

BAB II

TINJAUAN PUSTAKA

II.1. *Game*

Ada yang mengatakan *game* itu mainan, ada pula yang bilang *game* adalah *film* yang bisa dikendalikan, dan ada yang berkata *game* itu adalah hiburan yang penuh dengan teka-teki. Apa pun pendapat mereka, itu semua hanyalah sebagian dari pengertian *game* yang sebenarnya. *Game* bukan mainan / *film* / teka-teki, walaupun terdapat unsur-unsur tersebut didalamnya. *Game* adalah suatu sistem atau program di mana satu atau lebih pemain mengambil keputusan melalui kendali pada obyek di dalam *game* untuk suatu tujuan tertentu.

Game pada awal tahun 1970-an merupakan awal dari perkembangan *game* dunia, dimana *game* yang dibuat terdiri atas teks dan grafik standar. Beberapa *game* pada awal perkembangannya antara lain :

1. Teks *Game*, jenis *game* ini hanya terdiri atas teks, tidak ada grafik apa pun dalam *game* ini. *User* berinteraksi dengan mengetikan teks.
2. Grafik *Game*. Jenis *game* ini menggunakan grafik meskipun masih dalam bentuk grafik standar yaitu VGA, SVGA, dan lainnya.

II.1.1. Tujuan Pembuatan *Game*

Game dibuat untuk suatu tujuan tertentu, biasanya pembuat *game* memiliki suatu tujuan khusus dalam membuat sebuah *game*. Tujuan pembuatan sebuah *game* antara lain :

1. *Entertainment* (sebagai hiburan)
2. *Expand Skil* (melatih ketangkasan)
3. *Education* (mendidik)
4. *Embed Messages* (menyampaikan pesan tertentu)

II.1.2. Definisi Permainan Tebak Posisi

Definisi tebak posisi adalah posisi yang digunakan khusus untuk Menyusun strategi terbaik dan mempelajari setiap strategi untuk menghancurkan strategi lawan.

II.2. Komunikasi Data

Kata komunikasi merupakan suatu kata yang dapat diartikan sebagai cara untuk menyampaikan atau menyebarluaskan data dan informasi, sedangkan kata informasi berarti berita, pikiran, pendapat dalam berbagai bentuk. Manusia dapat melakukan komunikasi dengan berbagai cara, berbicara secara langsung, berbisik, mengirim surat dan lain sebagainya. (Ariyus, 2008:3)

Dari berbagai cara komunikasi manusia ini masih terdapat banyak kekurangan dan kelemahan yaitu :

1. Jarak yang jauh (bahkan sampai menyeberangi lautan),
2. Waktu yang lama untuk menyampaikan pesan.
3. Biaya yang relatif mahal.

Kekurangan tersebut bisa diatasi seiring dengan perkembangan teknologi informasi. Dengan teknologi komunikasi sekarang ini, hanya dengan menekan beberapa tombol maka jarak dan waktu untuk melakukan komunikasi tidak lagi

menjadi kendala yang berarti. (Ariyus, 2008:3)

Teknologi komunikasi terus dikembangkan dengan tujuan memudahkan manusia dalam melakukan komunikasi. Para ahli terdorong untuk mengembangkan teknik komunikasi jarak jauh yang lebih efisien dengan metode telekomunikasi yang memanfaatkan teknologi elektronika, yang dikenal dengan istilah teknik komunikasi data. (Ariyus, 2008:3)

Komunikasi data merupakan cara mengirimkan data menggunakan sistem transmisi elektronik dari satu komputer ke komputer lain atau dari satu komputer ke terminal tertentu. Sedangkan data itu sendiri merupakan sinyal elektromagnetik yang dibangkitkan oleh sumber data yang ditangkap dan dikirimkan ke terminal penerima. (Ariyus, 2008:3)

II.3. Jaringan Komputer

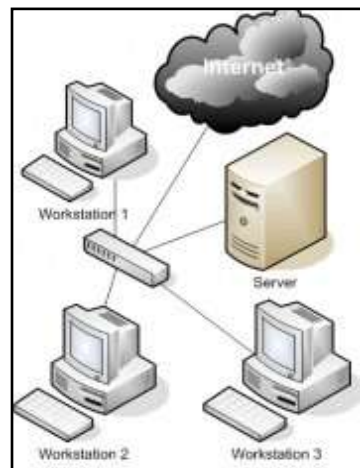
Jaringan komputer (*Computer Network*) dapat diartikan sebagai sekelompok komputer yang dihubungkan menggunakan media tertentu sehingga antar komputer dapat saling berhubungan untuk berbagai data, informasi, program aplikasi dan perangkat keras, seperti *printer*, *scanner*, *CD/DVD Drive*, ataupun *hardisk* : (Madcoms, 2013:1)

Sekelompok komputer dan *device* lain yang saling terhubung membentuk sebuah *network*, sedangkan konsep dari komputer-komputer yang saling berbagi *resource* dikenal dengan istilah *networking*. Komputer-komputer yang termasuk ke dalam sebuah jaringan dapat saling berbagi *resource* berupa pesan, *printer*, mesin *fax*, modem, dan *device-device* lain. (Arifin, 2005:02)

Berdasarkan lokasi geografis dan ukurannya, jaringan terbagi ke dalam beberapa jenis, diantaranya :

1. LAN (*Local Area Network*)

LAN (*Local Area Network*) merupakan sebuah kumpulan komputer yang terdapat pada sebuah lokasi dan saling terhubung, misalkan jaringan yang terdapat dalam sebuah gedung. Secara administrasi dapat dilakukan sendiri oleh seseorang atau beberapa *administrator* (orang yang bertugas dan bertanggung jawab memelihara jaringan agar tetap berjalan secara normal).



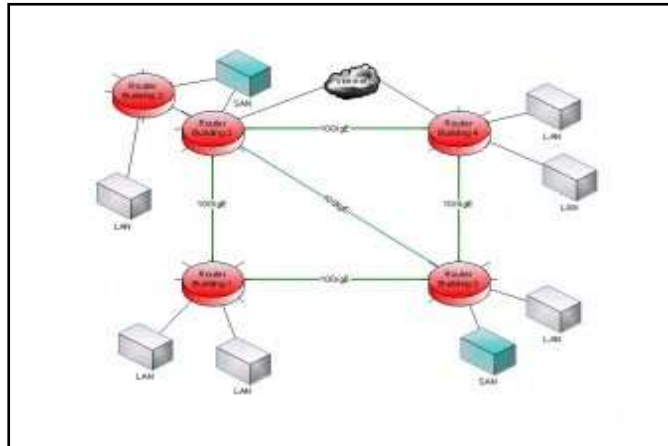
Gambar II.1. LAN (*Local Area Network*)

Sumber : (Rifki, Yosafine, 2011:4)

2. MAN (*Metropolitan Area Network*)

MAN (*Metropolitan Area Network*) biasanya meliputi area yang lebih besar dari LAN, misalnya antar wilayah dalam satu propinsi. Dalam hal ini jaringan menghubungkan beberapa buah jaringan-jaringan kecil ke dalam lingkungan area yang lebih besar, sebagai contoh yaitu jaringan Bank

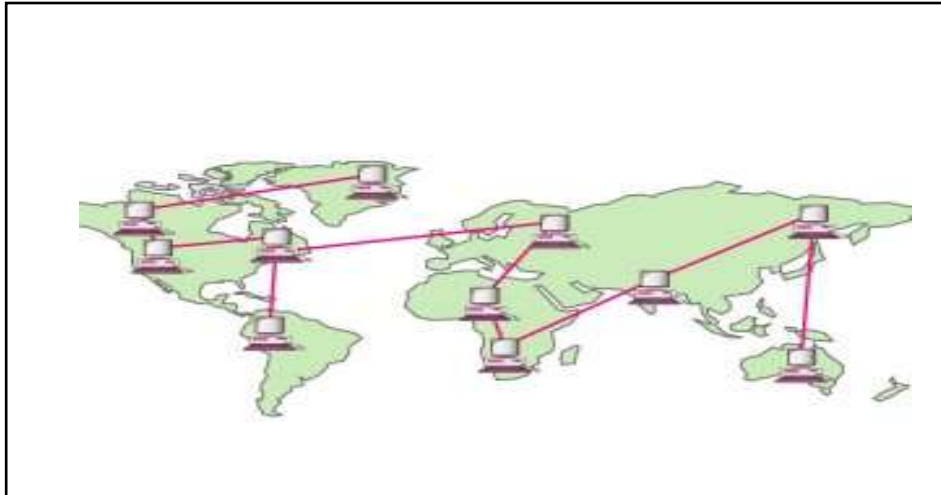
dimana beberapa kantor cabang sebuah Bank di dalam sebuah kota besar dihubungkan antara satu dengan lainnya.



Gambar II.2. MAN (Metropolitan Area Network)
 Sumber : (Rifki, Yosafine, 2011:4)

3. WAN (Wide Area Network)

WAN (*Wide Area Network*) Merupakan sistem jaringan yang saling menghubungkan beberapa LAN dan tidak tergantung pada sebuah lokasi, misalkan jaringan *internet*. Secara administrasi jaringan WAN membutuhkan pihak ketiga (*network provider*) untuk menyediakan media penghubung yang mampu menghubungkan jaringan-jaringan yang berbeda lokasi itu.

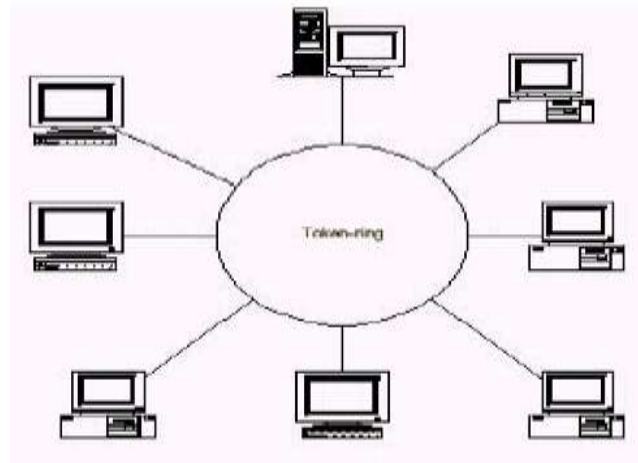


Gambar II.3. WAN (Wide Area Network)
 Sumber : (Rifki, Yosafine, 2011:5)

II.4. Topologi Jaringan Komputer

Topologi adalah suatu cara menghubungkan komputer yang satu dengan komputer lainnya sehingga membentuk jaringan. Cara yang saat ini banyak digunakan adalah *Bus*, *Token Ring*, dan *Star Network*. Masing-masing topologi ini mempunyai ciri khas, dengan kelebihan dan kekurangannya sendiri.

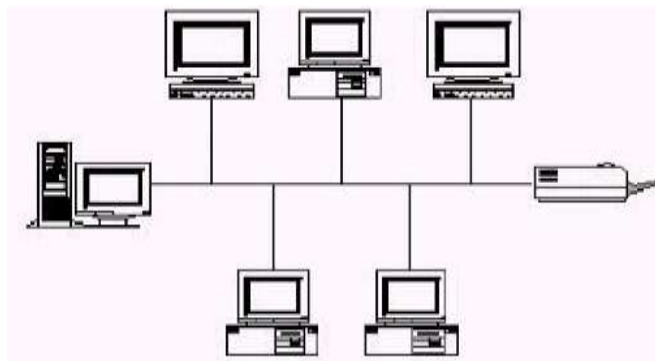
1. Topologi *Token Ring*, Di dalam topologi *Ring* semua *workstation* dan *server* dihubungkan sehingga terbentuk suatu pola lingkaran atau cincin. Tiap *workstation* ataupun *server* akan menerima dan melewatkan informasi dari satu komputer ke komputer lain, bila alamat-alamat yang dimaksud sesuai maka informasi diterima dan bila tidak informasi akan dilewatkan, yang dapat dilihat pada gambar II.4.



Gambar II.4. Topologi Token Ring

Sumber : (Wahidin, 2007:16)

2. Topologi Bus, Pada topologi Bus digunakan sebuah kabel tunggal atau kabel pusat dimana seluruh *workstation* dan *server* dihubungkan, dapat dilihat pada gambar II.5.

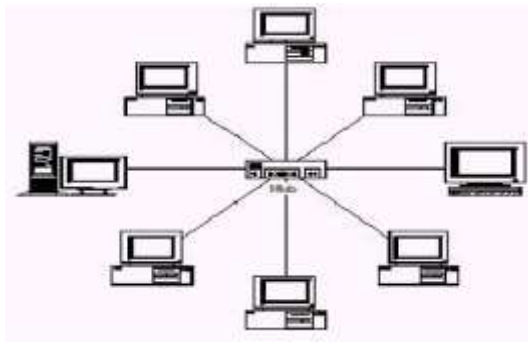


Gambar II.5. Topologi Jaringan Bus

Sumber : (Wahidin, 2007:16)

3. Topologi *Star*, Pada topologi *Star*, masing-masing *workstation* dihubungkan secara langsung ke *server* atau HUB. Keunggulan dari topologi tipe *Star* ini adalah bahwa dengan adanya kabel tersendiri untuk setiap *workstation* ke *server*, maka *bandwidth* atau lebar jalur komunikasi dalam kabel akan semakin lebar sehingga akan meningkatkan unjuk kerja jaringan secara

keseluruhan. Dan juga bila terdapat gangguan di suatu jalur kabel maka gangguan hanya akan terjadi dalam komunikasi antara *workstation* yang bersangkutan dengan *server*, dapat dilihat pada gambar II.6.



Gambar II.6. Topologi Jaringan Star
 Sumber: (Wahidin, 2007:16)

II.5. Jaringan Client Server

Jaringan *Client Server* menghubungkan komputer *server* dengan komputer klien/*workstation*. Komputer *server* adalah komputer yang menyediakan fasilitas bagi komputer-komputer klien/*workstation* yang terhubung dalam jaringan. Sedangkan komputer adalah komputer yang menggunakan fasilitas yang disediakan oleh komputer *server*. Komputer server pada sebuah jaringan tipe *client server* disebut dengan *dedicated server*. Karena komputer yang digunakan hanya sebagian penyedia fasilitas untuk komputer klien/*workstation*. Komputer *server* tidak dapat berperan sebagai komputer klien/*workstation*. (Madcoms, 2010:3)

Keunggulan tipe jaringan *client server* adalah sebagai berikut :

1. Terdapat administrator jaringan yang mengelola sistem keamanan dan administrasi jaringan, sehingga sistem keamanan dan administrasi jaringan akan lebih terkontrol.

2. Komputer *server* difungsikan sebagai pusat data, komputer klien dapat mengakses data yang ada dari komputer klien manapun. Apabila terdapat komputer klien yang rusak, pengguna masih dapat mengakses data dari komputer klien yang lain.
3. Pengaksesan data lebih tinggi karena penyediaan dan pengelolaan fasilitas jaringan dilakukan oleh komputer *server*. Dan komputer *server* tidak terbebani dengan tugas lain sebagai *workstation*.
4. Pada tipe jaringan *Client Server*, sistem *backup* data lebih baik, karena backup data dapat dilakukan terpusat di komputer *server*. Apabila data pada komputer klien/*workstation* mengalami masalah atau kerusakan masih tersedia *backup* pada komputer *server*.

Kelemahan tipe jaringan *Client server* adalah sebagai berikut :

1. Biaya mahal, karena membutuhkan komputer yang memiliki kemampuan yang difungsikan sebagai komputer server.
2. Kelancaran jaringan tergantung pada komputer server. Bila komputer server mengalami gangguan maka jaringan akan terganggu.

II.6. Protokol Komputer

Protokol Komputer adalah sebuah aturan atau standar yang mengatur atau mengizinkan terjadinya hubungan, komunikasi, dan perpindahan data antara dua atau lebih titik komputer. Protokol dapat diterapkan pada perangkat keras, perangkat lunak atau kombinasi dari keduanya. Pada tingkatan yang terendah, protokol mendefinisikan koneksi perangkat keras.

Protokol perlu diutamakan pada penggunaan standar teknis, untuk menspesifikasi bagaimana membangun komputer atau menghubungkan peralatan perangkat keras. Protokol secara umum digunakan pada komunikasi *real-time* dimana standar digunakan untuk mengatur struktur dari informasi untuk penyimpanan jangka panjang.

Sangat susah untuk *menggeneralisir* protokol dikarenakan protokol memiliki banyak variasi didalam tujuan penggunaanya. Kebanyakan protokol memiliki salah satu atau beberapa dari hal berikut:

1. Melakukan deteksi adanya koneksi fisik atau ada tidaknya komputer atau mesin lainnya.
2. Melakukan metoda “jabat-tangan” (*handshaking*).
3. Negosiasi berbagai masam karakteristik hubungan.
4. Bagaimana mengawali dan mengakhiri suatu pesan.
5. Bagaimana format pesan yang digunakan.
6. Yang harus dilakukan saat terjadi kerusakan pesan atau pesan yang tidak sempurna.
7. Meng akhiri suatu koneksi.

II.7. Alamat IP

Alamat IP (*Internet Protocol Address* atau sering disingkat IP) adalah deretan angka biner antar 32-bit sampai 128-bit yang dipakai sebagai alamat identifikasi untuk tiap komputer host dalam jaringan *Internet*. Panjang dari angka ini adalah *32-bit* (untuk IPv4 atau IP versi 4), dan *128-bit* (untuk IPv6 atau IP

versi 6) yang menunjukkan alamat dari komputer tersebut pada jaringan *Internet* berbasis TCP/IP. Sistem pengalamatan IP ini terbagi menjadi dua, yakni:

1. IP versi 4 (IPv4)
2. IP versi 6 (IPv6)

Berikut ini menjelaskan perbandingan karakteristik antara alamat IP versi 4 dan alamat IP versi 6 yang dapat dilihat pada tabel II.1 di bawah ini.

Tabel.II.1. Karakteristik Alamat IP

CRITERIA	IP ADDRESS VERSI 4	IP ADDRESS VERSI 6
Panjang alamat	32 bit	128 bit
Jumlah total <i>host</i> (teoritis)	$2^{32} = \pm 4$ miliar <i>host</i>	$2^{128} = 3,4 \times 10^{38}$ <i>host</i>
Menggunakan kelas alamat	Ya, kelas A, B, C, D, dan E. Belakangan tidak digunakan lagi, mengingat telah tidak relevan dengan perkembangan.	Tidak
Alamat <i>multicast</i>	Kelas D, yaitu 224.0.0.0/4	Alamat <i>multicast</i> IPv6, yaitu FF00:/8
Alamat <i>broadcast</i>	Ada	Tidak ada
Alamat yang belum ditentukan	0.0.0.0	::
Alamat <i>loopback</i>	127.0.0.1	::1
Alamat IP public	Alamat IP publik IPv4, yang ditetapkan oleh otoritas Internet (IANA)	Alamat IPv6 <i>unicast global</i>
Alamat IP pribadi	Alamat IP pribadi IPv4, yang ditetapkan oleh otoritas <i>Internet</i>	Alamat IPv6 <i>unicast site-local</i> (FEC0::/48)
Konfigurasi alamat otomatis	Ya (APIPA)	Alamat IPv6 <i>unicast link-local</i> (FE80::/64)
Representasi tekstual	<i>Dotted decimal format notation</i>	<i>Colon hexadecimal format notation</i>
Fungsi Prefiks	<i>Subnet mask</i> atau panjang <i>prefix</i>	Panjang prefiks
Resolusi alamat DNS	<i>A Resource Record (Single A)</i>	<i>AAAA Resource Record (Quad A)</i>

Sumber : (Wahidin, 2007:16)

II.8. Jaringan Wi-Fi

Jaringan *Wireless* atau jaringan nirkabel yang sering disebut WLAN/*Wireless* LAN merupakan jaringan LAN biasa yang tidak menggunakan kabel untuk transfer datanya. Media untuk transfer data adalah gelombang radio. Jaringan ini cakupannya bervariasi, bisa sekedar mencakup satu area kecil, ataupun area yang sangat luas. (Winarno, 2013:105)

Pada umumnya jaringan *wireless* ini dikombinasikan juga dengan jaringan kabel. Terutama untuk *backbone* biasanya tetap menggunakan kabel. Selain itu untuk menghubungkan satu jaringan *wireless* ke jaringan *wireless* lain, juga digunakan jaringan kabel. Ini karena jaringan kabel lebih stabil dalam transfer datanya. (Winarno, 2013:105)

II.8.1. Jaringan Ad-Hoc

Di jaringan komputer *wireless*, mode *ad-hoc* adalah metode yang memungkinkan piranti *wireless* yang berada di jangkauan sinyal untuk mencari dan berkomunikasi secara *peer to peer* tanpa melibatkan *access point*. Untuk bisa menerapkan *ad-hoc*, tiap adaptor *wireless* harus diset ke mode *ad-hoc* dan bukannya di mode infrastruktur. Selain itu semua adaptor *wireless* (*Wifi*) harus diset menggunakan satu nama SSID dan *channel* yang sama. (Winarno, 2013:114)

II.9. Aplikasi

Perangkat lunak aplikasi adalah suatu subkelas perangkat lunak komputer yang memanfaatkan kemampuan komputer langsung untuk melakukan suatu tugas yang diinginkan pengguna. Biasanya dibandingkan dengan perangkat lunak

sistem yang mengintegrasikan berbagai kemampuan komputer, tapi tidak secara langsung menerapkan kemampuan tersebut untuk mengerjakan suatu tugas yang menguntungkan pengguna. Contoh utama perangkat lunak aplikasi adalah pengolah kata, lembar kerja, dan pemutar media.

Beberapa aplikasi yang digabung bersama menjadi suatu paket kadang disebut sebagai suatu paket atau *suite* aplikasi (*application suite*). Contohnya adalah *Microsoft Office* dan *OpenOffice.org*, yang menggabungkan suatu aplikasi pengolah kata, lembar kerja, serta beberapa aplikasi lainnya. Aplikasi-aplikasi dalam suatu paket biasanya memiliki antarmuka pengguna yang memiliki kesamaan sehingga memudahkan pengguna untuk mempelajari dan menggunakan tiap aplikasi. Sering kali, mereka memiliki kemampuan untuk saling berinteraksi satu sama lain sehingga menguntungkan pengguna. Contohnya, suatu lembar kerja dapat dibenamkan dalam suatu dokumen pengolah kata walaupun dibuat pada aplikasi lembar kerja yang terpisah.

II.10. UML

Unified Modelling Language (UML) merupakan kesatuan dari bahasa pemodelan yang dikembangkan oleh *Booch*, *Object Modeling Technique* (OMT) dan *Object Oriented Software Engineering* (OOSE). Metode *Booch* dari *Grady Booch* sangat terkenal dengan nama metode *Design Object Oriented*. Metode ini menjadikan proses analisis dan design ke dalam empat tahapan iteratif, yaitu : identifikasi kelas-kelas dan obyek-obyek, identifikasi semantik dari hubungan obyek dan kelas tersebut, perincian *interface* dan implementasi. keunggulan metode *Booch* adalah pada detail dan kayanya dengan notasi dan elemen.

Pemodelan OMT yang dikembangkan oleh *Rumbaugh* didasarkan pada analisis terstruktur dan pemodelan *entity relationship*. Tahapan utama dalam metodologi ini adalah analisis, *design* sistem, *design* obyek dan implementasi. Keunggulan metode ini adalah dalam penotasian yang mendukung semua konsep OO. Metode OOSE dari *Jacobson* lebih memberi penekanan pada *use case*. OOSE memiliki tiga tahapan yaitu membuat model *requirement* dan analisis, *design* dan implementasi, dan model pengujian (*test model*). Keunggulan metode ini adalah mudah dipelajari karena memiliki notasi yang sederhana namun mencakup seluruh tahapan dalam rekayasa perangkat lunak.

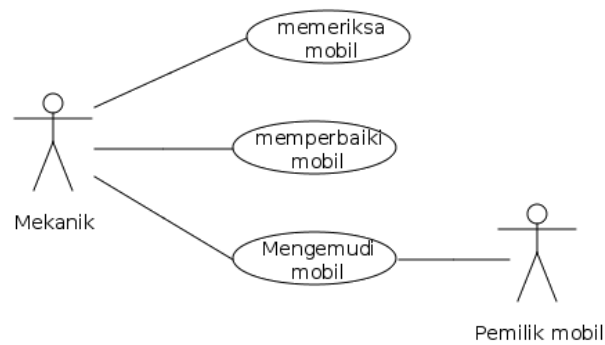
Adapun diagram-diagram UML yang digunakan dalam perancangan, diantaranya *Use Case*, *Activity Diagram* dan *Class Diagram*.

II.11. *Use Case Diagram*

Use case diagram menggambarkan fungsionalitas yang diharapkan dari sebuah sistem. *Use case* merepresentasikan sebuah interaksi antara aktor dengan sistem. Seseorang / sebuah aktor adalah sebuah entitas manusia atau mesin yang berinteraksi dengan sistem untuk melakukan pekerjaan-pekerjaan tertentu. *Use case* merupakan sebuah diagram aktivitas pekerjaan tertentu, misalnya login ke sistem, meng-*create* sebuah daftar belanja, dan sebagainya.

Use case diagram dapat digunakan untuk :

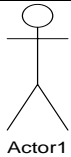


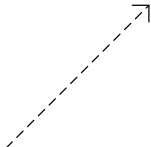
1. Menyusun *requirement* sebuah sistem,
2. Mengkomunikasikan rancangan dengan klien, dan
3. Merancang *test case* untuk semua *feature* yang ada pada sistem.



Gambar II.7. Contoh Use Case Diagram

Sumber : (Munawar, 2005)

Tabel.II.2. Simbol-simbol Use Case Diagram

Simbol	Notasi	Keterangan
 Actor1	<i>Actor</i>	Pengguna sistem atau yang berinteraksi langsung dengan sistem, bisa manusia, aplikasi, ataupun objek lain.
 UseCase1	<i>Use Case</i>	Digambarkan dengan lingkaran elips dengan nama use case-nya tertulis ditengah lingkaran.
	<i>Assotiation</i>	Digambarkan dengan sebuah garis yang berfungsi menghubungkan actor dengan use case.
	<i>Defendency</i>	<i>Depenciency</i> merupakan relasi yang menunjukkan bahwa perubahan pada salah satu elemen memberi pengaruh pada elemen lain.

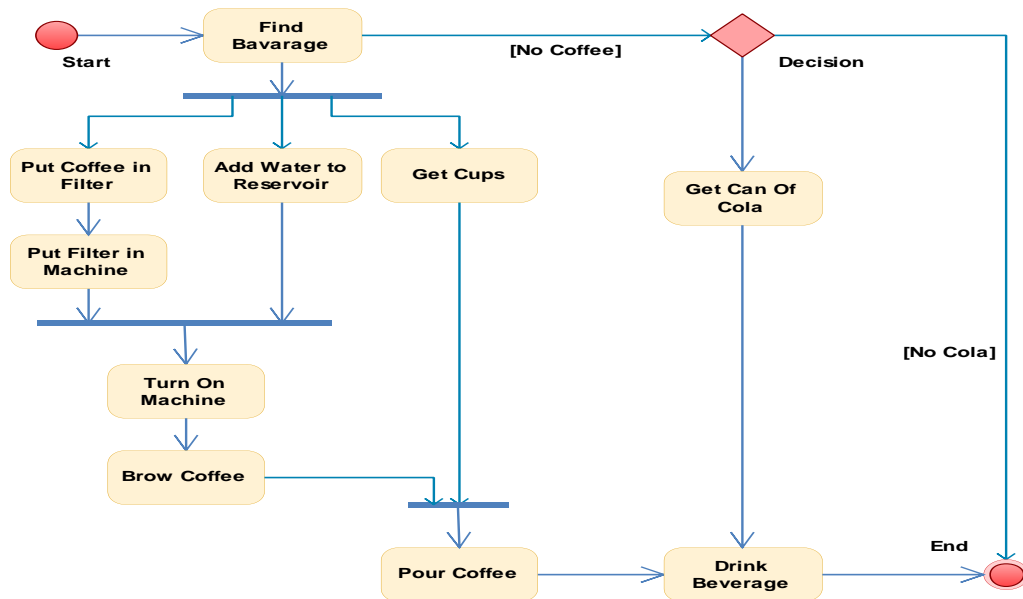
<p><<extended>> -----></p>	<p><i>Extended</i></p>	<p><i>Extended</i> menunjukkan bahwa suatu bagian dari <i>elemen</i> digaris tanpa panah bisa disisipkan kedalam <i>elemen</i> yang ada digaris dengan panah.</p>
---	------------------------	---

Sumber : (Munawar, 2005)

II.12. Activity Diagram

Menggambarkan berbagai alir aktivitas dalam sistem yang sedang dirancang, bagaimana masing-masing alir berawal, *decision* yang mungkin terjadi, dan bagaimana mereka berakhir.

1. *Activity diagram* juga dapat menggambarkan proses paralel yang mungkin terjadi pada beberapa eksekusi.
2. *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*).
3. *Activity diagram* dapat dibagi menjadi beberapa *object swimlane* untuk menggambarkan objek mana yang bertanggung jawab untuk aktivitas tertentu.



Gambar II.8. Contoh Activity Diagram

Sumber : (Munawar, 2005)

Tabel.II.3. Simbol-simbol Activity Diagram

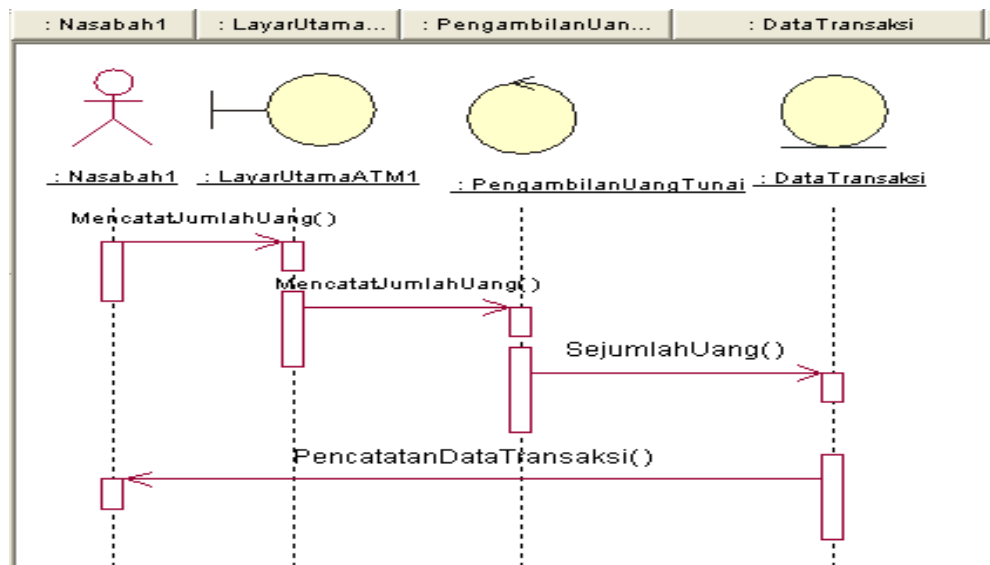
Simbol	Notasi	Keterangan
●	<i>Initial state</i> memulai	Titik awal untuk suatu aktivitas
◉	<i>Final State</i>	Titik akhir untuk mengakhiri suatu aktivitas
◇	<i>Decision</i>	Pilihan untuk pengambilan keputusan
⬭	<i>Activity</i>	Menandakan sebuah aktivitas
—	<i>Transition</i>	Menandakan proses transisi dari sebuah aktivitas

Sumber : (Munawar, 2005)

II.13. Sequence Diagram

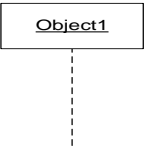



Sequency diagram menjelaskan secara detail tentang urutan proses yang dilakukan dalam sistem untuk mencapai tujuan dari use case : interaksi yang

terjadi antar class, operasi apa saja yang terlibat, urutan antar operasi, dan informasi yang diperlukan oleh masing-masing operasi. *Sequency diagram* biasanya tersusun dari elemen obyek, *interaction* dan *message*. *Interaction* menghubungkan 2 obyek dengan pesannya. Diagram ini menjelaskan aspek dinamis dari sistem yang sedang dibangun.



Gambar II.9. Contoh Sequence Diagram
 Sumber : (Munawar, 2005)

Tabel.II.4. Simbol-simbol *Sequence Diagram*

Simbol	Notasi	Keterangan
	<i>Object Lifeline</i>	<i>Object</i> adalah <i>instance</i> dari sebuah <i>class</i> yang dituliskan tersusun secara horizontal diikuti <i>lifeline</i>
	<i>Activation</i>	Indikasi dari sebuah objek yang melakukan suatu aksi
	<i>Message</i>	Indikasi untuk komunikasi antar <i>object</i>
	<i>Self-Message</i>	Komunikasi kembali kedalam <i>object</i> itu sendiri

Sumber : (Munawar, 2005)

II.14. Pembahasan Umum Java

Java menurut definisi dari *Sun* adalah nama untuk sekumpulan teknologi untuk membuat dan menjalankan perangkat lunak pada komputer *standalone* ataupun pada lingkungan jaringan. *Java2* adalah generasi kedua dari *Java platform* (generasi awalnya adalah *Java Development Kit*). *Java* berdiri atas sebuah mesin *interpreter* yang diberi nama *Java Virtual Machine (JVM)*. *JVM* inilah yang akan membaca *bytecode* dalam *file .class* dari suatu program sebagai representasi langsung program yang berisi bahasa mesin. Oleh karena itu, bahasa *Java* disebut sebagai bahasa pemrograman *portable* karena dapat dijalankan pada

berbagai sistem operasi, asalkan pada sistem operasi tersebut terdapat JVM. (Shalahuddin, 2008:01)

Platform java memiliki 3 (tiga) edisi berbeda, yaitu J2EE (*Java2 Enterprise Edition*), J2ME (*Java2 Micro Edition*), dan J2SE (*Java2 Second Edition*).

II.15. Java2 Standard Edition (J2SE)

J2SE adalah inti dari bahasa pemrograman Java. JDK (*Java Development Kit*) adalah salah satu tool dari J2SE untuk mengompilasi dan menjalankan program Java. Di dalamnya terdapat tool untuk mengompilasi program Java dan JRE. (Shalahuddin, 2008:03)

II.16. Java2 Enterprise Edition (J2EE)

Teknologi *Java* yang satu ini digunakan untuk pengembangan aplikasi-aplikasi *Enterprise*. J2EE meliputi beberapa teknologi pendukung, yaitu *Java Server Page* (JSP), *Java Servlet*, *Java. CORBA*, dan lain-lain. (Fikri, Rijalul, 2005:26)

II.17. Java2 Micro Edition (J2ME)

Java2 Micro Edition atau yang biasa disebut J2ME adalah lingkungan pengembangan yang didesain untuk meletakkan perangkat lunak *Java* pada barang elektronik beserta perangkat pendukungnya. pada J2ME, jika perangkat lunak berfungsi baik pada sebuah perangkat maka belum tentu juga berfungsi baik pada

perangkat yang lainnya. J2ME membawa *Java* ke dunia informasi, komunikasi, dan perangkat komputasi selain perangkat komputer *desktop* yang biasanya lebih kecil dibandingkan perangkat komputer *desktop*. J2ME biasa digunakan pada telepon selular, *pager*, *personal digital assistants* (PDA's) dan sejenisnya. (Shalahuddin , 2008:04)



Gambar II.10. Teknologi Java

Sumber : (JCC-ITB, 2007)

II.18. Netbeans Environment

Netbeans adalah salah satu IDE (*Integrated Development Environment*), yaitu sebuah lingkungan kerja yang digunakan untuk mengembangkan aplikasi dengan berbagai bahasa pemrograman khususnya *Java*. *Netbeans* sendiri dikembangkan oleh pengembang *Java*. Yaitu *Sun Microsystem* yang kini diakuisisi oleh *Oracle*. (Wahana Komputer : 2012 : 13)

II.19. Lingkungan Kerja *Netbeans*

Lingkungan kerja adalah tempat anda melakukan pemrograman, baik menulis kode maupun mendesain tampilan UI. *Netbeans* menyediakan beberapa panel yang membantu anda melakukan pemrograman dengan mudah, antara lain *Project*, *Navigators*, *Palette*, *Properties*, dan jendela editor utama.

1. Panel *Project*

Panel *Project* menampilkan daftar *project* yang telah dibuat dengan *Netbeans* beserta dengan *file-file* dan pengaturan yang dimilikinya. Pada *Project Java ME* terdapat 3 buah *node*, yaitu :

a. *Source Packages*

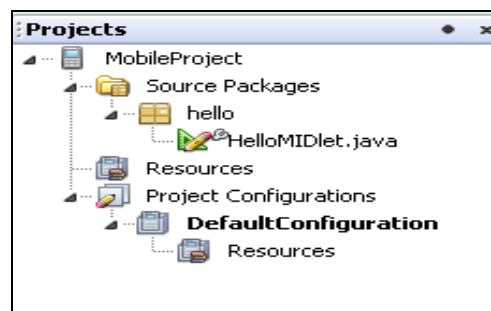
Node ini menampilkan *package* dan *file-file Java* yang telah anda buat.

b. *Resources*

Node ini menampilkan *resource* yang digunakan oleh *project* yang anda buat, misal gambar, suara, *font*, dan lain-lain.

c. *Project Configurations*

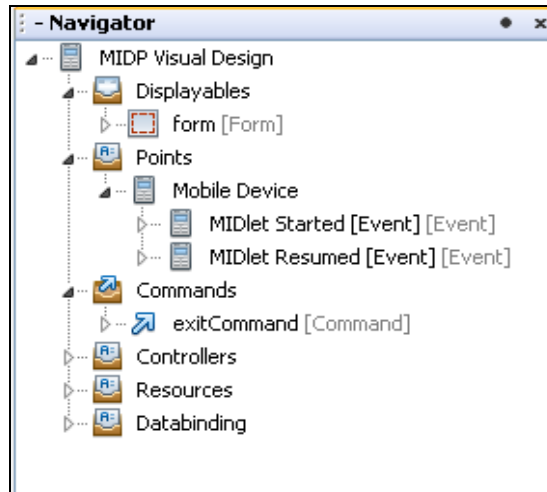
Node ini menampilkan *Configuration* yang digunakan oleh *project*.



Gambar II.11. Panel Project
(Wahana Komputer : 2012:14)

2. Panel *Navigator*

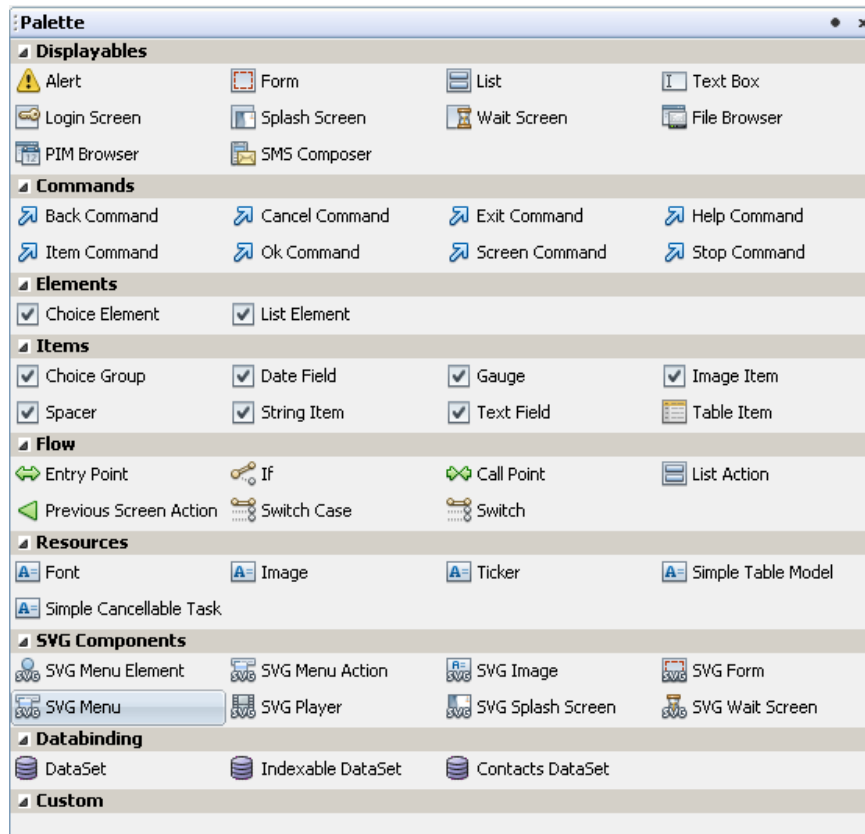
Panel *Navigator* menampilkan komponen/*object* yang telah ditambahkan ke dalam aplikasi yang anda buat. Informasi yang ditampilkan akan berubah sesuai dengan objek yang anda seleksi.



Gambar II.12. Panel Navigator
(*Wahana Komputer : 2012:14*)

3. Panel *Palette*

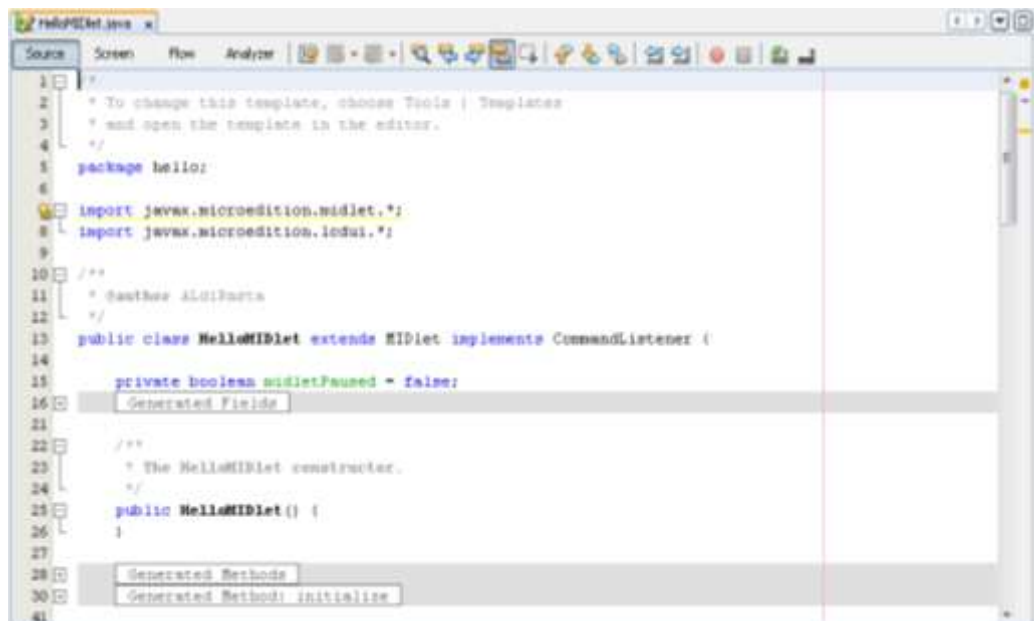
Panel *Palette* adalah panel yang berisikan objek komponen secara visual. Dengan panel ini anda cukup melakukan *Drag* dan *Drop* komponen yang akan anda buat dari *palette* ke dalam panel *editor* desain visual. Terdapat beberapa kategori seperti *Displayables*, *Commands*, *Elements*, *Items*, *Flow*, *Resources*, *SVG Components*, *Databinding*, dan *Custom*.



Gambar II.13. Panel Palette
(Wahana Komputer : 2012:15)

4. Panel Editor

Panel *Editor* menampilkan objek dimana anda melakukan perancangan, baik dari *source*, *screen*, *flow*, dan *analyzer*. Yang dapat dilihat pada gambar berikut ini.



```
1  /*
2   * To change this template, choose Tools | Templates
3   * and open the template in the editor.
4   */
5  package hello;
6
7  import javax.microedition.midlet.*;
8  import javax.microedition.lodui.*;
9
10 /**
11  * @author ALIIBRATA
12  */
13 public class HelloMIDlet extends MIDlet implements CommandListener {
14
15     private boolean midletPaused = false;
16     Generated Fields
17
18
19
20
21
22     /**
23     * The HelloMIDlet constructor.
24     */
25     public HelloMIDlet() {
26     }
27
28     Generated Methods
29     Generated Method: initialize
30
31 }
```

Gambar II.14. Panel Editor
(Wahana Komputer : 2012:16)