

BAB III

METODOLOGI PENELITIAN

3.1 Alat dan Bahan

3.1.1 Alat

Pada penelitian ini, peneliti telah menyiapkan berbagai peralatan pendukung yang digunakan dalam proses perakitan dan pengujian Sistem Deteksi Gempa Skala Ringan Berbasis IoT dan Android. Peralatan yang dimaksud seperti seperti laptop untuk pemrograman, solder untuk penyambungan komponen, serta alat bantu lainnya yang digunakan selama proses perakitan dan pengujian. Daftar lengkap peralatan yang digunakan ditunjukkan pada Tabel 3.1

Table 3. 1 Alat

NO	Nama Alat	Spesifikasi	Fungsi	Jumlah
1	Laptop	Ram 8GB/SSD 256GB	Sebagai perancangan dan pemrograman sistem	1 Buah
2	Obeng Mini	Obeng Set	Membuka atau mengencangkan baut	1 Buah
3	Solder	30 watt	Sebagai penyambung antara timah dan komponen	1 Buah
4	Arduino IDE	Arduino 2.0.3	Pembuatan serta penguploadan code alat yang dibuat	1 Buah
5	Motor Getar / Dinamo DC	Motor DC 12V	Untuk simulasi gempa pada miniatur	1 Buah
6	Multimeter	Digital	Mengukur tegangan, arus, dan koneksi rangkaian	1 Buah

3.1.2 Bahan

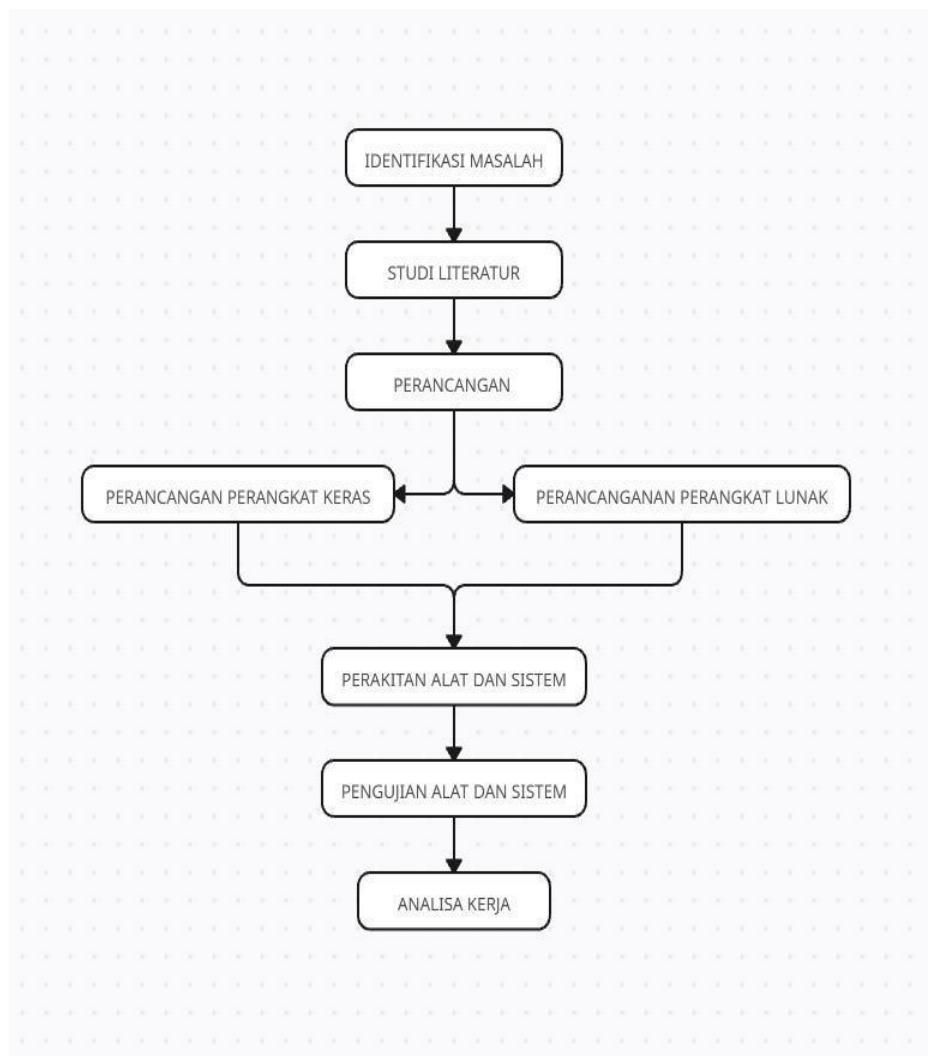
Selain peralatan, peneliti juga telah menyiapkan seluruh bahan utama yang digunakan dalam pembuatan sistem, meliputi komponen elektronik seperti mikrokontroler, sensor, modul komunikasi, dan perangkat keluaran. Bahan-bahan ini dipilih berdasarkan spesifikasi yang sesuai dengan kebutuhan sistem. Daftar lengkap bahan yang digunakan disajikan pada Tabel 3.2

Table 3. 2 Bahan

NO	Nama Alat	Spesifikasi	Fungsi	Jumlah
1	ESP 32	Dual Core, WiFi + Bluetooth, 30 GPIO	Mikrokontroler utama untuk membaca sensor dan kirim data	1 Buah
2	Sensor SW-420	6-axis accelerometer + gyroscope, I2C	Mendeteksi getaran sebagai indikator gempa	1 Buah
3	MPU6050	30 watt	Mengonfirmasi besaran dan arah getaran	1 Buah
4	LCD I2C 16x2	5V, I2C interface	Menampilkan status gempa dan nilai sensor	1 Buah
5	Buzzer Aktif	5V DC, 2kHz	Memberikan peringatan bunyi jika terdeteksi gempa	1 Buah
6	LED (Merah, Kuning, Hijau)	2V–3V, arus 20 mA	Indikator visual status sistem	3 buah
7	Miniatur Bangunan	Akrilik/triplex (ukuran 20x20x30 cm)	Media uji respon sistem terhadap simulasi gempa ringan	1 unit
8	Kabel Jumper	Male-female, panjang ±20 cm	Menghubungkan komponen ke ESP32 atau breadboard	tentative

3.2 Tahapan Penelitian

Tahapan penelitian pada Sistem Deteksi Gempa Skala Ringan Berbasis IoT dan Android disusun untuk memastikan setiap proses berjalan sesuai dengan tujuan yang telah ditetapkan. Tahapan tersebut meliputi studi literatur, perancangan sistem, implementasi perangkat keras, implementasi perangkat lunak, dan pengujian serta simulasi. Pelaksanaan tahapan secara berurutan ini bertujuan untuk memastikan desain, dan validasi sistem berjalan sistematis sehingga memungkinkan mengoreksi data sensor, akurasi pendekalian, dan efektivitas notifikasi secara menyeluruh. Blok diagram tahapan penelitian ditampilkan pada Gambar 3.1.



Gambar 3. 1 Tahapan Penelitian

1. Identifikasi Masalah

Tahap awal dalam penelitian ini adalah melakukan identifikasi masalah.

Pada tahap ini, permasalahan yang akan diuraikan secara jelas dan spesifik agar fokus penelitian dapat ditentukan dengan tepat.

2. Studi Literatur

Setelah masalah diidentifikasi, langkah berikutnya adalah melakukan studi literatur. Studi ini bertujuan untuk mengumpulkan dan menelaah berbagai referensi, teori, dan penelitian terdahulu yang relevan dengan topik penelitian guna memperkuat dasar teori dan kerangka pemikiran.

3. Perancangan

Pada tahap perancangan, dilakukan perencanaan sistem secara menyeluruh yang meliputi perangkat keras dan perangkat lunak. Perancangan ini menjadi acuan dalam pembangunan alat dan sistem yang akan dikembangkan.

4. Perancangan Perangkat Keras

Bagian dari perancangan yang khusus berfokus pada desain dan pemilihan komponen perangkat keras yang akan digunakan dalam sistem, seperti sensor, mikrokontroler, dan perangkat pendukung lainnya.

5. Perancangan Perangkat Lunak

Sama dengan perancangan perangkat keras, dilakukan juga perancangan perangkat lunak yang meliputi pemrograman, algoritma, dan pengembangan aplikasi yang akan mengontrol perangkat keras serta mengolah data.

6. Perakitan Alat dan Sistem

Setelah perancangan perangkat keras dan perangkat lunak selesai, tahap berikutnya adalah merakit seluruh komponen tersebut menjadi satu kesatuan sistem yang berfungsi secara utuh.

7. Pengujian Alat dan Sistem

Tahap ini dilakukan untuk menguji kinerja alat dan sistem yang telah dirakit. Pengujian bertujuan untuk memastikan bahwa alat dan sistem

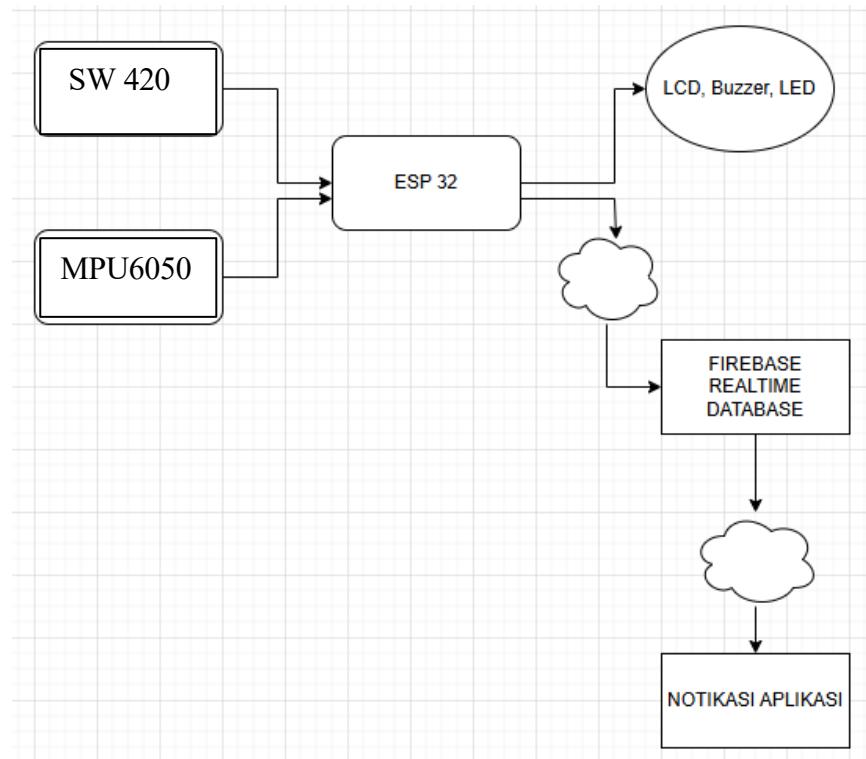
berjalan sesuai dengan rancangan dan mampu menyelesaikan permasalahan yang diidentifikasi

8. Analisis Kerja

Setelah pengujian, dilakukan analisa terhadap hasil kerja alat dan sistem. Analisa ini digunakan untuk mengevaluasi performa, mengidentifikasi kelebihan dan kekurangan, serta memberikan rekomendasi untuk perbaikan di masa mendatang.

3.3 Perancangan Sistem

Perancangan sistem merupakan tahap yang bertujuan untuk mendefinisikan struktur dan hubungan antar komponen, sehingga setiap fungsi dalam Sistem Deteksi Gempa Skala Ringan Berbasis IoT dan Android dapat diimplementasikan secara tepat. Konsep sistem digambarkan pada diagram blok yang dapat dilihat pada gambar 3.2



Gambar 3. 2 Blok Diagram Sistem

Dengan mengacu pada gambar diagram blok sistem, langkah awalnya adalah sebagai berikut;

1. Sensor SW-420

Bertugas mendeteksi getaran fisik kasar pada permukaan. Sensor ini bekerja sebagai pendekripsi awal, memberikan sinyal digital kepada mikrokontroler saat ada getaran.

2. Sensor Accelerometer MPU6050

Digunakan untuk mengukur percepatan getaran pada sumbu X, Y, dan Z secara presisi. Data dari sensor ini memungkinkan pengukuran tingkat intensitas getaran.

3. Mikrokontroler ESP32

Menerima input dari kedua sensor, memproses data, dan menentukan apakah getaran termasuk kategori aman, siaga, atau bahaya. Juga berfungsi sebagai pengendali semua output dan penghubung ke jaringan WiFi.

4. Tampilan Output

Meliputi;

- LCD I2C: menampilkan status gempa
- Buzzer: membunyikan alarm jika terdeteksi gempa
- LED: memberikan indikasi visual status (Hijau: Aman, Kuning: Siaga, Merah: Bahaya).

5. Firebase

Berfungsi menyimpan dan mengelola data getaran. Firebase juga digunakan untuk mengirimkan pop up notification ke aplikasi Android.

6. Aplikasi Android

Menerima data dari Firebase dan menampilkan status gempa. Aplikasi ini juga bisa memberikan notifikasi peringatan dini secara langsung kepada pengguna.

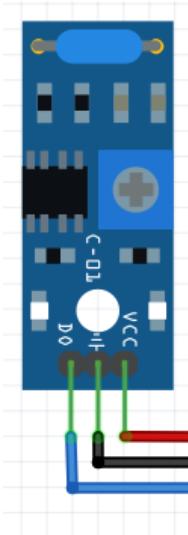
3.3.1 Perancangan Perangkat Keras

Perancangan menjadi salah satu bagian yang sangat penting dilakukan dalam pembuatan alat, karena dengan melakukan perancangan dahulu dengan komponen yang tepat maka rancangan akan berjalan sesuai dengan

konsep yang diinginkan. Dan juga untuk menghindarinya kerusakan komponen perlu dipahami juga akan karakteristik dari komponen komponen tersebut.

3.3.1.1 Rangkaian Sensor SW-420

Rangkaian sensor sw-420 digunakan untuk mendeteksi adanya getaran pada miniatur bangunan. Sensor ini berguna untuk mengetahui adanya aktivitas getaran yang berpotensi berasal dari simulasi gempa ringan. Data deteksi awal dari sensor akan dikirimkan ke ESP32 untuk dilakukan proses validasi lebih lanjut menggunakan sensor MPU6050, sehingga sistem dapat membedakan antara getaran biasa atau ada indikasi untuk gempa yang perlu ditindaklanjuti. Gambar rangkaian Sensor SW-420 dapat diliat pada gambar 3.3



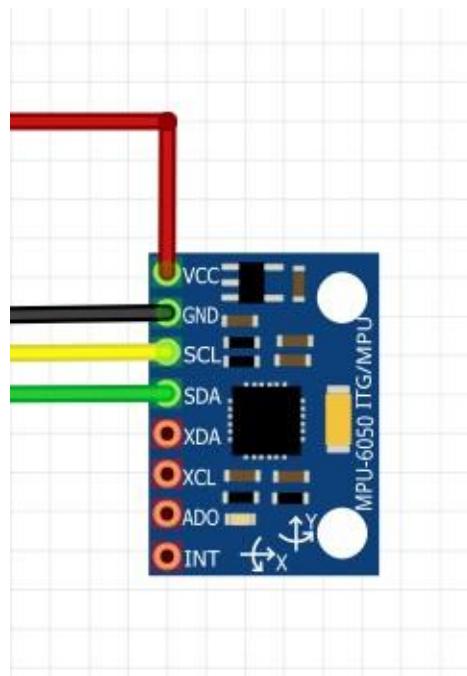
Gambar 3. 3 Rangkaian Sensor SW-420

Pada perancangan perangkat keras ini, sensor SW-420 dihubungkan ke mikrokontroler ESP32 melalui tiga jalur utama, yaitu VCC, GND, dan DO (Digital Output). Jalur VCC sensor dihubungkan ke pin 3.3V pada ESP32 sebagai sumber daya. Jalur GND dihubungkan ke pin ground (GND) ESP32. Jalur DO dihubungkan ke pin GPIO34 pada ESP32 yang telah ditentukan dalam program sebagai pin pembaca sinyal digital dari sensor. Pengkabelan dilakukan menggunakan kabel jumper yang sesuai, serta

sensor ditempatkan pada posisi strategis pada miniatur bangunan untuk memastikan sensitivitas deteksi getaran optimal.

3.3.1.2 Rangkaian Sensor MPU6050

Sensor MPU6050 digunakan untuk mengukur percepatan getara pada miniatur bangunan yang digunakan dalam simulasi gempa ringan. Data percepatan yang sudah diperoleh digunakan untuk memvalidasi hasil deteksi awal sensor SW-420, sehingga sistem dapat menentukan tingkat getaran yang terjadi. Gambar rangkaian sensor MPU6050 dapat dilihat pada gambar 3.4



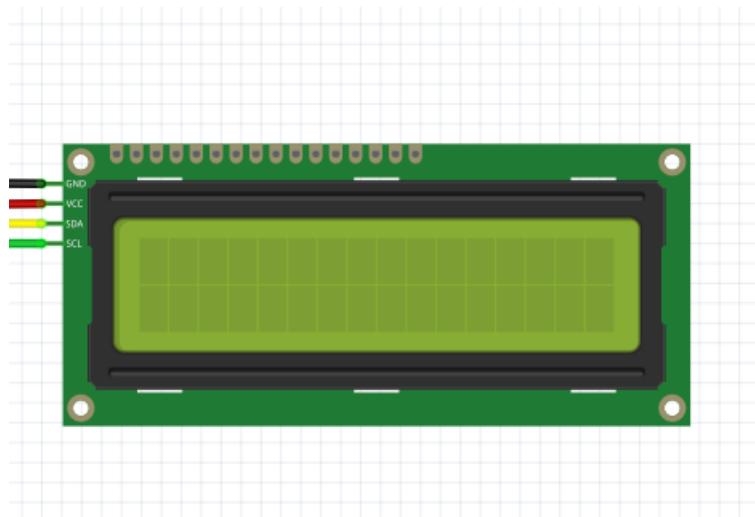
Gambar 3. 4 Rangkaian MPU6050

Pada perancangan perangkat keras, sensor MPU6050 dihubungkan ke mikrokontroler ESP32 menggunakan komunikasi I2C yang memanfaatkan empat jalur utama, yaitu VCC, GND, SDA, dan SCL. Jalur VCC dihubungkan ke pin 3.3V pada ESP32 sebagai sumber daya, sedangkan GND dihubungkan ke ground (GND) ESP32. Jalur SDA (Serial Data) dihubungkan ke pin GPIO21 ESP32, dan jalur SCL (Serial Clock) dihubungkan ke pin GPIO22 ESP32. Pengkabelan dilakukan menggunakan kabel jumper dengan isolasi yang baik untuk menghindari gangguan sinyal.

Penempatan sensor dilakukan pada bagian struktur miniatur bangunan yang paling sensitif terhadap getaran, sehingga data percepatan yang dihasilkan akurat dan representatif terhadap kondisi sebenarnya.

3.3.1.3 Rangkaian LCD I2C

Rangkaian LCD I2C pada alat ini berguna untuk menampilkan infomasi status sistem secara real time, seperti kondisi getaran yang terdeteksi oleh sensor yang akan menampilkan status (Aman, Siaga, atau Bahaya) Serta nilai percepatan dari sensor MPU6050. Gambar rangkaian LCD I2C dapat dilihat pada gambar 3.5



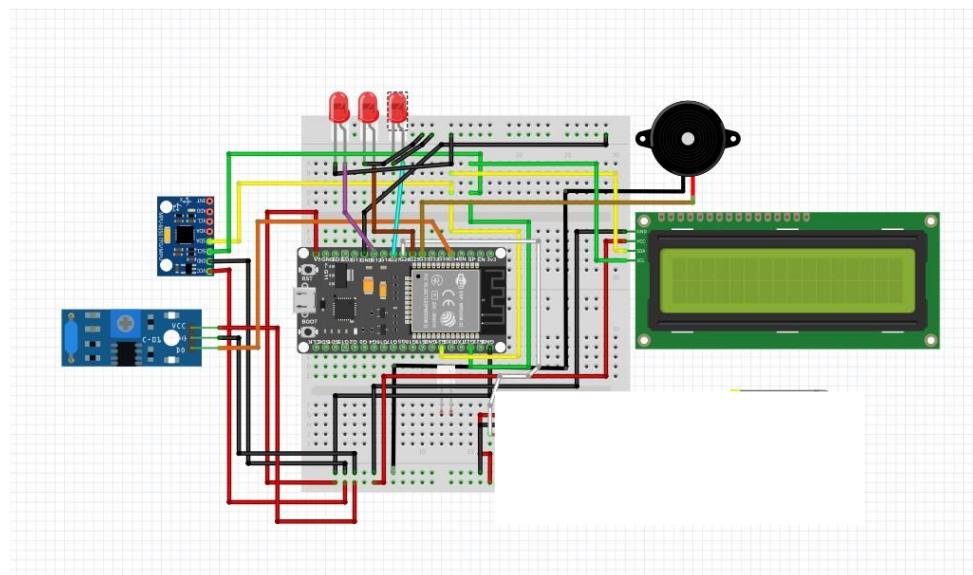
Gambar 3. 5 Rangkaian LCD I2C

Pada perancangan perangkat keras, LCD I2C dihubungkan ke mikrokontroler ESP32 menggunakan komunikasi I2C yang memanfaatkan empat jalur utama, yaitu VCC, GND, SDA, dan SCL. Jalur VCC dihubungkan ke pin 3.3V pada ESP32 sebagai sumber daya, sedangkan GND dihubungkan ke ground (GND) ESP32. Jalur SDA (Serial Data) dihubungkan ke pin GPIO21 ESP32, dan jalur SCL (Serial Clock) dihubungkan ke pin GPIO22 ESP32, yang sama dengan koneksi sensor MPU6050. Penggunaan alamat I2C yang berbeda memungkinkan kedua perangkat berjalan secara bersamaan pada satu jalur komunikasi. Pengkabelan dilakukan dengan kabel jumper yang terisolasi, dan LCD

ditempatkan pada posisi yang mudah terlihat untuk memudahkan pengguna membaca informasi yang ditampilkan.

3.3.1.4 Rangkaian Keseluruhan

Rangkaian keseluruhan merupakan tahap terakhir dari perancangan yang telah dilakukan. Dalam tahap ini seluruh komponen dipasang sesuai dengan sistem yang telah dibuat, adapun rangkaian keseluruhan dapat dilihat pada gambar 3.7

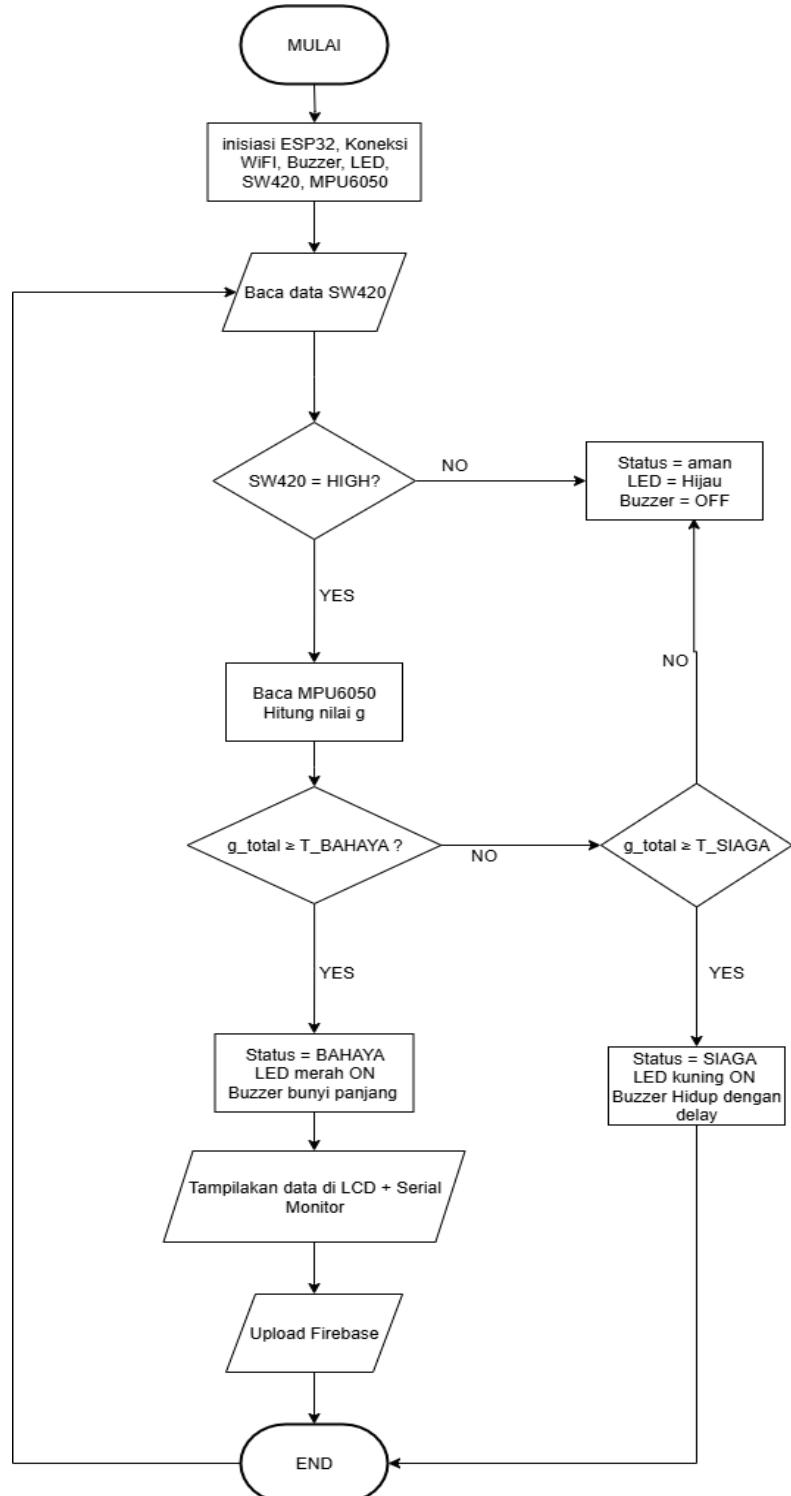


Gambar 3. 6 Rangkaian Keseluruhan

3.3.2 Perancangan Perangkat Lunak

Perangkat lunak sistem ini telah dirancang yang berfungsi sebagai penghubung antara perangkat keras, yang terdiri dari mikrokontroler ESP32, sensor getaran SW-420, sensor percepatan MPU6050, LCD I2C, buzzer, serta modul relay/motor, dengan layanan cloud Firebase Realtime Database dan aplikasi Android. Perangkat lunak ini bertugas mengelola pembacaan data sensor, melakukan pemrosesan awal (pre-processing) untuk mengurangi noise, mengklasifikasikan status getaran, menampilkan informasi pada LCD, mengaktifkan aktuator sesuai kondisi, serta mengirimkan data secara real-time ke Firebase agar dapat diakses dan ditampilkan oleh aplikasi Android, termasuk mengirimkan notifikasi kepada pengguna ketika status bahaya terdeteksi. Perancangan perangkat lunak

diawali dengan pembuatan flowchart. Pada gambar 3.8 akan ditampilkan flowchart dari program yang akan dibuat dalam penelitian ini.



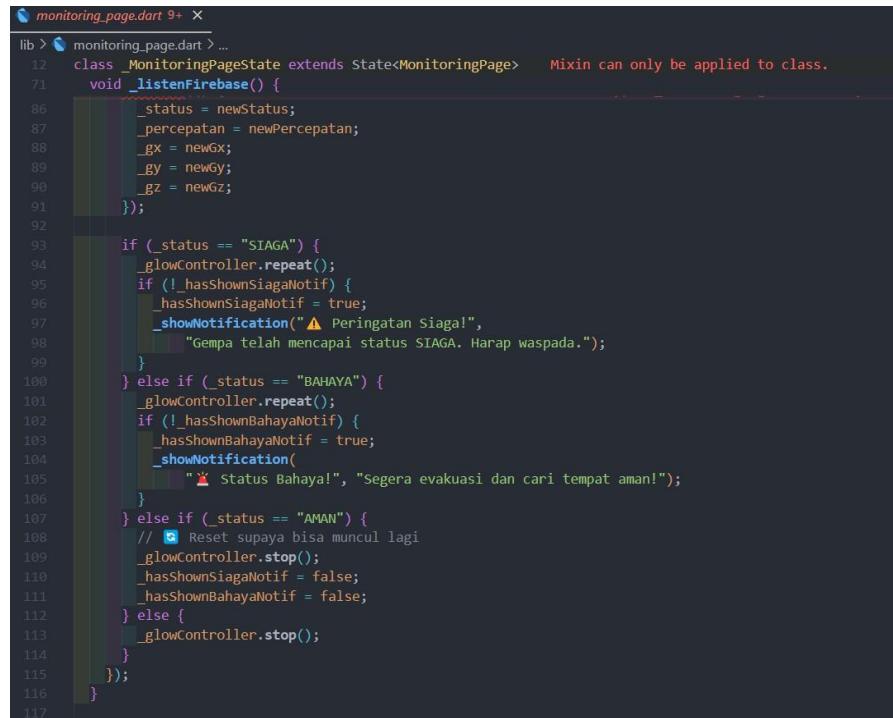
Gambar 3. 7 Flowchart

Perancangan dimulai dengan membuat alur logika sistem dalam bentuk flowchart, kemudian diturunkan menjadi kode program menggunakan Arduino IDE. Sistem dibagi menjadi tiga bagian utama:

1. Inisialisasi Sistem: Mengatur komunikasi I2C untuk LCD dan MPU6050, pin digital untuk SW-420, buzzer, dan LED, serta koneksi WiFi ke Firebase.
2. Pembacaan Sensor dan Logika Keputusan: Sistem membaca data digital dari SW-420 untuk deteksi getaran awal, dan data akselerasi dari MPU6050 jika getaran terdeteksi. Nilai percepatan kemudian dikalkulasi untuk menentukan status sistem (Aman, Siaga, atau Bahaya).
3. Kontrol Output dan Pengiriman Data: Berdasarkan hasil perhitungan, sistem menampilkan status pada LCD, mengatur LED dan buzzer, serta mengirimkan data ke Firebase dan aplikasi Android.

3.3.2.1 Perancangan Antarmuka Pengguna

Desain antarmuka aplikasi Android dibuat dengan mempertimbangkan kemudahan penggunaan, kejelasan informasi, dan responsivitas. Tampilan utama dirancang untuk menampilkan status terkini sistem (Aman, Siaga, Bahaya) dengan kode warna yang berbeda, nilai percepatan dari sensor MPU6050. Tampilan dari antarmuka aplikasi dapat dilihat dari gambar 3.9

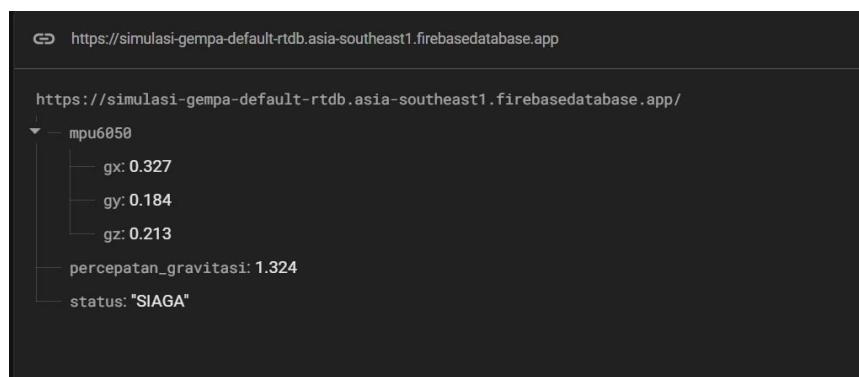


```
monitoring_page.dart 9+ ×
lib > monitoring_page.dart > ...
12  class _MonitoringPageState extends State<MonitoringPage>  Mixin can only be applied to class.
13    void _listenFirebase() {
14      _status = newStatus;
15      _percepatan = newPercepatan;
16      _gx = newGx;
17      _gy = newGy;
18      _gz = newGz;
19    );
20
21    if (_status == "SIAGA") {
22      _glowController.repeat();
23      if (!_hasShownSiagaNotif) {
24        _hasShownSiagaNotif = true;
25        _showNotification("⚠ Peringatan Siaga!",
26                           "Gempa telah mencapai status SIAGA. Harap waspada.");
27      }
28    } else if (_status == "BAHAYA") {
29      _glowController.repeat();
30      if (!_hasShownBahayaNotif) {
31        _hasShownBahayaNotif = true;
32        _showNotification(
33          "⚠ Status Bahaya!", "Segera evakuasi dan cari tempat aman!");
34      }
35    } else if (_status == "AMAN") {
36      // Reset supaya bisa muncul lagi
37      _glowController.stop();
38      _hasShownSiagaNotif = false;
39      _hasShownBahayaNotif = false;
40    } else {
41      _glowController.stop();
42    }
43  );
44}
45
```

Gambar 3. 8 Antarmuka Aplikasi Pengguna

3.3.2.2 Rancangan Firebase Realtime Database

Aplikasi dihubungkan dengan Firebase Realtime Database untuk menerima pembaruan data secara real-time. Listener Firebase dipasang pada path data yang digunakan ESP32, sehingga setiap perubahan status langsung ditampilkan di aplikasi tanpa perlu penyegaran manual. Tampilan dari Realtime Database dapat dilihat pada gambar 3.10



```
https://simulasi-gempa-default-rtbd.firebaseio.com/
```

```
https://simulasi-gempa-default-rtbd.firebaseio.com/
  mpu6050
    gx: 0.327
    gy: 0.184
    gz: 0.213
    percepatan_gravitasi: 1.324
    status: "SIAGA"
```

Gambar 3. 9 Firebase Realtime Database

3.4 Perakitan Sistem

3.4.1 Perakitan Rangkaian Sensor SW-420

Sensor SW-420 merupakan sensor getaran yang digunakan dalam sistem ini sebagai deteksi awal terhadap adanya getaran pada miniatur bangunan. Rangkaian sensor SW-420 terdiri dari tiga pin, yaitu VCC, GND, dan DO (Digital Output). Pada proses perakitan, pin VCC dihubungkan ke sumber daya 3.3V dari ESP32, pin GND dihubungkan ke ground, sedangkan pin DO dihubungkan ke salah satu pin input digital ESP32. Sensor SW-420 bekerja dengan memberikan keluaran digital berupa logika 1 ketika mendeteksi getaran dan 0 ketika tidak ada getaran. Data dari sensor ini diteruskan ke ESP32 untuk kemudian divalidasi lebih lanjut menggunakan sensor MPU6050. Dengan demikian, SW-420 berperan penting sebagai lapisan pertama dalam sistem deteksi getaran.

3.4.2 Perakitan Rangkaian Sensor MPU6050

Sensor MPU6050 bertujuan untuk mengetahui kemampuan sensor dalam mengukur percepatan akibat getaran pada miniatur bangunan dan mengklasifikasikan status sistem menjadi Aman, Siaga, atau Bahaya. Pengujian dilakukan dengan memberikan variasi waktu simulasi pada miniatur bangunan, mulai dari getaran 2 detik hingga maksimal 10 detik waktu simulasi. Nilai percepatan yang terbaca kemudian dicatat dan dibandingkan dengan ambang batas klasifikasi yang telah ditentukan sebelumnya untuk menilai keakuratan pembacaan sensor.

3.4.3 Perakitan LCD I2C

LCD I2C dilakukan untuk memastikan bahwa komponen mampu menampilkan informasi status sistem, nilai percepatan. Pengujian dilakukan dengan mengirimkan data uji dari mikrokontroler ke LCD untuk memeriksa keakuratan tampilan teks, keterbacaan karakter, dan respons tampilan terhadap pembaruan data secara real-time.

3.4.4 Upload Arduino Ide ke ESP 32

Setelah seluruh rangkaian perangkat keras selesai dirakit, tahap selanjutnya adalah mengunggah program ke mikrokontroler ESP32 menggunakan Arduino IDE. Proses ini bertujuan agar ESP32 dapat menjalankan logika sistem yang telah dibuat, mulai dari pembacaan sensor, pemrosesan data, hingga pengiriman data ke Firebase dan penampilan informasi pada perangkat output.

Gambar 3. 11 Uploading Arduino IDE

3.4.5 Konektivitas Flutter dengan Firebase

Konektivitas antara aplikasi Android berbasis Flutter dengan Firebase berfungsi untuk menghubungkan data hasil pembacaan sensor yang dikirim melalui ESP32 ke dalam aplikasi pengguna. Firebase Realtime Database dipilih karena memiliki kemampuan penyimpanan dan pembaruan data secara real-time sehingga setiap perubahan data dari sensor dapat langsung ditampilkan pada aplikasi tanpa jeda waktu yang signifikan. Proses

konektivitas dimulai dengan integrasi Firebase ke dalam proyek Flutter, yaitu dengan menambahkan dependensi Firebase pada file pubspec.yaml serta melakukan konfigurasi dengan file google-services.json yang diperoleh dari Firebase Console. Setelah konfigurasi selesai, aplikasi Flutter dapat terhubung dengan Firebase Realtime Database melalui Firebase Database Plugin. Pada tahap implementasi, aplikasi dibuat untuk membaca data yang tersimpan di Firebase pada node tertentu yang berisi informasi hasil pembacaan sensor MPU6050 (gx, gy, gz, dan nilai percepatan total g) serta status sistem (Aman, Siaga, Bahaya). Data ini ditampilkan secara real-time di dalam aplikasi dengan memanfaatkan fitur Stream Builder atau listener yang selalu aktif mengikuti perubahan data di Firebase. Dengan adanya konektivitas ini, pengguna dapat memantau kondisi sistem deteksi gempa skala ringan melalui aplikasi Android secara langsung, mulai dari melihat grafik atau nilai numerik getaran pada sumbu x, y, dan z, hingga mengetahui status peringatan. Hal ini memastikan bahwa data yang tersimpan di Firebase tidak hanya berfungsi sebagai tempat penyimpanan, tetapi juga sebagai media penghubung antara perangkat keras (ESP32 dan sensor) dengan perangkat lunak (aplikasi Android berbasis Flutter).

```
firebase_options.dart x
lib > firebase_options.dart
1 // File generated by FlutterFire CLI.
2 // ignore_for_file: type=lint
3 import 'package:firebase_core/firebase_core.dart' show FirebaseOptions;
4 import 'package:flutter/foundation.dart'
5   show defaultTargetPlatform, kIsWeb, TargetPlatform;
6
7 /// Default [FirebaseOptions] for use with your Firebase apps.
8 /**
9  * Example:
10 */
11 import 'firebase_options.dart';
12 /**
13  */
14 await Firebase.initializeApp(
15   options: DefaultFirebaseOptions.currentPlatform,
16 );
17
18 class DefaultFirebaseOptions {
19   static FirebaseOptions get currentPlatform {
20     if (kIsWeb) {
21       return web;
22     }
23     switch (defaultTargetPlatform) {
24       case TargetPlatform.android:
25         return android;
26       case TargetPlatform.iOS:
27         return ios;
28       case TargetPlatform.macOS:
29         return macos;
30       case TargetPlatform.windows:
31         return windows;
32       case TargetPlatform.linux:
33         throw UnsupportedError(
34           'DefaultFirebaseOptions have not been configured for linux - '
35           'you can reconfigure this by running the FlutterFire CLI again.');
36     }
37   }
38   default:
39     throw UnsupportedError(
40       'DefaultFirebaseOptions have not been configured for this platform - '
41       'you can reconfigure this by running the FlutterFire CLI again.');
42 }
```

Gambar 3. 12 Konekting Flutter dengan Firebase

3.4.6 Konektivitas ESP 32 ke Firebase

ESP32 berperan sebagai mikrokontroler utama yang mengolah data dari sensor SW-420 dan MPU6050, kemudian mengirimkan hasil olahan data tersebut ke Firebase Realtime Database. Konektivitas ini memungkinkan data sensor yang diperoleh dari alat simulasi gempa dapat diakses secara real-time oleh aplikasi Android berbasis Flutter. Langkah pertama konektivitas adalah konfigurasi jaringan WiFi pada ESP32. Mikrokontroler harus terhubung ke internet agar dapat berkomunikasi dengan Firebase. Proses ini dilakukan dengan menuliskan SSID dan password WiFi pada program Arduino IDE. Setelah ESP32 terkoneksi ke internet, tahap selanjutnya adalah integrasi dengan Firebase menggunakan HTTP protocol atau Firebase library. Data sensor seperti nilai percepatan pada sumbu X, Y,

Z dari MPU6050 serta sinyal getaran dari SW-420 diproses dan dikirimkan ke node tertentu dalam struktur Firebase Realtime Database.

```
1  #include <Wire.h>
2  #include <LiquidCrystal_I2C.h>
3  #include <MPU6050.h>
4  #include <WiFi.h>
5  #include <HTTPClient.h>
6
7  LiquidCrystal_I2C lcd(0x27, 16, 2);
8  MPU6050 mpu;
9
10 // ----- GANTI DATA WIFI & FIREBASE DI SINI -----
11 const char* WIFI_SSID = "realme 7i";
12 const char* WIFI_PASS = "Yasirli221";
13
14 // Firebase Realtime Database host (cukup base URL saja, TANPA https://)
15 const char* FIREBASE_HOST = "simulasi-gempa-default-rtbd.firebaseio.com";
16 // -----
```

Gambar 3. 13 Konekting Esp32 ke Firebase

3.4.4 Perakitan Sistem Keseluruhan

Dilakukan untuk memastikan bahwa seluruh komponen pada Sistem Deteksi Gempa Skala Ringan Berbasis IoT dan Android dapat bekerja secara terintegrasi sesuai fungsi yang telah dirancang. Pengujian ini melibatkan semua perangkat keras, yaitu sensor SW-420, sensor MPU6050, mikrokontroler ESP32, LCD I2C, buzzer, LED, motor DC beserta relay, serta koneksi ke layanan Firebase Realtime Database yang terhubung dengan aplikasi Android. Tujuan dari pengujian ini adalah untuk mengetahui apakah sistem mampu mendeteksi getaran ringan, memproses data secara tepat, mengirimkan informasi ke Firebase, dan menampilkan notifikasi pop-up pada aplikasi Android secara real-time.

3.5 Pengujian Sistem

3.5.1 Pengujian Sensor SW 420

Sensor getar SW-420 dihubungkan ke pin 34 pada ESP 32. Sensor ini berfungsi sebagai detektor awal ketika terjadi getaran pada miniatur bangunan. Pada saat dilakukan simulasi gempa dengan cara menggoyangkan meja uji, sensor SW-420 mendeteksi adanya getaran. Ketika tidak ada getaran, sensor memberikan output logika LOW ke ESP32. Namun, saat getaran terdeteksi, sensor memberikan output logika HIGH

sehingga memicu ESP32 untuk mulai melakukan proses validasi melalui sensor MPU6050.

3.5.2 Pengujian Sensor MPU6050

Sensor akselerometer dan gyroscope **MPU6050** dihubungkan ke ESP32 melalui komunikasi **I2C** dengan pin SDA (D21) dan SCL (D22). Sensor ini digunakan untuk mengukur percepatan pada sumbu X, Y, dan Z ketika terjadi getaran pada miniatur bangunan. Pada saat pengujian, meja uji digoyangkan untuk mensimulasikan gempa. MPU6050 membaca nilai percepatan dari ketiga sumbu, kemudian nilai tersebut diolah oleh ESP32 menjadi **percepatan total (g)** menggunakan rumus:

$$(g) = \sqrt{aX^2 + aY^2 + aZ^2}$$

Nilai g_{total} dibandingkan dengan ambang batas:

- $< 1.2 \text{ g} \rightarrow \text{status AMAN}$
- $1.2 - 1.5 \text{ g} \rightarrow \text{status SIAGA}$
- $\geq 1.5 \text{ g} \rightarrow \text{status BAHAYA}$

Dengan cara ini, MPU6050 berfungsi sebagai validator tingkat getaran untuk menentukan kondisi sistem secara lebih akurat setelah dipicu oleh sensor SW-420.

3.5.3 Pengujian LCD I2C

Pengujian LCD I2C dilakukan dengan memasang modul LCD 16x2 yang sudah menggunakan interface I2C ke mikrokontroler ESP32. Modul ini dihubungkan melalui jalur komunikasi I2C, yaitu pin SDA pada LCD dihubungkan ke pin D21 ESP32, sedangkan pin SCL LCD dihubungkan ke pin D22 ESP32. Pin VCC dihubungkan ke 3.3V ESP32, dan GND ke ground ESP32. Setelah seluruh rangkaian terpasang, LCD I2C difungsikan untuk menampilkan informasi yang diproses oleh ESP32. Proses dimulai dari sensor SW-420 yang memberikan input berupa deteksi adanya getaran awal, lalu sensor MPU6050 membaca data percepatan pada sumbu X, Y, dan Z. Data dari kedua sensor ini diproses oleh ESP32 untuk menghasilkan kondisi sistem. Hasil pemrosesan inilah yang kemudian dikirimkan melalui

komunikasi I2C ke modul LCD, sehingga modul dapat menampilkan informasi yang sesuai dengan instruksi dari mikrokontroler. Pengujian ini berfokus untuk memastikan LCD mampu menerima dan menampilkan data sesuai hasil proses yang berjalan di ESP32.

3.5.4 Pengujian Firebase

Pengujian Firebase Realtime Database dilakukan dengan menghubungkan mikrokontroler ESP32 ke jaringan Wi-Fi dan memastikan komunikasi data dapat terkirim secara real-time ke server Firebase. Pada tahap ini, ESP32 diprogram untuk mengakses database melalui URL Firebase dan API Key yang telah dikonfigurasi sebelumnya. Proses pengujian dimulai dengan memastikan modul Wi-Fi internal ESP32 dapat terkoneksi ke hotspot yang disediakan. Setelah koneksi berhasil, data hasil pembacaan sensor (SW-420 dan MPU6050) akan diproses terlebih dahulu oleh ESP32, kemudian dikirimkan ke Firebase menggunakan protokol HTTP POST atau PUT sesuai struktur database. Firebase dipilih karena kemampuannya menyimpan data secara real-time dan dapat langsung diakses oleh aplikasi Android yang dibangun dengan Flutter. Dalam proses ini, pengujian difokuskan pada kestabilan koneksi, kecepatan pengiriman data, serta struktur data yang tersimpan pada Firebase, tanpa menampilkan hasil keluaran akhir ke aplikasi. Dengan demikian, dapat dipastikan bahwa jalur komunikasi antara ESP32 dan Firebase berjalan dengan baik sebelum dilanjutkan ke tahap integrasi dengan aplikasi Android.

3.5.5 Pengujian Aplikasi

Pengujian Aplikasi Android dilakukan untuk memastikan bahwa data yang dikirim oleh mikrokontroler ESP32 melalui Firebase Realtime Database dapat diterima dan ditampilkan pada aplikasi dengan baik. Aplikasi Android dibangun menggunakan framework Flutter yang terhubung langsung dengan Firebase Realtime Database melalui Firebase SDK. Tahap pengujian dimulai dengan menjalankan aplikasi pada smartphone yang telah dipasang. Setelah aplikasi aktif, dilakukan pemeriksaan koneksi terhadap

Firestore untuk memastikan bahwa aplikasi mampu membaca data secara real-time. Selanjutnya, aplikasi diprogram untuk menampilkan status gempa (Aman, Siaga, Bahaya) serta nilai parameter yang dikirimkan dari sensor melalui database. Proses pengujian berfokus pada bagaimana aplikasi merespons perubahan data di Firestore dan menampilkannya secara langsung pada antarmuka pengguna. Selain itu, dilakukan pula pengujian terhadap fitur notifikasi yang muncul secara otomatis ketika status tertentu terdeteksi di Firestore.

3.5.6 Pengujian Buzzer

Pengujian Buzzer dilakukan untuk memastikan bahwa aktuator suara ini dapat bekerja sesuai instruksi dari mikrokontroler ESP32. Buzzer yang digunakan merupakan buzzer aktif, sehingga dapat berbunyi ketika diberi logika HIGH melalui pin digital. Pada rangkaian, buzzer terpasang pada pin 33 pada ESP32 dengan jalur ground yang terhubung langsung ke GND. Tahap pengujian dimulai dengan memberikan logika hidup-mati (HIGH–LOW) secara bergantian dari program ESP32 untuk melihat respons buzzer. Ketika logika HIGH diberikan, buzzer akan mengeluarkan bunyi sebagai tanda aktif, sedangkan saat logika LOW buzzer dalam kondisi diam. Dalam implementasi sistem, buzzer berfungsi sebagai indikator peringatan dini terhadap kondisi tertentu yang terdeteksi oleh sensor. Proses pengujian ini dilakukan untuk memastikan bahwa buzzer dapat merespons sinyal dari mikrokontroler secara cepat tanpa adanya delay yang signifikan, sehingga nantinya dapat diandalkan sebagai media peringatan suara ketika sistem mendeteksi kondisi gempa.

3.5.7 Pengujian LED

Pengujian LED Indikator dilakukan untuk memastikan bahwa setiap lampu LED dapat menyala sesuai instruksi dari mikrokontroler ESP32. Pada perancangan sistem ini digunakan tiga buah LED, masing-masing mewakili status Aman, Siaga, dan Bahaya. LED hijau sebagai indikator Aman dipasang pada pin 26 ESP32, LED kuning sebagai indikator Siaga pada pin

14, dan LED merah sebagai indikator Bahaya pada pin 27. Seluruh kaki negatif (katoda) LED dihubungkan ke jalur GND, sedangkan kaki positif (anoda) terhubung langsung ke pin digital ESP32. Tahap pengujian dimulai dengan menyalakan masing-masing LED secara bergantian melalui program sederhana yang memberikan logika HIGH untuk menyalakan LED dan logika LOW untuk mematikannya. Proses ini dilakukan untuk memastikan bahwa koneksi kabel, dan pin ESP32 berfungsi dengan baik. Selanjutnya dilakukan pengujian simultan, yaitu menyalakan LED secara berurutan (hijau–kuning–merah) untuk melihat apakah perpindahan status berjalan sesuai instruksi tanpa adanya kendala seperti LED berkedip tidak normal atau tidak menyala. Melalui pengujian ini dapat dipastikan bahwa setiap LED indikator mampu merespons perintah dari ESP32 dengan baik, sehingga nantinya dapat digunakan sebagai tanda visual terhadap status sistem deteksi gempa sesuai kondisi yang terdeteksi.

3.6 Analisis Kerja Sistem

Ketika sistem diaktifkan, seluruh komponen terhubung dengan sumber daya listrik, lalu mikrokontroler ESP32 mulai melakukan inisialisasi terhadap setiap sensor dan modul pendukung, seperti sensor SW-420, sensor MPU6050, LCD I2C, buzzer, LED indikator, serta koneksi Firebase Realtime Database. Pada tahap awal, sensor SW-420 berfungsi sebagai pendekripsi awal adanya getaran pada miniatur bangunan. Jika sensor mendekripsi adanya perubahan getaran yang signifikan, maka sinyal digital akan dikirimkan ke mikrokontroler ESP32. Setelah itu, ESP32 melakukan validasi dengan membaca data percepatan dari sensor MPU6050. Nilai percepatan yang diperoleh dari sumbu X, Y, dan Z diproses dengan menggunakan rumus percepatan total. Hasil perhitungan tersebut dibandingkan dengan ambang batas (threshold) yang telah ditentukan untuk mengklasifikasikan status sistem menjadi Aman, Siaga, atau Bahaya. Selanjutnya, hasil pengolahan data ini ditampilkan secara langsung pada LCD I2C sebagai informasi kondisi sistem. Bersamaan dengan itu, LED indikator akan menyala sesuai status (hijau untuk Aman, kuning untuk

Siaga, dan merah untuk Bahaya). Buzzer juga diaktifkan sebagai peringatan suara, dengan pola bunyi berbeda sesuai tingkat status yang terdeteksi. Selain itu, ESP32 mengirimkan data hasil deteksi ke Firebase Realtime Database melalui koneksi WiFi. Data yang tersimpan di Firebase kemudian dapat diakses secara real-time oleh aplikasi Android yang telah dibuat. Aplikasi menampilkan status deteksi serta memberikan notifikasi kepada pengguna jika terjadi kondisi Siaga atau Bahaya. Dengan demikian, analisis kerja sistem ini menunjukkan bahwa seluruh komponen bekerja secara terintegrasi, mulai dari proses input sensor, pengolahan data pada mikrokontroler, hingga output berupa tampilan, suara, lampu indikator, dan notifikasi aplikasi Android.