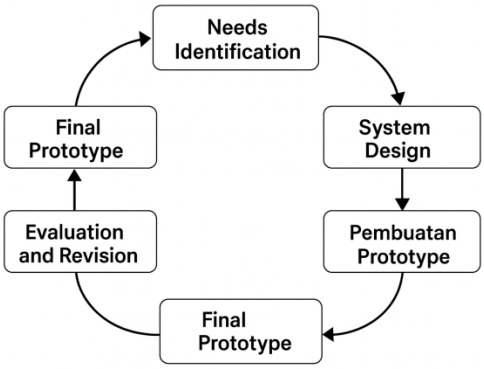
BAB III METODE PENELITIAN

Penelitian ini mengadopsi metode Prototyping karena pengembangan Smart Classroom berbasis IoT melibatkan penyempurnaan komponen perangkat keras dan perangkat lunak secara terus-menerus. Hal ini tidak dapat ditangani secara optimal oleh model linier seperti SDLC Waterfall. Model Waterfall mengharuskan semua kebutuhan didefinisikan secara penuh sejak awal, yang mana tidak realistis dalam pengembangan IoT di mana perilaku sensor, stabilitas jaringan, kalibrasi komponen, dan masalah aliran data sering kali baru muncul setelah pengujian praktis. Sebaliknya, prototyping mendukung visualisasi dini, memungkinkan sistem diuji dalam bentuk parsial sebelum implementasi akhir. Hal ini memudahkan identifikasi cepat terhadap masalah seperti konektivitas Wi-Fi yang tidak stabil, penundaan sensor (misalnya waktu pemanasan MQ-2), nilai ambang batas yang tidak akurat, dan ketidakkonsistenan respons relai. Dibandingkan dengan metode iteratif atau spiral, prototyping lebih sesuai karena sistem IoT sangat bergantung pada eksperimen coba-coba (trial-and-error) dengan komponen fisik, yang memerlukan penyesuaian berulang pada tingkat perangkat keras—sesuatu yang tidak sepenuhnya didukung oleh model perangkat lunak iteratif. Prototyping memungkinkan interaksi waktu nyata antara peneliti dan model kerja, sehingga memungkinkan perubahan pada pengabelan, penempatan sensor, logika program, atau konfigurasi Blynk tanpa harus mengulang seluruh siklus pengembangan dari awal. Fleksibilitas ini sangat krusial dalam penelitian IoT, di mana kinerja sistem tidak dapat divalidasi hanya melalui desain teoretis tetapi harus diamati langsung menggunakan prototipe yang berfungsi. Oleh karena itu, pendekatan prototyping menyediakan jalur pengembangan yang paling praktis, efisien, dan realistis untuk proyek Smart Classroom ini.

Gambar 3.1 Flowchart Metode Penelitian Prototipe



3.1 Tahapan Penelitian

Penelitian ini menggunakan metode **Prototyping** yang bersifat iteratif (berulang) untuk memastikan sistem yang dihasilkan sesuai dengan kebutuhan:

3.1.1 Analisis Kebutuhan

Mengidentifikasi ambang batas suhu (>32°C), kebutuhan deteksi gas, dan pemantauan akses pintu. Pada tahap awal, kebutuhan fungsional dan non-fungsional dari prototipe *Smart Classroom* diidentifikasi.

- Kebutuhan fungsional meliputi: 1) Pemantauan suhu, kelembapan, konsentrasi gas, dan aktivitas pintu secara *real-time*. 2) Pengendalian lampu dan kipas secara otomatis melalui modul relai. 3) Pemantauan jarak jauh melalui aplikasi seluler Blynk.
- Kebutuhan non-fungsional berfokus pada**:** Biaya rendah, responsivitas waktu nyata, dan konektivitas stabil melalui NodeMCU ESP8266.

  Tahap ini juga mencakup observasi masalah di kelas selama masa magang (kontrol perangkat manual, tidak adanya sistem pemantauan), yang menjadi landasan pembangunan prototipe terintegrasi. Selain observasi, dilakukan konsultasi langsung dengan 9 calon pengguna (3 guru, 3 teknisi sekolah, dan 3 siswa). Guru menekankan kenyamanan belajar; teknisi memberikan masukan mengenai kendala pengabelan dan pemeliharaan; sementara siswa berbagi pengalaman mengenai ketidaknyamanan akibat suhu tinggi dan kelalaian mematikan perangkat listrik.

3.1.2Desain Sistem

Perancangan skematik rangkaian (Tabel Pin D4, D5, D6, dll) dan alur kerja perangkat lunak (*pseudo-code*). Seluruh arsitektur IoT dirancang, yang terdiri dari:

1. Desain Perangkat Keras: Sensor DHT11, sensor MQ-2, sensor pintu magnetik, modul relai, dan NodeMCU ESP8266.
2. Desain Jaringan: Komunikasi antara NodeMCU dan *cloud* Blynk menggunakan Wi-Fi.
3. Desain Perangkat Lunak**:** Alur program Arduino IDE (pembacaan sensor → logika kontrol → unggah data) dan antarmuka *dashboard* Blynk (widget tampilan data, sakelar, dan notifikasi).

3.1.3Pembuatan Prototipe

Perakitan komponen fisik pada papan prototipe dan penulisan kode program pada Arduino IDE.Semua komponen elektronik dihubungkan sesuai desain pengabelan:

1. Menghubungkan DHT11 ke GPIO digital untuk pemantauan suhu dan kelembapan.
2. Menghubungkan MQ-2 untuk deteksi gas dan pembacaan analog.
3. Memasang sensor pintu magnetik pada pintu prototipe.
4. Mengintegrasikan modul relai 2-saluran untuk mengontrol lampu dan kipas.
5. Memberi daya pada sistem melalui USB/adaptor untuk memastikan operasi yang stabil. Kalibrasi komponen—terutama MQ-2—dilakukan untuk memastikan pembacaan sensor yang andal.

3.1.4Pengembangan Perangkat Lunak

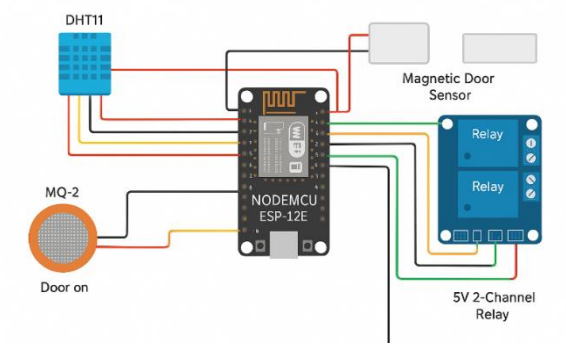
Pengembangan perangkat lunak difokuskan pada dua area utama:

- Pemrograman NodeMCU (Arduino IDE): Mencakup konfigurasi Wi-Fi, logika pembacaan sensor, logika kontrol relai, pengiriman data ke Blynk pada interval yang ditentukan, dan pemicu notifikasi kebocoran gas serta aktivitas pintu.
- Pengembangan Antarmuka Seluler Blynk: Menambahkan widget seperti *Gauge, LED Indicator, Label Value,* dan *Switch* untuk menampilkan data dan mengontrol perangkat secara manual.

3.1.5Evaluasi & Perbaikan

Mengidentifikasi kendala seperti *delay* transmisi data atau stabilitas koneksi Wi-Fi untuk penyempurnaan lebih lanjut. Setelah pengembangan, kedua sistem (perangkat keras dan lunak) diintegrasikan. Tahap ini meliputi pengunggahan program ke NodeMCU, menghubungkannya ke Wi-Fi dan *cloud* Blynk, serta menguji stabilitas koneksi. Penyesuaian nilai ambang batas (*threshold*) dilakukan (misalnya, kipas menyala pada suhu >30°C). Kesalahan selama integrasi diperbaiki melalui penyesuaian pengabelan, pengodean, atau konfigurasi jaringan.

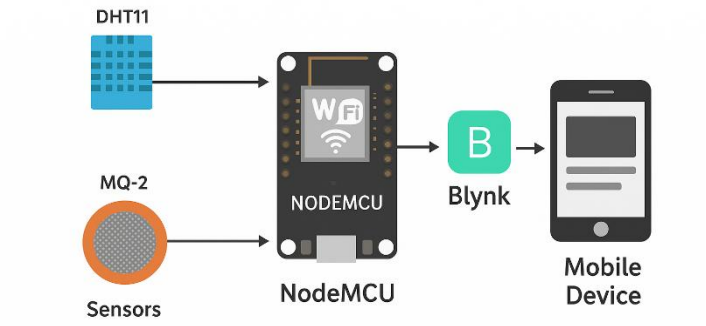
Gambar 3.2 Wiring Diagram



Tabel 1. Pin Konfigurasi

Komponen	**Pin pada Komponen**	**Pin NodeMCU**	**Deskripsi**
Sensor DHT11	VCC	3.3V	Catu daya
	GND	GND	*Ground*
	Data	D4	Data suhu & kelembapan
Sensor Gas MQ-2	VCC	5V	Operasi pemanas & sensor
	GND	GND	*Ground*
	AOUT	A0	Pembacaan konsentrasi gas analog
Sensor Pintu Magnetik	Signal	D5	Deteksi buka/tutup pintu
	VCC	3.3V	Catu daya sensor
	GND	GND	*Ground*
Modul Relai	IN1	D6	Kontrol kipas
	IN2	D7	Kontrol lampu
	VCC	5V	Catu daya relai
	GND	GND	*Ground*

Gambar 3.3 Diagram skematik alur data



Sensor gas MQ-2 memerlukan periode pemanasan dan stabilisasi sebelum dapat menghasilkan pembacaan yang andal. Sensor ini menggunakan kumparan pemanas internal yang harus mencapai suhu operasional tertentu untuk mengaktifkan lapisan semikonduktor oksida logam yang sensitif. Selama proses pemanasan ini, nilai sensor cenderung berfluktuasi dan belum merepresentasikan tingkat konsentrasi gas yang sebenarnya. Dalam aplikasi IoT pada umumnya, MQ-2 memerlukan waktu pemanasan minimum 20–30 detik untuk pengoperasian dasar; namun, untuk pembacaan yang stabil dan akurat, disarankan periode kalibrasi yang lebih lama yaitu 1–5 menit. Dalam penelitian ini, sensor dibiarkan memanas secara memadai sebelum akuisisi data dilakukan untuk mengurangi alarm palsu dan memastikan bahwa ambang batas deteksi gas hanya terpicu ketika terjadi perubahan konsentrasi yang signifikan. Persyaratan pemanasan ini merupakan pertimbangan utama dalam desain sistem, karena memengaruhi responsivitas saat awal dinyalakan dan harus diperhitungkan selama integrasi serta pengujian.

Gambar 3.4 kode program control lampu dan kipas

```
READ temperature from DHT11
READ humidity from DHT11
READ gas_value from MQ-2
READ door_status from magnetic sensor

// Automatic Fan Control
IF temperature > 30°C THEN
    TURN ON fan (relay1 = HIGH)
ELSE IF temperature < 28°C THEN
    TURN OFF fan (relay1 = LOW)
ENDIF

// Automatic Light Based on Door Activity
IF door_status = OPEN THEN
    TURN ON light (relay2 = HIGH)
ELSE
    TURN OFF light (relay2 = LOW)
ENDIF
```

Pseudo-kode dan cuplikan skrip ini mendemonstrasikan logika otomasi inti yang digunakan dalam purwarupa *Smart Classroom*. NodeMCU secara terus-menerus membaca data dari DHT11, sensor gas MQ-2, dan sensor pintu magnetik. Berdasarkan ambang batas (*threshold*) yang telah ditentukan sebelumnya, sistem secara otomatis mengontrol perangkat listrik melalui modul relai. Kipas angin diaktifkan ketika suhu melebihi 30°C dan dimatikan saat suhu turun di bawah 28°C. Lampu dipicu oleh aktivitas pintu untuk mensimulasikan penggunaan ruang kelas secara real-time. Deteksi gas mengaktifkan peringatan Blynk untuk memberi tahu pengguna seketika itu juga. Semua data sensor dikirim ke *dashboard* Blynk untuk pemantauan waktu nyata.
Dalam purwarupa ini, data sensor ditransmisikan dari NodeMCU ke server Blynk dengan laju pengambilan sampel (*sampling rate*) sebesar 1 pembaruan data per detik (1 Hz). Penundaan (*delay*) sebesar 1000 ms diterapkan pada siklus utama (*main loop*) untuk memastikan pewaktuan yang konsisten antar pembacaan. Laju ini dipilih karena menyediakan pemantauan real-time sambil tetap menjaga stabilitas komunikasi Wi-Fi dan mencegah kepadatan data pada mikrokontroler. Pada interval ini, suhu, kelembapan, konsentrasi gas, dan status pintu semuanya dikirim ke Blynk kira-kira sekali setiap detik, memungkinkan pengguna untuk mengamati perubahan lingkungan dengan latensi minimal.

3.1.6Pengujian

Black-box testing dilakukan untuk memverifikasi bahwa setiap fitur berfungsi sesuai dengan spesifikasi. Pengujian tersebut meliputi: akurasi pembacaan suhu & kelembapan, responsivitas deteksi gas (MQ-2), deteksi sensor pintu, kecepatan peralihan relai, transmisi data Blynk secara *real-time*, serta kontrol perangkat otomatis vs manual. Sistem diuji berulang kali untuk memastikan kinerja yang andal di bawah berbagai simulasi kondisi ruang kelas. Batasan yang ditemukan meliputi ketidakstabilan Wi-Fi, penundaan pemanasan MQ-2, dan sesekali terjadi keterlambatan transmisi data. Metode prototipe memastikan pengembangan yang fleksibel dan iteratif, cocok untuk sistem IoT yang memerlukan umpan balik waktu nyata serta penyempurnaan perangkat keras-perangkat lunak. Adaptabilitasnya memberikan cara yang realistis untuk menguji sensor, mengoptimalkan komunikasi jaringan, dan meningkatkan fungsionalitas sistem—keunggulan yang tidak ditawarkan oleh model SDLC kaku seperti *Waterfall*.
Pengujian sistem dilakukan di lingkungan dalam ruangan yang terkendali untuk mensimulasikan kondisi ruang kelas yang sebenarnya. Ruang uji berukuran sekitar 6 m × 7 m (42 m²), yang mencerminkan ukuran rata-rata ruang kelas standar. Tinggi langit-langit sekitar 3 meter, dan ruangan berisi furnitur umum seperti meja, kursi, dan lemari, guna memastikan bahwa aliran udara, dispersi gas, dan distribusi suhu menyerupai kondisi operasional nyata.
Penempatan sensor diatur untuk mengoptimalkan akurasi pengukuran:

1. Sensor Suhu & Kelembapan DHT11: Ditempatkan pada ketinggian 1,5 meter dari lantai, dipasang pada dinding yang jauh dari sinar matahari langsung, jendela, dan kipas angin. Diposisikan di tengah untuk menangkap kondisi rata-

rata ruangan, serta menghindari penempatan di dekat sumber panas untuk mencegah pembacaan yang bias.

2. Sensor Gas MQ-2: Dipasang pada 30–50 cm di atas lantai, mengikuti praktik terbaik untuk mendeteksi gas yang lebih berat seperti LPG. Diposisikan di dekat sumber gas/asap simulasi untuk menguji respons bahaya yang nyata, serta diberikan ruang ventilasi yang cukup untuk memungkinkan kalibrasi yang stabil.
3. Sensor Pintu Magnetik: Dipasang pada rangka pintu pada ketinggian 120 cm, dengan *reed switch* yang sejajar dengan magnet pintu untuk memastikan deteksi buka/tutup yang presisi.
4. Perangkat Terkontrol Relai (Lampu & Kipas): Ditempatkan dalam pengaturan papan prototipe, diposisikan untuk mencerminkan pola penggunaan perangkat kelas pada umumnya.

Mikrokontroler NodeMCU ESP8266 diletakkan di tengah ruang uji untuk memastikan konektivitas Wi-Fi yang stabil dan jarak yang sama ke semua sensor. Semua komponen diberi daya menggunakan adaptor 5V yang stabil untuk meminimalkan gangguan listrik (*electrical noise*).

Kondisi lingkungan selama pengujian meliputi: Suhu ruangan 28–32°C, kelembapan relatif 60–75%, aliran udara ventilasi alami dengan pintu/jendela tertutup selama siklus pengujian untuk menjaga konsistensi, serta kekuatan sinyal Wi-Fi –55 hingga –60 dBm, yang diukur untuk memastikan komunikasi yang andal dengan Blynk. Kondisi terkendali ini memastikan hasil pengujian mencerminkan kinerja sistem yang realistis dan memungkinkan evaluasi konsisten terhadap akurasi sensor, perilaku relai, serta konsistensi transmisi data Blynk.

Sistem menggunakan nilai ambang batas (*threshold*) yang telah ditentukan untuk memicu alarm suhu dan kelembapan. Berdasarkan kalibrasi awal dan standar kenyamanan ruang kelas, alarm suhu diaktifkan ketika suhu ruangan melebihi 32°C, menunjukkan kondisi yang dapat menyebabkan ketidaknyamanan siswa atau memerlukan peningkatan ventilasi. Untuk kelembapan, alarm dipicu ketika kelembapan relatif naik di atas 80%, karena kelembapan tinggi dapat memengaruhi kenyamanan kelas dan meningkatkan risiko pertumbuhan jamur. Ambang batas ini dimasukkan ke dalam logika kontrol otomatis dan sistem notifikasi Blynk, memastikan pengguna menerima peringatan waktu nyata setiap kali kondisi lingkungan menyimpang dari tingkat yang direkomendasikan.

Untuk mengevaluasi kinerja sistem, beberapa metrik digunakan untuk menilai akurasi sensor, responsivitas, stabilitas komunikasi, dan keandalan proses otomasi. Metrik ini memberikan gambaran kuantitatif tentang seberapa baik kinerja purwarupa *Smart Classroom* di bawah kondisi operasi tipikal.

Tabel 2. Metrik Kinerja Sistem

Jenis Metrik	Parameter	Hasil	Keterangan
Akurasi Sensor	Suhu (sensor DHT11)	±2°C	Dibandingkan dengan termometer raksa
Akurasi Sensor	Kelembapan (sensor DHT11)	±5% RH	Dibandingkan dengan higrometer digital
Respons Sensor	Deteksi gas MQ-2	2–5 detik	Setelah proses pemanasan awal
Respons Sistem	Peralihan modul relay	< 1 detik	Stabil
Respons Sistem	Aktivasi kipas	1–2 detik	Setelah nilai ambang batas terlampaui
Respons Sistem	Sensor pintu	< 1 detik	Tingkat keberhasilan 100%
Komunikasi	Waktu tunda pembaruan Blynk	200–500 ms	Bergantung pada kualitas Wi-Fi
Komunikasi	Kehilangan paket data	< 5%	Terjadi saat interferensi
Keandalan	Siklus relay	60/60 berhasil	Tidak terjadi kegagalan
Keandalan	Waktu aktif sistem (uptime)	100%	Pengujian selama 2 jam
Akurasi Otomasi	Kipas ON/OFF	28/30 berhasil	Terdapat keterlambatan minor

Metrik kinerja tersebut menunjukkan bahwa sistem mampu beroperasi secara andal dalam kondisi ruang kelas yang terkontrol serta memenuhi kebutuhan fungsional untuk pemantauan real-time dan pengendalian perangkat secara otomatis.

3.2 Alat dan Bahan

Untuk membangun sistem ini, diperlukan komponen perangkat keras sebagai berikut:

- **Perangkat Keras (Hardware):**
  - Mikrokontroler NodeMCU ESP8266 (1 unit).
  - Sensor Suhu & Kelembapan DHT11 (1 unit).
  - Sensor Gas/Asap MQ-2 (1 unit).
  - Sensor Magnetik Pintu (1 unit).
  - Modul Relai 2-Channel 5V (1 unit).
  - Adaptor 5V/2A sebagai catu daya utama.
- **Perangkat Lunak (Software):**
  - Arduino IDE (untuk pemrograman C++).
  - Aplikasi Blynk (untuk antarmuka monitoring)