

### BAB III

#### METODOLOGI PENELITIAN

Pada bab ini akan dibahas tentang gambaran umum sistem, identifikasi kebutuhan sistem, perancangan perangkat keras dan perancangan desain sistem.. Gambaran sistem akan menjelaskan cara kerja sistem secara garis besar yang akan digambarkan oleh blok diagram sistem. Identifikasi kebutuhan sistem akan menjelaskan apa saja yang dapat dilakukan oleh sistem yang dibuat meliputi kebutuhan fungsional dan kebutuhan non-fungsional. Perancangan perangkat keras akan menjelaskan komponen apa saja yang akan digunakan serta *pin* yang terhubung antar komponen. Selanjutnya, perancangan perangkat lunak akan menjelaskan alur kerja program yang akan ditanam pada perangkat keras. Pada Penelitian ini menggunakan metode penelitian *decision table* lebih dikenal dengan *table keputusan*. Metode Penelitian *decision table* adalah metode penelitian yang menggunakan *table* yang digunakan sebagai alat bantu untuk menyelesaikan logika dalam program. Dengan demikian *table keputusan* efektif digunakan bilamana kondisi yang akan diselesaikan didalam program jumlahnya cukup banyak dan rumit[18].

#### 1.1 Alat dan Bahan

Alat dan bahan yang digunakan dalam penelitian ini adalah sebagai berikut:

Tabel 3.1 Daftar Analisis Kebutuhan Pembangunan Sistem

No	Nama	Jumlah Unit
1	Node MCU V3 ESP 8266	1
2	Sensor Ultrasonik HC-SR04	1
3	Sensor RFID RC522	1
4	Motor Servo MG996R	1
5	Modul Berat HX711	1
7	Resistor 220 ohm	2
8	Led merah	1
9	Led hijau	1
10	Kabel Jumper	18
11	Pcb dot 7x5 cm	1
12	Adaptor 5v	1

### 3.2 Metode *Decision Table*

*Decision table* mendeskripsikan dan menganalisis situasi keputusan prosedural dimana keadaan sejumlah kondisi menentukan pelaksanaan serangkaian tindakan. Metode *decision table* dapat digunakan untuk menjelaskan dan menggambarkan aliran data secara logika yang tersimpan yang menghasilkan solusi dari masalah yang akan diselesaikan. *Decision table* mengolah data numerik untuk mengurai perhitungan secara logis. Dengan demikian tabel keputusan efektif digunakan bilamana kondisi yang akan diselesaikan didalam program jumlahnya cukup banyak dan rumit, dalam membuat tabel keputusan adanya serangkaian aturan (*rule*) untuk setiap tindakan sangat diperlukan. Aturan-aturan tersebut merupakan kombinasi dari semua kemungkinan kondisi yang memunculkan suatu tindakan. Dimana sebuah aturan adalah sebuah prosedur untuk memeriksa kondisi-kondisi yang berbeda dan pernyataan tindakan yang dicantumkan diambil, kemudian pemrosesan tabel diteruskan [15].

Dalam memantau pemberian pakan kucing dapat dilakukan dengan beberapa cara, salah satunya yaitu dengan cara mengambil data menggunakan sensor. Sensor yang digunakan pada penelitian ini adalah sensor ultrasonic sebagai sensor jarak untuk mendeteksi jarak sensor dengan pakan didalam wadah tampungan, sensor load cell sebagai pengukur berat dan sisa pakan didalam piring dan sensor rfid sebagai pendeteksi objek ketika kucing sedang makan. Setelah data diperoleh kemudian data diproses dan dikirim menuju server monitoring. Masukan dari sistem ini adalah berupa data berat pakan pada piring, sisa pakan pada piring, *get time pool.ntp.org*, *sensor load cell*, *sensor rfid*, *sensor ultrasonic*. Data atau aspek-aspek yang mempengaruhi penilaian yaitu : *internet connection*, ketepatan waktu, kinerja komponen, akurasi pengukuran. Keluaran dari sistem pendukung keputusan adalah informasi tentang pilihan terbaik dari berbagai alternatif yang ditawarkan. Rancang bangun alat pemberi makan dan monitoring sisa pakan kucing berbasis IoT menggunakan metode *decision table* ini diharapkan dapat membantu para pecinta kucing dalam memonitoring hewan peliharannya dengan data-data yang diterima.

Tabel 3.2 Metode Pemberian Pakan Kucing

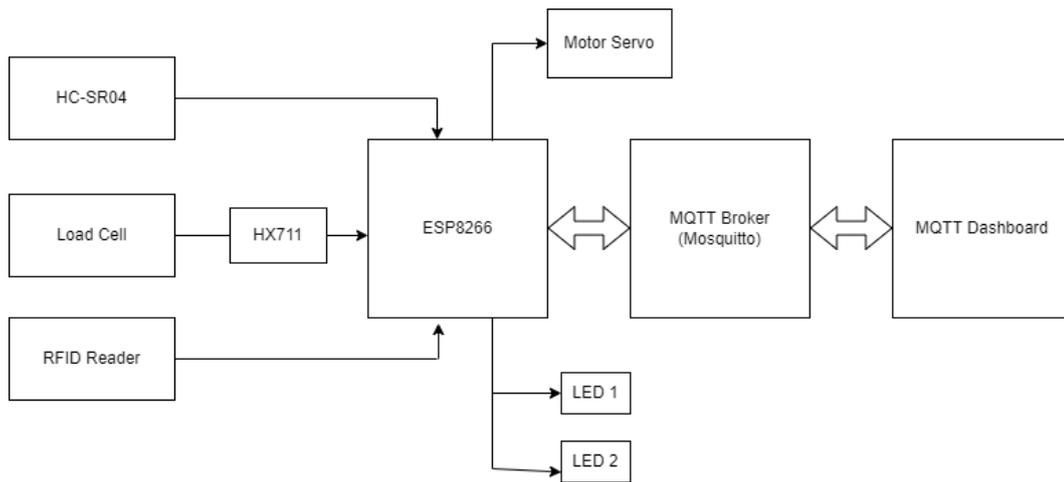
NO	Input	Proses	Waktu	Keputusan
1.	Berat pakan dalam tampungan $\leq$ 22gram	Servo siap..	07.00 13.00 19.00	Servo buka pintu sesuai waktu. Dan melakukan pemberian pakan sampai berat pakan dalam piring kucing $\geq$ 22gram
2.	Servo buka pintu sesuai inputan pada mqtt <i>dashboard</i>	Tangki pakan ngisi pakan kucing hingga beratnya $\geq$ 22 gram	Kapan saja	Servo dapat mebuca tidak menggunakan ketetapan jam
3.	<i>pool.ntp.org</i>	Referensi waktu	24 jam	Mengambil Waktu dari <i>server</i>
4.	<i>Load cell</i>	Pengukur berat pakan pada piring	Sesuai waktu jam makan	Sisa pakan dalam 1x makan
5.	RFID	Proxymity kucing mendekat	Sesuai waktu jam makan	Mendeteksi kucing sedang makan
6.	<i>Ultrasonic</i>	Pengukur volume pakan pada tanki	Ketika pakan $\geq$ 20cm	Motor servo tidak akan membuka pakan

Tabel 3.3 Metode Pengujian Sistem

No	Ketepatan Waktu	Ketersediaan Pakan	Motor Servo	Berat Pakan	Sisa Pakan	Status RFID	Presentase
1.	√	√	√	√	√	√	√
2.	√	√	√	√	√	√	√
3.	√	√	√	√	√	√	√
4.	√	√	√	√	√	√	√

### 3.3 Gambaran Umum Sistem

Rancang Bangun Alat Pemberi Makan dan Monitoring Sisa Pakan Kucing berbasis IoT menggunakan metode *decision table*. Protokol MQTT dirancang agar mampu melakukan fungsi pemantauan. Pada fungsi pemantauan, sistem dirancang untuk mampu melakukan monitoring jarak jauh sebagai pengawasan sistem ketika ada kondisi tertentu. Gambar 3.1 menunjukkan blok diagram sistem yang terdiri dari perangkat keras, VPS (*Virtual Privat Server*) dan *user*.



Gambar 3.1 Blok Diagram Sistem

Dari blok diagram cara kerja sistem untuk perangkat keras ada esp8266, RFID, *LoadCell*, Sensor *ultrasonic* dan motor servo. Data yang ada di perangkat keras tersebut akan ditampilkan pada *user*, melalui VPS (*virtual private server*) menggunakan *mqtt broker (mosquitto)* kemudian ditampilkan pada *user* dengan menggunakan *mqtt dashboard*. Apabila *user* ingin mengendalikan servo untuk membuka atau menutup pakan kucing dia akan mengirimkan perintah melalui VPS (*mqtt broker*) kemudian nanti perintah tersebut akan diterima oleh perangkat keras esp8266, dari esp8266 kemudian akan memerintahkan motor servo untuk menutup ataupun membuka.

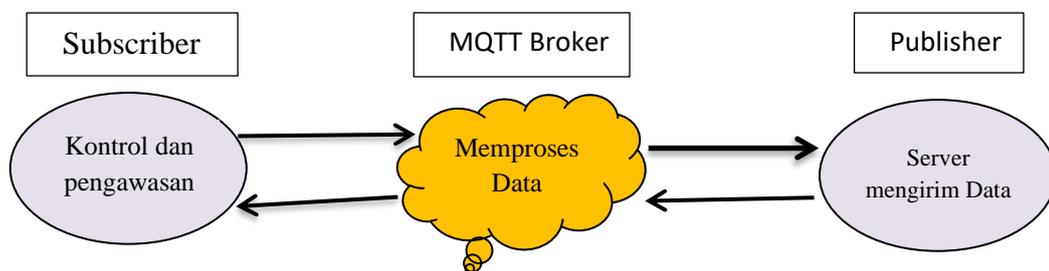
#### 3.3.1 Server dan Publisir

Dalam pembuatan tugas akhir ini terdapat beberapa fungsi yang dimiliki oleh *server*, diantaranya:

1. Sebagai penerima pesan berupa data yang dikirimkan oleh *client* yang disimpan sementara sebelum dikirimkan ke perangkat *user*.
2. Sebagai (*publisher*) atau mengirimkan pesan data yang didapat dari *client* dan kemudian dikirimkan ke MQTT *Broker*
3. Sebagai penghubung MQTT *Broker* agar pesan data tersebut sampai dan diterima oleh *user*

### 3.3.2 MQTT *Broker*

Pada tugas akhir ini MQTT *Broker* bertugas seperti halnya server yang bertugas dan menerima dan mengirimkan data, akan tetapi server nodeMCU tidak bisa mengirimkan secara langsung ke *subscriber(user)*. Alur dari pemrosesan data dari server MQTT *Broker* ditunjukkan pada gambar 3.2



Gambar 3.2 Alur pemrosesan data ke MQTT *Broker*

Alur dari proses pengiriman data yang dilakukan yaitu, *Server* mengirimkan data-data yang didapat dari *client* ke *broker* kemudian data tersebut akan diterima dan diproses oleh *broker*. Sebelum proses pengiriman tersebut terjadi, ada suatu syarat agar pengirim tersebut sampai di broker yaitu dengan mengatur *topic,username,password port* dan MQTT *server* pada *server* harus sama dengan yang dimiliki oleh *broker*. Apabila syarat tersebut sudah terpenuhi maka dari data *server* akan berhasil diterima ke *broker*. Selanjutnya broker akan mengirimkan data tersebut ke *subscriber*, ketika sudah diterima oleh *subscriber*, *subscriber* juga bisa *mem-publish* sebuah data dengan menggunakan sistem aplikasi pada *subscriber* *Broker* juga menerima dan memproses data tersebut lalu dikirimkan ke *server* berupa *Output* yang diinginkan.

### 3.3.3 *Subscriber 1*

Bertugas sebagai mengawasi atau memonitoring dari mqtt *broker* apakah pesan dari *publisher* sudah masuk atau tidak.

### 3.3.4 *Subscriber 2*

Bertugas sebagai mengawasi atau memonitoring dari mqtt *broker* apakah pesan dari *publisher* sudah masuk atau tidak dan mengontrol motor servo untuk membuka katup.

## 3.4 Identifikasi Kebutuhan Sistem

Identifikasi kebutuhan sistem meliputi kebutuhan fungsional dan kebutuhan non-fungsional. Berikut adalah kebutuhan sistem yang akan dikembangkan.

### 3.4.1 Kebutuhan Fungsional

Kebutuhan fungsional dalam perancangan sistem ini yaitu:

1. Sistem mampu melakukan pemantauan sisa pakan yang berada pada wadah tampungan pakan dan akan mengirimkan notifikasi jika dalam wadah tampungan pakan sudah mulai habis berdasarkan masukan dari sensor *ultrasonic*;
2. Sistem mampu melakukan pemantauan berat dan sisa pakan pada wadah pakan kucing berdasarkan masukan dari sensor *load cell*;
3. Sistem mampu melakukan pemantauan ketika kucing sedang makan berdasarkan masukan dari sensor *RFID*
4. Sistem mampu melakukan buka tutup katup pada wadah tampungan yang akan mengalir ke wadah pakan berdasarkan masukan motor servo.
5. Kandang *node* dan *server* pada sistem mampu berkomunikasi melalui protokol MQTT;
6. Sistem mampu menampilkan hasil pembacaan sensor *load cell* pada aplikasi *MQTT dashboard*
7. Sistem mampu berkomunikasi dengan aplikasi *MQTT dashboard* yang didalam aplikasi ini sudah tertera keluaran data penjadwalan pemberian pakan.

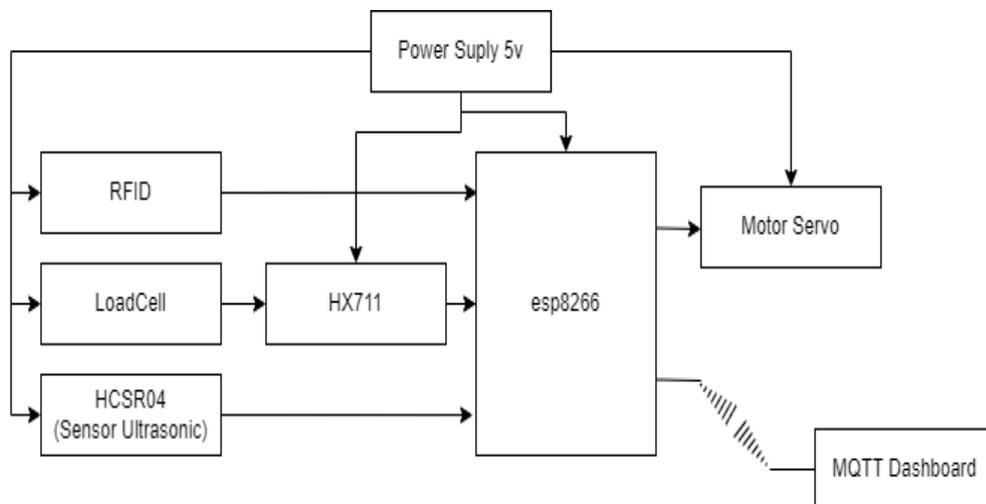
### 3.4.2 Kebutuhan Non-Fungsional

Kebutuhan non-fungsional dalam perancangan sistem ini yaitu:

1. Sistem menggunakan Node MCU ESP8266 sebagai pusat kendali kandang *node* untuk membaca keluaran sensor *ultrasonic* , sensor *load cell* serta sensor RFID.
2. Sistem menggunakan *virtual private server (VPS)* sebagai *server* untuk perantara pesan menuju *user/Mqtt Broker*
3. Sistem dijalankan dengan catu daya sebesar 5V 2A.
4. Sistem pada vertikultur *node* menggunakan bahasa pemrograman C dengan perangkat lunak aplikasi Arduino IDE versi 1.8.9 yang dijalankan pada sistem operasi *Windows10*.
6. *MQTT dashboard* berjalan pada aplikasi android
7. Purwarupa berupa wadah tampungan, wadah pakan dan komponen elektronika.

### 3.5 Perancangan Perangkat Keras

Perancangan perangkat keras menjelaskan tentang komponen elektronik yang akan digunakan untuk membangun sistem baik komponen utama maupun komponen pendukung. Gambar 3.2 menunjukkan blok diagram perangkat keras yang terdiri dari mikrokontroler, sensor, motor servo dan *mqtt dashboard*



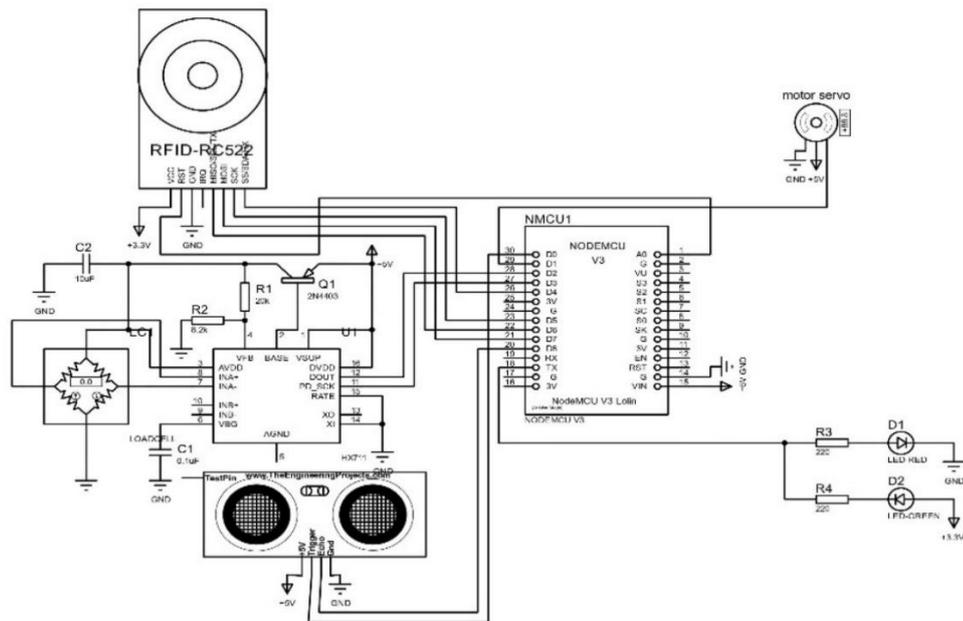
Gambar 3.3 Blok Diagram perangkat keras

Pada blok diagram perangkat keras komponen yang terhubung pada *Mikrokontroller NodeMCU ESP8266* yaitu sensor *ultasonic* sebagai sensor pendeteksi sisa pakan dalam tampungan, sensor *load cell* sebagai pendeteksi jumlah takaran pakan yang akan diberikan, sensor RFID sebagai pendeteksi bahwa objek atau kucing sedang makan, motor servo dimana motor servo ini sebagai pembuka dan penutup katup pada wadah tampungan.

Pada blok diagram perangkat keras disini menggunakan power suplay 5 volt untuk sumber daya dari masing-masing perangkat keras diantaranya motor servo, esp8266, hx711(modul berat), rfid, sensor *load cell*, sensor *ultasonic*. Disini terlihat jadi masing-masing sensor seperti rfid dan sensor *ultasonic* mengirimkan data ke mikrokontroler esp8266, akan tetapi untuk sensor *load cell* sendiri mengirimkan data pada Hx711 sebagai modul berat, kemudian baru dari modul berat hx711 dikirimkan pada esp8266 kemudian data diolah. Semisal apabila diprintahkan untuk motor servo bergerak maka esp8266 akan memerintahkan motor servo untuk bergerak, dan setiap informasi dari sensor esp8266 akan mengirimkan data pada *mqtt dashboard* melalui koneksi internet.

### 3.5.1 *Schematic* Rancangan Sistem

Desain sirkuit elektronik diperlukan sebagai pedoman dalam pembuatan sirkuit elektronik. Sebelum melanjutkan dengan pembuatan sirkuit elektronik, desain sirkuit elektronik harus didesain atau didesain terlebih dahulu. Proses perancangan sirkuit elektronik dapat dilakukan secara manual atau melalui *software* yaitu *Proteus*, agar dapat menjadi acuan dalam membuat rangkaian elektronika secara *real*. Berikut adalah desain rangkaian elektronika menggunakan *software Proteus*.



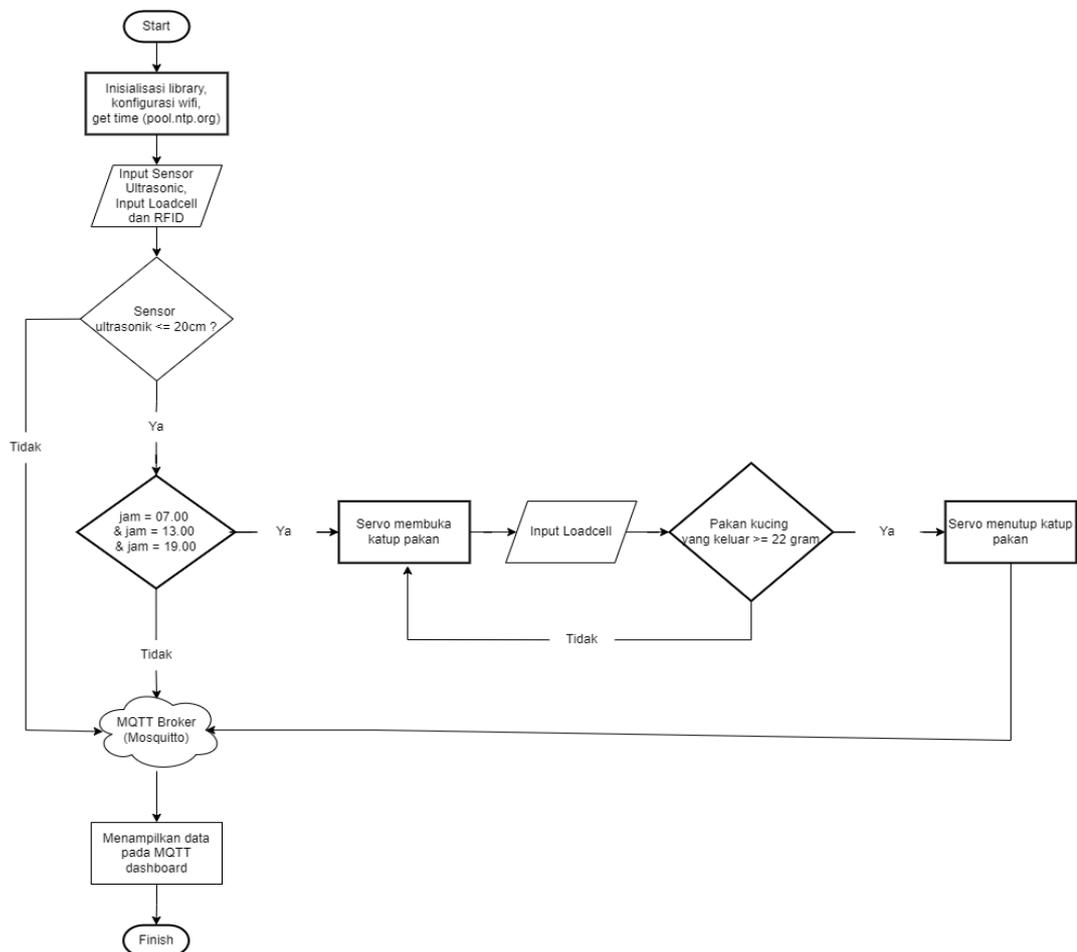
Gambar 3.4 *Schematic system*

Bisa dilihat pada gambar 3.4 merupakan rancangan *scematic* sistem dimana pada sensor rfid vcc terhubung ke tegangan 5 volt dan ground terhubung ke *ground* kemudian pin *reset* terhubung ke A0 kemudian pin miso terhubung ke D6 untuk pin mosi terhubung ke D7 dan SCK dihubungkan ke D5 lalu pin SDA terhubung ke D2. Kemudian pada *sensor ultrasonic* pin vcc terhubung ke tegangan 5 volt dan *ground* terhubung ke *ground* kemudian *trigger* terhubung ke pin D0 untuk echo terhubung ke pin D4. Kemudian motor servo pin data terhubung ke D1, tegangan terhubung ke 5 volt dan ground terhubung ke *ground*. Kemudian pada rangkaian *load cell*, dari *load cell* penulis hubungkan ke modul Hx711 terdapat rangkain modulnya rangkain ini semua sudah *ter-integrated* pada modul hx711. Kemudian dari modul hx711 untuk vcc terhubung ke 5 volt, *ground* ke *ground* sedangkan untuk data ke mikrokontrollernya Dout terhubung ke pin D6, dan Pin PD terhubung ke Gpio1.

### 3.5.2 Flowchart Sistem

*Flowchart* atau bagan alur adalah diagram yang menampilkan langkah-langkah dan keputusan untuk melakukan sebuah proses dari suatu program. Setiap langkah digambarkan dalam bentuk diagram dan dihubungkan dengan garis atau

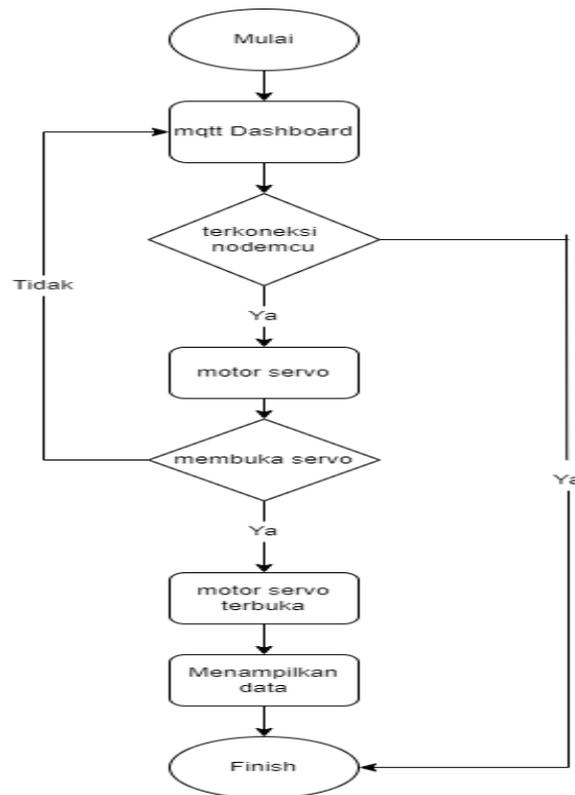
arah panah. *Flowchart* berperan penting dalam memutuskan sebuah langkah atau fungsionalitas dari sebuah proyek pembuatan program yang melibatkan banyak orang sekaligus. Selain itu dengan menggunakan bagan alur proses dari sebuah program akan lebih jelas, ringkas, dan mengurangi kemungkinan untuk salah penafsiran. Penggunaan *flowchart* dalam dunia pemrograman juga merupakan cara yang bagus untuk menghubungkan antara kebutuhan teknis dan non-teknis [17]. Berikut merupakan *flowchart* alur cara kerja *system* yang akan dibuat



Gambar 3.5 *Flowchart* system

Alur dari *flowchart* sistem diawali dengan *star* kemudian melakukan inisialisasi library, konfigurasi wifi dan *get time* atau mengambil waktu dari *poll.ntp.org*. Kemudian input sensor *ultrasonic*, *input sensor load cell* dan sensor *RFID*, dan jika sensor *ultrasonic* mendeteksi dari atas tampungan jaraknya lebih dari 20 cm maka proses pemberian pakan tidak akan bekerja melainkan akan mengirimkan data pada *mqtt broker* dan diterima oleh *mqtt dashboard* untuk melakukan pengisian pakan ulang, dan jika ketinggian pakan dari atas tampungan

kurang dari 20 cm dalam artian pakan kucing masih banyak maka sistem akan menjalankan proses pemberian pakan secara otomatis. Untuk yang pertama jika jamnya menunjukan pukul 07.00, 13.00, 19.00 maka servo akan membuka katup makan, kemudian *load cell* akan menimbang jika pakan yang keluar sudah lebih dari 22 gram maka servo akan menutup. Jika jamnya tidak menunjukan waktu 07.00, 13.00, 19.00 maka sistem akan langsung mengirimkan data pada *mqtt broker* dan menampilkan data pada *mqtt dashboard*. Kemudian data ditampilkan pada layar monitor *mqtt dashboard*.



Gambar 3.6 *Flowchart Mqtt*

Pada *Flowchart* mqtt dimulai pada aplikasi pemberi pakan mqtt dashboard yang terkoneksi dengan nodeMCU dengan motor servo terhubung ke aplikasi *smartphone*. Terdapat menu motor servo dengan perintah membuka motor servo maka aplikasi akan mengirimkan data kepada nodemcu maka servo akan terbuka dan data pemberian pakan akan ditampilkan pada layar aplikasi.