BAB III
PEMBAHASAN

3.1 Jadwal Kerja Praktek

Jadwal kerja praktek dilaksanakan di PT. Citra Niaga Bandung. Waktu dan penempatan kerja praktek adalah sebagai berikut :

Tempat : PT. Citra Niaga Technology.

Waktu : 05 Juni – 27 Agustus 2010

Jam : 07.00 – 16.00

Adapun kegiatan selama kerja praktek sebagai berikut :

Tabel 3.1 Kegiatan Kerja Praktek

Tanggal	Kegiatan
6-11 Juni 2010	Analisa Kebutuhan
16-18 Juni 2010	Desain Sistem
23 Juni -30 Juli 2010	Penulisan Kode Program
13 Juli 2010	Pengujian Program
20- 27 Agustus 2010	Penerapan Program

3.2 Data Kerja Praktek

Data kerja praktek sangat dibutuhkan untuk menunjang pembangunan aplikasi, data yang diperoleh oleh kami diambil melalui metode penelitan secara kualitatif dan wawancara terhadap *user* guna mengetahui kebutuhan yang diperlukan oleh perusahaan, adapun data yang kami dapatkan setelah kami melakukan penelitian diantaranya:

1. Data-data yang berkaitan dengan *Purchase Order.*

2. Data profil perusahaan yang berupa sejarah instansi, struktur *job* dan *description.*

3.3 Hasil Kerja Praktek

Hasil kerja prakek selama kami melakukan praktek diantaranya mencakup Perancangan Proses, Perancangan Basis Data, Perancangan Program, Kebutuhan Sistem, dan Implementasi.

3.3.1 Analisis Masalah

Sistem pembelian pemesanan yang terdapat di PT. Citra Niaga Technology (CNT) Bandung saat ini masih memiliki suatu kekurangan, yaitu masih menggunakan sistem aplikasi yang berbasis pada *desktop* padahal sebaiknya sistem pembelian pemesanan tersedia dalam bentuk web sehingga metode pembayaran dan proses pencetakan bukti pembelian dapat secara secara cepat dan mudah. Adapun masalah yang sering muncul dalam sistem aplikasi pembelian pemesanan yang sudah berjalan adalah sebagai berikut :

a. Konsumen kesulitan dalam melakukan proses pemesanan dan proses pembayaran karena konsumen melakukan proses tersebut hanya melalui email maupun telepon perusahaan.

b. Perusahaan melakukan pendataan konsumen berdasarkan permintaan melalui email dan telepon sehingga proses tersebut kurang efektif.

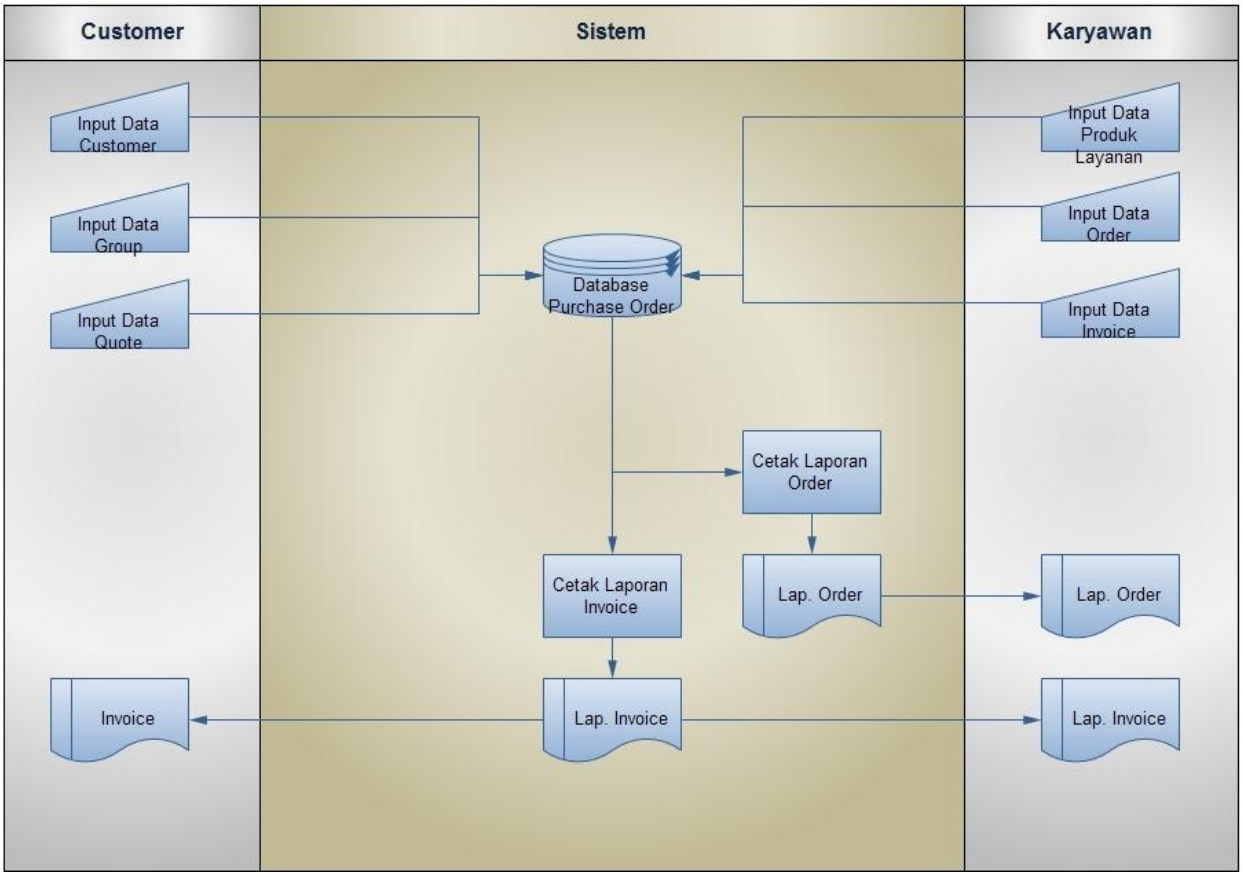
3.3.2 Analisis Prosedur Yang Sedang Berjalan

Adapun prosedur yang sedang berjalan adalah sebagai berikut :

a. *Customer* mengisi *form customer* serta melengkapi syarat-syarat untuk pembelian yang kemudian di*input*kan ke dalam *database*.
b. Karyawan menginputkan data produk layanan ke dalam *database*.
c. Setelah mengisi data *customer*, *customer* dapat mengisi data *quote* yang sebagian datanya diambil dari *database customer*.
d. Data *quote* yang diinputkan *customer* di simpan dalam *database quote*.
e. *Customer* mengisi data *order* yang sebagian datanya diambil dari *database quote*, yang kemudian di*input*kan ke dalam *database*.
f. Data *order* yang telah di*input*kan oleh *customer*, sistem akan mengolah *input* data *invoice* yang kemudian data *invoice* tersebut disimpan didalam *database invoice*.

g. *Detail invoice* yang telah disimpan dalam database *invoice*, dapat langsung dicetak oleh *customer*.
h. *Sistem* mencetak laporan *order* dan laporan *invoice* dari *database order* dan *database invoice* untuk data *internal* perusahaan.

Adapun bentuk dari alur dokumen seperti pada gambar 3.1.



Gambar 3.1 Alur Dokumen

3.3.3 Analisis Kebutuhan Non Fungsional

Analisis kebutuhan non fungsional dilakukan untuk menghasilkan spesifikasi kebutuhan non fungsional. Analisis kebutuhan ini diperlukan untuk menentukan keluaran yang akan dihasilkan sistem, masukan yang diperlukan sistem, lingkup proses yang digunakan untuk

mengolah masukan menjadi keluaran, volume data yang akan ditangani sistem, jumlah pemakai dan kategori pemakai, serta kontrol terhadap sistem.

3.3.3.1 Analisis Perangkat Keras

Adapun spesifikasi komputer pada PT. Citra Niaga Technology (CNT) Bandung adalah sebagai berikut :

1. Processor : Intel Dual Core
2. Memory : RAM DDR 512 MB
3. Harddisk : 120 GB
4. VGA : 128 MB
5. Monitor : 15” LCD
6. Keyboard dan Mouse : Standar
7. Switch : Linksys SRW2024 24 Port, 10/100/1000 Mbps Gigabit
8. Kabel : UTP Balden UTP cate 5e
9. Konektor : RJ45
10. Modem : ADSL D-Link DSL520T 1 port UTP 10/100 Mbps Modem

Perangkat keras yang telah dimiliki PT. Citra Niaga Technology (CNT) saat ini, sudah memenuhi standar kebutuhan minimum spesifikasi perangkat keras yang dibutuhkan untuk membangun sistem aplikasi *Purchase Order* yang berbasis web.

3.3.3.2 Analisis Perangkat Lunak

Di dalam pelaksanaan sistem baru selain perangkat keras (*Hardware*) dibutuhkan juga perangkat lunak (*Software*), Dimana perangkat lunak merupakan alat pendukung sistem yang terdiri dari sistem operasi dan aplikasi database. Adapun Perangkat lunak yang digunakan PT. Citra Niaga Technology (CNT) Bandung saat ini adalah sebagai berikut:

a. Sistem operasi : Microsoft Windows XP

b. Software lainnya : Microsoft Office 2007

Spesifikasi perangkat lunak yang dibutuhkan untuk mendukung aplikasi yang akan dibangun adalah sebagai berikut:

a. SistemOperasi : Window XP service pack 2

b. Database : MySQL
c. Bahasa Pemrograman : php
d. Tool yang digunakan : Macromedia Dreamweaver

3.3.3.3 Analisis User

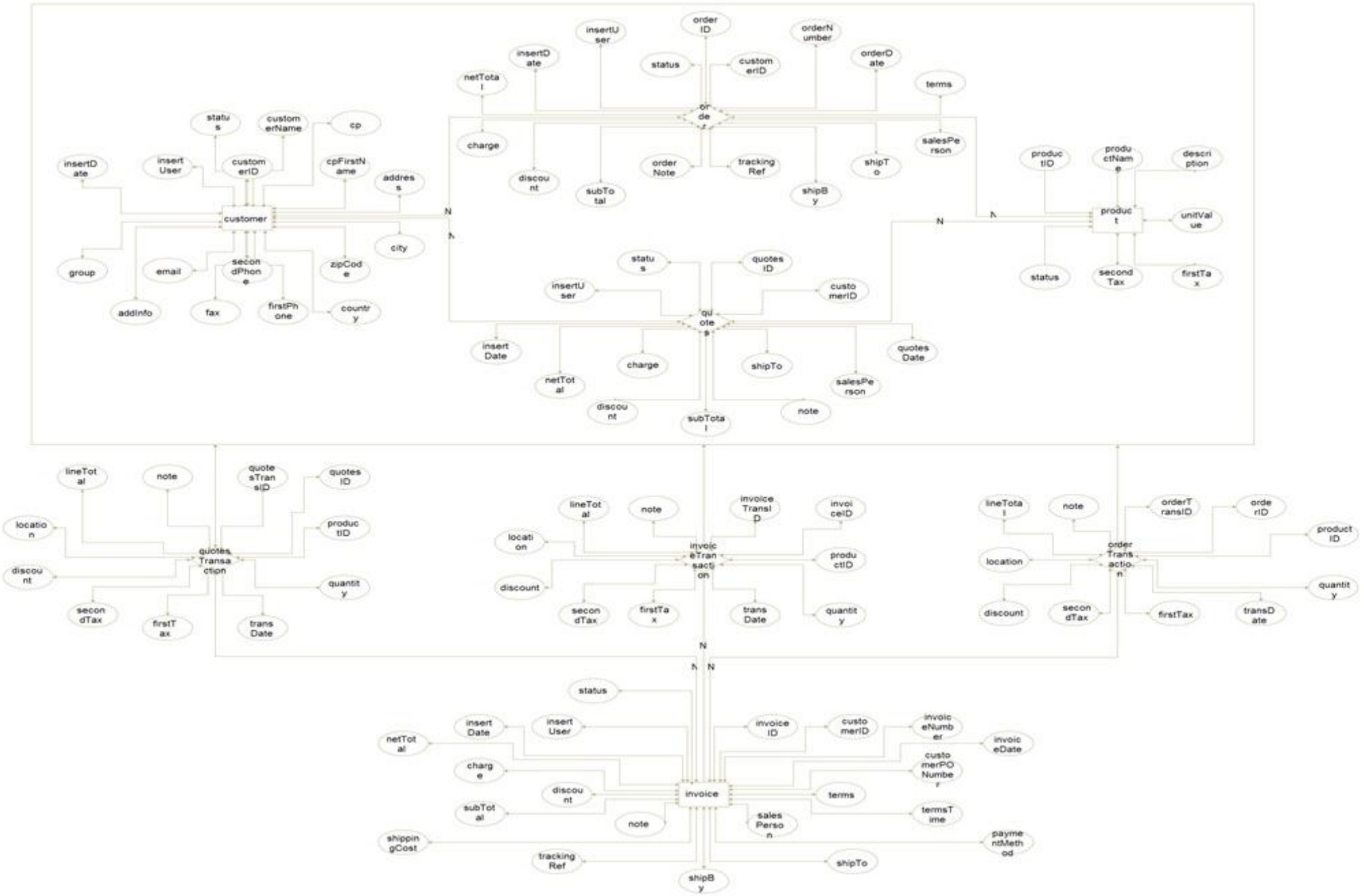
Pengguna utama sistem aplikasi ini adalah para karyawan dan konsumen PT. Citra Niaga Technology (CNT) Bandung yang tingkat kemampuan dalam menggunakan sistem aplikasi ini sangat bervariasi sehingga rancangan antarmuka pemakai harus sederhana dan mudah digunakan, adalah sebagai berikut :

1. Aplikasi yang akan dibangun berjalan dilingkungan sistem operasi.
2. Tampilan dari sistem aplikasi akan dibangun dalam modus web yang sangat *user friendly*.
3. Diharapkan user dapat menguasai komputer.
4. Navigasi menu dan pengimputan teks hanya dapat dilakukan melalui *keyboard* dan *mouse*.
5. Penggunaan tombol diharapkan dapat mempermudah pengaksesan maupun penggunaan aplikasi ini, sehingga semua informasi yang diperlukan dapat terpenuhi.

3.4 Analisis Basis Data

Dalam memodelkan data dan menggambarkan hubungan antara data yang ada pada sistem yang sedang berjalan digunakan alat bantu yaitu *Entity Relationship Diagram* (ERD).

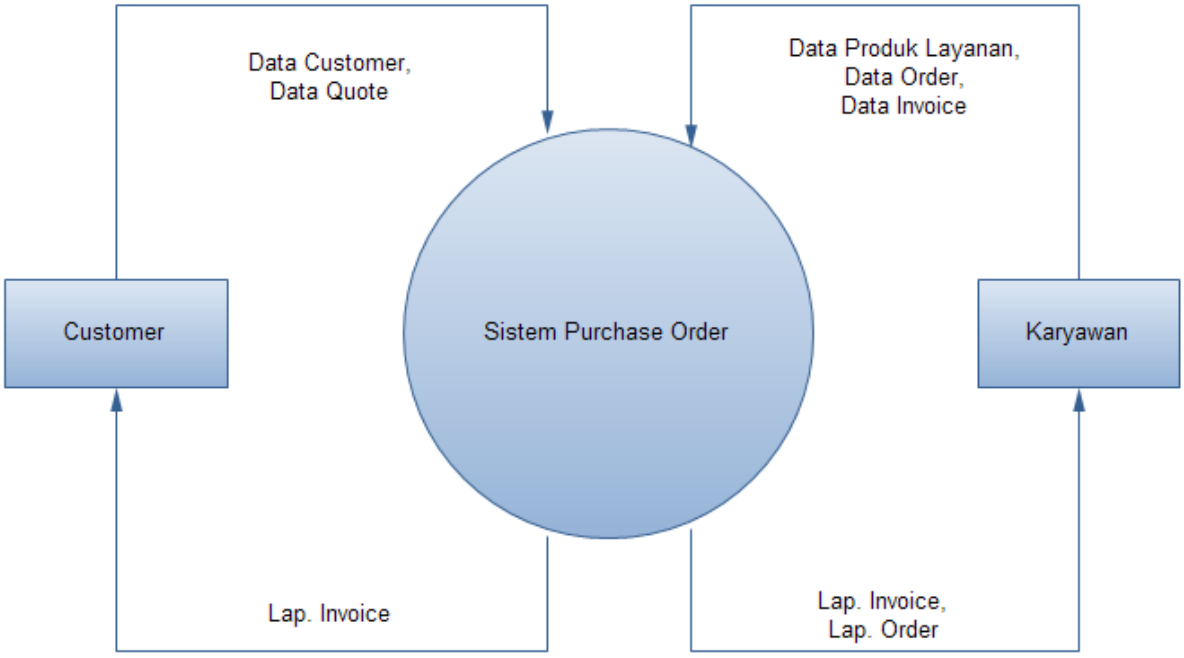
Entity Relation Diagram (ERD) merupakan diagram yang digunakan untuk merancang suatu sistem basis data dan memperlihatkan hubungan atau relasi antara *entity*/objek yang terlibat. Untuk lebih jelasnya dapat dilihat seperti pada gambar dibawah ini.



Gambar 3.2 Entity Relation Diagram

3.4.1 Diagram Konteks

Diagrarn Konteks merupakan gambaran secara umum mengenai sebuah sistem yang dirancang secara global, yaitu suatu diagram yang mempresentasikan atau menggambarkan hubungan antara sistem dengan lingkungan luar sistem yang mempengaruhi operasi sistem. Sistem ditunjukkan dalam satu lingkaran yang menggambarkan keseluruhan proses dalam sistem dan hubungannya dengan entitas lain. Bentuk Diagram Konteks aplikasi sistem *purchase order* adalah sebagai berikut :

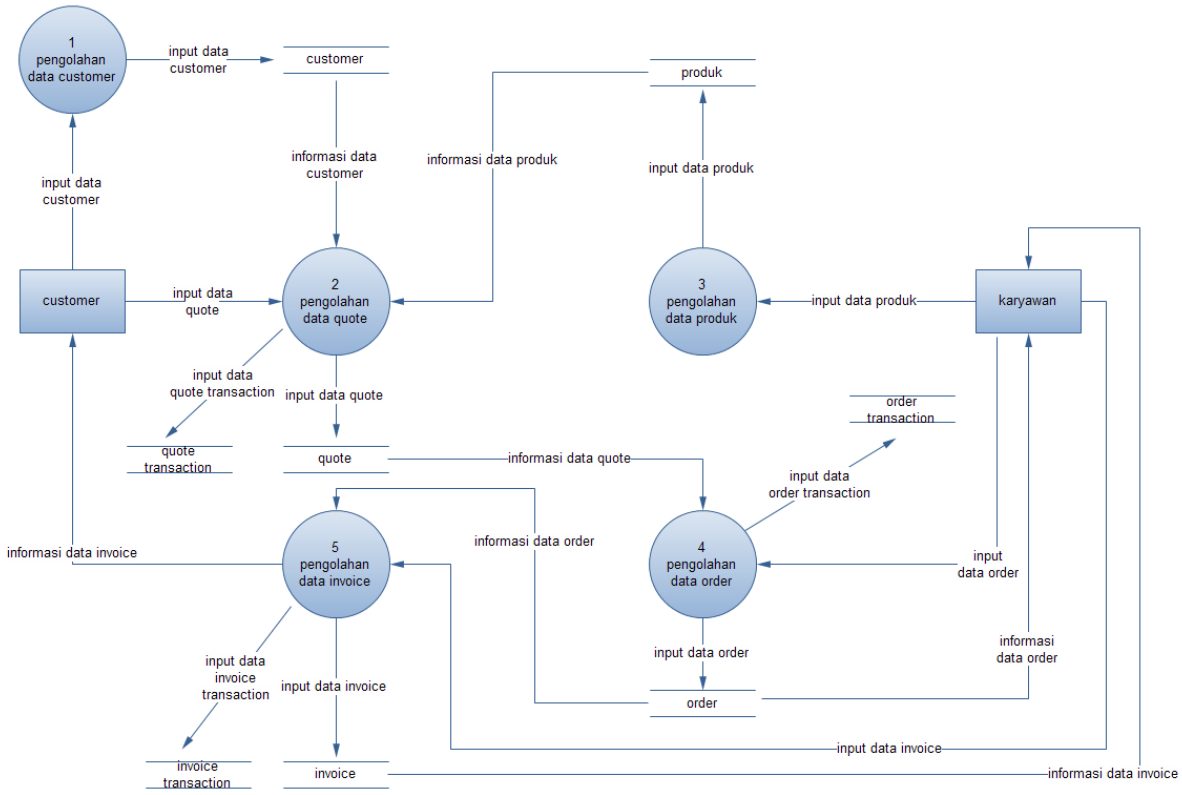


Gambar 3.3 Diagram Konteks

3.4.2 Data Flow Diagram

Penggambaran dalam diagram ini menekankan pada kerja atau proses apa saja yang terjadi mulai dari input data *customer*, membuat antrian, melakukan pemesanan barang, mencetak *invoice* untuk *customer* itu sendiri maupun untuk perusahaan, dan karyawan dapat menginputkan data layanan yang disajikan secara mendetail serta menjelaskan *input* dan *output* dari suatu proses penggambaran sistem.

3.4.2.1 DFD Level 0



Gambar 3.4 *Data Flow Diagram Purchase Order* Level 0

1. Proses Tambah Data *Customer*

   Proses ini adalah proses penambahan data *customer* untuk masuk kedalam proses antrian *quote*.

2. Proses Pengolahan Data *Quote*

   Proses ini adalah proses yang di lakukan *customer* untuk menambah data antrian *quote*.

3. Proses Tambah Data Produk

   Proses ini adalah proses yang di lakukan karyawan untuk menambah data produk.

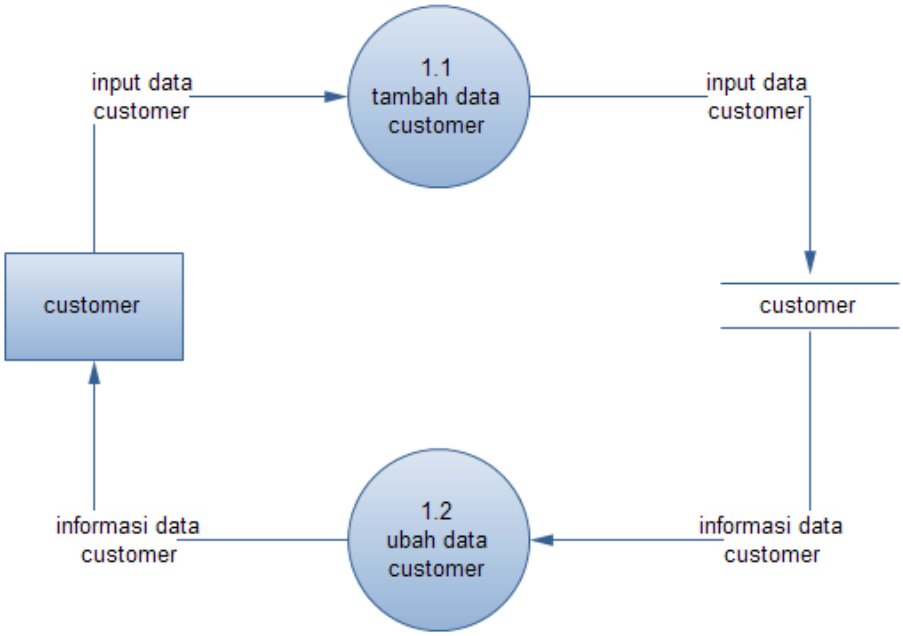
4. Proses Pengolahan Data *Order*

   Proses ini adalah proses yang di lakukan karyawan untuk mengolah data pemesanan.

5. Proses Pengolahan Data *Invoice*

   Proses ini adalah proses yang di lakukan karyawan untuk membuat *invoice*.

3.4.2.2 DFD Level 1 Proses Tambah Data *Customer*



Gambar 3.5 *Data Flow Diagram* Proses Tambah Data *Customer* Level 1

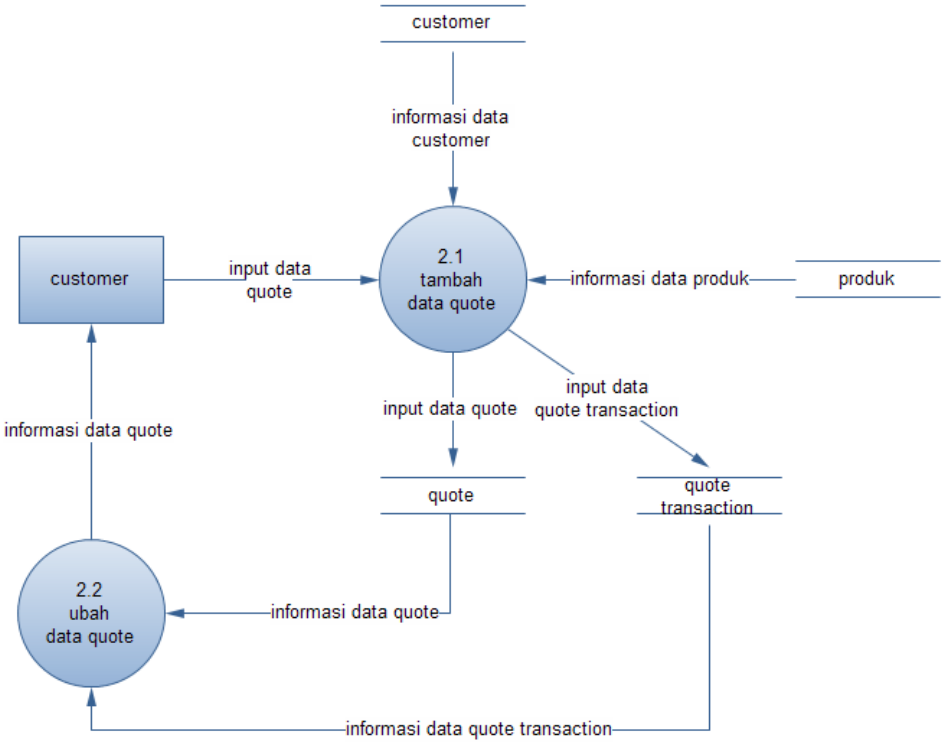
1. Proses Tambah *Customer*

   Proses ini adalah untuk menambah data *customer*.

2. Proses Ubah *Customer*

   Proses ini adalah untuk mengedit data *customer*.

3.4.2.3 DFD Level 1 Proses Pengolahan Data *Quote*



Gambar 3.6 *Data Flow Diagram* Proses Pengolahan Data *Quote* Level 1

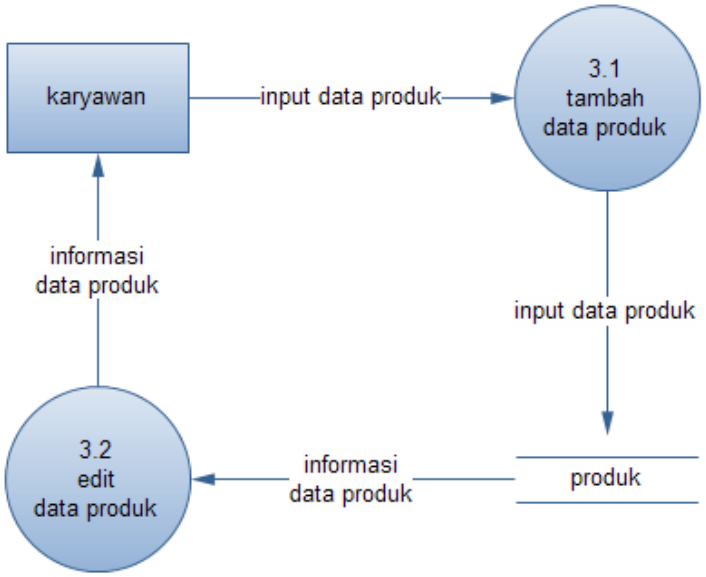
1. Proses Tambah *Quote*

   Proses ini adalah untuk menambah data *quote*.

2. Proses Ubah *Quote*

   Proses ini adalah untuk mengedit data *quote*.

3.4.2.4 DFD Level 1 Proses Tambah Data Produk



Gambar 3.7 *Data Flow Diagram* Proses Tambah Data Produk Level 1

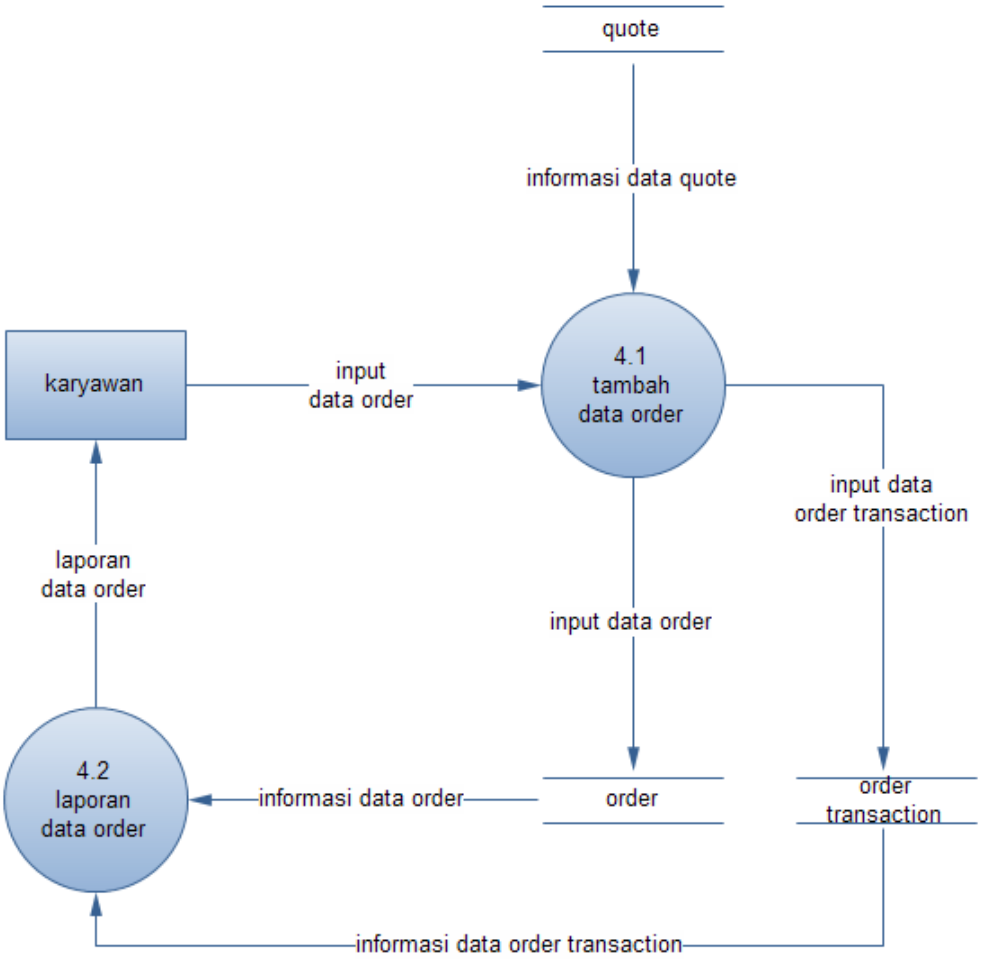
1. Proses Tambah Produk

   Proses ini adalah untuk menambah data produk.

2. Proses Ubah Produk

   Proses ini adalah untuk mengubah data produk.

3.4.2.5 DFD Level 1 Proses Pengolahan Data *Order*



Gambar 3.8 *Data Flow Diagram* Proses Pengolahan Data *Order* Level 1

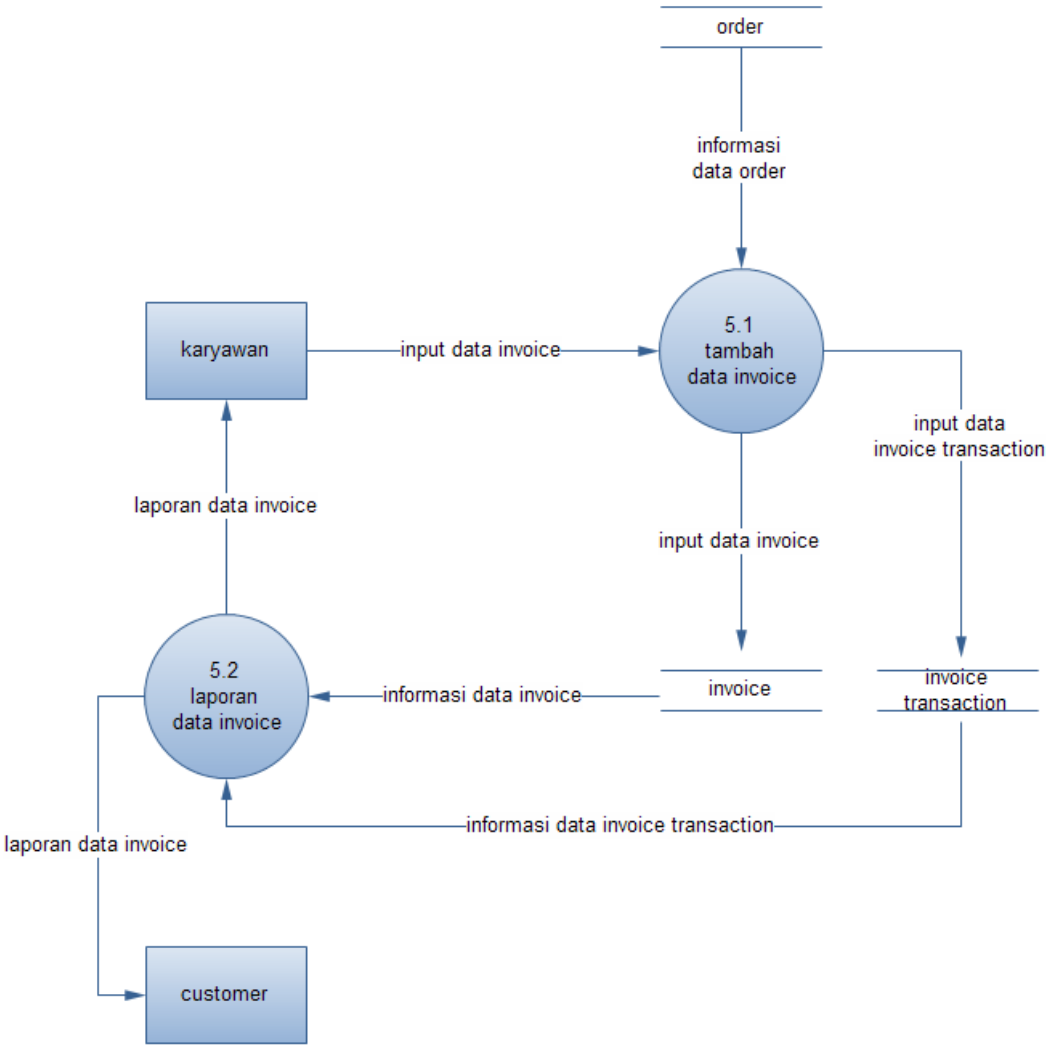
1. Proses Tambah *Order*

   Proses ini adalah untuk menambah data *order*.

2. Proses Laporan *Order*

   Proses ini adalah untuk membuat laporan *order*.

3.4.2.6 DFD Level 1 Proses Pengolahan Data *Invoice*



Gambar 3.9 *Data Flow Diagram* Proses Pengolahan Data *Invoice* Level 1

1. Proses Tambah *Invoice*

   Proses ini adalah untuk menambah data *invoice*.

2. Proses Laporan *Invoice*

   Proses ini adalah untuk membuat laporan *invoice* kepada karyawan dan *customer*.

3.4.3 Kamus Data

kamus data berfungsi membantu pelaku sistem untuk mengerti aplikasi secara detail dan mengorganisasi semua elemen data yang digunakan dalam sistem sehingga pemakai dan penganalisa sistem mempunyai dasar pengertian yang sama tentang masukan, keluaran, penyimpanan dan proses yang ada pada aplikasi sistem.

Kamus data digunakan untuk menjelaskan atau menguraikan arti aliran data dan penyimpanan data dalam DFD. Kamus data dibuat berdasarkan aliran data yang ada dalam Data Flow Diagram. Dibawah ini adalah kamus data dari masing-masing data :

Kamus Data :

Tabel 3.2 Kamus Data dari Data *Customer*

Nama arus data	:	Data Customer
Alias	:	Identitas Customer
Aliran Data	:	Customer – Proses 1.0 Proses 1.0 – F. Data Customer
Atribut	:	customerID + customerName + cp + cpFirstName + address + city + zipCode + country + firstPhone + secondPhone + fax + email + addInfo + group + insertDate + insertUser + status + groupID + groupName.

Tabel 3.3 Kamus Data dari Data *Quote*

Nama arus data	:	Data Quote
Alias	:	Identitas Quote
Aliran Data	:	Customer – Proses 3.0 Proses 3.0 – F. Data Quote

Atribut	:	quotesID + customerID + quotesDate + salesperson + shipTo + note + subtotal + discount + charge + netTotal + insertDate + insertUser + status + quotesTransID + productID + quantity + transDate + firstTax + secondTax + discount + location + lineTotal.

Tabel 3.4 Kamus Data dari Data *Order*

Nama arus data	:	Data Order
Alias	:	Identitas Order
Aliran Data	:	Customer – Proses 4.0 Proses 4.0 – F. Data Order
Atribut	:	orderID + customerID + orderNumber + orderDate + terms + salesperson + shipTo + shipBy + trackingRef + orderNote + subtotal + discount + charge + netTotal + insertDate + insertUser + status + orderTransID + productID + quantity + transDate + firstTax + secondTax + location + lineTotal + note.

Tabel 3.5 Kamus Data dari Data *Invoice*

Nama arus data	:	Data Invoice
Alias	:	Identitas Invoice
Aliran Data	:	F. Data Order – Proses 5.0 Proses 5.0 – F. Data Invoice F. Data Invoice – Proses 6.0 Proses 6.0 – Customer

Atribut	:	invoiceID + customerID + invoiceNumber + invoiceDate + customerPONumber + terms + termsTime + paymentMethod + salesperson + shipTo + shipBy + trackingRef + shippingCost + note + subtotal + discount + charge + netTotal + insertDate + insertUser + status + invoiceTransID + productID + quantity + transDate + firstTax + secondTax + location + lineTotal + paymentID + amount + method + referenceNumber.

Tabel 3.6 Kamus Data dari Data Produk Layanan

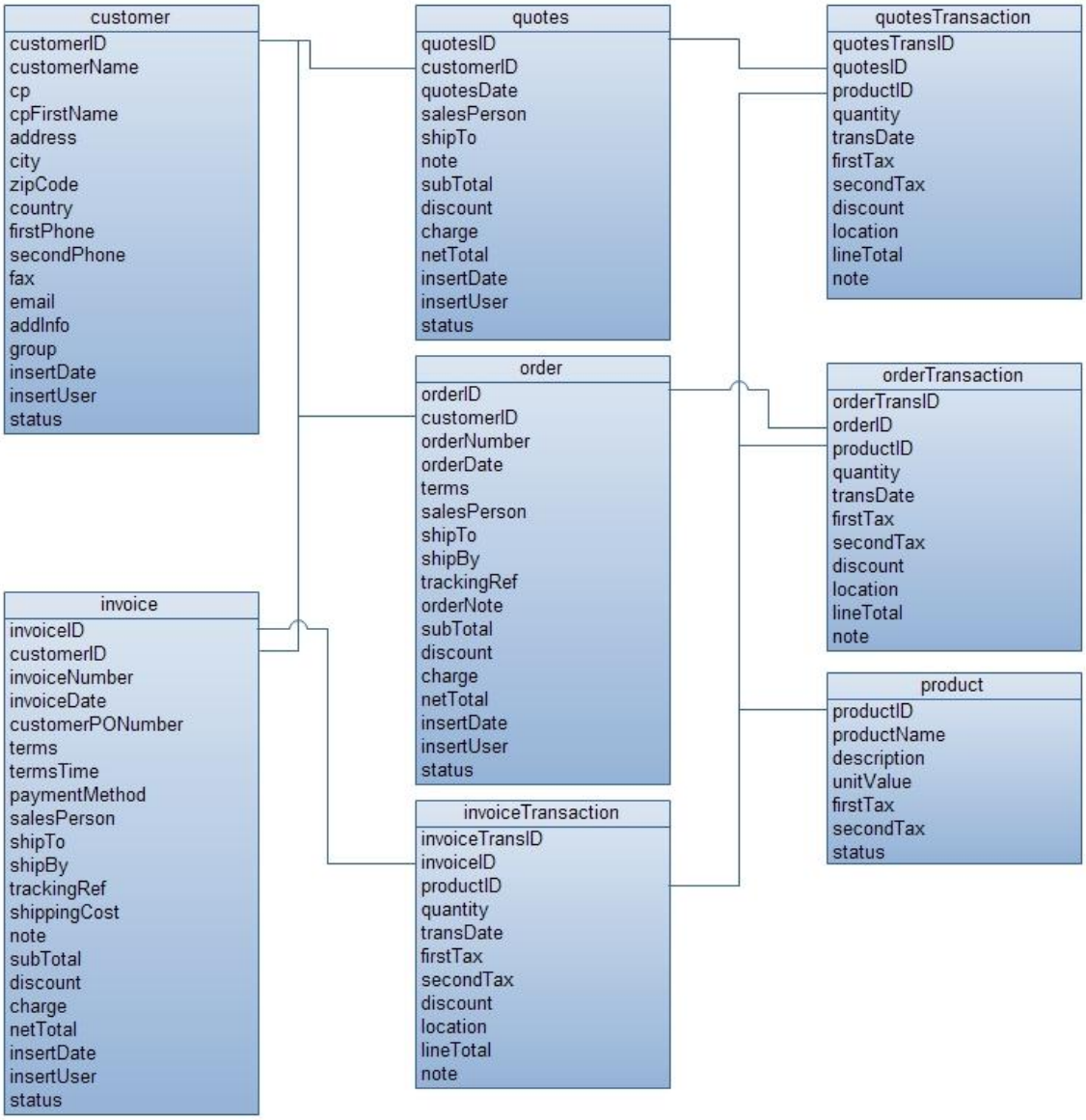
Nama arus data	:	Data Produk Layanan
Alias	:	Identitas Produk Layanan
Aliran Data	:	Karyawan – Proses 2.0 Proses 2.0 – F. Data Produk
Atribut	:	productID + productName + description + unitValue + firstTax + secondTax + status.

3.5 Perancangan Basis Data

Kemampuan untuk mengatur atau mengolah sejumlah data dan kecepatan untuk mencari informasi yang relevan adalah sangat penting bagi suatu organisasi. Untuk mendapatkan himpunan data yang besar dan kompleks harus memiliki alat (*tools*) yang akan menyederhanakan tugas manajemen data dan mengekstra informasi yang berguna secara tepat waktu. Basis data (database) merupakan himpulan dari file yang saling berkaitan satu dengan yang lainnya.

3.5.1 Tabel Relasi

Relasi table menggambarkan hubungan antara table-tabel yang ada pada suatu sistem pengolahan data. Model basis data *relation* sering disebut sebagai model *relation* atau basis data relasional menunjukan suatu cara mekanisme yang digunakan untuk mengelola atau mengorganisasi data secara fisik :



Gambar 3.10 Relasi Tabel

3.5.2 Struktur File

Struktur File merupakan suatu kumpulan data yang saling terkait dan berhubungan satu dengan lainnya. Dalam pemrograman diberikan data-data yang akan diolah. Data-data memberikan suatu tempat penyimpanan yang disebut database. Adapun struktur file database yang digunakan adalah seperti yang digambarkan dibawah ini.

Tabel 3.7 *Customer*

Nama Tabel : Tabel customer Media Penyimpanan : Primary Key : customerID				
No	**Variabel Name**	**Type**	**Field Size**	**Keterangan**
1	customerID	varchar	10	
2	customerName	varchar	50	
3	Cp	varchar	50	
4	cpFirstName	varchar	20	
5	Address	varchar	200	
6	City	varchar	20	
7	zipCode	varchar	10	
8	Country	varchar	3	
9	firstPhone	varchar	15	
10	secondPhone	varchar	15	
11	Fax	varchar	15	
12	Email	varchar	50	
13	addInfo	varchar	100	
14	Group	varchar	2	
15	insertDate	datetime		
16	insertUser	varchar	30	
17	Status	enum	y','n'	

Tabel 3.8 *Product*

Nama Tabel : Tabel product Media Penyimpanan : Primary Key : productID				
No	**Variabel Name**	**Type**	**Field Size**	**Keterangan**
1	productID	varchar	10	
2	productName	varchar	30	
3	Description	text		
4	unitValue	double(20,0)		
5	firstTax	double(20,0)		
6	secondTax	double(20,0)		
7	Status	enum	y','n'	

Tabel 3.9 *Quote*

Nama Tabel : Tabel quotes Media Penyimpanan : Primary Key : quotesID				
No	**Variabel Name**	**Type**	**Field Size**	**Keterangan**
1	quotesID	varchar	10	
2	customerID	varchar	10	
3	quotesDate	date		
4	salesperson	varchar	50	
5	shipTo	varchar	200	
6	Note	text		
7	subtotal	double(20,0)		
8	Discount	double(20,0)		
9	Charge	double(20,0)		
10	netTotal	double(20,0)		

11	insertDate	datetime		
12	insertUser	varchar	30	
13	Status	enum	'open','ordered'	

Tabel 3.10 *Quote Transaction*

Nama Tabel : Tabel quotesTransaction Media Penyimpanan : Primary Key : quotesTransID				
No	**Variabel Name**	**Type**	**Field Size**	**Keterangan**
1	quotesTransID	varchar	10	
2	quotesID	varchar	10	
3	productID	varchar	10	
4	Quantity	int		
5	transDate	date		
6	firstTax	double(20,0)		
7	secondTax	double(20,0)		
8	Discount	double(20,0)		
9	Location	varchar	50	
10	lineTotal	double(20,0)		
11	Note	varchar	100	

Tabel 3.11 *Order*

Nama Tabel : Tabel order Media Penyimpanan : Primary Key : ordered				
No	**Variabel Name**	**Type**	**Field Size**	**Keterangan**
1	orderID	varchar	10	
2	customerID	varchar	10	
3	orderNumber	varchar	30	
4	orderDate	date		
5	Terms	varchar	15	
7	salesperson	varchar	50	
8	shipTo	varchar	200	
9	shipBy	varchar	20	
10	trackingRef	varchar	30	
11	orderNote	text		
12	subtotal	double(20,0)		
13	Discount	double(20,0)		
14	Charge	double(20,0)		
15	netTotal	double(20,0)		
16	insertDate	datetime		
17	insertUser	varchar	30	
18	Status	enum	'open','invoiced'	

Tabel 3.12 ***Order Transaction***

Nama Tabel : Tabel orderTransaction Media Penyimpanan : Primary Key : orderTransID				
No	**Variabel Name**	**Type**	**Field Size**	**Keterangan**
1	orderTransID	varchar	10	
2	orderID	varchar	10	
3	productID	varchar	10	
4	Quantity	int		
5	transDate	date		
6	firstTax	double(20,0)		
7	secondTax	double(20,0)		
8	Discount	double(20,0)		
9	Location	varchar	50	
10	lineTotal	double(20,0)		
11	Note	varchar	100	

Tabel 3.13 ***Invoice***

Nama Tabel : Tabel invoice Media Penyimpanan : Primary Key : invoiceID				
No	**Variabel Name**	**Type**	**Field Size**	**Keterangan**
1	invoiceID	varchar	10	
2	customerID	varchar	10	
3	invoiceNumber	varchar	30	
4	invoiceDate	date		
5	customerPONumber	varchar	50	
6	Terms	varchar	15	

7	termsTime	int		
8	paymentMethod	varchar	20	
9	salesperson	varchar	50	
10	shipTo	varchar	200	
11	shipBy	varchar	20	
12	trackingRef	varchar	30	
13	shippingCost	double(20,0)		
14	Note	text		
15	subtotal	double(20,0)		
16	Discount	double(20,0)		
18	Charge	double(20,0)		
19	netTotal	double(20,0)		
20	insertDate	datetime		
21	insertUser	varchar	30	
22	Status	enum	'open','paid'	

Tabel 3.14 ***Invoice Transaction***

Nama Tabel : Tabel invoiceTransaction

Media Penyimpanan :

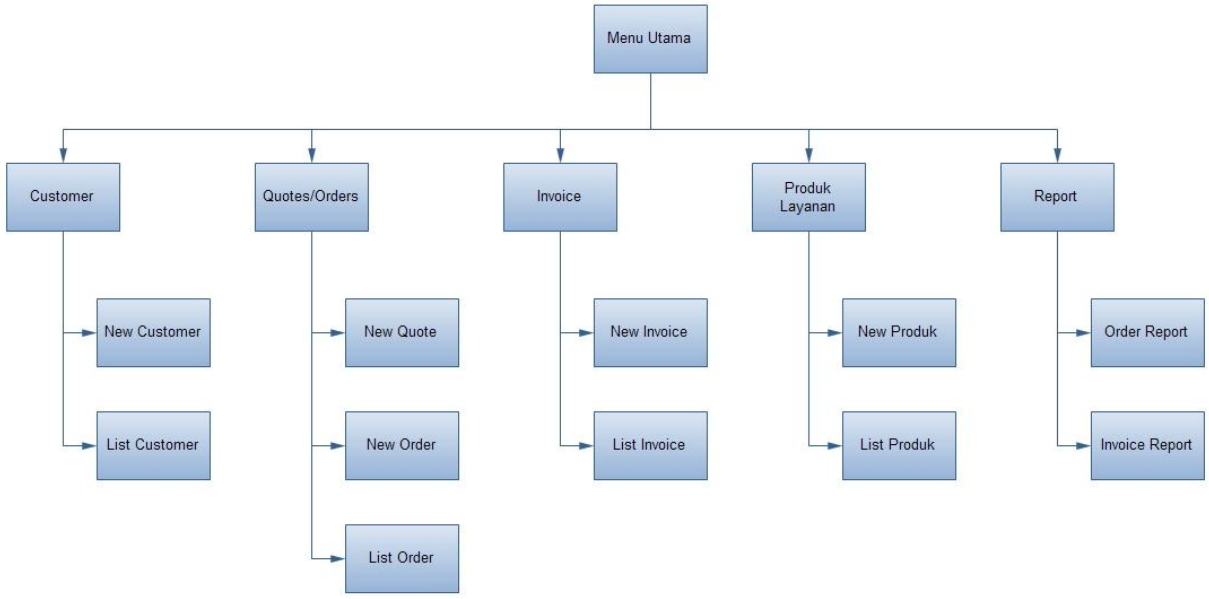
Primary Key : invoiceTransID

No	Variabel Name	Type	Field Size	Keterangan
1	invoiceTransID	varchar	10	
2	invoiceID	varchar	10	
3	productID	varchar	10	
4	Quantity	int		
5	transDate	date		
6	firstTax	double(20,0)		
7	secondTax	double(20,0)		
8	Discount	double(20,0)		

9	Location	varchar	50	
10	lineTotal	double(20,0)		
11	Note	varchar	100	

3.6 Strukfur Menu Program

Perancangan dari bentuk suuktur menu program yang diusulkan bertujuan di dalam penyusunan pada suatu program sehingga program tersebut tersusun dengan benar dan dapat digunakan dengan mudah oleh penggunanya. Adapun bentuk dari struktur menu yang digunakan dapat dilihat pada gambar barikut.



Gambar 3.11 Strukfur Menu Program

3.7 Kebutuhan Sistem

Dalam pembuatan suatu sistem informasi berupa program aplikasi yang akan diterapkan dibutuhkan perangkat keras dan perangkat lunak yang menunjang kerja sistem agar program aplikasi yang dihasilkan sesuai dengan yang diharapkan. Adapun kebutuhan perangkat keras dan perangkat lunak yang digunakan dalam pengembangan aplikasi sistem *Purchase Order* di PT. Citra Niaga Technology adalah sebagai berikut :

a) Kebutuhan perangkat keras (*Hardware*)

Untuk mendukung kelancaran sistem yang dibuat, diperlukan satu keras (*Hardware*) sebagai sarana pendukung. Adapun perangkat keras minimal yang diusulkan adalah sebagai berikut :

a. *Processor* : Intel Pentium IV 1.6 Ghz
b. *Memory* : 512 Mb
c. *Monitor* : 14”
d. *Harddisk* : 80Gb
e. *Port USB* : PS/2
f. Printer
g. Piranti pendukung seperti *Keyboard*, *Mouse* dan *Floppy Drive.*

b) Kebutuhan perangkat lunak (*Sofiware*)

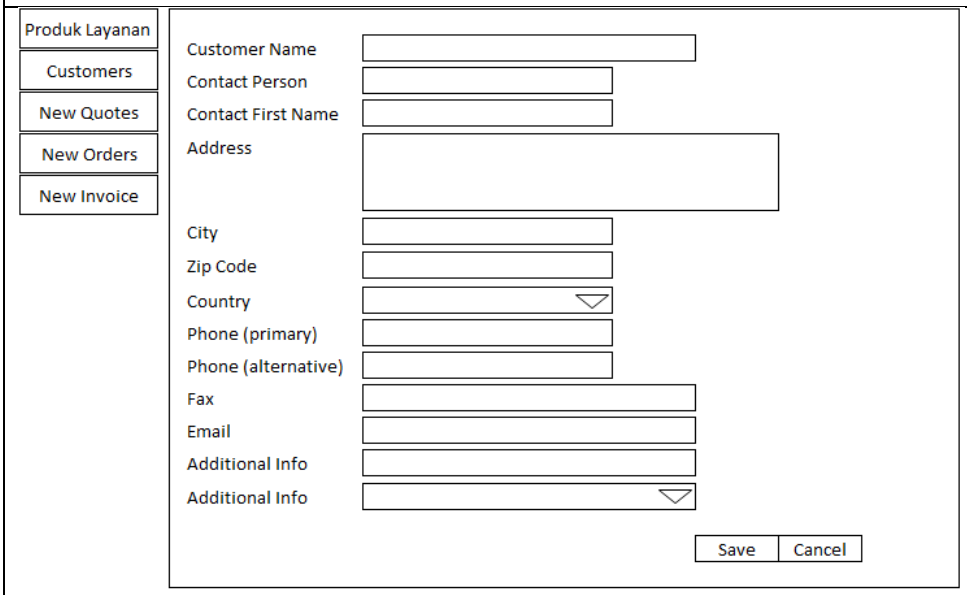
Di dalam pelaksanaan sistem baru selain perangkat keras (*Hardware*) dibutuhkan juga perangkat lunak (*Software*), Dimana perangkat lunak merupakan alat pendukung sistem yang terdiri dari sistem operasi dan aplikasi database. Adapun perangkat lunak yang digunakan adalah sebagai berikut :

e. Sistem Operasi : Window XP service pack 2
f. Database : MySQL
g. Bahasa Pemrograman : php
h. *Tool* yang digunakan : Macromedia Dreamweaver

3.8 Perancangan Antar Muka

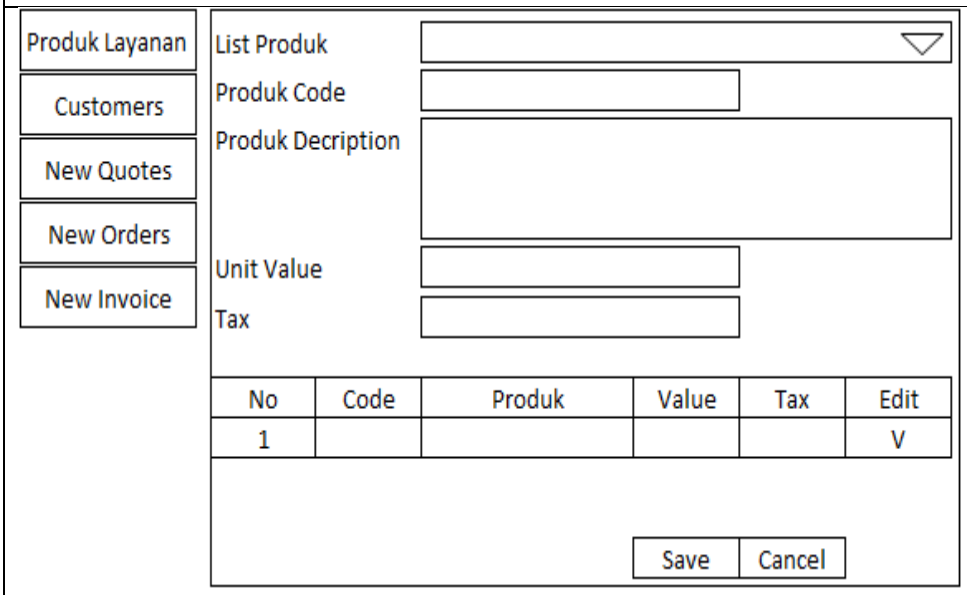
Tahapan perancangan antarmuka dalam pembuatan suatu aplikasi merupakan suatu tahapan yang sangat penting, karena tahapan ini akan menetukan apakah suatu aplikasi akan memiliki konstruksi yang baik, proses pengolahan data yang tepat dan akurat, sera memiliki dasar-dasar untuk pengembangan dimasa yang akan datang. Desain program yang dibuat harus mudah untuk digunakan, dapat dipahami dan dimengerti oleh pemakai (*user*), serta informasi yang dihasilkan dari program tersebut dapat dimengerti oleh pemakai.

3.8.1 Perancangan antarmuka *Form Customer*

<table>
<tr><td>T01</td><td></td></tr>
<tr><td>
</td><td>- Klik SAVE untuk menyimpan
- Klik Cancel untuk membatalkan
- Klik Produk Layanan untuk menuju ke T02
- Klik New Quotes untuk menuju ke T03
- Klik New Orders untuk menuju ke T04
- Klik New Invoice untuk menuju ke T05</td></tr>
<tr><td>General Layout
Ukuran tampilan : 900 X 1200 pixel
Font color :
Font size :
Font Typefaces :</td><td></td></tr>
</table>

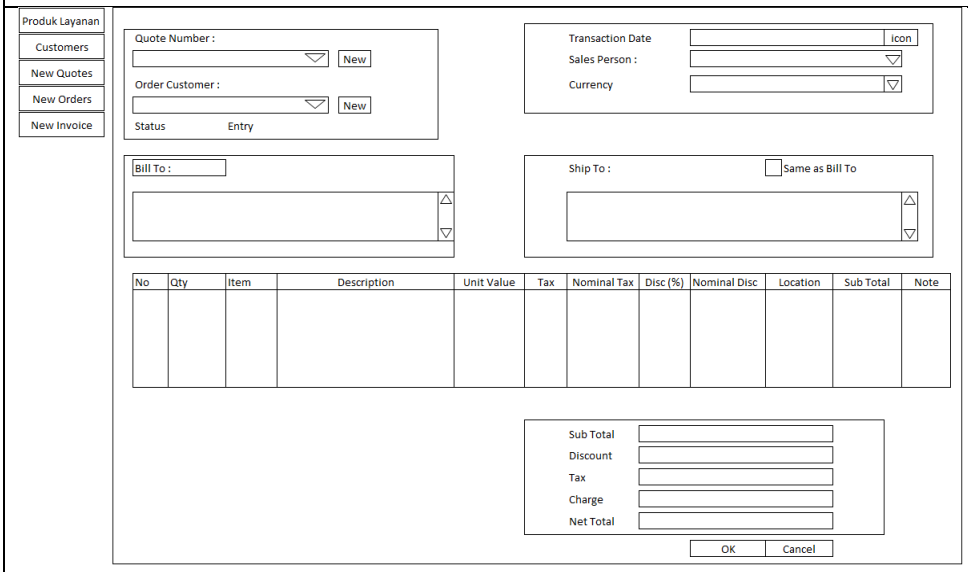
Gambar 3.12 Form Customer

3.8.2 Perancangan antarmuka *Form* Produk Layanan

<table>
<tr><td>T02

</td><td>- Klik SAVE untuk menyimpan
- Klik Cancel untuk membatalkan
- Klik Customers untuk menuju ke T01
- Klik New Quotes untuk menuju ke T03
- Klik New Orders untuk menuju ke T04
- Klik New Invoice untuk menuju ke T05</td></tr>
<tr><td>General Layout
Ukuran tampilan : 900 X 1200 pixel
Font color :
Font size :
Font Typefaces :</td><td></td></tr>
</table>

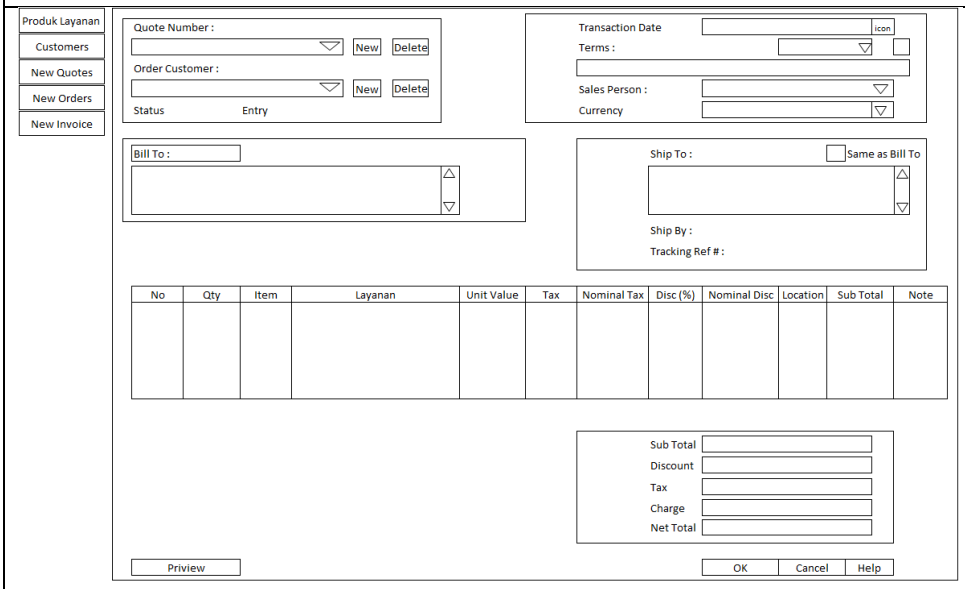
Gambar 3.13 *Form* Produk Layanan

3.8.3 Perancangan antarmuka *Form Quote*

<table>
<tr><td>T03</td><td></td></tr>
<tr><td>
</td><td>- Klik OK untuk menyimpan
- Klik Cancel untuk membatalkan
- Klik Customers untuk menuju ke T01
- Klik Produk Layanan untuk menuju ke T02
- Klik New Order untuk menuju ke T04
- Klik New Invoice untuk menuju ke T05</td></tr>
<tr><td>General Layout
Ukuran tampilan : 900 X 1200 pixel
Font color :
Font size :
Font Typefaces :</td><td></td></tr>
</table>

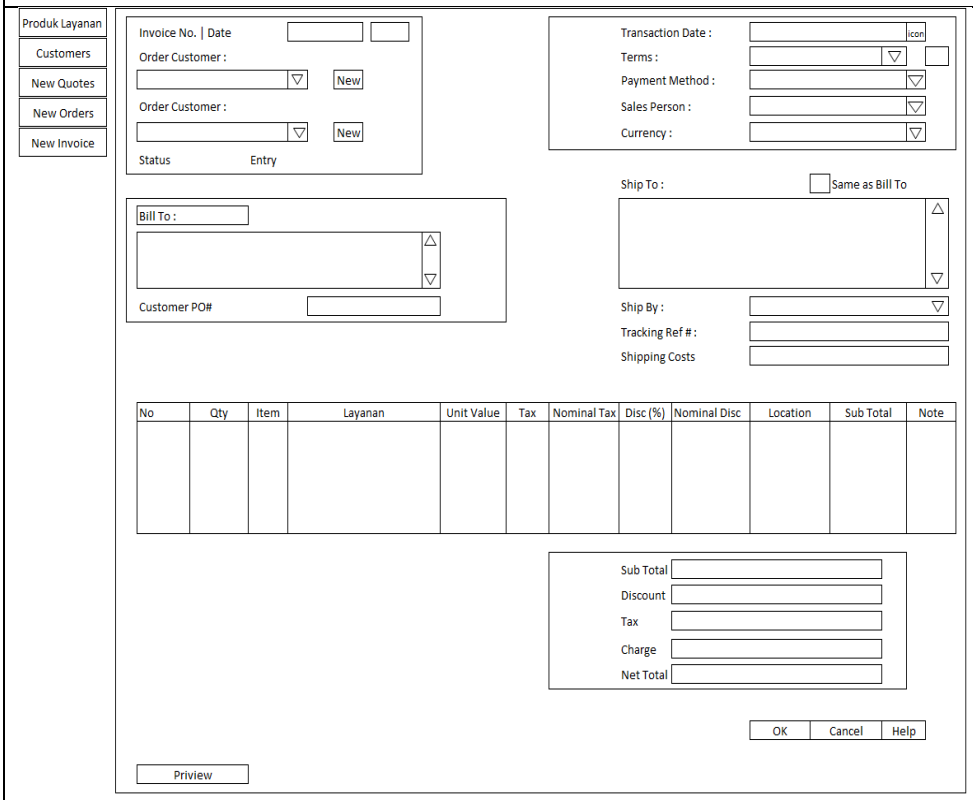
Gambar 3.14 Form Quote

3.8.4 Perancangan antarmuka *Form Order*

T04	
	- Klik *OK* untuk menyimpan - Klik *Cancel* untuk membatalkan - Klik *Customers* untuk menuju ke T01 - Klik Produk Layanan untuk menuju ke T02 - Klik *New Quotes* untuk menuju ke T03 - Klik *New Orders* untuk menuju ke T04 - Klik *New Invoice* untuk menuju ke T05
General Layout Ukuran tampilan : 900 X 1200 pixel Font color : Font size : Font Typefaces :	

Gambar 3.15 Form Order

3.8.5 Perancangan antarmuka *Form Invoice*

<table>
<tr><td>T05</td><td></td></tr>
<tr><td>
</td><td>- Klik OK untuk menyimpan
- Klik Cancel untuk membatalkan
- Klik Customers untuk menuju ke T01
- Klik Produk Layanan untuk menuju ke T02
- Klik New Quotes untuk menuju ke T03
- Klik New Order untuk menuju ke T04
- Klik New Invoice untuk menuju ke T05
- Klik Preview untuk membuat laporan</td></tr>
<tr><td>General Layout
Ukuran tampilan : 900 X 1200 pixel
Font color :
Font size :
Font Typefaces :</td><td></td></tr>
</table>

Gambar 3.16 Form Invoice

3.9 Implementasi

Pembuatan data dilakukan dengan menggunakan *database* MySQL 5.0. Implementasi basis data dalam bahasa SQL adalah sebagai berikut:

Database : ‘dbpo’

a. Implementasi struktur tabel ‘*Customer*’

```
CREATE TABLE IF NOT EXISTS `customer` (
  `customerID` int(10) NOT NULL AUTO_INCREMENT,
  `customerName` varchar(50) NOT NULL,
  `cp` varchar(50) NOT NULL,
  `cpFirstName` varchar(20) NOT NULL,
  `address` varchar(200) NOT NULL,
  `city` varchar(20) NOT NULL,
  `zipCode` varchar(10) NOT NULL,
  `country` varchar(20) NOT NULL,
  `firstPhone` varchar(15) NOT NULL,
  `secondPhone` varchar(15) NOT NULL,
  `fax` varchar(15) NOT NULL,
  `email` varchar(50) NOT NULL,
  `addinfo` varchar(100) NOT NULL,
  `group` varchar(20) NOT NULL,
  `insertDate` datetime NOT NULL,
  `insertUser` varchar(30) NOT NULL,
  `status` enum('y','n') NOT NULL,
  PRIMARY KEY (`customerID`)
) ENGINE=InnoDB  DEFAULT CHARSET=latin1 AUTO_INCREMENT=7 ;
```

b. Implementasi struktur tabel ‘*invoice*’

```
CREATE TABLE IF NOT EXISTS `invoice` (
  `invoiceID` int(10) NOT NULL AUTO_INCREMENT,
  `customerID` int(10) NOT NULL,
  `invoiceNumber` varchar(30) NOT NULL,
  `invoiceDate` date NOT NULL,
```

```
    `customerPONumber` varchar(50) NOT NULL,
    `terms` varchar(15) NOT NULL,
    `termsTime` int(11) NOT NULL,
    `paymentMethod` varchar(20) NOT NULL,
    `salesPerson` varchar(50) NOT NULL,
    `shipTo` varchar(200) NOT NULL,
    `shipBy` varchar(200) NOT NULL,
    `trackingRef` varchar(30) NOT NULL,
    `shippingCost` double(20,0) NOT NULL,
    `note` text NOT NULL,
    `subTotal` double(20,0) NOT NULL,
    `discount` double(20,0) NOT NULL,
    `charge` double(20,0) NOT NULL,
    `netTotal` double(20,0) NOT NULL,
    `insertDate` datetime NOT NULL,
    `insertUser` varchar(30) NOT NULL,
    `status` enum('open','paid') NOT NULL,
    PRIMARY KEY (`invoiceID`),
    KEY `customerID` (`customerID`)
  ) ENGINE=InnoDB DEFAULT CHARSET=latin1 AUTO_INCREMENT=1 ;
```

c. Implementasi struktur tabel '*invoicetransaction*'

```
CREATE TABLE IF NOT EXISTS `invoicetransaction` (
  `invoiceTransID` int(10) NOT NULL AUTO_INCREMENT,
  `invoiceID` varchar(10) NOT NULL,
  `productID` varchar(10) NOT NULL,
  `quantity` int(11) NOT NULL,
  `transDate` date NOT NULL,
  `firstTax` double(20,0) NOT NULL,
  `secondTax` double(20,0) NOT NULL,
  `discount` double(20,0) NOT NULL,
  `location` varchar(50) NOT NULL,
  `lineTotal` double(20,0) NOT NULL,
  `note` varchar(200) NOT NULL,
  PRIMARY KEY (`invoiceTransID`),
  KEY `productID` (`productID`),
  KEY `invoiceID` (`invoiceID`)
```

```
) ENGINE=InnoDB DEFAULT CHARSET=latin1 AUTO_INCREMENT=1 ;
```

d. Implementasi struktur tabel '*order*'

```
CREATE TABLE IF NOT EXISTS `order` (
  `orderID` int(10) NOT NULL AUTO_INCREMENT,
  `customerID` int(10) NOT NULL,
  `orderNumber` varchar(30) NOT NULL,
  `orderDate` date NOT NULL,
  `terms` varchar(15) NOT NULL,
  `salesPerson` varchar(50) NOT NULL,
  `shipTo` varchar(200) NOT NULL,
  `shipBy` varchar(20) NOT NULL,
  `trackingRef` varchar(30) NOT NULL,
  `orderNote` text NOT NULL,
  `subTotal` double(20,0) NOT NULL,
  `discount` double(20,0) NOT NULL,
  `charge` double(20,0) NOT NULL,
  `netTotal` double(20,0) NOT NULL,
  `insertDate` datetime NOT NULL,
  `insertUser` varchar(30) NOT NULL,
  `status` enum('open','invoiced') NOT NULL,
  PRIMARY KEY (`orderID`),
  KEY `customerID` (`customerID`)
) ENGINE=InnoDB  DEFAULT CHARSET=latin1 AUTO_INCREMENT=2 ;
```

e. Implementasi struktur tabel '*ordertransaction*'

```
CREATE TABLE IF NOT EXISTS `ordertransaction` (
  `orderTransID` varchar(10) NOT NULL,
  `orderID` varchar(10) NOT NULL,
  `productID` varchar(10) NOT NULL,
  `quantity` int(11) NOT NULL,
  `transDate` date NOT NULL,
  `firstTax` double(20,0) NOT NULL,
  `secondTax` double(20,0) NOT NULL,
  `discount` double(20,0) NOT NULL,
```

```
  `location` varchar(50) NOT NULL,
  `lineTotal` double(20,0) NOT NULL,
  `note` varchar(100) NOT NULL,
  PRIMARY KEY (`orderTransID`),
  KEY `orderID` (`orderID`),
  KEY `productID` (`productID`)
) ENGINE=InnoDB DEFAULT CHARSET=latin1;
```

f. Implementasi struktur tabel '*product*'

```
CREATE TABLE IF NOT EXISTS `product` (
  `productID` varchar(10) NOT NULL,
  `productName` varchar(30) NOT NULL,
  `description` text NOT NULL,
  `unitValue` double(20,0) NOT NULL,
  `firstTax` double(20,0) NOT NULL,
  `secondTax` double(20,0) NOT NULL,
  `status` enum('y','n') NOT NULL,
  PRIMARY KEY (`productID`)
) ENGINE=InnoDB DEFAULT CHARSET=latin1;
```

g. Implementasi struktur tabel '*quotes*'

```
CREATE TABLE IF NOT EXISTS `quotes` (
  `quotesID` int(10) NOT NULL AUTO_INCREMENT,
  `customerID` int(10) NOT NULL,
  `quotesDate` date NOT NULL,
  `salesPerson` varchar(50) NOT NULL,
  `shipTo` varchar(200) NOT NULL,
  `note` text NOT NULL,
  `subTotal` double(20,0) NOT NULL,
  `discount` double(20,0) NOT NULL,
  `charge` double(20,0) NOT NULL,
  `netTotal` double(20,0) NOT NULL,
  `insertDate` datetime NOT NULL,
  `insertUser` varchar(30) NOT NULL,
  `status` enum('open','ordered') NOT NULL,
  PRIMARY KEY (`quotesID`),
  KEY `customerID` (`customerID`)
```

```
) ENGINE=InnoDB  DEFAULT CHARSET=latin1 AUTO_INCREMENT=2 ;
```

h. Implementasi struktur tabel '*quotestransaction*'

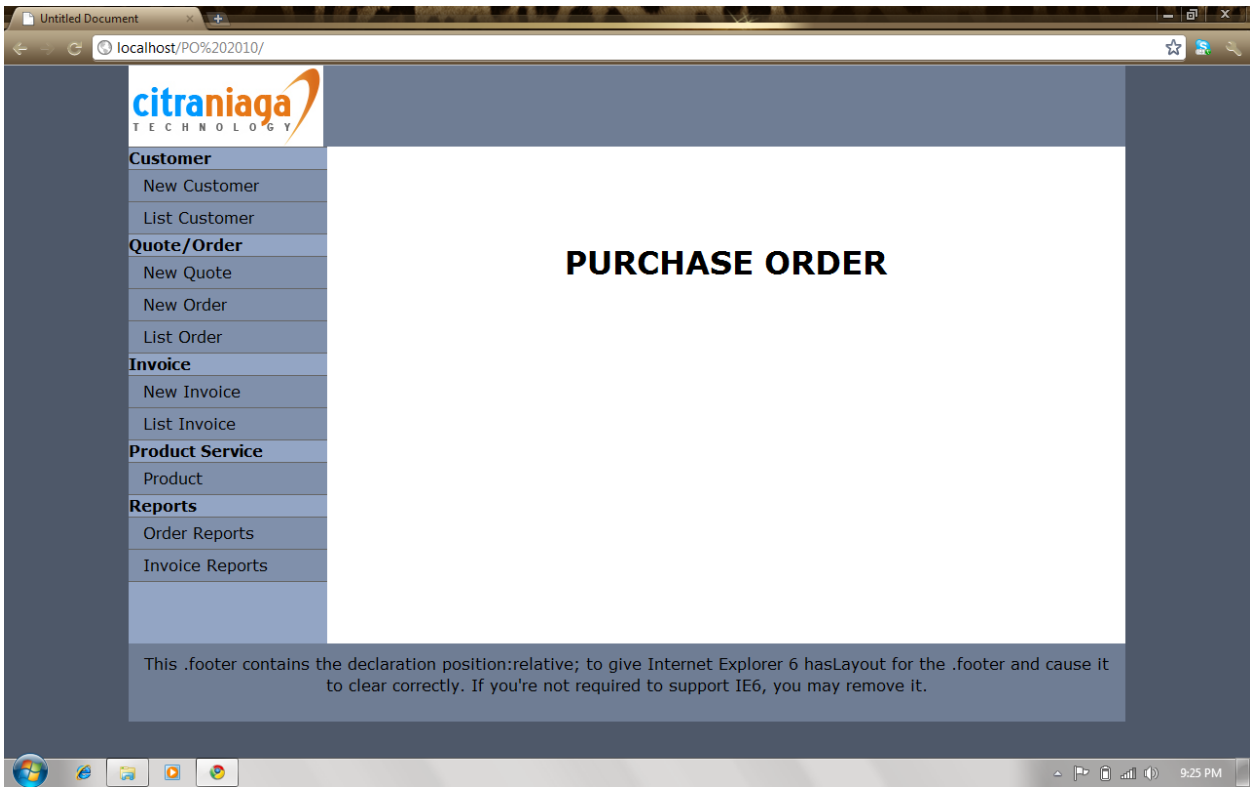
```
CREATE TABLE IF NOT EXISTS `quotestransaction` (
  `quotesTransID` int(10) NOT NULL AUTO_INCREMENT,
  `quotesID` varchar(10) NOT NULL,
  `productID` varchar(10) NOT NULL,
  `quantity` int(11) NOT NULL,
  `transDate` date NOT NULL,
  `firstTax` double(20,0) NOT NULL,
  `secondTax` double(20,0) NOT NULL,
  `discount` double(20,0) NOT NULL,
  `location` varchar(50) NOT NULL,
  `lineTotal` double(20,0) NOT NULL,
  `note` varchar(100) NOT NULL,
  PRIMARY KEY (`quotesTransID`),
  KEY `productID` (`productID`),
  KEY `quotesID` (`quotesID`)
) ENGINE=InnoDB  DEFAULT CHARSET=latin1 AUTO_INCREMENT=2 ;
```

3.9.1 Implementasi Antarmuka

Implementasi menjelaskan *interface* yang terdapat pada sistem aplikasi *Purchase Order*, diantarnya Tampilan *index*, tampilan *customer*, tampilan *quote / order*, tampilan *invoice*, dan tampilan *product* dan *list product*.

3.9.1.1 Index

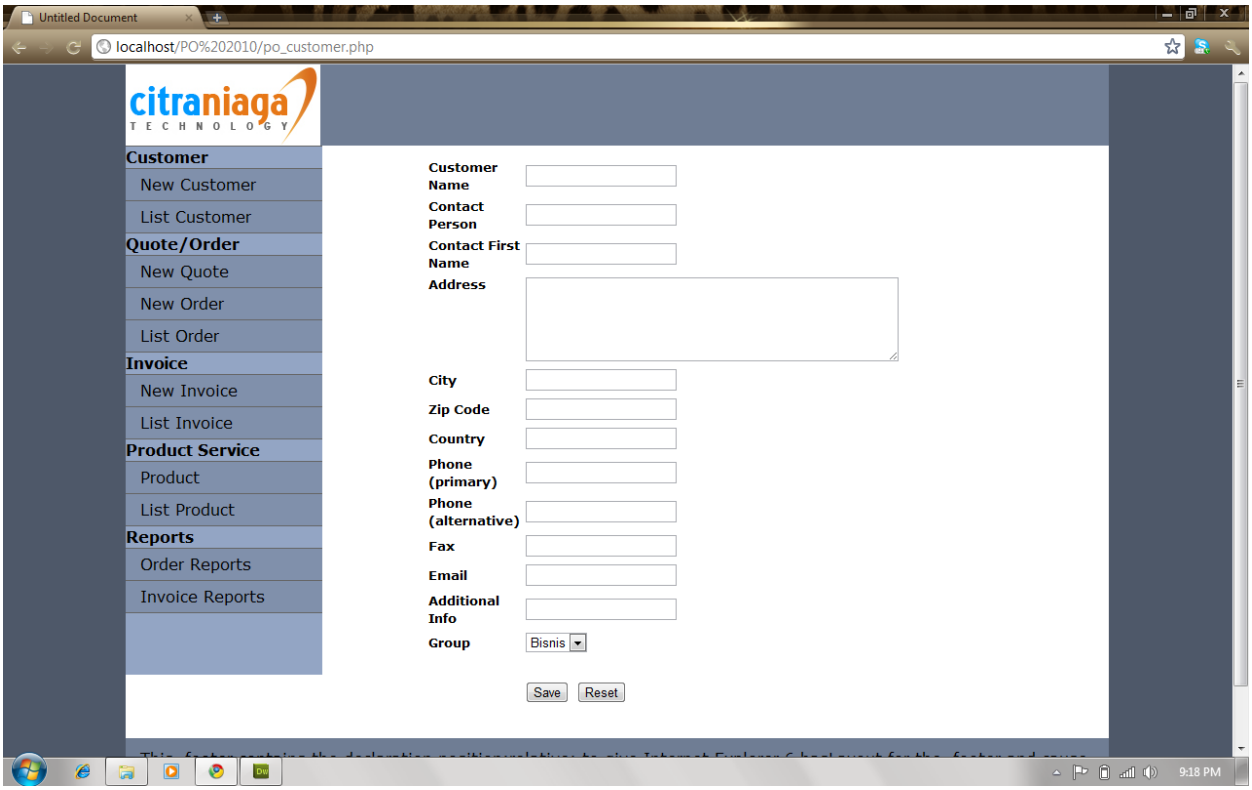
Ini adalah tampilan awal saat anda masuk kedalam program aplikasi *Purchase Order*. Dan terdapat pilihan menu lain untuk melanjutkan kepada menu lain dari *Purchase Order*. Tampilan menu *index* dapat dilihat pada gambar 3.17.



Gambar 3.17 Tampilan Index

3.9.1.2 Customer

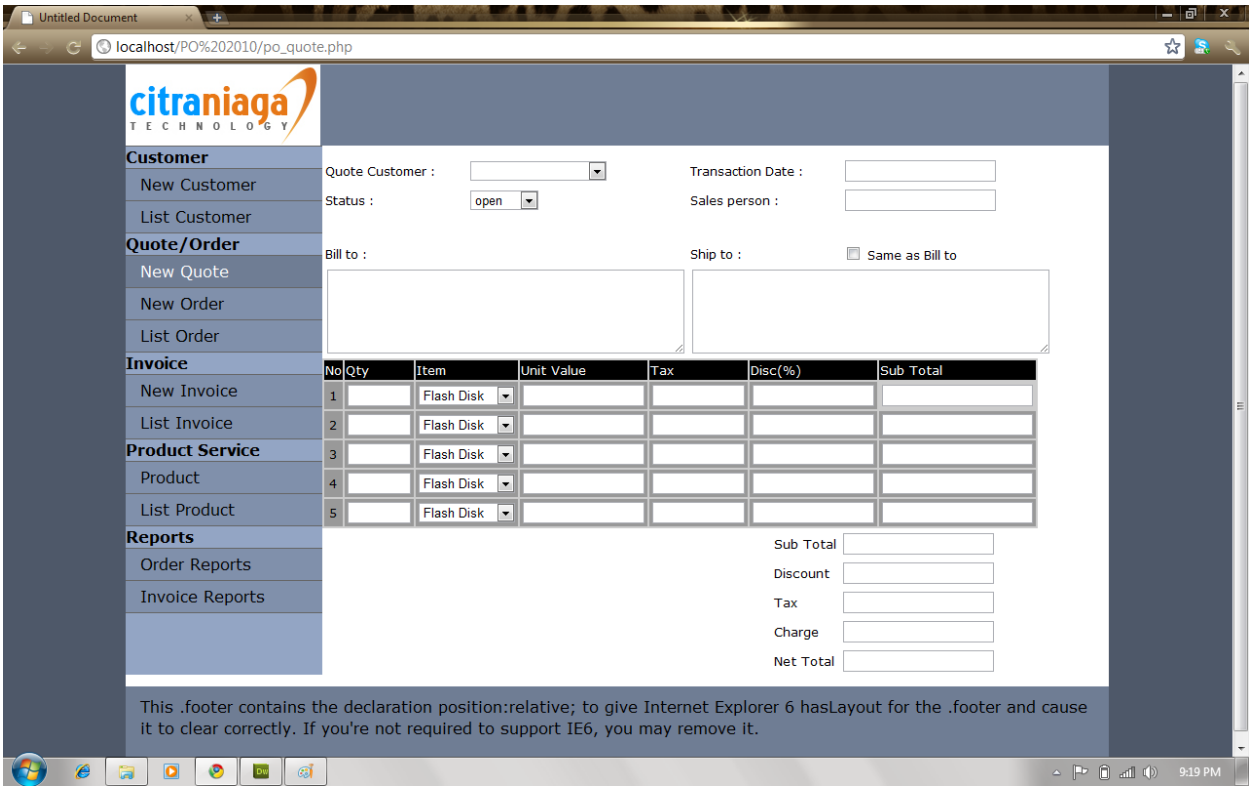
Tampilan ini berfungsi untuk mencatat data pelanggan jika ingin melakukan pemesanan dengan tersedianya tombol *save* yang digunakan untuk menyimpan data yang telah dimasukkan dan tombol *reset* jika ingin mengulang data yang telah ditulis atau diketikkan tadi.



Gambar 3.18 Customer

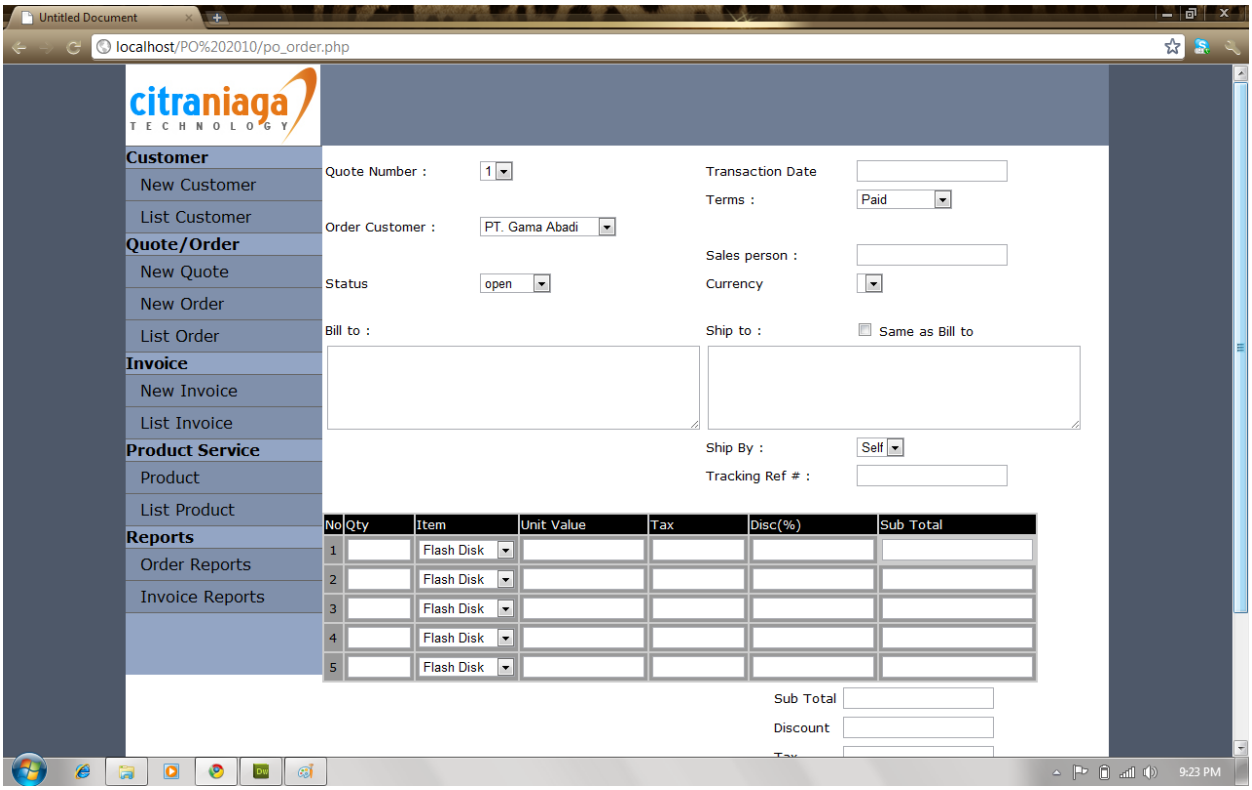
3.9.1.3 *Quote* dan *Order*

Pada tampilan ini berguna untuk membuat antrian pengiriman barang. Dan tampilan ini terdapat kolom untuk pengisian alamat tagihan dan alamat pengiriman barang. Tampilan layar *Quote* dapat dilihat pada gambar 3.19.



Gambar 3.19 Quote

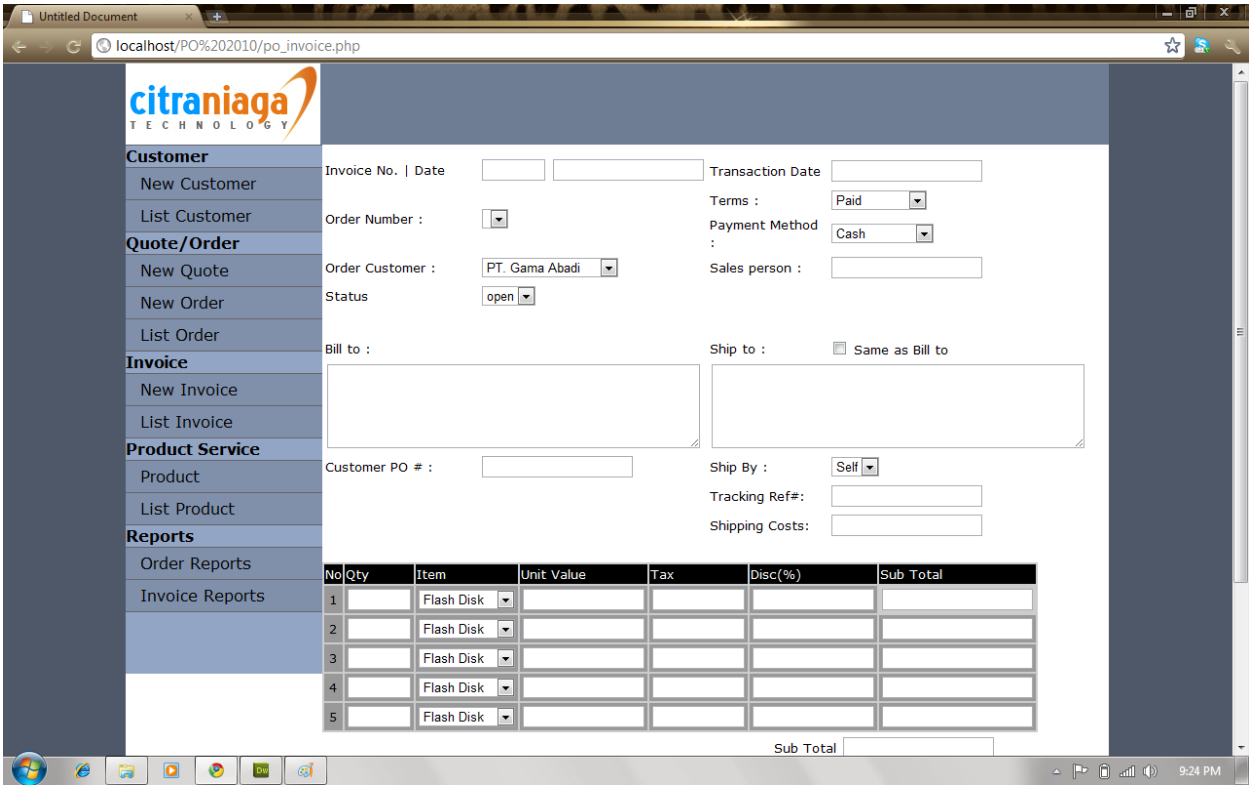
Tampilan ini digunakan untuk melakukan pemesanan barang. Disini juga akan menampilkan sebagian data yang telah dimasukkan pada saat pengisian pada *form Quote*. Pada tampilan ini pun dapat menghitung jumlah bayar. Tampilan pemesanan dapat dilihat pada gambar 3.20.



Gambar 3.20 Order

3.9.1.4 Invoice

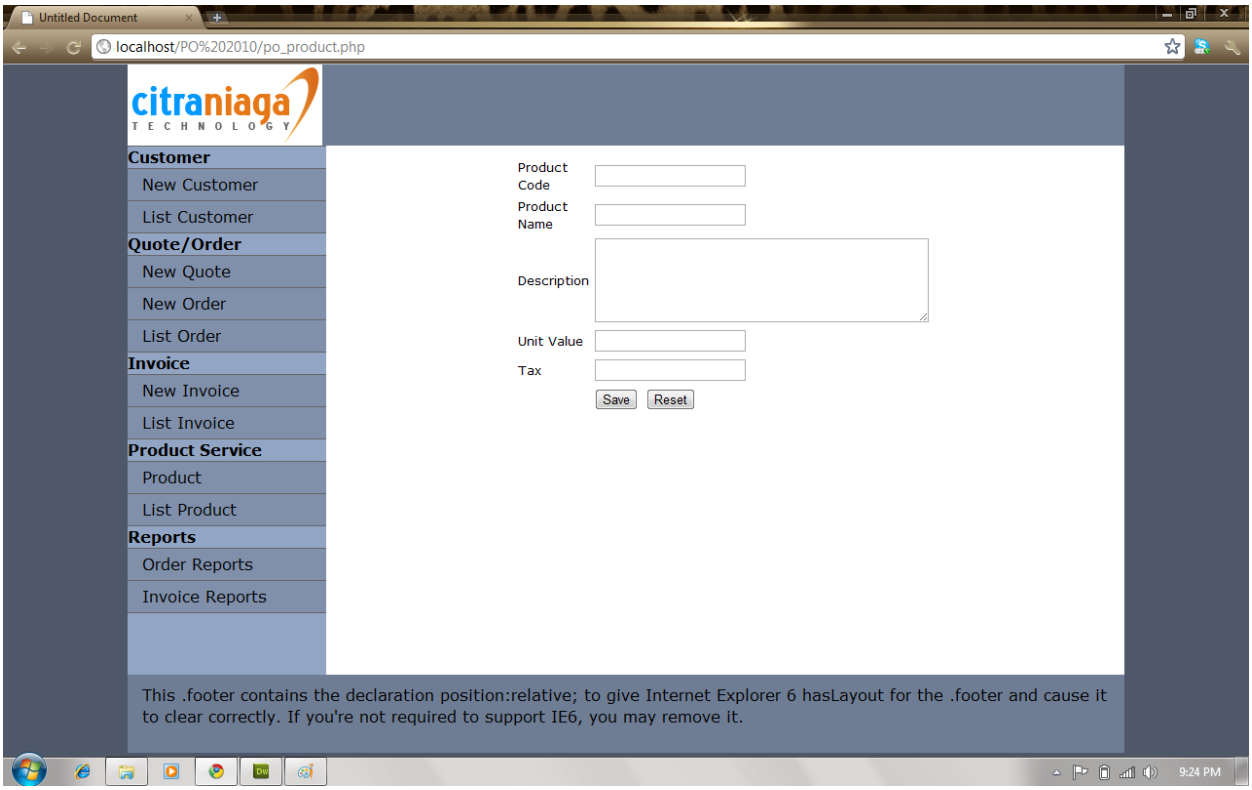
Pada tampilan ini akan mengisikan *Invoice* baru setelah *New Invoice* di klik. Setelah disimpan maka akan dapat dilihat pada form *List Invoice* dan dapat dicetak atau langsung di *print.* Untuk melihat gambar mengenai tampilan *Invoice* secara lebih lengkap dapat dilihat pada gambar 3.21.



Gambar 3.21 Invoice

3.9.1.5 *Product* dan *List Product*

Tampilan ini digunakan untuk mengetahui produk dan deskripsi dari produk yang tersedia. Deskripsi produk akan ditampilkan secara jelas sampai dengan harga yang ditawarkan. Tampilan *Product* dapat dilihat pada gambar 3.22.



Gambar 3.22 Product

Tampilan ini menampilkan daftar dari produk yang ditawarkan, selain itu kode dari produk yang ditawarkan pun akan ditampilkan karena dapat dicocokan kembali jika merasa ragu dengan *Invoice* yang dikeluarkan. Untuk lebih jelas mengenai tampilan *List Product* dapat dilihat pada gambar 3.13.



Gambar 3.23 List Produc