BAB 3
ANALISIS DAN PERANCANGAN

3.1 Analisis Sistem

Pembangunan Sistem Pengendali Lampu dan Pemanfaatan RFID Pada Kunci Pintu Elektronik Untuk Meningkatkan Keamanan Rumah Menggunakan Mikrokontroller Arduino Berbasis Android adalah sistem yang digunakan untuk mengendalikan listrik pada lampu rumah dan mengontrol kunci pintu. Sistem ini akan menggabungkan rangkaian listrik pada lampu rumah, kunci (*solenoid*) yang sudah terintegrasi RFID dan sensor PIR untuk memonitor ruangan, sistem ini digabungkan dengan sebuah mikrokontroler arduino yang diprogram dengan bahasa pemrograman C. Dengan sistem ini pengguna dapat mengontrol dan memonitor lampu sesuai dengan kebutuhan yang dikendalikan dari jarak jauh, monitoring kunci (*solenoid*) yang sudah terintegrasi RFID yang akan memberikan informasi seseorang yang masuk ruangan berdasarkan *tag* yang digunakan untuk membuka pintu, kemudian memonitor ruangan yang akan memberikan membunyikan alarm (*buzzer*) tanda pemberitahuan ketika ada seseorang masuk ke ruangan yang dimonitoring secara ilegal dan akan mengirimkan notifikasi atau pesan pada ponsel android. Sistem ini dikendalikan dengan sebuah aplikasi yang terkoneksi dengan internet pada perangkat android untuk setiap pengaturan serta pengontrolan lampu, sehingga sistem dapat dimonitor dan dikendalikan jarak jauh.

Dalam pembangunan sistem ini ada beberapa ketentuan yang harus dipenuhi, baik secara penggunaan maupun secara teknis, penguna aplikasi Sistem Pengendali Lampu dan Pemanfaatan RFID Pada Kunci Pintu Elektronik ini tidak semua orang yang memiliki aplikasi ini dapat menggunakannya, aplikasi ini akan meminta *id user* dan *password* untuk menggunakannya, untuk menjaga penyalahgunaan aplikasi dalam pengunaaanya. Dalam pelaksanaannya sistem ini akan melakukan perintah sesuai dengan perintah dari penggunanya.

Berikut analisis sistem yang ada pada Sistem Pengendali Lampu dan Pemanfaatan RFID Pada Kunci Pintu Elektronik:

1. Menyalakan dan mematikan lampu denggan button *ON-OFF* pada ponsel android.
2. Membuka kunci pintu elektronik (*Selonoid*) dengan *tag* RFID dan mengirim notifikasi ke ponsel berdasarkan tag yang digunakan untk membuka kunci.
3. Membunyikan *buzzer* sebagai tanda peringatan ketika ada seseorang masuk secara ilegal ke ruangan yang di monitoring.
4. Sistem Pengendali Lampu dan Pemanfaatan RFID Pada Kunci Pintu Elektronik dikontrol pada aplikasi berbasis android yang terkoneksi *internet*.

3.1.1 Analisis Masalah

Dari fenomena yang terjadi terdapat beberapa masalah yang sering terjadi pada penggunaan energi listrik lampu, kunci pintu dan keamanan ruangan:

1. Sulitnya memonitor pemakaian lampu.
2. Terbatasnya jarak dalam mengendalikan lampu.
3. Tidak tepatnya waktu pemakaian lampu.
4. Sulitnya mendapatkan informasi siapa saja orang yang masuk ke dalam ruangan yang dimonitor.

Oleh karena itu dibutuhkan suatu solusi untuk menangani permasalahan yang sedang terjadi yaitu dibutuhkan suatu Sistem Pengendali Lampu dan Pemanfaatan RFID Pada Kunci Pintu Elektronik yang bertujuan untuk:

1. Memudahkan pengguna dalam mendapatkan informasi siapa saja yang masuk ruangan berdasarkan ID tag.
2. Memudahkan pengguna dalam memonitor pemakaian lampu.
3. Mempermudah pengguna dalam mengendalikan pemakaian lampu dari jarak jauh.
4. Tepatnya waktu pemakaian lampu dengan otomatisasi waktu yang telah ditentukan.

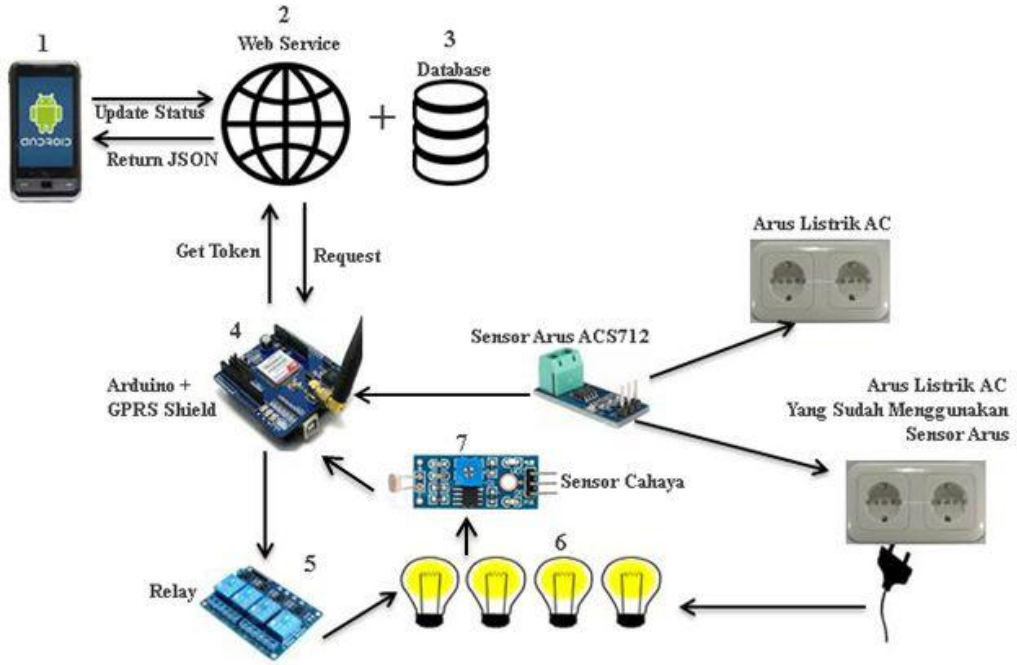
3.1.2 Analisis Arsitektur Perancangan Sistem

Analisis Arsitektur Sistem merupakan sistem yang akan dibangun, aplikasi Pengendali Lampu dan Pemanfaatan RFID Pada Kunci Pintu Elektronik yang berbasis internet ini berkomunikasi dengan alat mikrokontroler yang dijembatani oleh *web service* dengan menggunakan pertukaran bahasa JSON, adapun rancangan sistem yang akan dibangun dapat dilihat pada Gambar 3.1 dan Gambar 3.2.

Analisis Sistem merupakan proses untuk mendeskripsikan fisik sistem yang akan dibangun, komponen-komponen sistem yang akan dibangun meliputi:

1. *Web service* berfungsi sebagai perantara atau jembatan untuk menghubungkan atau menajalankan perintah dari ponsel Android (Ponsel Pengguna) supaya perintah dapat diterima oleh arduino.
2. *GSM/GPRS Shield* sebagai alat untuk *merequest* data dari *web service* dan mengirimkan nya ke Arduino untuk di eksekusi.
3. Ponsel Android berfungsi sebagai alat untuk mengontrol lampu jarak jauh dan memonitoring ruangan .
4. Tag RFID bersungsi sebagai id atau tanda pengenal yang digunakan untuk membuka kunci.
5. RFID *reader* PN532 berfungsi sebagai scanner pembaca kartu *tag* RFID.
6. *Relay* berfungsi sebagai pengganti sakelar.
7. *Sensor PIR* sebagai sensor untuk memonitoring ruangan dan *buzzer* sebagai alert atau alarm jika sensor mendeteksi adanya pencuri.
8. *Sensor Cahaya Photorisestor* sebagai sensor yang digunakan untuk memonitoring nyala atau tidaknya lampu.
9. *Selonoid (door lock)* berfungsi sebagai pengganti kunci mekanik.
10. *LCD keypad* berfungsi sebagai layar untuk menampilkan pesan ketika *menscan* tag RFID ke *reader* dan menu untuk tambah dan hapus *tag* RFID.

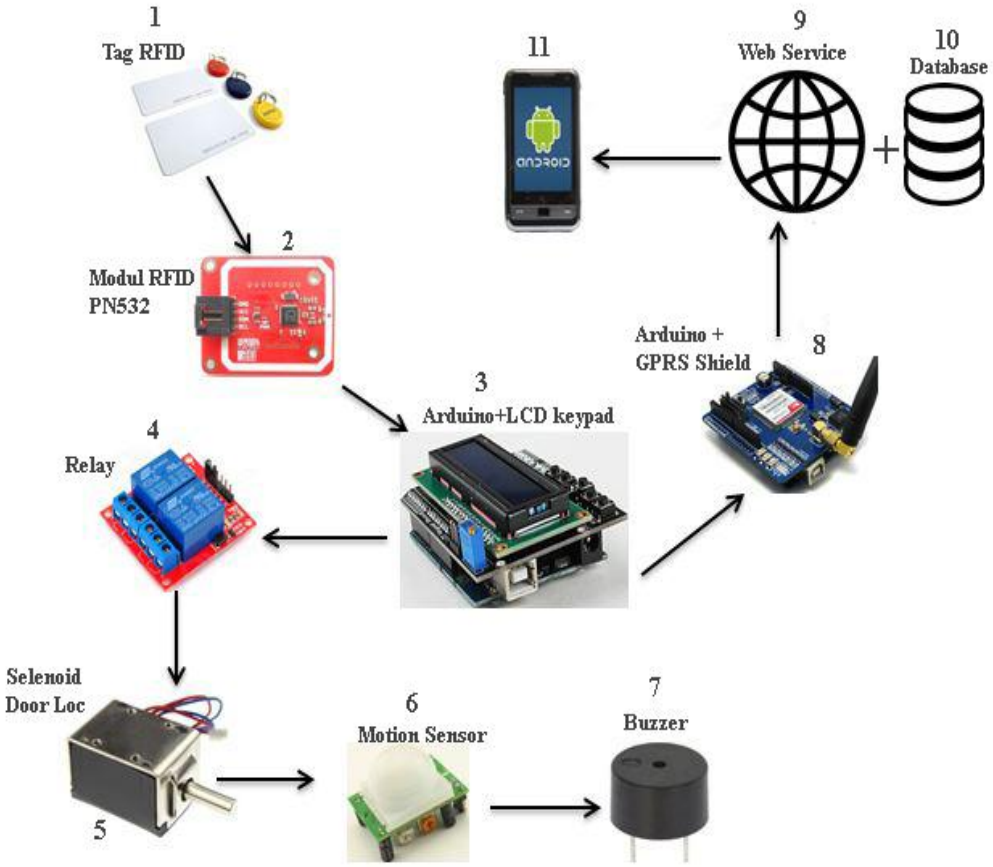
1. Rancangan Arsitektur Sistem Pengendali Lampu



Gambar 3.1 Arsitektur Perancangan Sistem Pengendali Lampu

Pada Gambar 3.1 menggambarkan sistem pengendali lampu yang akan dibangun, diamana perangkat android akan mengakses token dari database melalui koneksi *web service*, kemudian *web service* akan mengembalikan nilai kepada perangkat android berupa bahasa JSON yang telah di *encod* pada *web service* yang akan diterima oleh perangkat android, begitupun dengan mikrokontroler arduino yang akan *merequest* token yang berada *pada web service* dan *web service* akan mengembalikan nilai token dengan bahasa JSON yang akan diparsing oleh mikrokontroler.

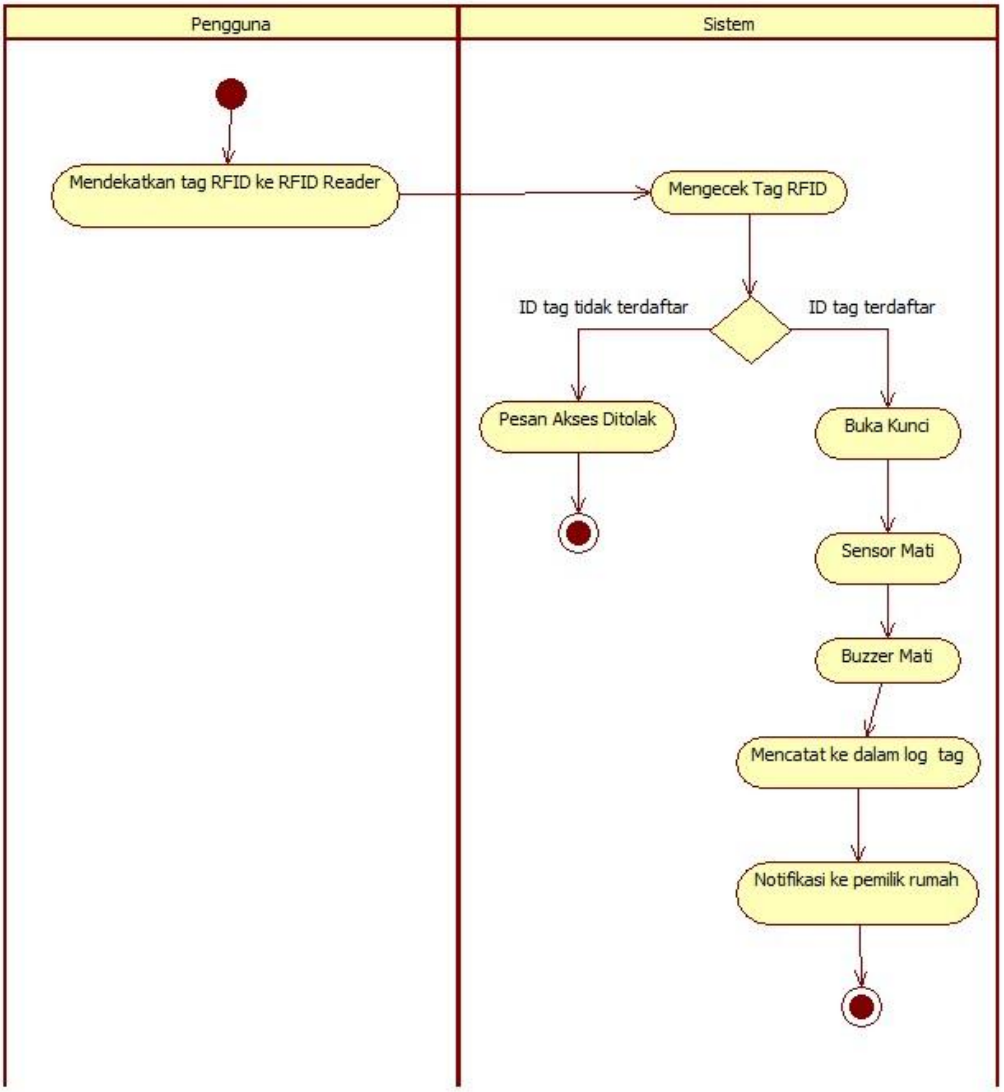
2. Rancangan Arsitektur Sistem Buka Kunci



Gambar 3.2 Arsitektur Perangcangan Sistem Buka Kunci

Pada Gambar 3.2 menggambarkan sistem buka kunci yang akan dibangun, dimana pengguna mendekatkan tag RFID yang telah didaftarkan pada RFID *reader*, kemudian LCD *keypad* menampilkan pesan buka kunci kemudian *relay* mati dan kunci *selonoid* terbuka serta *motion sensor* dan *buzzer* menjadi mati, kemudian Arduino mengirim data ID tag yang dipakai untuk membuka kunci ke *web service* kemudian membandingkan dengan ID tag yang telah diberi nama pada database lalu mengirimkan notifikasi kepada ponsel android.

1. Pengguna akan membuka kunci pintu menggunakan kartu atau tag RFID.



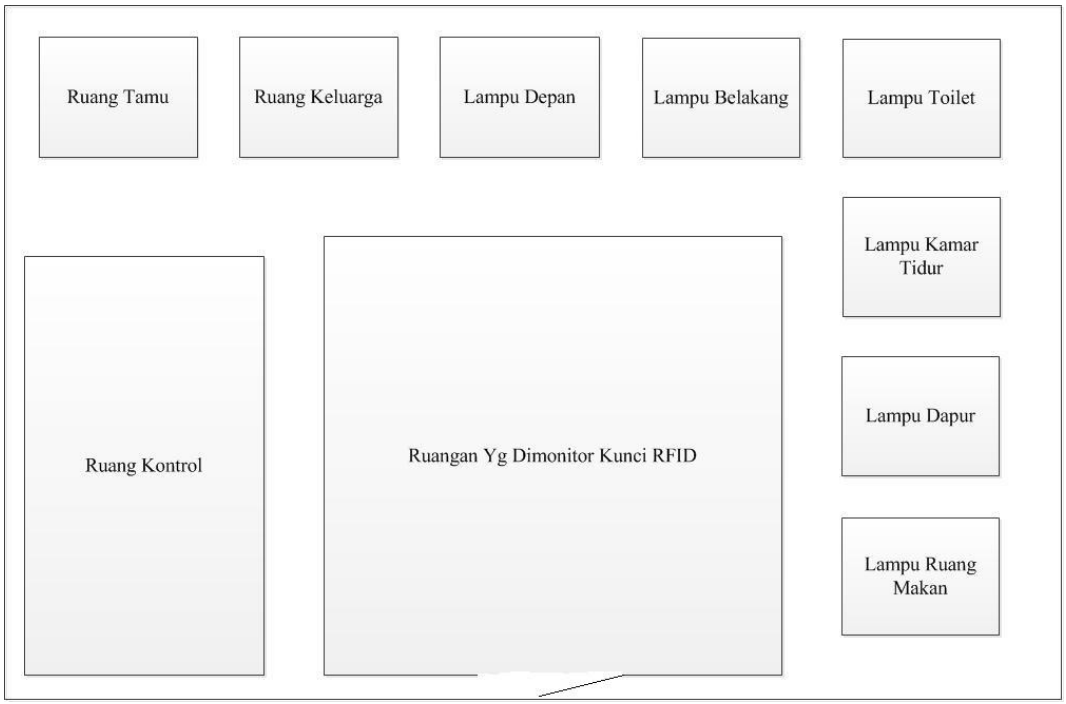
Gambar 3.3 Analisis Arsitektur Alur Sistem Buka Pintu

3.1.3 Analisis Miniatur Rumah

Pada penelitian ini terdapat 8 buah lampu, lampu ruang tamu, ruang keluarga, lampu depan rumah, lampu belakang rumah, lampu toilet, lampu kamar tidur, lampu dapur dan lampu ruang makan.

Pada miniatur rumah juga terdapat pintu dan ruangan sebagai implementasi RFID *door lock* (*Seolenoid*) dan monitoring keamanan ruangan. Berdasarkan survey dari 10 rumah, 7 diantaranya memiliki satu ruangan khusus

tempat penyimpanan barang berharga, adapun ruangan penelitian yang dibangun pada miniatur rumah yang di dalamnya terdapat satu ruangan yang dimonitoring, satu pintu untuk penerapan *door lock (Solenoid)* yang terintegrasi dengan *RFID*, dan satu ruangan yang dimonitoring yang daigambarkan pada gambar Gambar 3.4.



Gambar 3.4 Perancangan Pada Miniatur Rumah

.

3.1.4 Analisis *Web Service*

Pada pembangunan *web service* ini sebelumnya telah dijelaskan pada landasan teori, yang mana *web service* terbagi pada 3 komponen yaitu

1. *Simple Object Access Protocol* (SOAP), yaitu protokol yang bertanggungjawab dalam pertukuran informasi dalam lingkungan jaringan terdistribusi.
2. *Web service definition Language* (WDSL), dokumen standart yang dituliskan pada format XML, dan mendifinikasan kehadiran *web service* dalam suatu jaringan.

3. *Universal description, Discovery and Inttegration* (UDDI), yaitu suatu lokasi direktori yang beriskan *service* (layanan) dan bersifat bebas *platform* (*platform independent*), dituliskan berbasis XML dan dapat diakses oleh entitas yang berada di dalam.

Dalam pembangunan *web service* ini berbasis SOAP dengan pertukaran data berbasis JSON, pembangunan web *service* dibangun menjadi 2 bagian yaitu *web service* untuk perangkat android dan *web service* untuk perangakat arduino, kedua *web service* ini akan terkoneksi pada satu *database*.

3.1.5 Analisis Kebutuhan Non Fungsional

Analisis kebutuhan non fungsional merupakan analisis yang dibutuhkan untuk menentukan spesifikasi kebutuhan sistem. Spesifikasi ini juga meliputi elemen atau komponen-komponen apa saja yang dibutuhkan untuk sistem yang akan dibangun sampai dengan sistem tersebut diimplementasikan. Analisis kebutuhan ini juga menentukan spesifikasi masukan yang diperlukan sistem, keluaran yang akan dihasilkan sistem dan proses yang dibutuhkan untuk mengolah masukan sehingga menghasilkan suatu keluaran yang diinginkan. Kebutuhan non fungsional yang dibutuhkan untuk membangun ulang sistem ini terdiri dari dua hal, yaitu: kebutuhan perangkat keras dan kebutuhan perangkat lunak.

3.1.5.1 Analisis Perangkat Keras

Analisis kebutuhan perangkat keras yang digunakan untuk mengimplementasikan aplikasi ini dapat dilihat pada Tabel 3.1.

Tabel 3.1 Spesifikasi Kebutuhan Perangkat Keras

No	Perangkat Keras	Spesifikasi
1	Sistem Operasi	ICS
2	RAM	512 GB
3	Memory Internal	1 GB
4	Processor	800 MHz

Sedangkan spesifikasi perangkat keras yang direkomendasikan untuk penggunaan aplikasi ini, yaitu untuk Pengguna dapat dilihat pada Tabel 3.2.

Tabel 3.2 Spesifikasi Perangkat Keras Ponsel Android

No	Perangkat Keras	Spesifikasi
1	Sistem Operasi	Jelly Bean
2	RAM	1 GB
3	Memory Internal	4 GB
4	Processor	1 GHz

Setelah evaluasi mengenai spesifikasi perangkat keras, untuk pengguna aplikasi Pengendali Lampu dan Pemanfaatan RFID Pada Kunci Pintu Elektronik, yaitu dengan membandingkan kebutuhan spesifikasi minimal untuk pembuatan sistem, dan dapat disimpulkan spesifikasi perangkat keras sistem untuk pengguna sudah di atas spesifikasi minimal sistem dan memingkinkan dapat berjalan diperangkat keras pengguna.

3.1.5.2 Mikrokontroler

Mikrokontroler yang digunakan dalam penelitian ini adalah menggunakan dua buah mikrokontroler Arduino uno r3, GPRS/GSM *Shield*, *tag* RFID 13,56 MHz, *relay*, *sensor* PIR, *buzzer* dan LCD *key pad*.

3.1.5.2.1 Arduino UNO R3

Arduino uno adalah papan mikrokontroler berbasis ATMEGA328 yang memiliki 14 pin digital *input*/*output*, 6 diantarnya digunakan sebagai *output Pulse Width Modulation* (PWM), 6 *input* analog, clock speed 16MHz, koneksi USB, Jack listrik, *header* ICSP, dan tombol *reset*. Board ini menggunakan daya yang terhubung ke computer dengan kabel USB atau daya external dengan adapator AC-DC atau baterai.

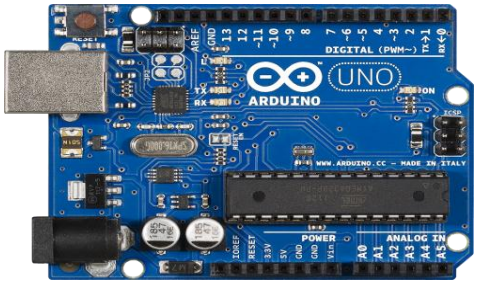
Arduino Uno adalah pilihan yang baik untuk pertama kali atau pemula yang ingin mengenal Arduino. Di samping sifat nya yang reliabel juga harga yang terjangkau.

Tabel 3.3 Spesifikasi Arduino UNO R3

Mikrokontroler	ATMEGA328
Tegangan Operasi	5V
Tegangan Input	7-12V
Batas Tegangan Input	6-20V
Pin Digital I/O	14(6 pin PWM)
Pin Analog Input	6
Arus DC Per I/O Pin	40mA
Arus DC untuk Pin 3,3V	50mA
Flash Memory	32 KB (ATmega328), 0,5 KB bootloader
SRAM	2 KB (ATmega328)
EEPROM	1 KB (ATmega328)
Clock	16MHz

Arduino dapat di aktifkan melalui koneksi USB atau dengan catu daya Eksternal. Sumber daya dipilih secara otomatis. Sumber daya Eksternal (*Non-USB*) dapat berasal dari adapter AC ke DC atau baterai. Adaptor ini dapat dihubungkan dengan menancapkan *Power Jack*, dapat juga dihubungkan pada *Power pin* (Gnd dan Vin).

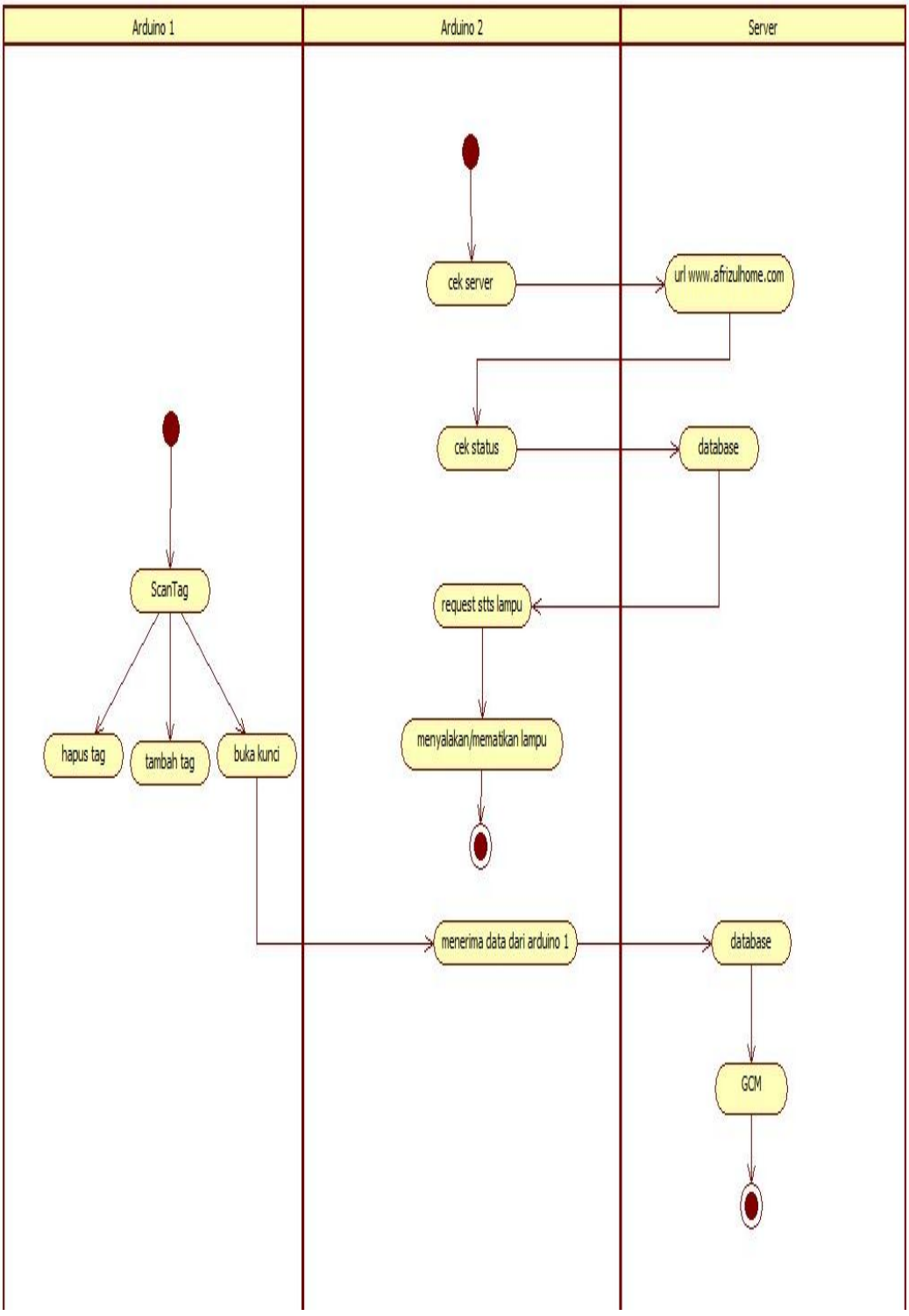
Board Arduino Uno dapat beroperasi pada pasokan Eksternal dari 6 sampai 20 *volt*. Jika disuplai kurang 7V. Meskipun pin 5V dapat di suplai kurang dari 5V, *board* mungkin tidak stabil. Jika menggunakan teganggan lebih dari 12V, regular tegangan bisa panas dan merusak *board*. Kisaran yang di sarankan adalah 7 sampai 12 *Volt*. Dalam Penelitian ini menggunakan 2 buah Arduino uno R3 dikarenakan banyaknya modul yang digunakan dan supaya tidak bentrok dengan modul RFID PN532. Dalam penelitian ini penulis menggunakan 2 buah *board* arduino uno R3 dikarenakan keterbatasan *port* pin pada board arduino uno R3.



Gambar 3.5 Arduino UNO R3 [8]

3.1.5.3 Analisis Kerja Sistem Kerja Arduino

Arduino berjalan dengan program yang ditanam didalam *chip* mikrokontroler, dalam penelitian ini arduino terkoneksi pada *internet* yang dihubungkan melalui internet *shield* dan terkoneksi pada *server* yang didalam nya terdapat *web service*.Adapun sistem kerja arduino dapat dilihat pada Gambar 3.6.



Gambar 3.6 Analisis Alur Kerja Arduino

3.1.6 Analisis Kebutuhan Pengguna

Analisis pengguna merupakan ketentuan pengguna yang dapat mengoprasionalkan atau menjalankan aplikasi Pengendali Lampu dan Pemanfaatan RFID Pada Kunci Pintu Elektronik ini, adapun ketentuan pengguna adalah sebagai berikut:

1. Pengguna dapat mengoperasionalkan perangkat Android.
2. Perangkat android yang digunakan pengguna harus terkoneksi internet.
3. Pengguna harus memiliki aplikasi Home Automation Control (HAC) agar dapat mengendalikan lampu, membuka kunci pintu dan memonitoring ruangan.
4. Pengguna dianjurkan berumur 12 sampai 40 tahun.

Dengan karakteristik pengguna di atas dapat disimpulkan bahwa penguna yang dapat menggunakan aplikasi ini minimal harus memahami perangkat Android atau dapat menjalankan aplikasi android.

Adapun Ketentuan untuk admin yaitu:

1. Admin dapat mengoperasikan perangkat Android dan web.
2. Mengerti konsep rfid untuk tambah tag, hapus tag, dan daftarkan tag.

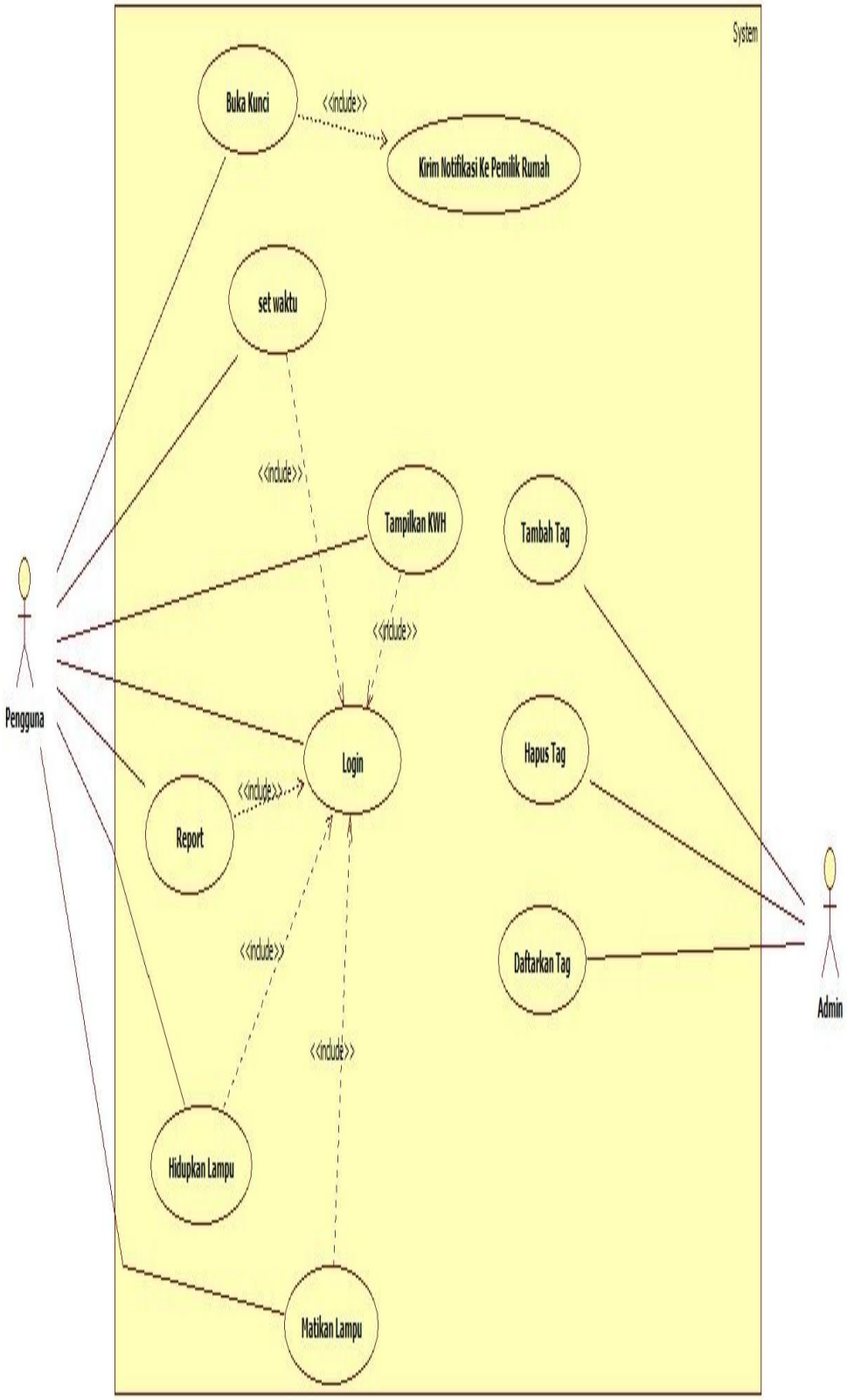
3.1.7 Analisis Kebutuhan Fungsional

Analisis kebutuhan fungsional menggambarkan proses kegiatan yang akan diterapkan dalam sebuah sistem dan menjelaskan kebutuhan yang diperlukan sistem agar sistem dapat berjalan dengan baik serta sesuai dengan kebutuhan. Analisis kebutuhan fungsional ini meliputi analisis kebutuhan sistem, analisis kebutuhan data, spesifikasi dan pemodelan sistem.

Pemodelan sistem dimodelkan dengan UML (*Unified Modeling Language*). Tahap-tahap pemodelan dalam analisis tersebut antara lain *Use Case Diagram, Class Diagram, Activity Diagram* dan *Sequence Diagram.*

3.1.7.1 *Use Case Diagram*

Use case diagram adalah diagram yang menunjukkan fungsionalitas suatu sistem atau kelas dan bagaimana sistem tersebut berinteraksi dengan dunia luar dan menjelaskan sistem secara fungsional yang terlihat pengguna. *Use Case Diagram* dapat dilihat pada Gambar 3.7.



Gambar 3.7 ***Use Case Diagram***

3.1.7.2 Sekenario *Use Case*

Skenario *Use case* bertujuan untuk mendeskripsikan urutan langkah-langkah dalam proses bisnis baik yang dilakukan aktor terhadap sistem maupun yang dilakukan oleh sistem terhadap aktor. Berdasarkan *Use Case Diagram* pada Gambar maka *use case scenario* dijelaskan sebagai berikut :

1. **Skenario *Use Case Login***

   Interaksi antara aktor pengguna dengan *use case* dijelaskan dalam *scenario use case* pada Tabel 3.4.

Tabel 3.4 Skenario *Use Case Login*

Skenario	
Nama *Use Case* : *Login*	
Nama Aktor : Pengguna	
Deskripsi : Menampilkan Halaman Utama *Login*	
Kondisi Awal : Pengguna sudah membuka aplikasi HAC	
Aksi Aktor	**Respon Sistem**
Skenario Normal	
1. Pengguna membuka aplikasi HAC	
	2. Sistem menampilkan halaman *login*
3. Pengguna memasukkan *username* dan *password*	
	4. Cek *validasi*
	5. User dan *password Valid*
	6. Sistem akan masuk ke menu utama
	7. *User* dan *Password* Tidak *Valid*
	8. Menampilkan pesan *error*
Kondisi Akhir	Masuk ke halaman utama aplikasi

2. **Skenario *Use Case* Hidupkan Lampu**

   Interaksi antara aktor pengguna dengan *use case* dijelaskan dalam *scenario use case* pada Tabel 3.5.

Tabel 3.5 Skenario *Use Case* Hidupkan Lampu

Skenario		
Nama Use Case	:	Hidupkan Lampu
Nama Aktor	:	Pengguna
Deskripsi	:	Sistem akan manghidupkan lampu yang diperintahkan oleh pengguna
Kondisi Awal	:	Pengguna sudah memilih lampu yang akan di hidupkan
Aksi Aktor		**Respon Sistem**
Skenario Normal		
1. Pengguna memilih ruangan yang lampunya akan dinyalakan		
		2. Menampilkan status lampu dalam keadaan OFF
3. Menggeser *button* (tombol) ke posisi *ON*		
		4. Menampilkan pesan sukses
Kondisi Akhir		Status lampu menjadi hidup

3. Skenario *Use Case* Matikan Lampu

Interaksi antara aktor pengguna dengan *use case* dijelaskan dalam *scenario use case* pada Tabel 3.6.

Tabel 3.6 Skenario *Use Case* Mematikan Lampu

Skenario		
Nama Use Case	:	Matika Lampu
Nama Aktor	:	Pengguna
Deskripsi	:	Sistem akan mematikan lampu yang diperintahkan oleh pengguna
Kondisi Awal	:	Pengguna sudah memilih lampu yang akan di matikan
Aksi Aktor		**Respon Sistem**
Skenario Normal		
1. Pengguna memilih ruangan yang lampunya akan dimatika		
		2. Menampilkan status lampu dalam keadaan *ON*

3. Menggeser *button* ke posisi *OFF*	4. Menampilkan pesan sukses
Kondisi Akhir	Sataus lampu menjadi mati

4. Skenario *Use Case Report*

Interaksi antara aktor pengguna dengan *use case* dijelaskan dalam *scenario use case* pada Tabel 3.7.

Tabel 3.7 Skenario *Use Case Report*

Skenario		
Nama Use Case	:	*Report*
Nama Aktor	:	Pengguna
Deskripsi	:	Sistem akan menampilkan *report* penggunaan tag RFID yang digunakan untuk membuka kunci
Kondisi Awal	:	Pemilikrumah sudah memilih *Report*
Aksi Aktor		**Respon Sistem**
Skenario Normal		
1. Pengguna memilih menu *report*		
		2. Sistem menampilkan menu *filter report*berdasarkan kurun waktu
3. Pengguna menentukan kurun waktu		
		4. Menampilakn *report* berdasarkan kurun waktu.
Kondisi Akhir		Menampilkan *report* tad RFID yang digunakan untuk membuka kunci

5. Skenario *Use Case* Buka Kunci

Interaksi antara aktor pengguna dengan *use case* dijelaskan dalam *scenario use case* pada Tabel 3.8.

Tabel 3.8 Skenario *Use Case* Buka Kunci

Skenario		
Nama *Use Case*	:	Buka Kunci
Nama Aktor	:	Pengguna
Deskripsi	:	Sistem akan membuka kunci dan mengunci pintu.
Kondisi Awal	:	Pengguna berada pada menu buka kunci
Aksi Aktor		**Respon Sistem**

Skenario	
Skenario Normal	
1. Pengguna mendekatkan tag RFID ke *reader*	
	2. Cek *Validasi*
	3. Valid
	4. Tampil Pesan pada LCD Kunci Terbuka.
	5. Kunci *Selenoid* terbuka
	6. *Sensor* dan *buzzer* (alarm) ruangan mati (*OFF*)
	7. Tidak *Valid*
	8. Tampil Pesan pada LCD Akses Ditolak.
	9. Memberikan notifikasi berupa pesan dan notifikasi pada aplikasi bahwa pintu telah dibuka dengan tag ID yang dipakai.
10. Pengguna kembali mendekatkan *tag* ke *reader* untuk kunci pintu	
	13. Kunci *Selonoid* terkunci kembali
	14. *Sensor* dan *buzzer* (alarm) ruangan aktif.
Kondisi Akhir	Transaksi buka pintu tersimpan di *report*

6. Skenario *Use Case* Set Waktu

Interaksi antara aktor pengguna dengan *use case* dijelaskan dalam *scenario use case* pada Tabel 3.9

Tabel 3.9 Skenario Use Case Set Waktu

Skenario		
Nama Use Case	:	Set Waktu
Nama Aktor	:	Pengguna
Deskripsi	:	Sistem akan menampilkan menu untuk menentukan waktu menyalakan dan mematikan lampu berdasarkan waktu yang di inputkan.

Skenario	
Kondisi Awal :	Pengguna sudah memilih ruangan yang akan di set *on-off* lampunya
Aksi Aktor	**Respon Sistem**
Skenario Normal	
1. Pengguna memilih lampu yang akan di set waktunya	
	2. Menampilkan menu untuk mengisikan jam nyala dan jam mati lampu
3. Pengguna meng set waktu nyala dan mati lampu	
	4. Menampilkan pesan sukses
Kondisi Akhir	Lampu akan menyala dan mati sesuai waktu yang di set pengguna.

7. Skenario *Use Case* Tampilkan KWH

Interaksi antara aktor pengguna dengan *use case* dijelaskan dalam *scenario use case* pada Tabel 3.10

Tabel 3.10 Skenario *Use Case* Tampilkan KWH

Skenario		
Nama Use Case	:	Tampilkan KWH
Nama Aktor	:	Pengguna
Deskripsi	:	Pengguna akan melihat berapa KWH yang telah digunakan
Kondisi Awal	:	Pengguna sudah memilih menu KWH
Aksi Aktor		**Respon Sistem**
Skenario Normal		
1. Pengguna memilih menu KWH		
		2. Sistem menampilkan KWH yang telah digunakan
3. Pengguna memasukan harga per KWH		
		4. Sistem menampilkan harga KWH
Kondisi Akhir		Menampilkan prediksi harga kwh yang harus dibayar.

8. Sekenario *Use Case* Tambah *Tag*

Interaksi antara aktor Admin dengan *use case* dijelaskan dalam *scenario use case* pada Tabel 3.13

Tabel 3.11 Skenario *Use Case* Tambah *Tag*

Skenario		
Nama *Use Case*	:	Tambah Tag
Nama Aktor	:	Admin
Deskripsi	:	Admin akan menambahkan tag yang akan digunakan untuk membuka kunci
Kondisi Awal	:	Admin berada pada menu tambah tag
Aksi Aktor		**Respon Sistem**
Skenario Normal		
1. Admin memilih menu tambah tag		
		2. Menampilkan pesan silahkan scan tag (tambah)
3. Admin mendekatkan tag ke reader untuk di scan		
		5. Tampil PesanTag berhasil ditambahkan
Kondisi Akhir		tag berhasil ditambahakan dan dapat digunakan untuk buka kunci.

9. Skenario *Use Case* Hapus *Tag*

Interaksi antara aktor Admin dengan *use case* dijelaskan dalam *scenario use case* pada Tabel 3.12

Tabel 3.12 Skenario *Use Case* Hapus *Tag*

Skenario		
Nama *Use Case*	:	Hapus Tag
Nama Aktor	:	Admin
Deskripsi	:	Pengguna akan menghapus tag yang digunakan untuk membuka kunci
Kondisi Awal	:	Admin berada pada menu hapus tag
Aksi Aktor		**Respon Sistem**
Skenario Normal		
1. Admin memilih menu tambah tag		

	2. Menampilkan pesan silahkan scan tag (hapus)
3. Admin mendekatkan tag ke reader untuk di scan	
	4. Tampil PesanTag berhasil dihapus
Kondisi Akhir	tag berhasil dihapus dan tidak dapat dugunakan untuk membuka kunci

10. Skenario Use Case Daftar *Tag*

Interaksi antara aktor Admin dengan *use case* dijelaskan dalam *scenario use case* pada Tabel 3.13

Tabel 3.13 Skenario *Use Case* Daftar *Tag*

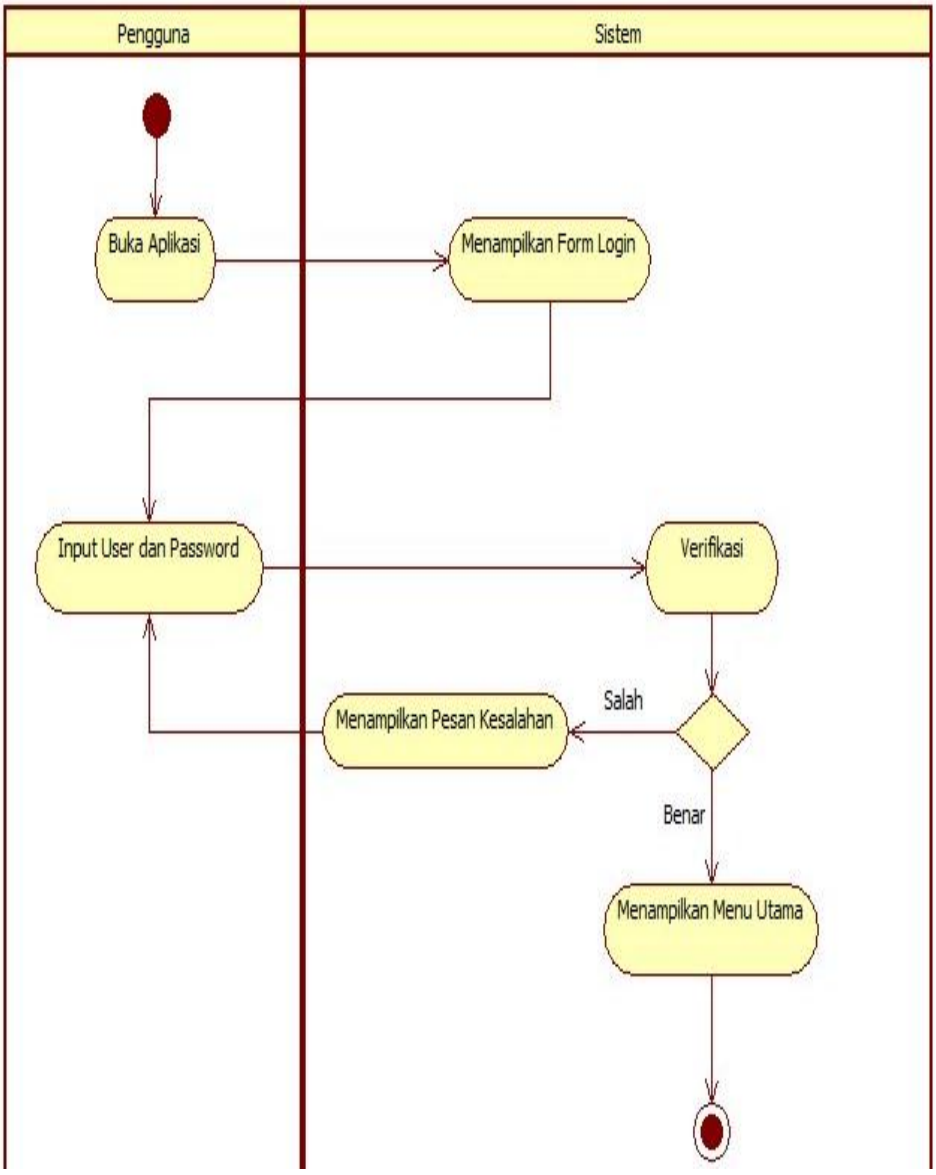
Skenario		
Nama Use Case	:	Daftarkan Tag
Nama Aktor	:	Admin
Deskripsi	:	Admin akan mendaftarkan ID tag agar pengguna mendapatkan informasi siapa saja yang membuka kunci berdasarkan ID tag yang digunakan untuk membuka kuncinya.
Kondisi Awal	:	Admin sudah berada pada menu daftar
Aksi Aktor		**Respon Sistem**
Skenario Normal		
1. Admin memilih menu daftar pengguna		
		2. Sistem menampilkan menu daftar pengguna
3. Admin mendekatkan tag ke reader		
		4. Sistem otomatis mengisi id tag
5. Memasukan nama pengguna		
6. Menekan button register		
		7. Sistem menampilkan pesan registrasi berhasil
Kondisi Akhir		ID tag tersimpan pada database.

3.1.7.3 *Activity Diagram*

Activity diagram memodelkan aliran kerja atau *workflow* dari urutan aktifitas dalam suatu proses yang mengacu pada *Use Case diagram* yang ada.

1. *Activity Diagram Login*

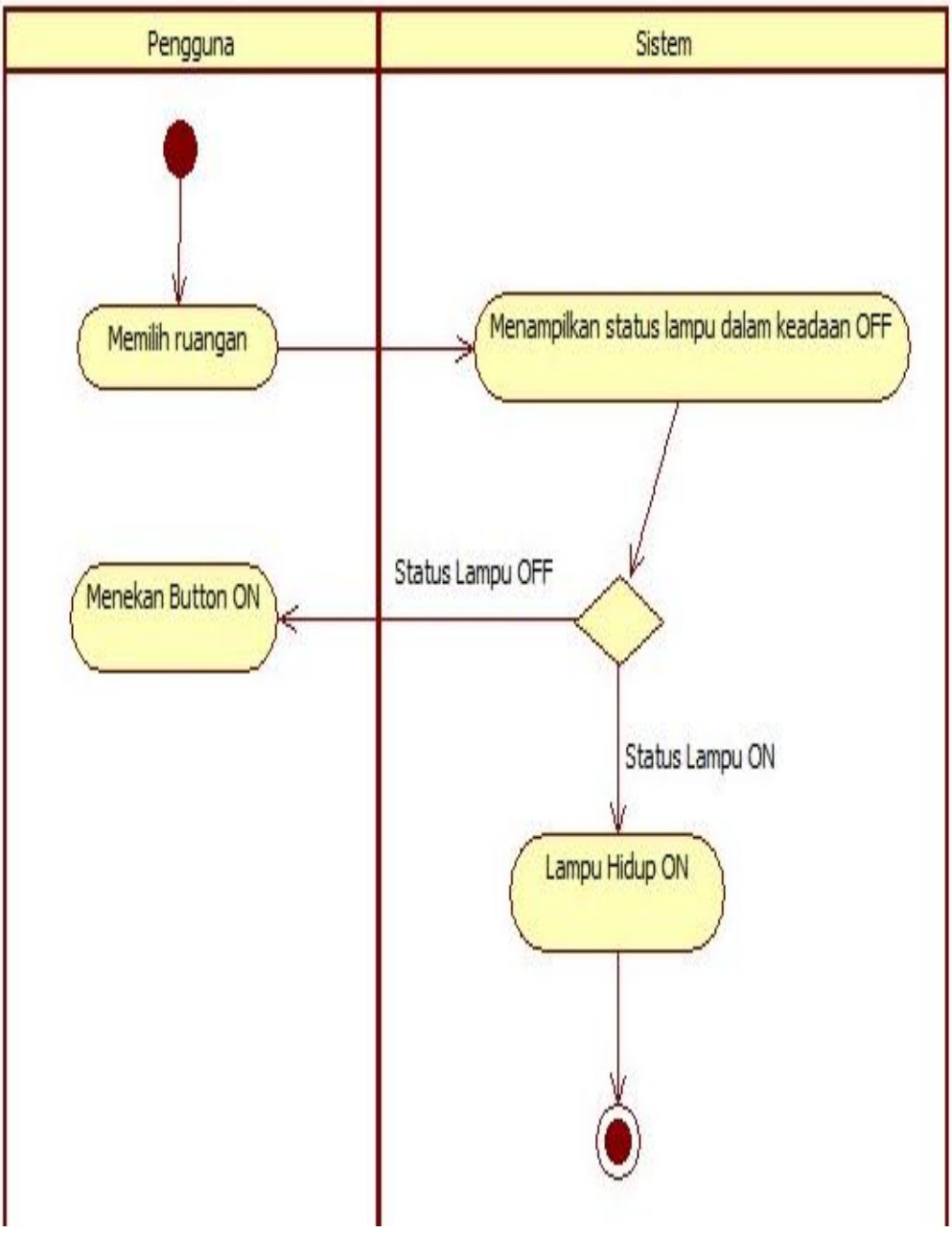
Activity diagram login dapat dilihat pada Gambar 3.8



Gambar 3.8 *Activity Diagram Login*

2. *Activity Diagram* Hidupkan Lampu

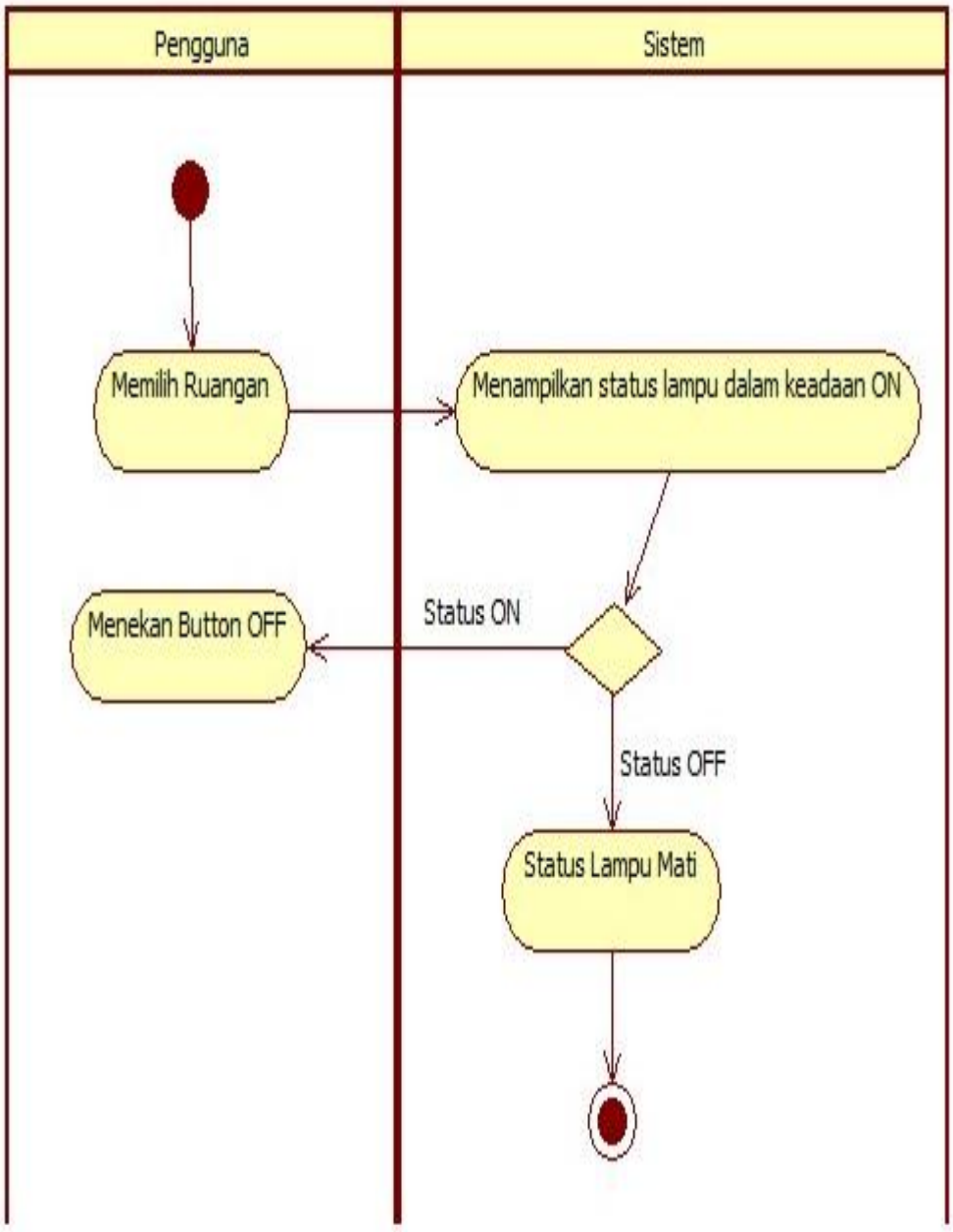
Activity diagram hidupkan lampu dapat dilihat pada Gambar 3.9.



Gambar 3.9 *Activity Diagram* Hidupkan Lampu

3. *Activity Diagram* Matikan Lampu

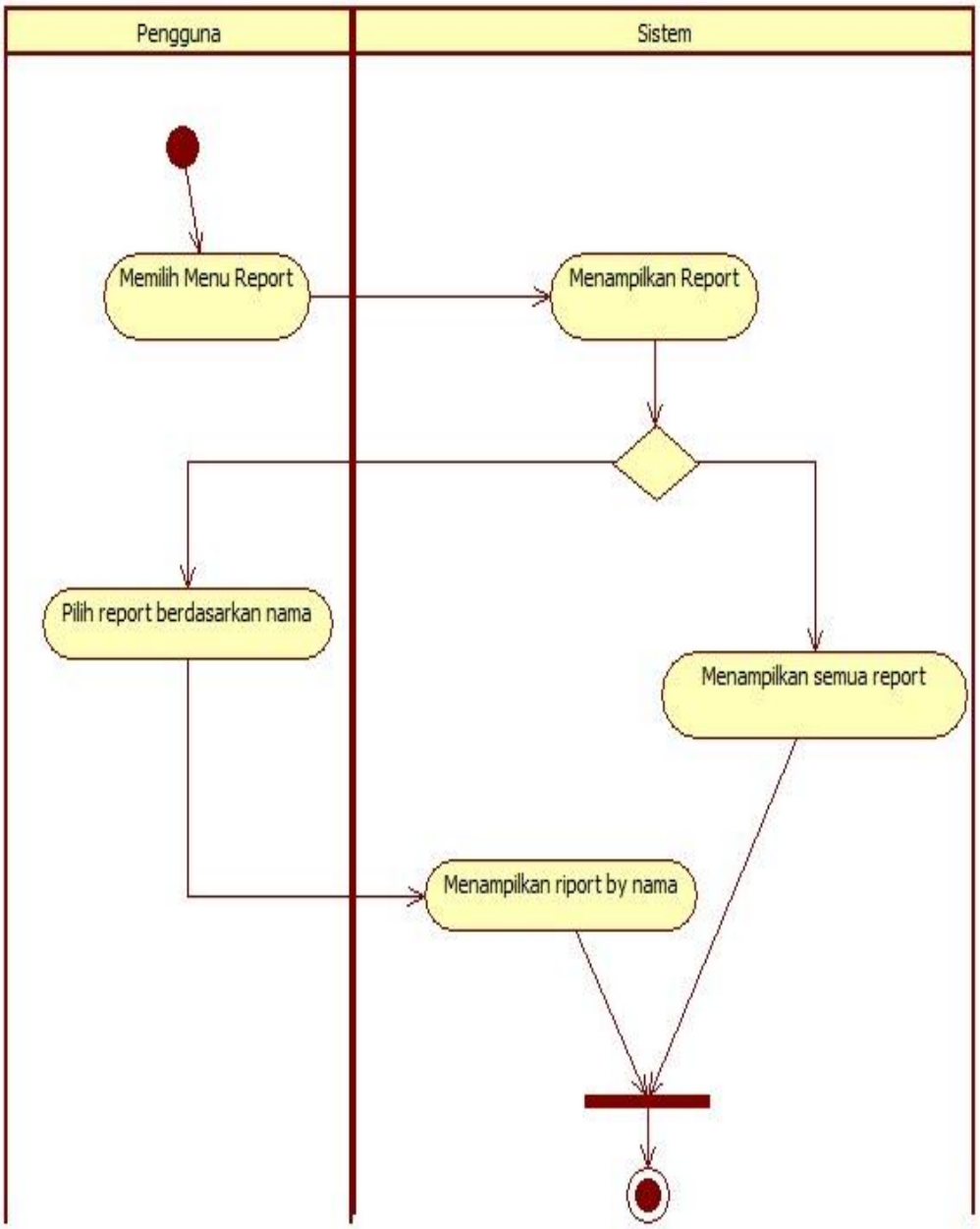
Activity Diagram Matikan Lampu dapat dilihat pada Gambar 3.10



Gambar 3.10 *Activity Diagram* Matikan Lampu

4. *Activity Diagram Report*

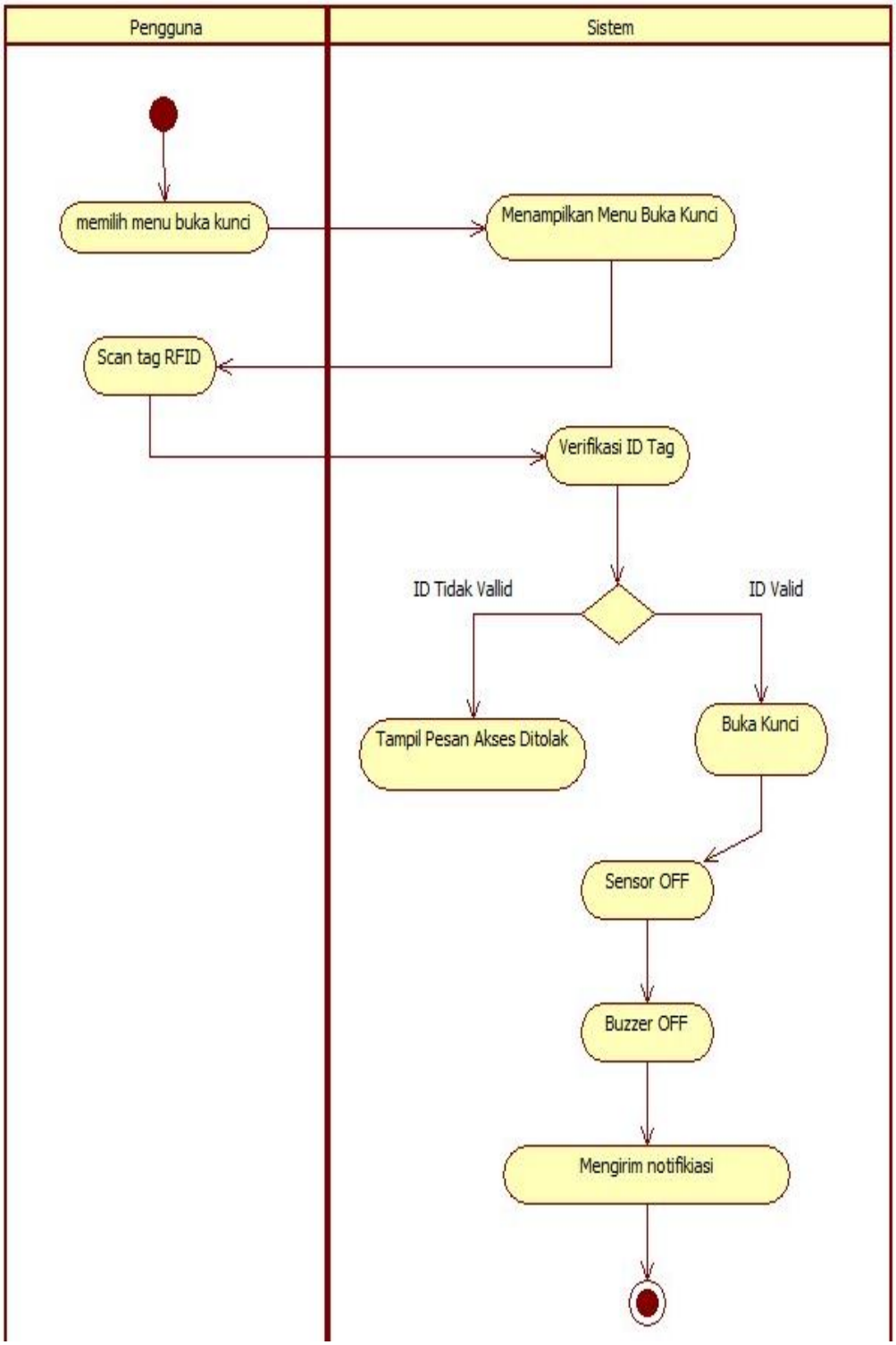
Activity diagram report dapat dilihat pada Gambar 3.11.



Gambar 3.11 *Activity Diagram Report*

5. *Activity Diagram* Buka Kunci

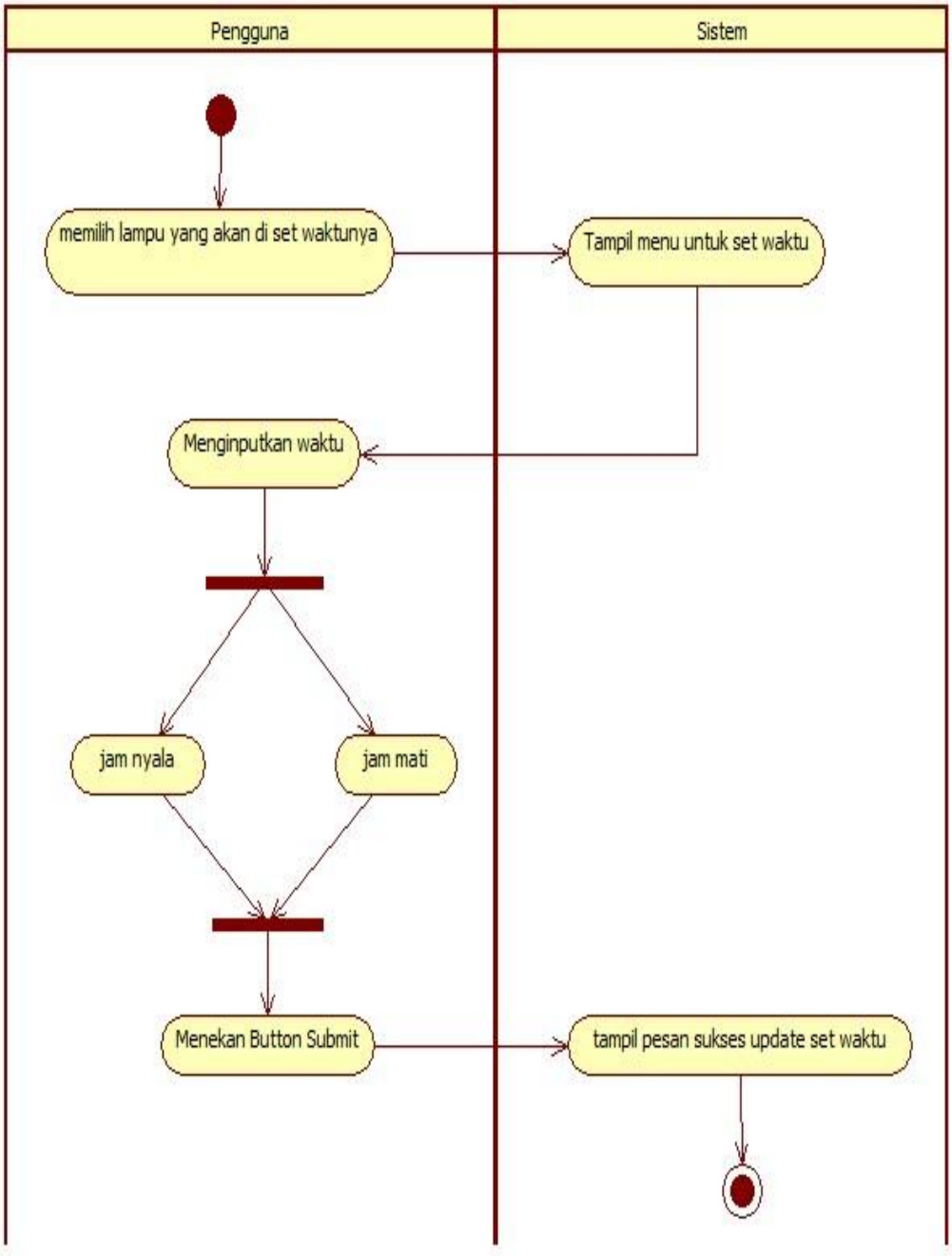
Activity diagram buka kunci dapat dilihat pada Gambar 3.12.



Gambar 3.12 *Activity Diagram* Buka Kunci

6. *Activity Diagram* Set Waktu

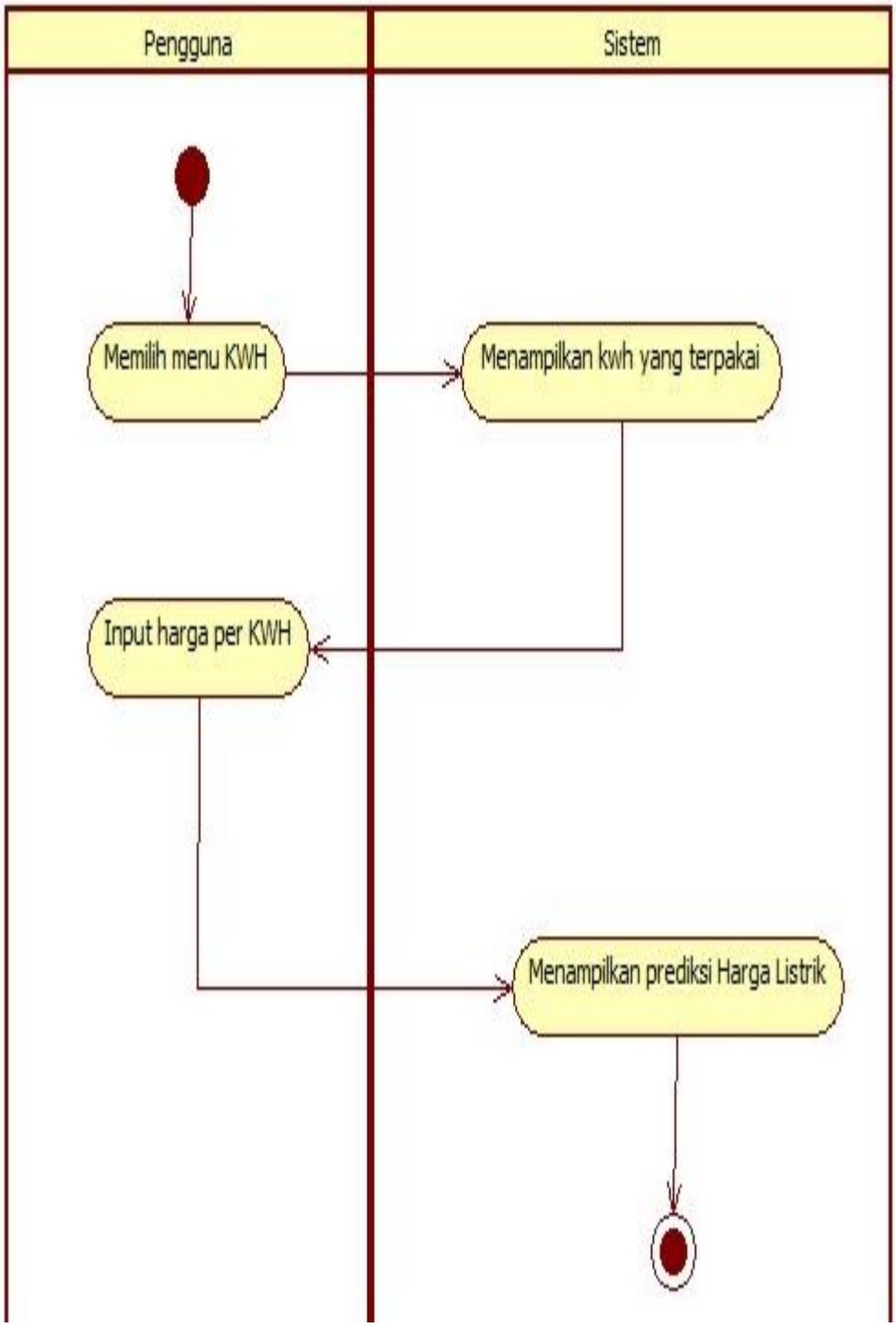
Activity diagram set waktu dapat dilihat pada Gambar 3.13



Gambar 3.13 *Activity Diagram* Set Waktu

7. *Activity Diagram* Tampilkan KWH

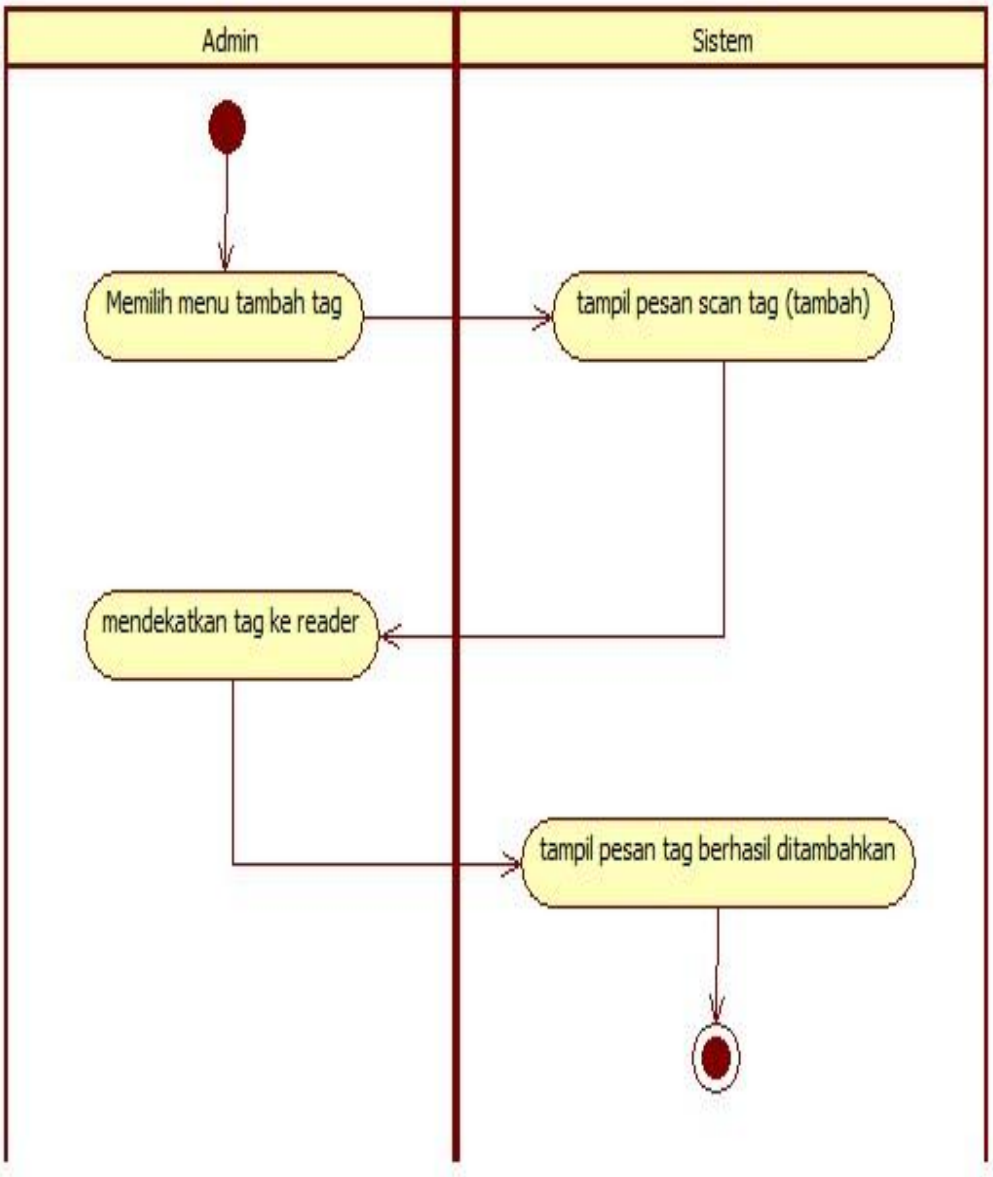
Activity Diagram Tampilkan KWH dapat dilihat pada Gambar 3.14



Gambar 3.14 *Activity Diagram* Tampilkan KWH

8. *Activity Diagram* Tambah *Tag*

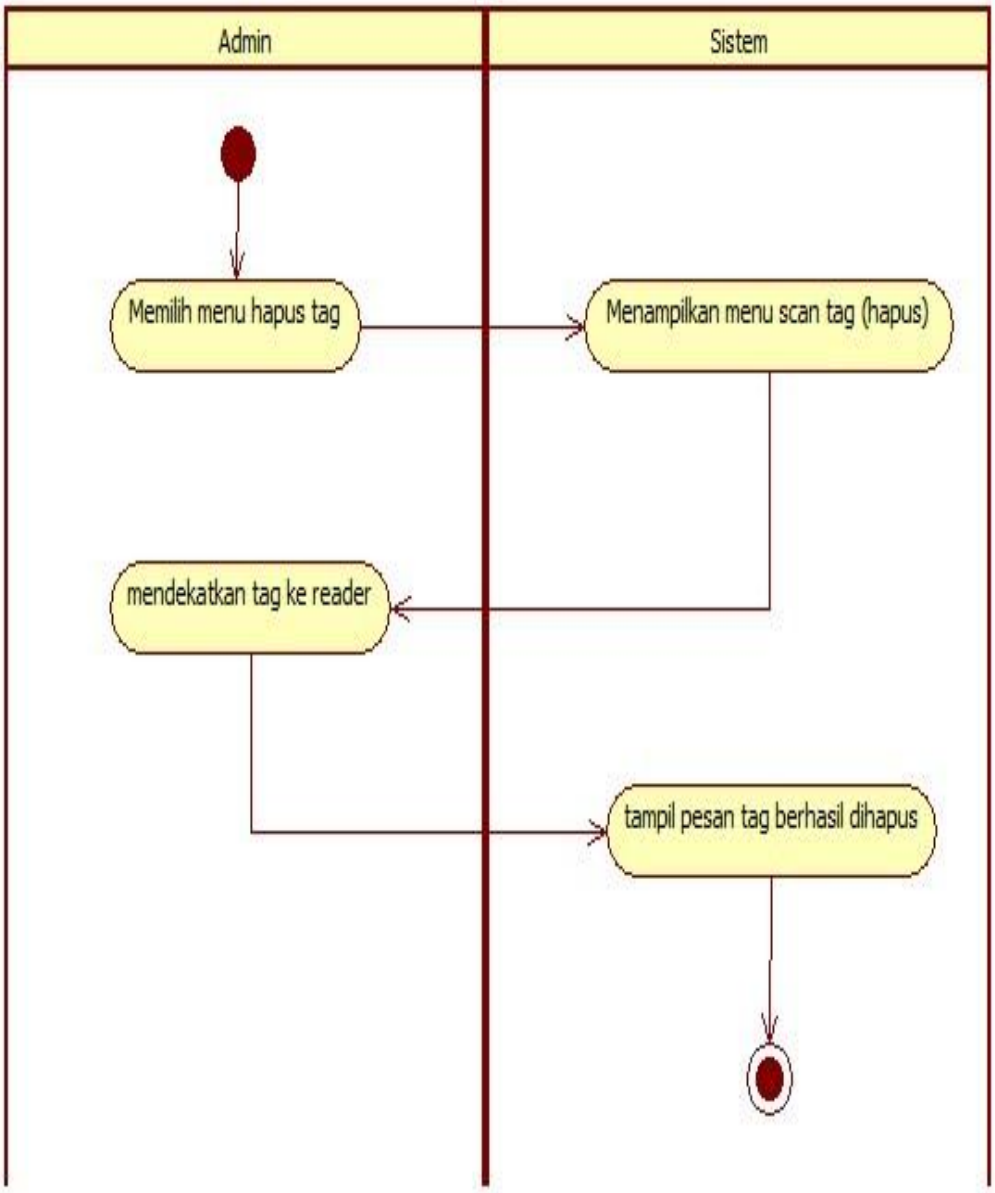
Activity diagram tambah *tag* dapat dilihat pada Gambar 3.15.



Gambar 3.15 *Activity Diagram* Tambah *Tag*

9. *Activity Diagram* Hapus *Tag*

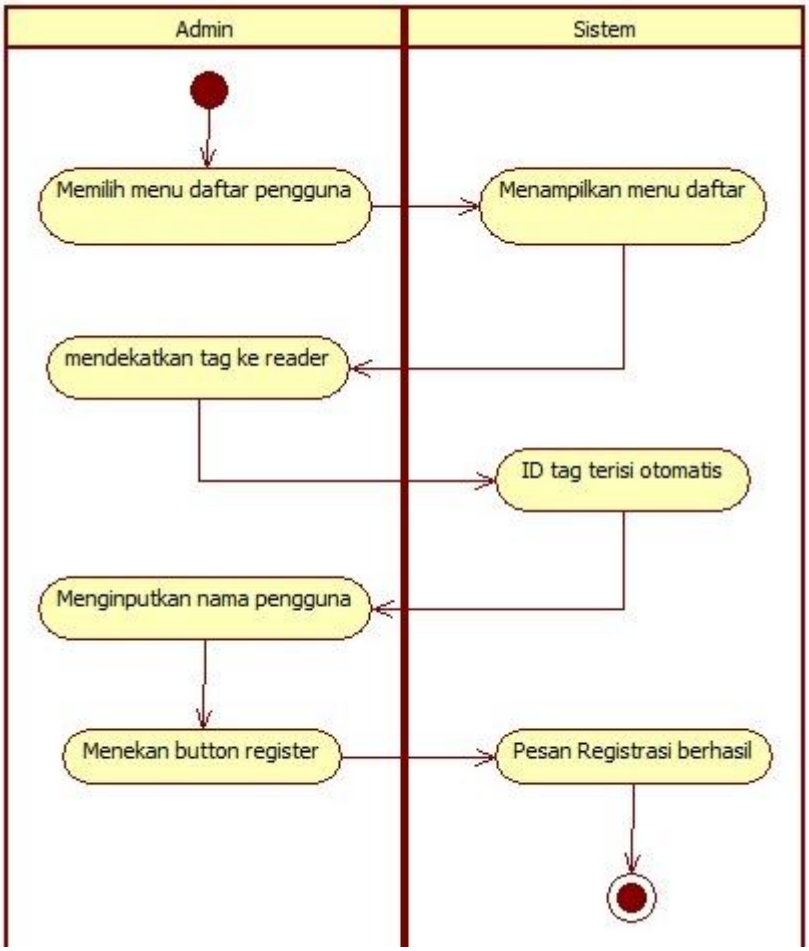
Activity Diagram Hapus *tag* dapat dilihat dari Gambar 3.16.



Gambar 3.16 *Activity Diagram* Hapus *Tag*

10. *Activity Diagram* Daftar *Tag*

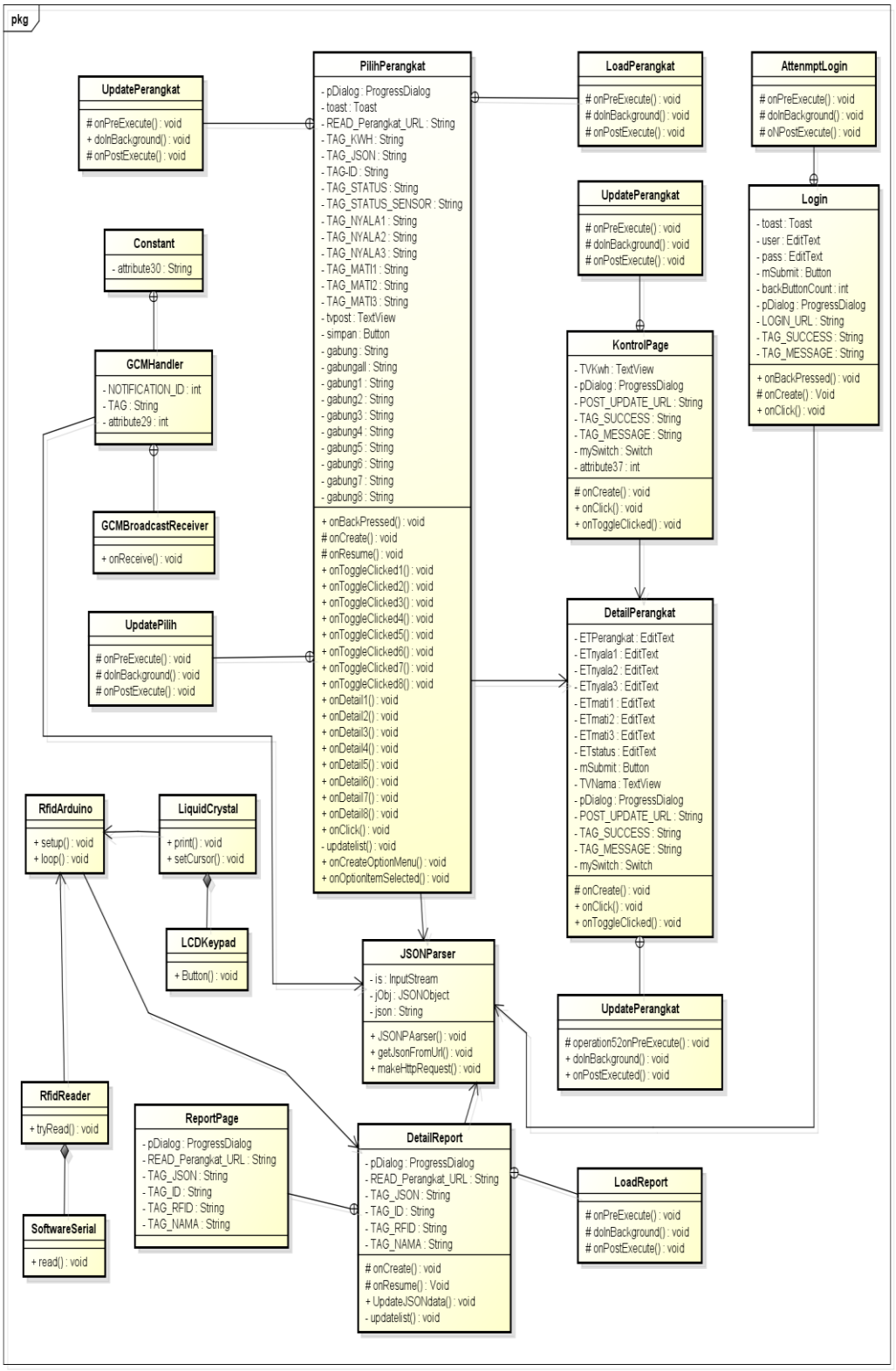
Activity Diagram Daftar *tag* dapat dilihat dari Gambar 3.17.



Gambar 3.17 *Activity Diagram* Daftar *Tag*

3.1.7.4 *Class Diagram*

Class diagram menunjukkan kelas – kelas dalam domain masalah beserta relasinya. *Class diagram* dibuat supaya kelas yang dibuat pada aplikasi sesuai dengan rancangan diagram kelas yang ada pada dokumentasi perancangan. *Class diagram* pada Sistem Pengendali Lampu dan Pemanfaatan RFID Pada Kunci Pintu Elektronik dapat dilihat pada Gambar 3.18.



Gambar 3.18 ***Class Diagram***

Deskripsi tiap class diagram dapat di lihat pada Tabel 3.14.

Tabel 3.14 Deskripsi *Class Diagram*

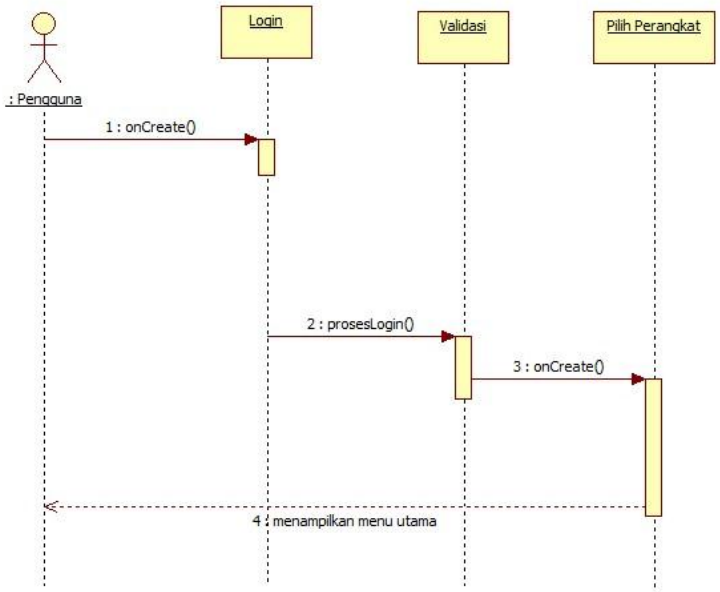
No	Nama Kelas	Keterangan
1	*Login*	Kelas login berfungsi mengambil *username* dan *password* yang telah di simpan pada *database*
2	Pilih Perangkat	Kelas yang berfungsi untuk mengendalikan lampu
3	Detail Perangkat	Kelas yang menampilkan fungsi set waktu untuk otomatisai waktu nyala dan mati lampu
4	*GCM Handler*	Kelas yang berfungsi untuk mengirim notifikasi ke ponsel pengguna
5	*Report Page*	Kelas yang berfungsi untuk menampilkan laporan atau histori dari id tag yang digunakan untuk membuka kunci
6	Detail *Report*	Kelas yang berfungsi untuk menampilkan report berdasarkan ID tag yang digunakan
7	LCDKeypad	Kelas yang berfungsi untuk menampilkan menu tambah dan hapus tag
7	Rfid_Arduino	Kelas yang berfungsi mengirim untuk memverifikasi tag dan membuka kunci serta mengirim notifikasi.

3.1.7.5 *Sequence Diagram*

Sequence Diagram digunakan untuk menggambarkan perilaku pada sebuah skenario. Diagram ini menunjukan sejumlah contoh objek dan pesan yang diletakan diantara objek-objek ini didalam *Use case*. Oleh karena itu, untuk menggambarkan *sequence diagram* maka harus deketahui objek-objek yang terlibat dalam sebuah *Use case* beserta metode-metode yang dimiliki oleh kelas yang diinstansiasi menjadi objek itu.

1. ***Sequence Diagram Login***

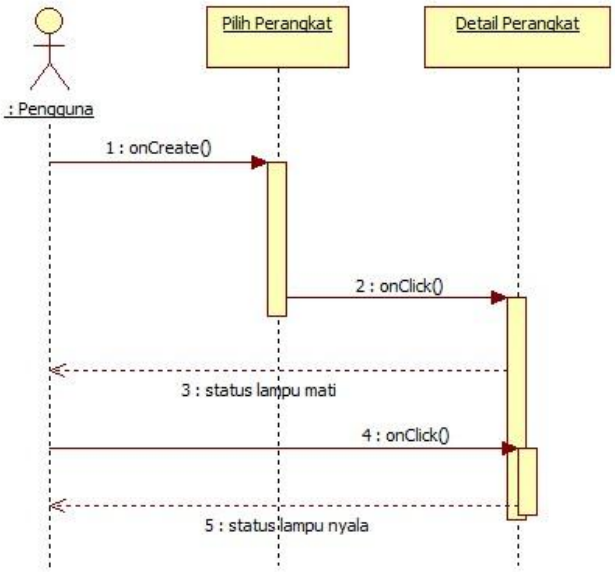
Sequence Diagram Login dapat dilihat pada Gambar 3.19



Gambar 3.19 *Sequence Diagram Login*

2. ***Sequence Diagram* Hidupkan Lampu**

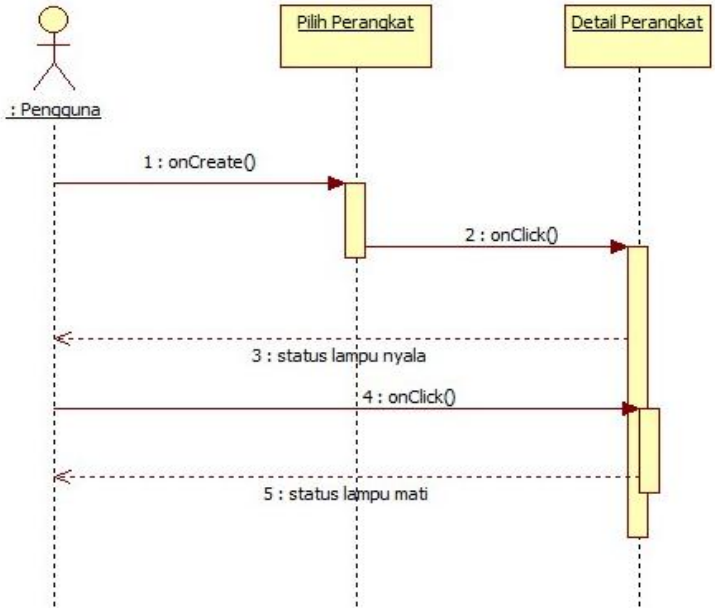
Sequence Diagram Hidupkan Lampu dapat dilihat pada Gambar 3.20



Gambar 3.20 *Sequence Diagram* Hidupkan Lampu

3. *Sequence Diagram* Matikan Lampu

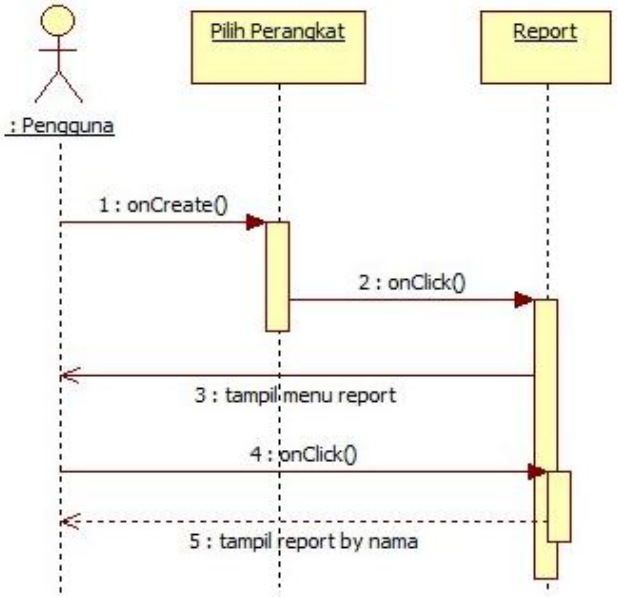
Sequence diagram matikan lampu dapat dilihat pada Gambar 3.21



Gambar 3.21 *Sequence Diagram* Matikan Lampu

4. *Sequence Diagram Report*

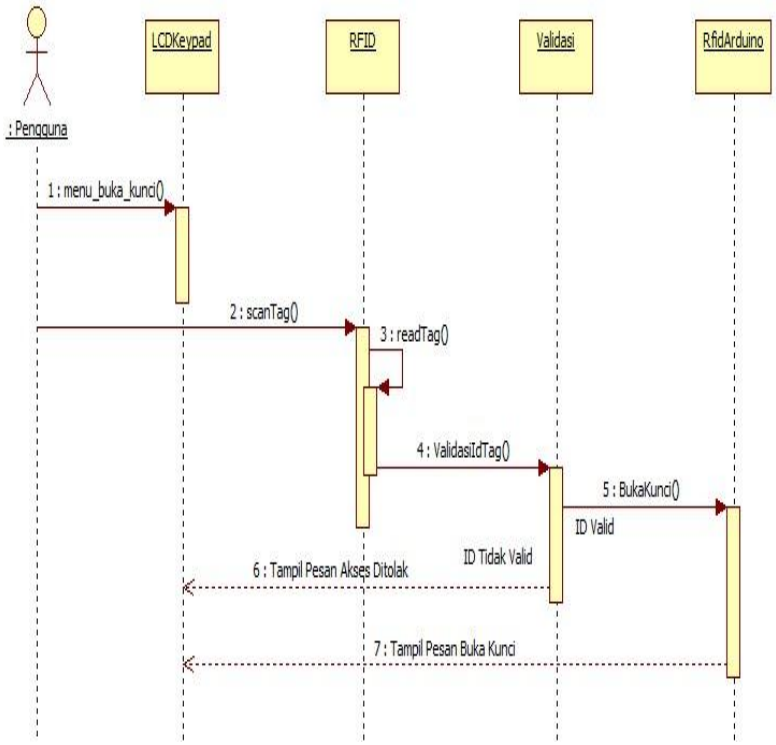
Sequence diagram report dapat dilihat pada Gambar 3.22.



Gambar 3.22 *Sequence Diagram Report*

5. *Sequence Diagram* Buka Kunci

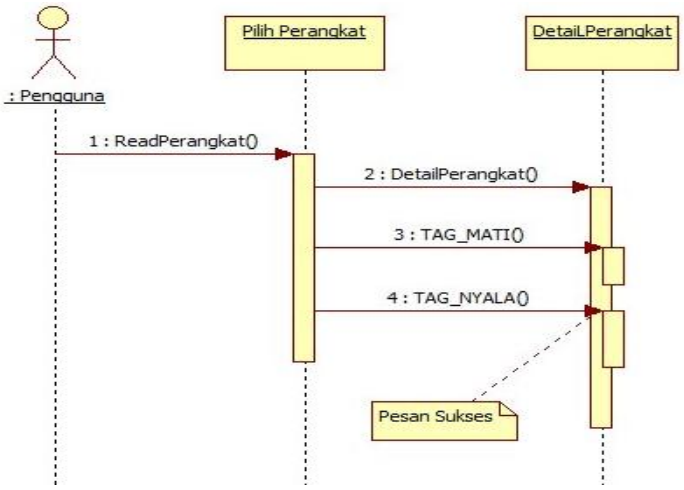
Sequence diagram buka kunci dapat dilihat pada Gambar 3.23.



Gambar 3.23 *Sequence Diagram Buka Kunci*

6. *Sequence Diagram* Set Waktu

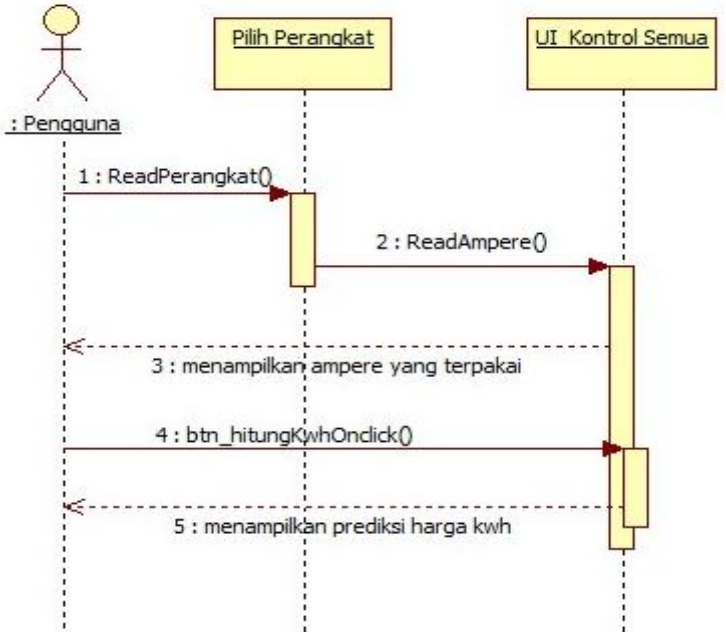
Sequence diagram tambah tag dapat dilihat pada Gambar 3.24



Gambar 3.24 Set Waktu

7. *Sequence Diagram* Tampilkan KWH

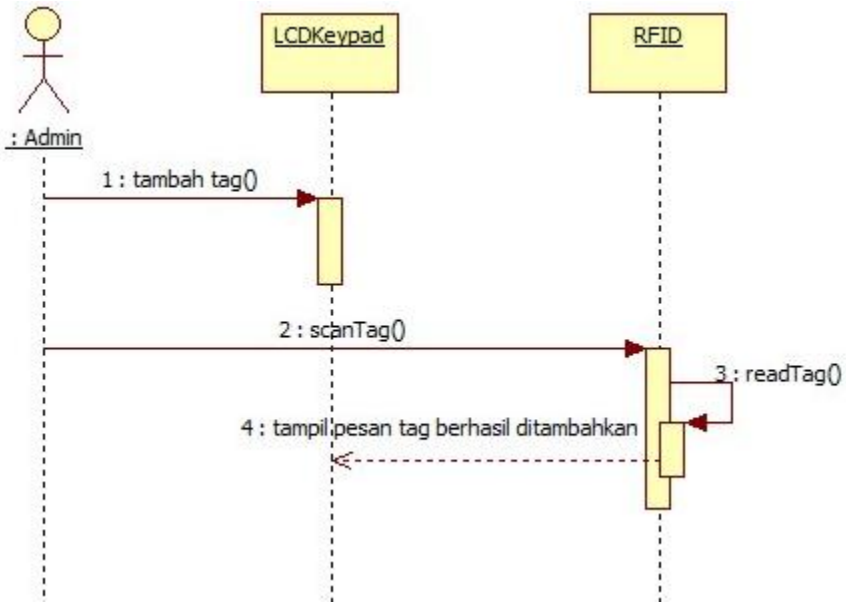
Sequence diagram Tampilkan KWH dapat dilihat pada Gambar 3.25



Gambar 3.25 *Sequence Diagram* Tampilkan KWH

8. *Sequence Diagram* Tambah *Tag*

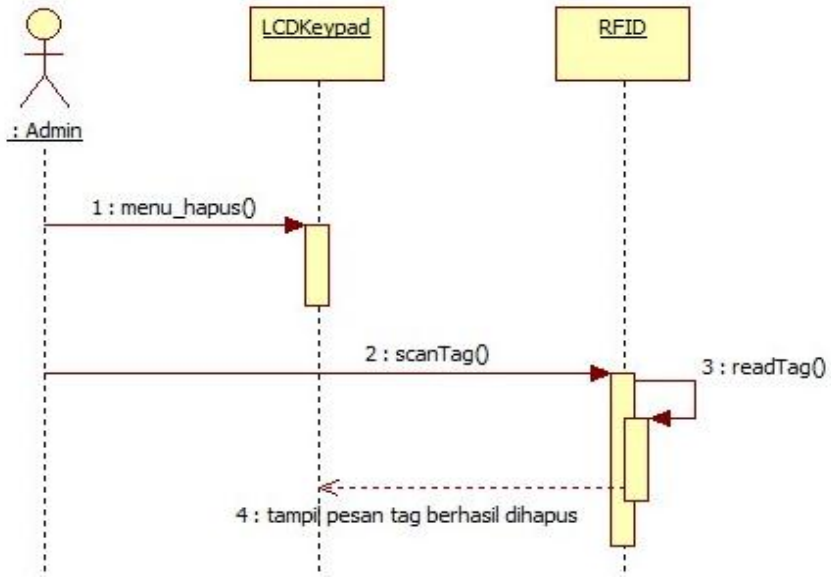
Sequence diagram tambah tag dapat dilihat pada Gambar 3.26



Gambar 3.26 *Sequence Diagram* Tambah *Tag*

9. *Sequence Diagram* Hapus *Tag*

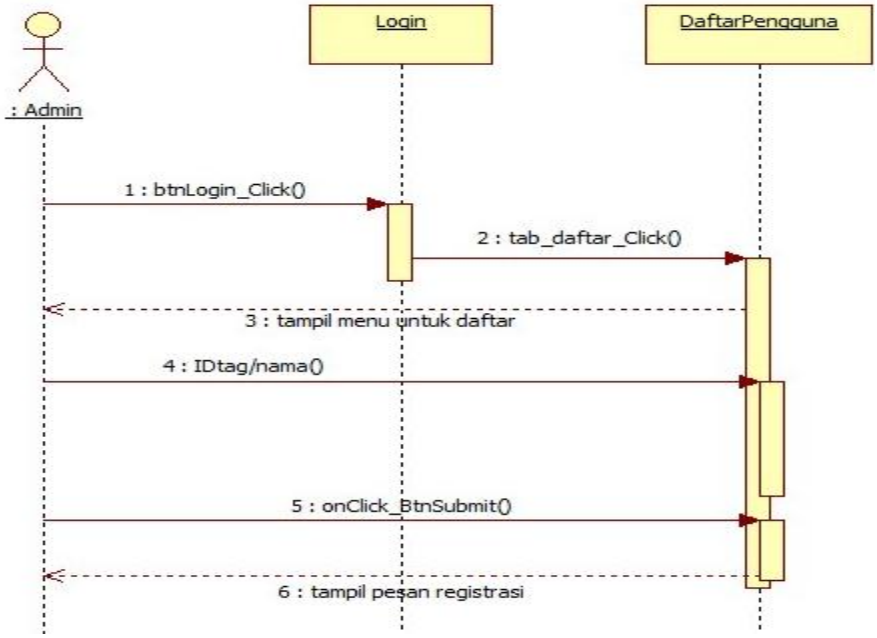
Sequence diagram hapus *tag* dapat dilihat pada Gambar 3.27



Gambar 3.27 *Sequence Diagram* Hapus *Tag*

10. *Sequence Diagram* Daftar *Tag*

Sequence diagram daftar tag dapat dilihat pada Gambar 3.28.



Gambar 3.28 *Sequence Diagram* Daftar *Tag*

3.2 Perancangan Sistem

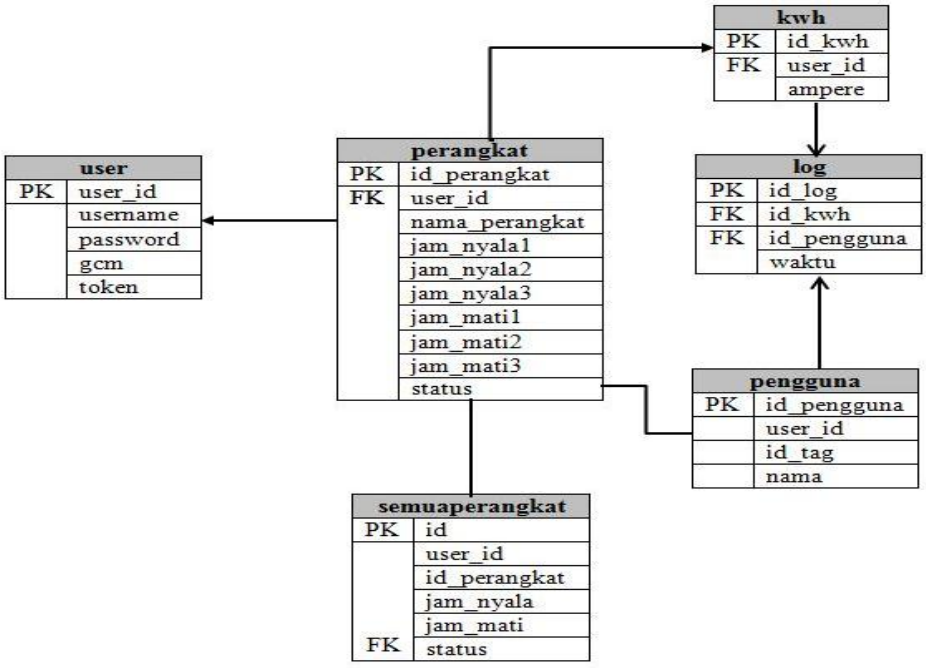
Perancangan merupakan penggambaran, perencanaan, dan pembuatan sketsa atau pengaturan dari beberapa elemen yang terpisah ke dalam suatu kesatuan yang utuh.

3.2.1 Perancangan Basis Data

Basis data (*database*) merupakan kumpulan dari data yang saling berhubungan satu dengan yang lainnya. Perancangan basis data meliputi relasi antar tabel dan deskripsi struktur tabel.

3.2.1.1 Skema Relasi

Skema relasi merupakan rangkaian hubungan antara dua tabel atau lebih pada sistem basis data. Berikut adalah perancangan skema relasi untuk aplikasi pengendali lampu dan pemanfaatan rfid pada kunci pintu elektronik dapat dilihat pada Gambar 3.29



Gambar 3.29 Skema Relasi

3.2.1.2 Struktur Tabel

Struktur tabel menggambarkan detail tabel yang berisi field, tipe data, panjang data, dan keterangan lainnya. Berikut ini deskripsi dari masing-masing tabel tersebut. Berikut ini adalah struktur tabel dari tabel-tabel yang ada pada *database* :

1. Tabel User

Tabel 3.15 Tabel *User*

No	Nama Field	Tipe Data	Ukuran	Key
1	user_id	Int	11	Primarykey
2	username	Varchar	30	
3	password	Varchar	8	
4	gcm	Text		
5	token	Int	255	

2. Tabel Pengguna

Tabel 3.16 Tabel Pengguna

No	Nama Field	Tipe Data	Ukuran	Key
1	id_pengguna	Int	11	Primarykey
2	user_id	Int	11	
3	id_tag	Varchar	30	
4	nama	varchar	50	

3. Tabel Perangkat

Tabel 3.17 Tabel Perangkat

No	Nama Field	Tipe Data	Ukuran	Key
1	id_perangkat	Int	3	Primarykey
2	user_id	Int	11	
3	nama_perangkat	Varchar	50	
4	jam_nyala1	Datetime		
5	jam_nyala2	Datetime		
6	jam_nyala3	Datetime		
7	jam_mati1	Datetime		
8	jam_mati2	Datetime		
9	jam_mati3	Datetime		

No	Nama Field	Tipe Data	Ukuran	Key
10	status	Int	1	

4. Tabel Semua Perangkat

Tabel 3.18 Tabel Semua Perangkat

No	Nama Field	Tipe Data	Ukuran	Key
1	id	Int	11	Primarykey
2	id_perangkat	Int	3	Foreignkey
3	user_id	Int	11	
4	jam_nyala	Datetime		
5	jam_mati	Datetime		
6	status	Datetime	1	

5. Tabel KWH

Tabel 3.19 Tabel KWH

No	Nama Field	Tipe Data	Ukuran	Key
1	id_kwh	Int	11	Primarykey
2	user_id	Int	11	Foreignkey
3	ampere	Double		
4	waktu	Datetime		

6. Tabel log

Tabel 3.20 Tabel *Log*

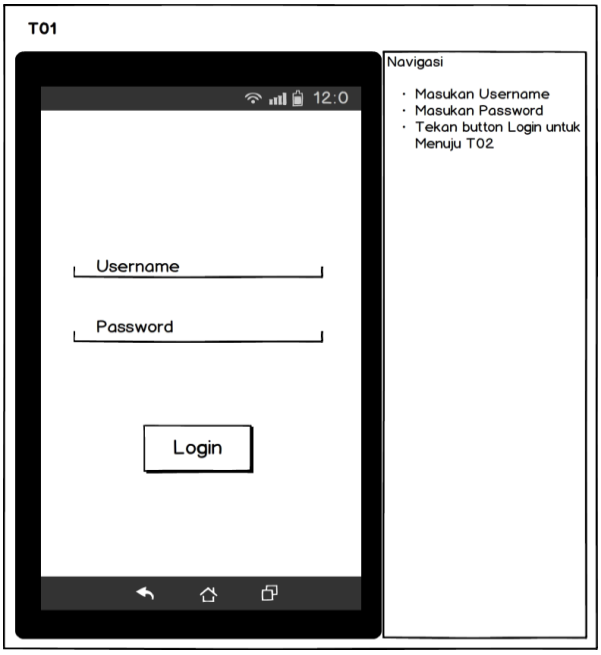
No	Nama Field	Tipe Data	Ukuran	Key
1	Id_log	Int	11	Primarykey
2	id_kwh	Int	3	Foreignkey
3	id_pengguna	Int	11	
4	waktu	Datetime		

3.2.2 Perancangan Antarmuka

Antarmuka (*interface*) merupakan bentuk tampilan dari program yang tampil pada perangkat android yang bertujuan memberikan gambaran tentang aplikasi yang akan dibangun, sehingga akan memudahkan pengimplementasikan aplikasi sesuai ukuran layar dan memudahkan dalam pembuatan aplikasi.

3.2.2.1 Perancangan Antarmuka *Login*

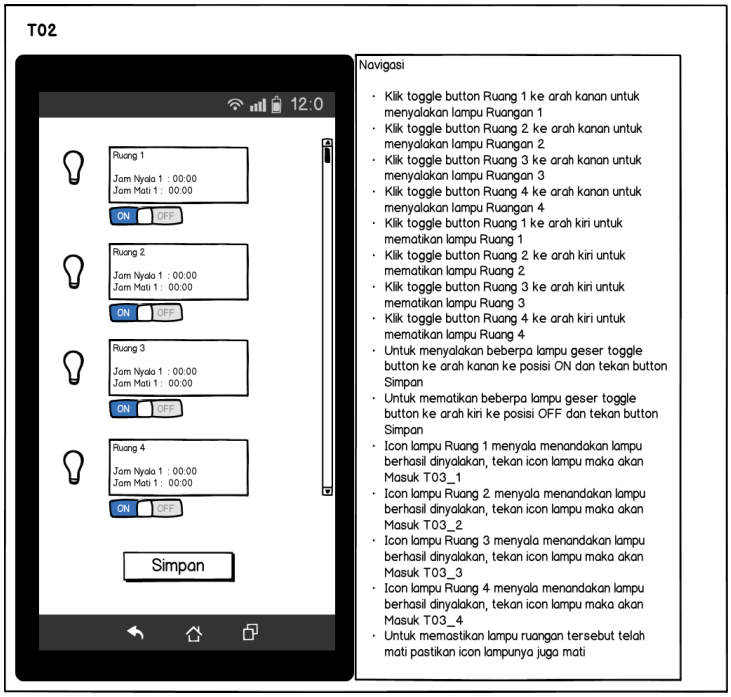
Perancangan antarmuka *login* ketika akan masuk menu utama *user* diminta memasukan *username* dan *password* yang sudah terverifikasi di *database*.



Gambar 3.30 Perancangan Antarmuka *Login*

3.2.2.2 Perancangan Antarmuka Menu Utama

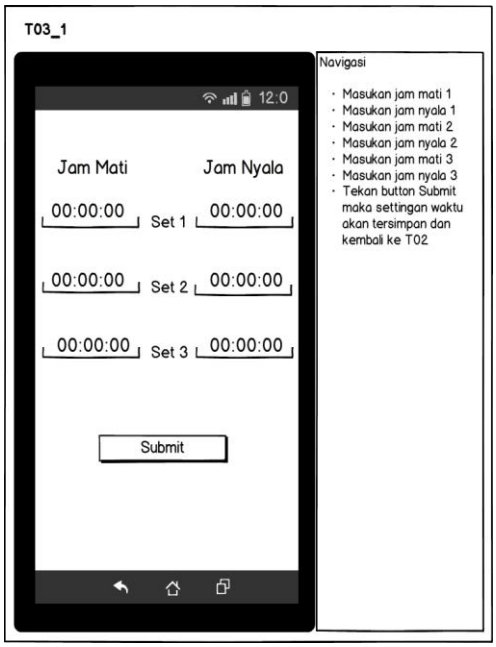
Perancangan antarmuka menu utama ini dimana pengguna bisa mengendalikan perangkat yang diinginkan.



Gambar 3.31 Perancangan Antarmuka Menu Utama

3.2.2.3 Perancangan Antarmuka Set Waktu Lampu R. Tamu

Perancangan Antarmuka dimana pengguna bisa mengendalikan lampu ruang tamu dengan otomatosai waktu nyala dan mati lampu.



Gambar 3.32 Perancangan Antarmuka Set Waktu Lampu R. Tamu

3.2.2.4 Perancangan Antarmuka Set Waktu Lampu R. Keluarga

Perancangan Antarmuka dimana pengguna bisa mengendalikan lampu ruang keluarga dengan otomatosai waktu nyala dan mati lampu.



Gambar 3.33 Perancangan Antarmuka Set Waktu Lampu R.Keluarga

3.2.2.5 Perancangan Antarmuka Set Waktu Lampu Depan Rumah

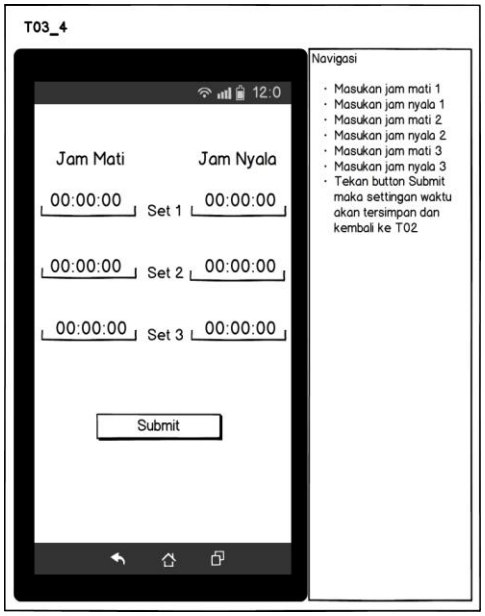
Perancangan Antarmuka dimana pengguna bisa mengendalikan lampu depan rumah dengan otomatosai waktu nyala dan mati lampu.



Gambar 3.34 Perancangan Antarmuka Set Waktu Lampu Depan Rumah

3.2.2.6 Perancangan Antarmuka Set Waktu Lampu Belakang Rumah

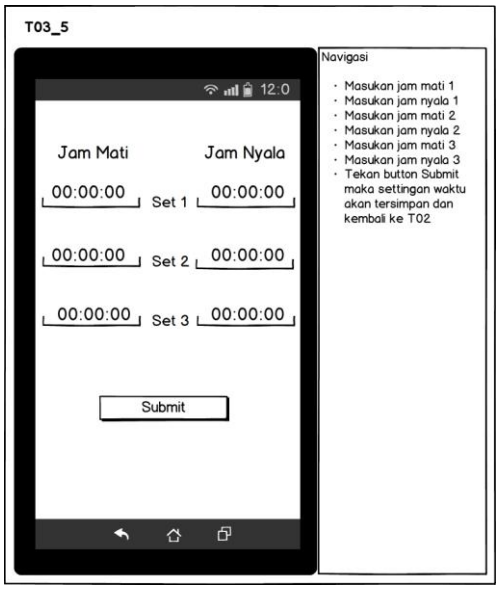
Perancangan Antarmuka dimana pengguna bisa mengendalikan lampu belakang rumah dengan otomatosai waktu nyala dan mati lampu.



Gambar 3.35 Perancangan Antarmuka Set Waktu Lampu Belakang Rumah

3.2.2.7 Perancangan Antarmuka Set Waktu Lampu Toilet

Perancangan Antarmuka dimana pengguna bisa mengendalikan lampu toilet dengan otomatosai waktu nyala dan mati lampu.



Gambar 3.36 Perancangan Antarmuka Set Waktu Lampu Toilet

3.2.2.8 Perancangan Antarmuka Set Waktu Lampu Kamar Tidur

Perancangan Antarmuka dimana pengguna bisa mengendalikan lampu kamar tidur dengan otomatosai waktu nyala dan mati lampu.



Gambar 3.37 Perancangan Antarmuka Set Waktu Lampu Kamar Tidur

3.2.2.9 Perancangan Antarmuka Set Waktu Lampu Dapur

Perancangan Antarmuka dimana pengguna bisa mengendalikan lampu dapur dengan otomatosai waktu nyala dan mati lampu.



Gambar 3.38 Perancangan Antarmuka Set Waktu Lampu Dapur

3.2.2.10 Perancangan Antarmuka Set Waktu Lampu R. Makan

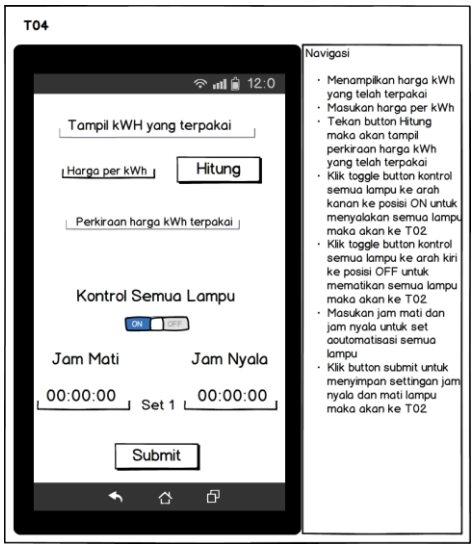
Perancangan Antarmuka dimana pengguna bisa mengendalikan lampuruang makan dengan otomatosai waktu nyala dan mati lampu.



Gambar 3.39 Perancangan Antarmuka Set Waktu Lampu R. Makan

3.2.2.11 Perancangan Antarmuka Kontrol Semua Lampu dan KWH

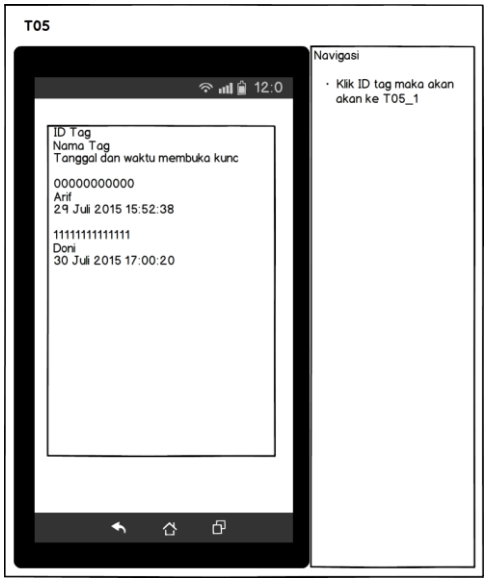
Perancangan antarmuka untuk menampilkan menu kontrol semua lampu dan perkiraan kWh yang telah terpakai.



Gambar 3.40 Perancangan Antarmuka Kontrol Semua Lampu dan KWH

3.2.2.12 Perancangan Antarmuka Report Tag RFID

Perancangan antarmuka untuk menampilkan laporan dari tag RFID yang digunakan untuk membuka kunci.



Gambar 3.41 Perancangan Antarmuka Report Tag RFID

3.2.2.13 Perancangan Antarmuka Detail Report

Perancangan Antarmuka dimana pengguna melihat detail report berdasarkan ID tag yang digunakan.



Gambar 3.42 Perancangan Antarmuka Detail Report

3.2.2.14 Perancangan Antarmuka *Notifikasi*

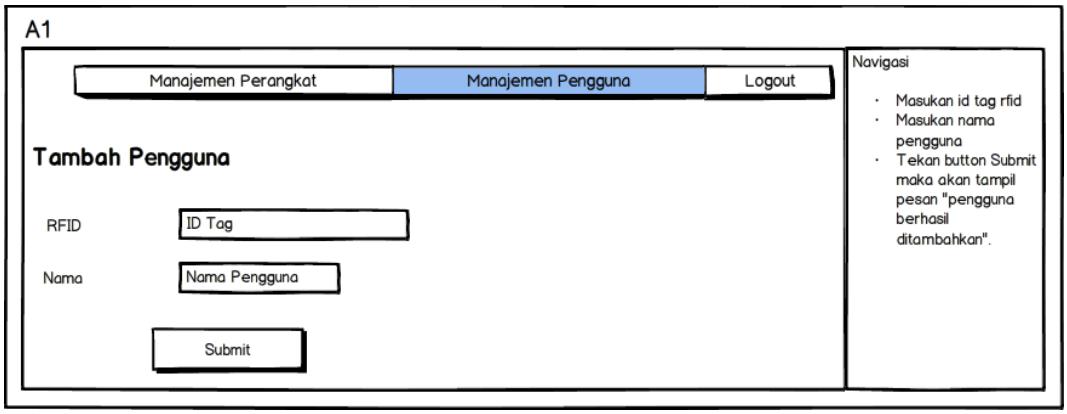
Perancangan Antarmuka dimana pengguna bisa melihat *notifikasi* atau pemberitahuan ketika ada transaksi seperti *tag* yang digunakan untuk membuka kunci pintu rumah.



Gambar 3.43 Perancangan Antarmuka *Notifikasi*

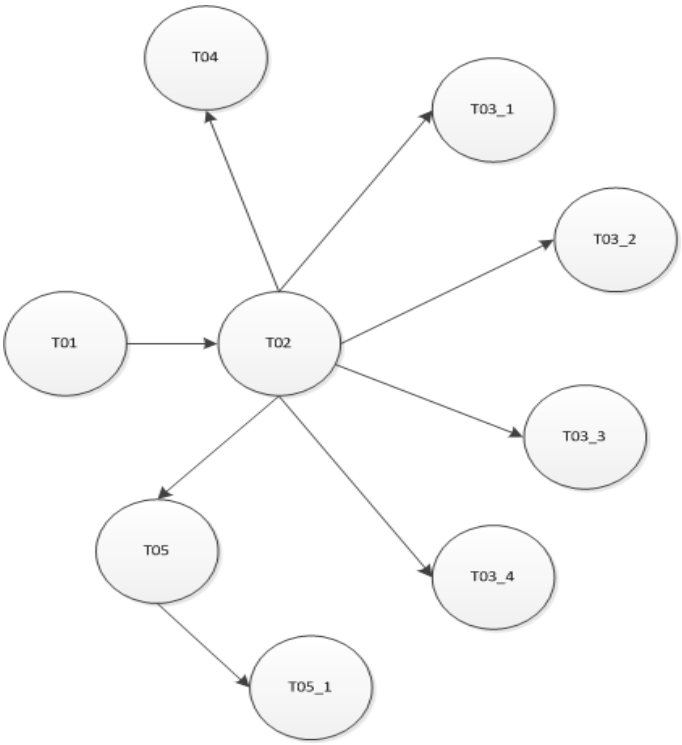
3.2.2.15 Perancangan Antarmuka Daftar Pengguna (Daftar Tag)

Perancangan antarmuka dimana admin bisa mendaftarkan tag untuk buka kunci supaya tag terdaftar pada database dan pengguna dapat memperoleh informasi ketika tag digunakan untuk membuka kunci.



Gambar 3.44 Perancangan Antarmuka Daftar Pengguna

3.2.3 Jaringan Semantik



Gambar 3.45 Jaringan Semantik