

BAB II

TINJAUAN PUSTAKA

II.1. Proyek Terapan (Implementasi)

Proyek terapan/implementasi (*applied project*) sering sekali berhubungan dengan melakukan kegiatan eksperimen dan mengumpulkan bukti-bukti dari suatu kaidah yang diterapkan dan mengambil pengalaman dari kegiatan tersebut.

Dalam kegiatan ini bisa saja terdapat tujuan-tujuan seperti (1) menyiapkan simulator, (2) menerapkan sebuah algoritma, (3) melakukan percobaan dan menganalisa hasil yang didapat, dan (4) menyarankan perbaikan terhadap suatu algoritma. ^[3]

II.2. Teori Dasar Jaringan Komputer

II.2.1. Jaringan Komputer

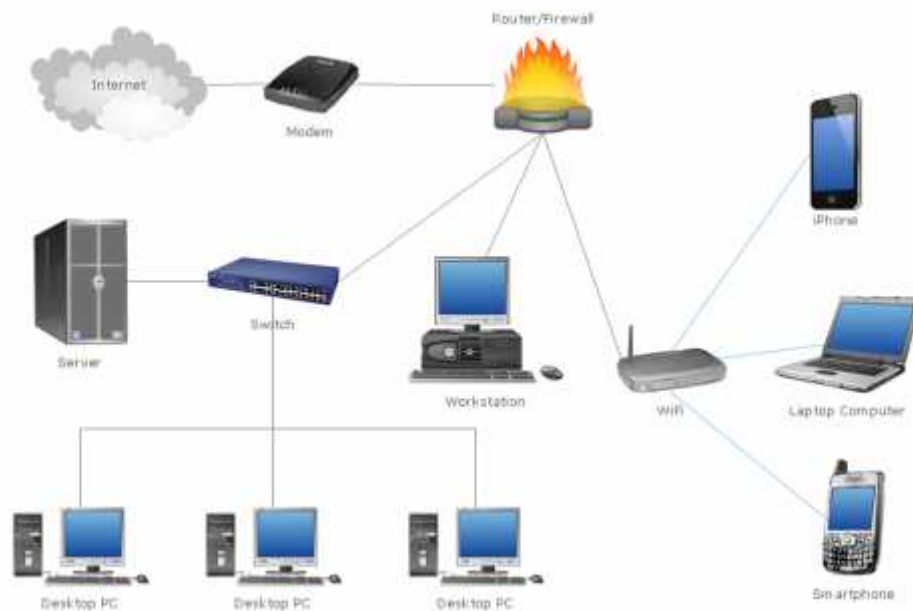
Jaringan komputer merupakan gabungan antara teknologi komputer dan teknologi telekomunikasi. Gabungan teknologi ini melahirkan pengolahan data yang dapat didistribusikan, mencakup pemakaian *database*, *software* aplikasi dan peralatan *hardware* secara bersamaan, sehingga penggunaan komputer yang sebelumnya hanya berdiri sendiri, kini telah diganti dengan sekumpulan komputer yang terpisah-pisah akan tetapi saling berhubungan dalam melaksanakan tugasnya, sistem seperti inilah yang disebut jaringan komputer (*computer network*). ^[17]

II.2.2. Jenis-Jenis Jaringan Komputer

Dilihat dari skop dan luas jaringan, jaringan komputer secara geografis dapat dibedakan menjadi tiga kelompok :

1. *Local Area Network (LAN)*

Local Area Network (LAN) merupakan jaringan yang bersifat internal dan biasanya milik pribadi di dalam sebuah perusahaan kecil atau menengah dan biasanya berukuran sampai beberapa kilometer. *LAN* seringkali digunakan untuk menghubungkan komputer pribadi dan *workstation* dalam kantor suatu perusahaan atau pabrik-pabrik untuk pemakaian sumber daya bersama (*resource*, baik *hardware* maupun *software*) serta sarana untuk bertukar informasi. ^[17]

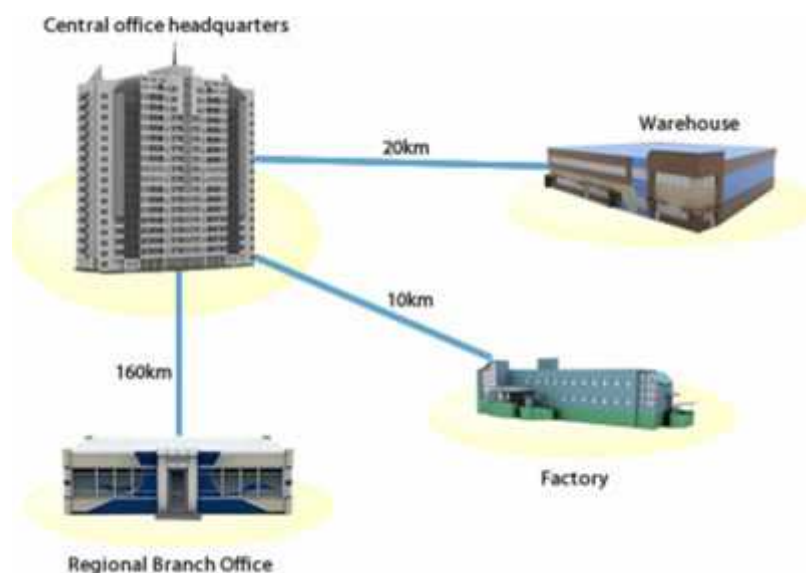


Gambar II.1. *Local Area Network (LAN)*

Sumber : <http://www.conceptdraw.com>

2. *Metropolitan Area Network (MAN)*

Metropolitan Area Network (MAN) adalah sebuah jaringan yang menggunakan teknologi yang sama dengan *LAN*, hanya ukurannya biasanya lebih luas daripada *LAN*. *MAN* dapat mencakup kantor-kantor perusahaan yang letaknya berdekatan atau antar sebuah kota dan dapat dimanfaatkan untuk keperluan pribadi (swasta) atau umum. ^[17]

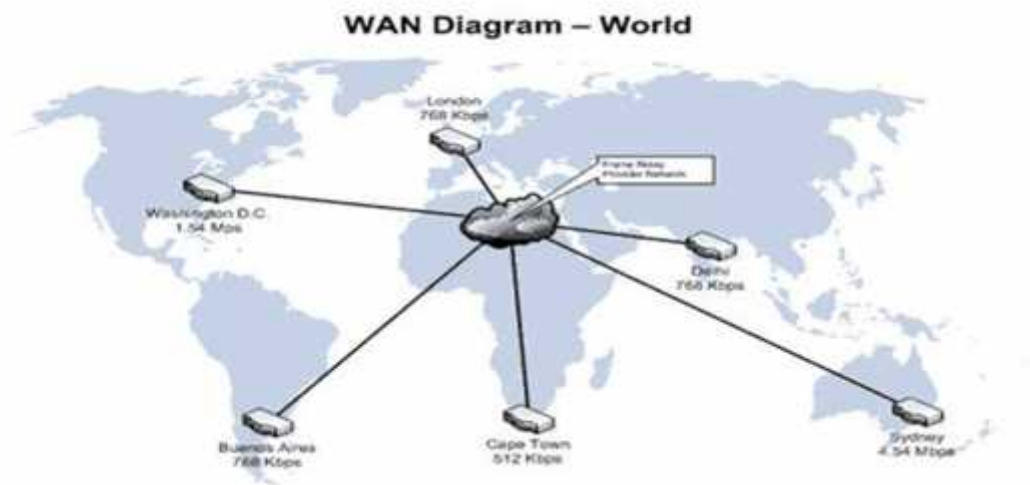


Gambar II.2. *Metropolitan Area Network (MAN)*

Sumber : <https://www.perle.com>

3. *Wide Area Network (WAN)*

Wide Area Network (WAN) jangkauannya mencakup daerah geografis yang lebih luas, seringkali mencakup sebuah negara bahkan benua. *WAN* terdiri dari kumpulan *LAN*, *MAN*, dan mesin-mesin yang bertujuan untuk menjalankan program aplikasi pemakai. ^[17]



Gambar II.3. Wide Area Network (WAN)
 Sumber : <http://www.sunrisepcsupport.com/>

II.3. Perangkat-Perangkat Jaringan

II.3.1. Server

Secara umum *server* adalah sebuah komputer yang berisi program baik sistem operasi maupun program aplikasi yang menyediakan pelayanan kepada komputer atau program lain yang sama ataupun berbeda. Komputer *server* adalah komputer yang biasanya dikhususkan untuk penyimpanan data yang akan digunakan bersama, atau sebagai basis data. Selain itu, jika menggunakan sistem operasi berbasis *network* (*Network Operating System*) maka komputer *server* berisi informasi daftar *user* yang diperbolehkan masuk ke *server* tersebut, berikut otoritasnya yang dapat di-*manage* oleh *supervisor* atau *administrator*. Dalam model program *client/server*, *server* adalah program yang menunggu dan memenuhi permintaan dari *client* program yang sama atau berbeda.^[17]

II.3.2. Network Interface Card (LAN Card)

LAN Card adalah sebuah perangkat jaringan yang dipasang pada *slot* ekspansi pada sebuah *motherboard* komputer sehingga komputer dapat dihubungkan ke dalam sistem jaringan. Dilihat dari jenis *slot*-nya umumnya terbagi dua yaitu *PCI (Peripheral Component Interconnect)* dan *ISA (Industry Standard Architecture)*.^[17]



Gambar II.4. LAN Card
Sumber : www.digitus.info

II.3.3. Hub

Hub adalah istilah yang umum yang digunakan untuk menerangkan sebuah *central connection point* untuk komputer pada jaringan. *Hub* tidak memiliki fasilitas *routing*, sehingga semua informasi yang datang akan dikirimkan ke semua komputer (*broadcast*). Fungsi dasar yang dilakukan oleh *hub* adalah menerima sinyal dari satu komputer dan mentransmisikannya ke komputer yang lain. Sebuah *Hub* dapat bersifat *active* atau *passive*. *Active hub* bertindak sebagai penguat sinyal (*repeater*). *Passive hub* hanya bertindak sebagai kotak sambungan yang membagi sinyal yang masuk untuk ditransmisikan ke seluruh jaringan.^[17]



Gambar II.5. Hub

Sumber : <http://www.flytec.com.py>

II.3.4. Switch

Switch adalah *hub* pintar yang juga berfungsi sebagai penguat sinyal (*repeater*) dan mempunyai kemampuan untuk menentukan tujuan *MAC Address* (*Media Access Control Address*; atau *physical address*) dari sebuah paket data. *Switch* mampu bekerja secara *full-duplex* yang artinya mampu menerima dan mengirimkan data pada saat yang bersamaan. ^[17]



Gambar II.6. Switch

Sumber : www.tjskl.org.cn

II.3.5. Kabel Jaringan

Kabel merupakan salah satu bagian yang terpenting dalam media koneksi antara komputer dengan komputer lainnya, berikut ini penjelasan ringkas mengenai tipe dan jenis kabel yang biasa digunakan dalam sebuah jaringan :

1. Kabel *Unshielded Twisted Pair* (UTP)

Merupakan sepasang kabel yang dibelit satu sama lain dengan tujuan untuk mengurangi interferensi listrik. Kabel ini terdiri dari delapan pasang kabel yang terdiri dari warna (orange-putih, orange), (hijau-putih, hijau), (coklat-putih, coklat) dan (putih-biru, biru). Pada umumnya kabel ini menggunakan konektor RJ-45. Pada saat ini penggunaan kabel UTP merupakan pilihan yang paling efisien dan banyak dipakai dalam pengembangan jaringan komputer berkecepatan tinggi.^[17]



Gambar II.7. Kabel UTP
Sumber : <http://www.sagi.no>

2. Kabel *Shielded Twisted Pair* (STP)

Secara fisik kabel *shielded* sama dengan kabel *unshielded* tetapi perbedaannya sangat besar di mulai dari konstruksi. Kabel *shielded* mempunyai selubung tembaga atau aluminium foil yang khusus dirancang untuk mengurangi gangguan elektromagnetik. Kekurangan STP adalah tidak

samanya standar antar perusahaan yang memproduksinya dan lebih mahal serta lebih tebal sehingga susah dalam penanganan fisiknya. ^[17]

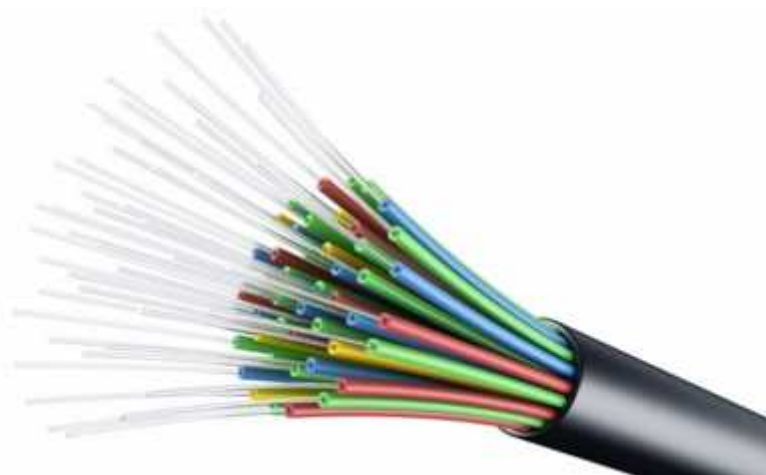


Gambar II.8. Kabel STP

Sumber : www.directcablesystems.com

3. Kabel Serat Optik (*Fiber Optic*)

Teknologi *fiber optic* atau serat cahaya memungkinkan menjangkau jarak yang besar dan menyediakan perlindungan total terhadap gangguan elektrik. Kecepatan transfer data dapat mencapai 1000 Mbps serta jarak dalam satu segmen dapat lebih dari 3.5 Km. Kabel serat optik tidak terganggu oleh lingkungan atau cuaca. Keuntungan lain dengan menggunakan kabel serat optik adalah tidak mudah disadap sehingga lebih aman. ^[17]



Gambar II.9. Kabel *Fiber Optic*

Sumber : www.androidauthority.com

II.3.5. Router

Router adalah inti dari jaringan. *Router* adalah sebuah perangkat jaringan yang bertanggung jawab untuk meneruskan paket dari satu jaringan ke jaringan yang lain, dari sumber ke tujuan. *Router* menghubungkan beberapa jaringan. *Router* memiliki beberapa *interface* jaringan dimana setiap *interface* tersebut terhubung ke jaringan yang berbeda. Layaknya sebuah komputer personal, *router* juga memiliki *Central Processing Unit (CPU)*, *Random-Access Memory (RAM)* dan *Read-Only Memory (ROM)*.^[6]

Fungsi utama dari *router* adalah menghubungkan beberapa jaringan dan meneruskan paket yang ditujukan kepada jaringan yang terhubung langsung dengannya ataupun jaringan luar. Saat sebuah *router* menerima sebuah paket, ia memeriksa *IP Address* tujuannya. Jika *IP Address* tujuan dari paket tersebut tidak ada pada jaringan yang terhubung langsung dengan *router*, maka *router* harus meneruskan paket tersebut kepada *router* yang lain.^[6]

II.4. High-Availability (Ketersediaan Yang Terjamin)

Kini, ketersediaan jaringan yang terjamin sangat dibutuhkan dalam jaringan telekomunikasi. Tidak hanya pada *level* perangkat, namun juga pada jaringan telekomunikasi secara keseluruhan. Ketersediaan adalah waktu rata-rata operasi yang diharapkan dari sebuah perangkat. *Provider* jaringan modern, operator jaringan dan produsen perangkat jaringan menargetkan ketersediaan jaringan mereka hingga 99,999% (“5 nine” *availability*), ini berarti sebuah jaringan hanya diperkenankan mengalami gangguan selama 5 menit dalam kurun waktu satu

tahun. Mereka bertanya-tanya bagaimana cara mendesain jaringan (atau perangkat jaringan) agar mampu mencapai ketersediaan yang tinggi dengan investasi sekecil mungkin. *Provider* tahu bahwa dimasa depan, *client* mereka tidak akan mentolerir gangguan layanan dalam bentuk apapun, maka dari itu mereka ingin jaringan mereka tetap dapat berfungsi terus-menerus. ^[4]

High-availability cluster atau dikenal juga sebagai *failover cluster* adalah sekumpulan komputer yang mendukung aplikasi *server* yang handal dengan waktu *down-time* yang rendah. Kumpulan komputer tersebut beroperasi dengan memanfaatkan *redundancy* yang ada pada *cluster* tersebut yang tetap menyediakan layanan saat komponen sistem lainnya mengalami gangguan. Tanpa *clustering*, jika sebuah *server* yang menjalankan suatu aplikasi tertentu mengalami gangguan maka aplikasi tersebut tidak akan berfungsi hingga *server* diperbaiki. *High-availability clustering* membantu situasi seperti ini dengan mendeteksi kegagalan *software/hardware* dan melakukan *restart* ke sistem lainnya tanpa membutuhkan tindakan administratif apapun, proses ini disebut dengan *failover*. ^[12]

II.5. Redundancy

Salah satu aspek terpenting dalam menjamin ketersediaan yang tinggi adalah *redundancy*. Pendekatan yang biasanya dilakukan untuk mendapatkan ketersediaan yang tinggi adalah membangun sistem *backup* yang tetap akan berfungsi bila terjadi kegagalan pada sistem utama. ^[9]

Redundant link adalah *link* cadangan yang membuat sebuah jaringan memiliki fitur *failover*. *Failover* itu sendiri adalah teknik yang memungkinkan sebuah *link* untuk menggantikan tugas *link* yang lain jika terjadi kegagalan. ^[18]

Pada *high-availability cluster*, terdapat beberapa *server* yang digunakan untuk memberikan layanan secara kontinyu pada *client*. Bila terdapat aktifitas internal atau eksternal yang mengganggu kinerja *server*, maka semua *client* akan dialihkan ke *server* cadangan. ^[14]

Simpul [perangkat] yang mengalami gangguan pada sebuah *high-availability cluster* tidak dapat terlihat oleh *client* yang berada diluar *cluster*. ^[13]

II.6. *Failover*

Pada dunia komputasi, *failover* adalah proses pengalihan kepada komputer *server*, sistem, *hardware* atau jaringan yang disebabkan oleh terjadinya gangguan atau ketidak-normalan aplikasi, *server*, sistem, *hardware* atau jaringan. *System designer* biasanya memberikan fitur *failover* pada *server*, sistem atau jaringan yang membutuhkan ketersediaan yang kontinyu dan handal. ^[12]

Didalam sebuah *failover cluster*, terdapat dua buah komputer (atau dalam keadaan tertentu bisa juga terdapat banyak komputer). Yang pertama (*primary*) memberikan layanan saat keadaan normal. Komputer yang kedua (*failover*) akan menyediakan layanan saat sistem utama mengalami gangguan. ^[14]

II.7. UML (*Unified Modelling Language*)

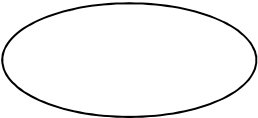
Penjadwalan telah lama diteliti, contohnya dalam penghasilan tenaga oleh Windu Gata dan Grace Gata (2013), *UML (Unified Modelling Language)* adalah bahasa spesifikasi standar yang dipergunakan untuk mendokumentasikan, menspesifikasikan dan membangun perangkat lunak. *UML (Unified Modelling Language)* merupakan metodologi dalam mengembangkan sistem berorientasi objek dan juga merupakan alat untuk mendukung pengembangan sistem. ^[8]

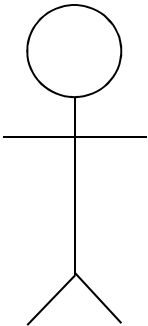
II.7.1 *UseCase Diagram*

UseCase Diagram merupakan pemodelan untuk kelakuan (*behavior*) sistem informasi yang akan dibuat. Mendeskripsikan sebuah interaksi antara satu atau lebih aktor dengan sistem informasi yang akan dibuat. Dapat dikatakan *usecase* digunakan untuk mengetahui fungsi apa saja yang ada di dalam sistem informasi dan siapa saja yang berhak menggunakan fungsi-fungsi tersebut. ^[8]

Simbol-simbol yang digunakan dalam *UseCase Diagram*, yaitu:

Tabel II.1. *UseCase Diagram*

Gambar	Keterangan
	<i>UseCase</i> menggambarkan fungsionalitas yang disediakan sistem sebagai unit-unit yang bertukar pesan antar unit dengan aktor, biasanya dinyatakan dengan menggunakan kata kerja diawal nama <i>UseCase</i>.

	<i>Actor</i> atau Aktor adalah <i>abstraction</i> dari orang atau sistem yang lain yang mengaktifkan fungsi dari target sistem. Untuk mengidentifikasi aktor, harus ditentukan pembagian tenaga kerja dan tugas-tugas yang berkaitan dengan peran pada konteks target sistem. Orang atau sistem bisa muncul dalam beberapa peran. Perlu dicatat bahwa aktor berinteraksi dengan <i>usecase</i>, tetapi tidak memiliki kontrol terhadap <i>usecase</i>.
	Asosiasi antara aktor dan <i>usecase</i>, digambarkan dengan garis tanpa panah yang mengindikasikan siapa atau apa yang meminta interaksi secara langsung dan bukannya mengindikasikan aliran data.
	Asosiasi antara aktor dan <i>usecase</i> yang menggunakan panah terbuka untuk mengindikasikan bila aktor berinteraksi secara pasif dengan sistem.

(Sumber : Windu Gata, 2013)

II.7.2 Diagram Kelas (*Class Diagram*)

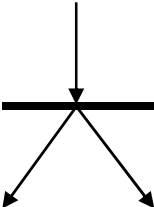
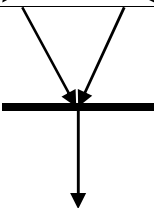
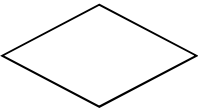
Class Diagram merupakan hubungan antar kelas dan penjelasan detail tiap-tiap kelas di dalam model desain dari suatu sistem, juga memperlihatkan aturan-aturan dan tanggung jawab entitas yang menentukan perilaku sistem. *Class Diagram* juga menunjukkan atribut-atribut dan operasi-operasi dari sebuah kelas dan *constraint* yang berhubungan dengan objek yang dikoneksikan. ^[8]

II.7.3 Diagram Aktivitas (*Activity Diagram*)

Activity Diagram menggambarkan *workflow* (aliran kerja) atau aktivitas dari sebuah sistem atau proses bisnis. ^[8]

Simbol-simbol yang digunakan dalam *activity diagram*, yaitu:

Tabel II.2. Activity Diagram

Gambar	Keterangan
	<i>Start Point</i> diletakkan pada pojok kiri atas dan merupakan awal aktifitas.
	<i>End Point</i> , akhir aktifitas.
	<i>Activities</i> , menggambarkan suatu proses/kegiatan bisnis.
	<i>Fork</i> (percabangan), digunakan untuk menunjukkan kegiatan yang dilakukan secara paralel atau untuk menggabungkan dua kegiatan paralel menjadi satu.
	<i>Join</i> (penggabungan) atau <i>Rake</i> , digunakan untuk menunjukkan adanya dekomposisi.
	<i>Decision Points</i> , menggambarkan pilihan untuk pengambilan keputusan, <i>true</i> atau <i>false</i> .

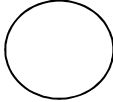
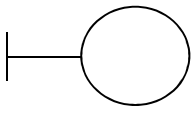
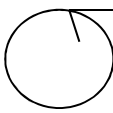
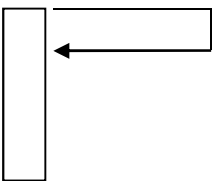
(Sumber : Windu Gata, 2013)

II.7.4 Diagram Urutan (*Sequence Diagram*)

Sequence Diagram menggambarkan kelakuan objek pada *usecase* dengan mendeskripsikan waktu hidup objek dan pesan yang dikirimkan dan diterima antar objek. ^[8]

Simbol-simbol yang digunakan dalam *sequence diagram*, yaitu:

Tabel II.3. *Sequence Diagram*

Gambar	Keterangan
	<i>Entity Class</i> , merupakan bagian dari 25ymbol yang berisi kumpulan kelas berupa entitas-entitas yang membentuk gambaran awal 25ymbol dan menjadi landasan untuk menyusun basis data.
	<i>Boundary Class</i> , berisi kumpulan kelas yang menjadi <i>interface</i> atau interaksi antara satu atau lebih 25ymbo dengan 25ymbol, seperti tampilan <i>formentry</i> dan <i>form</i> cetak.
	<i>Control Class</i> , suatu objek yang berisi logika aplikasi yang tidak memiliki tanggung jawab kepada entitas, contohnya adalah kalkulasi dan aturan bisnis yang melibatkan berbagai objek. <i>Control object</i> mengkoordinir pesan antara <i>boundary</i> dengan entitas.
	<i>Message</i> , 25ymbol mengirim pesan antar <i>class</i> .
	<i>Recursive</i> , menggambarkan pengiriman pesan yang dikirim untuk dirinya sendiri.

	<i>Activation</i> , mewakili sebuah eksekusi operasi dari objek, panjang kotak ini berbanding lurus dengan durasi aktivitas sebuah operasi.
	<i>Lifeline</i> , garis titik-titik yang terhubung dengan objek, sepanjang <i>lifeline</i> terdapat <i>activation</i> .

(Sumber : Windu Gata, 2013)

II.8. MikroTik RouterOS

MikroTik RouterOS adalah sistem operasi dari perangkat MikroTik RouterBOARD. MikroTik RouterOS juga dapat diinstalasi pada *personal computer* (PC) dan akan mengubahnya menjadi *router* dengan semua fitur yang diperlukan - *routing*, *firewall*, manajemen *bandwidth*, *wireless access point*, *backhaul link*, *hotspot gateway*, *VPN server* dan lainnya. RouterOS adalah sistem operasi *stand-alone* yang dibuat berdasarkan *Linux kernel* v2.6. ^[11]

RouterOS mendukung komputer *multi-CPU* dan *multi-core* serta dapat berjalan pada *motherboard* Intel terbaru dan terbesar yang menggunakan *CPU multicore* terbaru. RouterOS mendukung instalasi pada media penyimpanan *Integrated Drive Electronics (IDE)*, *Serial ATA (SATA)* dan *Universal Serial Bus (USB)*, seperti *Hard Disk Drive (HDD)*, *Compact Flash (CF)* dan *Secure Digital Card (SD Card)*, *Solid-State Drive (SSD)* dan banyak lagi. Dibutuhkan setidaknya 64MB ruang penyimpanan yang akan digunakan untuk menginstal RouterOS.

Tentu saja *RouterOS* mendukung banyak antarmuka jaringan, termasuk 10 *Gigabit ethernet card*, 802.11a/b/g/n *wireless card* dan modem 3G. ^[11]

II.9. *Microsoft Visual Studio .Net*

Microsoft Visual Studio .Net merupakan salah satu *software* buatan *Microsoft* yang didesain khusus dalam pembuatan program-program profesional berbasis *windows platform*. *Microsoft Visual Studio .Net* merupakan perangkat lunak yang terintegrasi, di dalamnya terdapat beberapa paket *software* yang dapat digunakan oleh programmer dalam membangun sebuah program profesional, diantaranya adalah *Visual Basic*, *Visual J#*, *Visual C#* dan *Visual C++* yang sama-sama berada dalam naungan *Microsoft .NET Framework platform*. Bagian - bagian dari *software* ini diantaranya *toolbox*, jendela *properties*, *server explorer* dan *solution explorer*. ^[15]

II.10. *Policy-Based Routing (PBR)*

Policy-Based Routing (PBR) adalah metode *routing* yang dilakukan berdasarkan kebijakan (*policy rules*) yang ditetapkan oleh *network administrator*. Dengan *PBR* data dapat ditransmisikan secara terpisah melalui jalur yang berbeda, sebelum ditransmisikan ke tujuan data harus dicocokkan terlebih dahulu dengan *policy rules* yang telah ditetapkan. Hal ini memungkinkan pengecekan data apakah data tersebut diizinkan lewat atau tidak. ^[1]

Policy rules dapat ditetapkan berdasarkan *IP Address*, nomor *port* atau jenis *protocol*. ^[5]