

BAB II

LANDASAN TEORI

2.1 Sistem

Definisi sistem dapat dibagi menjadi dua pendekatan, yaitu pendekatan secara prosedur dan pendekatan secara komponen. Berdasarkan pendekatan prosedur, sistem didefinisikan sebagai kumpulan dari beberapa prosedur yang mempunyai tujuan tertentu. Sedangkan berdasarkan pendekatan komponen, system merupakan kumpulan dari komponen-komponen yang saling berkaitan mencapai tujuan tertentu. Dalam perkembangan sistem yang ada, sistem dibedakan menjadi dua jenis, yaitu sistem terbuka dan sistem tertutup. Sistem terbuka merupakan sistem yang dihubungkan dengan arus sumber daya luar dan tidak mempunyai elemen pengendali. Sedangkan sistem tertutup tidak mempunyai elemen pengontrol dan dihubungkan pada lingkungan sekitarnya. (Indrakarna, Sutanto, Taufik, 2012)

2.2 Android

Android adalah sistem operasi yang berbasis Linux untuk telepon seluler seperti telepon pintar dan komputer tablet. Android menyediakan platform terbuka para pengembang untuk menciptakan aplikasi mereka sendiri untuk digunakan oleh bermacam piranti bergerak. Android Incorporation didirikan di Palo Alto, California, Amerika Serikat pada bulan Oktober, 2003 oleh Andy Rubin: co-pendiri Danger (Bahaya Incorporation adalah sebuah perusahaan eksklusif di platform, software, desain, dan layanan untuk perangkat komputasi mobile), Kaya Miner: co-pendiri Kebakaran liar Communications, Incorporation, Nick Sears: sekali VP di TMobile, dan Chris White: desain kepala dan pengembangan antarmuka di TV Web. (Anwar, Jaya, Kusuma, 2014)

2.3 Android Studio

Android Studio adalah Lingkungan Pengembangan Terpadu - *Integrated Development Environment* (IDE) untuk pengembangan aplikasi Android, berdasarkan IntelliJ IDEA. Selain merupakan editor kode IntelliJ dan alat pengembang yang berdaya guna, Android Studio menawarkan fitur lebih banyak untuk meningkatkan produktivitas saat membuat aplikasi Android. (

Android Developers, 2017) misalnya :

- 1) Sistem versi berbasis Gradle yang fleksibel
- 2) Emulator yang cepat dan kaya fitur
- 3) Lingkungan yang menyatu untuk pengembangan bagi semua perangkat Android
- 4) Instant Run untuk mendorong perubahan ke aplikasi yang berjalan tanpa membuat APK baru
- 5) Template kode dan integrasi GitHub untuk membuat fitur aplikasi yang sama dan mengimpor kode contoh
- 6) Alat pengujian dan kerangka kerja yang ekstensif
- 7) Alat Lint untuk meningkatkan kinerja, kegunaan, kompatibilitas versi, dan masalah-masalah lain
- 8) Dukungan C++ dan NDK
- 9) Dukungan bawaan untuk Google Cloud Platform, mempermudah pengintegrasian Google Cloud Messaging dan App Engine

2.4 Global Positioning System (GPS)

Global Positioning System atau GPS merupakan sistem navigasi yang berbasis satelit yang menyediakan informasi koordinat posisi, kecepatan dan waktu kepada pengguna di seluruh dunia. GPS dikembangkan oleh Departemen Pertahanan Amerika Serikat (US DoD = *United States Department of Defense*).

GPS memungkinkan kita mengetahui posisi geografis (lintang, bujur, dan ketinggian di atas permukaan laut). Jadi, dimanapun kita berada di muka bumi ini kita dapat mengetahui posisi kita dengan tepat. (Sendow, Longdong, 2012)

2.5 Google Maps

Menurut (Muslim, Sunyoto, 2012), *Google Maps* adalah layanan oleh Google yang menawarkan teknologi pemetaan yang *userfriendly*. Layanan ini dapat diakses melalui situs <http://maps.google.com> atau <http://maps.google.co.id> untuk *Google Maps* Indonesia. Pada situs ini kita dapat melihat informasi geografis pada hampir semua wilayah di bumi. Fitur *Google maps* menawarkan beberapa fitur antara lain:

- a) Hasil pencarian bisnis terpadu Menemukan lokasi bisnis dan kontak informasi semua dalam satu lokasi, terintegrasi dengan peta.
- b) Peta yang dapat digeser Untuk melihat lokasi yang berada disekitar peta yang ditampilkan, kita hanya perlu menggeser peta tersebut.
- c) Citra satelit Melihat citra satelit dari lokasi yang Anda inginkan dan dapat diperbesar.
- d) *Earth view* Klik tombol Earth untuk melihat citra 3D dan medan (terrain) dari *Google Earth* pada peta yang dapat Anda perbesar maupun atur kemiringannya.
- e) *Street view* Melihat dan bernavigasi pada berbagai level jalan.
- f) Petunjuk arah yang rinci Anda tinggal memasukkan alamat dan biarkan *Google Maps* menunjukkan lokasi dan arah mengemudi untuk Anda.

2.6 Google Maps API

Google API bisa dikatakan bagian dari *Framework Google* menyediakan berbagai API (*Application Programming Interface*) yang sangat berguna bagi pengembang web maupun aplikasi desktop untuk memanfaatkan berbagai fitur yang disediakan oleh Google seperti misalnya: *AdSense*, *Search Engine*, *Translation* maupun *YouTube*. API secara sederhana bisa diartikan sebagai kode program yang merupakan antarmuka atau penghubung antara aplikasi atau web yang kita buat dengan fungsi-fungsi yang dikerjakan. (Turmuzi, Wibowo, Chafid, 2015)

2.7 Web Service

Web service adalah layanan yang diidentifikasi dengan URI (*Uniform Resource Identifier*) yang mengekspos fiturnya melalui internet menggunakan protokol dan bahasa standar internet serta dapat diimplementasikan menggunakan standar internet seperti XML (*Extensible Markup Language*). Sampai dengan saat ini teknologi *web service* terus berkembang. Salah satu teknologi yang populer saat ini adalah REST (*Representational State Transfer*) atau terkadang disebut dengan RESTful. Beberapa contoh RESTful *web service* adalah *Amazon's Simple Storage Service (S3)*, *Atom Publishing Protocol*, dan *Google Maps*. Pada prinsipnya *request* ke suatu RESTful *web service* sebenarnya adalah suatu HTTP Request. (Surendra, 2014)

2.8 PHP

Bahasa pemrograman PHP merupakan bahasa pemrograman untuk membuat web yang bersifat *server-side scripting*. PHP memungkinkan kita untuk membuat halaman web yang bersifat dinamis. PHP dapat dijalankan pada berbagai macam *Operating System (OS)*, misalnya *Windows, Linux dan Mac OS*. Selain *Apache*, PHP juga mendukung beberapa *web server* lain, misalnya *Microsoft IIS, Caudium, PWS* dan lain-lain. Seperti pernah disinggung sebelumnya bahwa PHP dapat memanfaatkan *database* untuk menghasilkan halaman web yang dinamis.(Adriyani, 2016)

2.9 Web Server

Menurut (Novianta, Setyaningsih, 2015). Web server merupakan *software* yang memberikan layanan data, berfungsi menerima permintaan HTTP atau HTTPS dari *client* yang dikenal dengan *browser web* dan mengirimkan kembali hasilnya dalam bentuk halaman-halaman web yang umumnya berbentuk dokumen HTML.

Konsep *web server* antara lain:

- a) *Web server* merupakan mesin aplikasi atau software yang beroperasi dalam mendistribusikan *web page* ke user, tentu saja sesuai dengan permintaan *user*.
- b) Hubungan antara *web server* dan *browser* internet merupakan gabungan atau jaringan komputer yang berada diseluruh dunia. Setelah terhubung secara fisik, protocol TCP/IP (*networking protocol*) yang memungkinkan semua komputer dapat berkomunikasi antar satu dengan lainnya. Pada saat aplikasi *browser* meminta data web page ke server maka instruksi permintaan data oleh *browser* tersebut dikemas dalam TCP yang merupakan *protocol transport* dan dikirim ke alamat yang merupakan *protocol* berikutnya yaitu *hyper text transfer protocol (HTTP)*. Data yang diparsing dari browser ke *web server* disebut sebagai HTTP request yang meminta halaman web dan kemudian *web server* akan mencari data HTML yang dibutuhkan dan dikemas dalam TCP protocol kemudian dikirim kembali ke *browser*. Data yang dikirim dari *server* ke *browser* disebut sebagai HTTP *response*. Jika data yang diminta oleh *browser* tidak

ditemukan pada *web server* maka akan menampilkan *error* pada *web page* yaitu *Error: 404 Page Not Found*.

2.10 MYSQL

MySQL adalah suatu perangkat lunak *database* relasi (*Relational Database Management System* atau DBMS), seperti halnya ORACLE, POSTGRESQL, MSSQL, dan sebagainya. SQL merupakan singkatan dari *Structure Query Language*, didefinisikan sebagai suatu sintaks perintah-perintah tertentu atau bahasa program yang digunakan untuk mengelola suatu database. Jadi MySQL adalah *software*nya dan SQL adalah bahasa perintahnya.(Anisya, 2013)

2.11 Location Based Service

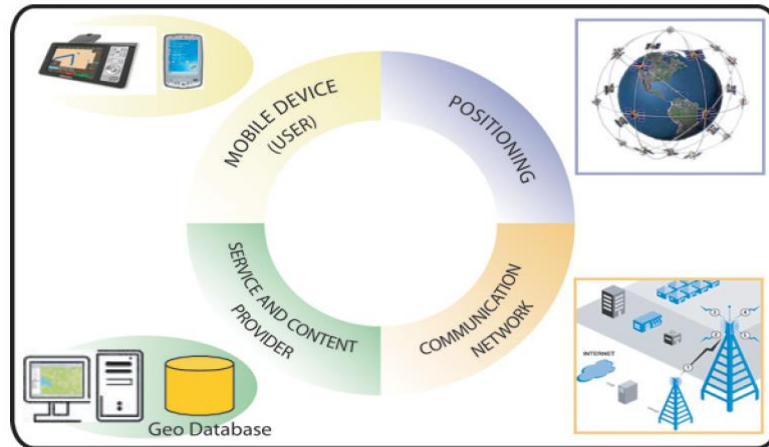
LBS merupakan layanan yang mengidentifikasi lokasi atau objek, seperti keberadaan seseorang, lokasi bank, rumah sakit atau sekolah terdekat. Dengan teknologi LBS (*Location Based Service*) yang dapat digunakan untuk mengetahui posisi berdasarkan titik geografis dari lokasi pengguna dan lokasi lain yang diinginkan.(Hati, Suprayogi, Sasmito, 2013)

2.11.1 Komponen LBS

Dalam Layanan Berbasis Lokasi terdapat Lima komponen penting yaitu meliputi:

- 1) *Mobile Devices*: Suatu alat yang digunakan oleh pengguna untuk meminta informasi yang dibutuhkan. Informasi dapat diberikan dalam bentuk suara, gambar, dan text.
- 2) *Communication Network*: Komponen kedua adalah jaringan komunikasi yang mengirim data pengguna dan informasi yang diminta dari mobile terminal ke *Service Provider* kemudian mengirimkan kembali informasi yang diminta ke pengguna. *Communication network* dapat berupa jaringan seluler (GSM, CDMA), Wireless Local Area Network (WLAN), atau Wireless Wide Area Network (WWAN)
- 3) *Positioning Component*: Untuk memproses suatu layanan maka posisi pengguna harus diketahui
- 4) *Service and Application Provider*: Penyedia layanan menawarkan berbagai macam layanan kepada pengguna dan bertanggung jawab untuk memproses informasi yang diminta oleh pengguna.

- 5) *Data and Content Provider*: Penyedia layanan tidak selalu menyimpan semua data yang dibutuhkan yang bisa diakses oleh pengguna. Untuk itu, data dapat diminta dari data and *content provider*.



Gambar 2.1 Komponen Dasar LBS

Selanjutnya *Service and Application Provider* mengirim informasi yang telah diolah melalui jaringan internet dan jaringan komunikasi. Pada akhirnya pengguna dapat menerima informasi yang diinginkan.

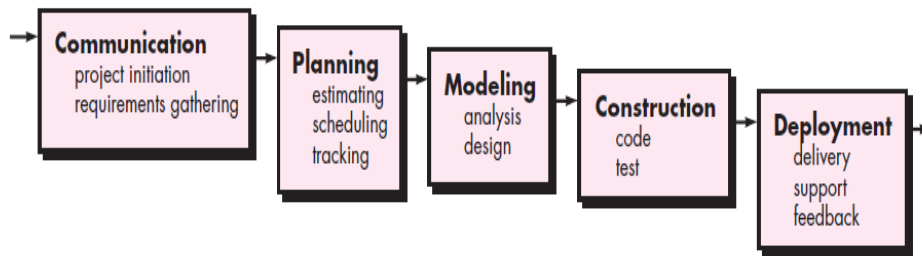
2.12 Software Development LifeCycle (SDLC)

Pada awal pengembangan perangkat lunak, para pembuat program (*programmer*) langsung melakukan pengodean perangkat lunak tanpa menggunakan prosedur atau tahapan pengembangan perangkat lunak. Dan ditemuilah kendala-kendala seiring dengan perkembangan skala sistem-sistem perangkat yang semakin besar. SDLC (*Software Development LifeCycle*) adalah proses mengembangkan atau mengubah suatu sistem perangkat lunak dengan menggunakan model-model dan metodologi yang digunakan orang untuk mengembangkan sistem-sistem perangkat lunak sebelumnya berdasarkan *best practice* atau cara-cara yang sudah teruji sebelumnya. (Rosa A.S, Salahudin, 2016)

2.12.1 Model Waterfall

Menurut Presman (Dalam Rohayati, 2014) Model proses yang digunakan dalam pembangunan perangkat lunak menggunakan paradigma secara *waterfall* adalah model klasik yang bersifat sistematis, berurutan dalam membangun sebuah

software. Berikut ini adalah gambaran dari waterfall yang meliputi beberapa proses, yaitu:



Gambar 2.2 Skema Model *Waterfall* (Presman,2010)

Tahap dari skema model *waterfall* adalah :

- 1) *Communication* Langkah ini merupakan analisis terhadap kebutuhan software, dan tahap untuk mengadakan pengumpulan data dengan melakukan pertemuan dengan customer, maupun mengumpulkan data-data tambahan baik yang ada di jurnal, artikel, maupun dari internet.
- 2) *Planning* Proses *planning* merupakan lanjutan dari proses *communication* (analysis requirement). Tahapan ini akan menghasilkan dokumen user requirement atau bisa dikatakan sebagai data yang berhubungan dengan keinginan user dalam pembuatan software, termasuk rencana yang akan dilakukan.
- 3) *Modeling* Proses *modeling* ini akan menerjemahkan syarat kebutuhan ke sebuah perancangan software yang dapat diperkirakan sebelum dibuat coding. Proses ini berfokus pada rancangan struktur data, arsitektur software, representasi interface, dan detail (algoritma) prosedural. Tahapan ini akan menghasilkan dokumen yang disebut *software requirement*.
- 4) *Construction* merupakan proses membuat kode. Coding atau pengkodean merupakan penerjemahan desain dalam bahasa yang bisa dikenali oleh komputer. Programmer akan menerjemahkan transaksi yang diminta oleh user. Tahapan inilah yang merupakan tahapan secara nyata dalam mengerjakan suatu *software*, artinya penggunaan komputer akan dimaksimalkan dalam tahapan ini. Setelah pengkodean selesai maka akan

menemukan kesalahan-kesalahan terhadap sistem untuk kemudian bisa diperbaiki.

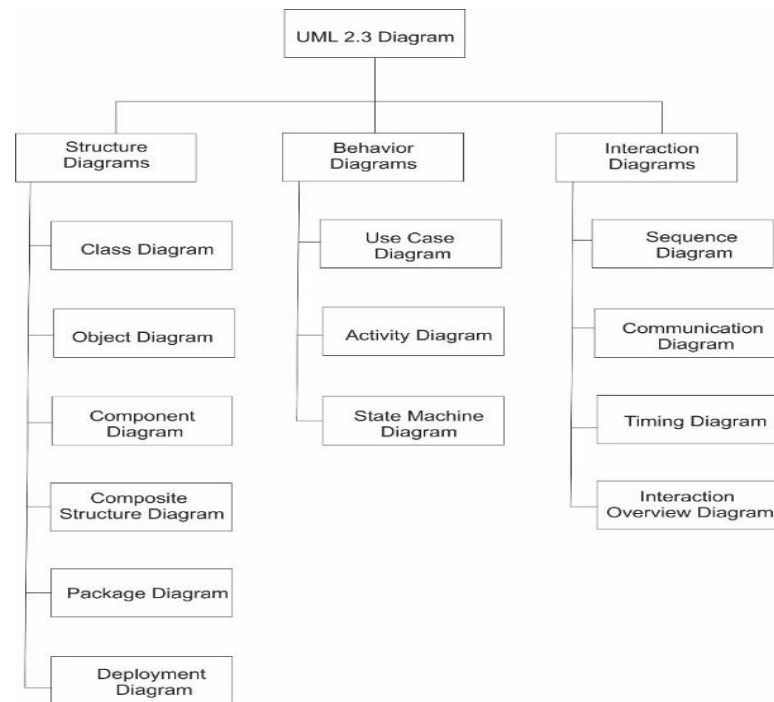
- 5) *Deployment* Tahapan ini bisa dikatakan final dalam pembuatan sebuah *software* atau sistem. Setelah melakukan analisis, desain dan pengkodean maka sistem yang sudah jadi akan digunakan oleh user. Kemudian *software* yang telah dibuat harus dilakukan pemeliharaan secara berkala

2.13 Unified Modelling Language (UML)

UML (*Unified Modeling Language*) adalah salah satu standar bahasa yang banyak di gunakan di dunia industry untuk mendefinisikan requirement, membuat analisis dan desain, serta menggambarkan arsitektur dalam pemrograman berorientasi objek. UML muncul karena adanya kebutuhan pemodelan visual untuk menspesifikasikan, menggambarkan, membangun, dan mendokumentasikan dari system perangkat lunak. UML merupakan bahasa visual untuk pemodelan dan komunikasi mengenai sebuah system dengan menggunakan diagram dan teks-teks pendukung. UML hanya berfungsi untuk melakukan pemodelan. Jadi pengguna UML tidak terbatas pada metodologi tertentu, meskipun pada kenyataanya UML paling banyak digunakan pada metodologi berorientasi objek. (Rosa A.S, Salahudin, 2016)

2.13.1 Diagram UML

Pada UML versi terbaru yaitu UML 2.3 terdiri dari 13 macam diagram yang di kelompokkan dalam 3 katagori. Pembagian katagori dan macam-macam diagram tersebut dapat di lihat pada gambar di bawah ini. (Rosa A.S, Saahudin, 2016) :



Gambar 2.3 Diagram UML (Rosa A.S, Salahudin, 2016)

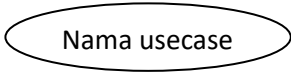
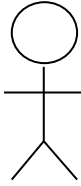

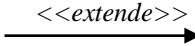
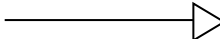
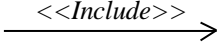
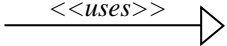
Berikut ini penjelasan singkat dari pembagian katagori tersebut.

- 1) *Structure diagrams* yaitu kumpulan diagram yang di gunakan untuk menggambarkan suatu struktur statis dari sistem yang di modelkan
- 2) *Behavior diagrams* yaitu kumpulan diagram yang di gunakan untuk menggambarkan kelakuan sistem atau rangkaian perubahan yang terjadi pada sebuah sistem
- 3) *Instruction diagrams* yaitu kumpulan diagram yang di gunakan untuk menggambarkan interaksi sistem dengan sistem lain maupun interaksi antar subsistem pada suatu sistem.

2.13.2 Use case Diagram

Use case atau diagram merupakan pemodelan untuk kelakuan sistem informasi yang akan dibuat. Use case mendeskripsikan sebuah interaksi antara satu atau lebih aktor dengan sistem informasi yang akan dibuat. Use case di gunakan untuk mengetahui fungsi apa saja yang ada di dalam sebuah sistem informasi dan siapa saja yang berhak menggunakan fungsi-fungsi itu. Simbol-simbol yang di gunakan untuk pembuatan use case diagram dapat di lihat pada tabel di bawah ini. (Rosa A.S, Salahudin, 2016) :

Tabel 2.1 Simbol use case diagram (Rosa A.S, Salahudin, 2016)


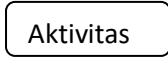


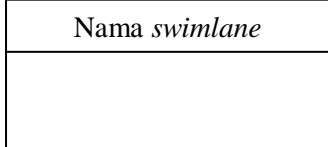
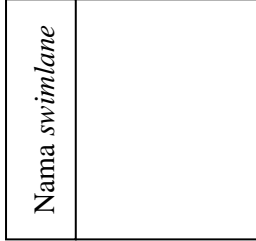
Simbol	Deskripsi
<p><i>Use case</i></p>  <p>Nama usecase</p>	<p>Fungsionalitas yang disediakan sistem sebagai unit-unit yang saling bertukar pesan antara unit atau aktor; biasanya dinyatakan dengan menggunakan kata kerja di awal frase nama <i>use case</i>.</p>
<p>Aktor / <i>actor</i></p>  <p>Nama aktor</p>	<p>Orang, prose, atau sistem lain yang berinteraksi dengan sistem informasi yang akan di buat di luar sistem informasi yang akan di buat itu sendiri, jadi walaupun simbol dari aktor adalah gambar orang, biasanya dinyatakan menggunakan kata benda di awal frase nama aktor.</p>
<p>Asosiasi / <i>Association</i></p> 	<p>Komunikasi antara aktor dan <i>use case</i> yang berpartisipasi pada <i>use case</i> atau <i>use case</i> memiliki interaksi dengan aktor.</p>
<p>Ekstensi / <i>Extend</i></p> 	<p>Relasi use case tambahan ke sebuah <i>use case</i> dimana <i>use case</i> yang di tambahkan dapat berdiri sendiri walau tanpa use case tamahan itu.</p>
<p>Generalisasi / <i>generalization</i></p> 	<p>Hubungan generalisasi dan spasialisasi (umum – khusus) antara dua buah <i>use case</i> dimana fungsi yang satu adalah fungsi yang lebih umum dari lainnya.</p>
<p>Menggunakan / <i>include / uses</i></p>  	<p>Relasi use case tambahan ke sebuah <i>use case</i> di mana use case yang di tambahkan memerlukan <i>use case</i> ini untuk menjalankan fungsi atau sebagai syarat dijalankan <i>use case</i> ini.</p>

2.13.3 Activity Diagram

Diagram Aktivitas atau activity diagram menggambarkan aliran kerja atau aktivitas dari sebuah system atau menu yang ada pada perangkat lunak. Yang perlu di perhatikan adalah bahwa diagram aktivitas menggambarkan aktivitas

system bukan aktor. Simbol-simbol yang di gunakan untuk pembuatan use case diagram dapat di lihat pada tabel di bawah ini. (Rosa A.S Salaudin, 2016) :

Tabel 2.2 Simbol activity diagram (Rosa A.S, Salahudin, 2016)

Simbol	Deskripsi
Status awal 	Status awal aktivitas sistem, sebuah diagram aktifitas memiliki sebuah status awal
Aktivitas 	Aktivitas yang di lakukan sistem, aktivias biasanya diawali dengan kata kerja.
Percabangan / <i>decision</i> 	Asosiasi percabangan dimana jika ada pilihan aktivitas lebih dari satu.
Status akhir 	Status akhir yang di lakukan sistem, sebuah diagram aktivitas memiliki sebuah status akhir.
<i>Swimlane</i>  Atau 	Memisahkan organisasi bisnis yang bertanggung jawab terhadap aktivitas yang terjadi.

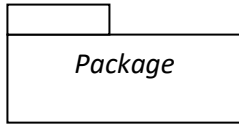
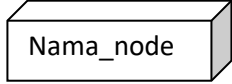
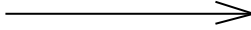

2.13.4 Deployment Diagram

Deployment diagram menunjukkan konfigurasi komponen dalam proses eksekusi aplikasi. Diagram deployment juga dapat digunakan untuk memodelkan hal-hal berikut. (Rosa A.S, Salahudin, 2016) :

- 1) Sistem tambahan yang menggambarkan rancangan *device*, *node*, dan *hardware*
- 2) Sistem *client / server*
- 3) Sistem terdistribusi murni
- 4) Rekayasa ulang aplikasi

Berikut adalah simbol-simbol yang di gunakan pada diagram *deployment* :

Tabel 2.3 Simbol *deployment* diagram (Rosa A.S, Salahudin, 2016)

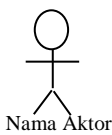
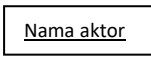
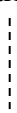

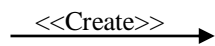
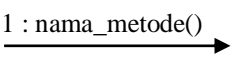
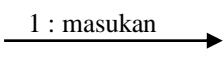
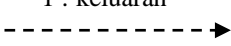
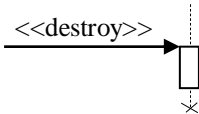
Simbol	Deskripsi
<p><i>Package</i></p> 	<p><i>Package</i> merupakan sebuah bungkusuan dari satu atau lebh <i>node</i>.</p>
<p><i>Node</i></p> 	<p>Biasanya mengacu pada perangkat keras (<i>hardware</i>), perangkat lunak yang tidak di buat sendiri (<i>software</i>).</p>
<p>Keergantungan / <i>dependency</i></p> 	<p>Kebergantungan antar <i>node</i>, arah panah mengarah pada <i>node</i> yang di pakai.</p>
<p><i>Link</i></p> 	<p>Relasi antar <i>node</i></p>

2.13.5 Sequence Diagram

Diagram sekuen menggambarkan kelakuan objek pada use case dengan mendeskripsikan waktu hidup objek dan message yang dikirimkan dan diterima antar objek. Banyaknya diagram sekuen yang harus digambar adalah minimal sebanyak pendefinisian use case yang memiliki proses sendiri atau yang penting semua use case yang telah didefinisikan interaksi jalannya pesan sudah dicakup

pada diagram sekuen. Sehingga, semakin banyak use case yang didefinisikan diagram sekuen yang harus dibuat juga semakin banyak. Berikut adalah simbol-simbol yang ada pada diagram sekuen. (Rosa A.S, Salahudin 2016) :

Tabel 2.4 Simbol *sequence* diagram (Rosa A.S, Salahudin, 2016)

Simbol	Deskripsi
<p>Aktor</p>  <p>Nama Aktor</p> <p>Atau</p>  <p>Nama aktor</p> <p>Tanpa waktu aktif</p>	<p>Orang, prose, atau sistem lain yang berinteraksi dengan sistem informasi yang akan di buat d luar sistem informasi yang akan di buat itu sendiri, jadi walaupun simbol dari aktor adalah gambar orang, biasanya dinyatakan menggunakan kata benda di awal frase nama aktor.</p>
<p>Garis hidup / <i>lifeline</i></p> 	<p>Menyatakan kehidupan suatu objek</p>
<p>Objek</p> 	<p>Menyatakan objek dalam keadaan aktif dan berinteraksi, semua yang terhubung dengan waktu aktif ini adalah sebuah tahapan yang dilakukan di dalamnya.</p>
<p>Pesan tipe create</p> 	<p>Menyatakan suatu objek membuat objek yang lain, arah panah mengarah pada objek yang di buat.</p>
<p>Pesan tipe call</p> 	<p>Menyatakan suatu objek memanggil operasi/metode yang ada pada objek lain atau dirinya sendiri.</p>
<p>Pesan tipe send</p> 	<p>Menyatakan bahwa suatu objek mengirim data/masukan/informasi ke objek lain, arah panah mengarah pada objek yang di kirim.</p>
<p>Pesan tipe return</p> 	<p>Menyatakan bahwa suatu objek yang telah menjalankan suatu oprasi atau metode menghasilkan suatu kembalian ke objek tertentu, arah panah mengarah pada objek yang menerima kembalian.</p>
<p>Pesan tipe destroy</p> 	<p>Menyatakan suatu objek mengakhiri hidup objek yang lain, arah panah mengarah pada objek yang diakhiri, sebaiknya jika ada create maka ada destroy</p>

2.14 Pengujian Perangkat Lunak

Pengujian perangkat lunak adalah satu set aktivitas yang di rancang untuk menguji atau mengavaluasi kebenaran yang di inginkan, pengujian perangkat lunak memiliki cakupan luas dan sering di kaitkan dengan verifikasi dan validasi. Verifikasi mengacu pada sekumpulan aktifitas yang menjamin bahwa perangkat lunak mengimplementasikan dengan benar sebuah fungsi yang spesifik. Validasi mengacu pada sekumpulan aktivitas yang di bangun dapat di telusuri sesuai dengan kebutuhan pelanggan (*Customer*). Pengujian validasi memiliki beberapa pendekatan sebagai berikut (Rosa A.S, Salahudin, 2016) :

1. *Black-Box Testing* (pengujian kotak hitam)

Black box testing adalah menguji perangkat lunak dari segi spesifikasi fungsional tanpa menguji desain dan kode program. Pengujian dimaksudkan untuk mengetahui apakah fungsi-fungsi, masukan, dan keluaran dari perangkat lunak sesuai dengan spesifikasi yang dibutuhkan. Pengujian kotak hitam dilakukan dengan membuat kasus uji yang bersifat mencoba semua fungsi dengan memakai perangkat lunak apakah sesuai dengan spesifikasi yang dibutuhkan. Kasus uji yang dibuat untuk melakukan pengujian black box testing harus dibuat dengan kasus benar dan kasus salah.

2. *White-Box Testing* (pengujian kotak putih)

ini adalah pengujian perangkat lunak dari segi design kode program apakah mampu menghasilkan fungsi-fungsi, masukan dan keluaran yang sesuai dengan spesifikasi kebutuhan. Pengujian kotak putih di lakukan dengan memeriksa logik dari kode program.

2.15 Penelitian terdahulu

No	Nama peeliti	Judul penelitian	Hasil penlitian
1	Daniarta, 2013	Pembuatan Aplikasi Pencarian Dan Pelacakan Lokasi “My Relation Loc” Berbasis Blackberry Webworks Aplication	pada penelitian ini peneliti membuat sebuah aplikasi pelacakan teman yang mengutamakan prvasi penggunaanya, aplikasi ini memiliki beberapa fitur dalam pelacakan yaitu untuk melihat lokasi privat teman di butuhkan sebuah password yang di sebut dengan lockkey, adanya fitur block/unblock teman, fitur tracking dan chackin pada lokasi-lokasi tertentu yang di inginkan pengguna. Aplikasi yang di buat di bagi menjadi dua bagian yaitu aplikasi pada blackberry dan website sebagai web server.
Perbedaan : Pada penelitian yang di lakukan oleh Daniarta aplkasi yang di buat berbasis Backbary sedangkan penulis membuat aplikasi berbasis android.			
2	Purnama, 2009	Pemanfaatan Global Positioning System Untuk Pelacakan Objek Bergerak	Pada penelitian ini peneliti berhasil membuat sebuah sistem pelacakan objek benda bergerak dengan menggunakan perangkat GPS CT-58 yang di pasang pada benda bergerak dan dapat di moitoring menggunakan web.
Perbedaan : Paada Penelitian yang di lakukan oleh Purnama, sistem yang di buat menggunakan perangkat GPS dan berbasis web sedangkan penulis memanfaatkan			

gps yang ada pada smartphone dan berbasis android.			
3	Indrakarna, Sutanto Dan Taufik, 2012	Rancang Bangun Sistem Informasi Pelacakan Dan Pemantauan Paket Kiriman Berbasis Web Dengan Bantuan Mobile Android	Pada penelitian ini peneliti berhasil membangun web untuk melihat informasi posisi kendaraan pengiriman paket dengan bantuan smartphone android yang memiliki fitur GPS, hasil dari penelitian ini adalah menampilkan posisi dari kendaraan pengiriman paket yang sedang aktif saat itu dalam bentuk titik-titik merah yang ada pada lokasi di peta, dan histori lokasi kendaraan dapat di lihat dan di tampilkan dalam bentuk PDF.
Perbedaan : Paada Penelitian yang di lakukan oleh Indrakarna, Sutanto Dan Taufik sistem yang di bangun berbasis web sedangkan penulis membuat sistem berbasis aplikasi android dan memiliki fitur notifikasi.			