

## **BAB III**

### **METODE PENELITIAN**

#### **3.1 Alat dan Bahan**

Agar penelitian berjalan dengan baik dan lancar maka diperlukan peralatan dan bahan yang digunakan pada penelitian baik untuk membuat *software* (tampilan GUI), maupun untuk membuat hardware (Robot Tank). Adapun peralatan serta bahan yang digunakan pada penelitian ini yaitu:

##### **3.1.1 Alat**

Alat yang digunakan untuk pembuatan serta pengujian sistem adalah sebagai berikut:

- A. Seperangkat Komputer
- B. *Software* Aplikasi *Processing* 3.3.6
- C. Multimeter Digital Merk SANWA seri CD800a
- D. *Battery Capacity Checker*
- E. *Toolset* (Obeng, Tang, dll)
- F. Solder
- G. Atraktor
- H. Gunting
- I. Pelarut PCB
- J. Kabel jumper
- K. PiloX
- L. Konektor kabel
- M. Lem G
- N. Lem bakar
- O. Amplas

### 3.1.2 Bahan

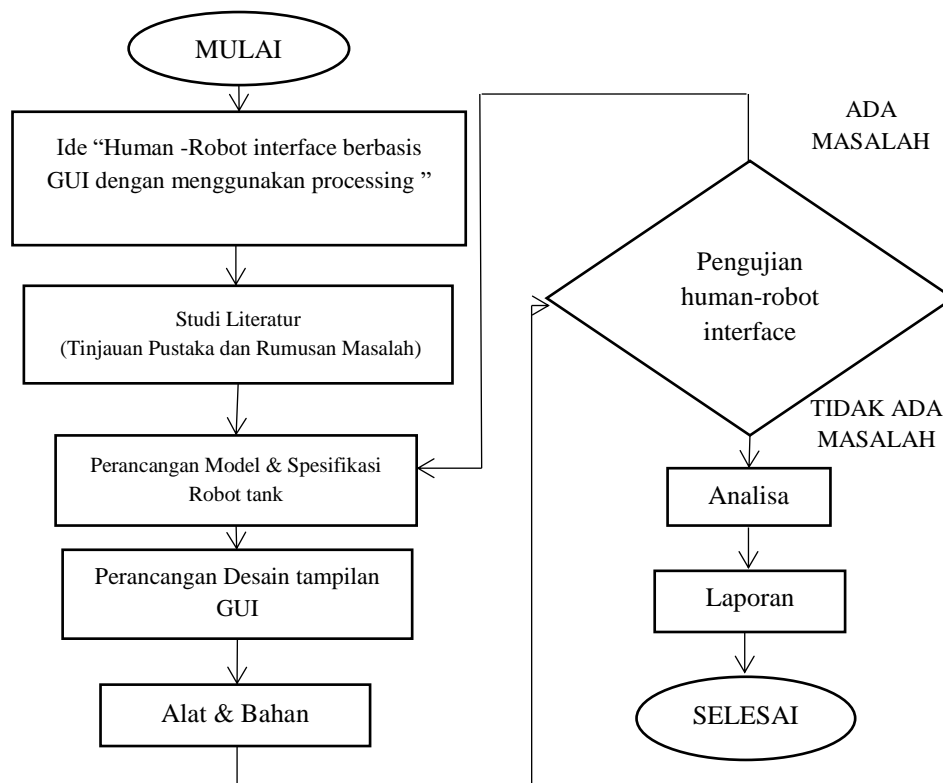
Bahan serta spesifikasi yang digunakan dalam pembuatan dan pengujian sistem pemanen energi adalah sebagai berikut:

- A. Baterai lippo dengan kapasitas penyimpanan 1450MAh, 3 sell.
- B. Arduino UNO R3 dimana arduino R3 menggunakan basic Atmega328, terdapat 14 pin I/O digital I/O dan 6 diantaranya dapat berfungsi sebagai output PWM, 6 input analog yang di lengkapi dengan ADC. Board arduino juga dilengkapi dengan pembangkit detak eksternal crystal keramik 16 MHz, memiliki koneksi USB, jack DC power, konektor ICSP, serta sebuah tombol reset.
- C. Telemetri 3DR, adalah modul untuk mengirim data menggunakan komunikasi *full-duplex*, artinya perangkat ini dapat berkomunikasi secara dua arah, antara *transmitter* dengan *receiver*
- D. Regulator tegangan tipe stepdown L-298.
- E. Traking wheel tank
- F. Motor DC
- G. Papan PCB viber
- H. 1 set gear box
- I. Tripleks
- J. Resin

### 3.2 Tahapan Penelitian

Metode interaksi manusia dengan robot (*human-robot interface*) dengan menggunakan tampilan visual pada komputer diharapkan mampu menjadi solusi terbaru yang dapat memudahkan manusia untuk melakukan pengontrolan terhadap kinerja sebuah robot. Metode interaksi berbasis GUI lebih fleksibel bila dibandingkan dengan metode interaksi yang telah ada sebelumnya baik menggunakan saklar, maupun tombol-tombol. karena metode interaksi manusia

dengan robot berbasis GUI dapat dilakukan pengembangan terhadap sistem kontrolnya, sistem GUI memungkinkan *user* untuk lebih mudah menambahkan berbagai tombol dalam tampilan, menambahkan indicator untuk memonitor performa dari robot dan sebagainya. Untuk mencapai tujuan penelitian “Interaksi Robot Dengan Manusia Berbasis Gui (*Graphical User Interface*) Menggunakan *Software Processing*“ maka disusunlah tahapan-tahapan dalam penelitian yang dapat digambarkan melalui diagram alir sebagai berikut:



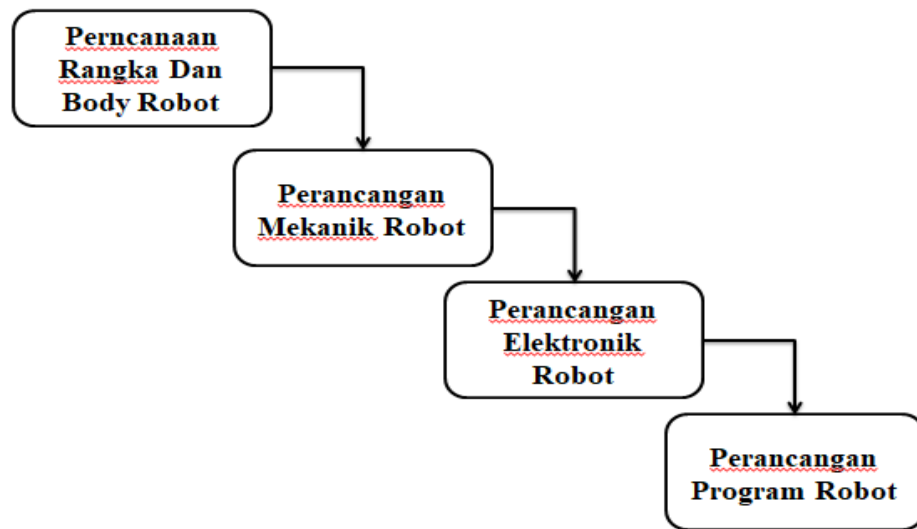
**Gambar 3.1** Flowchat Tahapan Penelitian

### 3.3 Studi Literatur

Studi literatur dilakukan dengan tujuan mendapatkan wawasan umum yang berhubungan dengan sistem yang akan dibuat, landasan teori yang digunakan dan dapat mengetahui berbagai penelitian serupa yang telah dilakukan sebelumnya. Studi literatur ini juga digunakan untuk mempelajari prosedur perancangan yang paling tepat selain itu dapat memperbaiki, dan mengembangkan kelemahan pada penelitian yang sudah ada sebelumnya.

Adapun sumber literatur yang digunakan antara lain adalah; buku, jurnal, internet, dan tugas akhir.

### 3.4 Perancangan Model dan Spesifikasi Robot Tank



Gambar 3.2 Diagram Block Proses Perancangan Robot

Dari gambar 3.2 blok digram diatas dapat diamati bahwa dalam membangun robot tank terdapat empat tahapan yang harus dilalui. Tahapan-tahapan tersebut dimulai dari perancangan rangka dan body dari robot, perancangan sistem mekanik, perancangan sistem elektronik, dan perancangan program (*software*). Adapun tujuan dari beberapa tahapan di atas adalah untuk mendapatkan spesifikasi-spesifikasi yang cocok untuk selanjutnya diterapkan pada proses pembuatan robot. Dengan demikian robot tank yang akan dibangun dapat beroperasi sesuai dengan fungsi dan kegunaannya.

#### 3.4.1 Perancangan Rangka Dan Body Robot

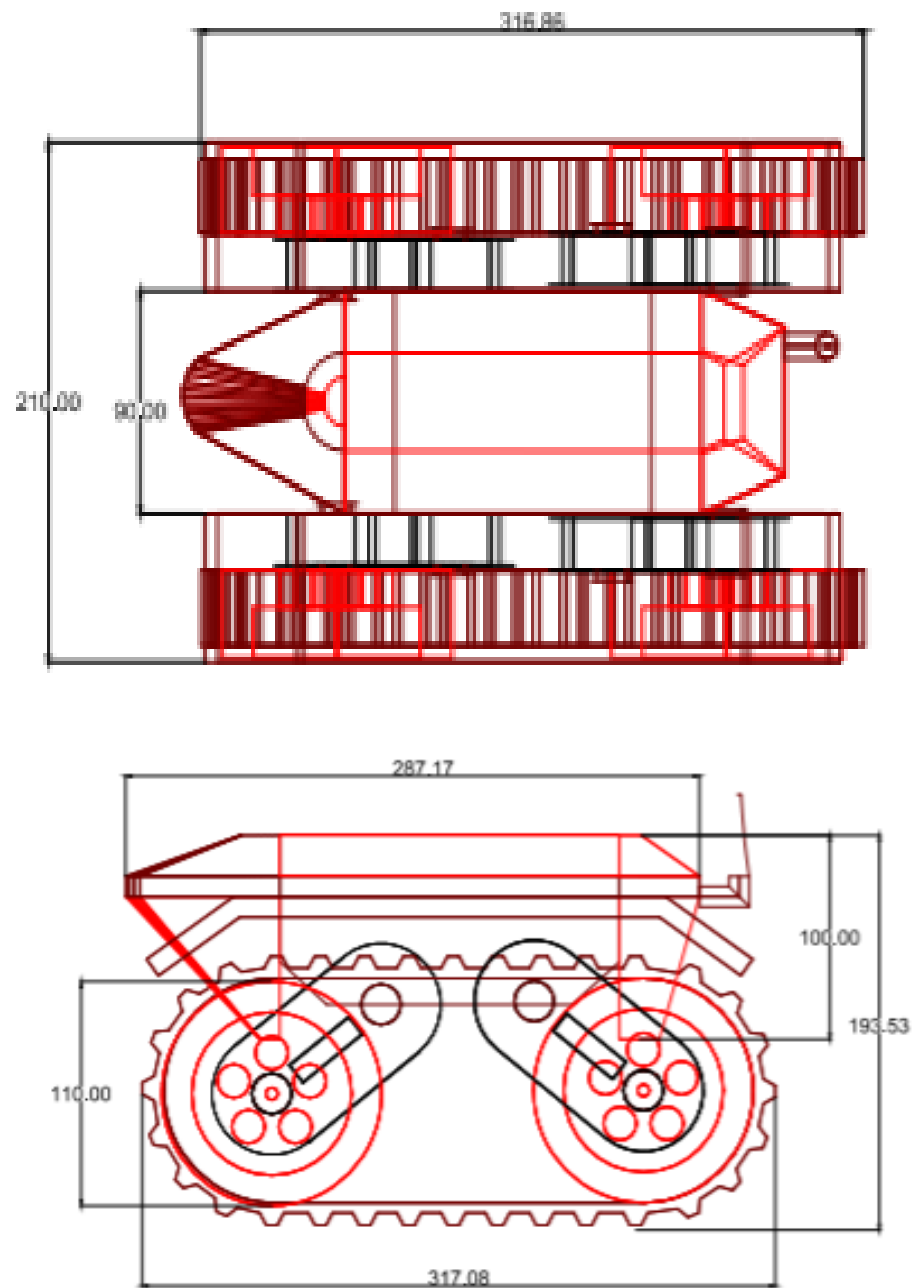
Pada tahapan perancangan rangka dan body robot meliputi perancangan awal, gambaran bentuk fisik robot, dan material yang akan digunakan. Dalam penelitian ini robot yang digunakan sebagai

basic struktur adalah robot Rover 5. Berikut merupakan tampilan robot rover 5 yang akan dimodifikasi menjadi robot tank.

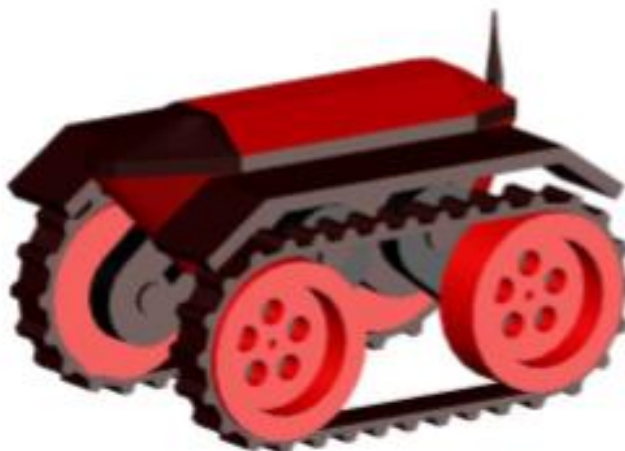


**Gambar 3.3** Robot Rover 5

Selanjutnya robot rover ini akan dimodifikasi di beberapa bagian sehingga robot dapat beroperasi sebagaimana mestinya. Modifikasi yang dilakukan pada Rover 5 ini meliputi modifikasi pada bagian body sehingga robot menjadi lebih menarik, selain itu body juga berfungsi sebagai pelindung komponen elektronika yang ditempatkan di dalam robot tank. Beberapa tahapan yang akan dilakukan dalam proses ini adalah mendesain bentuk fisik yang akan dibuat baik secara 2 dimensi maupun 3 dimensi lengkap dengan ukuran dari robot. Selanjutnya menentukan material yang akan digunakan untuk membuat body agar robot mampu menopang struktur fisiknya dan mendukung robot untuk melakukan beberapa manuver gerakan saat beroperasi. Berikut merupakan gambar rancangan body robot tank yang akan dibuat.



**Gambar 3.4** Rancangan 2 Dimensi Robot Tank



**Gambar 3.5** Rancangan Rangka dan Body Robot 3Dimensi

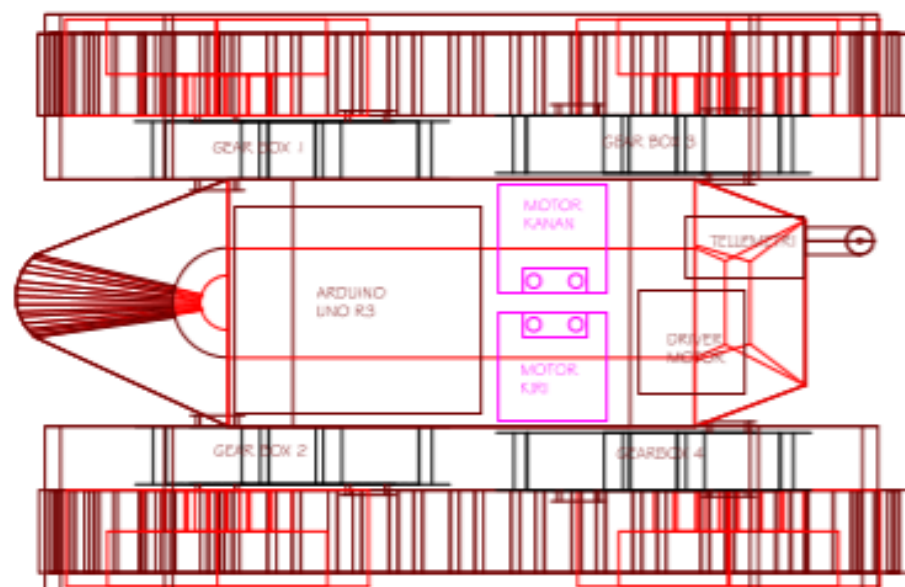
Bahan yang digunakan untuk membentuk struktur robot tank terdiri dari beberapa macam diantaranya adalah plastik, aluminium, karet, akrilik, PCB dan *polywood*. Bahan plastik digunakan untuk membuat bagian bawah dari robot tank. Bahan ini dipilih karena bahan ini memiliki karakteristik yang kuat namun ringan, selain itu plastik mudah dibentuk sehingga bahan ini sangat cocok untuk dipilih sebagai bahan dasar pembuatan struktur. Pada bagian roda *tracking wheels* menggunakan bahan karet karena bahan ini elastis dan ulet kelebihan lain yaitu karet dapat melekat pada permukaan sehingga roda tidak mengalami *slip* saat berputar. Untuk membentuk bagian *body* atas digunakan *polywood* yang dilapis dengan resin. Bahan ini ringan kuat dan sangat mudah dibentuk sehingga body dapat dibuat dengan bentuk sesuai keinginan. Untuk menguatkan strukturnya maka dipakai aluminium yang ditempatkan pada bagian-bagian tertentu sehingga struktur mampu menumpu beban total dari robot.

Robot didesain dengan mengutamakan aspek fungsinya, dimana robot harus mampu berjalan dalam berbagai medan tanpa mengalami hambatan, selain itu robot dapat dikendalikan secara jauh dan melakukan berbagai macam manuver baik maju, mundur, belok kanan,

maupun belok kiri. Sehingga rangka dan *body* robot harus seimbang dan kokoh. Hal ini dimaksudkan agar robot tahan terhadap benturan, selain itu robot harus memiliki struktur yang bagus sehingga memudahkan dalam pemasangan sistem mekanik, dan rangkaian elektronika.

### 3.4.2 Perancangan Mekanik Robot

Tahapan perancangan mekanik adalah tahap untuk menentukan sistem penggerak dari robot. Dimana dalam tahap ini ditentukan konstruksi mekanik yang memungkinkan agar robot dapat melakukan berbagai gerakan. Dalam tahap perancangan mekanik ini juga dirancang *actuator* penggerak yang memungkinkan robot untuk dapat bergerak dari satu tempat ketempat lain. Gerakan manuver yang dapat dilakukan oleh robot meliputi bergerak maju, mundur, belok kanan, dan belok kiri. Semua gerakan pada robot saat beroperasi didesain dan dibuat agar robot dapat bekerja sesuai kebutuhan yang diinginkan.



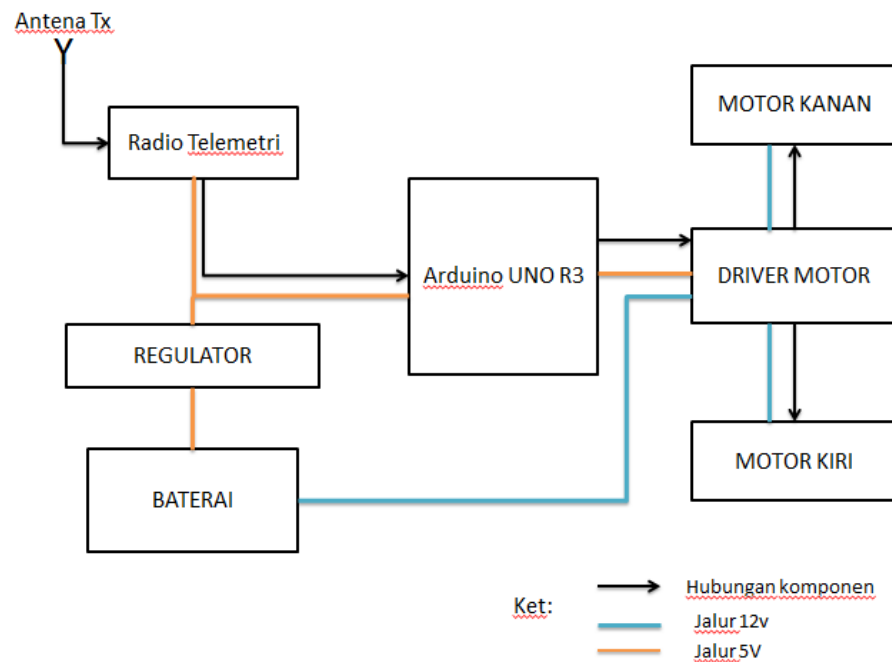
**Gambar 3.6** Perancangan Mekanik Robot



Nantinya robot yang akan digunakan adalah mobile robot dengan type tank. Robot ini menggunakan roda *tracking wheels* sehingga robot dapat berjalan dalam berbagai medan tanpa ada hambatan. Selain itu robot dibekali dengan *actuator* 2 buah motor DC yang ditempatkan pada bagian kanan dan kiri. Motor DC ini digunakan untuk bernavigasi. Motor DC ini akan menggerakkan setiap *tracking wheels* robot, sehingga memungkinkan robot untuk bergerak maju, mundur, belok kanan, maupun belok kiri. Masing-masing *actuator* memiliki fungsi dan tugas yang berbeda dalam melakukan setiap gerakan. Motor kanan digunakan untuk menggerakkan *tracking wheels* sebelah kanan baik berputar CW maupun CCW. Demikian pula motor bagian kiri digunakan untuk menggerakkan *tracking wheels* sebelah kiri baik CW maupun CWW. Untuk mendukung mobilitas robot, maka robot didesain dengan *Ground clearance* yang tinggi, sehingga robot tidak mengalami hambatan saat berjalan pada medan yang sulit.

### **3.4.3 Perancangan Elektronik Robot**

Pada tahap perancangan elektronik meliputi pembuatan rangkaian elektronik yang bertujuan sebagai sistem pengontrol dari setiap gerakan robot. Selain itu rangkaian elektronik juga berperan sebagai *main board* yaitu sebagai pusat pengendali robot saat dioperasikan. Dalam tahap ini akan dirancang konfigurasi hubungan antar komponen, konfigurasi antar komponen dapat diamati dalam block diagram sistem berikut:



**Gambar 3.7** Block Diagram Robot

Berdasarkan block diagram 3.7 diatas dapat diamati hubungan konfigurasi sistem pada robot tank. Berikut merupakan penjelasan prinsip kerja dari masing-masing block sistem elektronik robot tank yaitu:

- A. Baterai, sumber tegangan baterai yang digunakan robot tank menggunakan baterai jenis LiPo dengan kapasitas 1450mAh 3 cell. Baterai ini memiliki tegangan kerja 12 V yang digunakan untuk mencatu daya pada keseluruhan sistem robot tank. Untuk bagian motor membutuhkan tegangan 12 V yang diperoleh langsung dari batterai. Sedangkan untuk bagian mikrokontroler digunakan tegangan 5 V yang didapatkan dari regulator penurun tegangan 12 V diturunkan menjadi 5 V.
- B. Rangkain *regulator*, rangkaian regulator berfungsi sebagai rangkaian pembagi tegangan pada robot, hal ini dikarenakan kebutuhan tegangan tiap *sub* sistem pada robot berbeda-beda. Selain itu rangkaian *regulator* juga berfungsi sebagai penstabil tegangan yang akan digunakan untuk mensuplay sistem. Rangkain *regulator* yang digunakan dalam robot tank ini

merupakan *regulator* jenis *stepdown* (penurun tegangan) dimana regulator akan menurunkan tegangan sebesar 12 V dari baterai diturunkan menjadi 5 V.

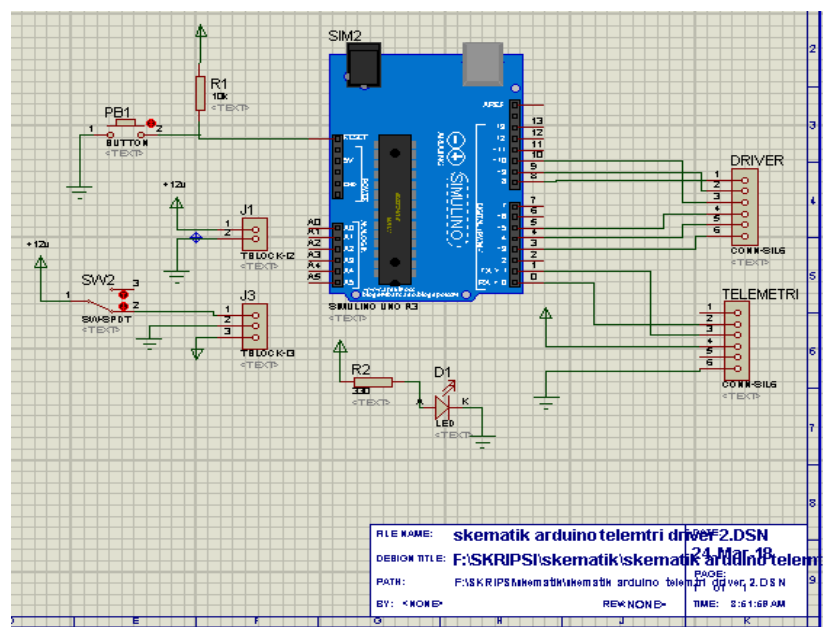
- C. Radio telemetri, rangkaian radio telemetri berfungsi untuk menerima data GUI yang dikirimkan dari komputer kepada robot tank. Data pengontrolan robot ini dikirimkan melalui media *wireless* sehingga robot dapat dikendalikan secara jarak jauh. Rangkaian radio telemetri yang ini dapat melakukan komunikasi dengan perangkat komputer melalui komunikasi serial.
- D. Arduino uno R3, module arduino uno R3 digunakan sebagai mikrokontroler pengendali utama robot tank, melalui mikrokontroler ini data akan diproses melalui algoritma tertentu, selanjutnya data hasil pemrosesan ini digunakan untuk memberikan perintah kepada *driver* untuk menggerakkan setiap motor sesuai dengan perintah dari *user*. Sehingga dapat dikatakan bahwa arduino Uno R3 ini berfungsi sebagai otak pengontrol dari robot.
- E. *Driver* motor, rangkain *driver* motor yang digunakan adalah modul *driver* motor L-298 dimana dalam rangakaian ini dibekali 2 chanel output, sehingga rangkaian ini memungkinkan untuk mengendalikan 2 buah motor sekaligus. Rangkaian *driver* motor L-298 ini menggunakan rangkaian *H-bridge* yang telah terintegrasi dalam sebuah chip. Dengan demikian maka rangakaian ini mampu mengendalikan arah putaran motor baik CW (*clockwise*) maupun CCW (*clockcounterwise*).
- F. Motor DC, motor DC digunakan sebagai *actuator* penggerak dari robot tank. Motor DC dapat mengubah energy listrik menjadi energy mekanik berupa putaran. Energy mekanik putaran ini yang selanjutnya dimanfaatkan untuk menggerakkan roda *tracking wheels*. Dalam robot terdapat dua buah motor ditempatkan pada

bagian kiri dan kanan dimana masing-masing motor DC akan menggerakkan sebuah roda *tracking wheels*.

### 3.4.4 Perancangan Rangkaian Skematik

Berdasarkan dari blok diagram sistem 3.7 diatas dapat dirancang sebuah rangkaian skematik. Pembuatan skematik rangkaian dimaksudkan untuk menghubungkan setiap komponen yang akan dipasang pada robot sehingga robot dapat bekerja sebagaimana mestinya. Dalam merancang rangkain skematik digunakan *software* pendukung yaitu Proteus Professional. Berikut merupakan gambar skematik rangkaian yang digunakan pada robot tank.

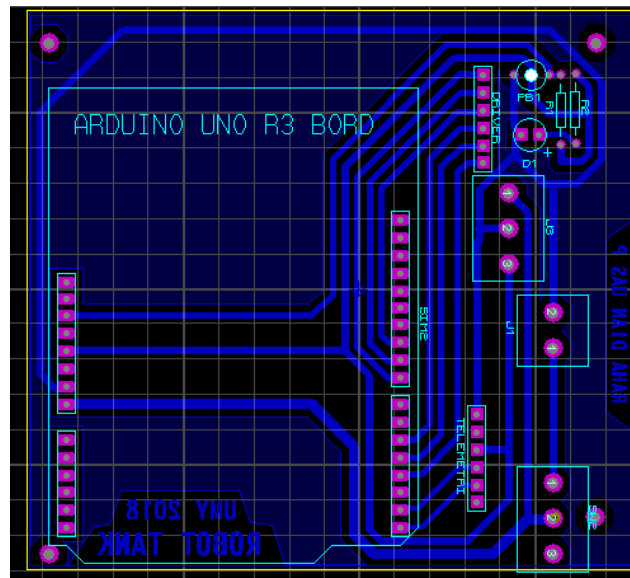
#### A. Skematik Sistem Minimum



Gambar 3.8 Rangkaian Skematik Sistem Minimum

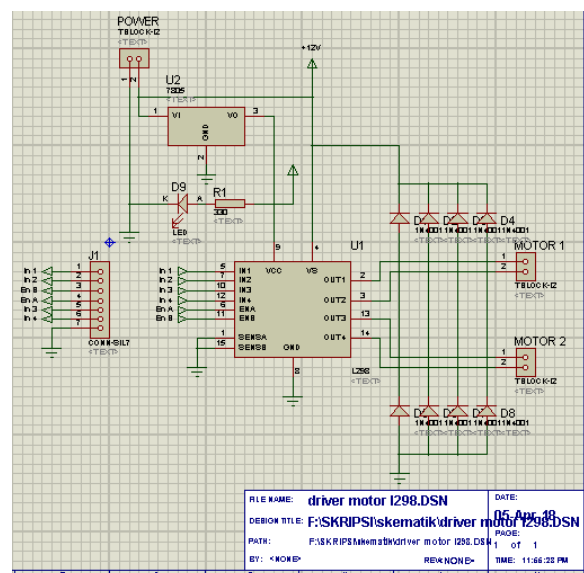
Rangkaian pada gambar 3.8 di atas merupakan rangkaian minimum arduino UNO R3. Dalam gambar 3.8 diatas dapat diamati konfigurasi hubungan setiap pin pada arduino Uno dan hubungannya dengan komponen lain seperti module radio telemetri, ataupun *driver* motor. Selain itu dalam rangkaian diatas juga terdapat rangkaian tambahan berupa penambahan saklar untuk menyalakan dan mematikan sistem, juga dilengkapi dengan

LED sebagai indicator bahwa rangkaian tersebut dalam keadaan aktif. Selanjutnya untuk merealisasikan rangkaian diatas maka didesain pula rangkaian dalam layout PCB yang nantinya akan dicetak, berikut merupakan hasil desain PCB dari skematik di atas.



Gambar 3.9 Desain Layout PCB

## B. Rangkaian *Driver Motor* L-298

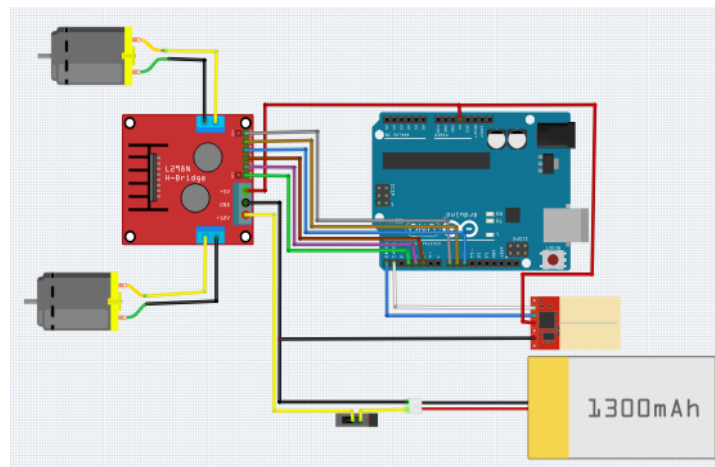













Gambar 3.10 Rangkaian Driver Motor L-298

Rangkaian pada gambar 3.10 diatas merupakan rangkaian yang digunakan untuk melakukan driver terhadap motor. Rangkaian diatas menggunakan IC L-298 dimana dalam IC tersebut telah terintegrasi dengan rangkaian *H-bridge* yang biasa digunakan untuk mengendalikan motor DC, pengendalian yang dilakukan merupakan pengendalian arah putaran motor dan juga kecepatan putar motor. Rangkaian *driver* L-298 ini nantinya digunakan sebagai rangkaian penggerak utama pada saat robot bernavigasi. Rangkaian *driver* ini mempunyai 2 output yang langsung diumpankan pada 2 buah motor untuk mengendalikan gerakan robot baik gerak maju, mundur, belok kanan, maupun belok kiri. Dalam rangkaian *driver* ini menggunakan 2 jenis tegangan yang berbeda yaitu 12 V dan 5 V. Dalam rangkain *driver* ini dilengkapi dengan IC *regulator* 7805 yang dapat menurunkan tegangan 12 V menjadi 5 V. Tegangan 12 V digunakan untuk menggerakkan output motor baik kiri maupun kanan. Sedangkan tegangan 5 V digunakan untuk mengaktifkan IC *driver* L-298.

### 3.4.5 Rancangan Seluruh Sistem Robot Tank

Arduino digunakan sebagai otak pengontrol robot tank dalam melakukan setiap gerakan. Arduino juga akan menerima data yang dikirimkn dari komputer dan akan diproses sedemikian rupa untuk mengendalikan gerakan robot tank, baik bergerak maju, kanan, kiri, maupun mundur. Pada saat beroperasi arduino berperan sebagai penerima (receiver), dari data yang diterima arduino akan melakukan pengontrolan terhadap input dan output sesuai dengan skenario yang telah ditentukan. Berikut merupakan skematik hubungan input dan output arduino dengan komponen-komponen lainnya:



	: jalur GND
	: +12 V
	: + 5V
	: jalur Rx Arduino
	: Jalur Tx Arduino
	: D3 Arduino - EnA
	: D4 Arduino - In1
	: D5 Arduino - In2
	: D10 Arduino - In3
	: D9 Arduino - In4
	: D8 Arduino - EnB

**Gambar 3.11** Hubungan Pin Arduino Dengan Komponen Lain

Dari gambar 3.11 di atas menjelaskan dalam membuat robot tank dibutuhkan beberapa komponen yaitu 2 buah motor DC, 1 buah driver L298N, 1 buah receiver radio telemetri 3DR, dan Arduino Uno. Motor DC dihubungkan dengan output diver L298N yaitu PIN OUT1 dan OUT 2 digunakan untuk motor sebelah kanan, sedangkan PIN OUT 3 dan 4 dihubungkan dengan motor sebelah kiri. Untuk mengendalikan putaran motor driver L298N menggunakan dua buah *H-bridge* yang memiliki enam buah pin input yaitu EnA, In 1, In 2, In 3, In 4, dan EN B. Pin EnA, In1, dan In2 digunakan untuk mengendalikan motor sebelah kanan, sedangkan Pin EnA, In3, dan In4 digunakan untuk mengontrol motor sebelah kiri. PIN ENA dan ENB merupakan PIN inputan PWM yang akan digunakan untuk mengendalikan kecepatan

putar motor. Koneksi antar PIN Arduino dengan driver motor L298N adalah sebagai berikut:

**Tabel 3.1** Konfigurasi Pin Arduino Dengan Driver Motor

<b>PIN Arduino</b>	<b>PIN Driver Motor</b>
PIN digital 3	EnA
PIN digital 4	In 1
PIN digital 5	In2
PIN digital 8	EnB
PIN digital 9	In3
PIN digital 10	In4

Selain itu driver motor L298N memiliki tiga PIN power yang terdiri dari +12, GND, dan +5V. PIN +12V digunakan sebagai inputan tegangan 12V yang akan mensuplay dua buah motor DC. Sedangkan PIN +5 V digunakan sebagai inputan tegangan 5V yang selanjutnya akan digunakan untuk mensuplay rangkaian IC L298N.

Selain hubungan Arduino dengan Motor driver, dari gambar 3.11 tersebut juga dapat diamati hubungan PIN dari arduino dengan modul radio telemetri 3DR. Modul radio telemetri 3DR memiliki 6 buah Pin yang terdiri dari EN, Rx, Tx, VCC, CF, dan GND. Dalam konfigurasinya tidak semua pin dihubungkan dengan arduino. hanya empat Pin saja yang dihubungkan dengan arduino yaitu Rx, Tx, VCC, dan GND untuk konfigurasi hubungan PIN dapat disajikan dalam tabel berikut:



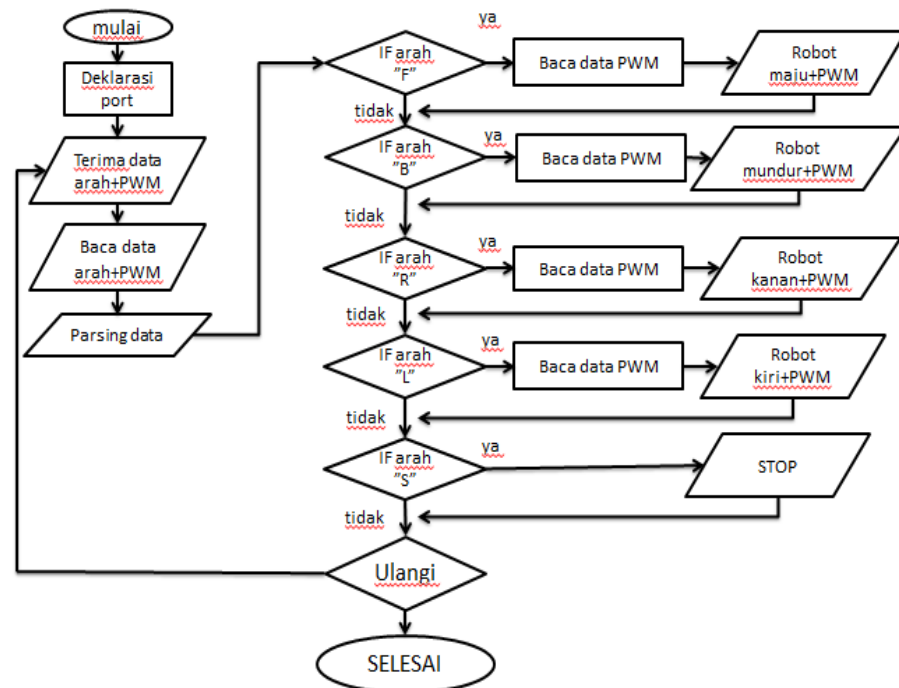
**Tabel 3.2** Konfigurasi PIN Arduino Dengan Radio Telemetri 3DR

<b>PIN Arduino</b>	<b>PIN Modul Radio Telemetri 3DR</b>
PIN Rx ARduino	PIN Tx
PIN TX Arduino	PIN Rx
+5V	VCC
GND	GND

### 3.4.6 Perancangan Program Robot Tank

Tahap perancangan program dimaksudkan untuk menentukan algoritma-algoritma pengontrolan yang dapat memberikan perintah dan mengendalikan rangkaian yang terhubung dengan mikrokontroler. Dengan algoritma-algoritma ini robot mendapat acuan untuk melakukan berbagai macam gerakan berdasarkan dari data-data yang diterima oleh mikrokontroler. Program dibangun dengan menggunakan bantuan *software* Arduino IDE.

Untuk mempermudah pembangunan program yang akan dimasukan kedalam robot tank, sebelumnya dibuatlah diagram alir (*flowcart*) program sebagai rancangan awal. Dalam pembuatan *flowchat* akan dirancang berbagai kondisi yang harus dipenuhi sebelum robot malakukan aksi atau pergerakan. Selain itu *flowchat* juga menentukan gambaran umum prinsip kerja program. Dalam robot tank ini alur *flowchat* dapat diamati dalam gambar berikut:



Gambar 3.12 Flowchart Robot Tank

## A. Sketsa program Pada Arduino

### 1. Pengaturan `void setup()`

Langkah pertama dalam membangun program arduino adalah dengan pengaturan `void setup()`, dalam void setup ini berfungsi untuk mendeklarasikan PIN input output apa saja yang digunakan oleh arduino, melakukan pengaturan komunikasi serial, selain itu dalam Void setup ini dapat digunakan untuk menuliskan program. Hanya saja dalam fungsi void setup program hanya akan tereksekusi satu kali, yaitu pada saat dinyalakan (*PowerUp*) dan pada saat melakukan reset.

Beberapa pengaturan yang akan dituliskan dalam fungsi `void setup()` ini adalah sebagai berikut:

**Tabel 3.3** Perintah Pengaturan Void Setup()

<b>Sketsa Program Arduino</b>	<b>Fungsi</b>
<code>Serial.begin(57600);</code>	Mendeklarasikan bahwa akan menggunakan komunikasi serial dengan baudrate sebesar 57600
<code>pinMode(3,OUTPUT);</code>	Mengatur PIN D3 sebagai OUTPUT terhubung ke EnA
<code>pinMode(4,OUTPUT);</code>	Mengatur PIN D4 sebagai OUTPUT terhubung ke In1
<code>pinMode(5,OUTPUT);</code>	Mengatur PIN D5 sebagai OUTPUT terhubung ke In2
<code>pinMode(8,OUTPUT);</code>	Mengatur PIN D8 sebagai OUTPUT terhubung ke EnB
<code>pinMode(9,OUTPUT);</code>	Mengatur PIN D9 sebagai OUTPUT terhubung ke In3
<code>pinMode(10,OUTPUT);</code>	Mengatur PIN D10 sebagai OUTPUT terhubung ke In4

Berdasarkan dari konfigurasi diatas dapat diamati bahwa PIN digital 3 dan 8 terhubung ke En A dan EnB, sehingga PIN ini yang nantinya akan mngatur besarnya PWM yang diberikan pada motor, pada PIN digital 4, dan5 akan mengirimkan logika 0 dan 1 untuk mengendalikan arah putaran motor sebelah kanan. Sedangkan pada PIN 9 dan 10 juga mengirimkan logika 0 dan 1 yang akan mengendalikan arah putaran motor sebelah kiri.

## 2. Perintah `void parsingData();`

Parsing data merupakan perintah yang digunakan untuk melakukan pemecahan data pada komunikasi serial, hal ini dibuat karena pada saat GUI dioperasikan setiap kali *user* menekan tombol data yang dikirm adalah jenis yaitu data arah (navigasi) ditambah dengan data nilai Slider yang digunakan untuk menentukan kecepatan putar motor. Dengan demikian butuh sebuah rutin program yang dapat digunakan untuk memecah 2 data yang berbeda ini.

`Void parsingData();` ini nantinya akan dipanggil pada void loop (), sehingga, akan dijalankan terus menerus selama rangkaian mendapatkan suplay tegangan. Arduino akan terus berada dalam posisi *standby* untuk selalu menerima data yang dikirmkan dari komputer. Setelah itu apabila ada data masuk maka data tersebut akan ditampung terlebih dahulu dalam sebuah variable. Variable tersebut akan dicek terus menerus, apabila jumlah karakter yang dikirimkan lebih dari 2 maka proses parsing data mulai dilakukan, bila data yang diterima kurang dari 2 maka data tersebut akan ditampung terlebih dahulu. Dalam arduino IDE perintah ini ditulis dengan “`if (dataIn.length())>2;` ”. Apabila data yang dikirim lebih dari 2 maka menandakan proses parsing data dimulai. Proses parsing data dimulai dengan pemanggilan fungsi `parsingData()`.

```

void loop() {
while (Serial.available())
{
char c = Serial.read();
dataIn += c;
}
if (dataIn.length()>2)
{

parsingData();
Serial.print(dt[0]+' '+dt[1]);
PWM=dt[1].toInt();
}
}

```

Setelah parsing data dipanggil maka data akan terpisah menjadi 2 macam. Dalam sketsa program diatas ditulis dt[0] dan dt[1]. Variable dt[0] digunakan sebagai tampungan dari data karakter yang dikirimkan, selanjutnya data ini akan digunakan untuk menentukan arah navigasi dari robot. Sedangkan untuk dt[1] merupakan variable dari nilai kecepatan yang selanjutnya data ini digunakan untuk mengontrol besarnya PWM yang diberikan kepada motor.

Adapun Proses parsing data terjadi pada fungsi parsingData(). Proses parsing data berjalan dengan megecek setiap karakter yang dikirimkan melalui port serial. Data yang dikirimkan dibandingkan dengan tanda “,” sebagai pemisah antar data Pengecekan dimulai dari index 1) variabel dataIn, hal ini dikarenakan index 0 merupakan *header* dalam paket data.

Data yang dikirimkan dari komputer akan melalui proses pengecekan dan ditampung dalam variable bertipe string. Selanjutnya data ini akan kelompokkan dalam bentuk array ditampung pada variable bernama dt. Apabila dalam proses pengecekan karakter terdapat tanda pemisah koma (,), maka secara otomatis variable penampung data akan berpindah dengan cara increment index array atau penambahan dari index yang dikirimkan. Proses ini akan terus berulang hingga kondisi syarat parsing data tidak terpenuhi lagi.

Setelah proses parsing selesai maka data yang diterima menjadi 2 data yang digunakan untuk melakukan pengendalian pada 2 aspek yang berbeda. Berikut merupakan skema program void parsing data yang diterapkan dalam arduino.

```

void parsingData(){
int j=0;
dt[j]="";
//proses parsing data
for(int i=0;i<dataIn.length();i++)
{
//pengecekan tiap karakter dengan karakter (#) dan (,)
if ( (dataIn[i] == ',')
{
//increment variabel j, digunakan untuk merubah index array pen
j++;
dt[j]=""; //inisialisasi variabel array dt[j]
}
else
{
//proses tampung data saat pengecekan karakter selesai.
dt[j] = dt[j] + dataIn[i];
}
}
}
}

```

### 3. Perintah Mengatur Arah Putar Motor

Seperti yang telah dijelaskan diatas data yang dikirimkan dari komputer adalah 2 data yang dikirim sekaligus, selanjutnya 2 data ini akan dipisahkan oleh arduino menjadi dt[0] dan dt[1]. Variable dt[0] berisikan karakter tertentu yang digunakan untuk mengendalikan gerakan robot, maju, mundur, belok kanan, dan belok kiri.

Dalam perancangan GUI akan dibuat pada saat tombol ditekan maka komputer akan mengirimkan karakter tertentu kepada arduino. Pada saat tombol maju ditekan maka komputer akan mengirimkan data string ("F"+ nilai PWM). Pada saat tombol mundur ditekan

maka akan mengirimkan (“B”+ nilai PWM). Saat tombol kanan ditekan maka akan mengirim (“R”+ nilai PWM). Saat tombol kiri ditekan maka akan mengirim (“L”+ nilai PWM). Pada saat tombol alloff ditekan maka komputer akan mengirimkan (“S”+ 0).

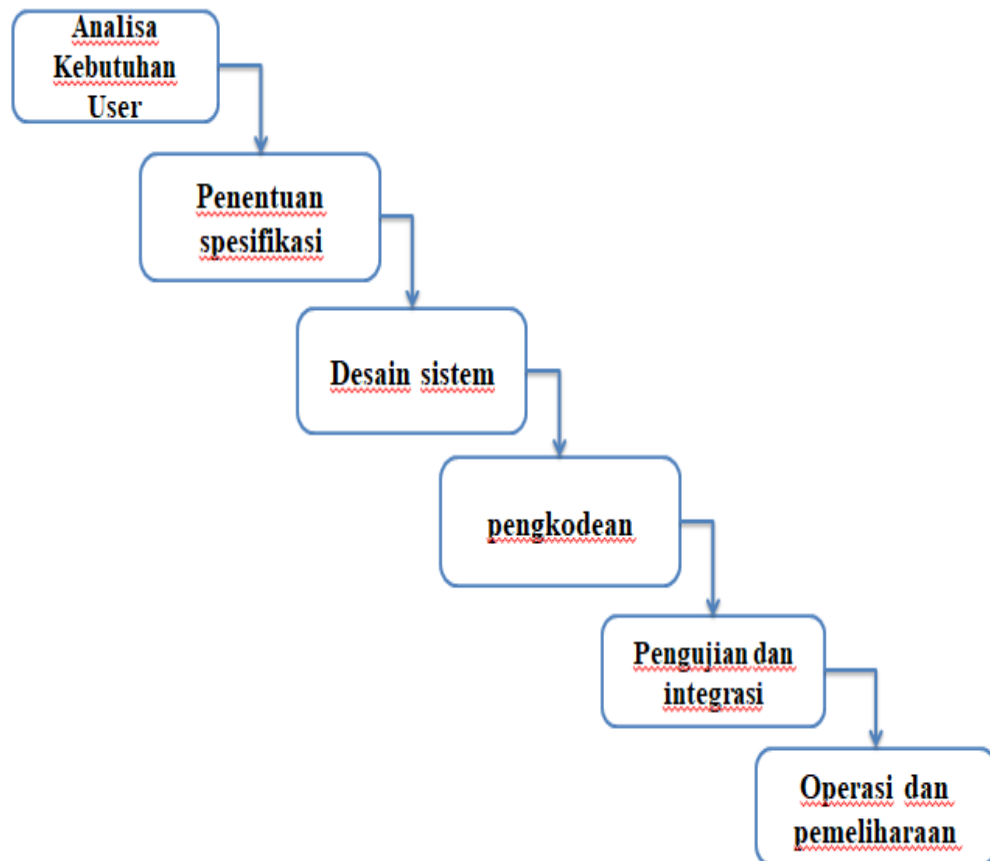
Data karakter yang dikirim ini akan diterima oleh arduino dan ditampung dalam variabel `dt[0]`. Selanjutnya arduino akan memerintahkan kedua buah motor untuk berputar maju (CW) saat `dt[0] = “F”`, pada saat `dt[0] = “B”` maka arduino akan menggerakkan motor untuk berputar mundur (CCW), demikian juga saat `dt[0] = “R”` maka arduino akan memberikan perintah sehingga motor kanan akan berputar ke belakang (CCW) dan motor kiri berputar ke depan (CW), dan pada saat `dt[0] = “L”` maka motor kanan akan berputar ke depan (CW) dan motor kiri akan berputar ke belakang (CCW).

Dengan demikian memungkinkan robot tank dapat bergerak maju saat GUI ditekan maju, bergerak mundur saat GUI ditekan mundur, berbelok ke kanan saat tombol GUI ditekan kanan, dan berbelok ke kiri saat GUI ditekan kiri. Kondisi selanjutnya akan dituliskan dalam bahasa pemrograman Arduino IDE sehingga arduino dapat mengontrol robot sesuai dengan skenario yang telah ditentukan.

### 3.5 Gambaran Umum Proses Perancangan tampilan GUI

Perencanaan pembangunan tampilan GUI menggunakan metode pendekatan *waterfall*. Metode *waterfall* adalah metode pendekatan yang menggambarkan pendekatan secara sistematis dan tahapan pengembangan perangkat lunak dilakukan secara berurutan. Dengan demikian metode pendekatan *waterfall* ini dapat membantu mengatasi kerumitan yang terjadi akibat pengembangan perangkat lunak. Metode *waterfall* memungkinkan pemecahan misi pengembangan yang rumit menjadi beberapa langkah yang sistematis yaitu dimulai dari analisa kebutuhan, dilanjutkan dengan spesifikasi, desain, kode, unit test, pengujian, dan terakhir adalah

pemeliharaan. Dengan melalui beberapa tahap di atas, maka akan menjadi produk akhir yang siap untuk digunakan. Demi sempurnanya sebuah perangkat lunak maka membutuhkan validasi dan masukan guna membangun dan menyempurnakan perangkat lunak. Berikut merupakan diagram metode pendekatan *waterfall*.



Gambar 3.13 Block Diagram Metode Waterfall

### 3.5.1 Analisa Kebutuhan *User*

Tahap awal yang dilakukan dalam pendekatan *waterfall* adalah melakukan analisa terhadap kebutuhan *user*. Tahap ini bertujuan untuk menyesuaikan kebutuhan apa saja kebutuhan *user*, sehingga perangkat lunak yang akan dibangun cocok dengan harapan dari *user*. Selain itu dengan melakukan analisa kebutuhan *user*, dapat memberikan indikasi batasan dari perangkat lunak tersebut.



Selanjutnya informasi-informasi kebutuhan dari *user* dapat diperoleh dari beberapa sumber diantaranya, wawancara, diskusi, ataupun dilakukan dengan melakukan survey secara langsung mengenai informasi untuk mendapatkan data yang dibutuhkan.

Untuk menentukan hambatan-hambatan yang terjadi, dan kebutuhan-kebutuhan yang diharapkan, serta usulan perbaikan-perbaikan. Pada tahap ini akan dilakukan analisa sistem antara lain :

#### A. Analisa Identifikasi Masalah

Identifikasi masalah merupakan langkah awal dari analisis sistem. Dalam tahap ini akan dilakukan analisa dari masalah-masalah yang timbul. Sehingga aplikasi yang nantinya diciptakan akan hadir membawa solusi untuk memecahkan permasalahan tersebut.

#### B. Analisa Kebutuhan

Tahapan analisa kebutuhan dilakukan untuk menganalisis kebutuhan pemakai sistem perangkat lunak (*user*) dan mengembangkan kebutuhan *user*.

#### C. Analisa Kelayakan Sistem

Dalam tahap ini akan dilakukan analisa terhadap kelayakan digunakan untuk menentukan kemungkinan keberhasilan solusi yang diusulkan. Tahapan ini berguna untuk memastikan bahwa solusi yang diusulkan tersebut benar-benar dapat dicapai.

### 3.5.2 Spesifikasi

Langkah yang kedua yang dilakukan adalah penentuan spesifikasi dari perangkat lunak yang sesuai dengan kebutuhan *user*, spesifikasi ditetapkan setelah melakukan analisis kebutuhan *user*, dan mengetahui keinginan dari *user* mengenai perangkat lunak yang akan dibangun. Data hasil dari analisa kebutuhan *user* telah divalidasi dan sesuai dengan kriteria yang diinginkan oleh *user*.

### 3.5.3 Desain sistem

Dalam tahap ini akan dirancang desain Spesifikasi kebutuhan dari tahap sebelumnya, selain itu dalam tahap ini juga akan dirancang arsitektur tampilan GUI, menentukan perangkat keras (*hardware*) serta menentukan sistem persyaratan yang dapat membantu mendefinisikan sistem secara keseluruhan. Dalam proses desain ini adalah proses multi langkah dimana akan berfokus dalam empat aspek, diantaranya adalah arsitektur perangkat lunak, struktur data, representasi *interface*, serta detail *procedural*. Dalam tahap perancangan desain akan dilakukan beberapa tahapan diantaranya adalah:

#### A. Perancangan Proses

Perancangan proses merupakan rancangan sistem yang akan dibuat dalam perangkat lunak perancangan ini dapat divisualisasikan dengan menggunakan diagram alir (*flowchart*). Selanjutnya diagram alir ini dijelaskan dalam model logika yang mana model logika ini akan memberikan gambaran secara lebih detail tentang bagaimana nantinya fungsi-fungsi dalam sistem juga bagaimana logika akan bekerja.

#### B. Perancangan Basis Data

Perancangan basis data yang dimaksud adalah merancang menunjukkan hubungan antara kenyataan dan harapannya.

#### C. Perancangan Antarmuka

Dalam tahap perancangan antar muka akan dibuat desain aplikasi. Desain aplikasi adalah tahap yang harus dilakukan sebelum mulai pembuatan aplikasi. Perancangan dalam mendesain halaman aplikasi adalah menentukan tampilan pada halaman aplikasi yang akan digunakan oleh *user* untuk melakukan interaksi.

### 3.5.4 Pengkodean

Desain harus ditranslasikan kedalam program perangkat lunak. Dalam tahap pengkodean ini akan dilakukan perubahan konsep sebelumnya yang telah didesain untuk buat realisasinya dalam perangkat lunak, pengkodean ini akan menghasilkan tampilan aplikasi sebagaimana telah dirancang sebelumnya. Dengan demikian aplikasi yang telah dibuat dapat digunakan oleh *user* sesuai dengan kebutuhan yang telah dianalisis sebelumnya.

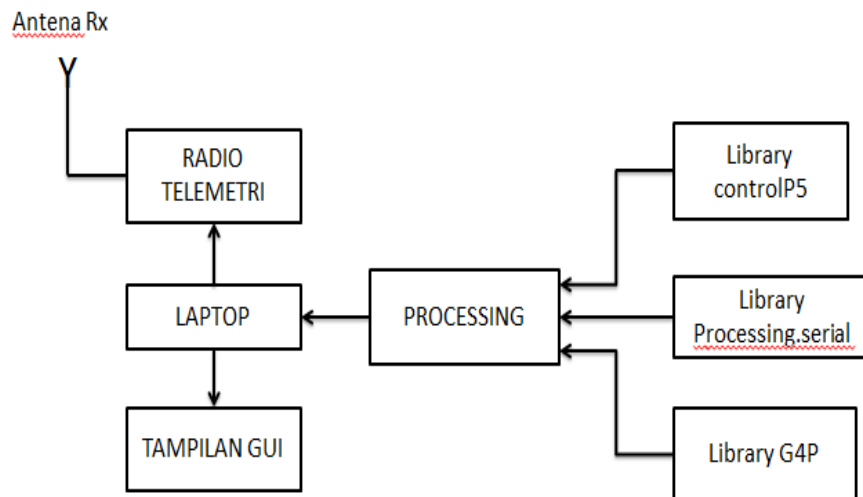
### 3.5.5 Pengujian Dan Integrasi

Tahapan pengujian dilakukan pengujian pada perangkat lunak dari segi *logik* dan *fungsi* selain itu pengujian juga dilakukan untuk memastikan bahwa semua sistem yang ada telah berjalan sebagai mana mestinya bekerja dengan baik sesuai dengan kebutuhan dari *user*. Tahapan pengujian dilakukan untuk meminimalisir kesalahan (*error*) dan memastikan keluaran yang dihasilkan sesuai yang diinginkan.

### 3.5.6 Operasi dan pemeliharaan

Tahap selanjutnya adalah tahap operasi dan pemeliharaan , dalam tahap ini akan dilakukan perbaikan-perbaikan dari sistem perangkat lunak, baik struktur, tampilan, ataupun fitur-fitur yang dirasa perlu untuk ditambahkan. Penambahan dan perbaikan ini berdasarkan dari kebutuhan dari *user*, dan disesuaikan dengan keinginan *user*. Perbaikan ini akan dilakukan secara bertahap sehingga akan tercipta sebuah perangkat lunak yang mendekati sempurna dan sesuai dengan kebutuhan dari *user*.

### 3.5.7 Blok Diagram Pembangunan GUI



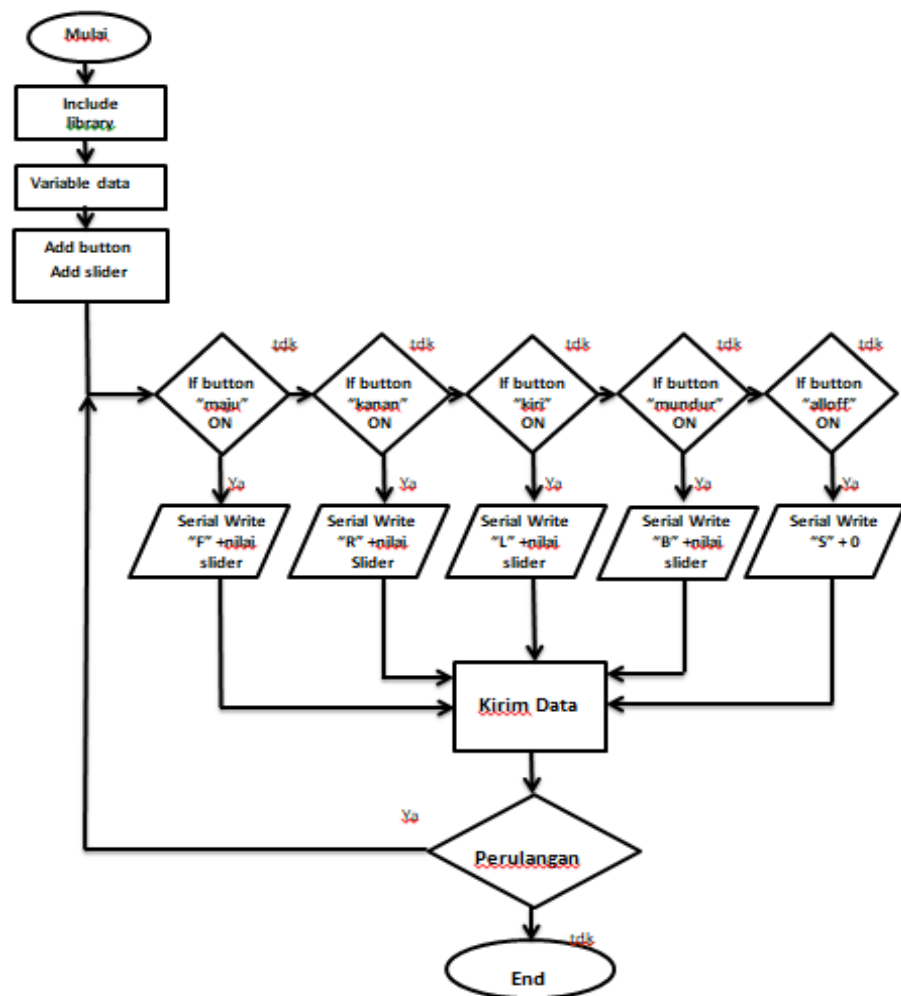
**Gambar 3.14** Block Skema Sistem Human-Robot Interface

Diagram block 3.14 diatas merupakan diagram blok sistem dari pembangunan tampilan GUI. Dalam penelitian ini GUI digunakan sebagai media interaksi antara manusia dengan robot. Tampilan visual GUI pada komputer akan dibangun menggunakan *software processing*. Dengan memanfaatkan *Processing IDE*, dimana *software* ini merupakan tempat dimana GUI dibangun. Untuk mendukung beberapa fitur yang dapat dilakukan oleh perangkat lunak, maka dibutuhkan beberapa *library* diantaranya adalah *library ControlP5*, *library G4P*, serta *library Processing Serial*. *Library ControlP5* digunakan agar perangkat lunak mampu menambahkan tombol-tombol, selanjutnya tombol-tombol ini yang dapat digunakan untuk mengendalikan pergerakan dari robot. *Library G4P* digunakan untuk mendukung fitur slider dan import gambar sehingga tampilan halaman dari aplikasi terlihat lebih menarik. Selanjutnya *library Processing Serial* digunakan untuk

membantu fitur komunikasi, sehingga dengan *library* ini memungkinkan perangkat komputer dan Arduino dapat berkomunikasi melalui Serial.

### 3.5.8 Flowcart Sistem Kerja GUI

Selanjutnya untuk mendukung pengendalian jarak jauh menggunakan tampilan GUI pada komputer maka dibuatlah algoritma tertentu yang dapat digunakan untuk mengontrol pergerakan robot yang dikirim melalui sebuah komputer. Algoritma pembangun tampilan GUI pada komputer yaitu sebagai berikut :



Gambar 3.15 Flowchart Tampilan GUI

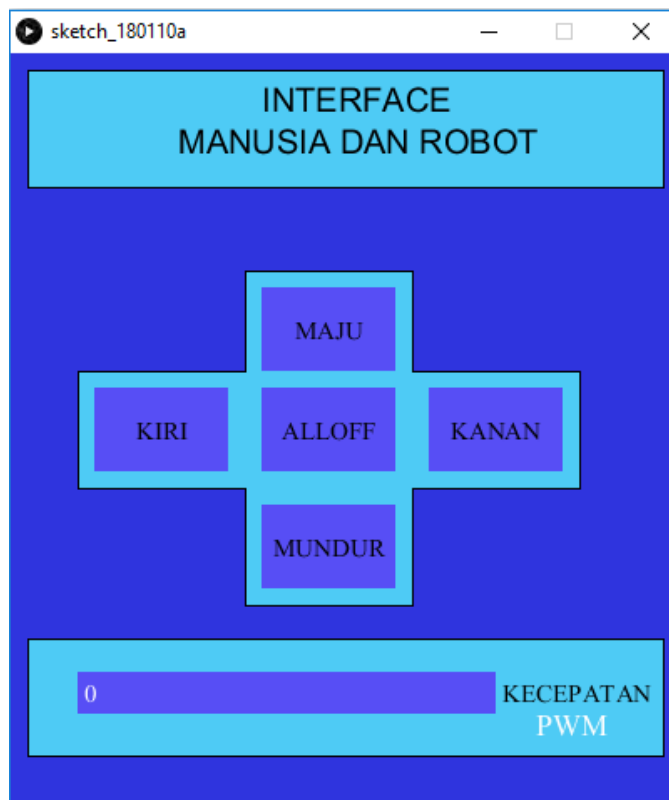
### 3.6 Pembuatan GUI Interface Robot

Dalam penelitian ini akan dibuat GUI sebagai media interface antara manusia dengan robot. Akan dibangun 3 versi tampilan GUI yang akan dibagikan kepada 25 *user*. Dari 3 versi GUI yang ditampilkan memiliki kombinasi warna dan penyusunan struktur yang berbeda. Selanjutnya *user* dimintai keterangan GUI mana yang lebih dominan disukai oleh *user*, baik dari desain struktur, tata letak, pemilihan warna dan keefektifan pengontrolan robot.

Dalam tampilan GUI yang akan dibangun nantinya terdiri dari 2 tipe pengontrolan robot tank, diantaranya adalah control navigasi robot dan control kecepatan robot. Dalam bagian control navigasi akan terdapat empat buah tombol yang digunakan untuk mengontrol arah gerak robot yaitu tombol maju, tombol mundur, tombol kanan, dan tombol kiri. Sedangkan pada bagian control kecepatan terdapat sebuah skroll slider yang dapat diatur nilainya, pengarutan nilai ini dilakukan dengan cara menggeser slider. Nilai yang dapat dihasilkan oleh slider ini berada pada range 0-255. Dalam pengoperasiannya, tombol dan slider ini dikombinasikan melalui algoritma tertentu sehingga ketika tombol navigasi ditekan dan nilai slider diatur maka GUI akan mengirimkan data arah dan kecepatan untuk mengendalikan robot tank.

### 3.6.1 Perancangan GUI Versi 1

#### A. Tampilan GUI



Gambar 3.16 Rancangan GUI Versi 1

#### B. Sketsa Program

Seperti yang telah dipaparkan diatas bahwa GUI terdiri dari 2 bagian utama yaitu tombol navigasi dan tombol pengatur kecepatan. Tampilan GUI versi 1 dapat di amati dalam Gambar 3.16. dari gambar 3.16 tersebut dapat diamati bahwa tombol navigasi terdiri dari 5 buah tombol yang mempunyai fungsinya sendiri-sendiri. Tombol-tombol tersebut diantaranya adalah maju, mundur, kiri, kanan, dan Alloff. Sedangkan untuk tombol pengatur kecepatan adalah berupa slider.

Tampilan GUI diatas dibangun menggunakan *software processing* 3.3.6. adapun sketsa pemrogramannya adalah sebagai berikut:

#### 1. Sketsa Program Membuat Tombol

Sebelum menambahkan beberapa tombol untuk navigasi robot tank, perlu dilakukan pengaturan awal terlebih dahulu. Pengaturan ini meliputi warna tombol saat stanby, aktif, dan ketika disorot. Perintah untuk membuat tombol ini adalah sebagai berikut:

**Tabel 3.4** Seting Awal Tombol Navigasi

<b>Sketsa Processing</b>	<b>Fungsi</b>
<code>cp5 = new ControlP5(this);</code>	Memanggil library ControlP5
<code>cp5.setColorForeground(c olor(50,250,247));</code>	Warna ketika tombol disorot (R, G, B)
<code>cp5.setColorActive(color(4 7,222,123));</code>	Warna ketika tombol ditekan (R, G, B)
<code>cp5.setColorBackground(c olor(87, 78, 245));</code>	Warna ketika tombol standby (R, G, B)
<code>cp5.setColorCaptionLabel( color(0,0,0));</code>	Warna tulisan pada tombol (R, G, B)

**Tabel 4.5** Penambahan Tombol Maju

<b>Sketsa Processing</b>	<b>Fungsi</b>
<code>cp5.addButton("maju")</code>	Penambahan tombol dengan nama maju
<code>.setPosition(150,140)</code>	Peletakan posisi tombol



	pada koordinat (150,140)
.setSize(80,50)	Ukuran tombol (80x50 piksel)
.setFont(font);	Jenis tulisan (Times New Roman, ukuran 15)

Tabel 3.6 Penambahan Tombol Kanan

<b>Sketsa Processing</b>	<b>Fungsi</b>
cp5.addButton("kanan")	Penambahan tombol dengan nama kanan
.setPosition(250, 200)	Peletakan posisi tombol pada koordinat (250, 200)
.setSize(80,50)	Ukuran tombol (80x50 piksel)
.setFont(font);	Jenis tulisan (Times New Roman, ukuran 15)

Tabel 3.7 Penambahan Tombol Kiri

<b>Sketsa Processing</b>	<b>Fungsi</b>
cp5.addButton("kiri")	Penambahan tombol dengan nama kiri
.setPosition(50, 200)	Peletakan posisi tombol pada koordinat (50, 200)
.setSize(80,50)	Ukuran tombol (80x50 piksel)
.setFont(font);	Jenis tulisan (Times New Roman, ukuran 15)

**Tabel 3.8** Penambahan Tombol Mundur

<b>Sketsa Processing</b>	<b>Fungsi</b>
<code>cp5.addButton("mundur ")</code>	Penambahan tombol dengan nama mundur
<code>.setPosition(150, 270)</code>	Peletakan posisi tombol pada koordinat(150, 270)
<code>.setSize(80,50)</code>	Ukuran tombol (80x50 piksel)
<code>.setFont(font);</code>	Jeis tulisan (Times New Roman, ukuran 15)

**Tabel 3.9** Penambahan Tombol STOP

<b>Sketsa Processing</b>	<b>Fungsi</b>
<code>cp5.addButton("alloff")</code>	Penambahan tombol dengan nama alloff
<code>.setPosition(150, 200)</code>	Peletakan posisi tombol pada koordinat(150, 200)
<code>.setSize(80,50)</code>	Ukuran tombol (80x50 piksel)
<code>.setFont(font);</code>	Jeis tulisan (Times New Roman, ukuran 15)

## 2. Sketsa Program Membuat Slider

Tabel 3.10 Perintah Penambahan Slider

<b>Sketsa Processing</b>	<b>Fungsi</b>
<code>cp5 = new ControlP5(this);</code>	Memanggil library ControlP5, dan mengganti nama ControlP5 menjadi cp5
<code>Slider kecepatan= cp5.addSlider("kecepatan")</code>	Menambahkan slider dengan nama "kecepatan"
<code>.setPosition(40,370)</code>	Slider Berada pada titik koordinat (40, 370)
<code>.setSize(250,25)</code>	Ukuran slider adalah (250,25)
<code>.setValue(0)</code>	Nilai awal slider adalah 0
<code>.setColorForeground(color(50,250,247))</code>	Warna slider dalam format(R,G,B)
<code>.setColorActive(color(47,222,123))</code>	Warna slider saat active
<code>.setColorBackground(color(87, 78, 245))</code>	Warna background slider
<code>.setColorCaptionLabel(color(0,0,0))</code>	Warna tulisan pada slider
<code>.setRange(0, 255)</code>	Setting rentang nilai mulai 0-255
<code>.setFont(createFont("Times New Roman", 15))</code>	Setting jenis font tulisan(Times New Roman, ukuran 15)

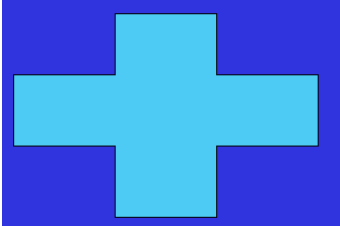
**Tabel 3.11** Perintah Penambahan Label Nilai PWM

<b>Sketsa Processing</b>	<b>Fungsi</b>
labelNilai = cp5.addLabel("labelNilai")	Menambahkan label dengan nama "labelNilai"
.setColorCaptionLabel(color(255,0,0))	Setting warna tulisan dengan format (R,G, B)
.setPosition(310, 390)	Posisi label berada pada koordinat (310, 390);
.setFont(createFont("Times New Roman", 18))	Setting jenis tulisan dan ukurannya
.setText("PWM")	Menambahkan tulisan "PWM"

## 3. Sketsa Program Tampilan

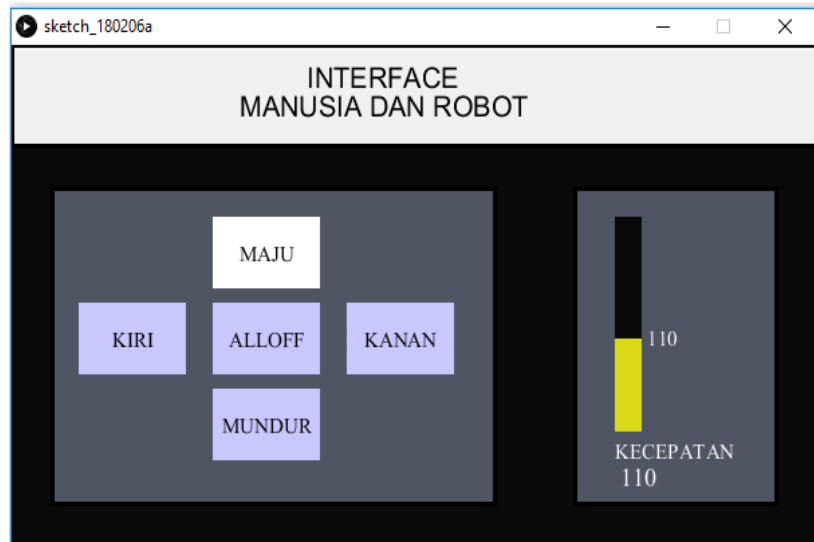
**Tabel 3.12** Sketsa Pada Void Draw();

<b>Sketsa Processing</b>	<b>Fungsi</b>
background(47,52,222);	Berfungsi untuk memberikan warna dasar pada GUI dengan format (R, G, B)
fill(78,203,245);	warna pada objek
beginShape();  vertex(140, 130);  vertex(240,130);  vertex(240,190);  vertex(340,190);	Perintah ini digunakan untuk membentuk objek bebas sesuai dengan keinginan. Setiap titik yang akan dihubungkan untuk membentuk sebuah objek dipanggil dengan perintah vertex,

<pre>vertex(340,260); vertex(240,260); vertex(240,330); vertex(140,330); vertex(140,260); vertex(40,260); vertex(40,190); vertex(140,190); vertex(140,130); endShape();</pre>	<p>perintah ini akan menghubungkan dari titik-titik. Program tersebut akan membentuk sebuah bentuk seperti berikut:</p> 
<pre>text("INTERFACE", 150, 35);</pre>	<p>Menuliskan INTERFACE koordinat (150, 35);</p>
<pre>text("MANUSIA DAN ROBOT", 100, 60);</pre>	<p>Menuliskan MANUSIA DAN ROBOT koordinat (100,60);</p>
<pre>textFont(createFont("Arial", 20));</pre>	<p>Tipe tulisan Arial ukuran 20;</p>

### 3.6.2 Perancangan GUI Versi 2

#### A. Tampilan GUI



Gambar 3.17 Tampilan GUI Versi 2

#### B. Sketsa Program

Sama halnya dengan GUI pada Versi 1, GUI versi 2 terdiri dari 2 bagian utama yaitu tombol navigasi dan tombol pengatur kecepatan. Adapun Tampilan dari GUI versi 2 dapat diamati dalam Gambar 3.17. Dari gambar 3.17 tersebut dapat diamati bahwa tombol navigasi terletak pada bagian sebelah kiri layar. Tombol navigasi terdiri dari 5 buah tombol yang masing-masing mempunyai fungsi tersendiri. Tombol-tombol tersebut diantaranya adalah maju, mundur, kiri, kanan, dan Alloff. Sedangkan untuk tombol pengatur kecepatan ditampilkan dalam bentuk slider ditempatkan di sebelah kanan layar.

GUI versi 2 juga dibangun menggunakan *software processing 3.3.6*. secara garis besar GUI versi 2 memiliki banyak kesamaan dengan GUI versi 1, hanya saja terdapat beberapa modifikasi pada aspek kombinasi warna, dan tata

letak tombol. Adapun beberapa perbedaan tersebut memiliki sketsa pemrograman adalah sebagai berikut:

1. Modifikasi Sketsa Pembuatan Tombol

**Tabel 3.13** Perubahan Tombol Navigasi Versi 2

<b>Perubahan Warna Pada Tombol</b>		
<b>Awal</b>	<b>Perubahan</b>	<b>Fungsi</b>
cp5.setColorForeground(color(50,250,247));	cp5.setColorForeground(color(255,255,255));	Mengubah warna menjadi putih
cp5.setColorActive(color(47,222,123));	cp5.setColorActive(color(217,217,24));	Mengubah warna menjadi kuning
cp5.setColorBackground(color(87,78,245));	cp5.setColorBackground(color(200,200,255));	Mengubah warna menjadi abu-abu
cp5.setColorCaptionLabel(color(0,0,0));	cp5.setColorCaptionLabel(color(0,0,0));	Mengubah warna menjadi Hitam
<b>Perubahan Letak Tombol</b>		
<b>Awal</b>	<b>Perubahan</b>	<b>Fungsi</b>
.setPosition(150,140)  // maju	.setPosition(150,120)  //maju	Mengubah koordinat
setPosition(250,140)	.setPosition(250,120)	

0, 200) // kanan	180) //kanan	tombol sesuai dengan ukuran layar windows
.setPosition(50, 200) //kiri	.setPosition(50, 180) //kiri	
.setPosition(150, 270) //mundur	.setPosition(150, 240) //mundur	
setPosition(150, 200) //alloff	.setPosition(150, 180) //alloff	

## 2. Modifikasi Tombol kecepatan

**Tabel 3.14** Tombol PWM Versi 2

<b>Sketsa Processing</b>	<b>Fungsi</b>
cp5 = new ControlP5(this);	Memanggil library ControlP5, dan mengganti nama ControlP5 menjadi cp5
Slider kecepatan= cp5.addSlider("kecepatan")	Menambahkan slider dengan nama "kecepatan"
.setPosition(450,120)	Slider Berada pada titik koordinat (450,120)
.setSize(20,150)	Perubahan posisi dari horizontal menjadi tegak ditandai dengan nilai X



	lebih kecil dari pada nilai Y(20,150)
<code>..setValue(50)</code>	Perubahan nilai awal yaitu 50
<code>.setColorForeground(color(217,217,24))</code>	Warna slider dalam format(R,G,B)
<code>.setColorActive(color(217,217,24))</code>	Warna slider saat active (R,G,B)
<code>.setColorBackground(color(10 , 10, 13))</code>	Warna background slider (R,G,B)
<code>.setRange(0, 255)</code>	Setting rentang nilai mulai 0-255
<code>.setFont(createFont("Times New Roman",15))</code>	Setting jenis font tulisan(Times New Roman, ukuran 15)

### 3. Modifikasi Sketsa Pada program tampilan

**Tabel 3.15** Perubahan Tombol Kecepatan Versi

<b>Sketsa Processing</b>	<b>Fungsi</b>
<code>background(10 , 10, 13);</code>	Membuat warna latar berwarna (10,10,13) atau hitam
<code>fill(240,240,240);</code>	Mewarnai objek dengan warna abu-abu terang
<code>rect(0,0, 600, 70);</code>	Membuat kotak dengan ukuran 600x70

<code>fill(0,0,0);</code>	Mewarnai objek dengan warna hitam
<code>text("INTERFACE", 220, 30);</code>	Menulis INTERFACE pada koordinat (220, 30);
<code>text("MANUSAI DAN ROBOT", 170, 50);</code>	Menulis MANUSIA ROBOT pada koordinat (170, 50);
<code>textFont(createFont("Arial", 20));</code>	Jenis tulisan Arial ukuran 20
<code>fill(80,85,100);</code>	Mewarnai objek dengan warna abu-abu tua
<code>strokeWeight(3);</code>	Tebal garis 3 piksel
<code>rect(30,100, 330,220);</code>	Membuat kotak pada koordinat (30.100) dengan ukuran 330x220
<code>rect(420,100, 150,220);</code>	Membuat kotak pada koordinat (420.100) dengan ukuran 150x220

### 3.6.3 Perancangan GUI Versi 3

#### A. Tampilan GUI Versi 3



Gambar 3.18 Tampilan GUI Versi 3

#### B. Sketsa Program

Pada program GUI versi ke 3 juga mengalami beberapa modifikasi yang dilakukan pada beberapa bagian diantaranya pada bagian tombol navigasi tidak menggunakan tulisan lagi melainkan menggunakan symbol arah ke atas yang mewakili tombol untuk memberikan perintah bergerak maju, arah kanan yang digunakan untuk memberikan perintah berbelok ke kanan, demikian juga arah ke kiri dan ke bawah yang masing-masing mewakili setiap gerakan dari robot. Pada bagian tengah terdapat tombol alloff yang digunakan untuk menstop pergerakan robot. Sedangkan pada bagian tombol kecepatan, penentuan nilai PWM menggunakan knop putar, bukan menggunakan slider seperti pada versi 1, dan 2. Beberapa perubahan yang dilakukan pada program versi 3 ini adalah sebagai berikut :

## 1. Modifikasi Tombol Kecepatan

Tabel 3.16 Sketsa Knop Tombol Kecepatan

<b>Sketsa <i>processing</i></b>	<b>Fungsi</b>
<code>cp5 = new ControlP5(this);</code>	Memanggil library <code>controlP5</code>
<code>Knob kecepatan = cp5.addKnob("kecepatan")</code>	Membuat knob dengan nama "kecepatan"
<code>.setPosition(110, 180)</code>	Mengset posisi knob pada koordinat (110, 180)
<code>.setValue(50)</code>	Memberikan nilai awal knob 50
<code>.setRange(0,255)</code>	Mengatur range knob dari 0 hingga 255
<code>.setRadius(60)</code>	Mengatur besarnya radius knob sebesar 60
<code>.setNumberOfTickMarks(20)</code>	Mengatur jumlah tick pada knob yaitu sejumlah 20

	buah
<code>.setTickMarkLength(4)</code>	Mengatur panjang dari tiap tick, sebesar 4
<code>.setColorForeground(color(34,139,34))</code>	Menentukan warna untuk knop berwarna biru muda
<code>.setColorActive(color(220,20,60))</code>	Menentukan warna ketika knop active berwarna merah
<code>.setColorBackground(color(0, 191, 255))</code>	Menentukan warna ketika knop standby berwarna hijau
<code>.setColorCaptionLabel(color(255,255,255))</code>	Mengatur warna tulisan pada knop berwarna putih
<code>.setFont(createFont("Arial", 20));</code>	Mengatur jenis tulisan dan ukuran pada knop

## 2. Perubahan tombol Navigasi

Pada program GUI versi 3 ini tombol yang digunakan merupakan symbol berupa arah, sehingga dalam membuat tombol ini sedikit berbeda dengan program pada versi sebelumnya. Dimana pada pembuatan tombol jenis ini membutuhkan deklarasi terlebih dahulu gambar yang akan digunakan sebagai tombol baru setelahnya dipanggil untuk dimasukan kedalam GUI. untuk lebih jelasnya dapat diamati dalam seperti berikut:

```

9
10 void setup(){
11   size(650,400);
12   printArray(Serial.list()); // print semua available serial
13   //port = new Serial(this, "COM9",57600);
14   cp5 =new ControlPS(this);
15
16   // muat gambar
17   PImage[] gambarkan = { loadImage("kananbiru.png"),loadImage("kananijo.png"), loadImage("kananmerah.png")};
18   PImage[] gambarbawah = { loadImage("bawahbiru.png"),loadImage("bawahijo.png"), loadImage("bawahmerah.png")};
19   PImage[] gambarkiri= { loadImage("kiribiru.png"),loadImage("kiriijo.png"), loadImage("kirimerah.png")};
20   PImage[] gambaratas= { loadImage("atasbiru.png"),loadImage("atasijo.png"), loadImage("atasmerah.png")};
21

```

**Tabel 3.17** Import Gambar Sebagai Tombol

<b>Sketsa <i>processing</i></b>	<b>Fungsi</b>
<pre> PImage[] gambarkan = { loadImage("kananbiru.png"), loadImage("kananijo.png"), loadImage("kananmerah.png")}; </pre>	<p><b>Tombol kanan</b> Perintah berikut merupakan perintah untuk memanggil gambar secara berurutan. Secara sederhana dapat diartikan bahwa, gambar tombol ketika stanby adalah kananbiru.png, sedangkan ketika disorot gambar akan berubah menjadi kananijo.png, dan ketika tombol active maka gambar berubah menjadi kananmerah.png</p>

<pre>PImage[] gambarbawah = { loadImage("bawahbiru.png"), loadImage("bawahijo.png"), loadImage("bawahmerah.png");</pre>	Hal yang sama akan terjadi pada tombol bawah, kiri, dan atas
<pre>PImage[] gambarkiri= { loadImage("kiribiru.png"), loadImage("kiriijo.png"), loadImage("kirimerah.png");</pre>	
<pre>PImage[] gambaratas= { loadImage("atasbiru.png"), loadImage("atasijo.png"), loadImage("atasmerah.png");</pre>	

### 3. Modifikasi Sketsa Pada Program Pampilan

**Tabel 3.18** Perubahan Void Draw Versi 3

<b>Skema Processing</b>	<b>Fungsi</b>
<code>background(72,61,139); //</code>	Mengatur warna background menjadi <i>dark state blue</i>
<code>strokeWeight(3);</code>	Mengatur tebal garis menjadi 3 piksel
<code>rect(10,10, 630, 70)</code>	Membuat kotak pada koordinat (10,10) dengan ukuran 630x70
<code>fill(255,255,255);</code>	Mewarnai objek dengan warna putih
<code>text("INTERFACE", 250, 40);</code>	Menuliskan tulisan INTERFACE pada

	koordinat (250,40)
<code>text("MANUSIA DAN ROBOT", 200, 60);</code>	Menuliskan tulisan MANUSIA DAN ROBOT pada koordinat (200, 60)
<code>setFont(createFont("Arial", 20));</code>	Mengatur tipe tulisan Arial ukuran 20
<code>fill(119,136,153);</code>	Mewarnai objek dengan warna abu-abu tua
<code>strokeWeight(3);</code>	Mengatur tebal garis 3 piksel
<code>ellipse(465,245, 280,280);</code>	Menggambar lingkaran dengan titik pusat (465,245) dengan ukuran 280x280
<code>ellipse(170,265,200,200);</code>	Menggambar lingkaran dengan titik pusat (170,265) dengan ukuran 200x200
<code>fill(255,255,255);</code>	Mewarnai dengan warna putih
<code>text("ARAH", 437, 370);</code>	Menuliskan tulisan ARAH pada koordinat (437, 370)
<code>fill(79,86,103);</code>	Mewarnai dengan warna



	abu-abu
<code>strokeWeight(0);</code>	Mengatur tebal garis
<code>rect(515,220, 60,60);</code>	Membuat kotak pada koordinat 515, 220 dengan ukuran 60x60
<code>rect(435,290, 60, 60); //bawah</code>	Membuat kotak pada koordinat 435,290 dengan ukuran 60x60
<code>rect(355,220, 60, 60); // kiri</code>	Membuat kotak pada koordinat 355,220 dengan ukuran 60x60
<code>rect(435,140, 60, 60); //atas</code>	Membuat kotak pada koordinat 435,140 dengan ukuran 60x60
<code>rect(430,220,70, 58); //stop</code>	Membuat kotak pada koordinat 430,240 dengan ukuran 60x60

### 3.7 Pengujian Human-Robot Interface

Pada tahap penerapan dan pengujian sistem interface manusia dan robot, akan dibuat 3 buah versi GUI yang akan diuji cobakan terhadap 25 orang. Pengujian ini dimaksudkan agar didapatkan data versi GUI yang paling diminati oleh *user*. Setelah itu, akan dilakukan beberapa skenario pengujian GUI untuk mengetahui performa dari GUI. Dalam pengujiannya, GUI ini akan dilakukan lima skenario. Skenario pertama adalah pengujian tombol maju, dalam pengujian ini GUI akan dijalankan untuk mengendalikan robot tank. Pengujian dilakukan dengan menekan

tombol maju dan memberikan beberapa nilai PWM mulai dari nilai rendah hingga nilai tinggi. Pada skenario ini, robot akan membaca perintah dari GUI yang dikirimkan, kemudian melakukan respon untuk bergerak maju dengan kecepatan sesuai dengan nilai kecepatan yang diinginkan. Demikian juga untuk skenario 2 dan selanjutnya hanya saja untuk skenario 2, tombol yang ditekan adalah tombol kanan, skenario 3, tombol yang ditekan adalah tombol kiri, skenario 4 tombol yang ditekan adalah tombol mundur dan skenario 5 tombol yang ditekan adalah tombol alloff.

### **3.8 Analisa**

Seperti yang telah dijelaskan pada tahap pengujian sistem interaksi manusia dengan robot, analisa dilakukan terhadap sistem interaksi dengan metode tampilan visual komputer dengan menggunakan GUI ini adalah sebagai berikut:

1. Aspek kemudahan mengontrol robot dengan menggunakan GUI dibandingkan dengan metode tombol.
2. Pengaruh desain tampilan GUI yang digunakan terhadap kenyamanan *user*.
3. Efisiensi tampilan GUI untuk mengontrol robot saat dioperasikan.

### **3.9 Laporan**

Setelah seluruh tahapan penelitian tercapai, maka tahap terakhir adalah pembuatan laporan. Pembuatan laporan mengacu pada tahap-tahap pelaksanaan sebelumnya dan menjelaskan seluruh proses kegiatan terutama pada analisis penelitian.