

## **BAB III**

### **ANALISA DAN PERANCANGAN SISTEM**

Dalam pembuatan aplikasi ini menerapkan konsep SDLC (*Systems Development Life Cycle* (Siklus Hidup Pengembangan Sistem) yang berfungsi untuk menggambarkan tahapan-tahapan utama dan langkah-langkah dari setiap tahapan. Langkah-langkah yang akan dilakukan dalam pembuatan Langkah-langkah yang akan dilakukan dalam pembuatan Aplikasi Panduan Lokasi dan Informasi Perguruan Tinggi di Surabaya berbasis Android yaitu sebagai berikut:

#### **3.1 Analisis Permasalahan**

##### **3.1.1 Identifikasi masalah**

Seleksi Nasional Masuk Perguruan Tinggi Negeri (SNMPTN) merupakan jalur yang paling banyak diambil oleh siswa setelah lulus dari SMA. Untuk menembus perguruan tinggi negeri memang diperlukan kerja keras. Jalan satu-satunya yaitu dengan belajar ekstra. Sulit rasanya untuk kita menjawab soal apabila tidak melakukan banyak latihan menjawab soal yang mirip.

Sebanyak 618.804 peserta SNMPTN 2012 yang bersaing untuk mendapatkan kursi di perguruan tinggi negeri. Pasalnya siswa harus memperebutkan 106.363 kursi di perguruan tinggi negeri. Total pendaftar ujian SNMPTN ini meningkat 14 % dari tahun sebelumnya. Pada 2011 total pendaftar sebanyak 540.953, sedangkan tahun 2012 ini meningkat menjadi 618.804 peserta. Yang terdiri dari 570.771 pendaftar reguler dan 48.032 pendaftar beasiswa bidik misi. Perguruan tinggi negeri yang berpartisipasi dalam SNMPTN sebanyak 61

PTN dengan 2.340 program studi. Yakni 54 PTN dari Kemendikbud dengan 2.123 program studi dan tujuh PTN di bawah Kementerian Agama dengan 217 program studi (Kompas, 2012). Salah satu perguruan tinggi negeri yang paling banyak diminati terdapat di kota Surabaya.

Surabaya merupakan salah satu kota terbesar di Indonesia. Selain sebagai ibukota Jawa Timur, Surabaya menjadi pusat perkembangan politik, ekonomi, pendidikan, budaya maupun industry khususnya di Indonesia bagian timur. Sebagai pusat perkembangan pendidikan, Surabaya memiliki sejumlah perguruan tinggi negeri maupun swasta yang ternama.

Buktinya pada tahun 2012, Surabaya menyumbangkan 3 pada 20 besar peringkat Perguruan Tinggi terbaik di Indonesia. Diantaranya adalah Institut Teknologi Sepuluh Nopember (ITS) yang berada di peringkat ke 3, Universitas Airlangga berada di peringkat ke 11 dan Universitas Kristen Petra berada di peringkat 20 (*Webometrics Ranking of World Universities*, 2012).

Di Jawa Timur sendiri, Perguruan Tinggi di Surabaya menempati urutan pertama dalam hal jumlah terbanyak di banding dengan kota-kota lainnya. Surabaya, total memiliki 83 Perguruan Tinggi yang terdiri dari 78 Perguruan Tinggi Swasta dan 5 Perguruan Tinggi Negeri (DIKTI, 2011).

Keadaan inilah yang menarik minat para pelajar baik dari dalam kota maupun luar kota untuk menjadi mahasiswa dari salah satu perguruan tinggi yang ada di Surabaya. Akan tetapi masih banyak pelajar yang kesulitan untuk mengetahui informasi mengenai data dan lokasi Perguruan Tinggi di Surabaya. Memang beberapa Perguruan Tinggi baik negeri ataupun swasta di kota Surabaya

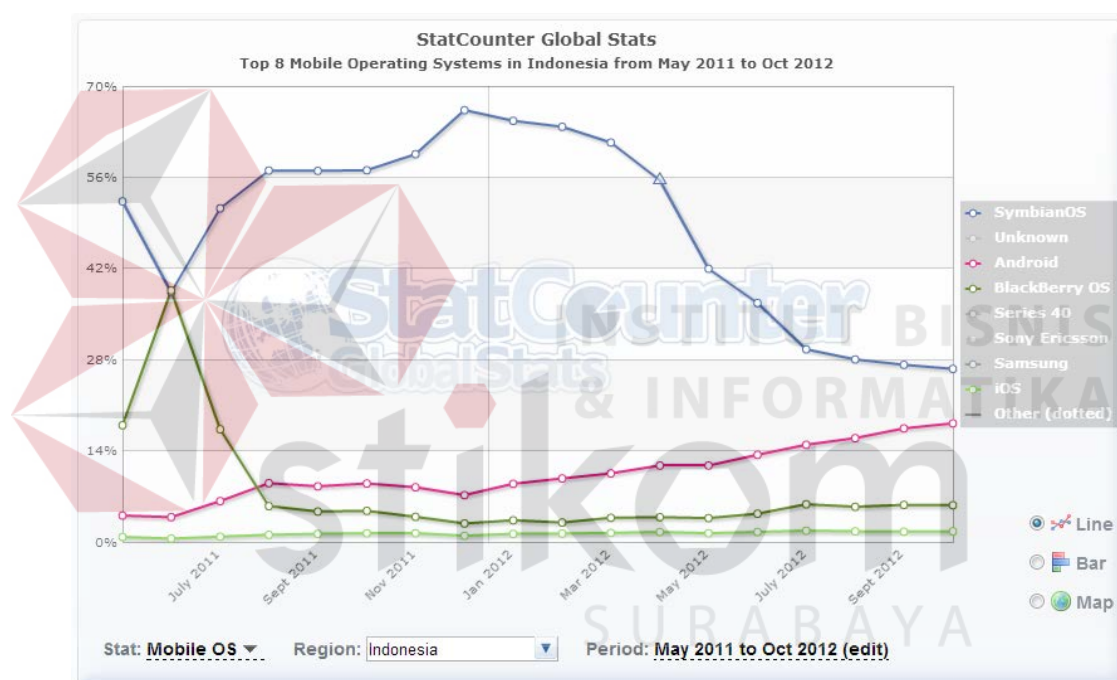
sebagian sudah memiliki situs resmi akan tetapi masih banyak juga yang belum memiliki situs resmi (DIKTI, 2011). Pelajar juga kebingungan untuk mencari situs-situs resmi tersebut dikarenakan tidak mengetahui Perguruan Tinggi mana saja yang menyediakan program studi yang diinginkan. Kemudian, didalam situs-situs tersebut hanya berisi informasi dari segi profil saja seperti alamat, daftar program studi ataupun prestasi yang telah di capai tanpa adanya informasi secara visual yang dapat membantu para pelajar mengetahui posisi kampus dari perguruan tinggi yang ada.

### 3.1.2 Identifikasi Kebutuhan

Berdasarkan identifikasi permasalahan, dapat disimpulkan bahwa diperlukan Mobile GIS yang dapat menampilkan data spasial seperti profil kemudian data tabular seperti lokasi dari Perguruan Tinggi tersebut di Surabaya, dalam hal ini aplikasi yang dibuat menggunakan perangkat *mobile smartphone android*. Seperti yang kita ketahui, saat ini *Smartphone Android* sedang mengalami peningkatan yang sangat pesat. Dan ada beberapa faktor penting yang menyebabkan peningkatan pengguna *Android* yaitu harga *handset* yang cukup terjangkau sehingga *handset Android* bisa dengan mudah dimiliki oleh berbagai kalangan, tampilan antar muka yang cukup menarik dan tidak membosankan, dan yang terakhir adalah dikarenakan sistem operasi *Android* yang digunakan oleh banyak produsen telepon genggam seperti Sony, Samsung, Motorola dan bahkan merk lokal juga menggunakan sistem operasi *Android*.

Seperti yang dimuat di Harian Seputar Indonesia pada tanggal 27 Juni 2012, Gideon Edi Purnomo selaku Head of VAS, Applications and Device Management Group Telkomsel menyebutkan bahwa pertumbuhan pengguna

Android mencapai 15 kali lipat dibandingkan tahun 2011 atau sekitar 2,5 juta pengguna saat ini dan diperkirakan akan terus mengalami peningkatan lebih besar pada tahun depan. dimana solusi alternatif tersebut didapat dari pertimbangan jarak dan biaya. Hal ini juga dapat dilihat dari hasil survey *StatCounter Global State* yang menunjukkan jumlah peningkatan pengguna *Smartphone Android* pada periode Mei 2011 hingga Oktober 2012 yang cukup signifikan seperti yang terlihat pada gambar berikut.



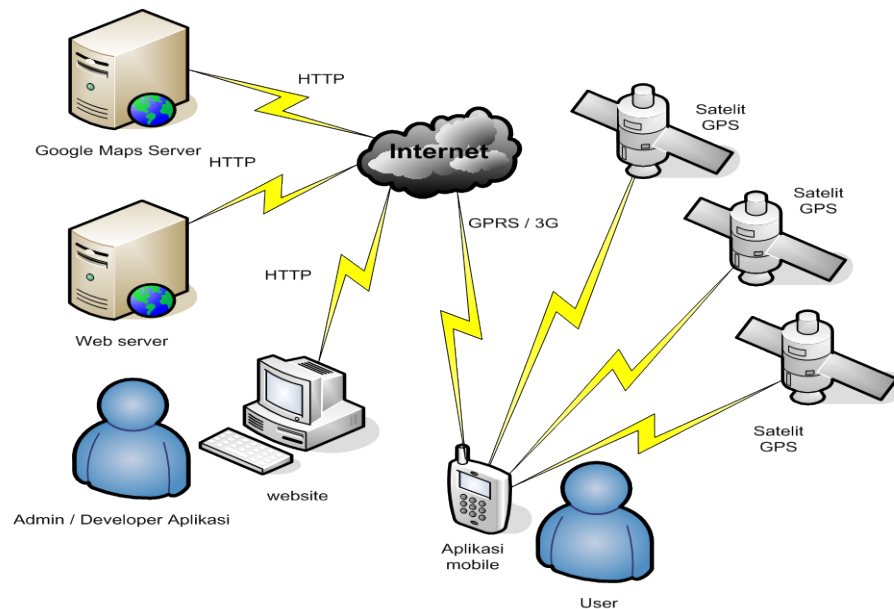
Gambar 3.1 Grafik Peningkatan Pengguna *Smartphone* Berdasarkan *Mobile Operating System*

*Smartphone Android* sendiri di dalamnya sudah tertanam mobile GIS yang merupakan sebuah integrasi cara kerja perangkat lunak/keras untuk pengaksesan data dan layanan geospasial melalui perangkat bergerak via jaringan kabel atau nirkabel.

Aplikasi akan menampilkan secara spesifik apa yang dibutuhkan para pelajar, karena terdapat pilihan yang menu yang disesuaikan terhadap kebutuhan pencarian informasi tentang perguruan tinggi. Kemudian dengan menggunakan Mobile GIS dalam sistem operasi Android, dapat mempermudah para pelajar dalam hal navigasi yang berkaitan dengan pengaksesan informasi dan lokasi perguruan tinggi yang ada di Surabaya. Dengan memanfaatkan Android sebagai sistem operasi yang open source serta tersedianya layanan peta berbasis Google Maps yang dapat digunakan secara gratis dan juga tidak perlu membuat server khusus untuk mengelola data yang berkaitan dengan sebuah peta virtual.

### **3.2 Rancangan Sistem**

Pada pengembangan sistem ini akan dibuat suatu yang diharapkan dapat membantu para pengguna dalam memberikan informasi tentang Perguruan Tinggi yang diinginkan, range area yang dimaksud adalah memberikan informasi tentang Perguruan Tinggi yang berada dekat pada pengguna, dan memberikan informasi tentang Program Studi beserta nilai akreditasi pada masing-masing Perguruan Tinggi. Pada bab ini model pengembangan dalam sistem informasi berupa Aplikasi Panduan Perguruan Tinggi pada Smartphone Android. Model pengembangan yang akan digunakan berupa workflow sebagaimana terlihat pada gambar di bawah ini.

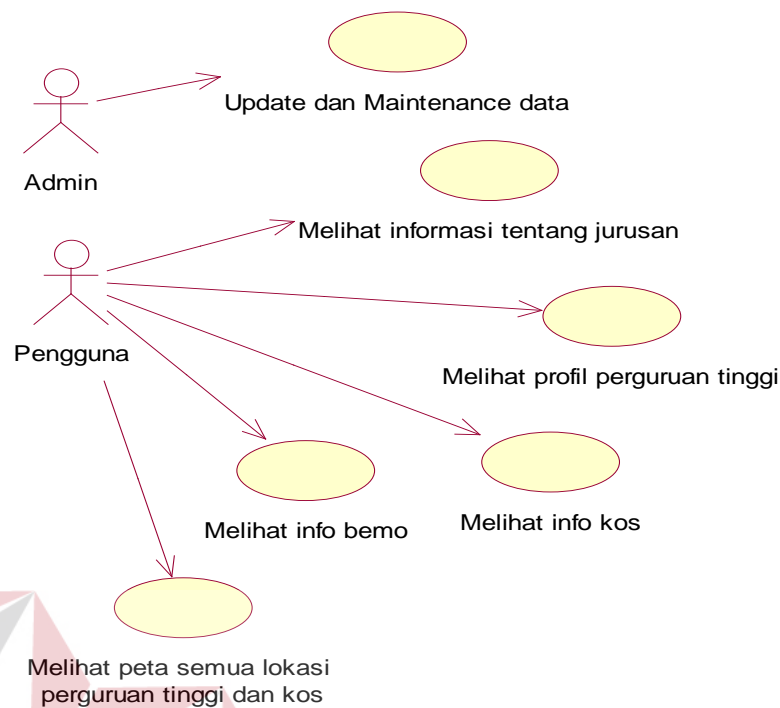


Gambar 3.2 Work Flow Aplikasi Panduan Informasi dan Lokasi Perguruan Tinggi di Surabaya berbasis *Android*

Pada saat aplikasi dijalankan, dengan menggunakan Layanan Berbasis Lokasi (LBS) sistem akan secara otomatis menangkap lokasi *user*, namun untuk dapat menggunakan layanan ini perangkat *mobile* atau handphone harus terkoneksi dengan internet atau dengan menyalakan fitur GPS yang ada di handphone. Lokasi akan di tampilkan secara visual berupa tampilan *map* dari lokasi *user* berada sampai ke Perguruan Tinggi yang dituju atau biasa di sebut juga *GPS Tracking*. Tidak hanya menampilkan perguruan tinggi yang di tuju saja akan tetapi apabila rute tersebut melewati beberapa perguruan tinggi lain maka secara otomatis *map* juga akan menampilkannya.

### 3.3 Use Case Diagram

*Use case diagram* digunakan untuk menspesifikan apa yang dapat dilakukan oleh sistem atau untuk menspesifikan kebutuhan fungsional utama dari sistem. Berikut akan dijelaskan *use case diagram* untuk sistem.



Gambar 3.3. *Use Case Diagram* Aplikasi Panduan Informasi dan Lokasi

Perguruan Tinggi di Surabaya berbasis *Android*

Berikut adalah penjelasan singkat *use case* yang dimiliki oleh aplikasi.

Tabel 3.1 Penjelasan singkat *use case diagram*

Use Case	Keterangan
Melihat informasi tentang jurusan	Proses ini ditujukan kepada pengguna yang masih bingung informasi tentang program studi yang sesuai dengan mereka.
Melihat Profil Perguruan Tinggi	Proses ini berisi informasi lengkap tentang Perguruan Tinggi seperti profil, alamat, daftar program studi beserta akreditasinya.
Melihat Peta Lokasi Perguruan Tinggi dan Kos	Proses ini menampilkan lokasi semua perguruan tinggi dan kos akan tetapi terdapat fitur yg bisa membatasi apa saja yg yang akan ditampilkan .
Melihat Profil Kos	Proses ini berisi informasi tentang Kos sesuai dengan pencarian berdasarkan perguruan tinggi .
Melihat Jalur Bemo	Proses ini berisi informasi tentang jalur bemo sesuai dengan pencarian berdsarkan perguruan tinggi jg.
Update dan Maintenance Data	Proses ini di lakukan oleh Admin untuk proses mengisi data sampai proses update data.

Aplikasi ini nantinya akan menyediakan beberapa menu diantaranya, menu untuk memberikan rekomendasi program studi yang sesuai maksudnya adalah menu ini ditujukan kepada pelajar yang masih bingung memilih program studi yang sesuai dengan mereka. Lalu menu untuk mencari informasi Perguruan Tinggi yang diinginkan, output yang dikeluarkan adalah informasi lengkap tentang Perguruan Tinggi seperti profil, alamat, daftar program studi beserta akreditasinya. Kemudian, menu yang berisi pencarian Perguruan Tinggi berdasarkan kriteria program studi yang diinginkan, misalkan program studi yang diinginkan “Teknik Industri” maka secara otomatis akan keluar output berupa Perguruan Tinggi mana saja yang menyediakan program studi tersebut.

### 3.4 Activity Diagram

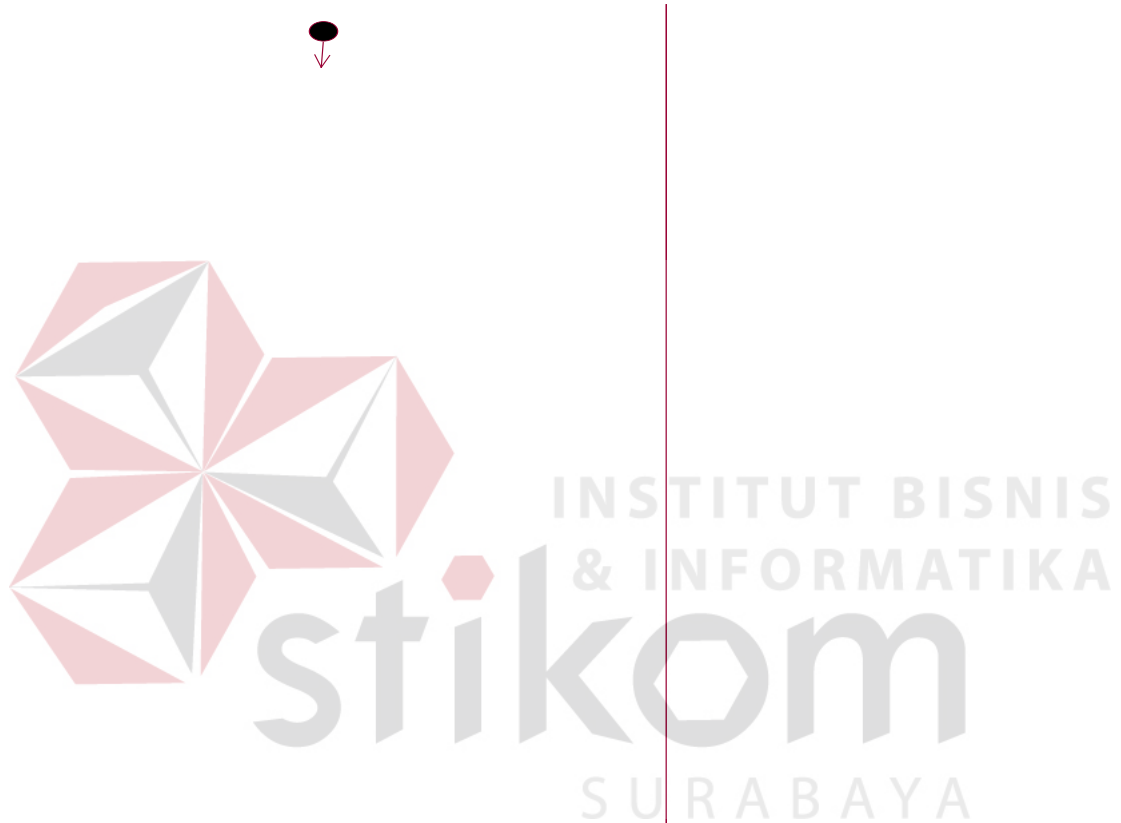
*Activity Diagram* digunakan untuk memodelkan aliran kerja proses dalam bentuk simbol untuk menspesifikasikan bagaimana sistem akan mencapai tujuan. *Activity Diagram* adalah salah satu bentuk diagram UML yang paling mudah dimengerti dikarenakan diagram ini memiliki simbol yang menyerupai simbol *flowchart*, yang sangat berguna untuk menerangkan langkah-langkah proses ke pihak lain.

#### A. Activity Diagram untuk Proses Pencarian Jurusan

Proses Pencarian Jurusan dimulai dengan membuka halaman *home* dan memilih menu Panduan Jurusan. Pencariannya dilakukan dengan memasukkan filter-filter yang diinginkan. Setelah obyek dimasukkan, maka sistem akan melakukan pencarian pada *database*. Jika ada data yang sesuai dengan obyek yang dicari maka sistem akan mengirimkan semua data yang sesuai dengan filter-



filter pencarian yang diinginkan. Setelah menemukan jurusan yang diinginkan maka pengguna dapat melihat data detailnya dengan mengklik salah satu jurusan yang diinginkan tersebut. Untuk penjelasan proses Pencarian Jurusan ini dapat dilihat pada Gambar 3.4.

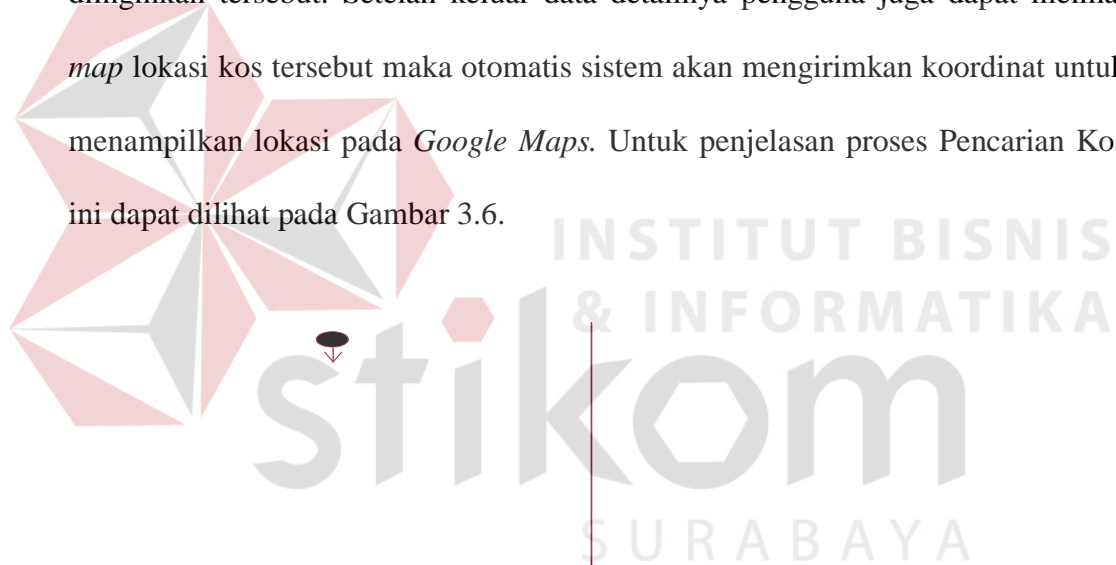


sistem akan melakukan pencarian pada *database*. Jika ada data yang sesuai dengan obyek yang dicari maka sistem akan mengirimkan semua data yang sesuai dengan filter-filter pencarian yang diinginkan. Setelah menemukan perguruan tinggi yang diinginkan maka pengguna dapat melihat data detailnya dengan mengklik salah satu perguruan tinggi yang diinginkan tersebut. Setelah keluar data detailnya pengguna juga dapat melihat *map* lokasi perguruan tinggi tersebut maka otomatis sistem akan mengirimkan koordinat untuk menampilkan lokasi pada *Google Maps*. Untuk penjelasan proses Pencarian Perguruan Tinggi ini dapat dilihat pada Gambar 3.5.



### C. *Activity Diagram* untuk Pencarian Kos

Proses Pencarian Kos dimulai dengan membuka halaman *home* dan memilih menu Info Kos. Pencariannya dilakukan dengan memasukkan filter-filter yang diinginkan. Setelah obyek dimasukkan, maka sistem akan melakukan pencarian pada *database*. Jika ada data yang sesuai dengan obyek yang dicari maka sistem akan mengirimkan semua data yang sesuai dengan filter-filter pencarian yang diinginkan. Setelah menemukan kos yang diinginkan maka pengguna dapat melihat data detailnya dengan mengklik salah satu kos yang diinginkan tersebut. Setelah keluar data detailnya pengguna juga dapat melihat *map* lokasi kos tersebut maka otomatis sistem akan mengirimkan koordinat untuk menampilkan lokasi pada *Google Maps*. Untuk penjelasan proses Pencarian Kos ini dapat dilihat pada Gambar 3.6.



#### D. *Activity Diagram* untuk Proses Pencarian Bemo

Proses Pencarian Bemo dimulai dengan membuka halaman *home* dan memilih menu Info Bemo. Pencariannya dilakukan dengan memasukkan filter-filter yang diinginkan. Setelah obyek dimasukkan, maka sistem akan melakukan pencarian pada *database*. Jika ada data yang sesuai dengan obyek yang dicari maka sistem akan mengirimkan semua data yang sesuai dengan filter-filter pencarian yang diinginkan. Setelah menemukan bemo yang diinginkan maka pengguna dapat melihat data detailnya dengan mengklik salah satu bemo yang diinginkan tersebut. Untuk penjelasan proses Pencarian Bemo ini dapat dilihat pada Gambar 3.7.



### E. *Activity Diagram* untuk Proses Maintenance Data

Proses *Maintenance* Data adalah proses yang melakukan manipulasi data (*insert, update, and delete*) untuk semua data yang terdiri dari: Master perguruan tinggi, alamat perguruan tinggi, prodi perguruan tinggi, detail prodi, data kos, data bemo. Proses ini dapat dilakukan hanya bagi *admin* yang telah terdaftar di database. Untuk lebih jelas bisa dilihat pada Gambar 3.8.

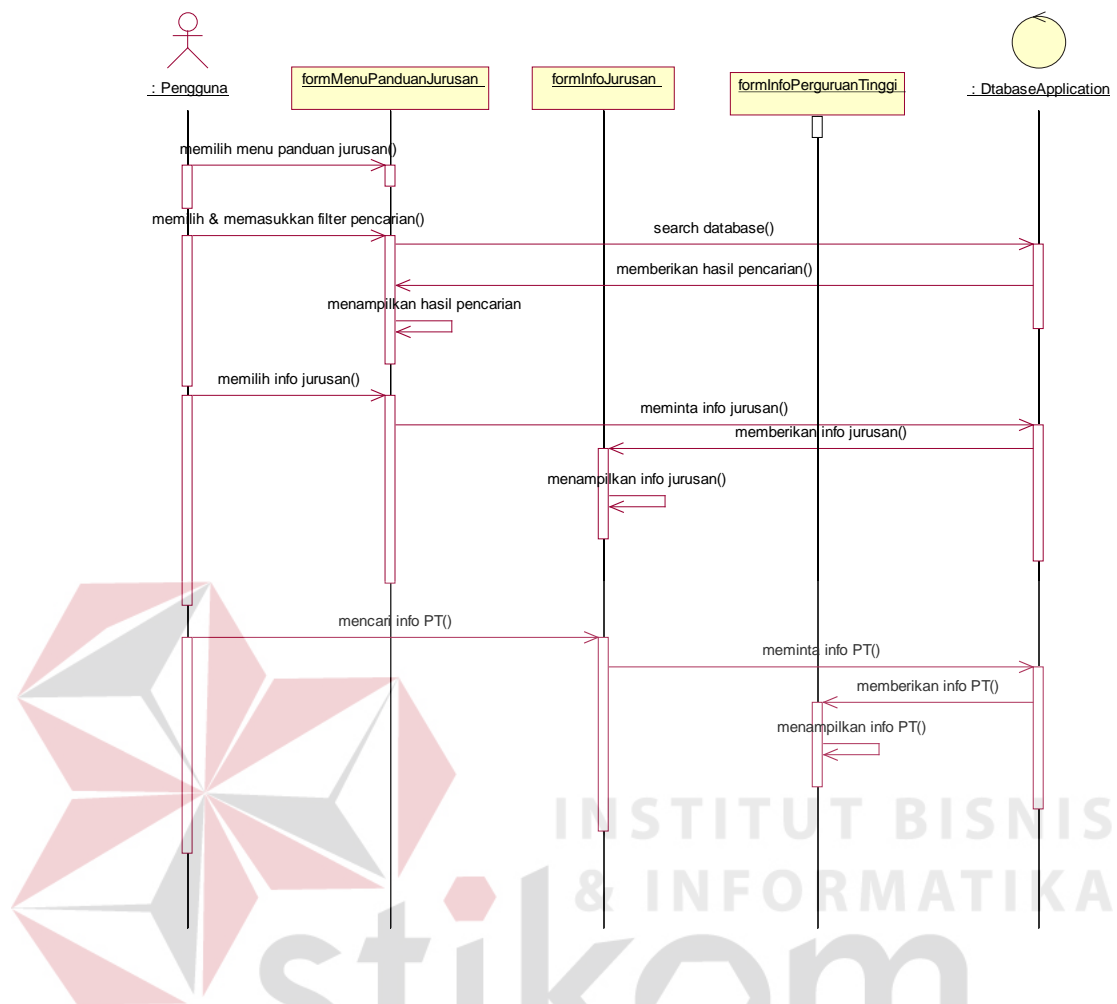


### 3.5 Sequence Diagram

*Sequece Diagram* digunakan untuk menggambarkan interaksi antar objek berdasarkan urutan waktu yang digambarkan dari atas ke bawah.

#### A. Sequence Diagram untuk Proses Pencarian Jurusan

Proses pencarian jurusan dimulai ketika *pengguna* memilih menu pencarian pada halaman awal menu panduan jurusan. Di halaman pencarian jurusan terdapat menu pencarian *advance*. Pencarian *advance* adalah pencarian dengan menggunakan beberapa filter yang diinginkan sehingga akan menampilkan jurusan-jurusan yang sesuai dengan kriteria pencari. Pada halaman pencarian *advance* ini akan terdapat beberapa filter yang selanjutnya diisi sesuai keinginan pengguna. Filter-filter yang diisi selanjutnya akan dicocokkan dengan data pada *database*. Jika data ditemukan maka *database* akan memberikan hasil pencarian yang selanjutnya akan ditampilkan oleh sistem. Apabila pengguna yang ingin melihat *detail* info jurusan tersebut dapat mengklik salah satu hasil pencarian tadi dan sistem akan menampilkan *detail* data dari jurusan tersebut. Di dalam detail tersebut juga terdapat perguruan tinggi mana saja yang menyediakan jurusan yang dicari pengguna. Dan apabila pengguna mengklik salah satu perguruan tinggi tersebut maka otomatis akan menampilkan info perguruan tinggi yang ada di menu profil perguruan tinggi. Untuk lebih jelasnya lagi dapat dilihat pada Gambar 3.9.

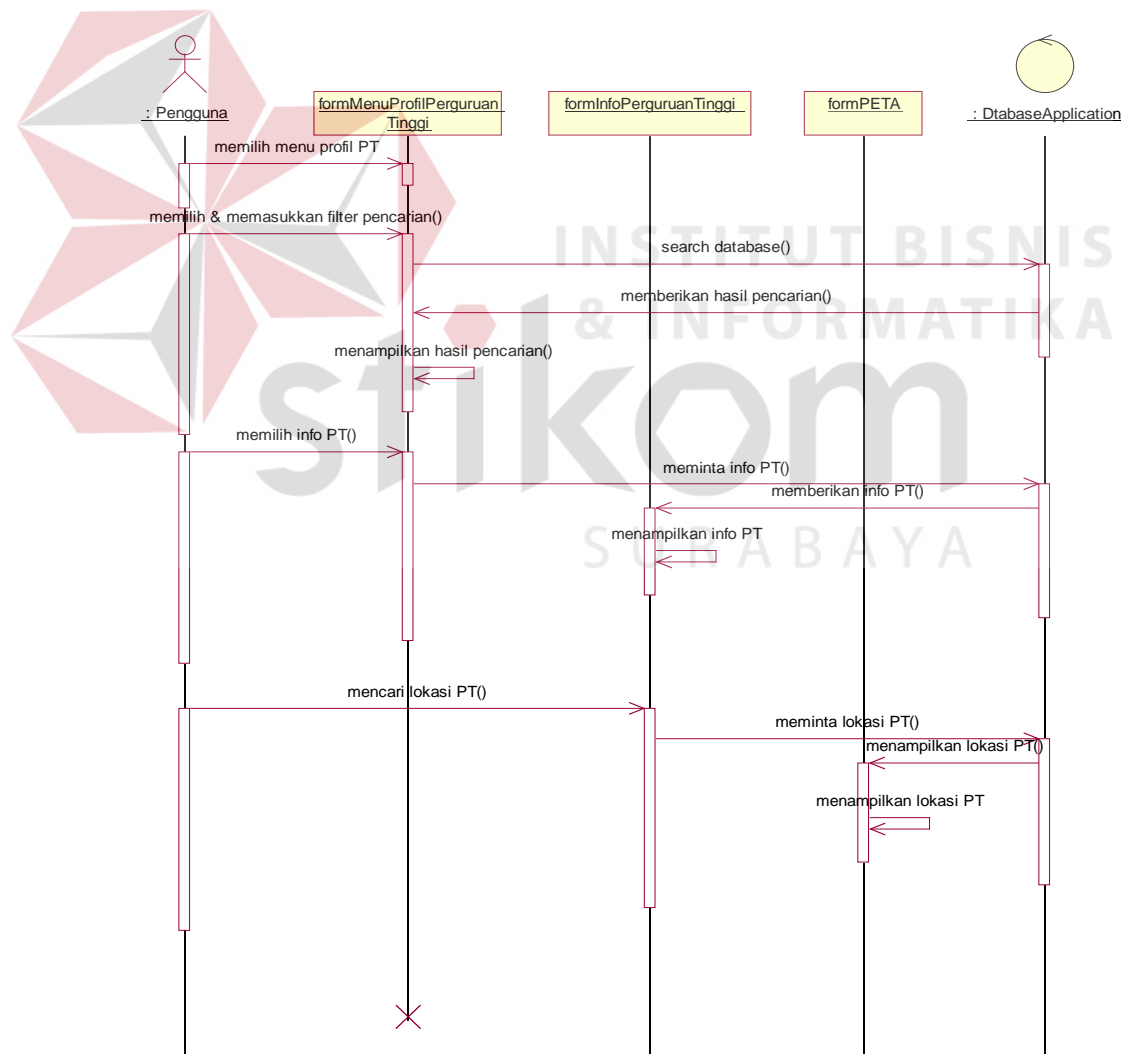


Gambar 3.9 Sequence diagram untuk proses “Cari Jurusan”

### B. Sequence Diagram untuk Proses Pencarian Perguruan Tinggi

Proses pencarian perguruan tinggi dimulai ketika pengguna memilih menu pencarian pada halaman awal menu profil perguruan tinggi. Di halaman pencarian perguruan tinggi tersebut terdapat menu pencarian *advance*. Pencarian *advance* adalah pencarian dengan menggunakan beberapa filter yang diinginkan sehingga akan menampilkan perguruan tinggi yang sesuai dengan kriteria pencari. Pada halaman pencarian *advance* ini akan terdapat beberapa filter yang selanjutnya diisikan sesuai keinginan pengguna. Filter-filter yang diisikan selanjutnya akan

dicocokkan dengan data pada *database*. Jika data ditemukan maka *database* akan memberikan hasil pencarian yang selanjutnya akan ditampilkan oleh sistem. Apabila pengguna yang ingin melihat *detail* info profil perguruan tinggi tersebut dapat mengklik salah satu hasil pencarian tadi dan sistem akan menampilkan *detail* data dari perguruan tinggi tersebut. Di dalam detail tersebut juga terdapat alamat perguruan tinggi yang apabila pengguna mengklik alamatnya maka otomatis akan menampilkan *map* lokasi perguruan tinggi tersebut. Untuk lebih jelasnya lagi dapat dilihat pada Gambar 3.10.

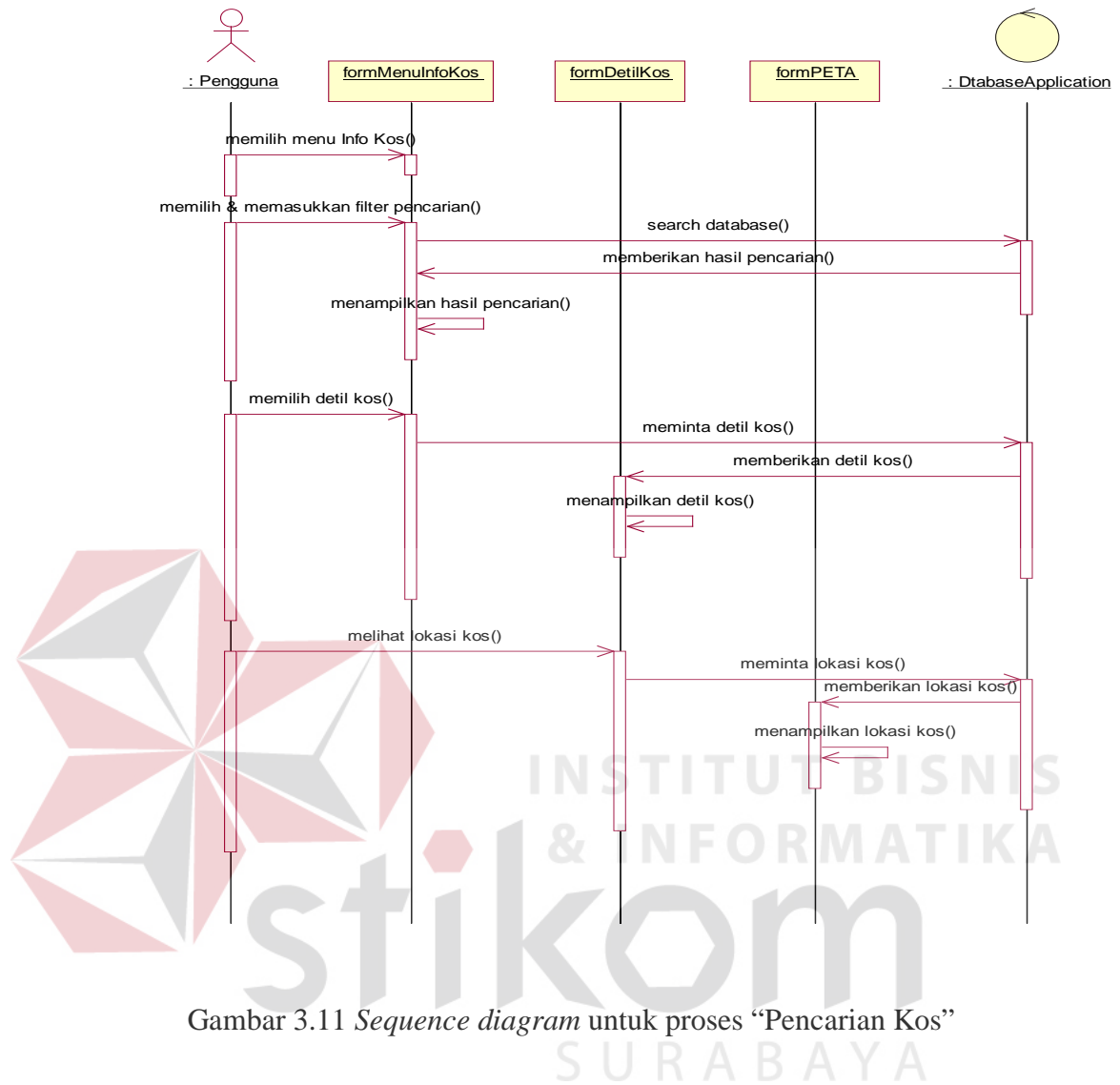


Gambar 3.10 *Sequence diagram* untuk proses “Cari Perguruan Tinggi”



### C. *Sequence Diagram* untuk Proses Pencarian Kos

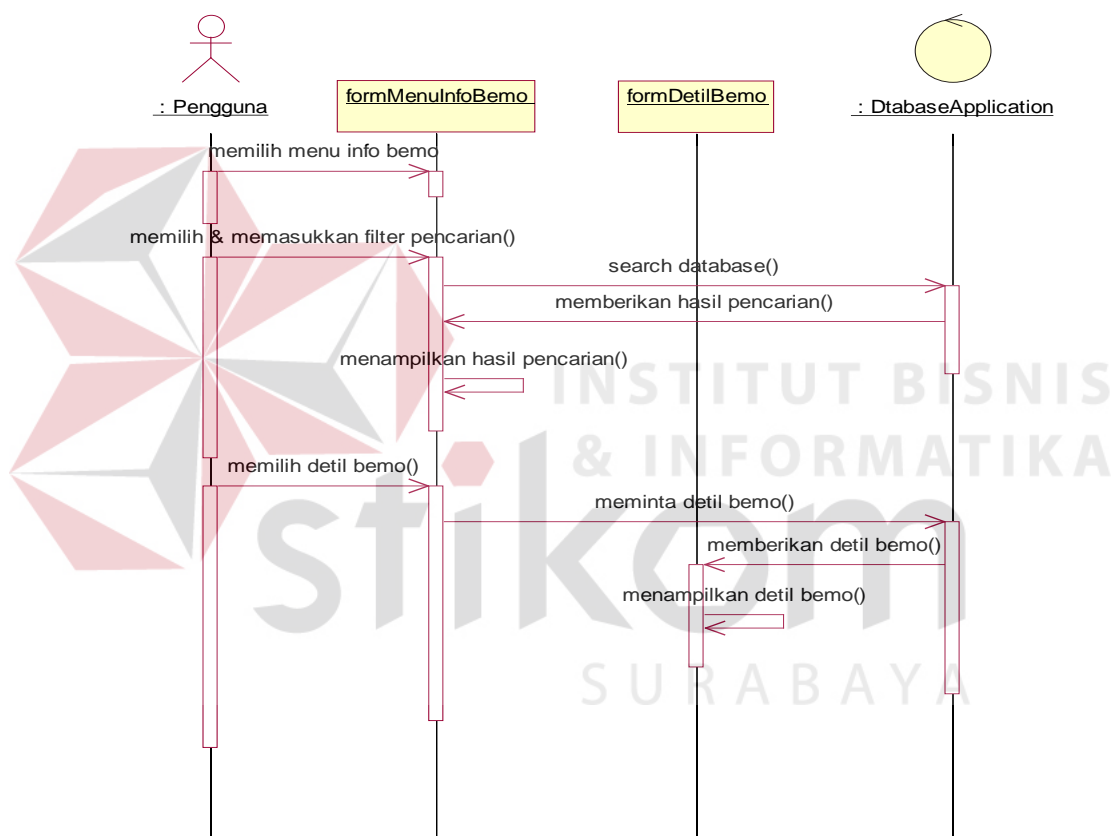
Proses pencarian kos dimulai ketika pengguna memilih menu pencarian pada halaman awal menu info kos. Di halaman pencarian kos tersebut terdapat menu pencarian *advance*. Pencarian *advance* adalah pencarian dengan menggunakan beberapa filter yang diinginkan sehingga akan menampilkan kos yang sesuai dengan kriteria pencari. Pada halaman pencarian *advance* ini akan terdapat beberapa filter yang selanjutnya diisikan sesuai keinginan pengguna. Filter-filter yang diisikan selanjutnya akan dicocokkan dengan data pada *database*. Jika data ditemukan maka *database* akan memberikan hasil pencarian yang selanjutnya akan ditampilkan oleh sistem. Apabila pengguna yang ingin melihat *detail* info kos tersebut dapat mengklik salah satu hasil pencarian tadi dan sistem akan menampilkan *detail* data dari perguruan tinggi tersebut. Di dalam detail tersebut juga terdapat alamat kos yang apabila pengguna mengklik alamatnya maka otomatis akan menampilkan *map* lokasi kos tersebut. Untuk lebih jelasnya lagi dapat dilihat pada Gambar 3.11.



#### D. Sequence Diagram untuk Proses Pencarian Bemo

Proses pencarian bemo dimulai ketika *pengguna* memilih menu pencarian pada halaman awal menu info bemo. Di halaman pencarian jurusan terdapat menu pencarian *advance*. Pencarian *advance* adalah pencarian dengan menggunakan beberapa filter yang diinginkan sehingga akan menampilkan jurusan-jurusan yang sesuai dengan kriteria pencari. Pada halaman pencarian *advance* ini akan terdapat beberapa filter yang selanjutnya diisi sesuai keinginan pengguna. Filter-filter

yang diisikan selanjutnya akan dicocokkan dengan data pada *database*. Jika data ditemukan maka *database* akan memberikan hasil pencarian yang selanjutnya akan ditampilkan oleh sistem. Apabila pengguna yang ingin melihat *detail* info bemo tersebut dapat mengklik salah satu hasil pencarian tadi dan sistem akan menampilkan *detail* data dari bemo tersebut. Untuk lebih jelasnya lagi dapat dilihat pada Gambar 3.12.



Gambar 3.12 *Sequence diagram* untuk proses “Cari Bemo”

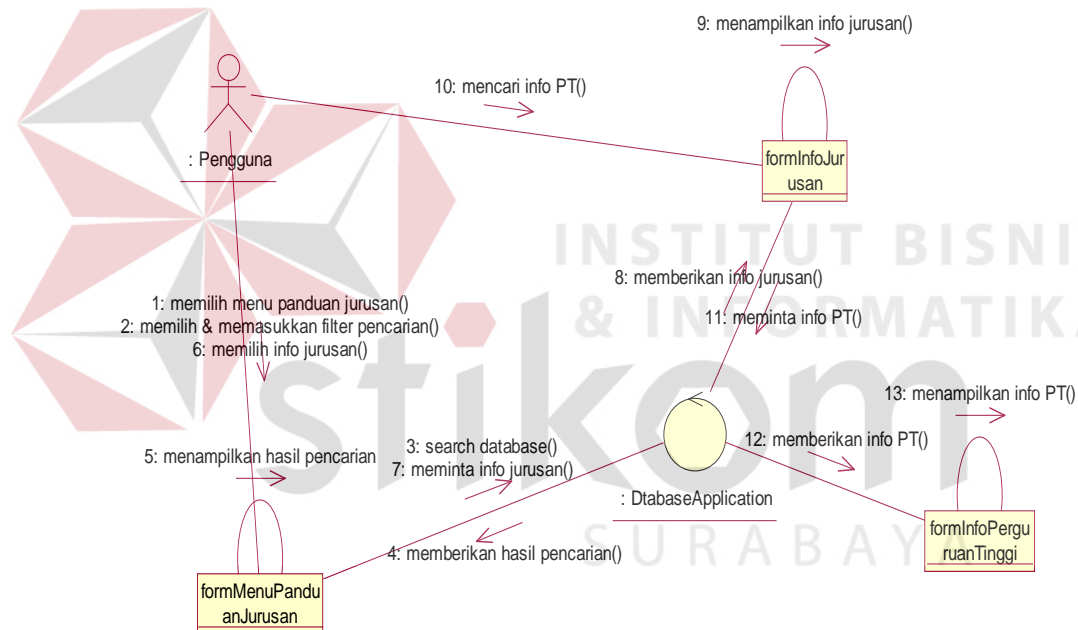
### 3.6 Collaboration Diagram

*Collaboration Diagram* menunjukkan informasi yang sama persis dengan diagram sekuensial (*Sequence Diagram*), tetapi dalam bentuk dan tujuan yang berbeda. Pada diagram sekuensial, keseluruhan interaksi berdasarkan urutan

waktu tetapi pada diagram kolaborasi interaksi antar obyek atau aktor ditunjukkan dengan arah panah tanpa keterangan waktu. (Sholiq, 2010).

#### A. Collaboration Diagram untuk Proses Pencarian Jurusan

Proses pada diagram ini diawali dengan pengguna membuka halaman *form* menu panduan jurusan. Kemudian pengguna memilih menu pencarian untuk memulai proses pencarian jurusan ini. Untuk proses ini pencariannya memiliki filter yang membantu pengguna pencari jurusan yang dapat melakukan pencarian lebih lanjut. Untuk lebih jelasnya dapat dilihat pada Gambar 3.13.

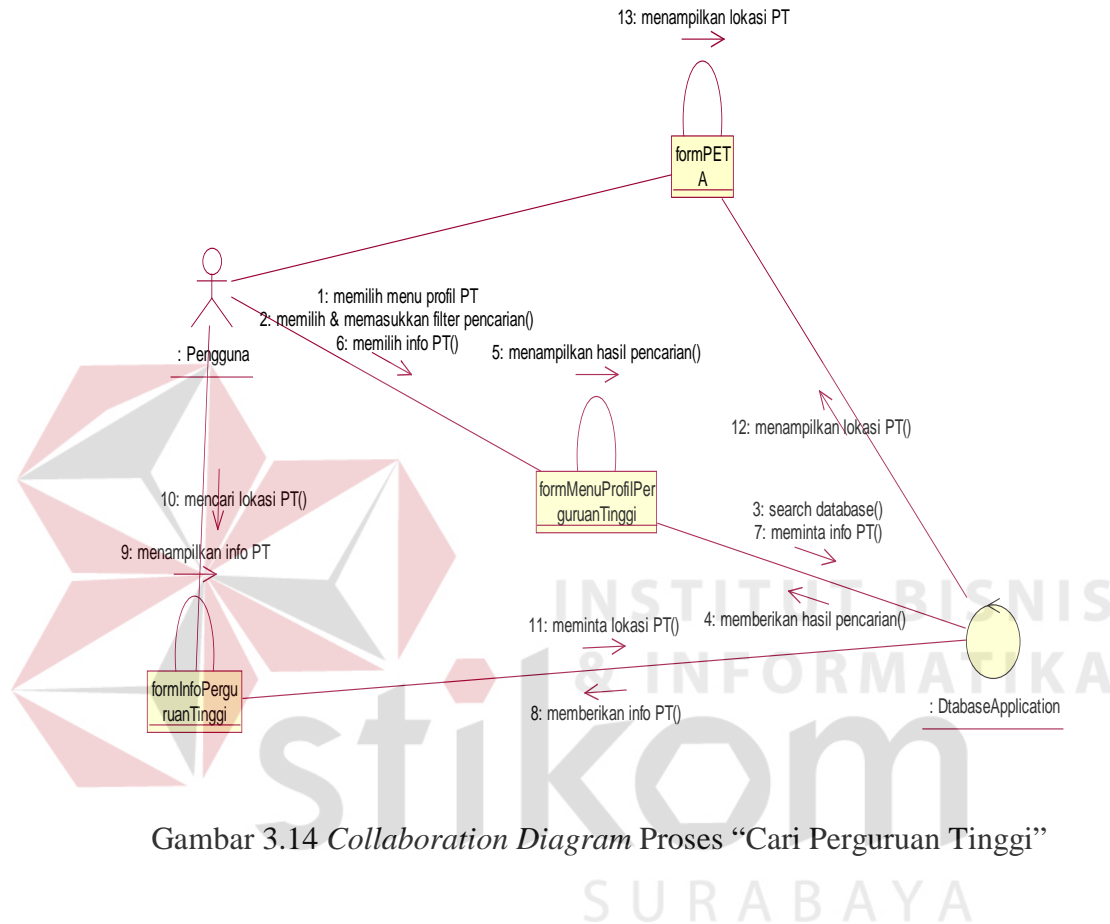


Gambar 3.13 Collaboration Diagram Proses “Cari Jurusan”

#### B. Collaboration Diagram untuk Proses Pencarian Perguruan Tinggi

Proses pada diagram ini diawali dengan pengguna membuka halaman *form* menu profil perguruan tinggi. Kemudian pengguna memilih menu pencarian untuk memulai proses pencarian perguruan tinggi ini. Untuk proses ini

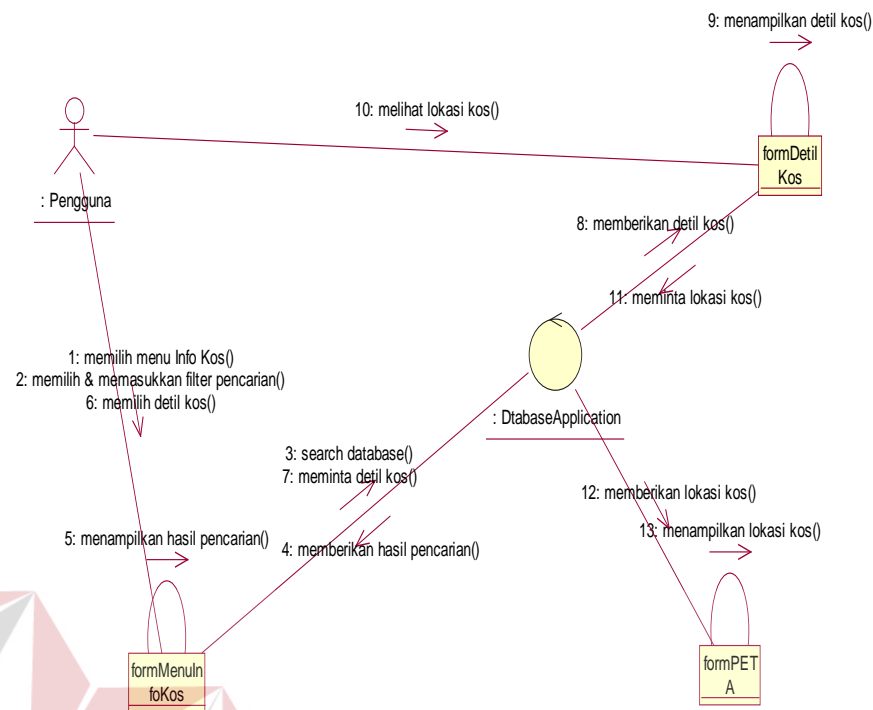
pencariannya memiliki filter yang membantu pengguna pencari perguruan tinggi yang dapat melakukan pencarian lebih lanjut. Untuk lebih jelasnya dapat dilihat pada Gambar 3.14.



Gambar 3.14 Collaboration Diagram Proses “Cari Perguruan Tinggi”

### C. Collaboration Diagram untuk Proses Pencarian Kos

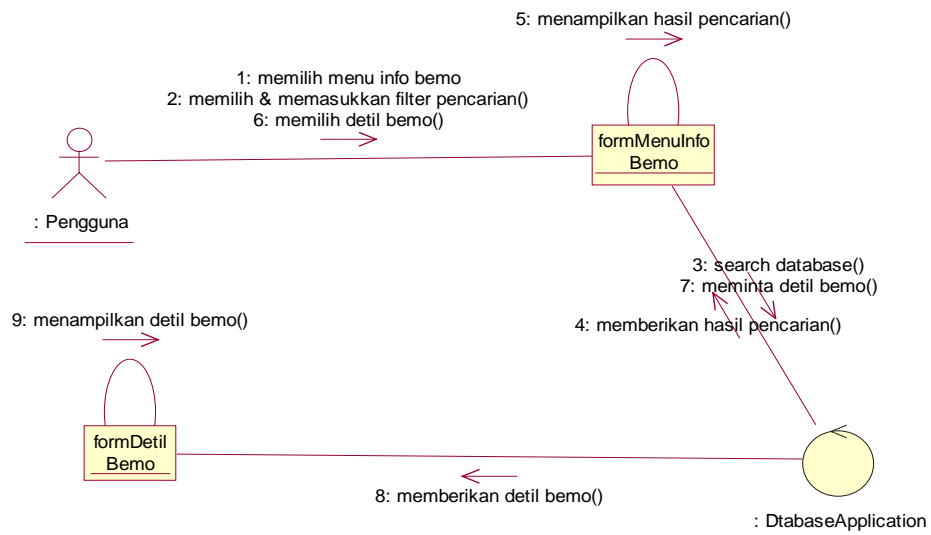
Proses pada diagram ini diawali dengan pengguna membuka halaman *form* menu info kos. Kemudian pengguna memilih menu pencarian untuk memulai proses pencarian kos ini. Untuk proses ini pencariannya memiliki filter yang membantu pengguna pencari kos yang dapat melakukan pencarian lebih lanjut. Untuk lebih jelasnya dapat dilihat pada Gambar 3.15.



Gambar 3.15 Collaboration Diagram Proses “Pencarian Kos”

#### D. Collaboration Diagram untuk Proses Pencarian Bemo

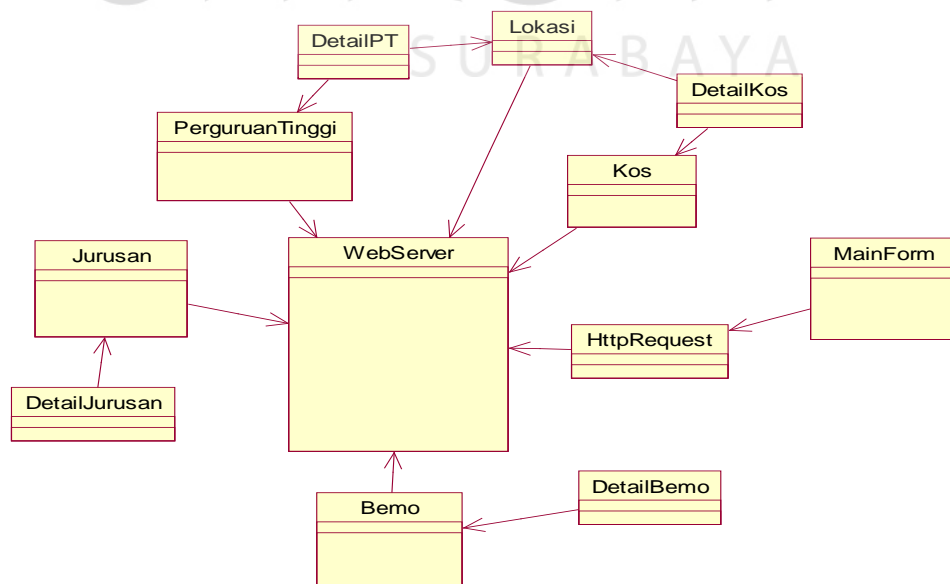
Proses pada diagram ini diawali dengan pengguna membuka halaman *form* menu info bemo. Kemudian pengguna memilih menu pencarian untuk memulai proses pencarian bemo ini. Untuk proses ini pencariannya memiliki filter yang membantu pengguna pencari bemo yang dapat melakukan pencarian lebih lanjut. Untuk lebih jelasnya dapat dilihat pada Gambar 3.16.



Gambar 3.16 Collaboration Diagram Proses “Cari Bemo”

### 3.7 Class Diagram

*Class Diagram* digunakan untuk menampilkan kelas-kelas atau paket-paket didalam sistem dan relasi antar kelas tersebut (menunjukkan interaksi antar kelas di dalam aplikasi). Seperti pada gambar 3.17.

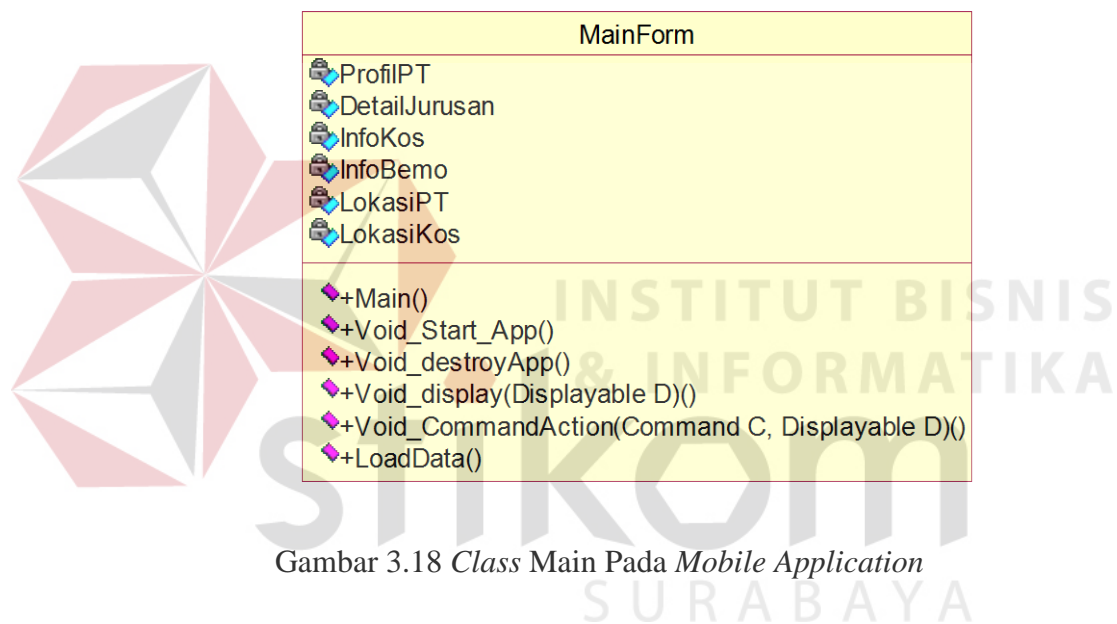


Gambar 3.17 Class Diagram Pada Mobile Application

### 3.8 Class Diagram Pada Mobile Application

#### A. Class Main

Kelas *Main* digunakan sebagai *form* utama pada *mobile application*. Kelas ini mengkoordinasikan beberapa operasi seperti inisialisasi data awal saat aplikasi dijalankan, penentuan tampilan awal aplikasi, dan lainnya. Dengan kata lain kelas ini digunakan sebagai penghubung dengan kelas-kelas yang lain. Untuk lebih jelasnya dapat dilihat pada gambar 3.18.

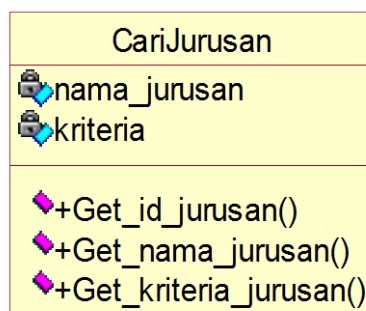


Gambar 3.18 Class Main Pada Mobile Application

#### B. Class Cari Jurusan

*Class* Cari Jurusan digunakan untuk menangani pencarian jurusan yang sesuai. Kelas ini memiliki prosedur *get\_id\_jurusan()* untuk mengambil data jurusan. Untuk lebih jelasnya dapat dilihat pada gambar 3.19.

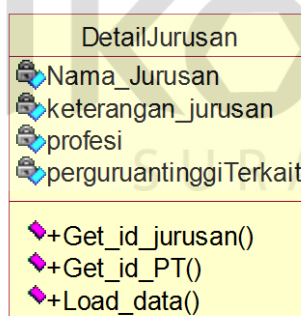




Gambar 3.19 *Class Cari Jurusan Pada Mobile Application*

### C. *Class Detail Jurusan*

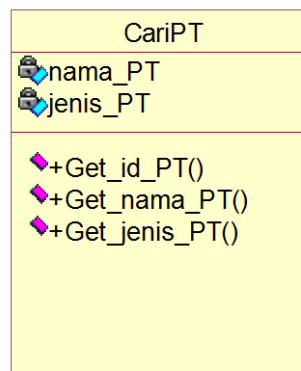
*Class* ini digunakan untuk menampilkan data jurusan yang melekat pada suatu menu. Di dalam *class* detail jurusan ini juga mempunyai atribut Nama\_Jurusan, keterangan\_jurusan, profesi dan perguruan tinggiTerkait. *Class* ini juga mempunyai operasi `get_id_jurusan`, `get_id_PT` dan `Load_data`. *Class diagram* DetailJurusan pada sistem ini dapat dilihat pada Gambar 3.20.



Gambar 3.20 *Class Detail Jurusan Pada Mobile Application*

### D. *Class Cari Perguruan Tinggi*

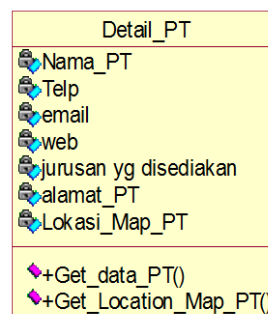
*Class* Cari Perguruan Tinggi digunakan untuk menangani pencarian PT yang sesuai. *Class* ini memiliki prosedur `get_id_PT()` untuk mengambil data perguruan tinggi. Untuk lebih jelasnya dapat dilihat pada gambar 3.21.



Gambar 3.21 *Class Cari Perguruan Tinggi Pada Mobile Application*

### E. *Class Detail Perguruan Tinggi*

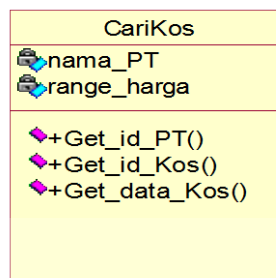
*Class* ini digunakan untuk menampilkan data perguruan tinggi yang melekat pada suatu menu. Di dalam *class* detail perguruan tinggi ini juga mempunyai atribut Nama\_PT, Telp, email, web, jurusan yg disediakan, alamat\_PT, Lokasi\_Map\_PT. *Class* ini juga mempunyai operasi *get\_data\_PT*, *get\_location\_map\_pt*. *Class diagram* DetailJurusan pada sistem ini dapat dilihat pada Gambar 3.22.



Gambar 3.22 *Class Detail Perguruan Tinggi Pada Mobile Application*

## F. Class Cari Kos

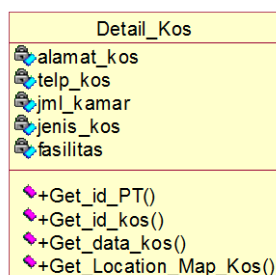
*Class* Cari Kos digunakan untuk menangani pencarian kos yang sesuai. *Class* ini memiliki prosedur *get\_id\_PT()* dan *get\_id\_Kos* untuk mengambil data kos yang berada di sekitar perguruan tinggi yang dipilih. Untuk lebih jelasnya dapat dilihat pada gambar 3.23.



Gambar 3.23 *Class Cari Kos Pada Mobile Application*

## G. Class Detail Kos

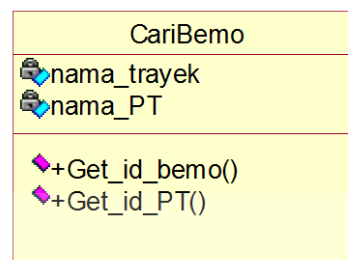
*Class* ini digunakan untuk menampilkan data kos yang melekat pada suatu menu. Di dalam *class* detail kos ini juga mempunyai atribut **alamat\_kos**, **telp\_kos**, **jml\_kamar**, **jenis\_kos** dan **fasilitas**. *Class* ini juga mempunyai operasi **get\_data\_PT** serta **get\_location\_map\_pt** yang berfungsi untuk menampilkan lokasi Kos di *Map*. *Class diagram* **Detail\_Kos** pada sistem ini dapat dilihat pada Gambar 3.24.



Gambar 3.24 *Class Detail Kos Pada Mobile Application*

## H. Class Cari Bemo

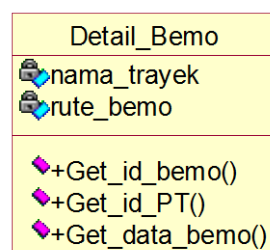
*Class* Cari Bemo digunakan untuk menangani pencarian bemo yang sesuai. *Class* ini memiliki prosedur *get\_id\_PT()* dan *get\_id\_bemo* untuk mengambil data kos yang rute jalannya melewati perguruan tinggi yang dipilih. Untuk lebih jelasnya dapat dilihat pada gambar 3.25.



Gambar 3.25 *Class* Cari Bemo Pada *Mobile Application*

## I. Class Detail Bemo

*Class* ini digunakan untuk menampilkan data jurusan yang melekat pada suatu menu. Di dalam *class* detail bemo ini juga mempunyai atribut *nama\_trayek* dan *rute\_bemo*. *Class* ini juga mempunyai operasi *get\_id\_bemo*, *get\_id\_PT* dan *get\_data\_bemo*. *Class diagram* Detail\_Bemo pada sistem ini dapat dilihat pada Gambar 3.26.



Gambar 3.26 *Class* Detail Bemo Pada *Mobile Application*

### J. Class Lokasi

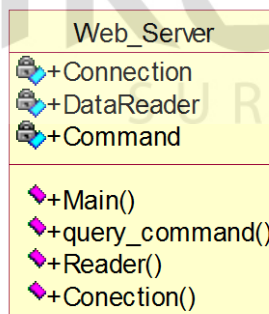
*Class* ini digunakan untuk mengambil koordinat lokasi PT dan Kos yang sudah di set di google maps. Untuk lebih jelasnya dapat dilihat pada gambar 3.27.



Gambar 3.27 *Class Lokasi Pada Mobile Application*

### K. Class Web Server

Kelas *Web Server* digunakan oleh *mobile application* untuk melakukan koneksi dengan *web server*. Kelas ini mendefinisikan semua atribut dan operasi yang berguna untuk melakukan koneksi. Untuk lebih jelasnya dapat dilihat pada gambar 3.28.

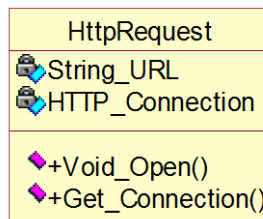


Gambar 3.28 *Class Webserver Pada Mobile Application*

### L. Class HTTPRequest

Kelas *HTTPRequest* digunakan oleh *mobile application* untuk melakukan koneksi dengan *web server*. Kelas ini mendefinisikan semua atribut dan operasi

yang berguna untuk melakukan koneksi. Untuk lebih jelasnya dapat dilihat pada gambar 3.29.



Gambar 3.29 Class HttpRequest Pada *Mobile Application*

### 3.9 Component Diagram

*Component Diagram* atau diagram komponen adalah diagram UML yang menampilkan komponen dalam sistem dan hubungan antara mereka. Hanya ada satu tipe relasi di dalam diagram ini yaitu relasi dependensi yang berarti suatu komponen memiliki ketergantungan dengan komponen yang lain atau satu komponen harus dikompilasi sebelum komponen lain yang bergantung padanya dikompilasi.

Pada Tugas Akhir ini, komponen-komponen di dalam *mobile application* dan *web application* dibagi menjadi dua komponen, yaitu **komponenGUI** yang menangani antar muka dengan pengguna dan **komponenControl** yang berisi semua kelas kontrol.

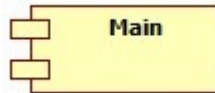


Gambar 3.30 Hubungan antar komponen

### 3.9.1 Package Spesification Pada Mobile Application

#### A. Package Spesification KomponenGUI

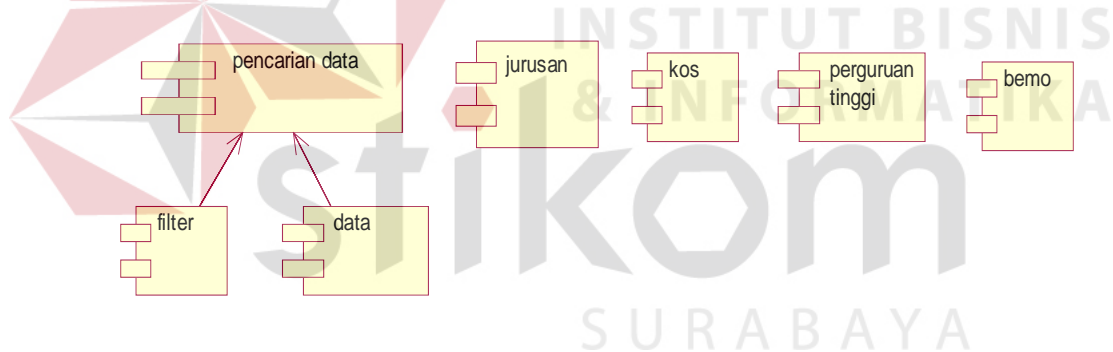
*Package Spesification* yang terdapat dalam *KomponenGUI mobile application* ini dapat dilihat pada gambar 3.31.



Gambar 3.31 *Package Spesification* *KomponenGUI Mobile Application*

#### B. Package Spesification KomponenKontrol

*Package Spesification* yang terdapat dalam *KomponenKontrol mobile application* ini dapat dilihat pada gambar 3.32.

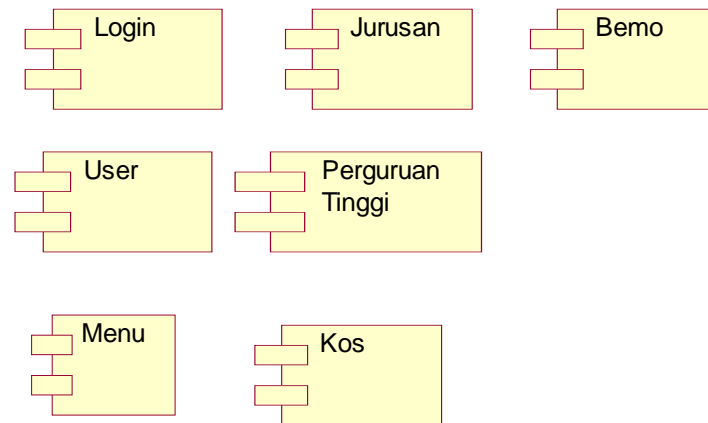


Gambar 3.32 *Package Spesification* *KomponenKontrol Mobile Application*

### 3.9.2 Package Spesification Pada Web Application

#### A. Package Spesification KomponenGUI

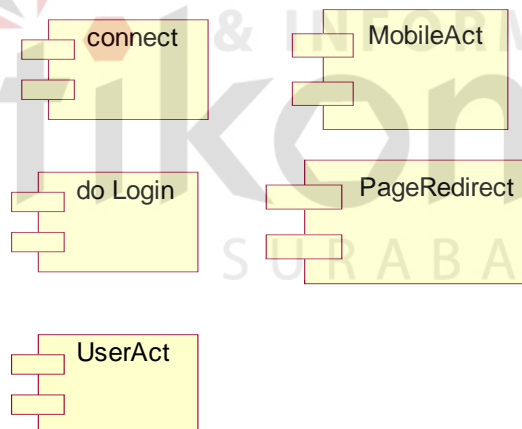
*Package Spesification* yang terdapat dalam *KomponenGUI web application* ini dapat dilihat pada gambar 3.33.



Gambar 3.33 *Package Spesification* KomponenGUI Web Application

### B. *Package Spesification* KomponenKontrol

*Package Spesification* yang terdapat dalam KomponenKontrol web application ini dapat dilihat pada gambar 3.34.



Gambar 3.34 *Package Spesification* KomponenKontrol Web Application

### 3.10 *Deployment Diagram*

*Deployment Diagram* menunjukkan pandangan secara fisik dari suatu sistem dan menunjukkan bagaimana sistem diimplementasikan di perangkat



nyata. *Deployment Diagram* menampilkan semua *node* dalam suatu jaringan dan hubungan di antara mereka. *Node* adalah perangkat keras yang dapat menjadi *host* dari suatu aplikasi. *Deployment Diagram* pada Tugas Akhir ini digambarkan pada gambar 3.35.



Gambar 3.35 *Deployment diagram*

### 3.11 Struktur Tabel

Tabel-tabel yang digunakan dalam aplikasi panduan perguruan tinggi berbasis Android adalah sebagai berikut:

#### 1. Nama Tabel : User

Fungsi : Menyimpan data *users*/pengguna yang dapat mengakses *web application*.

Tabel 3.2 Struktur Tabel *User*

Nama Kolom	Tipe	Ukuran	Keterangan
ID_user	Integer	11	PK
Username	Varchar	50	
Password	Varchar	50	

#### 2. Nama Tabel : Perguruan Tinggi

Fungsi : Menyimpan data-data Perguruan Tinggi

Tabel 3.3 Struktur Tabel Perguruan Tinggi

Nama Kolom	Tipe	Ukuran	Keterangan
ID_PT	Varchar	5	PK

Nm_PT	Varchar	50	
Alamat_PT	Varchar	50	
Logo	Image	-	
Telp_PT	Varchar	50	
Fax	Varchar	50	
Website	Varchar	50	
Email	Varchar	50	
Jumlah_Dosen	Varchar	10	
Pendidikan	Varchar	10	
Latitude	-	-	
Longitude	-	-	
Foto_PT	Image	-	
Keterangan	Varchar	50	
Jenis_PT	Varchar	50	
Thn_Berdiri	Varchar	10	

### 3. Nama Tabel : Jurusan

Fungsi : Menyimpan data-data Jurusan.

Tabel 3.4 Struktur Tabel Jurusan

Nama Kolom	Tipe	Ukuran	Keterangan
ID_Jurusan	Varchar	10	PK
ID_PT	Varchar	10	FK
Nama_Jurusan	Varchar	50	
Kriteria_Jurusan	Varchar	50	
Jenjang	Varchar	50	
Akreditasi	Varchar	50	
Kadaluarsa	date	-	
Keterangan_Jurusan	Varchar	50	
Profesi	Varchar	50	

4. Nama Tabel : KOS

Fungsi : Menyimpan data-data KOS.

Tabel 3.5 Struktur Tabel Kos

Nama Kolom	Tipe	Ukuran	Keterangan
ID_Kos	Varchar	10	PK
ID_PT	Varchar	10	FK
Alamat_Kos	Varchar	50	
Jenis_Kos	Varchar	10	
Telp_Kos			
Lat_Kos	-	-	
Long_Kos	-	-	
Foto_Kos	Image	-	
Fasilitas	Varchar	50	
Harga	Varchar	10	
Kamar	Varchar	5	

5. Nama Tabel : Bemo

Fungsi : Menyimpan data jalur atau rute angkutan.

Tabel 3.6 Struktur Tabel Bemo

Nama Kolom	Tipe	Ukuran	Keterangan
ID_Bemo	Varchar	10	PK
ID_PT	Varchar	10	FK
Kode_Trayek	Varchar	10	
Jalur_Bemo	Varchar	50	
Foto_Bemo	Image		

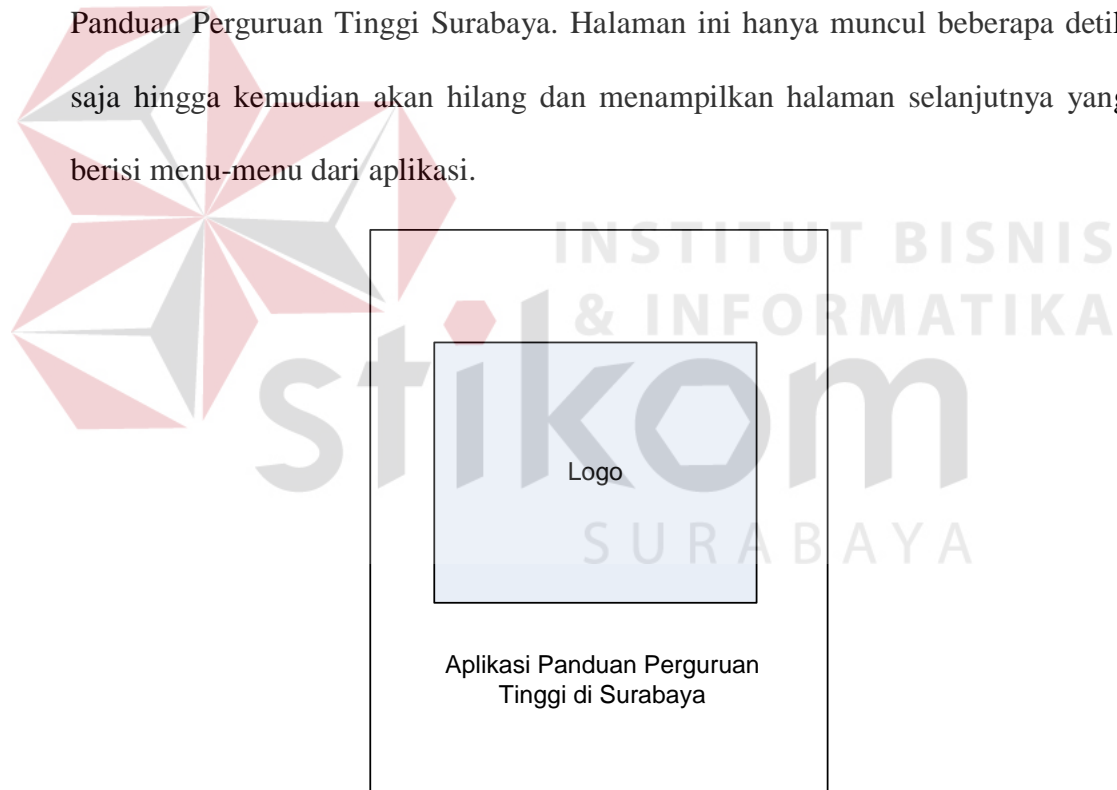
### 3.12 Desain *Interface* Aplikasi

Pembuatan desain *input/output* diperlukan untuk membantu pengguna berinteraksi dengan sistem. Desain *input/output* yang dibuat meliputi desain untuk *mobile application*.

#### 3.12.1 Desain *Mobile Application*

##### A. Halaman Pembuka

Halaman pembuka merupakan halaman yang akan ditampilkan pertama kali ketika aplikasi dijalankan. Halaman pembuka akan menampilkan simbol Panduan Perguruan Tinggi Surabaya. Halaman ini hanya muncul beberapa detik saja hingga kemudian akan hilang dan menampilkan halaman selanjutnya yang berisi menu-menu dari aplikasi.

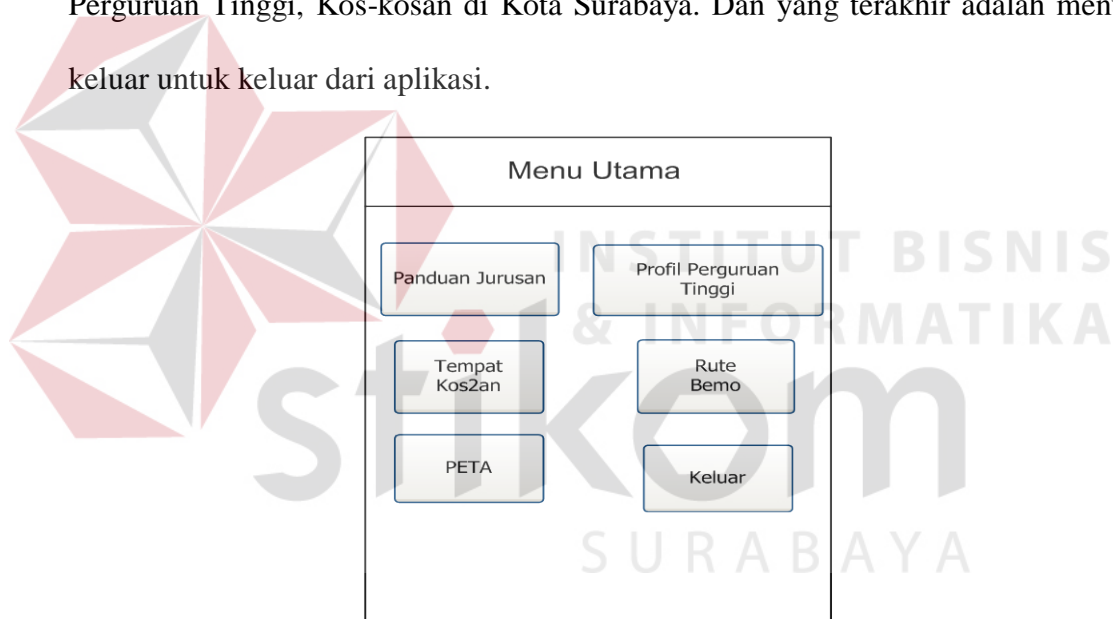


Gambar 3.37 Desain Halaman Pembuka *Mobile Application*

##### B. Halaman Utama

Halaman utama terdiri dari 7 menu, yaitu Panduan Jurusan menu ini untuk memberikan rekomendasi program studi yang sesuai, maksudnya adalah

menu ini ditujukan kepada pelajar yang masih bingung memilih program studi yang sesuai dengan mereka. Lalu Profil Perguruan Tinggi menu untuk mencari informasi Perguruan Tinggi yang diinginkan, output yang dikeluarkan adalah informasi lengkap tentang Perguruan Tinggi seperti profil, alamat, daftar program studi beserta akreditasinya ataupun prestasi yang telah dicapai. Kemudian, Tempat Kos2an menu ini berisi tempat kos-kosan sesuai pilihan tempat perguruan tinggi masing-masing. Lalu Jalur bemo menu ini berisi jalur bemo yg melewati tiap-tiap perguruan tinggi. PETA, menu ini berisi map lokasi-lokasi semua Perguruan Tinggi, Kos-kosan di Kota Surabaya. Dan yang terakhir adalah menu keluar untuk keluar dari aplikasi.

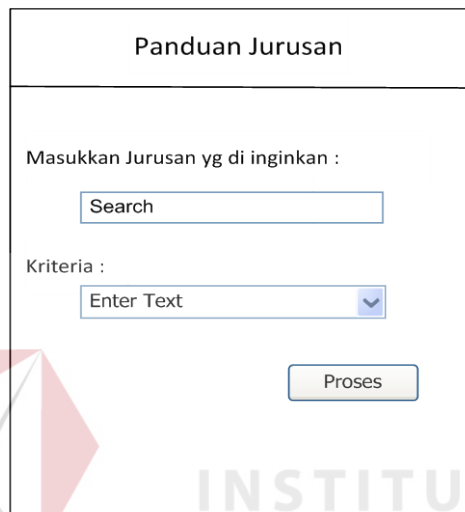


Gambar 3.38 Desain Halaman Utama *Mobile Application*

### C. Form Menu Panduan Jurusan

Form menu panduan jurusan digunakan untuk mencari informasi tentang jurusan tertentu sesuai yang diinginkan *user*. Pencarian bisa dilakukan dengan 2 cara, yaitu berdasarkan kode bis kota atau suatu lokasi tertentu yang menjadi trayek bis kota. Akan tetapi hanya bisa dipilih salah satu saja, berdasarkan Jurusan

atau Kriteria Jurusan. Pilihan yang di tampilkan kriteria yaitu antara lain : IPA, IPS, Bahasa, IT, Seni dan ALL(Semua kriteria di tampilkan). Setelah itu tekan tombol proses, dan akan tampil hasil pencarian berdasarkan pilihan *user* (jurusan atau kriteria).



Panduan Jurusan

Masukkan Jurusan yg di inginkan :

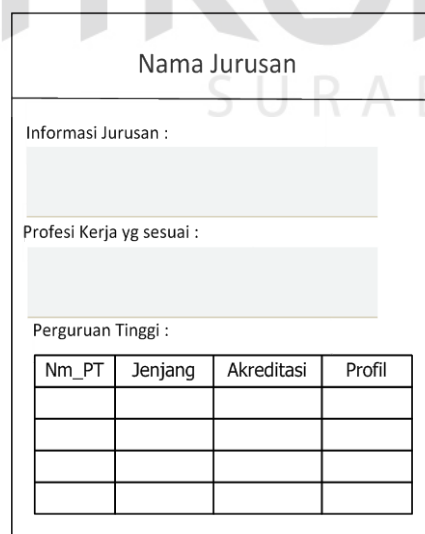
Search

Kriteria :

Enter Text

Proses

Gambar 3.39 Desain Halaman Menu Panduan Jurusan (*mobile*)



Nama Jurusan

Informasi Jurusan :

Profesi Kerja yg sesuai :

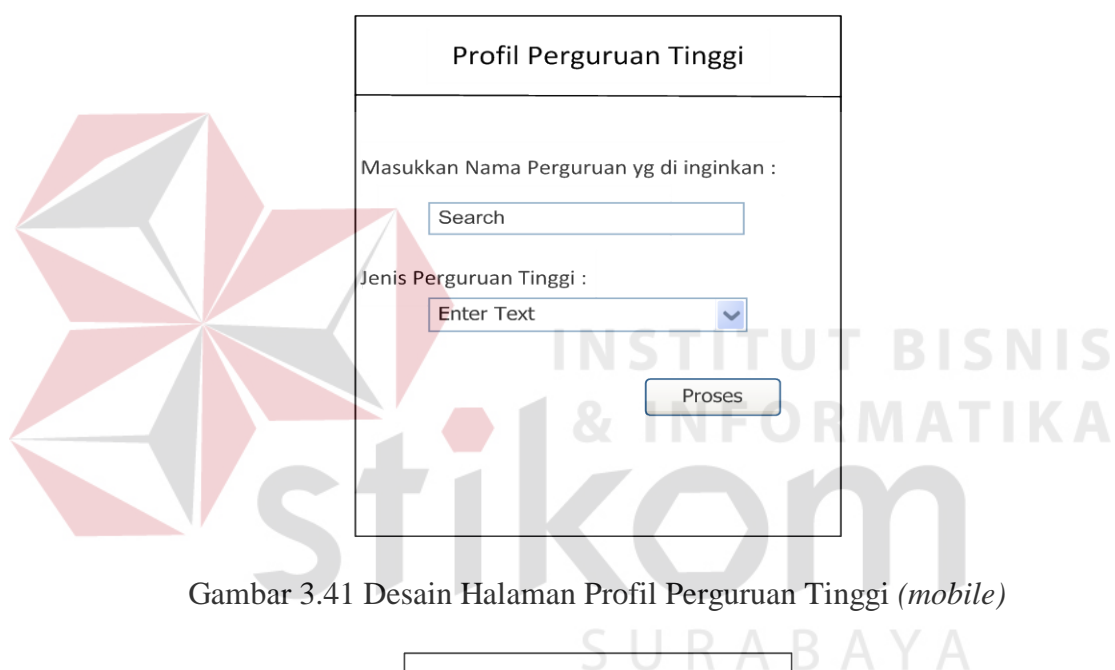
Perguruan Tinggi :

Nm_PT	Jenjang	Akreditasi	Profil

Gambar 3.40 Desain Halaman Menu Panduan Jurusan (*mobile*)

#### D. Form Profil Perguruan Tinggi

Form profil perguruan tinggi digunakan untuk mencari informasi tentang perguruan tinggi yang sesuai yang keinginan *user*. Pencarian bisa dilakukan dengan 2 cara, yaitu berdasarkan nama perguruan tinggi yang diinginkan atau berdasarkan jenis perguruan tinggi (Negeri dan Swasta). Dan hanya bisa dipilih salah satu saja, berdasarkan nama atau jenis. Setelah itu tekan tombol proses, dan akan tampil hasil pencarian berdasarkan pilihan *user* (nama atau jenis).



**Profil Perguruan Tinggi**

---

Masukkan Nama Perguruan yg di inginkan :

Search

Jenis Perguruan Tinggi :

Enter Text ▼

Proses

Gambar 3.41 Desain Halaman Profil Perguruan Tinggi (*mobile*)

**Profil Perguruan Tinggi**

---

Nama Perguruan Tinggi

Tgl Berdiri

Telp,Fax,dan Email

Website

Alamat

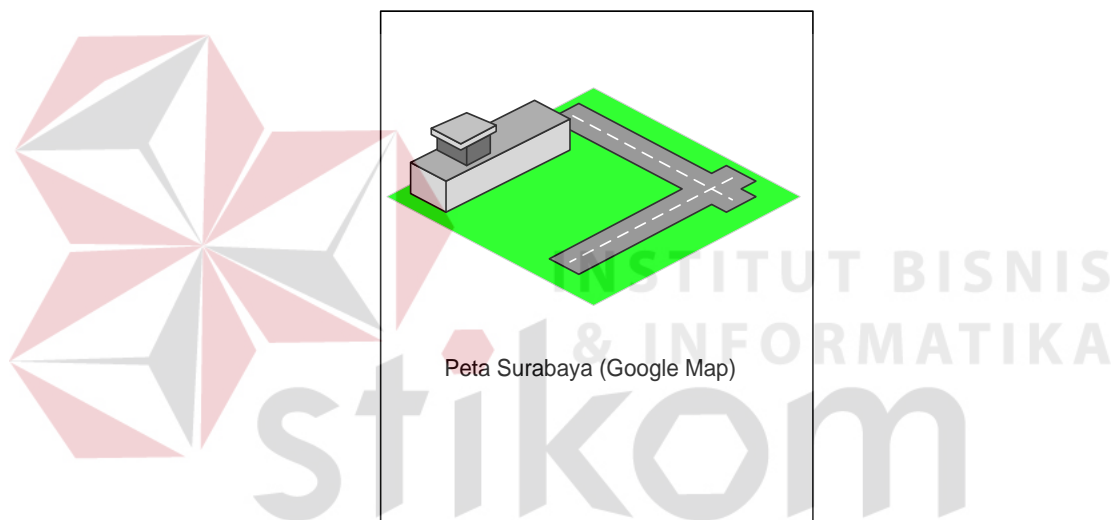
Jenjang	Jurusan	Akreditasi

Jml Dosen dan Pendidikannya

Gambar 3.42 Desain Halaman Detail Profil Perguruan Tinggi (*mobile*)

### E. Form Menu PETA

Form menu Peta menampilkan letak *user* dan tempat-tempat tertentu di Kota Surabaya dalam bentuk peta digital. Pada form cari lokasi ini *user* dapat melihat dimana lokasi dia berada dan beberapa tempat yang sudah ditandai (*mark*) pada sistem di kota Surabaya. *User* juga bisa mengetahui jarak antara dia berada dengan tempat yang sudah ditandai dalam peta tersebut dengan cara mengklik tempat tersebut. *User* juga bisa melakukan pembesaran (*zoom in*) atau pengecilan (*zoom out*) tampilan peta.

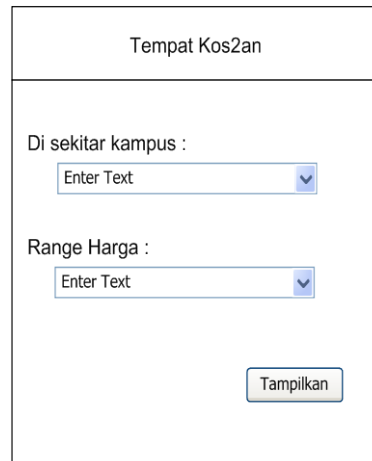


Gambar 3.43 Desain Halaman Peta (*mobile*)

### F. Form Menu Tempat Kos

Form Menu Tempat Kos digunakan untuk mencari kos sesuai yang diinginkan *user*. Pencarian bisa dilakukan dengan 2 cara, yaitu berdasarkan kampus yang di pilih dan range harga kos yang sesuai. Setelah itu tekan tombol proses, dan akan tampil hasil pencarian berdasarkan pilihan *user* (kampus dan range).





Tempat Kos2an

Di sekitar kampus :

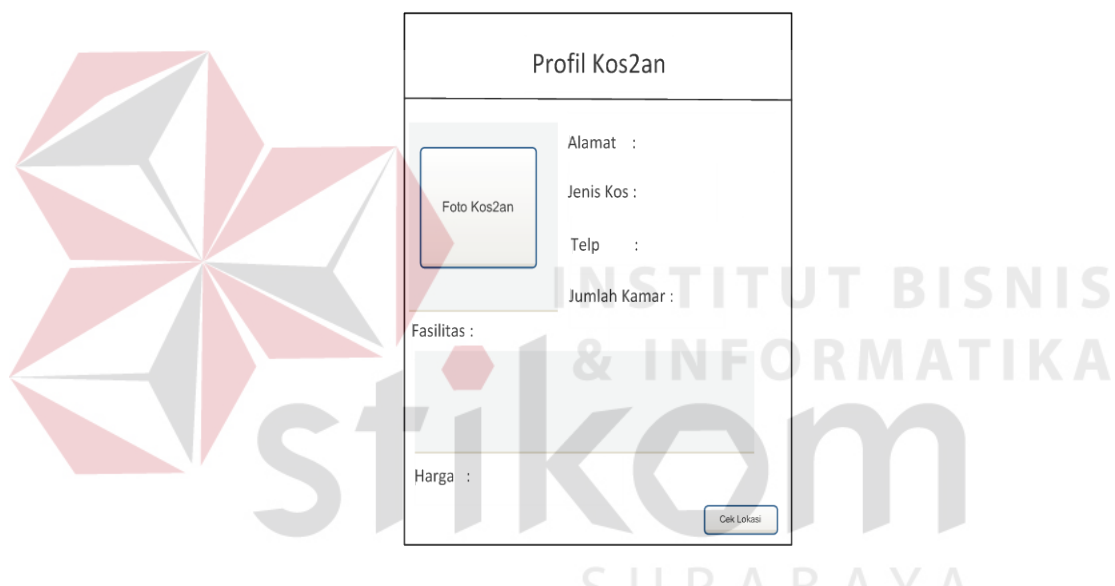
Enter Text

Range Harga :

Enter Text

Tampilkan

Gambar 3.44 Desain Halaman Pencarian Tempat Kos-kosan (*mobile*)



Profil Kos2an

Foto Kos2an

Alamat :

Jenis Kos :

Telp :

Jumlah Kamar :

Fasilitas :


Harga :

Cek Lokasi

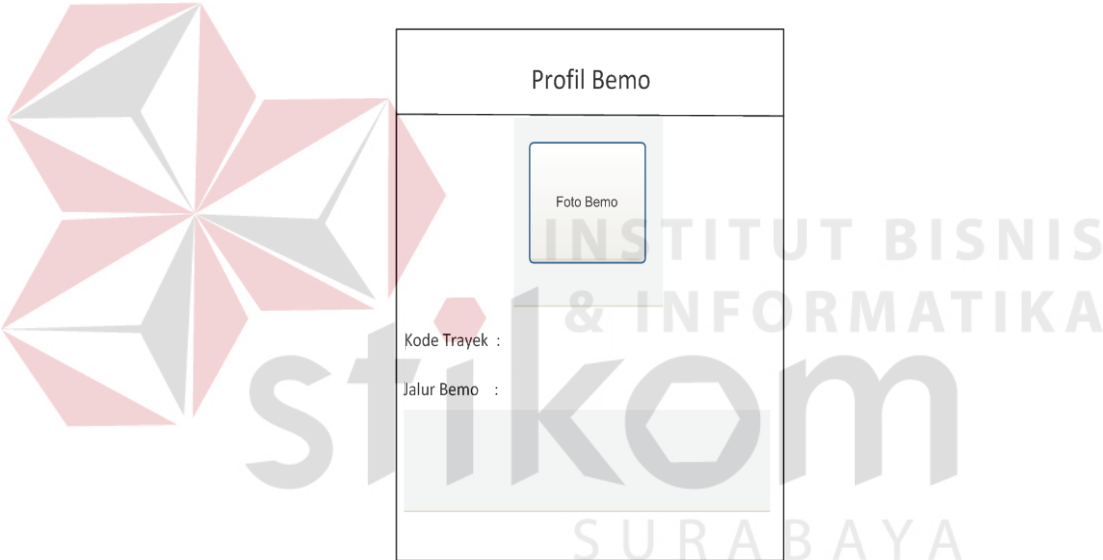
Gambar 3.45 Desain Halaman Informasi Tempat Kos2an (*mobile*)

### G. Form Rute Bemo

Form cari bemo digunakan untuk mencari bemo yang melewati rute tertentu sesuai yang diinginkan *user*. Pencarian bisa dilakukan dengan 2 cara, yaitu berdasarkan kode bemo atau berdasarkan kampus yang di pilih. Dan hanya bisa dipilih salah satu saja, berdasarkan kode atau kampus. Setelah itu tekan tombol proses, dan akan tampil hasil pencarian berdasarkan pilihan *user* (kode atau kampus).



Gambar 3.46 Desain Halaman pencarian Rute Bemo (*mobile*)



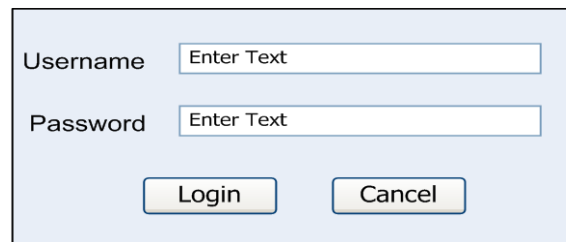
Gambar 3.47 Desain Halaman Informasi Rute bemo (*mobile*)

### 3.12.2 Desain *Web Application*

#### A. Form Login

Form login merupakan halaman utama yang akan ditampilkan ketika *users/pengguna* membuka halaman *web* panduan perguruan tinggi di Surabaya.

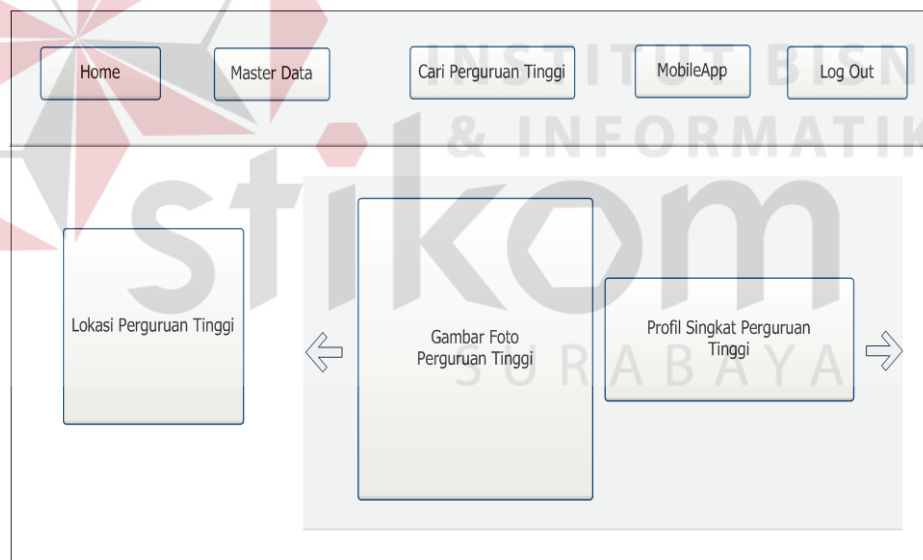
Form login digunakan untuk memverifikasi pengguna yang melakukan login, karena hanya admin yang memiliki akses penuh.



Gambar 3.48 Desain Halaman Form Login (*web*)

## B. Halaman Utama

Pada halaman utama terdapat waktu sistem yang ditampilkan pada bagian kiri atas dari halaman utama dan pilihan menu yang berupa ikon.



Gambar 3.49 Desain Halaman Utama (*web*)

## C. Form Data Perguruan Tinggi

Form Perguruan Tinggi digunakan untuk memelihara master data perguruan tinggi, seperti menyimpan, mengubah dan menghapus data perguruan

tinggi. Data yang ditambahkan antara lain, nama perguruan tinggi, nomor telpon, website, fax, email, jml dosen, jenis perguruan tinggi, tgl berdiri, foto dan logo. Pada form ini terdapat tombol tambah jalan baru, tombol ubah dan tombol hapus.

Id	Nama	Telp	Fax	Website	Email	Jml Dosen	Jenis PT	Tgl Berdiri	Foto & Logo	Action

Gambar 3.50 Desain Halaman Data Perguruan Tinggi (*web*)

**Data Perguruan Tinggi**

Nama PT : Enter Text

Telp : Enter Text

Fax : Enter Text

Website : Enter Text

Email : Enter Text

Jml Dosen : Enter Text

Jenis PT : Enter Text ▼

Tanggal Berdiri : Enter Text

Foto & Logo :

Save Reset

Gambar 3.51 Desain Halaman Pengisian Data Perguruan Tinggi (*web*)

#### D. Form Alamat Perguruan Tinggi

Form Alamat Perguruan Tinggi digunakan untuk memelihara master data alamat perguruan tinggi, seperti menyimpan, mengubah dan menghapus data alamat perguruan tinggi. Data yang ditambahkan antara lain, nama perguruan

tinggi, alamat, keterangan, Latitude, Longitude. Pada form ini terdapat tombol tambah jalan baru, tombol ubah dan tombol hapus.

Id Alamat	Nama Perguruan Tinggi	Alamat	Keterangan	Lat	Long	Action

Gambar 3.52 Desain Form Alamat Perguruan Tinggi (*web*)

Gambar 3.53 Desain Form Pengisian Data Alamat Perguruan Tinggi (*web*)

### E. Form Prodi Perguruan Tinggi

Form Prodi Perguruan Tinggi digunakan untuk memelihara master prodi, seperti menyimpan, mengubah dan menghapus data prodi. Data yang ditambahkan antara lain, nama perguruan tinggi, jenjang, nama prodi, akreditasi kadaluarsa. Pada form ini terdapat tombol tambah prodi baru, tombol ubah dan tombol hapus.

Id Prodi	Nama Perguruan Tinggi	Jenjang	Nama Prodi	Akreditasi	Kadaluarsa	Action

Gambar 3.54 Desain Form Prodi Perguruan Tinggi (*web*)

Gambar 3.55 Desain Form Pengisian Data Prodi Perguruan Tinggi (*web*)

## F. Form Detail Prodi Perguruan Tinggi

Form Detail Prodi Perguruan Tinggi digunakan untuk memelihara master detail prodi, seperti menyimpan, mengubah dan menghapus data detail prodi. Data yang ditambahkan antara lain, nama prodi, keterangan, profesi, kriteria jurusan. Pada form ini terdapat tombol tambah detail prodi baru, tombol ubah dan tombol hapus.

Id Detail Prodi	Nama Prodi	Keterangan	Profesi	Kriteria Jurusan	Action

Gambar 3.56 Desain Form Detail Prodi Perguruan Tinggi (*web*)

**Data Detail Prodi**

Id Detail Prodi

Nama Prodi  ▼

Keterangan

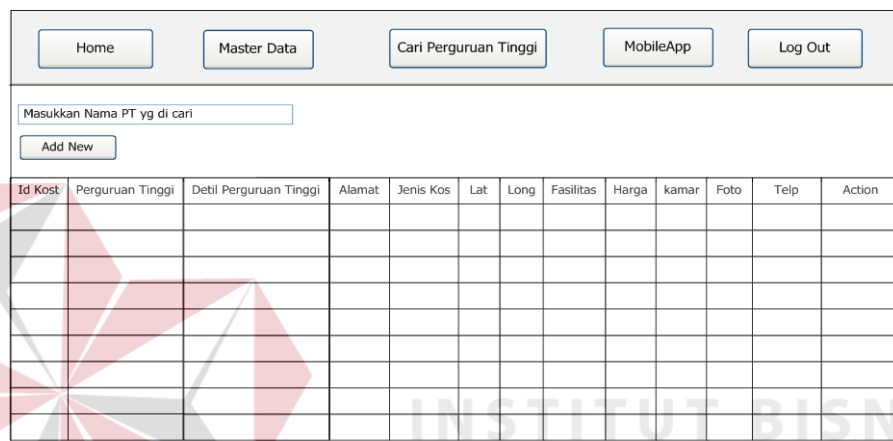
Profesi

Kriteria Jurusan  ▼

Gambar 3.57 Desain Form Pengisian Data Detail Prodi Perguruan Tinggi (*web*)

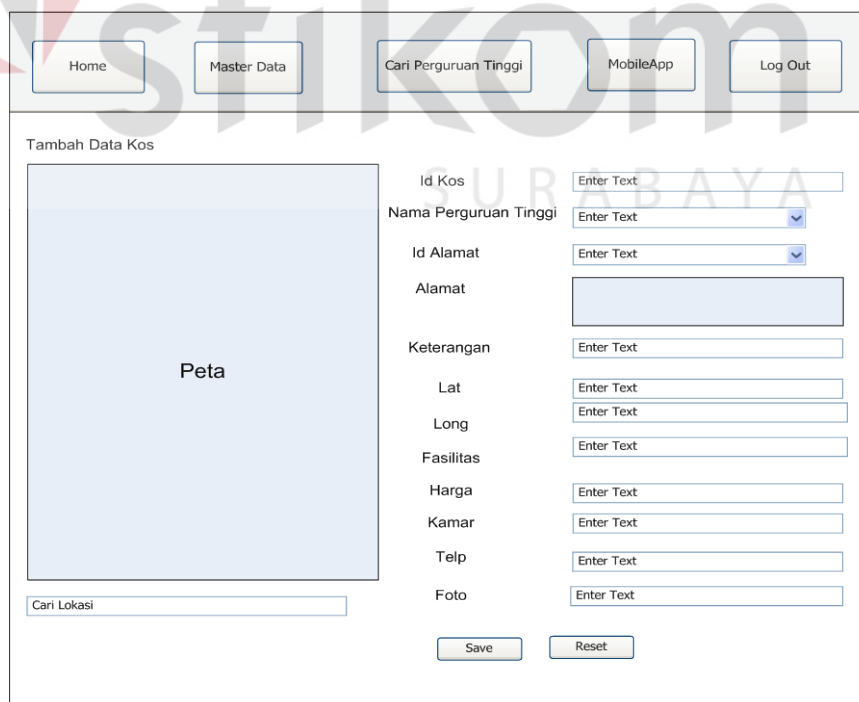
### G. Form Data Kos

Form Data Kos digunakan untuk memelihara data kos, seperti menyimpan, mengubah dan menghapus data kos. Data yang ditambahkan antara lain, nama perguruan tinggi, detail perguruan tinggi, alamat, jenis kos, Latitude, Longitude, fasilitas, harga, kamar, foto, telp. Pada form ini terdapat tombol tambah kos baru, tombol ubah dan tombol hapus.



Id Kos	Perguruan Tinggi	Detail Perguruan Tinggi	Alamat	Jenis Kos	Lat	Long	Fasilitas	Harga	kamar	Foto	Telp	Action

Gambar 3.58 Desain Form Data Kos (web)



Home
Master Data
Cari Perguruan Tinggi
MobileApp
Log Out

Masukkan Nama PT yg di cari  
Add New

Tambah Data Kos

Peta

Cari Lokasi

Id Kos
Enter Text

Nama Perguruan Tinggi
Enter Text

Id Alamat
Enter Text

Alamat

Keterangan
Enter Text

Lat
Enter Text

Long
Enter Text

Fasilitas
Enter Text

Harga
Enter Text

Kamar
Enter Text

Telp
Enter Text

Foto
Enter Text

Save
Reset

Gambar 3.59 Desain Form Pengisian Data Kos (web)



## H. Form Bemo

Form bemo digunakan untuk memelihara master bemo, seperti menyimpan, mengubah dan menghapus data bemo. Data yang ditambahkan antara lain, nama perguruan tinggi, detail perguruan tinggi, jalur bemo dan foto. Pada form ini terdapat tombol tambah bemo baru, tombol ubah dan tombol hapus.

Id Bemo	Nama Perguruan Tinggi	Detil Perguruan Tinggi	Kode Trayek	Jalur Bemo	Foto	Action

Gambar 3.60 Desain Form Jalur Bemo (*web*)

**Data Bemo**

Id Bemo

Id Perguruan Tinggi  ▼

Id Alamat

Kode Trayek

Jalur Bemo  ▼

Foto

Gambar 3.61 Desain Form Pengisian Data Bemo (*web*)

### 3.13 Desain Uji Coba

Desain uji coba bertujuan untuk memastikan bahwa aplikasi telah dibuat dengan benar sesuai dengan kebutuhan atau tujuan yang diharapkan. Kekurangan atau kelemahan sistem pada tahap ini akan dievaluasi sebelum diimplementasikan. Proses pengujian menggunakan *Black Box Testing* yaitu aplikasi akan diuji dengan melakukan berbagai percobaan untuk membuktikan bahwa aplikasi yang telah dibuat sudah sesuai dengan tujuan. Uji coba yang akan dilakukan antara lain:

1. Uji coba fungsi aplikasi.
2. Uji coba kompatibilitas sistem.

#### 3.13.1 Uji Coba Fungsi Aplikasi

##### A. Desain Uji Coba Aplikasi Web

Proses uji coba ini dilakukan untuk mengetahui apakah fungsi-fungsi dari Aplikasi Panduan Informasi dan Lokasi Perguruan Tinggi di Surabaya ini telah berjalan dengan benar. Setiap fitur yang disediakan akan diuji hasilnya sesuai dengan tabel *test case*. Desain uji coba fungsi aplikasi adalah sebagai berikut:

##### 1. Desain Uji Coba Login

Fungsi *login* digunakan untuk melakukan pembatasan hak akses kepada pengguna. Pada fungsi ini, terdapat pemeriksaan kelengkapan masukan nama pengguna dan kata kunci. Sistem memberikan pesan agar pengguna mengisi data ketika ada yang masih kosong. Selain itu, sistem juga akan memberi peringatan apabila nama pengguna dan kata kuncinya salah. Desain untuk uji coba fungsi *login* dapat dilihat pada Tabel 3.7.

Tabel 3.1 Desain Uji Coba Fungsi *Login*

Test Case ID	Tujuan	Input	Output yang diharapkan
	Menghindari data username dan atau password kosong	Tidak memasukkan nama pengguna dan kata kunci	Tampil pesan “nama pengguna dan kata kunci masih kosong”
	Menghindari username dan password yang salah	Memasukkan “Admin” sebagai nama pengguna dan “admin” sebagai kata kunci yang salah	Tampil pesan “nama pengguna dan kata kunci salah”
	Memastikan pemeriksaan username dan password yang dimasukkan benar dan hak akses dengan level pengguna sudah sesuai	Memasukkan “Admin” sebagai nama pengguna dan “admin” sebagai kata kunci.	Tampil dengan hak akses navigasi sesuai dengan level pengguna

## 2. Desain Uji Coba Tambah Data Perguruan Tinggi

Fungsi Tambah Data Perguruan Tinggi digunakan oleh *admin* untuk melakukan penambahan data perguruan tinggi yang dimiliki dan menyimpannya dalam *database*. *Admin* memilih menu tambah dan mengisikan kolom-kolom yang tersedia dengan data perguruan tinggi yang ingin ditambahkan. Desain untuk uji coba fungsi tambah perguruan tinggi dapat dilihat pada Tabel 3.8.

Tabel 3.8 Desain Uji Coba Fungsi Tambah Data Perguruan Tinggi

Test Case ID	Tujuan	Input	Output yang diharapkan
	Menghindari kolom nama perguruan tinggi kosong karena data tersebut banyak menjadi foreignkey di table lainya.	Mengisikan nama perguruan tinggi dengan data kosong.	Tampil pesan “Nama PT harus diisi”

### 3. Desain Uji Coba Tambah Data Detil Prodi

Fungsi Tambah Data Detil Prodi digunakan oleh *admin* untuk melakukan penambahan data detil prodi yang dimiliki dan menyimpannya dalam *database*. *Admin* memilih menu tambah dan mengisi kolom-kolom yang tersedia dengan data detil prodi yang ingin ditambahkan. Desain untuk uji coba fungsi tambah data detil prodi dapat dilihat pada Tabel 3.9.

Tabel 3.9 Desain Uji Coba Fungsi Tambah Data Detil Prodi

Test Case ID	Tujuan	Input	Output yang diharapkan
	Menghindari mengisi detail prodi dengan nama prodi yang sudah di isi. Di karenakan nantinya akan muncul dua nama prodi yang sama di menu panduan jurusan di	Memilih nama prodi yang sudah di isi detail prodinya.	Tampil pesan “Maaf, nama prodi tersebut sudah dipilih”

## B. Desain Uji Coba Aplikasi Mobile

### 1. Desain Uji Coba Pencarian Jurusan

Fungsi Pencarian Jurusan digunakan oleh pengguna pencari jurusan untuk melakukan pencarian jurusan yang diinginkan. Pencari jurusan memilih menu yang melakukan pencarian jurusan yang sesuai dengan kriteria. Desain untuk uji coba fungsi pencarian jurusan dapat dilihat pada Tabel 3.10.

Tabel 3.10 Desain Uji Coba Pencarian Jurusan

	Memastikan hasil pencarian tampil pada	Memilih dan Memasukkan nama jurusan “Sistem	Hasil Pencarian tampil pada sesuai dengan
--	--	---	---

	dan sesuai dengan inputan user.	Informasi”	inputan user dan kriteria pencarian.
	Memastikan hasil pencarian tampil pada dan sesuai dengan kriteria pencarian	Memilih kriteria jurusan yang dicari “IT”	Hasil Pencarian tampil dan sesuai dengan inputan user.

## 2. Desain Uji Coba Detail Jurusan

Fungsi Detail Jurusan digunakan oleh pengguna untuk mengetahui informasi lebih lanjut mengenai info jurusan yang dipilih. Pengguna memilih salah satu dari hasil pencarian yang dipilih dan sistem akan menampilkan halaman Detail Jurusan. Desain untuk uji coba fungsi detail indeks dapat dilihat pada Tabel 3.11

Tabel 3.11 Desain Uji Coba Detail Jurusan

Test Case ID	Tujuan	Input	Output yang diharapkan
	Memastikan detail jurusan sesuai dengan data	Memilih detail nama jurusan “Sistem Informasi”	Hasil Detail Jurusannya sesuai dengan data

## 3. Desain Uji Coba Pencarian Perguruan Tinggi

Fungsi Pencarian Perguruan Tinggi digunakan oleh pengguna untuk melakukan pencarian perguruan tinggi yang diinginkan. Pengguna memilih menu yang melakukan pencarian perguruan tinggi yang sesuai dengan kriteria. Desain untuk uji coba fungsi pencarian perguruan tinggi dapat dilihat pada Tabel 3.12.

Tabel 3.12 Desain Uji Coba Pencarian Perguruan Tinggi

Test Case ID	Tujuan	Input	Output yang diharapkan
	Memastikan hasil pencarian tampil pada dan sesuai dengan inputan user.	Memilih dan Memasukkan nama perguruan tinggi “Stmik Surabaya”	Hasil Pencarian tampil pada sesuai dengan inputan user dan kriteria pencarian.
	Memastikan hasil pencarian tampil pada dan sesuai dengan kriteria pencarian	Memilih jenis perguruan tinggi yang dicari “Swasta”	Hasil Pencarian tampil dan sesuai dengan inputan user.

#### 4. Desain Uji Coba Detail Perguruan Tinggi

Fungsi Detail Perguruan Tinggi digunakan oleh pengguna untuk mengetahui informasi lebih lanjut mengenai profil perguruan tinggi yang dipilih. Pengguna memilih salah satu dari hasil pencarian yang dipilih dan sistem akan menampilkan halaman Detail Perguruan Tinggi. Desain untuk uji coba fungsi detail perguruan tinggi dapat dilihat pada Tabel 3.13

Tabel 3.13 Desain Uji Coba Detail Perguruan Tinggi

Test Case ID	Tujuan	Input	Output yang diharapkan
	Memastikan lokasi perguruan tinggi sesuai dengan data	Memilih detail profil perguruan tinggi dan mengklik alamatnya	Hasil lokasi perguruan tinggi yang ditampilkan di sesuai dengan
	Memastikan kadaluarsa prodi di tiap perguruan tinggi berfungsi dan sesuai dengan data	Memilih detail profil perguruan tinggi “Stmik Surabaya” kemudian melihat prodi yang disediakan	Tgl kadaluarsa yang sesuai dengan data

### 3.14 Perancangan Angket

Selain melakukan uji coba pada sistem, dilakukan pengujian ulang terhadap aplikasi ini oleh pengguna yang bersangkutan. Uji coba aplikasi dilakukan dengan memberikan angket kepada setiap pengguna yang telah mencoba Aplikasi Diagnosis kerusakan komputer dan penanganannya. Angket berisi pertanyaan dan pilihan jawaban mengenai aplikasi.

Angket terdiri dari beberapa pertanyaan, yang setiap jawabannya memiliki skala nilai tertentu. Skala penilaian dibuat untuk memudahkan pengguna memberikan penilaian dalam bentuk kuantitatif. Skala penilaian ini yang nantinya akan digunakan dalam proses perhitungan skor, untuk menentukan tingkat kelayakan penggunaan aplikasi.

Pertanyaan yang diajukan berjumlah sembilan pertanyaan, yang mewakili penilaian tentang tampilan, menu, atau fitur yang terdapat pada aplikasi, baik aplikasi dalam *versi web* maupun *mobile web*. Setiap pertanyaan terdiri dari jawaban berupa skala penilaian menggunakan angka 1 sampai 5. Angka 5 merupakan nilai tertinggi yang mewakili penilaian sangat baik, angka 4 mewakili penilaian baik, angka 3 mewakili penilaian cukup, angka 2 mewakili penilaian kurang dan angka 1 mewakili penilaian sangat kurang. Rancangan angket yang akan diberikan kepada pengguna umum seperti yang terlihat pada Tabel 3.14.

Tabel 3.14 Rancangan Angket Pengguna Umum

No.	Pertanyaan	Penilaian (skor)					Jumlah
		5	4	3	2	1	
1.	Apakah tampilan setiap halaman <i>web</i> sudah jelas?						
2.	Apakah tampilan setiap halaman <i>mobile application</i> sudah jelas?						
3.	Apakah informasi tentang Perguruan Tinggi di aplikasi ini sudah jelas?						
4.	Apakah informasi tentang Program Studi di tiap2 Perguruan Tinggi di Aplikasi ini sudah jelas?						
5.	Apakah fitur tambahan tentang Rute Bemo & Kos2an di aplikasi ini diperlukan menurut anda?						
6.	Apakah setelah memakai aplikasi ini anda sudah bisa menentukan Perguruan Tinggi mana yg akan anda pilih?						
7.	Apakah setelah memakai aplikasi ini anda sudah bisa menentukan jurusan apa yg akan anda pilih?						
Jumlah skor hasil pengumpulan data							
Prosentase hasil uji coba							
Saran dan Kritik Anda :							

### 3.15 Konsep Pengambilan Data dari Google Maps

Pertama-tama yang dibutuhkan adalah Google Maps API. Google Maps Application Interface (API) merupakan suatu fitur aplikasi yang di keluarkan oleh google untuk memfasilitasi pengguna yang ingin mengintegrasikan Google Maps ke dalam website masing-masing dengan menampilkan data point milik sendiri.



Dengan menggunakan Google Maps API, Google Maps dapat di pakai pada website eksternal. Agar aplikasi Google Maps dapat muncul di website tertentu, diperlukan adanya API key. API key merupakan kode unik yang di generasikan oleh google untuk suatu website tertentu, agar server Google Maps dapat mengenalinya.

GoogleMaps API terdapat fitur yang memungkinkan pengguna untuk mencari tahu angka suatu koordinat kedalam suatu program, yaitu geocode alamat. *Geocoding* (geocode alamat) adalah proses merubah alamat-alamat ke koordinat geografi ( latitude dan longitude) sehingga dapat menandai posisi pada peta (Developers, 2014). *Geocoding* memungkinkan menerjemahkan antara alamat jalan dengan bujur/ lintang koordinat peta. Hal ini dapat memberikan konteks dikenali untuk lokasi dan koordinat yang digunakan dalam layanan berbasis lokasi dan peta berbasis *Activity*. Pencarian *geocoding* dilakukan di *server*, sehingga aplikasi akan meminta untuk memasukkan sebuah izin penggunaan internet. Kelas *geocoder* menyediakan akses untuk dua fungsi *geocoding*:

1. *ForwardGeocoding*: mencari lintang dan bujur alamat.
2. *ReverseGeocoding*: Mencari alamat dan jalan untuk sesuai lintang dan bujur.