

BAB III

METODOLOGI

3.1 Analisis Kebutuhan

Analisis kebutuhan adalah tahap dimana dilakukan proses pengumpulan data yang dapat menunjang atau mendukung dari aplikasi yang akan dibuat serta dapat memperoleh jawaban dari rumusan masalah yang telah dibuat sebelumnya. Pengumpulan data yang digunakan penulis adalah menggunakan metode studi pustaka. Studi pustaka adalah sebuah metode dalam pengumpulan data dengan melakukan pencarian informasi melalui media seperti buku dan internet. Studi pustaka dilakukan dengan membaca buku dan mencari literatur dari internet mengenai raspberry pi home automation, mencari informasi mengenai perangkat pendukung baik perangkat keras ataupun perangkat lunak agar sistem ini dapat digunakan sebagaimana fungsi yang diharapkan. Raspberry Pi dapat digunakan sebagai server untuk *monitoring* kondisi lampu dan dibuat untuk melakukan *monitoring* suhu dan otomatisasi lampu yang dikendalikan dari aplikasi web dan aplikasi Android.

3.1.1 Analisis Kebutuhan Fungsi

Analisis kebutuhan fungsi adalah tahapan untuk menentukan fungsi yang dapat dilakukan oleh sistem ini nantinya, sehingga dapat menjawab rumusan masalah yang ada. Sistem ini nantinya akan mempunyai fungsi sebagai berikut:

- a) *Monitoring* dan penjadwalan untuk menyalakan atau mematikan lampu.
- b) Pengendalian menggunakan aplikasi Android dan *website*.
- c) *Monitoring* suhu ruangan untuk menyalakan atau mematikan pendingin.
- d) Manajemen untuk lampu, penjadwalan, *microcontroller*, dan akun.

3.1.2 Analisis Kebutuhan Input

Pada tahapan analisis kebutuhan input/masukan adalah tahapan untuk menentukan masukan yang dibutuhkan dalam *home automation system* ini.

Kebutuhan input/masukan yang dimaksud adalah sebagai berikut:

- a) Manajemen Lampu : Nama, Pin GPIO, ID Microcontroller
- b) Kontrol Lampu : Tombol ON/OFF
- c) Penjadwalan : Judul, Waktu ON, Waktu OFF, Lampu
- d) Microcontroller : Nama, Mac Address
- e) *Monitoring* Suhu : Data Suhu

3.1.3 Analisis Kebutuhan Output

Pada tahapan analisis kebutuhan output/keluaran adalah tahapan untuk menentukan keluaran informasi dari *home automation system* ini. Keluaran informasi yang dimaksud adalah sebagai berikut:

- a) Status nyala/mati lampu ruangan.
- b) Daftar lampu ruangan yang tersedia.
- c) Daftar lampu ruangan yang terjadwal.
- d) Daftar *microcontrolle* yang tersedia.
- e) Suhu ruangan.

3.1.4 Analisis Kebutuhan Antarmuka

Pada tahapan analisis kebutuhan antarmuka adalah menentukan antarmuka yang dibutuhkan dalam *home automation system*, antarmuka yang dimaksudkan adalah sebagai berikut:

- a) Antarmuka untuk aplikasi Android.
- b) Antarmuka untuk *website*.

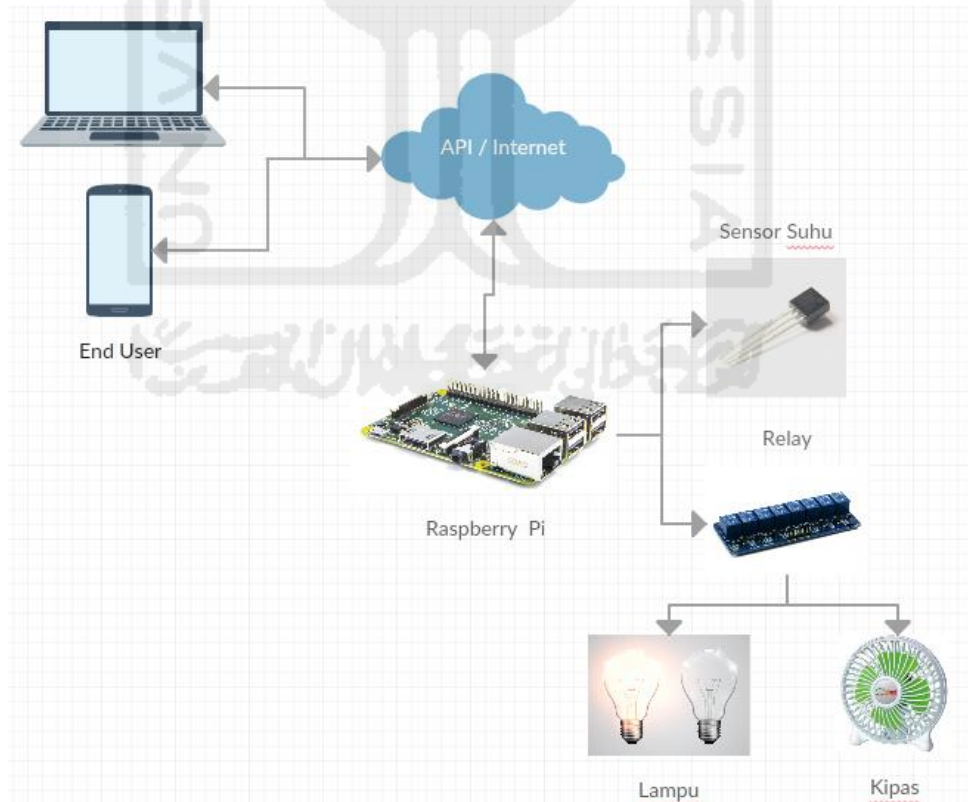
3.2 Perancangan

Pada tahapan data-data yang sudah dianalisa akan dirangkum sehingga menghasilkan rancangan dasar sistem yang akan dibuat.

3.2.1 Perancangan Sistem

Sistem kendali *home automation* disini mempunyai dua jenis kendali yaitu sistem kendali otomatis menggunakan jadwal, dan menggunakan aplikasi Android

dan menggunakan *website*. Dari rangkaian *home automation system* pada gambar dibawah ini seperti yang terdapat pada Gambar 3. 1, terdapat beberapa bagian yang mempunyai peran yang berbeda antara satu dengan yang lain dalam *home automation system* ini, seperti API yang menjadi perantara antara *end user* dengan Raspberry Pi. Jika ada permintaan atau *request* seperti untuk menyalakan atau mematikan lampu dari *end user*, maka permintaan tersebut dikirim ke API, kemudian API melakukan *broadcast event* (apabila terjadi perubahan statu lampu pada *database* sistem kendali rumah otomatis) pada Raspberry Pi yang terhubung dengan API setelah itu Raspberry Pi memberikan *response* dari permintaan sebelumnya untuk melakukan yang perintah sesuai, kemudian menghasilkan keluaran atau *output* status lampu nyala atau mati. Gambaran alur kerja dan hubungan antar komponen dalam sistem ini seperti yang terlihat pada gambar dibawah ini:

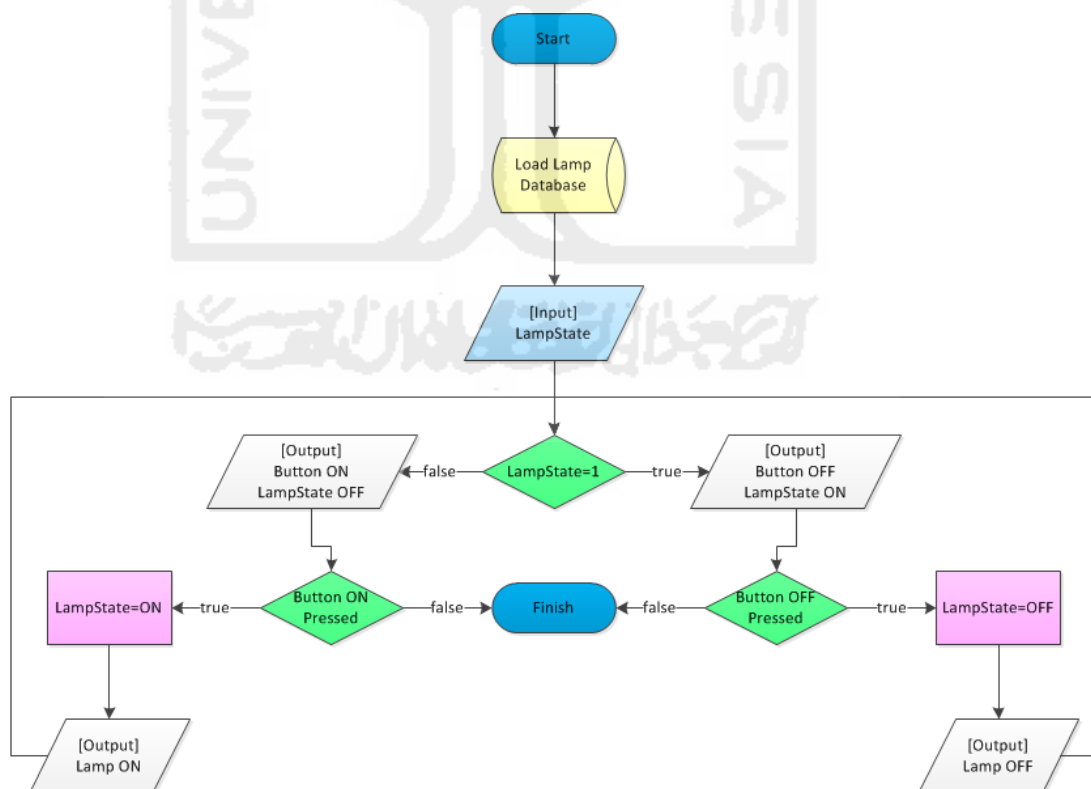


Gambar 3. 1: Rancangan sistem

3.2.2 Perancangan Perangkat Lunak

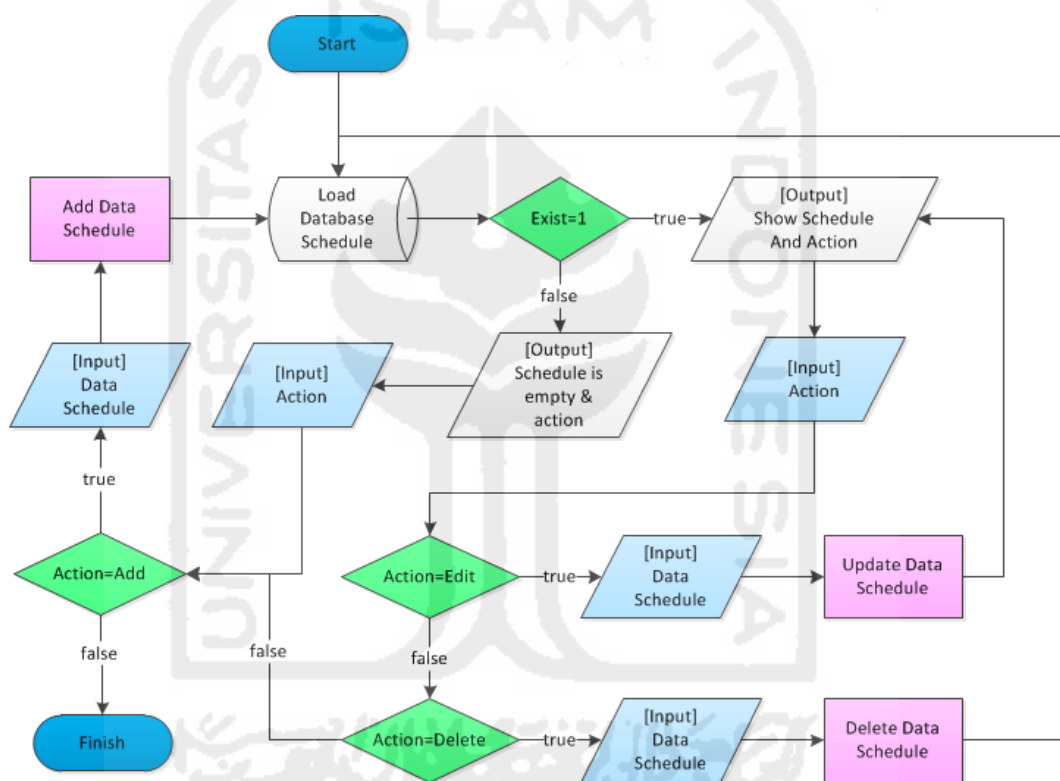
Perancangan perangkat lunak pada *home automation system* ini terdapat 3 alur program (kontrol nyala/mati lampu, penjadwalan, dan Raspberry Pi) yang berjalan. Berikut adalah diagram alur program untuk *home automation system* beserta penjelasannya:

Pertama kali program kontrol lampu dijalankan, program memuat *database* lampu dan mengecek status terakhir lampu, kemudian dari status tersebut menjadi *input*/masukan untuk proses selanjutnya, apabila lampu berstatus *on*/menyala maka *output*/keluarannya adalah status *on*/menyala dan tombol *off*, sebaliknya apabila lampu berstatus *off*/mati, maka *output*/keluarannya adalah status *off*/mati dan tombol *on*. Tombol pada bagian ini diibaratkan seperti saklar, apabila tombol *on* ditekan, maka status lampu berubah menjadi *on*/nyala begitu juga sebaliknya seperti yang terdapat pada Gambar 3. 2.



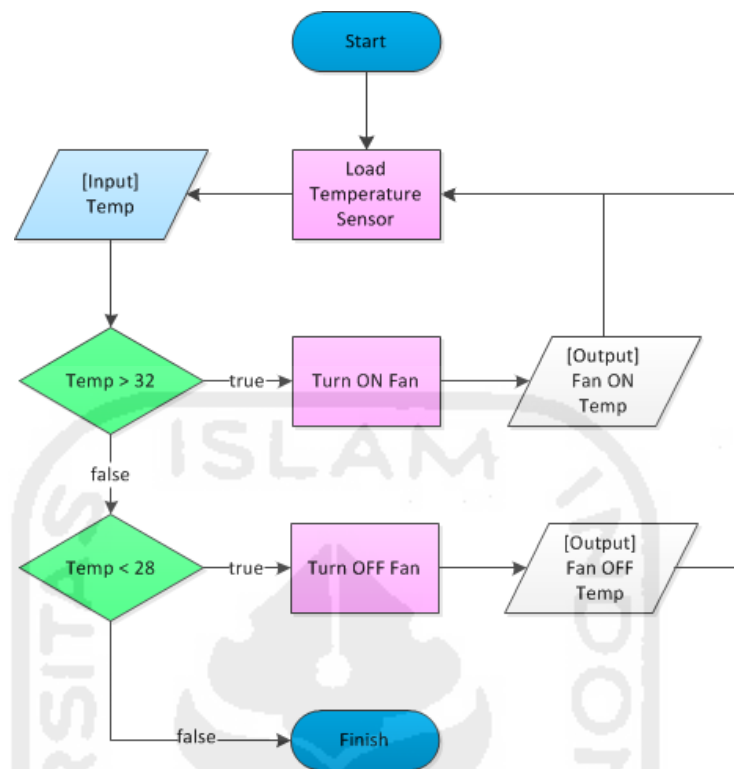
Gambar 3. 2: Flowchart program nyala/mati lampu

Pertama kali program penjadwalan untuk menyalakan/mematikan lampu dijalankan, program memuat *database* penjadwalan, apabila *database* penjadwalan kosong maka sistem menampilkan bahwa status jadwal adalah kosong dan muncul tombol untuk menambah jadwal, tetapi apabila *database* sudah terisi maka menampilkan daftar jadwal yang tersedia pada *database* dan muncul juga tombol (tambah, edit, hapus) untuk manajemen jadwal, seperti yang terdapat pada Gambar 3. 3 dibawah ini.

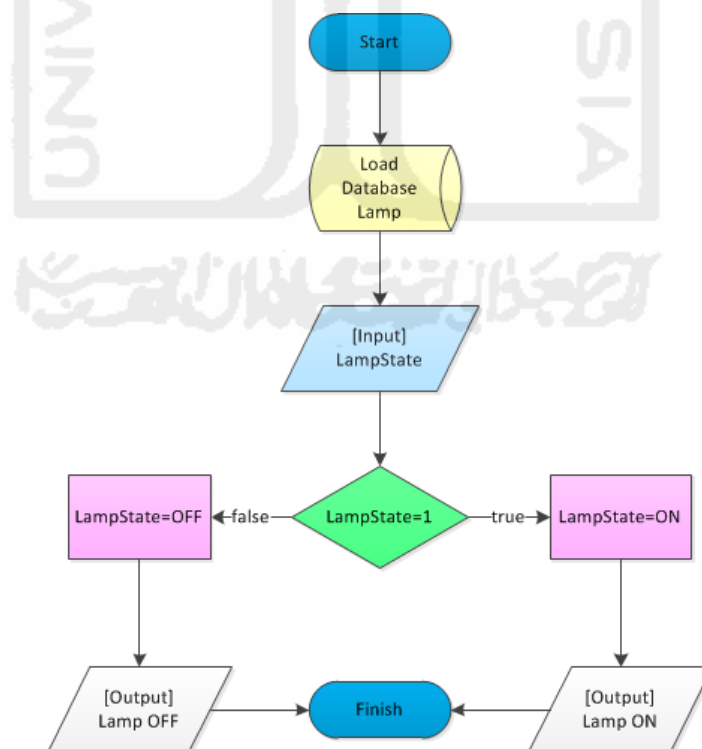


Gambar 3. 3: Flowchart program penjadwalan

Gambar 3. 4 menjelaskan alur program untuk *monitoring* suhu ruangan, ketika program tersebut dijalankan raspberry pi memuat data sensor suhu kemudian hasilnya yang berupa suhu menjadi *input* untuk proses selanjutnya, apabila suhu $> 32^{\circ}\text{C}$ maka kipas menyala, apabila suhu $< 28^{\circ}\text{C}$ maka kipas mati.



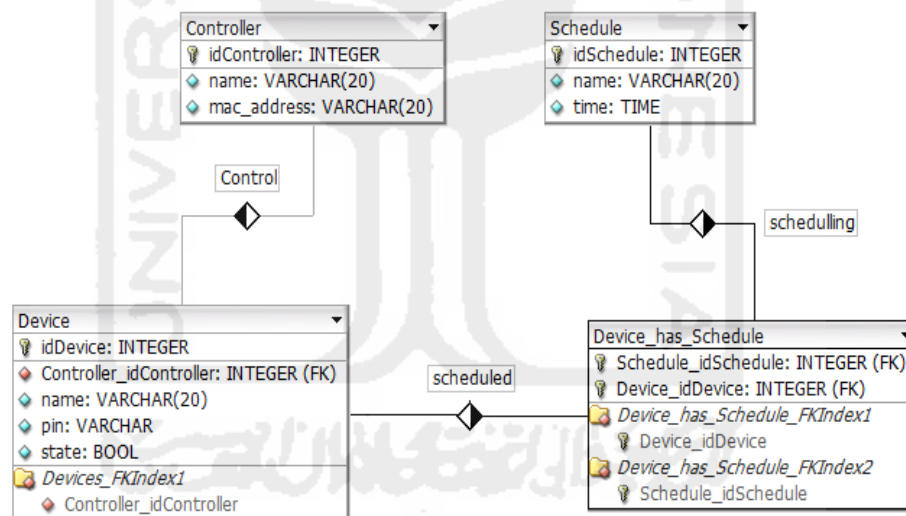
Gambar 3. 4: Flowchart program monitoring suhu ruangan pada Raspberry Pi



Gambar 3. 5: Flowchart program kontrol lampu pada Raspberry Pi

Setelah melakukan perancangan alur program dilakukan perumusan untuk menentukan entitas apa saja yang nantinya akan terlibat dan kemudian entitas tersebut akan menjadi *database* dalam sistem ini. Berikut adalah perancangan entitas *database* sistem.

Seperti yang terdapat pada Gambar 3. 6, entitas yang akan terlibat dalam *home automation system* ini ada 3 entitas, (a) *Controller*, merupakan *microcontroller* yang mempunyai relasi atau hubungan dengan *Device* dan sekaligus bertugas untuk mengontrol *Device* tersebut, (b) *Device* sendiri adalah perangkat-perangkat elektronik yang terhubung dengan *microcontroller*, (c) *Schedule* adalah entitas yang mengatur waktu menyalakan atau mematikan perangkat-perangkat elektronik, sedangkan entitas *Device_has_Schedule* merupakan entitas yang terbentuk dari hasil relasi *Device* dan *Schedule*.

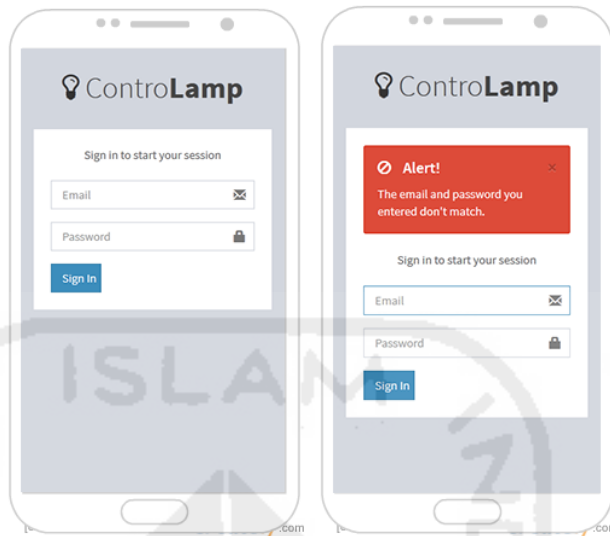


Gambar 3. 6: Relasi tabel *database*

3.2.3 Perancangan Antarmuka

Perancangan antarmuka untuk *home automation system* ini akan menggunakan dua jenis antarmuka, yaitu antarmuka untuk *smartphone* Android dan antarmuka *website*. Berikut adalah rancangan antarmuka *home automation system*-nya:

- Antarmuka pada aplikasi Android



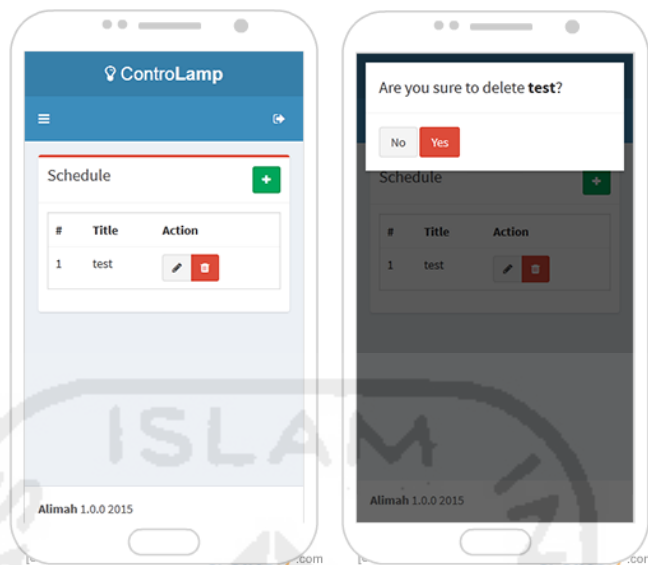
Gambar 3. 7: Antarmuka login

Antarmuka login adalah halaman yang pertama kali dimuat pada saat aplikasi baru berjalan.



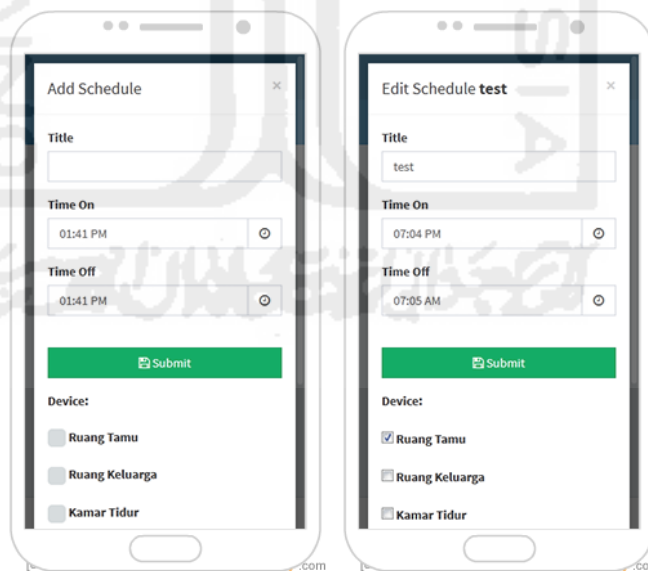
Gambar 3. 8: Antarmuka navigasi dan kontrol ruangan

Setelah *user* berhasil login, *user* diarahkan pada halaman dashboard yang berisi tombol untuk menyalakan atau mematikan lampu. Jika ingin berpindah halaman *user* dapat menekan tombol menu navigasi.

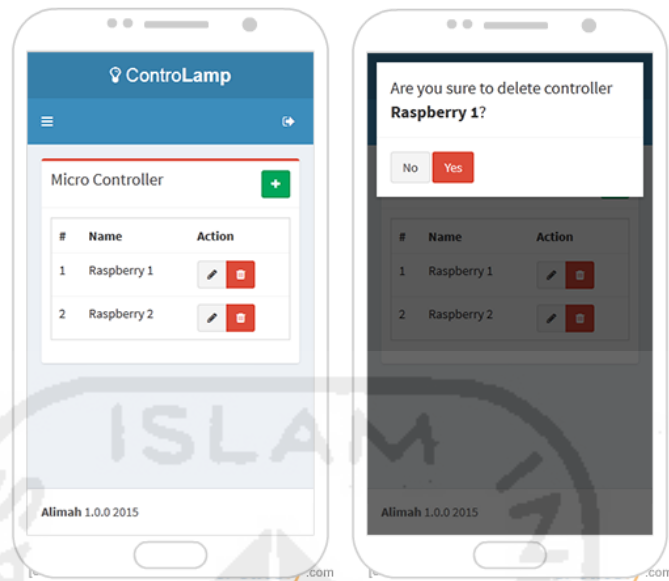


Gambar 3. 9: Antarmuka penjadwalan ruang

Halaman ini tampil apabila *user* memilih menu *Scheduling* pada navigasi. Halaman ini berfungsi untuk manajemen (tambah, edit, hapus) jadwal lampu ruangan seperti yang terlihat pada Gambar 3. 9 dan Gambar 3. 10.

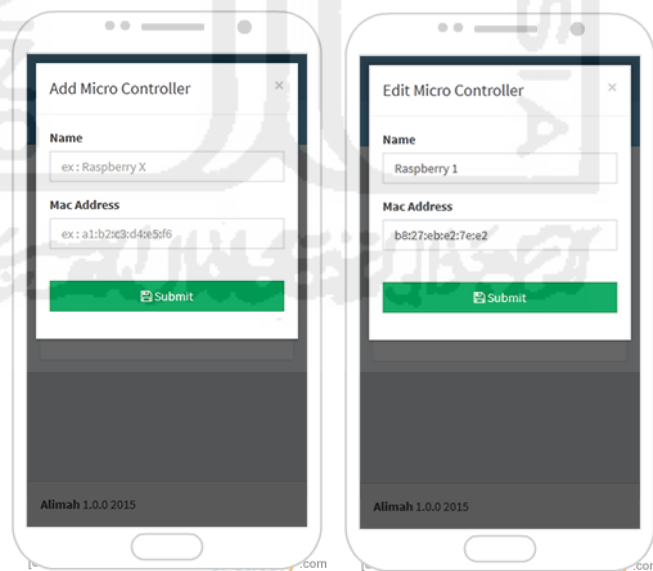


Gambar 3. 10: Antarmuka penjadwalan ruang (tambah dan edit)



Gambar 3. 11: Antarmuka *microcontroller*

Halaman ini tampil apabila *user* memilih menu *Controller* pada navigasi. Halaman ini berfungsi untuk manajemen (tambah, edit, hapus) *microcontroller* seperti yang terlihat pada Gambar 3. 11 dan Gambar 3. 12.

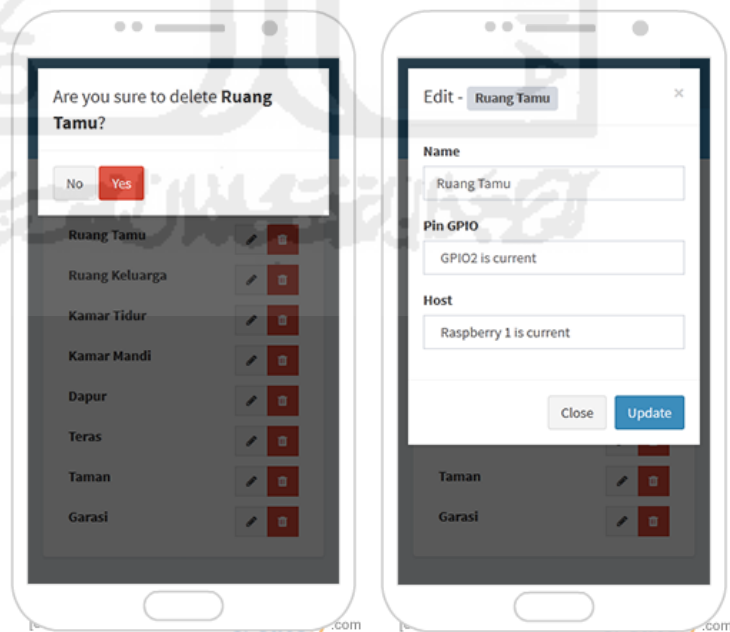


Gambar 3. 12: Antarmuka *microcontroller* (tambah dan edit)



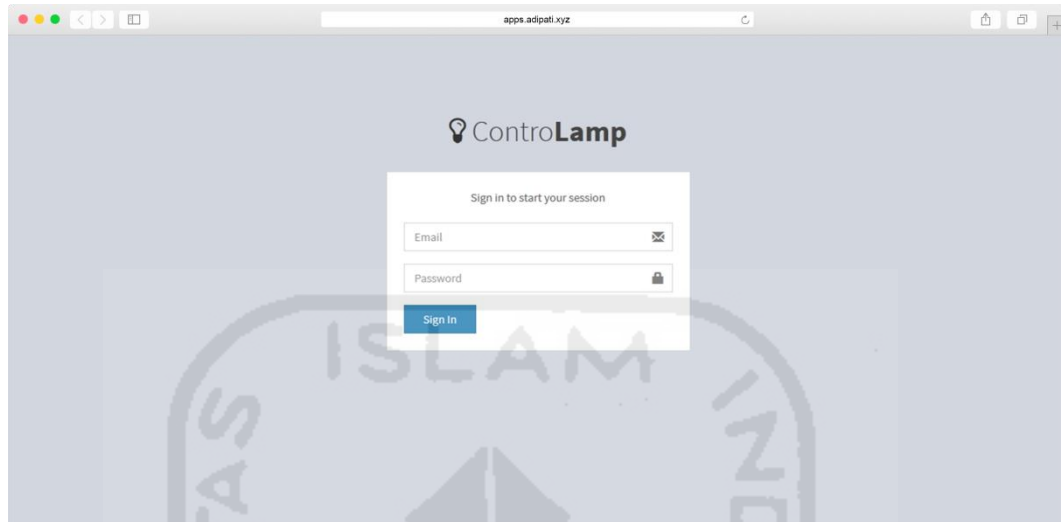
Gambar 3. 13: Antarmuka pengaturan

Halaman ini tampil apabila *user* memilih menu *Setting* pada navigasi. Halaman ini berfungsi untuk manajemen (tambah, edit, hapus) ruangan dan manajemen (*update email* dan *update password*) akun.



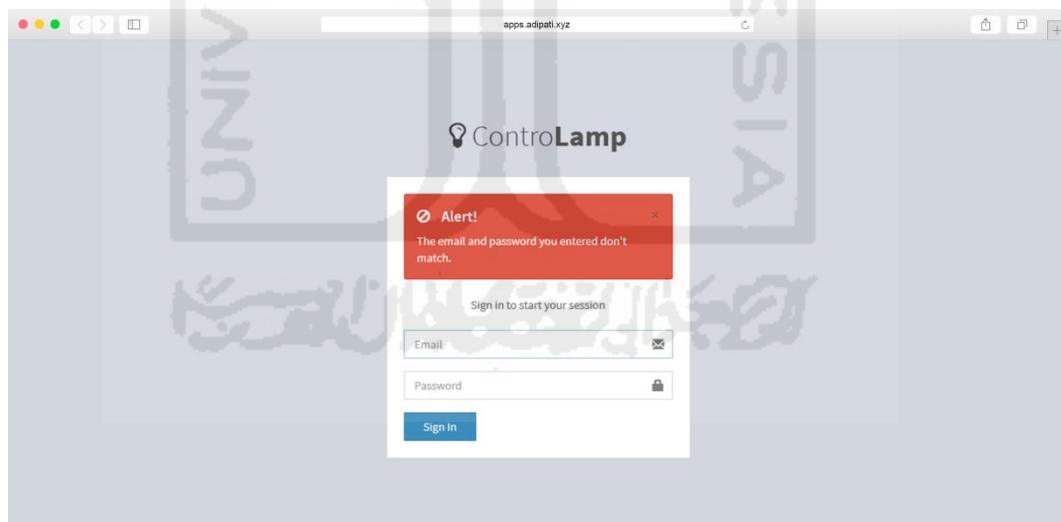
Gambar 3. 14: Antarmuka hapus dan edit ruangan

- Antarmuka *website*

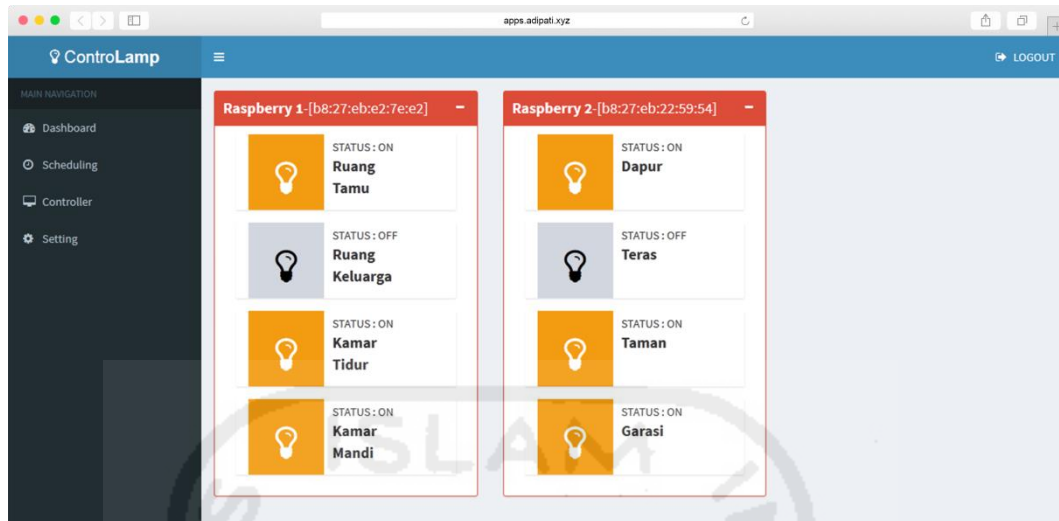


Gambar 3. 15: Antarmuka *login*

Sama seperti tampilan antarmuka pada aplikasi Android, halaman *login* pada *website* dimuat pertama kali ketika url diakses oleh *user*.

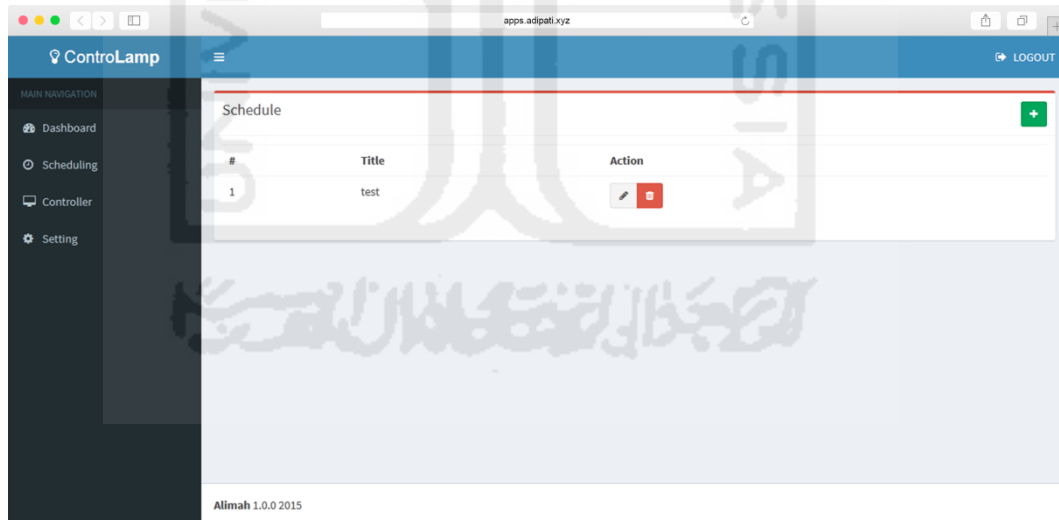


Gambar 3. 16: Antarmuka *login* jika terjadi *error*



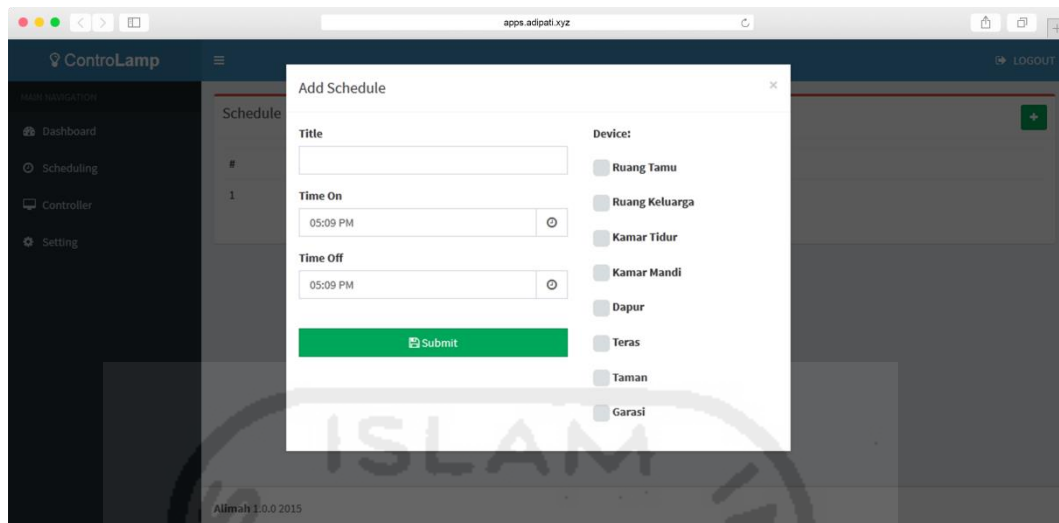
Gambar 3. 17: Antarmuka kontrol ruangan

Halaman kontrol ruang atau *Dashboard* adalah halaman yang ditampilkan ketika *user* berhasil *login*. Pada aplikasi *home automation system* ini terdapat empat bagian menu, yaitu: *Dashboard*, *Scheduling*, *Controller*, dan *Setting* yang mempunyai fungsi sendiri-sendiri.




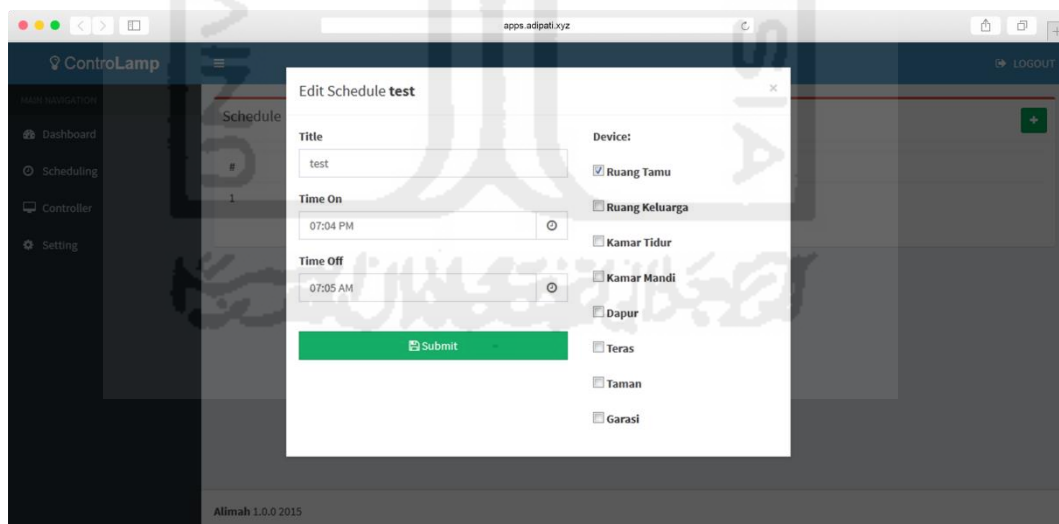
Gambar 3. 18: Antarmuka penjadwalan

Halaman penjadwalan berfungsi untuk manajemen (tambah, edit, hapus) jadwal seperti yang sudah dijelaskan sebelumnya pada perancangan antarmuka aplikasi Android .




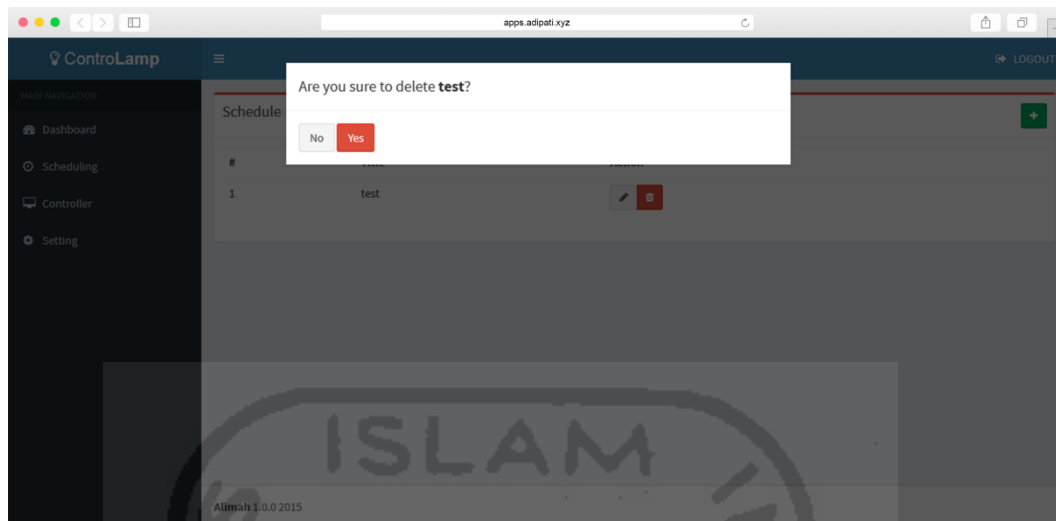
Gambar 3. 19: Antarmuka penjadwalan (tambah)

Halaman tambah jadwal tampil ketika *user* menekan tombol  pada halaman penjadwalan, kemudian *user* menentukan waktu *on*, waktu *off*, serta memilih lampu yang akan masuk penjadwalan seperti yang terlihat pada Gambar 3. 19.




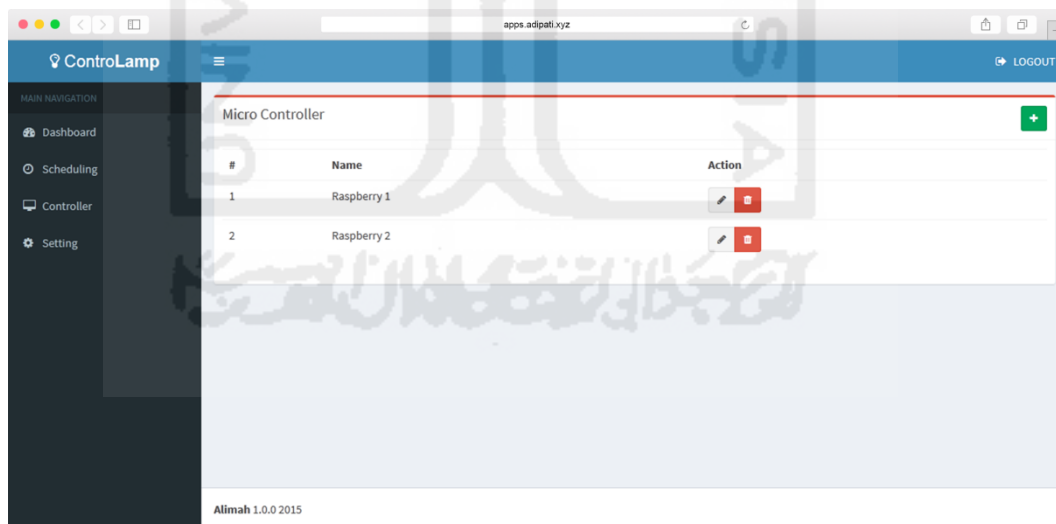
Gambar 3. 20: Antarmuka penjadwalan (edit)

Halaman edit jadwal tampil ketika *user* menekan tombol  yang berisi data dari jadwal yang sudah dibuat sebelumnya. Apabila *user* ingin mengubah jadwal sebelumnya.



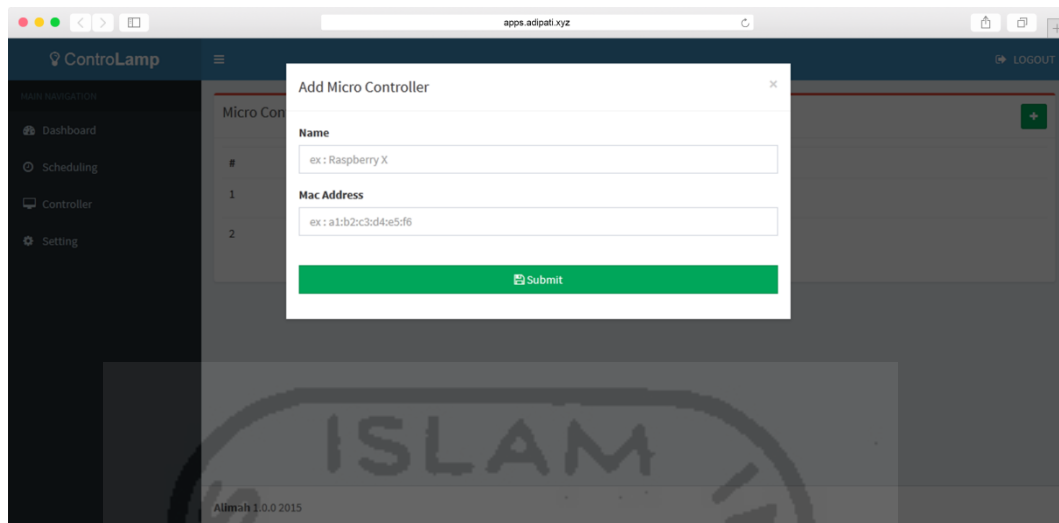
Gambar 3. 21: Antarmuka penjadwalan (hapus)

Apabila tombol hapus  tertekan baik dengan sengaja atau tidak, maka keluar konfirmasi untuk melanjutkan menghapus atau tidak. Konfirmasi ini berguna untuk mencegah proses hapus yang terjadi secara tidak sengaja seperti yang terlihat pada Gambar 3. 21.



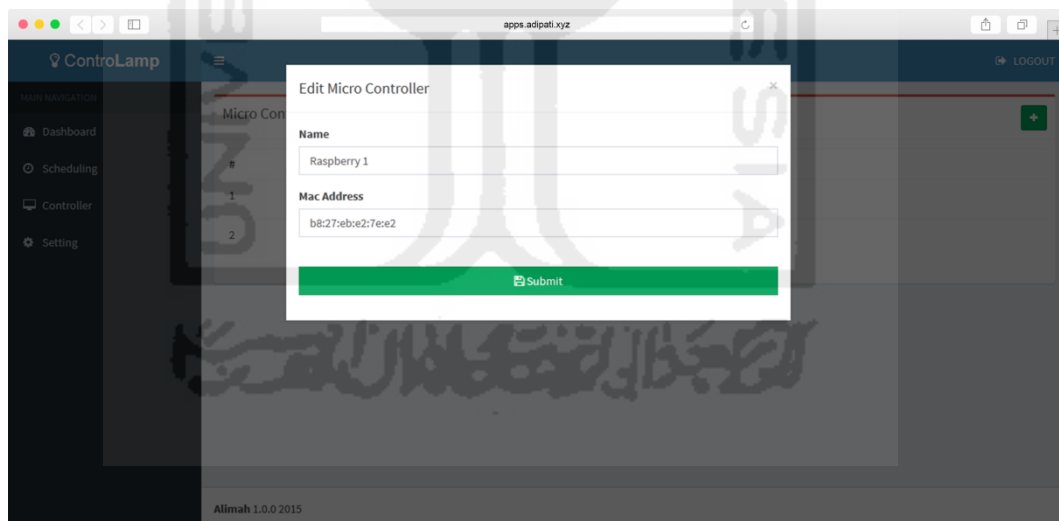
Gambar 3. 22: Antarmuka *microcontroller*

Halaman antarmuka *microcontroller* menampilkan daftar *microcontroller* yang terdaftar pada *home automation system* ini.



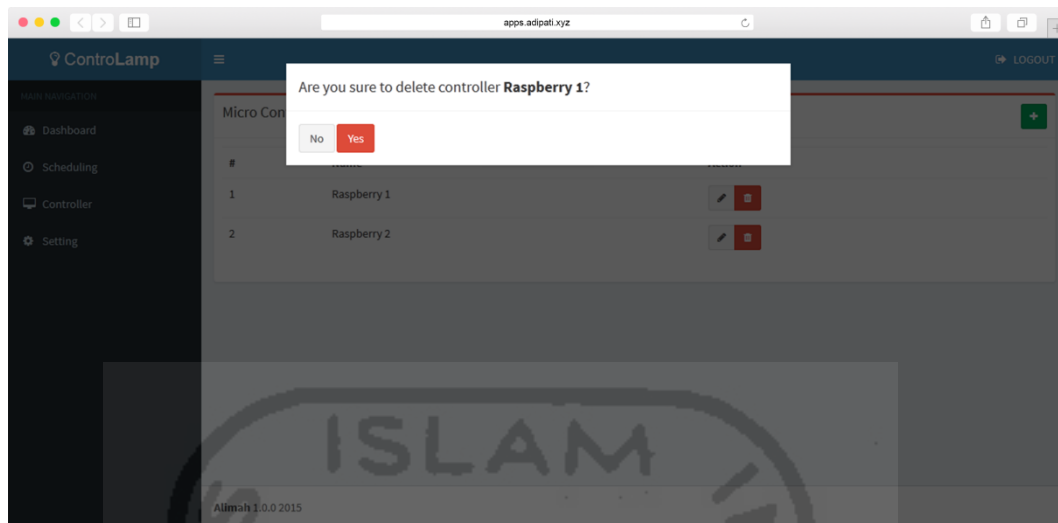
Gambar 3. 23: Antarmuka *microcontroller* (tambah)

Halaman antarmuka tambah *microcontroller* digunakan untuk mendaftarkan *microcontroller* baru pada *home automation system* dengan memasukkan ‘nama’ dan ‘mac address’ dari *microcontroller*.



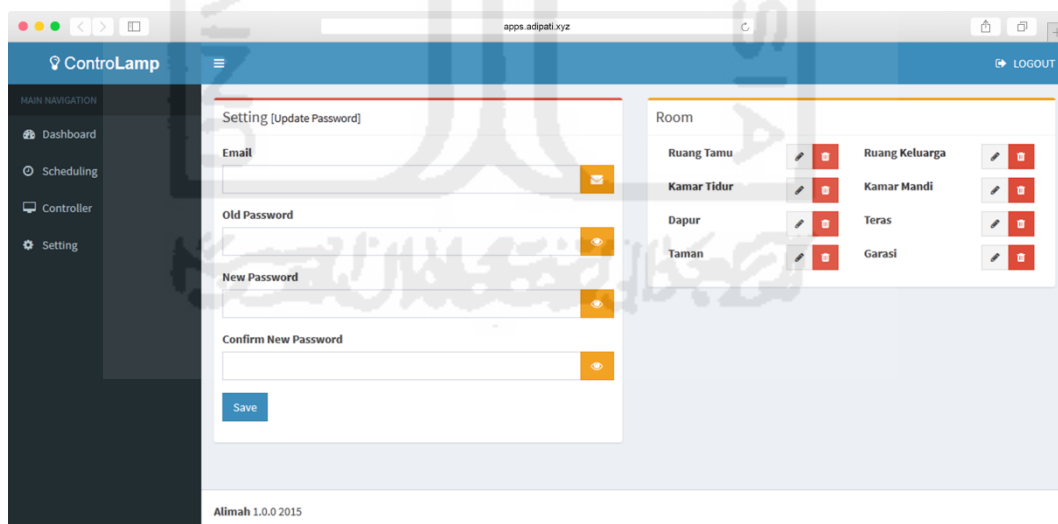
Gambar 3. 24: Antarmuka *microcontroller* (edit)

Halaman antarmuka edit *microcontroller* berfungsi untuk melakukan *update* jika terjadi perubahan/kesalahan saat melakukan pendaftaran *microcontroller* yang dilakukan sebelumnya.



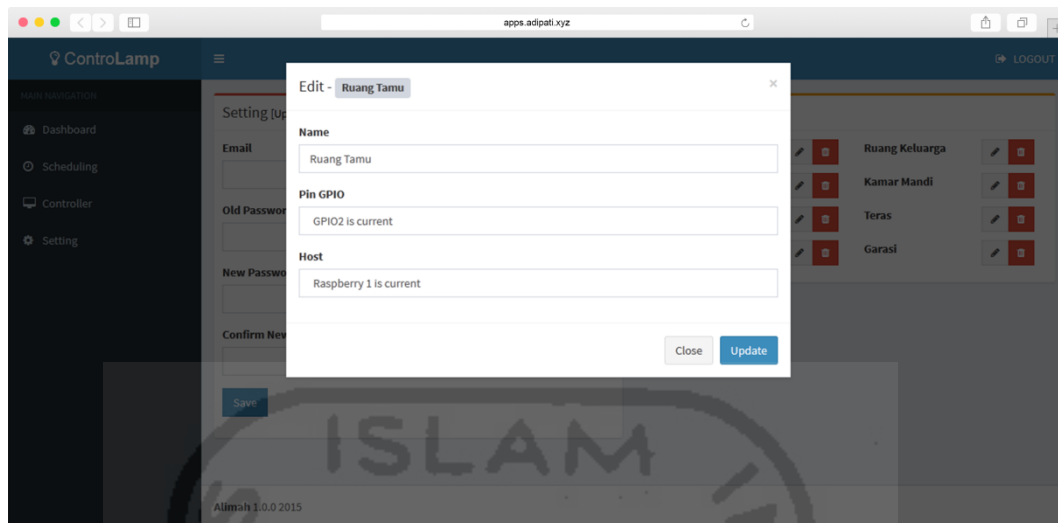
Gambar 3. 25: Antarmuka *microcontrolle* (hapus)

Halaman antarmuka hapus *microcontroller* berfungsi untuk menghapus *microcontroller* dari *database* sistem ini. Jika *user* ingin menghapus *microcontroller* dari *database* maka keluar konfirmasi untuk melanjutkan atau tidak proses hapus, konfirmasi ini bertujuan untuk mencegah terjadinya proses hapus apabila tombol hapus tertekan secara tidak sengaja.



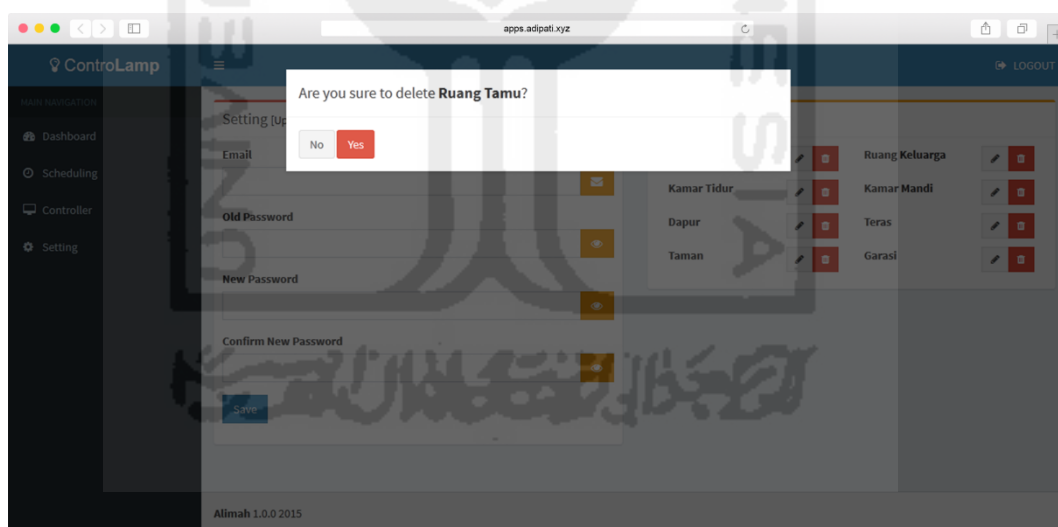
Gambar 3. 26: Antarmuka pengaturan

Halaman antarmuka pengaturan berfungsi apabila *user* berkehendak untuk memperbarui *email* atau *password* dan memperbarui atau menghapus ruangan yang terdaftar pada sistem.



Gambar 3. 27: Antarmuka edit ruangan

Halaman edit ruangan berfungsi untuk memperbarui informasi ruangan seperti nama, pin gpio, dan *host (microcontroller)* yang menangani ruangan tersebut.



Gambar 3. 28: Antarmuka hapus ruangan

Halaman antarmuka hapus ruangan berfungsi untuk menghapus ruangan yang terdaftar pada *database* sistem, seperti fungsi hapus yang lainnya apabila *user* berkehendak ingin menghapus ruangan maka keluar konfirmasi untuk melanjutkan atau tidak menghapus.

3.2.4 Perancangan Perangkat Keras

Perancangan perangkat keras yaitu seluruh perangkatnya yang ada akan diintegrasikan sehingga fungsi-fungsi yang ada dapat dijalankan, seperti yang sudah dijelaskan sebelumnya. Maka perancangan pertama yaitu menghubungkan pin *relay* dengan pin *raspberry pi*. *Raspberry Pi* versi 1 sendiri mempunyai beberapa jenis model seperti model A, B, dan B+ mempunyai perbedaan satu dengan yang lainnya. Tetapi pada *Raspberry Pi* model A dan B mempunyai kesamaan yaitu sama memiliki 26 jumlah total pin, sedangkan model B+ memiliki 40 jumlah total pin seperti yang digunakan dalam penelitian ini tetapi untuk berkomunikasi dengan relay yang digunakan cukup menggunakan 10 pin.

Dalam perancangan perangkat keras dalam *home automation system* ini adalah menghubungkan antara *raspberry pi* dan *relay*. Untuk menghubungkan antara keduanya membutuhkan penghubung berupa kabel, kabel yang digunakan yaitu jenis kabel *jumper female to female* yang menghubungkan antara pin *raspberry pi* dengan pin *relay* seperti Gambar 3. 29 di bawah ini:



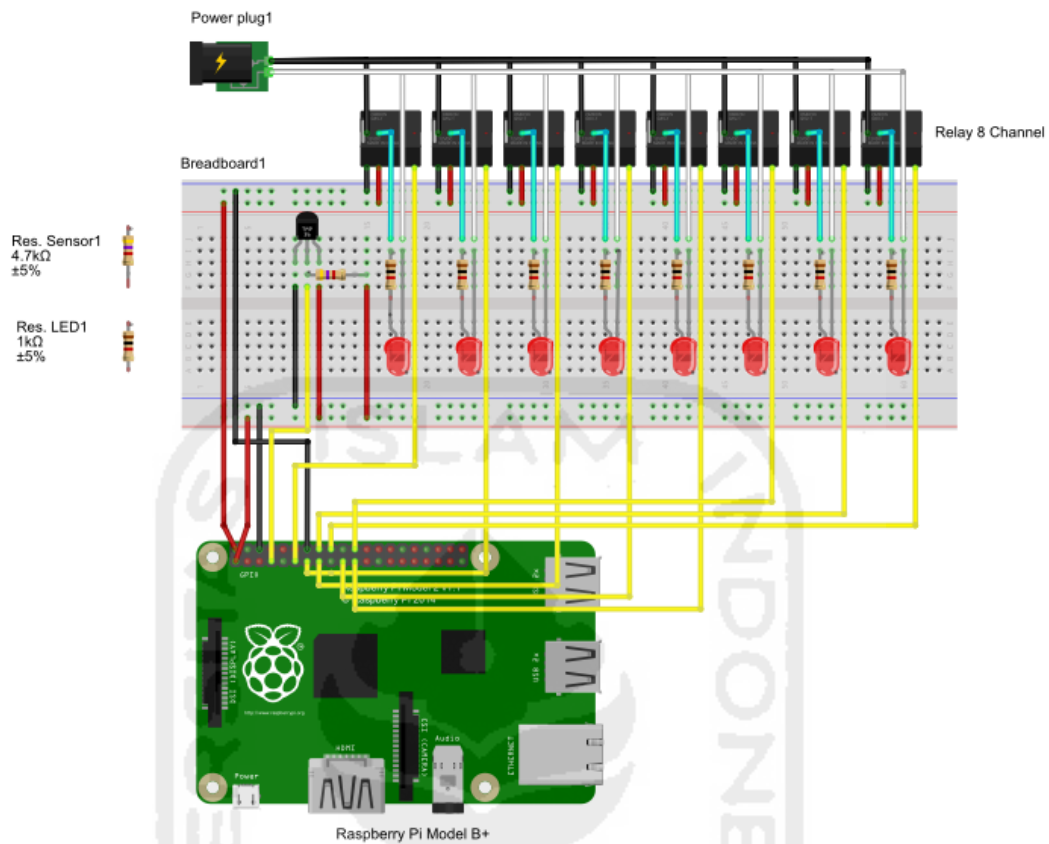
Gambar 3. 29: Kabel *jumper female to female*

Kabel *jumper female to female* (kedua ujung berlubang) seperti pada Gambar 3. 29 merupakan perantara dari *raspberry pi* untuk mengontrol status *relay* menjadi *on* atau *off*. Teknis penghubungan antara *raspberry pi* dan *relay* adalah dengan cara salah satu ujungnya dihubungkan dan disesuaikan dengan pin yang ada pada *raspberry pi* dan ujung kabel satunya di hubungkan pada pin *relay*. Pin pada *raspberry pi* mempunyai fungsi atau tugas yang berbeda antar masing-masing pin, seperti yang terlihat pada Gambar 3. 30, pin yang berwarna merah berfungsi sebagai power(+) atau dengan nama lain VCC terdapat dua jenis tipe power yaitu 5V dan 3.3V, kemudian pin berwarna hitam berfungsi sebagai power(-) atau dengan nama lain GND, pin berwarna kuning atau pin yang bernama GPIOX berfungsi *input/output*, dan lain-lain.

Raspberry Pi B+
B+ J8 GPIO Header

	Pin No.		
3.3V	1	2	5V
GPIO2	3	4	5V
GPIO3	5	6	GND
GPIO4	7	8	GPIO14
GND	9	10	GPIO15
GPIO17	11	12	GPIO18
GPIO27	13	14	GND
GPIO22	15	16	GPIO23
3.3V	17	18	GPIO24
GPIO10	19	20	GND
GPIO9	21	22	GPIO25
GPIO11	23	24	GPIO8
GND	25	26	GPIO7
DNC	27	28	DNC
GPIO5	29	30	GND
GPIO6	31	32	GPIO12
GPIO13	33	34	GND
GPIO19	35	36	GPIO16
GPIO26	37	38	GPIO20
GND	39	40	GPIO21

Gambar 3. 30: Pin GPIO Raspberry Pi



Gambar 3. 31: Skema pemasangan *relay* dan sensor suhu

3.3 Implementasi

Setelah melakukan proses analisis kebutuhan sistem dan perancangan sistem, tahap selanjutnya adalah melakukan tahap implementasi yaitu melakukan penerapan dari proses yang sudah dilakukan sebelumnya sehingga menghasilkan sistem yang dapat menjalankan beberapa fungsi sebagai berikut:

- Monitoring* dan kendali saklar lampu melalui *website* dan aplikasi Android.
- Monitoring* suhu ruangan untuk menyalakan/mematikan kipas.
- Penjadwalan saklar lampu sesuai dengan waktu yang ditentukan.
- Monitoring* Raspberry Pi apabila tidak terhubung dengan API.

3.3.1 Implementasi Perangkat Keras

Perangkat keras yang dibutuhkan dalam sistem *home automation* yang sesuai dengan analisa kebutuhan masukan, berikut daftar komponen yang digunakan:

a) Raspberry Pi

Raspberry Pi merupakan komputer yang memiliki ukuran sebesar kartu kredit atau yang disebut juga sebagai *mini computer*, Raspberry Pi yang akan digunakan untuk *home automation system* adalah Raspberry Pi 2 model B+ dengan kemampuan yang cukup mumpuni untuk menjalankan sistem ini selain itu dengan harganya yang terjangkau. Raspberry Pi ini yang nanti akan berperan sebagai pusat kendali *home automation system* dengan menggunakan sistem operasi yang berbasis *open source*.

b) *Smartphone*

Smartphone adalah bagian dari *home automation system* ini dan sebagai *end device* atau perangkat yang berinteraksi langsung dengan pengguna. *Smartphone* yang digunakan dalam *home automation system* ini adalah *smartphone* berbasis Android yang nantinya bertugas untuk mengirim perintah dari aplikasi ke *server* kemudian diproses sehingga menghasilkan keluaran sesuai kebutuhan.

c) *Relay*

Relay adalah sebuah saklar yang dikendalikan dengan sinyal arus listrik untuk mengubah menjadi *on* atau *off* secara otomatis. Dalam penelitian ini penulis menggunakan *Relay* buatan pabrik yang sudah jadi, sehingga bisa langsung digunakan. *Relay* yang digunakan yaitu berjumlah 8 *port* (8 saklar) dan membutuhkan arus listrik 5 volt untuk menyalakan *relay* tersebut. *Relay* ini akan terpasang pada Raspberry Pi dengan menggunakan beberapa kabel 8 kabel untuk kontrol dan 2 kabel untuk *power*.

d) Sensor

Sensor merupakan komponen yang nantinya akan bertugas mengendalikan perintah sesuai dengan keadaan yang telah ditentukan. Sensor yang digunakan dalam *home automation system* ini adalah sensor suhu yang

nantinya berperan untuk menyalakan pendingin apabila suhu ruangan tinggi atau melebihi batas suhu yang telah ditentukan sebelumnya.

3.3.2 Implementasi Perangkat Lunak

Perangkat lunak yang dibutuhkan untuk mendukung pembangunan *home automation system* dalam penelitian ini adalah sebagai berikut:

a) Sistem Operasi (Raspbian Wheezy)

Sistem operasi adalah program yang pada sistem komputer yang mengelola semua perangkat dan menjalankan aplikasi dan juga berfungsi sebagai pengelola operasi-operasi dasar pada sistem komputer.

Sistem operasi Raspbian Wheezy ini merupakan sistem operasi linux yang dikembangkan dari sistem operasi Debian Wheezy yang dikhususkan untuk *mini computer* Raspberry Pi. sistem operasi ini yang digunakan pada Raspberry Pi nantinya.

b) Database (MySQL)

Database atau basis data adalah merupakan kumpulan data yang disimpan secara sistematis di dalam komputer yang dapat diolah atau dimanipulasi menggunakan perangkat lunak (program aplikasi) untuk menghasilkan informasi.

MySQL adalah sebuah aplikasi *database* atau basis data yang digunakan untuk menyimpan semua informasi *home automation system*.

c) Web Server (Nginx)

Web server merupakan perangkat lunak pada komputer *server* yang berfungsi menerima permintaan(*request*) berupa halaman web melalui protokol HTTP atau HTTPS dari *client(browser)* dan mengirimkan kembali(*response*) dari permintaan tersebut kedalam bentuk halaman web atau dokumen HTML.

Nginx adalah sebuah aplikasi *web server* bertugas menjalankan *request* HTTP. Aplikasi inilah yang akan menjadi antar muka dari pengguna ke *home automation system*.

3.4 Pengujian dan Analisis Sistem

Setelah sistem diimplementasikan, tahap selanjutnya adalah melakukan pengujian sistem yang mengacu pada poin-poin sebagai berikut:

- a) Pengujian *monitoring* saklar lampu melalui aplikasi berbasis *website* dan aplikasi berbasis Android.
- b) Pengujian *monitoring* suhu ruangan.
- c) Pengujian penjadwalan saklar lampu.
- d) Pengujian menyalakan dan mematikan saklar lampu.

Setelah sistem diuji, maka langkah selanjutnya adalah perlu analisis tentang kelebihan dan kekurangan terhadap sistem yang telah dibangun. Analisis penting untuk dilakukan karena untuk keberlanjutan penelitian selanjutnya dan perbaikan sistem yang telah dibuat.