

SISTEM KENDALI PERANGKAT LISTRIK BERBASIS ANDROID

Moch. Ali Ramdhani
Mohammad Latief

ABSTRAK

Seiring dengan perkembangan teknologi dan kebutuhan manusia akan kemudahan di era globalisasi yang semakin meningkat tentunya setiap orang cenderung ingin mengerjakan segala sesuatu dengan mudah dan praktis. Penggunaan *remote control* untuk mengendalikan perangkat listrik tentu menjadi salah satu penerapan prinsip kerja yang praktis dan fleksibel. Selain kemudahan dalam media komunikasi efisiensi kendali sistem juga diharapkan dapat membantu dalam kehidupan. Penggunaan *remote control* menjadi salah satu pilihan untuk lebih memudahkan dalam mengendalikan lampu ruangan. Kolaborasi dari kemajuan komunikasi dan kecanggihan sistem kendali dapat mempermudah seorang pengguna untuk mengendalikan sebuah sistem dari jarak jauh, tidak hanya antar kota, bahkan antar pulau dan Negara dalam lingkup jaringan komunikasi yang digunakan.

Aplikasi kendali peralatan bersumber tegangan AC (*Alternating Current*) melalui layanan aplikasi Android dimana nantinya pengguna aplikasi tidak perlu harus mendatangi kotak saklar untuk dapat menyalakan peralatan tersebut, namun hanya dengan mengirimkan sebuah pesan atau mengontrolnya melalui aplikasi Android untuk mengaktifkan peralatan listrik yang diinginkan.

Sistem yang dibentuk ini digunakan pada aplikasi ini seperti halnya pencatat laju komunikasi pesan yang digunakan (*log* pesan) untuk dicocokkan tipe atau format pesan yang dikirim untuk mengaktifkan alat ataupun saat mengontrol melalui android dengan tombol *on/off* yang sudah tersedia di aplikasi tersebut.

Aplikasi ini dibangun menggunakan metode penelitian *Waterfall* sistem. Basis data aplikasi ini dibangun dengan menggunakan MySQL dan bahasa pemrograman yang digunakan *mobile programming* dengan bahasa java.

Kata Kunci : *Remote Control, Android, Waterfall*

1. PENDAHULUAN

Pengendali jarak jauh atau *remote control* adalah teknologi elektronik yang digunakan untuk mengoperasikan sebuah mesin dari jarak jauh. Istilah *remote control* juga sering disingkat menjadi “*remote*” saja. Pada umumnya, pengendali jarak jauh digunakan untuk memberikan perintah dari kejauhan kepada televisi atau barang-barang elektronik lainnya. Teknologi pengendali jarak jauh ini dapat diimplementasikan untuk mengoperasikan sebuah sistem pengendalian listrik lainnya seperti lampu.

Pada zaman yang modern ini, masyarakat mempunyai kebiasaan untuk fokus dalam pekerjaannya sendiri tanpa memikirkan keadaan sekelilingnya. Hal itu yang menyebabkan munculnya kesalahan-kesalahan sederhana namun berdampak besar

apabila kebiasaan itu dilakukan terus menerus. Contohnya adalah lupa untuk mematikan lampu yang menyebabkan lampu itu menyala seharian. Menyalakan dan mematikan (mengontrol) peralatan listrik lampu dengan mudah dan mampu melakukannya dimana saja akan sangat membantu masyarakat untuk mengatasi kebiasaan lupa untuk mematikan perangkat listrik lampu.

2. METODE PENELITIAN

Metode penelitian dalam pengembangan sistem yang akan diterapkan oleh adalah menggunakan metode *waterfall* yang sering disebut *classic life cycle* atau model *sekuensial linier*. Metode ini dipilih karena merupakan metode pengembangan sistem yang sistematis dan sekuensial. Adapun tahapan-tahapannya sebagai berikut :

a. Analisa kebutuhan perangkat lunak

Proses pengumpulan kebutuhan diintensifkan dan difokuskan, khususnya pada perangkat lunak untuk memahami sifat program yang dibangun. Analisis harus memahami domain informasi, tingkah laku dan antar muka yang diperlukan.

b. Desain

Merupakan proses yang berfokus pada 4 (empat) atribut sebuah program yang berbeda yaitu struktur data, arsitektur perangkat lunak, representasi antarmuka dan detail (algoritma) procedural. Proses desain menterjemahkan kebutuhan sistem menjadi sebuah representasi perangkat lunak sebelum dimulainya pengkodean. Kemudian hasil perancangan ini didokumentasikan sehingga menjadi bagian dari konfigurasi perangkat lunak.

c. Pengkodean

Merupakan proses menerjemahkan hasil perancangan ke dalam bentuk yang dapat dimengerti oleh mesin (komputer) dengan menggunakan bahasa pemrograman tertentu.

d. Testing

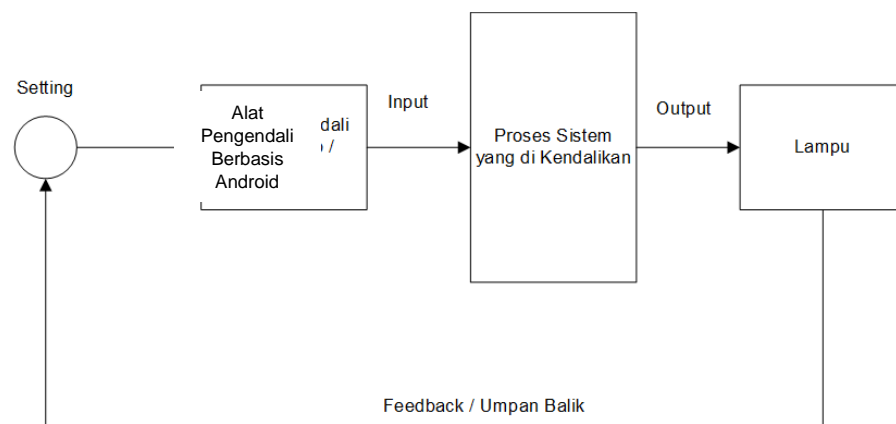
Setelah melakukan pengkodean, pengujian program dilakukan. Proses ini berfokus pada logika internal perangkat lunak untuk memastikan bahwa semua pernyataan sudah diuji dan pengujian eksternal fungsional yaitu mengarahkan

pengujian untuk menemukan kesalahan dan memastikan pengujian system berjalan dengan baik.

3. PEMBAHASAN

a. Analisis Dan Perancangan

Pada tahap identifikasi sistem ini dimulai dengan mengidentifikasi sistem yang akan dibangun. Sistem yang akan dibangun merupakan suatu sistem pengendalian pada peralatan listrik. Berdasarkan teori yang didapat, pada dasarnya sistem pengendalian terdiri dari tiga elemen pokok, yaitu *input*, *proses* dan *output*. *Output* merupakan hal yang dihasilkan oleh kendalian artinya yang dikendalikan. Sedangkan input adalah yang mempengaruhi kendalian, yang mengatur output. Dalam hal ini yang dikendalikan adalah peralatan listrik. Kebutuhan minimum pengendalian pada peralatan listrik yaitu umumnya menghidupkan dan mematikan peralatan listriknya seperti pada lampu, televisi, AC (Air Conditioner), dan sebagainya. Oleh sebab itu, pada penelitian ini dibangun suatu sistem pengendalian peralatan listrik yang disimulasikan pada lampu dalam melakukan pengendalian menghidupkan dan mematikan (*on/off*).



Gambar 1 Alur Kerja Sistem

Pada Gambar 1 dapat dilihat bahwa sistem pengendalian yang dibangun bersifat sistem loop tertutup (*close loop system*). Alat pengendali aplikasi Android, selain memberikan *input* berupa perintah hidup (*on*) dan mati (*off*) dalam pengendaliannya juga mendapatkan umpan balik (*feedback*) dari hasil keluaran (*output*) pada masukan (*input*) yang diberikan. Sehingga koreksi kesalahan dapat dilakukan untuk pengendaliannya sudah sesuai dengan masukan (*input*) atau tidak.

1. Relay *USB-2REL*

Relay ini adalah modul *hardware* yang digunakan sebagai pengekseskusi *ON/OFF* lampu dengan dengan konektor *USB (Universal Serial Bus)* yang disambung ke komputer.



Gambar 2 Modul *Relay USB-2REL*

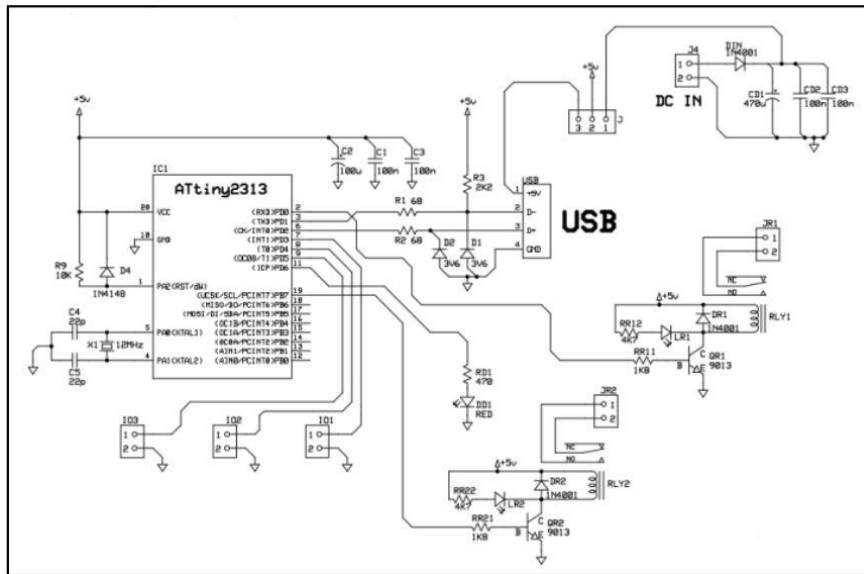
USB-2REL mengambil catudaya dari Port USB (tidak membutuhkan catudaya dari luar), oleh karenanya sangat praktis - tinggal colok dan jalankan program.

Berikut spesifikasi *USB-2REL* :

Tabel 1 Spesifikasi *USB-2REL*

Tegangan Kerja	: DC 5V (USB Powered - tidak perlu catudaya eksternal)
Jumlah Output	: 2 buah
Tipe Output	: Relay (Kontak NO=10A, Kontak NC=6A @250VAC)
Mikrokontroler	: ATtiny2313A
Driver Relay	: Transistor
LED Indikator	: Power On, Relay1 On, Relay2 On
Watchdog Timer	: 30ms
Sistem Operasi	: Windows, Linux, Raspberry Pi
Software Driver	: libusb
Command-Line	: usb2rc.exe, usb2rc

Interface

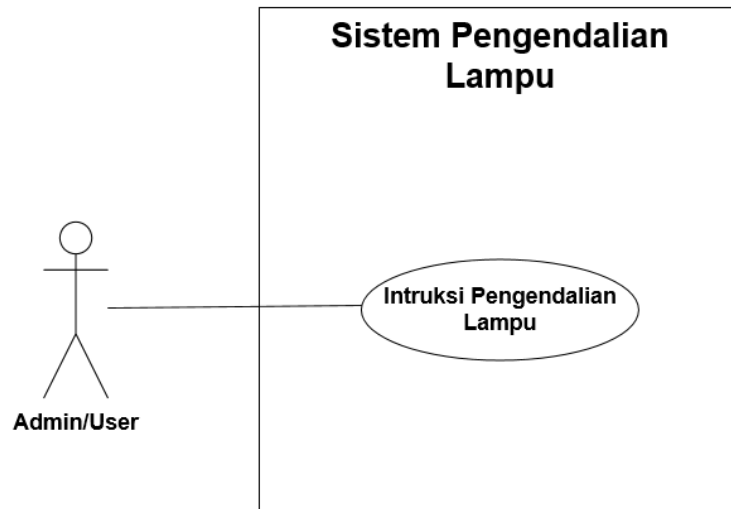


Gambar 1 Rangkaian *USB-2REL* Relay

sumber : <http://www.usbinov.com/usb-2rel/index.htm>

2. Use Case Diagram

Diagram *Use Case* merupakan bagian tertinggi dari fungsionalitas yang dimiliki sistem yang akan menggambarkan bagaimana seseorang atau aktor akan menggunakan dan memanfaatkan sistem. Sasaran pemodelan *use case* diantaranya adalah mendefinisikan kebutuhan fungsional dan operasional sistem dengan mendefinisikan skenario penggunaan sistem yang akan dibangun.



Gambar 4 Use Case Diagram yang di usulkan pada aplikasi Android

Berdasarkan use case diagram di atas, maka ditarik sebuah penjelasan bahwa terdapat dua aktor utama yaitu Admin dan user . Admin memiliki akses untuk mengolah akses dan user meliputi aksi menambah, mengedit atau menghapus akses ataupun user di sistem dengan melakukan akses melalui aplikasi *web* sedangkan user hanya digunakan dapat memberikan instruksi itupun dengan syarat user harus sudah terdaftar di sistem. Untuk aplikasi Android admin dan user sama sama memiliki hak yang sama yaitu hanya bisa untuk melakukan instruksi pengendalian lampu dengan syarat terdaftar di sistem. Berikut ini adalah deskripsi dari admin dan user dari *case* yang terdapat pada *use case diagram* di atas:

Tabel 2 Use Case olah akses

Use Case	Olah Akses
Actor	Admin
Description	Admin memberikan instruksi pada sistem untuk melakukan pengolahan akses (tambah, edit, hapus) pada sistem
Pre-Condition	Admin yang terdaftar disistem membuat, mengedit atau menghapus akses pada sistem
Post-Condition	Sistem melakukan eksekusi perintah sesuai dengan perintah yang diberikan admin.
Basic Flow	<ol style="list-style-type: none"> a. Admin login ke sistem b. Sistem memvalidasi admin. c. Setelah login admin dapat membuat, mengedit atau menghapus akses pada sistem.

Tabel 1 Use Case olah user

Use Case	Olah User
Actor	Admin
Description	Admin memberikan instruksi pada sistem untuk melakukan pengolahan user (tambah, edit, hapus) pada sistem.
Pre-Condition	Admin yang terdaftar disistem membuat, mengedit atau menghapus user pada sistem.
Post-Condition	Sistem melakukan eksekusi perintah sesuai dengan perintah yang diberikan admin.
Basic Flow	<ul style="list-style-type: none"> d. Admin login ke sistem e. Sistem memvalidasi admin. f. Setelah login admin dapat membuat, mengedit atau menghapus user pada sistem.

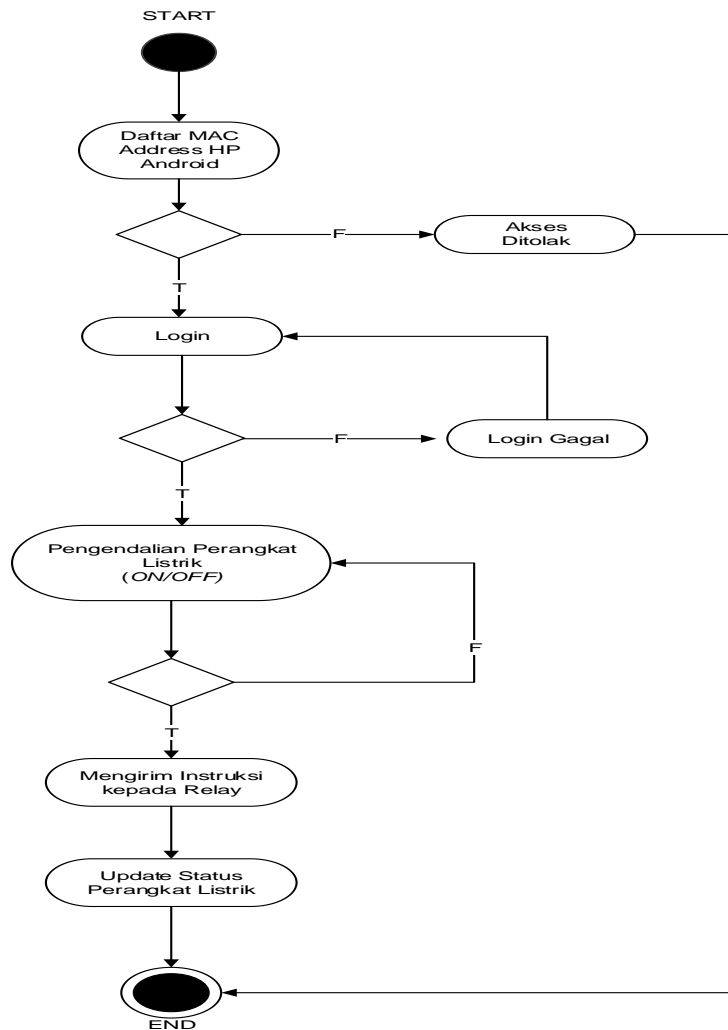
Tabel 4 Use Case Instruksi Pengendalian Lampu

Use Case	Instruksi Pengendalian lampu
Actor	Admin, User
Description	Admin atau User memberikan instruksi pada sistem untuk melakukan instruksi pada perangkat.
Pre-Condition	Admin atau User terdaftar disistem dan melakukan instruksi sesuai dengan aturan yang sudah ditentukan oleh sistem.
Post-Condition	Sistem melakukan eksekusi perintah sesuai instruksi yang diberikan pengguna.
Basic Flow	<ul style="list-style-type: none"> a. Admin atau User memberikan instruksi pada sistem melalui aplikasi Android. b. Sistem akan mengeksekusi instruksi sesuai dengan perintah yang diberikan admin atau user.

3. Activity Diagram

Activity Diagram menggambarkan berbagai alur aktivitas dalam aplikasi yang dirancang, bagaimana masing-masing alur berawal, keputusan yang mungkin terjadi dan bagaimana mereka berakhir.

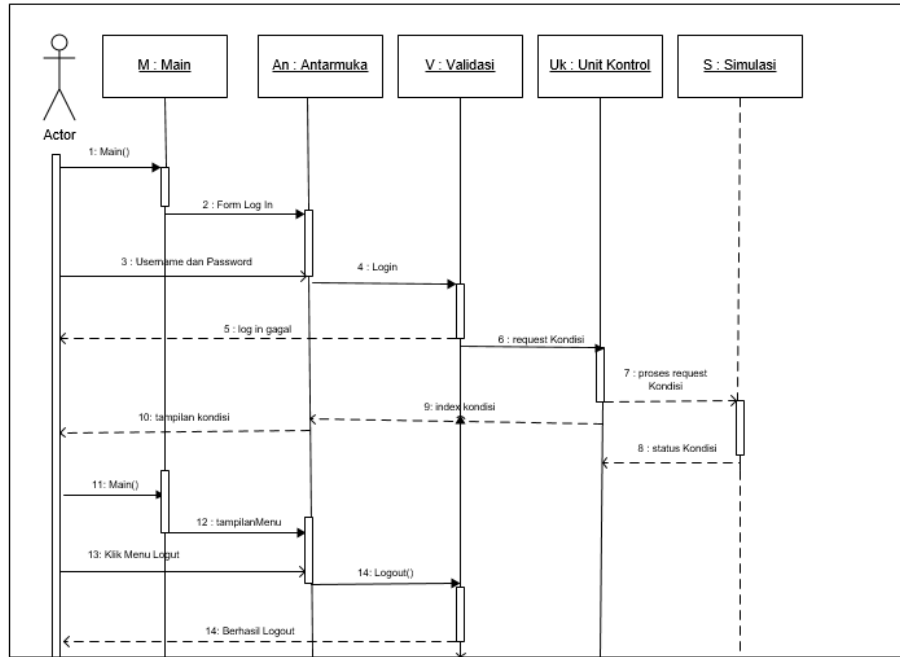
Berikut adalah skema *activity diagram* pada sistem ini.



Gambar 5 Activity Diagram pada aplikasi Android

4. Sequence Diagram

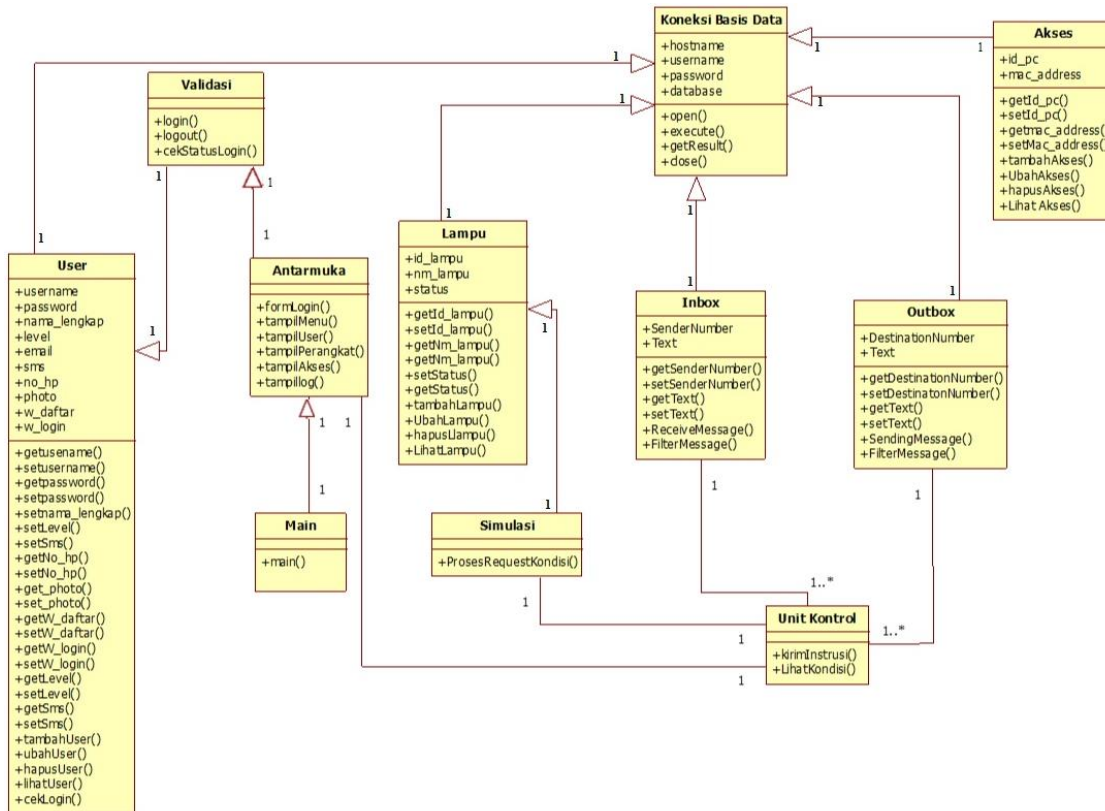
Merupakan bentuk dari interaksi diagram, menggambarkan interaksi antar objek di dalam dan di sekitar sistem berupa pesan yang di gambarkan terhadap waktu. Masing-masing objek termasuk aktor memiliki *line vertical*, pesan digambarkan garis berpanah dari satu objek ke objek lainnya, dan *activation bar* menunjukkan lamanya eksekusi sebuah proses. Sebuah *sequence diagram*, secara khusus menjabarkan *behavior* sebuah skenario tunggal. Diagram tersebut menunjukkan sejumlah objek dan pesan-pesan yang melewati objek- objek ini di dalam *use case*. Berikut ini adalah *sequence diagram* dari aplikasi yang dibangun.



Gambar 6 *Sequence Diagram* pada aplikasi Android

5. Class Diagram

Class diagram adalah sebuah spesifikasi yang jika diinstansiasi akan menghasilkan sebuah objek dan merupakan inti dari pengembangan dan desain berorientasi objek. *Class* menggambarkan keadaan (*attribut* atau *property*) suatu sistem, sekaligus menawarkan layanan untuk memanipulasi keadaan tersebut (metoda atau fungsi). Berikut *class diagram* dari sistem kendali perangkat listrik berbasis Android :



Gambar 7 Class Diagram pada sistem

6. Prosedur aplikasi berbasis Android

1. Untuk akses melalui aplikasi Android ini maka *MAC Address device mobile* tersebut harus sudah terdaftar di aplikasi.
2. Bila *MAC Address device mobile* sudah terdaftar maka silahkan *login* aplikasi Android dengan user yang sudah terdaftar di aplikasi.
3. Setelah login menu halaman aplikasi Android dengan kebijakan sebagai berikut.

Tabel 5 Menu Pada Level Akses Android

Level Akses	Tampilan Menu
Administrator	<ul style="list-style-type: none"> • Remote Lampu
User	<ul style="list-style-type: none"> • Remote Lampu

4. Saat masuk ke menu *remote* perangkat lampu maka akan ada tombol lampu yang sudah di daftarkan ke sistem.

5. Jika akan melakukan *ON/OFF remote* lampu maka tinggal menekan tombol yang tampil di aplikasi tersebut dengan indikator warna tombol pada tiap lampu.
6. Bila akan keluar dari aplikasi maka masuk ke menu *logout* yang sudah tersedia di aplikasi ini.

7. Pengujian

Setelah implementasi selesai, perlu dilakukan pengujian dari setiap spesifikasi yang diperlukan. Berikut ini adalah kasus-kasus yang akan diujikan pada sistem kendali berbasis aplikasi Android, hasil yang di harapkan serta hasil pengujian.

Tabel 6 Hasil Pengujian Pada Aplikasi Android

Kasus dan Hasil Pengujian				
No .	Data Masukan	Yang Diharapkan	Hasil Pengujian	Kesimpulan
1.	Username tidak diisi	Ditampilkan pesan kesalahan	Aplikasi menampilkan pesan kesalahan	[<input checked="" type="checkbox"/>] Diterima [<input type="checkbox"/>] Ditolak
2.	Password tidak diisi	Ditampilkan pesan kesalahan	Aplikasi menampilkan pesan kesalahan	[<input checked="" type="checkbox"/>] Diterima [<input type="checkbox"/>] Ditolak
3.	Username dan Password tidak diisi	Ditampilkan pesan kesalahan	Aplikasi menampilkan pesan kesalahan	[<input checked="" type="checkbox"/>] Diterima [<input type="checkbox"/>] Ditolak
4.	Username dan Password salah	Ditampilkan pesan kesalahan	Aplikasi menampilkan pesan kesalahan	[<input checked="" type="checkbox"/>] Diterima [<input type="checkbox"/>] Ditolak
5.	Menekan tombol “ON” pada tombol remote lampu 1 pada saat kondisi lampu 1 “OFF”	Database <i>status</i> lampu <i>ter-update</i> dan lampu 1 akan menyala dengan adanya perubahan warna pada tombol lampu 1	Database <i>status</i> lampu <i>ter-update</i> dan lampu 1 akan menyala dengan adanya perubahan warna pada tombol lampu 1	[<input checked="" type="checkbox"/>] Diterima [<input type="checkbox"/>] Ditolak
6.	Menekan tombol “ON” pada tombol remote lampu 2 pada saat kondisi lampu 2 “OFF”	Database <i>status</i> lampu <i>ter-update</i> dan lampu 2 akan menyala dengan adanya perubahan pada tombol lampu 2	Database <i>status</i> lampu <i>ter-update</i> dan lampu 2 akan menyala dengan adanya perubahan pada tombol lampu 2	[<input checked="" type="checkbox"/>] Diterima [<input type="checkbox"/>] Ditolak

7.	Menekan tombol “OFF” pada tombol remote lampu 1 pada saat kondisi lampu 1 “ON”	Database <i>status</i> lampu ter- <i>update</i> dan lampu 1 akan mati dengan adanya perubahan warna pada tombol lampu 1	Database <i>status</i> lampu ter- <i>update</i> dan lampu 1 akan mati dengan adanya perubahan warna pada tombol lampu 1	[<input checked="" type="checkbox"/>] Diterima [<input type="checkbox"/>] Ditolak
8.	Menekan tombol “OFF” pada tombol remote lampu 2 pada saat kondisi lampu 2 “ON”	Database <i>status</i> lampu ter- <i>update</i> dan lampu 2 akan mati dengan adanya perubahan warna pada tombol lampu 2	Database <i>status</i> lampu ter- <i>update</i> dan lampu 2 akan mati dengan adanya perubahan warna pada tombol lampu 2	[<input checked="" type="checkbox"/>] Diterima [<input type="checkbox"/>] Ditolak
9.	Pilih Menu “Logout”	Muncul Pesan “Anda yakin akan keluar aplikasi ?” Dan pengguna akan keluar aplikasi	Muncul Pesan “Anda yakin akan keluar aplikasi ?” Dan pengguna akan keluar aplikasi	[<input checked="" type="checkbox"/>] Diterima [<input type="checkbox"/>] Ditolak

Berdasarkan hasil pengujian aplikasi yang dilakukan, maka dapat disimpulkan bahwa pengendali perangkat listrik (dalam aplikasi ini lampu) yang berbasis aplikasi Android yang dibangun sudah berjalan sesuai dengan yang diharapkan baik itu dari segi validasi, instruksi, proses *database* maupun proses *output* yang diinginkan.

4. KESIMPULAN

Sistem kendali berbasis Android ini mempermudah kita untuk mengontrol peralatan listrik dirumah dengan *control* jarak jauh tanpa harus menyalakan saklar listrik secara *manual* (harus mendekat ke saklar listrik untuk menyalakan atau mematikannya).

Sistem kendali berbasis aplikasi Android ini secara *funksional* memenuhi kebutuhan kendali jarak jauh perangkat listrik.

Pada aplikasi ini tidak semua orang dapat terhubung dengan perangkat dikarenakan ada *filter* pengguna yang dipasang pada aplikasi ini, maka tidak sembarang orang bisa menggunakannya, sehingga meminimalisir penyalahgunaan pada sistem ini.

Dalam fungsinya sendiri aplikasi ini dapat menjadi solusi bagi masyarakat yang terkadang sering lupa saat mengontrol peralatan listrik sehingga bisa meminimalisir tagihan listrik yang meningkat ataupun kerugian lainnya seperti hubungan arus pendek

listrik yang diakibatkan karena lupa mematikan perangkat listrik di rumah ataupun dikantor yang dapat mengakibatkan kebakaran dan bahaya lainnya.

5. DAFTAR PUSTAKA

- Booch, G., Rumbaugh, J., & Jacobson, I. 2005, *“The Unified Modeling Language User Guide. 2nd ed.”* Boston, USA: Addison Wesley Professional.
- Budi Irawan, 2005, *“Jaringan Komputer,”* Yogyakarta : Graha Ilmu.
- Camden R., & Matthews A. 2012, *“jQuery Mobile Web Development Essentials,”* United Kingdom: Packt Publishing.
- Elektronika Dasar, 2012, <http://elektronika-dasar.web.id/teori-relay-elektro-mekanik/>
- Jogiyanto Hartono, 2004, *“Pengenalan Komputer: Dasar Ilmu Komputer, Pemrograman, Sistem Informasi dan Intelegensi Buatan,”* Yogyakarta: Andi.
- Jogiyanto, HM, 2005, *“Analisis dan Desain Sistem Informasi : Pendekatan Terstruktur Teori dan Praktik Aplikasi Bisnis,”* Yogyakarta : Andi.
- Mulyanto, Agus. 2009, *“Sistem Informasi (Karakteristik Sistem),”* Bandung : Lingga Jaya.
- Murtiwiayati, & Lauren, Glenn, *“Rancang Bangun Aplikasi Pembelajaran Budaya Indonesia Untuk Anak Sekolah Dasar Berbasis Android,”* Jurnal Ilmiah KOMPUTASI, Volume 12 Nomor : 2, Desember 2013.
- Roger S, Pressman, 2010, *“Software Engineering : a Practitioner’s Approach 7th Edition,”* New York : McGraw-Hill.
- Sutarman, 2009, *“Pengenalan Sistem Informasi,”* Yogyakarta : Ismail.
- Sutarman, 2012, *“Pengantar Teknologi Informasi”* .Jakarta: Bumi Aksara.
- <http://www.usbinov.com/usb-2rel/index.htm>