

BAB II

LANDASAN TEORI

2.1. Konsep Dasar Sistem

Terdapat dua kelompok pendekatan di dalam mendefinisikan sistem, yaitu yang menekankan pada prosedurnya dan yang menekankan pada komponen atau elemennya. Pendekatan sistem yang lebih menekankan pada prosedur. Menurut Baridwan dalam (Purnama, 2016) bahwa “Suatu sistem adalah suatu jaringan kerja dari prosedur-prosedur yang saling berhubungan, berkumpul bersama-sama untuk melakukan suatu kegiatan atau untuk menyelesaikan suatu sasaran yang tertentu”.

A. Pengertian Sistem

Sistem digunakan organisasi dan perusahaan sebagai sekumpulan entitas yang memiliki hubungan dan menjamin kelancaran pengguna sistem tersebut. Sistem memudahkan aliran informasi, materi, energi atau keluaran sistem untuk mencapai tujuan yang ingin dicapai.

Menurut (Hutahaeen, 2015) mengemukakan bahwa “Sistem adalah suatu jaringan kerja dari prosedur-prosedur yang saling berhubungan, berkumpul bersama-sama untuk melakukan kegiatan atau untuk melakukan sasaran yang tertentu”.

Maka dari itu, dapat ditarik sebuah kesimpulan bahwa sistem merupakan sekumpulan elemen/entitas/komponen yang berada dalam suatu jaringan kerja, saling berinteraksi, berhubungan, dan ketergantungan antar komponen untuk mencapai suatu tujuan yang menghasilkan keluaran (*output*).

B. Karakteristik Sistem

“Suatu sistem dapat dikatakan sebagai sistem yang baik apabila memiliki karakteristik-karakteristik tertentu” (Hutahaeen, 2015). Karakteristik sistem yang dimaksud, yaitu:

1. Komponen (*component*)

Komponen-komponen sistem tersebut dapat berupa suatu bentuk subsistem.

Setiap subsistem memiliki sifat dari sistem yang menjalankan suatu fungsi tertentu dan mempengaruhi proses sistem secara keseluruhan.

2. Batasan sistem (*boundary*)

Ruang lingkup sistem merupakan daerah yang membatasi antara sistem dengan sistem yang lain atau sistem dengan lingkungan luarnya. Batasan sistem ini memungkinkan suatu sistem dipandang sebagai satu kesatuan yang tidak dapat dipisahkan.

3. Lingkungan luar sistem (*environment*)

Bentuk apapun yang ada diluar ruang lingkup atau batasan sistem yang mempengaruhi operasi sistem tersebut disebut lingkungan luar sistem. Lingkungan luar sistem ini dapat bersifat menguntungkan dan dapat juga bersifat merugikan sistem tersebut.

4. Penghubung sistem (*interface*)

Media yang menghubungkan sistem dengan subsistem yang lain disebut penghubung sistem atau *interface*. Penghubung ini memungkinkan sumberdaya mengalir dari satu subsistem ke subsistem lain. Bentuk keluaran dari satu subsistem akan menjadi masukan untuk subsistem lain melalui penghubung tersebut. Dengan demikian, dapat terjadi suatu integrasi sistem yang membentuk satu kesatuan.

5. Masukan sistem (*input*)

Energi yang dimasukkan ke dalam sistem disebut masukan sistem, yang dapat berupa pemeliharaan (*maintenance input*) dan sinyal (*signal input*).

6. Keluaran sistem (*output*)

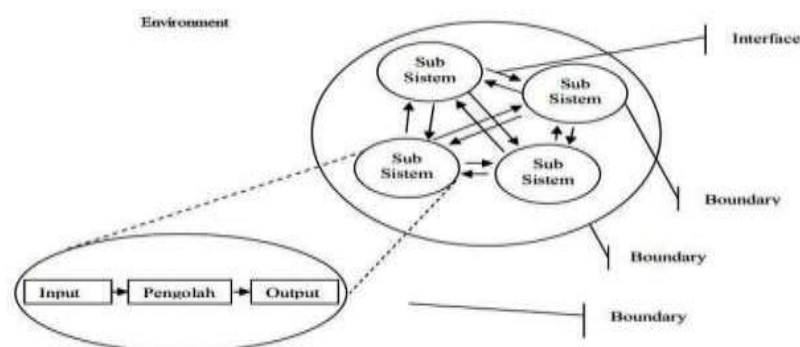
Hasil energi yang diolah dan diklasifikasikan menjadi keluaran yang berguna. Keluaran ini dapat menjadi masukan bagi subsistem yang lain seperti system informasi. Keluaran yang dihasilkan adalah informasi. Informasi ini dapat digunakan sebagai masukan untuk pengambilan keputusan atau hal-hal lain yang menjadi input bagi subsistem lain.

7. Pengolah sistem

Suatu sistem dapat mempunyai suatu proses yang akan mengubah masukan menjadi keluaran.

8. Sasaran sistem

Suatu sistem mempunyai tujuan dan sasaran yang pasti dan bersifat deterministik. Kalau suatu sistem tidak memiliki sasaran maka operasi sistem tidak ada gunanya. Suatu sistem dikatakan berhasil bila mengenai sasaran atau tujuan yang telah direncanakan.



Sumber: (Hutahaean, 2015)

Gambar II.1.

Karakteristik Dari Suatu Sistem

C. Klasifikasi Sistem

“Sistem juga diklasifikasikan dari beberapa sudut pandang sistem” (Hutahaeen, 2015). Klasifikasi sistem tersebut terdiri dari:

1. Sistem Abstrak dan Sistem Fisik

Sistem abstrak merupakan sistem yang berupa pemikiran atau ide tidak tampak secara fisik, misalnya sistem telogi. Sedangkan sistem fisi diartikan sebagai sistem yang nampak secara fisik sehingga setiap makhluk dapat melihatnya, misalnya sistem komputer.

2. Sistem Alamiah dan Sistem Buatan Manusia

Sistem alamiah merupakan sistem yang terjadi melalui proses alam, tidak dibuat oleh manusia, misalnya misalnya sistem tata surya, sistem galaksi, sistem reproduksi dan lain-lain. Sedangkan sistem buatan manusia merupakan system yang dirancang oleh manusia. Sistem buatan yang melibatkan interaksi manusia, misalnya sistem akuntansi, sistem informasi, dan lain-lain.

3. Sistem Deterministik dan Sistem Probabilistik

Sistem deterministik merupakan sistem yang beroperasi dengan tingkah laku yang sudah dapat diprediksi. Interaksi bagian-bagiannya dapat dideteksi dengan pasti sehingga keluaran dari sistem dapat diramalkan, misalnya sistem komputer, adalah contoh sistem yang tingkah lakunya dapat dipastikan berdasarkan program-program komputer yang dijalankan. Sedangkan sistem robabilistik merupakan sistem yang kondisi masa depannya tidak dapat diprediksi karena mengandung unsur probabilitas, misalnya sistem manusia.

4. Sistem Terbuka dan Sistem Tertutup

Sistem terbuka merupakan sistem yang berhubungan dan terpengaruh dengan lingkungan luarnya. Lebih sepesifik dikenal juga yang disebut dengan sistem

terotomasi, yang merupakan bagian dari sistem buatan manusia dan berinteraksi dengan kontrol oleh satu atau lebih komputer sebagai bagian dari sistem yang digunakan dalam masyarakat modern. Sistem ini menerima masukan dan menghasilkan keluaran untuk subsistem lainnya, misalnya sistem kebudayaan manusia. Sedangkan sistem tertutup merupakan sistem yang tidak berhubungan dan tidak terpengaruh dengan lingkungan luarnya. Sistem ini bekerja secara otomatis tanpa adanya campur tangan dari pihak luar. Secara teoritis sistem tersebut ada, tetapi kenyataannya tidak ada sistem yang benar-benar tertutup, yang ada hanyalah *relatively closed system* (secara relatif tertutup, tidak benar-benar tertutup).

D. Informasi

Menurut (Hutahaean, 2015) mengemukakan bahwa “Informasi adalah data yang diolah menjadi bentuk yang lebih berguna dan lebih berarti bagi penerimanya”.

Informasi yang berkualitas memiliki kriteria-kriteria tertentu. Menurut (Romney & Steinbart, 2015) mengemukakan bahwa kriteria informasi yang terdiri dari:

1. Relevan

Informasi harus bisa dikatakan relevan apabila informasi yang termuat di dalamnya dapat mempengaruhi keputusan pengguna dengan membantu mereka mengevaluasi peristiwa masa lalu atau masa kini, dan memprediksi masa depan, serta menegaskan atau mengoreksi hasil evaluasi mereka di masa lalu.

2. Andal

Informasi harus bebas dari pengertian yang menyesatkan dan kesalahan material, menyajikan setiap fakta secara jujur, serta dapat diverifikasi.

3. Lengkap

Informasi disajikan selengkap mungkin, yaitu mencakup semua informasi yang dapat mempengaruhi pengambilan keputusan.

4. Tepat waktu

Informasi disajikan tepat waktu sehingga dapat berpengaruh dan berguna dalam pengambilan keputusan.

5. Dapat dipahami

Informasi yang disajikan dalam bentuk serta istilah yang disesuaikan dengan batas pemahaman para pengguna.

6. Dapat diverifikasi

Informasi yang disajikan dapat diuji, dan apabila pengujian dilakukan lebih dari sekali oleh pihak yang berbeda, hasilnya tetap menunjukkan simpulan yang tidak berbeda jauh.

7. Dapat diakses

Informasi yang tersedia pada saat dibutuhkan dan dengan format yang dapat digunakan.

2.2. Teori Pendukung

A. UML (*Unified Modeling Language*)

Pada perkembangan teknik pemrograman berorientasi objek, munculah sebuah standarisasi bahasa pemodelan untuk pembangunan perangkat lunak yang dibangun dengan menggunakan teknik pemrograman berorientasi objek, yaitu *Unified Modelling Language (UML)*. UML muncul karena adanya kebutuhan pemodelan visual untuk menspesifikasikan, menggambarkan, membangun, dan dokumentasi dari sistem perangkat lunak. UML Merupakan bahasa visual untuk pemodelan dan komunikasi mengenai sebuah sistem dengan menggunakan diagram dan teks-teks pendukung (Rosa & Shalahuddin, 2018).

Seperti bahasa-bahasa lainnya, UML mendefinisikan notasi dan *syntax/semantik*. Notasi UML merupakan sekumpulan bentuk khusus untuk menggambarkan berbagai diagram piranti lunak. Setiap bentuk memiliki makna tertentu, dan UML *syntax* mendefinisikan bagaimana bentuk-bentuk tersebut dapat dikombinasikan. Notasi UML terutama diturunkan dari 3 notasi yang telah ada sebelumnya : *Grady Booch OOD (Object Oriented Design)*, *Jim Rumbaugh OMT (Object Modeling Technique)*, dan *Ivar Jacobson OOSE (Object Oriented Software Engineering)*.

Abstraksi konsep dasar UML terdiri dari *structural classification*, *dynamic behavior* dan *model management*. Abstraksi konsep dasar UML terdiri dari *structural*, *class*, *ification*, *dynamic*, *behavior* dan *model manajemen*. UML mendefinisikan diagram-diagram sebagai berikut:

1. Use Case Diagram

“*Use case* adalah pemodelan untuk kelakuan sistem informasi yang akan dibuat. Kesimpulannya *use case* digunakan untuk mengetahui fungsi apa saja yang ada di dalam sebuah sistem informasi dan siapa saja yang berhak menggunakan fungsi itu” (Rosa & Shalahuddin, 2018).

Stereotype adalah sebuah model khusus yang terbatas untuk kondisi tertentu. Untuk menunjukkan *stereotype* digunakan symbol “<<” diawalnya dan ditutup “>>” diakhirnya. <<*extend*>> digunakan untuk menunjukkan bahwa satu *use case* merupakan tambahan fungsional dari *use case* yang lain jika kondisi atau syarat tertentu yang dipenuhi. Sedangkan <<*include*>> digunakan untuk menggambarkan bahwa suatu *use case* seluruhnya merupakan fungsionalitas dari *use case* lainnya. Biasanya <<*include*>> digunakan menghindari pengcopian suatu *use case* karena sering dipakai. Sebuah skema *level use case*, *use case* utama ada pada

level „, sea level’. *Sea level use case* mewakili interaksi *diskret* diantara *actor* utama dan *system*. *Use case* ini akan mendeliver beberapa nilai ke *actor* utama dan biasanya membutuhkan beberapa menit sampai setengah jam bagi *actor* untuk melakukan. *Use case* yang ada karena di *include* oleh *use case sea level* disebut *fish level*.

2. **Activity Diagram**

Activity Diagram menggambarkan *workflow* (aliran kerja) atau aktivitas dari sebuah sistem atau proses bisnis atau menu yang ada pada perangkat lunak. Aktivitas menggambarkan proses yang berjalan, sementara *Use Case* menggambarkan bagaimana *actor* menggunakan sistem untuk melakukan aktivitas. Sama seperti *state*, *standart* UML menggunakan segi empat dengan sudut membulat untuk menggambarkan aktivitas (Rosa & Shalahuddin, 2018).

Ilustrasikan proses paralel (*fork and join*) digunakan titik sinkronisasi yang dapat berupa titik, garis *horizontal* atau *vertikal*. *Activity Diagram* dapat dibagi menjadi beberapa *object swimlane* untuk menggambarkan objek mana yang bertanggung jawab untuk aktivitas tertentu.

3. **Class Diagram**

Menurut Whitten dalam (Suendri, 2018) “Kelas sebagai suatu set objek yang memiliki atribut dan perilaku yang sama, kelas kadang disebut kelas objek”. *Class* memiliki tiga area pokok yaitu :

- a. Nama, kelas harus mempunyai sebuah nama.
- b. Atribut, adalah kelengkapan yang melekat pada kelas. Nilai dari suatu kelas hanya bisa diproses sebatas atribut yang dimiliki.
- c. Operasi, adalah proses yang dapat dilakukan oleh sebuah kelas, baik pada kelas itu sendiri ataupun kepada kelas lainnya.

4. **Sequence Diagram**

Menurut Havaluddin dalam (Suendri, 2018) “Secara mudahnya *sequence diagram* adalah gambaran tahap demi tahap, termasuk kronologi (urutan) perubahan

secara logis yang seharusnya dilakukan untuk menghasilkan sesuatu sesuai dengan *use case diagram*”.

B. Basis Data

1. Entity Relationship Diagram

Entity Relationship Diagram merupakan suatu model untuk menjelaskan hubungan antar data dalam basis data berdasarkan suatu persepsi bahwa *real world* terdiri dari *object-object* dasar yang mempunyai hubungan atau relasi antar *object-object* tersebut. Relasi antar *object* dilukiskan dengan menggunakan symbol-symbol grafis tertentu (Bakhri, 2015).

Komponen yang terdapat dalam *Entity Relationship Diagram* adalah sebagai berikut:

1. Entity (entitas)

Entitas adalah suatu data yang dapat disimpan dan berguna bagi badan atau perusahaan, dengan kata lain suatu obyek yang dapat dibedakan dengan objek lainnya. Entitas digambarkan dengan kotak persegi panjang.

Terdapat juga Entitas Lemah (*Weak Entity*), yaitu suatu entitas sangat bergantung dengan entitas biasa, dengan kata lain, entitas lemah tidak akan ada apabila tidak ada entitas biasa. Entitas lemah digambarkan dengan kotak persegi panjang dengan garis ganda.

2. Atribut

Atribut menunjukkan karakteristik dari tiap-tiap entitas. Atribut digambarkan dengan bentuk *oval*.

3. Relasi

Relasi menunjukkan hubungan yang terjadi diantara entitas. Relasi digambarkan dengan bentuk belah ketupat atau *diamond*.

4. Line Connector

Line Connector digambarkan dengan bentuk garis tunggal.

5. Atribut Utama

Atribut utama digambarkan dengan bentuk *oval*, dengan keterangan diberi garis bawah *absolute*.

6. Atribut Pilihan

Atribut Pilihan digambarkan dengan bentuk *oval* dengan keterangan diberi garis bawah putus-putus.

7. Kardinalitas

Kardinalitas merupakan tingkat hubungan yang terjadi diantara entitas di dalam sebuah sistem. Terdapat tiga tingkat hubungan yang terjadi, yaitu:

a. Hubungan Satu pada Satu (*One to One* atau 1:1)

Tingkat hubungan dinyatakan satu pada satu jika satu kejadian pada entitas pertama hanya mempunyai satu hubungan dengan suatu kejadian pada entitas kedua. Demikian juga sebaliknya satu kejadian pada entitas kedua hanya bisa mempunyai satu hubungan dengan satu kejadian pada entitas yang pertama.

b. Hubungan Satu pada Banyak (*One to Many* atau 1:M)

Tingkat hubungan satu pada banyak (1:M) adalah sama dengan banyak pada satu (M:1), tergantung dari arah mana hubungan tersebut dilihat. Untuk satu kejadian pada entitas yang pertama dapat mempunyai banyak hubungan dengan kejadian pada entitas yang kedua. Sebaliknya satu kejadian pada entitas yang kedua, hanya bisa mempunyai satu hubungan dengan satu kejadian pada entitas yang pertama.

c. Hubungan Banyak pada Banyak (*Many to Many* atau M:N)

Tingkat hubungan banyak pada banyak (M:N) terjadi jika tiap kejadian pada sebuah entitas akan mempunyai banyak hubungan dengan kejadian pada entitas lainnya, baik dilihat dari sisi entitas yang pertama maupun dilihat dari sisi entitas yang kedua.

2. LRS (*Logical Relationship Structure*)

LRS merupakan transformasi dari penggambaran ERD dalam bentuk yang lebih jelas dan mudah untuk dipahami Pratama (Rosidin & Lubis, 2017). Penggambaran LRS hampir mirip dengan penggambaran normalisasi *file*, hanya saja tidak digambarkan simbol *asterix* (*) sebagai simbol *primary key* (kunci utama) dan *foreign key* (kunci tamu).

LRS merupakan hasil pemodelan *Entity Relationship* (ER) beserta atributnya sehingga bisa terlihat hubungan-hubungan antar entitas.

Dalam pembuatan LRS terdapat tiga hal yang dapat mempengaruhi, yaitu:

1. Jika tingkat hubungan (*cardinality*) satu pada satu (*one to one*), maka digabungkan dengan entitas yang lebih kuat (*strong entity*), atau digabungkan dengan entitas yang memiliki atribut yang lebih sedikit.
2. Jika tingkat hubungan (*cardinality*) satu pada banyak (*one to many*), maka hubungan relasi atau digabungkan dengan entitas yang tingkat hubungannya banyak.
3. Jika tingkat hubungan (*cardinality*) banyak pada banyak (*many to many*), maka hubungan relasi tidak akan digabungkan dengan entitas manapun, melainkan menjadi sebuah LRS.

3. *Constraint Basis Data*

Constraint basis data merupakan struktur yang dibuat oleh pengguna atau perancang basis data yang mencerminkan tingkah laku dari suatu tabel dan kolom. *Constraint* dirancang pada saat mendefinisikan basis data dengan tujuan utama memproteksi validitas data. *Constraint* pertama dan kedua yang digunakan untuk menjaga integritas entitas dan referensial sudah umum digunakan dalam perancangan basis data yakni dengan penggunaan kunci primer (*primary key*) dan kunci tamu (*foreign key*). Sedangkan tipe ke tiga yakni *constraint* domain masih hanya digunakan sampai dengan pemilihan tipe data yang tepat seperti contoh tabel di atas. Hampir semua penelitian yang membahas pembangunan sistem informasi dengan basis data relasional,

perancangan basis datanya masih hanya menggunakan kunci primer, kunci tamu dan pemilihan tipe data. (Andria & Lenawati, 2015)

C. Model Pengembangan Perangkat Lunak

1. *Agile Software Development*

Menurut Ferdiana dalam (Lubis, 2016) “Proses ini ditandai dengan model yang dikenal dengan iteratif dan inkremental. Proses yang meyakini bahwa suatu perangkat lunak dapat dikembangkan dengan desain minimalis, pengujian bertahap, dan dokumentasi yang tidak berlebihan.”

Agile Software Development adalah sebuah metode dalam pengembangan atau pembuatan sistem dengan cara cepat, artinya kebutuhan sistem sudah terpenuhi dengan melibatkan client dalam pembuatan *software*nya, sehingga kemungkinan sistem gagal itu sangat sedikit.

2. *Global Extreme Programming*

Menurut Ferdiana dalam (Lubis, 2016) “*Extreme Programming (XP)* dikenal dengan metode atau “*technical how to*” bagaimana suatu tim teknis mengembangkan perangkat lunak secara efisien melalui berbagai prinsip dan teknik praktis pengembangan perangkat lunak. XP menjadi dasar bagaimana tim bekerja sehari-hari”. *Global Extreme Programming* yaitu pengembangan dari XP (*Extreme Programming*) *Life Cycle*, dalam *Global Extreme Programming* terdapat lima *phase* yaitu *Exploration*, *Planning*, *Iteration*, *Production* dan *Maintenance*.

1) *Exploration phase* (Fase eksplorasi).

Fase eksplorasi memfokuskan pada pengambilan kebutuhan klien. Pada tahap ini visi produk dan tujuan bisnis aplikasi dirumuskan dan diatur kembali. Hasil dari tahap ini adalah sekumpulan kebutuhan pengguna beserta prototipe yang ditunjukkan untuk tahap berikutnya.

2) *Planning phase* (Fase Perencanaan).

Fase perencanaan memfokuskan pada pemilihan kebutuhan yang sesuai dengan batasan-batasan yang dimiliki klien dan tim. Kesepakatan fase ini menghasilkan rencana rilis dan rencana iterasi. Rencana rilis adalah melampirkan fitur-fitur yang akan dikembangkan dalam jangka waktu yang telah disepakati. Rencana iterasi menghasilkan sekumpulan tahapan-tahapan yang akan dilakukan beserta keluaran yang diperoleh untuk setiap tahapan.

3) *Iteration Phase* (Fase Iterasi).

Fase ini dikenal juga dengan fase pengembangan solusi. Fase iterasi adalah melakukan eksekusi perencanaan iterasi melalui serangkaian aktivitas teknis seperti pembuatan arsitektur, pembuatan kode, dan melakukan unit test untuk setiap modul. Hasil keluaran setiap iterasi dikenal dengan rilis parsial/small release.

4) *Production Phase* (Fase Produksi).

Fase ini melakukan pengujian terhadap hasil setiap iterasi. Tim akan melakukan pengujian dengan melibatkan klien. Pada tahapan ini dilakukan juga verifikasi dan integrasi terhadap hasil dengan rencana rilis yang ada. Hasil fase ini adalah sebuah solusi yang sudah dites baik oleh tim dan klien.

5) *Maintenance Phase* (Fase Pemeliharaan).

Fase ini memfokuskan pada layanan didukung setelah software tersebut dikembangkan. Fase ini melakukan perbaikan kesalahan pada solusi dan penyesuaian kecil pada solusi. Pada fase ini dimungkinkan pula terjadi kesepakatan untuk pengembangan sistem pada rilis selanjutnya.

D. Website

Menurut Agus Hariyanto dalam (Destiningrum & Adrian, 2017), *Website* adalah : “*Web* dapat diartikan sebagai kumpulan halaman yang menampilkan informasi data teks, data gambar, data animasi, suara, video dan gabungan dari semuanya, baik yang bersifat statis maupun dinamis yang membentuk satu rangkaian bangunan yang saling terkait, dimana masing-masing dihubungkan dengan jaringan-jaringan halaman (*hyperlink*).

Secara termologi, pengertian *website* adalah kumpulan dari halaman-halaman situs atau link yang biasanya terangkum dalam sebuah domain atau subdomain yang tepatnya berada didalam *World Wide Web* (WWW) di *Internet*. Sebuah halaman web adalah dokumen yang ditulis dalam format HTML (*Hyfer Text Markup Language*), yang hampir selalu bisa di akses melalui HTTP, yaitu protokol yang menyampaikan informasi dari server website untuk ditampilkan kepada para pemakai melalui *web browser*. Halaman-halaman dari sebuah *website* yang diidentifikasi oleh URL (*Uniform Resource Locator*) biasa Disebut *homepage* atau *domain name*.

Untuk membangun situs diperlukan beberapa unsur yang harus ada agar situs dapat berjalan dengan baik dan sesuai dengan yang diharapkan. Unsur-unsur yang harus ada dalam situs antara lain :

a. Domain Name

Domain name atau biasa disebut nama *domain* adalah alamat permanen situs di dunia *internet* yang digunakan untuk mengidentifikasi sebuah situs atau dengan kata lain *domain name* adalah alamat yang digunakan untuk menemukan situs kita pada dunia *internet*.

b. Hosting

Hosting dapat diartikan sebagai ruangan yang terdapat dalam *harddisk* tempat menyimpan berbagai data, *file-file*, gambar dan lain sebagainya, yang akan di

tampilkan di situs. Besarnya *hosting* di tentukan ruangan *harddisk* dengan ukuran MB (*Mega Byte*) atau GB (*Giga Byte*).

c. *Script/Bahasa Program*

Adalah bahasa yang digunakan untuk menerjemahkan setiap perintah dalam situs pada saat di akses. Jenis *script* sangat menentukan statis, dinamis atau interaktifnya sebuah situs. Jenis-jenis *script* yang banyak di gunakan para *designer* antara lain HTML, ASP, PHP, *Java Script*, *Java Aplets* dan sebagainya.

d. *Design Web*

Setelah melakukan penyewaan *domain* dan *hosting* serta penguasaan *scripts*, unsure situs yang paling penting dan utama adalah *design*. *Design web* sangat menentukan kualitas dan keindahan situs.

e. Publikasi

Keberadaan situs tidak ada gunanya di bangun tanpa di kunjungi atau dikenal masyarakat. Untuk mengenalkan situs kepada masyarakat memerlukan apa yang disebut publikasi atau promosi. Cara yang biasanya dilakukan dan paling efektif adalah dengan publikasi langsung di internet melalui *search-engine* (mesin pencari, seperti : *Yahoo*, *Google*, *Search Indonesia*).

E. Internet

Menurut Krisianto dalam (Puspitasari, 2015) *Internet* adalah salah satu bentuk media komunikasi dan informasi interaktif. Wujud *internet* adalah jaringan komputer yang terhubung di seluruh dunia. *Internet* digunakan untuk mengirim informasi antar komputer di seluruh dunia. *Internet* diartikan sebagai “*The global public Transmission Control Protocol/Internet Protocol (TCP/IP) internetwork*”. Jadi *Internet* adalah gabungan dari seluruh komputer di dunia yang di satukan oleh sebuah “bahasa” yang sama, adapun bahasa yang dimaksud adalah *Transmission Control Protokol/Internet Protokol (TCP/IP)*.”.

Banyak keuntungan yang didapat dari jaringan komputer, diantaranya produktivitas dan efisien. Jaringan komputer menurut area atau lokasi dapat dibagi menjadi empat yaitu:

- a. *Local Area Network (LAN)*, yaitu jaringan komputer dimana komputer-komputer yang terhubung masih dalam satu area atau lokasi.
- b. *Wide Area Network (WAN)*, yaitu koneksi antara LAN-LAN yang berbeda lokasi/area. Ciri utamanya adalah memiliki *bandwidth* yang terbatas karena disesuaikan dengan fungsi harga, adanya *problem delay antarstasiun*.
- c. *Metropolitan Area Network (MAN)*, yaitu sama seperti LAN hanya saja lebih luas areanya semisal dalam satu kota/daerah dengan range mencapai 50 km.
- d. *Internet*, yaitu kepanjangan dari *interconnection networking* atau juga yang telah menjadi *international networking* merupakan suatu jaringan yang menghubungkan komputer di seluruh dunia tanpa dibatasi oleh jumlah unit menjadi satu jaringan yang bisa saling mengakses.

F. Web Browser

Menurut Kasiman dalam (Susanti, 2016) *Web browser* adalah program untuk menampilkan halaman yang berbentuk kode HTML. Semua halaman *web* ditulis dengan bahasa HTML (*Hypertext Mark Up Language*). Walaupun beberapa file mempunyai ekstensi yang berbeda (contoh: .html, .php, .php3), *output file-file* tersebut tetap HTML. HTML adalah medium yang selalu dikirimkan ke *web browser* baik halaman itu berupa halaman statis, sebuah *script* (seperti PHP), ataupun yang dibuat oleh program CGI (*Common Gateway Interface*).