BAB II

LANDASAN TEORI

2.1 Konsep Dasar Sistem

Pada subbab ini akan diuraikan teori yang berkaitan dengan konsep dasar sistem, mulai dari pengertian sistem itu sendiri hingga contoh-contoh sistem.

2.1.1 Pengertian Sistem

Definisi sistem dapat dilihat dari dua pendekatan, yaitu pendekatan prosedur dan pendekatan komponen/elemen.

1. Pendekatan prosedur

   Sistem adalah suatu jaringan kerja dari prosedur-prosedur yang saling berhubungan, berkumpul, bersama-sama untuk melakukan suatu kegiatan atau untuk menyelesaikan suatu sasaran tertentu.

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

2. Pendekatan komponen/elemen

   Sistem adalah kumpulan dari elemen-elemen yang berinteraksi untuk mencapai tujuan tertentu.

2.1.2 Karakteristik Sistem

Menurut JOG[3], sistem mempunyai karakteristik atau sifat tertentu, yaitu komponen sistem *(system component)* , batasan sistem (*system boundary*),

lingkungan luar sistem (*environtments*), penghubung(*interface*), masukan (*input)*, keluaran(*output)*, pengolah (*process)*, dan sasaran (*objectives*) atau tujuan (*goal*) :

1. Komponen Sistem

   Suatu sistem terdiri dari sejumlah komponen yang saling berinteraksi yang artinya saling bekerja sama dalam membentuk satu kesatuan komponen-komponen sistem atau elemen-elemen sistem, dapat berupa subsistem atau bagian-bagian dari sistem. Sistem selalu mengandung subsistem. Dimana tiap subsistem yang ada mempunyai fungsi tertentu dan mempengaruhi kerja sistem tersebut.

2. Batasan Sistem

   Merupakan daerah yang membatasi antara suatu sistem dengan sistem lainnya atau dengan lingkungan luarnya. Dengan adanya batasan sistem ini memungkinkan suatu sistem dipandang sebagai suatu kesatuan.

3. Lingkungan Luar Sistem

   Lingkungan luar sistem adalah segala sesuatu diluar ruang lingkup sistem, namun mempengaruhi kinerja sistem.

4. Penghubung

   Merupakan media penghubung antara subsistem yang satu dengan subsistem lainnya. Melalui media ini, sumber data sistem mengalir ke satu subsistem ke subsistem lainnya, baik itu sumber data masukan maupun keluaran.

5. Masukan

   Masukan suatu sistem dapat berupa masukan perawatan ataupun masukan sinyal. Masukan perawatan *(maintenance input)* adalah masukan sistem yang digunakan untuk operasi sistem tersebut, sedangkan masukan sinyal *(signal input)* adalah masukan sistem yang diproses untuk menghasilkan keluaran *(output)*.

6. Keluaran

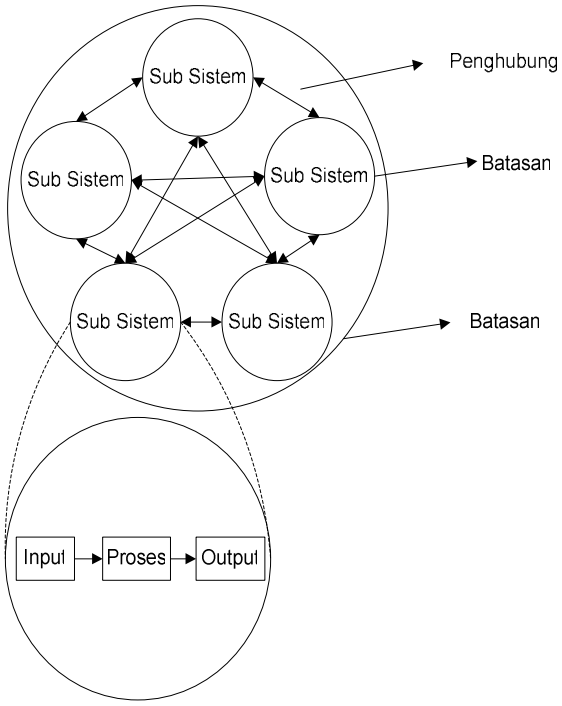
   Keluaran merupakan hasil pengolahan masukan pada suatu sistem maupun subsistem. Keluaran suatu subsistem dapat menjadi masukan bagi subsistem lainnya.

7. Pengolah

   Pengolah mempunyai fungsi untuk mengolah masukan menjadi keluaran pada suatu sistem.

8. Sasaran atau tujuan

   Suatu sistem mempunyai tujuan atau sasaran. Jika suatu sistem tidak mempunyai tujuan, maka operasi sistem tidak berguna. Sasaran sistem berguna untuk menentukan segala masukan dan keluaran dari suatu sistem. Suatu sistem dikatakan berhasil jika sasaran sistem tercapai.



Gambar 2.1 Karakteristik Sistem

2.1.3 Klasifikasi Sistem

Menurut JOG[3], klasifikasi sistem terbagi menjadi delapan macam, yaitu:

1. Sistem Abstrak

   Merupakan jenis sistem berupa pemikiran atau ide-ide. Contoh : Sistem Agama, Sistem Adat.

2. Sistem Fisik

   Merupakan sistem yang ada secara fisik. Contoh : Sistem Komputer, Sistem Akuntansi.

3. Sistem Alamiah

   Merupakan sistem yang terjadi melalui proses alamiah. Contoh : Sistem Perputaran Bumi, Sistem Gravitasi.

4. Sistem Buatan

   Merupakan sistem yang melibatkan interaksi manusia dalam cara kerjanya. Contoh : Sistem Komputer, Sistem Informasi.

5. Sistem Tertentu

   Merupakan sistem yang beroperasi dengan tingkah laku yang sudah dapat diprediksi. Contoh : Sistem Komputer

6. Sistem Tak Tertentu

   Merupakan sistem yang kondisi masa depannya tidak dapat diprediksi. Contoh : Sistem Penjualan Produk

7. Sistem Tertutup

   Merupakan sistem yang tidak berhubungan dan tidak terpengaruh dengan lingkungan luar.

8. Sistem Terbuka

   Merupakan sistem yang berhubungan dan ada pengaruhnya dengan lingkungan luar.

2.2 Konsep Dasar Data dan Informasi

Pada subbab ini akan diuraikan teori mengenai konsep data dan informasi

2.2.1 Data

Berdasarkan JOG[3], data adalah kumpulan dari fakta-fakta, kejadian-kejadian yang dapat berupa simbol, angka, huruf, dan lain-lain yang berguna bagi suatu pengolahan data (*process*) atau sebagai masukan (*input*) bagi suatu proses. Data merupakan deskripsi dari sesuatu dan kejadian yang kita hadapi.

2.2.2 Informasi

Menurut JOG[3], informasi merupakan hasil pengolahan dari sebuah model, formasi, organisasi, ataupun suatu perubahan bentuk dari data yang memiliki nilai tertentu, dan bisa digunakan untuk menambah pengetahuan bagi yang menerimanya. Dalam hal ini, data bisa dianggap sebagai obyek dan informasi adalah suatu subyek yang bermanfaat bagi penerimanya. Informasi juga bisa disebut sebagai hasil pengolahan atau pemrosesan data.

Sumber dari informasi adalah data. Data adalah kenyataan yang menggambarkan suatu kejadian-kejadian dan kesatuan nyata, tetapi data merupakan bentuk yang masih mentah dan belum dapat memberikan arti banyak bagi pemakai, sehingga perlu diolah lebih lanjut untuk menghasilkan informasi yang dibutuhkan. Dengan kata lain informasi adalah hasil dari pengolahan data.

2.2.2.1 Kualitas Informasi

Kualitas informasi sangat dipengaruhi atau ditentukan 3 hal, yaitu :

1. Relevan (*relevancy*)

   Relevan berarti informasi harus memberikan manfaat bagi pemakainya. Relevansi informasi berbeda-beda untuk tiap-tiap orang satu dengan yang lainnya.

2. Akurat (*accuracy*)

   Informasi harus bebas dari kesalahan-kesalahan dan tidak bisa atau menyesatkan, dan harus jelas mencerminkan maksudnya. Ketidakakuratan dapat terjadi karena sumber informasi (data) mengalami gangguan atau kesengajaan sehingga merusak atau merubah data-data asli tersebut.

Komponen akurat : lengkap (*completeness*), benar (*correctness*), aman (*security*).

3. Tepat waktu (*timeliness*)

   Informasi yang dihasilkan atau dibutuhkan tidak boleh terlambat (usang). Informasi yang using tidak mempunyai nilai yang baik, sehingga kalau digunakan sebagai dasar dalam pengambilan keputusan akan berakibat fatal atau kesalahan dalam keputusan dan tindakan. Kondisi demikian menyebabkan mahalnya nilai suatu infomasi, sehingga kecepatan untuk mendapatkan, mengolah, dan mengirimkannya memerlukan teknologi-teknologi terbaru.

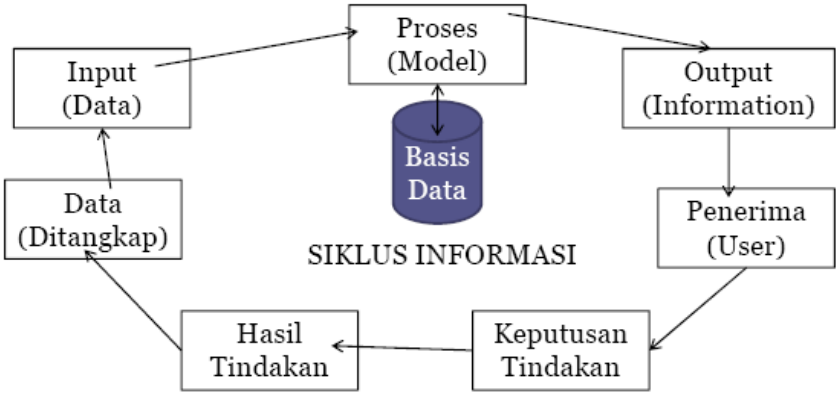
2.2.2.2 Nilai Informasi

Nilai dari informasi *(value of information)* ditentukan dari dua hal, yaitu manfaat dan biaya mendapatkannya. Suatu informasi dikatakan bernilai bila manfaatnya lebih efektif dibandingkan dengan biaya mendapatkannya. Kegunaan informasi adalah untuk mengurangi hal ketidakpastian di dalam proses pengambilan keputusan tentang suatu keadaan.

2.2.2.3 Siklus Informasi

Data agar menjadi lebih berarti dan berguna dalam bentuk informasi, maka perlu diolah melalui suatu model tertentu. Data yang telah diolah tersebut kemudian diterima oleh penerima, lalu penerima membuat suatu keputusan dan melakukan tindakan, yang berarti menghasilkan suatu tindakan yang lain yang akan membuat sejumlah data kembali. Data tersebut akan ditangkap sebagai *input*, dan diproses kembali lewat suatu model dan seterusnya sehingga membentuk

suatu siklus. Siklus ini disebut dengan siklus informasi (*information cycle*) atau disebut pula siklus pengolahan data (*processing cycles*).



Gambar 2.2 Siklus Informasi

2.3 Konsep Dasar Sistem Informasi

Sistem informasi dapat didefinisikan sebagai suatu sistem yang dibuat oleh manusia yang terdiri dari komponen-komponen dalam organisasi untuk mencapai suatu tujuan yaitu menyajikan informasi. Pengertian lain dari sistem informasi adalah sekumpulan prosedur organisasi yang pada saat dilaksanakan dan memberikan informasi bagi pengambil keputusan dan/atau untuk mengendalikan informasi.

2.3.1 Komponen Sistem Informasi

Dalam membangun suatu sistem informasi diperlukan penggabungan elemen-elemen pendukung tersebut antara lain :

1. *Software*, merupakan suatu program komputer, struktur data, dan dokumen-dokumen yang saling berhubungan yang digunakan dalam metode logika dan prosedur yang dibutuhkan.

2. *Hardware*, merupakan perangkat elektronik yang memiliki kemampuan untuk melakukan proses komputerisasi.
3. *User*, adalah pengguna dan operator perangkat keras atau perangkat lunak.
4. Data, berupa salinan-salinan manual dan deskripsi informasi yang menggambarkan operasi sistem.

2.3.2 Tujuan Pembangunan Sistem Informasi

1. Integrasi sistem
   a. Menghubungkan sistem individu/kelompok
   b. Pengkolektifan data dan penyambungan secara otomatis
   c. Peningkatan koordinasi dan pencapaian sinergi
2. Efisiensi pengelolaan sistem
   a. Penggunaan basis data dalam upaya kesamaan pengadministrasian data
   b. Pengelolaan data berkaitan dengan karakteristik informasi
   c. Penggunaan dan pengambilan informasi
3. Dukungan keputusan untuk manajemen
   a. Melengkapi informasi guna kebutuhan proses pengambilan kebutuhan
   b. Akuisisi informasi eksternal melalui jaringan komunikasi
   c. Ekstraksi dari informasi internal yang terpadu.

2.3.3 Manfaat Sistem Informasi

Sistem informasi memiliki banyak manfaat, diantaranya :

a. Menghemat tenaga kerja
b. Peningkatan efisiensi
c. Mempercepat proses

d. Perbaikan dokumentasi

e. Pencapaian standar

f. Perbaikan keputusan

2.4 Konsep Rekayasa Perangkat Lunak

2.4.1 Pengertian Rekayasa Perangkat Lunak

Berdasarkan PRE[5], rekayasa perangkat lunak adalah aplikasi dari sebuah pendekatan kuantifiabel, disiplin, dan sistematis kepada pengembangan, operasi, dan pemeliharaan perangkat lunak. Usaha yang berhubungan dengan rekayasa perangkat lunak dapat dikategorikan ke dalam tiga fase umum dengan tanpa mempedulikan area aplikasi, ukuran proyek, atau kompleksitasnya, yaitu :

1. Fase Definisi (*Definition Phase*)

   Fase ini berfokus pada "apa" (*what*); dimana, pada definisi ini pengembang perangkat lunak harus mengidentifikasi informasi apa yang akan diproses, fungsi dan unjuk kerja apa yang dibutuhkan, tingkah laku sistem seperti apa yang diharapkan, antarmuka apa yang akan dibangun, batasan perancangan serta kriteria validasi untuk mendefinisikan sistem yang sukses. Tugas teknis yang harus selalu ada dalam fase ini yaitu rekayasa sistem atau informasi, perencanaan proyek perangkat lunak, serta analisis kebutuhan.

2. Fase Pengembangan (*Development Phase*)

   Fase ini berfokus pada "bagaimana" (*how*), yaitu dimana selama masa pengembangan perangkat lunak, teknisi harus mendefinisikan bagaimana data dikonstruksikan, bagaimana fungsi-fungsi diimplementasikan sebagai

sebuah arsitektur perangkat lunak, bagaimana detail prosedur akan diimplementasikan, bagaimana antarmuka dikarakterisasi, bagaimana rancangan akan diterjemahkan ke dalam bahasa pemrograman serta bagaimana pengujian akan dilakukan. Tugas teknis yang harus selalu ada dalam fase ini yaitu rancangan perangkat lunak, pemunculan kode, dan pengujian perangkat lunak.

3. Fase Pemeliharaan (*Maintenance Phase*)

   Fase ini berfokus pada "perubahan" (*change*), yang dihubungkan dengan koreksi kesalahan, penyesuaian yang dibutuhkan ketika lingkungan perangkat lunak berkembang, serta perubahan kebutuhan pelanggan. Fase ini mengaplikasikan kembali langkah-langkah pada fase definisi dan pengembangan namun semuanya tetap bergantung pada konteks perangkat lunak yang ada.

Untuk menyelesaikan masalah aktual di dalam sebuah setting industri, rekayasa perangkat lunak atau tim perekayasa harus menggabungkan strategi pengembangan yang melingkupi lapisan proses, metode, dan alat-alat bantu serta fase-fase generik. Strategi ini sering diacukan sebagai model proses atau paradigma rekayasa perangkat lunak. Model proses untuk rekayasa perangkat lunak dipilih berdasarkan sifat aplikasi dan proyeknya, metode dan alat-alat bantu yang akan dipakai, dan kontrol penyampaian yang dibutuhkan.

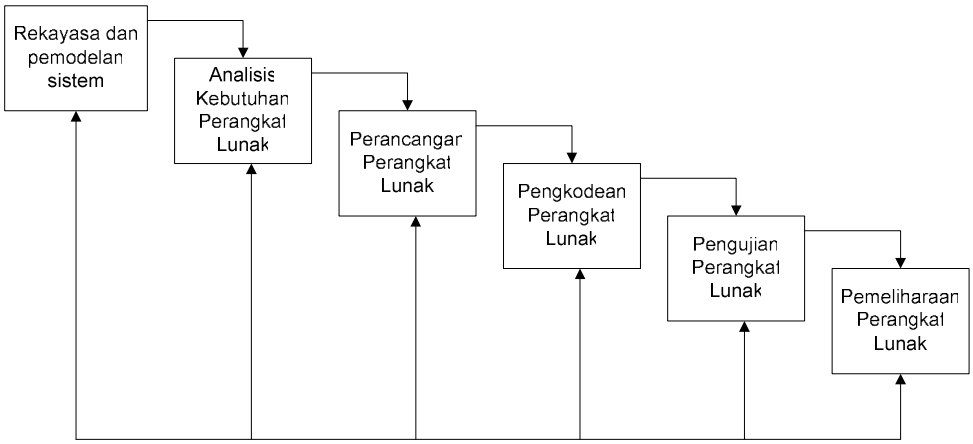
Model proses untuk rekayasa perangkat lunak dibagi menjadi empat model yaitu :

1. Model Sekuensial Linear
2. Model Prototipe
3. Model RAD
4. Model Proses Perangkat Lunak Evolusioner

2.4.2 Model Sekuensial Linear (*Waterfall*)

Model ini mengusulkan sebuah pendekatan kepada perkembangan perangkat lunak yang sistematik dan sekuensial yang mulai pada tingkat dan kemajuan sistem pada seluruh analisis, desain, kode, pengujian dan pemeliharaan.

Model waterfall dapat digambarkan sebagai berikut :



Gambar 2.3 Diagram *Waterfall*

1. Rekayasa dan Pemodelan Sistem

   Merupakan langkah awal dari rekayasa perangkat lunak, yang dimulai dari membangun syarat dari semua elemen sistem dan mengalokasikan beberapa bagian dari kebutuhan ke perangkat lunak tersebut.

2. Analisis Kebutuhan Perangkat Lunak

   Tahap ini digunakan untuk menjelaskan permasalahan yang ada dan menganalisa proses-proses yang akan dilakukan dalam perangkat lunak yang akan dibuat. Dalam tahap ini dilakukan pengumpulan kebutuhan perangkat lunak secara intensif.

3. Perancangan Perangkat Lunak

   Merupakan proses penerjemahan kebutuhan sistem ke dalam representasi perangkat lunak yang dapat diperkirakan sebelum dimulainya pemunculan kode perangkat lunak (pengkodean). Proses ini berfokus perancangan pada struktur data, arsitektur program, representasi *interface* dan detail (algoritma) prosedural.

4. Pengkodean Perangkat Lunak

   Pengkodean merupakan suatu kegiatan mengolah hasil rancangan ke dalam bentuk bahasa pemrograman (kode-kode pemrograman) agar dapat dibaca mesin. Pengkodean dapat diselesaikan secara mekanis jika perancangan dilakukan dengan lengkap.

5. Pengujian Perangkat Lunak

   Pengujian perangkat lunak atau program merupakan proses pengujian suatu program yang dilakukan secara terintegrasi maupun secara per-unit, untuk mengetahui kesalahan yang terjadi pada program yang sedang berjalan, dan memastikan bahwa input yang dibatasi akan memberikan hasil aktual yang sesuai dengan yang dibutuhkan.

6. Pemeliharaan Perangkat Lunak

   Perangkat lunak yang telah diuji dan berhasil, maka akan digunakan dan dipelihara. Pemeliharaan terjadi jika terdapat perubahan kebutuhan perangkat lunak. Pemeliharaan perangkat lunak mengaplikasikan lagi setiap fase program sebelumnya dan tidak membuat aplikasi baru lagi.

2.4.3 Model Prototipe (*Prototyping*)

Merupakan suatu teknik untuk mengumpulkan informasi tertentu mengenai kebutuhan-kebutuhan informasi pengguna secara cepat.

Tahapan yang dilakukan :

1. Reaksi awal dari pengguna, diawali dengan menampilkan sebuah protipe sistem informasi, kemudian melihat reaksi dari pengguna saat bekerja dengan prototipe apakah fitur-fitur sistem pada prototipe tersebut sudah sesuai dengan kebutuhannya. Reaksi tersebut dikumpulkan dalam lembar observasi, wawancara dan kuesioner.
2. Saran-saran pengguna, saran-saran merupakan hasil interaksi pengguna dengan prototipe yang ditampilkan (evaluasi pengguna) yang merupakan masukan untuk perbaikan, pengubahan atau ‘menghentikan’ prototipe sehingga dapat memenuhi kebutuhan pengguan dengan lebih baik.
3. Inovasi, adalah kemampuan-kemampuan sistem baru yang sebelumnya tidak ada pada saat pengguna berinteraksi dengan prototipe. Inovasi prototipe jika berhasil akan menjadi bagian dari sistem hasil jadi.

4. Rencana revisi, prototipe menggambarkan sistem di masa datang. Rencana revisi membantu mengidentifikasikan prioritas-prioritas apa saja yang akan diprototipekan selanjutnya.

2.4.4 Model RAD (*Rapid Application Development*)

Rapid Application Development (RAD) adalah sebuah model proses perkembangan perangkat lunak sekuensial linier yang menekankan siklus perkembangan yang sangat pendek. Model RAD ini merupakan sebuah adaptasi "kecepatan tinggi" dari model sekuensial linier dimana perkembangan cepat dicapai dengan menggunakan pendekatan konstruksi berbasis komponen. RAD menekankan perkembangan komponen program yang bisa digunakan kembali (reusabilitas).

1. Pemodelan Bisnis (*Business Modelling*)

   Aliran informasi di antara fungsi-fungsi bisnis dimodelkan untuk mengetahui informasi yang mengendalikan proses bisnis, informasi yang dimunculkan, pelaku yang memunculkan informasi, tujuan informasi, dan siapa saja yang memproses informasi tersebut.

2. Pemodelan Data (*Data Modelling*)

   Aliran informasi yang didefinisikan sebagai bagian dari fase pemodelan bisnis disaring ke dalam serangkaian objek data yang dibutuhkan untuk menopang bisnis tersebut. Karakteristik masing-masing objek diidentifikasi dan hubungan antara objek-objek tersebut didefinsikan.

3. Pemodelan Proses (*Process Modelling*)

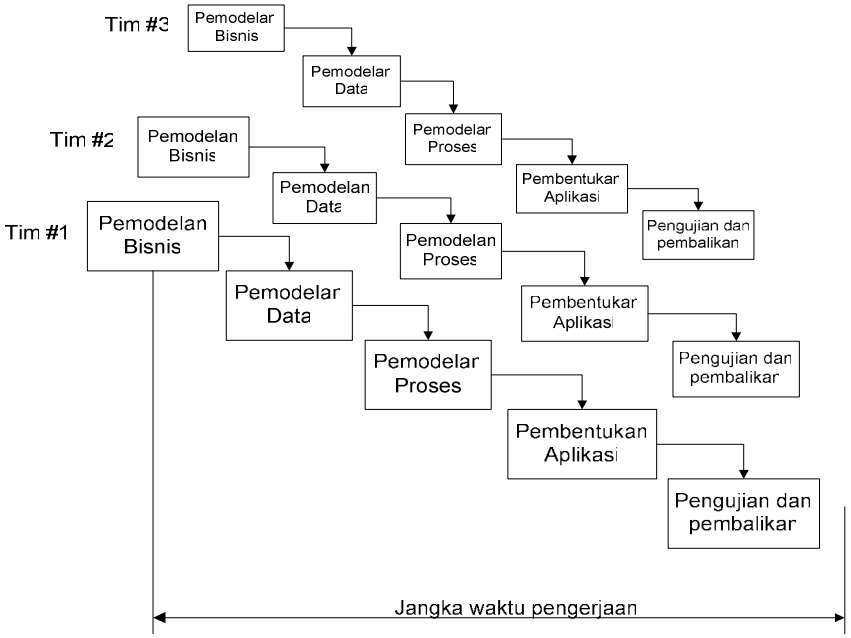
   Aliran informasi yang didefinisikan di dalam fase pemodelan data ditransformasikan untuk mencapai aliran informasi yang perlu bagi implementasi sebuah fungsi bisnis. Gambaran pemrosesan diciptakan untuk menambah, memodifikasi, menghapus, atau mendapatkan kembali sebuah objek data.

4. Penggunaan Generasi (*Application Generation*)

   RAD mengasumsikan pemakaian teknik generasi keempat. Selain menciptakan perangkat lunak dengan menggunakan bahasa pemrograman generasi ketiga yang konvensional, RAD lebih banyak memproses kerja untuk memakai lagi komponen program yang ada (pada saat memungkinkan) atau menciptakan komponen yang bisa digunakan lagi (bila perlu).

5. Pengujian dan Pembalikan (*Testing and Turnover*)

   Karena proses RAD menekankan pada pemakaian kembali, banyak komponen program telah diuji. Hal ini mengurangi keseluruhan waktu pengujian. Tetapi komponen baru harus diuji dan semua antarmuka harus dilatih secara penuh. RAD digambarkan sebagai berikut :



Gambar 2.4 Model RAD

2.4.5 Model Proses Perangkat Lunak Evolusioner (*Evolutionary Process Model*)

Merupakan model iteratif, yang ditandai dengan tingkah laku yang memungkinkan perekayasa perangkat lunak mengembangkan versi perangkat lunak yang lebih lengkap sedikit demi sedikit. Jenis model ini meliputi :

2.4.5.1 Model Pertambahan (*Incremental Process Model*)

Model ini menggabungkan elemen-elemen model sekuensial linear (diaplikasikan secara berulang-ulang) dengan filosofi prototipe iteratif, dimana setiap urutan linier menghasilkan pertambahan perangkat lunak "yang bisa disampaikan".

2.4.5.2 Model Spiral (Spiral Model)

Merupakan model proses perangkat lunak yang evolusioner yang merangkai sifat iteratif dari prototipe dengan cara kontrol dan aspek sistematis dari model sekuensial linear. Model ini mempunyai enam tugas yaitu :

a. Komunikasi Pelanggan

   Merupakan tugas-tugas yang dibutuhkan untuk membangun komunikasi yang efektif di antara pengembang dan pelanggan.

b. Perencanaan

   Merupakan tugas-tugas yang dibutuhkan untuk mendefisikan sumber-sumber daya, ketepatan waktu, dan proyek informasi yang berhubungan.

c. Analisis Resiko

   Merupakan tugas-tugas yang dibutuhkan untuk menaksir resiko-resiko, baik manajemen maupun teknis

d. Perekayasaan

   Merupakan tugas-tugas yang dibutuhkan untuk membangun satu atau lebih representasi dari aplikasi tersebut.

e. Konstruksi dan peluncuran

   Merupakan tugas-tugas yang dibutuhkan untuk mengkonstruksi, menguji,memasang (*install*) dan memberikan pelayanan kepada pemakai (contohnya pelatihan dan dokumentasi)

f. Evaluasi Pelanggan

Merupakan tugas-tugas yang dibutuhkan untuk memperoleh umpan balik dari pelanggan dengan didasarkan pada evaluasi representasi perangkat lunak yang dibuat selama masa perekayasaan, dan diimplementasikan selama masa pemasangan.

2.4.5.3 Model Rakitan Komponen (*Component-Based Development*)

Model Rakitan Komponen menggabungkan beberapa karakteristik model spiral. Model ini bersifat revolusioner, sehingga membutuhkan pendekatan iteratif untuk menciptakan perangkat lunak. Tetapi model rakitan komponen merangkai aplikasi dari komponen perangkat lunak sebelum dipaketkan (kadang disebut kelas).

2.4.5.4 Model Perkembangan Konkuren (*The Concurrent Development Models*)

Model ini disebut juga rekayasa konkuren. Model ini digunakan sebagai paradigma bagi pengembangan aplikasi client/server. Bila diaplikasikan kepada client/server, model proses ini akan mendefinisikan aktifitas ke dalam dua dimensi yaitu dimensi sistem dan dimensi komponen.

2.4.6 Model Formal (*The Formal Methods Model*)

Model metode formal mencakup sekumpulan aktifitas yang membawa kepada spesifikasi matematis perangkat lunak komputer. Metode formal memungkinkan perekayasa perangkat lunak untuk mengkhususkan, mengembangkan, dan memverifikasikan sistem berbasis komputer dengan menggunakan notasi matematis yang tepat.

2.4.7 Teknik Generasi Keempat (4GT)

Metode ini dimulai dengan pengumpulan kebutuhan, strategi perancangan, implementasi menggunakan 4GL dan pengujian. Untuk aplikasi yang kecil dimungkinkan dari pengumpulan kebutuhan langsung mengimplementasikannya menggunakan 4GL, tetapi untuk aplikasi yang besar perlu adanya pengembangan strategi perancangan.

2.5 Konsep Perancangan Sistem

Perancangan sistem secara umum adalah suatu tahap dimana di dalamnya terdapat identifikasi komponen-komponen sistem informasi yang akan dirancang secara rinci yang bertujuan untuk memberikan gambaran kepada pengguna atau *user* mengenai sistem yang baru. Sedangkan desain sistem secara terinci dimaksudkan untuk pembuat program komputer dan ahli teknik lainnya yang akan mengimplementasikan sistem.

Penggambaran dan rancangan model sistem informasi secara logika dapat dibuat dalam bentuk Diagram Konteks dan Diagram Alir Data (DAD) atau *Data Flow Diagram* (DFD).

2.5.1 Diagram Konteks

Diagram Konteks adalah diagram tingkat tinggi dari Diagram Alir Data yang merupakan gambaran global dari sistem informasi yang menggambarkan aliran-aliran data ke dalam maupun keluar suatu sistem dan merupakan alat yang digunakan untuk melihat batasan antara sistem dengan *eksternal entity* .

2.5.2 Diagram Alir Data

Diagram Alir Data atau *Data Flow Diagram* (DFD) adalah suatu model yang menjelaskan arus data mulai dari pemasukan sampai dengan keluaran data. Tingkatan DFD dimulai dari diagram konteks yang menjelaskan secara umum suatu sistem atau batasan sistem aplikasi yang akan dikembangkan. Kemudian DFD dikembangkan menjadi DFD tingkat 1 atau level 1 dan kemudian DFD level 1 dikembangkan lagi menjadi level 2 dan selanjutnya sampai sistem tersebut tergambarkan secara rinci menjadi tingkatan-tingkatan lebih rendah lagi.

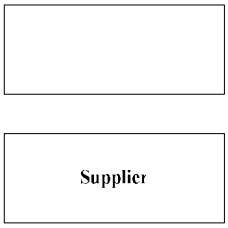
DFD merupakan penurunan atau penjabaran dari diagram konteks. Dalam pembuatan DFD harus mengacu pada ketentuan sebagai berikut :

1. Setiap penurunan level yang lebih rendah harus mempresentasikan proses tersebut dalam spesifikasi proses yang jelas.
2. Penurunan dilakukan apabila memang diperlukan.
3. Tidak semua bagian dari sistem harus ditunjukkan dengan jumlah level yang sama.

Simbol-simbol yang digunakan pada diagram alir data atau *data flow diagram* antara lain :

1. Kesatuan Luar (*External Entity*)

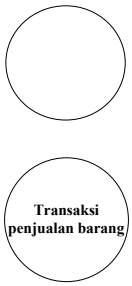
   Kesatuan luar atau *external entity* merupakan kesatuan luar di lingkungan sistem, bisa berupa orang, organisasi, atau sistem lainnya yang berada di lingkungan luarnya yang akan memberikan masukan bagi sistem atau keluaran dari sistem.



Gambar 2.5 Kesatuan luar (*External Entity*)

2. Proses (*Process*)

   Proses adalah kegiatan kerja yang dilakukan oleh manusia, mesin atau komputer dari hasil suatu arus data yang masuk kedalam proses untuk arus data yang akan keluar dari proses.



Gambar 2.6 Proses (*Process*)

3. Arus Data (*Data Flow*)

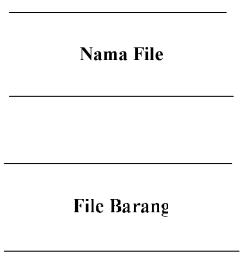
   Arus data mengalir diantara proses, simpanan data dan kesatuan luar. Arus data ini menunjukkan masukan dan keluaran dari suatu proses.



Gambar 2.7 Arus Data (*Data Flow*)

4. Penyimpanan Data (*Data Storage*)

   Simpanan data merupakan simpanan dari data yang dapat berupa suatu arsip atau *database* pada sistem komputer, suatu arsip atau catatan manual, suatu kotak tempat data di meja seseorang, atau suatu agenda/buku.



Gambar 2.8 Penyimpanan Data (*Data Storage*)

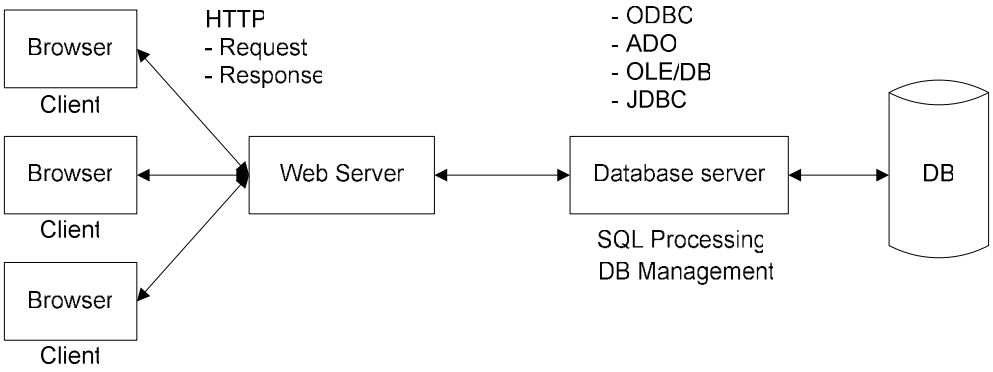
2.6 Konsep Database

Database dalam istilah sederhana, adalah kumpulan informasi yang terorganisasi sehingga seseorang dapat dengan cepat mengambil informasi yang dibutuhkan.

2.6.1 Database Server

Merupakan sebuah database yang digunakan sebagai tempat penyimpanan data (dataware house). Dimana database ini memiliki sebuah nama server sehingga dapat diakses oleh intranet maupun internet ,namun tidak didistribusikan secara langsung (dengan mekanisme tertentu) karena tersimpan dalam bentuk *logic*. Database ini dikhususkan untuk menyimpan record yang besar dan transaksi yang padat.

Tipe database ini merupakan database yang dibuat untuk memenuhi kebutuhan aplikasi berupa client-server maupun berbasis web.



Gambar 2.9 Pengaksesan database server pada website

2.6.2 Database Relasional

Konsep sebuah database adalah terdiri atas tabel-tabel yang terorganisasi. Tabel-tabel tersebut dapat saling berelasi untuk menghasilkan suatu informasi, untuk mengakses data yang ada dalam tabel-tabel tersebut digunakan sebuah perintah SQL (*Structured Query Language*)

2.6.3 SQL (*Structured Query Language*)

Structured Query Language atau SQL merupakan sebuah bahasa query standar yang digunakan untuk mengakses database relasional yang meliputi perintah-perintah untuk menyimpan, menerima, memelihara dan mengatur akses-akses ke database serta digunakan untuk memanipulasi dan menampilkan data dari database tersebut.

Dimana query itu sendiri adalah suatu proses melakukan pengambilan atau pencarian informasi yang ada dalam database relasional berdasarkan kriteria tertentu.

Berdasarkan fungsinya, menurut JOS[2], deklarasi atau perintah SQL dibagi ke dalam dua kategori yaitu DDL dan DML.

1. DDL (Data Defintion Language)

DDL atau *Data Definition Language* adalah bagian dari SQL yang digunakan untuk mendefinisikan data dan objek database, dimana terdapat perintah-perintah sebagai berikut :

Tabel 2.1 Perintah DDL

Perintah	Keterangan
CREATE	Untuk mendefinisikan database, maupun tabel sebagai data yang akan disimpan maupun diakses
ALTER	Untuk memodifikasi tabel, baik itu menambah, menghapus, maupun mengganti kolom/field pada tabel
DROP	Untuk menghapus tabel dan database

2. DML (Data Manipulation Language)

DML atau *Data Manipulation Language* adalah bagian dari SQL yang digunakan untuk memulihkan dan memanipulasi data.

Tabel 2.2 Perintah DML

Perintah	Keterangan
SELECT	Untuk mengambil atau menampilkan data dari tabel pada database
INSERT	Untuk menyisipkan data pada tabel
UPDATE	Untuk memperbaharui nilai suatu data dalam database
DELETE	Untuk menghapus record pada tabel

Terdapat dua buah teknik perancangan basis data, yaitu dengan membuat *Entity Relationship Diagram* atau dengan menerapkan normalisasi terhadap struktur tabel yang telah diketahui.

2.6.4 Normalisasi

Normalisasi adalah sebuah teknik untuk mengoptimasi rancangan basis data relasional dan membebaskan rancangan tersebut dari keganjilan dan persoalan yang potensial. Normalisasi juga dapat diterapkan ke model basis data lainnya. Secara sederhana, normalisasi melibatkan pemecahan data dalam tabel ke dalam tabel yang lebih kecil sampai tiap atribut dalam tiap tabel hanya bergantung pada (beberapa) kunci dalam tabel tersebut.

Adapun tujuan dari normalisasi ini adalah :

1. Meminimumkan duplikasi data
2. Menyediakan fleksibilitas yang diperlukan untuk kebutuhan fungsional yang berbeda
3. Memungkinkan suatu model untuk digambarkan dalam suatu basis data
4. Berguna untuk men-strukturkan data dalam cara-cara tertentu untuk membantu mengurangi atau mencegah munculnya masalah yang berhubungan dengan pengolahan data.

Rancangan database yang buruk, yang tidak dinormalisasi akan menyebabkan persoalan selama basis data tersebut di*install*. Persoalan yang agak ringan, basis data menjadi tidak efisien untuk dijalankan dan sulit untuk dipelihara. Sedangkan persoalan yang terberat adalah setelah basis data di*install*,

pemakai menemukan bahwa basis data tersebut tidak menghasilkan apa yang dibutuhkan atau memberikan hasil yang tidak akurat.

2.6.5 Entity Relational Diagram

Merupakan diagram yang menggambarkan struktur *logic* keseluruhan basis data, diagram ini merupakan pemodelan data yang ada pada suatu sistem. Diagram ini didasarkan pada persepsi bahwa dunia nyata merupakan sekumpulan dari sejumlah obyek dasar (entitas) dan relasi antar obyek-obyek tersebut.

Terdapat beberapa istilah dan simbol yang digunakan dalam ERD, yaitu :

1. Entitas

   Entitas adalah sebuah obyek yang dapat dibedakan dari obyek lainnya. Entitas direpresentasikan dengan bentuk persegi empat dalam ERD, sebagai berikut :



Gambar 2.10 Simbol Entitas

2. Atribut

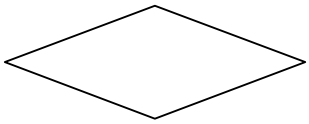
   Atribut merupakan properti dari sebuah entitas ataupun relasi. Atribut direpresentasikan dengan bentuk elips dalam ERD, sebagai berikut :



Gambar 2.11 Simbol Atribut

3. Relasi atau keterhubungan

   Sebuah relasi menggambarkan suatu asosiasi antar sejumlah entitas, dimana sebuah relasi pun dapat mempunyai atribut. Keterhubungan atau relasi direpresentasikan dengan bentuk wajik dalam ERD, sebagai berikut :



Gambar 2.12 Simbol Relasi

4. Simbol Garis

   Simbol ini sebagai penghubung simbol lainnya pada diagram. Simbol ini digambarkan sebagai berikut :

Gambar 2.13 Penghubung simbol dalam ERD

5. Kardinalitas Relasi (Derajat Keterhubungan)

   Merupakan kemungkinan hubungan yang terjadi antar entitas, yang dibagi menjadi empat macam yaitu :

   a. 1 – 1
   b. 1 – N
   c. N – 1
   d. N – N / N - M

2.6.6 Kamus Data

Kamus data atau *data dictionary* adalah katalog fakta tentang data dan kebutuhan-kebutuhan informasi dari suatu sistem informasi. Kamus data berisikan beberapa tabel yang memuat nama dari data, tipe dari data yang akan kita gunakan

di dalam menyusun program. Kamus data digunakan untuk menghindari duplikasi elemen-elemen dan menghindari konflik antara elemen-elemen tersebut. Kamus data mendefinisikan elemen data dengan fungsi sebagai berikut :

1. Menjelaskan aliran data dan penyimpanan dalam DFD
2. Mendeskripsikan komposisi paket data yang bergerak melalui aliran
3. Mendeskripsikan komposisi penyimpanan data
4. Menspesifikasikan nilai dan satuan yang relevan bagi penyimpanan dan aliran data
5. Mendeskripsikan hubungan detil antar penyimpanan yang akan menjadi titik perhatian dalam diagram keterhubungan entitas (ERD).

2.7 Konsep Dasar Internet

Pada subbab ini akan dijelaskan mengenai internet dan komponen yang ada di dalamnya.

2.7.1 Pengertian Internet

Internet yang berasal dari kata *interconnection networking*, merupakan teknologi yang tercipta dari hubungan komputer-komputer yang berbeda tipe dan sistemnya yang membentuk sistem jaringan yang mencakup seluruh dunia dengan menggunakan jalur komunikasi.

2.7.2 World Wide Web

World Wide Web (WWW) dan internet sangat berkaitan satu sama lain. WWW merupakan pelayanan internet terdistribusi dengan konsep *hypertext* antar dokumen yang menggunakan bahasa HTML untuk format dokumen.

WWW adalah suatu set aplikasi komunikasi dan sistem perangkat lunak yang memiliki karakteristik sebagai berikut, berdasarkan YAH[7] :

1. Terletak pada internet
2. Menggunakan protokol TCP/IP
3. Mengenal *tag-tag* HTML (*Hypertext Mark Language*)
4. Menggunakan model *client/server*
5. Mampu merelasikan komunikasi data dua arah
6. Memungkinkan *client* untuk mengakses *server* dengan berbagai protokol seperti HTTP, FTP, Telnet dan lain-lain.
7. Memungkinkan *client* untuk mengakses *server* dalam berbagai media seperti teks, audio, dan video.
8. Menggunakan model alamat *Uniform Resource Language* (URL)

2.7.3 Web Statis

Merupakan salah satu jenis web, yang mempunyai struktur dan isi halaman yang tetap. Dimana tampilan web ini tidak akan pernah berubah jika tidak ada proses *editing* pada halaman web tersebut. Proses editing web dilakukan dengan cara mengetikkan *tag* HTML pada *script editor*

2.7.4 Web Dinamis

Web Dinamis merupakan solusi dari permasalahan yang dihadapi web statis. Web dinamis memberikan kemudahan dalam *maintenance web*, dimana pengelola situs tidak harus perlu melakukan proses *editing* dokumen HTML, tetapi cukup dengan melakukan penambahan atau perubahan dalam *database* yang digunakan web dinamis tersebut.

2.7.5 Web Server

Menurut SUT[7], *web server* identik dengan sebuah perangkat lunak yang terintegrasi dengan sistem operasi yang digunakan, *web server* akan menerima input dari user (client request) untuk diproses melalui penerjemahan *server side script* (seperti ASP) dan menghasilkan *output* (response). *Web server* juga dapat berinteraksi dengan penyimpanan data seperti database. Pemilihan *web server* sangat bergantung pada *web programming* yang akan digunakan.

Web server yang dapat meng-*interpreter* script ASP antara lain adalah :

1. PWS (*Personal Web Server*) , yang dapat diinstall pada sistem operasi Windows 98 dan Windows ME. PWS merupakan produk dari *Microsoft*
2. IIS (*Internet Information Services*), yang dapat diinstall pada sistem operasi Windows NT, 2000, XP, dan Windows Server 2003. IIS merupakan produk dari Microsoft
3. ChiliASP, InstantASP, dan vWebServer untuk sistem operasi *non-Windows*.

2.7.5.1 IIS 5.1 (Microsoft Internet Information Services 5.1)

Dalam pembangunan website dinamis Gereja Isa Almasih Jemaat Lengkong Besar, web server yang digunakan adalah *IIS (Internet Information Services)*.

Internet Information Services adalah program aplikasi web server yang dapat berjalan dengan baik di lingkungan Windows NT, Windows 2000, Windows XP, dan Windows 2003. IIS adalah bagian dari sistem operasi Windows, dan merupakan kumpulan tiga internet service yang terdiri dari :

1. Web server. Sebuah *service* yang menyediakan layanan protokol HTTP (web), secara *default* berada pada port 80.
2. *FTP server (File Transfer Protocol)*. Sebuah *service* yang menyediakan layanan distribusi file, secara default berada pada port 21.
3. *SMTP server (Simple Mail Transfer Protocol)*. Sebuah *service* yang menyediakan layanan pengiriman email, secara default berada pada port 25.

2.7.6 Web Browser

Merupakan perangkat lunak yang digunakan untuk melakukan navigasi berbagai data dan informasi pada halaman *web* ataupun untuk menampilkan file-file pendukung halaman web.

2.8 Konsep Dasar Pemrograman Berbasis Web

Pada subbab ini diuraikan teori mengenai pemrograman berbasis web dan *script* pemrograman yang digunakan dalam pembangunan website dinamis Gereja Isa Almasih Jemaat Lengkong Besar Bandung.

2.8.1 Pengertian Pemrograman Berbasis Web

Menurut SUT[7], pemrograman berbasis *web* atau *web programming*, menggunakan basis pada proses *interpreter* (penerjemahan perintah). Perintah kode yang di*interpreter* adalah dalam bentuk file teks biasa, sehingga sering disebut *web scripting*. *Web Scripting* adalah file teks yang berisi *script* (kode yang digunakan untuk pemrograman) akan diperiksa kevalidannya setiap kali *script* tersebut diakses sehingga pada akhir prosesnya akan dihasilkan *output* bertipe HTML yang ditampilkan melalui *web* browser. Hal inilah yang menyebabkan

kecepatan proses *web* programming jauh lebih cepat dibandingkan dengan *compiler programming*. Karena perintah kode dalam *web programming* hanya berupa teks biasa dan tidak perlu dikompilasi, sedangkan *compiler programming* langsung menjalankan *file executeable*nya saja yang sebelumnya telah diperiksa kevalidannya.

2.8.2 HTML (*Hyper Text Markup Language*)

HTML adalah suatu format data yang digunakan untuk membuat dokumen *hypertext* yang dapat dibaca dari suatu *platform* komputer ke *platform* komputer lainnya tanpa perlu melakukan suatu perubahan apapun. HTML tidak hanya mampu menampilkan teks tetapi juga dapat diberikan format pada teks tersebut misal frame, tabel, dapat digabungkan dengan obyek suara, dan video

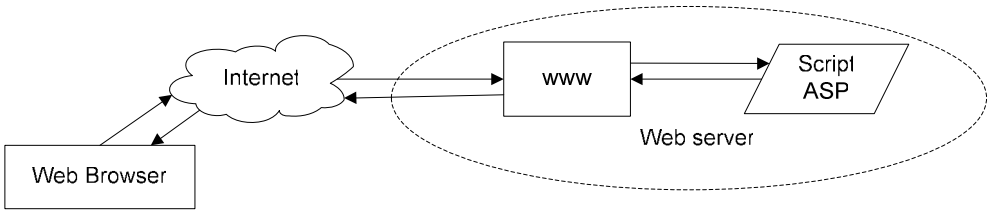
2.8.3 Javascript

Javascript adalah bahasa *script* yang dikembangkan oleh Netscape untuk membuat *web* dinamis, *script* ini mempunyai kemiripan dengan bahasa pemrograman C, serta berorientasi objek. Javascript digunakan bersama dengan HTML, dimana *script* ini disisipkan ke dalam HTML dengan *delimiter* <script language=’javascript’> dan diakhiri </script>

JavaScript adalah bahasa yang *“case sensitive”* artinya membedakan penamaan variabel dan fungsi yang menggunakan huruf besar dan huruf kecil, contoh variabel atau fungsi dengan nama *TEST* berbeda dengan variabel dengan nama *test*. Dan setiap instruksi diakhiri dengan karakter titik koma (;).

2.8.4 ASP (*Active Server Page*) VBscript

Menurut SUT[7], ASP (*Active Server Pages*) merupakan salah satu dari script pemrograman *web* berbasis teks. ASP, yang merupakan teknologi Microsoft, adalah suatu *script* yang bersifat *server side*, yang berarti seluruh proses pengerjaan *script* berlangsung pada *server*.



Gambar 2.14 Proses kerja *server side scripting*

Dengan ASP, dapat dibuat aplikasi *database* berbasis *web*. ASP menggunakan teknologi VBScript dan Jscript. Secara *default* ASP menggunakan VBscript, dimana VBscript ini merupakan *script* yang dikembangkan Microsoft dan salah satu bagian dari pemrograman Visual Basic namun tidak menggunakan *design time-environtment* (lingkungan desain kerja) seperti pemrograman Visual Basic umumnya, *script* ini dapat digunakan bersama dengan HTML untuk membuat *web* semakin menarik, interaktif, dan dinamis. *Script* ASP selalu menggunakan *delimiter* berupa karakter <% dan %>. *Script* ASP dapat dijalankan pada lingkungan *Windows*, dan dapat menggunakan *web server* seperti *Internet Information Services (IIS)*, ataupun *Microsoft Personal Web (PWS)*. ASP juga dapat dijalankan pada sistem operasi *non-Windows*, seperti *ChiliASP* ataupun *InstantASP*.

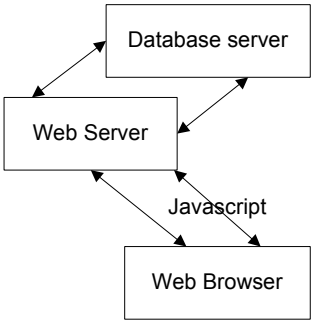
File ASP mempunyai ekstensi ".asp". File ini juga dapat berisi *script* HTML, XML, dan *script* lainnya.

Kelebihan dari ASP dan apa yang dapat dilakukan ASP adalah sebagai berikut. :

1. Mengubah ataupun menambahkan isi halaman *web* dalam satu basis data dan tidak lagi pada kode HTMLnya.
2. Merespon pada *query user* atau data yang di-*submit* dari *form* HTML
3. *Client* dapat mengakses *database* pada *server* dan menampilkan hasilnya di *browser client*
4. Menyediakan keamanan. Dalam hal ini pada *server*, *script* ASP telah diubah menjadi HTML murni, yang kemudian HTML tersebut dikirimkan ke *browser client*, sehingga *script* ASP tidak dapat terbaca pengunjung dan tidak dapat ditiru oleh pengunjung web.
5. Pemrograman ASP yang lebih baik dapat meminimalisasikan kerja jaringan karena terhindarinya percakapan bolak balik antara *client* dan *server*.

2.8.5 AJAX (Asynchronous Javascript and XML)

Berdasarkan ALI[1], AJAX sendiri diperkenalkan oleh Jesse James Garret pada bulan Februari 2005. AJAX digunakan pada *website* yang berinteraksi dengan *server* melalui *javascript* secara asinkron (*background*), sehingga pengguna tidak perlu memproses keseluruhan isi halaman *web*.



Gambar 2.15 Arsitektur AJAX

Untuk mengaplikasikan AJAX dalam *website*, yang dibutuhkan adalah browser yang menyediakan layanan Javascript, dan komponen XMLHTTP bagi pengguna Internet Explorer (IE), dan XMLHttpRequest untuk pengguna Firefox, Safary, Opera dan browser lainnya. Kelebihan AJAX adalah :

1. Penggunaan *bandwidth* yang lebih efisien, dikarenakan halaman *web* tidak perlu adanya *refresh* untuk menampilkan data yang lebih baru, selain itu halaman web yang dibuat dengan AJAX dapat diproses relatif cepat
2. Pendekatan AJAX membuat adanya pemisahan antara metode dan format yang digunakan untuk penyampaian informasi melalui *web*.

AJAX menggunakan teknologi yang ada untuk memungkinkan sebuah data di-*generate* di *background*, berdasarkan ALI[1], teknologi pendukung AJAX adalah:

1. XHTML (*eXtensible Markup Language*) serta CSS (*Cascading Style Sheet*) untuk melakukan *marking* dan pengaturan *style* terhadap informasi.
2. DOM (*Document Object Model*), berfungsi untuk melakukan pengolahan dokumen HTML secara dinamis. DOM bersifat *multiplatform*, yang diakses menggunakan *script client side* / ECMAScript seperti Javascript atau Jscript dapat menampilkan informasi secara dinamis dan interaktif.

3. XMLHttpRequest yang berguna untuk menukar data secara asikronus dengan web server.
4. XML merupakan format umum yang dipakai untuk melakukan transfer data antar *server* atau *client*.

2.9 Video On Demand

Video on demand merupakan fasilitas multimedia, dimana *client* dapat memutar video yang diiinginkan dengan *request* video tersebut kepada server. Namun fasilitas ini bersifat *unicast* , yaitu transmisi informasi dilakukan dari satu pengirm ke satu penerima (*point to point*) dalam arti kata sumber daya yang ada di jaringan dan *server* dialokasikan pada permintaan *client* tersebut.

2.10 Macromedia Dreamweaver 8.0

Macromedia Dreamweaver 8.0 adalah *web editor* untuk merancang, mengkodekan (*coding*), dan mengembangkan *website, webpage*, dan aplikasi berbasis web lainnya. Macromedia Dreamweaver 8 juga dilengkapi dengan *visual editing environment*, sehingga memudahkan untuk merancang tampilan *web*. Dalam aplikasi ini, semua elemen *web* yang dibangun, dapat ditampilkan sehingga lebih memudahkan pengaturan elemen *web*. Dengan aplikasi ini, dapat dibangun web dinamis, maupun aplikasi berbasis *web* dengan teknologi *server* seperti CFML, ASP.NET, ASP, JSP, dan PHP.

2.10.1 Fasilitas baru Macromedia Dreamweaver 8.0

Macromedia Dreamweaver 8.0 mempunyai fasilitas tambahan, diantaranya adalah :

1. *Visual authoring* dengan data XML
2. Panel CSS yang *unified*.
3. Visualisasi tampilan CSS
4. *CSS Rendering*
5. Mendukung untuk WCAG/W3C dengan prioritas 2 checkpoint
6. WebDAV, menyediakan authentikasi dan SSL untuk keamanan file transfer.

2.10.2 Kelebihan Macromedia Dreamweaver 8.0

Kelebihan yang dimiliki oleh Macromedia Dreamweaver 8.0 adalah :

1. *Background file transfer*, yaitu pengerjaan tetap dapat dilakukan ketika Dreaweaver 8 melakukan upload file ke server.
2. *Zoom*, dapat memperbesar tampilan desain sehingga dapat mengerjakan desain dengan lebih detail
3. *Guides*, dengan kelebihan ini dapat membandingkan tampilan halaman dan halaman mockup dengan akurasi pixel untuk measure tampilan halaman.
4. *Coding Toolbar*, menyediakan sejumlah tombol untuk fitur pengkodean terbaru dalam *gutter bar* disepanjang sisi *Code View*
5. *Code Collapse*, berfokus pada code yang hanya ingin dilihat dengan cara menyembunyikan dan meng-ekspan blok-blok kode.
6. *Workspace Layouts*, mengatur dan menyimpan konfigurasi *workspace*. Dreamweaver 8.0 mempunyai empat macam konfigurasi yang dibutuhkan *designer* dan *coder*, selain itu juga dapat dibuat *workspace* yang baru.

7. *Tabbed documents for the Mac*, dokumen tab baru pada Mac dapat membantu membuat antarmuka dan menampilkan dokumen secara mudah
8. *New Starter Pages*, layout dan desain yang baru dapat memudahkan dalam pembuatan situs secara cepat
9. *Improved site synchronize and check-in/check-out*, mengatur situs dengan kenyamanan dan reliabilitas yang meningkat. Mencegah ketidaksengajaan penimpaan penulisan pada pekerjaan yang lainnya dengan melakukan fungsioal *check-in/check-out*
10. *Compare files*, membandingkan file-file untuk mengidentifikasi kapan file tersebut diubah. File yang dibandingkan dapat berupa file lokal, file dalam komputer lokal, file dalam komputer *remote*.
11. *Paste special*, dengan opsi *pasting* baru dalam Dreamweaver, dapat memilih antara *paste formatting* maupun *paste text only*.
12. *Site-relative references*, bekerja seperti pada sisi server pada saat desain dan saat eksekusi dengan memastikan referensi file lokal pada situs yang dikerjakan.
13. *Code editing improvements*, menggabungkan kendali yang lebih baik untuk menyediakan *code hint* dan melengkapi *tag-tag* untuk menyesuaikan dengan *coding* yang dilakukan.

2.11 Microsoft SQL Server 7.0

Berdasarkan JOS[2], Microsoft SQL *Server* 7 adalah sebuah *database* relasional yang dirancang untuk mendukung aplikasi dengan struktur *client/server* yang dikeluarkan oleh *Microsoft,* dimana *database* terdapat pada komputer pusat

yang disebut *server* dan informasi yang digunakan bersama-sama oleh beberapa *user* yang menjalankan aplikasi didalam komputer lokalnya yang disebut *client.*

2.11.1 Arsitektur SQL Server

Menurut JOS[2], database SQL Server dibagi ke dalam beberapa komponen logikal, seperti tabel, view, dan elemen-elemen lain yang terlihat oleh user. Elemen-elemen ini secara fisik disimpan di dalam dua atau lebih file di dalam disk. Dimana format file tidak diketahui oleh user sistem.

2.11.2 Komponen SQL Server

Berdasarkan JOS[2], SQL Server terdiri dari beberapa komponen kunci sebagai berikut :

1. Database

   Database adalah sekumpulan tabel,view, indeks, trigger, prosedur dan objek-objek lain, mengandung objek-objek yang digunakan untuk mewakili, menyimpan, dan mengakses data.

2. Tabel

   Tabel adalah inti dari database. Komponen ini menyimpan record-record data dan hubungannya dengan tabel lainnya.

3. Diagram Database

   Diagram database merupakan representasi grafik dari tabel, indeks, dan view yang disimpan oleh database dan bisa dimanipulasi dengan teknik *drag-and-drop* dan interaksi dengan kotak dialog.

4. Indeks

   Komponen ini berupa file-file tambahan yang meningkatkan kecepatan akses dari baris-baris tabel. Indeks adalah file jenis khusus yang bekerja sama dengan tabel. Tujuannya adalah untuk mempercepat proses mengakses record atau sekelompok record tertentu.

5. View

   Komponen ini menyediakan cara melihat data yang berbeda di dalam satu atau lebih tabel.

6. *Stored Procedure*

   Komponen ini berupa prosedur yang dibuat dalam *Transact SQL* yang digunakan untuk mengakses data dalam database. *Stored procedure* atau prosedur tersimpan bisa menerima parameter (nilai yang dikirimkan ke prosedur) untuk memprosesnya tetapi tidak bisa mengembalikan nilai apapun.

7. Trigger

   Trigger adalah prosedur tersimpan yang secara otomatis dijalankan apabila data didalam tabel berubah karena eksekusi perintah SQL INSERT, UPDATE , atau DELETE

8. *Full Text Indexes*

   Index khusus ini mengizinkan eksekusi *query* berdasarkan pada kolom yang mengandung tipe *varchar* dan *text*. Indeks ini dibuat dan dihapus melalui penggunaan prosedur tersimpan, hanya bisa ada satu indeks ini pada setiap tabel, dan hanya di-*update* apabila diminta.