

BAB I

PENDAHULUAN

1.1 Latar Belakang

Perkembangan ilmu pengetahuan dan teknologi yang pesat berdampak besar terhadap kehidupan manusia dalam mempelajari dan mengembangkan ilmu pengetahuan serta mempermudah aktivitas sehari-hari (Dotulong dkk., 2022). Semakin berkembangnya teknologi informasi dan beragamnya kebutuhan manusia menghadirkan inovasi dalam bidang teknologi, salah satu diantaranya teknologi *Internet of Things* (IoT) (Prasiani dkk., 2022). *Internet of Things* (IoT) merupakan suatu sistem yang mampu mengolah data secara berkelanjutan, serta memungkinkan monitoring antar perangkat dari jarak jauh menggunakan internet.

Kehadiran teknologi *Internet of Things* (IoT) sangat membantu aktivitas manusia, terutama dalam penggunaan alat elektronika, sistem kendali, serta jaringan komputer (Prasiani dkk., 2022). Perkembangan teknologi komputer sangat menekankan efektivitas dan efisiensi dalam menghasilkan *output* yang optimal, sehingga aktivitas manusia semakin bergantung pada teknologi yang canggih dan otomatis. Dahulu, aktivitas manusia banyak dilakukan secara manual dengan menggunakan perangkat elektronik yang bersifat konvensional. Efektivitas aktivitas manusia tidak hanya dipengaruhi oleh kemajuan teknologi dan peralatan yang semakin canggih, melainkan juga dari perubahan iklim dan cuaca.

Di negara beriklim tropis seperti Indonesia, terdapat dua musim, yaitu musim kemarau dan musim hujan, perubahan musim yang terjadi sangat berdampak pada kesehatan manusia (Ravelliani dkk., 2022). Fenomena pemanasan global menyebabkan pergantian musim sulit diprediksi dengan akurat (Malau, 2022). Pada musim kemarau, paparan sinar matahari berperan penting dalam pembentukan vitamin D yang bermanfaat untuk mempertahankan keseimbangan mineral, mendukung kesehatan tulang, serta memperkuat sistem kekebalan tubuh (Istiqomah dkk., 2024). Waktu optimal memperoleh vitamin D dari sinar matahari berkisar pada pukul 09.00 hingga 10.00 pagi (Fazalina dkk., 2021).

Sebaliknya, musim hujan cenderung meningkatkan kelembaban udara yang memicu pertumbuhan jamur dan mempercepat kerusakan fisik bangunan (Hamsyani dkk., 2021). Tingginya kelembaban udara serta keberadaan jamur tersebut dapat menyebabkan perubahan fisik pada bangunan, seperti munculnya noda jamur pada dinding, pengelupasan cat, hingga terjadi retakan pada struktur bangunan (Nurdin dkk., 2024). Kondisi cuaca yang tidak menentu menjadi tantangan bagi penyedia fasilitas umum yang menerapkan konsep luar ruangan (*Outdoor*), termasuk tempat rekreasi dan area kampus. Salah satu fasilitas yang banyak dimanfaatkan pada fasilitas ruang terbuka adalah gazebo. Gazebo menjadi tempat interaksi sosial, beristirahat, dan belajar, terutama jika memiliki desain yang nyaman dan fungsional (Sari dkk., 2022).

Gazebo memiliki beragam jenis dan model desain, beberapa diantaranya yaitu gazebo dengan konsep natural, kearifan lokal, klasik, industrial, dan lain sebagainya. Pada umumnya, konstruksi gazebo terdiri dari pondasi, tiang sebagai penyangga utama,

lantai yang umumnya terbuat dari bambu dan kayu, serta atap permanen yang terbuat dari bahan seperti jerami (Sari dkk., 2022). Konstruksi atap gazebo yang masih manual menjadi tantangan utama dalam penggunaannya, terutama ketika kondisi cuaca tidak stabil. Hal ini menyebabkan fungsi gazebo sering kali hanya terbatas sebagai tempat berteduh dan kurang memberikan fungsi yang optimal bagi penggunanya.

Berdasarkan permasalahan tersebut, penulis mengembangkan rancangan Gazebo Sehat berbasis *Internet of Things* (IoT) untuk meningkatkan kenyamanan dan kesehatan pengguna. Sistem ini dirancang untuk mengontrol pergerakan atap secara otomatis berdasarkan pembacaan sensor suhu, sensor hujan. Sensor *Light Dependent Resistor* (LDR), dan sensor ultrasonik dengan kondisi yang sudah ditetapkan. Perancangan sistem atap otomatis pada gazebo sehat ini juga dilengkapi dengan integrasi sistem notifikasi menggunakan aplikasi *Blynk*, yang memungkinkan pengguna untuk mengontrol atap gazebo dan mendapatkan informasi mengenai kondisi cuaca tertentu yang terjadi pada gazebo. Inovasi ini diharapkan tidak hanya meningkatkan kenyamanan pengguna, tetapi juga memperpanjang umur gazebo dengan meminimalkan kerusakan akibat cuaca. Dengan demikian, pengembangan Gazebo Sehat menjadi langkah penting dalam menciptakan ruang terbuka yang lebih adaptif, fungsional, dan mendukung kesehatan masyarakat.

1.2 Rumusan Masalah

Perkembangan teknologi, khususnya dalam bidang *Internet of Things* (IoT) dan sistem otomatisasi, menghadirkan peluang inovatif pada berbagai fasilitas luar

ruangan, termasuk gazebo, guna meningkatkan kenyamanan serta efisiensi dalam pengoperasiannya. Pemanfaatan sensor sebagai pengendali sistem otomatisasi, disertai integrasi notifikasi melalui aplikasi mobile, menjadi komponen utama dalam mewujudkan konsep gazebo berbasis IoT yang responsif dan adaptif terhadap kondisi lingkungan dan kesehatan manusia. Berdasarkan latar belakang tersebut, maka rumusan masalah dalam penelitian ini terfokus pada beberapa aspek penting berikut:

1. Bagaimana cara mendesain gazebo sehat dengan atap otomatis yang dapat mendeteksi kondisi suhu, hujan, dan cahaya?
2. Bagaimana cara kerja sensor hujan, sensor suhu, sensor LDR, dan sensor ultrasonik dalam menjalankan sistem otomatisasi atap gazebo?
3. Bagaimana integrasi aplikasi *Blynk* sebagai sistem notifikasi dan pengontrolan jarak jauh dalam pengoperasian atap gazebo?
4. Bagaimana penggunaan gazebo sehat berbasis IoT terhadap kenyamanan dan kesehatan pengguna di ruang terbuka?

1.3 Batasan Masalah

Dalam upaya menjaga fokus penelitian dan menghindari peluasan ruang lingkup yang berlebihan, penulis menetapkan sejumlah batasan masalah sebagai acuan dalam proses perancangan dan pengembangan sistem otomatisasi gazebo sehat berbasis *Internet of Things* (IoT). Penetapan masalah ini bertujuan agar penelitian dapat berjalan secara lebih terarah, sistematis, dan sesuai tujuan yang sudah ditetapkan. Adapun batasan permasalahan yang diangkat adalah sebagai berikut:

1. Penelitian ini hanya fokus pada perancangan gazebo sehat dengan atap otomatis berbasis IoT, tanpa membahas nilai estetika gazebo.
2. Sensor yang digunakan terbatas pada sensor suhu, sensor hujan, sensor cahaya, dan ultrasonik.
3. Integrasi sistem dengan aplikasi *Blynk* hanya mencakup notifikasi kondisi beserta informasi suhu dan pengontrolan atap secara manual tanpa monitoring suhu dan cuaca secara *real-time*.
4. Faktor kenyamanan pengguna pada gazebo hanya berdasarkan sistem atap otomatis yang responsif dan kebebasan pengguna dalam mengontrol atap dan faktor pendukung kesehatan pada gazebo ini hanya terfokus pada dampak paparan sinar matahari pada jam tertentu.

1.4 Hipotesa

Penerapan teknologi *Internet of Things* (IoT) pada desain gazebo sehat dengan atap otomatis diharapkan mampu mendeteksi perubahan cuaca secara akurat sehingga dapat meningkatkan kenyamanan pengguna. Selain berfokus pada kenyamanan, gazebo ini juga dirancang untuk memperhatikan kesehatan pengguna yang berkaitan dengan dampak paparan sinar matahari bagi kesehatan pada waktu tertentu. Atap otomatis pada gazebo berfungsi untuk melindungi penggunanya dari penyakit yang mungkin disebabkan oleh faktor cuaca. Sistem atap otomatis ini juga diharapkan dapat memperpanjang umur struktural dan usia pemakaian gazebo dengan sistem yang memberikan perlindungan terhadap kerusakan akibat cuaca ekstrem.

1.5 Tujuan Penelitian

Berdasarkan rumusan masalah yang telah disampaikan, penelitian ini memiliki beberapa tujuan utama yang ingin dicapai untuk mengembangkan dan mengoptimalkan sistem gazebo sehat dengan atap otomatis berbasis *Internet of Things* (IoT). Tujuan tersebut disusun agar penelitian ini dapat memberikan manfaat yang jelas dan relevan dari segala aspek baik fungsional, teknologis, maupun kenyamanan pengguna. Adapun tujuan penelitian ini adalah sebagai berikut:

1. Mendesain gazebo sehat dengan atap otomatis yang dapat mendeteksi suhu dan cuaca.
2. Mengidentifikasi cara kerja sensor yang dimanfaatkan untuk sistem gazebo sehat dengan atap otomatis.
3. Mengintegrasikan aplikasi *Blynk* sebagai media pengontrolan jarak jauh dan sistem notifikasi dalam pengoperasian atap gazebo serta penyampaian informasi kondisi tertentu kepada pengguna.
4. Menyesuaikan gazebo berbasis IoT terhadap kenyamanan dan kesehatan pengguna di ruang terbuka.

1.6 Manfaat Penelitian

1.6.1 Bagi Penulis

Penelitian ini tidak hanya berkontribusi terhadap pengembangan ilmu pengetahuan tetapi juga memberikan manfaat langsung bagi penulis sebagai pelaksana penelitian. Melalui proses perancangan, pengujian, dan penerapan sistem, penulis

memperoleh banyak pengalaman dari penelitian ini. Adapun Manfaat yang dapat dirasakan penulis dari penelitian ini adalah sebagai berikut:

1. Menambah pengetahuan dan keterampilan dalam desain sistem berbasis *IoT*.
2. Menciptakan inovasi baru pada bangunan gazebo yang baik untuk kesehatan
3. Memberikan pengalaman praktis dalam penelitian dan pengembangan teknologi.
4. Memenuhi syarat untuk untuk mendapatkan gelar sarjana sistem komputer

1.6.2 Bagi Program Studi

Dalam melaksanakan penelitian ini, tentunya penulis tidak bekerja secara individu, melainkan memperoleh dukungan dari lingkungan akademik yang turut berperan secara tidak langsung dan juga merasakan dampak positif dari kegiatan penelitian ini. Dari penelitian ini, penulis berharap agar program studi juga memperoleh berbagai manfaat sebagai berikut:

1. Menjadi referensi bagi pengembangan kurikulum dalam matakuliah yang berkaitan dengan project *Internet of Things (IoT)*.
2. Meningkatkan reputasi program studi melalui inovasi yang diterapkan dalam proyek.

1.6.3 Bagi Masyarakat

Penelitian ini diharapkan tidak hanya memberikan kontribusi dalam bidang akademik dan pengembangan teknologi, tetapi juga menghadirkan manfaat nyata bagi masyarakat luas. Melalui penelitian ini penulis berupaya menghadirkan inovasi yang

dapat meningkatkan kenyamanan, keamanan, serta kualitas hidup masyarakat dalam beraktivitas di ruang terbuka. Menyediakan solusi ruang terbuka yang lebih nyaman dan aman untuk beraktivitas di luar ruangan. Dari penelitian manfaat yang dapat dirasakan oleh masyarakat adalah sebagai berikut:

1. Mendorong kesadaran akan pentingnya teknologi dalam meningkatkan kualitas hidup dan interaksi sosial di lingkungan masyarakat.
2. Menerapkan sarana yang menunjang kesehatan dan bermanfaat bagi masyarakat.
3. Memperkenalkan inovasi baru yang bermanfaat bagi fasilitas umum dan daya tarik dalam dunia usaha.

BAB IV

ANALISA DAN HASIL

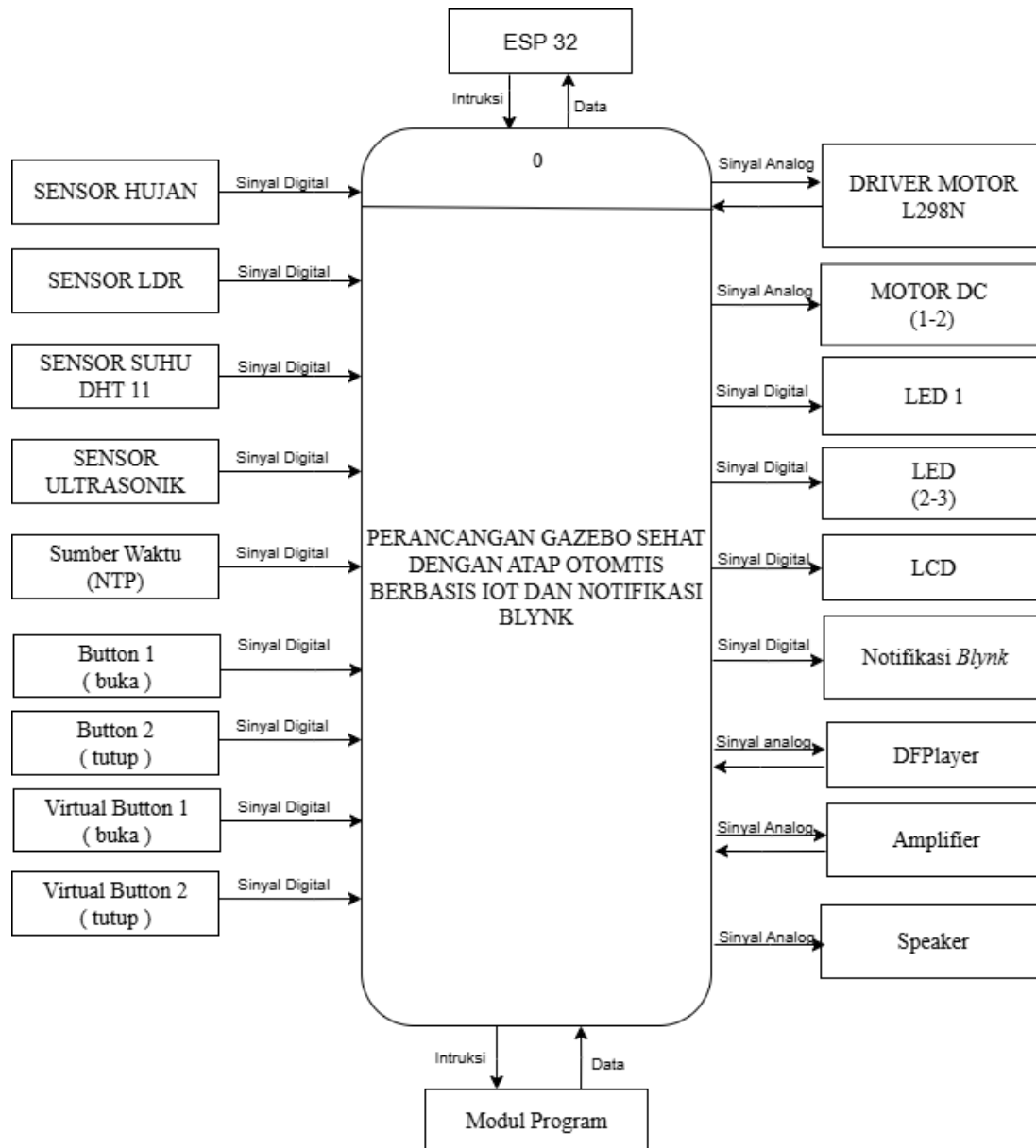
4.1 Desain Sistem Secara Umum

Analisa sistem memerlukan pendefinisian terhadap sistem yang akan dirancang secara menyeluruh. Dalam melakukan perancangan harus memiliki gambaran sistem yang jelas mengenai ruang lingkup pembahasan masalah, dimana medianya adalah *context diagram*, *data flow diagram*, *blok diagram*, dan *Flowchart*. Setiap diagram yang telah disebutkan memiliki fungsi yang berbeda tetapi saling melengkapi dalam memberikan pemahaman yang komprehensif tentang sistem yang sedang dianalisis. *Context diagram* memberikan gambaran umum tentang interaksi sistem dengan entitas eksternal. *Data flow diagram* menggambarkan aliran data di dalam sistem. Blok diagram menyajikan komponen utama sistem dan hubungan antar komponen. *Flowchart* menunjukkan langkah-langkah proses secara berurutan. Untuk lebih jelasnya dapat dilihat pada uraian berikut:

4.1.1 Context Diagram

Context diagram adalah definisi dari sistem yang dirancang secara menyeluruh. *Context diagram* berfungsi sebagai media yang terdiri dari suatu proses dan beberapa *external entity*. Dengan adanya *context diagram* dapat memudahkan analisis terhadap

sistem yang dirancang secara keseluruhan. *Context diagram* pada sistem ini dapat dilihat pada gambar 4.1.



Gambar 4.1 Context Diagram

Sistem ini berinteraksi dengan beberapa *entity* yaitu ESP32, Sensor Hujan, Sensor LDR, Sensor Suhu, NTP, Sensor Ultrasonik, *Push Button*, *Virtual Button*,

Motor DC, LCD, LED, DF Player, Amplifier, *Speaker* dan Aplikasi *Blynk*. Selanjutnya *entity* tersebut akan dijelaskan sebagai berikut.

1. ESP32

ESP32 berfungsi sebagai pusat mikrokontroler untuk pengelolaan data dan intruksi.

2. Sensor Hujan

Sensor Hujan berfungsi sebagai pendeteksi turunnya air hujan untuk menentukan jika kondisi cuaca disekitar gazebo tengah turun hujan.

3. Sensor LDR

Sensor LDR berfungsi mendeteksi perubahan intensitas cahaya untuk mengetahui apakah cuaca disekitar gazebo terang atau gelap.

4. Sensor Suhu

Sensor Suhu berfungsi untuk mendeteksi perubahan suhu udara pada gazebo.

5. Sumber Waktu (NTP)

Sumber Waktu (NTP) berfungsi untuk menyediakan informasi waktu secara *real-time* kepada sistem melalui internet.

6. Sensor Ultrasonik

Sensor Ultrasonik berfungsi membaca jarak untuk mendeteksi adanya pengunjung yang memasuki gazebo.

7. *Push Button*

Push Button berfungsi untuk mengirimkan logika untuk melakukan pengontrolan buka tutup atap secara manual.

8. *Virtual Button*

Virtual Button berfungsi untuk mengirimkan logika seperti *Push Button* secara virtual melalui aplikasi *Blynk* untuk melakukan pengontrolan buka tutup atap secara manual.

9. Motor DC

Motor DC berfungsi untuk menggerakkan atap gazebo agar dapat terbuka dan tertutup otomatis.

10. LCD 16x4

LCD (*Liquid Qrystal Display*) digunakan untuk menampilkan jam, perubahan kondisi cuaca yang sedang terjadi pada gazebo, monitoring suhu, mode pengontrolan sistem gazebo, serta menampilkan ucapan selamat datang kepada pengguna yang datang.

11. LED

LED (*Light Emiting Diode*) digunakan pada saat sistem atap otomatis pada Gazebo sedang berjalan dan sebagai lampu penerangan.

12. Notifikasi *Blynk*

Notifikasi *Blynk* berfungsi untuk memberikan notifikasi jarak jauh kepada pemilik Gazebo terkait perubahan kondisi cuaca yang sedang terjadi, suhu udara pada gazebo, serta pemberitahuan bahwa ada orang yang berkunjung ke gazebo.

13. Modul Program

Modul program digunakan untuk melakukan pembacaan terhadap pin-pin mikrokontroler serta memberikan intruksi untuk mengaktifkan pin-pin *output*.

Modul program menggunakan bahasa pemrograman arduino IDE untuk mengontrol semua proses yang dilakukan oleh sistem.

14. Driver L298n

Driver L298n digunakan untuk menggerakkan dan mengendalikan motor DC untuk membuka dan menutup atap pada Gazebo.

15. *DFPlayer*

DFPlayer digunakan sebagai modul untuk memutar audio ucapan selamat datang kepada pengunjung gazebo dalam bentuk file suara dengan format mp3 yang tersimpan pada media penyimpanan seperti *microSD* yang dikendalikan oleh ESP32.

16. Modul amplifier

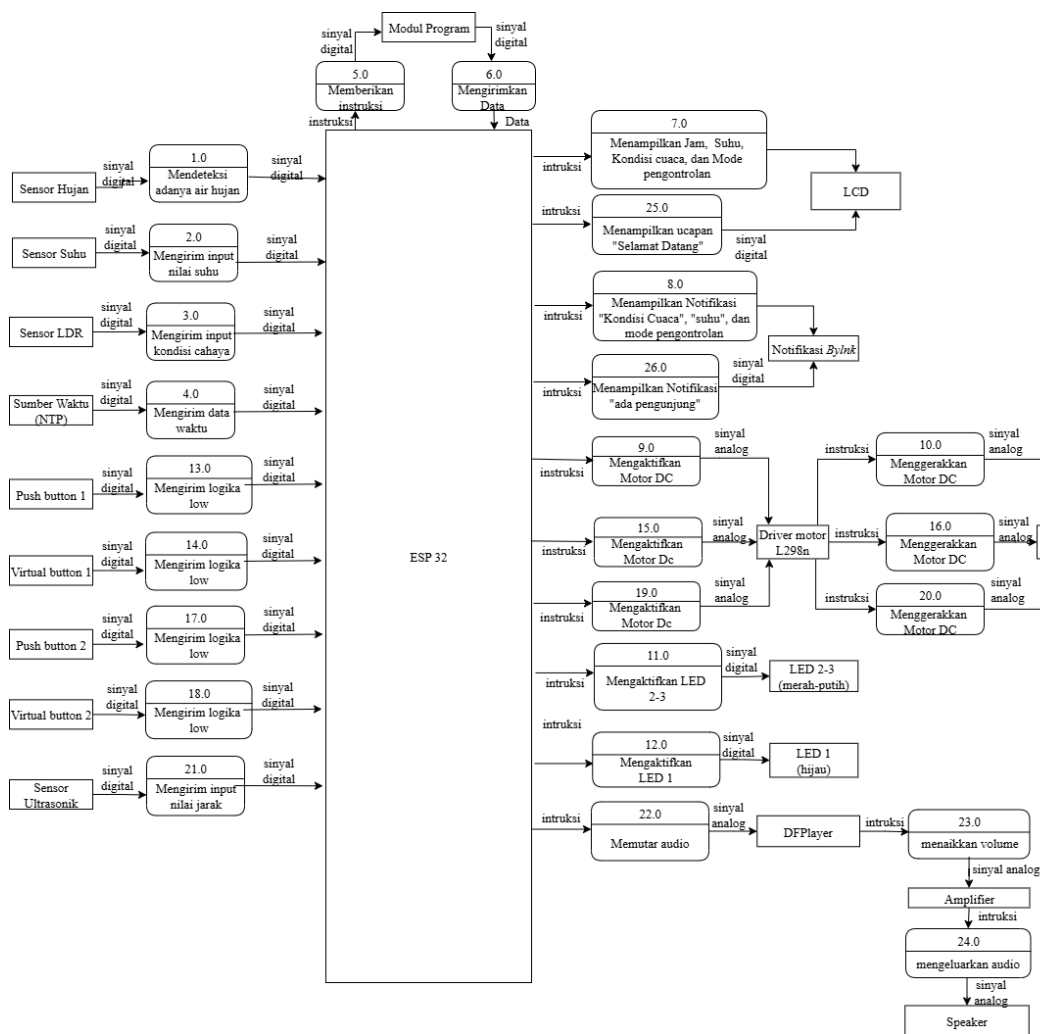
Modul amplifier berfungsi sebagai penguat sinyal audio yang dihasilkan oleh *DFPlayer* sebelum diteruskan ke *speaker* agar terdengar lebih jelas oleh pengunjung gazebo.

17. *Speaker*

Speaker pada sistem ini digunakan untuk menyampaikan informasi dan ucapan selamat datang kepada pengunjung yang memasuki bangunan gazebo.

4.1.2 Data Flow Diagram

Data Flow Diagram adalah gambaran lebih rinci dari alat yang akan dirancang. Gambar *data flow diagram* dari sistem atap otomatis pada gazebo sehat dapat dilihat pada gambar 4.2.



Gambar 4.2 Data Flow Diagram

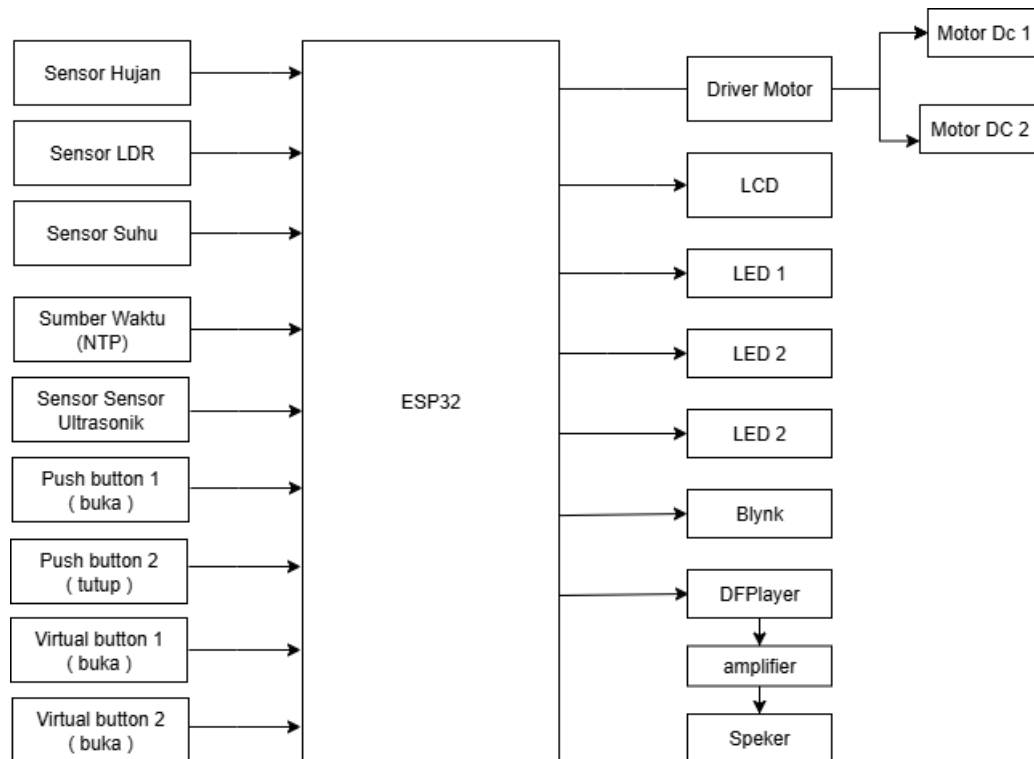
Berdasarkan gambar diatas dapat dijelaskan bahwa rancang bangun sistem atap otomatis pada gazebo sehat, memiliki beberapa proses urutan sebagai berikut:

1. Proses yang pertama dimulai dari Sensor hujan mendeteksi turunnya hujan dan mengirimkan input ke ESP32 (1.0).
2. Sensor suhu mendeteksi suhu udara dan mengirimkan nilai input ke ESP32 (2.0).
3. Sensor LDR mengirimkan input nilai intensitas cahaya ke ESP32 (3.0).
4. Sumber waktu (NTP) mengirimkan informasi waktu ke ESP32 (4.0).
5. ESP32 mengirim perintah ke modul program untuk melakukan proses (5.0).
6. Hasil dari proses modul program dikirimkan kembali ke ESP32, setelah itu dihasilkan data yang akan diolah (6.0).
7. ESP32 mengirim perintah ke LCD untuk menampilkan informasi jam, kondisi cuaca, dan suhu udara dan mode pengontrolan sistem gazebo (7.0).
8. ESP32 mengirim perintah pada aplikasi *Blynk* untuk menampilkan notifikasi tentang kondisi cuaca, suhu udara, serta mode pengontrolan sistem gazebo (8.0).
9. ESP32 mengirimkan perintah ke *Motor Driver* L298n untuk mengaktifkan motor DC (9.0).
10. *Motor Driver* L298n memberikan perintah kepada motor DC untuk aktif (10.0).
11. ESP32 mengirim perintah kepada LED 2 dan 3 untuk menyala saat sistem menutup atap gazebo (11.0).
12. ESP32 mengirim perintah kepada LED 1 untuk menyala saat sistem membuka atap gazebo (12.0).
13. *Push Button* 1 memberikan input sinyal logika *low* ke ESP32 (13.0).
14. *Virtual Button* 1 memberikan input sinyal logika *low* ke ESP32 (14.0).

15. ESP32 mengirimkan perintah ke *Motor Driver* L298n untuk mengaktifkan motor DC (15.0).
16. *Motor Driver* L298n memberikan perintah kepada motor DC untuk aktif (16.0).
17. *Virtual Button* 1 memberikan input sinyal logika *low* ke ESP32 (17.0).
18. *Push Button* 2 memberikan input sinyal logika *low* ke ESP32 (18.0).
19. ESP32 mengirimkan perintah ke *Motor Driver*L298n untuk mengaktifkan motor DC (19.0).
20. *Motor Driver* L298n memberikan perintah kepada motor DC untuk aktif (20.0).
21. Sensor Ultrasonik mengirimkan input nilai jarak ke ESP32 (21.0).
22. ESP32 mengirim perintah pada *DFPlayer* untuk memutar file audio berupa ucapan selamat datang ketika sistem mendeteksi adanya pengunjung yang memasuki bangunan gazebo (22.0).
23. *DFPlayer* meneruskan sinyal audio yang dihasilkan ke modul amplifier untuk dilakukan proses penguatan sinyal suara agar volume suara menjadi lebih optimal (23.0).
24. Amplifier mengirimkan sinyal audio yang telah diperkuat ke *speaker*, sehingga *speaker* dapat mengeluarkan suara dengan kualitas yang lebih baik sebagai output sistem (24.0).
25. ESP32 mengirim perintah ke LCD untuk menampilkan ucapan selamat datang pada pengguna yang berkunjung (25.0).
26. ESP32 mengirim perintah pada aplikasi *Blynk* untuk menampilkan notifikasi pemberitahuan tentang pengunjung yang memasuki gazebo (26.0).

4.1.3 Blok Diagram

Blok diagram merupakan desain dari alat yang dibuat merupakan gambaran dari alat secara keseluruhan. Dengan adanya desain ini maka prinsip kerja dari alat serta komponen dari sistem dapat diketahui dengan jelas. Desain dari alat dapat digambarkan menggunakan blok diagram yang dapat dilihat pada gambar 4.3 berikut.



Gambar 4.3 Blok Diagram

4.2 Prinsip Kerja Sistem

Secara umum alat ini terdiri dari rangkaian komponen elektronika dan sistem yang berkerja secara terstruktur berdasarkan logika yang telah dibuat untuk

menggerakkan atap pada gazebo secara otomatis. Berikut adalah cara kerja sistem yang lebih terstruktur.

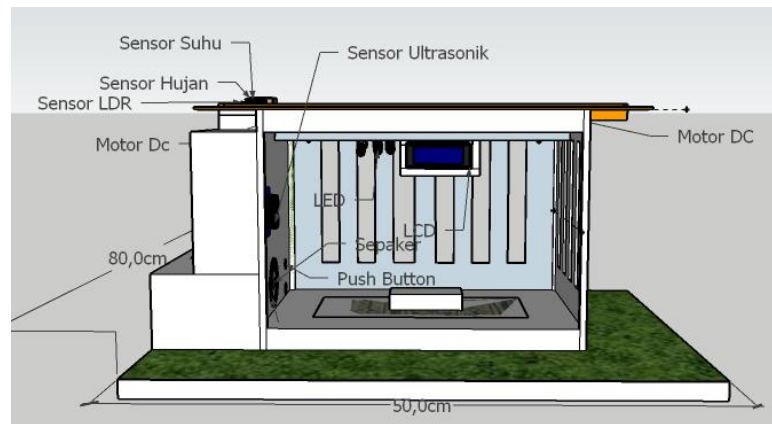
1. ESP32 digunakan untuk mengontrol seluruh rangkaian agar dapat berjalan dengan baik.
2. Sensor hujan digunakan untuk mendeteksi adanya curah hujan dan mengirimkan data ke ESP32.
3. Sensor suhu digunakan untuk mengukur suhu udara disekitar gazebo dan mengirimkan data *real-time* ke ESP32.
4. Sensor LDR digunakan untuk mendeteksi perubahan intensitas cahaya di sekitar gazebo dan mengirimkan data ke ESP32 apakah cuaca sedang gelap atau terang.
5. NTP digunakan untuk memberikan informasi waktu secara *real-time* ke ESP32.
6. Sensor ultrasonik digunakan untuk mendeteksi keberadaan tamu pada tangga masuk gazebo.
7. ESP32 akan mengendalikan *Motor Driver* untuk membuka atau menutup atap gazebo secara otomatis berdasarkan kondisi yang diberikan oleh sensor.
8. Saat terjadi perubahan kondisi cuaca buruk seperti hujan, panas terik dengan suhu $>32^{\circ}\text{C}$, atau gelap, ESP32 mengirim perintah kepada LCD untuk menampilkan informasi yang berisi jam, kondisi cuaca, suhu dan mode pengontrolan gazebo dan notifikasi *Blynk* yang berisi informasi kondisi cuaca, suhu udara, dan mode pengontrolan gazebo, serta menyalakan LED II sebagai indikator berwarna merah dan LED III sebagai lampu penerangan gazebo.

9. Saat semua kondisi cuaca baik yang menandakan cuaca cerah, ESP32 mengirim perintah kepada LCD untuk menampilkan informasi yang berisi jam, kondisi cuaca, suhu dan mode pengontrolan gazebo dan notifikasi *Blynk* yang berisi informasi kondisi cuaca, suhu udara, dan mode pengontrolan gazebo, serta menyalakan LED I sebagai indikator berwarna hijau.
10. Ketika waktu menunjukkan pukul 06.00 pagi hingga pukul 10.00 pagi maka ESP 32 akan mengirim perintah untuk menggerakkan motor DC agar membuka atau menutup atap sesuai kondisi cuaca, jika waktu sudah melewati pukul 10.00 pagi, maka atap akan tertutup otomatis dalam kondisi cuaca apapun namun sistem akan tetap menampilkan informasi perubahan kondisi cuaca melalui LCD dan notifikasi *Blynk*.
11. Ketika sensor ultrasonik mendeteksi tamu, ESP32 mengirim perintah kepada LCD untuk menampilkan pesan “Selamat Datang”, menampilkan notifikasi “Ada Pengunjung” pada notifikasi *Blynk* dan juga memerintahkan *DFPlayer* untuk memutar audio berupa ucapan “selamat datang” kepada pengunjung yang dikeluarkan melalui *speaker*.
12. Ketika *Push Button 1* atau *Virtual Button 1* ditekan, ESP akan mengendalikan *Motor Driver* untuk membuka atap gazebo dan pengontrolan atap beralih ke mode manual.
13. Ketika *Push Button 2* atau *Virtual Button 2* ditekan, ESP akan mengendalikan *Motor Driver* untuk menutup atap gazebo dan pengontrolan atap beralih ke mode manual.

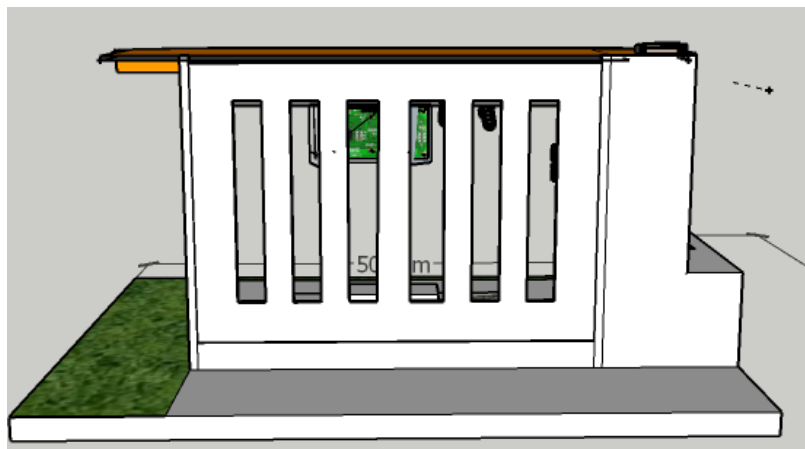
14. Saat *Push Button* 1 dan 2 atau *Virtual Button* 1 dan 2 ditekan bersamaan, ESP akan mengirim perintah untuk sistem kembali ke otomatis.

4.3 Rancang Fisik Alat

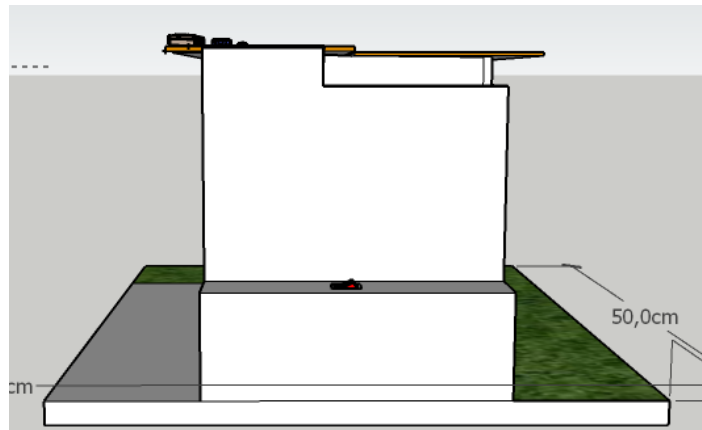
Perancangan alat ini merupakan langkah awal dari pemasangan dan menganalisa permasalahan yang dihadapi berdasarkan literatur yang menunjang perancangan alat. Rancangan fisik alat yang dibuat dapat dilihat pada gambar berikut.



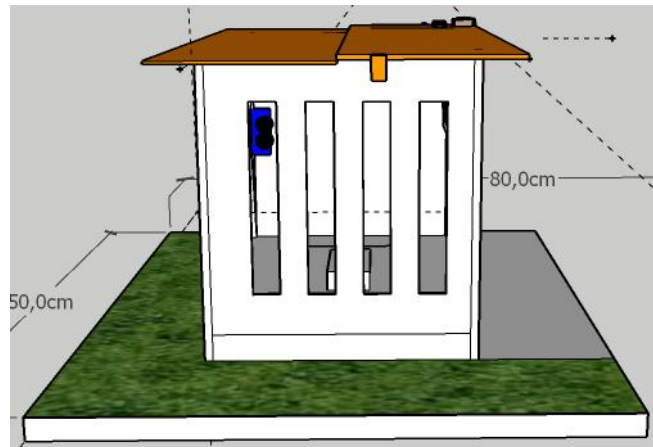
Gambar 4.4 Rancangan Tampak Depan



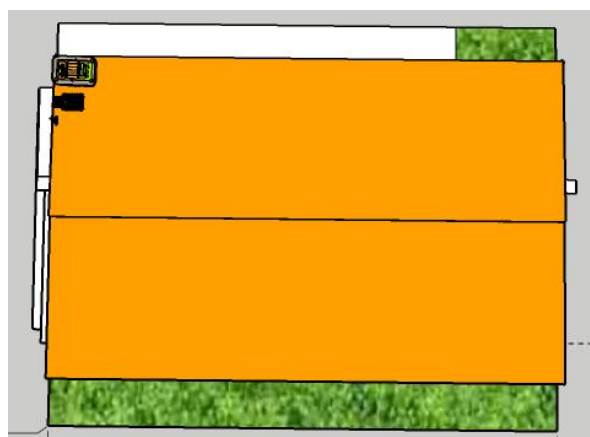
Gambar 4.5 Rancangan Tampak Belakang



Gambar 4.6 Rancangan Samping Kiri



Gambar 4.7 Rancangan Samping Kanan



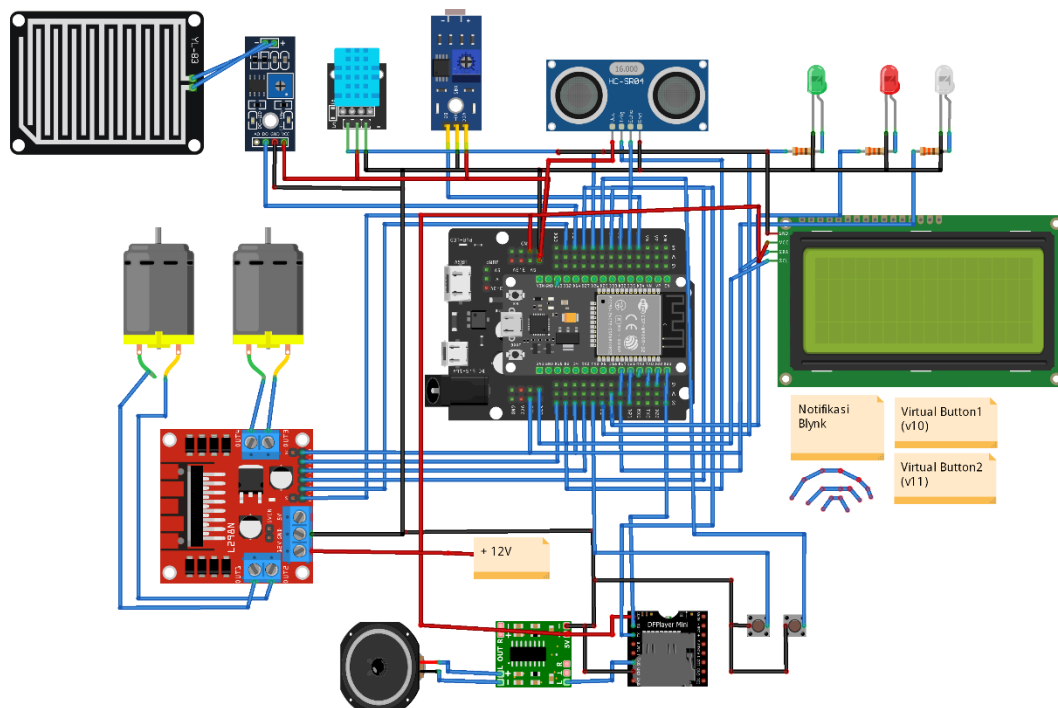
Gambar 4 8 Rancangan Tampak Atas

4.4 Desain Sistem Terperinci

Desain dari sistem yang dibuat merupakan gambaran dari sistem secara keseluruhan. Dengan adanya desain ini maka prinsip kerja dari sistem serta komponen-komponen dari sistem yang digunakan akan dapat dilihat dengan jelas.

4.4.1 Sistem Minimum ESP32

Rangkaian sistem minimum berfungsi untuk menjalankan ESP32 agar dapat bekerja sesuai dengan yang dibutuhkan dimana perancangannya bertujuan untuk mempermudah penggunaan alat. Rangkaian modul program ESP32 dapat dilihat pada gambar 4.9.



Gambar 4.9 Rangkaian ESP32

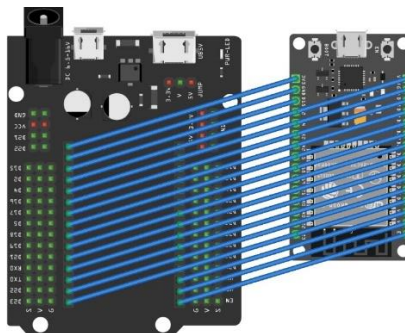
Penjelasan pin-pin yang terhubung pada ESP32 adalah sebagai berikut:

1. Semua pin GPIO pada ESP32 terhubung ke semua pin ESP32 *Expansion board*.
2. Pin D14 pada ESP32 *Expansion board* terhubung ke pin D0 pada Sensor Hujan.
3. Pin D26 pada ESP32 *Expansion board* terhubung ke pin DAT pada Sensor Suhu.
4. Pin D34 pada ESP32 *Expansion board* terhubung ke pin A0 pada Sensor LDR.
5. Pin D2 pada ESP32 *Expansion board* terhubung ke pin Trigger pada Sensor Ultrasonik.
6. Pin D35 pada ESP32 *Expansion board* terhubung ke pin Echo pada Sensor Ultrasonik.
7. Pin D32 pada ESP32 *Expansion board* terhubung ke pin ENA pada *Motor Driver*.
8. Pin D12 pada ESP32 *Expansion board* terhubung ke pin IN1 pada *Motor Driver*.
9. Pin D33 pada ESP32 *Expansion board* terhubung ke pin IN2 pada *Motor Driver*.
10. Pin D16 pada ESP32 *Expansion board* terhubung ke pin IN3 pada *Motor Driver*.
11. Pin D15 pada ESP32 *Expansion board* terhubung ke pin IN4 pada *Motor Driver*.
12. Pin D4 pada ESP32 *Expansion board* terhubung ke pin ENB pada *Motor Driver*.
13. Pin D21 pada ESP32 *Expansion board* terhubung ke pin SDA pada LCD.
14. Pin D22 pada ESP32 *Expansion board* terhubung ke pin SCL pada LCD.
15. Pin D5 pada ESP32 *Expansion board* terhubung ke pin positif pada LED.
16. Pin D18 pada ESP32 *Expansion board* terhubung ke pin positif pada LED.
17. Pin D19 pada ESP32 *Expansion board* terhubung ke pin positif pada LED.
18. Pin D27 pada ESP32 *Expansion board* terhubung ke pin TX *DFPlayer* mini.
19. Pin D23 pada ESP32 *Expansion board* terhubung ke pin RX *DFPlayer* mini.

20. Pin D25 pada ESP32 *Expansion board* terhubung ke pin positif *Push Button*.
21. Pin D17 pada ESP32 *Expansion board* terhubung ke pin positif *Push Button*.
22. Modul WiFi pada ESP32 terhubung ke fitur notifikasi pada aplikasi *Blynk*.
23. Modul WiFi pada ESP32 terhubung ke pin V10 *Virtual Button* Pada aplikasi *Blynk*.
24. Modul WiFi pada ESP32 terhubung ke pin V11 *Virtual Button* Pada aplikasi *Blynk*.
25. Pin 5V pada ESP32 *Expansion board* terhubung ke pin VCC seluruh komponen.
26. Pin GND pada ESP32 *Expansion board* terhubung ke pin GND seluruh komponen.

4.4.2 Rangkaian ESP32 *Expansion board*

Pada perancangan alat ini ESP32 *Expansion board* digunakan sebagai papan perantara *output* dari ESP32 agar lebih mudah digunakan dan terstruktur. Rangkaian dari ESP32 *Expansion board* dapat dilihat pada gambar 4.10 berikut.



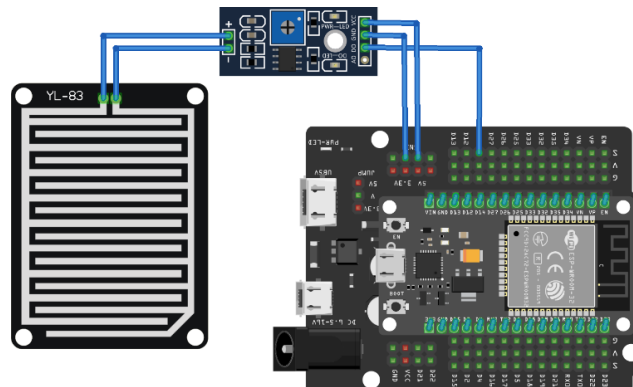
Gambar 4. 10 ESP32 *Expansion Board*

Penjelasan pin pada ESP32 *Expansion board* yang terhubung pada ESP32 adalah sebagai berikut:

1. Semua pin GPIO pada ESP32 terhubung ke semua pin ESP32 *Expansion board*.

4.4.3 Rangkaian Sensor Hujan

Pada perancangan alat ini sensor hujan digunakan sebagai input untuk mendeteksi adanya air hujan yang membasahi atap gazebo. Rangkaian dari sensor hujan dapat dilihat pada gambar 4.11 berikut.



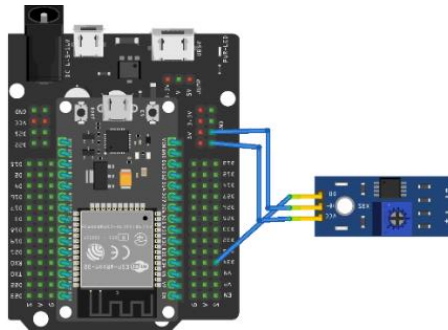
Gambar 4. 11 Rangkaian Sensor Hujan

Penjelasan pin pada sensor hujan yang terhubung pada ESP32 adalah sebagai berikut:

1. Pin D14 pada ESP32 *Expansion board* terhubung ke pin D0 pada Sensor Hujan.
2. Pin 5V pada ESP32 *Expansion board* terhubung ke pin VCC pada sensor Hujan.
3. Pin GND pada ESP32 *Expansion board* terhubung ke pin GND pada sensor Hujan.

4.4.4 Rangkaian Sensor Sensor *Light Dependent Resistor (LDR)*

Pada perancangan alat ini sensor LDR digunakan sebagai input untuk mendeteksi adanya perubahan cahaya disekitar gazebo. Rangkaian dari sensor LDR dapat dilihat pada gambar 4.12 berikut.



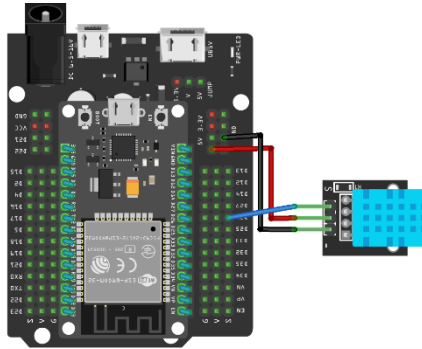
Gambar 4. 12 Rangkaian Sensor LDR

Penjelasan pin pada sensor LDR yang terhubung pada ESP32 adalah sebagai berikut.

1. Pin D34 pada ESP32 *Expansion board* terhubung ke pin A0 pada Sensor LDR.
2. Pin 5V pada ESP32 *Expansion board* terhubung ke pin VCC pada Sensor LDR.
3. Pin GND pada ESP32 *Expansion board* terhubung ke pin GND pada Sensor LDR.

4.4.5 Rangkaian Sensor Suhu

Pada perancangan alat ini sensor suhu digunakan sebagai input untuk mengukur suhu udara disekitar gazebo. Rangkaian dari sensor suhu dapat dilihat pada gambar 4.13 berikut.



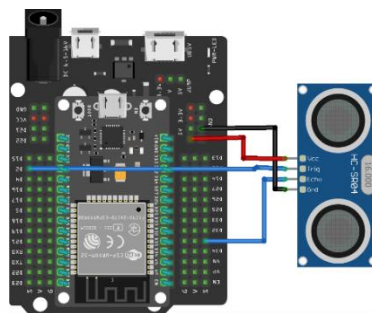
Gambar 4. 13 Rangkaian Sensor Suhu

Penjelasan pin pada sensor hujan yang terhubung pada ESP32 adalah sebagai berikut.

1. Pin D26 pada ESP32 *Expansion board* terhubung ke pin DAT pada Sensor Suhu.
2. Pin 5V pada ESP32 *Expansion board* terhubung ke pin VCC pada Sensor Suhu
3. Pin GND pada ESP32 *Expansion board* terhubung ke pin GND pada Sensor Suhu.

4.4.6 Rangkaian Sensor Ultrasonik

Pada perancangan alat ini sensor ultrasonik digunakan sebagai input untuk mendeteksi adanya pengunjung yang memasuki bangunan gazebo. Rangkaian dari sensor ultrasonik dapat dilihat pada gambar 4.14 berikut.



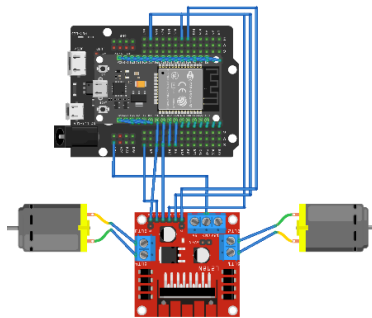
Gambar 4. 14 Rangkaian Sensor Ultrasonik

Penjelasan pin pada sensor ultrasonik yang terhubung pada ESP32 adalah sebagai berikut.

1. Pin D2 pada ESP32 *Expansion board* terhubung ke pin Trigger pada Sensor Ultrasonik.
2. Pin D35 pada ESP32 *Expansion board* terhubung ke pin Echo pada Sensor Ultrasonik.
3. Pin 5V pada ESP32 *Expansion board* terhubung ke pin VCC pada Sensor Ultrasonik.
4. Pin GND pada ESP32 *Expansion board* terhubung ke pin GND pada Sensor Ultrasonik.

4.4.7 Rangkaian Motor DC dan *Motor Driver*

Pada perancangan alat ini Motor DC digunakan sebagai *output* untuk menggerakkan atap gazebo agar dapat terbuka dan tertutup otomatis. *Motor Driver* digunakan untuk mengatur pergerakan dari Motor DC. Rangkaian dari Motor DC dan Driver L298n dapat dilihat pada gambar 4.15 berikut.



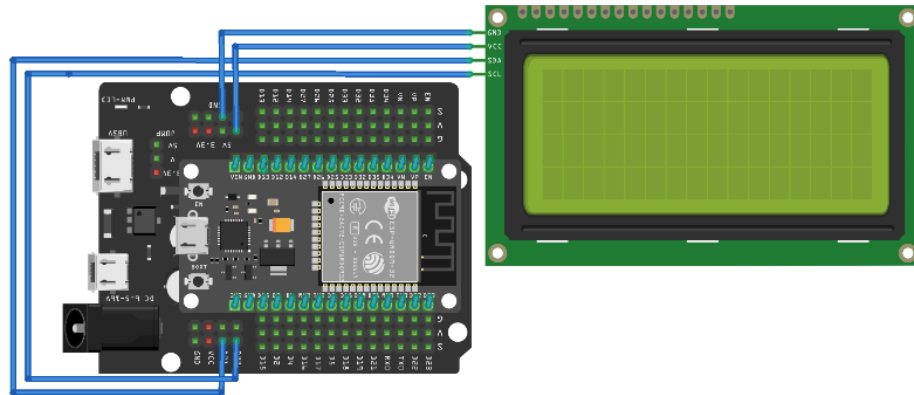
Gambar 4. 15 Rangkaian Motor DC

Penjelasan pin pada Motor DC dan *Motor Driver* l298n yang terhubung pada ESP32 adalah sebagai berikut.

1. Pin pada Motor DC terhubung ke pin OUT1 pada *Motor Driver*.
2. Pin pada Motor DC terhubung ke pin OUT2 pada *Motor Driver*.
3. Pin pada Motor DC terhubung ke pin OUT3 pada *Motor Driver*.
4. Pin pada Motor DC terhubung ke pin OUT4 pada *Motor Driver*.
5. Pin D32 pada ESP32 *Expansion board* terhubung ke pin ENA pada *Motor Driver*.
6. Pin D12 pada ESP32 *Expansion board* terhubung ke pin IN1 pada *Motor Driver*.
7. Pin D33 pada ESP32 *Expansion board* terhubung ke pin IN2 pada *Motor Driver*.
8. Pin D16 pada ESP32 *Expansion board* terhubung ke pin IN3 pada *Motor Driver*.
9. Pin D15 pada ESP32 *Expansion board* terhubung ke pin IN4 pada *Motor Driver*.
10. Pin D4 pada ESP32 *Expansion board* terhubung ke pin ENB pada *Motor Driver*.
11. Pin GND pada ESP32 *Expansion board* terhubung ke pin GND pada *Motor Driver*.

4.4.8 Rangkaian *Liquid Cristal Display* (LCD)

Pada perancangan alat ini LCD digunakan sebagai *output* untuk menampilkan kondisi cuaca dan suhu udara disekitar gazebo serta ucapan selamat datang pada pengunjung yang memasuki gazebo. Rangkaian dari LCD dapat dilihat pada gambar 4.16 berikut.



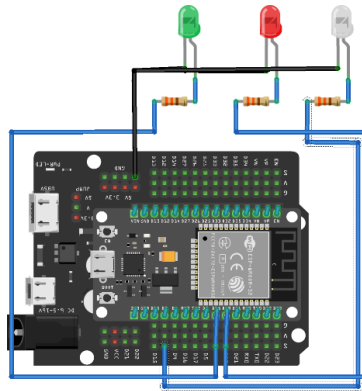
Gambar 4. 16 Rangkaian LCD

Penjelasan pin pada LCD yang terhubung pada ESP32 adalah sebagai berikut.

1. Pin D21 pada ESP32 *Expansion board* terhubung ke pin SDA pada LCD.
2. Pin D22 pada ESP32 *Expansion board* terhubung ke pin SCL pada LCD.
3. Pin 5V pada ESP32 *Expansion board* terhubung ke pin VCC pada LCD.
4. Pin GND pada ESP32 *Expansion board* terhubung ke pin GND pada LCD.

4.4.9 Rangkaian LED

Pada perancangan alat ini LED digunakan sebagai indikator yang berfungsi sebagai penanda saat sistem atap otomatis gazebo sedang berjalan atau sudah selesai, serta sebagai lampu penerangan pada gazebo saat kondisi atap gazebo sedang tertutup karna disekitar gazebo sedang turun hujan atau cuaca gelap. Rangkaian dari LED dapat dilihat pada gambar 4.17 berikut.



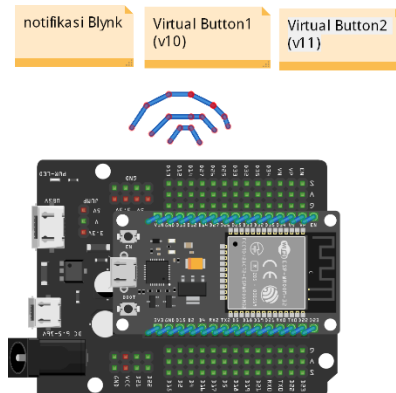
Gambar 4. 17 Rangkaian LED

Penjelasan pin pada LED yang terhubung pada ESP32 adalah sebagai berikut.

1. Pin D5 pada ESP32 *Expansion board* terhubung ke pin positif pada LED.
2. Pin D18 pada ESP32 *Expansion board* terhubung ke pin positif pada LED.
3. Pin D19 pada ESP32 *Expansion board* terhubung ke pin positif pada LED.
4. Pin GND pada ESP32 *Expansion board* terhubung ke pin GND pada LED.

4.4.10 Rangkaian Aplikasi *Blynk*

Pada perancangan alat ini aplikasi *Blynk* digunakan sebagai media pemberitahuan tentang kondisi yang sedang terjadi disekitar gazebo dan pengontrolan atap gazebo secara manual dari jarak jauh dengan bantuan sinyal internet, sehingga pemilik bangunan gazebo mengetahui kondisi cuaca disekitar gazebo dan juga dapat mengontrol atap gazebo dengan mudah. Rangkaian dari aplikasi *Blynk* dapat dilihat pada gambar 4.18 berikut.



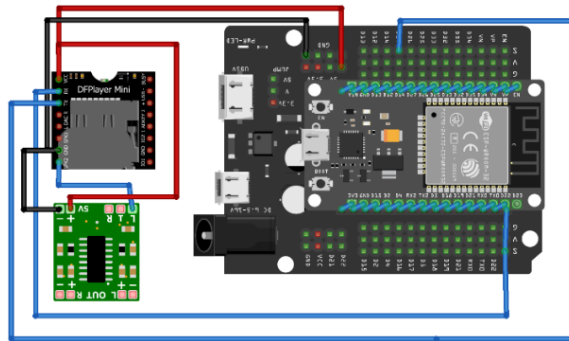
Gambar 4. 18 Rangkaian Aplikasi *Blynk*

Penjelasan pin pada aplikasi *Blynk* yang terhubung pada ESP32 adalah sebagai berikut.

1. Modul WiFi pada ESP32 terhubung ke fitur notifikasi pada aplikasi *Blynk*.
2. Modul WiFi pada ESP32 terhubung ke pin V10 *Virtual Button* Pada aplikasi *Blynk*.
3. Modul WiFi pada ESP32 terhubung ke pin V11 *Virtual Button* Pada aplikasi *Blynk*.

4.4.11 Rangkaian *DFPlayer* mini Mp3

Pada perancangan alat ini *DFPlayer* mini berperan sebagai modul pemutar audio yang memutar file suara dalam bentuk format mp3 sesuai perintah dari ESP32. Sinyal audio dari *DFPlayer* mini diteruskan ke modul amplifier untuk meningkatkan kualitas suara agar dapat terdengar dengan jelas oleh pengunjung. Rangkaian dari *DFPlayer* mini dapat dilihat pada gambar 4.19 berikut.



Gambar 4. 19 Rangkaian *DFPlayer* Mini

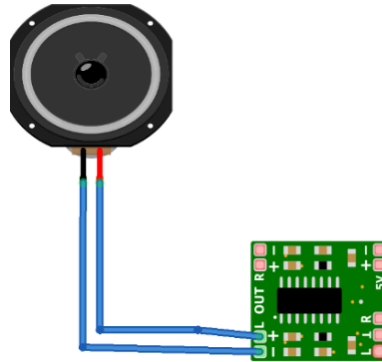
Penjelasan pin pada *Speaker*, Modul Amplifier dan *DFPlayer* mini yang terhubung pada ESP32 adalah sebagai berikut :

1. Pin L pada Modul Amplifier terhubung ke pin SPK2 pada *DFPlayer* mini.
2. Pin 5v pada Modul Amplifier terhubung ke VCC *DFPlayer* mini.
3. Pin GND pada Modul Amplifier terhubung ke GND *DFPlayer* mini.
4. Pin D27 pada ESP32 *Expansion board* terhubung ke pin TX *DFPlayer* mini.
5. Pin D23 pada ESP32 *Expansion board* terhubung ke pin RX *DFPlayer* mini.
6. Pin 5V pada ESP32 *Expansion board* terhubung ke pin VCC pada *DFPlayer* mini.
7. Pin GND pada ESP32 *Expansion board* terhubung ke pin GND pada *DFPlayer* mini.

4.4.12 Rangkaian *Speaker* dan Modul Amplifier

Pada perancangan alat ini *Speaker* digunakan sebagai perangkat *output* audio untuk mengeluarkan suara berupa ucapan selamat datang kepada pengunjung yang memasuki gazebo. Suara yang telah diperkuat oleh modul amplifier dikeluarkan

melalui *speaker*. Rangkaian dari *Speaker* dan Modul amplifier dapat dilihat pada gambar 4.20 berikut.



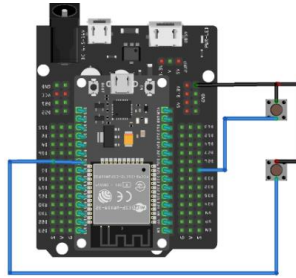
Gambar 4. 20 Rangkaian *Speaker* dan Modul Amplifier

Penjelasan pin pada *Speaker*, Modul Amplifier dan *DFPlayer* mini yang terhubung pada ESP32 adalah sebagai berikut :

1. Pin positif pada *Speaker* terhubung ke pin out L positif pada modul Amplifier.
2. Pin negatif pada *Speaker* terhubung ke pin out L negatif pada modul Amplifier.

4.4.13 Rangkaian *Push Button*

Pada perancangan alat ini *Push Button* digunakan sebagai media input yang memberikan sinyal logika *low* untuk mengendalikan pembukaan atau penutupan atap gazebo, serta pengalihan antara mode manual dan otomatis. Rangkaian dari *Push Button* dapat dilihat pada gambar 4.21 berikut.



Gambar 4. 21 Rangkaian *Push Button*

Penjelasan pin pada *Push Button* yang terhubung pada ESP32 adalah sebagai berikut:

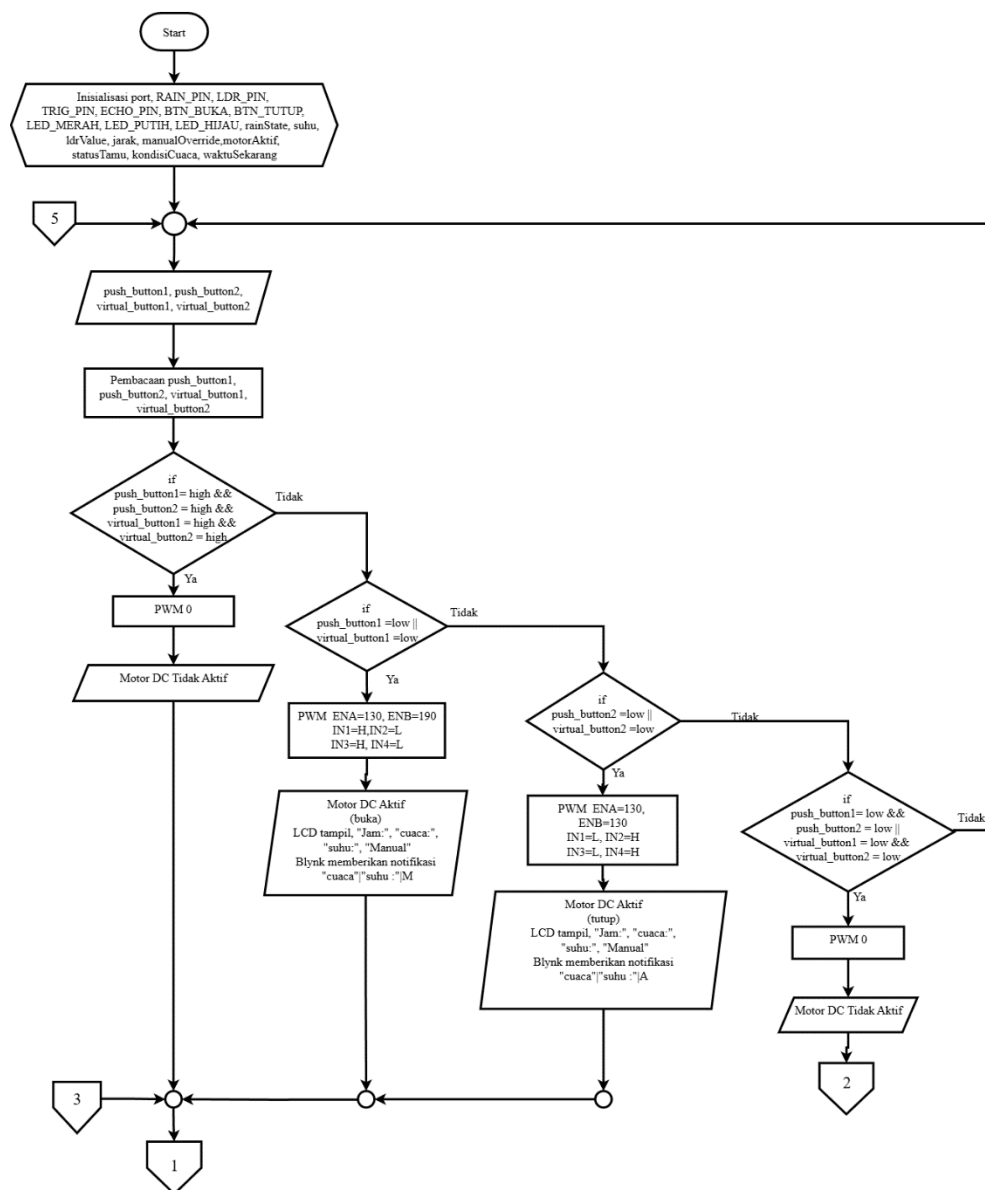
1. Pin D25 pada ESP32 *Expansion board* terhubung ke pin positif *Push Button*.
2. Pin D17 pada ESP32 *Expansion board* terhubung ke pin positif *Push Button*.
3. Pin GND pada ESP32 *Expansion board* terhubung ke pin GND pada *Push Button*.

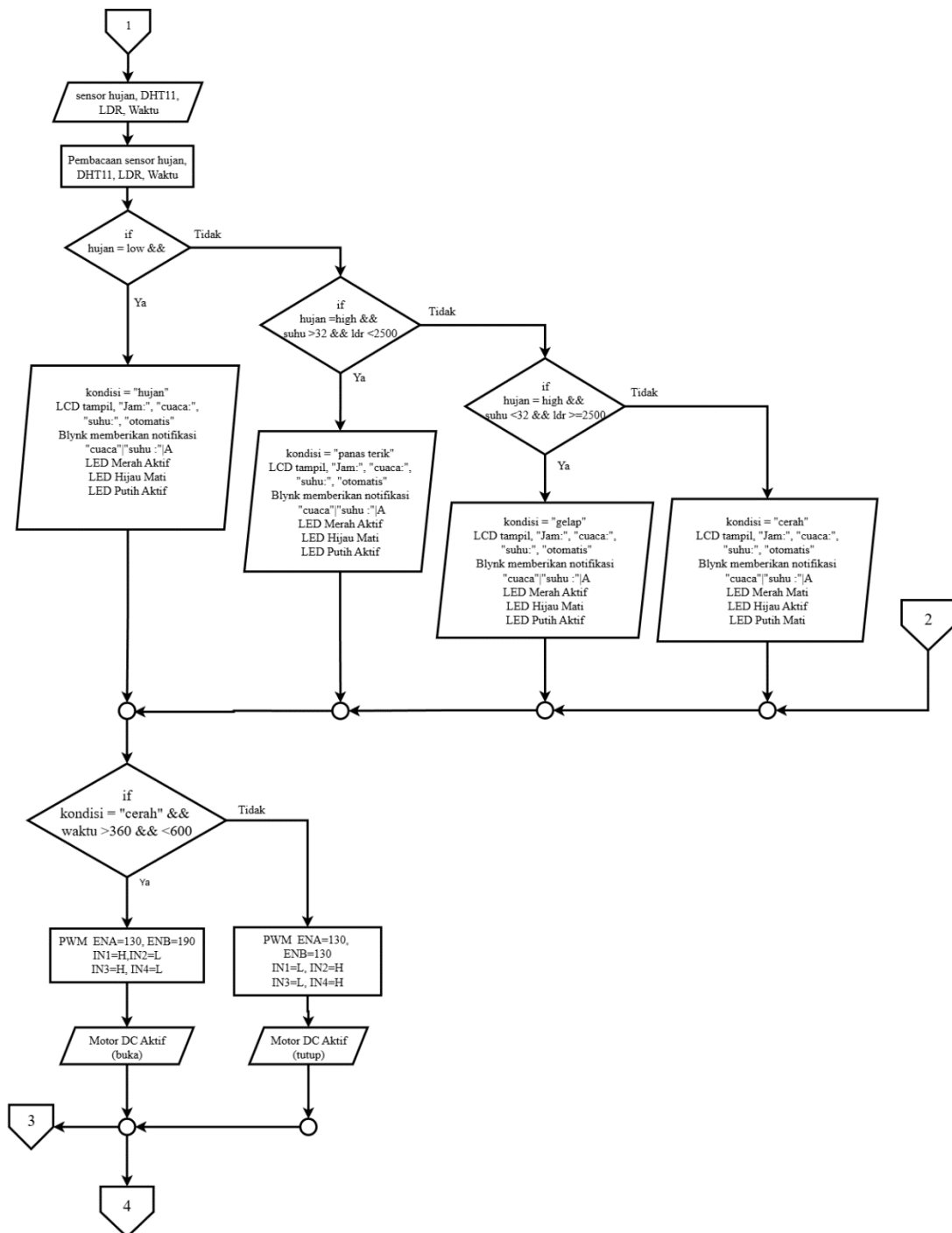
4.5 Rancangan Modul Program

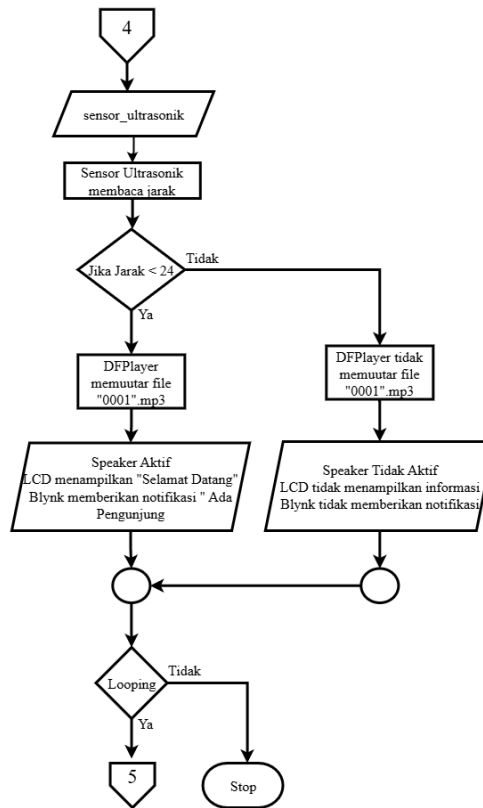
Sub bab ini menjelaskan tentang Modul Program yang digunakan sebagai pengontroll sistem yang dirancang. Untuk lebih mudah dimengerti rancangan dapat menjadi dua bagian yaitu *Flowchart*, dan *Listing Program*. *Flowchart* memberikan gambaran visual tentang alur logika dan proses yang terjadi dalam sistem sehingga memudahkan dalam pengembangan alat. *Listing program* menyajikan kode sumber yang konkret, yang merupakan implementasi dari logika yang telah digambarkan dalam *Flowchart*.

4.5.1 Flowchart

Modul program dirancang memiliki struktur dengan kualitas yang baik dan mudah dimengerti, maka sebelum pembuatan *Listing* program perlu diawali dengan penentuan logika program. Logika dasar gambaran pada penulisan ini adalah dengan menggunakan *Flowchart* seperti pada gambar 4.22.







Gambar 4. 22 Flowchart

4.5.2 Listing Program

Berikut dibawah ini merupakan *listing* program yang terdiri dari pemograman keseluruhan pada ESP32.

1. Library Sensor

```

#define BLYNK_TEMPLATE_ID "TMPL6_LEXfXyn"
#define BLYNK_TEMPLATE_NAME "GAZEBONOTIF"
#define BLYNK_AUTH_TOKEN "xb1hNeIr3d-fpkVAkfrM0RQWm5N3W9gY"
#define BLYNK_PRINT Serial
#include <WiFi.h>
#include <WiFiClient.h>
#include <BlynkSimpleEsp32.h>
#include <DHT.h>
#include <Wire.h>

```

```
#include <LiquidCrystal_I2C.h>
#include <DFRobotDFPlayerMini.h>
#include "time.h"
```

2. Koneksi wifi

```
char ssid[] = "hani";
char pass[] = "88888888";
```

3. Pin Komponen

```
#define RAIN_PIN 14
#define LDR_PIN 34
#define DHT_PIN 26
#define DHTTYPE DHT11
#define TRIG_PIN 2
#define ECHO_PIN 35
#define BTN_BUKA 25
#define BTN_TUTUP 17
#define DF_RX 27
#define DF_TX 23
#define LED_MERAH 18
#define LED_PUTIH 19
#define LED_HIJAU 5
#define ENA 32
#define IN1 12
#define IN2 33
#define ENB 4
#define IN3 16
#define IN4 15
```

4. Konfigurasi Waktu

```
const char* ntpServer = "pool.ntp.org";
const long gmtOffset_sec = 25200; // GMT+7 (WIB) = 7 * 3600
const int daylightOffset_sec = 0;
```

5. Settingan Komponen dan Sensor

```
DHT dht(DHT_PIN, DHTTYPE);
BlynkTimer timer;
HardwareSerial dfSerial(2);
DFRobotDFPlayerMini dfplayer;
```

```

#define LDR_THRESHOLD 2500
#define BATAS_TAMU 24
#define LCD_NOTIF_DURATION 5000
bool statusCuacaBuruk = false;
bool statusTamu = false;
bool lastCuacaBuruk = false;
bool motorAktif = false;
bool audioSudahDiputar = false;
bool manualOverride = false;
bool btnBukaState = false;
bool btnTutupState = false;
bool btnBukaFisik = false;
bool btnTutupFisik = false;
bool lastBtnBukaFisik = false;
bool lastBtnTutupFisik = false;
bool lastBtnBukaBlynk = false;
bool lastBtnTutupBlynk = false;
String kondisiCuacaTerakhir = "";
String kondisiSebelumnya = "TERTUTUP";
unsigned long lcdNotifTime = 0;
unsigned long motorStartTime = 0;
bool lcdSedangNotif = false;

```

6. Program Sensor Ultrasonik

```

long bacaUltrasonik() {
    digitalWrite(TRIG_PIN, LOW);
    delayMicroseconds(2);
    digitalWrite(TRIG_PIN, HIGH);
    delayMicroseconds(10);
    digitalWrite(TRIG_PIN, LOW);
    long durasi = pulseIn(ECHO_PIN, HIGH, 18000);
    if (durasi == 0) return 0;
    return durasi * 0.034 / 2;
}

```

7. Program Motor Driver

```

void motorMajuLow() {
    digitalWrite(IN1, HIGH); digitalWrite(IN2, LOW);
    digitalWrite(IN3, HIGH); digitalWrite(IN4, LOW);
    analogWrite(ENA, 130);
    analogWrite(ENB, 190);
}

```

```

}
void motorMundurLow() {
    digitalWrite(IN1, LOW); digitalWrite(IN2, HIGH);
    digitalWrite(IN3, LOW); digitalWrite(IN4, HIGH);
    analogWrite(ENA, MOTOR_PWM_LOW);
    analogWrite(ENB, MOTOR_PWM_LOW);
}
void motorStop() {
    analogWrite(ENA, 0);
    analogWrite(ENB, 0);
}

int waktuSekarang = getWaktuDalamMenit();
bool waktuBuka = (waktuSekarang >= 360 && waktuSekarang < 600);
bool atapHarusBuka = (kondisiLCD == "CERAH" && waktuBuka);

if (!manualOverride && !motorAktif) {

    if (atapHarusBuka != (kondisiSebelumnya == "TERBUKA")) {

        motorStartTime = millis();
        motorAktif = true;

        if (atapHarusBuka) {
            motorMajuLow();    // buka atap
            kondisiSebelumnya = "TERBUKA";
        } else {
            motorMundurLow();  // tutup atap
            kondisiSebelumnya = "TERTUTUP";
        }
    }
}
}

```

8. Program LCD

```

void tampilkanNotifLCD(String pesan) {
    lcd.clear();
    lcd.setCursor(0, 1);
    lcd.print(pesan);
    lcdNotifTime = millis();
    lcdSedangNotif = true;
}

```

```

void tampilkanStatusNormal(float suhu, bool cuaca) {
    lcd.clear(); // bersihkan total dulu
    lcd.setCursor(0, 0);
    lcd.print("JAM:");
    lcd.print(getJamNTP());

    lcd.setCursor(0, 1);
    lcd.print("CUACA:");
    lcd.print(cuaca ? "BURUK" : "CERAH");

    lcd.setCursor(0, 2);
    lcd.print("SUHU:");
    if (isnan(suhu)) lcd.print("0.0");
    else lcd.print(suhu, 1);

    lcd.setCursor(0, 3);
    lcd.print(manualOverride ? "MANUAL" : "OTOMATIS");
}

```

9. Program Sensor Hujan, Suhu, DHT

```

void cekSemuaSensor() {
    int rainState = digitalRead(RAIN_PIN);
    int ldrValue = analogRead(LDR_PIN);
    float suhu = dht.readTemperature();
    long jarak = bacaUltrasonik();
    if (!isnan(suhu)) Blynk.virtualWrite(V1, suhu);
    int jamSekarang = getJamSekarang();
    String kondisiLCD;
    if (rainState == LOW) {
        kondisiLCD = "HUJAN";
    }
    else if (rainState == HIGH && suhu < 32 && ldrValue >= 2500) {
        kondisiLCD = "GELAP";
    }
    else if (rainState == HIGH && suhu > 32 && ldrValue < 2500) {
        kondisiLCD = "PANAS TERIK";
    }
    else {
        kondisiLCD = "CERAH";
    }
}

```

10. Program Notifikasi Blynk

```

if (kondisiLCD != kondisiCuacaTerakhir) {
  String mode = manualOverride ? "M" : "A";
  if (kondisiLCD == "HUJAN") {
    Blynk.logEvent("hujan", "Cuaca hujan | Suhu: " + String(suhu,1) + " | " + mode);
  }
  else if (kondisiLCD == "GELAP") {
    Blynk.logEvent("gelap", "Cuaca gelap | Suhu: " + String(suhu,1) + " | " + mode);
  }
  else if (kondisiLCD == "PANAS TERIK") {
    Blynk.logEvent("panas", "Cuaca panas | Suhu: " + String(suhu,1) + " | " + mode);
  }
  else if (kondisiLCD == "CERAH") {
    Blynk.logEvent("cerah", "Cuaca cerah | Suhu: " + String(suhu,1) + " | " + mode);
  }
  kondisiCuacaTerakhir = kondisiLCD;
}

```

11. Program DFPLAYER

```

if (jarak > 0 && jarak < BATAS_TAMU) {
  if (!statusTamu) { // hanya saat pertama kali terdeteksi
    statusTamu = true;
    dfplayer.play(1);
    Blynk.logEvent("tamu", "Ada pengunjung");
    tampilkanNotifLCD("SELAMAT DATANG");
  }
}
else {
  statusTamu = false;
}

```

12. Program LED

```

if (kondisiLCD == "HUJAN" ||
    kondisiLCD == "PANAS TERIK" ||
    kondisiLCD == "GELAP") {

  digitalWrite(LED_MERAH, HIGH);
  digitalWrite(LED_PUTIH, HIGH);
}

```

```

    digitalWrite(LED_HIJAU, LOW);
}
else {

    digitalWrite(LED_MERAH, LOW);
    digitalWrite(LED_PUTIH, LOW);
    digitalWrite(LED_HIJAU, HIGH);
}

```

13. Program BUTTON

```

void cekTombolFisik() {
    bool buka = (digitalRead(BTN_BUKA) == LOW);
    bool tutup = (digitalRead(BTN_TUTUP) == LOW);
    bool comboNow = buka && tutup;
    bool comboLast = lastBtnBukaFisik && lastBtnTutupFisik;
    if (comboNow && !comboLast) {
        manualOverride = false;
        Serial.println("MODE AUTO (FISIK)");
    }
    else if (buka && !lastBtnBukaFisik && !motorAktif && !comboNow) {
        manualOverride = true;
        motorAktif = true;
        motorStartTime = millis();
        motorMajuLow();
        Serial.println("MODE MANUAL - BUKA (FISIK)");
    }
    else if (tutup && !lastBtnTutupFisik && !motorAktif && !comboNow) {
        manualOverride = true;
        motorAktif = true;
        motorStartTime = millis();
        motorMundurLow();
        Serial.println("MODE MANUAL - TUTUP (FISIK)");
    }
    lastBtnBukaFisik = buka;
    lastBtnTutupFisik = tutup;
}

BLYNK_WRITE(V10) {
    bool buka = param.asInt();
    bool comboNow = buka && btnTutupState;
    bool comboLast = lastBtnBukaBlynk && lastBtnTutupBlynk;

```

```

if (comboNow && !comboLast) {
    manualOverride = false;
}
else if (buka && !lastBtnBukaBlynk && !motorAktif && !comboNow) {
    manualOverride = true;
    motorAktif = true;
    motorStartTime = millis();
    motorMajuLow();
}
lastBtnBukaBlynk = buka;
btnBukaState = buka;
}

BLYNK_WRITE(V11) {
    bool tutup = param.asInt();
    bool comboNow = btnBukaState && tutup;
    bool comboLast = lastBtnBukaBlynk && lastBtnTutupBlynk;
    if (comboNow && !comboLast) {
        manualOverride = false;
    }
    else if (tutup && !lastBtnTutupBlynk && !motorAktif && !comboNow) {
        manualOverride = true;
        motorAktif = true;
        motorStartTime = millis();
        motorMundurLow();
    }
    lastBtnTutupBlynk = tutup;
    btnTutupState = tutup;
}

```