BAB II

TINJAUAN PUSTAKA

2.1 Tinjauan Perusahaan

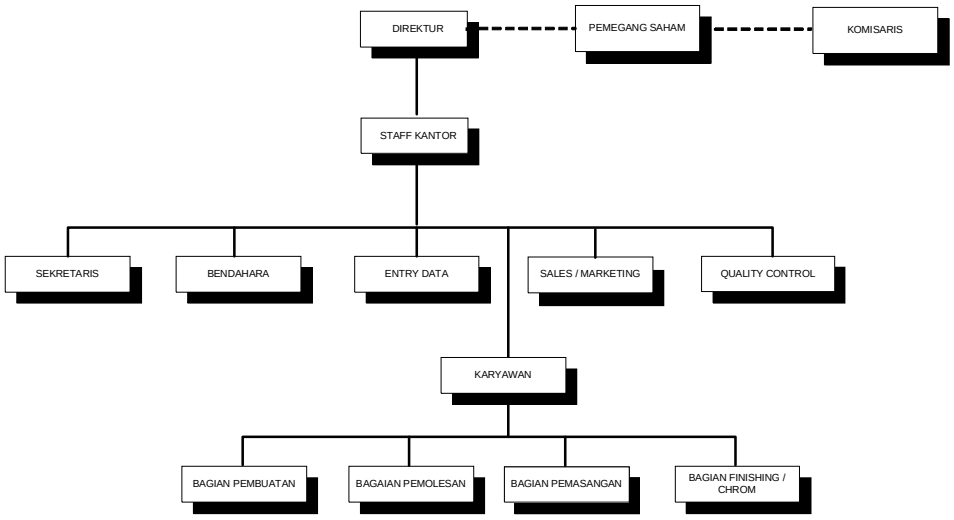
2.1.1 Sejarah Perusahaan

Latar Belakang Usaha dan Kondisi saat ini CV. En-Ty Gold adalah sebuah perusahaan keluarga yang bergerak di bidang Industri Kerajinan Perhiasan, usaha ini telah berjalan sejak tahun 1998, dimana berawal dari Usaha Jasa Tenaga (*Manpower Management and Services*) yang diperoleh dari Toko-toko Emas dan perorangan. Pada awalnya kesempatan untuk mewujudkan keinginan bergerak dalam bidang industri terbuka lebar, dengan bantuan beberapa partner yang menguasai dibidang kerajinan perhiasan, yaitu dari Pembuatan, pemolesan, pemasangan, dan finishing. Dimana pembuatan sebuah barang, dari bahan baku menjadi rangka barang jadi, dipoles dan pemasangan apabila ada yang di pasang, setelah itu finishing. Contoh : Cincin, Giwang, Gelang, Koye/Kalung, Bross dan sebagainya. Barang - barang tersebut adalah barang mewah yang pastinya di sukai semua kalangan, terutama kaum hawa. Penjualan /Sales Bahwa barang-barang tersebut diatas adalah barang yang diminati kaum hawa dan sudah menjadi kebutuhan sebagai investasi yang menguntungkan. Maka kami mendesain dengan bagus, cantik, dan seindah mungkin, untuk disuguhkan ke customer. Pemasaran yang dilakukan dengan cara langsung ke toko-toko perhiasan dan perorangan. Sampai saat ini hubungan kerjasama dengan toko-toko, perorangan masih berjalan dengan lancar dan saling cooperative, mereka memberikan kepercayaan kepada

kami untuk mengadakan pesanan-pesanan baru, terbukti *sales servis / maintenance* berkala sampai saat ini masih tetap berjalan dengan baik. Harga setiap barang bervariasi, tergantung barang apa yang dipesan dan tingkat kesulitannya. Lama pembuatan minimal satu minggu sampai dua minggu. Acuan pembuatan barang setiap pesanan barang, para customer kita sodorkan desain gambar untuk memilih yang sesuai selera pemesan, gambar yang dipilih tersebut untuk dijadikan pedoman pembuatan barang. Pembelian Bahan Baku Bahan baku pembuatan barang ini adalah emas murni, logam mulia dan bahan penunjang lainnya, yang sumber pembelian / pengadaannya ada di pasar, toko dalam negeri dan dari luar negeri.

2.1.2 Struktur Organisasi

Dalam kegiatan sehari-hari yang dilakukan oleh perusahaan CV. En-Ty Gold Jakarta terdapat suatu tingkatan atau *hierarki* kekuasaan yang digambarkan dalam struktur organisasi yang dapat dilihat pada gambar dibawah ini:



Gambar 2.1 Struktur organisasi CV. En-Ty Gold Jakarta

2.2 Landasan Teori

Pada landasan teori ini akan diuraikan sejumlah teori-teori yang membantu dan memecahkan permasalahan yang ada. Beberapa landasan 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

Untuk dapat memahami sistem informasi, akan lebih baik jika konsep dari sistem itu dipahami terlebih dahulu, konsep dasar sistem ini meliputi pengertian dasar sistem, bentuk umum sistem, karakteristik sistem, dan analisis sistem.

2.2.2 Pengertian Dasar Sistem

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

2.2.2.1 Bentuk Umum Sistem

Bentuk umum dari suatu sistem terdiri atas masukan (*input*), proses dan keluaran (*output*), dalam bentuk umum sistem ini bisa melakukan satu atau lebih masukan yang akan diproses dan menghasilkan keluaran sesuai dengan rencana yang telah ditentukan sebelumnya sebelumnya.



Gambar 2.2 Bentuk Umum Sistem

2.2.2.2 Karakteristik Sistem

Adapun karakteristik suatu sistem yaitu:

1. Komponen-Komponen (*Components)*

   Suatu sistem terdiri dari sejumlah komponen yang saling berinteraksi, yang artinya saling bekerja sama untuk membentuk satu kesatuan.

2. Batas Sistem *(Boundary*)

   Batas sistem (*Boundary*) merupakan daerah yang membatasi antara satu sistem dengan sistem yang lain atau dengan lingkungan luarnya.

3. Lingkungan Luar Sistem *(Environment)*

   Lingkungan luar dari satu sistem adalah apapun diluar batas dari sistem yang mempengaruhi operasi sistem.

4. Perhubungan *(Interface)*

   Perhubungan merupakan media penghubung antara satu sub sistem dengan sub sistem lain.

5. Masukan *(Input)*

   Masukan adalah energi yang dimasukan ke dalam sistem.

6. Keluaran *(Output)*

   Keluaran adalah hasil dari energi yang diolah dan diklasifikasikan menjadi keluaran yang berguna.

7. Pengolahan

   Suatu sistem dapat mempunyai sutau bagian pengolahan yang akan merubah masukan menjadi keluaran.

8. Sasaran *(objective)* atau tujuan *(goal)*

   Suatu sistem pasti mempunyai tujuan atau sasaran-sasaran dari sistem sangat menentukan sekali masukan yang dibutuhkan sistem dan keluaran yang akan dihasilkan sistem.

2.2.3 Definisi dan Konsep Dasar Sistem Informasi

Konsep dasar informasi adalah data yang kemudian diolah dengan kriteria tertentu untuk menghasilkan informasi yang dibutuhkan. Data dapat diartikan segala sesuatu yang perlu diolah terlebih dahulu untuk mendapatkan suatu informasi

2.2.3.1 Definisi Sistem Informasi

Informasi merupakan hal yang sangat penting bagi manajemen dalam mengambil suatu keputusan. Suatu informasi dapat diperoleh dari sistem informasi atau juga disebut dengan *Processing systemi* atau *information Processing system* atau *information-generating system.* Sistem informasi didefinisikan oleh Robert A. Leitchdan K. Roscoe Davis adalah "*suatu sistem didalam sebuah organisasi yang mempertemukan kebutuhan pengolahan transaksi harian, mendukung orperasi, bersifat menejerial dan kegiatan strategi dari suatu organisasi dan menyediakan pihak luar tertentu dengan laporan-laporan yang diperlukan"*.

2.2.3.2 Konsep Dasar Sistem Informasi

John Burch dan Gary Grudnitski menerangkan bahwa sistem informasi terdiri dari komponen-komponen yang disebut dengan istilah blok bangunan

(*building block*), yaitu blok masukan (*input block*), blok model (*model block*), blok keluaran (*Output block*), blok teknologi (*technologi*), blok basis data (*database block*), dan blok kendali (*controls block*). Sebagai suatu sistem, keenam blok tersebut masing-masing berinterakasi satu dengan yang lainnya membentuk satu kesatuan untuk mencapai sasarannya.

1. Blok Masukan

   Input mewakili data yang masuk kedalam sistem informasi. Input disini termasuk metode-metode dan media untuk menangkap data yang akan dimasukan, 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 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 digunakan untuk menerima input, menjalankan model, menyimpan dan mengakses data, menghasilkan dan mengirimkan keluaran dan membantu pengendalian dari sistem keseluruhan. Teknologi ini 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 operator komputer, pemrograman, operator pengolahan kata, spesialis telekomunikasi, analia sistem, penyimpanan data dan 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 yang disebut dengan DBMS (*Database Management System*). Data perlu disimpan dalam basis data untuk keperluan penyediaan informasi lebih lanjut. Data didalam basis data perlu di organisasikan sedemikian rupa, supaya informasi yang dihasilkan berkualitas. Organisasi basis data yang baik juga berguna untuk efisiensi kapasitas penyimpanannya.

6. Blok Kendali

   Terdapat banyak hal yang dapat merusak sebuah informasi, misalnya bencana alam, api, temperatur, air, debu, kecurangan-kecurangan, kegagalan-kegagalan sistem itu sendiri, kesalahan-kesalahan, ketidak efisienan, sabutase 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.

2.2.4 Data

Data adalah representasi dari suatu fakta, yang dimodelkan dalam bentuk gambar, kata, dan/atau angka. Manfaat data adalah sebagai satuan representasi yang dapat diingat, direkam, dan dapat diolah menjadi informasi. Karakteristiknya

data bukanlah fakta, namun representasi dari fakta. Kata sederhananya, data adalah catatan tentang fakta, atau data merupakan rekaman catatan tentang fakta. Data yang baik adalah yang sesuai dengan faktanya.

2.2.5 Rekayasa Perangkat Lunak

2.2.5.1 Definisi Rekayasa Perangkat Lunak

Rekayasa perangkat lunak adalah pengembangan dari penggunaan prinsip pengembangan untuk memperoleh perangkat lunak secara otomatis yang reliabel dalam bekerja secara efisien pada mesin nyata.

2.2.5.2 Model Sirkulasi Linier

Model sekuensial linier untuk *software engineering*, sering disebut juga dengan siklus kehidupan klasik atau model air terjun. Model ini mengusulkan sebuah pendekatan kepada perkembangan *software* yang sistematik dan sekuensial yang mulai pada tingkat dan kemajuan sistem pada seluruh analisis, desain, kode, pengujian, dan pemeliharaan. Dimodelkan setelah siklus rekayasa konvensional, model sekuensial linier melingkupi aktivitas–aktivitas sebagai berikut :

1. **Rekayasa dan pemodelan sistem informasi**

Rekayasa dan anasisis sistem menyangkut pengumpulan kebutuhan pada tingkat sistem dengan sejumlah kecil analisis serta desain tingkat puncak. Rekayasa informasi mencakup juga pengumpulan kebutuhan pada tingkat bisnis strategis dan tingkat area bisnis.

2. Analisis kebutuhan *Software*

Proses pengumpulan kebutuhan diintensifkan dan difokuskan, khusunya pada *software*. Untuk memahami sifat program yang dibangun, analis harus memahami domain informasi, tingkah laku, unjuk kerja, dan interface yang diperlukan. Kebutuhan baik untuk sistem maupun *software* didokumentasikan dan dilihat lagi dengan pelanggan.

3. Desain

Desain *software* sebenarnya adalah proses multi langkah yang berfokus pada tempat atribut sebuah program yang berbeda; struktur data, arsitektur *software*, representasi interface, dan detail (algoritma) prosedural. Proses desain menterjemahkan syarat/kebutuhan ke dalam sebuah representasi *software* yang dapat diperkirakan demi kualitas sebelum dimulai pemunculan kode.

4. Generasi Kode

Desain harus diterjemahkan kedalam bentuk mesin yang bisa dibaca. Langkah pembuatan kode melakukan tugas ini. Jika desain dilakukan dengan cara yang lengkap, pembuatan kode dapat diselesaikan secara mekanis.

5. Pengujian

Proses pengujian berfokus pada logika internal *software*, memastikan bahwa semua pernyataan sudah diuji, dan pada eksternal fungsional, yaitu mengarahkan pengujian untuk menemukan kesalahan-kesalahan dan memastikan bahwa *input* yang dibatasi akan memberikan hasil aktual yang sesuai dengan hasil yang dibutuhkan.

6. Perawatan

Software akan mengalami perubahan setelah disampaikan kepada pelanggan (pengecualian yang mungkin adalah *software* yang dilekatkan). Perubahan akan terjadi karena kesalahan-kesalahan ditentukan, karena *software* harus disesuaikan untuk mengakomodasi perubahan-perubahan di dalam lingkungan eksternalnya (contohnya perubahan yang dibutuhkan sebagai akibat dari perangkat peripheral atau sistem operasi yang baru), atau karena pelanggan membutuhkan perkembangan fungsional atau unjuk kerja. Perawatan *software* mengaplikasikan lagi setiap fase program sebelumnya dan tidak membuat yang baru lagi

2.2.6 Basis Data dan Sistem Basis Data

Basis data (*database*) merupakan kumpulan data yang saling berhubungan satu dengan yang lainnya, tersimpan diperangkat keras komputer dan digunakan perangkat lunak untuk memanipulasinya. Penerapan *database* dalam sistem informasi disebut dengan *database* sistem. Sistem basis data (*database system*) adalah sekumpulan subsistem yang terdiri atas basis data dengan pra pemakai yang menggunakan basis data secara bersama-sama, personal yang merancang dan mengelola basis data , serta sistem komputer untuk mendukungnya.

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

2.2.7 Metode Analisis dan Perancangan 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 :

2.2.8 Flow Map

Flow Map adalah diagram yang menunjukan aliran data berupa formulir-formulir ataupun keterangan berupa dokumentasi yang mengalir atau beredar dalam suatu sistem.

Notasi yang digunakan dalam suatu flow map merupakan penggabungan notasi *flow chart* program.

Hal-hal yang harus diperhatikan dalam menggambarkan *flow map* adalah sebagai berikut:

1. Penggambaran *flow map* dimulai dari atas halaman ke bagian bawah, kmudian dari kiri kebagian kanan
2. Penggambaran *flow map* dilakukan berdasarkan pembagian atau sub sistem
3. Dalam *flow map* harus jelas dimana awal suatu status informasi, kemudian dimana akhir silkus

4. Semua bagian siklus informasi harus jelas Menggunakan kertas kerja yang jelas sesuai dengan yang akan dilakukan dalam sistem
5. Semua sub sistem yang digambarkan, harus mengalami siklus informasi
6. Jika penggambaran suatu siklus informasi terpotong, maka gunakan penghuibung antar bagian.

2.2.8.1 Diagram Konteks

Keadaan sistem secara umum dan hubungan-hubungan sistem tersebut dengan komponen-komponen diluar sistem atau sistem yang lain dapat digambarkan secara logika dengan diagram konteks. Definisi diagram konteks adalah penggambaran semua elemen-elemen yang terlibat dalam suatu sistem dan elemen-elemen yang terlibat dalam suatu sistem arus data yang masuk ke dalam sistem dan luar sistem digambarkan dengan jelas.

Diagram konteks meliputi beberapa sistem antara lain:

1. Kelompok pemakai, organisasi atau pihak lain
2. Data yang diterima oleh sistem dari lingkungan
3. Data dihasilkan oleh sistem
4. Penyimpangan data
5. Batasan antara sistem yang dirancang dengan lingkungan.

2.2.8.2 Diagram Alir Data/*Data Flow Diagram*

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. DFD 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. DFD 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.

2.2.8.3 Spesifikasi Proses

Spesifikasi proses 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.

Contoh Spesifikasi Proses :

Tabel 2.1 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

2.2.8.4 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.

2.2.9 Teknik Pengujian Perangkat Lunak

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

2.2.9.1 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.

2.2.9.2 Sasaran-sasaran Pengujian

Dalam buku klasiknya memngenai 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.

2.2.9.3 Prinsip Pengujian

Sebelum mengaplikasikan metode untuk mendesain *test case* yang efektif, perekayasa harus memahami prinsip dasar yang menuntun pengujian perangkat lunak. Davis 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 yanglebih besar.
5. Pengujian yang mendalam tidak mungkin.

6. Untuk menjadi paling efektif pengujian harus dilakukan oleh pihak ketiga yang *independen.*

2.2.9.4 Testabilitas

Testibilitas perangkat lunak adalah seberapa mudah program computer dapat diuji. Karena pengujian sulitk, 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 perangkat lunak yang dibangun dapat ditelusuri kepersyaratan pelanggan. "Apakah kita membangun produk yang benar".

2.2.9.5 Pengujian *Black Box*

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

Penguian *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.

2.2.10 Database SQL Server

SQL Server adalah suatu perangkat lunak *database* relasi (bahasa kerennya *Relational Database Management System* atau RDBMS), seperti halnya *ORACLE, Postgresql*, MS SQL**,** dan sebagainya. *SQL* (singkatan dari *Structured Query Language*) sendiri didefinisikan sebagai suatu sintaks perintah-perintah tertentu atau bahasa (program) yang digunakan untuk mengelola suatu *database*.

SQL Server kini mulai menjauhkan diri dan melangkah jauh lebih ke depan, terutama dengan munculnya versi SQL Server 2000. Menentukan bahasa mana yang terbaik untuk aplikasi database akan bersifat sangat subyektif. Namun, biasanya dukungan akan bahasa SQL *(structure query language)*, kriteria kecepatan, pemakaian memori, mudah tidaknya program, dan daya tampung data

menjadi kriteria utama. Selain keutamaan SQL Server sebagai penampung database cukup besar dan dukungannya terhadap bahasa SQL, SQL server juga memberikan dukungan terhadap sistem database terdistribusi yaitu *client server.*

2.2.11 Borland Delphi

Delphi adalah suatu program berbasis bahasa *Pascal* yang berjalan dalam lingkungan *Windows*. Delphi telah memanfaatkan suatu teknik pemrograman yang disebut RAD yang telah membuat pemrograman menjadi lebih mudah. Delphi adalah suatu bahasa pemrograman yang telah memanfaatkan metode pemrograman *Object Oriented Programming* (OOP).

Fungsi dari elemen-elemen pada delphi adalah :

1. *Object Inspector* : suatu window yang berguna untuk mengatur suatu object baik properti, *events* dan *method*.
2. *Form* : Digunakan sebagai layar/window yang digunakan sebagai lembar kerja kita. Di form-lah semua komponen seperti tombol dan komponen lainnya disimpan.
3. *Window Unit/Source Code* : Window/layar yang berisi perintah-perintah yang akan dieksekusi oleh komputer. Di layar inilah kita mengisikan program-program.
4. *Component Palette* : Layar yang berisikan komponen-komponen yang dipakai dipakai dalam program kita.

Perbedaan konsep pemrograman berbasis objek dengan konsep pemrograman biasa adalah sebagai berikut :

1. Dalam pemrograman biasa, suatu benda hanya memiliki properti (ciri) yang membentuk dirinya.

   Contoh : Objek manusia memiliki nama, tinggi, berat, warna kulit dan lain-lain.

2. Dalam pemrograman berorientasi objek, suatu benda tidak hanya memiliki properti (ciri) tetapi juga memiliki method dan event.

   Contoh: Objek manusia tidak hanya memiliki ***properti*** nama, tinggi, berat, warna kulit tetapi juga mempunyai metode / langkah yang dapat dikerjakan dan juga event (suatu kejadian yang bisa terjadi pada objek).

   a. ***Event*** dapat berupa : Event ketika lapar, ketika haus, ketika ngantuk dan lain-lain.

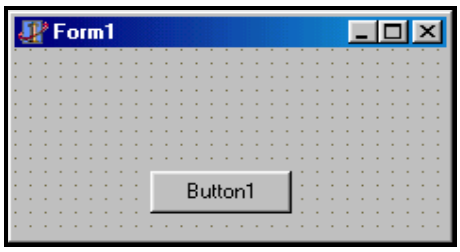
   b. ***Method*** dapat berupa : Makan, minum, tidur.

Suatu *method* bisa saja dipanggil ketika suatu *event* terjadi. Contoh ketika manusia lapar, maka dia akan melakukan method makan. Contoh nyata dalam pemrograman adalah sebagai berikut : Misalkan program yang kita buat memiliki sebuah *form* dan sebuah tombol. Kita dapat memberikan suatu perintah ke masing-masing *event* dari kedua objek tersebut. Misalkan ketika *form* diklik, maka warna dari *form* berganti menjadi merah, dan ketika tombol ditekan maka program akan ditutup.

Untuk membuat program tersebut maka lakukan langkah-langkah sebagai berikut :

1. Buat suatu aplikasi baru dengan mengklik menu ***File*** – ***New Application***. Sehingga akan muncul suatu form kosong yang akan kita beri program.

2. Kemudian tempatkan komponen (objek) *Button* yang ada pada *Component Palette* Standar ke *form*. Caranya : Klik *Button* kemudian geserkan kursor ke *form* dan kemudian klik ditempat yang diinginkan. Lihat gambar di bawah ini.



Gambar 2.3 Contoh program

3. Sekarang klik *form* kemudian klik tab *event* di window ***object inspector*** dan pilih tab *Event* ***onclick***. *Double klick event* tersebut. Maka akan muncul suatu *procedure*/***method*** kosong yang harus diisi sesuai keinginan. Contoh di bawah ini akan mengganti warna (*properti color*) dari *Form1* menjadi warna merah.

   Isilah programnya sebagai berikut :

```
procedure TForm1.FormClick(Sender: TObject);
begin
   Form1.Color:=clred;
End;
```

Keterangan program :

a. Tanda titik (.) adalah tanda pemisah antara objek dengan propertinya.

b. Setiap akhir dari perintah harus diberi perintah titik koma (;)

c. Pengisian nilai ke suatu variabel atau properti menggunakan simbol titik dua sama dengan (:=).

4. Sekarang klik objek tombol kemudian klik tab *event* pada *object inspector* dan pilih *event onclick*. *Double* klik atau tulis nama fungsi yang akan digunakan sebagai *method* yang akan dikerjakan ketika tombol diklik. Kemudian tulis perintah untuk menutup *form* sebagai berikut :

```
procedure TForm1.Button1Click(Sender: TObject);
begin
   Form1.Close;
End;
```

Tekan tombol F9 atau menu ***Run*** – ***Run*** atau tekan tombol  untuk menjalankan program.