

BAB 5

KESIMPULAN DAN SARAN

5.1. Kesimpulan

Setelah sistem MPathFinding ini selesai diimplementasikan maka dapat ditarik beberapa kesimpulan, yaitu:

1. Perangkat lunak MPathFinding telah berhasil dibangun dengan memanfaatkan pustaka *HttpConnection* pada J2ME untuk mengoneksikan perangkat mobile ke internet serta memanfaatkan pustaka tambahan *Location API* (JSR-179) untuk mendapatkan data lokasi device secara *real time*.
2. Perangkat Lunak MPathFinding telah berhasil dikembangkan menggunakan teknologi *Google Maps API* untuk menampilkan peta lokasi device secara *real time* pada aplikasi mobile yaitu dengan memanfaatkan pustaka tambahan *Location API* (JSR-179) untuk mendapatkan posisi secara *real time* kemudian mengirimkan data lokasi serta perintah khusus ke server *Google* dengan pustaka *HttpConnection* sehingga mendapatkan kembalian berupa gambar peta.
3. Perangkat Lunak MPathFinding telah berhasil diimplementasikan sebagai aplikasi yang dapat menemukan landmark terdekat dengan menghitung jarak posisi mobile dengan landmark yang tersedia dalam web service dengan menggunakan pendekatan Haversine Formula.
4. Dari hasil pengujian dapat diketahui bahwa fungsi-fungsi yang disediakan oleh perangkat lunak

MPathFinding berjalan dengan benar dan sesuai dengan yang diharapkan.

5.2. Saran

Penulis ingin memberikan beberapa saran untuk pengembangan lebih lanjut perangkat lunak MPathFinding ini:

1. Memperbanyak data-data Landmark tidak hanya sebatas kota Yogyakarta.
2. Menambahkan fasilitas *Chating*, sehingga user yang online dapat saling berkomunikasi dengan user lain, sehingga tidak lagi menggunakan fasilitas SMS (*Short Message Service*).
3. Menambahkan status sehingga user dapat memberikan informasi ke user lain apa yang sedang di kerjakan.
4. Memecah-mecah peta lebih baik daripada menggunakan peta static dimana jauh lebih ringan dan lebih dinamis.

Daftar Pustaka

Indrajani, Martin, 2004, ***Pemrograman Berorientasi Objek dengan JAVA***, Elex Media Komputindo

Budi Raharjo, Imam Hermantyo, Arif Haryono, 2007
Tuntunan Pemrograman JAVA Untuk Handphone,
Informatika Bandung

Mehdi Achour, Friedhelm Betz, Antony Dovgal, Nuno
Lopes, Philip Olson, Georg Richter, Damien Seguy,
Jakub Vrana, 2009, ***PHP Manual***, per 10 Januari
2009, <http://www.php.net/download-docs.php>

Lam Jason, 2008, ***PHP Interacting with J2ME***,
per 10 Januari 2009,
http://www.devarticles.com/c/a/Java/PHP_Interacting_with_J2ME/

Econym.org.uk, 2002, ***Google Map API Tutorial***, per 10
Januari 2009, <http://econym.org.uk/gmap>



SKPL

SPESIFIKASI KEBUTUHAN PERANGKAT LUNAK

MPathFinding

(Aplikasi Berbasis Lokasi Untuk Pencarian
Jalur)

Untuk :

Universitas Atma Jaya Yogyakarta

Dipersiapkan oleh:

Anak Agung Krisna Putra / 4546

Program Studi Teknik Informatika

Fakultas Teknologi Industri

Universitas Atma Jaya Yogyakarta

	Nomor Dokumen		Halaman
	SKPL-MPF		1/37
	Revisi		16 sep 2009
Program Studi Teknik Informatika Fakultas Teknologi Industri			

DAFTAR PERUBAHAN

Revisi	Deskripsi
A	
B	
C	
D	
E	
F	

INDEX TGL	-	A	B	C	D	E	F	G
Ditulis oleh	AAKP							
Diperik sa oleh	YSP							
Disetuj ui oleh								

Daftar Halaman Perubahan

Halaman	Revisi	Halaman	Revisi

Daftar Isi

Pendahuluan	89
Tujuan	89
Lingkup Masalah	89
Definisi, Akronim dan Singkatan	90
API	90
GPS	90
GUI	90
Referensi	91
Deskripsi umum (Overview)	91
Deskripsi Kebutuhan	92
Perspektif produk	92
Kebutuhan khusus	96
Kebutuhan antarmuka eksternal	96
Antarmuka pemakai	96
Antarmuka perangkat keras	96
Antarmuka perangkat lunak	97
Antarmuka Komunikasi	97
Kebutuhan fungsionalitas Perangkat Lunak	98
Use Case Diagram	98
Spesifikasi Rinci Kebutuhan Fungsionalitas	99
Spesifikasi Kebutuhan Fungsionalitas	99
4.2.1 Use case Spesification : <i>Registrasi Member</i>	99
Entity Relationship Diagram (ERD)	110
Kamus Data	111
Data User	111
Data Friend	111
Data Locations	111

Data JenisLandmarks	112
Data Landmarks	112



Daftar Gambar

Pendahuluan	89
Tujuan	89
Lingkup Masalah.....	89
Definisi, Akronim dan Singkatan.....	90
API	90
GPS	90
GUI	90
Referensi.....	91
Deskripsi umum (Overview).....	91
Deskripsi Kebutuhan	92
Perspektif produk.....	92
Kebutuhan khusus	96
Kebutuhan antarmuka eksternal	96
Antarmuka pemakai	96
Antarmuka perangkat keras.....	96
Antarmuka perangkat lunak.....	97
Antarmuka Komunikasi	97
Kebutuhan fungsionalitas Perangkat Lunak.....	98
Use Case Diagram.....	98
Spesifikasi Rinci Kebutuhan Fungsionalitas	99
Spesifikasi Kebutuhan Fungsionalitas	99
4.2.1 Use case Spesification : <i>Registrasi Member</i>	99
Entity Relationship Diagram (ERD)	110
Kamus Data	111
Data User.....	111
Data Friend.....	111
Data Locations	111
Data JenisLandmarks	112
Data Landmarks.....	112

Pendahuluan

Tujuan

Tujuan dari dokumen spesifikasi kebutuhan perangkat lunak (SKPL) dalam pengembangan perangkat lunak MPathFinding (Aplikasi *Path Finder* dengan Peta online berbasis GPS dan J2ME) yaitu mendefinisikan kebutuhan perangkat lunak yang meliputi antarmuka eksternal (antarmuka antara sistem dengan sistem lain perangkat lunak dan perangkat keras, dan pengguna) performansi (kemampuan perangkat lunak dari segi kecepatan, tempat penyimpanan yang dibutuhkan, serta keakuratan), dan atribut (*feature-feature* tambahan yang dimiliki sistem), serta mendefinisikan fungsi perangkat lunak. SKPL ini juga mendefinisikan batasan perancangan perangkat lunak.

Dokumen SKPL ini menjadi dasar kesepakatan antara pihak pelanggan dengan pihak pengembang mengenai perangkat lunak yang akan dikembangkan.

Lingkup Masalah

Perangkat lunak MPathFinding dikembangkan dengan tujuan untuk :

- d. Memberikan informasi mengenai koordinat (Latitude, Longitude) posisi device pada saat itu (real time).
- e. Mentransformasikan dan menampilkan informasi posisi real time device ke dalam Google Map di perangkat genggam.
- f. Menampilkan Landmark terdekat yang menunjukkan lokasi dari tempat-tempat tertentu beserta

jalurnya didalam Google Map melalui perangkat device.

Dan berjalan pada perangkat mobile yang mendukung platform Java Midlet (J2ME)

Definisi, Akronim dan Singkatan

Daftar definisi akronim dan singkatan :

Keyword/Phrase	Definisi
SKPL	Merupakan spesifikasi kebutuhan dari perangkat lunak yang akan dikembangkan.
SKPL-MPF-XXX	Kode yang merepresentasikan kebutuhan pada MPathFinding (Aplikasi Path Finding dengan Peta Online berbasis GPS dan J2ME)
MPathFinding	Perangkat lunak untuk mencari tempat-tempat terdekat dari posisi kita secara <i>real time</i> .
J2ME	Java 2 Micro Edition, Merupakan versi Java API yang dikhususkan untuk piranti kecil dan mobile.
API	<i>Application Programming Interface</i> , merupakan kumpulan fungsi, prosedur atau class yang menyediakan service atau <i>library</i> tertentu.
GPS	<i>Global Positioning System</i> , merupakan perangkat untuk mengetahui posisi koordinat di permukaan bumi melalui system satelite.
GUI	<i>Graphical User Interface</i> , merupakan tampilan antarmuka program untuk pengguna.

Referensi

Dokumen yang digunakan sebagai acuan dalam rencana pengembangan perangkat lunak ini adalah :

1. Purvis, Michael, Sambells, Jeffrey dan Turner, Cameron, *Beginning Google Maps Application with PHP and Ajax Form Novoce to Professional*, Appress, United States of America.
2. Software Engineering, 1997, Roger S. Pressman, Mc Graw-Hill International Edition.
3. Stefanov, Stoyan, *Object-Oriented JavaScript*, Packt Publising, Birmingham, 2008.
4. Developing Mobile Applications Using J2ME, NIIT, 2002.

Deskripsi umum (Overview)

Secara umum dokumen SKPL ini terbagi atas 3 bagian utama. Bagian utama berisi penjelasan mengenai dokumen SKPL tersebut yang mencakup tujuan pembuatan SKPL, ruang lingkup masalah dalam pengembangan perangkat lunak tersebut, definisi, referensi dan deskripsi umum tentang dokumen SKPL ini.

Bagian kedua berisi penjelasan umum tentang perangkat lunak MPathFinding yang akan dikembangkan, mencakup perspektif produk yang akan dikembangkan, fungsi produk perangkat lunak, karakteristik pengguna, batasan dalam penggunaan perangkat lunak dan asumsi yang dipakai dalam pengembangan perangkat lunak MPathFinding.

Bagian ketiga berisi penjelasan secara lebih rinci tentang kebutuhan perangkat lunak MPathFinding yang akan dikembangkan.

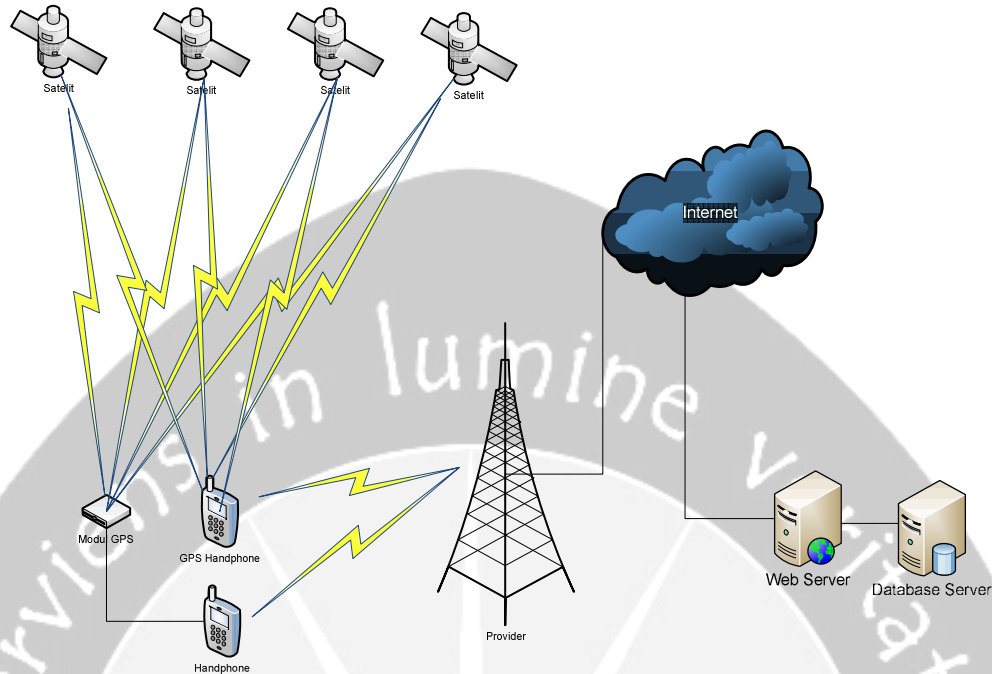
Deskripsi Kebutuhan

Perspektif produk

MPathFinding merupakan perangkat lunak atau aplikasi yang dikembangkan untuk membantu mencari tempat terdekat dan mengetahui posisi teman yang sedang online. Aplikasi ini memberikan informasi mengenai posisi real device atau handphone user. Posisi real time perangkat mobile ini akan ditampilkan pada peta menggunakan Google Maps API. Peta digital pada aplikasi ini memiliki layanan informasi mengenai tempat-tempat umum (seperti pom bensin, hotel, universitas, kantor polisi, dsb) dan posisi teman yang online.

Aplikasi MPathFinding ini memberikan layanan yang berjalan pada platform Java pada perangkat mobile dan dibuat dengan menggunakan bahasa pemrograman J2ME (Java 2 Micro Edition) dengan lingkungan emulator menggunakan NetBeans 6.7 dan Sun WTK(Wireless Tool Kit) 2.5.2.

Pengguna akan berinteraksi dengan system melalui antarmuka GUI. Pada system ini, seperti yang terlihat pada gambar 1, arsitektur perangkat lunak yang digunakan bersifat *client server*. Data lokasi yang diterima modul GPS dari satelit ditransmisikan ke handphone. Kemudian aplikasi akan mengirimkan data lokasi ke web service, kemudian data tersebut akan disimpan di database server. Web service kemudian menggunakan data tersebut untuk menentukan jarak tempat terdekat dan posisi *real time* dari client(user).



Gambar 1. Arsitektur Perangkat Lunak MPathFinding

1.3 Fungsi Produk

Fungsi produk perangkat lunak MPathFinding adalah sebagai berikut:

1. Fungsi *Register Member* (**SKPL-MPF-001**).
Merupakan fungsi yang digunakan untuk menambahkan data user baru.
2. Fungsi *Display Friend Online* (**SKPL-MPF-002**).
Merupakan fungsi untuk menampilkan teman yang sedang online.
3. Fungsi *Manage User* (**SKPL-MPF-003**).
Merupakan fungsi yang digunakan untuk mengatur user lain.

Fungsi *Manage Friend* mencakup:

- a. Fungsi *Adding Friend* (**SKPL-MPF-003-01**)

Merupakan fungsi untuk meminta persetujuan user lain untuk menunjukkan lokasinya.

b. Fungsi *Request Friend* (**SKPL-MPF-003-02**)

Merupakan fungsi untuk menyetujui atau tidak permintaan user lain untuk mengetahui posisi *device*.

c. Fungsi *Delete Friend* (**SKPL-MPF-003-03**)

Merupakan fungsi untuk menghapus user lain dari list teman.

4. Fungsi *View Map* (**SKPL-MPF-004**).

Merupakan fungsi untuk menampilkan peta posisi *device*, jalur ke suatu tempat dan jalur ke user lain.

5. Fungsi *Searching Near Location* (**SKPL-MPF-005**).

Merupakan fungsi untuk mencari tempat terdekat dari user(*device*).

6. Fungsi *View Route* (**SKPL-MPF-006**).

Merupakan fungsi untuk menampilkan rute ke suatu tempat atau rute ke user lain.

7. Fungsi *Login* (**SKPL-MPF-07**).

Merupakan Fungsi yang digunakan oleh client untuk dapat masuk ke dalam system yang akan digunakan.

8. Fungsi *Manage Profile*(**SKPL-MPF-08**).

Merupakan Fungsi yang digunakan untuk mengelola data profile user.

Fungsi *Manage Profile* mencakup:

a. Fungsi *Change Password* (**SKPL-MPF-008-01**).

Merupakan fungsi yang digunakan untuk mengubah password user.

b. Fungsi *Change Phone Number*(**SKPL-MPF-008-02**).

Merupakan fungsi yang digunakan untuk mengubah data phone number dari client.

c. Fungsi Change Nama User (**SKPL-MPF-008-03**).

Merupakan fungsi yang digunakan untuk mengubah nama user.

9. Fungsi *Show My Location* (**SKPL-MPF-09**).

Merupakan fungsi yang digunakan untuk menampilkan lokasi user secara *real time* pada peta digital.

10. Fungsi Send SMS (**SKPL-MPF-10**).

Merupakan fungsi yang digunakan untuk mengirim SMS (*Short Message Service*) kepada teman (user lain) yang online.

2.3 Karakteristik Pengguna

Karakteristik dari pengguna perangkat lunak MPathFinding adalah sebagai berikut :

1. Memahami pengoperasian handpone.
2. Mengerti tentang konsep GPS.
3. Mengerti cara kerja perangkat kerja yang dipakai.

2.4 Batasan-batasan

Batasan-batasan dalam pengembangan perangkat lunak MPathFinding tersebut adalah:

1. Kebijakan umum
Berpedoman pada tujuan dari pengembangan perangkat lunak MPathFinding.
2. Keterbatasan perangkat keras
Dapat diketahui kemudian setelah system ini berjalan (sesuai dengan kebutuhan).

2.5 Asumsi dan Ketergantungan

Asumsi yang digunakan dalam pengembangan perangkat lunak MPathFinding yaitu:

1. Sistem ini hanya dapat dijalankan pada perangkat mobile (Handphone, SmartPhone, PDA, dll) yang mendukung platform Java MIDP 2.0 dengan spesifikasi JSR-179.
2. Tersedia komputer Server dengan spesifikasi prosesor 3.0 GHZ, memori primer (RAM) minimal 1024 MB, spasi yang tersimpan dalam media penyimpanan sekunder (Hardisk) server yaitu 1 Terabyte.

Kebutuhan khusus

Kebutuhan antarmuka eksternal

Kebutuhan antar muka eksternal pada perangkat lunak MPathFinding meliputi kebutuhan antarmuka pemakai, antarmuka perangkat keras, antarmuka perangkat lunak, antarmuka komunikasi.

Antarmuka pemakai

Pengguna berinteraksi dengan antarmuka yang ditampilkan dalam bentuk form dan gambar.

Antarmuka perangkat keras

Antarmuka perangkat keras yang digunakan dalam perangkat lunak MPathFinding adalah:

3. Perangkat mobile (Handphone, SmartPhone, PDA, dsb).
4. Perangkat Modul GPS yang compatible dengan perangkat mobile yang digunakan.

Antarmuka perangkat lunak

Perangkat lunak yang dibutuhkan untuk mengoperasikan perangkat lunak MPathFinding adalah sebagai berikut :

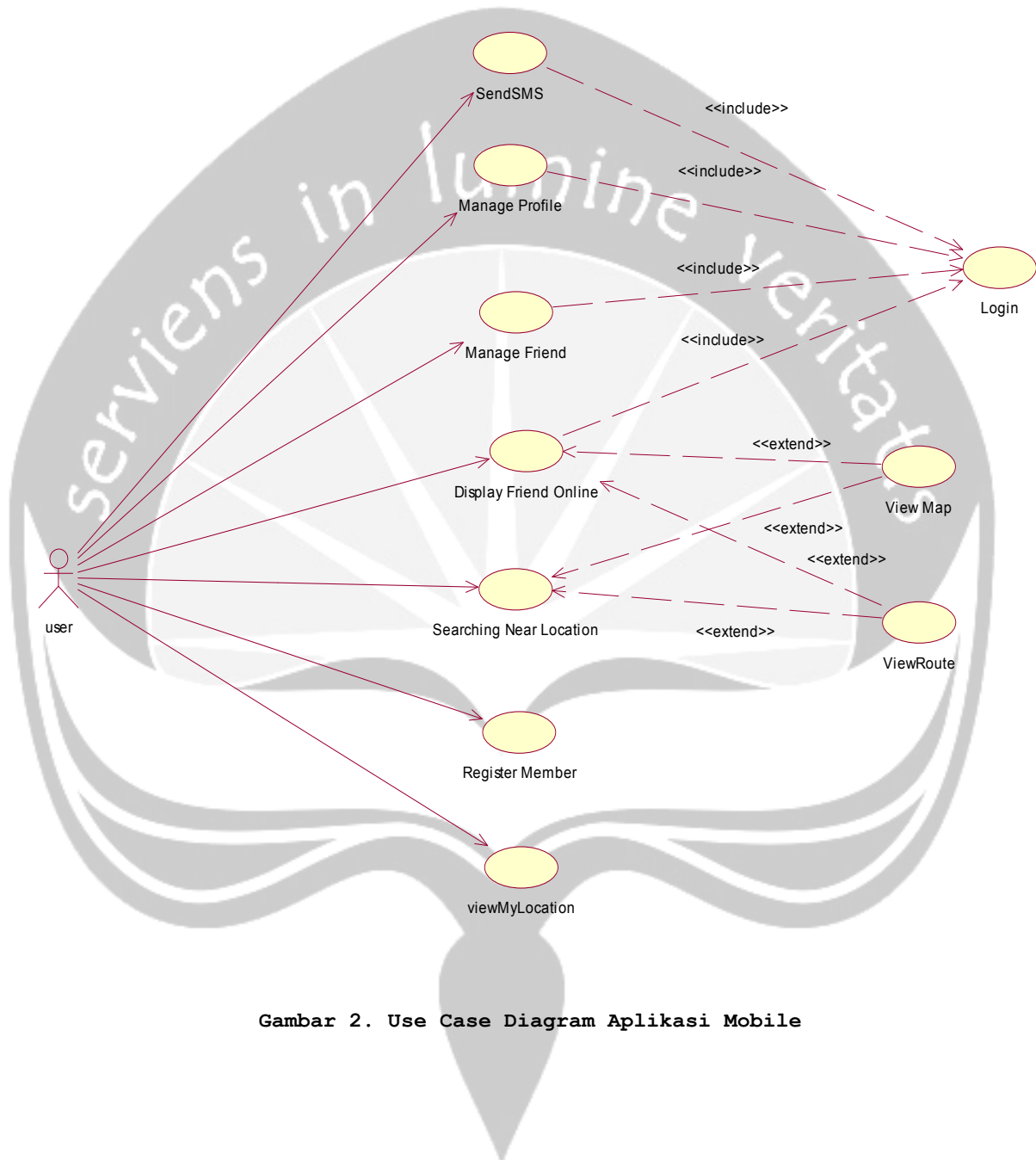
7. Nama : MIDP&CLDC
Sumber : Sun Microsystems.
Sebagai *platform environment* pada perangkat mobile.
8. Nama : Symbian OS
Sumber : Symbian Software.
Sebagai sistem operasi untuk perangkat mobile.
9. Nama : Apache
Sumber : Apache.
Sebagai Web Server.
10. Nama : MySql
Sumber : MySql.
Sebagai *Data Base Management System*.
11. Nama : Google Maps API
Sumber : Google
Sebagai API dalam pembangunan aplikasi peta.
12. Nama : Location API
Sumber : Sun Java
Sebagai library tambahan untuk menggunakan data berbasis lokasi.

Antarmuka Komunikasi

Antarmuka komunikasi perangkat lunak MPathFinding menggunakan protocol Location API dan protokol HTTP.

Kebutuhan fungsionalitas Perangkat Lunak

Use Case Diagram



Gambar 2. Use Case Diagram Aplikasi Mobile

Spesifikasi Rinci Kebutuhan Fungsionalitas

Spesifikasi Kebutuhan Fungsionalitas

4.2.1 Use case Spesification : *Registrasi Member*

1. Brief Description

Use Case ini memungkinkan user untuk melakukan entry data user.

2. Primary Actor

1. User

3. Supporting Actor

none

4. Basic Flow

1. Use Case ini dimulai ketika aktor masuk ke dalam aplikasi pada *mobile device* pertama kali.
2. Sistem menampilkan form entry data member.
3. Aktor memasukkan data member.
4. Aktor meminta sistem untuk menyimpan data user.
5. Sistem akan mengecek data member yang telah diinputkan.

E-1 Data member yang diinputkan aktor salah.

E-2 Data member yang diinputkan aktor sudah terdapat dalam aplikasi.

6. Sistem menyimpan data user ke database.
7. Use Case ini selesai

5. Alternative Flow

none

6. Error Flow

E-1 Data member yang diinputkan aktor salah.

1. Sistem menampilkan peringatan bahwa data member tidak sesuai
2. Kembali ke Basic Flow langkah ke 3.

E-2 Data User yang diinputkan aktor sudah terdapat dalam aplikasi.

1. Sistem menampilkan peringatan bahwa phone number dan nama user sudah ada dalam system.
2. Kembali ke Basic Flow langkah ke 3.

7. PreConditions

none

8. PostConditions

1. Aktor memasuki sistem dan dapat menggunakan fungsi-fungsi pada sistem sesuai dengan hak aksesnya masing-masing sesuai tipe user.

4.2.2 Use case Spesification : Display Friend online

1. Brief Description

Use case ini digunakan aktor untuk menampilkan data teman yang sedang online.

2. Primary Actor

1. user

3. Supporting Actor

none

4. Basic Flow

1. Use Case ini dimulai ketika aktor memilih untuk menampilkan data teman-teman yang sedang online.
2. Sistem menampilkan teman-teman dari user yang sedang online.
3. Use case selesai.

5. Alternative Flow

none

6. Error Flow

none

7. PreConditions

1. Use Case Login User telah dilakukan

2. Aktor telah memasuki sistem

8. PostConditions

none

4.2.3 Use case Spesification : *Manage Profile*

1. Brief Description

Use Case ini memungkinkan user untuk mengelola datanya sendiri. Aktivitas yang dilakukan oleh actor meliputi entry, edit dan display data user.

2. Primary Actor

1. User

3. Supporting Actor

none

4. Basic Flow

1. Use Case ini dimulai ketika aktor memilih untuk mengelola data user.
2. Sistem menampilkan pilihan untuk melakukan *change password, change phone number, change name user*.
3. Aktor memilih untuk melakukan *change password*.
 A-1 Aktor memilih untuk melakukan *change phone number*.
 A-2 Aktor memilih untuk melakukan *change name user*.
4. Aktor memasukkan password user yang lama dan password user baru.
5. Aktor meminta sistem untuk menyimpan data user.
6. Sistem akan mengecek data password user yang telah diinputkan.
 E-1 Data password user yang di inputkan user salah.
7. Sistem menyimpan data user ke database.
8. Use Case selesai.

5. Alternative Flow

A-1 Aktor memilih untuk melakukan *change phone number*.

1. Sistem menampilkan data user.
2. Aktor memasukan memasukkan *phone number* baru.
3. Aktor meminta sistem menyimpan data *phone number* user yang baru.
4. Sistem melakukan pengecekan terhadap data *phone number* user yang telah diubah.

E-2 Data *phone number* yang telah diubah salah.

5. Sistem akan menyimpan data *phone number* user ke dalam database.
6. Berlanjut ke Basic flow langkah ke 8.

A-2 Aktor memilih untuk melakukan *change name user*

1. Sistem menampilkan data user.
2. Aktor memasukan memasukkan nama user baru.
3. Aktor meminta sistem menyimpan data nama user yang baru.
4. Sistem melakukan pengecekan terhadap data nama user yang telah diubah.

E-3 Data nama user yang telah diubah salah.

5. Sistem akan menyimpan data nama user ke dalam database.
6. Berlanjut ke Basic flow langkah ke 8.

6. Error Flow

E-1 Data password user yang di inputkan user salah.

1. Sistem akan memberikan pesan peringatan bahwa data password yang telah di inputkan salah.
2. Kembali ke Basic Flow langkah ke 4.

E-2 Data *phone number* yang telah diubah salah.

1. Sistem akan memberikan pesan peringatan bahwa data *phone number* user yang baru yang telah diinputkan user salah.

2. Kembali ke Alternative Flow A-1 langkah ke 2.
- E-3 Data nama user yang telah diubah salah.
1. Sistem akan memberikan pesan peringatan bahwa data nama user yang baru yang telah diinputkan user salah.
 2. Kembali ke Alternative Flow A-2 langkah ke 2.

7. PreConditions

1. Aktor telah melewati use case login
2. Aktor telah memasuki sistem

8. PostConditions

1. Aktor telah melakukan manage profile.

4.2.4 Use case Spesification : *Manage Friend*

1. Brief Description

Use Case ini memungkinkan user memungkinkan user untuk mengelola data teman (user lain). Aktivitas yang dilakukan oleh actor meliputi *adding*, *request*, dan *delete* user lain.

2. Primary Actor

1. User

3. Supporting Actor

none

4. Basic Flow

1. Use Case ini dimulai ketika aktor memilih untuk mengelola data user lain.
2. Sistem menampilkan pilihan untuk melakukan *adding friend*, *request friend*, dan *delete friend*.
3. Aktor memilih untuk melakukan *adding friend*.
 - A-1 Aktor memilih untuk melakukan *request friend*.
 - A-2 Aktor memilih untuk melakukan *delete friend*.
4. Aktor akan memasukkan nomor telepon user yang hendak di tambahkan(*adding*).

5. Sistem akan mengecek nomor telepon yang telah diinputkan.

E-1 Nomor telepon yang diinputkan salah.

6. Sistem akan menyimpan data *Adding Friend* ke database.
7. Use Case selesai

5. Alternative Flow

A-1 Aktor memilih untuk melakukan *request friend*.

1. Sistem menampilkan data user yang me-request.
2. Aktor akan meng-*Confirm* user yang telah me-request.

A-1-1 Aktor meng-*Ignore* user yang telah me-request.

3. Sistem mengirim data *requesting* ke web server untuk kemudian disimpan di basis data.
4. Kembali ke Basic Flow langkah ke 7.

A-2 Aktor memilih untuk melakukan *delete friend*.

1. Sistem akan menampilkan semua teman atau user lain.
2. Aktor memilih user lain yang ingin dihapus
3. Aktor menghapus data user lain yang sudah dipilih.
4. Sistem melakukan penghapusan terhadap user lain dari list teman user
5. Kembali ke Basic Flow langkah ke 7.

A-1-1 Aktor meng-*Ignore* user yang telah me-request.

1. Sistem akan mengirim ke webserver untuk meng-*ignore* data menghapus data *requesting*.
2. Kembali ke Basic Flow langkah ke 7.

6. Error Flow

E-1 Nomor telepon yang diinputkan salah

1. Sistem memberikan pesan peringatan bahwa data yang diinputkan salah
2. Kembali ke Basic Flow Langkah ke 4

7. PreConditions

1. Use Case Login User sudah dilakukan
2. Aktor telah memasuki sistem

8. PostConditions

1. Aktor telah melakukan *Manage User*.

4.2.5 Use case Spesification : View Map

1. Brief Description

Use Case ini memungkinkan user untuk menampilkan lokasi user, menampilkan *directions* ke suatu tempat dan menampilkan *directions* ke lokasi teman yang dipilih pada peta yang ditampilkan pada *mobile device*.

2. Primary Actor

1. User

3. Supporting Actor

None

4. Basic Flow

1. Use Case ini dimulai ketika aktor memilih untuk menampilkan peta dalam pada *device mobile*.
2. Sistem menampilkan data lokasi user , *directions* ke suatu tempat atau menampilkan *directions* ke lokasi teman yang sedang online berupa komponen *image*.
3. Use Case selesai

5. Alternative Flow

none

6. Error Flow

None

7. PreConditions

1. Aktor telah memasuki sistem

8. PostConditions

1. Gambar peta ditampilkan pada layar.

4.2.6 Use case Spesification : Searching Near Location

9. Brief Description

Use Case ini memungkinkan user untuk menampilkan 4 lokasi yang memiliki jarak terdekat sesuai dengan tipe yang dipilih oleh user.

10. Primary Actor

1. User

11. Supporting Actor

none

12. Basic Flow

1. Use Case ini dimulai ketika aktor memilih untuk melakukan *Searching Near Location* sesuai dengan tipe lokasi yang dipilih user.
2. Sistem akan memberikan 4 lokasi yang terdekat sesuai dengan tipe yang dipilih oleh user.
3. Use Case selesai

13. Alternative Flow

none

14. Error Flow

none

15. PreConditions

2. Aktor telah memasuki sistem

16. PostConditions

1. System menampilkan lokasi terdekat.

4.2.7 Use case Spesification : View Route

3. Brief Description

Use Case ini memungkinkan user untuk mengetahui jalur yang harus ditempuh, guna mencapai tujuan ke lokasi atau user lain yang di tampilkan dalam peta.

4. Primary Actor

2. User

1. Supporting Actor

none

2. Basic Flow

1. Use Case ini dimulai ketika aktor memilih untuk melakukan *route* tempat atau user lain yang ditampilkan pada peta dalam *mobile device*.
2. Sistem akan menampilkan jalur menuju tempat atau user lain yang dituju.
3. Use Case selesai.

3. Alternative Flow

None.

4. Error Flow

None.

5. PreConditions

1. Aktor telah memasuki sistem.

6. PostConditions

1. System menampilkan jalur yang ditempuh untuk ke tujuan.

4.2.8 Use case Spesification : Login

9. Brief Description

Use Case ini digunakan oleh actor untuk memperoleh akses ke sistem. Login di dasarkan sebuah id unik yaitu phone number dari user, username dari user dan password.

10. Primary Actor

1. User.

11. Supporting Actor

None.

12. Basic Flow

7. Use Case ini dimulai ketika aktor memilih untuk melakukan login.

8. Sistem akan menampilkan antarmuka untuk login.

9. Aktor memasukkan phone number, username dan password.

10. Sistem akan memeriksa username dan phone number dan password yang diinputkan user.

E-1 Username, phone number, dan password tidak sesuai.

11. Sistem memberikan akses ke aktor.

12. Use Case selesai.

13. Alternative Flow

none

14. Error Flow

E-1 Username, phone number, dan password tidak sesuai.

3. Sistem memberikan pesan peringatan bahwa data username, phone number atau password yang diinputkan salah

4. Kembali ke Basic Flow Langkah ke 3.

15. PreConditions

3. Use Case Login User sudah dilakukan

4. Aktor telah memasuki sistem

16. PostConditions

b. Aktor telah memasuki system

4.2.9 Use case Spesification : Show My Location**1. Brief Description**

Use Case ini digunakan oleh actor untuk menampilkan lokasi user (*mobile device*) secara *real time* di peta digital pada *mobile device* sesuai dengan koordinat user.

2. Primary Actor

1. User.

3. Supporting Actor

None.

4. Basic Flow

1. Use Case ini dimulai ketika aktor memilih untuk melihat lokasinya pada peta.
2. Sistem menampilkan peta dengan icon yang menggambarkan posisi user.
3. Use Case selesai.

5. Alternative Flow

none

6. Error Flow

none

7. PreConditions

1. Aktor telah memasuki sistem.

8. PostConditions

1. System menampilkan gambar peta.

4.2.10 Use case Spesification : Send SMS**1. Brief Description**

Use Case ini digunakan oleh actor untuk mengirim SMS (*Short Message Service*) ke teman (user lain) yang online.

2. Primary Actor

1. User.

3. Supporting Actor

None.

4. Basic Flow

1. Use Case ini dimulai ketika aktor memilih untuk mengirim SMS ke teman yang online.
2. Sistem menampilkan antarmuka untuk mengirim SMS.
3. Sistem mengirim SMS ke teman lain yang terpilih.

4. Use Case selesai.

5. Alternative Flow

none

6. Error Flow

none

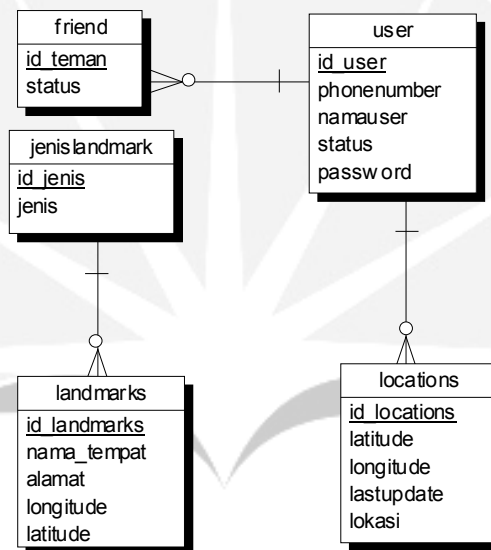
7. PreConditions

2. Aktor telah memasuki sistem.

8. PostConditions

2. System berhasil mengirim SMS.

Entity Relationship Diagram (ERD)



Gambar 3. Entity Relationship Diagram

Kamus Data

Data User

Elemen Data	Representasi	Domain	Range	Format	Presisi	Struktur Data
user_id	Untuk id user dari pengguna sistem	number	-	-	-	Number (11)
phonenumber	Untuk nomor handphone	Number	-	-	-	Varchar(20)
Namauser	Untuk nama user	Text	-	-	-	Varchar(20)
Password	Untuk password dari user	Text	-	-	-	Varchar(20)
Status	Untuk status dari user	Number	0-1	-	0=tidak aktif, 1=aktif	Number (11)

Data Friend

Elemen Data	Representasi	Domain	Range	Format	Presisi	Struktur Data
id_teman	Untuk Id teman	Number	-	-	-	Number(11)
status	Untuk status dari teman	number	-	-	0=request, 1=teman	Number (11)

Data Locations

Elemen Data	Representasi	Domain	Range	Format	Presisi	Struktur Data
id_locations	Untuk id locations user	Number	-	-	-	Number (11)
Latitude	Untuk koordinat latitude lokasi	Double	-	-	-	Double
Longitude	Untuk koordinat longitude dari lokasi	Double	-	-	-	Double
lastUpdate	Untuk data waktu	DateTime	-	-	-	DateTime

lokasi	Untuk lokasi user	text	-	-	-	Varchar (100)
--------	-------------------	------	---	---	---	---------------

Data JenisLandmarks

Elemen Data	Representasi	Domain	Range	Format	Presisi	Struktur Data
id_jenislandmark	Untuk id jenis landmark	Number	-	-	-	Number (11)
Jenis	Untuk jenis dari landmarks	text	-	-	-	Varchar (50)

Data Landmarks

Elemen Data	Representasi	Domain	Range	Format	Presisi	Struktur Data
id_landmark	Untuk id landmarks	Number	-	-	-	Number (11)
Nama_tempat	Untuk nama tempat landmark	Text	-	-	-	Varchar (500)
Alamat	Untuk alamat dari landmark	Alamat	-	-	-	Varchar (500)
Latitude	Untuk koordinat latitude landmark	Double	-	-	-	Double
Longitude	Untuk koordinat longitude dari landmark	Double	-	-	-	Double



DPPL

DESKRIPSI PERANCANGAN PERANGKAT LUNAK

MPathFinding

(Aplikasi Berbasis Lokasi untuk Pencarian Jalur)

Untuk :

Universitas Atma Jaya Yogyakarta

Dipersiapkan oleh:

Anak Agung Krisna Putra / 4546

Program Studi Teknik Informatika-Fakultas Teknologi Industri

Universitas Atma Jaya Yogyakarta

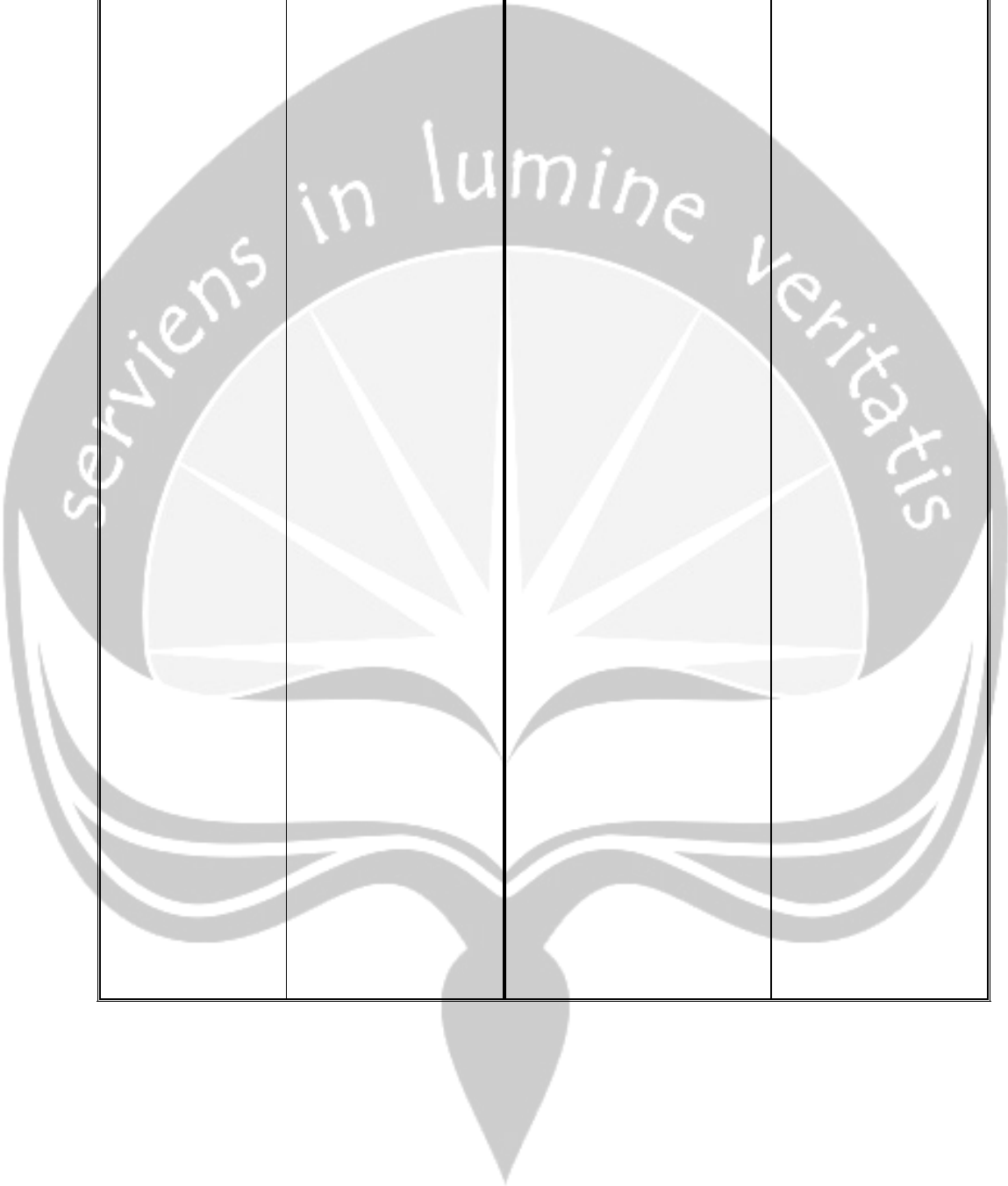
	Program Studi Teknik Informatika Fakultas Teknologi	Nomor Dokumen		Halaman
		<i>DPPL-MPF</i>		1/54
		Revisi		

DAFTAR PERUBAHAN

Revisi	Deskripsi
A	
B	
C	
D	
E	
F	
G	

INDEX TGL	-	A	B	C	D	E	F	G
Ditulis oleh								
Diperiksa oleh								
Disetujui oleh								

Daftar Halaman Perubahan

Halaman	Revisi	Halaman	Revisi
			

Daftar Isi

1Pendahuluan	120
1.1 Tujuan	120
1.2 Ruang Lingkup	120
1.3 Definisi dan Akronim	121
1.4 Referensi	121
2Perancangan Sistem	122
2.1 Perancangan Arsitektur.	122
2.1.1 Perancangan Arsitektur Aplikasi	122
2.2 Use Case	123
2.2.1 Registrasi Member	123
2.2.2 Display Friend Online	124
2.2.3 Manage Friend	125
2.2.4 View Map	128
2.2.5 Search Near Location	129
2.2.6 View Route	130
2.2.7 Login	131
2.2.8 Manage Profile	132
2.2.9 Show My Location	135
2.2.10 Send SMS	136
2.3 Diagram Kelas	137
2.3.1 Diagram Kelas Pada Perangkat Mobile	137
2.3.2 Diagram Kelas Pada Web Service	138
2.4 Deskripsi Kelas	139
2.4.1 Deskripsi Kelas Pada Perangkat Mobile	139
2.4.2 Deskripsi Kelas Pada Webservice	151
3Perancangan Data	157
3.1 Dekomposisi Data	157
3.1.1 Deskripsi Entitas Data user	157
3.1.2 Deskripsi Entitas Data Friend	157
3.1.3 Deskripsi Entitas Data Locations	158

3.1.4	Deskripsi Entitas Data JenisLandmark	158
3.1.5	Deskripsi Entitas Data Landmarks	158
3.2	Physical Data Model	159
4	Perancangan Antarmuka	160
4.1	Antarmuka Main Screen	160
4.2	Antarmuka Login	161
4.3	Antarmuka ChangeName	162
4.4	Antarmuka ChangePhonenumber	163
4.5	Antarmuka ChangePassword	164
4.6	Antarmuka DisplayFriendOnline	165
4.7	Antarmuka RegisterManager	166
4.8	Antarmuka SearchNearLocation	167
4.9	Antarmuka Route	169
4.10	Antarmuka AddingFriend	170
4.11	Antarmuka DeleteFriend	171
4.12	Antarmuka FriendRequest	172
4.13	Antarmuka ViewMap	173

Daftar Gambar

Gambar 2.1 Arsitektur Aplikasi MPathFinding di Mobile Device	123
Gambar 2.2 Sequence Diagram: Register Member	123
Gambar 2.3 Sequence Diagram: Display Friend Online	124
Gambar 2.4 Sequence Diagram: Adding Friend	125
Gambar 2.5 Sequence Diagram: Request Friend	126
Gambar 2.6 Sequence Diagram: Delete Friend	127
Gambar 2.7 Sequence Diagram Mobile : View Map	128
Gambar 2.8 Sequence Diagram : Search Near Location	129
Gambar 2.9 Sequence Diagram : View Route	130
Gambar 2.10 Sequence Diagram : Login	131
Gambar 2.11 Sequence Diagram : Change Password	132
Gambar 2.12 Sequence Diagram : Change Phone Number	133
Gambar 2.13 Sequence Diagram : Change Nama Number	134
Gambar 2.10 Sequence Diagram : Show My Location	135
Gambar 2.10 Sequence Diagram : Show My Location	136
Gambar 2.26 Diagram Kelas Perangkat Mobile	137
Gambar 2.26 Diagram Kelas Webservice	138
Gambar 3 Physical Data Model.....	159

Pendahuluan

Tujuan

Dokumen Deskripsi Perancangan Perangkat Lunak (DPPL) bertujuan untuk mendefinisikan perancangan perangkat lunak yang akan dikembangkan. Dokumen DPPL tersebut digunakan oleh pengembang perangkat lunak sebagai acuan untuk implementasi pada tahap selanjutnya.

Ruang Lingkup

Perangkat Lunak MPathFinding dikembangkan dengan tujuan untuk :

1. Memberikan informasi mengenai koordinat(*Latitude, Longitude*) posisi device pada saat itu(*real time*).
2. Mentransformasikan dan menampilkan informasi posisi device ke dalam Google Map pada *mobile device*.
3. Menampilkan *Landmark* yang menunjukkan lokasi dari tempat-tempat terdekat dalam Google Map pada *mobile device*.
4. Menampilkan user lain yang *online* yang menunjukkan lokasinya di dalam Google Map pada *mobile device*.

Definisi dan Akronim

Daftar definisi akronim dan singkatan :

Keyword/Phrase	Definisi
DPPL	Deskripsi Perancangan Perangkat Lunak disebut juga <i>Software Design Description</i> (SDD) merupakan deskripsi dari perancangan produk/perangkat lunak yang akan dikembangkan.
MPathFinding	Perangkat lunak untuk mencari tempat-tempat terdekat dari posisi kita secara <i>real time</i> .

Referensi

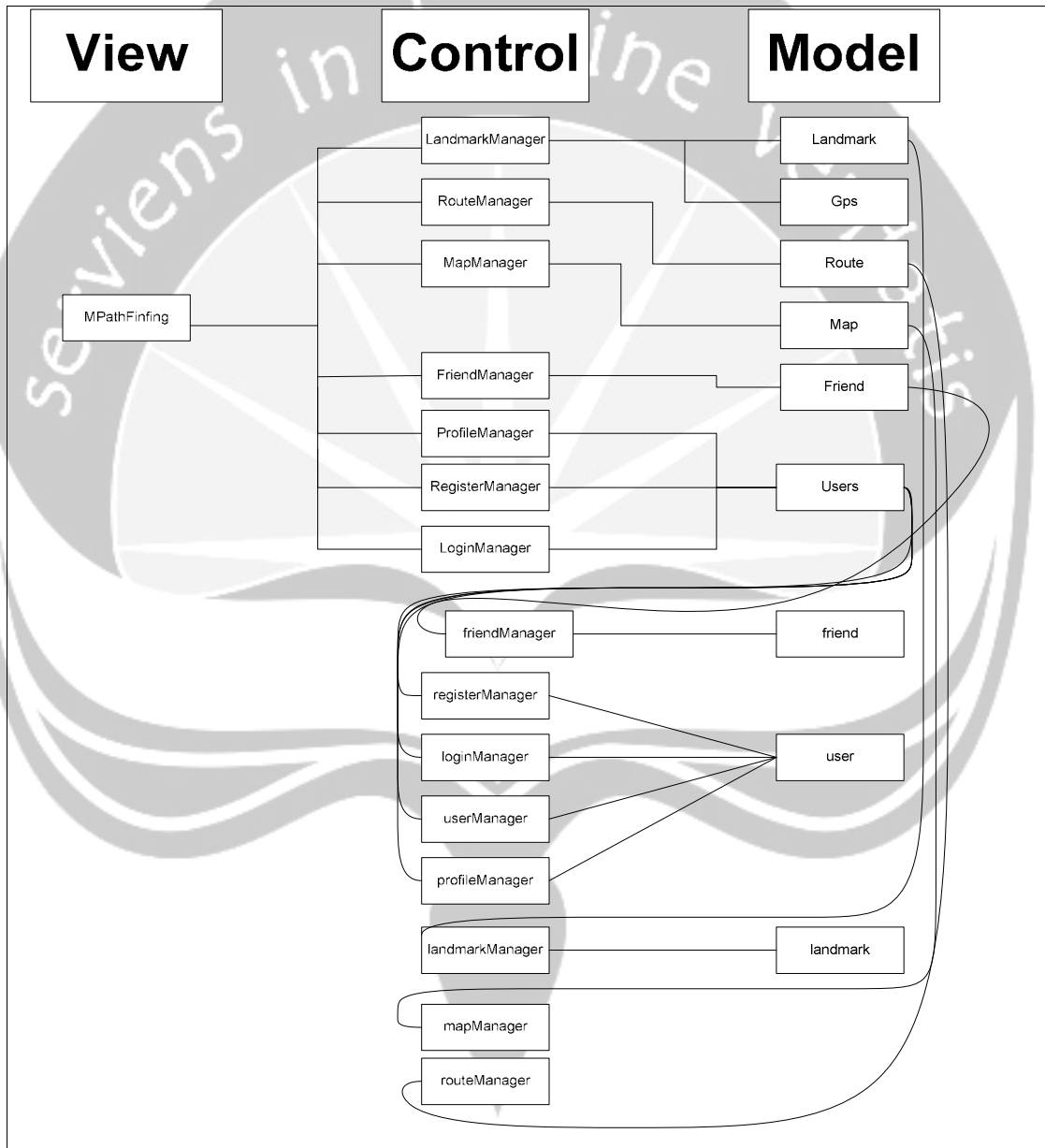
Referensi yang digunakan pada perangkat lunak tersebut adalah:

5. Purvis, Michael, Sambells, Jeffrey dan Turner, Cameron, *Beginning Google Maps Application with PHP and Ajax Form Novice to Professional*, Appress, United States of America.
6. Software Engineering, 1997, Roger S. Pressman, McGraw-Hill International Edition.
7. Stefanov, Stoyan, *Object-Oriented JavaScript*, Packt Publishing, Birmingham, 2008.
8. Developing Mobile Applications Using J2ME, NIIT, 2002.

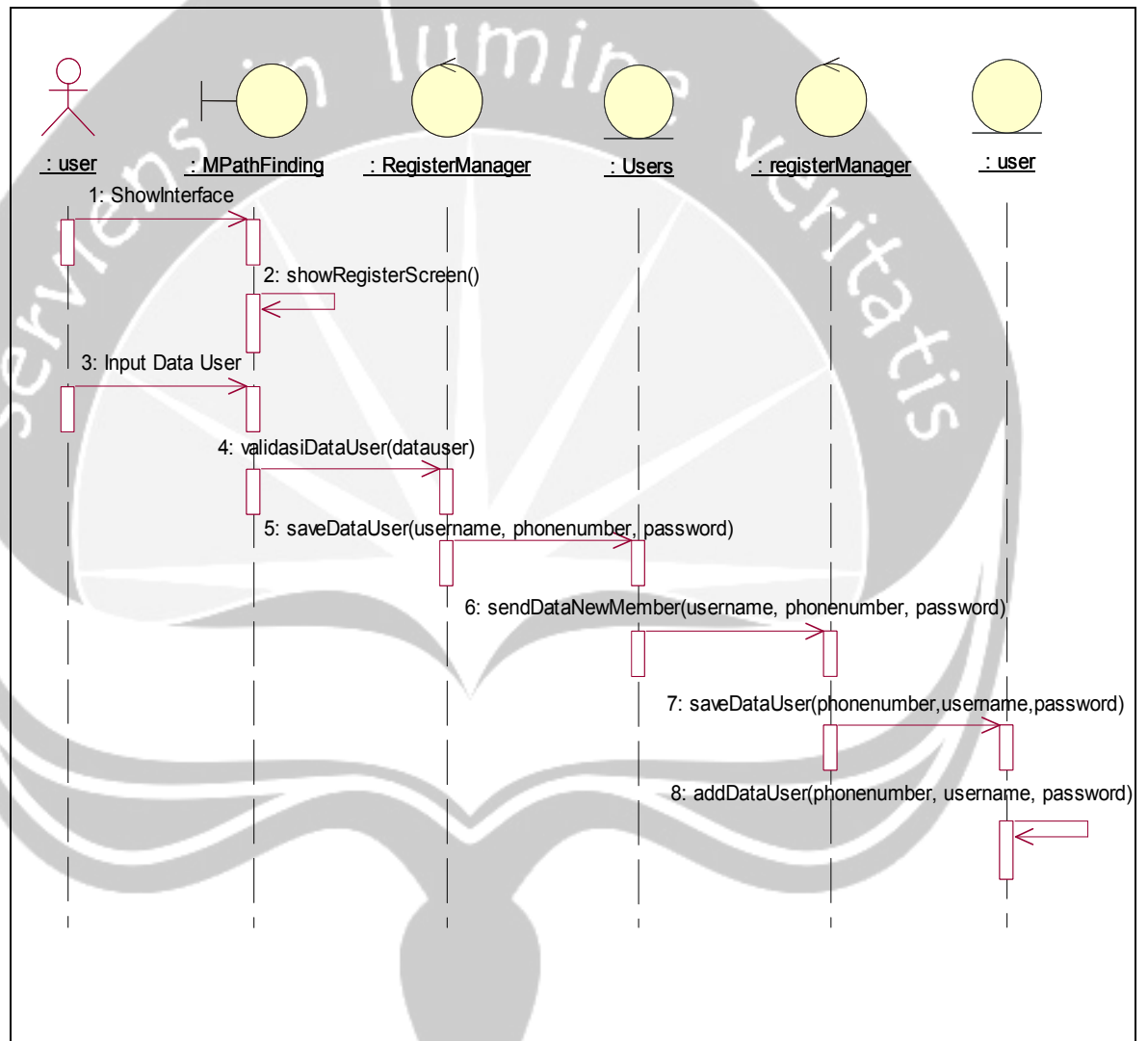
Perancangan Sistem

Perancangan Arsitektur.

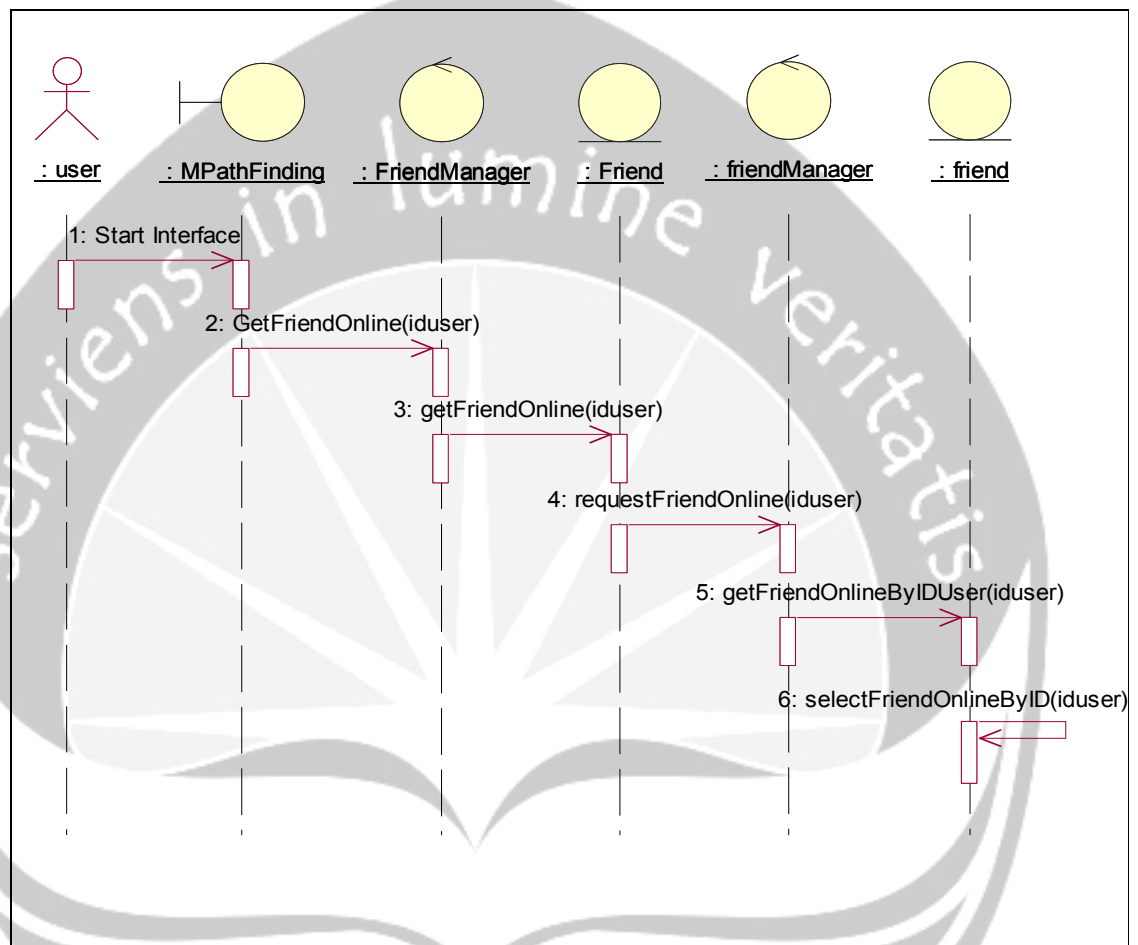
Perancangan Arsitektur Aplikasi MPathFinding



Gambar 2.1 Arsitektur Aplikasi MPathFinding di Mobile Device

Use Case**Registrasi Member**

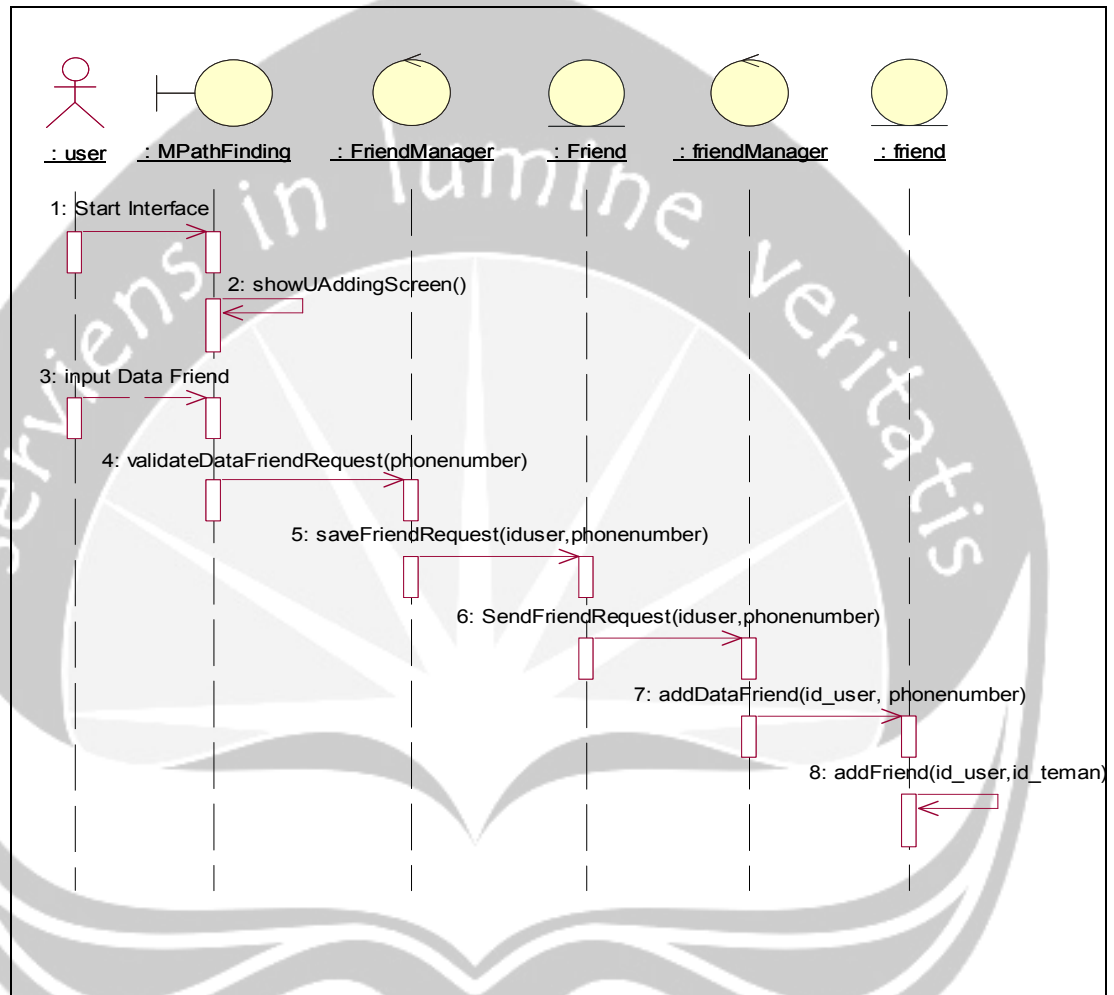
Gambar 2.2 Sequence Diagram: Register Member

Display Friend Online

Gambar 2.3 Sequence Diagram: Display Friend Online

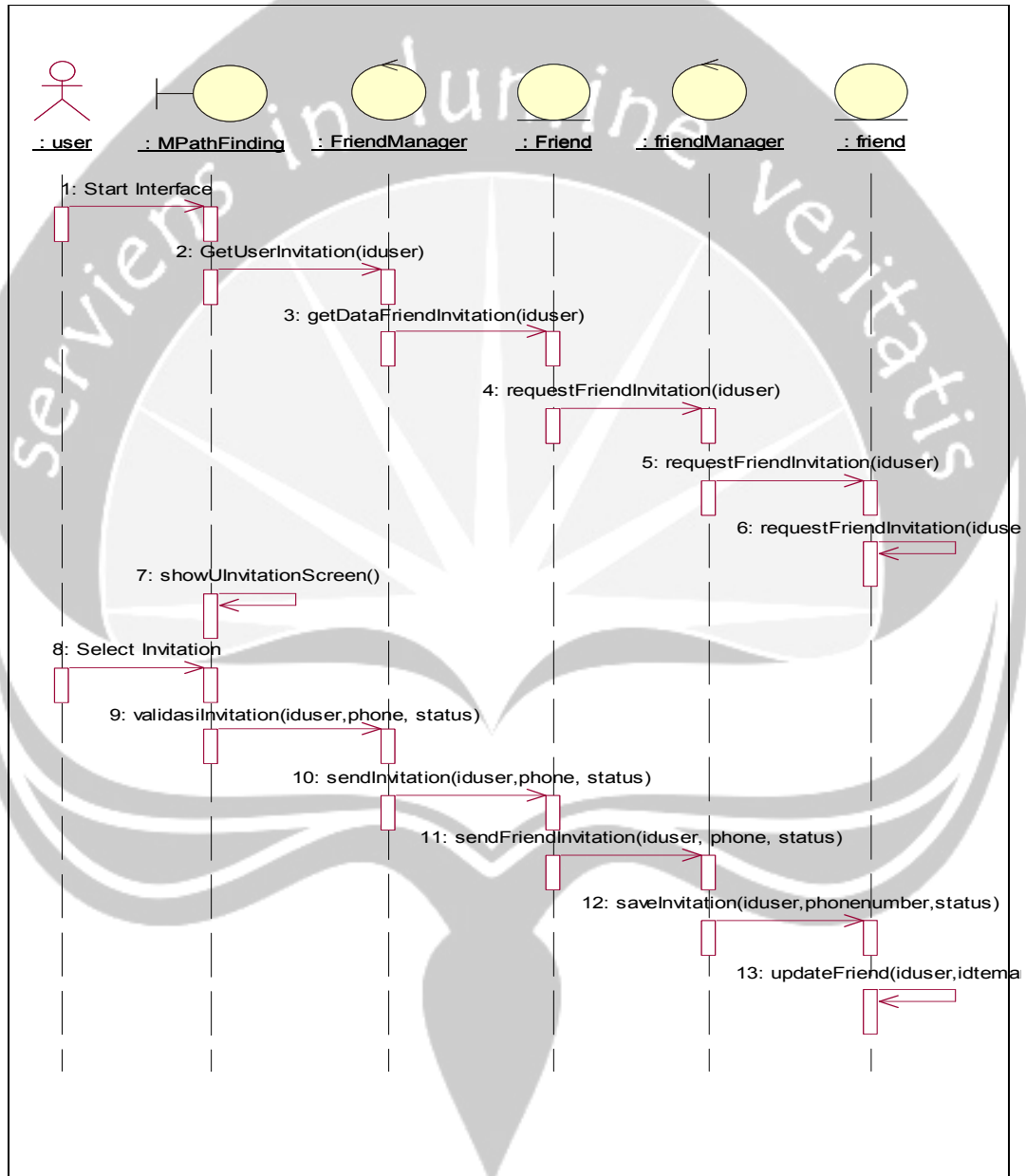
Manage Friend

Adding Friend



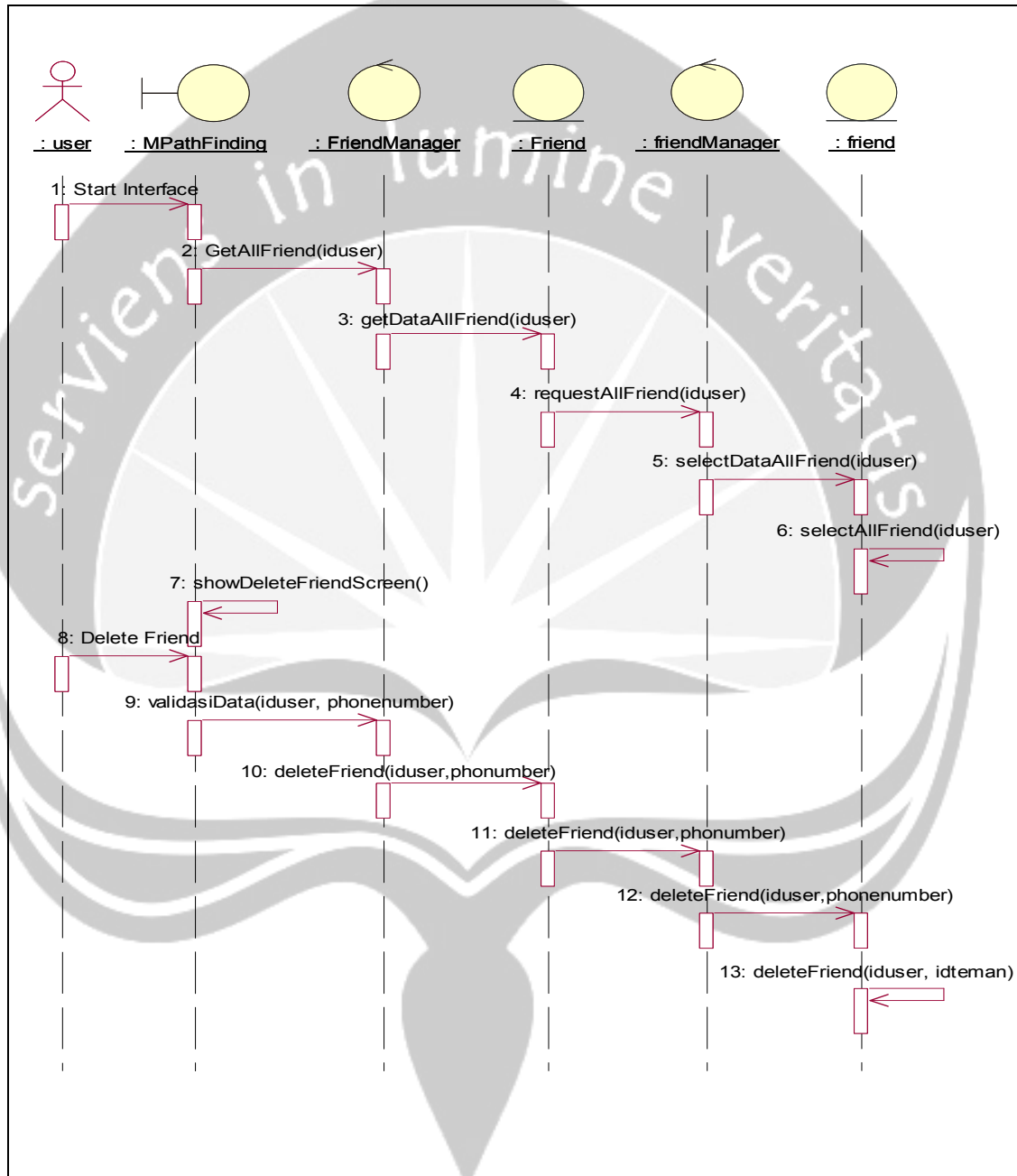
Gambar 2.4 Sequence Diagram: Adding Friend

Request Friend



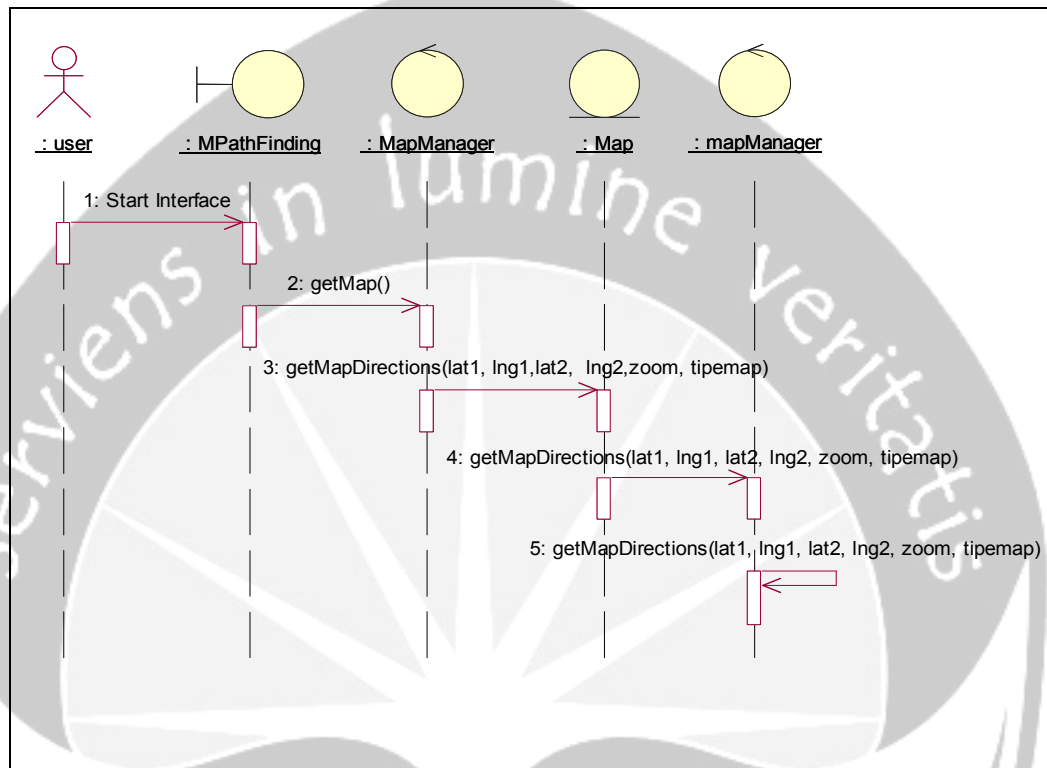
Gambar 2.5 Sequence Diagram: Request Friend

Delete Friend



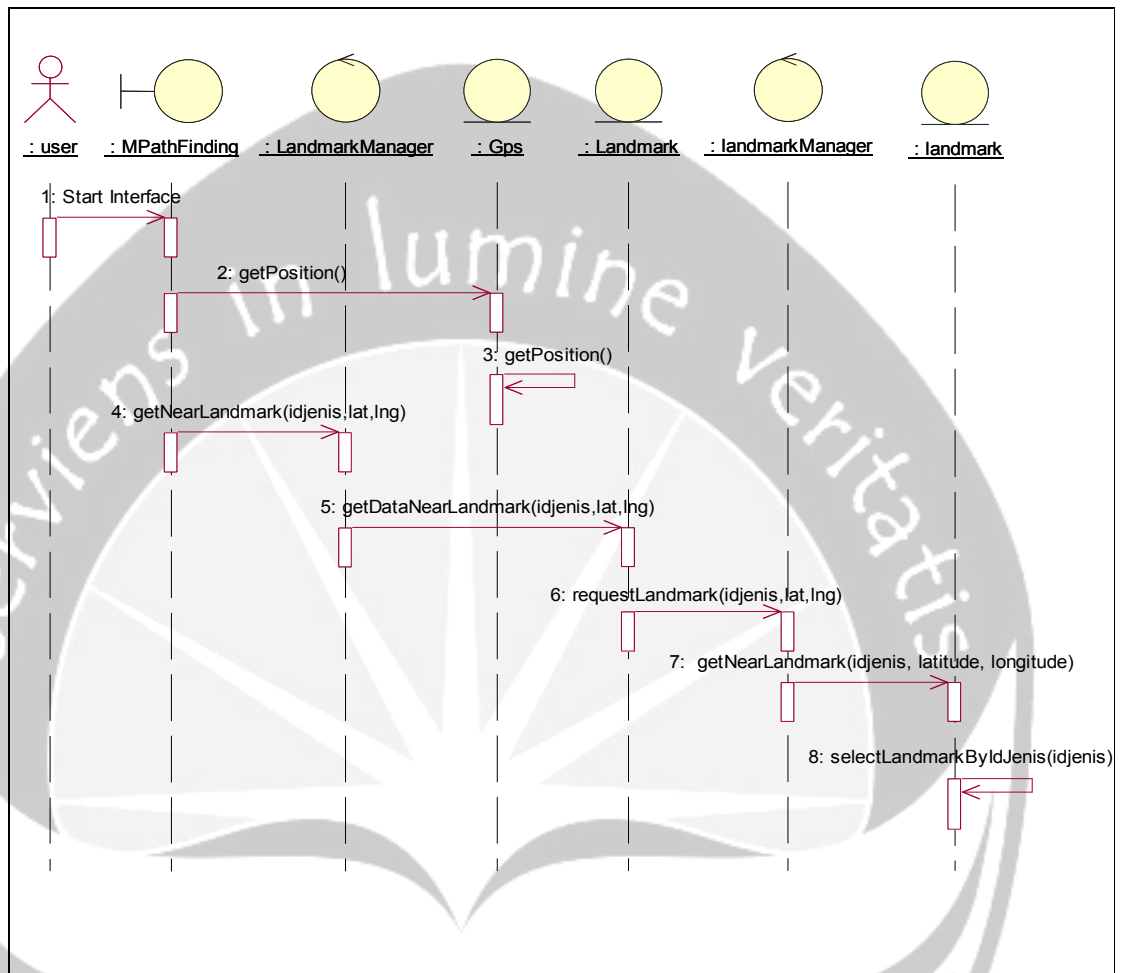
Gambar 2.6 Sequence Diagram: Delete Friend

View Map

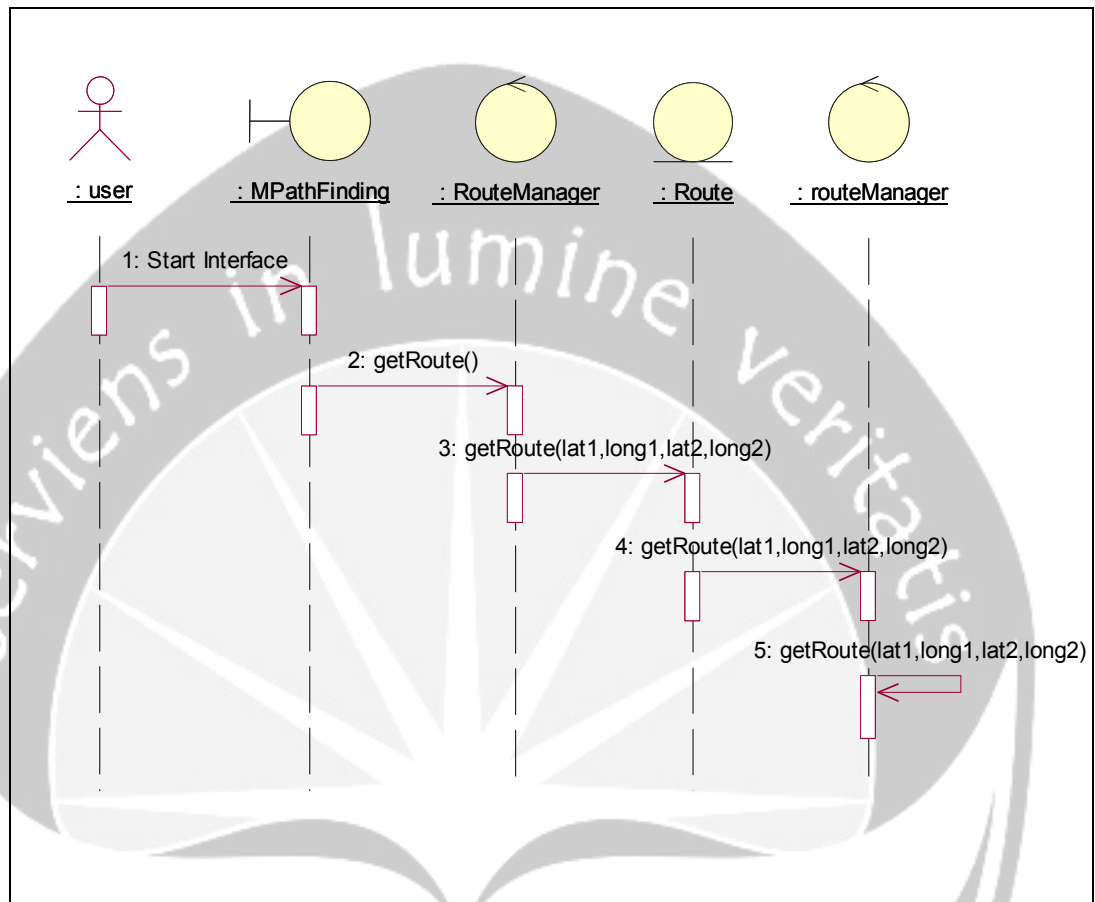


Gambar 2.7 Sequence Diagram Mobile : View Map

Search Near Location

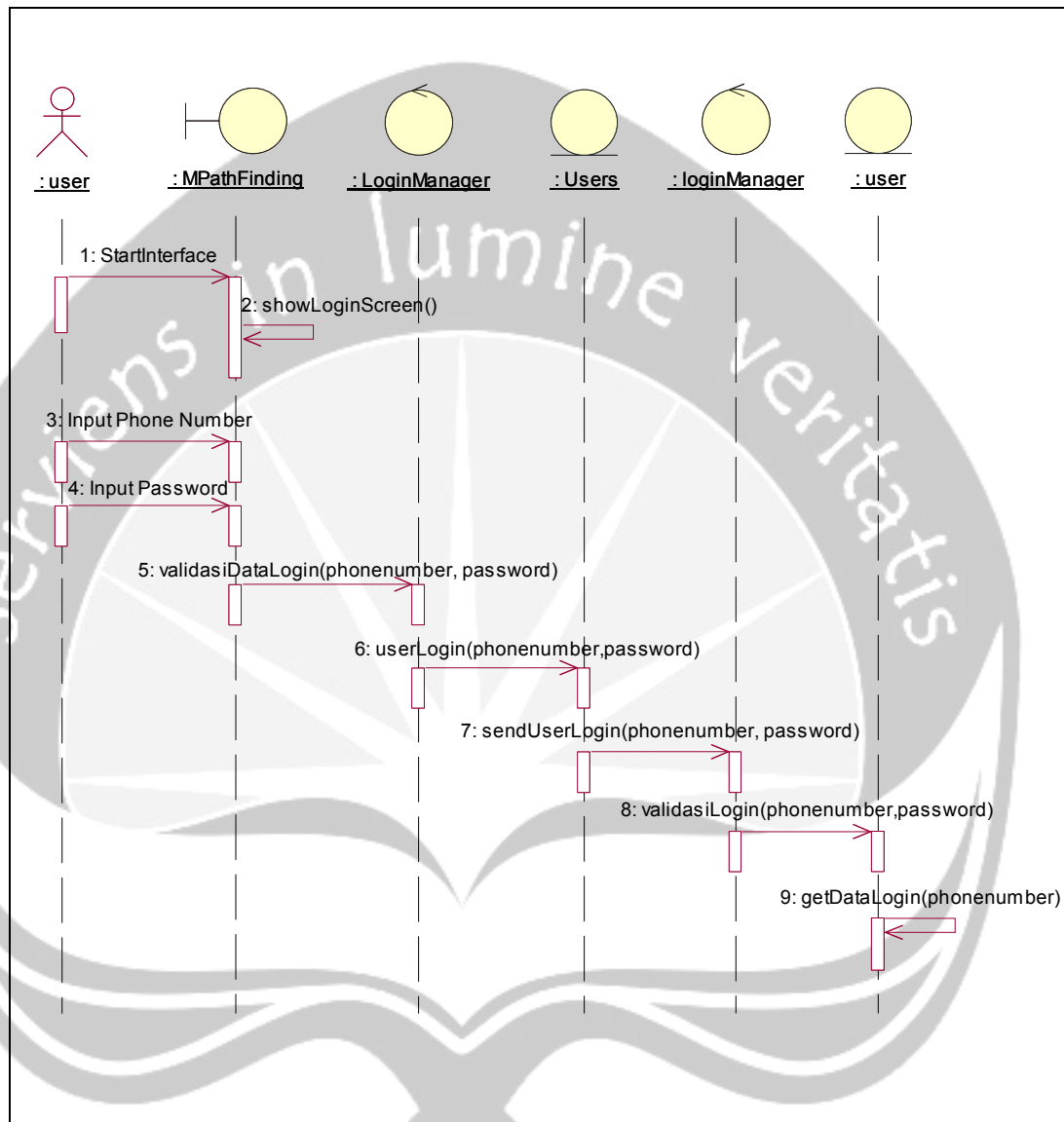


Gambar 2.8 Sequence Diagram : Search Near Location

View Route

Gambar 2.9 Sequence Diagram : View Route

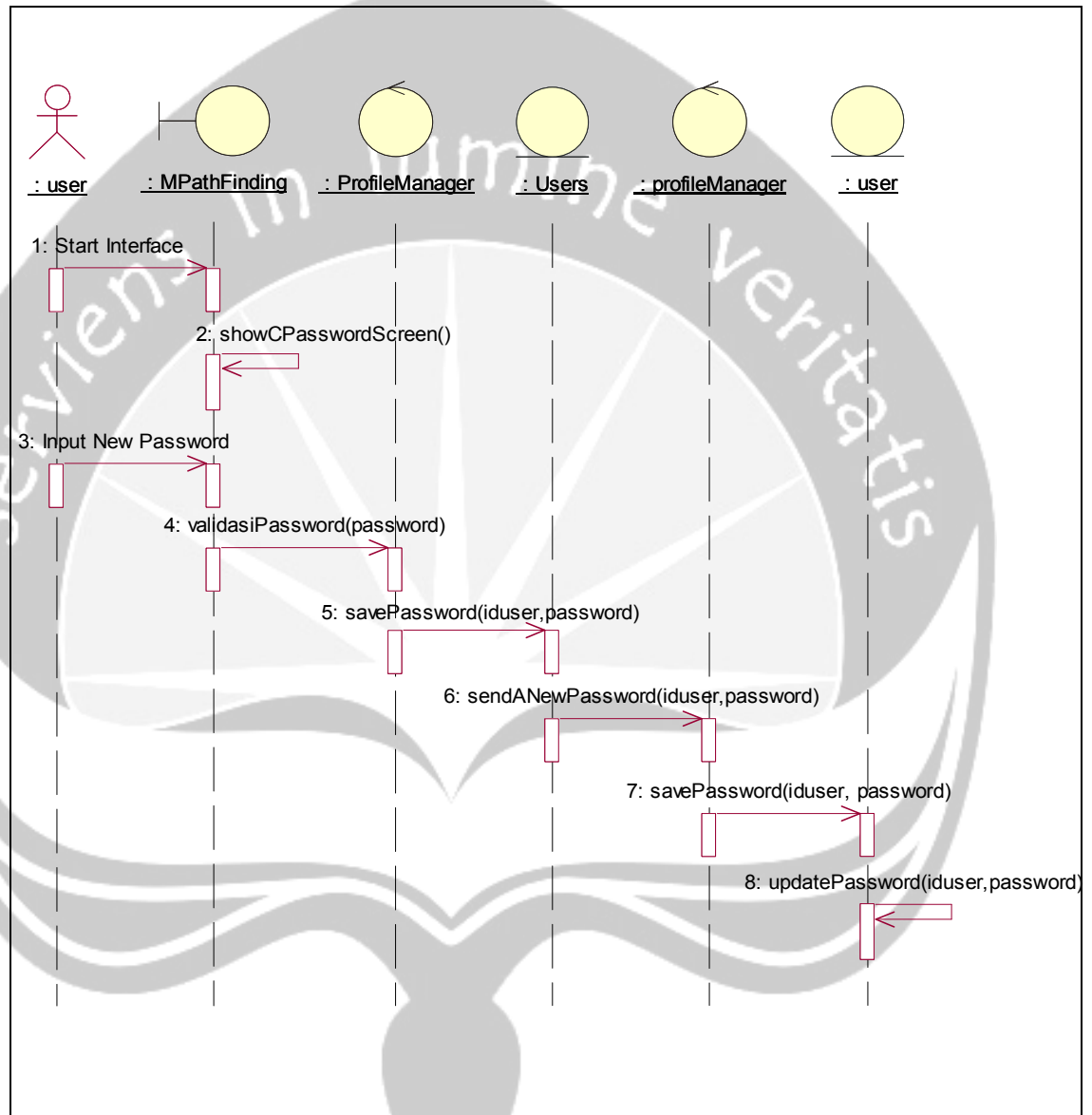
Login



Gambar 2.10 Sequence Diagram : Login

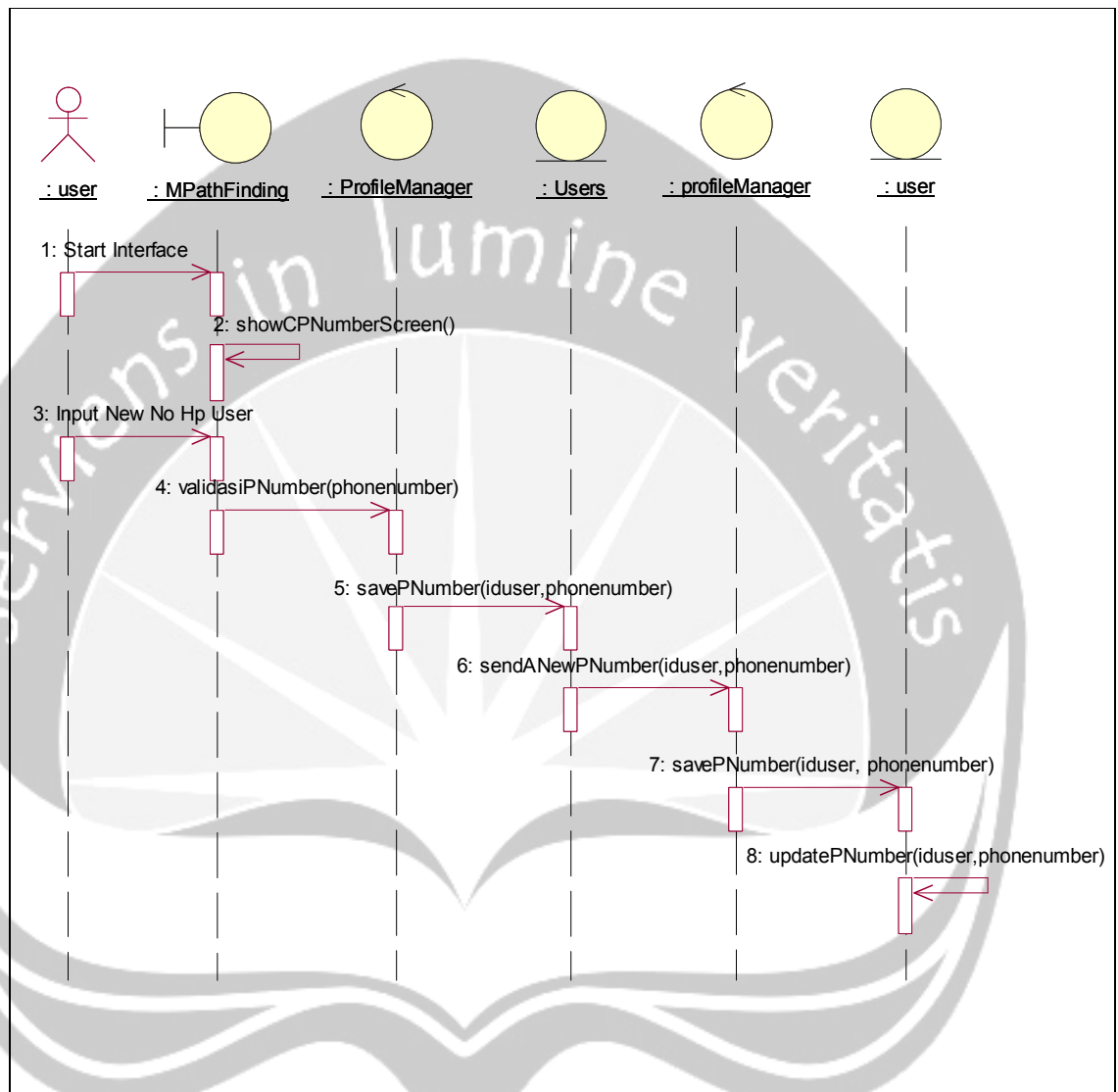
Manage Profile

Change Password



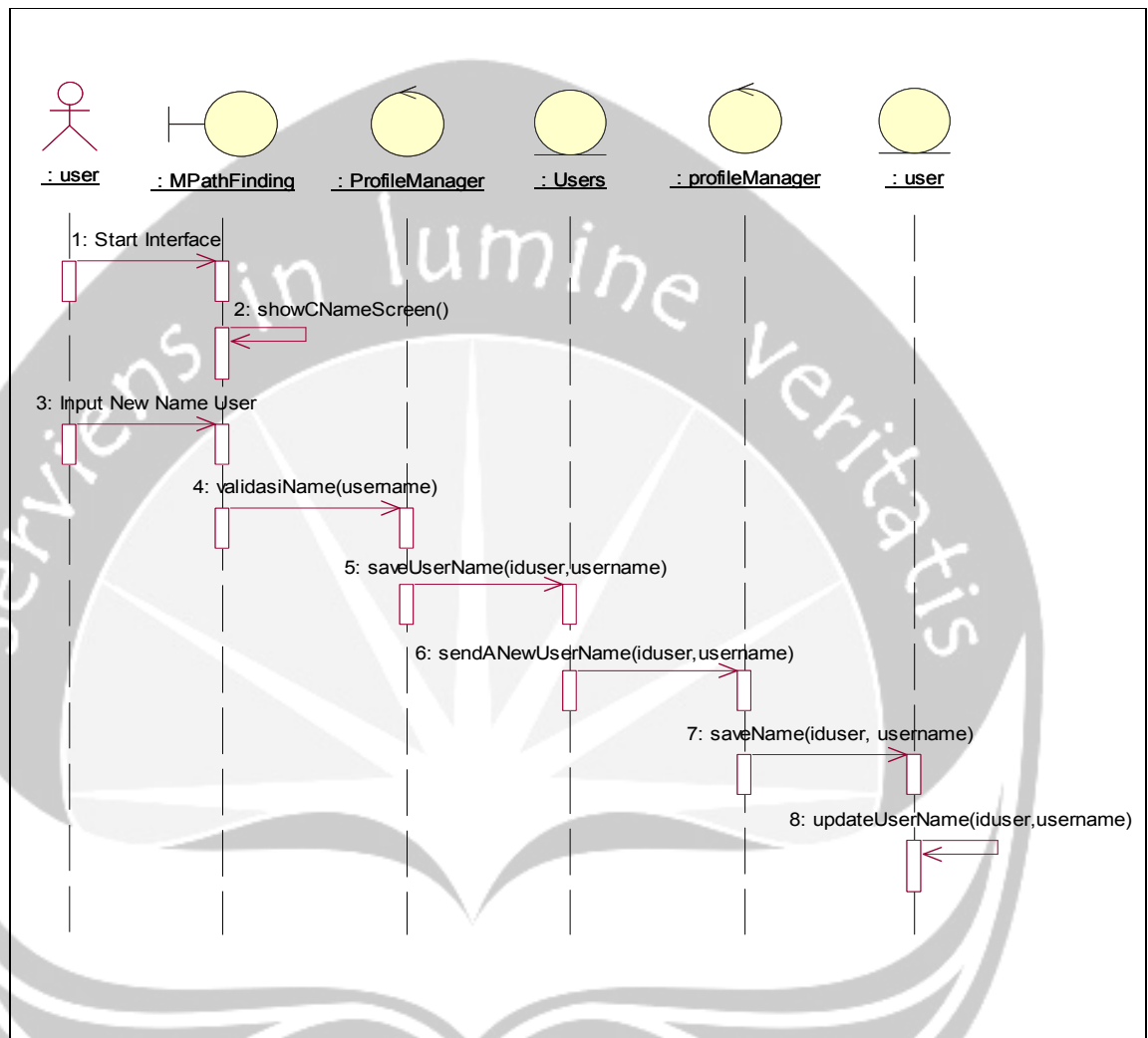
Gambar 2.11 Sequence Diagram : Change Password

Change Phone Number



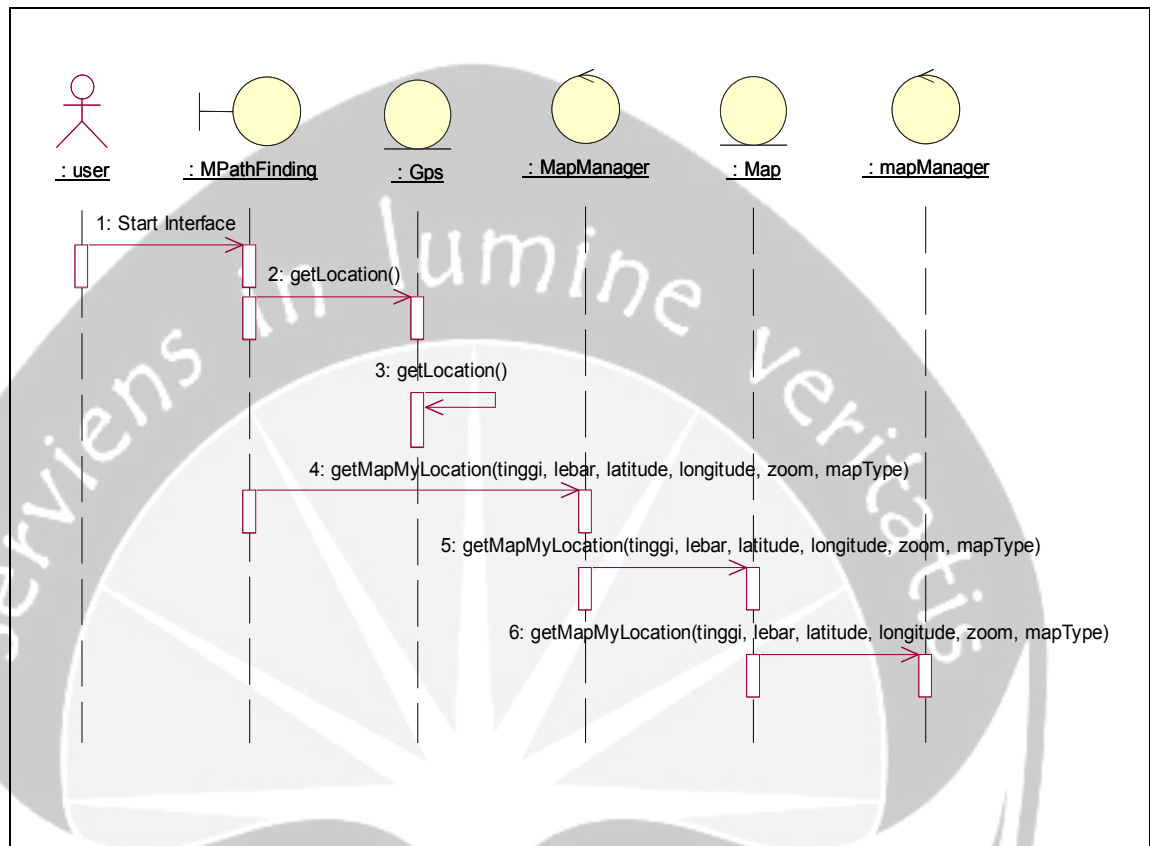
Gambar 2.12 Sequence Diagram : Change Phone Number

Change User Name

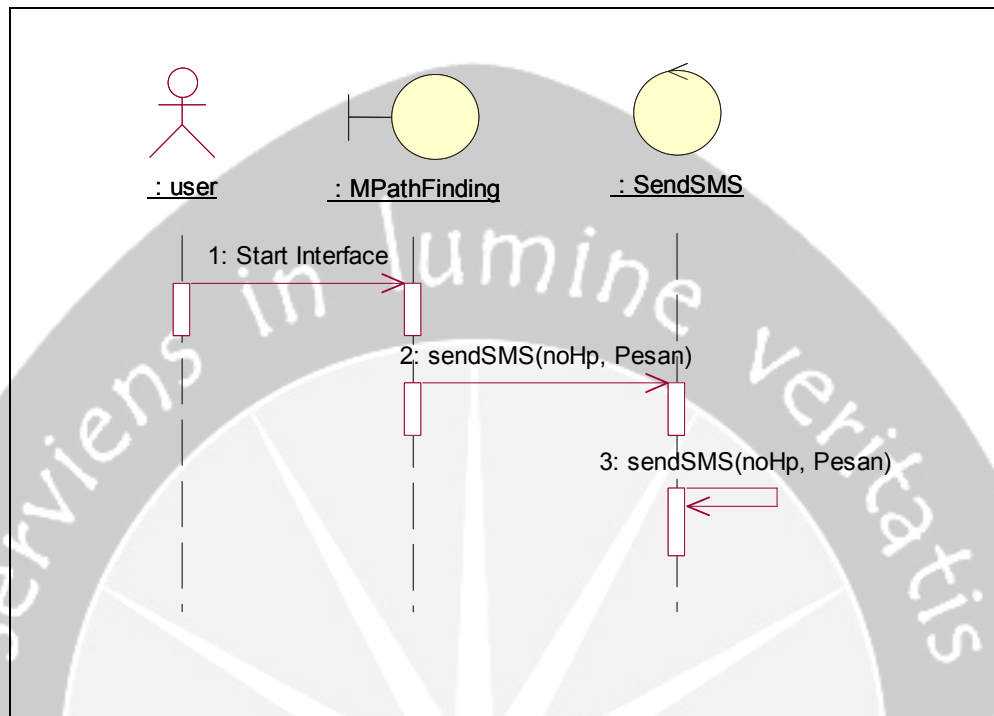


Gambar 2.13 Sequence Diagram : Change Nama Number

Show My Location



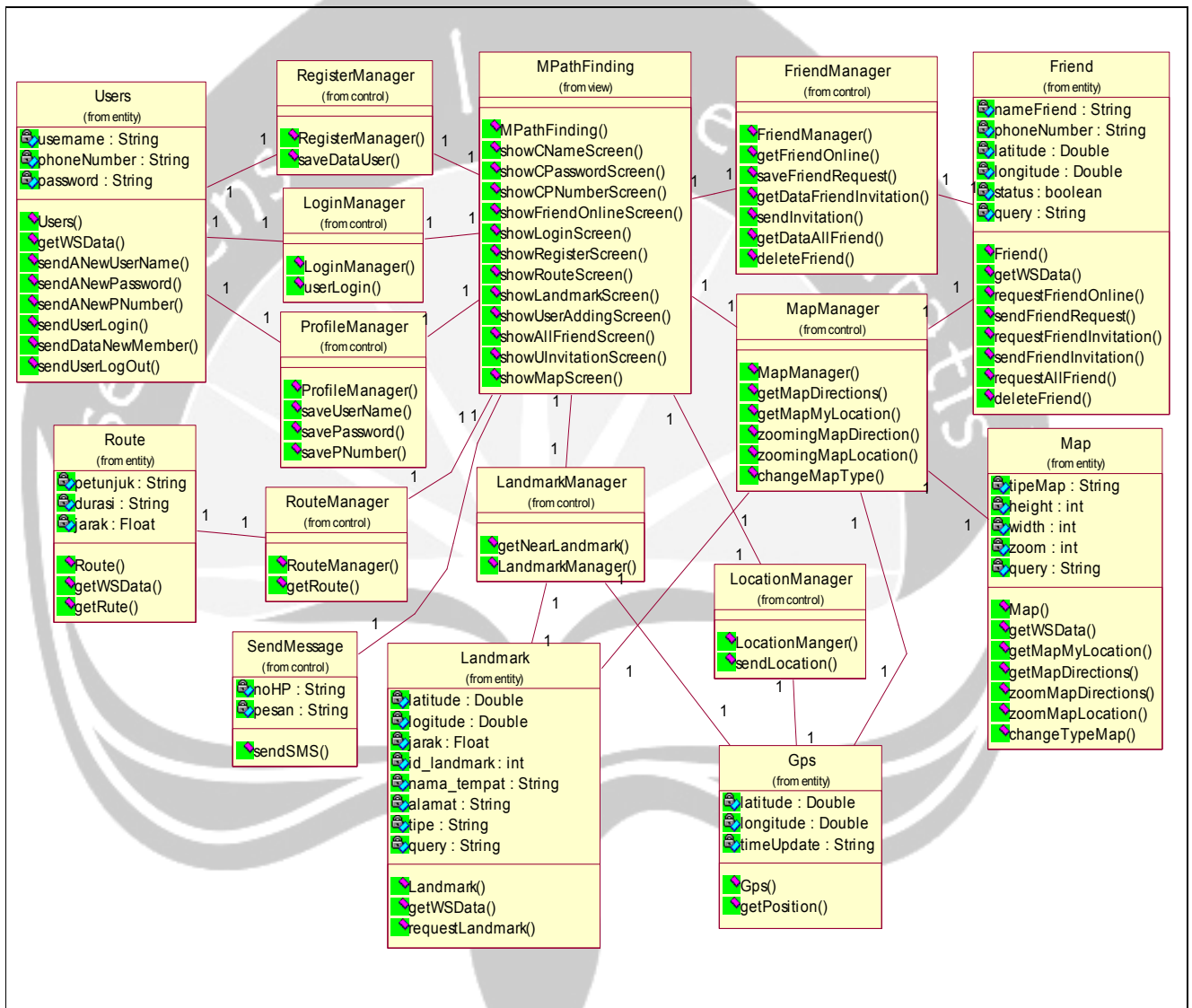
Gambar 2.10 Sequence Diagram : Show My Location

Send SMS

Gambar 2.10 Sequence Diagram : Show My Location

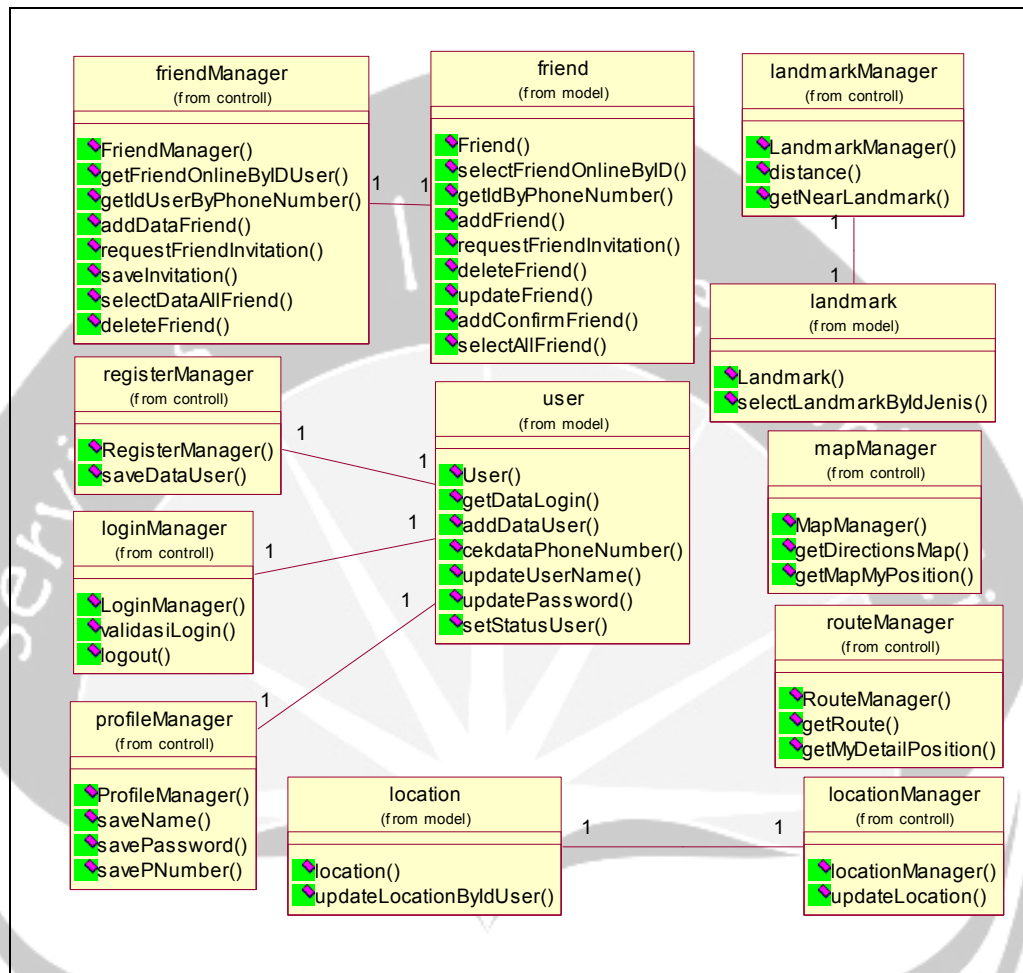
Diagram Kelas

Diagram Kelas Pada Perangkat Mobile



Gambar 2.26 Diagram Kelas Perangkat Mobile

Diagram Kelas Pada Web Service



Gambar 2.26 Diagram Kelas Webservice

Deskripsi Kelas

Deskripsi Kelas Pada Perangkat Mobile

Specific Design Class MPathFinding

MPathFinding	<<view>>
<pre> +MPathFinding() Default konstruktor, digunakan untuk inisialisasi semua attribute dari kelas ini. +showCNameScreen() Operasi ini digunakan sebagai interface untuk pengubahan nama dari user. +showCPasswordScreen() Operasi ini digunakan sebagai interface untuk pengubahan password user. +showCPNumberScreen() Operasi ini digunakan sebagai interface untuk pengubahan No. Handphone user. +showFriendOnlineScreen() Operasi ini digunakan sebagai interface untuk menampilkan user lain yang sedang online. +showLoginScreen() Operasi ini digunakan sebagai interface untuk user login ke system. +showRegisterScreen() Operasi ini digunakan sebagai interface untuk register member. +showRouteScreen() Operasi ini digunakan sebagai interface untuk rute perjalanan menuju ke loksai. +showLandmarkScreen() </pre>	

Operasi ini digunakan sebagai interface untuk menampilkan landmarks yang terdekat.

+showUserAddingScreen()

Operasi ini digunakan sebagai interfae untuk penambahan user lain(teman).

+showUserInvitationScreen()

Operasi ini digunakan untuk menampilkan user yang meng-invite sebagai teman.

+showMapScreen()

Operasi ini digunakan untuk menampilkan peta.

Specific Design Class RegisterManager

RegisterManager	<<control>>
+RegisterManager() Default konstruktor, digunakan untuk inisialisasi semua attribute dari kelas ini. +saveDataUser(username, phonenumber, password) Operasi ini digunakan untuk menyimpan data user yang akan dikirim ke webservice.	

Specific Design Class LoginManager

LoginManager	<<control>>
+LoginManager Default konstruktor, digunakan untuk inisialisasi semua attribute dari kelas ini. +userLogin(username, password, invisible) Operasi ini digunakan untuk user login ke system.	

```
+userLogout()
```

Operasi ini digunakan untuk user keluar dari sistem.

Specific Design Class ProfileManager

ProfileManager	<<control>>
<pre>+ProfileManager() Default konstruktor, digunakan untuk inisialisasi semua attribute dari kelas ini. +saveUserName(iduser, username) Operasi ini digunakan untuk menyimpan perubahan nama yang diubah oleh user. +savePassword(iduser, password) Operasi ini digunakan untuk menyimpan perubahan password yang diubah oleh user. +savePNumber(iduser, phonenumber) Operasi ini digunakan untuk menyimpan perubahan No. Handpone yang diubah oleh user.</pre>	

Specific Design Class RouteManager

RouteManager	<<control>>
<pre>+RouteManager() Default konstruktor, digunakan untuk inisialisasi semua attribute dari kelas ini. +getRoute(latitude1,longitude1,latitude2,longitude2) Operasi ini digunakan untuk mendapatkan data rute.</pre>	

```
+getMyPosition(latitude, longitude)
```

Operasi ini digunakan untuk mendapatkan data posisi *device* berada.

Specific Design Class LandmarkManager

LandmarkManager	<<control>>
<pre>+LandmarkManager()</pre> Default konstruktor, digunakan untuk inisialisasi semua attribute dari kelas ini. <pre>+getNearLandmark(idjenislandmark,latitudeuser,longitudeuser)</pre> Operasi ini digunakan untuk mendapatkan landmark yang dekat dengan user sesuai dengan jenis landmarknya.	

Specific Design Class MapManager

MapManager	<<control>>
<pre>+MapManager()</pre> Default konstruktor, digunakan untuk inisialisasi semua attribute dari kelas ini. <pre>+getMapDirections(latitude1,longitude1,latitude2,longitude2, zoom,tipemap)</pre> Operasi ini digunakan untuk mendapatkan peta <i>directions</i> dari google Map static. <pre>+ getMapMyLocation(tinggi, lebar, latitude, longitude, int zoom, String tipemap)</pre> Operasi ini digunakan untuk mendapatkan peta lokasi <i>device</i> dari google Map static.	

```
+zoomingMapDirection(zoom)
```

Operasi ini digunakan untuk merubah skala peta *directions*.

```
+ zoomingMapLocation(zoom)
```

Operasi ini digunakan untuk merubah skala peta *device*.

```
+ changeMapType (maptype)
```

Operasi ini digunakan untuk merubah tipe peta.

Specific Design Class FriendManager

FriendManager	<<control>>
<pre>+FriendManager</pre> Default konstruktor, digunakan untuk inisialisasi semua attribute dari kelas ini. <pre>+saveFriendRequest(iduser, phonenumber):</pre> Operasi ini digunakan untuk menyimpan data user lain guna mengundang menjadi teman. <pre>+deleteFriend(iduser, phonenumberfriend):</pre> Operasi ini digunakan untuk menghapus teman. <pre>+sendInvitation(iduser,phonenumber,sts)</pre> Operasi ini digunakan untuk memberikan persetujuan atau tidak permintaan user lain untuk menjadikan teman. <pre>+getFriendOnline(iduser):</pre> Operasi ini digunakan untuk menampilkan user lain yang sedang Online. <pre>+getDataFriendInvitation(iduser)</pre> Operasi ini digunakan untuk menampilkan user lain yang meminta persetujuan kita sebagai teman. <pre>+ getDataAllFriend(iduser)</pre> Operasi ini digunakan untuk mendapatkan semua teman baik yang online ataupun tidak.	

Specific Design Class SendMessage

SendMessage	<<control>>
-noHP : String Atribut ini digunakan untuk menyimpan no hp tujuan user. -pesan: String Atribut ini digunakan untuk menyimpan pesan yang akan dikirim.	
+SendMessage Default konstruktor, digunakan untuk inisialisasi semua attribute dari kelas ini. +sendSMS(noHP, pesan) Operasi ini digunakan untuk mengirim pesan (SMS) ke teman (user lain yang online).	

Specific Design Class LocationManager

LocationManager	<<control>>
+LocationManager Default konstruktor, digunakan untuk inisialisasi semua attribute dari kelas ini. +sendLocation(iduser, latitude, longitude) Operasi ini digunakan untuk mengirim lokasi user ke web service.	

Specific Design Class Users

Users	<<model>>
-userName:String Atribut ini digunakan untuk menyimpan data user name.	

-phoneNumber:String

Atribut ini digunakan untuk menyimpan data phone number.

-password:String

Atribut ini digunakan untuk menyimpan data password.

-query :String

Atribut ini digunakan untuk menyimpan data query ke web service.

+Users()

Default konstruktor, digunakan untuk inisialisasi semua attribute dari kelas ini.

+getWsData():

Operasi ini digunakan untuk mengirim atau meminta data di web service.

+sendANewName(iduser,username):

Operasi ini digunakan untuk mengirim nama baru dari user ke web service dan menyimpannya di database.

+sendANewPassword(iduser, password):

Operasi ini digunakan untuk mengirim password baru dari user ke webservice dan menyimpannya dalam database.

+sendANewPNumber(iduser, phonenumber)

Operasi ini digunakan untuk mengirim data no handpone baru ke Web Service dan disimpan di database.

+sendDataNewMember(username, phonenumber, password)

Operasi ini digunakan untuk mengirim data member baru ke web service untuk disimpan ke database.

+ sendUserLogin(phonenumber, password, invisible)

Operasi ini digunakan untuk melakukan proses login.

+ sendUserLogout(phoneNumber)

Operasi ini digunakan untuk melakukan proses logout.

Specific Design Class Route

Route	<<model>>
-petunjuk : String Atribut ini digunakan untuk menyimpan data petunjuk dari rute. -Durasi : String Atribut ini digunakan untuk menyimpan data durasi dari rute. -jarak :Float Atribut ini digunakan untuk menyimpan data jarak dari rute.	
+Route Default konstruktor, digunakan untuk inisialisasi semua attribute dari kelas ini. +getWSData() Operasi ini digunakan untuk data dari web service. +getRute(latitude1, longitude1, latitude2, longitude2) Operasi ini digunakan untuk meminta rute ke web service. + getMyPosition(latitude, longitude) Operasi ini digunakan untuk meminta detail posisi ke webservice.	

Specific Design Class Landmark

Landmark	<<model>>
-latitude : Double Atribut ini digunakan untuk menyimpan data koordinat latitude. -longitude : Double	

Atribut ini digunakan untuk menyimpan data koordinat longitude.

-jarak :Float

Atribut ini digunakan untuk menyimpan data jarak.

-id_landmark:int

Atribut ini digunakan untuk menyimpan data id dari landmark

-nama_tempat : String

Atribut ini digunakan untuk menyimpan data nama tempat dari landmark.

-alamat : String

Atribut ini digunakan untuk menyimpan data alamat dari landmark.

-tipe: string

Atribut ini digunakan untuk menyimpan data tipe dari landmark.

-query :String

Atribut ini digunakan untuk mengeset query yang akan dikirimkan ke web service.

+Landmark

Default konstruktor, digunakan untuk inisialisasi semua attribute dari kelas ini.

+getWSData()

Operasi ini digunakan untuk data dari web service.

+requestLandmark(idjenislandmark,latitudeuser,longituduser)

Operasi ini digunakan untuk meminta data-data landmark terdekat dari web service sesuai dengan jenis landmark.

Specific Design Class Gps

Gps	<<model>>
-latitude : Double Atribut ini digunakan untuk menyimpan data koordinat latitude.	

-longitude : Double Atribut ini digunakan untuk menyimpan data koordinat longitude. -timeupdate :time Atribut ini digunakan untuk menyimpan data waktu permintaan posisi dari <i>device</i>.
+Gps() Default konstruktor, digunakan untuk inisialisasi semua attribute dari kelas ini. +getPosition(): Operasi ini digunakan untuk mendapatkan posisi <i>device</i> secara <i>real time</i>.

Specific Design Class Map

Map	<<model>>
-tipeMap : String Atribut ini digunakan untuk menyimpan tipe peta. -height : int Atribut ini digunakan untuk menyimpan tinggi dari peta. -width :int Atribut ini digunakan untuk menyimpan lebar dari peta. -zoom :int Atribut ini digunakan untuk menyimpan tingkat <i>zooming</i> dari peta. -query : String Atribut ini digunakan untuk menyimpan query yang akan dikirim ke web service.	
+Map() Default konstruktor, digunakan untuk inisialisasi semua attribute dari kelas ini. +getWSData():	

Operasi ini digunakan untuk meminta data dari web service.
 + getMapMyLocation(tinggi, lebar, latitude, longitude, zoom, tipemap)
 operasi ini digunakan untuk meminta gambar peta lokasi device dari web service.
 +getMapDirections(latitude1, longitude1, latitude2, longitude2, zoom, tipemap)
 Operasi ini digunakan untuk meminta gambar peta directions dari webservice.
 + zoomMapLocation(zoom)
 Operasi ini digunakan untuk merubah skala peta lokasi device.
 + zoomMapDirection(zoom)
 Operasi ini digunakan untuk merubah skala peta *directions*.
 + changeTypeMap(typemap)
 Operasi ini digunakan untuk mengganti jenis peta.

Specific Design Class Friend

Friend	<<model>>
-namaFriend : String Atribut ini digunakan untuk menyimpan data nama teman. -phoneNumber : String Atribut ini digunakan untuk menyimpan data No. handphone. -latitude :Double Atribut ini digunakan untuk menyimpan koordinat <i>latitude</i>. -longitude :Double Atribut ini digunakan untuk menyimpan data koordinat <i>longitude</i>. -status :Boolean Atribut ini digunakan untuk menyimpan data status dari teman.	

-query : String

Atribut ini digunakan untuk menyimpan query yang akan dikirim ke web service.

+Friend()

Default konstruktor, digunakan untuk inisialisasi semua attribute dari kelas ini.

+getWSData():

Operasi ini digunakan untuk meminta data dari web service.

+requestFriendOnline(iduser)

Operasi ini digunakan untuk meminta ke webservice user lain yang online dengan menggunakan iduser.

+sendFriendRequest(iduser, phoneno)

Operasi ini digunakan untuk meng-invite user lain dengan menggunakan no handphone.

+requestFriendInvitation(iduser)

Operasi ini digunakan untuk meminta ke webservice mengirimkan user lain yang meminta persetujuan untuk mengetahui posisi dari user(device).

+sendFriendInvitation(iduser, phone, sts)

Operasi ini untuk memberikan persetujuan atau tidak kepada user lain yang telah meng-invite.

+requestAllFriend(iduser)

Operasi ini digunakan untuk meminta data seluruh teman baik yang online ataupun tidak.

+deleteFriend(iduser, phone)

Operasi ini digunakan untuk menghapus user lain dari pertemanan.

Deskripsi Kelas Pada Webservice

Specific Design Class FriendManager

friendManager	<<control>>
<pre> +friendManager() Default konstruktor, digunakan untuk inisialisasi semua attribute dari kelas ini. +getFriendOnlineByIdUser(iduser) Operasi ini digunakan untuk mendapatkan teman yang online berdasarkan iduser. +getIdUserByPhoneNumber(phoneNumber) Operasi ini digunakan untuk mendapatkan iduser berdasarkan nomor handphone. +addDataFriend(id_user, phonenumber) Operasi ini digunakan untuk menambahkan teman. +requestFriendInvitation(iduser) Operasi ini digunakan untuk meminta semua undangan dari user lain. +saveInvitation(iduser,phonenumber,status) Operasi ini digunakan untuk menyetujui atau tidak undangan pertemanan dari user lain. +selectDataAllFriend(iduser) Operasi ini digunakan untuk menampilkan semua teman baik yang online ataupun tidak. +deleteFriend(iduser,phonenumber) Operasi ini digunakan untuk menghapus pertemanan dengan user lain. </pre>	

Specific Design Class RegisterManager

registerManager	<<controll>>
<pre>+registerManager() Default konstruktor, digunakan untuk inisialisasi semua attribute dari kelas ini. +saveDataUser(phonenummer,username,password) Operasi ini digunakan untuk melakukan pendaftaran member.</pre>	

Specific Design Class LoginManager

loginManager	<<controll>>
<pre>+loginManager() Default konstruktor, digunakan untuk inisialisasi semua attribute dari kelas ini. +validasiLogin(phonenummer,password,invisible) Operasi ini digunakan untuk memvalidasi data user yang hendak masuk ke sistem. +logout(phonenummer) Operasi ini digunakan untuk user keluar dari sistem.</pre>	

Specific Design Class ProfileManager

profileManager	<<controll>>
<pre>+profileManager() Default konstruktor, digunakan untuk inisialisasi semua attribute dari kelas ini.</pre>	

```
+saveName(iduser, username)
```

Operasi ini digunakan untuk menyimpan perubahan nama user.

```
+savePassword(iduser, password)
```

Operasi ini digunakan untuk menyimpan perubahan password user.

```
+savePNumber(iduser, phonenumber)
```

Operasi ini digunakan untuk menyimpan perubahan nomor handphone user.

Specific Design Class LandmarkManager

LandmarkManager	<<control>>
<pre>+LandmarkManager()</pre> Default konstruktor, digunakan untuk inisialisasi semua attribute dari kelas ini. <pre>+distance(lat1, lon1, lat2, lon2)</pre> Operasi ini digunakan untuk mengukur jarak dua koordinat. <pre>+getNearLandmark(idjenislandmark, latitude, longitude)</pre> Operasi ini digunakan untuk mendapatkan data-data landmark terdekat sesuai dengan jenis landmark.	

Specific Design Class MapManager

MapManager	<<control>>
<pre>+MapManager()</pre> Default konstruktor, digunakan untuk inisialisasi semua attribute dari kelas ini. <pre>+getDirectionsMap(lat1, long1, lat2, long2, zm, mapType)</pre> Operasi ini digunakan untuk menampilkan peta <i>direction</i>. <pre>+getMapMyPosition(height, width, lat, lng, zoom, mapType)</pre> Operasi ini digunakan untuk menampilkan peta lokasi device.	

Specific Design Class RouteManager

RouteManager	<<control>>
<pre>+RouteManager()</pre> Default konstruktor, digunakan untuk inisialisasi semua attribute dari kelas ini. <pre>+ getRoute(lat1,lng1,lat2,lng2)</pre> Operasi ini digunakan untuk menampilkan rute perjalanan yang dapat ditempuh oleh user. <pre>+ getMyDetailPosition(latitudeuser,longitudeuser)</pre> Operasi ini digunakan untuk menampilkan detail posisi dari device.	

Specific Design Class LocationManager

LocationManager	<<control>>
<pre>+ LocationManager()</pre> Default konstruktor, digunakan untuk inisialisasi semua attribute dari kelas ini. <pre>+ updateLocation(id,lat,long)</pre> Operasi ini digunakan untuk mengubah posisi dari device.	

Specific Design Class User

User	<<model>>
<pre>+ User()</pre> Default konstruktor, digunakan untuk inisialisasi semua attribute dari kelas ini. <pre>+ getDataLogin(phonenummer)</pre> Operasi ini digunakan untuk mendapatkan data berdasarkan	

phonenumber dari database.

+addDataUser(phonenumber, username, password)

Operasi ini digunakan untuk menambahkan data user ke database.

+ cekdataPhoneNumber(phonenumber)

Operasi ini digunakan untuk mengecek apakah nomor handphone telah ada dalam database atau tidak.

+ updateUserName(iduser, Username)

Operasi ini digunakan untuk mengubah username dalam database.

+ updatePassword(iduser, password)

Operasi ini digunakan untuk mengubah password dalam database.

+ updatePhoneNumber(iduser, phonenumber)

Operasi ini digunakan untuk mengubah nomor handphone dalam database.

+ setStatusUser(phonenumber, status)

Operasi ini digunakan untuk mengubah status user dalam database.

Specific Design Class Location

Location	<<model>>
+Location() Default konstruktor, digunakan untuk inisialisasi semua attribute dari kelas ini. + updateLocation(id,lat,long) Operasi ini digunakan untuk mengubah posisi dari device dalam database.	

Specific Design Class Landmark

Landmark	<<model>>
<pre> +Landmark() Default konstruktor, digunakan untuk inisialisasi semua attribute dari kelas ini. + selectLandmarkByIdJenis(idjenis) Operasi ini digunakan untuk mendapatkan landmark sesuai dengan idJenisLandmark.</pre>	

Specific Design Class Friend

Friend	<<model>>
<pre> + Friend() Default konstruktor, digunakan untuk inisialisasi semua attribute dari kelas ini. + selectFriendOnlineByID(id_user) Operasi ini digunakan untuk mendapatkan teman yang online berdasarkan user id dalam database. + getIdByPhoneNumber(phonenummer) Operasi ini digunakan untuk mendapatkan id user berdasarkan nomor handphone dalam database. + addFriend(iduser, idteman) Operasi ini digunakan untuk merequest userlain. + requestFriendInvitation(id_user) Operasi ini digunakan untuk mendapatkan data-data user lain yang sudah meng-invite. + deleteFriend(iduser, idteman) Operasi ini digunakan untuk menghapus user lain dari pertemanan. + addConfirmFriend(iduser, idteman, status)</pre>	

Operasi ini digunakan untuk menyetujui undangan dari user lain untuk menjadi teman.

```
+ selectAllFriend(iduser)
```

Operasi ini digunakan untuk mendapatkan data seluruh teman baik yang online maupun tidak.

Perancangan Data

Dekomposisi Data

Deskripsi Entitas Data user

Nama	Tipe	Panjang	Keterangan
<u>Id user</u>	Integer	11	Id dari user, Primar key
Phonenumber	Numeric	20	No handpone dari user
Namauser	Varchar	20	Nama user
Password	Varchar	20	Password untuk user
Status	Integer	11	Status online user

Deskripsi Entitas Data Friend

Nama	Tipe	Panjang	Keterangan
<u>Id user</u>	Integer	11	Id dari user, Foreign key, Primary key gabungan(id_teman)
<u>Id teman</u>	Integer	11	Id dari teman, Foreign key, Primary key gabungan(id_user)
Status	Integer	11	Status pertemanan user

Deskripsi Entitas Data Locations

Elemen Data	Tipe	Panjang	Keterangan
<u>Id locations</u>	Integer	11	id untuk locations user, Primary Key
<u>Id user</u>	Integer	11	Id dari user, Foreign Key
Latitude	Double		Koordinat latitude lokasi
Longitude	Double		Koordinat longitude dari lokasi
lastUpdate	Time		Waktu keberadaan device
Lokasi	Varchar	100	Untuk nama tempat lokasi user.

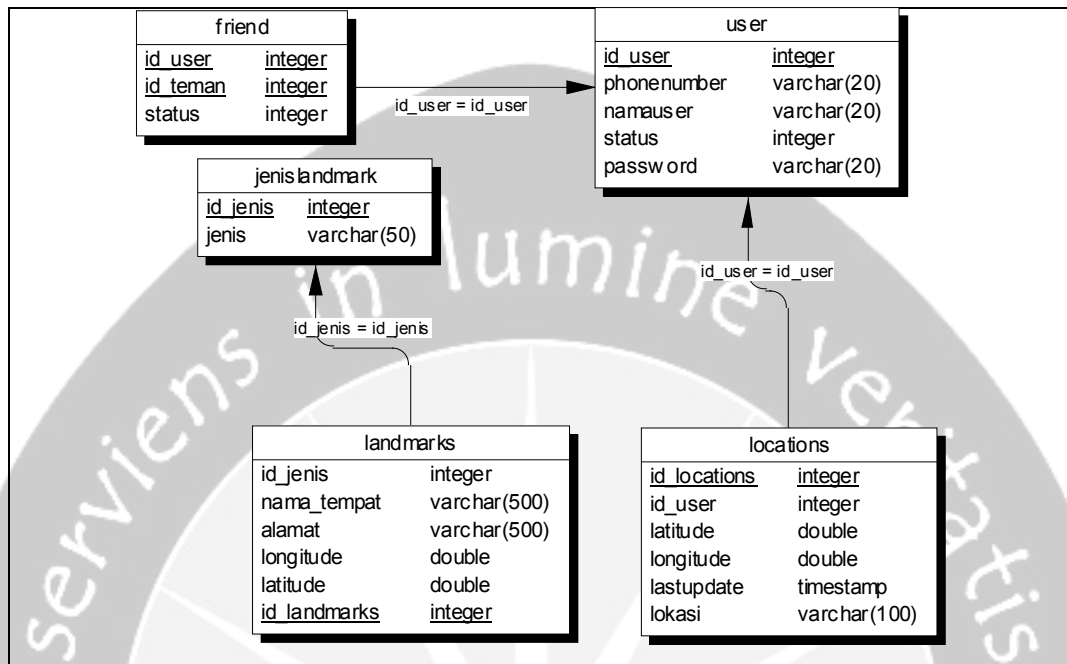
Deskripsi Entitas Data JenisLandmark

Elemen Data	tipe	Panjang	Keterangan
<u>Id jenisLandmark</u>	Integer	11	Id untuk jenis landmark, Primary key
Jenis	Varchar	50	jenis dari landmarks

Deskripsi Entitas Data Landmarks

Elemen Data	Tipe	Panjang	Keterangan
<u>Id landmarks</u>	Integer	11	Id dari landmark, Primary Key
Nama_tempat	Varchar	500	Nama dari landmark
Alamat	Varchar	500	Alamat dari landmark
Latitude	Double		Koordinat latitude landmark
Longitude	Double		Koordinat longitude dari landmark

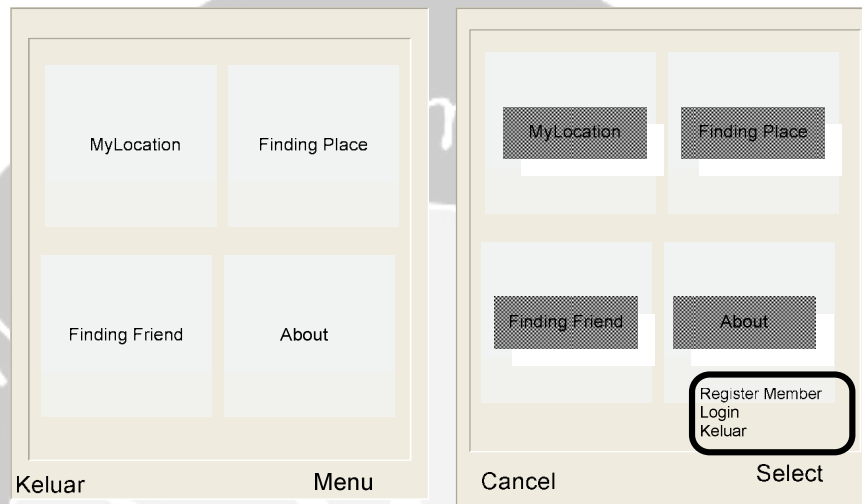
Physical Data Model



Gambar 3 Physical Data Model

Perancangan Antarmuka

Antarmuka Main Screen

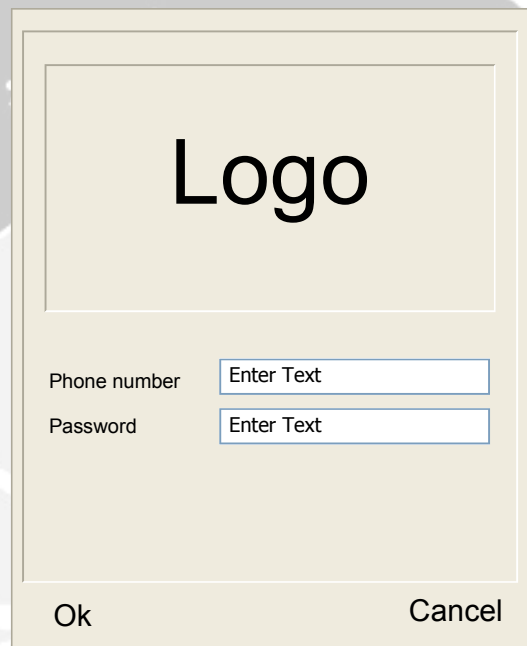


Gambar 4.1 Antarmuka MainScreen

Antarmuka MainScreen adalah antarmuka yang pertamakali oleh user setiap user masuk ke aplikasi MPathFinding. Pada Antarmuka ini memiliki 4 button diantaranya Button MyLocation adalah button yang digunakan untuk menampilkan peta lokasi *device* secara realtime. Button Findding Place digunakan untuk melakukan pencarian landmark-landmark terdekat sesuai dengan tipenya. Button Findding Friend digunakan untuk menampilkan user lain yang sedang online. Sedangkan Button About digunakan untuk menampilkan pembuat aplikasi ini. Disamping mempunyai 4 button pada antarmuka ini juga memiliki 2 command yaitu command keluar digunakan untuk keluar dari aplikasi. Command

Menu memiliki submenu Register member digunakan untuk pendaftaran member. Submenu Login digunakan untuk melakukan proses login ke sistem.

Antarmuka Login



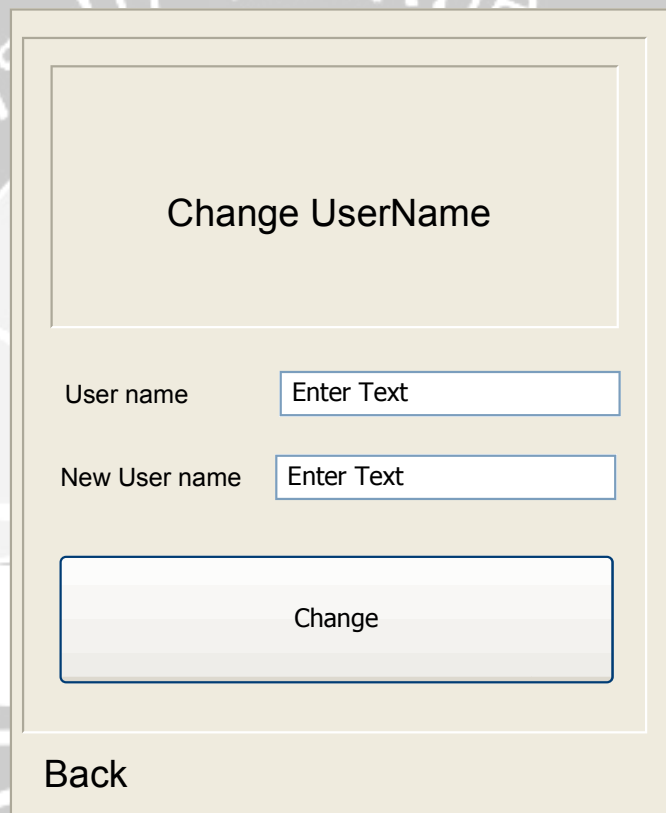
The image shows a login dialog box with a light beige background. At the top center, the word "Logo" is displayed in a large, bold, black font. Below this, there are two input fields. The first is labeled "Phone number" and the second is labeled "Password". Both labels are in a standard black font. To the right of each label is a text box with a blue border and the placeholder text "Enter Text". At the bottom of the dialog box, there are two buttons: "Ok" on the left and "Cancel" on the right, both in a bold black font.

Gambar 4.2 Antarmuka Login

Antarmuka login digunakan untuk melakukan proses login ke dalam sistem. Untuk mendapat akses masuk ke dalam sistem, user harus menginputkan *phone number* dan *password* dengan benar pada *textbox* yang telah disediakan. Pada saat *command* Ok ditekan, sistem akan mengecek *phone number* dan *password* yang telah diinputkan user dengan data *phone number* dan *password* yang ada di basis data. Jika data *phone number* dan *password* benar atau cocok maka user akan masuk ke dalam sistem, sebaliknya jika *phone number* dan *password*

salah atau tidak cocok maka akan diberikan pesan peringatan.

Antarmuka ChangeName



Change UserName

User name

New User name

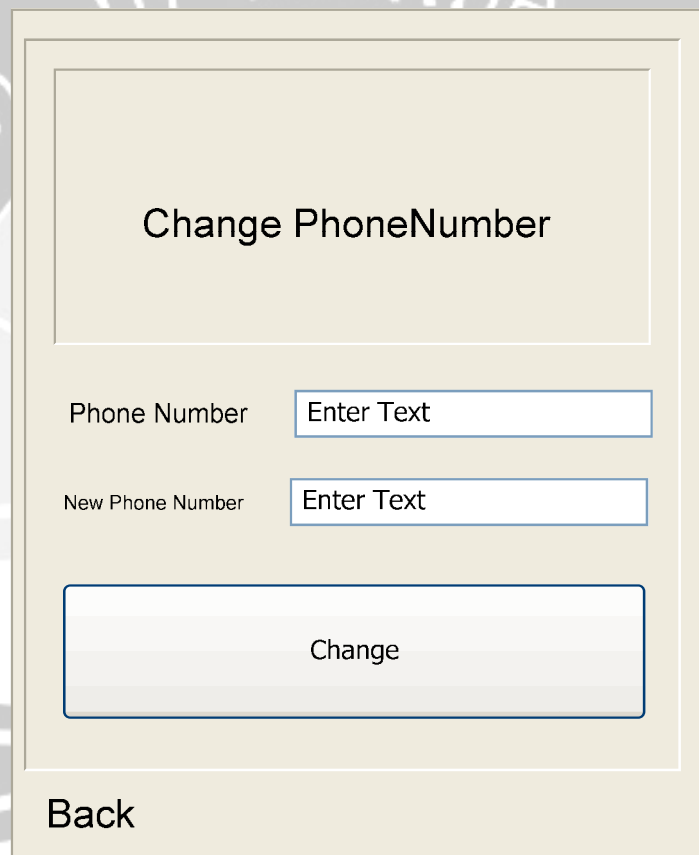
Back

Gambar 4.3 Antarmuka ChangeName

Antarmuka ini digunakan untuk mengubah *username* user. *Textbox username* akan terisi secara otomatis ketika user login, sehingga user hanya perlu memasukkan username yang baru pada *textbox* yang telah disediakan kemudian klik tombol *change*. Sistem akan memberikan

konfirmasi tentang perubahan *username* kemudian klik ok. Maka sistem akan mengubah *username user*. Untuk kembali ke menu sebelumnya tekan *command back*.

Antarmuka ChangePhonenumber



Change PhoneNumber

Phone Number

New Phone Number

Change

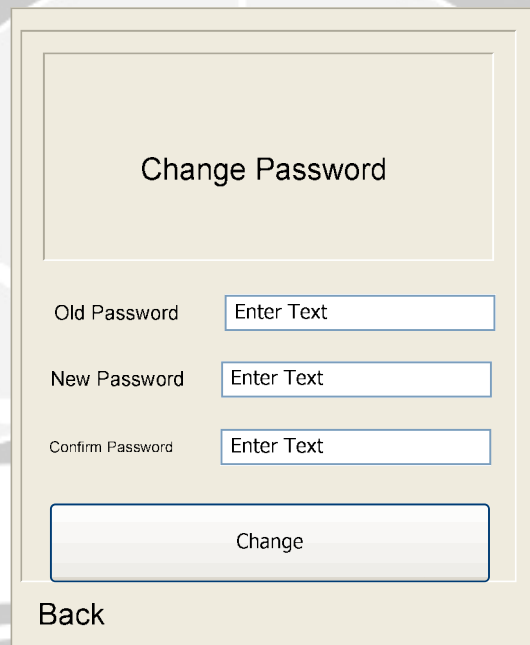
Back

Gambar 4.4 Antarmuka Phone Number

Antarmuka ini digunakan untuk mengubah *phone number user*. *Textbox phone number* akan terisi secara otomatis ketika *user login*, sehingga *user* hanya perlu memasukkan *phone number* yang baru pada *textbox* yang

telah disediakan kemudian klik tombol *change*. Sistem akan memberikan konfirmasi tentang perubahan *phone number* kemudian klik *ok*. Maka sistem akan mengubah *phone number user*. Untuk kembali ke menu sebelumnya tekan *command back*.

Antarmuka ChangePassword



Change Password

Old Password

New Password

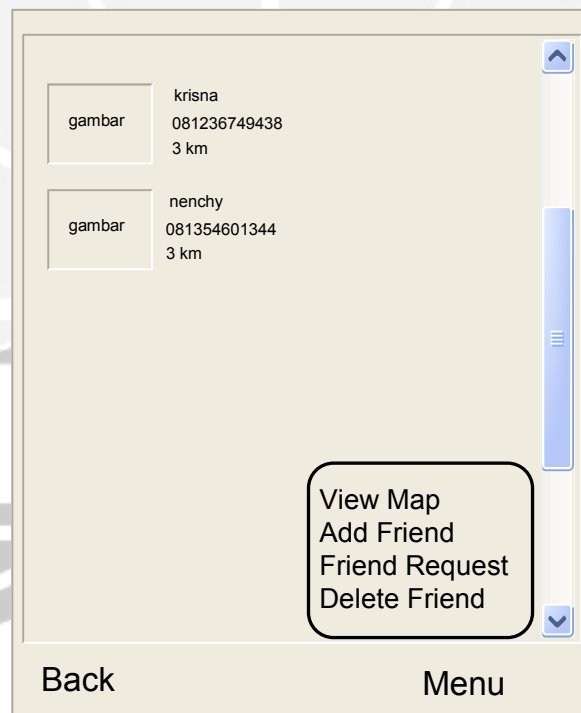
Confirm Password

Gambar 4.5 Antarmuka ChangeName

Antarmuka ini digunakan untuk mengubah *password user*. *User* akan menginputkan *password* yang lama, *password* baru dan *confirm password* yang baru kemudian klik tombol *change*. Jika *password* baru dan *confirm password* cocok atau sama dan *password* yang lama cocok dengan *password* yang ada di database maka sistem akan

menyimpan perubahan data di dalam basis data, Sedangkan jika *password* baru dan *password* konfirmasi tidak cocok ataupun *password* lama tidak cocok dengan yang di basis data maka sistem akan memberikan pesan peringatan . Untuk kembali ke menu sebelumnya tekan *command back*.

Antarmuka DisplayFriendOnline

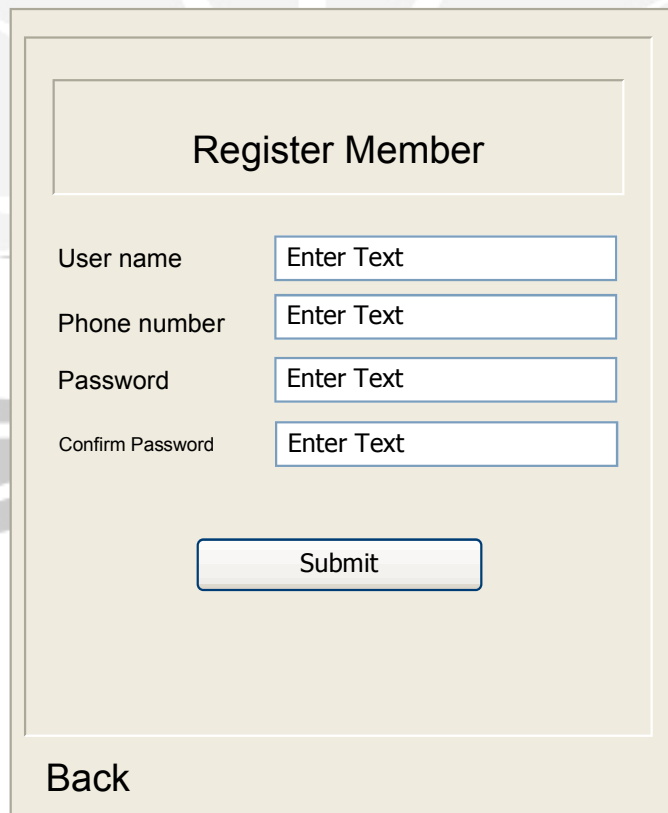


Gambar 4.6 Antarmuka DisplayFriendOnline

Antarmuka ini digunakan untuk menampilkan *user* lain yang sedang *online*. *Command back* digunakan untuk kembali ke menu yang sebelumnya. Terdapat menu yang

terdiri *View Map*, *Add Friend*, *Friend Request* dan *Delete Friend*. *Command View Map* digunakan untuk menampilkan data peta. *Command Add Friend* digunakan untuk menambahkan user lain sebagai teman. *Command Friend Request* digunakan untuk menampilkan permintaan menjadi teman user lain. *Command Delete Friend* digunakan untuk menampilkan semua teman baik yang online ataupun tidak, sehingga proses untuk penghapusan teman bisa dilakukan.

Antarmuka RegisterManager



The image shows a web form titled "Register Member". It contains four text input fields labeled "User name", "Phone number", "Password", and "Confirm Password", each with a placeholder text "Enter Text". Below the fields is a "Submit" button. At the bottom left of the form is a "Back" link. The form is overlaid on a background featuring a large, faint watermark of a university seal with the Latin motto "serviens in lumine veritatis".

Register Member	
User name	Enter Text
Phone number	Enter Text
Password	Enter Text
Confirm Password	Enter Text
Submit	
Back	

Gambar 4.7 Antarmuka RegisterManager

Antarmuka RegisterManager digunakan untuk melakukan proses pendaftaran member. User akan memasukkan *username*, *phone number*, *password* dan *confirm password* pada *textbox* yang disediakan kemudian tekan tombol submit. Sistem akan mengecek validasi data yang dimasukkan oleh user. Jika semua data yang dimasukkan oleh user valid maka sistem akan menyimpan data user ke dalam basis data sedangkan jika data user tidak valid maka sistem akan menampilkan pesan peringatan. Command back digunakan untuk kembali ke menu sebelumnya.

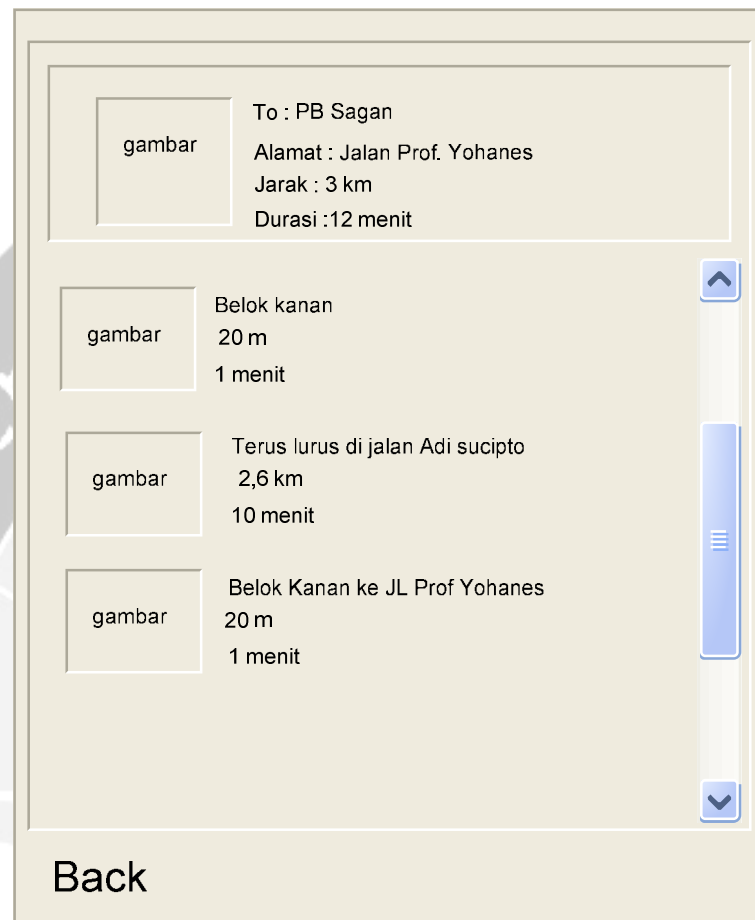
Antarmuka SearchNearLocation



Gambar 4.8 Antarmuka SearchNearLocation

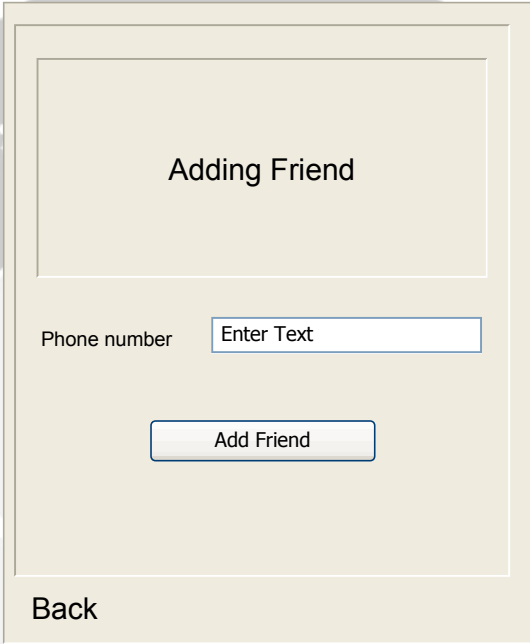
Antarmuka SearchNearLocation digunakan untuk menampilkan 4 data landmark yang terdekat dengan posisi user. Data-data landmark yang ditampilkan berurut secara ascending menurut jarak tempuh dari user ke lokasi. Command View Map digunakan untuk menampilkan peta sekaligus jalur yang ditempuh. Command back digunakan untuk kembali ke menu sebelumnya.

Antarmuka Route



Gambar 4.9 Antarmuka Route

Antarmuka Ruote digunakan untuk menampilkan data rute perjalanan yang akan membimbing user ke tujuan. Adapun Data-data yang ditampilkan yaitu data tujuan atau nama tujuan, alamat tujuan, jarak tempuh ke tujuan dan durasi ke tujuan. Selain data lokasi tujuan ditampilkan juga urutan rute yang ditempuh ke tujuan beserta dengan jarak dan durasinya. Command back digunakan untuk kembali ke menu sebelumnya.

Antarmuka AddingFriend

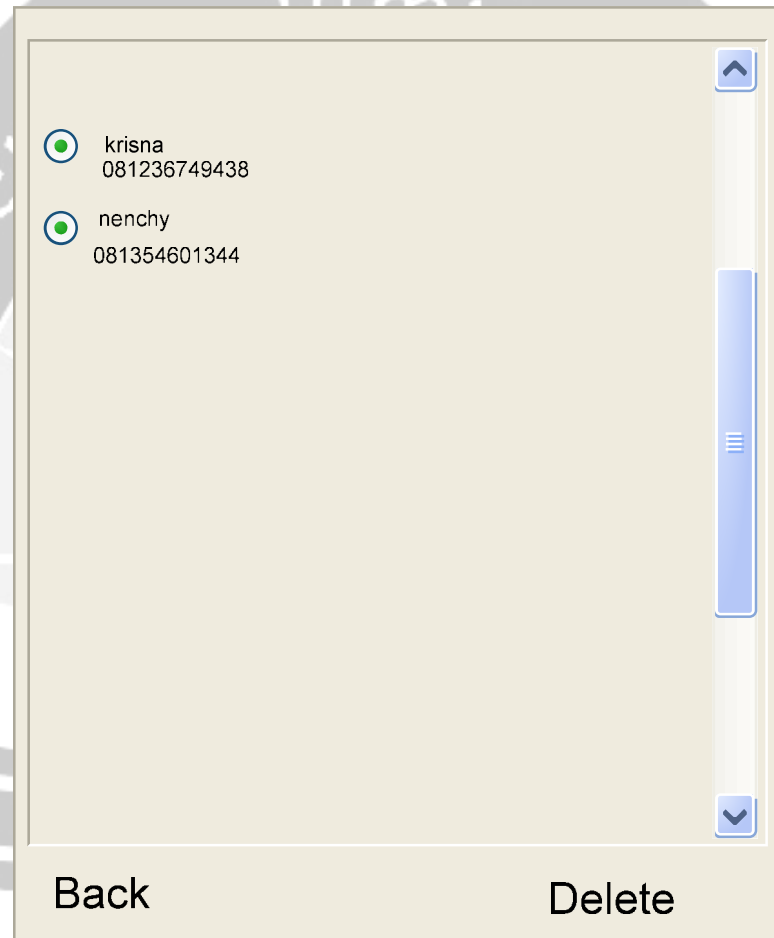
The image shows a screenshot of a web application interface for adding a friend. The interface is titled "Adding Friend" and is displayed within a light beige frame. Below the title, there is a label "Phone number" followed by a text input field containing the placeholder text "Enter Text". Below the input field is a button labeled "Add Friend". At the bottom of the frame, there is a "Back" button. The background of the image features a large, faint watermark of a circular emblem with the text "serviens" and "veritatis" and a central graphic.

Gambar 4.10 Antarmuka AddingFriend

Antarmuka AddingFriend digunakan untuk meng-invite user lain untuk menjadi teman. *User* akan memasukkan *phone number* user lain di *textboxt* yang telah disediakan kemudian tekan tombol *Add Friend*. Sistem akan memvalidasi data jika data valid maka data akan dikirim ke web service guna disimpan di basis data dan jika data tidak valid maka sistem akan menampilkan pesan peringatan. Ketika data telah di sampai di web service maka akan di cek data di basis data jika data user yang dikirimkan cocok atau tidak terdaftar. Kemudian setelah di cek pesan akan ditampilkan baik itu

sukses meng-invite user lain ataupun data user lain yang di invite tidak ada dalam basis data. Command back digunakan untuk kembali ke menu sebelumnya.

Antarmuka *DeleteFriend*

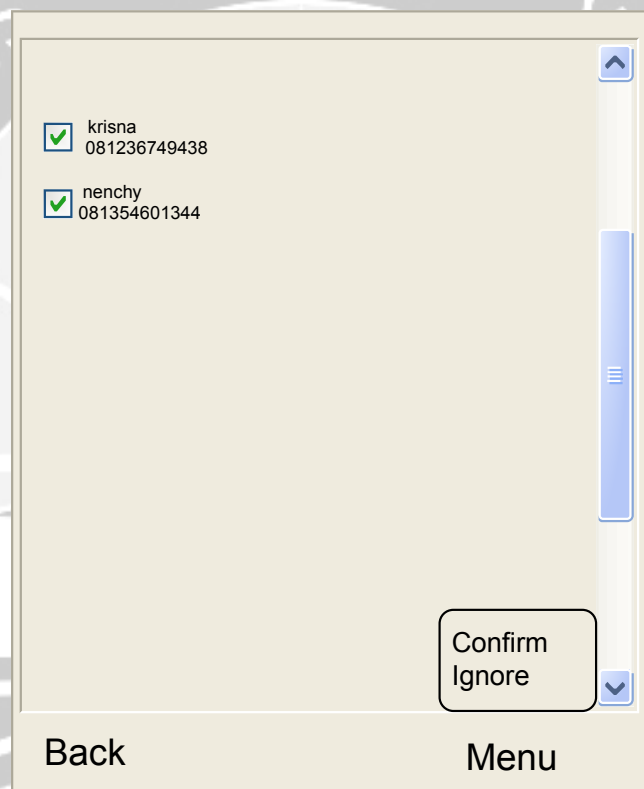


Gambar 4.11 Antarmuka DeleteFriend

Antarmuka DeleteFriend digunakan untuk menghapus user lain dari list teman. Sistem akan menampilkan semua data teman baik yang online ataupun tidak. User

akan user lain di *radiobutton* kemudian tekan command *Delete*. Sistem akan menanyakan kembali ke user apakah yakin akan dihapus. Jika user klik OK maka sistem akan menghapus relasi user dan disimpan di basis data. Command back digunakan untuk kembali ke menu sebelumnya.

Antarmuka *FriendRequest*



Gambar 4.12 Antarmuka FriendRequest

Antarmuka *FriendRequest* digunakan untuk melakukan proses menyetujui atau tidak bila ada user lain yang mengirim invitation. Pada tampilan ini terdapat list Check Box yang berisi user yang melakukan *invitation*.

Pada antarmuka ini terdapat tombol menu dan back. Tombol menu memiliki dua submenu yaitu Confirm dan Ignore. Submenu Confirm digunakan untuk menyetujui invitation dari user lain. Submenu Ignore digunakan untuk tidak menyetujui invitation dari user lain. Menu Back digunakan untuk kembali ke screen yang sebelumnya.

Antarmuka ViewMap



Gambar 4.13 Antarmuka ViewMap

Antarmuka ViewMap digunakan untuk menampilkan peta dan icon User yang menggambarkan posisi user aktif, icon lokasi tujuan yang menggambarkan tujuan user beserta dengan garis jalur yang harus ditempuh. Menu Back digunakan untuk kembali ke screen yang sebelumnya.