

## **BAB III**

### **METODELOGI PENELITIAN**

#### **3.1 Metode Penelitian.**

##### **3.1.1 Teknik Pengumpulan data**

Dalam tahapan ini yang dilakukan dalam pengumpulan data ini meliputi sebagai berikut:

##### **3.1.1.1 Wawancara**

Dalam hal ini penulis melibatkan beberapa pihak-pihak terkait untuk dilakukan wawancara demi mendapatkan data yang akurat. Diantara pihak-pihak yang terlibat dalam proses wawancara adalah Pemilik Perusahaan ASTRA HONDA AUTHORIZED SERVICE STATION ( AHASS ) TUNAS DWIPA MATRA.

##### **3.1.1.2 Studi Pustaka**

Pengumpulan data dengan cara mengumpulkan literatur yang bersumber dari jurnal, buku, pakar ataupun dari hasil penelitian orang lain yang berkaitan dengan penelitian yang akan dilakukan.

##### **3.1.1.3 Observasi**

Dalam metode observasi ini penulis diberikan kesempatan untuk melakukan pengumpulan data dengan cara mendatangi langsung Kantor ASTRA HONDA AUTHORIZED SERVICE STATION ( AHASS ) TUNAS DWIPA MATRA.

#### **3.2 Metode Pengembangan Perangkat Lunak**

Pada tahapan pengumpulan perangkat lunak, penelitian ini dilakukan berdasarkan metode pengembangan sistem yang dipilih yaitu metode pengembangan sistem *Prototype*.

### 3.2.1 Komunikasi

Komunikasi harus dilakukan dengan yang tepat. Data objektif dan relevan dengan pokok pembahasan menjadi indikator keberhasilan suatu penelitian. Komunikasi dilakukan dengan cara mengadakan interaksi dengan *admin* dan *user* sehingga kebutuhan perangkat lunak dapat terpenuhi.

### 3.2.2 Perencanaan Cepat

*Quick Plan* (rencana cepat) merupakan tahapan dengan melakukan analisis dan perencanaan setelah kita mendapatkan data-data dari tahapan komunikasi. Analisa kebutuhan *non fungsional* adalah sebuah langka dimana seseorang pembangun perangkat lunak menganalisis sumber daya yang akan menggunakan perangkat lunak yang dibangun. Analisis kebutuhan *non fungsional* tidak hanya menganalisis siapa saja yang akan menggunakan aplikasi tetapi juga menganalisis perangkat keras dan perangkat lunak agar aplikasi dapat berjalan dengan baik. Analisis *non fungsional* yang dilakukan dibagi dalam tiga tahapan, yaitu

#### 3.2.2.1 Analisis Kebutuhan Pengguna (*Admin Dan User*)

Aplikasi untuk menentukan konsentrasi skripsi dan rekomendasi bahasa pemrograman ini akan digunakan oleh *user* dan *admin* dengan ketentuan sebagai berikut:

- a. Menggunakan Sistem Operasi berbasis Android.
- b. Menggunakan Sistem Operasi Android minimum versi 6.0 *Marshmallow* level API 23.

#### 3.2.2.2 Analisis Kebutuhan Perangkat Lunak (*Software*)

Analisis kebutuhan perangkat lunak yang digunakan untuk membangun aplikasi Notifikasi pada android adalah sebagai berikut :

- a. Perangkat lunak sistem operasi pada PC adalah *Microsoft Windows 7 64 bit*.
- b. Perangkat lunak sistem operasi pada android minimum adalah Android versi 6.0. (*Marshmallow*) dengan level API 23.
- c. Perangkat lunak untuk pembuatan program adalah *Android Studio*.

- d. Perangkat lunak yang di gunakan untuk pembuatan *data base* adalah *Firebase*.

### 3.2.2.3 Analisis Kebutuhan Perangkat Keras (*Hardware*)

Analisis kebutuhan perangkat keras yang digunakan untuk membangun sebuah sistem adalah sebagai berikut :

#### Spesifikasi minimum untuk PC :

- a. *Processor Intel(R) Core(TM) i7-2330M @2.20GHZ 2.20GHZ*
- b. *Ram 8 Gb.*
- c. *Harddisk 512 GB.*
- d. *Keyboard dan Mouse.*

#### Spesifikasi minimum untuk android :

- a. *Prosesor Marshmallow 6.0.*
- b. *Ram 4 Gb.*
- c. *Storage 64 Gb.*

Analisa tersebut bukanlah hal yang mutlak, namun merupakan pendapat peneliti tentang minimum penggunaan perangkat keras yang dipakai dalam pengembangan aplikasi.

## 3.3 Pemodelan Desain

Analisis kebutuhan dari tahap sebelumnya akan dipelajari dalam fase ini dan desain *system* disiapkan. Desain Sistem membantu dalam mendefinisikan arsitektur *system* secara keseluruhan.

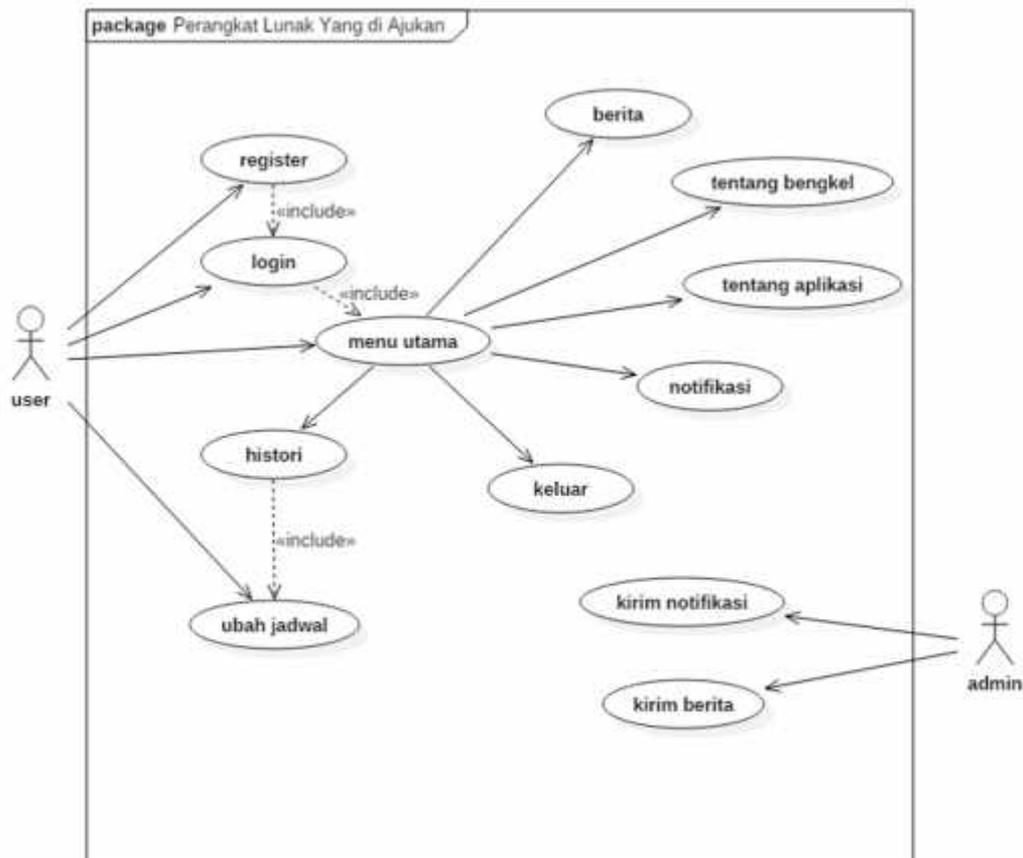
### 3.3.1 Desain UML (*Unified Modelling Language*)

UML (*Unified Modelling Language*) yaitu suatu metode permodelan secara visual untuk sarana perancangan system berorientasi objek, atau definisi UML yaitu sebagai suatu bahasa yang sudah menjadi standar pada visualisasi,

perancangandan juga pendokumentasian sistem software. Pada Penelitian kali ini UML yang dipakai adalah *Use Case*, *Activity Diagram*, dan *Class Diagram*.

### 3.3.1.1 Rancangan Use Case Diagram

a. Rancangan *Use Case* Diagram Perangkat Lunak Yang Diajukan.



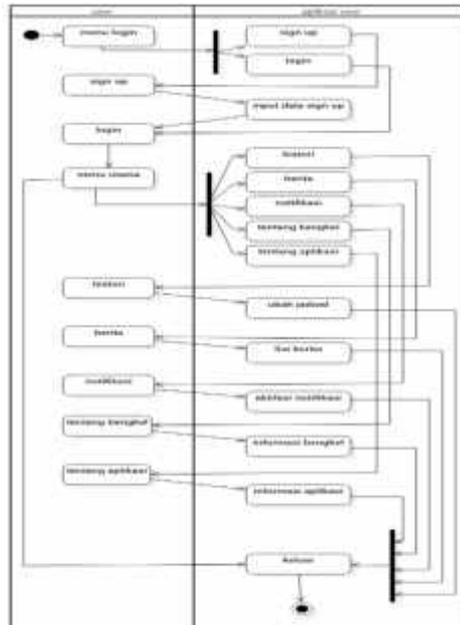
Rancangan *Use Case* dapat dilihat pada gambar 3.1.

**Gambar 3.1** *Use Case* Diagram

### 3.3.1.2 Rancangan Activity Diagram

#### a. Rancangan Activity Diagram User

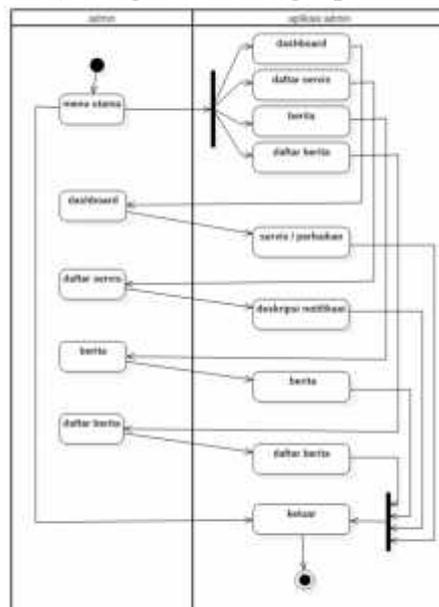
Rancangan *activity diagram* User dapat dilihat pada gambar 3.2



Gambar 3.2 Activity Diagram User

#### b. Rancangan Activity Diagram Admin

Rancangan *activity diagram* tentang aplikasi dapat dilihat pada gambar 3.3 :

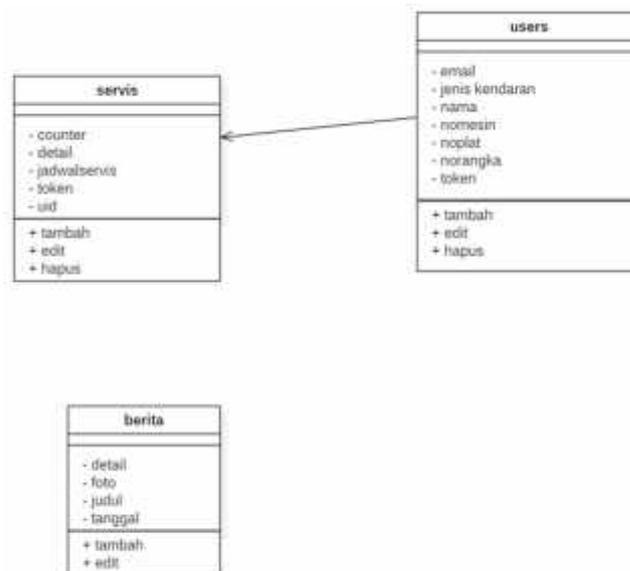




**Gambar 3.5** *Sequence Diagram Admin*

### 3.3.1.4 Rancangan Class Diagram

Rancangan *Class Diagram* dapat dilihat pada gambar 3.6 :



**Gambar 3.6** *Class Diagram*

## 3.4 Basis Data

Basis data yang dipergunakan adalah layanan yang ada pada basis data *firebase*. Berdasarkan pernyataan yang dikeluarkan oleh *Google Developers* (<https://firebase.google.com/products/database/>) berikut adalah keunggulan dari basis data *firebase* :

### a. Sinkronisasi real-time untuk data JSON

*Firestore Real-time Database* adalah *data base No SQL* yang diunggah dan disimpan di *cloud*, yang dapat digunakan untuk menyimpan dan menyinkronkan data antara pengguna secara *real-time*.

### b. Membangun aplikasi tanpa server

Basis data yang *real time* dikirimkan dengan SDK seluler dan web sehingga pengembang dapat membuat aplikasi tanpa memerlukan server. pengembang juga dapat menjalankan kode *back end* yang merespons peristiwa yang dipicu oleh database pengembang menggunakan *Cloud Function* untuk *Firebase*.

#### **c. Dioptimalkan untuk penggunaan *offline***

Ketika pengguna Anda *offline*, SDK *Realtime Database* menggunakan *cache* lokal pada perangkat untuk menayangkan dan menyimpan perubahan. Ketika perangkat online, data lokal otomatis disinkronkan.

#### **d. Keamanan berbasis pengguna yang kuat**

*Realtime Database* terintegrasi dengan *Firebase Authentication* untuk menyediakan proses autentikasi yang mudah dan intuitif bagi *developer*. pengembang dapat menggunakan model keamanan deklaratif kami (*Firebase*) untuk mengizinkan akses berdasarkan identitas pengguna atau pencocokan pola pada data.

#### **e. Bagian dari *platform Firebase***

*Firebase* membantu pengembang membuat aplikasi berkualitas tinggi, mengembangkan basis pengguna, dan menghasilkan lebih banyak uang. Setiap fitur berfungsi secara independen dan bekerja sangat baik jika dipakai bersama.

### **3.5 Desain Antar Muka Aplikasi**

Desain aplikasi merupakan rancangan bagaimana bentuk dan rupa tampilan aplikasi yang ingin di buat.

### 3.5.1 Rancang Tampilan Aplikasi

#### a. Rancang Tampilan Halaman SplashScreen

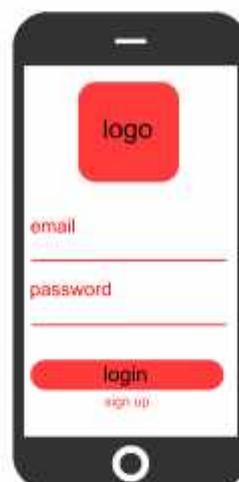
Pada rancangan tampilan splashscreen akan memunculkan halaman splashscreen tyang menunjukkan logo utama aplikasi ini. Tampilan rancangan halaman login dapat dilihat pada gambar 3.7



Gambar 3.7 Halaman SplashScreen

#### b. Rancang Tampilan Halaman Login

Pada rancangan tampilan halaman login terdapat dua fitur input text diantaranya input text username, input text password dan tombol untuk login di dalam aplikasi tersebut. Tampilan rancangan halaman login dapat dilihat pada gambar 3.8



**Gambar 3.8** Halama Login

**c. Rancangan Tampilan Halaman Register**

Pada rancangan tampilan halaman regiater terdapat fitur pengisian data diri seperti email, password, nama, nomer plat, nomer rangka, nomer mesin, dan jenis kendaraan di dalam halaman tampilan tersebut. Tampilan rancangan halaman register dapat dilihat pada gambar 3.9



**Gambar 3.9** Halaman Register

**d. Rancang Tampilan Halaman Menu Utama**

Pada rancangan tampilan halaman menu utama terdapat enam fitur diantaranya histori, berita, notifikasi, tentang bengkel, tentang aplikasi dan keluar di dalama aplikasi tersebut. Tampilan rancangan halaman menu utama dapat dilihat pada gambar 3.10



**Gambar 3.10** Halaman Menu Utama

**e. Rancang Tampilan Halaman Menu Histori**

Pada rancangan tampilan halaman menu histori terdapat fitur yang menampilkan tentang jadwal servis yang akan di lakukan, nomer plat kendaraan, dan keterangan data servis dll, serta terdapat fitur ubah jadwal servis. Tampilan rancangan halaman menu pesanan dapat dilihat pada gambar 3.11



**Gambar 3.11** Halaman Histori

**f. Rancang Tampilan Halaman Menu Berita**

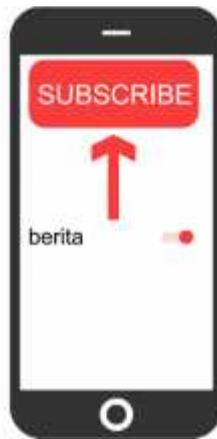
Pada rancangan tampilan halaman menu berita terdapat fitur yang menampilkan beberapa berita yang terdapat di AHASS Tunas Dwipa Matra. Tampilan rancangan halaman menu berita dapat dilihat pada gambar 3.12



**Gambar 3.12** Halaman Berita

**g. Rancang Tampilan Halaman Notifikasi**

Pada rancangan tampilan halaman menu notifikasi terdapat fitur pengaktifan notifikasi yang dapat mengaktifkan atau menonaktifkan notifikasi. Tampilan rancangan halaman menu pesanan dapat dilihat pada gambar 3.13



**Gambar 3.13** Halaman Notifikasi

**h. Rancang Tampilan Halaman Tentang Bengkel**

Pada rancangan tampilan halaman menu tentang bengkel terdapat keterangan alamat bengkel, layanan yang tersedia pada bengkel, customer servis, dan mekanik mobile. Tampilan rancangan halaman menu tentang bengkel dapat dilihat pada gambar 3.14



### **Gambar 3.14** Halaman Tentang Bengkel

#### **i. Rancang Tampilan Halaman Tentang Aplikasi**

Pada rancangan tampilan halaman menu tentang aplikasi terdapat keterangan tentang DARMAJAYA sebagai Institusi yang menaungi penelitian ini. Tampilan rancangan halaman menu tentang aplikasi dapat dilihat pada gambar 3.15



**Gambar 3.15** Halaman Tentang Aplikasi

### **3.6 Pembentukan Prototype**

Pada tahap inilah aplikasi dibuat dengan menggunakan bahasa pemrograman tertentu dan diuji bagaimana program berjalan.

#### **3.6.1 Pembuatan aplikasi**

Dalam penelitian ini aplikasi dibuat pada dasarnya dengan menggunakan *Android Studio* dan juga perangkat lunak penunjang yang telah dijelaskan pada sub bab analisis kebutuhan perangkat lunak. Sedangkan untuk bahasa pemrograman yang digunakan adalah Java.

#### **3.6.2 Pengujian aplikasi**

Pengujian aplikasi dilakukan dengan menggunakan metode *black-box testing*.

Pengujian yang akan dilakukan yaitu dengan menguji lama waktu *loading* atau *respon time* dari masing-masing halaman yang terdapat pada aplikasi. Pengujian dilakukan dengan menggunakan tiga buah perangkat yang berbeda yang memiliki spesifikasi dengan kriteria tinggi, sedang, dan rendah dilihat dari segi perangkat kerasnya.

Berikut adalah ketiga perangkat tersebut dan spesifikasinya :

1. Perangkat Kategori Rendah
  - a. Versi Sistem Operasi : Android Versi 6.0 (*Marsmallow*)
  - b. Kecepatan *Processor* : 2,02 Ghz
  - c. Kapasitas RAM : 3 Gb
2. Perangkat Kategori Sedang
  - a. Versi Sistem Operasi : Android Versi 7.0 (*Nougat*)
  - b. Kecepatan *Processor* : 1,4 Ghz
  - c. Kapasitas RAM : 3 Gb
3. Perangkat Kategori Tinggi
  - a. Versi Sistem Operasi : Android versi 8.1.0(*Oreo*)
  - b. Kecepatan *Processor* : 1,99 Ghz
  - c. Kapasitas RAM : 4 Gb

### 3.7 Penyerahan Sistem dan Umpan Balik

Penyerahan system dan umpan balik merupakan tahapan ketika aplikasi telah selesai dibuat. Aplikasi yang telah dibuat akan didistribusikan melalui *Google Play Store* sehingga semua orang bisa mengunduhnya.

### 3.8 Pengujian Sistem

Pengujian sistem yang akan menggunakan metode *black box testing* untuk menemukan hal-hal berikut:

1. Kesalahan inisialisasi dan terminasi (percobaan instalasi).
2. Kesalahan interface atau antarmuka.

3. Kesalahan dalam struktur data atau akses data.
4. Pengujian notifikasi servis berkala.
5. Pengujian notifikasi berita.