BAB II

TINJAUAN PUSTAKA

Bab ini menjelaskan mengenai tempat dilakukannya penelitian dan sejarah perusahaan sampai dengan struktur organisasi perusahaan serta menjelaskan mengenai teori-teori yang akan digunakan dalam membangun sistem

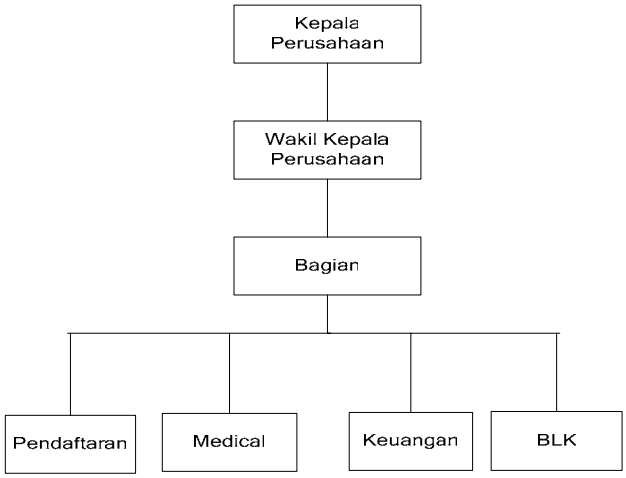
2.1 Tinjauan Perusahaan

2.1.1 Sejarah Berdirinya Dinas Tata Ruang dan Permukiman

Apa itu PJTKI, Perusahaan Jasa Tenaga Kerja Indonesia yang selanjutnya disingkat PJTKI adalah Badan Usaha yang berbentuk Perseroan Terbatas atau Koperasi dan berusaha dibidang jasa penempatan tenaga kerja Indonesia ke luar negeri. Tenaga Kerja Indonesia adalah warga Negara Indonesia baik laki-laki maupun perempuan yang berkerja di luar negeri dalam jangka waktu tertentu berdasarkan perjanjian kerja.

Dengan diberikannya surat izin usaha penempatan Perusahaan Jasa Tenaga Kerja Indonesia (SIUP-PJTKI) pada tanggal 10 Februari 2003 dengan nomor 10/SP/YBA/II/03 untuk melaksanakan penempatan tenaga kerja Indonesia ke luar negeri kepada **PT Youmba Biba Abadi** yang beralamat Jl. Raya Ciangsana RT.03 RW 06 Lt.1 A Pabuaran Wetan Gunung Putri Cibubur Bogor 16968 Telp (021) 84934100 sampai sekarang masih berjalan.

2.1.2 Struktur Organisasi



Gambar 2.1 struktur organisasi

2.2 Landasan Teori

2.2.1 Sistem Informasi

2.2.1.1 Definisi dan Konsep Dasar Sistem Informasi

Telah diketahui bahwa informasi merupakan hal yang sangat penting bagi manajemen di dalam pengambilan keputusan. Pertanyaannya adalah dari mana informasi tersebut didapatkan? Informasi dapat diperoleh dari sistem informasi atau disebut juga dengan *processing system* atau *information processing system* atau *information-generating system*. Sistem informasi didefinisikan oleh Robert A. Leitchdan K. Roscoe Davis adalah suatu sistem di dalam suatu 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 diperlukan.[1]

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 (*technology block*), blok basis data (*database block*), dan blok kendali (*controls block*). Sebagai suatu sistem, keenam blok tersebut masing-masing berinteraksi satu dengan yang lainnya membentuk dengan 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 input, menjalankan model, menyimpan

dan mengakses data,menghasilkan dan mengirimkan keluaran dan membantu pengendalian dari sistem cara keseluruhan. Teknologi terdiri dari 3 bagian utama, yaitu teknisi (*humanware atau brainware*), perangkat lunak (*software*) dan perangakat keras (*hardware*). Teknisi dapat berupa orang-orang yang mengetahui teknologi dan membuatnya dapat beroperasi. Misalnya teknisi adalah operator komputer, pemrograman, operator pengolah kata, spesialis telekomunikasi, analis sistem, penyimpan 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. Data perlu di simpan didalam basis data untuk keperluan penyediaan informasi lebih lanjut. Data didalam basis data perlu di organisasikan sedeimikian rupa, supaya informasi yang dihasilkan berkualitas. Organisasi basis data yang baik juga berguna untuk efisiensi kapasitas penyimpannya. 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 informasi, seperti 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.[1]

2.2.2 Model Proses Perangkat Lunak

Model-model proses perangkat lunak terdiri dari :

1. Linear Sequential Model

Model ini sering disebut model klasik atau *waterfall*. Model ini menyarankan pendekatan pengembangan secara sekuen dan sistematik untuk pengembangan perangkat lunak dimulai di level sistem, berlanjut ke analisis, lalu perancangan, pemrograman, pengujian dan pemeliharaan. Model ini merupakan model yang tertua.[2]

Kelemahan model ini adalah:

1. Proyek-proyek nyata jarang mengikuti alur sekuen yang diusulkan model. Meskipun linear model dapat mengakomodasikan iterasi, namun model melakukan secara tidak langsung. Sebagai hasilnya, perubahan-perubahan dapat menyebabkan kebingungan saat tim pengembangan melakukannya.
2. Sering *customer* sulit menyatakan semua kebutuhannya secara eksplisit. Model ini memerlukan pernyataan eksplisit itu dan sulit mengakomodasi ketidakpastian yang terdapat di awal dari kebanyakan proyek.
3. *Customer* harus memiliki kesabarann. *Versi* yang dapat bekerja dari program tidak akan tersedia sampai akhir dari proyek. Kesalahan besar utama, jika tidak terdeteksi sampai pada program kerja dikaji ulang, maka kesalahan itu dapat mengakibatkan program sama sekali tidak dapat digunakan. Menyiapkan banyak sumber daya.

Alasan kelemahan model ini adalah:

1. Kebutuhan harus telah ditetapkan di awal di siklus hidup.
2. Kebutuhan divalidasi terlalu lambat.

Meskipun terdapat kelemahan, namun model ini memberikan daftar lengkap jumlah aktivitas minimal yang perlu terdapat di model pengembangan perangkat lunak. Umumnya, model-model pengembangan yang lain merupakan superset dari model pengembangan model *waterfall*, memuat aktivitas-aktivitas yang lebih banyak daripada model *waterfall* ini.

2. Prototyping Model

Seringkali pemesan mendefinisikan kumpulan sasaran untuk diakomodasi oleh perangkat lunak tanpa dapat mengidentifikasi rincian-rincian masukan, pengolahan dan keluaran yang diharapkannya. Model *Prototype* cocok untuk situasi ini.

Pada model ini dimulai dengan pengumpulan informasi mengenai kebutuhan, dimana :

1. Pengembang dan pemesan bertemu dan mendefinisikan sasaran-sasaran umum
2. Mengidentifikasikan kebutuhan yang telah diketahui, dan
3. Mencari bidang-bidang yang masih memerlukan pendefinisian.

Setelah itu pengembang melakukan “perancangan kilat” terhadap kebutuhan yang telah teridentifikasi pada pertemuan. Perancangan berfokus pada representasi yang tampak oleh pengguna. Perancangan ini menuntun pembangunan *Prototype* perangkat lunak yang akan diberikan pembeli/pemakai. *Prototype* itu dievaluasi olah pemakai dan digunakan sebagai landasan untuk memperbaiki spesifikasi kebutuhan. Proses ini berulang sampai *Prototype* yang dikembangkan memenuhi kebutuhan

pemakai, dan pengembang telah memahami permasalahan dengan lebih baik.[2]

Variasi model ini adalah :

1. *Throw away prototyping*

   Prototype benar-benar hanya digunakan untuk pengumpulan kebutuhan-kebutuhan pemakai, dibuang kemudian dibuat perangkat lunak yang akan benar-benar digunakan di operasi sehari-hari.

2. *Dead prototyping*

   Prototype yang *versi* terakhir merupakan produk perangkat lunak yang benar-benar dipasang dan digunakan di operasi sehari-hari.

3. *Rapid Application Development (RAD) Model*

   RAD adalah proses pengembangan perangkat lunak yang semakin meningkat (*incremental*) yang menekankan pada siklus pengembangan yang cepat. Model RAD merupakan adaptasi “berkecepatan tinggi” dari *linear sequential model* dimana pembangunan berbasis pada komponen.

   RAD harus didukung oleh perangkat dan lingkungan pengembangan yang memadai, biasanya dikembangkan berdasarkan orientasi komponen. Lingkungan pengembangan telah memiliki pustaka yang luar biasa besar lengkap seperti Java Development Kit beserta seluruh pustaka dari Sun dan vendor-vendor lain pendukung, *Visual Basic* beserta lingkungan *Microsoft* yang luar biasa dasyat, *Borland Delphi* berikut pustakanya.

   Pemrograman biasanya berupa pembuatan antarmuka dan kode perekat untuk merekatkan beragam komponen-komponen yang telah terdapat di lingkungan pemrograman itu.

4. *Evolutionary Software Process Model*

Model evolusi adalah model perulangan. Model ini dicirikan dengan pengembang mengembangkan *versi-versi* sistem yang semakin lebih lengkap. Model telah mempertimbangkan unutk mengakomodasikan evolusi proyek secara lengkap.

Model ini terdiri dari

1. *Incremental Model*

   Model ini mengkombinasikan antara *linear sequential* model dengan filosofi iteratif pada *prototyping*. Pada masing-masing sekuen linear menghasilkan perangkat lunak yang semakin meningkat kompleksitasnya.

2. *Spiral Model*

   Model ini menggabungkan antara sifat alami iterasi dari *prototyping* dengan aspek sistematik dan terkendali dari *linear sequential* model. Model ini memberi peluang unutk pengembangan cepat.

3. *WINWIN Spiral Model*

   Model spiral menyarankan aktivitas komunikasi dengan pembeli. Sasaran aktivitas ini untuk memperoleh deskripsi kebutuhan pemakai. Sering pengembang dan pembeli terlibat negosiasi, pembeli harus menyeimbangkan antara fungsionalitas, kinerja dan karakteristik lain seperti ongkos dan waktu. Negosiasi terbaik adalah *WIN-WIN* dimana pembeli memperoleh kebutuhan mayoritasnya dan pengembang mengembangkan perangkat lunak yang realistik, di dalam anggaran yang tepat dan terpenuhi *deadline*.

4. *Concurrent Development Model*

   Model ini biasa digunakan untuk pengembangan sistem *client/server*.[2]

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 ini.

> "Basis data sebagai kumpulan informasi bermanfaat yang diorganisasikan ke dalam tatacara yang khusus".[3]

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.

2.2.3.2 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.[3]

Beberapa komponen yanng 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 Perancangan Sistem

2.2.4.1 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. Proses ini mewakili proses dari seluruh sistem. Diagram konteks ini menggambarkan hubungan *input* atau *output* antara sistem dengan dunia luarnya (kesatuan luar).

2.2.4.2 *Entity- Relationship Diagram*

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 merepresentasikan 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)

2.2.4.3 *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. 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 komputer 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 atua 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.

2.2.4.4 Kamus Data

Kamus data (KD) atau *data dictionary* (DD) atau disebut juga dengan istilah *system data dictionary* 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 dan 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.[1]

2.2.5 *Borland Delphi*

Dengan adanya bahasa pemrograman yang semakin pesat dan dengan seiring perkembangan ilmu pengetahuan dan teknologi yang semakin canggih, banyak perusahaan besar *software* yang mengembangkan bahasa pemrograman menjadi sebuah *software,* yaitu pemrograman yang memberikan kemudahan yang memberikan kemudahan kepada pengguna *software* untuk mendesain sebuah aplikasi yang berorientasi pada *database* maupun *server*. Salah satu bahasa pemrograman yaitu Borland Delphi yang telah meluncurkan berbagai versi yang menawarkan kemudahan dengan fasilitas yang lengkap.

Karena Delphi merupakan generasi lanjutan dari Turbo Pascal yang telah terkenal sebagai bahasa pemrograman yang terstruktur yang diluncurkan oleh *Borland International Incoorporation* pada tahun 1983. Sesuai dengan sistem operasi yang popular pada saat itu Turbo Pascal memang dirancang untuk dijalankan pada sistem operasi DOS, seiring dengan perkembangan zaman dan ilmu pengetahuan *Borland International Incoorporation* memilih Turbo Pascal for Windows, dalam versi berjalan pada windows 3.X.

Penggabungan Turbo Pascal dengan Turbo Pascal for windows pada tahun 1992 melahirkan suatu bahasa pemrograman yang dinamakan Borland Pascal versi 7.0, tetapi masih terdengar banyak keluhannya mengenai cara penggunaan bahasa pemrograman sehingga mendorong pihak Borland untuk membuat bahasa pemrograman *visual* yang berbasis bahasa pascal, maka lahirlah Borland Delphi

pada tahun 1995 dan kemudian disusul setahun kemudian lahirlah Borland Delphi versi 2.0.

Pada tahun 1997 lahirlah Borland Delphi versi 3.0 yang mempunyai kemampuan yang semakin bertambah terutama dalam merancang aplikasi untuk *multi user* dan *database* yang semakin canggih. Dua tahun kemudian diluncurkan kembali Borland Delphi versi 4.0 yang lebih canggih dari versi sebelumnya. Tetapi Borland Delphi versi 4.0 masih perlu dikembangkan kembali agar mempermudah dalam pengoperasiannya.

Meskipun begitu Borland Delphi memiliki kelebihan dibandingkan dengan aplikasi pemrograman visual berbasis windows yang lain diantaranya :

1. Borland Delphi menyediakan fasilitas yang luas mulai dari fungsi membuat *form* hingga untuk menggunakan format *file* berbasis data yang popular seperti Dbase dengan Paradoks.
2. Dalam Borland Delphi template aplikasi dan *template* format yang dapat digunakan untuk membuat semua aplikasi dengan lebih cepat.
3. Borland Delphi dapat diatur sesuai sesuai dengan kebutuhan dan menunjukkan bagaimana memiliki Borland Delphi dilingkungan dengan pekerjaan lebih produktif.
4. Program terkompilasi dan *windows* menyatakan dapat mengkompilasi program tetapi sebenarnya hanya dapat mengkompilasi sebagai dan kemudian menghubungkan *interpreter* dan program kode dalam sebuah *file* sehingga didapat eksekusi yang lambat. Didalam Delphi sebaliknya program yang kecil dapat diserahkan dalam bentuk *file exe,* tanpa harus menyertakan *file.*

Kelebihan dalam menggunakan Delphi 5.0 yaitu Delphi 5.0 merupakan bahasa pemrograman dengan fasilitas-fasilitas yang menjadikannya memiliki struktur dan format yang lebih efisien dan efektif untuk pemrograman sehingga dapat dengan mudah membuat suatu aplikasi yang sesuai dengan apa yang kita butuhkan.

2.2.6 *Microsoft SQL 2000 Server*

Microsoft SQL 2000 *Server* merupakan program aplikasi *database* untuk *client – server*. Program ini dapat beroperasi pada berbagai sistem operasi, mulai dari windows 98, Windows 2000 dengan macam–macam versinya dan juga dengan Windows NT.

Microsoft SQL 2000 Server merupakan sebuah program *Relational Database Management System* (RDBMS) yang menyediakan dukungan untuk mengorganisasikan data dengan cara menyimpan data ke dalam tabel *database*. Microsoft SQL Server adalah sistem manajemen bisnis data yang memakai perintah-perintah *Transact-SQL* untuk mengirimkan perintah dari komputer *client* ke komputer *server*. *Transact-SQL* adalah bahasa SQL yang dikembangkan oleh Microsoft dengan menambahkan dialek-dialek tertentu. Microsoft SQL Server berisi database, mesin database, dan aplikasi yang diperlukan untuk mengelola data dan komponen-komponennya**.**