BAB II

TINJAUAN PUSTAKA

2.1. Tinjauan Perusahaan

Pada tinjauan perusahaan ini berisi tentang sejarah singkat, tempat dan kedudukan, serta bentuk dan badan hukum perusahaan tempat penelitian tugas akhir. Dalam penyusunan tugas akhir ini, perusahaan tempat penelitian dilakukan di Bengkel Teknik Tohid Mulyono.

2.1.1. Sejarah Singkat Perusahaan

Bengkel Teknik Tohid Mulyono berdiri pada tahun 1990 tepatnya pada tanggal 10 Juni. Bengkel ini bergerak dibidang perdagangan dan perindustrian mesin dengan jenis barang aneka macam logam. Nama Tohid Mulyono sendiri diambil dari nama pemilik bengkel tersebut yaitu Bapak Tohid Mulyono. Bengkel ini semula hanya bengkel kecil biasa. Karena hasil produksi mesin yang dibuat bengkel ini memilik kualitas yang bagus, bengkel ini sedikit demi sedikit mulai berkembang hingga saat ini.

Dalam kegiatan sehari-harinya, bengkel ini membuat mesin yang dipesan oleh perusahaan-perusahaan makanan ringan baik perusahaan kecil maupun besar yang memerlukan bantuan mesin dalam kegiatan operasionalnya. Mesin yang dibuat di bengkel ini, disebut mesin teknik. Selain itu, bengkel ini juga menawarkan jasa reparasi mesin.

Bengkel teknik ini semula terletak di Cibaduyut, Bandung. Kemudian pada akhir tahun 2007 beralih tempat ke daeerah Rancamanyar, Bandung.

2.1.2. Tempat dan Kedudukan Perusahaan

Bengkel Teknik Tohid Mulyono terletak di Jalan Rancamanyar No. 09 Rt.01 Rw.08 Baleendah Bandung 40375.

2.1.3. Bentuk dan Badan Hukum

Bengkel Teknik Tohid Mulyono berdiri sejak dikeluarkannya Surat Izin Usaha Perdagangan (SIUP) Kecil, Nomor : 0543/10-25/PK/VI/1990.

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. Definisi Program Aplikasi

2.2.1.1. Definisi Program

Program adalah ekspresi, pernyataan atau kombinasi yang disusun dan dirangkai menjadi satu kesatuan prosedur yang berupa urutan langkah untuk menyelesaikan masalah yang diimplementasikan dengan menggunakan bahasa pemrograman, sehingga dapat dieksekusi oleh komputer.

2.2.1.2. Definisi Aplikasi

Aplikasi merupakan penerapan, menyimpan sesuatu hal, data, permasalahan, pekerjaan ke dalam suatu sarana atau media yang digunakan untuk menerapkan atau mengimplementasikan hal atau permasalahan tersebut sehingga berubah menjadi suatu bentuk yang baru tanpa menghilangkan nilai-nilai dasar dari hal, data, permasalahan dan pekerjaan. Jadi dalam hal ini hanya bentuk dari tampilan data yang berubah, sedangkan isi yang memuat dalam data tersebut tidak mengalami perubahan.

Jadi Program Aplikasi adalah sederetan kode yang digunakan untuk mengatur komputer supaya dapat melakukan pekerjaan sesuai dengan keinginan *programmer* atau *user*. Seperti program gaji pada suatu perusahaan. Maka program ini hanya digunakan oleh bagian keuangan saja tidak dapat digunakan oleh departemen yang lain. Umumnya program aplikasi ini dibuat oleh seorang programmer komputer sesuai dengan permintaan atau kebutuhan seseorang, lembaga, ataupun perusahaan guna keperluan interennya.

2.2.2. Definisi dan Konsep Dasar Sistem Informasi

2.2.2.1.Definisi Sistem Informasi

Informasi merupakan hal yang sangat penting bagi manajemen dalam mengambil suatu keputusan. Informasi dapat dikatakan sebagai data yang telah terorganisasikan kedalam bentuk yang sesuai dengan kebutuhan. Suatu informasi dapat diperoleh dari sistem informasi atau juga disebut dengan *Processing system* atau *information processing system* atau *information generating system.* Sistem

informasi didefinisikan oleh Robert A. Leitchdan K. Roscoe Davis adalah sebagai berikut : *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.*

Kualitas dari suatu informasi tergantung dari tiga hal, sebagai berikut :

1. Akurat

   Berarti informasi harus bebas dari kesalahan - kesalahan dan tidak bias atau menyesatkan.

2. Tepat pada waktunya

   Berarti informasi yang datang pada penerima tidak boleh terlambat.

3. Relevan

   Berarti informasi tersebut mempunyai manfaat untuk pemakainya.

2.2.2.2.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 dimasukan, 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, analisa 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 diorganisasikan 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.

2.2.3. Data

Data adalah kenyataan yang menggambarkan suatu kejadian - kejadian yang masih mentah dan tidak berarti bagi pemakainya. Sehingga perlu diolah lebih lanjut serta kesatuan nyata (dapat berupa satu objek) yang merupakan sumber informasi. Data dinyatakan dengan nilai (angka, deretan karakter, atau simbol). 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.

2.2.4. Model Proses Sekuensial Linier

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.

2. Model *Prototipe*

   Prototyping paradigma dimulai dengan pengumpulan kebutuhan. Pengembang dan pelanggan bertemu dan mendefinisikan objektif 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*).

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.

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.

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*.

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”).

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.

5. Model Formal

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

2.2.5. Konsep Merancang Basis Data

2.2.5.1.Definisi Basis Data (*Database*)

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.

Menurut C.J.Date, basis data dapat didefinisikan sebagai berikut :

Basis data dapat dianggap sebagai tempat untuk sekumpulan berkas data terkomputerisasi.

Basis data dimaksudkan untuk mengatasi problem 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.

2.2.5.2. Tujuan Basis Data

Tujuan dari basis data erat hubungannya dengan masalah-masalah yang timbul dalam *file* basis data, yaitu:

1. Penyediaan akses yang *fleksibel*, yaitu untuk memberikan kemudahan dalam menampilkan semua informasi yang diperlukan.
2. Pemeliharaan integritas data, basis data berisi *file* yang saling berkaitan yaitu dengan adanya *field* kunci yang menghubungkan kedua *file* tersebut.
3. Keamanan data, basis data dapat mendefinisikan prosedur otoritas untuk memastikan bahwa data hanya dapat diakses oleh *user* yang berhak dan dapat mengijinkan *user* yang berbeda untuk melakukan tiap akses yang berbeda pada tahap data yang sama.
4. Menghilangkan ketergantungan data, Program yang berinteraksi dengan DBMS *(Database Management System)* relatif mandiri terhadap data aktual dalam basis data. Hanya perubahan yang mungkin dilakukan terhadap struktur data tanpa membutuhkan perubahan terhadap program aplikasi yang telah ada.
5. Mengurangi kerangkapan data atau duplikasi data karena data dapat dipakai oleh beberapa program aplikasi.
6. Penggunaan data bersama-sama, data yang sama dapat diakses atau digunakan oleh beberapa *user* pada saat bersamaan. Hal ini menggunakan sistem basis

data, tiap aplikasi mempunyai *file* tersendiri, sehingga suatu data tunggal dalam basis data dapat digunakan untuk beberapa kegunaan.

7. Standarisasi data, yaitu memberikan fasilitas-fasilitas kamus data untuk mendefiniskan nama-nama data secara rinci.

2.2.5.3.Tahap Perancangan Basis Data

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

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.

Pada perancangan model konseptual penekanan tinjauan dilakukan pada struktur data dan relasi antara file. Tidak perlu dipikirkan tentang terapan dan operasi yang akan dilakukan pada *database*. Pendekatan yang dilakukan pada perancangan model konseptual adalah menggunakan model data *relational*. Terdapat dua buah teknik yaitu :

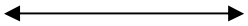
1. Teknik normalisasi
2. Teknik *entity relationship*.

2.2.5.3.1.*Entity Relationship Diagram* (ERD)

Entity Relationship Diagram (ERD) dibuat untuk merancang suatu basis data dengan memperlihatkan hubungan antara *entity* atau objek yang terlibat beserta atributnya. Untuk membantu gambaran relasi secara lengkap ada tiga macam relasi dalam hubungan *attribute* dalam suatu *file* :

1. *One to One Relationship*

   Hubungan antara *file* kesatu dengan *file* kedua adalah satu berbanding satu. Dengan simbol :



Gambar 2.1. One to One Relationship

2. *One to Many Relationship*

   Hubungan antara *file* pertama dengan *file* kedua adalah satu berbanding banyak atau dapat pula dibalik banyak berbanding satu. Dengan simbol :



Gambar 2.2. One to Many Relationship

3. *Many to Many Relationship*

   Hubungan antara *file* pertama dan *file* kedua adalah banyak ke banyak. Dengan simbol :



Gambar 2.3. Many to Many Relationship

2.2.5.3.2.Teknik Normalisasi

Hal-hal yang mendasari proses normalisasi, yaitu prinsip kebergantungan fungsional (*functional dependency*) serta prinsip dekomposisi tanpa kehilangan informasi yang dibutuhkan. Berikut ini tahapan dari teknik normalisasi :

1. Bentuk Normal Kesatu (1NF)

   Bentuk normal yang pertama kali kita akan pelajari adalah bentuk normal ke satu (*first normal form/*1NF*)*. Bentuk normal ke satu ini, tidak seperti bentuk normal, ia tidak membutuhkan informasi tambahan seperti informasi kebergantungan fungsional (*functional dependendency*).

   Ranah nilai (domain) dikatakan *atomic* jika nilai suatu atribut tidak bisa dibagi-bagi lagi menjadi unit-unit yang lebih kecil. Kita katakan sebuah skema relasi, sebut saja R, adalah dalam bentuk normal kesatu (*first normal form/*1NF) jika ranah nilai untuk semua atribut adalah *atomic*. Suatu contoh nilai yang tidak atomic adalah nilai pada atribut alamat sebab atribut ini kadang bisa dibagi lagi menjadi nilai-nilai jalan, kota, serta kode pos. Selain itu, dapat kita katakan juga bahwa atribut komposit adalah atribut yang punya nilai atribut tidak *atomic*.

2. Bentuk Normal Kedua (2NF)

   Suatu relasi adalah dalam bentuk normal kedua (2NF) jika dia berada dalam bentuk normal pertama dan setiap atribut bukan kunci bergantung penuh pada kunci primer. Sehingga, tidak ada atribut bukan kunci yang bergantung pada sebagian (tetapi tidak seluruhnya) kunci primer. Relasi yang berada dalam bentuk normal pertama akan menjadi bentuk normal kedua jika salah satu kondisi dibawah ini terpenuhi :

   a. Kunci primer hanya mengandung satu atribut (seperti atribut NIP dalam relasi karyawan).

b. Tidak ada atribut bukan kunci hadir pada relasi (sehingga semua atribut pada relasi adalah komponen dari kunci primer).
c. Setiap atribut bukan kunci bergantung penuh pada himpunan/sekumpulan atribut yang berfungsi sebagai kunci primer.

3. Bentuk Normal ketiga (3NF)

Relasi adalah dalam bentuk normal ketiga (3NF) adalah jika berada dalam bentuk normal kedua dan tidak dijumpai kebergantungan transitif. Kebergantungan transitif (*transitive dependency*) dalam suatu relasi adalah kebergantungan fungsional antara dua atau lebih atribut bukan kunci.

2.2.6. Metode Arah Terstruktur

2.2.6.1. 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.

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).

2.2.6.2. Diagram Arus 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 dan 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.

2.2.7. 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 analsisis 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.

2.2.8. *Visual Basic 6.0*

2.2.8.1. Sekilas Mengenai *Visual Basic 6.0*

Visual Basic 6.0 merupakan salah satu aplikasi pemrograman visual yang dibuat oleh *Microsoft*. *Visual Basic 6.0* berjalan dalam sistem operasi *Windows* dan tergabung dalam *suite* aplikasi *Microsoft Visual Studio 6.0* yang dikeluarkan pada akhir tahun 1998.

Aplikasi *Visual Basic* mulai diproduksi pertama kali pada tahun 1991. setelah itu munculah versi – versi lanjutan dari *Visual Basic*, yaitu *Visual Basic* 3, 4, 5, dan 6. pada *Visual Basic 4*, dukungan terhadap aplikasi 32 bit mulai diberikan. Versi *Visual Basic* yang terbaru adalah *Visual Basic.NET* yang dirilis pada tahun 2002.

Visual Basic 6.0 terdiri atas tiga buah edisi, yaitu :

1. *Standart Edition*, merupakan produk dasar.
2. *Profesional Edition*, berisi tambahan *Microsoft Jet Data Access Engine* dan pembuatan server *OLE Automation*.
3. *Enterpise Edition*, merupakan edisi untuk membuat program aplikasi *client–server*.

Visual Basic 6.0 menyediakan berbagai perangkat yang dapat digunakan untuk membuat program aplikasi baik aplikasi kecil dan sederhana untuk keperluan sendiri, hingga aplikasi untuk sistem *enterpise* yang besar dan rumit, atau bahkan aplikasi yang dijalankan melalui internet.

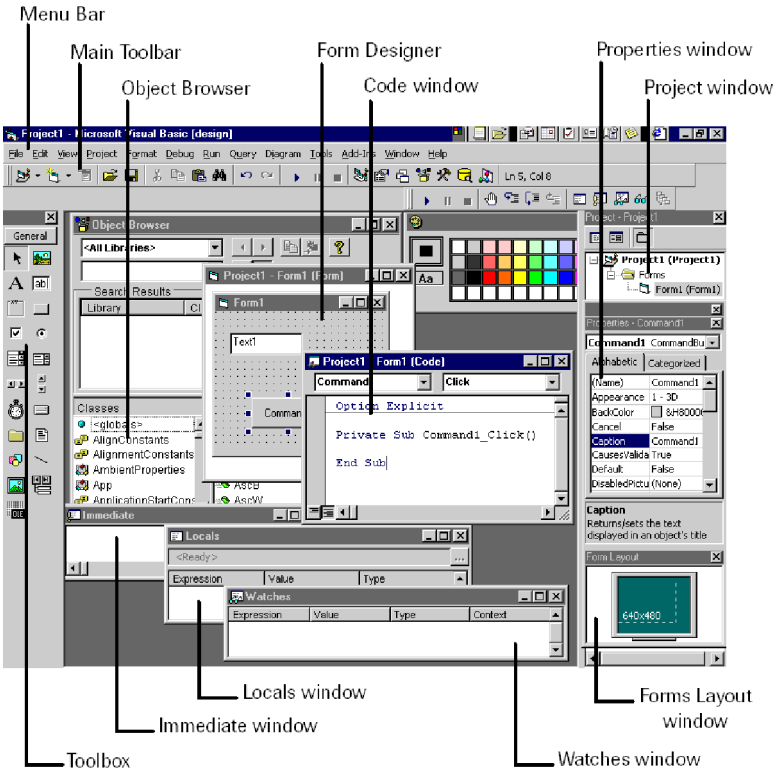
Visual Basic 6.0 memanfaatkan pendekatan visual/GUI (*General User Interface*) dalam proses penggunaanya. Dengan pendekatan GUI, proses pembuatan program aplikasi menjadi lebih mudah dan nyaman.

Basis bahasa pemrograman yang digunakan dalam *Visual Basic 6.0* adalah bahasa BASIC (*Beginners All-Purpose Symbolic Intruction Code*). Bahasa BASIC merupakan salah satu bahasa pemrograman tingkat tinggi yang sederhana dan mudah dipelajari.

Karena dibuat oleh *Microsoft*, *Visual Basic 6.0* memiliki keunggulan dalam hal pengaksesan terhadap beberapa pustaka (*library*) yang dimiliki oleh sistem operasi *Windows*. Para pemrogram (*programmer*) dapat memanfaatkan *Windows* API (*Application Programming Interface*) untuk membuat program aplikasi yang lebih kompleks dan *powerfull*.

2.2.8.2. Antarmuka *Visual Basic 6.0*

Antarmuka yang dimiliki oleh *Visual Basic 6.0* adalah antarmuka yang berupa ruang kerja yang terpadu dan disebut IDE (*Integrated Development Environment*). Antarmuka *Visual Basic 6.0* dapat diatur sesuai selera dan kebutuhan pengguna. Namun, biasanya *Visual Basic 6.0* memiliki tampilan antarmuka seperti gambar dibawah ini.



Gambar 2.4. Antarmuka *Visual Basic 6.0*

Pada gambar diatas terlihat ruang kerja *Visual Basic 6.0* memiliki komponen – komponen sebagai berikut :

1. *Menu Bar*, berisi menu – menu utama yang dimiliki *Visual Basic 6.0*. menu – menu tersebut antara lain menu *File, Edit, View, Project, Run* dan lain – lain. Pada masing – masing menu terdapat beberapa *submenu* yang spesifik.

2. *Main Toolbar*, adalah *toolbar* utama berisi ikon – ikon yang dapat diklik untuk melakukan suatu perintah khusus secara cepat.
3. *Form Designer*, adalah tempat yang digunakan untuk merancang aplikasi yang sedang dibuat. *Form* dapat diibaratkan ’kanvas’ utnuk ’melukis’ *user interface*.
4. *Project Window*, adalah jendela untuk menampilkan *project – project*, *form – form* atau modul – modul yang terlibat dalam proses pembuatan aplikasi.
5. *Properties Window*, adalah jendela untuk menampilkan dan mengubah *properties – properties* yang dimiliki sebuah objek. Ada dua pilihan tampilan, yaitu *Alphabetic* (urut berdasarkan abjad) dan *Categorized* (urut berdasarkan kelompok).
6. *Toolbox* terdiri atas beberapa *class* objek yang dapat digunakan dalam proses pembuatan aplikasi.
7. *Code Window*, adalah jendela yang digunakan untuk menampilkan atau mengeklik kode program. Setiap *Form* dalam *Visual Basic 6.0* memiliki satu buah *Code Window* sendiri.
8. *Form Layout Window*, adalah jendela yang menunjukkan posisi relative form terhadap layer monitor.
9. *Object Browser*, sebuah jendela untuk melihat objek – objek yang terlibat dalam pembuatan aplikasi. Pada jendelan ini kita bisa melihat metode, *event*, atau properti yang dimiliki sebuah objek. Untuk menampilkan *Object Browser* dapat dilakukan dengan cara pada *menu bar* , klik *View* Ò *Object Browser.*

10. *Immediate Window,* adalah jendela yang digunakan untuk melihat dan memeriksa hasil ekspresi secara langsung. Untuk menampilkan jendela *Immediate Window,* pada *menu bar* klik *View* Ò *Immediate Window.*

11. *Locals Window* dan *Watches Window,* biasa digunakan dalam proses *debug* program. *Debug* biasanya dilakukan untuk melihat perubahan nilai suatu variabel atau untuk mengetahui letak kesalahan kode program. *Watches Window,* jendela ini memiliki dua tujuan. Di sini anda secara kontinyu dapat mengawasi nilai dari suatu peubah atau ekspresi dalam program anda. Meliputi peubah global, dimana di luar kemampuan *local window.* Dia juga memberikan anda kemampuan untuk menghentikan eksekusi dari program ketika sebuah ekspresi yang diberikan menjadi *True* atau kapanpun dia merubah nilainya.

2.2.8.3. Memahami Istilah *Object, Property, Method,* dan *Event*

Dalam pemrograman berbasis objek (OOP), anda perlu memahami istilah *object, property*, *method,* dan event sebagai berikut :

1. *Object* : komponen di dalam sebuah program
2. *Property* : karakteristik yang dimiliki object
3. *Method* : aksi yang dapat dilakukan oleh object
4. *Event* : kejadian yang dapat dialami oleh object

Sebagai ilustrasi anggap sebuah mobil sebagai objek yang memiliki properti, *method* dan *event*. Dimana sebuah mobil tersebut memiliki properti yaitu

merk, tipe, dan warna. *Method* yaitu maju, mundur, dan berhenti. Kemudian *Event* sebuah mobil yaitu misalkan pintu dibuka, didorong, atau ditabrak.

Implementasinya dalam sebuah aplikasi misalnya membuat *form*, maka *form* tersebut memiliki properti yaitu *caption, backcolor,* dan lain – lain; *method* yaitu *hide, move, show* dan lainya; dan *event* yaitu *click, load,* dan sebaganya. Sebagaimana pemrograman visual lain seperti *Delphi* dan *Java*, VB juga bersifat *event driven progamming*. Artinya anda dapat menyisipkan kode program pada event yang dimiliki suatu objek.

2.2.8.4. Mengenal Data dan Variabel

Ketika seorang *user* (pengguna) menggunakan sebuah program komputer, seringkali komputer memintanya untuk memberikan informasi. Informasi ini kemudian disimpan atau diolah oleh komputer. Informasi inilah yang disebut dengan DATA.

Visual Basic 6 mengenal beberapa type data, antara lain :

1. *String* adalah tipe data untuk teks (huruf, angka dan tanda baca).
2. *Integer* adalah tipe data untuk angka bulat.
3. *Single* adalah tipe data untuk angka pecahan.
4. *Currency* adalah tipe data untuk angka mata uang.
5. *Date* adalah tipe data untuk tanggal dan jam.
6. *Boolean* adalah tipe data yang bernilai TRUE atau FALSE.

Data yang disimpan di dalam memori komputer membutuhkan sebuah wadah. Wadah inilah yang disebut dengan VARIABEL. Setiap variabel untuk

menyimpan data dengan tipe tertentu membutuhkan alokasi jumlah memori (*byte*) yang berbeda.

Sebuah variabel mempunyai ruang-lingkup (*scope*) dan waktu-hidup (*lifetime*) :

1. Variabel global, adalah variabel yang dapat dikenali oleh seluruh bagian program. Nilai data yang tersimpan didalamnya akan hidup terus selama program berjalan.
2. Variabel lokal, adalah variabel yang hanya dikenali oleh satu bagian program saja. Nilai data yang tersimpan didalamnya hanya hidup selama bagian program tersebut dijalankan.

Variabel yang nilai datanya bersifat tetap dan tidak bisa diubah disebut KONSTANTA.

2.2.9. *Microsoft Access*

2.2.9.1. Sekilas Mengenai *Microsoft Access*

Bila kita mempunyai daftar berisi nama dan alamat mahasiswa berarti kita telah mempunyai *database*. Demikian pula juga kita memiliki buku telepon di rumah, buku itu pun sebuah *database*. Ada banyak hal yang tidak mungkin atau sukar sekali dilakukan secara manual. Disinilah perlunya program komputer seperti *Microsoft Access* untuk mengelola *database*.

2.2.9.2. Jenis Data didalam *Microsoft Access*

Jenis data yang dimiliki *Microsoft Access* adalah sebagai berikut :

1. *Text*

   Jenis data *text* (teks) digunakan untuk menyimpan data seperti nama, alamat, atau bilangan yang tidak membutuhkan perhitungan, seperti nomor telepon, kode pos, atau nomor serial produk.

2. *Memo*

   Jenis data *memo* digunakan untuk menyimpan data teks yang panjangnya lebih dari 225 karakter.

3. *Number*

   Jenis data *number* (bilangan) digunakan untuk menyimpan nilai numerik untuk perhitungan matematis, kecuali perhitungan yang menyangkut uang.

4. *Date/Time*

   Digunakan bagi *field* yang menampung data tanggal atau jam.

5. *Currency*

   Digunakan untuk menampung bilangan. Kelebihan *field* jenis ini dari *field number* adalah bahwa bilangan yang disimpan dalam *field* ini tidak akan kena pembulatan pada saat perhitungan.

6. *AutoNumber*

   Digunakan untuk membuat *field* yang secara otomatis akan memasukkan bilangan pada saat kita menambahkan sebuah *record*.

7. *Yes/No*

   Membuat *field* untuk menampung dua jenis keadaan , ya atau tidak. *Field* ini menyimpan bilangan -1 untuk entri *Yes* dan 0 untuk entri *No*.

8. *OLE Object*

   Membuat *field* untuk menyimpan objek *OLE*, file dokumen, gambar, dan file lainnya yang dibuat dalam program lain.

9. *Hiperlink*

   Teks atau gabungan teks dan bilangan yang disimpan sebagai teks dan digunakan sebagai alamat *hiperlink*.

2.2.9.3. Pembuatan *Database* Baru dalam *Microsoft Access XP*

Di jendela *database*, ada enam tab yang bisa diplih, yaitu *Tables*, *Queries*, *Forms*, *Reports*, *Pages*, *Macros* dan *Modules*.

a. Tables

Untuk menyimpan data yang kita buat.

b. Queries

Select Query adalah jenis *query* yang paling umum. Dengan *select query* bisa didapatkan data dari satu atau beberapa tabel dan menampilkan hasilnya dalam sebuah *datasheet* dimana dapat juga memperbaharui *record - record* dengan beberapa pembatas.

c. Forms

Sebuah *form* membutuhkan sumber data yang berasal dari tabel atau *query*.

d. Report

Report adalah tampilan untuk melihat data pada tabel atau *query*. Berbeda dengan *form* yang biasa menerima masukan data, *report* hanya menampilkan data saja.

e. Pages

Pages adalah tampilan untuk membuka data *access pages* di dalam *microsort access database* atau *microsoft access project* yang digunakan untuk membuka atau mengubah *page* yang dipilih atau membuat *page* baru.

f. Macros

Makros bisa megotomasi beberapa tindakan seperti membuka *form* atau membuka *report*.

g. Modules

Moduls adalah tampilan untuk di dalam *microsoft access database* atau *microsoft access project* yang digunakan untuk mengubah dengan memilih modul atau membuat modul baru.

10. *Lookup Wizard*

Membuat *field* yang menampilkan salah satu dari dua macam daftar yang memudahkan pemasukan data. Kedua macam daftar itu adalah :

a. Daftar *Lookup* yang menampilakn nilai-nilai yang dirujuk dari sebuah tabel atau *query* yang telah ada.
b. Daftar nilai yang menampilkan kumpulan nilai yang tidak bisa diubah yang kita masukkan saat kita membuat *field* ini.

2.2.10. Kebutuhan Bahan

Dalam pekerjaan pembuatan mesin selain faktor tenaga kerja dan peralatan, yang tidak kalah pentingnya adalah faktor bahan yang digunakan. Bahan yang digunakan disini sangat berpengaruh terhadap hasil pekerjaannya. Dengan bahan yang sesuai akan memberikan hasil yang sesuai dengan yang diharapkan. Maka dari itu bahan harus diperiksa terlebih dahulu, apakah sudah sesuai dengan kebutuhan atau belum, dalam hal ini baik mutu maupun jumlahnya. Bahan atau material ialah besarnya jumlah bahan yang dibutuhkan untuk menyelesaikan bagian pekerjaan dalam satu kesatuan pekerjaan.

Kebutuhan bahan diperlukan untuk mendapatkan volume masing-masing pekerjaan dengan satuan yang beragam agar keluar harga yang dibutuhkan. Untuk menghitung kebutuhan bahan perlu dilakukan upaya untuk menilai atau memperkirakan suatu nilai melalui analisis perhitungan dan berlandaskan pada pengalaman. Perhitungan kebutuhan ini bertujuan untuk menghitung semua kebutuhan dalam suatu pekerjaan dan dinilai dalam rupiah agar tidak terjadi pengeluaran yang besar.