BAB II

TINJAUAN PUSTAKA

2.1 Tinjauan Perusahaan

Pada tinjauan perusahaan akan dibahas mengenai sejarah berdirinya perusahaan, tempat dan kedudukan perusahaan, bidang pekerjaan perusahaan, dan struktur organisasi beserta uraian tugas dari masing-masing bagian.

2.1.1 Sejarah Singkat PT. PERTAMINA Depot Bula

Pada tahun 1978 Pertamina Depot Bula mulai di bangun dengan semua sarana & pra sarana sampai dengan tahun 1982. Pada tanggal, 30 Maret 1992 Pertamina Depot Bula diresmikan, oleh Gubernur Maluku : Bapak HASAN SLAMET .

Jumlah Karyawan 18 orang terdiri dari :

1. Kepala Pertamina = 1 orang (Bpk . SABIRIN)
2. Pemuka Distribusi Teknik (Dis/Tek) = 1 orang
3. Pemuka Administrasi = 1 orang
4. Juru Tulis = 2 orang
5. Motoris = 4 orang
5. Tukang Pemeliharaan = 2 orang

6.Sekurity (Pengamanan) = 6 orang

7.Tukang isi (Pengisian) = 1 orang

Pembangunan Tangki Bahan Bakar Minyak (BBM) Samy Burried (Tangki Pendam) Premium,M.Tanah, M.Solar masing - masing 1 buah tangki & sarana penunjang operasional lainya.

Dengan adanya program Pemerintah, semulanya pelayanan BBM di tingkat kecamatan kini telah berubah status menjadi Kabupaten Seram Bagian Timur (SBT), sehingga arus pelayanan pendisrtibusin BBM semakin tajam dan cukup significant hingga saat ini .

Kondisi Sekarang :

Dengan adanya perubahan Restrukturisasi (Perampingan) di Pertamina secara menjeluruh sehingga data riil jumlah Personil (SDM) menjadi 8 orang terdiri dari :

1. Kepala Pertamina = 1 orang (Bapak YUNUS MATATULA)

2. Pengawas Penerimaan Penimbunan Penyaluran (Pws P3) = 1 orang

3. Asisten Keuangan = 1 orang

4. Asisten Pemeliharaan / LK3 = 1 orang

5. Asisten Penerimaan & Penimbunan = 1 orang

6. Asisten Layanan Jual = 1 orang

7. Asisten Layanan Pekerja = 1 orang

NABAN ; Tenaga Bantuan :

1. Tenaga Outsouching Pengamanan = 5 orang

2. Tenaga Tanker Discharge (regu pembongkaran) = 2 orang

3. Tenaga Cleaning Service = 1 orang

4. Tenaga Babat Rumput = 2 orang

Sarana Penunjang Operasional.

- Pembangunan tangki Premium vertikal (Tegak lurus) = 1 buah capasitas 500 kilo liter (Ton)

M.Tanah = 1 buah capasitas 250 kilo liter (Ton)

M.Solar = 2 buah masing masing : 250 kilo liter & 1000 kilo liter (Ton)

- 1 buah Gudang

- 1 buah Bengkel

- 1 buah Rumah pompa : (4 unit Mesin Motor Pompa)

- 1 buah Dermaga

- 1 buah Rumah Dinas

- 1 buah Kantor

serta pembangunan sarana penunjang operasional lainnya.

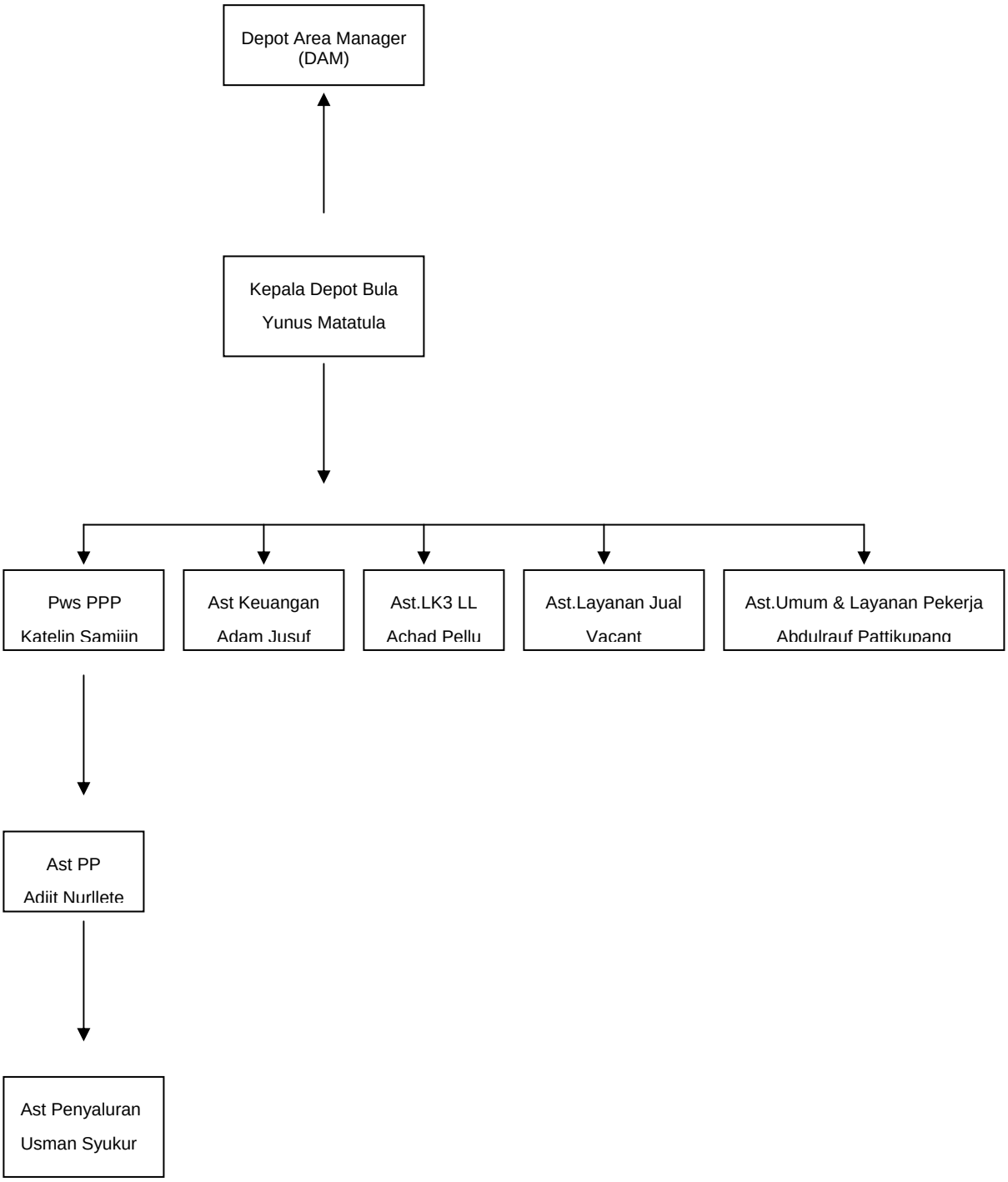
2.1.2 Visi dan Misi

Visi : Menjadi Perusahaan yang unggu, maju dan terpandang.

Misi : - Melakukan usaha dalam bidang energi dan Petro kimia

- Merupakan Entitas bisnis yang dikelola secara proposional,
- kompetitif dan berdasarkan tata nilai unggulan.
- Memberikan nilai plus lebih bagi pemegang saham,pelayanan, pekerja dan masyarakat serta mendukung pertumbuhan ekonomi nasional.

2.1.3 Struktur Organisasi



Keterangan :

Pws = Pengawas

Ast = Asisten

PPP = Penerimaan Pembongkaran dan Penimbunan

PP = Penerimaan & Penimbunan

P = Penyaluran

LK3LL = Lindungan ,lingkungan Kesehatan & Keselamatan

2.1.4 Uraian Tugas (Job Discription)

Kepala Depot :

- Bertanggung jawab penuh terhadap semua asset termasuk manusia yang ada dalam lokasi dan diberi wewenang untuk pengambilan keputusan. (Decetion Marking) dalam melaksanakan tugasnya
- Melaksanakan pembinaan lingkungan dengan pola kemitraan dan menumbuhkan aktifitas sosial dengan mekanisme sosial Responsibility & Community Development.
- Menciptakan hubungan kerja yang baik secara internal maupun eksternal ,seta membangun kerja sama dengan Instansi terkait.
- Bertanggung jawab atas anggaran Perusahaan di lokasi kerja.

- Bertanngung jawab terhadap pendistribusian BBM kepada seluruh pelanggan,

Pws.PPP :

- Melaksanakan kegiatan penerimaan Bahan Bakar Minyak (BBM) melalui pembongkaran kapal tanker sehingga aman, lancar, dan terkendali dengan sasaran hasil penerimaan dan kerugian pembongkaran dapat dipertanggung jawabkan.
- Melaksanakan pemeriksaan hasil pengukuran dan perhitungan stock BBM yang benar dan tepat waktu pelaporan dengan sasaran mendapatkan identifikasi dari jumlah produk BBM.
- Melaksanakan perencanaan penyaluran produk kepada lembaga penyalur / konsumen secara tepat waktu, mutu, volume dengan sasaran sesuai Objective Thruput / alokasi dan ketahanan stock selalu ada.
- Melaksanakan pemeriksaan fasilitas penerimaan, penimbunan dan penyaluran dengan sasaran terjaminnya kontinuitas produk.

Asisten Keuangan :

- Melaksanakan pekerjaan perencanaan biaya operasi dengan sasaran pembiayaan/pendanaan operasional dapat terpenuhi
- Melaksanakan pemeriksaan rekapitulasi harian Kas / Bank dengan sasaran pembukuan biaya operasional dapat dipertanggung jawabkan.

- Koordinasi dengan Customer / pelanggan terhadap penyetoran BBM pada bank persepsi dan kerja sama dengan pihak bank dengan sasaran penyetoran BBM harus sesuai dengan jumlah uang yang disetor.
- Membuat laporan keuangan secara periodik, harian, mingguan, bulanan dan dapat dipertanggung jawabkan secara upto date ke kantor pusat.

Asisten LK3LL :

- Melaksanakan perencanaan perbaikan fasilitas operasi dengan sasaran tercapai pelayanan produk secara kontinyu.
- Melaksanakan perawatan dan latihan pemadam kebakaran degan sasaran perlindungan pekerja dan fasilitas siap untuk selalu dipakai.
- Membuat laporan secara periodik, harian, mingguan, bulanan serta mengawasi pekerjaan pemeliharaan oleh kontraktor di lokasi kerja dengan sasaran hasil pekerjaan mencapai nilai proyek dan dapat dipertanggung jawabkan.

Asisten PP :

- Melaksanakan kegiatan penerimaan BBM melalui kapal tanker secara aman, lancar dan terkendali dengan sasaran hasil penerimaan dan kerugian dapat di pertanggung jawabkan.
- Melaksanakanpemeriksaan hasil pengukuran dan perhitungan stock BBM yang benar dan tepat waktu pelaporannya dengan sasaran mendapatkan identifikasi dari jumlah produk BBM.

- Melaksanakan perencanaan penyaluran produk kepada lembaga penyalur / konsumen tepat waktu,mutu,volume, dengan sasaran sesuai objective thruput / alokasi dan ketahanan stock tetap tersedia.

Asisten Penyaluran :

- Melaksanakan pencatatan setiap hari penyaluran BBM kepada konsumen dengan sasaran pelayanan kepada masyarakat tetap terjamin sehingga tidak terjadi kekosongan BBM di pasaran.
- Membuat laporan penyaluran setiap periode, harian,mingguan, bulanan dan melapor ke kepala Pejualan di pusat secara up to date dengan sasaran jumlah pemakaian BBM bulanan dapat di perhitungkan sesuai kondisi yang ada di daerah penyaluran.
- Melaksanakan perencanaan penyaluran produk kepada lembaga penyalur / konsumen tepat waktu, mutu, volume, dengan sasaran sesuai objective thruput / alokasi dan ketahanan stock tetap tersedia.

Asisten umum dan layanan pekerja :

- Melaksanakan kegiatan pelayanan kepada kebutuhan pekerja sesuai kebutuhannya dengan sasaran tidak terjadi kesenjangan sosial terhadap kebutuhan pekerja.

- Melayani kesejahteraan pekerja terutama kesehatan dengan sasaran pekerja tetap sehat dalam melaksanakan tugas .
- Membuat laporan secara periode, harian, mingguan, bulanan, serta bertanggung jawab terhadap pelaksanaan sesuai bidangnya.
- Melaksanakan pencatatan / agenda surat menyurat, koordinasi dengan pihak medis, memperhatikan kebutuhan operasional kantor dengan sasaran kemudahan dalam menjalankan tugas masing-masing fungsi.

2.2 Landasan Teori

Pada landasan teori ini akan dibahas teori-teori yang dapat membantu memecahkan permasalahan yang ada. Teori-teori tersebut meliputi konsep dasar dan definisi-definisi yang berkaitan dengan perangkat lunak yang digunakan serta faktor-faktor pendukung dalam pelaksanaan perancangannya.

2.2.1 Konsep Dasar Sistem Informasi

2.2.1.1 Definisi Sistem

Terdapat dua kelompok pendekatan dalam mendefinisikan sistem, yaitu yang menekankan kepada prosedur dan menekankan kepada komponen atau elemen. Pendekatan sistem yang lebih menekankan pada prosedur mendefinisikan sebagai berikut:

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

Pendekatan sistem yang menekankan pada komponen atau elemen-elemen mendefinisikan sebagai berikut :

*"Sistem adalah elemen-elemen yang berinteraksi untuk mencapai suatu tujuan tertentu"***.**

Sistem informasi didefinisikan oleh Robert A. Leitch dan K. Roscoe Davis sebagai berikut:

"Sistem informasi adalah suatu sistem didalam organisasi yang mempertemukan kebutuhan pengolahan transaksi harian, mendukung operasi, bersifat manajerial dan kegiatan strategi dari suatu organisasi dan menyediakan pihak luar tertentu dengan laporan-laporan yang diperluka".

Sistem informasi merupakan suatu sistem di dalam organisasi yang mempertemukan kebutuhan pengolahan data (kejadian), mendukung operasi atau proses, menyediakan laporan atau dokumen yang diperlukan. Suatu sistem informasi yang dibuat berisi himpunan terintegrasi dari komponen manual dan komponen terkomputerisasi yang bertujuan untuk mengumpulkan data, menyimpan data dan menghasilkan informasi untuk pemakai.

Adapun pengenalan untuk sistem informasi biasanya terdiri dari :

1. Memahami sistem yang ada dengan cara menyampaikan informasi dan menganalisis sistem yang ada.
2. Mendefinisikan kebutuhan sistem baru yaitu perimbangan, perencanaan, kebutuhan keluaran, masukan, simpanan, pengolahan dan mendefinisikan kriteria penilaian.
3. Proses desain sistem yaitu desain keluaran, desain masukan, desain file, desain pengolahan sistem, pengendalian sistem dan dokumentasi.
4. Pengembangan dan implementasi sistem, yaitu menilai perangkat lunak, dokumentasi sistem dan pelatihan, pengetesan sistem dan implementasi sistem.

2.2.1.2 Komponen Sistem Informasi

John Burch dan Gary Grudnitski mengemukakan bahwa sistem informasi terdiri dari komponen-komponen yang disebutnya dengan istilah blok bangunan (*building block*), yaitu blok masukan (*input block*), blok model (*model block*), blok keluaran (*output block*), blok teknologi (*technologi block*), blok basis data (*database block*), dan blok kendali (*komputers block*). Sebagai suatu sistem, keenam blok tersebut masing-masing saling berinteraksi satu dengan yang lainnya membentuk satu kesatuan untuk mencapai sasarannya.

1. Blok Masukan

   Input mewakili data yang masuk ke dalam sistem informasi. Input disini termasuk metode-metode dan media untuk menangkap data yang akan dimasukkan, yang dapat berupa dokumen-dokumen dasar.

2. Blok Model

   Blok ini terdiri dari kombinasi prosedur, logika dan model matematik yang akan memanipulasi data input dan data yang tersimpan di basis data dengan cara yang sudah tertentu untuk menghasilkan keluaran yang diinginkan.

3. Blok Keluaran

   Produk dari sistem informasi adalah keluaran yang merupakan informasi yang berkualitas dan dokumentasi yang berguna untuk semua tingkatan manajemen serta semua pemakai sistem.

4. Blok Teknologi

   Teknologi merupakan “kotak alat” (*tool box*) dalam sistem informasi. Teknologi digunakan untuk menerima output, menjalankan model, menyimpan dan mengakses data, menghasilkan dan mengirimkan keluaran dan membantu pengendalian dari sistem secara keseluruhan. Teknologi terdiri dari 3 bagian utama, yaitu teknisi (*humanware* atau *brainware*), perangkat lunak (*software*), dan perangkat keras (*hardware*). Teknisi dapat berupa orang-orang yang mengetahui teknologi dan membuatnya dapat beroperasi.

Misalnya teknisi adalah operator komputer, pemrogram, operator pengolah kata, spesialis telekomunikasi, analis sistem, penyimpanan data dan lain sebagainya.

5. Blok Basis Data

   Basis data (*database*) merupakan kumpulan dari data yang saling berhubungan satu dengan yang lainnya, tersimpan di perangkat keras komputer dan digunakan perangkat lunak untuk memanipulasinya. Data perlu disimpan di dalam basis data untuk keperluan penyediaan informasi lebih lanjut. Data di dalam basis data perlu diorganisasikan sedemikian rupa, supaya informasi yang dihasilkan berkualitas. Organisasi basis data yang baik juga berguna untuk efisiensi kapasitas penyimpanannya. Basis data diakses atau dimanipulasi dengan menggunakan perangkat lunak paket yang disebut dengan DBMS (*Database Management Systems*).

6. Blok Kendali

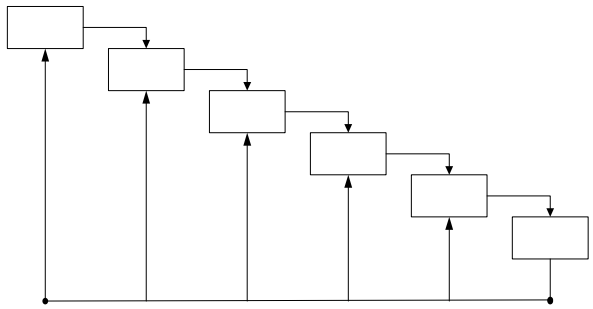
   Banyak hal yang dapat merusak sistem informasi, seperti misalnya bencana alam, api, temperatur, air, debu, kecurangan-kecurangan, kegagalan-kegagalan sistem itu sendiri, kesalahan-kesalahan, ketidak efisienan, sabotase dan lain sebagainya. Beberapa pengendalian perlu dirancang dan diterapkan untuk meyakinkan bahwa hal-hal yang dapat merusak sistem dapat dicegah ataupun bila terlanjur terjadi kesalahan-kesalahan dapat langsung cepat diatasi.

2.2.2 Model Proses 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 serta penyampaian yang dibutuhkan. Perkembangan perangkat lunak bisa dianggap sebagai lingkaran pemecahan masalah dimana terdapat empat keadaan berbeda, yaitu status quo, definisi masalah, perkembangan teknis memecahkan masalah di keseluruhan aplikasi dari banyak aplikasi, dan integrasi pemecahan menyampaikan hasil kepada siapa yang membutuhkan pertama kali. Bermacam-macam model proses yang berbeda pada perangkat lunak sebagai berikut:

1. Model Sekuensial Linier atau *Waterfall*

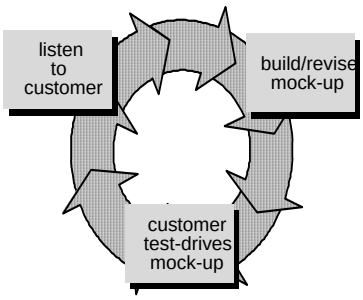
   Sekuensial linier mengusulkan sebuah pendekatan kepada perkembangan perangkat lunak yang sistematik dan sekuensial yang mulai pada tingkat dan kemajuan sistem pada seluruh analisis, desain, kode, pengujian, dan pemeliharaan.



Gambar Diagram Waterfall

2. Model Prototipe

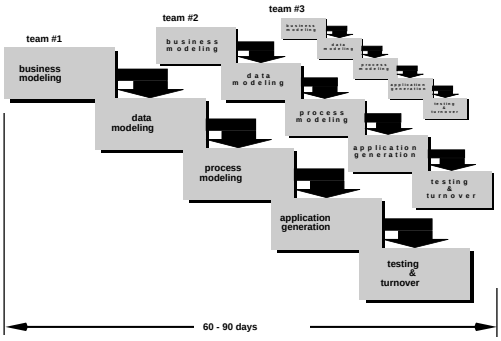
Prototyping paradigma dimulai dengan pengumpulan kebutuhan. Pengembang dan pelanggan bertemu dan mendefinisikan obyektif keseluruhan dari perangkat lunak, mengidentifikasi segala kebutuhan yang diketahui, dan area garis besar dimana definisi lebih jauh merupakan keharusan kemudian dilakukan "perancangan kilat". Perancangan kilat berfokus pada penyajian dari aspek-aspek perangkat lunak tersebut,yang akan nampak bagi pelanggan / pemakai (contohnya pendekatan input dan format output).



Gambar Model Prototipe

3. Model RAD

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 di mana perkembangan cepat dicapai dengan menggunakan pendekatan konstruksi berbasis komponen.



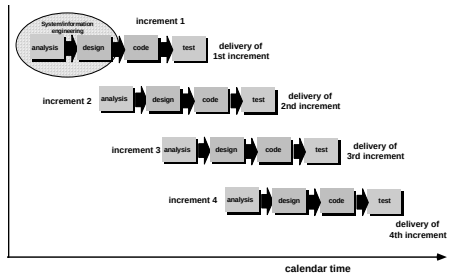
Gambar Model RAD

4. Model Proses Perangkat Lunak Evolusioner

Model evolusioner adalah model iteratif. Model itu ditandai dengan tingkah laku yang memungkinkan perekayasa perangkat lunak mengembangkan versi perangkat lunak yang lebih lengkap sedikit demi sedikit.

a. Model Pertambahan

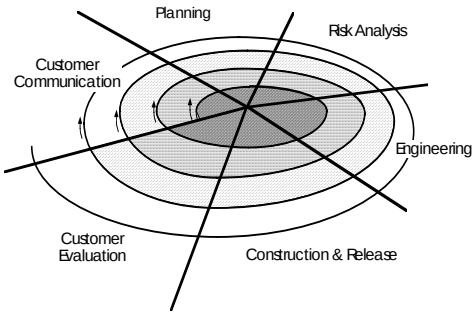
Model inkeremental menggabungkan elemen-elemen model sekuensial linier (diaplikasikan secara berulang) dengan filosofi prototipe iteratif. Model pertambahan memakai urutan-urutan linier di dalam model yang membingungkan, seiring dengan laju waktu kalender. Model pertambahan berfokus pada penyampaian produk operasional dalam setiap pertambahannya.



Gambar Model Pertambahan

b. Model Spiral

Model spiral (*spiral model*) adalah model proses perangkat lunak yang evolusioner yang merangkai sifat iteratif dari prototipe dengan cara komputer dan aspek sistematis dari model sekuensial linier.

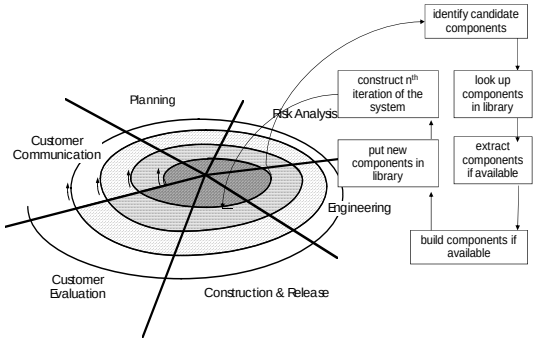


Gambar Model Spiral

c. Model Rakitan Komponen

Model ratikan komponen menggabungkan beberapa karakteristik model spiral. Model ini bersifat evolusioner, sehingga membutuhkan pendekatan iteratif untuk mencapai perangkat lunak. Tetapi model rakitan komponen

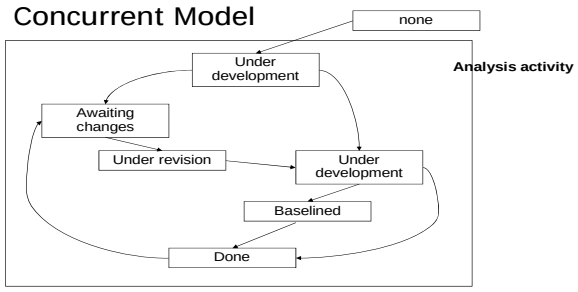
merangkai aplikasi dari komponen perangkat lunak sebelum dipaketkan (kadang-kadang disebut “kelas”).



Gambar Model Rakitan Komponen

d. Model Perkembangan Konkuren

Model proses yang konkuren dapat disajikan secara skematis sebagai sederetan aktivitas teknik mayor, tugas-tugas, dan keadaannya yang lain. Contohnya aktivitas rekayasa yang dibatasi untuk model spiral dipenuhi dengan melakukan tugas-tugas sebagai berikut: prototyping dan atau pemodelan analisis, spesifikasi kebutuhan, dan rancangan.



Gambar Model Perkembangan Konkuren

2.2.3 Konsep Basis Data

2.2.3.1 Definisi

Istilah basis data banyak menimbulkan interpretasi yang berbeda. Anthoni J. Fabbri dan A. Robert Schwab, mendefinisikan basis data sebagai berikut :

> "*Basis data adalah sistem berkas terpadu yang dirancang terutama untuk meminimalkan pengulangan data"*.

Menurut George Tsu-der Chou, basis data dapat didefinisikan sebagai berikut:

> *"Basis data sebagai kumpulan informasi bermanfaat yang diorganisasikan ke dalam tatacara yang khusus"*.

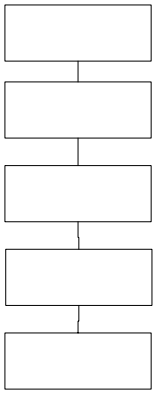
Basis data dimaksudkan untuk mengatasi masalah pada sistem yang memakai pendekatan berbasis berkas. Sistem basis data adalah suatu sistem menyusun dan mengelola *record-record* menggunakan komputer untuk menyimpan atau merekam serta memelihara data opersional lengkap sebuah organisasi/perusahaan sehingga mampu menyediakan informasi yang optimal yang diperlukan pemakai untuk proses mengambil keputusan.

Untuk mengelola basis data diperlukan perangkat lunak yang disebut DBMS. DBMS adalah perangkat lunak sistem yang memungkinkan para pemakai membuat, memelihara, mengontrol, dan mengakses basis data dengan cara yang praktis dan efisien.

Mengapa diperlukan *database* :

1. Salah satu komponen penting dalam sistem informasi, karena merupakan dasar dalam menyediakan informasi.
2. Menentukan kualitas informasi : akurat, tepat pada waktunya danrelevan.
3. Informasi dapat dikatakan bernilai bila manfaatnya lebih efektif dibandingkan dengan biaya mendapatkanya.
4. Mengurangi duplikasi data (*data redudancy*).
5. Hubungan data dapat ditingkatkan.
6. Mengurangi pemborosan tempat simpanan luar.

Sampai dengan membentuk suatu database, data mempunyai jenjang mulai dari karakter-karakter,item data, *record*, *file* dan kemudian *database*. Jenjang data dapat digambarkan sebagai berikut :



Gambar 2.2 Jenjang Data

1. Character : merupakan bagian data yang terkecil, dapat berupa karakter numeric, huruf ataupun karakter-karakter khusus(*special character*) yang mrmbentuk suatu *field.*
2. Field : merepresentasikan suatu atribut dari *record* yang menunjukan suatu item dari data, seperti misalnya nama, alamat dan lain sebagainya. Kumpulan dari *field* membentuk *record.*
3. Record : kumpulan dari *field* membentuk suatu *record*. *Record* menggambarkan suatu unit data individu yang tertentu. Kumpulan dari record membentuk suatu *file*.
4. File : *file* terdiri dari *record-record* yang menggambarkan satu kesatuan data yang sejenis.
5. Database : kumpulan dari *file* yang membentuk suatu database.

2.2.3.2 Tahap Perancangan Basis Data

Perancangan basis data merupakan langkah untuk menentukan basis data yang diharapkan dapat mewakili seluruh kebutuhan pengguna. Perancangan basis data terdiri atas perancangan basis data secara konseptual, perancangan basis data secara logis, dan perancangan basis data secara fisis.

Beberapa komponen yang terdapat pada perancangan basis data secara konseptual antara lain:

1. Entitas

   Entitas terkadang disebut tipe entitas atau kelas entitas. Entitas adalah objek yang dapat dibedakan dari objek-objek lainnya.

2. Atribut

   Atribut adalah item data yang menjadi bagian dari suatu entitas. Istilah lain dari attribut adalah properti.

3. Hubungan

   Hubungan adalah asosiasi atau kaitan antara dua entitas.

4. Kekangan

   Kekangan digunakan untuk melindungi integritas data (misalnya, melindungi kesalahan sewaktu pengisian data).

5. Domain

   Domain adalah himpunan yang berlaku bagi suatu atribut. Kekangan domain mendefinisikan nama, tipe, format, panjang, dan nilai masing-masing item data.

6. Integritas Referensial

   Integritas referensial adalah aturan-aturan yang mengatur hubungan antara kunci primer dengan kunci tamu milik tabel-tabel yang berbeda dalam suatu basis data relasional untuk menjaga konsistensi data.

2.2.4. Metode Analis Sistem Terstruktur

Salah satu pendekatan yang digunakan dalam suatu analisis dan desain adalah pendekatan terstruktur. Suatu pendekatan yang bekerja dari sudut pandang yang lebih tinggi menuju tingkat lebih rendah yang lebih rinci, dimana keinginan pemakai disajikan dalam diagram aliran data. Desain terstruktur adalah implementasi secara fisik dan pembagian struktur modular secara hirarki dengan pendekatan atas bawah. Beberapa alat bantu yang digunakan dalam analisis dan desain tersetruktur antara lain:

2.2.4.1 Entity Relationship Diagram (ERD)

ERD hanya berfokus pada data, dengan menunjukkan “jaringan data” yang ada untuk suatu sistem yang diberikan. ERD sangat berguna bagi aplikasi di mana data dan hubungan yang mengatur data sangatlah kompleks. ERD pada mulanya diusulkan oleh Peter Chen untuk desain sistem *database* relasional dan telah dikembangkan oleh yang lainnya. Serangkaian komponen utama diidentifikasikan untuk ERD : objek data, atribut, hubungan dan berbagai tipe indikator. Tujuan utama dari ERD adalah untuk mewakili objek data dan hubungan mereka.

Kardinalitas model data harus dapat merepresentsikan jumlah peristiwa dari objek di dalam hubungan yang diberikan. Tillmann mendefinisikan kardinalitas dari *object-relationship pair* dengan cara sebagai berikut: kardinalitas merupakan spesifikasi dari sejumlah peristiwa dari satu [objek] yang dapat dihubungkan ke

sejumlah peristiwa dari [objek] yang lain. Dengan mempertimbangkan semua kombinasi dari ‘satu’ dan ‘banyak’, dua [objek] dapat dihubungkan sebagai:

1. Satu-ke-satu (1:1)
2. Satu-ke-banyak (1:N)
3. Banyak-ke-satu (N:1)
4. Banyak-ke-banyak (M:N)

2.2.4.2 Diagram Konteks

Diagram konteks menggambarkan hubungan antara sistem dengan entitas luarnya. Diagram konteks berfungsi sebagai transformasi dari satu proses yang melakukan transformasi data *input* menjadi data *output*. Entitas yang dimaksud adalah entitas yang mempunyai hubungan langsung dengan sistem. Diagram konteks ini merupakan alat-alat untuk struktur analisis. Pendekatan struktur ini mencoba untuk menggambarkan sistem secara garis besar atau secara keseluruhan. Diagram konteks ialah kasus khusus dari DFD atau bagian dari DFD yang berfungsi memetakan modul lingkungan yang direpresentasikan dengan lingkaran yang mewakili keseluruhan sistem.

Suatu diagram konteks selalu mengandung satu dan hanya satu proses saja. Diagram konteks ini menggambarkan hubungan *input* atau *output* antara sistem dengan dunia luarnya (kesatuan luar).

Diagram Kontek meliputi Beberapa sistem antara lain:

1. Kelompok pemakai
2. Data yang diterima oleh sistem dari lingkaran
3. Data yang dihasilkan oleh sistem
4. Penyimpanan data

2.2.4.3 Data Flow Diagram (DFD)

Data Flow Diagram (DFD–DAD/Diagram Alir Data) memperlihatkan hubungan fungsional dari nilai yang dihitung oleh sistem, termasuk nilai masukan, nilai keluaran, serta tempat penyimpanan internal. DAD adalah gambaran grafis yang memperlihatkan aliran data dari sumbernya dalam objek kemudian melewati proses yang mentransformasinya ke tujuan yang lain, yang ada pada objek lain. DAD sering digunakan untuk menggambarkan suatu sistem yang telah ada atau sistem baru yang akan dikembangkan secara logika tanpa mempertimbangan lingkungan fisik dimana data tersebut mengalir. DFD merupakan alat yang digunakan pada metodologi pengembangan sistem yang terstruktur (*structured analysis and design*). DFD merupakan alat yang cukup populer sekarang ini, karena dapat menggambarkan arus data di dalam sistem dengan terstruktur jelas.

Beberapa simbol yang digunakan dalam *Data Flow Diagram* (DFD) antara lain:

1. *External Entity* (kesatuan luar) atau *boundary* (batas sistem)

   Setiap sistem pasti mempunyai batas sistem (*boundary*) yang memisahkan suatu sistem dengan lingkungan luarnya. Sistem akan menerima input dan

menghasilkan output kepada lingkungan luarnya. Kesatuan luar (*external entity*) merupakan kesatuan (*entity*) di lingkungan luar sistem yang dapat berupa orang, organisasi atau sistem lainnya yang berada di lingkungan luarnya yang akan memberikan *input* atau menerima *output* dari sistem.

2. *Data Flow* (arus data)

   Arus data (*data flow*) di DFD diberi simbol suatu panah. Arus data ini mengalir diantara proses (*process*), simpanan data (*data strore*) dan kesatuan luar (*external entity*). Arus data ini menunjukkan arus dari data yang dapat berupa masukan untuk sistem atau hasil dari proses sistem.

3. *Process* (proses)

   Suatu proses adalah kegiatan atau kerja yang dilakukan oleh orang , mesin atau kompuiter dari hasil suatu arus data yang masuk ke dalam proses untuk dihasilkan arus data yang akan keluar dari proses. Untuk *physical data flow diagram* (PDFD), proses dapat dilakukan oleh orang, mesin atau komputer, sedangkan untuk *logical data flow diagram* (LDFD), suatu proses hanya menunjukkan proses dari komputer. Setiap proses harus diberi penjelasan yang lengkap meliputu identifikasi proses, nama proses dam pemroses.

4. *Data Store* (simpanan luar)

   Simpanan data (*data store*) merupakan simpanan dari data yang dapat berupa, yaitu suatu file atau *database* di sistem komputer, suatu arsip atau catatan

manual, suatu kotak tempat data di meja seseorang, suatu tabel acuan manual, dan suatu agenda atau buku.

2.2.4.4 Spesifikasi Proses

Spesifikasi porses adalah proses yang menjelaskan tentang kejadian – kejadian yang terjadi pada buble yang terdapat di level terbawah pada *data flow diagram*.

"Spesifikasi proses menggambarkan kejadian di dalam setiap *buble* pada level terbawah pada *data flow diagram*. Spesifikasi proses mendefinisikan kegiatan yang harus dilakukan untuk mengubah *input* menjadi *output"*.YAH[9]

Contoh Spesifikasi Proses :

No proses	No .proses berdasarkan DFD pada level terbawah
Nama proses	Nama proses berdasarkan DFD pada level terbawah
Deskripsi	Penjelasan singkat tentang proses diatas
Dipanggil oleh proses	Proses diatas dipanggil oleh no proses berapa
Memanggil proses	Proses diatas memanganggil no proses berapa
Flow Input	Masukan aliran data ke proses
Flow Output	Keluaran aliran data ke proses
Store Input	Pengambilan data store oleh suatu proses
Store Output	Penyimpanan data store oleh suatu proses
Logika proses	Berisi narasi/logika setiap proses pada level terbawah

Tabel 2.1 Spesifikasi Proses

2.2.4.5 Kamus Data

Kamus data adalah katalog fakta tentang data dan kebutuhan-kebutuhan informasi dari suatu sistem informasi. Dengan menggunakan kamus data, analisis sistem dapat mendefinisikan data yang mengalir di sistem dengan lengkap. Kamus data dibuat pada tahap analisis sistem dan digunakan baik pada tahap analisis maupun pada tahap perancangan sistem.

Pada tahap analisis dan perancangan, kamus data dapat digunakan sebagai alat komunikasi antara analisis sistem dengan pemakai sistem tentang data yang mengalir di sistem, yaitu tentang data yang masuk ke sistem dana tentang informasi yang dibutuhkan oleh pemakai sistem.

Pada tahap perancangan, kamus data digunakan untuk merancang *input*, merancang laporan-laporan dan *database*. Kamus data dibuat berdasarkan arus data yang ada di DFD. Arus data di DFD sifatnya global, hanya ditunjukkan nama arus datanya saja. Keterangan lebih lanjut tentang struktur data dari arus data di DFD secara lebih rinci dapat dilihat di kamus data. Kamus data harus dapat mencerminkan keterangan yang jelas tentang data yang dicatatnya, maka kamus data harus memuat nama arus data, alias, bentuk data, arus data, penjelasan, periode, volume, dan struktur data.

Konstruktur data	Notasi	Arti
	=	Disusun atas
Berurutan	+	dan
Pilihan	[|]	Baik ini – atau
Pengulangan	$( )^n$	Pengulangan ke-n dari
	( )	Data opsional
	* *	Komentar tidak dibatasi

Tabel 2.2 Notasi *deskripsi* isi untuk kamus data

2.2.5. Teknik Pengujian Perangkat Lunak

Pengujian perangkat lunak adalah elemen kritis dari jaminan kualitas perangkat lunak dan mempesentasikan kajian pokok dari spesifikasi, desain, dan pengkodean.

2.2.5.1 Dasar Pengujian Perangkat Lunak

Pengujian menyajikan anomali yang menarik bagi perekayasa perangkat lunak. Pada proses perangkat lunak, perekayasa pertama-tama berusaha membangun perangkat lunak dari konsep abstrak ke implementasi yang dapat dilihat, baru dilakukan pengujian. Perekayasa menciptakan sederetan *test case* yang dimaksudkan untuk "membongkar" perangkat lunak yang sudah dibangun. Pada dasarnya

pengujian merupakan salah satu langkah dlam proses rekayasa perangkat lunak yang dianggap sebagai hal yang destruktif daripada konstruktif.

2.2.5.2 Sasaran-sasaran Pengujian

Dalam buku klasiknya mengenai pengujian perangkat lunak, Glen Myers menyatakan sejumlah aturan yang berfungsi sebagai sasaran pengujian:

1. Pengujian adalah proses eksekusi suatu program dengan meksud menemukan kesalahan.
2. *Test case* yang baik adalah *test case* yang memiliki probabilitas tinggi untuk menemukan kesalahan yang belum ditemukan sebelumnya.
3. Pengujian yang sukses adalah pengujian yang mengungkap semua kesalahan yang belum pernah ditemukan sebelumnya.

2.2.5.3 Prinsip Pengujian

Sebelum mengaplikasikan metode untuk mendesain *test case* yang efektif, perekayasa harus memahami prinsip dasar yang menuntun pengujian perangkat lunak. mengusulkan serangkaian prinsip-prinsip pengujian diantaranya:

1. Semua pengujian harus dapat ditelusuri sampai kepesyaraan pelanggan.
2. Pengujian harus direncanakan lama sebelum pengujian itu mulai.
3. Prinsip pareto berlaku untuk pengujian perangkat lunak.

4. Pengujian harus mulai dari yang kecil dan yang berkembang kepengujian yang lebih besar.
5. Pengujian yang mendalam tidak mungkin.
6. Untuk menjadi paling efektif pengujian harus dilakukan oleh pihak ketiga yang *independen.*

2.2.5.4 Testabilitas

Testibilitas perangkat lunak adalah seberapa mudah program computer dapat diuji. Karena pengujian sulit, maka perlu diketahui apa yang harus dilakukan agar manjadi lebih mudah. *Cheklist* berikut ini memberikan serangkaian karakteristik yag membawa peragkat lunak yang dapat diuji.

1. *Operabilitas,* "semakin baik dia bekerja, semakin efisien dia diuji".
2. *Observabilitas,* "apa yanganda lihat adalah apa yang anda uji".
3. *Kontrabilitas,* semakin baik kita dapat mengontrol perangkat lunak semakin banyak pengujian yang diotomatisasi dan dioptimalkan".
4. *Dekomposabilitas,* "dengan mengontrol ruang lingkup pengujian, kita dapat lebih cepat mengisolasi masalah dan melakukan pengujian kembali secara lebih halus".
5. Kesederhanaan, "semakin cepat yang diuji, semakin sedikit kita dapat mengujinya".

6. *Stabilitas,* “semakin sedikit perubahan, semakin gagguan dalam pengujian”.
7. *Verifikasi,* mengacu kepada rangkaian aktivitas yang memastikan bahwa perangkat lunak secara tepatmengimplementasikan suatu fungsi terentu.
8. *Validasi,* mengacu pada rangkaian aktivitas berbeda yang memastikan bahwa prangkat lunak yang dibangun dapat ditelusuri kepersyaratan pelanggan. “Apakah kita membangun produk yang benar”.

2.2.5.5 Pengujian *Black Box*

Pengujian *Black-box* berfokus padapersyaratan fungsional perangkat lunak. Dengan demikian, pengujian *black-box* memungkinkan perekayasa peangkat lunak mendapatkan serangkaian kondisi input yang sepenuhnya semua persyaratan fungsional untuk suatu program.

Pengujian *black-box* berusaha menemukan kesalahan dalam kategori sebagai berikut:

1. Fungsi-fungsi yang tidak benar atau hilang.
2. Kesalahan *Interface.*
3. Kesalahan dalam struktur data atau akses *database* eksternal.
4. Kesalahan kinerja.
5. Inisialisasi dan kesalahan terminasi.

2.2.6. Definisi Pengolahan Data

Berikut ini akan dikemukakan definisi dari pengolahan data :

"*Pengolahan data adalah serangkaian operasi atas informasi yang direncanakan guna mencapai tujuan atau hasil yang diinginkan"*.

Tujuan Utama dalam pengolahan data dalam sebuah *database* adalah agar kita dapat memperoleh kembali data (yang kita cari) dengan mudah dan cepat, selain itu pemanfaatan *database* memiliki beberapa tujuan. Secara lengkap pemanfaatan *database* dilakukan untuk memenuhi sejumlah tujuan (*objektif*) antara lain :

1. Kecepatan dan Kemudahan (*Speed*)
2. Efisiensi Ruang (*Space*)
3. Ketersediaan (*Availability*)
4. kelengkapan (*Completely*)
5. keamanan (*Security*)
6. Kebersamaan Pemakai (*Sharability*)

2.2.6.1 Sistem *Client-Server*

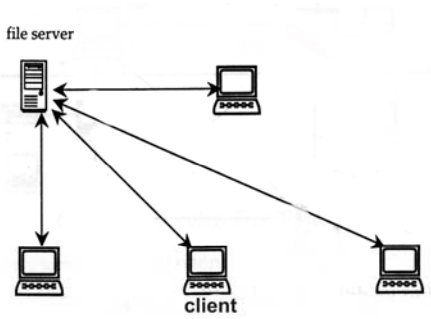
Sistem *Client-Sever* dapat mempunyai dua atau tiga tingkat *setup* (pengaturan). Terdapat sebuah komputer yang berfungsi sebagai *server* (pusat) dan beberapa komputer lainnya yang bersifat sebagai *client*. Beberapa bentuk pelayanan yang dapat diberikan komputer *server* antara lain :

1. Mengontrol hak akses perangkat-perangkat yang ada dalam jaringan.
2. Mengatur keamanan data dalam jaringan

3. Penggunaan perangkat, data dan sistem aplikasi komputer secara bersama.

Sistem *Client- Server* atau disebut juga sistem tersentralisasi diterapkan pada sebuah sistem jaringan. Sistem *Client-Server* ini ditujukan untuk mengatasi kelemahan-kelemahan yang terdapat pada sistem sebelumnya. Sistem *Client- Server* terdiri dari dua komponen utama yaitu *client* dan *server*. *Client* berisi aplikasi basis data dan *server* berisi DBMS dan basis data. Setiap aktivitas yang dikehendaki para pemakai akan lebih dulu ditangani oleh *client*. Bila ada proses yang harus melibatkan data yang tersimpan pada basis data barulah *client* mengadakan hubungan dengan *server*.

Pada sistem *Client-Server* untuk memenuhi kebutuhan, *client* akan mengirimkan *message* (perintah) *query* pengambilan data. Selanjutnya, *server* yang menerima *message* tersebut akan menjalankan *query* tersebut dan hasilnya akan dikirimkan kembali ke *client*. Dengan begitu *transfer* datanya jauh lebih *efisien*. Adapun bentuk dari Sistem *Client-Server* yang sederhana adalah sebagai berikut :



Gambar II.1 Sistem *Client Server* Sederhana

2.2.7 *Software* Pendukung

Software Pendukung yang digunakan dalam pembuatan sistem informasi ini adalah *Visyal Basic.NET* dan *MySQL*.

2.2.7.1 Visual Studio.NET

Program Visual Studio.NET merupakan kumpulan bahasa pemrograman. Dengan Visual Studio.NET kita dapat menggunakan bahasa program yang sudah kita kuasai, tanpa harus belajar dari awal. Visual Studio.NET terdiri dari 4 versi yaitu :

1. Enterprise Architect
2. Enterprise Developer
3. Academic
4. Professional

Secara umum Visual Studio.NET terdiri dari bahasa pemrograman sebagai berikut :

1. Visual Basic.NET
2. Visual C#.NET
3. Visual J#.NET
4. Visual C++.NET

Visual Studio.NET memiliki komponen dasar yang dinamakan NET Framework, yaitu pola kerja pemrograman untuk mengolah dan mengatur Windows Application, Web Application, maupun XML Web Service.

Hampir semua orang, terutama para pemrogram, mengenal dengan baik Visual Basic, yaitu bahasa pemprograman berbasis *Graphical User interface* (GUI) buatan Microsoft yang mampu setiap pekerjaan menjadi mudah dan mampu meningkatkan prodktivirtas pemrogram. Dalam sejarahnya, Visual Basic pernah meraih penghargaan sebagai tool *Rapid Application Development* (RAD) terbaik selama tiga kali. Presentasi ini cukup membuktikan bahwa kemampuan Visual Basic sudah teruji dan di akui oleh banyak kalangan.

Seiring dengan semakin meningkatnya kebutuhan didalam pengembangan perangkat lunak, pengembangan Visual Basic melihat ada beberapa kelemahan pada Visual Basic. Dan ini dapat Anda rasakan sendiri sebagai seorang pemrogram. Bagaimanapun juga, dibalik keunggulan-keunggulan yang dimiliki Visual Basic, terdapat keterbatasan fitur dan kemampuan yang mengakibatkan para pemprogram mengalami kesulitan.

Beberapa kelemahan yang dianggap dapat menimbaulkan masalah antara lain :

1. Tidak memiliki multi-threading, yakni kemampuan yang memingkinkan suatu tugas dapat di jalankan pada thread terpisah

2. Tidak menyediakan fitur-fitur yang lengkap untuk mendukung pemrograman berorientasi objek
3. Kemampuan dalam menangani kesalahan (error handling) yang sangat teterbatas
4. Antar muka User yang kurang efektif untuk pemrograman aplikasi berbasis internet

Visual Studio.NET sendiri menyediakan lingkungan pengembangan high-level untuk membangun aplikasi-aplikasi pada NET Framework. Dilingkungan inilah anda akan merasakan teknologi yang mampu menyederhanakan pembuatan dan penyebaran aplikasi. Selain itu Visual Studio.NET menawarkan generasi baru aplikasi berbasis Windows dengan fitur-fitur yang tersedia melalui.NET Framework

Visual studio.NET sendiri menyediakan lingkungan pengembangan high-level untuk membangun aplikasi-aplikasi pada NET Framework. Dilingkungan inilah ada akan merasakan teknologi yang mampu menyederhanakan pembuatan dan penyebaran apikasi. Seain itu Visual Basic>NET menawarkan generasi baru aplikasi berbasis Windows dengan fitur-fitur yang tersedia melalui NET Framework.

Sebagai salah satu bahasa pemrograman yang terpaket dalam Visual Studio.NET, Visual Basic.NET dapat memanfaatkan semua kemampuan yang telah tersedia. Begitu juga dengan kekurangan-kekurangan dari Visual Basic, semua sudah

tidak ditemukan lagi pada Visual Basic.NET. selain itu, fitur-fitur baru juga akan anda temukan pada Versi ini di antaranya :

1. Fitur-fitur yang lebih lengkap untuk mendukung pemrograman berorientasi objek
2. Kemampuan untuk menangani kesalahan (error handling) yang lebih terstruktur
3. Kemampuan baru untuk bekerja dengan console
4. Dukungan inherent pada XML (Extensible Markup Language) dan web services.
5. Pendekatan aplikasi desktop yang lebih baik menggunakan Windows Forms
6. Pemrograman database yang lelih baik melalui pendekatan ADO.NET
7. IDE (Integrated Development Environment) yang bekerja lebih cepat dari responsip dalam menerima perintah

Sebagai bukti bahwa Visual Basic.NET memiliki kemampuan lebih, adanya dukungan .NET Framework memungkinkan anda untuk mengembangkan berbagai aplikasi sebagai aplikasi Windows, Aplikasi Web, Aplikasi Web Mobile, bahkan sampai pembuatan Web service, menggunakan pemrograman Visual Basic.NET.

Untuk memudahkan para pengembang didalam membangun aplikasi database, Visual Studio.NET juga sudah menyertakan perangkat lunak pembuat laporan, yaitu

Crystal Report. Fitur-fitur tambahkan yang disediakan dapat anda gunakan melalui bahasa pemrograman didalam Visual Studio.NET.

Oleh karena itu bahasa-bahasa pemprograman di dalam Visual Studio.NET menggunakan IDE yang sama, semua fasilitas dapat dipakai secara bersama. Hal ini juga memungkinkan interaksi mudah dilakukan

Seiring perkembangannya, saat ini Visual Basic.NET sudah dimulai memasuki versi ketiga. Versi yang pertama bermula ketika Visual Basic.NET di rilis dan juga di kenal sebagai Visula Basic.NET 2002. Versi kedua adalah Visual Basic.NET 2003, adapun versi terbaru atau ke tiga adalah Visual Basic 2005 dan sekarang sudah ada versi terbaru lagi yaitu Visual Basic 2008

2.2.7.2. MySQL

MySQL adalah *Relational Database Management System* (RDMS) yang didistribusikan secara gratis disebuah lisensi GPL (*General Public License*). Dimana setiap orang bebas untuk menggunakan *MySQL*, namun tidak boleh dijadikan produk turunan yang bersifat *close source* atau komersial. *MySQL* sebenarnya merupakan turunan salah satu konsep utama dalam *database* sejak lama, SQL (*Structured Query Language*) adalah sebuah konsep pengoperasian *database*, terutama untuk pemilihan (seleksi) dan pemasukan data yang memungkinkan pengoperasian data dikerjakan dengan mudah secara otomatis. Keandalan suatu sistem *database* (DBMS) dapat diketahui dengan cara kerja *optimizer*nya dalam melakukan proses perintah-perintah *SQL* yang dibuat oleh user maupun program-program aplikasinya sebagai *database*

server lainnya dalam *query* data. *MySQL* adalah satu dari sekian banyak sistem *database* yang merupakan solusi tepat dalam aplikasi *database*.

Menurut ANSI (*American National Standards Institute*), bahasa ini merupakan standar untuk *relational database menagement sistems* (RDBMS). Pernyataan-pernyataan *SQL* digunakan untuk melakukan beberapa tugas seperti : *update* data pada *database*, atau menampilkan data dari *database*. Beberapa *software* RDBMS dapat menggunakan *SQL*, seperti: *Oracle, Sybase, Microsoft Access, Ingres,* dsb. Setiap *software database* mempunyai bahasa perintah / sintaks yang berbeda, namun pada prinsipnya mempunyai arti dan fungsi yang sama.

2.2.7.3 ODBC (*Open Database Connectivity*)

ODBC adalah sebuah standar yang dikembangkan mocrosoft untuk mendukung driver database yang bersifat *indefendent* terhadap produk, dengan adanya *ODBC* kita dapat menggunakan database yang tidak didukung oleh *BDE* (*Borland Database Engine*). *Driver ODBC* dapat diperoleh dari produsen *database* yang kita pakai. *Driver ODBC* ada yang 16 bit dan ada yang 32 bit. *Delphi* memakai *driver* yang 32 bit. *Microsoft* memproduksi *driver ODBC* untuk *Access, Excell, SQL Server, MySql*, dan lain lain.