BAB II

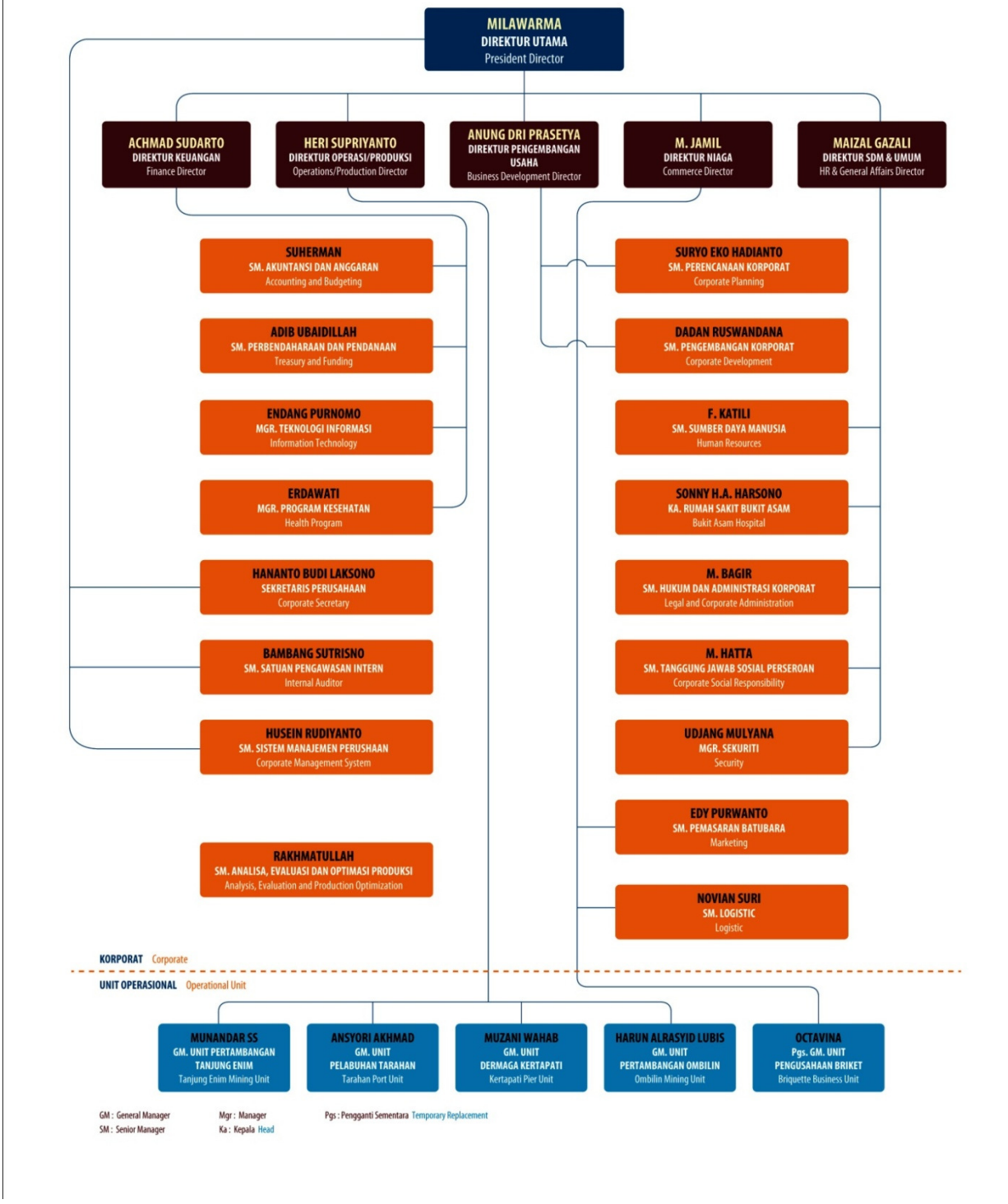
TINJAUAN PUSTAKA

2.1 Tinjauan Objek Penelitian

Objek yang digunakan sebagai sasaran dari aplikasi sistem inventory Gudang ini adalah Gudang Barang kebutuhan IT di PT.Bukit Asam

2.1.1 Struktur Organisasi

Adapun Struktur Organisasi yang terbentuk adalah sebagai berikut



Gambar 2.1 Struktur Organisasi

2.1.2 Visi Dan Misi

Visi

Menjadi perusahaan energi berbasis batubara yang ramah lingkungan

Misi

a. Fokus kepada core competency dan pertumbuhan yang berkesinambungan
b. Memberikan tingkat pengembalian yang optimal kepada pemegang saham
c. Meningkatkan budaya korporasi yang mengutamakan kinerja
d. Memberikan kontribusi pengembangan ekonomi nasional
e. Memberikan kontribusi yang maksimal dalam meningkatkan kesejahteraan masyarakat dan pelestarian lingkungan

2.2 Landasan Teori

2.2.1 Sistem Informasi

2.2.1.1 Definisi Sistem

Terdapat dua kelompok pendekatan di dalam pendefinisian sistem, yaitu yang menekankan pada prosedur dan yang menekankan pada komponen atau elemen-elemen. Pendekatan sistem yang menekankan pada prosedurnya, didefinisikan oleh Jerry Fitzgerald dalam bukunya Fundamentals of System Amalysis dan diterjemahkan oleh H.M Jogiyanto dalam bukunya "*Analisis Dan Desain Sistem Informasi*" adalah sebagai berikut :

"Suatu sistem adalah suatu jaringan kerja dari prosedur-prosedur yang saling berhubungan, berkumpul bersama-sama untuk melakukan suatu kegiatan atau untuk menyelesaikan suatu sasaran tertentu" (H.M Jogiyanto, 2005:1).

1. Pendekatan Sistem

Pendekatan sistem dapat diartikan suatu cara berpikir yang menguraikan suatu sistem sebagai kerangka atau jaringan guna memperhatikan faktor-faktor yang berbeda di dalam dan di luar sistem sebagai suatu kumpulan yang terpadu. Pendekatan sistem merupakan suatu persepsi tentang

struktur yang mengkoordinasikan bagian-bagian dan operasi-operasi dalam suatu organisasi atau sistem dengan cara yang efisien.

2. Ciri-ciri Utama Suatu Sistem

1. Mengarah pada suatu tujuan tertentu, yaitu setiap elemen sistem melakukan kegiatan masing-masing sesuai dengan fungsinya, tetapi mengarah kepada tujuan sistem yang sama dan secara keseluruhan
2. Bersifat terbuka, artinya sistem berinteraksi dengan lingkungannya
3. Mempunyai mekanisme kontrol, yaitu suatu sistem mempunyai kemampuan untuk mengatur dirinya sendiri kesuatu posisi yang lebih baik dengan menjalankan kedinamisan dan mempersatukan keharmonisan dalam sistem itu sendiri
4. Merupakan status keseluruhan, yaitu sistem merupakan suatu keseluruhan yang bulat dan utuh, dimana tujuan masing-masing bagian yang membentuk sistem tersebut saling menunjang untuk mencapai tujuan sistem secara keseluruhan
5. Adanya proses transformasi, yaitu didalam sistem selalu ada proses transformasi, yang mengubah *input* menjadi *output*, atau mengubah suatu masukan menjadi suatu keluaran untuk mencapai tujuan yang diinginkan
6. Adanya keterkaitan atau hubungan timbal-balik, yaitu adanya hubungan timbal-balik antara elemen-elemen sistem yang satu dengan yang lainnya ataupun dengan lingkungannya, sehingga apabila terjadi perubahan elemen dari suatu sistem, maka akan berpengaruh terhadap sistem secara keseluruhan

3. Karakteristik Sistem

Suatu sistem mempunyai karakteristik atau sifat-sifat tertentu antara lain sebagai berikut :

1. Komponen-komponen sistem (*Components*)

Merupakan salah satu karakteristik sistem yang berupa sub-sistem atau gagasan sistem

2. Batas sistem (*Boundary*)

Merupakan daerah yang membatasi antara suatu sistem dengan sistem lainnya atau dengan lingkungan lainnya

3. Lingkungan luar sistem (*Environment*)

4. Segala sesuatu yang berada diluar batas dari sistem yang mempengaruhi operasi baik itu menguntungkan operasi sistem yang merupakan energi dari sistem yang harus dipelihara atau ditahan sehingga tidak menggangu atau merusak sistem

5. Penghubung sistem (*Interface*)

Merupakan media penghubung antara satu sub-sistem dengan sub-sistem lainnya

6. Masukan sistem (*Input*)

Masukan sistem adalah data yang dimasukkan ke dalam sistem yang di proses untuk mendapat satu kesatuan atau informasi

7. Keluaran sistem (*Output*)

Keluaran dari sistem adalah data yang diolah serta diklasifikasikan menjadi keluaran yang berguna atau menjadi informasi yang dibutuhkan

8. Pengolah sistem (*Process*)

Suatu sistem yang mempunyai bagian pengolahan data yang akan mengubah masukan menjadi keluaran yang berguna

9. Sasaran sistem (*Goal*)

Merupakan maksud tertentu atau tujuan atas sasaran, dimana yang menentukan masukan serta keluaran sistem yang dibutuhkan pada sasaran atau tujuan yang maksud.

2.2.1.2 Definisi Informasi

Menurut Jogiyanto dalam bukunya yang berjudul *Analisis dan Desain Sistem Informasi, mendefinisikan informasi sebagai berikut* :

"Informasi adalah data yang diolah menjadi bentuk yang lebih berarti bagi yang menerimanya". (H.M Jogiyanto , 2005:8)

1. Kualitas Informasi

Sumber dari informasi adalah data, kualitas dari suatu informasi tergantung dari empat hal antara lain :

1. Akurat

Akurat berarti suatu informasi harus bebas dari kesalahan-kesalahan dan tidak menyesatkan

2. Tepat

Tepat berarti suatu informasi harus diterima oleh pemakai informasi tepat pada waktunya

3. Lengkap

Lengkap berarti suatu informasi diberikan dengan selengkap-lengkapnya agar dapat dimanfaatkan secara maksimal oleh pemakai informasi

4. Relevan

Relevan berarti suatu informasi tersebut mempunyai manfaat yang sesuai dengan pemakainya

2. Nilai Informasi

Nilai dari informasi (*Value of Information*) ditentukan dari dua hal yaitu manfaat dan biaya mendapatkannya. Suatu informasi dikatakan bernilai bila manfaatnya lebih efektif dibandingkan dengan biaya mendapatkannya. Akan tetapi perlu diperhatikan bahwa informasi yang digunakan didalam suatu sistem informasi umumnya digunakan untuk beberapa kegunaan. Sebagian informasi tidak dapat ditaksir keuntungannya dengan suatu nilai tetapi dapat ditaksir dari nilai keefektifannya.

2.2.1.3 Definisi Sistem Informasi

Definisi sistem informasi menurut Jogiyanto dalam bukunya yang bejudul *Analisis dan Desain Sistem Informasi* adalah sebagai berikut:

"Sistem informasi 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". (H.M Jogiyanto , 2005:11)

Dalam sebuah sistem informasi terdapat beberapa kegiatan yang dilakukan antara lain :

1. *Input*

Suatu kegiatan yang dilakukan untuk menyediakan data yang akan di proses

2. *Process*

Suatu kegiatan yang dibutuhkan untuk memproses suatu data yang akan menghasilkan suatu informasi yang memiliki nilai tambah

3. *Output*

Suatu kegiatan yang dilakukan untuk menghasilkan laporan dari proses yang di atas

2.2.2 Basis Data

Basis data (*database*) terdiri dari dua kata yaitu basis dan data. Basis dapat diartikan sebagai markas atau gudang tempat berkumpul. Sedangkan data adalah representasi fakta dunia nyata yang

mewakili objek seperti manusia, barang, hewan, peristiwa konsep dan sebagainya, yang direkam dalam bentuk angka, huruf, simbol teks, gambar, bunyi atau kombinasinya. Basis data sendiri dapat didefinisikan dalam sejumlah sudut pandang seperti berikut ini:

1. Himpunan data atau arsip yang saling berhubungan yang diorganisasikan sedemikian rupa agar kelak dapat dimanfaatkan kembali dengan cepat dan mudah.(wikipedia)

2. Menurut Gordon C. Everest : Database adalah koleksi atau kumpulan data yang mekanis, terbagi/shared, terdefinisi secara formal dan dikontrol terpusat pada organisasi.

3. Menurut C.J. Date : Database adalah koleksi "data operasional" yang tersimpan dan dipakai oleh sistem aplikasi dari suatu organisasi.

1. Data input adalah data yang masuk dari luar sistem
2. Data output adalah data yang dihasilkan sistem
3. Data operasional adalah data yang tersimpan pada sistem

4. Menurut Toni Fabbri : Database adalah sebuah sistem file-file yang terintegrasi yang mempunyai minimal primary key untuk pengulangan data.

5. Menurut S. Attre :D atabase adalah koleksi data-data yang saling berhubungan mengenai suatu organisasi / enterprise dengan macam-macam pemakaiannya.

6. Menurut Fathansyah, 1999:

1. Himpunan kelompok data (arsip) yang saling berhubungan yang diorganisasi sedemikian rupa agar kelak dapat dimanfaatkan kembali dengan cepat dan mudah.
2. Kumpulan data yang saling berhubungan yang disimpan secara bersama sedemikian rupa dan tanpa pengulangan (redundansi) yang tidak perlu, untuk memenuhi berbagai kebutuhan.
3. Kumpulan file/tabel/arsip yang saling berhubungan yang disimpan dalam media penyimpanan elektronis.

7. Menurut Awaludin, 2004: "Database adalah Sekumpulan informasi yang terkait pada subjek tertentu atau terkait pada tujuan tertentu".

8. Menurut Tutang, 2001: "Database adalah kumpulan informasi yang disusun berdasarkan cara tertentu, dengan sistem atau cara tertentu".

9. Menurut Yuswanto, 2001: "Database adalah sekumpulan data/informasi yang teratur berdasarkan kriteria tertentu yang saling berhubungan".

2.2.3 DBMS (Database Management System)

Sistem manajemen *database* atau *database management system* (DBMS) adalah merupakan suatu sistem *software* yang memungkinkan seorang *user* dapat mendefinisikan, membuat, dan memelihara serta menyediakan akses terkontrol terhadap data.

DBMS yang utuh biasanya terdiri dari :

1. *Hardware*

Hardware merupakan sistem komputer aktual yang digunakan untuk menyimpan dan mengakses *database*. Dalam sebuah organisasi berskala besar, *hardware* terdiri dari jaringan dengan sebuah *server* pusat dan beberapa program *client* yang berjalan di komputer desktop.

2. *Software* beserta *utility software* adalah DBMS yang aktual. DBMS memungkinkan para *user* untuk berkomunikasi dengan *database*. Dengan kata lain DBMS merupakan mediator antara *database* dengan *user*. Sebuah *database* harus memuat seluruh data yang diperlukan oleh sebuah organisasi.

3. Prosedur

Bagian integral dari setiap sistem adalah sekumpulan prosedur yang mengontrol jalannya sistem, yaitu praktik-praktik nyata yang harus diikuti *user* untuk mendapatkan, memasukkan, menjaga, dan mengambil data

4. Data

Data adalah jantung dari DBMS. Ada dua jenis data. Pertama, adalah kumpulan informasi yang diperlukan oleh suatu organisasi. Jenis data kedua adalah metadata, yaitu informasi mengenai *database*.

5. *User* (Pengguna)

Ada sejumlah *user* yang dapat mengakses atau mengambil data sesuai dengan kebutuhan penggunaan aplikasi-aplikasi dan *interface* yang disediakan oleh DBMS, antara lain adalah:

1. *Database* administrator adalah orang atau group yang bertanggungjawab mengimplementasikan sistem *database* di dalam suatu organisasi
2. *End user* adalah orang yang berada di depan *workstation* dan berinteraksi secara langsung dengan sistem.
3. *Programmer aplikasi*, orang yang berinteraksi dengan *database* melalui cara yang berbeda.

2.2.4 ERD (Entity Relationship Diagram)

Setelah semua memory esensial ditransformasikan menjadi *object data store,* maka dapat disusun relasi antara satu *object data store* dengan yang lain. *Entity Relationship* (E-R) *Diagram* dibuat untuk memperlihatkan relasi itu. *Entity Relationship Diagram* adalah ilustrasi grafis objek-objek (atau *entities*) dan atribut, serta relasi antara keduanya. Diagram E-R diperkenalkan oleh Peter Chen dan sifatnya independen terhadap teknologi *database* yang digunakan. Salah satu metode pembuatan diagram E-R, yaitu dengan menggambarkan objek-objek sebagai segiempat, relasi sebagai belah ketupat kecil, dan atribut sebagai lingkaran. Notasi yang lain adalah notasi. E-R Diagram merupakan suatu bahasa pemodelan yang dimana posisinya dapat dianalogikan dengan *stroy board* dalam industri film, *blu print* arsitektur suatu bangunan, miniatur, dan lain-lain.

Dalam praktiknya, membangun suatu sistem terlebih dahulu dilakukannya suatu perencaan. Pemodelan merupakan suatu sub bagian dari perencanaan secara keseluruhan sebagai salah satu upaya *feedback* evaluasi perampungan suatu perencanaan. ERD untuk memodelkan struktur data dan hubungan antar data, untuk menggambarkannya digunakan beberapa notasi dan simbol. Pada dasarnya ada tiga simbol yang digunakan, yaitu:

1. Entiti

Entiti merupakan objek yang mewakili sesuatu yang nyata dan dapat dibedakan dari sesuatu yang lain (Fathansyah, 1999: 30). Simbol dari entity ini biasanya digambarkan dengan persegi panjang.

2. Atribut

Setiap entitas pasti mempunyai elemen yang disebut atribut yang berfungsi untuk mendeskripsikan karakteristik dari entitas tersebut. Isi dari atribut mempunyai sesuatu yang dapat mengidentifikasikan isi elemen satu dengan yang lain. Gambar atribut diwakili oleh simbol elips.

3. Hubungan / Relasi

Hubungan antara sejumlah entitas yang berasal dari himpunan entitas yang berbeda. E-R Diagram sebagai suatu pemodelan setidaknya memiliki beberapa karakteristik dan manfaat sebagai berikut:

1. Memudahkan untuk dilakukannya analisis dan perubahan sistem sejak dini, bersifat murah dan cepat
2. Memberikan gambaran umum akan sistem yang akan di buat sehingga memudahkan developer.

3. Menghasilkan dokumentasi yang baik untuk *client* sebagai bahan diskusi dengan bentuk E-R Diagram itu sendiri, dan

4. Kamus data bagi bagi para pengembang *database*. Struktur dari E-R Diagram secara umum ialah terdiri dari:

1. Entitas merupakan objek utama yang informasi akan disimpan, biasanya berupa kata benda. Contoh: mahasiswa, dosen, nasabah, mata kuliah, ruangan, dan lain-lain. Objek dapat berupa benda nyata maupun abstrak.

2. Atribut merupakan deskripsi dari objek yang bersangkutan

3. *Relationship* merupakan suatu hubungan yang terjalin antara dua entitas yang ada. Kardinalitas Relasi ERD yang mempersentasikan suatu basisdata relasional senantiasa memiliki relasi-relasi dari sejumlah entitas yang dapat ditentukan banyaknya. Banyaknya suatu relasi yang dimiliki oleh suatu relasi entitas disebut derajat relasi. Derjat relasi maksimum disebut dengan kardinalitas sedangkan derajad minimum disebut dengan modalitas. Relasi yang terjadi diantara dua himpunan entitas (misalnya A dan B) dalam satu basis data yaitu (Abdul Kadir, 2002: 48):

1. *One to One* (1-1), relasi yang terjadi jika sebuah *entry* dalam sebuah *object data sore* dihubungkan dengan hanya sebuah *entry* dalam *object data store* yang lain.

2. *One to Many* (1-M), relasi yang terjadi jika sebuah *entry* dalam sebuah *object data store* dihubungkan dengan satu atau lebih *entry* dalam *object data store* yang lain.

3. Many to Many (M-M), relasi yang terjadi jika satu atau lebih *entry* dalam sebuah *object data store* dihubungkan dengan satu atau lebih *entry* dalam *object data store.*

Dalam semua definisi relasi di atas, “1” menyatakan hanya satu, dan “M” menyatakan satu atau lebih. Jika suatu saat dibutuhkan pernyataan relasi paling sedikit sebuah dan juga lebih dari satu, maka relasi itu dapat dinyatakan dengan “M*” (* = *minimum of one*).

2.2.5 DFD (*Data Flow Diagram*)

Data Flow Diagram (DFD) adalah alat yang digunakan untuk menggambarkan suatu sistem yang telah ada atau sistem baru yang akan dikembangkan secara logika tanpa mempertimbangkan lingkungan fisik dimana data tersebut mengalir ataupun lingkungan fisik dimana data tersebut akan disimpan (Jogiyanto, HM, 2005 :700). Elemen dasar dari *data flow diagram* adalah :

a. Entitas Luar (*External Entity*)

Sesuatu yang berada diluar sistem, tetapi ia memberikan data ke dalam sistem atau memberikan data dari sistem, disimbolkan dengan suatu kotak notasi. *External Entity* tidak termasuk bagian dari sistem. Bila sistem informasi dirancang untuk satu bagian maka bagian lain yang masih terkait menjadi *external entity*.

b. Arus Data (*Data Flow*)

Arus data merupakan tempat mengalirnya informasi dan digambarkan dengan garis yang menghubungkan komponen dari sistem. Arus data ditunjukan dengan arah panah dan garis diberi nama atas arus data yang mengalir. Arus data ini mengalir diantara proses, *data store* dan menunjukan arus data dari data yang berupa masukan untuk sistem atau hasil proses sistem.

1. Proses (*Process*)

Proses merupakan apa yang dikerjakan oleh sistem. Proses dapat mengolah data atau aliran data masuk menjadi aliran data keluar. Proses berfungsi mentransformasikan satu atau beberapa data keluaran sesuai dengan spesifikasi yang diinginkan. Setiap proses memiliki satu atau beberapa masukan serta menghasilkan satu atau beberapa data keluaran. Proses sering juga disebut *bubble*.

2. Simpanan Data (*Data Store*)

Simpanan data merupakan tempat penyimpanan data yang ada dalam sistem. Data store dapat disimbolkan dengan dua garis sejajar atau dua garis dengan salah satu sisi samping terbuka. Proses dapat mengambil data dari atau memberikan data ke simpanan data (*database*).

3. Kamus Data

Kamus data berfungsi untuk membantu pelaku sistem untuk mengartikan aplikasi secara detail dan mengorganisasi semua elemen data yang digunakan dalam sistem secara persis sehingga pemakai dan penganalisis sistem mempunyai dasar pengertian yang sama tentang masukan, keluaran, penyimpanan dan proses.

2.2.6 Structure Query Language (SQL)

SQL merupakan bahasa yang memungkinkan pemakai untuk melakukan akses data ke *DBMS* dan pemakai dapat menggambarkan data yang ingin diperoleh (Hoffman, 2001). SQL juga memungkinkan pemakai untuk melakukan manipulasi data (*input, edit*, dan *delete*). Dalam perkembangannya, SQL telah menjadi bahasa standar dalam kegiatan manipulasi data pada *database* dari berbagai *vendor* yang ada di dunia. *Terminator*, menandai awal atau akhir dari suatu proses .*Decision*, untuk menandai percabangan dalam proses *Input/*output*,* menggambarkan adanya

input atau *output* dari system Process*, menggambarkan proses yang dilakukan. Flow, menggambarkan arah urutan proses yang dilakukan Preparation,* simbol untuk dimulainya proses looping *Predefined process,* untuk menggambarkan proses yang telah dimasukkan sebelumnya *Stored procedure,* simbol untuk mewakili proses yang dilakukan pada data yang tersimpan, seperti *insert*, *delete* dan *update*

Pada subbab ini akan diberikan penjelasan tentang dasar penggunaan dari perintah SQL. Tabel dan data yang akan dipergunakan sebagai contoh dapat dilihat pada Tabel 2.1 untuk Tabel barang dan Tabel 2.2 untuk tabel kelompok.

ID_Barang	Kode_Umum	Nama_Barang	ID_Kelompok
1000000012	KT278-220	Arbalest	K0001
2003002001	JTC9-84788	Rancing Car	K0003
4421222236	YUBERUA	Toy Gun	K0002
2131244532	82-CTCH	Buldozer	K0003

Tabel 2.1. Contoh Tabel Barang

ID_Kelompok	Nama_Kelompok
K0001	Robot
K0002	Pistol
K0003	Mobil

Tabel 2.2. Contoh Tabel Kelompok

Beberapa Perintah yang umum dipergunakan dalam *SQL* adalah sebagai berikut:

a. *SELECT*, merupakan perintah untuk melakukan pengambilan data yang berada dalam tabel. Bila ingin melakukan pengambilan terhadap data yang berada dalam:

• Tabel barang secara keseluruhan

SELECT * from barang

• Tabel barang untuk field ‘ID Barang’ dan ‘Kode Umum’ saja

SELECT id_barang,kode_umum FROM barang

b. *Conditional*, merupakan perintah untuk mengambil data dengan kondisi tertentu, menggunakan "WHERE" dan *relational operator* seperti "<", "> ","=", "<>". Berikut pengambilan data barang dengan "ID_barang" 1000000012.

SELECT * from barang WHERE id_barang='1000000012'.

c. *Joins*, merupakan bentuk perintah untuk melakukan pengambilan data dari dua atau lebih tabel sekaligus. Berikut adalah cara pangambilan untuk *field* id_barang, nama_barang yang berasal dari tabel barang dan nama kelompoknya yang berasal dari tabel kelompok.

```
SELECT barang.id_barang, barang.nama_barang,
kelompok.Nama_kelompok FROM barang, kelompok
WHERE barang.id_kelompok=kelompok.id_kelompok
```

d. *Aggregate Function* seperti *Sum()*, *Count()*, *Avg()*, *Max()*, *Min()*. Bila ingin mengetahui berapa jumlah "id_barang" dengan id_kelompok tertentu pada tabel barang maka perintah yang diberikan :

```
SELECT COUNT(id_kelompok) FROM barang WHERE
id_kelompok='K0002'
```

Hasil dari perintah ini adalah 2 yang berarti terdapat 2 barang dengan id_kelompok "K0002".

e. *CREATE TABLE*, perintah ini dipergunakan untuk membuat sebuah tabel baru dengan *field*-nya. Contoh: untuk membuat sebuah tabel *supplier* dengan *field* "Id_supplier" dan "nama_supplier" diberikan perintah sebagai berikut:

```
CREATE TABLE supplier (id_supplier char(5), nama_supplier
varchar(30))
```

f. *ALTER TABLE*, perintah ini berfungsi melakukan perubahan pada struktur tabel (bukan isi tabel). Bila kita ingin menambahkan *field* telepon pada tabel *supplier*, maka perintah yang diberikan sebagai berikut:

```
ALTER TABLE supplier ADD (telepon varchar(25))
```

g. *INSERT*, merupakan perintah yang dipergunakan untuk menambahkan data pada tabel. Untuk menambahkan data *supplier* dengan nama "Sinar Terang", maka perintah yang dipergunakan :

```
INSERT INTO supplier VALUES ('S00001','Sinar Terang'
,'03120030020')
```

h. *UPDATE*, berfungsi untuk mengubah data pada tabel. Bila nama kelompok pada tabel kelompok dengan kode “K0001” ingin diganti dengan“boneka” maka perintah yang diberikan :
UPDATE kelompok SET nama_kelompok =’boneka’ WHERE
id_kelompok=’K0001’

i. *DELETE*, merupakan perintah untuk menghapus data pada tabel baik satu atau seluruhnya. Untuk menghapus data tertentu maka dapat dipergunakan *WHERE* sedangkan untuk menghapus seluruh isi tabel maka *WHERE* tidakdipergunakan lagi. Contoh menghapus seluruh isi tabel barang:
DELETE FROM barang;

Secara keseluruhan perintah SQL sangatlah beragam dan dapatdipergunakan sesuai dengan kebutuhan manipulasi data pada aplikasi *database*. Untuk setiap *database server* akan terdapat beberapa perintah *SQL* dengan sintaks tersendiri.

2.2.7 Borland Delphi

Delphi adalah perangkat pengembangan aplikasi yang berjalan di sistem operasi Windows.Delphi merupakan kelanjutan dari Turbo Pascal yang merupakan produk Borland, sekarang telah berganti menjadiInprise Corporation. Delphi diluncurkan pertama kali pada tahun 1995.

Borland menyatakan bahwa Delphi merupakan alat yang dapat digunakan untuk Rapid Application Development (RAD), yang hingga kini tetap terbukti menjadi bahasa pemrograman yang paling baik. Dengan keberhasilan ini maka Borlang (sebelum menjadi Inprise), mengembangkan produk lain yang mirip yakni C++Builder, Jbuilder (Java Language). Dengan demikian jika kita telah menguasai Delphi dengan mudah kita dapat berpindah menggunakan bahasa lainnya yaitu C++Builder atau Jbuilder. Kedua produk ini sengaja dibuatmemiliki IDE yang sama dengan Delphi.
Selain sebagai aplikasi biasa dengan mudah Delphi membuat aplikasi untuk internet. Seperti Win-CGI(Windows Common Gateway Interface), ISAPI (Internet Service Application Program Interface) . Bahkan Borland menyatakan sebagai satu langkah menuju ActiveX. Seperti dengan mudah membuat sebuah aplikasi ActiveForm yang dapat berjalan pada Internet Explorer sebagai suatu aplikasi biasa. Selain itu Delphi dapat membuat aplikasi MIDAS, pada aplikasi ini kita harus membuat dua lapis, lapis pertama adalah aplikasi Server yang melayani permintaan aplikasi kedua yaitu Client.

Delphi dapat mengakses database Paradox, xBase, MS-Access juga dengan menggunakan ODBC Delphi dapat mengakses database lain seperti Oracle, Sybase, Interbase, DB2, MS-SQL, MySQL. Sehingga kita dapat berpindah dari satu database ke database lain tanpa perlu mengubah aplikasi secara keseluruhan.

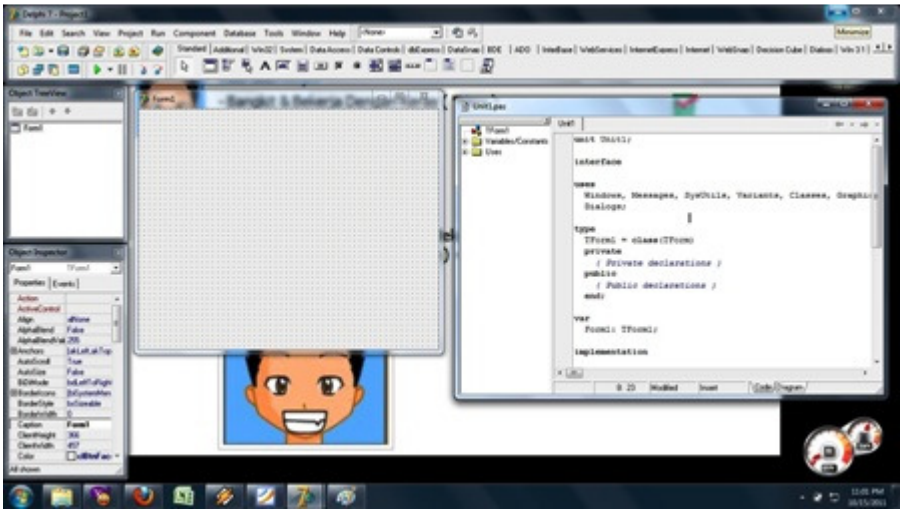
Delphi adalah suatu bahasa pemograman (development language) yang digunakan untuk merancang suatu aplikasi program. keunggulan Delphi dibanding bahasa pemograman lainnya :

1. IDE (integrated Development Environment) yakni lingkungan aplikasi yang didalamnya terdapat menu menu yang memudahkan kita untuk membuat suatu proyek program.
2. mudah digunakan, source kode delphi yang merupakan turudan dari pascal.
3. sifatnya multi purphase yakni mudah digunakan untuk mengembangkan berbagai keperluan pengembangan aplikasi

Berikut ini Lembar Kerja Delphi 7 :

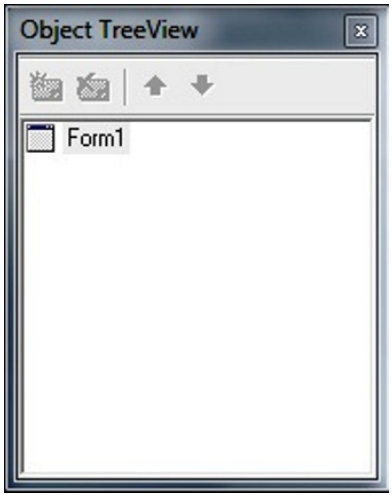
Untuk Membuka Program Pilih Start -> Program File -> Borland Delphi 7 -> Delphi 7.exe

Berikut ini adalah Bagian Bagian dari Lembar Kerja Delphi, Berikut Penjabarannya :



Gambar 2.1 Lembar Kerja Delphi

1. Object Tree View -> diagram pohon yang menggambarkan hubungan logis komponen kompenen dalam project program. meliputi Form, Modul, Frame yang sesuai dengan penempatannya.



Gambar 2.2 Jendela Object Tree View

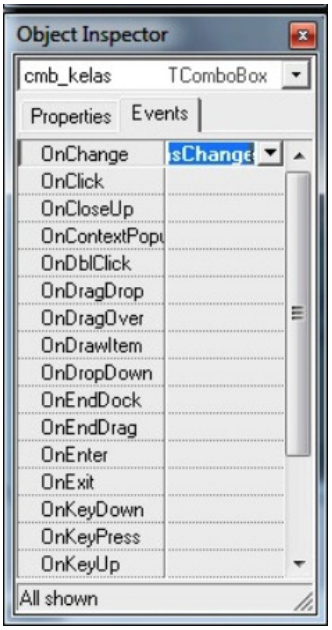
2. Object Inspector -> Merupakan jendela yang digunakan guna mengatur tampilan komponen pada form, seperti menulisan tampilan Caption, Penamaan dari suatu komponent. Object Inspector secara umum terbagi menjadi 2 tab, Yakni :

Object Properties yang digunakan untuk mengatur tampilan dalam suatu komponen baik itu nama, warna, font, border dan lain sebagainya.



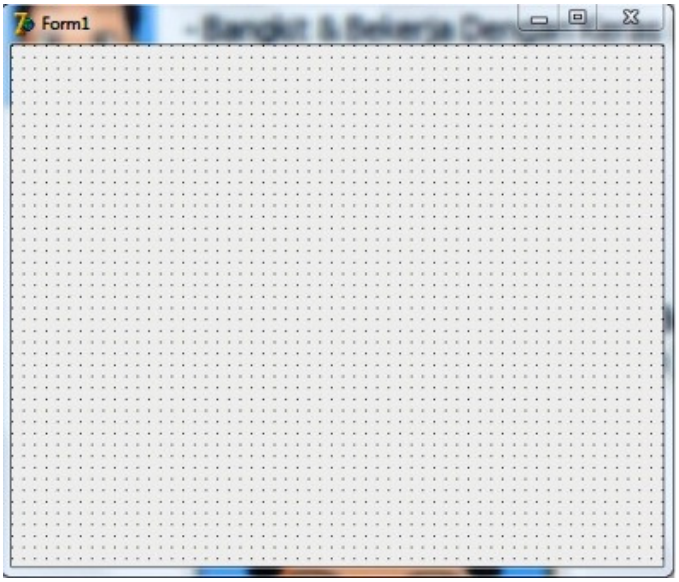
Gambar 2.3 Jendela Object Inspektor Properties

Object Event yang digunakan untuk memberikan fungsi yang lebih detail dari perintah-perintah coding yang kita buat didalam komponent, agar berjalan seperti yang kita inginkan.



Gambar 2.4 Jendela Object Inspector Event

3. Form Designer -> Merupakan tempat yang digunakan untuk merancang suatu tampilan aplikasi program yang dapat dimasukkan komponen komponent pallete.



Gambar 2.5 Tampilan Form Designer

4. Component Pallete -> Merupakan Kumpulan Icon yang Digunakan untuk merancang suatu aplikasi, dimana mereka akan membentuk suatu interface sebuah aplikasi. Dalam Setiap komponen

Pallete semua icon dikelompokkan dalam berbagai komponen sesuai dengan fungsi dan kegunaannya.



Gambar 2.6 Komponen Pallete

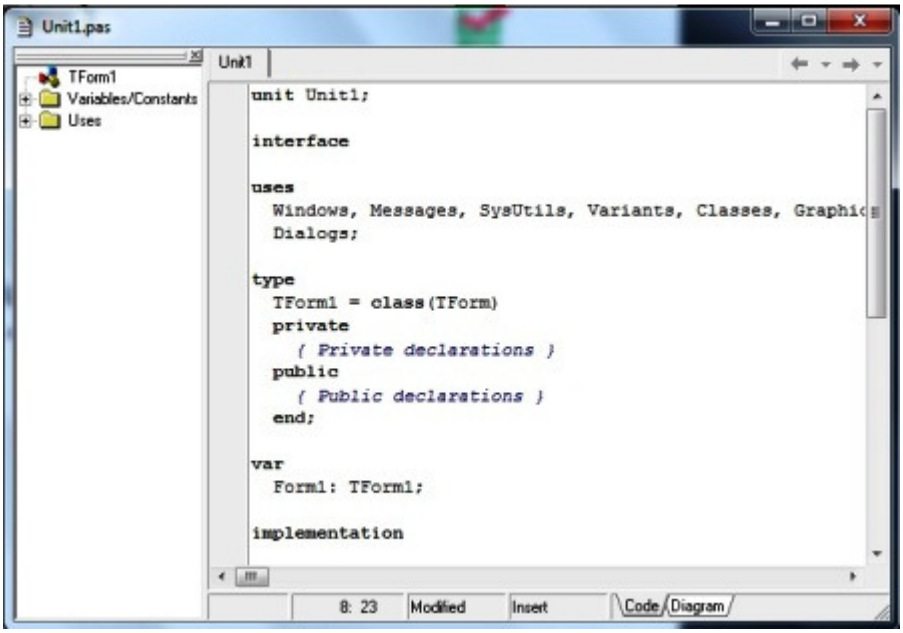
Berikut Beberapa Fungsi Dari Suatu Pallete Standart :

No	Icon	Name	Fungsi
1		**Pointer**	Mengembalikan fungsi mouse ke defaultnya
2		Frame	Membentuk suatu frame terhadap obyek yang ada didalamnya
3		Main menu	Membuat menu Utama
4		Popup Menus	
5		label	Hanya untuk menampilkan Teks
6		Edit	Untuk menampilkan dan input data (1 baris)
7		Memo	Sama seperti edit tetapi mempunyai kapasitas lebih besar (lebih dari 1 baris)
8		Button	Digunakan untuk melakukan eksekusi terhadap suatu proses
9		Checkbox	Digunakan untuk menentukan pilihan lebih dari satu
10		Radio Button	Digunakan untuk menentukan pilhan, tetapi hanya satu pilhan yang bisa digunakan
11		List Box	Menmpilkan pilihan dalam bentuk list
12		Combo Box	Menampilkan pilihan dalam bentuk popup
13		Scroll Bar	Merupakan icon yang berupa baris status
14		Group Box	Digunakan untuk mengelompokan suatu icon
15		Radio Group	Digunakan untuk mengelompokan pilihan

Gambar 2.7 Component dan Fungsi Pallete

4. Code Editor -> Bagian dari Delphi yang kita gunakan dalam penulisan Kode Program. Pada bagian Code Editor terdapat 3 bagian tama yakni = bagian paling kiri yang berupa angka menunjukan

baris dan kolom dan keterangan modified menunjukan bahwa telah terjadi modifikasi terhadap baris program. dan yang paling kanan menunjukkan tentang tombol insert atau overwrite.



Gambar 2.8 Jendela Unit.

2.2.8 Database Paradox

Paradoks adalah sistem manajemen database relasional saat ini diterbitkan oleh Corel Corporation. Awalnya dirilis untuk DOS oleh Ansa Software, tapi versi Windows dirilis oleh Borland pada tahun 1992.

Paradox Untuk DOS

Paradox untuk DOS adalah sistem manajemen database relasional aslinya ditulis oleh Richard Schwartz dan Robert Shostak, dan dirilis oleh perusahaan mereka Ansa Software pada tahun 1985. Pada bulan September 1987, Borland dibeli Ansa Software, termasuk / mereka software DOS Paradox 2.0. versi klasik terkenal adalah 3,5 dan 4,5. Versi sampai 3.5 adalah evolusi dari 1,0. Versi 4.0 dan 4,5 telah retooled di Borland C + + windowing toolkit dan menggunakan skema akses memori yang berbeda diperpanjang.

Paradoks / DOS adalah database berbasis DOS sukses akhir tahun delapan puluhan dan awal tahun sembilan puluhan. Pada waktu itu, dBase dan klon xBase nya (FOxPro, Clipper) mendominasi pasar. pesaing terkenal lainnya adalah Clarion, DataEase, R: Base, dan DataFlex.

Fitur yang membedakan Paradox / DOS adalah:

Sebuah Query visual dengan implementasi Contoh yang didukung oleh mesin AI.Efektif penggunaan memori (konvensional serta diperpanjang / diperluas) – tabel data caching dan terutama, indeks yang menyebabkan Paradox untuk melaksanakan tugas-tugas yang sangat cepat dalam kontras dengan keterampilan eksplisit diperlukan untuk optimasi kinerja xBase. [1]Sebuah bahasa pemrograman inovatif Paradox Aplikasi Bahasa (PAL) yang dibaca, kuat, dan dapat direkam dari tindakan keyboard (bukan seperti perekaman makro Lotus 1-2-3).lotus menu seperti teks dan jendela yang merupakan antarmuka asli (kontras dengan dBase yang memiliki antarmuka baris perintah dengan menu berlapis di atas).

Khususnya di Paradox 1.0, dan 2.0 pengguna dan manual pemrograman memenangkan penghargaan dibaca [rujukan?] – Deras diilustrasikan mereka, baik meletakkan dan penjelasan ditulis dalam bahasa Inggris umum.

Paradox Untuk Windows

Paradox untuk Windows adalah produk jelas berbeda dihasilkan oleh tim yang berbeda dari pemrogram. Meskipun fitur utama dari produk DOS, QBE dan mesin database, adalah port menjaga kode DOS, ada sebuah terobosan besar dalam kompatibilitas dari PAL ke ObjectPAL dan perpindahan ke metafora desain GUI untuk Formulir dan Laporan. Perubahan ObjectPAL kontroversial tapi terpaksa karena PAL didasarkan pada tindakan merekam keystroke yang tidak setara di Windows. Sebuah bahasa pemrograman berbasis berdasarkan ide dari Hypercard digunakan di tempat merekam keystroke. Bentuk-bentuk dan Laporan desainer digunakan perangkat independen scaling termasuk kemampuan untuk bekerja dalam mode yang diperbesar untuk tata letak rinci. Hak-klik mouse digunakan untuk mengakses Formulir dan properti Laporan, terinspirasi oleh Xerox Alto dan Smalltalk, dengan cara yang sekarang hampir universal untuk program Windows. The ObjectPAL adalah (seperti Hypercard) terkait dengan objek visual – juga mengungkapkan dengan klik kanan. Properti inspeksi dan alat-alat tata letak bisa “ditempelkan up” untuk tinggal di layar, ide yang dipinjam dari NeXT dan sekarang cukup banyak digunakan pada Windows.

Untuk tahun pertama sekitar pembangunan kode berorientasi obyek ditulis dalam C dibantu oleh macro, sampai Turbo C + + yang tersedia di titik mana sisa bagian kode ditulis dalam C + +. Para

manajer produk sampai pengiriman versi 1.0 Joe Duncan. Pengembangan dan tim QA berjumlah sekitar 30 orang.

Kedua Paradox untuk Windows dan Quattro Pro untuk Windows, sebuah proyek erat terkait, mulai pembangunan menggunakan versi beta dari Windows 3.0, pada musim semi 1990. Paradoks / Windows berakhir sampai tertunda sekitar satu tahun di luar rencana semula, pengiriman tahun 1993 awal. Alasan yang banyak, tapi tidak sepenuhnya mengejutkan untuk penulisan ulang besar, dalam bahasa OO dengan alat baru, beralih ke paradigma GUI, pada dasarnya apa yang sistem operasi versi pertama. Namun hal ini merupakan masalah besar bagi perusahaan dan Microsoft Access Pengarungan berhasil beberapa bulan menjelang Paradox untuk Windows, pemasaran utama dimenangkan oleh Microsoft.

Pada tahun 1990 Borland juga mulai bekerja pada sebuah tiruan dBASE internal untuk kedua DOS dan Windows, ditulis dalam assembler, yang direncanakan untuk kapal pada tahun 1992. Pada awal tahun 1992 menjadi jelas bahwa Ashton-Tate dalam kesulitan pada pengembangan versi Windows produk mereka dan diaktifkan Borland rencana, bukan mengakuisisi perusahaan dan pengurapan proyek internal mereka sebagai penerus resmi. Bagian dari akuisisi Ashton-Tate adalah database Interbase dan diputuskan bahwa Paradox / W harus dapat bekerja dengan Interbase serta mesin Paradox dan hal ini menyebabkan terciptanya sebuah mesin IDAPI berbasis di sekitar Interbase.

Meskipun padat tindak pada versi dengan perbaikan untuk kegunaan untuk pengguna entry-level, Paradox memudar dari pasar. Tersebut termasuk dalam penjualan produk-produk Borland ke WordPerfect, yang pada gilirannya dijual kembali sebagai WordPerfect masuk ke produk-produk keuangan, dan pada saat saat menulis Paradox untuk Windows, WordPerfect dan Quattro Pro untuk Windows semua dimiliki oleh Corel dan dijual sebagai bagian suite kantor mereka. dBASE untuk Windows datang terlambat untuk menjadi pemain signifikan di pasar Windows, paling dBASE programmer saat itu telah pindah ke Microsoft FoxBASE, alat database yang sangat mirip. Borland sendiri mempertahankan Interbase / server IDAPI dan memfokuskan upaya pada alat nya Delphi yang selama bertahun-tahun memberinya sebuah berpengaruh tapi kecil bagian dari pasar data berorientasi pengembang.

Corel Paradox

Corel memperoleh hak tertentu untuk mengembangkan dan memasarkan Paradox pada pertengahan tahun 90-an dan merilis Corel Paradox 8 pada tahun 1997. Paradoks ini juga dibundel dalam versi profesional yang WordPerfect Office suite. Hal ini telah merilis versi Office Professional Edition 9, 10, 11, 12, X3 dan X4. Versi terbaru dari Paradoks dirilis oleh Corel adalah 11.0.0.663 yang dipaketkan dengan Service Pack 1 X4

Paradox Users

Ada Paradox basis pengguna yang kuat, terutama berpusat di sekitar Komunitas Paradox dan newsgroup yang berhubungan. Banyak yang merasa dikecewakan oleh Borland dan Corel karena mereka percaya bahwa Paradox lebih unggul daripada semua DBMSes desktop lain di sekitar. [Rujukan?]

Meskipun ada banyak penggemar ObjectPAL, bahasa pemrograman untuk Paradox / Windows, PAL / script DOS tidak dapat dengan mudah bermigrasi; model objek dan peristiwa-benar berbeda memaksa pengembang menggunakan PAL untuk menulis ulang sepenuhnya aplikasi database mereka.