**BAB III**

**PEMBAHASAN**

Pada bab ini penulis mencoba menguraikan serta menjelaskan tentang perancangan dan pembuatan aplikasi antrian loket rumah sakit yang dibagi ke dalam beberapa tahapan. Tahapan-tahapan tersebut dikerjakan dan dibuat secara berurutan dan berkesinambungan supaya hasilnya dapat lebih baik dan lebih efisien. Proses pembuatan aplikasi ini melalui beberapa tahapan :

1. Analisis
2. Use Case Diagram
3. Struktur Navigasi
4. Rancangan Tampilan
5. Rancangan Struktur Database
6. Implementasi
7. Pengujian Aplikasi

# Analisis

# Pada pembuatan aplikasi, diperlukan analisa sistem. Analisa sistem adalah penguraian dari suatu sistem yang utuh kedalam bagian-bagian komponennya dengan maksud untuk mengidentifikasi dan mengevaluasi permasalahan. Mengidentifikasi (mengenal) masalah merupakan langkah pertama yang dilakukan dalam tahap analisis sistem. Masalah dapat didefinisikan sebagai suatu pertanyaan yang ingin dipecahkan. Sama halnya dalam pembuatan aplikasi ini. Penulis memilih informasi mengenai antrian di suatu rumah sakit swasta untuk mempermudah pihak rumah sakit dalam melayani pasien dalam hal pendaftaran dan validasi data. Pada bagian analisis, terdiri dari analisis fungsional dan analisis permasalahan resource.

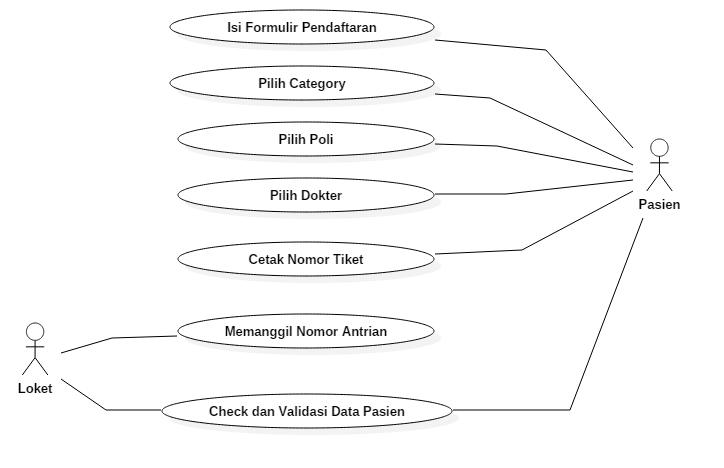
# Analisis Fungsional

Analisis fungsional merupakan paparan mengenai fitur-fitur yang akan dimasukkan kedalam aplikasi ini. Fitur-fitur tersebut antara lain pengaturan kuota dokter, pendaftaran pasien, print tiket, pemanggilan suara berdasarkan nomor tiket, validasi data pasien.

# Analisis Permasalahan Resource (Sumber Daya)

Dikarnakan aplikasi yang akan di buat adalah webapp dengan menggunakan javascript pada bagian interface, maka untuk mempercepat load resource sangat di perlukan penempatan file javascript di akhir file index, dan meminimkan pemakaian gambar.

# Use Case Diagram

 Gambar 3.1 Use Case Diagram

Use Case Diagram digunakan untuk menggambarkan secara ringkas pengguna system dan apa saja yang dapat dilakukannya. Melalui diagram Use Case dapat diketahui fungsi – fungsi apa saja yang ada didalam system.

Pada gambar diagram diatas diketahui terdapat 2 aktor utama yang menjalankan system antrian tersebut, yaitu pasien dan loket. Berikut adalah penjelasan process alur system dari diagram diatas :

* + - 1. Pasien mengisi form pendaftaran jika belum terdaftar dalam system tersebut.
      2. Pasien memilih kategori loket farmasi (pengambilan obat), regular, dan bpjs.
      3. Pasien memilih poli.
      4. Pasien memilih dokter.
      5. Pasien mencetak dan mendapatkan bukti nomor antrian.
      6. Loket memanggil nomor antrian pasien secara berurutan.
      7. Loket dan pasien melakukan validasi data pasien

# Struktur Navigasi

Struktur navigasi dapat dikatakan sebagai penggambar dari hubungan atau rantai kerja dari seluruh element yang akan digunakan dalam aplikasi. Dalam Pembuatan aplikasi ini, struktur navigasi sangat diperlukan untuk menggambarkan setiap menu dan halaman yang akan ditampilkan. Pada Struktur Navigasi aplikasi ini setiap user harus melewati proses login terlebih dahulu agar terhindar dari perbuatan yang tidak bertanggung jawab.

# Struktur Navigasi Admin

# 

Gambar 3.2 Struktur Navigasi Admin

Struktur Navigasi Admin digunakan digunakan untuk menggambarkan hubungan antara menu yang terdapat pada tampilan admin. Pada tampilan admin hanya berisikan halaman – halaman yang penunjang berjalannya system. Struktur Navigasi Admin ini terdapat 4 menu utama yaitu :

* + - 1. Menu System yaitu menu yang digunakan untuk mengatur user, role dan permission
      2. Menu Master yaitu menu yang digunakan untuk menginput data yang akan digunakan pada proses system, seperti kategori, poli, kuota dan loket.
      3. Menu BOD (Begin Of Day) digunakan untuk memulai suatu system sebagai tanda libur/tidaknya rumah sakit tersebut dan kuota yang diberikan pada hari tesebut.
      4. Menu Report yaitu menu yang digunakan untuk melihat hasil transaksi seperti laporan kuota dan nomor antrian berdasarkan tanggal transaksi.

# Struktur Navigasi Pasien/User

# 

Gambar 3.3 Struktur Navigasi User

Struktur Navigasi User digunakan untuk menggambarkan alur pendaftaran hingga user mendapat nomor antrian. Pada struktur navigasi ini proses login dilakukan oleh petugas rumah sakit, setelah login petugas mempersilahkan user untuk mendaftar jika belum terdaftar, kemudian user melanjutkan memilih kategori, poli dan dokter hingga mendapat nomor antrian.

# Struktur Navigasi Loket

# 

Gambar 3.4 Struktur Navigasi Loket

Struktur Navigasi Loket terdapat menu loket yang merupakan tampilan dari form antrian dan data user yang sedang diproses.

# Struktur Navigasi Information Screen

# 

Gambar 3.5 Struktur Navigasi Information Screen

Pada struktur navigasi ini login dilakukan oleh petugas rumah sakit dan halaman pada menu information screen di tampilkan pada monitor atau tv yang di tempatkan pada ruang tunggu user agar dapat dilihat oleh semua user/pasien.

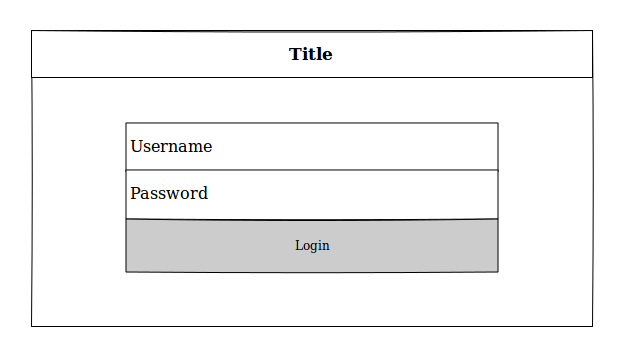
* 1. **Racangan Tampilan**

Rancangan tampilan aplikasi merupakan hal yang sangat penting untuk kemudahan para pengguna saat berinteraksi dengan sebuah aplikasi dan kemudahan mendapatkan informasi yang dibutuhkan dengan cepat dan efisien.

Dengan memberikan deskripsi ini, penulis berharap sebelum tahap penjelasan proses pembuatan, aplikasi ini sudah dapat dipandang secara utuh dengan kata lain dapat dipahami secara jelas apa yang akan dibahas pada proses pembuatan aplikasi ini.

* + 1. **Rancangan Tampilan Login**

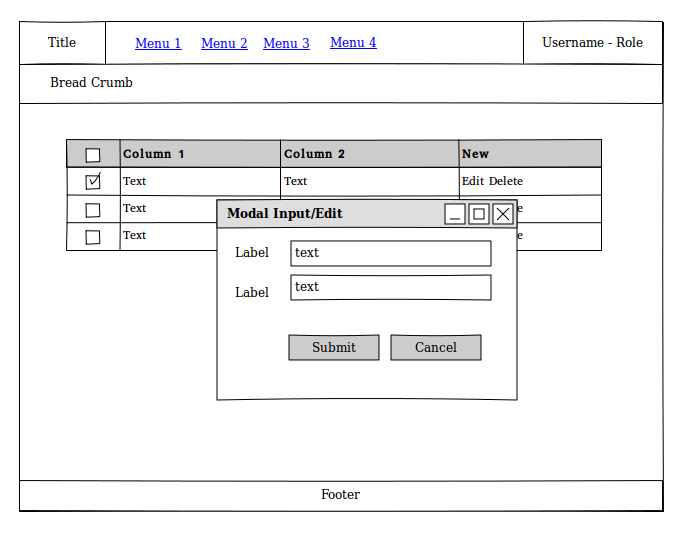
Berikut ini adalah rancangan halaman login yang merupakan halaman yang pertama kali di akses sebelum user masuk ke halaman lainnya. Pada halaman ini terdapat 2 input yang harus diisi yaitu username dan password.



Gambar 3.6 Rancangan Tampilan Login

# Rancangan Tampilan System Dan Master

Berikut ini adalah rancangan tampilan untuk semua halaman system dan master. Pada halaman ini hanya dapat diakses oleh admin dan dikhususkan untuk CRUD data sebagai penunjang aplikasi, seperti pada gambar 3.6 terdapat table untuk menampilkan data, form input/edit di dalam modal dan delete data.



Gambar 3.7 Rancangan Tampilan Master

# Rancangan Tampilan BOD (Begin Of Day)

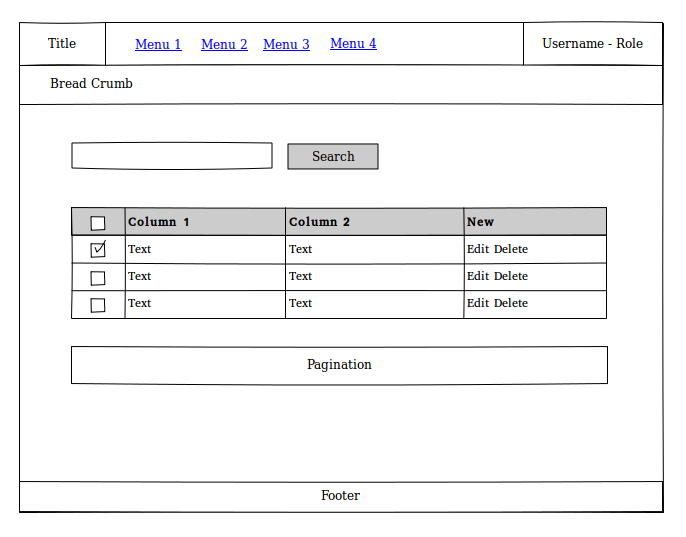
# Berikut ini adalah rancangan tampilan dari menu BOD yang digunakan admin untuk generate kuota dari table dokte. Pada rancangan BOD hanya terdapat 1 button utama untuk memanggil function BOD.

# 

Gambar 3.8 Rancangan Tamapilan BOD

# Rancangan Halaman Report/Laporan

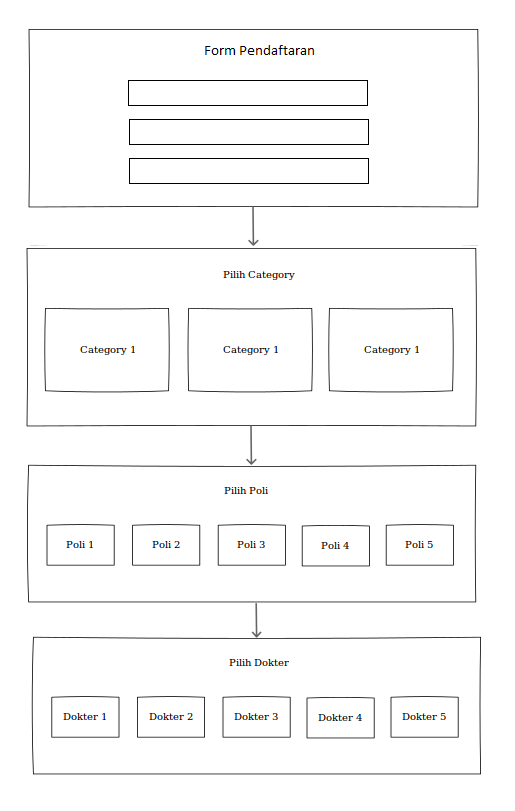
Berikut ini adalah rancangan tampilan dari menu laporan, pada rancangan ini terdapat menu search sebagai filter data yang akan ditampilkan, lalu di bawahnya terdapat table sebagai tampilan dari list data, kemudian di bawah table terdapat pagination yang berfungsi membagi data menjadi beberapa halaman.



Gambar 3.9 Rancangan Tampilan Report

# Rancangan Tampilan Pendaftaran

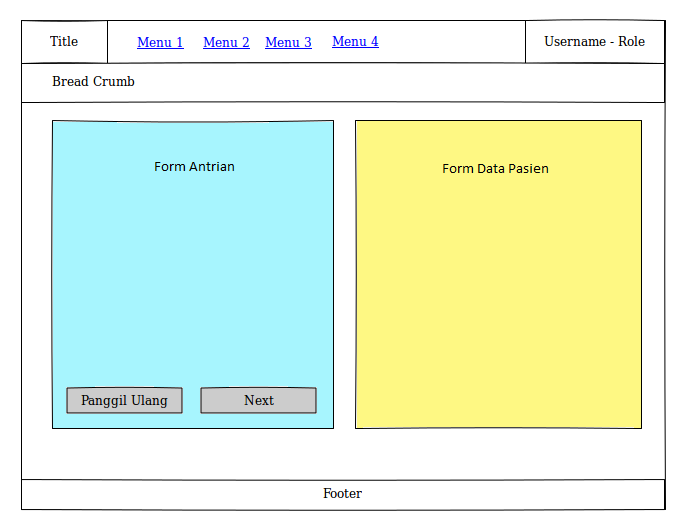
Berikut ini adalah rancangan tampilan halaman-halaman hingga pasien mendapat nomor antrian. Pertama pasien melakukan pendaftaran melalui form yang tersedia, jika pasien sudah terdaftar atau telah mengisi form pendaftaran halaman akan berpindah ke halaman pilih kategori antrian, setelah memilih kategori pasien berpindah ke halaman pilih poli dan terakhir setelah memilih poli pasen berpindah ke halaman pilih dokter dan mendapatkan hasil print nomor antrian.



Gambar 3.10 Rancangan Tampilan Pendaftaran

# Rancangan Tampilan Loket

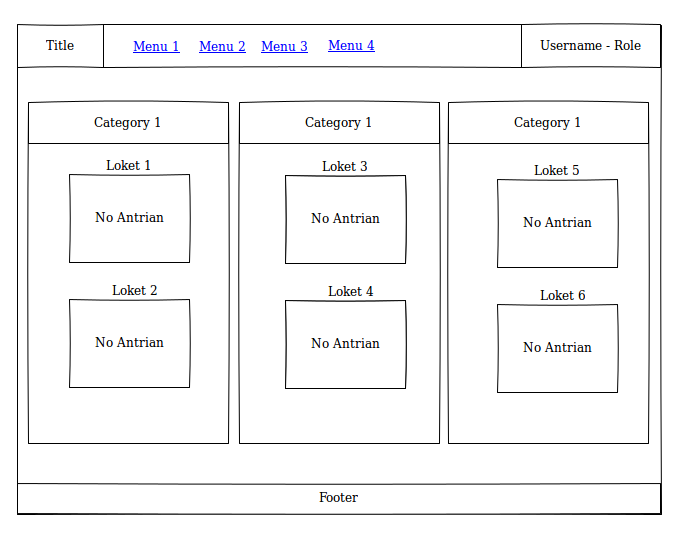
Berikut ini adalah rancangan tampilan halaman transaksi loket. Pada halaman ini di bagi menjadi form antrian dan form data pasien. Form antrian digunakan untuk menampilkan informasi nomor antrian yang akan di proses sedangkan form data pasien digunakan untuk menampilkan informasi data pasien yang sedang di proses.



Gambar 3.11 Rancangan Tampilan Loket

# Rancangan Tampilan Information Screen

Berikut ini adalah rancangan tampilan yang digunakan sebagai sarana untuk menampilkan nomor antrian yang sedang diproses oleh loket-loket.



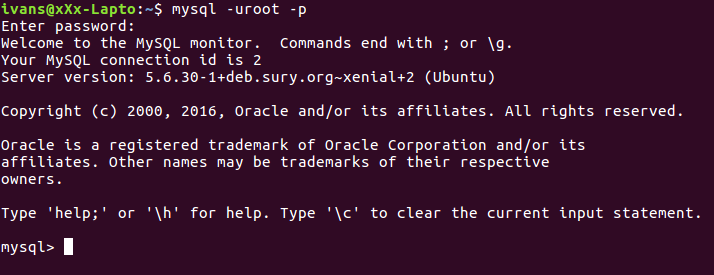
Gambar 3.12 Rancangan Tampilan Information Screen

# Perancangan Struktur Database

Perancangan database merupakan proses untuk membuat suatu database sebagai penampung data yang akan digunakan oleh aplikasi. DBMS yang akan digunakan oleh penulis adalah MySql. Dalam perancangan database kali ini penulis akan membuat schema database yang nantinya akan disimpan dalam aplikasi.

# Membuat Database

Langkah pertama untuk membuat database kita harus masuk kedalam console MySql dengan perintah seperti gambar di bawah ini :



Gambar 3.13 Perintah MySql

Kemudian buat database baru dengan nama antrian\_development, Berikut merupakan perintah untuk membuat database baru pada MySql :

**create database antrian\_development;**

# Membuat Schema Dan Struktur Database

Schema adalah sekumpulan dari object-object database. Penulis tidak membuat struktur dan objek database langsung melalui console MySql melainkan menulis semua query kedalam suatu file yang nantinya akan dijalankan saat aplikasi pertama kali di running, tujuannya adalah mempermudah pengguna aplikasi saat memindahkan program kedalam tempat lain tanpa harus membuat struktur database lagi.

Berikut merupakan query untuk membuat table :

* + - 1. **Table permission**

CREATE TABLE IF NOT EXISTS `c\_security\_permission` (

`id` varchar(255) NOT NULL,

`permission\_label` varchar(255) NOT NULL,

`permission\_value` varchar(255) NOT NULL,

PRIMARY KEY (`id`),

UNIQUE KEY `UK\_k4suda9cvcsoikdgquscypmt6` (`permission\_value`)

) ENGINE=InnoDB DEFAULT CHARSET=latin1;

* + - 1. **Table role**

CREATE TABLE IF NOT EXISTS `c\_security\_role` (

`id` varchar(255) NOT NULL,

`description` varchar(255) DEFAULT NULL,

`name` varchar(255) NOT NULL,

PRIMARY KEY (`id`),

UNIQUE KEY `UK\_hliaoojt6u3a11d8svttju10l` (`name`)

) ENGINE=InnoDB DEFAULT CHARSET=latin1;

* + - 1. **Table relasi role dan permission**

CREATE TABLE IF NOT EXISTS `c\_security\_role\_permission` (

`id\_role` varchar(255) NOT NULL,

`id\_permission` varchar(255) NOT NULL,

PRIMARY KEY (`id\_role`,`id\_permission`),

KEY `FK\_d89p0a0x87scb5s3830jx7xq0` (`id\_permission`),

CONSTRAINT `FK\_fvynt2q4rxk27e0bxuon50tp4` FOREIGN KEY (`id\_role`) REFERENCES `c\_security\_role` (`id`),

CONSTRAINT `FK\_d89p0a0x87scb5s3830jx7xq0` FOREIGN KEY (`id\_permission`) REFERENCES `c\_security\_permission` (`id`)

) ENGINE=InnoDB DEFAULT CHARSET=latin1;

* + - 1. **Table master kategori**

CREATE TABLE IF NOT EXISTS `m\_category` (

`id` varchar(255) NOT NULL,

`code` varchar(255) NOT NULL,

`name` varchar(255) NOT NULL,

`title` varchar(255) NOT NULL,

PRIMARY KEY (`id`),

UNIQUE KEY `UK\_805u58snn6h9tr2jcao7libdu` (`code`)

) ENGINE=InnoDB DEFAULT CHARSET=latin1;

* + - 1. **Table master loket**

CREATE TABLE IF NOT EXISTS `m\_loket` (

`id` varchar(255) NOT NULL,

`nomor\_loket` int(11) NOT NULL,

`id\_category` varchar(255) NOT NULL,

PRIMARY KEY (`id`),

UNIQUE KEY `UK\_kdw3h16608yjtegmqu0ud4hf5` (`nomor\_loket`),

KEY `FK\_bfdtpow629veh96jl7dpgn7ag` (`id\_category`),

CONSTRAINT `FK\_bfdtpow629veh96jl7dpgn7ag` FOREIGN KEY (`id\_category`) REFERENCES `m\_category` (`id`)

) ENGINE=InnoDB DEFAULT CHARSET=latin1;

* + - 1. **Table user**

CREATE TABLE IF NOT EXISTS `c\_security\_user` (

`id` varchar(255) NOT NULL,

`active` bit(1) NOT NULL,

`address` varchar(255) NOT NULL,

`email` varchar(255) NOT NULL,

`fullname` varchar(255) NOT NULL,

`phone` varchar(255) NOT NULL,

`username` varchar(255) NOT NULL,

`id\_loket` varchar(255) DEFAULT NULL,

`id\_role` varchar(255) NOT NULL,

PRIMARY KEY (`id`),

UNIQUE KEY `UK\_at8if7a9lnl90wxllb9divpdf` (`username`),

KEY `FK\_oi1dd3h3m0m8p2xqq6ki23gh3` (`id\_loket`),

KEY `FK\_my18sie96bgbncypva3fxboxy` (`id\_role`),

CONSTRAINT `FK\_my18sie96bgbncypva3fxboxy` FOREIGN KEY (`id\_role`) REFERENCES `c\_security\_role` (`id`),

CONSTRAINT `FK\_oi1dd3h3m0m8p2xqq6ki23gh3` FOREIGN KEY (`id\_loket`) REFERENCES `m\_loket` (`id`)

) ENGINE=InnoDB DEFAULT CHARSET=latin1;

* + - 1. **Table user password**

CREATE TABLE IF NOT EXISTS `c\_security\_user\_password` (

`id\_user` varchar(255) NOT NULL,

`user\_password` varchar(255) NOT NULL,

PRIMARY KEY (`id\_user`),

CONSTRAINT `FK\_9a26m4sjx4ddi35n3w0s6b5os` FOREIGN KEY (`id\_user`) REFERENCES `c\_security\_user` (`id`)

) ENGINE=InnoDB DEFAULT CHARSET=latin1;

* + - 1. **Table master poli**

CREATE TABLE IF NOT EXISTS `m\_poli` (

`id` varchar(255) NOT NULL,

`code` varchar(255) NOT NULL,

`name` varchar(255) NOT NULL,

PRIMARY KEY (`id`),

UNIQUE KEY `UK\_qy81ge6f8xaceyhpyymvdhsyp` (`code`)

) ENGINE=InnoDB DEFAULT CHARSET=latin1;

* + - 1. **Table running number**

CREATE TABLE IF NOT EXISTS `runnum` (

`id` varchar(255) NOT NULL,

`run\_date` varchar(255) NOT NULL,

`lastnumber` bigint(20) NOT NULL,

`run\_month` varchar(255) NOT NULL,

`run\_prefix` varchar(255) NOT NULL,

`type` varchar(255) NOT NULL,

`run\_year` varchar(255) NOT NULL,

PRIMARY KEY (`id`),

UNIQUE KEY `UK\_qtnff32gosvs4k8s6by1jqas7` (`run\_date`,`run\_month`,`run\_year`,`run\_prefix`)

) ENGINE=InnoDB DEFAULT CHARSET=latin1;

* + - 1. **Table jadwal dan kuota dokter**

CREATE TABLE IF NOT EXISTS `m\_jadwal\_dokter` (

`id` varchar(255) NOT NULL,

`kuota\_jumat` int(11) NOT NULL,

`kuota\_kamis` int(11) NOT NULL,

`kuota\_minggu` int(11) NOT NULL,

`kuota\_rabu` int(11) NOT NULL,

`kuota\_sabtu` int(11) NOT NULL,

`kuota\_selasa` int(11) NOT NULL,

`kuota\_senin` int(11) NOT NULL,

`name` varchar(255) NOT NULL,

`description` varchar(255) DEFAULT NULL,

`code` varchar(255) NOT NULL,

`id\_poli` varchar(255) NOT NULL,

PRIMARY KEY (`id`),

UNIQUE KEY `UK\_itaena7o6t8teu8ry8a3qo6g7` (`code`),

KEY `FK\_sg5solicf7m5kw4jmx430mw06` (`id\_poli`),

CONSTRAINT `FK\_sg5solicf7m5kw4jmx430mw06` FOREIGN KEY (`id\_poli`) REFERENCES `m\_poli` (`id`)

) ENGINE=InnoDB DEFAULT CHARSET=latin1;

* + - 1. **Table transaksi hasil bod kuota**

CREATE TABLE IF NOT EXISTS `t\_kuota` (

`id` varchar(255) NOT NULL,

`current\_kuota` int(11) NOT NULL,

`kuota\_date` date NOT NULL,

`maximum\_kuota` int(11) NOT NULL,

`nama\_dokter` varchar(255) NOT NULL,

`description\_dokter` varchar(255) NOT NULL,

`code\_dokter` varchar(255) NOT NULL,

`id\_poli` varchar(255) NOT NULL,

PRIMARY KEY (`id`),

KEY `FK\_cedsi46x2ngu7rc9g47s4rvsa` (`id\_poli`),

CONSTRAINT `FK\_cedsi46x2ngu7rc9g47s4rvsa` FOREIGN KEY (`id\_poli`) REFERENCES `m\_poli` (`id`)

) ENGINE=InnoDB DEFAULT CHARSET=latin1;

* + - 1. **Table laporan antrian**

CREATE TABLE IF NOT EXISTS `t\_log\_antrian` (

`id` varchar(255) NOT NULL,

`antrian\_date` varchar(255) NOT NULL,

`timestamp` datetime NOT NULL,

`jenis\_loket` varchar(255) NOT NULL,

`nomor\_antrian` varchar(255) NOT NULL,

`nomor\_loket` int(11) DEFAULT NULL,

`status` bit(1) NOT NULL,

`updated\_date` datetime DEFAULT NULL,

`id\_kuota` varchar(255),

PRIMARY KEY (`id`),

UNIQUE KEY `UK\_fkkhvtkmwjybkpef32flp4nli` (`nomor\_antrian`),

KEY `FK\_jrrcfni173ooo9dg24ha9w6hs` (`id\_kuota`),

CONSTRAINT `FK\_jrrcfni173ooo9dg24ha9w6hs` FOREIGN KEY (`id\_kuota`) REFERENCES `t\_kuota` (`id`)

) ENGINE=InnoDB DEFAULT CHARSET=latin1;

* + - 1. **Table transaksi pemanggilan antrian**

CREATE TABLE IF NOT EXISTS `t\_log\_antrian\_sound` (

`id` varchar(255) NOT NULL,

`nomor\_antrian` varchar(255) NOT NULL,

`status` bit(1) NOT NULL,

`terbilang` varchar(255) NOT NULL,

`timestamp` datetime NOT NULL,

PRIMARY KEY (`id`)

) ENGINE=InnoDB DEFAULT CHARSET=latin1;

* + - 1. **Table pendaftaran pasien**

CREATE TABLE IF NOT EXISTS `t\_pasien` (

`id` varchar(255) NOT NULL,

`address` varchar(255) NOT NULL,

`blood\_type` varchar(255) NOT NULL,

`date\_of\_birth` date NOT NULL,

`height` int(11) NOT NULL,

`identity\_number` varchar(255) NOT NULL,

`identity\_type` varchar(255) NOT NULL,

`name` varchar(255) NOT NULL,

`no\_pasien` varchar(255) NOT NULL,

`phone` varchar(255) NOT NULL,

`updated\_date` date NOT NULL,

`weight` int(11) NOT NULL,

`verify` bit(1) NOT NULL,

PRIMARY KEY (`id`),

UNIQUE KEY `UK\_k4dckuj5tynq9x1bq70kwpjtd` (`identity\_number`),

UNIQUE KEY `UK\_3jcb93l2vg225lcqxty99kpv4` (`no\_pasien`)

) ENGINE=InnoDB DEFAULT CHARSET=latin1;

# Membuat DB Migration

Flyway merupakan library java yang digunakan sebagai DB migration dan untuk menjalankan schema database yang sudah di buat. Flyway secara otomatis akan membuat tabel schema\_version di database yang sedang dipakai. Bila aplikasi memiliki file SQL dengan versi yang lebih tinggi dari versi yang tercatat di database yang sedang dipakai, maka Flyway secara otomatis akan mengexecute file SQL tersebut. Dengan demikian, selama setiap perubahan database di-“catat” ke dalam file SQL, maka perubahan ini akan diterapkan secara konsisten pada data. Gunakan perintah berikut untuk menggunakan plugin flyway db melalui maven xml :

<plugin>

<groupId>org.flywaydb</groupId>

<artifactId>flyway-maven-plugin</artifactId> <version>4.0.3</version>

<configuration>

....

</configuration>

</plugin>

# Implementasi

Pada pembuatan aplikasi antrian loket rumah sakit berbasis web ini penulis menggunakan sistem operasi Linux Ubuntu 16.04 dengan Java versi "1.8.0\_91".

# Instalasi Java

Java Runtime Environment dan Java Development Kit harus sudah terinstall karena aplikasi yang akan dibuat menggunakan pemprograman java. Langkah-langkah installasinya adalah :

1. Download JDK di <http://www.oracle.com/>
2. Extract file JDK yang sudah di download
3. Open terminal
4. Open file /etc/profile dengan perintah

*“$ sudo nano /etc/profile”*

1. Export JDK pada /etc/profile dengan menambahkan *:*

*“export PATH=$PATH:/lokasi\_jdk/bin”*

1. Restart, dan untuk mengetahui apakah java sudah terinstall dengan perintah *“$ java -version”*

# Instalasi Maven

Maven harus sudah terinstall karena aplikasi akan di jalankan menggunakan aplikasi ini. Langkah-langkah installasinya adalah :

1. Open terminal
2. Install Mysql dengan perintah

*“$ sudo apt-get install maven”*

1. Restart, dan untuk mengetahui apakah maven sudah terinstall dengan perintah *“$ mvn -version”*

# Instalasi MySql

Sebelum aplikasi ini berjalan DBMS Mysql harus sudah terinstall, Langkah-langkah installasinya :

1. Open Terminal
2. Install Mysql dengan perintah

*“$ sudo apt-get install mysql-server-5.6”*

1. Setelah instalasi selesai, open Mysql dengan perintah

*“$ mysql -uroot -p”*

1. Buat database baru untuk penyimpanan data antrian dengan perintah

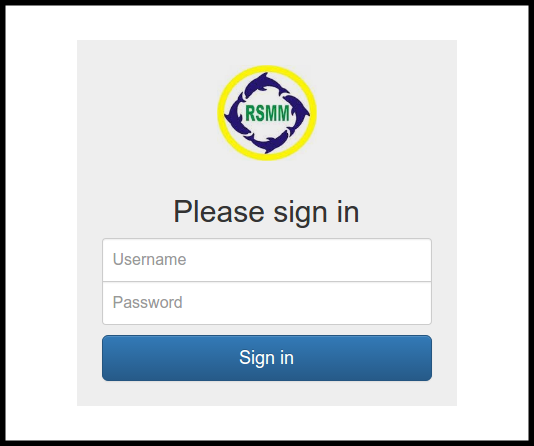
*“create database antrian\_development”*

# Implementasi Aplikasi

Setelah melakukan instalasi java, mysql dan MySql, Berikutnya penulis akan menampilkan desain dari tampilan aplikasi antrian ini. Untuk lebih jelasnya berikut ini adalah beberapa tampilan nya:

1. Halaman Login

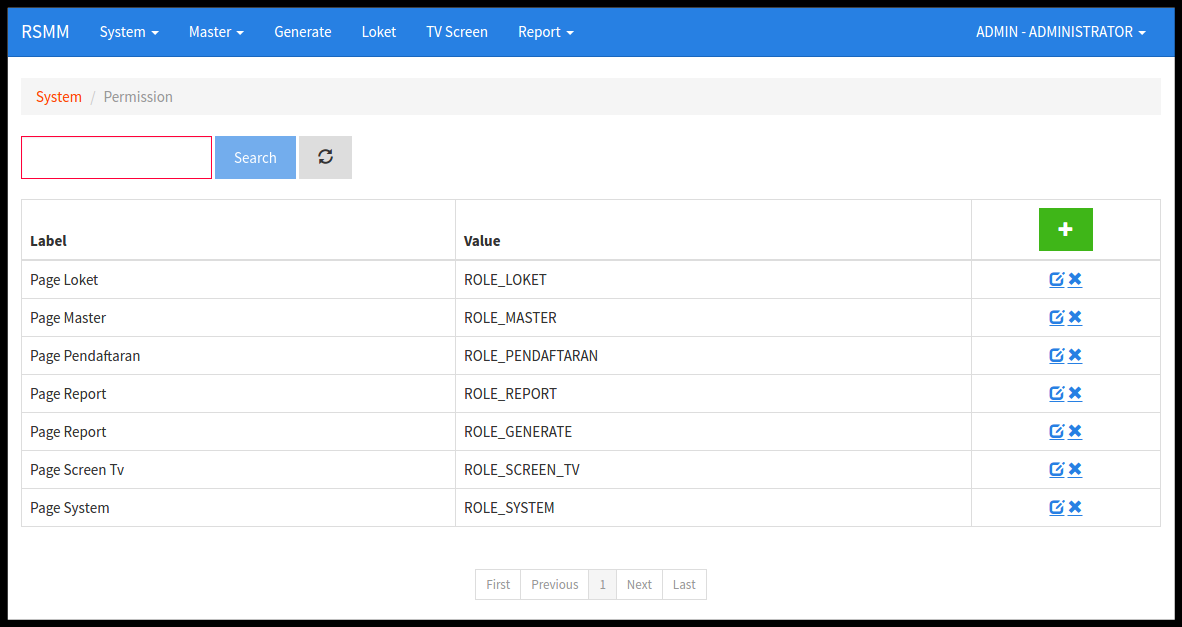
Halaman login adalah halaman yang pertama kali diakses oleh semua pengguna, user harus menginputkan username dan password yang valid untuk melanjutkan ke halaman lainnya



Gambar 3.14 Tampilan Menu Login

1. Halaman Permission

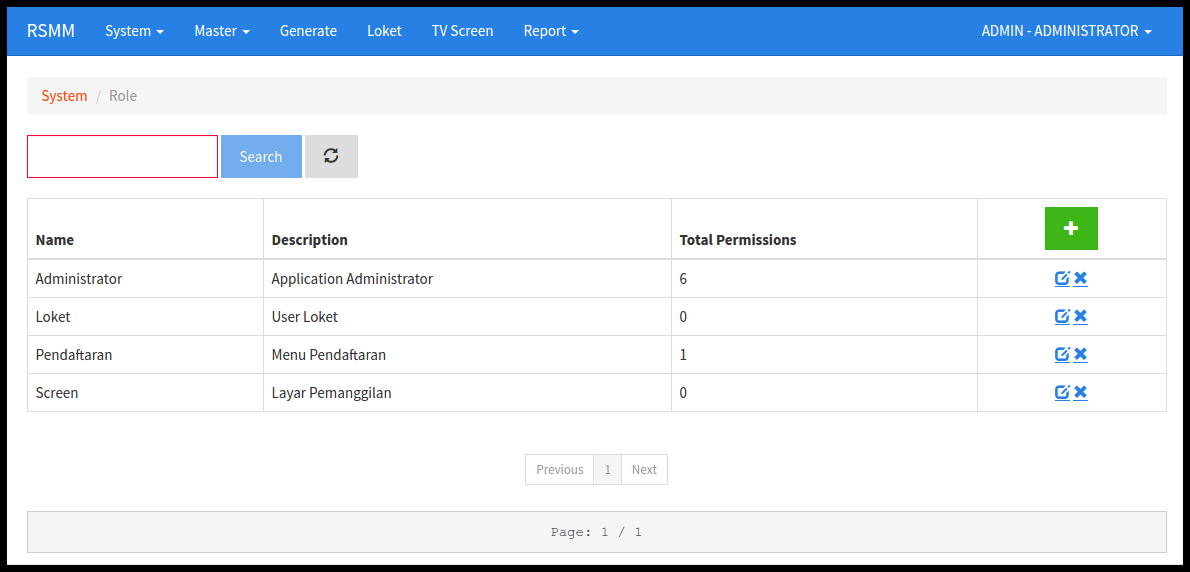
Halaman permission adalah halaman yang digunakan untuk menampilkan list, tambah, edit dan delete permission. Permission digunakan sebagai data hak akses aplikasi.



Gambar 3.15 Tampilan Menu Permission

1. Halaman Role

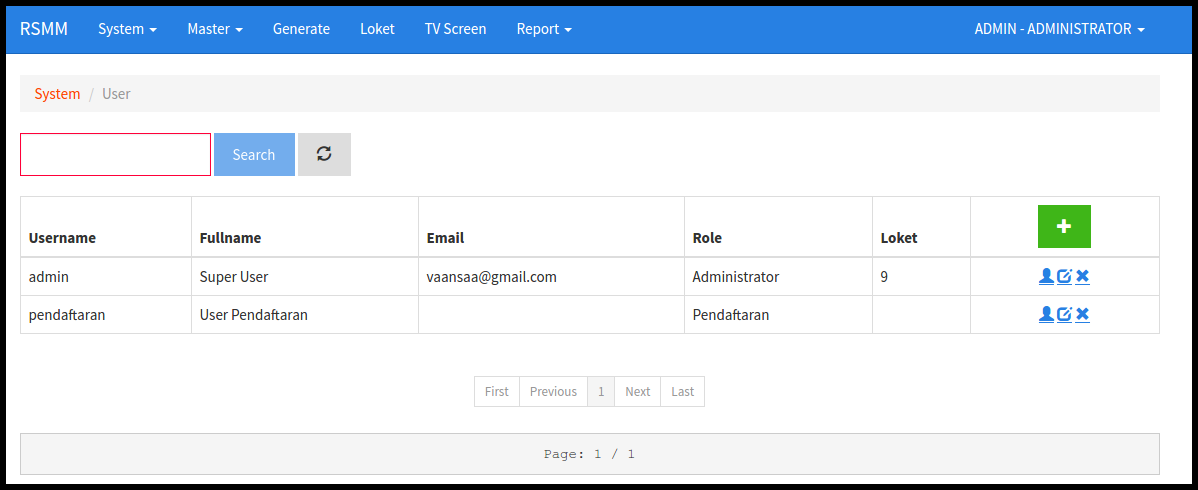
Halaman role adalah halaman yang digunakan untuk menampilkan list, tambah, edit dan delete role. Role digunakan untuk membagi/mengelompokkan data permission.



Gambar 3.16 Tampilan Menu Role

1. Halaman User

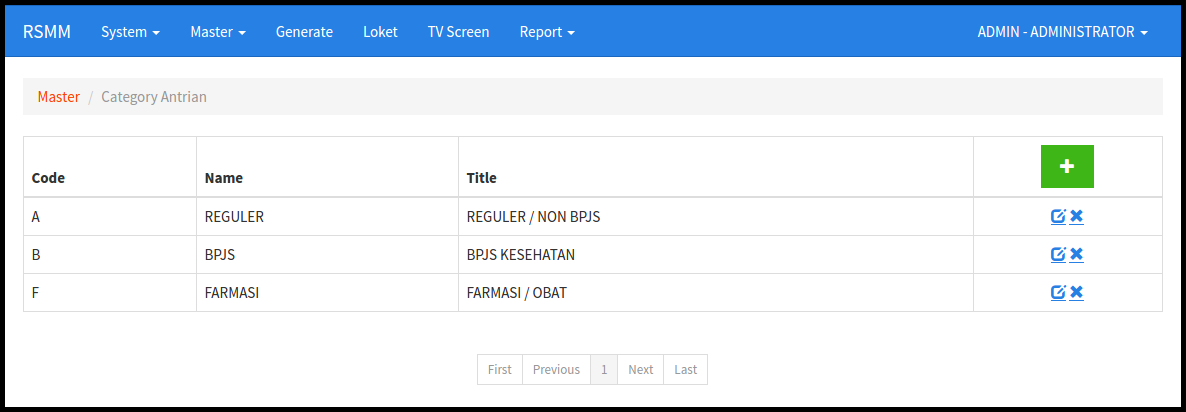
Halaman user adalah halaman yang digunakan untuk menampilkan list, tambah, edit dan delete user. User adalah data pengguna yang akan digunakan saat login aplikasi.



Gambar 3.17 Tampilan Menu User

1. Halaman Master Kategori

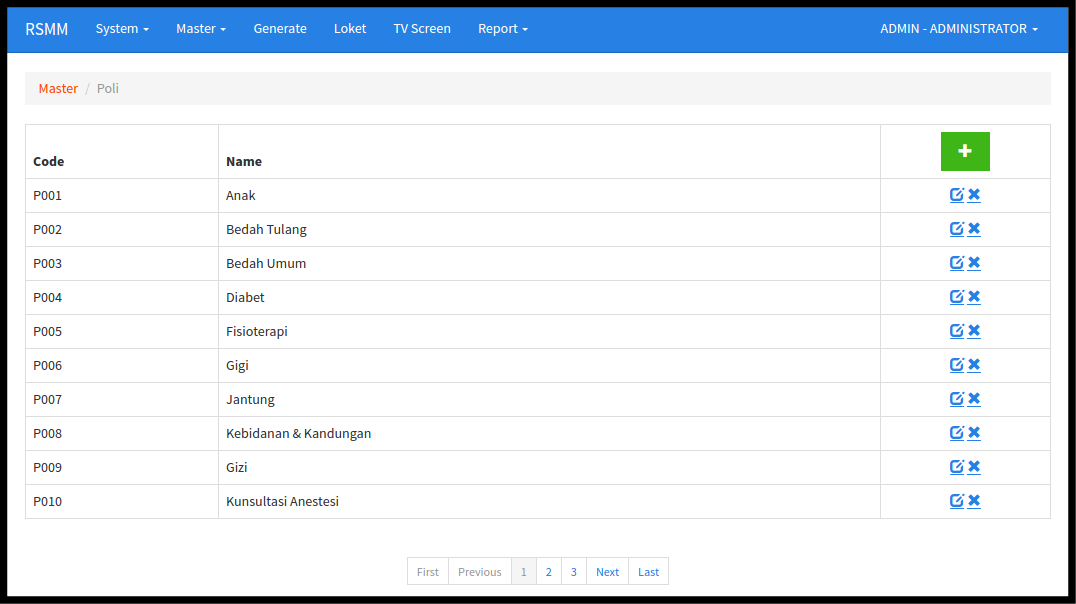
Halaman master kategori adalah halaman yang digunakan untuk menampilkan list, tambah, edit dan delete kategori. Kategori digunakan untuk membagi/mengelompokkan loket atrian.



Gambar 3.18 Tampilan Menu Master Kategori

1. Halaman Master Poli

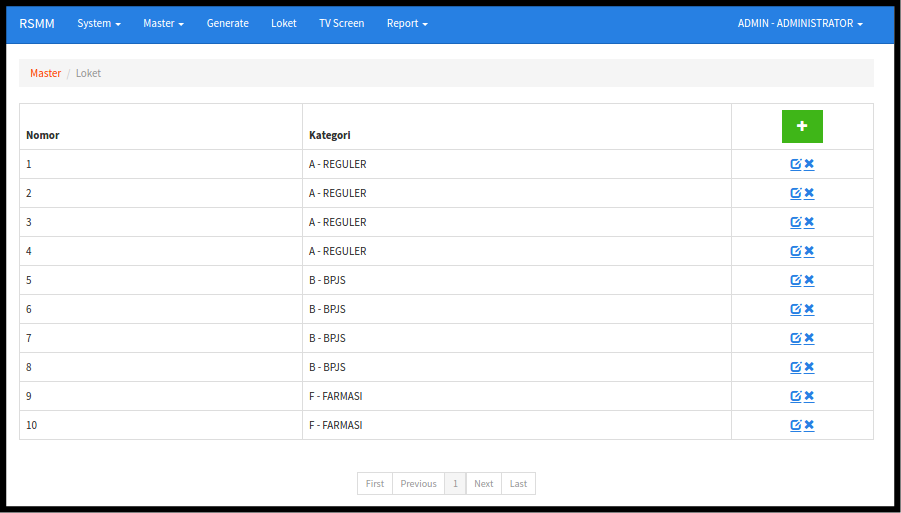
Halaman master poli adalah halaman yang digunakan untuk menampilkan list, tambah, edit dan delete poli. Poli digunakan sebagai kategori dokter.



Gambar 3.19 Tampilan Menu Master Poli

1. Halaman Master Loket

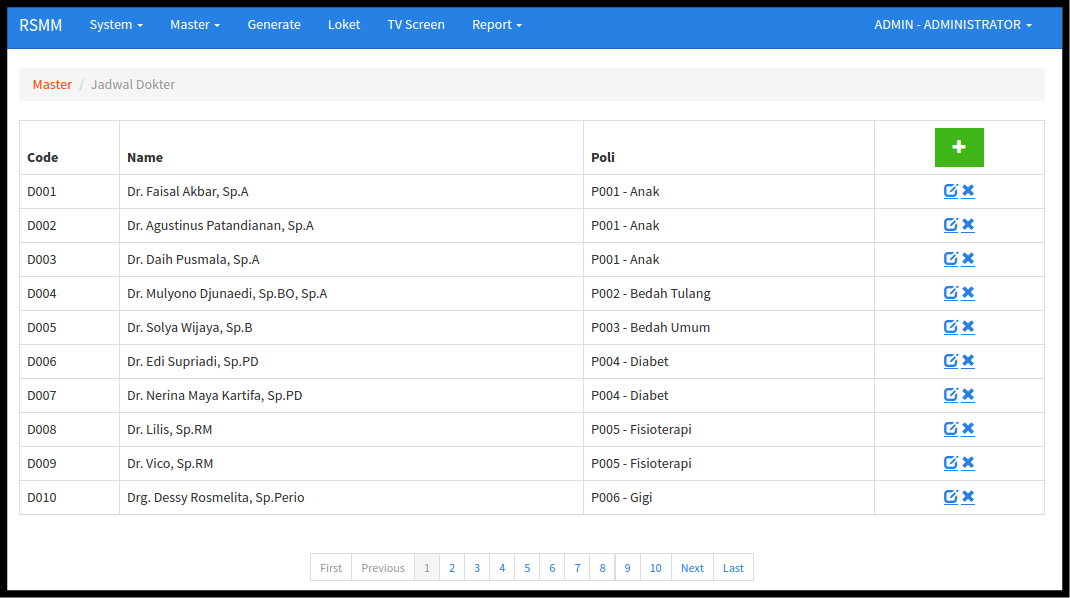
Halaman master loket adalah halaman yang digunakan untuk menampilkan list, tambah, edit dan delete loket.



Gambar 3.20 Tampilan Menu Master Loket

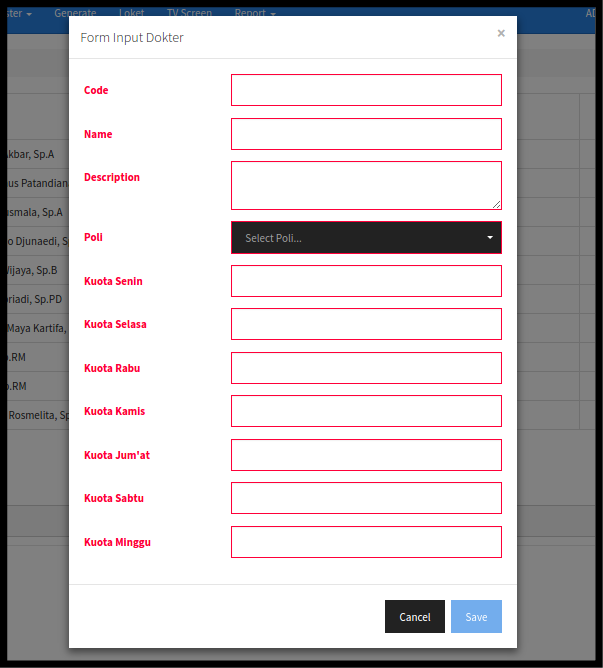
1. Halaman Master Jadwal Dokter

Halaman master jadwal dokter adalah halaman yang digunakan untuk menampilkan list, tambah, edit dan delete jadwal dokter.



Gambar 3.21 Tampilan Menu Master Jadwal Dokter

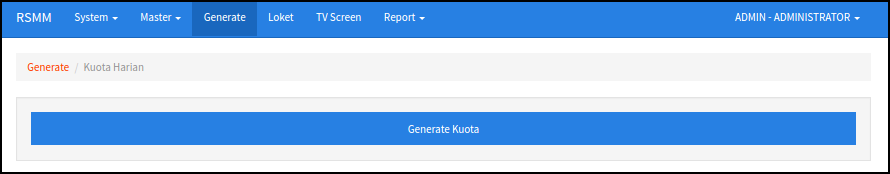
Halaman ini digunakan untuk menambahkan kuota dokter mulai dari hari senin sampai minggu, jika kuota diisi 0 maka dokter dianggap libur.



Gambar 3.22 Tampilan Input Jadwal Dokter

1. Halaman BOD/Generate

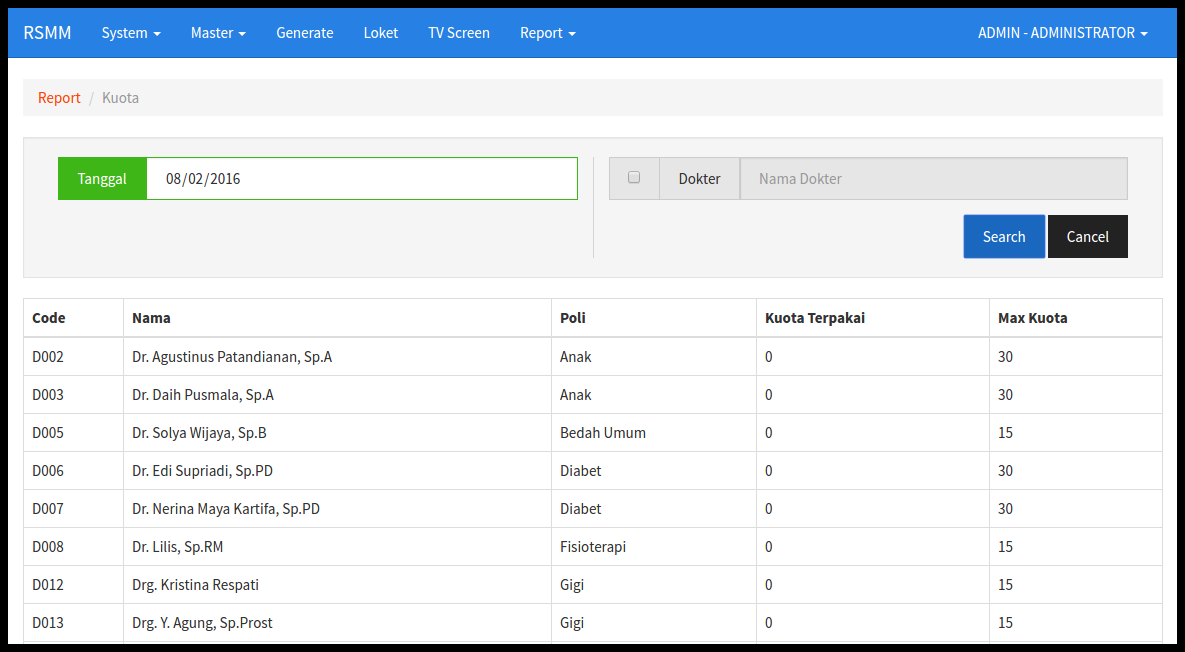
Halaman BOD ini hanya berisi 1 button untuk memulai proses BOD, proses ini berfungsi sebaagai penanda apakah rumah sakit libur atau tidak, selain itu digunakan untuk memindahkan kuota dokter ke table transaksi.



Gambar 3.23 Tampilan Menu Generate Kuota/BOD

1. Halaman Laporan Kuota

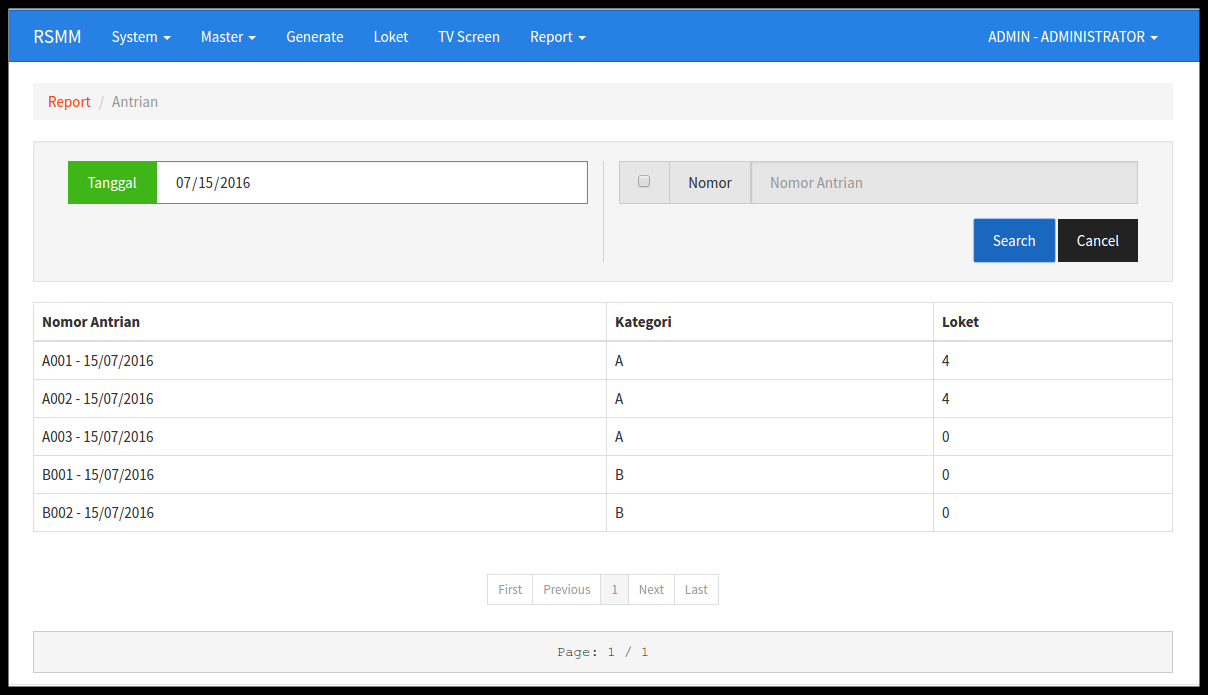
Halaman laporan kuota digunakan untuk menampilkan list kuota dengan filter tanggal. Pada halaman ini diketahui jumlah maximum kuota dan kuota yang sudah dipakai oleh setiap dokter.



Gambar 3.24 Tampilan Menu Laporan Kuota

1. Halaman Laporan Antrian

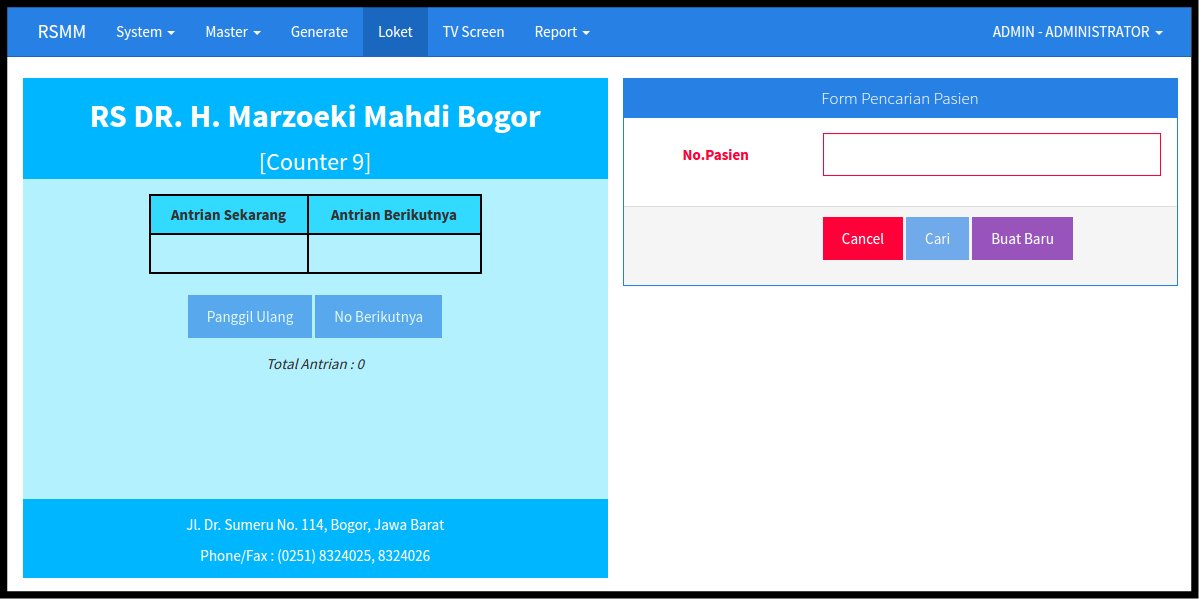
Halaman laporan antrian digunakan untuk menampilkan list nomor antrian dengan filter tanggal. Pada halaman ini diketahui nomor antrian dan loket yang memproses.



Gambar 3.25 Tampilan Menu Laporan Antrian

1. Halaman Transaksi Loket

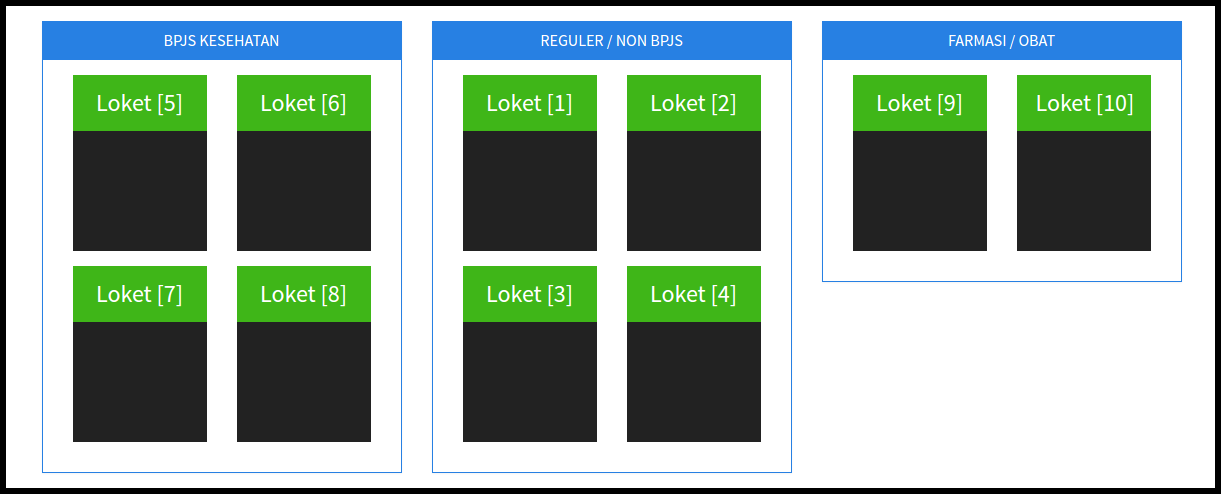
Halaman digunakan oleh user dengan role loket dan berfungsi sebagai halaman untuk mengakses nomor antrian dan data pasien yang sedang di proses. Pada halaman ini juga digunakan untuk memanggil nomor antrian.



Gambar 3.26 Tampilan Menu Transaksi Loket

1. Halaman Information Screen

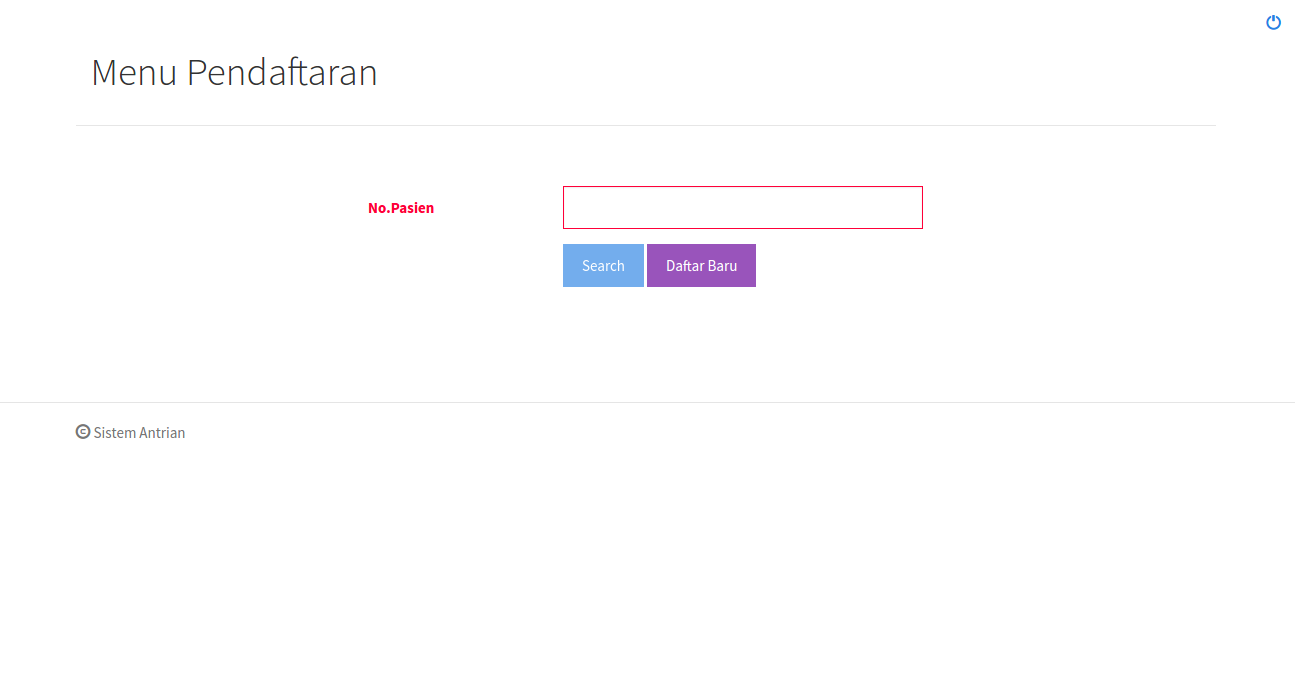
Halaman ini ditampilkan sebagai sarana informasi kepada pasien yang menunggu. Pada halaman ini di beritahukan informasi nomor antrian yang sedang diproses oleh setiap loket yang ada di rumah sakit tersebut.



Gambar 3.27 Tampilan Menu Information Screen

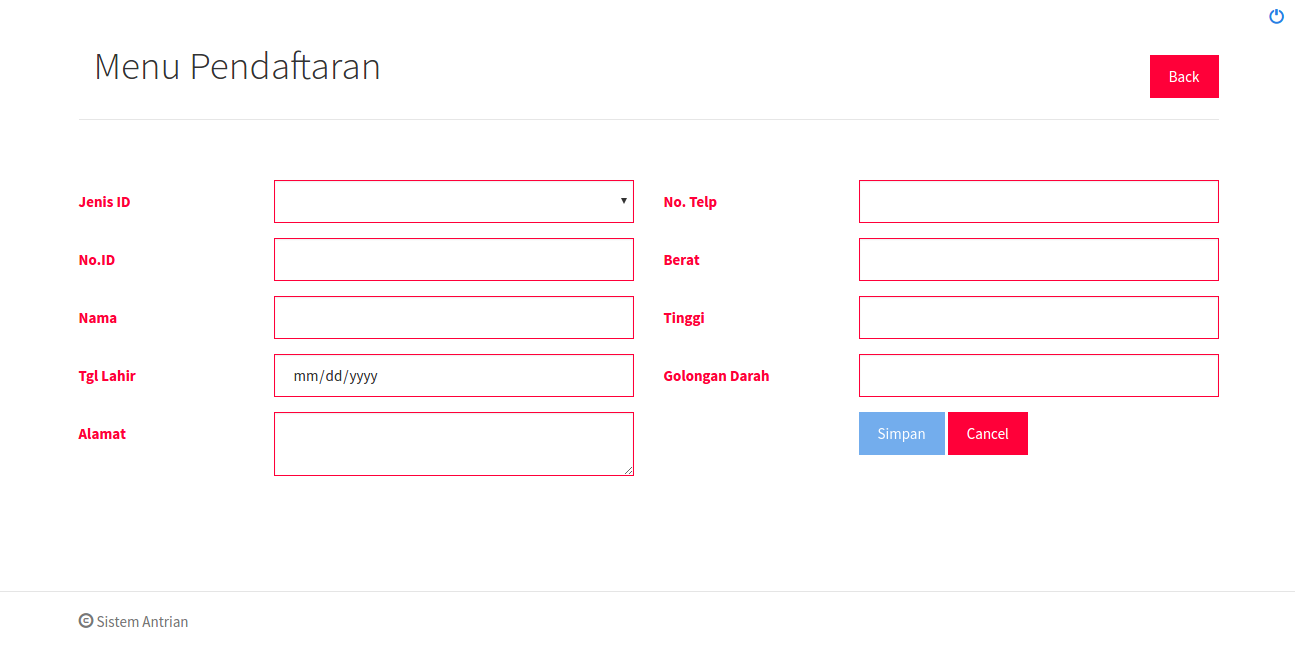
1. Halaman Pendaftaran

Halaman menu Pendaftaran terdapat 2 form, yang pertama di tampilkan adalah form pencarian nomor pasien yang dikhususkan untuk pasien yang telah terdaftar di system.



Gambar 3.28 Tampilan Menu Pendaftaran

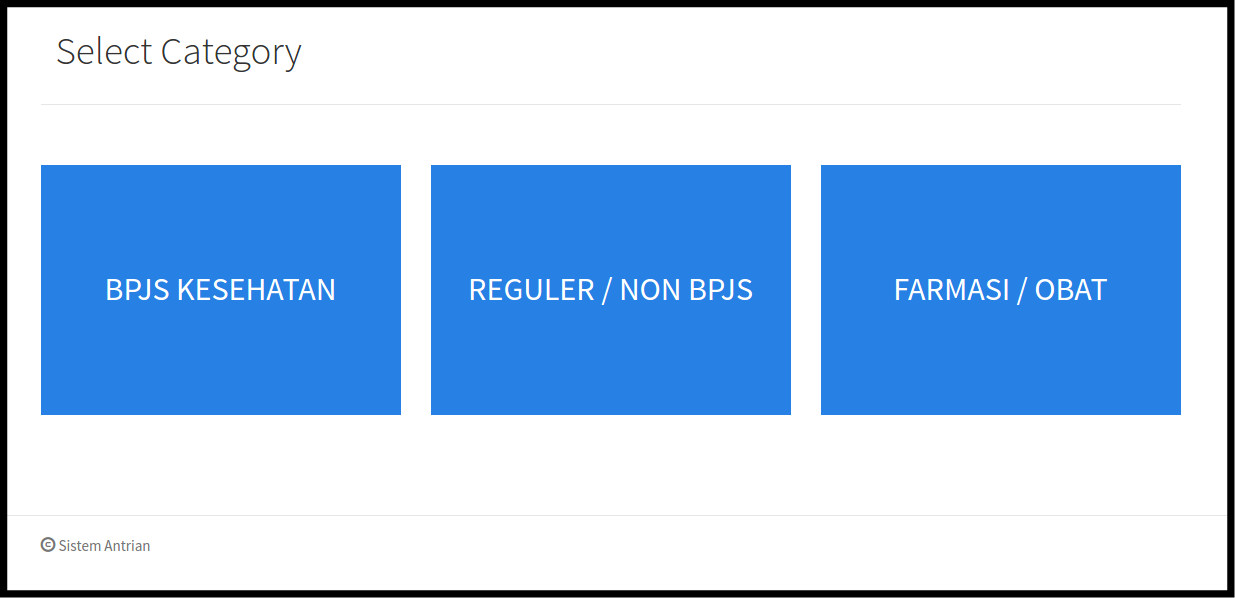
Jika pasien belum terdaftar maka pasien akan masuk ke form berikutnya dengan mengklik button daftar baru, pada form yang kedua user akan mengisi formulir/form pendaftaran secara bener seperti gambar 3.24



Gambar 3.29 Tampilan Menu Pendaftaran

1. Halaman Pengambilan Nomor Tiket

Halaman ini tampil setelah pasien melakukan pendaftaran, pada halaman ini terdapat 3 tingkat/form pemilihan. Form yang pertama adalah memilih kategori loket seperti gambar 3.25



Gambar 3.30 Tampilan Menu Pilih Kategori

Setelah memilih salah satu kategori selanjutnya akan masuk ke tampilan pemilihan poli seperti pada gambar 3.26



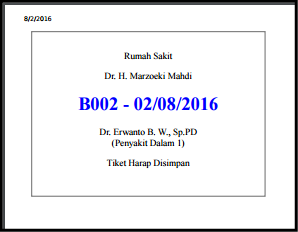
Gambar 3.26 Tampilan Menu Pilih Poli

Setelah memilih salah satu poli selanjutnya akan masuk ke tampilan pemilihan dokter pada poli tersebut seperti pada gambar 3.27



Gambar 3.27 Tampilan Menu Pilih Dokter

Terakhir setelah memilih dokter, pasien akan mendapat nomor antrian loket seperti gambar 3.28



Gambar 3.28 Tampilan Print Nomor Antrian

# Pengujian

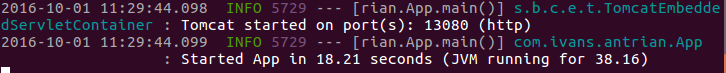
Pada tabel 3.1 dapat dilihat spesifikasi server yang digunakan sebagari testing aplikasi. Pada testing ini hanya dilakukan secara offline dan dapat diakses melalui LAN dengan mengakses ip dari server tempat aplikasi di jalankan.

Tabel 3.1 Tabel Spesifikasi Server

|  |  |  |
| --- | --- | --- |
| **No** | **Name** | **Value** |
| 1 | Laptop | Dell Inspiron 3458 |
| 2 | Size | 14 “ |
| 3 | OS | Ubuntu 16.04 LTS 64-bit |
| 4 | Browser | Chrome 51.0.2704.106 (64-bit) |
| 5 | Java | Java versi "1.8.0\_91" |
| 6 | Project Builder | Maven |
| 7 | DBMS | MySql 5.6 |

Os server menggunakan Linux Ubuntu dan dijalankan melalui terminal, terdapat 4 aplikasi yang akan dijalankan dengan port yang berbeda.

* + - 1. API Antrian (Java), berfungsi sebagai penyedia API yang digunakan untuk distribusi data dari DBMS.



* + - 1. Auth Server (Java), berfungsi sebagai generate token.
      2. UI-Pendaftaran (AngularJS), berfungsi untuk menampilkan halaman pendaftaran.



* + - 1. UI-Management (AngularJS), berfungsi untuk menampikan halaman management, loket dan information screen



Berikut ini adalah table pengujian aplikasi, pengujian dilakukan melalui browser chrome. Pada table dilakukan pengujian secara berurutan mulai dari generate kuota hingga pasien dilayani oleh loket.

Tabel 3.2Pengujan Aplikasi Browser Chrome

|  |  |  |
| --- | --- | --- |
| **No** | **Tampilan Menu** | **Gambar** |
| **1** | **Halaman Pendaftaran sebelum BOD** |  |
| **2** | **Melakukan Proses BOD (Admin)** |  |
| **3** | **Halaman Pendaftaran (Melakukan Pencarian Pasien)** |  |
| **4** | **Melakukan Pendaftaran Baru** |  |
| **5** | **Memilih Kategori (Pilih BPJS)** |  |
| **6** | **Memilih Poli (Pilih Paru-Paru)** |  |
| **7** | **Memilih Dokter** |  |
| **8** | **Pasien Mendapat Printout Nomor Antrian** |  |
| **9** | **Loket Mendapat Nomor Antrian Yang Siap Diproses** |  |
| **10** | **Loket Mengambil Nomor Antrian** |  |
| **11** | **Halaman Screen Information Memanggil Dan Menampilkan Nomor Antrian yang Diproses loket** |  |