseluruhan, diagram blok ini menggambarkan integrasi antara perangkat input, proses pengolahan data oleh mikrokontroler, komunikasi berbasis *web service*, serta perangkat output dalam sistem *Smart Class* yang dirancang.

3.3 Rancangan Gambar Rangkaian Keseluruhan

3.3.1 Rangkaian Keseluruhan



Gambar 3. 2 Rangkaian Keseluruhan Alat

Gambar 3.2 merupakan rangkaian keseluruhan sistem *Smart Class* berbasis *Internet of Things* (*IoT*) yang dirancang pada tugas akhir ini. Pada rangkaian tersebut, seluruh perangkat input dan aktuator dihubungkan ke mikrokontroler *DOIT ESP32 DevKit V1* yang berfungsi sebagai pusat kendali utama sistem.

Perangkat input pada sistem ini terdiri dari *keypad* 4×4 yang digunakan untuk autentikasi akses pintu, sensor *DHT22* yang berfungsi untuk membaca suhu dan kelembapan ruang kelas, serta sensor *Light Dependent Resistor* (*LDR*) yang digunakan untuk mendeteksi intensitas cahaya ruangan. Data yang diperoleh dari seluruh perangkat input tersebut dikirimkan ke *ESP32* untuk diproses sesuai dengan logika sistem yang telah dirancang.

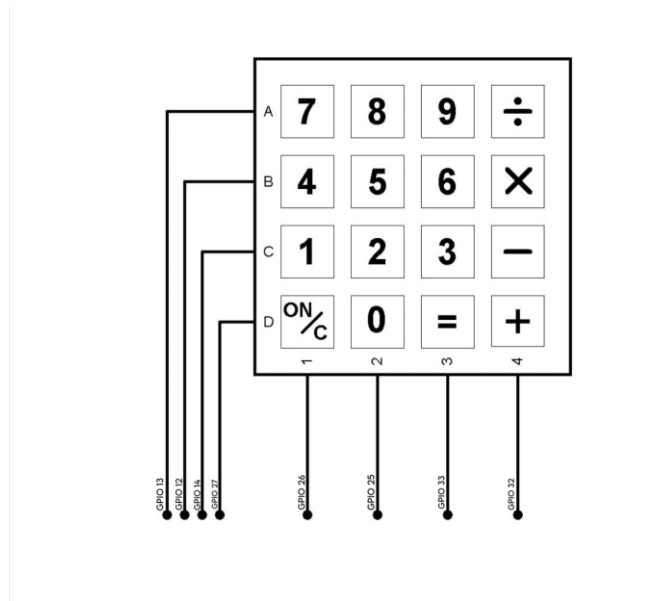
Perangkat output pada rangkaian ini meliputi motor *servo* yang digunakan sebagai mekanisme buka dan tutup pintu ruang kelas, dua unit kipas (*Fan 1* dan *Fan 2*) yang dikendalikan melalui modul *relay 2-channel*, serta *LED* putih yang berfungsi sebagai lampu atau indikator ruang kelas. *LED* dikendalikan oleh sistem berdasarkan nilai intensitas cahaya yang diperoleh dari sensor *LDR* pada mode otomatis. Selain itu, sistem juga dilengkapi dengan *Liquid Crystal Display (LCD) I2C 20×4* yang digunakan untuk menampilkan informasi kondisi ruang kelas, seperti suhu, kelembapan, intensitas cahaya, dan status sistem secara langsung.

Untuk catu daya, sistem menggunakan sumber tegangan 9 V yang dialirkan ke modul *relay* dan kipas, sedangkan mikrokontroler *ESP32* mendapatkan suplai daya melalui jalur 5 V yang telah distabilkan menggunakan modul *buck converter (LM2596)*. Seluruh rangkaian dirancang agar setiap komponen dapat bekerja secara terintegrasi, sehingga mendukung proses monitoring dan pengendalian fasilitas ruang kelas melalui *dashboard web* secara jarak jauh.

Tabel 3. 1 Rangkaian Keseluruhan Alat

No	Komponen	Pin Komponen	Pin ESP32 DevKit V1
1	Keypad 4×4	R1	GPIO 13
		R2	GPIO 12
		R3	GPIO 14
		R4	GPIO 27
		C1	GPIO 26
		C2	GPIO 25
		C3	GPIO 33
		C4	GPIO 32
2	DHT22	DATA	GPIO 18
3	LDR	A/O	GPIO 15
4	Servo Pintu	Signal	GPIO 4
5	Relay Fan 1	IN	GPIO 16
6	Relay Fan 2	IN	GPIO 17
7	LED	Anoda	GPIO 3
9	LCD I2C 20×4	SDA	GPIO 21
		SCL	GPIO 22
		VCC	5V
		GND	GND
10	Power Supply	PSU 9V → Relay/Kipas	—
		Buck Converter → 5V	5V ESP32
		GND	Semua komponen

3.3.2 Rangkaian keypad 4x4



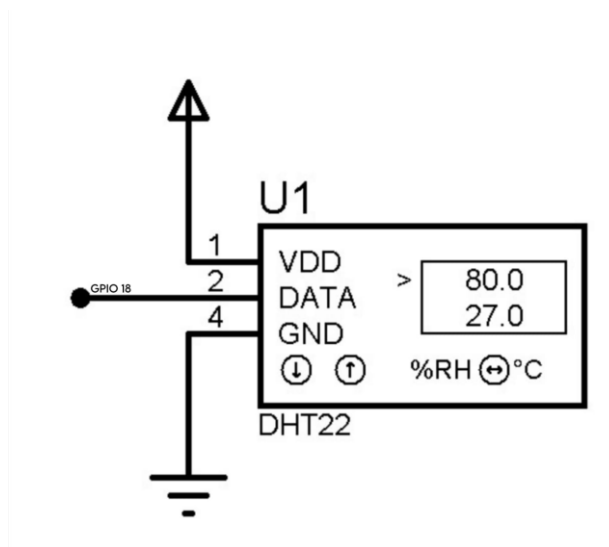
Gambar 3. 3 Rangkaian Keypad 4x4

Pada Gambar 3.3 menunjukkan rangkaian keypad 4×4 yang terhubung ke ESP32 DevKit V1. Pada rangkaian ini, empat pin baris keypad (R1–R4) dihubungkan ke pin GPIO 13, 12, 14, dan 27, sedangkan empat pin kolom (C1–C4) dihubungkan ke GPIO 26, 25, 33, dan 32. Konfigurasi ini memungkinkan ESP32 membaca kombinasi tombol yang ditekan pengguna dan digunakan sebagai input PIN untuk sistem akses pintu.

Tabel 3. 2 Rangkaian Keypad 4x4

No	Pin Keypad	Pin ESP32
1	R1	GPIO 13
2	R2	GPIO 12
3	R3	GPIO 14
4	R4	GPIO 27
5	C1	GPIO 26
6	C2	GPIO 25
7	C3	GPIO 33
8	C4	GPIO 32

3.3.3 Rangkaian Sensor DHT 22



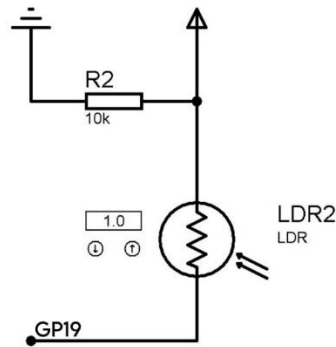
Gambar 3. 4 Rangkaian Sensor DHT 22

Pada Gambar 3.4 menunjukkan rangkaian sensor DHT22 yang terhubung ke ESP32. Sensor ini menggunakan pin data yang terhubung ke GPIO 18. DHT22 menerima suplai daya dari pin 5V dan GND. Dengan konfigurasi ini sensor dapat mengirimkan data suhu dan kelembapan ke ESP32 secara digital.

Tabel 3. 3 Rangkaian Sensor DHT 22

No	Pin DHT22	Pin ESP32
1	VCC	5V
2	DATA	GPIO 18
3	GND	GND

3.3.4 Rangkaian LDR



Gambar 3. 5 Rangkaian LDR

Pada Gambar 3.5 menunjukkan rangkaian sensor Light Dependent Resistor (LDR) yang terhubung ke mikrokontroler ESP32 DevKit V1. Sensor LDR digunakan untuk mendeteksi intensitas cahaya di dalam ruang kelas sebagai dasar pengendalian LED putih pada sistem Smart Class.

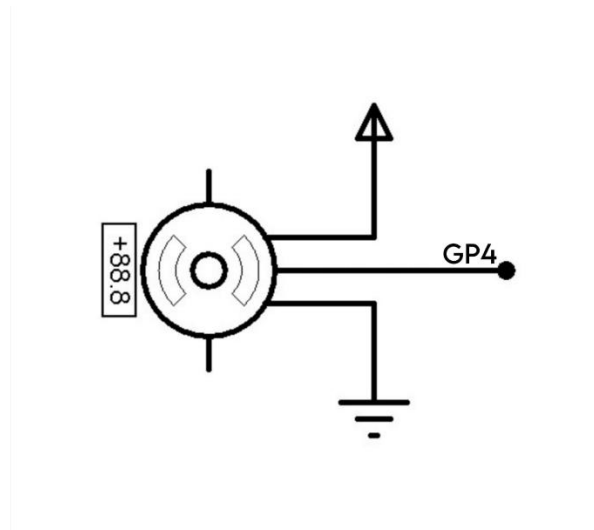
Rangkaian sensor LDR disusun menggunakan konfigurasi pembagi tegangan (voltage divider) sehingga menghasilkan sinyal analog yang berubah sesuai dengan intensitas cahaya lingkungan. Output rangkaian LDR dihubungkan ke salah satu pin Analog to Digital Converter (ADC) pada ESP32, yaitu GPIO 15. Sensor LDR mendapatkan suplai tegangan dari pin 3,3 V dan GND pada ESP32.

Nilai tegangan analog yang dihasilkan oleh sensor LDR akan dibaca oleh ESP32 dan digunakan sebagai parameter pengendalian LED putih secara otomatis. Pada kondisi cahaya terang (siang hari), sistem akan mengaktifkan LED, sedangkan pada kondisi cahaya rendah atau gelap (malam hari), LED akan dimatikan sesuai dengan logika sistem yang telah dirancang.

Tabel 3. 4 Rangkaian Sensor LDR

No	Pin LDR	Pin ESP32
1	VCC	3.3 V
2	Output	GPIO 15
3	GND	GND

3.3.5 Rangkaian Servo



Gambar 3. 6 Rangkaian Servo

Gambar 3.6 merupakan rangkaian motor servo yang digunakan pada sistem Smart Class sebagai aktuator untuk mekanisme buka dan tutup pintu ruang kelas. Motor servo ini terhubung langsung ke mikrokontroler ESP32 DevKit V1 dan mendapatkan suplai daya dari jalur tegangan 5 V.

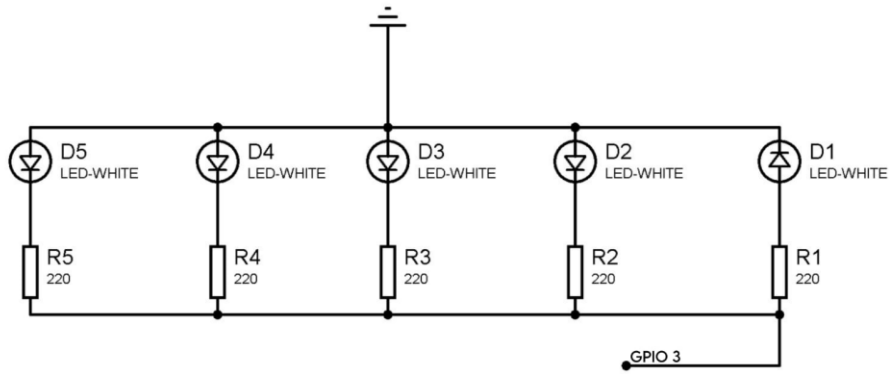
Pada rangkaian tersebut, pin sinyal motor servo dihubungkan ke salah satu pin General Purpose Input Output (GPIO) pada ESP32, yaitu GPIO 4. Motor servo dikendalikan menggunakan sinyal Pulse Width Modulation (PWM) yang dihasilkan oleh ESP32 untuk mengatur sudut putaran sesuai dengan perintah sistem.

Motor servo berfungsi untuk membuka dan menutup akses pintu ruang kelas secara otomatis berdasarkan input dari keypad maupun perintah yang diberikan melalui dashboard web. Dengan konfigurasi ini, motor servo dapat bekerja secara presisi dan responsif sesuai dengan logika pengendalian yang diterapkan pada sistem Smart Class.

Tabel 3. 5 Rangkaian Servo

No	Komponen	Pin Servo	Pin ESP32
1	Servo Pintu	Signal	GPIO 4
2	VCC	5V	5V
3	GND	GND	GND

3.3.6 Rangkaian LED



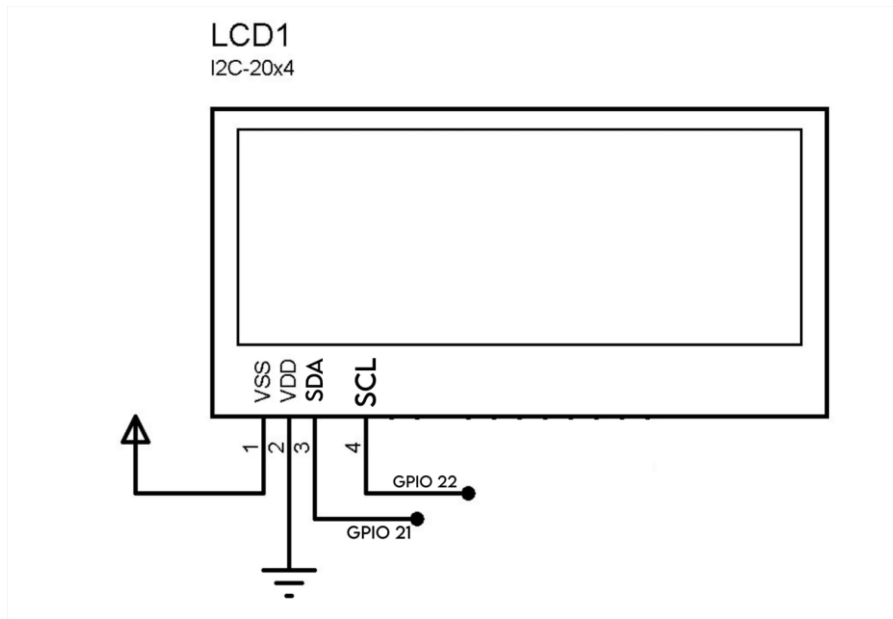
Gambar 3. 7 Rangkaian LED

Gambar 3. 7 merupakan rangkaian LED yang digunakan sebagai indikator pada sistem Smart Class. LED berfungsi sebagai lampu ruangan yang dapat menyala berdasarkan intensitas cahaya atau perintah manual dari web. Pada rangkaian ini, LED terhubung ke pin GPIO 3 pada ESP32. LED dihubungkan ke ground melalui resistor untuk menjaga kestabilan arus.

Tabel 3. 6 Rangkaian LED

No	Komponen	Pin LED	Pin ESP32
1	LED	Anoda	GPIO 3
		Katoda	GND

3.3.7 Rangkaian Liquid Crystal Display 20x4 I2C



Gambar 3. 8 Rangkaian Liquid Crystal Display 20x4 I2C

Pada Gambar 3.8 menunjukkan rangkaian LCD I2C 20×4 yang terhubung ke ESP32 melalui jalur komunikasi I2C. Pin SDA LCD dihubungkan ke GPIO 21 dan pin SCL dihubungkan ke GPIO 22. LCD mendapat suplai daya 5V dan ground. Dengan konfigurasi ini, LCD dapat menampilkan informasi suhu, kondisi cuaca, status kipas, serta status pintu secara real-time.

Tabel 3. 7 Rangkaian Liquid Crystal Display 20x4 I2C

No	Pin LCD	Pin ESP32	Fungsi
1	SDA	GPIO 21	Transfer data I2C
2	SCL	GPIO 22	Clock I2C
3	VCC	5V	Suplai daya LCD
4	GND	GND	Ground

3.4 Rancangan Metodologi Pengembangan *Software*

Metodologi pengembangan software yang digunakan pada penelitian ini bertujuan untuk menggambarkan alur kerja sistem Smart Class secara menyeluruh, mulai dari proses penerimaan input hingga menghasilkan output berupa pengendalian perangkat dan informasi monitoring. Metodologi ini dirancang agar sistem dapat berjalan secara terstruktur, mudah dipahami, serta mampu mendukung proses otomatisasi dan kontrol manual yang diterapkan pada sistem.

Pengembangan perangkat lunak dilakukan dengan pendekatan alur proses (*process flow-based*), di mana setiap tahapan saling terhubung dan berjalan secara berulang (*looping*) untuk memastikan sistem dapat bekerja secara realtime dan berkelanjutan.

3.4.1 Langkah Alur Metodologi

Secara umum, alur metodologi pengembangan *software* pada sistem *Smart Class* dapat dijelaskan sebagai berikut:

1. Inisialisasi Sistem

Proses dimulai ketika sistem pertama kali diaktifkan. Mikrokontroler *ESP32 DevKit V1* melakukan inisialisasi terhadap seluruh komponen perangkat keras yang digunakan, meliputi sensor *DHT22*, sensor *Light Dependent Resistor (LDR)*, keypad 4×4, motor *servo*, modul *relay*, *LED*, serta *Liquid Crystal Display (LCD)*. Selain itu, sistem juga melakukan konfigurasi dan koneksi jaringan *WiFi* agar dapat terhubung dengan server.

2. Penerimaan Input

Input pada sistem diperoleh dari dua sumber utama, yaitu:

1. Input sensor, berupa data suhu dan kelembapan ruang kelas yang diperoleh dari sensor *DHT22* serta data intensitas cahaya yang diperoleh dari sensor *LDR*.
2. Input pengguna, yang berasal dari *dashboard web* dalam bentuk pemilihan mode operasi (*auto* atau *manual*) serta perintah pengendalian perangkat.

3. Pengambilan Data dari Server
Setelah terhubung ke jaringan, *ESP32* secara berkala melakukan permintaan data ke server menggunakan metode *HTTP GET*. Data yang diambil meliputi mode sistem dan status aktuator pada mode manual, seperti pengaturan kipas, lampu (*LED*), dan pintu.
4. Proses pengolahan data
Pada tahap ini, sistem melakukan pengolahan data berdasarkan mode operasi yang aktif:
 - a. Mode otomatis, sistem mengambil keputusan berdasarkan nilai sensor yang diperoleh.
 - b. Mode manual, sistem menonaktifkan logika sensor dan hanya menjalankan perintah yang diterima dari server.
5. Eksekusi Output
Hasil pengolahan data digunakan untuk mengendalikan perangkat output, yaitu mengatur posisi motor *servo* sebagai mekanisme buka dan tutup pintu ruang kelas serta menghidupkan atau mematikan kipas melalui modul *relay* sesuai dengan kondisi sistem.
6. Pengiriman data ke server
Setelah sistem dijalankan, *ESP32* mengirimkan data sensor dan status perangkat ke server menggunakan metode *HTTP POST*. Data ini disimpan ke dalam database dan ditampilkan secara realtime pada dashboard web.
7. Monitoring dan Evaluasi
Seluruh proses yang berjalan ditampilkan pada Serial Monitor sebagai media monitoring dan evaluasi sistem. Informasi yang ditampilkan meliputi mode sistem, nilai sensor, serta status aktuator untuk memastikan sistem bekerja sesuai dengan perancangan.

Seluruh proses yang berjalan ditampilkan pada Serial Monitor sebagai media monitoring dan evaluasi sistem. Informasi yang ditampilkan meliputi mode sistem, nilai sensor, serta status aktuator untuk memastikan sistem bekerja sesuai dengan perancangan.

3.4.2 Formulasi Matematika

Formulasi matematika pada sistem Smart Class digunakan sebagai dasar dalam proses pengolahan data sensor dan pengambilan keputusan sistem, khususnya pada mode otomatis. Perhitungan yang diterapkan bersifat sederhana, namun cukup untuk mendukung proses monitoring dan pengendalian perangkat secara akurat dan realtime.

1. Formulasi Pengukuran Suhu dan Kelembapan (Sensor DHT22)

Sensor DHT22 menghasilkan data suhu dan kelembapan dalam bentuk nilai digital yang langsung dibaca oleh mikrokontroler ESP32. Nilai suhu dan kelembapan yang terbaca dinyatakan sebagai:

$$T = T_{sensor}$$

$$H = H_{sensor}$$

dengan:

- a. T = suhu ruangan dalam derajat Celcius ($^{\circ}C$)
- b. H = kelembapan udara dalam persen (%)

Nilai suhu dan kelembapan ini digunakan sebagai parameter utama dalam pengendalian kipas (fan). Logika pengambilan keputusan ditentukan dengan persamaan kondisi berikut:

$$Fan1 = \begin{cases} 1, & \text{jika } T > 30^{\circ}C \text{ atau } H > 80\% \\ 0, & \text{jika } T \leq 30^{\circ}C \text{ dan } H \leq 80\% \end{cases}$$

Gambar 3. 9 Rumus Fan 1

Artinya, kipas akan menyala secara otomatis apabila suhu atau kelembapan melebihi batas yang telah ditentukan, dan akan mati ketika kondisi ruangan kembali normal.

2. Formulasi Pengukuran Intensitas Cahaya (Sensor LDR)

Sensor *Light Dependent Resistor* (LDR) digunakan untuk mendeteksi intensitas cahaya di dalam ruang kelas. Sensor LDR menghasilkan nilai analog yang berubah sesuai dengan kondisi cahaya lingkungan, di mana nilai resistansi LDR akan menurun ketika intensitas cahaya meningkat dan meningkat ketika intensitas cahaya menurun. Nilai analog dari sensor LDR dibaca oleh *Analog to Digital Converter* (ADC) pada mikrokontroler ESP32 dan dinyatakan sebagai berikut:

$$C = C_{sensor}$$

Nilai intensitas cahaya ini digunakan sebagai parameter dalam pengendalian LED putih secara otomatis. Logika pengambilan keputusan ditentukan dengan menggunakan nilai ambang batas (threshold) sebagai berikut:

$$LED = \begin{cases} 1, & \text{jika } C > C_{threshold} \\ 0, & \text{jika } C \leq C_{threshold} \end{cases}$$

Gambar 3. 10 Rumus Pengendalian LED Berdasarkan Sensor LDR

3. Formulasi Mode Operasi Sistem (Manual dan Otomatis)
Mode operasi sistem dinyatakan dengan variabel:

$$M = \begin{cases} Auto, & \text{jika } M = 1 \\ Manual, & \text{jika } M = 0 \end{cases}$$

Gambar 3. 11 Rumus Motor dc

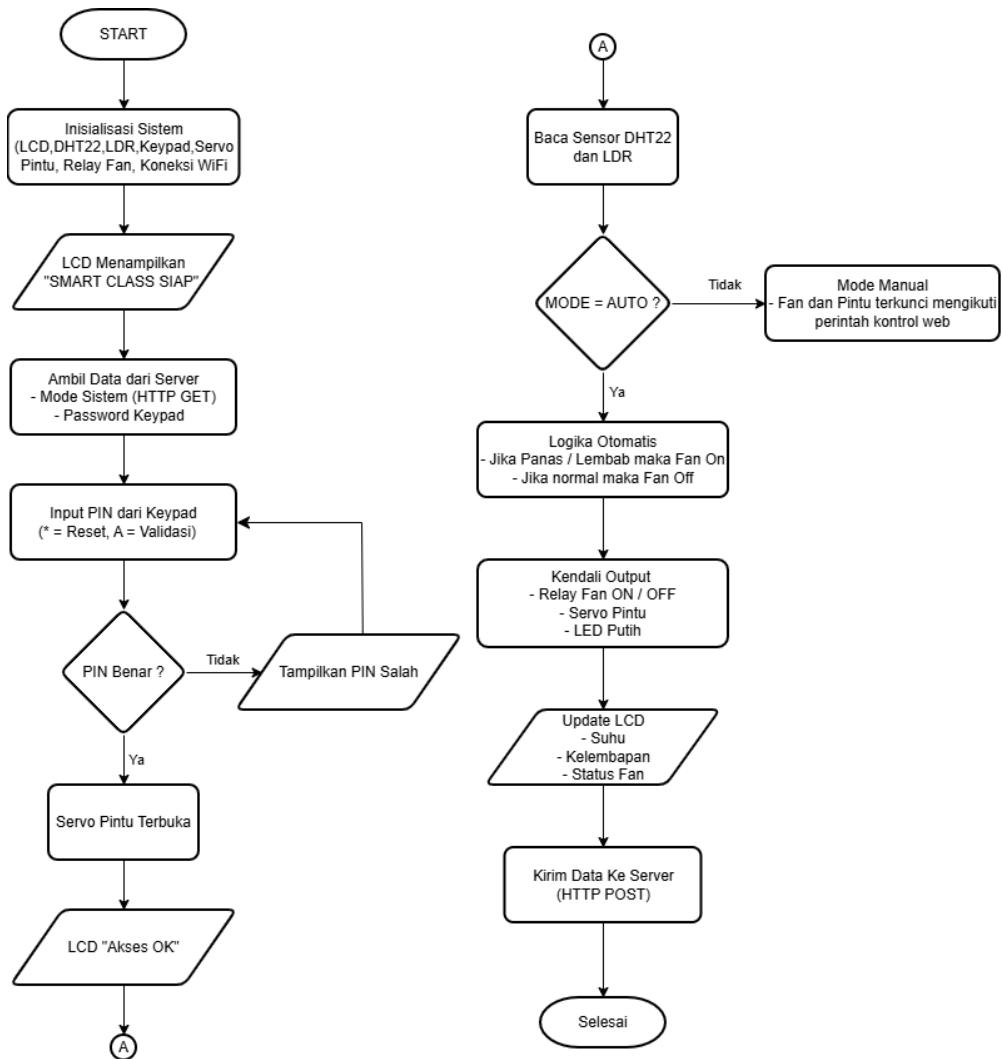
Pada mode otomatis, seluruh aktuator dikendalikan berdasarkan hasil perhitungan sensor dan formulasi matematika yang telah dijelaskan sebelumnya. Sedangkan pada mode manual, status perangkat ditentukan langsung oleh perintah pengguna melalui dashboard web tanpa memperhatikan nilai sensor.

4. Formulasi Validasi PIN Keypad
Validasi PIN dilakukan dengan membandingkan input pengguna dengan PIN yang tersimpan di server. Proses validasi dinyatakan sebagai:

$$Akses = \begin{cases} Diterima, & \text{jika } PIN_{input} = PIN_{server} \\ Ditolak, & \text{jika } PIN_{input} \neq PIN_{server} \end{cases}$$

Jika PIN benar, servo pintu akan bergerak membuka. Sebaliknya, jika PIN salah sebanyak tiga kali berturut-turut, sistem akan mengaktifkan buzzer dan indikator alarm.

3.5 Flowchart Sistem



Gambar 3. 12 Flowchart Sistem

Flowchart di atas menggambarkan alur kerja keseluruhan dari sistem *Smart Class* yang dikembangkan pada tugas akhir ini. Proses dimulai ketika mikrokontroler *ESP32 DevKit V1* diaktifkan dan melakukan inisialisasi seluruh perangkat yang digunakan, meliputi *Liquid Crystal Display (LCD)*, sensor *DHT22*, sensor *Light Dependent Resistor (LDR)*, *keypad 4x4*, motor *servo* pintu, modul *relay* kipas, *LED* putih, serta konfigurasi koneksi jaringan *WiFi* menggunakan *WiFi Manager*. Setelah seluruh perangkat berhasil diinisialisasi, *LCD* menampilkan informasi “SMART CLASS SIAP” sebagai tanda bahwa sistem siap digunakan.

Selanjutnya, *ESP32* mengambil data awal dari server menggunakan metode *HTTP GET*, yang meliputi mode operasi sistem (*auto* atau *manual*) serta *password* untuk autentikasi *keypad*. Proses ini bertujuan untuk menjaga sinkronisasi antara perangkat keras dan *dashboard* web.

Setelah data diperoleh, pengguna melakukan proses autentikasi dengan memasukkan PIN melalui *keypad*. Sistem kemudian melakukan validasi PIN dengan mencocokkan input pengguna dengan data *password* yang diperoleh dari server. Apabila PIN yang dimasukkan benar, sistem akan membuka pintu menggunakan motor *servo* dan *LCD* menampilkan pesan “Akses OK”. Sebaliknya, jika PIN yang dimasukkan tidak sesuai, sistem akan menampilkan pesan kesalahan dan kembali meminta pengguna untuk memasukkan PIN yang benar.

Setelah proses autentikasi berhasil, sistem melanjutkan dengan melakukan pembacaan sensor secara berkala. Sensor *DHT22* digunakan untuk membaca nilai suhu dan kelembapan ruang kelas, sedangkan sensor *LDR* digunakan untuk mendeteksi intensitas cahaya ruangan. Data hasil pembacaan sensor ini menjadi dasar dalam pengambilan keputusan sistem.

Sistem kemudian melakukan pengecekan mode operasi. Apabila sistem berada pada *mode otomatis*, *ESP32* akan menjalankan logika otomatis untuk mengendalikan perangkat. Kipas akan dinyalakan ketika suhu atau kelembapan melebihi batas yang telah ditentukan dan dimatikan ketika kondisi kembali normal. Selain itu, *LED* putih akan dikendalikan berdasarkan nilai intensitas cahaya dari sensor *LDR*, di mana *LED* akan menyala pada kondisi cahaya terang dan mati pada kondisi cahaya rendah atau gelap.

Apabila sistem berada pada *mode manual*, maka seluruh pengendalian perangkat, termasuk kipas dan pintu, mengikuti perintah yang diberikan melalui *dashboard* web tanpa mempertimbangkan nilai sensor.

Setelah logika sistem dijalankan, *ESP32* mengendalikan perangkat output yang terdiri dari *relay* kipas, motor *servo* pintu, dan *LED* putih. Sistem kemudian memperbarui tampilan *LCD* dengan informasi terbaru yang meliputi suhu, kelembapan, serta status kipas. Pada interval waktu tertentu, *ESP32* juga mengirimkan data sensor dan status perangkat ke server menggunakan metode *HTTP POST* agar dapat disimpan ke dalam *database* dan ditampilkan secara *real-time* pada *dashboard* web.

Seluruh proses tersebut berjalan secara berulang (*loop*), sehingga sistem *Smart Class* dapat melakukan monitoring dan pengendalian fasilitas ruang kelas secara terus-menerus dan berkelanjutan sesuai dengan perancangan sistem.

3.6 Rancangan Layar

Rancangan layar sistem *Smart Class* berbasis web dibuat untuk memudahkan pengguna dalam melakukan monitoring, pengendalian, dan pengaturan sistem secara terpusat. Antarmuka sistem dirancang dengan tampilan yang sederhana dan mudah dipahami agar pengguna dapat mengakses informasi dan melakukan kontrol sistem secara *real-time* melalui web browser.

3.6.1 Layar Dashboard

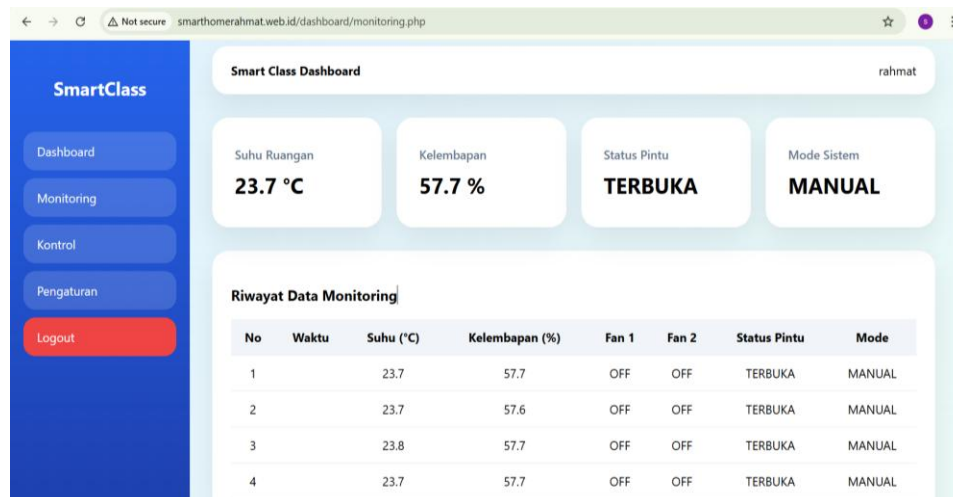


Gambar 3. 13 Layar Dashboard

Gambar 3.19 menunjukkan rancangan layar Dashboard pada sistem Smart Class berbasis web. Halaman ini merupakan halaman utama yang menampilkan ringkasan kondisi ruang kelas secara real-time.

Informasi yang ditampilkan pada halaman Dashboard meliputi suhu ruangan, kelembapan udara, kualitas udara, dan kondisi cuaca. Selain itu, halaman ini juga menampilkan status perangkat seperti kipas, lampu, dan pintu. Dengan adanya halaman Dashboard, pengguna dapat memantau kondisi sistem secara cepat tanpa perlu berpindah ke halaman lain.

3.6.2 Layar Monitoring



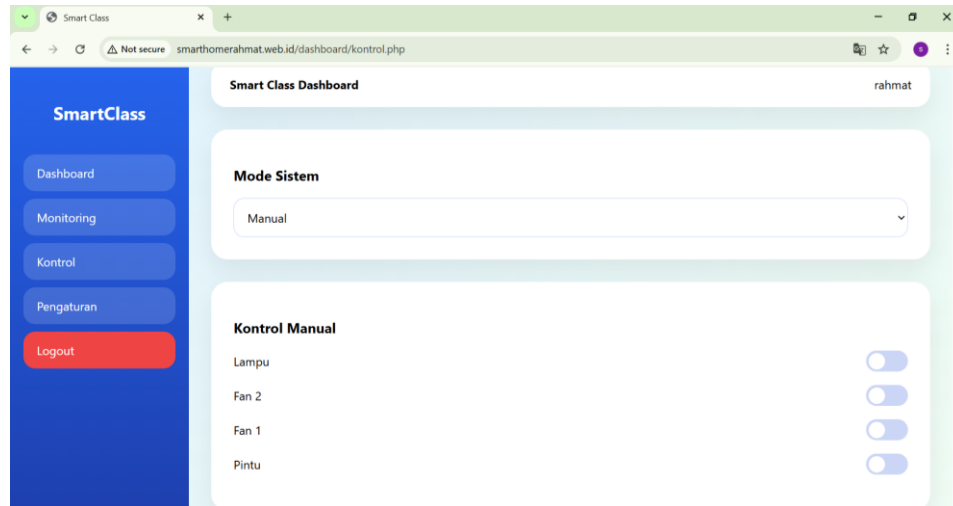
Gambar 3. 14 Layar Monitoring

Gambar 3.20 menunjukkan rancangan layar Monitoring yang digunakan untuk menampilkan data sensor secara detail. Pada halaman ini

ditampilkan nilai suhu, kelembapan, kualitas udara, kondisi cuaca, serta tabel riwayat data sensor yang tersimpan pada database.

Halaman Monitoring berfungsi untuk membantu pengguna memantau perubahan kondisi ruang kelas berdasarkan data historis yang diperoleh secara berkala.

3.6.3 Layar Kontrol

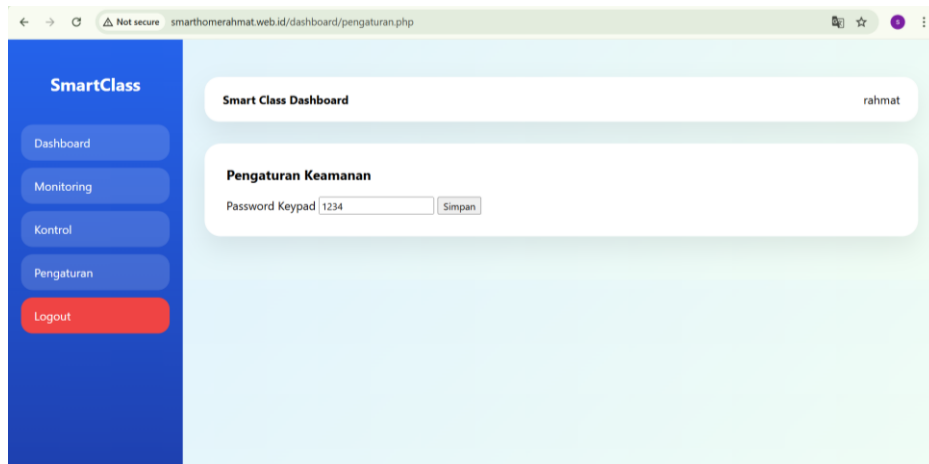


Gambar 3. 15 Layar Kontrol

Gambar 3.21 menunjukkan rancangan layar Kontrol yang digunakan untuk mengatur mode operasi sistem. Pengguna dapat memilih mode otomatis atau mode manual sesuai dengan kebutuhan.

Pada mode manual, pengguna dapat mengendalikan perangkat seperti kipas, lampu, dan pintu melalui tombol kontrol yang tersedia. Halaman Kontrol berfungsi sebagai media pengendalian perangkat secara langsung melalui sistem berbasis web.

3.6.4 Layar Pengaturan



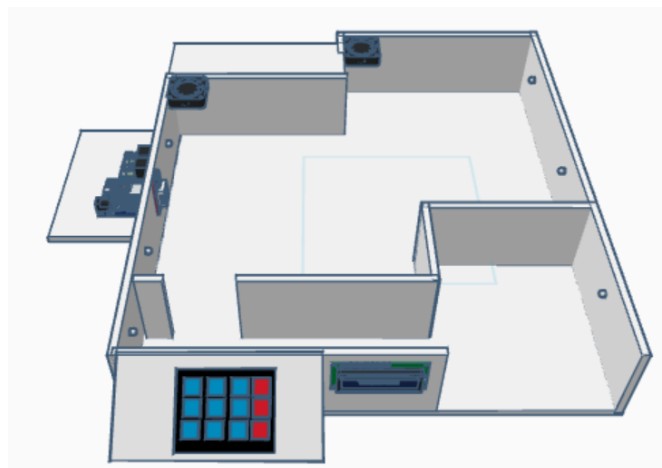
Gambar 3. 16 Layar Pengaturan

Gambar 3.22 menunjukkan rancangan layar Pengaturan yang digunakan untuk mengatur aspek keamanan sistem. Pada halaman ini pengguna dapat melakukan pengaturan password keypad yang digunakan sebagai sistem autentikasi akses pintu.

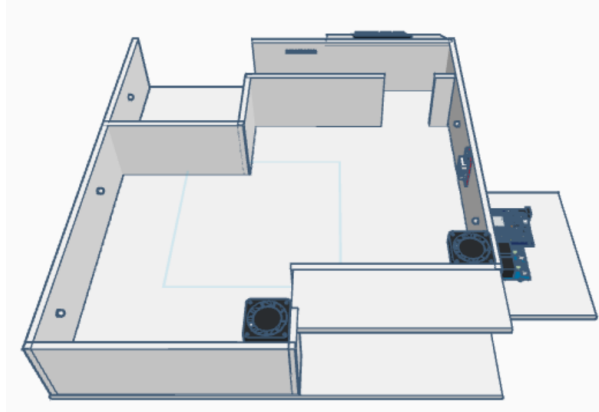
Halaman Pengaturan berfungsi untuk meningkatkan keamanan sistem Smart Class dan membatasi akses hanya kepada pengguna yang berwenang.

3.7 Rancangan Desain 3D

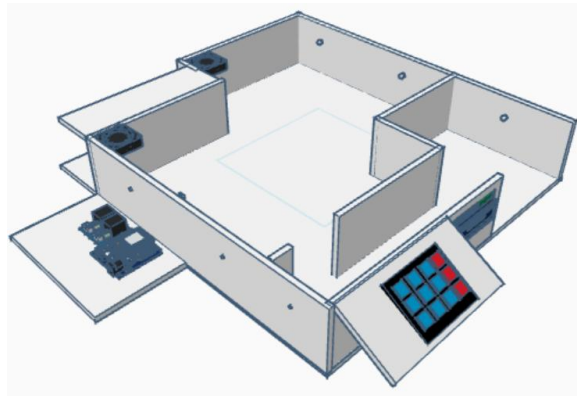
Berikut adalah gambar rancangan desain 3D dari sistem Smart Class yang menampilkan tampak atas, tampak samping kiri, tampak samping kanan, dan tampak belakang sebagai visualisasi tata letak komponen serta posisi instalasi perangkat pada prototipe.



Gambar 3. 17 Tampak Depan



Gambar 3. 18 Tampak Belakang



Gambar 3. 19 Tampak Samping

3.8 Rancangan Pengujian

Dalam proses pembuatan prototipe sistem *Smart Class*, dilakukan penyusunan rancangan pengujian sebelum sistem diimplementasikan secara menyeluruh. Rancangan pengujian ini bertujuan untuk memastikan bahwa setiap perangkat input dan aktuator yang digunakan dapat berfungsi sesuai dengan logika sistem yang telah dirancang.

Rancangan pengujian mencakup pengujian pembacaan sensor *DHT22* untuk mengetahui kemampuan sistem dalam memonitor suhu dan kelembapan ruang kelas, pengujian *keypad* 4×4 sebagai media autentikasi akses pintu, serta pengujian motor *servo* sebagai aktuator pembuka dan penutup pintu. Selain itu, dilakukan juga pengujian modul *relay* dalam mengendalikan kipas baik pada mode otomatis maupun mode manual, serta pengujian *LCD* sebagai media tampilan informasi kondisi sistem.

Setiap komponen diuji secara terpisah untuk memastikan fungsinya berjalan dengan baik, kemudian diuji kembali secara terintegrasi sebagai satu

kesatuan sistem. Pengujian terintegrasi dilakukan untuk memastikan bahwa mikrokontroler *ESP32 DevKit V1* mampu memproses data sensor secara *real-time* dan mengeksekusi perintah pengendalian perangkat sesuai dengan mode operasi yang dipilih, baik *mode otomatis* maupun *mode manual* melalui *dashboard web*. Dengan adanya rancangan pengujian ini, diharapkan sistem *Smart Class* dapat berjalan secara stabil, responsif, dan sesuai dengan kebutuhan penggunaan di lingkungan ruang kelas.

Tabel 3. 8 Rancangan Pengujian

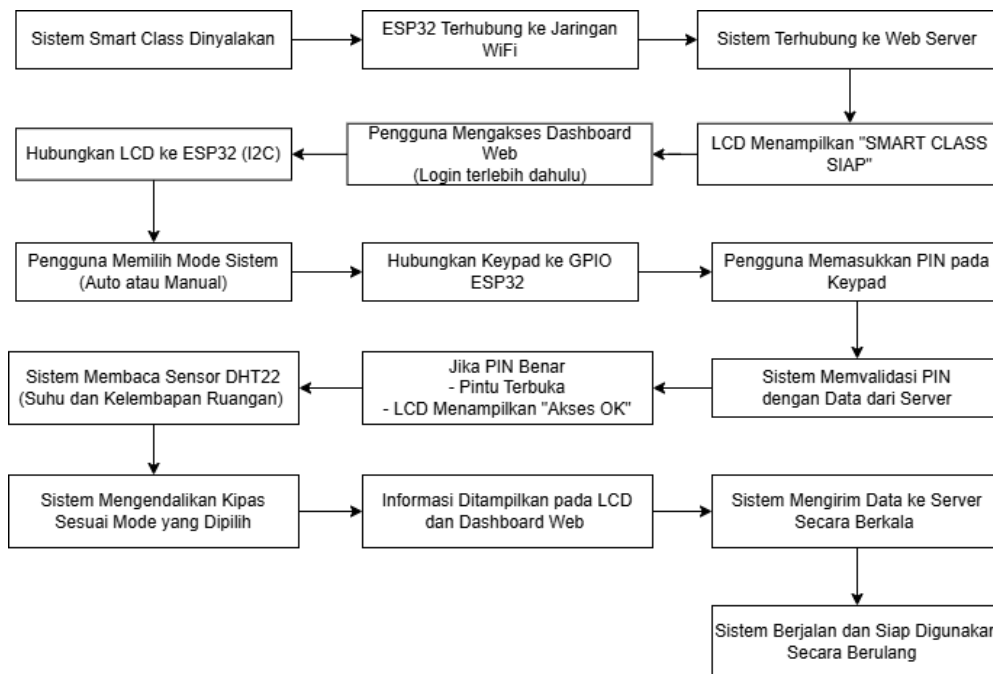
No	Komponen	Tujuan Pengujian	Metode Pengujian	Hasil yang Diharapkan
1	Keypad 4×4	Menguji proses input PIN pengguna	Menekan kombinasi PIN benar / salah	PIN benar → pintu terbuka; PIN salah → counter bertambah; 3× salah → alarm aktif
2	Sensor DHT22	Menguji pembacaan suhu & kelembapan	Mengarahkan sensor pada lingkungan berbeda (panas/lembap)	Nilai suhu & kelembapan berubah sesuai kondisi ruangan
3	Sensor LDR	Menguji pendeteksian intensitas cahaya	Mengubah kondisi cahaya (terang/gelap) di sekitar sensor	Sistem mendeteksi perubahan cahaya dan mengontrol LED secara otomatis sesuai kondisi
4	Servo Pintu	Menguji mekanisme gerak aktuator	Memberikan perintah buka/tutup	Servo bergerak pada sudut yang sesuai (0°/90°)
5	Relay (Kipas 1 & 2)	Menguji kendali beban kipas	Mengaktifkan logika otomatis & manual	Relay <i>ON/OFF</i> sesuai perintah; kipas menyala stabil
6	LED Putih	Menguji LED	Mengaktifkan kondisi otomatis	LED menyala sesuai kondisi sistem
7	LCD I2C 20×4	Menguji tampilan proses sistem	Menyalakan sistem dan mengamati update LCD	LCD menampilkan suhu, cuaca, status perangkat, dan pesan PIN

8	Koneksi API / Web Server	Menguji komunikasi GET & POST	Mengambil status perangkat & mengirim data sensor	Data berhasil terkirim dan dashboard menampilkan perubahan secara realtime
---	-----------------------------	----------------------------------	--	---

BAB IV IMPLEMENTASI DAN EVALUASI PROGRAM

Setelah tahap perancangan yang telah dibahas pada BAB III, maka pada BAB IV ini dilakukan proses implementasi dan pengujian terhadap sistem *Smart Class* yang telah dirancang. Bab ini berfokus pada bagaimana setiap komponen diintegrasikan menjadi satu kesatuan sistem, cara kerja sistem saat dijalankan, serta hasil pengujian yang diperoleh dari masing-masing fungsi yang telah diuji. Melalui pembahasan pada bab ini, dapat diketahui apakah seluruh perancangan yang telah dilakukan sebelumnya dapat berjalan secara optimal dan sesuai dengan tujuan penelitian.

4.1 Cara Kerja Sistem



Gambar 4. 1 Diagram Blok Cara Kerja

Alur kerja sistem *Smart Class* dimulai ketika sistem pertama kali dinyalakan. Pada tahap awal, mikrokontroler *ESP32 DevKit V1* diaktifkan dan melakukan koneksi ke jaringan *WiFi* yang tersedia. Setelah koneksi jaringan berhasil, sistem terhubung dengan *web server* sebagai media pertukaran data antara perangkat keras dan *dashboard web*.

Setelah sistem terhubung dengan *web server*, *Liquid Crystal Display (LCD)* akan menampilkan informasi awal berupa pesan “SMART CLASS SIAP” sebagai indikasi bahwa sistem telah aktif dan siap digunakan. Pengguna kemudian dapat mengakses *dashboard web* melalui perangkat yang terhubung ke jaringan internet dengan melakukan proses *login* terlebih dahulu.

Pengguna selanjutnya memilih mode sistem yang akan digunakan, yaitu *mode otomatis* atau *mode manual*, melalui *dashboard* web. Mode yang dipilih akan dikirimkan ke *ESP32* dan digunakan sebagai acuan dalam proses pengendalian sistem.

Untuk mengakses ruang kelas, pengguna melakukan proses autentikasi dengan memasukkan PIN melalui *keypad* yang terhubung ke *ESP32* melalui pin *General Purpose Input Output (GPIO)*. Data PIN yang dimasukkan oleh pengguna kemudian divalidasi oleh sistem dengan membandingkannya dengan data PIN yang tersimpan di *web server*. Apabila PIN yang dimasukkan benar, sistem akan membuka pintu menggunakan motor *servo* dan *LCD* akan menampilkan pesan “Akses OK”.

Setelah proses autentikasi berhasil, sistem mulai melakukan pembacaan sensor secara berkala. Sensor *DHT22* digunakan untuk membaca nilai suhu dan kelembapan ruang kelas, sedangkan sensor *Light Dependent Resistor (LDR)* digunakan untuk mendeteksi intensitas cahaya ruangan. Data dari kedua sensor tersebut diproses oleh *ESP32* sebagai dasar pengambilan keputusan sistem.

Berdasarkan nilai intensitas cahaya yang diperoleh dari sensor *LDR*, sistem secara otomatis mengendalikan *LED* putih sebagai lampu ruang kelas. Pada kondisi cahaya ruangan terang, *LED* akan dimatikan, sedangkan pada kondisi cahaya rendah atau gelap, *LED* akan menyala sesuai dengan logika sistem yang telah dirancang.

Selain pengendalian *LED*, data suhu dan kelembapan dari sensor *DHT22* digunakan untuk mengendalikan kipas sesuai dengan mode sistem yang dipilih. Pada *mode otomatis*, kipas akan menyala ketika suhu atau kelembapan melebihi batas yang ditentukan, sedangkan pada *mode manual* pengendalian kipas mengikuti perintah dari *dashboard* web.

Seluruh informasi hasil pembacaan sensor, termasuk suhu, kelembapan, dan intensitas cahaya, serta status perangkat seperti kipas, *LED*, dan pintu, ditampilkan pada *LCD* dan *dashboard* web secara *real-time*. Selain itu, sistem secara berkala mengirimkan data sensor dan status perangkat ke *web server* untuk disimpan ke dalam *database*.

Seluruh proses tersebut berjalan secara berulang (*loop*), sehingga sistem *Smart Class* selalu berada dalam kondisi siap digunakan untuk melakukan monitoring dan pengendalian fasilitas ruang kelas secara berkelanjutan.

4.2 Hasil Pengujian Perangkat Keras

Pengujian perangkat keras dilakukan untuk memastikan bahwa seluruh komponen yang digunakan dalam prototype sistem *Smart Class* berbasis *Internet of Things (IoT)* dapat berfungsi dengan baik dan sesuai dengan perancangan yang telah dilakukan. Pengujian ini bertujuan untuk mengetahui kinerja masing-masing perangkat keras dalam mendukung sistem secara keseluruhan.

Setiap pengujian dilakukan secara terpisah pada masing-masing komponen utama yang digunakan, meliputi *keypad* 4×4 sebagai perangkat input autentikasi, sensor *DHT22* sebagai pembaca suhu dan kelembapan ruangan, motor *servo* sebagai penggerak pintu, modul *relay* untuk mengendalikan kipas, serta *Liquid Crystal Display (LCD)* sebagai media tampilan informasi. Pengujian dilakukan dengan cara memberikan input tertentu pada setiap komponen dan mengamati respons output yang dihasilkan oleh sistem.

Hasil dari setiap pengujian dianalisis untuk memastikan bahwa perangkat keras dapat bekerja sesuai dengan fungsi yang diharapkan dan terintegrasi dengan baik dengan mikrokontroler *ESP32*. Dengan adanya pengujian perangkat keras ini, dapat diketahui tingkat keberhasilan integrasi antar komponen serta kesiapan sistem *Smart Class* untuk dioperasikan, baik dalam mode otomatis maupun mode manual melalui sistem berbasis web.

4.2.1 Pengujian keypad 4x4

Pengujian keypad 4×4 dilakukan untuk memastikan bahwa setiap tombol pada keypad dapat terdeteksi dengan benar oleh mikrokontroler *ESP32* serta mampu menjalankan fungsi autentikasi akses pintu. Proses pengujian dilakukan dengan menekan tombol angka secara berurutan untuk membentuk PIN, kemudian menekan tombol konfirmasi untuk melakukan verifikasi.

Tabel 4. 1 Pengujian Keypad

No	Input PIN	Respon LCD	Status Sistem
1	PIN benar	PIN BENAR	Pintu Terbuka
2	PIN salah	PIN SALAH	Akses Ditolak
3	Tombol reset	PIN direset	Standby

Berdasarkan hasil pengujian, keypad 4×4 pada Tabel 4.1 mampu membaca setiap input tombol dengan baik tanpa adanya kesalahan pembacaan. Sistem berhasil membandingkan input PIN dengan data password yang diperoleh dari server. Ketika PIN yang dimasukkan benar, sistem memberikan respon berupa tampilan pesan keberhasilan pada LCD dan membuka pintu melalui motor *servo*. Sebaliknya, apabila PIN salah, sistem menolak akses dan menampilkan peringatan. Hal ini menunjukkan bahwa mekanisme autentikasi berbasis keypad berjalan sesuai dengan fungsi keamanan yang dirancang.

4.2.2 Pengujian Sensor DHT 22

Pengujian sensor DHT22 dilakukan untuk mengukur tingkat akurasi sensor dalam membaca suhu dan kelembapan ruangan. Data pengujian diambil secara real-time dan disimpan ke dalam tabel `sensor_data` pada database.

Tabel 4. 2 Pengujian Sensor DHT 22

No	Waktu	Suhu (°C)	Kelembapan (%)	Keterangan
1	11:41:24	23.3	87.2	Berhasil
2	11:41:25	23.4	83.5	Berhasil
3	11:41:38	23.5	83.3	Berhasil
4	11:41:44	23.5	83.2	Berhasil
5	11:41:56	23.5	82.6	Berhasil

Hasil pengujian menunjukkan bahwa sensor DHT22 mampu membaca suhu dan kelembapan secara stabil dengan perubahan nilai yang relatif kecil antar waktu pengukuran. Data sensor berhasil dikirim ke server dan tersimpan di database tanpa adanya kehilangan data. Informasi suhu dan kelembapan ini selanjutnya digunakan oleh sistem sebagai dasar pengambilan keputusan dalam mode otomatis, seperti pengendalian kipas ruangan.

4.2.3 Pengujian Sensor LDR

Pengujian sensor *Light Dependent Resistor* (LDR) dilakukan untuk mengetahui kemampuan sensor dalam mendeteksi perubahan intensitas cahaya di dalam ruang kelas serta memastikan sensor dapat bekerja sesuai dengan perancangan sistem *Smart Class*. Pengujian dilakukan dengan mengamati nilai pembacaan sensor LDR pada kondisi cahaya yang berbeda, yaitu kondisi terang dan kondisi gelap.

Data hasil pengujian sensor LDR diperoleh secara *real-time* melalui mikrokontroler *ESP32 DevKit V1* dan disimpan ke dalam tabel `sensor_data` pada *database*. Nilai intensitas cahaya yang diperoleh kemudian digunakan sebagai dasar pengendalian *LED* putih secara otomatis pada sistem.

Tabel 4. 3 Pengujian Sensor LDR

No	Waktu	Nilai Cahaya (ADC)	Kondisi Cahaya	Status LED	Keterangan
1	11:45:12	2850	Terang	ON	Berhasil
2	11:45:25	2785	Terang	ON	Berhasil
3	11:45:39	1920	Redup	OFF	Berhasil
4	11:45:52	1650	Gelap	OFF	Berhasil

Hasil pengujian menunjukkan bahwa sensor LDR mampu mendeteksi perubahan intensitas cahaya dengan baik. Pada kondisi cahaya terang, nilai ADC yang dihasilkan sensor berada di atas nilai ambang batas sehingga *LED* putih menyala secara otomatis. Sebaliknya, pada kondisi cahaya redup atau gelap, nilai ADC berada di bawah ambang batas dan *LED* akan mati.

Data intensitas cahaya yang diperoleh dari sensor LDR juga berhasil dikirim ke server dan disimpan ke dalam *database* tanpa mengalami kehilangan data. Berdasarkan hasil pengujian tersebut, dapat disimpulkan bahwa sensor LDR berfungsi dengan baik sebagai perangkat input untuk pengendalian pencahayaan ruang kelas secara otomatis pada sistem *Smart Class*.

4.2.4 Pengujian Motor Servo

Pengujian motor servo dilakukan untuk memastikan aktuator dapat bergerak sesuai perintah sistem.

Tabel 4. 4 Pengujian Motor Servo

No	Servo	Parameter Uji	Nilai Sistem	Status Servo	Keterangan
1	Servo Pintu	PIN Benar	door_state = 1	Terbuka	Berhasil
2	Servo Pintu	PIN Salah	door_state = 0	Tertutup	Berhasil
3	Servo Pintu	Mode Auto	mode = auto	Tertutup	Berhasil

Pengujian motor servo dilakukan untuk memastikan aktuator mampu bergerak sesuai dengan perintah sistem. Servo pintu diuji berdasarkan input PIN melalui keypad serta status *door_state* yang tersimpan pada sistem. Ketika PIN dimasukkan dengan benar, servo pintu bergerak membuka, sedangkan ketika PIN salah atau sistem berada pada kondisi terkunci, servo pintu tetap tertutup.

4.2.5 Pengujian Relay dan Fan

Pengujian relay dan fan dilakukan untuk memastikan modul relay mampu mengendalikan kipas sesuai dengan perintah sistem. Pengujian dilakukan dengan mengaktifkan dan menonaktifkan kipas baik pada mode otomatis maupun mode manual melalui dashboard web.

Tabel 4. 5 Pengujian Relay dan Fan

No	Mode Operasi	Kondisi Sensor	Status Fan	Keterangan
1	Otomatis	Suhu Normal	Mati	Berhasil
2	Otomatis	Suhu Tinggi	Menyala	Berhasil
3	Manual	–	Menyala	Berhasil
4	Manual	–	Mati	Berhasil

Pada mode otomatis, relay bekerja berdasarkan data suhu, kelembapan, dan kualitas udara yang dibaca oleh sensor. Kipas menyala ketika nilai sensor melebihi ambang batas yang telah ditentukan dan mati saat kondisi kembali normal. Pada mode manual, kipas dapat dikendalikan langsung melalui web tanpa memperhatikan nilai sensor. Hasil pengujian menunjukkan bahwa relay dan fan dapat bekerja dengan baik sesuai perintah sistem.

4.2.6 Pengujian Liquid Crystal Display 20x4 I2C

Pengujian LCD I2C 20x4 dilakukan untuk memastikan tampilan informasi dapat ditampilkan dengan benar dan stabil. Informasi yang ditampilkan meliputi suhu ruangan, kondisi cuaca, status kipas, serta status pintu.

Tabel 4. 6 Pengujian LCD

No	Informasi Ditampilkan	Hasil Tampilan	Keterangan	No
1	Suhu dan Kelembapan	Sesuai	Berhasil	1
2	Status Fan	Sesuai	Berhasil	2
3	Status Pintu	Sesuai	Berhasil	3

Hasil pengujian menunjukkan bahwa LCD mampu menampilkan seluruh informasi dengan jelas dan sesuai dengan data yang diproses oleh ESP32. Perubahan nilai sensor dan status perangkat dapat langsung diperbarui pada LCD secara real-time. Dengan demikian, LCD I2C 20x4 dinyatakan berfungsi dengan baik sebagai media informasi lokal pada sistem Smart Class.

4.2.7 Pengujian Komunikasi API (GET/POST)

Pengujian komunikasi API dilakukan untuk memastikan proses pertukaran data antara ESP32 dan web server berjalan dengan baik. Metode HTTP GET digunakan untuk mengambil data pengaturan sistem seperti mode operasi dan status perangkat, sedangkan metode HTTP POST digunakan untuk mengirimkan data sensor dan status aktuator ke server.

Tabel 4. 7 Pengujian Komunikasi API (GET/POST)

No	Metode API	Data Dikirim/Diterima	Status	Keterangan
1	GET	Mode Sistem	Berhasil	Data diterima
2	GET	Status Perangkat	Berhasil	Data diterima
3	POST	Data Sensor	Berhasil	Data tersimpan
4	POST	Status Aktuator	Berhasil	Data tersimpan

Hasil pengujian menunjukkan bahwa ESP32 berhasil melakukan komunikasi dua arah dengan server. Data yang dikirim menggunakan metode POST dapat tersimpan di database dan ditampilkan pada dashboard web, sedangkan data yang diambil menggunakan metode GET dapat diterima dan dijalankan oleh sistem dengan benar. Berdasarkan hasil tersebut, komunikasi API pada sistem Smart Class dinyatakan berjalan dengan baik.

4.3 Hasil Pengujian Perangkat Lunak

Berdasarkan seluruh rangkaian pengujian yang telah dilakukan pada sistem Smart Class berbasis *Internet of Things (IoT)* menggunakan *ESP32 DevKit V1*, dapat dianalisis bahwa perangkat lunak yang dikembangkan mampu bekerja sesuai dengan perancangan sistem. Pengujian difokuskan pada kinerja program, proses pembacaan data sensor, pengolahan logika kendali, komunikasi dengan server, serta integrasi dengan dashboard web.

Hasil pengujian menunjukkan bahwa program pada ESP32 mampu membaca data dari sensor DHT22 secara konsisten dan stabil. Nilai suhu dan kelembapan yang diterima diproses oleh sistem sebagai parameter dalam pengambilan keputusan otomatis. Logika program berjalan sesuai dengan algoritma yang telah dirancang, di mana ketika nilai suhu atau kelembapan melebihi batas ambang yang ditentukan, sistem secara otomatis mengaktifkan kipas melalui relay. Sebaliknya, ketika nilai kembali normal, sistem mematikan kipas secara otomatis. Hal ini menunjukkan bahwa proses pengolahan data sensor dan eksekusi logika kendali berjalan dengan baik tanpa error.

Pengujian pembacaan sensor Light Dependent Resistor (LDR) juga menunjukkan bahwa perangkat lunak mampu mengolah nilai analog menjadi parameter kendali pencahayaan. Program secara otomatis membandingkan nilai intensitas cahaya dengan batas threshold yang telah ditentukan. Berdasarkan hasil pengujian, sistem mampu mengaktifkan atau menonaktifkan LED sesuai kondisi cahaya ruangan. Proses ini berlangsung secara real-time dan tidak ditemukan kegagalan dalam eksekusi logika program.

Pada pengujian fitur autentikasi keypad 4×4, perangkat lunak berhasil membaca input PIN dari pengguna dan membandingkannya dengan data yang tersimpan pada server. Proses validasi dilakukan melalui komunikasi data berbasis HTTP. Ketika PIN yang dimasukkan sesuai, sistem menjalankan perintah pembukaan pintu serta menampilkan informasi akses pada LCD. Sebaliknya, ketika PIN tidak sesuai, sistem menolak akses dan kembali ke kondisi awal. Hasil ini menunjukkan bahwa algoritma autentikasi dan pengelolaan data akses berjalan sesuai dengan rancangan.

Pengujian kontrol perangkat melalui dashboard web menunjukkan bahwa sistem mampu menerima perintah dari antarmuka web menggunakan metode HTTP POST dan HTTP GET. Perintah yang dikirimkan dari dashboard berhasil diterima dan diproses oleh ESP32 tanpa kehilangan data. Status perangkat yang diperbarui juga berhasil dikirim kembali ke server dan ditampilkan secara real-time pada dashboard. Hal ini membuktikan bahwa komunikasi antara perangkat IoT dan sistem berbasis web berjalan stabil dan responsif.

Selain itu, pengujian mode otomatis dan manual menunjukkan bahwa perangkat lunak mampu mengatur prioritas logika kendali dengan baik. Pada mode otomatis, sistem bekerja berdasarkan pembacaan sensor dan algoritma yang telah ditentukan. Pada mode manual, kontrol sistem dialihkan sepenuhnya ke pengguna melalui dashboard web. Proses pergantian mode berjalan tanpa konflik logika maupun gangguan pada sistem.

Selama proses pengujian berulang, program yang berjalan pada ESP32 tidak mengalami crash, hang, maupun restart mendadak. Loop program berjalan stabil dengan waktu respon rata-rata satu hingga dua detik tergantung kondisi jaringan internet. Tidak ditemukan error komunikasi yang signifikan selama pengujian berlangsung.

Secara keseluruhan, hasil pengujian perangkat lunak menunjukkan bahwa sistem Smart Class berbasis IoT telah mampu menjalankan fungsi monitoring, pengendalian otomatis, autentikasi akses, serta komunikasi data berbasis web sesuai dengan tujuan penelitian. Integrasi antara logika program pada ESP32 dan sistem web service berjalan dengan baik, sehingga perangkat lunak dapat dinyatakan berfungsi secara optimal dan siap diimplementasikan pada lingkungan ruang kelas.

4.4 Kelebihan Sistem

Sistem *Smart Class* yang dirancang memiliki beberapa kelebihan yang mendukung penerapannya dalam lingkungan ruang kelas. Salah satu kelebihan utama sistem ini adalah kemampuannya dalam mengintegrasikan perangkat input dan aktuator ke dalam satu sistem kendali yang terpusat menggunakan mikrokontroler *ESP32*. Integrasi ini memungkinkan proses monitoring dan pengendalian perangkat dapat dilakukan secara efisien dan terkoordinasi.

Sistem mampu melakukan monitoring kondisi ruang kelas berupa suhu dan kelembapan secara *real-time* serta mengendalikan perangkat kipas baik secara otomatis maupun manual melalui *dashboard web*. Pada mode otomatis, sistem dapat menyesuaikan kondisi lingkungan ruang kelas berdasarkan parameter sensor, sedangkan pada mode manual pengguna dapat melakukan pengendalian perangkat secara langsung melalui antarmuka web. Hal ini memberikan fleksibilitas dalam pengoperasian sistem sesuai dengan kebutuhan pengguna.

Penggunaan mikrokontroler *ESP32* sebagai pusat kendali memungkinkan sistem terhubung dengan jaringan internet sehingga proses pertukaran data dengan *server* dapat dilakukan secara *real-time*. Selain itu, penerapan sistem keamanan akses pintu menggunakan *keypad* memberikan nilai tambah dalam mengontrol akses masuk ke ruang kelas. Sistem juga dirancang dalam bentuk *prototype*, sehingga memudahkan pengembangan dan penyesuaian fitur di masa mendatang sesuai dengan kebutuhan pengguna.

4.5 Kekurangan Sistem

Meskipun sistem *Smart Class* yang dirancang memiliki berbagai kelebihan, sistem ini juga memiliki beberapa keterbatasan yang perlu diperhatikan. Salah satu kekurangan sistem adalah keterbatasan parameter monitoring lingkungan, di mana sistem hanya memantau suhu dan kelembapan ruang kelas. Kondisi lingkungan lain yang juga dapat mempengaruhi kenyamanan ruang kelas belum diakomodasi dalam sistem ini.

Ketergantungan sistem terhadap koneksi jaringan internet juga menjadi salah satu keterbatasan. Apabila koneksi *WiFi* mengalami gangguan atau tidak tersedia, maka proses komunikasi antara *ESP32* dan *server* tidak dapat berjalan secara optimal. Kondisi ini dapat mempengaruhi fungsi monitoring dan pengendalian perangkat melalui *dashboard web*, khususnya pada mode manual.

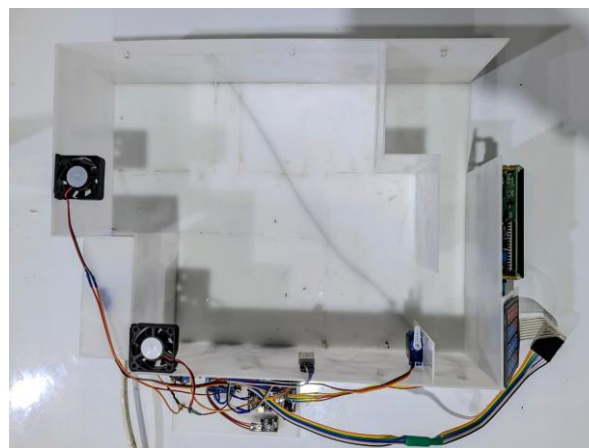
Selain itu, sistem keamanan yang digunakan masih terbatas pada autentikasi PIN melalui *keypad*. Meskipun metode ini cukup efektif untuk mengontrol akses pintu, tingkat keamanan sistem masih dapat ditingkatkan dengan menambahkan metode autentikasi lain agar lebih aman dan fleksibel. Sistem juga masih berupa *prototype*, sehingga belum sepenuhnya diuji untuk penggunaan jangka panjang dan kondisi lingkungan yang beragam.

Keterbatasan lain dari sistem ini adalah belum adanya mekanisme pencatatan kesalahan (*error handling*) dan notifikasi secara otomatis apabila terjadi gangguan pada perangkat keras atau jaringan. Oleh karena itu, pemantauan kondisi sistem masih memerlukan pengawasan dari pengguna melalui *dashboard web*.

4.6 Foto Sistem

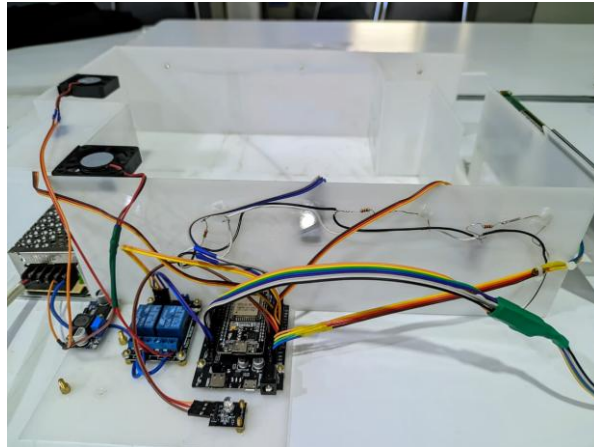
Bagian ini menampilkan dokumentasi berupa foto prototype Sistem Smart Class berbasis Internet of Things yang telah dirancang dan diimplementasikan. Foto sistem digunakan untuk menunjukkan bentuk fisik prototype serta susunan komponen perangkat keras yang digunakan dalam sistem.

Melalui dokumentasi ini, dapat dilihat integrasi antara mikrokontroler ESP32, sensor, dan aktuator yang terpasang pada prototype. Foto sistem juga berfungsi sebagai bukti implementasi perangkat keras yang mendukung proses pengujian dan evaluasi sistem Smart Class yang telah dilakukan.



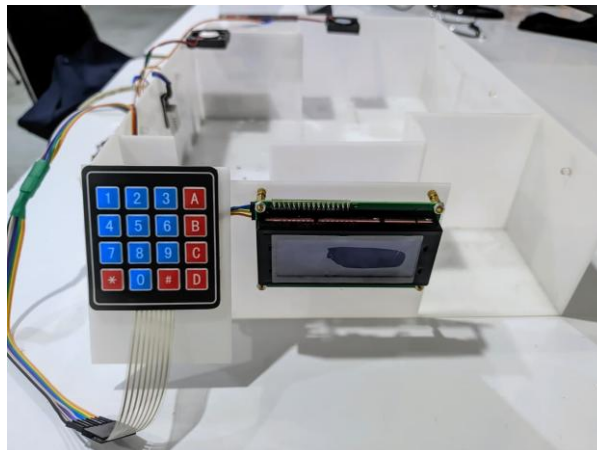
Gambar 4. 2 Tampak atas

Gambar 4.2 menunjukkan prototype Smart Class berbasis IoT yang memperlihatkan susunan komponen di dalam box, meliputi kipas sebagai simulasi sirkulasi udara, rangkaian kendali berbasis ESP32 DevKit V1, modul relay, serta jalur pengkabelan yang terintegrasi.



Gambar 4. 3 Tampak samping

Gambar 4.3 menunjukkan prototype Smart Class dengan penempatan mikrokontroler ESP32 DevKit V1, modul relay, dan pengkabelan antar komponen sebagai bagian dari sistem kendali dan monitoring.



Gambar 4. 4 Tampak Depan

Gambar 4.4 menunjukkan prototype Smart Class yang menampilkan modul keypad sebagai sistem keamanan akses pintu dan modul LCD sebagai media penampil informasi kondisi sistem.

BAB V PENUTUP

5.1 Kesimpulan

Berdasarkan seluruh tahapan yang telah dilalui, mulai dari perancangan sistem, proses implementasi, hingga pengujian yang dilakukan, dapat disimpulkan bahwa sistem Smart Class berbasis Internet of Things (IoT) yang dikembangkan pada penelitian ini telah berhasil menjawab rumusan masalah yang diajukan.

Proses perancangan dan pembangunan prototype Smart Class menunjukkan bahwa berbagai perangkat keras dapat diintegrasikan ke dalam satu sistem yang terpusat dan saling terhubung dengan baik. Sistem mampu melakukan monitoring kondisi ruang kelas, yang meliputi suhu, kelembapan udara, dan intensitas cahaya ruangan. Sensor DHT22 digunakan untuk memperoleh data suhu dan kelembapan, sedangkan sensor Light Dependent Resistor (LDR) digunakan untuk mendeteksi tingkat pencahayaan ruangan. Selain fungsi monitoring, sistem juga mampu mengendalikan fasilitas ruang kelas seperti kipas, lampu, dan pintu secara terkoordinasi. Hasil pengujian menunjukkan bahwa seluruh komponen dapat berfungsi sesuai dengan perancangan dan bekerja secara terintegrasi dalam satu sistem.

Penggunaan mikrokontroler ESP32 DevKit V1 sebagai pusat kendali terbukti efektif dalam mengolah data sensor dan mengendalikan perangkat ruang kelas. ESP32 mampu melakukan komunikasi dua arah dengan server melalui metode HTTP GET dan HTTP POST, sehingga data sensor dan status perangkat dapat dikirim, disimpan ke dalam database, serta ditampilkan secara real-time melalui dashboard web. Hal ini menunjukkan bahwa sistem monitoring dan pengendalian berbasis web dapat diimplementasikan dengan baik pada sistem Smart Class ini.

Penerapan dua mode pengoperasian, yaitu mode otomatis dan mode manual, juga dapat berjalan sesuai dengan tujuan perancangan. Pada mode otomatis, sistem mampu mengambil keputusan berdasarkan nilai sensor, seperti menyalakan kipas ketika suhu atau kelembapan melebihi batas yang telah ditentukan serta mengendalikan lampu berdasarkan intensitas cahaya ruangan. Lampu akan menyala ketika kondisi ruangan gelap dan akan mati ketika kondisi ruangan terang. Sementara itu, pada mode manual, pengguna diberikan kebebasan untuk mengendalikan perangkat secara langsung melalui dashboard web tanpa bergantung pada nilai sensor. Keberadaan dua mode ini memberikan fleksibilitas dalam pengelolaan fasilitas ruang kelas sesuai dengan kebutuhan pengguna.

Dari sisi keamanan, sistem akses pintu yang menggunakan keypad sebagai media input PIN dapat berfungsi dengan baik dalam membatasi akses masuk ke ruang kelas. Proses autentikasi berjalan sesuai dengan logika sistem dan terintegrasi dengan pengendalian motor servo sebagai mekanisme buka dan tutup pintu.

Secara keseluruhan, sistem Smart Class berbasis IoT yang dikembangkan mampu meningkatkan efisiensi dalam monitoring dan pengendalian fasilitas ruang kelas. Prototype ini menunjukkan bahwa penerapan teknologi Internet of Things dapat digunakan sebagai solusi terintegrasi dalam pengelolaan ruang kelas, serta dapat dijadikan dasar untuk pengembangan sistem Smart Class yang lebih lanjut dan berskala lebih luas.

5.2 Saran

Berdasarkan hasil penelitian yang telah dilakukan, terdapat beberapa saran yang dapat diberikan untuk pengembangan sistem ke depannya, yaitu:

1. Sistem dapat dikembangkan dengan menambahkan mekanisme penyimpanan data lokal sebagai cadangan apabila koneksi internet mengalami gangguan.
2. Pengujian sistem dapat dilakukan secara langsung pada ruang kelas sebenarnya untuk memperoleh hasil yang lebih akurat.
3. Antarmuka dashboard web dapat dikembangkan agar lebih informatif dan mudah digunakan.
4. Penambahan sensor atau fitur lain dapat dilakukan untuk meningkatkan fungsi dan keandalan sistem Smart Class.

DAFTAR PUSTAKA

- Aswaldi, H. (2025). Penerapan Teknologi Internet of Things (IoT) untuk Monitoring Kualitas Udara dalam Ruangan Article history. *Journal of Computer Science and Information Technology*, 1(2). <https://doi.org/10.70716/jocsit.v1i2.225>
- Budiyanto, A., Pramudita, G. B., Adinandra, S., Studi, P., & Elektro, T. (2020). Kontrol Relay dan Kecepatan Kipas Angin Direct Current (DC) dengan Sensor Suhu LM35 Berbasis Internet of Things (IoT). *Jurnal Ilmiah Elektroteknika*, 19, 43–54.
- Espressif Systems. (2023). *ESP32 Series Datasheet Version 5.2 2.4 GHz Wi-Fi + Bluetooth® + Bluetooth LE SoC Including*. www.espressif.com
- Fauziman, H., & Mukhaiyar, R. (2023). Rancang Bangun Sistem Keamanan Pintu Menggunakan Fingerprint Berbasis Internet Of Things (IoT). *JTEIN: Jurnal Teknik Elektro Indonesia*, 4(2). <https://doi.org/10.24036/jtein.v4i2.438>
- Irdayanti, Y., Alrasyid, J., & Azzahra, A. (2025). *Sistem Monitoring dan Pengaturan Kualitas Udara Dalam Ruangan Berbasis Internet of Things*. 10(1). <https://doi.org/10.31851/ampere>
- Kalbuana, N., & Kurnianto, B. (2024). Desain Sistem Deteksi Asap Berbasis Sensor Mikrokontroler Sebagai Upaya Pencegahan Kebakaran. *MALCOM: Indonesian Journal of Machine Learning and Computer Science*, 4(1), 266–272. <https://doi.org/10.57152/malcom.v4i1.1158>
- Latief, M. F., Irmansyah, I., & Rosyidi, L. (2025). Sistem Pemantauan Ruang Kelas berbasis Internet of Things (IoT) untuk Proses Pendidikan yang Efektif. *Digital Transformation Technology*, 4(2), 1278–1284. <https://doi.org/10.47709/digitech.v4i2.5485>
- Made, N., Astari, A. J., Ardiyasa, W., Made, I., Gautama, B., Susiriyanti, P., Teknologi, I., Bisnis, D., & Bali, S. (2023). Pengumpulan Data Suhu dan Kelembapan Ruangan Menggunakan Embedded System Berbasis IoT. In *Sains Teknik Elektro* (Vol. 4, Number 2). <http://jurnal.bsi.ac.id/index.php/insantek47>
- N Somayasa, D. M., Nursalam, M. N., Taslimah, I. I., Sutiari, D. K., & Muhammad Abidin, dan Z. (2024). PROTOTIPE PENGONTROLAN NYALA DAN PADAMNYA LAMPU BERBASIS IOT (INTERNET OF THINGS). *Jurnal Nasional Hasil Penelitian Bidang Multidisiplin*, 1, 1–14. <https://doi.org/10.30598/jnbcxxxxxxxxxxxxxx>
- Nur, A., & Sari, E. (2021). SMART MONITORING SYSTEM SUHU RUANGAN SERVER BERBASIS IOT MENGGUNAKAN METODE PROTOKOL

KOMUNIKASI HTTP PADA STMIK MARDIRA INDONESIA BANDUNG.
In *Jurnal* (Vol. 2, Number 1).

- Ontowirjo Y.Q Fauzah, & Poekoel C. Vecky. (2018). Implementasi Internet of Things Pada Sistem Monitoring Suhu dan Kelembaban Pada Ruang Pengereng Berbasis Web. *Jurnal Teknik Elektro Dan Komputer*, 7, 331–338. www.cec-unsrat.com.
- Pramudita, R., & Setyawan, K. (2022). Sistem Smart Class Berbasis Internet Of Things Dengan Menggunakan Metode Prototype. *SMARTICS Journal*, 8(1). <https://doi.org/10.21067/smartics.v8i1.7209>
- Putri Ida Sunaryathy Samad. (2024). Rancang Bangun Alat Monitoring Ruang Berbasis IoT Dan MQTT. *JETC, Volume 9, Nomor 1*.
- Putri, J. W. D., Firdaus, J. R., Khaerunnisa, L. S. K., & Sobur, S. (2025). SISTEM MONITORING SUHU, KELEMBAPAN DAN GAS BERBASIS IOT PADA LAB KOMPUTER. *Jurnal Informatika Dan Teknik Elektro Terapan*, 13(3S1). <https://doi.org/10.23960/jitet.v13i3S1.8175>
- Ridla, M. A., & Rahman, M. F. (2024). Perancangan Prototype Monitoring Suhu Berbasis Internet Of Things (IoT). *JUSIFOR : Jurnal Sistem Informasi Dan Informatika*, 3(1), 72–79. <https://doi.org/10.33379/jusifor.v3i1.4367>
- Riyanto, J., Nurlaila, F., Haerudin, H., & Jarastino, B. T. (2021). Rancang Bangun Sistem Monitoring Ruang Kelas Berbasis Internet of Things pada Universitas Pamulang. *Jurnal Informatika Universitas Pamulang*, 5(4), 483. <https://doi.org/10.32493/informatika.v5i4.7018>
- Samsugi, S., Mardiyansyah, Z., & Nurkholis, A. (2020). SISTEM PENGONTROL IRIGASI OTOMATIS MENGGUNAKAN MIKROKONTROLER ARDUINO UNO. In *JTST* (Vol. 01, Number 01).
- Sari, I. P., Apdilah, D., & Guntur, S. (2025). Sistem Smart Class Berbasis Internet of Things (IoT). *Sudo Jurnal Teknik Informatika*, 4(1), 33–39. <https://doi.org/10.56211/sudo.v4i1.778>
- Sholeha, D., Sri Rezeki, N., Sinaga, J., & Tarigan, K. (2025). Impression: Jurnal Teknologi dan Informasi Penerapan Media Pembelajaran Sistem Monitoring Energi Listrik Berbasis Internet of Things (IoT) Menggunakan ESP32. In *Jurnal Teknologi dan Informas* (Vol. 4, Number 3).
- Susilania. (2025). Perancangan Sistem Monitoring Berbasis Iot Untuk Peningkatan Kepatuhan Aturan Kecepatan Dalam Berkendara di Area Tambang. *Jurnal Ilmu Komputer*, 1, 145–158.
- Tiyas, A. W., Erwanto, D., & Yanuartanti, I. (2025). Peningkatan Akurasi Sensor Suhu dan Kelembaban DHT11 dengan Kalibrasi Suhu Berbasis IoT pada Platform Thingspeak. *Jurnal Pendidikan Dan Teknologi Indonesia*, 5(3), 625–633. <https://doi.org/10.52436/1.jpti.709>

Yunardi, D. H., Misbullah, A., & Gemilang, D. G. (2023). RANCANG BANGUN SISTEM WEB MONITORING KUALITAS UDARA DI DALAM RUANGAN MENGGUNAKAN LOGIKA FUZZY. *Jurnal Pendidikan Teknologi Informasi*, 7(1), 25–34.

<https://www.theengineeringprojects.com/2018/10/introduction-to-arduino-ide.html>

<https://www.ardutech.com/mengenal-esp32-development-kit-untuk-iot-internet-of-things/>

<https://jagorobotik.com/blog/2025/06/28/pengenalan-modul-dht22-sensor-suhu-dan-kelembaban-andal-untuk-proyek-elektronikmu/>

<https://nusabot.id/blog/menggunakan-input-ldr/>

<https://www.aksesoriskomputerlampung.com/2019/07/keypad-membran-4x4-arduino.html>

<https://www.edukasiaelektronika.com/2020/12/motor-servo-sg90.html>

<https://h5.lazada.co.id/products/kipas-dc-12v-4x4-cm-cooling-fan-12v-dc-kipas-mini-pendingin-4-x-4cm-i829118165.html>

<https://1ohm.in/product/2-channel-5v-relay-module-for-arduino-with-optocoupler-2/>

https://soldered.com/learn/led-light-emitting-diode-explained/?srsltid=AfmBOorbb_nOXm67IhLiKW8gw3m1nPeYfhdobGNMhYfBUoCPE5ccVc0

https://www.sunfounder.com/products/i2c-lcd2004-module?srsltid=AfmBOoqs1_eZLHG9p0SxR9ZcwaCK3uuHkYC0fQ3thK9cuqLeqIJpExE

<https://www.codepolitan.com/blog/ingin-bikin-website-sendiri-ini-dia-rahasia-dibaliknya-ipk88n/>

LAMPIRAN

a. Listing Program

```
#include <WiFi.h>
#include <HTTPClient.h>
#include <ArduinoJson.h>
#include <WiFiManager.h>
#include <Keypad.h>
#include <ESP32Servo.h>
#include <LiquidCrystal_I2C.h>
#include "DHT.h"

/* ===== LCD ===== */
LiquidCrystal_I2C lcd(0x27, 20, 4);

/* ===== KEYPAD ===== */
const byte ROWS = 4;
const byte COLS = 4;

char keys[ROWS][COLS] = {
  {'1', '2', '3', 'A'},
  {'4', '5', '6', 'B'},
  {'7', '8', '9', 'C'},
  {'*', '0', '#', 'D'}
};

byte rowPins[ROWS] = {13, 12, 14, 27};
byte colPins[COLS] = {26, 25, 33, 32};

Keypad keypad = Keypad(makeKeymap(keys), rowPins,
colPins, ROWS, COLS);

/* ===== SERVO ===== */
Servo servoPintu;
int pinServoPintu = 4;

/* ===== OUTPUT ===== */
int fan1 = 16;
int fan2 = 17;
int ledPutih = 3;

/* ===== SENSOR ===== */
#define DHTPIN 18
#define DHTTYPE DHT22
```

```

DHT dht(DHTPIN, DHTTYPE);

int ldrPin = 15;
int ldrValue = 0;

/* ===== STATE ===== */
String inputPass = "";
String correctPass = "";

float suhu = 0;
float kelembapan = 0;

String currentMode = "auto";

int stateFan1 = 0;
int stateFan2 = 0;
int stateDoor = 0;

/* ===== TIMER ===== */
unsigned long lastHttpGet = 0;
unsigned long lastHttpSend = 0;

unsigned long doorOpenTime = 0;
bool doorIsOpening = false;

WiFiManager wm;

/* ===== LCD ===== */
void lcdStandby() {
    lcd.clear();
    lcd.setCursor(0,0); lcd.print("SMART CLASS");
    lcd.setCursor(0,1); lcd.print("PIN : ****");
}

void updateLCD() {
    lcd.setCursor(0,2);
    lcd.print("Suhu : "); lcd.print(suhu); lcd.print("
C ");

    lcd.setCursor(0,3);
    lcd.print("Lembab: "); lcd.print(kelembapan);
    lcd.print("% ");
}

/* ===== SENSOR ===== */

```

```

void bacaSensor() {
  float t = dht.readTemperature();
  float h = dht.readHumidity();

  if (!isnan(t)) suhu = t;
  if (!isnan(h)) kelembapan = h;

  ldrValue = analogRead(ldrPin);
}

/* ===== AUTO LOGIC ===== */
void logikaAuto() {
  stateFan1 = (suhu > 30 || kelembapan > 80);
  stateFan2 = stateFan1;

  // Logika LED berdasarkan cahaya
  // Terang (siang) → LED ON
  // Gelap (malam) → LED OFF
  if (ldrValue > 2000) {
    digitalWrite(ledPutih, HIGH);
  } else {
    digitalWrite(ledPutih, LOW);
  }
}

/* ===== OUTPUT ===== */
void applyOutputs() {
  digitalWrite(fan1, stateFan1);
  digitalWrite(fan2, stateFan2);
  servoPintu.write(stateDoor ? 90 : 0);
}

/* ===== KEYPAD ===== */
void prosesPIN() {
  char key = keypad.getKey();
  if (!key) return;

  if (key == 'B') {
    inputPass = "";
    lcdStandby();
    return;
  }

  if (key == 'A') {
    lcd.clear();
  }
}

```

```

    if (inputPass == correctPass && correctPass.length()
    >= 4) {
        lcd.setCursor(0,0); lcd.print("PIN BENAR");
        lcd.setCursor(0,1); lcd.print("PINTU DIBUKA");

        stateDoor = 1;
        doorIsOpening = true;
        doorOpenTime = millis();
    } else {
        lcd.setCursor(0,0); lcd.print("PIN SALAH");
        lcd.setCursor(0,1); lcd.print("COBA LAGI");
        delay(1500);
    }

    inputPass = "";
    lcdStandby();
    return;
}

if (isDigit(key) && inputPass.length() < 4) {
    inputPass += key;
}

lcd.setCursor(0,1);
lcd.print("PIN : ");
for (int i=0;i<inputPass.length();i++) lcd.print("*");
for (int i=inputPass.length();i<4;i++) lcd.print("_");
}

/* ===== HTTP GET ===== */
void fetchRemoteStatus() {
    if (WiFi.status() != WL_CONNECTED) return;

    HTTPClient http;
    http.begin("http://smarthomerahmat.web.id/api/dashboar
d.php");

    if (http.GET() == 200) {
        StaticJsonDocument<1024> doc;
        deserializeJson(doc, http.getString());

        currentMode = String(doc["status"]["mode"] |
"auto");
        stateFan1 = doc["status"]["fan1"] | stateFan1;

```

```

        stateFan2 = doc["status"]["fan2"] | stateFan2;
        stateDoor = doc["status"]["door_state"] |
stateDoor;
    }
    http.end();
}

void fetchKeypadPassword() {
    if (WiFi.status() != WL_CONNECTED) return;

    HTTPClient http;
    http.begin("http://smarthomerahmat.web.id/api/keypad.p
hp");

    if (http.GET() == 200) {
        StaticJsonDocument<512> doc;
        deserializeJson(doc, http.getString());
        const char* p = doc["password"];
        if (p && strlen(p) >= 4) correctPass = String(p);
    }
    http.end();
}

/* ===== HTTP POST ===== */
void sendDataToServer() {
    if (WiFi.status() != WL_CONNECTED) return;

    HTTPClient http;
    http.begin("http://smarthomerahmat.web.id/api/create.p
hp");
    http.addHeader("Content-Type", "application/json");

    String json = "{";
    json += "\"suhu\":" + String(suhu) + ",";
    json += "\"kelembapan\":" + String(kelembapan) + ",";
    json += "\"cahaya\":" + String(ldrValue) + ",";
    json += "\"fan1\":" + String(stateFan1) + ",";
    json += "\"fan2\":" + String(stateFan2) + ",";
    json += "\"door_state\":" + String(stateDoor) + ",";
    json += "\"mode\":" + currentMode + "\"";
    json += "}";

    http.POST(json);
    http.end();
}

```

```

/* ===== SETUP ===== */
void setup() {
  Serial.begin(115200);

  lcd.init();
  lcd.backlight();
  lcdStandby();

  servoPintu.attach(pinServoPintu);

  pinMode(fan1, OUTPUT);
  pinMode(fan2, OUTPUT);
  pinMode(ledPutih, OUTPUT);
  pinMode(ldrPin, INPUT);

  dht.begin();

  WiFi.mode(WIFI_STA);
  wm.autoConnect("SmartClass", "12345678");
}

/* ===== LOOP ===== */
void loop() {
  prosesPIN();
  bacaSensor();

  unsigned long now = millis();

  if (now - lastHttpGet >= 2000) {
    lastHttpGet = now;
    fetchRemoteStatus();
    fetchKeypadPassword();
  }

  if (currentMode == "auto") {
    logikaAuto();
  }

  if (doorIsOpening && millis() - doorOpenTime >= 3000)
  {
    stateDoor = 0;
    doorIsOpening = false;
  }
}

```

```

applyOutputs();
updateLCD();

if (now - lastHttpSend >= 3000) {
    lastHttpSend = now;
    sendDataToServer();
}
}

```

b. Listing Program Dashboard

```

<?php
include "config/db.php";
include "layout/header.php";

$data = mysqli_fetch_assoc(
    mysqli_query($conn,"SELECT * FROM sensor_data ORDER BY
id DESC LIMIT 1")
);
?>

<style>
.cards{
    display:grid;
    grid-template-columns:repeat(4,1fr);
    gap:24px;
    margin-bottom:30px;
}

.section{
    display:grid;
    grid-template-columns:1fr;
    gap:30px;
}

.cards .card{
    min-height:160px;
    display:flex;
    flex-direction:column;
    justify-content:center;
}

.value{
    font-size:32px;
    font-weight:700;
    margin-top:12px;
}

```

```

}

@media(max-width:1200px){
  .cards{
    grid-template-columns:repeat(2,1fr);
  }
}

@media(max-width:768px){
  .cards{
    grid-template-columns:1fr;
  }
}
</style>

<div class="cards">
  <div class="card">
    <h4>Suhu Ruangan</h4>
    <div class="value">
      <?=$data['suhu'] ?? '-' ?> &deg;C
    </div>
  </div>

  <div class="card">
    <h4>Kelembapan</h4>
    <div class="value">
      <?=$data['kelembapan'] ?? '-' ?> %
    </div>
  </div>

  <div class="card">
    <h4>Status Pintu</h4>
    <div class="value">
      <?=( $data['door_state'] == 1) ? 'TERBUKA' :
'TERTUTUP' ?>
    </div>
  </div>

  <div class="card">
    <h4>Mode Sistem</h4>
    <div class="value">
      <?=$strtoupper($data['mode'] ?? '-') ?>
    </div>
  </div>
</div>

```

```

<div class="section">
  <div class="box">
    <h3>Status Perangkat</h3>

    <p>Fan 1 :
      <b><?=( $data['fan1'] == 1) ? 'ON' : 'OFF' ?></b>
    </p>

    <p>Fan 2 :
      <b><?=( $data['fan2'] == 1) ? 'ON' : 'OFF' ?></b>
    </p>

    <p>Pintu :
      <b><?=( $data['door_state'] == 1) ? 'TERBUKA' :
'TERTUTUP' ?></b>
    </p>
  </div>
</div>

<?php include "layout/footer.php"; ?>

```

c. Listing Program Kontrol

```

<?php
include "../config/db.php";
include "../layout/header.php";

/* Ambil status sistem */
$q = mysqli_query($conn,"SELECT * FROM status_system
WHERE id=1");
$status = mysqli_fetch_assoc($q);
?>

<style>
/* ===== CSS KHUSUS KONTROL ===== */

.control-grid{
  display:grid;
  grid-template-columns:repeat(2,1fr);
  gap:30px;
}

/* MODE BOX */
.mode-box{
  margin-bottom:30px;

```

```

}

.mode-box select{
  width:100%;
  padding:14px;
  font-size:16px;
  border-radius:14px;
  border:1px solid #c7d2fe;
}

/* KONTROL */
.control-box{
  background:white;
  border-radius:22px;
  padding:30px;
  box-shadow:0 15px 35px rgba(0,0,0,.08);
}

.control-row{
  display:flex;
  justify-content:space-between;
  align-items:center;
  margin:18px 0;
  font-size:16px;
}

/* SLIDER */
.switch{
  position:relative;
  width:56px;
  height:28px;
}
.switch input{display:none}
.slider{
  position:absolute;
  inset:0;
  background:#cbd5f5;
  border-radius:30px;
  cursor:pointer;
  transition:.3s;
}
.slider:before{
  content:"";
  position:absolute;
  width:22px;

```

```

height:22px;
background:white;
border-radius:50%;
left:3px;
top:3px;
transition:.3s;
}
.switch input:checked + .slider{
background:#22c55e;
}
.switch input:checked + .slider:before{
transform:translateX(28px);
}

/* DISABLED */
.disabled{
opacity:.5;
pointer-events:none;
}

/* RESPONSIVE */
@media(max-width:768px){
.control-grid{
grid-template-columns:1fr;
}
}
</style>

<!-- ===== MODE SISTEM ===== -->
<div class="control-box mode-box">
<h3>Mode Sistem</h3>
<select onchange="setMode(this.value)">
<option value="auto" <?=$status['mode']=="auto"? "selected":"" ?>>Auto</option>
<option value="manual" <?=$status['mode']=="manual"? "selected":"" ?>>Manual</option>
</select>
</div>

<!-- ===== KONTROL MANUAL ===== -->
<div class="control-box <?=$status['mode']=="auto"? 'disabled':'' ?>">
<h3>Kontrol Manual</h3>

```

```

<?php
$device = [
    'fan1'=>'Lampu',
    'fan2'=>'Fan 2',
    'lampu'=>'Fan 1',
    'door_state'=>'Pintu',
];

foreach($device as $key=>$label){
    $checked = !empty($status[$key]) ? 'checked' : '';
    echo "
    <div class='control-row'>
        <span>$label</span>
        <label class='switch'>
            <input type='checkbox' $checked
                onchange=\"toggle('$key',this.checked)\">
            <span class='slider'></span>
        </label>
    </div>";
}
?>
</div>

<script>
function setMode(mode){
    fetch('../api/update.php',{
        method:'POST',
        headers:{'Content-Type':'application/x-www-form-
urlencoded'},
        body:'mode='+mode
    }).then(()=>{
        setTimeout(()=>location.reload(),300);
    });
}

function toggle(dev,val){
    fetch('../api/update.php',{
        method:'POST',
        headers:{'Content-Type':'application/x-www-form-
urlencoded'},
        body:`device=${dev}&value=${val?1:0}`
    });
}
</script>

```

```
<?php include "../layout/footer.php"; ?>
```

d. Listening Program Keypad

```
<?php
header("Content-Type: application/json");
include "../config/db.php";

/*
  Ambil password keypad terakhir
  (sesuaikan nama tabel & kolom kalau beda)
*/
$q = mysqli_query(
    $conn,
    "SELECT password FROM keypad ORDER BY id DESC LIMIT
1"
);

$data = mysqli_fetch_assoc($q);

echo json_encode([
    "password" => $data['password'] ?? ""
]);
```

e. Listening Program Upgrade

```
<?php
include "../config/db.php";

/* ===== UPDATE MODE ===== */
if(isset($_POST['mode'])){
    $mode = $_POST['mode']=='manual' ? 'manual' : 'auto';

    mysqli_query($conn,"
        UPDATE status_system
        SET mode='$mode'
        WHERE id=1
    ");

    echo "MODE UPDATED";
    exit;
}

/* ===== UPDATE DEVICE ===== */
if(isset($_POST['device']) && isset($_POST['value'])){
    $device = $_POST['device'];
    $value = $_POST['value']=='1' ? 1 : 0;
```

```
// whitelist kolom biar aman
$allowed = [
    'fan1',
    'fan2',
    'lampu',
    'door_state',
    'jemuran_state'
];

if(in_array($device,$allowed)){
    mysqli_query($conn,"
        UPDATE status_system
        SET $device=$value
        WHERE id=1
    ");
    echo "DEVICE UPDATED";
}else{
    echo "INVALID DEVICE";
}
exit;
}

echo "NO ACTION";
```