**BAB II**

**TINJAUAN PUSTAKA DAN LANDASAN TEORI**

1. **Tinjauan Pustaka**

Sebagai penguat hasil penelitian dan untuk lebih memahami permasalahan yang penulis bahas dalam penelitian ini, 2 (dua) kajian penelitian berikut penulis jadikan sebagai tinjauan kepustakaan.

1. Junaidi (2015) yakni Rancang Bangun Sistem Ujian Semester *Online* (*e*-*learning*) pada SMK Negeri 1 Selong Lombok Timur, menyimpulkan bahwa dengan dibangunnya sebuah Sistem Ujian *Online* yaitu untuk membantu sekolah dan memfasilitasi kegiatan belajar mengajar kepada para siswa agar dapat meningkatkan mutu pendidikan bagi siswa di SMK Negeri 1 Selong dan dengan adanya *system* Ujian *Online* ini, maka Pengajar SMK Negeri 1 Selong bisa lebih mudah mengawasi siswa ketika melaksanakan ujian dan dengan target waktu yang ditentukan serta mudah memberikan materi pelajaran kepada siswa.
2. Sezi Zalina dan Azis Ahmadi (2015) Perancangan Aplikasi *Test* Masuk Pada Sekolah Menengah Kejuruan (SMK) Negeri 3 Pacitan, menyimpulkan dengan adanya sistem ini dapat mempermudah proses penerimaan siswa baru pada SMK Negeri 3 Pacitan. (<http://eprints.upnjatim.ac.id/3687/1/file1.pdf1307-1381-2-PB>)
3. **Landasan Teori**

Untuk memahami beberapa istilah penting, berikut penulis uraikan kajian teoritis dari berbagai sumber yang bersesuain dengan tema penelitian ini.

1. Sistem

Ada beberapa pendapat tentang pengertian sistem diantaranya menurut Webster’s Dictionary, sistem adalah suatu rangkaian atau tatanan hal-hal yang saling berhubungan untuk membentuk suatu kesatuan atau keseluruhan organik, serangkaian kenyataan, prinsip, aturan dan lain-lain, yang diklasifikasikan dan diatur di dalam bentuk teratur dengan maksud memperlihatkan suatu rencana logis yang menghubungkan bagian-bagian yang berbeda, sebuah metode atau rencana klasifikasikan atau penataan, cara mengerjakan sesuatu, metode, prosedur.

Berdasarkan definisi tersebut di atas maka dapat kita definiskan bahwa sistem berbasis komputer adalah serangkaian atau tatanan elemen-elemen yang diatur untuk mencapai tujuan yang ditentukan sebelumnya melalui pemrosesan informasi.

Tujuannya adalah untuk mendukung berbagai fungsi bisnis atau untuk mengembangkan suatu produk yang dapat dijual untuk menghasilkan keuntungan bisnis. Untuk mencapai tujuan tersebut, sistem berbasis komputer menggunakan berbagai elemen sistem. Adapun elemen-elemen sistem tersebut adalah sebagai berikut :

* 1. Perangkat Lunak

Terdiri dari program komputer, struktur data, dan dokumen yang berhubungan yang berfungsi untuk mempengaruhi metode logis, prosedur, dan kontrol yang dibutuhkan.

* 1. Perangkat Keras

Terdiri dari perangkat elektronik yang memberikan kemampuan penghitungan, dan perangkat elektromekanik (misalnya sensor, rotor, pompa) yang memberikan fungsi dunia eksternal.

* 1. Manusia

Yaitu pemakai dan operator perangkat keras dan perangkat lunak.

* 1. *Databases*

Adalah kumpulan informasi yang besar dan terorganisasi yang diakses melalui perangkat lunak.

* 1. Dokumentasi

Bersifat manual, formulir, dan informasi deskriptif lainnya yang menggambarkan penggunaan atau pengoperasian sistem.

* 1. Prosedur

Adalah langkah-langkah yang menentukan penggunaan khusus dari masing-masing elemen sistem atau konteks prosedural dimana sistem berada.

1. 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 *Open-Office*, 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 setiap aplikasi, aplikasi ini 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. (<http://elib.unikom.ac.id/files/disk1/690/jbptunikompp-gdl-beatrixlia-34471-9-unikom_b-2.pdf>)

1. Konsep Dasar *Multiuser*

Sistem *Multiuser*  atau banyak pengguna atau pengguna ganda adalah istilah dalam sistemoperasi atau perangkat lunak aplikasi yang memperbolehkan akses oleh beberapa pengguna dalam waktu bersamaan ke sistem operasi atau aplikasi tersebut. Istilah lawannya yaitu *Single user* mengacu kepada suatu sistem operasi yang hanya bisa digunakan oleh satu pengguna setiap saat.

Sistem operasi *Multiuser*  adalah sistem operasi yang memungkinkan pengguna lebih dari satu dalam mengakses sistem komputer dalam waktu yang bersamaan. Dimana untuk mengakses komputer biasanya dilakukan melalui jaringan, sehingga para *user* dapat mengakses komputer dari jarak jauh menggunakan terminal atau komputer lain.

1. Fungsi Random

Abdul Kadir (2008) dalam bukunya yang berjudul *Dasar Pemprograman WEB Dinamis Menggunakan PHP*, fungsi – fungsi yang terkait dengan pembangkitan bilangan acak ada tiga.

* 1. Fungsi *getranmax*

Kegunaan fungsi ini adalah untuk memperoleh bilangan acak yang terbesar yang mungkin dihasilkan ( oleh fungsi *rand* ( ) ). Bentuk pemanggilan:

getrandmax( )

Nilai balik yang diberikan bertipe integer.

* 1. Fungsi *Rand*

Fungsi *rand* ( ) berguna untuk mendapatkan nilai acak.. Bentuk pemanggilannya :

Rand ([ min [, max]])

Bila tidak ada argumen yang diberikan, fungsi ini akan menghasilkan bilangan acak yang terletak antara 0 dan RAND\_MAX. Namun, apabila argumen min dan max diberikan maka nilai acak yang diberikan antara min dan max.

* 1. Fungsi *srand*

Fungsi *srand* ( ) berguna untuk memberi umpan kepada pembangkit bilangan acak. Bentuk pemanggilannya :

srand(biji)

untuk memberi nilai biji yang acak, kita bisa menggunakan perintah sebagi berikut :

srand ((double)) microtime() \* 1000000

1. Jaringan Komputer

Menurut Nana Suarna (2007:9), jaringan komputer adalah kumpulan dari beberapa komputer, baik jaringan yang berskala kecil seperti di rumah atau di kantor maupun yang berskala besar seperti antar kota dan provinsi, atau jaringan yang mendunia (internasional) seperti antar benua atau antar dunia, dimana komputer-komputer tersebut saling berhubungan dan teroganisir (berintegrasi) antara komputer yang satu dengan komputer yang lain yaitu antara komputer server (sebagai induknya) dengan komputer terminal/*client* (sebagai anak buahnya).

1. Jenis-jenis jaringan komputer
2. Berdasarkan area, yaitu :
3. *Local Area Network* (*LAN*)

LAN (*Local Area Network*)merupakan arsitektur jaringan yang terdiri dari beberapa komputer. Pada arsitektur jenis LAN,komputeryang terhubung sangat sedikit. Ada yang menyebutkan bahwa jenis ini rata-rata hanya memiliki 24 *workstation* (PC yang terhubung pada jaringan) dalam satu jaringan LAN. Pada kenyataannya, apabila Anda memiliki dua komputer yang terhubung satu sama lain, kedua komputer itu dapat dikatakan telah terhubung dengan sistem jaringan lokal atau LAN. Antar komputer yang terhubung dalam sistem jaringan lokal dapat melakukan komunikasi, seperti *sharing* data maupun menggunakan sumber daya printer secara bersamaan. Dari penggambaran tersebut, arsitektur jaringan LAN dapat dicontohkan pada 2 unit komputer yang saling berkomunikasi dan dapat berbagi data dan sumber daya printer.

1. *Metropolitan Area Network* (*MAN*)

MAN merupakan arsitektur komputer yang kapasitas komputernya lebih banyak dari pada model LAN. Arsitektur ini disebut medium karena komputer yang terhubung tidak hanya berada dalam satu tempat atau satu ruangan saja. Jaringan disebut jenis MAN apa bila menghubungkan dua buah gedung yang sama-sama memiliki jaringan lokal (LAN). MAN dapat diartikan sebagai arsitektur jaringan yang di dalamnya terdapat dua atau lebih jaringan LAN yang dihubungkan menjadi satu.

1. *Wide Area Network* (*WAN*)

Selain LAN dan MAN yang dapat memuat beberapa komputer ada satu lagi yang disebut WAN. Pada jenis ini, komputer yang terhubung tidak hanya dalam satu ruangan, antar lantai dalam satu gedung ataupun antar gedung yang berdekatan. Akan tetapi, jenis ini memungkinkan hubungan jaringan lokal (LAN) dan MAN yang letaknya antarkota, antarpropinsi, antarpulau, bahkan antar benua. WANadalah jaringan yang lingkupnya biasanya sudah menggunakan sarana satelit ataupun kabel bawah laut. Dalam dunia TI (Teknologi Informasi), istilah WAN juga sering disebut *internet*. Di dalamnya Anda dapat menghubungkan banyak jaringan kecil maupun besar dengan media transmisi kabel sampai dengan tanpa kabel (*wireless*). Untuk menghubungkan antara jaringan satu dengan jaringan lain yang letaknya berbeda kota atau daerah, Anda dapat menggunakan sarana telepon sebagai media transmisinya.

1. Berdasarkan kebutuhan, yaitu :
2. Jaringan *peer-to-peer*

*Peer-to-peer* atau sering disebut *point-to-point* merupakan jenis jaringan yang tidak melibatkan sumber daya terlalu tinggi. Pada setiap *workstation* tidak memiliki batasan yang khusus dalam hal pengaksesan data dan penggunaan sumber daya. Setiap komputer/*workstation* yang terhubung tidak dibatasi oleh hak akses. Semua *workstation* yang terhubung dapat menggunakan semua data dan dapat mengirimkan data tanpa ada batasan. Pada bentuk *peer-to- peer*, semua *node* akan dianggap sebagai *workstation*. *Node-node* tersebut dapat bekerja secara bersama atau tidak bersama-sama dalam satu kesatuan waktu. Jadi, apabila salah satu komputer mati, komputer lain masih dapat bekerja karena tidak memiliki ketergantungan pada *workstation* lain. Dalam penggunaan sehari-hari, jenis jaringan *peer-to-peer* adalah model yang paling sering digunakan.

1. Jaringan *client* *server*

Komputer *client* merupakan komputer yang selalu memerlukan sumber daya berupa data maupun sistem dari komputer *server*. Komputer *server* merupakan komputer yang mampu menyediakan dan menerima permintaan yang dilakukan oleh komputer klien. Di dalam jenis ini, *client* dan *server* akan sangat berhubungan erat.

1. Berdasarkan media transmisinya, yaitu :
2. Jaringan menggunakan media kabel

Jenis jaringan komputer yang satu ini biasanya menggunakan media kabel sebagai media transmisinya ada Jenis jaringan ini disebut juga dengan jenis jaringan menggunakan media transmisi terpadu ( *Guided Transmission Media* ). beberapa kabel yang bisa digunakan untuk membuat model jaringan seperti ini diantaranya, kabel *Coaxial*, kabel *Twisted* *Pair* ( STP dan UTP ), serta kabel Fiber Optic.

1. Jaringan tanpa menggunakan kabel

Jenis jaringan komputer ini menggunakan sistem gelombang sebagai media Transmisinya, jenis jaringan ini juga disebut sebagai jenis jaringan yang menggunakan media transmisi tidak terpandu ( *Unguided transmission media* ), beberapa contoh media yang bisa digunakan antara lain : Gelombang Mikro, Satelit dan Sinar Infra Merah. (Bunafit Nugroho, 2005)

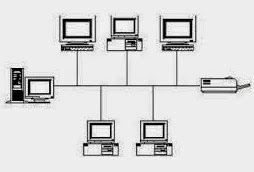
1. Topologi jaringan

Topologi jaringan komputer merupakan sebuah konsep atau cara yang digunakan untuk menghubungkan dua atau lebih komputer menjadi suatu jaringan yang saling terkoneksi. Setiap model dari topologi jaringan itu sendiri akan memiliki perbedaan antara lain perbedaan dari segi biaya untuk membuatnya, kecepatan dalam mentransfer data dan juga kemudahan dari segi pembuatan dan maintenennya.

Jenis-jenis topologi jaringaan antara lain :

1. Topologi *bus*

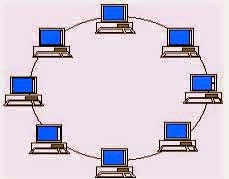
Pada model topologi *bus* komputer yang terhubung ke dalam sebuah jaringan akan tersusun rapi seperti antrian yang dihubungkan dengan satu kabel *coaxial* serta konektor BNC, kemudian pada ujung jaringan menggunakan terminator sebagai perangkat penutup.



Gambar 2.1 Topologi *bus*

1. Topologi *ring*

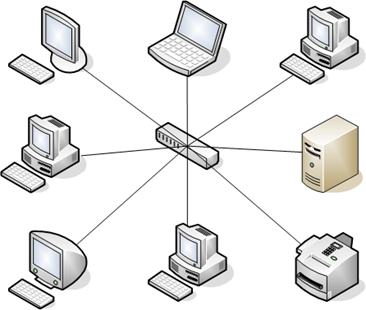
Pada model topologi *ring* setiap komputer akan dihubungkan dengan komputer lainnya secara terus menerus dan kemudian akan terhubung lagi ke komputer pertama, sehingga pada model jaringan ini akan membentuk lingkaran atau ring, topologi ini menggunakan kabel *coaxial* serta konektor BNC, untuk dapat berkomunikasi topologi ini menggunakan data token yang digunakan sebagai pengontrol hak akses komputer untuk menerima data yang dikirimkan. Contohnya apabila komputer 1 akan mengirimkan data ke komputer 3, maka data yang dikirim akan melewati komputer 2 dan kemudian komputer 3, apabila IP Andress komputer yang dituju tidak cocok maka data akan diteruskan sampai IP Address tujuan yang cocok.



Gambar 2.2 Topologi ring

1. Topologi *star*

Model topologi jaringan yang satu ini sangat unik karena membentuk seperti bintang. Untuk dapat menghubungkan komputer yang satu dengan komputer yang lain, topologi ini membutuhkan perangkat hub atau *switch* dan kabel yang digunakan adalah kabel jenis UTP dilengkapi konektor RJ45. Perangakat hub atau *switch* bertugas untuk mengontrol data dan menjadi pusat dari jaringan ini. Jadi apabila ada pengiriman data dari komputer 1 menuju ke komputer 2, maka hub atau switch yang menentukan IP Komputer mana yang akan dituju tanpa melewati komputer yang lainnya. Dewasa ini topologi ini paling sering digunakan karena banyak sekali kelebihanya dibanding model topologi jaringan *Bus* dan *Ring*.



Gambar 2.3 Topologi star

(<http://id.wikipedia.org/wiki/Topologi_jaringan>)

1. Perangkat keras jaringan

Untuk membangun sebuah jaringan komputer, tentu saja ada banyak hal yang harus diperhatikan, salah satunya adalah faktor hardware. Hardware atau perangkat keras merupakan elemen yang sangat vital dalam pembentukan jaringan komputer. Apabila kekurangan satu elemen hardware saja, maka dapat dipastikan suatu jaringan tidak akan dapat berjalan dengan sempurna. Berikut ini adalah beberapa hardware atau perangkat keras yang bisa digunakan pada jaringan komputer :

1. Komputer *server*

*server* merupakan salah satu perangkat yang sangat penting, karena server merupakan pusat dari jaringan komputer. Semua data penting yang nantinya akan disebarkan melalui jaringan internet semuanya berada pada komputer *server*. Komputer *server* ini nantinya akan saling terhubung dengan computer-komputer *client*, yang dapat mengakses data dari *server* tersebut. Fungsi utama dari *server* adalah sebagai *database* informasi yang nantinya akan dikirimkan dan juga disebarkan oleh jaringan. Semua data tersebut akan ditransmisikan melalui sistem jaringan agar nantinya dapat sampai ke komputer *client*. Kecanggihan fungsi CPU pada komputer *server* ini sangat penting dalam menunjang keandalan jaringan agar tidak terjadi komputer sering hang.



Gambar 2.4 Komputer server

1. Komputer *client*

Komputer *client* atau *user* sering juga dkenal dengan istilah terminal ataupun *workstation*. Secara umum, komputer *client* ini merupakan komputer umum yang digunakan untuk memperoleh data dari *server*. barat rantai makanan di dalam ilmu ekologi, komputer *server* adalah tanaman penyedia makanan, sedangkan komputer *client* adalah hewan yang memakan hasil buah dari tanaman tersebut. Artinya, komputer *client* ini di tugaskan untuk menarik data yang ada di komputer *server*. Komputer *client* yang sudah terhubung ke dalam jaringan, akan dapat memperoleh informasi dan juga akses terhadap komputer *server*. Komputer *client* ini dioperasikan secara manual oleh *user*-nya, yaitu manusia untuk berbagai macam kebutuhan, seperti mencari data, bermain game online, ataupun keperluan lainnya.



Gambar 2.5 Komputer *client*

1. *Hub*

Fungsi utama dari *hub* adalah untuk membagi jaringan dari satu *server* menuju ke *client* komputer dalam satu jaringan, terutama jaringan LAN alias lokal. Secara teoritis, *hub* sendiri adalah sebuah *hardware* atau perangkat keras yang merupakan suatu *central* *connection point* pada suatu jaringan, yang berfungsi untuk menerima sinyal dari *server* atau *host* dan kemudian mentransmisikannya ke *client* yang akan membentuk suatu jaringan. Dengan adanya *central connection point* ini, maka *hub* dapat mentransmisikan data dari *server* menuju lebih dari satu *client* yang terhubung dalam satu jaringan LAN. Apabila anda hanya ingin sekedar membagi satu jaringan lokal ke dalam beberapa komputer saja, maka hub merupakan salah satu perangkat keras yang tepat.



Gambar 2.6 Hub

1. *Router*

*Router* sendiri secara teknis memiliki fungsi menyalurkan koneksi internet melalui protokol TCP IP menuju komputer *client*. Secara spesifik fungsi *router* adalah untuk mengarahkan paket data atau informasi ke lokasi tertentu dari satu jaringan ke yang lain.  Ketika sebuah paket data yang dikirimkan dari jaringan, *router* berguna mengarahkan ke lokasi yang diperlukan melalui rute terbaik untuk mentransfer data tertentu.  *Router* menentukan rute terbaik dengan bantuan tabel forwarding, header dan protokol misalnya *Internet Control Message Protocol*(ICMP). Dalam menjalankan fungsinya,  Router menggunakan koneksi jenis RJ45 untuk menghubungkannya dengan komputer.  fungsi *router* yang  utama dari adalah untuk meneruskan sinyal dari komputer ke komputer sehingga tidak mengkonversi sinyal.  Fungsi *router* yang lain adalah sebagai alat keamanan seperti *firewall* yang diimplementasikan melalui *router*.  *Router* Juga tidak  harus terkoneksi ke  internet.  Tapi harus memerankan kewajibannya untuk membangun akses nirkabel ke Internet. Penggunaan *router* lainnya yang penting adalah menghubungkan dua atau lebih komputer untuk satu koneksi Internet. Secara khusus, fungsi *router* bisa seperti *access-point*, dimana bisa meneruskan koneksi internet dari *broadband* atau *provider* menuju lokasi komputer *client*. Namun demikian, pada dasarnya fungsi *router* jauh lebih banyak dari pada itu, karena *router* digunakan untuk memberikan rute atau jalan-jalan tertentu bagi paket data yang ditransmisikan.



Gambar 2.7 Router

1. *Access*-*point*

*Acces-point* hanya digunakan untuk memancarkan sinyal *wireless* yang diterima dari *router* ataupun *broadband* untuk membuat suatu jaringan WLAN atau *wireless* area *network*. Fungsi ini jauh lebih sederhana apabila dibandingkan dengan *router*, yang memiliki fungsi yang sangat kompleks dan banyak.



Gambar 2.8 *Access*-*point*

1. Modem

Modem merupakan kepanjangan dari modulator dan demodulator. Secara harfiah itu berarti modem merupakan suatu bentuk komunikasi dua arah yang dimana modem bisa melakukan proses perubahan data sinyal digital menjadi analog, dan kemudian bisa merubah kembali sinyal tersebut menjadi digital agar bisa digunakan di dalam komputer.



Gambar 2.9 Modem

1. NIC atau *ethernet card*

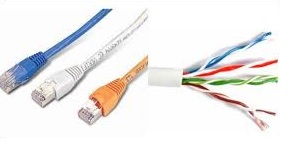
NIC merupakan kependekan dari *Network Interface Card*, yang merupakan suatu kartu yang juga kita kenal dengan istilah *ethernet card* yang memegang peranan penting dalam jaringan komputer. *Ethernet card* merupakan kartu jaringan yang ditanamnkan pada komputer, yang mana akan membuat sebuha komputer mampu terhubung ke dalam jaringan LAN. Dengan adanya *ethernet card* ini, maka setiap komputer bisa saling terhubung ke dalam jaringan dengan menggunakan koneksi kabel. Namun demikian, mungkin saat ini *ethernet card* sudah jarang digunakan, meskipun merupakan salah satu protokol standar dalam sebuah sistem komputer. Hal ini tidak lain dan juga tidak bukan adalah karena penemuan teknologi *wireless* yang lebih praktis dan juga mudah untuk diaplikasikan. Untuk lebih jelasnya mengenai *ethernet*, anda bisa membaca manfaat LAN card dan juga fungsi LAN Card.



Gambar 2.10 NIC

1. Kabel jaringan

Kabel berfungsi untuk membawa paket data dari *server* menuju *client*, tentunya dalam suatu jaringan yang bukan merupakan jaringan *wireless*. Biasanya, kabel yang digunakan adalah jenis kabel UTP atau *Unshielded* *Twisted Pair, Coaxial*, dan juga kabel Fiber Optic. Tentu saja penggunaan kabel dalam jaringan komputer sangat efisien, terutama ketika anda ingin membangun sebuah jaringan lokal, karena biaya yang lebih murah dibandingakn dengan perangkat *wireless*.



Gambar 2.11 Kabel jaringan

1. Konektor

Konektor merupakan alat yang dipasang pada masing-masing ujung kabel jaringan untuk menghubungkan adapter *network* dengan kabel. Berbagai jenis konektor jaringan ini harus disesuaikan dengan tipe dan jenis kabel jaringan yang dipakai.



Gambar 2.12 Konektor RJ45

(<http://www.jaringankomputer.org/perangkat-keras-jaringan-komputer/>)

1. Manfaat Jaringan

Secara umum, jaringan mempunyai beberapa manfaat yang lebih dibandingkan dengan komputer yang berdiri sendiri dan dunia usaha telah pula mengakui bahwa akses ke teknologi informasi modern selalu memiliki keunggulan kompetitif dibandingkan pesaing yang terbatas dalam bidang teknologi.

1. Jaringan memungkinkan manajemen sumber daya lebih efisien.

Misalnya, banyak pengguna dapat saling berbagi printer tunggal dengan kualitas tinggi, dibandingkan memakai printer kualitas rendah di masing-masing meja kerja. Selain itu, lisensi perangkat lunak jaringan dapat lebih murah dibandingkan lisensi *stand-alone* terpisah untuk jumlah pengguna sama.

1. Jaringan membantu mempertahankan informasi agar tetap andal dan *up-to-date*.

Sistem penyimpanan data terpusat yang dikelola dengan baik memungkinkan banyak pengguna mengaskses data dari berbagai lokasi yang berbeda, dan membatasi akses ke data sewaktu sedang diproses.

1. Jaringan membantu mempercepat proses berbagi data *(data sharing).*

*Transfer* data pada jaringan selalu lebih cepat dibandingkan sarana berbagi data lainnya yang bukan jaringan.

1. Jaringan memungkinkan kelompok-kerja berkomunikasi dengan lebih efisien.

Surat dan penyampaian pesan elektronik merupakan substansi sebagian besar sistem jaringan, disamping sistem penjadwalan, pemantauan proyek, konferensi *online* dan *groupware*, dimana semuanya membantu team bekerja lebih produktif.

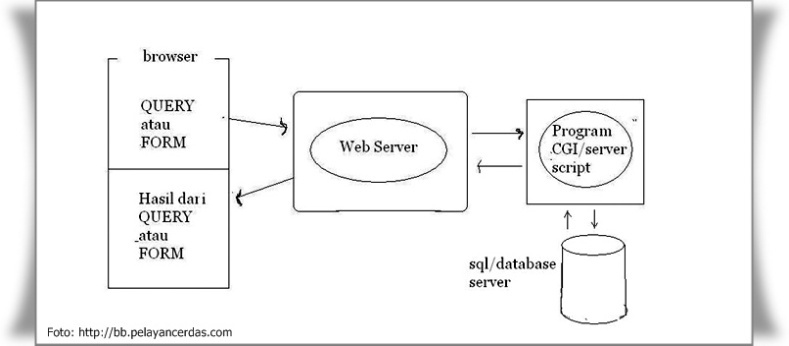
1. Jaringan membantu usaha dalam melayani klien mereka secara lebih efektif.

Akses jarak-jauh ke data terpusat memungkinkan karyawan dapat melayani klien di lapangan dan klien dapat langsung berkomunikasi dengan pemasok.

1. *World Wide Web (WWW)*

*World wide web* atau yang dikenal dengan www adalah suatu layanan internet yang menggunakan konsep *hypertext* (dokumen elektronik yang saling terkait). Bermula dari ide seorang ilmuwan komputer yang bekerja di CERN hingga akhirnya ide tersebut menjadi kenyataan pada tahun 1992, www telah diwujudkan menjadi sebentuk informasi yang dapat diakses melalui internet dimana file-file komputer (dokumen-dokumen *hypermedia*) disimpan dan kemudian diambil dengan cara-cara yang menggunakan metode penentuan alamat yang unik. Walaupun terdengar sangat umum ditelinga masyarakat, namun tidak semua orang mengerti tentang cara kerja www.

Cara kerja www mengikuti aturan - aturan tertentu untuk bisa berinteraksi antara *server* dan *client*. Gambaran tentang cara kerja www bisa dilihat pada gambar 2.13.



Gambar 2.13 cara kerja www

Cara kerja pertama kali adalah seseorang menggunakan suatu komputer atau pun perangkat terbaru seperti *Tablet* atau *Android* dan selainnya yang terkoneksi dengan internet, kemudian ia menggunakan salah satu aplikasi *browser* seperti *Mozilla Firefox* atau *Google Chrome*, lalu mengetikkan suatu alamat website dengan menggunakan www, misalnya *www.google.com*, maka selanjutnya tugas registarlah yang bekerja memerintahkan DNS *server* untuk menemukan alamat situs [*www.google.com*](http://www.google.com)*.*

Cara kerja selanjutnya adalah setelah DNS *server* menemukan alamat situs yang dicari, selanjutnya ia meminta kepada *hosting* server untuk menampilkan file-file yang diminta oleh pengguna internet tersebut. Dan terakhir, adalah tugas dari *browser* untuk menerima dan menampilakan file-file yang diminta yang terlebih dahulu diterjemahkan kode-kodenya agar tampilan filenya sesuai dengan yang diinginkan dan diharapkan oleh pengguna internet.

1. HTML

HTML adalah singkatan dari *HyperText Markup Language* adalah salah satu bahasa pemrograman *web design* dan juga biasa disebut *script* untuk menyusun dokumen-dokumen *Web*. Dokumen HTML disimpan dalam format teks reguler dan mengandung *tag-tag* yang memerintahkan *web browser* untuk mengeksekusi perintah-perintah yang dispesifikasikan. Struktur dasar HTML adalah sebagai berikut:

* 1. Struktur Dasar HTML

Struktur dasar HMTL adalah sebagai berikut:

|  |
| --- |
| <hmtl>  <head>  <title>judul dokumen HTML</title>  </head>  <body>  Penulisan Inforamasi web  </body>  </html> |

Dari struktur HTML diatas dapat dijelaskan sebagi berikut:

1. *Tag*

*Tag* adalah teks khusus (*markup*) berupa dua karakter “<”dan”>”, sebagai contoh <body> adalah *tag* dengan nama *body*. *Tag* harus ditulis secara berpasangan, yang teridiri atas *tag* pembuka dan *tag* penutup (ditambahkan karakter “/” setelah karatker “<”). sebagai contoh <body> ini adalah *tag* pembuka isi dokumen HTML, dan </body> ini adalah *tag* penutup isi dokumen HTML.

1. *Element*

*Element* teridiri dari tiga bagian, yaitu *tag* pembuka dan *tag* penutup. Sebagai contoh untuk menampilkan judul dokumen HTML pada *web browser* digunakan *element title*, dimana: <title> ini adalah *tag* pembuka judul dokumen HTML ini adalah isi judul dari dokumen HTML </title> ini adalah *tag* penutup judul dokumen HTML.

1. *Attribute*

*Attribute* mendefinisikan *property* dari suatu *element* HTML, yang terdiri atas nama dan nilai. Penulisannya adalah sebagai berikut:

|  |
| --- |
| <tag>  Nama-attr=”nilai-attr”  Nama-attr=”nilai-attr”  </tag> |

Secara umum nilai *attribute* harus berada dalam tanda petik satu atau dua.

1. *Element* HTML

Menyatkan pada *browser* bahwa dokumen *Web* yang digunakan adalah HTML.

|  |
| --- |
| <html>  …….  </html> |

1. *Element* TITLE

Merupakan judul dari dokumen HTML yang ditampilkan pada judul jendela *browser*. *Tag* <titile> dan *tag* </title> terletak diantara *tag* <head > dan *tag* </head>.

|  |
| --- |
| <title>  …….  </title> |

1. *Element* BODY

*Element* ini untuk menampilkan isi dokumen HTML. *Tag* <body> dan *tag* </body> terletak di bawah *tag* <head> dan *tag* </head>. *Element* BODY mempunyai *attribute*-*attribute* yang menspesifikasikan khususnya warna dan latar belakang dokumen yang akan ditampilkan pada *browser*.

(<https://www.academia.edu/6329892/Struktur_Dasar_HTML_Struktur_Dasar_HTML_Lisensi_Dokumen>)

1. CSS

*Cascading Style Sheet (CSS)* merupakan kumpulan *syntax* untuk mengatur elemen-elemen pada dokumen *HTML* yang akan ditampilkan. *CSS* pertama kali dijadikan standar dalam desain *website* melalui dokumen *HTML* pada *HTML 4.0*. Ekstensi berkas *CSS* adalah *.css*.

* 1. Penuliasan kode *CSS*

CSS didefinisikan langsung pada tag HTML yang bersangkutan. Cara penulisannya cukup dengan menambahkan atribut ***style = “......”***dalam tag *HTML* tersebut. *Style* hanya akan berlaku pada tag yang bersangkutan, dan tidak akan memengaruhi tag *HTML* yang lain.

* 1. Pemasangan *CSS* dengan *HTML*

Ada tiga cara untuk memasukkan sebuah *style sheet*:

* + 1. Eksternal *Style Sheet*

Sebuah *style sheet* eksternal sangat ideal bila gaya diterapkan pada banyak halaman. Dengan *style sheet* eksternal, Anda dapat mengubah tampilan seluruh situs Web dengan mengubah satu file. Setiap halaman harus link ke *style* *sheet* menggunakan *tag* <*link*>. *Tag* <*link*> masuk ke dalam bagian kepala:

|  |
| --- |
| <head> <link rel=”stylesheet” type=”text/css” href=”mystyle.css” />  </head> |

Sebuah *style sheet* eksternal dapat ditulis dalam editor teks apapun. File tidak boleh mengandung *tag* *html*. *Style sheet* Anda harus disimpan dengan ekstensi *css..* Sebuah contoh dari sebuah file style sheet adalah sebagai berikut:

|  |
| --- |
| hr{color:sienna;} p{margin-left:20px;} body{background image:url(“images/back40.gif”);} |

Jangan menambahkan spasi antara nilai properti dan unit (seperti *margin-left:* 20 px). Cara yang benar adalah: *margin-left:* 20px.

* + 1. Internal *Style Sheet*

Sebuah *style sheet* internal harus digunakan bila dokumen tunggal memiliki gaya yang unik. Anda mendefinisikan gaya internal di bagian kepala halaman *HTML*, dengan menggunakan *tag style*, seperti ini:

|  |
| --- |
| <head> <style type=”text/css”>  hr{color:sienna;}  p{margin-left:20px;}  body{background- image:url(“images/back40.gif”);}  </style>  </head> |

* + 1. *Inline Style*

*Inline Style* yaitu menuliskan secara langsung script CSS kedalam *tag* HTML

|  |
| --- |
| <p style=”color:sienna;margin-left:20px”> This is a paragraph.</p> |

1. PHP

Berikut beberapa definisi PHP (*Personal Home Page*) menurut Agus Saputra (2013) adalah sebagai berikut :

1. PHP merupakan bahasa pemrograman yang difungsikan untuk membagun *website* dinamis, PHP menyatu dengan kode HTML, maksudnya adalah beda kondisi. HTML digunakan sebagai pembangun atau pondasi dari kerangka *layout web*, sedangkan PHP difungsikan sebagai prosesnya, sehingga dengan adanaya PHP tersebut sebuah *web* akan lebih mudah di-*maintenance.*
2. PHP merupakan bahasa pemrograman yang berjalan di sisi *server,* sehingga PHP disebut juga bahasa *server side scripting,* artinya bahwa dalam setiap menjalankan *script* PHP, wajib membutuhkan *web server* dalam menjalankannya.
3. PHP adalah bahasa perograman yang bersifat *open source,* sehingga dapat dipakai secara cuma-cuma, dan mampu lintas *platform*, yaitu dapat berjalan pada sistem operasi *windows* maupun *linux.* PHP juga dibangun sebagai modul pada *web server apache* dan sebagai *binary* yang dapat berjalan sebagai CGI.

Dalam merancang sistem aplikasi *test* masuk ini penulis menggunakan bahasa peerograman PHP versi 5.3.1 dengan struktur penulisan sebagai berikut:

1. *Case Sensitivity* (Perbedaan huruf besar dan kecil) dalam PHP

PHP tidak membedakan huruf besar dan kecil *(****c****ase**insensitive*) untuk penamaan fungsi **(**function**)**, nama *class*, maupun keyword bawaan PHP seperti *echo*, *while*, dan *class.* Ketiga baris berikut akan dianggap sama dalam PHP:

|  |
| --- |
| <?php  Echo “Hello World”;  Echo “Hello World”;  Echo “Hello World”;  ?> |

Akan tetapi, *PHP* membedakan huruf besar dan huruf kecil (***c****ase sensitive*) untuk *penamaan variabel*, sehingga $nama, $Nama dan $NAMA akan dianggap sebagai 3 variabel yang berbeda. Sering kali error terjadi dikarenakan salah menuliskan nama variabel, yang seharusnya menggunakan huruf kecil, ditulis dengan huruf besar.

|  |
| --- |
| <?php  $andi="Andi";  echo $Andi; // Notice: Undefined variable: Andi  ?> |

Untuk mengatasi perbedaan ini, disarankan menggunakan huruf kecil untuk seluruh kode PHP, termasuk variabel, fungsi maupun *class*. Jika membutuhkan nama variabel yang terdiri dari 2 suku kata, karakter spasi bisa digantikan dengan *underscore* (\_).

1. Penulisan Baris Perintah dalam PHP

Statement (baris perintah) di dalam PHP adalah kumpulanperintahPHPyang menginstruksikanPHPuntuk melakukansesuatu. Baris perintah ini bisa terdiri dari satu baris singkat (seperti perintah ***echo*** untuk menampilakan text di layar) atau bisa sesuatu yang lebih rumit dan terdiri dari beberapa baris, seperti kondisi*if*, atau kode perulangan (*loop*).

Berikut adalah contoh beberapa baris perintah dalam PHP:

|  |
| --- |
| <?php  echo "Hello, world";  sebuah\_fungsi(21, "duniailkom");  $a = 1;  $nama = "duniailkom";  $b = $a / 25.0;  if ($y == $z) {  echo "Tampilkan Tabel";  }  ?> |

Terlihat dari beberapa contoh baris perintah diatas, PHP menggunakan tanda *semicolon* (titik koma) “;” sebagai tanda akhir baris perintah. Kumpulan baris perintah yang menggunakan tanda kurung kurawal seperti kodisiIF atau perulangan(*loop*) tidak membutuhkan tanda titik koma setelah kurung penutup.

|  |
| --- |
| <?php  if (true) {  echo "Perintah dijalankan";  // tanda titik koma harus ditulis  } // tidak diperlukan tanda titik koma setelah tanda kurung kurawal  ?> |

1. Karakter Spasi dan Tab dalam PHP

Secara umum, karakter spasi dan tab diabaikan di dalam eksekusi program *PHP*. Anda boleh memecah sebuah statement menjadi beberapa baris, atau menyatukan beberapa statement dalam sebuah baris yang panjang. Seperti contoh berikut:

|  |
| --- |
| <?php  echo "Ini kalimat pertama"; echo "Ini kalimat kedua";  $nama="duniailkom";  ?>  Baris perintah itu sama artinya dengan  <?php  echo "Ini kalimat pertama";  echo "Ini kalimat kedua";  $nama = "duniailkom";  ?> |

1. *Javascript*

Naista (2016) *Javascript* adalah bahasa skrip *client side* yang populer di internet dan dapat bekerja di sebagian *besar web browser* populer seperti *Internet Explore* (IE*), Muzilla Firefox, Netscape* dan *Opera.*

Kode *javascript* dapat disisipkan dalam halaman *web* menggunakan *tag* *<script>.* Dengan adanya *javascript* ini, kita bisa membuat *website* yang kita buat menjadi lebih menarik dengan animasi-animasi yang bisa dibuat menjadi interaktif, contohnya ketika kita ingin login, kemudian muncul *alert* ketika kita kosongkan *fieldnya*, dan lain-lain.

*Javascript* versi 5 inilah yang saat ini menjadi standar untuk web browser modern seperti Internet *Explorer*, *Google Chrome, Firefox, Opera, dan Safari*. Yang penulis gunakan mengikuti dari web *browser* yang digunakan di laptop dengan struktur penulisan sebagi berikut:

*JavaScript* biasanya dituliskan dalam bentuk [fungsi](https://id.wikipedia.org/wiki/Fungsi) yang ditaruh di tag <head> yang dibuka dengan tag berikut contohnya.

|  |
| --- |
| [tag](https://id.wikipedia.org/wiki/Tag) <script type="text/javascript">  <script type="text/javascript">  alert("Halo Dunia!");  </script>s |

1. *Type – Type* Data Pada *JavaScript*
2. Skrip *head*

Skrip ini akan dieksekusi ketika dipanggil (biasanya berbentuk *function*) atau dipanggil berdasarkan *trigger* pada *event* tertentu. Peletakkan skrip di *head* akan menjamin skrip dimuat terlebih dahulu sebelum dipanggil.

Berikut Contoh dari Skrip di *head*

|  |
| --- |
| <html>  <head>  <script type="text/javascript">  ...  </script>  </head>  </html> |

1. Skrip *body*

Skrip ini dieksekusi ketika halaman dimuat sampai di bagian *<body>.* Ketika menempatkan skrip pada bagian *<body>* berarti antara isi dan *JavaScript* dijadikan [satu](https://id.wikipedia.org/wiki/Satu) bagian.

Berikut Contoh dari Skrip di *body*

|  |
| --- |
| <html>  <head>  </head>  <body>  <script type="text/javascript">  ...  </script>  </body>  </html> |

1. Skrip eksternal

Terkadang ada yang menginginkan menjalankan *JavaScript* yang sama dalam beberapa kali pada halaman yang berbeda, tetapi tidak mau disibukkan jika harus [menulis](https://id.wikipedia.org/wiki/Menulis) ulang *script* yang diinginkan di setiap halaman.Maka *JavaScript* dapat ditulis di file secara eksternal.Jadi, antara dokumen [HTML](https://id.wikipedia.org/wiki/HTML) dan *JavaScript* dipisahkan, kemudian berkas tersebut dipanggil dari *dokument* HTML. Berkas *JavaScript* tersebut disimpan dengan ekstensi .js. *JavaScript : js/xxx.js* *document.write* ("pesan ini tampil ketika halaman diload");. Untuk menggunakan eksternal *JavaScript* (.js) dipakai atribut *"src"* pada *tag* <*script*> pada halaman HTML-nya.

Berikut Contoh dari Skrip di eksternal

|  |
| --- |
| <html>  <head>  </head>  <body>  <script src="xxx.js">  </script>  <p>Script di atas berada di berkas "xx.js" (eksternal) </p>  </body>  </html> |

1. Xampp

XAMPP merupakan singkatan dari X (empat sistem operasi apapun), *Apache*, MySQL, PHP dan Perl. XAMPP sendiri adalah perangkat lunak bebas, yang mendukung banyak sistem operasi, merupakan kompilasi dari beberapa program. (<https://id.wikipedia.org/wiki/XAMPP>)

1. *Apache*

Server HTTP *Apache* atau server *web*/www *Apache* adalah server *web* yang dapat dijalankan di banyak sistem operasi (Unix, BSD, Linux, Microsoft Windows dan Novell Netware serta *platform* lainnya) yang berguna untuk melayani dan memfungsikan situs *web*. Protokol yang digunakan untuk melayani fasilitas *web*/www ini menggunakan HTTP.

<https://id.wikipedia.org/wiki/Apache_HTTP_Server>

1. MySQL

Menurut Abdul Kadir (2008) dalam bukunya yang berjudul *Dasar Pemrograman WEB Dinamis Menggunakan PHP*, mendefinisikan MySQL sebagai berikut :

MySQL adalah salah satu jenis *databases* *server* yang sangat terkenal. Populerannya disebabkan MySQL menggunakan SQL sebagai bahasa dasar untuk mengakses *database-*nya. Selain itu, ia bersifat *open* souce. MySQL termasuk jenis RDBMS *(Relation Databases Management System ).*

Tipe-tipe data pada MySQL

1. Tipe Numerik

Tipe data numerik digunakan untuk menyimpan data *numeric* (angka). Berikut ini beberapa tipe *field* (kolom) di MySQL yang termasuk kedalam kelompok tipe *numerik* dapat dilihat pada tabel di bawah ini.

Tabel 2.1. Tipe Data *Numerik* (Angka)

|  |  |  |  |
| --- | --- | --- | --- |
| **No** | **Tipe Data** | **Penggunaan** | **Nilai** |
| 1 | Int | Digunakan untuk menyimpan bilangan bulat positif dan negatif | -2.147.483.648 s/d 2.147.483.647 |
| 2 | Float | Digunakan untuk menyimpan data bilangan pecahan positif dan negatif presisi tunggal | 3.402823466E+38 s/d -1.175494351E- 38,0, dan 1.175494351E-38 s/d 3.402823466E+38 |

Sumber : [Firdaus](https://id.wikipedia.org/wiki/MySQL) (2007)

1. Tipe *Date* dan *Time*

Tipe data *date* dan *time* digunakan untuk menyimpan data tanggal dan waktu. Berikut ini tipe *field* (kolom) di MySQL yang termasuk kedalam kelompok tipe *date* dan *time* dapat dilihat pada tabel 2.2.

Tabel 2.2. Tipe Data *Date* dan *Time*

|  |  |  |  |
| --- | --- | --- | --- |
| **No** | **Tipe Data** | **Penggunaan** | **Nilai** |
| 1 | Date | Digunakan untuk menyimpan data tanggal | 1000-01-01 s/d 9999-12-31 (yyyy-mm-dd) |
| 2 | Time | Digunakan untuk menyimpan data waktu | -838:59:59 s/d +838:59:59 (hh:mm:ss) |
| 3 | Datetime | Digunakan untuk menyimpan data tanggal dan waktu | ‘1000-01-01 00:00:00’ s/d ‘9999-12-3123:59:59’ |
| 4 | Year | Digunakan untuk menyimpan dara tahun dari tanggal | 1900 s/d 20155 |

Sumber : [Firdaus](https://id.wikipedia.org/wiki/MySQL) (2007

1. Tipe *String* (*Text*)

Tipe data *string* digunakan untuk menyimpan data *string* (*text*), berikut tipe *filed* (kolom) di MySQL yang termasuk ke dalam kelompok tipe *string*, Adapun tipe *string* yang penulis gunakan yaitu:

Tabel 2.3. Tipe Data *String* (*Text*)

|  |  |  |  |
| --- | --- | --- | --- |
| **No** | **Tipe Field** | **Penggunaan** | **Nilai** |
| 1 | Char | Digunakan untuk menyimpan data string ukuran tetap | 0 s/d 255 karakter |
| 2 | Varchar | Digunakan untuk menyimpan data string ukuran dinamis | 0 s/d 255 karakter (v4.1), 0 s/d 65.535 (v5.0.3) |
| 3 | Text | Digunakan untuk menyimpan data text | 0 s/d 65.535 (216-1) karakter |

Sumber : [Firdaus](https://id.wikipedia.org/wiki/MySQL) (2007)

1. Tipe Data yang lain

Selain tipe data diatas, MySQL juga menyediakan tipe data yang lain, berikut ini beberapa tipe data tambahan MySQL yaitu:

Tabel 2.4. Tipe Data yang lain

|  |  |  |  |
| --- | --- | --- | --- |
| **No** | **Tipe Field** | **Penggunaan** | **Nilai** |
| 1 | Enum | Enumerasi (Kumpulan Data) | 0 s/d 65.535 (216-1) karakter |

Sumber : [Firdaus](https://id.wikipedia.org/wiki/MySQL) (2007)

### *Web Browser*.

Agar pengguna dapat menampilkan situs *web*, dibutuhkan program aplikasi berupa *web browser* (umumnya disebut *browser*). *Browser* diartikan sebagai sebuah aplikasi yang berfungsi untuk menghubungi, mengambil, dan menampilkan informasi yang terdapat pada *website*. Pengguna memasukkan *Uniform Resource Locator (URL)* pada kolom alamat yang terdapat pada *browser* yang kemudian disambungkan oleh *browser* ke alamat *server website* yang dituju untuk selanjutnya meminta dan menampilkan data berdasarkan *Uniform Resource Identifier (URI)*.

* 1. Macam-macam *Web Browser*

1). *Mozilla Firefox*

*Mozilla Firefox* (aslinya bernama *Phoenix* dan kemudian untuk sesaat dikenal sebagai *Mozilla Firebird*) adalah penjelajah *web* antar *platform* gratis yang dikembangkan oleh Yayasan Mozilla dan ratusan sukarelawan.

2). *Google Chrome*

*Google Chrome* adalah sebuah penjelajah web *open source* yang dikembangkan oleh *Google* dengan menggunakan mesin *rendering* *WebKit*. Proyek *open source* ini dinamakan *Chromium*.

Fitur-fitur *google chrome* antara lain :

1. *Google Chrome* terbuka dengan sangat cepat dari *desktop* Anda.
2. *Google Chrome* memuat laman web dalam waktu singkat.
3. *Google Chrome* menjalankan laman web interaktif, aplikasi web, dan *JavaScript* lebih cepat dari sebelumnya.
4. *Google Chrome* sengaja dirancang untuk efisiensi, simple dan mudah digunakan:
5. Menelusur dan menavigasi ke laman *web* dari kotak yang sama.
6. Mengatur dan mengorganisasi tab semau Anda dengan cepat dan mudah.
7. Memasuki situs web favorit Anda hanya dengan satu klik dari *thumbnail* situs yang sering Anda kunjungi pada laman Tab Baru.

3). *Internet Explorer*

*Internet Explorer* disingkat IE atau MSIE, adalah sebuah penjelajah web dan perangkat lunak tak bebas yang gratis dari *Microsoft*. IE tersedia untuk banyak versi *Microsoft Windows*, namun *Microsoft* telah berhenti mengeluarkan versi mutakhir untuk semua *platform* kecuali *Windows XP*. Untuk beberapa waktu lamanya, *Microsoft* juga mengeluarkan *Internet Explorer* untuk *Mac* (yang berdasarkan mesin pengarsir yang berbeda) dan versi-versi untuk penggunaan melalui X *Window System* pada Solaris dan HP-UX, tetapi kini mereka sudah tidak dikembangkan.

4). *Opera*

*Opera* adalah penjelajah web dan paket perangkat lunak Internet antar *platform*. Opera terdiri dari kumpulan perangkat lunak untuk Internet seperti penjelajah web, serta perangkat lunak untuk membaca dan mengirim surat elektronik. Opera dibuat oleh *Opera Software* yang bermarkas di Oslo, Norwegia. Opera dapat dijalankan di berbagai sistem operasi, termasuk *Microsoft Windows, Mac OS X, Solaris, FreeBSD* dan *Linux*.  
Pengembangan Opera dimulai pada tahun 1994. Kala itu Opera merupakan sebuah proyek penelitian di Telenor, perusahaan telekomunikasi terbesar di Norwegia. Hingga Juli 2006, *Opera* telah mencapai versi 9 yang bisa diunduh gratis. Alasan penggunannya memilih *Opera* adalah karena *Opera* dianggap ringan dan cepat dibanding *Internet Explorer*.

5). *Safari*

*Safari* adalah sebuah penjelajah web buatan Apple Inc. yang awalnya ditujukan khusus bagi sistem operasi Mac OS. Safari dipasang bersama Mac OS X dan merupakan penjelajah web bawaan (default) di sistem operasi tersebut sejak Mac OS X v10.3. Sebelumnya dari tahun 1997 hingga 2003, Mac OS X menggunakan *Internet Explorer for Mac* sebagai penjelajah web bawaan.

1. UML (*Unified Modelling Language)*

UML (*Unified Modeling Language*) adalah bahasa untuk menspesifikasi, memvisualisasikan, serta mengkonstruksi bangunan dasar dari sistem perangkat lunak, termasuk di dalamnya dengan melibatkan pemodelan aturan-aturan bisnis (Nugroho, 2004:105). Menurut Hariyanto (2004:259) UML (*Unified Modelling Language*) adalah bahasa grafis untuk mendokumentasi, menspesifikasikan dan membangun sistem perangkat lunak. UML berorientasi objek menerapkan banyak level abstraksi, tidak bergantung proses pengembangan, tidak bergantung bahasa dan teknologi, pemaduan beberapa notasi diberagam metodologi, pengembangan UML dimulai dari kerjasama Grady Booch dan James Rumbauh pada 1994 untuk mengkombinasikan dua metodologi terkenal Booch dan OMT, kemudian Ivar Jacobson pencipta metode OOSE (*Object Oriented Software Engineering*) bergabung. Di dalam UML memiliki diagram data notasi untuk menjelaskan secara visual mengenai elemen-elemen pemodelan antara lain.

* 1. *Use case diagram*

*Use case diagram* menggambarkan fungsionalitas yang diharapkan dari sebuah sistem. Yang ditekankan adalah ”apa” yang diperbuat sistem, bukan “bagaimana”. Sebuah *use case* mempresentasikan sebuah interaksi antara aktor dan sistem. Simbol-simbol *use case* *diagram* diperlihatkan pada tabel 2.5.

Tabel 2.5. Simbol-simbol *Use* *Case* *Diagram*

|  |  |  |  |
| --- | --- | --- | --- |
| **No** | **Simbol** | **Nama** | **Deskripsi** |
| 1 |  | *Use* *Case* | Fungsionalitas yang disediakan sistem sebagai unit-unit yang saling bertukar pesan antar unit atau aktor; biasanya dinyatakan dengan menggunaka kata kerja diawal *frase* nama *use case*. |
| 2 | nama\_aktor | Aktor | Orang, proses atau sistem lain yang berinteraksi dengan sistem informasi yang akan dibuat diluar sistem informasi yang akan dibuat itu sendiri. |
| 3 |  | Asosiasi/ *Association* | Komunikasi antara aktor dan *use case* yang berpartisipasicpada *use case* atau *use case* memiliki interaksi dengan aktor. |
| 3 | <<extend>> | Ekstensi/*extend* | Relasi *use case* tambahan ke sebuah *use case* yang dimana *use case* yang ditambahkan dapat berdiri sendiri walaupun tanpa *use* *case* tambahan itu. |
| 4 |  | Generalisasi/ *generalization* | Hubungan generalisasi dan spesialisasi (umum-khusus) antara dua buah *use case* dimana fungsi yang satu adalah fungsi yang lebih umum dari lainnya. |
| 5 | <<extend>>  <<uses>> | *Unclude/uses* | Relasi *use case* tambahan ke sebuah *use case* dimana *use* *case* yang ditambahkan Memerlukan *use case* ini untuk menjalankan fungsinya atau sebagai syarat dijalankan *use case* ini. |

Sumber :<http://site.iugaza.edu.ps/lalsaedi/files/2010/02/(Ebook_->\_Pdf)\_Visual\_Modeling\_With\_Rational\_Rose\_2000\_And\_Uml\_(Addison-Wesley).pdf (Diakses Tanggal 22 Juni 2016)

* 1. *Class* *Diagram*

*Class Diagram* menggambarkan struktur dan deskripsi *Class, Packge* dan objek berserta hubungan satu sama lain seperti pewarisan, asosiasi, dan lain-lain. *Class* memiliki tiga area pokok yaitu nama, atribut, metode. *Class Diagram* membantu dalam visualisasi struktur kelas-kelas dari suatu sistem dan merupakan tipe diagram yang paling banyak dipakai. *Class Diagram* memperlihatkan hubungan antar kelas dan penjelasan detail tiap-tiap kelas di dalam model desain dari suatu sistem. *Class* *diagram* disimbolkan seperti pada tabel 2.6.

Berikut adalah simbol-simbol yang ada pada *class* *diagram*

Tabel 2.6. Simbol-simbol *class* *diagram*

|  |  |  |  |
| --- | --- | --- | --- |
| **No** | **Simbol** | **Nama** | **Deskripsi** |
| 1 | |  | | --- | | Nama Kelas | | +atribut | | +operasi() | | Kelas | Kelas pada struktur Sistem |
| 2 | nama\_*interface* | Antarmuka/*Interface* | Sama dengan konsep *interface* dalam pemrograman berorientasi objek |
| 3 |  | Asosiasi/*association* | Relasi antar kelas dengan makna umum, asosiasi biasanya juga disertai dengan *multiplicity*. |
| 4 |  | Assosiasi berarah/*directed* *association* | Relasi antar kelas dengan makna kelas yang satu digunakan oleh kelas yang lain, asosiasi biasanya juga disertai dengan *multiplicity*. |
| 5 |  | Generalisasi | Relasi antar kelas dengan makna generalisasi spesialisasi (umum khusus). |
| 6 |  | Kebergantungan/ *dependency* | Relasi antar kelas dengan makna kebergantungan antar kelas. |
| 7 |  | Agregasi/*aggregation* | Relasi antar kelas dengan makna semua bagian (*whole-part*). |

Sumber :<http://site.iugaza.edu.ps/lalsaedi/files/2010/02/(Ebook_->\_Pdf)\_Visual\_Modeling\_With\_Rational\_Rose\_2000\_And\_Uml\_(Addison-Wesley).pdf (Diakses Tanggal 22 Juni 2016)

* 1. *Activity Diagram*

*Activity Diagram* menggambarkan berbagai alur aktivitas dalm sistem yang sedang dirancang, bagaimana masing-masing aliran berawal, *decision* yang mungkin terjadi, dan bagaimana mereka berakhir. *Activity Diagram* juga dapat menggambarkan proses paralel yang mungkin terjadi pada beberapa eksekusi. Berikut adalah simbol-simbol yang ada pada *Activity Diagram*.

Tabel 2.7. Simbol-simbol *Activity* *Diagram*

|  |  |  |  |
| --- | --- | --- | --- |
| **No** | **Simbol** | **Nama** | **Deskripsi** |
| 1 |  | Status awal | Status awal aktivitas sistem. |
| 2 | x | Aktivitas | Aktivitas yang dilakukan sistem. |
| 3 |  | Percabangan/*decision* | Asosiasi percabangan dimana jika ada pilihan aktivitas lebih dari satu. |
| 4 |  | Penggabungan/join | Asosiasi penggabungan dimana lebih dari satu aktivitas digabungkan menjadi satu. |
| 4 |  | Status akhir | Status akhir yang dilakukan sistem. |
| 5 | Nama Swimlane | Swimlane | Memisahkan organisasi bisnis yang bertanggung jawab terhadap aktivitas yang terjadi. |

Sumber:<http://site.iugaza.edu.ps/lalsaedi/files/2010/02/(Ebook_->\_Pdf)\_Visual\_Modeling\_With\_Rational\_Rose\_2000\_And\_Uml\_(Addison-Wesley).pdf (Diakses Tanggal 22 Juni 2016)