



Hak cipta dan penggunaan kembali:

Lisensi ini mengizinkan setiap orang untuk menggubah, memperbaiki, dan membuat ciptaan turunan bukan untuk kepentingan komersial, selama anda mencantumkan nama penulis dan melisensikan ciptaan turunan dengan syarat yang serupa dengan ciptaan asli.

Copyright and reuse:

This license lets you remix, tweak, and build upon work non-commercially, as long as you credit the origin creator and license it on your new creations under the identical terms.

BAB III

METODOLOGI PENELITIAN

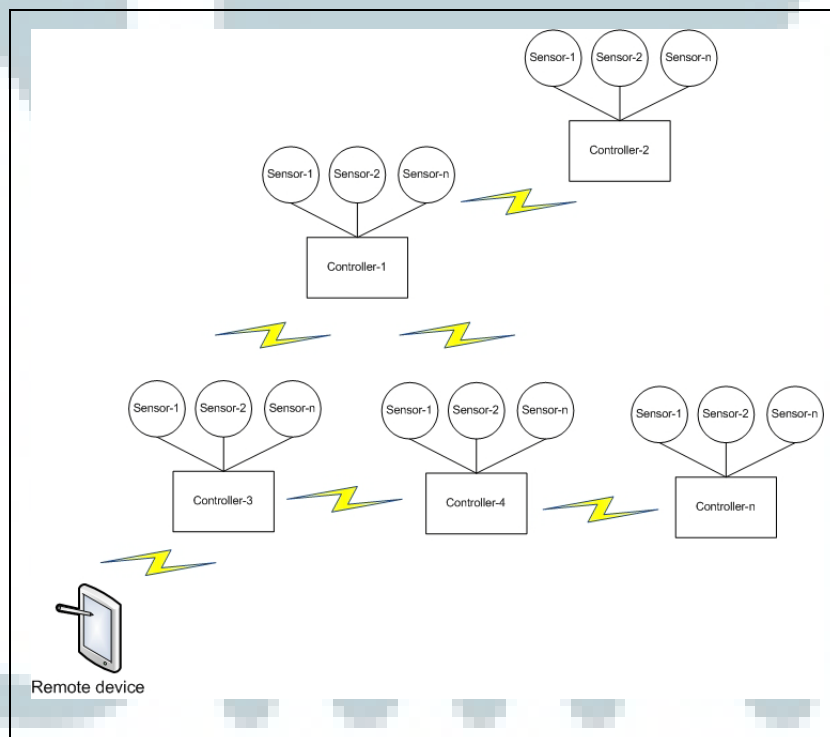
3.1. Studi Literatur

Penelitian ini diawali dengan studi literatur untuk mengkaji lebih dalam metode pemrograman yang akan digunakan pada MINI2440, protokol-protokol yang digunakan pada *ZigBee* agar dapat berkomunikasi dari satu *node* ke *node* yang lain, serta penerapannya pada *building management system*.

3.2. Perancangan Aplikasi

3.2.1. Diagram Umum

Diagram *Building Management System* (BMS) yang digunakan pada penelitian (secara umum) dapat dilihat pada gambar 3.1.



Gambar 3.1 Gambaran umum sistem BMS (adaptasi dari (Yanfei, Cheng, Xiaojun, Yunhe, & Chengbo, 2009))

Sebuah kontroler terhubung dengan beberapa sensor dan perangkat lain seperti lampu dengan menggunakan kabel. Antara kontroler yang satu dengan kontroler lainnya dapat saling berkomunikasi dengan menggunakan ZigBee yang terpasang pada kontroler. ZigBee yang terpasang pada kontroler berfungsi sebagai ZigBee *router* yang akan mengatur rute pengiriman paket informasi. *Remote device* (Mini2440) digunakan untuk memantau dan mengendalikan perangkat-perangkat yang terpasang di gedung. Oleh karena itu, agar *remote device* dapat berkomunikasi dengan kontroler-kontroler di gedung, maka perangkat ini juga akan dihubungkan dengan sebuah modul ZigBee.

Komunikasi antar ZigBee menggunakan protokol yang dikenal dengan nama AT-Command. Apabila Mini2440 ingin mengirimkan paket informasi ke kontroler yang berada di luar cakupannya, maka paket informasi tersebut akan dikirim ke kontroler yang ada di dalam cakupannya terlebih dahulu untuk diteruskan ke kontroler tujuan. Hal ini dimungkinkan karena menggunakan topologi jaringan *mesh*.

3.2.2. Fungsionalitas Aplikasi

Fungsionalitas pada aplikasi yang akan dikembangkan pada Mini2440 adalah:

- a. Untuk halaman *login*
 - Untuk identifikasi dan autentikasi pengguna.
- b. *User management* :
 - membuat *user account* baru (*create user*)
 - mengubah *password* dan *user type* (*edit user*)
 - menghapus *user account* (*delete user*)
- c. *Controller management* :
 - Menambahkan kontroler yang dapat diakses oleh *user* (*add controller*)
 - Mengurangi kontroler yang dapat diakses oleh *user* (*delete controller*)

- Menambahkan, mengurangi hak akses pada *user* terhadap perangkat gedung yang terhubung ke kontroler tertentu (*edit privileges*). Hak akses tersebut meliputi:
 - i. Memantau dan mengendalikan lampu
 - ii. Memantau nilai pada sensor suhu
 - iii. Memantau nilai pada sensor kelembaban
 - iv. Memantau nilai pada sensor cahaya
 - v. Memantau status pada PIR (*passive infrared*)
 - vi. Memantau status pada sensor asap
 - vii. Memantau dan mematikan alarm

d. *Monitoring & control* :

- Memantau dan mengendalikan keadaan lampu;
- Memantau nilai suhu, kelembaban cahaya suatu ruangan;
- Memantau tingkat penerangan dalam suatu ruangan
- Memantau sensor pergerakan (PIR) dan sensor asap;
- Memantau dan mematikan *alarm* yang sedang menyala.

e. *Maintenance*

- *Network checking*

Untuk mengecek ZigBee yang tidak aktif dalam suatu jaringan.

3.2.3. Masukan dan Keluaran

Masukan yang dibutuhkan oleh aplikasi adalah:

a. Untuk halaman login

- *Username* dan *password* yang telah terdaftar dalam file DB.txt. File ini memuat semua informasi *user account* berupa *username*, *password*, dan *user type*. Untuk meningkatkan keamanan, maka informasi *password* yang disimpan ke dalam file DB.txt akan diproses terlebih dahulu menggunakan *hash function* dengan algoritma djb2 (York University). Selain mudah diimplementasikan, fungsi ini juga memiliki distribusi *key* yang baik dengan kecepatan proses yang cepat (Harding, 2011).

b. Untuk modul *user management*

- *Create user*

- *Username* : ID yang digunakan oleh *user* ketika menggunakan menggunakan aplikasi.
- *Password* : sebagai kata kunci untuk autentikasi pengguna
- *Confirmation password*: digunakan pada saat *create user* untuk memastikan *password* yang diisi sebelumnya tidak salah.
- *User type* : jenis *user*, terdiri atas *Administrator* dan *operator*. *administrator* dapat mengakses semua fungsi yang ada pada aplikasi, sedangkan *operator* hanya dapat mengakses fungsi yang ditentukan oleh *administrator*.

- *Edit password*

- *Username* dari *account* yang *passwordnya* akan diganti
- *Password*
- *Confirmation password*

- *Edit user type*

- *Username* yang *type-nya* akan diganti
- *User type*

- *Delete user*

- *Username* dari *account* yang akan dihapus.

c. Untuk modul *controller management*

- *Add controller* :

- *Username* dari pengguna yang akan ditambahkan kontroler yang dapat diakses.
- *Extended Unique Identifier* dari ZigBee pada Kontroler
- Nama alias untuk Kontroler. Alias digunakan untuk mempermudah *user* membedakan kontroler satu dengan kontroler lain.

- *Delete controller*

- *Username*

- Alias dari kontroler yang akan dikurangi
- *Edit privileges* :
 - Username
 - Jenis perangkat yang bisa dan tidak bisa diakses.
- d. Untuk modul *monitoring & control* :
 - Memantau dan mengendalikan keadaan lampu:
 - AT-Command yang berisi informasi berupa EUI kontroler pengirim, *packet number*, dan staus Lampu.
 - Tampilan informasi status lampu.
 - Memantau nilai suhu, kelembaban, dan tingkat penerangan pada suatu ruangan:
 - AT-Command yang berisi informasi berupa EUI kontroler pengirim, *packet number*, data nilai suhu, dan kelembaban.
 - Tampilan informasi nilai suhu dan kelembaban.
 - Memantau tingkat penerangan pada suatu ruangan:
 - AT-Command yang berisi informasi berupa EUI kontroler pengirim, *packet number* dan tingkat penerangan.
 - Tampilan informasi tingkat penerangan.
 - Memantau sensor pergerakan (PIR) dan sensor asap:
 - AT-Command yang berisi informasi berupa EUI kontroler pengirim, *packet number*, status PIR, dan status sensor asap.
 - Tampilan informasi status PIR dan sensor asap.
 - Memantau dan mematikan alarm
 - AT-Command yang berisi informasi berupa EUI kontroler pengirim, *packet number*, dan status *alarm*.
 - Tampilan informasi status alarm.
- e. Untuk modul *maintenance* :

- Daftar ZigBee yang digunakan dalam suatu topologi jaringan dan disimpan dalam file ZigBee.txt.

Keluaran dari aplikasi berupa:

- Untuk halaman *login*
 - Akses ke menu utama (*main menu*)
- Untuk modul *user management* :
 - File DB.txt yang menyimpan informasi semua informasi *username, password, dan type user.*
- Untuk modul *controller management* :
 - Sebuah file (.txt) dengan nama sesuai dengan *username* pengguna (selanjutnya file ini akan disebut sebagai *userX.txt*). File *userX.txt* menyimpan informasi *EUI, alias, dan privileges* yang dapat diakses oleh masing-masing *user.*
- Untuk modul *monitoring & control* :
 - Memantau dan mengendalikan keadaan lampu:
 - AT-Command yang berisi *EUI* kontroler tujuan, *packet number*, dan perintah *request* untuk dikirimkan status lampu.
 - AT-Command yang berisi *EUI* kontroler tujuan, *packet number*, dan perintah pengontrolan lampu.
 - Memantau nilai suhu dan kelembaban, pada suatu ruangan:
 - AT-Command yang berisi *EUI* kontroler tujuan, *packet number*, dan perintah *request* untuk dikirimkan nilai suhu dan tingkat kelembaban.
 - Memantau tingkat penerangan pada suatu ruangan:
 - AT-Command yang berisi *EUI* kontroler tujuan, *packet number*, dan perintah *request* untuk dikirimkan nilai tingkat penerangan.
 - Memantau sensor pergerakan (PIR) dan sensor asap:

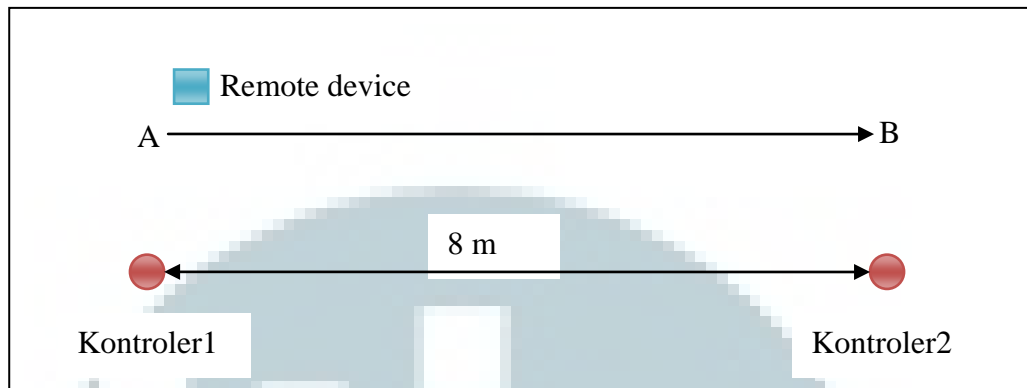
- AT-Command yang berisi EUI kontroler tujuan, *packet* number, dan perintah *request* untuk dikirimkan status PIR dan sensor asap.
- Memantau dan Mematikan *alarm*
 - AT-Command yang berisi EUI kontroler tujuan, *packet* number, dan perintah *request* untuk dikirimkan status *alarm*.
 - AT-Command yang berisi EUI kontroler tujuan, *packet* number, dan perintah untuk mematikan *alarm*.
- e. Untuk modul *maintenance* :
 - AT-Command untuk melakukan *network scanning*.
 - Jumlah dan daftar ZigBee yang diperkirakan tidak aktif.

3.3. Penelitian Awal

Penelitian ini dilakukan untuk memperoleh sejumlah informasi yang akan digunakan untuk saat mendesain modul aplikasi.

3.3.1. Penelitian ZigBee Handover

Penelitian ini dilakukan untuk mencari nilai indikator ZigBee (berdasarkan jumlah NACK) yang dibutuhkan oleh *remote device* selama mekanisme *handover*. Sebagai lingkungan kontrol, kuat sinyal ZigBee yang terhubung ke *remote device* terlebih dahulu diatur menjadi 20dBm. Dengan kuat sinyal seperti itu, maka jarak maksimal transmisi yang dapat dijangkau oleh ZigBee pada *remote device* adalah 5 m (Hartono, 2011). Jarak antara dua kontroler yang masing-masing terhubung dengan ZigBee router diatur menjadi 8m (Gambar 3.2).



Gambar 3.2 Percobaan ZigBee handover

Dengan pengaturan seperti di atas, apabila ZigBee pada *remote device* dipindahkan dari posisi A ke posisi B, maka ZigBee tersebut akan mengalami *handover*. Penelitian dilakukan dengan cara *polling* informasi status lampu sambil melakukan perpindahan dari posisi A ke posisi B. Proses *polling* akan berhenti apabila jumlah NACK yang diatur sama dengan jumlah NACK yang diterima. Apabila proses *polling* berhenti meskipun *remote device* telah berada pada posisi B, maka penelitian akan diulangi lagi dengan menaikkan nilai indikator NACK. Penelitian terus diulangi hingga proses *polling* berhasil dilakukan meskipun *remote device* dipindahkan dari posisi A ke posisi B. Tabel 3.1 menunjukkan hasil percobaan yang dilakukan.

Tabel 3.1 Hasil percobaan *handover*

Percobaan ke	Jumlah pengaturan NACK	Status
1	1	Tidak Berhasil
2	2	Tidak Berhasil
3	3	Tidak Berhasil
4	4	Berhasil

Berdasarkan Tabel 3.1, jumlah NACK yang dapat digunakan mengindikasikan bahwa ZigBee masih berada dalam mekanisme *handover* adalah sebanyak 4 kali.

3.3.2. Penelitian Waktu *Network Scanning*

Penelitian ini bertujuan untuk mengetahui berapa lama waktu maksimal yang dibutuhkan oleh ZigBee pada saat melakukan *scan network*. Penelitian dilakukan dengan cara melakukan *scan network* sebanyak 10 kali pada ZigBee yang terhubung dengan laptop. Akses ke ZigBee pada *laptop* menggunakan program Terminal v1.9 (Com Port Development Tool). Berikut adalah hasil penelitian waktu *network scanning*:

Tabel 3.2 Hasil penelitian waktu *network scanning*

Percobaan ke	Lama Waktu <i>Scan Network</i> (detik)
1	8.5
2	9.4
3	7.1
4	7.0
5	6.6
6	7.2
7	9.3
8	9.8
9	7.3
10	7.8

Berdasarkan Tabel 3.2 dapat diketahui bahwa waktu maksimal yang dibutuhkan untuk *network scanning* adalah 9.8 detik.

3.4. Desain Aplikasi

Sistem akan dibuat sebagai sebuah aplikasi yang terdiri atas beberapa modul sebagai berikut:

1. Modul *Login*
2. Modul *user management*
3. Modul *controller management*

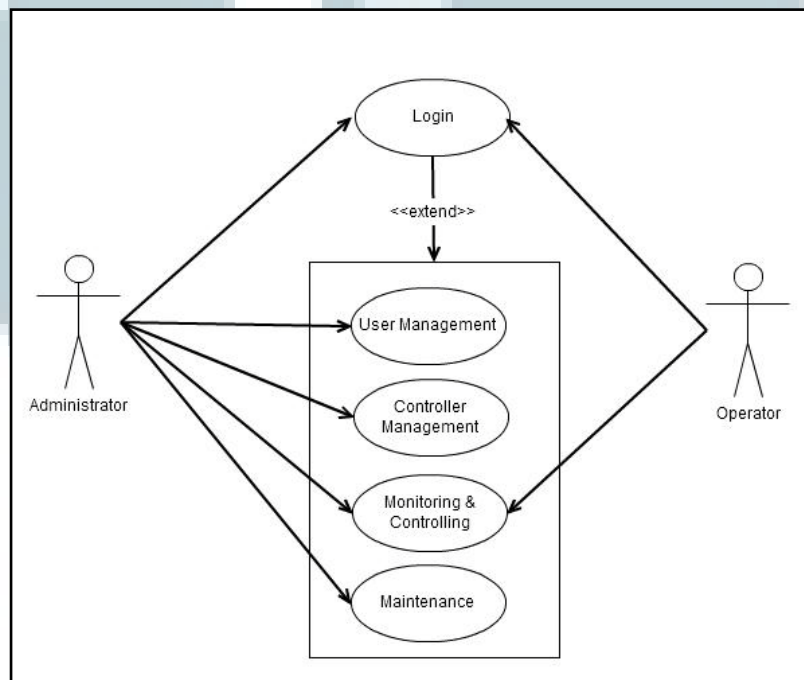
4. Modul *monitoring & controlling*

5. Modul *maintenance*

Desain sistem akan digambarkan mulai dari *use case*, desain modul, hingga perancangan antar muka. Masing-masing perancangan tersebut mewakili penjelasan aktivitas, arus interaksi, dan proses yang nantinya akan berjalan pada aplikasi yang akan dikembangkan.

3.4.1. Use Case

Use case dari aplikasi yang akan dikembangkan dapat dilihat pada Gambar 3.3 yang merupakan keseluruhan aktivitas yang dapat dilakukan oleh *user*.

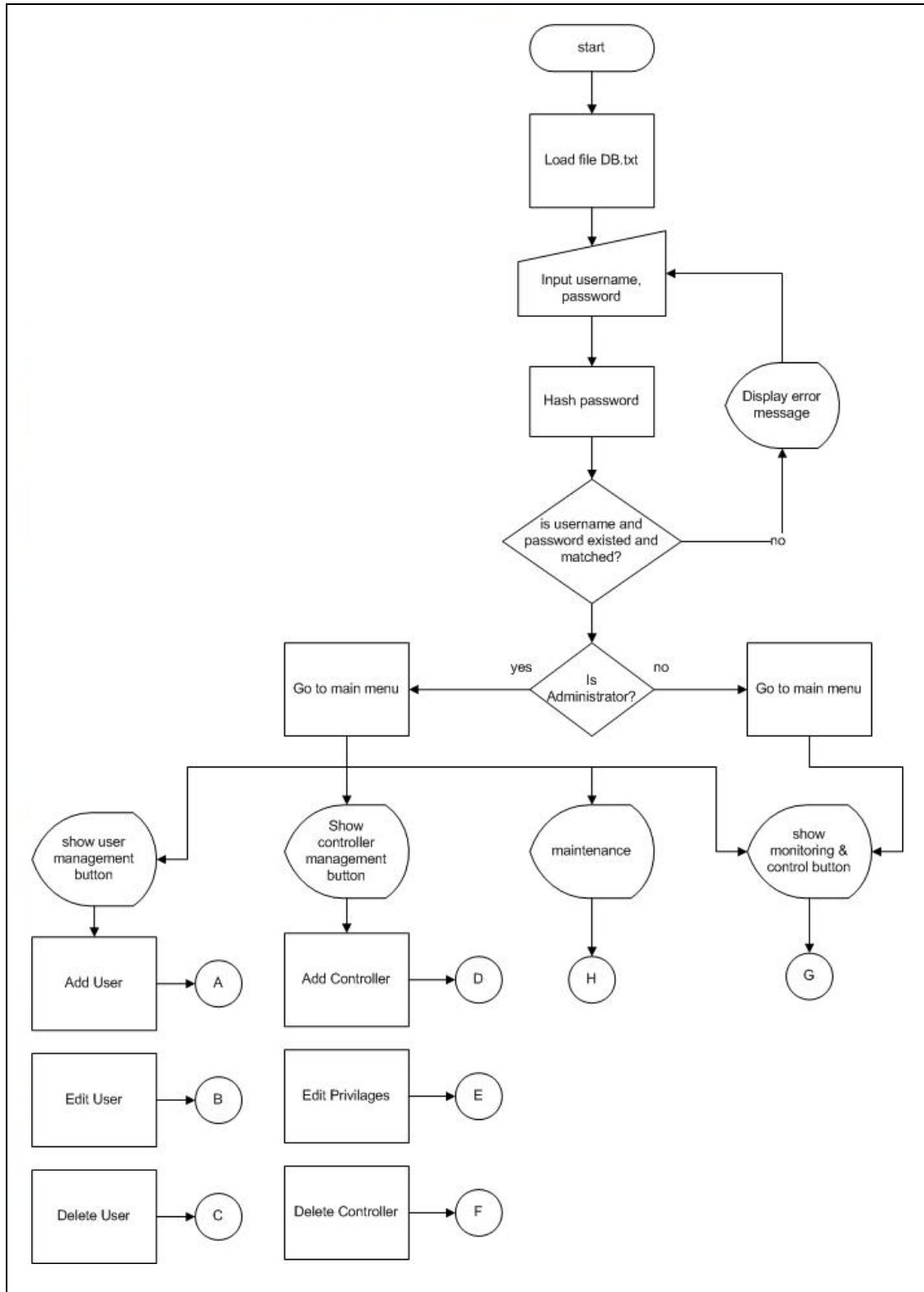


Gambar 3.3 Use Case Aplikasi

Terdapat dua tipe *user* yang akan diterapkan di dalam aplikasi, yaitu administrator dan operator. Administrator dapat mengakses semua fitur yang ada di dalam aplikasi termasuk melakukan pengaturan, sedangkan operator hanya dapat mengakses fitur untuk memantau dan mengendalikan perangkat elektronik saja.

3.4.2. Desain Modul

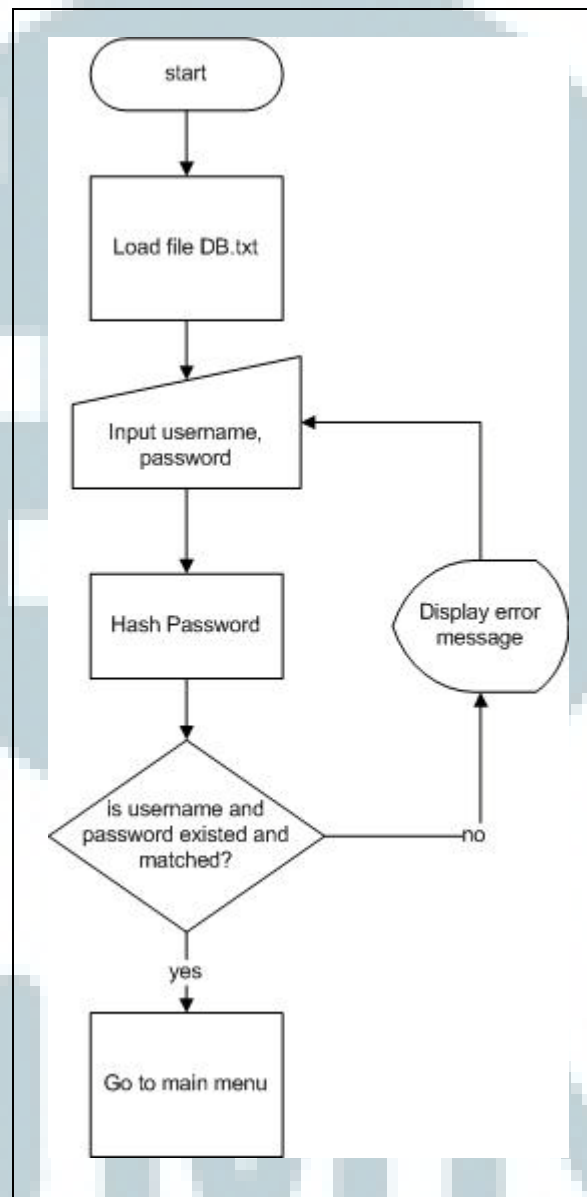
Desain keseluruhan modul pada aplikasi dapat dilihat pada Gambar 3.4.



Gambar 3.4 Flowchart Aplikasi

a. Modul *login*

Modul ini berfungsi untuk identifikasi dan autentikasi *user*. *Login* hanya akan berhasil jika *username* dan *password* yang dimasukan telah terdaftar (Gambar3.5).



Gambar 3.5 Flowchart proses *login*

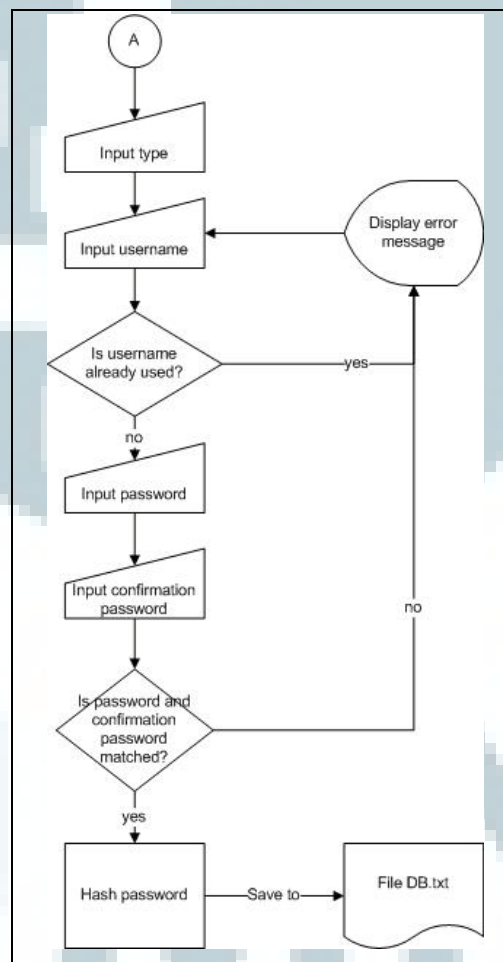
b. Modul *user management*

modul *user management* hanya dapat diakses oleh pengguna dengan tipe administrator. Modul ini terbagi atas 3 sub-modul, yaitu:

i. Modul *create user*

Modul ini berfungsi untuk membuat *user account* yang dapat menggunakan aplikasi. *user account* akan berhasil ditambahkan jika

Username yang dimasukan belum terdaftar, dan *password* yang dimasukan sama dengan *confirmation password*. Informasi tersebut selanjutnya akan disimpan ke dalam file DB.txt (Gambar 3.6).

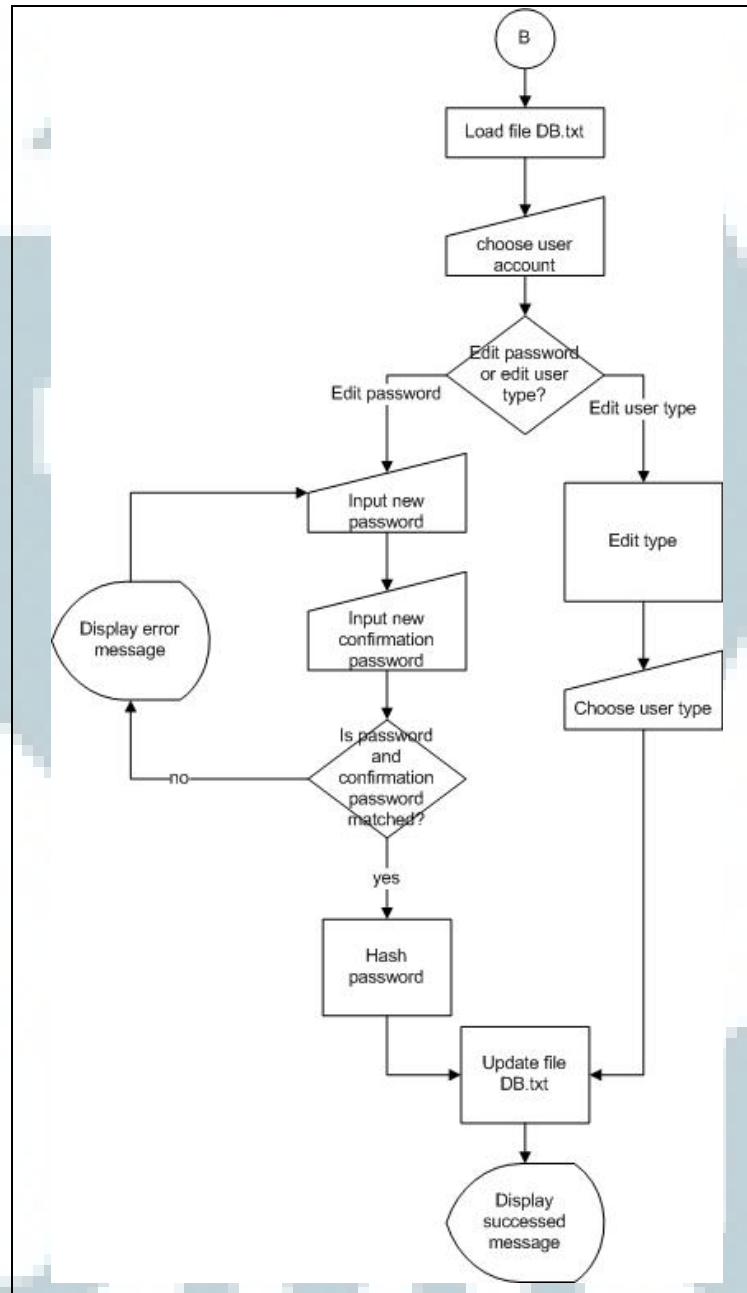


Gambar 3.6 Flowchart proses *create user*

ii. Modul *edit user*

Modul ini berfungsi untuk mengubah *Password* atau *user type* pada *user account*. Proses mengubah *password* akan berhasil jika *password* yang dimasukan cocok dengan *confirmation password*

yang dimasukan. File DB.txt kemudian akan diperbarui sesuai informasi yang baru saja diubah (Gambar 3.7).

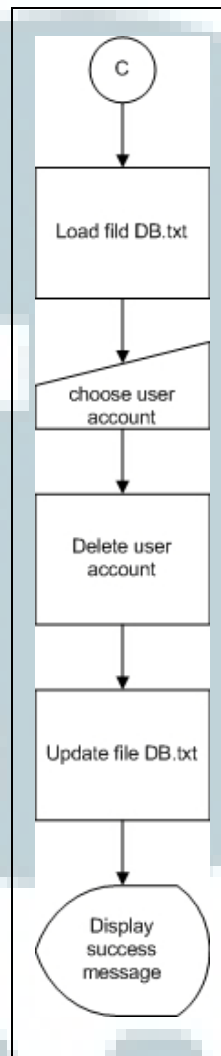


Gambar 3.7 Flowchart proses *edit password & edit type user*

iii. Modul *delete user*

Modul ini berfungsi untuk mengurangi *user account* yang telah terdaftar. *User* akan diminta untuk memilih *user account* yang

akan dihapus. Informasi mengenai *user* tersebut kemudian akan dihapus dengan memperbarui DB.txt (Gambar 3.8).



Gambar 3.8 Flowchart proses delete user account

c. Modul *controller management*

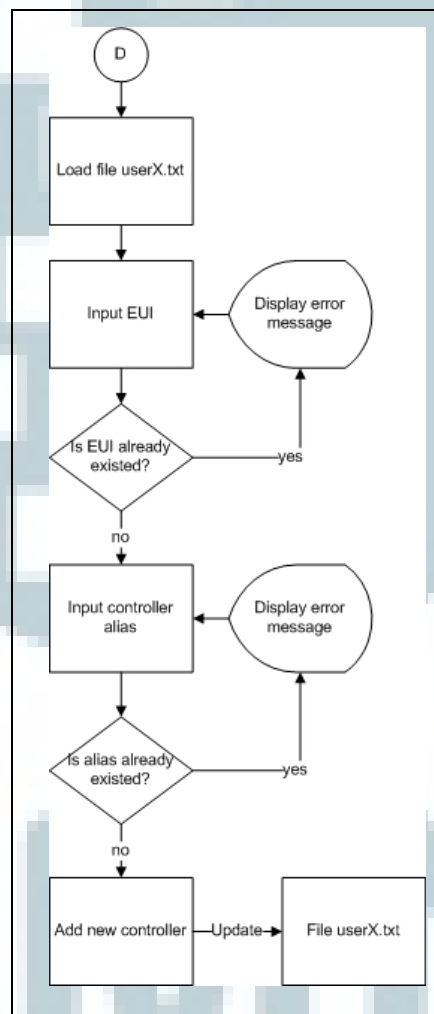
Modul *controller management* hanya dapat diakses oleh pengguna dengan tipe Administrator. Modul ini memiliki tiga sub modul, yaitu:

i. *Add controller*

Karena setiap kontroler terhubung dengan sebuah ZigBee yang memiliki sebuah EUI (*extended unique identifier*), maka EUI akan

dibutuhkan sebagai alamat tujuan ketika akan mengirimkan paket informasi ke suatu kontroler.

Sebuah kontroler akan berhasil ditambahkan jika EUI yang dimasukan belum terdaftar pada sebuah file userX.txt, dan alias yang dimasukan juga belum terdaftar. File userX.txt kemudian akan diperbarui dengan menambahkan informasi di atas. (Gambar 3.9).

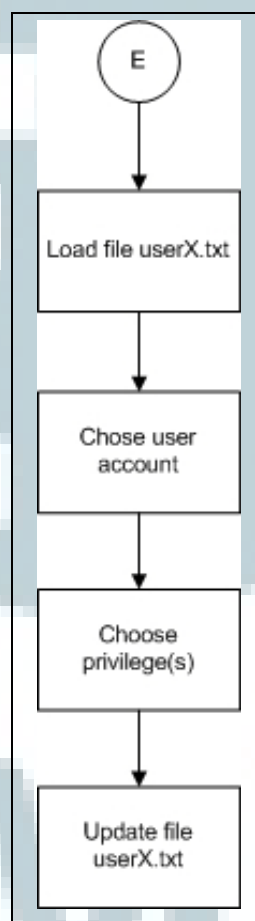


Gambar 3.9 Flowchart proses *add controller*

ii. *Edit user privileges*

Sebuah kontroler terhubung dengan beberapa perangkat elektronik seperti lampu, sensor, dan alarm. Meskipun seorang *user* telah memiliki hak akses terhadap suatu kontroler, belum tentu ia dapat

mengakses setiap perangkat elektronik yang terhubung dengan kontroler tersebut. Untuk dapat mengakses setiap perangkat yang terhubung dengan kontroler tersebut maka administrator harus terlebih dahulu mengatur apa saja yang dapat diakses oleh seorang *user* terhadap suatu kontroler (Gambar 3.10). Informasi mengenai hak akses terhadap perangkat apa saja akan disimpan pada file *userX.txt*.

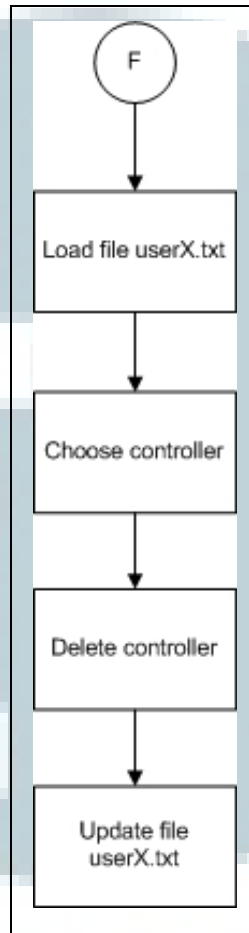


Gambar 3.10 Flowchart proses *edit user privileges*

iii. *Delete controller*

Administrator dapat juga mengurangi kontroler-kontroler yang sebelumnya dapat diakses oleh *user*. untuk dapat mengurangi suatu kontroler maka dibutuhkan informasi kontroler mana yang akan

dikurangi. Setelah itu file userX.txt akan diperbarui dengan mengurangi informasi tentang kontroler yang bersangkutan (Gambar 3.11).



Gambar 3.11 Flowchart proses *delete controller*

c. Modul *monitoring & controlling*:

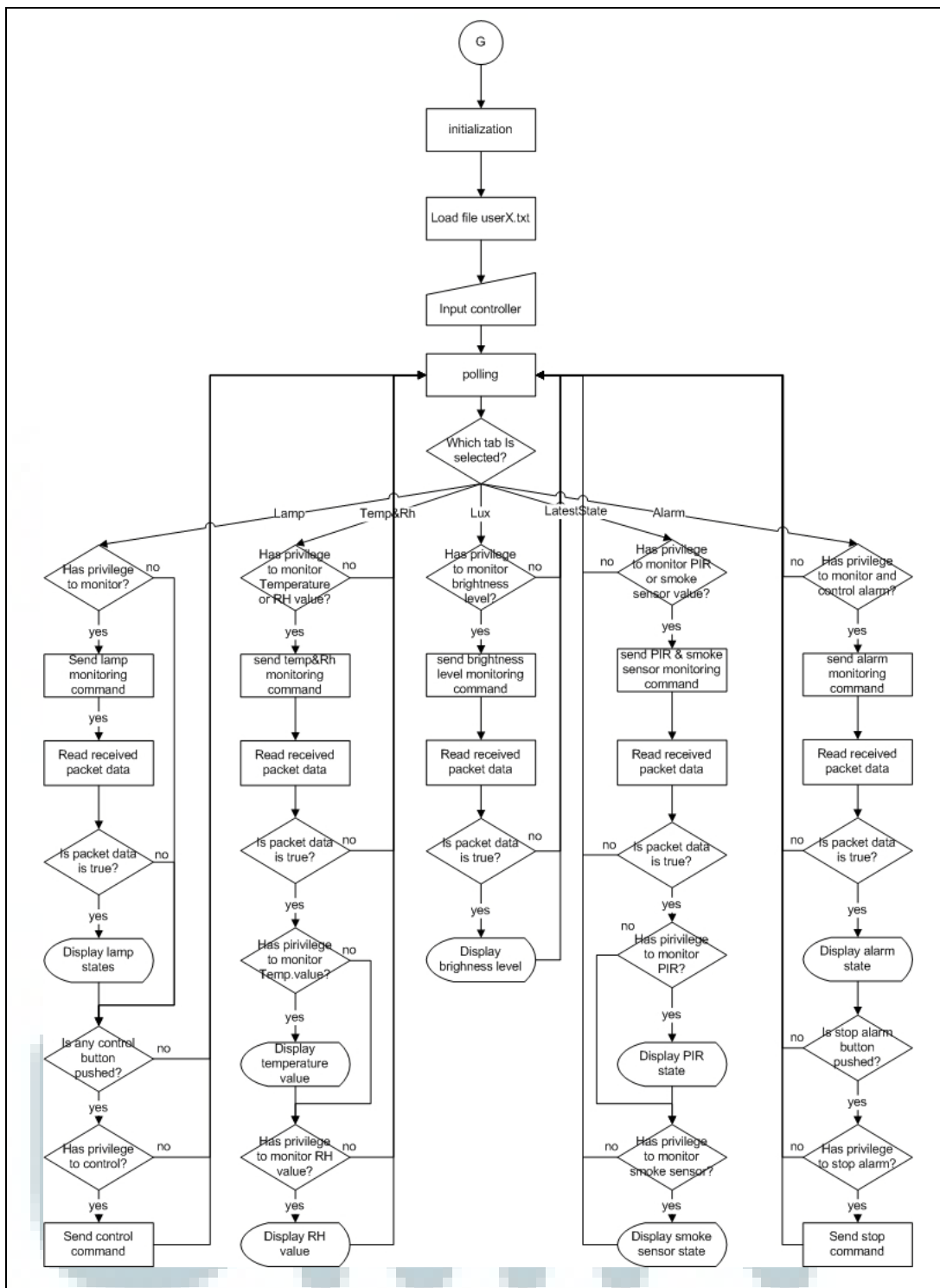
Modul ini berfungsi untuk:

- memantau dan mengendalikan lampu
Kondisi setiap lampu akan ditampilkan pada *buttons*. Warna hitam pada sebuah *button* menunjukkan bahwa lampu tersebut dalam keadaan mati. Warna kuning menunjukkan bahwa lampu yang bersangkutan dalam keadaan menyala. Jika warna *button* adalah putih, maka *user* yang bersangkutan tidak dapat melakukan *control* pada lampu.

- Memantau nilai pada sensor suhu
Nilai suhu akan ditampilkan dalam derajat Celcius ($^{\circ}\text{C}$)
- Memantau nilai pada sensor kelembaban
Menampilkan nilai persentase jumlah air dalam udara (%)
- Memantau nilai pada sensor cahaya
- Memantau status pada PIR (*passive infrared*)
Sensor PIR akan aktif jika mendeksi adanya suatu pergerakan.
- Memantau status pada sensor asap
- Memantau dan mematikan alarm.
Kondisi alarm *on* jika alarm dalam keadaan aktif. Sebuah *button* disediakan untuk mematikan alarm yang sedang aktif (menyala).

User hanya dapat memantau dan mengendalikan perangkat yang terhubung pada suatu kontroler apabila telah mendapat hak akses(*privileges*) oleh administrator (Gambar 3.12).

UMMN



Gambar 3.12 Flowchart proses *monitoring dan controlling*

Ketika aplikasi menerima packet data yang dikirim dari kontroler tujuan, maka data tersebut akan diperiksa terlebih dahulu untuk autentikasi.

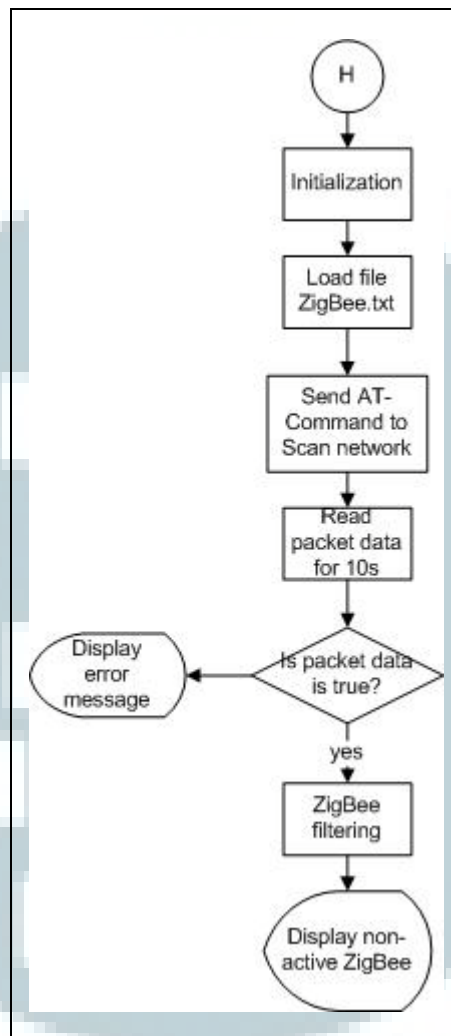
Informasi yang diperiksa adalah *packet number* dan EUI pengirim. Informasi *packet number* dan EUI pengirim harus sesuai dengan informasi yang diminta oleh aplikasi pada *remote device* agar dapat diproses dan ditampilkan.

Koneksi akan dinyatakan terputus (*connection lost*) apabila aplikasi tidak menerima jawaban dari kontroller tujuan (NACK) sebanyak 4 kali. Nilai ini diperoleh berdasarkan percobaan yang dilakukan sebelumnya (lihat 3.3.1 Penelitian ZigBee *Handover*) untuk mengetahui jumlah NACK yang diterima oleh ZigBee pada saat proses *handover*.

d. Modul *maintenance*:

- *Network checking*

Modul ini berfungsi untuk mengetahui alamat (EUI) dari ZigBee yang telah dipasang dalam suatu topologi jaringan dan dalam kondisi tidak aktif. Metode yang digunakan yaitu mendaftarkan ZigBee yang digunakan dalam jaringan terlebih dahulu (disimpan di dalam file *ZigBee.txt*), kemudian melakukan *network scanning*. Hasil keluaran dari *network scanning* berupa daftar ZigBee yang terdapat dalam suatu jaringan. Hasil keluaran ini kemudian akan dibandingkan dengan daftar yang telah dibuat sebelumnya. Apabila suatu ZigBee telah terdaftar dan tidak ditemukan pada keluaran *network scanning* maka ZigBee tersebut akan diperkirakan tidak berada dalam satu jaringan yang sama atau ZigBee tersebut tidak aktif (Gambar 3.13).

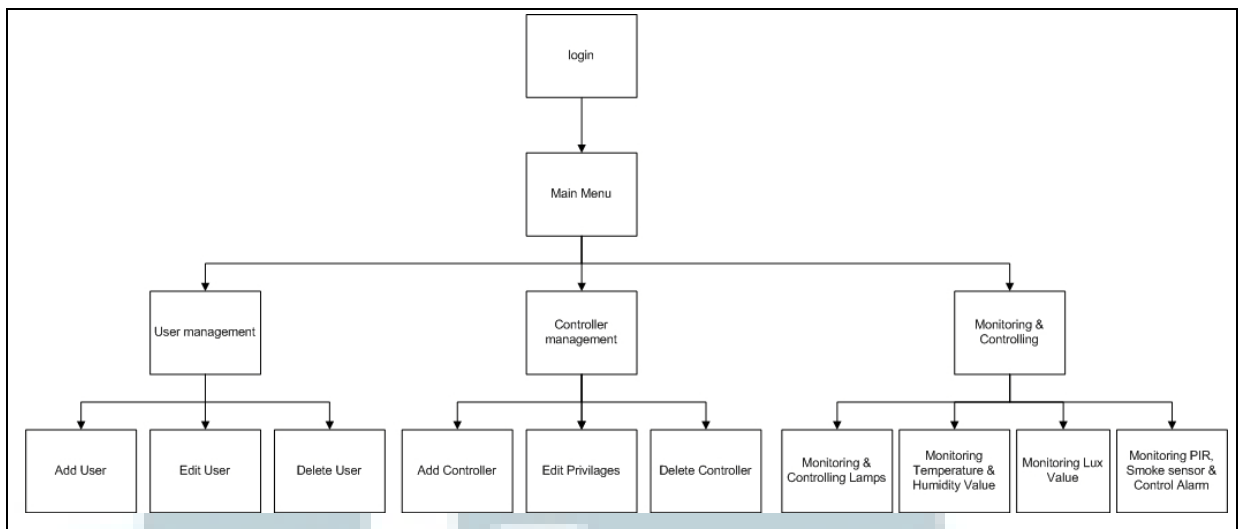


Gambar 3.13 Flowchart proses *network checking*

Lama waktu yang didesain untuk membaca informasi pada proses *network scanning* adalah 10 detik. Nilai tersebut dipilih diterapkan berdasarkan percobaan yang dilakukan untuk mengukur lama waktu yang diperlukan untuk proses *network scanning* (lihat 3.3.2 Penelitian waktu *Network Scanning*).

3.4.3. Hirarki Menu

Berikut merupakan hirarki menu dari aplikasi:



Gambar 3.14 Hirarki menu

Setelah melakukan *login*, *user* akan dihadapkan pada tampilan *main menu* yang berisi *link* ke menu *user management*, *controller management*, dan *monitoring & controlling*. Menu *user management* akan membuka halaman untuk pengaturan terhadap *user account*, seperti membuat *user account* baru, mengubah *password* atau *user type* dari *user account* yang sudah ada, dan menghapus *user account* yang sudah ada.

Menu *controller management* akan membuka halaman untuk melakukan pengaturan terhadap kontroler yang akan diakses, seperti menambahkan kontroler yang dapat diakses oleh *user*, mengatur hak akses(*privileges*) *user* terhadap perangkat-perangkat yang terhubung ke kontroler, dan mengurangi kontroler yang dapat diakses oleh *user*.

Menu *monitoring & controlling* akan membuka halaman untuk melakukan pemantauan dan pengendalian perangkat-perangkat yang terhubung dengan kontroler seperti lampu, sensor suhu, sensor kelembaban, sensor cahaya, PIR, sensor asap, dan alarm.

3.5. Desain Antarmuka

Agar fungsi aplikasi dapat ditampilkan secara maksimal pada layar Mini2440 yang berukuran kecil, maka desain antarmuka pada aplikasi akan didesain dengan tampilan yang sederhana dan *userfriendly*. Berikut adalah

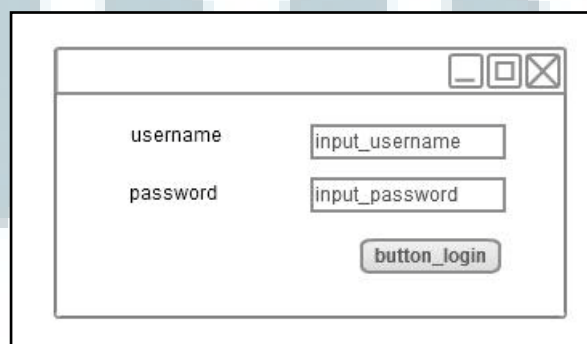
beberapa hal yang diterapkan dalam desain antarmuka agar dapat memaksimalkan penggunaan aplikasi pada layar minimalis dari Friendly ARM Mini2440:

- Peletakkan komponen *interfaces* pada setengah ukuran layar untuk halaman yang membutuhkan *input* dari virtual keyboard. Desain ini bertujuan agar virtual keyboard tidak menutupi komponen *interface* tersebut.
- Penggunaan komponen *tabs* untuk menampilkan beberapa modul dalam satu halaman.
- Penggunaan *combo box* dan mekanisme pengisian data secara berurutan (contoh: pada saat membuat *user account* baru) untuk menghemat *space* yang digunakan.
- Selain penggunaan komponen *Button(s)* pada modul *monitoring & controlling* untuk mengontrol lampu, warna pada setiap *button* didesain agar dapat berubah sesuai dengan status dari setiap lampu di dalam ruangan.

Berikut adalah desain antarmuka untuk aplikasi *remote monitoring* dan *controlling* yang akan dikembangkan pada Mini2440:

3.5.1. Desain tampilan halaman *login*

Pada halaman *login*, pengguna akan diminta untuk mengisi *input* berupa *username* dan *password*. Tombol *login* disediakan agar pengguna dapat melakukan *login* berdasarkan *input* yang dimasukkan (Gambar 3.15).

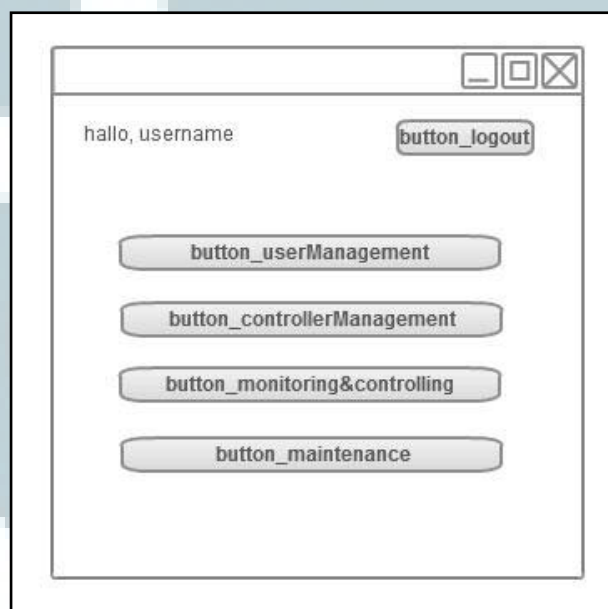


The image shows a graphical user interface for a login page. It consists of a rectangular window with a title bar at the top containing three control icons: a square, a square with a horizontal line, and a square with an 'X'. Inside the window, there are two labels on the left: 'username' and 'password'. To the right of 'username' is a rectangular input field labeled 'input_username'. To the right of 'password' is a rectangular input field labeled 'input_password'. Below these two input fields is a rounded rectangular button labeled 'button_login'.

Gambar 3.15 Desain antarmuka pada *halaman login*

3.5.2. Desain tampilan halaman *main menu*

Antarmuka pada halaman *main menu* menggunakan sebuah label dan lima button (Gambar 3.16). label digunakan untuk informasi pengguna yang sedang *login*, empat button untuk mengakses fitur-fitur pada aplikasi yaitu: *user management*, *controller management*, *monitoring & controlling*, dan *network checking*. Sebuah *button logout* juga disediakan dan dapat digunakan jika pengguna ingin melakukan *login* dengan *user account* yang lain.



Gambar 3.16 Antarmuka halaman *main menu*

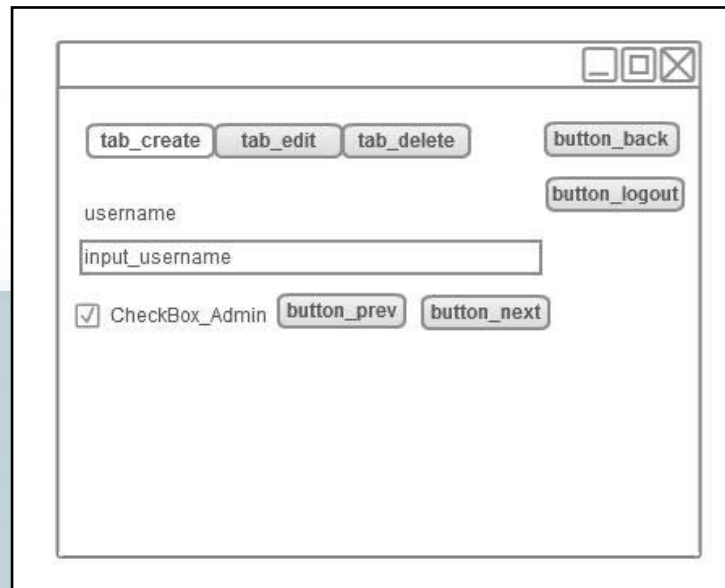
3.5.3. Desain tampilan halaman *user management*

Antarmuka pada halaman *user management* didesain menggunakan *tabs* untuk menampilkan modul *create user*, *edit user*, *delete user* dalam satu halaman.

a. *Create user*

Antarmuka untuk modul *create user* didesain menggunakan sebuah *line edit* dan sebuah *check box* (Gambar 3.17). *Line edit* digunakan untuk menerima input berupa *username*, *password*, dan *confirmation password* secara berurutan. *Button prev* dan *next* digunakan untuk menuju ke tahap sebelum atau sesudahnya. *Check box* merupakan masukkan untuk *user type* yang akan dibuat

(tipe Administrator jika ditandai, dan tipe Operator jika tidak ditandai).

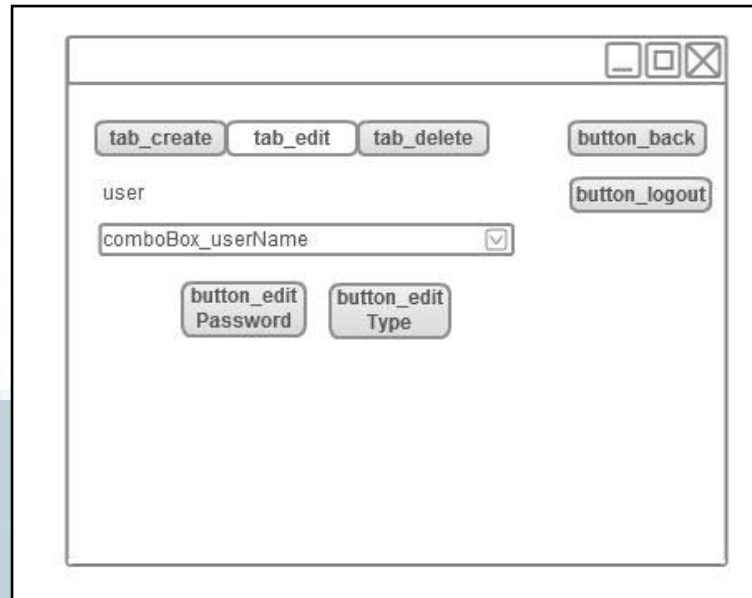


Gambar 3.17 Desain antarmuka *create user*

Button back digunakan untuk kembali ke *main menu*, sedangkan *button logout* dapat digunakan jika pengguna ingin *login* dengan *user account* yang lain (*button back* dan *logout* dengan fungsi seperti yang dijelaskan di atas akan digunakan pada beberapa halaman aplikasi lainnya dan tidak akan dijelaskan fungsinya lagi).

b. *Edit user*

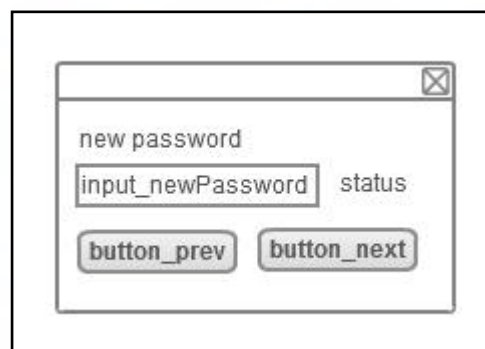
Antarmuka pada modul *edit user* didesain menggunakan sebuah *combo box*, dan dua buah *button* (Gambar 3.18). *Combo box* digunakan agar Administrator dapat dengan mudah memilih *user* yang akan perbaharuinya. *Button editPassword* digunakan untuk mengakses modul *edit password* sedangkan *button editType* digunakan untuk mengakses modul *edit type*.



Gambar 3.18 Desain antarmuka edit user

- *Edit password*

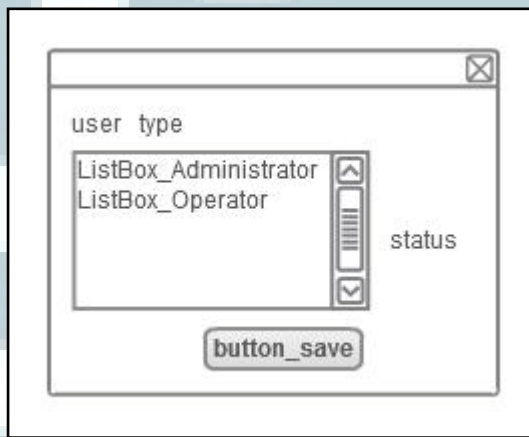
Antarmuka pada modul *edit password* didesain menggunakan sebuah *line edit* dua buah *button*, dan sebuah *label* (Gambar 3.19). *Line edit* digunakan untuk menerima *input* berupa *new password* dan *password confirmation* secara berurutan. *Button prev* dan *next* digunakan untuk menuju ke tahap sebelum atau sesudahnya. Label digunakan untuk menampilkan status berhasil atau tidaknya proses *edit password*.



Gambar 3.19 Desain antarmuka edit password

- *Edit user type*

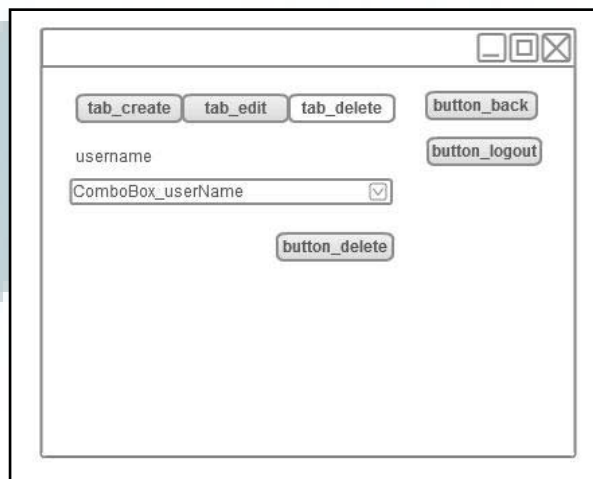
Antarmuka pada modul *edit user type* didesain menggunakan sebuah *List Box*, sebuah *button*, dan sebuah *label* (Gambar 3.20). *List box* digunakan *user type* yang akan diterapkan ke pengguna. *Button save* digunakan untuk menyimpan *user type* yang telah dipilih. Label digunakan untuk menampilkan status berhasil apabila proses *edit user type* berhasil dilakukan.



Gambar 3.20 Desain antarmuka *edit user type*

- c. *Delete user*

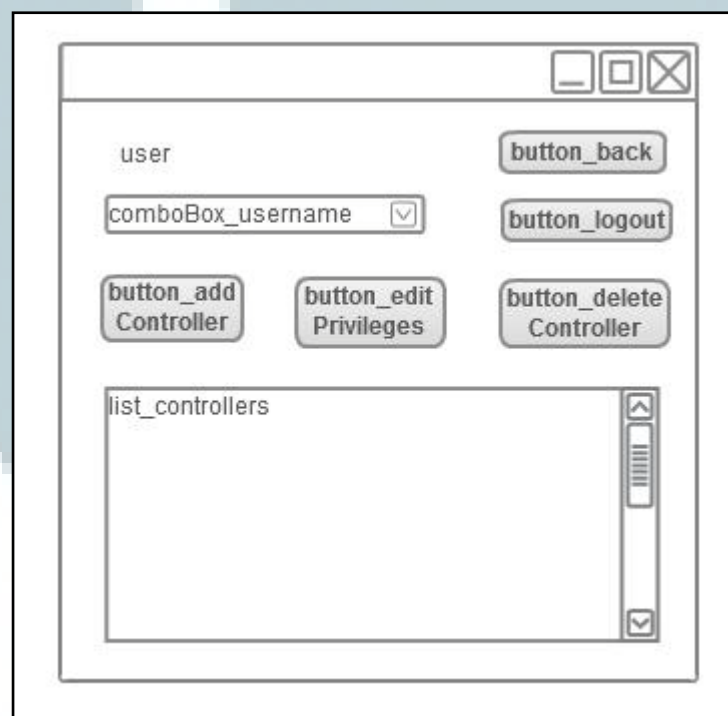
Antarmuka pada modul *delete user* didesain menggunakan sebuah *combo box* dan sebuah *button* (Gambar 3.21). *combo box* digunakan untuk memilih *user* yang akan dihapus, dan *button delete* digunakan untuk menghapus *user* yang telah dipilih.



Gambar 3.21 Desain antarmuka *delete user*

3.5.4. Desain tampilan halaman *controller management*

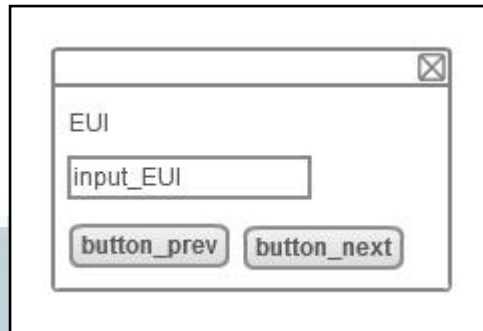
Antarmuka pada halaman *controller management* didesain menggunakan sebuah *combo box*, *tabs*, dan sebuah *text view* (Gambar 3.22). *combo box* digunakan untuk memilih yang akan diperbaharui haknya dalam mengakses kontroler. *tabs* digunakan untuk mengakses modul *add controller*, *edit privilege(s)*, dan *delete controller* dalam satu *window*. *Text view* digunakan untuk menampilkan daftar kontroler yang dapat diakses oleh *user* yang dipilih.



Gambar 3.22 Desain antarmuka *controller management*

a. *Add controller*

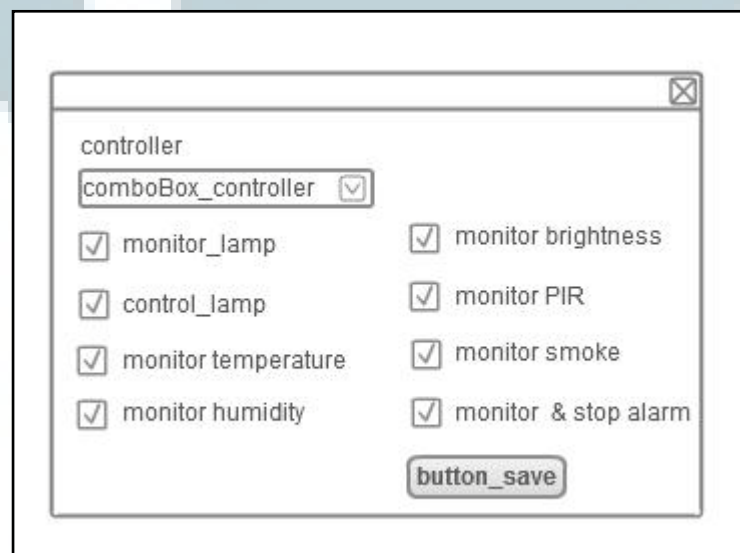
Antarmuka pada *add controller* didesain menggunakan sebuah *line edit*, dan dua buah *button* (Gambar 3.23). *Line edit* digunakan untuk menerima *input* berupa EUI dari ZigBee yang terhubung ke kontroler, dan alias yang digunakan pada kontroler bersangkutan (secara berurutan). *Button prev* dan *next* digunakan untuk menuju ke tahap sebelum atau sesudah dari proses *add controller*.



Gambar 3.23 Desain antarmuka *add controller*

b. *Edit privilege(s)*

Antarmuka pada modul *edit privilege(s)* didesain menggunakan sebuah *combo box*, delapan *check box*, dan sebuah *button* (Gambar 3.24). *Combo box* digunakan untuk memilih kontroler, *combo box* digunakan untuk memberikan hak kepada pengguna terhadap perangkat yang terhubung dengan kontroler. *Button save* digunakan untuk menyimpan pengaturan yang telah dilakukan.

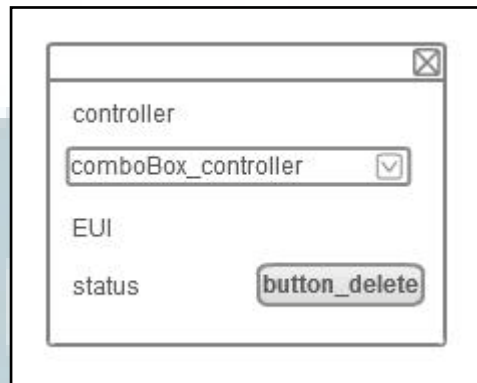


Gambar 3.24 Desain antarmuka *edit privileges*

c. *Delete controller*

Antarmuka pada modul *delete controller* menggunakan sebuah *combo box*, dua *label*, dan sebuah *button* (Gambar 3.25). *Combo box* digunakan untuk memilih kontroler, label EUI untuk

menampilkan EUI dari kontroler yang dipilih, label status menampilkan status apabila proses *delete controller* berhasil.



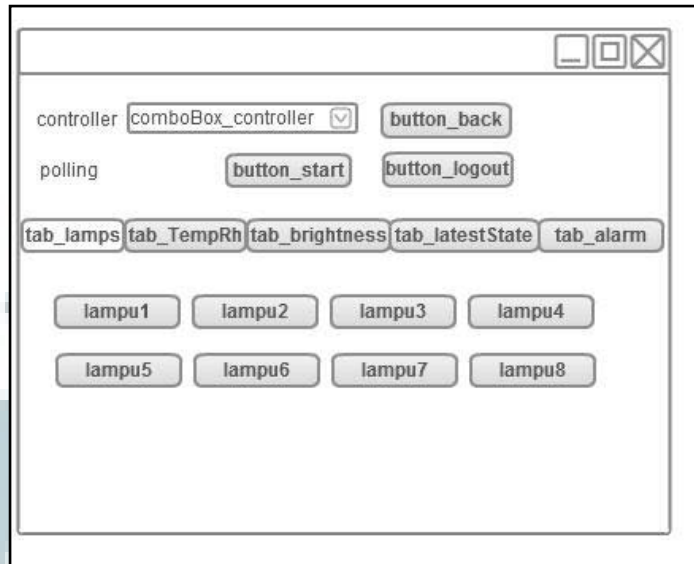
Gambar 3.25 Desain antarmuka *delete controller*

3.5.5. Desain tampilan halaman *monitoring & controlling*

Antarmuka pada halaman *monitoring & controlling* didesain menggunakan sebuah *combo box*, tabs, dan sebuah *button*. *Combo box* digunakan untuk memilih kontroler yang akan dipantau atau dikendalikan. *Button polling* digunakan untuk malakukan pemantauan pada perangkat berdasarkan *tab* yang sedang aktif. *Tabs* digunakan untuk mengakses perangkat lampu, sensor suhu, kelembaban, tingkat penerangan, sensor asap, PIR, dan *alarm* dalam satu *window*.

a. *Lamps monitoring & controlling*

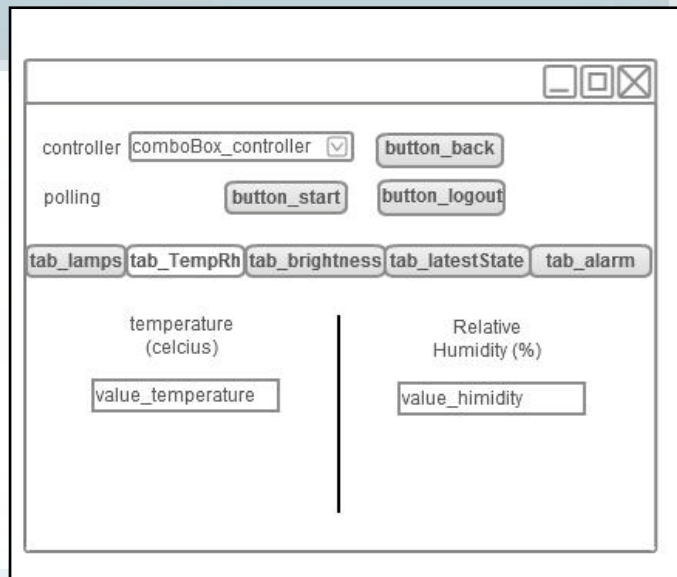
Antarmuka pada modul *lamps monitoring & controlling* didesain menggunakan 8 buah *buttons* yang mewakili 8 kanal lampu (Gambar 3.26). Keadaan lampu (menyala atau mati) akan ditampilkan melalui warna pada *button(s)*. Selain untuk menampilkan keadaan setiap lampu, *button(s)* tersebut juga didesain untuk dapat melakukan *toggle* terhadap kanal lampu yang bersangkutan.



Gambar 3.26 Desain antarmuka *lamps monitoring*

b. *Temperature & humidity monitoring*

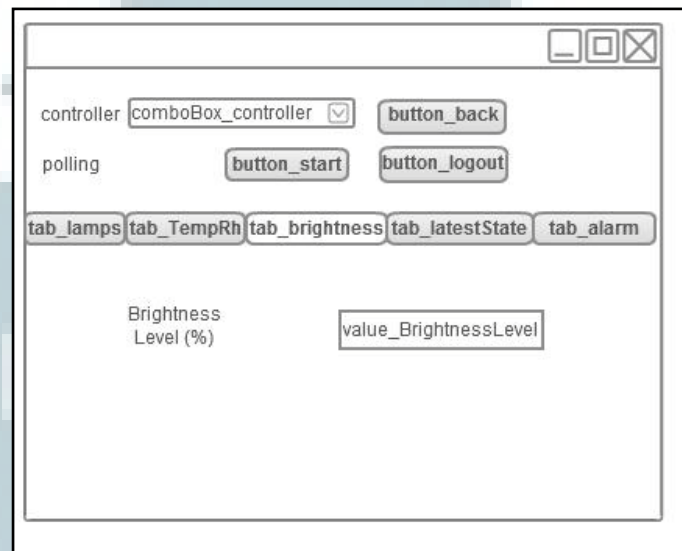
Antarmuka pada modul *temperature & humidity monitoring* didesain menggunakan LCD number untuk menampilkan suhu ($^{\circ}\text{C}$) dan tingkat kelembaban (%) (Gambar 3.27).



Gambar 3.27 Desain antarmuka *temperature & humidity monitoring*

c. *Brightness level monitoring*

Antarmuka pada modul *brightness level monitoring* didesain menggunakan LCD number untuk menampilkan tingkat intensitas cahaya dalam suatu ruangan (%) (Gambar 3.28).

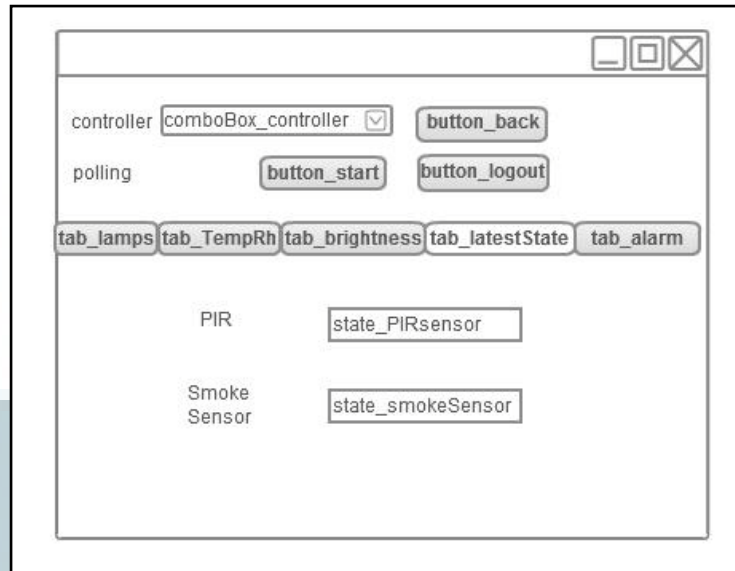


Gambar 3.28 Desain antarmuka *brightness monitoring*

d. *Latest state monitoring (PIR & Smoke Sensor)*

Antarmuka pada modul *latest state monitoring* didesain menggunakan 2 label yang akan menampilkan status dari masing-masing sensor (sensor asap dan PIR) (Gambar 3.29)

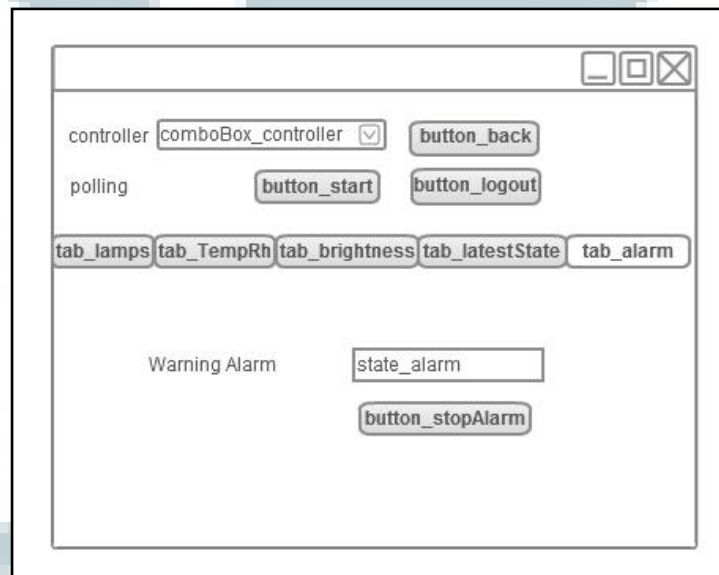
U M N



Gambar 3.29 Desain antarmuka *latest state monitoring*

e. *Alarm monitoring & Controlling*

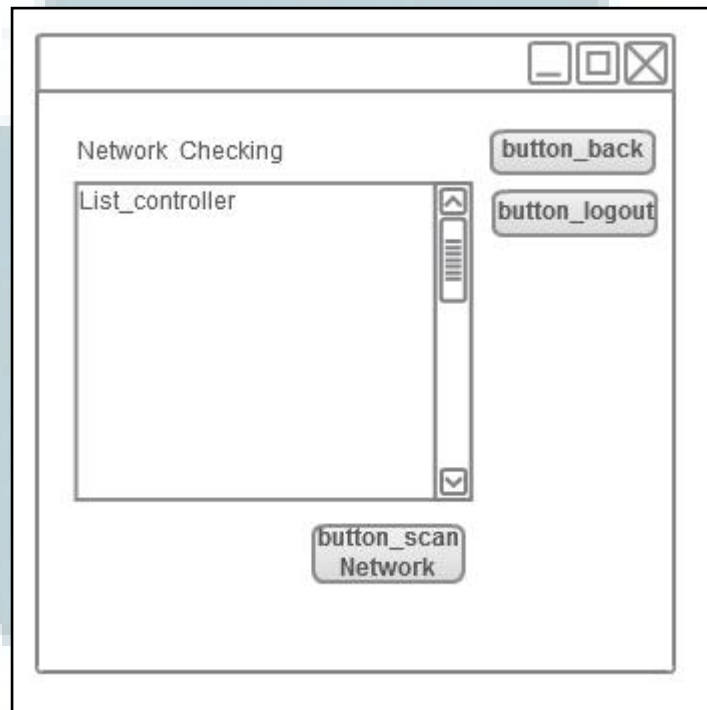
Antarmuka pada modul *alarm monitoring & controlling* didesain menggunakan sebuah label yang akan menampilkan status dari *warning alarm* dan sebuah *button* yang dapat digunakan untuk mematikan *alarm* (Gambar 3.30)



Gambar 3.30 Desain antarmuka *alarm monitoring & controlling*

3.5.6. Desain tampilan halaman *network checking*

Antarmuka pada halaman *network checking* didesain menggunakan sebuah *text view* untuk menampilkan daftar ZigBee yang tidak ditemukan di dalam jaringan (PAN) dan sebuah *button* untuk melakukan *network scanning* (Gambar 3.31).

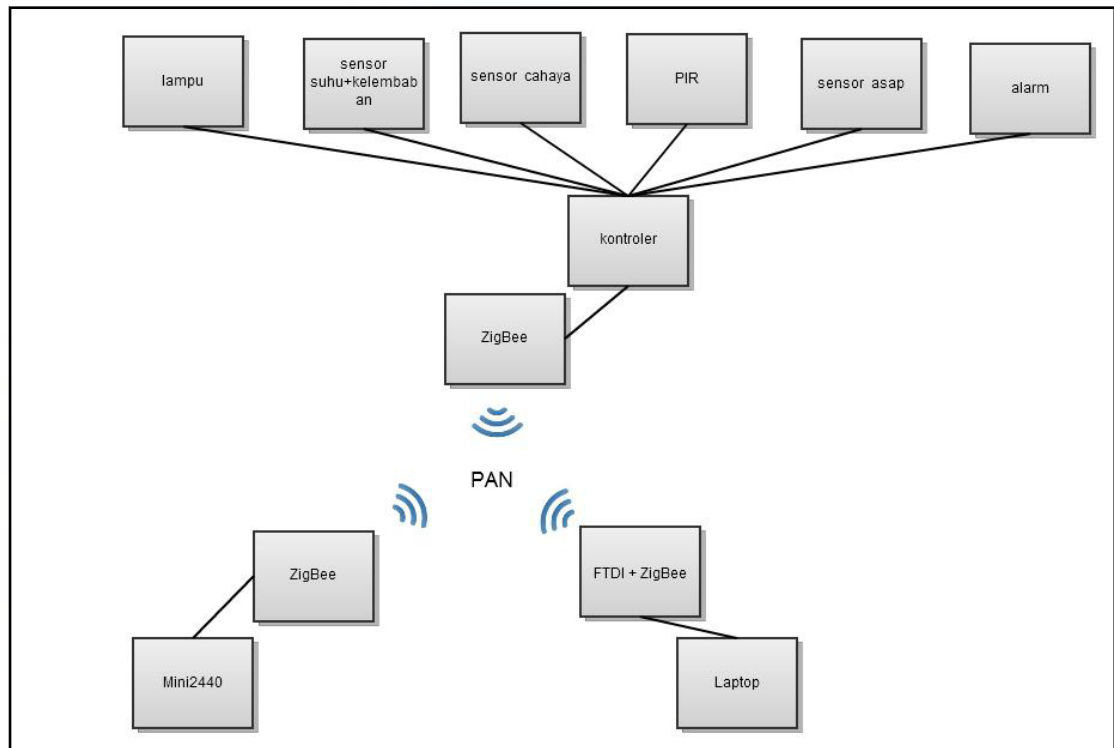


Gambar 3.31 Desain antarmuka *network checking*

3.6. Metode Pengujian

Aplikasi yang dikembangkan pada Friendly ARM Mini2440 akan diuji performanya sebagai *remote device* untuk *building management system*. Pengujian yang akan dilakukan berupa pengujian fungsionalitas aplikasi untuk *user management* (*add user, edit user, delete user*), *controller management* (*add controller, edit privileges, delete controller*), *monitoring & controlling* (keadaan lampu, nilai suhu, tingkat kelembaban, tingkat penerangan, keadaan sensor asap, keadaan PIR, dan status alarm), dan *network checking*. Data dan dokumentasi hasil pengujian akan dikumpulkan untuk kemudian digunakan sebagai bahan pengambilan kesimpulan dari penelitian yang dilakukan.

Bentuk jaringan yang digunakan pada saat penelitian dapat dilihat pada Gambar 3.32.



Gambar 3.32 Topologi jaringan yang digunakan

Pada gambar di atas, PAN dibentuk oleh ZigBee router yang saling terhubung satu sama lain. ZigBee router tersebut terpasang pada setiap kontroler

Proses pengujian memerlukan sejumlah perangkat keras (*hardware*) dan perangkat lunak (*software*).

Berikut adalah Spesifikasi perangkat keras yang digunakan:

a. Laptop Asus M50Vn dengan spesifikasi:

- Processor Intel(R) core(TM) Duo CPU T9400 @2.53GHz (2CPUs), ~2.5GHz
- RAM 4096MB
- Harddisk 300GB

b. Friendly ARM Mini2440, dengan spesifikasi:

- Processor Samsung S3C2440, 405MHz
- RAM 64MB SDRAM

- FLASH *storage* 128MB
 - LCD interface (resistive touch panel interface)
- c. Board Tester yang terhubung dengan:
- *Solid State Relay (SSR) 8 channel*
 - *temperature sensor* (sensor suhu)
 - *Radius humidity sensor* (sensor kelembaban)
 - *Ambient light sensor* (sensor cahaya), ADC (analog to digital converter)
 - *passive infrared (PIR)*
 - *Smoke sensor* (sensor asap)
- d. ZigBee (*hardware: ETRX2, firmware: ETRX3*)
- e. FTDI (Agar ZigBee dapat dihubungkan dan digunakan pada Laptop melalui *USB port*)

Berikut adalah spesifikasi perangkat lunak yang digunakan:

- a. Sistem Operasi Windows 7 - 64bit
- b. PuTTY (untuk *remote* ke Mini2440)
- c. Terminal v1.9b (untuk mengakses ZigBee yang terhubung dengan Laptop)
- d. VMware® Workstation v7.1.0
- e. Sistem Operasi Linux Ubuntu 9.10 - Karmic Koala
- f. QT Designer by Trolltech
- g. Cross compiler ARM-Linux GCC 4.4.3
- h. Qtopia ARM-Qtopia 2.2.0
- i. Text Editor Geany
- j. Terminal
- k. Sistem Operasi Linux FriendlyARM 2.6.32.2-FriendlyARM

3.7. Skenario Pengujian

Penelitian akan dilakukan dengan melakukan sejumlah pengujian dengan menggunakan aplikasi yang telah dikembangkan pada MINI2440 untuk *user management*, *controller management*, dan *monitoring&controlling* terhadap

kontroler (*board tester*) yang terhubung dengan ZigBEE dan terpasang di dalam gedung. Pengujian ini bertujuan untuk mengetahui tingkat keberhasilan dari aplikasi pada Mini2440 untuk *Building Management System*. Berikut adalah skenario pengujian yang akan dilakukan untuk mencapai tujuan dari penelitian:

3.7.1. Pengujian fungsi *login page*

Pengujian *login* dilakukan dengan memasukkan input *username* dan *password* secara bervariasi (terdaftar, tidak terdaftar, tidak diisi). Data yang terdaftar pada file DB.txt hanya *Username: "admin"* dan *password: "admin"*. Tabel 3.3 menampilkan variasi *username* dan *password* yang akan dimasukkan.

Tabel 3.3 Variasi masukan yang akan diuji pada *login page*

No.	<i>Username</i>	<i>Password</i>
1.	-	-
2.	-	admin
3.	-	asdf
4.	admin	-
5.	admin	admin
6.	admin	asdf
7.	asdf	-
8.	asdf	admin
9.	asdf	asdf

3.7.2. Pengujian fungsi *create user*

Pengujian membuat *user account* dilakukan dengan memasukkan input *username* (terdaftar atau tidak terdaftar), *password*, *confirmation password* (sama atau berbeda) secara bervariasi (Tabel 3.4). Informasi *user account* yang telah terdaftar yaitu *Username: "admin"* dan *password: "admin"*.

Tabel 3.4 Variasi masukkan untuk menguji modul *create user*

No.	username	Password dan confirmation password
1.	admin	-
2.	admin	Berbeda
3.	admin	Sama
4.	user1	-
5.	user1	Berbeda
6.	user1	Sama
7.	-	-
8.	-	Berbeda
9.	-	Sama

3.7.3. Pengujian fungsi *edit user*

a. *Edit password*

Pengujian ini dilakukan dengan langkah-langkah sebagai berikut:

- Membuat sebuah *user account* baru (menggunakan *user admin*)
- *Login* dengan menggunakan *username* dan *password* yang telah dibuat pada langkah sebelumnya.
- Mengubah *password* pada *user* yang baru dibuat (menggunakan *user admin*)
- *Login* dengan menggunakan *password* yang baru.

b. *Edit type*

Perbedaan *user type* yaitu pada tampilan *main menu*, yaitu administrator dapat mengakses modul *user management*, *controller management*, dan *monitoring & controlling*. Operator hanya dapat mengakses modul *monitoring & controlling*.

Pengujian dilakukan dengan mengubah sebuah *user* yang tipenya administrator menjadi operator kemudian melakukan *login* ulang dan masuk ke tampilan *main menu* untuk mengetahui apakah pengujian berhasil atau tidak. Hal yang sama akan dilakukan dengan mengubah *user* yang tipenya operator menjadi administrator

3.7.4. Pengujian fungsi *delete user account*

Pengujian dilakukan dengan menghapus sebuah *user* (menggunakan *user admin*), kemudian melakukan *login* dengan menggunakan user yang telah dihapus untuk memastikan apakah user tersebut telah dihapus.

3.7.5. Pengujian fungsi *add controller*

Pengujian dilakukan dengan memasukkan input EUI ZigBee pada kontroler dan alias untuk kontroler secara bervariasi (Tabel 3.5). Setelah itu, dilakukan pengecekan pada daftar kontroler untuk memastikan bahwa informasi kontroler tersebut berhasil ditambahkan.

Tabel 3.5 Variasi masukkan untuk menguji modul *add controller*

No.	EUI	Alias
1.	-	-
2.	-	Telah digunakan
3.	-	Belum digunakan
4.	terdaftar	-
5.	terdaftar	Telah digunakan
6.	terdaftar	Belum digunakan
7.	tidak terdaftar terdaftar	-
8.	tidak terdaftar terdaftar	Telah digunakan
9.	tidak terdaftar terdaftar	Belum digunakan

3.7.6. Pengujian fungsi *edit privileges*

Pengujian dilakukan dengan mencoba variasi privileges terhadap perangkat yang ada, kemudian melakukan pengecekan ulang untuk mengetahui apakah informasi tersebut telah berhasil diperbaharui.

3.7.7. Pengujian fungsi *delete controller*

Pengujian dilakukan dengan menghapus salah satu kontroler yang terdaftar, setelah itu melakukan pengecekan pada daftar kontroler untuk mengetahui apakah kontroler tersebut berhasil dihapus.

3.7.8. Pengujian *ZigBee handover*

3.7.9. Pengujian fungsi *monitoring & controlling*

a. Monitoring & controlling lamp(s)

Pengujian dilakukan dengan memantau status lampu yang di dalam ruangan kemudian membandingkannya dengan informasi yang ditampilkan oleh aplikasi. Setelah itu, dilakukan pengendalian terhadap keadaan lampu melalui aplikasi yang dibuat pada perangkat Mini2440.

b. Monitoring temperature value dan radius humidity (RH) value

Pengujian dilakukan dengan memantau nilai suhu dan kelembaban di dalam ruangan ketika AC dalam keadaan mati dan nilai yang dihasilkan pada saat AC dinyalakan.

c. Monitoring brightness level (%)

*Pengujian dilakukan dengan memantau tingkat penerangan yang dibaca sensor cahaya (*ambient light sensor*) ketika sensor tersebut tidak diberi penerangan dan diberi penerangan.*

d. Monitoring PIR

*Pengujian dilakukan dengan memantau status PIR pada saat sensor menerima *input* berupa gerakan tubuh manusia dan ketika sensor tidak menerima *input*.*

e. *Monitoring smoke sensor*

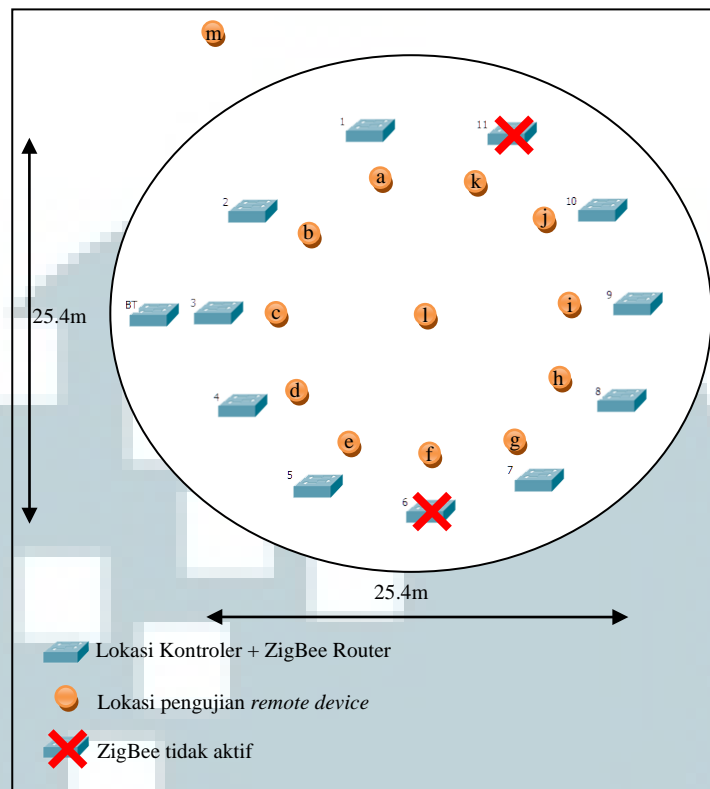
Prinsip kerja sensor asap yang digunakan yaitu sensor ini akan aktif apabila cahaya inframerah yang dipancarkan berhasil dipantulkan oleh asap dan diterima oleh sensor. Pengujian dilakukan dengan memantau kondisi terakhir dari sensor asap ketika sebelum dan setelah menerima *input*. *Input* pada sensor dilakukan dengan menutup permukaan sensor dengan tangan yang akan berfungsi untuk memantulkan cahaya inframerah pada sensor.

f. *Monitoring & control warning alarm*

Ketika sensor asap dalam keadaan aktif, maka status *warning alarm* akan ikut aktif. Pengujian dilakukan dengan cara memantau status *warning alarm* sebelum dan sesudah sensor asap diberikan *input*. Untuk menguji keberhasilan fungsi pengendalian *warning alarm*, maka *alarm* akan dimatikan dengan menggunakan aplikasi yang dikembangkan pada Mini2440 untuk dimonitor.

3.7.10. Pengujian koneksi

Pengujian ini dilakukan untuk mengetahui apakah ZigBee pada *remote device* dapat berfungsi di setiap lokasi di dalam dan di luar suatu topologi jaringan. Topologi jaringan yang akan digunakan untuk pengujian koneksi yaitu topologi yang diterapkan pada gedung Surya Research Center (Gambar 3.33).



Gambar 3.33 Topologi jaringan pada pengujian koneksi

Jarak terjauh antar kontroler (antar dua kontroler yang saling berseberangan) yaitu 25.5m. Pada penelitian sebelumnya (Hartono, 2011), jarak terjauh jangkauan yang dapat dicapai oleh ZigBee tanpa adanya penghalang yaitu 29m sedangkan jika ada penghalang yaitu 13m . Dengan begitu, maka diasumsikan setiap ZigBee dalam topologi tersebut dapat menjangkau ZigBee lainnya. *Remote device* yaitu Mini2440 yang terhubung dengan ZigBee akan dipindahkan berdasarkan 13 posisi berbeda untuk memantau perangkat yang terhubung pada *board tester* (posisi *board tester* tidak berubah-ubah).

3.7.11. Pengujian *network checking*

Pengujian bertujuan untuk menguji apakah daftar ZigBee tidak aktif yang ditampilkan oleh aplikasi benar atau tidak. Pengujian dilakukan dengan melakukan *scan network* untuk memperoleh

daftar perangkat ZigBee yang tidak aktif. Kemudian dari daftar yang diperoleh akan dilakukan pengecekan satu-persatu dengan cara melakukan *unicast* ke setiap perangkat menggunakan program terminal v1.9 pada komputer untuk mengetahui apakah ZigBee tersebut benar dalam keadaan tidak aktif.

3.7.12. Pengujian jumlah penggunaan *memory* oleh aplikasi

Penelitian ini bertujuan untuk mengetahui seberapa besar penggunaan *memory* yang digunakan oleh aplikasi ketika dijalankan. Penelitian dilakukan dengan mengambil data *free memory* pada saat aplikasi sebelum dijalankan, kemudian membandingkannya dengan data *free memory* ketika aplikasi telah dijalankan

UMMN