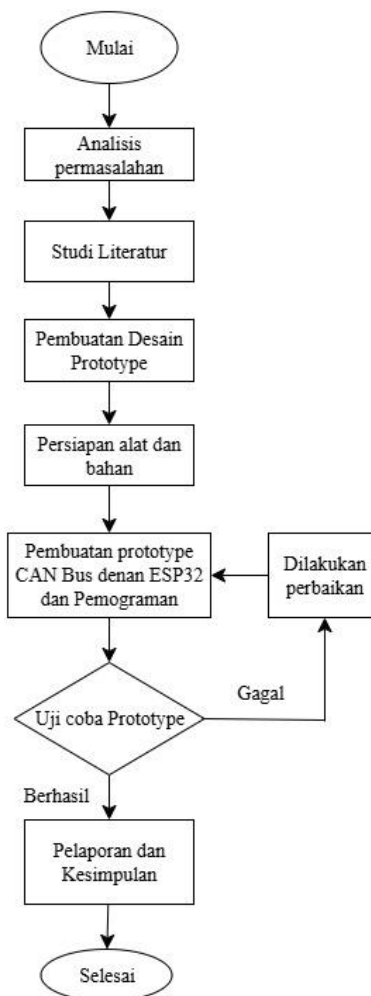


BAB III

METODELOGI PENELITIAN

3.1 Tahap Metode Penelitian

Dalam proses penelitian digunakan metode eksperimen dengan *Research and Development* (R&D). Penelitian ini bertujuan untuk membuat jaringan CAN Bus untuk *monitoring* logistik pengiriman vaksin. Proses perancangan sistem ini dapat digunakan untuk *monitoring* vaksin di dalam *box* pengiriman bagi staf pengiriman atau pihak terkait. Pada Gambar 3.1 dijelaskan alur diagram tahapan metode penelitian.



Gambar 3.1 Diagram alir perancangan jaringan CAN Bus

Tahapan awal penelitian ini, ialah analisis permasalahan yang dilakukan dengan cara mempelajari terkait permasalahan-permasalahan yang ada berkaitan

dengan *monitoring* logistik pengiriman vaksin. Maka ditemukan sebuah ide yang dapat menjadi solusi dari permasalahan tersebut melalui sistem yang akan dibuat. Setelah dilakukan analisis permasalahan, selanjutnya melakukan studi literatur. Pada prosesnya melakukan cara pengamatan dan pemahaman disertai pengumpulan informasi. Sumber informasi yang dicari berasal dari berbagai sumber seperti jurnal ilmiah dan artikel *website*. Informasi yang dibutuhkan ialah teori tentang ESP32, ESP32-CAM, CAN Bus, IoT, dan pemrograman tentang Arduino IDE.

Tahap pembuatan desain prototipe, melakukan skema perancangan model alat yang dibuat dengan sederhana dan praktis. Proses perancangan ini pun mulai dipikirkan bagaimana cara kerja alat tersebut. Sistem desain perencanaan dibuat meliputi *input*, proses, dan *output*. Kemudian, melakukan tahapan persiapan peralatan dan bahan yang akan digunakan pada saat proses perancangan prototipe CAN Bus *monitoring* logistik pengiriman vaksin dengan mementingkan kualitas yang baik dengan harga yang terjangkau. Penelitian ini dalam prosesnya menggunakan alat dan bahan dalam bentuk perangkat keras (*Hardware*) dan perangkat lunak (*Software*).

Pada tahap pembuatan prototipe jaringan CAN Bus *monitoring* terdapat komponen *hardware* yang digunakan yaitu modul CAN TJA1050, ESP32 yang berfungsi sebagai modul *server* untuk mengirim dan menerima pesan dari sensor-sensor yang akan dihubungkan. Dalam rangkaian ini terdapat ESP32-CAM sebagai modul *monitoring* secara visual yang memungkinkan untuk dapat melakukan pemantauan secara *real-time*. GPS Neo-6M Ublox sebagai modul untuk menentukan titik kordinat lokasi *real-time* pengiriman. Penghubungan dengan sensor suhu dan kelembapan untuk dapat melihat kondisi *box* vaksin, mengingat vaksin memiliki rentang suhu sensitif harus tetap pada 2°C sampai 8°C. Untuk proses pemrograman dilakukan menggunakan *software* Arduino IDE.

Tahapan uji coba prototipe ialah sebuah proses pengujian alat yang bertujuan untuk dapat memastikan apakah prototipe tersebut dapat bekerja dengan optimal. Apabila terdapat keadaan prototipe yang telah dibuat tidak dapat bekerja dengan baik, maka dilakukan langkah perbaikan dan pengujian ulang pada prototipe untuk memperbaiki masalah yang terjadi.

Terakhir, tahap pelaporan ialah berisikan data-data yang telah didapatkan dari pengujian prototipe jaringan CAN Bus *monitoring* logistik pengiriman vaksin. Serta kesimpulan yang terdapat rangkuman dari hasil penelitian yang berdasarkan pada rumusan masalah dan tujuan penelitian yang telah dirumuskan sebelumnya.

Pada penelitian ini, proses pengujian dilakukan dengan tujuan untuk menilai efektivitas dan mengidentifikasi sejauh mana keberhasilan dari prototipe yang telah dibuat. Untuk parameter keberhasilan di dasarkan pada pengujian fungsionalitas dan uji kinerja.

a. Pengujian Fungsionalitas

Pengujian fungsionalitas tersebut dilakukan dengan bertujuan untuk dapat mengetahui jika sistem yang telah dibuat dapat beroperasi dengan baik. Aspek-aspek yang akan diuji dalam pengujian fungsionalitas antar lain :

- Pengaruh penggunaan jaringan CAN Bus pada *monitoring* logistik pengiriman vaksin.
- Kemampuan stabilitasi dan cepat jaringan CAN dalam proses mengirim menerima pesan pada *monitoring* logistik pengiriman vaksin.

b. Pengujian Kinerja

Pengujian kinerja berfungsi untuk dapat mengukur kinerja pada sistem dalam berbagai keadaan. Aspek yang akan diuji ialah kemampuan respon pada CAN Bus yang di pasang beberapa sensor untuk *monitoring* untuk mengetahui vaksin dalam *box*.

3.2 Perancangan Sistem

Perancangan sistem jaringan CAN Bus untuk *monitoring* logistik pengiriman vaksin terdiri dari beberapa komponen yang dapat menghasilkan *output* berupa suhu kelembapan, *monitoring* visual, dan titik kordinat lokasi yang dapat dilihat melalui Blynk IoT. Tahapan perancangannya sebagai berikut.

3.2.1 Perangkat Penunjang Penelitian

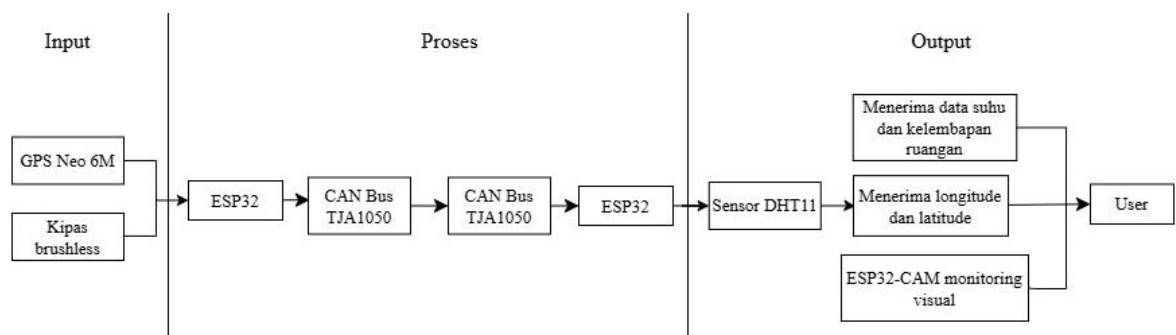
Pada proses penelitian digunakan dua jenis perangkat penunjang, yaitu perangkat keras (*hardware*) dan perangkat lunak (*software*). Untuk bagian *hardware* alat yang akan digunakan diantaranya ESP32 yang berfungsi sebagai mikrokontroler pada perangkat ini kemudian modul CAN TJA1050, Sensor

DHT11, ESP32-CAM, GPS Neo-6M Ublox dan kipas *brushless*. Kemudian perangkat lunak yang digunakan pada penelitian ini ialah Arduino IDE.

Penelitian kali ini terdapat batasan pengujian yang perlu diperhatikan diantaranya alat yang digunakan pada rangkaian jaringan CAN Bus ialah ESP32 dan ESP32-CAM, dimana modul tersebut dapat bekerja untuk IoT dengan jaringan internet yang stabil. Pada saat terdapat kendala pada internet yang banyak digunakan atau tidak stabil dan jarak yang jauh dari sumber WiFi maka akan membuat kualitas penerimaan data dan visual ESP32 serta ESP32-CAM dapat terpengaruh. Hal tersebut menyebabkan tantangan dalam proses pengiriman dan penerimaan data yang dilakukan.

3.2.2 Prinsip Kerja

Prinsip kerja yang diterapkan pada penelitian ini dilihat pada diagram blok yang ditunjukkan pada Gambar 3.2 dibawah ini.



Gambar 3.2 Diagram blok prinsip kerja alat

Pada sistem *monitoring* logistik pengiriman vaksin ini terdiri dari 3 tahapan yaitu *input*, *proses*, dan *output*. Pada tahap *input* terdapat Sensor DHT11 yang memiliki fungsi untuk sebagai modul pendeteksi keadaan ruangan seperti suhu dan kelembapan. Kipas *brushless* memiliki fungsi sebagai pendingin pada *box* vaksin. GPS Neo-6M Ublox berfungsi sebagai menampilkan titik kordinat untuk lokasi *real-time* vaksin saat pengiriman.

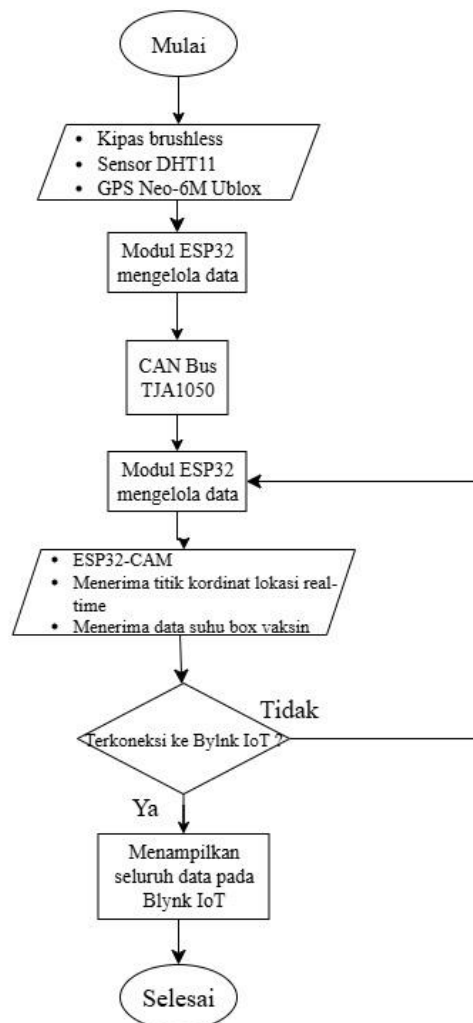
Kemudian, tahap *proses* terdapat ESP32 berfungsi sebagai mikrokontroler yang melakukan proses pengiriman pesan dari data yang didapatkan dari sensor-sensor tersebut. CAN Bus TJA1050 berfungsi sebagai perantara pengiriman pesan untuk ke mikrokontroler lainnya yang cukup hanya menghubungkan CAN_H dan CAN_L agar dapat terjadinya komunikasi antar mikrokontroler. Adanya CAN Bus

tersebut tidak hanya dapat mengirimkan pesan, akan tetapi bisa menerima pesan satu sama lain, yang diberikan ID untuk memprioritaskan pesan mana yang perlu dikirimkan terlebih dahulu.

Tahap terakhir ialah *output*, merupakan data yang telah dikirimkan oleh modul CAN Bus TJA1050 dan ESP32 yang kemudian data dapat dilihat pada tampilan Blynk IoT yaitu berupa suhu, kelembapan, titik kordinat dan *monitoring* visual melalui ESP32-CAM.

3.2.3 Alur Kerja Perangkat Keras (*Hardware*) untuk Prototipe.

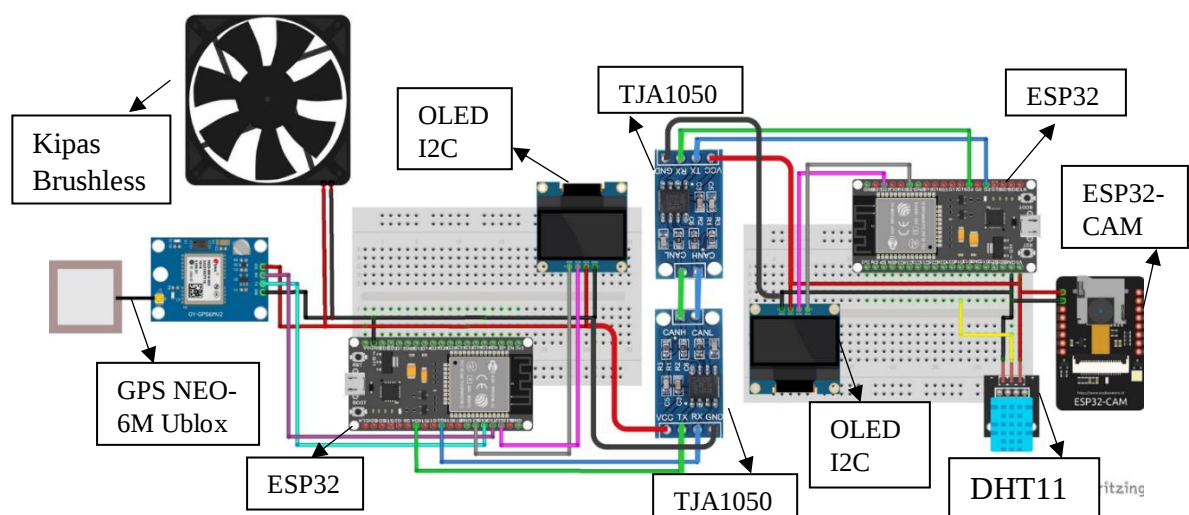
Perancangan sistem jaringan CAN Bus pada *monitoring* logistik pengiriman vaksin terdapat beberapa tahapan untuk mengerjakan rangkaian yang diharapkan. Adapun tahapan yang harus dilakukan terdapat pada Gambar 3.3 dibawah ini.



Gambar 3.3 Diagram alir sistem jaringan CAN Bus

Pada *flowchart* Gambar 3.3 menunjukkan proses kerja pada jaringan CAN Bus untuk *monitoring* logistik pengiriman vaksin. Sistem tersebut memerlukan *power supply* untuk bekerja berupa listrik DC 5 Volt. Kemudian dihubungkan dengan modul ESP32 sebagai mikrokontroler jalannya sistem tersebut. Penggunaan mikrokontroler dua buah pada ESP32, dimana pada ESP32 pertama digunakan agar dapat terhubung dengan GPS Neo-6M untuk mendeteksi titik kordinat lokasi dan kipas *brushless* untuk pendingin lingkungan sekitar *box* pada vaksin. Kedua, ESP32 digunakan untuk sensor DHT11 mendeteksi keadaan suhu dan kelembapan pada *box*. Kemudian di kirimkan data tersebut pada tampilan aplikasi Blynk IoT untuk dapat melakukan pemantauan keadaan vaksin. Penghubungan antar ESP32 digunakan protokol CAN Bus dengan modul TJA1050 untuk dapat berkomunikasi mengirimkan data yang didapat dari beberapa sensor tersebut. Pada tampilan di Blynk IoT pengguna dapat melakukan *monitoring* berupa suhu ruangan, lokasi vaksin, dan visual secara *real-time*. ESP32-CAM menampilkan visual *real-time* yang dapat diatur langsung untuk kualitas yang dihasilkan, karena hanya digunakan 2MP pada kamera OV2640 akan lebih baik untuk tidak lebih dari 1600x1200 untuk visual yang ditampilkan.

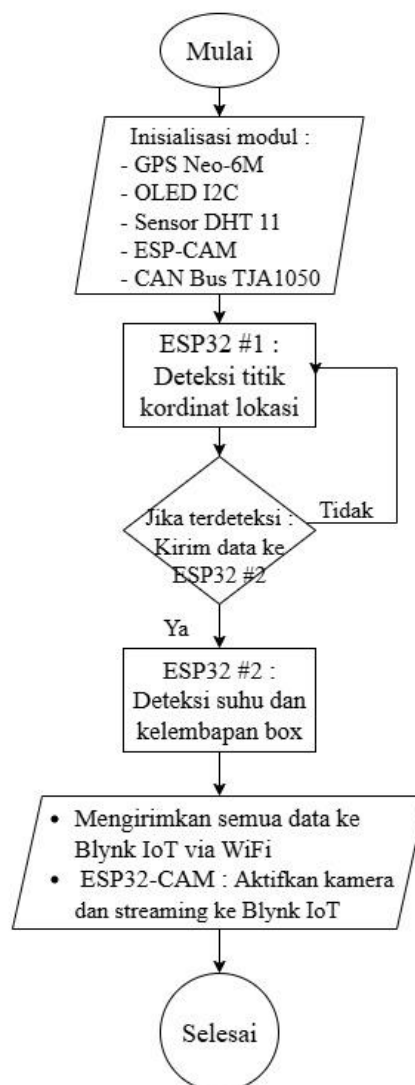
Rangkaian komponen pada Gambar 3.4 dalam sistem jaringan CAN Bus *monitoring* logistik pengiriman vaksin dapat dilihat di bawah ini. Pada sistem ini digunakan mikrokontroler, modul TJA1050, ESP32-CAM, GPS Neo-6M, Kipas *brushless*, Oled dan modul DHT11.



Gambar 3.4 Skema Rangkaian Sistem Monitoring

3.2.4 Alur Kerja Perangkat Lunak (*Software*) untuk Prototipe.

Proses pembuatan prototipe jaringan CAN Bus untuk *Monitoring* logistik pengiriman vaksin ini terdapat tahapan yang dilakukan pada perangkat lunak (*software*) ialah Arduino IDE yang berfungsi menjadi tempat untuk dapat memasukkan kode pemrograman kedalam ESP32 dan ESP32-CAM. Rangkaian pada *hardware* tersebut dapat berjalan dengan baik ketika telah di masukkan kode pemrograman tersebut. Untuk alur kerja perangkat lunak (*software*) tersebut dapat dilihat pada Gambar 3.5.

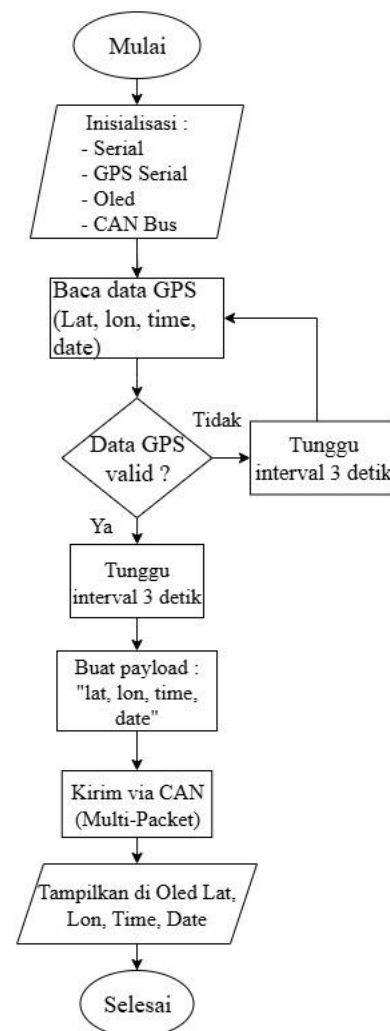


Gambar 3.5 Alur Kerja Software

Alur kerja *software* diatas ialah beberapa tahap yang dilakukan pada *software* Arduino IDE, dimana diantaranya ialah melakukan proses inisialisasi modul yang

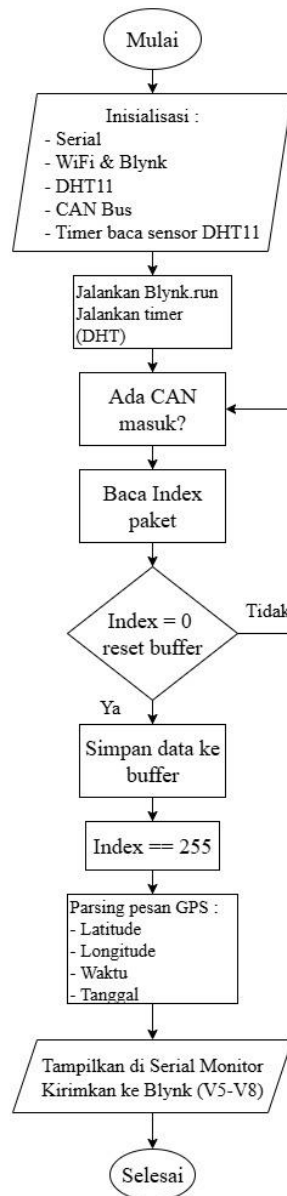
digunakan dengan penambahan *library* sebagai pendukung jalannya program. *Library* yang digunakan diantaranya CAN, WiFi, *esp_camera*, *WebServer*, DHT, dan *BlynkSimpleEsp32*. Setelah itu, kode pemrograman yang dituliskan pada *software* Arduino IDE bertujuan agar perangkat *hardware* dapat berjalan sesuai dengan instruksi yang diberikan melalui kode pemrograman. Kemudian, mikrokontroler #1 digunakan untuk Kipas *Brushless* membuat ruang pada *box* vaksin dingin dan GPS Neo-6M untuk mendeteksi titik kordinat lokasi vaksin menampilkan pada OLED I2C. Selanjutnya, jika tidak terdeteksi maka kembali kepada pemrograman #1 untuk deteksi permasalahan. Jika terdeteksi, maka proses akan melanjutkan ke proses pengiriman data ke ESP32 #2 menggunakan modul CAN Bus TJA1050 pada saat proses pengiriman paket pesan tersebut. Pada mikrokontroler ESP32 #2 digunakan untuk sensor DHT 11 agar mendeteksi suhu dan kelembapan pada ruangan ditampilkan pada OLED I2C. Seluruh data yang telah dihasilkan dikirimkan ke Blynk IoT dari ESP32 #2 melalui WiFi. Modul ESP32-CAM diaktifkan kameranya dan *streaming* agar dapat terhubung dan di *monitoring* ke Blynk IoT. Jika proses kerja sistem sudah sampai tahap ini atau akhir, maka dapat dikatakan sistem dapat berkerja dengan baik dan sesuai perintah yang telah dibuat pada kode pemrograman.

Untuk alur kerja berdasarkan koding terbagi menjadi 2, yaitu *Transmitter* dan *Receiver*. Pada Gambar 3.6 *Flowchart* untuk koding pengirim, menunjukkan alur kerja sistem komunikasi berbasis CAN Bus yang digunakan untuk membaca dan menampilkan data GPS. Proses dimulai dengan melakukan inisialisasi modul yang meliputi Serial, GPS Serial, OLED, dan CAN Bus. Tahap inisialisasi ini bertujuan untuk memastikan seluruh perangkat keras dapat berfungsi dan siap melakukan pertukaran data. Selanjutnya, sistem membaca data GPS berupa *latitude*, *longitude*, *time*, dan *date*. Data hasil pembacaan tersebut kemudian divalidasi untuk memastikan kebenaran dan kelengkapannya. Apabila data GPS yang diterima tidak valid, sistem akan menunggu selama interval 3 detik sebelum melakukan pembacaan ulang.



Gambar 3.6 Flowchart Program Bagian Transmitter

Proses ini dilakukan secara berulang hingga data GPS yang diterima valid. Apabila data GPS sudah dinyatakan valid, sistem tetap memberikan jeda waktu selama 3 detik untuk menjaga kestabilan pembacaan. Setelah itu, data yang diperoleh akan dibentuk menjadi sebuah *payload* yang berisi informasi *latitude*, *longitude*, *time*, dan *date*. *Payload* yang telah disusun kemudian dikirimkan melalui protokol CAN Bus dengan metode *multi-packet transmission*. Pada tahap akhir, data yang telah berhasil dikirim melalui CAN Bus ditampilkan pada OLED display. Informasi yang ditampilkan meliputi *latitude*, *longitude*, *time*, dan *date*, sehingga pengguna dapat memantau posisi dan waktu secara langsung. Dengan demikian, sistem ini mampu melakukan akuisisi, pengolahan, serta pengiriman data GPS secara andal dan *real-time* melalui media komunikasi CAN Bus.



Gambar 3.7 Flowchart Program Bagian Receiver

Pada bagian *Receiver* di Gambar 3.7, menggambarkan alur kerja sistem penerimaan data melalui CAN Bus dan pengiriman hasil *monitoring* ke *platform* Blynk. Proses dimulai dengan tahap inisialisasi perangkat, meliputi Serial, WiFi & Blynk, sensor DHT11, CAN Bus, serta *timer* pembacaan sensor DHT11. Inisialisasi dilakukan agar seluruh perangkat siap digunakan dalam proses komunikasi data dan *monitoring*. Setelah inisialisasi, sistem menjalankan perintah *Blynk.run* dan mengaktifkan *timer* untuk membaca sensor DHT11. Selanjutnya, sistem memeriksa apakah terdapat data masuk melalui CAN Bus.

Jika tidak ada data diterima, maka sistem akan kembali melakukan pengecekan secara berulang. Apabila terdapat data masuk, sistem akan membaca index paket yang diterima. Jika nilai index = 0, maka *buffer* akan di-reset untuk menyiapkan ruang penyimpanan data baru. Data hasil pembacaan kemudian disimpan ke dalam *buffer* secara berurutan hingga mencapai index = 255, yang menandakan bahwa data telah lengkap. Setelah data lengkap, sistem melakukan parsing pesan GPS untuk memisahkan informasi yang terdiri dari *latitude*, *longitude*, waktu, dan tanggal. Tahap terakhir ialah *output*, di mana data yang telah diproses akan ditampilkan pada *Serial Monitor* dan dikirimkan ke aplikasi Blynk (V5–V8) melalui koneksi WiFi. Dengan demikian, pengguna dapat memantau hasil *monitoring* posisi dan kondisi lingkungan secara *real-time* melalui Blynk.

3.3 Teknik Pengambilan Data

Dalam proses teknik pengambilan data ini akan terbagi menjadi 2 bagian yaitu teknik pengambilan data pada uji fungsionalitas dan pengambilan data pada uji kinerja alat tersebut.

3.3.1 Uji Fungsionalitas

Dalam pengujian fungsionalitas terdapat beberapa tahapan untuk di lakukan agar memperoleh data yang diinginkan.

a. Pengamatan Langsung

Langkah dalam pengujian jaringan CAN Bus *monitoring* ialah melakukan sebuah pengamatan secara langsung terkait berjalan nya alat tersebut. Keadaan dan penempatan alat harus dapat mendeteksi situasi yang dapat terjadi pada vaksin. Dalam pengujian ini dilakukan beberapa pengujian, diantaranya :

- Pemeriksaan untuk tiap sensor (DHT11, GPS Neo-6M, OLED I2C) apakah mendeteksi dengan benar.
- Pengamatan perangkat pada kamera, CAN Bus, dan pengiriman data ke Blynk apakah dapat berkerja sesuai dengan arahan.

b. Pencatatan Status Sistem

Pengujian pencatatan status sistem ini dilakukan untuk mengetahui apakah sistem atau alat bekerja sesuai dengan benar. Dimana, status yang dihasilkan pada alat tersebut diantaranya :

- GPS Neo-6M mendeteksi titik koordinat lokasi – Pesan dikirim.
- DHT11 mendeteksi suhu dan kelembapan *box* – Pesan dikirim.
- OLED I2C menerima data – Tampilan muncul tanpa internet.
- Kamera aktif saat kondisi tertentu – Tampilan muncul di Blynk.

c. Pengujian Skenario Manual

Melakukan pengujian secara manual untuk tiap modul agar mengetahui tiap sensor dapat berfungsi dengan baik. Kombinasi dilakukan pada tiap modul untuk mengetahui sistem dapat berfungsi dan berjalan sesuai dengan arahan yang di berikan dari mikrontroler. Pengujian secara manual pada modul, diantaranya :

- Pergerakan titik lokasi GPS Neo-6M dengan tempat berbeda dan melihat hasil reaksi pada sistem.
- Data pada GPS Neo-6M dan DHT11 terlihat pada OLED I2C.
- Pengujian suhu dan kelembapan dengan DHT11 dengan memastikan nilainya dapat masuk ke dalam Blynk.

d. Menghitung Keakuratan GPS *Tracking*

Dalam perhitungan keakuratan GPS *Tracking* yang telah dibuat, maka dilakukan perbandingan dengan titik koordinat sesungguhnya yang mengacu pada Google Maps yang dilakukan pada 10 titik berbeda pada penelitian ini. Dapat dihitung perbedaan selisih jarak *error* antara data prototipe GPS *Tracking* dengan posisi sebenarnya yang merujuk pada Google Maps menggunakan persamaan 3.1 (Wicaksana & Rachman, 2018).

$$Z = \sqrt{(B - A^2) + (D - C)^2}$$

$$\text{Jarak Error} = Z \times 111.322 \text{ kilometer} \dots \dots \dots (3.1)$$

Keterangan :

Z = Nilai derajat	D = Nilai <i>longitude</i> dari GPS
A = Nilai <i>latitude</i> sebenarnya	<i>Tracking</i>
B = Nilai <i>latitude</i> dari GPS <i>Tracking</i>	1 derajat pada maps = 111.322
C = Nilai <i>longitude</i> sebenarnya	kilometer

Selain itu, menghitung nilai rata-rata dari hasil selisih jarak error yang telah didapat. Rumus yang digunakan untuk menghitung nilai rata-rata ditunjukkan pada persamaan 3.2.

$$Rata - rata = \frac{Jumlah\ seluruh\ selisih\ jarak\ error}{Jumlah\ seluruh\ data} \dots\dots\dots(3.2)$$

3.3.2 Uji Kinerja

Pada pengujian kinerja memiliki tujuan untuk dapat mengetahui bagaimana performa sistem yang dihasilkan dalam kondisi yang sebenarnya. Parameternya ialah pada kecepatan, stabilitas, dan keandalan.

a. Waktu Respons

Respon yang dihasilkan pada saat proses pengiriman pesan jika terdeteksi antar sensor. Dimana fokus utama respon yang di jadikan perhitungan ialah :

- Waktu pengiriman pesan ke mikrokontroler lain melalui CAN Bus.
- Waktu data yang muncul pada aplikasi Blynk.
- Waktu kamera aktif di *monitoring* pada aplikasi Blynk.

b. Pengukuran Beban

Perlunya di lakukan pengukuran karena dapat berpengaruh pada mikrokontroler yang memiliki batas kapasitas pada memori. Jika program yang diberikan melebihi batas maksimal, akan membuat sistem tidak dapat bekerja bahkan memasukkan program pun tidak bisa. Pengamatan yang dilakukan diantaranya :

- Pengecekan penggunaan memori ESP32 pada saat semua modul bekerja.
- Pengamatan untuk kestabilan koneksi Wi-Fi dan pada Blynk apakah tetap terhubung.

c. Durasi Operasional

Pengamatan pada sistem perlu dilakukan untuk dapat memaksimalkan kerja sistem tersebut. Diperlukan beberapa keadaan untuk dapat menghasilkan perbandingan kerja sistem tersebut bagaimana. Pengamatan yang dilakukan diantaranya :

- Menjalankan sistem dengan waktu yang lama untuk mengamati apakah ada penurunan performa.
- Pengiriman data yang dihasilkan apakah dapat tetap terkirim secara konsisten.

3.4 Teknik Analisis Data

Pada saat telah terkumpulkan seluruh data yang dihasilkan maka dapat dilakukan tahap selanjutnya ialah analisis data. Untuk teknik analisis yang digunakan pada penelitian kali ini ialah teknik statistik deskriptif. Metode statistik deskriptif adalah metode yang berfokus pada analisis data dengan cara memberikan deskripsi atau gambaran tentang data yang telah dikumpulkan, tanpa maksud membuat kesimpulan yang berlaku secara umum atau generalisasi (Manju & Mathur, 2015)