



Hak cipta dan penggunaan kembali:

Lisensi ini mengizinkan setiap orang untuk menggubah, memperbaiki, dan membuat ciptaan turunan bukan untuk kepentingan komersial, selama anda mencantumkan nama penulis dan melisensikan ciptaan turunan dengan syarat yang serupa dengan ciptaan asli.

Copyright and reuse:

This license lets you remix, tweak, and build upon work non-commercially, as long as you credit the origin creator and license it on your new creations under the identical terms.

BAB III

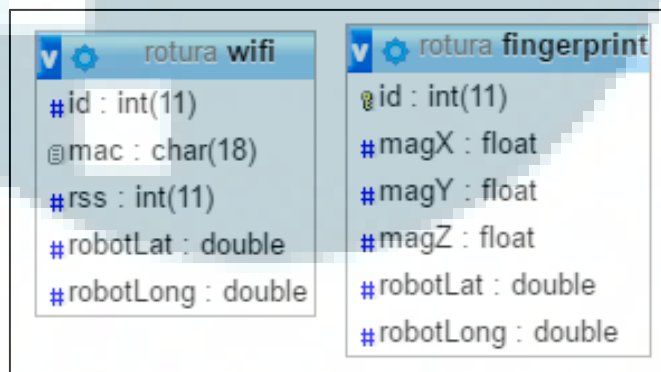
RANCANGAN SISTEM

3.1 Perancangan Metode Indoor Positioning System

Perancangan yang dibutuhkan untuk membandingkan metode *Indoor Positioning System* (IPS) yang menggunakan *magnetic fingerprint* dan WiFi RSS dibagi menjadi perancangan *database* dan perancangan denah.

3.1.1 Perancangan Database

Perancangan *database* dibagi menjadi 2 (dua) bagian, yaitu *database* untuk menganalisa lokasi dan *database* untuk hasil uji coba. *Database* yang digunakan adalah SQLite untuk digunakan pada android. *Database* untuk menganalisa lokasi dibuat seperti pada Gambar 3.1.



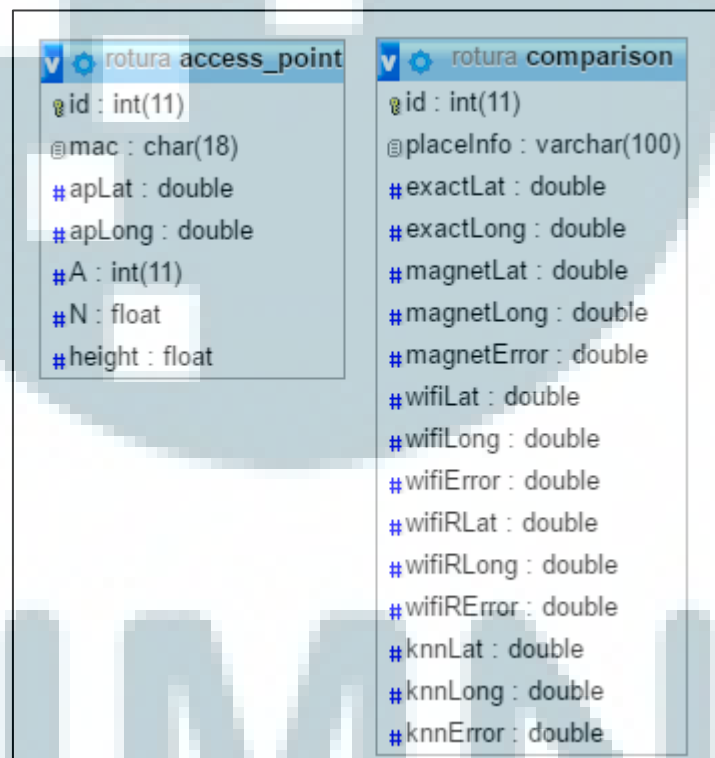
rotura wifi	rotura fingerprint
#id : int(11)	#id : int(11)
#mac : char(18)	#magX : float
#rss : int(11)	#magY : float
#robotLat : double	#magZ : float
#robotLong : double	#robotLat : double
	#robotLong : double

Gambar 3.1 Database Analisa Lokasi

Database analisa lokasi digunakan untuk menyimpan *magnet fingerprint* dan RSS dari *access point* (AP). Tabel *Fingerprint* yang berisi besaran magnet bumi dan posisi nantinya digunakan untuk uji coba pada metode *magnet fingerprint*. Besaran magnet menggunakan tipe data float disesuaikan dengan tipe data yang didapatkan oleh sensor, sedangkan robotLat dan robotLong mengikuti tipe data dari *latitude* dan *longitude*.

Tabel wifi yang berisi informasi RSS dari AP pada posisi tertentu digunakan untuk menganalisa jangkauan AP pada Gedung C Universitas Multimedia Nusantara. MAC menggunakan tipe data char 18 karena nilai digit MAC akan selalu sama yaitu 12 digit ditambah dengan pemisah “:” setiap 2 digit, dan pada tipe data char, diperlukan 1 digit untuk char ‘\0’ yang merupakan penanda akhir, sehingga memiliki total 18 digit. RSS menggunakan tipe data integer karena memiliki nilai diantara 0 s.d -110 dBm.

Setelah menganalisa lokasi, tabel lokasi AP pada denah dan tabel hasil komparasi akan ditambahkan seperti pada Gambar 3.2.



rotura access_point	rotura comparison
id : int(11)	id : int(11)
mac : char(18)	placeInfo : varchar(100)
#apLat : double	#exactLat : double
#apLong : double	#exactLong : double
#A : int(11)	#magnetLat : double
#N : float	#magnetLong : double
#height : float	#magnetError : double
	#wifiLat : double
	#wifiLong : double
	#wifiError : double
	#wifiRLat : double
	#wifiRLong : double
	#wifiRError : double
	#knnLat : double
	#knnLong : double
	#knnError : double

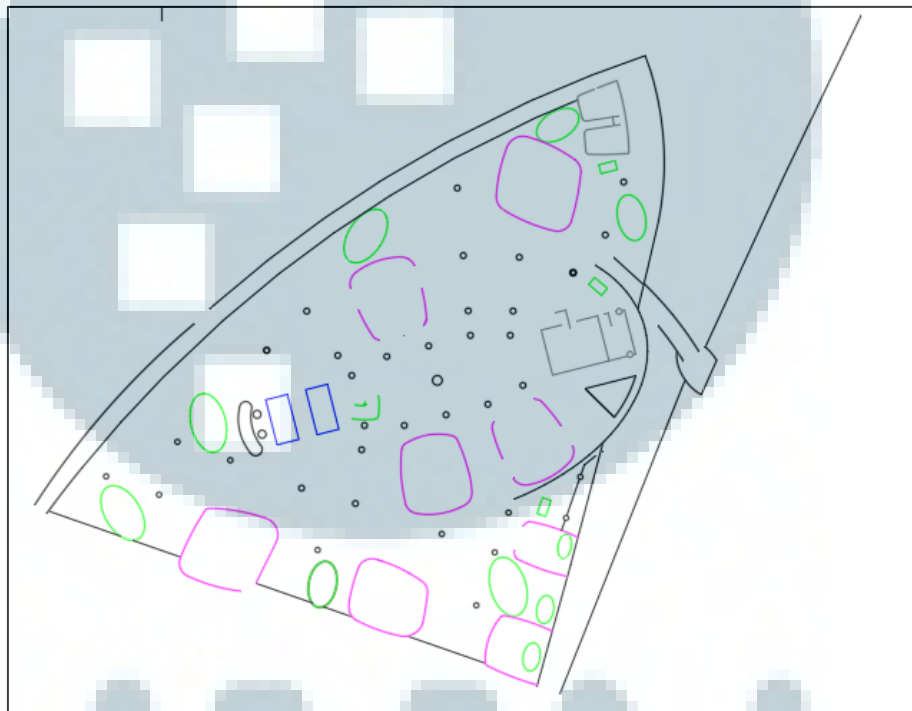
Gambar 3.2 Tabel Komparasi dan Akses Poin

Tabel AP digunakan untuk mengetahui lokasi dari tiap-tiap AP pada denah, dan juga menyimpan informasi tinggi AP, A yang merupakan RSS (dBm) AP pada jarak 1 m, dan konstanta N. Tabel *comparison* digunakan untuk mempermudah dalam membandingkan metode IPS. Id digunakan

sebagai *identifier* baris, placeinfo digunakan untuk keterangan tempat, sedangkan sisanya merupakan nilai *latitude*, *longitude*, dan kesalahan tiap algoritma yang digunakan.

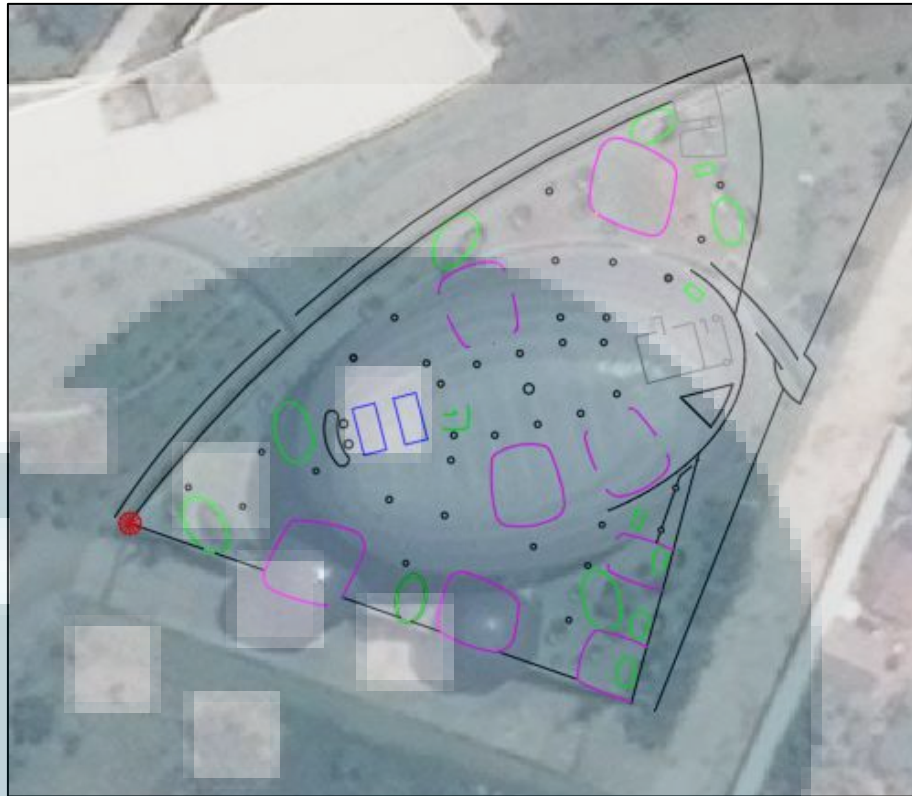
3.1.2 Perancangan Denah

Desain denah diambil dari denah Gedung C Universitas Multimedia Nusantara yang menggunakan format DWG yang dapat dibuka dengan Autocad Civil 3D dan merupakan salah satu format gambar vektor.



Gambar 3.3 Denah Gedung C UMN

Dari denah tersebut, akan diberi *georeferencing* menggunakan Autocad Civil 3D 2017 dengan koordinat sistem yang dipakai pada uji coba adalah WGS84 WORLD-MERCATOR dengan Time Zone Bangkok, Hanoi, Jakarta. Hasil dari *georeferencing* tersebut adalah seperti pada Gambar 3.4.



Gambar 3.4 Denah Setelah Georeferencing

Setelah itu denah dapat di-export menjadi format *Keyhole Markup language Zipped* (KMZ) dan diubah menjadi *Keyhole Markup Language* (KML) dengan menggunakan aplikasi Google Earth. KML ini nantinya akan diterjemahkan menjadi gambar vektor pada *map* di Android.

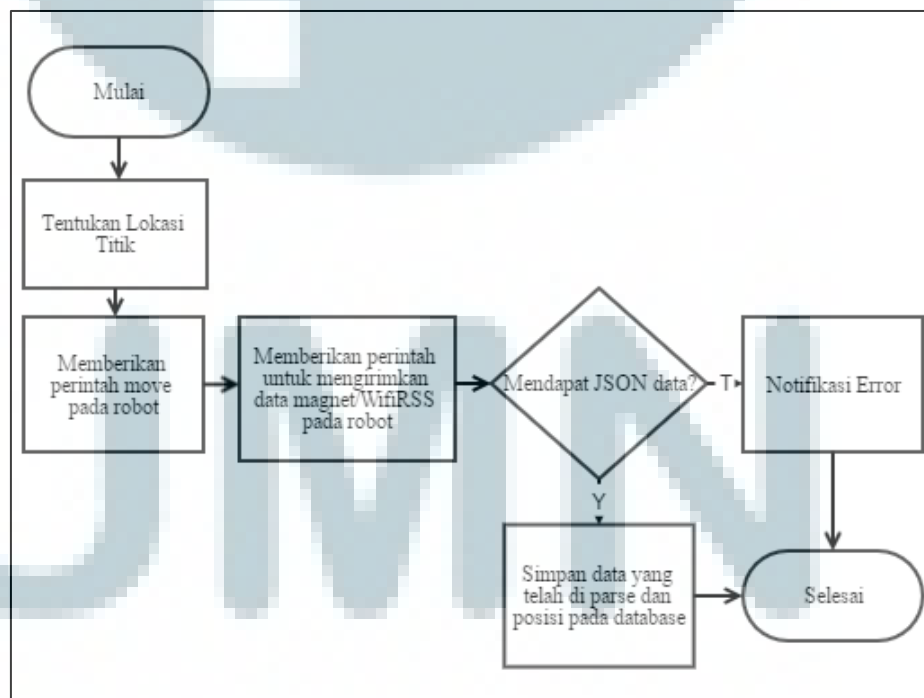
Gambar vektor memiliki kelebihan-kelebihan dibandingkan gambar dengan format lainnya [27]. Dengan menggunakan vektor, gambar dapat diperbesar tanpa menghasilkan distorsi. Gambar vektor juga lebih detail dan terlihat tajam. Selain itu vektor juga memiliki ukuran file yang relatif kecil sehingga tidak menghabiskan memori *smartphone*.

3.2 Perancangan Aplikasi Android

Aplikasi android dibuat dengan Android Studio versi 2.2.2. Pada aplikasi android, terdapat 3 (tiga fungsi utama) yaitu *location analyzer*, *database management*, dan *detect location*.

3.2.1 Fungsi Analisa Lokasi

Location analyzer digunakan untuk mendapatkan data-data yang digunakan untuk analisa lokasi dan juga pengambilan *magnet fingerprint*. Pada *location analyzer*, android dapat menggerakkan robot dan meminta data RSS maupun besaran magnet pada titik yang telah ditentukan dan menyimpannya dalam *database* seperti pada Gambar 3.5. Data magnet tersebut nantinya digunakan sebagai *magnet fingerprint*, sedangkan data RSS digunakan untuk menganalisa cakupan akses poin dan juga nilai konstanta n dan A yang akan digunakan dalam perhitungan nantinya.



Gambar 3.5 Diagram Alir Analisa Lokasi

3.2.2 Fungsi Database Management

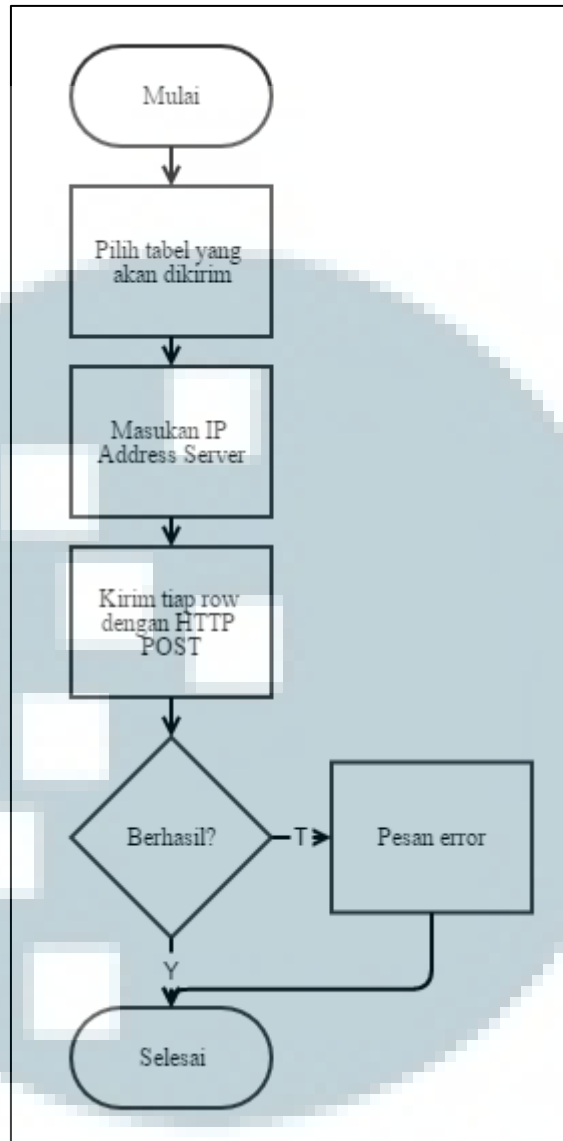
Database management digunakan untuk melihat *database* yang telah disimpan, mengirim *database* yang didapat, dan meminta *database* pada *server*.

Database yang disimpan akan ditampilkan dalam bentuk *list* sesuai dengan pilihan tabel yang ingin dilihat. Gambar 3.6 adalah diagram alir untuk melihat data. *User* akan diminta untuk memilih tabel yang ingin dilihat, setelah itu data akan ditampilkan dalam *list*.



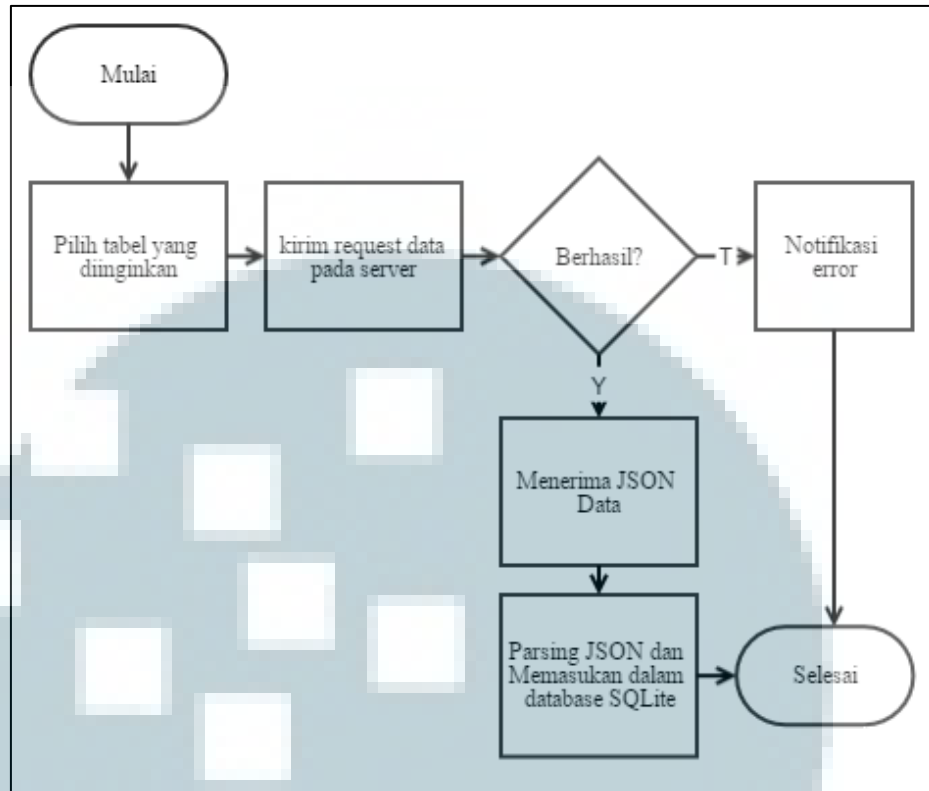
Gambar 3.6 Diagram Alir Melihat Data

Gambar 3.7 adalah diagram alir untuk mengirim data. Pengiriman *database* dilakukan dengan memilih tabel yang akan dikirim dan memasukkan IP Address dari *server*. Setelah itu aplikasi akan mengirimkan ke *server* tersebut tiap *row* tabel dengan HTTP POST dan akan menampilkan pesan *error* apabila gagal.



Gambar 3.7 Diagram Alir Mengirim Data

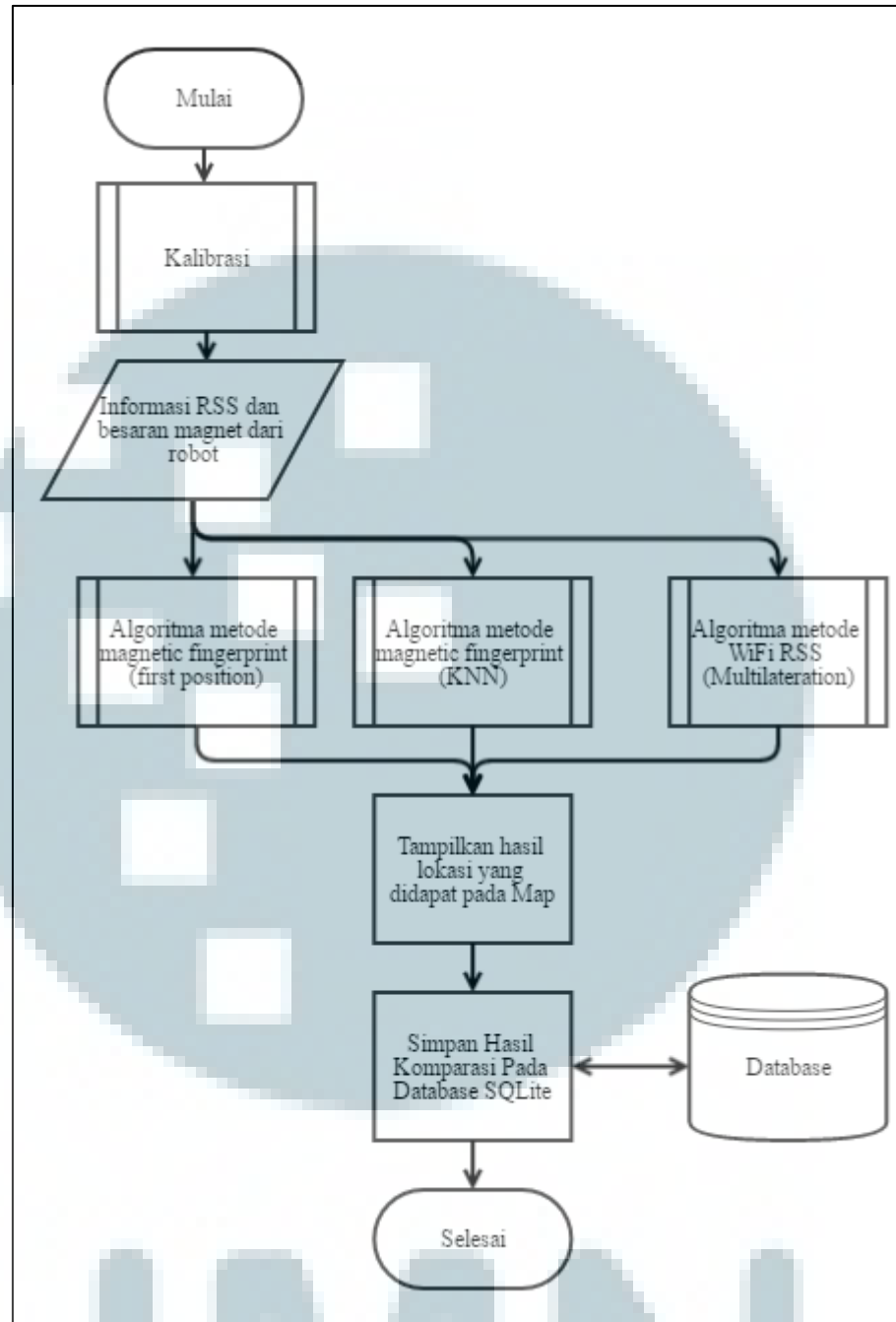
Gambar 3.8 adalah diagram alir untuk meminta data. Pada saat meminta data pada *server*, aplikasi kan meminta *user* untuk memilih tabel dan IP Address dari *server*. *Server* akan mengembalikan data berupa JSON yang akan diolah oleh aplikasi lalu dimasukan dalam *database* aplikasi. Aplikasi akan menampilkan pesan *error* apabila permintaan ke *server* gagal.



Gambar 3.8 Diagram Alir Meminta Data

3.2.3 Fungsi Deteksi Lokasi

Detect location digunakan untuk membandingkan hasil dari metode-metode yang digunakan untuk uji coba. Gambar 3.9 adalah diagram alir untuk membandingkan hasil. Hasil komparasi antara posisi sebenarnya, titik dari metode *magnet fingerprint*, dan titik dari metode WiFi RSS akan ditampilkan pada *map* di Android dengan titik yang berbeda. Setelah itu lokasi titik tersebut akan dimasukan di dalam *database* untuk dianalisa nantinya.

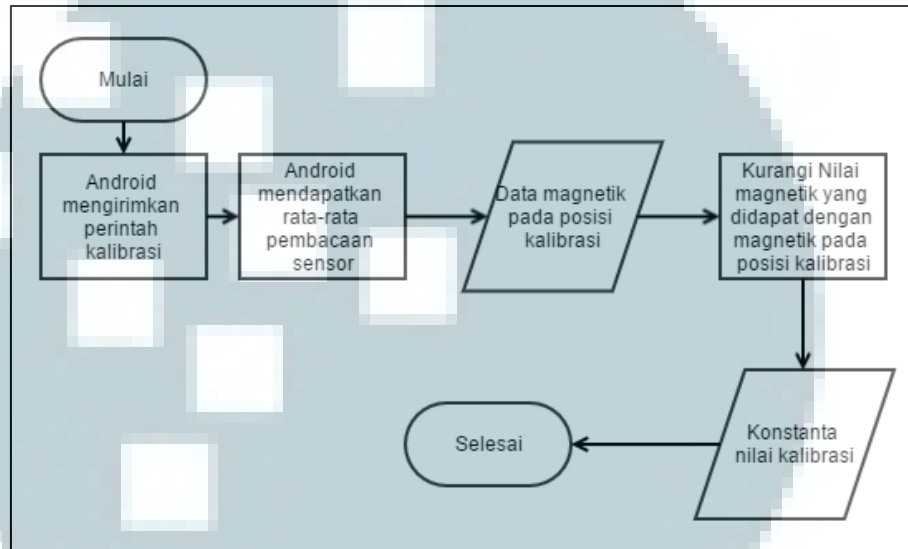


Gambar 3.9 Diagram Alir Membandingkan Hasil

Algoritma yang digunakan untuk metode *magnet fingerprint* adalah *first position* [1] dan KNN[28]. Sedangkan algoritma yang digunakan untuk WiFi RSS adalah Multilateration [14].

Subproses fungsi *first position* [1] dilakukan dengan melakukan kalibrasi titik awal untuk mendapatkan nilai *offset* dari komponen magnet X

dan Y seperti pada Gambar 3.10. Nilai *offset* tersebut digunakan untuk mengurangi tiap nilai magnet X dan Y yang terbaca oleh sensor sebelum dibandingkan dengan *database*. Pada penelitian ini, besaran dari sensor yang didapat android berasal dari robot. Hasil rata-rata pembacaan tersebut akan dikirimkan ke Android untuk mencari *offset*.



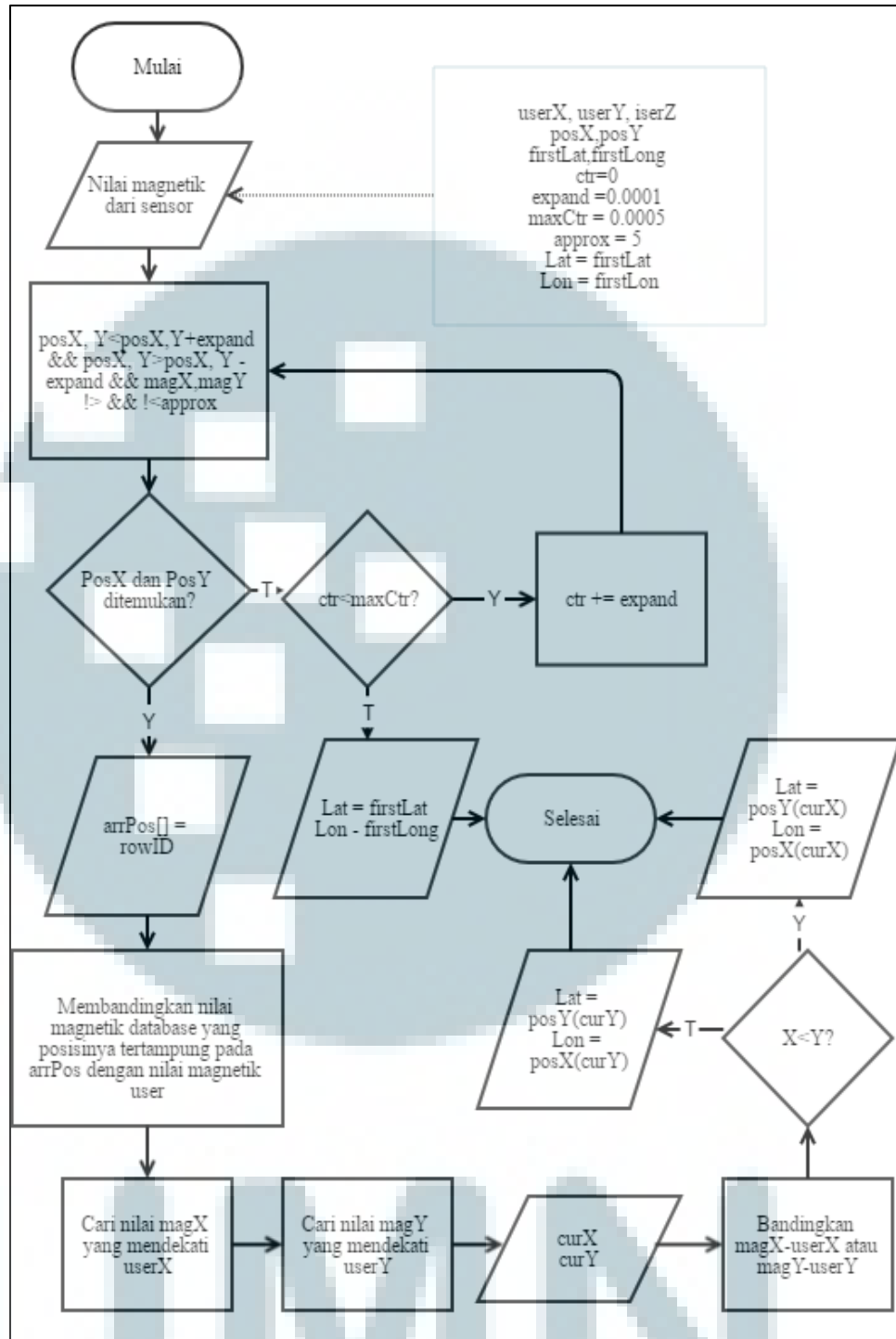
Gambar 3.10 Diagram Alir Subproses Kalibrasi

Selain untuk menentukan *offset*, titik awal tersebut juga akan menjadi acuan posisi untuk menentukan posisi pada saat melakukan uji coba. Setelah Android mendapat nilai magnetik dari robot, akan dicari posisi disekitar titik *first position* dengan nilai *expand* yang telah ditentukan seperti pada Gambar 3.11. Apabila dalam *expand* tersebut tidak ditemukan, maka *expand* akan ditambah selama tidak melebihi batas maks *expand*. Apabila posisi ditemukan, maka tiap posisi yang ditemukan akan ditampung dalam *array*. Setelah itu nilai magnetik dari sensor akan dibandingkan dengan nilai magnetik didalam *database*. Sistem akan mencari nilai magX yang mendekati userX dan nilai magY yang mendekati userY dan menyimpan

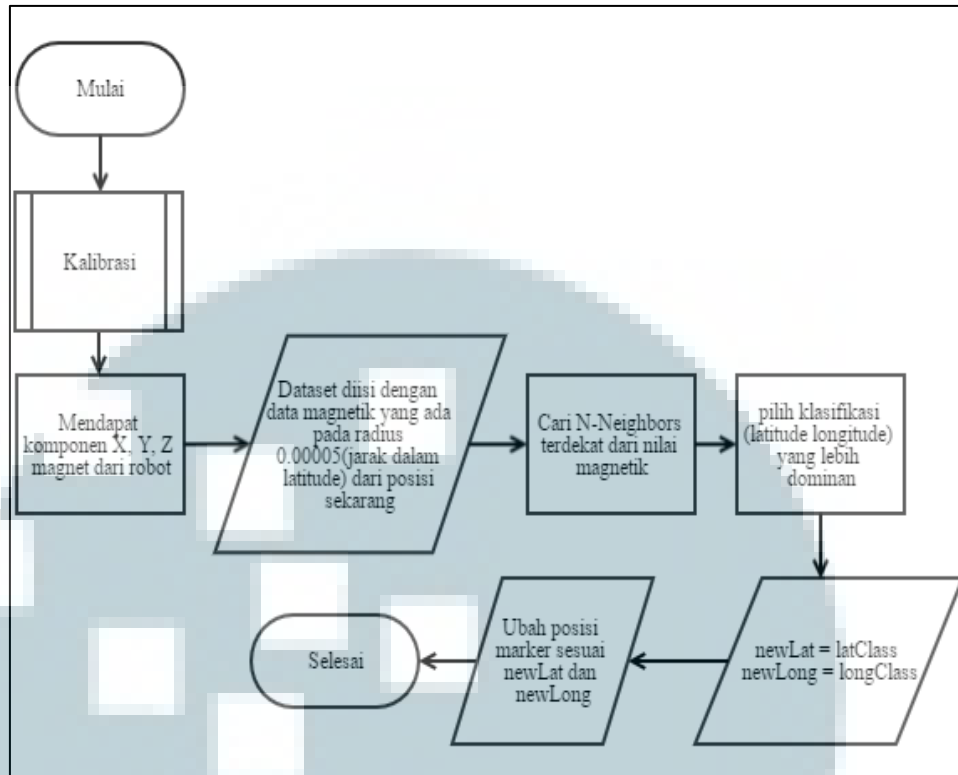
index dari *array* tersebut dalam *curX* dan *curY*. Nilai dari $\text{magX} - \text{userX}$ dan $\text{magY} - \text{userY}$ akan dibandingkan. Apabila perbandingan nilai X lebih kecil dari Y, maka sistem akan menggunakan *array index* ke-*currX* untuk mendapatkan lokasi. Jika tidak, maka *array index* ke-*currY* yang akan digunakan.

Algoritma lainnya adalah algoritma KNN [28] yang ditunjukkan pada Gambar 3.12. KNN memiliki *dataset* yang tiap datanya berisi besaran magnetic bumi dan klasifikasinya. Klasifikasi yang dimaksud adalah nilai *latitude* dan *longitude*. KNN menentukan posisi berdasarkan klasifikasi yang lebih dominan pada *n* nilai magnetik terdekat. Oleh Karena itu, satu titik *latitude* dan *longitude* akan memiliki beberapa nilai besaran magnetik.

Setelah melakukan kalibrasi seperti pada algoritma *first position*, *dataset* akan dibuat berdasarkan nilai disekitar posisi awal dengan radius 0.00005 (jarak dalam *latitude longitude*). Setiap robot mengirimkan nilai magnetik, *dataset* akan diubah berdasarkan posisi sebelumnya. Setelah itu KNN akan menentukan *n* nilai magnetik terdekat dan mengklasifikasikan nilai magnetik yang didapat dalam *latitude* dan *longitude*. Setelah itu *latitude* dan *longitude* yang didapat akan menjadi posisi robot yang baru.



Gambar 3.11 Diagram Alir Subproses Metode Magnetic Fingerprint (First Position) [1]



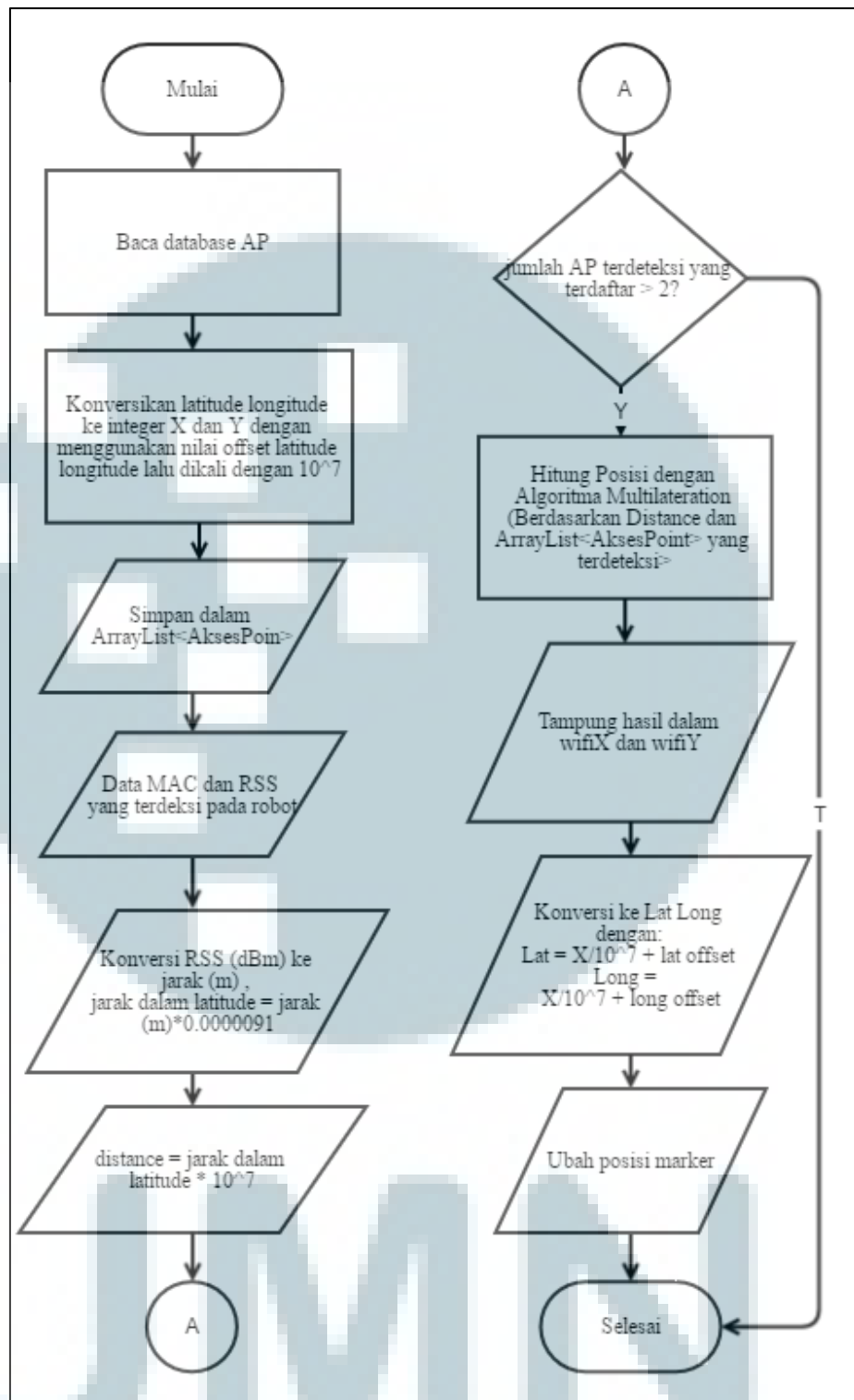
Gambar 3.12 Diagram Alir Subproses Metode Magnetic Fingerprint (KNN)

Pada multilateration seperti pada Gambar 3.13, perhitungan yang digunakan menggunakan posisi dalam bentuk X dan Y. Nilai *latitude* dan *longitude* posisi AP pada database akan diubah dalam bentuk X dan Y dengan cara menguranginya dengan *latitude* dan *longitude offset* (digit yang tidak berubah) dan dikali dengan 10^7 untuk mendapatkan 7 digit dalam X dan Y. RSSI (dBm) dari AP yang terbaca dan sudah terdaftar dalam *database* akan diubah menjadi jarak (m) dengan Equation (1).

Jarak yang didapat akan dirubah lagi menjadi jarak dalam nilai *latitude* dengan 0.0000091 untuk 1 meternya lalu dikali dengan 10^7 untuk mendapatkan nilai jarak dalam X dan Y. Aplikasi juga akan melihat apakah jumlah AP yang terdeteksi oleh robot dan terdaftar dalam *database* melebihi 2 (dua). Apabila melebihi 2 (dua) maka algoritma multilateration dapat

dijalankan. Algoritma tersebut akan menghasilkan posisi dalam X dan Y. Nilai X dan Y tersebut akan dikonversikan menjadi latitude (X) dan longitude (Y) dengan membaginya dengan 10^7 lalu menambahnya dengan nilai *offset latitude* untuk X dan *offset longitude* untuk Y. Nilai *latitude* dan *longitude* yang didapat tersebut akan dipersentasikan dalam posisi *marker* pada map.





Gambar 3.13 Diagram Alir Subproses Algoritma Metode WiFi RSS (Multilateration)

3.3 Perancangan Robot

Perancangan Robot Penunjuk Lokasi dibedakan menjadi 2 (dua) bagian, yaitu perancangan *hardware* dan perancangan *software*.

3.3.1 Perancangan Hardware

Robot dibuat dengan komponen-komponen sebagai berikut.

- a. ESP8266 NodeMCU
- b. Modul GY-273 HMC5883L (kompas digital) sebagai modul untuk mendapatkan nilai X, Y, Z dari medan magnet.
- c. Motor DC, *gearbox*, dan Rodanya sebagai alat penggerak.
- d. L293D *motor driver* sebagai modul agar NodeMCU dapat menggerakkan motor DC ke dua arah.
- e. *Logic Level Converter* (LLC) untuk membantu NodeMCU yang memiliki *output* 3.3V dalam mengontrol L293D yang membutuhkan *input* 5V.
- f. Baterai Alkaline 1.5V 4 buah sebagai sumber daya untuk motor DC.
- g. Baterai Alkaline 9V 1 buah sebagai sumber daya untuk NodeMCU.
- h. L7805CV untuk menurunkan tegangan baterai Alkaline 9V ke 5V.
- i. Saklar ON/OFF 2 buah.
- j. Akrilik untuk *body* robot.

Gambar 3.14 merupakan skematik dari robot. Robot menggunakan 2 (dua) sumber tenaga yaitu 9V dan 6V. Sumber tegangan 9V akan diturunkan menjadi 5V dengan L7805CV dan digunakan sebagai tenaga dari NodeMCU dan LLC, sedangkan sumber tegangan 6V digunakan sebagai sumber tenaga dari L293D yang outputnya akan dipakai untuk

menggerakan motor DC. Untuk mengontrol L293D, pin D3, D4, D5, dan D6 dari NodeMCU yang memiliki tegangan 3V akan dihubungkan ke LLC bagian *low voltage* (pin A1, A2, A3, dan A4). LLC akan menaikkan tegangan tersebut menjadi 5V sebagai hasil pada output pin B1, B2, B3, dan B4. Output pin tersebut digunakan sebagai input pada L293D pada pin In1, In2, In3, dan In4 yang digunakan untuk menggerakan dan mengatur arah gerak motor DC.





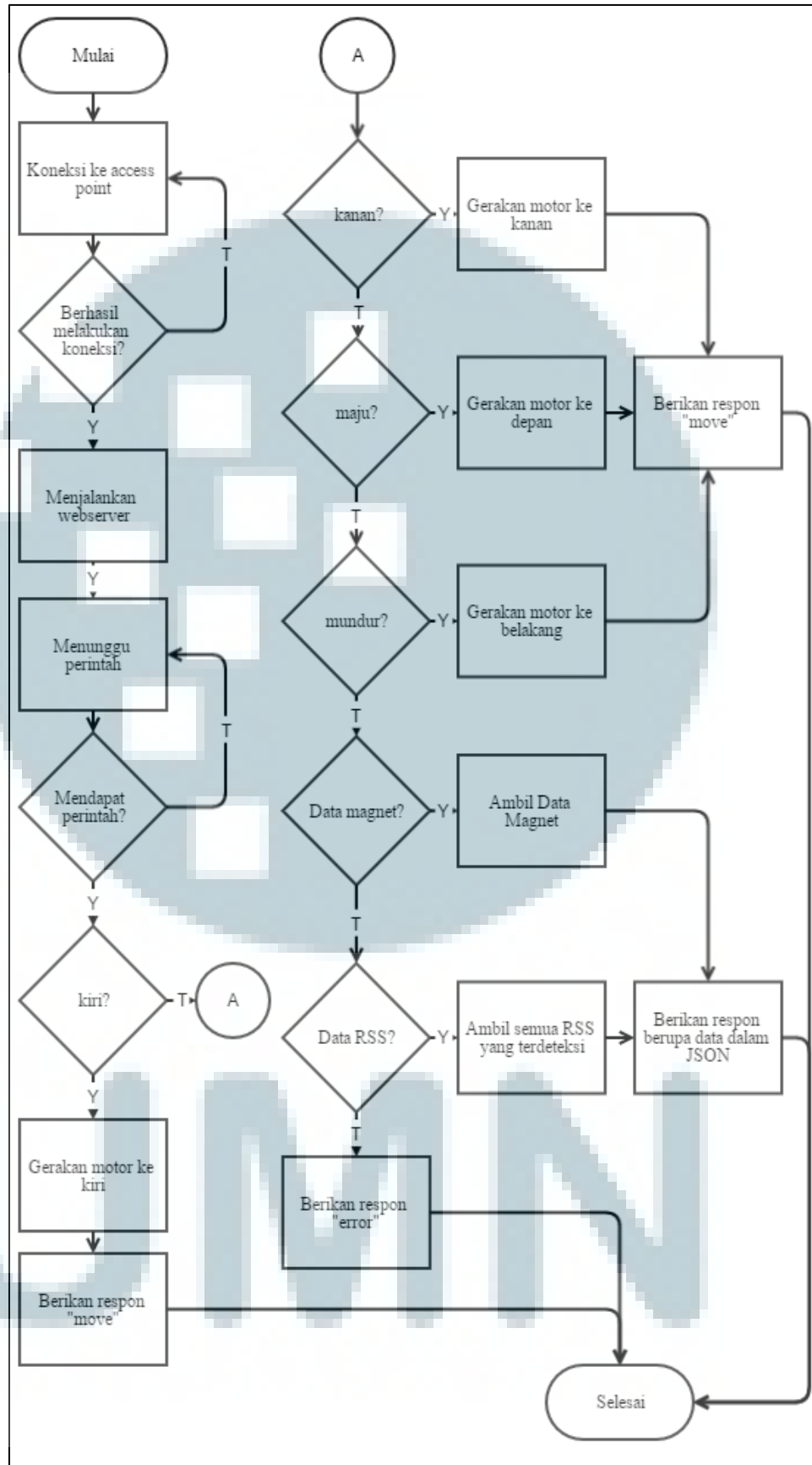
34

3.3.2 Perancangan Software Robot

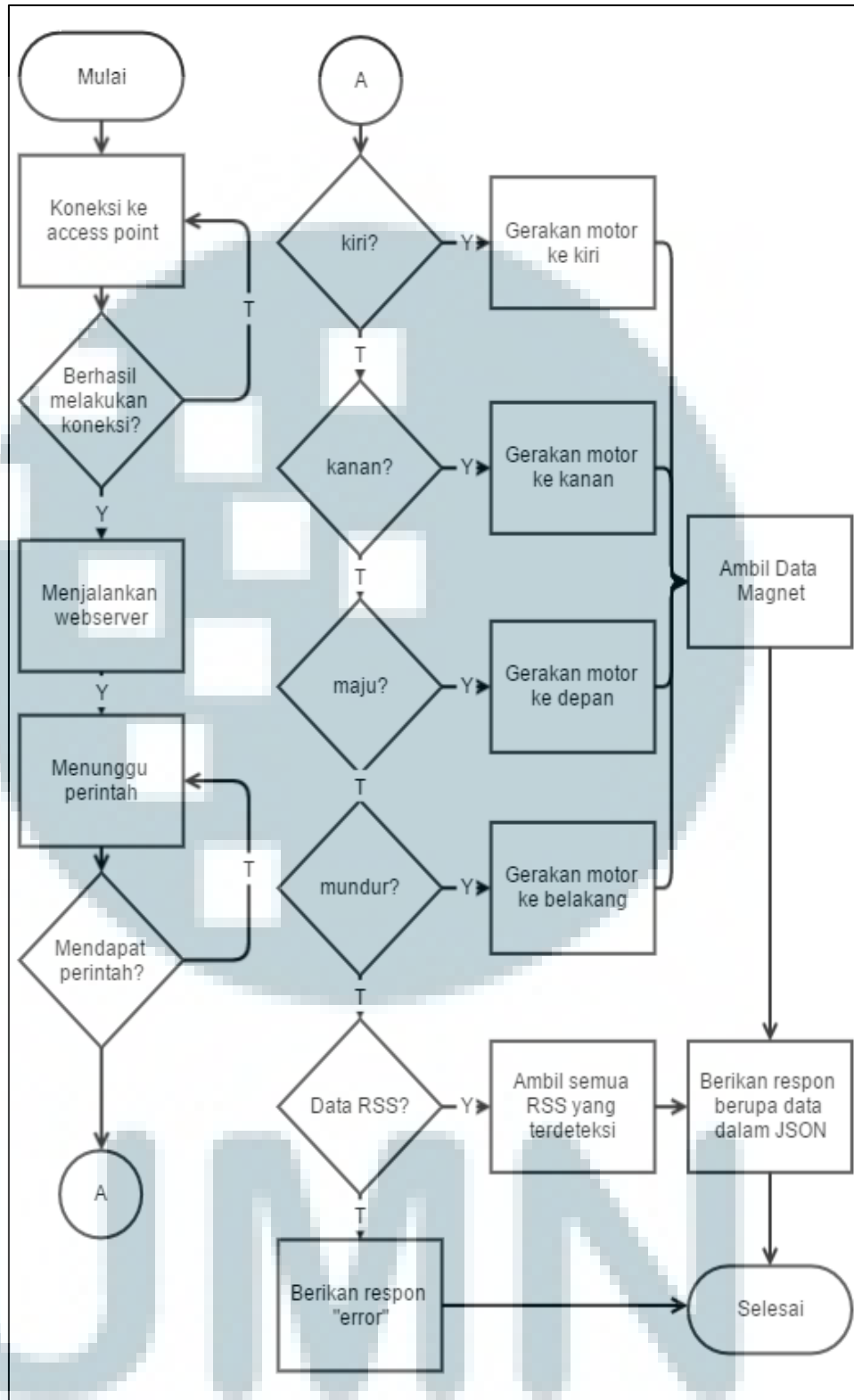
Software robot dibuat dengan menggunakan Arduino IDE. *Software* yang dibuat dibedakan menjadi 2 (dua) yaitu *software* untuk mengambil data dan analisa lokasi, serta *software* untuk uji coba.

Gambar 3.15 merupakan diagram alir pada *software* robot untuk mengambil data dan analisa lokasi. Robot akan terkoneksi ke *access point* dan menjalankan fungsi *webserver*. Apabila robot menerima perintah bergerak, maka robot akan menjalankan motor sesuai dengan arah yang diperintahkan. Robot akan mengirimkan data magnet sebanyak 10 (sepuluh) kali pengambilan data magnet dengan format JSON jika menerima perintah mengambil data magnet. Kondisi lainnya adalah pada saat robot menerima perintah mengirimkan data RSS. Pada saat robot menerima perintah tersebut, robot akan mengirimkan data RSS yang terbaca oleh robot dalam format JSON.

Gambar 3.16 menunjukan diagram alir untuk uji coba. Alir *software* uji coba memiliki beberapa perbedaan dengan *software* untuk menganalisa lokasi. Pada bagian *software* untuk uji coba, perintah yang diterima oleh robot hanya perintah untuk bergerak, tetapi data yang menjadi respon robot adalah data besaran magnet dan data RSS dalam bentuk JSON.



Gambar 3.15 Diagram Alir Program Robot Analisa Lokasi



Gambar 3.16 Diagram Alir Program Robot Uji Coba