

BAB VI

KESIMPULAN DAN SARAN

6.1 Kesimpulan

Setelah aplikasi obral (*sale*) dengan nama SENDER ini selesai diimplementasikan dan berdasarkan dari pembahasan-pembahasan pada bab-bab sebelumnya, maka dapat ditarik beberapa kesimpulan, yaitu:

1. Perangkat Lunak SENDER (*Sale Finder*) telah berhasil dibangun sebagai aplikasi pencarian lokasi obral (*sale*) perangkat *mobile* yang dilengkapi modul GPS.
2. Perangkat Lunak SENDER telah berhasil dikembangkan menggunakan teknologi *Google Maps API*.
3. Dari hasil pengujian dapat diketahui bahwa fungsi-fungsi yang disediakan oleh perangkat lunak SENDER berjalan dengan benar dan sesuai dengan yang diharapkan.

6.2 Saran

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

1. Mengembangkan aplikasi agar dapat menambahkan lebih dari satu produk dan gambar untuk satu lokasi.
2. Menambahkan fitur untuk dapat melakukan rotasi gambar sebelum melakukan *upload* gambar sehingga foto atau gambar yang di-*upload* sesuai dengan keinginan.

DAFTAR PUSAKA

Belluccini, Luca. 2008. Monografia Di Laurea Android Platform and Application Development. Politecnico di Torino - III Facoltà di Ingegneria Corso di laurea in Ingegneria Informatica.

Chang, Shuchih Ernest, dkk. 2006. "Location-Based Services for Tourism Industryian Empirical Study". *SpringerLink*, 4156:1144-1153.

Cisco System, Inc. 2008. *Wi-Fi Location-Based Services 4.1 Design Guide*.

Falchuk, Ben, dkk, 2008. *Encyclopedia of Wireless and Mobile Communications*, New Jersey:Telcordia Technologies, Inc.

Hanafi, D., 2006, Mengungkap Cara Kerja GPS Receiver (Global Positioning System), Orari Daerah Jakarta.

Kazadi, Sanga-Ngoie. 2003. GIS And Remote-Sensing For High Resolution Mapping and Digital Cartographic Databasebuilding In African Remote Areas. *African Study Monographs*, Vol. 24, No. 4, pp 247-269.

Kiser, L. Chris., Kelly, J. Michael. 2010. GPS- vs. DEM-Derived Elevation Estimates from a Hardwood Dominated Forest Watershed. *Journal of Geographic Information System*, Vol. 2, pp 147-151.

Krievs, R., 2002, Using Fading to improve Accuracy of Cell Id Based Mobile Positioning Algorithms: Analysis of Special Cases, Scientific Proceedings of RTU. Series 7. Telecommunications and Electronics, 2002.

Kupper, Axel., Treu, Georg. 2005 From Location to Position Management: User Tracking for Location-based Services. Mobile and Distributed Systems Group, Institute for Informatics University of Munich, Germany.

Makodian, Nuraksa, Lingga Wardhana, 2010. *Teknologi Wireless Communication dan Wireless Broadband*, Yogyakarta:Andi Offset.

Meneses, Filipe., Moreira, Adriano. 2003. Using GSM CellID Positioning for Place Discovering. Dept of Information Systems University of Minho Guimarães, Portugal.

Muryani, Chatarina. 2005, Analisis ekosistem hutan mangrove di Pantai Pasuruan Menggunakan Metode Sistem Informasi Geografi. Universitas Negeri Surakarta Library.

Post, C. Christopher., Woodrow, Stephen. 2008. Location is Everything Balancing Innovation, Convenience, and Privacy in Location-based Technologies.

Prajapati, N. B., Agravat, R. R., Hasan , M. I. 2010. Simulated Annealing For Location Area Planning In Cellular Networks. International journal on applications of graph theory in wireless ad hoc network and sensor network (Grap-Hoc), Vol.2, No.1, March 2010.

Saor, Marganda Romel. 2006. Peta Zonasi Longsoran dengan Metode Anbalagan dan Sistem Informasi Geografi; Studi Kasus di Sub Daerah Aliran Sungai Citarum, Kecamatan Cipongkor, Kabupaten Bandung, Jawa Barat. Laboratory of Engineering Geology Department of Geological Engineering Faculty of Earth Sciences and Technology (FITB).

Simmonds, Chris. 2010. What else can you do with Android?.2net Limited.

Workman, Robbert., Gschwender, Adam., Chan, Jian Lian. 2005. Campus Google Map Applications.

Yudisputra H.,2000 Pengaruh Advertising and Sales Promotion terhadap penjualan pada PT. Supernusa Indogita Sidoharjo,Falkultas Ekonomi,Universitas Kristen Petra.

LAMPIRAN



DPPL

DESKRIPSI PERANCANGAN PERANGKAT LUNAK

SENDER (*Sale Finder*)

Untuk :
Universitas Atma Jaya Yogyakarta

Dipersiapkan oleh:
Jimmy Samuel Pardede. / 07 07 05432

Program Studi Teknik Informatika – Fakultas Teknologi
Industri
Universitas Atma Jaya Yogyakarta

	Program Studi Teknik Informatika Fakultas Teknologi Industri	Nomor Dokumen		Halama
		DPPL - SENDER		1/48

DAFTAR PERUBAHAN

Revisi	Deskripsi
A	
B	
C	
D	
E	
F	

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

Daftar Halaman Perubahan

Halaman	Revisi	Halaman	Revisi
			

Daftar Isi

1	Pendahuluan.....	7
1.1	Tujuan.....	7
1.2	Ruang Lingkup.....	7
1.3	Definisi dan Akronim.....	7
1.4	Referensi.....	8
2	Perancangan Sistem.....	10
2.1	Perancangan Arsitektur.....	10
2.2	Perancangan Rinci.....	11
2.2.1	Sequence Diagram	11
2.2.1.1	Login	11
2.2.1.2	Signup	12
2.2.1.3	Find Sale	13
2.2.1.4	Submit Location	14
2.2.1.5	Search Sale	15
2.2.1.6	View Details	16
2.2.1.7	View On Map	17
2.2.1.8	Submit Comment	18
2.2.1.9	Give Like	19
2.2.1.10	Show Route	20
2.2.1.11	View Newest Sale	21
2.2.1.12	Take Picture	22
2.2.1.13	Approve User	22
2.2.1.14	Approve Sale	23
2.2.2	Class Diagram	24
2.2.3	Spesifikasi Deskripsi Kelas Diagram ...	25
2.2.3.1	Spesifikasi Design Kelas informationDetail.....	25
2.2.3.2	Spesifikasi Design Kelas search	25
2.2.3.3	Spesifikasi Design Kelas nearMe	26
2.2.3.4	Spesifikasi Design Kelas bulletin	26
2.2.3.5	Spesifikasi Design Kelas comment	26
2.2.3.6	Spesifikasi Design Kelas mapping	27
2.2.3.7	Spesifikasi Design Kelas admin	27
2.2.3.8	Spesifikasi Design Kelas SenderActivity	28
2.2.3.9	Spesifikasi Design Kelas saveLocation .	28
2.2.3.10	Spesifikasi Design Kelas registration	29
2.2.3.11	Spesifikasi Design Kelas landmarksMngr	29
2.2.3.12	Spesifikasi Design Kelas userMngr	30
2.2.3.13	Spesifikasi Design Kelas googleApi ...	31
2.2.3.14	Spesifikasi Design Kelas like	31
2.2.3.15	Spesifikasi Design Kelas comments	32
2.2.3.16	Spesifikasi Design Kelas user	33
2.2.3.17	Spesifikasi Design Kelas saleLocations	34
3	Perancangan Data.....	38
3.1	Dekomposisi Data.....	38

3.1.1	Deskripsi Entitas Data USERS	38
3.1.2	Deskripsi Entitas Data SALE_LOCATIONS .	38
3.1.3	Deskripsi Entitas Data COMMENTS	39
3.1.4	Deskripsi Entitas Data LIKES	39
3.1.5	Deskripsi Entitas Data catagories	39
3.1.6	Physical Data Model	40
4	Perancangan Antarmuka.....	41
4.1	Sketsa Antarmuka dan Deskripsinya.....	41
4.1.1	Antarmuka <i>Login</i>	41
4.1.2	Antarmuka <i>Menu</i>	41
4.1.3	Antarmuka Halaman <i>Near Me</i>	42
4.1.4	Antarmuka Halaman <i>View Detail</i>	43
4.1.5	Antarmuka Halaman <i>View Comments</i>	43
4.1.6	Antarmuka Halaman <i>View On Map</i>	44
4.1.7	Antarmuka Halaman <i>View Route</i>	45
4.1.8	Antarmuka Halaman <i>Save Sale Location</i> ..	46
4.1.9	Antarmuka Halaman <i>View Neweset</i>	47
4.1.10	Antarmuka Halaman <i>Search</i>	47
4.1.11	Antarmuka Halaman <i>View My Position</i> ...	48

Daftar Gambar

Gambar 2.1	Rancangan Arsitektur SENDER.....	10
Gambar 2.2	Sequence Diagram : Login.....	11
Gambar 2.3	Sequence Diagram : Signup.....	12
Gambar 2.4	Sequence Diagram : Find Sale.....	13
Gambar 2.5	Sequence Diagram : Submit Location.....	14
Gambar 2.6	Sequence Diagram : Search Location.....	15
Gambar 2.7	Sequence Diagram : View Details.....	16
Gambar 2.8	Sequence Diagram : View On Map.....	17
Gambar 2.9	Sequence Diagram : Submit Comment.....	18
Gambar 2.10	Sequence Diagram : Give Like.....	19
Gambar 2.11	Sequence Diagram : Show Route.....	20
Gambar 2.12	Sequence Diagram : View Newest Sale.....	21
Gambar 2.13	Sequence Diagram : Take Picture.....	22
Gambar 2.14	Sequence Diagram : Approve User.....	22
Gambar 2.15	Sequence Diagram : Approve Sale.....	23
Gambar 2.16	Class Diagram SENDER.....	24
Gambar 3.1	Physical Data Model.....	40
Gambar 4.1	Rancangan Antarmuka Login.....	41
Gambar 4.2	Rancangan Antarmuka Menu.....	41
Gambar 4.3	Rancangan Antarmuka Near Me.....	42
Gambar 4.4	Rancangan Antarmuka View Detail.....	43
Gambar 4.5	Rancangan Antarmuka View Comments.....	43
Gambar 4.6	Rancangan Antarmuka View On Map.....	44
Gambar 4.8	Rancangan Antarmuka Save Sale Location...	46
Gambar 4.9	Rancangan Antarmuka View Neweset.....	47
Gambar 4.10	Rancangan Antarmuka Search.....	47
Gambar 4.11	Rancangan Antarmuka View My Position....	48

1 Pendahuluan

1.1 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.

1.2 Ruang Lingkup

Perangkat Lunak SENDER dikembangkan dengan tujuan untuk :

1. Pencarian posisi *device* pada saat itu (*real time*).
2. Menangani pencarian lokasi obral terdekat dari posisi pengguna.
3. Menangani penyimpanan data lokasi obral.
4. Menangani informasi seputar obral yang ditampilkan.
5. Menangani navigasi rute perjalanan dari posisi pengguna sekarang hingga sampai pada lokasi obral.
6. Menangani pemberian komentar dan penandaan suka pada suatu obral.
7. Dan berjalan pada lingkungan dengan platform Android.

1.3 Definisi dan Akronim

Daftar definisi akronim dan singkatan :

Keyword/Phrase	Definisi
SKPL	Merupakan spesifikasi kebutuhan dari perangkat lunak yang akan dikembangkan.

SKPL-SENDER	Kode yang merepresentasikan kebutuhan pada SENDER (SALE FINDER)
SENDER	Aplikasi <i>mobile</i> untuk mencari lokasi-lokasi obral terdekat dari <i>user</i>
Android	Sebuah sistem operasi yang dirancang oleh google untuk perangkat <i>mobile</i> .
Internet	Internet merupakan istilah umum yang dipakai untuk menunjuk <i>Network</i> global yang terdiri dari komputer dan layanan servis dengan sekitar 30 sampai 50 juta pemakai komputer dan puluhan layanan informasi termasuk e-mail, FTP, dan World Wide Web.
GPS	<i>Global Positioning System</i> (SENDERS) merupakan perangkat untuk mengetahui posisi koordinat di permukaan bumi
aGPS	<i>Asisted Global Positioning System</i> (aGPS) suatu terobosan baru SENDERS di masa sekarang ini dengan ukuran relatif kecil
Google Map API	Suatu <i>library</i> yang berbentuk <i>JavaScript</i>

1.4 Referensi

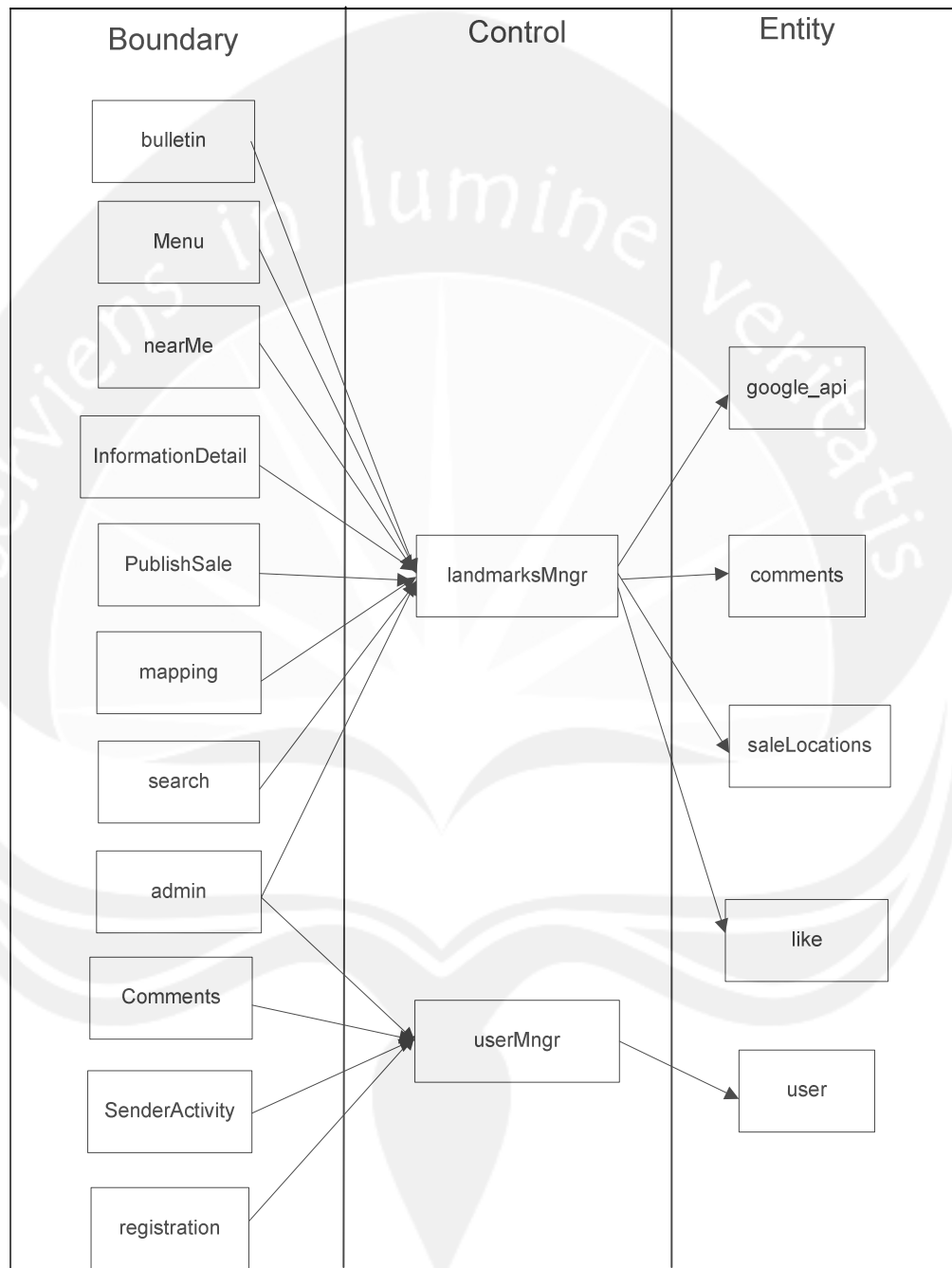
Referensi yang digunakan pada perangkat lunak tersebut adalah:

1. Bennet Simon, McRobb Steve, Farmer Ray, *Object-Oriented System Analysis and Design Using UML*, McGraw-Hill Companies, 2002.
2. Boggs Wendy, Boggs Michael, *Mastering UML with Rational Rose 2002*, SYBEX Inc, 2002.

3. Samuel P., Jimmy. *Spesifikasi Kebutuhan Perangkat Lunak SENDER*, Universitas Atma Jaya Yogyakarta, 2011.
4. Sapta, Juli. *Deskripsi Perancangan Perangkat Lunak SC3*, Universitas Atma Jaya Yogyakarta, 2006.
5. Samuel P. Jimmy, *Spesifikasi Kebutuhan Perangkat Lunak WHISPER*, Universitas Atma Jaya Yogyakarta, 2009.

2 Perancangan Sistem

2.1 Perancangan Arsitektur

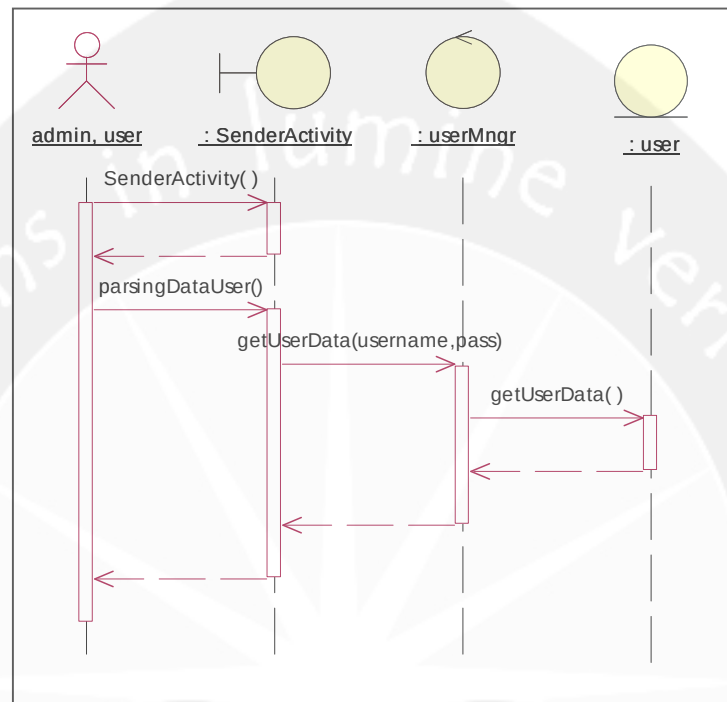


Gambar 2.1 Rancangan Arsitektur SENDER

2.2 Perancangan Rinci

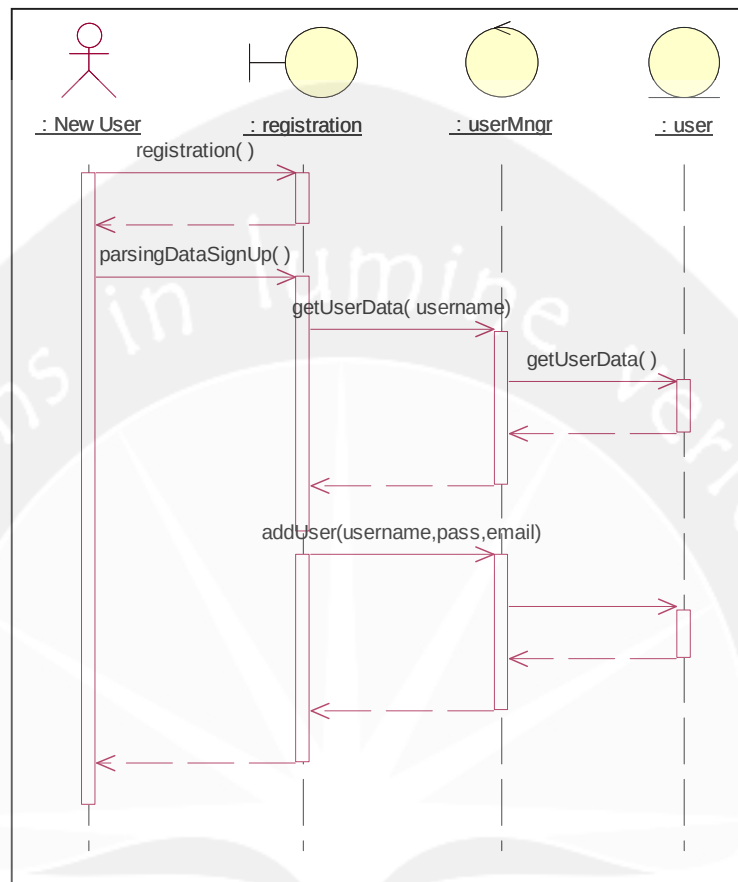
2.2.1 Sequence Diagram

2.2.1.1 Login



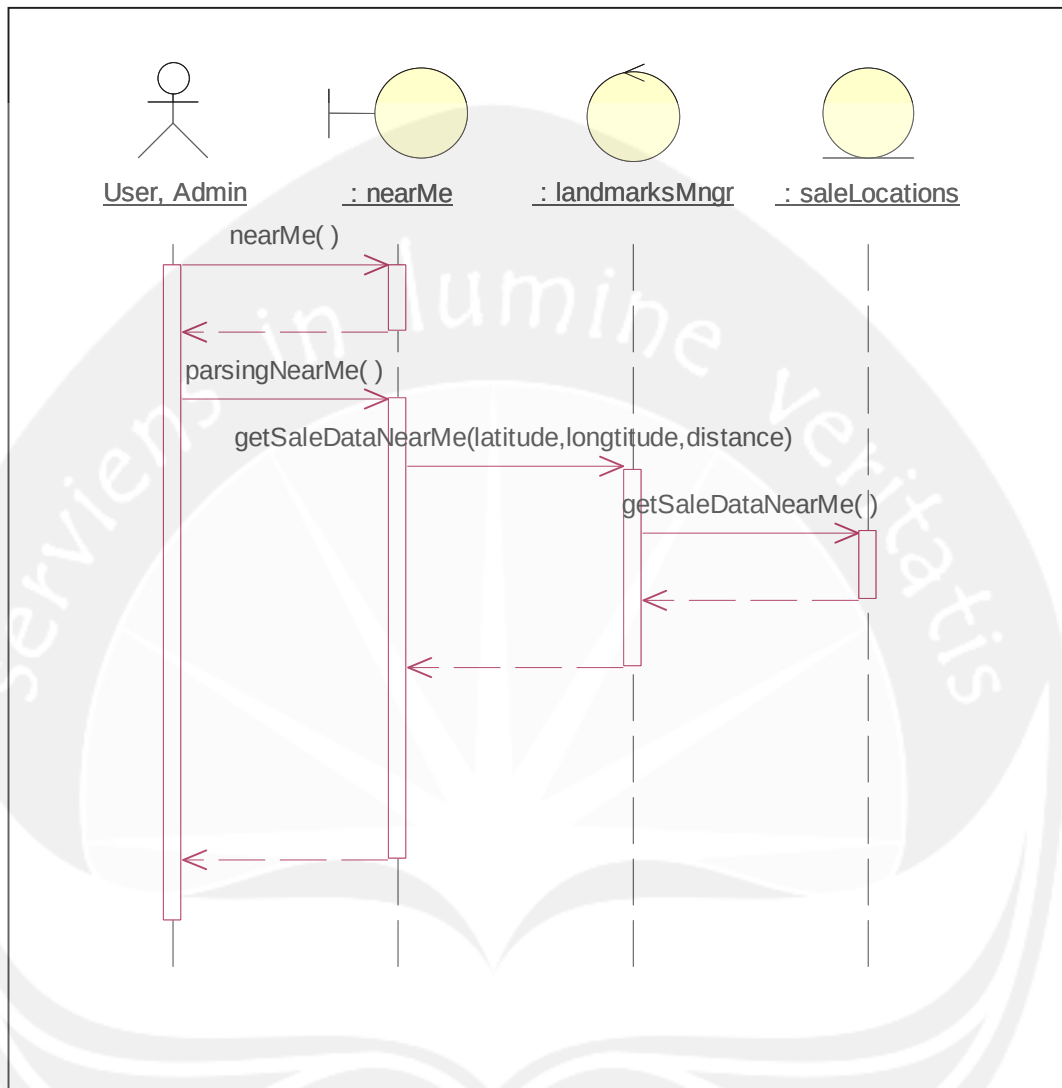
Gambar 2.2 Sequence Diagram : Login

2.2.1.2 Signup



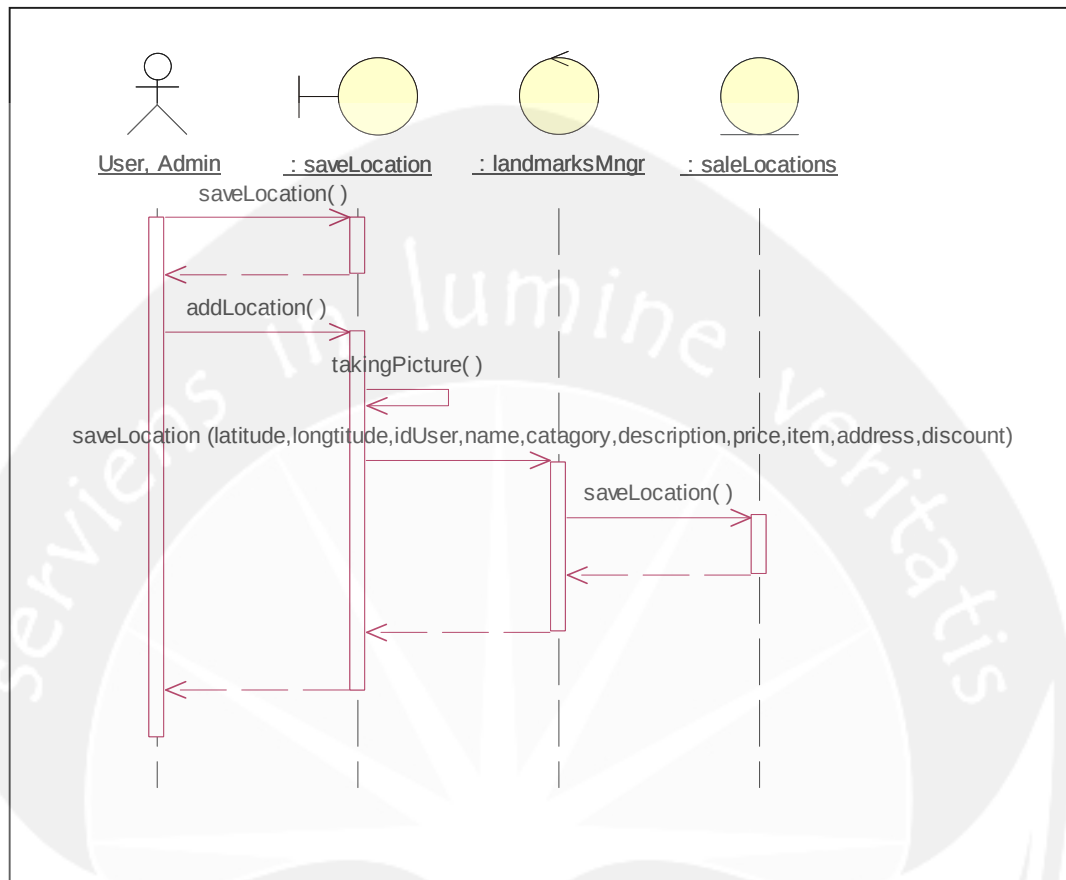
Gambar 2.3 Sequence Diagram : Signup

2.2.1.3 Find Sale



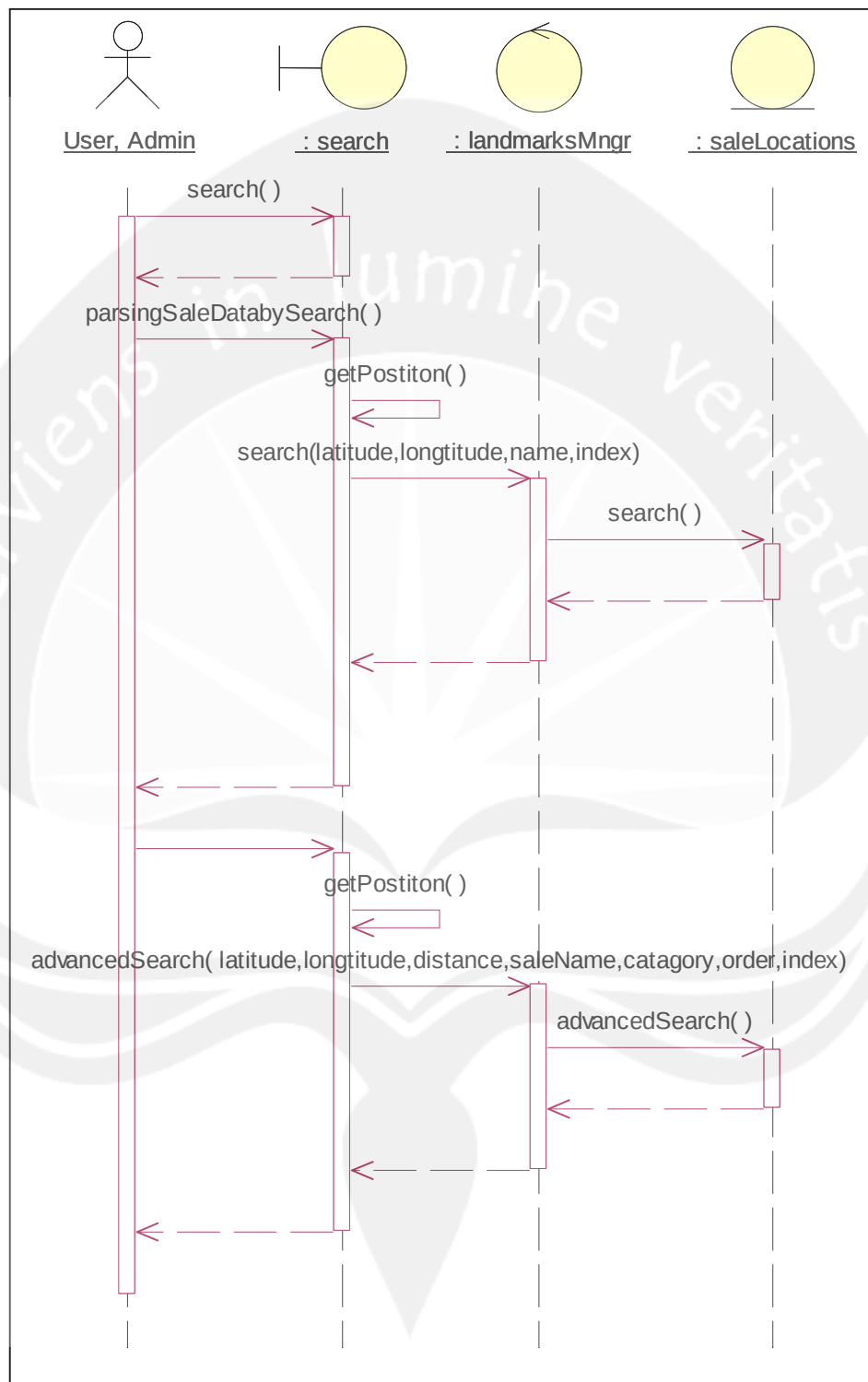
Gambar 2.4 Sequence Diagram : Find Sale

2.2.1.4 *Submit Location*



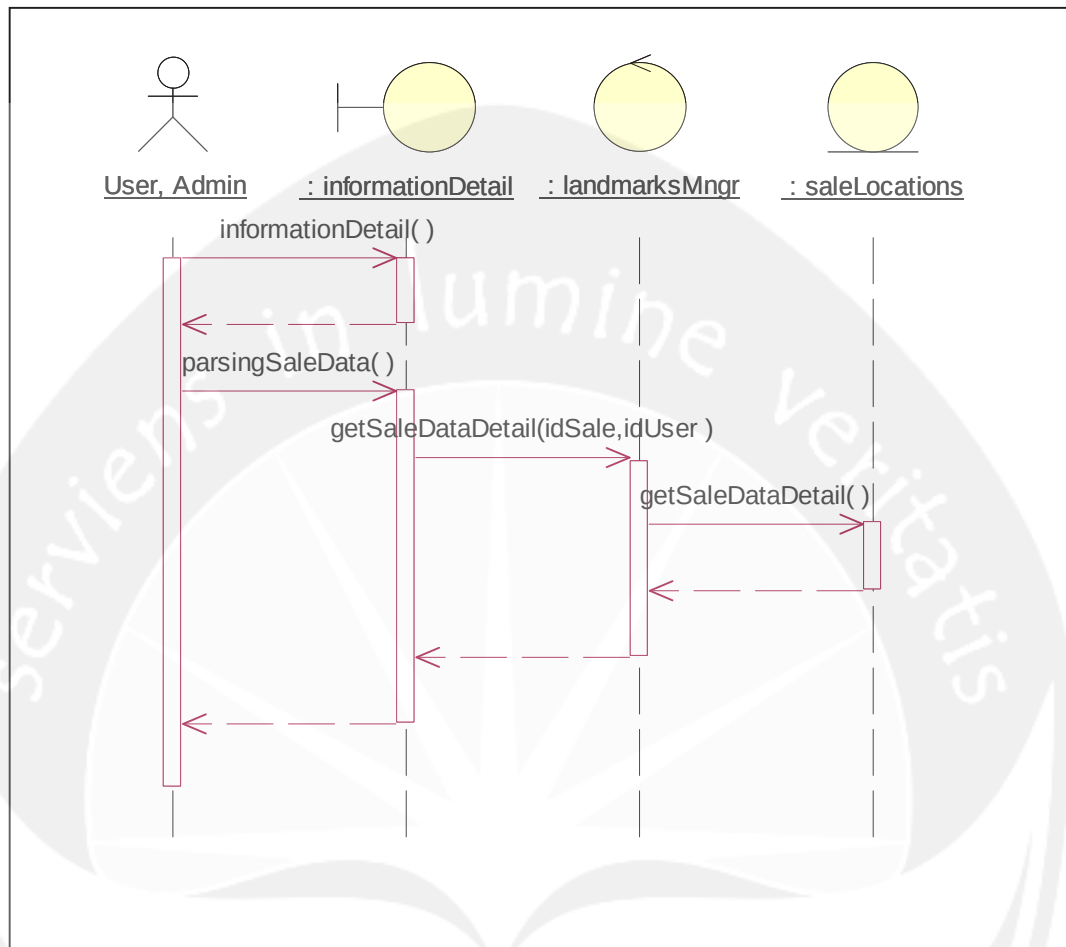
Gambar 2.5 Sequence Diagram : *Submit Location*

2.2.1.5 Search Sale



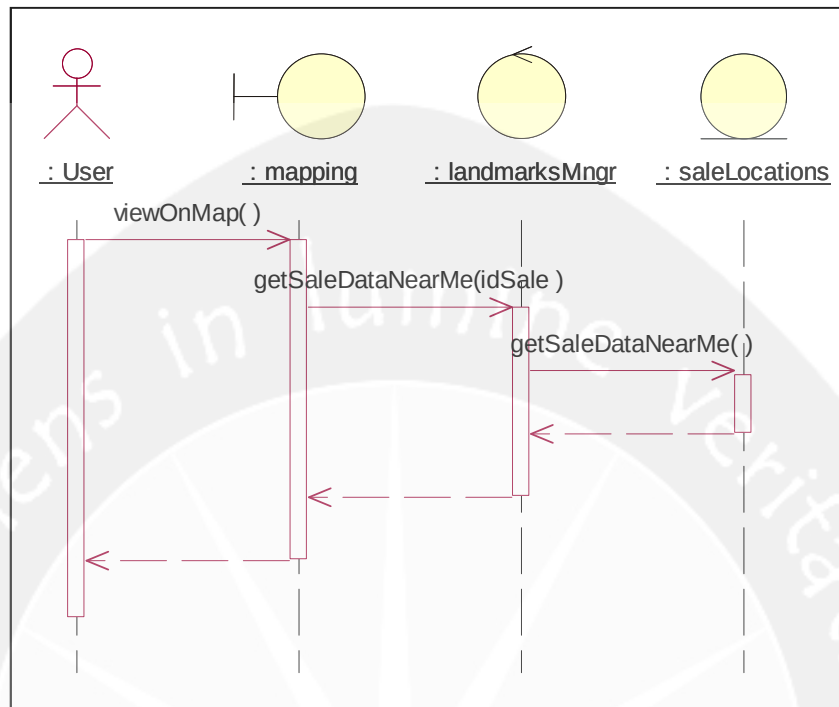
Gambar 2.6 Sequence Diagram : Search Location

2.2.1.6 View Details



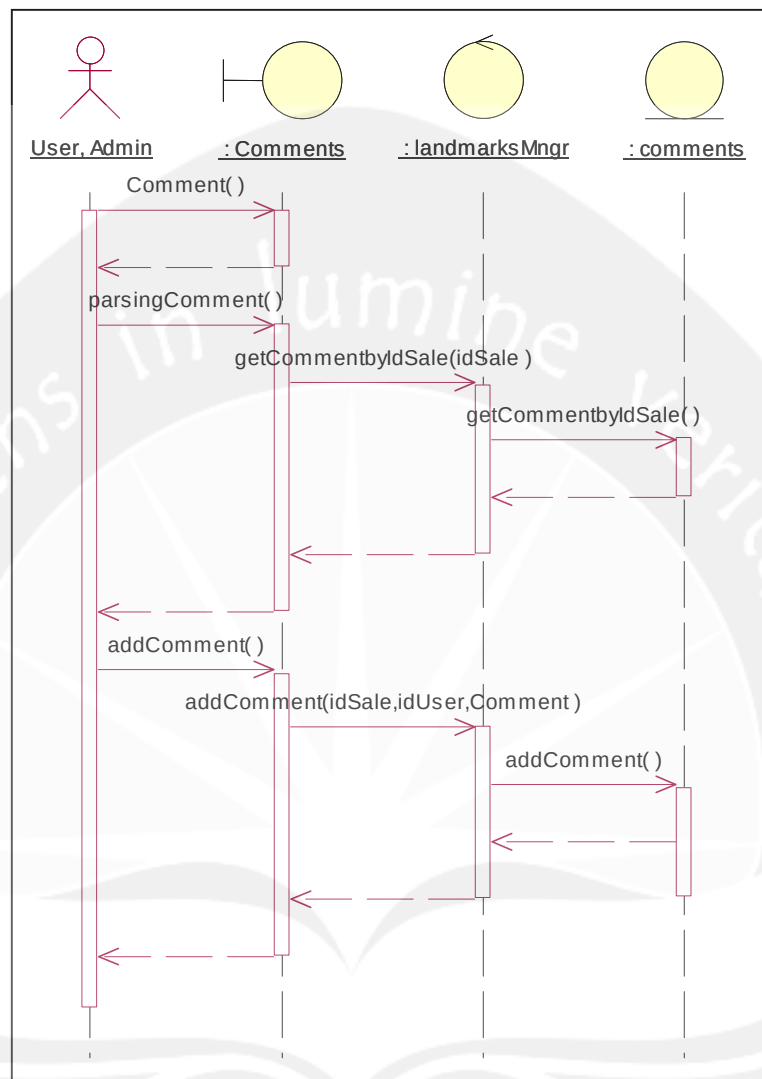
Gambar 2.7 Sequence Diagram : View Details

2.2.1.7 View On Map



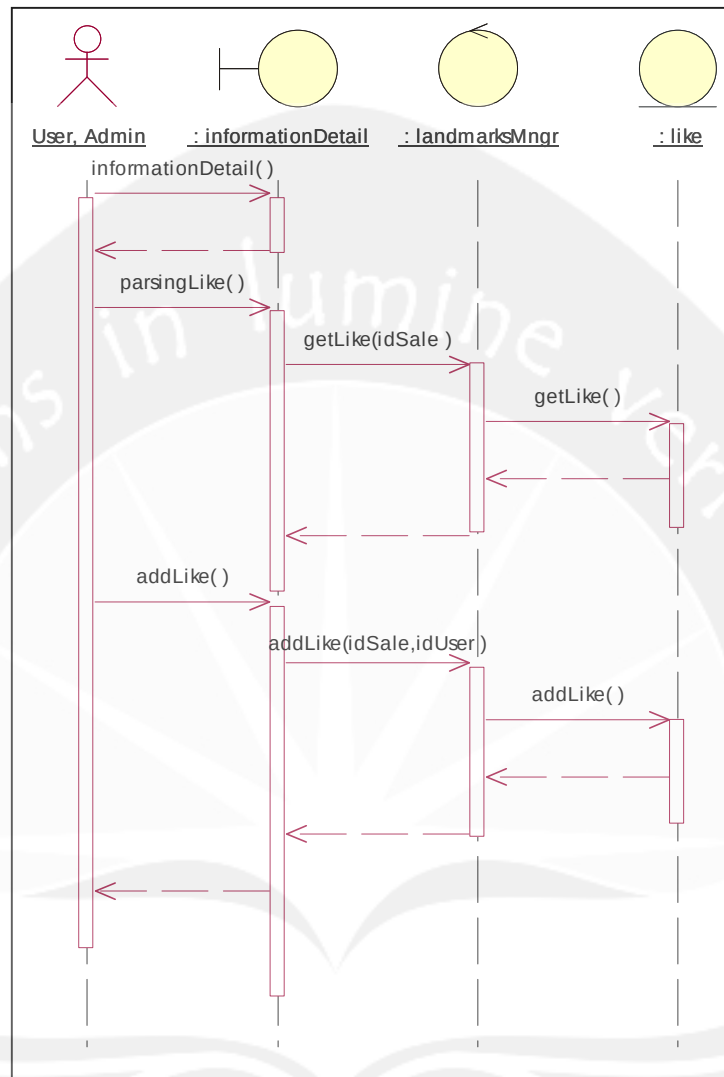
Gambar 2.8 Sequence Diagram : View On Map

2.2.1.8 *Submit Comment*



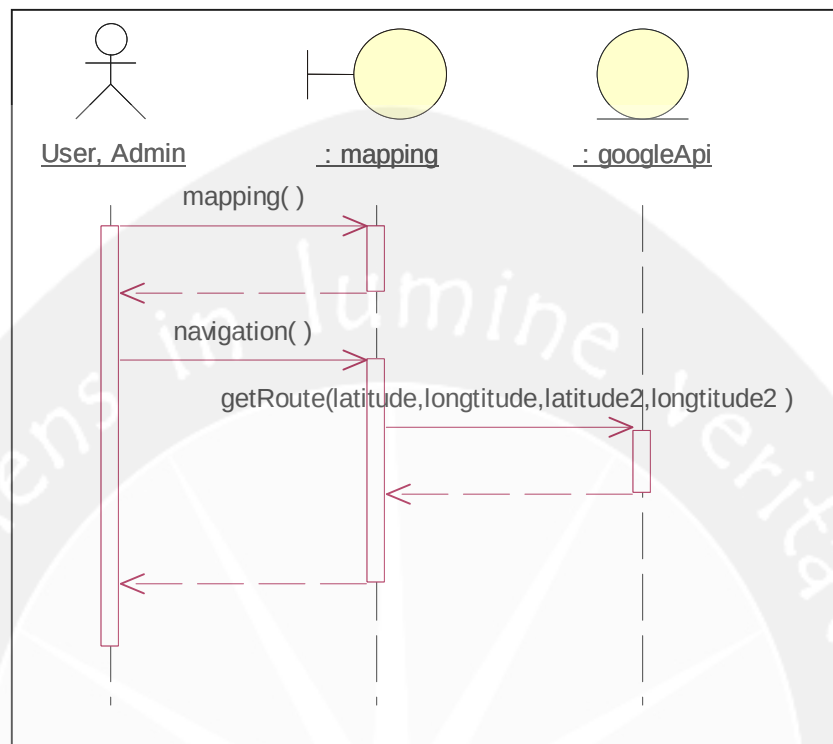
Gambar 2.9 Sequence Diagram : *Submit Comment*

2.2.1.9 Give Like



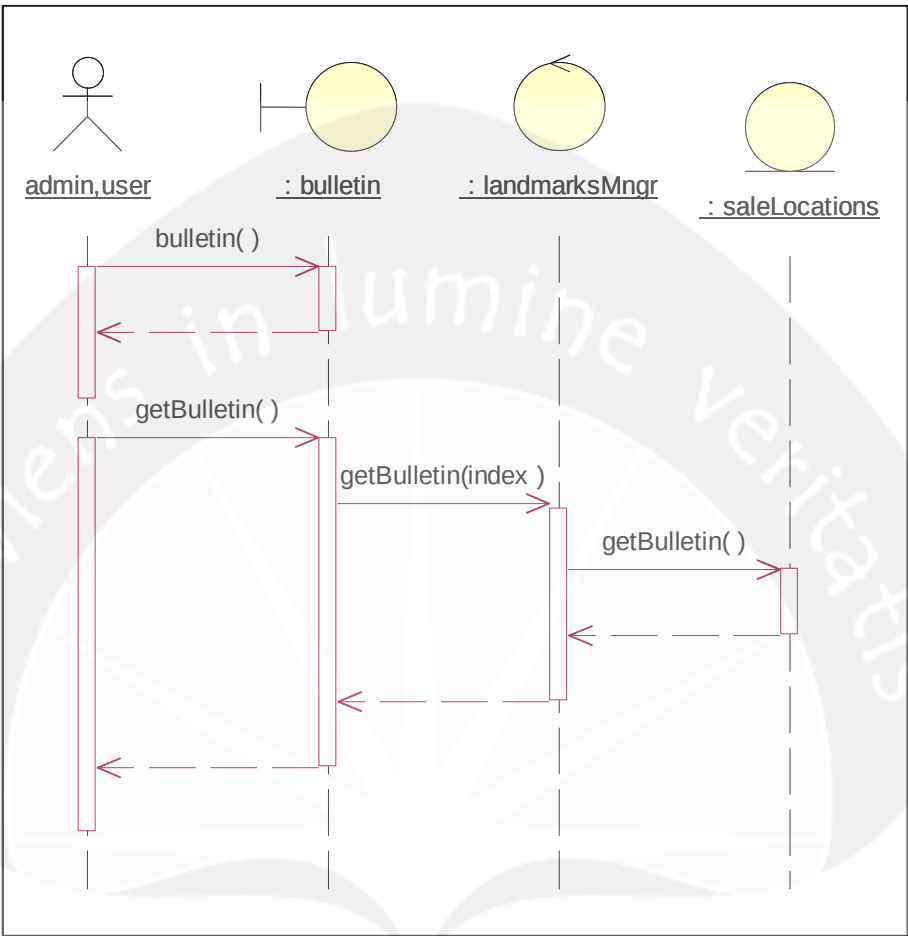
Gambar 2.10 Sequence Diagram : Give Like

2.2.1.10 Show Route



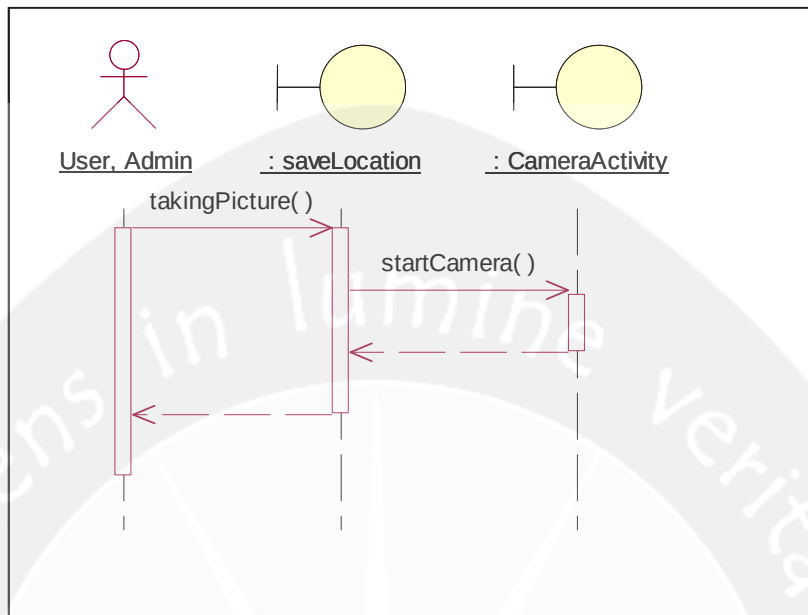
Gambar 2.11 Sequence Diagram : Show Route

2.2.1.11 *View Newest Sale*



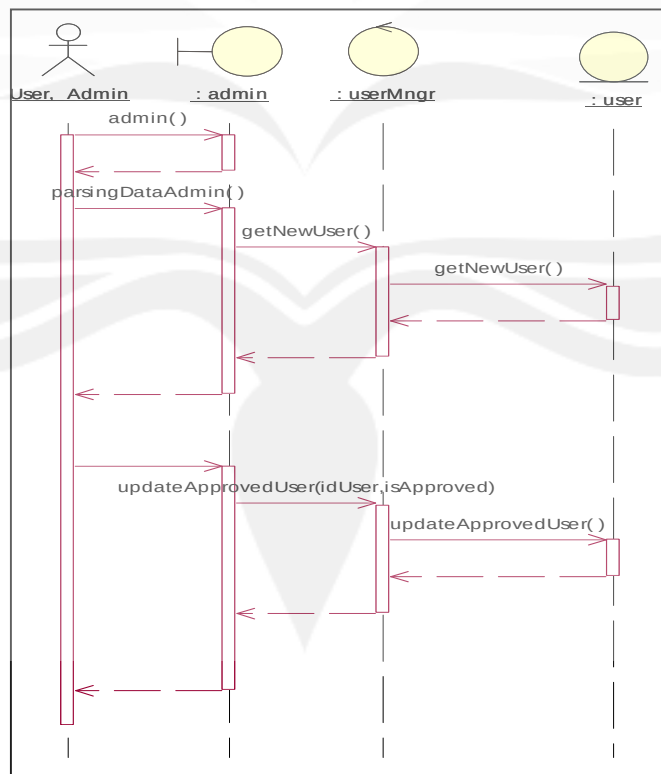
Gambar 2.12 Sequence Diagram : *View Newest Sale*

2.2.1.12 Take Picture



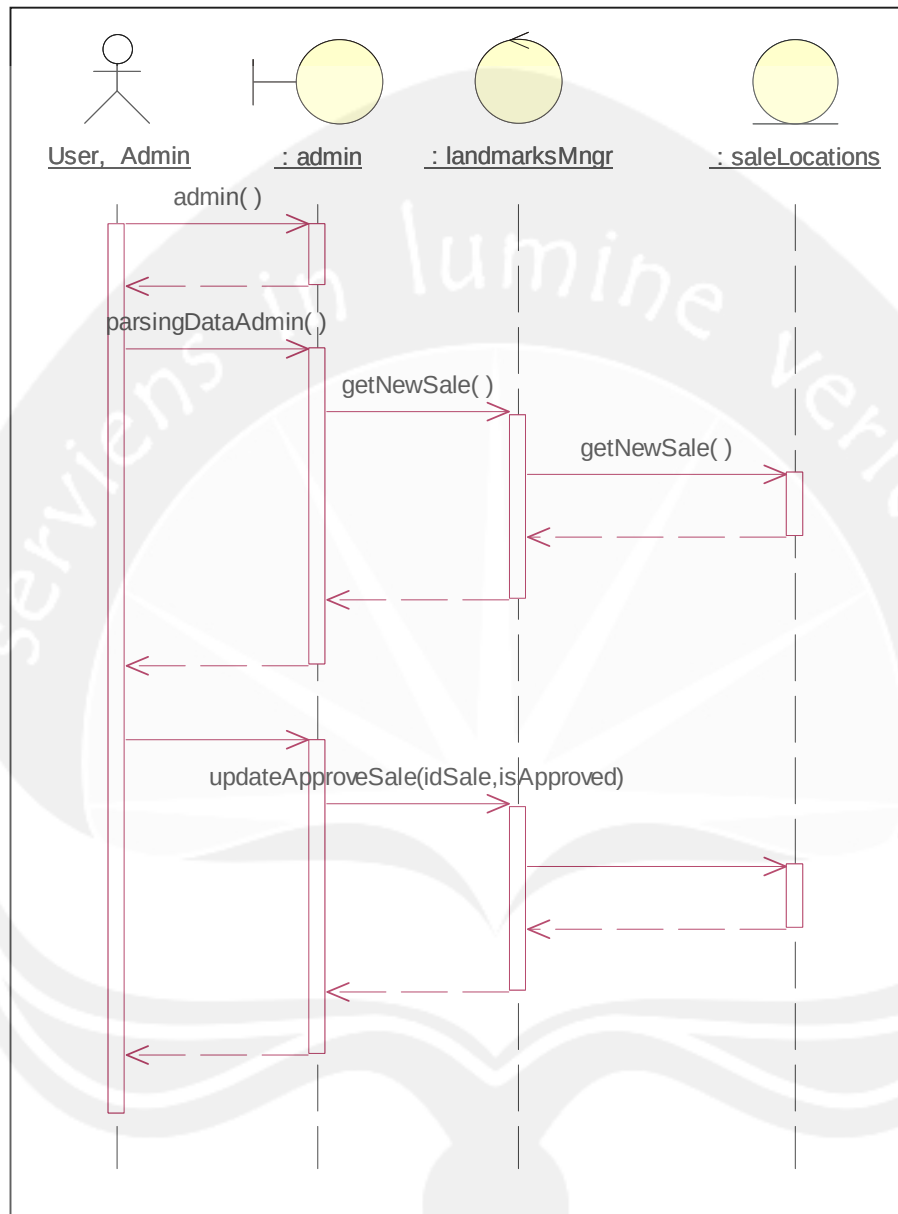
Gambar 2.13 Sequence Diagram : Take Picture

2.2.1.13 Approve User



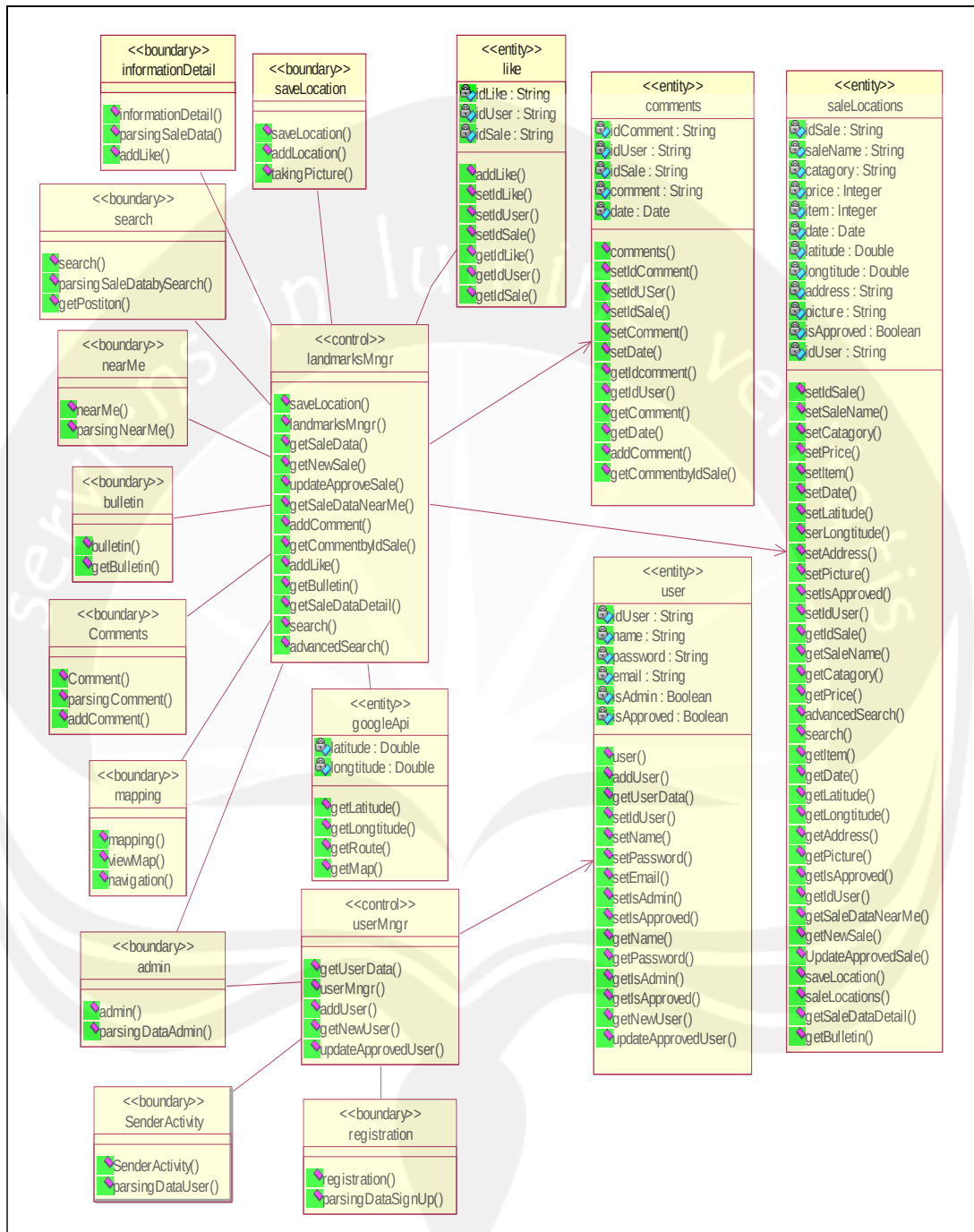
Gambar 2.14 Sequence Diagram : Approve User

2.2.1.14 Approve Sale



Gambar 2.15 Sequence Diagram : Approve Sale

2.2.2 Class Diagram



Gambar 2.16 Class Diagram SENDER

2.2.3 Spesifikasi Deskripsi Kelas Diagram

2.2.3.1 Spesifikasi Design Kelas `informationDetail`

<code>informationDetail</code>	<code><<boundary>></code>
<pre>+informationDetail() Default konstruktor, digunakan untuk inisialisasi semua atribut dari kelas ini. +parsingSaleData() Operasi ini digunakan untuk membaca informasi sale dengan detail. +addLike(idSale,idUser) : String Fungsi ini digunakan untuk menambah data like sale bersangkutan. +getSaleDataDetail(idSale,idUser) : String Fungsi ini digunakan untuk menampilkan informasi lengkap dari idSale tertentu berserta data 'like'.</pre>	

2.2.3.2 Spesifikasi Design Kelas `search`

<code>Search</code>	<code><<boundary>></code>
<pre>+search() Default konstruktor, digunakan untuk inisialisasi semua atribut dari kelas ini. +search(latitude,longtitude,name,index) : String Fungsi ini digunakan untuk mendapatkan sale berdasarkan inputan pengguna berupa nama produk atau toko. +advanceSearch(latitude,longtitude,distance,name,catagory,or der,index) : String Fungsi ini digunakan untuk mendapatkan sale berdasarkan filter yang dimasukkan pengguna berupa nama produk atau toko.</pre>	

2.2.3.3 Spesifikasi Design Kelas nearMe

nearMe	<<boundary>>
<pre>+nearMe() Default konstruktor, digunakan untuk inisialisasi semua atribut dari kelas ini. +getSaleDataNearMe(latitude,longtitude,distance) : String Fungsi yang digunakan untuk mendapatkan sale terdekat dari pengguna.</pre>	

2.2.3.4 Spesifikasi Design Kelas bulletin

Bulletin	<<boundary>>
<pre>+bulletin() Default konstruktor, digunakan untuk inisialisasi semua atribut dari kelas ini. +getBulletin(index) : String Fungsi yang digunakan untuk mendapatkan informasi sale yang terbaru dengan index tertentu.</pre>	

2.2.3.5 Spesifikasi Design Kelas comment

Comment	<<boundary>>
<pre>+comment() Default konstruktor, digunakan untuk inisialisasi semua atribut dari kelas ini. +parsingComment() operasi yang digunakan untuk membaca komentar pada suatu sale. +getCommentbyIdSale(idSale) : String Fungsi yang digunakan untuk mendapatkan komentar-komentar pada idSale tertentu. +addComment(idSale,idUser,Comment) : String Fungsi yang digunakan untuk menambah komentar pada idSale</pre>	

tertentu

2.2.3.6 Spesifikasi Design Kelas mapping

Comment	<<boundary>>
-mapView : MapView Atribut ini untuk menyimpan tampilan peta Google -point : GeoPoint Atribut ini untuk menyimpan titik koordinat lokasi -overlayItem : OverlayItem Atribut ini untuk menyimpan item tertentu, untuk ditampilkan pada tampilan peta Google	
+mapping() Default konstruktor, digunakan untuk inisialisasi semua attribute dari kelas ini + viewOnMap() Operasi yang digunakan untuk menampilkan posisi sale di peta. +viewRoute() Operasi yang digunakan untuk menampilkan <i>route</i> dari posisi ke lokasi sale di peta.	

2.2.3.7 Spesifikasi Design Kelas admin

Admin	<<boundary>>
+admin() Default konstruktor, digunakan untuk inisialisasi semua atribut dari kelas ini. + parsingDataAdmin() Operasi yang digunakan untuk membaca <i>sale</i> dan pengguna yang terbaru. +updateApproveSale(idSale,isApproved) : String Fungsi yang digunakan untuk mengubah status <i>approved</i> pada suatu <i>sale</i>. +updateApproveUser(idUser,isApproved) : String	

Fungsi yang digunakan untuk mengubah status *approved* pada seorang user.

+ `getNewSale() : String`

Fungsi yang digunakan untuk mendapatkan *sale* yang terdaftar dan belum di *approved*.

+ `getNewUser() : String`

Fungsi yang digunakan untuk mendapatkan pengguna yang terdaftar dan belum di *approved*.

2.2.3.8 Spesifikasi Design Kelas `SenderActivity`

<code>SenderActivity</code>	<<boundary>>
+ <code>SenderActivity()</code> Default konstruktor, digunakan untuk inisialisasi semua atribut dari kelas ini. + <code>parsingDataUser()</code> operasi yang digunakan untuk membaca data pengguna. + <code>getUserData(username,pass) : String</code> Fungsi yang digunakan untuk mendapatkan data pengguna dengan nama 'username' dan passwordnya bernilai 'pass'.	

2.2.3.9 Spesifikasi Design Kelas `saveLocation`

<code>saveLocation</code>	<<boundary>>
+ <code>saveLocation()</code> Default konstruktor, digunakan untuk inisialisasi semua atribut dari kelas ini. + <code>addLocation()</code> operasi yang digunakan menambah <i>sale location</i>. + <code>takingPicture()</code> Operasi yang digunakan untuk mengambil lokasi <i>sale</i> atau produk yang dijual. + <code>addLocation(latitude,idUser, longtitude,name,category,description,price,item,address,discount)</code>	

Operasi yang digunakan untuk menambah *sale location* berserta data-datanya

2.2.3.10 Spesifikasi Design Kelas registration

Registration	<<boundary>>
<pre>+ registration() Default konstruktor, digunakan untuk inisialisasi semua atribut dari kelas ini. + parsingDataSignUp() Operasi yang digunakan untuk membaca data-data yang diinput untuk melakukan pendaftaran pengguna baru. + getUserData(username) : String Fungsi yang digunakan untuk mendapatkan data yang sama dengan username yang di <i>input</i> pengguna. + addUser(username,pass,email) : String Fungsi yang digunakan untuk menambah pengguna baru berserta data-datanya.</pre>	

2.2.3.11 Spesifikasi Design Kelas landmarksMngr

landmarksMngr	<<control>>
<pre>+ landmarksMngr() Default konstruktor, digunakan untuk inisialisasi semua atribut dari kelas ini. + addComment(idSale,comment) : String Fungsi yang digunakan untuk menambah komentar pada <i>sale</i> tertentu. + updateApprovedSale(idSale) : String Operasi yang digunakan untuk mengubah status approved pada idSale tertentu. + getSaleData(idSale) : String Fungsi yang digunakan untuk mendapatkan informasi <i>sale</i> berdasarkan idSale tertentu.</pre>	

```
+saveLocation(latitude,idUser,longtitude,name,category,
description,price,item,address,discount) : String
```

Fungsi yang digunakan untuk menambah *location sale* berserta data-datanya.

```
+addLike(idSale,idUser) : String
```

Fungsi yang digunakan untuk menambah data jumlah orang yang menyukai sale pada idSale tertentu.

```
+search(latitude,longtitude,name,index) : String
```

Fungsi ini digunakan untuk mendapatkan sale berdasarkan inputan pengguna berupa nama produk atau toko.

```
+advanceSearch(latitude,longtitude,distance,name,catagory,order,index) : String
```

Fungsi ini digunakan untuk mendapatkan sale berdasarkan filter yang dimasukkan pengguna berupa nama produk atau toko.

```
+getLike(idSale) : String
```

Fungsi yang digunakan untuk mendapatkan jumlah orang yang menyukai sale pada idSale tertentu.

```
+getBulletin(index) : String
```

Fungsi yang digunakan untuk mendapatkan informasi bulletin berdsarkn index tertentu

2.2.3.12 Spesifikasi Design Kelas userMngr

userMngr	<<control>>
<pre>+userMngr ()</pre> Default konstruktor, digunakan untuk inisialisasi semua atribut dari kelas ini. <pre>+getUserData(username,pass) : String</pre> Fungsi yang digunakan untuk mendapatkan informasi dengan nama username dan passwordnya pass <pre>+addUser(username,pass,email) : String</pre> Fungsi yang digunakan untuk menambah pengguna baru berserta data-datanya.	

+updateApprovedUser(isApproved) : String
Fungsi yang digunakan untuk mendapatkan pengguna dengan status isApproved bernilai false.

2.2.3.13 Spesifikasi Design Kelas googleApi

googleApi	<<entity>>
-latitude : Double Atribut ini untuk menyimpan data latitude. -longitude : Double Atribut ini untuk menyimpan data longitude.	
+getLatitude() : Double Fungsi yang digunakan untuk mendapatkan latitude. +getLongitude() : Double Fungsi yang digunakan untuk mendapatkan longitude. getRoute() Operasi untuk menampilkan rute dari posisi <i>user</i> ke suatu <i>sale location</i> .	

2.2.3.14 Spesifikasi Design Kelas like

like	<<entity>>
-idLike: Integer Atribut ini untuk menyimpan data id <i>like</i> . -idUser : Integer Atribut ini untuk menyimpan data id pengguna. -idSale: String Atribut ini untuk menyimpan data id Sale.	
+like() Default konstruktor, digunakan untuk inisialisasi semua atribut dari kelas ini. +setIdLike() Prosedur untuk menginisialisasi atribut id <i>like</i> . +setIdUser() Prosedur untuk menginisialisasi atribut id pengguna. +setIdSale()	

Prosedur untuk menginisialisasi atribut id Sale.

+getIdLike() : Integer

Fungsi untuk mendapatkan nilai dari atribut id *like*.

+getIdUser() : Integer

Fungsi untuk mendapatkan nilai dari atribut id pengguna.

+getIdSale() : Integer

Fungsi untuk mendapatkan nilai dari atribut id Sale.

+getLike() : String

Fungsi untuk mendapatkan jumlah orang yang menyukai pada idSale tertentu.

+addLike() : String

Fungsi untuk menambah jumlah orang menyukai id sale tertentu.

2.2.3.15 Spesifikasi Design Kelas comments

comments	<<entity>>
-idComment : Integer Atribut ini untuk menyimpan data id komentat. - idUser : Integer Atribut ini untuk menyimpan data id pengguna. -idSale : Integer Atribut ini untuk menyimpan data id <i>sale location</i>. -comment : String Atribut ini untuk menyimpan data komentar. -date : Date Atribut ini untuk menyimpan data tanggal.	
+comments() Default konstruktor, digunakan untuk inisialisasi semua atribut dari kelas ini. +setIdComment() Prosedur untuk menginisialisasi atribut id komentar. +setIdUser() Prosedur untuk menginisialisasi atribut id pengguna. +setIdSale() Prosedur untuk menginisialisasi atribut id <i>sale location</i>.	

+setComment()
Prosedur untuk menginisialisasi atribut komentar.

+setDate()
Prosedur untuk menginisialisasi atribut tanggal.

+getIdComment() : Integer
Fungsi untuk mendapatkan nilai dari atribut id komentar.

+getIdUser() : Integer
Fungsi untuk mendapatkan nilai dari atribut id pengguna.

+getComment() : String
Fungsi untuk mendapatkan nilai dari atribut komentar.

+getDate() : Date
Fungsi untuk mendapatkan nilai dari atribut Tanggal.

+getCommentbyIdSale() : String
Fungsi untuk mendapatkan komentar berdasarkan id *sale location*.

+addComment() : String
Fungsi untuk menambah komentar.

2.2.3.16 Spesifikasi Design Kelas user

User	<<entity>>
-idUser: Integer Atribut untuk menyimpan data id pengguna. -name : String Atribut untuk menyimpan data nama pengguna. -password : String Atribut untuk menyimpan data password pengguna. -isAdmin : Boolean Atribut untuk menyimpan data status <i>rule</i> dari pengguna. -isApproved : Boolean Atribut untuk menyimpan data status penerimaan pengguna.	
+user() Default konstruktor, digunakan untuk inisialisasi semua atribut dari kelas ini. +setIdUser()	

Prosedur untuk menginisialisasi atribut id pengguna.
+setName()
Prosedur untuk menginisialisasi atribut nama pengguna.
+setPassword()
Prosedur untuk menginisialisasi atribut password pengguna.
+setIsAdmin()
Prosedur untuk menginisialisasi status *rule* dari pengguna.
+setIsApproved()
Prosedur untuk menginisialisasi status penerimaan pengguna.
+getIdUser() : Integer
Fungsi untuk mendapatkan nilai atribut id pengguna.
+getName() : String
Fungsi untuk mendapatkan nilai atribut nama pengguna.
+getPassword() : String
Fungsi untuk mendapatkan nilai atribut password pengguna.
+getIsAdmin() : Boolean
Fungsi untuk mendapatkan nilai status *rule* dari pengguna.
+getIsApproved() : Boolean
Fungsi untuk mendapatkan nilai status penerimaan pengguna.
+addUser() : String
Fungsi untuk menambah pengguna baru.
-updateApprovedUser() : String
Fungsi untuk mengubah status *Approved* pengguna baru.

2.2.3.17 Spesifikasi Design Kelas saleLocations

User	<<entity>>
-idSale: Integer Atribut untuk menyimpan data id <i>sale</i> . -saleName : String Atribut untuk menyimpan data nama <i>sale</i> . -catagory : String Atribut untuk menyimpan data katagory <i>sale</i> . -price: Integer Atribut untuk menyimpan data harga <i>sale</i> .	

-item : Integer

Atribut untuk menyimpan data jumlah produk/barang pada sale.

-date : Date

Atribut untuk menyimpan data tanggal sale.

-latitude : Double

Atribut untuk menyimpan data latitude sale.

-longtitude : Double

Atribut untuk menyimpan data longtitude sale.

-address : String

Atribut untuk menyimpan data alamat sale.

-picture : Bitmap

Atribut untuk menyimpan data gambar lokasi atau produk sale.

-isApproved : Bollean

Atribut untuk menyimpan data status penerimaan sale.

+saleLocations()

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

+setIdSale()

Prosedur untuk menginisialisasi atribut id sale.

+setSaleName()

Prosedur untuk menginisialisasi atribut nama sale.

+setCatagory()

Prosedur untuk menginisialisasi atribut katagory sale.

+setPrice()

Prosedur untuk menginisialisasi atribut harga sale.

+setItem()

Prosedur untuk menginisialisasi atribut jumlah produk/barang pada sale.

+setDate()

Prosedur untuk menginisialisasi atribut tanggal sale.

+setLatitude()

Prosedur untuk menginisialisasi atribut latitude sale.

+setLongtitude()

Prosedur untuk menginisialisasi atribut longitude sale.
+setAddress()
Prosedur untuk menginisialisasi atribut alamat sale.
+setPicture()
Prosedur untuk menginisialisasi atribut gambar lokasi atau produk sale.
+setIsApproved()
Prosedur untuk menginisialisasi atribut status penerimaan sale.
+getIdSale()() : String
Fungsi untuk mendapatkan nilai atribut id sale.
+getSaleName() : String
Fungsi untuk mendapatkan nilai atribut nama sale.
+getCatagory() : String
Fungsi untuk mendapatkan nilai atribut katagory sale.
+getPrice(): Integer
Fungsi untuk mendapatkan nilai atribut harga sale.
+getItem() : Integer
Fungsi untuk mendapatkan nilai atribut jumlah produk/barang pada sale.
+getDate() : Date
Fungsi untuk mendapatkan nilai atribut tanggal sale.
+getLatitude() : Double
Fungsi untuk mendapatkan nilai atribut latitude sale.
+getLongitude() : Double
Fungsi untuk mendapatkan nilai atribut longitude sale.
+getAddress() : String
Fungsi untuk mendapatkan nilai atribut alamat sale.
+getPicture() : Bitmap
Fungsi untuk mendapatkan nilai atribut gambar lokasi atau produk sale.
+getIsApproved() : Bollean
Fungsi untuk mendapatkan nilai atribut status penerimaan sale.

+getNewSale() : String

Fungsi untuk mendapatkan data *sale* yang status penerimaannya bernilai *false* atau mendapatkan *sale* terbaru.

+getSaleDataDetail() : String

Fungsi untuk mendapatkan data lengkap *sale* dengan id *sale* tertentu.

+search() : String

Fungsi untuk mendapatkan data *sale* berdasarkan inputan pengguna.

+advancedSearch(distance,order) : String

Fungsi untuk mendapatkan data *sale* berdasarkan inputan pengguna dengan jarak dan hasil yang diurutkan sesuai dengan keinginan user.

+getSaleDataNearMe() : String

Fungsi untuk mendapatkan data *sale* terdekat dengan pengguna.

+saveLocation() : String

Fungsi untuk menambah *sale* baru berserta data-datanya.

+UpdateApprovedSale() : String

Fungsi untuk mengubah status *Approved sale location* baru.

+getBulletin(index) : String

Fungsi untuk mendapatkan data *sale* yang paling baru.

3 Perancangan Data

3.1 Dekomposisi Data

3.1.1 Deskripsi Entitas Data USERS

Nama	Tipe	Panjang	Keterangan
ID_USER	Integer	-	ID pengguna, Primary Key
NAME_USER	VarChar	10	Nama pengguna
PASSWORD	VarChar	10	Password pengguna
EMAIL	VarChar	100	Email pengguna
IS_ADMIN	Boolean	1	Role pengguna
IS_APPROVEDUSER	Boolean	1	Status <i>user</i> yang ditambahkan

3.1.2 Deskripsi Entitas Data SALE_LOCATIONS

Nama	Tipe	Panjang	Keterangan
ID_SALE	Integer	-	ID Sale, Primary Key
ID_CATAGORY	Integer	-	ID katagori, Foreign Key
NAME	VarChar	50	Nama produk
ID_USER	Integer	-	ID pengguna, Foreign Key
LONGITUDE	Double	-	Koordinat longitude lokasi
LATITUDE	Double	-	Koordinat latitude lokasi
DISCOUNT	Integer	-	Besaran discount yang diperoleh
DESCRIPTION	VarChar	200	Deskripsi dari <i>sale</i>
ITEMS	Integer	-	Banyaknya barang/produk
PRICE	Integer	-	Harga produk
DATE_SALE	Date	-	Tanggal <i>sale</i> di <i>submit</i>

ADDRESS	VarChar	200	Alamat lokasi sale
PICTURES	VarChar	100	Lokasi photo disimpan
IS_APPROVED	Integer	-	Status lokasi sale yang ditambahkan

3.1.3 Deskripsi Entitas Data COMMENTS

Nama	Tipe	Panjang	Keterangan
ID_COMMENT	Integer	-	ID komentar, Primary Key
ID_USER	Integer	-	ID pengguna, Foreign Key
ID_SALE	Integer	-	ID Sale, Primary Key
COMMENT	VarChar	100	Isi comment
DATE_COMMENT	Date	-	Tanggal comentar masuk

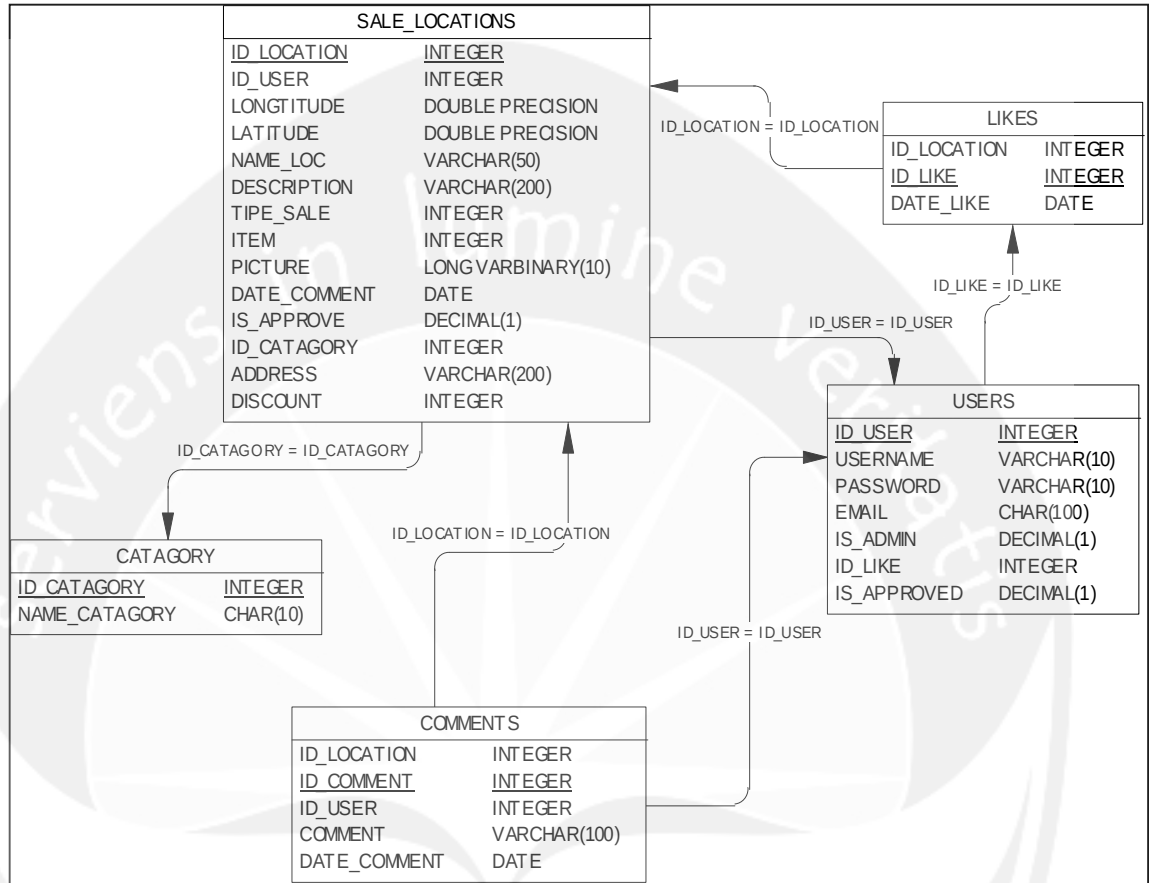
3.1.4 Deskripsi Entitas Data LIKES

Nama	Tipe	Panjang	Keterangan
ID_LIKE	Integer	-	ID like, Primary Key
ID_USER	Integer	-	ID user, Foreign Key
ID_SALE	Integer	-	ID sale , Foreign Key

3.1.5 Deskripsi Entitas Data catagories

Nama	Tipe	Panjang	Keterangan
ID_CATAGORY	Integer	-	ID katagori, Primary Key
NAME_CATAGORY	VarChar	20	ID user, Foreign Key

3.1.6 Physical Data Model

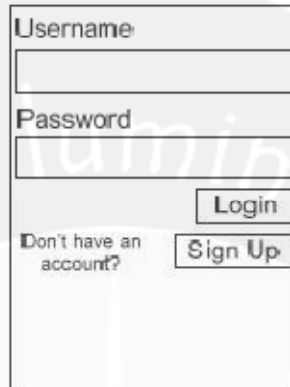


Gambar 3.1 Physical Data Model

4 Perancangan Antarmuka

4.1 Sketsa Antarmuka dan Deskripsinya

4.1.1 Antarmuka *Login*

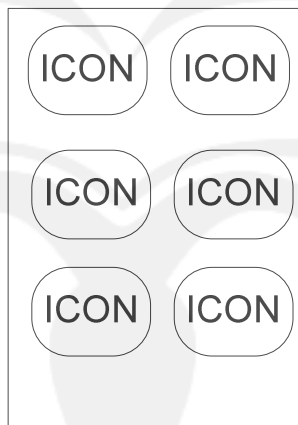


A sketch of a login form. It contains a 'Username' label above a text input field, a 'Password' label above another text input field, a 'Login' button, and a 'Sign Up' button. Below the 'Sign Up' button is the text 'Don't have an account?'.

Gambar 4.1 Rancangan Antarmuka *Login*

Antarmuka Gambar 4.1 digunakan untuk melakukan proses login ke dalam sistem. Setelah sukses proses login maka akan menuju ke tampilan menu.

4.1.2 Antarmuka *Menu*

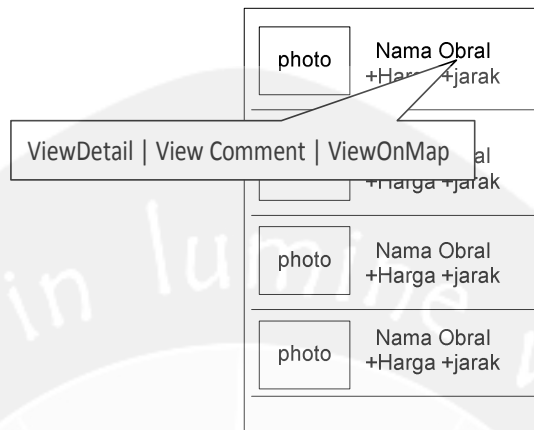


A sketch of a menu screen. It consists of a rectangular container holding six circular icons arranged in a 3x2 grid. Each icon contains the word 'ICON'.

Gambar 4.2 Rancangan Antarmuka *Menu*

Antarmuka Gambar 4.2 merupakan antarmuka utama yang berisi gambar menu. Jika dipilih, tiap gambar akan menuju ke masing-masing antarmuka.

4.1.3 Antarmuka Halaman *Near Me*



Gambar 4.3 Rancangan Antarmuka *Near Me*

Antarmuka Gambar 4.3 merupakan antarmuka yang berisi informasi mengenai lokasi sale terdekat secara rinci. Pada masing-masing label akan berisi informasi nama sale, harga, jarak lokasi sale dengan pengguna berserta photo sale bersangkutan dan jika memilih salah satu sale akan menampilkan menu berisi untuk menampilkan lebih detail sale bersangkutan, komentar dan tampilkan di peta.

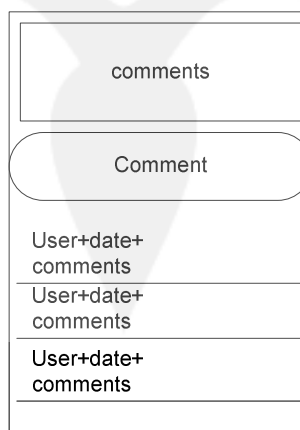
4.1.4 Antarmuka Halaman *View Detail*



Gambar 4.4 Rancangan Antarmuka *View Detail*

Antarmuka Gambar 4.6 merupakan antarmuka yang berisi informasi detail sale yang berupa gambar sale bersangkutan, name sale, katagori sale, harga sale, banyaknya *item*, pengguna yang mengenalkan beserta emailnya dan tanggal sale dikenalkan. Kemudian, terdapat label beserta gambar untuk menampilkan jumlah orang yang menyukai sale tersebut dan jika gambar di tekan akan menandakan kita menyukai sale tersebut.

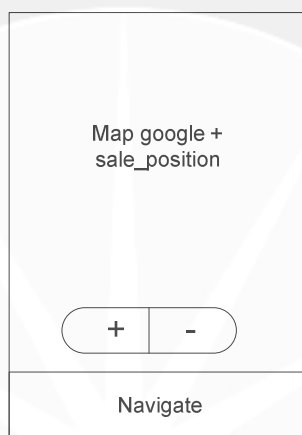
4.1.5 Antarmuka Halaman *View Comments*



Gambar 4.5 Rancangan Antarmuka *View Comments*

Antarmuka Gambar 4.5 merupakan antarmuka yang berisi peta Google. Posisi pengguna akan ditampilkan dalam bentuk titik di tengah peta. Kemudian disediakan menu untuk melihat kategori tempat wisata, menambahkan lokasi wisata, memperbesar peta, dan memperkecil peta.

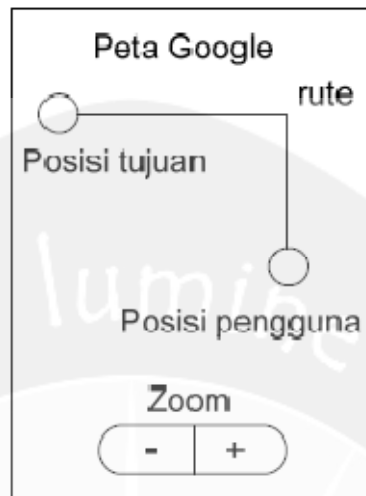
4.1.6 Antarmuka Halaman *View On Map*



Gambar 4.6 Rancangan Antarmuka *View On Map*

Antarmuka Gambar 4.6 merupakan antarmuka yang berisi peta Google. Lokasi *sale* bersangkutan akan ditampilkan dalam bentuk titik di tengah peta. Kemudian disediakan tombol untuk perbesar dan perkecil tampilan peta. Kemudian *button* untuk melakukan navigasi dari posisi pengguna ke lokasi *sale* bersangkutan.

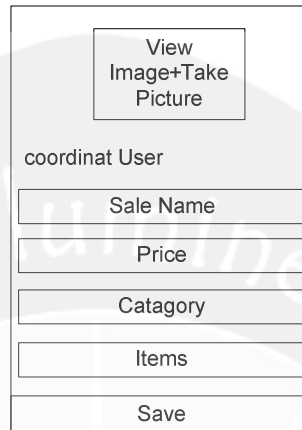
4.1.7 Antarmuka Halaman *View Route*



Gambar 4.7 Rancangan Antarmuka Tampilan Navigasi

Antarmuka Gambar 4.7 merupakan antarmuka yang berisi navigasi rute dari posisi pengguna sampai dengan posisi lokasi yang dipilih.

4.1.8 Antarmuka Halaman *Save Sale Location*



View Image+Take Picture
koordinat User
Sale Name
Price
Catagory
Items
Save

Gambar 4.8 Rancangan Antarmuka *Save Sale Location*

Antarmuka Gambar 4.12 merupakan antarmuka yang berisi kolom pengisian data lokasi *sale* yang akan didaftarkan pengguna ke basis data. Masing-masing terdapat kolom *editText* yang harus diisi oleh pengguna jika ingin mendaftarkan lokasi tersebut. Kemudian terdapat image view untuk menampilkan gambar *sale* dan jika disentuh akan memberi akses kepada user untuk menggunakan kamera untuk mengambil gambar produk atau lokasi *sale*. Jika sudah terisi semua pengguna akan memilih tombol *Save* yang terletak di paling bawah dari antarmuka. Antarmuka ini dilengkapi dengan *scroll* karena masukan data akan banyak dan cukup memakai banyak ruang. Untuk itu antarmuka ini dilengkapi *scroll* secara vertikal sehingga pengguna bisa memasukkan banyak data.

4.1.9 Antarmuka Halaman *View Neweset*



Gambar 4.9 Rancangan Antarmuka *View Neweset*

Antarmuka Gambar 4.9 merupakan antarmuka yang berisi gambar sale kemudian informasi sale terbaru berupa nama sale, harga, pengguna aplikasi yang mengenalkannya, dan jarak dengan pengguna aplikasi dengan lokasi sale. Kemudian terdapat 2 *button* untuk menampilkan informasi sale yang terbaru dan sebelumnya.

4.1.10 Antarmuka Halaman *Search*

Name	
Catagory:	▼
< Price	
1 Km	▼
Order by:	▼
Search	

Gambar 4.10 Rancangan Antarmuka *Search*

Antarmuka pada gambar 4.10 digunakan untuk mencari lokasi *sale* terdekat dari posisi user dengan menentukan pilihan-pilihan yang telah disediakan oleh aplikasi. Pilihan-pilihan tersebut antara lain adalah name *sale*, kategori *Isale*, harga, dan jarak lokasi untuk menentukan jarak dari user. Kemudian pilihan diurutkan berdasarkan apa. Jika pilihan *search place* telah dipilih maka tekan tombol *Search* dan sistem akan menampilkan lokasi-lokasi yang ada.

4.1.11 Antarmuka Halaman *View My Position*



Gambar 4.11 Rancangan Antarmuka *View My Position*

Antarmuka Gambar 4.10 merupakan antarmuka yang berisi peta Google. Posisi pengguna akan ditampilkan dalam bentuk titik di tengah peta. Kemudian disediakan tombol untuk perbesar dan perkecil tampilan peta. Untuk peta Google digunakan dalam beberapa menu, yaitu menu untuk melihat peta atau menu untuk mendaftarkan lokasi. Untuk menu mendaftarkan lokasi *sale* nantinya akan disediakan dua tombol menu tambahan yaitu *save* dan *back*. Jika memilih tombol *save* maka akan menuju ke form penyimpanan lokasi, jika memilih tombol *back* akan kembali ke menu awal.

SKPL

SPEKIFIKASI KEBUTUHAN PERANGKAT LUNAK

SENDER (Sale Finder)

Untuk :
Universitas Atma Jaya Yogyakarta

Dipersiapkan oleh:
Jimmy Samuel Pardede / 07 07 05432

Program Studi Teknik Informatika - Fakultas Teknologi
Industri

Universitas Atma Jaya Yogyakarta

	Program Studi Teknik Informatika Fakultas Teknologi Industri	Nomor Dokumen		Halaman
		SKPL - SENDER		1/32
		Revisi		

Program Studi Teknik Informatika	SKPL – SENDER	1/ 32
Dokumen ini dan informasi yang dimilikinya adalah milik Program Studi Teknik Informatika-UAJY dan bersifat rahasia. Dilarang untuk me-reproduksi dokumen ini tanpa diketahui oleh Program Studi Teknik Informatika		

DAFTAR PERUBAHAN

Revisi	Deskripsi
A	
B	
C	
D	
E	
F	

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

Daftar Halaman Perubahan

Halaman	Revisi	Halaman	Revisi
			

Daftar Isi

1	Pendahuluan	6
1.1	Tujuan	6
1.2	Lingkup Masalah	6
1.3	Definisi, Akronim dan Singkatan	7
1.4	Referensi	8
1.5	Deskripsi umum (Overview)	8
2	Deskripsi Kebutuhan	9
2.1	Perspektif produk	9
2.2	Fungsi Produk	10
2.3	Karakteristik Pengguna	12
2.4	Batasan-batasan	13
2.5	Asumsi dan Ketergantungan	13
3	Kebutuhan khusus	13
3.1	Kebutuhan antarmuka eksternal	13
3.2	Kebutuhan fungsionalitas Perangkat Lunak.	15
4	Spesifikasi Rinci Kebutuhan	16
4.1	Spesifikasi Kebutuhan Fungsionalitas	16
5	Entity Relationship Diagram (ERD)	32

Daftar Gambar

1. Arsitektur Perangkat Lunak SENDER	10
2. Use Case Diagram	15
3. ERD	32



1 Pendahuluan

1.1 Tujuan

Dokumen Spesifikasi Kebutuhan Perangkat Lunak (SKPL) ini merupakan dokumen spesifikasi kebutuhan perangkat lunak SENDER (*Sale Finder*) untuk mendefinisikan kebutuhan perangkat lunak yang meliputi antarmuka *external* (antarmuka antara system dengan perangkat lunak, perangkat keras, dan pengguna), dan atribut (*feature-feature* tambahan yang dimiliki sistem), serta mendefinisikan fungsi perangkat lunak. SKPL-SENDER ini juga mendefinisikan batasan-batasan dari perancangan perangkat lunak.

1.2 Lingkup Masalah

Adapun batasan sistem adalah sesuai dengan lingkup tujuan pembuatan perangkat lunak SENDER yaitu:

1. Pencarian posisi *device* pada saat itu (*real time*).
2. Menangani pencarian lokasi obral terdekat dari posisi pengguna.
3. Menangani penyimpanan data lokasi obral.
4. Menangani informasi seputar obral yang ditampilkan.
5. Menangani navigasi rute perjalanan dari posisi pengguna sekarang hingga sampai pada lokasi obral.
6. Menangani pemberian komentar dan penandaan suka pada suatu obral.

1.3 Definisi, Akronim dan Singkatan

Daftar definisi akronim dan singkatan :

Keyword/Phrase	Definisi
SKPL	Merupakan spesifikasi kebutuhan dari perangkat lunak yang akan dikembangkan.
SKPL-SENDER	Kode yang merepresentasikan kebutuhan pada SENDER (SALE FINDER)
SENDER	Aplikasi <i>mobile</i> untuk mencari lokasi-lokasi obral terdekat dari <i>user</i>
Android	Sebuah sistem operasi yang dirancang oleh google untuk perangkat <i>mobile</i> .
Internet	Internet merupakan istilah umum yang dipakai untuk menunjuk <i>Network</i> global yang terdiri dari komputer dan layanan servis dengan sekitar 30 sampai 50 juta pemakai komputer dan puluhan layanan informasi termasuk e-mail, FTP, dan World Wide Web.
GPS	<i>Global Positioning System</i> (SENDERS) merupakan perangkat untuk mengetahui posisi koordinat di permukaan bumi
aGPS	<i>Asisted Global Positioning System</i> (aGPS) suatu terobosan baru SENDERS di masa sekarang ini dengan ukuran relatif kecil
Google Map API	Suatu <i>library</i> yang berbentuk <i>JavaScript</i>

1.4 Referensi

Referensi yang digunakan pada perangkat lunak tersebut adalah:

1. Jimmy Samauel P., *Spesifikasi Kebutuhan Perangkat Lunak WHISPER*, Universitas Atma Jaya Yogyakarta, 2009.
2. Yogi Santoso, *Spesifikasi Kebutuhan Perangkat Lunak GP (Golek Pangan)*, Universitas Atma Jaya Yogyakarta, 2011.

1.5 Deskripsi umum (Overview)

Secara umum dokumen SKPL ini terbagi atas 5 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 SENDER 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 SENDER tersebut.

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

Bagian keempat berisi penjelasan tentang spesifikasi rinci kebutuhan dari tiap use case yang ada pada use case diagram pada bagian ketiga.

Bagian kelima berisi gambaran ERD (Entity Relationship Diagram) yang digunakan dalam mengembangkan perangkat lunak SENDER.

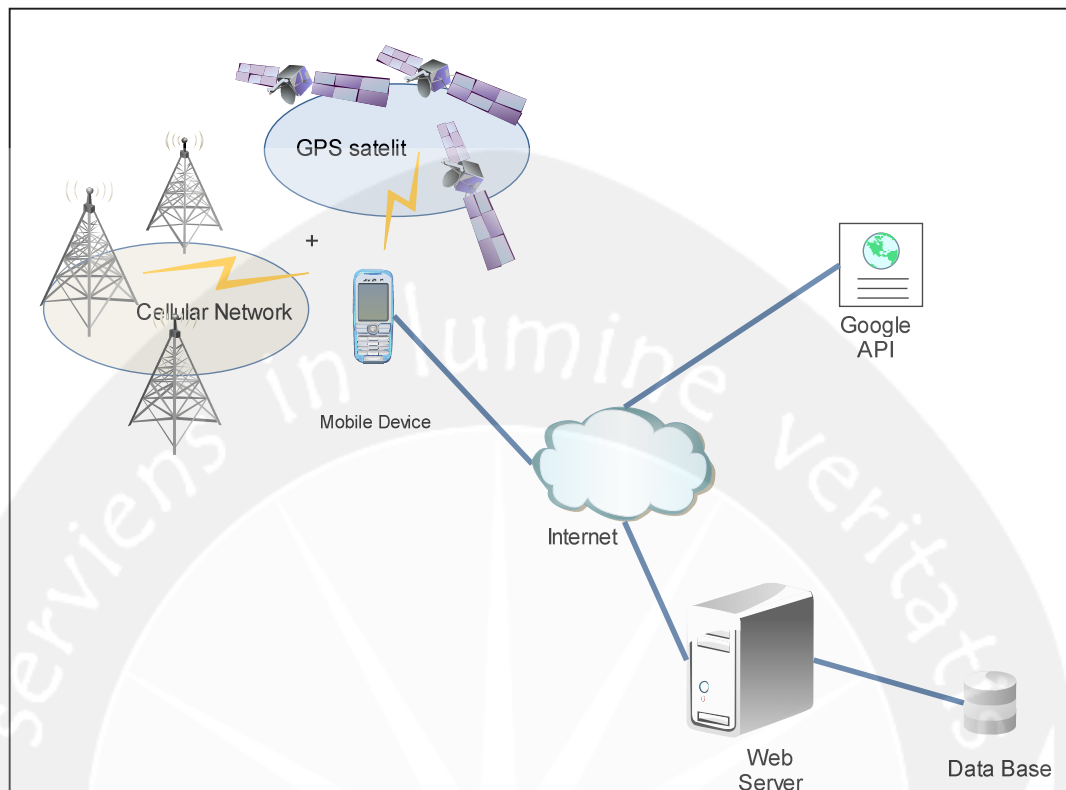
2 Deskripsi Kebutuhan

2.1 Perspektif produk

Aplikasi *Sale Finder* (SENDER) merupakan perangkat lunak atau aplikasi yang dikembangkan untuk mengetahui lokasi tempat obral. Aplikasi ini akan memberikan informasi mengenai posisi *real device* atau handphone orang yang bersangkutan, kemudian dengan mengetahui posisi pengguna dapat menyimpan atau mencari posisi *sale* terdekat dari pengguna. Posisi perangkat *mobile* dan *sale* ini akan ditampilkan pada peta yang berbasis web yang dibangun menggunakan *Google Maps API* (*Application Programming Interface*).

Aplikasi *Sale Finder* ini berjalan pada platform Android untuk perangkat *mobile* dan dibuat menggunakan bahasa pemrograman Java dengan lingkungan emulator menggunakan Eclipse.

Pengguna akan berinteraksi dengan sistem melalui antarmuka GUI (Graphical User Interface) pada perangkat *mobile*. Pada sistem ini, seperti terlihat pada gambar 1, aplikasi ini bersifat *client-server*, yaitu pengguna mengakses data yang terdapat pada *web server* dan masukan data disimpan dalam database.



Gambar 1. Arsitektur Perangkat Lunak SENDER

2.2 Fungsi Produk

Adapun fungsionalitas yang terdapat pada SENDER ini adalah sebagai berikut:

1. Fungsi *Login* (SKPL-SENDER-001)

Merupakan fungsi yang digunakan agar dapat masuk ke dalam sistem dalam hal ini adalah pengguna layanan SENDER.

2. Fungsi *Sign Up* (SKPL-SENDER-002)

Merupakan fungsi yang digunakan untuk melakukan pendaftaran agar dapat menggunakan aplikasi SENDER.

3. Fungsi *Find Sale* (**SKPL-SENDER-003**)

Merupakan fungsi yang digunakan untuk menampilkan lokasi obral yang terdekat dari pengguna aplikasi pada saat itu.

4. Fungsi *Submit Sale* (**SKPL-SENDER-004**)

Merupakan fungsi yang digunakan untuk melakukan pendaftaran lokasi obral baru oleh pengguna aplikasi SENDER.

5. Fungsi *Search Sale* (**SKPL-SENDER-005**)

Merupakan fungsi yang digunakan untuk membantu pengguna layanan mencari lokasi obral yang sesuai dengan masukkan pengguna dan pilihan untuk melakukan filter.

6. Fungsi *View Details* (**SKPL-SENDER-006**)

Merupakan fungsi yang digunakan untuk menampilkan detail informasi dari obral bersangkutan.

7. Fungsi *View On Map* (**SKPL-SENDER-007**)

Merupakan fungsi yang digunakan untuk membantu pengguna dalam menampilkan lokasi obral dan pengguna dalam bentuk map.

8. Fungsi *Submit Comment* (**SKPL-SENDER-008**)

Merupakan fungsi yang digunakan oleh pengguna layanan untuk memberikan komentar pada suatu lokasi obral.

9. Fungsi *Give Like* (**SKPL-SENDER-009**)

Merupakan fungsi yang digunakan oleh pengguna untuk menandakan pengguna menyukai obral tersebut.

10. Fungsi *Show Route* (**SKPL-SENDER-010**)

Merupakan fungsi yang digunakan untuk memberikan rute perjalanan dari lokasi pengguna berada menuju lokasi obral.

11. Fungsi *Approve Sale* (**SKPL-SENDER-011**)

Merupakan fungsi yang digunakan untuk menyetujui penambahan sale (obral) yang baru dan pengguna SENDER baru.

12. Fungsi *Approve User* (**SKPL-SENDER-012**)

Merupakan fungsi yang digunakan untuk menyetujui penambahan obral yang baru dan pengguna SENDER baru.

13. Fungsi *Get Newest Sale* (**SKPL-SENDER-013**)

Merupakan fungsi yang digunakan untuk menampilkan informasi berupa obral terbaru.

14. Fungsi *Take Picture* (**SKPL-SENDER-014**)

Merupakan fungsi yang digunakan untuk mengambil gambar obral yang akan di submit.

2.3 Karakteristik Pengguna

Pengguna perangkat lunak SENDER terdiri dari :

1. User

Karakteristik User sebagai berikut :

- a. Mampu mengoperasikan Handphone dengan sistem operasi Android.

2. Administrator

Karakteristik Administrator sebagai berikut :

- a. Memahami pengoperasian handphone dengan sistem operasi Android.

2.4 Batasan-batasan

Batasan-batasan dalam pengembangan perangkat lunak SENDER tersebut adalah :

1. Kebijakan Umum
Berpedoman pada tujuan dari pengembangan perangkat lunak SENDER.
2. Keterbatasan Perangkat Lunak
Dapat diketahui kemudian setelah sistem ini berjalan (sesuai dengan kebutuhan).

2.5 Asumsi dan Ketergantungan

Sistem ini dapat dijalankan pada perangkat mobile (*handphone*) yang menggunakan sistem operasi Android.

3 Kebutuhan khusus

3.1 Kebutuhan antarmuka eksternal

Kebutuhan antar muka eksternal pada perangkat lunak SENDER meliputi kebutuhan antarmuka pemakai, antarmuka perangkat mobile, dan antarmuka perangkat lunak.

3.1.1 Antarmuka pemakai

Pengguna berinteraksi dengan antarmuka yang ditampilkan dalam bentuk form-form yang ada pada layar *handphone*.

3.1.2 Antarmuka perangkat keras

Antarmuka perangkat keras yang digunakan dalam perangkat lunak SENDER adalah:

1. Perangkat mobile berbasis Android
2. Perangkat mobile mempunyai model input berupa layar sentuh.
3. Perangkat mobile mempunyai aGPS.

3.1.3 Antarmuka perangkat lunak

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

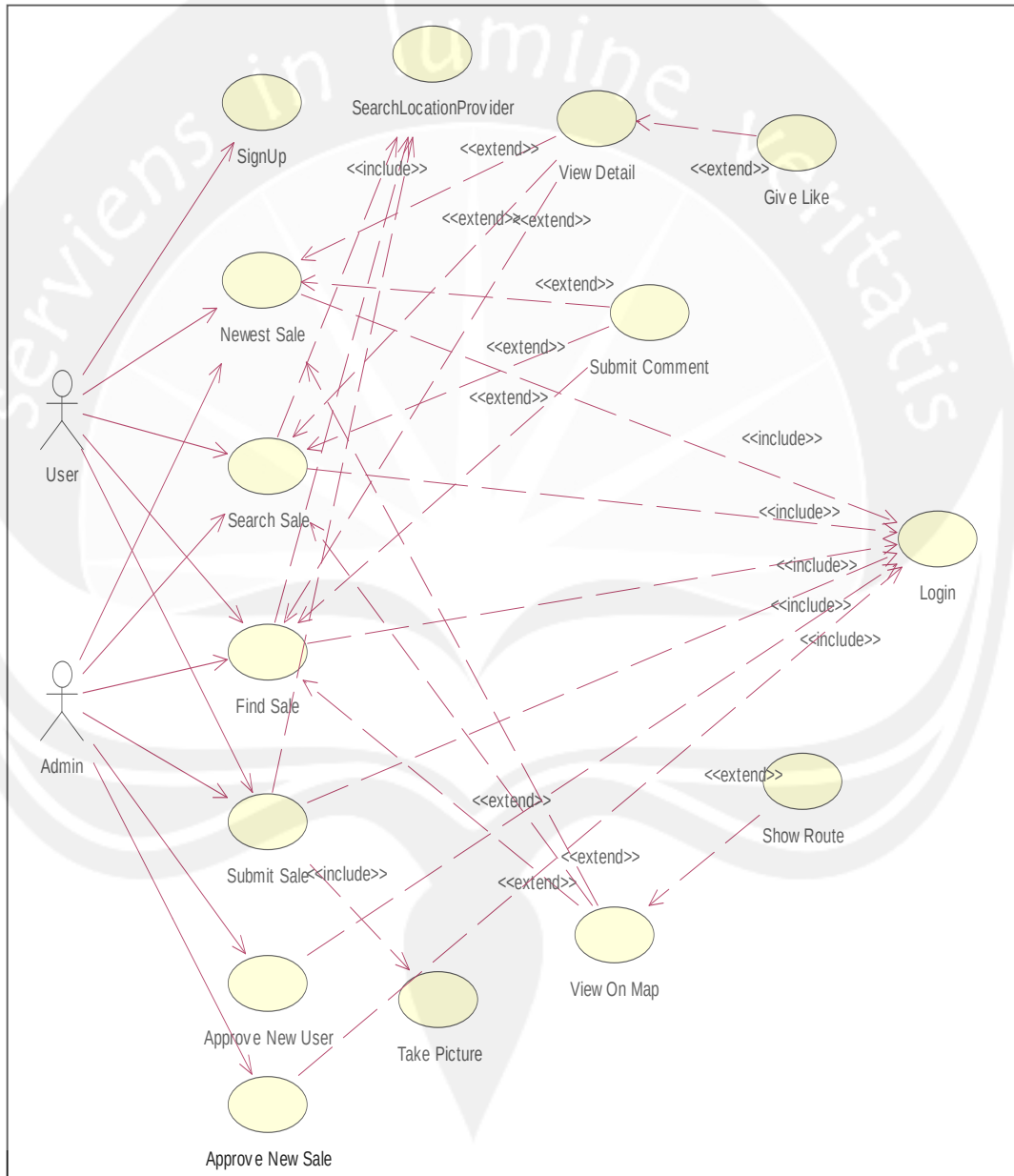
1. Nama : Android OS
Sumber : Android Software.
Sebagai sistem operasi untuk perangkat *mobile*.
2. Nama : Mysql
Sumber : Mysql
Sebagai *Data Base Management System* (DBMS).
3. Nama : Google API V.8
Sumber : Google
Sebagai API dalam pembangunan aplikasi peta.
4. Nama : Apache
Sumber : Apache
Sebagai web server local
5. Nama : PHP
Sumber : PHP
Sebagai pencari lokasi di database pada sisi server.

3.1.4 Antarmuka Komunikasi

Antarmuka komunikasi perangkat lunak SENDER menggunakan protokol http.

3.2 Kebutuhan fungsionalitas Perangkat Lunak.

3.2.1 Use Case Diagram



Gambar 2. Use Case Diagram

4 Spesifikasi Rinci Kebutuhan

4.1 Spesifikasi Kebutuhan Fungsionalitas

4.1.1 Use case Spesification : *Login*

1. Brief Description

Use Case ini digunakan oleh user untuk memperoleh akses ke sistem. *Login* didasarkan pada sebuah id unik yaitu username dan password dari user.

2. Primary Actor

1. Anggota
2. Administrator

3. Supporting Actor

none

4. Basic Flow

1. Use Case ini dimulai ketika aktor memilih untuk melakukan *login*.
2. Sistem menampilkan antarmuka untuk *login*.
3. Aktor memasukkan username dan password.
4. Sistem memeriksa username dan password yang diinputkan aktor.
E-1 Password atau username user tidak sesuai.
5. Sistem memberikan akses ke aktor.
6. Use Case ini selesai.

5. Alternative Flow

none.

6. Error Flow

E-1 Password atau username tidak sesuai

1. Sistem menampilkan peringatan bahwa username atau password tidak sesuai.
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 statusnya.

4.1.2 Use case Spesification : Sign Up

1. Brief Description

Use Case ini memungkinkan user untuk melakukan pendaftaran.

2. Primary Actor

1. Calon Anggota.

3. Supporting Actor

none

4. Basic Flow

1. Use Case ini dimulai ketika aktor masuk ke dalam aplikasi pada *mobile device* untuk pertama kali dan melakukan pendaftaran menjadi member.
2. Sistem menampilkan *form entry* data user.
3. Aktor memasukkan data user dan memilih rule member.
4. Aktor meminta sistem untuk menyimpan data user.
5. Sistem mengecek data user yang telah diinputkan.
E-1 Data user yang diinputkan salah.
6. Sistem menyimpan data data user ke database.
7. Use Case selesai.

5. Alternative Flow

None.

6. Error Flow

E-1 Data user yang diinputkan aktor salah.

1. Sistem memberikan pesan peringatan bahwa data yang diinputkan salah.
2. Kembali ke Basic Flow Langkah ke 3.

7. PreConditions

1. Aktor pertama kali memasuki sistem.

8. PostConditions

1. Aktor telah memasukkan data user yang diperlukan dan data user tersimpan dalam database.

4.1.3 Use case Spesification : *Find Sale*

1. Brief Description

Use Case ini digunakan untuk menampilkan lokasi-lokasi obral terdekat dari pengguna aplikasi.

2. Primary Actor

1. Anggota
2. Administrator

3. Supporting Actor

None.

4. Basic Flow

1. Use Case ini dimulai ketika aktor memilih untuk mencari lokasi obral terdekat (*Find Sale*).
2. Sistem menampilkan lokasi-lokasi obral terdekat dari pengguna.

E-1 Lokasi obral terdekat dari user tidak ada

3. User memilih salah satu lokasi obral terdekat.

4. Sistem menampilkan informasi detail obral dan menyediakan akses untuk melihat komentar, tampilan posisi obral di peta serta fitur untuk rute ke obral tersebut.

5. Use Case ini selesai

5. Alternative Flow

none

6. Error Flow

E-1 obral terdekat dari user tidak ada

1. Sistem menampilkan peringatan bahwa tidak ada lokasi obral terdekat dari user.
2. Kembali ke Basic Flow langkah ke 2.

7. PreConditions

none

8. PostConditions

1. Informasi detail obral ditampilkan.

4.1.4 Use case Spesification : *Submit Sale*

1. Brief Description

Use Case ini digunakan untuk menambah lokasi obral baru.

2. Primary Actor

1. Anggota
2. Administrator

3. Supporting Actor

none

4. Basic Flow

1. Use Case ini dimulai ketika user memilih untuk menambah lokasi obral baru.
2. Sistem menampilkan antarmuka *submit location*.
3. User memasukkan data lokasi obral baru.

4. User meminta sistem untuk menyimpan data yang telah dimasukkan.

<<Take Picture>>

E-1 Data yang dimasukkan salah.

5. Sistem menyimpan data lokasi obral.
6. Use Case selesai.

5. Alternative Flow

none

6. Error Flow

E-1 Data yang dimasukkan salah

1. Sistem memberikan pesan peringatan bahwa data yang dimasukkan salah.
2. Kembali ke Basic Flow Langkah ke 3.

7. PreConditions

none

8. PostConditions

1. Pengguna telah menambah lokasi obral.

4.1.5 Use case Spesification : Search Sale

1. Brief Description

Use Case ini digunakan untuk mencari lokasi obral dengan cara melakukan *filter* terhadap pencarian lokasi obral.

2. Primary Actor

1. Anggota
2. Administrator

3. Supporting Actor

None

4. Basic Flow

1. Use Case ini dimulai ketika aktor memilih untuk mencari lokasi obral (*Search Location*).

2. Sistem menampilkan antarmuka dari *search*.
3. User memasukan nama obral dan dapat melakukan filter dengan mengaktifkan fitur *advance search* terhadap informasi obral yang akan ditampilkan.
4. Sistem menampilkan informasi obral berdasarkan masukan user.
E-1 Lokasi obral yang dicari user tidak ada.
5. User memilih pilihan lokasi obral yang dimunculkan oleh system.
6. Sistem menampilkan informasi detail obral dan menyediakan akses untuk melihat komentar, tampilan posisi obral di peta serta fitur untuk rute ke obral tersebut.
7. Use Case ini selesai.

5. Alternative Flow

none

6. Error Flow

E-1 Lokasi obral yang dicari user tidak ada

1. Sistem menampilkan peringatan bahwa tidak ada lokasi obral yang dimaksud oleh user
2. Kembali ke Basic Flow langkah ke 3.

7. PreConditions

none

8. PostConditions

1. Informasi detail obral ditampilkan.

4.1.6 Use case Spesification : *View Details*

1. Brief Description

Use Case ini digunakan untuk menampilkan informasi detail obral.

Program Studi Teknik Informatika	SKPL – SENDER	21/ 32
Dokumen ini dan informasi yang dimilikinya adalah milik Program Studi Teknik Informatika-UAJY dan bersifat rahasia. Dilarang untuk me-reproduksi dokumen ini tanpa diketahui oleh Program Studi Teknik Informatika		

2. Primary Actor

1. Anggota
2. Administrator

3. Supporting Actor

None.

4. Basic Flow

1. Use Case ini dimulai ketika user memilih obral dan ingin mengetahui informasi lebih obral bersangkutan.
2. Sistem menampilkan detail informasi bersangkutan dan menyediakan tombol untuk menyukai obral bersangkutan.
3. Use Case selesai.

5. Alternative Flow

None.

6. Error Flow

None.

7. PreConditions

None.

8. PostConditions

1. Informasi detail obral telah ditampilkan

4.1.7 Use case Spesification : View On Map

1. Brief Description

Use Case ini untuk menampilkan lokasi user dan lokasi obral pada peta yang ditampilkan pada *mobile device*.

2. Primary Actor

1. Anggota
2. Administrator

3. Supporting Actor

Program Studi Teknik Informatika	SKPL – SENDER	22/ 32
Dokumen ini dan informasi yang dimilikinya adalah milik Program Studi Teknik Informatika-UAJY dan bersifat rahasia. Dilarang untuk me-reproduksi dokumen ini tanpa diketahui oleh Program Studi Teknik Informatika		

None.

4. Basic Flow

1. Use Case ini dimulai ketika user memilih untuk menampilkan posisi obral di peta pada *mobile device*.
2. Sistem menampilkan lokasi user dan sejumlah lokasi obral pada peta berupa komponen *Icon* dan informasi singkat obral yang dipilih.
3. Use Case selesai

5. Alternative Flow

None.

6. Error Flow

None.

7. PreConditions

None.

8. PostConditions

1. Gambar peta dan lokasi user beserta obral ditampilkan di layar.

4.1.8 Use case Spesification : *Submit Comment*

1. Brief Description

Use Case ini digunakan oleh user untuk memberikan komentar terhadap suatu lokasi obral.

2. Primary Actor

1. Anggota
2. Administrator

3. Supporting Actor

none

4. Basic Flow

1. Use Case ini dimulai ketika aktor memilih komentar dari obral bersangkutan atau ingin

Program Studi Teknik Informatika	SKPL – SENDER	23/ 32
Dokumen ini dan informasi yang dimilikinya adalah milik Program Studi Teknik Informatika-UAJY dan bersifat rahasia. Dilarang untuk me-reproduksi dokumen ini tanpa diketahui oleh Program Studi Teknik Informatika		

menambah komentar terhadap obral bersangkutan (*Submit Comment*).

2. Sistem menampilkan komentar-komentar yang ada berkaitan dengan obral bersangkutan dan menyediakan field untuk menambah komentar.
3. User memasukkan komentar ke dalam *field* yang sudah disediakan.
4. Sistem menyimpan komentar yang dimasukkan user.
5. Sistem menampilkan komentar-komentar yang ada berkaitan dengan obral bersangkutan dan menyediakan field untuk menambah komentar.
6. Use Case ini selesai.

5. Alternative Flow

None.

6. Error Flow

None.

7. PreConditions

None.

8. PostConditions

1. Komentar baru telah disimpan dan ditampilkan.

4.1.9 Use case Spesification : Give Like

1. Brief Description

Use Case ini digunakan oleh user untuk memberikan informasi bahwa aktor menyukai terhadap suatu lokasi obral.

2. Primary Actor

1. Anggota
2. Administrator

3. Supporting Actor

Program Studi Teknik Informatika	SKPL – SENDER	24/ 32
Dokumen ini dan informasi yang dimilikinya adalah milik Program Studi Teknik Informatika-UAJY dan bersifat rahasia. Dilarang untuk me-reproduksi dokumen ini tanpa diketahui oleh Program Studi Teknik Informatika		

none

4. Basic Flow

1. Use Case ini dimulai ketika aktor memilih informasi detail obral.
2. Sistem menampilkan informasi detail obral dan *button* yang akan menandakan menyukai obral tersebut beserta jumlah orang yang menyukai.
A-1 Sistem menampilkan bahwa aktor telah melakukan " Like " terhadap obral bersangkutan
3. User menekan *button* " Like " untuk detail informasi obral yang ditampilkan.
4. Sistem menyimpan id pengguna yang menyukai obral tersebut.
5. Sistem menampilkan jumlah orang yang menyukai obral tersebut dan menampilkan bahwa user menyukai obral bersangkutan .
6. Use Case ini selesai

5. Alternative Flow

1. Kembali ke Basic Flow langkah ke 5.

6. Error Flow

None.

7. PreConditions

1. Use case *View Details* telah dilakukan.

8. PostConditions

1. Jumlah like untuk obral tersebut bertambah.

4.1.10 Use case Spesification : *Show Route*

1. Brief Description

Use Case ini digunakan untuk menampilkan rute jalan dari aktor ke lokasi obral bersangkutan.

Program Studi Teknik Informatika	SKPL – SENDER	25/ 32
Dokumen ini dan informasi yang dimilikinya adalah milik Program Studi Teknik Informatika-UAJY dan bersifat rahasia. Dilarang untuk me-reproduksi dokumen ini tanpa diketahui oleh Program Studi Teknik Informatika		

2. Primary Actor

1. Anggota
2. Administrator

3. Supporting Actor

None.

4. Basic Flow

1. Use Case ini dimulai ketika user memilih untuk menampilkan rute jalan (*show route*) .
2. Sistem menampilkan icon yang menggambarkan posisi user dan icon lokasi obral beserta rute jalan yang dapat ditempuh oleh user.
3. Use Case selesai.

5. Alternative Flow

None.

6. Error Flow

None.

7. PreConditions

1. Use Case View On Map telah dilakukan

8. PostConditions

1. Rute jalan telah ditampilkan

4.1.11 Use case Spesification : Approve Sale

1. Brief Description

Use Case ini digunakan untuk menyetujui lokasi obral yang baru.

2. Primary Actor

1. Administrator

3. Supporting Actor

None.

4. Basic Flow

1. Use Case ini dimulai ketika user memilih untuk menyetujui lokasi obral baru.
2. Sistem menampilkan obral-obral baru yang dimasukkan oleh user yang belum disetujui.
E-1 Tidak ada lokasi obral baru yang ditambah
3. Aktor memilih obral baru yang akan disetujui.
A-1 Aktor memilih obral baru yang akan tidak disetujui.
4. Sistem menyimpan data baru ke dalam database.
5. Use Case selesai.

5. Alternative Flow

A-1 Aktor tidak menyetujui obral

1. Sistem menghapus obral baru tersebut dalam database.
2. Kembali ke Basic Flow Langkah ke 5.

6. Error Flow

E-1 Tidak ada lokasi obral baru yang ditambah

1. Sistem memberikan pesan peringatan bahwa tidak ada lokasi obral baru.
2. Kembali ke Basic Flow Langkah ke 5.

7. PreConditions

None.

8. PostConditions

1. Lokasi obral yang telah di setujui tampil dan bisa diakses oleh user selain adminstrator.

4.1.12 Use case Spesification : Approve User

1. Brief Description

1. Use Case ini digunakan untuk menyetujui lokasi obral yang baru.

2. Primary Actor

Program Studi Teknik Informatika	SKPL – SENDER	27/ 32
Dokumen ini dan informasi yang dimilikinya adalah milik Program Studi Teknik Informatika-UAJY dan bersifat rahasia. Dilarang untuk me-reproduksi dokumen ini tanpa diketahui oleh Program Studi Teknik Informatika		

2. Administrator

3. Supporting Actor

3. None.

4. Basic Flow

4. Use Case ini dimulai ketika user memilih untuk menyetujui calon pengguna SENDER baru.
5. Sistem menampilkan calon-calon pengguna SENDER yang baru yang telah melakukan pendaftaran untuk menjadi pengguna aplikasi SENDER.
6. E-1 Tidak ada lokasi obral baru yang ditambah
7. Aktor memilih obral baru yang akan disetujui.
8. A-1 Aktor memilih calon pengguna yang akan tidak disetujui.
9. Sistem menyimpan data baru ke dalam database.
10. Use Case selesai.

5. Alternative Flow

1. A-1 Aktor tidak menyetujui calon pengguna Sistem menghapus calon pengguna tersebut dalam database.
2. Kembali ke Basic Flow Langkah ke 5.

6. Error Flow

1. E-1 Tidak ada calon pengguna baru yang ditambah
2. Sistem memberikan pesan peringatan bahwa tidak ada calon pengguna baru.
3. Kembali ke Basic Flow Langkah ke 5.

7. PreConditions

1. None.

8. PostConditions

1. Calon pengguna yang telah di setuju tampil dan bisa diakses oleh user selain administrator.

4.1.13 Use case Spesification : View Newest Sale

1. Brief Description

Use Case ini digunakan untuk menampilkan obral terbaru.

2. Primary Actor

1. User
2. Administrator

3. Supporting Actor

none

4. Basic Flow

1. Use Case ini dimulai ketika aktor ingin memilih menu untuk menampilkan informasi obral terbaru.
2. Sistem mengambil data informasi obral yang paling baru kemudian menampilkannya.
3. User ingin menampilkan informasi obral yang agak lama

A-1 User ingin menampilkan informasi obral yang lama dari informasi obral yang sedang tampil.

A-2 User memilih menampilkan komentar dari informasi obral yang sedang tampil.

4. Sistem menampilkan informasi obral lebih lama dari sebelumnya.

5. Use Case ini selesai.

5. Alternative Flow

A-1 Menampilkan informasi obral yang lebih baru.

1. Sistem mengambil data informasi obral yang lebih baru dari yang ditampilkan.

2. Sistem menampilkan informasi obral lebih baru dari sebelumnya.

3. Kembali ke Basic Flow langkah ke 5.

A-2 Menampilkan informasi detail informasi obral terbaru.

1. Use case View Details dijalankan.

A-3 Menampilkan komentar-komentar dari informasi obral terbaru.

1. Use case View Details dijalankan.

6. Error Flow

none

7. PreConditions

none

8. PostConditions

1. Informasi obral terbaru berhasil ditampilkan.

4.1.14 Use case Spesification : Take Picture

1. Brief Description

Use Case ini digunakan untuk mengambil gambar yang digunakan sebagai informasi yang menarik terhadap obral yang akan di *submit*.

2. Primary Actor

1. User

2. Administrator

Program Studi Teknik Informatika	SKPL – SENDER	30/ 32
Dokumen ini dan informasi yang dimilikinya adalah milik Program Studi Teknik Informatika-UAJY dan bersifat rahasia. Dilarang untuk me-reproduksi dokumen ini tanpa diketahui oleh Program Studi Teknik Informatika		

3. Supporting Actor

None.

4. Basic Flow

1. Use Case ini dimulai pada use case *submit sale* user memilih untuk mengambil gambar.
2. Sistem memberi akses aplikasi kamera yang tersedia pada *handphone* kepada user.
3. User mengambil gambar dengan menggunakan aplikasi yang diberikan sistem.
4. Sistem menampilkan gambar yang diambil oleh aktor dan menyimpan sementara.
5. Use Case ini selesai.

5. Alternative Flow

6. Error Flow

none

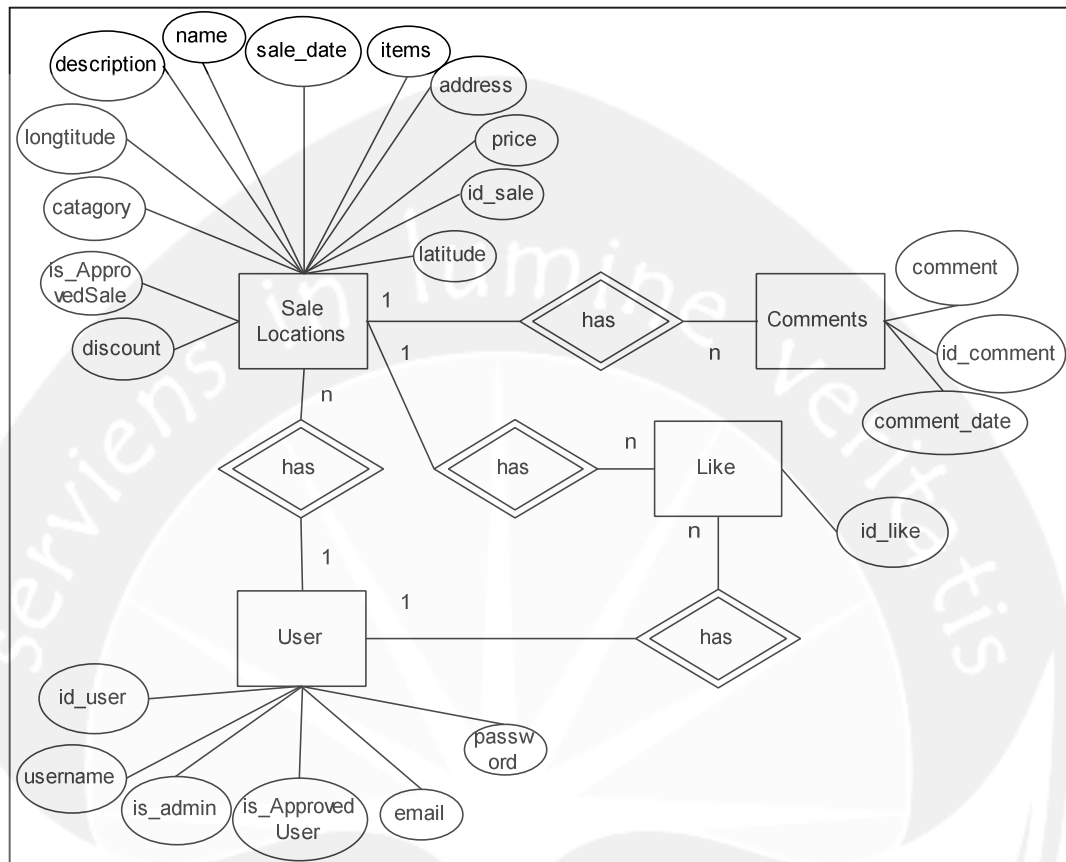
7. PreConditions

none

8. PostConditions

1. Aktor dapat melihat gambar yang diambilnya.

5 Entity Relationship Diagram (ERD)



Gambar 3. ERD