BAB II

TINJAUAN PUSTAKA

II.1. Aplikasi

Aplikasi menurut Jogiyanto adalah penggunaan dalam suatu komputer, instruksi (instructiom) atau pernyataan (statement) yang disusun sedemikian rupa sehingga komputer dapat memproses input menjadi output.(Ihsanudinet al;2014:2)

II.2. Informasi

Informasi adalah data yang telah diolah menjadi bentuk yang lebih berarti bagi penerimanya. Data adalah kenyataan yang menggambarkan kejadian-kejadian dan kesatuan nyata. Kejadian adalah sesuatu yang terjadi pada saat tertentu.(Fahrudinet al;2011:2)

II.3.Akademik

Kata akademik berasal dari bahasa Yunani yakni *Academos* yang berarti sebuah taman umum (plasa) di sebalah barat laut kota Athena. Nama *Academos* adalah nama seorang pahlawan yang terbunuh pada saat perang legendaris Troya. Pada plasa inilah filosof Socrates berpidato dan membuka arena perdebatan tentang berbagai hal. Tempat ini juga menjadi tempat plato melakukan dialog dan mengajarkan pikiran-pikiran filosofisnya kepada orang-orang yang datang.

Sesudah itu, kata *Acadomos* berubah menjadi akademik, yaitu semacam tempat perguruan. Para pengikut perguruan disebut *Academist*, sedangkan perguruan semacam itu disebut *Academia*. Berdasarkan hal ini, inti dari pengertian akademik adalah menyampaikan dan

menerima gagasan, pemikiran, ilmu pengetahuan, dan sekaligus dapat mengujinya secara jujur, terbuka, dan leluasa.

II.4. Android

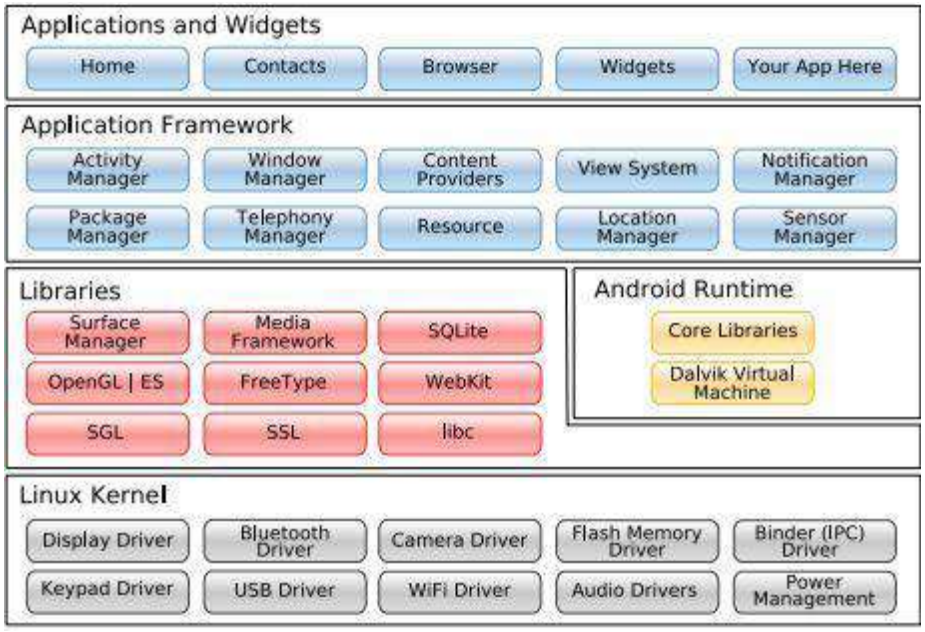
Android adalah sebuah sistem operasi untuk perangkat *mobile* berbasis *linux* yang mencakup sistem operasi, *middleware* dan aplikasi. *Android* menyediakan *platform* terbuka bagi para pengembang untuk menciptakan aplikasi mereka. Awalnya *Google inc* membeli *android inc* yang merupakan pendatang baru yang membuat piranti lunak untuk ponsel/*smartphone*. Kemudian untuk mengembangkan *android*, dibentuklah *Open Handesti Alliance*, konsorium dari 34 perusahaan peranti keras, peranti lunak, dan telekomunikasi, termasuk *Google, HTC, Intel, Motorola, Qualcomm, T-Mobile, dan Nvidia.* Pada saat perilisan perdana *android*, 5 November 2007, *Android* bersama *Open Handest Alliance* menyatakan mendukung pengembangan *open source* pada perangkat *mobile*. Di lain pihak, *Google* merilis kode – kode *android* dibawah lisensi *Apache*, sebuah lisensi perangkat lunak dan *open platform* perangkat seluler. (Safaat H& Nazruddin;2012)

Gambar II.1. *Android 6.0 Marshmallow* Versi *Android* Terbaru

(http://collegemag.net)

Aplikasi *Android* ditulis dalam bahasa pemrograman *java*, yaitu kode *java* yang terkompilasi bersama-sama dengan data dan *file-file* sumber yang dibutuhkan oleh aplikasi yang digabungkan oleh *app tools* menjadi paket aplikasi *Android*, sebuah file yang ditandai dengan akhiran *.apk* file inilah yang didistribusikan sebagai aplikasi dan diinstal pada *handset Android.* (Ahmad;2015:191)

II.4.1.Arsitektur Android



Gambar II.2. Arsitektur Android

(Hartono;2013:23)

Arsitektur Android seperti yang ditunjukkan pada Gambar terdiri dari 5 bagian utama, yaitu Application dan Widgets, Application Framework, Libraries, Android Runtime, dan Linux Kernel.(Hartono;2013:23)

II.4.2.Versi Android

Versi Android diawali dengan dirilisnya Android beta pada bulan November 2007. Versi komersial pertama, Android 1.0, dirilis pada September 2008. Android dikembangkan secara berkelanjutan oleh Google dan Open Handset Alliance (OHA), yang telah merilis sejumlah pembaruan sistem operasi ini sejak dirilisnya versi awal.

Sejak April 2009, versi Android dikembangkan dengan nama kode yang dinamai berdasarkan makanan pencuci mulut dan penganan manis. Masing – masing versi dirilis sesuai urutan alfabet,

yakni Cupcake (1.5), Donut (1.6), Eclair (2.0–2.1), Froyo (2.2–2.2.3), Gingerbread (2.3–2.3.7), Honeycomb (3.0–3.2.6), Ice Cream Sandwich (4.0–4.0.4), Jelly Bean (4.1–4.3), KitKat (4.4+), Lollipop (5.0+), Marshmallow (6.0) Pada tanggal 3 September 2013, Google mengumumkan bahwa sekitar 1 miliar perangkat seluler aktif di seluruh dunia menggunakan OS Android. Android pun di gosipkan akan Membuat pembaruan untuk versi 7.0 Dengan nama Nuttela.

1. Android 1.0 (API level 1)

   Android 1.0, versi komersial pertama Android, dirilis pada 23 September 2008.Perangkat Android pertama yang tersedia secara komersial adalah HTC Dream.

2. Android 1.1 (API level 2)

   Pada 9 Februari 2009, pemutakhiran Android 1.1 dirilis, awalnya hanya untuk HTC Dream. Android 1.1 juga dikenal dengan "*Petit Four*", meskipun nama ini tidak digunakan secara resmi.

3. Android 1.5 Cupcake (API level 3)

   Pada 27 April 2009, Android 1.5 dirilis, menggunakan kernel Linux 2.6.27.Versi ini adalah rilis pertama yang secara resmi menggunakan nama kode berdasarkan nama – nama makanan pencuci mulut ("*Cupcake*"), nama yang kemudian digunakan untuk semua versi rilis selanjutnya.

4. Android 1.6 Donut (API level 4)

   Pada 15 September 2009, SDK Android 1.6 – dinamai Donut – dirilis, berdasarkan kernel Linux 2.6.29.

5. Android 2.0 Eclair(API level 5)

Pada 26 Oktober 2009, SDK Android 2.0 – dinamai Eclair – dirilis, berbasis kernel Linux 2.6.29.

6. Android 2.0.1 Eclair (API level 6)

   Perbaikan versi Eclair sebelumnya.

7. Android 2.0.2 Eclair (API level 7)

   Perbaikan versi Eclair sebelumnya.

8. Android 2.2–2.2.3 Froyo (API level 8)

   Pada 20 Mei 2010, SDK Android 2.2 (Froyo, singkatan untuk frozen yogurt) dirilis, yang berbasis kernel Linux 2.6.32.

9. Android 2.3–2.3.2 Gingerbread (API level 9)

   Pada tanggal 6 Desember 2010, SDK Android 2.3 (Gingerbread) dirilis, berbasis kernel Linux 2.6.35.

10. Android 2.3.3–2.3.7 Gingerbread (API level 10)

    Perbaikan versi Gingerbread sebelumnya.

11. Android 3.0 Honeycomb (API level 11)

    Pada 22 Februari 2011, SDK Android 3.0 (Honeycomb) – pembaruan pertama Android yang ditujukan hanya untuk komputer tablet – dirilis, berdasarkan kernel Linux 2.6.36.Perangkat pertama yang menggunakan versi ini adalah tablet Motorola Xoom, yang dirilis pada 24 Februari 2011.

12. Android 3.1 Honeycomb (API level 12)

    Perbaikan versi Honeycomb sebelumnya.

13. Android 3.2 Honeycomb (API level 13)

Perbaikan versi Honeycomb sebelumnya Google TV generasi pertama dan kedua menggunakan Honeycomb 3.2.

14. Android 4.0–4.0.2 Ice Cream Sandwich (API level 14)

    SDK Android 4.0.1 (Ice Cream Sandwich), berdasarkan kernel Linux 3.0.1,dirilis pada 19 Oktober 2011.Petinggi Google, Gabe Cohen, menyatakan bahwa Android 4.0 "secara teoritis kompatibel" dengan perangkat Android 2.3x yang diproduksi pada saat itu. Kode sumber untuk Android 4.0 tersedia pada tanggal 14 November 2011.

15. Android 4.0.3–4.0.4 Ice Cream Sandwich (API level 15)

    Ice Cream Sandwich adalah versi terakhir yang mendukung Flash player Adobe Systems.

16. Android 4.1 Jelly Bean (API level 16)

    Google mengumumkan Android 4.1 (Jelly Bean) dalam konferensi Google I/O pada tanggal 27 Juni 2012. Berdasarkan kernel Linux 3.0.31, Jelly Bean adalah pembaruan penting yang bertujuan untuk meningkatkan fungsi dan kinerja antarmuka pengguna (UI). Pembaruan ini diwujudkan dalam "Proyek Butter", perbaikan ini termasuk antisipasi sentuh, triple buffering, perpanjangan waktu vsync, dan peningkatan frame rate hingga 60 fps untuk menciptakan UI yang lebih halus.Android 4.1 Jelly Bean dirilis untuk Android Open Source Project pada tanggal 9 Juli 2012.Perangkat pertama yang menggunakan sistem operasi ini adalah tablet Nexus 7, yang dirilis pada 13 Juli 2012.

17. Android 4.2 Jelly Bean (API level 17)

    Google berencana merilis Jelly Bean 4.2 pada sebuah acara di New York City pada 29 Oktober 2012, tapi dibatalkan karena Badai Sandy. Jelly Bean 4.2 didasarkan pada kernel Linux 3.4.0, dan pertama kali digunakan pada Nexus 4 LG dan Nexus 10 Samsung, yang dirilis pada 13 November 2012.

18. Android 4.3 Jelly Bean (API level 18)

    Google merilis Jelly Bean 4.3 pada 24 Juli 2013 di San Francisco. Kebanyakan perangkat Nexus menerima pembaruan dengan segera. Nexus 7 generasi kedua adalah perangkat pertama yang menggunakan sistem operasi ini. Sebuah pembaruan minor dirilis pada tanggal 22 Agustus 2013.

19. Android 4.4 KitKat (API level 19)

    Google mengumumkan Android 4.4 KitKat (dinamai dengan izin dari Nestlé dan Hershey) pada 3 September 2013, dengan tanggal rilis 31 Oktober 2013.Sebelumnya, rilis berikutnya setelah Jelly Bean diperkirakan akan diberi nomor 5.0 dan dinamai 'Key Lime Pie'.

20. Android 5.0 Lollipop (API level 21)
21. Android 6.0 Marshmallow (API level 23)

II.5. PHP (PHP Hypertext Preprocessor)

PHP adalah bahasa pemrograman yang digunakan dalam pengembangan web untuk memproses data dinamis dengan cepat. PHP adalah server-side embedded script language, yaitu script berisi syntax perintah yang sepenuhnya dijalankan oleh server tetapi disertakan pada halaman HTML. PHP merupakan script yang terbilang baru dan tersedia secara bebas, dan masih memungkinkan untuk dikembangkan lebih lanjut. PHP dapat diintegrasikan (embedded) ke dalam web server, atau dapat berperan sebagai program CGI yang terpisah (Andiloloet al;2014:2).

II.6. Laravel

Laravel merupakan web development framework dengan Model View Controller (MVC) yang ditulis dalam PHP. Laravel telah didesain untuk meningkatkan kualitas software dengan mengurangi biaya development awal dan biaya maintenance serta untuk meningkatkan pengalaman bekerja dengan menyediakan syntax yang ekspresif yang akan menghemat waktu di dalam proses implementasinya (Awaludin;2014:10).

II.7. MySQL

MySQL merupakan suatu software sistem manajemen database yang open source. MySQL adalah database server yang dibuat dan distribusikan oleh perusahaan komersial yaitu MySQL AB. MySQL didistribusikan secara gratis di bawah lisensi General Public License (GPL). MySQL adalah sistem yang mendukung relational database. Artinya, dalam sebuah database memiliki beberapa table untuk menyimpan data – data dimana masing – masing tabel memiliki hubungan atau relasi satu sama lain sehingga dapat dilakukan kombinasi data dari beberapa tabel dalam satu saat. Sistem semacam ini sering disebut pula dengan RDBMS (Relational DataBase Management System). Sistem manajemen database seperti MySQL diperlukan untuk menambahkan, mengakses, memproses data yang disimpan di server (Andilolo et al;2014:2).

II.8. Software

Software merupakan program – program komputer yang berguna untuk menjalankan atau mengoperasikan suatu pekerjaan sesuai dengan yang dikehendaki. Program tersebut ditulis

dengan bahasa khusus yang dimengerti oleh komputer. Program dapat dianalogikan sebagai instruksi atau perintah – perintah untuk mengoperasikan atau menjalankan *hardware*. *Software* terdiri dari berbagai jenis, yaitu sistem operasi, program *utility* (bantuan), program aplikasi, dan program paket (Sariadin;2009:3).

Gambar II.3. Tampilan Salah Satu Software *IDE*

(http://google.com)

II.9. Eclipse

*Eclipse*adalah sebuah IDE(*Integrated Development Environment*) untuk mengembangkan perangkat lunak dan dapat dijalankan di semua platform (*platform-independent*). Eclipse dikembangkan dengan bahasa pemrograman Java, akan tetapi Eclipse mendukung pengembangan aplikasi berbasis bahasa pemrograman lainnya, seperti C/C++, Cobol, Python, Perl, PHP, dan lain sebagainya. Selain sebagai IDE untuk pengembangan aplikasi, Eclipse pun bisa digunakan untuk aktivitas dalam siklus pengembangan perangkat lunak, seperti dokumentasi, test perangkat lunak, pengembangan web, dan lain sebagainya. Eclipse pada saat

ini merupakan salah satu IDE favorit dikarenakan gratis dan *open source*, yang berarti setiap orang boleh melihat kode pemrograman perangkat lunak ini. Selain itu, kelebihan dari Eclipse yang membuatnya populer adalah kemampuannya untuk dapat dikembangkan oleh pengguna dengan komponen yang dinamakan *plug-in*. *Eclipse+AVR plugin*, dengan tambahan *plugin* tersebut kita dapat memprogram mikrokontroler AVR menggunakan IDE ini, selain itu keuntungan menggunakan *Eclipse* ialah dapat bekerja di berbagai sistem operasi seperti Microsoft Windows, Linux, Solaris, AIX, HP-UX dan Mac OS X.

II.10. Android Studio

Android Studio adalah sebuah IDE (*Integrated Development Environment*) untuk mengembangkan aplikasi Android. *Android Studio* ini adalah lingkungan pengembangan baru dan terintegrasi penuh, yang baru saja dirilis oleh *Google* untuk sistem operasi Android. *Android Studi*o dirancang untuk menjadi peralatan baru dalam pengembangan aplikasi dan juga memberi alternatif lain selain Eclipse yang saat ini menjadi IDE yang paling banyak dipakai.

Saat Anda memulai proyek baru dengan *Android Studio*, struktur proyek akan muncul bersama dengan hampir semua berkas yang ada di dalam direktori *SDK*, peralihan ke sistem manajemen berbasis *Gradle* ini memberikan fleksibilitas yang lebih besar pada proses pembangunanya.

Android Studio mengijinkan Anda untuk melihat perubahan visual apapun yang Anda lakukan pada aplikasi secara langsung. Anda juga bisa melihat perbedaannya jika dipasang pada beberapa perangkat Android berbeda, termasuk konfigurasi dan resolusinya secara bersamaan.

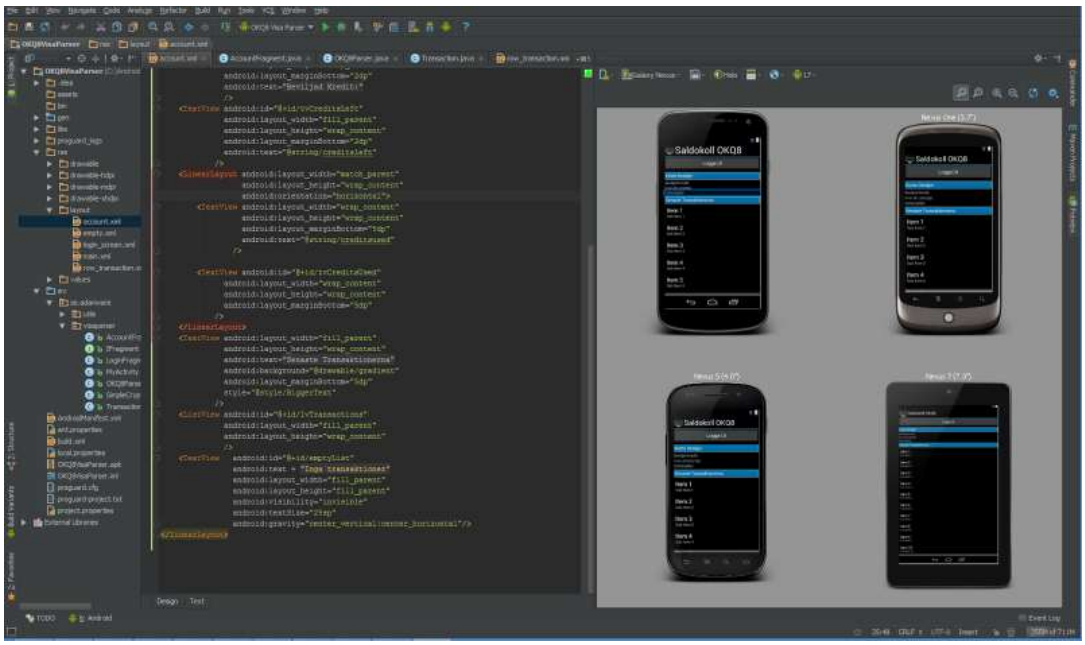
Fitur lain di *Android Studio* adalah alat-alat baru untuk mengepak dan memberi label kode. Dengan begitu mengijinkan Anda tatap menjadi yang teratas ketika berurusan dengan

banyak kode. Program ini juga memakai sistem seret dan jatuhkan untuk memindahkan komponen melalui antar muka pengguna.

Selain itu, lingkungan baru ini juga mendukung *Google Cloud Messaging*. Sebuah fitur yang mengijinkan Anda untuk mengirim data dari server ke perangkat Android Anda melalui *cloud*, cara terbaik untuk mengirim Pengingat pada apps anda.

Program ini juga membantu Anda untuk melokalisasi aplikasi anda, memberi anda gambaran visual untuk tetap memprogram sambil mengontrol alur dari aplikasi.

Android Studio merupakan sebuah *IDE (Integrated Development Environment)*, aplikasi untuk pengembangan software, dikhususkan untuk pembuatan aplikasi Android yang dikenalkan pada Google I/O Mei 2013 (Gerber, A. & Craig, C; 2015).



Gambar II.4. TampilanSoftware *Android Studio*

(http://google.com)

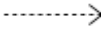
II.11. Unified Modeling Language (UML)

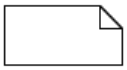
UML yang merupakan singkatan dari *Unified Modelling Language* adalah sekumpulan pemodelan konvensi yang digunakan untuk menentukan atau menggambarkan sebuah sistem perangkat lunak dalam kaitannya dengan objek. UML dapat juga diartikan sebuah bahasa grafik standar yang digunakan untuk memodelkan perangkat lunak berbasis objek. UML pertama kali dikembangkan pada pertengahaan tahun 1990an dengan kerjasama antara James Rumbaugh, Grady Booch dan Ivar Jacobson, yang masing-masing telah mengembangkan notasi mereka sendiri di awal tahun 1990an (Lethbride dan Leganiere, 2009:11).

II.11.1. *Use Case Diagram*

Use case diagram, adalah sebuah gambaran dari fungsi sistem yang dipandang dari sudut pandang pemakai. *Actor* adalah segala sesuatu yang perlu berinteraksi dengan sistem untuk pertukaran informasi. *Systemboundary* menunjukkan cakupan dari sistem yang dibuat dan fungsi dari sistem tersebut (Lethbride dan Leganiere;2009:11).

Tabel II.1. Simbol *Use Case Diagram*

NO	GAMBAR	NAMA	KETERANGAN
1		*Actor*	Menspesifikasikan himpuan peran yang pengguna mainkan ketika berinteraksi dengan *use case*.
2		*Dependency*	Hubungan dimana perubahan yang terjadi pada suatu elemen mandiri *(independent)* akan mempengaruhi elemen yang bergantung padanya elemen yang tidak mandiri (*independent*).
3		*Generalization*	Hubungan dimana objek anak (*descendent*) berbagi perilaku dan struktur data dari objek yang ada di atasnya objek induk (*ancestor*).
4		*Include*	Menspesifikasikan bahwa *use case* sumber secara *eksplisit*.

5		*Extend*	Menspesifikasikan bahwa *use case* target memperluas perilaku dari *use case* sumber pada suatu titik yang diberikan.
6		*Association*	Apa yang menghubungkan antara objek satu dengan objek lainnya.
7		*System*	Menspesifikasikan paket yang menampilkan sistem secara terbatas.
8		*Use Case*	Deskripsi dari urutan aksi-aksi yang ditampilkan sistem yang menghasilkan suatu hasil yang terukur bagi suatu aktor
9		*Collaboration*	Interaksi aturan-aturan dan elemen lain yang bekerja sama untuk menyediakan prilaku yang lebih besar dari jumlah dan elemen-elemennya (sinergi).
10		*Note*	Elemen fisik yang eksis saat aplikasi dijalankan dan mencerminkan suatu sumber daya komputasi

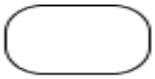
(Lethbride dan Leganiere;2009:11)

II.11.2. *Activity Diagram*

Activity Diagram menggambarkan berbagai alir aktivitas dalam sistem yang sedang dirancang, bagaimana masing – masing alir berawal, *decision* yang mungkin terjadi, dan bagaimana mereka berakhir. *Activity Diagram* juga dapat menggambarkan proses paralel yang mungkin terjadi pada beberapa eksekusi. *Activity Diagram* merupakan *state* diagram khusus, di mana sebagian besar *state* adalah *action* dan sebagian besar *transisi di-trigger* oleh selesainya *state* sebelumnya (*internal processing*). Oleh karena itu *Activity Diagram* tidak menggambarkan *behaviour* internal sebuah sistem (dan interaksi antar subsistem) secara eksak, tetapi lebih

menggambarkan proses – proses dan jalur – jalur aktivitas dari level atas secara umum. Menggambarkan proses bisnis dan urutan aktivitas dalam sebuah proses. Dipakai pada *business modeling* untuk memperlihatkan urutan aktifitas proses bisnis. Struktur diagram ini mirip *flowchart* atau Data *Flow* Diagram pada perancangan terstruktur. *Activity Diagram* dibuat berdasarkan sebuah atau beberapa *use case* pada *use case* diagram (Lethbride dan Leganiere;2009:13).

Tabel II.2. Simbol *Activity Diagram*

NO	GAMBAR	NAMA	KETERANGAN
1		*Actifity*	Memperlihatkan bagaimana masing-masing kelas antarmuka saling berinteraksi satu sama lain
2		*Action*	State dari sistem yang mencerminkan eksekusi dari suatu aksi
3		*Initial Node*	Bagaimana objek dibentuk atau diawali.
4		*Actifity Final Node*	Bagaimana objek dibentuk dan dihancurkan
5		*Fork Node*	Satu aliran yang pada tahap tertentu berubah menjadi beberapa aliran

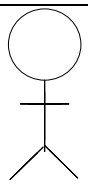
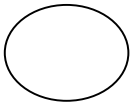
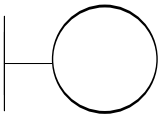
(Lethbride dan Leganiere;2009:15)

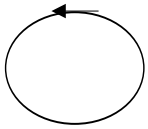
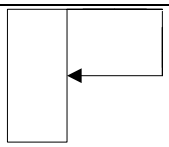
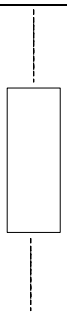
II.11.3. *Sequence diagram*

Sequence diagram menambahkan dimensi waktu pada interaksi diantara obyek. Pada diagram ini participant diletakkan di atas dan waktu ditunjukkan dari atas ke bawah. *Life line participant* diurutkan dari setiap paricipant. Kotak kecil pada life line menyatakan *activation* : yaitu menjalankan salah satu operation dari participant. Satate bisa ditambahkan dengan

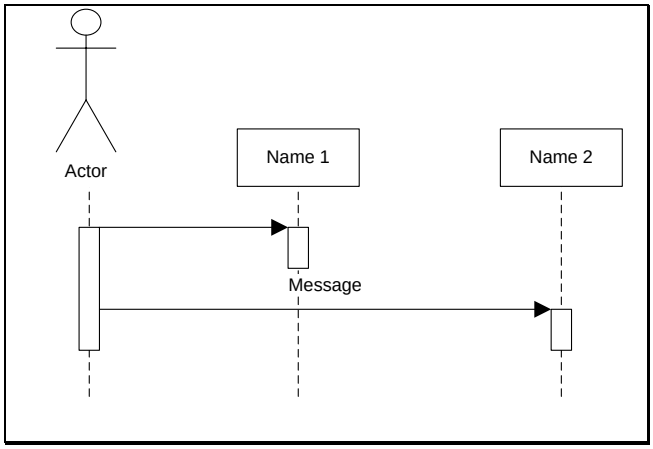
menempatkannya sepanjang life line. *Message* (sederhana, synchronous atau *asynchroneous*) adalah tanda panah yang menghubungkan suatu *life line* ke *life line* yang lain. Lokasi *life line* dalam dimensi vertikal mewakili urutan waktu dalam sequence diagram. Message yang pertama terjadi adalah yang paling dekat dengan bagian atas diagram dan yang terjadi belakangan adalah yang dekat dengan bagian bawah. Pada beberapa sistem, operasi bisa dilakukan kepada dirinya sendiri. Hal ini disebut dengan rekursif. Untuk melukiskannya digunakan anak panah dar activation kembali ke dirinya sendiri, dan sebuah kotak kecil diletakkan pada bagian atas dari *activation*.

Tabel II.3 Simbol *Sequence Diagram*

Gambar	Keterangan
	*Actor* dapat berupa manusia, sistem atau *device* yang memliki peranan dalam keberhasilan operasi dari sistem.
	*Entity Class* merupakan bagian dari sistem yang berisi kumpulan kelas berupa entitas-entitas yang membentuk gambaran awal sistem dan menjadi landasan untuk menyusun basis data
	*Boundary Class* berisi kumpulan kelas yang menjadi *interface* atau interaksi antara satu atau lebih actor dengan sistem.

	*Control Class* suatu objek yang berisi logika aplikasi yang tidak memiliki tanggung jawab kepada entitas.
	Message simbol mengirim pesan antar *class*.
	*Recursive* menggambarkan pengiriman pesan yang dikirim untuk dirinya sendiri.
	*Activation* mewakili sebuah eksekusi operasi dari objek, panjang kotak ini berbanding lurus dengan durasi aktivitas sebuah operasi .
	Lifeline yaitu garis titik titik yang terhubung dengan objek, sepanjang *lifeline* terdapat *activation.*

(Gellysa Urva dkk; 2015 : 95)



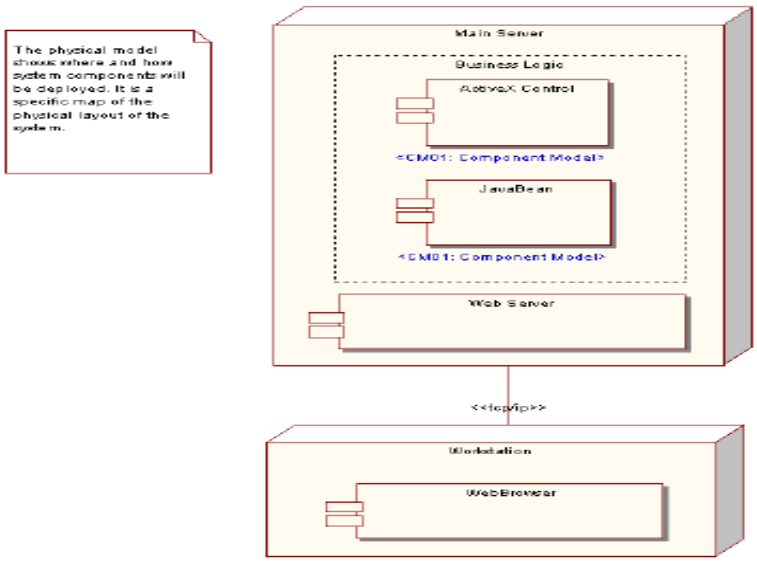
Gambar II.5. Simbol-simbol yang ada pada *sequence diagram*

(Agus Putranto;2009:14)

II.11.4. *Deployment diagram*

Deployment diagram menggambarkan sumber fisik dalam sistem, termasuk node, komponen dan koneksi (model implementasi sistem yang statistik). Dalam hal ini meliputi topologi *hardware* yang dipakai sistem.

Deployment/physical diagram menggambarkan detail bagaimana komponen di-*deploy* dalam infrastruktur sistem, di mana komponen akan terletak (pada mesin, server atau piranti keras apa), bagaimana kemampuan jaringan pada lokasi tersebut, spesifikasi server, dan hal – hal lain yang bersifat fisikal. Sebuah *node* adalah server, *workstation*, atau piranti keras lain yang digunakan untuk men-*deploy* komponen dalam lingkungan sebenarnya. Hubungan antar *node* (misalnya TCP/IP) dan *requirement* dapat juga didefinisikan dalam diagram ini.



Gambar II.6. Simbol-simbol yang ada pada *Deployment* Diagram

(Agus Putranto;2009:9)