PEMBANGUNAN PROTOTAIP APLIKASI JAVA

UNTUK SISTEM PENGELUARAN

MOHD NAZRI BIN SAAD

UNIVERSITI UTARA MALAYSIA
2001

Sekolah Siswazah
(Graduate School)
Universiti Utara Malaysia

PERAKUAN KERJA KERTAS PROJEK
(Certification of Project Paper)

Saya, yang bertandatangan, memperakukan bahawa
(I, the undersigned, certify that)

MOHD NAZRI BIN SAAD

calon untuk Ijazah
(candidate for the degree of) Sarjana Sains (Teknologi Maklumat)

telah mengemukakan kertas projek yang bertajuk
(has presented his/her project paper of the following title)

PEMBANGUNAN PROTOTAIP APLIKASI JAVA UNTUK SISTEM PENGELUARAN

seperti yang tercatat di muka surat tajuk dan kulit kertas projek
(as it appears on the title page and front cover of project paper)

bahawa kertas projek tersebut boleh diterima dari segi bentuk serta kandungan, dan meliputi bidang ilmu dengan memuaskan.
(that the project paper acceptable in form and content, and that a satisfactory knowledge of the field is covered by the project paper).

Nama Penyelia
(Name of Supervisor) : En. Abd Nasir bin Zulkifli

Tandatangan
(Signature) : Abd. Nasir

Tarikh
(Date) : 10-10-01

PEMBANGUNAN PROTOTAIP APLIKASI JAVA UNTUK SISTEM PENGELUARAN

Projek Sarjana ini dikemukakan kepada Sekolah Siswazah
merupakan sebahagian keperluan penganugerahan
Ijazah Sarjana Sains (Teknologi Maklumat)
Universiti Utara Malaysia

Oleh

Mohd Nazri Bin Saad

© Mohd Nazri Bin Saad, 2001. Hak Cipta Terpelihara.

KEBENARAN MENGGUNAKAN LAPORAN PROJEK

Laporan ini merupakan sebahagian daripada syarat pengijazahan program Sains Sarjana (Teknologi Maklumat) Universiti Utara Malaysia. Dengan ini saya bersetuju untuk membenarkan pihak perpustakaan untuk mempamerkannya sebagai bahan rujukan umum. Saya juga bersetuju membenarkan mana-mana pihak membuat salinan samada sebahagian atau keseluruhan projek ini bagi tujuan akademik dengan mendapat kebenaran daripada penyelia projek ataupun melalui Dekan Sekolah Siswazah. Sebarang bentuk cetakan atau salinan bagi tujuan komersial adalah dilarang tanpa merujuk kepada penyelidik.

Kebenaran perlu diperolehi untuk menyalin atau menggunakan projek ini, sebahagian atau keseluruhannya. Bagi tujuan rujukan sebagai ulasan karya di dalam sebarang laporan, rujukan kepada penulis dan Universiti Utara Malaysia perlu dinyatakan di dalam laporan berkenaan.

Dekan Sekolah Siswazah

Universiti Utara Malaysia

06010 UUM, Sintok

Kedah DarulAman

ABSTRAK

Projek ini adalah sebahagian daripada syarat penganugerahan Sarjana Sains Teknologi Maklumat dibawah Sekolah Siswazah, Universiti Utara Malaysia. Tajuk projek ini adalah Pembangunan Prototaip Aplikasi Java Untuk Sistem Pengeluaran yang akan digunakan untuk tujuan mengoptimumkan operasi pengeluaran yang berdasarkan penggunaan bahan mentah, kos dan juga penggunaan tenaga kerja seperti buruh yang bersesuaian dengan menggunakan sistem maklumat (aplikasi komputer) bagi menjamin produk yang dihasilkan dapat bersaing apabila berada di pasaran.

Di dalam projek ini penekanan dibuat terhadap pembangunan satu sistem maklumat yang menggunakan teknologi pengaturcaraan komputer untuk sistem pengeluaran yang dapat berfungsi sepenuhnya di dalam sesuatu industri pengeluaran.

Adalah diharapkan dengan terlaksananya projek ini ianya akan dapat memperkenalkan satu lagi teknologi komputer khususnya sistem maklumat yang akan digunakan di dalam bidang industri pengeluaran di dalam pembuatan sesuatu barangan/produk.

ABTRACT

This project is under Graduate School to complete the MSc.IT program. This project title is building the prototype of Java application in production system database.

First at all this project will determine the basic concept of production system design and analysis whereby to capture the area of business activities and to give some solution in the area of production system database. That means this project can proceed the newest technology in system database design like Java application.

After that this project can show the effectiveness of Java language using in the database system especially in Java 2 software. That means this project also defined and determines useful of Java language such as graphical analysis, embedded database and networks capabilities.

This project also captured all of production database needed and the solution is the Java programming language. I hope this project can give the newest database technology development especially in the production environment.

PENGHARGAAN

Bersyukur kehadrat Allah S.W.T diatas taufik dan hidayah-Nya serta selawat serta salam terhadap Rasulluhlah S.A.W.

Alhamdulillah dengan limpah kurnianya projek Sarjana Sains ini telah dapat dilaksanakan dengan jayanya dan ucapan terima kasih kepada semua pihak yang terlibat sama ada secara langsung atau tidak langsung di dalam menjayakan projek ini.

Dikesempatan yang terhad ini, diucapkan jutaan terima kasih kepada En. Abd Nasir Zulkifli selaku penyelia yang banyak membantu dalam menjayakan projek ini dan terima kasih diucapkan.

Segala sokongan dan semangat dari keluarga dan rakan-rakan amat dihargai dengan ucapan ribuan terima kasih.

SENARAI RAJAH

		Halaman
Rajah 1.1	Senibina asas teknologi JDBC menggunakan "PureJjava JDBC Driver " (http://java.sun.com)	4
Rajah 1.2.	Gabungan dua driver yang utama iaitu JDBC dan juga ODBC yang telah sedia ada didalam platfom Windows	5
Rajah 1.3	Senibina Asas Aplikasi Java Yang akan Direkabentuk	6
Rajah 2.1	Perkara-perkara asas didalam sistem pengeluran termasuk operasi, permintaan (demand) dan lain-lain. (Dewan, 1997)	18
Rajah 3.1	Paparan bagi menu utama sistem asas (datasource)	28
Rajah 3.2	Paparan Kemasukkan Rekod	28
Rajah 3.3	Paparan Carian Maklumat/Rekod	29
Rajah 3.4	Borang Pesanan	30
Rajah 3.5	Borang Bagi Produk	31
Rajah 3.6	Borang bagi maklumat operasi pengeluaran	32
Rajah 3.7	Borang rekod pekerja/operator pengeluaran	33
Rajah 3.8	Borang kermasukkan rekod keperluan bahan mentah, operasi pengeluaran yang terlibat berdasarkan produk	34
Rajah 3.9	Borang kemasukkan rekod bagi stok bahan mentah	35
Rajah 3.10	Borang pesanan bahan mentah	36

Rajah 3.1	Borang kemasukkan rekod invoice	37
Rajah 4.1	Paparan Utama bagi aplikasi java	39
Rajah 4.2	Table Display	40
Rajah 4.3	Paparan untuk Query 1	41
Rajah 4.4	Paparan bagi query 2	42
Rajah 4.5	Paparan bagi query 3	43
Rajah 4.6	Paparan bagi query 4	44
Rajah 4.7	Paparan borang pengisian rekod bagi produk yang hendak dihasilkan	45
Rajah 4.8	Paparan bagi borang pesanan	46
Rajah 4.9	Contoh paparan perancangan yang dihasilkan melaui sistem yang akan digunakan untuk proses pembuatan seperti tarikh awal operasi pengeluaran, tarikh lewat dan juga kuatiti pesanan	47
Rajah 5.1	Menu Utama bagi sistem untuk pesanan produk terus kepada sistem	49
Rajah 5.2	Penganalisaan Penggunaan Bahan Mentah Melalui Sistem	50

KANDUNGAN

		Muka surat
Kebenaran Menggunakan Laporan Projek		i
Abstrak (Bahasa Malaysia)		ii
Abstract (English)		iii
Penghargaan		iv
Senarai Rajah		v
Kandungan		vii
BAB 1		
PENGENALAN KEPADA APLIKASI JAVA		
1.1	Mengapa Pentingnya Bahasa Java	1
1.2	Senibina *JDBC API*	2
1.3	Kesesuaian Penggunaan Aplikasi Java Untuk Sistem Kajian	8
1.4	Objektif projek	9
1.5	Pernyataan masalah	9
BAB 2		
REKABENTUK PANGKALAN DATA ASAS SISTEM PENGELUARAN (DATASOURCE)		
2.1	Sistem Pengeluaran Secara Am	10
2.2	Ciri – Ciri Sistem yang akan dibangunkan	11
2.3	Operasi Asas Sistem Yang Akan Dibangunkan	14
2.4	Entiti-Entiti yang Terlibat	16
2.5	Strutur Unit Organisasi Industri	19

2.6 *Process Modelling* bagi Sistem Kajian 20

2.7 *Data Modeling* 21

2.8 ***Key-based Data Model*** 22

2.9 ***Fully Attributs Data Model*** 23

BAB 3

REKABENTUK FIZIKAL PANGKALAN DATA (REPOSITORI ASAS SISTEM PENGELUARAN)

3.1 Paparan Utama Repositori Asas 24

3.2 Paparan Kemasukkan Rekod 26

BAB 4

REKABENTUK PROTOTYPE APLIKASI JAVA

4.1 Paparan Utama Sistem Pengeluaran 34

4.2 Contoh-contoh *Query* yang dihasilkan melalui *SQL* 37

4.3 Borang Pengisisan Rekod Melalui Aplikasi Java 41

4.2 Maklumat Yang Boleh Didapati Daripada Sistem Secara Terus 43

BAB 5

APPLET APPLICATION 44

BAB 6

KESIMPULAN 46

RUJUKAN

APPENDIKS

BAB 1

PENGENALAN

1.0 Pengenalan

Secara umumnya aplikasi Java dihasilkan melalui pengaturcaraan sepenuhnya dan selalunya dibahagikan kepada beberapa pakej yang utama iaitu *Java Virtual Machine*, *JDBC API* dan *Java Class File*. Kesemua pakej tersebut selalunya digabungkan menjadi satu aplikasi sama ada untuk tujuan capaian rekod, penganalisaan, penyelesaian masalah dan juga aplikasi di dalam sistem maklumat dan oleh itu ianya memerlukan kepakaran dan kebolehan yang tinggi semasa merekabentuk sistem terutamanya dalam pengaturcaraan Java. Oleh itu projek ini akan menjadikan Java sebagai satu alternatif untuk merekabentuk dan membangunkan sistem maklumat yang dianggap penting dan kompleks (Lu S.C.-Y *et.at all,* 1990) berbanding dengan sistem maklumat yang lain khususnya untuk kegunaan di dalam bidang pengeluaran ataupun industri. Ini adalah perlu kerana operasi bagi sistem maklumat yang akan dibangunkan adalah amat kompleks dan sibuk dan memerlukan aplikasi yang agak canggih

terutamanya di dalam penganalisaan data yang dihasilkan terus daripada sistem maklumat secara automatik. Di dalam bab ini juga disertakan perbincangan tentang senibina asas sistem terutamanya Aplikasi Java.

1.0.1 Mengapa Pentingnya Bahasa Java

Bahasa Java adalah bahasa yang *universal* (Dittel, 2000) yang mampu menghasilkan apa-apa jua aplikasi komputer melalui pengaturcaraan sepenuhnya. Sekarang ini aplikasi Java boleh digunakan bersama senibina rangkaian yang berbagai jenis terutamanya senibina rangkaian *Client Access Computing* ataupun *Network Computing* (http://java.sun.com) yang mana ianya mempunyai sifat-sifat seperti berikut:

- *Platform Independence*
- *Network Mobility*
- Keselamatan

(http://java.sun.com)

1.0.2 Senibina *JDBC API*

Secara umumnya JDBC digunakan untuk:

- Menyambungkan antara pangkalan data (*datasource*) yang berbilang jenis dengan aplikasi dengan menggunakan rangkaian setempat (*local area network*).

- Capaian data melalui SQL (*structured query language*) daripada pangkalan data asas (*data source*).

- Memproses data yang telah dicapai melalui capaian data tersebut melalui antaramuka yang telah direkabentuk secara grafik ataupun applet.

(Lu S.C.-Y *et at al,* 1990)

Di dalam projek ini JDBC digunakan sebagai aplikasi tambahan (*juga sebagai keperluan*) terhadap penggunaan pangkalan data bagi sistem pengeluaran seperti penganalisaan secara grafik, pengiraan dan aplikasi pangkalan data yang serba lengkap yang tidak ada pada perisisan pangkalan data yang berada di pasaran pada masa sekarang.

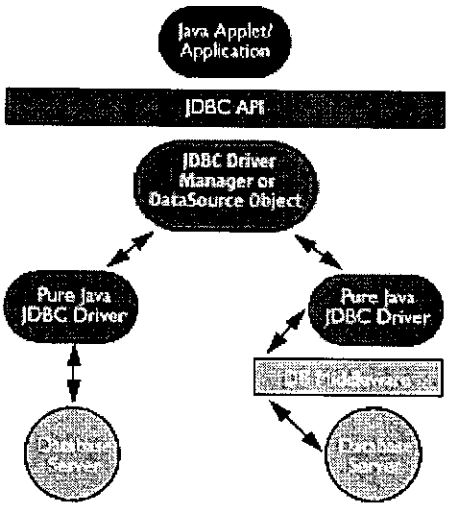
Di dalam projek ini juga pangkalan data asas (*datasource*) direkabentuk dengan menggunakan *MS Access* yang akan digunakan bersama platform *Windows*.

Secara umumnya JDBC API mempunyai dua antaramuka major yang utama iaitu:

- JDBC untuk penulisan aplikasi (*digunakan untuk projek ini*).
- "*lower-lavel*" untuk JDBC driver dan hanya untuk sambungan tanpa menggunakan antaramuka.

JDBC juga mempunyai dua kategori yang utama iaitu:

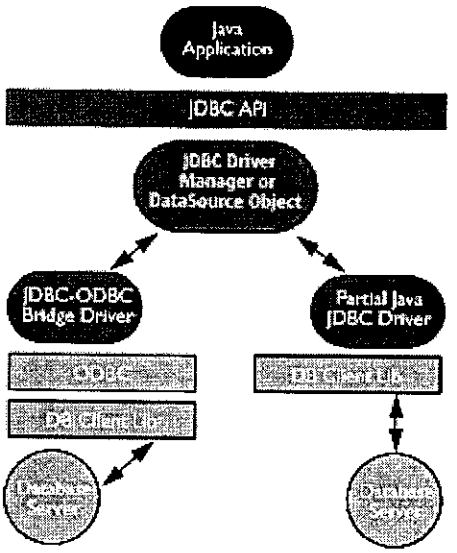
1. Aplikasi dan juga Applet yang digunakan untuk capaian data daripada pangkalan data dengan menggunakan teknologi "JDBC driver" yang tulen seperti yang ditunjukkan di dalam gambarajah 1 samada menggunakan *Database Middleware* (*Client Access Computing*) ataupun tidak.



Rajah 1.1 Senibina asas teknologi JDBC menggunakan "Pure Java JDBC Driver "

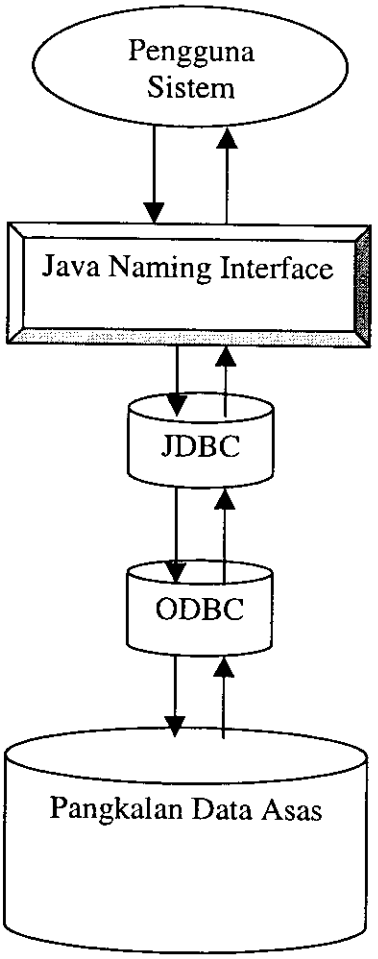
(http://java.sun.com)

2. Gabungan dua driver yang utama iaitu ODBC-JDBC driver yang digunakan bersama dan kaedah ini akan digunakan bagi merekabentuk sistem pengeluaran seperti yang ditunjukkan di dalam rajah 2.



Rajah 1.2. Gabungan dua driver yang utama iaitu JDBC dan juga ODBC yang telah sedia ada di dalam platfom Windows (http://java.sun.com)

Di bawah ini adalah gambaran bagi sistem pengeluaran yang akan direkabentuk dan ilustrasinya adalah seperti ditunjukkan di dalam rajah 3.



Rajah 1.3. Senibina Asas Aplikasi Java Yang akan Direkabentuk

Di dalam *rajah 3* Pengguna Sistem (*Client*) akan mendapatkan maklumat atau mendapatkan *service* melalui *Java Naming Interface* yang terdapat di dalam pakej *Swing* dan *AWT* (*Abtract Window Toolkit*) didalam bahasa Java. Antaramuka ini boleh dihasilkan melalui pengaturcaraan Java sepenuhnya. Berikut adalah pakej yang terdapat di dalam pengaturcaraan Java untuk *Java Naming Interface*:

```
import java awt.*;
import java awt.event.*;
import javax swing.*;
import javax swing.table.*;
```

Sistem yang akan direkabentuk juga akan menggunakan Windows sebagai platfom untuk aplikasi sistem pengeluaran. Oleh itu diperlukan *driver* untuk capaian yang berada di dalam repositori utama (*datasource*) yang dihasilkan dengan menggunakan Microsoft Access. *Driver* tersebut adalah JDBC untuk aplikasi Java dan ODBC yang telah sedia ada di dalam platfom Windows.

Di dalam platfom Windows, berikut adalah pakej yang akan digunakan untuk capaian data:

```
import java.sql.*;
```

dan untuk tujuan capaian data adalah perlu menggunakan dan menitikberatkan jenis-jenis data yang akan dicapai sebagai contohnya untuk data yang berbentuk tarikh dan pakejnya adalah seperti berikut:

```
import java.sql.Date.*;
```

dan pernyataan untuk sambungan antara *Java Bridge* dengan ODBC adalah:

```
// Set up database connection
      try {
         url = "jdbc:odbc:ProductionSystem1";

         Class.forName(  "sun.jdbc.odbc.JdbcOdbcDriver"
);
         connect = DriverManager.getConnection( url );
         output.append( "Connection successful\n" );
      }
      catch ( ClassNotFoundException cnfex ) {
         // process ClassNotFoundExceptions here
         cnfex.printStackTrace();
         output.append( "Connection unsuccessful\n" +
                        cnfex.toString() );
      }
      catch ( SQLException sqlex ) {
         // process SQLExceptions here
         sqlex.printStackTrace();
         output.append( "Connection unsuccessful\n" +
                        sqlex.toString() );
      }
```

```
catch ( Exception ex ) {
    // process remaining Exceptions here
    ex.printStackTrace();
    output.append( ex.toString() );
}
```

1.1 Kesesuaian Penggunaan Aplikasi Java Untuk Sistem Kajian

Di dalam sistem pengeluaran yang juga dipanggil *Crash & Burn environment* ataupun dengan bahasa yang mudah yang dipanggil suasana tunggal (*stand-alone environment*) yang digunakan bersama DBMS dan juga perisian-perisian lain dengan hanya satu tujuan yang utama iaitu mengoptimakan pengeluaran dengan pengurangan kos pegeluaran, penggunaan tenaga kerja yang efektif dan juga penggunaan bahan mentah yang optimum. Di dalam suasana ini (*C&B environment*) pelbagai unit ataupun segmen bagi operasi pengeluaran yang terlibat adalah rumit dan digabungkan kesemuanya melalui integrasi berkomputer yang mempunyai beberapa platfom seperti *Sun Solaris* ataupun *Unixs* (Houtzeel A, 1976).

Aplikasi-aplikasi tersebut mempunyai tempat penyimpanan data yang juga dipanggil pangkalan data asas (*datasource*) yang bilangannya bergantung kepada saiz industri ataupun operasi syarikat dan juga permintaan daripada pengguna sistem tersebut. Oleh itu teknologi dalam pembinaan sistem maklumat perlulah sesuai dengan penggunaan sistem tersebut terutamanya di dalam bidang industri

dan di dalam sistem ini suasana yang ujud adalah *client-access computing* yang mempunyai lebih daripada satu *workstation* ataupun *paltform* bergantung kepada saiz/operasi syarikat.

Oleh itu secara ringkasnya pengeluar-pengeluar perisisan pangkalan data seperti *Oracle*, *SQL Server* mahupun *Informix* tidak mampu menampung aplikasi yang dikehendaki oleh pengguna sistem terutamanya di dalam rangkaian setempat ataupun "*client access computing*". Dengan ujudnya bahasa Java ia akan dapat menampung segala kekurangan pangkalan data tersebut. Ini adalah kerana bahasa Java boleh diaturcarakan sepenuhnya melalui bahasa pengaturcaraan dan boleh ditambah dengan penggunaan aplikasi keselamatan (*security*) dan juga berapa teknologi rangkaian.

Kesemua ini dapat dilakukan dengan pengaturcaraan sepenuhnya di dalam bahasa Java. Oleh itu projek ini akan menguji kesesuaian bahasa Java sebagai satu alternatif yang paling baik di dalam bidang pengeluaran atupun khususnya bidang industri terutamanya yang berkaitan dengan sistem maklumat.

1.2 Objektif Projek

- Membangunkan sistem pengeluaran (*prototype*) dengan penggunaan aplikasi java sebagai keperluan seperti penggunaan di dalam rangkaian, penganalisaan rekod dan juga *user friendly*.
- Mengkaji keperluan-keperluan sistem pengeluaran dari segi rekabentuk sistem maklumat dan juga penggunaannya serta membangunkan aplikasi-aplikasi penganalisaan data/rekod melalui pengaturcaraan *Applet* yang boleh dihasilkan terus daripada sistem.

1.3 Definisi Masalah

Sistem pengeluaran adalah satu sistem maklumat yang agak komplek berbanding sistem yang lain. Ini adalah kerana sistem tersebut memerlukan aplikasi tambahan untuk tujuan penggunaan penganalisaan rekod, pengiraan dan juga kemudahan pengggunaannya. Sistem tersebut mestilah canggih dan berteknologi tinggi yang dapat menampung segala keperluan pengguna akhir sistem. Oleh itu projek ini akan menggunakan bahasa Java sebagai bahasa pengaturcaraan untuk membangunkan aplikasi tersebut pada sistem pengeluaran.

BAB 2

REKABENTUK PANGKALAN DATA ASAS SISTEM PENGELUARAN (DATASOURCE)

2.0 Pendahuluan

Secara umumnya sistem yang akan direkabentuk perlu dikaji kesemua keperluan penggunaan sistem sama ada dari aspek pengguna ataupun dari segi kos pembangunan sistem tersebut. Di dalam projek Sarjana Sains ini kajian adalah tertumpu kepada keperluan asas sistem pengeluaran dan oleh itu segala faktor yang akan dikaji adalah meliputi semua aktiviti perniagaan yang terlibat di dalam semua jenis industri tanpa mengira jenis produk ataupun jenis industri tersebut.

2.1 Sistem Pengeluaran Secara Umum

Secara umumnya, sistem pengeluaran yang dipunyai oleh sesebuah industri mestilah merangkumi keseluruhan aktiviti syarikat dan juga jabatan-jabatan yang terlibat. Di dalam projek ini antara jabatan-jabatan yang dikenalpasti adalah Jabatan Sumber Manusia (*Head Human Resource*) , Jabatan Bahan Mentah (*Raw Material Department*), Jabatan Pengeluaran yang dibahagikan ke beberapa segmen ataupun seksyen seperti seksysen pemasangan, seksysen pembuatan dan beberapa lagi seksysen yang terlibat dan untuk projek ini kajian adalah terhad kepada keperluan sistem secara umum tanpa mengira jenis produk ataupun jenis industri kerana tumpuan kajian hanyalah kepada aplikasi java yang akan di bangunkan.

2.2 Ciri – Ciri Sistem yang akan dibangunkan

Secara umumnya sistem yang akan dibangunkan mestilah mempunyai ciri-ciri berikut:

- Penyelenggaraan yang mudah
- Senang digunakan
- Mampu menanggung kesemua kehendak pengguna akhir sistem
- Kos yang bersesuaian

(Dini, 1997)

Untuk sistem yang akan dibangunkan ini kajian penggunaan mestilah dilakukan terlebih dahulu kerana ia mempengaruhi kesesuaian penggunaannya.

Sistem pengeluaran secara umumnya adalah mengikut jenis-jenis industri khususnya pengeluaran dan ianya mempunyai jenis-jenis yang tertentu iaitu susunan mengikut produk, mengikut *batch* dan juga susunan mengikut operasi mesin yang digunakan (Dilts, 2000).

Secara umumnya sistem pengeluaran dirujuk terus kepada *master schedule* yang dipunyai oleh sesebuah organisasi pengeluaran tertentu bergantung kepada operasi pengeluaran. Oleh itu tujuan sebenar sistem pengeluaran adalah untuk mengenalpasti keperluan dan penggunaan bahan mentah yang paling ekonomik, dan juga mengawal operasi semasa pengeluaran termasuk pekerja, masa dan juga operasi mesin-mesin yang terlibat dan ianya bergantung kepada pesanan daripada pengguna. Ini bermakna sistem tersebut mestilah merangkumi syeksyen kawalan inventori seperti bahan mentah dan pesanan bahan mentah, pesanan daripada pelanggan dan juga keseluruhan operasi pengeluaran yang terlibat. Berikut adalah perkara-perkara asas yang perlu terdapat di dalam *Master Schedule* bagi merekabentuk sistem pengeluaran yang dapat berfungsi sepenuhnya dan mencapai kehendak pengguna akhir sistem:

- **"Forecast"**

  Forecast dihasilkan untuk item individu ataupun daripada tingkat agregat di dalam sistem perancangan pemasaran. Sistem tersebut kemudiannya menukarkan ia kepada beberapa item yang mempunyai sifat-sifat yang sama.

- **Gross Material Requirement (Keperluan Bahan Mentah Kasar)**

  Keperluan kasar bahan mentah dikira untuk dibeli daripada pembekal yang ditentukan melalui pesanan daripada pelanggan terus daripada sistem (*online*) ataupun melalui manual (*customer-on-hand*). Segala keperluan dikira untuk satu jangkamasa tertentu untuk operasi pengeluaran.

- **Capacity requirement**

  Berdasarkan daripada *forecast,* sistem yang direkabantuk akan mengira kapasiti yang diperlukan untuk mesin, tenaga kerja ataupun pembekal (*sub-contractor*) untuk tujuan pengeluran.

- **Integration of material requirement planning**

  Kuantiti *forecast* daripada beberapa item akan dikira bersama *order-on - hand* daripada pelanggan dan ini adalah *input* bagi sistem yang akan dibangunkan.

- **Bill of material**

  Keperluan bahan dikira daripada jenis barangan yang akan dihasilkan daripada *super bill. Super bill* adalah perlu bagi semua jenis industri kerana ianya mempunyai banyak variasi barangan yang akan dihasilkan.

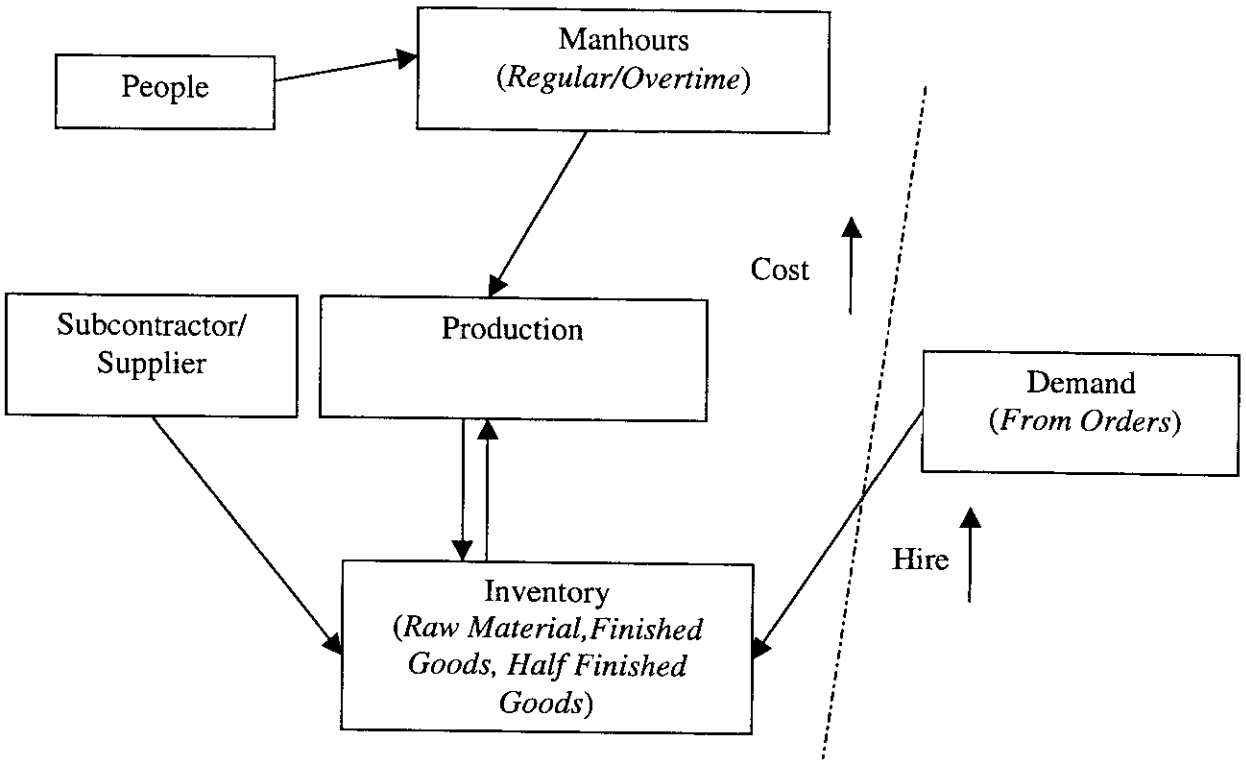
- **System Requirement**

  Di dalam keadaan sebenar pangkalan data ataupun sistem maklumat yang digunakan bergantung kepada keperluan sistem tersebut. Jika diteliti sistem yang sebenar menggunakan *SQL Server* ataupun *Oracle* sebagai pangkalan data utama (*main datasource*) (Feld, 1994) dan juga terdapat berbagai jenis rangkaian komputer yang digunakan bersama. Oleh kerana sistem yang akan dihasilkan agak komplek dengan keperluan penganalisaan data yang diperlukan maka Java adalah alternatif yang terbaik kerana ianya dapat mencakupi pelbagai keperluan khususnya di dalam bidang industri.

2.3 Operasi Asas Sistem Yang Akan Dibangunkan

Di dalam projek ini tumpuan hanya diberikan kepada operasi asas bagi sistem pengeluaran tanpa mengira susunatur kilang atau industri mahupun jenis produk. Berikut adalah operasi asas bagi sistem pengeluaran yang akan direkabentuk dan dibangunkan:

"Pengeluaran mengambilkira jangkamasa tertentu yang diukur di dalam minggu samada di dalam shift yang normal ataupun tambahan masa yang mempunyai tiga shift secara keseluruhannya. Item-item yang lengkap untuk bahan mentah diperolehi terus daripada sub-kontraktor. Sistem yang akan dibangunkan juga akan menentukan jumlah keperluan kos untuk operasi pengeluaran yang melibatkan tenaga kerja, kos mesin, bahan mentah dan lain-lain. Tambahan pesanan (backorders) daripada pelanggan juga diambilkira supaya sistem yang akan dibina mencakupi segala keperluan dari segi pengeluaran. Oleh itu tambahan jangkamasa operasi pengeluaran adalah diperlukan".



Rajah 2.1. Perkara-perkara asas di dalam sistem pengeluran termasuk operasi, permintaan (demand) dan lain-lain. (Dewan, 1997)

Rajah 4 di atas menunjukkan jenis-jenis situasi dan faktor-faktor yang diambilkira semasa operasi pengeluaran dijalankan. Di sebelah kanan adalah melibatkan kos yang agak tinggi dan perlu dititikberatkan manakala di sebelah kiri melibatkan permintaan bahan mentah dan juga produk yang telah siap dan ianya dikawal (*inventory control*) supaya modal pusingan syarikat/organisasi digunakan pada keadaan yang paling maksimum.

Di dalam rajah di atas Pekerja (*People*) bekerja melalui dua shift utama iaitu *Regular Shift* dan *Overtime Shift.* Di dalam sistem pengeluaran yang akan direkabentuk segala maklumat mengenai pekerja diperlukan seperti kos/jam, jawatan, ketua operasi pengeluaran yang selalunya dikenali sebagai *"Shift Supervisor"*, bilangan dan sebagainya untuk mengoptimakan pengeluaran dari segi produktiviti dan kos pengeluaran. Selepas itu sistem yang akan direkabentuk juga akan mengawal kadar inventori berdasarkan pesanan daripada subkontraktor ataupun pembekal , pengeluaran (*seperti barangan separa siap dan juga siap*) dan daripada pesanan daripada pelanggan. Segala analisis mengenai penggunaan sistem akan diterangkan di dalam penggunaan aplikasi Java.

2.4 Entiti-Entiti yang Terlibat

Di dalam sistem asas pengeluaran, antara jabatan yang terlibat adalah Unit Kawalan Inventori, Jabatan Pengeluaran dan Jabatan Sumber Manusia. Oleh itu operasi-operasi yang terlibat adalah seperti di bawah:

- Pesanan daripada pengguna (*Orders*) dengan mengambilkira masa untuk disiapkan, kuatiti pesanan, jenis dan spesifikasi produk yang dilakukan terus daripada sistem ataupun manual.

- Spesifikasi Produk yang hendak dihasilkan seperti Id, jenis, bahan mentah yang diperlukan, masa untuk disiapkan, jenis–jenis proses yang terlibat termasuk masa setiap proses, penggunaan pekerja (*manhours*), *earliest start times* (*EST*), *latest start times* (*LST*), masa bagi proses yang berulang-ulang (*cycle time*) dan sebagainya.

- Kawalan Inventori yang ditentukan nilai minimum dan maksimum yang boleh disimpan untuk produk separa siap ataupun siap dan juga bahan mentah yang mempengaruhi modal pusingan operasi syarikat serta mencukupi untuk operasi pengeluaran. Ini termasuklah penentuan kuantiti pesanan yang menguntungkan bagi operasi pengeluaran yang ditentukan dengan penentuan garis *Reorder point* yang diplotkan pada graf dengan mengambil nilai simpanan inventori bahan mentah, produk separa siap dan juga produk yang telah siap. Ini adalah perlu kerana ia banyak mempengaruhi modal pusingan syarikat.

- Operasi pengeluaran yang terlibat seperti penggunaan mesin, operasi mesin, pemasangan dan juga operasi-operasi lain yang perlu dirancang bagi menjimatkan kos pengeluaran serta meningkatkan produktiviti. Ini termasuklah penentuan jadual yang sesuai dan juga sumber gunatenaga yang paling effektif dan teratur bagi menjamin operasi syarikat dapat berjalan dengan lancar dengan masa yang cepat. Ini adalah kerana masa bagi operasi pengeluaran adalah kos yang perlu ditanggung oleh syarikat

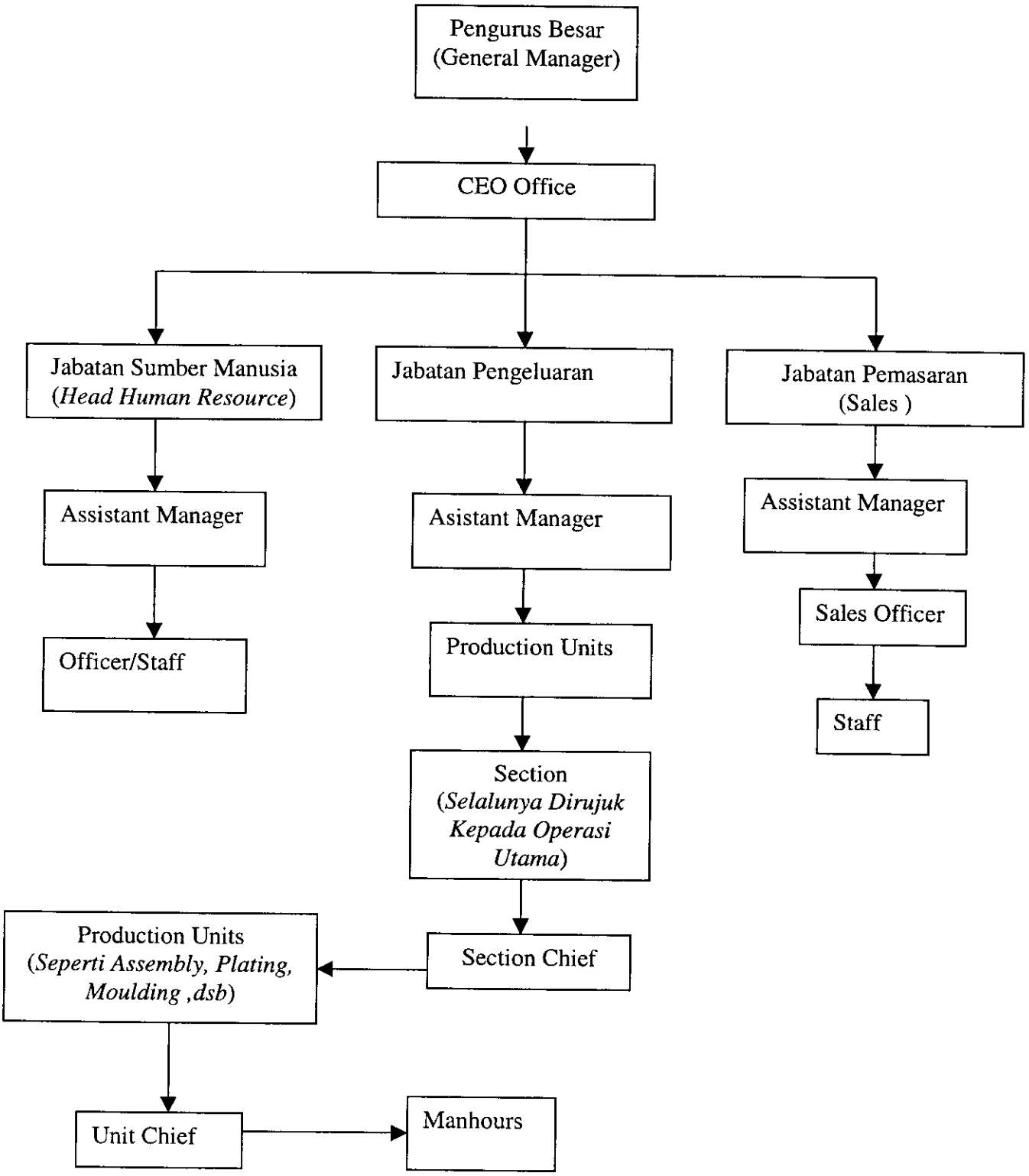
oleh sebab pengiraan kos bagi jangkahayat mesin dan kos bagi pekerja ataupun buruh adalah mengikut jam. Ini penting bagi menjamin kualiti dan mutu barangan (*produk*) yang telah dihasilkan murah apabila berada di pasaran.

- Jenis pelanggan yang terlibat dengan pesanan sama ada secara terus atau manual serta penghantaran produk berdasarkan pesanan dengan menggunakan kapal ataupun pelabuhan yang sesuai (*shipping method*).

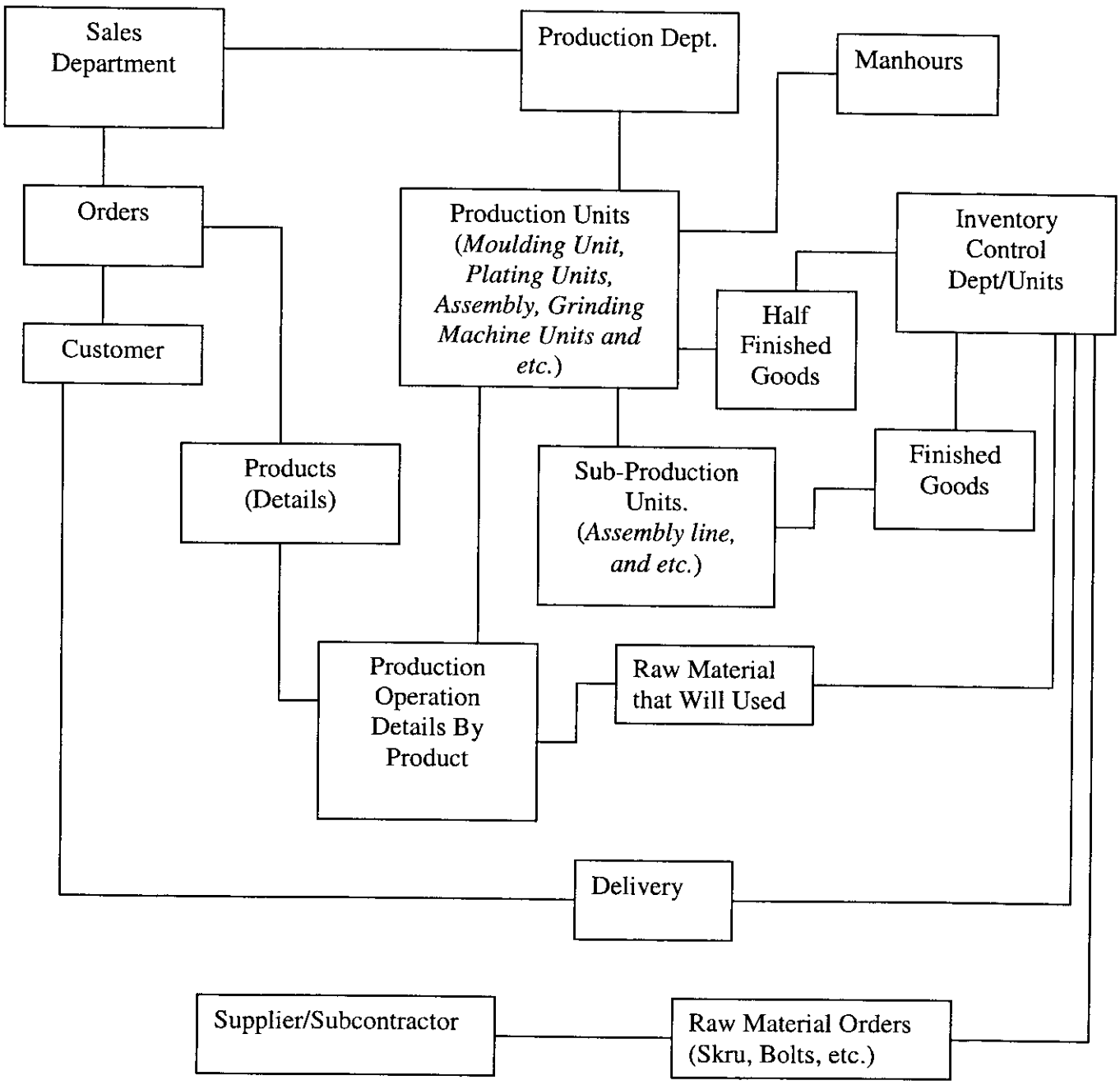
Secara umumnya operasi bagi sistem pengeluaran yang dihasilkan mestilah mempunyai operasi ataupun faktor-faktor di atas dan oleh itu di dalam projek ini kriteria-kriteria di atas digunakan untuk merekabentuk aplikasi java.

2.5 Strutur Unit Organisasi Industri

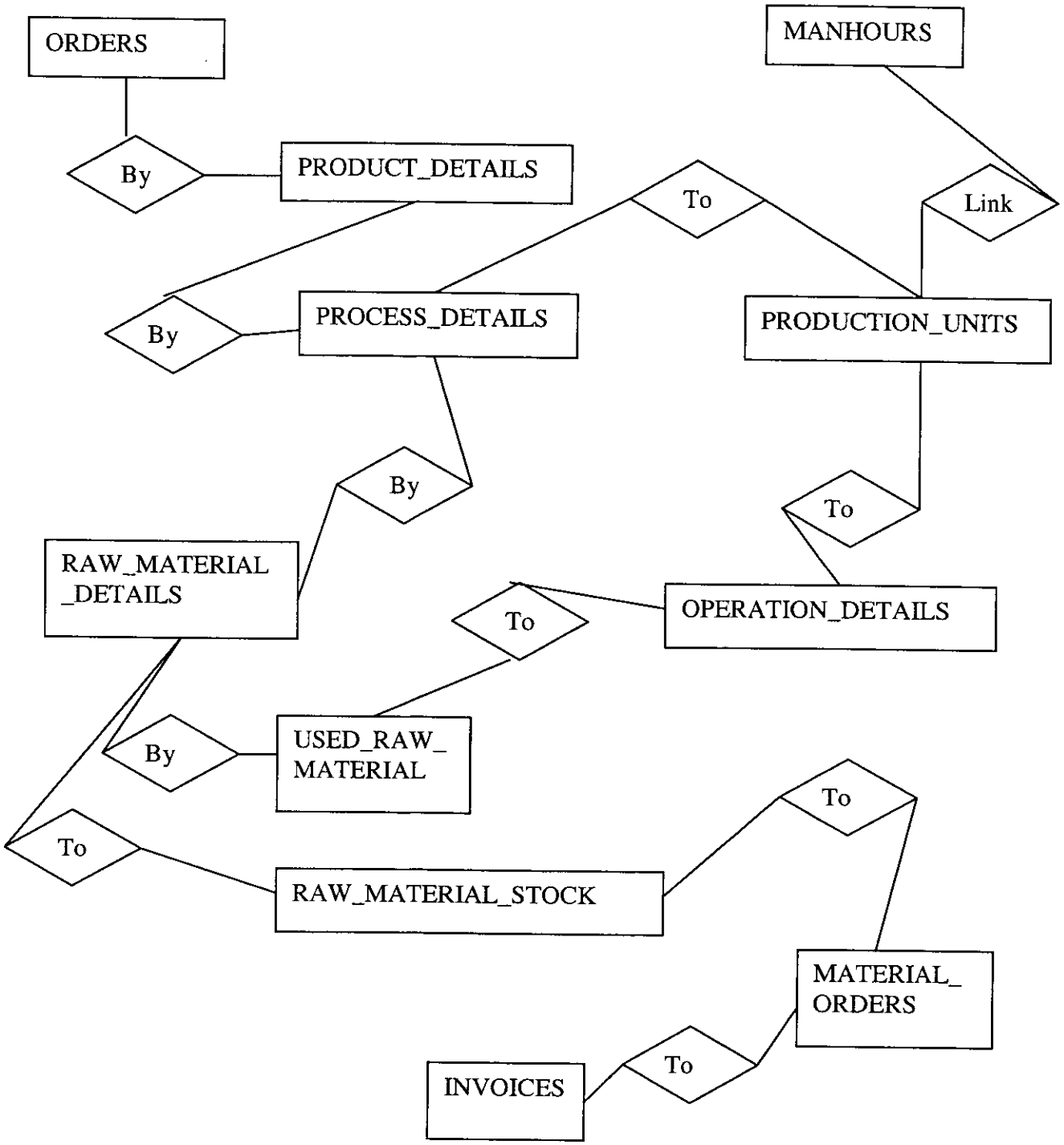
Di bawah ini adalah carta struktur unit bagi operasi di dalam bidang industri terutamanya pengeluaran.



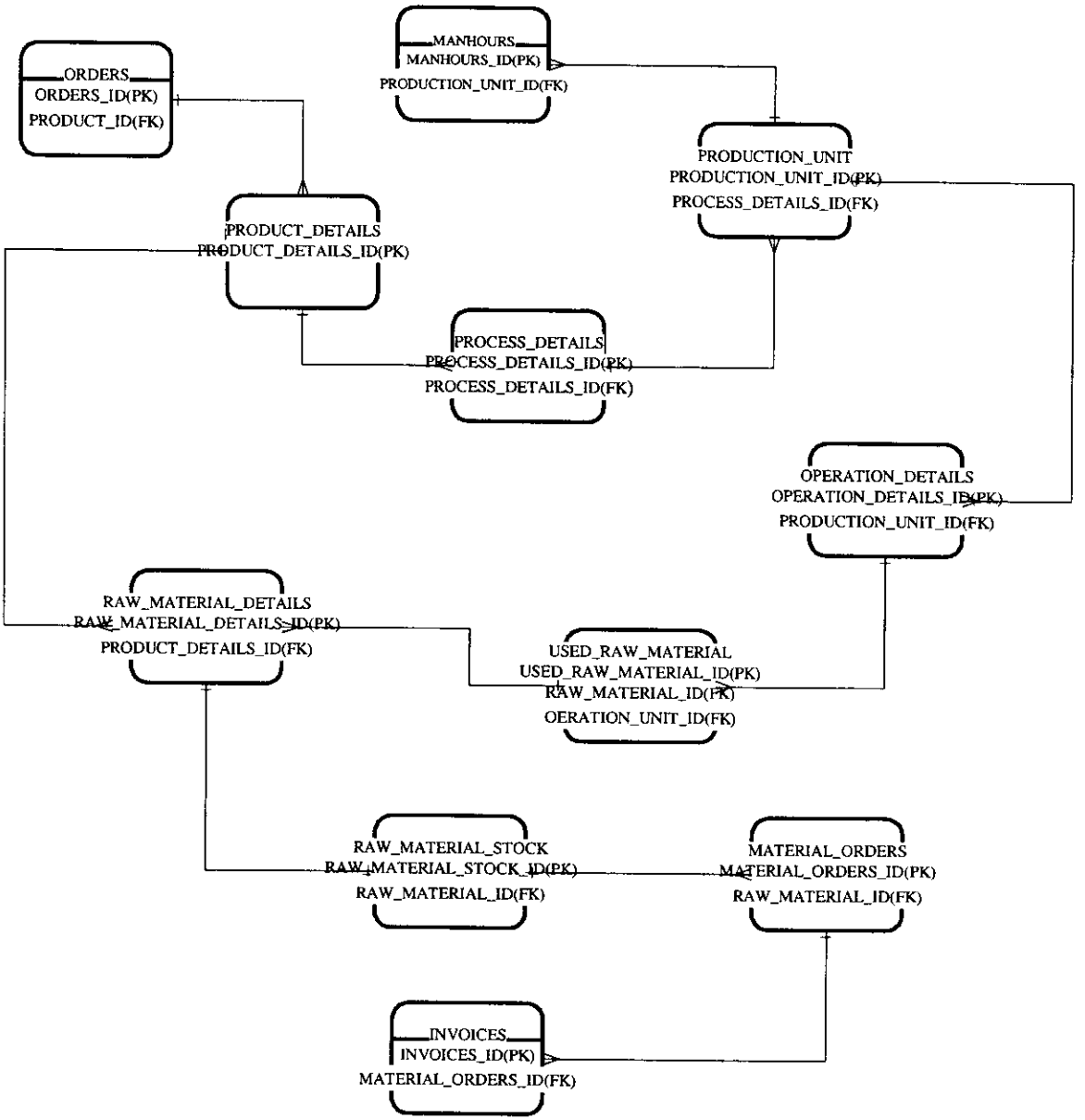
2.6 *Process Modelling* bagi Sistem Kajian



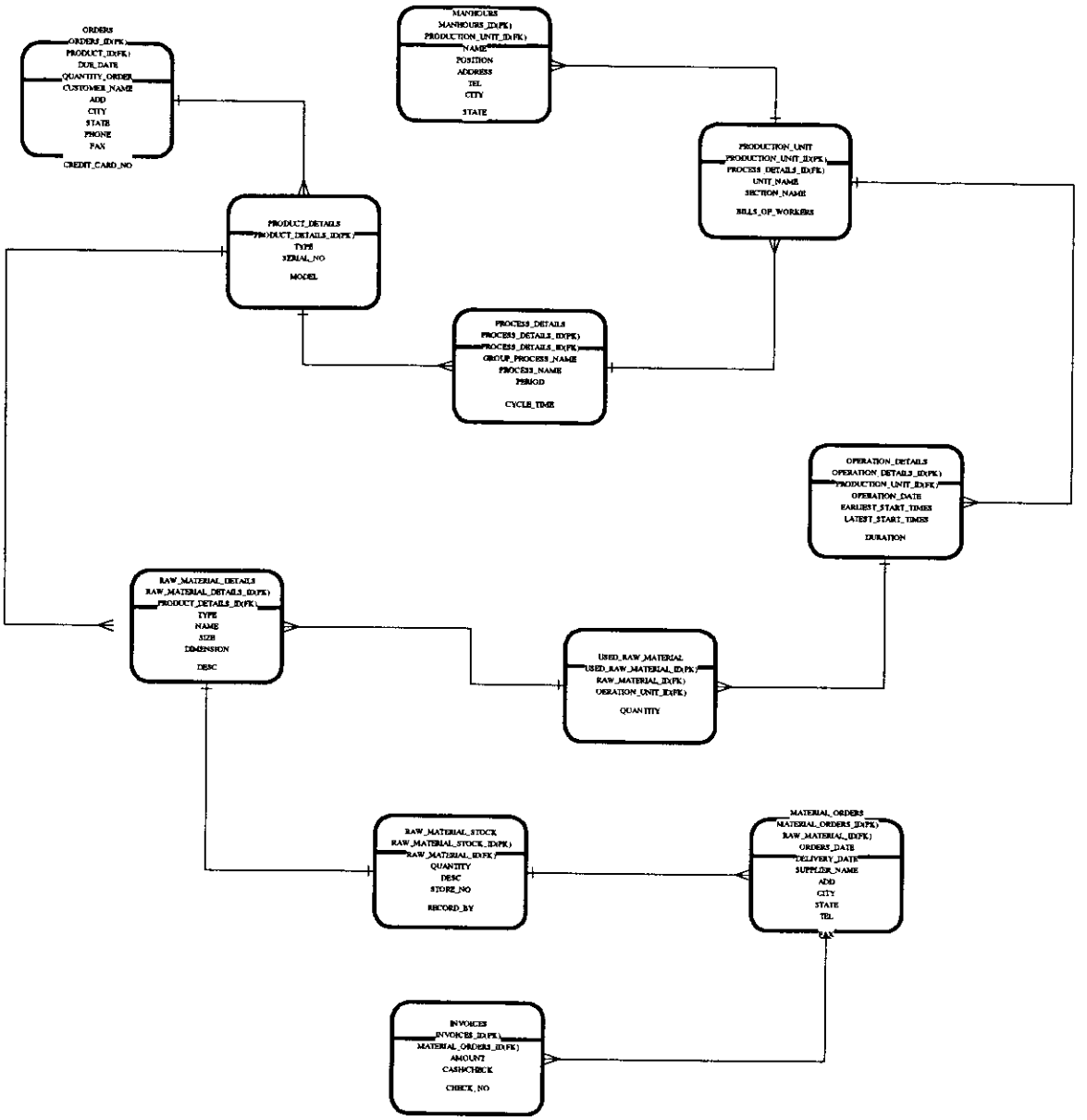
2.7 *Data Modelling*



2.8 *Key-based Data Model*



2.9 *Fully Attributs Data Model*



BAB 3

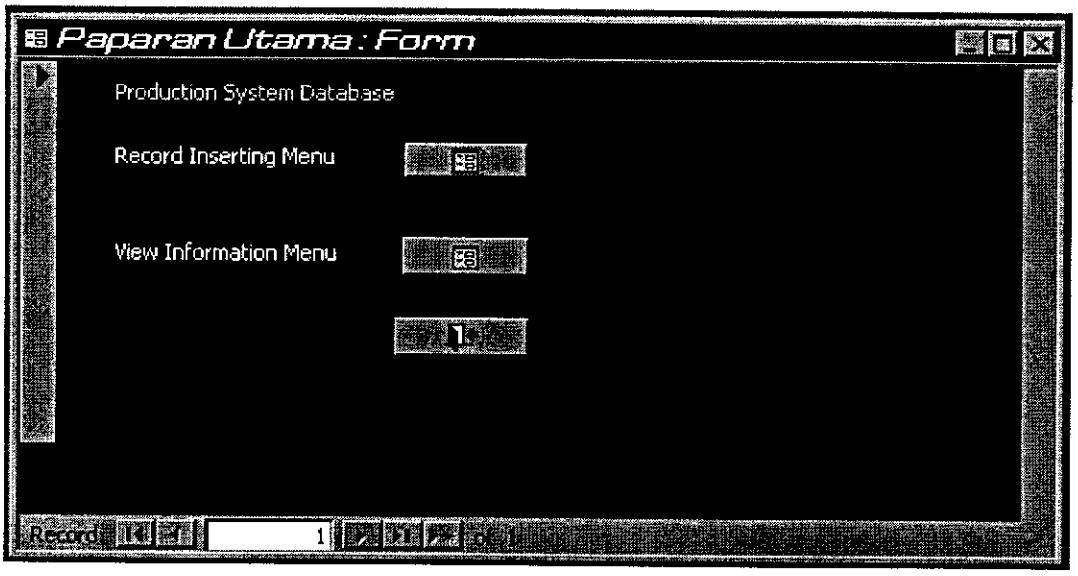
REKABENTUK FIZIKAL PANGKALAN DATA (REPOSITORI ASAS SISTEM PENGELUARAN)

3.0 Pengenalan

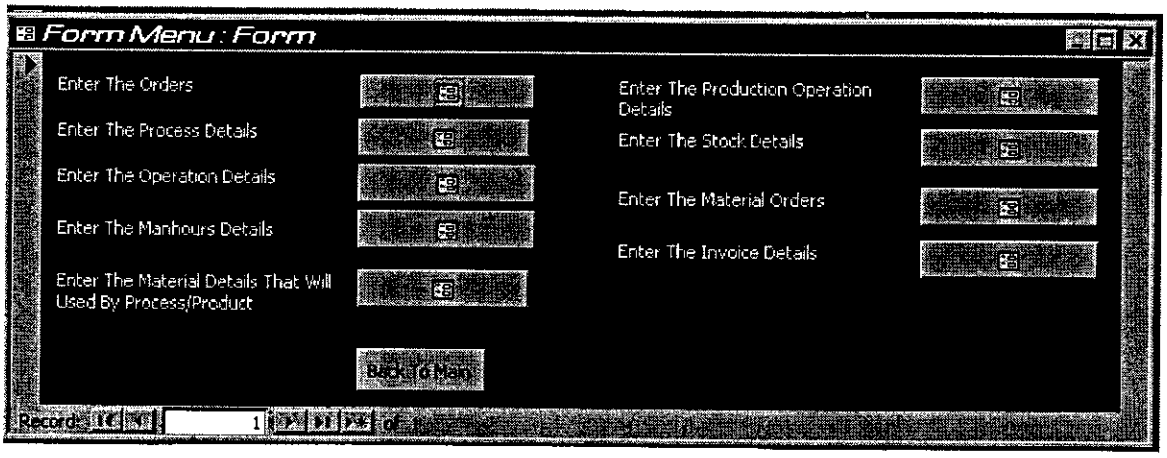
Di dalam bab ini pangkalan data asas yang dihasilkan adalah dengan menggunakan MS Acess sebagai repositori asas sistem pengeluaran. Berdasarkan kajian awal yang telah dilakukan pangkalan data fizikal tersebut mestilah berfungsi sepenuhnya dengan segala keperluan sistem yang telah dikaji sebelumnya. Oleh itu pangkalan data tersebut mestilah senang digunakan semasa ia beroperasi. Oleh itu di dalam bab ini disertakan paparan antaramuka sistem yang telah dibangunkan dan juga dengan penerangan ringkas yang perlu ada untuk setiap sistem.

3.1 Paparan Utama Repositori Asas

Di bawah ini ditunjukkan antaramuka bagi sistem yang telah dibangunkan dengan menggunakan *Ms Access Database*. Di dalam rajah dibawah terdapat dua menu yang utama (*bagi paparan utama*) iaitu paparan kemasukkan rekod dan juga paparan maklumat hasil daripada gabungan kesemua maklumat (*query*).

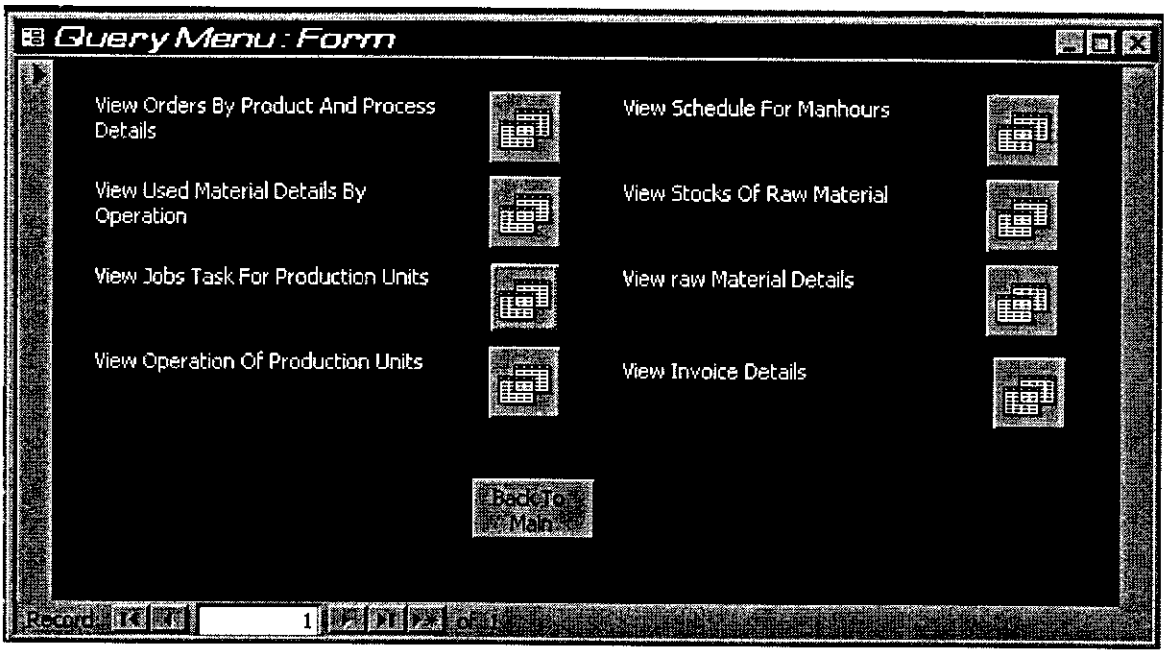


Rajah 3.1 Paparan bagi menu utama sistem asas (datasource)



Rajah 3.2 Paparan Kemasukan Rekod

Untuk paparan kemasukan rekod pengguna akhir sistem perlu memasukkan maklumat seperti pesanan daripada pelanggan, maklumat proses pembuatan produk mengikut jenis produk, maklumat pekerja/buruh (*manhours*), maklumat penggunaan bahan mentah mengikut jenis produk/proses pembuatan, maklumat operasi pembuatan (*details*) dan juga maklumat tentang pesanan bahan mentah dan juga kos yang terlibat semasa pesanan.

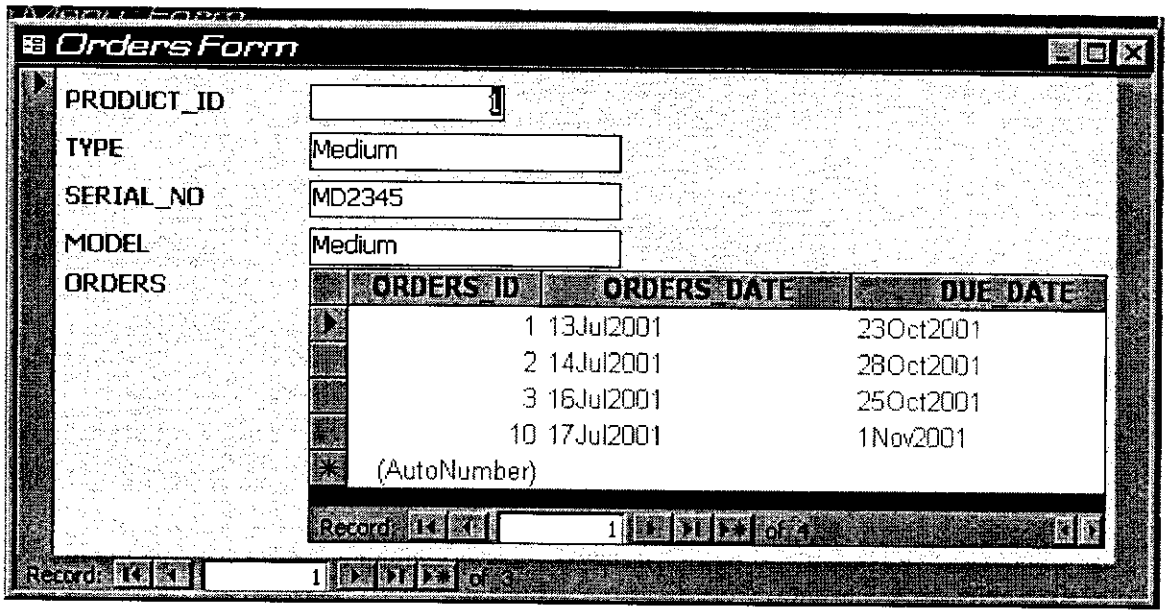


Rajah 3.3 Paparan Carian Maklumat/Rekod

Di atas (*rajah 3.3*) ditunjukkan paparan bagi maklumat yang diperlukan semasa operasi perniagaan dilakukan. Antara maklumat yang boleh didapati adalah maklumat pesanan berdasarkan produk dan juga proses pembuatan secara jelas. Ini adalah perlu kerana pengguna sistem tidak perlu berulangalik semasa menjalankan tugas kerana maklumat ini boleh didapati terus daripada sistem semasa operasi pengeluaran dijalankan.

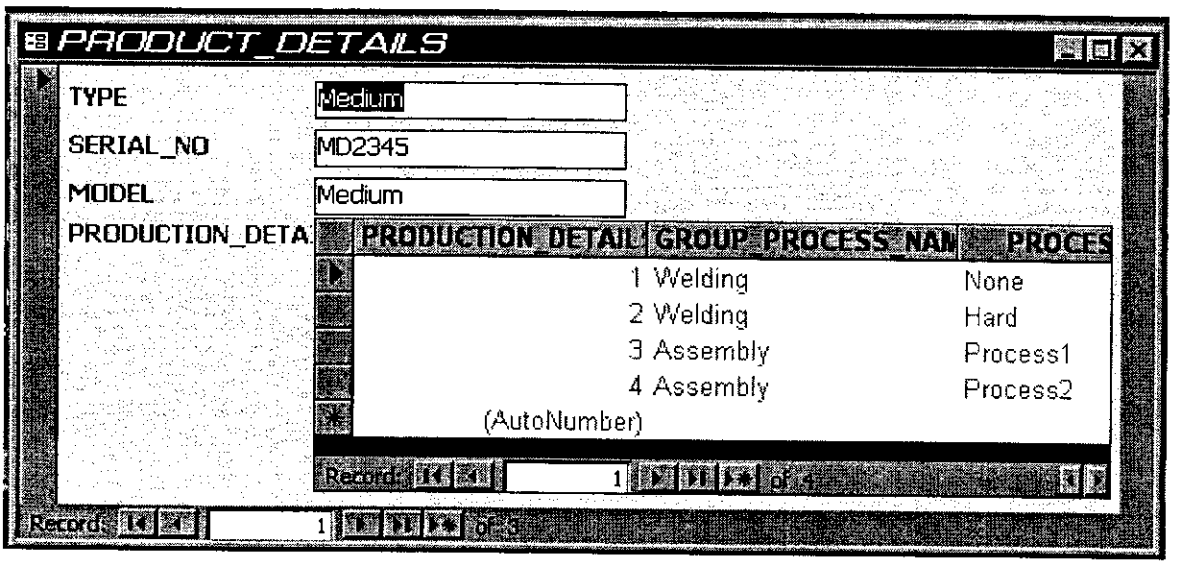
3.2 Paparan Kemasukkan Rekod

Untuk sistem pangkalan data asas (*datasource*) rekod dimasukkan dengan menggunakan borang yang terdapat didalam sistem asas. Oleh itu pekerja atau staf perlu memasukkan maklumat terlebih dahulu sebelum operasi pembuatan dilakukan.



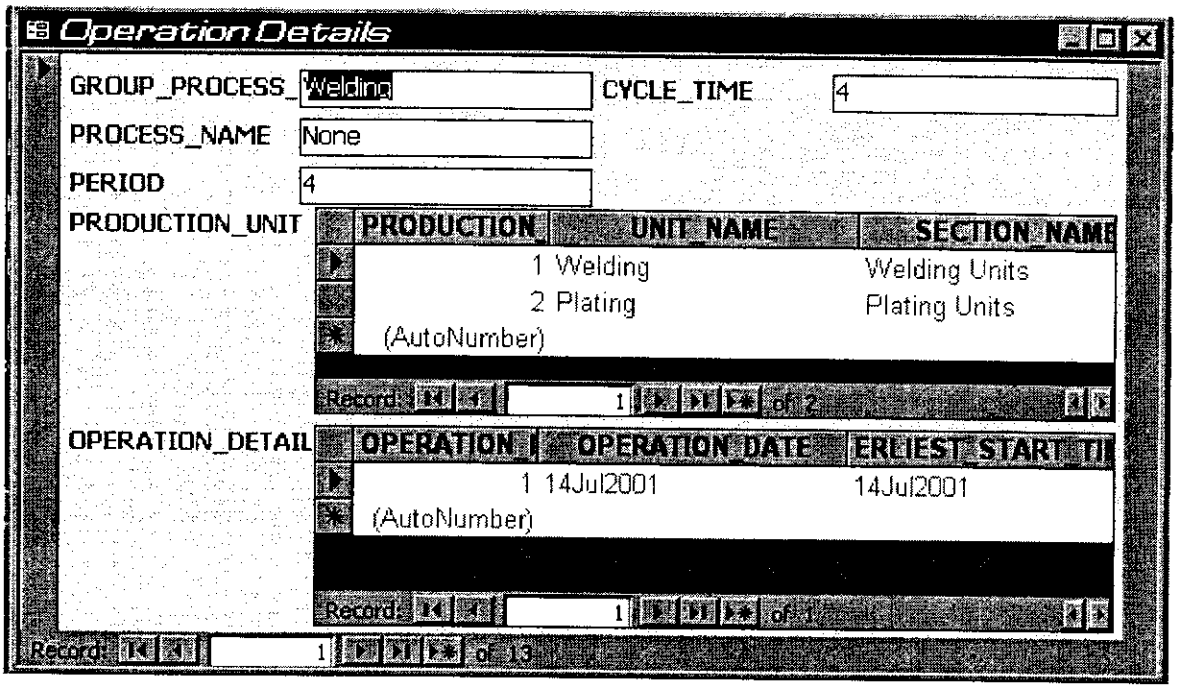
Rajah 3.4 Borang Pesanan

Untuk kemasukan rekod bagi borang pesanan antara maklumat yang perlu diisi adalah jenis produk, *serial number*, tarikh pesanan , tarikh penghantaran yang juga dikenali sebagai *due date* di dalam sistem kuantiti pesanan dan juga maklumat pelanggan.



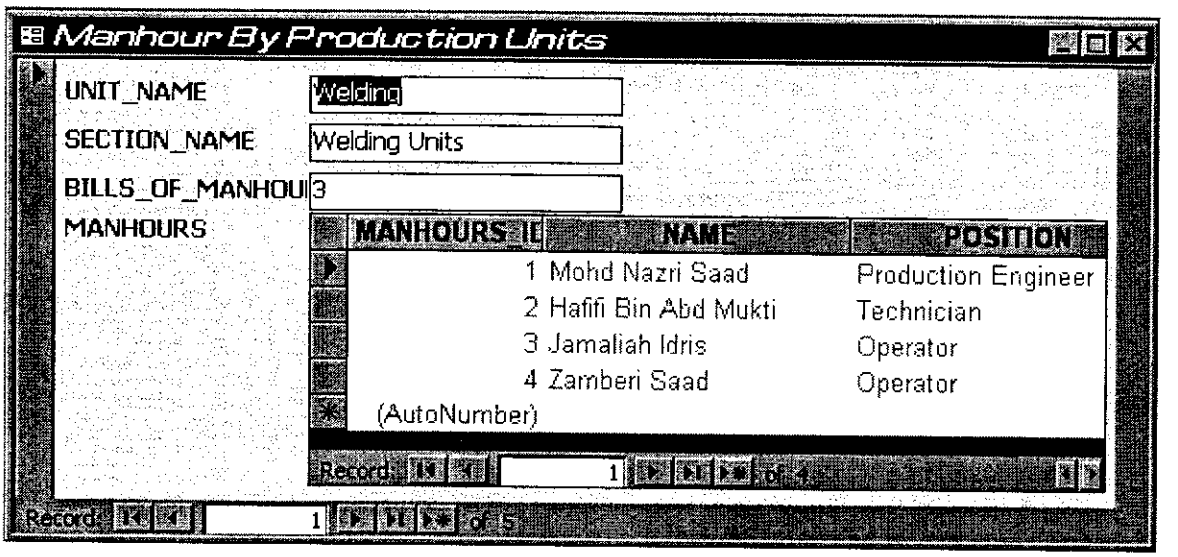
Rajah 3.5 Borang Bagi Produk

Untuk kemasukan rekod bagi proses pembuatan berdasarkan produk adalah *group process name*, *process name*, *period* dan *cycle time*. Ini adalah perlu kerana maklumat ini penting semasa proses pembuatan hendak dilakukan kerana ia juga menentukan operasi bagi setiap mesin untuk setiap masa atau hari.



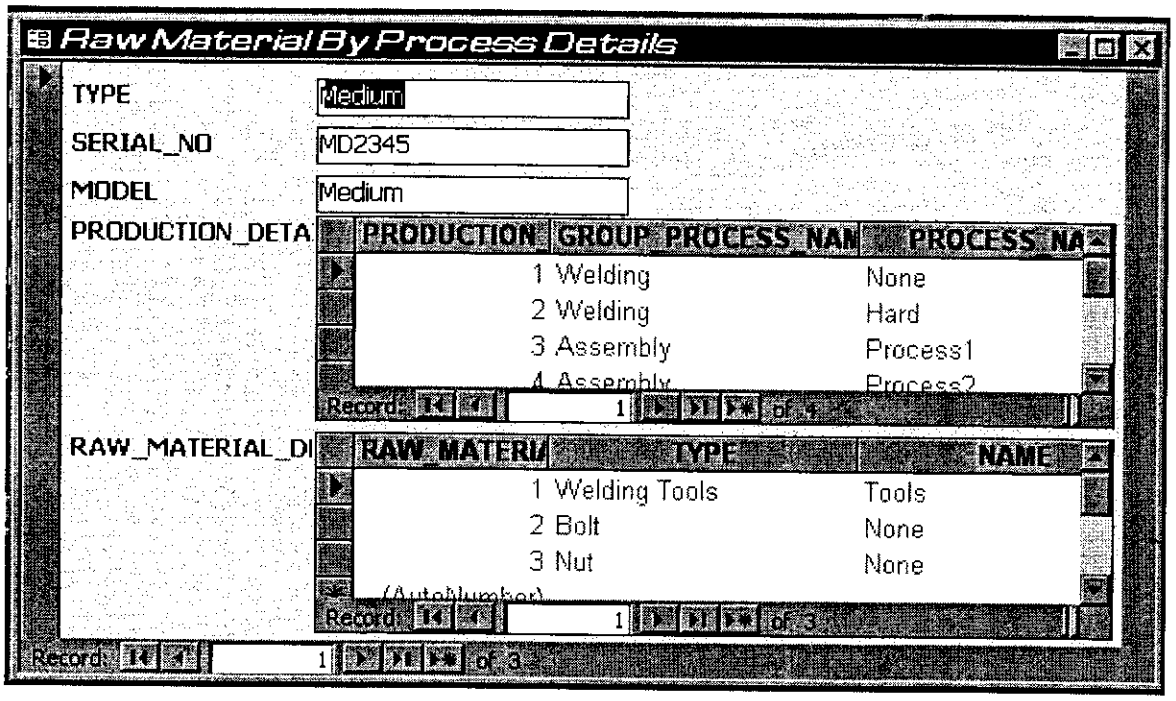
Rajah 3.6 Borang bagi maklumat operasi pengeluaran

Untuk kemasukan bagi maklumat operasi pembuatan berdasarkan unit pembuatan adalah maklumat bagi unit seperti nama unit, seksysen dan juga bilangan buruh untuk unit tersebut. Selepas itu pengguna sistem perlu memasukkan maklumat operasi bagi setiap unit seperti tarikh operasi, *erliest start times* (*masa terawal melakukan operasi*), *latest start times* (*masa terakhir melakukan operasi*) dan juga jangkamasa operasi dijalankan. Maklumat yang diisi ini bergantung kepada jenis-jenis proses yang terlibat dan ditentukan semasa hendak menghasilkan produk tersebut dan ianya telahpun berada di dalam sistem semasa pengisian maklumat produk.



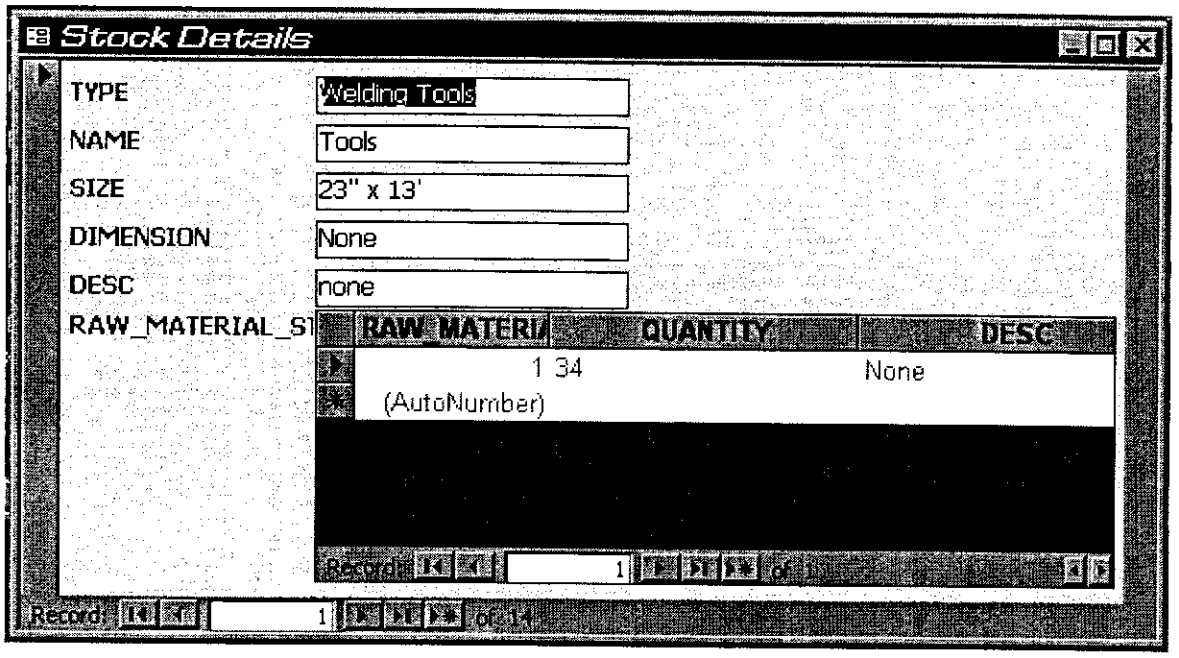
Rajah 3.7 Borang rekod pekerja/operator pengeluaran

Untuk kemasukan maklumat buruh ataupun kakitangan pengguna sistem perlu memasukkan maklumat seperti nama, jawatan, alamat dan sebagainya berdasarkan unit pengeluaran.



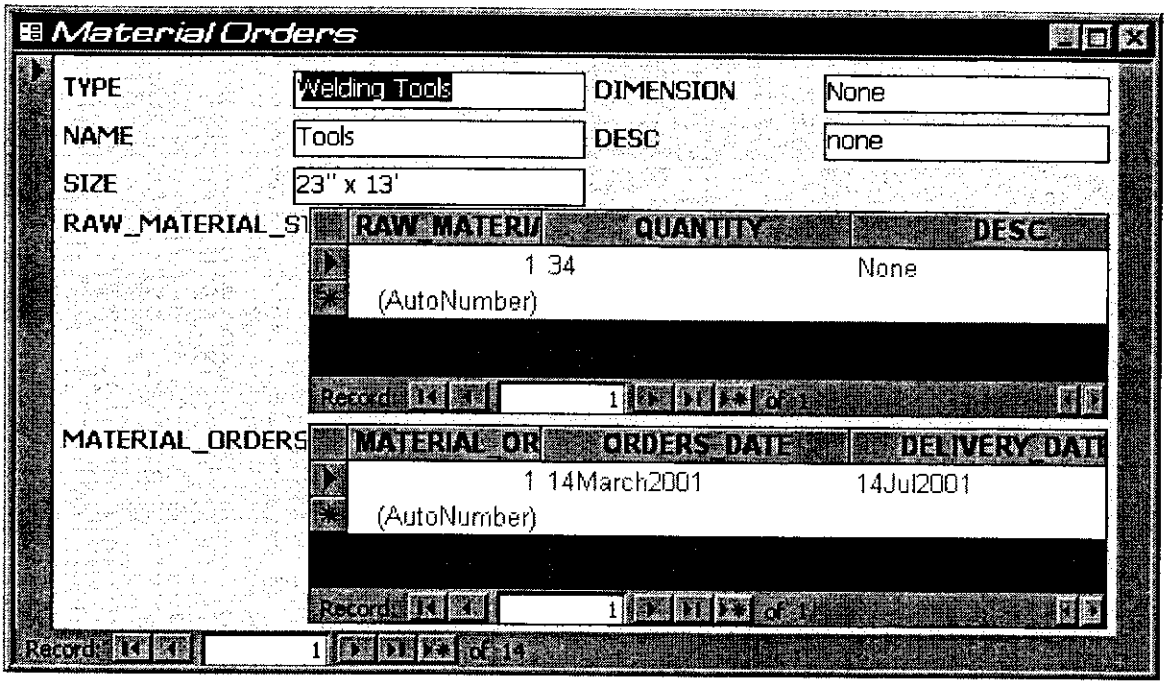
Rajah 3.8 Borang kemasukan rekod keperluan bahan mentah, operasi pengeluaran yang terlibat berdasarkan produk

Untuk kemasukan rekod bagi penggunaan bahan mentah ia mestilah berdasarkan operasi, proses dan juga produk yang hendak dihasilkan. Oleh itu maklumat tersebut adalah jenis bahan mentah, nama bahan mentah, kantiti yang diperlukan, saiz, dimensi dan sebagainya.



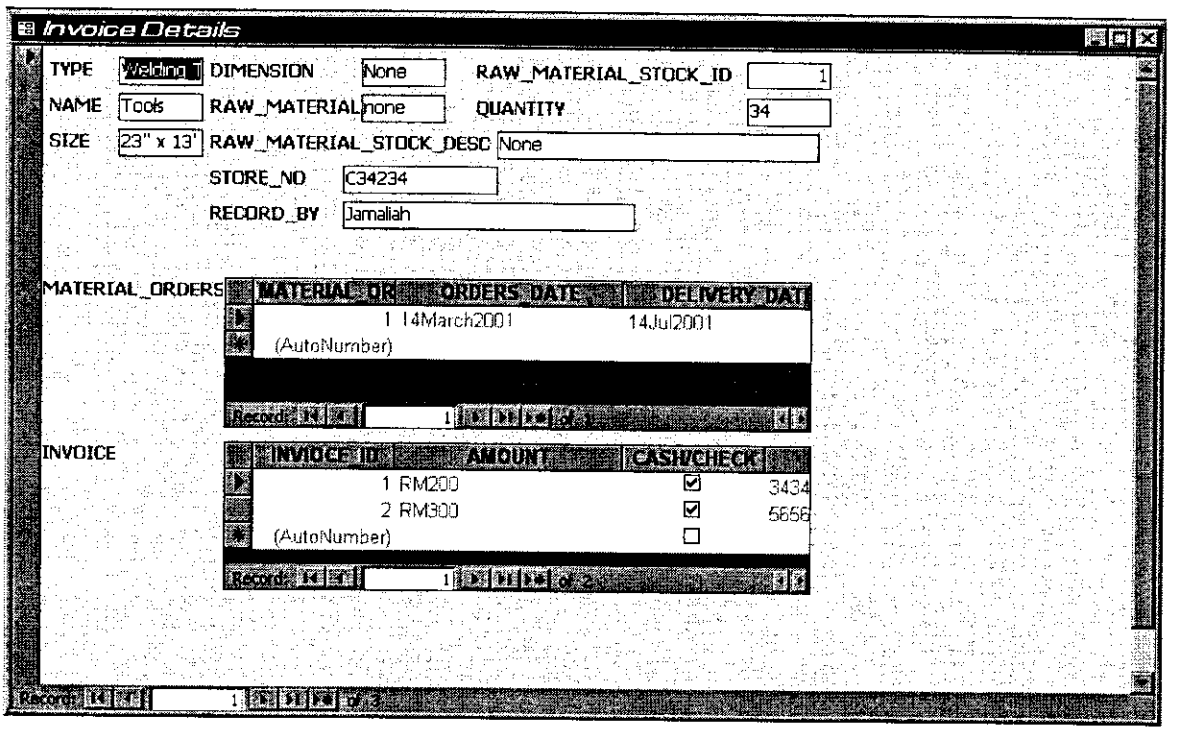
Rajah 3.9 Borang kemasukan rekod bagi stok bahan mentah

Untuk kemasukan rekod bagi stok bahan mentah pengguna sistem perlu memasukkan kuantiti bahan mentah, tempat disimpan dan semakan rekod oleh kakitangan. Ini bagi menentukan barangan/produk yang hendak dihasilkan berjalan dengan lancar.



Rajah 3.10 Borang pesanan bahan mentah

Untuk kemasukan rekod bagi pesanan bahan mentah daripada pembekal untuk tujuan pengeluaran, pengguna akhir sistem perlu memasukkan tarikh pesanan, tarikh penghantaran dan juga maklumat tentang pembekal. Ini adalah perlu bagi menentukan pengeluaran tidak terhenti semasa beroperasi dengan kekurangan bahan mentah. Di bawah pula ditunjukkan pengisian maklumat *invoice* bagi pesanan bahan mentah seperti jumlah dan juga kaedah bayaran.



Rajah 3.11 Borang kemasukan rekod invoice

BAB 4

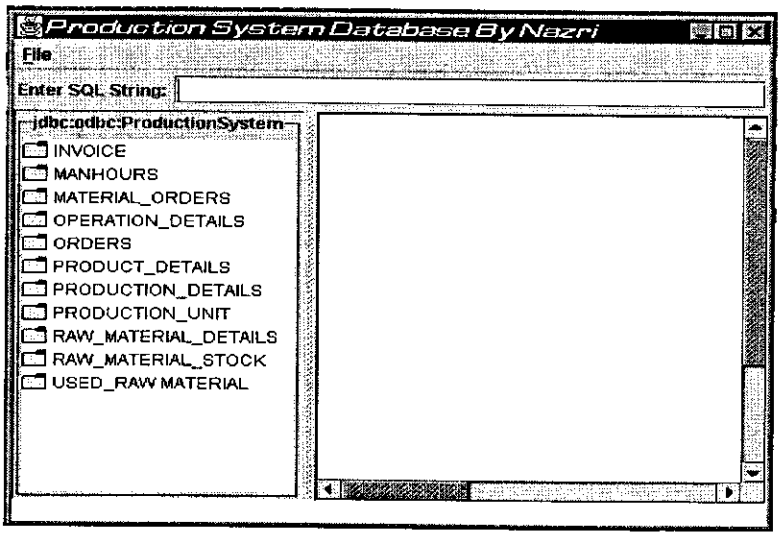
REKABENTUK PROTOTYPE APLIKASI JAVA

4.0 Pengenalan

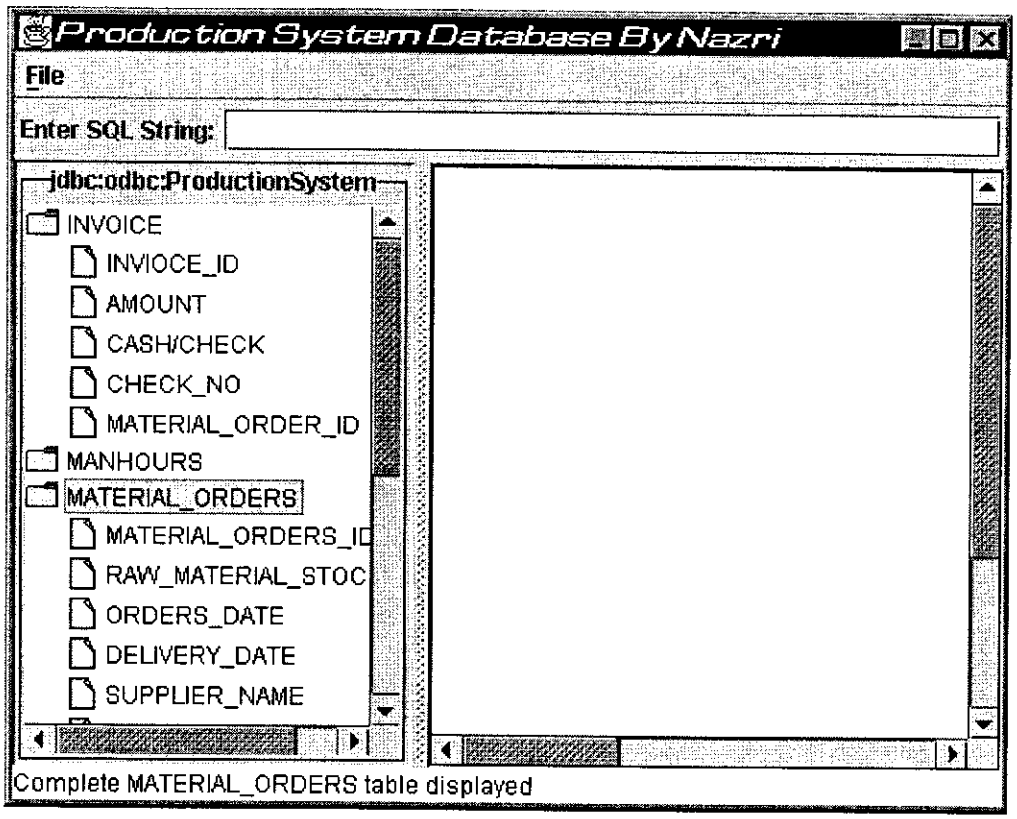
Secara umumnya aplikasi yang menggunakan bahasa Java sebagai sistem maklumat adalah kompleks dan memerlukan kepakaran yang tinggi terutamanya di dalam pengaturcaraan. Ini juga bertujuan bagi menambahkan lagi kesesuaian penggunaan seperti penganalisaan yang berbentuk graf (*graphic chart*) terutamanya bagi sistem pengeluaran. Kaedah ini pada masa dahulunya dilakukan dengan cara manual bagi menentukan kadar inventori yang sesuai untuk tujuan kawalan. Sistem ini juga bertujuan bagi mengawal operasi pengeluaran dengan cepat dan tepat bagi menjamin kos yang ditanggung oleh syarikat atau organisasi pengeluaran adalah rendah. Di dalam bab ini ditunjukkan serba ringkas tentang penggunaan aplikasi java yang telah siap dibangunkan melalui paparan antaramuka. Sistem yang sebenar atau prototype bagi aplikasi ini disertakan di belakang (*di bahagian appendik*).

4.1 Paparan Utama Sistem Pengeluaran

Di bawah ini ditunjukkan paparan utama bagi sistem pengeluaran yang telah siap dibangunkan. Di dalam paparan ini ditunjukkan label bagi *SQL* (*Structured Query Language*), jadual pengisian maklumat/rekod dan juga ruang kosong untuk paparan maklumat. Antara jadual (*table*) yang tertera di dalam paparan antaramuka sistem adalah *invoice*, *manhours*, *material orders*, *operation details*, *orders*, *product details*, *production details*, *production unit*, *raw material details*, *stocks* dan juga penggunaan bahan mentah semasa operasi pengeluaran dijalankan.



Rajah 4.1 Paparan Utama bagi aplikasi java



Rajah 4.2 Table Display

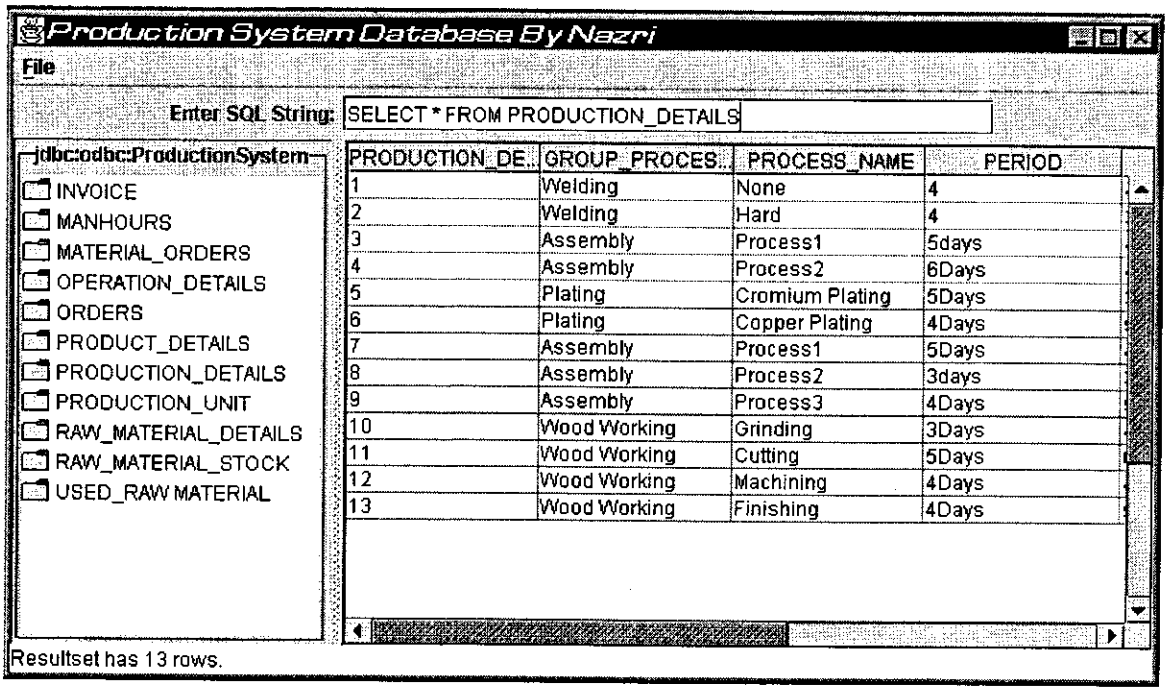
Di atas ditunjukkan paparan antaramuka bagi sistem yang disertakan dengan atribut-atribut yang terlibat untuk kemudahan pengguna sistem dan sebagai contohnya diambil rekod bagi *invoice* adalah seperti *invoice id*, *amount*, *cash/check* dan juga *material orders id.* Ini adalah perlu bagi kemudahan pengguna bagi mencari maklumat yang diperlukan semasa operasi pengeluaran dijalankan.

4.2 Contoh-contoh *Query* yang dihasilkan melalui *SQL*

Di dalam bahagian ini akan dibincangkan serba ringkas tentang penggunaan dan kelebihan sistem yang telah dibangunkan. Bagi query yang pertama ditunjukkan di dalam paparan antaramuka dan penyataan *SQL*nya adalah seperti berikut:

SELECT* FROM PRODUCTION_DETAILS

Dan hasilnya adalah seperti yang tertera di bawah (*paparan antaramuka*). Di dalam paparan tersebut antara maklumat yang didapati adalah *group process name*, *process name* dan juga *period* yang digunakan semasa operasi pengeluaran.



PRODUCTION_DE...	GROUP_PROCES...	PROCESS_NAME	PERIOD
1	Welding	None	4
2	Welding	Hard	4
3	Assembly	Process1	5days
4	Assembly	Process2	6Days
5	Plating	Cromium Plating	5Days
6	Plating	Copper Plating	4Days
7	Assembly	Process1	5Days
8	Assembly	Process2	3days
9	Assembly	Process3	4Days
10	Wood Working	Grinding	3Days
11	Wood Working	Cutting	5Days
12	Wood Working	Machining	4Days
13	Wood Working	Finishing	4Days

Rajah 4.3 Paparan untuk Query 1

Untuk *query* yang kedua pula adalah seperti berikut:

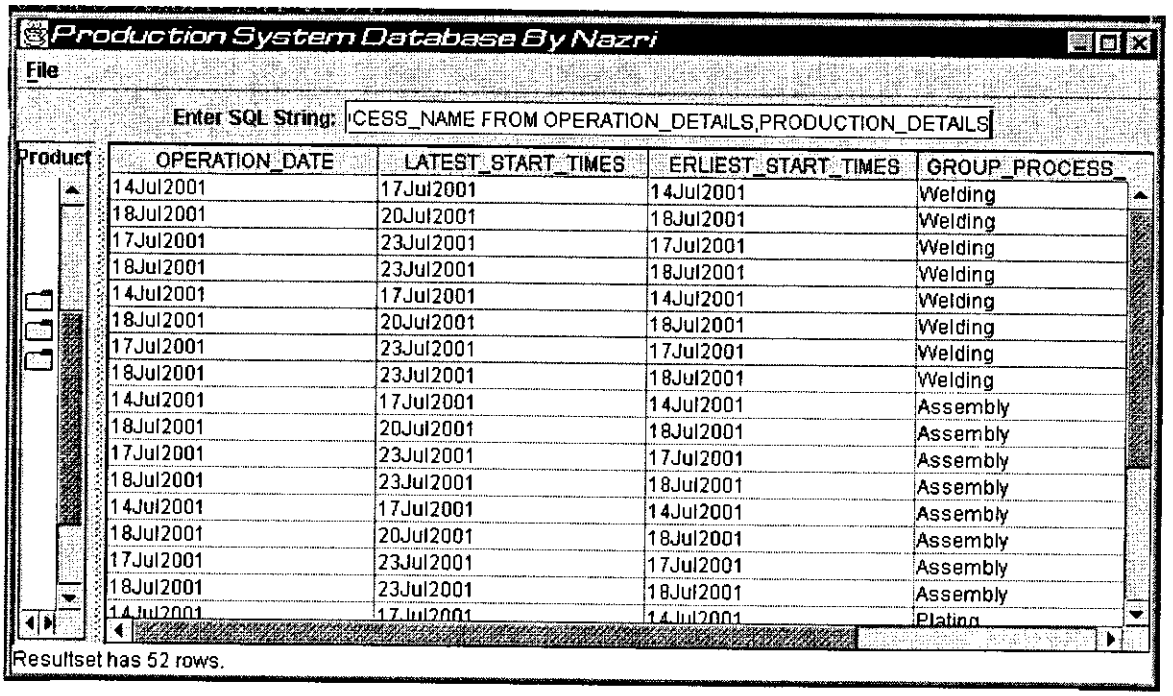
SELECT OPERATION_DATE,

LATEST_START_TIMES,ERLIEST_START_TIMES,GROUP_PROCESS_NA

ME,PROCESS_NAME FROM

OPERATION_DETAILS,PRODUCTION_DETAILS

Dan paparannya ditunjukkan di dalam rajah di bawah yang mengandungi maklumat seperti *operation date, latest start times, earliest start times, group process name* yang didapati di dalam jadual *operation details* dan juga *production details.*



OPERATION_DATE	LATEST_START_TIMES	ERLIEST_START_TIMES	GROUP_PROCESS_
14Jul2001	17Jul2001	14Jul2001	Welding
18Jul2001	20Jul2001	18Jul2001	Welding
17Jul2001	23Jul2001	17Jul2001	Welding
18Jul2001	23Jul2001	18Jul2001	Welding
14Jul2001	17Jul2001	14Jul2001	Welding
18Jul2001	20Jul2001	18Jul2001	Welding
17Jul2001	23Jul2001	17Jul2001	Welding
18Jul2001	23Jul2001	18Jul2001	Welding
14Jul2001	17Jul2001	14Jul2001	Assembly
18Jul2001	20Jul2001	18Jul2001	Assembly
17Jul2001	23Jul2001	17Jul2001	Assembly
18Jul2001	23Jul2001	18Jul2001	Assembly
14Jul2001	17Jul2001	14Jul2001	Assembly
18Jul2001	20Jul2001	18Jul2001	Assembly
17Jul2001	23Jul2001	17Jul2001	Assembly
18Jul2001	23Jul2001	18Jul2001	Assembly
14Jul2001	17Jul2001	14Jul2001	Plating

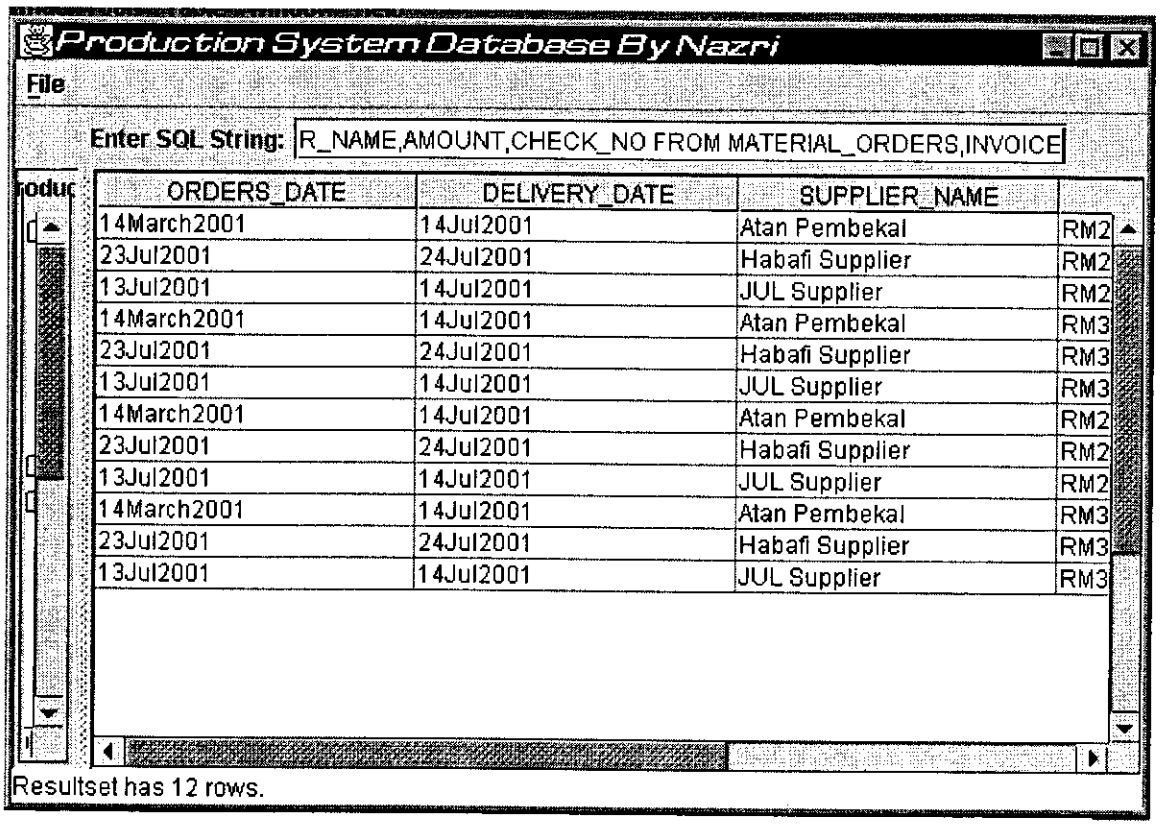
Rajah 4.4 Paparan bagi query 2

Di bawah ini pula adalah *query* yang ketiga dan penyataannya adalah seperti berikut:

SELECT

ORDERS_DATE,DELIVERY_DATE,SUPPLIER_NAME,AMOUNT,CHECK_NO FROM MATERIAL_ORDERS,INVOICE

Dan paparannya menunjukkan *orders date*, *delivery date* dan juga maklumat mengenai pembekal untuk tujuan capaian rekod dengan lebih tepat dan jitu.



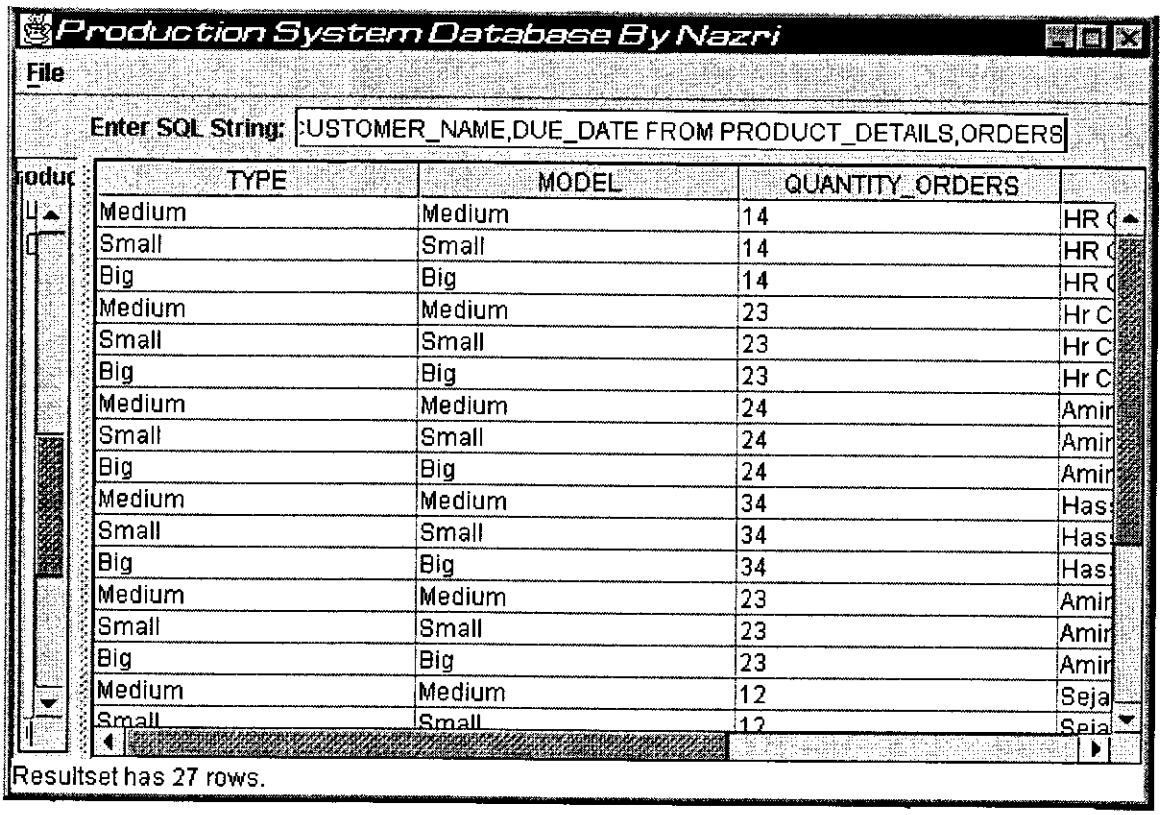
ORDERS_DATE	DELIVERY_DATE	SUPPLIER_NAME	
14March2001	14Jul2001	Atan Pembekal	RM2
23Jul2001	24Jul2001	Habafi Supplier	RM2
13Jul2001	14Jul2001	JUL Supplier	RM2
14March2001	14Jul2001	Atan Pembekal	RM3
23Jul2001	24Jul2001	Habafi Supplier	RM3
13Jul2001	14Jul2001	JUL Supplier	RM3
14March2001	14Jul2001	Atan Pembekal	RM2
23Jul2001	24Jul2001	Habafi Supplier	RM2
13Jul2001	14Jul2001	JUL Supplier	RM2
14March2001	14Jul2001	Atan Pembekal	RM3
23Jul2001	24Jul2001	Habafi Supplier	RM3
13Jul2001	14Jul2001	JUL Supplier	RM3

Rajah 4.5 paparan bagi query 3

Bagi query yang keempat pula ditunjukkan tentang maklumat pesanan dengan lebih jelas untuk menganalisa prestasi syarikat berdasarkan pesanan daripada pengguna. Pernyataan bagi query tersebut adalah:

```
SELECT TYPE,MODEL,QUANTITY_ORDERS,CUSTOMER_NAME,DUE_DATE
FROM PRODUCT_DETAILS,ORDERS
```

Dan paparannya tertera dibawah:

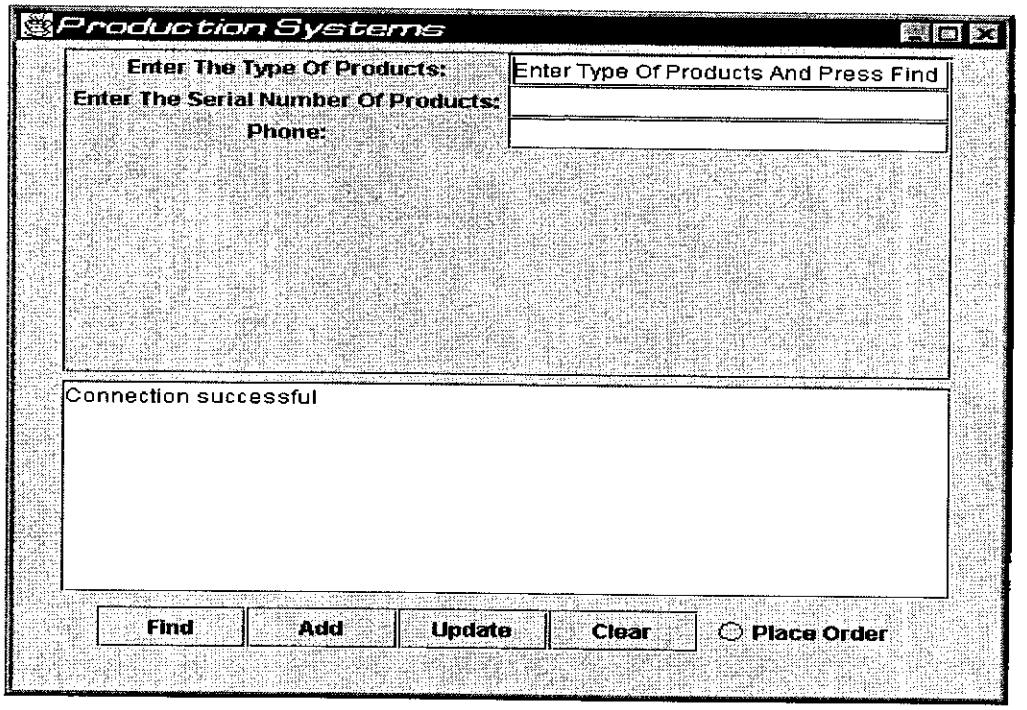


TYPE	MODEL	QUANTITY_ORDERS	
Medium	Medium	14	HR (
Small	Small	14	HR (
Big	Big	14	HR (
Medium	Medium	23	Hr C
Small	Small	23	Hr C
Big	Big	23	Hr C
Medium	Medium	24	Amir
Small	Small	24	Amir
Big	Big	24	Amir
Medium	Medium	34	Has
Small	Small	34	Has
Big	Big	34	Has
Medium	Medium	23	Amir
Small	Small	23	Amir
Big	Big	23	Amir
Medium	Medium	12	Seja
Small	Small	12	Seja

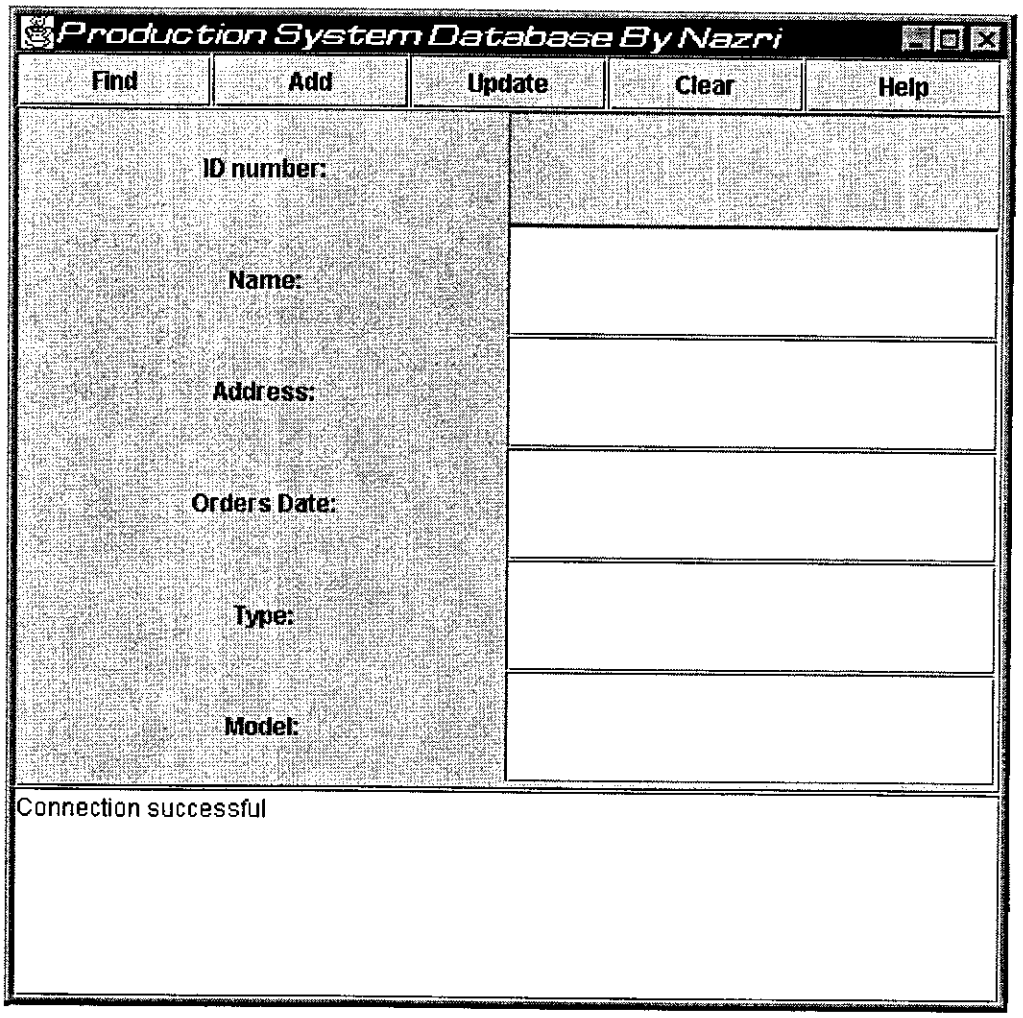
Rajah 4.6 Paparan bagi query 4

4.3 Borang Pengisisan Rekod Melalui Aplikasi Java

Di bawah ini ditunjukkan paparan pengisian data melalui aplikasi java yang telah dibangunkan. Antara maklumat yang boleh dimasukkan adalah jenis produk, *serial number* dan juga maklumat mengenai pesanan daripada pelanggan.



Rajah 4.7 Paparan borang pengisian rekod bagi produk yang hendak dihasilkan



Rajah 4.8 Paparan bagi borang pesanan

4.2 Maklumat Yang Boleh Didapati Daripada Sistem Secara Terus

Antara maklumat yang boleh didapati daripada sistem secara terus adalah seperti yang ditunjukkan di bawah. Maklumat tersebut adalah maklumat yang akan digunakan sebelum operasi pengeluaran dijalankan. Ini perlu bagi mengelakkan pengeluaran yang berlebihan kerana ia perlu dilakukan melalui maklumat pesanan serta maklumat tarikh operasi pengeluaran bermula kerana kelewatan pembuatan produk merugikan syarikat ataupun organisasi pengeluaran.

Production Schedule Query Viewer By Nazri

OPERATI...	ERLIEST...	LATEST_...	DURATI...	PRODU...	TYPE	SERIAL_...	MODEL	ORDERS...	ORDER...	ORDERS...	DUE_DA...
14Jul2001	14Jul2001	17Jul2001	23	1	Medium	MD2345	Medium	1	13Jul20...	1	23Oct20...
14Jul2001	14Jul2001	17Jul2001	23	1	Medium	MD2345	Medium	2	14Jul20...	1	28Oct20...
14Jul2001	14Jul2001	17Jul2001	23	1	Medium	MD2345	Medium	3	16Jul20...	1	25Oct20...
18Jul2001	18Jul2001	20Jul2001	24	1	Medium	MD2345	Medium	1	13Jul20...	1	23Oct20...
18Jul2001	18Jul2001	20Jul2001	24	1	Medium	MD2345	Medium	2	14Jul20...	1	28Oct20...
18Jul2001	18Jul2001	20Jul2001	24	1	Medium	MD2345	Medium	3	16Jul20...	1	25Oct20...
17Jul2001	17Jul2001	23Jul2001	34	1	Medium	MD2345	Medium	1	13Jul20...	1	23Oct20...
17Jul2001	17Jul2001	23Jul2001	34	1	Medium	MD2345	Medium	2	14Jul20...	1	28Oct20...
17Jul2001	17Jul2001	23Jul2001	34	1	Medium	MD2345	Medium	3	16Jul20...	1	25Oct20...
18Jul2001	18Jul2001	23Jul2001	34	1	Medium	MD2345	Medium	1	13Jul20...	1	23Oct20...
18Jul2001	18Jul2001	23Jul2001	34	1	Medium	MD2345	Medium	2	14Jul20...	1	28Oct20...
18Jul2001	18Jul2001	23Jul2001	34	1	Medium	MD2345	Medium	3	16Jul20...	1	25Oct20...

Production Schedule Query Viewer By Nazri

GROU...	PROC...	PERIOD	CYCLE...	PROD...	UNIT_...	SECTI...	BILLS_...	ORDE...	PROD...	DUE_...	QUANT...	TYPE	SERIA...	MODEL
Welding	None	4	4	1	Welding	Weldi...	3	13Jul2...	1	23Oct...	14	Medium	MD2345	Medium
Welding	None	4	4	1	Welding	Weldi...	3	14Jul2...	1	28Oct...	23	Medium	MD2345	Medium
Welding	None	4	4	1	Welding	Weldi...	3	16Jul2...	1	25Oct...	24	Medium	MD2345	Medium
Welding	None	4	4	2	Plating	Platin...	2	13Jul2...	1	23Oct...	14	Medium	MD2345	Medium
Welding	None	4	4	2	Plating	Platin...	2	14Jul2...	1	28Oct...	23	Medium	MD2345	Medium
Welding	None	4	4	2	Plating	Platin...	2	16Jul2...	1	25Oct...	24	Medium	MD2345	Medium
Welding	Hard	4	3	3	Welding	Weldi...	4	13Jul2...	1	23Oct...	14	Medium	MD2345	Medium
Welding	Hard	4	3	3	Welding	Weldi...	4	14Jul2...	1	28Oct...	23	Medium	MD2345	Medium
Welding	Hard	4	3	3	Welding	Weldi...	4	16Jul2...	1	25Oct...	24	Medium	MD2345	Medium
Asse...	Proce...	5days	4	4	Asse...	Proce...	5	13Jul2...	1	23Oct...	14	Medium	MD2345	Medium
Asse...	Proce...	5days	4	4	Asse...	Proce...	5	14Jul2...	1	28Oct...	23	Medium	MD2345	Medium
Asse...	Proce...	5days	4	4	Asse...	Proce...	5	16Jul2...	1	25Oct...	24	Medium	MD2345	Medium
Asse...	Proce...	6Days	4	5	Asse...	Proce...	4	13Jul2...	1	23Oct...	14	Medium	MD2345	Medium
Asse...	Proce...	6Days	4	5	Asse...	Proce...	4	14Jul2...	1	28Oct...	23	Medium	MD2345	Medium
Asse...	Proce...	6Days	4	5	Asse...	Proce...	4	16Jul2...	1	25Oct...	24	Medium	MD2345	Medium

Rajah 4.9 Contoh paparan perancangan yang dihasilkan melalui sistem yang akan digunakan untuk proses pembuatan seperti tarikh awal operasi pengeluaran, tarikh lewat dan juga kuantiti pesanan

BAB 5

APPLIKASI APPLET

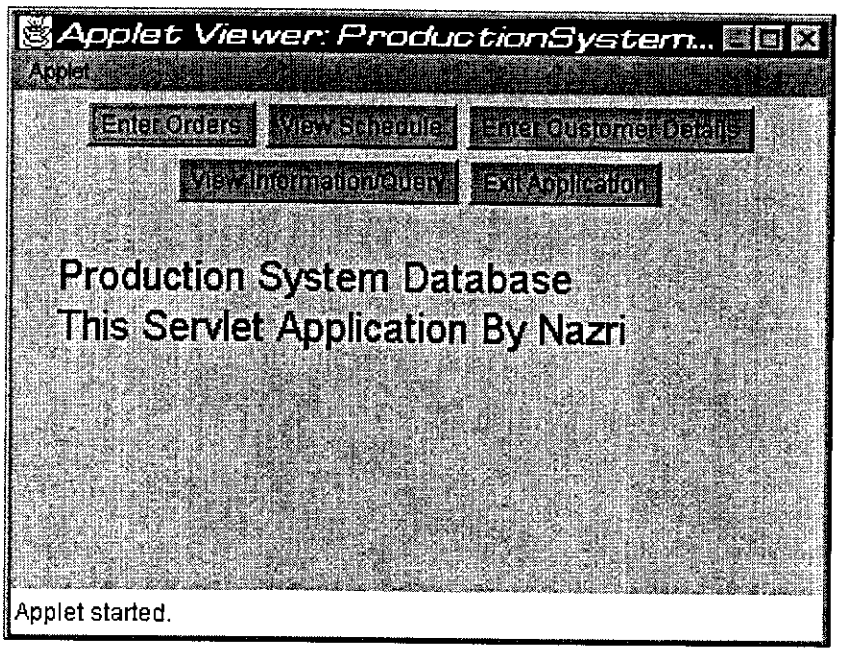
5.0 Pendahuluan

Di dalam bahasa java aplikasi Internet ataupun melalui Internet Explorer pada senibina rangkaian setempat dipanggil Applet. Di dalam sistem pengeluaran yang telah dibangunkan Applet digunakan untuk menganalisa rekod-rekod daripada pangkalan data asas (*datasource*) dalam bentuk grafik ataupun untuk pesanan terus kepada sistem (*rujuk Rajah 5.1*).

Di dalam projek ini applet ini juga digunakan untuk menganalisa rekod-rekod yang berada di dalam pangkalan data asas (*datasource*) untuk kemudahan unit kawalan inventory dan juga kawalan bahan mentah. Oleh itu paparan secara grafik diperlukan kerana operasi pengeluaran dan rekod-rekod yang terlibat adalah banyak dan ianya akan menimbulkan masalah jika dilakukan secara manual.

5.1 Paparan Utama Aplikasi Applet

Untuk sistem ini diperlukan paparan antaramuka utama bagi sistem kerana ia adalah perlu bagi kemudahan penggunaannya. Antara menu yang terdapat pada paparan utama adalah masukkan rekod pesanan, maklumat perancangan pengeluaran dan juga masukkan aplikasi penganalisaan rekod secara grafik seperti yang ditunjukkan di dalam *rajah 5.1*.

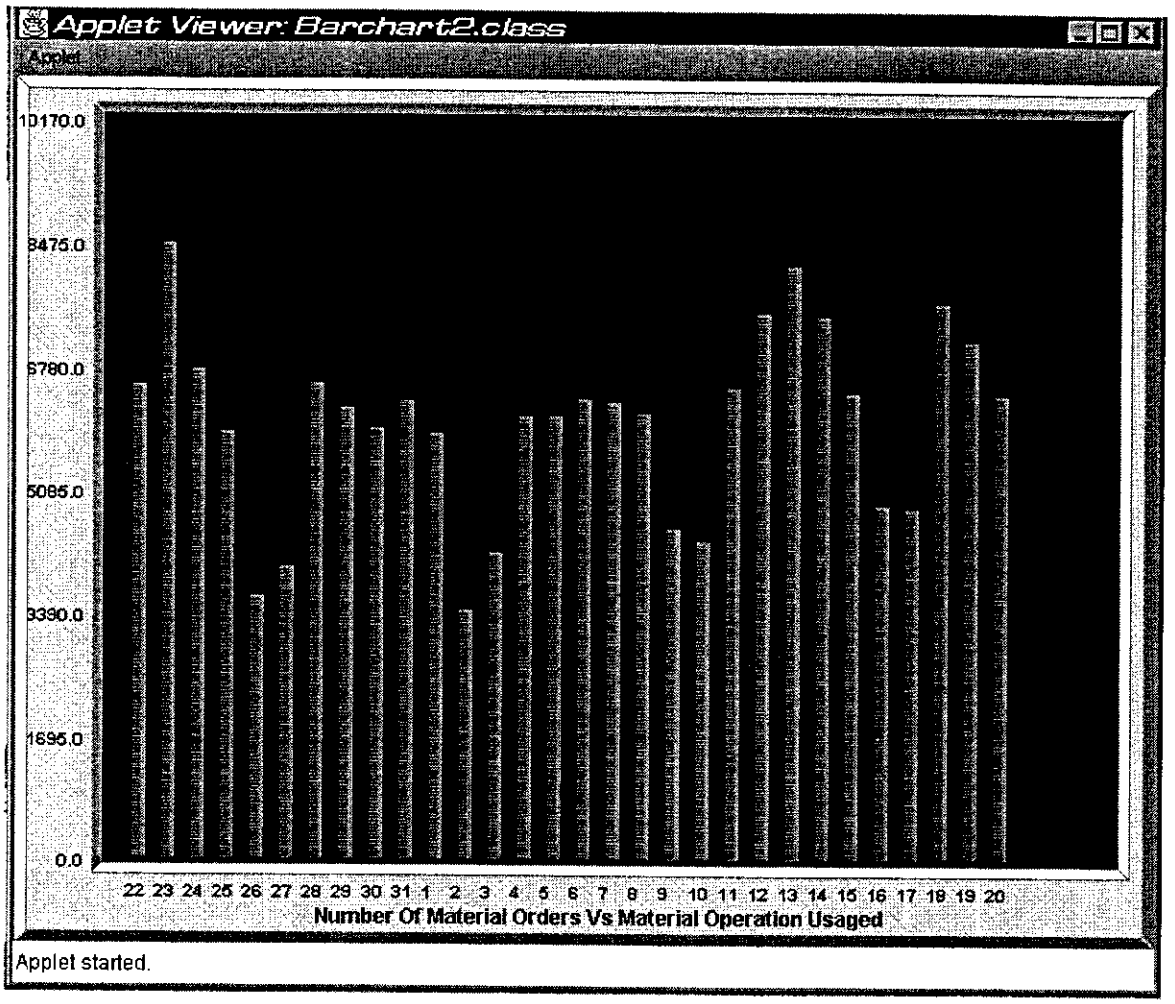


Rajah 5.1 Menu Utama bagi sistem untuk pesanan produk terus kepada sistem

5.2 Penganalisaan Secara Grafik

Bagi projek ini paparan penganalisaan secara grafik diperlukan untuk kemudahan membuat pesanan semula, mengurangkan pesanan serta mengawal operasi pengeluaran seperti memberhentikan pengeluaran jika produk yang dihasilkan adalah berlebihan. Ini adalah perlu kerana di dalam organisasi pengeluaran ia

adalah melibatkan kos yang agak banyak dan ianya perlu dikawal supaya syarikat tidak mengalami kerugian.



Rajah 5.2 Penganalisaan Penggunaan Bahan Mentah Melalui Sistem

Rajah 5.2 menunjukkan paparan grafik bagi penggunaan bahan mentah berdasarkan operasi pengeluaran. Ini adalah perlu bagi menentukan keperluaan bahan mentah adalah cukup untuk operasi pengeluaran bagi tujuan kawalan inventori. Oleh itu pengguna sistem mengetahui kekurangan bahan mentah dan terus membuat pesanan kepada pembekal dan jika bahan mentah adalah

berlebihan (seperti yang dipaparkan di dalam rajah) dan melebihi had tertentu maka pesanan bahan mentah dikurangkan. Ini adalah perlu bagi mengawal modal pusingan syarikat bagi mengoptimakan operasi pengeluaran.

BAB 6

KESIMPULAN

Secara keseluruhannya projek ini dianggap berjaya kerana ianya dapat menunjukkan aplikasi sebenar bahasa java dan kesesuaiannya dalam penggunaanya di dalam bidang pengeluaran. Ini adalah kerana keperluan sistem kajian adalah kompleks dan rumit berbanding sistem maklumat yang lain dan memerlukan pengaturcaraan sebagai aplikasi tambahan namun dari segi teknikal seperti rekabentuk sistem menggunakan java adalah terhad dengan kekurangan kepakaran didalam bidang pengaturcaraan bahasa java.

Sistem ini juga dapat berfungsi sepenuhnya dan ianya boleh digunakan pada keadaan sebenar tanpa sebarang masalah namun ianya tidak mampu menganalisa rekod-rekod kerana ianya melibatkan pengaturcaraan yang agak rumit namun ianya dapat juga menghasilkan aplikasi Applet yang dipamirkan melalui *Internet Explorer* yang dapat menganalisa rekod-rekod bahan mentah secara automatik daripada sistem bagi tujuan kawalan inventori.

Dari segi penggunaan bahasa java sebagai aplikasi untuk sistem maklumat ianya dapat berfungsi sepenuhnya dalam apa jua keadaan dengan bertambahnya kelebihan di dalam bahasa tersebut seperti pengaturcaraan Servlet, Remote Method Invocation dan juga java JDNI dengan wujudnya *compiler* J2SDK dan juga J2SDKEE. Aplikasi-aplikasi ini dapat digunakan pada sebarang senibina rangkaian dan ianya adalah canggih berbanding bahasa lain seperti Visual Basic, C/C++ dan juga bahasa lain seperti Perl.

Dengan berjalannya projek ini ianya serba sedikit dapat menunjukkan kebolehan bahasa java yang digunakan pada sistem pengeluaran dan ianya mampu menampung pelbagai aplikasi yang dikehendaki oleh pengguna akhir sistem.

RUJUKAN

Lu S.C.-Y., Zhang G.: A combined inductive learning and experimental design approach to manufacturing operation planning. Journal of Manufacturing Systems, 9(2) 1990.

Houtzeel A.: The MICLASS System. Proceedings of CAM-I's Executive Seminar-Coding, Classification and Group Technology for Automated Planning, CAM-I, Arlington, Texas. 1976.

Feld M.: Planning and automation of technological processes of machine elements. WN-T Warszawa, 1994 (in polish).

Collective work: Uniform design-technological classifier of production parts. Standard Publications, Warszawa, 1977 (in polish).

Opitz H.: A Classification System to Describe Work pieces, Part I, Pergamon Press, UK, 1970.

CAM Software Inc. DCLASS Technical and System Manual, CAM Software Inc., Provo, Utah, USA, 1985.

DCLASS Technical Manual, CAM Software Research Laboratory, Brigham Young University, Provo, Utah, USA, 1985.

Devireddy, C.R., and Ghosh, K., *Feature-based modelling and neural networks-based CAPP for integrated manufacturing*, Int. J. Computer Integrated Manufacturing, ISSN 0951-192X, 1999, Vol. 12, No. 1, pp.61-74.

Dewan, P., and Joshi, S., *Distributed scheduling of job shops using auctions*, Proceedings of the second int. workshop on Intelligent Manufacturing Systems (IMS'99), ISBN 90-73802- 69-5, Leuven, Belgium, pp. 33-40.

Dilts, D.M., Boyd, N.P., and Whorms, H.H., *The evolution of control architectures for automated manufacturing systems*, Journal of Manufacturing Systems, ISSN 0278-6125, Vol. 10, No. 1, pp.79-93.

Dini, G., *Literature database on applications of artificial intelligence methods in manufacturing engineering*, CIRP annals, ISSN 0007-8506, 1997, Vol. 46 (Issue 2), pp. 681-690.

Duffie, N.A., Chitturi, R., and Mou, J., *Fault tolerant heterarchical control of heterogeneous manufacturing system entities*, Journal of Manufacturing Systems, ISSN 0278-6125, 1988, Vol. 7, pp. 315-328.

Duffie, N.A., and Prabhu, V.V., *Heterarchical control of highly distributed manufacturing systems*, Int. J. Computer Integrated Manufacturing, ISSN 0951-192X, 1996, Vol. 9, No. 4, pp. 270- 281.

ElMaraghy, H.A., and ElMaraghy, W.H., *Bridging the gap between process planning and production planning and control*, Manufacturing Systems, 1993, Vol.22, No.1, pp. 5-11.

Erkoc, M., Plebani, L.J., and Sathyanarayanan, G., *A model for integrating process planning and scheduling in a job shop environment*, 31st CIRP International Seminar on Manufacturing Systems, Berkeley, May 1998, pp.201-206.

Dittel & Dittel, Java Swing Programming, Pentice Hall, New York, 2000

Zhang H. C., Alting L.: Computerized Manufacturing Process Planning Systems. Chapman & Hall, 1994.

APPENDIK
PAKEJ JAVA

Interface java.sql.Driver

public interface **Driver**

The Java SQL framework allows for multiple database drivers.

Each driver should supply a class that implements the Driver interface.

The DriverManager will try to load as many drivers as it can find and then for any given connection request, it will ask each driver in turn to try to connect to the target URL.

It is strongly recommended that each Driver class should be small and standalone so that the Driver class can be loaded and queried without bringing in vast quantities of supporting code.

When a Driver class is loaded, it should create an instance of itself and register it with the DriverManager. This means that a user can load and register a driver by doing Class.forName("foo.bah.Driver").

See Also:
DriverManager, Connection

Method Index

- **acceptsURL**(String)
  Returns true if the driver thinks that it can open a connection to the given URL.
- **connect**(String, Properties)
  Try to make a database connection to the given URL.
- **getMajorVersion**()
  Get the driver's major version number.
- **getMinorVersion**()
  Get the driver's minor version number.
- **getPropertyInfo**(String, Properties)

  The getPropertyInfo method is intended to allow a generic GUI tool to discover what properties it should prompt a human for in order to get enough information to connect to a database.

- **jdbcCompliant**()
  Report whether the Driver is a genuine JDBC COMPLIANT (tm) driver.

Methods

connect

```
public abstract Connection connect(String url,
                                   Properties info) throws SQLException
```

Try to make a database connection to the given URL. The driver should return "null" if it realizes it is the wrong kind of driver to connect to the given URL. This will be common, as when the JDBC driver manager is asked to connect to a given URL it passes the URL to each loaded driver in turn.

The driver should raise a SQLException if it is the right driver to connect to the given URL, but has trouble connecting to the database.

The java.util.Properties argument can be used to passed arbitrary string tag/value pairs as connection arguments. Normally at least "user" and "password" properties should be included in the Properties.

Parameters:
url - The URL of the database to connect to
info - a list of arbitrary string tag/value pairs as connection arguments; normally at least a "user" and "password" property should be included
Returns:
a Connection to the URL
Throws: SQLException
if a database-access error occurs.

- **acceptsURL**

```
public abstract boolean acceptsURL(String url) throws SQLException
```

Returns true if the driver thinks that it can open a connection to the given URL. Typically drivers will return true if they understand the subprotocol specified in the URL and false if they don't.

Parameters:
url - The URL of the database.
Returns:
True if this driver can connect to the given URL.
Throws: SQLException
if a database-access error occurs.

- **getPropertyInfo**

```
public abstract DriverPropertyInfo[] getPropertyInfo(String url,
                                                     Properties info) throws SQL
```

The getPropertyInfo method is intended to allow a generic GUI tool to discover what properties it should prompt a human for in order to get enough information to connect to a database. Note that depending on the values the human has supplied so far, additional values may become necessary, so it may be necessary to iterate though several calls to getPropertyInfo.

Parameters:
url - The URL of the database to connect to.
info - A proposed list of tag/value pairs that will be sent on connect open.
Returns:
An array of DriverPropertyInfo objects describing possible properties. This array may be an empty array if no properties are required.

Throws: SQLException
if a database-access error occurs.

- **getMajorVersion**

```
public abstract int getMajorVersion()
```

Get the driver's major version number. Initially this should be 1.

- **getMinorVersion**

```
public abstract int getMinorVersion()
```

Get the driver's minor version number. Initially this should be 0.

- **jdbcCompliant**

```
public abstract boolean jdbcCompliant()
```

Report whether the Driver is a genuine JDBC COMPLIANT (tm) driver. A driver may only report "true" here if it passes the JDBC compliance tests, otherwise it is required to return false. JDBC compliance requires full support for the JDBC API and full support for SQL 92 Entry Level. It is expected that JDBC compliant drivers will be available for all the major commercial databases. This method is not intended to encourage the development of non-JDBC compliant drivers, but is a recognition of the fact that some vendors are interested in using the JDBC API and framework for lightweight databases that do not support full database functionality, or for special databases such as document information retrieval where a SQL implementation may not be feasible.

Submit a bug or feature - Version 1.1.8 of Java Platform API Specification
Java is a trademark or registered trademark of Sun Microsystems, Inc. in the US and other countries.
Copyright 1995-1999 Sun Microsystems, Inc. 901 San Antonio Road,
Palo Alto, California, 94303, U.S.A. All Rights Reserved.

Java™ Platform 1.1 API Specification

Package Index

- package java.applet
- package java.awt
- package java.awt.datatransfer
- package java.awt.event
- package java.awt.image
- package java.beans
- package java.io
- package java.lang
- package java.lang.reflect
- package java.math
- package java.net
- package java.rmi
- package java.rmi.dgc
- package java.rmi.registry
- package java.rmi.server
- package java.security
- package java.security.acl
- package java.security.interfaces
- package java.sql
- package java.text
- package java.util
- package java.util.zip

For a searchable version of these specifications, see the Documentation Search Engine.

Get code examples for practically every class, method and field from:
The Java Class Libraries: An Annotated Reference
by Patrick Chan and Rosanna Lee

Submit a bug or feature - Version 1.1.8 of Java Platform API Specification
Java is a trademark or registered trademark of Sun Microsystems, Inc. in the US and other countries.
Copyright 1995-1999 Sun Microsystems, Inc. 901 San Antonio Road,
Palo Alto, California, 94303, U.S.A. All Rights Reserved.

Interface java.sql.Driver

public interface **Driver**

The Java SQL framework allows for multiple database drivers.

Each driver should supply a class that implements the Driver interface.

The DriverManager will try to load as many drivers as it can find and then for any given connection request, it will ask each driver in turn to try to connect to the target URL.

It is strongly recommended that each Driver class should be small and standalone so that the Driver class can be loaded and queried without bringing in vast quantities of supporting code.

When a Driver class is loaded, it should create an instance of itself and register it with the DriverManager. This means that a user can load and register a driver by doing Class.forName("foo.bah.Driver").

See Also:

DriverManager, Connection

Method Index

- **acceptsURL**(String)
  Returns true if the driver thinks that it can open a connection to the given URL.
- **connect**(String, Properties)
  Try to make a database connection to the given URL.
- **getMajorVersion**()
  Get the driver's major version number.
- **getMinorVersion**()
  Get the driver's minor version number.
- **getPropertyInfo**(String, Properties)

  The getPropertyInfo method is intended to allow a generic GUI tool to discover what properties it should prompt a human for in order to get enough information to connect to a database.

- **jdbcCompliant**()
  Report whether the Driver is a genuine JDBC COMPLIANT (tm) driver.

Methods

- **connect**

```
public abstract Connection connect(String url,
                                   Properties info) throws SQLException
```

Try to make a database connection to the given URL. The driver should return "null" if it realizes it is the wrong kind of driver to connect to the given URL. This will be common, as when the JDBC driver manager is asked to connect to a given URL it passes the URL to each loaded driver in turn.

The driver should raise a SQLException if it is the right driver to connect to the given URL, but has trouble connecting to the database.

The java.util.Properties argument can be used to passed arbitrary string tag/value pairs as connection arguments. Normally at least "user" and "password" properties should be included in the Properties.

Parameters:

url - The URL of the database to connect to

info - a list of arbitrary string tag/value pairs as connection arguments; normally at least a "user" and "password" property should be included

Returns:

a Connection to the URL

Throws: SQLException

if a database-access error occurs.

- **acceptsURL**

```
public abstract boolean acceptsURL(String url) throws SQLException
```

Returns true if the driver thinks that it can open a connection to the given URL. Typically drivers will return true if they understand the subprotocol specified in the URL and false if they don't.

Parameters:

url - The URL of the database.

Returns:

True if this driver can connect to the given URL.

Throws: SQLException

if a database-access error occurs.

- **getPropertyInfo**

```
public abstract DriverPropertyInfo[] getPropertyInfo(String url,
                                                     Properties info) throws SQL
```

The getPropertyInfo method is intended to allow a generic GUI tool to discover what properties it should prompt a human for in order to get enough information to connect to a database. Note that depending on the values the human has supplied so far, additional values may become necessary, so it may be necessary to iterate though several calls to getPropertyInfo.

Parameters:

url - The URL of the database to connect to.

info - A proposed list of tag/value pairs that will be sent on connect open.

Returns:

An array of DriverPropertyInfo objects describing possible properties. This array may be an empty array if no properties are required.

Throws: SQLException
if a database-access error occurs.

getMajorVersion

```
public abstract int getMajorVersion()
```

Get the driver's major version number. Initially this should be 1.

getMinorVersion

```
public abstract int getMinorVersion()
```

Get the driver's minor version number. Initially this should be 0.

jdbcCompliant

```
public abstract boolean jdbcCompliant()
```

Report whether the Driver is a genuine JDBC COMPLIANT (tm) driver. A driver may only report "true" here if it passes the JDBC compliance tests, otherwise it is required to return false. JDBC compliance requires full support for the JDBC API and full support for SQL 92 Entry Level. It is expected that JDBC compliant drivers will be available for all the major commercial databases. This method is not intended to encourage the development of non-JDBC compliant drivers, but is a recognition of the fact that some vendors are interested in using the JDBC API and framework for lightweight databases that do not support full database functionality, or for special databases such as document information retrieval where a SQL implementation may not be feasible.

Submit a bug or feature - Version 1.1.8 of Java Platform API Specification
Java is a trademark or registered trademark of Sun Microsystems, Inc. in the US and other countries.
Copyright 1995-1999 Sun Microsystems, Inc. 901 San Antonio Road,
Palo Alto, California, 94303, U.S.A. All Rights Reserved.