SURAT PERNYATAAN

Saya yang bertanda tangan dibawah ini :

Nama	: Harry Supriyadi
NPM	: 11.08.036
Tempat Dan Tanggal Lahir	: Bandung, 20 Agustus 1989
Alamat	: Jl. Jati Permai F10-6 RT:04/14 Komp. Pasir Jati Ujung Berung, Bandung

Menyatakan bahwa Laporan Tugas Akhir ini yang berjudul : **"PEMBANGUNAN APLIKASI PENJUALAN RAJUTAN ONLINE DENGAN MENGGUNAKAN CODEIGNITER DI NEKATUN BANDUNG"** merupakan hasil karya saya sendiri. Bila terbukti tidak demikian, saya bersedia menerima sanksi yang berlaku di Universitas Widyatama. Demikian surat pernyataan ini dibuat sebagaimana mestinya dan benar adanya.

Bandung, 15 Juni 2012

Harry Supriyadi

ABSTRAK

Nekatun merupakan usaha yang bergerak dibidang rajutan atau renda. Asal dari kata Nekatun tidak ada arti khusus, hanyalah sebuah nama yang digunakan untuk sebuah hasil kreasi rajutan atau renda yang dibuat dari benang rami dikombinasikan dengan kain blacu menjadi kreasi-kreasi perlengkapan rumah tangga yang unik.

Selama ini proses penyampaian informasi mengenai barang yang dijual menggunakan blog, namun sistem ini masih memiliki kekurangan di mana sistem sebelumnya tidak dapat mengelola data produk dengan baik, tidak adanya perhitungan biaya transaksi secara otomatis, tidak tersedianya form pemesanan untuk memesan secara *online*, tidak adanya tampilan mengenai *record* pemesanan dan belum adanya pembuatan laporan penjualan secara otomatis. Oleh karena itu, pembangunan aplikasi penjualan *online* dirasa tepat untuk mengatasi permasalahan yang dihadapi oleh Nekatun.

Metode yang diterapkan dalam Pembangunan Aplikasi Penjualan *Online* ini yaitu dengan metode *Object Oriented Analysis and Design* dengan menggunakan bahasa pemodelan *Unified Modeling Language (UML)* dan menggunakan *framework* Codeigniter versi 2.0.2 dengan bahasa pemrograman PHP dan MySQL sebagai pengelola basis data. CodeIgniter dipilih karena bersifat gratis (*open source*) dan memiliki model MVC yang membantu programmer dalam mengembangkan aplikasi.

Maka dengan adanya pembangunan aplikasi penjualan *online* ini, Nekatun diharapkan dapat meningkatkan pelayanan yang lebih baik kepada konsumen, mempermudah peng-*input*-an produk baru, mengelola produk yang tersedia, menyediakan form pemesanan secara *online*, menampilkan *record* pemesanan dan membantu dalam pembuatan laporan penjualan.

Kata Kunci : Penjualan Online, Nekatun, Rajutan, CodeIgniter

ABSTRACT

Nekatun is a business engaged in knitting or lace. Origin of the word Nekatun no special meaning, just a name used for the creation of a knitted or crocheted lace made from hemp yarn combined with a calico cloth creations of unique home furnishings.

During this process of delivering information about the goods sold using the blog, but the system still has shortcomings in the previous system which can not properly manage product data, the absence of automatic calculation of transaction costs, lack of order form to order online, no display the record book and not a sales report automatically. Therefore, the development of an online sales application deemed appropriate to address the problems faced by Nekatun.

The method applied in this Online Sales Application Development is a method of Object Oriented Analysis and Design using Unified Modeling Language modeling language (UML) and using the CodeIgniter framework version 2.0.2 with PHP programming language and MySQL as database managers. CodeIgniter was chosen because it is free (open source) and have the MVC model that helps programmers to develop applications.

So with the development of this online sales application, Nekatun expected to enhance better service to consumers, facilitate the input-an attorney-new products, managing products available, providing an online booking form, booking records and displays help in making sales report.

Keywords: Online Sales, Nekatun, Knitwear, CodeIgniter

KATA PENGANTAR

Puji dan syukur kehadirat Allah SWT yang telah memberikan rahmat dan karunia-Nya, sehingga peneliti dapat menyelesaikan laporan Tugas Akhir yang berjudul "**PEMBANGUNAN APLIKASI PENJUALAN RAJUTAN ONLINE DENGAN MENGGUNAKAN CODEIGNITER DI NEKATUN BANDUNG".** Tujuan penyusunan laporan ini adalah untuk memenuhi syarat menyelesaikan Tugas Akhir Jurusan Sistem Informasi Fakultas Teknik Universitas Widyatama Bandung.

Peneliti menyadari dengan sepenuhnya bahwa dalam penyusunan laporan ini masih jauh dari kata sempurna. Hal ini mengingat keterbatasan pengetahuan, pengalaman dan kemampuan peneliti dalam mengolah serta menyajikannya. Namun demikian, peneliti telah berusaha untuk menyusun laporan tugas akhir ini dengan sebaik-baiknya dan peneliti berharap semoga laporan ini dapat bermanfaat bagi semua pihak, khususnya bagi peneliti sendiri.

Selama pelaksanaan dan penyusunan laporan tugas akhir ini peneliti banyak menemui hambatan dan kesulitan. Namun berkat dorongan, bantuan dan bimbingan baik secara moril ataupun material dari berbagai pihak peneliti dapat menyelesaikannya. Oleh karena itu, peneliti ingin menyampaikan rasa terima kasih yang sebesar-besarnya kepada:

1. Allah SWT yang telah memberikan rahmat dan hidayah-Nya kepada peneliti sehingga peneliti dapat menyelesaikan laporan ini.
2. Kedua orang tua, adik dan seluruh keluarga, atas do'a dan dukungan yang terus menerus sehingga peneliti dapat menyelesaikan laporan ini.
3. Sayangku Viona Ameliza, A.Md. yang selalu memberikan dorongan, semangat dan motivasi kepada peneliti.
4. Kawan seperjuangan Deni Imam Hakim, yang telah banyak membantu dan mengingatkan peneliti.

5. Ibu Maniah, Ir., M.T. selaku dosen pembimbing Tugas Akhir dan sekaligus selaku Dosen Wali mahasiswa Jurusan Sistem Informasi angkatan 2008.
6. Ibu Nunu Rismayanti Idris, Ir. selaku pembimbing lapangan di Nekatun.
7. Bapak M. Rozahi Istambul, S.Komp., M.T. sebagai Ketua Jurusan Sistem Informasi Universitas Widyatama.
8. Staff Dosen Universitas Widyatama yang telah memberikan banyak ilmu dan masukan saran kepada peneliti.
9. Bapak Dhani dan Bapak Fauzi yang telah banyak membantu dalam penyelesaian segala administrasi peneliti.
10. Teman-teman Mahasiswa Jurusan Sistem Informasi 2008 yang telah memberikan dukungan dan masukan kepada peneliti.
11. Semua pihak yang terlibat yang telah ikut membantu dalam penulisan laporan ini baik secara langsung maupun tidak langsung.

Akhir kata peneliti mengucapkan Alhamdullilah dan semoga Allah SWT melimpahkan segala Rahmat-Nya serta membalas segala kebaikan kepada semua pihak yang telah membantu dalam menyelesaikan laporan tugas akhir ini. Amin

Bandung, Juni 2012

Peneliti

DAFTAR ISI

LEMBAR JUDUL
LEMBAR PENGESAHAN ..... **i**
SURAT PERNYATAAN ..... **ii**
ABSTRAK ..... **iii**
ABSTRACT ..... **iv**
KATA PENGANTAR ..... **v**
DAFTAR ISI ..... **vii**
DAFTAR TABEL ..... **xi**
DAFTAR GAMBAR ..... **xiv**
DAFTAR KODE PROGRAM ..... **xviii**

BAB I PENDAHULUAN **1**
1.1 Latar Belakang 1
1.2 Identifikasi Masalah 2
1.3 Rumusan Masalah 3
1.4 Maksud dan Tujuan 3
1.5 Batasan Masalah 4
1.6 Metode Penelitian 4
1.7 Sistematika Penulisan 5

BAB II LANDASAN TEORI **6**
2.1 Pengertian Perancangan 6
2.2 Pengertian Penjualan 6
2.3 Definisi Bisnis *Online* 7
2.4 *Website* dan Internet 7
2.4.1 Pengertian *Website* 7

2.4.2 Internet 9
2.5 Bahasa Pemrograman dan Teknik Pemrograman Pembangun Sistem 10
2.5.1 PHP 10
2.5.2 XAMPP 11
2.5.3 Pengenalan CodeIgniter 13
2.6 Metode Pengembangan Sistem 17
2.6.1 Rekayasa Perangkat Lunak 18
2.6.1.1 Definisi Rekayasa Perangkat Lunak 18
2.6.1.2 Tujuan Rekayasa Perangkat Lunak 18
2.6.1.3 Metode Proses Rekayasa Perangkat Lunak 19
2.6.1.4 *Waterfall Process Model* 21
2.6.2 *Object Oriented Programming* (OOP) 23
2.6.3 *Unified Modelling Language* (UML) 24
2.6.3.1 Pengertian UML 24
2.6.3.2 Mengenal *Rational Rose* 24
2.6.3.3 Diagram UML 25

BAB III ANALISIS SISTEM 32
3.1 Tinjauan Organisasi 32
3.1.1 Sejarah Organisasi 32
3.1.2 Struktur Organisasi dan *Job Description* 33
3.2 Analisis Sistem yang Berjalan 33
3.2.1 *Process* 34
3.2.1.1 *Use Case* Sistem Penjualan yang Sedang Berjalan 34
3.2.1.2 Skenario Sistem Penjualan yang Sedang Berjalan 35
3.2.2 *People* 35
3.2.3 *Technology* 36
3.3 Evaluasi Sistem yang Berjalan 37
3.4 Analisis Kebutuhan Sistem 38

3.5 Hasil Akhir Analisis 39
3.6 Analisis Sistem yang Diusulkan 39
3.6.1 *Use Case* Sistem Penjualan yang Diusulkan 39

BAB IV PERANCANGAN SISTEM 41
4.1 Gambaran Umum Sistem yang Diusulkan 41
4.2 Pengembangan Sistem 41
4.2.1 Perancangan Awal Sistem 41
4.2.2 Perancangan Sistem yang Dikembangkan 43
4.2.2.1 Konfigurasi CodeIgniter 43
4.2.2.2 Fungsi Pustaka autentifikasi.php 45
4.2.2.3 Fungsi Pustaka template.php 45
4.2.2.4 Routes Halaman Admin dan Publik 48
4.3 Perancangan Model UML 48
4.3.1 Skenario *Use Case* Aplikasi Penjualan Berbasis *Website* 48
4.3.2 *Activity Diagram* 65
4.3.3 *Sequence Diagram* 75
4.3.4 *Class Diagram* 85
4.3.5 *Entity Relationship Diagram* (ERD) 86
4.3.6 Kamus Data 86
4.4 Perancangan Menu 92
4.5 *Layout* Antarmuka 92
4.5.1 *Layout* Antarmuka *Back Office* 93
4.5.2 *Layout* Antarmuka *Front Office* 98

BAB V IMPLEMENTASI SISTEM 105
5.1 Lingkungan Implementasi 105
5.1.1 Perangkat Kebutuhan Sistem 105
5.1.2 Karakteristik Pengguna 106

5.2 Implementasi Antarmuka 107
5.2.1 Implementasi Antarmuka *Back Office* 107
5.2.2 Implementasi Antarmuka *Front Office* 115
5.3 Pengujian Sistem 122
5.3.1 Metode Pengujian 122
5.3.2 Kasus dan Hasil Pengujian 123
5.3.3 Kesimpulan Hasil Pengujian 130

BAB VI PENUTUP 131
6.1 Kesimpulan 131
6.2 Saran 132

DAFTAR PUSTAKA 133

LAMPIRAN-LAMPIRAN:
LAMPIRAN A : SOURCE CODE PROGRAM
LAMPIRAN B : DAFTAR RIWAYAT HIDUP
LAMPIRAN C : KARTU BIMBINGAN

DAFTAR TABEL

Tabel 2.1 Daftar Simbol *Use Case Diagram* 25
Tabel 2.2 Daftar Simbol *Class Diagram* 27
Tabel 2.3 Daftar Simbol *Statechart Diagram* 27
Tabel 2.4 Daftar Simbol *Actifity Diagram* 28
Tabel 2.5 Daftar Simbol *Sequence Diagram* 29
Tabel 2.6 Daftar Simbol *Collaboration Diagram* 30
Tabel 2.7 Daftar Simbol *Component Diagram* 30
Tabel 2.8 Daftar Simbol *Physical Diagram* 31
Tabel 3.1 Evaluasi Sistem yang Berjalan 37
Tabel 4.1 Perbandingan template yang dikembangkan 46
Tabel 4.2 Skenario *Use Case Login* Admin 49
Tabel 4.3 Skenario *Use Case Login User* 49
Tabel 4.4 Skenario *Use Case* Menambah Ketegori 50
Tabel 4.5 Skenario *Use Case* Mengubah Ketegori 51
Tabel 4.6 Skenario *Use Case* Menghapus Ketegori 52
Tabel 4.7 Skenario *Use Case* Menambah Produk 53
Tabel 4.8 Skenario *Use Case* Mengubah Produk 54
Tabel 4.9 Skenario *Use Case* Menghapus Produk 55
Tabel 4.10 Skenario *Use Case* Registrasi *User* 55
Tabel 4.11 Skenario *Use Case* Mengubah Profil *User* 56
Tabel 4.12 Skenario *Use Case* Mengubah Akun *User* 57
Tabel 4.13 Skenario *Use Case* Menghapus *User* 58
Tabel 4.14 Skenario *Use Case* Status Pemesanan 59
Tabel 4.15 Skenario *Use Case* Mengubah Halaman Depan 60
Tabel 4.16 Skenario *Use Case* Homepage 60
Tabel 4.17 Skenario *Use Case* Contact 61

Tabel 4.18 Skenario *Use Case* Cara Pemesanan 62
Tabel 4.19 Skenario *Use Case Record* Pemesanan 62
Tabel 4.20 Skenario *Use Case* Galeri Produk 63
Tabel 4.21 Skenario *Use Case* Pemesanan 64
Tabel 4.22 Struktur tabel foto_produk 87
Tabel 4.23 Struktur tabel kategori 87
Tabel 4.24 Struktur tabel option 88
Tabel 4.25 Struktur tabel order 88
Tabel 4.26 Struktur tabel order_data 89
Tabel 4.27 Struktur tabel produk 90
Tabel 4.28 Struktur tabel user 90
Tabel 4.29 Struktur tabel user_data 91
Tabel 5.1 Perangkat kebutuhan sistem 105
Tabel 5.2 Karakteristik pengguna 106
Tabel 5.3 Pengujian *login* admin 123
Tabel 5.4 Pengujian menambah kategori 123
Tabel 5.5 Pengujian mengubah kategori 124
Tabel 5.6 Pengujian menghapus kategori 124
Tabel 5.7 Pengujian menambah produk baru 124
Tabel 5.8 Pengujian menambah gambar produk 125
Tabel 5.9 Pengujian mengubah data produk 125
Tabel 5.10 Pengujian menghapus produk 125
Tabel 5.11 Pengujian menambah user 126
Tabel 5.12 Pengujian mengubah user 126
Tabel 5.13 Pengujian menghapus user 126
Tabel 5.14 Pengujian mengubah status pesanan 127
Tabel 5.15 Pengujian mengubah pengaturan situs 127
Tabel 5.16 Pengujian registrasi 127

Tabel 5.17 Pengujian *login user* 128
Tabel 5.18 Pengujian mengubah profil *user* 128
Tabel 5.19 Pengujian mengubah akun *user* 129
Tabel 5.20 Pengujian menambah kantong belanja 129
Tabel 5.21 Pengujian *update* kantong belanja 129
Tabel 5.22 Pengujian transaksi pemesanan 130

DAFTAR GAMBAR

Gambar 2.1 Cara kerja *framework* CodeIgniter 13
Gambar 2.2 MVC pada halaman web 16
Gambar 2.3 Struktur file CodeIgniter 17
Gambar 2.4 Tujuan RPL 19
Gambar 2.5 Waterfall Model menurut Sommerville 22
Gambar 3.1 Struktur Organisasi Nekatun Bandung 33
Gambar 3.2 Ruang lingkup yang akan dianalisis 34
Gambar 3.3 *Use Case Diagram* Sistem Penjualan Berjalan 34
Gambar 3.4 *Use Case Diagram* Sistem Penjualan yang Diusulkan 40
Gambar 4.1 Struktur Direktori CodeIgniter 42
Gambar 4.2 Halaman Awal CodeIgniter 43
Gambar 4.3 Template palmtrees 46
Gambar 4.4 *Activity Diagram login* admin 65
Gambar 4.5 *Activity Diagram login user* 66
Gambar 4.6 *Activity Diagram* menambah kategori 66
Gambar 4.7 *Activity Diagram* mengubah kategori 67
Gambar 4.8 *Activity Diagram* menghapus kategori 67
Gambar 4.9 *Activity Diagram* menambah produk 68
Gambar 4.10 *Activity Diagram* mengubah produk 68
Gambar 4.11 *Activity Diagram* menghapus produk 69
Gambar 4.12 *Activity Diagram* registrasi *user* 69
Gambar 4.13 *Activity Diagram* mengubah profil *user* 70
Gambar 4.14 *Activity Diagram* mengubah akun *user* 70
Gambar 4.15 *Activity Diagram* menghapus *user* 71
Gambar 4.16 *Activity Diagram* status pemesanan 71
Gambar 4.17 *Activity Diagram* mengubah halaman depan 72
Gambar 4.18 *Activity Diagram* homepage 72

Gambar 4.19 *Activity Diagram* contact 73
Gambar 4.20 *Activity Diagram* cara pemesanan 73
Gambar 4.21 *Activity Diagram record* pemesanan 73
Gambar 4.22 *Activity Diagram* galeri produk 74
Gambar 4.23 *Activity Diagram* pemesanan 74
Gambar 4.24 *Sequence Diagram login* admin 75
Gambar 4.25 *Sequence Diagram login user* 76
Gambar 4.26 *Sequence Diagram* menambah kategori 76
Gambar 4.27 *Sequence Diagram* mengubah kategori 77
Gambar 4.28 *Sequence Diagram* menghapus kategori 77
Gambar 4.29 *Sequence Diagram* menambah produk 78
Gambar 4.30 *Sequence Diagram* mengubah produk 79
Gambar 4.31 *Sequence Diagram* menghapus produk 79
Gambar 4.32 *Sequence Diagram* registrasi *user* 80
Gambar 4.33 *Sequence Diagram* mengubah profil *user* 80
Gambar 4.34 *Sequence Diagram* mengubah akun *user* 81
Gambar 4.35 *Sequence Diagram* menghapus *user* 81
Gambar 4.36 *Sequence Diagram* status pemesanan 82
Gambar 4.37 *Sequence Diagram* mengubah halaman depan 82
Gambar 4.38 *Sequence Diagram* homepage 83
Gambar 4.39 *Sequence Diagram* contact 83
Gambar 4.40 *Sequence Diagram* cara pemesanan 83
Gambar 4.41 *Sequence Diagram record* pemesanan 84
Gambar 4.42 *Sequence Diagram* galeri produk 84
Gambar 4.43 *Sequence Diagram* pemesanan 85
Gambar 4.44 *Class Diagram* 85
Gambar 4.45 *Entity Relationship Diagram* 86
Gambar 4.46 Perancangan Menu 92
Gambar 4.47 Antarmuka form *login* admin 93

Gambar 4.48 Antarmuka halaman dashboard 93
Gambar 4.49 Antarmuka admin kategori 94
Gambar 4.50 Antarmuka tambah kategori 94
Gambar 4.51 Antarmuka admin produk 95
Gambar 4.52 Antarmuka tambah produk baru 95
Gambar 4.53 Antarmuka tambah gambar produk 96
Gambar 4.54 Antarmuka *User* 96
Gambar 4.55 Antarmuka Laporan 97
Gambar 4.56 Antarmuka ubah status pemesanan 97
Gambar 4.57 Antarmuka pengaturan situs 98
Gambar 4.58 Antarmuka homepage 98
Gambar 4.59 Antarmuka registrasi *user* 99
Gambar 4.60 Antarmuka *login user* 99
Gambar 4.61 Antarmuka cara pemesanan 100
Gambar 4.62 Antarmuka kontak 100
Gambar 4.63 Antarmuka profil *user* 101
Gambar 4.64 Antarmuka *record* 101
Gambar 4.65 Antarmuka edit akun 102
Gambar 4.66 Antarmuka produk 102
Gambar 4.67 Antarmuka detail produk 103
Gambar 4.68 Antarmuka kantong belanja 103
Gambar 4.69 Antarmuka transaksi pemesanan 104
Gambar 5.1 Antarmuka *login* admin 107
Gambar 5.2 Antarmuka halaman dashboard 108
Gambar 5.3 Antarmuka admin kategori 108
Gambar 5.4 Antarmuka tambah kategori 109
Gambar 5.5 Antarmuka admin produk 109
Gambar 5.6 Antarmuka tambah produk 110
Gambar 5.7 Antarmuka tambah gambar produk 110

Gambar 5.8 Antarmuka mengubah produk 111
Gambar 5.9 Antarmuka *User* 111
Gambar 5.10 Antarmuka tambah *User* 112
Gambar 5.11 Antarmuka admin mengubah profil *User* 112
Gambar 5.12 Antarmuka admin mengubah akun *User* 113
Gambar 5.13 Antarmuka Laporan 113
Gambar 5.14 Antarmuka mengubah status pesanan 114
Gambar 5.15 Antarmuka pengaturan situs 114
Gambar 5.16 Antarmuka homepage 115
Gambar 5.17 Antarmuka registrasi *user* 116
Gambar 5.18 Antarmuka *login user* 116
Gambar 5.19 Antarmuka cara pemesanan 117
Gambar 5.20 Antarmuka kontak 118
Gambar 5.21 Antarmuka profil *user* 118
Gambar 5.22 Antarmuka rekaman belanja 119
Gambar 5.23 Antarmuka mengubah akun *user* 119
Gambar 5.24 Antarmuka galeri produk 120
Gambar 5.25 Antarmuka detail produk 120
Gambar 5.26 Antarmuka kantong belanja 121
Gambar 5.27 Antarmuka transaksi pemesanan 121

DAFTAR KODE PROGRAM

Kode Program 1. config.php 43
Kode Program 2. database.php 44
Kode Program 3. autoload.php 44
Kode Program 4. config_store.php 44
Kode Program 5. routes.php untuk halaman admin 48
Kode Program 6. routes.php untuk halaman publik 48

BAB I
PENDAHULUAN

1.1 Latar Belakang Masalah

Banyaknya perubahan-perubahan dari tahun ke tahun, baik dalam teknologi maupun non-teknologi yang dapat menjadikan negara maju dan berkembang sehingga tanpa adanya perubahan maka negara tidak akan bisa berkembang seperti negara-negara lainnya. Komputer merupakan salah satu bentuk teknologi yang perkembangannya dalam hitungan hari saja. Bukan hanya perusahaan-perusahaan saja yang merasakan perkembangannya, melainkan masyarakat luas sudah dapat memanfaatkan teknologi ini. Hal ini disebabkan karena pentingnya suatu informasi.

Teknologi informasi merupakan salah satu wadah informasi yang sedang berkembang dengan pesat. Maka dengan kemajuan teknologi informasi, pengaksesan terhadap data atau informasi yang tersedia berlangsung dengan cepat, efisien dan akurat. Berbagai cara yang dapat dilakukan dalam mencapai keberhasilan dan kesuksesan suatu perusahaan. Strategi yang diterapkan perusahaan untuk tiap-tiap bidang di dalamnya pun berbeda-beda, mulai dari bagian produksi, distribusi, penjualan, maupun promosinya.

Seiring dengan pesatnya perkembangan teknologi informasi, terutama teknologi web, berkembang pula *framework* dan lingkungan pengembangan aplikasi berbasis web. *Framework* adalah kerangka kerja atau dapat diartikan sebagai kunmpulan script (terutama *class* dan *function*) yang membantu developer atau programmer dalam menangani berbagai masalah dalam pemrograman. Terdapat banyak framework yang digunakan dalam pengembangan aplikasi berbasis web diantaranya yaitu CodeIgniter. CodeIgniter merupakan sebuah *framework* yang digunakan untuk membuat sebuah aplikasi berbasis web yang disusun dengan menggunakan bahasa PHP. CodeIgniter bersifat *free opensource software* di mana setiap orang berhak menggunakannya tanpa harus dikenai biaya lisensi.

Teknologi informasi sangatlah penting dalam segala bidang usaha baik besar maupun usaha kecil, di mana teknologi yang sering digunakan untuk meningkatkan usaha yang salah satunya yaitu sistem informasi penjualan berbasis *website* sebagai sarana promosi dan penjualan. Akan tetapi masih banyak pengusaha yang belum menggunakan teknologi tersebut, maka dari itu sebagai mahasiswa program studi Sistem Informasi Universitas Widyatama peneliti berinisiatif untuk membuatkan aplikasi penjualan berbasis *website* di perusahaan Nekatun.

Nekatun merupakan usaha yang bergerak dibidang rajutan atau renda yang dibuat dari benang rami dikombinasikan dengan kain blacu yang menjadikan kreasi-kreasi perlengkapan rumah tangga yang unik. Selama ini proses penyampaian informasi mengenai barang yang dijual menggunakan blog, namun sistem ini masih memiliki kekurangan dari pengelolaan data produk dan pemesanan sehingga timbulnya permasalahan tentang pengelolaan data produk dan pemesanan di perusahaan tersebut.

Jadi dengan adanya aplikasi ini, pengguna dapat dengan mudah memperoleh pelayanan dan informasi, serta memudahkan *admin* dalam mengelola data produk dan pemesanan. Aplikasi ini dibuat dengan menggunakan *framework* CodeIgniter dengan bahasa pemrograman PHP dan MySQL sebagai basis datanya.

Dengan demikian berdasarkan latar belakang di atas, maka peneliti bermaksud membuat Tugas Akhir dengan mengajukan judul: "PEMBANGUNAN APLIKASI PENJUALAN RAJUTAN ONLINE DENGAN MENGGUNAKAN CODEIGNITER DI NEKATUN BANDUNG".

1.2 Identifikasi Masalah

Berdasarkan penelitian dan pengamatan peneliti, maka yang menjadi identifikasi masalah adalah sebagai berikut :

1. Nekatun membutuhkan sistem yang memudahkan dalam peng-*input*-an data produk baru dan mengelola produk yang tersedia.
2. Nekatun membutuhkan sistem yang dapat melakukan perhitungan biaya pemesanan secara otomatis.

3. Nekatun membutuhkan sistem yang memudahkan bagi konsumen dalam melakukan pemesanan secara langsung dan menyediakan informasi mengenai status pesanan konsumen.
4. Nekatun memerlukan sistem pembuatan laporan penjualan secara otomatis.

1.3 Rumusan Masalah

Berdasarkan identifikasi masalah di atas, maka yang menjadi rumusan masalah adalah sebagai berikut:

1. Bagaimana membangun sebuah sistem informasi penjualan *online* dengan menggunakan *framework* CodeIgniter?
2. Bagaimana membangun sistem yang dapat mengelola data produk dengan baik, menghitung biaya transaksi secara otomatis dan pembuatan laporan penjualan secara otomatis?
3. Bagaimana upaya untuk membangun sebuah sistem yang memudahkan dalam peng-*input*-an data produk baru, mengelola produk yang tersedia, dapat melakukan perhitungan biaya pemesanan, dapat melakukan pemesanan secara langsung, menyediakan informasi mengenai status pesanan konsumen dan pembuatan laporan penjualan secara otomatis?

1.4 Maksud dan Tujuan

Maksud dan tujuan dari pembangunan aplikasi bisnis *online* dalam Tugas Akhir ini adalah :

1. Meningkatkan pelayanan yang lebih baik kepada konsumen, di mana konsumen diharapkan dapat mengakses secara *online* untuk mendapatkan informasi tentang produk dan pemesanan secara langsung.
2. Dapat mempermudah peng-*input*-an data produk baru dan mengelola produk yang tersedia.
3. Konsumen dapat mengontrol pesanannya melalui *record* pemesanan.
4. Memudahkan Nekatun dalam perekapan laporan penjualan.
5. Mengefisiensikan biaya promosi penjualan produk Nekatun.

1.5 Batasan Masalah

Batasan masalah dalam pembangunan aplikasi penjualan berbasis *website* ini adalah :

1. Sistem yang akan dibangun yaitu sebagai media informasi dan promosi.
2. Sistem ini dapat digunakan sebagai pengelola data produk dan pengelola data pemesanan, akan tetapi mekanisme pembayaran masih melalui transfer antar Bank.
3. Pada transaksi penjualan tidak mempengaruhi perhitungan stok.
4. Konten *website* menampilkan halaman beranda, profil perusahaan, katalog produk, kantong belanja, cara pemesanan dan status pemesanan.
5. Aplikasi ini dikembangkan dengan menggunakan *framework* CodeIgniter versi 2.0.2 dengan bahasa pemrograman PHP dan MySQL sebagai pengelola basis data.
6. Untuk pembuatan desain tampilan dan perancangan sistem menggunakan Adobe Dreamweaver CS5.

1.6 Metodologi Penelitian

Metode pengumpulan data yang digunakan peneliti sebagai penunjang dalam penyusunan Tugas Akhir adalah sebagai berikut :

1. Studi Literatur

   Teknik pengumpulan data dilakukan dengan cara membaca buku-buku, dokumen-dokumen, diktat dan referensi-referensi yang kaitannya dengan judul penelitian.
2. Studi Lapangan

   Pengumpulan data dilakukan dengan cara observasi dan wawancara langsung.

   a. Pengamatan (*Observasi*)

      Teknik pengumpulan data dengan mengadakan penelitian dan peninjauan langsung terhadap permasalah yang diambil.

b. Wawancara (*Interview*)

Teknik pengumpulan data yang digunakan peneliti untuk mendapatkan keterangan-keterangan lisan melalui bercakap-cakap dan berhadapan muka dengan orang yang dapat memberikan keterangan kepada peneliti.

Metode perancangan sistem yang digunakan dalam pembangunan aplikasi ini menggunakan model rekayasa perangkat lunak yaitu *waterfall model*.

1.7 Sistematika Penulisan

Adapun sistematika penulisan Tugas Akhir ini akan diuraikan menjadi beberapa bab yang tersusun sebagai berikut :

BAB I PENDAHULUAN

Bab ini menjelaskan mengenai latar belakang masalah, identifikasi masalah, rumusan masalah, maksud dan tujuan, batasan masalah, metodologi penelitian, rencana aktivitas dan sistematika penulisan.

BAB II LANDASAN TEORI

Bab ini membahas mengenai teori-teori yang digunakan oleh peneliti untuk membangun aplikasi penjualan.

BAB III ANALISIS SISTEM

Pada bab ini membahas mengenai analisa sistem yang sudah ada, evaluasi dan solusi sistem yang akan dibuat.

BAB IV PERANCANGAN SISTEM

Bab ini membahas mengenai batasan dan kebutuhan sistem baru, perancangan data, perancangan arsitektural, serta perancangan antarmuka.

BAB V IMPLEMENTASI SISTEM

Berisi tentang implementasi hasil desain pada bab empat dan penyesuaian kebutuhan sistem agar sistem berjalan dengan optimal.

BAB VI PENUTUP

Bab ini menguraikan tentang kesimpulan keseluruhan dari penyusunan tugas akhir dan saran–saran yang diperoleh dari hasil tugas akhir untuk penelitian berikutnya.

BAB II
LANDASAN TEORI

Bab ini akan memaparkan berbagai landasan teori yang mendukung pelaksanaan Tugas Akhir. Landasan teori ini akan memberikan pemahaman yang berhubungan dengan judul Tugas Akhir yang diajukan.

2.1 Pengertian Perancangan

Menurut Pressman perancangan adalah suatu langkah pertama dalam fase pengembangan rekayasa produk atau sistem dalam proses penerapan berbagai teknik dan prinsip yang bertujuan untuk mendefinisikan sebuah peralatan, satu proses atau satu sistem secara detail yang membolehkan dilakukan realisasi fisik.

Fase ini adalah inti dari proses perangkat lunak sehingga elemen-elemen dari model analisa dikonversikan. Dengan menggunakan satu dari sejumlah metode perancangan, fase perancangan akan menghasilkan perancangan data, perancangan antarmuka, perancangan arsitektur dan perancangan prosedur. Dalam perancangan perangkat lunak dapat menggambarkan struktur data, struktur program, karakteristik antarmuka dan detail prosedur yang merupakan sintesa dari keperluan-keperluan informasi. [2]

2.2 Pengertian Penjualan

Pengertian penjualan menurut Sutanto adalah “usaha yang dilakukan oleh manusia untuk menyampaikan barang kebutuhan yang dihasilkannya kepada mereka yang membutuhkan dengan imbalan uang menurut harga yang telah disepakati.”

Menurut Philip Kotler yang dialih bahasakan oleh Benyamin Molan penjualan adalah “menjual lebih banyak barang kepada lebih banyak orang untuk membeli suatu barang yang ditawarkan perusahaan yang bertujuan untuk memperoleh laba.”

Berdasarkan kedua Pengertian di atas dapat diartikan bahwa penjualan adalah usaha kegiatan menyampaikan barang kebutuhan yang dihasilkan produsen ke

konsumen dengan harga yang telah disepakati yang bertujuan mengarahkan pembeli untuk membeli produk. [9]

2.3 Definisi Bisnis *Online*

Bisnis *online* sering kali diasumsikan oleh internet marketer bahwa semua bisnis yang dilakukan secara *online* dengan menggunakan media internet sebagai media pemasaran produk dan jasa adalah mencakup bagian-bagian dari bisnis *online*. Jadi dalam hal ini internet adalah media karena internet sendiri yang terdiri dari ribuan-ribuan halaman bahkan jutaan halaman *web* mampu menyediakan informasi bagi pengguna internet.

Jadi bisnis online merupakan lahan bisnis yang tak terbatas. Cakupannya bahkan bisa mencapai seluruh penjuru dunia, menjelajah seluruh benua dan melampaui semua samudera. Praktisnya bisnis online sring kali tidak membutuhkan modal sebagaimana bisnis lainnya bahkan kerjanya sederhana hanya membutuhkan media untuk menyambung internet dan cukup bekerja di balik layar computer atau laptop. [3]

2.4 *Website* dan Internet

2.4.1 Pengertian *Website*

Website dapat di artikan sebagai kumpulan halaman- halaman yang digunakan untuk menampilkan informasi teks, gambar diam atau gerak, animasi, suara, dan atau gabungan dari semuanya itu yang bersifat statis maupun dinamis yang membentuk satu rangkaian bangunan yang saling terkait dimana masing-masing dihubungkan dengan jaringan-jaringan halaman (*hyperlink*).

Unsur-unsur penunjang *website* adalah sebagai berikut:

1. Nama Domain (*Domain name*/URL – *Uniform Resource Locator*)

   Pengertian Nama Domain atau bisa disebut dengan *Domain Name* atau URL adalah alamat yang digunakan untuk mengidentifikasi sebuah *website* pada dunia internet. Nama domain diperjualbelikan secara bebas di internet dengan status

sewa tahunan serta Nama Domain sendiri mempunyai identifikasi ekstensi/akhiran sesuai dengan kepentingan dan lokasi keberadaan *website* tersebut.

2. Rumah Tempat *Website* (*Web Hosting*)

   Pengertian *Web Hosting* dapat dartikan sebagai ruangan yang terdapat dalam harddisk tempat menyimpan berbagai data, file- file, gambar dan lain sebagainya yang akan ditampilkan di *website*.

   Besarnya data yang bisa dimasukkan tergantung dari besarnya *web hosting* yang disewa/ dipunyai, semakin besar *web hosting* semakin besar pula data yang dapat dimasukkan dan ditampilkan dalam *website*. *Web Hosting* juga diperoleh dengan menyewa. Besarnya *hosting* ditentukan ruangan harddisk dengan ukuran MB (Mega Byte) atau GB (Giga Byte). Lama penyewaan *web hosting* rata-rata dihitung per-tahun dan sering dilakukan dari perusahaan-perusahaan penyewa *web hosting* yang banyak dijumpai baik di Indonesia maupun luar negeri.

3. Bahasa Program (*Scripts Program*)

   Bahasa Program yaitu bahasa yang digunakan untuk menerjemahkan setiap perintah dalam *website* yang pada saat diakses. Jenis bahasa program sangat menentukan statis, dinamis atau interaktifnya sebuah *website*. Semakin banyak ragam bahasa program yang digunakan maka akan terlihat *website* semakin dinamis dan interaktif serta terlihat bagus.

   Jenis-jenis bahasa program yang banyak dipakai para desainer *website* antara lain: HTML, ASP, PHP, JSP, Java Scripts, Java Applets dsb. Bahasa dasar yang dipakai setiap situs adalah HTML sedangkan PHP,ASP, JSP dan lainnya merupakan bahasa pendukung yang bertindak sebagai pengatur dinamis, dan interaktifnya situs. Bahasa Program ini biasanya digunakan untuk membangun portal berita, artikel, forum diskusi, buku tamu, anggota organisasi, email, mailing list dan lain sebagainya yang memerlukan *update* setiap saat.

4. Desain *Website*

   Unsur *website* yang paling penting dan utama adalah desain karena desain *website* dapat menentukan kualitas dan keindahan sebuah *website* serta sangat berpengaruh kepada penilaian pengunjung akan bagus tidaknya sebuah *website*.

   Untuk membuat *website* biasanya dapat dilakukan sendiri atau menyewa jasa *website designer*. Saat ini banyak jasa *web designer*, terutama dikota-kota besar. Semakin banyak penguasaan *web designer* tentang beragam program/*software* pendukung pembuatan situs maka akan dihasilkan situs yang semakin berkualitas demikian pula sebaliknya. Jasa *web designer* ini yang umumnya memerlukan biaya yang tertinggi dari seluruh biaya pembangunan situs dan semuanya itu tergantung kualitas designer. [11]

2.4.2 Internet

Internet merupakan hubungan antar berbagai jenis komputer dan jaringan di dunia yang berbeda sistem operasi maupun aplikasinya di mana hubungan tersebut memanfaatkan kemajuan media komunikasi (telepon dan satelit) yang menggunakan protokol standar dalam berkomunikasi yaitu protokol TCP/IP yang berisikan informasi dan sebagai sarana komunikasi data yang berupa suara, gambar, video dan juga teks.

Dunia dewasa ini menjadi semakin terhubung (*interconnected*) satu dengan lainnya. Dengan kata lain kita dapat menggunakan telepon genggam untuk menghubungi teman yang berada dibelahan dunia lain. Kita dapat menonton siaran langsung pertandingan bola piala dunia yang berlangsung di benua lain. Atau bahkan kita dapat berinvestasi dalam ekonomi dunia karena pasar modal di Tokyo, London dan New York terhubung secara elektronis selama 24 jam sehari. Keterhubungan inilah yang disebut Interconnected Network atau lebih populer dengan sebutan internet.

Satu hal terbesar tentang internet adalah bahwa tak seorangpun yang benar-benar memilikinya. Internet adalah kumpulan global dari orang-orang dan jaringan

komputer besar dan kecil, di mana semuanya tersambung oleh ribuan kilometer kabel dan *line* telepon yang semuanya dapat saling berkomunikasi. Beberapa fungsi internet antara lain: menemukan dan mengakses informasi, kursus/sekolah jarak jauh, saling bertukar informasi dengan rekan bisnis dan hobi, atau mendapatkan pekerjaan secara *on-line* dan lainnya. [1]

2.5 Bahasa Pemrograman dan Teknik Pemrograman Pembangun Sistem

Dalam membangun aplikasi penjualan ini penulis menggunakan bahasa pemrograman PHP dengan *tools* paket *database* XAMPP dan *framework* CodeIgniter.

2.5.1 PHP

Pada awalnya PHP merupakan kependekan dari Personal Home Page (Situs Personal). PHP pertama kali dibuat oleh Rasmus Lerdorf pada tahun 1995. Pada waktu itu PHP masih bernama FI (*Form Interpreted*), yang wujudnya berupa sekumpulan *script* yang digunakan untuk mengolah data form dari *web*.

Selanjutnya Rasmus merilis kode sumber tersebut untuk umum dan menamakannya PHP/FI. Dengan perilisan kode sumber ini menjadi *opensource*, maka banyak programmer yang tertarik untuk ikut mengembangkan PHP.

Pada November 1997, dirilis PHP/FI 2.0. Pada rilis ini interpreter PHP sudah diimplementasikan dalam program C. Dalam rilis ini disertakan juga modul-modul ekstensi yang meningkatkan kemampuan PHP/FI secara signifikan.

Pada tahun 1997, sebuah perusahaan bernama Zend menulis ulang interpreter PHP menjadi lebih bersih, lebih baik dan lebih cepat. Kemudian pada Juni 1998, perusahaan tersebut merilis interpreter baru untuk PHP dan meresmikan rilis tersebut sebagai PHP 3.0 dan singkatan PHP dirubah menjadi akronim berulang "*PHP: Hypertext Preprocessing*".

PHP adalah bahasa pemrograman script yang paling banyak dipakai saat ini. PHP banyak dipakai untuk memrogram situs web dinamis, walaupun tidak tertutup kemungkinan digunakan untuk pemakaian lain.

Kelebihan PHP dari bahasa pemrograman lain yaitu:

1. Bahasa pemrograman PHP adalah sebuah bahasa *script* yang tidak melakukan sebuah kompilasi dalam penggunaanya.
2. *Web Server* yang mendukung PHP dapat ditemukan di mana-mana.
3. Dalam sisi pengembangan lebih mudah, karena banyaknya milis-milis dan developer yang siap membantu dalam pengembangan.
4. Dalam sisi pemahamanan, PHP adalah bahasa *scripting* yang paling mudah karena memiliki referensi yang banyak.
5. PHP adalah bahasa *open source* yang dapat digunakan di berbagai mesin (Linux, Unix, Macintosh, Windows) dan dapat dijalankan secara *runtime* melalui *console* serta juga dapat menjalankan perintah-perintah sistem. [7]

Namun ada kekurangan dari PHP ini adalah:

1. Tidak detail untuk pengembangan skala besar.
2. Tidak memiliki sistem pemrograman berorientasi objek yang sesungguhnya.
3. Tidak bisa memisahkan antara tampilan dan logic dengan baik.
4. PHP memiliki kelemahan security tertentu apabila programmer tidak teliti dalam melakukan pemograman dan kurang memperhatikan isu konfigurasi PHP.
5. Kode PHP dapat dibaca semua orang dan kompilasi hanya dapat dilakukan dengan tool yang mahal dari Zend. [15]

2.5.2 XAMPP

XAMPP merupakan singkatan dari X (empat sistem operasi apapun), Apache, MySQL, PHP, Perl. XAMPP merupakan tool yang menyediakan paket perangkat lunak ke dalam satu buah paket. Dalam paketnya sudah terdapat Apache (*web server*), MySQL (*database*), PHP (*server side scripting*), Perl, FTP server, phpMyAdmin dan berbagai pustaka bantu lainnya.

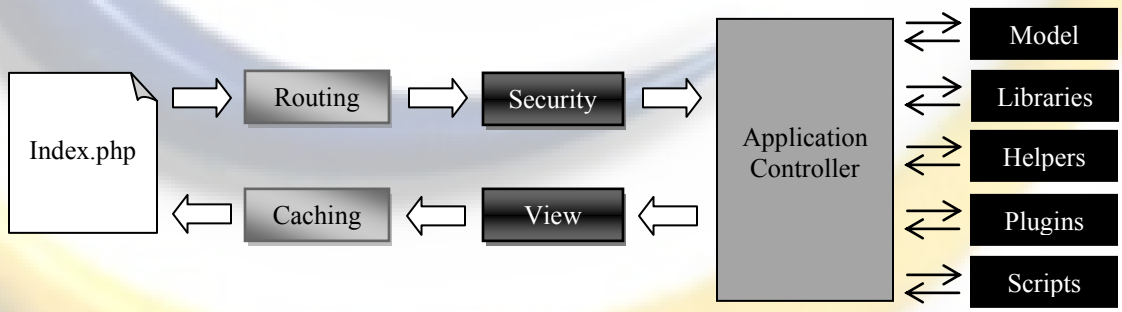
Berikut ini penjelasan tentang *web server* Apache, PHP, MySQL, phpMyAdmin dan perl:

1. Apache bersifat open source, artinya setiap orang boleh menggunakannya, mengambil dan bahkan mengubah kode programnya. Tugas utama apache adalah menghasilkan halaman web yang benar kepada peminta, berdasarkan kode PHP yang dituliskan oleh pembuat halaman web.
2. PHP, Bahasa pemrograman PHP merupakan bahasa pemrograman untuk membuat web yang bersifat *server-side scripting*. PHP memungkinkan kita untuk membuat halaman web yang bersifat dinamis. PHP dapat memanfaatkan *database* untuk menghasilkan halaman web yang dinamis. Sistem manajemen *database* yang sering digunakan bersama PHP adalah MySQL.
3. MySQL, pengembangannya disebut SQL yang merupakan kepanjangan dari *Structured Query Language*. SQL merupakan bahasa terstruktur yang khusus digunakan untuk mengolah *database*. SQL pertama kali didefinisikan oleh American National Standards Institue (ANSI) pada tahun 1986. MySQL adalah sebuah sistem manajemen *database* yang bersifat *open source* dan relational yang artinya data-data yang dikelola dalam *database* akan diletakkan pada beberapa tabel yang terpisah sehingga manipulasi data akan menjadi jauh lebih cepat.
4. PhpMyAdmin, pengelolaan *database* dengan MySQL harus dilakukan dengan mengetik baris-baris perintah yang sesuai (*command line*) untuk setiap maksud tertentu. Dengan phpMyAdmin dapat memudahkan dalam pembuatan table, mengisi data dan lain-lain tanpa harus menghafal perintahnya.
5. Perl adalah bahasa pemrograman untuk segala keperluan, dikembangkan pertama kali oleh Larry Wall di mesin Unix. Dirilis pertama kali pada tanggal 18 Desember 1987 ditandai dengan keluarnya Perl 1. Dua karakteristik utama Perl adalah penanganan teks dan berbagai jalan pintas untuk menyelesaikan persoalan-persoalan umum. Perl sangat popular digunakan dalam program-program CGI (*Commaon Gateway Interface*) dan berbagai protokol internet lainnya. [13]

2.5.3 Pengenalan CodeIgniter

CodeIgniter adalah *framework* yang digunakan untuk membuat sebuah aplikasi berbasis web yang disusun dengan menggunakan bahasa PHP. Di dalam CodeIgniter ini terdapat beberapa macam kelas yang berbentuk *library* dan *helper* yang berfungsi untuk membantu pemrogram dalam mengembangkan aplikasinya. CodeIgniter sangat mudah dipelajari oleh seorang pemogram web pemula sekalipun. Alasan, karena CodeIgniter mempunyai file dokumentasi yang sangat memadai untuk menjelaskan setiap fungsi yang ada pada *library* dan *helper*. File dokumentasi ini disertakan secara langsung pada saat mengunduh paket *framework* CodeIgniter.

Seperti halnya PHP, CodeIgniter juga bersifat *free opensource software* di mana setiap orang berhak menggunakannya tanpa harus dikenai biaya lisensi. Saat ini, banyak sekali pengembangan yang telah menggunakan CodeIgniter sebagai *framework* untuk menyusun aplikasi berbasis web. Terdapat banyak forum serta milis (*mailing list*) yang membahas penggunaan CodeIgniter dalam berbagai macam kasus pemrograman. Cara kerja *framework* CodeIgniter dapat dilihat pada Gambar 2.1 berikut ini.



Gambar 2.1 Cara Kerja *Framework* CodeIgniter [10]

1. File index.php sebagai file yang bertindak sebagai control halaman depan. File ini menginisialisasi semua *resource* yang diperlukan untuk menjalankan CodeIgniter atau halaman situs.

2. Routing memeriksa setiap *request* HTTP dan melakukan apa yang perlu dilakukan terhadap *request* tersebut, tergantung *rule* yang ada.
3. Apabila terdapat Cache maka sistem akan mengirimkan *cache* tersebut secara langsung ke *browser* tanpa harus melewati birokrasi sistem secara normal.
4. Sebelum membuka Application Controller, setiap *request* HTTP dan data yang dikirim oleh pengguna maka akan di-*filter* terlebih dahulu melalui sebuah sistem pengamanan.
5. Controller kemudian memuat seluruh *resource*, baik itu Model, Library, maupun Helper yang digunakan untuk memproses *request* atau data yang dikirimkan.
6. Tahapan terakhir adalah View, berfungsi me-render hasil olahan Controller dan mengirimkannya ke *web browser*. Apabila sistem Cache diaktifkan maka View akan di *cache* terlebih dahulu sebelum muncul ke *browser*.

Kelebihan dari *framework* CodeIgniter ini adalah sebagai berikut:

1. CodeIgniter sangat ringan jika djalankan pada semua platform

   CodeIgniter ringan dijalankan pada berbagai platform. Sistem utama dari CodeIgniter hanya memerlukan sedikit ruang pada server untuk membentuk sebuah class library. Ini yang membedakan CodeIgniter dengan framework-framework sejenisnya. Class library tambahan hanya digunakan ketika diminta oleh pengguna, sehingga dipakai ketika diperlukan. Ini yang menyebabkan CodeIgniter sangat ringan jika dijalankan.
2. CodeIgniter menggunakan M-V-C

   CodeIgniter menggunakan model M-V-C *framework*, model M-V-C adalah singkatan untuk "Model-View-Controller". Model ini dapat mempermudah pembedaan antara tampilan dan program. Ini secara khusus sangat baik untuk suatu proyek di mana designer bekerja dengan file template yang akan memperkecil banyaknya kode di setiap template karena sudah dipisahkan dengan baik.

3. CodeIgniter dikemas dalam sebuah framework yang lengkap.

   CodeIgniter dikemas dalam kumpulan class dalam sebuah *library* yang cukup lengkap yang dapat menunjang proses pembuatan web, seperti mengakses *database*, mengirim email, validasi form data, membuat sessions, memanipulasi gambar, bekerja dengan XML-RPC.
4. Konfigurasi yang sangat minim (*nearly zero configuration*)

   Tentu saja untuk menyesuiakan dengan *database* dan keleluasaan *routing* tetap diijinkan melakukan konfigurasi dengan mengubah beberapa file konfigurasi, seperti database.php atau autoload.php, namun untuk menggunakan CodeIgniter dengan *setting* standard hanya perlu merubah sedikit saja file pada folder config.
5. CodeIgniter mempunyai dokumentasi (*user guide*) yang lengkap.

   CodeIgniter mempunyai dokumentasi yang sangat rapi dan sangat jelas. Sehingga dengan adanya dokumentasi ini akan banyak membantu para programmer pemula yang ingin mempelajari *framework* CodeIgniter. Dokumentasi ini dapat dilihat pada website resmi CodeIgniter di http://www.codeigniter.com/user_guide.
6. Bisa langsung digunakan

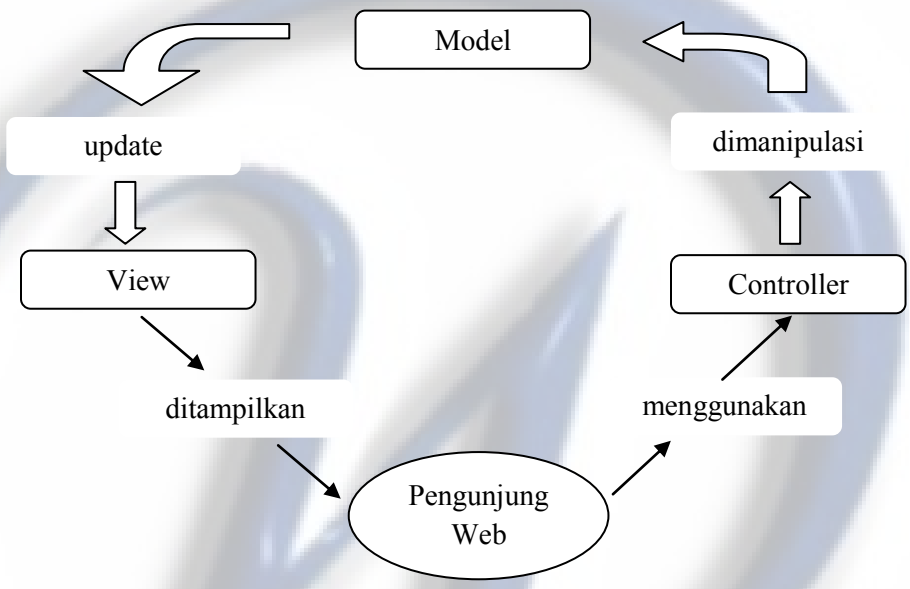
   Hanya dengan mengunduh, mengekstrak file dan meletakkan pada folder kerja *web server* (htdocs atau public html) maka akan bisa langsung digunakan.

Diantara kelebihan *framework* CodeIgniter di atas, *framework* CodeIgniter ini memiliki kekurangan atau kelemahan juga diantaranya adalah:

1. CodeIgniter tidak support AJAX dan ORM
2. CodeIgniter tidak bisa dipakai jika membuat aplikasi berskala besar, karena pengembangan akan semakin sulit dilakukan.

MVC (Model View Controller) adalah sebuah *software* yang memisahkan antara aplikasi logika dengan presentasi pada halaman web. Sehingga hal ini akan menyebabkan halaman web akan mengandung kode yang sedikit karena sudah terjadi

pemisahan antara tampilan dan pemrograman. Berikut di bawah ini adalah gambaran ilustrasi dari MVC.

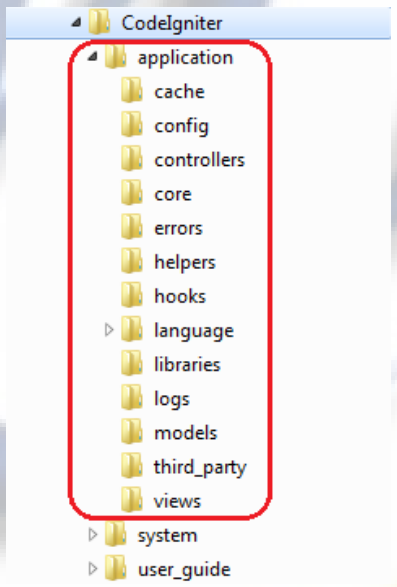


Gambar 2.2 MVC pada halaman web [14]

a. Model merupakan Bagian yang didesain untuk bekerja dengan *database*. Model bisa juga diartikan sebagai bagian yang digunakan untuk membuat semacam prototype data. Bagian model ini biasanya berisikan *query* yang dialamatkan pada sebuah *database*. Berbagai *query* tersebut disesuaikan dengan kebutuhan data yang akan diolah melalui controller, sebelum akhirnya ditampilkan menggunakan view.
b. Controller merupakan bagian yang digunakan untuk menempatkan fungsi-fungsi logika pengolah data yang telah diperoleh dari model (atau sumber lainnya), kemudian ditampilkan melalui (atau tanpa) view. Inti dari program yang dibuat dengan CodeIgniter terletak pada controller.
c. View adalah bagian presentasi yang berfungsi sebagai antarmuka program aplikasi dengan pengguna. View akan menampilkan data yang telah diolah sebelumnya melalui controller. Pengguna view tergantung pada controller, view

mana yang akan digunakan dan data apa saja yang akan ditampilkan. Pada bagian view biasanya tidak terdapat fungsi logika sama sekali, yang ada hanyalah kode-kode HTML dan sebagian kecil kode PHP yang digunakan untuk melakukan parsing data.

CodeIgniter adalah sebuah php *framework* yang berupa kumpulan folder dan file php, java script, css, txt dan file berbasis web lainnya dengan setting tertentu untuk menggunakannya serta menyediakan *library* dan *helper* yang dapat dimanfaatkan di dalam pemrograman php. CodeIgniter dijalankan *under web* dan harus dengan *web server*. Program CodeIgniter cukup diletakkan di bawah folder direktori *web server* yang telah ada. Berikut adalah struktur file CodeIgniter:



Gambar 2.3 Struktur file CodeIgniter

2.6 Metode Pengembangan Sistem

Metode yang digunakan untuk mengembangkan aplikasi bisnis *online* ini meliputi : *Waterfall Process Model*, *Object Oriented Programming* (OOP) dan *Unified Modelling Language* (UML).

2.6.1 Rekayasa Perangkat Lunak

2.6.1.1 Definisi Rekayasa Perangkat Lunak

Istilah Rekayasa Perangkat Lunak (RPL) secara umum disepakati sebagai terjemahan dari istilah *Software engineering*. Istilah *Software Engineering* mulai dipopulerkan pada tahun 1968 pada Software Engineering Conference yang diselenggarakan oleh NATO. Sebagian orang mengartikan RPL hanya sebatas pada bagaimana membuat program komputer. Padahal ada perbedaan yang mendasar antara perangkat lunak (*software*) dan program komputer.

Perangkat lunak adalah seluruh perintah yang digunakan untuk memproses informasi. Perangkat lunak dapat berupa program atau prosedur. Program adalah kumpulan perintah yang dimengerti oleh komputer sedangkan prosedur adalah perintah yang dibutuhkan oleh pengguna dalam memproses informasi.

Pengertian RPL sendiri adalah suatu disiplin ilmu yang membahas semua aspek produksi perangkat lunak, mulai dari tahap awal yaitu analisa kebutuhan pengguna, menentukan spesifikasi dari kebutuhan pengguna, disain, pengkodean, pengujian sampai pemeliharaan sistem setelah digunakan. Dari pengertian ini jelaslah bahwa RPL tidak hanya berhubungan dengan cara pembuatan program komputer. Pernyataan ”semua aspek produksi” pada pengertian di atas, mempunyai arti semua hal yang berhubungan dengan proses produksi seperti manajemen proyek, penentuan personil, anggaran biaya, metode, jadwal, kualitas sampai dengan pelatihan pengguna merupakan bagian dari RPL.

2.6.1.2 Tujuan Rekayasa Perangkat Lunak

Secara umum tujuan RPL tidak berbeda dengan bidang rekayasa yang lain. Hal ini dapat kita lihat pada gambar di bawah ini.

Dalam perkembangan Rekayasa Perangkat Lunak banyak berbagai model yang digunakan untuk membantu dalam pengembangan perangkat lunak (*software*). Pada umumnya model-model tersebut berpatokan pada model proses pengembangan system yang disebut SDLC (*System Development Live Cycle*).

2. Waterfall model

   Model ini merupakan model yang dianggap kuno dan disebut juga sebagai "classic life cycle". Tetapi model ini paling banyak digunakan dalam Software Engineering. Disebut dengan *waterfall* karena tahap yang pertama harus dilalui dan diselesaikan terlebih dahulu untuk dapat melanjutkan ke tahapan selanjutnya.

3. Incremental Model

   Incremental model mengkombinasikan elemen dari Waterfall model yang diaplikasikan dalam bentuk iterative. Proses tahapan pada model ini samahalnya dengan waterfall model akan tetapi dilakukan secara berbaris-baris dan setiap baris disebut dengan increment. Dimana setiap increment menghasilkan fungsi produksi yang berbeda. Dan biasanya hasil increment pertama merupakan bagian inti dari software yang akan dikirimkan lebih dulu kepada customer. Sedangkan hasil increment yang lainnya merupakan tambahan atau fitur tambahan dari *software* utama.

4. RAD Model

   Rapid Application Development (RAD) merupakan model proses incremental software yang menekankan pada *short development cycle*. RAD model adalah adaptasi berkecepatan tinggi "high-speed" dari waterfall model yang mana secara cepat perkembangan dicapai menggunakan pendekatan konstruksi berbasis komponen. Jika kebutuhan dipahami dengan baik, proses RAD memungkinkan tim pengembang menciptakan "sistem fungsional yang utuh" dalam periode waktu yang singkat kira-kira 60 sampai 90 hari.

5. Prototyping Model

   Prototyping Model digunakan ketika customer tidak tahu secara pasti apa yang dibutuhkannya sehingga developer pun kurang memahami apa yang diinginkan

oleh customer. Paradigma Prototyping membantu *Software Engineer* dan customer lebih memahi akan apa yang akan dibuatnya ketika kebutuhan tidak jelas.

6. Spiral Model

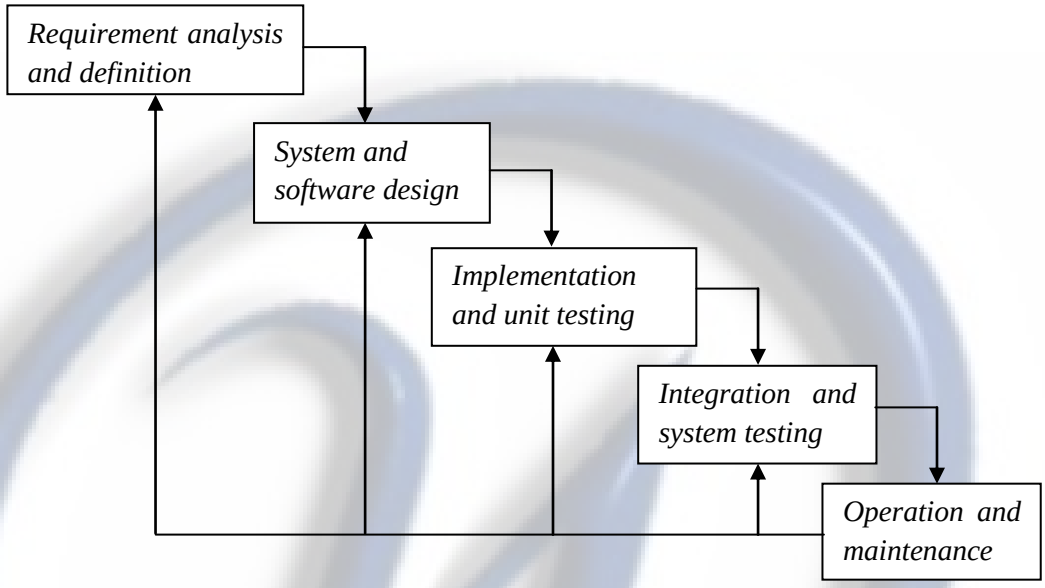
   Spiral model adalah salah satu bentuk evolusi yang menggunakan metode iterasi natural yang dimiliki oleh model Prototyping dan digabungakan dengan aspek sistematis yang dikembangkan dengan model Waterfall. Tahap desain umumnya digunakan pada model Waterfall, sedangkan tahap Prototyping adalah suatu model dimana software dibuat prototype (incomplete model), "blue-print"-nya, atau contohnya dan ditunjukkan ke user / customer untuk mendapatkan feedback-nya. Jika prototype-nya sudah sesuai dengan keinginan user / customer, maka proses SE dilanjutkan dengan membuat produk sesungguhnya dengan menambah dan memperbaiki kekurangan dari prototype tadi. [8]

2.6.1.4 *Waterfall Process Model*

Model proses yang akan digunakan untuk perancangan aplikasi ini menggunakan *Waterfall Model,* karena kelebihan dari waterfall adalah:

a. Mudah dalam pengaplikasiannya
b. Ketika semua kebutuhan sistem dapat didefinisikan secara utuh dan benar sejak awal project, maka Software Engineering dapat berjalan dengan baik tanpa masalah.
c. Walaupun dalam pengumpulan kebutuhan tidak selalu dapat didefinisikan secara utuh seperti yang diinginkan akan tetapi masalah yang timbul saat pengumpulan kebutuhan sistem ketika awal project tidak mengeluarkan biaya yang besar baik uang, waktu dan usaha.

   Waterfall model tersebut dapat dilihat pada Gambar 2.3 di bawah ini.



Gambar 2.5 Waterfall Model Menurut Sommerville [4]

Berikut adalah penjelasan dari tahap-tahap yang dilakukan di dalam model ini:

1. *Requirement analysis and definition*

   Mengumpulkan kebutuhan secara lengkap kemudian kemudian dianalisis dan didefinisikan kebutuhan yang harus dipenuhi oleh program yang akan dibangun. Fase ini harus dikerjakan secara lengkap untuk bisa menghasilkan desain yang lengkap.

2. *System and software design*

   Desain dikerjakan setelah kebutuhan selesai dikumpulkan secara lengkap.

3. *Implementation and unit testing*

   Desain program diterjemahkan ke dalam kode-kode dengan menggunakan bahasa pemrograman yang sudah ditentukan. Program yang telah dibangun langsung diuji baik secara unit.

4. *Integration and system testing*

   Penyatuan unit-unit program dengan langsung melakukan pengujian secara keseluruhan (*system testing*)

5. *Operation and maintenance*

Mengoperasikan program dilingkungannya dan melakukan pemeliharaan, seperti penyesuaian atau perubahan karena adaptasi dengan situasi sebenarnya. [4]

2.6.2 *Object Oriented Programming* (OOP)

Object Oriented Programming (OOP) adalah metode yang baik dalam pengembangan aplikasi. Sebuah variabel biasa dapat diubah menjadi sebuah *object* dan dapat melakukan banyak hal seperti simpan data, memformat penampilan data, dan lain-lain.

Sama hal-nya dengan bahasa pemrograman lainnya, JavaScript juga membutuhkan konsep OOP dalam pengembangan aplikasi yang lebih *complex* dan *powerful*. Saat ini banyak *framework* JavaScript yang menawarkan kemudahan dalam pengembangan aplikasi namun tidak banyak diantara *framework* tersebut yang *support* dengan konsep OOP seperti pemrograman lainnya. [6]

Ada 3 (tiga) konsep yang harus dimiliki oleh suatu bentuk pemograman yang berorientasi objek, diantaranya:

1. Enkapsulasi/Pengkapsulan (*Encapsulation*)

   Enkapsulasi adalah pembungkus, pembungkus disini dimaksudkan untuk menjaga suatu proses program agar tidak dapat diakses secara sembarangan atau di intervensi oleh program lain. Konsep enkapsulasi sangat penting dilakukan untuk menjaga kebutuhan program agar dapat diakses sewaktu-waktu, sekaligus menjaga program tersebut.

2. *Inheritance* (Pewarisan)

   Inheritance (penurunan sifat / pewarisan), ini merupakan ciri khas dari OOP yang tidak terdapat pada pemrograman prosedural gaya lama. Dalam hal ini, *inheritance* bertujuan membentuk obyek baru yang memiliki sifat sama atau mirip dengan obyek yang sudah ada sebelumnya (pewarisan). Obyek turunan dapat digunakan membentuk obyek turunan lagi dan seterusnya. Setiap perubahan pada obyek induk, juga akan mengubah obyek turunannya. Susunan obyek induk

dengan obyek turunannya disebut dengan hirarki obyek. Atau *Inheritance* yaitu pewarisan sifat-sifat suatu object kepada object turunannya.

3. *Polymorphism*

   Polymorphism adalah suatu aksi yang memungkinkan pemrogram menyampaikan pesan tertentu keluar dari hirarki obyeknya, dimana obyek yang berbeda memberikan tanggapan/respon terhadap pesan yang sama sesuai dengan sifat masing-masing obyek. Atau *Polymorphic* dapat berarti banyak bentuk, maksudnya yaitu kita dapat menimpa (*override*), suatu method, yang berasal dari parent class (super class) dimana object tersebut diturunkan, sehingga memiliki kelakuan yang berbeda.

2.6.3 *Unified Modelling Language* (UML)

2.6.3.1 Pengertian UML

Unified Modeling Language (UML) merupakan sistem arsitektur yang bekerja dalam OOAD dengan satu bahasa yang konsisten untuk menentukan, visualisasi, mengkontruksi, dan mendokumentasikan *artifact* yang terdapat dalam sistem *software*.

UML merupakan bahasa pemodelan yang paling sukses dari tiga metode OO yang telah ada sebelumnya, yaitu *Booch, OMG, and OOSE*. UML merupakan kesatuan dari ketiga metode pemodelan tersebut dan ditambah kemampuan lebih karena mengandung metode tambahan untuk mengatasi masalah pemodelan yang tidak dapat ditangani ketiga pemodelan tersebut.

2.6.3.2 Mengenal *Rational Rose*

Rasional Rose adalah *software* yang memiliki perangkat-perangkat pemodelan secara visual untuk membangun suatu solusi dalam rekayasa *software* dan pemodelan bisnis. *Rational Rose* dikeluarkan oleh perusahaan *software* yang bernama Rational Software, perusahaan yang mencetus ide pembentukan konsorsium bagi perusahaan-perusahaan yang memakai standar UML sebagai bahasa pemodelan di

perusahaannya. *Rational Rose* memakai UML sebagai bahasa pemodelannya, ditambah beberapa fitur lain yang membuat *Rational Rose* menjadi *software* pemodelan visual yang terkemuka.

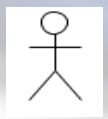
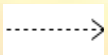
Visual Modeling adalah proses menggambarkan cetak biru suatu sistem secara grafis, terdiri dari komponen-komponen, *interfaces*, dan koneksi-koneksi yang ada dalam sistem tersebut, agar mudah dipahami dan dikomunikasikan. *Visual modeling* dapat membantu untuk menampilkan elemen-elemen yang penting secara detil dari suatu masalah yang kompleks dan menyaring untuk kemudian membuang elemen-elemen yang tidak penting. [12]

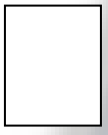
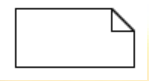
2.6.3.3 Diagram UML

UML mendefinisikan diagram-diagram berikut ini :

1. *Use case diagram* menggambarkan fungsionalitas yang diharapkan dari sebuah sistem. Yang ditekankan adalah "apa" yang diperbuat sistem, dan bukan "bagaimana". Sebuah *use case* merepresentasikan sebuah interaksi antara aktor dengan sistem.

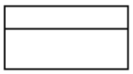
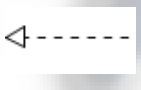
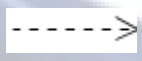
Tabel 2.1 Daftar Simbol *Usecase Diagram*

NO	GAMBAR	NAMA	KETERANGAN
1		*Actor*	Menspesifikasikan himpuan peran yang pengguna mainkan ketika berinteraksi dengan *use case*.
2		*Dependency*	Hubungan dimana perubahan yang terjadi pada suatu elemen mandiri *(independent)* akan mempengaruhi elemen yang bergantung padanya elemen yang tidak mandiri (*independent*).

3		*Generalization*	Hubungan dimana objek anak (*descendent*) berbagi perilaku dan struktur data dari objek yang ada di atasnya objek induk (*ancestor*).
4		*Include*	Menspesifikasikan bahwa *use case* sumber secara *eksplisit*.
5		*Extend*	Menspesifikasikan bahwa *use case* target memperluas perilaku dari *use case* sumber pada suatu titik yang diberikan.
6		*Association*	Apa yang menghubungkan antara objek satu dengan objek lainnya.
7		*System*	Menspesifikasikan paket yang menampilkan sistem secara terbatas.
8		*Use Case*	Deskripsi dari urutan aksi-aksi yang ditampilkan sistem yang menghasilkan suatu hasil yang terukur bagi suatu aktor
9		*Collaboration*	Interaksi aturan-aturan dan elemen lain yang bekerja sama untuk menyediakan prilaku yang lebih besar dari jumlah dan elemen-elemennya (sinergi).
10		*Note*	Elemen fisik yang eksis saat aplikasi dijalankan dan mencerminkan suatu sumber daya komputasi

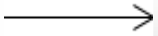
2. *Class diagram* menggambarkan struktur dan deskripsi *class, package* dan *object* beserta hubungan satu sama lain seperti *containment*, pewarisan, asosiasi, dan lain-lain.

Tabel 2.2 Daftar Simbol *Class Diagram*

NO	GAMBAR	NAMA	KETERANGAN
1		*Generalization*	Hubungan dimana objek anak *(descendent)* berbagi perilaku dan struktur data dari objek yang ada di atasnya objek induk (*ancestor*).
2		*Nary Association*	Upaya untuk menghindari asosiasi dengan lebih dari 2 objek.
3		*Class*	Himpunan dari objek-objek yang berbagi atribut serta operasi yang sama.
4		*Collaboration*	Deskripsi dari urutan aksi-aksi yang ditampilkan sistem yang menghasilkan suatu hasil yang terukur bagi suatu aktor
5		*Realization*	Operasi yang benar-benar dilakukan oleh suatu objek.
6		*Dependency*	Hubungan dimana perubahan yang terjadi pada suatu elemen mandiri *(independent)* akan mempegaruhi elemen yang bergantung padanya elemen yang tidak mandiri
7		*Association*	Apa yang menghubungkan antara objek satu dengan objek lainnya

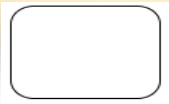
3. *Statechart diagram* menggambarkan transisi dan perubahan keadaan (dari satu *state* ke *state* lainnya) suatu objek pada sistem sebagai akibat dari stimuli yang diterima. Pada umumnya *statechart diagram* menggambarkan *class* tertentu (satu *class* dapat memiliki lebih dari satu *statechart diagram*).

Tabel 2.3 Daftar Simbol *Statechart Diagram*

NO	GAMBAR	NAMA	KETERANGAN
1		*State*	Nilai atribut dan nilai link pada suatu waktu tertentu, yang dimiliki oleh suatu objek.
2		*Initial Pseudo State*	Bagaimana objek dibentuk atau diawali.
3		*Final State*	Bagaimana objek dibentuk dan dihancurkan.
4		*Transition*	Sebuah kejadian yang memicu sebuah state objek dengan cara memperbaharui satu atau lebih nilai atributnya
5		*Association*	Apa yang menghubungkan antara objek satu dengan objek lainnya.
6		*Node*	Elemen fisik yang eksis saat aplikasi dijalankan dan mencerminkan suatu sumber daya komputasi.

4. *Actifity diagram* menggambarkan berbagai alir aktivitas dalam sistem yang sedang dirancang, bagaimana masing-masing alir berawal, *decision* yang mungkin terjadi, dan bagaimana mereka berakhir. *Actifity diagram* juga dapat menggambarkan proses paralel yang mungkin terjadi pada beberapa eksekusi.

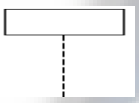
Tabel 2.4 Daftar Simbol *Actifity Diagram*

NO	GAMBAR	NAMA	KETERANGAN
1		*Actifity*	Memperlihatkan bagaimana masing-masing kelas antarmuka saling berinteraksi satu sama lain
2		*Action*	State dari sistem yang mencerminkan eksekusi dari suatu aksi

3		*Initial Node*	Bagaimana objek dibentuk atau diawali.
4		*Actifity Final Node*	Bagaimana objek dibentuk dan dihancurkan
5		*Fork Node*	Satu aliran yang pada tahap tertentu berubah menjadi beberapa aliran

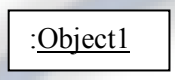
5. *Sequence diagram* menggambarkan interaksi antar objek di dalam dan di sekitar sistem (termasuk pengguna, *display*, dan sebagainya) berupa *message* yang digambarkan terhadap waktu. *Sequence diagram* terdiri atar dimensi vertikal (waktu) dan dimensi horizontal (objek-objek yang terkait).

Tabel 2.5 Daftar Simbol *Sequence Diagram*

NO	GAMBAR	NAMA	KETERANGAN
1		*LifeLine*	Objek *entity*, antarmuka yang saling berinteraksi.
2		*Message*	Spesifikasi dari komunikasi antar objek yang memuat informasi-informasi tentang aktifitas yang terjadi
3		*Message*	Spesifikasi dari komunikasi antar objek yang memuat informasi-informasi tentang aktifitas yang terjadi

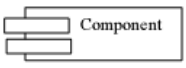
6. *Collaboration diagram* juga menggambarkan interaksi antar objek seperti *sequence diagram*, tetapi lebih menekankan pada peran masing-masing objek dan bukan pada waktu penyampaian *message*. Setiap *message* memiliki *sequence number*, di mana *message* dari level tertinggi memiliki nomor 1. *Messages* dari level yang sama memiliki *prefiks* yang sama.

Tabel 2.6 Daftar Simbol *Collaboration Diagram*

NO	GAMBAR	NAMA	KETERANGAN
1		*Object*	*Object* merupakan *instance* dari sebuah *class* dan dituliskan tersusun secara *horizontal*. Digambarkan sebagai sebuah *class* (kotak) dengan nama *object* di dalamnya yang diawali dengan sebuah titik koma.
2		*Actor*	*Actor* juga dapat berkomunikasi dengan *object* , maka *actor* juga dapat diurutkan sebagai kolom. Simbol *Actor* sama pada *Actor Use Case* Diagram.
3		*Message*	*Message*, digambarkan dengan anak panah yang mengarah antar obyek dan diberi label urutan nomor yang mengindikasikan urutan komunikasi yang terjadi antar obyek.

7. *Component diagram* menggambarkan struktur dan hubungan antar komponen piranti lunak, termasuk ketergantungan (*dependency*) di antaranya.

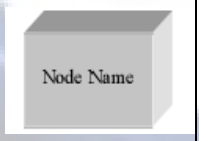
Tabel 2.7 Daftar Simbol *Component Diagram*

NO	GAMBAR	NAMA	KETERANGAN
1		*Component*	Sebuah komponen melambangkan sebuah entitas *software* dalam sebuah sistem. Sebuah komponen dinotasikan sebagai sebuah kotak segiempat dengan dua kotak kecil tambahan yang menempel disebelah kirinya.

2		*Depedency*	Sebuah *Dependency* digunakan untuk menotasikan relasi antara dua komponen. Notasinya adalah tanda panah putus-putus yang diarahkan kepada komponen tempat sebuah komponen itu bergantung.

8. *Deployment/physical diagram* menggambarkan detail bagaimana komponen di-*deploy* dalam infrastruktur sistem, di mana komponen akan terletak (pada mesin, server atau piranti keras apa), bagaimana kemampuan jaringan pada lokasi tersebut, spesifikasi server, dan hal-hal lain yang bersifat fisikal

Tabel 2.8 Daftar Simbol *Physical Diagram*

NO	GAMBAR	NAMA	KETERANGAN
1		*Component*	Pada *deployment* diagram, komponen-komponen yang ada diletakkan didalam node untuk memastikan keberadaan posisi mereka.
2		*Node*	*Node* menggambarkan bagian-bagian *hardware* dalam sebuah sistem. Notasi untuk node digambarkan sebagai sebuah kubus 3 dimensi.
3		*Association*	Sebuah *association* digambarkan sebagai sebuah garis yang menghubungkan dua node yang mengindikasikan jalur komunikasi antara komponen-komponen *hardware*.

BAB III
ANALISIS SISTEM

Pada bab analisis sistem ini merupakan penguraian dari suatu sistem informasi yang utuh ke dalam bagian-bagian komponennya dengan maksud untuk mengidentifikasikan dan mengevaluasi permasalahan, hambatan yang terjadi dan kebutuhan yang diharapkan sehingga dapat diusulkan perbaikan.

3.1 Tinjauan Organisasi

3.1.1 Sejarah Organisasi

Nekatun merupakan usaha yang bergerak dibidang rajutan atau renda. Asal dari kata Nekatun tidak ada arti khusus, hanyalah sebuah nama yang digunakan untuk sebuah hasil kreasi rajutan atau renda yang dibuat dari benang rami dikombinasikan dengan kain blacu menjadi kreasi – kreasi perlengkapan rumah tangga yang unik.

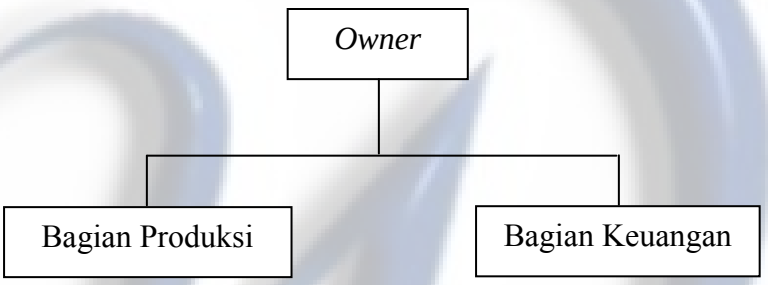
Nekatun berdiri pada awal September tahun 2000 yang bertempat di jalan Sersan Surip No. 129A / 169A Ledeng – Bandung. Awalnya hanya ingin membuat sesuatu yang unik dari material yang mudah diperoleh tapi jarang dilirik orang. Blacu menjadi pilihan, karena memang selalu ada setiap waktu dan asli buatan negeri sendiri. Memadukan blacu dan benang adalah ide awal dari pemilik Nekatun ini dari sosok ibu rumah tangga yang bernama Nunu Rismayanti Idris. Tekstur benang ramie yang sedikit kasar dan kaku, ternyata sangat cocok bila dipadukan dengan kain blacu. Selain warnanya yang natural, membuat blacu yang berwarna putih tulang tampak lebih menarik dan unik.

Kreasi-kreasi perlengkapan rumah tangga yang unik hadir karena memang diperlukan para wanita yang peduli mempercantik rumahnya. Sarung bantal, taplak meja, penutup tisu, penutup galon, *bed cover* dan aneka perlengkapan lainnya menjadi pilihan Nekatun untuk berkreasi. Bagi Nekatun kreatifitas tak boleh padam, selalu mencari sesuatu yang baru dan bermanfaat adalah tujuan akhir Nekatun. Paduan rajutan dan sulaman tangan yang apik selalu menjadi ciri khas kreasi

Nekatun. Meski hanya berasal dari selembar kain blacu tak berarti tak ada kreatifitas, kreasi perlengkapan rumah tangga yang unik Nekatun hadirkan untuk anda yang mengerti keindahan.

3.1.2 Struktur Organisasi dan *Job Description*

1. Struktur Organisasi



Gambar 3.1 Struktur Organisasi Nekatun Bandung

2. *Job Description*

a. *Owner*

Bertugas mengontrol semua kegiatan yang ada di Nekatun Bandung, mengelola semua transaksi penerimaan pesanan, mengontrol keuangan perusahaan, mengelola dan mengontrol sumber daya tenaga kerja serta memasarkan produk Nekatun.

b. Bidang Produksi

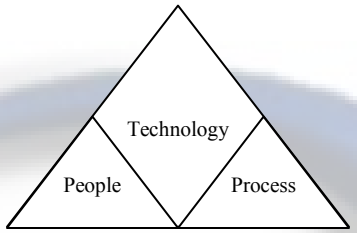
Bertugas mengelola dan mengontrol pemroduksian produk Nekatun.

c. Bidang Keuangan

Bertugas mengelola keuangan perusahaan dan membuat laporan keuangan perbulanan dan pertahunan.

3.2 Analisis Sistem yang Berjalan

Berdasarkan hasil pengamatan dan wawancara yang telah dilakukan di Nekatun, penulis akan menguraikan ruang lingkup yang perlu di analisis, diantaranya yaitu teknologi, person dan proses.



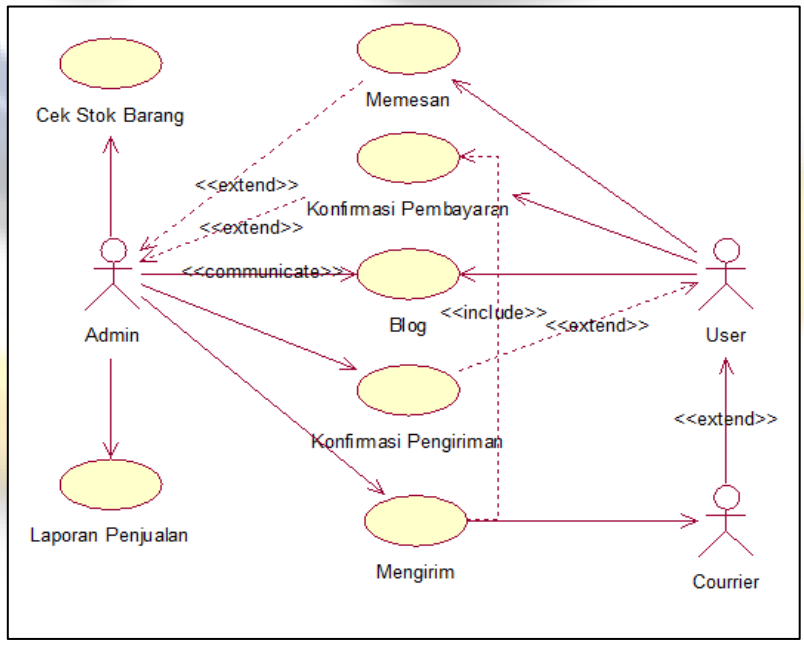
Gambar 3.2 Ruang Lingkup Yang Akan Di Analisis

3.2.1 *Process*

Analisis pada prosedur atau mekanisme kerja yang sedang berjalan diuraikan secara sistematis aktivitas-aktivitas yang terjadi dalam sistem informasi yang ada dan sedang berjalan di Nekatun Bandung.

3.2.1.1 *Use Case* Sistem Penjualan yang Sedang Berjalan

Sistem penjualan yang dilakukan oleh Nekatun selama ini yaitu menggunakan blog statik, dimana blog tersebut berfungsi sebagai salah satu alat promosi yang dilakukan oleh Nekatun kepada masyarakat luas.



Gambar 3.3 *Use Case* Sistem Penjualan Berjalan

3.2.1.2 Skenario Sistem Penjualan yang Sedang Berjalan

Skenario dari alur proses sistem penjualan yang sedang berjalan seperti pada Gambar 3.3 berikut penjelasannya :

1. *Admin* menyediakan informasi dari produk yang dijual oleh Nekatun kepada konsumen melalui blog.
2. Konsumen melihat informasi mengenai produk-produk yang dijual di Nekatun.
3. Konsumen konsultasi kepada *admin* melalui telepon atau SMS mengenai produk yang tersedia dan harga per-setiap produk.
4. Konsumen memesan produk melalui telepon atau SMS.
5. Konsumen mentransfer biaya yang harus dibayar dan konfirmasi kepada *admin*.
6. *Admin* mengecek mutasi pembayaran konsumen.
7. Setelah transferan terbukti validasinya, maka *admin* akan mengirimkan barang melalui *courrier*.
8. *Admin* memberikan konfirmasi pengiriman produk melalui SMS.

3.2.2 *People*

Sumber daya manusia yang terlibat dalam perusahaan yaitu meliputi:

1. Bagian produksi

   Dalam bagian produksi para pengrajin yang dipilih yaitu:

   a. Ibu rumah tangga yang memiliki kreatifitas dan inovatif di lingkungan sekitar,
   b. Siswi Sekolah Menengah Kejuruan dan Mahasiswi yang sedang melakukan Praktek Kerja Lapangan (PKL) ataupun magang.
   c. Mahasiswi-mahasiswi yang tinggal di kontrakan pemilik Nekatun yang memiliki keterampilan untuk menjadi karyawan paruh waktu (*part time*).

2. Bagian Keuangan

   Dalam bagian keuangan ini hanya terdapat satu orang karyawan yang memiliki keahlian khusus dalam mengelola keuangan di Nekatun.

3.2.3 *Technology*

Sampai saat ini infrastruktur IT yang ada di Nekatan yaitu meliputi :

1. Telepon kantor

   Yang digunakan untuk jalur komunikasi dengan *supplier* dan *customer*.

2. Notebook (Laptop)

   Yang digunakan untuk menyimpan data, file foto atau dokumen-dokumen tentang acara yang pernah diikuti dan laporan keuangan menggunakan microsoft excel serta untuk mencari informasi-informasi terkait produk yang akan dikembangkan melalui internet.

   Analisis perangkat keras dan perangkat lunak diantaranya adalah.

   a. Analisis Perangkat Keras (*Hardware*)
      1. *Processor* Intel Core 2 Duo 2,00 GHz + Motherboard,
      2. VGA 256 MB
      3. RAM / Memory 2 GB,
      4. *Harddisk* 250 GB.
      5. *Mouse*

   b. Analisis Perangkat Lunak (*Software*)
      1. Sistem Operasi : Windows 7 Ultimate 32-bit,
      2. Antivirus : Avira AntiVir Personal,
      3. Laporan Penjualan : Microsoft Office Word 2007,
      4. Laporan Keuangan : Microsoft Office Excel 2007,
      5. *Editing Photo* dan *Design* : Adobe Photoshop CS3 dan CorelDRAW Graphics Suite 12,
      6. Browser : Internet Explorer, Mozilla Firefox dan Google Chrome,
      7. Obrolan : Yahoo Messenger,
      8. Koneksi internet: Mobile Partner,
      9. Alat Penghitung : Calculator,
      10. Media player : Winamp dan Windows Media Player.

Berdasarkan analisis perangkat keras dan perangkat lunak di atas, spesifikasi yang ada sangat mencukupi, hanya saja perlu printer standar untuk mencetak laporan penjualan dan keuangan.

3.3 Evaluasi Sistem yang Berjalan

Setelah menganalisis sistem yang sedang berjalan pada Nekatun ternyata pada sistem penjualan yang ada masih terdapat beberapa kekurangan, sehingga perlu adanya perbaikan-perbaikan guna memberikan solusi untuk melengkapi kekurangan yang terjadi pada sistem yang sedang berjalan. Berikut adalah permasalahan yang terjadi pada sistem yang berjalan :

Tabel 3.1 Evaluasi Sistem yang Berjalan

No.	Permasalahan	Solusi
1	Sistem yang berjalan tidak dapat mengelola data produk atau stok yang tersedia, sehingga konsumen harus menanyakan kepada *admin* bahwa produk tersebut tersedia atau tidaknya.	Perlunya dibangun sebuah database yang dapat mengelola persediaan barang guna mempermudah konsumen untuk mengetahui informasi produk yang tersedia.
2	Sistem yang berjalan tidak dapat menghitung biaya transaksi pembeli secara langsung, sehingga konsumen harus bertanya-tanya kepada *admin* untuk harga-harga produk yang menyebabkan proses transaksi lebih lama dan terkadang kurang akurat.	Perlunya dibangun sistem transaksi pembelian secara otomatis pada sistem yang akan dibangun guna untuk mengefisienkan proses transaksi pembelian.

3	Sistem yang berjalan tidak tersedianya form pemesanan sehingga konsumen harus menghubungi *admin* langsung, hal tersebut menyebabkan lamanya proses pemesanan ketika *admin* tidak ditempat atau sedang sibuk.	Perlunya dibangun sistem pemesanan secara *online* dengan membuatkan form pemesanan bagi konsumen.
4	Sistem yang berjalan tidak dapat menampilkan status pemesanan para konsumen, sehingga admin harus terus memberikan konfirmasi kepada konsumen. Ketika kontak konsumen hilang, maka admin tidak dapat mengkonfirmasi konsumen.	Perlunya dibangun sistem yang menampilkan status pesanan konsumen guna memudahkan konsumen untuk mendapatkan informasi mengenai sudah atau belumnya pesanan konsumen diproses.
5	Sistem yang berjalan tidak dapat membuat laporan secara otomatis, sehingga *admin* harus membuat laporan secara manual dengan waktu yang cukup lama.	Perlunya dibangun sistem untuk pembuatan pelaporan secara otomatis sehingga dapat mengefisienkan waktu pembuatan laporan penjualan.

3.4 Analisis Kebutuhan Sistem

Dari permasalahan yang timbul di atas, maka perlunya dibangun aplikasi penjualan yang terkomputerisasi diharapkan dapat membantu melengkapi kekurangan yang terjadi di perusahaan. Aplikasi yang dibutuhkan meliputi :

1. Sistem yang dapat memudahkan dalam peng-*input*-an data produk.
2. Sistem yang dapat mengelola ketersediaan barang dengan otomatis.

3. Sistem yang dapat melakukan perhitungan biaya langsung secara otomatis yang harus dibayar oleh konsumen.
4. Sistem yang dapat melakukan pemesanan secara langsung.
5. Sistem yang menyediakan informasi status pemesanan bagi konsumen.
6. Sistem yang dapat menghasilkan laporan penjualan secara otomatis.

3.5 Hasil Analisis

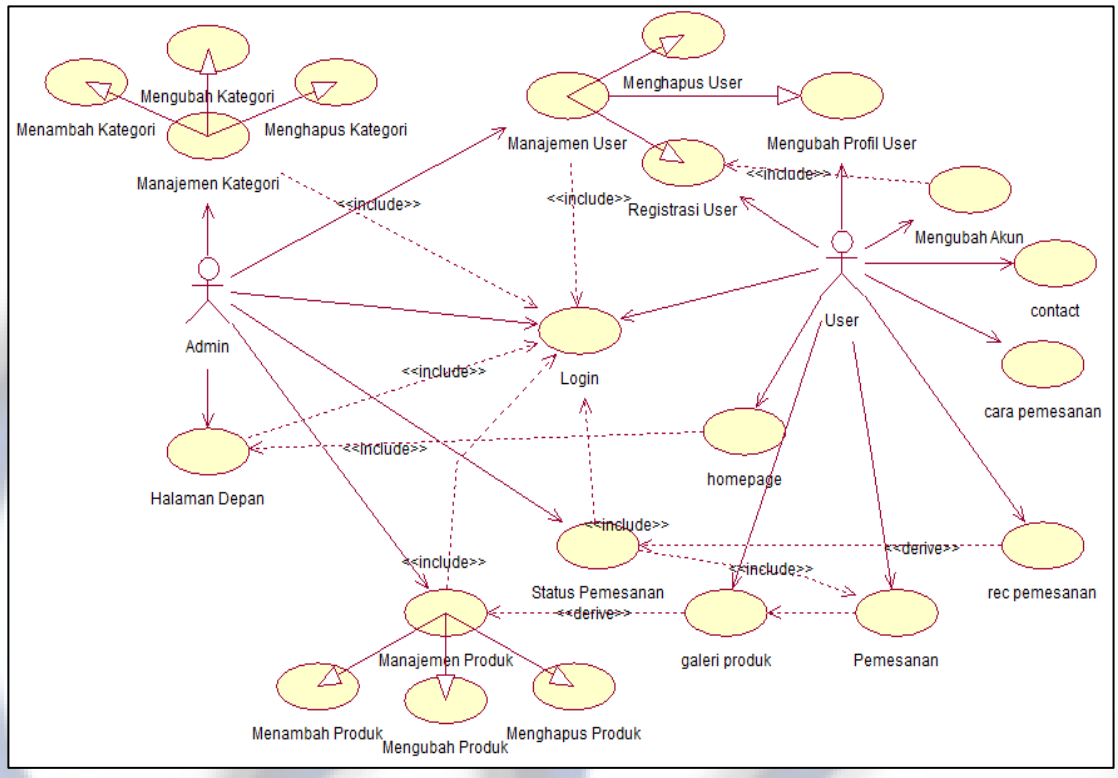
Berdasarkan uraian analisis sistem di atas serta memberikan upaya untuk mengatasi permasalahan-permasalahan yang ada seperti membangun database, membuat sistem transaksi pembelian, sistem pemesanan dan pembuatan laporan secara otomatis. Maka dari upaya-upaya tersebut akan dirangkum ke dalam aplikasi yang akan dibangun yaitu aplikasi bisnis online dengan menggunakan CodeIgniter guna semoga dapat memperbaiki kekurangan-kekurangan dari permasalahan yang terdapat di Nekatun Bandung.

3.6 Analisis Sistem yang Diusulkan

Setelah melakukan analisis sistem yang berjalan, peneliti menemukan beberapa kelemahan sistem yang sedang berjalan maka pada tahap perancangan ini peneliti akan membuat sebuah aplikasi penjualan *online* dengan menggunakan *framework* CodeIgniter yang diharapkan dapat memperbaiki kelemahan-kelemahan pada sistem yang sedang berjalan

3.6.1 *Use Case* Sistem Penjualan yang Diusulkan

Use case menggambarakan fungsionalitas dari yang diharapkan dari sebuah sistem. Berikut ini *use case diagram* dari sistem penjualan yang diusulkan.



Gambar 3.4 *Use case diagram* sistem penjualan yang diusulkan

BAB IV
PERANCANGAN SISTEM

Perancangan sistem adalah gambaran, perancangan dan pembuatan skema atau pengaturan dari beberapa elemen yang terpisah ke dalam satu kesatuan yang utuh dan mempunyai fungsi dan tujuan. Elemen-elemen sistem informasi dirancang dengan tujuan untuk dikomunikasikan kepada user. Dalam pengembangan sistem yang ada, baik secara keseluruhan maupun memperbaiki sistem yang telah ada.

4.1 Gambaran Umum Sistem yang Diusulkan

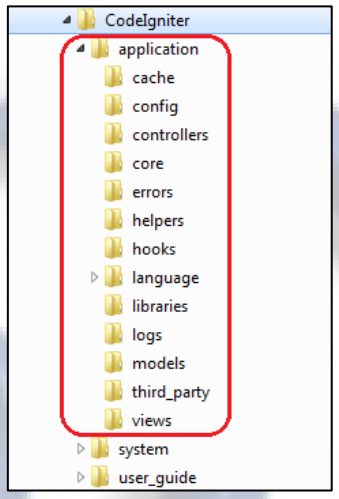
Sistem yang dirancang adalah aplikasi penjualan *online* yang dapat memudahkan dalam peng-*input*-an data produk, mengelola ketersediaan barang, melakukan perhitungan biaya secara otomatis, melakukan pemesanan langsung dan menampilkan sebuah laporan penjualan. Aktivitas tersebut yang secara keseluruhan tersimpan dalam sebuah *database*, sehingga dapat meminimalisir adanya kesalahan data.

Aplikasi penjualan *online* ini menggunakan *framework* CodeIgniter versi 2.0.2 dengan bahasa pemrograman PHP dengan program aplikasi yang digunakan adalah Adobe Dreamweaver CS5 dan XAMPP MySQL sebagai program aplikasi untuk membangun *database*-nya

4.2 Pengembangan Sistem

4.2.1 Perancangan Awal Sistem

Pada perancangan awal ini dilakukan instalasi CodeIgniter dan menyimpan folder CodeIgniter pada folder htdoc pada XAMPP. CodeIgniter mempunyai struktur direktori seperti pada gambar berikut ini.



Gambar 4.1 Struktur Direktori CodeIgniter

Folder application merupakan tempat aplikasi yang akan dikembangkan. Dari sekian banyak subfolder yang ada, yang utama dalam pengembangan ini adalah:

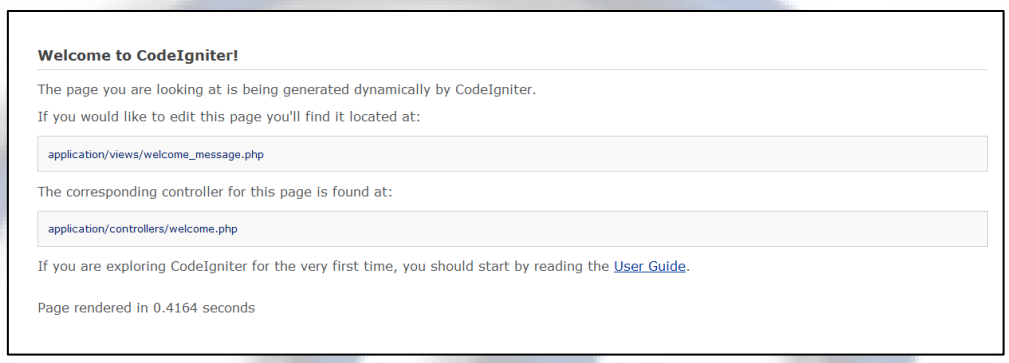
a. config, digunakan untuk menyimpan berbagai macam file konfigurasi sistem aplikasi yang akan dibuat.
b. controller, digunakan untuk meletakkan controller sistem aplikasi yang akan dibuat.
c. models, digunakan untuk meletakkan file model sistem aplikasi yang akan dibuat.
d. views, digunakan untuk meletakkan file view sistem aplikasi yang akan dibuat.

Pada direktori di atas akan ditambah folder baru yang digunakan untuk menyimpan file-file aplikasi dengan peran dan fungsi tertentu. Ketiga folder tambahan yang dimaksud adalah:

a. asset, digunakan untuk menyimpan file-file asset berupa CSS, JavaScript dan Images yang digunakan oleh halaman admin.
b. template, digunakan untuk menyimpan file template atau desain untuk halaman publik.

c. uploads, digunakan untuk menampung file-file hasil unggahan, seperti file foto produk dan *thumbnail*-nya.

Berikut adalah tampilan halaman awal CodeIgniter:



Gambar 4.2 Halaman Awal CodeIgniter

4.2.2 Perancangan Sistem yang Dikembangkan

4.2.2.1 Konfigurasi CodeIgniter

Konfigurasi CodeIgniter merupakan folder yang di dalamnya terdapat pengaturan-pengaturan yang berkaitan dengan sistem yang akan dibuat. Ada beberapa pengaturan atau konfigurasi yang harus dilakukan, diantaranya:

1. Konfigurasi file config.php

   pada file ini yang perlu dilakukan konfigurasi adalah penentuan *base* URI aplikasi. Berikut adalah potongan kode program dari file config.php

```
$config['base_url'] = 'http://localhost/nekatun/';
```

Kode Program 1. config.php

2. Konfigurasi file database.php

   Konfigurasi ini mutlak dibutuhkan karena kedepan akan dibangun aplikasi dengan dukungan *database.* Berikut adalah beberapa konfigurasi yang perlu disesuaikan.

```
$active_group = 'default';
$active_record = TRUE;

$db['default']['hostname'] = 'localhost';
$db['default']['username'] = 'harry';
$db['default']['password'] = '123456';
$db['default']['database'] = 'nekatun';
$db['default']['dbdriver'] = 'mysql';
$db['default']['dbprefix'] = '';
$db['default']['pconnect'] = TRUE;
$db['default']['db_debug'] = TRUE;
$db['default']['cache_on'] = FALSE;
$db['default']['cachedir'] = '';
$db['default']['char_set'] = 'utf8';
$db['default']['dbcollat'] = 'utf8_general_ci';
$db['default']['swap_pre'] = '';
$db['default']['autoinit'] = TRUE;
$db['default']['stricton'] = FALSE;
```

Kode Program 2. database.php

3. Konfigurasi file autoload.php

   Konfigurasi ini berfungsi untuk memanggil berbagai pustaka(*library*) helper dan config yang akan digunakan secara umum sehingga tidak perlu memanggil pustaka tersebut secara berulang-ulang pada controller yang akan dibuat. Berikut adalah potongan kode autoload.php.

```
$autoload['libraries'] = array('database','session','parser','cart',
'template','autentifikasi');

$autoload['helper'] = array('url','form','inflector');

$autoload['config'] = array('config_store');
```

Kode Program 3. autoload.php

4. Konfigurasi file config_store.php

   File ini yang akan menampung atau mencatat konfigurasi yang akan dibutuhkan oleh aplikasi penjualan *online* yang akan dibuat. Ada dua macam konfigurasi, yaitu konfigurasi tentang lokasi folder asset dan template.

```
$config['template_base'] = 'template/';
$config['asset_base'] = 'asset/';
```

Kode Program 4. config_store.php

4.2.2.2 Fungsi Pustaka autentifikasi.php

Pustaka ini akan mempunyai beberapa fungsi, di antaranya:

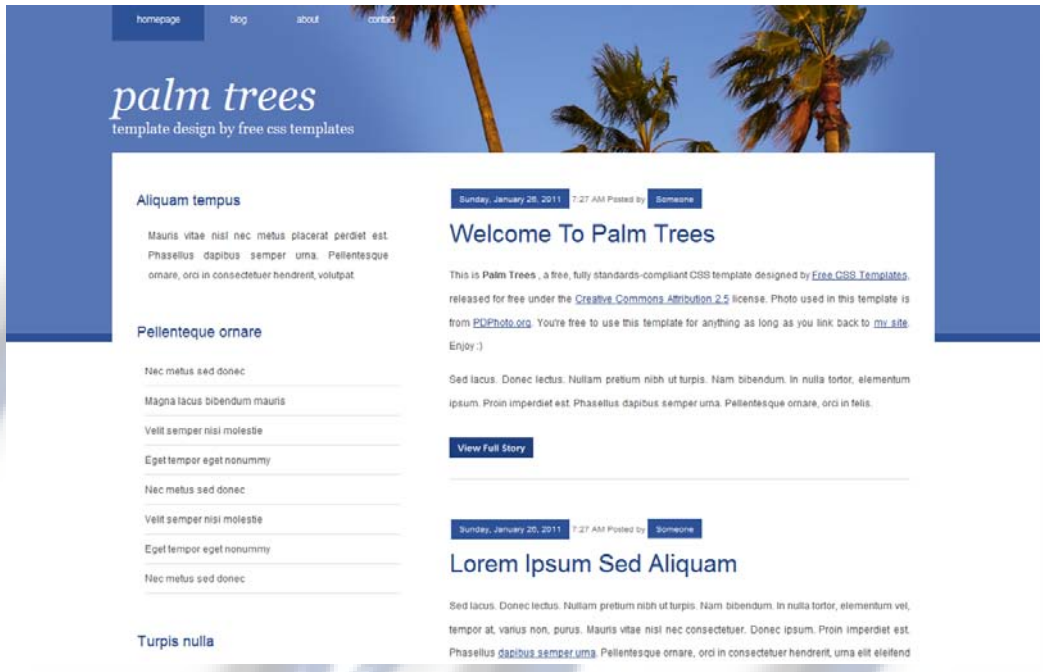
a. Proses login atau autentifikasi dengan disertai penyimpanan data session
b. Proses logout
c. Penambahan data pengguna (*user*) baru.
d. Perubahan data pengguna yang sudah ada
e. Pengecekkan status login
f. Pengecekan level atau *user role*

Dalam file autentifikasi.php ini ada beberapa fungsi (*method*) yang ada pada pustaka ini, yaitu:

1. Fungsi login() digunakan untuk membantu dalam proses pengenalan dengan menggunakan dua parameter, yaitu username dan password.
2. Fungsi logout() akan menghapus session login yang telah dibuat para proses login sebelumnya.
3. Fungsi tambah() digunakan pada proses pendaftaran pengguna baru.
4. Fungsi ubah() digunakan untuk merubah data pengguna yang ada di dalam database dengan menggunakan parameter id sebagai petunjuk data
5. Fungsi sudah_login() akan melakukan pencetakan terhadap status pengunjung.
6. Fungsi role() menggunakan parameter berupa array yang memiliki array value berupa level yang akan disesuaikan dengan jenis level yang ada pada session.

4.2.2.3 Fungsi Pustaka template

Pada aplikasi yang akan dikembangkan peneliti menggunakan template CSS untuk halaman admin dan halaman publik. Berikut template yang dipilih dalam pengembangan aplikasi.



Gambar 4.3 Template palmtrees

Pada template palmtrees tersebut terdapat perubahan dan tambahan dalam parancangan pengembangan yang dilakukan. Berikut tabel penjelasan mengenai perbedaan template palmtrees dan template nekatun.

Tabel 4.1 Perbandingan template yang dikembangkan

Menu Aplikasi	**Menu Template Awal**	**Menu Template dikembangkan**
Admin	Pada template ini hanya terdapat menu bar	Pada template ini dikembangkan menu berikut: 1. Login 2. Manajemen Kategori 3. Manajemen Produk 4. Manajemen User 5. Laporan Status Pemesanan 6. Pengaturan Halaman Depan

Publik	Pada template ini terdapat menu: 1. Homepage 2. About 3. Blog 4. Contact	Pada template ini yang diambil adalah: 1. homepage 2. contact Template yang ditambah adalah sebagai berikut: 1. registrasi 2. login 3. cara pemesanan 4. mengubah profil 5. record pemesanan 6. mengubah akun 7. galeri produk 8. pemesanan pada poin-poin di atas yang akan dikembangkan ke dalam aplikasi penjualan *online* dengan menggunakan CodeIgniter.

Pustaka template ini digunakan secara langsung pada saat aplikasi dijalankan. Berikut fungsi yang digunakan dalam pustaka template:

1. Fungsi get_metadata() akan mengambilkan nilai berupa *string* yang terdiri dari rangkaian data link CSS dan file JavaScript.
2. Fungsi get_judul() digunakan untuk memanggil judul yang telah diset sebelumnya.
3. Fungsi set_template() digunakan untuk memilih file template yang akan digunakan pada tampilan.

4. Fungsi set_judul() digunakan untuk memberikan input judul ke dalam pustaka.
5. Fungsi set_css() digunakan untuk memasukkan nama-nama file CSS yang akan dipanggil

4.2.2.4 Routes Halaman Admin dan Publik

Routing ini akan menerjemahkan URI yang terdapat di address bar sesuai dengan role yang ditulis pada file routes.php yang terdapat di dalam folder config. Dengan menggunakan fasilitas ini dapat memanipulasi susunan direktori dan URI yang akan digunakan, terutama pada controller. Berikut potongan kode dari file routes.php untuk halaman admin.

```
$route['admin'] = 'juragan/dashboard';
$route['admin/logout'] = 'juragan/dashboard/logout';
$route['admin/(:any)'] = 'juragan/$1';
```

Kode Program 5. routes.php untuk halaman admin

Berikut potongan kode dari file routes.php untuk halaman publik.

```
$route['produk/(:any)/(:any)'] = 'store/produk/$2';
$route['produk/(:any)'] = 'store/kategori/$1';
```

Kode Program 6. routes.php untuk halaman publik

4.3 Perancangan Model UML

4.3.1 Skenario *Use Case* Aplikasi Penjualan Berbasis *Website*

Skenario (*flow of event*) untuk masing-masing *use case* dari aplikasi penjualan *online*, sebagai berikut :

1. Skenario *Use Case Login* Admin

Berikut ini adalah skenario *use case login* admin dari aplikasi penjualan *online* :

Tabel 4.2 Skenario *Use Case Login* Admin

Nomor	SUC-001	
Nama *Use Case*	*Login* Admin (Admin harus melakukan *login* terlebih dahulu sebelum masuk ke halaman admin dengan memasukkan *username* dan *password*)	
Aktor	Admin	
Type	*Primary*	
Goal	Untuk mengatur segala informasi pada sistem	
Precondition	Memasukan *username* dan *password*	
Postcondition	Berhasil masuk ke dalam halaman *admin*, bisa menambah, mengubah, menghapus, atau meng*update* informasi-informasi yang akan ditampilkan pada sistem.	
Steps	Aktor	Sistem
	1. Memasukkan *username* dan *password*	
		2. Melakukan validasi *username* dan *password*
		3. Menampilkan halaman admin

2. Skenario *Use Case Login User*

Berikut ini adalah skenario *use case login user* dari aplikasi penjualan *online* :

Tabel 4.3 Skenario *Use Case Login User*

Nomor	SUC-002	
Nama *Use Case*	*Login User* (*User* harus melakukan *login* terlebih dahulu sebelum melakukan pemesanan)	
Aktor	*User*	
Type	*Primary*	
Goal	*User* melakukan *login* setelah melakukan registrasi untuk melakukan pemesanan.	
Precondition	Memasukkan *username* dan *password*	
Postcondition	Berhasil masuk ke dalam halaman publik *user*.	
Steps	Aktor	Sistem
	1. Memasukkan *username* dan *password*	
		2. Melakukan validasi *username* dan *password*
		3. Menampilkan halaman publik

3. Skenario *Use Case* Menambah Ketegori

Berikut ini adalah skenario *use case* menambah kategori dari aplikasi penjualan *online* :

Tabel 4.4 Skenario *Use Case* Menambah Kategori

Nomor	SUC-003
Nama *Use Case*	Menambah Kategori (Merupakan proses menambah kategori produk)
Aktor	Admin

Type	*Primary*	
Goal	Menambahkan kategori produk	
Precondition	Masuk pada halaman kategori	
Postcondition	Berhasil menambah kategori	
Steps	Aktor	Sistem
	1. Klik "Kategori"	
		2. Menampilkan halaman kategori
	3. Klik "Tambah Kategori"	
		4. Menampilkan form penambahan kategori
	5. Mengisi form kategori	
	6. Klik "Submit"	
		7. Menyimpan kategori baru

4. Skenario *Use Case* Mengubah Kategori

Berikut ini adalah skenario *use case* mengubah kategori dari aplikasi penjualan *online* :

Tabel 4.5 Skenario *Use Case* Mengubah Kategori

Nomor	SUC-004
Nama *Use Case*	Mengubah Kategori (Merupakan proses mengubah kategori produk)
Aktor	Admin
Type	*Primary*
Goal	Mengubah kategori produk
Precondition	Masuk pada halaman kategori
Postcondition	Berhasil mengubah kategori

Steps	Aktor	Sistem
	1. Klik “Kategori”	
		2. Menampilkan halaman kategori
	3. Klik “Ubah”	
		4. Menampilkan form mengubah kategori
	5. Mengubah kategori	
	6. Klik “Simpan”	
		7. Kategori diubah

5. Skenario *Use Case* Menghapus Kategori

Berikut ini adalah skenario *use case* menghapus kategori dari aplikasi penjualan *online* :

Tabel 4.6 Skenario *Use Case* Menghapus Kategori

Nomor	SUC-005	
Nama *Use Case*	Menghapus Kategori (Merupakan proses menghapus kategori produk)	
Aktor	Admin	
Type	*Primary*	
Goal	Menghapus kategori produk	
Precondition	Masuk pada halaman kategori	
Postcondition	Berhasil menghapus kategori	
Steps	Aktor	Sistem
	1. Klik “Kategori”	
		2. Menampilkan halaman kategori

	3. Klik "Hapus" pada kategori yang dipilih	
		4. Kategori terhapus

6. Skenario *Use Case* Menambah Produk

Berikut ini adalah skenario *use case* menambah produk dari aplikasi penjualan *online*:

Tabel 4.7 Skenario *Use Case* Menambah Produk

Nomor	SUC-006	
Nama *Use Case*	Menambah Produk (Merupakan proses menambah produk pada setiap ketegori)	
Aktor	Admin	
Type	*Primary*	
Goal	Menambahkan produk baru	
Precondition	Masuk pada halaman produk	
Postcondition	Berhasil menambah produk baru	
Steps	Aktor	Sistem
	1. Klik "Produk"	
		2. Menampilkan halaman produk
	3. Klik "Tambah Produk Baru"	
		4. Menampilkan form penambahan produk
	5. Mengisi data produk	
	6. Klik "Submit"	
		7. Menyimpan produk baru

7. Skenario *Use Case* Mengubah Produk

Berikut ini adalah skenario *use case* mengubah produk dari aplikasi penjualan *online*:

Tabel 4.8 Skenario *Use Case* Mengubah Produk

Nomor	SUC-007	
Nama *Use Case*	Mengubah Produk (Merupakan proses mengubah data produk pada setiap kategori)	
Aktor	Admin	
Type	*Primary*	
Goal	Mengubah data produk	
Precondition	Masuk pada halaman produk	
Postcondition	Berhasil mengubah produk	
Steps	Aktor	Sistem
	1. Klik "Produk"	
		2. Menampilkan halaman produk
	3. Klik "Ubah" pada produk yang akan diubah	
		4. Menampilkan form perubahan produk
	5. Mengubah data produk	
	6. Klik "Simpan"	
		7. Mengubah produk

8. Skenario *Use Case* Menghapus Produk

Berikut ini adalah skenario *use case* menghapus produk dari aplikasi penjualan *online*:

Tabel 4.9 Skenario *Use Case* Menghapus Produk

Nomor	SUC-008	
Nama *Use Case*	Menghapus Produk (Merupakan proses menghapus produk pada setiap ketegori)	
Aktor	Admin	
Type	*Primary*	
Goal	Menghapus produk	
Precondition	Masuk pada halaman produk	
Postcondition	Berhasil menghapus produk	
Steps	Aktor	Sistem
	1. Klik "Produk"	
		2. Menampilkan halaman produk
	3. Klik "Hapus" pada produk yang akan dihapus	
		4. Produk terhapus

9. Skenario *Use Case* Registrasi *User*

Berikut ini adalah skenario *use case* registrasi *user* dari aplikasi penjualan *online*:

Tabel 4.10 Skenario *Use Case* Registrasi *User*

Nomor	SUC-009
Nama *Use Case*	Registrasi *User* (Merupakan proses pendaftaran *user*)

Aktor	*User*	
Type	*Primary*	
Goal	Menambah *user* baru	
Precondition	Masuk pada halaman *login*	
Postcondition	Berhasil mendaftarkan *user* baru	
Steps	Aktor	Sistem
	1. Klik daftar "disini" pada halaman *login*.	
		2. Menampilkan form pendaftaran
	3. Masukkan *username*, email dan *password*	
	4. Klik "Registrasi"	
		5. Menambah *user* baru

10. Skenario *Use Case* Mengubah Profil *User*

Berikut ini adalah skenario *use case* mengubah profil *user* dari aplikasi penjualan *online*:

Tabel 4.11 Skenario *Use Case* Mengubah Profil *User*

Nomor	SUC-010
Nama *Use Case*	Mengubah Profil *User* (Merupakan proses mengubah data *user*)
Aktor	*User*
Type	*Primary*
Goal	Mengubah data *user*
Precondition	Masuk ke menu profil
Postcondition	Berhasil mengubah data *user*

Steps	Aktor	Sistem
	1. Klik “Profil”	
		2. Menampilkan profil *user*
	3. Mengisi perubahan data *user*	
	4. Klik “Simpan”	
		5. Data profil tersimpan

11. Skenario *Use Case* Mengubah Akun *User*

Berikut ini adalah skenario *use case* mengunah akun *user* dari aplikasi penjualan *online*:

Tabel 4.12 Skenario *Use Case* Mengubah Akun *User*

Nomor	SUC-011	
Nama *Use Case*	Mengubah Akun *User* (Merupakan proses mengubah akun *user* pada saat *login*)	
Aktor	Admin	
Type	*Primary*	
Goal	Mengubah akun *user*	
Precondition	Masuk ke halaman edit akun	
Postcondition	Berhasil mengubah akun	
Steps	Aktor	Sistem
	1. Klik “edit akun”	
		2. Menampilkan halaman edit akun
	3. Masukkan *username*, email dan *password* yang akan diubah	

	4. Klik “Update”	
		5. Akun baru tersimpan

12. Skenario *Use Case* Menghapus *User*

Berikut ini adalah skenario *use case* menghapus *user* dari aplikasi penjualan *online*:

Tabel 4.13 Skenario *Use Case* Menghapus *User*

Nomor	SUC-012	
Nama *Use Case*	Menghapus *User* (Merupakan proses menghapus *user*)	
Aktor	Admin	
Type	*Primary*	
Goal	Menghapus *User*	
Precondition	Masuk ke halaman *user*	
Postcondition	Berhasil menghapus *user*	
Steps	Aktor	Sistem
	1. Klik “User”	
		2. Menampilkan daftar *user*
	3. Klik “Hapus” pada *user* yang akan dihapus	
		4. *User* terhapus

13. Skenario *Use Case* Status Pemesanan

Berikut ini adalah skenario *use case* status pemesanan dari aplikasi penjualan *online*:

Tabel 4.14 Skenario *Use Case* Status Pemesanan

Nomor	SUC-013	
Nama *Use Case*	Status Pemesanan (Merupakan proses mengubah status pemesanan konsumen)	
Aktor	Admin	
Type	*Primary*	
Goal	Mengubah status pemesanan	
Precondition	Masuk ke halaman laporan	
Postcondition	Berhasil mengubah status pemesanan	
Steps	Aktor	Sistem
	1. Masuk ke halaman laporan	
		2. Menampilkan halaman laporan pemesanan
	3. Klik "Detail" pada laporan pemesanan yang akan diubah	
		4. Menampilkan detail data pesanan
	5. Pilih status pemesanan sekarang	
	6. Klik "Simpan Perubahan"	
		7. Status pemesanan berubah

14. Skenario *Use Case* Mengubah Halaman Depan

Berikut ini adalah skenario *use case* mengubah halaman depan dari aplikasi penjualan *online*:

Tabel 4.15 Skenario *Use Case* Mengubah Halaman Depan

Nomor	SUC-014	
Nama *Use Case*	Mengubah Halaman Depan (Merupakan proses mengubah halaman depan *website*)	
Aktor	Admin	
Type	*Primary*	
Goal	Mengubah halaman depan *website*	
Precondition	Masuk ke halaman admin	
Postcondition	Berhasil mengubah tampilan halaman depan *website*	
Steps	Aktor	Sistem
	1. Klik "Pengaturan"	
		2. Menampilkan halaman pengaturan
	3. Ubah data situs atau tampilan halaman depan *website*	
	4. Klik "Simpan"	
		5. Halaman depan tersimpan

15. Skenario *Use Case* Homepage

Berikut ini adalah skenario *use case* homepage dari aplikasi penjualan *online*:

Tabel 4.16 Skenario *Use Case* Homepage

Nomor	SUC-015
Nama *Use Case*	Homepage (Merupakan proses menampilkan halaman depan website)
Aktor	*User*
Type	*Primary*

Goal	Menampilkan halaman depan website	
Precondition	Masukkan alamat website	
Postcondition	Berhasil menampilkan halaman depan	
Steps	Aktor	Sistem
	1. Masuk atau klik "homepage"	
		2. Menampilkan halaman depan

16. Skenario *Use Case* Contact

Berikut ini adalah skenario *use case* contact dari aplikasi penjualan *online*:

Tabel 4.17 Skenario *Use Case* Contact

Nomor	SUC-016	
Nama *Use Case*	Contact (Merupakan proses menampilkan kontak dari perusahaan)	
Aktor	*User*	
Type	*Primary*	
Goal	Menampilkan contact person	
Precondition	Masuk ke	
Postcondition	Berhasil mengubah data *user*	
Steps	Aktor	Sistem
	1. Klik "kontak"	
		2. Menampilkan kontak yang dapat dihubungi atau bisa langung *chat*

17. Skenario *Use Case* Cara Pemesanan

Berikut ini adalah skenario *use case* cara pemesanan dari aplikasi penjualan *online*:

Tabel 4.18 Skenario *Use Case* Cara Pemesanan

Nomor	SUC-017	
Nama *Use Case*	Cara Pemesanan (Merupakan proses menampilkan langkah pemesanan yang harus dilakukan *user*)	
Aktor	*User*	
Type	*Primary*	
Goal	Menampilkan cara pemesanan	
Precondition	Masuk ke website	
Postcondition	Berhasil menampilkan menu cara pemesanan	
Steps	Aktor	Sistem
	1. Klik "cara pemesanan"	
		2. Menampilkan cara-cara pemesanan pada website

18. Skenario *Use Case Record* Pemesanan

Berikut ini adalah skenario *use case record* pemesanan dari aplikasi penjualan *online*:

Tabel 4.19 Skenario *Use Case Record* Pemesanan

Nomor	SUC-018
Nama *Use Case*	*Record* Pemesanan (Merupakan proses menampilkan riwayat belanja *user* beserta menampilkan status pemesanannya)
Aktor	*User*

Type	*Primary*	
Goal	Menampilkan status pemesanan	
Precondition	Masuk ke menu record	
Postcondition	Berhasil menampilkan status pemesanan *user*	
Steps	Aktor	Sistem
	1. Klik "record"	
		2. Menampilkan riwayat pemesanan

19. Skenario *Use Case* Galeri Produk

Berikut ini adalah skenario *use case* galeri produk dari aplikasi penjualan *online*:

Tabel 4.20 Skenario *Use Case* Galeri Produk

Nomor	SUC-019	
Nama *Use Case*	Galeri Produk (Merupakan proses menampilkan produk-produk berdasarkan kategori yang ada)	
Aktor	*User*	
Type	*Primary*	
Goal	Menampilkan produk-produk yang dijual	
Precondition	Pilih kategori produk yang diinginkan	
Postcondition	Berhasil menampilkan produk-produk yang dijual	
Steps	Aktor	Sistem
	1. Klik nama kategori yang dipilih	
		2. Menampilkan galeri produk sesuai kategori

20. Skenario *Use Case* Pemesanan

Berikut ini adalah skenario *use case* pemesanan dari aplikasi penjualan *online*:

Tabel 4.21 Skenario *Use Case* Pemesanan

Nomor	SUC-020	
Nama *Use Case*	Pemesanan (Merupakan proses transaksi pembelian produk)	
Aktor	*User*	
Type	*Primary*	
Goal	Dapat memesan produk.	
Precondition	Masuk ke halaman *website*	
Postcondition	Berhasil melakukan pemesanan produk dengan muncul pesan "Data pesanan telah kami terima".	
Steps	Aktor	Sistem
	1. Masuk ke halaman produk setiap kategori	
		2. Menampilkan halaman produk
	3. Pilih produk-produk yang akan dibeli dengan mengklik produk tersebut	
	4. Ketikkan Kuantitas produk yang dibeli	
	5. Klik "Add to Cart"	
		6. Produk masuk ke kantong belanja

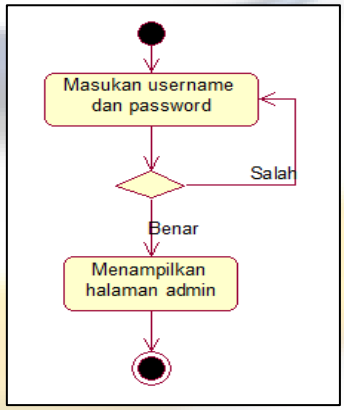
	7. Klik “Check Out”	
		8. Menampilkan produk-produk yang dibeli dan jumlah biaya yang harus dibayar
	9. Klik “Lanjut”	
		10. Menampilkan data alamat pengiriman.
	11. Klik “Pesan Sekarang”	
		12. Data pesanan sudah terkirim

4.3.2 *Activity Diagram*

Aliran kegiatan atau kerja yang terjadi di dalam aplikasi penjualan *online* ini akan digambarkan dalam *activity diagram,* berikut penjelasannya di bawah ini:

1. *Activity Diagram Use Case Login* Admin

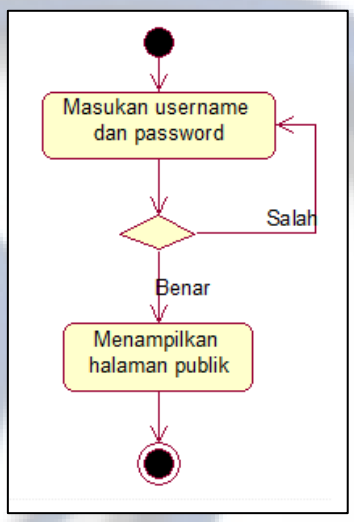
Berikut adalah *activity diagram login* admin dari aplikasi penjualan *online*:



Gambar 4.4 *Activity Diagram Login* Admin

2. ***Activity Diagram Use Case Login User***

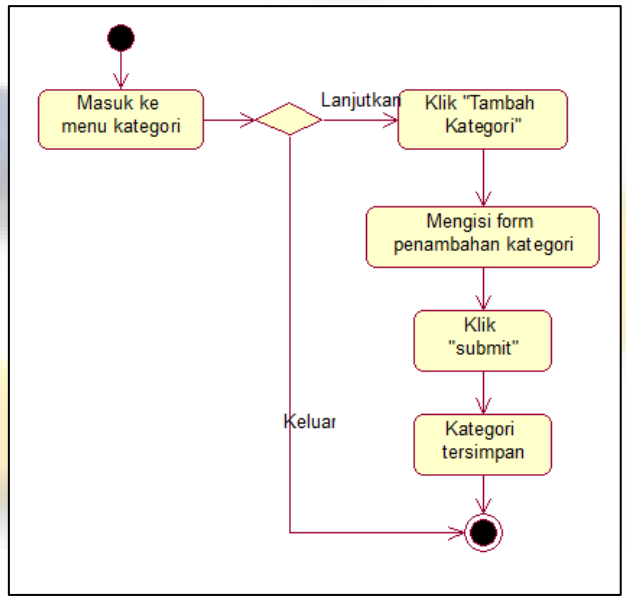
Berikut adalah *activity diagram login* user dari aplikasi penjualan *online*:



Gambar 4.5 *Activity Diagram Login User*

3. ***Activity Diagram Use Case* Menambah Kategori**

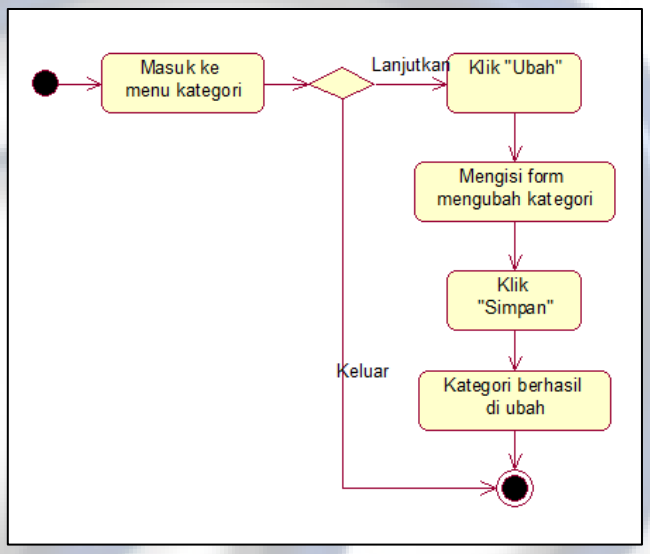
Berikut adalah *activity diagram* menambah kategori dari aplikasi penjualan *online*:



Gambar 4.6 *Activity Diagram* Menambah Kategori

4. *Activity Diagram Use Case* Mengubah Kategori

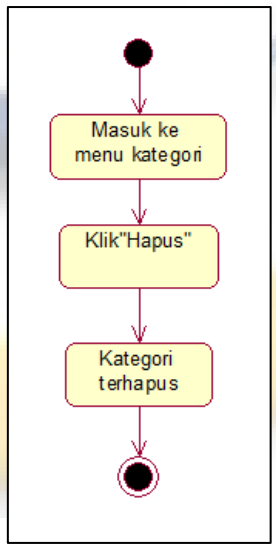
Berikut adalah *activity diagram* mengubah kategori dari aplikasi penjualan *online*:



Gambar 4.7 *Activity Diagram* Mengubah Kategori

5. *Activity Diagram Use Case* Menghapus Kategori

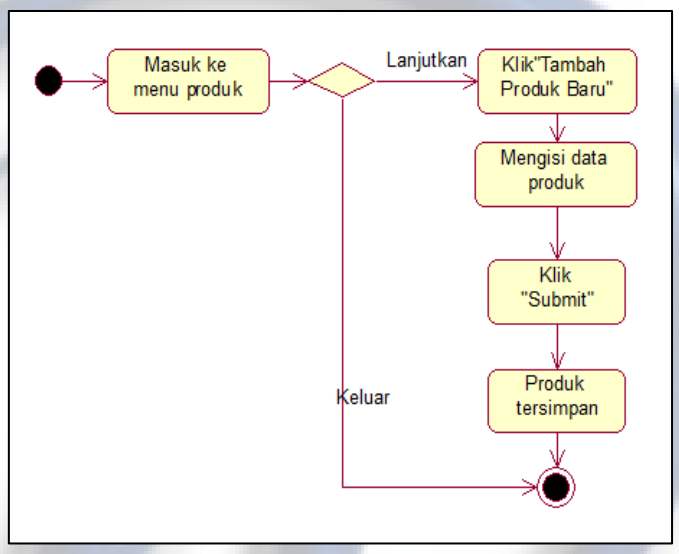
Berikut adalah *activity diagram* mengubah kategori dari aplikasi penjualan *online*:



Gambar 4.8 *Activity Diagram* Menghapus Kategori

6. *Activity Diagram Use Case* Menambah Produk

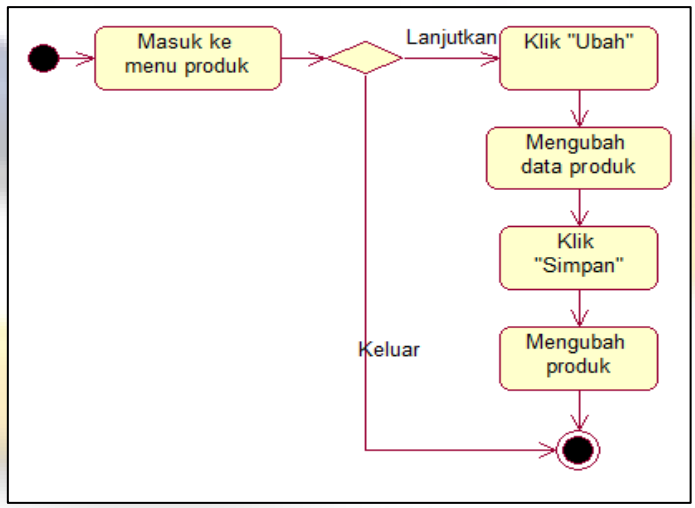
Berikut adalah *activity diagram* menambah produk dari aplikasi penjualan *online*:



Gambar 4.9 *Activity Diagram* Menambah Produk

7. *Activity Diagram Use Case* Mengubah Produk

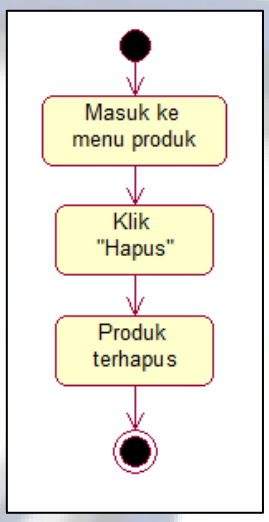
Berikut adalah *activity diagram* mengubah produk dari aplikasi penjualan *online*:



Gambar 4.10 *Activity Diagram* Mengubah Produk

8. *Activity Diagram Use Case* Menghapus Produk

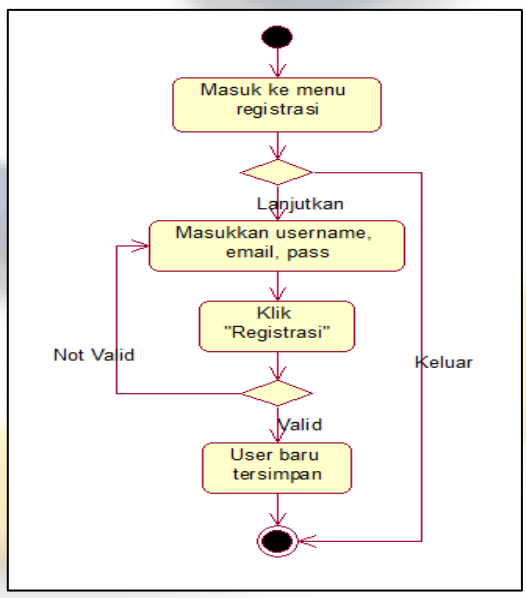
Berikut adalah *activity diagram* menghapus produk dari aplikasi penjualan *online*:



Gambar 4.11 *Activity Diagram* Menghapus Produk

9. *Activity Diagram Use Case* Registrasi *User*

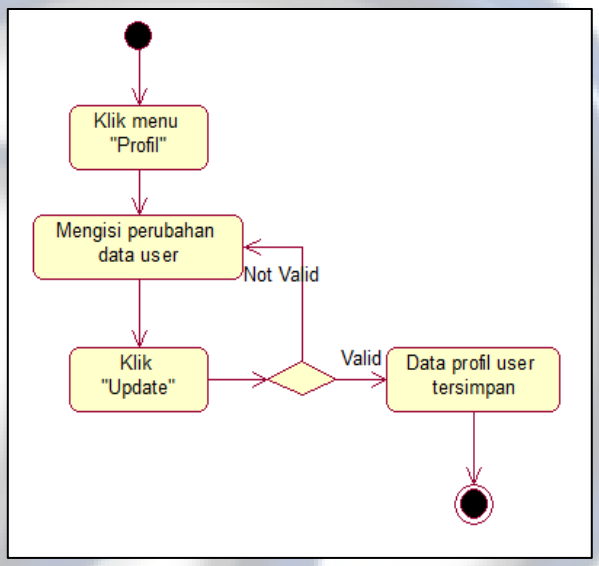
Berikut adalah *activity diagram* registrasi *user* dari aplikasi penjualan *online*:



Gambar 4.12 *Activity Diagram* Registrasi *User*

10. *Activity Diagram Use Case* Mengubah Profil *User*

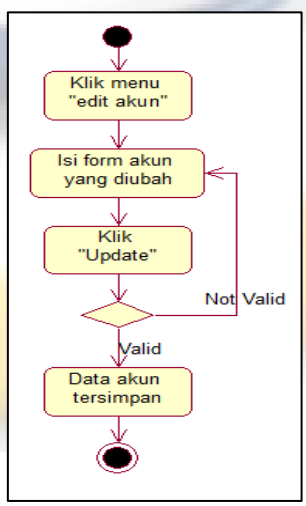
Berikut adalah *activity diagram* mengubah profil *user* dari aplikasi penjualan *online*:



Gambar 4.13 *Activity Diagram* Mengubah Profil *User*

11. *Activity Diagram Use Case* Mengubah Akun *User*

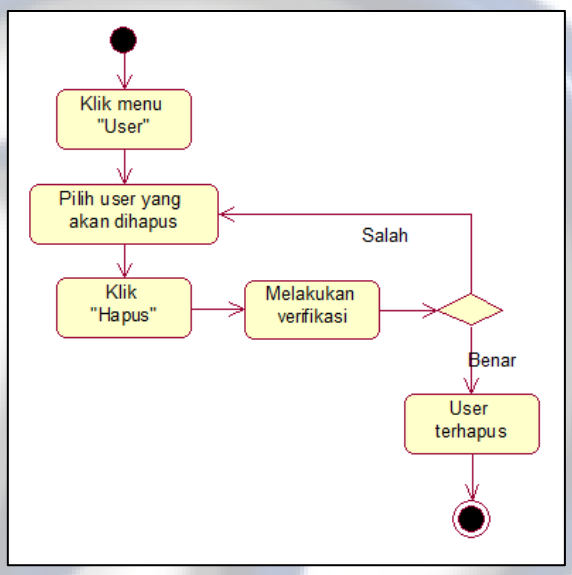
Berikut adalah *activity diagram* mengubah akun *user* dari aplikasi penjualan *online*:



Gambar 4.14 *Activity Diagram* Mengubah Akun *User*

12. *Activity Diagram Use Case* Menghapus *User*

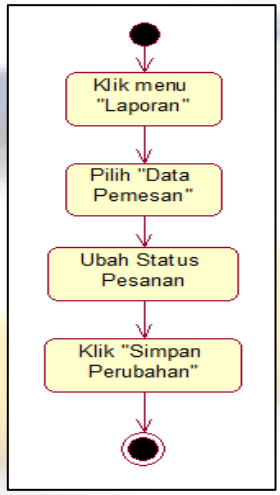
Berikut adalah *activity diagram* menghapus *user* dari aplikasi penjualan *online*:



Gambar 4.15 *Activity Diagram* Menghapus *User*

13. *Activity Diagram Use Case* Status Pemesanan

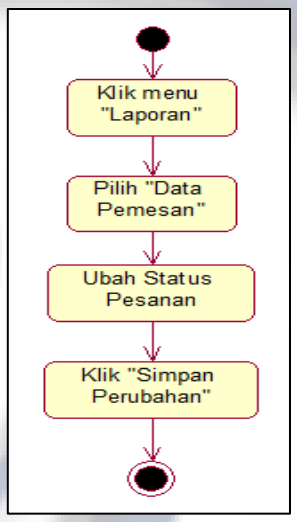
Berikut adalah *activity diagram* status pemesanan dari aplikasi penjualan *online*:



Gambar 4.16 *Activity Diagram* Status Pemesanan

14. *Activity Diagram Use Case* Mengubah Halaman Depan

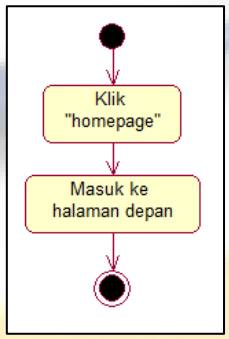
Berikut adalah *activity diagram* mengubah halaman depan dari aplikasi penjualan *online*:



Gambar 4.17 *Activity Diagram* Mengubah Halaman Depan

15. *Activity Diagram Use Case* Homepage

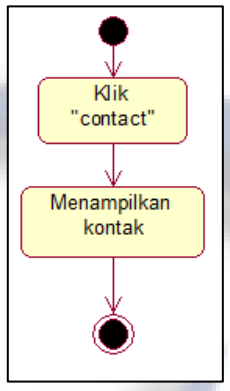
Berikut adalah *activity diagram* homepage dari aplikasi penjualan *online*:



Gambar 4.18 *Activity Diagram* Homepage

16. *Activity Diagram Use Case* Contact

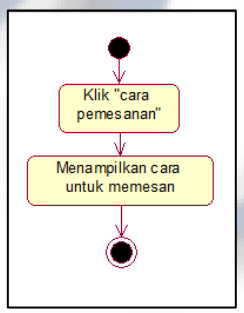
Berikut adalah *activity diagram* Contact dari aplikasi penjualan *online*:



Gambar 4.19 *Activity Diagram* Contact

17. *Activity Diagram Use Case* Cara Pemesanan

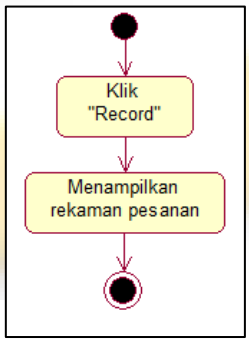
Berikut adalah *activity diagram* cara pemesanan dari aplikasi penjualan *online*:



Gambar 4.20 *Activity Diagram* Cara Pemesanan

18. *Activity Diagram Use Case Record* Pemesanan

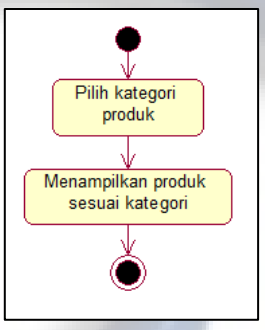
Berikut adalah *activity diagram record* pemesanan dari aplikasi penjualan *online*:



Gambar 4.21 *Activity Diagram Record* Pemesanan

19. *Activity Diagram Use Case* Galeri Produk

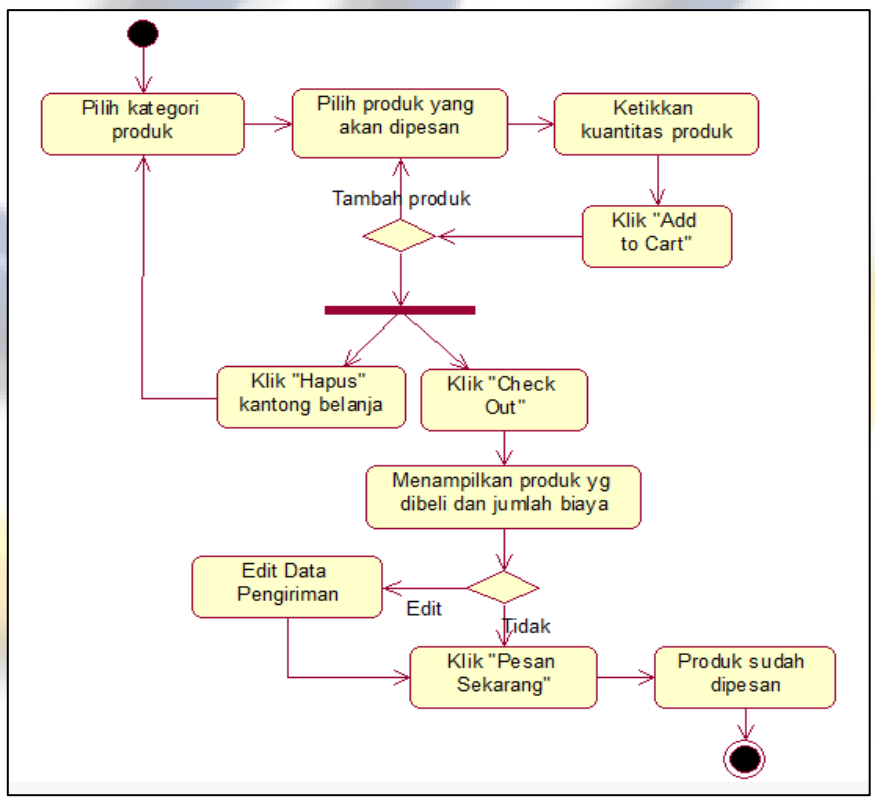
Berikut adalah *activity diagram* galeri produk dari aplikasi penjualan *online*:



Gambar 4.22 *Activity Diagram* Galeri Produk

20. *Activity Diagram Use Case* Pemesanan

Berikut adalah *activity diagram* pemesanan dari aplikasi penjualan *online*:



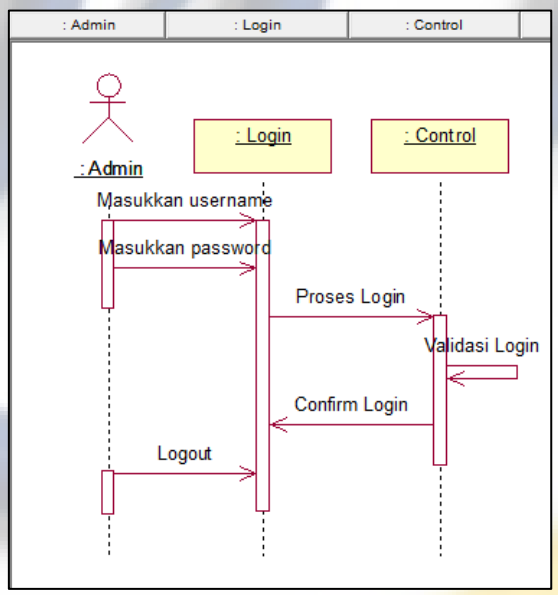
Gambar 4.23 *Activity Diagram* Pemesanan

4.3.3 *Sequence Diagram*

Menggambarkan kolaborasi dinamis antara sejumlah *object*. Kegunaannya untuk menunjukan rangkaian pesan yang dikirim antara *object* juga interaksi antara *object*, sesuatu yang terjadi pada titik tertentu dalam eksekusi sistem. Komponen utama *sequence diagram* terdiri dari atas *object* yang dituliskan dengan kotak segiempat bernama pesan diwakili oleh garis dengan tanda panah dan waktu yang ditunjukkan dengan proses vertical.

1. *Sequence Diagram Login* Admin

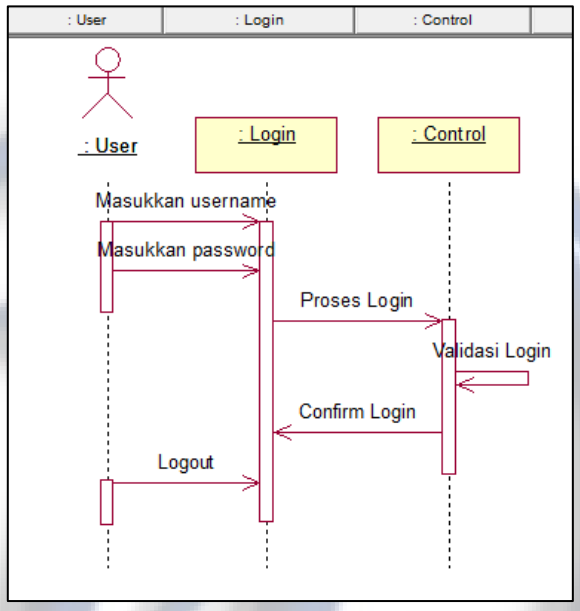
Berikut adalah *sequence diagram login* admin dari aplikasi penjualan *online*:



Gambar 4.24 *Sequence Diagram Login* Admin

2. *Sequence Diagram Login User*

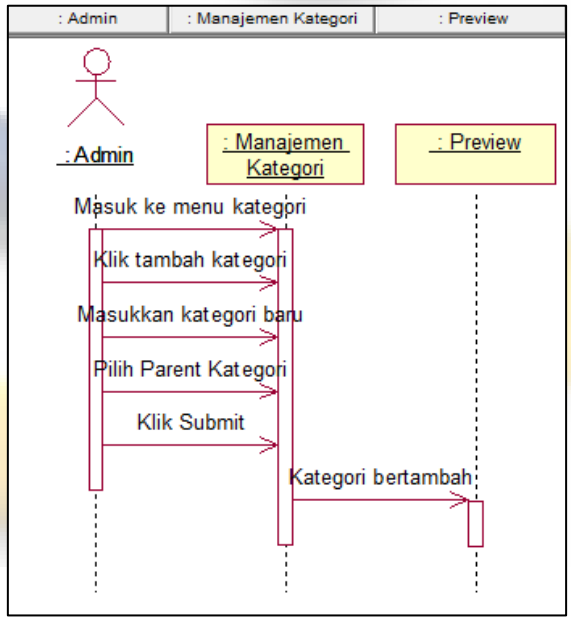
Berikut adalah *sequence diagram login* user dari aplikasi penjualan *online*:



Gambar 4.25 *sequence diagram Login User*

3. *Sequence Diagram* Menambah Kategori

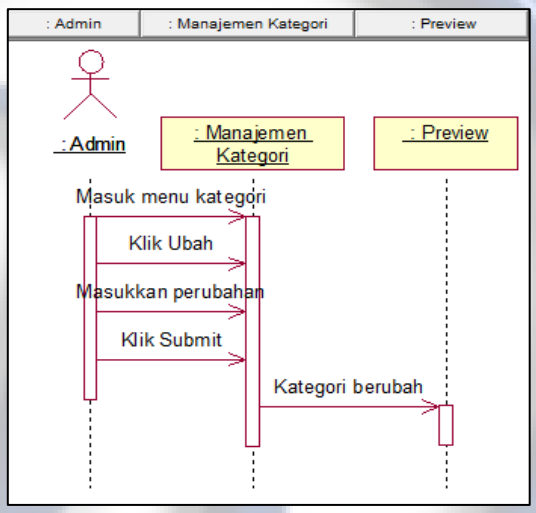
Berikut adalah *sequence diagram* menambah kategori dari aplikasi penjualan *online*:



Gambar 4.26 *Sequence Diagram* Menambah Kategori

4. *Sequence Diagram* Mengubah Kategori

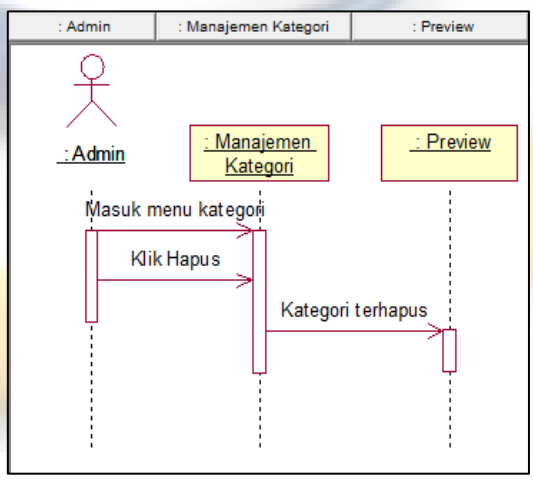
Berikut adalah *sequence diagram* mengubah kategori dari aplikasi penjualan *online*:



Gambar 4.27 *Sequence Diagram* Mengubah Kategori

5. *Sequence Diagram* Menghapus Kategori

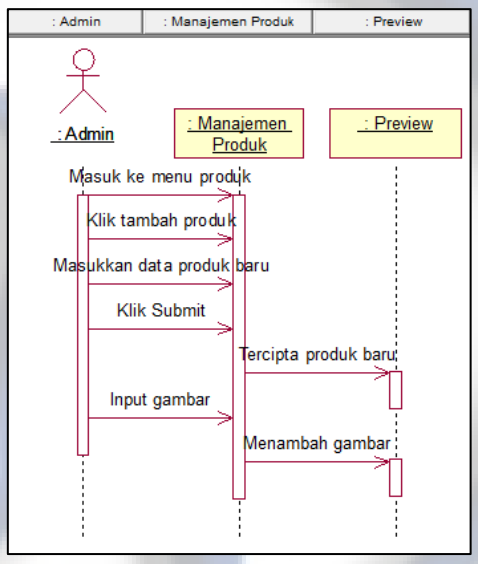
Berikut adalah *sequence diagram* mengubah kategori dari aplikasi penjualan *online*:



Gambar 4.28 *Sequence Diagram* Menghapus Kategori

6. *Sequence Diagram* Menambah Produk

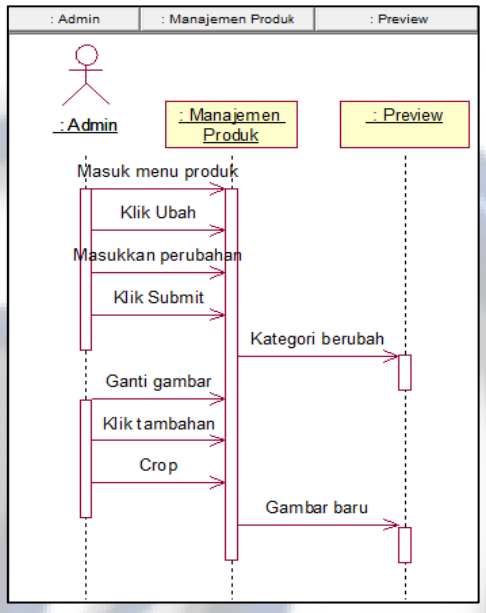
Berikut adalah *sequence diagram* menambah produk dari aplikasi penjualan *online*:



Gambar 4.29 *Sequence Diagram* Menambah Produk

7. *Sequence Diagram* Mengubah Produk

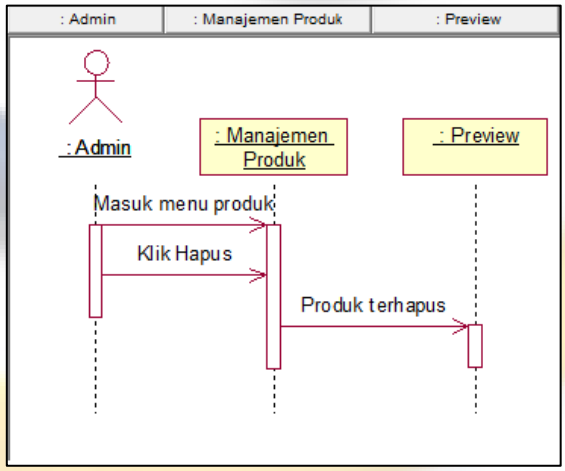
Berikut adalah *sequence diagram* mengubah produk dari aplikasi penjualan *online*:



Gambar 4.30 *Sequence Diagram* Mengubah Produk

8. *Sequence Diagram* Menghapus Produk

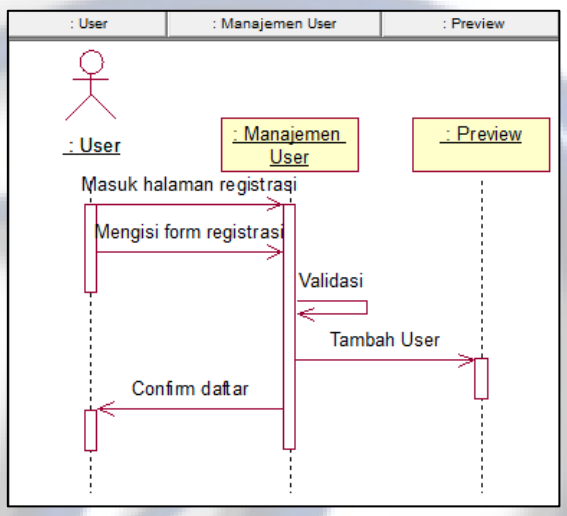
Berikut adalah *sequence diagram* menghapus produk dari aplikasi penjualan *online*:



Gambar 4.31 *Sequence Diagram* Menghapus Produk

9. *Sequence Diagram* Registrasi *User*

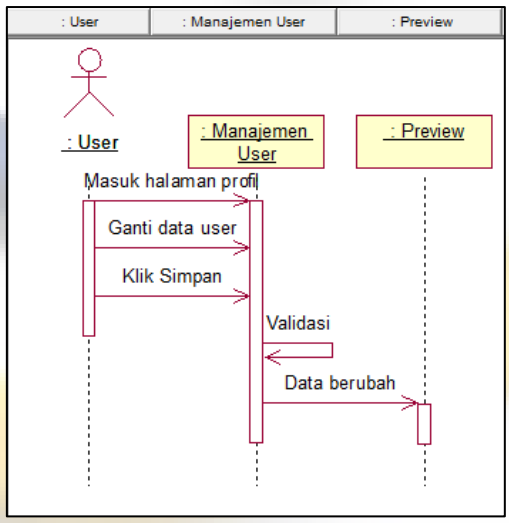
Berikut adalah *sequence diagram* registrasi *user* dari aplikasi penjualan *online*:



Gambar 4.32 *Sequence Diagram* Registrasi *User*

10. *Sequence Diagram* Mengubah Profil *User*

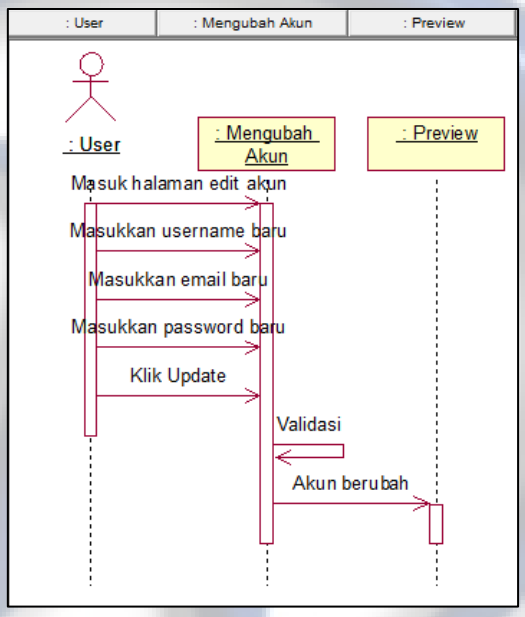
Berikut adalah *sequence diagram* mengubah profil *user* dari aplikasi penjualan *online*:



Gambar 4.33 *Sequence Diagram* Mengubah Profil *User*

11. *Sequence Diagram* Mengubah Akun *User*

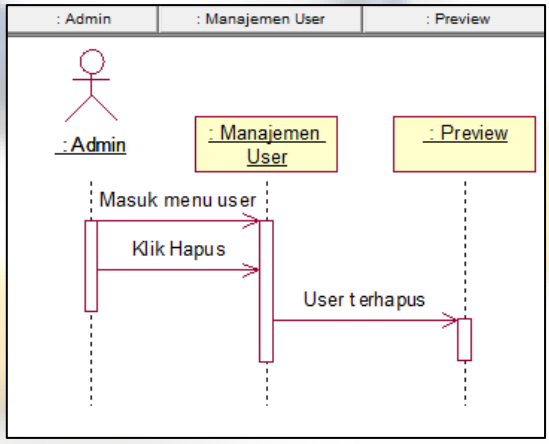
Berikut adalah *sequence diagram* mengubah akun *user* dari aplikasi penjualan *online*:



Gambar 4.34 *Sequence Diagram* Mengubah Akun *User*

12. *Sequence Diagram* Menghapus *User*

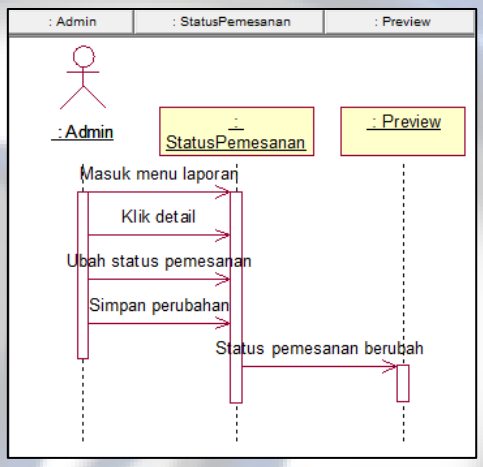
Berikut adalah *sequence diagram* menghapus *user* dari aplikasi penjualan *online*:



Gambar 4.35 *Sequence Diagram* Menghapus *User*

13. *Sequence Diagram* Status Pemesanan

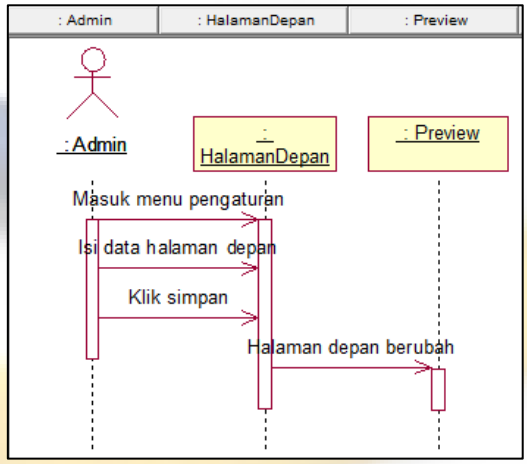
Berikut adalah *sequence diagram* status pemesanan dari aplikasi penjualan *online*:



Gambar 4.36 *Sequence Diagram* Status Pemesanan

14. *Sequence Diagram* Mengubah Halaman Depan

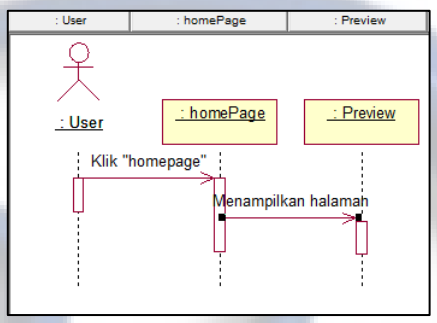
Berikut adalah *sequence diagram* mengubah halaman depan dari aplikasi penjualan *online*:



Gambar 4.37 *Sequence Diagram* Mengubah Halaman Depan

15. *Sequence Diagram* Homepage

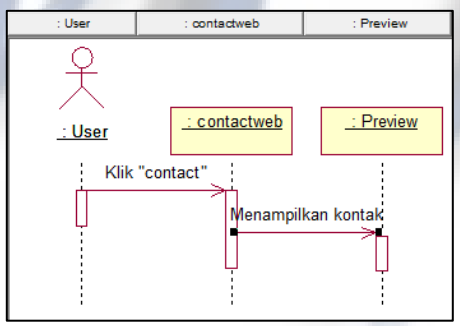
Berikut adalah *sequence diagram* homepage dari aplikasi penjualan *online*:



Gambar 4.38 *Sequence Diagram* Homepage

16. *Sequence Diagram* Contact

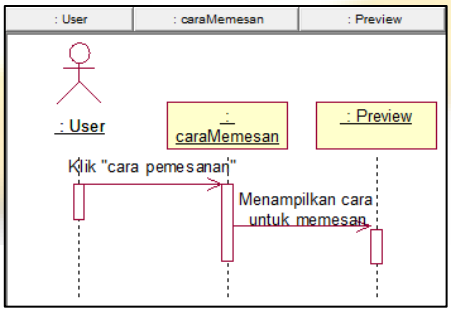
Berikut adalah *sequence diagram* contact dari aplikasi penjualan *online*:



Gambar 4.39 *Sequence Diagram* Contact

17. *Sequence Diagram* Cara Pemesanan

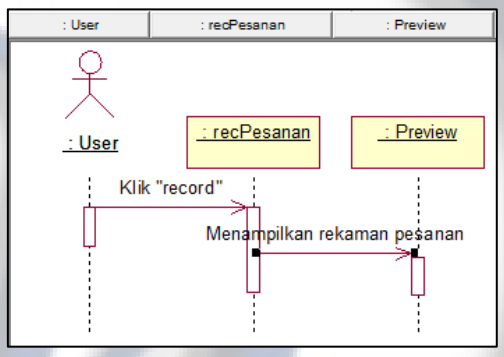
Berikut adalah *sequence diagram* cara pemesanan dari aplikasi penjualan *online*:



Gambar 4.40 *Sequence Diagram* Cara Pemesanan

18. *Sequence Diagram* Record Pemesanan

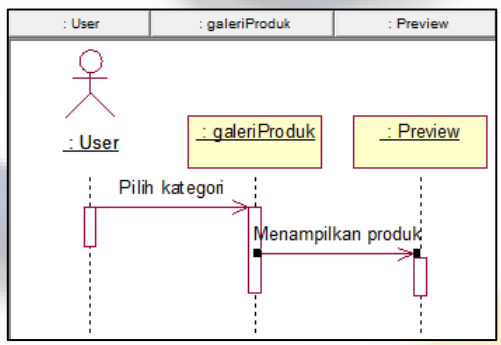
Berikut adalah *sequence diagram record* pemesanan dari aplikasi penjualan *online*:



Gambar 4.41 *Sequence Diagram Record* Pemesanan

19. *Sequence Diagram* Galeri Produk

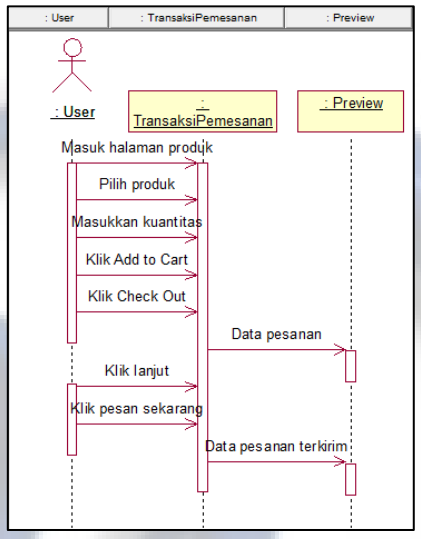
Berikut adalah *sequence diagram* galeri produk dari aplikasi penjualan *online*:



Gambar 4.42 *Sequence Diagram* Galeri Produk

20. *Sequence Diagram* Pemesanan

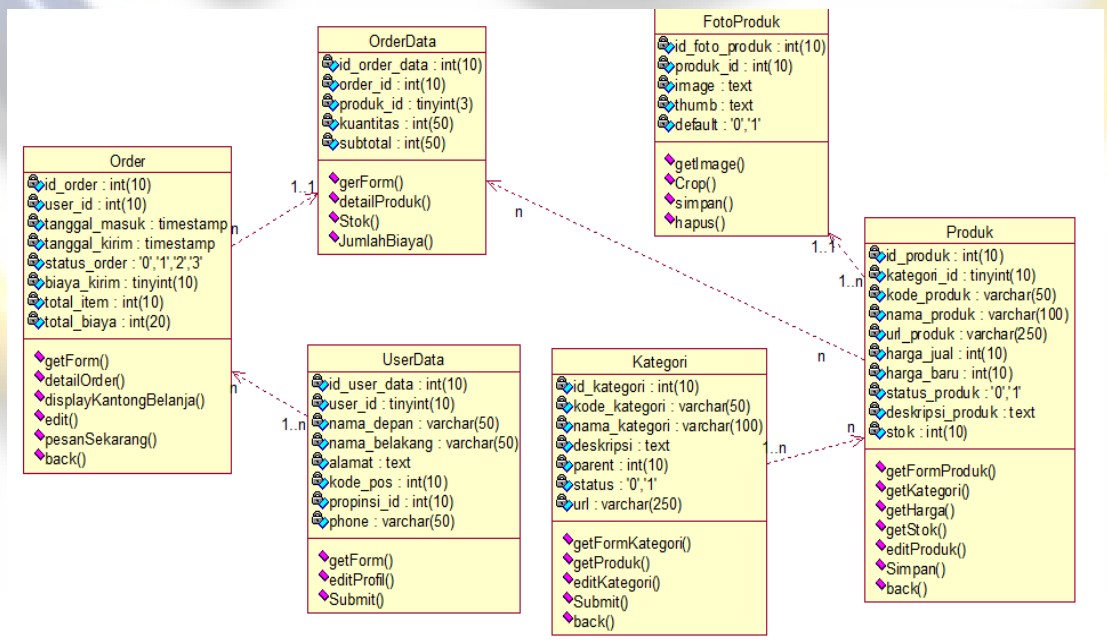
Berikut adalah *sequence diagram* pemesanan dari aplikasi penjualan *online*:



Gambar 4.43 *Sequence Diagram* Pemesanan

4.3.4 *Class Diagram*

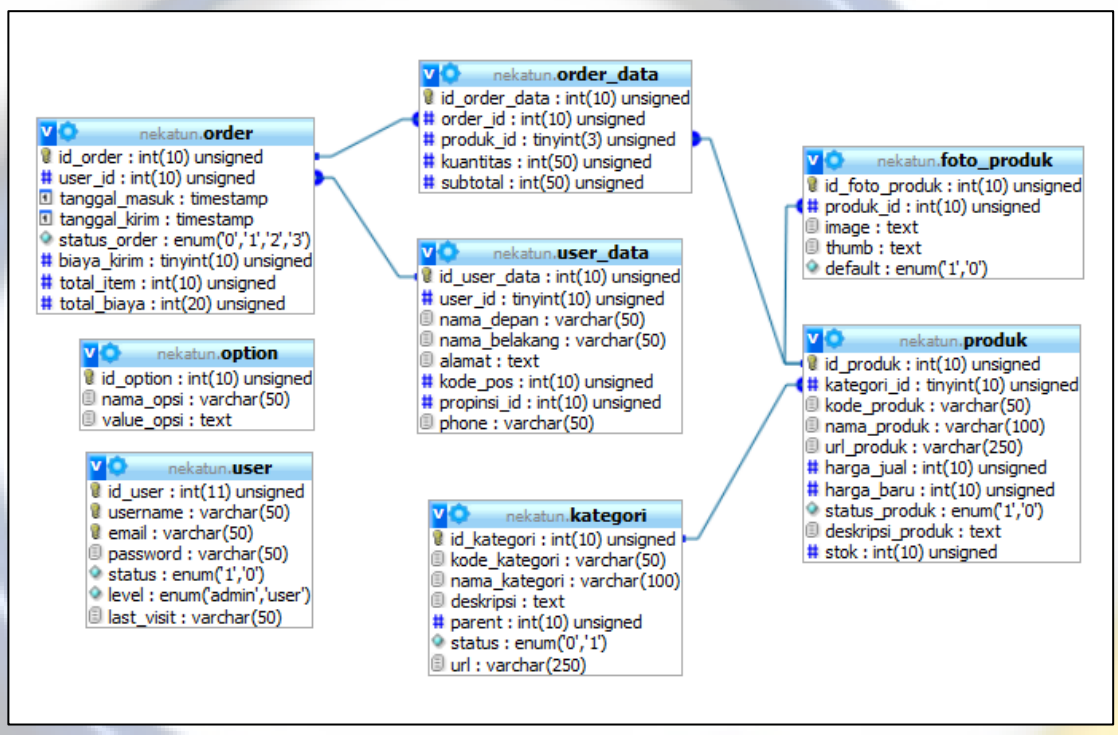
Class diagram adalah diagram yang menggambarkan kelas-kelas dalam sebuah sistem dan hubungannya antara satu dengan yang lain, serta dimasukkan pula atribut dan operasi.



Gambar 4.44 *Class Diagram*

4.3.5 *Entity Relationship Diagram* (ERD)

Entity Relationship Diagram adalah suatu model untuk menjelaskan hubungan antar data dalam basis data berdasarkan objek-objek dasar data yang mempunyai hubungan antar relasi.



Gambar 4.45 *Entity Relationship Diagram*

4.3.6 Kamus Data

Kamus data berfungsi sebagai perancangan dalam *database* aplikasi yang akan dibuat. Berikut di bawah ini penjelasan dari tabel-tabel yang dibuat:

1. Tabel foto_produk

   Keterangan : Sebagai penyimpan gambar yang digunakan untuk merepresentasikan produk yang akan dijual secara *online.*

   Primary Key : id_foto_produk

Tabel 4.22 Struktur tabel foto_produk

Field	Type	Size/Value	Keterangan
id_foto_produk	int	3	Menyimpan data id foto produk
produk_id	int	10	Menyimpan data id produk
image	text	-	Menyimpan data lokasi dimana data gambar suatu produk di simpan
thumb	text	-	Menyimpan data lokasi dimana data gambar thumbnail suatu produk di simpan
default	enum	'1','0'	Menyimpan data default untuk id foto produk

2. Tabel kategori

   Keterangan : Untuk mengelompokkan produk berdasarkan jenis tertentu.

   Primary Key : id_kategori

Tabel 4.23 Struktur tabel kategori

Field	Type	Size/Value	Keterangan
id_kategori	int	2	Menyimpan data id kategori produk
kode_kategori	varchar	50	Menyimpan data kode kategori produk
nama_kategori	varchar	100	Menyimpan data nama produk
deskripsi	text	-	Menyimpan data deskripsi kategori produk
parent	int	10	Menyimpan data parent kategori produk

status	enum	’0’,’1’	Menyimpan data status untuk kategori
url	varchar	250	Menyimpan data url suatu kategori produk

3. Tabel option

   Keterangan : Data situs yang muncul di halaman depan pada halaman publik

   Primary Key : id_option

Tabel 4.24 Struktur tabel option

Field	Type	Size/Value	Keterangan
id_option	int	2	Menyimpan data id option
nama_opsi	varchar	50	Menyimpan data variabel nama opsi
value_opsi	text	-	Menyimpan data nilai dari variabel opsi

4. Tabel order

   Keterangan : Untuk menyimpan data order

   Primary Key : id_order

Tabel 4.25 Struktur tabel order

Field	Type	Size/Value	Keterangan
id_order	int	4	Menyimpan data id order
user_id	int	10	Menyimpan data id user yang melakukan order
tanggal_masuk	timestamp	-	Menyimpan data tanggal masuk order
tanggal_kirim	timestamp	-	Menyimpan data tanggal pengiriman produk

status_order	enum	’0’,’1’,’2’,’3’	Menyimpan data status order
biaya_kirim	tinyint	10	Menyimpan data biaya kirim produk yang dipesan
total_item	int	10	Menyimpan data total keseluruhan item produk yang dipesan
total_biaya	int	20	Menyimpan data total biaya yang harus dibayar oleh user atas harga produk yang dipesan

5. Tabel order_data

   Keterangan : Untuk menyimpan daftar produk dan kuantitas produk yang dipesan.

   Primary Key : id_order_data

Tabel 4.26 Struktur tabel order_data

Field	Type	Size/Value	Keterangan
id_order_data	int	4	Menyimpan data id order data
order_id	int	10	menyimpan data id order
produk_id	tinyint	3	Menyimpan data id produk yang dipesan
kuantitas	int	50	Menyimpan data kuantitas item produk
subtotal	int	50	Menyimpan data subtotal biaya

6. Tabel produk

   Keterangan : Untuk menyimpan data produk

   Primary Key : id_produk

Tabel 4.27 Struktur tabel produk

Field	Type	Size/Value	Keterangan
id_produk	int	3	Menyimpan data id produk
kategori_id	tinyint	10	Menyimpan data id kategori suatu produk
kode_produk	varchar	50	Menyimpan data kode produk
nama_produk	varchar	100	Menyimpan data nama produk
url_produk	varchar	250	Menyimpan data url di mana halaman produk bisa diakses
harga_jual	int	10	Menyimpan data harga jual suatu produk
harga_baru	int	10	Menyimpan data harga baru suatu produk
status_produk	enum	'1','0'	Menyimpan data status produk
deskripsi_produk	text	-	Menyimpan data deskripsi suatu produk
stok	int	10	Menyimpan data stok suatu produk

7. Tabel user

   Keterangan : Untuk menyimpan akun *user*

   Primary Key : id_user

Tabel 4.28 Struktur tabel user

Field	Type	Size/Value	Keterangan
id_user	int	4	Menyimpan data id user
username	varchar	50	Menyimpan data nama pengguna untuk melakukan login

email	varchar	50	Menyimpan data email pengguna
password	varchar	50	Menyimpan data password untuk melakukan login
status	enum	’1’,’0’	Menyimpan data status user
level	enum	’admin’,’user’	Menyimpan data level user
last_visit	varchar	50	Menyimpan data waktu terakhir melakukan login

8. Tabel user_data

   Keterangan : Untuk menyimpan data profil pengguna

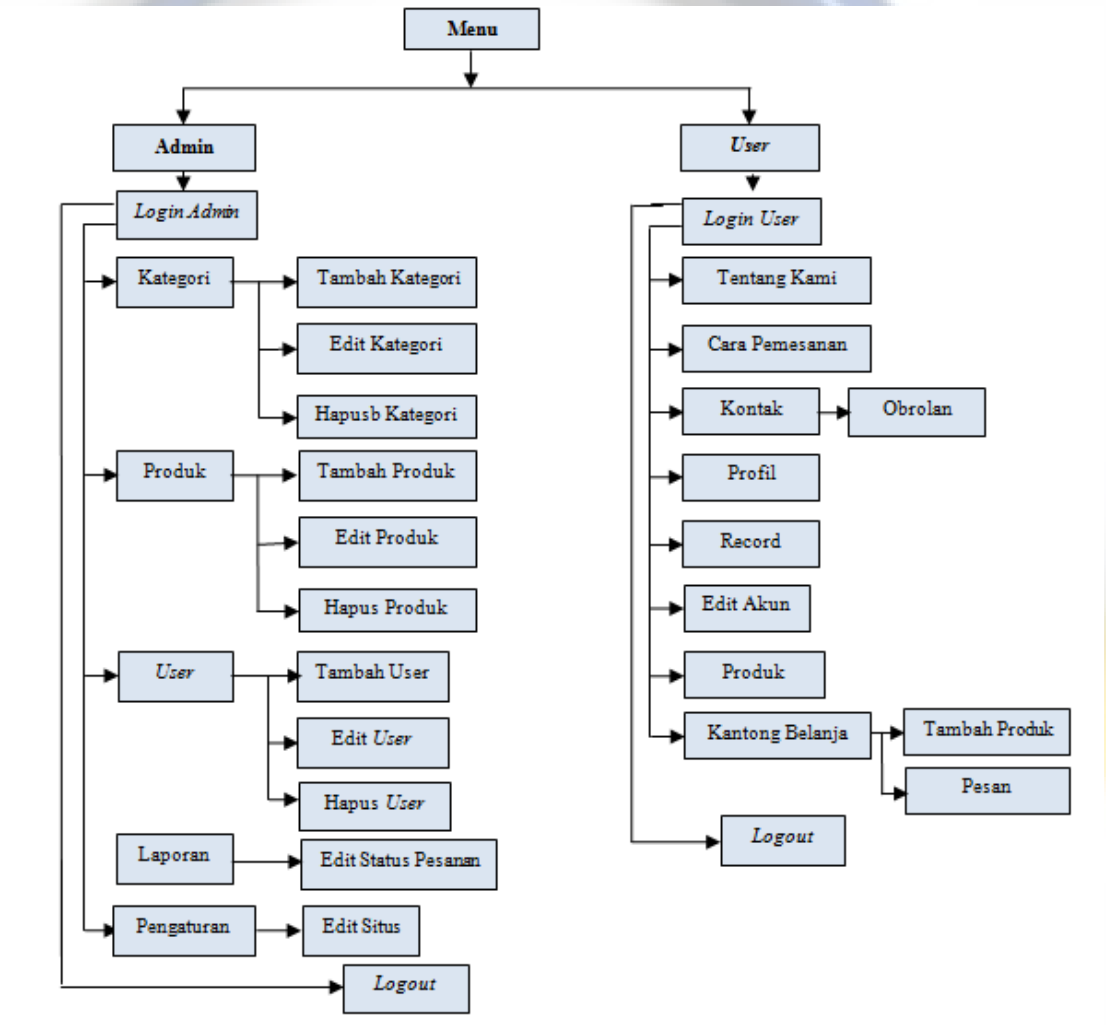
   Primary Key : id_user_data

Tabel 4.29 Struktur tabel user_data

Field	Type	Size/Value	Keterangan
id_user_data	int	4	Menyimpan data id user data
user_id	tinyint	10	Menyimpan data id user
nama_depan	varchar	50	Menyimpan data nama depan user
nama_belakang	varchar	50	Menyimpan data nama belakang user
alamat	text	-	Menyimpan data alamat rumah atau alamat di mana produk yang di pesan dikirim
kode_pos	int	10	Menyimpan data kode pos sesuai dengan alamat user
propinsi_id	int	10	Menyimpan data id propinsi user
phone	varchar	50	Menyimpan data nomor telepon atau HP user

4.4 Perancangan Menu

Berikut di bawah ini merupakan gambar perancangan menu pada Aplikasi Penjualan *Online*:



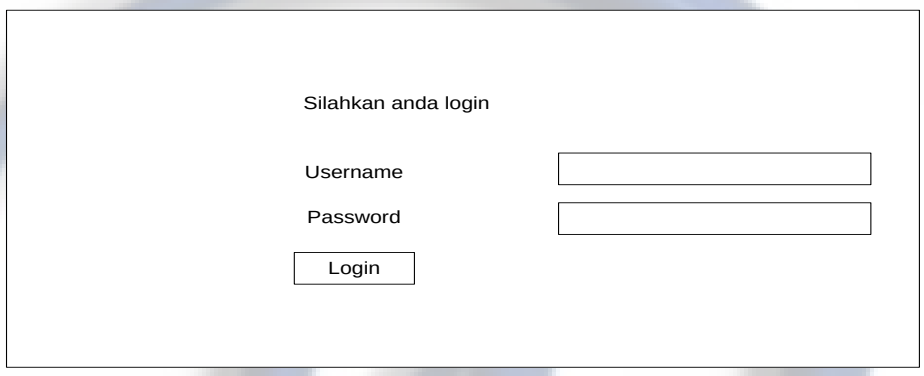
Gambar 4.46 Perancangan Menu

4.5 *Layout* Antarmuka

Layout antarmuka merupakan rancangan antarmuka yang akan digunakan sebagai perantara *user* dengan perangkat lunak yang dikembangkan. *Layout* antarmuka dari aplikasi penjualan *online* ini adalah sebagai berikut:

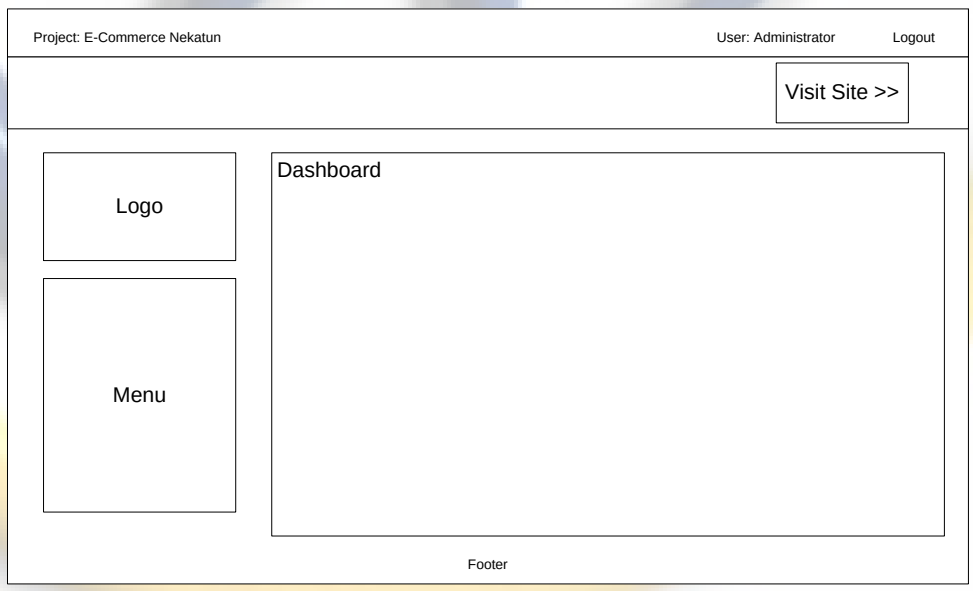
4.5.1 *Layout* Antarmuka *Back Office*

1. Antarmuka Form *Login* Admin



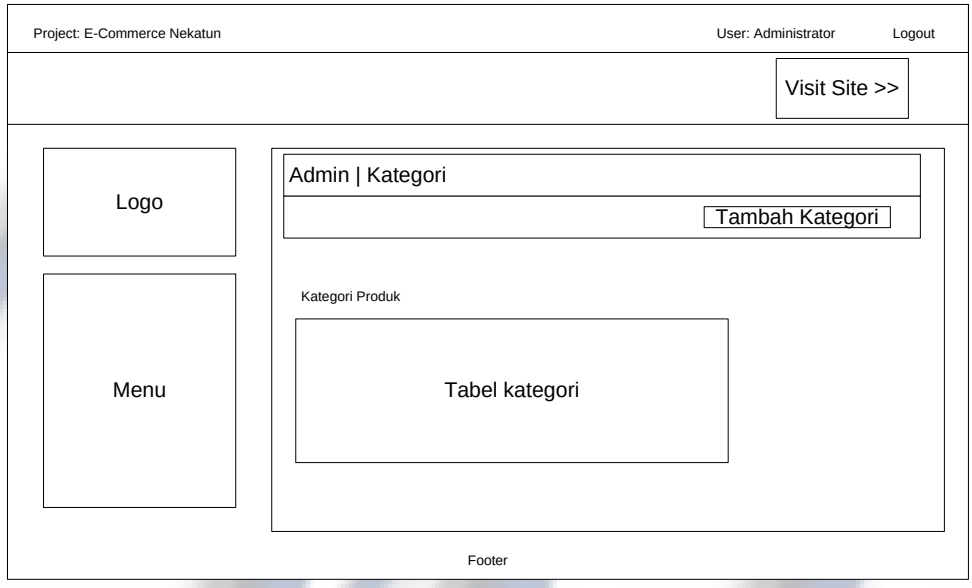
Gambar 4.47 Antarmuka Form *Login* Admin

2. Antarmuka Halaman Dashboard



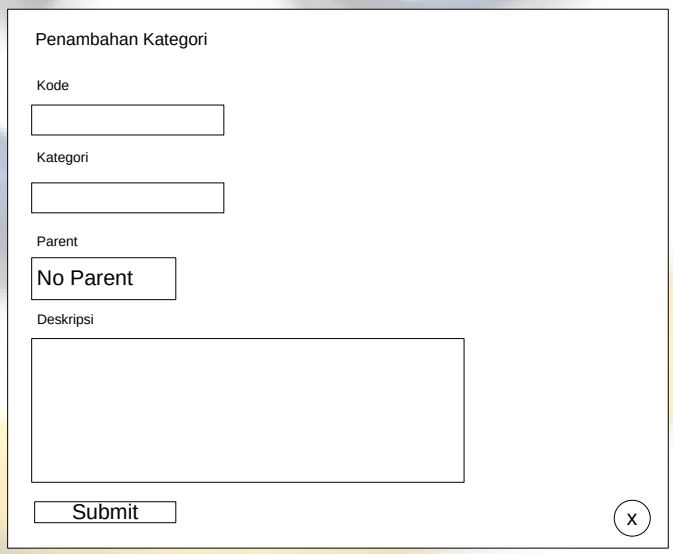
Gambar 4.48 Antarmuka Halaman Dashboard

3. Antarmuka Admin Kategori



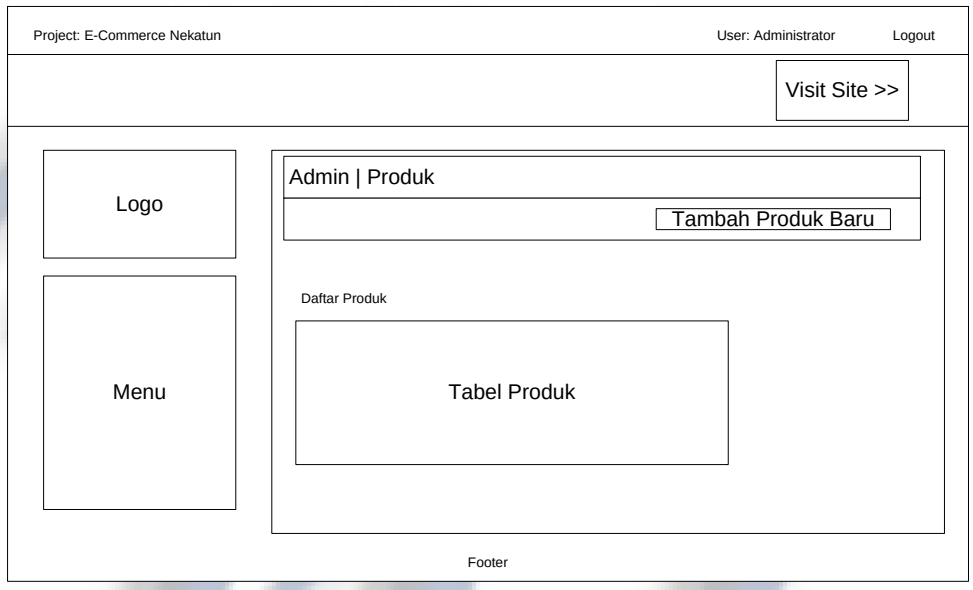
Gambar 4.49 Antarmuka Admin Kategori

4. Antarmuka Tambah Kategori



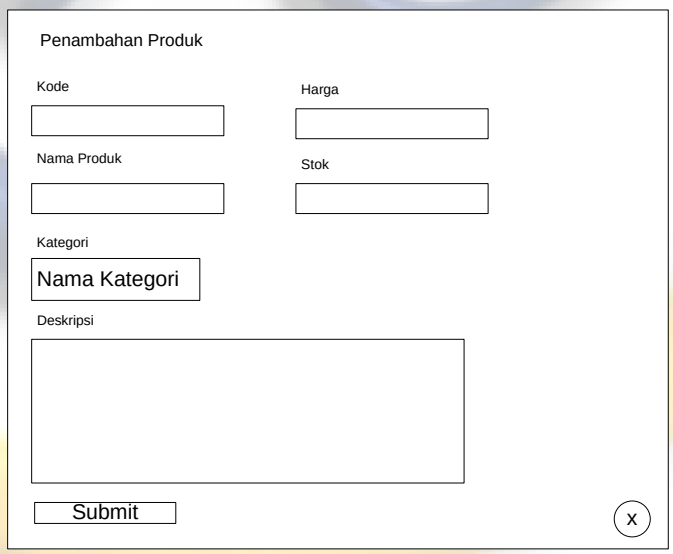
Gambar 4.50 Antarmuka Tambah Kategori

5. Antarmuka Admin Produk



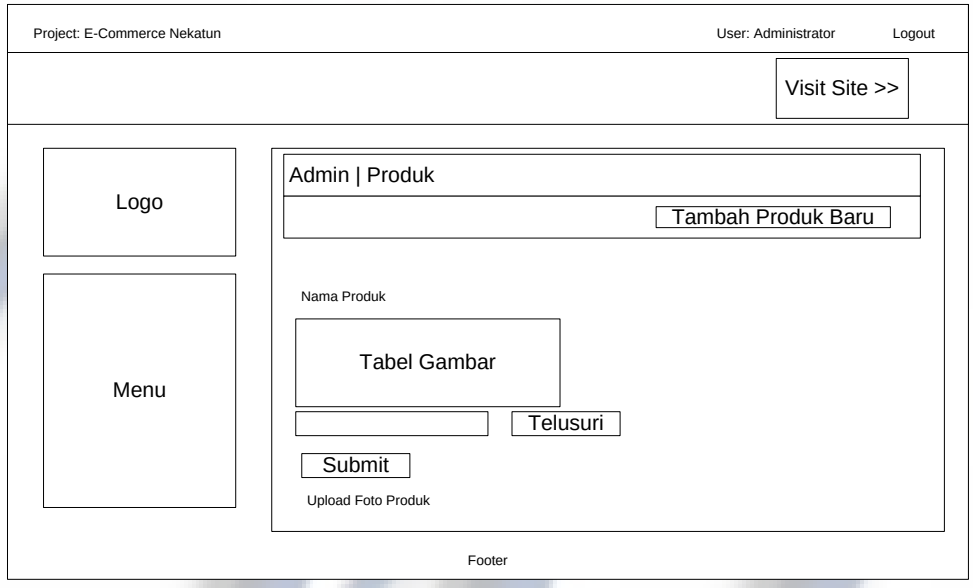
Gambar 4.51 Antarmuka Admin Produk

6. Antarmuka Tambah Produk Baru



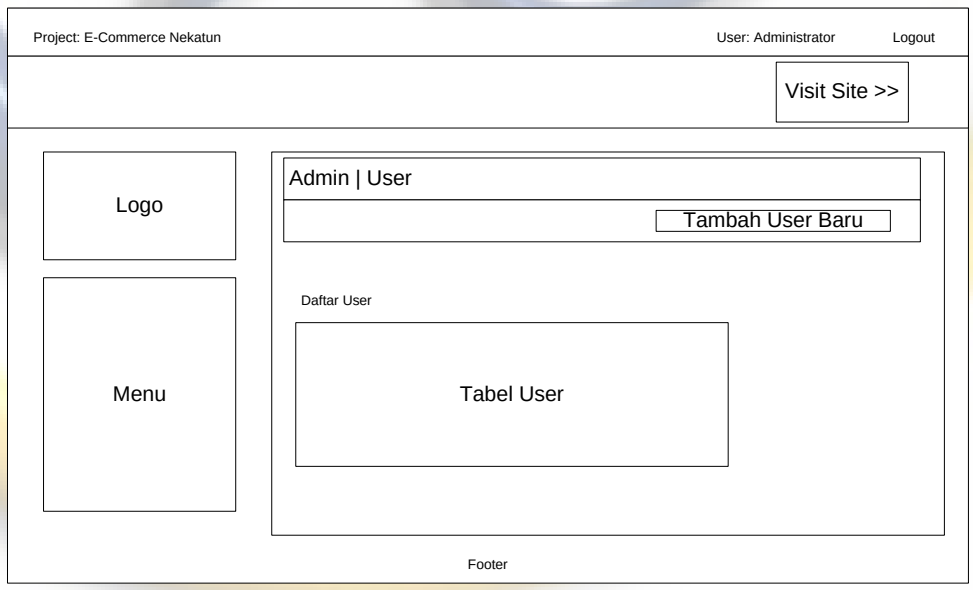
Gambar 4.52 Antarmuka Tambah Produk Baru

7. Antarmuka Tambah Gambar Produk



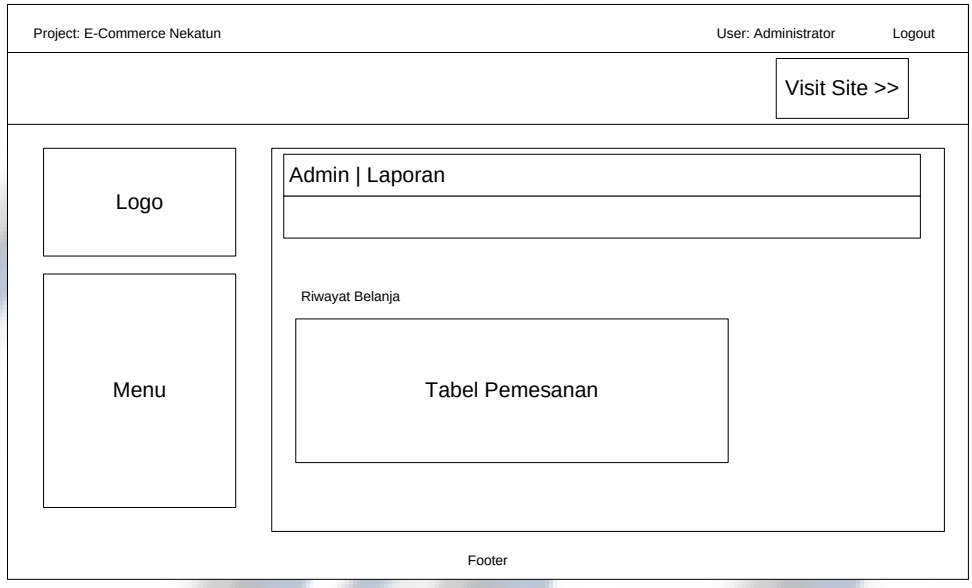
Gambar 4.53 Antarmuka Tambah Gambar Produk

8. Antarmuka *User*



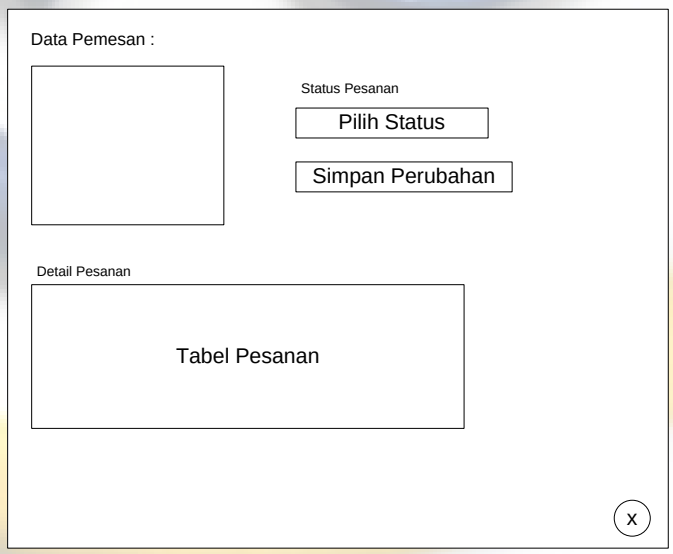
Gambar 4.54 Antarmuka *User*

9. Antarmuka Laporan



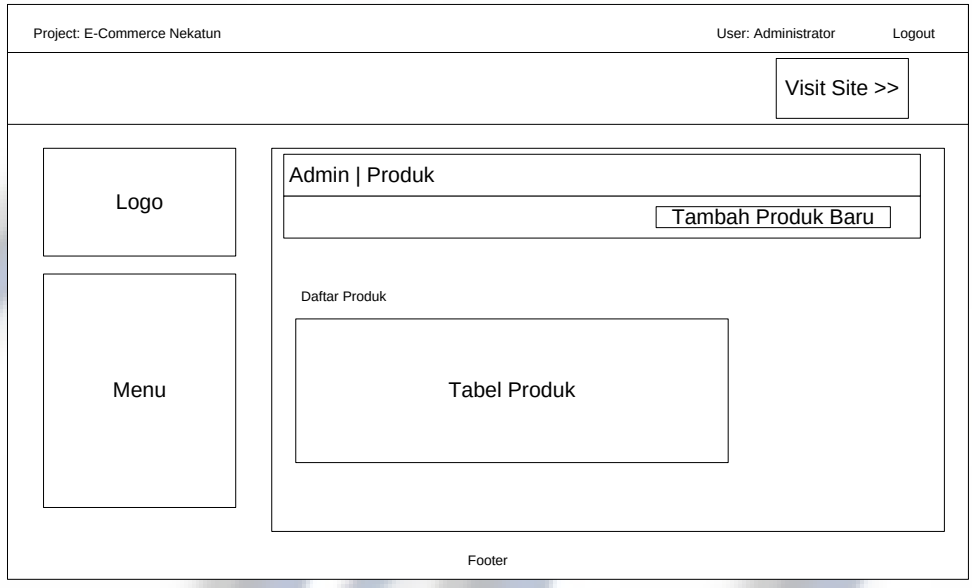
Gambar 4.55 Antarmuka Laporan

10. Antarmuka Ubah Status Pemesanan



Gambar 4.56 Antarmuka Ubah Status Pemesanan

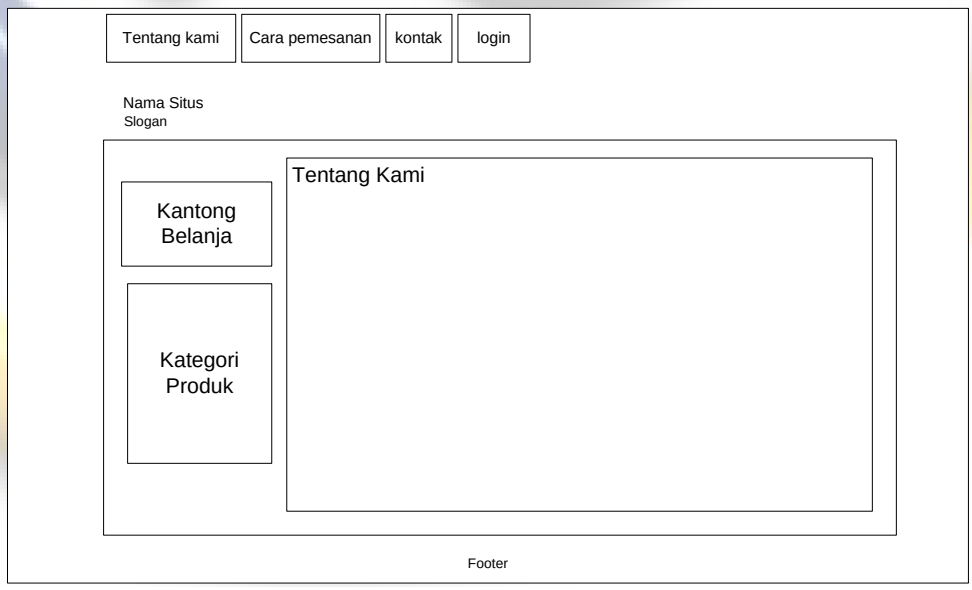
11. Antarmuka Pengaturan Situs



Gambar 4.57 Antarmuka Pengaturan Situs

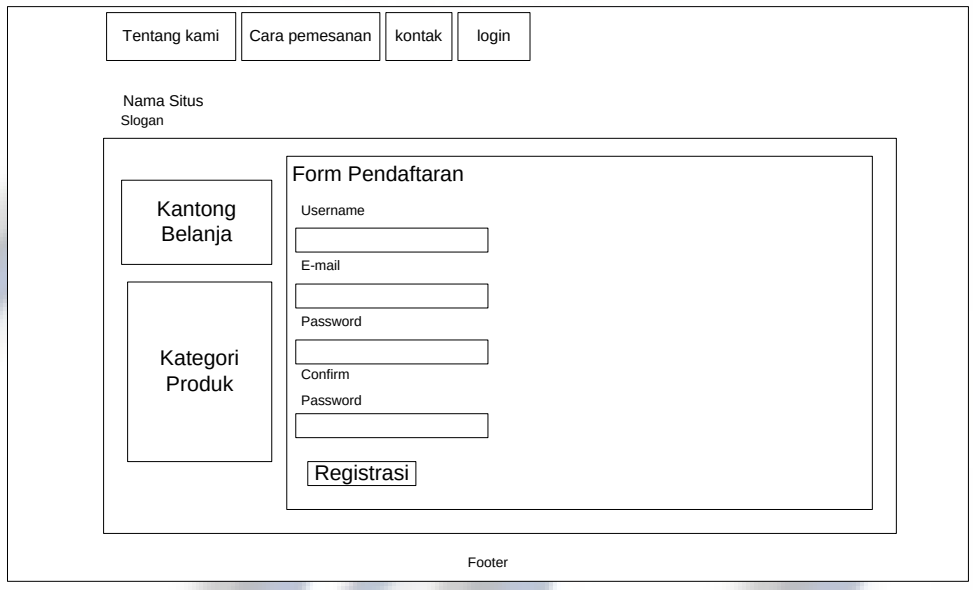
4.5.2 *Layout* Antarmuka *Front Office*

1. Antarmuka *Homepage*



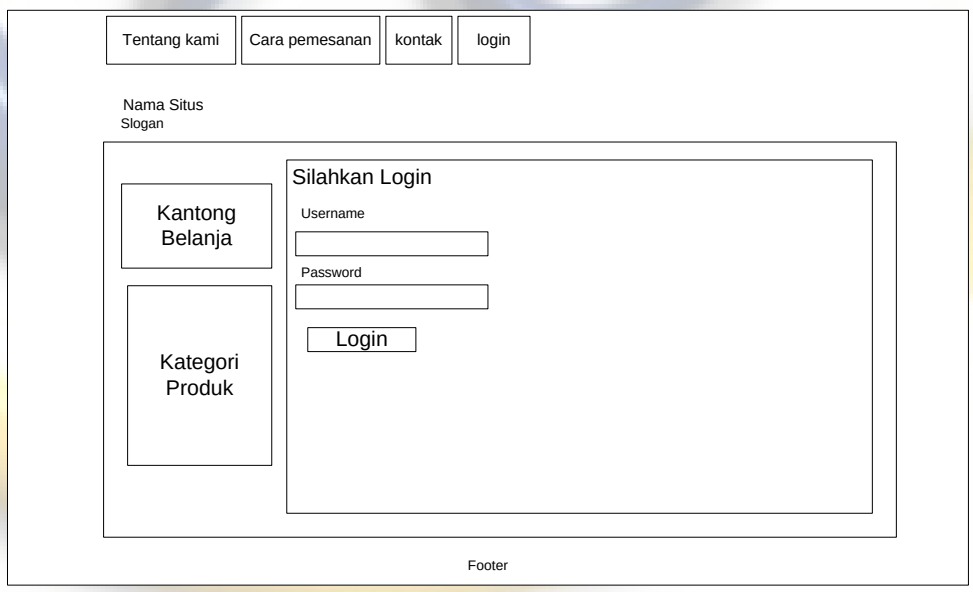
Gambar 4.58 Antarmuka *Homepage*

2. Antarmuka Registrasi *User*



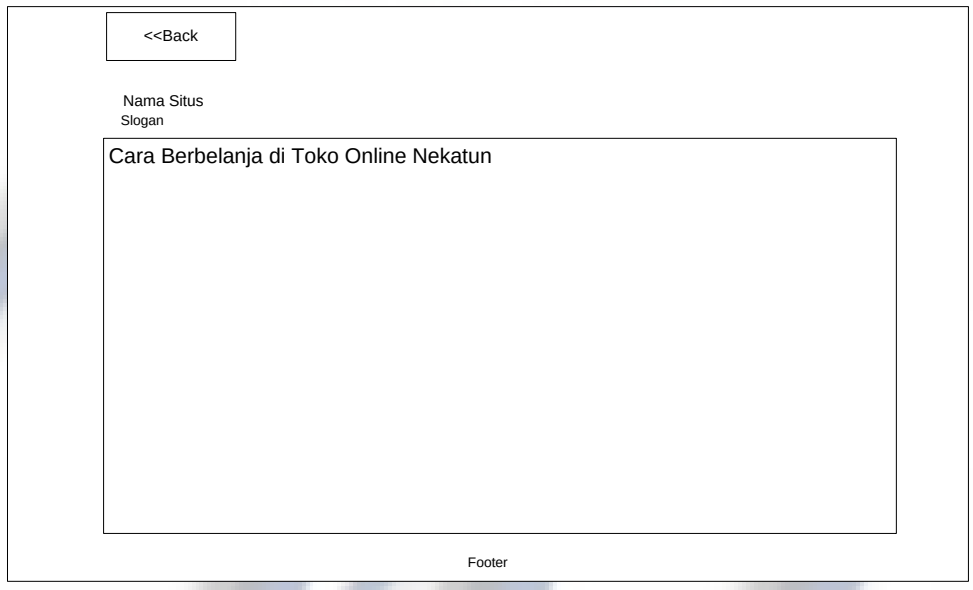
Gambar 4.59 Antarmuka Registrasi *User*

3. Antarmuka *Login User*



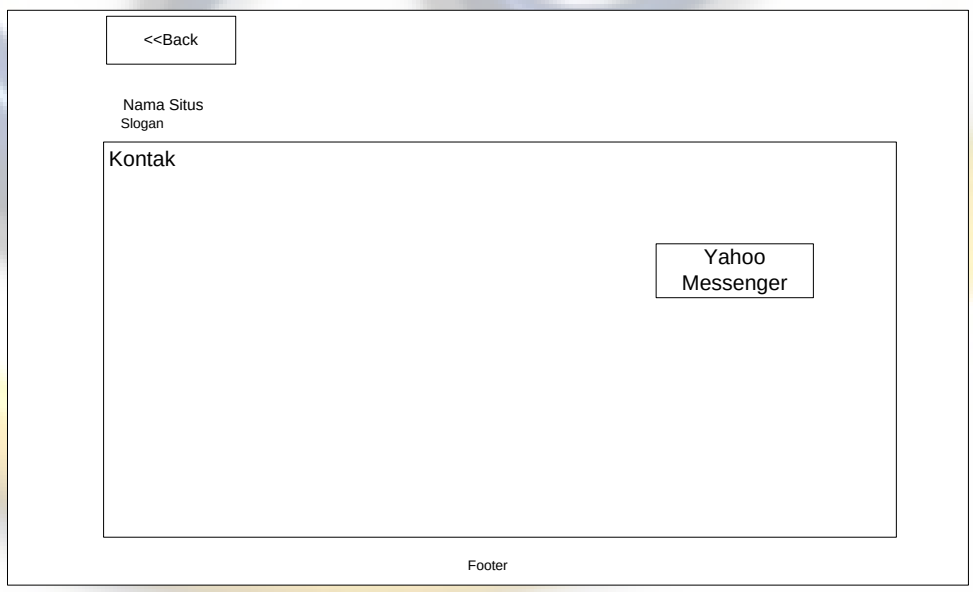
Gambar 4.60 Antarmuka *Login User*

4. Antarmuka Cara Pemesanan



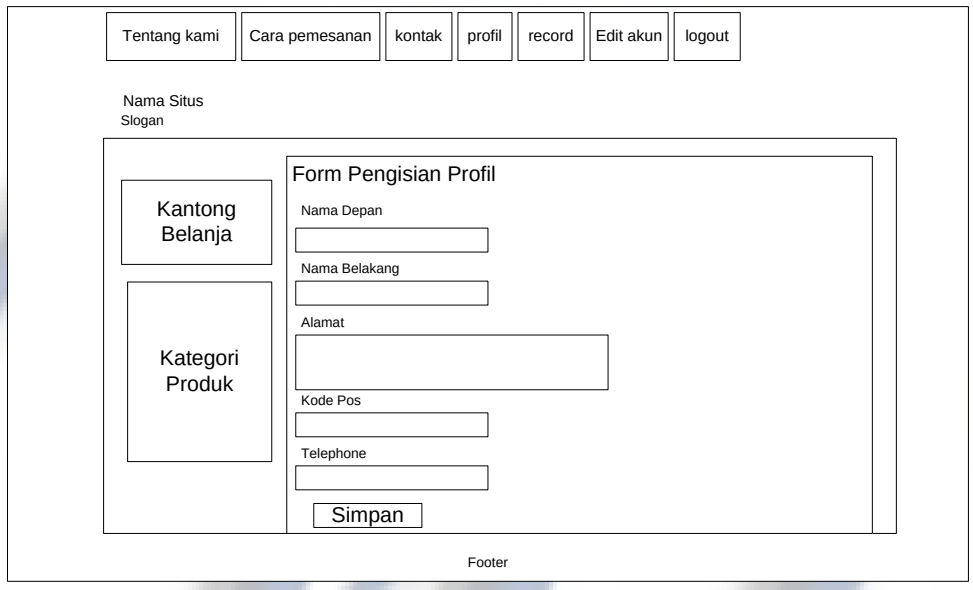
Gambar 4.61 Antarmuka Cara Pemesanan

5. Antarmuka Kontak



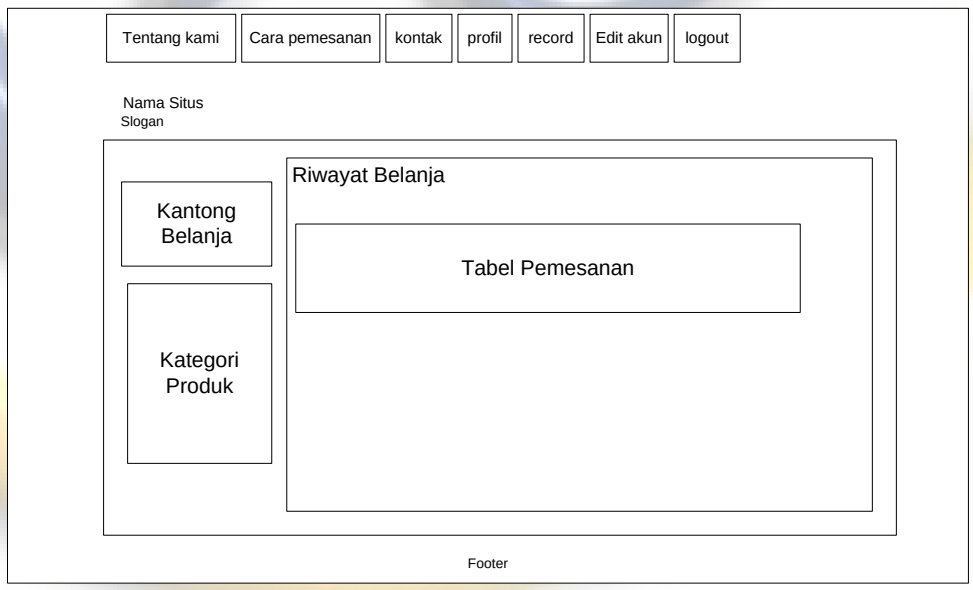
Gambar 4.62 Antarmuka Kontak

6. Antarmuka Profil *User*



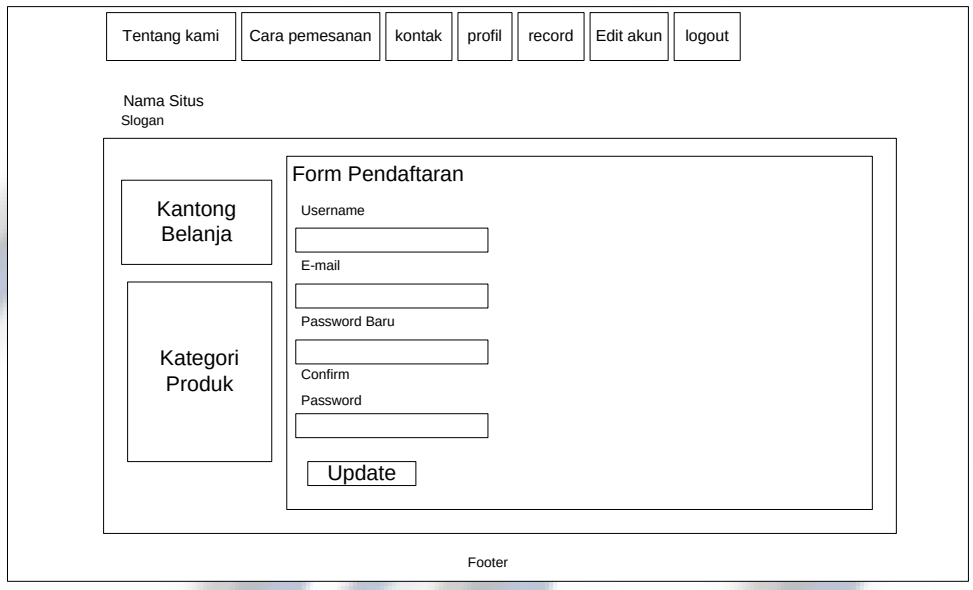
Gambar 4.63 Antarmuka Profil *User*

7. Antarmuka *Record*



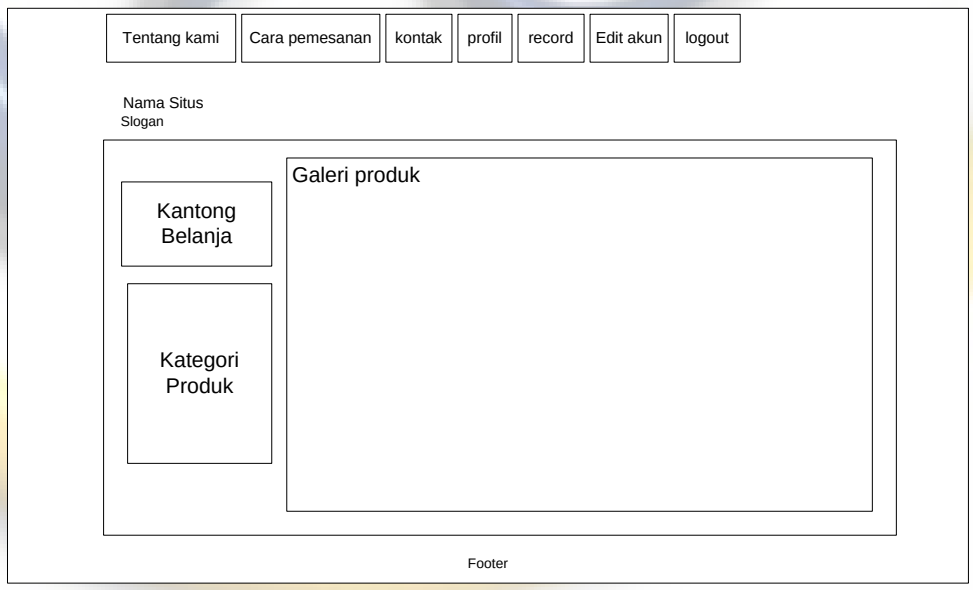
Gambar 4.64 Antarmuka *Record*

8. Antarmuka Edir Akun



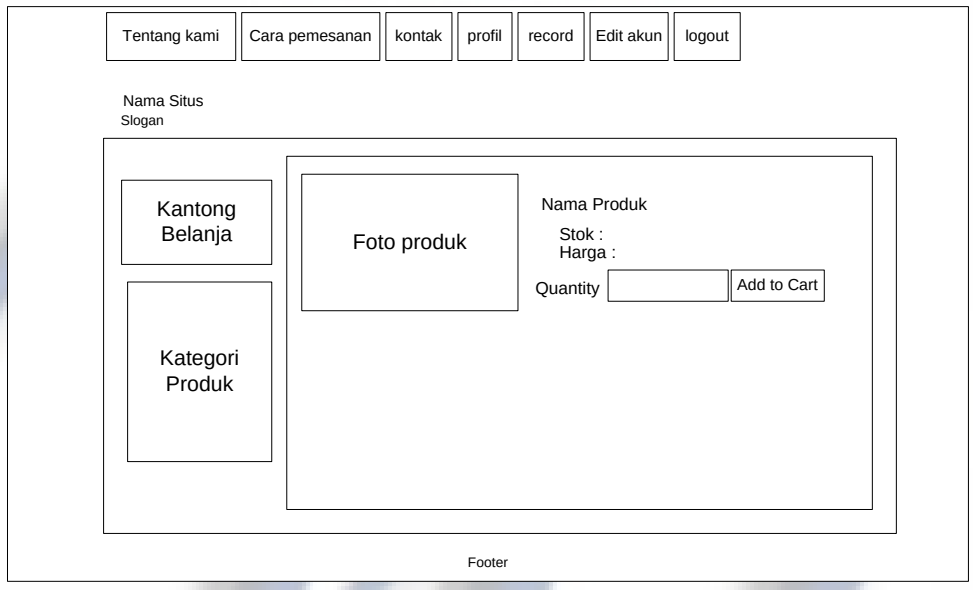
Gambar 4.65 Antarmuka Edit Akun

9. Antarmuka Produk



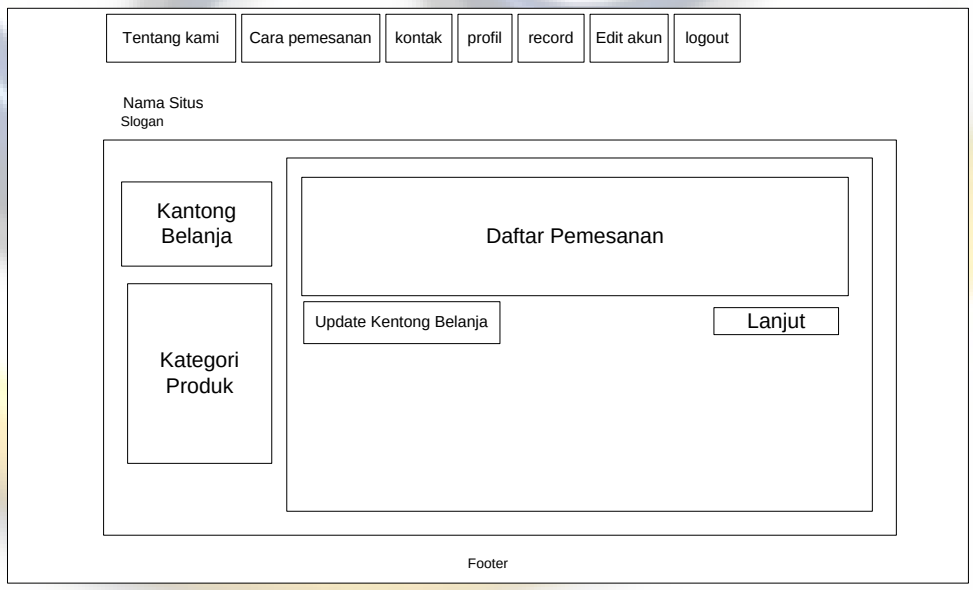
Gambar 4.66 Antarmuka Produk

10. Antarmuka Detail Produk



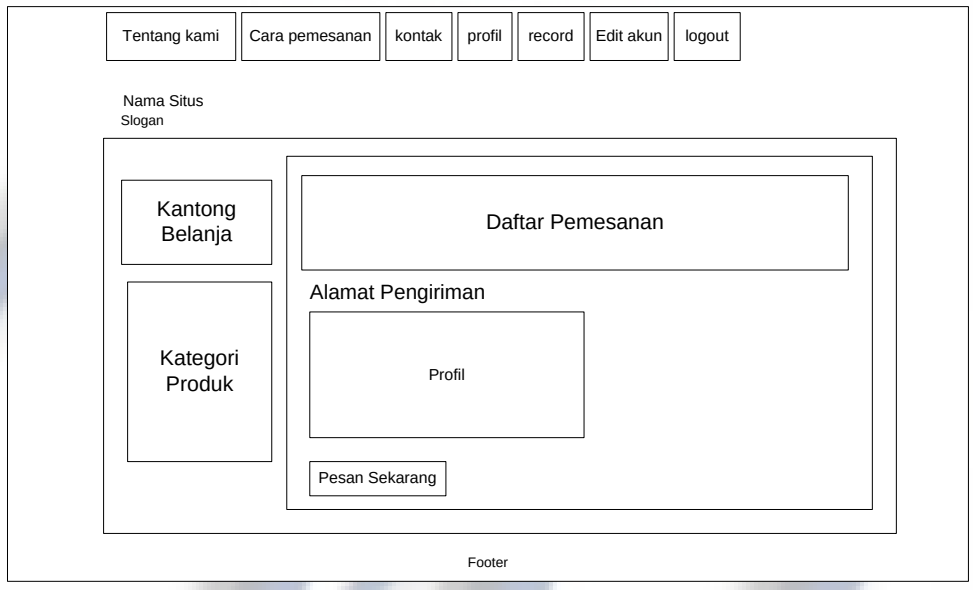
Gambar 4.67 Antarmuka Detail Produk

11. Antarmuka Kantong Belanja



Gambar 4.68 Antarmuka Kantong Belanja

12. Antarmuka Transaksi Pemesanan



Gambar 4.69 Antarmuka Transaksi Pemesanan

BAB V
IMPLEMENTASI SISTEM

Pada bab ini akan dijelaskan tentang implementasi hasil desain pada bab empat dan penyesuaian kebutuhan sistem agar sistem berjalan dengan optimal.

5.1 Lingkungan Implementasi

Lingkungan implementasi meliputi setiap kebutuhan sumber daya yang dibutuhkan Aplikasi Penjualan *Online* dengan Menggunakan CodeIgniter saat pengembangan maupun saat implementasi, berupa perangkat keras (*Hardware*), perangkat lunak (*Software*) dan Pengguna (*Brainware*).

5.1.1 Perangkat Kebutuhan Sistem

Perangkat minimum yang dibutuhkan saat pengembangan dan implementasi Aplikasi Penjualan *Online* dengan Menggunakan CodeIgniter ini adalah sebagai berikut:

Tabel 5.1 perangkat Kebutuhan Sistem

Pengembangan	**Implementasi**	
Hardware Develop	*Hardware* Admin	*Hardware* User
a. *Processor* Intel Pentium 4 b. RAM 512 MB, c. 1 GB ruang *harddisk*, d. 1280x800 display dengan 16-bit video card, e. DVD-ROM drive, f. Jaringan internet.	a. *Processor* Intel Pentium III 500 Mhz, b. RAM 64 MB, c. 52 MB ruang *Harddisk*, d. Jaringan internet.	a. *Processor* Intel Pentium III 500 Mhz, b. RAM 64 MB, c. 52 MB ruang *Harddisk*, d. Jaringan internet.

Software Develop	*Software* Admin	*Software* User
a. Windows XP, b. Mozilla Firefox, c. Adobe Dreamweaver CS5, d. XAMPP, e. Adobe Photoshop CS5 f. Yahoo Messenger	a. Windows XP, b. Mozilla Firefox, c. Yahoo Messenger	a. Windows XP, b. Mozilla Firefox, c. Yahoo Messenger

5.1.2 Karakteristik Pengguna

Berikut karakteristik pengguna yang diperlukan untuk menjalankan Aplikasi Penjualan *Online*:

Tabel 5.2 Karakteristik Pengguna

Pengguna	Hak Akses	Tingkat Keterampilan
Admin	Melakukan pengolahan data master	Mampu mengoperasikan komputer, pemahaman yang cukup untuk mengelola data dalam *website* dan memiliki pengetahuan tentang internet.
User	Melakukan registrasi, mengubah profil dan akun, memesan melalui *online* serta melihat *record* pemesanan.	Mampu mengoperasikan dan melakukan *browsing* di internet.

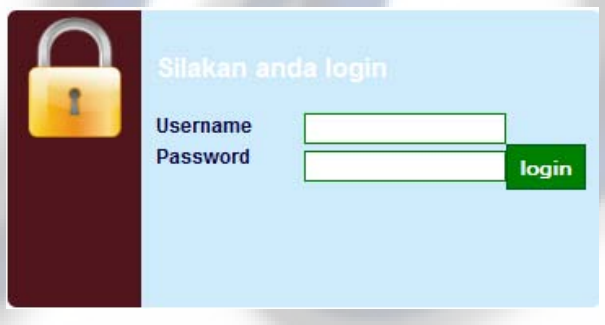
5.2 Implementasi Antarmuka

Berikut ini adalah implementasi antarmuka yang telah dikembangkan dari bab sebelumnya:

5.2.1 Implementasi Antarmuka *Back Office*

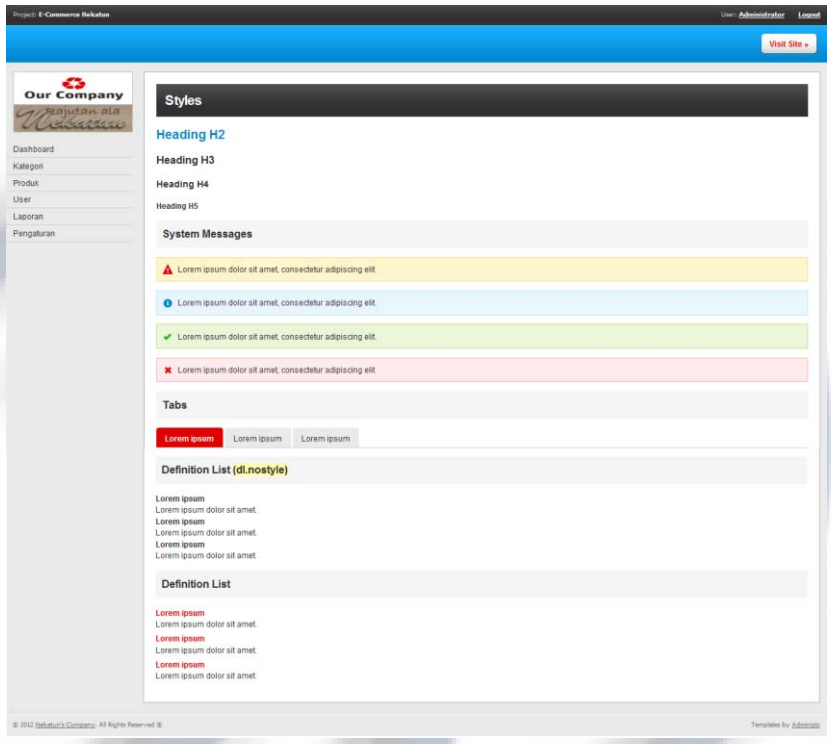
Pada halaman *back office* ini dapat melakukan penambahan kategori, edit kategori, hapus kategori, penambahan produk, penambahan gambar produk, hapus gambar, edit produk, hapus produk, penambahan *user*, edit *user*, hapus *user*, edit laporan status pesanan dan edit pengaturan situs. Berikut di bawah ini adalah gambar mengenai halaman *back office*:

1. Antarmuka *Login* Admin



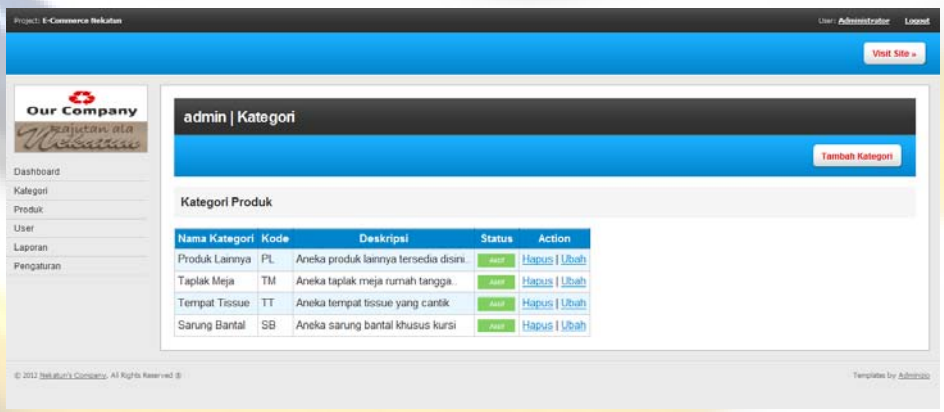
Gambar 5.1 Antarmuka *Login* Admin

2. Antarmuka Halaman Dashboard



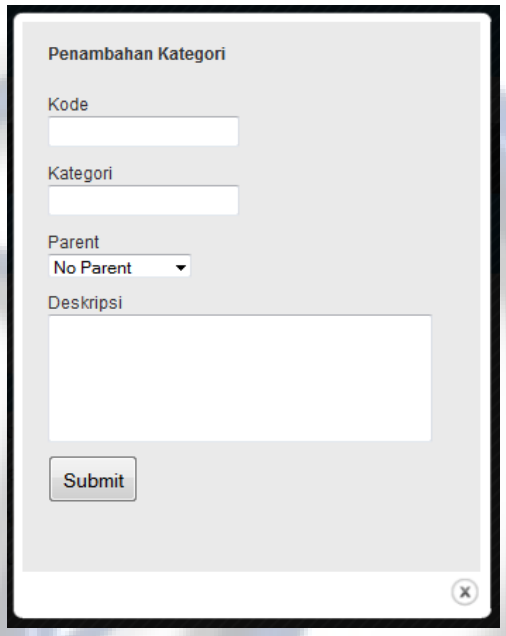
Gambar 5.2 Antarmuka Halaman Dashboard

3. Antarmuka Admin Kategori



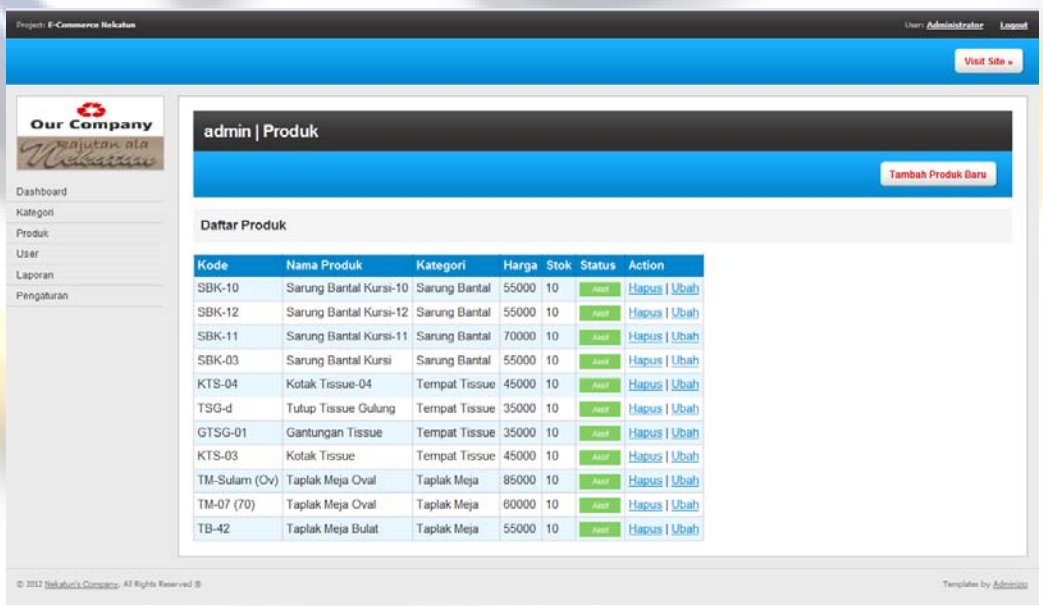
Gambar 5.3 Antarmuka Admin Kategori

4. Antarmuka Tambah Kategori



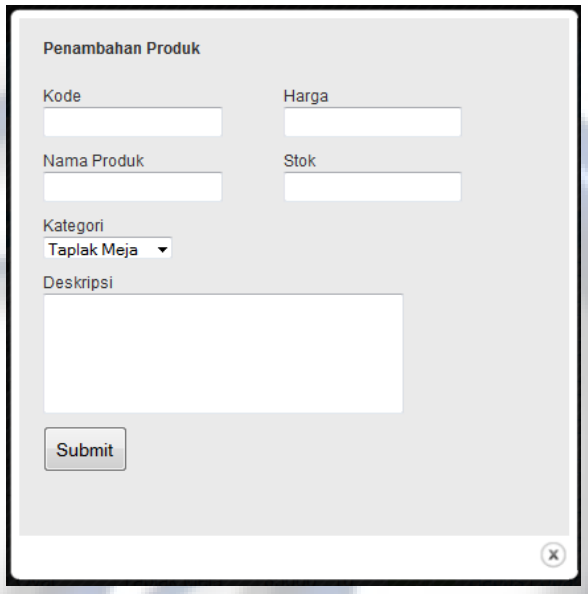
Gambar 5.4 Antarmuka Tambah Kategori

5. Antarmuka Admin Produk



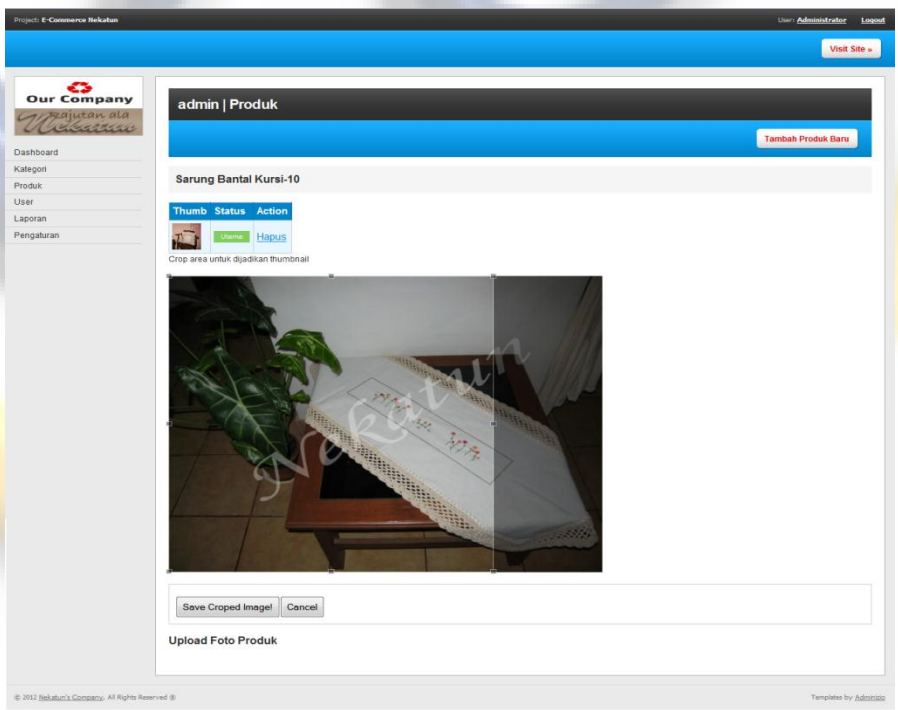
Gambar 5.5 Antarmuka Admin Produk

6. Antarmuka Tambah Produk



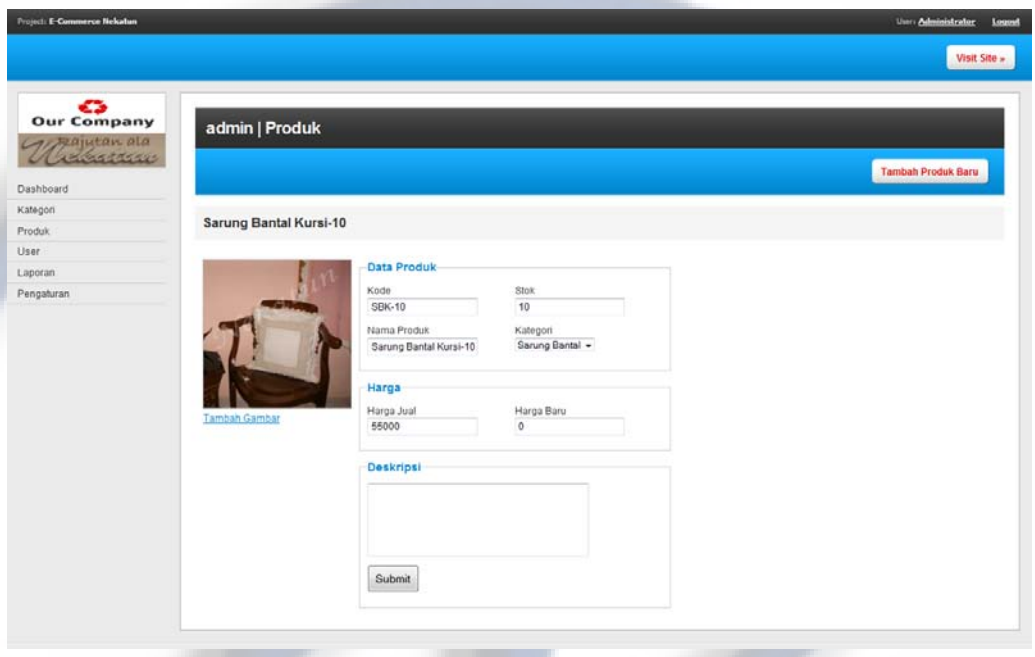
Gambar 5.6 Antarmuka Tambah Produk

7. Antarmuka Tambah Gambar Produk



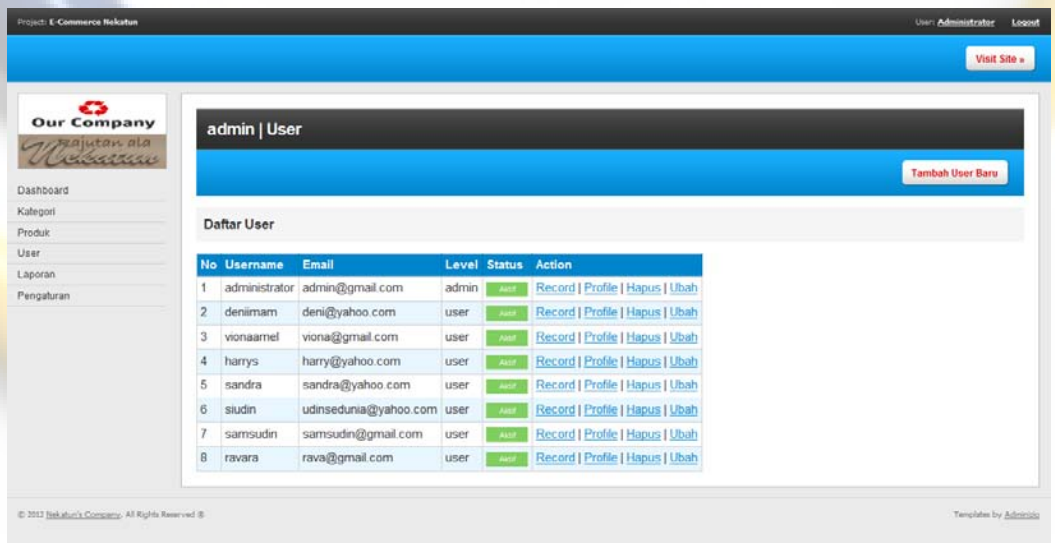
Gambar 5.7 Antarmuka Tambah Gambar Produk

8. Antarmuka Mengubah Produk



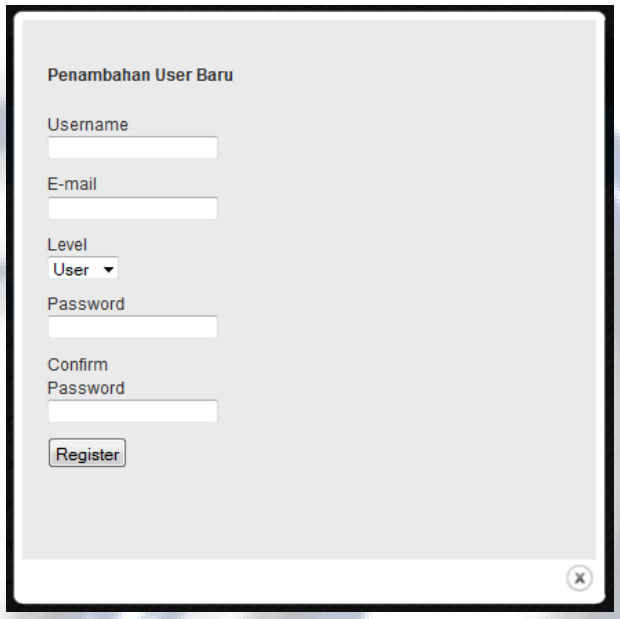
Gambar 5.8 Antarmuka Mengubah Produk

9. Antarmuka *User*



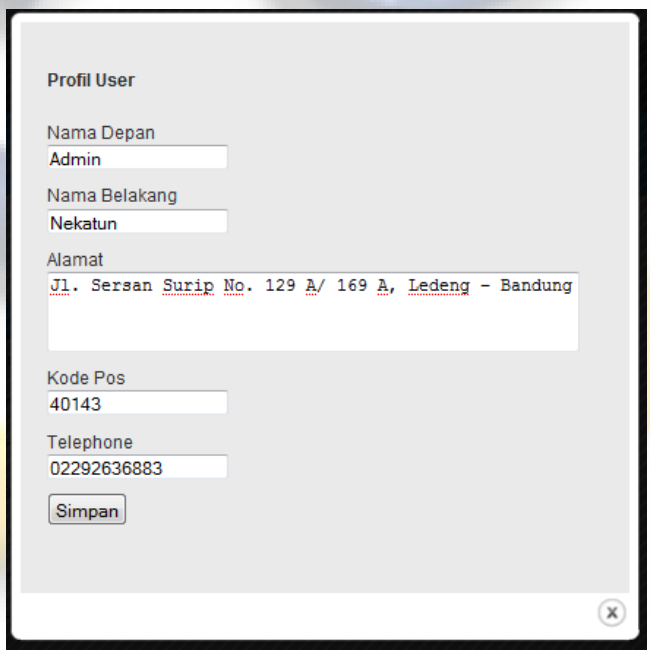
Gambar 5.9 Antarmuka *User*

10. Antarmuka Tambah *User*



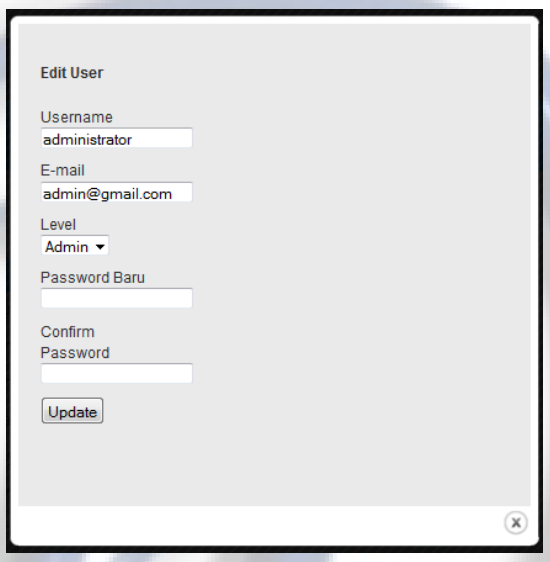
Gambar 5.10 Antarmuka Tambah *User*

11. Antarmuka Admin Mengubah Profil *User*



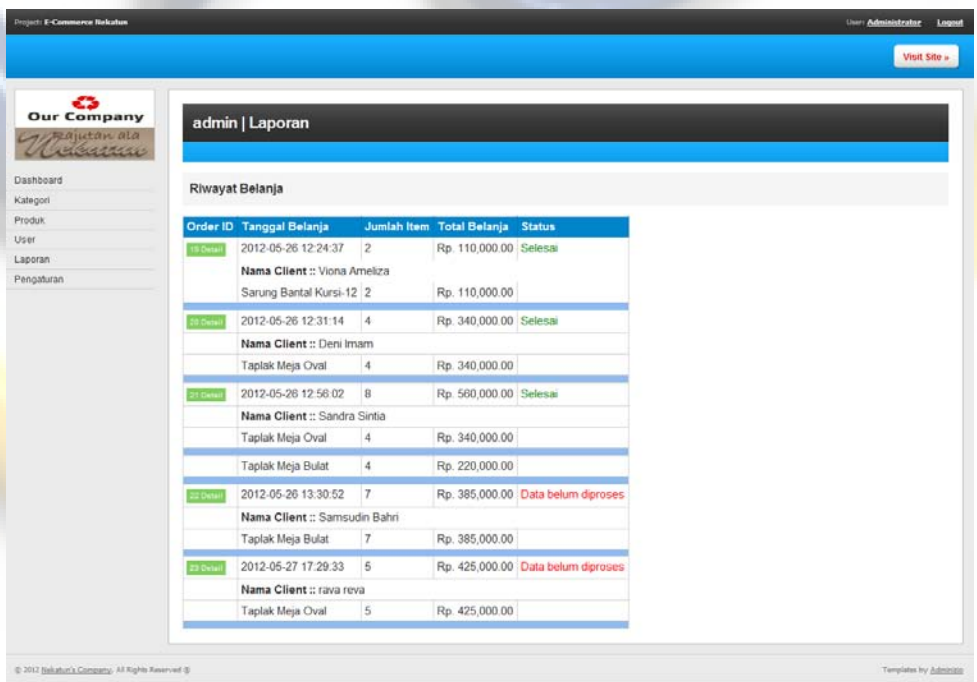
Gambar 5.11 Antarmuka Admin Mengubah Profil *User*

12. Antarmuka Admin Mengubah Akun *User*



Gambar 5.12 Antarmuka Admin Mengubah Akun *User*

13. Antarmuka Laporan

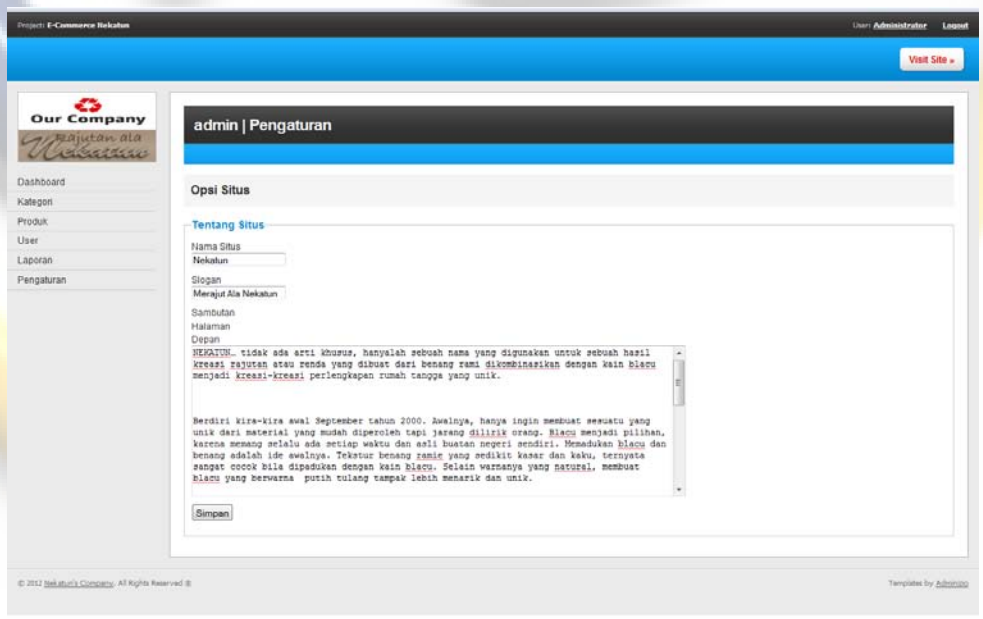


Gambar 5.13 Antarmuka Laporan

14. Antarmuka Mengubah Status Pesanan

Gambar 5.14 Antarmuka Mengubah Status Pesanan

15. Antarmuka Pengaturan Situs



Gambar 5.15 Antarmuka Pengaturan Situs

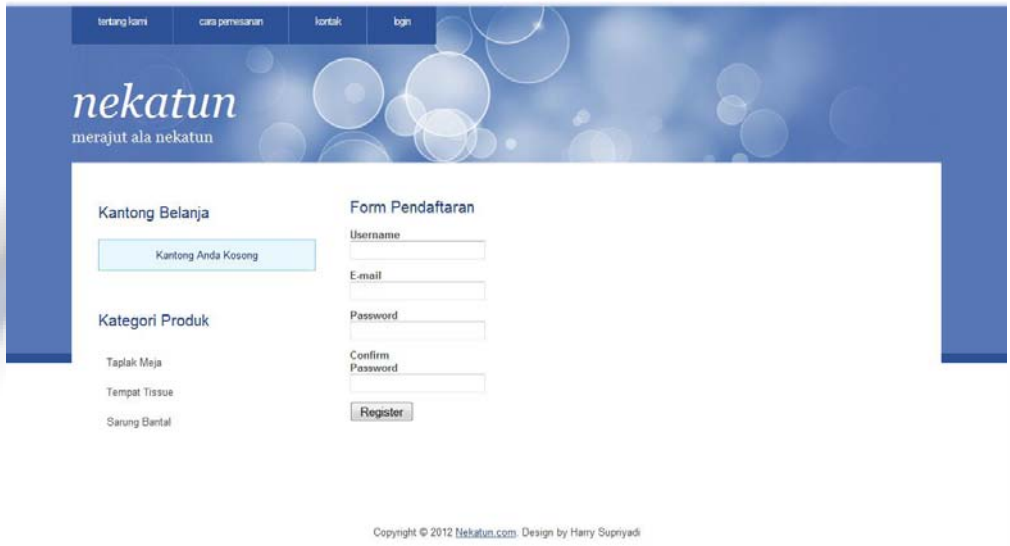
5.2.2 Implementasi Antarmuka *Front Office*

Pada halaman *front office* ini di mana *user* dapat melakukan akses halaman tentang kami, cara pemesanan, kontak, registrasi dan *login*. Setelah melakukan *login*, *user* dapat melakukan akses halaman profil, *record* belanja, edit akun, produk dan melakukan pemesanan. Berikut di bawah ini adalah gambar mengenai halaman *front office*:

1. Antarmuka *Homepage*

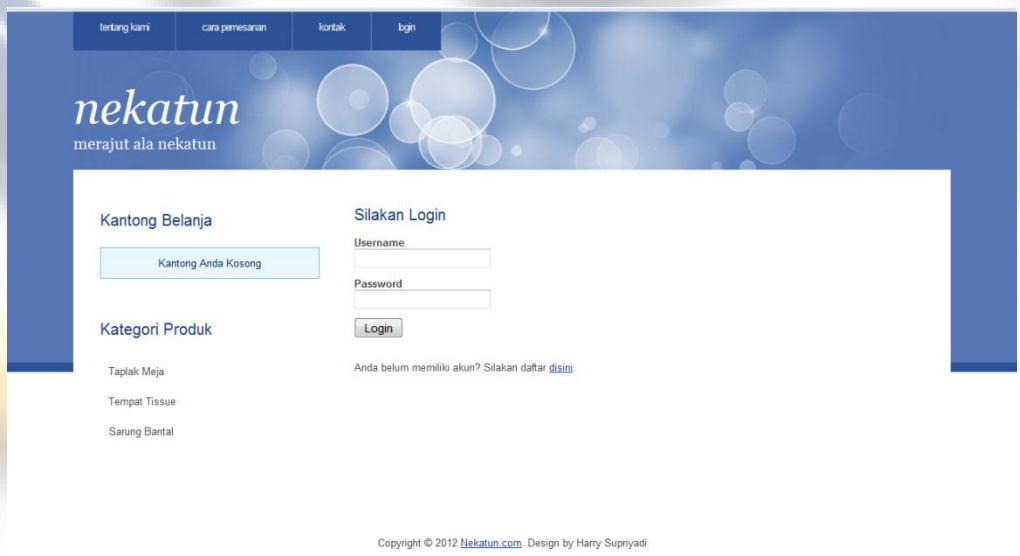
Gambar 5.16 Antarmuka *Homepage*

2. Antarmuka Registrasi *User*



Gambar 5.17 Antarmuka Registrasi *User*

3. Antarmuka *Login User*

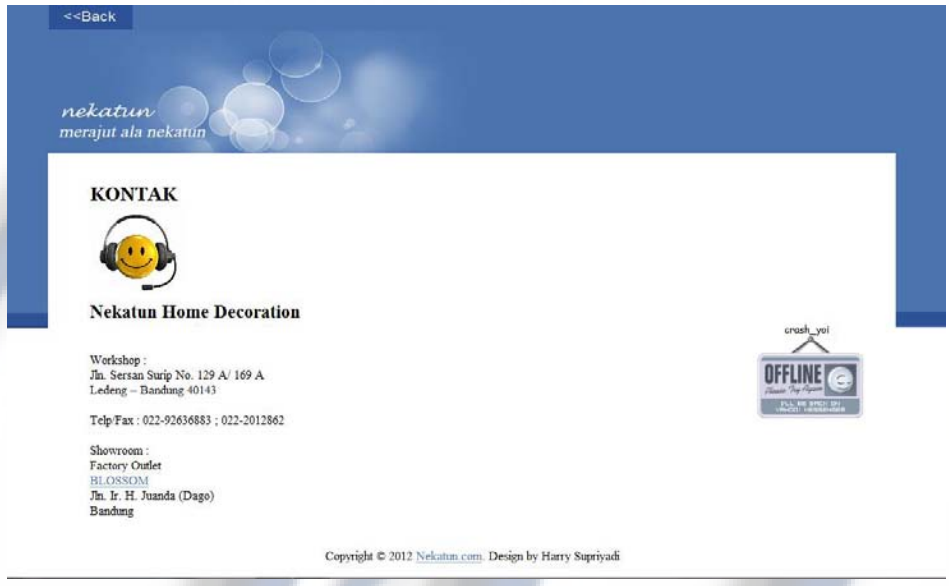


Gambar 5.18 Antarmuka *Login User*

4. Antarmuka Cara Pemesanan

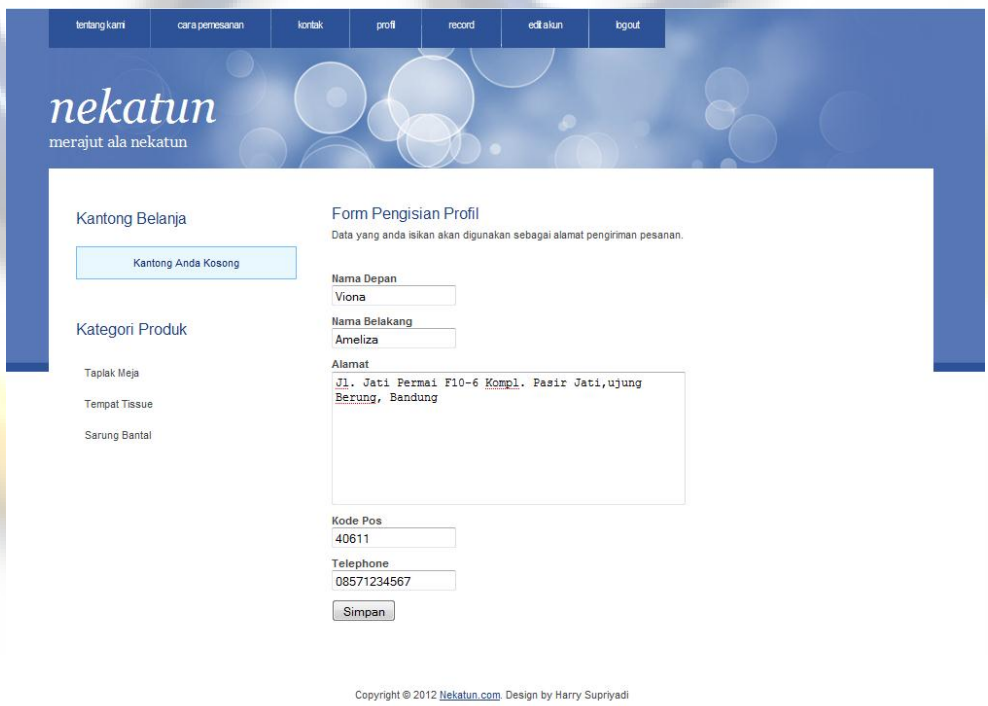
Gambar 5.19 Antarmuka Cara Pemesanan

5. Antarmuka Kontak



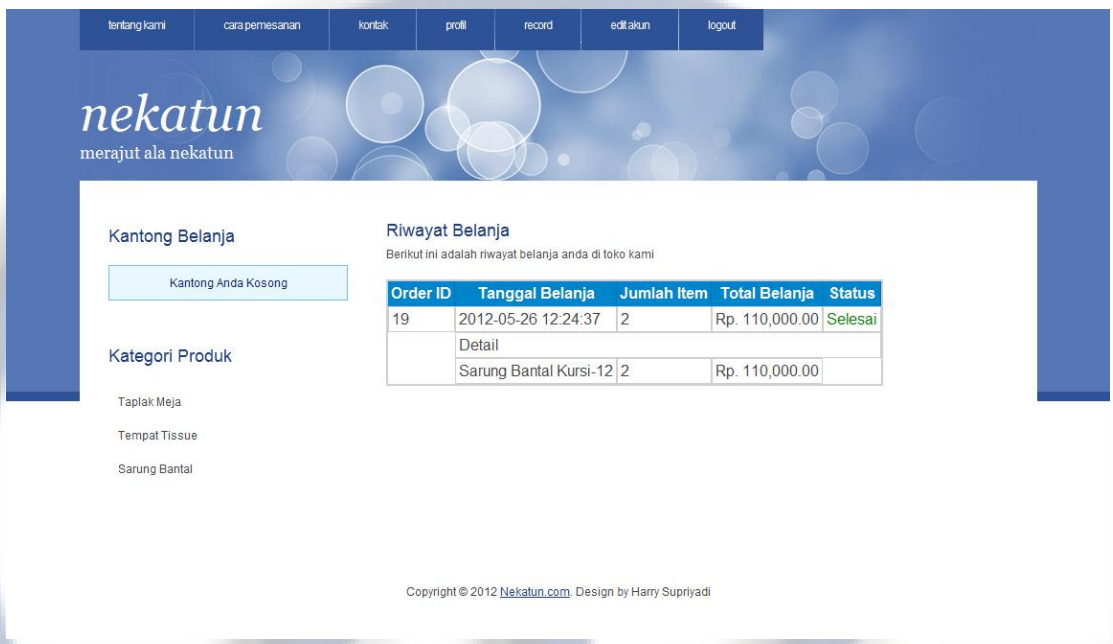
Gambar 5.20 Antarmuka Kontak

6. Antarmuka Profil *User*



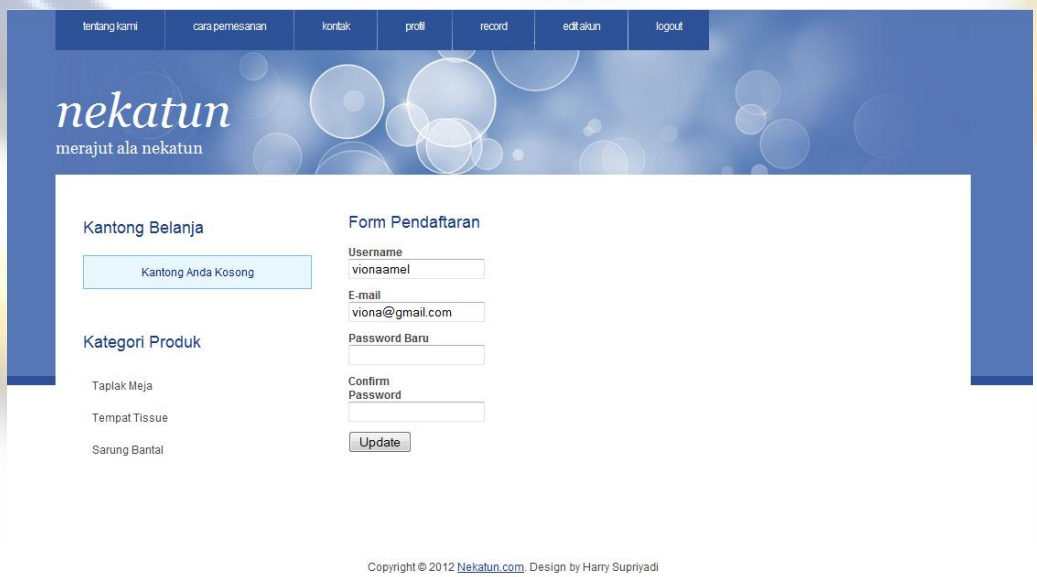
Gambar 5.21 Antarmuka Profil *User*

7. Antarmuka Rekaman Belanja



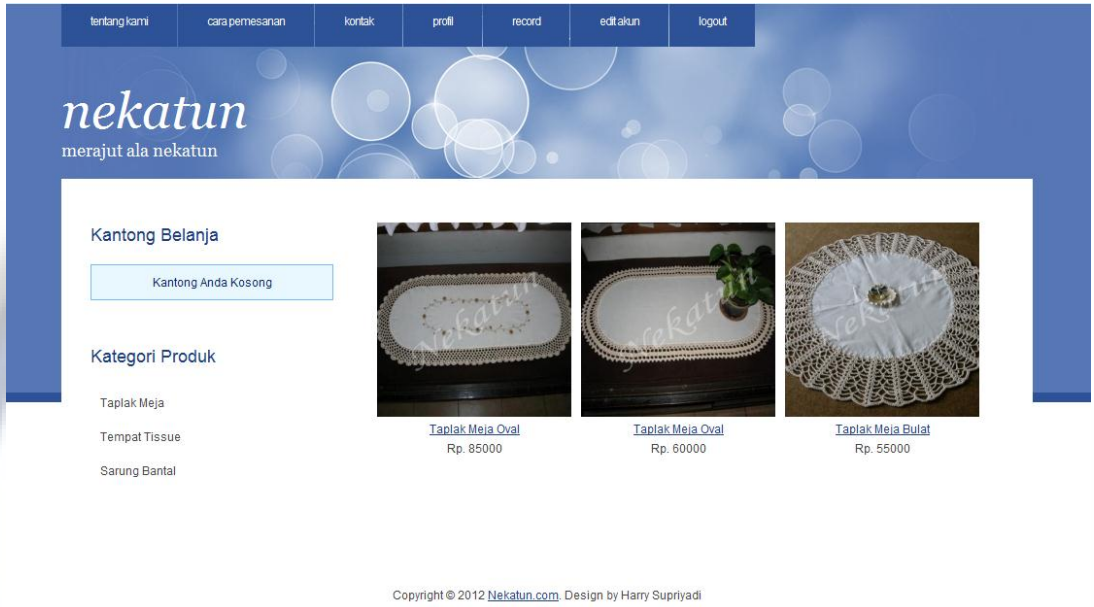
Gambar 5.22 Antarmuka Rekaman Belanja

8. Antarmuka Mengubah Akun *User*



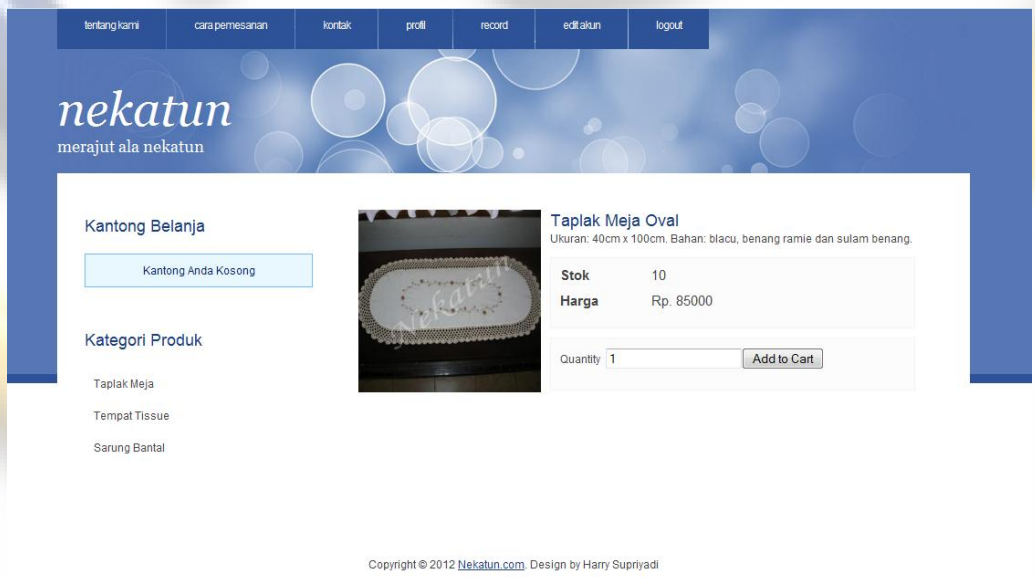
Gambar 5.23 Antarmuka Mengubah Akun *User*

9. Antarmuka Galeri Produk



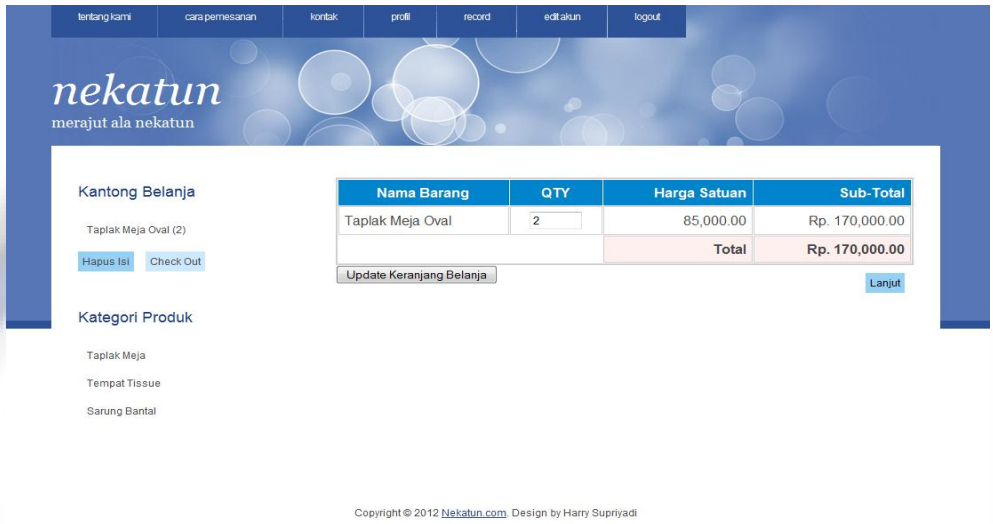
Gambar 5.24 Antarmuka Galeri Produk

10. Antarmuka Detail Produk



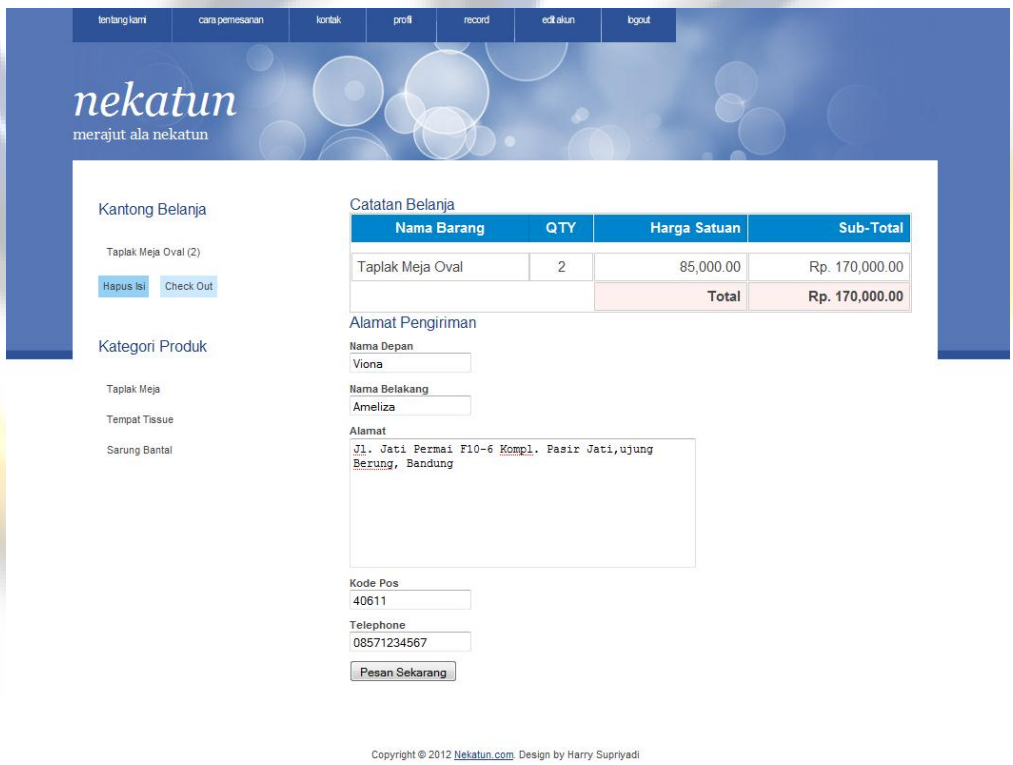
Gambar 5.25 Antarmuka Detail Produk

11. Antarmuka Kantong Belanja



Gambar 5.26 Antarmuka Kantong Belanja

12. Antarmuka Transaksi Pemesanan



Gambar 5.27 Antarmuka Transaksi Pemesanan

5.3 Pengujian Sistem

Pengujian ini bertujuan untuk menguji komponen sistem yang telah dirancang sebelumnya dan untuk memastikan bahwa setiap elemen dari sistem telah berfungsi sesuai dengan yang diharapkan. Dalam pengujian ini akan membahas mengenai metode pengujian, kasus dan hasil pengujian serta kesimpulan hasil pengujian

5.3.1 Metode Pengujian

Metode pengujian perangkat lunak diantaranya yaitu:

1. Black box testing

   Pengujian black box berfokus pada persyaratan fungsional perangkat lunak. Pengujian ini memungkinkan memperoleh kumpulan kondisi input yang akan mengerjakan seluruh fungsional program. Tujuan dari metode ini mencari kesalahan pada:

   a. Fungsi yang salah atau hilang
   b. Kesalahan pada *interface*
   c. Kesalahan pada struktur data atau akses *database*
   d. Kesalahan performansi
   e. Kesalahan inisialisasi dan tujuan akhir.

2. White box testing

   Pengujian white box adalah metode desain test case yang menggunakan struktur control desain procedural untuk memperoleh test case. Dengan pengujian white box dapat:

   a. Memberikan jaminan bahwa semua jalur independen pada suatu modul telah digunakan paling tidak satu kali.
   b. Menggunakan semua keputusan logis pada sisi *true and false.*
   c. Mengeksekusi semua loop pada batasan mereka dan pada batas operasional mereka.
   d. Menggunakan struktur data internal untuk menjamin validasinya.

Maka dalam rencana pengujian perangkat lunak ini akan menggunakan metode *Black Box*, karena pengujian *Black Box* memfokuskan pada keperluan fungsional dari perangkat lunak.

5.3.2 Kasus dan Hasil Pengujian

Kasus dan hasil pengujian Aplikasi Penjualan *Online* Menggunakan CodeIgniter ini sebagai berikut:

1. Pengujian *Login* Admin

Tabel 5.3 Pengujian *Login* Admin

Kasus dan Hasil Uji (Data Normal)			
Data masukan	**Yang diharapkan**	**Pengamatan**	**Kesimpulan**
Memasukkan *username* dan *password*	Masuk ke halaman admin	Dapat masuk ke halaman admin	Sesuai

2. Pengujian Menambah Kategori

Tabel 5.4 Pengujian Menambah Kategori

Kasus dan Hasil Uji (Data Normal)			
Data masukan	**Yang diharapkan**	**Pengamatan**	**Kesimpulan**
Klik "Tambah Kategori", masukkan nama kategori baru dan tekan tombol "Submit"	Muncul form penambahan kategori dan terciptanya kategori baru	Menampilkan form penambahan kategori dan Kategori bertambah	Sesuai

3. Pengujian Mengubah Kategori

Tabel 5.5 Pengujian Mengubah Kategori

Kasus dan Hasil Uji (Data Normal)			
Data masukan	**Yang diharapkan**	**Pengamatan**	**Kesimpulan**
Klik "Ubah" pada menu admin kategori dan masukkan data baru	Muncul form kategori dan mengubah kategori	Menampilkan form kategori dan Kategori berhasil diubah	Sesuai

4. Pengujian Menghapus Kategori

Tabel 5.6 Pengujian Menghapus Kategori

Kasus dan Hasil Uji (Data Normal)			
Data masukan	**Yang diharapkan**	**Pengamatan**	**Kesimpulan**
Klik "Hapus" pada menu admin kategori	Kategori yang dipilih terhapus	Kategori berhasil dihapus dari daftar	Sesuai

5. Pengujian Menambah Produk Baru

Tabel 5.7 Pengujian Menambah Produk Baru

Kasus dan Hasil Uji (Data Normal)			
Data masukan	**Yang diharapkan**	**Pengamatan**	**Kesimpulan**
Klik "Tambah Produk Baru", masukkan nama data produk dan tekan tombol "Submit"	Muncul form penambahan produk dan terciptanya produk baru	Menampilkan form penambahan produk dan Produk bertambah	Sesuai

6. Pengujian Menambah Gambar Produk

Tabel 5.8 Pengujian Menambah Gambar Produk

Kasus dan Hasil Uji (Data Normal)			
Data masukan	**Yang diharapkan**	**Pengamatan**	**Kesimpulan**
Klik "Tambah Gambar", lalu telusuri dan "Save Crop Image"	Bertambahnya gambar produk	Gambar produk bertambah dan berhasil di Crop	Sesuai

7. Pengujian Mengubah Data Produk

Tabel 5.9 Pengujian Mengubah Data Produk

Kasus dan Hasil Uji (Data Normal)			
Data masukan	**Yang diharapkan**	**Pengamatan**	**Kesimpulan**
Klik "Ubah" pada menu admin produk dan masukkan data baru	Muncul form detail data produk dan mengubah produk	Menampilkan data produk dan produk berhasil diganti	Sesuai

8. Pengujian Menghapus Produk

Tabel 5.10 Pengujian Menghapus Produk

Kasus dan Hasil Uji (Data Normal)			
Data masukan	**Yang diharapkan**	**Pengamatan**	**Kesimpulan**
Klik "Hapus" pada menu admin produk	Produk yang dipilih terhapus	Produk berhasil dihapus dari daftar	Sesuai

9. Pengujian Menambah *User*

Tabel 5.11 Pengujian Menambah *User*

Kasus dan Hasil Uji (Data Normal)			
Data masukan	**Yang diharapkan**	**Pengamatan**	**Kesimpulan**
Klik "Tambah *User* Baru", masukkan data *user,* tentukan level *user* dan tekan tombol "Registrasi"	Muncul form penambahan *user* baru dan terciptanya *user* baru	Menampilkan form penambahan *user* baru dan *user* bertambah	Sesuai

10. Pengujian Mengubah *User*

Tabel 5.12 Pengujian Mengubah *User*

Kasus dan Hasil Uji (Data Normal)			
Data masukan	**Yang diharapkan**	**Pengamatan**	**Kesimpulan**
Klik "Ubah" pada menu admin *user* dan masukkan data baru	Muncul form edit *user* dan mengubah *user*	Menampilkan form edit *user* dan *user* berhasil diubah	Sesuai

11. Pengujian Menghapus *User*

Tabel 5.13 Pengujian Menghapus *User*

Kasus dan Hasil Uji (Data Normal)			
Data masukan	**Yang diharapkan**	**Pengamatan**	**Kesimpulan**
Klik "Hapus" pada menu admin *User*	*User* yang dipilih terhapus	*User* berhasil dihapus dari daftar	Sesuai

12. Pengujian Mengubah Status Pesanan

Tabel 5.14 Pengujian Mengubah Status Pesanan

Kasus dan Hasil Uji (Data Normal)			
Data masukan	**Yang diharapkan**	**Pengamatan**	**Kesimpulan**
Klik "Detail" pada menu admin Laporan dan ganti status kemudian tekan "simpan perubahan"	Mengubah status pesanan pelanggan	Menampilkan detail pesanan dan berhasil mengubah status pesanan	Sesuai

13. Pengujian Mengubah Pengaturan Situs

Tabel 5.15 Pengujian Mengubah Pengaturan Situs

Kasus dan Hasil Uji (Data Normal)			
Data masukan	**Yang diharapkan**	**Pengamatan**	**Kesimpulan**
Masuk ke menu admin Pengaturan dan masukkan data situs terbaru kemudian tekan "simpan"	Mengubah halaman *homepage*	Menampilkan form opsi situs dan halaman *homepage* berhasil diubah	Sesuai

14. Pengujian Registrasi

Tabel 5.16 Pengujian Registrasi

Kasus dan Hasil Uji (Data Normal)			
Data masukan	**Yang diharapkan**	**Pengamatan**	**Kesimpulan**
Masukkan *username*, email	*User* dapat mendaftarkan	Berhasil mendaftarkan	Sesuai

dan *password* kemudian tekan tombol "Registrasi"	sebagai pelanggan	sebagai pelanggan	

15. Pengujian *Login User*

Tabel 5.17 Pengujian *Login User*

Kasus dan Hasil Uji (Data Normal)			
Data masukan	**Yang diharapkan**	**Pengamatan**	**Kesimpulan**
Memasukkan *username* dan *password*	Masuk ke halaman public dan melakukan pemesanan	Berhasil masuk ke halaman publik dan sudah dapat melakukan pemesanan	Sesuai

16. Pengujian Mengubah Profil *User*

Tabel 5.18 Pengujian Mengubah Profil *User*

Kasus dan Hasil Uji (Data Normal)			
Data masukan	**Yang diharapkan**	**Pengamatan**	**Kesimpulan**
Masukkan data *user* beserta alamat lengkap.	Muncul form profil dan menyimpan data *user*	Berhasil mengubah profil *user*	Sesuai

17. Pengujian Mengubah Akun *User*

Tabel 5.19 Pengujian Mengubah Akun *User*

Kasus dan Hasil Uji (Data Normal)			
Data masukan	**Yang diharapkan**	**Pengamatan**	**Kesimpulan**
Masukkan *username*, email, *password* baru dan confirm password kemudian tekan tombol "Update"	Menampilkan akun *user* sekarang dan dapat diubah dengan akun yang baru	Berhasil mengubah akun *user*	Sesuai

18. Pengujian Menambah Kantong Belanja

Tabel 5.20 Pengujian Menambah Kantong Belanja

Kasus dan Hasil Uji (Data Normal)			
Data masukan	**Yang diharapkan**	**Pengamatan**	**Kesimpulan**
Memilih produk dan meng-*input*-kan kuantitas produk yang dibeli	Produk yang dipilih dapat masuk ke kantong belanja dan menjumlahkan biaya belanjaan	Berhasil memasukkan produk pada kantong belanja dan berhasil mengakumulasikan total dan biaya	Sesuai

19. Pengujian *Update* Kantong Belanja

Tabel 5.21 Pengujian *Update* Kantong Belanja

Kasus dan Hasil Uji (Data Normal)			
Data masukan	**Yang diharapkan**	**Pengamatan**	**Kesimpulan**
Masukkan kuantitas yang	Dapat mengubah kuntitas sekaligus	Berhasil mengubah kuantitas produk	Sesuai

diinginkan setiap produknya pada kantong belanja, kemudian tekan tombol “Update Kantong Belanja”	total biayanya	dan total biaya	

20. Pengujian Transaksi Pemesanan

Tabel 5.22 Pengujian Transaksi Pemesanan

Kasus dan Hasil Uji (Data Normal)			
Data masukan	**Yang diharapkan**	**Pengamatan**	**Kesimpulan**
Masukkan produk yang dipilih beserta alamat pengiriman lengkap kemudian tekan “Pesan sekarang”	Dapat mengirim data pesanan pelanggan kepada admin	Berhasil mengirim data pesanan dengan munculnya pesan “Data pesanan sudah terkirim”	Sesuai

5.3.3 Kesimpulan Hasil Pengujian

Berdasarkan hasil pengujian dengan metode *black box* pada data table di atas dapat diambil kesimpulan bahwa perangkat lunak Aplikasi Penjualan *Online* dengan Menggunakan CodeIgniter ini secara fungsional telah memberikan hasil dari setiap proses sesuai dengan yang diharapkan.

BAB VI
PENUTUP

6.1 Kesimpulan

Berdasarkan hasil dari tugas akhir yang dilalukan oleh peneliti dan dari uraian-uraian yang telah peneliti paparkan pada bab-bab sebelumnya mengenai Aplikasi Penjualan *Online* dengan Menggunakan *framework* CodeIgniter, maka diambil kesimpulan bahwa:

1. *Framework* CodeIgniter terdapatnya model MVC yang dapat mempermudah pembedaan antara tampilan dan program sehingga sangat baik untuk proyek yang di mana designer bekerja dengan file template yang akan memperkecil banyaknya kode di setiap template karena sudah dipisahkan dengan baik. maka dari itu peneliti menggunakan *framework* CodeIgniter tersebut karena sangat sederhana dalam pengembangan aplikasi berbasis web.
2. Pada aplikasi penjualan *online* dengan menggunakan Codeigniter ini dapat mengelola produk dengan baik, dapat menghitung pemesanan dan pembuatan laporan penjualan secara otomatis.
3. Upaya yang dilakukan untuk mengatasi permasalahan yang terjadi maka dibangunlah Aplikasi Penjualan *Online* dengan Menggunakan CodeIgniter, yang berfungsi dapat meminimalisir waktu, karena pelanggan dapat memesanan langsung secara *online* tanpa harus bertanya-tanya mengenai harga produk dan mengurangi ketidakakuratan data, memudahkan konsumen untuk mengontrol pesanan mereka karena adanya rekaman pesanan konsumen yang diproses, memudahkan bagi admin dalam melakukan peng-*input*-an data dan mengelola data pemesanan serta membantu admin dalam pembuatan laporan penjualan.

6.2 Saran

Dalam hal ini peneliti memberikan saran-saran sebagai bahan pertimbangan dan masukkan untuk lebih mengembangkan usahanya dengan memanfaatkan media *online*. Adapun saran-saran dari peneliti adalah sebagai berikut:

1. Aplikasi ini dikembangkan pada transaksi pembayaran dengan menggunakan Paypal.
2. Aplikasi ini dapat dikembangkan dengan adanya aplikasi laporan keuangan untuk memudahkan bagian keuangan dalam membuat laporan.
3. Dibuatkan perbedaan bagi pelanggan *reseller* dan pelanggan biasa dengan membuatkan ID *user* khusus bagi *reseller*.

DAFTAR PUSTAKA

1. Abdillah, E. (2010, Agustus 23). *Pengertian Internet dan Intranet*. Retrieved Mei 13, 2012, from Cara Membuat Blog: http://www.membuatblog.web.id/2010/08/pengertian-internet-dan-intarnet.html

2. Afnarius, S. (2010, Mei 3). *Perancangan Sistem*. Retrieved Mei 13, 2012, from Kompasiana: http://edukasi.kompasiana.com/2010/05/03/perancangan-sistem/

3. Ahira, A. (2011, November 1). *Mengenal Pengertian Bisnis.* Retrieved Mei 15, 2012, from Anne Ahira.com: http://www.anneahira.com/pengertian-bisnis.htm

4. Guntara, A. (2011, Oktober 31). *Rekayasa Perangkat Lunak Software Process Model I*. Retrieved Mei 13, 2012, from Blog Agun Guntara: http://nugtachinsui.blogspot.com/2011_10_01_archive.html

5. Irfan. (2009, Juni). *Tugas 1 RPL*. Retrieved Mei 15, 2012, from Irfante blog: http://irfante06.blog.unsoed.ac.id/files/2009/06/tugas-1-rpl.doc

6. Jatnika, C. (2009, Maret 11). *Konsep Object Oriented Programming Pada Framework PrototypeJs dan JQuery*. Retrieved Mei 13, 2012, from T-10 Homepage: http://chandrajatnika.com/2009/03/konsep-object-oriented-programming-pada-framework-prototypejs-dan-jquery/

7. Naufal. (2011, September 24). *Pengertian PHP dan Kelebihan Bahasa Pemrograman PHP*. Retrieved Mei 13, 2012, from Dunia Informasi dan Warta: http://www.diwarta.com/1000/pengertian-php-dan-kelebihan-bahasa-pemrograman-php/

8. Pradipta, P. A. (2011, September 21). *Metode dan Model Proses Rekayasa Perangkat Lunak*. Retrieved Mei 15, 2012, from L-SPICA: http://lindroidspica.blogspot.com/2011/09/metode-dan-model-rekayasa.html

9. Rhomadhona, Fahmi, & dkk. (2012, Februari 16). *Pengertian Penjualan*. Retrieved Mei 13, 2012, from Desain Website.net: http://desainwebsite.net/artikel-ilmiah/pengertian-penjualan

10. Riyanto. (2011). *Membuat Sendiri Aplikasi E-Commerce dengan PHP dan MySQL Menggunakan CodeIgniter dan JQuery.* Yogyakarta: Andi.

11. Saputro, H. W. (2007, Agustus 1). *Pengertian Website dan Unsur-Unsurnya*. Retrieved Mei 13, 2012, from balebengong.net: http://www.balebengong.net/topik/teknologi/2007/08/01/pengertian-website-dan-unsur-unsurnya.html

12. Suhendar, A., & Gunadi, H. (2002). *Visual Modeling Menggunakan UML dan Rational Rose.* Bandung: Informatika.

13. Tim Dunia Script, c. (2009). *Pengertian Tentang XAMPP*. Retrieved Mei 13, 2012, from Kumpulan Artikel dan Tutorial Terlengkap: http://blog.duniascript.com/pengertian-tentang-xampp.html

14. Wiswakarma, K. (2010). *9 Langkah Menjadi Master Framework Codeigniter.* Bekasi: Lokomedia.

15. Yuli, I. (2011, September 26). *Kelebihan dan Kekurangan PHP*. Retrieved Mei 13, 2012, from Indriyuli-yuliindri: http://iin-indriyuli17.blogspot.com/2011/09/kelebihan-dan-kekurangan-php_26.html