# BAB 3
# ANALISIS DAN PERANCANGAN SISTEM

## 3.1. Analisis Sistem

Pembangunan purwarupa pencarian tempat parkir cerdas menggunakan mikrokontroller Arduino uno adalah sistem yang digunakan untuk mencari tempat parkir secara otomatis menggunakan bantuan perangkat keras yang saling terintegrasi. Sistem ini dibangun menggunakan Arduino mikrokontroler dengan pemrograman Arduino IDE. Sistem ini membantu *user* untuk mencari tempat parkir yang paling terdekat dengan keadaan pengendara saat ini.

Berikut analisis sistem yang ada pada sistem :

1. Sistem mengelola data dengan sendiri nya untuk mendapatkan tempat parkir terdekat
2. Sistem menampilkan data tempat parkir yang kosong dan sudah terisi

### 3.1.1. Analisis Masalah

Dari fenomena yang terjadi terdapat beberapa masalah yang sering terjadi didalam tempat parkir gedung maupun basement yaitu :

1. Sering terjadinya pencarian tempat parkir yang tidak pasti, sehingga membuat pencarian tempat parkir tidak efisien dan membuat antrian panjang didalam gedung maupun basement.
2. Sering terjadinya kesalahan pada saat menampilkan data tempat parkir yang kosong ataupun terisi.

Oleh karena itu dibutuhkan suatu solusi untuk menangani permasalahan yang sedang terjadi yaitu dibutuhkan suatu sistem yang berfungsi untuk :

1. Memudahkan pencarian tempat parkir dalam gedung maupun basement, sehingga dapat mengurangi adanya antrian dan ketidak efisien dalam pencarian tempat parkir dalam gedung maupun basement.

2. Menampilkan data secara valid atau real time, sehingga tidak terjadi kesalahan jumlah tempat parkir yang kosong atau terisi.

### 3.1.2. Analisis Sistem Sejenis

Analisis aplikasi sejenis dimaksudkan untuk menganalisis fungsionalitas dan alur aplikasi yang telah ada pada domain yang sama dengan perangkat lunak pada penelitian ini. Hal ini bertujuan untuk memilah fungsionalitas dan alur yang akan diadopsi pada sistem yang diteliti berdasarkan manfaat yang sudah di hasilkan dari aplikasi yang dianalisis dan mencari uniqueness dari perangkat lunak yang akan dibangun pada penelitian ini.

Pada penelitian ini teknologi yang akan diteliti adalah “Implementasi Sistem Perparkiran Otomatis dengan Menentukan Posisi Parkir Berbasis RFId” yang dilakukan oleh Freeon Alkapon Imbiri,Nandang Taryana dan Decy Nataliana jurusan Teknik Elektro Institut Teknologi Nasional (ITENAS) Bandung.

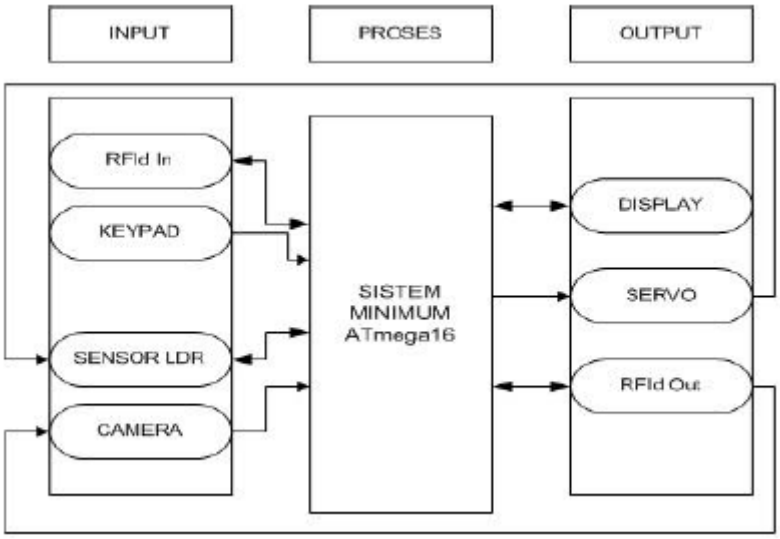
1. Implementasi Sistem Perparkiran Otomatis dengan Menentukan Posisi Parkir Berbasis RFId

Sistem perparkiran yang dirancang untuk mengatur dan memudahkan pengelolaan lahan parkir, mengkomputerisasi input data base nama serta informasi apartemen merupakan tempat tinggal yang ideal di kota metropolitan yang besar dan ramai, namun memiliki area parkir yang tidak efektif karena area parkir yang disediakan biasanya berada di basement atau disekililing area apartemen tersebut. Karena letak tempat parkir tersebut letaknya jauh maka mobil yang hendak memarkirkan kendaraannya kurang mendapatkan informasi mengenai status area parkir serta letak posisi parkir yang masih kosong di perlukan dan membuat sistem parkir otomatis yang memanfaatkan teknologi RFId. Akibat dari kurangnya informasi yang didapat, seringkali para pengguna mobil perlu memakan waktu yang lama untuk sekedar menemukan tempat yang kosong, dan cenderung memarkirkan kendaraannya di sembarang tempat. Untuk itu dibuat suatu sistem perparkiran yang dapat mampu mengatasi masalah yang sering terjadi pada area parkir.

Spesifikasi dari sistem alat ini adalah sebagai berikut :

1. Akses masuk lahan parkiran menggunakan kartu RFId
2. Tag RFId menggunakan ID-12 yang bekerja pada frekuensi 125-134 Khz
3. Komunikasi antara PC menggunakan kabel UTP model Cross
4. Komunikasi dari LCD ke mikro menggunakan kabel serial
5. Komunikasi pengiriman data dari kamera ke pc menggunakan kabel sepanjang ± 1 meter
6. Komunikasi pengiriman data dari PC ke mikrokontroler menggunakan kabel serial dengan panjang 30 centi meter.
7. Sensor LDR mampu menangkap pancaran cahaya dari sensor LED ± 30 cm
8. Motor servo mampu berputar sampai 360 °, dimana pada palang pintu digunakan sudut 0° - 90°

Berdasarkan spesifikasi sistem yang diinginkan, maka blog diagram sistem yang dirancang dapat dilihat pada Gambar 3.1




**Gambar 3.1** Blok Diagram Sistem Secara Umum

Berdasarkan blok diagram pada Gambar 3.1 dapat dilihat bahwa bagian input terdiri dari 3 input utama (*RFId,Numpad* dan LDR dan LED) dan 3 output

utama (*Servo, Display* dan *Camera File*) yang terhubung langsung ke Sistem Mikrokontroler.

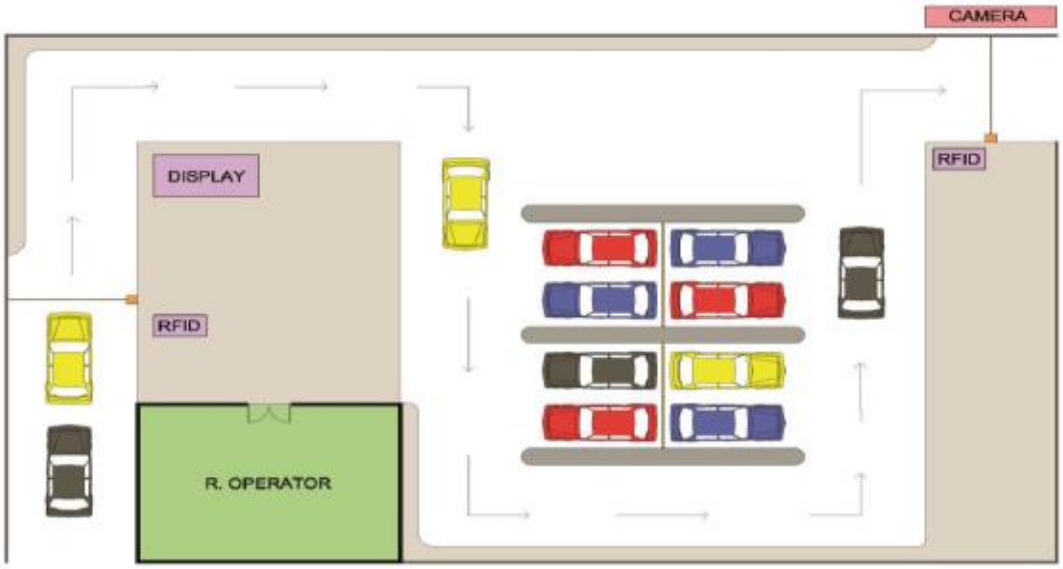
Bagian *Input* Sistem ini terdiri dari :

1. *RFId* yang digunakan untuk memeriksa data member terdaftar atau tidak pada area parkir tersebut.
2. *Numpad* yang digunakan untuk memilih posisi tempat parkir pada lcd dipintu masuk
3. *Sensor LDR dan LED* sensor LDR yang digunakan untuk memberi instruksi pada servo yang pada saat kendaraan sebelum ataupun sesudah melewati palang pintu, dan LED yang berguna untuk memberikan informasi kepada pengendara mengenai status slot parkir yang kosong ataupun terisi

Bagian Output Sistem ini terdiri dari :

1. *Servo* yang digunakan untuk memberikan akses masuk ke pengendara, *Servo* ini akan terbuka jika pengendara sudah tag kartu *RFId* dan akan menutup setelah sensor LDR mengirim data kendaraan sudah melewati palang pintu
2. *Display* yang digunakan untuk memberikan informasi kepada user mengenai slot parkir
3. *Camera File* yang digunakan untuk mengcapture bagian plat nomor kendaraan sebagai data pelengkap dari *RFId*

Berikut arsitektur denah parkir yang akan dirancang dalam sebuah miniatur sistem perparkiran dapat dilihat pada Gambar 3.2




**Gambar 3.2** Arsitektur Denah Parkir

Spesifikasi perangkat lunak yang dirancang adalah :

1. Perancangan dan realisasi perangkat lunak menggunakan bahasa Atmel studio.
2. Perangkat lunak yang direalisasikan dengan menggunakan sensor LDR, RFId, Numpad, Servo, Display LCD serta mikrokontroler.
3. Perangkat lunak dirancang dan direalisasikan untuk melakukan pemilihan posisi lahan parkir, mencatat segala informasi mengenai pengakses lahan parkir yang disimpan dalam *database*.
4. Proses download programing ke mikrokontroler menggunakan *USB Flasher*.

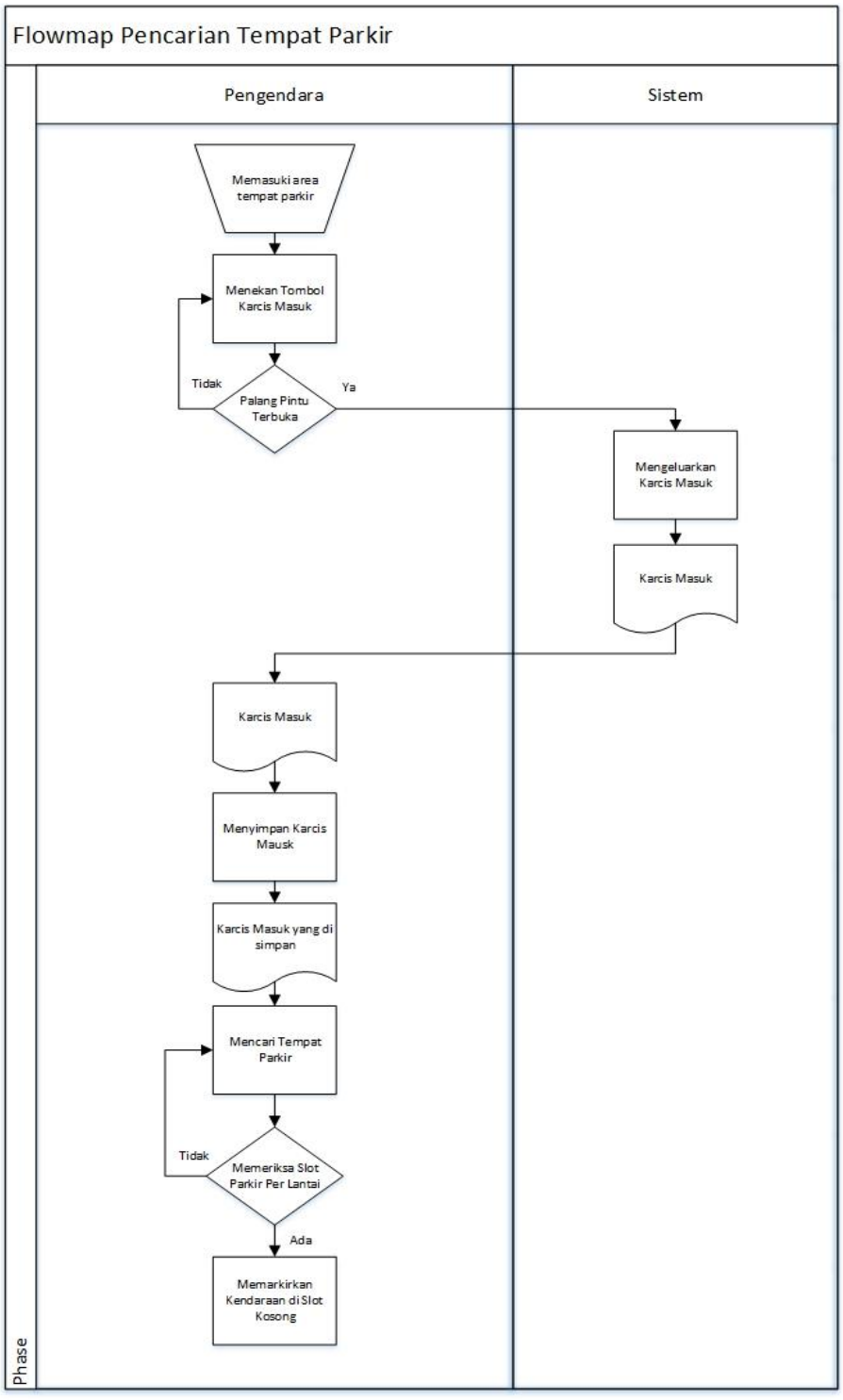
### 3.1.3. Analisis Sistem Yang Sedang Berjalan

Analisis sistem yang sedang berjalan adalah tahapan yang memberikan gambaran sistem yang sedang berjalan saat ini. Analisis itu bertujuan untuk memberi gambaran yang lebih detail bagaimana cara kerja dari sistem yang sedang berjalan. Yang menjadi focus pada penelitian ini adalah prosedur pencarian tempat parkir.

#### 3.1.3.1. Prosedur Pencarian Tempat Parkir Manual

Prosedur pencarian tempat parkir manual adalah prosedur pencarian tempat parkir yang dimana pengendara mencari tempat parkir sendiri. Prosedur yang berlangsung sebagai berikut :

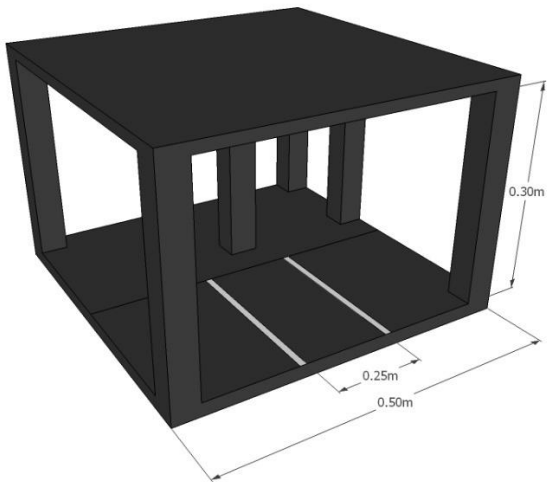
1. Pengendara menekan tombol karcis pada pintu masuk
2. Jika palang pintu terbuka maka karcis akan keluar, jika palang pintu tidak terbuka karcis tidak keluar dan menekan kembali tombol karcis
3. Pengendara memasuki gedung perparkiran
4. Pengendara mencari tempat parkir kosong dari satu lantai ke lantai lain
5. Jika pada lantai pertama terdapat slot kosong makan pengendara akan memarkirkan kendaraannya langsung, jika tidak maka pengendara akan melanjutkan pencarian ke lantai berikutnya




**Gambar 3.3** Flowmap Prosedur Pencarian Tempat Parkir Manual

### 3.1.4. Analisis Denah

Analsis denah digunakan untuk mengetahui seberapa besar denah tempat parkir yang digunakan, analisis denah ini bertujuan untuk mempermudah menentukan pergerakan suatu objek dari satu tempat ke tempat lain berdasarkan keadaan denah dan objek lainnya. Denah yang digunakan pada prototype ini berbentuk persegi dengan panjang 50cm x 50cm untuk alas dan tutupnya, sedangkan tiang tiang yang dipakai untuk membatasi dan membuat prorotype ini membentuk sebuah pilaar pilaar yang sesuai dengan tempat parkir aslinya berbentuk balok yang tingginya 30cm. Untuk lebih jelasnya dapat dilihat pada gambar berikut :

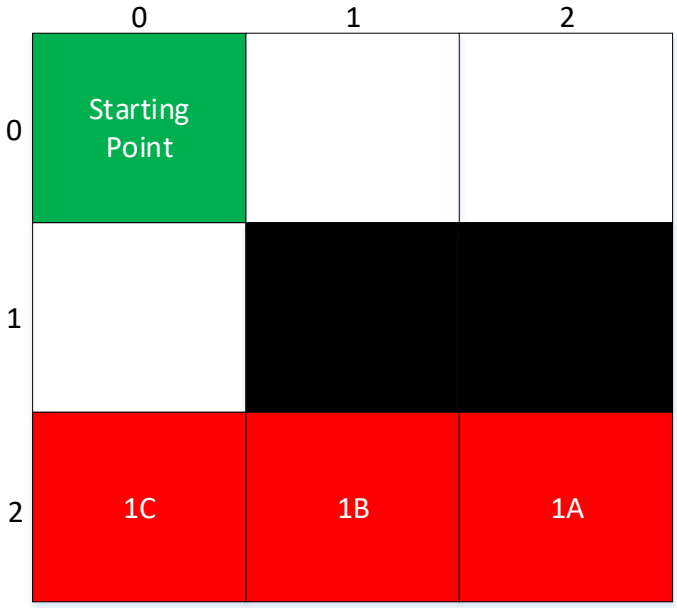



**Gambar 3.4** Analisis Denah

### 3.1.5. Analisis Algoritma A* / A-*Star*

Analisis algoritma ini digunakan untuk menentukan arah ataupun pergerakan pada suatu objek dari satu tempat ke tempat lain berdasarkan keadaan. Pada *prototype* pencarian tempat parkir cerdas ini akan digunakan metode *path finding* untuk mencari jarak terdekat dari pintu masuk untuk mendapatkan slot parkir yang paling dekat dengan kondisi pengendara di pintu masuk tempat parkir. Dalam pemecahan masalah untuk mencari jarak terpendek pada metode *path finding,* algoritma yang akan digunakan adalah algoritma A-*star.* Analisis

algoritma A-*star* pada metode *path finding* akan dimodelkan dalam sebuah kotak yang berbentuk (3x3), dapat dilihat pada gambar berikut :




**Gambar 3.5** Kondisi Pemodelan 3x3

Seperti terlihat pada gambar 3.5, kotak berwarna hitam adalah penghalang atau disebut *unwalkable* yang tidak bisa di lewati, kotak warna hijau merupakan titik *node* awal / *starting point* dimana algoritma A-*star* diterapkan, sedangkan kotak berwarna merah adalah *node* tujuan yang merupakan slot parkir.

Langkah-langkah pencarian jarak terdekat dengan menggunakan algoritma A-*star* adalah sebagai berikut :

1. Masukan *node* awal kedalam *open list*
2. Periksa semua *node* yang berdekatan dengan *node* A apakah *walkable* atau tidak, abaikan *node* dengan penghalang lalu tambahkan semua *node* tersebut ke dalam *open list*. Untuk setiap *node* yang baru ditambahkan, simpan *node* A sebagai *"parent"*
3. Hapus node A dari *open list*, dan tambahkan ke dalam *closed list*
4. Setiap node yang berada pada open list akan dihitung nilai g, dan h-nya maka dapat diperoleh niali f-nya, nilai g diperoleh dengan cara

menghitung nilai (*cost*) jarak dari *open list* terhadap *closed list*. Jika pergerakan nilai g secara horizontal atau vertikal, maka nilai g-nya di definisikan sebesar 10, dan jika pergerakan nilai g secara diagonal, maka nilai g-nya di definisikan sebesar 14. Nilai f didapat dari pertambahan nilai g dan h.

5. Menambahkan node terdekat dari node awal kedalam open list, dari gambar 3.5 node terdekat dari node awal adalah (0,1) dan (1,0). Contoh perhitungan node terdekat menggunakan rumus *manhattan distance* :

   h(n) = D * (abs(n.x-goal.x) + abs(n.y-goal.y))

   Hasil Perhitungan :

   h(0,1) = 10 * (abs(0-0) + abs(0-2))

   h(0,1) = 10 * (abs(0) + abs(-2))

   h(0,1) = 10 * (0 + 2)

   h(0,1) = 10 * 2

   h(0,1) = 20

   Perhitungan f(n) :

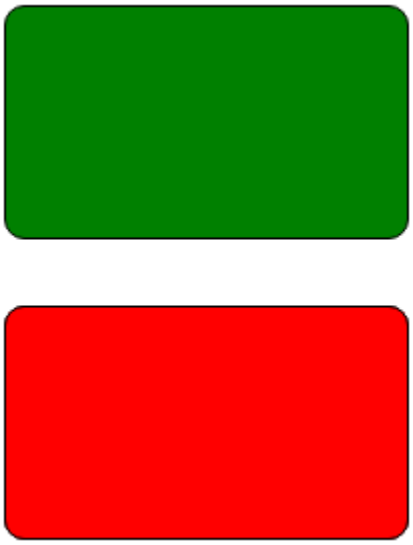
   f(n) = g(n) + h(n)

   f(0,1) = 10 + 20

   f(0,1) = 30

### 3.1.6. Analisis Warna

Analisis warna ini digunakan untuk menentukan warna yang ada pada web *monitoring* yang di gunakan oleh petugas yang bertugas memonitor slot slot parkir yang ada pada web *monitoring*. Warna warna yang ada pada web monitoring diantaranya adalah merah dan hijau, warna warna pada web *monitoring* dapat dilihat pada gambar berikut.

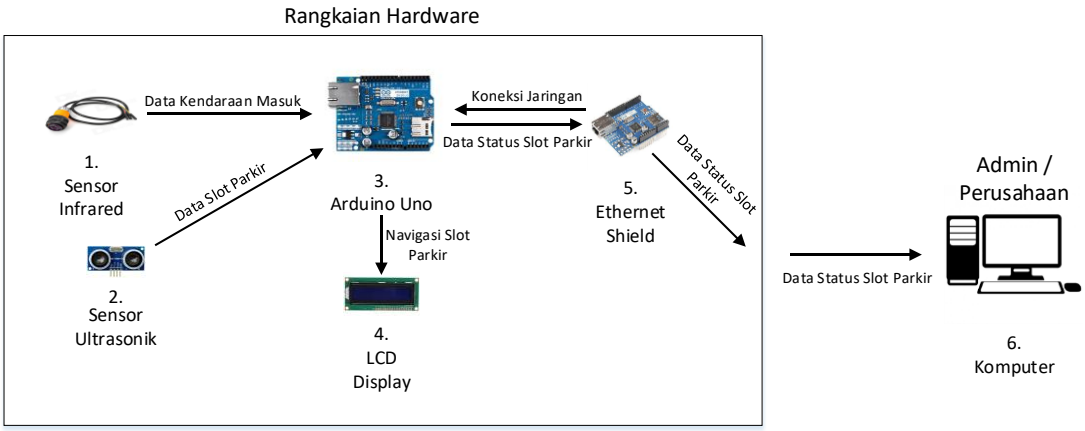


**Gambar 3.6** Warna Slot Parkir Pada Web *Monitoring*

Terdapat dua warna yang ada pada web monitoring, warna merah dan hijau. Warna pada slot monitoring ini sama dengan warna yang ada pada lampu LED Indikator yang ada pada *prototype*. Warna hijau pada web monitoring menunjukkan bahwa slot tersebut tersedia atau kosong, sedangkan warna merah menunjukkan bahwa slot tersebut tidak tersedia atau terisi. Warna slot pada web monitoring dapat berubah ketika sensor ultrasonik mendeteksi objek pada slot tersebut.

### 3.1.7. Analisis Arsitektur Sistem

Analisis arsitektur sistem merupakan sebuah proses untuk mendeskripsikan fisik sistem yang akan dibangun dan juga komponen – komponen pendukungnya. Berikut ini adalah gambaran dari arsitektur sistem yang akan dibangun.



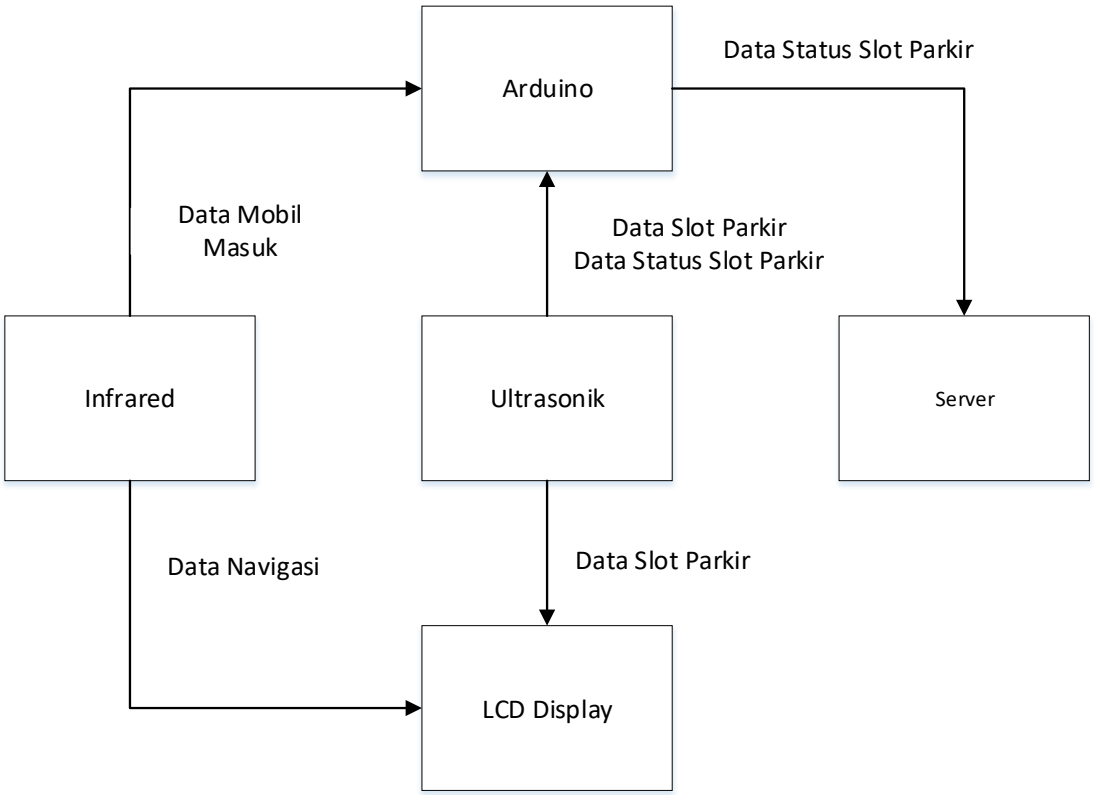

**Gambar 3.7** Arsitektur Sistem

Berikut ini adalah penjelasan mengenai Arsitektur Sistem yang terlihat pada gambar 3.5.

1. Sensor Infrared, berfungsi sebagai alat pendeteksi kendaraan masuk dan dikirimkan ke Arduino uno agar dapat mengelola data kendaraan masuk
2. Sensor Ultrasonik, berfungsi untuk mendapatkan data tempat parkir secara *realtime* yang akan dikirim ke Arduino untuk di kelola sebagai data tempat parkir yang sudah terisi maupun data tempat parkir yang tersedia.
3. Arduino Uno, berfungsi sebagai pusat kendali utama pengumpulan data-data yang di deteksi oleh sensor dan mengirimkan data-data yang telah di hasilkan oleh sensor tersebut ke dalam server.
4. LCD Display, berfungsi untuk memberikan informasi navigasi kepada user yang sedang mencari tempat parkir dan memberikan informasi detail tempat parkir yang tersedia dan sudah terisi
5. Ethernet Shield, berfungsi sebagai papan penyedia layanan *ethernet*. Papan *ethernet* tersebut akan menjembatani Arduino agar dapat terhubung dengan jaringan LAN yang terkoneksi melalui *Router*.
6. Komputer, berfungsi untuk monitoring tempat parkir,

### 3.1.8. Analisis Komunikasi Data

Komunikasi data merupakan suatu hal yang sangat penting, karena tanpa adanya komunikasi data, suatu aplikasi yang akan dibangun tidak akan bias berjalan dengan baik atau secara optimal dapat di lihat pada gambar berikut :




**Gambar 3.8** Analisis Komunikasi Data

### 3.1.9. Analisis Kebutuhan Perangkat Keras

Perangkat keras atau *hardware* merupakan salah satu hal yang sangat penting dalam pembuatan atau pembangunan sebuah sistem. Sistem tersebut tidak akan dapat berjalan dengan baik apabila tidak didukung dengan komponen *hardware* yang memadai.

Berikut ini adalah spesifikasi perangkat keras yang digunakan untuk mendeteksi tempat parkir :

**Tabel 3.1** Tabel Spesifikasi Alat Pendeteksi

| Hardware | Detail |
|---|---|
| Arduino Uno | ATMega328 |
| Ethernet Shield | Wiznet W5100 |
| Sensor Ultrasonik | HC – SR04 |
| Router | TL-MR3020 |
| Modem | 4G-E3276 |
| LCD Display | 16x2 |
| Breadboard | 400 Tie Point |
| Sensor Infrared | E18-D50NK |

Berikut ini adalah spesifikasi perangkat keras server yang digunakan oleh pihak perusahaan :

**Tabel 3.2** Spesifikasi Komputer Server

| Hadrware | Detail |
|---|---|
| Processor | Intel Core 2 Duo |
| RAM | 2 GB |
| Harddisk | 250GB |
| VGA | On-Board |

Berikut ini adalah spesifikasi perangkat keras client yang digunakan oleh pihak perusahaan

**Tabel 3.3** Spesifikasi Komputer Client

| Hadrware | Detail |
|---|---|
| Processor | Intel Core 2 Duo |
| RAM | 2 GB |
| Harddisk | 250GB |
| VGA | On-Board |

Berikut ini adalah perangkat input output yang digunakan oleh pihak perusahaan

**Tabel 3.4** Perangkat Input Output

| No | Perangkat I/O |
|---|---|
| 1. | Mouse |
| 2. | Keyboard |
| 3. | Monitor |
| 4. | Printer |

Sedangkan perangkat keras (*Hardware*) yang di perlukan untuk membangun dan menjalankan sistem ini dapat dilihat pada tabel **3.4**

**Tabel 3.5** Spesifikasi yang dibutuhkan

| Hardware | Detail |
|---|---|
| Processor | Intel Core 2 Duo |
| RAM | 2 GB |
| Harddisk | 250GB |
| VGA | On-Board |

**Tabel 3.6** Perangkat Input / Output

| No | Perangkat I/O |
|---|---|
| 1. | Mouse |
| 2. | Keyboard |
| 3. | Monitor |
| 4. | Printer |

### 3.1.10. Analisis Kebutuhan Perangkat Lunak

Perangkat lunak (*software*) merupakan sebuah komponen penting pendamping perangkat keras (*hardware*). Secara umum *software* merupakan sekumpulan data-data elektronik yang tersimpan dan diatur oleh komputer yang berupa program atau instruksi untuk menjalankan dan mengeksekusi suatu

perintah. Perangkat lunak berfungsi sebagai saran interaksi antara pengguna dan perangkat keras.

Perangkat lunak yang dibutuhkan untuk membangun prototype ini adalah sebagai berikut :

**Tabel 3.7** Spesifikasi Software

| | |
|---|---|
| Sistem Operasi | Windows 7 (32 bit) |
| Tools Pemrograman | Notepad++<br>Arduino IDE |
| Bahasa Pemrograman | PHP<br>C/C++ |
| Web Server | XAMPP |
| Database | MySQL |

### 3.1.11. Analisis Pengguna

Analisis pengguna dimaksudkan untuk mengetahui siapa saja pengguna yang terlibat dalam sistem. Berikut adalah analisis pengguna dapat dilihat pada tabel berikut.

**Tabel 3.8** Analisis Pengguna

| Pengguna | Kemampuan |
|---|---|
| Petugas | Dapat mengoperasikan komputer |
| Pengendara | Dapat membaca navigasi yang di berikan sistem |

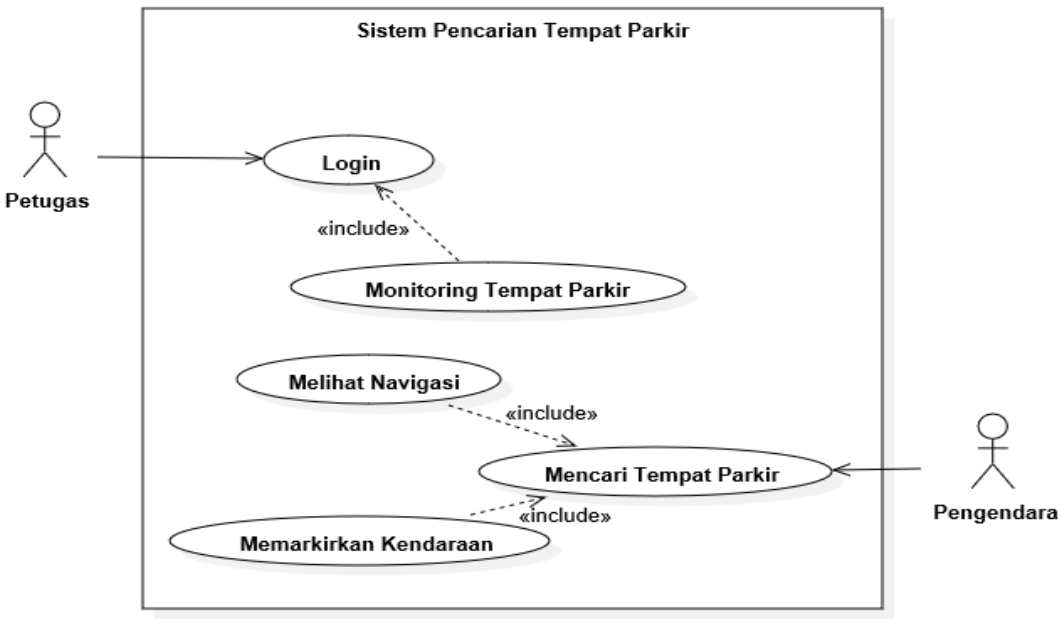
### 3.1.12. Analisis Kebutuhan Fungsional

Analisis kebutuhan fungsional berhubungan dengan fitur aplikasi yang akan dibuat atau dikembangkan. Berikut adalah tahapan analisis kebutuhan fungsional dari pembangunan prototype pencarian tempat parkir cerdas menggunakan mikrokontroller Arduino uno. Analisis kebutuhan fungsional menggambarkan proses kegiatan yang akan diterapkan dalam sebuah sistem dan menjelaskan kebutuhan yang diperlukan sistem agar sistem dapat berjalan dengan baik. Analisis yang akan dilakukan dimodelkan dengan menggunakan UML(Unified Modeling Language). Tahap-tahap pemodelan dalam analisis

tersebut antara lain identifikasi aktor, usecase diagram, scenario, sequence diagram, activity diagram, class diagram.

### 3.1.12.1. *Use Case Diagram*

*Use case diagram* merupakan pemodelan untuk kelakuan (*behavior*) sistem yang akan dibuat [9]. Untuk mendeskripsikan hubungan yang terjadi antar aktor dengan aktivitas yang terdapat pada sistem. Sasaran pada pemodelan *use case* diantaranya adalah mendefinisikan kebutuhan fungsional dan operasional sistem dengan mendefinisikan skenario penggunaan sistem yang akan dibangun. Dari hasil analisis aplikasi yang ada maka *use case* diagram untuk aplikasi ini dapat dilihat pada gambar berikut :




**Gambar 3.9** *Use Case Diagram* Sistem

### 3.1.12.2. Definisi Aktor

Definisi aktor berfungsi untuk menjelaskan aktor yang terdapat pada *Use Case Diagram*. Definisi aktor diterangkan pada Tabel 3.9

**Tabel 3.9** Definisi Aktor

| No | Aktor | Deskripsi |
|---|---|---|
| 1 | Petugas | Karyawan secure parking yang memonitoring tempat parkir |
| 2 | Pengendara | Mengikuti navigasi dan memarkirkan mobil di tempat yang telah di sediakan oleh sistem |

### 3.1.12.3. Definisi *Use Case*

Definisi use case untuk menjelaskan fungsi use case yang terdapat pada *Use Case Diagram*. Definisi use case dijelaskan pada Tabel 3.10

**Tabel 3.10** Definisi *Use Case*

| No | Use Case | Deskripsi |
|---|---|---|
| 1 | Login | Melakukan login di aplikasi pengelolaan data tempat parkir |
| 2 | Monitoring Tempat Parkir | Menampilkan Kondisi Tempat Parkir |
| 3 | Memarkirkan Mobil | User memarkirkan mobil sesuai tempat yang di lihat pada navigasi |
| 4 | Melihat navigasi | User melihat navigasi untuk mendapatkan tempat parkir |

### 3.1.12.4. *Use Case* Skenario

Use Case Skenario menggambarkan alur penggunaan sistem dimana setiap skenario yang di gambarkan dari sudut pandang aktor. Seseorang, atau piranti yang berinteraksi dengan perangkat lunak dalam berbagai cara

1. *Use Case* Skenario Login

   *Use Case* dari melakukan *login* awal dapat dilihat pada Tabel 3.11

**Tabel 3.11** *Use Case* Skenario Login

| Nama Use Case | Login |
|---|---|
| Nomor<br>Aktor | 1<br>Admin |
| Kondisi Awal | Sebelum Membuka Aplikasi |
| Skenario Utama | |
| Aksi Aktor | Reaksi Sistem |
| 1. Admin Membuka Aplikasi | |
| 3. Admin Memasukkan Username dan Password | 2. Tampil Halaman Login<br><br>4. Tampil Halaman Monitoring |

2. *Use Case* Skenario *Monitoring* Tempat Parkir

Use Case dari Skenario Monitoring tempat parkir dapat dilihat pada Tabel 3.12

**Tabel 3. 12** *Use Case* Skenario Monitoring Tempat Parkir

| Nama Use Case | Monitoring Tempat Parkir |
|---|---|
| Nomor<br>Aktor | 2<br>Admin |
| Kondisi Awal | Sebelum Membuka Aplikasi |
| Skenario Utama | |
| Aksi Aktor | Reaksi Sistem |
| 1. Admin Membuka Aplikasi | |
| 3. Admin Membuka menu Monitoring Tempat Parkir yang diinginkan | 2. Tampil Halaman Utama<br><br>4. Menampilkan Data Tempat Parkir |

3. *Use Case* Memarkirkan Mobil

*Use Case* skenario memarkirkan mobil dapat dilihat pada Tabel 3.13

**Tabel 3. 13** *Use Case* Skenario Memarkirkan Mobil

| Nama Use Case | Memarkirkan Mobil |
|---|---|
| Nomor<br>Aktor | 3<br>User |
| Kondisi Awal | Sebelum Melihat Navigasi |
| Skenario Utama | |
| Aksi Aktor | Reaksi Sistem |
| 1. User Memasukkan Mobil ke dalam gedung perparkiran | |
| 2. Melihat navigasi pada lcd display<br><br>4. Memarkirkan mobil ke tempat yang telah di arahkan oleh sistem | 3. Menampilkan navigasi pada lcd display |

4. *Use Case* Skenario Melihat Navigasi

*Use Case* Skenario Melihat Navigasi dapat dilihat pada Tabel 3.14

**Tabel 3. 14** *Use Case* Skenario Melihat Navigasi

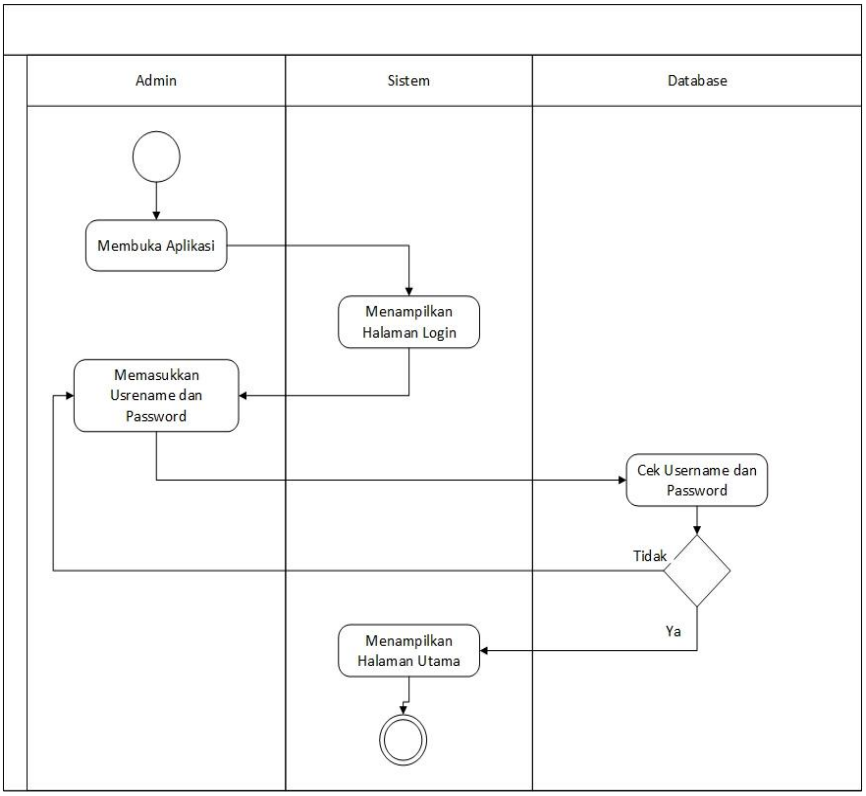
| | |
|---|---|
| Nama Use Case | Melihat Navigasi |
| Nomor<br>Aktor | 4<br>User |
| Kondisi Awal | Sebelum Memasuki Gedung Parkir |
| Skenario Utama | |
| Aksi Aktor | Reaksi Sistem |
| 1. User Memasukkan Mobil ke dalam gedung perparkiran | |
| 3. Melihat navigasi pada lcd display | 2. Menampilkan navigasi pada lcd display |

### 3.1.13. *Activity Diagram*

Diagram aktivitas atau *Activity Diagram* menggambarkan workflow (aliran kerja) atau aktivitas dari sebuah sistem atau proses bisnis. Penggambaran activity diagram memiliki kemiripan dengan flowchart diagram. Activity diagram memodelkan event-event yang terjadi pada use case dan digunakan untuk pemodelan aspek dinamis dari sistem.

1. *Activity Diagram* Login

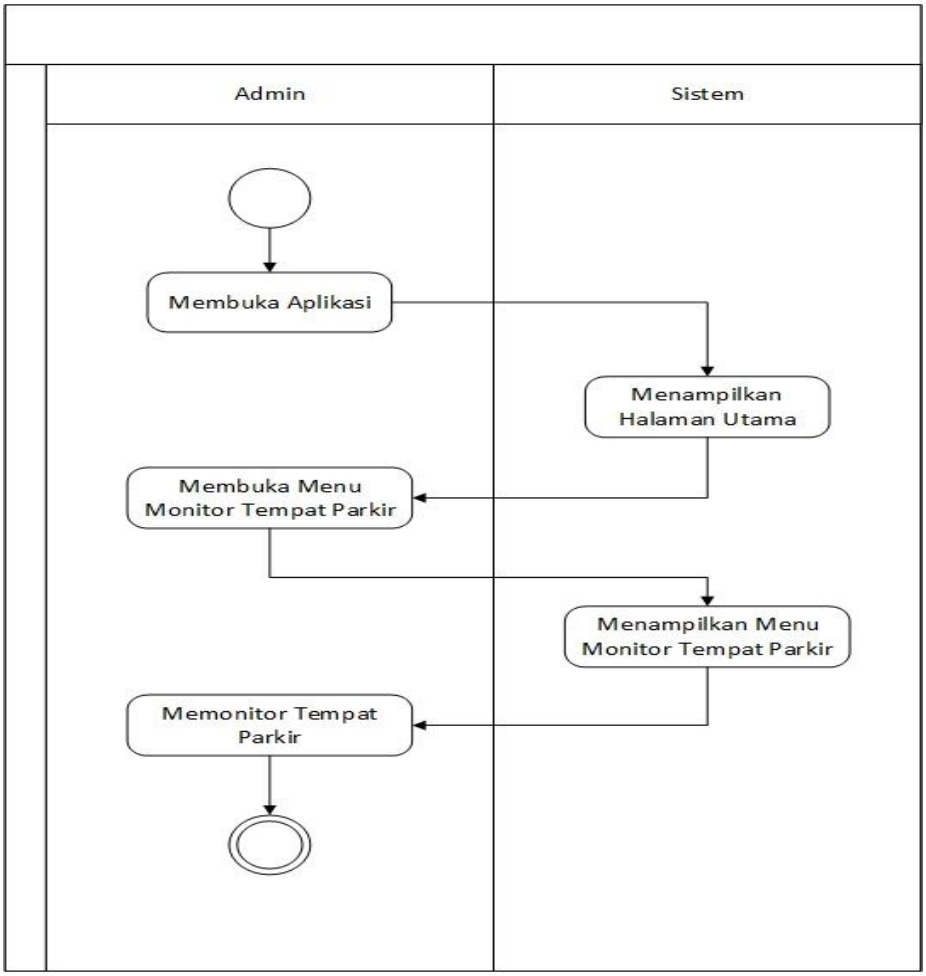
*Activity diagram* login menjelaskan alur kerja dan sistem dalam melakukan monitor status alat elektronik.




**Gambar 3.10** *Activity Diagram* Login

2. *Activity Diagram Monitoring* Tempat Parkir

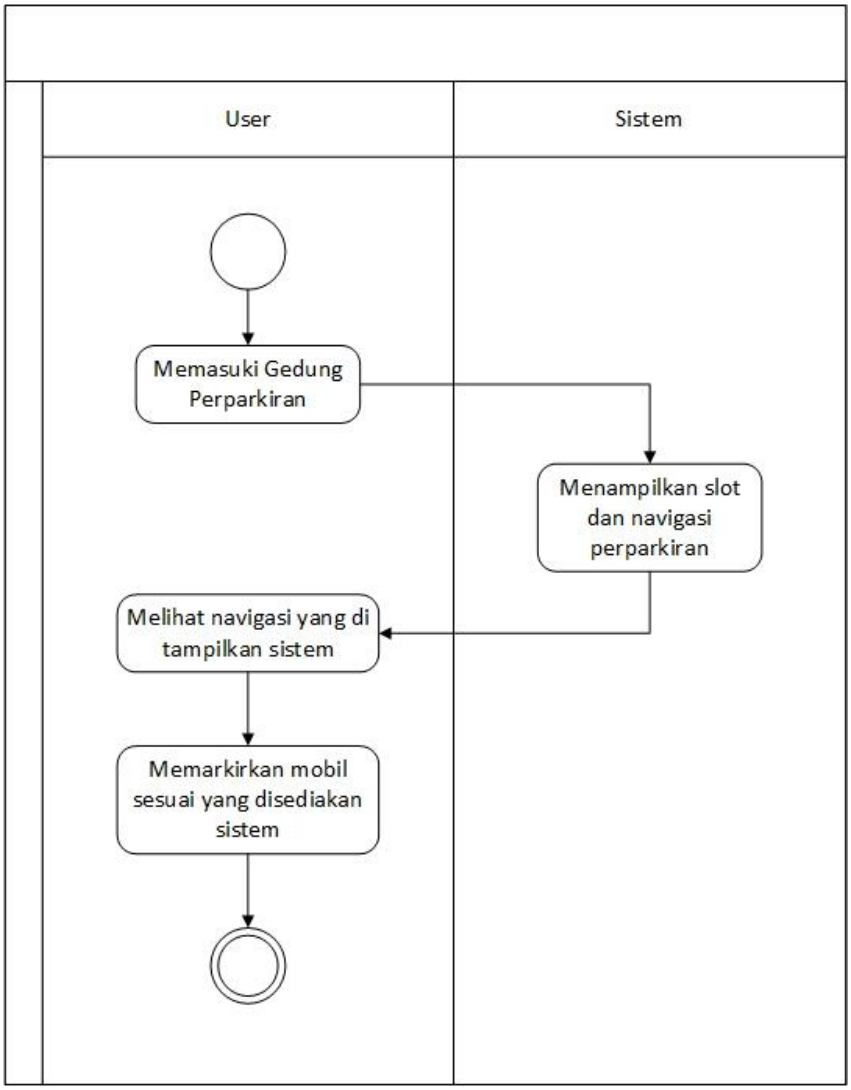
*Activity diagram monitoring* tempat parkir menjelaskan bagaimana alur kerja dan sistem dalam melakukan monitoring tempat parkir.




**Gambar 3.11** *Activity Diagram Monitoring* Tempat Parkir

3. *Activity Diagram* Memarkirkan Mobil

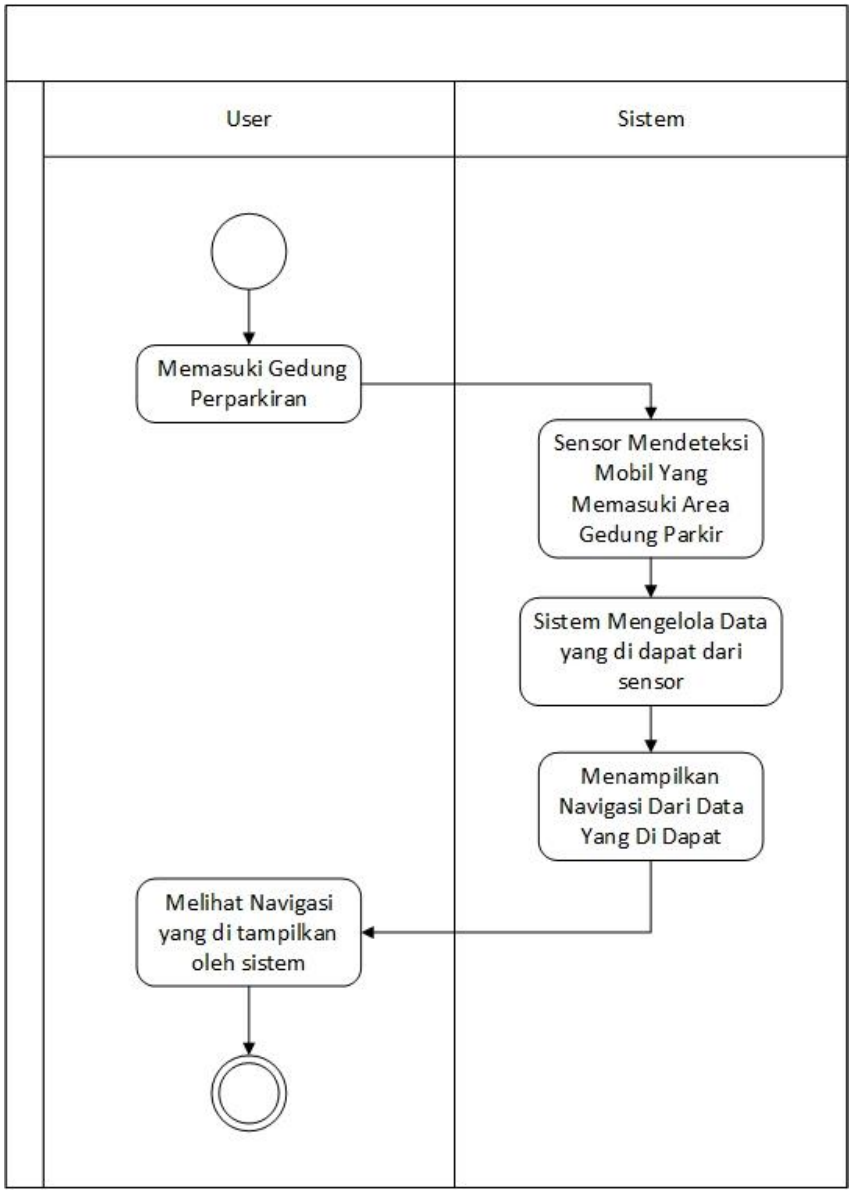
*Activity diagram* memarkirkan mobil menjelaskan bagaimana alur kerja dan sistem dalam memarkirkan mobil.




**Gambar 3.12** *Activity Diagram* Memarkirkan Mobil

4. *Activity Diagram* Melihat Navigasi

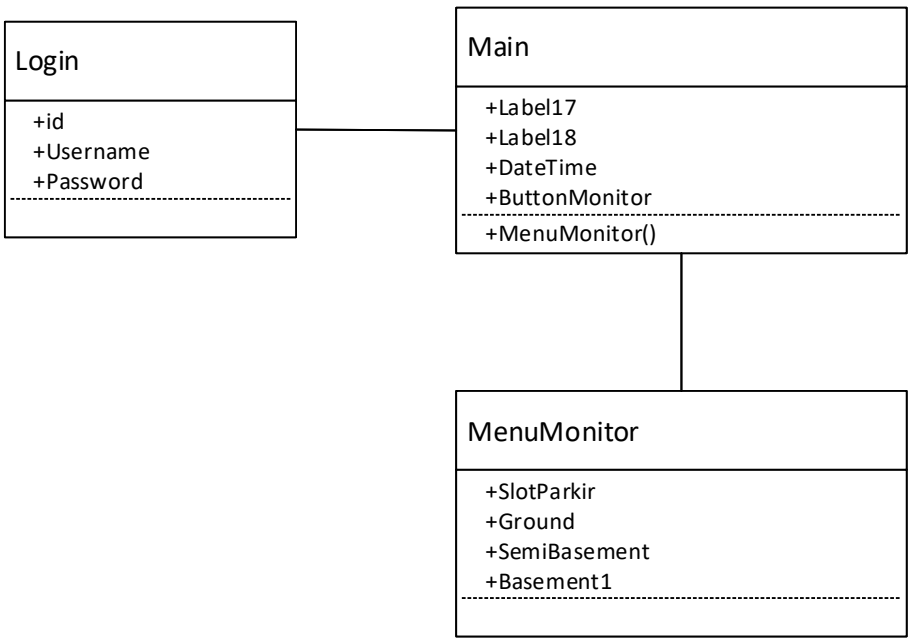
*Activity* diagram melihat navigasi menjelaskan bagaimana alur kerja dan sistem dalam melihat navigasi.




**Gambar 3.13** *Activity Diagram* Melihat Navigasi

### 3.1.14. *Class Diagram*

*Class Diagram* merupakan gambaran dari struktur dan hubungan pada setiap objek-objek yang berjalan pada sistem. Pada diagram ini digambarkan atribut-atribut dan metode-metode yang ada pada masing-masing kelas. Adapaun gambaran class diagram dari prototype pencarian tempat parkir cerdas menggunakan mikrokontroller Arduino uno yang dibangun dapat dilihat pada berikut.




**Gambar 3.14** *Class Diagram*

### 3.1.11.1 Definisi *Class*

*Definisi Class* berisi pemaparan setiap kelas yang terdapat pada *Class Diagram*. Berikut ini adalah penjelasan daftar kelas yang terdapat pada *class diagram* dapat dilihat pada Tabel 3.15

**Tabel 3. 15** Definisi *Class*

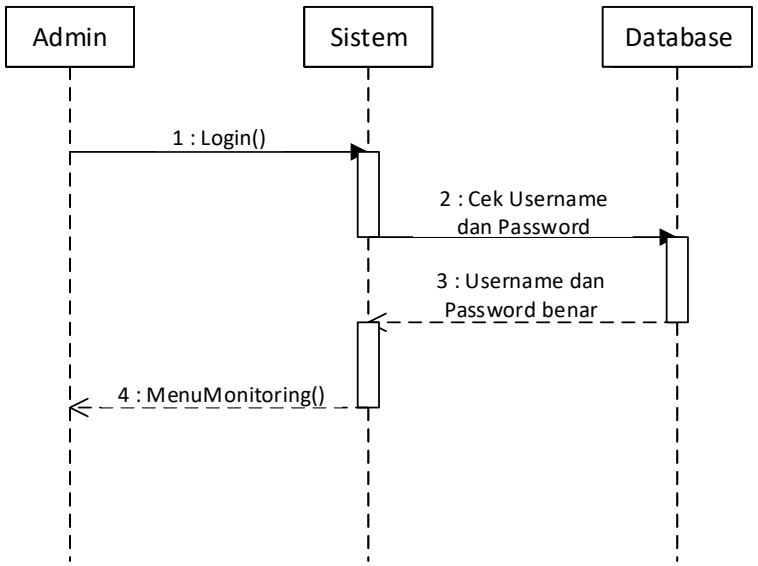
| Kelas | Deskripsi | Jenis |
|---|---|---|
| Login | Login digunakan untuk admin memasuki aplikasi | Interface |
| Main | Menu ini digunakan untuk masuk ke menu control dan menu monitor, menghubungkan koneksi, cek resolusi lcd | Interface |
| MenuMonitor | Menu ini jika dipilih akan menampilkan halaman untuk memonitor tempat parkir | Kontrol |

### 3.1.15. *Sequence Diagram*

Pada tahap ini *Sequence Diagram* menjelaskan secara detil urutan proses yang dilakukan sistem untuk mencapai tujuan dari *Use Case*. Interaksi yang terjadi antar *class*, operasi apa saja yang terlibat, urutan antar operasi, dan informasi yang diperlukan oleh masing-masing operasi.

1. *Sequence Diagram Login*

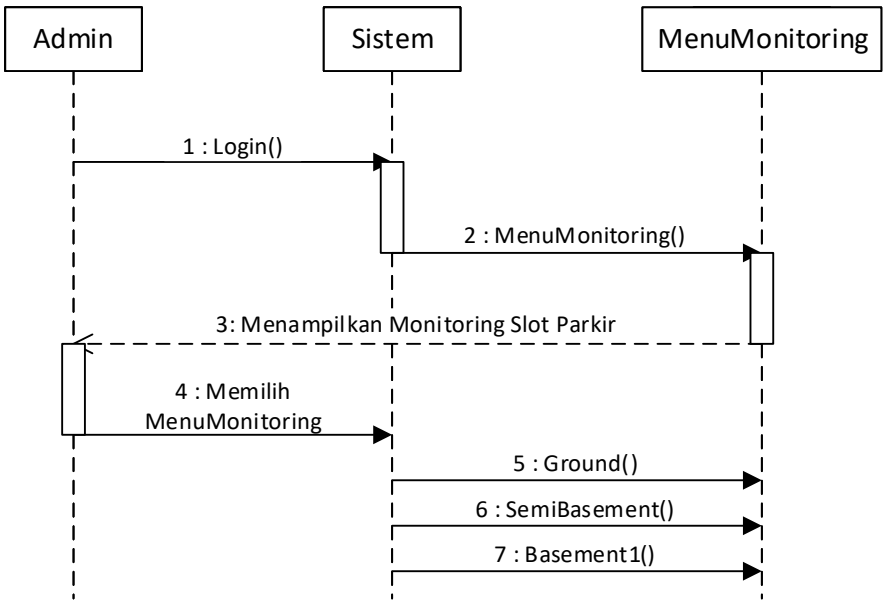
Sequence diagram melakukan login dari pembangunan purwarupa pencarian tempat parkir menggunakan mikrokontroller Arduino Uno ini dapat dilihat pada gambar berikut.




**Gambar 3.15** *Sequence Diagram Login*

2. *Sequence Diagram Monitoring* Slot Parkir

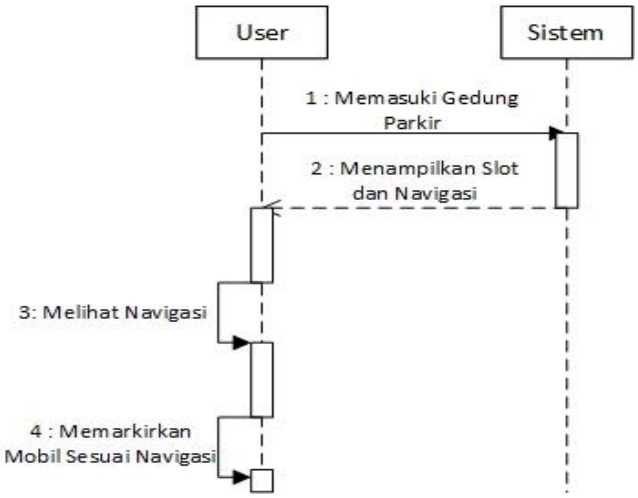
*Sequence Diagram* melakukan *monitoring* slot parkir pada pembangunan purwarupa pencarian tempat parkir menggunakan mikrokontroller Arduino Uno ini dapat dilihat pada gambar berikut.




**Gambar 3.16** *Sequence Diagram Monitoring* Slot Parkir

3. *Sequence Diagram* Memarkirkan Mobil

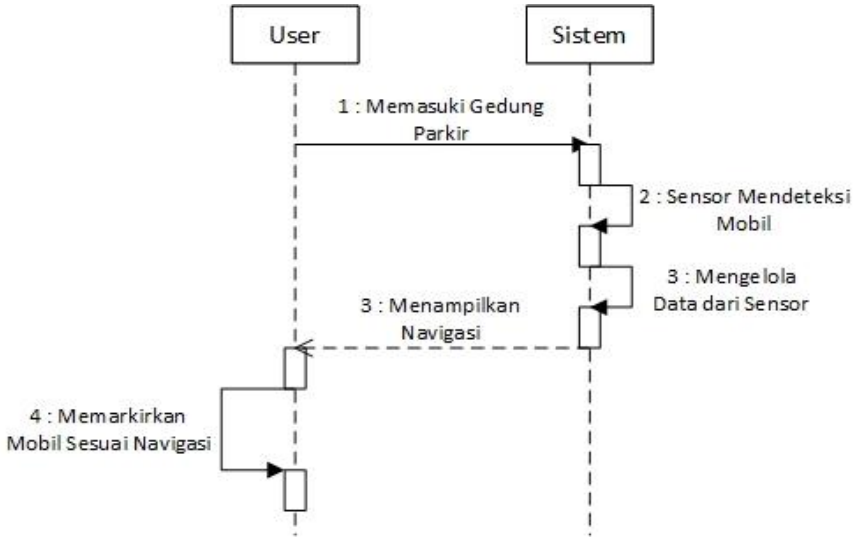
*Sequence Diagram* memarkirkan mobil pada pembangunan purwarupa pencarian tempat parkir menggunakan mikrokontroller Arduino Uno ini dapat dilihat pada gambar berikut.




**Gambar 3.17** *Sequence Diagram* Memarkirkan Mobil

4. *Sequence Diagram* Melihat Navigasi

*Sequence Diagram* melihat navigasi pada pembangunan purwarupa pencarian tempat parkir menggunakan mikrokontroller Arduino Uno ini dapat dilihat pada gambar berikut.



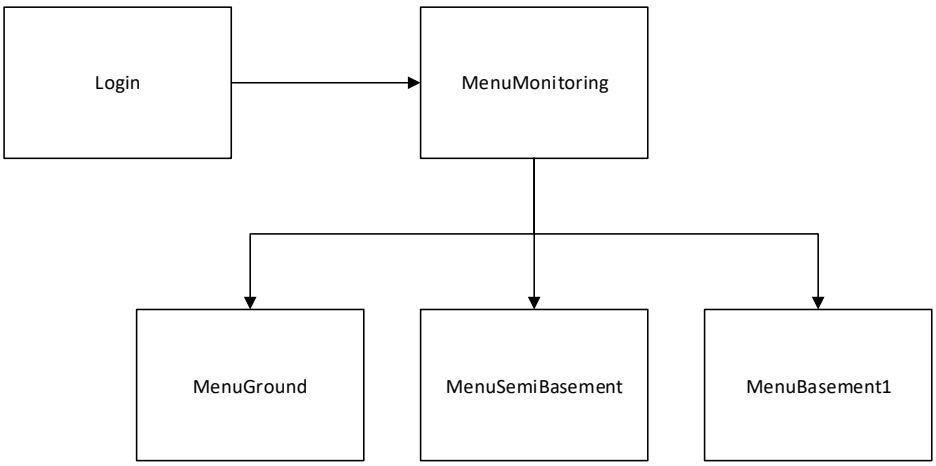

**Gambar 3.18** *Sequence Diagram* Melihat Navigasi

## 3.2. Perancangan Sistem

Perancangan arsitektur menu adalah perancangan dengan sebelumnya menggambarkan diagram alir sistem yang merupakan bagan yang menunjukan atas pekerjaan secara keseluruhan dari sistem yang akan dibuatkan menu pada sistem yang akan dibangun. Diagram ini menjelaskan urutan prosedur-prosedur yang ada dalam sistem. Diagram alir sistem menunjukkan apa yang dikerjakan oleh sistem.

### 3.2.1. Perancangan Struktur Menu

Proses perancangan struktur menu untuk sistem ini adalah sebagai berikut :



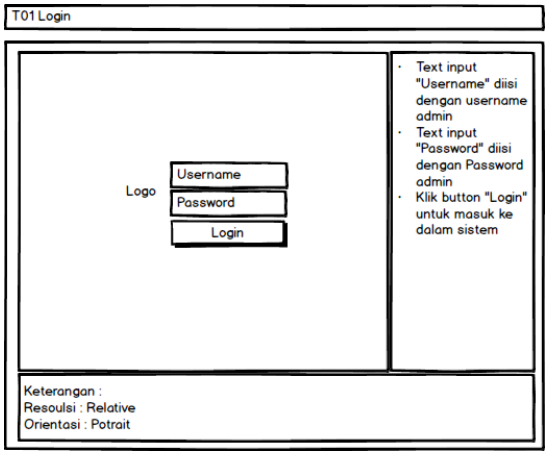

**Gambar 3.19** Struktur Menu

### 3.2.2. Perancangan AntarMuka

Bentuk Spesifikasi antarmuka merupakan suatu bentuk tampilan dari program yang akan dibuat untuk kebutuhan interface dengan *user*. Adapun perancangan antarmuka pada kontrol sistem yang akan dibangun adalah sebagai berikut :

1. Login (T01)

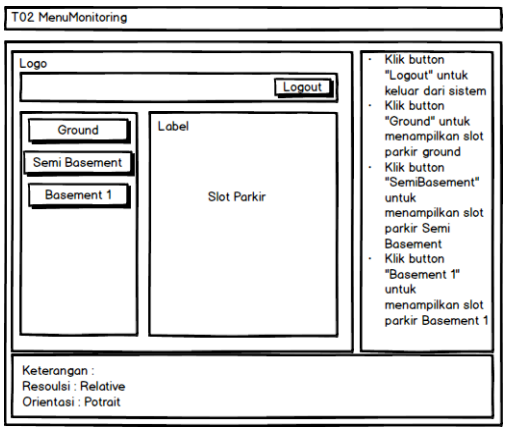
Merupakan tampilan utama saat memasuki aplikasi yang akan dibangun. *User* diharuskan mengisi username dan password untuk bias memasuki aplikasi dan ketika menekan tombol “LOGIN” maka *user* akan menuju ke menu monitoring.




**Gambar 3.20** Perancangan AntarMuka Login

2. Menu Monitoring (T02)

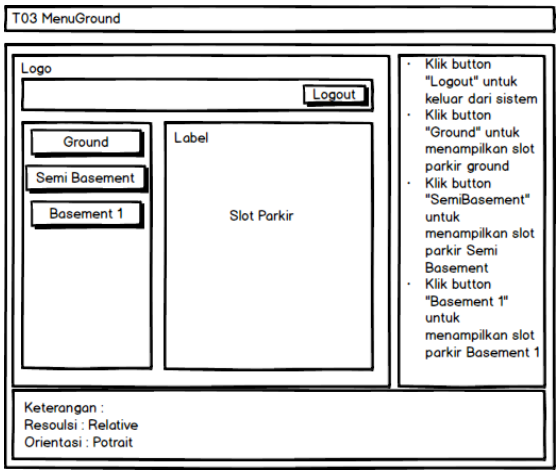
Di menu monitoring ini bias melihat slot parkir mana saja yang tersedia sesuai pada levelnya masing masing.




**Gambar 3.21** Perancangan AntarMuka Menu Monitoring

3. MenuGround (T03)

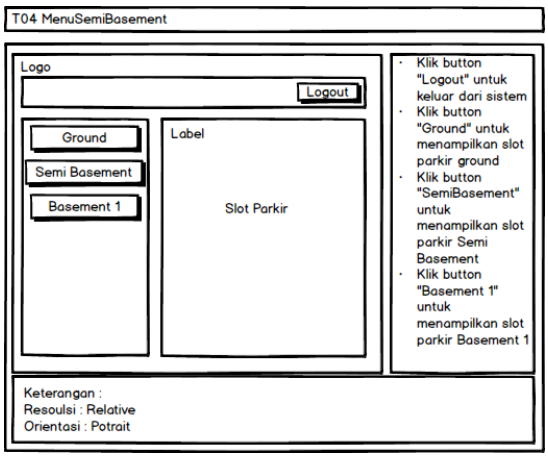
Di menu monitoring ini dapat melihat slot parkir mana saja yang tersedia pada level Ground.




**Gambar 3.22** Perancangan AntarMuka MenuGround

4. MenuSemiBasement (T04)

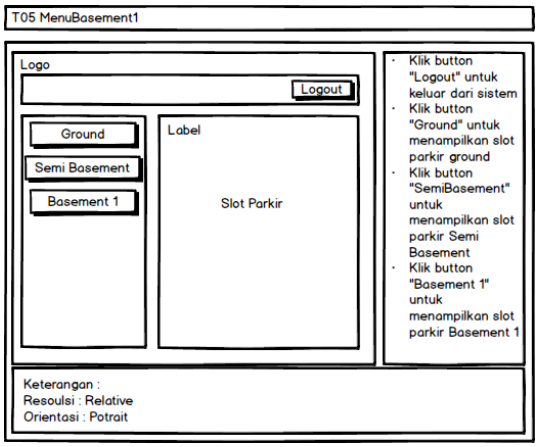
Di menu monitoring ini dapat melihat slot parkir mana saja yang tersedia pada level Semi Basement.




**Gambar 3.23** Perancangan AntarMuka MenuSemiBasement

5. MenuBasment1 (T05)

Di menu monitoring ini dapat melihat slot parkir mana saja yang tersedia pada level Basement 1.

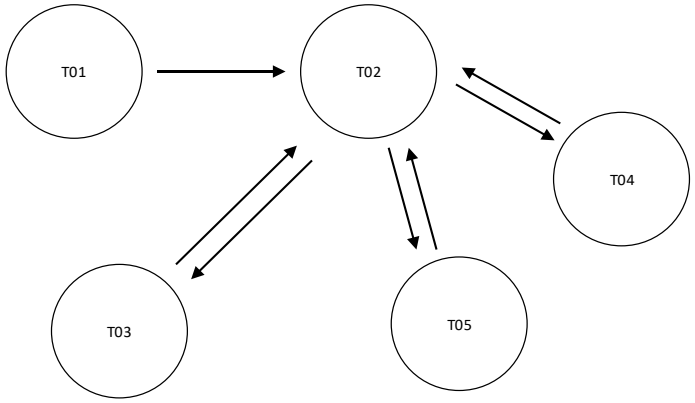



**Gambar 3.24** Perancangan Antarmuka MenuBasement1

### 3.2.3. Jaringan Semantik

Tahap selanjutnya yaitu membuat jaringan semantik, yaitu penggambaran grafis yang menunjukkan hubungan antar berbagai objek. Adapun jaringan

semantik yang menggambarkan kontrol sistem yang akan dibangun adalah sebagai berikut :




**Gambar 3.25** Jaringan Semantik