

# LAPORAN TUGAS AKHIR

## IMPLEMENTASI PEMESANAN BELANJA PADA MINIMARKET VIA SMS KHUSUS MEMBER



*Disusun oleh :*

**Nama : Fariz Rizqy A.**

**NIM : 0652902**



**JURUSAN TEKNIK ELEKTRO D-III  
KONSENTRASI TEKNIK KOMPUTER  
FAKULTAS TEKNOLOGI INDUSTRI  
INSTITUT TEKNOLOGI NASIONAL MALANG  
2010**

2010

REKAM JEJAK PERUSAHAAN  
KEMENTERIAN PERUSAHAAN  
KEMENTERIAN PERUSAHAAN  
KEMENTERIAN PERUSAHAAN

NOMOR : 000000  
TANGGAL : 00/00/00  
KEMENTERIAN PERUSAHAAN



KEMENTERIAN PERUSAHAAN  
KEMENTERIAN PERUSAHAAN

000000  
00/00/00

**LEMBAR PERSETUJUAN  
LAPORAN TUGAS AKHIR**

**IMPLEMENTASI PEMESANAN BELANJA PADA  
MINIMARKET VIA SMS KHUSUS MEMBER**



*Disusun oleh :*

Nama : Fariz Rizqy A.  
NIM : 0652902  
Jurusan : Teknik Elektro D-III  
Konsentrasi : Teknik Komputer  
Fakultas : Teknologi Industri

**Mengetahui**

**Ketua Jurusan Teknik Elektro D-III**

**Diperiksa dan Disetujui**

**Dosen Pembimbing**

(Ir. H. Taufik Hidayat, MT)  
NIP.Y.101 870 0151

(Ir. Yusuf Ismail Nakhoda, MT)  
NIP.X.1018800189

**JURUSAN TEKNIK ELEKTRO D-III  
KONSENTRASI TEKNIK KOMPUTER  
FAKULTAS TEKNOLOGI INDUSTRI  
INSTITUT TEKNOLOGI NASIONAL MALANG**

**2010**

**LEMBAR PENGESAHAN  
LAPORAN TUGAS AKHIR**

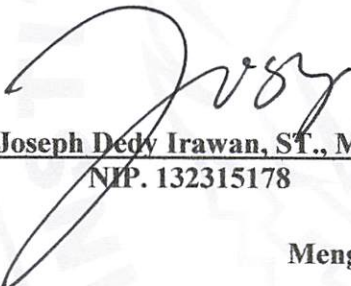
**IMPLEMENTASI PEMESANAN BELANJA PADA  
MINIMARKET VIA SMS KHUSUS MEMBER**

*Disusun oleh :*


Nama : Fariz Rizqy A.  
NIM : 0652902  
Jurusan : Teknik Elektro D-III  
Konsentrasi : Teknik Komputer  
Fakultas : Teknologi Industri

Laporan ini Telah Diperiksa dan Disahkan Oleh :

Dosen Penguji I

  
(Joseph Dedy Irawan, ST., MT)  
NIP. 132315178

Dosen Penguji II

  
(Sotyahadi, ST)  
NIP. Y. 1039700309

Mengetahui dan Menyetujui :

Ketua Jurusan Teknik Elektro D-III

  
(Ir. H. Taufik Hidayat, MT)  
NIP. Y. 1018700151

Dosen Pembimbing

  
(Ir. Yusuf Ismail Nakhoda, MT)  
NIP. Y. 1018800189

**JURUSAN TEKNIK ELEKTRO D-III  
KONSENTRASI TEKNIK KOMPUTER  
FAKULTAS TEKNOLOGI INDUSTRI  
INSTITUT TEKNOLOGI NASIONAL MALANG**

2010

## ABSTRAK

### “IMPLEMENTASI PEMESANAN BELANJA PADA MINIMARKET VIA SMS KHUSUS MEMBER”

(Fariz Rizqy A., 0652902, Teknik Komputer D-III, 64 halaman)

(Dosen Pembimbing : Ir. Yusuf Ismail Nakhoda, MT)

**Kata kunci :** Java, MySQL, SMS.

Pada zaman sekarang ini, mobilitas dan kesibukan masyarakat pada umumnya semakin meningkat karena adanya tuntutan zaman untuk memenuhi kebutuhan hidupnya. Hal ini tentu sangat berkaitan dengan kesempatan mereka meluangkan waktu untuk berbelanja guna memenuhi kebutuhan pokok. Karena didampingi dengan semakin pesatnya teknologi *cellular* dan saat ini hampir setiap kalangan sudah memilikinya sebagai identitas pribadi. Dalam kesempatan ini, penulis mencoba untuk memaparkan perencanaan dan implementasi pemesanan belanja pada *minimarket via sms khusus member* dengan menggunakan Java dan MySQL yang akan memberikan kemudahan bagi seseorang yang memiliki kesibukan yang padat dan tidak mempunyai cukup waktu untuk berbelanja dapat memesan barang-barang yang diinginkan melalui SMS saja.

## KATA PENGANTAR

Puji Syukur kepada Tuhan Yang Maha Esa atas berkat dan rahmat-Nya, sehingga kami selaku penyusun dapat menyelesaikan Laporan Tugas Akhir ini yang berjudul **“IMPLEMENTASI PEMESANAN BELANJA PADA MINIMARKET VIA SMS KHUSUS MEMBER”** dapat terselesaikan.

Adapun maksud dan tujuan dari penulisan laporan ini merupakan salah satu syarat untuk dapat menyelesaikan studi dan mendapatkan gelar Ahli Madya Jurusan Teknik Elektro D-III Konsentrasi Teknik Komputer ITN Malang.

Penyusun menyadari tanpa adanya kemauan dan usaha serta bantuan dari berbagai pihak, maka laporan ini tidak dapat diselesaikan dengan baik. Oleh sebab itu, penyusun mengucapkan terima kasih kepada yang terhormat :

1. Prof. Dr. Eng. Ir. Abraham Lomi, MSEE selaku Rektor Institut Teknologi Nasional Malang.
2. Ir. H. Taufik Hidayat, MT selaku Ketua Jurusan Teknik Elektro D-III ITN Malang.
3. Ir. Yusuf Ismail Nakhoda, MT selaku Dosen Pembimbing Tugas Akhir.
4. Sahabat-sahabat dan rekan-rekan yang tidak kami sebutkan satu-persatu, kami ucapkan banyak terima kasih atas bantuannya dalam proses Tugas Akhir yang telah kami kerjakan, begitu juga dengan penyelesaian laporan ini.

Usaha ini telah kami lakukan semaksimal mungkin, namun jika ada kekurangan dan kejanggalan dalam penyusunan, kami mohon saran dan kritik yang sifatnya membangun. Begitu juga sangat kami perlukan untuk menambah kesempurnaan laporan ini dan dapat bermanfaat bagi rekan-rekan mahasiswa pada khususnya dan pembaca pada umumnya.

Malang, 16 Pebruari 2010

Penyusun

# DAFTAR ISI

<b>LEMBAR PERSETUJUAN</b> .....	i
<b>LEMBAR PENGESAHAN</b> .....	ii
<b>ABSTRAK</b> .....	iii
<b>KATA PENGANTAR</b> .....	iv
<b>DAFTAR ISI</b> .....	vi
<b>DAFTAR GAMBAR</b> .....	x
<b>DAFTAR TABEL</b> .....	xiii
<b>BAB I PENDAHULUAN</b> .....	1
1.1. Latar Belakang .....	1
1.2. Tujuan .....	2
1.3. Batasan Masalah .....	2
1.4. Metodologi .....	2
1.5. Sistematika Penyusunan Laporan .....	3
<b>BAB II TEORI DASAR PEMROGAMAN JAVA</b> .....	5
2.1. <i>Object Oriented Programming</i> .....	5
2.2. <i>Berpikir Object Oriented</i> .....	5
2.3. Latar Belakang Bahasa Java .....	6
2.3.1. Sejarah Singkat JAVA .....	6
2.3.2. Apa itu Teknologi JAVA .....	9
2.3.2.1. Sebuah Bahasa Pemrograman .....	9



2.3.2.2. Sebuah <i>Development Environment</i> .....	9
2.3.2.3. Sebuah Aplikasi .....	9
2.3.2.2. Sebuah <i>Deployment Environment</i> .....	9
2.3.3. Mengapa Mempelajari JAVA .....	10
2.3.4. Sebagian fitur dari JAVA .....	12
2.3.4.1. Java <i>Virtual Machine</i> (JVM) .....	12
2.3.4.2. <i>Garbage Collection</i> .....	13
2.3.4.2. <i>Code Security</i> .....	14
2.3.5. Edisi JAVA .....	15
2.3.6. Fase-fase Pemrograman JAVA .....	16
2.3.7. Instalasi JAVA .....	17
2.3.8. Editor NetBeans .....	19
2.3.8.1. Membuat <i>Project</i> .....	20
2.3.8.2. Mengatur Tampilan <i>Form</i> .....	21
2.3.8.3. Menambahkan Komponen dalam <i>Form</i> .....	22
2.4. MySQL .....	24
2.5. Cara Kerja SMS .....	26
2.6. <i>Port</i> I/O Komputer .....	29
2.7. Pengiriman dan Penerimaan SMS .....	30

### **BAB III PERENCANAAN PEMBANGUNAN APLIKASI**

#### **PEMESANAN BELANJA VIA SMS DENGAN JAVA DAN**

MySQL .....	37
3.1. Blok Diagram .....	37

3.2. Alat yang digunakan untuk Implementasi Sistem .....	38
3.3. Perancangan <i>Database</i> .....	39
3.3.1. Perancangan Tabel .....	40
3.4. Contoh Daftar Barang .....	47
3.5. Perencanaan Menu Sistem Aplikasi Program .....	49
3.5.1. Struktur Program .....	49
3.5.2. Desain Laporan .....	49

#### **BAB IV IMPLEMENTASI PEMESANAN BELANJA PADA**

<b><i>MINIMARKET VIA SMS KHUSUS MEMBER</i></b> .....	52
4.1. Penjelasan dan Tampilan Program .....	52
4.1.1. <i>Form Login</i> .....	53
4.1.2. <i>Form Operator</i> .....	53
4.1.3. <i>Form Data Barang</i> .....	54
4.1.4. <i>Form Pelanggan</i> .....	55
4.1.5. <i>Form Pemesanan Baru Masuk</i> .....	55
4.1.6. <i>Form Pemesanan Terposting</i> .....	56
4.1.7. Laporan Daftar <i>Operator</i> .....	57
4.1.8. Laporan Daftar Pelanggan .....	57
4.1.9. Laporan Daftar Barang .....	58
4.1.10. Laporan Daftar Pemesanan .....	59
4.1.11. <i>SMS Server</i> .....	60

<b>BAB V PENUTUP</b> .....	63
----------------------------	----

5.1. Kesimpulan .....	63
5.2. Saran-Saran .....	63
<b>DAFTAR PUSTAKA .....</b>	<b>65</b>
<b>LAMPIRAN .....</b>	<b>66</b>

## DAFTAR GAMBAR

Gambar 2.1. Arsitektur Program .....	8
Gambar 2.2. Hirarki dan Portabilitas Teknologi Java .....	16
Gambar 2.3. Fase dari sebuah Program Java .....	16
Gambar 2.4. <i>Setting Path</i> pada Windows XP .....	18
Gambar 2.5. Proses instalasi JAVA berhasil .....	19
Gambar 2.6. Tampilan Awal NetBeans .....	20
Gambar 2.7. Penamaan <i>Project</i> Baru .....	20
Gambar 2.8. Penentuan Tipe Kelas .....	21
Gambar 2.9. Pemilihan <i>Layout</i> .....	22
Gambar 2.10. Desain <i>Form</i> dan Komponen .....	23
Gambar 2.11. Jendela Pemberian Nama <i>Event</i> .....	23
Gambar 2.12. Dialog Penamaan <i>Event</i> .....	24
Gambar 2.13. Kode <i>Event</i> .....	24
Gambar 2.14. Skema Cara Kerja SMS .....	28
Gambar 2.15. Tipe koneksi SMS <i>Center</i> .....	29
Gambar 3.1. Sistem Perencanaan Aplikasi .....	37
Gambar 3.2. Pengelolaan <i>database</i> dengan phpMyAdmin .....	39
Gambar 3.3. Pembuatan <i>database</i> sms_pemesanan .....	39
Gambar 3.4. Pembuatan tabel pelanggan .....	40
Gambar 3.5. Penamaan <i>field-field</i> pada tabel pelanggan .....	41
Gambar 3.6. Pembuatan tabel <i>operator</i> .....	41
Gambar 3.7. Penamaan <i>field-field</i> pada tabel operator .....	42

Gambar 3.8. Pembuatan tabel barang .....	42
Gambar 3.9. Penamaan <i>field-field</i> pada tabel barang .....	43
Gambar 3.10. Pembuatan tabel kirim .....	43
Gambar 3.11. Penamaan <i>field-field</i> pada tabel kirim .....	44
Gambar 3.12. Pembuatan tabel terima .....	44
Gambar 3.13. Penamaan <i>field-field</i> pada tabel terima .....	45
Gambar 3.14. Pembuatan tabel pemesanan_d .....	45
Gambar 3.15. Penamaan <i>field-field</i> pada tabel pemesanan_d .....	46
Gambar 3.16. Pembuatan tabel pemesanan .....	46
Gambar 3.17. Penamaan <i>field-field</i> pada tabel pemesanan .....	47
Gambar 3.18. Struktur program aplikasi <i>admin</i> .....	49
Gambar 3.19. Rancangan laporan Daftar <i>Operator</i> .....	50
Gambar 3.20. Rancangan laporan Daftar Pelanggan .....	50
Gambar 3.21. Rancangan laporan Daftar Barang .....	50
Gambar 3.22. Rancangan laporan Daftar Pemesanan .....	51
Gambar 4.1. Menu Utama .....	52
Gambar 4.2. <i>Form Login</i> .....	53
Gambar 4.3. <i>Form Operator</i> .....	54
Gambar 4.4. <i>Form Data Barang</i> .....	54
Gambar 4.5. <i>Form Pelanggan</i> .....	55
Gambar 4.6. <i>Form Pemesanan Baru Masuk</i> .....	56
Gambar 4.7. <i>Form Pemesanan Terposting</i> .....	56
Gambar 4.8. Tampilan Daftar <i>Operator</i> .....	57
Gambar 4.9. Tampilan Daftar Pelanggan .....	58

Gambar 4.10. Tampilan Daftar Barang .....	59
Gambar 4.11. Tampilan Daftar Pemesanan .....	60
Gambar 4.12. SMS <i>Server</i> .....	61
Gambar 4.13. Balasan dari SMS <i>Server</i> pada modem .....	62
Gambar 4.14. Penulisan <i>format</i> pemesanan pada HP .....	62
Gambar 4.15. Balasan dari SMS Sever pada HP .....	62

## DAFTAR TABEL

Tabel 2.1. Ringkasan Fase dari sebuah Program Java .....	17
Tabel 2.2. Skema <i>Format</i> SMS PDU Pengirim .....	31
Tabel 2.3. PDU <i>Type</i> (Pengirim) .....	32
Tabel 2.4. Skema <i>Format</i> SMS PDU Penerima .....	34
Tabel 2.5. PDU <i>Type</i> (Penerima) .....	34
Tabel 3.1. Keterangan pembuatan tabel pelanggan .....	40
Tabel 3.2. Keterangan tabel <i>operator</i> .....	41
Tabel 3.3. Keterangan tabel barang .....	42
Tabel 3.4. Keterangan tabel kirim .....	43
Tabel 3.5. Keterangan tabel terima .....	44
Tabel 3.6. Keterangan tabel pemesanan_d .....	45
Tabel 3.7. Keterangan tabel pemesanan .....	46
Tabel 3.8. Detil barang yang dipesan .....	48
Tabel 3.9. Contoh daftar harga barang khusus <i>member</i> .....	48

# BAB I

## PENDAHULUAN

### 1.1 Latar Belakang

Dengan kemajuan teknologi dan zaman yang semakin berkembang maka aktivitas yang dilakukan manusia menjadi semakin padat, maka dari itu seseorang tidak akan mampu melakukan beberapa aktivitas secara bersamaan.

Salah satu perkembangan dari teknologi yang cukup pesat adalah *handphone/cellular* dengan fitur yang disertakan tidak hanya berguna untuk melakukan panggilan dan sms saja tetapi saat ini *handphone* saat ini juga disertai fitur-fitur seperti kamera, *internet (browsing dan chatting)* yang merupakan salah satu menu wajib untuk *handphone* keluaran saat ini.

Karena saat ini hampir setiap orang memiliki *handphone* sebagai salah satu identitas pribadi, maka dengan *handphone* inilah saya akan mencoba mengimplementasikan pemesanan belanja pada *minimarket via sms* agar jika aktivitas seseorang yang setiap harinya cukup padat tidak perlu untuk belanja pada *minimarket* terlebih dahulu untuk menghemat waktu.

Kenapa harus memakai *handphone*? Karena seperti uraian diatas *handphone* sudah banyak dimiliki hampir setiap orang berbeda dengan *internet* yang harganya saat ini sudah semakin terjangkau tetapi masih kalangan tertentu saja yang menggunakannya.



## 1.2 Tujuan

Adapun tujuan dari pembuatan Tugas Akhir ini adalah untuk membuat dan mengimplementasikan pemesanan belanja pada *minimarket* via sms khusus *member* yang dapat memudahkan seseorang untuk berbelanja.

## 1.3. Batasan Masalah

Pada tugas akhir ini permasalahan yang akan dibahas adalah sebagai berikut:

- a) Desain tampilan dari program pengolah *database*.
- b) Pengolah *database* yang digunakan adalah MYSQL
- c) Aplikasi ini hanya meliputi daftar pemesanan pada *item* yang ada pada *minimarket*.
- d) Daftar *item* belanja berupa kode angka untuk menghemat karakter sms
- e) Pelanggan sudah mendaftarkan nomor *handphone* atau sudah daftar sebagai *member* pada *minimarket*

## 1.4 Metodologi

Untuk mencapai tujuan diatas maka ditumpuh langkah-langkah sebagai berikut:

1. Penentuan dan pengumpulan literatur

Mengumpulkan dan mempelajari literatur sehubungan dengan permasalahan yang dihadapi, seperti bahasa pemrograman yang digunakan.

2. Studi literatur tentang teori penunjang tugas akhir.

Mempelajari secara teoritis dan praktis tentang cara kerja dari SMS *Gateway* serta peralatan penunjang lain (HP dan kabel data).

3. Perancangan sistem

Melakukan perancangan sistem tugas akhir secara umum, yaitu membuat program aplikasi yang digunakan untuk SMS *Gateway*.

4. Pembuatan sistem

Membuat tugas akhir bagian per bagian dimulai dari membuat *database* sampai membuat program pengolahnya.

5. Pengujian sistem

Melakukan pengujian dan analisa terhadap hasil program aplikasi SMS *Gateway* yang telah dibuat.

6. Penyempurnan sistem

Perbaikan terhadap kekurangan dan penyempurnaan dari sistem yang dibuat agar sesuai dengan harapan.

7. Penyusunan buku

Menyimpulkan hasil perencanaan dan pembuatan serta penyempurnaan aplikasi dengan hasil pengujian sehingga tersusunlah buku laporan tugas akhir.

### **1.5 Sistematika Pembahasan**

Setelah dilakukan proses pelaksanaan dan implementasi hasil pada tugas akhir ini, mulai dari studi literatur, perencanaan, pembuatan, pengujian dan perbaikan, serta analisa dan hasil – hasil yang didapat maka untuk pembahasan

selengkapnya diwujudkan dalam bentuk buku laporan tugas akhir ini dengan sistematika sebagai berikut:

**BAB I : PENDAHULUAN**

Pada bab ini membahas pendahuluan yang terdiri dari latar belakang, tujuan, metodologi, batasan masalah, dan sistematika pembahasan tugas akhir.

**BAB II : TEORI PENUNJANG**

Pada bab ini meliputi literatur, teori dasar, serta referensi, yang berguna sebagai acuan dan landasan bagi penulis.

**BAB III : PERANCANGAN SISTEM APLIKASI**

Membahas analisis kebutuhan perangkat lunak dan perancangan perangkat lunak

**BAB IV : IMPLEMENTASI DAN PENGUJIAN HASIL**

Membahas implementasi hasil dari perancangan, pengujian dan analisa program yang dibuat

**BAB V : KESIMPULAN DAN SARAN**

Pada bab ini berisikan kesimpulan dan juga saran serta masukan setelah melihat hasil analisa dari pengujian hasil. Untuk meningkatkan hasil akhir yang lebih baik diberikan saran-saran terhadap hasil pembuatan tugas akhir.

## BAB II

### TEORI DASAR PEMROGRAMAN JAVA

#### **2.1 *Object Oriented Programming***

Teknologi *Object Oriented* memandang software sebagai sebuah interaksi antarbagian atau subsistem dalam sebuah sistem, dan menggambarkan satu bagian tersebut dalam satu objek yang memiliki sifat/properti/data dan kemampuan untuk melakukan suatu tugas tertentu<sup>2</sup>.

Contoh, objek manusia memilikitangan dan kaki itu dipandang sebagai sifat/data (properti), sedangkan memegang dan berlari adalah pekerjaan/tugas-tugas yang bisa dilakukan (metode). Jika diperhatikan, sebetulnya properti dan metode memiliki hubungan walaupun tidak selamanya harus ada (memegang dengan tangan, berlari dengan kaki).

OOP (*Object Oriented Programming*) adalah metode pemrograman yang mengikuti konsep-konsep berorientasi objek, yakni *encapsulation* (pengapsulan), *inheritance* (pewarisan), *polimorfisme* (keragaman/bentuk lain).

#### **2.2 Berpikir *Object Oriented***

Berpikir OOP adalah selalu melihat persoalan program sebagai objek yang memiliki sekumpulan data dan metode, serta menggabungkan data dan metode dalam satu objek dan meniadakan/mengurangi penggunaan variabel global dalam program<sup>2</sup>.

Dalam pemrograman klasik (pemrograman terstruktur) biasanya memisahkan data dan metode. Akibatnya metode akan disimpan dalam sebuah

unit tersendiri. Begitu juga dengan datanya sehingga baik data dan metode akan bersifat global.

Untuk aplikasi berskala kecil, mungkin belum menimbulkan persoalan. Akan tetapi, jika aplikasi menjadi besar, akan ditambahkan terus metode demi metode, variabel demi variabel sehingga suatu saat *programmer* bisa terjebak pada kondisi menulis metode yang berfungsi sama lebih dari sekali dengan nama yang berbeda. Atau *programmer* akan mengecek sudah ada apa belum metode baruyang akan dituliskan, dengan membaca seluruh unit yang sudah dibuat satu demi satu (ratusan atau mungkin ribuan). Belum lagi tentang variabelnya yang berada dimana-mana. Kepala bisa pusing sebelum sempat menyelesaikan persoalan logik program yang sebenarnya.

Konsep objek selalu menyatukan data sebagai badan dan metode sebagai roh dalam satu wadah layaknya manusia hidup. Akan tetapi, pemrograman klasik memisahkan data sebgai badan manusia dan metode sebagai roh. Akibatnya manusia bisa menjadi hantu dan mayat (kurang relevan dengan kenyataan).

## **2.3 Latar Belakang Bahasa JAVA**

### **2.3.1 Sejarah Singkat JAVA**

Pada 1991, sekelompok insinyur Sun dipimpin oleh Patrick Naughton dan James Gosling ingin merancang bahasa komputer untuk perangkat konsumen seperti *cable TV Box*. Dikarenakan perangkat tersebut tidak memiliki banyak memori, bahasa harus berukuran kecil dan mengandung kode yang liat. Juga karena manufaktur-manufaktur berbeda memilih *processor* yang berbeda pula, maka bahasa harus bebas dari manufaktur manapun. Proyek diberi nama kode

”Green”<sup>8</sup>.

Kebutuhan untuk fleksibilitas, kecil, liat dan kode yang netral terhadap *platform* mengantar tim mempelajari implementasi Pascal yang pernah dicoba. Niklaus Wirth, pencipta bahasa Pascal telah merancang bahasa portabel yang menghasilkan *intermediate code* untuk mesin hipotesis. Mesin ini sering disebut dengan mesin maya (*virtual machine*). Kode ini kemudian dapat digunakan di sembarang mesin yang memiliki *interpreter*. Proyek Green menggunakan mesin maya untuk mengatasi isu utama tentang netral terhadap arsitektur mesin.

Karena orang-orang di proyek Green berbasis C++ dan bukan Pascal maka kebanyakan sintaks diambil dari C++, serta mengadopsi orientasi objek dan bukan prosedural. Mulanya bahasa yang diciptakan diberi nama ”Oak” oleh James Gosling yang mendapat inspirasi dari sebuah pohon yang berada pada seberang kantornya, namun dikarenakan nama Oak sendiri merupakan nama bahasa pemrograman yang telah ada sebelumnya, kemudian SUN menggantinya dengan JAVA. Nama JAVA sendiri terinspirasi pada saat mereka sedang menikmati secangkir kopi di sebuah kedai kopi yang kemudian dengan tidak sengaja salah satu dari mereka menyebutkan kata JAVA yang mengandung arti asal biji kopi. Akhirnya mereka sepakat untuk memberikan nama bahasa pemrograman tersebut dengan nama Java.

Produk pertama proyek Green adalah Star 7 (\*7), sebuah kendali jarak jauh yang sangat cerdas. Dikarenakan pasar masih belum tertarik dengan produk konsumen cerdas maka proyek Green harus menemukan pasar lain dari teknologi yang diciptakan. Pada saat yang sama, implementasi WWW dan Internet sedang mengalami perkembangan pesat. Di lain pihak, anggota dari proyek Green juga

menyadari bahwa Java dapat digunakan pada pemrograman *internet*, sehingga penerapan selanjutnya mengarah menjadi teknologi yang berperan di web.

Bahasa/Alat pengembangan	Arsitektur Program			
	Modul Web Server	Scripting Web Server	Modul Web Browser	Scripting Web Browser
Java	servlet	JSP	Applet	Javascript
C++	CGI exe		ActiveX*	
Perl	CGI script			
Phyton	CGI script			
PHP		PHP script		
Visual Basic		ASP*	ActiveX*	VB Script*

\*) Hanya di landasan Windows, tidak bisa di Linux.

Gambar 2.1 Arsitektur Program<sup>8</sup>

Java telah mengakomodasi hampir seluruh fitur penting bahasa-bahasa pemrograman yang ada semenjak perkembangan komputasi *modern* manusia<sup>8</sup>:

1. Dari SIMULA, bahasa pada tahun 65-an, bahasa yang paling mempengaruhi Java sekaligus C++. Dari bahasa ini diadopsi bentukan-bentukan dasar dari pemrograman berorientasi objek.
2. Dari LISP, bahasa tahun 55-an. Diadopsi fasilitas *garbage collection*, serta kemampuan untuk meniru *generic list processing*, meski fasilitas ini jarang yang memanfaatkannya.
3. Dari Algol, bahasa pada tahun 60-an, diambil struktur kendali yang dimilikinya.
4. Dari C++, diadopsi sintaks, sebagian semantiks dan *exception handling*.
5. Dari bahasa Ada, diambil *strongly type*, dan *exception handling*.
6. Dari *Objective C*, diambil fasilitas *interface*.
7. Dari bahasa SmallTalk, diambil pendekatan *single-root class hiérarchie*, dimana *object* adalah satu kesatuan hirarki pewarisan.

8. Dari bahasa Eiffel, fasilitas *assertion* yang mulai diterapkan di sebagian JDK 1.4

## **2.3.2 Apa itu Teknologi JAVA**

### **2.3.2.1 Sebuah Bahasa Pemrograman**

Sebagai sebuah bahasa pemrograman, Java dapat membuat seluruh bentuk aplikasi, *desktop*, *web* dan lainnya, sebagaimana dibuat dengan menggunakan bahasa pemrograman konvensional yang lain<sup>8</sup>.

Java adalah bahasa pemrograman yang berorientasi objek (OOP) dan dapat dijalankan pada berbagai *platform* sistem operasi. Perkembangan Java tidak hanya terfokus pada satu sistem operasi, tetapi dikembangkan untuk berbagai sistem operasi dan bersifat *open source*.

### **2.3.2.2 Sebuah *Development Environment***

Sebagai sebuah peralatan pembangun, teknologi Java menyediakan banyak *tools*: *compiler*, *interpreter*, penyusun dokumentasi, paket kelas dan sebagainya.

### **2.3.2.3 Sebuah Aplikasi**

Aplikasi dengan teknologi Java secara umum adalah aplikasi serba guna yang dapat dijalankan pada seluruh mesin yang memiliki *Java Runtime Environment (JRE)*<sup>8</sup>.

### **2.3.2.4 Sebuah *Deployment Environment***

Terdapat dua komponen utama dari *Deployment Environment*. Yang



pertama adalah JRE, yang terdapat pada paket J2SDK, mengandung kelas-kelas untuk semua paket teknologi Java yang meliputi kelas dasar dari Java, komponen GUI dan sebagainya. Komponen yang lain terdapat pada *Web Browser*. Hampir seluruh *Web Browser* komersial menyediakan *interpreter* dan *runtime environment* dari teknologi Java<sup>8</sup>.

### 2.3.3 Mengapa Mempelajari JAVA

Berdasarkan *white paper* resmi dari SUN, Java memiliki karakteristik berikut<sup>8</sup>:

1. Sederhana

Bahasa pemrograman Java menggunakan sintaks mirip dengan C++ namun sintaks pada Java telah banyak diperbaiki terutama menghilangkan penggunaan pointer yang rumit dan *multiple inheritance*. Java juga menggunakan *automatic memory allocation* dan *memory garbage collection*.

2. Berorientasi objek (*Object Oriented*)

Java menggunakan pemrograman berorientasi objek yang membuat program dapat dibuat secara modular dan dapat dipergunakan kembali. Pemrograman berorientasi objek memodelkan dunia nyata kedalam objek dan melakukan interaksi antar objek-objek tersebut.

3. Dapat didistribusi dengan mudah

Java dibuat untuk membuat aplikasi terdistribusi secara mudah dengan

adanya *libraries* networking yang terintegrasi pada Java.

#### 4. *Interpreter*

Program Java dijalankan menggunakan *interpreter* yaitu *Java Virtual Machine* (JVM). Hal ini menyebabkan *source code* Java yang telah dikompilasi menjadi *Java bytecodes* dapat dijalankan pada *platform* yang berbeda-beda.

#### 5. *Robust*

Java mempunyai reliabilitas yang tinggi. *Compiler* pada Java mempunyai kemampuan mendeteksi *error* secara lebih teliti dibandingkan bahasa pemrograman lain. Java mempunyai *runtime-Exception handling* untuk membantu mengatasi *error* pada pemrograman.

#### 6. Aman

Sebagai bahasa pemrograman untuk aplikasi *internet* dan terdistribusi, Java memiliki beberapa mekanisme keamanan untuk menjaga aplikasi tidak digunakan untuk merusak sistem komputer yang menjalankan aplikasi tersebut.

#### 7. *Architecture Neutral*

Program Java merupakan *platform independent*. Program cukup mempunyai satu buah versi yang dapat dijalankan pada *platform* yang berbeda dengan *Java Virtual Machine*.

## 8. *Portable*

*Source code* maupun program Java dapat dengan mudah dibawa ke *platform* yang berbeda-beda tanpa harus dikompilasi ulang.

## 9. *Performance*

*Performance* pada Java sering dikatakan kurang tinggi. Namun *performance* Java dapat ditingkatkan menggunakan kompilasi Java lain seperti buatan CodeGear, Microsoft ataupun Symantec yang menggunakan *Just In Time Compilers (JIT)*.

## 10. *Multithreaded*

Java mempunyai kemampuan untuk membuat suatu program yang dapat melakukan beberapa pekerjaan secara sekaligus dan simultan.

## 11. Dinamis

Java didesain untuk dapat dijalankan pada lingkungan yang dinamis. Perubahan pada suatu *class* dengan menambahkan properties ataupun *method* dapat dilakukan tanpa mengganggu program yang menggunakan *class* tersebut.

### 2.3.4 Sebagian Fitur dari JAVA

#### 2.3.4.1 Java *Virtual Machine (JVM)*

JVM adalah sebuah mesin imajiner (maya) yang bekerja dengan menyerupai aplikasi pada sebuah mesin nyata. JVM menyediakan spesifikasi

*hardware* dan *platform* dimana kompilasi kode Java terjadi. Spesifikasi inilah yang membuat aplikasi berbasis Java menjadi bebas dari *platform* manapun karena proses kompilasi diselesaikan oleh JVM<sup>8</sup>.

Aplikasi program Java diciptakan dengan *file* teks berekstensi *.java*. Program ini dikompilasi menghasilkan satu berkas *bytecode* berekstensi *.class* atau lebih. *Bytecode* adalah serangkaian instruksi serupa instruksi kode mesin. Perbedaannya adalah kode mesin harus dijalankan pada sistem komputer dimana kompilasi ditujukan, sementara *bytecode* berjalan pada *java interpreter* yang tersedia di semua *platform* sistem komputer dan sistem operasi.

#### **2.3.4.2 Garbage Collection**

Banyak bahasa pemrograman lain yang mengizinkan seorang programmer mengalokasikan memori pada saat dijalankan. Namun, setelah menggunakan alokasi memori tersebut, harus terdapat cara untuk menempatkan kembali blok memori tersebut supaya program lain dapat menggunakannya. Dalam C, C++ dan bahasa lainnya, adalah programmer yang mutlak bertanggung jawab akan hal ini. Hal ini dapat menyulitkan bilamana *programmer* tersebut alpa untuk mengembalikan blok memori sehingga menyebabkan situasi yang dikenal dengan nama *memory leaks*<sup>8</sup>.

Program Java melakukan *garbage collection* yang berarti program tidak perlu menghapus sendiri objek-objek yang tidak digunakan lagi. Fasilitas ini mengurangi beban pengelolaan memori oleh *programmer* dan mengurangi atau mengeliminasi sumber kesalahan terbesar yang terdapat pada bahasa yang memungkinkan alokasi dinamis.

### 2.3.4.3 Code Security

*Code Security* terimplementasi pada Java melalui penggunaan *Java Runtime Environment (JRE)*. Java menggunakan model pengamanan 3 lapis untuk melindungi sistem dari *untrusted Java Code*<sup>8</sup>.

1. Pertama, *class-loader* menangani pemuatan kelas Java ke *runtime interpreter*. Proses ini menyediakan pengamanan dengan memisahkan kelas-kelas yang berasal dari *local disk* dengan kelas-kelas yang diambil dari jaringan. Hal ini membatasi aplikasi Trojan karena kelas-kelas yang berasal dari *local disk* yang dimuat terlebih dahulu.
2. Kedua, *bytecode verifier* membaca *bytecode* sebelum dijalankan dan menjamin *bytecode* memenuhi aturan-aturan dasar bahasa Java.
3. Ketiga, manajemen keamanan menangani keamanan tingkat aplikasi dengan mengendalikan apakah program berhak mengakses sumber daya seperti sistem file, *port* jaringan, proses eksternal dan sistem *windowing*.

Setelah seluruh proses tersebut selesai dijalankan, barulah kode program di eksekusi. Java juga menyediakan beragam teknik pengamanan lain:

1. Bahasa dirancang untuk mempersulit eksekusi kode perusak. Peniadaan *pointer* merupakan langkah besar pengamanan. Java tidak mengenal operasi *pointer*. Di tangan programmer handal, operasi *pointer* merupakan hal yang luar biasa untuk optimasi dan pembuatan program yang efisien serta mengagumkan. Namun mode ini dapat menjadi petaka di hadapan programmer jahat. *Pointer* merupakan sarana luar biasa untuk pengaksesan tak diotorisasi. Dengan peniadaan operasi *pointer*, Java dapat menjadi

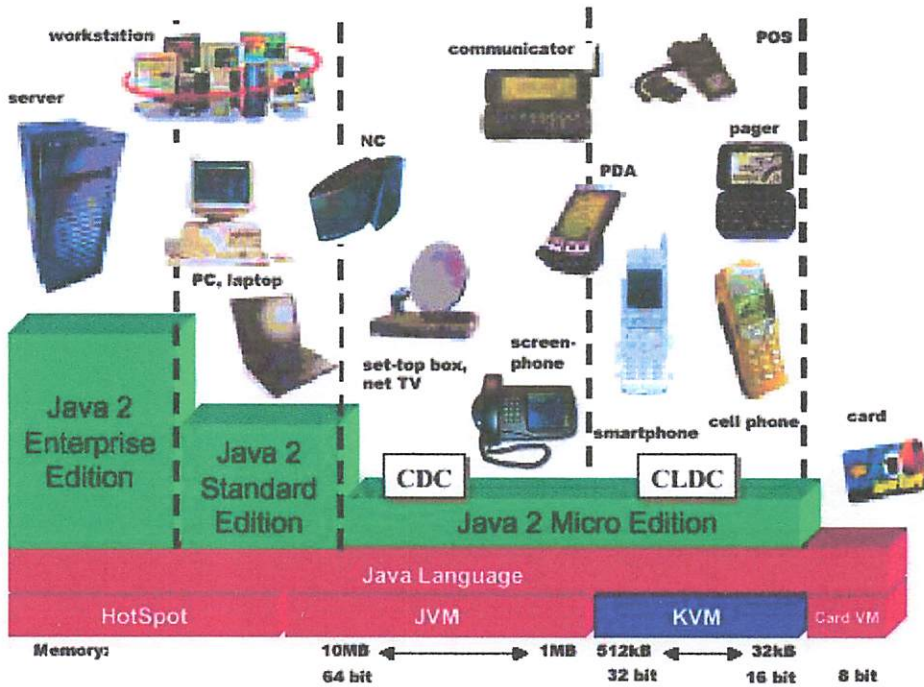
bahasa yang lebih aman.

2. Java memiliki beberapa pengamanan terhadap *applet*. Untuk mencegah program bertindak mengganggu media penyimpanan, maka *applet* tidak diperbolehkan melakukan *open*, *read* ataupun *write* terhadap berkas secara sembarangan. Karena Java *applet* dapat membuka jendela *browser* yang baru, maka jendela mempunyai logo Java dan teks identifikasi terhadap jendela yang dibuka. Hal ini mencegah jendela *pop-up* menipu sebagai permintaan keterangan *username* dan *password*.

### 2.3.5 Edisi JAVA

Bukan sekedar bahasa tapi juga *platform* sekaligus arsitektur. Java mempunyai portabilitas yang sangat tinggi. Ia dapat berada pada smartcard, pager, POS (*Point of Service*), *handphone*, PDA, palm, TV, *Embedded device* (PLC, *microcontroller*), laptop, pc, dan bahkan *server*). Menyadari akan hal ini (*one size doesn't fit all*) Sun membagi arsitektur Java menjadi tiga bagian, yaitu<sup>7</sup>:

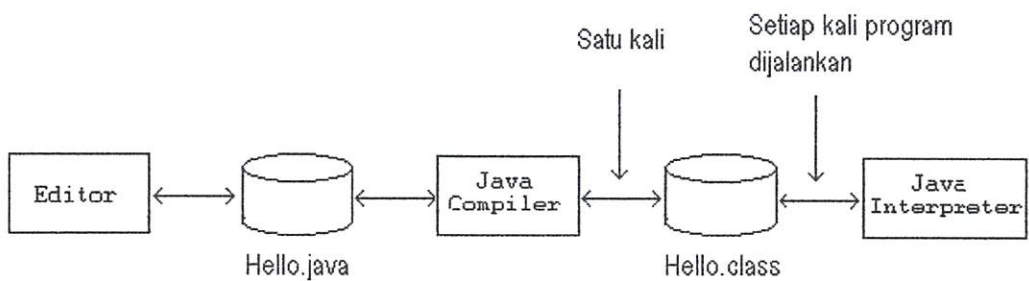
1. Enterprise Java (J2EE) untuk aplikasi berbasis *web*, aplikasi sistem tersebar dengan beraneka ragam klien dengan kompleksitas yang tinggi. Merupakan *superset* dari Standar Java
2. Standar Java (J2SE), ini adalah yang biasa kita kenal sebagai bahasa Java, dan merupakan fokus kita sekarang.
3. Micro Java (J2ME) merupakan *subset* dari J2SE dan salah satu aplikasinya yang banyak dipakai adalah untuk *wireless device / mobile device*



Gambar 2.2 Hirarki dan Portabilitas Teknologi Java<sup>7</sup>

### 2.3.6 Fase-fase Pemrograman JAVA

Gambar dibawah ini menjelaskan aliran proses kompilasi dan eksekusi sebuah program Java<sup>8</sup>:



Gambar 2.3 Fase dari sebuah Program Java<sup>8</sup>

Langkah pertama dalam pembuatan sebuah program berbasis Java adalah menuliskan kode program pada *text editor*. Contoh *text editor* yang dapat digunakan antara lain: notepad, vi, emacs dan lain sebagainya. Kode program

yang dibuat kemudian tersimpan dalam sebuah berkas berekstensi *.java*.

Setelah membuat dan menyimpan kode program, kompilasi file yang berisi kode program tersebut dengan menggunakan Java Compiler. Hasil dari kompilasi berupa berkas *bytecode* dengan ekstensi *.class*.

Berkas yang mengandung *bytecode* tersebut kemudian akan dikonversikan oleh Java Interpreter menjadi bahasa mesin sesuai dengan jenis dan *platform* yang digunakan.

Tabel 2.1 Ringkasan Fase dari sebuah Program Java<sup>8</sup>

<i>Proses</i>	<i>Tool</i>	<i>Hasil</i>
Menulis kode program	<i>Text Editor</i>	Berkas berekstensi <i>.java</i>
Kompilasi program	Java Compiler	Berkas berekstensi <i>.class</i> (Java Bytecodes)
Menjalankan program	Java Interpreter	Program Output

### 2.3.7 Instalasi JAVA

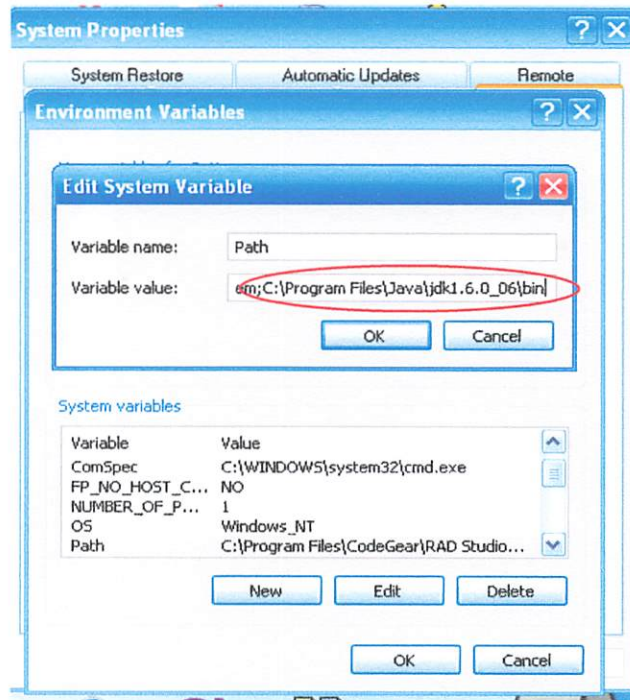
Untuk mendapatkan JDK Java terakhir, kita bisa langsung membuka situs <http://java.sun.com>. Proses instalasi seperti halnya pada aplikasi program berbasis Windows lainnya, yaitu jalankan file installer-nya dan ikuti dialog instalasinya dari lisensi sampai penentuan direktori program<sup>2</sup>.

Pada saat menginstall kita harus menentukan *Path* dan *ClassPath*. Perlu diketahui, *Path* adalah pemetaan suatu direktori pada sistem operasi agar direktori tersebut bisa dikenal secara global pada sistem. Sedangkan *Classpath* adalah untuk memberitahukan pada Java bahwa sebuah *class* java tertentu (bisa berbentuk jar) kita simpan pada direktori/alamat file tertentu.



Setting PATH pada Windows:

Masuklah pada halaman **Control Panel** → **System**, sehingga muncul jendela **System Properties**. Lanjutkan dengan memilih **Advanced**, kemudian klik tombol **Environment Variables**, klik pada **System Variables** → **Path**, dan tekan tombol **Edit**.



Gambar 2.4 Setting Path pada Windows XP

Pada jendela **Edit SystemVariable**, ketik pada **variable value ;C:\Program Files\Java\jdk1.6.0\_06\bin** . Lalu restart komputer, setelah di restart lakukan tes apakah java kita sudah terinstall dengan benar dengan, buka **command prompt**, ketik **javac**, jika kita install dengan benar maka dia akan tampil seperti gambar di bawah ini.

```

C:\WINDOWS\system32\cmd.exe
Microsoft Windows XP [Version 5.1.2600]
(C) Copyright 1985-2001 Microsoft Corp.

C:\Documents and Settings\Batro>javac
Usage: javac <options> <source files>
where possible options include:
  -g                Generate all debugging info
  -g:none           Generate no debugging info
  -g:(lines,vars,source) Generate only some debugging info
  -nowarn           Generate no warnings
  -verbose          Output messages about what the compiler is doing
  -deprecation      Output source locations where deprecated APIs are used
  -classpath <path> Specify where to find user class files and annotation processors
  -cp <path>        Specify where to find user class files and annotation processors
  -sourcepath <path> Specify where to find input source files
  -bootclasspath <path> Override location of bootstrap class files
  -extdirs <dirs>    Override location of installed extensions
  -endorseddirs <dirs> Override location of endorsed standards path
  -proc:(none,only) Control whether annotation processing and/or compilation is done.
  -processor <class1>[,<class2>,<class3>...] Names of the annotation processors to run; bypasses default discovery process
  -processorpath <path> Specify where to find annotation processors
  -d <directory>     Specify where to place generated class files
  -s <directory>     Specify where to place generated source files
  -implicit:(none,class) Specify whether or not to generate class files for implicitly referenced files
  -encoding <encoding> Specify character encoding used by source files
  -source <release>  Provide source compatibility with specified release
  -target <release>  Generate class files for specific VM version
  -version           Version information
  -help             Print a synopsis of standard options
  -Xkey=value1      Options to pass to annotation processors
  -X               Print a synopsis of nonstandard options
  -J<flag>          Pass <flag> directly to the runtime system

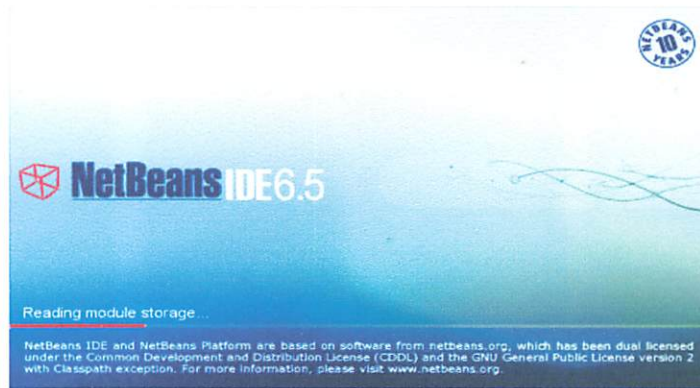
C:\Documents and Settings\Batro>

```

Gambar 2.5 Proses instalasi JAVA berhasil

### 2.3.8 Editor NetBeans

*Editor* ini cukup luar biasa untuk membuat aplikasi Java karena didukung dengan fasilitas drag and drop komponen. Yaitu dukungan *Rapid Application Development* (RAD) atau pemrograman berbasis *visual*. Satu hal lagi yang paling penting adalah produk ini *Free*, yang dibuat oleh Sun Microsystem<sup>3</sup>.



Gambar 2.6 Tampilan Awal NetBeans

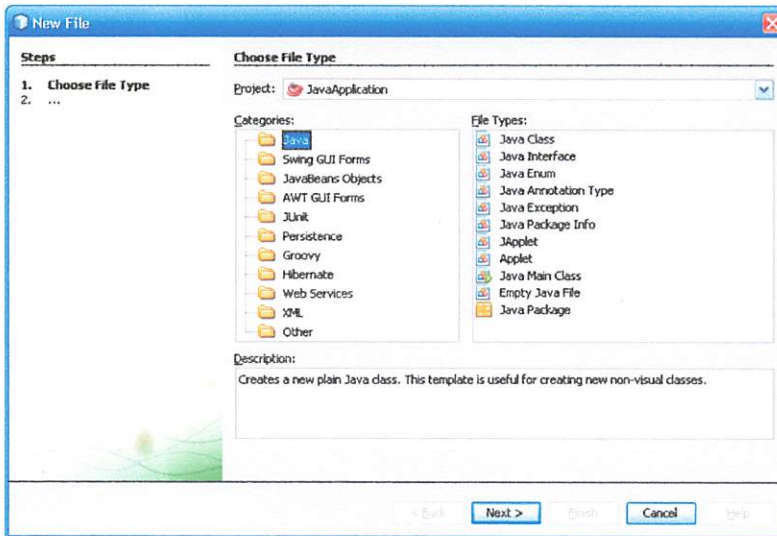
### 2.3.8.1 Membuat *Project*

Untuk membuat *project* pertama dengan menggunakan *Editor* NetBeans adalah dengan cara sebagai berikut<sup>3</sup>:

1. Pilih menu **File** → **New Project**.
2. Pada jendela New Project pilih **Java** pada kolom **Categories**.
3. Pada kolom **Project** pilih **Java Application** untuk form atau **Java Dekstop Application** untuk MDI/multi form, lanjutkan dengan mengklik tombol **Next**.
4. Pada langkah Name And Location masukkan nama projectnya dan tentukan juga lokasi serta foldernya

Gambar 2.7 Penamaan *Project* Baru

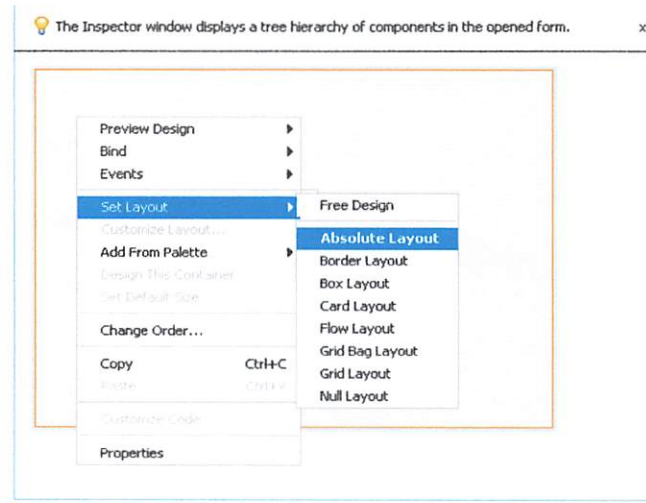
5. Jika pengaturan telah selesai, klik tombol **Finish**.
6. Menambahkan form file baru berupa form/kelas/paket, bisa memulai dari jendela Projects, pilih dan klik kanan nama proyek kemudian pilih **New → Other**.



Gambar 2.8 Penentuan Tipe Kelas

### 2.3.8.2 Mengatur Tampilan *Form*

Untuk memudahkan *design form*, klik kanan pada form kosong → **Set Layout** → **AbsoluteLayout**. Peletakan ini didasarkan pada koordinat form<sup>3</sup>.

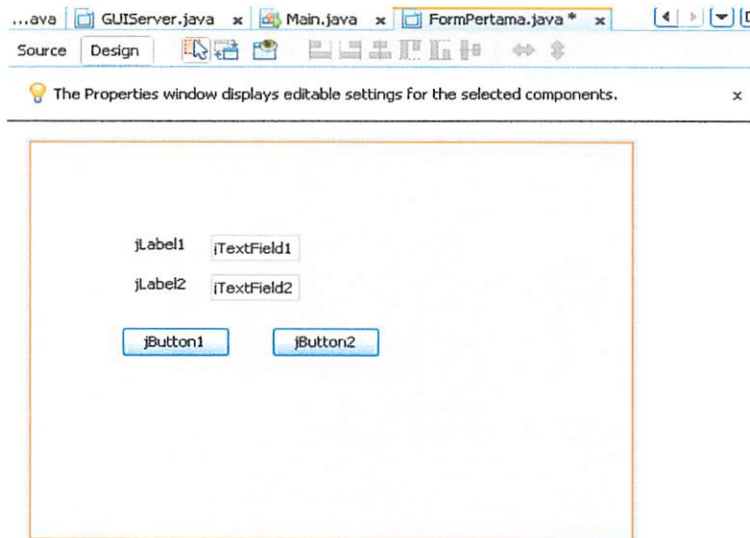


Gambar 2.9 Pemilihan Layout

### 2.3.8.3 Menambahkan Komponen dalam *Form*

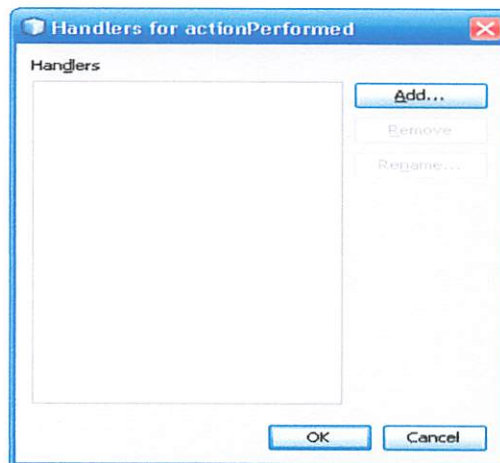
Komponen tersedia dalam palet sebelah kanan, mulai dari palet *Swing* sampai *Borders*, untuk meletakkan dalam *form*, berikut langkahnya<sup>3</sup>:

1. Klik sekali salah satu komponen, kemudian klik dalam area *form*, maka objek form akan masuk area form.
2. Atur beberapa propertinya melalui jendela **Properties** sebelah kiri, untuk memberikan event klik tab **Event**.



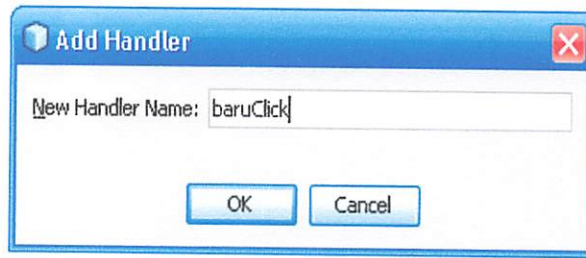
Gambar 2.10 Desain Form dan Komponen

3. Untuk menambahkan event, klik pada tab Event, selanjutnya pilih jenis event.
4. Klik tanda ... di sebelah kanan nama event tersebut, kemudian klik tombol **Add**.



Gambar 2.11 Jendela Pemberian Nama Event

5. Pada dialog **Add Handler** tepatnya pada kolom masukan **New Handler Name**, masukkan nama "baruClick" dan akhiri dengan mengklik **OK**



Gambar 2.12 Dialog Penamaan Event

- Setelah itu kita akan dibawa kedalam kode event tersebut seperti gambar di bawah.

```
private void baruClick(java.awt.event.ActionEvent evt) {  
    // TODO add your handling code here:  
}
```

Gambar 2.13 Kode Event

## 2.4 MySQL

### 2.4.1 Sejarah MySQL

MySQL adalah sebuah perangkat lunak sistem manajemen basis data SQL (Database management System) atau DBMS yang *multithread*, *multiuser*, dengan sekitar 6 juta instalasi di seluruh dunia. MySQL AB membuat MySQL tersedia sebagai *software* gratis dibawah lisensi GNU General Public License (GPL), tetapi mereka juga menjual dibawah lisensi komersial untuk kasus-kasus dimana penggunaannya tidak cocok dengan penggunaan GPL. MySQL dimiliki dan disponsori oleh sebuah perusahaan komersial Swedia MySQL AB, dimana memegang hak cipta hampir atas semua kode sumbernya. Kedua orang Swedia dan satu orang Finlandia yang mendirikan MySQL AB adalah: David Axmark, Allan Larsson, dan Michael "Monty" Widenius<sup>6</sup>.

## 2.4.2 Bahasa SQL

Dalam konteks bahasa SQL, pada umumnya informasi tersimpan dalam tabel-tabel yang secara logic merupakan struktur dua dimensi yang terdiri atas baris-baris data (*row* atau *record*) yang berada dalam satu atau lebih kolom. Baris pada tabel sering disebut sebagai instance dari data, sedangkan kolom sering disebut sebagai atribut atau *field*<sup>6</sup>.

Keseluruhan tabel yang terhimpun dalam satu kesatuan disebut dengan *database*. SQL adalah bahasa pemrograman yang digunakan dalam pembuatan *database* dan tabel-tabel. Di SQL terdapat 3 kategori perintah yaitu: DDL, DML, DCL.

### 2.4.2.1 DDL (*Data Definition Language*)

DDL ini berfungsi lebih ke dalam memanipulasi struktur dari *database*. Contohnya DDL ini bisa digunakan untuk membuat tabel atau menghapus tabel. Kita juga bisa membuat *key* atau *index* dengan menggunakan DDL ini, membuat relasi antar tabel juga bisa dilakukan dengan DDL ini<sup>6</sup>.

Beberapa statemen atau sintaks yang sering dijumpai dalam DDL adalah sebagai berikut<sup>6</sup>:

- CREATE TABLE, bertugas untuk membuat tabel
- ALTER TABLE, bertugas untuk merubah struktur suatu tabel
- DROP TABLE, bertugas untuk menghapus suatu tabel
- CREATE INDEX, bertugas untuk membuat suatu index dalam tabel
- DROP INDEX, bertugas untuk menghapus suatu index dalam tabel



#### **2.4.2.2 DML (*Data Manipulation Language*)**

SQL adalah sekumpulan sintaks-sintaks atau statemen untuk mengakses data dalam *database*, tetapi SQL sendiri juga bisa digunakan untuk melakukan proses *insert*, *update* atau *delete* ke dalam suatu *database*. Sintaks-sintaks ini yang disebut dengan *Data manipulation Language* (DML) yang merupakan bagian dari SQL. Berikut ini adalah penjelasan singkat dari sintaks-sintaks tersebut<sup>6</sup>.

- SELECT, bertugas untuk mengakses data dari suatu tabel dalam *database*
- UPDATE, bertugas untuk mengupdate (merubah) data dalam suatu tabel pada *database*
- DELETE, bertugas untuk menghapus data dari suatu tabel dalam *database*
- INSERT, bertugas untuk menambahkan data ke dalam suatu tabel dalam *database*

#### **2.4.2.3 DCL (*Data Control Language*)**

DCL digunakan untuk membuat suatu rule atau aturan, *system permission* dan integritas referensial yang baik untuk mengontrol terhadap akses data ke dalam *database* dengan cara mengamankannya. Contoh dari DCL adalah GRANT, dan statement REVOKE.

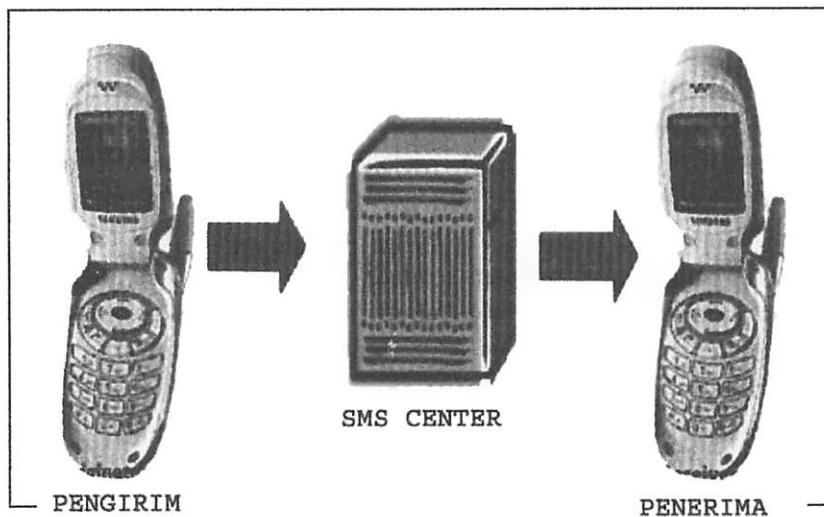
### **2.5 Cara Kerja SMS**

*Short Message Service* (SMS) merupakan salah satu fitur dari GSM yang dikembangkan dan distandarisasi oleh *European Telecommunication Standard Institute*(ETSI), yang memberikan kemudahan kepada masyarakat dalam penyampaian pesan yang singkat secara cepat<sup>9</sup>.

### 2.5.1 Konsep SMS

Pada saat kita mengirim pesan SMS dari *handphone* (*mobile originated*) maka pesan tersebut tidak langsung dikirim ke *handphone* tujuan (*mobile terminated*), melainkan pesan tersebut dikirimkan terlebih dahulu ke SMS Center (SMSC) dengan prinsip *Store and Forward*, setelah itu baru dikirimkan ke *handphone* yang dituju<sup>9</sup>.

Dengan adanya SMSC, kita dapat mengetahui status pesan SMS yang telah dikirim, apakah telah sampai atau gagal diterima oleh *handphone* tujuan. Apabila *handphone* tujuan dalam keadaan aktif dan menerima SMS yang dikirim, ia akan mengirimkan kembali pesan konfirmasi ke SMSC yang menyatakan bahwa SMS telah diterima. Kemudian SMSC mengirimkan kembali status tersebut kepada si pengirim. Tetapi jika *handphone* tujuan dalam keadaan mati atau diluar jangkauan SMS yang dikirimkan akan disimpan pada SMSC sampai periode validitas terpenuhi. Jika periode validitas terlewat maka SMS itu akan dihapus dari SMSC dan tidak dikirimkan ke *handphone* tujuan. Di samping itu, SMSC juga akan mengirim pesan informasi ke nomor pengirim yang menyatakan pesan yang dikirim belum diterima atau gagal.



Gambar 2.14 Skema Cara Kerja SMS

### 2.5.2 Koneksi ke SMSC

Untuk dapat mengirim dan menerima pesan, kita harus melakukan koneksi ke SMSC. Ada beberapa cara untuk melakukan koneksi ke SMSC antara lain :

1. Menggunakan terminal baik berupa GSM modem atau *handphone*

Cara ini adalah yang paling mudah, tetapi memiliki kekurangan antara lain jumlah pesan yang dikirim per menit sangat terbatas (sekitar 6-10 pesan per menit). Untuk mengantisipasi hal ini biasanya digunakan lebih dari satu terminal.

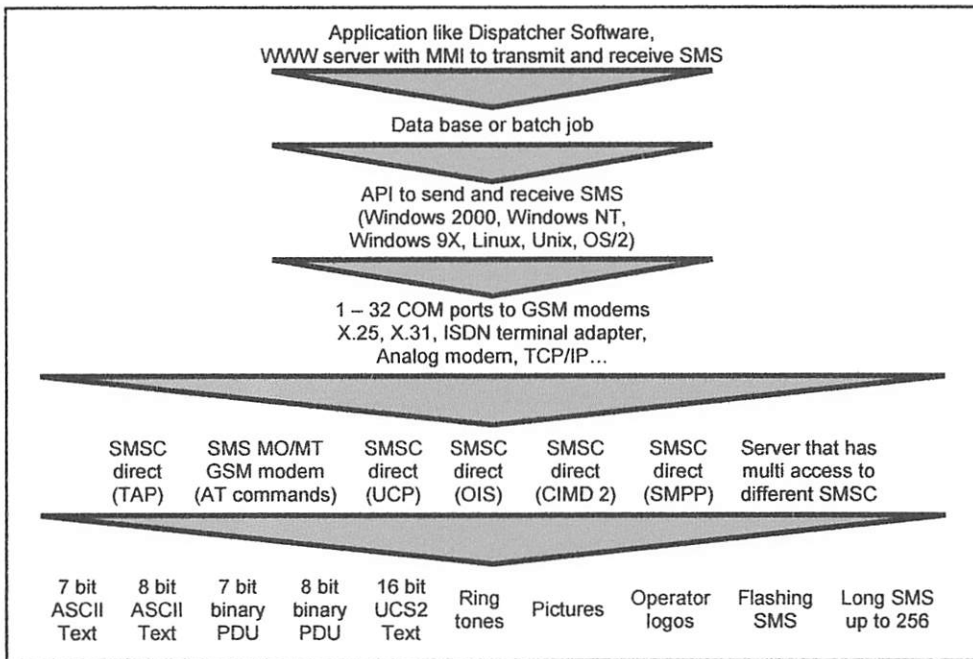
2. Koneksi langsung ke SMSC

Dengan melakukan koneksi langsung ke SMSC kita dapat mengirim pesan dalam jumlah banyak, dapat mencapai sekitar 600 sms per menit bergantung pada kapasitas dari SMSC itu sendiri. Untuk melakukan koneksi ke SMSC diperlukan protokol penghubung. Protokol yang umum digunakan adalah UCP, SMPP, CIMD2, OIS dan TAP. Masing-masing operator GSM menyediakan tipe protokol yang berbeda-beda.

3. Menggunakan *software* bantu

Saat ini banyak vendor telekomunikasi menawarkan *software* bantu untuk melakukan koneksi ke SMSC, dari yang bersifat *freeware*, *open source* sampai dengan komersial.

Pemilihan koneksi ke SMSC biasanya disesuaikan dengan jumlah pesan SMS yang akan dikirim.



Gambar 2.15 Tipe koneksi SMS Center<sup>9</sup>

## 2.6 Port I/O Komputer

Komputer adalah alat pemroses data yang akurat dan cepat serta merupakan salah satu komponen penting dalam sistem berbasis komputer. *Port* merupakan tempat keluar masuknya data dari dan ke dalam komputer, sehingga memungkinkan komputer berkomunikasi dengan perangkat lain yang terhubung pada *port* tersebut. Di dalam komputer terdapat beberapa port yang berbeda cara

kerja dan karakteristiknya, *port I/O* tersebut ada yang bekerja 1 bit dan ada juga yang bekerja 8 bit. Pemilihan *port* untuk sistem teleoperasi ini disesuaikan dengan kebutuhan pembangunan sistem.

Dalam pembangunan sistem informasi akademik ini, *port* yang dipakai adalah *port COM1*, karena disesuaikan dengan handphone yang digunakan yaitu Siemens M55 dengan menggunakan kabel data RS232. Dalam kasus lain dengan *handphone* dan kabel data yang berbeda, maka harus disesuaikan dengan *Serial Port* yang ada di komputer.

## **2.7 Pengiriman dan Penerimaan SMS**

Dalam pengiriman dan penerimaan pesan SMS sesungguhnya terdapat dua *mode*, yaitu *mode teks* dan *mode PDU (Protocol Data Unit)*.

### **2.7.1 Mode Teks**

Mode ini adalah cara termudah untuk mengirim pesan. Pada *mode teks*, pesan yang dikirim tidak dilakukan konversi. *Teks* yang dikirim tetap dalam bentuk aslinya yang merupakan hasil *encode* yang direpresentasikan dalam *format PDU*. Kelemahannya, kita tidak dapat menyisipkan gambar dan nada dering ke dalam pesan yang akan dikirim serta terbatasnya *tipe encoding*. Sistem *mode teks* juga tidak didukung oleh semua *operator GSM* ataupun terminal.

### **2.7.2 Mode PDU**

Mode PDU adalah *format* pesan dalam bentuk oktet heksadesimal dan oktet semidesimal dengan panjang mencapai 160 (7 bit) atau 140 (8 bit) karakter. Kelebihan menggunakan mode PDU adalah kita dapat melakukan *encoding*

sendiri yang tentunya harus pula didukung oleh hardware dan operator GSM, melakukan kompresi data, menambahkan nada dering dan gambar pada pesan yang akan dikirim.

Pada pengiriman pesan terdapat dua jenis mobile, yaitu *Mobile Terminated (Handphone Penerima)* dan *Mobile Originated (Handphone Pengirim)*.

### 2.7.2.1 SMS PDU Pengirim (*Mobile Originated*)

SMS PDU Pengirim adalah pesan yang dikirim dari handphone ke terminal yang kemudian dikirimkan ke SMSC. Pada prinsipnya apabila kita mengirim pesan ke nomor tujuan, pesan itu akan melalui SMSC<sup>9</sup>.

Pesan yang akan dikirimkan oleh terminal masih dalam bentuk *teks*, sedangkan dalam pengiriman ke SMSC harus dalam bentuk PDU. Untuk itu sebelum dikirim, terminal atau *handphone* akan melakukan perubahan dari *format teks* menjadi format PDU, proses inilah yang kemudian kita sebut *encodec*. Berikut adalah skema dari format PDU Pengirim yang telah diatur dan ditetapkan oleh ETSI.

Tabel 2.2 Skema *Format SMS PDU Pengirim*<sup>9</sup>

SCA	PDU Type	MR	DA	PID	DCS	VP	UDL	UD
-----	----------	----	----	-----	-----	----	-----	----

Penjelasannya adalah sebagai berikut<sup>9</sup>:

a. *Service Center Address (SCA)*

SCA adalah informasi dari alamat (nomor) SMSC. SCA memiliki tiga komponen utama, yaitu *len*, *type of number* dan *service center number*.

Dalam pengiriman pesan SMS, nomor SMSC tidak dicantumkan.

b. *Protocol Data Unit (PDU)*

Nilai *default* dari PDU Type untuk SMS pengirim adalah 11 hexa, yang memiliki arti bahwa 11 hexa=00010001

Tabel 2.3 PDU Type (Pengirim)<sup>9</sup>

Bit no	7	6	5	4	3	2	1	0
Nama	RP	UDHI	SRR	VPF	VPF	RD	MTI	MTI
Nilai	0	0	0	1	0	0	0	1

Keterangan<sup>9</sup>:

- RP : *Reply Path*. Parameter yang menunjukkan bahwa alur jawaban ada.
- UDHI : *User Data Header Indicator*. Bit ini bernilai 1 jika data pengirim dimulai dengan suatu judul/tema.
- SRR : *Status Report Request*. Bit ini bernilai 1 jika laporan status pengiriman diminta.
- VPF : *Validity Period Format*. *Format* batas waktu pengiriman jika pesan gagal.
- RD : *Reject Duplicates*. Parameter yang menandakan ya atau tidaknya *Service Center* akan menerima suatu pengiriman pesan SMS suatu pesan yang masih disimpan dalam *Service Center* tersebut.
- MTI : *Message Type Indicator*. Bit ini bernilai 0 untuk menunjukkan bahwa PDU ini adalah suatu SMS-Deliver.

c. *Message Reference (MR)*

*Message Reference* adalah acuan dari pengeturan pesan SMS. Untuk membiarkan pengaturan pesan SMS dilakukan sendiri oleh *handphone* tujuan, maka nilai yang diberikan adalah "00".

d. *Destination Address (DA)*

*Destination Address* adalah alamat (nomor) tujuan, yang terdiri atas panjangnya nomor tujuan (*Len*), format dari nomor tujuan (*Type Number*) dan nomor tujuan (*Destination Number*)

e. *Protocol Identifier (PID)*

*Protocol Identifier* adalah tipe atau format dari cara pengiriman pesan, yang biasanya diatur dari *handphone* pengirim. Misalnya tipe *Standard Text*, *Fax*, *E-mail*, *Telex*, *X400* dan lain-lain.

f. *Data Coding Scheme (DCS)*

*Data Coding Scheme* adalah rencana dari pengkodean data untuk menentukan kelas dari pesan tersebut apakah berupa SMS teks standar, *Flash SMS*, atau *Blinking SMS*.

g. *Validity Period (VP)*

*Validity Period* adalah lama waktu pesan SMS disimpan di SMSC apabila pesan tersebut gagal diterima oleh *handphone* (nomor) tujuan.

h. *User Data Length (UDL)*

*User Data Length* adalah panjangnya pesan SMS yang akan dikirim dalam bentuk teks standar.

i. *User Data (UD)*



*User Data* adalah isi pesan yang akan dikirim dalam format heksadesimal.

### 2.7.2.2 SMS PDU Penerima (*Mobile Terminated*)

SMS PDU Penerima adalah terminal menerima pesan yang datang atau masuk dari SMSC ke handphone dalam *format* PDU. Pada prinsipnya pesan yang kita terima dari SMSC masih dalam *format* PDU setelah itu terminal *handphone* yang menerima pesan akan melakukan pengkodean menjadi *teks*, proses ini yang selanjutnya kita sebut dengan *decodec*. Berikut adalah skema dari format PDU Penerima yang telah diatur dan ditetapkan oleh ETSI<sup>9</sup>.

Tabel 2.4 Skema *Format* SMS PDU Penerima<sup>9</sup>

SCA	PDU Type	OD	PID	DCS	SCTS	UDL	UD
-----	----------	----	-----	-----	------	-----	----

Penjelasannya adalah sebagai berikut<sup>9</sup>:

a. *Service Center Address* (SCA)

*Service Center Address* adalah alamat (nomor) dari SMSC. SCA memiliki tiga komponen utama yaitu *len*, *type of number* dan *service center number*.

b. *PDU Type*

Nilai *default* dari *PDU Type* untuk *SMS-Deliver* adalah 04 hexa, yang memiliki arti 04 hexa=00000100.

Tabel 2.5 *PDU Type* (Penerima)<sup>9</sup>

Bit no	7	6	5	4	3	2	1	0
Nama	RP	UDHI	SRI	<nn>	<nn>	MMS	MTI	MTI
Nilai	0	0	0	0	0	1	0	0

**Keterangan<sup>9</sup>:**

**RP** : *Reply Path*. Parameter yang menunjukkan bahwa alur jawaban ada.

**UDHI** : *User Data Header Indicator*. Bit ini bernilai 1 jika data pengirim dimulai dengan suatu judul/tema.

**SRI** : *Status Report Indication*. Bit ini bernilai 1 jika suatu status laporan akan dikembalikan ke SME.

**MMS** : *More Messages to Send*. Bit ini bernilai 0 jika ada pesan lebih yang akan dikirim.

**MTI** : *Message Type Indicator*. Bit ini bernilai 0 untuk menunjukkan bahwa PDU ini adalah suatu *SMS-Deliver*.

**c. Originator Address (OA)**

*Originator Address* adalah alamat (nomor) dari pengirim, yang terdiri atas penjangnya nomor pengirim (*Len*), format dari nomor pengirim (*Type Number*) dan nomor pengirim (*Originator Number*).

**d. Protocol Identifier (PID)**

*Protocol Identifier* adalah tipe atau format dari cara pengiriman pesan, yang biasanya diatur dari *handphone* pengirim. Misalnya tipe *Standard Text*, *Fax*, *E-mail*, *Telex*, *X400* dan lain-lain

**e. Data Coding Scheme (DCS)**

*Data Coding Scheme* adalah rencana dari pengkodean data untuk menentukan kelas dari pesan tersebut apakah berupa *SMS teks* standar, *Flash SMS*, atau *Blinking SMS*.

f. *Service Center Time Stamp (SCTS)*

*Service Center Time Stamp* adalah waktu dari penerimaan pesan oleh SMSC penerima. SCTS terdiri atas tahun, bulan, tanggal, jam, menit dan detik, serta zona waktu.

g. *User Data Length (UDL)*

*User Data Length* adalah panjangnya pesan SMS yang akan dikirim dalam bentuk *teks* standar.

h. *User Data (UD)*

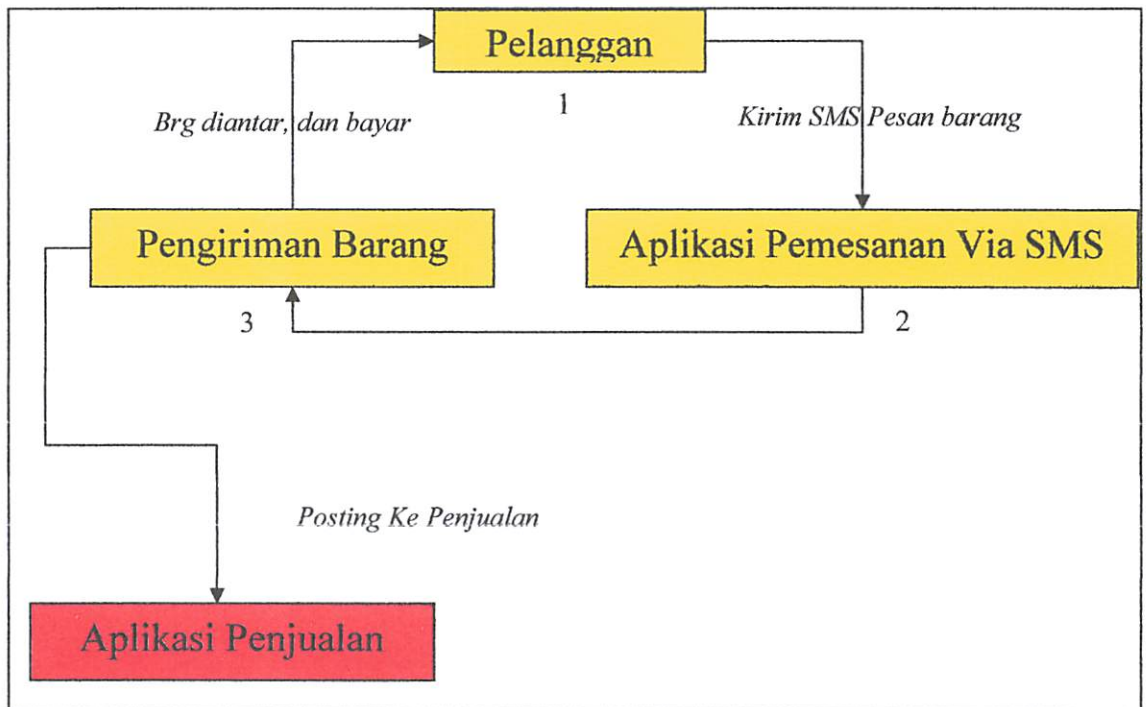
*User Data* adalah isi pesan yang akan dikirim dalam *format* heksadesimal.

# BAB III

## PERENCANAAN PEMBANGUNAN APLIKASI PEMESANAN BELANJA VIA SMS DENGAN JAVA DAN MySQL

### 3.1 Blok Diagram

Adapun perencanaan sistem aplikasi pemesanan belanja *via* sms seperti pada gambar dibawah ini.



Gambar 3.1 Sistem Perencanaan Aplikasi

Dari proses yang terdapat pada diagram di atas, warna kuning adalah batasan implementasi yang dibuat dengan keterangan sebagai berikut:

1. Pelanggan atau member adalah pihak pemesanan yang mengirimkan perintah berupa SMS dengan *format* yang sudah ditetapkan dan akan diterima/diolah oleh SMS *Server*

2. Aplikasi Pemesanan *Via* SMS disini bertugas untuk mengolah SMS yang masuk dari pelanggan atau *member* yang melakukan transaksi pemesanan
3. Pengiriman Barang akan dilakukan jika dua poin diatas sudah memenuhi syarat yang benar.

### **3.2 Alat yang digunakan untuk implementasi sistem**

#### **1. Perangkat Keras**

Perangkat keras yang digunakan dalam penelitian ini adalah:

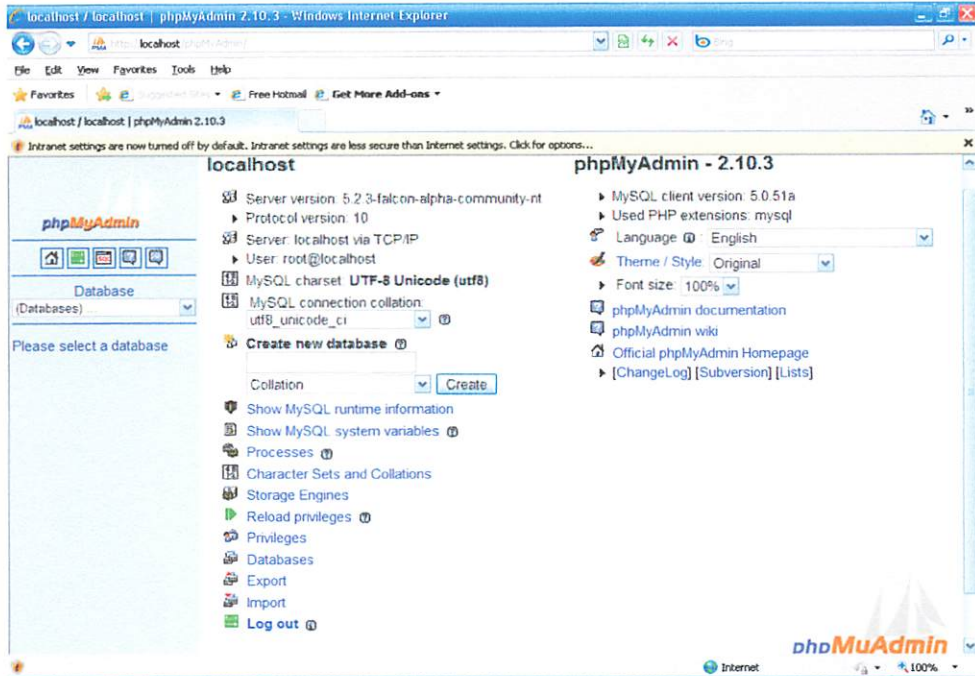
- a. 1 buah PC (*Personal Computer*) dengan spesifikasi sebagai berikut:
  - a. *Processor* Intel Pentium 4 2.67 GHz.
  - b. RAM 2 GB
  - c. *Harddisk* 160 GB
  - d. *Monitor* LG 15" T530S
  - e. *Handphone* Siemens C55 beserta kabel data

#### **2. Perangkat Lunak**

- a. Sistem Operasi Windows XP SP 2 *Professional*
- b. NetBeans IDE 6.5
- c. MYSQL 5.2
- d. AppServ Open Project – 2.5.10 for Windows

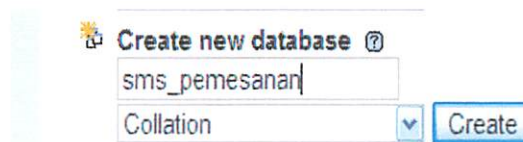
### 3.3 Perancangan Database

Database Aplikasi ini dibuat dengan menggunakan database MySQL dengan bantuan phpMyAdmin yang sudah menjadi satu paket *installer* dalam AppServ seperti pada gambar berikut.



Gambar 3.2 Pengelolaan *database* dengan phpMyAdmin

Dalam program aplikasi sms ini menggunakan satu *database* dengan nama yang didalamnya terdapat banyak tabel-tabel. *Database* dibuat dengan cara menuliskan nama *database* seperti pada gambar berikut:



Gambar 3.3 Pembuatan *database* sms\_pemesanan

Setelah membuat *database* menekan tombol *create*, maka *database* dengan nama sms\_pemesanan telah terbentuk untuk kemudian diisi dengan tabel-tabel yang diperlukan.

### 3.3.1 Perancangan Tabel

Dalam program aplikasi sms ini, harus dibuat beberapa tabel dengan beberapa ketentuan sesuai kebutuhan. Tabel-tabel yang harus dibuat antara lain:

#### 3.3.1.1 Tabel Pelanggan

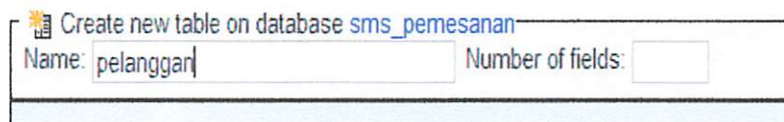
Tabel pelanggan dibuat dengan tujuan untuk menyimpan data-data pelanggan yang menjadi anggota/*member*. Data-data pelanggan yang harus disimpan adalah seperti yang tertera pada tabel berikut:

Tabel 3.1 Keterangan pembuatan tabel pelanggan

No.	Nama Field	Type Data	Panjang	Keterangan
1	kd_pel	int	3	<i>auto_increment, key</i>
2	nm_pel	varchar	25	
3	no_telp	varchar	25	
4	no_hp	varchar	12	
5	alamat	varchar	50	
6	pin	varchar	4	

Berdasarkan keterangan diatas, maka dibuat tabel dengan menggunakan phpMyAdmin dengan urutan sebagai berikut:

1. Membuat nama tabel dengan cara seperti pada gambar berikut



Create new table on database sms\_pemesanan  
Name: pelanggan Number of fields:

Gambar 3.4 Pembuatan tabel pelanggan

2. Memasukkan nama *field* dan kriteria-kriteria lainnya seperti pada gambar berikut:

Field	Type	Length/Values <sup>1</sup>	Collation	Attributes	Null	Default <sup>2</sup>	Extra
kd_pel	INT	3		UNSIGNED ZEROFILL	not null		auto_increment
nm_pel	VARCHAR	25			not null		
no_telp	VARCHAR	25			not null		
no_hp	VARCHAR	12			not null		
alamat	VARCHAR	50			not null		
pin	VARCHAR	4			not null		

Table comments:

Storage Engine: MyISAM

Collation:

Save Or Add 1 field(s) Go

Gambar 3.5 Penamaan field-field pada tabel pelanggan

Dengan menekan tombol *save* maka sebuah tabel dengan nama pelanggan telah terbentuk dan siap diisi dengan daftar pelanggan

### 3.3.1.2 Tabel Operator

Tabel *operator* dibuat untuk menyimpan data-data dari *operator* yang berhak menjalankan program aplikasi SMS dengan kriteria seperti tertera pada tabel dibawah ini.

Tabel 3.2 Keterangan tabel operator

No.	Nama Field	Type Data	Panjang	Keterangan
1	kd_operator	char	4	key
2	nm_operator	varchar	45	
3	kelamin	char	1	
4	alamat	varchar	100	
5	user_id	varchar	25	
6	pass_id	varchar	25	

Pembuatan nama tabel dilakukan dengan bantuan phpMyAdmin seperti pada gambar berikut:

Create new table on database sms\_pemesanan

Name: operator Number of fields: 6

Gambar 3.6 Pembuatan tabel operator



Setelah menentukan nama tabel, selanjutnya adalah *memasukkan field-field* berdasarkan ketentuan atau atribut yang telah dibuat sebelumnya seperti pada gambar berikut.

Field	Type	Length/Values	Collation	Attributes	Null	Default	Extra
kd_operator	CHAR	4			not null		
nm_operator	VARCHAR	45			not null		
kelamin	CHAR	1			not null		
alamat	VARCHAR	100			not null		
user_id	VARCHAR	25			not null		
pass-id	VARCHAR	25			not null		

Table comments:

Storage Engine: MyISAM

Collation:

Or Add 1 field(s)

Gambar 3.7 Penamaan *field-field* pada tabel *operator*

### 3.3.1.3 Tabel Barang

Tabel barang digunakan untuk menyimpan nama-nama barang yang dijual pada pelanggan, data-data tabel barang yang harus disimpan adalah seperti yang tertera pada tabel berikut:

Tabel 3.3 Keterangan tabel barang

No.	Nama Field	Type Data	Panjang	Keterangan
1	kd_brg	varchar	4	key
2	nm_brg	varchar	30	
3	harga	double		

Langkah selanjutnya adalah membuat tabel barang dengan menggunakan phpMyAdmin seperti pada gambar berikut:

Create new table on database sms\_pemesanan

Name:  Number of fields:

Gambar 3.8 Pembuatan tabel barang

Setelah pembuatan nama tabel selesai, selanjutnya adalah penentuan *type* data, nama atribut/*field* seperti yang terdapat pada gambar berikut:

Field	Type	Length/Values	Collation	Attributes	Null	Default	Extra
kd_brg	VARCHAR	4			not null		
nm_brg	VARCHAR	30			not null		
harga	DOUBLE				not null		

Table comments:

Storage Engine: MyISAM

Collation:

Save Or Add 1 field(s) Go

Gambar 3.9 Penamaan *field-field* pada tabel barang

### 3.3.1.4 Tabel Kirim

Tabel ini digunakan untuk menyimpan pesan yang sudah terkirim pada pengirim SMS. Tabel ini dibuat dengan kriteria seperti tertera pada tabel dibawah ini:

Tabel 3.4 Keterangan tabel kirim

No.	Nama Field	Type Data	Panjang	Keterangan
1	id	Int	5	<i>auto increment, key</i>
2	notelp	Varchar	13	
3	pesan	Varchar	255	
4	date	Date		
5	status	Char	1	

Pembuatan nama tabel dilakukan dengan menggunakan phpMyAdmin seperti pada gambar berikut

Create new table on database sms\_pemesanan

Name: kirim Number of fields: 5

Gambar 3.10 Pembuatan tabel kirim

Penentuan *type* data dan atribut-atribut tabel lainnya dilakukan dengan cara seperti ditunjukkan pada gambar berikut

Field	Type	Length/Values <sup>1</sup>	Collation	Attributes	Null	Default <sup>2</sup>	Extra
id	INT	5		UNSIGNED ZEROFILL	not null		auto_increment
notelp	VARCHAR	13			not null		
pesan	VARCHAR	255			not null		
date	DATE				not null		
status	CHAR	1			not null		

Table comments:

Storage Engine: MyISAM

Collation:

Save Or Add 1 field(s) Go

Gambar 3.11 Penamaan *field-field* pada tabel kirim

### 3.3.1.5 Tabel Terima

Tabel terima digunakan untuk menyimpan SMS masuk dari pelanggan/*member* yang melakukan transaksi. Ketentuan dari tabel tersebut adalah seperti tertera pada tabel dibawah ini.

Tabel 3.5 Keterangan tabel terima

No.	Nama Field	Type Data	Panjang	Keterangan
1	Id	Int	5	<i>auto_increment, key</i>
2	No telp	Varchar	13	
3	Pesan	Varchar	255	
4	Date	Varchar	100	
5	status	Varchar	100	

Pembuatan nama tabel dilakukan dengan menggunakan phpMyAdmin seperti pada gambar berikut

Create new table on database sms\_pemesanan

Name:  Number of fields:

Gambar 3.12 Pembuatan tabel terima

Setelah pembuatan nama tabel, selanjutnya adalah penentuan nama atribut-atribut tabel yang akan kita buat seperti ditunjukkan gambar berikut

Field	Type	Length/Values	Collation	Attributes	Null	Default	Extra
id	INT	5		UNSIGNED ZEROFILL	not null		auto_increment
notelp	VARCHAR	13			not null		
pesan	VARCHAR	255			not null		
date	VARCHAR	100			not null		
status	VARCHAR	100			not null		

Table comments:

Storage Engine: MyISAM

Collation:

Save Or Add 1 field(s) Go

Gambar 3.13 Penamaan *field-field* pada tabel terima

### 3.3.1.6 Tabel Pemesanan\_d

Tabel pemesanan\_d digunakan untuk menampung data pemesanan pelanggan yang berasal dari tabel terima yang sudah dilakukan pemecahan sebelumnya menurut item barang. Atribut-atribut yang terdapat pada tabel pemesanan\_d ini adalah seperti ditunjukkan tabel dibawah ini.

Tabel 3.6 Keterangan tabel pemesanan\_d

No.	Nama Field	Type Data	Panjang	Keterangan
1	no_pesanan	Int	5	
2	kd_brg	Varchar	4	
3	harga	Double		
4	jumlah	Double		

Pembuatan nama tabel dilakukan dengan menggunakan phpMyAdmin seperti pada gambar berikut

Create new table on database sms\_pemesanan

Name:  Number of fields:

Gambar 3.14 Pembuatan tabel pemesanan\_d

Langkah selanjutnya adalah memasukkan nama-nama field atau atribut-atribut yang telah ditentukan seperti ditunjukkan pada gambar dibawah ini

Gambar 3.15 Penamaan *field-field* pada tabel pemesanan\_d

### 3.3.1.7 Tabel Pemesanan

Tabel pemesanan ini digunakan untuk menyimpan setiap transaksi yang dilakukan oleh pelanggan sebagai nomor faktur. Tabel ini dibuat dengan kriteria seperti tertera pada tabel dibawah ini

Tabel 3.7 Keterangan tabel pemesanan

No.	Nama Field	Type Data	Panjang	Keterangan
1	no_pesan	Int	5	<i>auto increment, key</i>
2	tgl	Date		
3	kd_pel	Int	3	
4	posted	Char	1	

Pembuatan nama tabel dilakukan dengan menggunakan phpMyAdmin seperti pada gambar berikut

Gambar 3.16 Pembuatan tabel pemesanan

Yang dilakukan selanjutnya adalah memasukkan nama *field* atau atribut-atribut yang telah ditentukan seperti ditunjukkan pada gambar dibawah ini

Field	Type	Length/Values	Collation	Attributes	Null	Default	Extra
no_pesanan	INT	5		UNSIGNED ZEROFILL	not null		auto_increment
tgl	DATE				not null		
kd_pel	INT	3			not null		
posted	CHAR	1			not null		

Table comments:

Storage Engine: MyISAM

Collation:

Save Or Add 1 field(s) Go

Gambar 3.17 Penamaan *field-field* pada tabel pemesanan

### 3.4 Contoh Daftar Barang

Agar pelanggan tidak bingung dalam memesan barang maka akan ditetapkan *format* SMS yang akan memudahkan pelanggan yang ingin bertransaksi. Format SMS adalah sebagai berikut:

**BELI<spasi><PIN>(<kd brg>,<jml>)(<kd brg>,<jml>)**

Contoh : BELI 0011(B01,5)

Artinya : BELI = transaksi pembelian,  
 0011 = Pin pelanggan  
 B01 = kode barang  
 5 = jumlah barang

Contoh lain;

Contoh : BELI 0012(B01,5)(BC02,10)(B33, 40)

Artinya : BELI =>transaksi pembelian,  
 0012 =>Pin pelanggan

Tabel 3.8 Detil barang yang dipesan

No	Kode barang	Jumlah
1	B01	5
2	BC02	10
3	B33	40

*Format* penulisan kode barang dan jumlahnya selalu diawali dengan tanda kurung buka dan diakhiri kurung tutup, untuk memisahkan kode barang dan jumlahnya digunakan koma. Dan agar pelanggan tidak bingung maka setiap transaksi pemesanan hanya dibatasi 10 item barang saja, disini agar jumlah karakter yang diketikkan pada *handphone* tidak terlalu banyak. Setiap transaksi yang dilakukan, pelanggan akan menerima SMS balik tentang jumlah item yang dipesan beserta jumlah harga total barang yang telah dipesan. Pembayaran akan dilakukan pemesan jika barang sudah diantar beserta dengan nota belanja.

Supaya pelanggan/*member* yang sudah terdaftar tidak perlu mengingat nama barang dan kodenya maka setiap pelanggan akan diberi daftar nama barang beserta kode barang. Contoh nya adalah sebagai berikut:

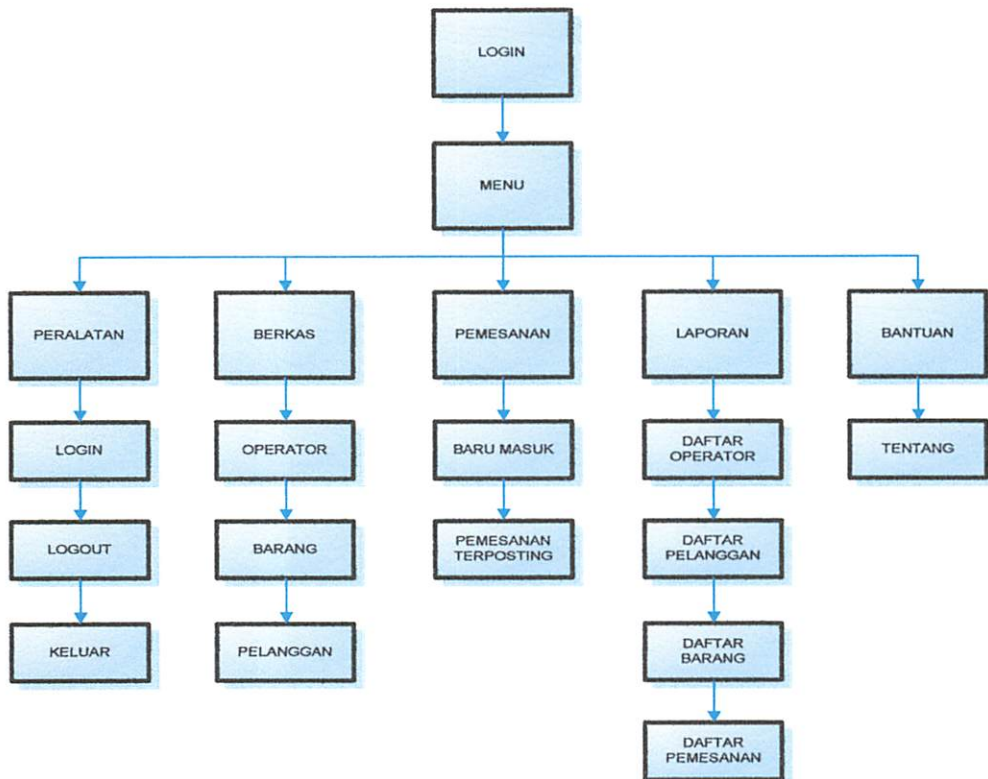
Tabel 3.9 Contoh daftar harga barang khusus *member*

No.	Nama Barang	Kode Barang	Harga (Rp.)
1	Kecap ABC 275ml	BM01	5500
2	Supermi Gokar	BM02	1500
3	Supermi Gobang	BM03	1500
4	Supermi Goso	BM04	1500
5	Aqua Galon	BM05	12000
6	Sirup ABC Jeruk	BM06	13000
7	Gulaku	BM07	6000
8	Close Up 125g	B01	7500
9	Pepsodent 125g	B02	8000
10	Shampo Pcntenc 200ml	B03	13000
11	Rinso 700g	B04	12500
12	Attack 700g	B05	14000
13	Batere Alkaline	BL01	5000
14	Batere ABC	BL02	3000

### 3.5 Perencanaan Menu Sistem Aplikasi Program

#### 3.5.1 Struktur Program

Program aplikasi SMS ini nanti akan dibagi menjadi 2, yaitu program pengolah *database/admin* nya dan SMS *gateway* itu sendiri. Untuk struktur program aplikasi admin dapat dilihat seperti pada gambar di bawah ini.



Gambar 3.18 Struktur program aplikasi admin

#### 3.5.2 Desain Laporan

Desain laporan pada program aplikasi admin ini dibuat dengan bantuan software iReport dari JasperSoft Corp., tampilan desain rancangan laporan seperti gambar-gambar berikut ini.



Daftar Operator					
Kode	Nama Operator	Sex	Alamat	User ID	Password
\$F	\$F	\$F	\$F	\$F	\$F
new Date()			"Page " + \$V{PAGE_NUMBER} + " of "" + \$V		

Gambar 3.19 Rancangan laporan Daftar Operator

Daftar Pelanggan					
Kode	Nama Pelanggan	Alamat	No Telp	No Hp	PIN
\$F	\$F{nm_pel}	\$F{alamat}	\$F{no_hp}	\$F{no_telp}	\$F{pin}
new Date()			"Page " + \$V{PAGE_NUMBER} + " of "" + \$V		

Gambar 3.20 Rancangan laporan Daftar Pelanggan

Daftar Barang		
Kode Barang	Nama Barang	Harga
\$F{k_d_brg}	\$F{nm_brg}	\$F{harga}
new Date()		"Page " + \$V{PAGE_NUMBER} + " of "" + \$V

Gambar 3.21 Rancangan laporan Daftar Barang

Daftar Pemesanan					
No. pesan \${pemesanan_no_pesan}					
Tgl	Kd.Pelanggan		Nama Pelanggan		
Alamat: \${pelanggan_alamat}					
	Kd Brg	Nama Barang	Jmlh	Harga	Sub Total
	\${F}	\${barang_nm_brg}	\${F}	\${F}	\${F {subtotal}}
new Date()			"Page " + \${PAGE_NUMBER} + " of "" + \${V		

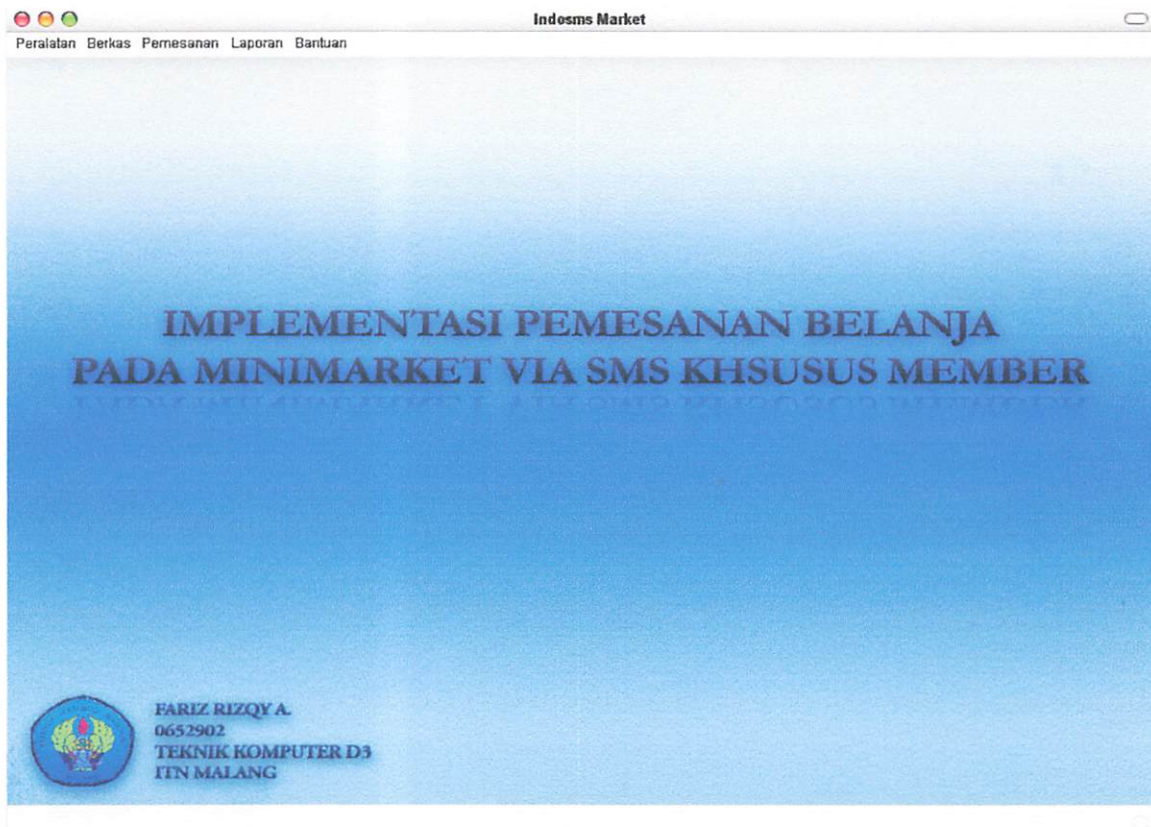
Gambar 3.22 Rancangan laporan Daftar Pemesanan

## BAB IV

# IMPLEMENTASI PEMESANAN BELANJA PADA *MINIMARKET* VIA SMS KHUSUS *MEMBER*

### 4.1 Penjelasan dan Tampilan Program

Dalam pengujian program aplikasi pada Tugas Akhir ini dilakukan dengan tujuan untuk mengetahui kinerja program, berupa kehandalan serta ketepatan analisa antara perencanaan dan hasil akhir yang diperoleh. Seperti pada gambar dibawah yang merupakan hasil dari tampilan menu utama program pengolah *database*.



Gambar 4.1 Menu Utama

#### 4.1.1 Form Login

Menu ini digunakan sebagai jendela antarmuka untuk melakukan *login* bagi pengguna. Tanpa adanya hak akses tersebut, seseorang tidak akan mampu untuk menggunakan aplikasi ini. Dalam setiap aplikasi, hampir dipastikan memiliki *form* ini.



The image shows a standard Windows-style dialog box titled "Login Form". At the top, there is a red close button on the left and a minimize button on the right. The main content area contains the instruction "Silahkan memasukkan login" centered at the top. Below this, there are two labels: "ID User" and "Password". The "ID User" label is positioned to the left of a text input field containing the text "admin". The "Password" label is positioned to the left of a password input field containing six asterisks "\*\*\*\*\*". Below the input fields are two buttons: "OK" and "Batal", both with rounded rectangular shapes. At the bottom of the dialog box, the text "Otentikasi pengguna" is centered.

Gambar 4.2 Form Login

#### 4.1.2 Form Operator

*Form Operator* digunakan untuk manajemen data *operator*. *Operator* adalah user atau pengguna yang sah mengelola program dan setiap pengguna tersebut memiliki *user id* dan *password*. Dengan fasilitas ini, penggantian nama *user* dan *password* bisa dimungkinkan oleh *operator*.

**Operator**

Silahkan Memasukkan Informasi Operator

ID. Operator

Nama

Kelamin

Alamat

ID User

Password

Password lagi

Daftar Operator

ID Opera...	Nama Operator	Kelamin	Alamat	ID User	Password
OP01	Ardiansyah	Pria	Jl. Langsep no 15, Malang	admin	admin
0002	Susi Susanti	Wanita	Jl. Gatot Subroto no.44, Bogor	indah	a
0003	Ali Abadi	Pria	Jl. A.Yani no 44,Surabaya	ali	ali
0004	Sule Praktiew	Pria	Jl. Kartini no 20, Bandung	awas	awas
0005	Joko Wijaya	Pria	Jl. Kuningan no.33, Semarang	joko	joko

Operator adalah pengguna sah dari program ini

Gambar 4.3 Form Operator

#### 4.1.3 Form Data Barang

Form ini digunakan untuk memasukkan data barang yang tersedia. Seperti ditunjukkan pada gambar dibawah

**Data Barang**

Silahkan Memasukkan Informasi Barang

KD Barang

Nama

Harga

Daftar barang

Kd. Brg	Nama barang	harga
BL01	Radio Fm 44	35,000
B1	Batere Alkaline A3	5,000
BL02	Televisi 41 inch	2,300,000
B13	Lampu Phillip 18 W	30,000
B01	Close Up 125g	8,500
B02	Pepsodent 125g	8,000
B03	Shampo pantene 220ml	12,500

Data Barang ini adalah semua barang yang dijual

Gambar 4.4 Form Data Barang

#### 4.1.4 Form Pelanggan

Form ini digunakan untuk menampilkan manajemen data pelanggan, termasuk tambah, edit, dan hapus. Dalam aplikasi ini, setiap orang yang mau memesan barang melalui sms harus menjadi pelanggan dengan cara memberikan data pribadi berupa tanda pengenal setempat atau yang berlaku, untuk dimasukkan sebagai data dan informasi dalam form pelanggan.

The screenshot shows a web application window titled "Pelanggan". The main heading is "Silahkan Memasukkan Informasi Pelanggan". Below this is a form with the following fields: "ID. Suplayer", "Nama", "No Hp", "No Telepon", "Alamat", and "PIN". At the bottom of the form are four buttons: "Baru", "Edit", "Simpan", and "Batal". Below the form is a section titled "Daftar Pelanggan" containing a table with the following data:

ID Pelanggan	Nama	No Telepon	No HP	Alamat	PIN
001	Surya	094374682	08854526...	Jl Kertajaya no.102 Surabaya	1234
002	Anita	98275324	345436456	Jl. Pademangan Barat 22, Jakarta	3333
003	Haris Suganda		08574845...	Jl. Hasanudin 25, Malang	1111
004	Tatang Sutarma		08576604...	Jl. Diponegoro no.55, Bandung	2222

At the bottom of the window, there is a note: "Pelanggan adalah pihak yang berhak untuk melakukan pemesanan via SMS".

Gambar 4.5 Form Pelanggan

#### 4.1.5 Form Pemesanan Baru Masuk

Form ini digunakan untuk melihat daftar pesan yang masuk. Dimana pelanggan yang mengirimkan sms daftar pemesanan, akan ditampung dalam tabel yang terdapat pada form ini. Yang kemudian akan diposting untuk membedakan pesan yang baru masuk tersebut agar data pemesanan dapat segera diproses.

**Pemesanan baru masuk (belum diposting ke penjualan)**

No. Pesan	Tgl. Pesan	No. Pelanggan	Nama Pelanggan	Telepon	Alamat
00017	Feb 28, 2010	003	Haris Suganda		Jl. Hasanudin 25, Malang

Detail Pemesanan

Kode	Nama Barang	Harga	Jumlah	Subtotal
B02	Pepsodent 125g	8,000	1	8,000
BM02	Supermi Gokar	1,500	3	4,500
BM07	Gulaku 500g	7,500	1	7,500

Total Harga

Gambar 4.6 *Form* Pemesanan Baru Masuk

#### 4.1.6 *Form* Pemesanan Terposting

*Form* pemesanan terposting digunakan untuk melihat data sms pesanan dari pelanggan yang sudah diposting sebelumnya.

**Pemesanan terposting**

No. Pesan	Tgl. Pesan	No. Pelanggan	Nama Pelanggan	Telepon	Alamat
00014	Jan 29, 2010	003	Haris Suganda		Jl. Hasanudin 25, Malang
00015	Feb 15, 2010	003	Haris Suganda		Jl. Hasanudin 25, Malang
00017	Feb 28, 2010	003	Haris Suganda		Jl. Hasanudin 25, Malang
00008	Jan 25, 2010	004	Tatang Sutarma		Jl. Diponegoro no.55, Band...
00016	Feb 22, 2010	004	Tatang Sutarma		Jl. Diponegoro no.55, Band...

Detail Pemesanan

Kode	Nama Barang	Harga	Jumlah	Subtotal
B02	Pepsodent 125g	8,000	1	8,000
BM02	Supermi Gokar	1,500	3	4,500
BM07	Gulaku 500g	7,500	1	7,500

Total Harga

Gambar 4.7 *Form* Pemesanan Terposting

#### 4.1.7 Laporan Daftar Operator

Laporan ini akan menampilkan semua *operator* yang menggunakan aplikasi/program



Kode	Nama Operator	Sex	Alamat	User ID	Password
OP01	Ardiansyah	P	Jl. Langsep no 15, Malang	admin	admin
0002	Susi Susanti	W	Jl. Gatot Subroto no 44, Bogor	indah	a
0003	Ali Abadi	P	Jl. A.Yani no 44, Surabaya	ali	ali
0004	Sule Pre-kitiew	P	Jl. Kartini no 20, Bandung	awas	awas
0005	Joko Wijaya	P	Jl. Kuningan no.33, Semarang	joko	joko

Gambar 4.8 Tampilan Daftar Operator

#### 4.1.8 Laporan Daftar Pelanggan

Laporan ini menampilkan daftar pelanggan/*member* yang sudah mendaftar kan diri.





The image shows a screenshot of a JasperViewer application window. The window title is 'JasperViewer'. The report content is titled 'Daftar Pelanggan' and contains a table with the following data:

Kode	Nama Pelanggan	Alamat	No Telp	No Hp	PIN
001	Surya	Jl Kentajaya no.102 Surabaya	094374682	08854526521	1234
002	Anita	Jl. Pademangan Barat 22, Jakarta	98275324	345436456	3333
003	Haris Suganda	Jl. Hasanudin 25, Malang		085748452363	1111
004	Tatang Sutama	Jl. Diponegoro no.55, Bandung		085766041256	2222

Page 1 of 1

Gambar 4.9 Tampilan Daftar Pelanggan

#### 4.1.9 Laporan Daftar Barang

Laporan ini digunakan untuk menampilkan daftar barang yang disediakan dan dapat diberikan pada pelanggan/member yang sudah terdaftar agar tidak perlu untuk mengingat barang-barang yang tersedia.

**Daftar Barang**

Kode Barang	Nama Barang	Harga
B01	Close Up 125g	8500.0
B02	Pepsodent 125g	8000.0
B03	Shampo pantene 220ml	12500.0
B04	Rinso 700g	12400.0
B05	Attack 700g	13600.0
B1	Batere Alkaline A3	5000.0
B13	Lampu Philip 18 W	30000.0
BL01	Radio Fm 44	35000.0
BL02	Televisi 41 inch	2300000.0
BM01	Kecap ABC 275ml	5500.0
BM02	Supermi Gokar	1500.0
BM03	Supermi Gobang	1500.0
BM04	Supermi Goso	1500.0
BM05	Aqua Galon	12000.0
BM06	Sirup ABC Jeruk	13000.0
BM07	Gulaku 500g	7500.0

Page 1 of 1

Gambar 4.10 Tampilan Daftar Barang

#### 4.1.10 Laporan Daftar Pemesanan

Laporan Daftar Pemesanan ini digunakan untuk sebagai tanda bukti pemesanan barang dari pelanggan yang dapat diberikan pada pelanggan saat pengantaran barang.



Gambar 4.11 Tampilan Daftar Pemesanan

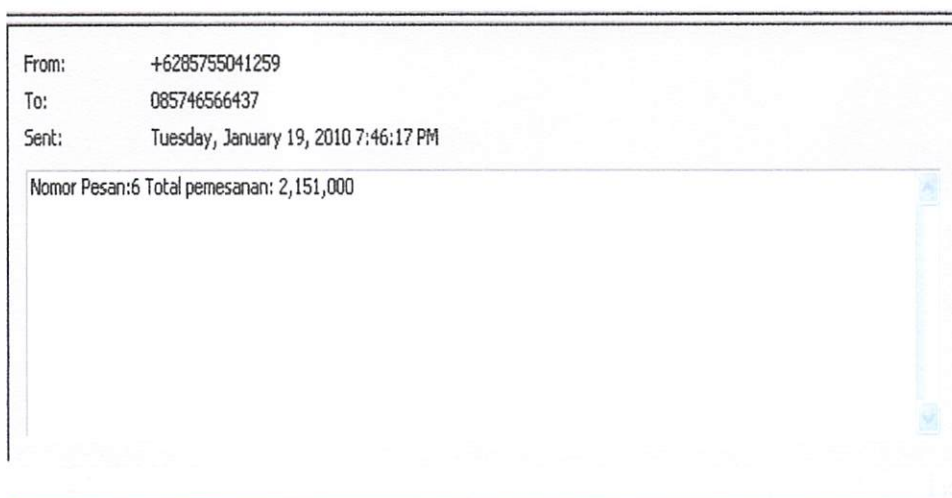
#### 4.1.11 SMS Server

Sebelum menjalankan aplikasi ini pastikan terlebih dahulu untuk memasukkan data-data seperti nama *server*, *database*, *username* dan *password* dari MySQL serta no *port* dari kabel data yang terkoneksi dengan *handphone* dengan benar agar program dapat berjalan dengan baik. Agar program dapat *autoreplay* sebaiknya tidak perlu ditutup sesuai dengan kebutuhan. Setelah semua syarat yang dibutuhkan terpenuhi, maka aplikasi dapat berjalan dengan baik seperti ditunjukkan pada gambar dibawah ini.



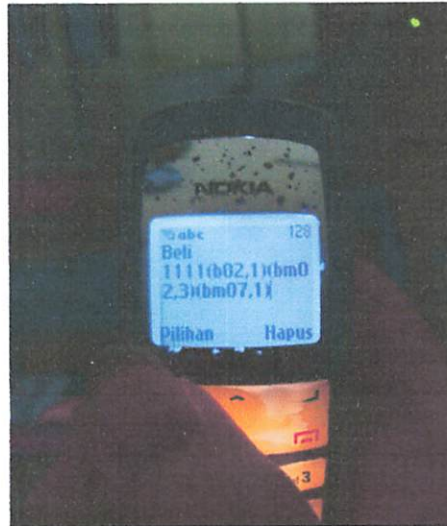
Gambar 4.12 SMS Server

Pada percobaan pengiriman pesan dilakukan dengan menggunakan modem Sierra 881U seperti tampilan gambar 4.13. Jika *member* mengirimkan SMS pemesanan maka akan ada pesan balik yang menyatakan no faktur pemesanan dan jumlah total yang harus dibayarkan pada pemesan.



Gambar 4.13 Balasan dari SMS *Server* pada modem

Pada percobaan pengiriman SMS berikutnya menggunakan HP Nokia 2115i, dan mengirim SMS pemesanan dengan *format* yang sudah ditentukan seperti pada gambar dibawah.



Gambar 4.14 Penulisan *format* pemesanan pada HP

Maka akan mendapatkan balasan no pesan dan jumlah dari total barang yang dipesan seperti pada gambar 4.15



Gambar 4.15 Balasan dari SMS Sever pada HP

## BAB V

### KESIMPULAN DAN SARAN

#### 5.1 Kesimpulan

Setelah melakukan segala rangkaian perencanaan dan pembuatan desain serta pembuatan *script-script* program dengan menggunakan Java dan MySQL, maka dalam laporan Tugas Akhir ini dapat diambil kesimpulan sebagai berikut;

1. Sistem pemesanan belanja yang berbasis SMS ini mempermudah *member* yang ingin berbelanja tanpa harus repot menuju lokasi dan lebih efisien, cepat serta dapat menghemat banyak waktu.
2. Penulisan *format* SMS lebih mudah dan menghemat karakter karena sudah diwakilkan menjadi kode angka yang menghemat penulisan pada karakter SMS.
3. Pembuatan program dengan mengelompokkan menjadi kelas-kelas, memudahkan dalam pengembangan dan pemeliharaan, karena kelas yang telah dibuat dapat digunakan ulang untuk pengembangan selanjutnya.
4. Penggunaan bahasa Java yang berorientasi objek untuk tahap implementasi, memungkinkan aplikasi dapat dijalankan di berbagai *platform*, tanpa perlu dikompilasi ulang.

#### 5.2 Saran

Saran-saran terutama untuk pengembangan sistem belanja *via* SMS ini lebih lanjut adalah sebagai berikut :

1. Dalam perancangan sistem informasi akademik disarankan untuk menggunakan Pendekatan Berorientasi Objek, mulai dari tahap perancangan sampai tahap implementasi, karena dengan pendekatan berorientasi objek memiliki keuntungan, yang membuat sistem akan lebih mudah untuk dikembangkan.
2. Untuk program *database* yang akan dipakai, disarankan untuk menggunakan MySQL.

## Daftar Pustaka

- [1] Sanjaya, Ridwan. (2005). "*Pengolahan Database MySQL 5 dengan Java 2 disertai Teknik Pencetakan Laporan*". Yogyakarta : ANDI.
- [2] Huda, Miftahul, Cuk Subiantoro, dan Bunafit Nugroho. (2008). "*Membuat Aplikasi Penjualan dengan JAVA dan MySQL*". Jakarta : PT Elex Media Komputindo.
- [3] Huda, Miftahul, dan Bunafit Nugroho. (2009). "*Membuat Aplikasi Rental dengan JAVA dan MySQL*". Jakarta : PT Elex Media Komputindo.
- [4] Haryanto, Steven. (2005). "*SQL : Kumpulan Resep Query Menggunakan MySQL*". Jakarta : Dian Rakyat.
- [5] Tomy, SE. (2008). "Tip dan Trik Profesional MySQL 5". Jakarta : PT Elex Media Komputindo.
- [6] Kadir, Abdul. (2008). "*tuntunan Praktis Belajar Database Menggunakan MySQL*". Yogyakarta : ANDI.
- [7] Utama, Ginanjar. (2002). "*Berpikir Objek: Cara Efektif Menguasai Java*". IlmuKomputer.Com.
- [8] JARDIKNAS. "*JENI ( Java Education Network Indonesia )*".  
<http://www.jeni.or.id>
- [9] Indra, S.Kom. "*Dasar-dasar SMS Gateway*". Fakultas Teknologi Informasi :  
Fakultas Budi Luhur



# ***LAMPIRAN***



## LEMBAR ASISTENSI LAPORAN TUGAS AKHIR

Nama : FARIZ RIZQY ARDIANSYAH  
NIM : 0652902  
Jurusan : TEKNIK ELEKTRO D-III  
Konsentrasi : TEKNIK KOMPUTER  
Dosen Pembimbing : IR. YUSUF ISMAIL NAKHODA, MT  
Judul Tugas Akhir : IMPLEMENTASI PEMESANAN BELANJA PADA  
MINIMARKET VIA SMS KHUSUS MEMBER

NO	TANGGAL	KETERANGAN	PARAF
1	21-01-1010	Split format SMS dengan Java	
2	28-01-2010	Penjumlahan MySQL dengan iReport	
3	01-02-2010	Gambar pada main menu yang bersifat dinamis	
4	09-02-2010	Lengkapi kesimpulan pada Bab V	
5	15-02-2010	Huruf bahasa Inggris di cetak miring	
6	16-02-2010	Nomor halaman di tengah bawah	
7	19-02-2010	Pemberian judul dan nama pada abstrak	

Mengetahui,

Dosen Pembimbing

(Ir. Yusuf Ismail Nakhoda, MT)  
NIP.Y.1018800189



**BERITA ACARA UJIAN TUGAS AKHIR  
PROGRAM STUDI TEKNIK ELEKTRO D-III  
FAKULTAS TEKNOLOGI INDUSTRI**

Nama Mahasiswa : Fariz Rizqy Ardiansyah  
Nim : 0652902  
Program Studi : Teknik Elektro D-III  
Konsentrasi : Teknik Komputer  
Judul Tugas Akhir : Implementasi Pemesanan Belanja Pada Minimarket  
Via SMS Khusus Member

Dipertahankan di hadapan Tim Penguji Tugas Akhir Jenjang Program Diploma Tiga (D-III) pada:

Hari, Tanggal : Selasa, 23 Februari 2010  
Dengan Nilai : 85,50 (A)

*Panitia ujian Tugas Akhir*



Ir. Sidik Noertjahjono, MT  
NIP. Y 102 8700 163

Sekretaris

Ir. H. Taufik Hidayat, MT  
NIP. Y 101 8700 151

Anggota Penguji I

Joseph Dedy Irawan, ST., MT  
NIP. 132 3151 78

Anggota Penguji II

Sotyohadi, ST  
NIP. Y 103 9700 309