

## BAB III

### METODE PENELITIAN

#### 3.1 Bahan Penelitian

Penelitian ini menggunakan bahan – bahan berupa dokumen yang berbentuk *hardcopy* seperti buku, publikasi, dan jurnal, maupun dokumen yang *softcopy* seperti *ebook* atau *PDF*. Penelitian ini juga menggunakan perangkat lunak (*software*) seperti *driver* perangkat keras (*hardware*), *library* yang dibutuhkan perangkat lunak atau fungsi – fungsi yang tersedia pada sistem operasi. *Software* yang digunakan diantaranya NetBeans IDE 8.2 dan Arduino IDE 1.6.13, untuk *hardware* menggunakan laptop dengan *processor core i5*, RAM 4GB, sistem operasi *Windows 10 64 bit* dan Arduino Mega 2560, dan untuk *library* menggunakan RxTx.

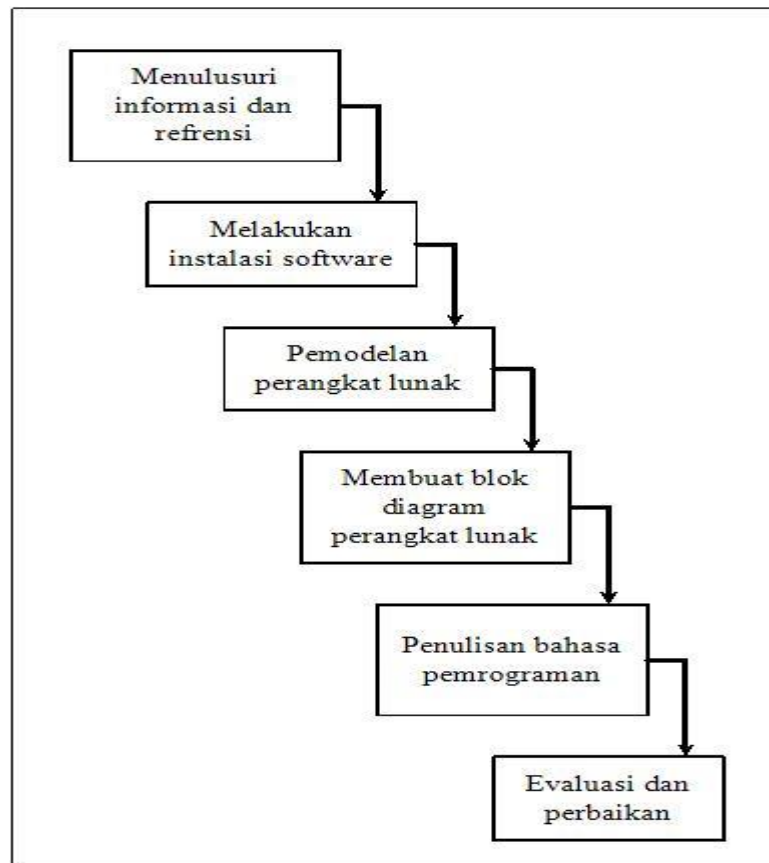
#### 3.2 Alat Penelitian

Penelitian ini menggunakan alat – alat sebagai berikut :

1. Seperangkat laptop Sony Vaio VPCEG18FG dengan spesifikasi Intel Core I5 2410M (2.30 GHz dengan *Turbo Boost Up To 2.90 GHz Sandy Bridge*), HDD 500 GB (SATA III), RAM 4 GB, VGA NVIDIA Geforce 410M (512 MB *Total With Shared 2.3 GB*), *display 14 Inch*, dengan sistem operasi *Windows 10 64 bit*.
2. Arduino Mega 2560 dengan spesifikasi memiliki 54 digital pin *input / output* (yang 15 dapat digunakan sebagai *output PWM*), 16 analog *input*, 4 UART (*hardware port serial*), 16 MHz osilator kristal, koneksi USB, jack listrik, *header ICSP*, dan tombol reset.
3. *Software* NetBeans IDE 8.2 beserta *library* RxTx.
4. *Software* Arduino IDE 1.6.13.
5. Modul *RCTimer Radio Telemetry Kit 433 MHz*.
6. 1 set kit robot mobil pintar dan kabel *jumper male to male*.
7. Baterai Turnigy Lipo 1800 mah 3S 20C.

### 3.3 Cara Penelitian

Penelitian ini dilakukan dengan beberapa tahapan seperti yang ditunjukkan oleh gambar 3.1 sebagai berikut :

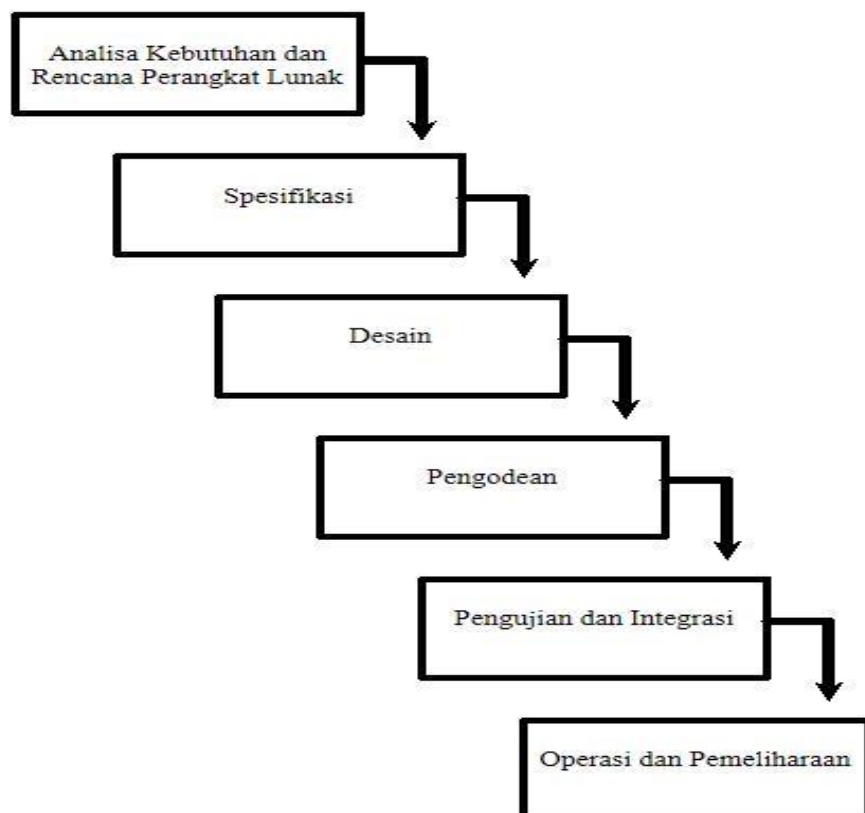


**Gambar 3.1.** Diagram blok tahapan penelitian

1. Menelusuri informasi – informasi dari *literature – literature*, internet, maupun dari dosen pembimbing.
2. Melakukan instalasi – instalasi *software* seperti NetBeans IDE 8.2 beserta *library* RxTx, Arduino IDE 1.6.13, dan *driver RCTimer Radio Telemetry*.
3. Melakukan pemodelan tentang proses dari perangkat lunak yang akan dikembangkan. Pemodelan ini disertai dengan kebutuhan fungsi, *user requirement*, dan perkiraan penyelesaian dalam bentuk *pseudocode*.

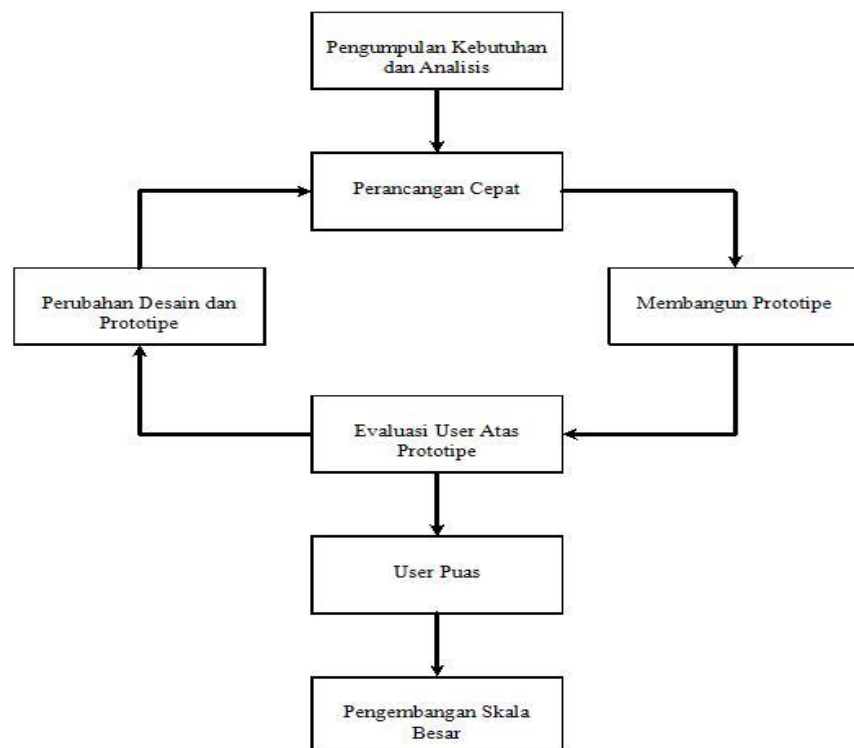
4. Membuat diagram blok atau flowchart tentang bagaimana perangkat lunak berjalan.
5. Melakukan penulisan bahasa pemrograman.
6. Melakukan evaluasi dan perbaikan.

Penelitian ini menggunakan metode *waterfall model* (model air terjun) yaitu sebuah metode yang bertujuan untuk membantu mengatasi kerumitan yang terjadi akibat proyek – proyek pengembangan perangkat lunak. Model ini memungkinkan pemecahan misi pengembangan yang rumit menjadi beberapa langkah logis (kebutuhan, spesifikasi, desain, kode, unit tes, pengujian, dan pemeliharaan) dengan beberapa langkah yang pada akhirnya akan menjadi produk akhir yang siap pakai. Untuk memastikan bahwa sistem bisa dijalankan, setiap langkah membutuhkan validasi, masukan, dan kriteria yang ada. Gambar 3.2 memperlihatkan diagram dari metode *waterfall model*.



**Gambar 3.2.** Diagram *waterfall model*

Langkah awal pada pendekatan metode *waterfall* adalah pengumpulan dan penganalisisan kebutuhan user. Ketika kebutuhan telah ditetapkan, desain dan pengembangan akan dapat dikerjakan. Model ini mengasumsikan bahwa kebutuhan sudah diketahui dan apabila sudah ditetapkan, mereka tidak akan berubah atau segala perubahan menjadi tidak berarti. Kondisi ini menjadi kasus untuk pengembangan sistem dengan arsitektur dan tujuan sistem yang secara menyeluruh telah diinvestigasi. Bagaimanapun, jika terjadi perubahan kebutuhan yang signifikan pada saat spesifikasi sistem sudah final dan produk sudah selesai dikembangkan, model tersebut bukanlah model yang terbaik untuk mengatasi masalah yang akan timbul. Untuk memenuhi kebutuhan dan kepuasan *user* dibuatlah sebuah model pendekatan yang menyediakan sebuah bentuk prototipe dari sebagian maupun keseluruhan sistem. Sebuah protipe adalah bagian dari produk yang mengekspresikan logika maupun fisik antarmuka *eksternal* yang ditampilkan. Gambar 3.3 menunjukkan *flowchart* pendekatan prototipe pada umumnya.



**Gambar 3.3.** *Flowchart* pendekatan prototipe

Langkah yang kedua adalah spesifikasi dari perangkat lunak yang sesuai dengan kebutuhan *user*, spesifikasi ditetapkan setelah analisis kebutuhan user telah divalidasi dan sesuai dengan kriteria yang diinginkan oleh user.

Langkah yang ketiga adalah desain, dalam *water fall model* desain ada dua, yaitu desain tingkat tinggi (*High Level Design* [HLD]) dan desain tingkat rendah (*Low Level Design* [LLD]).

a. Desain Tingkat Tinggi (*High Level Design* [HLD])

HLD adalah proses untuk menentukan sisi *internal* dan *eksternal* dari *perspektif* sebuah komponen. Tujuannya adalah sebagai berikut :

1. Untuk mengembangkan fungsi *eksternal* dan antarmuka , yaitu :
  - Antarmuka pengguna *eksternal*.
  - Antarmuka program aplikasi.
  - Antarmuka program sistem.
  - Antarmuka antarkomponen dan struktur data.
2. Desain struktur komponen *internal* yang mencakup antarmuka antarkomponen dan struktur data.
3. Memastikan bahwa semua kebutuhan fungsi telah terpenuhi.
4. Memastikan bahwa semua komponen sesuai dengan struktur produk dan sistem.
5. Memastikan bahwa desain komponen diselesaikan dengan sempurna.
6. Memastikan fungsi *eksternal* bisa dilakukan.

b. Desain Tingkat Rendah (*Low Level Design* [LLD])

LLD adalah sebuah proses untuk mengubah HLD menjadi desain yang lebih terperinci dari sudut pandang per bagian (*modules, macros, includes*, dan seterusnya). Tujuan LLD adalah sebagai berikut :

1. *Finalisasi* desain komponen dan bagaian (*modules, macros, includes*) di dalam sistem atau produk.
2. Menyelesaikan seluruh rencana uji komponen.
3. Memberikan umpan balik tentang HLD dan verifikasi perubahan yang dilakukan pada HLD.

Langkah yang keempat adalah tahap pengodean, bagian pengodean ini menghasilkan perubahan fungsi dari LLD menjadi benar – benar terkodekan. Tugasnya adalah membuat kode untuk setiap bagian (*modules, macros, includes*), melakukan pengujian terhadap pengodean yang telah dilakukan, melakukan verifikasi terhadap perubahan HLD dan LLD.

Langkah yang kelima unit tes, unit tes adalah pengujian pertama dari modul yang mampu dieksekusi (*executable*). Tujuannya adalah sebagai berikut:

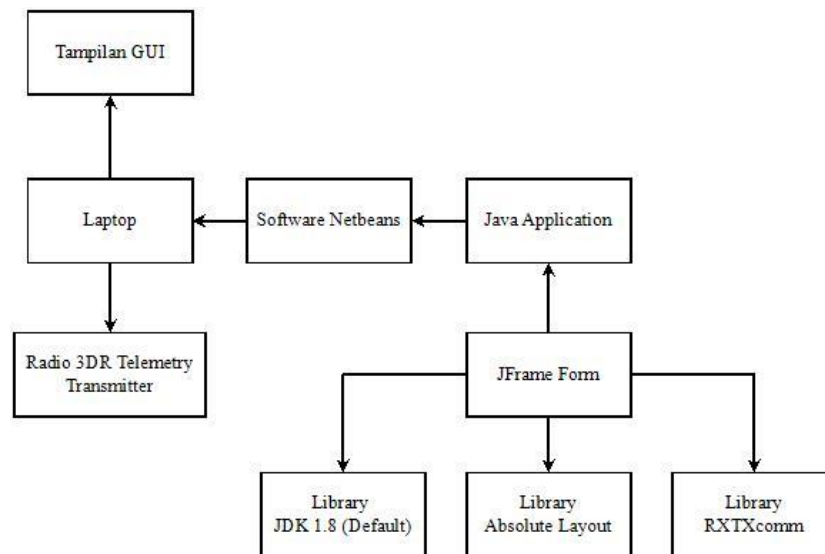
1. Memverifikasi kode terhadap desain tingkat tinggi dan desain tingkat rendah.
2. Mecoba seluruh kode, baik kode baru ataupun lama untuk memastikan bahwa fungsinya dilakukan ke seluruh sistem, membenaran logika, dan verifikasi jalur data.
3. Melihat kembali seluruh pesan kesalahan (*error*), kode kembali, dan reaksi.
4. Memberikan umpan balik mengenai kode , HLD, dan LLD.

Langkah yang keenam uji komponen, uji komponen mengevaluasi gabungan dari bagian perangkat lunak setelah seluruhnya diintegrasikan ke dalam sistem. Tujuannya adalah sebagai berikut :

1. Uji antarmuka dengan pengguna *eksternal* terhadap dokumentasi desain komponen.
2. Uji antarmuka program aplikasi dengan dokumentasi desain komponen.
3. Uji fungsi terhadap dokumentasi desain komponen.
4. Uji antarmuka antarkomponen (tingkat modul) terhadap dokumentasi desain komponen.
5. Uji pemulihan kerusakan dan pesan terhadap dokumentasi desain komponen.
6. Verifikasi komponen sudah selesai dan bisa diterima dalam batasan kualitas.
7. Uji jalur gabungan (*multitasking*) dan sumber gabungan (*file, lock, queues*, dan seterusnya) terhadap dokumentasi desain komponen.
8. Uji fungsi yang tidak diubah terhadap dokumentasi desain komponen.

Perancangan alat harus mempunyai gambaran pembangun GUI dan objek kendali dan bagaimana alat dapat bekerja. GUI dan objek kendali mempunyai beberapa komponen sebagai pembangunnya yang digambarkan dalam bentuk blok diagram maupun *flowchart*. Pada penelitian ini berikut masing –masing cara kerja dan komponen – komponen pembangunnya :

## 1. Diagram Blok Pembangun GUI

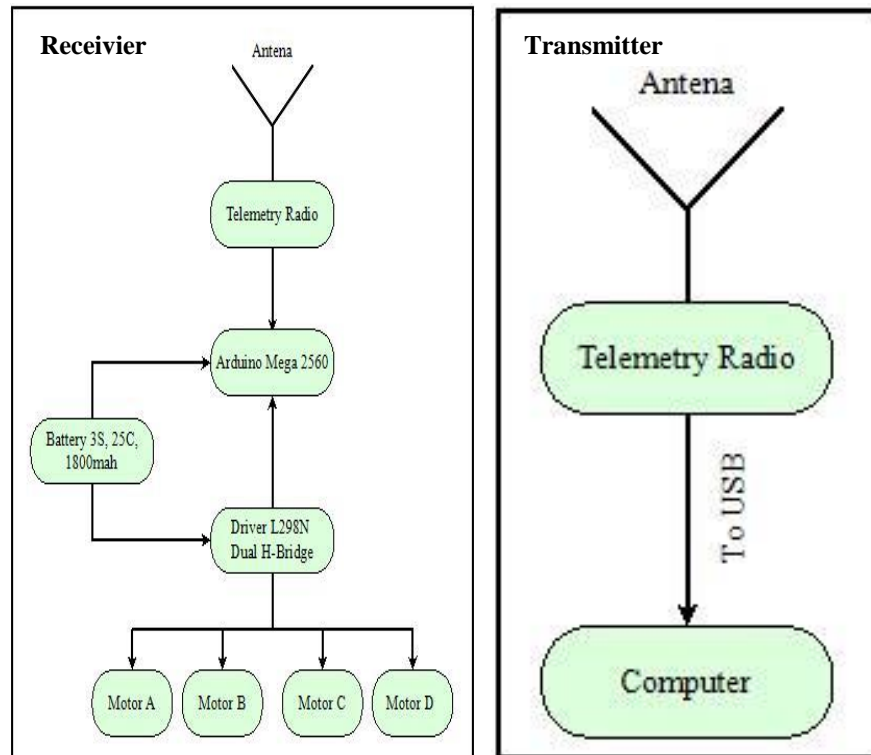


**Gambar 3.3.** Diagram blok pembangun GUI

Pada penelitian ini komponen yang dibutuhkan untuk membangun sebuah GUI adalah laptop, modul radio 3DR telemetry transmitter dan software NetBeans 8.2. GUI dirancang di dalam software NetBeans dengan memanfaatkan Java application berupa JFrame Form. JFrame Form merupakan tempat GUI di rancang dan desain. Agar GUI bekerja sesuai dengan penelitian yang dilakukan dibutuhkan beberapa library seperti *JDK 1.8 (Default)*, *absolute layout* dan *RxTxcomm*. Library *JDK 1.8 (Default)* dan *absolute layout* berfungsi untuk mendesain GUI sedangkan library *RxTxcomm* berfungsi sebagai komunikasi serial. Gambar 3.3 di atas menunjukkan diagram blok dari pembangun GUI.



## 2. Diagram Blok Sistem



**Gambar 3.4.** Diagram blok sistem

Pada gambar 3.4 menunjukkan diagram blok sistem penerima (*receiver*) dan pengirim (*transmitter*). Sistem penerima (*receiver*) terdiri dari beberapa komponen. Fungsi dari tiap - tiap komponen sebagai berikut :

- Arduino Mega 2560 berfungsi sebagai sistem kendali.
- Battery 3S, 25C, 1800mah berfungsi sebagai sumber tegangan Arduino Mega 2560 dan *Driver L298N Dual H-Bridge*.
- Driver L298N Dual H-Bridge* berfungsi sebagai *driver* motor DC.
- Motor DC berfungsi sebagai penggerak tiap roda pada mobil kit pintar.
- Radio 3DR Telemetry 433MHz *Receiver* berfungsi sebagai penerima data.

Sistem pengirim (transmitter) terdiri dari beberapa komponen. Fungsi dari tiap - tiap komponen sebagai berikut :

- a. Radio 3DR Telemetry 433MHz Transmitter berfungsi sebagai pengirim data.
- b. Computer berfungsi sebagai alat pengontrolan robot melalui perancangan GUI yang berjalan berdasarkan software NetBeans 8.2.

Pada penelitian ini komunikasi Arduino Mega 2560 dengan GUI menggunakan metode *simplex* (satu arah). Gambar 3.4 menunjukkan diagram blok komunikasi antara Arduino Mega 2560 dengan GUI.