BAB II

TINJAUAN PUSTAKA

2.1 Tinjauan Perusahaan

Pada tinjauan perusahaan ini akan diuraikan mengenai sejarah berdirinya perusahaan, struktur organisasi perusahaan dan tugas dari masing-masing *staff* apotek.

2.1.1 Sejarah Singkat Perusahaan

Apotek Linggar Sari merupakan salah satu badan usaha yang berdiri di bawah naungan CV Linggar Kencana. CV Linggar Kencana ini sudah berdiri sejak tahun 1995 yang diprakarsai oleh Bapak Dudung. Selain menaungi Apotek Linggar Sari, CV Linggar Kencana ini juga membawahi beberapa bidang usaha lain seperti klinik Dokter Gigi, Klinik Dokter Umum dan beberapa bidang usaha lainnya yang bergerak di bidang niaga.

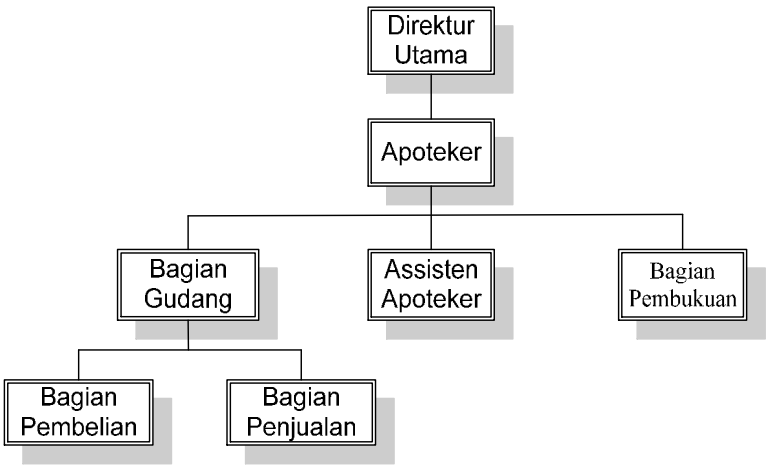
Apotek Linggar Sari ini didirikan oleh Bapak Agus Andri, S.Si, Apt. Setelah beliau mendapatkan surat izin kerja dengan No./SPNO: KP.01.01.V.5.2.1005 tanggal 15 Maret 2000 kemudian beliau mengajukan surat permohonan untuk mendirikan apotek di bawah naungan CV Linggar Kencana, akhirnya apotek Linggar Sari ini didirikan pada pertengahan tahun 2000 tepatnya pada tanggal 22 Juni 2000 dengan surat izin No. 1815/SIA/JB-B/2000.

Adapun tujuan utama dari Apotek Linggar Sari ini yaitu melayani masyarakat di bidang kesehatan dengan menyediakan berbagai macam obat-

obatan baik itu obat-obatan bebas (non resep) maupun obat-obatan yang disertai resep dokter.

2.1.2 Struktur Organisasi Perusahaan

Adapun struktur organisasi dari Apotek Linggar Sari yang sedang berjalan saat ini yaitu sebagai berikut:



Gambar 2.1 Struktur Organisasi Apotek Linggar Sari

2.1.3 Deskripsi Kerja

Adapun uraian dan tugas dari tiap bagian dimana tempat pelaksanaan penelitian dilakukan yaitu apotek Linggar Sari secara keseluruhan adalah :

1. Direktur Utama/Pimpinan

   Direktur utama merupakan pimpinan perusahaan yang berperan sebagai penyandang dana/modal apotek. Tugas pokok dari direktur utama ini adalah mengevaluasi kinerja apotek berdasarkan dari laporan-laporan yang diterimanya dalam kurun waktu tertentu.

2. Kepala Apotek/Apoteker

   Kepala apotek/apoteker merupakan seseorang yang mempunyai izin pendirian apotek. Tugas pokok dari seorang kepala apotek/apoteker adalah:

   a. Menciptakan sistem dan prosedur di bagian unit usaha apotek.
   b. Mengawasi jalannya operasional apotek.
   c. Mengevaluasi apotek berdasarkan dari laporan yang diterima dalam kurun waktu tertentu.

3. Asisten Apoteker

   Asisten apoteker merupakan orang yang bertugas sebagai pembuat/peracik obat sesuai dengan resep dokter yang diterima dari konsumen. Asisten apoteker ini harus merupakan seseorang yang mempunyai keahlian di bidang farmasi.

4. Bagian Pembukuan

   Bagian pembukuan merupakan bagian dari apotek yang bertugas untuk mengelola apotek dalam hal audit laba dan rugi apotek.

5. Bagian Gudang

   Bagian gudang merupakan bagian dari *staff* perusahaan dalam hal ini yaitu apotek yang bertanggung jawab atas segala macam kegiatan yang terjadi di gudang. Bagian gudang ini mempunyai beberapa tugas pokok yaitu :

   a. Menerima barang dari penjual atau *supplier*.
   b. Menyimpan barang ke gudang.
   c. Mendistribusikan barang ke bagian penjualan.

d. Membuat laporan stok barang.
e. Melakukan koordinasi dengan bagian pembelian dan bagian penjualan.
f. Menerima barang retur dan mengurus barang retur ke *supplier*.

6. Bagian Pembelian

   Bagian pembelian ini merupakan bagian yang berhubungan langsung dengan bagian gudang dalam hal penyediaan barang/obat. Bagian pembelian ini mempunyai tugas pokok sebagai berikut :

   a. Melakukan pemesanan barang yang sudah minim di gudang.
   b. Melakukan negosiasi harga ke *supplier* dan mencari harga yang sesuai atau kompetitif, sehingga dapat menaikkan *margin* dari penjualan barang.
   c. Membuat laporan pembelian barang.

7. Bagian Penjualan (Kasir)

   Bagian penjualan (kasir) ini merupakan bagian dari apotek yang berinteraksi secara langsung dengan konsumen dan juga dengan bagian gudang. Bagian penjualan ini mempunyai beberapa tugas pokok diantaranya yaitu :

   a. Mengenal produk atau barang yang dijual secara baik.
   b. Melayani konsumen dengan menerima permintaan obat resep maupun non resep.
   c. Melakukan pemberitahuan kepada bagian pembelian bahwa obat yang ada di apotek telah habis.
   d. Membuat laporan penjualan barang.

2.2 Landasan Teori

Pada landasan teori ini akan dibahas materi-materi yang berhubungan dengan pembangunan sistem informasi pengolahan data barang/obat ini.

2.2.1 Konsep Dasar Sistem

2.2.1.1 Pengertian Sistem

Terdapat dua kelompok pendekatan di dalam mendefinisikan sistem yaitu yang menekankan pada prosedurnya dan yang menekankan pada komponen atau elemennya.[1]

Sistem menurut pendekatan prosedur adalah *suatu jaringan kerja dari prosedur-prosedur yang saling berhubungan berkumpul bersama-sama untuk melakukan kegiatan atau untuk menyelesaikan suatu tujuan tertentu*. Sistem menurut pendekatan elemen atau komponennya adalah *kumpulan dari elemen-elemen yang saling berinteraksi untuk mencapai suatu tujuan tertentu*.

2.2.1.2 Karakteristik Sistem

Suatu sistem mempunyai karakteristik atau sifat-sifat tertentu, yaitu:

1. Komponen-komponen (*Components*)

   Suatu sistem terdiri dari sejumlah komponen yang saling berinteraksi, yang artinya saling bekerja sama membentuk satu kesatuan. Komponen-komponen sistem dapat berupa suatu sub sistem atau bagian dari sistem.

2. Batas (*Boundary*)

   Batas sistem merupakan daerah yang membatasi antara suatu sistem dengan sistem yang lainnya atau dengan lingkungan luarnya. Batas suatu sistem menunjukkan ruang lingkup (*scope*) dari sistem tersebut.

3. Lingkungan Luar (*Environments*)

   Lingkungan luar sistem adalah apapun diluar batas dari sistem yang mempengaruhi operasi sistem. Lingkungan luar sistem dapat bersifat menguntungkan dan juga dapat bersifat merugikan sistem tersebut. Yang menguntungkan merupakan energi dari sistem yang harus tetap dijaga dan dipelihara, sedangkan yang merugikan sistem harus ditahan dan dikendalikan, kalau tidak maka akan mengganggu dari kelangsungan hidup dari sistem tersebut.

4. Penghubung (*Interface*)

   Penghubung suatu sistem merupakan media penghubung antara satu sub sistem dengan sub sistem yang lainnya. *Output* dari suatu sub sistem akan menjadi input bagi sub sistem yang lainnya dengan melalui penghubung sistem ini.

5. Masukan (*Input*)

   Input adalah energi yang dimasukkan ke dalam sistem. Input bisa berupa input perawatan (*maintenance input*) dan input sinyal (*signal input*). *Maintenance input* adalah energi yang dimasukkan supaya sistem tersebut dapat beroperasi, sedangkan *signal input* adalah energi yang diproses untuk mendapatkan keluaran. Sebagai contoh di alam sistem komputer, program adalah *maintenance input* yang digunakan untuk mengoperasikan komputernya dan data adalah *signal input* untuk diolah menjadi informasi.

6. Keluaran (*Output*)

   Output adalah hasil dari energi yang diolah dan diklasifikasikan menjadi keluaran yang berguna dan sisa pembuangan. *Output* dapat merupakan masukan untuk sub sistem yang lain atau kepada supra sistem.

7. Pengolah (*Process*)

   Suatu sistem dapat mempunyai suatu bagian pengolah yang akan merubah input menjadi output.

8. Sasaran (*Objective*) dan Tujuan (*Goal*)

   Suatu sistem pasti mempunyai tujuan atau sasaran. Sasaran dari sistem sangat menentukan sekali input yang dibutuhkan sistem dan output yang akan dihasilkan sistem.

2.2.1.3 Klasifikasi Sistem

Sistem dapat diklasifikasikan dari beberapa sudut pandangan, di antaranya adalah sebagai berikut : [2]

1. Sistem diklasifikasikan sebagai sistem abstrak (*abstract system*) dan sistem fisik (*physical system*). Sistem abstrak adalah sistem yang berupa pemikiran atau ide-ide yang tidak tampak secara fisik.
2. Sistem diklasifikasikan sebagai sistem alamiah *(natural system*) dan sistem buatan manusia (*human made system*). Sistem alamiah adalah sistem yang terjadi melalui proses alam, tidak dibuat manusia. Sistem buatan manusia adalah sistem yang dirancang oleh manusia.
3. Sistem diklasifikasikan sebagai sistem tertentu (*deterministic system*) dan sistem tak tentu (*probabilistic system*). Sistem tertentu beroperasi dengan

tingkah laku yang sudah dapat diprediksi, interaksi diantara bagian-bagiannya dapat dideteksi dengan pasti sehingga keluaran dari sistem dapat diramalkan. Sistem tak tentu adalah sistem yang kondisi masa depannya tidak dapat diprediksi karen amengandung unsur probabilitas.

4. Sistem diklasifikasikan sebagai sistem tertutup (*closed system*) dan sistem terbuka (*open system*). Sistem tertutup merupakan sistem yang tidak berhubungan dengan lingkungan luarnya. Sistem terbuka adalah sistem yang berhubungan dan terpengaruh dengan lingkungan luarnya.

2.2.2 Konsep Dasar Informasi

2.2.2.1 Pengertian Informasi

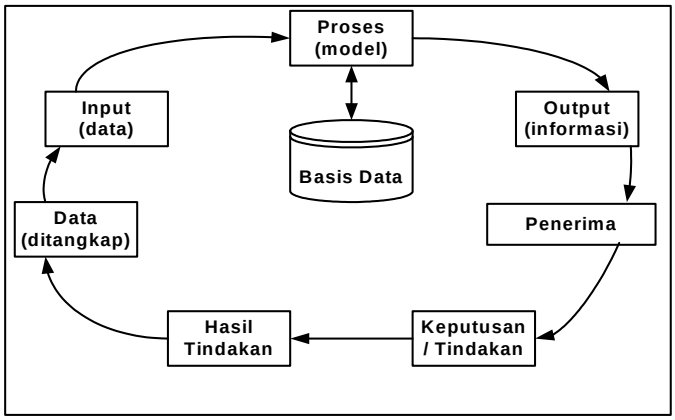
Informasi dapat didefinisikan sebagai *hasil dari pengolahan data dalam suatu bentuk yang lebih berguna dan lebih berati bagi penerimanya yang menggambarkan suatu kejadian-kejadian (event) yang nyata (fact) yang digunakan untuk pengambilan keputusan*.[2]

Sumber dari informasi adalah data. Data adalah kenyataan yang menggambarkan kejadian-kejadian dan kesatuan nyata.[2]

2.2.2.2 Siklus Informasi

Data merupakan bentuk yang masih mentah, belum dapat bercerita banyak, sehingga perlu diolah lebih lanjut. Data yang diolah melalui suatu model menjadi informasi, penerima kemudian menerima informasi tersebut, membuat suatu keputusan dan melakukan tindakan, yang berarti menghasilkan suatu tindakan yang lain yang akan membuat sejumlah data kembali. Data tersebut akan

ditangkap sebagai *input*, diproses kembali lewat suatu model dan seterusnya membentuk suatu siklus. Siklus ini oleh John Burch disebut dengan siklus informasi (*information cycle*) atau ada yang menyebutnya dengan istilah siklus pengolahan data (*data processing cycles*).[2]



Gambar 2.2 Siklus Informasi

2.2.3 Konsep Dasar Sistem Informasi

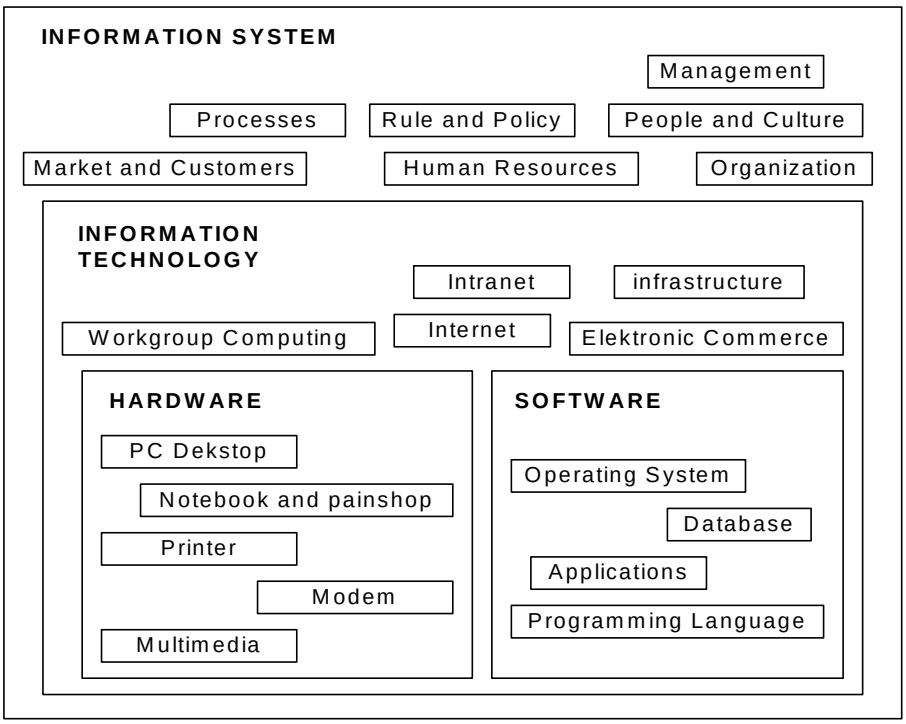
2.2.3.1 Pengertian Sistem Informasi

Informasi dapat didapatkan dari sebuah sistem informasi (*information system)* atau disebut juga dengan *processing systems* atau *information processing systems* atau *information-generating systems*.

Sistem informasi dapat didefinisikan sebagai *suatu sistem didalam suatu organisasi yang merupakan kombinasi dari orang-orang, fasilitas, teknologi, media, prosedur-prosedur dan pengendalian yang ditujukan untuk mendapatkan jalur komunikasi penting, memproses tipe transaksi rutin tertentu, memberi sinyal kepada manajemen dan yang lainnya terhadap kejadian-kejadian internal dan*

eksternal yang penting dan menyediakan suatu dasar informasi untuk pengambilan keputusan yang cerdik.[2]

Sistem informasi merupakan *suatu kumpulan dari komponen-komponen dalam perusahaan atau organisasi yang berhubungan dengan proses penciptaan dan pengaliran informasi*.[3]



Gambar 2.3 Sistem Informasi

2.2.3.2 Komponen Sistem Informasi

Sistem informasi terdiri dari komponen-komponen yang disebut dengan istilah blok bangunan (*building block*), yaitu:

1. Blok Masukan (input block)

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

2. Blok Model (*model block*)

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

3. Blok Keluaran (*output block*)

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

4. Blok Teknologi (*technology block*)

   Teknologi merupakan "kotak alat" (*tool box*) dari pekerjaan sistem informasi. Teknologi terdiri dari 2 bagian utama yaitu perangkat lunak (*software*) dan perangkat keras (*hardware*).

5. Blok Basis Data (*database block*)

   Basis data merupakan kumpulan dari data yang saling berhubungan satu dengan yang lainnya, tersimpan di perangkat keras komputer dan digunakan perangkat lunak untuk memanipulasinya.

6. Blok Kendali (*control block*)

   Blok kendali ini diterapkan agar sistem informasi dapat berjalan sesuai dengan yang diinginkan

2.2.4 Sistem Informasi Manajemen

Sistem informasi manajemen (*management information systems* atau sering dikenal dengan singkatannya MIS) merupakan penerapan sistem informasi

didalam organisasi untuk mendukung informasi-informasi yang dibutuhkan oleh semua tingkatan manajemen.[2]

SIM (Sistem Informasi Manajemen) dapat didefinisikan sebagai *kumpulan dari interaksi sistem-sistem informasi yang bertanggung jawab mengumpulkan dan mengolah data untuk menyediakan informasi yang berguna untuk semua tingkatan manajemen di dalam kegiatan perencanaan dan pengendalian*.[2]

SIM merupakan kumpulan dari sistem-sistem informasi. SIM tergantung dari besar-kecilnya organisasi, terdiri dari sistem-sistem informasi sebagai berikut:

1. Sistem informasi akuntansi (*accounting information systems*), menyediakan informasi dari transaksi keuangan.
2. Sistem informasi pemasaran (*marketing information systems*), menyediakan informasi untuk penjualan, promosi penjualan, kegiatan-kegiatan pemasaran, kegiatan-kegiatan penelitian pasar dan lain sebagainya yang berhubungan dengan pemasaran.
3. Sistem manajemen persediaan (*inventory management information systems*).
4. Sistem informasi personalia (*personnel information systems*).
5. Sistem informasi distribusi (*distribution information systems*).
6. Sistem informasi pembelian (purchasing information systems).
7. Sistem informasi kekayaan (*treasury information systems*).
8. Sistem informasi analisis kredit (*credit analysis information systems*).
9. Sistem informasi penelitian dan pengembangan (*research and development information systems*).

10. Sistem informasi teknik (*engineering information systems*).

2.2.5 Model Proses Perangkat Lunak

Model-model proses perangkat lunak terdiri dari *Linear Sequential Model* dan *Prototyping Model*.

2.2.5.1 *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.[4]

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:

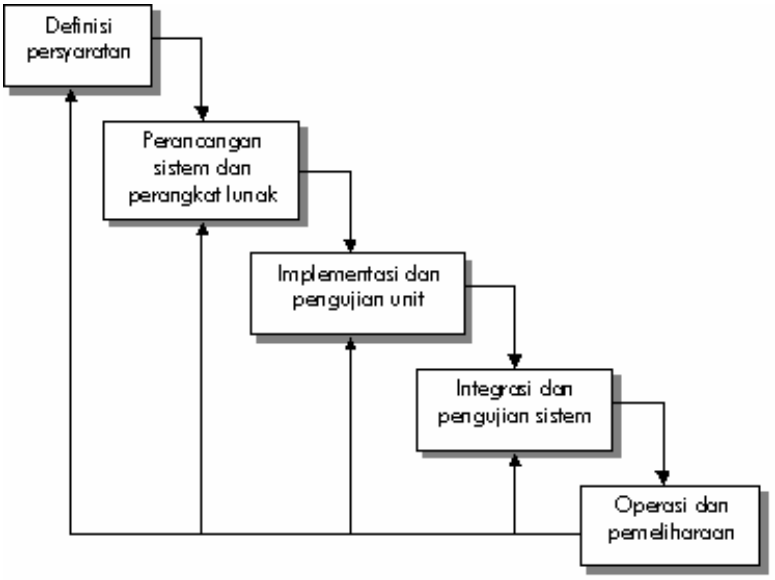
1. Kebutuhan harus telah ditetapkan di awal di siklus hidup.
2. Kebutuhan divalidasi terlalu lambat.

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.

Tahapan yang digunakan untuk membangun perangkat lunak dalam model ini yaitu sebagai berikut:

a. Rekayasa sistem,yaitu mengumpulkan data dengan meminta penjelasan dan data dari pihak-pihak terkait. Dilaksanakan dengan cara wawancara.
b. Menganalisa data, dokumen, perangkat lunak, perangkat keras dan user yang berkaitan dengan pembangunan sistem
c. Desain, yang sesuai dengan kebutuhan perangkat lunak sistem.
d. Pengkodean, pengembangan perangkat lunak dengan bahasa pemrograman untuk membuat aplikasi yang bersangkutan.
e. Pengujian, dilakukan untuk menyesuaikan kebutuhan awal dengan perangkat lunak yang dibangun.

f. Pemeliharaan, meliputi segala hal yang diperlukan dalam pengembangan perangkat lunak. Meskipun disebutkan dalam metodologi penelitian, namun tahap ini tidak akan dibahas dalam tugas akhir ini.



Gambar 2.4 Diagram Waterfall

2.2.5.2 *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[4]. 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 untuk 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*.

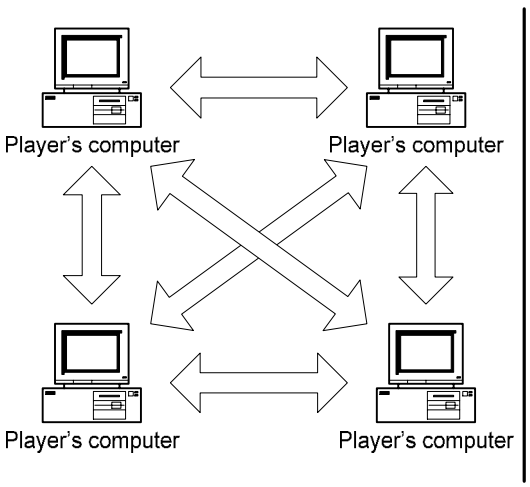
4. *Concurrent Development Model*

   Model ini biasa digunakan untuk pengembangan sistem *client/server*.[4]

2.2.6 Model Jaringan

2.2.6.1 *Model Peer To Peer*

Menurut model ini, setiap host dapat menawarkan layanan ke *peer* lain dan juga mengambil layanan dari *peer* lain. Model ini cocok untuk jaringan kecil. *Windows for Workgroup* menggunakan model ini.

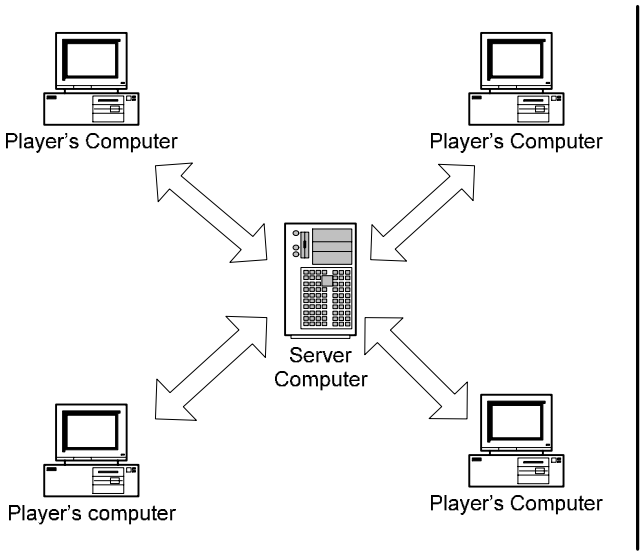


Gambar 2.5 Model Jaringan Peer to Peer

2.2.6.2 Model *Client-Server*

Menurut Gallaugher dan Ramanathan (1996) : "*client/server* adalah *client* mengirim permintaan ke *server*, *serve*r menterjemahkan pesan, kemudian berusaha memenuhi permintaan". Sedangkan menurut Blaha dan Premerlani (1998) : *client/server* adalah suatu arsitektur dimana sumber daya *server* menyediakan komputasi untuk banyak komponen *client*. *Client* dapat mengakses satu server atau *multiple server*. *Client* dan *server* bisa berjalan pada mesin yang sama ataupun berbeda, ditulis dalam berbagai bahasa dan menggunakan sistem operasi yang berbeda.

Secara umum *client-server* adalah arsitektur jaringan aplikasi yang memisahkan *client* dari *server* [7]. Setiap satuan perangkat lunak *client* berhubungan dengan perangkat lunak *server*. *Client/server* adalah arsitektur berskala dimana setiap komputer atau proses pada jaringan berperan sebagai *client* atau *server*. Piranti lunak *server* pada umumnya walaupun tidak selalu akan menjalankan komputer dengan kekuatan penuh yang secara khusus dan eksklusif menjalankan aplikasi bisnis. Sedangkan piranti lunak *client* pada umumnya berjalan pada PC atau *workstation* biasa. *Client* akan mendapatkan seluruh informasinya dan mengirimkannya kepada perangkat lunak *server* untuk sebuah keperluan, sebagai contoh yaitu konfigurasi file, kuota penyimpanan, program aplikasi bisnis atau untuk membebaskan intensifitas pekerjaan komputasi dan mengkondisikan komputer *client* bebas dan siap menjalankan pekerjaan lainnya.



Gambar 2.6 Model Jaringan Client-Server

2.2.7 Konsep Basis Data

2.2.7.1 Pengertian Basis Data

Istilah basis data banyak menimbulkan interpretasi yang berbeda. Anthoni J. Fabbri dan A. Robert Schwab, mendefinisikan basis data sebagai *sistem berkas terpadu yang dirancang terutama untuk meminimalkan pengulangan data*. Sedangkan menurut George Tsu-der Chou, basis data dapat didefinisikan sebagai *kumpulan informasi bermanfaat yang diorganisasikan ke dalam tatacara yang khusus*.[5]

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.

2.2.7.2 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.[5]

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 atribut 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.

2.2.8 Metode Perancangan Sistem

2.2.8.1 Diagram Konteks

Diagram konteks (*Context Diagram*) 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).

2.2.8.2 *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.

2.2.8.3 *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)

2.2.8.4 Normalisasi

Normalisasi adalah metode untuk mendapatkan susunan basis data relasional tanpa redundansi, dengan menghilangkan masalah-masalah yang potensial dalam proses modifikasi serta pengaksesan data. Ide dasar normalisasi berkaitan erat denga kunci relasi dan ketergantungan antarkunci.

Normalisasi terdapat 5 bentuk. Tiga bentuk pertama digunakan untuk mengurangi redundansi *functional dependency* (FD), yaitu nilai dan atribut unit di suatu tabel digunakan untuk menentukan nilai atau atribut di tabel yang lain.

Sedangkan dua yang lain digunakan untuk mengurangi redundansi *multiple Valued dependency* (MVD).

1. Bentuk normal I

   Relasi berada pada bentuk normal pertama bila pada satu sel di baris dan kolom tertentu hanya berisi satu unit data saja dan sederhana, tidak lagi mengandung kelompok data berulang (*reapeting groups*). Pada bentuk normal pertama redundansi data masih sangat terlihat.

2. Bentuk normal II

   Suatu entitas dikatakan sudah dalam bentuk normal kedua jika sudah ditempatkan pada bentuk normal pertama dan setiap item data dalam suatu entitas secara fungsional tergantung pada kunci primer (*primary key*) dari entitas tersebut. Jika ditemukan item data yang hanya bergantung secara parsial terhadap kunci primer maka item data tersebut harus dikeluarkan dan membentuk entitas yang baru.

3. Bentuk normal III

   Suatu entitas dikatakan sudah dalam bentuk normal kedua jika sudah ditempatkan pada bentuk normal kedua dan semua field yang tidak merupakan bagian dari kunci primer sudah bersifat independen, sehingga tidak ada lagi ketergantungan secara transitif. Bentuk normal ketiga mensyaratkan proses dekomposisi (penguraian dalam beberapa tabel) untuk mendapatkan basis data tanpa redundansi.

4. Bentuk normal untuk MVD

   Hubungan banyak ke banyak perlu diuraikan guna mereduksi perulangan data. Hubungan banyak ke banyak dapat diuraikan dengan menambahkan tabel relasi tambahan yang menghubungkan tabel pokok hasil uraian. Hubungan dibuat hingga redundansi tidak terlihat dan menganut FD, hubungan satu ke satu atau satu ke banyak.

2.2.8.5 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]

Notasi yang ada pada kamus data dibagi menjadi dua, yaitu notasi tipe data dan notasi struktur data.

Notasi	Keterangan
X	Setiap karakter
9	Angka numerik
A	Karakter alphabet
Z	Angka nol yag ditampilkan dalam spasi kosong
.	Titik, sebagai pemisah ribuan
,	Koma, sebagai pemisah pecahan
~	Hipen, sebagai tanda penghubung
/	Slash, sebagai tanda pembagi

Tabel 2.1 Notasi tipe data pada kamus data

Notasi	Keterangan
=	Terdiri dari
+	And (dan)
( )	Pilihan (ya / tidak)
{ }	Pengulangan proses
[ ]	Pilih salah satu pilihan
|	Pemisah pilihan pada tanda [ ]
*	Keterangan / catatan
@	Petunjuk (key field)

Tabel 2.1 Notasi struktur data pada kamus data

2.2.9 Perangkat Lunak Pendukung

2.2.9.1 Sekilas Pemrograman *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.

Singkatnya, Delphi merupakan perpaduan teknologi pemrograman terkini, lingkungan pengembangan terpadu (IDE) dan kekuatan *visual component library* (VCL).

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.*

2.2.9.2 MySQL

MySQL adalah *Relational Database Management System* (RDMS) yang didistribusikan secara gratis disebuah lisensi GPL (*General Public License*).

Dimana setiap orang bebas untuk menggunakan MySQL, namun tidak boleh dijadikan produk turunan yang bersifat *close source* atau komersial. MySQL sebenarnya merupakan turunan salah satu konsep utama dalam database sejak lama, SQL (*Structured Query Language*). SQL adalah sebuah konsep pengoperasian database, terutama untuk pemilihan (seleksi) dan pemasukan data yang memungkinkan pengoperasian data dikerjakan dengan mudah secara otomatis. Keandalan suatu sistem basis data (DBMS) dapat diketahui dengan cara kerja optimizernya dalam melakukan proses perintah-perintah SQL yang dibuat oleh *user* maupun program-program aplikasinya sebagai *database server* lainnya dalam *query data*. MySQL adalah satu dari sekian banyak sistem basis data merupakan terobosan solusi yang tepat dalam aplikasi database.

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

2.2.9.3 Koneksi Borland Delphi dengan MySQL

Telah diketahui bahawa BDE 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 oleh Interbase. BDE dapat mengakses server SQL seperti Ms.SQL Server dengan memanfaatkan *driver* ODBC buatan Microsoft, tetapi *driver* tersebut tidak dapat mengakses basisdata MySQL.

dbExpress merupakan komponen yang disediakan oleh Delphi untuk membangun 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 *client-server* dibuat dengan menggunakan dbExpress. dbExpress juga dijadikan metode akses data pada Borland Kylix yang berjalan di Linux sehingga aplikasi berbasis dbExpress di window juga dapat dikompilasi dan dijalankan di Linux.

This document was created with Win2PDF available at http://www.daneprairie.com.
The unregistered version of Win2PDF is for evaluation or non-commercial use only.