BAB 2

LANDASAN TEORI

2.1 Definisi Desa

Desa, berdasarkan Undang-undang Nomor 32 Tahun 2004 tentang Pemerintahan Daerah, adalah kesatuan masyarakat hukum yang memiliki batas-batas wilayah yurisdiksi, berwenang untuk mengatur dan mengurus kepentingan masyarakat setempat berdasarkan asal-usul dan adat istiadat setempat yang diakui dan dibentuk dalam sistem Pemerintahan Nasional dan berada di kabupaten/kota, sebagaimana dimaksud dalam Undang-Undang Dasar Negara Republik Indonesia Tahun 1945. Landasan pemikiran dalam pengaturan mengenai desa adalah keanekaragaman, partisipasi, otonomi asli, demokratisasi, dan pemberdayaan masyarakat.[9]

Undang-udang Nomor 32 Tahun 2004 tentang Pemerintahan Daerah mengakui otonomi yang dimiliki oleh desa ataupun dengan sebutan lainnya dan kepada desa melalui pemerintah desa dapat diberikan penugasan ataupun pendelegasian dari Pemerintah ataupun pemerintah daerah untuk melaksanakan urusan pemerintahan tertentu. Sedang terhadap desa di luar desa geneologis yaitu desa yang bersifat administratif seperti desa yang dibentuk karena transmigrasi ataupun karena alasan lain yang warganya pluralistis, majemuk, ataupun heterogen, maka otonomi desa akan diberikan kesempatan untuk tumbuh dan berkembang mengikuti perkembangan dari desa itu sendiri. sebagai perwujudan demokrasi, dalam penyelenggaraan pemerintahan Desa

dibentuk Badan Permusyawaratan Desa atau sebutan lain yang sesuai dengan budaya yang berkembang di Desa bersangkutan, yang berfungsi sebagai lembaga pengaturan dalam penyelenggaraan pemerintah Desa, seperti dalam pembuatan dan pelaksanaan Peraturan Desa, Anggaran Pendapatan dan Belanja Desa, Keputusan Kepala Desa. Di desa dibentuk lembaga kemasyarakatan yang berkedudukan sebagai mitra kerja pemerintah desa dalam memberdayakan masyarakat desa.

2.1.1 Visi dan Misi desa

Visi desa adalah **"TERWUJUDNYA MASYARAKAT DESA YANG SEJAHTERA DAN DINAMIS DALAM NUANSA RELIGIUS DAN BERWAWASAN LINGKUNGAN SEBAGAI DESA PENDIDIKAN DAN WISATA"**. Adapun makna dari Visi tersebut adalah:

1. **Terwujudnya Masyarakat Desa yang Sejahtera dan Dinamis**, mengandung arti kondisi desa yang masyarakatnya memiliki keberdayaan secara sosial dan ekonomi sehingga mampu melangsungkan kehidupan individu maupun kemasyarakatan secara layak serta senantiasa penuh tenaga dan semangat untuk cepat bergerak dan mudah menyesuaikan diri dengan keadaan untuk terus menuju perubahan ke arah yang lebih baik.
2. **Dalam Nuansa Religius dan Berwawasan Lingkungan**, mengandung arti bahwa segala aktivitas kehidupan di desa senantiasa dijiwai oleh nilai-nilai budaya, norma, agama, dan kepedulian terhadap lingkungan.

3. **Sebagai tempat Pendidikan dan Wisata**, desa berdasarkan potensi kewilayahan merupakan desa tempat diselenggarakannya berbagai kegiatan pendidikan dan kegiatan pariwisata yang berkualitas.

2.1.2 Dasar Hukum

Pelaksanaan program pemerintah baik pusat maupun daerah senantiasa dikoordinasikan dengan Pemerintah Desa. Karena salah satu fungsi pemerintah desa adalah pelayanan dan perlindungan masyarakat. Dasar hokum tugas pembantuan ;

1. Undang-Undang Nomor 14 Tahun 1950 tentang Pemerintahan Daerah Kabupaten dalam Lingkungan Jawa Barat (Berita Negara Tahun 1950) sebagaimana telah diubah dengan Undang-Undang Nomor 4 Tahun 1968 tentang Pembentukan Kabupaten Puwakarta dan Kabupaten Subang dengan mengubah Undang-Undang Nomor 14 Tahun 1950 tentang pembentukan daerah-daerah kabupaten dalam lingkungan Provinsi Jawa Barat (Lembaran Negara Republik Indonesia Tahun 1968 Nomor 31, Tambahan Lembaran Negara Republik Indonesia Nomor 2851);
2. Undang-Undang Nomor 10 Tahun 2004 tentang pembentukan peraturan Perundang-undangan (Lembaran Negara Tahun 2004) Nomor 53, Tambahan Lembaran Negara Nomor 4389);
3. Undang-Undang Nomor 32 Tahun 2004 tentang Pemerintahan Daerah (Lembaran Negara Tahun 2004 Nomor 125, Tambahan Lembaran Negara Nomor 4437), sebagaimana telah diubah dengan Undang-

Undang Nomor 12 Tahun 2008 tentang perubahan kedua Atas Undang-Undang Nomor 32 Tahun 2004 tentang Pemerintahan Daerah (Lembaran Negara Republik Indonesia Tahun 2008 Nomor 59, Tambahan Lembaran Negara Republik Indonesia Nomor 4844);

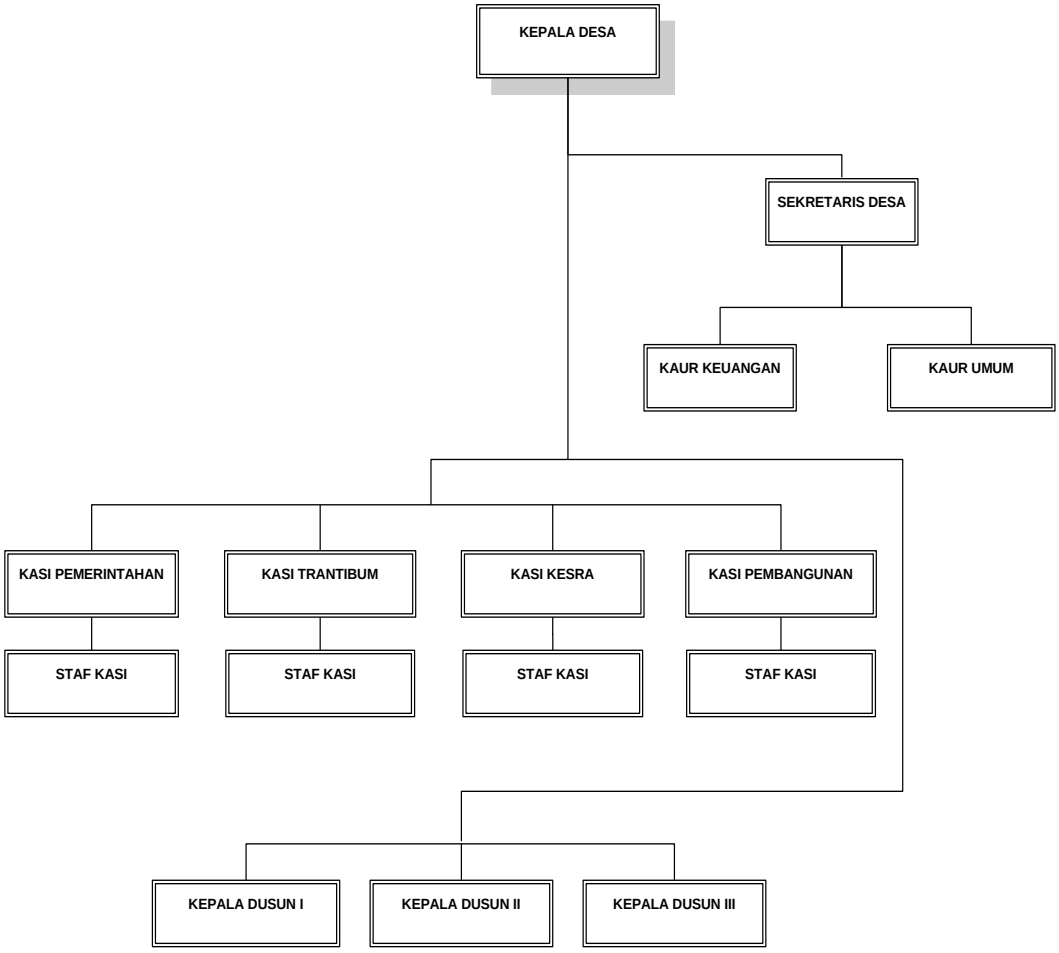
4. Keputusan Bupati Bandung Nomor 8 Tahun 2004 tentang pelimpahan Sebagai Kewenangan Bupati Kepada Camat di Lingkungan Pemerintahan Kabupaten Bandung (Lembaran Daerah Kabupaten Bandung Tahun 2004 Nomor 9 seri D);

2.1.3 Struktur Organisasi dan *Job Description*

Dalam pelaksanaan pemerintahan desa terdapat struktur organisasi, yang berfungsi sebagai parameter jabatan dari masing-masing perangkat desa. Sedangkan *Job Description*, berfungsi menjelaskan tugas-tugas yang harus dikerjakan tiap perangkat desa sesuai dengan jabatan yang tertera dalam Struktur Organisasi.

2.1.3.1 Struktur Organisasi Desa

Struktur Organisasi Desa dapat dilihat pada Gambar 2.1



Gambar 2.1 Struktur Organisasi Desa

2.1.3.2 *Job Description* Struktur Organisasi desa

a. Kepala Desa

Bertugas dan berkewenangan sebagai pemimpin dalam penyelenggaraan pemerintahan desa, yaitu membina kehidupan masyarakat desa, membina perekonomian desa, memelihara ketentraman dan ketertiban masyarakat desa, mendamaikan perselisihan masyarakat di desa dan mengajukan rancangan peraturan desa dan menetapkannya sebagai peraturan desa bersama dengan BPD.

b. Sekretaris Desa

Bertugas dan berkewenangan memberikan saran dan pendapat kepada kepala desa, memimpin, mengkoordinasikan dan mengendalikan serta mengawasi semua kegiatan yang dilaksanakan unsur teknis lapangan, dan unsur pembantu kepala desa, memberikan informasi mengenai keadaan sekretariat desa dan keadaan desa, merumuskan program kegiatan kepala desa, membantu kepala desa dalam penyusunan atau perumusan rancangan peraturan desa, membantu kepala desa dalam menyusun laporan penyelenggaraan pemerintahan desa kepada bupati melalui camat, membantu kepala desa dalam menyusun laporan keterangan pertanggungjawaban kepada BPD, mengadakan dan melaksanakan persipan rapat dan mencatat hasil-hasil rapat, menyusun rancangan anggaran pendapatan dan belanja desa, menyusun keuangan desa, mengadakan kegiatan inventarisasi kekayaan desa, melaksanakan administrasi kepegawaian aparat desa, melakukan kegiatan pencatatan mengenai penghasilan kepala desa dan perangkat desa sesuai dengan peraturan perundang-undangan yang berlaku, mengumpulkan dan menganalisa data sumber penghasilan desa baru untuk

dikembangkan, melakukan kegiatan administrasi pendapatan yang dikelola oleh desa, melakukan kegiatan administrasi desa, melakukan, menerima dan mengendalikan surat-surat masuk dan keluar serta melaksanakan kearsipan; melaksanakan pengetikan surat-surat hasil persidangan rapat-rapat atau naskah dinas lainnya; melaksanakan penyediaan, penyimpanan dan pendistribusian alat-alat tulis kantor serta pemeliharaan dan perbaikan peralatan kantor; menyusun jadwal serta mengikuti perkembangan pelaksanaan piket; melaksanakan dan mengusahakan ketertiban dan kebersihan kantor dan bangunan milik desa lainnya; menyelenggarakan pengelolaan Buku Administrasi umum; melaksanakan persiapan penyelenggaraan rapat dn penerimaan tamu dinas dan kegiatan kerumahtanggaan pada umumnya; dan melaksanakan tugas lainnya yang diberikan oleh kepala desa.

c. KAUR Keuangan

Melakukan pengelolaan administrasi keuangan desa yang meliputi penyusunan anggaran, pembukuan, pertanggungjawaban keuangan desa dan laporan realisasi keuangan serta membantu pemungutan dan penyetoran PBB kepada kas negara, memungut dan menyetorkan pajak lainnya; dan melaksankan tugas lain yang diberikan oleh kepala desa.

d. KAUR Umum

Melakukan pengumpulan administrasi kepegawaian, penyelenggaraanbrapat- rapat, tata usaha desa, surat menyurat, kearsipan, penyajian data dan kepustakaan serta dokumentasi; dan melaksankan tugas lain yang diberikan oleh kepala desa.

e. KASI Pemerintahan

Melaksanakan administrasi pemerintahan desa, melaksanakan administrasi penduduk di desa, mengadakan kegiatan pencatatan mutasi tanah dan pencatatan administrasi pertanahan, melaksanakan dan memberikan pelayanan terhadap masyarakat dalam hal pembuatan Kartu Tanda Penduduk (KTP), melaksanakan kegiatan monografi/profil desa, melaksanakan penyelenggaraan buku administrasi desa dan keputusan kepala desa, dan melakukan tugas lain yang diberikan oleh kepala desa.

f. KASI Trantibum

Menjaga ketenteraman, kemanan dan ketertiban masyarakat secara umum; mengkoordinasikan dan melaksanakn upaya-upaya dalam rangka menjaga keamanan dan ketertiban lingkungan; melaksanakan kegiatan kemasyarakatan termasuk kegiatan ketenteraman dan ketertiban serta perlindungan masyarakat (LINMAS); menginventarisasi kegiatan dan personil keamanan lingkungan; dan melaksankan tugas lain yang diberikan oleh kepala desa.

g. KASI Kesra

Mengumpulkan, mengolah, mengevaluasi dan pelaporan data dibidang kesejahteraan masyarakat, sosial, serta mengadakan pembinaan keagamaan, kesehatan, keluarga berencana dan pendidikan masyarakat; dan melaksankan tugas lain yang diberikan oleh kepala desa.

g. KASI Pembangunan

Mengkoordinasi pelaksanaan tugas dalam unit kerja dalam bidang pembangunan, antar unit kerja dengan lembaga kemasyarakatan yang terkait baik secara Formal ataupun informal guna memperoleh kesatuan pendapat; dan melaksankan tugas lain yang diberikan oleh kepala desa.

h. Kepala Dusun

Membantu pelaksanaan tugas Kepala Desa; dan melaksankan tugas lain yang diberikan oleh kepala desa.

2.2 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 mendefinisikan sistem sebagai berikut :

"sistem adalah suatu jaringan kerja dari prosedur-prosedur yang saling berhubungan, berkumpul bersama untuk melakukan suatu kegiatan atau bentuk penyelesaian suatu sasaran tertentu."

Pendekatan system merupakan jaringan kerja dari prosedur yang lebih menekankan urutan operasi di dalam sistem. Prosedur disini menurut Richard F. Neuschel adalah

"suatu urutan operasi *klerikal* (tulis-menulis), yang biasanya melibatkan beberapa orang dalam satu atau lebih departemen, yang diterapkan untuk menjamin penanganan yang seragam dari transaksi-transaksi bisnis yang terjadi."

Lebih lanjut dalam bukunya, Jerry FitzGerrald, Ardra F.FitzGerald dan Warren D. Stallings, Jr, mendefinisikan prosedur sebagai berikut :

"Prosedur adalah urutan yang tepat dari tahapan-tahapan instruksi yang menerangkan apa (*what*) yang harus dikerjakan, siapa (*who*) yang mengerjakannya, kapan (*when*) dikerjakan dan bagaimana (*who*) mengerjakannya."

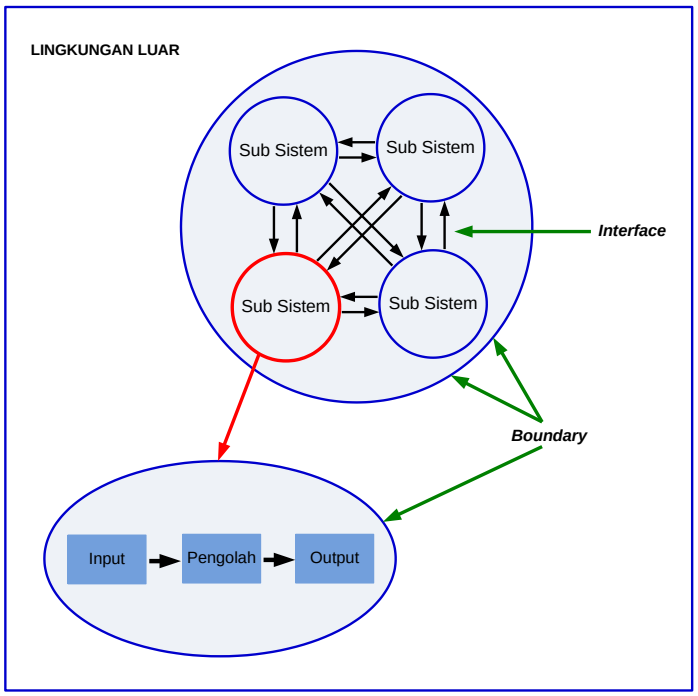
Pendekatan sistem yang lebih menekankan pada elemen atau komponennya mendefiniskan sistem sebagai berikut :

"sistem adalah kumpulan dari elemen-elemen yang berinteraksi untuk mencapai suatu tujuan tertentu."

Kedua kelompok definisi ini adalah benar dan tidak bertentangan, yang berbeda adalah cara pendekatannya. Pendekatan sistem yang merupakan kumpulan dari elemen-elemen atau komponen-komponen atau subsistem-subsistem merupakan definisi yang luas. Definisi inilah yang banyak diterima, karena kenyataannya suatu sistem dapat terdiri dari beberapa subsistem atau bagian-bagian sistem. Suatu sistem mempunyai maksud tertentu. Ada yang menyebutkan maksud dari suatu sistem adalah untuk mencapai suatu tujuan (*goal*) dan ada yang menyebutkan untuk mencapai suatu sasaran (*objectives*). Perbedaan tujuan dan sasaran terletak pada ruang lingkup yang dilibatkan, jika yang dilibatkan adalah sistem bisnis maka istilah "tujuan" lebih cocok untuk dijadikan maksud dari sistem itu sendiri, tetapi jika sistem akuntansi sebagai contoh lingkup yang dilibatkannya, maka sasaran lebih cocok untuk dijadikan maksudnya.

2.2.1 Karakteristik Sistem

Suatu sistem mempunyai karakteristik atau sifat-sifat yang tertentu, yaitu mempunyai komponen-komponen (*components*), batas sistem (*boundary*), lingkungan luar system (*environment*), penghubung (*interface*), masukan (*input*), keluaran (*output*), pengolah (*process*) dan tujuan (*goal*) atau sasaran (*objectives*). Karakteristik Sistem dapat dilihat pada Gambar 2.2



Gambar 2.2. Karakteristik Sistem

2.2.2 Komponen Sistem

Suatu sistem terdiri dari sejumlah komponen yang saling berinteraksi, yang artinya saling bekerja sama membentuk suatu kesatuan. Komponen-komponen sistem yang atau elemen-elemen sistem dapat berupa suatu subsistem atau bagian-bagian dari sistem. Setiap sistem tidak peduli berapapun kecilnya,

Selalu mempunyai sifat-sifat dari sistem untuk menjalanka suatu fungsi tertentu dan mempengaruhi proses sistem secara keseluruhan. Suatu sistem dapat mempunyai suatu sistem yang lebih besar yang disebut *supra system.*

2.2.3 Batas Sistem

Batas sistem (*boundary*) merupakan daerah yang membatasi antara suatu sistem dengan sistem yang lainnya atau dengan lingkungan luarnya. Batas sistem ini memungkinkan sistem dipandang sebagai satu kesatuan. Batas uatu sistem menunjukan ruang lingkup (*scope*) dari sistem tersebut.

2.2.4 Lingkungan Luar Sistem

Lingkungan luar (*environment*) dari suatu sistem adalah apapun diluar batas dari sistem yang mempengaruhi operasi sistem. Lingkungan luar sistem dapat bersifat menguntungkan dan dapat juga merugikan sistem tersebut. Lingkungan luar yang menguntungkan merupakan energi dari sistem dan dengan demikian harus tetap dijaga dan dipelihara. Sedang lingkungan luar yang merugikan harus ditahan dan dikendailakn, jika tidak maka akan mengganggu kelangsungan hidup dari sistem.

2.2.5 Penghubung Sistem

Penghubung (*interface*) merupakan media penghubung antara satu subsistem dengan subsistem yang lainnya. Melalui penguhubung ini memungkinkan sumber-sumber daya mengalir dari satu subsistem ke subsistem yang lainnya. Keluaran (*output*) dari satu subsistem akan menjadi masukan (*input*) untuk subsistem yang lainnya melalui penghubung. Dengan penghubung satu

subsistem dapat berintegrasi dengan subsistem yang lainnya membentuk satu kesatuan.

2.2.6 Masukan Sistem

Masukan (*input*) adalah energi yang dimasukkan ke dalam sistem. Masukan dapat berupa masukan perawatan (*maintenance input*) dan masukan sinyal (*signal input*). *Maintenance input* merupakan energi yang dimasukan supaya sistem tersebut beroperasi. *Signal input* adalah energi yang diproses untuk didapatkan keluaran (*output*). Sebagai contoh didalam sistem komputer, program adalah *maintenance input* yang digunakan untuk mengoperasikan komputernya dan data adalah *signal input* untuk diolah menjadi informasi.

2.2.7 Keluaran Sistem

Keluaran (*output*) adalah hasil energi yang diolah dan diklasifikasikan menjadi keluaran yang berguna dan sisa pembuangan. Keluaran dapat merupakan masukan untuk subsistem yang lain atau kepada supra sistem. Misalnya untuk sistem komputer, panas yang dihasilkan merupakan keluaran yang tidak berguna dan merupakan hasil sisa pembuangan, sedang informasi adalah keluaran yang dibutuhkan.

2.2.8 Pengolah Sistem

Suatu sistem dapat mempunyai suatu bagian pengolah yang akan merubah masukan menjadi keluaran. Suatu sistem produksi akan mengolah masukkan berupa bahan baku menjadi keluaran berupa barang jadi.

2.2.9 Sasaran dan Tujuan Sistem

Suatu sistem pasti mempunyai tujuan (*goal*) atau sasaran (*objectives*). Tujuan dari sistem sangat mempengaruhi dan menentukan sekali masukan yang dibutuhkan sistem dan keluaran yang akan dihasilkan sistem.

2.2.10 Klasifikasi Sistem

Sistem dapat diklasifikasikan dari beberapa sudut pandang, diantaranya adalah sebagai berikut :

1. Sistem diklasifikasikan sebagai **sistem abstrak (*abstract system*)** dan **sistem fisik (*physic system*).** Sistem abstrak adalah sistem yang berupa pemikiran atau ide-ide yang tidak tampak secara fisik. Misalnya sistem teologia, yaitu sistem yang berupa pemikiran-pemikiran hubungan antara manusia dengan Tuhan. Sistem fisik merupakan sistem yang ada secara fisik, misalnya sistem komputer, sistem akuntansi, sistem produksi dan lain sebagainya.
2. Sistem diklasifikasikan sebagai **sistem alamiah (*natural system*)** dan **sistem buatan manusia (*human made system*).** Sistem alamiah adalah sistem yang terjadi melalui proses alam, tidak dibuat manusia. Misalnya sistem perputaran bumi. Sistem buatan manusia adalah sistem yang dirancang oleh manusia. Sistem buatan manusia yang melibatkan interaksi manusia dengan mesin disebut dengan *human-machine system* atau ada yang menyebutnya dengan *man-machine system*. Sistem informasi merupakan contoh *man-machine system*, karena menyangkut penggunaan komputer yang berinterkasi dengan manusia.

3. Sistem diklasifikasikan sebagai **sistem tertentu (*deterministic system*)** dan **sistem tak tentu (*probabilistic system*).** Sistem tertentu beroperasi dengan tingkah laku yang sudah dapat diprediksi. Interaksi diantara bagian-bagiannya dapat dideteksi dengan pasti, sehingga keluaran dari sistem dapat diramalkan. Sistem komputer adalah contoh dari sistem tertentu yang tingkah lakunya dapat dipastikan berdasarkan program-program yang dijalankannya. Sistem tak tentu adalah sistem yang kondisi masa depannya tidak dapat diprediksi karena mengandung unsur probabilitas.
4. Sistem diklasifikasikan sebagai **sistem tertutup (*closed system*)** dan **sistem terbuka (*open system*).** Sistem tertutup merupakan sistem yang tidak berhubungan dan tidak terpengaruh dengan lingkungan luarnya. Sistem ini bekerja secara otomatis tanpa adanya turut campur tangan dari pihak diluarnya. Secara teoritis sistem tertutup ini ada, tetapi kenyataannya tidak ada sistem sistem yang benar-benar tertutup, yang ada hanyalah *relatively closed system* (secara relative tertutup, tidak benar-benar tertutup). Sistem terbuka adalah sistem yang berhubungan dan terpengaruh dengan lingkungan luar sistem. Sistem ini menerima masukan dan menghasilkan keluaran untuk lingkungan luar sistem, maka suatu sistem harus mempunyai pengendalian yang baik. Sistem yang baik harus dirancang sedemikian rupa, sehingga secara relatif tertutup karena sistem tertutup akan bekerja secara otomatis dan terbuka hanya untuk pengaruh yang baik saja.

2.3 Basis Data

Kata “basis data” digunakan untuk menguraikan segala sesuatu dari sekumpulan data tunggal, seperti daftar telepon. Istilah “basis data” tidak termasuk aplikasi, yang terdiri dari *form* dan *report* dimana pengguna akan saling berhubungan. Basis data terdiri dari *file-file* fisik yang ditetapkan berdasarkan komputer saat menerapkan perangkat lunak basis data. Disisi lain, suatu model basis data lebih kepada konsep dibandingkan objek fisik dan digunakan untuk menciptakan di dalam basis data.

Basis data terdiri dari 2 kata, yaitu “Basis” dan “Data”. “Basis” dapat diartikan sebagai markas atau gudang, tempat bersarang atau berkumpul. Sedangkan “Data” adalah representasi dari fakta dunia nyata yang mewakili suatu objek seperti manusia, barang, hewan, peristiwa, konsep, keadaan, dan sebagainya, yang direkam dalam bentuk angka, huruf, symbol, teks, gambar, bunyi, atau kombinasinya.

Basis Data pun dapat diartikan dalam sejumlah sudut pandang, seperti :

1. Basis Data adalah himpunan kelompok data (arsip) yang saling berhubungan yang diorganisasikan sedemikian rupa agar kelak dapat dimanfaatkan kembali dengan cepat dan mudah.
2. Basis Data adalah kumpulan data yang saling berhubungan yang disimpan secara bersama sedemikian rupa dan tanpa pengulangan (redudansi) yang tidak perlu, untuk memenuhi berbagai kebutuhan.
3. Basis Data adalah kumpulan *file*/tabel/arsip yang saling berhubungan yang disimpan dalam media penyimpanan elektronis.

Basis data sesungguhnya memiliki prinsip dan tujuan yang sama. Prinsip utamanya adalah pengaturan data/arsip. Dan tujuan utamanya adalah kemudahan dan kecepatan dalam pengambilan kembali data/arsip. Basis data juga bukan hanya sekedar disimpan dalam media penyimpanan elektronis. Artinya tidak semua bentuk penyimpanan data secara elektronis bisa disebut basis data.

Media penyimpanan elektronis juga dapat berisi *file* teks (yang diolah oleh program pengolah kata), *file spreadsheet*, dan lain-lain, tetapi tidak disebut sebagai basis data karena didalamnya tidak terdapat pemilahan dan pengelompokan data sesuai jenis atau fungsi data, sehingga akan menyulitkan pada saat pencarian data. Yang sangat diutamakan pada basis data adalah pengaturan/pemilahan/pengorganisasian/pengelompokan data yang akan kita simpan sesuai fungsi atau jenisnya. Pemilahan/pengorganisasian/pengelompokan ini dapat berupa sejumlah file atau tabel yang terpisah atau dalam bentuk pendefinisian kolom-kolom (*field*) data dalam setiap *file* atau tabel.

2.3.1 Sifat-sifat Basis Data

Istilah basis data pada umunya juga menyiratkan serangkaian sifat yang terkait seperti, berbagi data, integrasi data, integritas data, keamanan data, abstraksi data, dan independensi data.

1. Berbagi Data

   Data yang disimpan dalam basis data tidak secara umum dipegang semata-mata untuk digunakan oleh seseorang. Suatu basis data secara normal diharapkan bisa diakses oleh lebih dari satu orang, dan mungkin dapat diakses di waktu yang sama.

2. Integrasi Data

   Data bersama membawa banyak keuntungan bagi organisasi. Salah satu bentuk tanggung jawab pemakaian basis data yang utama adalah memastikan bahwa data terintegrasi. Hal tersebut menyiratkan bahwa suatu basis data harus menjadi koleksi data yang mana setidaknya secara ideal tidak menjadi rendudansi data. Suatu nilai dikatakan redundasi data jika suatu atribut memiliki dua atau lebih nilai yang sama.

3. Integritas Data

   Tanggung jawab lain yang timbul sebagai konsekuensi dari data bersama adalah bahwa basis data perlu menunjukan integritas. Dengan kata lain, basis data perlu secara akurat mencerminkan seluruh bidang yang mencoba pada model. Artinya, jika hubungan antar objek yang ada pada dunia nyata diwakili oleh data di dalam basis data, maka perubahan menjadi salah satu dari mitra pada suatu hubungan harus secara akurat menggambarkan perubahan yang terjadi di dalam hubungan tersebut.

4. Keamanan Data

   Salah satu cara yang utama untuk memastikan integritas basis data adalah dengan melakukan pembatasan akses, dengan kata lain pengamanan basis data. Cara yang utama tersebut telah dilakukan pada sistem basis data dengan penjelasan atas beberapa detail pada sekumpulan pengguna yang diberi hak, atau pada umumnya lebih pada bagian basis data.

5. Abstraksi Data

   Suatu basis data dipandang sebagai model nyata. Informasi yang disimpan di dalam basis data pada umumnya merupakan sebuah usaha untuk menyajikan sifat dari beberapa objek sesungguhnya. Oleh karena itu, sebagai contoh sebuah basis data akademik akan dimaksudkan untuk merekam rincian aktivitas universitas yang relevan. Dikatakan relevan karena tidak ada basis data yang bisa menyimpan semua sifat objek sesungguhnya. Oleh karena itu, suatu basis data adalah suatu abstraksi dari dunia nyata.

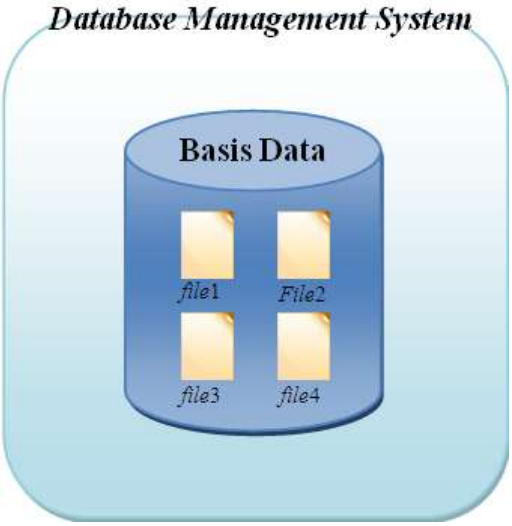
6. Independensi Data

   Salah satu konsekuensi dari abstraksi data adalah gagasan untuk *buffering data* dari proses yang menggunakan data. Sebagai contoh, suatu perubahan dibuat dalam beberapa bagian basis data dasar. Tidak ada program aplikasi yang menggunakan data yang perlu diubah. Dengan demikian jika suatu perubahan dibuat ke dalam beberapa bagia dari suatu sistem aplikasi, maka semestinya perubahan itu tidak mempengaruhi struktur data dasar yang digunakan oleh aplikasi.

2.4 Sistem Basis Data

Basis data hanyalah sebuah objek yang pasif/mati. Ia ada karena pembuatnya. Ia tidak akan pernah berguna jika tidak ada pengelola atau pembuatnya. Yang menjadi pengelola atau penggeraknya secara langsung adalah program atau aplikasi (*software*). Gabungan keduanya yaitu basis data dan

pengelolanya akan menghasilkan suatu sistem. Karena itu, secara umum sebuah sistem basis data merupakan sistem yang terdiri dari kumpulan tabel (*file*) yang saling berhubungan (dalam sebuah basis data pada sebuah sistem komputer) dan sekumpulan program (DBMS) yang memungkinkan beberapa pemakai dan/atau program lain untuk mengakses dan memanipulasi tabel (*file*) tersebut. Sistem Basis Data dapat dilihat pada Gambar 2.3



Gambar 2.3 Sistem Basis Data

2.4.1 Komponen Sistem Basis Data

Dalam sebuah sistem basis data terdapat beberapa komponen utama, yaitu :

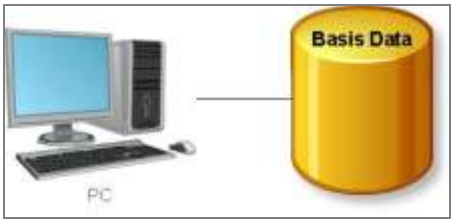
1. Perangkat Keras (*Hardware*)
2. Sistem Operasi (*Operating System*)
3. Basis Data (*Database*)
4. DBMS (*Database Management System*)

5. Pemakai (*User*)
6. Perangkat lunak lain yang sifatnya *optional*

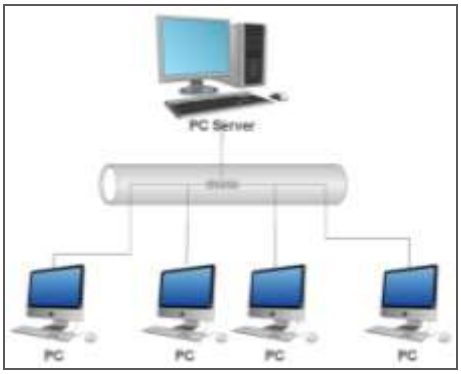
2.4.1.1 Perangkat Keras (*Hardware*)

Perangkat keras yang biasanya ada dalam sebuah basis data adalah :

1. Komputer (satu untuk sistem yang *stand-alone*, seperti pada Gambar 2.4 atau lebih dari satu untuk sistem jaringan, seperti Gambar 2.5)
2. Memori sekunder yang *on-line* (*Harddisk*)
3. Memori sekunder *off-line* (*Tape* atau *Removable Disk*) untuk keperluan *Backup* data
4. Media atau perangkat komunikasi (untuk sistem jaringan)



Gambar 2.4 Sistem *Stand-Alone*



Gambar 2.5 Sistem Jaringan (*Multiuser*)

2.4.1.2 Sistem Operasi (Operating System)

Secara sederhana sistem operasi merupakan program yang mengaktifkan atau menjalankan sistem komputer, mengendalikan seluruh sumber daya (*resource*) dalam komputer dan melakukan operasi-operasi dasar dalam komputer (operasi *I/O*, pengelolaan file, dan lain-lain) sejumlah sistem operasi yang banyak digunakan seperti : MS-DOS, MS-Windows, UNIX, Linux, Mac, Sun-Solaris, dan lain-lain. Program basis data hanya akan aktif (*running*) jika sistem operasinya telah aktif.

2.4.1.3 Basis Data (Database)

Sebuah sistem basis data dapat memiliki beberapa basis data. Setiap basis data dapat berisi sejumlah objek basis data (seperti tabel, indeks dan lain-lain). Di samping menyimpan data, setiap basis data juga menyimpan definisi struktur basis data dan objek – objeknya secara detail).

2.4.1.4 DBMS (Database Management System)

Pengelolaan basis data secara fisik tidak dilakukan oleh pemakaian secara langsung, tetapi ditangani oleh sebuah perangkat lunak (sistem) yang khusus. Perangkat lunak ini disebut sebagai DBMS, yang akan menentukan bagaimana data diorganisasi, disimpan, diubah dan diambil kembali. DBMS juga menerapkan mekanisme pengamanan data, pemakaian data secara bersama, keakuratan dan konsistensi data.

Perangkat lunak yang termasuk DBMS adalah dBase III+, DbaseIV, FoxBase, RBase, MS-Access dan Borland Paradox, Borland Interbase, MS-SQLServer, MySQL, CA-Open Ingres, Oracle, Informix dan Sybase.

2.4.1.5 Pemakai (User)

Ada beberapa tipe pemakai terhadap suatu sistem basis data yang dibedakan berdasarkan cara mereka berinteraksi terhadap sistem, diantaranya :

1. *Programmer* Aplikasi

   Pemakai yang berinteraksi dengan basis data melalui *Data Manipulation Language* (DML), yang disertakan dalam program yang ditulis menggunakan bahasa pemrograman induk (C, Pascal, Cobol dan lain-lain)

2. *User* Mahir (*Casual User*)

   Pemakai yang berinteraksi dengan sistem tanpa menulis program. Mereka menyatakan *query* dengan bahasa yang sudah disediakan oleh DBMS.

3. *User* Umum (*End User*)

   Pemakai yang berinteraksi dengan sistem basis data melalui pemanggilan suatu program aplikasi permanen (*executable program*) yang telah disediakan sebelumnya

4. *User* Khusus (*Specialized User*)

   Pemakai yang menulis aplikasi basis data non konvensional, tetapi untuk keperluan-keperluan khusus, seperti untuk aplikasi *AI*, Sistem Pakar, Pengelolaan Citra, dan lain-lain, yang bisa saja mengakses basis data dengan/tanpa DBMS yang bersangkutan.

Untuk sebuah sistem basis data yang *stand-alone,* maka pada suatu saat hanya ada satu pemakai yang dapat bekerja. Sedangkan untuk sistem basis data dalam suatu jaringan, maka pada suatu saat ada banyak pemakai yang dapat berhubungan untuk menggunakan basis data yang sama.

2.4.1.6 Perangkat Lunak Lain

Aplikasi (perangkat lunak) ini bersifat *optional*. Artinya, ada atau tidak adanya tergantung pada kebutuhan DBMS yang digunakan dan lebih berperan dalam pengorganisasian data dalam basis data, sementara bagi pemakai basis data khususnya *end user* dapat disediakan program khusus untuk melakukan pengisian, perubahan, dan pengambilan data. Program ini ada yang sudah disediakan bersama DBMS-nya ada juga yang harus dibuatkan sendiri dengan aplikasi lain yang khusus.

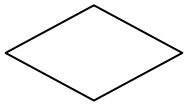
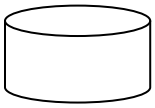
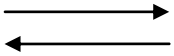
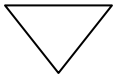
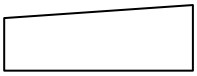
2.5 Perancangan Sistem

Tahap perancangan disebut juga tahap pemecahan masalah, yaitu dengan menyusun suatu algoritma, alur sistem, masukan, prosedur proses, keluaran, dan database. Proses perancangan diperlukan untuk menghasilkan suatu rancangan sistem yang baik, karena dengan rancangan yang tepat akan menghasilkan sistem yang stabil dan mudah dikembangkan di masa mendatang. Berikut ini akan dijelaskan rangkaian atau ruang lingkup sistem yang akan dirancang dengan memanfaatkan alat bantu seperti :

2.5.1 Flowmap

Flow Map berfungsi mendefinisikan hubungan antara bagian (pelaku proses), proses (manual/berbasis komputer) dan aliran data (dalam bentuk dokumen keluaran/masukan). Berikut simbol-simbol sistem prosedur diagram/flow map.

Tabel 2.1 Daftar simbol *flow map*

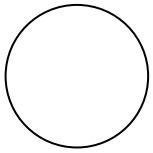
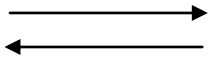
SIMBOL	NAMA	KETERANGAN
	Proses	Untuk menunjukkan proses yang dilakukan secara komputer
	Dokumen	Untuk menunjukkan dokumen baik untuk proses komputer maupun manual
	Operasi Manual	Untuk menunjukkan proses yang dilakukan secara manual
	Keputusan	Untuk menunjukkan pilihan proses yang dilakukan secara komputer maupun manual
	Tampilan	Untuk menampilkan Data
	Basis Data	Sebagai tempat penyimpanan data secara komputer
	Garis Alir	Arah aliran dari proses atau dokumen
	Arsip	Sebagai tempat penyimpanan data secara manual
	Manual *Input*	Untuk menunjukkan proses *input* yang dilakukan secara manual

2.5.2 Diagram Konteks

Diagram konteks adalah diagram yang menggambarkan ruang lingkup secara global seperti interaksi entitas luar terhadap sistem, entitas disini seringkali disebut sebagai terminator dapat berupa sistem lain, suatu perangkat keras, orang atau organisasi. Diagram konteks adalah diagram yang memperlihatkan sistem

sebagai suatu proses yang berfungsi untuk memetakan model lingkungan yang mempresentasikan lingkaran tunggal yang mewakili keseluruhan sistem.

Tabel 2.2 Daftar simbol Diagram Konteks

SIMBOL	NAMA	KETERANGAN
	Terminator	Digunakan untuk berkomunikasi langsung dengan sistem aliran data
	Proses	Berfungsi untuk mewakili sistem
	Aliran Data	Menunjukkan arah dari aliran data

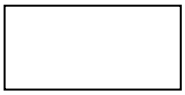
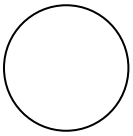
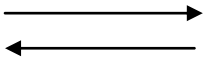
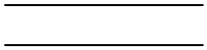
2.5.3 *Data Flow Diagram* (DFD)

Data Flow Diagram atau diagram alir data adalah perangkat pemodelan sistem yang menggambarkan ruang lingkup dari sistem berupa keterkaitan lingkungan dengan sistem. DFD ini cenderung memproseskan model-model yang terjadi dalam sistem dan data yang mengalir pada sistem serta interaksi entitas luar sistem. Pada DFD ini terdapat beberapa tingkat pemodelan, yang pertama adalah tingkat paling tinggi yaitu diagram konteks, yang kedua adalah DFD level 0, level 1, level 2, level 3, dan seterusnya.

Diagram konteks adalah diagram yang menggambarkan ruang lingkup secara global seperti interaksi entitas luar terhadap sistem, entitas disini seringkali disebut sebagai terminator. Sementara DFD leveled adalah diagram yang menggambarkan ruang lingkup secara bertahap (leveled) dan sistem berupa

keterkaitan entitas dengan proses-proses yang ada. DFD leveled menggambarkan proses-proses mulai dari tingkat global (level 0) hingga ke proses yang spesifik atau primitif agar memudahkan penggambaran dari sebuah sistem.

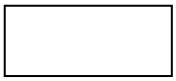
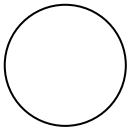
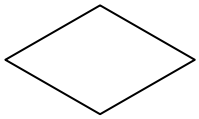
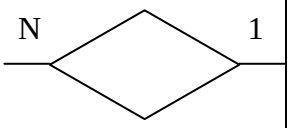
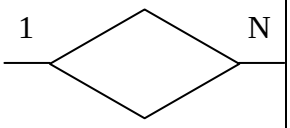
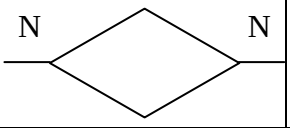
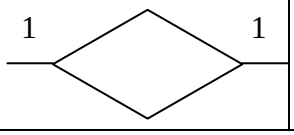
Tabel 2.3 Daftar simbol *Data Flow Diagram*

SIMBOL	NAMA	KETERANGAN
	Entitas Luar	Menggambarkan kegiatan proses dari operasi program komputer
	Proses	Menggambarkan kegiatan atau kerja yang dilakukan oleh manusia atau komputer
	Aliran Data	Menunjukkan arus data yang mengalir antara proses, simpanan data dan kesatuan luar dari ke suatu arus
	Simpanan Data	Menggambarkan suatu tempat penyimpanan data

2.5.4 *Entity Relationship Diagram* (ERD)

Entity Relationship Diagram atau diagram hubungan entitas adalah model konsep yang mendeskripsikan relasi antar data dalam aliran data. ERD digunakan untuk memodelkan struktur data dan hubungan antar data. Pada ERD ini dikenal hubungan atau kardinalitas (*Cardinality*) yaitu hubungan antar entitas dengan nilai hubungan yang beragam dan akan diterjemahkan ke dalam table.

Tabel 2.4 Daftar simbol *Entity Relationship Diagram*

SIMBOL	NAMA	KETERANGAN
	Himpunan Entitas	Menunjukkan suatu objek berupa benda, konsep atau entitas
	Atribut	Menunjukkan atribut suatu entitas
	Hubungan/Relasi	Menunjukkan hubungan *logical* antara satu *file* dengan *file* lainnya
	Hubungan (N-1)	Menunjukkan hubungan banyak ke satu (N-1)
	Hubungan (1-N)	Menunjukkan hubungan satu ke banyak (1-N)
	Hubungan (N-N)	Menunjukkan hubungan banyak ke banyak (N-N)
	Hubungan (1-1)	Menunjukkan hubungan satu ke satu (1-1)

2.5.5 Kamus Data (*Data Dictionary*)

Kamus Data adalah peralatan yang ikut berperan dalam perancangan dan pembangunan system informasi karena berfungsi untuk menjelaskan atri aliran data dan penyimpanan dalam penggambaran pada *data flow diagram*, mendeskripsikan komposisi paket data yang bergerak melalui aliran, dan

menjelaskan spesifikasi nilai dan satuan yang relevan terhadap data yang mengalir dalam sistem tersebut.

2.6 Jaringan Komputer

Jaringan Komputer adalah sekumpulan peralatan komputer (hardware dan software) yang dihubungkan agar dapat saling berkomunikasi dengan tujuan komunikasi dan berbagi sumber daya (seperti file dan printer).

2.6.1 Klasifikasi Jaringan Komputer

Berdasarkan kriterianya, jaringan komputer dibedakan menjadi 4 yaitu :

1. Berdasarkan distribusi sumber informasi/data.

   a) Jaringan terpusat

   Jaringan ini terdiri dari komputer *client* dan *server*, dimana komputer *client* yang berfungsi sebagai perantara untuk mengakses sumber informasi/data yang berasal dari satu komputer *server*.

   b) Jaringan terdistribusi

   Merupakan perpaduan beberapa jaringan terpusat sehingga terdapat beberapa komputer server yang paling berhubungan dengan *client* membentuk sistem jaringan tertentu.

2. Berdasarkan jangkauan geografis.

   a) Jaringan LAN

   Merupakan jaringan yang mencakup luas daerah yang kecil

   b) Jaringan MAN

   Merupakan jaringan yang mencakup satu kota besar

c) Jaringan WAN

Merupakan jaringan yang mencakup seluruh dunia

3. Berdasarkan peranan dan hubungan tiap komputer dalam memproses data.

a) Jaringan *Client-Server*

Pada jaringan ini terdapat 1 atau beberapa komputer server dan komputer client. Komputer yang akan menjadi komputer server maupun menjadi komputer client dan diubah-ubah melalui software jaringan pada protokolnya. Komputer client sebagai perantara untuk dapat mengakses data pada komputer server sedangkan komputer server menyediakan informasi yang diperlukan oleh komputer client.

b) Jaringan *Peer-to-peer*

Pada jaringan ini tidak ada komputer *client* maupun komputer *server* karena semua komputer dapat melakukan pengiriman maupun penerimaan informasi sehingga semua komputer berfungsi sebagai *client* sekaligus sebagai *server*.

4. Berdasarkan media transmisi data

a) Jaringan Berkabel (*Wired Network*)

Pada jaringan ini, untuk menghubungkan satu komputer dengan komputer lain diperlukan penghubung berupa kabel jaringan. Kabel jaringan berfungsi dalam mengirim informasi dalam bentuk sinyal listrik antar komputer jaringan.

b) Jaringan Berkabel (*Wired Network*)

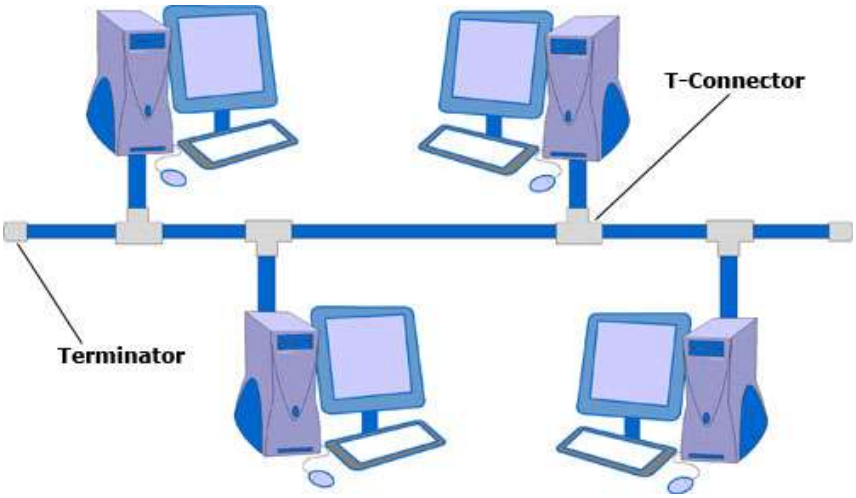
Merupakan jaringan dengan medium berupa gelombang elektromagnetik. Pada jaringan ini tidak diperlukan kabel untuk menghubungkan antar komputer karena menggunakan gelombang elektromagnetik yang akan mengirimkan sinyal informasi antar komputer jaringan.

2.6.2 Topologi Jaringan Komputer

Topologi menggambarkan struktur dari suatu jaringan atau bagaimana sebuah jaringan didesain. Berikut ini beberapa model Topologi yang ada saat ini.

2.6.2.1 Topologi Bus

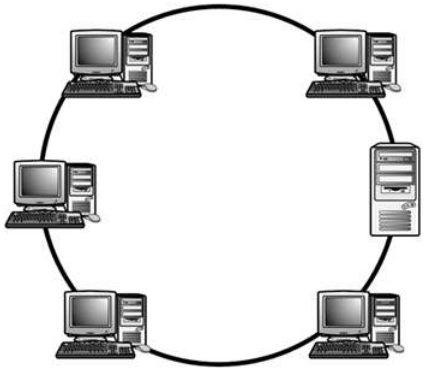
Topologi ini menggunakan satu *segment* (panjang kabel) *backbone*, yaitu yang menyambungkan semua host secara langsung. Apabila komunikasinya dua arah di sepanjang ring, maka jarak maksimum antara dua simpul pada ring dengan simpul adalah n/2. Topologi ini cocok untuk jumlah prosesor yang relatif sedikit dengan komunikasi data minimal. Model Topologi Bus dapat dilihat pada gambar 2.6



Gambar 2.6 Topologi Bus

2.6.2.2 Topologi Ring

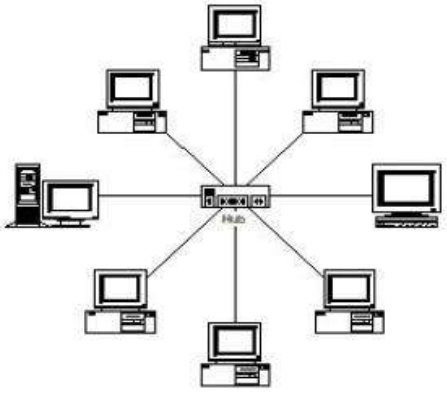
Topologi ini menghubungkan satu *host* ke *host* setelah dan sebelumnya. Secara fisik jaringan ini berbentuk *ring* (lingkaran). Model Topologi Ring dapat dilihat pada gambar 2.7



Gambar 2.7 Topologi Ring

2.6.2.3 Topologi Star

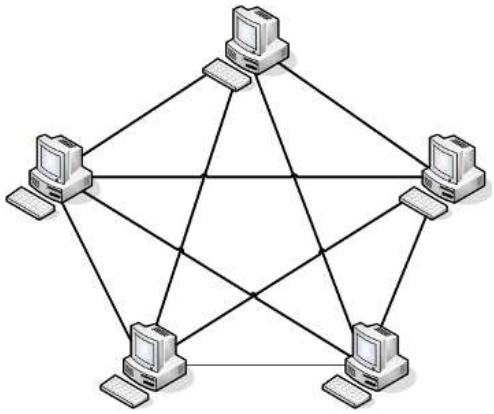
Menghubungkan semua kabel pada *host* ke satu titik utama. Titik ini biasanya menggunakan *Hub* atau *Switch*. Topologi bintang merupakan bentuk topologi jaringan yang berupa konvergensi dari *node* tengah ke setiap *node* atau pengguna. Model Topologi Star dapat dilihat pada gambar 2.8



Gambar 2.8 Topologi Star

2.6.2.4 Topologi Mesh

Topologi Mesh adalah suatu topologi yang memang didisain untuk memiliki tingkat restorasi dengan berbagai *alternatif* rute atau penjaluran yang biasanya disiapkan dengan dukungan perangkat lunak atau *software*. Model Topologi Mesh dapat dilihat pada gambar 2.9



Gambar 2.9 Topologi Mesh

2.6.3 Manfaat Jaringan

1. *Resouce Shering*

Dalam menggunakan sumberdaya yang ada secara bersama – sama, missal seorang pengguna yang berada 100 km jauhnya dari suatu data, tidak mendapatkan kesulitan dalam menggunakan data tersebut, seolah – olah data tersebut berada didekatnya. Hal ini sering dikaitkan bahwa jaringan komputer mengatasi masalah jarak.

2. Realibiltas Tinggi

Dalam jaringan komputer kita akan mendapatkan realibilitas yang tinggi dalam memiliki sumber alternatif persediaan, misalnya semua file dapat disimpan atau dicopy ke dua, tiga atau lebih komputer yang terkoneksi kejaringan. Sehingga bila salah satu mesin rusak, salinan dimesin yang lain bisa digunakan.

3. Menghemat Biaya

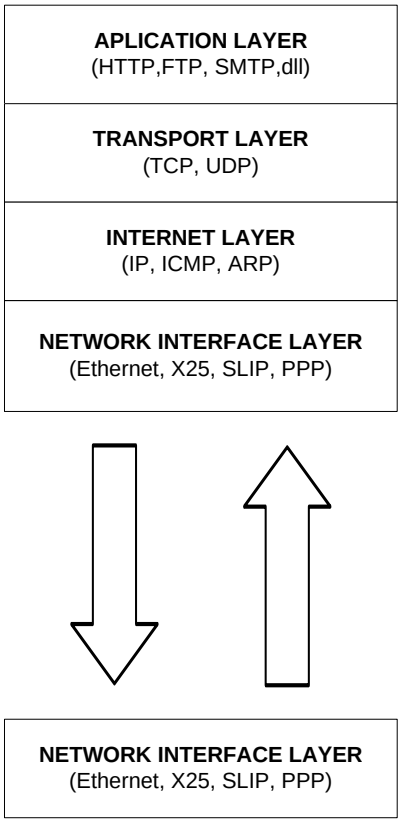
Komputer berukuran kecil mempunyai rasio harga / kinerja yang lebih baik dibandingkan dengan komputer yang besar. Komputer besar seperti *mainframe* memiliki kecepatan kira – kira sepuluh kali lipat kecepatan komputer kecil / pribadi. Ketidak seimbangan rasio harga / kinerja dan kecepatan inilah membuat para perancang sistem untuk membangun system yang terdiri dari komputer – komputer pribadi.

2.7 Internet

Internet adalah himpunan informasi dan sumber daya komputer yang paling besar di dunia. Internet merupakan koneksi berbagai macam jaringan komputer dengan berbagai jenis *hardware* dan teknologi yang berbeda. Komunikai yang dibangun pada jaringan internet menggunakan mekanisme tertentu yang disebut *protocol*. *Protocol* yang digunakan untuk menjalankan komunikasi antar jaringan komputer pada internet ini adalah *Transmission Control Protocol/Internet Protocol* (TCP/IP). Dengan menggunakan protokol

yang sama yaitu TCP/IP, perbedaan jenis komputer dan sistem operasi tidak menjadi masalah.

TCP/IP pada awal perkembangannya digunakan oleh *Defense Advanced Research Project Agency* (DARPA) untuk membangun jaringan untuk keperluan militer. Nama TCP/IP sendiri diambil dari nama protokol pada layer *network* yaitu *Internet Protocol* (IP) dan *protokol* pada layer transfort yaitu *Transmission Control Protocol* (TCP). Dengan mengacu pada nama ini, TCP/IP internet kemudian sering disebut sebagai internet. Arsitektur protokol TCP/IP ini dapat dimodelkan dengan empat layer sebagaimana yang dapat dilihat pada gambar 2.10

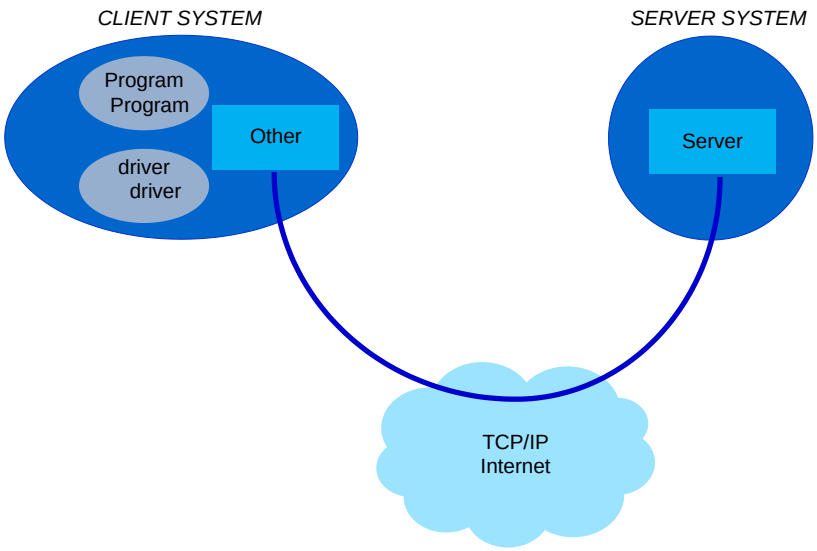


Gambar 2.10 Arsitektur *Protocol* TCP/IP

TCP/IP terdiri dari empat lapis kumpulan protokol yang bertingkat. Empat perangkat itu adalah :

1. *Application Layer*, yaitu layer yang bertanggung jawab terhadap aplikasi-aplikasi yang menggunakan TCP/IP.
2. *Transport Layer*, yaitu layer yang bertanggung jawab memberikan layanan pengiriman data dari satu node ke node yang lain.
3. *Internet Layer*, yaitu layer yang menentukan jenis datagram dan menangani perutean data yang dikirim.
4. *Network Interface Layer*, yaitu layer yang bertanggung jawab untuk mengirim dan menerima data pada media fisik transmisi yang digunakan.

Proses yang terjadi pada internet mempunyai mekanisme yang telah diatur berdasarkan protokol standar. Proses yang terjadi di internet adalah ada satu program yang meminta kepada program yang lain untuk memberikan suatu layanan. Dua program tersebut dapat terletak dalam subnet yang sama ataupun berbeda. Komputer program yang meminta layanan disebut dengan *client* sedangkan yang memberikan layanan disebut *server*.



Gambar 2.11 Model *Client Server Internet*

2.7.1 Media

Kata media berasal dari bahasa latin *Medius* yang secara *harafiah* berarti tengah, perantara, atau pengantar. Namun pengertian media dalam proses pembelajaran cenderung diartikan sebagai alat-alat grafis, fotografis atau elektronis untuk menangkap, memproses, dan menyusun kembali informasi visual atau verbal.

2.7.2. Web Portal

Web portal disebut dengan portal atau portal internet (*internet portal*). Di dalam dunia internet istilah ini dimaksudkan untuk website yang menyediakan beraneka ragam informasi untuk para pengunjungnya. Portal adalah aplikasi berbasis web yang menyediakan fasilitas akses dari satu titik terhadap sumber-sumber informasi online terdistribusi, seperti dokumen yang dapat melalui

pencarian, kanal berita dan link ke situs khusus. Untuk mempermudah user, biasanya disediakan kemampuan pencarian dan pengorganisasian informasi.

Portal dalam perkembangannya tidak hanya mendukung satu komunitas tapi beberapa komunitas. Portal juga mendukung berbagai macam servis atau layanan yang membedakannya dengan web site HTML biasa. Portal juga melibatkan berbagai macam komponen teknik yang mendukung berfungsinya portal tersebut. Portal juga terdiri dari berbagai macam infrastruktur yang mendukung portal itu berdiri. Karena tanpa infrastruktur ini dia tidak akan dapat memberikan layanan. Untuk bisa mendirikan dan memberikan portal ada beberapa strategi dan implementasi yang harus diperhatikan oleh pengolahnya.

2.7.3 Web Server

Web server adalah server internet yang mampu melayani koneksi transfer data dalam protokol HTTP. Web server dirancang untuk melayani bahasa jenis data, muali dari text, *hypertext*, gambar (*image*), suara, *plug in*, dan lain sebagainya. Web server pada umumnya melayani data dalam bentuk file HTML.

2.7.4 *Domain Name Server* (DNS)

Untuk mengidentifikasi suatu entitas, protokol TCP/IP menggunakan alamat IP. Namun apabila dalam aplikasi setiap orang harus menghafal alamat IP untuk melakukan komunikasi bisa berakibat timbulnya kesulitan untuk mengingat. Apalagi jika perkembangan internet sudah demikian pesat. Untuk itu protokol TCP/IP memiliki suatu metode untuk membuat suatu map yang menterjemahkan nama kepada alamat IP atau sebaliknya. Metode ini disebut juga sebagai *Domain Name System* (DNS).

Domain Name System (DNS) adalah *distribute database system* yang digunakan untuk pencarian nama komputer (*name resolution*) di jaringan yang menggunakan TCP/IP. DNS bisa digunakan pada aplikasi yang terhubung ke internet seperti web browser atau e-mail, dimana DNS membantu memetakan host name sebuah komputer ke IP address. Selain digunakan di Internet, DNS juga dapat di implementasikan ke *private network* atau *internet* dimana DNS memiliki keunggulan seperti:

1. Mudah, DNS sangat mudah karena *user* tidak lagi direpotkan untuk mengingat IP address sebuah komputer cukup *host name* (nama Komputer).
2. Konsisten, IP address sebuah komputer bisa berubah tapi *host name* tidak berubah.
3. *Simple*, *user* hanya menggunakan satu nama domain untuk mencari baik di Internet maupun di Intranet.

2.7.5 *World Wide Web* (WWW)

World Wide Web (WWW) merupakan suatu sistem penyedia informasi dengan skala yang besar yang mengelola informasi tersebut secara terdistribusi dalam internet dengan menggunakan teknologi hypermedia. WWW mempunyai karakteristik:

a. Dukungan *interface* yang terintegrasi

   WWW menyediakan suatu *interface* yang terintegrasi terhadap protokol, format data, sistem pengalamatan, dan lain-lain sehingga berbagai macam layanan dan basis data yang ada di internet dapat diakses secara langsung.

b. Kemudahan pada sisi pengguna

WWW mendukung secara transparan sebagian besar aplikasi-aplikasi yang terdapat di internet seperti layanan *telnet, gopher, anonymous ftp, finger,* dan lain-lain. Kombinasi layanan ini memberikan *user interface* yang terintegrasi dengan aplikasi-aplikasi internet yang lain.

c. Kemudahan dalam perkembangan

Kapabilitas *server* WWW dapat dikembangkan secara mudah dengan menggunakan standar PHP antara server WWW dengan aplikasi yang lain. Program PHP juga memungkinkan perubahan informasi secara dinamis yang dapat diperoleh secara *real time*.

d. Tidak tergantung pada *platform* tertentu

WWW memungkinkan seseorang dapat membangun *server* WWW di berbagai sistem yang berlainan dan memberikan informasi dalam bentuk *hypermedia* WWW tidak didesain untuk mendukung sistem tertentu.

WWW menggunakan model *client server*. WWW juga menggunakan ekspresi informasi, transfer informasi, metode penamaan informasi yang standar sehingga dapat memproses dan mentransfer informasi secara terdistribusi yang dilakukan secara sistematik.

Informasi yang terdistribusi disimpan di dalam *server* WWW dan *user* mengakses informasi tersebut menggunakan suatu *software* yang disebut dengan browser. *Server* menyimpan informasi di dalam file-file terstruktur menggunakan *Hypertext Markup Language* (HTML). *Hypertext* merupakan kumpulan teks-teks yang saling berhubungan satu sama lainnya. *Hypermedia* serupa dengan *hypertext*,

hanya saja media yang digunakan bukan hanya teks akan tetapi meliputi juga *audio*, *image*, dan *video*.

WWW menggunakan standar untuk mencapai konsistensi dalam menghasilkan dan melakukan transfer informasi. WWW menggunakan HTML sebagai standar produksi informasi dengan menggunakan *Hypertext Transfer Protocol* (HTTP) sebagai standar transfer informasi. WWW menggunakan standar penamaan yang disebut dengan *Universal Resource Locator* (URL) untuk mendukung penyediaan informasi secara *universal* dalam *server* yang terdistribusi.

2.7.6 HTTP

HTTP merupakan sebuah protokol yang didesain untuk men-*transfer* informasi dalam bentuk *hypermedia* antara *server* dengan sebuah *client*. HTTP juga men-*transfer* data suatu informasi melalui *header*-nya. *Header* HTTP ini merupakan bentuk pengembangan dari *Multipurpose Internet Mail Extentions* (MIMEs). Pengembangan ini memungkinkan HTTP untuk men-*transfer* informasi dalam bentuk biner dan informasi dalam bentuk yang tidak standar yang berhasil dinegosiasi antara *server* dan *client*. Secara umum, *delay* akan terjadi pada saat melakukan suatu negosiasi sebelum proses *transfer* data. Karena lama dari *delay* yang disebabkan oleh *overhead* akan menjadi *relative* lebih besar pada *transfer* data yang sebetulnya singkat.

HTTP merupakan protokol yang sederhana yang didesain untuk mereduksi *delay*. HTTP merupakan *protokol* yang bersifat *stateless*, sehingga *server* akan memproses setiap *request* dan *user* secara terpisah dan *request* yang lain,

independen terhadap *request* yang sebelumnya. HTTP menggunakan 8 bit untuk mentransfer semua tipe data yang mungkin.

Mekanisme yang terjadi pada HTTP bila suatu *client* menginginkan layanan dari *server* di bagi dalam 4 langkah, yaitu:

1. *Connection Setup*

   Client mengakses sebuah *server* dengan menggunakan *internet address* dan *port number*. *Default* dari *port number* adalah 80.

2. *Request*

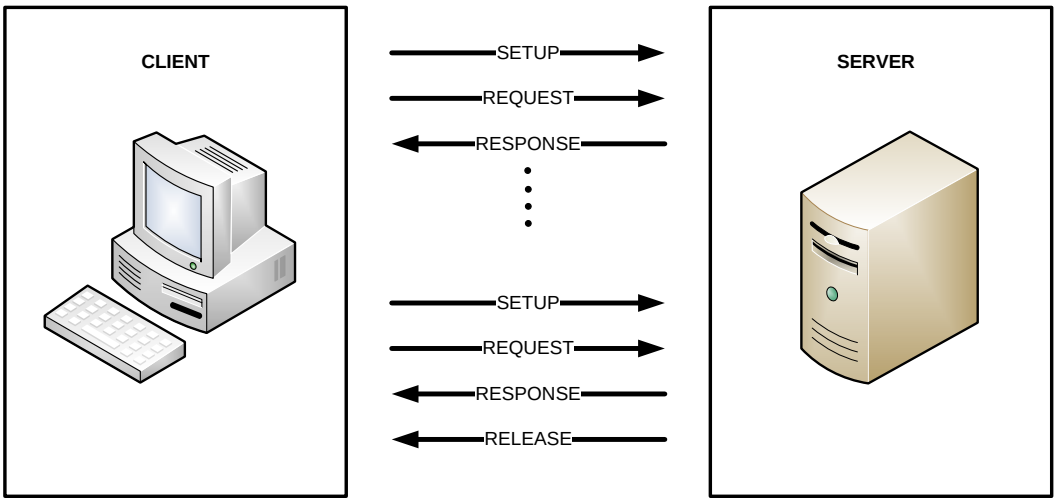
   Client mengirimkan *message* berupa informasi dari metode transaksi dan kapabilitas *client*.

3. *Response*

   Server mengirimkan *response* kepada *client* sesudaha itu menyelesaikan *request*-nya. *Response message* meliputi informasi dari transaksi dan data yang diminta.

4. *Connection Release*

   Client mengakhiri koneksi ke *server*.

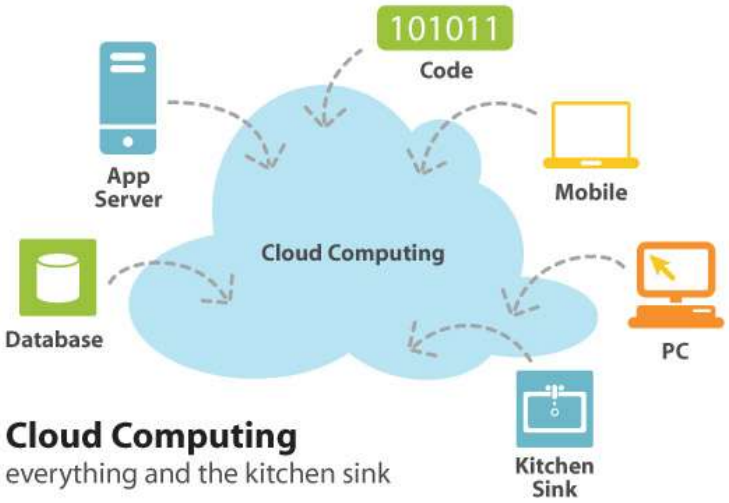


Gambar 2.12 Proses Koneksi *Protokol* HTTP antara *client* dan *server*

2.8 Cloud Computing

Cloud Computing terdiri dari dua kata yaitu "*cloud*" yang berarti awan dan "*computing*" yang berarti komputasi. Kunci dari *cloud computing* sebenarnya terletak pada kata "*cloud*" atau "awan", ini dapat dikatakan juga sebagai internet yang merupakan suatu lingkungan yang cukup besar yang didalamnya terdapat komputer yang saling berhubungan. Komputer tersebut bisa *PC Personal* atau suatu jaringan komputer, yang masing-masing bisa bersifat *public* atau *private*. Dapat dilihat pada Gambar 2.13

Sebelumnya pada komputer *desktop* biasa, perangkat lunak dijalankan pada komputer tersebut. Semua dokumen (*file*) yang dibuat, diolah dan disimpan pada komputer tersebut, ini dikenal dengan istilah PC Sentris. Sedangkan *cloud computing* menyediakan *storage* dan perangkat lunak pada server yang ada di internet, ini dikenal dengan istilah Dokumen Sentris.



Gambar 2.13 Ilustrasi *Cloud Computing*
(Sumber : David Robbins, 2009)

2.8.1 Definisi *Cloud Computing*

Terdapat beberapa sudut pandang terhadap pengertian atau definisi tentang *cloud computing*, diantarnya :

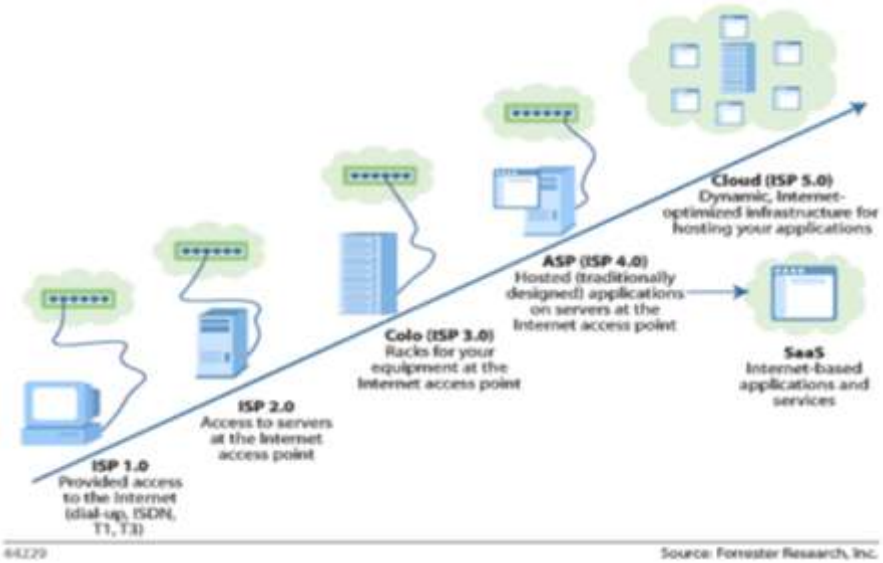
1. *Cloud Computing* adalah teknologi komputasi komputer dengan memanfaatkan internet sebagai terminal utamanya guna mengelola piranti lunak hingga infrastruktur sebagai bentuk layanannya.
2. *Cloud Computing* adalah evolusi selanjutnya dari internet (“awan”) mulai dari tenaga komputasi hingga infrastruktur komputasi, aplikasi-aplikasi, proses bisnis, hingga kolaborasi yang muncul sebagai layanan yang dapat diakses pada saat dibutuhkan kapanpun dan dimanapun.
3. *Cloud Computing* adalah sebuah gaya komputasi yang besar, elastis dan terukur, yang berhubungan dengan kemampuan IT sebagai penyedia

layanan (*as a service*) kepada pelanggan eksternal menggunakan teknologi internet.

Cloud computing biasanya dipakai pada suatu perusahaan, organisasi, ataupun komunitas. Aplikasi dan data berada pada komputer *server* yang bersifat "*cloud*" untuk digunakan oleh kelompok tertentu pada suatu perusahaan tersebut, baik lintas perusahaan maupun lintas *platform* dan diakses via internet. Sehingga para pengguna yang berhak mengakses *file* atau aplikasinya dapat menggunakan komputer manapun yang terhubung dengan koneksi internet. *Services* yang ada pada teknologi *cloud* berbasis HTTP, HTML, XML, JavaScript, teknologi sejenis yang lainnya.

2.8.2 Evolusi *Cloud Computing*

Salah satu faktor yang menjadi kunci terjadi evolusi *cloud computing* adalah adanya *Internet Service Provider* (ISP) yang ada pada saat ini. Dapat dilihat pada Gambar 2.14



Gambar 2.14 Evolusi dari ISP
(Sumber :Forrester Research, Inc)

Pertama kali internet disediakan adalah untuk menawarkan aplikasi sebagai *service,* ISP 1.0 lah yang menyediakan fasilitas koneksi internet untuk kebutuhan tersebut. Selanjutnya, ISP 2.0 menambah fasilitasnya dengan memberikan kemampuan untuk menyimpan data atau menyediakan *hosting*. Lalu, pada tahap selanjutnya, ISP mulai menyewakan *space* dan *bandwidth* mereka. ISP 3.0 sudah berkembang dengan menyediakan untuk berlangganan aplikasi yang nantinya menghasilkan model *Application Service Provider* (ASP). *Software as a service*, merupakan model terakhir yang ada saat ini untuk ASP tersebut.

2.8.3 Sejarah *Cloud Computing*

Pada waktu itu, para *developers* dan arsitekur teknologi menggunakan sebuah gambar awan untuk mengilustrasikan sumber daya yang dapat di-*remote* (diakses) via web. Sejak saat itulah mereka mendefinisikan bahwa awan sangat logis untuk dijadikan suatu penghubung antara media lokal dengan sumber daya yang ada pada suatu jaringan internet. Kebanyakan *developers* kebingungan untuk memastikan terminologi dari suatu *cloud computing.* Sebenarnya para *developers* tersebut sudah menyediakan suatu *host* pada "*cloud"* dan ini bisa dikatakan sebagai "*Cloud Services*". Hanya saja dari kebenaran argument tersebut, deskripsi dari *Cloud Computing* belum cukup akurat, mereka beranggapan bahwa syarat *cloud computing* adalah:

1. *Web Services*

   Konsep sederhana ini berfungsi untuk menerima *request* dari pengguna dan mengrimkan respon kepada pengguna tersebut, *request* ini bisa berupa kumpulan kode.

2. *Cloud OS (Operating Systems)*

   Cloud OS bertindak sebagai *interface* antara *web service* dengan CPU atau perangkat keras lainnya agar dapat memanggil layanan yang ada pada *cloud* lalu memproses layanan tersebut.

3. *Cloud FX*

   Ini berfungsi untuk para *developer* yang ingin mengembangkan layanan atau aplikasi *cloud*. Ini bisa berupa bahasa pemrograman seperti yang biasa digunakan oleh para developer, yaitu .NET, SDK dan lain-lain.

4. *Cloud Application*

   Ini merupakan aplikasi yang mampu berjalan pada *cloud*, dan bisa diakses oleh para pengguna

5. *Infrastructure as a Service*

6. *Platform as a Service*

7. *Software as a Service*

8. *IT Professionals* dan *System Administrator*

   Karakterisitik ini sangat berpengaruh untuk kelangsungan dari *cloud computing*, ini bertugas untuk me*monitoring*, mengatur, dan bertanggung jawab atas layanan yang telah dibangun.

9. *Developers* dan Arsitektur Teknologi

   Dibutuhkan *developer* atau arsitektur teknologi untuk membangun dan mengembangkan aplikasi *cloud computing.*

10. Pengguna

    Tanpa adanya pengguna yang menggunakan aplikasi *cloud* yang telah dibuat oleh para pembangun aplikasi, maka *cloud computing* ini akan terasa mubazir.

2.8.4 Sifat *Cloud Computing*

Ada enam sifat yang dimiliki oleh *cloud computing*, yaitu :

1. *User-Centric*

   Setiap pengguna menyimpan dan/atau mengakses dokumen, aplikasi, gambar, pesan, dan lain-lain, data-data tersebut akan menjadi pengguna tersebut. Tidak hanya itu pengguna tersebut pun dapat men-*share* kepada pengguna lainnya.

2. *Task-Centric*

   Tidak hanya berfokus pada aplikasinya dan apa yang mampu aplikasi kerjakan, tetapi *cloud computing* berfokus pada bagaimana memenuhi pekerjaan dan kebutuhan pengguna dan bagaimana aplikasi tersebut dapat melakukannya. Contoh, *word-processing, spreadsheets, email,* dan lain-lain.

3. *Powerfull*

   Cloud computing memungkinkan pengguna dapat terhubung dengan ratusan bahkan ribuan komputer secara bersamaan, ini menjadi suatu kekuatan bagi *cloud computing* yang mustahil dilakukan oleh PC *desktop.*

4. *Accessible*

   Karena data tersimpan di internet dan tidak dibatasi oleh satu penyimpanan saja, pengguna dapat secara instan mengakses datanya dimanapun, kapanpun.

5. *Intelligent*

   Dengan banyaknya variasi data yang tersimpan pada "*cloud*", maka ada cara yang dilakukan untuk menganalisis dan memilah informasi dari data yang tersimpan tersebut.

6. *Programmable*

   Banyak sekali tugas atupun proses yang dilakukan pada *cloud computing* secara otomatis. Sebagai contoh, jika pada komputer yang ada pada *cloud* mengalamai offline, maka program akan didistribusikan kepada komputer lain yang *online* yang ada pada "*cloud*" sehingga data yang disimpan selalu bisa diakses.

2.8.5 Model Layanan *Cloud Computing*

Konsep dari pengembangan layanan sistem *cloud* dibagi menjadi beberapa tipe layanan, diantaranya adalah :

1. *Software as a Service (SaaS)*
2. *Platform as a Services (PaaS)*
3. *Infrastructure as a Service (Iaas)*

2.8.5.1 *Software as a Service* (SaaS)

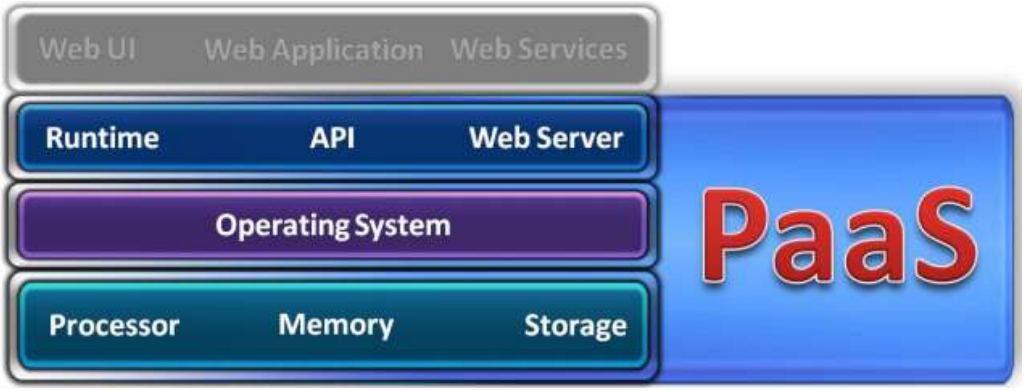
Software as a Service (*SaaS*) merupakan evolusi lanjutan dari konsep ASP (*Application Service Provider*). *Software as a Service* adalah istilah terhadap *software* atau aplikasi tertentu berbasis internet yang ditawarkan oleh *provider* kepada pengguna. Dalam hal ini, *provider* sebagai pemegang *license* atau *software* tersebut hanya memberikan *service* atau layanan kepada pengguna untuk menggunakannya sesuai kebutuhan pengguna dengan demikian menghilangkan kerumitan dalam hal pemeliharaan *software*, operasional dan *support*. *License*, *maintenance*, *support*, tingkat kenyamanan dan keamanan atas *software* tersebut sepenuhnya menjadi tanggung jawab dari *provider*. Dapat dilihat pada Gambar 2.15



Gambar 2.15 Model Layanan SaaS
(Sumber : Exigen Insurance Solutions, Inc, 2009)

2.8.5.2 *Platform as a Service* (PaaS)

Model layanan *platform as a service* merupakan layanan yang menyediakan *platform* seperti bahasa pemrograman, *tools*, ataupun sistem operasi yang nantinya mampu mengembangkan aplikasi berbasis konsumen. Keuntungan dari PaaS adalah bahwa perusahaan yang menggunakan layanan ini tidak perlu khawatir untuk instalasi, perawatan, dan keamanan pada server karena *provider* PaaS akan menangani itu semua. Dapat dilihat pada Gambar 2.16



Gambar 2.16 Model Layanan PaaS
(Sumber : Exigen Insurance Solutions, Inc, 2009)

2.8.5.3 *Infrastructure as a Service* (IaaS)

Model layanan disediakan dari IaaS adalah sumber daya pemroses, storage, kapasitas jaringan, dan sumber daya komputasi lainnya. Dimana konsumen dapat mengembangkan dan menjalankan aplikasi khusus. IaaS memungkinkan pengguna untuk menjalankan beberapa tugas komputer pada mesin *virtual* yang tak terbatas. Dapat dilihat pada Gambar 2.17



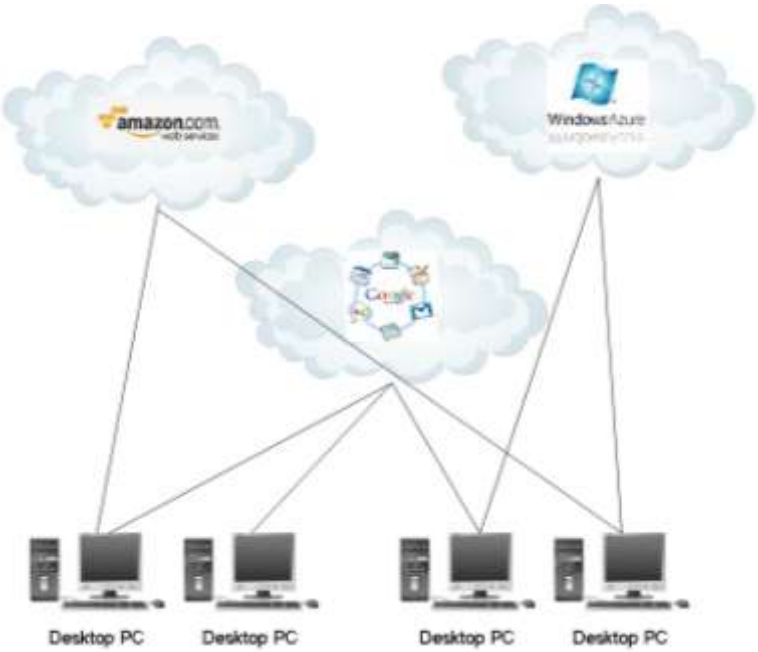
Gambar 2.17 Model Layanan IaaS
(Sumber : Exigen Insurance Solutions, Inc, 2009)

2.8.6 Model Pengembangan *Cloud Computing*

Ada beberapa model *cloud computing* yang sudah dikembangkan saat ini dengan berbagai fungsi dari masing-masing model pengembangannya, yaitu :

1. *Public Cloud*

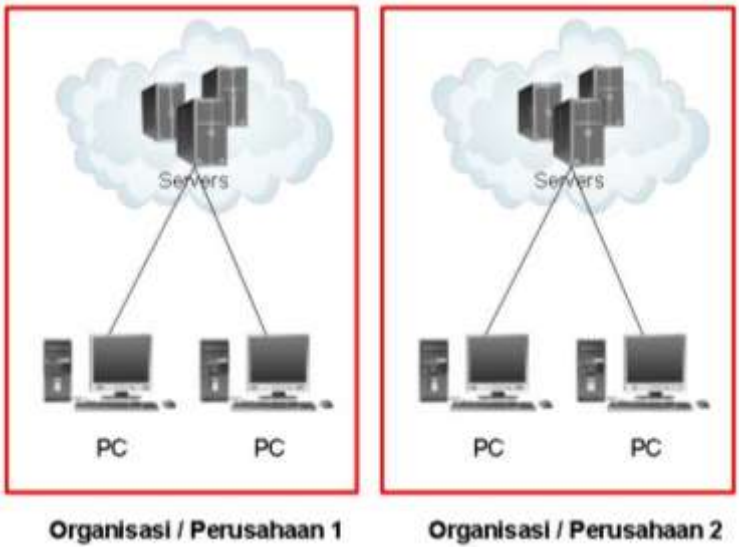
   Model pengembangan ini adalah model pengembangan yang paling popular. Model ini diimplementasikan kepada berbagai server yang berjalan pada berbagai *data center* agar dapat diakses dibelahan bumi manapun. Ini akan mengurangi *latency*, atau beban dari server ketika pengguna dimanapun mengakses data yang telah disediakan. Model pengembangan ini telah dilakukan oleh salah satunya adalah Google. Dapat dilihat pada Gambar 2.18



Gambar 2.18 Model Pengembangan *Public Cloud*
(Sumber : Exigen Insurance Solutions, Inc, 2009)

2. *Private Cloud*

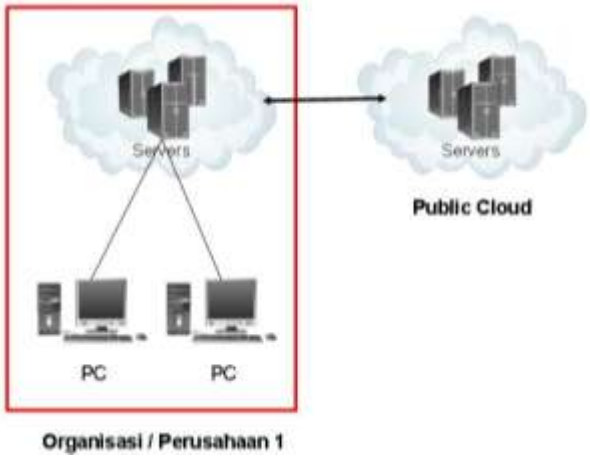
   Model pengembangan ini biasanya dilakukan oleh suatu perusahaan atau organisasi dengan memegang atribut dari *cloud computing* yaitu, elastisitas, *Self Service, Pay-by-Use,* dan *programmability*. Ini ada karena mampu mengefisiensikan biaya seperti mengurangi konsumsi listrik yang berlebihan, dan mengurangi pendingin untuk server mereka sebelumnya. Vendor yang menyediakan model ini diantaranya adalah vmWare, Microsoft, IBM, Sun, dan lain-lain. Dapat dilihat pada Gambar 2.19



Gambar 2.19 Model Pengembangan *Private Cloud*
(Sumber : Exigen Insurance Solutions, Inc, 2009)

3. *Hybrid Cloud*

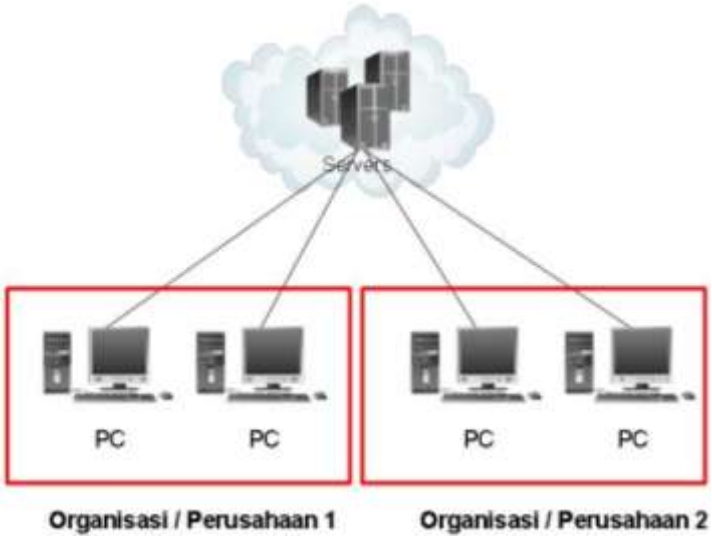
   Model ini merupakan kombinasi atau gabungan dari model *public* dan *private* yang masih memiliki entitas unik namun terikat bersama-sama oleh standar teknologi atau kepemilikan data dan portabilitas aplikasi. Dapat dilihat pada Gambar 2.20



Gambar 2.20 Model Pengembangan *Hybrid Cloud*
(Sumber : Exigen Insurance Solutions, Inc, 2009)

4. *Community Cloud*

   Model ini diimplementasikan ketika beberapa perusahaan atau organisasi memiliki kesamaan konteks data. Contohnya adalah pemerintah Indonesia membuat suatu *Community Cloud* untuk aplikasi kependudukan disetiap daerahnya masing-masing dengan data yang sama. Dapat dilihat pada Gambar 2.21



Gambar 2.21 Model Pengembangan *Community Cloud*
(Sumber : Exigen Insurance Solutions, Inc, 2009)

2.8.7 Keuntungan *Cloud Computing*

Terdapat berbagai keuntungan yang dapat diperoleh dalam penggunaan teknologi *Cloud* Computing, diantaranya adalah :

1. Mengurangi biaya untuk merakit suatu komputer atau PC

   Pengguna tidak perlu mengeluarkan banyak biaya untuk merakit komputer yang nantinya mampu untuk menjalankan aplikasi yang

mereka butuhkan, karena segala bentuk proses, media penyimpanan akan dilakukan pada *cloud*.

2. Improvisasi *Perfomance*

   Cloud computing biasanya akan berusaha untuk menyesuaikan kinerja aplikasinya, jika kebutuhan tugas yang dilakukan semakin banyak maka, *cloud computing* akan berusaha untuk menyesuaikan bebean kerja tersebut.

3. Mengurangi biaya infrastruktur teknologi

   Ini akan terasa pada suatu perusahaan atau organisasi, karena investasi perusahaan untuk menggunakan infrastruktur teknologinya akan lebih hemat.

4. *Maintenance*

   Segala bentuk *maintenance* akan dilakukan di *cloud*, sehingga suatu perusahaan atau perorangan tidak perlu repot untuk *maintenance* komputasi, atau aplikasi.

5. Mengurangi biaya *software*

   Biasanya *software desktop* akan selalu ada yang lebih baru, dan itu harus dibeli kembali untuk menginstallnya, maka akan dibutuhkan lagi suatu biaya untuk membeli *software* tersebut. Jika ada *cloud* software akan merubah versinya tanpa perlu menginstalnya kembali.

6. *Update software* otomatis

   Software yang dipakai pengguna pada *cloud* biasanya akan terus melakukan *update* secara otomatis.

7. Meningkatkan kinerja komputer
8. *Storage* yang besar / tak terbatas

   Karena sudah tersimpan pada *cloud*, maka media penyimpanan yang kita miliki sangat besar bahkan tak terbatas dibandingkan dengan media penyimpanan yang ada pada *PC desktop* biasa.
9. Data lebih *safe*

   Safe ini berarti jika PC yang biasa dipakai mengalami suatu bencana, *crash hardware* maupun *software*, maka data yang ada pada PC tersebut tidak akan terselamatkan, sedangkan pada *cloud* dijamin keamanannya, karena data yang disimpan tidak hanya disimpan pada satu *server* saja.
10. Improvisasi kompatibilitas Sistem Operasi

    Sistem komputasi yang ada pada *cloud* tidak tergantung pada sistem operasi. Misalnya pengguna lebih sering menggunakan aplikasi berbasis Windows, sedangkan perusahaan di tempat ia bekerja menggunakan sistem operasi Linux. Hal ini mampu diatasi oleh *cloud computing*.
11. Improvisasi kompatibilitas format *file*

    Pada aplikasi berbasis web, *file* dokumen apapun bisa diakses, sebagai contoh, jika pengguna lebih sering menggunakan Microsoft Office 2007, sedangkan pada PC yang dipakai masih menggunakan Microsoft 2003, maka pengguna tidak bisa mengakses *file* tersebut. Maka aplikasi berbasis *web* akan mampu untuk menangani masalah tersebut.

12. Kolaborasi lebih mudah

    Cloud computing akan memudahkan dalam men-*share* atau membagi data-data sesuai keinginan pengguna

13. Akses *file* dimanapun dan kapanpun

    Pengguna tidak akan disulitkan atau tergantung pada tempat dan waktu untuk mengakses *file*, karena aplikasi berbasis web tidak akan pernah "mati".

14. *Latest Version Availability*

    Jika pengguna mengakses dan melakukan *update* disuatu tempat yang jauh dari biasanya, maka pada saat pengguna berada ditempat yang berbeda, *file* yang baru di*update* yang akan tampil.

2.8.8 Kerugian *Cloud Computing*

Suatu teknologi diciptakan, pasti ada keuntungan dan kerugian. Dan disinilah beberapa kerugian ataupun kendala yang masih mungkin terjadi pada *cloud computing* :

1. Membutuhkan Koneksi Internet

   Untuk mengakses aplikasi, dokumen atau *file* yang ada pada *cloud* sangat mustahil jika tidak dibantu menggunakan koneksi internet.

2. Tidak berjalan dengan baik jika koneksinya lambat

   Karena biasanya aplikasi yang dipakai butuh *bandwidth* yang sesuai untuk men*download* aplikasi yang dipakai.

3. Fitur yang terbatas

   Sekarang ini fitur-fitur pada *cloud computing* masih terbatas dibandingkan dengan aplikasi pada PC *desktop*. Contohnya adalah Google Docs, yang memiliki fitur yang umum atau sederhana dibandingkan dengan Microsoft Office yang sudah *full-featured*.

4. Penyimpanan data yang tidak aman

   Data yang disimpan pada *cloud*, tidak menjamin bahwa pengguna lain yang tidak berkewenangan dapat mengakses data yang telah disimpan.

5. *Service Level*

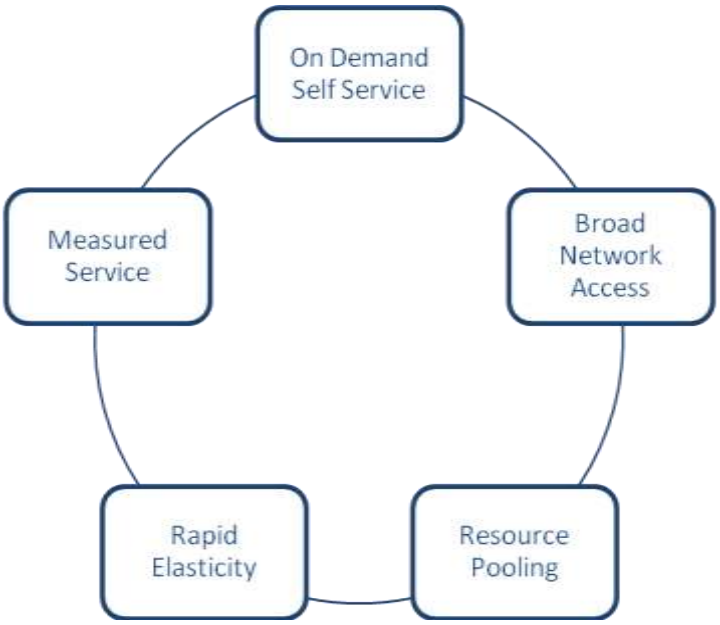
   Cloud Provider mungkin tidak akan konsisten dengan *performance* dari aplikasi atau transaksi, sehingga pengguna harus paham mengenai *service level* untuk *transaction response time*, *data protection*, kecepatan data *recovery*.

6. *Privacy*

   Karena pihak lain pun kemungkinan menggunakan *cloud provider* yang sama, maka ada kemungkinan juga data yang disimpan bisa dibaca oleh pihak lain tersebut.

2.8.9 Karakteristik *Cloud Computing*

Terdapat beberapa karakteristik dari *Aplikasi Cloud Computing*, yaitu:



Gambar 2.22 Karakteristik *Cloud Computing*
(Sumber : Exigen Insurance Solutions, Inc, 2009)

1. *On Demand Self Service*

   Aplikasi yang dibuat sesuai dengan kebutuhan para pengguna, sehingga pengguna tidak perlu untuk menggunakan aplikasi yang besar jika didalamnya tidak terdapat kebutuhan dari pengguna

2. *Measured Service*

   Layanan atau fiturnya terukur oleh para pengguna

3. *Broad Network Access*

   Aplikasi hanya bisa diakses via jaringan internet, sehingga pengguna harus terhubung ke internet untuk dapat menggunakan aplikasi *Cloud Computing*.

4. *Rapid Elasticity*

   Pengguna bisa dibebaskan untuk memilih aplikasi dan juga bisa berhenti menggunakan aplikasi bahkan pengguna pun bisa melakukan *upgrade* fitur kepada *cloud provider*.

5. *Resource Pooling*

   Data dan aplikasi bersifat terpusat, sehingga data dan aplikasi yang diakses oleh para pengguna, berada di lingkungan yang sama yaitu "*cloud*" / server yang telah disediakan oleh para *cloud provider*.

2.9 Hosting

Hosting merupakan jasa layanan ruang (space) untuk menyimpan data text, gambar, audio dan video yang selanjutnya file-file tersebut bias diakses dari mana saja (online). Hosting juga dilengkapi dengan fasilitas protocol seperti FTP dan HTTP untuk memudahkan dalam mengakses data, selain itu hosting juga bias digunakan untuk mengelola email.

Biasanya penyedia layanan hosting (hosting provider) menyediakan paket-paket tertentu yang disesuaikan dengan kebutuhan pelanggan, seperti quota *space* harddisk, quota bandwidth, bahasa pemograman, dan database.

2.9.1 Mengenal beberapa perbedaan jenis hosting

Pada implementasinya terdapat beberapa jenis hosting sesuai kegunaannya, diantaranya sebagai berikut :

2.91.1 *Shared Hosting*

Merupakan layanan hosting yang digunakan bersama-sama oleh pengguna lainnya. Para pengguna layanan ini menggunakan sumber daya baik *software* maupun *hardware* atau IP *Address* secara bersama-sama. Hosting seperti ini dipengaruhi oleh setiap proses yang dilakukan oleh pengguna, jika ada pengguna hosting yang melakukan proses yang berlebihan dan membebani server, maka pengguna lainnya akan terkena imbasnya, seperti server menjadi lambat bahkan ada kemungkinan website menjadi tidak bisa diakses.

Sebagai contoh jika server *Shared hosting* terkena *hack*, atau *hardware failure* maka seluruh pengguna *Shared hosting* akan terkena dampaknya seperti website tidak bias diakses. Pemilik server tentunya akan melakukan pamantauan secara bertahap dan rutin terhadap proses yang terjadi didalam server, biasanya admin akan melakukan *suspend account* terhadap pengguna yang melakukan pelanggaran atau mengganggu terhadap kinerja server.

Shared hosting cocok bagi Anda yang mempunyai blog, web presence, took online. Dari sisi harga jenis hosting ini lebih murah jika dibandingkan dengan jenis hosting lainnya, kembali kepada paket yang disediakan oleh hosting provider.

Kekurangan dari *Shared hosting* ini, pengguna tidak bias leluasa dalam menggunakan sumber daya server, atau menggunakan *software* sesuai dengan kebutuhan karena hak penuh terhadap administrasi server dipegang sepenuhnya oleh admin.

2.9.1.2 *Dedicated Hosting*

Jenis hosting ini biasa digunakan untuk solusi hosting terhadap kebutuhan yang berskala besar atau kompleks. Pengguna bias memilih atau menyediakan spesifikasi server yang diinginkan, yang kemudian ditempatkan pada sebuah *data center*. Pengguna bias menggunakan sumber daya server secara leluasa karena administrasi server dikelola atau dimiliki oleh pemilik server.

Dedicated server bias digunakan oleh penyedia layanan hosting (*Shared hosting* vps), *application service provider*, atau website yang memiliki trafik dan atau membutuhkan *resource* yang tinggi. Jika dilihat dari harga, pengelolaan *dedicated server* cukup mahal jika dibandingkan dengan *shared-hostinga* atau VPS.

2.9.1.3 *Virtual Private Server* (VPS)

Virtual Private Server (VPS) merupakan teknologi perangkat lunak yang memungkinkan pembagian sumber daya ke dalam beberapa virtual mesin (server). Tiap-tiap *virtual server* akan mempunyai sistem operasi yang berjalan secara mandiri. VPS menyediakan akses penuh terhadap user root, setiap VPS mempunyai konfigurasi yang bias diatur layaknya pada server fisik (*dedicated server*).

Kelebihan dari penggunaan VPS, antara lain :

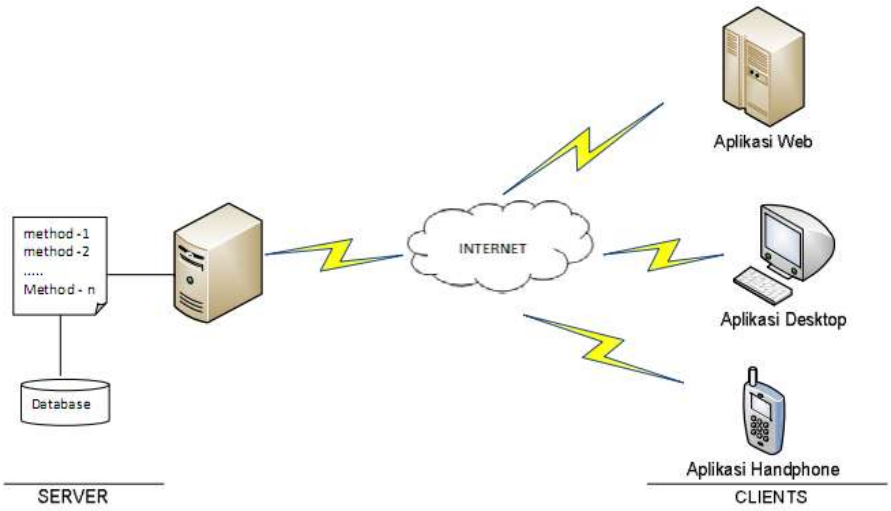
a. Performa yang baik seperti layaknya dedicated server
b. Kebebasan dalam menggunakan sistem operasi atau software yang dibutuhkan
c. Menghemat biaya, jika dibandingkan dengan menggunakan *dedicated server*.

VPS biasanya digunakan pada website atau aplikasi yang memerlukan *resource* medium diatas *shared hosting*. Layanan VPS memungkinkan jka pengguna memiliki website atau aplikasi yang memerlukan *resource* cukup tinggi dengan *budget* rendah.

2.10 *Web Service*

Web service adalah kumpulan dari fungsi atau *method* yang terdapat pada sebuah server yang dapat dipanggil oleh klien dari jarak jauh. Untuk memanggil *method-method* tersebut, kita bebas menggunakan aplikasi yang dibuat dengan bahasa pemrograman apa saja yang dijalankan diatas *platform* apa saja. Inilah keunggulan utama dari *web service*. Pada masa sekarang ini perangkat keras, sistem operasi, aplikasi, hingga bahasa pemrograman semakin beraneka ragam jenisnya. Keadaan tersebut dapat menimbulkan masalah dalam proses pertukaran data antar perangkat yang menggunakan aplikasi dan *platform* yang berbeda.

Untuk memecahkan permasalahan tersebut, diciptakanlah *web service*. Penggunaan *web service* memugkinkan perangkat-perangkat yang menggunakan sistem operasi dan aplikasi yang berbeda satu sama lain dapat saling bertukar data dan informasi dengan mudah. Dapat dilihat pada Gambar 2.23



Gambar 2.23 *Web Service* dapat diakses berbagai Jenis Aplikasi

Penyediaan service pada aplikasi web yang dibuat menjadi penting dalam mencapai kesuksesan. Ini memberikan pengguna kontrol yang lebih baik terhadap data yang disediakan situs dan membuka pintu kreatifitas guna-ulang dari data tersebut. Web service juga mempromosikan sebuah visi dari Web dimana situs mampu menspesialisasikan fokusnya dan bekerjasama dengan lebih efisien dalam mencapai tujuan pemrograman. Contohnya, sekarang pengembang Web lebih cenderung menggunakan salah satu peta (map) yang ditawarkan Google, Yahoo! Atau Microsoft ketimbang membuat sendiri petanya. Dalam area teknologi manapun, guna-ulang komponen dan kemampuan untuk spesialisasi merupakan faktor kunci dalam meraih

kemajuan. Penggunaan dan pengembangan *service* pada Web memungkinkan peningkatan kualitas dan pengalaman aplikasi web untuk semua pengguna.

2.10.1 URL yang Mengalamati Konsep, Bukan File

Dahulu, URL beroperasi pada lapisan abstraksi yang disediakan oleh *file system* fisik (file dan direktori), tetapi sekarang URL dioperasikan dalam satu lapisan abstraksi baru yang ditambahkan di atas aplikasi dan digunakan sebagai ruang yang dapat dialamati.

Dahulu, walaupun kode aplikasi dan strukturnya berarti bagi pengembang, tapi itu tidak berarti bagi pengguna. Sekarang dengan adanya lapisan abstraksi baru ini, maka URL bisa terlihat "cantik" dan berarti bagi pengguna juga. Ini misalnya terdapat pada Flickr, URL akan berbentuk seperti di bawah ini: **http://flickr.com/photos/deritaf**

URL ini bisa dibandingkan dengan URL pada masa dahulu, misalnya URL berikut ini yang menunjuk pada merchandise Dr.Seuss dari situs Amazon pada tanggal 2 Maret 2000 yang didapatkan dari Internet Archive: **http://s1.amazon.com/exec/varzea/search-handle-rl/ref=gw_m_col_7/?index=fixed-price%26rank=-price%26field-status=open%26field-browse=68457%26field-titledesc%3DDr.%20Seuss.**

Konsep baru URL ini dapat dijelaskan sebagai berikut:

a. URL dipandang sebagai alamat menuju ruang-konsep dalam aplikasi ketimbang sebagai alamat menuju kode di dalam aplikasi. URL ini mungkin merepresentasikan noun (*resource* yang diatur oleh aplikasi) dan verb (action pada resource-resource tersebut). URL merupakan

entiti yang berarti walau tanpa parameter apapun, tetapi parameter tetap dapat digunakan untuk mengklarifikasi dan menambahkan parameter tambahan terhadap request.

b. URL yang dipetakan menuju ruang-konsep aplikasi benar-benar direncanakan (engineered), sebagaimana halnya dengan interface objek pada bahasa pemrograman Java dan C++. Struktur dari URL harus mengikuti pola-pola yang mudah dibaca dan dimengerti. Pola-pola ini dituliskan dan dipaksakan sebagai bagian dari aplikasi web.

c. Kecuali konten statik, maka tidak terdapat hubungan antara URL dengan file di web server. URL mengalamati lokasi dalam ruang-konsep, di filesystem. Suatu langkah yang disebut "routing" akan terjadi manakala sebuah *web request* diterima yang akan mengambil alamat konseptual ini dan memutuskan bagian mana dari kode aplikasi yang akan digunakan untuk menjawab *request* tersebut.

2.10.2 Aplikasi Sebagai *Service*

Setiap kali suatu halaman pada situs dimuat, halaman tersebut merupakan merupakan respon terhadap suatu *request service*. Fakta bahwa halaman web tersebut direnderdalam HTML adalah hanya karena itu merupakan tipe respon dari *request service* tersebut. Informasi yang direpresentasikan oleh halaman web tersebut dapat juga direpresentasikan dengan baik dalam bentuk file teks, XML, RDF atau format lainnya yang dipilih.

Jika ide sebelumnya diikuti dimana URL seharusnya merepresentasikan konsep ketimbang file, maka ini berarti pengembang sudah mendefinisikan "antarmuka" bagi *service*, serangkaian konsep yang akan membuat website tersedia bagi penggunanya. Sekarang tinggal mengimplementaikan back-end yang mampu merespon terhadap request non-HTML pada endpoint URL yang sama.

2.10.3 Anatomi Pemanggilan *Web Service*

Adanya Routing URL, memungkinkan bagi API berbasis HTTP yang *user-friendly* dan mengizinkan URL untuk merepresentasikan endpoint virtual di dalam sebuah fungsionalitas aplikasi. *Endpoint-endpoint* ini saja tidak cukup untuk menspesifikasikan pemanggilan *web service* secara sempurna melainkan diperlukan empat komponen yang berbeda.

Empat komponen ini, seperti dideskripsikan pada tabel di bawah ini, terlihat hampir mirip dengan komponen-komponen pada pemanggilan metode tradisional, dengan beberapa kekecualian. Fungsi pemanggil dapat meminta tipe pengembalian, dan sebuah perintah HTTP diberikan bersamaan dengan metode pemanggilan sebagai potongan data yang akan memandu eksekusi metode tersebut.

Tabel 2.5 Komponen pada Pemanggilan *Web Service*

Komponen	Disediakan Oleh	Tujuan
Controller dan Action	Definisi *route*	Memilih sebuah kelas *controller* dan metode dalam kelas tersebut untuk dieksekusi sebagai respon terhadap *web request*

Komponen	Disediakan Oleh	Tujuan
Format Respon	*Header* HTTP atau definisi *route* (variable: format)	Menspesifikasikan format respon dari action yang tersedia
Parameter *Request*	*Form* data atau parameter yang di *encode* pada URL	Memberikan parameter tambahan untuk memenuhi *request* dasar
Perintah HTTP	*Request* HTTP	Menyatakan *request* yang Diinginkan, apakah itu mengambil data, memodifikasi data atau menghapus data

Ketika mendefinisikan route, ini merupakan konteks yang lebih luas dimana mereka akan saling sesuai. Website merupakan koleksi dari *endpoint-endpoint*, masing-masingnya didefinisikan dan diakses menggunakan empat komponen dalam tabel di bawah. Biasanya, endpoint-endpoint ini dilayani dalam HTML, tetapi karena klien dapat me-*request* format respon yang lain, maka *endpoint-endpoint* ini juga bertindak seperti layaknya programmatic API.

2.10.4 Kelebihan *Web Service*

1. Lintas *Platform*

   Penggunaan web service memungkinkan komputer-komputer yang berbeda sistem operasi dapat saling bertukar data. Sebagai contoh, jika akan mengakses sebuah web service, sistem operasi yang digunakan tidak perlu sama dengan sistem operasi yang digunakan oleh web service tersebut.

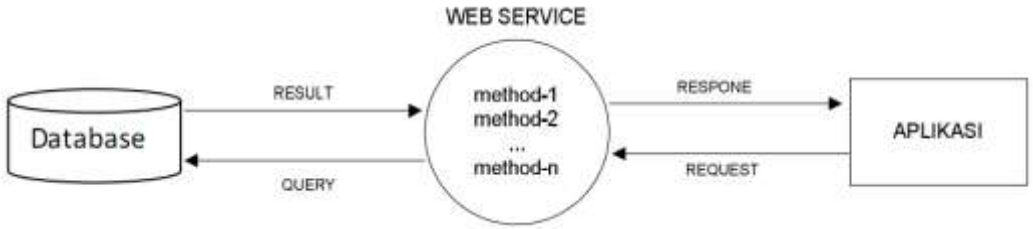
2. *Language Independent*

   Sebuah web service dapat diakses menggunakan bahasa pemrograman apa saja. Web service yang dibuat dengan PHP bisa diakses oleh berbagai bahasa pemrograman web seperti PHP itu sendiri, JSP hingga bahasa

lainnya seperti Delphi, Java, dan VB.NET. Web service juga memungkinkan untuk diakses oleh mobile device seperti handphone dengan memanfaatkan aplikasi berbasis Java untuk mobile device (Java 2 Micro Edition atau lebih dikenal dengan singkatan J2ME).

3. Jembatan Penghubung Dengan *Database*

Pada umumnya sebuah aplikasi memerlukan driver database agar bisa melakukan koneksi ke sebuah database. *Web service* dapat dijadikan sebagai "jembatan" penghubung antara aplikasi dengan database. Jadi dengan memanfaatkan *web service* sebagai "jembatan " penghubung, sebuah aplikasi tidak lagi memerlukan driver database dan tidak perlu mengetahui database apa yang digunakan oleh server serta bagaimana struktur database tersebut jika ingin mengaksesnya. Aplikasi tersebut cukup mengetahui *method* atau fungsi apa saja yang disediakan web service untuk memanfaatkan fasilitasnya. Dapat dilihat pada Gambar 2.24



Gambar 2.24 *Web Service* sebagai "Jembatan" Penghubung Aplikasi dengan database

4. Mempermudah Proses Pertukaran Data

Penggunaan web service dapat mempermudah dan mempercepat pertukaran data diantara mereka daripada mereka harus menyesuaikan aplikasi dan database yang digunakan yang tentu akan memakan waktu dan tenaga yang besar.

5. Pengguna Kembali Komponen Aplikasi

Beberapa aplikasi yang berbeda bisa saja memerlukan sebuah fungsi yang sama. Misalnya saja ada aplikasi web dan desktop yang memerlukan sebuah fungsi untuk melakukan konversi kurs antar mata uang yang berbeda.

2.10.4.1 Komponen Pendukung Web Service Extensible Markup Language (XML)

XML termasuk ke dalan keluarga Markup Language dan masih satu keluarga dengan HTML (Hypertext Markup Language) yang digunakan untuk membuat halaman web. Meskipun masih satu kelurga, kedua bahasa tersebut sebenarnya memilki fungsi yang berbeda. Jika fungsi HTML lebih diutamakan untuk mengatur tampilan, maka fungsi XML lebih ditujukan untuk menyimpan data dan informasi. Selain itu ada satu perbedaan lagi antara HTML dan XML. Di dalam XML kita bisa mendefinisikan tag milik kita sendiri. Contohnya sebagai berikut

```
<anggota>
        <nama> Seseorang</nama>
        <umur>25</umur>
</anggota>
```

Seperti terlihat pada contoh diatas, kita bisa mendefinisikan tag kita sendiri untuk menjelaskan sebuah struktur data bernama anggota yang memiliki atribut nama dan umur.

XML merupakan dasar yang penting atas terbentuknya *Web Service*. *Web service* dapat berkomunikasi dengan aplikasi-aplikasi yang memanggilkanya dengan menggunakn XML. Kenapa menggunakan XML? Karena XML berbentuk teks sehingga mudah untuk ditransportasikan menggunakan protokol HTTP. Selain itu XML juga bersifat platform independen sehingga informasi di dalamnya bisa baca oleh aplikasi apapun selama aplikasi tersebut mampu menerjemahkan tag-tag XML.

2.11 *Tool*s Yang Digunakan Untuk Membangun Aplikasi

Dalam membangun aplikasi diperlukan beberapa *tool*, yang dijelaskan pada sub bab–sub bab di bawah ini:

2.11.1 HTML (*Hypertext Markup Language)*

HTML (*Hypertext Markup Language*) adalah sebuah bahasa *markup* yang digunakan untuk membuat sebuah halaman web, menampilkan berbagai informasi di dalam sebuah penjelajah web internet dan *formatting hypertext* sederhana yang ditulis kedalam berkas format ASCII agar dapat menghasilkan tampilan wujud yang terintegerasi. Dengan kata lain, berkas yang dibuat dalam perangkat lunak pengolah kata dan disimpan kedalam format ASCII normal sehingga menjadi *home page* dengan perintah-perintah HTML. Bermula dari sebuah bahasa yang sebelumnya banyak digunakan di dunia penerbitan dan percetakan yang disebut

dengan SGML (*Standard Generalized Markup Language*), HTML adalah sebuah standar yang digunakan secara luas untuk menampilkan halaman web. HTML saat ini merupakan standar Internet yang didefinisikan dan dikendalikan penggunaannya oleh *World Wide Web Consortium* (W3C). HTML dibuat oleh kolaborasi Caillau TIM dengan Berners-lee Robert ketika mereka bekerja di CERN pada tahun 1989 (CERN adalah lembaga fisika energy tinggi di Jenewa).

HTML berupa kode-kode tag yang menginstruksikan penjelajah web untuk menghasilkan tampilan sesuai dengan yang diinginkan. Sebuah file yang merupakan file HTML dapat dibuka dengan menggunakan penjelajah web seperti Mozilla Firefox atau Microsoft Internet Explorer. HTML juga dapat dikenali oleh aplikasi pembuka surel ataupun PDA dan perangkat lunak lain yang memiliki kemampuan browser. Dengan menggunakan perintah-perintah HTML memungkinkan pengguna untuk melakukan tugas-tugas berikut :

1. Menentukan ukuran dan alur tulisan.
2. Mengintegrasikan gambar dengan tulisan.
3. Membuat pranala.
4. Mengintegrasikan berkas suara dan rekaman gambar hidup.

Kode HTML diawali dengan <html> dan diakhiri dengan </html>. Namun, sebelum <html> diharuskan untuk diberikan deklarasi !DOCTYPE. Fungsi deklarasi ini adalah sebagai *Public Text Identifier*. Beberapa hal penting dalam kode HTML :

1. Tanda <> menyatakan sebuah *tag*.
2. Pada umumnya *tag* berpasangan.

3. *Tag* yang tidak berpasangan antara lain adalah </br> dan <hr/>.
4. Pada *tag* yang berpasangan, *tag* yang berkedudukan sebagai *tag* penutup mempunyai bentuk </>.

2.11.2 PHP (*Personal Home Page*)

PHP (*Personal Home Page*), merupakan bahasa pemrograman *web* bersifat *serverside*. Hal tersebut mengandung arti bahwa PHP merupakan bahasa berbentuk *script* yang disimpan dan dijalankan di komputer *server* (*web server*) sedang hasilnya dikirimkan ke komputer *client* (*web browser*) dalam bentuk *script* HTML. Keuntungan penggunaan PHP, kode yang menyusun program tidak perlu dibagikan ke pemakai, yang berarti bahwa kerahasian kode dapat dilindungi.

Penggunaan PHP memungkinkan *web* dapat dibuat dinamis sehingga perawatan situs *web* tersebut menjadi lebih mudah dan efisien. Hal menarik yang didukung oleh PHP adalah kenyataan bahwa PHP bias digunakan untuk mengakses berbagai macam *database* seperti Accessm Oracle, MySql, dan lain-lain. PHP merupakan perangkat lunak *Open Source* yang disebarkan dan dilisensikan secara gratis serta dapat diunduh secara bebas dari situs resminya.

PHP diciptakan pertama kali oleh Rasmus Lerdorf pada tahun 1994. Awalnya, PHP digunakan untuk mencatat jumlah serta untuk mengetahui siapa saja pengunjung pada *homepage*-nya. Rasmus Lerdorf adalah salah satu pendukung *open source*. Oleh karena itu, ia mengeluarkan *Personal Home Page Tools* versi 1.0 secara gratis, kemudian menambah kemampuan PHP 1.o dan meluncurkan PHP 2.0.

Pada tahun 1996 PHP telah banyak digunakan dalam *website* di seluruh dunia. Sebuah kelompok pengembang perangkat lunak yang terdiri dari Rasmus, Zaew, Suraski, Andi Gutman, Stig Bakken, Shane Varaveo, dan Jim Winstead bekerja sama untuk menyempurnakan PHP 2.0. Akhirnya, pada tahun 1998, PHP 3.0 diluncurkan. Penyempurnaan terus dilakukan sehingga pada tahun 2000 dikeluarkan PHP 4.0, tidak berhenti sampai di situ, kemampuan PHP terus ditambah dan dikembangkan sehingga dilluncurkan PHP 5.0.

Fungsi yang dimiliki PHP sangat lengkap sehingga dapat dikatakan tidak perlu membuat fungsi sendiri. Hal tersebut dikarenakan daftar fungsi PHP yang lengkap menjadikan baris perintah semakin efisien. Selain itu juga, seperti yang telah disebutkan sebelumnya bahwa kelebihan PHP adalah mampu berintegrasi dengan berbagai macam *database.* Salah satu *database* yang paling umum diintegrasikan adalah MySql.

Adapun karakteristik *script* PHP dapat diuraikan sebagai berikut :

1. File PHP disimpan dengan ekstensi file seperti :*.php3,*.php4,*.php.
2. *Script* PHP biasanya diawali dengan *tag* <? Atau <?php dan ditutup dengan *tag* ?>.
3. File PHP dapat menginduk atau disisipkan pada bahasa *script* lainnya atau dapat berdiri sendiri.

2.11.3 *Javascript*

Javascript diperkenalkan pertama kali oleh *Netscape* pada tahun 1995. Pada awalnya bahasa ini dinamakan "*LiveScript*" yang berfungsi sebagai bahasa sederhana untuk *browser Netscape Navigator* 2. Pada masa itu bahasa ini banyak

di kritik karena kurang aman, pengembangannya yang terkesan buru-buru dan tidak ada pesan kesalahan yang ditampilkan setiap kali kita membuat kesalahan pada saat menyusun suatu program. Kemudian sejalan dengan sedang giatnya kerjasama antara *Netscape* dan *Sun* (pengembang bahasa pemrograman "*Java*") pada masa itu, maka *Netscape* memberikan nama "*JavaScript*" kepada bahasa tersebut pada tanggal 4 desember 1995. Pada saat yang bersamaan Microsoft sendiri mencoba untuk mengadaptasikan teknologi ini yang mereka sebut sebagai "*Jscript*" di *browser Internet Explorer 3*.

Javascript adalah bahasa yang berbentuk kumpulan skrip yang pada fungsinya berjalan pada suatu dokumen HTML, sepanjang sejarah internet bahasa ini adalah bahasa skrip pertama untuk web. Bahasa ini adalah bahasa pemrograman untuk memberikan kemampuan tambahan terhadap HTML dengan mengijinkan pengeksekusian perintah-perintah di sisi *user*, yang artinya di sisi browser bukan di sisi *server web*.

Javascript bergantung kepada *browser* (*navigator*) yang memanggil halaman web yang berisi skrip-skrip dari *Javascript* dan tentu saja terselip di dalam dokumen HTML. *Javascript* juga tidak memerlukan kompilator atau penterjemah khusus untuk menjalankannya (pada kenyataannya *kompilator Javascript* sendiri sudah termasuk di dalam *browser* tersebut). Lain halnya dengan bahasa "*Java*" (dengan nama *JavaScript* selalu di banding bandingkan) yang memerlukan kompilator khusus untuk menterjemahkannya di sisi *user*.

Javascript merupakan suatu bahasa yang perkembangannya lambat dibandingkan dengan *Java* yang berkembang sangat cepat. Di *Javascript* kita

tidak mungkin menyembunyikan kode skrip yang kita tulis, kode langsung ditulis di dalam dokumen HTML dan sangat mudah terlihat, sedangkan di *Java*, kode sudah berbentuk setengah terkompilasi (dalam bentuk *applet*) dan tidak mungkin terlihat dalam dokumen HTML, satu mesin virtual di sisi *user* yang bertanggung jawab untuk menterjemahkan program di dalam *applet* tersebut setiap kali halaman HTML yang memuat *applet* tersebut dipanggil oleh *browser*. Dibandingkan dengan *applet java* yang cukup lambat dibuka oleh *browser*, bias kita katakana bahwa *Javascript* cukup cepat dipanggil (di *load*) oleh *navigator*.

JavaScript sendiri merupakan bahasa yang mudah dipahami, dalam artian diperlukan *skill novice* atau dasar untuk mengerti bahasa ini, jika anda sudah terbiasa dan mengenal konsep bahasa pemrograman visual, maupun *Java* ataupun C, akan sangat mudah untuk memahami konsep *Javascript*.

JavaScript adalah bahasa yang "*case sensitive*" artinya membedakan penamaan variable dan fungsi yang menggunakan huruf besar dan huruf kecil, dan yang terakhir seperti bahasa *Java* dan C, setiap instruksi diakhiri dengan karakter titik koma (;).

2.11.4 CSS (*Cascading Style Sheet*)

CSS (*Cascading Style Sheet*) adalah bahasa *stylesheet* yang digunakan untuk mengatur tampilan suatu dokumen yang ditulis dalam bahasa *markup*.

Penggunaan yang paling umum dari CSS adalah memformat halaman *web* yang ditulis HTML dan XHTML. Walaupun demikian, bahasanya sendiri dapat digunakan untuk semua jenis dokumen XML termasuk SVG dan XVL. Spesifikasi CSS diatur oleh *world wide web consortium* (W3C).

CSS digunakan terutama untuk memisahkan antara isi dokumen (yang ditulis dengan HTML atau bahasa *markup* lainnya) dengan presentasi dokumen (yang ditulis dengan CSS). Pemisahan ini dapat meningkatkan aksesibilitasi isi, memberikan lebih banyak keleluasaan dan control terhadap tampilan, dan mengurangi kompleksitas serta pengulangan pada struktur isi.

CSS memungkinkan halaman yang sama untuk menampilkan cara yang berbeda, seperti melalui layar, cetak, suara (sewaktu dibacakan oleh *browser* basis suara atau pembaca layar), dan juga alat pembaca *braille*, halaman HTML atau XML yang sama juga dapat ditampilkan secara berbeda, baik dari segi gaya tampilan atau skema warna dengan menggunakan CSS.

2.11.5 *Apache Web Server*

Apache merupakan *web server* yang paling banyak digunakan di *internet*. *Web server* ini pertama kali didesain untuk sistem operasi lingkungan UNIX. Namun demikian, pada beberapa versi berikutnya *Apache* mengeluarkan versi yang dapat dijalankan di Windows NT.

Berdasarkan sejarahnya, *Apache* dimulai oleh veteran pengembang NCSA httpd (*National Center for Supercomputing Application*). Pada saat itu pengembangan NCSA httpd sebagai *web server* mengalami stagnansi. Rob McCool meninggalkan NCSA dan memulai sebuah proyek baru bersama para *webmaster* lainnya, dengan menambah *bug*, dan menambahkan fitur pada NCSA httpd. Mereka mengembangkan program ini lewat *mailing list*. Dengan berpijak pada httpd versi 1.3, tim *Apache* mengeluarkan rilis pertama kali secara resmi *Apache* versi 0.6.2. Tim inti pengembang *Apache* pada saat itu terdiri dari :

1. Brian Behlendorf
2. Roy T. Fielding
3. Rob Hartill
4. David Robinson
5. Cliff Skolnick
6. Randy Terbush
7. Robert S.Thau
8. Andrew Wilson

Dengan tambahan kontribusi dari:

1. Eric Hagberg
2. Frank Peters
3. Nicolas Pioch

Nama *Apache* diambil dari ungkapan "*A Patchy Server*", server perbaikan yang penuh dengan tambalan (*patch*). Tambalan yang dimaksud adalah penambahan fitur dan penambalan bug dari NCSA httpd versi 1.3.

Apache memiliki program pendukung yang cukup banyak. Hal ini memberikan layanan yang cukup lengkap bagi penggunanya. Beberapa dukungan tersebut diantaranya :

1. Kontrol Akses

   Kontrol ini dapat dijalankan berdasarkan nama *host* atau nomor IP.

2. CGI (*Common Gateway Interface*)

   CGI yang paling terkenal adalah Perl (*Practical Extraction and Report Language*), didukung oleh *Apache*.

3. PHP (*Personal Homepage/Hypertext Processor*)

   Apache mendukung PHP dengan menempatkannya sebagai salah satu modulnya (mod-php). Hal ini menjadikan kinerja PHP lebih baik.

4. SSI (*Server Side Includes*)

   Selain itu kelebihan lainnya dari *Apache* diantaranya yaitu *Apache* termasuk dalam kategori *frameware* dan *Apache* mudah dalam proses instalasi dibandingkan dengan *web browser* lainnya.

2.11.6 *MySQl Database Server*

MySQL dikembangkan sekitar tahun 1994 oleh sebuah perusahaan pengembang *software* dan konsultan *database* bernama MYSQL AB yang berada di Swedia. Waktu itu perusahaan tersebut masih bernama TcX DataKonsult AB, dan tujuan awal dikembangkannya *MySQL* adalah untuk mengembangkan aplikasi berbasis *web* pada *client*. Awalnya Michael "Monty" Widenius, pengembang satu-satunya di TcX memiliki sebuah aplikasi UNIREG dan rutin ISAM buatannya sendiri dan sedang mencari antarmuka *SQL* yang cocok untuk diimplementasikan ke dalamnya.

Mula-mula Monty memakai *miniSQL* (mSQL) pada eksperimennya itu, namun *SQL* dirasa kurang sesuai, karena terlalu lambat dalam pemrosesan *query*. Akhirnya Monty menghubungi David Hughes, pembuat mSQL yang sedang merilis versi kedua dari mSQL. Kemudian Monty mencoba membuat sendiri mesin *SQL* yang memiliki antarmuka mirip dengan *SQL*, tetapi dengan kemampuan yang lebih sesuai sehingga lahirlah *MySQL*. Tentang pengambilan nama *MySQL*, sampai saat ini masih belum jelas asal usulnya. Ada yang

berpendapat nama My diambil dari huruf depan dan belakang Monty, tetapi versi lain mengatakan nama itu diambil dari putrid Monty yang kebetulan juga bernama My.

Sebagai *database server* yang memiliki konsep *database modern*, MySQL memiliki banyak sekali keistimewaan. Berikut ini beberapa keistimewaan yang dimiliki oleh *MySQL* :

a. *Portability*

   MySQL dapat berjalan stabil pada berbagai OS seperti Windows, Linux, Unix, Mac OS, Solaris, Amiga, PHP-UX, Symbian.

b. *Open Source "Limited"*

   Dahulu *MySQL* didistribusikan secara *open source* (gratis), dibawah lisensi GPL sehingga kita dapat menggunakannya secara cuma-cuma tanpa dipungut biaya. Namun, saat ini karena MySQL telah dibeli oleh SUN maka kita tidak dapat lagi menikmati fitur-fitur baru yang ada di *MySQL*, karena SUN akan membatasi fitur-fitur baru ini hanya untuk *user* yang membeli lisensinya. Sehingga *MySQL* tidak lagi sebuah *opensource* yang benar-benar gratis lagi. *MySQL* sekarang hanya menyediakan fitur-fitur "dasar" saja yang saat ini sudah menggunakan versi 5.1.

c. *Multiuser*

   MySQL dapat digunakan oleh beberapa *user* dalam waktu yang bersamaan tanpa mengalami konflik. Hal ini memungkinkan sebuah *database server* MySQL dapat diakses *client* secara bersamaan.

d. *Performance Tuning*

MySQL memiliki kecepatan yang menakjubkan dalam menangani *query* sederhana, dengan kata lain dapat memproses lebih banyak *SQL* per satuan waktu.

e. *Column Types*

MySQL memiliki tiper kolom yang sangat kompleks, seperti *signed/unsigned integer, float, double, char, varchar, text, blob, date, time, datetime, timestamp, year, set* serta *enum.*

f. *Command and Functions*

MySql memiliki operator dan fungsi secara penuh yang mendukung perintah *SELECT* dan *WHERE* dalam *query*.

g. *Security*

MySQL memiliki beberapa lapisan sekuritas seperti level *subnetmask*, nama *host*, dan izin akses *user* dengan sistem perizinan yang mendetail serta *password* terenkripsi.

h. *Scalability* dan *Limits*

MySQL mampu menangani *database* dalam skala besar dengan jumlah *records* lebih dari 50 juta dan 60 ribu tabel serta 5 miliar baris. Selain itu, batas *index* yang dapat ditampung mencapai 32 indeks pada tiap tabelnya.

i. *Connectivity*

MySQl dapat melakukan koneksi dengan *client* menggunakan TCP/IP, Unix Soket (Unix), atau *Named Pipes* (NT).

j. *Localization*

MySQL dapat mendeteksi pesan kesalahan (error code) pada *client* dengan menggunakan lebih dari dua puluh bahasa.

k. *Interface*

MySQL memiliki *interface* terhadap berbagai aplikasi dan bahasa pemrograman dengan menggunakan fungsi API (*Application Programming Interface*).

l. *Clients and Tools*

MySQl dilengkapi dengan berbagai tool yang dapat digunakan untuk administarasi *database*, dan pada setiap *tool* yang ada disertakan petunjuk *online*.

m. Struktur Tabel

MySQL memiliki struktur tabel yang lebih fleksibel dalam menangani ALTER TABLE dibandingkan *database* lainnya.

2.12 Teknik Pengumpulan Data

Dalam penelitian dan pengujian dilakukan teknik pengumpulan data menggunakan kuesioner. Dari hasil kuesioner tersebut akan dilakukan perhitungan untuk mengambil kesimpulan terhadap penilaian penerapan sistem yang baru dan mengukur sikap dari pengguna aplikasi yang telah dibangun dengan menggunakan skala likert.

2.12.1 Kuesioner atau Angket

Sebagian besar penelitian umumnya menggunakan kuesioner sebagai metode yang dipilih untuk mengumpulkan data. Kuesioner atau angket merupakan suatu teknik pengumpulan data secara tidak langsung (peneliti tidak langsung bertanya jawab dengan responden). Kuesioner adalah suatu daftar yang berisi pertanyaan-pertanyaan untuk tujuan khusus yang memungkinkan penganalisis untuk mengumpulkan data mengenai sikap, keyakinan, perilaku, dan karakteristik dari orang yang berada di dalam organisasi serta pendapat dari responden yang dipilih. Responden mempunyai kebebasan untuk memberikan jawaban atau respon sesuai dengan persepsinya. Dalam membuat kuesioner, harus dilalui prosedur sebagai berikut:

1. Merumuskan tujuan yang akan dicapai dengan kuesioner.
2. Mengidentifikasikan variabel yang akan dijadikan sasaran kuesioner.
3. Menjabarkan setiap variabel menjadi sub-varibel yang lebih spesifik dan tunggal.
4. Menentukan jenis data yang akan dikumpulkan, sekaligusn untuk menentukan teknik analisisnya.

Ada dua jenis format kuesioner, yaitu

1. Format bebas (free format) berisi dengan pertanyaan-pertanyaan yang harus diisi oleh responden di tempat yang sudah disediakan.
2. Kuesioner format pasti (fixed format), kuesioner tipe ini mempunyai beberapa bentuk pertanyaan, yaitu:

a. Check-off Question, jenis dari pertanyaan-pertanyaan ini dibuat sehingga responden dapat memeriksa jawaban-jawaban yang sesuai.
b. Yes/No Question, jenis dari pertanyaan ini memungkinkan responden untuk menjawab "YA" atau "TIDAK".
c. Opinion/ Choice Question, jenis dari pertanyaan-pertanyaan ini memungkinkan responden untuk memberikan pendapatnya.

Kelebihan teknik kuesioner, yaitu

1. Kuesioner baik untuk sumber data yang banyak dan tersebar.
2. Responden tidak merasa terganggu, karena dapat mengisi kuesioner dengan memilih waktunya sendiri.
3. Kuesioner secara relatif lebih efisien untuk sumber data yang banyak.
4. Karena kuesioner baisanya tidak mencatumkan identitas responden, maka hasilnya dapat lebih objektif.

Kekurangan teknik kuesioner, yaitu

1. Kuesioner tidak menggaransi responden untuk menjawab pertanyaan dengan sepenuh hati.
2. Kuesioner cenderung tidak fleksibel, artinyaa pertanyaan yang harus dijawab terbatas yang dicantumkan di kuesioner saja, tidak dapat dikembangkan lagi sesuai dengan situasinya.
3. Pengumpulan sampel tidak dapat dilakukan secara bersama-sama dengan daftar pertanyaan.
4. Kuesioner yang lengkap sulit untuk dibuat.

2.12.2 Skala Likert

Skalla likert adalah suatu skala psikometrik yang umum digunakan dalam kuesioner, dan merupakan skala yang paling banyak digunakan dalam riset berupa survei. Nama skala ini diambil dari nama Rensis Likert, yang menerbitkan suatu laporan yang menjelaskan penggunaannya. Sewaktu menanggapi pertanyaan dalam skala Likert, responden menentukan tingkat persetujuan mereka terhadap suatu pernyataan dengan memilih salah satu dari pilihan yang tersedia. Biasanya disediakan lima pilihan skala dengan format seperti:

1. Sangat tidak setuju
2. Tidak setuju
3. Netral
4. Setuju
5. Sangat setuju

Selain pilihan dengan lima skala seperti contoh di atas, kadang digunakan juga skala dengan tujuh atau sembilan tingkat. Suatu studi empiris menemukan bahwa beberapa karakteristik statistik hasil kuesioner dengan berbagai jumlah pilihan tersebut ternyata sangat mirip.

Skala Likert merupakan metode skala bipolar yang mengukur baik tanggapan positif ataupun negatif terhadap suatu pernyataan. Empat skala pilihan juga kadang digunakan untuk kuesioner skala Likert yang memaksa orang memilih salah satu kutub karena pilihan "netral" tak tersedia.