BAB II

TINJAUAN PUSTAKA

2.1 Tinjauan Perusahaan

2.1.1 Sejarah Berdirinya KPSBU Lembang

KPSBU Lembang (Koperasi Peternak Sapi Bandung Utara), adalah koperasi susu primer tunggal di kecamatan Lembang, yang merupakan suatu wadah bagi para peternak sapi perah. Wilayah kerja KPSBU Lembang meliputi wilayah Bandung Utara dan Subang Selatan. Daerah Lembang memliliki data geografis ketinggian ± 1.200 meter dari permukaan laut, temperatur antara 17-25 ºC dan curah hujannya sekitar 1.800 - 2.500 mm/tahun yang tentunya sangat cocok untuk daerah peternaksn sapi perah.

KPSBU Lembang terletak di komplek pasar panorama Lembang dengan menempati lahan seluas 1.800 m², yang dibagi atas 400 m² untuk bagian produksi, 600 m² digunakan untuk produk pakan jadi atau makanan konsentrat, 400 m² digunakan untuk gudang bahan pollard dan dedak, 400 m² digunakan untuk perkantoran dan gudang .

KPSBU Lembang, didirikan pada tahun 1971, dan mendapat pengesahan Badan Hukum No.4891/BH/DK-10/20 tanggal 8 Agustus 1971. KPSBU lahir ditengah gejolak para peternak akibat adanya para kolektor atau pengumpul susu yang sudah banyak menguasai pemasaran dalam bidang susu di Lembang. Mulai saat itulah suatu pemikiran atau gagasan yang disertai rasa kebersamaan dan kebulatan tekad untuk mendirikan suatu koperasi, sehingga KPSBU Lembang ini lahir dari kehendak para peternak, oleh peternak dan untuk peternak. Rasa

kebersamaan dan kekeluargaan para peternak itu diwujudkan dengan timbulnya respon dari peternak yang saat itu berjumlah 35 orang peternak yang masuk menjadi anggota KPSBU Lembang sebagai anggota perintis.

Atas dasar musyawarah dan rapat anggota maka dibentuklah susunan pengurus sebagai berikut :

Ketua : R. Soebiantoro

Wakil Ketua : Kasim

Sekretaris : H.G Sutika

Bendahara : Ny Arwani S

Pembantu : H. Syamsudin

Badan hukum yang berlaku saat ini dengan No.4891/BH/PAD/KWK-10/X tertanggal 5 oktober 1995. Dengan rencana kerja sebagai berikut :

a. Mengadakan pembinaan terhadap para peternak yang sudah tergabung dalam Koperasi Peternak Sapi Bandung Utara (KPSBU),
b. Memberikan pengertian dan penyuluhan tentang pentingnya berkoperasi khususnya kepada peternak yang belum menjadi anggota,
c. Menyediakan kebutuhan pokok untuk para peternak dan ternaknya,
d. Melakukan penampungan produksi susu para anggota dan memasarkannya,
e. Memberikan penyuluhan cara meningkatkan kualitas dan kuantitas produksi,
f. Menyediakan ahli dan mantri untuk penyuluhan dan pelayanan kesehatan ternak.

Logo Perusahaan :



Gambar 2.1 Logo KPSBU Lembang

2.1.2 Struktur Organisasi

Struktur organisasi ada pada lampiran.

2.1.3 Visi, Misi dan Nilai KPSBU Lembang

Visi KPSBU Lembang :

Menjadi koperasi susu terdepan di Indonesia dalam mensejahterakan anggota.

Misi KPSBU Lembang :

a. Mensejahterakan anggota melalui layanan prima dalam industri persusuan dengan manajemen yang berkomitmen.
b. Meningkatkan kapasitas kelembagaan koperasi melalui pendidikan, pemberdayaan SDM dan kemitraan strategis.

Nilai-nilai KPSBU Lembang :

Inovatif, dinamis, berorentasi pada kualitas, keterbukaan, keadilan, demokratis, dan mandiri.

2.1.4 Uraian Tugas

Struktur organisasi yang ada di KPSBU Lembang sesuai dengan struktur organisasi yang berlaku di Indonesia dengan berpegang teguh pada undang – undang pokok perkoperasian nomor 25 Tahun 1992. Adapun uraian tugasnya adalah sebagai berikut :

1. Rapat Anggota Tahunan (RAT)

Rapat Anggota Tahunan merupakan perangkat organisasi koperasi yang berfungsi:

a. Mengesahkan laporan pertanggungjawaban Pengurus dan Pengawas
b. Memilih dan memberhentikan Pengurus dan Pengawas
c. Mengesahkan program kerja dan Anggaran Pendapatan serta Belanja KPSBU
d. Mengesahkan Anggaran Dasar dan Anggaran Rumah Tangga dan peraturan khusus
e. Menetapkan pembagian dan penggunaan Sisa Hasil Usaha (SHU).

2. Pengurus

Pengurus terpilih merupakan hasil pemilihan yang telah disyahkan dalam Rapat Anggota Tahunan (RAT) untuk memimpin dan menetapkan kebijakan dibidang organisasi. Keberadaan pengurus dalam struktur organisasi KPSBU adalah sebagai berikut :

a. Dipilih dan diberhentikan dalam Rapat Anggota Tahunan (RAT)
b. Masa jabatan pengurus selama 5 tahun dalam 1 (satu) periode dan dapat dipilih kembali sesuai dengan aspirasi anggota

c. Mengajukan program kerja dan rencana anggaran kepada Rapat Anggota Tahunan (RAT)
d. Menetapkan kebijakan dibidang organisasi, manajemen dan usaha koperasi
e. Melaporkan pelaksanaan Program Kerja berikut laporan keuangan kepada Rapat Anggota Tahunan (RAT)
f. Menyelenggarakan pendidikan bagi anggota
g. Bertanggung jawab terhadap jalannya organisasi dan usaha koperasi
h. Melaksanakan program kerja yang sudah ditetapkan dalam Rapat Anggota Tahunan (RAT).

3. Pengawas

Pengawas terpilih merupakan hasil pemilihan yang telah disyahkan dalam Rapat Anggota Tahunan untuk mengawasi kebijakan yang dikeluarkan/diputuskan oleh Pengurus. Keberadaan Pengawas dalam struktur organisasi KPSBU adalah sebagai berikut :

a. Dipilih dan diberhentikan dalam Rapat Anggota Tahunan (RAT)
b. Masa jabatan pengawas selama 5 tahun dalam 1 (satu) periode dan dapat dipilih kembali sesuai dengan aspirasi anggota.
c. Melaporkan hasil pengawasan kepada Rapat Anggota Tahunan (RAT)
d. Mengawasi pelaksanaan kebijakan Pengurus
e. Mengawasi dan memeriksa pembukuan KPSBU
f. Mengawasi jalannya usaha dan organisasi KPSBU.

4. Anggota

Anggota koperasi merupakan anggota pro aktif dan memenuhi kewajiban sebagai anggota, sedangkan ketentuan mengenai keanggotaan telah ditetapkan dalam Anggaran Dasar (AD) dan Anggaran Rumah Tangga (ART) KPSBU yang telah syahkan dalam Rapat Anggota Tahunan (RAT). Keberadaan anggota dalam Struktur organisasi KPSBU adalah sebagai berikut:

a. Memanfaatkan pelayanan usaha koperasi
b. Membayar simpanan pokok sebagai anggota tetap
c. Mengemukakan pendapat dalam Rapat Anggota Tahunan (RAT)
d. Memilih dan dipilih sebagai Pengurus atau Pengawas
e. Ikut serta menanggung resiko sesuai dengan Anggaran Dasar (AD) dan Anggaran Rumah Tangga (ART).

5. Manajer

Posisi Manajer dalam organisasi koperasi merupakan pengelola yang diangkat oleh pengurus untuk mengatur organisasi sesuai dengan fungsi manajerial. Keberadaan manajer dalam struktur organisasi KPSBU adalah sebagai berikut:

a. Mengatur pelayanan dan usaha kepada Anggota
b. Mengangkat karyawan atas persetujuan Pengurus
c. Dipilih dan diberhentikan oleh Pengurus
d. Bertanggung jawab kepada Pengurus
e. Menangggung kerugian usaha koperasi karena kelalaian.

6. Kepala Unit Pertokoan

Jabatan Kepala Unit dalam struktur organisasi KPSBU merupakan pelaksana, baik dalam bentuk pelayanan maupun usaha KPSBU. Posisi Kepala Unit harus mampu mengatur para stafnya sesuai dengan bidangnya dan keberadaan dalam struktur organisasi KPSBU adalah sebagai berikut :

a. Diangkat dan diturunkan jabatannya oleh Manajer atas persetujuan Pengurus
b. Sebagai pelaksana pelayanan dan usaha
c. Mengatur Sub Unit sesuai dengan bidangnya
d. Bertanggung jawab kepada Manajer KPSBU.

Adapun tugas Kepala Unit Pertokoan yaitu :

a. Membuat sistem pengembangan usaha agar volume penjualan meningkat.
b. Menganalisa dan mengevaluasi usaha pertokoan.
c. Mengupayakan harga penjualan barang pertokoan bersaing.
d. Mendelegasikan dan mengawasi tugas staff.
e. Membuat rencana anggaran dan biaya unit pertokoan.
f. Menjaga keamanan kerja di lingkungan unit pertokoan demi kelancaran aktifitas kerja.
g. Membuat laporan bulanan, triwulan dan tahunan.

7. Staf Pertokoan

Jabatan Staf pada struktur organisasi KPSBU merupakan pelaksana dalam bentuk pelayanan dan usaha KPSBU, Keberadaan dalam Struktur Organisasi KPSBU adalah sebagai berikut :

a. Diangkat oleh Manajer atas persetujuan Pengurus
b. Diberhentikan dari kekaryawanan oleh Manajer atas persetujuan Pengurus
c. Sebagai pelaksana pelayanan dan usaha

Adapun tugas staff Pertokoan yaitu :

a. Melayani anggota dalam pemesanan dan pembelian barang pertokoan
b. Menerima dan menganalisa pengajuan form pengajuan barang dari anggota.
c. Membuat faktur penjualan.
d. Membuat laporan harian, bulanan, triwulan dan tahunan.

8. Korwil (Kordinator Wilayah)

Bertanggung jawab atas ketepatan, keamanan dan kelancaran pembagian uang dan pendistribusian barang pertokoan ke anggota.

2.2 Landasan Teori

2.2.1 Definisi Penjualan

Menurut kamus besar bahasa Indonesia, definisi penjualan adalah proses atau kegiatan menjual barang, proses mengadakan taransaksi jual barang [478].

2.2.2 Definisi dan Konsep Dasar Sistem Informasi

Dalam membahas sistem informasi tentu tidak lepas dari data dan informasi. Data merupakan bentuk yang masih mentah, yang belum dapat bercerita banyak, sehingga perlu diolah lebih lanjut. Pengolahannya dilakukan melalui suatu sistem untuk menghasilkan informasi. Data yang telah menjadi informasi kemudian diterima oleh penerima, selanjutnya penerima membuat suatu keputusan dan melakukan tindakan, hasil dari tindakan tersebut akan menghasilkan data kembali dan membentuk suatu siklus yang disebut dengan siklus informasi.

Menurut Gordon B. Davis sistem dapat berupa abstrak maupun fisik. Sebuah sistem abstrak adalah suatu susunan teratur gagasan atau konsepsi yang saling tergantung. Sebuah sistem fisik lebih lanjut dapat didefinisikan melalui contoh seperti sistem peredran darah, sistem transportasi, sistem persenjataan, sistem komputer dan lain-lain [67].

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*. Menurut Azhar Susanto informasi adalah " hasil dari pengolahan data yang memberikan arti dan manfaat" [2]. Menurut Gordon B.Davis informasi adalah data yang doproses menjadi bentuk yang

berguna bagi pemakai, dan mempunyai nilai pikir yang nyata bagi pembuatan keputusan pada saat sedang berjalan untuk prospek masa depan [3].

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]

2.2.3 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]

2.2.4 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:

a. 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.
b. 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.

c. *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:

a. Kebutuhan harus telah ditetapkan di awal di siklus hidup.
b. 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 :

a. Pengembang dan pemesan bertemu dan mendefinisikan sasaran-sasaran umum
b. Mengidentifikasikan kebutuhan yang telah diketahui, dan
c. 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 :

a. *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.

b. *Dead prototyping*

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

c. *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.

d. *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]

2.2.5 Konsep Basis Data

2.2.5.1 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".[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.

2.2.5.2 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.

2.2.6 Metode Perancangan Sistem

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

2.2.6.2 *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)

2.2.6.3 *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.

2.2.6.4 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]

2.2.7 Sistem Client Server

Client adalah sembarang sistem atau proses yang melakukan sesuatu permintaan data atau layanan ke *server*. *Server* adalah sistem atau proses yang menyediakan data atau layanan yang diminta oleh *client*. Secara fisik, sebuah server dapat berupa komputer (*mainframe*, *mini-frame*, *work station*, ataupun PC atau peranti yang lain (misalnya *printe*r).

Client mempunyai kemampuan untuk melakukan proses sendiri. Ketika sebuah *client* meminta suatu data ke *server*, *server* akan segera menanggapinya dengan memberikan data yang diminta ke *client* bersangkutan. Setelah data diterima, *client* segera melakukan pemrosesan.

Keuntungan arsitektur *client/server* menurut majalah *Byte*, Juni 1993 (Turban, 1995) dapat dilihat pada tabel berikut :

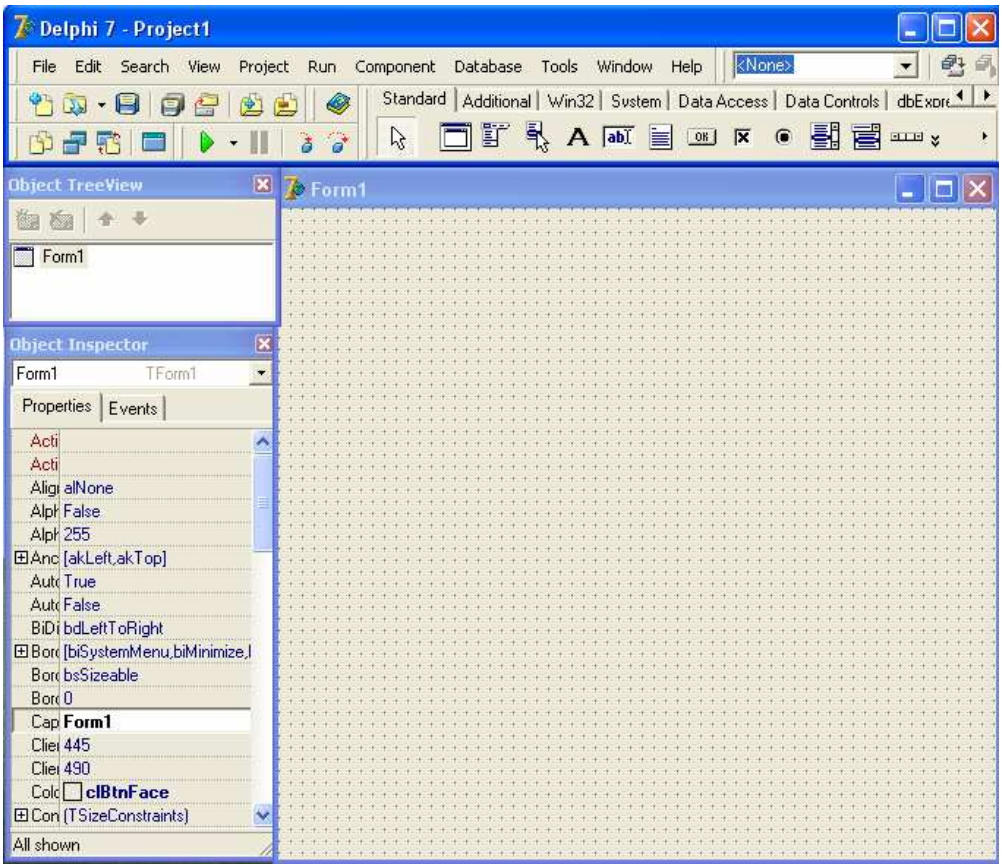
Tabel.2.1 Fitur *Client/server*

Fitur	Keuntungan
Jaringan mesin-mesin yang kecil tetapi berdaya guna.	Jika sebuah mesin macet, bisnis tetap berjalan.
Kumpulan komputer dengan ribuan MIPS (Million Instruction Per Second).	Sistem memberikan kekuatan dalam melaksanakan suatu tugas tanpa memonopoli sumber-sumber daya. Pemakai akhir diberi hak untuk bekerja secara lokal.
Beberapa workstation sangat handal seperti mainframe, tetapi dengan biaya 90 % lebih rendah.	Dengan memberikan kekuatan yang lebih untuk biaya yang kecil, sistem menawarkan keluwesan untuk melakukan pembelian pada hal-hal lain atau untuk meningkatkan keuntungan.
Sistem terbuka	User bisa memilih perangkat keras, perangkat lunak, dan layanan dari berbagai vendor.
Sistem tumbuh dengan mudah dan dapat diperluas secara tak terbatas	Sangat mudah untuk memperbaharui sistem saat kebutuhan berubah
Lingkungan operasi klien yang bersifat individual.	User dapat mencampur dan mencocokan flatform komputer yang sesuai dngan kebuthan masing-masing departemen dan pemakai.

2.2.8 Sekilas Pemrograman delphi7.0

Bahasa pemrograman di Delphi disebut bahasa prosedural artinya bahasa/sintaknya mengikuti urutan tertentu/prosedur. Ada jenis pemrograman non prosedural seperti pemrograman untuk kecerdasan buatan seperti bahasa prolog.Delphi merupakan keluarga visual sekelas Visual Basic atau Visual C, artinya perintah-perintah untuk membuat objak dapat dilakukan secara visual. Pemrogram tinggal memilih objek apa yang ingin dimasukan ke dalam *form/window*, lalu tingkah laku objek tersebut saat menerima *event* / aksi tinggal dibuat programnya. Delphi merupakan bahasa berorentasi objek, artinya nama objek, properti, dan metode / prosedur dikemas menjadi satu kemasan (*encapsulate*).

Delphi adalah sebuah perangkat lunak (Bahasa pemrograman) untuk membuat program / aplikasi komputer berbasis *Windows*. Delphi merupakan bahasa pemrograman berbasis objek, artinya semua komponen yang ada merupakan objek-objek. Ciri sebuah objek adalah memiliki nama, properti dan metode / prosedur.delphi disebut juga *visual programming* artinya komponen-komponen yang ada tidak hanya berupa teks (yang sebenarnya program kecil) tetapi muncul berupa gambar gambar.



Gambar 2.2 *Interface* Delphi

Berikut ini adalah IDE yang ada pada Borland Delphi 7.0 :

1. *Projek*

Projek adalah sekumpulan modul. Jadi *projek* adalah program aplikasi itu sendiri. *Projek* disimpan dalam file berakhiran .DFR. File ini menyimpan seluruh komponen program, termasuk pilihan objek, pilihan *Environment*, pilihan *file EXE* dan segala sesuatu yang berhubungan dengan *projek.*

2. *Form Designer*

Form Designer adalah suatu objek yang dipakai sebagai tempat bekerja dalam aplikasi. *Form* berbentuk jendela dan dapat di batangkan sebagai kertas atau meja kerja yang dapat dilukisi atau diletakan ke dalamnya objek-objek lain.

3. *Objek inspector*

Objek Inspector adalah kotak alat yang berisi *icon-icon* untuk memasukksn objek tertentu ke dalam jendela *form*. Ketika anda mulai membuat suatu projek, Delphi 7.0 akan otomatis menyediakan *icon-icon* objek yang sering dipakai, sesuai kategori aplikasi yang anda buat.

4. *Properties*

Properties digunakan untuk menentukan properti suatu objek. Suatu objek biasanya mempunyai beberapa properti yang dapat diatur langsung dari jendela *properties* atau lewat kode program. Setting properti akan menentukan cara kerja dari objek yang bersangkutan saat program aplikasi dijalankan, misalnya menentukan warna, bingkai, pengambilan data dan lain-lain.

5. *Code editor*

Code Editor adalah fasilitas yang disediakan oleh delphi untuk mengetikan kode program. Kode program ini akan mengontrol dan menentukan jalannya suatu objek.

6. *Event*

Event adalah peristiwa atau kejadian yang diterima oleh suatu objek, misalnya klik seret, tunjuk dan lain-lain.

2.2.8.1 Dukungan Basis Data pada Borland delphi 7.0

Borland delphi 7 menyediakan empat metode akses data (*data Acces*) yang masing-masing mempunyai kelebihan dan kekurangan, yaitu :

1. BDE (*Borland database Engine*)

BDE merupakan metode akses data tertua di delphi, telah hadir sejak delphi versi 1.0 yang berjalan di *windows* 3.1. Keunggulan BDE terletak dari dukungan *driver* untuk berkomunikasi dengan banyak perangkat lunak managemen basis data seperti Paradox, dBaseIII, SQL Server, Acces, dan Interbase. Bahkan BDE Menyediakan koneksi melalui ODBC (*Open Database Connectivity*).

2. ADO (*ActiveX Data Object*)

Merupakan kumpulan *driver* yang disediakan oleh Microsoft untuk mengakses data pada perangkat lunak managemen basis data milik Microsoft.

3. IBX (*Interbase Express*)

Sesuai dengan namanya merupakan metode akses data cepat (*express*) untuk pengembangan aplikasi basis data *client-server* menggunakan Interbase sebagai perangkat lunak untuk manajemen basis data.

4. dbExpress

Tambahan metode akses data mulai Delphi 6. dbExpress menyediakan dukungan akses data terhadap Interbase, MS SQL Server, MySQL, Postgress, sybase, dan DB2.

2.2.8.2 Koneksi Delphi dengan MySQL

Telah diketahui bahwa BDE (Borland Database Engine) adalah salah satu teknologi akses data yang sudah sangat populer di Delphi. Hampir semua pemrogram basis data di Delphi telah menggunakan BDE.

Hingga saat ini BDE belum dapat mengakses server MySQL secara langsung seperti yang dapat dilakukan pada interbase. BDE dapat mengakses server SQL seperti Ms. SQL Server dengan memanfaatkan *driver* ODBC buatan Microsoft, tetapi *driver* tersebut tidak dapat mengakses basis data MySQL.

dbExpress merupakan komponen yang disediakan oleh delphi untuk membangaun aplikasi *client-server*. dbExpress tidak mendukung *software database desktop* yang menjadi fokus BDE dan ADO. dbExpress menyediakan dukungan akses data terhadap interbase. MS SQL Server, Oracle, My SQL, Postgress, Sybase, dan DB2. Borland menganjurkan agar aplikasi basis data *cilent server* dibuat menggunakan dbExpress. dbExpress juga dijadikan metode akses data pada Borland Kylix yang berjalan di Linux sehingga aplikasi berbasis dbExpress di *Windows* juga dapat dikompilasi dan dijalankan di Linux.