BAB II

LANDASAN TEORI

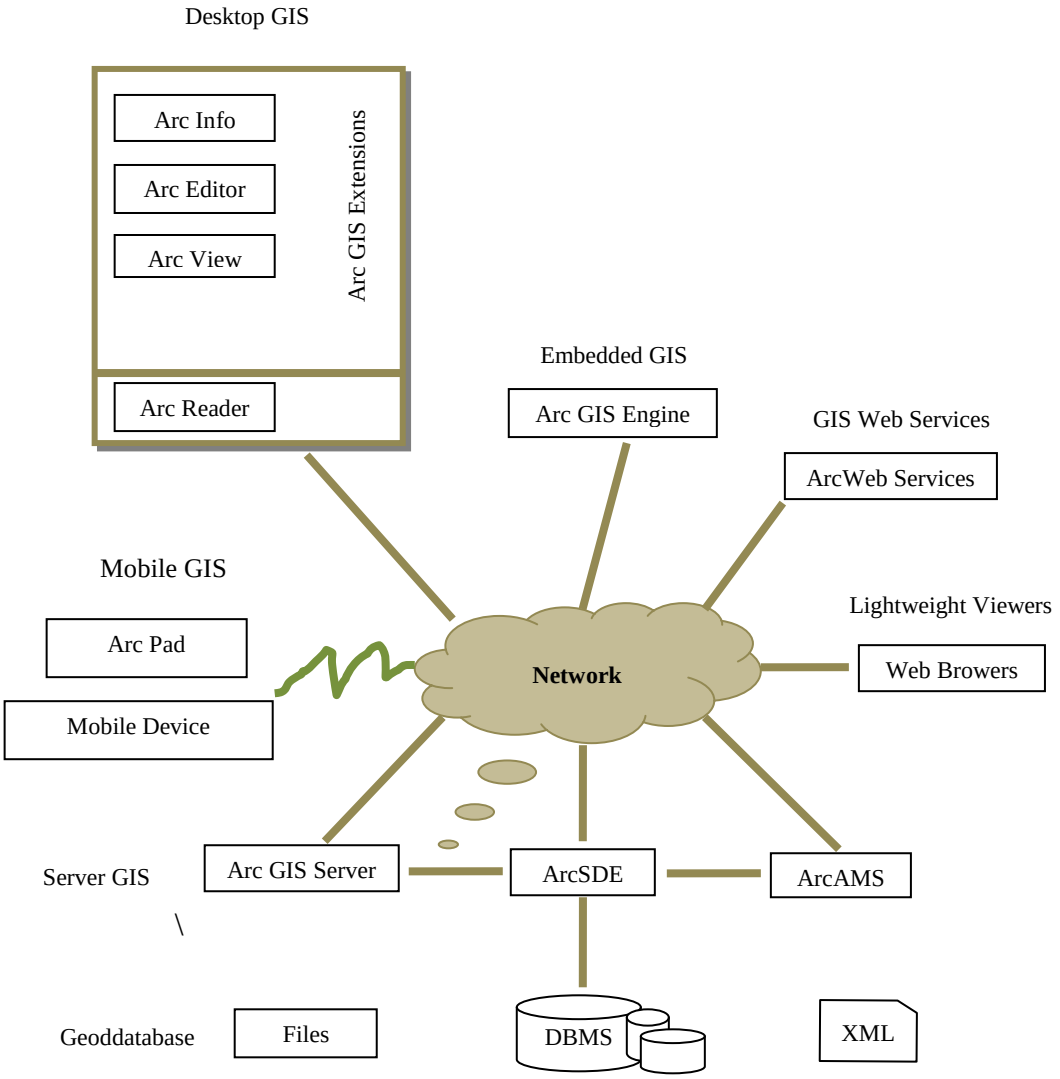
II.1. Sistem Informasi

Sebagaimana yang dikutip Jogiyanto dalam bukunya analisis dan desain sistem informasi Sistem, Robert A.Leitch dan K.Roscoe Davis mendefenisikan Sistem Informasi adalah suatu sistem didalam suatu organisasi yang mempertemukan kebutuhan pengolahan transaksi harian, mendukung operasi, bersifat manajerial dan kegiatan strategi dari suatu organisasi dan menyediakan pihak luar tertentu dengan laporan-laporan yang diperlukan (Riyantodkk, 2009:26).

II.2. Sistem Informasi Geografis

Sistem informasiGeografis (Bahasa Inggris : G*eografhic Information System*) adalah sistem informasi khusus yang mengelola data yang memiliki informasi spasial (bereferensi keruangan) atau dalam arti yang lebih sempit adalah sistem komputer yang memiliki kemampuan untuk membangun, menyimpan, mengelola dan menampilkan informasi bereferensi geografis, misalnya data yang diidentifikasi menurut lokasinya dalam sebuah database. Para praktisi juga memasukkan orang (yang membangun dan mengoperasikannya) dan data sebagai bagian dari sistem ini. (Riyanto dkk., 2009:35).

Adapun arsitektur sistem informasi geografis dapat dilihat pada gambar II.1



Gambar II.1. Arsitektur SIG-ESRI
(Sumber :Ryanto dkk, 2009 : 49).

II.2.1. Sejarah Pengembangan

35000 tahun yang lalu, di dinding Gua Lascaux, Prancis, para pemburu Crog-Magon menggambar hewan mangsa mereka,juga garis yang dipercaya sebagai rute migrasi hewan-hewan tersebut. Catatan awal ini sejalan dengan dua elemen struktur pada sistem informasi geografis modern sekarang ini, arsip garis

yang terhubung ke database attribute.Pada tahun 1700-an teknik survei untuk pemetaan topografis diterapkan, termaksuk juga versi awal pemetaan tematis, misalnya untuk keilmuan atau data sensus. Awal abad ke-20 memperhatikan pengembangan "litografi foto" dimana peta dipishakan menjadi beberapa lapisan (*layer*). Perkembangan perangkat keras komputer yang dipacu oleh penelitian senjata nuklir membawa aplikasi pemetaan menjadi multifungsi pada awal tahun 1960-an.

Tahun 1967 merupakan awal pengembangan SIG yang bisa diterapkan di Ottawa, Ontario oleh Departemen Energi, Pertambangan dan Sumber Daya. Dikembangkan oleh Roger Tomlinson, yaitu kemudian disebut CGIS (Canadian GIS-SIG Kanada), digunakan untuk menyimpan, menganalisis dan mengolah data yang dikumpulkan untun Inventarisasi Tanah Kanada (CLI-Canadian Land Invertory) – sebuah inisiatif untuk mengetahui kemampuan lahan di wilayah pedesaan Kanada dengan memetakan berbagai Informasi pada tanah, pertanian, pariwisata, alam bebas, ungags dan penggunaan tanah pada skala 1:250000. Faktor pemeringkatan klasifikasi juga diterapkan untuk keperluan analisis.(Riyanto dkk., 2009:33).

II.3. Android

Android merupakan OS *Mobile* yang tumbuh di tengah OS lainnya yang berkembang dewasa ini. OS lainnya seperti Windows Mobile, i-Phone OS, Symbian, dan masih banyak lagi juga menawarkan kekayaan isi dan keoptimalan berjalan di atas perangkat *Hardware* yang ada. Akan tetapi, OS yang ada ini

berjalan dengan memproritaskan aplikasi inti yang dibangun sendiri tanpa melihat potensi yang cukup besar dari aplikasi pihak ketiga. Oleh karena itu, adanya keterbasan dari aplikasi pihak ketiga untuk mendapatkan data asli ponsel, berkomunikasi antar proses serta keterbatasan distribusi aplikasi pihak ketiga untuk *platform* mereka.

Android menawarkan sebuah lingkungan yang berbeda untuk pengembang. Setiap aplikasi memiliki tingkatan yang sama. Android tidak membedakan antara aplikasi inti dengan aplikasi pihak ketiga. API yang disediakan menawarkan akses ke *hardware,* maupun data-data ponsel sekalipun, atau data *system* sendiri. Bahkan pengguna dapat menghapus aplikasi inti dan menggantikannya dengan aplikasi pihak ketiga(Stephanus Hermawan S; 2011: 1).

II.3.1. Sejarah Android

Android merupakan sistem operasi yang dikembangkan untuk perangkat *mobile* berbasis Linux. Pada awalnya sistem operasi ini dikembangkan oleh Android Inc yang kemudian dibeli oleh Google pada tahun 2005.Dalam usaha mengembangkan Android, pada tahun 2007 dibentuklah Open Handset Alliance (OHA), sebuah konsorium dari beberapa perusahaan, yaitu Texas Instruments, Broadcom Corporation, Google, HTC, Intel, LG, Marvell Technology Group, Motorola, Nvidia, Qualcomm, Samsung Electronics, Sprint Nextel, dan T-mobile dengan tujuan untuk mengembangkan standar terbuka untuk perangkat *mobile.* Pada tanggal 9 Desember 2008, ia diumumkan bahwa 14 anggota baru akan bergabung proyek Android, termaksuk PacketVideo, ARM Holdings, Atheros

Communication, Asustek Computer Inc, Garmin Ltd, Softbank, Sony Ericsson, Toshiba Corp, dan Vodafone Group Plc.Sistem operasi Android dirilis sebagai berikut :

1. Android versi 1.1

   Pada 9 Maret 2009, Google merilis Android versi 1.1 . Android versi ini dilengkapi dengan pembaruan pada aplikasi, jam alarm, *voice search* (pencarian suara), pengiriman pesan dengan *Gmail* dan pemberitahuan *email.*

2. Android versi 1.5 (*Cupcake*)

   Terdapat beberapa pembaruan termaksuk juga penambahan beberapa fitur dalam versi ini yakni kemampuan merekam dan menonton video dengan modus kamera, dukungan *Bluetooth* A2DP, kemampuan terhubung secara otomatis ke headset *Bluetooth,* animasi layar, dan *keyboard* pada layar yang dapat disesuaikan dengan sistem. Dirilis pada pertengahan Mei 2009.

3. Android versi 1.6 (*Donut*)

   Donut (versi 1.6) dirilis pada pertengahan September 2009 dengan menampilkan proses pencarian yang lebih baik dibanding sebelumnya, penggunaan baterai indicator dan control *applet VPN.* Fitur lainnya adalah galeri yang memungkinkan pengguna untuk memilih foto yang akan dihapus, kamera, *camcorder* dan galeri yang diintegrasikan; *CDMA/EVDO, VPN, Gestures, dan Text-to-speech engine;* kemampuan dial kontak; pengadaan resolusi WVGA.

4. Android versi 2.0/2.1 (*Éclair*)

Android ini diluncurkan pada 3 Desember 2009. Dilakukan perubahan, yaitu pengoptimalan *hardware*, perubahan *User Interface* (UI) dengan browser baru dan dukungan HTML5, daftar kontak yang baru, peningkatan Google Maps 3.1.2, dukungan flash untuk kamera 3,2 MP, *Digital Zoom*, dan *Bluetooth 2.1*.

5. Android versi 2.2 (*Froyo*)

Pada 20 Mei 2010 kembali diluncurkan ponsel Android dengan versi 2.2 (Froyo) perubahan yang dilakuakan meliputi optimasi kecepatan, memori, dan kinerja sistem operasi secara keseluruhan , dukungan untuk menginstal aplikasi pada memori eksternal, dukungan Adobe Flash 10.1 serta fungsi *USB tethering* maupun *Wi-Fi hotspot.*

6. Android versi 2.3 (*Gingerbread*)

1 Desember 2010 Google kembali meluncurkan versi terbaru yaitu Android versi 2.3. Pada versi ini terdapat peningkatan manajemen daya, control melalui aplikasi, penggunaan multiple kamera, peningkatan performa serta penambahan sensor seperti *gyroscope.*

7. Android versi 3.0/3.1 (*Honeycomb*)

Versi ini berbeda dengan versi-versi sebelumnya. Versi ini dirancang khusus untuk PC Tablet sehingga memiliki *User Interface* yang berbeda dan mendukung ukuran layar yang lebih besar. Selain itu, pada versi ini memungkinkan penggunaan multiprosesor dan akselerasi perangkat keras

untuk grafis. SDK versi pertama diluncurkan Februari 2011(Stephanus Hermawan S; 2011: 1).

8. Android versi 4.0 ICS (*Ice Cream Sandwitch*)

Android 4.0-4.0.2 API Level 14 dan 4.0.3-4.0.4API Level 15 pertama dirilis 19 Oktober 2001. Dinamai *Ice Cream Sandwich*. *Ice Cream Sandwich* adalah lapisan eskrim, biasanya rasa vanilla yang terjepit di antara dua kue coklat, dan biasanya berbentuk persegi panjang.(Hendra NugrahaLengkong; 2015: 21).

9. Android versi 4.1 (*Jelly Bean*)

Android Jelly Bean diluncurkan pertama kali pada Juli 2012, dengan berbasis *Linux Kernel*3.0.31.Terdiri dari Android 4.1 API Level 16, Android 4.2 API Level 17, Android 4.3 API Level 18.Penamaan *Jelly Bean* mengadaptasi nama sejenis permen dalam beraneka macam rasa buah. Ukurannya sebesar kacang merah. Permen ini keras di luar tapi lunak di dalam serta lengket bila di gigit(Hendra NugrahaLengkong; 2015: 21).

10. Android versi 4.4 (*KitKat*)

Android 4.4 Kitkat API level 19.Google mengumumkan Android KitKat (dinamai dengani zindari Nestle dan Hershey) pada 3 september 2013. Dengan tanggal rilis 31 Oktober 2013.KitKat merupakan merk sebuah coklat yang dikeluarkan oleh Nestle. Rilis berikutnya setelah nama KitKat diperkirakan banyak pengamat akan diberi nomor 5.0 dan dinamai '*Key LimePie'*.(Hendra Nugraha Lengkong; 2015: 21).

II.3.2. Fitur dan Arsitektur Android

Fitur yang tersedia pada Android adalah:

1. *Framework* aplikasi : memungkinkan penggunaan dan pemindahan dari komponen yang tersedia.
2. *Dalvik virtual machine* : *virtual machine* yang dioptimalkan untuk perangkat *mobile*
3. Grafik : grafik 2D dan grafik 3D yang didasarkan pada *library OpenGL*.
4. *SQLite* : untuk penyimpanan data.
5. Mendukung media : *audio, video,* dan berbagai format gambar (MPEG4, H.264, MP3, ACC, AMR, JPG, PNG, GIF)
6. *GSM, Bluetooth, EDGE, 3G, and WiFi* (tergantung *hardware)*
7. *Camera, Global Position System (GPS), compass,* dan *accelerometer* (tergantung *hardware*).
8. Lingkungan pengembangan yang kaya, termaksuk emulator, peralatan debugging, dan *plugin* untuk *Eclipse IDE*. (Stephanus Hermawan S; 2011: 6).

Sistem operasi Android dibangun berdasarkan karnel Linux, dan memiliki arsitektur sebagai berikut:

1. *Applications*

   Lapisan ini adalah lapisan aplikasi, serangkaian aplikasi akan terdapat pada perangkat *mobile*. Aplikasi inti yang telah terdapat pada Android

termaksuk kalender, kontak, SMS, dan lain sebagainya. Aplikasi-aplikasi ini ditulis dengan Bahasa pemrograman *Java.*

2. *Application Framework*

Pengembang aplikasi memiliki akses penuh ke Android sama dengan aplikasi inti yang telah tersedia. Pengembang dapat dengan mudah mengakses informasi lokasi, mengatur alarm, menambahkan pemberitahuan ke status bar dan lain sebagainya. Arsitektur aplikasi ini dirancang untuk menyederhanakan penggunaan kembali komponen, aplikasi apa pun dapat memublikasikan kemampuan dan aplikasi lain dapat menggunakan kemampuan mereka sesuai batasan keamanan. Dasar dari aplikasi adalah seperangkat layanan dan sistem, yaitu berbagai *view* yang digunakan untuk membangaun UI, *Conten Provider* yang memungkinkan apliaksi berbagi data, *Resource Manager* menyediakan akses bukan kode seperti grafik, *string*, dan *layout, Notification Manager* yang akan membauat aplikasi dapat menampilkan tanda pada status bar dan *Activity Manager* yang berguna mengatur daur hidup dari aplikasi.

3. *Libraries*

Satu set *libraries* dalam bahasa C/C++ yang digunakan oleh berbagai komponen pada sistem Android.

4. Android *Runtime*

Satu set *libraries* inti yang menyediakan sebagian besar fungsi yang tersedia di *libraries* inti dari bahasa pemrograman *Java*. Setiap aplikasi akan berjalan sebagai proses sendiri pada *Dalvik Virtual Machine* (VM).

5. Linux Karnel

Android teragantung pada Linux versi 2.6 untuk layanan sistem inti seperti keamanan, manajemen memori, manajemen proses, *network stack*, dan model *driver*. Karnel juga bertindak sebagai lapisan antara *hardware* dan seluruh *software*(Stephanus Hermawan S; 2011: 7).

II.5. Java

Java adalah bahasa pemrograman yang dapat dijalankan di berbagai komputer, termaksuk telepon genggam. Bahasa ini awalnya dibuat oleh James Gosling saat masih bergabung di Sun Microsystem yang sekarang ini merupakan bagaian dari Oracle.Bahasa Java mulai dirilis tahun 1995. Bahasa ini banyak mengadopsi sintaks yang terdapat pada C dan C++, namun dengan model objek yang lebih sederhana serta dukungan rutin-rutin level bawah yang minimal.

Aplikasi-aplikasi berbasis Java umumnya dikompilasi ke dalam p-code (*bytecode*) dan dapat dijalankan pada berbagai Mesin Virtual Java (JVM). Java merupakan bahasa pemrograman yang bersifat umum/non-spesifik (*general purpose*). Secara khusus, Java didesain untuk memanfaatkan dependensi implementasi seminimal mungkin. Karenanya aplikasi Java mampu berjalan dibeberapa *platform* sistem operasi yang berbeda. Java dikenal pula dengan

slogannya, “Tulis sekali, jalankan di mana pun”. Saat ini Java merupakan bahasa pemrogaraman yang paling popular digunakan, dan secara luas dimanfaatkan dalam pengembangan berbagai jenis perangkat lunak aplikasi ataupun aplikasi berbbasis web.

Salah satu hal yang paling sulit di Java adalah menginstal apa-apa yang diperlukan sampai bisa melakukan pemrograman. Bahkan sebelum anda bisa menuliskan satu baris kode, mungkin anda sudah mulai pusing bagaimana menginstal beragam prasyarat dari bahasa pemrograman Java ini. Untungnya, saat ini banyak tool yang langsung memudahkan bahasa pemrograman Java, misalnya *Eclipse* dan *Netbeans*. Keduanya selain memudahkan pemrograman, juga menyediakan tool perancangan GUI yang lengkap.Tool memudahkan pemrograman ini disebut IDE(*Integrated Development Environment*). Ini memudahkan pemrograman karena memudahkan anda menulis program Java. Contohnya disini penulis menggunakan tool *Eclipse*. Tapi sebelum menginstal *Eclipse,* anda perlu mengistal *Java Virtual Machine.*(Edy Winarno, dkk,. 2013 :1-3).

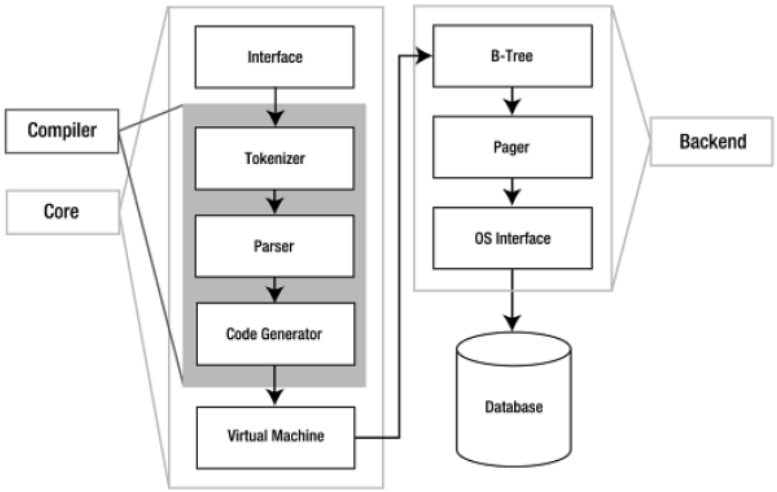
II.6. *SQLite*

Menurut Owens (2006:1), *SQLite* adalah sebuah open source relational database. Dirilis pada tahun 2000, *SQLite* di desain untuk menyediakan cara mudah bagi aplikas untuk mengatur data tanpa *overhead* yang sering timbul dari *dedicated relational management systems*. *SQLite* memiliki reputasi dalam hal

protabilitas, mudah digunakan, efisien, dan *reliable*.(Aris Dwi Fitriyanti, 2011 : 7).

II.6.1. Arsitektur *SQLite*

SQlite memiliki arsitektur modular yang menggunakan pendekatan unik untuk *relational database management*. *SQLite* terdiri dari delapan grup modul yang terpisah dalam tiga subsistem utama. Modul ini membagi proses query menjadi beberapa diskrit yang bekerja seperti sebuah perakitan. Stack teratas meng-compile query, bagian tengah mengeksuksi, dan bagan bawah menangani penyimpaan dan interface dengan sistem operasi. Arsitektur SQLite secara garis besar tergambar pada gambar 2.



Gambar II.1 *SQLite's architecture*
(Sumber :Aris Dwi Fitriyanti, 2011 : 7).

1. Interface

Interface adalah bagian atas dari stack dan terdiri dari *SQLite* C API, dimana terlepas dari program, bahasa script, dan *libraries* yang pada akhirnya berinteraksi dengan*SQLite.* (Aris Dwi Fitriyanti, 2011 : 7).

2. Compiler

Proses kompilasi dimulai dengan Tokenizer dan Parser. Pada dasarnya keduanya bekerja sama untuk mengambil perintah Structured Query Language (SQL) dalam bentuk teks, memvalidasi sintaks, dan kemudian mengubahnya menjadi struktur data hirarkis dimana lapisan atau layer bawah dapat bekerja dengan lebih mudah.

SQLite Tokenizer adalah *hand coded*. Parser dihasilkan oleh parser generator *SQLite*, yang disebut Lemon. Generator parser lemon dirancang untuk kinerja tinggi untuk mencegah kebocoran memori. Setelah perintah atau sintaks dipecah menjadi token, dievaluasi, dan dikembalikan dalam bentuk *parse tree*, karena parser melewati *tree* saat kembali ke generator kode.

Kode generator menerjemahkan parse tree menjadi bahasa assembler khusus untuk *SQLite*. Bahasa assembler ini terdiri dari perintah yang dieksekusi oleh *virtual machine*. Fungsi kode generator lainnya adalah untuk mengkonversi parse tree menjadi mini-program yang lengkap dan di tulis dalam assembler dan menyerahkannya ke virtual machine untuk proses selanjutnya(Aris Dwi Fitriyanti, 2011 : 7).

3. *Virtual Machine*

Bagian inti dari arsitektur *SQLite* adalah *virtual machine,* atau disebut dengan *Virtual Database Engine* (VDBE). VDBE bekerja menggunakan byte code, seperti Java virtual machine. *Byte Code* untuk VDBE adalah 128 upcode, dimana semuanya terkonsentrasi dalam operasi database. Setiap instruksi yang masuk, digunakan untuk menyelesaikan operasi database tertentu seperti membuka kursor tabel, atau untuk penyusunan ruang stack operasi, misalnya ditekan kedalam parameter.(Aris Dwi Fitriyanti, 2011 : 7).

4. *Beck End*

Back-end terbentuk dari B-tree, *page cac*he (pager) and OS Interface.B-tree dan page cache bertugas untuk mengolah data. B-tree mempunyai fungsi utama untuk mempertahankan hubungan yang kompleks antara berbagai halaman, untuk mempermudah dan mempercepat dalam menemukan data yang diinginkan. Sedangkan fungsi utama page cache (pager) dilewatkan antara B-tree dan Disk page melakui OS Interface.(Aris Dwi Fitriyanti, 2011 : 7).

II.7. Metode Dijkstra

Algoritma Dijkstra, (dinamai menurut penemunya, seorang ilmuwan komputer, Edsger Dijkstra), adalah sebuah algoritma rakus (*greedy algorithm*) yang dipakai dalam memecahkan permasalahan jarak terpendek (*shortest path problem*) untuk sebuah graf berarah (*directed graph*) dengan bobot-bobot sisi (*edge weights*) yang bernilai tak-negatif.

Misalnya, bila vertex dari sebuah graf melambangkan kota-kota dan bobot sisi (*edge weights*) melambangkan jarak antara kota-kota tersebut, maka algoritma

Dijkstra dapat digunakan untuk menemukan jarak terpendek antara dua kota (UswahHasanahdkk; 2014: 3).

II.8. *Unified Modeling Language* (UML)

Unified Modeling Language adalah sebuah bahasa yang diterima dan digunakan oleh software developer dan software analyst sebagai suatu bahasa yang cocok untuk merepresentasikan grafik dari suatu relasi antar entitas-entitas software (Gornik, 2003). Dengan menggunakan UML, tim pengembang software akan mempunyai banyak keuntungan, seperti memudahkan komunikasi dengan sesama anggota tim tentang software apa yang akan dibuat, memudahkan integrasi ke dalam area pengerjaan software karena bahasa ini berbasiskan meta-models dimana meta-models bisa mendefinisikan proses-proses untuk mengkonstruksikan konsep-konsep yang ada. UML juga menggunakan formatur input dan output yang sudah mempunyai bentuk standar yaitu XML Metadata Interchange (XMI), menggunakan aplikasi dan pemodelan data yang universal, merepresentasikan dari tahap analisis ke implementasi lalu ke *deployment* yang terpadu, dan mendeskripsikan keutuhan tentang spesifikasi software. (Edgar Winata, Johan Setiawan, 2013).

UML menyediakan kumpulan alat yang sudah terstandarisasi, yang digunakan untuk mendokumentasikan analisis dan perancangan sebuah sistem perangkat lunak. Peralatan utama UML adalah diagram-diagram yang digunakan untuk membantu manusia dalam memvisualisasikan proses pengembangan sebuah

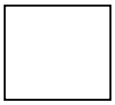
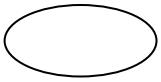
sistem perangkat lunak, sama seperti penggunaan denah (blueprint) dalam pembuatan bangunan. (Edgar Winata, Johan Setiawan, 2013).

II.8.1. Diagram *Unified Modeling Language* (UML)

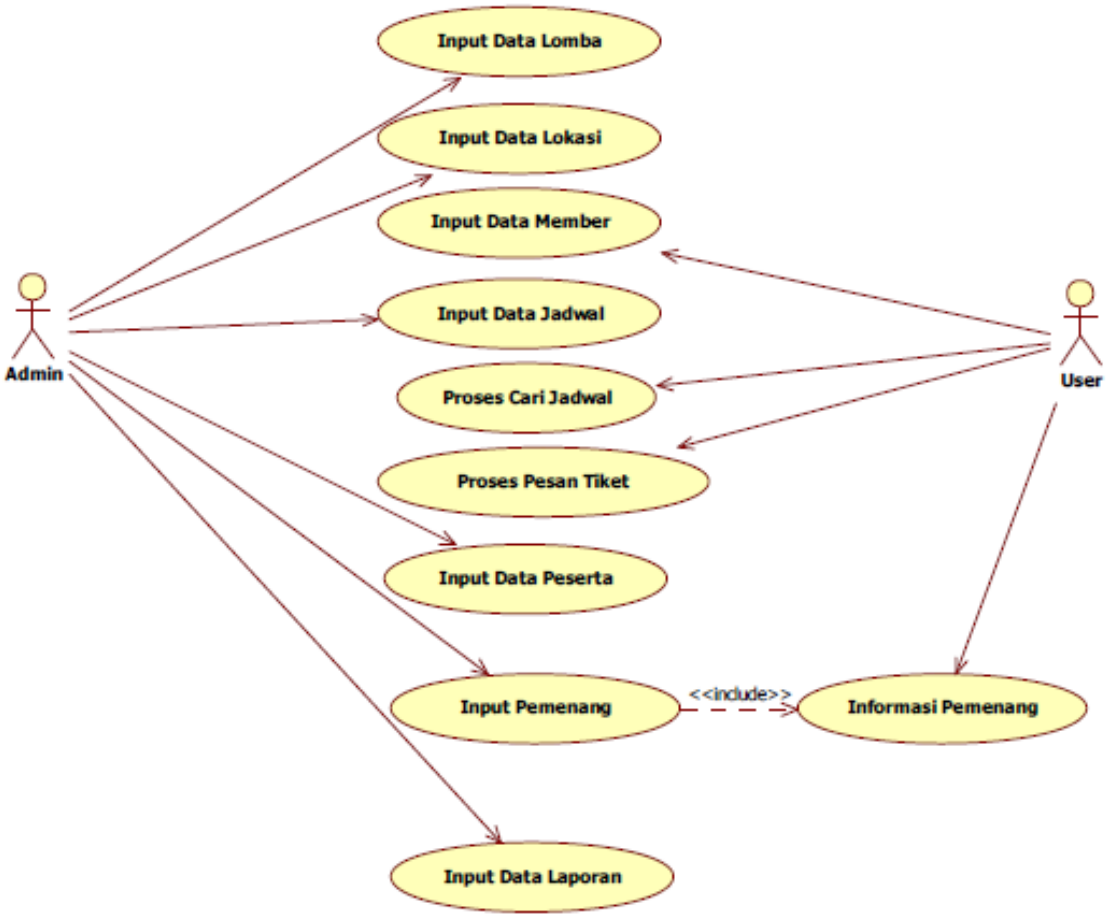
1. Use Case Model

Use case model adalah teknik pemodelan untuk mendapatkan *functional requirement* dari sebuah sistem, menggambarkan interaksi antara pengguna dan sistem, menjelaskan secara naratif bagaimana sistem akan digunakan, menggunakan skenario untuk menjelaskan setiap aktivitas yang mungkin terjadi. Ada beberapa bagian didalam use case model. (Edgar Winata, Johan Setiawan, 2013).

Tabel II.1. Komponen *Use Case Diagram*

	*System Boundry menggambarkan antara sistem dengan actor*
	Simbol ini menggambarkan interaksi antara actor dengan *software* aplikasi tersebut.
	Actor yang menggambarkan pengguna dari sistem, dapat berupa manusia atau sistem terotomatisasi lain yang berinteraksi dengan sistem lain untuk berbagi, mengirim, dan menerima informasi.
	Menggambarkan hubungan antar actor dan *use case*.

(Sumber :Indrajani, 2015 : 46).

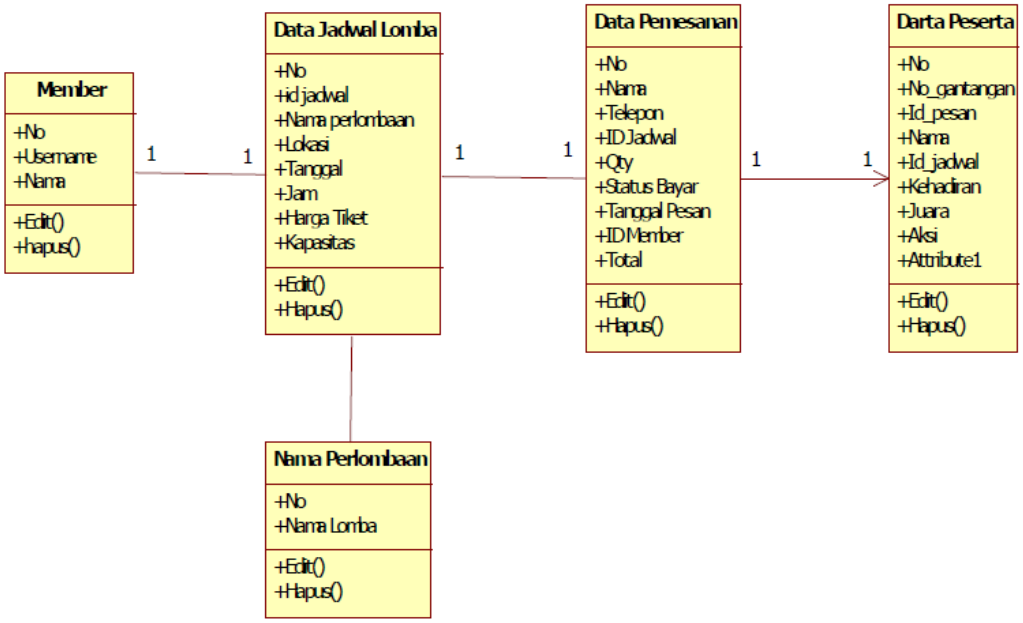


Gambar II.2. *Use Case* Diagram
(Sumber : Yohanes Krisdian, Ernes Cahyo Nugroho., 2015)

2. *Class Diagram*

Class diagram merupakan diagram paling umum yang dijumpai dalam pemodelan berbasis UML. Didalam Class diagram terdapat class dan interface beserta atribut-atribut dan operasinya, relasi yang terjadi antar objek, *constraint* terhadap objek-objek yang saling berhubungan dan *inheritance* untuk organisasi class yang lebih baik. Class diagram juga terdapat *static view* dari elemen pembangun sistem. Pada intinya Class

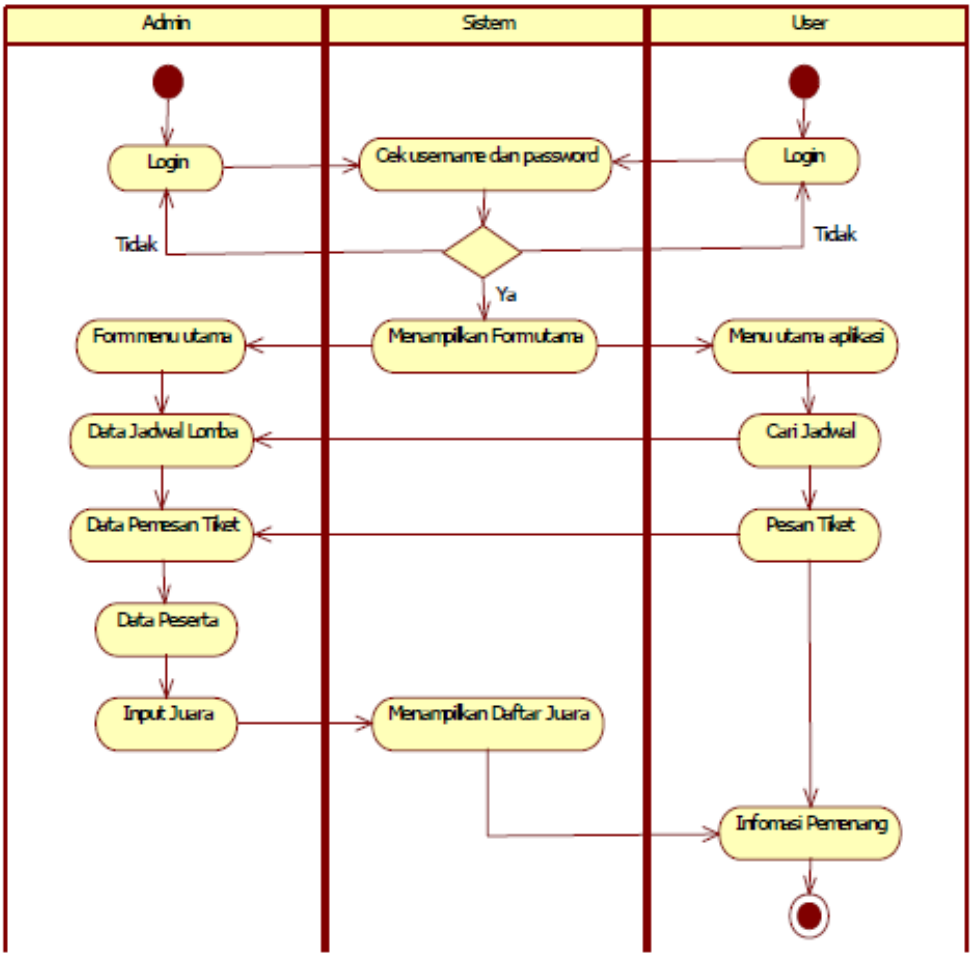
diagram mampu membantu proses pembuatan sistem dengan memanfaatkan konsep *forward* ataupun *reverse engineering*.(Edgar Winata, Johan Setiawan, 2013).



Gambar II.4.*Class* Diagram
(Sumber :Yohanes Krisdian, Ernes Cahyo Nugroho., 2015)

3. *Activity Diagram*

Teknik untuk menjelaskan *business process*, *procedural logic*, dan *work flow*. Bisa dipakai untuk menjelaskan teks use case dalam notasi grafis dengan menggunakan notasi yang mirip *flow chart*, meskipun terdapat sedikit perbedaan notasi. (Edgar Winata, Johan Setiawan, 2013).

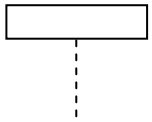
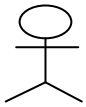


Gambar II.3. *Activity* Diagram
(Sumber : Yohanes Krisdian, Ernes Cahyo Nugroho., 2015)

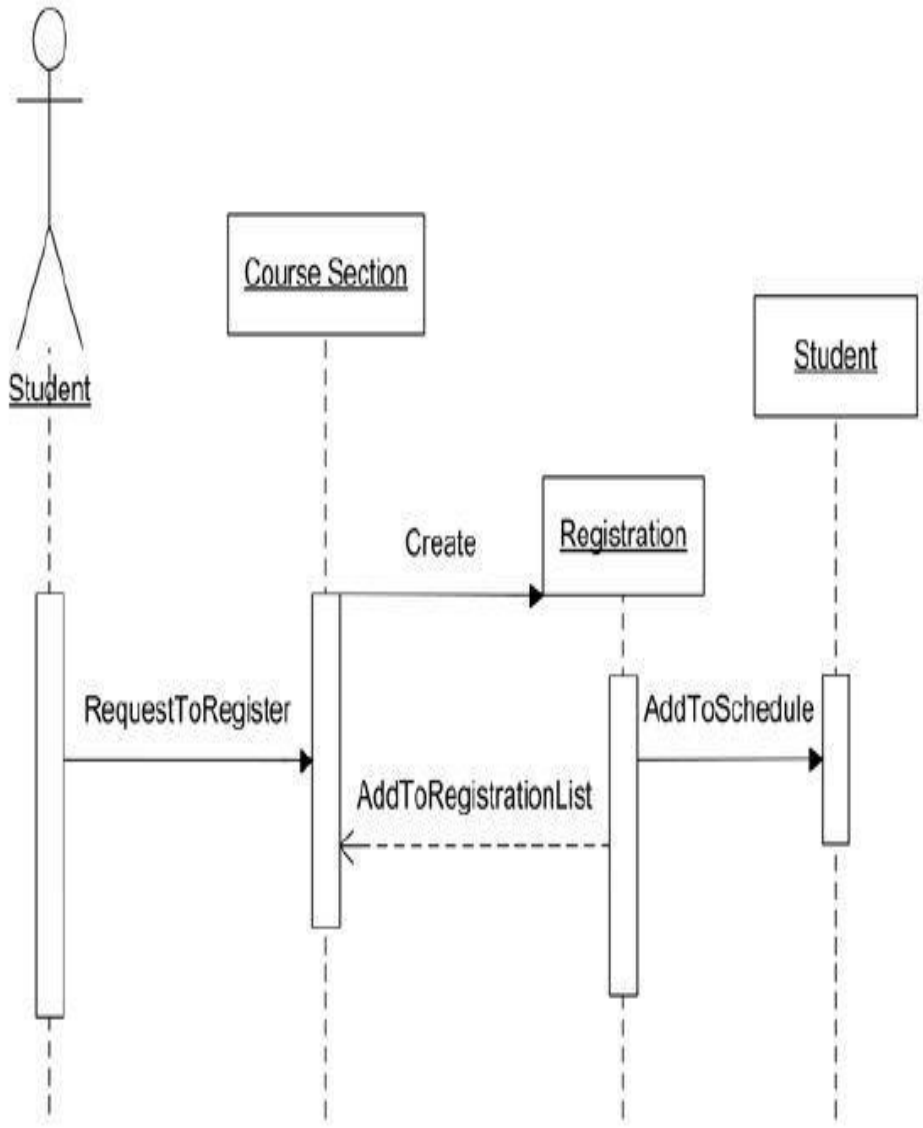
4. *Squence* Diagram

   Menjelaskan interaksi obyek-obyek yang saling berkolaborasi (berhubungan), mirip dengan *activity diagram* yaitu menggambarkan alur kejadian sebuah aktivitas tetapi lebih detil dalam menggambarkan aliran data termasuk data atau behaviour yang dikirimkan atau diterima namun kurang mampu menjelaskan detil dari sebuah algoritma. (Edgar Winata, Johan Setiawan, 2013).

Tabel II.1. Komponen *Use Case Diagram*

	*Object Lifeline* : menggambarkan object apa saja yang terlibat
	*Actor* : menggambarkan actor yang terlibat
	Activation : menggambarkan hubungan antara object dengan message.
	Message (call) : menggambarkan alur message yang merupakan kejadian dari objek pengirim lifeline ke objek penerima lifeline.
	Message (return) : menggambarkan alur pengembalian message ke objek pemanggil dan tanda bahwa objek penerima telah menyelesaikan prosesnya.

(Sumber :Indrajani, 2015 : 46)



Gambar II.5. *Sequence* Diagram
(Sumber : Edgar Winata, Johan Setiawan, 2013)