

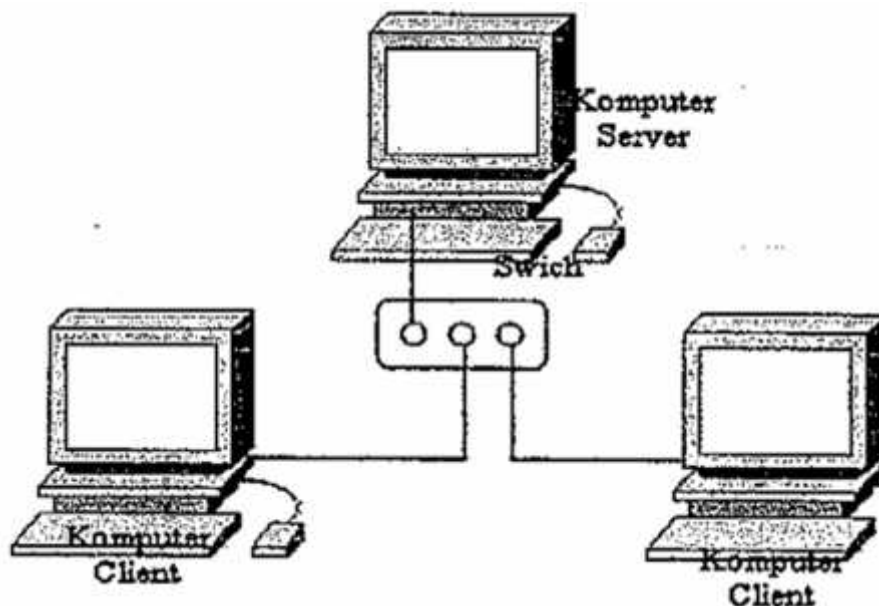
## BAB II

### TINJAUAN PUSTAKA

#### II.1. Tinjauan Umum Remote Desktop

*Remote desktop* adalah salah satu fitur yang terdapat di dalam sistem operasi *Microsoft Windows XP, Windows Server 2003, Windows Vista,* dan *Windows Server 2008,* yang mengizinkan penggunanya untuk terkoneksi ke sebuah mesin jarak jauh seolah-olah mereka duduk di depan mesin yang bersangkutan. *Remote desktop* menggunakan protokol *Remote desktop Protocol (RDP),* dan secara *default* berjalan di *TCP port 3389.* (Kusnendar, 2009:02)

*Remote desktop* adalah proses pengendalian tampilan (*desktop*) dari komputer ke komputer lain. (Helmi, 2009:81). Dapat dilihat pada gambar II.1.



**Gambar II.1. Jaringan Remote Desktop**

Sumber : (Helmi, 2009:81)

## II.2. *Bluetooth*

*Bluetooth* adalah sebuah teknologi komunikasi *wireless* (tanpa kabel) yang beroperasi dalam pita frekuensi 2,4 GHz *unlicensed ISM (Industrial, Scientific and Medical)* dengan menggunakan sebuah *frequency hopping tranceiver* yang mampu menyediakan layanan komunikasi data dan suara secara *real-time* antara *host-host bluetooth* dengan jarak jangkauan layanan yang terbatas. (Shalahuddin 2008:200)

Setiap kali *bluetooth* dijalankan diperlukan tiga hal agar koneksi dapat terhubung, yaitu:

1. Inisialisasi, untuk menjalankan koneksi *bluetooth* antara *server* dan *client* diperlukan inisialisasi pada *stack bluetooth*.
2. *Server, server* membuat pelayanan dapat diakses oleh *client, server* mendaftarkan *client* pada *Service Discovery Database (SDDB)*, *Server* menunggu koneksi dari *client* dan menerima koneksi itu yang pada akhirnya jika pelayanan dari *server* ke *client* sudah tidak diperlukan lagi maka *server* akan menghapus registrasi *client* di SDDB.
3. *Client, client* menggunakan pelayanan dari *Server* . Pada awalnya *client* akan mencari perangkat yang terdekat, kemudian melakukan pencarian pada perangkat-perangkat tersebut.

## II.3. Jaringan Komputer

Sebuah jaringan komputer paling sedikit terdiri dari dua komputer yang saling terhubung dengan sebuah media sehingga komputer-komputer tersebut

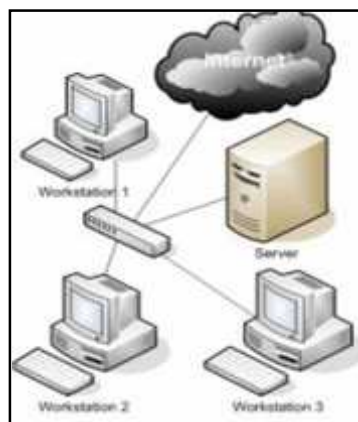
dapat saling berbagi *resource* dan saling berkomunikasi. Kenyataannya sebuah *network* biasanya terdiri dari banyak komputer (lebih dari dua) semua *network* berbasis pada konsep pembagian (*sharing*). (Arifin, 2005 : 1)

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 : 2)

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

#### 1. LAN (*Local Area Network*)

LAN 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). Dapat dilihat pada gambar II.2.



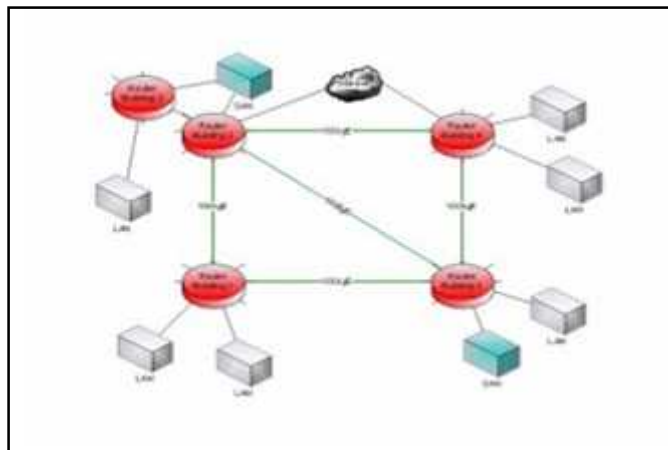
**Gambar II.2. LAN (*Local Area Network*)**

*Sumber* : (Arifin, 2005 : 4)

## 2. MAN (*Metropolitan Area Network*)

MAN 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.

Dapat dilihat pada gambar II.3.

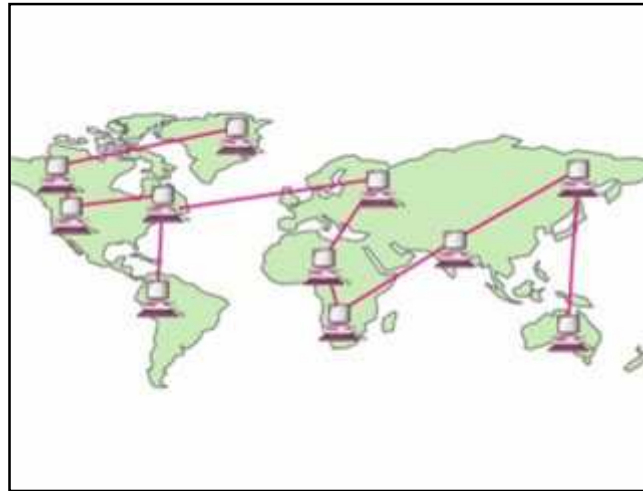


**Gambar II.3. MAN (*Metropolitan Area Network*)**

*Sumber : (Arifin ,2005 : 4)*

## 3. WAN (*World 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. Dapat dilihat pada gambar II.4.

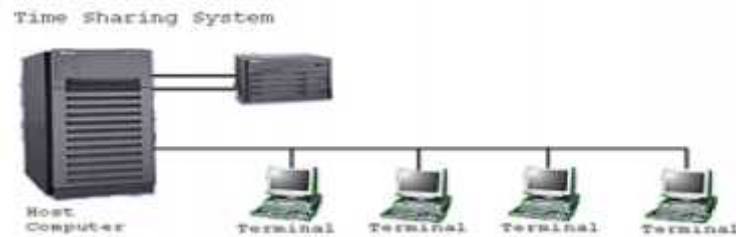


**Gambar II.4. WAN (Word Area Network)**

*Sumber :Arifin, (2005 : 5)*

*Local Area Network* (LAN), merupakan jaringan milik pribadi di dalam sebuah gedung atau kampus yang berukuran sampai beberapa kilometer. LAN seringkali digunakan untuk menghubungkan komputer-komputer pribadi dan *workstation* dalam kantor suatu perusahaan atau pabrik-pabrik untuk memakai bersama sumberdaya (*resource*, misalnya printer) dan saling bertukar informasi.

Pada tahun 1940-an di Amerika ada sebuah penelitian yang ingin memanfaatkan sebuah perangkat komputer secara bersama. Ditahun 1950-an ketika jenis komputer mulai membesar sampai terciptanya super komputer, karena mahalnya harga perangkat komputer maka ada tuntutan sebuah komputer mesti melayani beberapa terminal. Dari sinilah maka muncul konsep distribusi proses berdasarkan waktu yang dikenal dengan nama TSS (*Time Sharing System*), bentuk pertama kali jaringan (*network*) komputer diaplikasikan. Pada sistem TSS beberapa terminal terhubung secara seri ke sebuah *host* komputer yang dapat dilihat pada gambar II.5.



**Gambar II.5. Time Sharing System**

*Sumber : (Wahidin ,2005:15)*

Selanjutnya konsep ini berkembang menjadi proses distribusi (*Distributed Processing*). Dalam proses ini beberapa *host* komputer mengerjakan sebuah pekerjaan besar secara paralel untuk melayani beberapa terminal yang tersambung secara seri disetiap *host* komputer, dapat dilihat pada gambar II.6 .



**Gambar II.6. Distributed Processing**

*Sumber : (Wahidin, 2007:15)*

Selanjutnya ketika harga-harga komputer kecil sudah mulai menurun dan konsep proses distribusi sudah matang, maka penggunaan komputer dan jaringannya sudah mulai beragam dari mulai menangani proses bersama maupun komunikasi antar komputer (*Peer to Peer System*) saja tanpa melalui komputer pusat. Untuk itu mulailah berkembang teknologi jaringan lokal yang dikenal

dengan sebutan LAN (*Local Area Network*). Demikian pula ketika Internet mulai diperkenalkan, maka sebagian besar LAN yang berdiri sendiri mulai berhubungan dan terbentuklah jaringan raksasa ditingkat dunia yang disebut dengan istilah WAN (*World Area Network*).

Pada umumnya semua jaringan memiliki komponen, fungsi dan kemampuan sebagai berikut :

1. *Server*, komputer yang menyediakan *resource* agar dapat diakses oleh semua pengguna jaringan.
2. *Client*, komputer-komputer yang dapat mengakses *server* dan dapat menggunakan *resource* yang tersedia.
3. *Media*, cara komputer-komputer dalam sebuah jaringan terhubung.
4. *Shared data, file* atau *folder* yang disediakan *server* melalui jaringan.
5. *Shared printer* dan *device* lain, *resource* lain yang disediakan oleh *server*.
6. *Resource, file, printer*, atau *item-item* yang digunakan oleh pengguna jaringan.

Berdasarkan penjelasan hal diatas, *network* terbagi ke dalam dua kategori penggolongan yaitu:

1. *Peer to peer (workgroup)*. Dalam jaringan *peer to peer* tidak ada *server* khusus atau hierarki di antara komputer-komputer yang saling terhubung. Semua komputer berstatus sama dan dikenal sebagai *peer*. Komputer-komputer yang terdapat dalam jaringan *peer to peer* dapat bertindak sebagai *server* dan juga bertindak sebagai *client*. Jenis jaringan ini juga tidak memiliki administrator yang bertanggung jawab terhadap seluruh jaringan. Masing-masing *user* bertindak sebagai administrator pada komputer yang

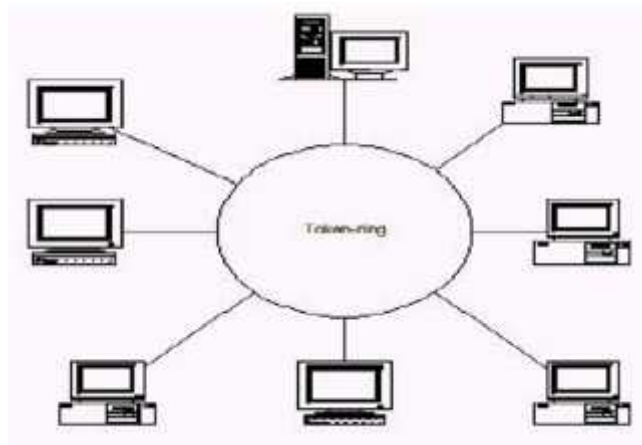
digunakannya. *User* masing-masing komputer dapat menentukan data mana yang diperbolehkan untuk diakses oleh *user* lain. (Arifin ,2005 : 7)

2. *Client-Server (server based)*, Jaringan *client-server* merupakan sebuah sistem jaringan yang terdiri dari dua bagian, yakni :
  - a. *Client*, mesin yang melakukan permintaan *resource*.
  - b. *Dedicated server*, mesin yang menyediakan *resource* dan memproses permintaan dari *client* kemudian mengembalikan hasil proses tersebut kembali ke *client*. Yang melakukan permintaan sebelumnya disebut *dedicated* karena mesin yang bertindak sebagai *server* dioptimalisasi untuk secara cepat melayani permintaan dari *client* jaringan dan menjamin sistem keamanan *file* dan *directory*. (Arifin ,2005 : 9)

#### **II.4. Topologi Jaringan**

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.

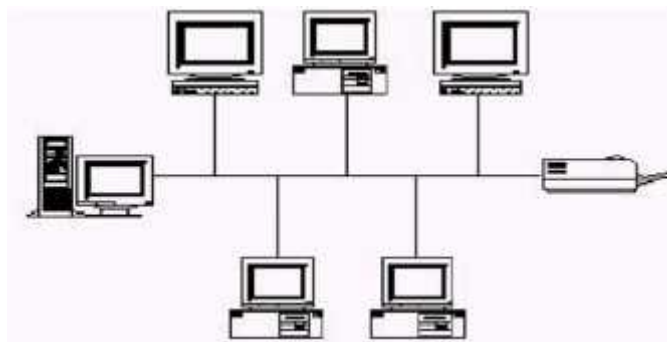
1. Topologi Token Ring, Di dalam topologi Ring semua *workstation* dan server dihubungkan sehingga terbentuk suatu pola lingkaran atau cincin. Tiap *workstation* ataupun *server* akan menerima dan melewatkan informasi dari satu komputer ke komputer lain, bila alamat-alamat yang dimaksud sesuai maka informasi diterima dan bila tidak informasi akan dilewatkan, yang dapat dilihat pada gambar II.7 .



**Gambar II.7. 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.8.

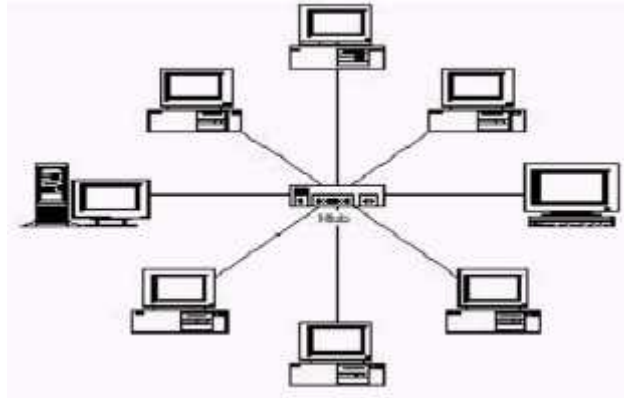


**Gambar II.8. Topologi Jaringan Bus**

*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.9 .



**Gambar II.9. Topologi Jaringan Star**

*Sumber : (Wahidin, 2007:16)*

## II.5. Tinjauan Umum TCP/IP

TCP/IP (singkatan dari *Transmission Control Protocol/Internet Protocol*) adalah standar komunikasi data yang digunakan oleh komunitas internet dalam proses tukar-menukar data dari satu komputer ke komputer lain di dalam jaringan Internet. Protokol ini tidaklah dapat berdiri sendiri, karena memang protokol ini berupa kumpulan protokol (*protocol suite*). Protokol ini juga merupakan protokol yang paling banyak digunakan saat ini. Data tersebut diimplementasikan dalam bentuk perangkat lunak (*software*) di sistem operasi. Istilah yang diberikan kepada perangkat lunak ini adalah TCP/IP stack.

Protokol TCP/IP dikembangkan pada akhir dekade 1970-an hingga awal 1980-an sebagai sebuah protokol standar untuk menghubungkan komputer-komputer dan jaringan untuk membentuk sebuah jaringan yang luas (WAN). TCP/IP merupakan sebuah standar jaringan terbuka yang bersifat independen

terhadap mekanisme transport jaringan fisik yang digunakan, sehingga dapat digunakan di mana saja. Protokol ini menggunakan skema pengalamatan yang sederhana yang disebut sebagai alamat IP (*IP Address*) yang mengizinkan hingga beberapa ratus juta komputer untuk dapat saling berhubungan satu sama lainnya di Internet. Protokol ini juga bersifat *routable* yang berarti protokol ini cocok untuk menghubungkan sistem-sistem berbeda (seperti Microsoft Windows dan keluarga UNIX) untuk membentuk jaringan yang heterogen.

Protokol TCP/IP selalu berevolusi seiring dengan waktu, mengingat semakin banyaknya kebutuhan terhadap jaringan komputer dan Internet. Pengembangan ini dilakukan oleh beberapa badan, seperti halnya *Internet Society* (ISOC), *Internet Architecture Board* (IAB), dan *Internet Engineering Task Force* (IETF). Macam-macam protokol yang berjalan di atas TCP/IP, skema pengalamatan, dan konsep TCP/IP didefinisikan dalam dokumen yang disebut sebagai *Request for Comments* (RFC) yang dikeluarkan oleh IETF. TCP/IP pun mempunyai beberapa layer, layer-layer itu adalah :

1. IP (*internet protocol*), yang berperan dalam penransmisian paket data dari node ke node. IP mendahului setiap paket data berdasarkan 4 byte (untuk versi IPv4) alamat tujuan (nomor IP). Internet authorities menciptakan range angka untuk organisasi yang berbeda. Organisasi menciptakan grup dengan nomornya untuk departemen. IP bekerja pada mesin *gateway* yang memindahkan data dari departemen ke organisasi kemudian ke region dan kemudian ke seluruh dunia.
2. TCP (*transmission transfer protocol*), berperan didalam memperbaiki

pengiriman data yang benar dari suatu *client* ke *server*. Data dapat hilang di tengah-tengah jaringan. TCP dapat mendeteksi *error* atau data yang hilang dan kemudian melakukan transmisi ulang sampai data diterima dengan benar dan lengkap.

3. *Sockets*, yaitu merupakan nama yang diberikan kepada subrutin paket yang menyediakan akses ke TCP/IP pada kebanyakan sistem.

## II.6. Protokol Komputer

Protokol Komputer adalah sebuah aturan atau standar yang mengatur atau mengijinkan 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 macam 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. Mendeteksi *rugi-rugi* pada hubungan jaringan dan langkah-langkah yang dilakukan selanjutnya
8. Mengakhiri 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.

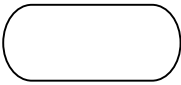

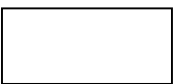
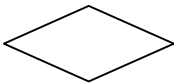
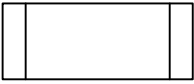


## **II.8. Flowchart**

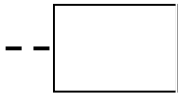
Bagan alir program (*program flowchart*) adalah suatu bagan yang menggambarkan arus logika dari data yang akan diproses dalam suatu program

dari awal sampai akhir. Bagan alir program merupakan alat yang berguna bagi *programmer* untuk mempersiapkan program yang rumit. Bagan alir terdiri dari simbol-simbol yang mewakili fungsi-fungsi langkah program dan garis alir (*flowlines*) menunjukkan urutan dari simbol-simbol yang akan dikerjakan. (Munawar, 2005)

Berikut adalah simbol-simbol program *flowchart* menurut ANSI (*American National Standard Institute*), yang dapat dilihat pada tabel II.1.

**Tabel II.1. Simbol *Flowchart***

<b>Simbol</b>	<b>Arti</b>	<b>Keterangan</b>
	<b>Terminal</b>	Untuk menunjukkan awal dan akhir program.
	<b>Persiapan (<i>preparation</i>)</b>	Untuk memberikan nilai awal pada suatu variabel atau <i>counter</i> .
	<b>Pengolahan (<i>processing</i>)</b>	Untuk pengolahan aritmatika dan pemindahan data.
	<b>Keputusan (<i>decision</i>)</b>	Untuk mewakili operasi perbandingan logika.
	<b>Proses terdefinisi (<i>predefined process</i>)</b>	Untuk proses yang detailnya dijelaskan terpisah, misalnya dalam bentuk subroutine.
	<b>Penghubung (<i>connector</i>)</b>	Untuk menunjukkan hubungan arus proses yang terputus masih dalam halaman yang sama.
	<b>Penghubung halaman lain (<i>off page connector</i>)</b>	Untuk menunjukkan hubungan arus proses yang terputus masih dalam halaman yang sama.

	<p align="center"><b>Penjelasan (<i>annotation flag</i>)</b></p>	<p>Untuk memberikan keterangan-keterangan guna memperjelas simbol-simbol yang lain.</p>
---	--	---

Sumber : (Munawar, 2005)

## II.9. 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 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 metodolgi 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*.

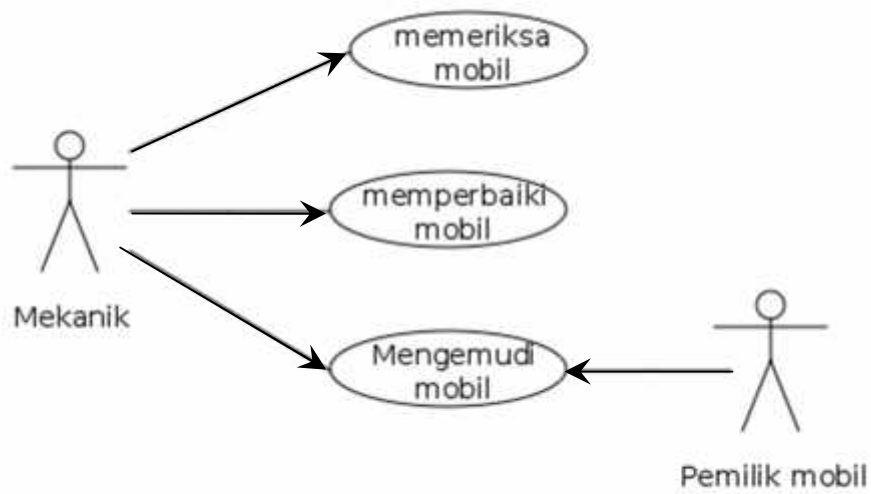
### **II.9.1. Use Case Diagram**

*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.

Dapat dilihat pada gambar II.10.



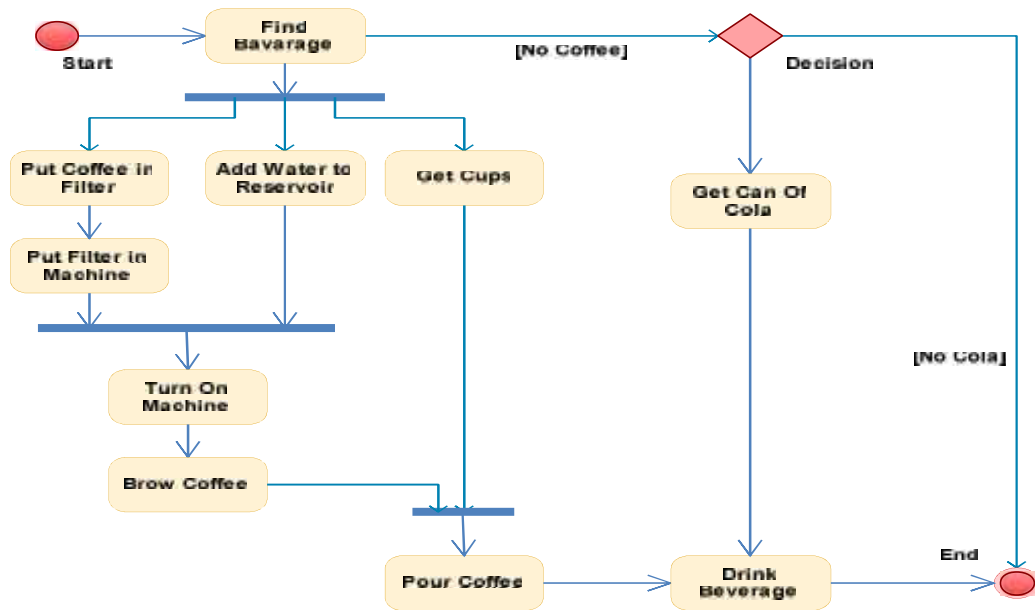
**Gambar II. 10 Contoh Use Case Diagram**  
 Sumber : (Munawar, 2005)

## II.9.2. 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.

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. Dapat dilihat pada gambar II.11.

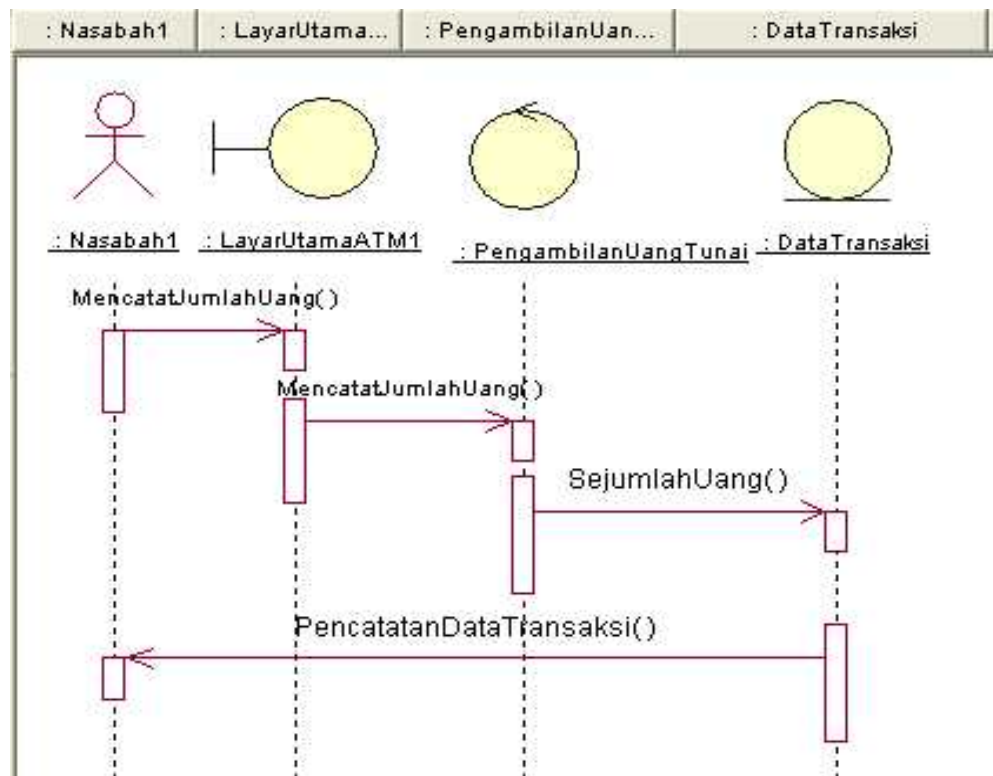


**Gambar II.11. Contoh Activity Diagram**

Sumber : (Munawar, 2005)

### II.9. 3. Sequence Diagram

*Sequence* 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. Dapat dilihat pada gambar II.12.



**Gambar II.12. Contoh Sequence Diagram**

Sumber : (Munawar, 2005)

## II.10. 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.10.1. Java2 Standard Edition (J2SE)**

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

### **II.10.2. 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.10.3. 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)

#### **II.10.2.1. Connected Limited Device Configuration (CLDC)**

*Connected Limited Device Configuration* (CLDC) menyediakan sebuah mesin maya dan pustaka inti yang digunakan oleh sebuah industri untuk mendefinisikan *Profile*. CLDC dirancang dan disusun oleh *Java Community Process* yang telah memenuhi standarisasi *Sun Microsystems* tentang portabilitas dan minimal terpeenuhinya *footprint* membangun blok aplikasi *Java* untuk perangkat yang mempunyai sumber daya terbatas. (Anonim, 2012:02).

#### **II.10.2.2. Connected Device Configuration (CDC)**

CDC atau *Connected Device Configuration* adalah spesifikasi dari konfigurasi J2ME. CDC merupakan komunitas proses pada *Java* yang memiliki standarisasi. CDC terdiri dari *Virtual Machine* dan kumpulan *library* dasar untuk dipergunakan pada *profile* industri. Implementasi CDC pada J2ME adalah *source code* yang menyediakan sambungan dengan macam-macam *platform*.

(Shalahuddin, 2008 : 06)

### II.10.2.3. Mobile Information Device Profile (MIDP)

MIDP adalah spesifikasi untuk sebuah profil J2ME. MIDP memiliki lapisan diatas CLDC, API tambahan untuk daur hidup aplikasi, antarmuka, jaringan, dan penyimpanan persistem. Pada saat ini terdapat MIDP 1.0 dan MIDP 2.0. Fitur tambahan MIDP 2.0 dibandingkan MIDP 1.0 adalah API untuk *multimedia*. Pada MIDP 2.0 terdapat pendukung memainkan *tone*, *tone squence*, dan *file wav* walaupun tanpa adanya *mobile media API (MMAPI)*. (Shalahuddin, 2008 : 07)

### II.10.4. 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*. (Anonim, 2012 : 13)

#### II.10.4.1 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*, *Navigator*, *Palette*.

## 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.

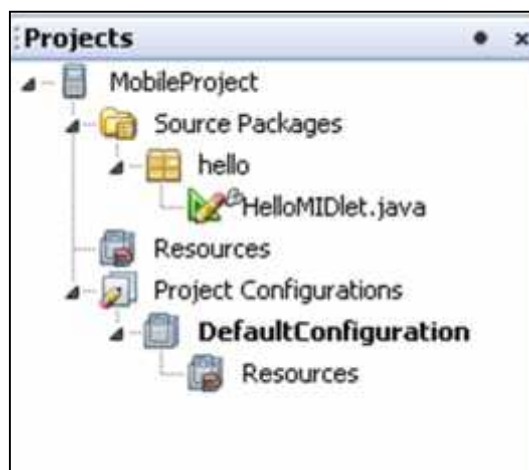
### b. *Resources*

*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*.

dapat dilihat pada gambar II.13.

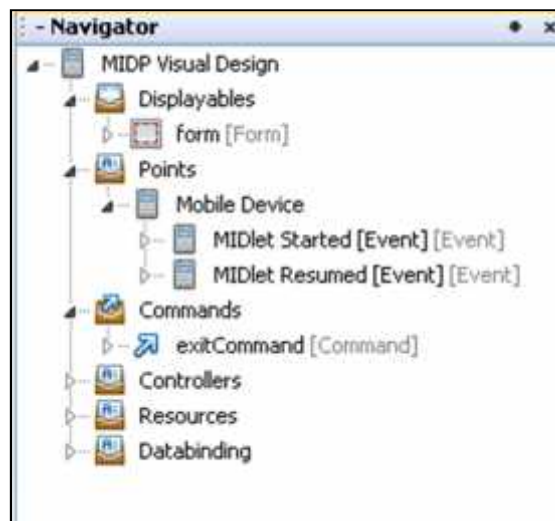


**Gambar II.13. Panel Project**

(Anonim : 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. Dapat dilihat pada gambar II.14



**Gambar II.14. Panel Navigator**  
(Anonim : 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*.