

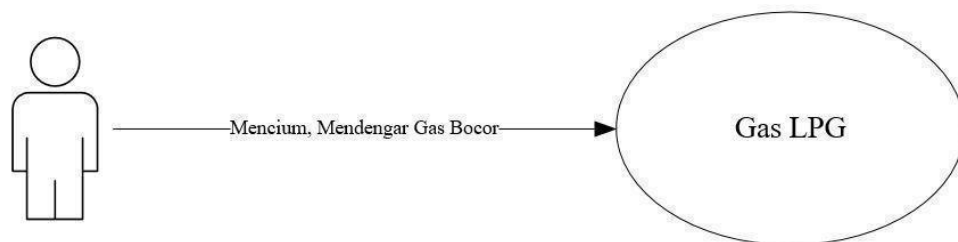
## BAB III PEMBAHASAN

### 1.1. Tahap Perencanaan

Tahap perencanaan merupakan langkah awal dalam proses penelitian yang bertujuan untuk merumuskan tujuan, menentukan kebutuhan alat dan bahan, serta menyusun strategi pelaksanaan proyek. Pada tahap ini, dilakukan identifikasi permasalahan yang akan diselesaikan, yaitu banyaknya kasus kebakaran akibat kebocoran gas LPG yang belum terdeteksi secara dini.

Perencanaan dilakukan dengan membuat rancangan sistem deteksi kebocoran gas berbasis *IoT* menggunakan *Arduino Nano* dan sensor *MQ-2*. Rancangan tersebut mencakup diagram blok sistem, penentuan komponen yang digunakan, serta alur kerja sistem. Selain itu, dilakukan juga perencanaan komunikasi data menggunakan modul *GSM SIM800L* untuk pengiriman notifikasi dan penggunaan *LCD* serta *buzzer* sebagai indikator peringatan lokal.

Tahap ini juga meliputi penyusunan jadwal kegiatan, mulai dari proses perancangan perangkat keras dan lunak, perakitan komponen, pengujian, hingga evaluasi sistem. Perencanaan yang matang sangat penting agar sistem dapat bekerja sesuai harapan dan penelitian berjalan dengan terstruktur serta tepat waktu.



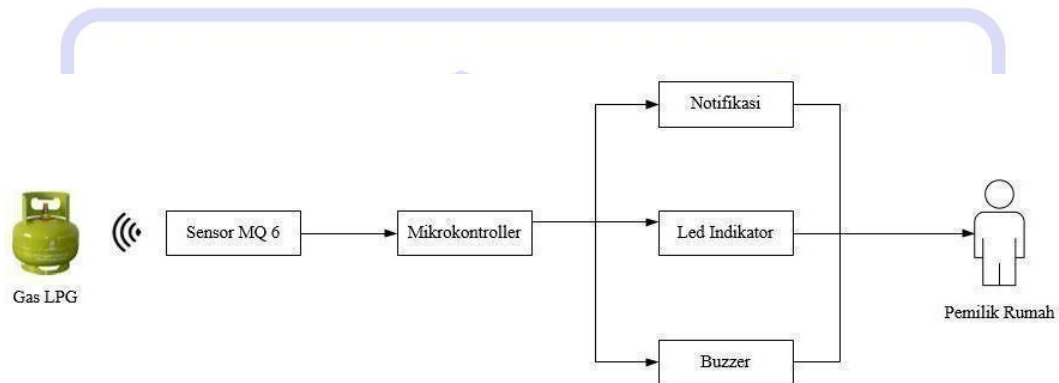
Pemilik rumah

Sumber : Hasil Penelitian 2025

Gambar III. 1 *Use Case Diagram* Mendeteksi Kebocoran Gas secara manual

### 3.2. Tahap Pemodelan Sistem

Tahap pemodelan sistem merupakan proses merancang bentuk dan alur kerja sistem deteksi kebocoran gas sebelum direalisasikan secara fisik. Pada tahap ini, dilakukan pembuatan model konseptual yang menggambarkan hubungan antar komponen, alur data, serta mekanisme kerja secara keseluruhan. Dapat dilihat pada gambar III.2 sebagai berikut :



Sumber : Hasil Penelitian 2025

Gambar III. 2 Perancangan Alat Pendeteksi Kebocoran Gas LPG

Model sistem digambarkan menggunakan diagram blok yang menunjukkan interaksi antara sensor *MQ-2*, mikrokontroler *Arduino Nano*, modul *LCD*, *buzzer*, dan modul *GSM SIM800L*. Sensor berfungsi sebagai input utama yang mendeteksi keberadaan gas, kemudian data dikirim ke mikrokontroler untuk dianalisis. Berdasarkan hasil analisis, mikrokontroler akan mengatur keluaran berupa tampilan informasi di *LCD*, aktivasi *buzzer*, serta pengiriman *SMS* peringatan melalui modul *GSM*.

Selain diagram blok, pada tahap ini juga dilakukan perancangan *flowchart* program untuk menjelaskan alur logika kerja sistem mulai dari pembacaan sensor, pengolahan data, hingga pengiriman peringatan. Pemodelan ini bertujuan untuk

mempermudah proses perakitan dan pemrograman, serta meminimalisir kesalahan dalam tahap implementasi.

Dengan pemodelan sistem yang jelas dan terstruktur, proses pengembangan dapat berjalan lebih efisien, dan setiap perubahan desain dapat dilakukan dengan mudah sebelum sistem dibangun secara nyata.

### **3.2.1. Tahap Analisis**

#### **1. Fungsi Alat**

Fungsi alat pendeteksi gas adalah sebuah sistem yang melakukan deteksi kebocoran gas dengan peringatan melalui SMS (*Short Message Service*) terdiri dari perangkat komponen yang bekerja secara berkesinambungan yaitu dimulai pada Sementara itu, dari sisi perangkat lunak, dilakukan analisis terhadap logika pemrograman yang akan digunakan dalam *Arduino IDE*. Termasuk di dalamnya adalah alur pembacaan data sensor, pemrosesan data, pengiriman *SMS*, serta tampilan informasi di *LCD*.

Hasil dari tahap analisis ini menjadi dasar dalam menyusun rancangan sistem secara keseluruhan agar sesuai dengan kebutuhan serta dapat diimplementasikan secara efektif.

#### **2. Cara Kerja Alat Secara Umum**

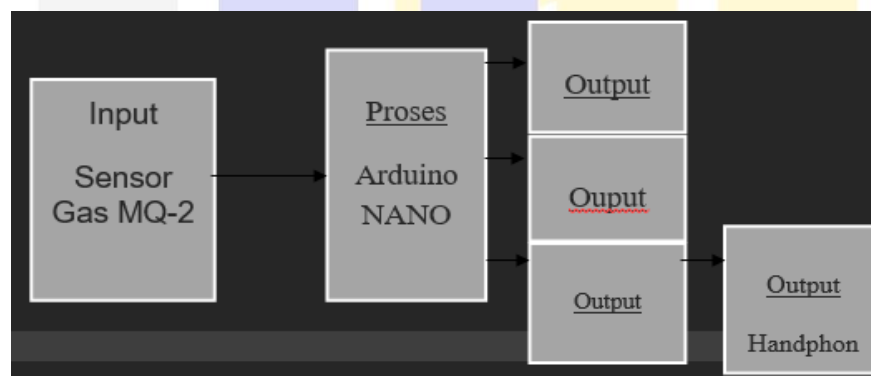
Secara umum, sistem ini bekerja dengan memantau konsentrasi gas di udara secara terus-menerus melalui sensor *MQ-2*. Jika terdeteksi konsentrasi gas di atas ambang batas, sensor mengirimkan sinyal ke *Arduino Nano* untuk diproses. Mikrokontroler kemudian memicu *buzzer* sebagai alarm suara, menampilkan

peringatan pada *LCD*, dan mengirimkan *SMS* ke nomor pengguna yang telah diprogram. Jika tidak ada kebocoran, sistem tetap dalam mode siaga dan terus melakukan pemantauan secara otomatis.

### 3.2.2. Tahap Desain

#### 1. Skema Blok Diagram

Perancangan blok diagram dimaksudkan untuk memberikan gambaran mengenai alat yang dirancang. Secara garis besar prinsip kerja dari sistem yang dibuat ini adalah seperti terlihat pada gambar III.3 di bawah ini :



Sumber : Hasil Penelitian 2025

Gambar III. 3 Blok Diagram

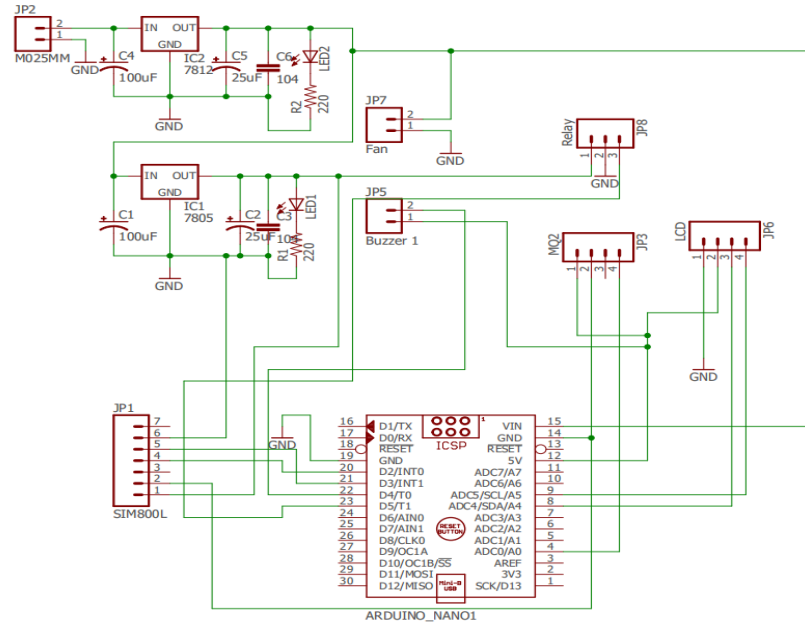
Adapun fungsi dari masing-masing blok diagram tersebut adalah:

1. Sensor MQ-2 – Mendeteksi konsentrasi gas LPG, metana, atau asap di udara dan mengubahnya menjadi sinyal listrik.
2. Arduino Nano – Berfungsi sebagai pengendali utama yang memproses data dari sensor dan mengatur keluaran.
3. LCD 16x2 – Menampilkan informasi kondisi gas secara langsung, seperti status normal atau adanya kebocoran.
4. Buzzer – Mengeluarkan bunyi peringatan jika terdeteksi kebocoran gas, sebagai alarm bagi pengguna di sekitar.



5. Modul GSM SIM800L – Mengirimkan notifikasi berupa SMS ke nomor tujuan yang telah diprogram saat terdeteksi kebocoran.

## 2. Skema Rangkaian



Sumber : Hasil Penelitian 2025

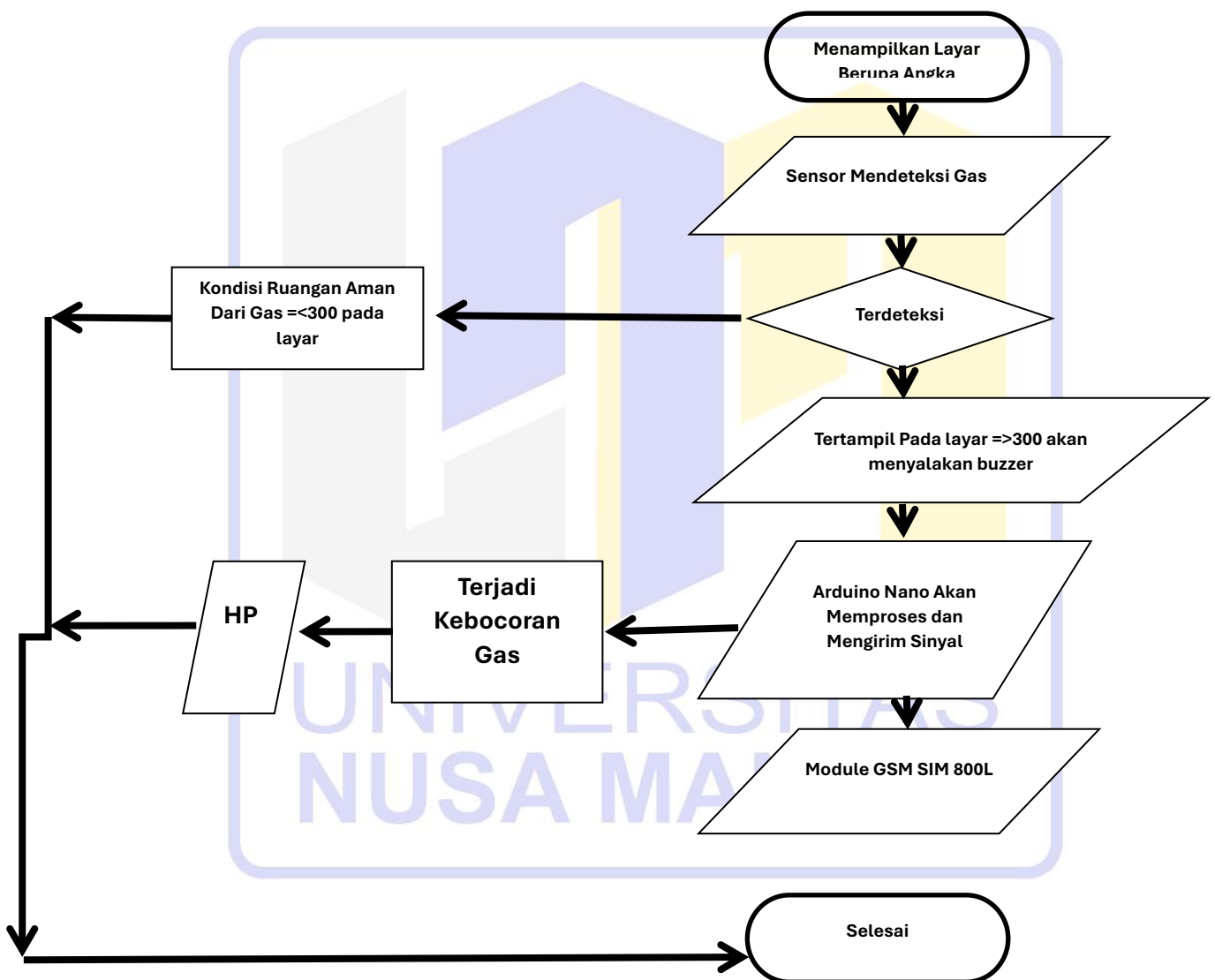
Gambar III. 4 Skema Rangkaian

terlihat bahwa sensor *MQ-2* terhubung ke pin input analog *Arduino Nano*, *LCD 16x2* terhubung melalui modul *I2C* ke pin komunikasi, *buzzer* disambungkan ke pin digital output, dan modul *GSM SIM800L* terkoneksi menggunakan jalur komunikasi serial (*TX* dan *RX*). Catu daya disalurkan melalui adaptor 12V yang kemudian distabilkan oleh regulator tegangan sebelum menuju komponen. Skema ini memudahkan proses perakitan karena menampilkan detail pengkabelan, penempatan resistor dan kapasitor, serta jalur distribusi daya.

### 3.3. Tahap Konstruksi

#### 1. Flowchart Program

Flowchart program menggambarkan alur logika kerja sistem, dimulai dari inisialisasi perangkat keras, pembacaan sensor, pemeriksaan nilai konsentrasi gas, hingga tindakan yang diambil.



Sumber : Hasil Penelitian 2025

Gambar III. 5 Flowchart

Secara umum langkah-langkahnya adalah:

1. **Mulai** – Sistem melakukan inisialisasi komponen seperti sensor *MQ-2*, *LCD*, *buzzer*, dan *modul GSM*.
2. **Baca Sensor** – Mikrokontroler membaca nilai konsentrasi gas dari sensor *MQ-2*.
3. **Periksa Ambang Batas** – Data yang terbaca dibandingkan dengan nilai batas aman yang telah ditentukan.
4. **Kondisi Normal** – Jika nilai gas di bawah batas aman, sistem kembali membaca sensor (looping).
5. **Kondisi Bahaya** – Jika nilai gas melebihi ambang batas, *buzzer* diaktifkan, peringatan ditampilkan di *LCD*, dan SMS dikirimkan melalui *modul GSM*.
6. **Kembali ke Pemantauan** – Sistem kembali memantau kondisi gas secara berulang.

Flowchart ini memastikan sistem berjalan sesuai logika yang dirancang dan membantu proses debugging saat pengujian.

## 7. Konstruksi Sistem Codingan

Konstruksi sistem dimulai dengan penulisan kode program menggunakan *Arduino IDE* dalam bahasa C/C++. Bagian **setup()** berisi inisialisasi semua perangkat keras, termasuk pengaturan pin untuk sensor, *LCD*, *buzzer*, dan modul *GSM SIM800L*. Bagian **loop()** menjalankan proses utama secara berulang: membaca data sensor, membandingkan dengan ambang batas, mengaktifkan alarm, menampilkan status di *LCD*, serta mengirim SMS jika terdeteksi kebocoran *software* arduino IDE 2.0.3 berikut codingan yang kami *input*:

### A. Inisialisasi

```
LiquidCrystal_I2C lcd(0x27, 16, 2);
```

```
SoftwareSerial sim(2, 3);
```

```
int _timeout;
```

```
String _buffer;
```

```
String number = "+6282246800344";
```

```
int buzzer = 4.;
```

```
int fan = 5.;
```

```
int sensor = A0.;
```

```
int threshold = 300;
```

Penjelasan :

Kode juga dilengkapi dengan fungsi-fungsi pendukung seperti format pesan SMS, kontrol bunyi *buzzer*, dan pengaturan tampilan *LCD*. Penulisan kode dilakukan dengan memanfaatkan pustaka (*library*) resmi Arduino dan pustaka tambahan untuk modul GSM dan LCD, sehingga sistem lebih efisien, mudah dibaca, dan dapat dioptimalkan saat pengujian.

## **B. Input**

```
void setup() {
```

```
    delay(7000);
```

```
    digitalWrite(fan, HIGH);
```

```
    pinMode(buzzer, OUTPUT);
```

```
    pinMode(fan, OUTPUT);
```

```
    pinMode(sensor, INPUT);
```

```
    Serial.begin(9600);
```

```
    _buffer.reserve(50);
```

```
Serial.println("Sistem Started...");

sim.begin(9600);

delay(1000);

lcd.begin();

}
```

Keterangan :

Pada alat kita memberikan sistem untuk mengkonfigurasi seperti booting pada komputer yaitu dengan menggunakan *delay(7000)*; agar sistem dapat menunggu sim gsm mendapatkan sinyal

Pada *digital write(fan, HIGH)*; kita posisikan kipas itu *high* karena jika ambang batas sensor terdeteksi maka kipas akan menyala, dan kita setting pada pinmode bahwa *buzzer* itu *output* keluaran akan mengeluarkan suara ketika sensor mencium adanya bau gas dan kipas juga keluaran sebagai penyedot gas, dikala ruangan itu dipenuhi oleh gas, pada sensor kita setting *input* dikarenakan dia memberikan informasi ke arduino sampai ke pengguna.

pada program *sim.begin(9600)*;

*delay(1000)*;

*lcd.begin()*;

kita mensetting bahwa ketika sistem dinyalakan maka kita memberikan *delay* 1 detik untuk sim mendapatkan sinyal dan juga pada lcd agar dapat menampilkan “sistem started bahwa alat elektronika kita menyala dengan baik

### C. Main Program

```
#include <SoftwareSerial.h>
```

```
#include <LiquidCrystal_I2C.h>
```

```
LiquidCrystal_I2C lcd(0x27, 16, 2);
```

```
SoftwareSerial sim(2, 3);
```

```
int _timeout;
```

```
String _buffer;
```

```
String number = "+62822246800344";
```

```
int buzzer = 4.;
```

```
int fan = 5.;
```

```
int sensor = A0.;
```

```
int threshold = 300;
```

```
void setup() {
```

```
    delay(7000);
```

```
    digitalWrite(fan, HIGH);
```

```
    pinMode(buzzer, OUTPUT);
```

```
    pinMode(fan, OUTPUT);
```

```
    pinMode(sensor, INPUT);
```

```
    Serial.begin(9600);
```

```
_buffer.reserve(50);

Serial.println("Sistem Started...");

sim.begin(9600);

delay(1000);

lcd.begin();

}

void loop() {

float MQ2SensorValues = analogRead(sensor);

Serial.println(MQ2SensorValues);

lcd.setCursor(0, 0);

lcd.print("Ur Home is Safe!");

lcd.setCursor(0, 1);

lcd.print("Sensor Value:");

lcd.setCursor(13, 1);

lcd.print(MQ2SensorValues);

Serial.println(MQ2SensorValues);

delay(500);

if (MQ2SensorValues > threshold) // Checks if MQ2 Sensor Value greater than
threshold value
```

```
{  
  
  lcd.setCursor(0, 0);  
  
  lcd.print("Gas Leakage...!!");  
  
  digitalWrite(buzzer, LOW);  
  
  digitalWrite(fan, LOW);  
  
  SendMessage();  
}  
  
if (MQ2SensorValues < threshold)  
{  
  digitalWrite(buzzer, HIGH);  
  digitalWrite(fan, HIGH);  
}  
  
  delay(500);  
  lcd.clear();  
}  
  
void SendMessage()  
{  
  
  //Serial.println ("Sending Message");  
  
  sim.println("AT+CMGF=1"); //Instructions the GSM Module in Text Mode
```



```
delay(200);

//Serial.println ("Instruction SMS Number");

sim.println("AT+CMGS=\"" + number + "\"\r"); //Mobile phone number to send
message

delay(200);

String SMS = "Terjadi Kebocoran GAS!!!";

sim.println(SMS);

delay(100);

sim.println((char)26); // ASCII code of CTRL+Z

delay(200);

_buffer = _readSerial();

delay(30000);

};

String _readSerial() {

_timeout = 0;

while (!sim.available() && _timeout < 12000 )

{

delay(13);

_timeout++;

}
```

```

}

if (sim.available()) {

    return sim.readString();

}

};

```

Keterangan :

Berikut adalah seluruh pemrograman yang kami buat pada alat mendeteksi kebocoran gas dari instalasi sampai *output* untuk memberikan perintah ke arduino agar alat yang kita buat bisa memberikan informasi kepada pengguna untuk mencegah terjadinya kebocoran gas

#### **D. Output**

```

void SendMessage()

{

    //Serial.println ("Sending Message");

    sim.println("AT+CMGF=1"); //Instructions the GSM Module in Text Mode

    delay(200);

    //Serial.println ("Instruction SMS Number");

    sim.println("AT+CMGS=\"" + number + "\"\r"); //Mobile phone number to send
message

    delay(200);

```

```
String SMS = "Terjadi Kebocoran GAS!!!";

sim.println(SMS);

delay(100);

sim.println((char)26); // ASCII code of CTRL+Z

delay(200);

_buffer = _readSerial();

delay(30000);

};

String _readSerial() {

    _timeout = 0;

    while (!sim.available() && _timeout < 12000 )

    {

        delay(13);

        _timeout++;

    }

    if (sim.available()) {

        return sim.readString();

    }

};
```

Keterangan :

Pada **loop()**, sistem membaca sensor, membandingkan nilai dengan ambang batas, dan jika terdeteksi kebocoran, *buzzer* serta kipas diaktifkan, status ditampilkan di *LCD*, dan SMS dikirim. Pesan dibuat dengan String SMS = "Terjadi Kebocoran GAS!!!"; lalu dikirim melalui `sim.println("AT+CMGS=\\\" + number + "\\\"\\r");`, di mana *number* adalah nomor pengguna yang telah didaftarkan. Pengiriman SMS dilakukan setelah sensor memicu alarm dan data diproses oleh *Arduino* untuk diteruskan ke modul GSM.

### 3.4. Tahap Implementasi

#### 3.4.1. Cara Kerja Alat

Alat pendeteksi kebocoran gas ini beroperasi dengan memantau konsentrasi gas di udara secara terus-menerus melalui sensor *MQ-2*. Ketika sensor mendeteksi kadar gas yang melebihi ambang batas yang telah ditentukan, sinyal dikirimkan ke mikrokontroler *Arduino Nano* untuk diproses. Berdasarkan hasil pemrosesan, sistem akan:

1. Mengaktifkan *buzzer* sebagai alarm suara untuk memberikan peringatan langsung.
2. Menyalakan kipas (*exhaust fan*) untuk membantu mengeluarkan gas dari ruangan.
3. Menampilkan pesan peringatan pada *LCD 16x2* agar pengguna mengetahui kondisi bahaya.
4. Mengirimkan pesan singkat (*SMS*) ke nomor telepon pengguna yang telah terdaftar menggunakan modul *GSM SIM800L*.

Jika tidak ada kebocoran terdeteksi, sistem tetap berada dalam kondisi siaga dan melanjutkan pemantauan secara berkala. Mekanisme ini memastikan bahwa

peringatan diberikan secara cepat baik secara lokal (melalui *buzzer* dan *LCD*) maupun jarak jauh (melalui SMS), sehingga risiko bahaya dapat diminimalisir.

### 3.4.2. Error Handling

Penambahan error handling pada alat pendeteksi kebocoran gas bukan hanya soal kode tambahan, tetapi merupakan penerapan prinsip keamanan, keandalan, dan fault tolerance dalam sistem tertanam (*embedded system*). Dalam konteks alat pendeteksi kebocoran gas berbasis IoT, error handling diperlukan agar system tetap stabil meskipun terjadi gangguan, memberikan informasi atau peringatan saat ada kesalahan, dan mengurangi risiko kegagalan fungsi, terutama untuk sistem keselamatan. Jenis-Jenis Error pada Sistem IoT dan Cara Menanganinya:

#### 1. Error Sensor

##### Penyebab:

- Sensor rusak, longgar, atau nilai ADC tidak terbaca.
- Pembacaan analog menghasilkan data di luar rentang (0–1023).

##### Solusi / Teori Penanganan:

- Gunakan **validasi data sensor** sebelum diproses.
- Jika nilai  $< 0$  atau  $> 1023$ , tandai sebagai error.
- Aktifkan buzzer atau tampilkan pesan di serial monitor.



Teori

yang

digunakan:

**Input Validation** — memastikan data masukan valid sebelum digunakan untuk keputusan sistem.

#### 2. Error Komunikasi Serial (ESP8266 ↔ Arduino)

##### Penyebab:

- Baud rate tidak sesuai, noise pada jalur RX/TX.

**Solusi / Teori Penanganan:**

- Gunakan fungsi *timeout* saat menunggu respons dari ESP8266.
- Jika tidak ada jawaban dalam jangka waktu tertentu, hentikan proses dan coba ulang.

■ *Teori yang digunakan:*  
*Timeout & Retry Mechanism* — sistem memberi waktu tunggu dan mencoba ulang bila komunikasi gagal.

### 3. **Error Handling** pada Notifikasi Telegram

Ketika alat mengirim data ke Bot Telegram, error bisa terjadi jika:

- Token atau Chat ID salah,
- Koneksi internet terputus,
- Server Telegram lambat merespons.

**Solusi / Teori Penanganan:**

- Pastikan setiap pesan Telegram dikirim dengan **status pengecekan respon** “OK”.
- Jika gagal, lakukan *retry* setelah beberapa detik.
- Jangan kirim pesan berulang terlalu cepat (hindari spam).

■ *Teori yang digunakan:*  
*Rate Limiting & Retry Logic* — mengontrol frekuensi pengiriman agar tidak overload.

### 4. **Error Logika Program atau Hang**

**Penyebab:**

- Program masuk loop tak berujung, variabel tidak terdefinisi.

**Solusi / Teori Penanganan:**

- Gunakan Watchdog Timer (WDT) agar Arduino restart otomatis jika hang.
- Batasi delay terlalu lama di dalam loop().



*Teori*

*yang*

*digunakan:*

Fault Recovery System — sistem bisa pulih sendiri setelah gangguan.

### 3.4.3. Catu daya

Catu daya merupakan sumber energi utama yang digunakan untuk mengoperasikan seluruh komponen dalam sistem. Dalam penelitian ini, sistem menggunakan adaptor bertegangan 12 volt sebagai sumber catu daya eksternal. Tegangan ini kemudian diatur menggunakan *voltage regulator* seperti *LM7812* agar sesuai dengan kebutuhan komponen yang digunakan.

*Arduino Nano* dan sensor-sensor yang terhubung membutuhkan tegangan kerja 5V, sehingga diperlukan *regulator* untuk menurunkan dan menstabilkan tegangan dari sumber utama. Dengan adanya catu daya yang stabil, sistem dapat beroperasi secara optimal tanpa gangguan akibat fluktuasi listrik.

Selain itu, penggunaan catu daya eksternal juga memungkinkan sistem untuk bekerja secara terus-menerus dalam jangka waktu panjang. Komponen-komponen seperti *LCD*, *buzzer*, dan *modul SIM800L* juga bergantung pada kestabilan pasokan daya agar dapat menjalankan fungsinya dengan baik.

Oleh karena itu, pemilihan dan pengaturan catu daya menjadi aspek penting yang harus diperhatikan dalam merancang sistem berbasis *IoT* seperti alat pendeteksi kebocoran gas ini.

### **3.5. Uraian Tugas**

#### **3.5.1. Sistem Analis**

Sistem analis dalam penelitian ini bertugas untuk merancang dan menentukan spesifikasi sistem yang akan dibangun. Peran utama dari sistem analis adalah menjembatani kebutuhan pengguna dengan perancangan teknis alat, sehingga sistem yang dikembangkan sesuai dengan tujuan dan dapat digunakan secara efektif.

Tugas dari sistem analis meliputi identifikasi kebutuhan sistem, pemilihan komponen, serta penyusunan alur kerja dan skenario penggunaan. Dalam proyek ini, sistem analis juga bertanggung jawab atas penentuan jenis sensor yang digunakan (*MQ-2*), jenis mikrokontroler (*Arduino Nano*), serta modul komunikasi (*SIM800L*) yang dapat mendukung fungsi pengiriman pesan.

Selain itu, sistem analis juga melakukan pengujian awal untuk memastikan bahwa desain sistem sudah berjalan sesuai harapan. Analisis kebutuhan pengguna, baik dari sisi teknis maupun operasional, menjadi dasar penting agar alat yang dirancang mampu memberikan solusi terhadap masalah kebocoran gas yang sering terjadi di lingkungan pengguna.

#### **3.5.2. Programmer**

Programmer dalam penelitian ini berperan untuk merancang, menulis, dan menguji kode program yang mengendalikan seluruh fungsi sistem. Tugas utamanya



mencakup pembuatan logika deteksi kebocoran gas, pengaturan komunikasi antara *Arduino Nano* dan modul-modul pendukung seperti *LCD*, *buzzer*, kipas, serta *GSM SIM800L*.

Selain itu, programmer bertanggung jawab untuk memastikan bahwa alur pemrograman berjalan sesuai rancangan *flowchart* dan dapat merespon input sensor secara cepat. Proses ini meliputi penulisan kode menggunakan bahasa C/C++ pada *Arduino IDE*, pengaturan parameter ambang batas sensor, serta pembuatan format pesan SMS yang akan dikirimkan.

Programmer juga melakukan debugging untuk mengidentifikasi dan memperbaiki kesalahan dalam program, serta melakukan optimasi agar sistem dapat berjalan stabil dan efisien. Dengan peran ini, programmer memastikan bahwa perangkat keras dan perangkat lunak dapat bekerja secara terpadu untuk memberikan fungsi deteksi kebocoran gas yang andal.

### **3.5.3. Pengujian Sistem**

Proses hasil percobaan pada alat untuk mendeteksi kebocoran gas sebagai berikut :

#### **Hasil Percobaan *Input***

Pengujian sistem dilakukan untuk memastikan bahwa seluruh komponen dan fungsi bekerja sesuai dengan rancangan. Proses pengujian dimulai dengan memeriksa koneksi fisik antar komponen seperti sensor, *LCD*, *buzzer*, kipas, dan modul *GSM SIM800L*. Selanjutnya dilakukan pengujian perangkat lunak untuk memverifikasi bahwa logika program merespon data sensor dengan tepat.

Tabel III. 1 Hasil Percobaan *Input*

No	<i>Input</i>	Test	Hasil
1	Sensor MQ-2	<i>Instruction</i> telah diberi daya 12 volt	Nyala
2	Tombol Reset Arduino	<i>Instruction</i> telah diberi daya 12 volt	Nyala
3	Sinyal GSM	<i>Instruction</i> telah diberi daya 12 volt	Nyala
4	Adaptor	<i>Instruction</i> telah diberi daya 12 volt	Nyala

Pengujian mencakup simulasi kebocoran gas untuk melihat apakah sistem dapat mendeteksi secara akurat dan memberikan respons sesuai prosedur, yaitu menyalakan *buzzer*, mengaktifkan kipas, menampilkan pesan di *LCD*, serta mengirimkan SMS ke pengguna. Pengujian juga dilakukan dalam kondisi normal untuk memastikan sistem tetap stabil dan tidak memberikan alarm palsu.

Hasil pengujian digunakan untuk melakukan penyesuaian atau perbaikan baik pada perangkat keras maupun perangkat lunak, sehingga alat dapat berfungsi optimal dalam mendeteksi dan memberikan peringatan kebocoran gas.

#### 1. Hasil Percobaan *Output*

Hasil percobaan menunjukkan bahwa ketika sensor *MQ-2* mendeteksi konsentrasi gas melebihi ambang batas, sistem memberikan respon sesuai rancangan. *Buzzer* berbunyi sebagai peringatan suara, kipas menyala untuk mengurangi konsentrasi gas di udara, dan *LCD* menampilkan pesan bahaya. Selain itu, modul *GSM*

*SIM800L* berhasil mengirimkan *SMS* dengan isi “Terjadi Kebocoran GAS!!!” ke nomor pengguna yang telah terdaftar.

Dalam kondisi normal tanpa kebocoran, sistem tetap siaga dan hanya menampilkan status aman pada *LCD* tanpa mengaktifkan alarm maupun mengirim pesan. Hal ini membuktikan bahwa sistem dapat beroperasi sesuai fungsinya dalam mendeteksi dan memberikan peringatan kebocoran gas.

Tabel III. 2 Hasil Percobaan *Output*

No	<i>Output</i>	Test	Hasil
1	LCD	<i>Instruction</i> telah diberi daya 12 volt	Nyala
2	Buzzer	<i>Instruction</i> telah diberi daya 12 volt	Nyala
3	Kipas	<i>Instruction</i> telah diberi daya 12 volt	Nyala
4	Led	<i>Instruction</i> telah diberi daya 12 volt	Nyala
5	SIM800L	<i>Instruction</i> telah diberi daya 12 volt	Nyala