

## **BAB III**

### **PERANCANGAN SISTEM**

#### **3.1 Gambaran Umum Sistem**

Sistem pengaduan dan penyebaran informasi tersegmentasi ini harus memenuhi dua kebutuhan interface, baik dari sisi pengirim dan penerima informasi. Sistem pengaduan sendiri dipicu oleh pengguna aplikasi Android yang nantinya mengirimkan informasi berupa gambar dan teks kepada biro terkait dimana presentasi informasi ada di halaman web. Kemudian untuk sistem penyebaran informasi sistem dipicu oleh biro terkait untuk menyebarkan informasi dari interface halaman web ke aplikasi Android pengguna.

Halaman web administrasi pengguna harus tersedia form untuk memasukkan informasi yang sedianya akan disebarkan. Informasi ini juga harus dapat disimpan di *database* sebagai arsip. Administrator dapat memilih segmentasi sebaran informasi atau audience dari informasi tersebut. Segmentasi ini dapat berupa pengelompokkan pengguna berdasarkan fakultas, jurusan atau kelompok mahasiswa tertentu. Pengelompokkan ini dapat diatur melalui halaman khusus untuk memasukkan pengguna ke kelompok tertentu.

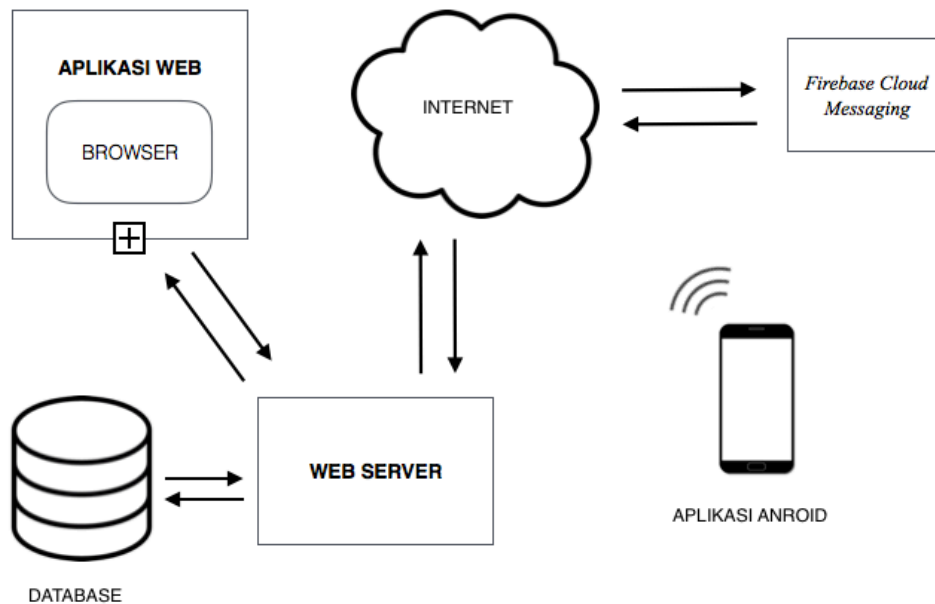
Selain itu halaman administrasi ini juga harus berisi laman khusus untuk melihat daftar informasi pengaduan yang telah disubmit oleh pengguna aplikasi Android. Administrator pengelola laman ini juga dapat dengan mudah memberikan *feedback* kepada pengguna yang telah memberikan laporannya.

Sementara itu pada sisi aplikasi Android pengguna akan mendapatkan notifikasi di area notifikasi (*system tray*) saat aplikasi pada keadaan *background / foreground* process. Aplikasi memungkinkan pengguna untuk mengumpulkan informasi laporan ke *server* baik berupa teks maupun gambar. Pengguna juga dapat terima

umpan balik dari administator terkait dengan informasi pelaporannya. Informasi yang didapat dari administrator disimpan secara offline sebagai arsip. Aplikasi akan secara otomatis mengunduh informasi terbaru dari *server* tiap kali dibuka selain juga ketika pengguna melakukan tab pada notifikasi yang tersedia jika terdapat informasi baru dari *server*.

### 3.2 Blok Diagram Sistem

Gambaran sistem secara keseluruhan ditunjukkan oleh blok diagram pada Gambar 3.1.



Gambar 3.1. Blok diagram sistem pelaporan dan penyebaran informasi tersegmentasi

*Building block* dari sistem terbagi menjadi dua bagian besar, yakni aplikasi web untuk administrasi biro tertentu dan aplikasi Android pengguna. Kedua sistem ini saling berkolaborasi untuk melakukan proses yang utuh.

Sistem administrasi berbasis web merupakan aplikasi web yang dibuat secara khusus untuk menanggapi fungsi *content management* mulai dari membuat, edit, hapus atau melihat konten yang sebelumnya ditambahkan. Aplikasi web ini membutuhkan *database server* untuk menyimpan arsip informasi yang diserbarkan di halaman administrasi dan informasi laporan yang dikumpulkan dari pengguna aplikasi Android. Aplikasi web ini disimpan dalam sebuah web *server* yang akan menanggapi semua permintaan dari *browser*.

Aplikasi Android pengguna akan menanggapi proses seperti notifikasi saat ada informasi baru yang diserbarkan, pengunduhan informasi dan pengumpulan informasi pengaduan. Aplikasi ini juga menanggapi penyimpanan sederhana untuk mengarsipkan semua informasi yang telah diterima atau dikumpulkan pengguna sebagai informasi laporan.

Antara aplikasi web di web *server* dan aplikasi Android pengguna harus terhubung melalui jaringan internet agar pertukaran informasi dapat dilakukan antara keduanya. Dalam hal ini pertukaran data secara umum dilakukan secara asinkron dengan format JSON. Selebihnya adalah notifikasi pada aplikasi Android yang dilakukan via FCM.

### **3.3 Perancangan Sistem**

Aplikasi web untuk sistem administrasi di *server* dikembangkan menggunakan pemrograman PHP untuk memproses sebagian besar proses administrasi dengan kolaborasi MySQL untuk melakukan penyimpanan data. Lebih dari itu JavaScript juga digunakan disini dengan bantuan pustaka JQuery untuk mendukung interaktifitas di sisi aplikasi web.

### 3.3.1 Desain Model

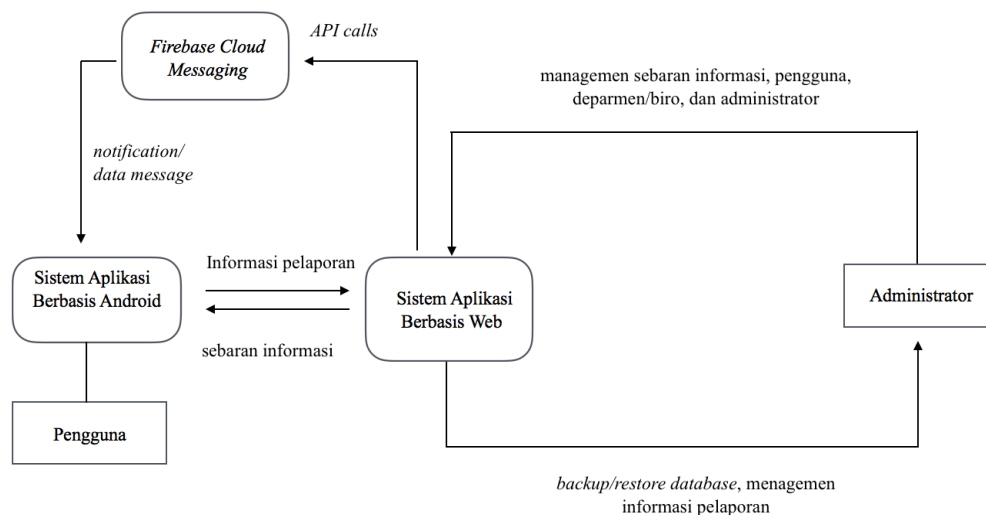
Desain model dalam sistem ini disajikan dalam bentuk *logical model* dan *physical model*. Bentuk *logical model* disajikan dalam bentuk Diagram Alir Data (DAD).

### 3.3.2 Diagram Arus Data

Seperti halnya semua desain sistem diagram arus data terbagi menjadi 3 bagian berdasarkan levelnya, yakni level 0, 1 dan 2.

#### Diagram Arus Data Level 0

Diagram arus data pada level ini menggambarkan sumber, proses, dan hasil secara umum. Ada pihak luar yang memberi masukan dan ada pihak yang menerima keluaran aplikasi. Dalam hal ini pihak luar disebut dengan entitas. Secara detail desain diagram konteks untuk sistem pelaporan dan penyebaran informasi tersegmentasi berbasis web dan Android seperti terlihat pada Gambar 3.2 berikut.



Gambar 3.2 Diagram konteks sistem pelaporan dan penyebaran informasi berbasis web dan Android

Seperti terlihat pada diagram bahwa terdapat dua entitas yang berhubungan dengan aplikasi, yakni pengguna dan administrator. Pengguna dalam hal ini adalah mahasiswa yang menggunakan aplikasi Android untuk mengumpulkan informasi laporan sekaligus menerima sebaran informasi yang disebar oleh administrator. Sementara itu administrator merupakan biro/departemen tertentu dalam institusi kampus yang ditugaskan untuk menyebarkan informasi khusus dan tentu saja menerima laporan secara khusus. Biro/departemen yang bertugas menyebarkan informasi dapat lebih dari satu selayaknya yang terdapat dalam institusi dengan objek sebaran yang berbeda pula. Begitu pula dengan biro yang dapat menerima informasi pelaporan. Pengguna dapat memilih biro tertentu yang tepat untuk menerima informasi tersebut.

Dalam sistem administrasi berbasis web, terdapat administrator utama (*top administrator*) yang memiliki otoritas untuk melakukan manajemen sistem informasi secara penuh. Seperti menambah/kurangkan administrator lain, setting otoritas seorang administrator, kelompok pengguna, penambahan biro tertentu, dll. Sementara administrator lain dapat pula melakukan fungsi manajemen tertentu dengan izin sesuai yang diberikan administrator utama.

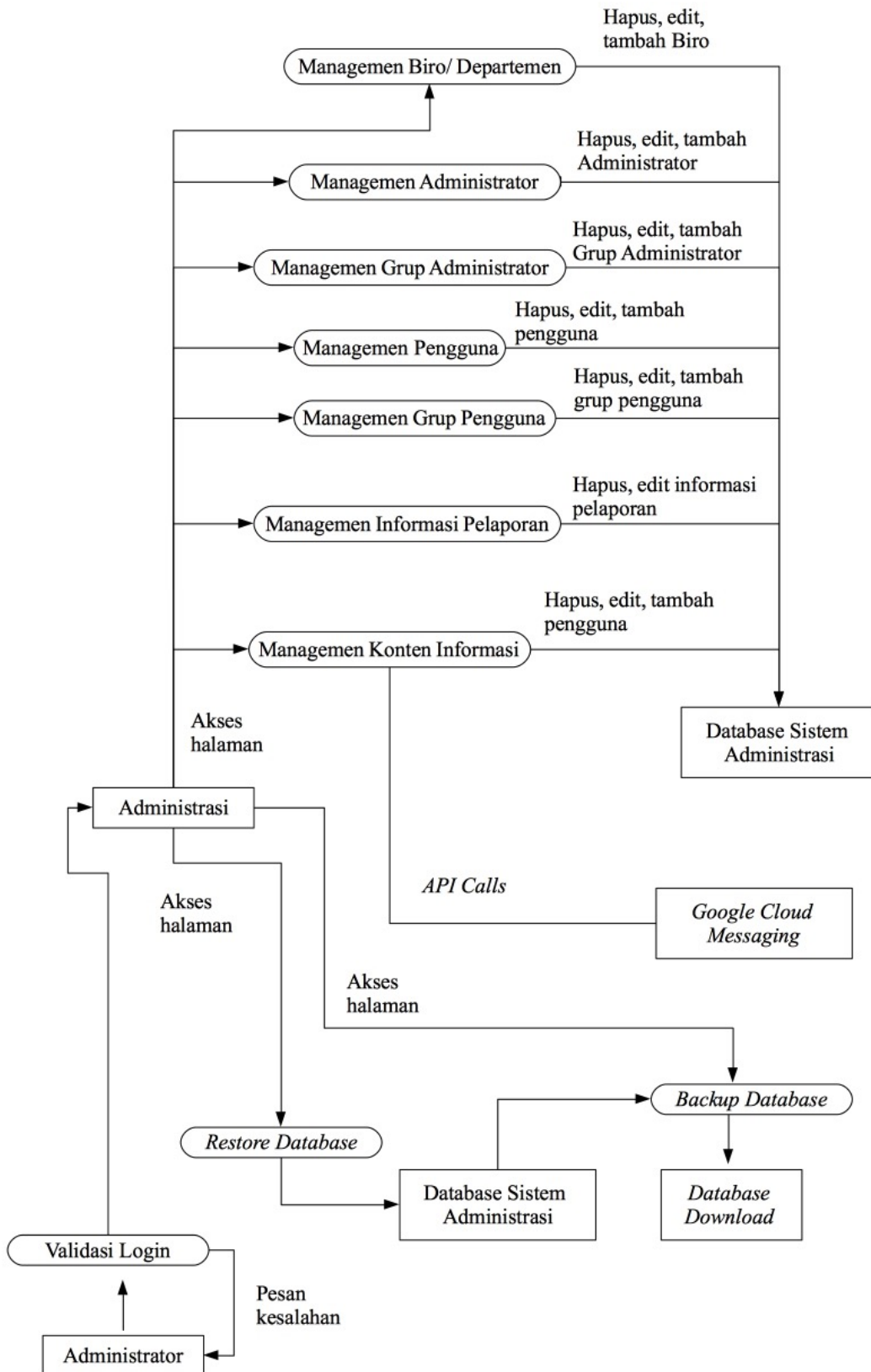
Sebuah halaman manajemen di sistem informasi web akan difungsikan untuk menambah/mengurangi pengguna, membuat kelompok/segmentasi pengguna, biro/departemen, konten informasi yang disebar atau sederhana melihat informasi laporan dari pengguna. Kemudian proses pencadangan (*backup*) dan pengembalian (*restore*) basis data dilakukan di halaman khusus sebagai langkah antisipasi keamanan data.

Kemudian pada sistem Android pengguna memiliki beberapa bagian penting, yakni manajemen informasi, pengguna, profil dan validasi *login*. Selain itu bagian lain dari sistem ini menangani notifikasi pengguna dan update informasi sebaran. Sistem aplikasi Android menerima data atau pesan notifikasi dari FCM yang sebelumnya telah dipanggil melalui sistem aplikasi *web*. Aplikasi Android

juga akan terintegrasi dengan sistem aplikasi web dengan memanggil *form* yang ada di halaman web untuk dimanfaatkan sebagai beberapa halaman manajemen di aplikasi Android. Seperti saat mengumpulkan laporan ke sistem aplikasi web atau merubah profil. Proses setelahnya akan terjadi melalui layanan *web*. Hal yang sama digunakan saat mengunduh informasi sebaran ke aplikasi.

### **Diagram Arus Data Level 1**

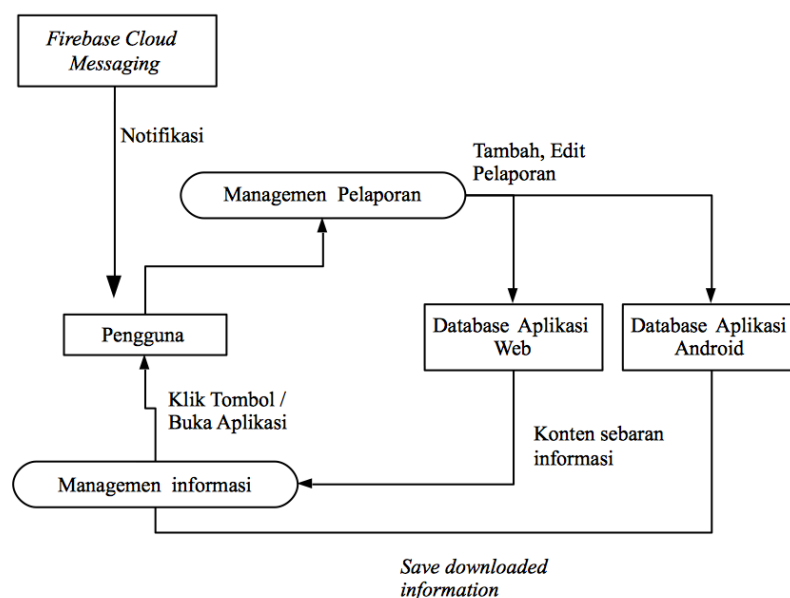
Diagram arus data level 0 merupakan jabaran yang lebih detail dari diagram konteks. Dibawah ini merupakan diagram arus level 0 dari sistem yang dibuat. Secara terpisah diagram dari sistem aplikasi web dan aplikasi Android masing-masing diperlihatkan pada Gambar 3.3 dan 3.4.



Gambar 3.3 Diagram arus data level 1 pada sistem aplikasi web

Sistem aplikasi berbasis web terdapat tiga proses utama, yakni *login* administrator, manajemen pengguna dan konten, dan pengiriman informasi. Semua data berkaitan dengan administrator dan pengguna akan disimpan di *database server*. Tiap informasi tersebut dapat ditambahkan, dihapus dan dimodifikasi sesuai kebutuhan secara dinamis melalui form yang disediakan. Pengguna, yakni target sebaran informasi juga harus ditambahkan secara manual.

Pemanggilan API *Firestore Cloud Messaging* hanya akan dilakukan ketika administrator mengirimkan pesan. Kemudian sistem akan meneruskannya ke masing-masing aplikasi pengguna.



Gambar 3.4 Diagram arus data level 1 pada sistem aplikasi Android

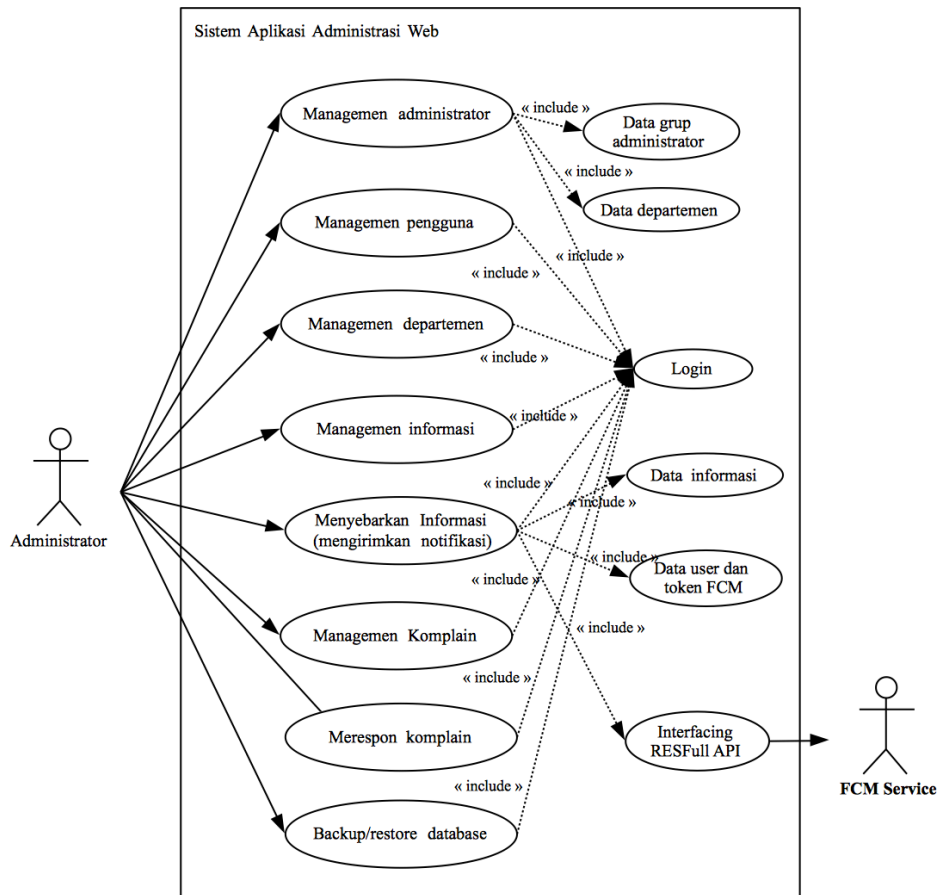
Sementara itu pada sisi aplikasi Android pengguna sistem secara keseluruhan lebih sederhana namun menjalankan fungsi yang hampir sama. Pengguna dapat melakukan manajemen pelaporan, yakni menambahkan informasi laporan dan memodifikasinya jika diperlukan. Data ini kemudian akan disimpan secara offline di aplikasi pengguna dan diunggah di *database* aplikasi web.



Notifikasi dari FCM akan berjalan satu arah, yakni ketika administrator memutuskan untuk mengirimkan informasi ke pengguna. Selanjutnya pengguna dapat membuka notifikasi dan atau sekaligus membuka aplikasi dimana ini menjadi pemicu agar aplikasi dapat mengunduh konten informasi yang lengkap dari *database* aplikasi web. Pengguna juga dapat secara manual mengunduh informasi ini dengan klik pada tombol yang disediakan. Semua informasi yang telah diunduh akan disimpan dalam *database* aplikasi Android.

### **3.3.3 Use Case Diagram**

*Use case diagram* merupakan salah satu jenis UML (*Unified Modelling Language*). Pada bab sebelumnya telah dibuat ERD untuk sistem aplikasi web. Pada bab ini interaksi antara *user* dengan sistem secara grafis dipresentasikan dalam sebuah *use case diagram*. Dari *bussiness rules* yang telah dirancang pada bab sebelumnya dapat dibuat *use case diagram* seperti halnya yang terdapat pada Gambar 3.5 berikut.



Gambar 3.5 Use case dengan aktor administrator pada sistem aplikasi administrasi berbasis web

Sebuah akun administrator akan memiliki akses terhadap banyak halaman manajemen, termasuk manajemen administrator yang mengatur penambahan administrator di sistem aplikasi web.

Halaman manajemen pengguna secara khusus digunakan untuk mengatur grup pengguna yang merupakan kelompok sebaran pengguna (mahasiswa). Manajemen pengguna juga meliputi penambahan, edit dan hapus pengguna. Data mahasiswa ini dapat diperoleh ketika mereka mendaftar via aplikasi Android. Data akan disimpan di *database* aplikasi web administrasi. Administrator juga dapat memasukkan data mahasiswa melalui manajemen pengguna. Lebih dari itu

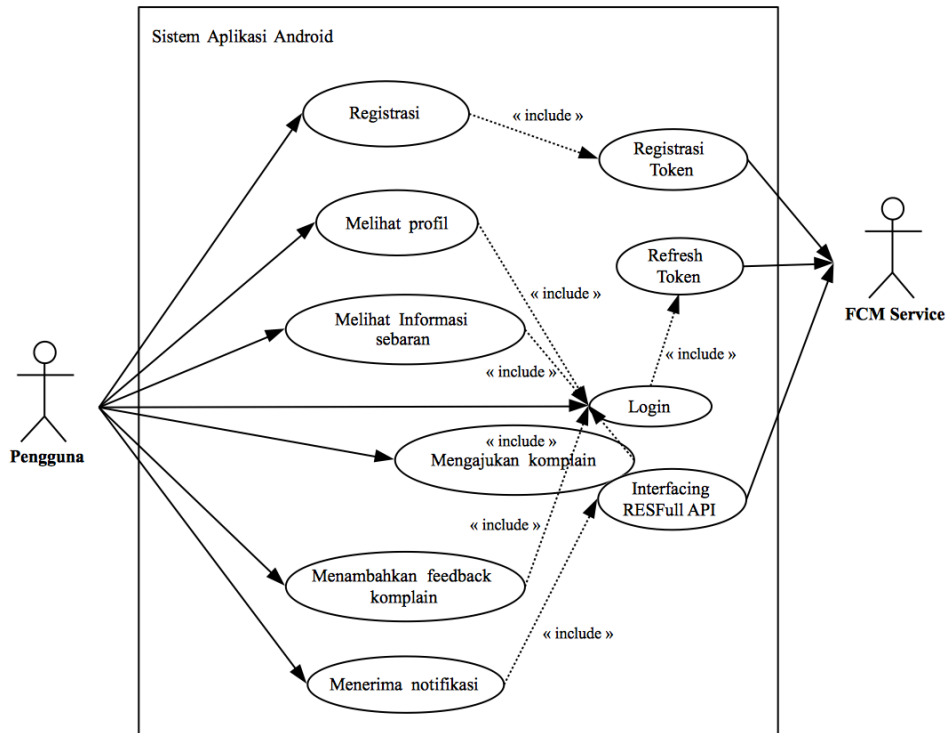
administrator juga dapat menyunting informasi profil mahasiswa, bahkan menghapusnya.

Pada manajemen informasi administrator dapat menambah, edit dan hapus informasi. Penyebaran informasi baru bisa dilakukan setelah penambahan informasi, baik untuk individual atau grup pengguna. Kemudian manajemen komplain baru bisa dilakukan setelah ada data komplain yang di-submit oleh pengguna. Administrator memiliki hak untuk menghapus dan mengedit konten komplain. Dan memberikan respon terhadap komplain tersebut.

Pengadaan fitur tambahan yang di sisi aplikasi web administrasi adalah fitur *backup* dan *restore*. Fitur *backup* akan mencadangkan semua baris data pada semua tabel di *database*. Sebaliknya, fitur *restore* akan mengembalikan data tabel sesuai baris data SQL yang ada di file *.sql* yang diimpor ke sistem *restore*.

Dari sisi pengguna mereka dapat melakukan beberapa administrasi via aplikasi Android yang telah terpasang. Fitur yang paling dasar, yakni registrasi via aplikasi Android. Data registrasi disimpan di *database* aplikasi web. Registrasi via aplikasi Android sekaligus melakukan *interfacing* terhadap layanan Google, yakni FCM untuk mendapatkan *token* registrasi. Identitas *token* yang sama juga diperoleh

ketika pengguna melakukan *login* pada aplikasi. *Use case* dari sisi pengguna dapat dilihat pada Gambar 3.6.



Gambar 3.6 *Use case* dengan aktor pengguna pada sistem aplikasi Android

### 3.3.4 Entry, Fetch dan Update Data

Baik pada sistem aplikasi web maupun aplikasi Android pengguna terdapat proses *entry*, *fetch*, *update* data yang merupakan tiga proses utama yang ada dalam aplikasi ini. Secaran khusus pada tiap halaman manajemen terdapat halaman *entry* digunakan untuk memasukkan informasi tertentu, baik informasi konten, administrator, biro, dll. Form isian akan mengumpulkan informasi yang dibutuhkan kemudian disimpan dalam *database server* yang telah diatur mekanismenya oleh MySQL. Secara teknis fungsi ini berjalan dengan kolaborasi *script* PHP di *server* yang menjalankan *query SQL* untuk menambahkan baris pada tabel MySQL.

Sementara pada proses *fetching* dan *update* yang ada di *database* dilakukan pada proses ekspor data, manajemen data, manajemen pengguna, manajemen

grup pengguna, manajemen administrator, manajemen grup administrator, *backup* dan *recovery*. Namun demikian implementasi dari *fetch* dan *update* ini berbeda dari sisi SQL yang digunakan. Pengambilan data (*fetch*) dilakukan dengan statement *select* pada SQL dan *update* dengan menggunakan statement *update*.

Pengambilan data dari *database* selanjutnya ditampilkan dalam bentuk kode HTML yang dapat dilihat di halaman web. Kode ini merupakan hasil *parsing* dari kode PHP yang diterjemahkan disisi *server*. Termasuk kerjasama PHP dengan MySQL juga merupakan satu rangkaian proses dalam proses pengambilan data.

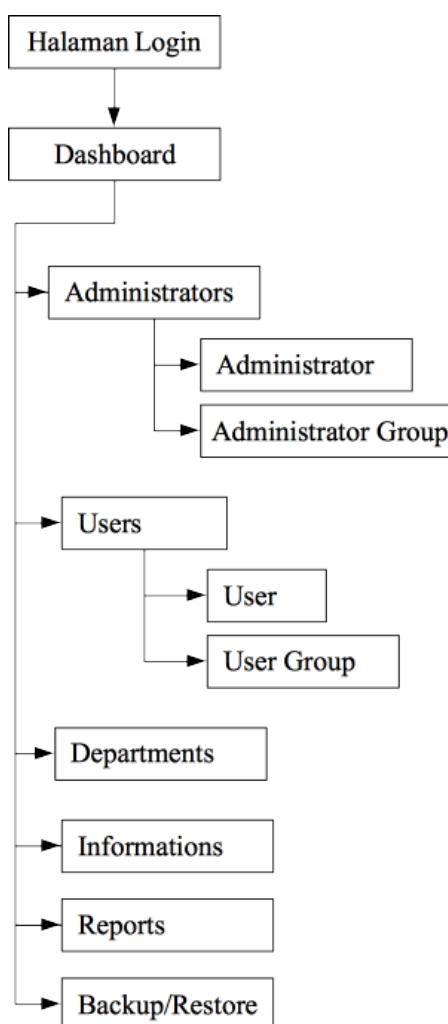
Pada diagram alir diatas terlihat bahwa proses pengambilan data dilakukan dengan *input* dari sisi pengguna berupa *trigger*. Secara fisik trigger ini diwujudkan dengan tombol *submit* atau akses terhadap suatu halaman. Beberapa *event trigger* memerlukan paramater khusus yang dijadikan patokan untuk menyeleksi data yang diambil dari *database*. Seperti pada pengambilan data pada periode tertentu.

Pada proses *update* mekanisme yang sama berlaku. Diperlukan sebuah pemicu untuk melakukan fungsi *update*. Form isian yang sebelumnya telah dipenuhi oleh pengguna dijadikan sumber konten untuk memperbarui konten di *database*. Perbaruan data dilakukan dengan statement SQL, yakni *update*. Setiap eksekusi suatu statement SQL dengan fungsi query di PHP maka akan mengembalikan status error atau sukses. Hal ini dijadikan sebagai parameter untuk menuliskan pesan ke pengguna. Jika ternyata muncul pesan error hal itu merupakan isyarat bagi pengguna untuk mengulangi proses *update*.

### **3.4 Desain Masukan dan Keluaran Sistem Aplikasi Web**

Manajemen konten dan pengguna pada sisi aplikasi web lebih kompleks dikarenakan harus mengatur banyak informasi pengguna dan konten. Secara fisik aplikasi web ini terdiri dari kumpulan halaman yang menjalankan fungsi khusus. Dimana tiap halaman berisi skrip PHP melakukan seleksi data yang telah diterjemahkan oleh web *server*.

Halaman administrasi memiliki menu yang menuju ke halaman tertentu dimana halaman tersebut memiliki fungsi khusus yang relevan dengan nama menunya. Untuk mendapatkan akses terhadap halaman administrasi ini seorang administrator harus melewati validasi di *halaman login*. Secara umum desain menu pada halaman administrasi seperti tampak pada Gambar 3.7.



Gambar 3.7 Desain menu halaman administrasi aplikasi web

Saat mengakses alamat web untuk administrasi maka administrator akan diajukan sebuah halaman *login*. Pada halaman ini diperlukan informasi *username* dan *password* untuk kemudian divalidasi dan mendapatkan akses ke halaman administrasi. Setelah berhasil masuk ke halaman administrasi maka administrator

akan dibawa ke halaman *dashboard*, yakni halaman di aplikasi web yang berisi rangkuman aktifitas terbaru yang dilakukan oleh sistem.

Pada menu lain *top administrator* memiliki hak akses penuh terhadap semua halaman. Tiap halaman tentu memiliki fungsi yang khusus. Seperti halnya pada menu *Department Group*, administrator dapat menambahkan nama departemen tertentu sesuai kebutuhan. Pada menu *Users* terdapat sub menu untuk *user* yang merupakan halaman untuk manajemen pengguna. Halaman *informations* akan berisi halaman untuk manajemen informasi yang akan disebarkan ke pengguna. Pada halaman ini informasi dapat ditambahkan, disimpan dan dimodifikasi jika diperlukan. Penyebaran informasi juga terdapat di halaman yang sama dengan objek sebaran individu pengguna tertentu atau kelompok pengguna yang telah ditentukan di halaman *user groups*. Sementara menu *backup/recovery* merupakan *tool* untuk melakukan *backup* dan *recovery database*.

### **3.4.1 Desain Antarmuka**

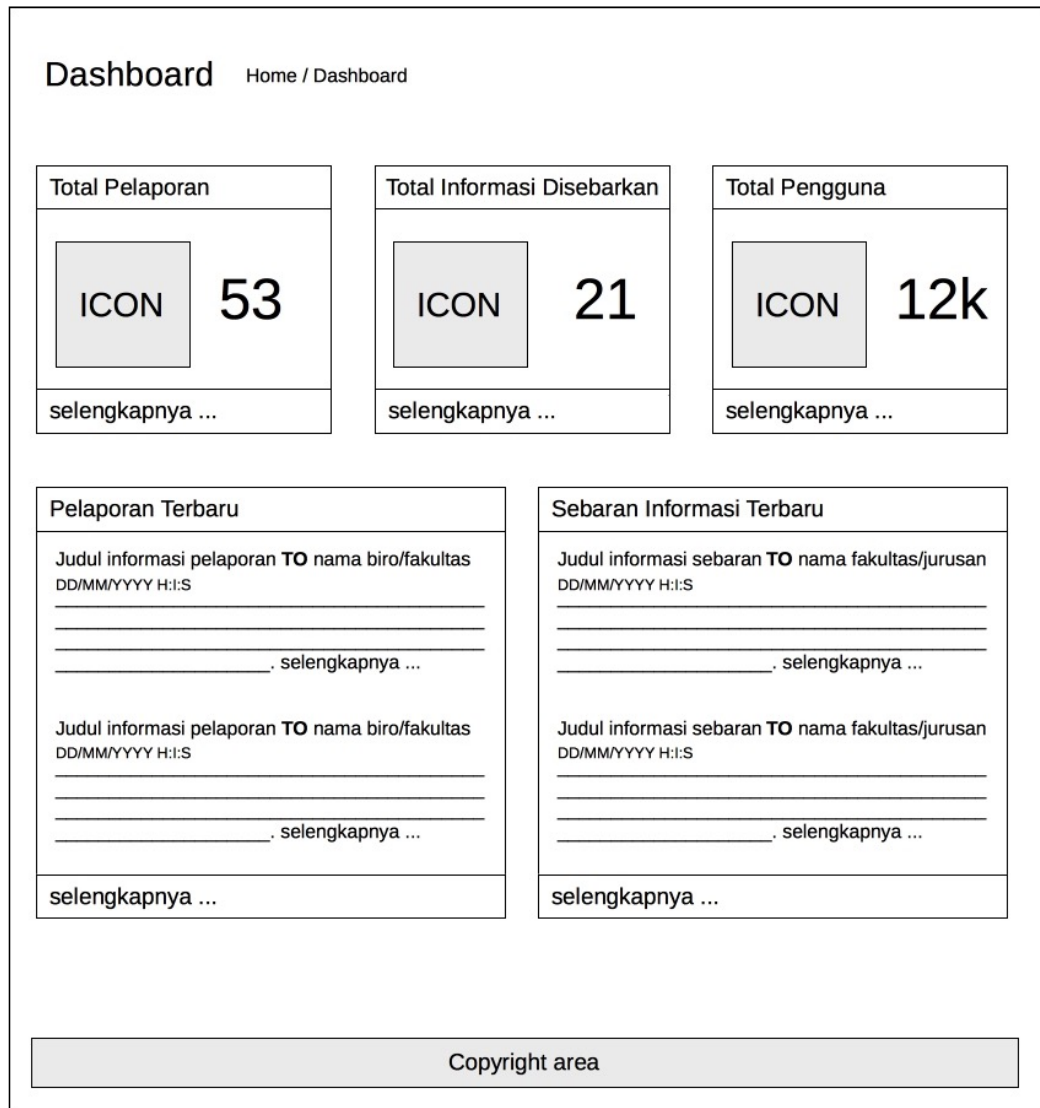
Desain antarmuka disini adalah desain form yang meliputi *input/output* pada halaman yang digunakan untuk memasukkan data informasi pada sistem aplikasi berbasis web. Form ini terdapat pada tampilan utama dari halaman manajemen dan penambahan informasi.

#### **3.4.1.1 Desain Dashboard**

Halaman ini digunakan untuk merangkum aktifitas sistem secara keseluruhan. Terutama mengenai pelaporan dan sebaran informasi yang telah disebarkan. Pada halaman ini terdapat akses cepat untuk beberapa halaman manajemen dan informasi tentang pelaporan terbaru yang ditambahkan oleh pengguna aplikasi Android. Secara rinci desain antarmuka halaman *dashboard* ini diperlihatkan pada Gambar 3.8.

Pada gambar diatas total pelaporan, total informasi sebaran dan total pengguna ditampilkan pada masing-masing kotak dengan *icon* yang relevan. Di bawahnya terdapat

link untuk menuju ke halaman manajemen terkait. Sementara itu di bawahnya terdapat daftar pelaporan dan sebaran informasi terbaru yang telah dilengkapi dengan judul informasi, biro/fakultas/jurusan yang menjadi objek informasi, tanggal dan jam.



Gambar 3.8 Desain halaman *dashboard* aplikasi Android pengguna

### 3.4.1.2 Desain Manajemen Administrator

Terdapat fungsi untuk menambahkan grup administrator dan administrator sendiri. Grup administrator ini dapat dikategorikan sesuai otoritasnya. Misal sebuah grup yang mengelola sirkulasi informasi laporan dan grup lain yang mengelola



informasi sebaran. Pada halaman ini administrator dapat menambahkan, memodifikasi, menghapus dan melihat daftar administrator yang ada. Hal yang sama dapat dilakukan untuk pengaturan grup administrator. Desain untuk manajemen grup administrator dapat dilihat di Gambar 3.9.

**Administrator Group** Home / Administrator

TAMBAH HAPUS

| <input type="checkbox"/> | Nama Grup Administrator | Keterangan | Aksi                          |
|--------------------------|-------------------------|------------|-------------------------------|
| <input type="checkbox"/> | _____                   | _____      | <input type="checkbox"/> EDIT |
| <input type="checkbox"/> | _____                   | _____      | <input type="checkbox"/> EDIT |

Menampilkan 1 hingga 2 dari 2 (1 Halaman)

Copyright area

Gambar 3.9 Antarmuka manajemen administrator *group*

Halaman ini digunakan untuk melihat semua grup administrator yang telah ditambahkan. Pada halaman ini administrator dapat menambahkan, menghapus dan mengedit grup administrator. Elemen HTML yang berupa *checkbox* digunakan untuk menyeleksi suatu baris data yang akan dihapus. Kemudian administrator dapat memilih tombol hapus untuk menghapus data tersebut. Tombol edit akan mengedit data sesuai barisnya. Kemudian tombol tambah merupakan *link* untuk menuju halaman tambah/edit grup administrator seperti terlihat pada Gambar 3.10.

Administrator Group Home / Administrator Group

SIMPAN BATAL

Tambah Grup Administrator

Nama Grup Administrator

Keterangan

Hak Akses  \_\_\_\_\_  
 \_\_\_\_\_  
 \_\_\_\_\_  
 \_\_\_\_\_

Hak Modifikasi  \_\_\_\_\_  
 \_\_\_\_\_  
 \_\_\_\_\_  
 \_\_\_\_\_

Copyright area

Gambar 3.10 Desain halaman tambah/edit grup administrator

Halaman ini akan secara otomatis menghadirkan daftar file yang memiliki fungsi tertentu. Elemen *checkbox* digunakan untuk memilih hak akses/modifikasi apa saja yang dimiliki oleh sebuah grup administrator. Nama grup administrator sendiri dapat ditambahkan di *field* yang tersedia. Tombol simpan akan memproses form untuk disimpan di *database* dan tombol batal merupakan *link* ke halaman manajemen grup administrator.

Selanjutnya setelah administrator menambahkan grup administrator, maka dapat ditambahkan administrator lain yang memiliki hak akses / modifikasi sesuai dengan grup administrator yang telah dibuat. Proses menambah, menghapus dan mengedit sebuah akun administrator ini terdapat di fungsi manajemen administrator yang desainnya ditunjukkan pada Gambar 3.11.

The image shows a web interface for managing administrators. At the top left, it says "Administrator" and "Home / Administrator". To the right, there are two buttons: "TAMBAH" (Add) and "HAPUS" (Delete). Below these is a table with the following structure:

| <input type="checkbox"/> | Username | Status | Administrator Group | Aksi                          |
|--------------------------|----------|--------|---------------------|-------------------------------|
| <input type="checkbox"/> | _____    | _____  | _____               | <input type="checkbox"/> EDIT |
| <input type="checkbox"/> | _____    | _____  | _____               | <input type="checkbox"/> EDIT |

Below the table, it says "Menampilkan 1 hingga 2 dari 2 (1 Halaman)". At the bottom, there is a "Copyright area" box.

Gambar 3.11 Desain halaman manajemen administrator

Daftar dari pengguna dapat dilihat di manajemen administrator. Seperti manajemen sebelumnya, halaman ini juga terdapat akses untuk menambahkan, menghapus dan mengedit administrator. Kolom yang ditampilkan adalah *username*, status dan grup administrator dari akun administrator yang relevan. Penambahan / modifikasi data administrator dilakukan pada halaman tambah/edit administrator seperti pada Gambar 3.12.

Form tambah administrator terdiri atas kolom *username* dan *password* untuk informasi *login* saat akan masuk ke sistem aplikasi web. Kemudian pemilihan grup administrator yang sebelumnya telah ditambahkan. Nama depan, belakang,

gambar dan status administrator apakah diaktifkan atau tidak. Kolom gambar memungkinkan administrator untuk menambahkan gambar untuk tiap administrator dalam sistem. Seorang administrator juga dapat diasosiasikan dengan departemen tertentu jika telah ditambahkan departemen sebelumnya. Hal ini untuk menyaring informasi yang dapat dilihat oleh administrator ketika membuka halaman manajemen pelaporan. Sehingga hanya melihat laporan yang ditujukan pada departemen terkait.

The image shows a web interface for managing administrators. At the top left, it says "Administrator" with a breadcrumb "Home / Administrator". On the top right, there are two buttons: "SIMPAN" (Save) and "BATAL" (Cancel). Below these is a title bar "Tambah Administrator". The form contains the following fields:

- Username: text input
- Grup Administrator: dropdown menu
- Nama Depan: text input
- Nama Belakang: text input
- Departemen: dropdown menu
- Gambar: image upload area (represented by a grey square)
- Password: text input
- Ulangi Password: text input
- Status: dropdown menu

At the bottom of the form area, there is a "Copyright area" label.

Gambar 3.12 Desain halaman tambah/edit administrator

### 3.4.1.3 Desain Manajemen Pengguna

Manajemen pengguna digunakan untuk mengelola data pengguna, yakni data mahasiswa yang meliputi informasi nama, nomer induk mahasiswa (NIM) dan profil lain. Dalam penelitian ini, informasi tersebut harus dimasukkan dalam *database* aplikasi *web*. Pengelolaan kelompok pengguna juga diadakan di untuk pengiriman pesan berdasarkan kelompok yang ditangani oleh manajemen grup pengguna. Detil desain untuk halaman manajemen grup pengguna ditunjukkan pada Gambar 3.13.

|                          | Nama Grup Pengguna | Keterangan | Aksi                          |
|--------------------------|--------------------|------------|-------------------------------|
| <input type="checkbox"/> | _____              | _____      | <input type="checkbox"/> EDIT |
| <input type="checkbox"/> | _____              | _____      | <input type="checkbox"/> EDIT |

Menampilkan 1 hingga 2 dari 2 (1 Halaman)

Copyright area

Gambar 3.13 Desain halaman manajemen pengguna

Halaman tambah/edit *user* grup ditunjukkan oleh Gambar 3.14. Pada halaman ini hanya dua informasi yang diperlukan, yakni nama grup pengguna dan keterangan yang bersifat optional.

Sementara itu halaman manajemen pengguna sendiri lebih kompleks yang dilengkapi dengan *filter* data pada daftar tabel pengguna. Hal ini dimaksudkan untuk mempermudah pencarian data jika diperlukan. Tampilan detil untuk manajemen pengguna ini ada pada Gambar 3.15. Terdapat *filter* yang meliputi hampir semua data tentang pengguna. Tombol *filter* digunakan untuk menyeleksi

data sesuai form isian *filter*. Daftar tabel dibawahnya akan menampilkan baris data sesuai *filter* yang dimaksud.

Gambar 3.14 Desain halaman tambah/edit grup pengguna

|                          | Nama Pengguna | NIM   | Grup Pengguna | Email | Status | Aksi                                |
|--------------------------|---------------|-------|---------------|-------|--------|-------------------------------------|
| <input type="checkbox"/> | _____         | _____ | _____         | _____ | _____  | <input type="button" value="EDIT"/> |
| <input type="checkbox"/> | _____         | _____ | _____         | _____ | _____  | <input type="button" value="EDIT"/> |

Gambar 3.15 Desain manajemen pengguna

Halaman tambah/edit pengguna telah disederhanakan dari profil pengguna yang sebenarnya. Karena yang dibutuhkan adalah beberapa informasi yang digunakan pada FCM. Form ini ditunjukkan pada Gambar 3.16.

The image shows a web form for adding a user. At the top left, it says 'User' and 'Home / User'. On the top right, there are two buttons: 'SIMPAN' and 'BATAL'. The main form area is titled 'Tambah Pengguna' and contains the following fields:

- Nama Depan: text input field
- Nama Belakang: text input field
- Grup Pengguna: dropdown menu
- NIM: text input field
- Email: text input field
- Nomer HP: text input field
- Password: text input field
- Ulangi Password: text input field
- Status: dropdown menu

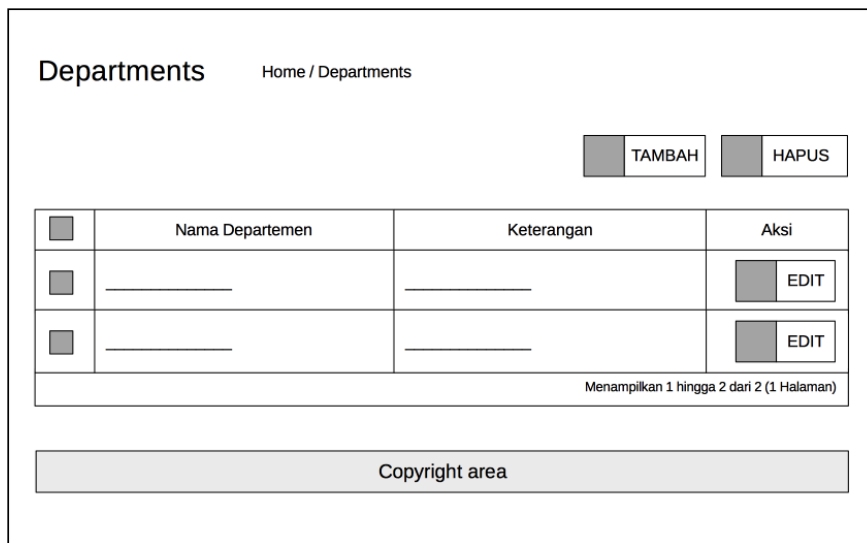
At the bottom of the form, there is a 'Copyright area' label.

Gambar 3.16 Desain halaman tambah/edit pengguna

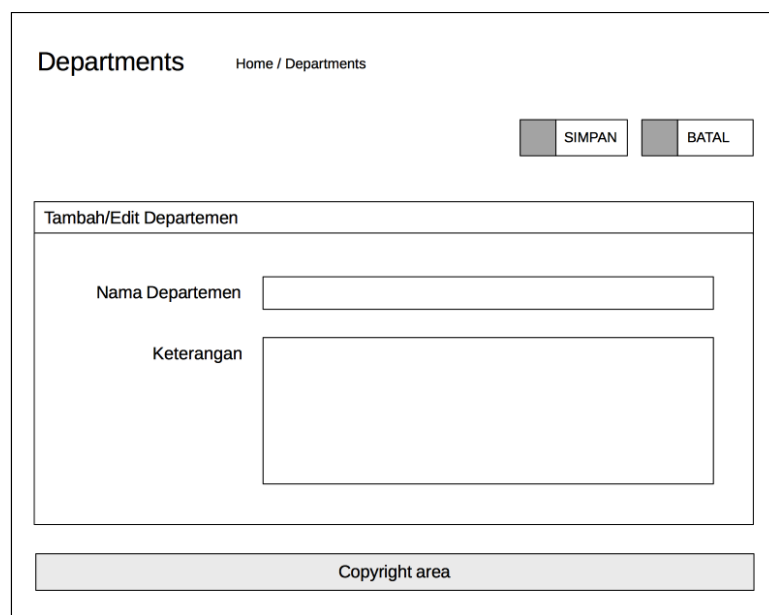
#### 3.4.1.4 Desain Manajemen Departemen

Manajemen departemen secara umum difungsikan untuk mengelola semua departemen yang ada di institusi kampus. Departemen ini sifatnya umum, dapat berupa nama jurusan, fakultas, biro tertentu di kampus, atau unit kegiatan

mahasiswa (UKM). Pengadaan manajemen ini dimaksudkan untuk mengkategorisasikan subjek dalam sistem pelaporan. Pengguna aplikasi Android dapat mengadukan masalah pada suatu departemen yang relevan sesuai departemen yang menangani hal tersebut. Detil manajemen departemen ditunjukkan pada Gambar 3.17 kemudian halaman tambah /edit departemen pada Gambar 3.18.



Gambar 3.17 Desain halaman manajemen departemen



Gambar 3.18 Desain halaman tambah/edit departemen



### 3.4.1.5 Desain Halaman Manajemen Informasi

Halaman ini secara khusus digunakan untuk mengelola informasi yang akan disebarakan kepada pengguna aplikasi Android. Halaman ini memiliki fitur *filter* seperti halnya halaman pengguna untuk mencari arsip informasi secara spesifik. Pengiriman informasi sendiri melalui prosedur khusus yang harus dipicu oleh sebuah klik pada tombol yang disediakan. Desain halaman manajemen informasi ini ditunjukkan oleh Gambar 3.19.

|                          | Judul Informasi | Penulis | Target Grup Pengguna | Tanggal Disebarakan | Status | Aksi  |
|--------------------------|-----------------|---------|----------------------|---------------------|--------|---|
| <input type="checkbox"/> | _____           | _____   | _____                | _____               | _____  | <input type="button" value="EDIT"/> <input type="button" value="SEBARKAN"/> |
| <input type="checkbox"/> | _____           | _____   | _____                | _____               | _____  | <input type="button" value="EDIT"/> <input type="button" value="SEBARKAN"/> |

Menampilkan 1 hingga 2 dari 2 (1 Halaman)

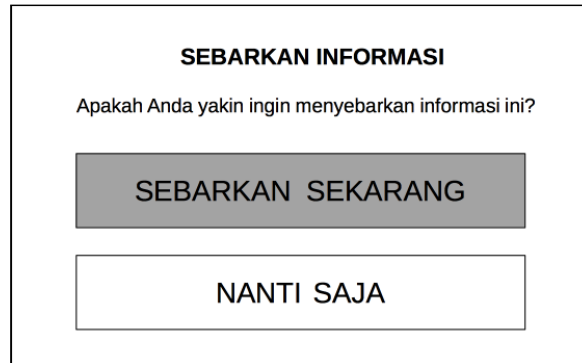
Copyright area

Gambar 3.19 Desain halaman manajemen informasi

Selain akses terhadap tambah/edit informasi, halaman ini berisi tombol untuk menyebarkan informasi sebaran. Sebuah informasi terlebih dulu harus ditambahkan baru dapat disebarakan kepada pengguna. Kemudian edit informasi pasca informasi tersebut disebarakan dapat dilakukan. Data pada sisi pengguna akan dimutakhirkan saat membuka aplikasi.

Administrator harus memilih informasi yang ingin disebarakan kemudian melakukan klik pada tombol sebarakan. *Event* ini akan menghadirkan *pop up* dialog konfirmasi apakah akan menyebarkannya saat itu juga atau tidak. Konfirmasi positif akan membuat sistem mengirim notifikasi kepada pengguna aplikasi

Android melalui *interfacing* terhadap API FCM. Desain *popup dialogue* ini sendiri ditunjukkan pada Gambar 3.20.



Gambar 3.20 *Popup dialogue* saat klik tombol sebar

Pada halaman tambah/edit informasi terdapat kolom informasi yang memiliki *teks editor* sehingga penulis dapat leluasa memberikan format tertentu pada informasi yang akan disebarkan. Kolom keterangan digunakan untuk memberikan informasi tambahan secara personal kepada penulis. Informasi tentang penulis akan diambil secara otomatis oleh sistem dengan memanfaatkan informasi *login*. Kolom status akan mengaktifkan / nonaktifkan suatu informasi. Hal ini akan menentukan ketersediaan informasi pada sisi aplikasi Android pengguna. Desain *form* isian untuk halaman tambah/edit informasi ditunjukkan pada Gambar 3.21.

The image shows a web interface for adding information. At the top left, it says 'Informasi' and 'Home / Informasi'. On the top right, there are two buttons: 'SIMPAN' and 'BATAL'. Below this is a form titled 'Tambah Informasi'. The form contains five input fields: 'Judul Informasi' (a text box), 'Target Grup Pengguna' (a dropdown menu), 'Informasi' (a large text area), 'Keterangan' (a text box), and 'Status' (a dropdown menu). At the bottom of the form area, there is a grey bar labeled 'Copyright area'.

Gambar 3.21 Desain halaman tambah/edit informasi

### 3.4.1.6 Desain Halaman Manajemen Pelaporan

Sesuai namanya halaman ini digunakan untuk mengelola informasi laporan yang masuk. Halaman ini harus dapat memberikan *feedback* kepada pelapor atas laporannya selain juga melihat informasi laporan tersebut. Administrator yang diasosiasikan pada departemen tertentu hanya akan melihat informasi laporan yang ditujukan pada departemen tersebut. Hal ini untuk membatasi informasi yang diterima oleh suatu departemen. Desain halaman ini ada pada Gambar 3.22 berikut.

Complains [Home / complains](#)

|                          | Judul Laporan | Penulis | Target Departemen | Tanggal Pelaporan | Status | Aksi   |
|--------------------------|---------------|---------|-------------------|-------------------|--------|--|
| <input type="checkbox"/> | _____         | _____   | _____             | _____             | _____  | <input type="button" value="EDIT"/> <input type="button" value="KIRIM BALASAN"/> |
| <input type="checkbox"/> | _____         | _____   | _____             | _____             | _____  | <input type="button" value="EDIT"/> <input type="button" value="KIRIM BALASAN"/> |

Menampilkan 1 hingga 2 dari 2 (1 Halaman)

Copyright area

Gambar 3.22 Halaman manajemen laporan

Pada halaman ini terdapat akses untuk membalas laporan yang masuk pada masing-masing baris informasi laporan. Klik pada tombol ini akan membawa administrator pada form khusus untuk membalas laporan. Edit terhadap laporan yang masuk juga memungkinkan untuk membenarkan redaksi, mengalihkan laporan ke departemen dan memasukkan teks balasan. Sementara pengiriman balasan ke pengguna aplikasi Android dilakukan tersendiri melalui klik pada tombol kirim balasan. Seperti sebelumnya klik pada tombol ini akan menghadirkan kotak dialog untuk mengkonfirmasi aksi administrator. Administrator juga punya wewenang untuk menghapus laporan yang tidak relevan. Profil pelapor juga dapat dilihat dengan cara klik pada tombol profil pelapor. Form untuk edit laporan ada pada Gambar 3.23.

Complain [Home / Complain](#)

**Edit Complain**

Judul Laporan

Pelapor

Target Departemen  ▼

Laporan

Balasan

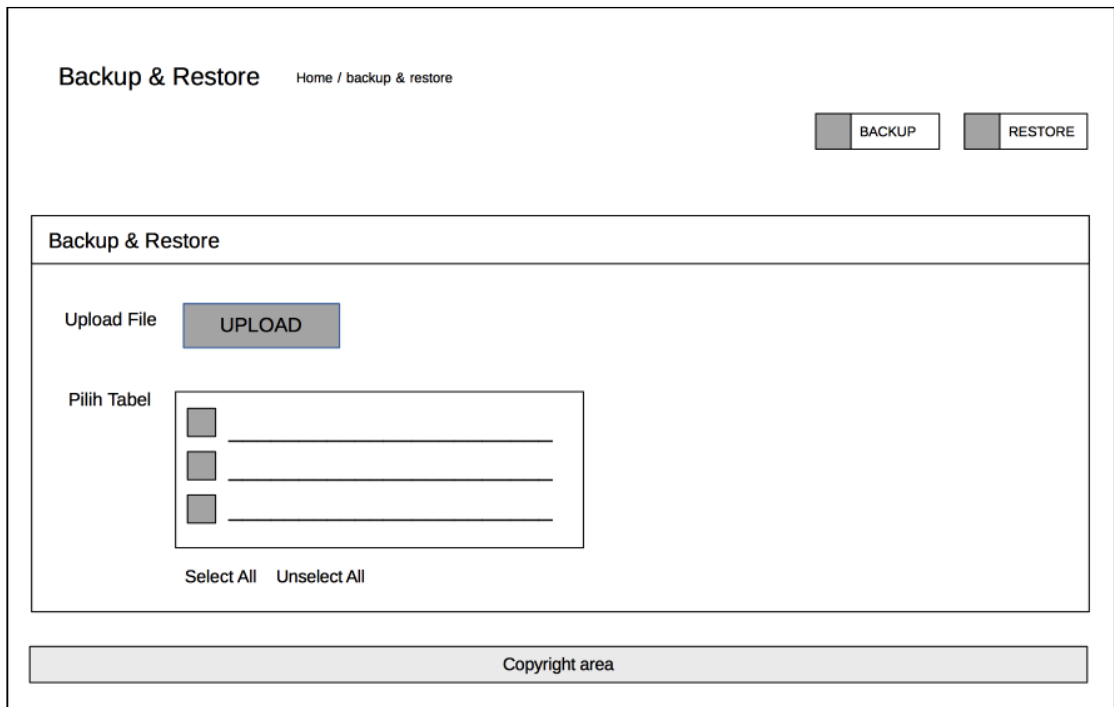
Status  ▼

Copyright area

Gambar 3.23 Desain *form* untuk mengedit laporan

### 3.4.1.7 Desain Halaman Manajemen Backup/Restore

Halaman ini digunakan untuk melakukan *backup/restore database* dari sistem aplikasi web. Data yang dapat dicadangkan adalah semua tabel dalam *database*. Fitur ini diadakan mengingat pentingnya data yang ada di sistem aplikasi web. *Database* yang telah dicadangkan ini dapat dikembalikan lagi dengan fitur *restore* di halaman yang sama. Desain halaman ditunjukkan pada Gambar 3.24.



Gambar 3.24 Desain halaman *backup* dan *restore*

Kolom pilih tabel digunakan untuk memilih tabel apa saja yang akan dicadangkan dari *database* aplikasi web. Secara *default* semua tabel akan terseleksi. Administrator dapat memilih tabel yang akan dicadangkan dan klik pada tombol *backup* untuk mengunduh *database*. Kolom *upload* untuk memilih berkas *database* yang telah dicadangkan sebelumnya untuk dilakukan proses *restore*. Tombol *restore* akan mengeksekusi isi *file* yang di-*upload*, yakni *query insert* untuk memasukkan data ke *database*.

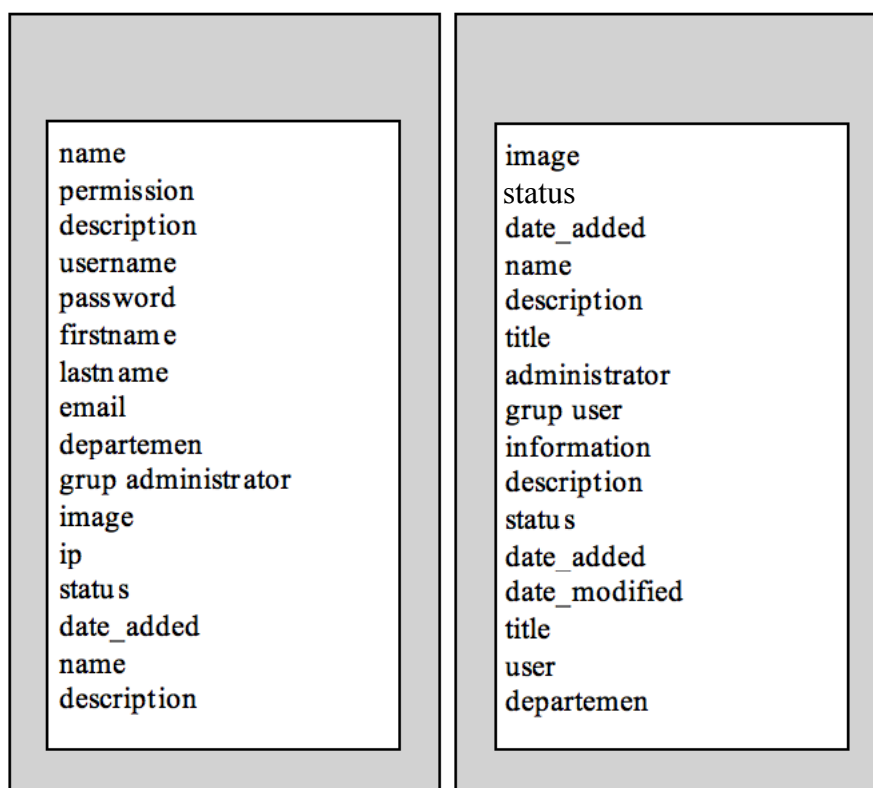
### 3.4.2 Rancangan Basis Data

Proses perancangan basis data harus melalui proses pemodelan data yang diawali dengan penentuan objek, yakni entitas dan atribut. Objek ini harus merupakan representasi dari tipe data yang akan disimpan. Pada tahap ini detail informasi tentang tabel, baris, dan kolom ditiadakan.

Identifikasi terhadap objek ini bergantung pada *bussiness rules* atau dapat kita katakan sebagai spesifikasi aplikasi. Berdasarkan informasi spesifikasi tersebut kemudian dibuat daftar objek yang mewakili. Untuk aplikasi web administrator ini spesifikasi yang dikehendaki adalah sebagai berikut:

1. Terdapat *top administrator* yang dapat mengelola administrator lain dengan mengelompokkan administrator pada kelompok hak akses dan atau modifikasi tertentu.
2. Dibutuhkan validasi administrator saat *login* ke dalam sistem.
3. Terdapat tingkat otoritas hak akses dan modifikasi sesuai dengan setting administrator yang dapat dilakukan secara dinamis.
4. Administrator juga dapat mengelola informasi sebaran, departemen dalam institusi kampus dan sistem pelaporan sesuai dengan hak akses yang dimilikinya.
5. Selain itu terdapat manajemen pengguna yang juga dapat dikelompokkan berdasarkan kategori tertentu yang relevan dalam institusi kampus, seperti jurusan, fakultas, kelompok studi, unit kegiatan mahasiswa, tim kepanitiaan, dll.
6. Tiap administrator yang terasosiasi dengan departemen tertentu hanya akan melihat laporan yang ditujukan ke departemen tersebut.
7. Terdapat *timestamp* pada tiap sirkulasi penambahan atau modifikasi suatu informasi.

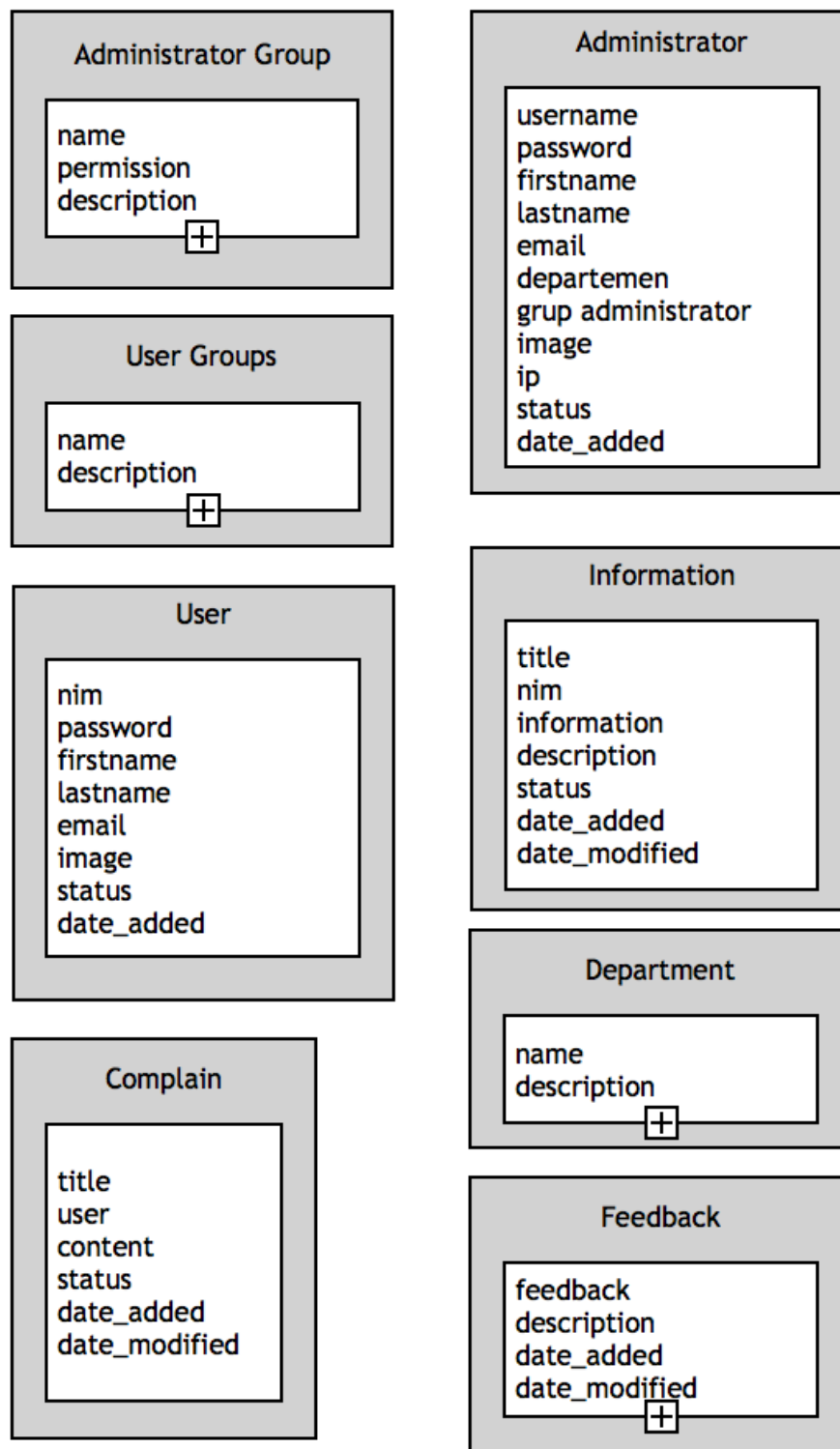
Dari *bussiness rules* diatas dapat dibuat sebuah objek seperti pada Gambar 3.25. Pada tahap ini semua entitas yang dibutuhkan oleh *database* yang merupakan bentuk *raw* atau tidak normal.



Gambar 3.25 Realisasi objek dari *bussiness rules*

Berdasarkan spesifikasi tersebut selanjutnya dapat diidentifikasi kebutuhan data apa saja yang harus disimpan dalam *database*. Daftar atribut tersebut kemudian dikelompokkan secara logikal berdasarkan relasi yang memungkinkan antar entitas yang merupakan bentuk normal pertama. Pengelompokkan ini akan menjadi bakal dari sebuah tabel. Sementara atribut yang ada dalam satu entitas akan menjadi kolom yang menyusun tabel. Pengelompokkan ini ditunjukkan pada Gambar 3.26.

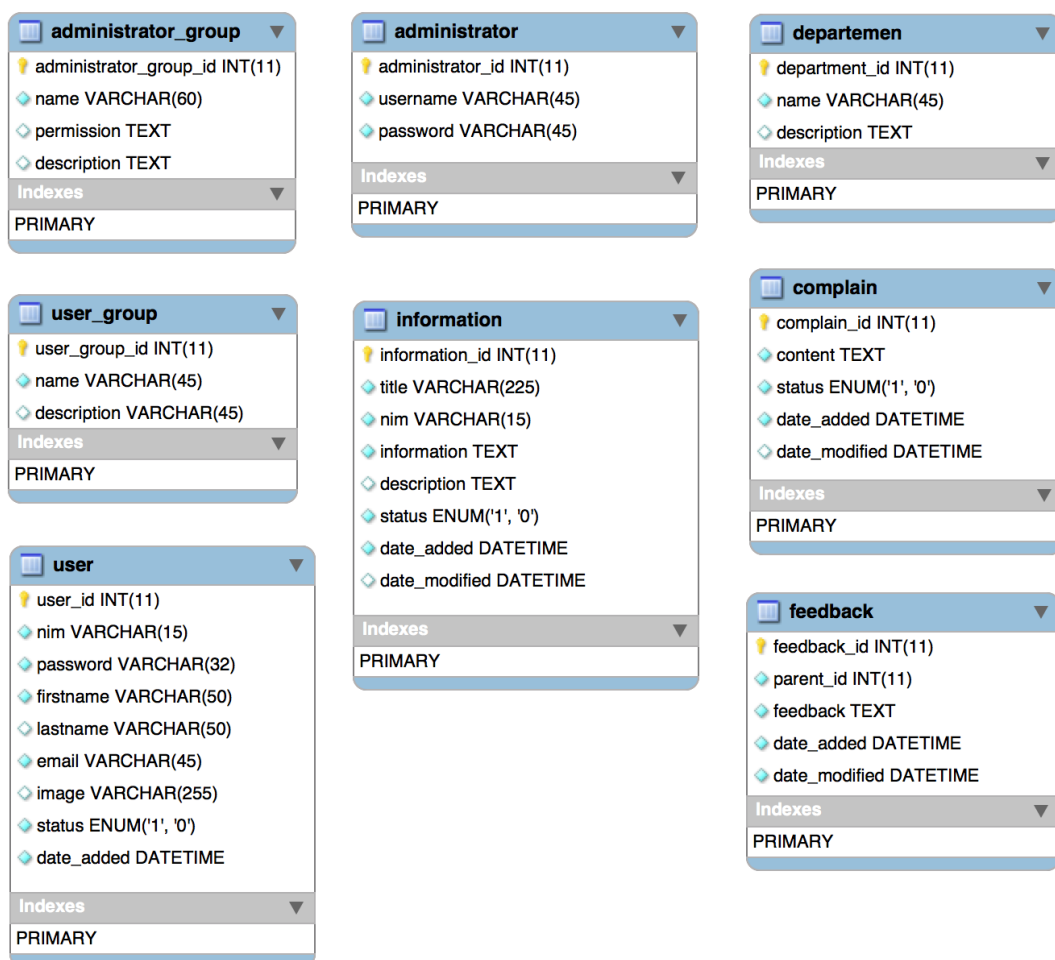




Gambar 3.26 Pengelompokkan properti dari entitas sesuai set data yang berelasi

### 3.4.3 Normalisasi Basis Data

Setelah memiliki kategorisasi entitas data maka dapat dilakukan dinormalisasi. Proses normalisasi ini akan membentuk tabel, kolom, dan *primary key* yang secara fisik dapat diimplementasikan pada *database*. Pada gambar sebelumnya terdapat enam entitas, yakni grup admin. Selanjutnya atribut dari ketiga entitas tersebut ditentukan tipe data dan *primari key* yang relevan seperti yang terlihat pada Gambar 3.27.

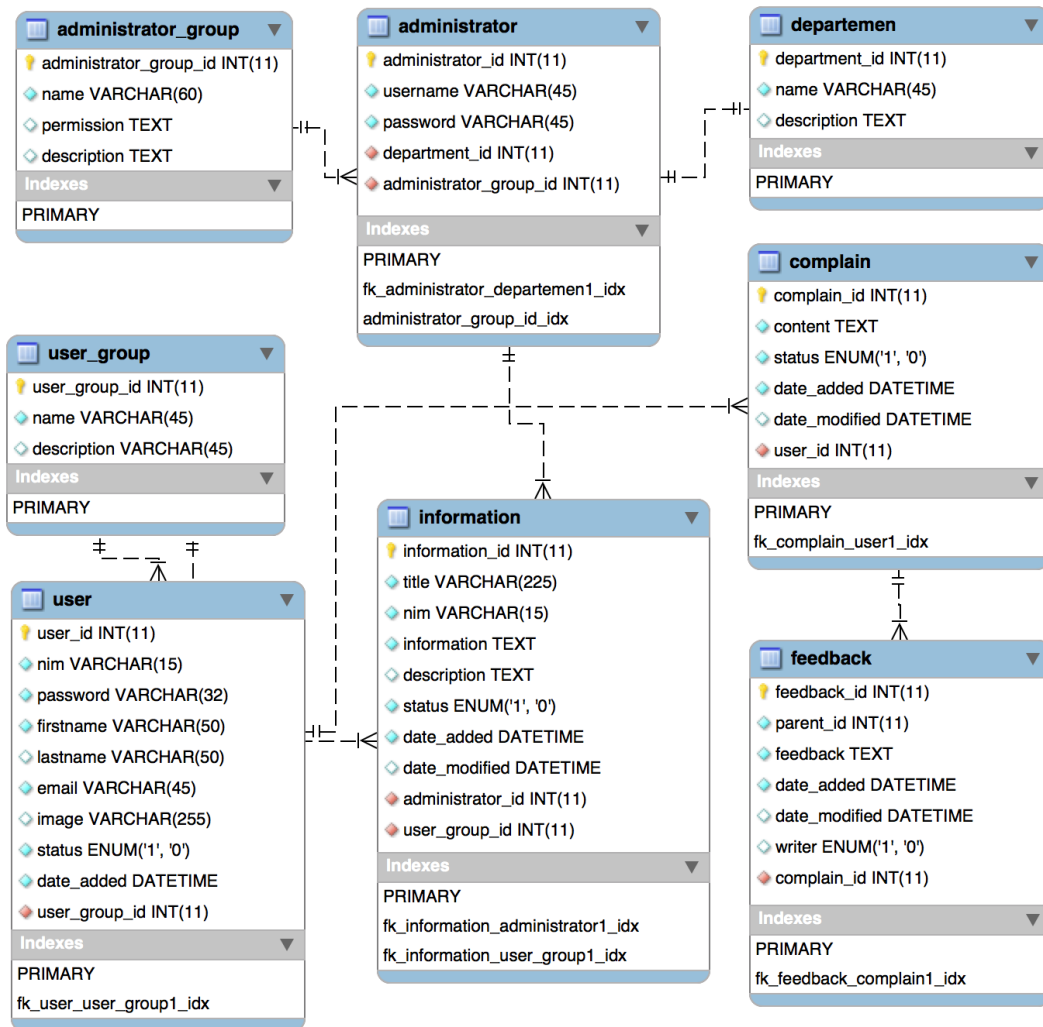


Gambar 3.27 Atribut yang telah diberi tipe data dan *primari key*

Pada tahap ini tabel yang telah terbentuk dicoba untuk dianalisis ketergantungannya dengan tabel lain sehingga timbul hubungan antar tabel

(*relationship*). Merujuk pada *bussiness rules* yang dibuat diawal maka relasi yang sesuai adalah seperti yang tampak pada Gambar 3.28.

Relasi yang timbul adalah *one to many*, *many to one* dan *one to one*. Relasi *many to one* terjadi pada tabel *administrator\_group* dan *administrator* dimana sebuah grup *administrator* dapat dirujuk oleh banyak baris dari administrator. Begitu juga pada tabel *user* dan *user\_group* relasi yang timbul adalah *one to many*. Relasi *one to one* pada tabel administrator dan departemen. Hal ini untuk memenuhi *bussiness rules* dimana administrator hanya dapat melihat laporan yang masuk jika memiliki akses terhadap halaman laporan dan jika telah terasosiasi dengan departemen tertentu maka laporan yang dapat dilihat adalah objek departemen terkait.



Gambar 3.28 Relasi antar tabel pada *database*

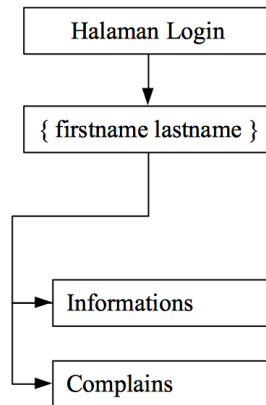
Relasi tabel diatas telah memenuhi bentuk normal yang ketiga, yakni bentuk normal kedua ditambah dengan pengadaan *foreign key*. Antara tabel *administrator\_group* dengan tabel *adminstrator* terhubung dengan *foreign key* *fk\_administrator\_administrator\_grup\_idx* dengan penambahan kolom *administrator\_group\_id* pada tabel *adminstrator*. Relasi *one to many* menyatakan bahwa sebuah grup administrator dapat berisi banyak administrator.

Sementara itu relasi tabel *user* dan *user\_group* mirip dengan tabel *administrator* yang dihubungkan oleh sebuah kolom *user\_group\_id*. Seorang *user* (pengguna) dapat melakukan banyak pelaporan sehingga relasi yang tepat adalah *one to many* untuk tabel *user* dan *complain*. Kemudian tabel *complain* berelasi *many to one* dengan tabel departemen yang menyatakan bahwa banyak laporan dapat ditujukan pada sebuah departemen. Tabel *information* berelasi *many to one* dengan *user\_group* dan *administrator*. Karena seorang administrator dapat menambahkan banyak informasi untuk satu target *user\_group*.

Tabel *complain* berelasi *many to one* dengan tabel *user* dan *department* karena sebuah seorang pengguna bisa memberikan banyak laporan ke sistem dan tiap laporan akan terasosiasi dengan sebuah departemen. Namun sebuah departemen dapat menerima banyak laporan dari pengguna yang sama pula. Kolom *parent\_id*. Sementara tabel *feedback* dari laporan pengguna ada di tabel lain karena sering kali dibutuhkan lebih dari satu jawaban kepada pengguna. Relasi tabel diatas telah cukup untuk mewakili struktur *database* secara fisik sehingga siap untuk diimplementasikan dengan penggunaan *SQL statement*.

### **3.5 Desain Masukan dan Keluaran Sistem Android Pengguna**

Aplikasi Android pengguna hanya akan memiliki beberapa fitur utama seperti *submit* laporan dan mengunduh informasi dari halaman web sehingga lebih sederhana. Selain itu juga terdapat halaman *login* pengguna sebelum memasuki aplikasi dan form registrasi agar pengguna dapat mendaftar akun menggunakan aplikasi Android yang disediakan. Pada aplikasi Android pengguna desain menu yang lebih sederhana ditunjukkan oleh Gambar 3.29.



Gambar 3.29 Desain menu halaman aplikasi Android pengguna

Kali pertama aplikasi dibuka, maka pengguna harus melakukan *login* untuk mengakses semua fungsi dalam aplikasi. Informasi yang diperlukan adalah NIM dan *password* yang ditambahkan sebelumnya menggunakan sistem informasi kampus yang lain. Validasi dari data ini membutuhkan akses terhadap *database* di aplikasi web. Setelah pengguna *login* dengan benar maka akan dibawa menuju halaman dashboard.

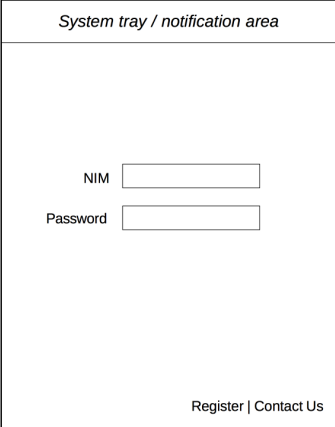
Halaman profil berisi rangkuman profil pengguna secara umum yang diakses melalui menu dengan nama pengguna. Pengguna hanya melihat informasi pasif di halaman ini. Pada halaman *Informations* pengguna akan melihat daftar informasi yang telah disebar oleh administrator. Sementara halaman *Complains* merupakan manajemen untuk pelaporan. Pengguna dapat menambahkan laporan baru dan memodifikasinya.

### 3.5.1 Desain Antarmuka

Desain antarmuka aplikasi Android secara umum lebih sederhana. Karena secara teknis semua jenis elemen dapat didefinisikan dengan mudah pada Android Studio.

#### 3.5.1.1 Desain Halaman *Login*

Halaman ini merupakan halaman yang dilihat pengguna setiap membuka aplikasi. Informasi yang diperlukan untuk *login* adalah NIM dan *password* yang telah dimasukkan ketika registrasi. Desain halaman ini ditunjukkan oleh Gambar 3.30.



System tray / notification area

NIM

Password

Register | Contact Us

Gambar 3.30 Desain halaman *login* pada aplikasi Android

### 3.5.1.2 Desain Navigasi Aplikasi

*User interface* yang tipikal ada di versi Android adalah *folding navigation* yang ada dibagian kiri halaman yang dapat dibuka dan tutup dengan input sentuhan pada *icon* navigasi. Lebih detil desain menu ini dapat dilihat pada gambar 3.31.

|  |  |
|--|--|
| <i>System tray / notification area</i> |  |
| {firstname lastname}                   |  |
| Informations                           |  |
| Complans                               |  |
| Log Out                                |  |
|  |  |

Gambar 3.31 Desain navigasi aplikasi Android pengguna

### 3.5.1.3 Desain Halaman Registrasi

Halaman ini digunakan untuk registrasi pengguna agar masuk sebagai daftar pengguna yang tampil di sistem aplikasi *web*. Informasi yang dikumpulkan dari *form* registrasi ini harus memuat NIM, password, jurusan dan fakultas. Detail desain halaman ini terdapat pada Gambar 3.32.



System tray / notification area

Nama Depan

Nama Belakang

NIM

Password

Jurusan  ▼

Fakultas  ▼

**REGISTER**

Gambar 3.32 Desain halaman registrasi pada aplikasi Android

#### 3.5.1.4 Desain Halaman Laporan

Halaman ini secara khusus digunakan untuk mengumpulkan laporan kepada biro/departemen tertentu yang telah dimasukkan sebelumnya di sistem aplikasi web. Informasi yang dikumpulkan pada *form* ini akan disimpan dalam *database* aplikasi Android dan sistem aplikasi web. Terdapat kolom judul, isi laporan dan biro/departemen yang relevan untuk menerima suatu laporan. Desain halaman ini ditunjukkan pada Gambar 3.33.

*System tray / notification area*

Laporan

Judul

Deskripsi Laporan

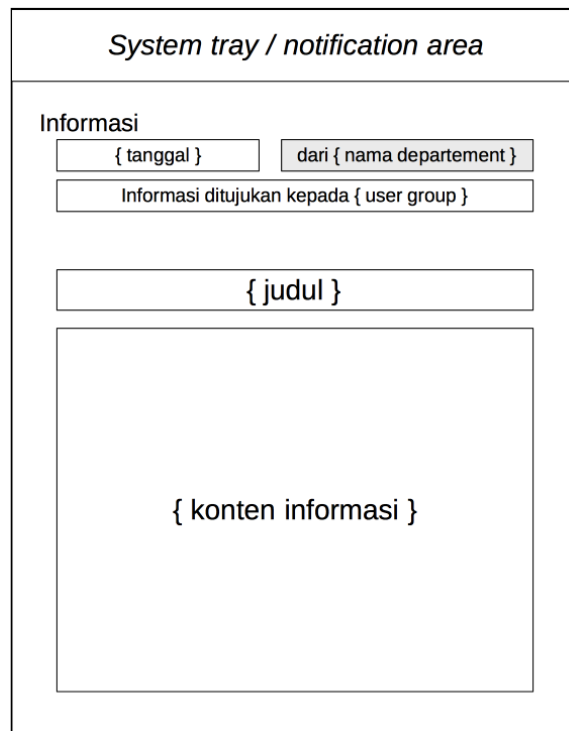
Penerima  ▼

**SUBMIT**

Gambar 3.33 Desain halaman registrasi pada aplikasi Android

### 3.5.1.5 Desain Halaman Informasi

Sesuai namanya halaman ini digunakan untuk melihat informasi sebaran yang diambil dari sistem aplikasi web. Pengambilan data ini dilakukan saat pengguna membuat aplikasi. Halaman ini akan berisi teks informasi dan gambar yang disebarakan administrator melalui sistem aplikasi *web*. Selain itu juga terdapat informasi tanggal, departemen yang menyebarkan informasi dan kelompok pengguna yang menerima informasi. Hal ini dimaksudkan sebagai informasi tambahan kepada pengguna agar tidak terjadi *misleading* terhadap informasi yang dibaca. Detail desain halaman ini ditunjukkan oleh Gambar 3.34.



Gambar 3.34 Desain halaman detail informasi

### 3.5.2 Local Storage

Penggunaan *local storage* juga berlaku pada *development* aplikasi Android menggunakan *PhoneGap*. Pada aplikasi Android pengguna *local storage* dapat digunakan untuk menyimpan data pengguna secara *offline*. Data yang disimpan secara offline tentu terbatas pada beberapa data penting pengguna seperti data profil, yakni: nama, nim, password dan *email*.

### 3.6 Penggunaan SDLC

Dalam pembuatan tugas akhir ini haruslah dipilih SDLC yang tepat sesuai kebutuhan pengembangan sistem dan kaidah ilmiah dalam penulisan laporannya. Dalam hal ini dipilih model SDLC *Waterfall* untuk beberapa alasan berikut ini.

- Tahapan proses dimulai dengan perancangan kebutuhan dan desain di awal.
- Proses dan hasil telah ditetapkan sehingga dapat diuji apakah telah sesuai dengan spesifikasi yang diinginkan atau tidak.

- *Milestone* yang jelas dan kemudahan dalam menyusun kerangka kerja.
- Setiap fase kerja dikerjakan dan diselesaikan satu kali tiap waktu.
- Tahapan proses yang serial dan tidak *overlapping* satu sama lain sehingga sesuai dengan kaidah penulisan ilmiah.

Lebih dari itu dengan penggunaan model *Waterfall* ini akan terdapat dokumentasi yang baik dari sistem yang dibuat. Sehingga memungkinkan untuk pengembangan dimasa mendatang.

- Diperlukan adanya sistem notifikator pada aplikasi web administrasi untuk tiap *event* yang berhubungan dengan fungsi aplikasi. Seperti saat ada pengguna yang melakukan registrasi, ada komplain yang masuk dan ada *feedback* dari komplain.