

PERANCANGAN APLIKASI SMART HOME BERBASIS ANDROID UNTUK PENGENDALIAN KEAMANAN RUMAH DENGAN MENGGUNAKAN ANDROID STUDIO

Hadyan Setiawan^{*)}, Aghus Sofwan, and Yuli Christyono

Departemen Teknik Elektro, Universitas Diponegoro
Jl. Prof. Sudharto, SH, Kampus UNDIP Tembalang, Semarang 50275, Indonesia

^{*)}E-mail: hadayn.setiawan@gmail.com

Abstrak

Semakin maraknya tindak pencurian membuat pemilik rumah khawatir saat meninggalkan rumahnya. Tidak jarang pemilik rumah melakukan pengecekan ganda pada setiap pintu rumah untuk memastikan semua pintu telah terkunci dengan baik. Saat ini sedang berkembang sebuah teknologi rumah pintar atau smart home yang membuat penggunaannya lebih mudah memantau rumah mereka. Smart Home merupakan dimana semua perangkat dan aplikasi pada suatu rumah terintegrasi sehingga memudahkan kita untuk mengontrol sebagian dari perangkat rumah. Smart Home ada dari beberapa dekade dalam hal keamanan dan kontrol perangkat sederhana. Hal seperti ini lebih dikenal dengan metode Internet of Things atau yang lebih dikenal dengan IoT. Salah satu fitur smart home yang mendukung peningkatan tingkat keamanan pada suatu rumah ialah fitur door lock system. Door lock system memungkinkan pengguna dapat mengunci, menutup, memantau, hingga menyalakan alarm secara otomatis apabila terjadi pembobolan pada rumah tersebut. Fokus Penelitian ini adalah implementasi user interface pengendalian keamanan rumah dengan door lock system kendali logika on-off dengan menggunakan mikrokontroler Atmega16. Kemudian diintegrasikan dengan aplikasi android melalui internet dengan metode IoT, sehingga sistem keamanan dirumah dapat dimonitor dan dikendalikan secara real time dengan menggunakan sensor PIR. Dengan fitur door lock system pada suatu rumah diharapkan dapat mengurangi dan mencegah tindak pencurian.

Kata kunci : Smart home, Android , Door Lock System, Security System, Mikrokontroler ATMEGA16.

Abstract

The more rampant act of theft makes people worried when leaving home. Not infrequently people double check on every door of the house to make sure all doors have been locked. Currently developing a smart home technology that makes users more easily monitor their homes. Smart Home is where all devices and applications in homes are integrated to allow us to control the home device. Smart Home exists from several decades in terms of security and simple device control. This is better known as the Internet of Things method or better known as IoT. One of the smart home features that support the increased level of security in a house is the door lock system. Door lock system allows users to lock, close, monitor, to automatically turn on the alarm in the house. The focus of this Research is the implementation of user interface of home security control with door lock system of on-off logic control by using Atmega16 microcontroller. Then integrated with android application via internet with IoT method, home security system can be monitored and controlled in real time by using PIR sensor. With this system in a house is expected to reduce and prevent theft.

Keywords: embedded coding, video stream, raspberry pi

1. Pendahuluan

Teknologi mengalami perkembangan yang sangat cepat, dengan banyaknya aplikasi yang mempermudah kinerja manusia. Menyuguhkan kemudahan dalam mengakses informasi secara cepat dan akurat, dengan berbagai macam jenis dan fungsi yang dibutuhkan masyarakat saat ini.

Smart Home merupakan dimana semua perangkat dan aplikasi pada suatu rumah terintegrasi sehingga memudahkan untuk mengontrol sebagian besar dari

perangkat rumah. Security Smart Home telah mengalami banyak perkembangan dalam beberapa dekade dalam pengamanan rumah dan antisipasi saat terjadinya pembobolan oleh pihak yang tidak diinginkan. Pada saat ini Smart Home memungkinkan kita untuk mengontrol penuh rumah darimana dan kapanpun diinginkan menggunakan aplikasi Android.

Pencurian merupakan masalah keamanan yang banyak terjadi di Indonesia. Berdasarkan data Badan Pusat Statistik, pada tahun 2013 terdapat 25.593 kasus pencurian

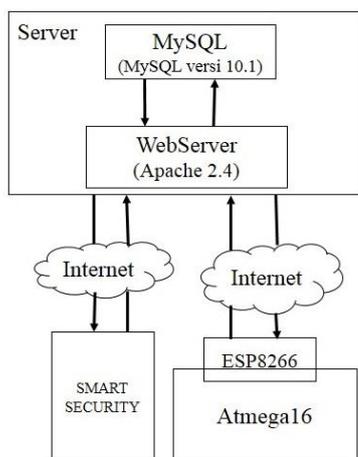
di Indonesia [1]. Diantara berbagai jenis pencurian, pencurian barang berharga, pencurian kendaraan bermotor, dan pencurian dengan kekerasan merupakan jenis pencurian yang mengalami peningkatan. Berdasarkan jumlah kasus pencurian yang semakin meningkat dan kasus pencurian yang semakin meningkat, diperlukan upaya peningkatan keamanan untuk melindungi hak milik.

Penelitian ini bertujuan untuk membuat perancangan aplikasi Smart Home berbasis Android untuk keamanan pintu rumah sebagai Penelitian, sehingga pemilik rumah dapat memantau keadaan rumahnya di manapun dan kapanpun. Penelitian ini diharapkan menghasilkan aplikasi Smart Home yang dapat memantau keadaan pintu rumah dan mengendalikan kunci pintu rumah sebagai usaha dalam menambah keamanan rumah.

2. Metode

2.1. Perancangan Sistem

Perancangan sistem ini membutuhkan alat dengan mikrokontroler atmega16 dan server dengan dua fungsi yaitu web server dan MySQL sebagai basis data.



Gambar 1. Perancangan sistem secara umum

Aplikasi Smart Security yang berbasis android harus terhubung internet untuk dapat mengendalikan kunci pintu. Segala transaksi dari aplikasi akan masuk ke *web server* yang kemudian diteruskan ke MySQL sebagai basis data. Segala perubahan kondisi pintu yang masuk ke basis data akan ditampilkan pada menu utama pada aplikasi sedangkan pada sisi alat setelah masuk ke basis data maka segala perubahan akan diteruskan hingga ke mikrokontroler yang pada akhirnya menggerakkan alat apakah itu menutup pintu, mengunci maupun membuka kunci.

2.2. Analisis Kebutuhan

Analisis kebutuhan sistem ini ditujukan untuk menguraikan kebutuhan-kebutuhan yang harus disediakan

oleh sistem agar dapat memenuhi kebutuhan pengguna dan sesuai dengan tujuan penelitian yaitu melakukan perancangan perangkat lunak berbasis Android pada sistem smart home sehingga penguncian pintu rumah dapat diatur secara *realtime* dan *mobile* melalui perangkat berbasis Android. Rancangan sistem ini menjelaskan kebutuhan antarmuka, kebutuhan data masukan dan data keluaran yang menunjukkan spesifikasi sistem yang dapat diakses.

2.2.1. Kebutuhan Fungsional

Kebutuhan fungsional merupakan gambaran mengenai fungsi-fungsi yang dapat dilakukan oleh sistem ini. Kebutuhan fungsional sistem meliputi:

- Sistem yang dirancang memberikan hak akses kepada pengguna terdaftar.
- Adanya fasilitas bagi *Admin* untuk menambahkan pengguna serta menentukan *level* dari pengguna dan melihat data terekam pada basis data.
- Adanya fasilitas untuk mengubah status pada basis data, yang pada nantinya akan mengubah kondisi pada mikrokontroler.
- Adanya fasilitas untuk melihat status pintu secara *real time*.

2.2.2. Kebutuhan Non Fungsional

Kebutuhan non-fungsional adalah kebutuhan sistem meliputi kinerja, kelengkapan operasi pada fungsi-fungsi yang ada, serta kesesuaian dengan lingkungan penggunaannya. Kebutuhan non-fungsional ini melingkupi beberapa kebutuhan yang mendukung kebutuhan fungsional, rumusan kebutuhan non-fungsional meliputi:

- Kebutuhan Operasional
 - Aplikasi dapat terinstal pada perangkat Android.
 - Data aman tersimpan di database dan hanya dapat dilihat oleh pengguna yang memiliki hak.
 - Yang dapat mengendalikan kunci pintu rumah hanya pengguna yang memiliki hak.
 - Sistem hanya dapat diakses melalui *file* format .apk yang telah terinstal di perangkat Android.
 - User interface pada aplikasi dibuat dengan sederhana untuk memudahkan pengguna.
 - Aplikasi dapat dijalankan pada minimal Android versi 4.4 (*kitkat*).
- Performansi Sistem

Sistem yang dibangun merupakan aplikasi yang berjalan pada lingkungan perangkat bergerak. Terdapat beberapa keterbatasan yang ditemui pada perangkat bergerak meskipun sistem operasi yang digunakan adalah Android. Oleh karena itu perlu diperhatikan guna menjadi acuan dalam pengembangan sistem, diantaranya:

 - Sumber daya baterai yang secara efektif hanya mampu bertahan selama kurang lebih 6 jam dengan

penggunaan secara terus menerus dan kurang lebih 200 jam dalam keadaan *standby*.

- Tampilan aplikasi antarmuka yang berpengaruh terhadap waktu tunggu sehingga aplikasi benar-benar siap digunakan, semakin banyak komponen yang digunakan akan semakin lama pula waktu tunggu yang dibutuhkan.

Dari keterbatasan pada perangkat Android tersebut, maka diusulkan beberapa alternatif untuk menunjang performa aplikasi dengan keterbatasan yang ada, diantaranya:

- Merancang aplikasi dengan pemanfaatan sumber daya seefisien mungkin namun tidak mengurangi fungsi dan performa aplikasi.
- Merancang aplikasi dengan antarmuka yang sederhana namun tetap menarik dan mudah digunakan oleh pengguna.

2.3. UML

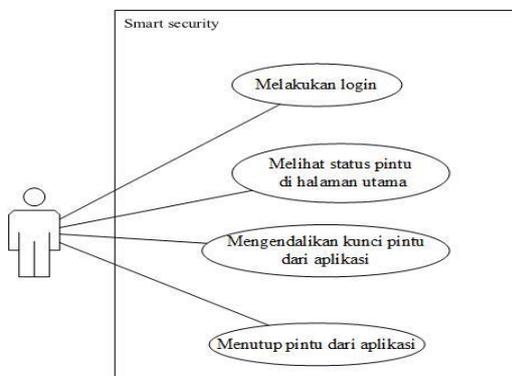
UML (*Unified Modelling Language*) adalah himpunan struktur dan teknik untuk pemodelan desain program berorientasi objek (OOP) serta aplikasinya. UML adalah metodologi untuk mengembangkan sistem OOP dan sekelompok perangkat untuk mendukung pengembangan sistem tersebut. UML mulai diperkenalkan oleh Object Management Group, sebuah organisasi yang telah mengembangkan model, teknologi, dan standar OOP sejak tahun 1980-an. Sekarang UML sudah mulai banyak digunakan oleh para praktisi OOP. UML merupakan dasar bagi perangkat (*tool*) desain berorientasi objek dari IBM[10].

2.3.1 Diagram Use Case

Diagram *use case* menggambarkan fungsi-fungsi yang ada pada sistem. Diagram ini lebih berfokus pada fitur-fitur sistem dari sudut pandang pihak luar, yang dalam hal ini pengguna digolongkan menjadi dua yaitu:

1. Member

Berikut merupakan diagram *Use Case member*.

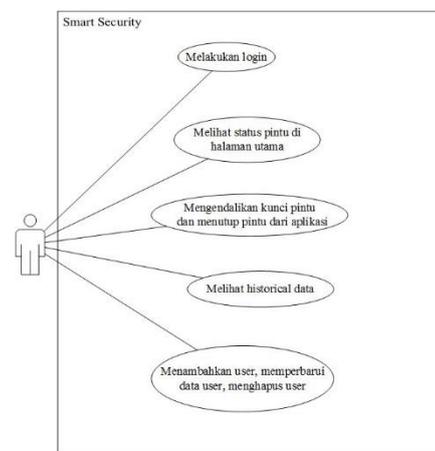


Gambar 2. Diagram Use Case Member

Pada gambar di atas menggambarkan fitur-fitur yang dapat diakses oleh pengguna dengan *level member* pada aplikasi Smart Security. Pengguna sebelumnya akan didaftarkan oleh *admin* dengan *level member*, setelah didaftarkan pengguna dapat langsung masuk ke menu *login*. Setelah berhasil *login* pengguna akan masuk ke halaman utama dan melihat status pintu terbaru, selanjutnya pengguna dapat mengendalikan kunci pintu dan menutup pintu dari aplikasi.

2. Admin

Berikut merupakan diagram *Use Case admin*.



Gambar 3. Diagram Use Case Admin

Pada gambar di atas menggambarkan fitur-fitur yang dapat diakses oleh pengguna dengan *level admin*. Pengguna akan langsung masuk ke menu *login*. Setelah berhasil *login* pengguna akan melihat status pintu di halaman utama, selanjutnya pengguna dapat mengendalikan kunci pintu dan menutup pintu dari aplikasi, pengguna dengan *level admin* juga dapat melihat Historical Data untuk melihat data terekam dari perubahan status pintu dan melakukan pengaturan pengguna. *Admin* dapat memperbaiki data dari pengguna yang terdaftar, menambahkan pengguna serta menentukan *level* pengguna, dan *admin* juga dapat menghapus pengguna.

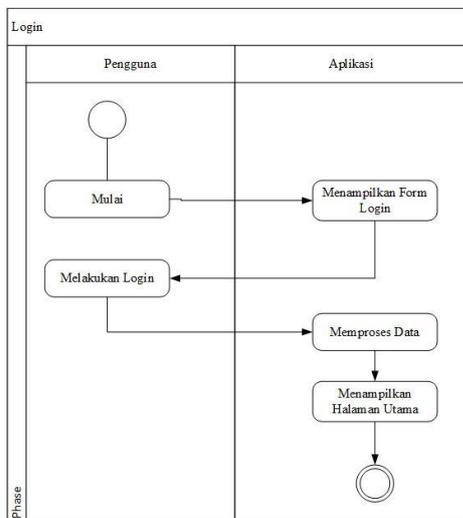
2.3.2 Diagram Aktivitas

Diagram aktivitas menggambarkan logika prosedural, dan aliran kerja dalam sistem yang sedang dirancang. Diagram aktivitas juga dapat digunakan untuk menggambarkan proses paralel yang mungkin terjadi pada beberapa eksekusi. Diagram aktivitas dapat dibagi menjadi beberapa untuk menggambarkan objek mana yang bertanggung jawab terhadap aktivitas tertentu.

1. Diagram Aktivitas Form Login

Diagram aktivitas ini menggambarkan aktivitas pengguna saat pertama kali memulai aplikasi. Pengguna akan melihat tampilan *form login* dimana pengguna akan memasukkan

alamat *email* dan kata sandi yang terdaftar. Diagram aktivitas *form login* dapat dilihat pada gambar berikut.

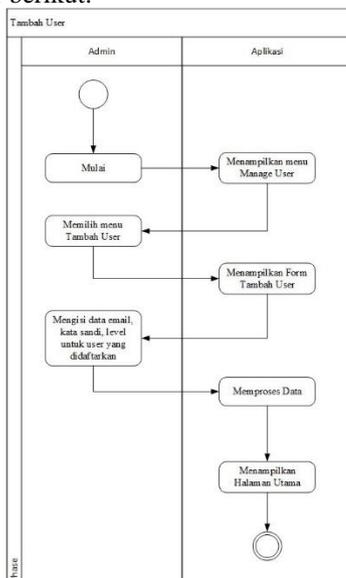


Gambar 4. Diagram Aktivitas *Login*

Pada gambar di atas merupakan gambar diagram aktivitas *login* untuk pengguna yang telah terdaftar di aplikasi. Pengguna akan melihat *form login* ketika memulai aplikasi.

2. Diagram Aktivitas Tambah User

Diagram aktivitas ini menggambarkan aktivitas pengguna pada *level admin* saat melakukan pendaftaran untuk pengguna. Diagram aktivitas pendaftaran dapat dilihat pada gambar berikut.

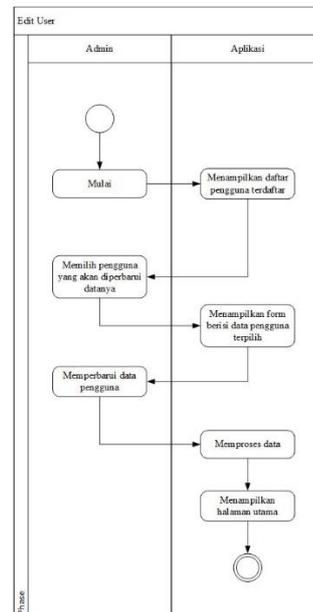


Gambar 5. Diagram Aktivitas Tambah *User*

Pada gambar di atas menggambarkan alur diagram aktivitas Tambah User pada pengguna dengan *level admin*. *Admin* akan masuk terlebih dahulu ke dalam menu Manage User untuk dapat menambahkan pengguna.

3. Diagram Aktivitas Mengubah Data Pengguna

Diagram aktivitas ini menggambarkan aktivitas pengguna pada *level admin* saat melakukan perubahan data pengguna.

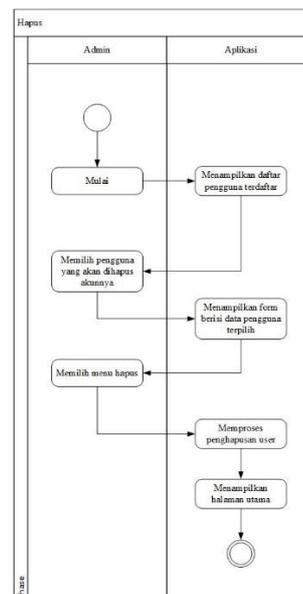


Gambar 6. Diagram Aktivitas Mengubah Data Pengguna

Pada gambar di atas menggambarkan aktivitas mengubah data pengguna pada *level admin*. *Admin* akan melihat seluruh pengguna terdaftar ketika memilih menu Manage User. *Admin* memilih pengguna yang akan dilakukan pembaruan data.

4. Diagram Aktivitas Hapus

Diagram aktivitas ini menggambarkan aktivitas pengguna pada *level admin* saat melakukan penghapusan akun pengguna.

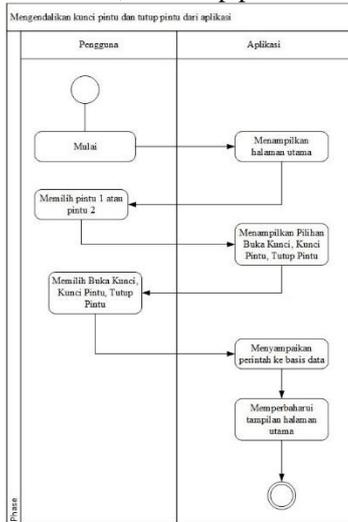


Gambar 7. Diagram Aktivitas Hapus

Pada gambar di atas adalah penggambaran dari aktivitas hapus yang di lakukan oleh pengguna dengan *level admin*. *Admin* akan melihat seluruh pengguna terdaftar ketika memilih menu Manage User. *Admin* akan memilih pengguna yang akan dihapus.

5. Diagram Aktivitas Mengendalikan Pintu

Diagram aktivitas ini menggambarkan aktivitas pengguna saat melakukan pengendalian kunci pintu, buka kunci, dan tutup pintu dari aplikasi. Diagram aktivitas mengendalikan kunci pintu, buka kunci, dan tutup pintu dari aplikasi.

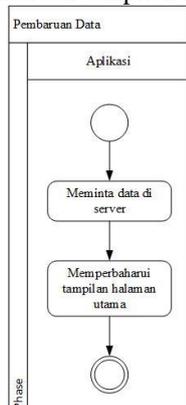


Gambar 8. Diagram Aktivitas Mengendalikan Pintu

Pada gambar di atas menggambarkan diagram aktivitas pengendalian kunci pintu, buka kunci dan tutup pintu. Ketika pengguna berhasil melakukan *login*, sistem akan menampilkan halaman utama dan status kondisi terbaru dari pintu satu dan pintu dua. Pengguna dapat memilih pintu satu atau pintu dua, setelah memilih pintu sistem akan menampilkan pilihan buka kunci, kunci pintu, tutup pintu dan pengguna dapat memilih dari pilihan tersebut. Sistem kemudian akan menyampaikan perintah ke basis data dan memperbarui tampilan di halaman utama.

6. Diagram Aktivitas Pembaruan Data

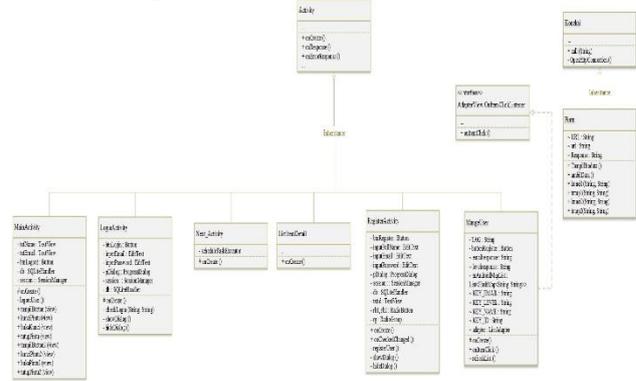
Diagram aktivitas ini menggambarkan aktivitas aplikasi saat melakukan pembaruan tampilan halaman utama.



Gambar 9. Diagram Aktivitas Pembaruan Data

Pada gambar di atas menggambarkan diagram aktivitas pembaruan data. Sistem akan meminta status kondisi terbaru di basis data dan memperbarui tampilan halaman utama. Sistem akan melakukan pembaruan data dengan interval waktu satu detik.

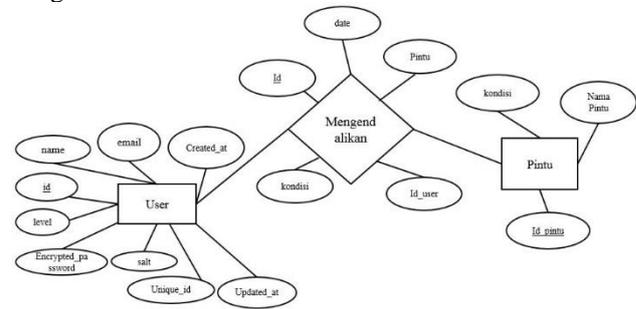
2.3.3 Diagram Class



Gambar 10. Diagram Class

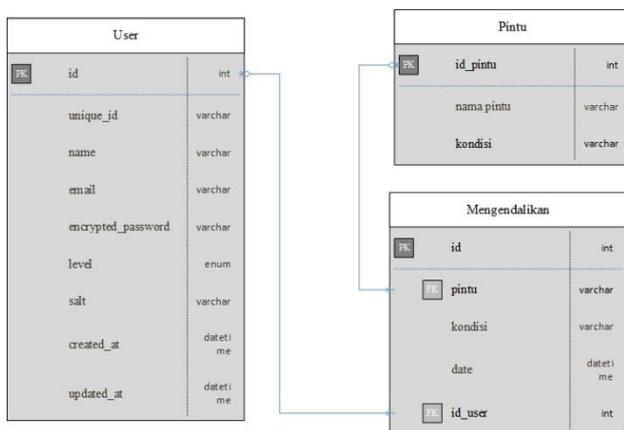
2.3.4. Desain Basis Data

Desain basis data dalam hal ini adalah *Entity Relationship Diagram (ERD)* berfungsi untuk memodelkan struktur basis data yang digunakan. Berguna untuk mengetahui kebutuhan yang diperlukan untuk membuat sistem berjalan dengan sesuai.



Gambar 11. Desain ERD

Pada gambar di atas merupakan desain dari ERD. Dalam aplikasi ini membutuhkan tiga entitas yaitu User, Mengendalikan dan Pintu. Entitas User terdapat beberapa atribut yaitu id sebagai *primary key*, unique id, name, email, encrypted password, level, salt, created at, updated at. Entitas Pintu terdapat atribut yaitu atribut id_pintu sebagai *primary key*, nama pintu, dan kondisi. Entitas Mengendalikan terdapat atribut yaitu atribut id sebagai *primary key*, date, pintu, kondisi, id_user. Hubungan antara ketiga entitas yaitu atribut id pada entitas User sebagai *primary key* dan atribut id_pintu sebagai *primary key*, dan kondisi pada entitas Pintu akan ada di dalam entitas Mengendalikan.



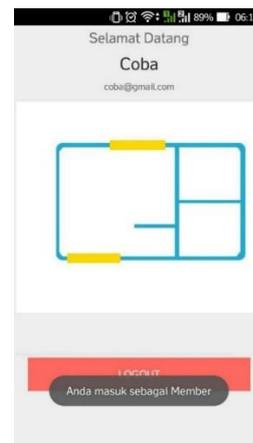
Gambar 12. Relasi Tabel Basis Data

Pada gambar di atas merupakan rancangan dari basis data. Dalam aplikasi ini dibutuhkan tiga tabel yaitu tabel User, tabel Mengendalikan, dan tabel Pintu. Tabel User berisi kumpulan data pengguna yang dapat mengakses aplikasi ini. Tabel Mengendalikan terdiri dari beberapa atribut yaitu id sebagai *primary key*, pintu, kondisi, date, dan id_user. Tabel Pintu berisi beberapa atribut yaitu id_pintu sebagai *primary key*, nama pintu, dan kondisi. Hubungan antara ketiga tabel yaitu atribut id pada tabel User sebagai *primary key* menjadi *foreign key* dan pada tabel Mengendalikan yaitu pada atribut id_user. Atribut id_pintu sebagai *primary key* pada tabel Pintu menjadi *foreign key* pada tabel Mengendalikan yaitu pada atribut pintu.

3. Hasil dan Analisa

3.1. Pengujian Fungsi Login

Pada penelitian ini, pengujian dilakukan dengan melakukan pengujian black box yaitu metode pengujian perangkat lunak yang menguji fungsionalitas aplikasi yang bertentangan dengan struktur internal atau kerja pengujian fungsionalitas dari aplikasi. Pengujian black box bertujuan untuk identifikasi dan menghilangkan masalah sebelum akhirnya sampai ke pengguna yang sebenarnya. Pengujian pada aplikasi ini terdiri dari beberapa variabel hasil implementasi dari aplikasi, dimana pengujian ini dilakukan dalam lima kali. Pengujian yang dilakukan pada tahap ini meliputi pengujian fungsi *login*, pendaftaran akun, memberikan perintah pada pintu rumah, mengambil data status pintu rumah, dan pengambilan data Historical Data.



Gambar 13. Tampilan program saat berhasil login

Berikut ini adalah hasil pengujian fungsi login

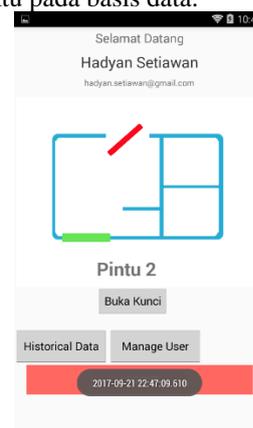
Tabel 1. Hasil pengujian fungsi login

Perintah	Peran	Keterangan
Login	Admin	Berhasil
Login	Member	Berhasil
Login	Admin	Berhasil
Login	Member	Berhasil
Login	Member	Berhasil

Tabel 1 menunjukkan hasil dari fungsi login. Aplikasi dilakukan percobaan login sebanyak 5 kali dengan berbagai varian tingkatan level akun.

3.2. Pengujian Fungsi Buka Kunci

Pada bagian ini adalah pengujian dari fungsi buka kunci pada pintu satu dan pintu dua dan mencari selisih waktu antara waktu pada *smarthphone* saat tombol buka kunci ditekan dan waktu pada basis data.



Gambar 14. Tampilan waktu saat menekan buka kunci

ID	Date	Pintu1	Pintu2	Status	id_user	millisecond
12695	2017-09-21 22:48:10	Terbuka	Tertutup		3	0.48112600
12694	2017-09-21 22:48:02	Terbuka	Terkunci		3	0.91531400
12693	2017-09-21 22:47:56	Terbuka	Tertutup		3	0.29303700
12692	2017-09-21 22:47:48	Terbuka	Terkunci		3	0.91407000
12691	2017-09-21 22:47:16	Terbuka	Tertutup		3	0.04830500
12690	2017-09-21 22:47:09	Terbuka	Terkunci		3	0.69705700
12689	2017-09-21 22:47:01	Terbuka	Tertutup		3	0.31192600
12688	2017-09-21 22:46:51	Terbuka	Terkunci		3	0.82581100
12687	2017-09-21 22:46:30	Terbuka	Tertutup		3	0.24231300
12686	2017-09-21 22:45:34	Terbuka	Terkunci		3	0.57942500
12685	2017-09-21 22:45:09	Terbuka	Tertutup		3	0.63651600
12684	2017-09-21 22:43:58	Terbuka	Terkunci		0	0.56972200
12683	2017-09-21 22:42:35	Terbuka	Terbuka		0	0.64028400

Gambar 15 Tampilan waktu di basis data

Pada gambar 14 adalah tampilan saat menekan tombol buka kunci, terlihat pula bahwa ditekan pada 09.610 milidetik, sedangkan pada gambar 15 terlihat data yang masuk ke basis data yaitu 0,697057 milidetik pada detik ke 9, hal ini menunjukkan terdapat perbedaan waktu antara di *smartphone* dengan di basis data, maka untuk mendapatkan selisih waktunya, waktu pada basis data dikurangi dengan waktu yang tertera pada *smartphone*. Agar dapat dilakukan perhitungan angka pada detik ditempelkan pada angka yang ada pada kolom milidetik dan mengambil tiga angka di belakang koma, contohnya pada kolom milidetik terlihat angkanya 0,697057 milidetik, pada detiknya ke 9, maka menjadi 09.610, sehingga operasi matematikanya 09.697 dikurangi 09.610 dengan hasil sebesar 87 milidetik.

Tabel 2. Pengujian buka kunci pintu pada pintu 1 dan pintu 2

Pintu	Perintah	Aplikasi (ms)	Basisdata (ms)	Selisih (ms)	Ket
1	BukaKunci	33.242	33.343	101	Berhasil
1	BukaKunci	8.441	8.541	100	Berhasil
1	BukaKunci	25.176	25.277	101	Berhasil
1	BukaKunci	51.155	51.256	101	Berhasil
1	BukaKunci	5.707	5.808	101	Berhasil
2	BukaKunci	30.156	30.242	86	Berhasil
2	BukaKunci	1.225	1.311	86	Berhasil
2	BukaKunci	15.961	16.048	87	Berhasil
2	BukaKunci	56.206	56.293	87	Berhasil
2	BukaKunci	10.394	10.481	87	Berhasil

3.3. Pengujian Fungsi Kunci Pintu

Pada bagian dilakukan pengujian untuk fungsi kunci pintu pada pintu satu dan pintu dua serta mencari selisih waktu antara waktu pada *smartphone* saat tombol kunci pintu ditekan dengan waktu pada basis data.



Gambar 16 Tampilan waktu saat kunci pintu

ID	Date	Pintu1	Pintu2	Status	id_user	millisecond
12695	2017-09-21 22:48:10	Terbuka	Tertutup		3	0.48112600
12694	2017-09-21 22:48:02	Terbuka	Terkunci		3	0.91531400
12693	2017-09-21 22:47:56	Terbuka	Tertutup		3	0.29303700
12692	2017-09-21 22:47:48	Terbuka	Terkunci		3	0.91407000
12691	2017-09-21 22:47:16	Terbuka	Tertutup		3	0.04830500
12690	2017-09-21 22:47:09	Terbuka	Terkunci		3	0.69705700
12689	2017-09-21 22:47:01	Terbuka	Tertutup		3	0.31192600
12688	2017-09-21 22:46:51	Terbuka	Terkunci		3	0.82581100
12687	2017-09-21 22:46:30	Terbuka	Tertutup		3	0.24231300
12686	2017-09-21 22:45:34	Terbuka	Terkunci		3	0.57942500
12685	2017-09-21 22:45:09	Terbuka	Tertutup		3	0.63651600
12684	2017-09-21 22:43:58	Terbuka	Terkunci		0	0.56972200
12683	2017-09-21 22:42:35	Terbuka	Terbuka		0	0.64028400

Gambar 17 Tampilan waktu di basis data

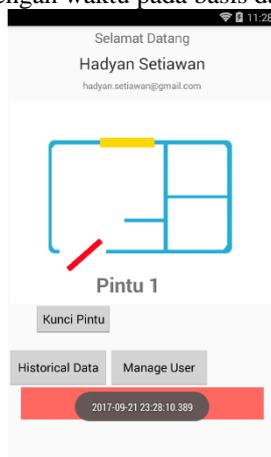
Pada gambar 16 adalah tampilan saat menekan tombol kunci pintu, terlihat pula bahwa ditekan pada 34.493 milidetik, sedangkan pada Gambar 17 terlihat data yang masuk ke basis data yaitu 0,579425 milidetik pada detik ke 34, hal ini menunjukkan terdapat perbedaan waktu antara di *smartphone* dengan di basis data, maka untuk mendapatkan selisih waktunya, waktu pada basis data dikurangi dengan waktu yang tertera pada *smartphone*. Agar dapat dilakukan perhitungan angka pada detik ditempelkan pada angka yang ada pada kolom milidetik dan mengambil tiga angka di belakang koma, contohnya pada kolom milidetik terlihat angkanya 0,579425 milidetik, pada detiknya ke 34, maka menjadi 34.579, sehingga operasi matematikanya 34. dikurangi 34.493 dengan hasil sebesar 86 milidetik.

Tabel 3. Pengujian kunci pintu pada pintu 1 dan pintu 2

Pintu	Perintah	Aplikasi (ms)	Basis data (ms)	Selisih (ms)	Ket
1	Kunci Pintu	3.422	3.523	101	Berhasil
1	Kunci Pintu	1.591	1.692	101	Berhasil
1	Kunci Pintu	16.178	16.277	99	Berhasil
1	Kunci Pintu	36.107	36.208	101	Berhasil
1	Kunci Pintu	58.625	58.725	100	Berhasil
2	Kunci Pintu	34.493	34.579	86	Berhasil
2	Kunci Pintu	51.739	51.825	86	Berhasil
2	Kunci Pintu	9.610	9.697	87	Berhasil
2	Kunci Pintu	48.831	48.914	83	Berhasil
2	Kunci Pintu	2.828	2.915	87	Berhasil

3.4. Pengujian Fungsi Tutup Pintu

Pada bagian ini dilakukan pengujian untuk fungsi tutup pintu pada pintu satu dan pintu dua serta mencari selisih waktu antara waktu pada *smarthphone* saat tombol tutup pintu ditekan dengan waktu pada basis data



Gambar 18. Tampilan waktu saat tutup Pintu

ID	Date	Pintu1	Pintu2	Status	id_user	millisecond
12722	2017-09-21 23:29:44	Tertutup	Tertutup		3	0.63850200
12721	2017-09-21 23:29:34	Tertutup	Terbuka		3	0.06798400
12720	2017-09-21 23:29:20	Terbuka	Terbuka		0	0.38046000
12719	2017-09-21 23:29:12	Tertutup	Tertutup		3	0.05253200
12718	2017-09-21 23:29:02	Tertutup	Terbuka		3	0.33603600
12717	2017-09-21 23:28:56	Terbuka	Terbuka		0	0.19883400
12716	2017-09-21 23:28:48	Tertutup	Tertutup		3	0.05150000
12715	2017-09-21 23:28:39	Tertutup	Terbuka		3	0.56220500
12714	2017-09-21 23:28:33	Terbuka	Terbuka		0	0.45516300
12713	2017-09-21 23:28:24	Tertutup	Tertutup		3	0.04967500
12712	2017-09-21 23:28:10	Tertutup	Terbuka		3	0.49880200
12711	2017-09-21 23:28:02	Terbuka	Terbuka		0	0.63950300
12710	2017-09-21 23:27:31	Tertutup	Tertutup		3	0.31818900
12709	2017-09-21 23:27:10	Tertutup	Terbuka		3	0.01314200
12708	2017-09-21 23:26:40	Terbuka	Terbuka		0	0.73982400

Gambar 19. Tampilan waktu di basis data

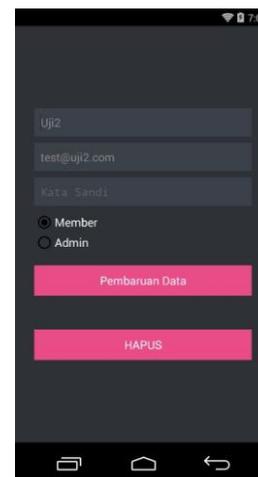
Pada gambar 18 adalah tampilan saat menekan tombol kunci pintu, terlihat pula bahwa ditekan pada 10.389 milidetik, sedangkan pada gambar 19 terlihat data yang masuk ke basis data yaitu 0,498802 milidetik pada detik ke 10, hal ini menunjukkan terdapat perbedaan waktu antara di *smartphone* dengan di basis data, maka untuk mendapatkan selisih waktunya, waktu pada basis data dikurangi dengan waktu yang tertera pada *smartphone*. Agar dapat dilakukan perhitungan angka pada detik ditempelkan pada angka yang ada pada kolom milidetik dan mengambil tiga angka di belakang koma, contohnya pada kolom milidetik terlihat angkanya 0,498802 milidetik, pada detiknya ke 10, maka menjadi 10.498, sehingga operasi matematikanya 10.498 dikurangi 10.389 dengan hasil sebesar 109 milidetik.

Tabel 4. Pengujian tutup pintu pada pintu 1 dan pintu 2

Pintu	Perintah	Aplikasi (ms)	Basis Data (ms)	waktu (ms)	Ket
1	Tutup Pintu	9.904	10.013	109	Berhasil
1	Tutup Pintu	10.389	10.498	109	Berhasil
1	Tutup Pintu	39.457	39.562	105	Berhasil
1	Tutup Pintu	2.226	2.336	110	Berhasil
1	Tutup Pintu	33.957	34.067	110	Berhasil
2	Tutup Pintu	31.209	31.318	109	Berhasil
2	Tutup Pintu	23.939	24.049	110	Berhasil
2	Tutup Pintu	47.949	48.051	102	Berhasil
2	Tutup Pintu	11.942	12.052	110	Berhasil
2	Tutup Pintu	24.528	24.638	110	Berhasil

3.5. Pengujian Fungsi Ubah Data Pengguna

Pada bagian ini adalah pengujian fungsi ubah data pengguna, pengujian dilakukan untuk mengetahui fungsi ubah data pengguna bekerja dengan sesuai dengan yang di harapkan.



Gambar 20. Tampilan form perubahan

Pada gambar 20 di atas adalah tampilan berupa *form* berisi data dari pengguna dalam hal ini adalah data dari pengguna Uji2, pada tampilan ini juga *admin* dapat mengubah data pengguna dengan *email* test@uji2.com dan nama Uji2.

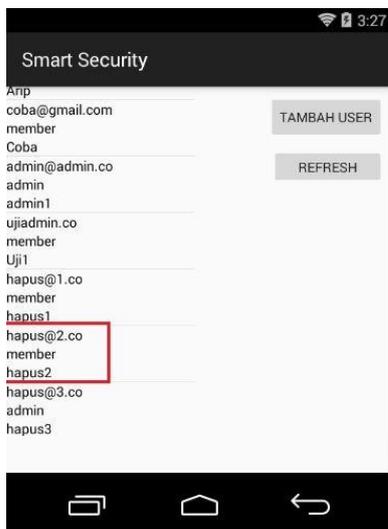
Tabel 5. Pengujian fungsi ubah data pengguna

No Pengujian	Edit email	Edi nama	Edit kata sandi	Edit level	Ket
1	v	-	-	-	Berhasil
2	-	v	-	-	Berhasil
3	-	-	v	-	Berhasil
4	-	-	-	v	Berhasil
5	v	v	-	-	Berhasil
6	-	-	v	v	Berhasil
7	-	v	v	-	Berhasil
8	v	-	-	v	Berhasil
9	v	-	v	v	Berhasil
10	v	v	-	v	Berhasil
11	-	v	v	v	Berhasil
12	v	v	v	v	Berhasil

Pada Tabel 5 terlihat pada hasil pengujian fungsi ubah data pengguna dengan berbagai variasi pengujian hasilnya adalah berhasil. Contoh variasi yang dilakukan adalah pengubahan data hanya pada *email* yang dirubah. Jika *email* tersebut belum terdaftar maka hasilnya berhasil.

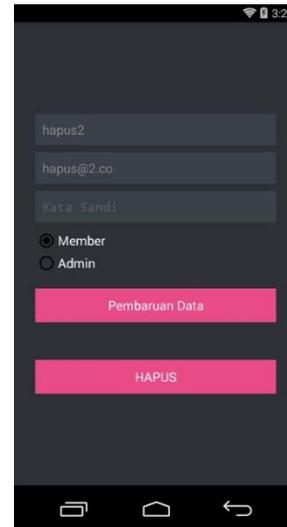
3.6. Pengujian fungsi hapus pengguna

Pada bagian ini adalah pengujian fungsi hapus pengguna, pengujian dilakukan untuk mengetahui fungsi hapus pengguna bekerja sesuai dengan yang diharapkan.



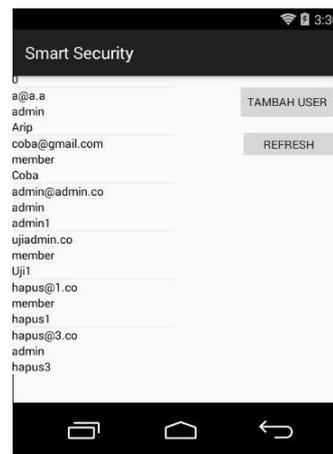
Gambar 21. Tampilan manage user

Pada gambar 21 adalah tampilan manage user, dimana pengguna dengan *level admin* akan melihat daftar pengguna terdaftar. Untuk menghapus pengguna, *admin* dapat memilih pengguna yang akan dihapus.



Gambar 22. Tampilan form menghapus pengguna

Pada Gambar 22 adalah tampilan *form* setelah pengguna dengan nama hapus2 dipilih oleh *admin*. Ketika *admin* memilih tombol hapus, maka pengguna tersebut akan terhapus dari daftar pengguna.

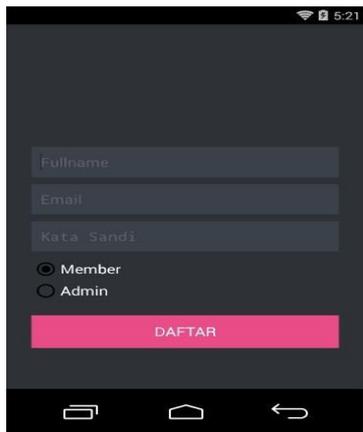


Gambar 23. Tampilan setelah menghapus pengguna

Pada gambar 23 adalah tampilan setelah pengguna dengan nama hapus2 berhasil di hapus. Terlihat pada gambar diatas bahwa pengguna dengan nama hapus2 dan *email* hapus@2.co tidak lagi terdaftar.

3.7. Pengujian Tambah User

Pada bagian ini adalah pengujian fungsi pendaftaran, pengujian dilakukan untuk mengetahui fungsi pendaftaran bekerja sesuai dengan yang diharapkan.



Gambar 24. Tampilan fungsi tambah user

Pada Gambar 24 adalah tampilan dari form pendaftaran. Pengguna dengan level admin akan mendaftarkan pengguna dengan mengisi kolom-kolom yang ada, seperti kolom nama, email, kata sandi dan level.

Tabel 6. Pengujian fungsi tambah user

Perintah	Peran	Keterangan
Pendaftaran	Member	Berhasil
Pendaftaran	Member	Berhasil
Pendaftaran	Admin	Berhasil
Pendaftaran	Member	Berhasil
Pendaftaran	Admin	Berhasil

Pada Tabel 6 terlihat hasil dari fungsi pendaftaran setelah dilakukan lima percobaan dan hasilnya berhasil.

3.8. Pengujian fungsi historical data

Pada bagian ini adalah pengujian melihat historical data oleh pengguna dengan level admin, tujuannya untuk memastikan bahwa basis data dan android terhubung.



Gambar 25. Tampilan historical data

ID	Date	Pintu1	Pintu2	Status	id_user	millisecond
12724	2017-09-22 16:54:14	Terbuka	Terbuka		0	0.14905400
12723	2017-09-22 13:36:46				0	0.34114200
12722	2017-09-21 23:29:44	Tertutup	Tertutup		3	0.63850200
12721	2017-09-21 23:29:34	Tertutup	Terbuka		3	0.06798400
12720	2017-09-21 23:29:20	Terbuka	Terbuka		0	0.38046000
12719	2017-09-21 23:29:12	Tertutup	Tertutup		3	0.05253200
12718	2017-09-21 23:29:02	Tertutup	Terbuka		3	0.33603600
12717	2017-09-21 23:28:56	Terbuka	Terbuka		0	0.19883400
12716	2017-09-21 23:28:48	Tertutup	Tertutup		3	0.05150000
12715	2017-09-21 23:28:39	Tertutup	Terbuka		3	0.56220500
12714	2017-09-21 23:28:33	Terbuka	Terbuka		0	0.45516300
12713	2017-09-21 23:28:24	Tertutup	Tertutup		3	0.04967500
12712	2017-09-21 23:28:10	Tertutup	Terbuka		3	0.49880200
12711	2017-09-21 23:28:02	Terbuka	Terbuka		0	0.63950300
12710	2017-09-21 23:27:31	Tertutup	Tertutup		3	0.31818900
12709	2017-09-21 23:27:10	Tertutup	Terbuka		3	0.01314200
12708	2017-09-21 23:26:40	Terbuka	Terbuka		0	0.73982400
12707	2017-09-21 23:26:08	Terbuka	Terbuka		0	0.80077500
12706	2017-09-21 23:14:05	Tertutup	Tertutup		3	0.80040100
12705	2017-09-21 23:13:58	Terluncur	Tertutup		3	0.72584200

Gambar 26. Tampilan pada basis data

Pada gambar 25 adalah tampilan menu historical data yang hanya terdapat pada akun admin, sedangkan gambar 26 adalah tampilan pada basis data di server. Dapat terlihat bahwa data dari keduanya sama, ini menandakan bahwa menu historical data berhasil mengambil data dari server basis data, sehingga perubahan pada basis data bisa langsung ditampilkan pada menu historical data.

4. Kesimpulan

Aplikasi Smart Security dapat melakukan fungsi-sungsi dasar untuk mengendalikan keamanan pintu rumah, yaitu buka kunci, tutup pintu, dan kunci pintu. Untuk perubahan status menjadi buka kunci memakan waktu rata-rata 93,7 milidetik tiap pintunya dari lima kali percobaan. Untuk perubahan status menjadi kunci pintu memakan waktu rata-rata 93,1 milidetik tiap pintunya dari lima kali percobaan. Untuk perubahan status menjadi tutup pintu memakan waktu rata-rata 108.4 milidetik tiap pintunya dari lima kali percobaan. Hasil pengujian dengan metode black box pada fitur-fitur aplikasi menunjukkan seluruh pengujian berhasil.

Hasil pengujian fitur login menunjukkan bahwa hanya yang telah terdaftar yang dapat melakukan login dan masuk ke halaman utama, ini sesuai dengan yang diharapkan. Hasil pengujian fitur tambah pengguna menunjukkan admin dapat menambahkan pengguna hingga dapat menentukan level pengguna, sesuai dengan yang diharapkan. Hasil pengujian fitur menu historical data untuk admin menunjukkan bahwa kesesuaian data antara server basis data dengan tampilan data pada menu historical data. Hasil pengujian fitur mengubah data pengguna menunjukkan admin dapat merubah data pengguna hanya sebagai atau keseluruhan dari nama, email, kata sandi hingga dapat merubah level pengguna, ini sesuai dengan apa yang diharapkan. Hasil pengujian fitur hapus pengguna menunjukkan bahwa admin dapat menghapus pengguna terdaftar. Fitur ini berada di menu manage user yang hanya terdapat pada akun admin.

Aplikasi ini masih mungkin untuk dikembangkan menjadi lebih modular, sehingga nantinya bukan hanya pintu yang dapat dikendalikan namun juga peralatan-peralatan lain yang ada dirumah seperti penyejuk ruangan, lampu dan lainnya.

Referensi

- [1]. Badan pusat Statistik, "Statistik kriminal 2014," Jakarta, 2014
- [2]. Dharma Kasman, Akhmad. "Kolaborasi Dahsyat Android dengan PHP & MySQL,Cetakan I". Lokomedia, Yogyakarta, 2013
- [3]. S. P. Hall and E. Anderson, "Operating systems for mobile computing," J. Comput. Sci. Coll., vol. 25, pp. 64–71, 2009"
- [4]. B. Hohensee, "Android for Beginners - Developing apps using Android Studio". Sweden, 2014
- [5]. Yudhanto, yudha."Apa itu IOT (Internet Of Things)?," Indonesia, 2007
- [6]. Peranginangin , Kasiman. "Aplikasi Web dengan PHP dan MySQL," Jakarta : Andi Publisher, 2006.
- [7]. Kadir, Abdul. "Dasar Pemrograman Java 2" , Andi publisher. Yogyakarta, 2007
- [8]. rickyanto, isak. "Dasar Pemrograman Berorientasi Objek dengan Java 2", Andi Publisher. Yogyakarta, 2003
- [9]. H, Nazrudin Safaat. "ANDROID : Pemograman Aplikasi Mobile Smartphone dan Tablet PC Berbasis Android". Bandung, 2012
- [10]. W. Komputer, "The Best 40 Java Applications." Jakarta: PT. Elex Media Computindo, 2010.
- [11]. D. M. Kroenke, "Dasar-Dasar, Desain dan Implementasi Database Processing." Penerbit Erlangga, 2015
- [12]. Simarmata, Janner. "Rekayasa Perangkat Lunak." Yogyakarta: ANDI. 2010.
- [13]. Bayu Prakasa A, Syahrill M, Dedy Hermanto, Yudha Adi, Otomatisasi *Smart Home* dengan Raspberry Pi dan *Smartphone* Android, STMIK Global Informatika MDP. Palembang, 2013.
- [14]. Mulyani, Sri, "Metode Analisis dan Perancangan Sistem." Bandung: Abdi Sistematika, 2016
- [15]. Brady, M.,& Loonam, J., "Exploring the use of entity-relationship diagramming as a technique to support grounded theory inquiry", Bradford: Emerald Group Publishing, 2010