

APLIKASI PENJUALAN DI WAROENG GUBRAK
PROYEK AKHIR 1

Oleh :

Novia Ramanda 3311211037

Busria Julita 3311211050

Yessi Sinaga 3311211062



PROGRAM STUDI TEKNIK INFORMATIKA
POLITEKNIK NEGERI BATAM
BATAM
2013

DAFTAR ISI

Bab I	Pendahuluan	1
I.1	Latar Belakang	1
I.2	Rumusan Masalah	1
I.3	Batasan Masalah.....	1
I.4	Tujuan.....	2
I.5	Sistematika Penulisan.....	2
Bab II	Landasan Teori	3
2.1	POS(poin opsel).....	3
2.2	Irepoart.....	4
2.3	Pengertian Netbeans IDE	5
2.4	Xampp	6
2.5	Mysql.....	7
2.6	Database	8
Bab III	Analisis dan Perancangan.....	10
3.1	Perancangan Sistem Dengan Menggunakan UML	10
3.2	Use Case Diagram	10
2.4	Database	10

DAFTAR GAMBAR

Gambar 3.1 Use Case.....	1
Gambar 3.2 Squery Kasir.....	13
Gambar 3.3 Squery Manager	13

DAFTAR TABEL

Tabel 3.1 Definisi Aktor	10
Tabel 3.2 Definisi Dalam Use Case	11

BAB I

Pendahuluan

I.1 Latar Belakang

Bidang usaha warung makan merupakan salah satu bidang yang masih bisa bertahan dan bahkan berkembang di dalam kondisi perekonomian Indonesia yang tidak menentu. Keunikan suatu warung dengan memiliki ciri khas yang menarik yaitu menu-menu yang akan disajikan bersumber dari makanan khas nusantara. Warung GUBRAKS ini beroperasi pukul 10.00 WIB dan tutup pukul 22.00 WIB. Desain bangunannya minimalis dengan suasana tradisional, Waroeng gubraks berdiri di kepri mall lantai 2 no 26, terletak di bagian yang sangat strategis, terdapat perkantoran OSEDU, dan banyaknya para pengunjung yang datang ke kepri mall pada saat hari libur dan makan siang.

Dengan kendala yang dialami pada saat melakukan penelitian, proses transaksi penjualan, perhitungan dan laporan penjualan oleh karena itu penulis menciptakan aplikasi yang dapat memudahkan dalam melakukan pembayaran dan mengelola stok kebutuhan. Sistem hanya biasa melakukan pembayaran dan mengelola stok kebutuhan, sistem hanya untuk warung gubraks dan sistem ini hanya biasa dilakukan oleh kasir dan manager. supaya dapat mempercepat transaksi, mempermudah transaksi pembayaran dan tidak terjadi kecurangan dalam penjualan.

I.2 Rumusan Masalah

Rumusan masalah yang dapat disimpulkan, sebagai berikut:

1. Bagaimana proses transaksi penjualan dan perhitungan transaksi secara cepat?
2. Bagaimana laporan penjualan dapat dilakukan dengan cepat?

I.3 Batasan Masalah

Batasan dari proyek akhir ini adalah:

1. Perancangan mesin kasir yang bisa di login hanya kasir dan managernya saja
2. Sistem hanya berhubungan dengan pembayaran dan stok barang
3. Sistem ini hanya untuk "Warung Gubrak" yang menjadi objek pada penelitian ini.

I.4 Tujuan

Adapun tujuan laporan ini adalah :

1. Untuk mempermudah transaksi nota penjualan pembelian makanan konsumen
2. Untuk lebih akurat dalam menghitung nota penjualan
3. Agar tidak terjadinya kecurangan dalam nota penjualan

I.5 Sistematika Penulisan

Bab I Pendahuluan bab ini menjelaskan tentang latar belakang waroeng gubraks dan penulisan proyek akhir .

Bab II Landasan Teori bab ini menjelaskan tentang teori-teori yang berhubungan dengan penelitian.

Bab III Analisis dan Perancangan bab ini menjelaskan tentang langkah-langkah dalam penyelesaian masalah.

Bab IV implementasi program dan pengujian aplikasi.

Bab V Kesimpulan dan Saran.

BAB II

TINJAUAN PUSTAKA

Dalam Bab ini penulis akan memberikan beberapa materi yang berhubungan dengan judul yang akan diajukan. Hal ini diharapkan untuk mempermudah proses pembuatan proyek serta informasi yang disajikan sesuai dengan hasil yang diharapkan.

Perkembangan dunia komputer semakin lama semakin berkembang dengan pesat, sehingga mendorong setiap individu maupun kelompok harus menerapkannya dalam segala bidang. Perkembangan tersebut dapat menimbulkan hal yang positif dan juga hal yang negatif (Darwin Sitompul, 1994).

2.1 POINT OF SALES (POS)

Dampak-dampak positif dari perkembangan dunia komputer salah satunya adalah dengan ditemukannya/dimodifikasinya mesin *Cash Register* menjadi mesin POS (*Point of sales*) yang digunakan dalam bidang penjualan, khususnya *retail* toko dan sebagainya. Yang terbukti mempermudah proses transaksi penjualan.

Point of sales adalah Sebuah aplikasi/program komputer yang terdapat pada mesin kasir digunakan untuk proses transaksi yang telah dilengkapi dengan sistem pelaporan manajemen yang terintegrasi dan biasanya berorientasi pada penjualan.

Ada 2 komponen penting dalam sistem *Point of sales* yaitu :

POS software adalah program yang membantu fungsi seluruh sistem dan bertanggung jawab untuk mengirimkan dan menerima informasi atau data dalam sistem.

POS hardware adalah layar *display* pelanggan, laci kas, perangkat menggesekkan (untuk kartu kredit), printer, komputer dan pembaca *barcode*.

2.2 IREPORT

Report / laporan sangat diperlukan dalam suatu aplikasi sistem informasi. Tools yang cukup dikenal untuk membuat laporan yaitu Crystal Report dan biasanya digabungkan dengan *Visual Basic*. Sebagai alternatif, terdapat *tools iReport* (dengan *library JasperReport*) yang dapat pula membantu kita dalam pembuatan laporan. *Library JasperReport* sendiri merupakan *Java Library (JAR)* yang bersifat open dan dirancang untuk menambahkan kemampuan pelaporan (*reporting capabilities*) pada aplikasi java.

JasperReport memiliki sejumlah fitur, antara lain :

- *Layout* dan desain laporan yang *fleksibel*
- Dapat menampilkan laporan dalam bentuk teks maupun gambar (*chart*)
- Dapat menghasilkan *report* dalam berbagai format : *html, pdf, rtf, xls, csv*.
- Dapat menerima data dari berbagai sumber data : *JDBC, Bean Collection, ResultSet, CSV, XML, Hibernate*.

Menggunakan *iReport/JasperReport* dan menghubungkannya dengan *file java*.

Beberapa *tools* yang dibutuhkan :

1. *JasperReports* 3.0.0 (atau versi terakhir, dapat dilihat di <https://olex.openlogic.com/packages/jasperreports>).
2. *iReport* 3.0.0 (atau versi terakhir, versi harus sama dengan versi *JasperReports* yang digunakan)
3. *Eclipse IDE*
4. *Sun JDK 1.5*
5. *Apache Tomcat 5.5*
6. *Adobe Acrobat Reader*

2.3 Netbeans 7.2.0

NetBeans adalah *platform* pengembangan kerangka kerja dan *open source IDE* (*Integrated Development Environment*) untuk *Java*, *JavaScript*, *PHP*, dll. Yang dikembangkan oleh *Oracle Corporation* dan bebas biaya.

IDE adalah lingkup pemrograman yang diintegrasikan kedalam suatu aplikasi perangkat lunak yang menyediakan pembangun *Graphic User Interface* (GUI), suatu *text* atau kode editor, suatu *compiler* atau *interpreter* dan suatu *debugger*.

NetBeans IDE digunakan untuk menulis, mengompilasi, mencari kesalahan dan menyebarkan program. *Netbeans IDE* ditulis dalam *Java*, namun dapat mendukung bahasa pemrograman lain. Selain itu *NetBeans IDE* merupakan produk bebas dan tanpa batasan apapun.

NetBeans IDE sangat cocok jika digunakan dalam proyek ini karena ditulis dalam bahasa *java*, bersifat bebas serta berbasis GUI. Sehingga mempermudah proses penelitian dan pembuatan proyek ini.

Fitur-fitur yang terdapat dalam netbeans antara lain :

- *Smart Code Completion*: untuk mengusulkan nama *variabel* dari suatu tipe, melengkapi *keyword* dan mengusulkan tipe parameter dari sebuah *method*.
- *Bookmarking*: fitur yang digunakan untuk menandai baris yang suatu saat hendak kita modifikasi.
- *Go to commands*: fitur yang digunakan untuk *jump* ke deklarasi variabel, *source code* atau file yang ada pada *project* yang sama.
- *Code generator*: jika kita menggunakan fitur ini kita dapat meng-*generate constructor, setter and getter method* dan yang lainnya.
- *Error stripe*: fitur yang akan menandai baris yang *error* dengan memberi *highlight* merah.

2.4 Database

Beberapa definisi tentang Database :

- Menurut Gordon C. Everest :
Database adalah koleksi atau kumpulan data yang mekanis, terbagi/shared, terdefinisi secara formal dan dikontrol terpusat pada organisasi.
- Menurut C.J. Date :
Database adalah koleksi “data operasional” yang tersimpan dan dipakai oleh sistem aplikasi dari suatu organisasi.
 - Data input adalah data yang masuk dari luar sistem
 - Data output adalah data yang dihasilkan sistem
 - Data operasional adalah data yang tersimpan pada system
- Menurut Toni Fabbri :
Database adalah sebuah sistem file-file yang terintegrasi yang mempunyai minimal primary key untuk pengulangan data.
- Menurut S. Attre :
Database adalah koleksi data-data yang saling berhubungan mengenai suatu organisasi / enterprise dengan macam-macam pemakaiannya.

Database yang didukung oleh Netbeans antara lain:

1. JDBC merupakan spesifikasi standar dari JavaSoft API (Application Programming Interface) yang memungkinkan program Java untuk mengakses sistem database manajemen. JDBC API terdiri dari satu set interface dan kelas yang ditulis dalam bahasa pemrograman Java. JDBC API Menggunakan interface standar dan kelas, programmer dapat menulis aplikasi yang terhubung ke database, mengirimkan pertanyaan ditulis SQL (Structured Query Language), dan memproses hasilnya.
2. ORACLE merupakan sebuah aplikasi basisdata yang didukung oleh NetBeans, Oracle secara umum hampir sama dengan *Mysql* namun yang membedakan adalah oracle dapat digunakan dan dihubungkan dengan netbeans dan harus menggunakan drivers untuk menyimpan data-data yang telah dibuat. Oracle merupakan salah satu dari beberapa aplikasi basisdata yang sering digunakan untuk koneksi ke basisdata pada NetBeans karena relatif mudah dan cepat.
3. *Mysql* merupakan sebuah aplikasi basisdata yang didukung oleh NetBeans, *Mysql* memiliki Fitur Yang ada sudah lumayan lengkap, dari input, update, delete serta

Search. Sebuah antarmuka ODBC memanggil MyODBC yang memungkinkan setiap bahasa pemrograman yang mendukung ODBC untuk berkomunikasi dengan basis data *Mysql*. Kebanyakan kode sumber *Mysql* dalam ANSI C.

4. DataBase Access (Microsoft Access) merupakan sebuah aplikasi basis data yang didukung oleh NetBeans, data dapat disimpan di dalam format Microsoft Access, Microsoft Jet Database Engine, Microsoft SQL Server, Oracle Database, atau semua kontainer basis data yang mendukung standar ODBC. Access juga mendukung teknik-teknik pemrograman berorientasi objek.

2.5 Xampp

Xampp merupakan singkatan dari x (empat system operasi apapun), *Apache* perangkat *Msql, PHP, Perl*. *Xampp* merupakan tool yang menyediakan paket lunak kedalam satu buah paket. Dalam paketnya sudah terdapat *Apache*(web server), *Mysql*(database), *PHP*(server side scripting), *Perl*, *ftp server*, *phpmyadmin*. Cukup dengan menginstal *xampp* kita dapat keempat paket tersebut.

2.6 MYSQL

Mysql adalah adalah sebuah perangkat lunak Pembuat database yang bersifat terbuka atau *open source* dan berjalan disemua platform baik *Linux* maupun si *Windows*, *Mysql* merupakan program pengakses database yang bersifat *network* sehingga dapat digunakan untuk aplikasi *Multi User* (Pengguna Banyak).

Mysql merupakan *server* basis data yang menggunakan teknik relasional untuk menghubungkan antara tabel-tabel dalam basis data. *Mysql* dikeluarkan secara gratis dibawah lisensi GNU *General PublicLicense* (GPL) .

Mysql sebenarnya merupakan turunan salah satu konsep utama dalam basisdata yang telah ada sebelumnya; *SQL* (*Structured Query Language*). *SQL* adalah sebuah konsep pengoperasian basisdata, terutama untuk pemilihan atau seleksi dan pemasukan data, yang memungkinkan pengoperasian data dikerjakan dengan mudah secara otomatis.

Mysql memiliki beberapa keistimewaan, antara lain :

- Portabilitas, *Mysql* dapat berjalan stabil pada berbagai sistem operasi seperti Windows, Linux, FreeBSD, Mac Os X Server, Solaris, Amiga, dan masih banyak lagi.
- Perangkat lunak sumber terbuka, *Mysql* didistribusikan sebagai perangkat lunak sumber terbuka, dibawah lisensi GPL sehingga dapat digunakan secara gratis.
- *Multi-user*, *Mysql* dapat digunakan oleh beberapa pengguna dalam waktu yang bersamaan tanpa mengalami masalah atau konflik.
- '*Performance tuning*', *Mysql* memiliki kecepatan yang menakjubkan dalam menangani *query* sederhana, dengan kata lain dapat memproses lebih banyak SQL per satuan waktu.
- Ragam tipe data, *Mysql* memiliki ragam tipe data yang sangat kaya, seperti signed / unsigned integer, float, double, char, text, date, timestamp, dan lain-lain.
- Perintah dan Fungsi, *Mysql* memiliki operator dan fungsi secara penuh yang mendukung perintah select dan where dalam perintah (*query*).
- Keamanan, *Mysql* memiliki beberapa lapisan keamanan seperti level subnet mask, nama host, dan izin akses user dengan sistem perizinan yang mendetail serta sandi terenkripsi.
- Skalabilitas dan Pembatasan, *Mysql* mampu menangani basis data dalam skala besar, dengan jumlah rekaman (records) lebih dari 50 juta dan 60 ribu tabel serta 5 milyar baris. Selain itu batas indeks yang dapat ditampung mencapai 32 indeks pada tiap tabelnya.
- Konektivitas, *Mysql* dapat melakukan koneksi dengan klien menggunakan protokol TCP/IP, Unix socket (UNIX), atau Named Pipes (NT).
- Lokalisasi, *Mysql* dapat mendeteksi pesan kesalahan pada klien dengan menggunakan lebih dari dua puluh bahasa. Meski pun demikian, bahasa Indonesia belum termasuk di dalamnya.
- Antar Muka, *Mysql* memiliki antar muka (*interface*) terhadap berbagai aplikasi dan bahasa pemrograman dengan menggunakan fungsi API (*Application Programming Interface*).

- Klien dan Peralatan, *Mysql* dilengkapi dengan berbagai peralatan (tool) yang dapat digunakan untuk administrasi basis data, dan pada setiap peralatan yang ada disertakan petunjuk online.
- Struktur tabel, *Mysql* memiliki struktur tabel yang lebih fleksibel dalam menangani ALTER TABLE, dibandingkan basis data lainnya semacam *PostgreSQL* ataupun *Oracle*

Bab III Analisis dan Perancangan

3.1 Deskripsi umum

3.1.1 kebutuhan antarmuka eksternal

Kebutuhan antarmuka eksternal pada perangkat lunak proyek akhir ini mencakup kebutuhan antarmuka pemakai dan antar muka perangkat lunak.

3.1.1.1 Antarmuka pemakai

Antarmuka pemakai akan dikembangkan dengan menggunakan mesin casregister dan komputer, pemakai berinteraksi dengan perangkat lunak melalui antarmuka kasir dengan customer. aplikasi akan dijalankan oleh kasir dan customer melakukan pembayaran secara langsung kepada kasir setelah mengambil barang yang mau dibeli oleh customer.

3.1.1.2 Antarmuka perangkat keras

Kebutuhan minimum perangkat keras yang akan digunakan oleh proyek akhir ini adalah

- ❖ PC IBM Compatible dan mesin casregister
- ❖ Papan kunci (Keyboard)
- ❖ Mouse

3.1.1.3 Antarmuka perangkat lunak

Kasir dan customer langsung melakukan transaksi setelah mengambil barang yang akan dibeli oleh customer, kasir langsung menginput data secara langsung dengan menggunakan aplikasi yang ada.

3.1.1.4 Antarmuka komunikasi

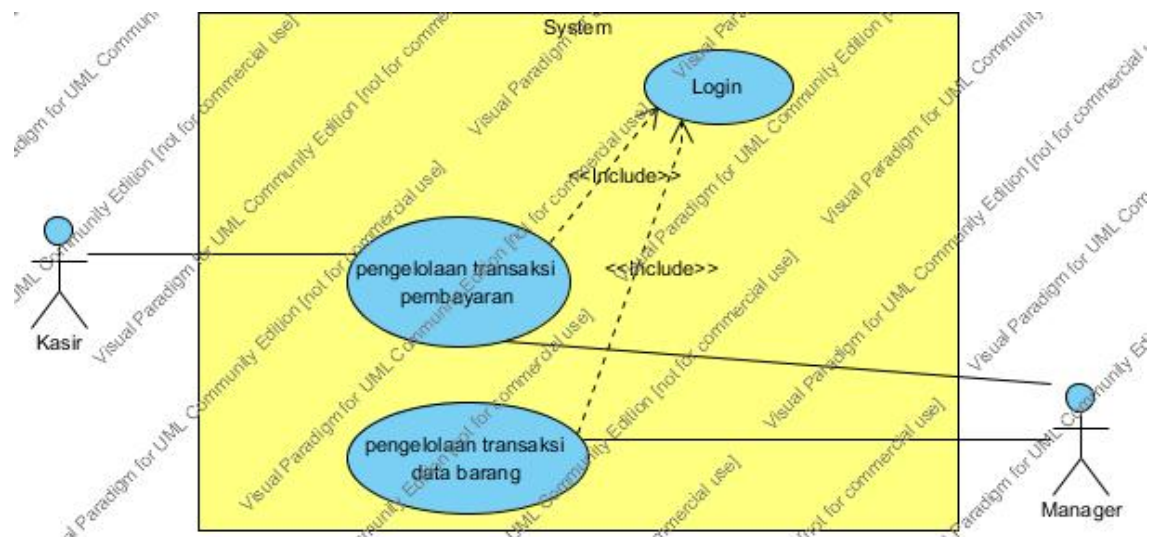
Dibutuhkan sebuah komputer untuk menginput pembelian barang yang dibeli oleh customer.

3.2 Perancangan Sistem Dengan Menggunakan UML

Perancangan aplikasi ini dirancang menggunakan alat bantu berupa UML (*Unified Modelling Language*) agar mempermudah memindahkan konsep sistem yang dirancang ke dalam bentuk program, dimana perancangannya digambarkan dalam bentuk diagram-diagram berikut:

3.3 Use Case Diagram

Use case menggambarkan bagaimana seseorang akan menggunakan atau memanfaatkan sistem, Sedangkan aktor adalah seseorang atau sesuatu yang berinteraksi dengan sistem. *Use case* diagram menggambarkan bagaimana proses-proses yang dilakukan oleh aktor terhadap sebuah system



Gambar 3.1 Use Case

3.2 Definisi Actor

Definisi aktor adalah aktivitas yang bisa dilakukan oleh para aktor dalam menggunakan sistem.

No	Aktor	Deskripsi
1	<i>Manager</i>	<i>Aktor</i> yang memonitoring segala sistem.
2	Admin	<i>Aktor</i> yang melakukan aktifitas didalam sistem yang terhubung dengan sistem pembayaran dan pengolahan sistem restoran

Tabel 3.1 Definisi Aktor

1. Definisi Use Case

Definisi dalam use case diagram adalah kegiatan-kegiatan yang akan terjadi di dalam sistem antara para aktor dengan *use case*.

No	Use Case	Deskripsi	Aktor
1	<i>Login</i>	Admin dan <i>manager</i> yang dapat mengakses login	Admin, <i>manager</i>
2	Pembayaran	Admin dan <i>manager</i> dapat melakukan transaksi pembayaran dan mencetak struk	Admin, <i>manager</i>
3	Mengelola stok kebutuhan	Admin dan <i>manager</i> dapat melihat stok kebutuhan dan dapat meng <i>update</i> stok-stok penjualan	Admin, <i>manager</i>
4	Mencetak	Admin dan <i>manager</i> dapat mencetak	Admin, <i>manager</i>

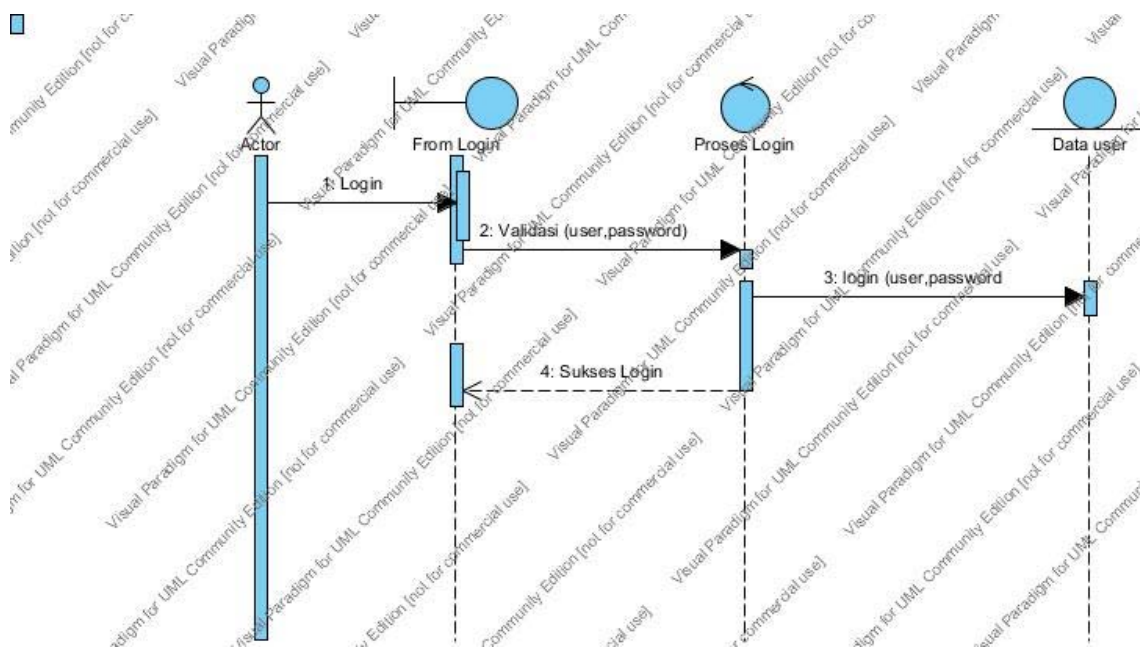
	Laporan Harian	laporan harian	
5	Mengupdate aplikasi	Manager dapat mengupdate aplikasi penjualan seperti harga makanan dan minuman	Manager

Tabel 3.2 Definisi Dalam Use Case Diagram

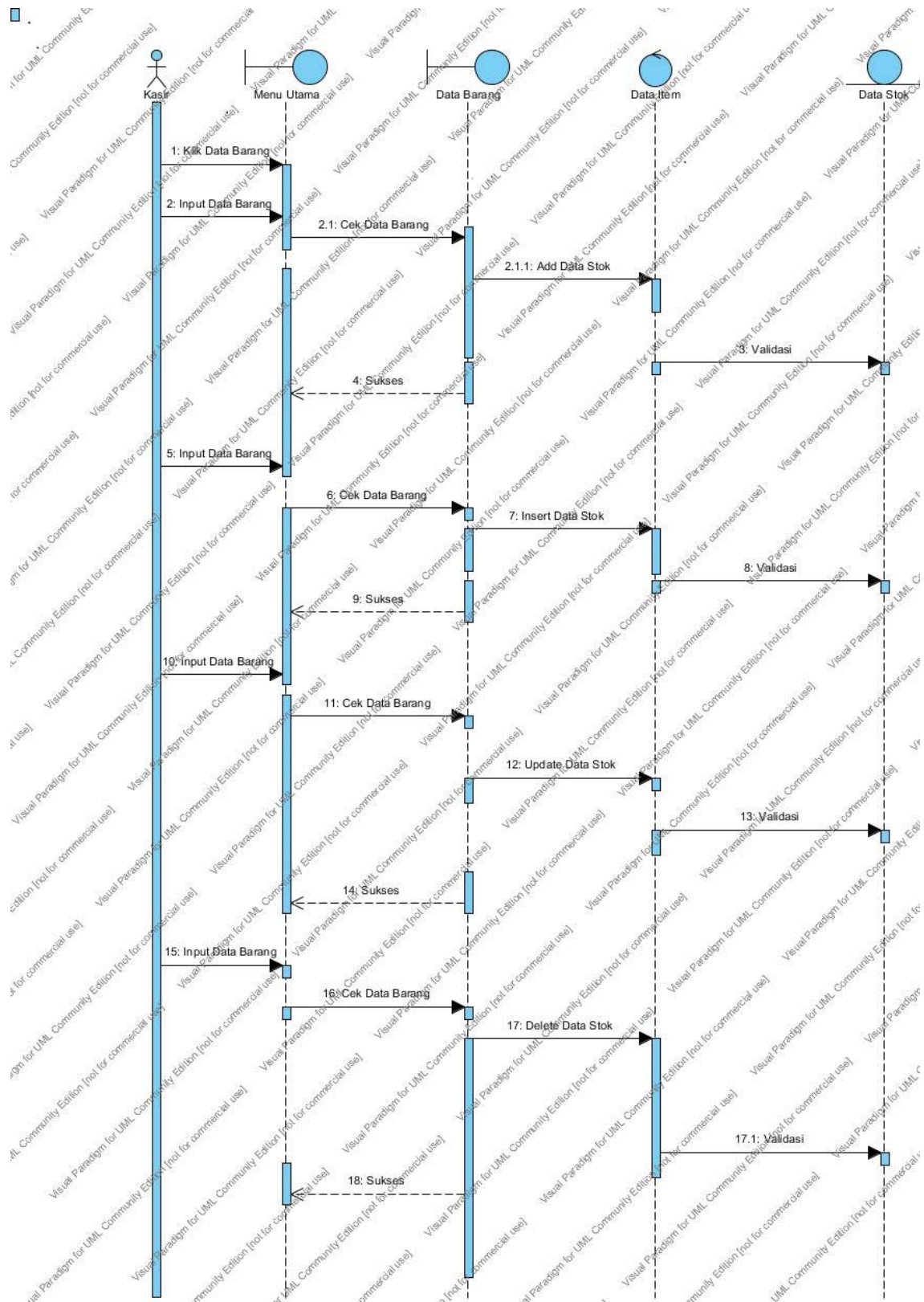
1.3 skenario use case

2. Sequence Diagram

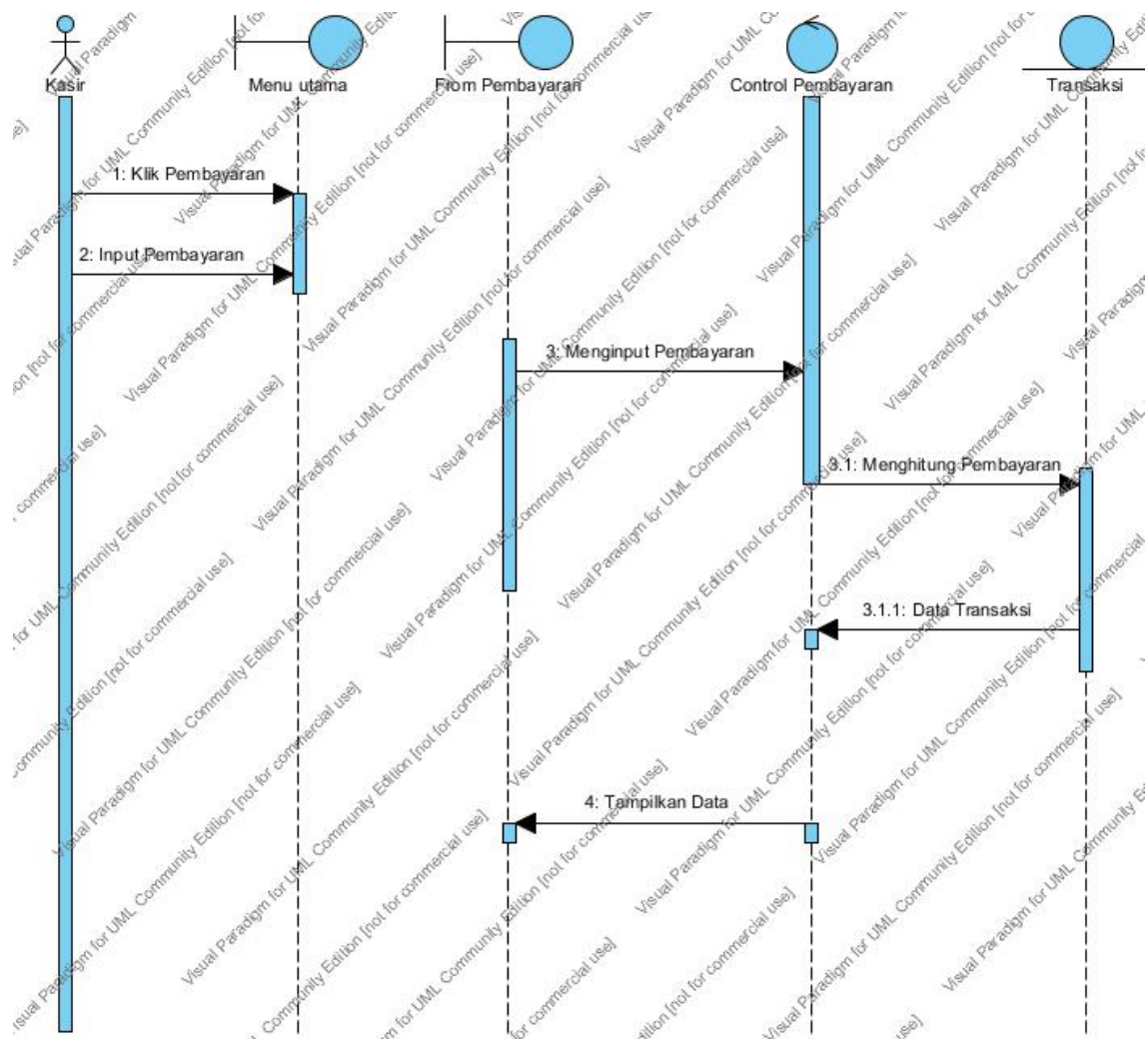
Sequence diagram menjelaskan secara detail urutan proses yang dilakukan dalam sistem untuk mencapai tujuan dari use case: interaksi yang terjadi antar *class*, operasi apa saja yang terlibat, urutan antar operasi, dan informasi yang diperlukan oleh masing-masing operasi



Gambar 3.2 Sequence Login



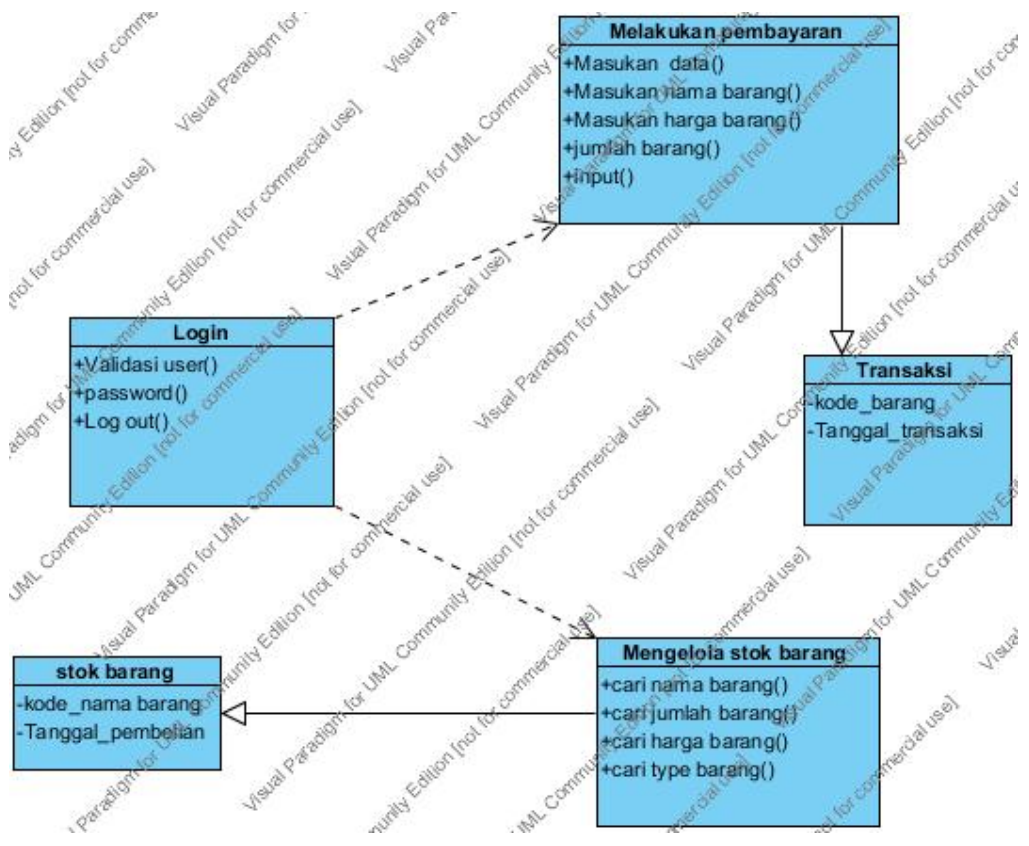
Gambar 3.4 Sequence pengelolaan transaksi data barang



Gambar 3.5 Sequence pengelolaan transaksi Pembayaran

3. Class Diagram

Class diagram adalah sebuah class yang menggambarkan struktur dan penjelasan class, paket, dan objek serta hubungan satu sama lain seperti *containment*, pewarisan, asosiasi, dan lain-lain. *Class diagram* juga menjelaskan hubungan antar *class* dalam sebuah sistem yang sedang dibuat dan bagaimana caranya agar mereka saling berkolaborasi untuk mencapai sebuah tujuan.



BAB IV IMPLEMENTASI

4.1 Membuat class Transaksi Pembayaran

Pada aplikasi ini terdapat 2 kelas yaitu kelas barang dan kelas jual dimana kelas barang merupakan superclass dan kelas jual adalah *subclass*. Pertama buat kelas barang, kelas ini mendeklarasikan kode makanan, nama makanan, harga makanan dan jumlah untuk detilnya bisa dilihat pada gambar *syntax* dibawah ini:

TRANSAKSI PEMBAYARAN

kode barang :

Nama Makanan :

Harga Makanan :

jumlah makanan :

pencarian berdasarkan nama :

Title 1	Title 2	Title 3	Title 4
AP	A.penyet	11000	1
NL	Nasi lemak	10000	2
n.kuning	Nasi kuning	9000	3

Untuk aplikasi pada kelas pengelolaan data dapat dituliskan seperti berikut:

Pengelolaan data

kode makanan :

nama makanan :

harga makanan :

Quantity :

Pencarian berdasarkan nama :

Title 1	Title 2	Title 3	Title 4
ST	Soto	9000	2
MT	Mie tiyau	12000	1
AB	Ayam Bakar	13000	3
PP	Pindang Pedas	15000	

Pada kelas ini terdapat fungsi pengelolaan data dimana akan diperoleh dari harga yang diinputkan user ditambah hasil dari harga makanan juga terdapat fungsi total harga untuk menghitung total harga yang mempermudah user dalam pengimputan data.

4.2 Membuat Tampilan Aplikasi

Selanjutnya membuat design aplikasi dengan menggunakan JFrame Form. Dengan menambahkan 3 buah tombol yaitu tombol proses, tombol reset dan tombol selesai. Juga terdapat beberapa textfield dan juga beberapa label. Terdapat 5 textfield dan 5 label yaitu kode barang, nama barang, harga, quantity, dan total harga.



The image shows a screenshot of a Java Swing application window with a light beige background. The window contains a form with five text input fields and three buttons. The labels and their corresponding text fields are as follows:

Label	Text Field
Kode Barang	jTextField1
Nama Barang	jTextField2
Harga	jTextField3
Quantity	jTextField4
Total	jTextField5

Below the text fields, there are three buttons: "Proses", "Reset", and "Selesai".

Gambar 4.1 Design Tampilan Aplikasi

Ubah beberapa properties seperti dalam table berikut:

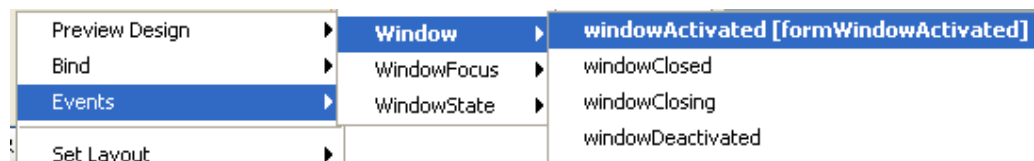
Class	Properti	Value
jFrame	Title	View of Transaction
jTextField1	Editable	Chek
jTextField2	Editable	Check
jTextField3	Editable	Check
jTextField4	Editable	Check
jTextField5	Editable	uncheck
jLabel	Text	Kode Barang
jLabel	Text	Nama Barang
jLabel	Text	Harga
jLabel	Text	Quantity
jLabel	Text	Total
jButton1	Text	Proses
	Mnemonic	P
	ToolTipText	Klik untuk menghitung
jButton2	Text	Reset
	Mnemonic	R
	Tooltiptext	Klik untuk mereset
jButton3	Text	Selesai
	Mnemonic	S
	Tooltiptext	Klik untuk keluar

Table4.1 Properties

4.3 Menambahkan Syntax

Agar aplikasi bisa berjalan sesuai dengan skenario maka pada aplikasi atau program tersebut harus diisi dengan *syntax* yang sesuai. Terdapat beberapa *syntax* yang akan ditambahkan diantaranya memanggil method yang telah dibuat.

Agar pada saat awal aplikasi di *run* semua isi textfield kosong maka yang perlu dilakukan adalah klik kanan pada *frame* lalu pilih *Event* → *Window* → *windowActivated [formWindowActivated]*



Gambar 4.2 WindowActivated

Bol *reset* berfungsi untuk mereset data artinya semua *textfield* kembali pada awal yaitu pada keadaan kosong dan kursor akan berada pada *textfield* paling atas. Fungsi *reset* ini dapat diambil dari *syntax* yang diberikan pada *frame* ketika awal *running*.

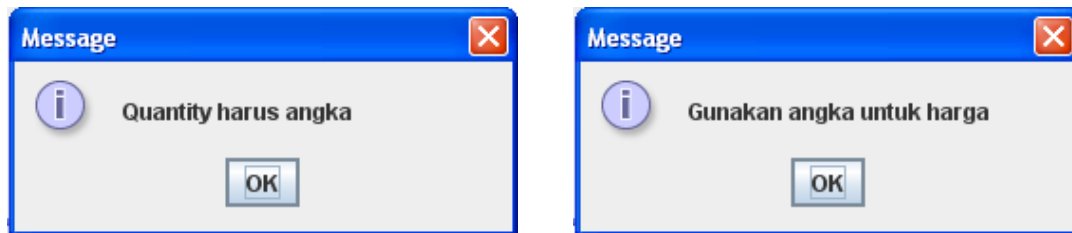
Dan terakhir adalah tombol selesai dimana jika mengklik tombol selesai akan langsung keluar dari aplikasi.

Sekarang semua tombol telah berfungsi tetapi masih terdapat kekurangan pada penginputan.

Maka batasi pengisian pada quantity dan harga hanya bisa diisi dengan angka saja. Caranya adalah klik kanan pada *textfield* harga atau quantity lalu pilih *event* → *key* → *keytype*

Setelah itu beri *syntax* untuk membatasi penginputan hanya bisa mengetik angka. Jika mengetik selain angka maka akan muncul kotak dialog peringatan.

Lakukan hal yang sama pada *textfield* harga. Sekarang penginputan telah dibatasi hanya bisa mengetik angka. Selain angka akan muncul kotak dialog peringatan.



Gambar 4.4 Message peringatan pengisian quantity dan harga

Kini aplikasi perhitungan transaksi telah selesai.



Gambar 4.5 Tampilan akhir aplikasi

BAB V

PENUTUP

5.1 KESIMPULAN

Class adalah merupakan suatu *blueprint* atau cetakan untuk menciptakan suatu instant dari object. *class* juga merupakan grup suatu object dengan kemiripan *attributes/properties, behaviour* dan relasi ke *object* lain. Sedangkan Object adalah merupakan sebuah entiti yang memiliki keadaan, *behaviour* dan identitas yang yang tugasnya dirumuskan dalam suatu lingkup masalah dengan baik. Inilah instance sebenarnya dari sebuah *class*. Ini juga dikenal sebagai *instance*. Pewarisan adalah salah satu ciri pemograman berorientasi objek, yang menyatakan suatu kelas dapat diturunkan lagi menjadi kelas-kelas baru yang lainnya sehingga dapat membentuk sebuah hiraki. Kelas yang merupakan kelas turunan ini bisa disebut dengan kelas anak (*subclass*) dan kelas yang menjadi dasar penurunan disebut kelas orang tua (*superclass*). Pada pemograman java menyediakan kata kunci *extends* yang digunakan untuk proses penurunan terhadap suatu kelas. *Polimorfisme* merupakan kemampuan suatu objek untuk mengungkap banyak hal melalui satu cara yang sama. Polimorfisme merupakan salah satu hal esensial dalam pemograman berorientasi objek karena alasan berikut: yakni *polimorfisme* mengizinkan kelas induk untuk mendefinisikan sebuah method general (bersifat umum) untuk semua kelas turunannya. *Polimorfisme* dapat berupa overloading atau *overriding*.

5.2 SARAN

Dalam Perancangan aplikasi ini masih banyak terdapat kekurangan, oleh karena itu kepada pihak penguji yang memiliki kemampuan lebih untuk dapat menyempurnakan proyek akhir aplikasi ini, penulis mengharapkan kritik dan saran dari penguji untuk bahan pelajaran untuk lebih baik lagi dimasa yang akan datang.

