

**APLIKASI PENJUALAN *PC GAME* SECARA *ONLINE***  
**(STUDI KASUS : PENJUALAN *PC GAME* DI *GO\_GAMING*)**

**TUGAS AKHIR**

Diajukan Untuk Memenuhi Salah Satu Syarat Dalam Menempuh Ujian Sidang  
Sarjana di Program Studi Sistem Informasi

Oleh :

**Febry Gustiawan**

**11.07.001**

**PROGRAM STUDI SISTEM INFORMASI**  
**FAKULTAS TEKNIK UNIVERSITAS WIDYATAMA**  
**BANDUNG**  
**2012**

# LEMBAR PENGESAHAN

## APLIKASI PENJUALAN *PC GAME* SECARA *ONLINE* (STUDI KASUS : PENJUALAN *PC GAME* DI *GO\_GAMING*)

### TUGAS AKHIR

Program Studi Sistem Informasi

Fakultas Teknik

Universitas Widyatama

Oleh :

Febry Gustiawan

1107001

Telah disetujui dan disahkan di Bandung, Tanggal ... Juni 2012

Pembimbing,

Maniah, Ir.,M.T.

NID. 0427076701

Ka.Prodi Sistem Informasi,

Dekan Fakultas Teknik,

M.Rozahi Istambul, S.Komp.,M.T.

NID. 0414106701

Setiadi Yazid, Ir.,M.Sc.,Phd.

NID. 0315085402

## **SURAT PERNYATAAN**

Saya yang bertanda tangan dibawah ini :

Nama : Febry Gustiawan  
NPM : 11.07.001  
Tempat Dan Tanggal Lahir : Merauke, 19 Februari 1989  
Alamat : Komplek Bumi Panyileukan Citra AB  
2 No 2, RT 02/RW 13. Bandung 40614

Menyatakan bahwa Laporan Tugas Akhir ini yang berjudul : **“APLIKASI PENJUALAN PC GAME SECARA ONLINE (STUDI KASUS : PENJUALAN PC GAME DI GO\_GAMING)”**. 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, ... Juni 2012

**Febry Gustiawan**

## ABSTRAK

Go\_Gaming adalah sebuah re-seller yang bergerak dibidang penjualan *pc game* dan mulai berjalan sejak bulan September 2011. Pada awal memulai bisnis ini go\_gaming menggunakan *Kaskus* dan *Twitter* sebagai media untuk menjual dan mempromosikan dagangannya, tidak hanya membeli bahkan konsumen dapat memesan game yang tidak ada didalam daftar. Apabila *game* tersebut ada maka *game* itu pun akan dijual kepada konsumen yang memesannya dan memasukkan kedalam daftar *game*. Lambat laun go\_gaming merasa membutuhkan sebuah aplikasi yang berbasis *web* untuk lebih bisa mengontrol data lebih baik lagi dan juga lebih praktis, sehingga bisnisnya dapat berjalan lebih lancar lagi. Untuk itu penulis mempunyai ide membuat sebuah aplikasi penjualan dan pemesanan secara *online* yang menggunakan bahasa pemrograman *php* dan sebagai *toolnya* penulis menggunakan *macromedia dreamweaver 8*, sedangkan untuk mengedit gambar menggunakan *paint*, *fireworks 8* atau juga bisa dengan *adobe photoshop*. Lalu untuk melihat hasil penulis menggunakan *browser mozilla firefox*, selain *firefox* bisa juga menggunakan *browser* yang lainnya. Untuk menjalankan aplikasi tidak lupa juga harus di instal xampp yang berfungsi sebagai perangkat yang menggabungkan tiga aplikasi ke dalam satu paket, yaitu *Apache*, *MySQL* dan *PHPMYAdmin*. Aplikasi ini terdapat konten-konten menu yang mudah diakses dan dimengerti oleh konsumen dan terdapat kategori-kategori untuk produk, sehingga konsumen dapat mencari game sesuai dengan kategori yang diinginkan. Bukan hanya itu, tapi juga terdapat produk-produk *best seller*.

**Kata kunci :** *aplikasi, game, penjualan*

## **ABSTRACT**

*Go\_Gaming is a re-seller who is engaged in sales of PC games and started running since September 2011. At the beginning of starting this business using Kaskus go\_gaming and Twitter as a medium for selling and promoting merchandise, not just buy even consumers can order a game that does not exist in the list. If the game is there then the game would be sold to consumers who order them and put into the game list. Gradually go\_gaming feel you need a web-based applications for more control over the data even better and also more practical, so the business can run much more smoothly. To the authors had the idea to make an application online sales and bookings using php programming language and as an author using macromedia dreamweaver toolnya 8, whereas for editing images using paint, fireworks 8 or also with adobe photoshop. Then to see the author using mozilla firefox browser, but firefox can also use another browser. To run the application must also not forget in install xampp which serves as a device that combines three applications into one package, namely Apache, MySQL and PHPMyAdmin. These applications have a menu content is easily accessible and understood by consumers and there are categories for products, so consumers can search for games according to the desired category. Not only that, but there is also a best seller products.*

**Key words:** *application, game, sales*

## KATA PENGANTAR

Puji syukur kehadirat ALLAH SWT yang telah memberi rahmat dan hidayah-Nya, sehingga penulis dapat menyelesaikan Laporan Tugas Akhir ini dengan judul **“APLIKASI PENJUALAN PC GAME SECARA ONLINE (STUDI KASUS : PENJUALAN PC GAME DI GO\_GAMING)”**

Dalam penyusunan Tugas Akhir ini, penulis tak jarang selalu mendapat hambatan, tantangan dan kesulitan, hal ini disebabkan karena banyak faktor dari keterbatasan dan kemampuan yang dimiliki penulis.

Penulis juga sadar sepenuhnya bahwa tanpa bantuan, bimbingan, petunjuk serta dorongan dari berbagai pihak, tak mungkin Tugas Akhir ini dapat selesai, sehubungan dengan hal tersebut, maka penulis dalam kesempatan ini dengan segala kerendahan hati penulis mengucapkan terima kasih kepada :

1. Ibu Maniah, Ir.,M.T. selaku pembimbing Tugas Akhir.
2. Bapak M. Rozahi Istambul, S.komp., M.T. selaku Ketua Jurusan Sistem Informasi Universitas Widyatama.
3. Staff Dosen Universitas Widyatama yang telah memberikan banyak ilmu kepada penulis.
4. Kedua orang tua, adik dan keluarga, atas segala dukungan, dorongan, bantuan, dan juga doanya.
5. Teman – teman semua angkatan jurusan SI(Sistem Informasi), khususnya SI 07(Puja, Budi, Faisal, Romasi, Deny, Tita, Karina, Rosid, Aji, Adit, Fahru, Deden, Desty, Mey, Ponco, Lucky, Acip, Rixon, Puji, Laura, Romasi.) Univ.Widyatama yang telah membatu dengan dukungan, dorongan dan doanya.
6. Teman – teman rumah sekaligus futsal(Andre, Rudi, Suhe, Dipta, Andria, Hari, Niko, Bayu, Bobby, Aris, Agunk, Zilka, Ando, Eka, Widi. Adit, Yoga, Gera, Yudha, Jeje.), yang telah memberikan dukunganya.

7. Ulfi Ambarwati Rahayu yang selalu setia membantu, memberi support, mendoakan dan mendampingi penulis, sehingga laporan Tugas Akhir ini selesai.
8. Staff Karyawan Universitas Widyatama yang telah banyak membantu dalam urusan administrasinya.

Penullis menyadari bahwa kajian yang penulis bahas belum dapat dikatakan sempurna. Tetapi walaupun demikian penulis mengharapkan semoga Kajian pada Laporan Tugas Akhir ini dapat bermanfaat baik bagi penulis maupun bagi yang memerlukannya.

Dan penulis minta maaf bila pada penyuguhan Laporan Tugas Akhir ini masih banyak kesalahan baik dari tulisan maupun bahasanya. Oleh karena itulah segala ucapan, teguran, kritik, serta saran-saran yang sifatnya membangun sangat diharapkan penulis.

Bandung, ... Juni 2012

**Febry Gustiawan**

Penulis



## DAFTAR ISI

### LEMBAR JUDUL

LEMBAR PENGESAHAN .....i

SURAT PERNYATAAN .....ii

ABSTRAK .....iii

*ABSTRACT* .....iv

KATA PENGANTAR .....v

DAFTAR ISI .....vii

DAFTAR TABEL .....xii

DAFTAR GAMBAR .....xiii

### BAB I PENDAHULUAN

1.1 Latar Belakang.....I-1

1.2 Identifikasi Masalah.....I-2

1.3 Rumusan Masalah.....I-2

1.4 Batasan Masalah.....I-3

1.5 Maksud dan Tujuan.....I-3

1.6 Metodologi Penelitian.....I-4

1.7 Sistematika Penulisan.....I-7



## **BAB II DASAR TEORI**

2.1	Dasar Teori.....	II-1
2.1.1	Aplikasi.....	II-1
2.1.2	<i>HTML</i> .....	II-2
2.1.3	<i>XAMPP</i> .....	II-3
2.1.3.1	<i>Apache</i> .....	II-3
2.1.3.2	<i>MySQL</i> .....	II-3
2.1.3.3	<i>PHP</i> .....	II-4
2.1.3.4	<i>Perl</i> .....	II-4
2.1.4	PHPMysqlAdmin.....	II-5
2.1.5	CSS (Cascade Style Sheet).....	II-5
2.1.6	<i>Web</i> .....	II-5
2.1.7	Database .....	II-6
2.1.8	Penjualan.....	II-7
2.1.9	Metode Penjualan Dengan Paypal.....	II-7
2.1.10	<i>E-Commerce</i> .....	II-9
2.2	Pengembangan Sistem Informasi .....	II-11
2.2.1	Metodologi Pengembangan Sistem Informasi .....	II-11
2.2.1.1	Model <i>Waterfall</i> .....	II-11
2.2.1.2	Model Sekuensial <i>Linier</i> .....	II-14
2.2.1.3	Model Prototipe.....	II-17
2.2.1.4	Model <i>Rapin Application Development</i> .....	II-19
2.2.1.5	Model Spiral.....	II-22

2.2.1.6 Model Yang Digunakan.....	II-24
2.3 Konsep Objek Berorientasi.....	II-25
2.3.1 Bahasa Pemrograman Yang digunakan .....	II-27
2.4 UML(Unified Modelling Language).....	II-28

### **BAB III GAMBARAN UMUM USAHA DAN ANALISIS**

3.1 Sejarah Singkat Usaha.....	III-1
3.1.1 Latar Belakang Usaha .....	III-1
3.1.2 Uraian Pekerjaan .....	III-2
3.2 Analisis .....	III-3
3.2.1 Identifikasi Masalah.....	III-3
3.3 Deskripsi Sistem Lama .....	III-3
3.3.1 Penjualan Di <i>Twitter</i> .....	III-3
3.3.1.1 Contoh penjualan di <i>Twitter</i> .....	III-4
3.3.1.1.1 Promosi Di <i>Twitter</i> .....	III-4
3.3.1.1.2 Pemesanan Di <i>Twitter</i> .....	III-5
3.3.2 Penjualan Di <i>Kaskus</i> .....	III-6
3.3.2.1 Contoh Penjualan Di <i>Kaskus</i> .....	III-7
3.3.2.1.1 Promosi Di <i>Kaskus</i> .....	III-7
3.3.2.1.2 Pemesanan Di <i>Kaskus</i> .....	III-8
3.3.2.2 Contoh Transaksi Pengiriman.....	III-9
3.4 Hasil Analisa Sistem.....	III-10
3.4.1 Evaluasi Sistem Saat ini.....	III-10
3.5 Analisa Sistem Yang Akan Dibuat.....	III-11

3.6	Analisa Kebutuhan Informasi.....	III-13
3.6.1	Kelemahan Sistem Saat ini .....	III-13
3.6.2	Kebutuhan Yang Terpenuhi Pada Kedua Sistem .....	III-13
3.6.3	Tujuan Sistem Baru.....	III-14
3.7	Model Metodologi <i>Waterfall</i> .....	III-14

## **BAB IV PERANCANGAN SISTEM**

4.1	Deskripsi Sistem.....	IV-1
4.1.1	Unified Modelling Language(UML) .....	IV-1
4.2	Perancangan Diagram Sistem.....	IV-1
4.2.1	Use Case Diagram.....	IV-1
4.2.2	Definisi <i>Actor</i> .....	IV-3
4.2.4	Definisi <i>Use Case</i> .....	IV-3
4.2.5	Skenario <i>Use Case</i> .....	IV-4
4.2.6	Sequence Diagram .....	IV-11
4.2.6.1	<i>Sequence Diagram Login User</i> .....	IV-11
4.2.6.2	<i>Sequence Diagram Menu Header</i> .....	IV-12
4.2.6.3	<i>Sequence Diagram Category</i> .....	IV-13
4.2.6.4	<i>Sequence Diagram Product Best Seller</i> .....	IV-13
4.2.6.5	<i>Sequence Diagram Shout Box</i> .....	IV-14
4.2.6.6	<i>Sequence Diagram Find Me On</i> .....	IV-14
4.2.6.7	<i>Sequence Diagram Guest Book</i> .....	IV-15
4.2.7	Collaboration Diagram.....	IV-15
4.2.7.1	<i>Collaboration Diagram Login User</i> .....	IV-16
4.2.7.2	<i>Collaboration Diagram Menu Header</i> .....	IV-17

4.2.7.3	<i>Collaboration Diagram Category</i> .....	IV-17
4.2.7.4	<i>Collaboration Product Best Seller</i> .....	IV-18
4.2.7.5	<i>Collaboration Diagram Shout Box</i> .....	IV-18
4.2.7.6	<i>Collaboration Diagram Find Me On</i> .....	IV-19
4.2.7.7	<i>Collaboration Diagram Guest Book</i> .....	IV-19
4.2.8	Kamus Data.....	IV-20
4.2.8.1	<i>Class Diagram Penjualan PC Game</i> .....	IV-24
4.3	Deployment Diagram .....	IV-25
4.4	Rancangan Menu .....	IV-25
4.4.1	Rancang <i>Menu User</i> .....	IV-25
4.4.2	Rancang <i>Menu Konsumen</i> .....	IV-26
4.5	Perancangan Antarmuka.....	IV-26
4.6	Perancangan Teknologi .....	IV-35
4.6.1	Kebutuhan Perangkat Keras <i>Developer</i> .....	IV-35
4.6.2	Pengembangan Aplikasi.....	IV-35
4.6.3	Perangkat Keras dan Lunak Untuk Menjalankan Aplikasi.....	IV-36

## **BAB V IMPLEMENTASI**

5.1	Analisa Kebutuhan Informasi.....	V-1
5.1.1	Analisa Kebutuhan Koneksi.....	V-1
5.1.2	Syarat Minimum Spesifikasi.....	V-1
5.1.3	Analisa Kebutuhan Perangkat Lunak.....	V-2
5.2	Implementasi Antarmuka .....	V-2
5.2.1	Antarmuka Form .....	V-2
5.3	Informasi Semua Menu Setelah Dijalankan.....	V-8

## **BAB VI KESIMPULAN DAN SARAN**

6.1	Kesimpulan.....	VI-1
6.2	Saran.....	VI-1

## **DAFTAR PUSTAKA**



## DAFTAR TABEL

Tabel 4.1 Definisi <i>Actor</i> .....	IV-3
Tabel 4.2 Definisi <i>Use Case</i> .....	IV-3
Tabel 4.3 Skenario <i>Login User</i> .....	IV-4
Tabel 4.4 Skenario <i>Home</i> .....	IV-5
Tabel 4.5 Skenario <i>Profile</i> .....	IV-5
Tabel 4.6 Skenario <i>Product</i> .....	IV-6
Tabel 4.7 Skenario <i>How To Buy</i> .....	IV-6
Tabel 4.8 Skenario <i>Shopping Chart</i> .....	IV-7
Tabel 4.9 Skenario <i>Contact Me</i> .....	IV-7
Tabel 4.10 Skenario <i>Status Orders</i> .....	IV-8
Tabel 4.11 Skenario <i>Category</i> .....	IV-8
Tabel 4.12 Skenario <i>Product Best Seller</i> .....	IV-9
Tabel 4.13 Scenario <i>Shout Box</i> .....	IV-9
Tabel 4.14 Skenario <i>Find Me On</i> .....	IV-10
Tabel 4.15 Tabel <i>Guest Book</i> .....	IV-10
Tabel 4.16 Tabel Kategori.....	IV-20
Tabel 4.17 Tabel Kota .....	IV-20
Tabel 4.18 Tabel <i>Orders</i> .....	IV-21
Tabel 4.19 Tabel <i>Orders Detail</i> .....	IV-21
Tabel 4.20 Tabel <i>Orders Temporary</i> .....	IV-22
Tabel 4.21 Tabel <i>Produk</i> .....	IV-23
 Tabel 5.1 Informasi Semua Menu Setelah Dijalankan.....	 V-8

## DAFTAR GAMBAR

Gambar 2.1 Model Transaksi <i>eCommerce</i> .....	II-10
Gambar 2.2 Pemodelan <i>Waterfall</i> .....	II-11
Gambar 2.3 Model <i>Waterfall</i> 6 Tahap .....	II-12
Gambar 2.4 Pemodelan Sekuensial Linear .....	II-16
Gambar 2.5 Pemodelan Prototipe .....	II-18
Gambar 2.6 Pemodelan <i>RAD</i> .....	II-21
Gambar 2.7 Pemodelan Spiral.....	II-23
Gambar 3.1 Promosi di <i>Twitter</i> .....	III-5
Gambar 3.2 Pemesanan di <i>Twitter</i> .....	III-6
Gambar 3.3 Promosi di <i>Kaskus</i> .....	III-7
Gambar 3.4 Pemesanan di <i>Kaskus</i> .....	III-8
Gambar 3.5 Bukti Resi Pengiriman .....	III-9
Gambar 3.6 <i>Flowchart</i> Sistem Lama .....	III-11
Gambar 3.7 <i>Flowchart</i> Sistem Baru.....	III-12
Gambar 4.1 <i>Use Case Diagram</i> .....	IV-2
Gambar 4.2 <i>Sequance Diagram Login User</i> .....	IV-11
Gambar 4.3 <i>Sequance Diagram Menu Header</i> .....	IV-12
Gambar 4.4 Sequence Diagram Category .....	IV-13
Gambar 4.5 <i>Sequance Diagram Product Best Seller</i> .....	IV-13
Gambar 4.6 <i>Sequance Diagram Shout Box</i> .....	IV-14
Gambar 4.7 <i>Sequence Diagram Find Me On</i> .....	IV-14
Gambar 4.8 <i>Sequance Diagram Guest Book</i> .....	IV-15



Gambar 4.9 <i>Collaboration Diagram Login User</i> .....	IV-16
Gambar 4.10 <i>Collaboration Diagram Menu Header</i> .....	IV-17
Gambar 4.11 <i>Collaboration Diagram Category</i> .....	IV-17
Gambar 4.12 <i>Collaboration Product Best Seller</i> .....	IV-18
Gambar 4.13 <i>Collaboration Diagram Shout Box</i> .....	IV-18
Gambar 4.14 <i>Collaboration Diagram Find Me On</i> .....	IV-19
Gambar 4.15 <i>Collaboration Diagram Guest Book</i> .....	IV-19
Gambar 4.16 <i>Class Diagram Penjualan PC Game</i> .....	IV-24
Gambar 4.17 <i>Deployment Diagram</i> .....	IV-25
Gambar 4.18 <i>Rancang Menu User</i> .....	IV-25
Gambar 4.19 <i>Rancang Menu Konsumen</i> .....	IV-26
Gambar 4.20 <i>Menu login User</i> .....	IV-27
Gambar 4.21 <i>Menu User</i> .....	IV-28
Gambar 4.22 <i>Menu Header</i> .....	IV-29
Gambar 4.23 <i>Menu Side Category</i> .....	IV-30
Gambar 4.24 <i>Menu Side Product Best Seller</i> .....	IV-31
Gambar 4.25 <i>Menu Side Shout Box</i> .....	IV-32
Gambar 4.26 <i>Menu Side Find Me On</i> .....	IV-33
Gambar 4.27 <i>Menu Side Guest Book</i> .....	IV-34
Gambar 5.1 <i>Login User</i> .....	V-3
Gambar 5.2 <i>Menu User</i> .....	V-3
Gambar 5.3 <i>Menu Header</i> .....	V-4
Gambar 5.4 <i>Menu Side Category</i> .....	V-5
Gambar 5.5 <i>Menu Side Product Best Seller</i> .....	V-5
Gambar 5.6 <i>Menu Side Shout Box</i> .....	V-6
Gambar 5.7 <i>Menu Side Find Me On</i> .....	V-7

Gambar 5.8 *Menu Side Guest book*..... V-7



## LAMPIRAN B

### *CURRICULUM VITAE*



#### ✚ DATA PRIBADI

Nama Lengkap : Febry Gustiawan  
Tempat, Tanggal lahir : Merauke, 19 February 1989  
Jenis Kelamin : Laki-laki  
Agama : Islam  
Status : Belum Menikah  
Kewarganegaraan : WNI(Warga Negara Indonesia)  
Alamat : Panyileukan Citra AB 2 No.2, Bandung  
No. Telp/HP : 022-7835072/085860264235  
Email : febry.gustiawan@gmail.com

#### ✚ PENDIDIKAN FORMAL

Sekolah/Universitas	Fakultas/Jurusan	Kota	Tahun Lulus
SDN Panyileukan 01 Bandung	-	Bandung	2001
SMPN 18 Bandung	-	Bandung	2004
SMA BPI 1 Bandung	IPS	Bandung	2007
Univ. Widyatama Bandung	Sistem Informasi	Bandung	2012

#### ✚ PENGALAMAN ORGANISASI

Organisasi	Posisi	Tahun
Karang Taruna	Sekretaris	2007
Karang Taruna	Divisi Olahraga	2008
HIMASI (Himpunan Mahasiswa Sistem Informasi)	Anggota	2007 – 2008

MAPSI (Malam Perkenalan Sistem Informasi)	Wakil Ketua	2008
Senat Mahasiswa Fakultas Teknik	Departemen Sosial	2009 – 2010
ITCNT (Information Technology Competition and Talkshow)	Anggota	2010
WTE (Widyatama Technology Event)	Logistik	2010

Demikian *Cirriculum Vitae* ini saya buat dengan sebenar-benarnya.

Bandung, Juni 2012

**Febry Gustiawan**  
Penulis



## **BAB I**

### **PENDAHULUAN**

#### **1.1 Latar Belakang**

Indonesia merupakan salah satu Negara yang mempunyai banyak wirausahaan yang cukup besar. Salah satunya adalah penjualan *pc game*, karena *game* mempunyai banyak manfaat yaitu dapat membantu manusia dalam menghilangkan kepenatan atau mengisi waktu luang, selain itu sebagai sarana penghibur, hobi, ataupun menjadi ajang bisnis yang dapat menghasilkan sejumlah uang. *Game* tidak hanya dimainkan oleh anak – anak dan remaja saja, bahkan orang dewasa pun juga memainkannya.

*Game* sendiri terbagi menjadi 2 golongan yaitu *game online* dan *game offline*. *Game online* adalah *game* yang dimainkan secara *online*/terhubung dengan internet yang dapat berinteraksi dengan orang banyak, sedangkan *game offline* adalah *game* yang hanya di mainkan oleh seorang ataupun beberapa *player* saja. Maka dari itu *user* mempunyai pemikiran untuk berwirausaha dalam penjualan *pc game* dan usahanya itu di beri nama “Go\_Gaming”.

Disini penulis mendapatkan inspirasi untuk membuatkan sebuah aplikasi berbasis *web*, yang nantinya akan membantu *user* dalam menjalankan usahanya, Aplikasi *web* tersebut mempunyai fungsi yaitu sebagai sarana dan prasana mediator dalam menjual yaitu dari *user* ke konsumen ataupun pemesanan *pc game* yaitu dari konsumen ke *user*.

Teknologi mempunyai peranan penting bagi manusia dewasa ini. perkembangan teknologi sangatlah pesat bagi umat manusia, dikarenakan kepraktisanya yang dapat membantu kinerja manusia dalam membantu tugasnya.

Adapun beberapa jenis jejaring sosial dan forum jual/beli yang dapat membantu *user* untuk mempromosikan dagangannya dan menjualnya.

Teknologi pun menjadi sarana dan prasarana hiburan bagi penggunanya, salah satunya adalah *game*. Para *developer* pun berlomba untuk menyajikan *game-game* jagoan mereka untuk bersaing di pasaran, begitupun dengan para *reseller*-nya. Mereka berlomba dengan para *reseller* lain untuk menjual *game-game update* para *developer* tersebut. Dari latar belakang tersebut penulis terilhami untuk membuat sebuah “**APLIKASI PENJUALAN PC GAME SECARA ONLINE (STUDI KASUS : PENJUALAN PC GAME DI GO\_GAMING).**”

## **1.2 Identifikasi Masalah**

Dari latar belakang masalah yang telah dijelaskan, maka dapat diuraikan identifikasi masalahnya adalah :

1. Selama ini *user* hanya memanfaatkan beberapa jenis jejaring sosial maupun forum jual/beli sebagai mediator untuk menjalankan usahanya yaitu menjual *pc game*, jadi dirasa masih kurang maksimal.
2. Diperlukannya sebuah aplikasi berbasis *web*, agar *user* dapat menjalankan usahanya menjual *pc game* dengan lancar secara luas dan *online* kepada masyarakat.
3. Dibutuhkannya data-data dari *user* agar proses pembuatan aplikasi dapat berjalan sesuai dengan rencana.

### **1.3 Rumusan Masalah**

Adapun yang menjadi permasalahan dalam penelitian ini, adalah sebagai berikut :

1. Bagaimana usaha penjualan *pc game* ini dapat berjalan dengan maksimal, selain menggunakan *kaskus* dan *twitter*?
2. Bagaimana proses membuat aplikasi *web* ini dari awal sampai akhir sesuai dengan keinginan *user*?
3. Bagaimana produk-produk *pc game* dapat tersusun dengan rapih agar konsumen dapat mencari dengan mudah?

### **1.4 Batasan Masalah**

Berdasarkan latar belakang tersebut diatas, maka permasalahan yang akan dijawab adalah sebagai berikut :

1. Aplikasi penjualan dan pemesanan secara *online* dibuat untuk membantu user dalam mempromosikan serta menjual barang dagangannya yaitu *pc game*.
2. Konsumen dapat melihat tatacara membeli pada konten *how to buy*.
3. Transaksi pembayaran dilakukan secara transfer via rekening atau bertemu secara langsung.
4. Tidak diberlakukannya sistem member, tetapi hanya nomor order.

### **1.5 Maksud dan Tujuan**

Tujuan dari penelitian ini adalah membuat sebuah aplikasi sistem informasi penjualan dan pemesanan *pc game* secara *online* berbasis *web* yang dapat diakses secara luas. Untuk memenuhi kebutuhan pembuatan aplikasi ini diperlukan :

1. Melakukan pengumpulan data sesuai dengan apa yang dibutuhkan.



2. Membuat laporan aplikasi penjualan dan pemesanan dengan menggunakan bahasa pemrograman berbasis *web*.
3. Menghasilkan sebuah aplikasi penjualan *pc game* yang bermanfaat dan dapat melancarkan *user* dalam menjalankan usahanya.

## **1.6 Metodologi Penelitian**

Metodologi yang digunakan dalam penulisan laporan kerja praktek ini adalah sebagai berikut :

a. Studi literatur

Pengumpulan data dengan cara mengumpulkan referensi yang menunjang pembuatan sistem.

b. Observasi

Mengamati dan terjun secara langsung ke dalam proses kerja yang dilakukan di dalam lingkungan perusahaan untuk mendapatkan gambaran yang jelas tentang objek yang diamati.

c. *Interview*

Pengumpulan data dengan cara melakukan tanya jawab secara langsung dengan pemilik usaha.

d. Metode Kepustakaan

Metode kepustakaan ini mengacu pada buku-buku pedoman yang dibutuhkan sebagai tambahan referensi peneliti, baik yang ada dipergustakaan ataupun pada literature-literatur lain.

## **1.7 Sistematika Penulisan**

Untuk memudahkan penyusunan yang sistematis, ringkas dan jelas maka penyusun akan menguraikan secara garis besar sistematik penulisan Laporan Tugas Akhir ini yang dibagi menjadi enam bab, yaitu:

### **BAB I : Pendahuluan**

Menjelaskan latar belakang masalah, identifikasi masalah, maksud dan tujuan, batasan masalah, metodologi penelitian, dan sistematika penulisan.

### **BAB II : Dasar Teori**

Membahas tentang teori-teori yang digunakan penulis untuk membangun sistem aplikasi penjualan dan pemesanan *pc game* secara *online* berbasis *web*.

### **BAB III : Gambaran Umum Usaha dan Analisis**

Membahas tentang awal mulanya sejarah terbangunnya usaha penjualan *pc game* oleh *go\_gaming*.

Menguraikan tentang analisis dari sistem yang telah ada sebelumnya dan sistem yang akan dibuat.

### **BAB IV : Perancangan Sistem**

Menguraikan tentang perancangan sistem yang mencakup pemodelan sistem dengan *unified modelling language* (UML) dalam menyelesaikan permasalahan yang ada.

### **BAB V : Implementasi**

Berisi apa saja yang terkait pada saat implementasi dan pengujian aplikasi.

## **BAB VI : Kesimpulan dan Saran**

Menguraikan tentang kesimpulan keseluruhan dari aplikasi yang telah dibuat dan saran-saran yang diperoleh dari hasil laporan.



## **BAB II**

### **DASAR TEORI**

#### **2.1 Dasar Teori**

##### **2.1.1 Aplikasi**

Aplikasi Komputer atau Aplikasi Software adalah program komputer yang ditulis dalam suatu bahasa pemrograman dan dipergunakan untuk menyelesaikan masalah tertentu. [8:url]

Pada awalnya aplikasi web dibangun dengan hanya menggunakan bahasa yang disebut *HTML (HyperText Markup Language)*. Pada perkembangan berikutnya, sejumlah skrip dan objek dikembangkan untuk memperluas kemampuan *HTML* seperti *PHP* dan *ASP* pada skrip dan *Applet* pada objek. Aplikasi *Web* dapat dibagi menjadi dua jenis yaitu aplikasi *web* statis dan dinamis.

*Web* statis dibentuk dengan menggunakan *HTML*. Kekurangan aplikasi seperti ini terletak pada keharusan untuk memelihara program secara terus menerus untuk mengikuti setiap perkembangan yang terjadi. Kelemahan ini diatasi oleh model aplikasi *web* dinamis. Pada aplikasi *web* dinamis, perubahan informasi dalam halaman *web* dilakukan tanpa perubahan program tetapi melalui perubahan data. Sebagai implementasi, aplikasi *web* dapat dikoneksikan ke basis data sehingga perubahan informasi dapat dilakukan oleh operator dan tidak menjadi tanggung jawab dari *webmaster*.

Arsitektur aplikasi *web* meliputi klien, *web server*, *middleware* dan basis data. Klien berinteraksi dengan *web server*. Secara internal, *web server* berkomunikasi dengan *middleware* dan *middleware* yang berkomunikasi dengan basis data. Contoh *middleware* adalah *PHP* dan *ASP*. Pada mekanisme aplikasi *web* dinamis, terjadi tambahan proses yaitu server menerjemahkan kode *PHP* menjadi kode *HTML*. Kode *PHP* yang diterjemahkan oleh mesin *PHP* yang akan diterima oleh klien.[7:buku].

### **2.1.2 HTML**

*HTML (Hyper Text Markup Language)* dalam arti umum sebuah web adalah sebuah dokumen, yang mana sebuah *HTML* itu sendiri merupakan bahasa yang menggunakan tanda-tanda tertentu (*Tag*) untuk menyatakan kode-kode yang harus diterjemahkan oleh *Browser*, agar halaman dapat di tampilkan secara benar.

*HTML* sendiri sebenarnya hanya dokumen biasa, hanya dokumen ini dapat dilakukan untuk pemformatan *text*, peletakan *object*, maupun *Hypertext* yakni *text* yang berfungsi ganda sebagai penghubung (*Link*) antara halaman satu dengan yang lain. Link inilah yang menjadi ciri sekaligus membentuk suatu *www* (Jaringan global antar *text* / halaman)

Bagaimana contoh sederhana bentuk sebuah halaman *HTML*? Sebuah halaman web yang baik, secara fisik akan terdiri dari sebuah *Header*, *Body*, *Sidebar* dan *Footer*. Untuk membentuk sebuah halaman *HTML* secara sederhana, kita memanfaatkan *Notepad* ataupun *tool* pendukung lainnya pada komputer anda. Setelah itu, buka *Notepad* atau *tool* pendukung lainnya dan tuliskan kode di bawah ini :  
[6:url]

```
<html>
<head>
<title>Belajar HTML</title>
</head>
<body>
<p>Hallo! aku sedang belajar web !</p>
</body>
</html>
```

### **2.1.3 XAMPP**

*XAMPP* adalah perangkat lunak bebas, yang mendukung banyak sistem operasi, merupakan kompilasi dari beberapa program. Fungsinya adalah sebagai server yang berdiri sendiri (*localhost*), yang terdiri atas program *Apache HTTP Server*, *MySQL database*, dan penerjemah bahasa yang ditulis dengan bahasa pemrograman *PHP* dan *Perl*. Nama *XAMPP* merupakan singkatan dari *X*(empat sistem operasi apapun), *Apache*, *MySQL*, *PHP* dan *Perl*. Program ini tersedia dalam *GNU General Public License* dan bebas, merupakan *web server* yang mudah digunakan yang dapat melayani tampilan halaman *web* yang dinamis. [4:url]

#### **2.1.3.1 Apache**

*Apache* merupakan aplikasi *web server*. Tugas utama *Apache* adalah menghasilkan halaman *web* yang benar kepada user berdasarkan kode *PHP* yang dituliskan oleh pembuat *web*. jika diperlukan juga berdasarkan kode *PHP* yang dituliskan, maka dapat saja suatu database diakses terlebih dahulu (misalnya dalam *MySQL*) untuk mendukung halaman *web* yang dihasilkan. [4:url]

#### **2.1.3.2 MySQL**

*MySQL* merupakan aplikasi *database server*. Perkembangannya disebut *SQL* yang merupakan kepanjangan dari *Structured Query Language*. *SQL* merupakan bahasa terstruktur yang digunakan untuk mengolah database. *MySQL* dapat digunakan untuk membuat dan mengelola database beserta isinya. Kita dapat memanfaatkan *MySQL* untuk menambahkan, mengubah, dan menghapus data yang berada dalam database. [4:url]

*SQL* adalah sebuah konsep pengoperasian *database*, terutama untuk pemilihan atau seleksi dan pemasukan data, yang memungkinkan pengoperasian data dikerjakan dengan mudah secara otomatis. Keandalan suatu sistem *database*(*DBMS*) dapat diketahui dari cara kerja optimizer-nya dalam melakukan proses perintah-perintah *SQL*, yang dibuat oleh *user* maupun program-program aplikasinya. Sebagai

*database server*, *MySQL* dapat dikatakan lebih unggul dibandingkan *database server* lainnya dalam *query data*. Hal ini terbukti untuk *query* yang dilakukan oleh *single user*, kecepatan *query MySQL* bisa sepuluh kali lebih cepat dari *PostgreSQL* dan lima kali lebih cepat dibandingkan *Interbase*.

### 2.1.3.3 PHP

Bahasa pemrograman *PHP(Personal Home Page)* adalah bahasa pemrograman yang bekerja dalam sebuah *web server*. *Script-script PHP* yang anda buat harus tersimpan dalam sebuah *server* dan dieksekusi atau diproses dalam *server* tersebut. Dengan menggunakan program *PHP*, sebuah *website* akan lebih interaktif dan dinamis. Sebagai media penulisan *script PHP*, anda juga dapat menggunakan beberapa program, di antaranya adalah *Notepad*, *Dreamweaver*, atau *PHP Expert Editor*. [4:buku]

### 2.1.3.4 Perl

*Perl* adalah bahasa pemrograman untuk segala keperluan, dikembangkan pertama kali oleh Larry Wall di mesin *Unix*. *Perl* dirilis pertama kali pada tanggal 18 Desember 1987 ditandai dengan keluarnya *Perl 1*. Pada versi-versi selanjutnya, *Perl* tersedia pula untuk berbagai sistem operasi varian *Unix* (*SunOS*, *Linux*, *BSD*, *HP-UX*), juga tersedia untuk sistem operasi seperti *DOS*, *Windows*, *PowerPC*, *BeOS*, *VMS*, *EBCDIC*, dan *PocketPC*. Dukungan terhadap pemrograman berbasis obyek *OOP(object oriented programming)* ditambahkan pada *Perl 5*, yang pertama kali dirilis pada tanggal 31 Juli 1993. Proyek pengembangan *Perl 6* dimulai pada tahun 2000, dan masih berlangsung hingga kini tanpa tanggal yang jelas kapan mau dirilis. Ini dikatakan sendiri oleh Larry Wall dalam satu pidatonya yang dikenal dengan seri *The State of the Onion*.

Dua di antara karakteristik utama *Perl* adalah penanganan teks dan berbagai jalan pintas untuk menyelesaikan persoalan-persoalan umum. *Perl* sangat populer digunakan dalam program-program *CGI(Common Gateway Interface)* dan berbagai protokol Internet lainnya. Seperti diketahui, *TCP/IP* sebagai basis bagi semua protokol Internet yang dikenal sekarang ini menggunakan format teks dalam komunikasi data. Seperti juga bahasa populer lainnya, *Perl* menerima banyak kritikan. Meski banyak di antaranya hanya berupa mitos, atau berlebih-lebihan, tapi terdapat juga sejumlah kritikan yang valid. Salah satunya adalah, sintaksnya susah dibaca, karena banyak menggunakan simbol-simbol yang bukan huruf dan angka. [4:url]



#### **2.1.4 PHPMYAdmin**

*PHPMYAdmin* adalah pengelolaan *database* dengan *MYSQL* yang dilakukan dengan mengetikkan baris-baris perintah yang sesuai (*command line*) untuk setiap maksud tertentu. Jika anda ingin membuat *database*, ketikkan baris perintah yang sesuai untuk membuat *database*. Jika kita ingin menghapus tabel, ketikkan baris perintah yang sesuai untuk menghapus tabel. Hal tersebut tentu cukup menyulitkan karena kita harus hafal dan mengetikkan perintahnya satu persatu. [7:url]

#### **2.1.5 CSS (Cascade Style Sheet)**

*CSS (Cascade Style Sheet)* digunakan para *web designer* untuk mengatur *style* elemen yang ada dalam halaman *web* mereka, mulai dari memformat text, sampai pada memformat layout. Tujuan dari penggunaan *CSS* ini adalah supaya diperoleh suatu kekonsistenan *style* pada elemen tertentu.

Keuntungan penggunaan *CSS*, anda (sebagai *webmaster*) akan lebih mudah untuk mengatur *style* elemen dalam halaman *web* anda. Sebagai contoh, misalnya untuk mengatur *style* elemen *heading*, anda menginginkan jenis *font* nya adalah Arial, 20 pixel dan berwarna merah. *Design CSS*, anda cukup menuliskan property dari elemen *heading* tersebut sekali saja, dan anda akan memperoleh hasil yang diinginkan. Bandingkan apabila tidak menggunakan *CSS*, anda akan mengatur *style* pada setiap elemen yang ada. [5:url]

#### **2.1.6 Web**

*Web* artinya Suatu sistem di internet yang memungkinkan siapapun agar bisa menyediakan informasi. Dengan menggunakan teknologi tersebut, informasi dapat diakses selama 24 jam dalam satu hari dan dikelola oleh mesin. Untuk mengakses informasi yang disediakan *web* ini, diperlukan berbagai perangkat lunak, yang disebut dengan *web browser*. [3:url]

*Website* adalah sebutan bagi sekelompok halaman *web(web page)*, dan umumnya merupakan bagian dari suatu nama domain (*domain name*), atau subdomain dalam *World wide web(www)* di internet. Jika anda sering menggunakan fasilitas internet dan mengunjungi *Yahoo, Google, Friendster, Facebook*, atau *Twitter*, maka nama-nama itu menunjukkan suatu domain di internet (*www.yahoo.com, www.google.co.id, dst*). Sedangkan *website* adalah halaman-halaman yang ditampilkan pada *domain-domain* tersebut. *Website* yang baik adalah *website* yang dari segi tampilan terlihat menarik, dari segi isi tidak asal-asalan, dan dari segi pengunjung cukup banyak peminatnya. [4:buku]

### **2.1.7 Database**

*Database*(basis data) merupakan kumpulan data yang saling berhubungan. Hubungan antar data dapat ditunjukkan dengan adanya *field/kolom* kunci dari tiap *file/tabel* yang ada. Dalam satu *file* atau *table* terdapat *record-record* yang sejenis, sama besar, sama bentuk, yang merupakan satu kumpulan entitas yang seragam. Satu *record* (umumnya digambarkan sebagai baris data) terdiri dari *field* yang saling berhubungan menunjukkan bahwa *field* tersebut dalam satu pengertian yang lengkap dan disimpan dalam satu *record*. [1:url]

Prinsip utama *Database* adalah pengaturan data dengan tujuan utama fleksibilitas dan kecepatan pada saat pengambilan data kembali. Adapun ciri-ciri basis data diantaranya adalah sebagai berikut :

1. Efisiensi meliputi kecepatan, ukuran, dan ketepatan
2. Data dalam jumlah besar.
3. Berbagi Pakai (dipakai bersama sama/*Sharebility*).

4. Mengurangi bahkan menghilangkan terjadinya duplikasi dan ketidakkonsistenan data.

### **2.1.8 Penjualan**

Aktivitas penjualan merupakan pendapatan utama perusahaan karena jika aktivitas penjualan produk maupun jasa tidak dikelola dengan baik maka secara langsung dapat merugikan perusahaan. Hal ini dapat disebabkan karena sasaran penjualan yang diharapkan tidak tercapai dan pendapatan pun akan berkurang.

Untuk lebih jelasnya dapat dilihat dari pengertian penjualan itu sendiri adalah sebagai berikut:

“Penjualan adalah pendapatan lazim dalam perusahaan dan merupakan jumlah kotor yang dibebankan kepada pelanggan atas barang dan jasa”. [2:buku]

“Penjualan artinya penjualan barang dagangan sebagai usaha pokok perusahaan yang biasanya dilakukan secara teratur”. [3:buku]

Berdasarkan pengertian di atas, maka dapat disimpulkan bahwa penjualan adalah persetujuan kedua belah pihak antara penjual dan pembeli, dimana penjual menawarkan suatu produk dengan harapan pembeli dapat menyerahkan sejumlah uang sebagai alat ukur produk tersebut sebesar harga jual yang telah disepakati.

### **2.1.9 Metode Pembayaran Dengan Paypal**

Paypal adalah perusahaan pihak ke tiga jasa keuangan yang digunakan sebagai sarana pembayaran *online* yang lahir pada bulan Oktober di tahun 1999, diawali dengan layanan transaksi sederhana via *online*.

Paypal dengan cepat naik daun, terlebih lagi dengan melambungnya *eBay* sebagai pasar global *online* yang mendorong kebutuhan akan rekening paypal sebagai media transaksi.

Banyak keuntungan yang didapat dari paypal. Inilah yang menjadi alasan mengapa paypal demikian populer dan banyak orang menggunakan paypal. Paypal merupakan pilihan yang tepat untuk mengirim dan menerima pembayaran *online* tanpa harus mengungkapkan informasi keuangan anda.

Dari sekian banyak jasa keuangan, paypal memiliki fitur-fitur keunggulan, antara lain :

- a. Metode pembayaran aman untuk berbelanja online.

Berbagai toko online seperti *eBay* dan ribuan toko online lainnya menerima pembayaran via paypal. Paypal aman, karena informasi konfidensial perihal data keuangan anda dijamin kerahasiannya.

- b. Metode transfer uang dengan cepat dan mudah.

Paypal memungkinkan anda melakukan pembayaran atau pembelian dan mengirim uang dari 190 negara dan wilayah. Yang anda butuhkan adalah alamat email penerima. Penerima tidak perlu rekening paypal, mereka dapat meregistrasi untuk menerima pembayaran anda.

- c. Diterima diseluruh dunia.

Paypal diterima oleh ribuan bisnis di seluruh dunia dan merupakan metode pembayaran populer yang digunakan oleh jutaan toko online di berbagai dunia, terlebih lagi untuk *eBay*.

- d. Registrasi cepat dan mudah digunakan.

Waktu yang dibutuhkan untuk mendaftar account paypal hanya beberapa menit. Setelah anda telah mendaftar, anda dapat mengirim pembayaran dalam hitungan menit.

Persyaratan untuk memiliki akun paypal yang sah, berikut beberapa persyaratannya :

- a. Minimal berusia 18 tahun.
- b. Memiliki alamat email.
- c. Memiliki kartu kredit.

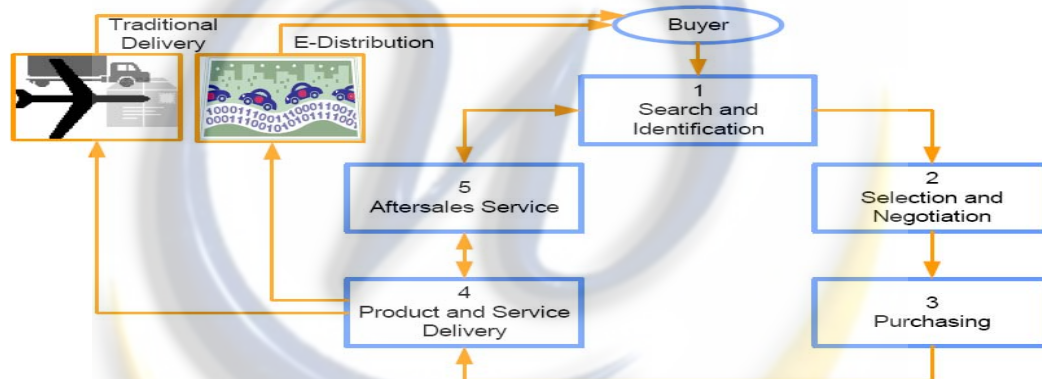
Bagi mereka yang tidak memiliki kartu kredit, mungkin dapat mulai melirik sarana lain untuk tetap bisa bertransaksi dengan menggunakan paypal. Salah satunya adalah VCC atau virtual Credit Card. Kebanyakan orang menggunakan VCC untuk bertransaksi pada situs online ‘asing’ dan berbagai metode pembayaran tahunan yang masih sulit untuk di estimasi. Jadi, akan lebih baik menggunakan virtual credit card sebagai penyangga dan membatasi kerugian finansial, seandainya ada kekurangan. VCC membuat keberadaan orang yang bertransaksi dibalikinya yang seharusnya tetap transparan menjadi tidak terlihat, termasuk pula alamat dari orang yang bersangkutan. [6:buku]

#### **2.1.10 E-Commerce**

Secara umum *Electronic Commerce (ECommerce)* dapat didefinisikan sebagai segala bentuk transaksi perdagangan/perniagaan barang atau jasa (*trade of goods and service*) dengan menggunakan media elektronik. Sedangkan *EBusiness* memiliki definisi yang lebih luas daripada *ECommerce*. *EBusiness* mencakup segala aspek yang ada di dalam *ECommerce*, plus bagaimana cara menjalankan bisnis (proses internal di dalam organisasi). Di dalam *ECommerce*, para pihak yang melakukan kegiatan perdagangan/perniagaan hanya berhubungan melalui suatu jaringan publik (*public network*) yang dalam perkembangan terakhir menggunakan media internet.[8:buku]

Secara umum, *Ecommerce* dapat diklasifikasikan menjadi dua jenis, yaitu *Business to Business (B2B)* dan *Business to Consumer (B2C)*. B2B adalah sistem

komunikasi bisnis *online* antar pelaku bisnis, sedangkan B2C merupakan mekanisme toko *online* (*electronic shopping mall*), yaitu transaksi antara pihak yang menawarkan barang atau jasa dengan pihak pembeli dengan menggunakan media internet. Dalam B2B, pada umumnya, transaksi dilakukan oleh para *trading partners* yang sudah saling kenal dengan format data yang telah disepakati bersama. Sedangkan dalam B2C sifatnya terbuka untuk publik, sehingga setiap individu dapat mengaksesnya melalui suatu *web server*. Dalam modul ini, untuk selanjutnya yang dibahas adalah B2C. Model transaksi *Ecommerce* dapat digambarkan sebagai proses seperti yang terlihat dalam Gambar 1.



Gambar 2.1 Model Transaksi *eCommerce*

Pada Gambar 1 terlihat calon pembeli (*buyer*) melihat atau mencari barang atau jasa yang dia inginkan yang ditawarkan di dalam *website* perusahaan dan mencari data atau informasi tertentu yang dibutuhkan sehubungan dengan proses transaksi jual beli yang akan dilakukan. Jika tertarik dengan produk atau jasa yang ditawarkan, *buyer* dapat melakukan transaksi perdagangan dengan cara melakukan pemesanan secara elektronik (*online orders*), yaitu dengan menggunakan perangkat komputer dan jaringan internet. Berdasarkan pesanan tersebut, perusahaan akan mendistribusikan barangnya kepada *buyer*. Selanjutnya, melalui internet dapat pula dilakukan aktivitas pasca pembelian, yaitu pelayanan purnajual (*electronic customer support*). Proses ini



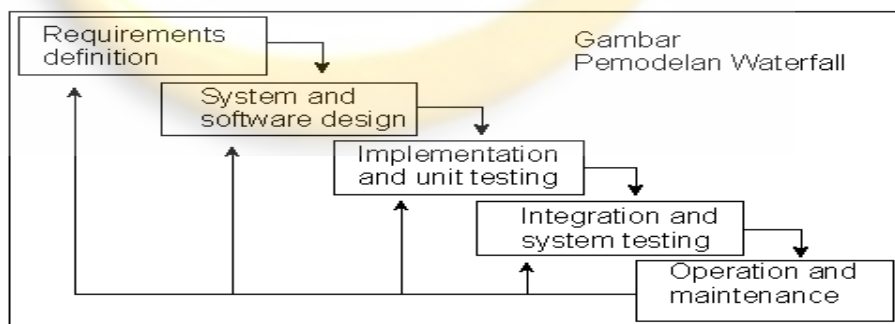
dapat dilakukan melalui jalur konvensional, seperti telepon, ataupun jalur internet, seperti *email*, *teleconference*, *chatting*, dan lainlain. Dari interaksi tersebut diharapkan *customers* dapat datang kembali dan melakukan pembelian produk atau jasa di kemudian hari (*followon sales*).[9:buku]

## 2.2 Pengembangan Sistem Informasi

### 2.2.1 Metodologi Pengembangan Sistem Informasi

#### 2.2.1.1 Model Waterfall

*Waterfall Processs Model* atau *Linear Sequential Model* merupakan model pengembangan perangkat lunak yang muncul pada tahun 1970, sehingga sering dianggap kuno, tetapi merupakan model yang paling banyak dipakai didalam *software engineering*. Model ini melakukan pendekatan secara sistematis dan urut mulai dari *level* kebutuhan sistem lalu menuju ke tahap analisis, desain, *coding*, *testing/verification*, dan *maintenance*. Disebut dengan *waterfall* karena tahap demi tahap yang dilalui harus menunggu selesainya tahap sebelumnya dan berjalan berurutan. Sebagai contoh tahap desain harus menunggu selesainya tahap sebelumnya yaitu tahap *requirement*. Secara umum tahapan pada model *waterfall* dapat dilihat pada gambar berikut :



**Gambar 2.2**

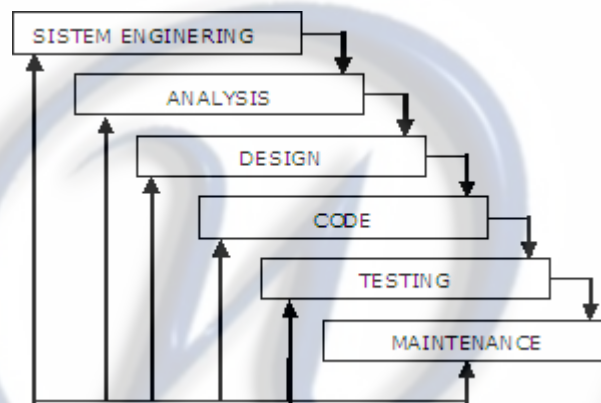
Pemodelan *Waterfall*

(Pressman, Roger S. (1992), *Software Engineering : A Practioner's Approach.*)



Gambar di atas adalah tahapan umum dari model proses ini. Model rekayasa piranti lunak yang diuraikan salah satunya adalah waterfall model. Model ini memberikan pendekatan-pendekatan sistematis dan berurutan bagi pengembangan piranti lunak.

Berikut adalah gambar pengembangan sistem perangkat lunak dengan proses SDLC (*System Development Life Cycle*) dengan model *waterfall*. [5:buku]



**Gambar 2.3**

Model *Waterfall* 6 Tahap

(Pressman, Roger S. (1992), *Software Engineering : A Practioner's Approach.*)

Penjelasan dari tahap-tahap model *waterfall* adalah sebagai berikut:

### **1. Perancangan Sistem (*System Engineering*)**

Perancangan sistem sangat diperlukan, karena piranti lunak biasanya merupakan bagian dari suatu sistem yang lebih besar. Pembuatan sebuah piranti lunak dapat dimulai dengan melihat dan mencari apa yang dibutuhkan oleh sistem. Dari kebutuhan sistem tersebut akan diterapkan kedalam piranti lunak yang dibuat.

## **2. Analisis Kebutuhan Piranti Lunak (*Software Requirement Analysis*)**

Merupakan proses pengumpulan kebutuhan piranti lunak. Untuk memahami dasar dari program yang akan dibuat, seorang analisis harus mengetahui ruang lingkup informasi, fungsi-fungsi yang dibutuhkan, kemampuan kinerja yang ingin dihasilkan dan perancangan antarmuka pemakai piranti lunak tersebut.

## **3. Perancangan (*Design*)**

Perancangan piranti lunak merupakan proses bertahap yang memfokuskan pada empat bagian penting, yaitu: Struktur data, arsitektur piranti lunak, detail prosedur, dan karakteristik antar muka pemakai.

## **4. Pengkodean (*Coding*)**

Pengkodean piranti lunak merupakan proses penulisan bahasa program agar piranti lunak tersebut dapat dijalankan oleh mesin.

## **5. Pengujian (*Testing*)**

Proses ini akan menguji kode program yang telah dibuat dengan memfokuskan pada bagian dalam piranti lunak. Tujuannya untuk memastikan bahwa semua pernyataan telah diuji dan memastikan juga bahwa input yang digunakan akan menghasilkan *output* yang sesuai. Pada tahap ini pengujian ini dibagi menjadi dua bagian, pengujian *internal* dan pengujian *eksternal*. Pengujian internal bertujuan menggambarkan bahwa semua statement sudah dilakukan pengujian, sedangkan pengujian *eksternal* bertujuan untuk menemukan kesalahan serta memastikan *output* yang dihasilkan sesuai dengan yang diharapkan.

## **6. Pemeliharaan (*Maintenance*)**

Proses ini dilakukan setelah piranti lunak telah digunakan oleh pemakai atau konsumen. Perubahan akan dilakukan jika terdapat kesalahan, oleh karena itu piranti lunak harus disesuaikan lagi untuk menampung perubahan kebutuhan yang diinginkan konsumen.

### **2.2.1.2 Model *Sekuensial Linear***

Model sekuensial linier untuk *software engineering*, sering disebut juga dengan siklus kehidupan klasik atau model air terjun. Model ini mengusulkan sebuah pendekatan kepada perkembangan *software* yang sistematis dan sekuensial yang mulai pada tingkat dan kemajuan sistem pada seluruh analisis, desain, kode, pengujian, dan pemeliharaan. Dimodelkan setelah siklus rekayasa konvensional, model sekuensial linier melingkupi aktivitas-aktivitas sebagai berikut : [2:url]

#### **1. Rekayasa dan pemodelan sistem/informasi**

Karena sistem merupakan bagian dari sebuah sistem yang lebih besar, kerja dimulai dengan membangun syarat dari semua elemen sistem dan mengalokasikan beberapa subset dari kebutuhan ke *software* tersebut. Pandangan sistem ini penting ketika *software* harus berhubungan dengan elemen-elemen yang lain seperti *software*, manusia, dan *database*. Rekayasa dan analisis sistem menyangkut pengumpulan kebutuhan pada tingkat sistem dengan sejumlah kecil analisis serta desain tingkat puncak. Rekayasa informasi mencakup juga pengumpulan kebutuhan pada tingkat bisnis strategis dan tingkat area bisnis.

#### **2. Analisis kebutuhan *Software***

Proses pengumpulan kebutuhan diintensifkan dan difokuskan, khususnya pada *software*. Untuk memahami sifat program yang dibangun, analis harus memahami *domain* informasi, tingkah laku, unjuk kerja, dan interface yang diperlukan.

Kebutuhan baik untuk sistem maupun *software* didokumentasikan dan dilihat lagi dengan pelanggan.

### **3. Desain**

Desain *software* sebenarnya adalah proses multi langkah yang berfokus pada empat atribut sebuah program yang berbeda: struktur data, arsitektur *software*, representasi *interface*, dan detail(algoritma) prosedural.

Proses desain menterjemahkan syarat/kebutuhan ke dalam sebuah representasi *software* yang dapat diperkirakan demi kualitas sebelum dimulai pemunculan kode. Sebagaimana persyaratan, desain didokumentasikan dan menjadi bagian dari konfigurasi *software*.

### **4. Generasi Kode**

Desain harus diterjemahkan kedalam bentuk mesin yang bisa dibaca. Langkah pembuatan kode melakukan tugas ini. Jika desain dilakukan dengan cara yang lengkap, pembuatan kode dapat diselesaikan secara mekanis.

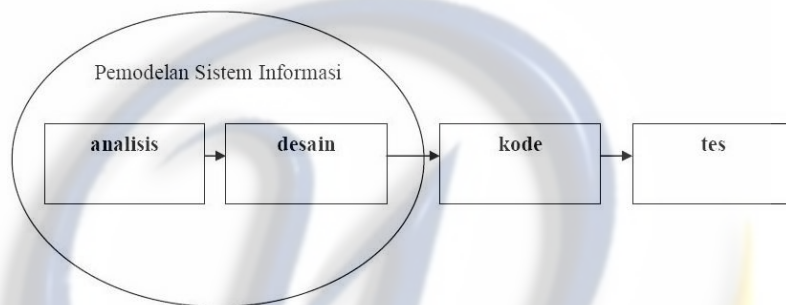
### **5. Pengujian**

Sekali program dibuat, pengujian program dimulai. Proses pengujian berfokus pada logika internal *software*, memastikan bahwa semua pernyataan sudah diuji, dan pada eksternal fungsional, yaitu mengarahkan pengujian untuk menemukan kesalahan - kesalahan dan memastikan bahwa *input* yang dibatasi akan memberikan hasil aktual yang sesuai dengan hasil yang dibutuhkan.

### **6. Pemeliharaan**

*Software* akan mengalami perubahan setelah disampaikan kepada pelanggan (perkecualian yang mungkin adalah *software* yang dilekatkan). Perubahan akan

terjadi karena kesalahan-kesalahan ditentukan, karena *software* harus disesuaikan untuk mengakomodasi perubahan-perubahan di dalam lingkungan eksternalnya (contohnya perubahan yang dibutuhkan sebagai akibat dari perangkat peripheral atau sistem operasi yang baru), atau karena pelanggan membutuhkan perkembangan fungsional atau unjuk kerja. Pemeliharaan *software* mengaplikasikan lagi setiap fase program sebelumnya dan tidak membuat yang baru lagi.



**Gambar 2.4**

Pemodelan Sekuensial Linear

<http://djogjakarta.blogdetik.com/hr/review-metodologi-pengembangan-perangkat-lunak/>

Masalah yang kadang terjadi ketika model sekuensial linier diaplikasikan adalah :

1. Jarang sekali proyek nyata mengikuti aliran sekuensial yang dianjurkan oleh model. Meskipun model linier bisa mengakomodasi iterasi, model ini melakukannya dengan cara tidak langsung. Sebagai hasilnya, perubahan-perubahan dapat menyebabkan keraguan pada saat tim proyek berjalan.
2. Kadang-kadang sulit bagi pelanggan untuk menyatakan semua kebutuhannya secara eksplisit. Model linier sekuensial memerlukan hal ini dan mengalami kesulitan untuk mengakomodasi ketidakpastian natural yang ada pada bagian awal beberapa proyek.

3. Pelanggan harus bersifat sabar. Sebuah versi kerja dari program-program kerja itu tidak akan diperoleh sampai akhir waktu proyek dilalui. Sebuah kesalahan besar, jika tidak terdeteksi sampai program yang bekerja tersebut dikaji ulang, bisa menjadi petaka.
4. Pengembang sering melakukan penundaan yang tidak perlu. Sifat alami dari siklus kehidupan klasik membawa kepada *blocking state* di mana banyak anggota tim proyek harus menunggu tim yang lain untuk melengkapi tugas yang saling memiliki ketergantungan. *Blocking state* cenderung menjadi lebih lazim pada awal dan akhir sebuah proses sekuensial linier

### **2.2.1.3 Model Prototype**

*Prototyping paradigma* dimulai dengan pengumpulan kebutuhan. Pengembang dan pelanggan bertemu dan mendefinisikan obyektif keseluruhan dari software, mengidentifikasi segala kebutuhan yang diketahui, dan area garis besar dimana definisi lebih jauh merupakan keharusan kemudian dilakukan “perancangan kilat”. Perancangan kilat berfokus pada penyajian dari aspek - aspek software tersebut yang akan nampak bagi pelanggan atau pemakai (contohnya pendekatan *input* dan format *output*). Perancangan kilat membawa kepada konstruksi sebuah prototipe. Prototipe tersebut dievaluasi oleh pelanggan/pemakai dan dipakai untuk menyaring kebutuhan pengembangan *software*. Iterasi terjadi pada saat prototipe disetel untuk memenuhi kebutuhan pelanggan, dan pada saat yang sama memungkinkan pengembang untuk secara lebih baik memahami apa yang harus dilakukannya. [2:url]



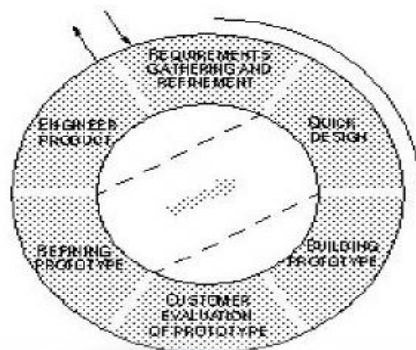


figure 3. The prototyping paradigm of software development

### Gambar 2.5

#### Pemodelan Prototipe

(<http://djogjakarta.blogdetik.com/hr/review-metodologi-pengembangan-perangkat-lunak/>)

Secara ideal prototipe berfungsi sebagai sebuah mekanisme untuk mengidentifikasi kebutuhan *software*. Bila prototipe yang sedang bekerja dibangun, pengembang harus mempergunakan fragmen-fragmen program yang ada atau mengaplikasikan alat-alat bantu (contohnya *report generator*, *windowmanager*, dll) yang memungkinkan program yang bekerja untuk dimunculkan secara cepat.

Prototipe bisa juga menjadi masalah karena alasan sebagai berikut:

1. Pelanggan melihat apa yang tampak sebagai versi *software* yang bekerja tanpa melihat bahwa prototipe itu dijalin bersama-sama “dengan permen karet dan *baling wire*“, tanpa melihat bahwa di dalam untuk membuatnya bekerja, kita belum menyantumkan kualitas *software* secara keseluruhan atau kemampuan pemeliharaan untuk jangka waktu yang panjang. Ketika diberi informasi bahwa produk harus dibangun lagi agar tingkat kualitas yang tinggi bisa dijaga, pelanggan akan meneriakkan kecurangan dan permintaan agar dipakai



“beberapa campuran” untuk membuat prototipe menjadi sebuah produk yang bekerja yang lebih sering terjadi, sehingga manajemen pengembangan *software* menjadi penuh dengan belas kasihan.

2. Pengembang sering membuat kompromi-kompromi implementasi untuk membuat prototipe bekerja dengan cepat. Sistem operasi atau bahasa pemrograman yang tidak sesuai bisa dipakai secara sederhana karena mungkin diperoleh dan dikenal, algoritma yang tidak efisien secara sederhana bisa diimplementasikan untuk mendemonstrasikan kemampuan. Setelah selang waktu tertentu, pengembang mungkin mengenali pilihan-pilihan tersebut dan melupakan semua alasan mengapa mereka tidak cocok. Pilihan yang kurang ideal telah menjadi bagian integral dari sebuah sistem.

Meskipun berbagai masalah bisa terjadi, prototipe bisa menjadi paradigma yang efektif bagi *Software Engineering*. Kuncinya adalah mendefinisikan aturan main pada saat awal, yaitu pelanggan dan pengembang keduanya harus setuju bahwa prototipe dibangun untuk berfungsi sebagai mekanisme pendefinisian kebutuhan. Prototipe kemudian disingkirkan(paling tidak sebagian), dan *software* aktual direalisasikan dengan tertuju kepada kualitas dan kemampuan pemeliharaan.

#### **2.2.1.4 Model *Rapid Application Development***

*Rapid Application Development*(RAD) adalah sebuah model proses perkembangan *software* sekuensial linier yang menekankan siklus perkembangan yang sangat pendek. Model RAD ini merupakan sebuah adaptasi “kecepatan tinggi” dari model sekuensial linier di mana perkembangan cepat dicapai dengan menggunakan pendekatan konstruksi berbasis komponen. Jika kebutuhan dipahami dengan baik, proses RAD memungkinkan tim pengembangan menciptakan “sistem fungsional yang utuh” dalam periode waktu yang sangat pendek (kira-kira 60 sampai

90 hari). Karena dipakai terutama pada aplikasi sistem konstruksi, pendekatan *RAD* melingkupi fase - fase sebagai berikut : [2:url]

### **1. *Bussiness modeling***

Aliran informasi di antara fungsi-fungsi bisnis dimodelkan dengan suatu cara untuk menjawab pertanyaan-pertanyaan berikut : informasi apa yang mengendalikan proses bisnis? Informasi apa yang di munculkan? Siapa yang memunculkanya? Ke mana informasi itu pergi? Siapa yang memprosesnya?

### **2. *Data modeling***

Aliran informasi yang didefinisikan sebagai bagian dari fase bussiness modelling disaring ke dalam serangkaian objek data yang dibutuhkan untuk menopang bisnis tersebut. Karakteristik (disebut atribut) masing-masing objek diidentifikasi dan hubungan antara objek - objek tersebut didefinisikan.

### **3. *Prosess modeling***

Aliran informasi yang didefinisikan di dalam fase data modeling ditransformasikan untuk mencapai aliran informasi yang perlu bagi implementasi sebuah fungsi bisnis. Gambaran pemrosesan diciptakan untuk menambah, memodifikasi, menghapus, atau mendapatkan kembali sebuah objek data.

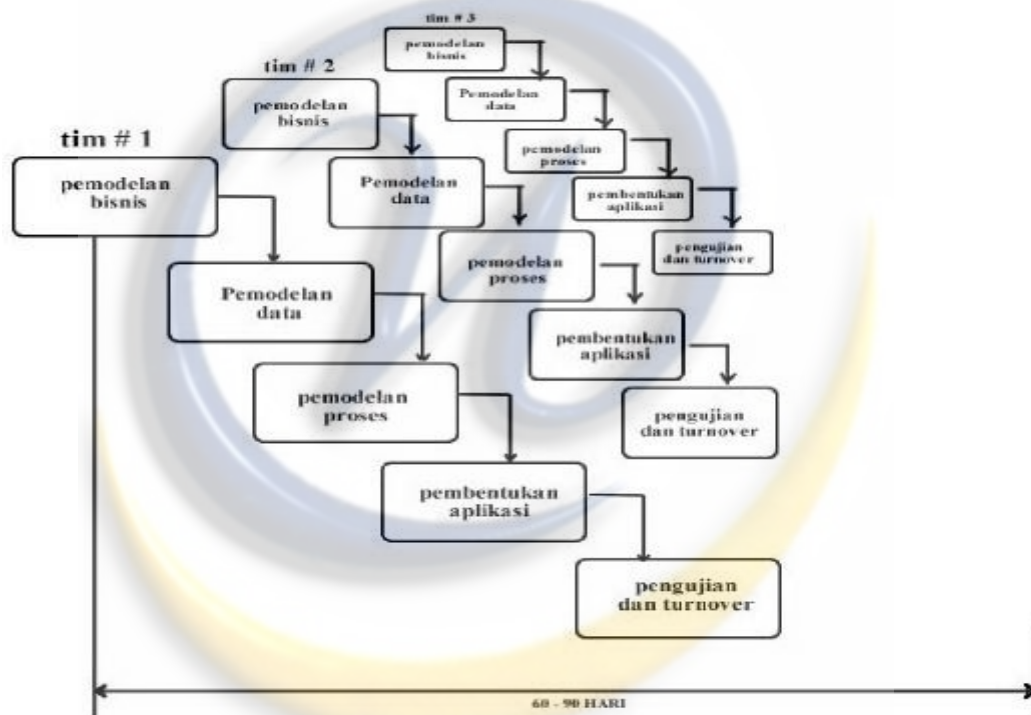
### **4. *Aplication generation***

*RAD* mengasumsikan pemakaian teknik generasi ke empat. Selain menciptakan perangkat lunak dengan menggunakan bahasa pemrograman generasi ketiga yang konvensional, *RAD* lebih banyak memproses kerja untuk memakai lagi komponen program yang ada (pada saat memungkinkan) atau menciptakan

komponen yang bisa dipakai lagi (bila perlu). Pada semua kasus, alat-alat bantu otomatis dipakai untuk memfasilitasi konstruksi perangkat lunak.

### 5. *Testing and turnover*

Karena proses *RAD* menekankan pada pemakaian kembali, banyak komponen program telah diuji. Hal ini mengurangi keseluruhan waktu pengujian. Tetapi komponen baru harus di uji dan semua *interface* harus dilatih secara penuh.



**Gambar 2.5**

Pemodelan *RAD*

(<http://djogjakarta.blogdetik.com/hr/review-metodologi-pengembangan-perangkat-lunak/>)

**Kekurangan model RAD adalah :**

1. Bagi proyek yang besar tetapi berskala, *RAD* memerlukan sumber daya manusia yang memadai untuk menciptakan jumlah tim *RAD* yang baik.
2. *RAD* menuntut pengembangan dan pelanggan memiliki komitmen di dalam aktivitas *rapid-fire* yang diperlukan untuk melengkapi sebuah sistem, di dalam kerangka waktu yang sangat diperpendek. Jika komitmen tersebut tidak ada, proyek *RAD* akan gagal.

**2.2.1.5 Model Spiral**

Model spiral (*spiral model*) adalah model proses *software* yang evolusioner yang merangkai sifat iteratif dari prototipe dengan cara kontrol dan aspek sistematis dari model sekuensial linier. Model ini berpotensi untuk pengembangan versi pertambahan *software* secara cepat. Di dalam model spiral, *software* dikembangkan di dalam suatu deretan pertambahan. Selama awal iterasi, rilis inkremental bisa merupakan sebuah model atau prototipe kertas. Selama iterasi berikutnya, sedikit demi sedikit dihasilkan versi sistem rekayasa yang lebih lengkap.

Model spiral dibagi menjadi sejumlah aktifitas kerangka kerja, disebut juga wilayah tugas, di antara tiga sampai enam wilayah tugas, yaitu : [2:url]

**1. Komunikasi Pelanggan**

Tugas-tugas yang dibutuhkan untuk membangun komunikasi yang efektif di antara pengembangan dan pelanggan.

**2. Perencanaan**

Tugas-tugas yang dibutuhkan untuk mendefinisikan sumber-sumber daya, ketepatan waktu, dan proyek informasi lain yang berhubungan.

### 3. Analisis Risiko

Tugas-tugas yang dibutuhkan untuk menaksir risiko-risiko, baik manajemen maupun teknis.

### 4. Perekayasaan

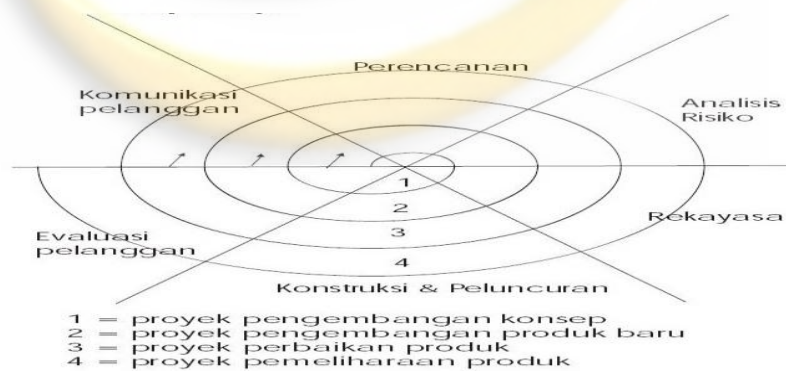
Tugas-tugas yang dibutuhkan untuk membangun satu atau lebih representasi dari aplikasi tersebut.

### 5. Konstruksi dan peluncuran

Tugas-tugas yang dibutuhkan untuk mengkonstruksi, menguji, memasang(*instal*) dan memberikan pelayanan kepada pemakai(contohnya pelatihan dan dokumentasi).

### 6. Evaluasi pelanggan

Tugas-tugas yang dibutuhkan untuk memperoleh umpan balik dari pelanggan dengan didasarkan pada evaluasi representasi *software*, yang dibuat selama masa perekayasaan, dan diimplementasikan selama masa pemasangan.



**Gambar 2.6**

Pemodelan Spiral

(<http://djogjakarta.blogdetik.com/hr/review-metodologi-pengembangan-perangkat-lunak/>)

Kekurangan model spiral adalah sulitnya untuk meyakinkan konsumen(khususnya dalam situasi kontrak) bahwa pendekatan evolusioner bisa dikontrol. Model spiral memerlukan keahlian penaksiran risiko yang masuk akal, dan sangat bertumpu pada keahlian ini untuk mencapai keberhasilan. Jika risiko mayor tidak ditemukan dan diatur, pasti akan terjadi masalah. Akhirnya model itu sendiri masih baru dan belum dipergunakan secara luas seperti paradigma sekuensial dan prototype.

#### **2.2.1.6 Model Yang Digunakan**

Model yang digunakan adalah model waterfall, karena terbilang sangat populer. Mengapa model ini sangat populer? Selain karena pengaplikasian menggunakan model ini mudah, kelebihan dari model ini adalah ketika semua kebutuhan sistem dapat didefinisikan secara utuh, eksplisit, dan benar di awal project, maka sistem dapat berjalan dengan baik dan tanpa masalah. Meskipun seringkali kebutuhan sistem tidak dapat didefinisikan secara eksplisit yang diinginkan, tetapi paling tidak, masalah pada kebutuhan sistem di awal *project* lebih ekonomis dalam hal uang (lebih murah), usaha, dan waktu yang terbuang lebih sedikit jika dibandingkan problem yang muncul pada tahap-tahap selanjutnya.

Meskipun demikian, karena model ini melakukan pendekatan secara urut/*sequential*, maka ketika suatu tahap terhambat, tahap selanjutnya tidak dapat dikerjakan dengan baik dan itu menjadi salah satu kekurangan dari model ini. Selain itu, ada beberapa kekurangan pengaplikasian model ini, antara lain adalah sebagai berikut:

1. Ketika masalah muncul, maka proses berhenti, karena tidak dapat menuju ke tahapan selanjutnya. Bahkan jika kemungkinan masalah tersebut muncul



akibat kesalahan dari tahapan sebelumnya, maka proses harus membenahi tahapan sebelumnya agar problem ini tidak muncul. Hal-hal seperti ini yang dapat membuang waktu pengerjaan sistem.

2. Karena pendekatannya secara *sequential*, maka setiap tahap harus menunggu hasil dari tahap sebelumnya. Hal itu tentu membuang waktu yang cukup lama, artinya bagian lain tidak dapat mengerjakan hal lain selain hanya menunggu hasil dari tahap sebelumnya. Oleh karena itu, seringkali model ini berlangsung lama pengerjaannya.

Pada setiap tahap proses tentunya dikerjakan sesuai spesialisasinya masing-masing. Oleh karena itu, ketika tahap tersebut sudah tidak dikerjakan, maka sumber dayanya juga tidak terpakai lagi. Oleh karena itu, seringkali pada model proses ini dibutuhkan seseorang yang *multi-skilled*, sehingga minimal dapat membantu pengerjaan untuk tahapan berikutnya.

### 2.3 Konsep Objek Berorientasi

Dalam konsep berorientasi objek, abstraksi dilakukan pada *level* objek. Objek adalah representasi dunia nyata kedalam bahasa pemrograman. Objek bisa dalam bentuk benda nyata seperti manusia, hewan, kendaraan. Di dalam pemrograman berorientasi objek, kode dan data yang saling terkait dibungkus dalam satu entitas.

Ada 3 prinsip utama berorientasi objek, yaitu:

#### a. Enkapsulasi

Enkapsulasi adalah mekanisme penggabungan kode dan data yang dimanipulasi oleh kode tersebut. Mekanisme ini juga menjaga kode dan data tetap aman dari *interferensi* dan penggunaan oleh pihak yang tidak berhak.



**b. Pewarisan ( *inheritance* )**

*Inheritance* adalah proses pewarisan data dan *method* dari suatu kelas kepada kelas yang lain. Pewarisan ini bersifat menyeluruh, sehingga semua data dan *method* yang dimiliki oleh kelas asalnya akan diturunkan kepada kelas yang baru. Pewarisan ini adalah konsep untuk meningkatkan guna ulang (*reuse*) perangkat lunak.

**c. *Polymorphism***

*Polymorphism* (dari bahasa Yunani yang berarti banyak rupa), adalah mekanisme yang memungkinkan suatu modul memiliki nama yang sama, namun memiliki *behavior* (tingkah laku) yang berbeda, sehingga implementasi dalam modulnya pun berbeda.

Bahasa pemrograman yang mendukung OOP antara lain :

- a. Visual Foxpro
- b. Java
- c. C++
- d. Pascal (bahasa pemrograman)
- e. Visual Basic.NET
- f. SIMULA
- g. Smalltalk
- h. Ruby
- i. Python
- j. PHP
- k. C#
- l. Delphi

- m. Eiffel
- n. Perl
- o. Adobe Flash AS 3.0

### **2.3.1 Bahasa Pemrograman Yang digunakan**

*PHP* Untuk membantu penulis dalam mengerjakan Perangkat Lunak Pengarsipan Data *Client* ini, penulis memakai bahasa pemrograman *PHP* dengan menggunakan *Microsoft MySQL* sebagai media penyimpanan datanya.

*PHP: Hypertext Preprocessor* adalah bahasa skrip yang dapat ditanamkan atau disisipkan ke dalam *HTML*. *PHP* banyak dipakai untuk memrogram situs *web* dinamis. *PHP* dapat digunakan untuk membangun sebuah *CMS*.

Beberapa kelebihan *PHP* dari bahasa pemrograman *web*, antara lain:

1. Bahasa pemrograman *PHP* adalah sebuah bahasa *script* yang tidak melakukan sebuah kompilasi dalam penggunaannya.
2. *Web Server* yang mendukung *PHP* dapat ditemukan dimana-mana dari mulai *apache*, *2S*, *Lighttpd*, hingga *Xitami* dengan konfigurasi yang relatif mudah.
3. Dalam sisi pengembangan lebih mudah, karena banyaknya milis-milis dan *developer* yang siap membantu dalam pengembangan.
4. Dalam sisi pemahaman, *PHP* adalah bahasa scripting yang paling mudah karena memiliki referensi yang banyak.

*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.

## 2.4 UML(Unified Modelling Language)

*UML* adalah keluarga notasi grafis yang didukung oleh *meta-model* tunggal, yang membantu pendeskripsian dan desain sistem perangkat lunak, khususnya sistem yang dibangun menggunakan pemrograman berorientasi objek. Definisi ini merupakan definisi yang sederhana. Pada kenyataannya, pendapat orang-orang tentang *UML* berbeda satu sama lain. Hal ini dikarenakan oleh sejarahnya sendiri dan oleh perbedaan persepsi tentang apa yang membuat sebuah proses rancangbangun perangkat lunak *efektif*.

Bahasa pemodelan grafis telah ada di *industry* perangkat lunak sejak lama. Pemicu utama di balik semuanya adalah bahwa bahasa pemrograman berada pada tingkat abstraksi yang tidak terlalu tinggi untuk memfasilitasi diskusi tentang desain.

Meskipun faktanya adalah bahasa pemodelan grafis telah ada sejak lama, masih terdapat banyak pertentangan di dalam *industry* perangkat lunak tentang perannya. Pertentangan-pertentangan ini berpengaruh langsung pada bagaimana orang-orang memandang peran *UML* itu sendiri. (Martin Fowler, 2003). [6:buku]

*UML* dideskripsikan oleh beberapa diagram, diantaranya :

### 1. Use Case Diagram

*Use case diagram* digunakan untuk menggambarkan sistem dari sudut pandang pengguna sistem tersebut (*user*), sehingga pembuatan *use case diagram* lebih dititikberatkan pada fungsionalitas yang ada pada sistem, bukan berdasarkan alur atau urutan kejadian. Sebuah *use case diagram* merepresentasikan sebuah interaksi antara aktor dengan sistem.

Komponen-komponen yang terlibat dalam *use case diagram* :

a. Aktor

Pada dasarnya aktor bukanlah bagian dari *use case diagram*, namun untuk dapat terciptanya suatu *use case diagram* diperlukan aktor, dimana aktor tersebut mempresentasikan seseorang atau sesuatu (seperti perangkat atau sistem lain) yang berinteraksi dengan sistem yang dibuat. Sebuah aktor mungkin hanya memberikan informasi inputan pada sistem, hanya menerima informasi dari sistem atau keduanya menerima dan memberi informasi pada sistem. Aktor hanya berinteraksi dengan *use case*, tetapi tidak memiliki kontrol atau *use case*. Aktor digambarkan dengan *stick man*.

b. *Use case*

Gambaran fungsionalitas dari sistem, sehingga pengguna sistem paham dan mengerti kegunaan sistem yang akan dibangun. Ada beberapa relasi yang terdapat pada *use case diagram* :

1. *Association*, menghubungkan *link* antar *element*;
2. *Generalization*, disebut juga *inheritance* (pewarisan), sebuah elemen dapat merupakan spesialisasi dari elemen lainnya;
3. *Dependency*, sebuah elemen bergantung dalam beberapa cara ke elemen lainnya.
4. *Aggregation*, bentuk *association* dimana sebuah elemen berisi elemen lainnya.

Tipe relasi yang mungkin terjadi pada *use case diagram*:

1. <<include>>, yaitu kelakuan yang harus terpenuhi agar sebuah *event* dapat terjadi, dimana pada kondisi ini sebuah *use case* adalah bagian dari *use case* lainnya.

2. <<extends>>, kelakuan yang hanya berjalan di bawah kondisi tertentu seperti menggerakkan peringatan.
3. <<communicates>>, merupakan pilihan selama asosiasi hanya tipe *relationship* yang dibolehkan antara aktor dan *use case*.

## 2. Class Diagram

*Class* adalah sebuah spesifikasi yang akan menghasilkan sebuah objek dan merupakan inti dari pengembangan dan desain berorientasi objek. *Class* menggambarkan keadaan (atribut/properti) suatu sistem, sekaligus menawarkan layanan untuk memanipulasi keadaan tersebut (metode/fungsi). *Class diagram* menggambarkan struktur dan deskripsi *class*, *package* dan objek beserta hubungan satu sama lain seperti pewarisan, asosiasi, dan lain-lain. *Class* memiliki tiga area pokok yaitu, nama, atribut, dan metode. Pada *UML*, *class* digambarkan dengan segi empat yang dibagi beberapa bagian. Bagian atas merupakan nama dari *class*. Bagian yang tengah merupakan struktur dari *class* (atribut) dan bagian bawah merupakan sifat dari *class* (metode/operasi).

Atribut dan metode dapat memiliki salah satu sifat berikut :

- a. *Private*, tidak dapat dipanggil dari luar *class* yang bersangkutan.
- b. *Protected*, hanya dapat dipanggil oleh *class* yang bersangkutan dan *class* lain yang mewarisinya.
- c. *Public*, dapat dipanggil oleh *class* lain.

Hubungan antar *class* :

- a. Asosiasi, yaitu hubungan statis antar *class*. Umumnya menggambarkan *class* yang memiliki atribut berupa *class* lain, atau *class* yang harus mengetahui eksistensi *class* lain.
- b. Agregasi, yaitu hubungan yang menyatakan bagian (“terdiri atas..”).
- c. Pewarisan, yaitu hubungan hirarki antar *class*. *Class* dapat diturunkan dari *class* lain dan mewarisi semua atribut dan metode *class* asalnya serta bisa menambahkan fungsionalitas baru. Sehingga *class* tersebut disebut anak dari *class* yang diwarisinya.
- d. Hubungan dinamis, yaitu rangkaian pesan yang di-*passing* dari satu *class* kepada *class* lain. Hubungan dinamis dapat digambarkan dengan menggunakan *sequence diagram* yang akan dijelaskan kemudian.

### **3. State Diagram**

Menggambarkan semua *state* (kondisi) yang dimiliki oleh suatu objek dari suatu *class* dan keadaan yang menyebabkan *state* berubah. *Statechart diagram* tidak digambarkan untuk semua *class*, hanya yang mempunyai sejumlah *state* yang terdefinisi dengan baik dan kondisi *class* berubah oleh *state* yang berbeda. *State* adalah sebuah kondisi selama kehidupan sebuah objek atau ketika objek memenuhi beberapa kondisi, melakukan beberapa aksi atau menunggu sebuah *event*. *State* dari sebuah objek dapat dikarakteristikkan oleh nilai dari satu atau lebih atribut-atribut dari *class*. *State* dari sebuah objek ditemukan dengan pengujian/pemeriksaan pada atribut dan hubungan dari objek.

Notasi *UML* untuk *state* adalah persegi panjang/bujur sangkar dengan ujung yang dibulatkan. Masing-masing diagram harus mempunyai satu dan hanya satu *start state* ketika objek mulai dibuat. Sebuah objek boleh mempunyai banyak *stop state*.



Sebuah *state transition* dapat mempunyai sebuah aksi dan/atau sebuah kondisi penjaga (*guard condition*) yang terasosiasi dengannya, dan mungkin juga memunculkan sebuah *event*. Sebuah aksi adalah kelakuan yang terjadi ketika *state transition* terjadi. Sebuah *event* adalah pesan yang dikirim ke objek lain di sistem. Kondisi penjaga adalah ekspresi *Boolean* (pilihan ya atau tidak) dari nilai atribut-atribut yang memungkinkan sebuah *state transition* hanya jika kondisinya benar. Kedua aksi dan penjaga adalah kelakuan dari objek dan secara tipikal menjadi operasi.

#### **4. Sequence Diagram**

Menggambarkan interaksi antara sejumlah objek dalam urutan waktu. Kegunaannya untuk menunjukkan rangkaian pesan yang dikirim antara objek juga interaksi antar objek yang terjadi pada titik tertentu dalam eksekusi sistem. Dalam *UML*, objek pada *sequence diagram* digambarkan dengan segi empat. Pada objek terdapat 3 cara untuk menamainya yaitu nama objek, *class* serta nama *class*.

#### **5. Collaboration Diagram**

Diagram ini menggambarkan interaksi objek yang diatur objek sekelilingnya dan hubungan antara setiap objek dengan objek yang lainnya. Dalam menunjukkan pertukaran pesan, *collaboration diagram* menggambarkan objek dan hubungannya (mengacu ke konteks). Jika penekannya pada waktu atau urutan gunakan *sequence diagram*.

*Collaboration diagram* berisi :

- a. Objek yang digambarkan dengan segi empat.
- b. Hubungan antara objek yang digambarkan dengan garis penghubung.
- c. Pesan yang digambarkan dengan teks dan panah dari objek yang mengirim pesan ke penerima pesan.



## **6. Activity Diagram**

Menggambarkan rangkaian aliran dari aktivitas, digunakan untuk mendeskripsikan aktivitas yang dibentuk dalam suatu operasi sehingga dapat juga digunakan untuk aktifitas lainnya. Diagram ini sangat mirip dengan *flowchart* karena memodelkan *workflow* dari satu aktivitas ke aktivitas lainnya atau dari aktivitas ke status. Pembuatan *activity diagram* pada awal pemodelan proses dapat membantu memahami keseluruhan proses. *Activity diagram* juga digunakan untuk menggambarkan interaksi antara beberapa *use case*.

## **7. Component Diagram**

Menggambarkan struktur dan hubungan antar komponen piranti lunak, termasuk ketergantungan (*dependency*) diantaranya. Komponen piranti lunak adalah modul berisi kode, baik berisi *source code* maupun *binary code*, baik *library* maupun *executable*. Umumnya komponen terbentuk dari beberapa *class* dan/atau *package*, tapi dapat juga dari komponen-komponen yang lebih kecil. Komponen dapat juga berupa *interface*, yaitu kumpulan layanan yang disediakan sebuah komponen untuk komponen lain.

## **8. Deployment Diagram**

Menggambarkan arsitektur fisik dari perangkat keras dan perangkat lunak sistem menunjukkan hubungan komputer dengan perangkat (*nodes*) satu sama lain dan jenis hubungannya. Didalam *nodes*, *executable component* dan objek yang dialokasikan untuk memperlihatkan unit perangkat lunak yang dieksekusi oleh *node* tertentu dan ketergantungan komponen.

## **BAB III**

### **GAMBARAN UMUM USAHA DAN ANALISIS**

#### **3.1 Sejarah Singkat Usaha**

Usaha penjualan *pc game* ini mulai dikembangkan oleh Eka Rudianto. Saat ini dia sedang menempuh perkuliahaan di salah satu universitas swasta di kota Bandung. Awal mulanya dia gemar bermain *game*, sehingga *game* menjadi salah satu dari hobinya. Dia memulai bermain *game* sejak duduk dibangku smp. Sejak bulan september 2011, terpikirkan olehnya untuk memulai usaha dibidang retail *game*. Dia menjual berbagai jenis *game offline* yang saat ini banyak digemari oleh masyarakat terutama dikalangan anak-anak dan remaja. Dari usahanya itu sekarang dia sudah bisa menghasilkan sejumlah uang yang dimana uang itu menjadi uang tambahan sakunya.

##### **3.1.1 Latar Belakang Usaha**

Go\_gaming adalah salah satu *re-seller* yang bergerak dibidang penjualan *pc game*, Go\_gaming berdiri/mulai berjaan pada bulan September 2011 dan saat ini masih berjalan dengan satu orang yaitu pemiliknya yang bernama Eka Rudianto. Dalam menjalankan usahanya Go\_gaming membutuhkan beberapa mediator sebagai alat untuk penjualan. Seperti yang sekarang dimanfaatkan adalah *Twitter* dan *Kaskus*, yaitu sebuah jejaring sosial dan forum jual/beli. Kedua mediator ini sangat membantu dalam hal mempromosikan hingga menjual *pc game-pc game* kepada konsumen. Setiap harinya Go\_gaming melakukan promosi-promosi *ter-update*, karena itu sangat diperlukan dalam penjualan sehingga banyak konsumen tetap ataupun calon konsumen yang akan banyak melihatnya, terutama pada saat ada *pc game-pc game*

terbaru. Go\_gaming juga memberikan sebuah *link* yang tertuju pada *link* untuk mendownload list games, yg berfungsi untuk membantu konsumen dalam mencari *pc game* kesukaannya.

**Visi :** menjadikan konsumen sebagai shabat dan memberi kepuasan.

**Misi :** menjadi toko game online yang terbaik dari kompetitor - kompetitor lainnya.

### **3.1.2 Uraian Pekerjaan**

Pada bagian uraian pekerjaan ini, akan menjelaskan bagaimana cara kerja penjualan di Go\_gaming. Berikut adalah penjelasannya :

a) Promosi

*User* akan melakukan promosi terlebih dahulu sebelum menjual *pc game*, agar konsumen mengetahui kalo Go\_gaming itu menjual *pc game*. *User* juga selalu *update* tentang *game-game* yang ada pada *list game* dan *game-game* yang terbaru.

b) Pemesanan

Konsumen akan memesan *pc game* yang telah di promosikan oleh *user*, ataupun memesan game yang ada pada *list game* yang disediakan oleh *user*.

Cara pemesanan bisa dengan :

- Langsung komen di *twitter* atau *kaskus*
- *Via sms*
- Dan *chatting* dengan *yahoo*

c) Transaksi

Konsumen dapat membayar langsung ke *user* apabila langsung bertemu atau dapat menstransfer ke nomer rekening bank yang disediakan oleh *user*.

d) Penjualan

Setelah pesanan dikonfirmasi oleh *user*, maka *user* akan mempersiapkan semua pesanan *pc game* sesuai dengan pesanan dari konsumen. Lalu *game* akan di *pack* dengan plastik *cd/dvd* dan siap dijual ke konsumen yang telah memesan *pc game* tersebut.

### **3.2 Analisis**

#### **3.2.1 Identifikasi Masalah**

Go\_Gaming merupakan salah satu *re-seller* yang bergerak di bidang penjualan *pc game*. Dalam proses penjualan *pc game* masih memanfaatkan jejaring sosial dan forum jual/beli, seperti *twitter* dan *kaskus*. Jadi dirasa masih kurang maksimal dalam menjual *pc game*-nya tersebut. Agar mampu bersaing dengan *re-seller pc game* yang telah lebih dulu ada. Dibutuhkan strategi dan promosi pemasaran yang baik untuk dapat meningkatkan jumlah penjualan. Untuk itu di bab analisis ini akan menguraikan deskripsi tentang sistem yang lama dan yang akan dibuat.

### **3.3 Deskripsi Sistem Lama**

#### **3.3.1 Penjualan Di *Twitter***

*Twitter* adalah situs mikro-blog yang gratis yang memungkinkan pengguna untuk mempublikasikan pesan singkat (140 karakter atau kurang-maka disebut mikro) melalui komputer dan ponsel. *Twitter* diluncurkan pada tahun 2007 dan hampir 1 juta orang menggunakannya selama kurun waktu satu tahun terakhir selama dia berdiri.

Pengguna atau *member Twitter* yang menggunakannya untuk menulis pesan singkat(SMS) disebut *BluProfondo*, merekalah yang mempublikasikan pesan-pesan singkatnya baik untuk dilihat umum (semua *members Twitterers* ) maupun hanya

untuk dilihat pribadi atau pengunjung tertentu (hanya untuk *members Twitterers* tertentu).

Biasanya, para pengguna akan mendaftar untuk mengikuti *BluProfondo* kepada pengguna lain yang mereka ajak. *Twitter* dapat menjadi alat yang memiliki prospek ke depan untuk menjadi jaringan komunikasi yang besar, membangun hubungan-bisnis dan tentunya alat yang baik untuk mempromosikan *blog*.

*Twitter* adalah cara menyenangkan dan berguna untuk mempromosikan *blog* dan mengarahkan lalu lintas *blog*. Kata para senior dan pakar internet *marketers* bahwa membangun hubungan adalah sebuah bagian penting dari pertumbuhan *blog*, dan *Twitter* adalah alat yang sangat baik untuk membangun hubungan yang positif tentunya.

### **3.3.1.1 Contoh Penjualan Di *Twitter***

#### **3.3.1.1.1 Promosi di *Twitter***

Disini *user* akan melakukan promosi penjualan *pc game*, yaitu dengan cara *user* akan *mention* atau mengirim *DM(Direct Message)* ke *account-account* iklan di *twitter*. Setelah itu tidak lama *account-account* tersebut akan meresponnya lalu akan me *RT(ReTweet)* *mention* ataupun *DM* dari *user*. Berikut adalah contoh gambar *RT* dari *account-account* yang di *mention/DM* oleh *user* :



Gambar 3.1  
Promosi di Twitter

#### 3.3.1.1.2 Pemesana Di Twitter

Pada bagian ini *user* akan menerima pesana dari konsumen, awalnya konsumen akan melihat promosi dari *user* lalu apabila konsumen tertarik dengan promosi dari *user*, maka konsumen akan memesan *game* dari *user*. Berikut gambar konsumen yang memesan *pc game* :





Gambar 3.2  
Pemesanan di *Twitter*

### 3.3.2 Penjualan Di *Kaskus*

*Kaskus* yg merupakan singkatan dari kasak kusuk ini, merupakan situs suatu forum atau komuniti dengan anggota terbanyak di Indonesia, dibandingkan dengan situs-situs lain. Nah kalo pengguna kaskus itu bisa disebut juga *kaskuser*. Kaskus ini katanya didirikan pada tanggal 6 november 1999 oleh 3 orang pemuda 3 orang itu bernama Andrew darwis, Ronald, dan Budi. Mereka membuat kaskus ini, saat mereka lagi melanjutkan studi di *seattle*, Amerika Serikat. Mereka semua orang Indonesia.

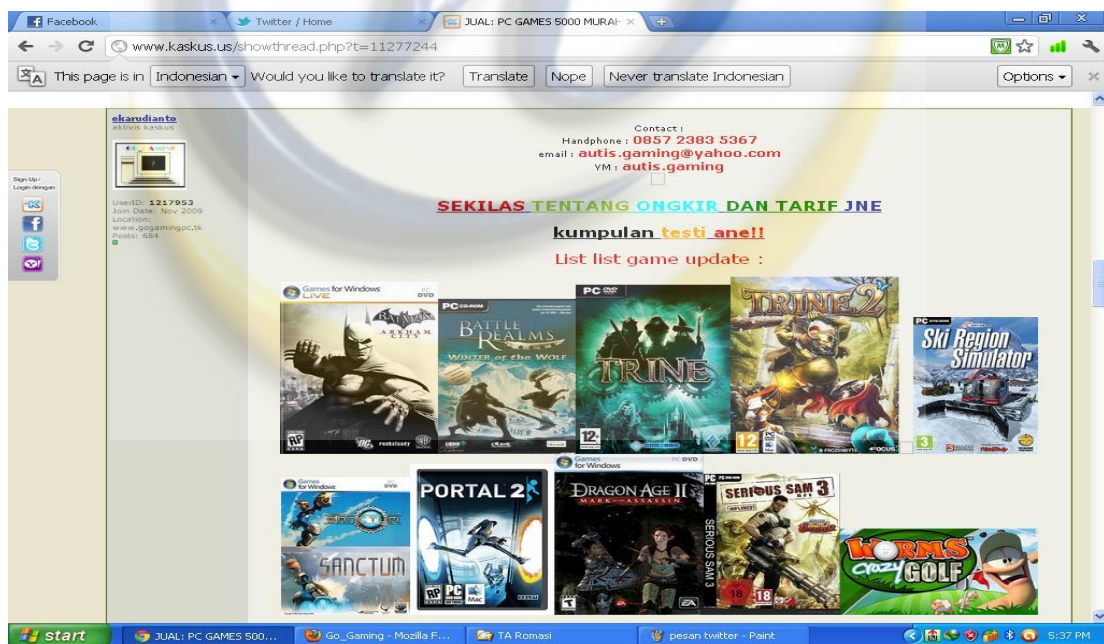


Kenapa domain *kaskus* menggunakan *.us* di situsnya. Awalnya *kaskus* menggunakan domain *.com*, nah gara-gara maraknya penyebaran virus *brontok* yang menyerang situs-situs besar di Indonesia pada saat itu yang dimana *kaskus* ini masuk dalam target penyerangan. Makanya manajemen dari *kaskus* ini terpaksa mengubah domain dari situs *kaskus* ini.

### 3.3.2.1 Contoh Penjualan Di *Kaskus*

#### 3.3.2.1.1 Promosi Di *Kaskus*

Disini tidak beda jauh dengan di *Twitter*, hanya yang membedakan adalah *user* akan mem-*posting* sendiri gambar dan spesifikasi *game* yang akan dipromosikannya dan tidak dibatasi karakter huruf. Disini juga terlihat lebih menarik daripada di *twitter*. Berikut adalah gambar *user* melakukan promosi di *kaskus* :

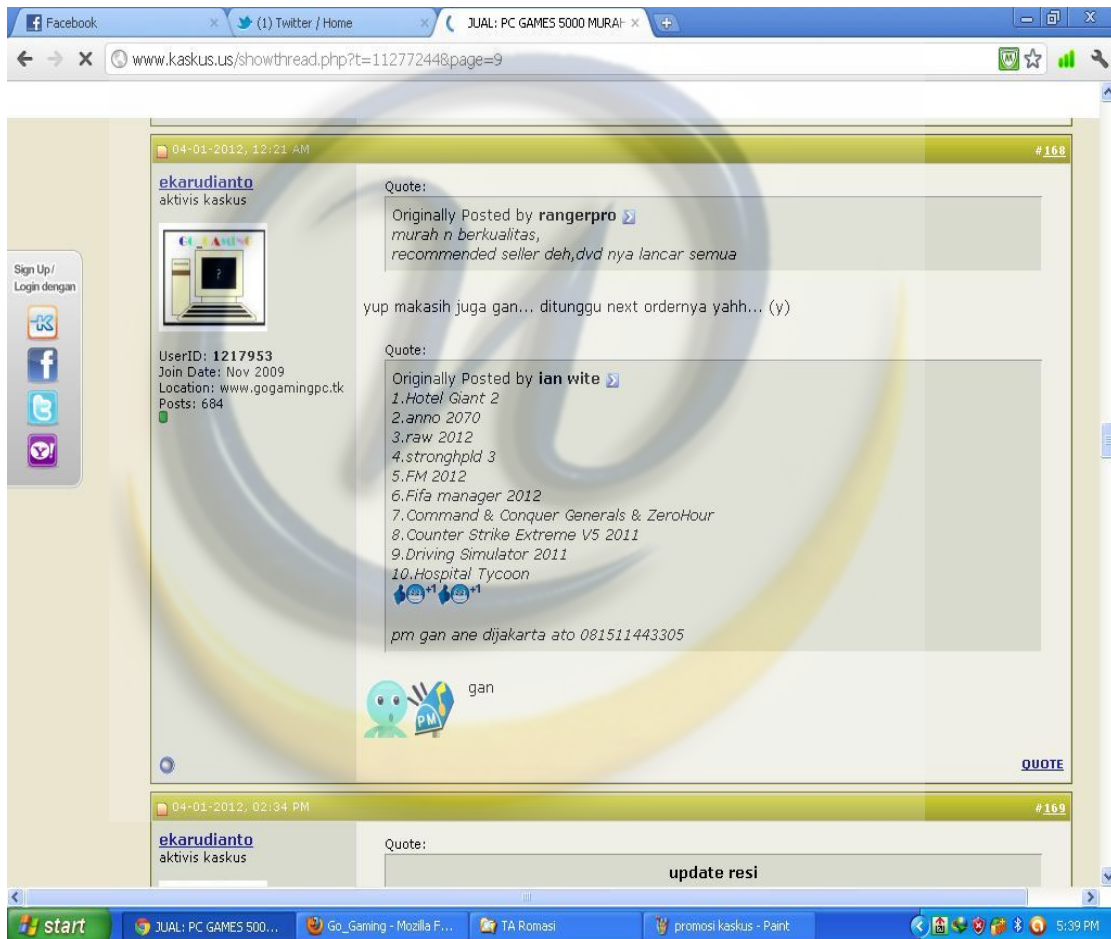


Gambar 3.3

Promosi di *Kaskus*

### 3.3.2.1.2 Pemesanan Di Kaskus

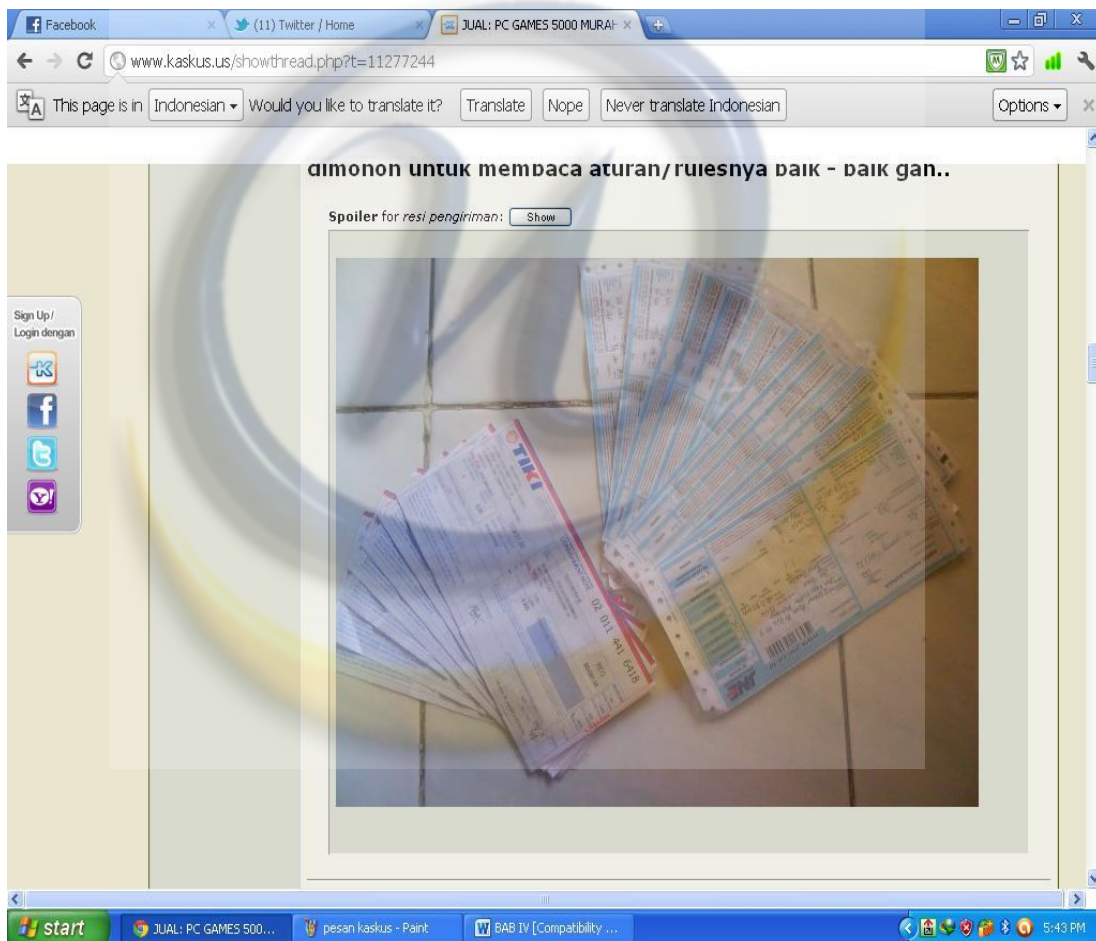
Disini juga tidak beda jauh bagaimana cara pemesanan pada *twitter*, yang membedakan adalah tampilannya saja. Terlihat lebih mudah dibaca dan tidak dibatasi karakter huruf, sehingga baik konsumen dan *user* bebas mengetikkan apa saja yang ia butuhkan. Berikut adalah contoh gambar pemesanan di *kaskus* :



Gambar 3.4  
Pemesanan di *Kaskus*

### 3.3.2.2 Contoh Transaksi Pengiriman

Sedangkan dibagian sini adalah contoh gambar bukti-bukti resi pengiriman yang dilakukan oleh user ke konsumen baik itu dari penjualan di *twitter* maupun *kaskus*, bukti-bukti ini diposting di *kaskus* karena di *kaskus* lebih mudah dilihat dibandingkan di *twitter*. Bukti-bukti ini diposting karena akan meningkatkan kepercayaan konsumen terhadap *user*. Berikut gambar bukti pengiriman :



Gambar 3.5  
Bukti Resi Pengiriman

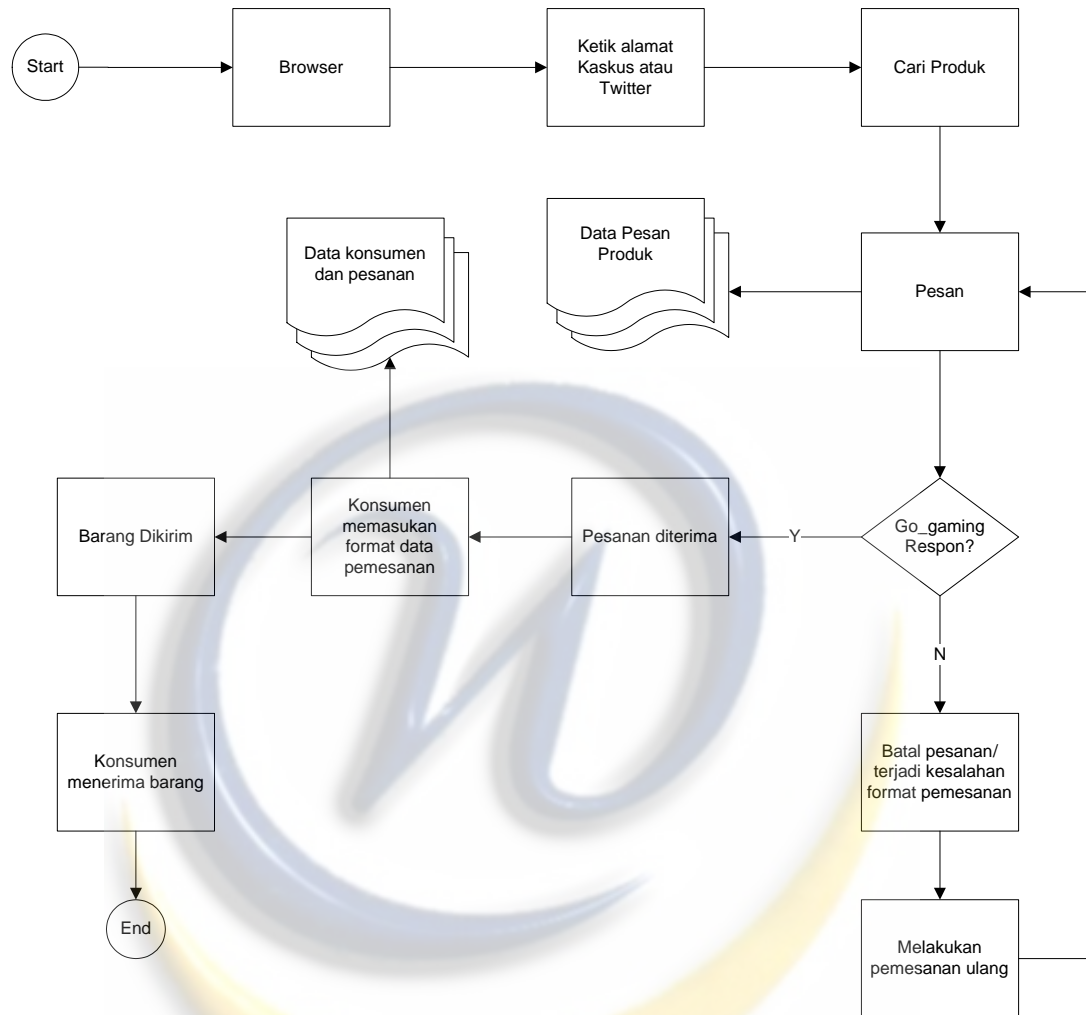
### **3.4 Hasil Analisis Sistem**

#### **3.4.1 Evaluasi Sistem Saat ini**

Pada sistem lama, ketika pengunjung ingin memesan atau membeli *pc game* mereka harus mengunjungi *twitter go\_gaming* atau mengunjungi ke *thread* kaskus *go\_gaming*. Setelah itu pengunjung dapat memilih sendiri apa *game* yang akan dipesan, lalu pengunjung menghubungi *go\_gaming* via komen, *sms*, dan *chat* di *ym* (*yahoo messenger*) ke *go\_gaming* untuk memesan *game-game* yang akan dipesan.

Setelah pesanan *game* sudah masuk ke *go\_gaming* maka proses penjualan dimulai, dengan cara *go\_gaming* meresponse *sms*, komen, dan *ym* dari pengunjung dan menentukan bagaimana cara bertransaksi, apakah ketemu secara langsung ataupun dikirim melalui jasa pengiriman dan harganya pun sudah disesuaikan terlebih dahulu. Tidak hanya membeli *game* yang ada di *list go\_gaming* itu saja, tetapi pengunjung dapat memesan *game* yang tidak ada pada *list*. *Go\_gaming* akan berusaha untuk memenuhi kebutuhan pengunjung, apabila *game* yg dipesan ada maka pengunjung dapat membelinya dari *go\_gaming*.

Berikut adalah analisis sistem yang dilakukan oleh penulis terhadap sistem lama seperti terlihat pada gambaran *flowchart diagram* pengunjung sistem lama sebagai berikut:



Gambar 3.6

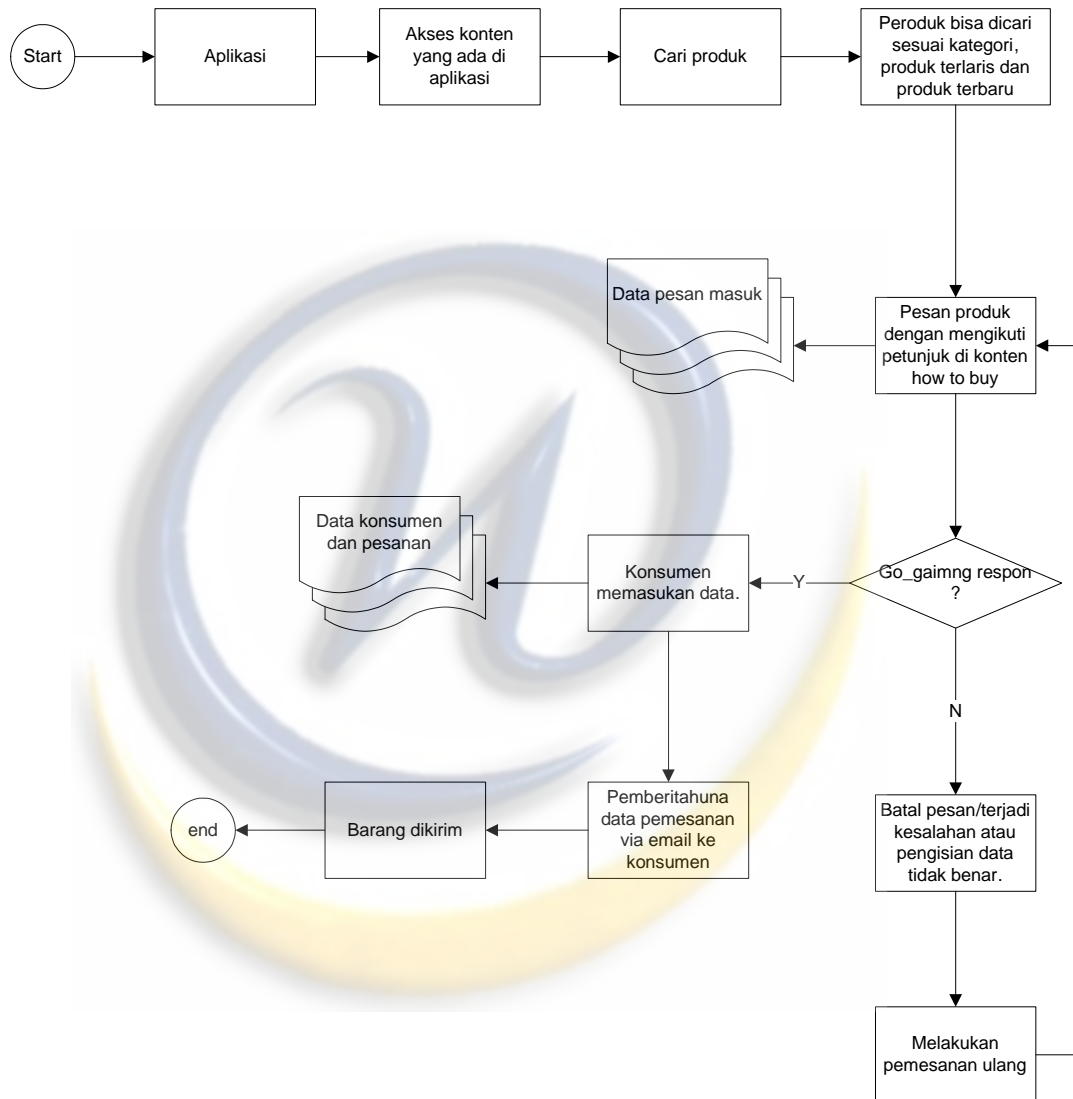
Flowchart Diagram Sistem Lama

### 3.5 Analisis Sistem Yang Akan Dibuat

Dilihat dari evaluasi sistem yang sudah ada diatas, maka dapat disimpulkan bahwa go\_gaming memerlukan sebuah *web* yang khusus diperuntukan untuk menjual *pc game*. Karena *web* ini akan sangat membantu Go\_gaming dalam menjual *pc game*, dan mempermudah Go\_gaming dalam *me-manage* data-data yang masuk kedalam



database. Dan gambar dibawah ini merupakan *flowchart diagram* pada sistem baru yang akan di buat, berikut Gambarnya :



Gambar 3.7  
*Flowchart Diagram Sistem Baru*

### **3.6 Analisis Kebutuhan Informasi**

#### **3.6.1 Kelemahan Sistem Saat ini**

Setelah menganalisis proses sistem lama di atas, maka disimpulkan kelemahan dari sistem lama adalah sebagai berikut :

1. Ketika masuk ke *twitter* atau *kaskus*, konsumen harus mencari *account* dan *thread* dari *go\_gaming*.
2. Pada *twitter* konsumen harus sering mengecek info-info ter-*update* dari *user*, karena sistem di *twitter* adalah *update* status tanpa ada kolom komen.
3. Pada *kaskus* hanya ada *cover* dari *game*, dan spesifikasi dari *game* konsumen harus mencarinya sendiri.
4. Ketika memesan *game*, konsumen harus menghubungi *user* terlebih dahulu dengan cara *sms* atau komen dari status di *twitter* dan *thread* di *kaskus* untuk membeli *game* yang diinginkan *user*.

#### **3.6.2 Kebutuhan Yang Terpenuhi Pada Kedua Sistem**

Pada sub ini penulis akan menganalisis kebutuhan apa saja yang sudah terpenuhi oleh kedua sistem, baik itu sistem lama maupun sistem yang akan dibuat. Berikut kebutuhan yang sudah terpenuhi oleh kedua sistem :

1. Sebagai mediator dalam mempromosikan dan menjual *pc game*.
2. Terhubung dengan internet.
3. Mudah diakses oleh konsumen.



### **3.6.3 Tujuan Sistem Baru**

Pada bagian ini penulis berusaha membuat sistem baru yang tentunya akan lebih menguntungkan *user* dari sistem yang sebelumnya, dan diharap *user* akan dengan mudah mengoperasikannya. Dibawah ini merupakan tujuan dibuatnya sistem baru :

1. Pembuatan kategori pada produk, untuk mempermudah konsumen dalam mencari produk yang diinginkan.
2. Ada produk-produk terlaris yang sering di beli konsumen, agar konsumen yang mempunyai referensi game apa yang sedang laku di pasaran.
3. Dapat memesan secara langsung(*online*) dan juga menghubungi langsung via email.
4. Adanya pemberitahuan email kepada konsumen secara langsung dan otomatis.

### **3.7 Model Metodologi *Waterfall***

Disini penulis menggunakan metodologi model *waterfall*, mengapa demikian? Karena metode model ini menurut penulis lebih efisien dan efektif dalam pembuatan sebuah program. Model ini dilakukan tahap demi tahap, sehingga apabila tahap yang satu belum beres maka tahap berikutnya belum bisa dilanjutkan. Sehingga pekerjaanpun dapat terkontrol dengan baik, dan mengurangi resiko kesalahan yang ada. Berikut 6 tahap dari model *waterfall* :

1. **Perancangan Sistem** : Penulis akan merancang aplikasi yang akan dibuat dan apa saja yang dibutuhkan.
2. **Analisis** : Pengumpulan data-data yang diperlukan untuk membuat aplikasi, agar penulis lebih memahami lagi.

3. **Desain** : Dilakukannya mendesain database dan antarmuka dari aplikasi yang akan di buat.
4. **Pengkodean** : Proses dimana penulisan bahasa pemrograman dalam pembuatan aplikasi.
5. **Pengujian** : Proses menguji kode-kode yang telah dibuat pada tahap pengkodean apakah masih terjadi kesalahan atau tidak.
6. **Pemeliharaan** : Proses pemeliharaan aplikasi ketika sudah selesai, dan perubahan aplikasi terjadi jika ada kesalahan atau adanya penambahan dan pengurangan yang diinginkan oleh user.

## **BAB IV**

### **PERANCANGAN SISTEM**

#### **4.1 Deskripsi Sistem**

##### **4.1.1 *Unified Modelling Language(UML)***

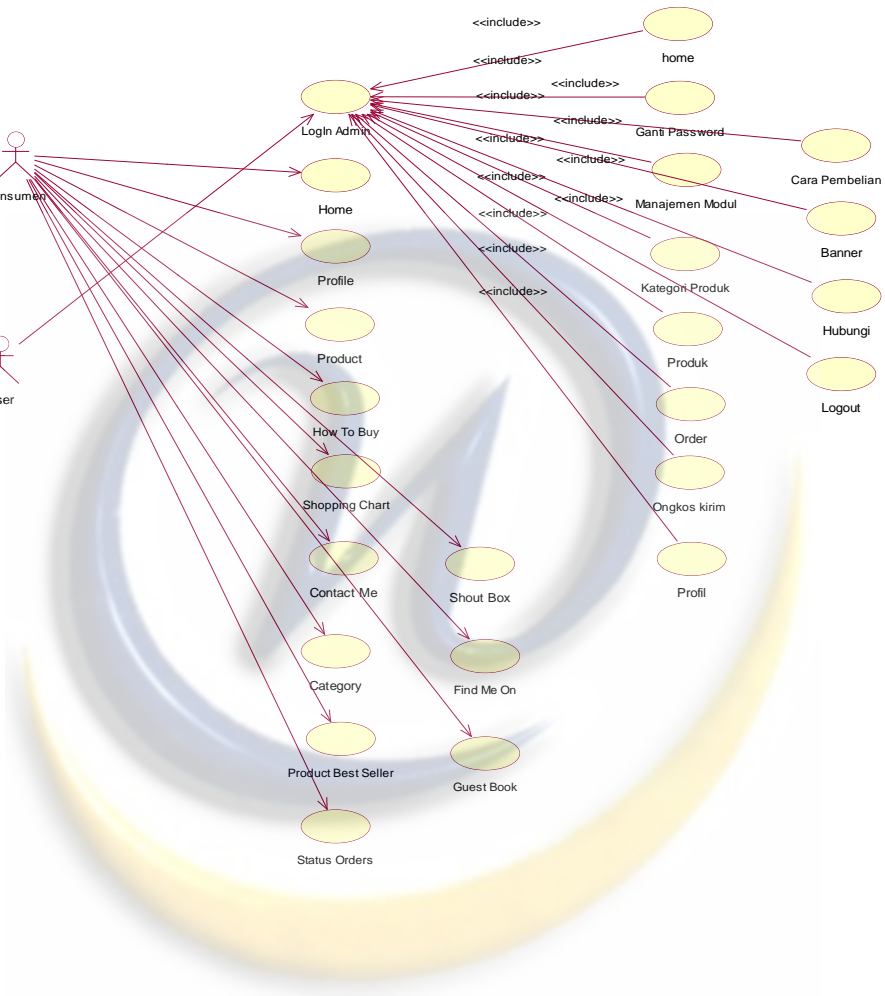
*Unified Modelling Language(UML)* adalah sebuah bahasa yang telah menjadi standar dalam *industry* untuk visualisasi, merancang, dan mendokumentasikan sistem piranti lunak. *UML* menawarkan sebuah standar untuk merancang model sebuah sistem. Bagian-bagian dari *UML* adalah *view*, *diagram*, *model element*, dan *general mechanism*.

Pada perancangan Perangkat Lunak Penjualan dan Pemesanan PC Game, bagian UML yang digunakan adalah diagram yang terdiri dari class diagram, use case diagram, dan activity diagram.

#### **4.2 Perancangan Diagram Sistem**

##### **4.2.1 *Use Case Diagram***

*Use case Diagram* menggambarkan fungsionalitas yang diharapkan dari sebuah sistem. Yang ditekankan adalah apa yang diperbuat sistem, dan bukan bagaimana sebuah *use case* mempresentasikan sebuah interaksi antara *actor* dengan sistem. *Use case* merupakan sebuah pekerjaan tertentu, misalnya *login* ke sistem, menginput data sebuah daftar kamar dan sebagainya. Seorang atau sebuah *actor* adalah sebuah entitas manusia atau mesin yang berinteraksi dengan sistem untuk melakukan pekerjaan – pekerjaan tertentu.



Gambar 4.1  
*Use Case Diagram*

#### 4.2.2 Definisi Actor

Actor yang terlibat dalam proses Aplikasi Penjualan dan Pemesanan *PC Game* pada Go\_Gaming ini yang dideskripsikan sebagai berikut:

Tabel 4.1  
Definisi Actor

No.	Aktor	Deskripsi
1	User	<ul style="list-style-type: none"><li>• Dapat mengolah <i>database</i></li><li>• Mengedit antarmuka aplikasi</li></ul>
2	Konsumen	<ul style="list-style-type: none"><li>• Memilih dan Membeli <i>PC Game</i></li><li>• Menghubungi <i>User</i></li></ul>

#### 4.2.4 Definisi Use Case

Definisi *use case* merupakan penjelasan dari *use case* atau proses-proses yang berlangsung didalam Aplikasi Penjualan dan Pemesanan *PC Game*. Adapun definisi dari *use case* tersebut adalah sebagai berikut :

Tabel 4.2  
Definisi Use Case

No.	Use Case	Deskripsi
1	<i>Login User</i>	<i>Validasi</i> untuk user sebelum masuk ke dalam sistem.
2	<i>Home</i>	Tampilan awal pada saat masuk ke <i>web</i> .
3	<i>Profile</i>	Gambaran umum tentang Go_Gaming.
4	<i>Product</i>	Tampilan produk-produk yang ada.
5	<i>How To Buy</i>	Menampilkan bagaimana cara berbelanja.
6	<i>Shopping Chart</i>	Menampilkan apa saja produk yang sudah di beli, bisa dilanjutkan membeli, menyudahi pembelian atau

		membatalkan pembelian.
7	<i>Contact Me</i>	Menghubungi <i>user</i> .
8	<i>Status Orders</i>	Melihat status konsumen.
9	<i>Category</i>	Produk di tampilkan sesuai kategori.
10	<i>Product Best Seller</i>	Menampilkan produk yang sering di beli.
11	<i>Shout Box</i>	Tempat untuk berkomentar.
12	<i>Banner</i>	<i>Link-link</i> yang tertuju dengan internet.
13	<i>Guest Book</i>	Tempat untuk mengisikan buku tamu.

#### 4.2.5 Skenario Use Case

Skenario (*flow of event*) untuk masing–masing *Use Case* dari aplikasi penjualan dan pemesanan *pc game* adalah sebagai berikut:

Tabel 4.3  
Skenario *Login User*

<i>Name</i>	<i>Login User</i>
<i>Actors</i>	<i>User</i>
<i>Goal</i>	Untuk menginjinkan user masuk ke halaman berikutnya.
<i>Precondition</i>	Sistem Aktif
<i>Postcondition</i>	<i>Login Berhasil</i>
<i>Steps</i>	<ol style="list-style-type: none"><li>1. Menyalakan komputer</li><li>2. Masuk ke halaman <i>administrator/user</i></li><li>3. Mengetikan <i>username</i> dan <i>password</i></li></ol>

Tabel 4.4  
Skenario *Home*

<i>Name</i>	<i>Home</i>
<i>Actors</i>	Konsumen
<i>Goal</i>	Untuk menampilkan menu utama
<i>Precondition</i>	Mengetikkan alamat situs
<i>Postcondition</i>	Masuk ke halaman <i>home</i>
<i>Steps</i>	<ol style="list-style-type: none"><li>1. Buka <i>browser</i></li><li>2. Mengetikkan alamat situs pada <i>browser</i></li><li>3. Masuk ke <i>web go_gaming</i></li><li>4. Muncul tampilan awal(<i>home</i>)</li></ol>

Tabel 4.5  
Skenario *Profile*

<i>Name</i>	<i>Profile</i>
<i>Actors</i>	Konsumen
<i>Goal</i>	Untuk menampilkan profil dari <i>go_gaming</i>
<i>Precondition</i>	Mengklik konten <i>profile</i>
<i>Postcondition</i>	Masuk ke halaman <i>profile</i>
<i>Steps</i>	<ol style="list-style-type: none"><li>1. Masuk ke <i>web go_gaming</i></li><li>2. Klik <i>profile</i></li><li>3. Muncul tampilan <i>profile go_gaming</i></li></ol>



Tabel 4.6  
Skenario *Product*

<i>Name</i>	<i>Product</i>
<i>Actors</i>	Konsumen
<i>Goal</i>	Untuk menampilkan produk-produk
<i>Precondition</i>	Mengklik konten <i>product</i>
<i>Postcondition</i>	Masuk ke halaman <i>product</i>
<i>Steps</i>	<ol style="list-style-type: none"><li>1. Masuk ke <i>web go_gaming</i></li><li>2. Klik <i>product</i></li><li>3. Muncul tampilan produk-produk <i>go_gaming</i></li></ol>

Tabel 4.7  
Skenario *How To Buy*

<i>Name</i>	<i>How To buy</i>
<i>Actors</i>	Konsumen
<i>Goal</i>	Untuk menampilkan tata cara pembelian
<i>Precondition</i>	Mengklik konten <i>How To Buy</i>
<i>Postcondition</i>	Masuk ke halaman <i>How To Buy</i>
<i>Steps</i>	<ol style="list-style-type: none"><li>1. Masuk ke <i>web go_gaming</i></li><li>2. Klik <i>How To Buy</i></li><li>3. Muncul tampilan tata cara membeli</li></ol>

Tabel 4.8  
Skenario *Shopping Chart*

<i>Name</i>	<i>Shopping Chart</i>
<i>Actors</i>	Konsumen
<i>Goal</i>	Untuk menampilkan produk yang telah di beli
<i>Precondition</i>	Mengklik konten <i>Shopping Chart</i>
<i>Postcondition</i>	Masuk ke halaman <i>Shopping Chart</i>
<i>Steps</i>	<ol style="list-style-type: none"><li>1. Masuk ke <i>web go_gaming</i></li><li>2. Membeli 1 atau beberapa produk</li><li>3. Klik <i>Shopping Chart</i></li><li>4. Muncul tampilan barang yang dibeli, apabila tidak membeli maka tidak akan muncul</li></ol>

Tabel 4.9  
Skenario *Contact Me*

<i>Name</i>	<i>Contact Me</i>
<i>Actors</i>	Konsumen
<i>Goal</i>	Dapat mengirim pesan kepada <i>go_gaming</i>
<i>Precondition</i>	Mengklik konten <i>Contact Me</i>
<i>Postcondition</i>	Masuk ke halaman <i>Contact Me</i>
<i>Steps</i>	<ol style="list-style-type: none"><li>1. Masuk ke <i>web go_gaming</i></li><li>2. Klik <i>Contact Me</i></li><li>3. Mengisikan form yang ada untuk mengirim pesan</li></ol>

Tabel 4.10  
Status Orders

<i>Name</i>	<i>Status Orders</i>
<i>Actors</i>	Konsumen
<i>Goal</i>	Dapat melihat status orders konsumen yang baru dan Lunas.
<i>Precondition</i>	Mengklik konten <i>Status Orders</i>
<i>Postcondition</i>	Masuk ke halaman <i>Status Orders</i>
<i>Steps</i>	<ol style="list-style-type: none"><li>1. Masuk ke <i>web go_gaming</i></li><li>2. Klik <i>Status orders</i></li><li>3. Muncul tampilan <i>status orders dari konsumen yang baru dan lunas.</i></li></ol>

Tabel 4.11  
Skenario *Category*

<i>Name</i>	<i>Category</i>
<i>Actors</i>	Konsumen
<i>Goal</i>	Untuk menampilkan produk–produk berdasarkan kategori
<i>Precondition</i>	Mengklik menu <i>Category</i> pada <i>side</i> sebelah kanan
<i>Postcondition</i>	Menampilkan kategori–kategori <i>game</i>
<i>Steps</i>	<ol style="list-style-type: none"><li>1. Masuk ke <i>web go_gaming</i></li><li>2. Klik <i>Category</i></li><li>3. Memilih dan membeli game sesuai kategori yang ada</li></ol>

Tabel 4.12  
Skenario *Product Best Seller*

<i>Name</i>	Product Best Seller
<i>Actors</i>	Konsumen
<i>Goal</i>	Untuk menampilkan produk–produk yang sering dibeli
<i>Precondition</i>	Mengklik konten <i>product best seller</i> pada <i>side</i> sebelah kanan
<i>Postcondition</i>	Menampilkan produk–produk yang sering dibeli
<i>Steps</i>	<ol style="list-style-type: none"><li>1. Masuk ke <i>web go_gaming</i></li><li>2. Klik <i>product best seller</i></li><li>3. Memilih dan membeli produk terlaris</li></ol>

Tabel 4.13  
Skenario *Shout Box*

<i>Name</i>	<i>Shout Box</i>
<i>Aktors</i>	Konsumen
<i>Goal</i>	Untuk menampilkan perbincangan–perbincangan konsumen
<i>Precondition</i>	Mengklik konten <i>shout box</i> pada <i>side</i> sebelah kanan
<i>Postcondition</i>	Mengetikan komentar dan menampilkan perbincangan konsumen
<i>Steps</i>	<ol style="list-style-type: none"><li>1. Masuk ke <i>web go_gaming</i></li><li>2. Klik <i>shout box</i> pada <i>side</i> sebelah kanan</li><li>3. Mengisi <i>form</i> yang ada untuk memulai perbincangan</li></ol>

Tabel 4.14  
Scenario *Find Me On*

<i>Name</i>	<i>Find Me On</i>
<i>Actors</i>	Konsumen
<i>Goal</i>	Untuk menampilkan <i>link-link</i> yang dapat tertuju pada situs–situs lain yang berhubungan dengan <i>go_gaming</i>
<i>Precondition</i>	Mengklik konten <i>find me on</i> pada <i>side</i> sebelah kanan
<i>Postcondition</i>	Menampilkan isi dari <i>find me on</i>
<i>Steps</i>	<ol style="list-style-type: none"><li>1. Masuk ke <i>web go_gaming</i></li><li>2. Klik <i>find me on</i> pada <i>side</i> sebelah kanan</li><li>3. Memilih dan mengklik gambar yang ada untuk terhubung di <i>Twitter</i> dan <i>Kaskus</i>.</li></ol>

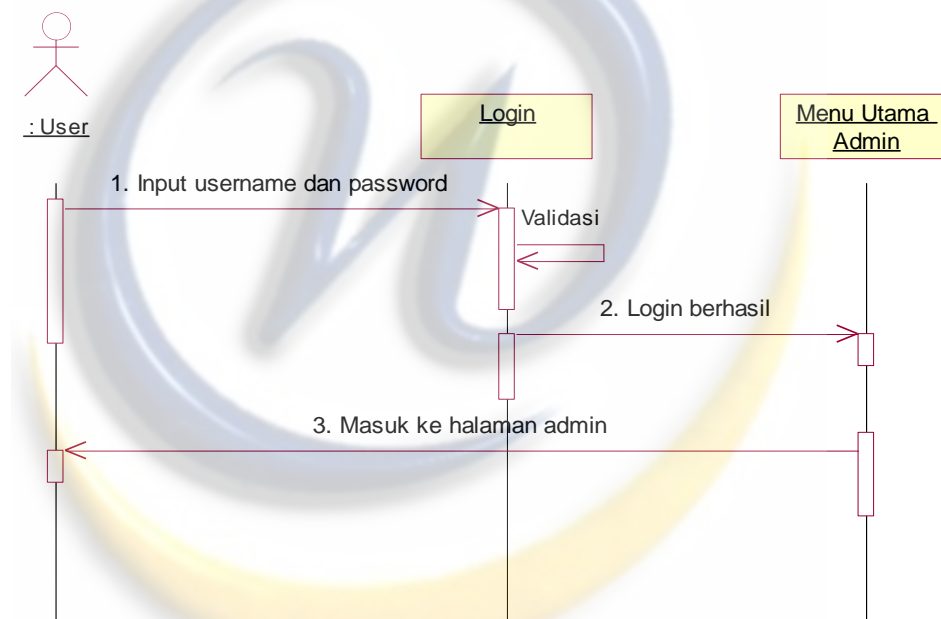
Tabel 4.15  
Skenario *Guest Book*

<i>Name</i>	<i>Guest Book</i>
<i>Actors</i>	Konsumen
<i>Goal</i>	Untuk menampilkan data–data konsumen yang sudah mengisi <i>guest book</i>
<i>Precondition</i>	Mengklik konten <i>guest book</i> pada <i>side</i> sebelah kanan
<i>Postcondition</i>	Menampilkan isi dari <i>guest book</i>
<i>Steps</i>	<ol style="list-style-type: none"><li>1. Masuk ke <i>web go_gaming</i></li><li>2. Klik <i>guest book</i> pada <i>side</i> sebelah kanan</li><li>3. Mengisi <i>form</i> yang ada</li><li>4. Data konsumen tertera di <i>web go_gaming</i></li></ol>

#### 4.2.6 Sequence Diagram

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

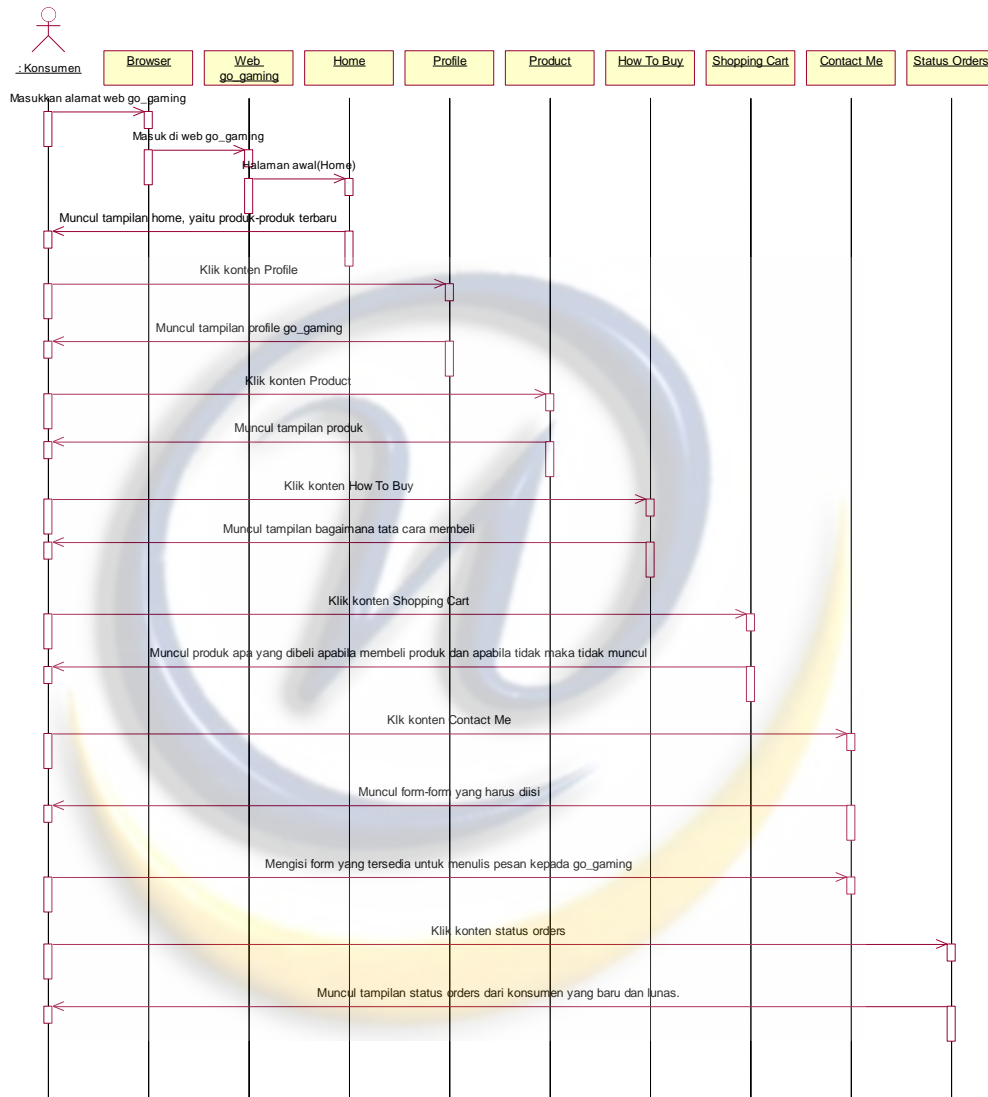
##### 4.2.6.1 Sequence Diagram Login User



Gambar 4.2

*Sequence Diagram Login User*

#### 4.2.6.2 Sequence Diagram Menu Header



Gambar 4.3

*Sequence Diagram Menu Header*

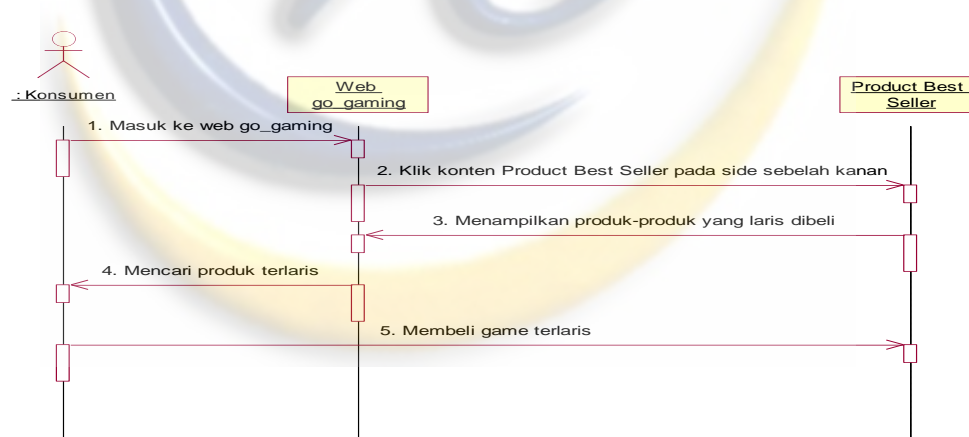


#### 4.2.6.3 Sequence Diagram Category



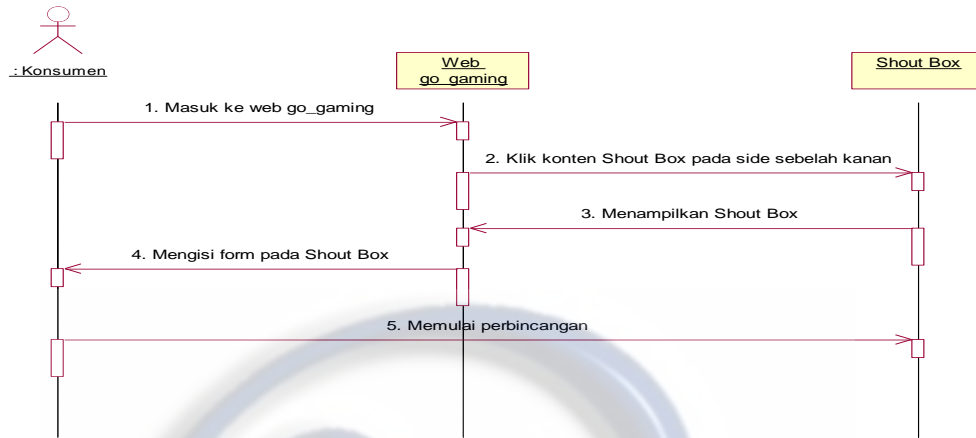
Gambar 4.4  
Sequence Diagram Category

#### 4.2.6.4 Sequence Diagram Product Best Seller



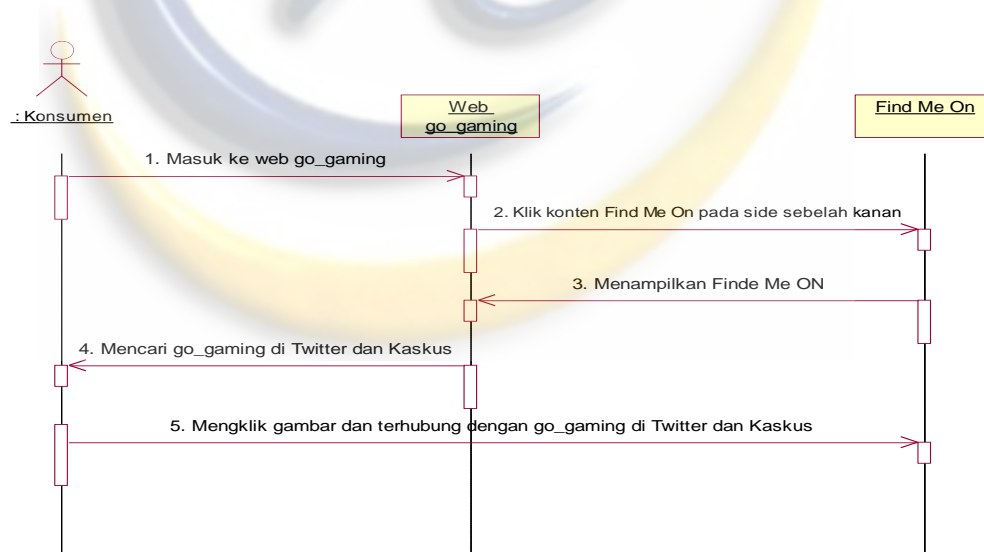
Gambar 4.5  
Sequence Diagram Product Best Seller

#### 4.2.6.5 Sequence Diagram Shout Box



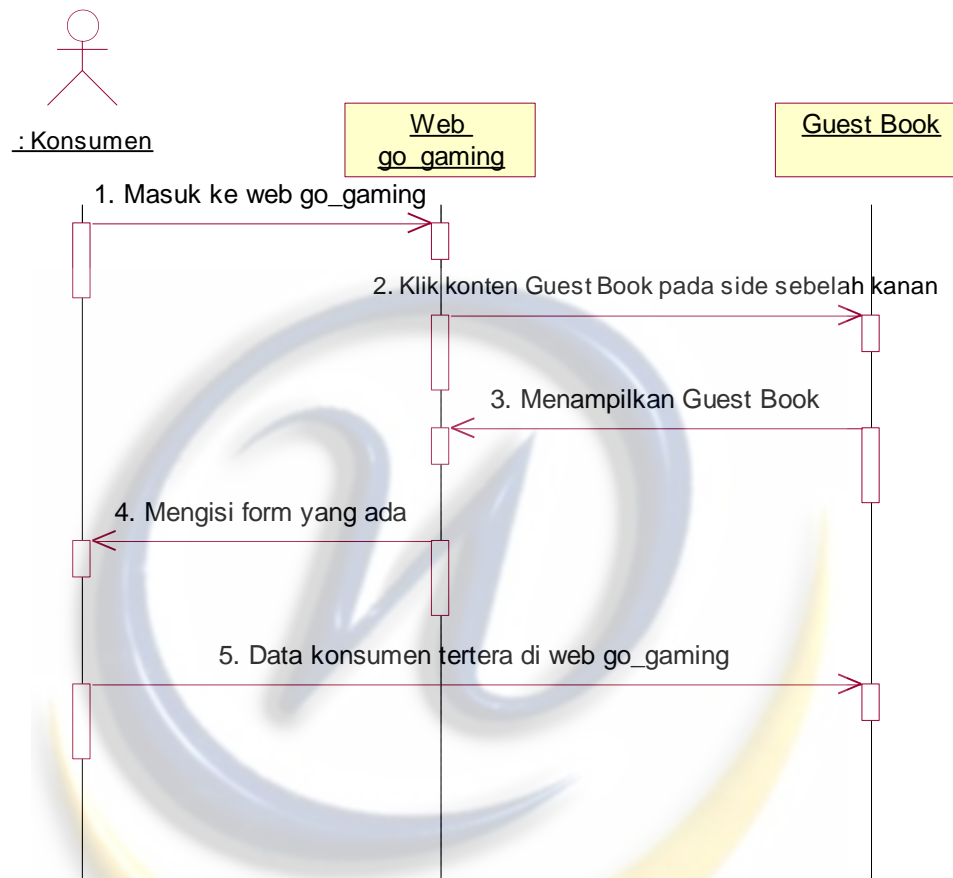
Gambar 4.6  
*Sequence Diagram Shout Box*

#### 4.2.6.6 Sequence Diagram Find Me On



Gambar 4.7  
*Sequence Diagram Find Me On*

#### 4.2.6.7 Sequence Diagram Guest Book



Gambar 4.8

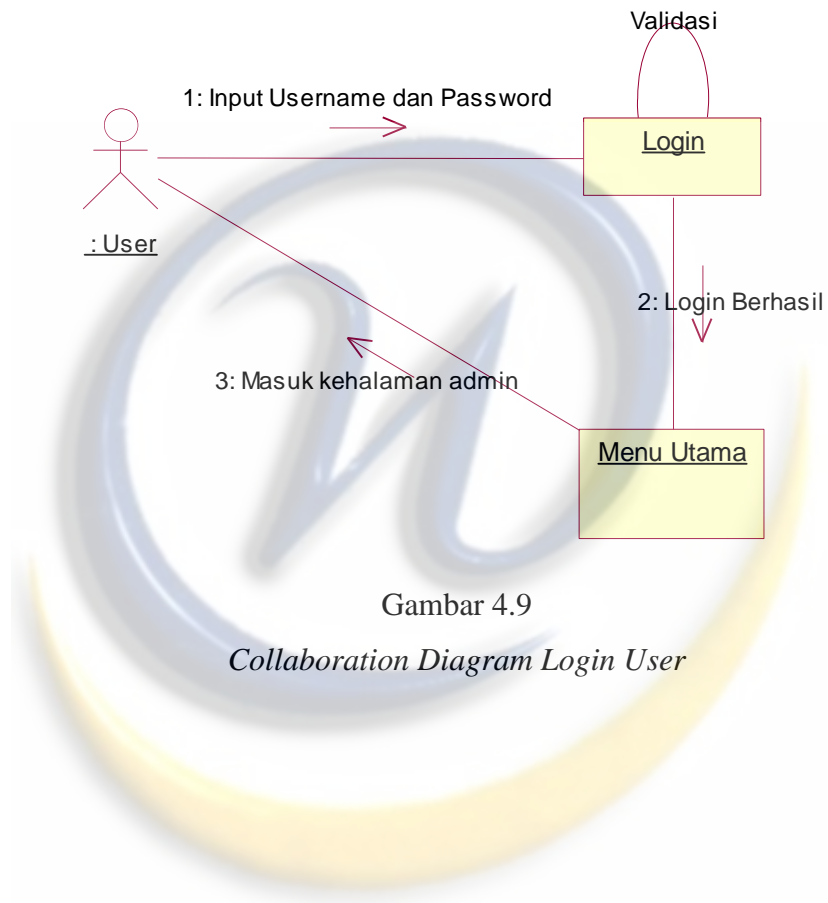
*Sequance Diagram Guest Book*

#### 4.2.7 Collaboration Diagram

*Collaboration* Diagram digunakan untuk memodelkan interaksi antar objek didalam sistem, berbeda dengan *sequence Diagram*, yang lebih menonjolkan kronologis dari operasi-operasi yang dilakukan, *collaboration* Diagram lebih fokus

pada pemahaman atas keseluruhan operasi yang dilakukan objek. Berikut ini *collaboration* Diagram dari aplikasi penjualan dan pemesanan pc game :

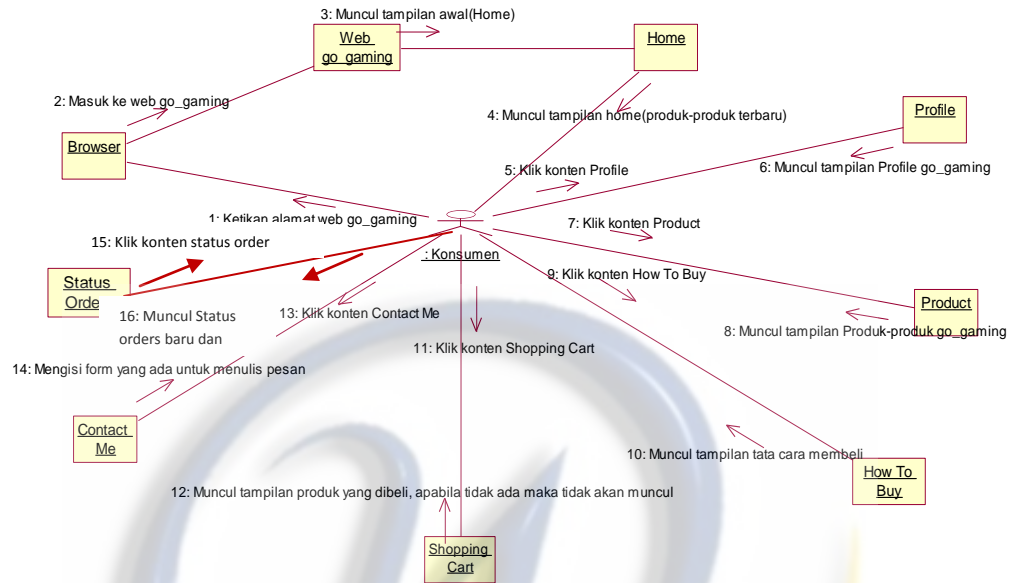
#### **4.2.7.1 Collaboration Diagram Login User**



Gambar 4.9

*Collaboration Diagram Login User*

#### 4.2.7.2 Collaboration Diagram Menu Header



Gambar 4.10

Collaboration Diagram Menu Header

#### 4.2.7.3 Collaboration Diagram Category



Gambar 4.11

Collaboration Diagram Category

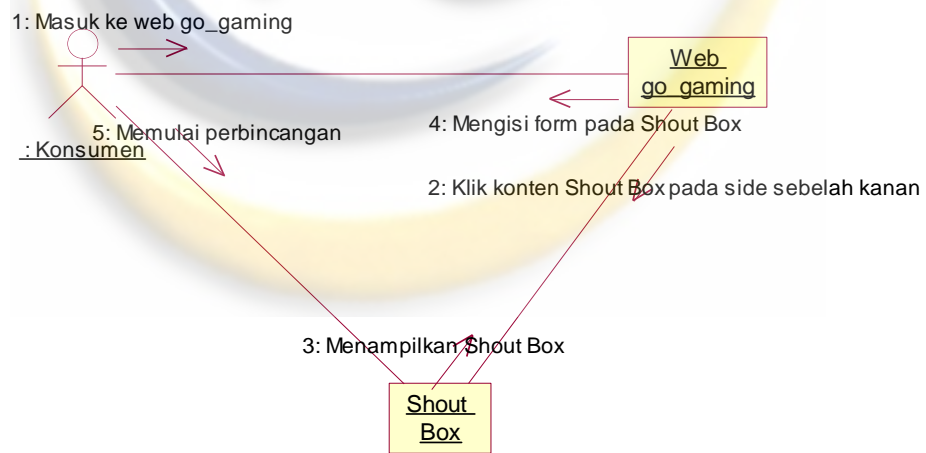
#### 4.2.7.4 Collaboration Product Best Seller



Gambar 4.12

*Collaboration Product Best Seller*

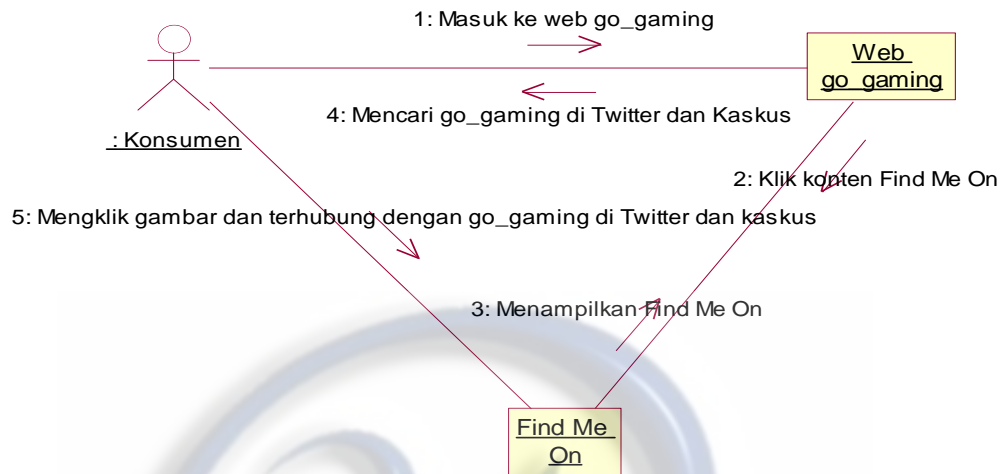
#### 4.2.7.5 Collaboration Diagram Shout Box



Gambar 4.13

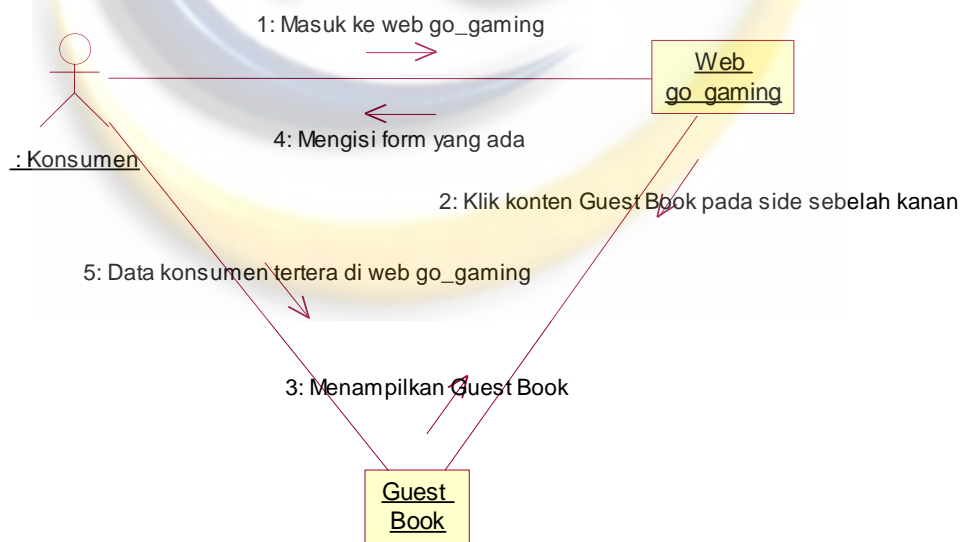
*Collaboration Diagram Shout Box*

#### 4.2.7.6 Collaboration Diagram Find Me On



Gambar 4.14  
*Collaboration Diagram Find Me On*

#### 4.2.7.7 Collaboration Diagram Guest Book



Gambar 4.15  
*Collaboration Diagram Guest Book*



#### 4.2.8 Kamus Data

Berikut ini akan dijelaskan tabel – tabel yang digunakan dalam perancangan proses *Unified Modelling Language* (UML).

1. Tabel Kategori

Nama Tabel : kategori

Primary key : id\_kategori

Keterangan : Tabel ini berisi data-data kategori.

Tabel 4.16

Tabel Kategori

Field Name	Data Type	Field Size	Description
id_kategori	Int	5	Primary Key(id kategori)
nama_kategori	Varchar	30	Nama kategori
kategori_seo	Varchar	30	

2. Tabel Kota

Nama Tabel : kota

Primary key : id\_kota

Keterangan : Tabel ini berisi data-data ongkos kirim.

Tabel 4.17

Tabel Kota

Field Name	Data Type	Field Size	Description
id_kota	Int	5	Primary Key(id kota)
nama_kota	Varchar	30	Nama kota
Ongkos_kirim	int	8	Ongkos kirim

3. Tabel Orders

Nama Tabel : orders

Primary key : id\_orders

Keterangan : Tabel ini berisi data-data orders.

Tabel 4.18

Tabel Orders

Field Name	Data Type	Field Size	Description
Id_orders	Int	5	Primary key(id Orders)
Nama_kostumer	Varchar	30	Nama konsumen
Alamat	Text	-	Alamat konsumen
Telepon	Varchar	20	Nomor telp konsumen
Email	Varchar	30	Email konsumen
Status_order	Varchar	10	Status pesanan
Tgl_order	Date	-	Tanggal pesan
Jam_order	Time	-	Jam pesan
Id_kota	Int	5	Id kota

4. Tabel Orders Detail

Nama Tabel : orders\_detail

Primary key : -

Keterangan : Tabel ini berisi data-data orders detail.

Tabel 4.19

Tabel Orders Detail

Field Name	Data Type	Field Size	Description
Id_orders	Int	5	Foreign key(id orders)
Id_produk	Int	5	Foreign key(id produk)

Jumlah	Int	5	Foreign key (jumlah produk yang dibeli)
--------	-----	---	---

5. Tabel Orders Temporary

Nama Tabel : orders\_temp

Primary key : id\_orders\_temp

Keterangan : Tabel ini berisi data-data order temporary.

Tabel 4.20

Tabel Orders Temporary

Field Name	Data Type	Field Size	Description
id_orders_temp	Int	5	Primary key (id orders temporary)
id_produk	Int	5	Id produk
id_session	Varchar	5	Id sesi
jumlah	Int	5	Jumlah produk
tgl_order_temp	Date	-	Tanggal pesanan
jam_order_temp	Time	-	Jam pesan
stok_temp	Int	5	Stok produk

6. Tabel Produk

Nama Tabel : produk

Primary key : id\_produk

Keterangan : Tabel ini berisi data-data produk.

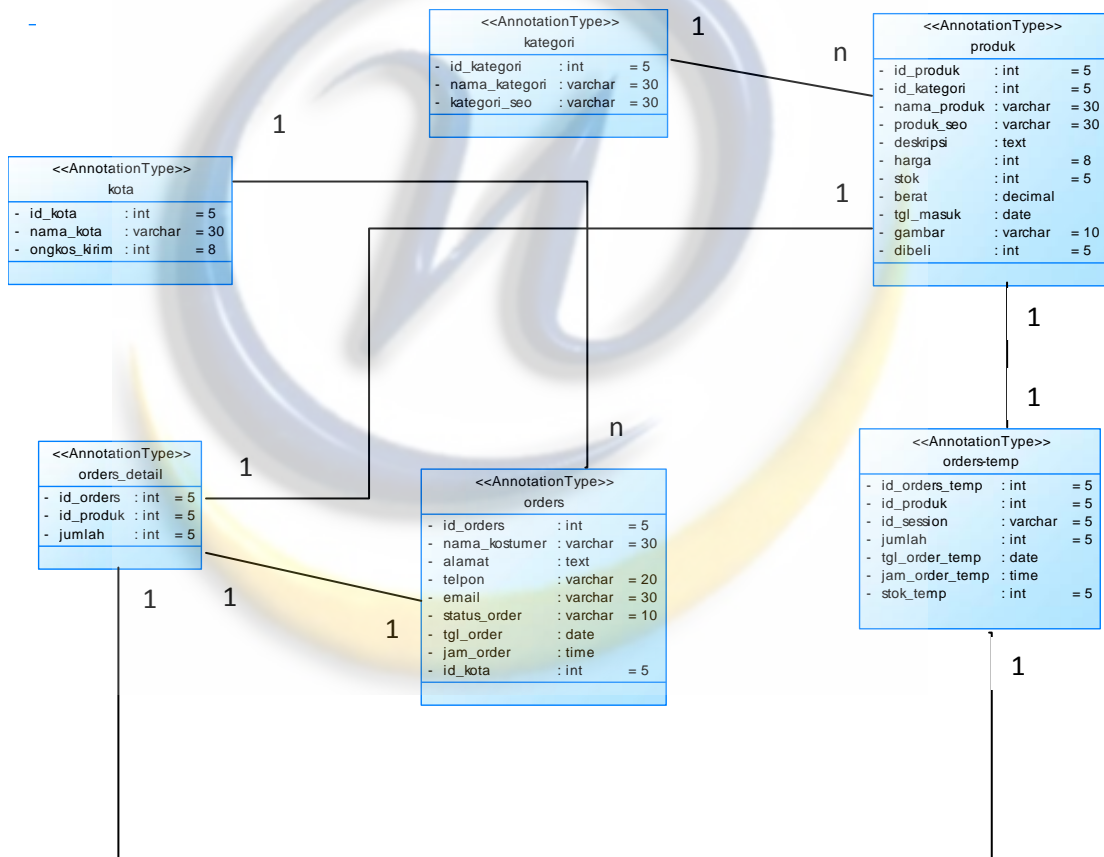
Tabel 4.21

Tabel Produk

Field Name	Data Type	Field Size	Description
id_produk	Int	5	Primary key(id produk)
id_kategori	Int	5	Id kategori
nama_produk	Varchar	30	Nama produk
produk_seo	Varchar	30	Produk terlaris (search engine optimization)
deskripsi	Text	-	Keterangan produk
harga	Int	8	Harga produk
stok	Int	5	Stok produk
berat	Decimal	5,2	Berat produk
tgl_masuk	Date	-	Tanggal upload produk
gambar	Varchar	10	Cover produk
dibeli	Int	5	Jumlah produk yang dibeli

#### 4.2.8.1 Class Diagram Penjualan PC Game

Untuk memodelkan *class* yang akan dibuat, maka digunakan *class diagram* yang akan menggambarkan *class-class* apa yang terdapat pada sistem yang akan dibuat. Selain itu, *class diagram* menunjukkan relasi antar *class* yang akan digunakan sebagai pedoman dalam perancangan sistem. Berikut *class diagram* yang digunakan oleh sistem:

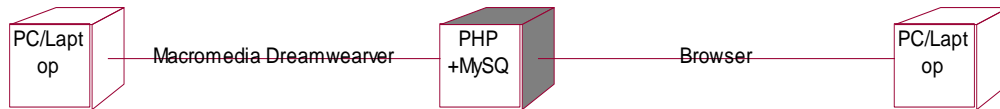


Gambar 4.16

*Class Diagram Penjualan PC Game*

### 4.3 Deployment Diagram

Kebutuhan perlengkapan *hardware* dan *software* untuk mendukung perealisasiian aplikasi penjualan dan pemesanan *pc game* yang telah dibuat sebagai berikut :

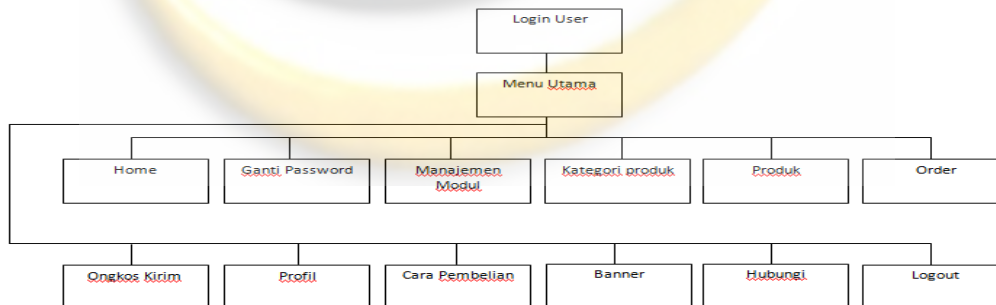


Gambar 4.17  
Deployment Diagram

### 4.4 Rancangan Menu

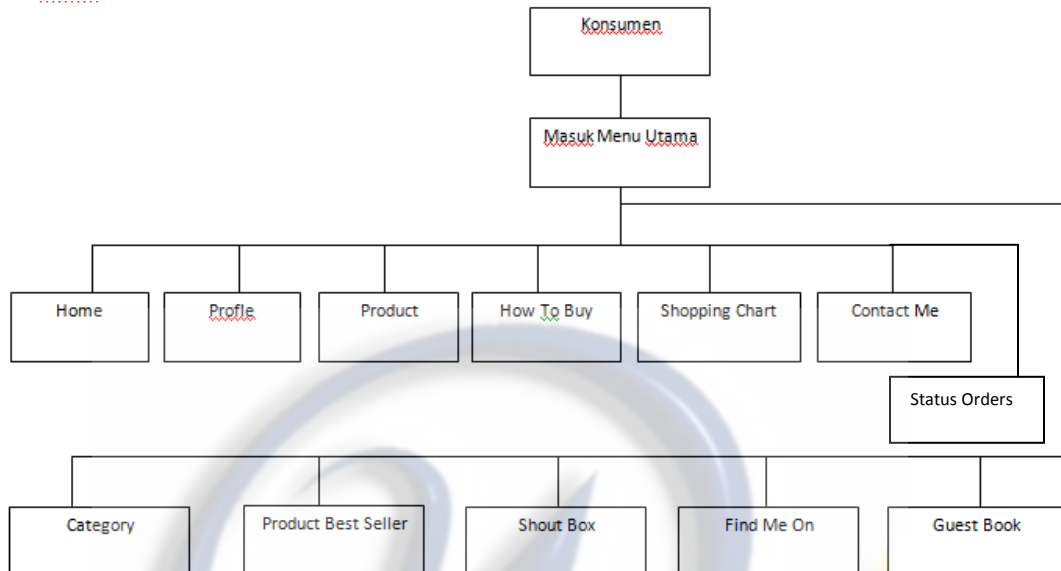
Rancangan menu digunakan untuk memudahkan pemakaian dalam menjalankan program komputer sehingga pada saat menjalankan program, pemakai tidak mengalami kesulitan dalam memilih menu – menu yang diinginkan. Struktur menu program dapat dilihat pada gambar dibawah ini :

#### 4.4.1 Rancang Menu User



Gambar 4.18  
Rancang Menu User

#### 4.4.2 Rancang Menu Konsumen



Gambar 4.19  
Rancang Menu Konsumen

#### 4.5 Perancangan Antarmuka

Perancangan Antarmuka menjelaskan rutinitas program yang akan dijalankan oleh sebuah sistem untuk menjelaskan interaksi antara pemakai (*user*) dan konsumen dengan program yang akan dibuat. Pada sub bab ini akan digambarkan rancangan antarmuka yang akan digunakan dalam sistem, berikut gambar perancangan antarmuka :



*a. Menu Login User*

HEADER							
<div>Login</div> <div><div>Picture</div><div><table border="1"><tr><td>Username</td><td>:</td><td></td></tr><tr><td>Password</td><td>:</td><td></td></tr></table><div>Login</div></div></div>		Username	:		Password	:	
Username	:						
Password	:						
FOOTER							

Gambar 4.20  
*Menu login User*

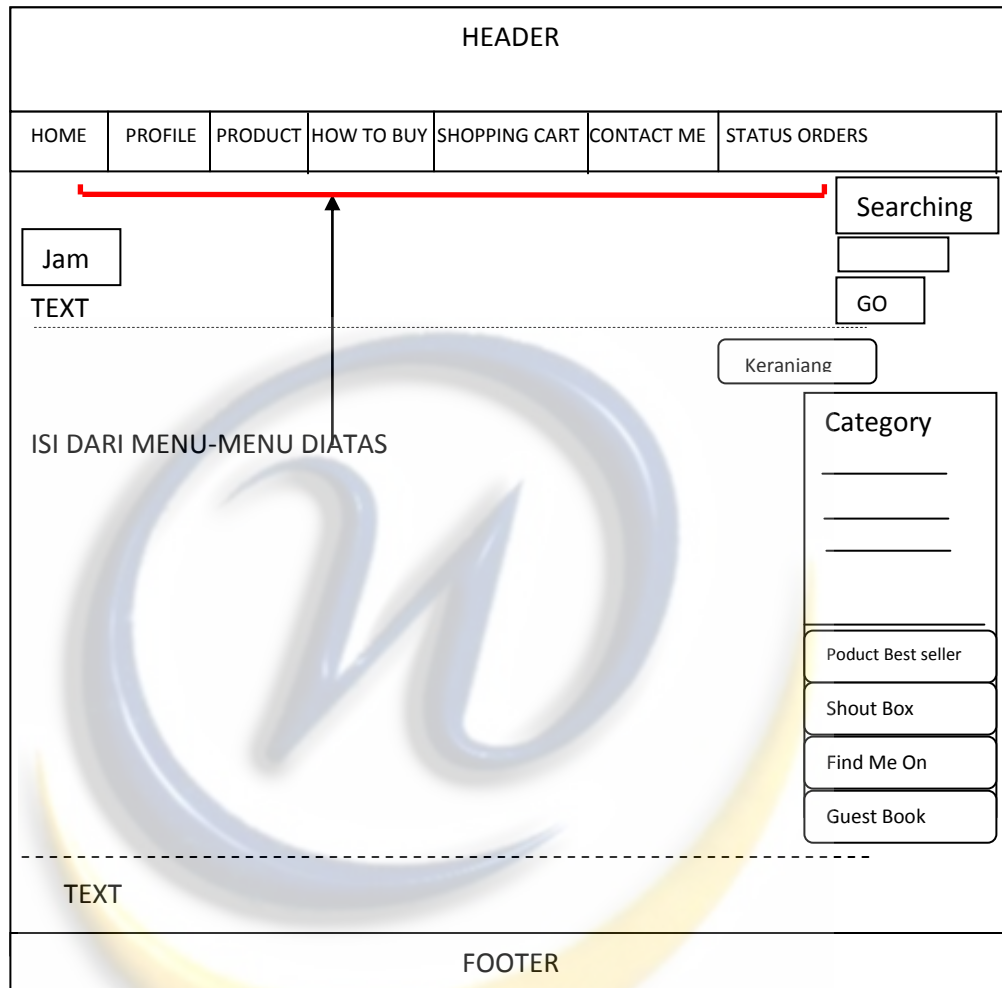
*b. Menu User*

HEADER	
Home	<p>Text</p> <p>-----</p> <p>Kata sambutan selamat datang untuk</p>
Ganti Password	
Manajemen Modul	
Kategori Produk	
Produk	
Order	
Ongkos Kirim	
Profil	
Cara Pembelian	
Banner	
Hubungi	
Logout	
<div>Waktu dan tanggal</div>	
FOOTER	

Gambar 4.21

*Menu User*

c. Menu Header



Gambar 4.22

Menu Header

*d. Menu Side Category*

HEADER						
HOME	PROFILE	PRODUCT	HOW TO BUY	SHOPPING CART	CONTACT ME	STATUS ORDERS
<div>Jam</div>				<div>Searching</div> <div></div> <div>GO</div>		
TEXT				<div>Keraniang</div>		
PAGE MENU UTAMA				<div>Category</div> <div></div> <div></div> <div></div> <div>Product Best seller</div> <div>Shout Box</div> <div>Find Me On</div> <div>Guest Book</div>		
TEXT						
FOOTER						

Gambar 4.23  
*Menu Side Category*

*e. Menu Side Product Best Seller*

HEADER						
HOME	PROFILE	PRODUCT	HOW TO BUY	SHOPPING CART	CONTACT ME	STATUS ORDERS
Jam					Searching	
TEXT						
PAGE MENU UTAMA					GO	
					Keraniang	
					Category	
					Product Best Seller	
					Shout Box	
					Find Me On	
					Guest Book	
TEXT						
FOOTER						

Gambar 4.24

*Menu Side Product Best Seller*

*f. Menu Side Shout Box*

HEADER						
HOME	PROFILE	PRODUCT	HOW TO BUY	SHOPPING CART	CONTACT ME	STATUS ORDERS
<div>Jam</div> <div>TEXT</div> <div>PAGE MENU UTAMA</div> <div>TEXT</div>				<div>Searching</div> <div></div> <div>GO</div> <div>Keraniang</div> <div>Category</div> <div>PRODUCT BEST SELLER</div> <div>Shout Box</div> <div>—</div> <div>=====</div> <div>Nama <div></div></div> <div></div> <div>Kirim Refresh</div> <div>Find Me On</div> <div>Guest Book</div>		
FOOTER						

Gambar 4.25

*Menu Side Shout Box*

g. *Menu Side Find Me On*

HEADER						
HOME	PROFILE	PRODUCT	HOW TO BUY	SHOPPING CART	CONTACT ME	STATUS ORDERS
Jam					Searching	
TEXT					GO	
		Keraniang				
PAGE MENU UTAMA					Category	
					PRODUCT BEST SELLER	
					Shout Box	
					Find Me On	
					Banner	
					Banner	
					Guest Book	
TEXT						
FOOTER						

Gambar 4.26

*Menu Side Find Me On*



*h. Menu Side Guest Book*

HEADER						
HOME	PROFILE	PRODUCT	HOW TO BUY	SHOPPING CART	CONTACT ME	STATUS ORDERS
Jam					Searching	
TEXT					<input type="text"/>	
					GO	
		Keraniang				
PAGE MENU UTAMA					Category	
					PRODUCT BEST SELLER	
					Shout Box	
					Find Me On	
					Guest Book	
					<input type="text"/> <input type="text"/> <input type="text"/> <input type="text"/>	
					Buku Tamu	
					<input type="text"/> <input type="text"/> <input type="text"/> <input type="text"/> <input type="text"/>	
					<input type="text"/>	
					Send	
					Reset	
TEXT						
FOOTER						

Gambar 4.27  
*Menu Side Guest Book*

## **4.6 Perancangan Teknologi**

### **4.6.1 Kebutuhan Perangkat Keras *Developer***

Disini penulis akan memberi keterangan perangkat keras yang digunakan selama pembuatan aplikasi tersebut, berikut keterangannya :

- a. Processor : Intel Core I3
- b. Memory : 3 GB
- c. VGA : 256 MB
- d. HardDisk : 320 GB
- e. Monitor : 14"
- f. Mouse & Keyboard

### **4.6.2 Pengembangan Aplikasi**

Pada pengembangan aplikasi ini, penulis menggunakan *localhost* untuk mengimplementasikannya dan juga berencana untuk menghubungkannya dengan internet. Untuk menghubungkan aplikasi ke internet, dibutuhkannya sebuah *domain* untuk mendaftar *web hosting* baik itu *domain* gratis maupun *domain* berbayar. Dan untuk menjalankan aplikasi secara nyata, penulis berniat untuk mencobanya dengan *domain* berbayar dengan jangka waktu 12 bulan.

Aplikasi *web* ini dibuat dengan tujuan untuk menjual dan mempromosikan barang dagangan yaitu *pc game*, konsumen juga dapat memesan *game* dengan cara menghubungi *user* di konten *contact me*. Aplikasi ini juga memudahkan *user* dalam mengolah data karena data akan tersimpan dalam *database*.

Dengan dibuatnya aplikasi ini penjualan dapat berjalan lebih lancar lagi dari media yang sebelumnya yaitu *twitter* dan *kaskus*, karena aplikasi dapat di akses dengan mudah dan lebih dapat dikontrol dengan lebih baik.

Untuk membuat aplikasi ini penulis mendapatkan referensi-referensi dari buku, internet dan juga hasil bertanya kepada orang yang memahami tentang pembuatan aplikasi *web*. Tidak hanya bertanya tetapi penulis juga mendiskusikannya.

Aplikasi ini dibuat dengan tujuan sasaran pengguna yaitu siapa saja, baik anak-anak, remaja, bahkan orang dewasa sekalipun. Karena *game* dapat dimainkan oleh semua kalangan dan tentunya game dapat menjadi sebuah sarana untuk menghibur diri ketika jenuh.

#### **4.6.3 Perangkat Keras dan Lunak Untuk Menjalankan Aplikasi**

Perangkat keras yang dibutuhkan untuk menjalankan aplikasi tersebut secara online adalah sebagai berikut:

- a. Komputer sebagai alat untuk mengoperasikan aplikasi.
- b. Modem sebagai alat untuk menghubungkan ke internet.
- c. *Browser* sebagai media untuk terhubung ke internet.

## **BAB V**

### **IMPLEMENTASI**

#### **5.1 Analisis Kebutuhan Informasi**

##### **5.1.1 Analisis Kebutuhan Koneksi**

Disini penulis menggunakan localhost untuk mengimplementasikan aplikasi *web* ini, tetapi penulis juga menghubungkan antara aplikasi dengan *domain* berbayar yang berjangka waktu selama 12 bulan. Pada saat implementasi penulis menggunakan domain berbayar ini, karena untuk mencoba menjalankan aplikasi secara nyata dan menguji apakah aplikasi jalan pada saat *running* dengan internet. Sebelumnya pada saat mencoba mendaftar ke *domain* berbayar, penulis mencobanya terlebih dahulu di localhost yaitu implementasi sementara pada *browser*. Berikut adalah beberapa fitur dari domain berbayar tersebut :

- a. Kapasitas 2 GB
- b. Anti spam
- c. SEO(Search Engine Optimization)
- d. Security
- e. Dll.

##### **5.1.2 Syarat Minimum Spesifikasi**

Untuk menjalankan aplikasi ini dapat menggunakan spesifikasi minimum pada komputer anda, berikut adalah spesifikasi minimum untuk menjalankan aplikasi dan spesifikasi ini diambil dari spesifikasi yang biasanya dipakai pada warnet :

- a. Prosesor : intel Pentium E2180 Core Duo 2 Ghz
- b. Motherboard : soket LGA 775, dapat menggunakan seri intel DG31GL/B
- c. Memori : DDR2 1 GB
- d. VGA: on board

- e. Sound card : on board
- f. Hard disk : 160 GB
- g. Monitor : CRT 15"
- h. Keyboard + Mouse

### **5.1.3 Analisis Kebutuhan Perangkat Lunak**

Disini penulis akan menjabarkan perangkat lunak apa saja yang digunakan oleh penulis dalam membuat aplikasi, berikut adalah *software-software* yang digunakan oleh penulis :

- a. Sistem Operasi Windows 7 Home Premium 64bit
- b. XAMPP version 3.4.10.1 atau versi lebih baru
- c. Macromedia Dreamweaver 8
- d. Paint/Macromedia Fireworks 8
- e. Browser (Firefox)

## **5.2 Implementasi Antarmuka**

Implementasi antarmuka pada Aplikasi Penjualan dan Pemesanan *PC Game* adalah sebagai berikut:

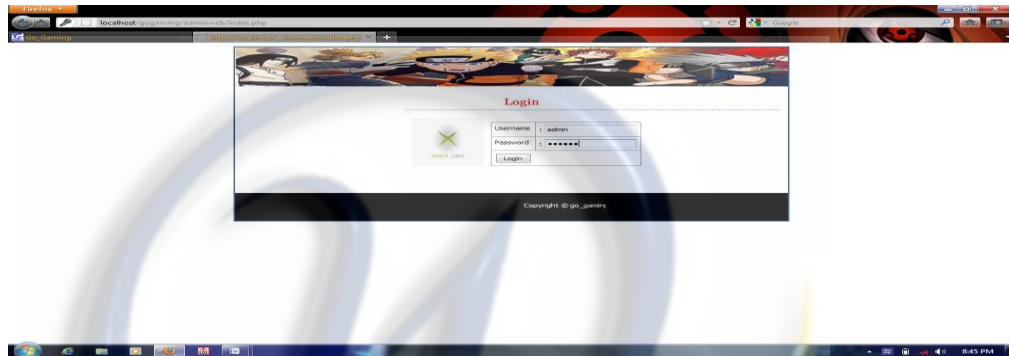
### **5.2.1 Antarmuka Form**

Pada saat program dijalankan(*running*), akan ditampilkan halaman utama yang menampilkan menu-menu yang dapat diakses oleh *user* yaitu menu admin dan konsumen yaitu *web* dari *go\_gaming*. Pada menu *user* terdapat beberapa konten yang dimana *user* mempunyai hak untuk mengolah data dan sebelum masuk, *user* harus login terlebih dahulu dengan cara memasukkan *username* dan *password*, sedangkan pada bagian konsumen yaitu terdapat menu-menu yang dapat di akses oleh konsumen seperti *Home*, *Profile*, *Product*, *How To Buy*, *Shopping Cart*, *Contact Me* dan menu

side yang berada disebelah kanan tampilan. Berikut adalah antarmuka dari Aplikasi Penjualan dan Pemesanan *PC Game* :

**a. Login User**

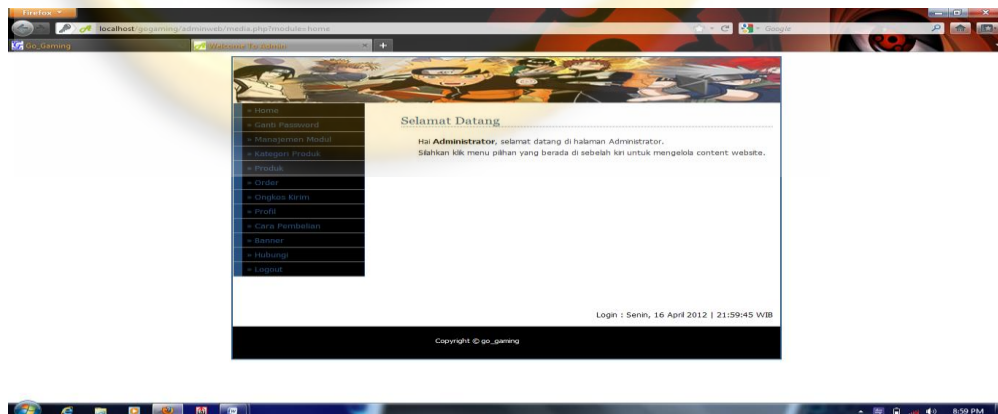
*Login User* tampilan awal dimana user harus memasukkan *username* dan *password* terlebih dahulu untuk mengakses *menu user*.



Gambar 5.1  
*Login User*

**b. Menu User**

*Menu User* ini berisikan konten-konten/data-data yang dapat diolah oleh *user*.



Gambar 5.2  
*Menu User*

c. *Menu Header*

Menu Header adalah konten-konten bagian header(*Home, Profile, Product, How To Buy, Shopping Cart, Contact Me, Status Orders*)



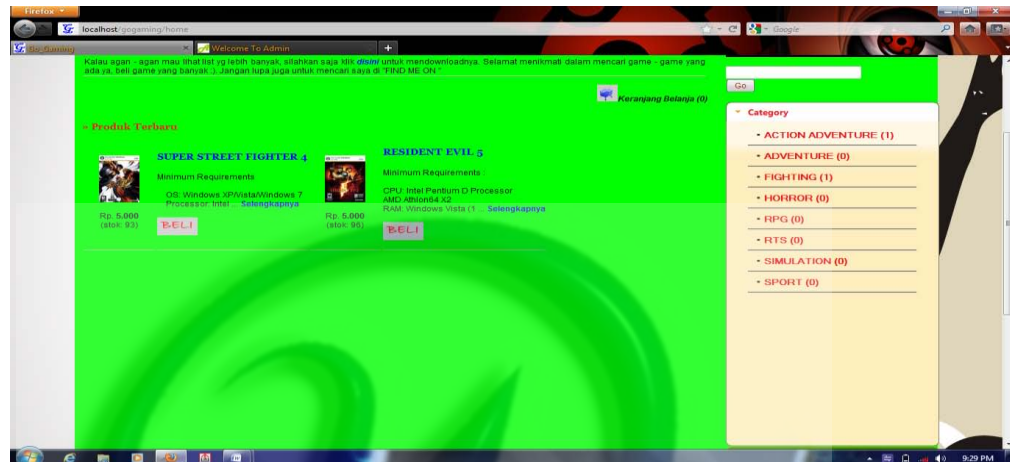
Gambar 5.3

*Menu Header*



**d. Menu Side Category**

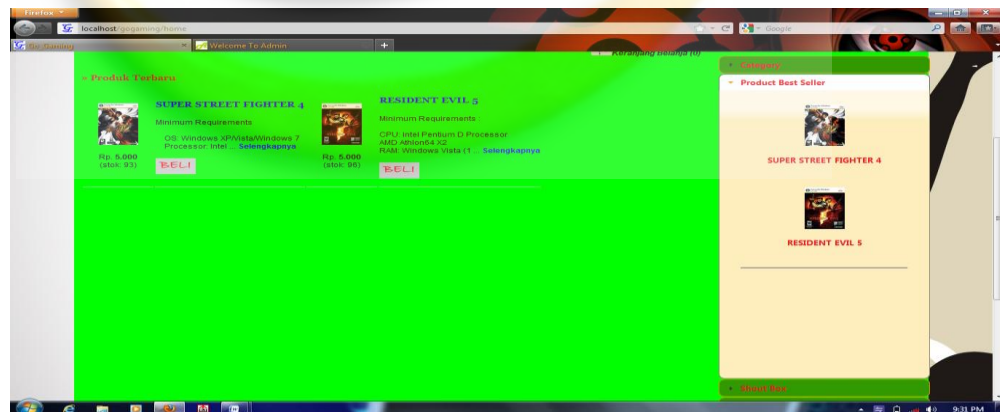
*Menu Side Category* merupakan produk yang dikelompokkan sesuai kategorinya.



Gambar 5.4  
*Menu Side Category*

**e. Menu Side Product Best Seller**

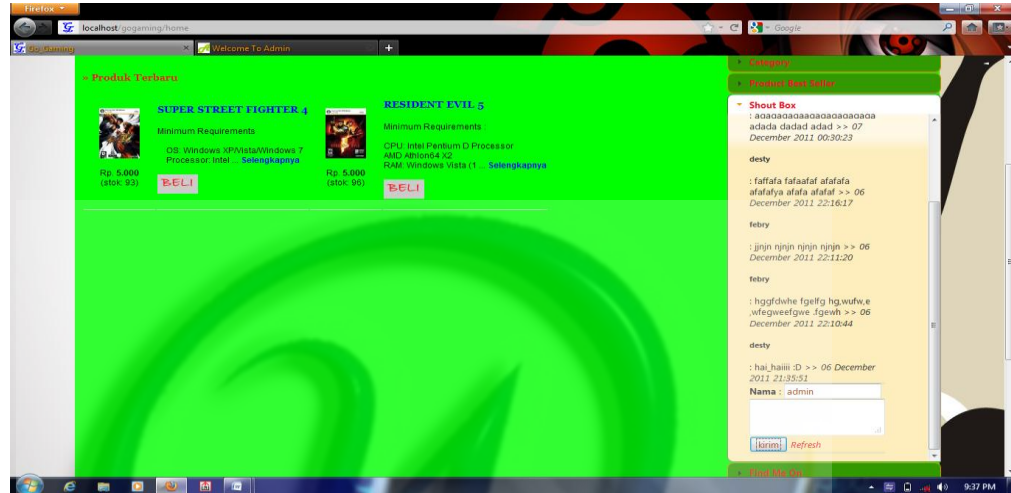
*Menu Side Product Best Seller* adalah produk-produk yang laris dibeli oleh konsumen.



Gambar 5.5  
*Menu Side Product Best Seller*

**f. Menu Side Shout Box**

*Menu Side Shout Box* adalah percakapan atau komen dari konsumen dan pengunjung *web go\_gaming*

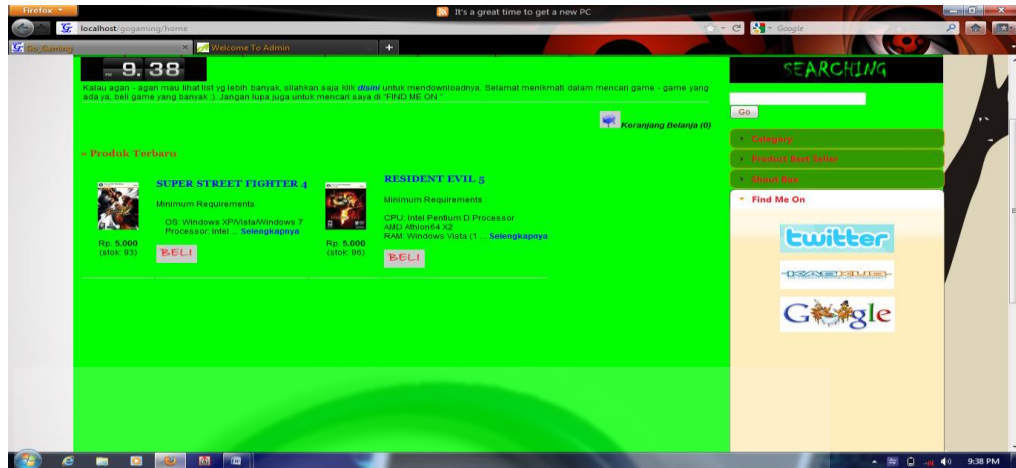


Gambar 5.6

*Menu Side Shout Box*

**g. Menu Side Find Me On**

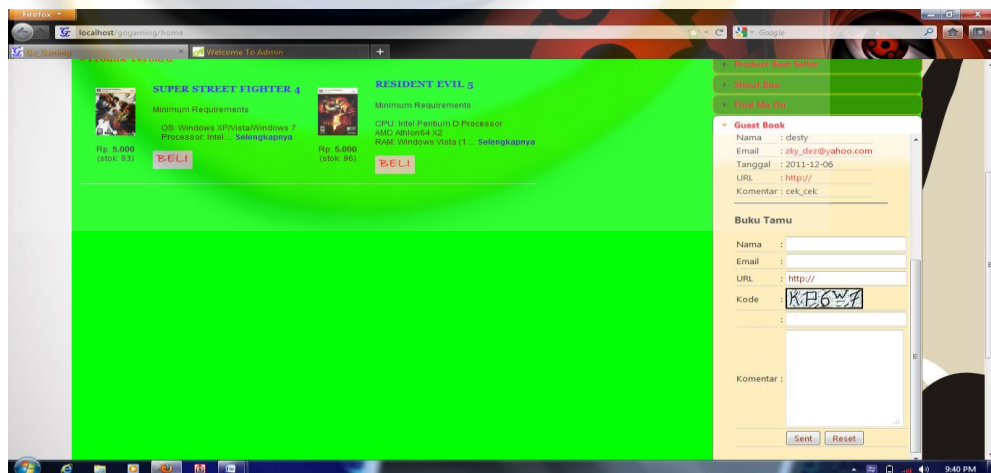
*Menu Side Find Me On* adalah link untuk menuju ke tempat lain *go\_gaming* berjualan (*Kaskus* dan *Twitter*) dan link-link lainnya, konsumen hanya mengklik gambar yang ada untuk terhubung dengan link tersebut.



Gambar 5.7  
*Menu Side Find Me On*

**h. Menu Side Guest book**

*Menu Side Guest book* merupakan buku tamu yang dapat diisi oleh konsumen, pada buku tamu ini konsumen bisa menuliskan *email*, *url* dan komentar dari konsumen.



Gambar 5.8  
*Menu Side Guest book*

### 5.3 Informasi Semua Menu Setelah Dijalankan

Tabel 5.1

Informasi Semua Menu Setelah Dijalankan

No	Nama Menu Yang Diuji	Yang diharapkan	Teruji		keterangan
			Y	T	
1	Login User	User dapat login ke menu user	Y		User berhasil masuk ke menu user
2	Menu User	User dapat mengelola konten	Y		User mengelola konten user
3	Menu Header	Konsumen dapat mengakses menu header	Y		Konsumen mengakses menu header
4	Menu Side Category	Dapat menampilkan kategori game	Y		Game muncul sesuai kategori
5	Menu Side Product Best Seller	Dapat menampilkan produk terlaris	Y		Menampilkan produk terlaris
6	Menu Side Shout Box	Konsumen dapat melakukan perbincangan	Y		Konsumen melakukan perbincangan
7	Menu Side Find Me On	Menghubungkan dengan kaskus dan twitter user	Y		Terhubung ketika konsumen

					mengklik banner
8	Menu Side Guest book	Konsumen mengisi buku tamu	Y		Data konsumen di buku tamu tercatat
9	Menu Status Orders	Untuk menampilkan status order konsumen yang Baru dan Lunas	Y		Tampil status orders konsumen yang Baru dan Lunas

Keterangan :

1. Y = YA
2. T = TIDAK

## **BAB VI**

### **KESIMPULAN DAN SARAN**

Berdasarkan hasil pengembangan sistem yang telah penulis lakukan, penulis mencoba membuat suatu kesimpulan dan mengajukan beberapa saran-saran yang berhubungan dengan pembahasan yang telah dikemukakan di bab-bab sebelumnya

#### **6.1 Kesimpulan**

Berdasarkan Sistem Informasi yang telah dibuat mengenai “**Aplikasi Penjualan PC Game Secara Online (Studi Kasus : Penjualan PC Game Di Go\_Gaming)**” dapat diperoleh beberapa kesimpulan sebagai berikut :

1. Dibutuhkannya sebuah aplikasi berbasis *web* yang terhubung dengan internet, agar *user* dapat menjual pc game secara *online* dan luas kepada masyarakat.
2. Dibutuhkannya data-data yang diperlukan dengan cara mencari data secara langsung kepada user dan bertanya apa yang *user* inginkan pada aplikasi.
3. *Game* dibuatkan dengan kategori-kategori sesuai dengan jenisnya.

#### **6.2 Saran**

Berdasarkan hasil analisa perangkat lunak yang telah dibuat masih jauh dari kesempurnaan, untuk peningkatan dan pengembangan dari perangkat lunak ini, agar hasil yang diperoleh jauh lebih baik, maka ada baiknya diberikan saran-saran sebagai berikut :

1. Mendiskripsikan *game* secara lebih detail dan seefisien mungkin untuk mempermudah konsumen dalam mencari info tentang *game* yang dicarinya.
2. Sering dilakukannya *backup database* untuk menghindari terjadinya hal-hal yang tidak diinginkan.

3. Untuk kedepannya dibuatkan jenis transaksi secara *online* seperti *paypal* atau yang lainnya.





## LAMPIRAN A

### A. Koneksi.php

```
<?php
$server = "localhost";
$username = "root";
$password = "";
$database = "gogaming";

// Koneksi dan memilih database di server
mysql_connect($server,$username,$password) or die("Koneksi gagal");
mysql_select_db($database) or die("Database tidak bisa dibuka");

// nama folder untuk upload file video
$uploadaddir = 'data/';

?>
```

### B. Media.php

```
<?php
error_reporting(0);
session_start();
include "config/koneksi.php";

    include "config/fungsi_indotgl.php";
    include "config/class_paging.php";
    include "config/fungsi_combobox.php";
    include "config/library.php";
include "config/fungsi_autolink.php";
include "config/fungsi_rupiah.php";

?>

<!DOCTYPE html PUBLIC "-//W3C//DTD XHTML 1.0 Transitional//EN" "http://www.w3.org/TR/xhtml1/DTD/xhtml1-transitional.dtd">
<html xmlns="http://www.w3.org/1999/xhtml"><head>

<script type="text/javascript" src="jquery.tools.min.js"></script>

<script>

    flashembed("jam", "images/flip_clock_black_12.swf");

</script>

<style>
```

```
#jam {
```

```
width: 20%;
```

```
height: 20%;
```

```
}
```

```
</style>
```

```
<title><?php include "dina_titel.php"; ?></title>
```

```
</script>
```

```
<meta http-equiv="Content-Type" content="text/html; charset=utf-8" />
```

```
<meta name="robots" content="index, follow">
```

```
<meta name="description" content="<?php include "dina_meta1.php"; ?>">
```

```
<meta name="keywords" content="<?php include "dina_meta2.php"; ?>">
```

```
<meta name="author" content="Lukmanul Hakim">
```

```
<meta http-equiv="imagetoolbar" content="no">
```

```
<meta name="language" content="Indonesia">
```

```
<meta name="revisit-after" content="7">
```

```
<meta name="webcrawlers" content="all">
```

```
<meta name="rating" content="general">
```

```
<meta name="spiders" content="all">
```

```
<link href="style.css" rel="stylesheet" type="text/css" />
```

```
<style type="text/css">
```

```
<!--
```

```
.style1 {color: #FF0000}
```

```
.style2 {color: #0000FF}
```

```
.style4 {color: #993399}
```

```
.style5 {font-size: 12px}
```

```
.style6 {
```

```
        color: #F0F0F0;

        font-size: 14px;

    }

    .style7 {font-size: 16px}

-->

</style>

</head>

</html>
```

```
<html xmlns="http://www.w3.org/1999/xhtml">

<head>

<link rel="shortcut icon" href="G.ico" />

</head>

<body>

<div id="wrapper">

    <div id="header" align="justify"></div>

    <div id="menuutama" align="center" style="background: #000000>

        <td height="100%">

            <ul>

                <li><a href="index.php">Home</a></li>

                <li><a href="profil-kami.html">Profile</a></li>

                <li><a href="semua-produk.html">Product</a></li>

                <li><a href="cara-pembelian.html">How To Buy</a></li>

                <li><a href="keranjang-belanja.html">Shopping Cart</a></li>

                <li><a href="hubungi-kami.html">Contact Me</a></li>

                <li><a href="statorders.html">Status Orders</a></li>

            </ul>

        </div>

        <div id="leftcontent">

            <p> </p>

        </div>
```

```
<div id="middlecontent">
```

```
<img src=images/pencarian.jpg width="288px"><br />
```

```
<br />
```

```
<form method=POST action='hasil-pencarian.html'>
```

```
<input name=kata type=text size=30 maxlength=40 />
```

```
<input type=submit value=Go />
```

```
</form><br />
```

```
<?php include "coba.php"; ?>
```

```
</div>
```

```
<div id="rightcontent" align="justify">
```

```
<div id="jam" align="left"></div>
```

```
<table width="100%" border="0">
```

```
<tr>
```

```
<td width="50%"><p>Kalau agan - agan mau lihat list yg lebih banyak, silahkan saja klik<a href="http://www.mediafire.com/?0afm0mardiy2r6q"><em> disini </em></a> untuk mendownloadnya. Selamat menikmati dalam mencari game - game yang ada ya, beli game yang banyak :). Jangan lupa juga untuk mencari saya di "FIND ME ON " </p>
```

```
<p>*Jam Operasinal : Senin s/d Sabtu pukul 08.00-16.00, selain dari waktu dan jam tersebut pelayanan tidak berjalan :D </p></td>
```

```
</tr>
```

```
</table>
```

```
<p>
```

```
<?php include "kanan.php"; ?>
```

```
</p>
```

```
</div>
```

```
<div id="clearer" align="center">
```

```
<p>&nbsp;</p>
```

```
<p>&nbsp;</p>
```

```
<p>&nbsp;</p>
```

```
<p>&nbsp;</p>
```

<p>&nbsp;</p>

<p>&nbsp;</p>

<p>&nbsp;</p>

<p>

<span class="style1">Untuk cek Spesifikasi game bisa kunjungi :</span></br>

<br>

<a href="http://www.systemrequirementslab.com/cyri/">http://www.systemrequirementslab.com/cyri/</a><br>

<a href="http://www.specrequirements.com/">http://www.specrequirements.com/</a></br>

</br>

<em></em>

</marquee>

</br>

</p>

</div>

<div id="footer"><p><a href="index.php" class="style5 style6">Home</a>

<a href="profil-kami.html" class="style5 style6">Profile</a>

<a href="semua-produk.html" class="style5 style6">Product</a>

<a href="cara-pembelian.html" class="style5 style6">How To Buy</a>

<a href="keranjang-belanja.html" class="style5 style6">Shopping Cart</a>

<a href="hubungi-kami.html" class="style5 style6">Contact Me</a>

<a href="statorders-kosumen.html" class="style5 style6">Status Order</a>

<p><marquee direction="right" scrollamount="6" height="30" width="50%">

<span class="style5"> Copyright &copy; go\_gaming</span>

</marquee></div>

</div>

</div>

</form>

</body>

</html>

### C. Kanan.php

```
<script language="javascript">
function validasi(form){
    if (form.nama.value == ""){
        alert("Anda belum mengisi Nama.");
        form.nama.focus();
        return (false);
    }
    if (form.alamat.value == ""){
        alert("Anda belum mengisi Alamat.");
        form.alamat.focus();
        return (false);
    }
    if (form.telpon.value == ""){
        alert("Anda belum mengisi Telpon.");
        form.telpon.focus();
        return (false);
    }
    if (form.email.value == ""){
        alert("Anda belum mengisi Email.");
        form.email.focus();
        return (false);
    }
    if (form.kota.value == 0){
        alert("Anda belum mengisi Kota.");
        form.kota.focus();
        return (false);
    }
    return (true);
}

function harusangka(jumlah){
    var karakter = (jumlah.which) ? jumlah.which : event.keyCode
    if (karakter > 31 && (karakter < 48 || karakter > 57))
        return false;

    return true;
}
</script>

<?php
// Halaman utama (Home)
if ($_GET[module]=='home'){
    // Tampilkan berapa item yang sudah dimasukkan ke Keranjang Belanja
    require_once "item.php";

    echo "<br/><span class=judul>&#187; <b>Produk Terbaru</b></span><br/>";
    // Tampilkan 6 produk terbaru
    $sql=mysql_query("SELECT * FROM produk ORDER BY id_produk DESC LIMIT 6");

    $kolom = 2;
    echo "<table><tr>";
```

```

$si=0;
while ($r=mysql_fetch_array($sql)){
    $harga = number_format($r[harga],0,"",".");
    $deskripsi = nl2br($r[deskripsi]); // membuat paragraf
    $isi = substr($deskripsi,0,120); // ambil sebanyak 120 karakter
    $isi = substr($deskripsi,0,strpos($isi, " ")); // potong per spasi kalimat
    if ($i >= $kolom){
        echo "</tr><tr>";
        $si=0;
    }
    $i++;
    echo "<td align=center><br><img src='foto_produk/small_{$r[gambar]}' hspace=20 border=0><br /><br />Rp. <b>{$harga}</b> <br />
    (stok: {$r[stok]}<br /><br /></td>
        <td><br /><span class=judul><a href=produk-{$r[id_produk]-{$r[produk_seo]}.html>{$r[nama_produk]}</a></span><br /><br />
        $isi ... <a href=produk-{$r[id_produk]-{$r[produk_seo]}.html>Selengkapnya</a><br /><br />
        <a href=aksi.php?module=keranjang&act=tambah&id={$r[id_produk]}><img src='images/beli.jpg' border=0></a><br /><br />
    </td>";
    }
    echo "</tr></table><br />";
}

```

```

// Modul detail produk
elseif ($_GET[module]=='detailproduk'){
    // Tampilkan berapa item yang sudah dimasukkan ke Keranjang Belanja
    require_once "item.php";

    // Tampilkan detail produk berdasarkan produk yang dipilih
    $detail=mysql_query("SELECT * FROM produk,kategori
        WHERE kategori.id_kategori=produk.id_kategori
        AND id_produk='{$_GET[id]}'");
    $d = mysql_fetch_array($detail);
    $tgl = tgl_indo($d[tanggal]);
    $deskripsi = nl2br($d[deskripsi]); // membuat paragraf pada isi berita
    $harga = number_format($d[harga],0,"",".");

    echo "<span class=date>$tgl</span><br />";
    echo "<span class=judul>$d[nama_produk]</span><br />";
    echo "Kategori: <a href=kategori-{$d[id_kategori]-{$d[kategori_seo]}.html><b>$d[nama_kategori]</b></a></span><br />";
    // Apabila ada gambar dalam berita, tampilkan
    if ($d[gambar]!=''){
        echo "<span class=image><img src='foto_produk/{$d[gambar]}' border=0></span>";
    }
    echo "$deskripsi <br />Rp. <b>{$harga}</b> (stok: {$d[stok]}<br /><br />
    <a href=aksi.php?module=keranjang&act=tambah&id={$d[id_produk]}><img src='images/beli.jpg' border=0></a><br />";
}

```

```

// Modul produk per kategori
elseif ($_GET[module]=='detailkategori'){
    // Tampilkan berapa item yang sudah dimasukkan ke Keranjang Belanja
    require_once "item.php";

    // Tampilkan nama kategori
    $sq = mysql_query("SELECT nama_kategori from kategori where id_kategori='{$_GET[id]}'");
}

```



```

$N = mysql_fetch_array($sq);
echo "<span class=judul_head>&#187; Kategori : <b>$N[nama_kategori]</b></span><br />";

$P = new Paging3;
$batas = 5;
$posisi = $P->cariPosisi($batas);

// Tampilkan daftar produk yang sesuai dengan kategori yang dipilih
$sql = "SELECT * FROM produk WHERE id_kategori='$_GET[id]'
ORDER BY id_produk DESC LIMIT $posisi,$batas";
$hasil = mysql_query($sql);
$jumlah = mysql_num_rows($hasil);

// Apabila ditemukan produk dalam kategori
if ($jumlah > 0){
$kolom = 2;
echo "<table><tr>";

$ii=0;
while($r=mysql_fetch_array($hasil)){
$harga = number_format($r[harga],0,"",".");
// Tampilkan hanya sebagian isi berita
$isi_produk = nl2br($r[deskripsi]); // membuat paragraf pada isi berita
$isi = substr($isi_produk,0,120); // ambil sebanyak 120 karakter
$isi = substr($isi_produk,0,strrpos($isi," ")); // potong per spasi kalimat

if ($i >= $kolom){
echo "</tr><tr>";
$ii=0;
}
$ii++;
echo "<td align=center><br><img src='foto_produk/small_{$r[gambar]}' hspace=20 border=0><br /><br />Rp. <b>$harga</b> <br
/>(stok: {$r[stok]})<br /><br /></td>
<td><br /><span class=judul><a href=produk-{$r[id_produk]}-{$r[produk_seo].html>{$r[nama_produk]</a></span><br /><br />
$isi ... <a href=produk-{$r[id_produk]}-{$r[produk_seo].html>Selengkapnya</a><br /><br />
<a href=aksi.php?module=keranjang&act=tambah&id={$r[id_produk]}><img src='images/beli.jpg' border=0></a><br /><br
/></td>";
}
echo "</tr></table><br />";

$jmldata = mysql_num_rows(mysql_query("SELECT * FROM produk WHERE id_kategori='$_GET[id]'"));
$jmldata = $P->jumlahHalaman($jmldata, $batas);
$linkHalaman = $P->navHalaman($_GET[halkategori], $jmldata);

echo "Hal: $linkHalaman<br /><br />";
}
else{
echo "<p align=center>Belum ada produk pada kategori ini.</p>";
}
}

// Menu utama di header

// Modul profil
elseif ($_GET[module]=='profilkami'){
// Tampilkan berapa item yang sudah dimasukkan ke Keranjang Belanja
require_once "item.php";

```

```

echo "<span class=judul>&#187; <b>Profil</b></span><br /><br />";

// Data profil mengacu pada id_modul=43
$profil = mysql_query("SELECT * FROM modul WHERE id_modul='43'");
$r = mysql_fetch_array($profil);

if ($r[gambar]!=''){
    echo "<span class=image><img src='images/$r[gambar]'"></span>";
}
$isi_profil=nl2br($r[static_content]);
echo "$isi_profil";
}

// Modul cara pembelian
elseif ($_GET[module]=='carabeli'){
    // Tampilkan berapa item yang sudah dimasukkan ke Keranjang Belanja
    require_once "item.php";

    echo "<span class=judul>&#187; <b>Cara Pembelian</b></span><br /><br />";

    // Data cara pembelian mengacu pada id_modul=45
    $profil = mysql_query("SELECT * FROM modul WHERE id_modul='45'");
    $r = mysql_fetch_array($profil);

    $isi_carabeli=nl2br($r[static_content]);
    echo "$isi_carabeli";
}

// Modul semua produk
elseif ($_GET[module]=='semuaproduk'){
    // Tampilkan berapa item yang sudah dimasukkan ke Keranjang Belanja
    require_once "item.php";

    // Tampilkan semua produk
    echo "<span class=judul>&#187; <b>Produk</b></span><br /><br />";

    $p = new Paging2;
    $batas = 3;
    $posisi = $p->cariPosisi($batas);

    // Tampilkan semua produk
    $sql=mysql_query("SELECT * FROM produk ORDER BY id_produk DESC LIMIT $posisi,$batas");
    while($r=mysql_fetch_array($sql)){
        $harga = number_format($r[harga],0,"",".");
        echo "<table><tr><td><span class=judul><a href=produk-$r[id_produk]-
$r[produk_seo].html>$r[nama_produk]</a></span><br />";
        // Apabila ada gambar dalam produk, tampilkan
        if ($r[gambar]!=''){
            echo "<span class=image><img src='foto_produk/small_$r[gambar]' border=0></span>";
        }
        // Tampilkan hanya sebagian isi berita
        $isi_produk = nl2br($r[deskripsi]); // membuat paragraf pada isi berita
        $isi = substr($isi_produk,0,300); // ambil sebanyak 300 karakter
        $isi = substr($isi_produk,0, strrpos($isi, " ")); // potong per spasi kalimat
    }
}

```

```

        echo "$isi ... <a href=produk-{$r[id_produk]-{$r[produk_seo].html}>Selengkapnya</a><br /><br />
        Rp. <b>{$harga}</b> (stok: {$r[stok]}<br /><a href=aksi.php?module=keranjang&act=tambah&id={$r[id_produk]}><img
src='images/beli.jpg' border=0></a><br />
        </td></tr></table><br />";
    }

    $jmldata = mysql_num_rows(mysql_query("SELECT * FROM produk"));
    $jmlhalaman = $p->jumlahHalaman($jmldata, $batas);
    $linkHalaman = $p->navHalaman($_GET[hal_produk], $jmlhalaman);

    echo "Hal: $linkHalaman<br /><br />";
}

// Modul keranjang belanja
elseif ($_GET[module]=='keranjangbelanja'){
    // Tampilkan berapa item yang sudah dimasukkan ke Keranjang Belanja
    require_once "item.php";

    // Tampilkan produk-produk yang telah dimasukkan ke keranjang belanja
    echo "<span class=judul>&#187; <b>Keranjang Belanja</b></span><br /><br />";
        $sid = session_id();
        $sql = mysql_query("SELECT * FROM orders_temp, produk
                                WHERE id_session='{$sid}' AND orders_temp.id_produk=produk.id_produk");
    $sketemu=mysql_num_rows($sql);
    if($sketemu < 1){
        echo "<script>window.alert('Keranjang Belanjanya Masih Kosong');
        window.location=('index.php')</script>";
    }
    else{
        echo "<form method=post action=aksi.php?module=keranjang&act=update>
        <table border=0 cellpadding=3 align=center>
        <tr bgcolor=#D3DCE3><th>No</th><th>Produk</th><th>Nama Produk</th><th>Berat(Kg)</th><th>Jumlah</th>
        <th>Harga</th><th>Sub Total</th><th>Hapus</th></tr>";

        $no=1;
        while($r=mysql_fetch_array($sql)){
            $subtotal = $r[harga] * $r[jumlah];
            $total = $total + $subtotal;
            $subtotal_rp = format_rupiah($subtotal);
            $total_rp = format_rupiah($total);
            $harga = format_rupiah($r[harga]);

            echo "<tr bgcolor=#cccccc><td>$no</td><input type=hidden name=id[$no] value={$r[id_orders_temp]}>
            <td align=center><br><img src=foto_produk/small_{$r[gambar]}></td>
            <td>{$r[nama_produk]}</td>
            <td align=center>{$r[berat]}</td>
            <td><input type=text name='jml[$no]' value={$r[jumlah]} size=1 onkeypress=\"return harusangka(event)\"></td>
            <td>{$harga}</td>
            <td>{$subtotal_rp}</td>
            <td align=center><a href='aksi.php?module=keranjang&act=hapus&id={$r[id_orders_temp]}'>
            <img src=images/kali.png border=0 title=Hapus></a></td>
            </tr>";
            $no++;
        }
        echo "<tr><td colspan=6 align=right><br><b>Total</b>:</td><td colspan=2><br>Rp. <b>{$total_rp}</b></td></tr>
        <tr><td colspan=2><br /><a href=javascript:history.go(-1)><img src=images/lanjutkan.jpg border=0></a><br /></td>";
    }

```

```

        <td colspan=2><br /><input type=image src='images/update.jpg' border=0><br /></td>
        <td colspan=4 align=right><br /><a href=selesai-belanja.html><img src=images/selesai.jpg border=0></a><br /></td></tr>
    </table></form><br />";
    echo "**) Apabila anda mengubah jumlah, setelah input data pada jumlah, tekan tombol <b>Update Keranjang</b>."
    "**) Total harga diatas belum termasuk ongkos kirim yang akan dihitung saat <b>Selesai Belanja</b>.<br /><br />";
}
}

// Modul hubungi kami
elseif ($_GET[module]=='hubungikami'){
    // Tampilkan berapa item yang sudah dimasukkan ke Keranjang Belanja
    require_once "item.php";

    echo "<span class=judul>&#187; <b>Hubungi Kami</b></span><br /><br />";
    echo "<b>Hubungi kami secara online dengan mengisi form dibawah ini:</b>";
    <table width=100% style='border: 1pt dashed #0000CC;padding: 10px;'>
    <form action=hubungi-aksi.html method=POST>
    <tr><td>Nama</td><td> : <input type=text name=nama size=40></td></tr>
    <tr><td>Email</td><td> : <input type=text name=email size=40></td></tr>
    <tr><td>Subjek</td><td> : <input type=text name=subjek size=55></td></tr>
    <tr><td valign=top>Pesan</td><td> <textarea name=pesan style='width: 315px; height: 100px;'></textarea></td></tr>
    </td><td colspan=2><input type=submit name=submit value=Kirim></td></tr>
    </form></table><br />";
}

// Modul hubungi aksi
elseif ($_GET[module]=='hubungiaksi'){
    mysql_query("INSERT INTO hubungi(nama,
        email,
        subjek,
        pesan,
        tanggal)
        VALUES('$_POST[nama]',
        '$_POST[email]',
        '$_POST[subjek]',
        '$_POST[pesan]',
        '$tgl_sekarang')");
    echo "<span class=posting>&#187; <b>Hubungi Kami</b></span><br /><br />";
    echo "<p align=center><b>Terimakasih telah menghubungi kami. <br /> Kami akan segera meresponnya.</b></p>";
}

// Modul hasil pencarian produk
elseif ($_GET['module']=='hasilcari'){
    echo "<span class=judul_head>&#187; <b>Hasil Pencarian</b></span><br />";
    // menghilangkan spasi di kiri dan kanannya
    $kata = trim($_POST['kata']);
    // mencegah XSS
    $kata = htmlentities(htmlspecialchars($kata), ENT_QUOTES);

    // pisahkan kata per kalimat lalu hitung jumlah kata
    $pisah_kata = explode(" ", $kata);
    $jml_katakan = (integer)count($pisah_kata);
    $jml_kata = $jml_katakan-1;

    $cari = "SELECT * FROM produk WHERE ";

```

```

for ($i=0; $i<=$jml_kata; $i++){
    $cari .= "deskripsi LIKE '%$pisah_kata[$i]%' OR nama_produk LIKE '%$pisah_kata[$i]%'";
    if ($i < $jml_kata ){
        $cari .= " OR ";
    }
}
$cari .= " ORDER BY id_produk DESC LIMIT 7";
$hasil = mysql_query($cari);
$ketemu = mysql_num_rows($hasil);

if ($ketemu > 0){
    echo "<p>Ditemukan <b>$ketemu</b> produk dengan kata <font style='background-color:#00FFFF'><b>$kata</b></font> : </p>";
    while($t=mysql_fetch_array($hasil)){
        echo "<table><tr><td><span class=judul><a href=produk-{$t[id_produk]-
$t[produk_seo].html>{$t[nama_produk]}</a></span><br />";
        // Tampilkan hanya sebagian isi produk
        $isi_produk = htmlentities(strip_tags($t['deskripsi'])); // mengabaikan tag html
        $isi = substr($isi_produk,0,250); // ambil sebanyak 250 karakter
        $isi = substr($isi_produk,0,strrpos($isi, " ")); // potong per spasi kalimat

        echo "$isi ... <a href=produk-{$t[id_produk]}-{$t[produk_seo].html>Selengkapnya</a>
        <br /><br /></td></tr>
        </table>";
    }
}
else{
    echo "<p>Tidak ditemukan produk dengan kata <b>$kata</b></p>";
}
}

// Modul selesai belanja
elseif ($_GET[module]=='selesaibelanja'){
    // Tampilkan berapa item yang sudah dimasukkan ke Keranjang Belanja
    require_once "item.php";

    // Form untuk input data pembeli
    echo "<span class=judul_head>&#187; <b>Data Pembeli</b></span><br /><br />
    <form name='form' action=simpan-transaksi.html method=POST onSubmit=\"return validasi(this)\">
    <table>
    <tr><td>Nama</td><td> : <input type=text name=nama size=30></td></tr>
    <tr><td>Alamat Lengkap</td><td> : <input type=text name=alamat size=70></td></tr>
    <tr><td>Telpon/HP</td><td> : <input type=text name=telpon></td></tr>
    <tr><td>Email</td><td> : <input type=text name=email></td></tr>
    <tr><td valign=top>Kota Tujuan</td><td> :
    <select name='kota'>
    <option value=0 selected>- Pilih Kota -</option>";
    $stampil=mysql_query("SELECT * FROM kota ORDER BY urutan");
    while($r=mysql_fetch_array($stampil)){
        echo "<option value={$r[id_kota]}>{$r[nama_kota]}</option>";
    }
    echo "</select> <br /><br />*) Apabila tidak terdapat nama kota tujuan Anda, Segera hubungi Admin.
    <br />**) Ongkos kirim dihitung berdasarkan kota tujuan</td></tr>
    <tr><td colspan=2><input type=submit value=Proses></td></tr>
    </table>";
}

```

```

// Modul simpan transaksi
elseif ($_GET[module]=='simpantransaksi'){
$kar1=strstr($_POST[email], "@");
$kar2=strstr($_POST[email], ".");

if (empty($_POST[nama]) || empty($_POST[alamat]) || empty($_POST[telpon]) || empty($_POST[email]) || empty($_POST[kota])){
    echo "Data yang Anda isikan belum lengkap<br />
        <a href='selesai-belanja.html'><b>Ulangi Lagi</b>";
    }
elseif (!ereg("[a-z|A-Z]", $_POST[nama])){
    echo "Nama tidak boleh diisi dengan angka atau simbol.<br />
        <a href=javascript:history.go(-1)><b>Ulangi Lagi</b></a>";
    }
elseif (strlen($kar1)==0 OR strlen($kar2)==0){
    echo "Alamat email Anda tidak valid, mungkin kurang tanda titik (.) atau tanda @.<br />
        <a href=javascript:history.go(-1)><b>Ulangi Lagi</b></a>";
    }
else{
// fungsi untuk mendapatkan isi keranjang belanja
function isi_keranjang(){
    $sikeranjang = array();
    $sid = session_id();
    $sql = mysql_query("SELECT * FROM orders_temp WHERE id_session='$sid'");

    while ($r=mysql_fetch_array($sql)) {
        $sikeranjang[] = $r;
    }
    return $sikeranjang;
}

$ttl_skrng = date("Ymd");
$jam_skrng = date("H:i:s");

// simpan data pemesanan
mysql_query("INSERT INTO orders(nama_kustomer, alamat, telpon, email, ttl_order, jam_order, id_kota)
    VALUES('$_POST[nama]', '$_POST[alamat]', '$_POST[telpon]', '$_POST[email]', '$ttl_skrng', '$jam_skrng', '$_POST[kota]')");

// mendapatkan nomor orders
$id_orders=mysql_insert_id();

// panggil fungsi isi_keranjang dan hitung jumlah produk yang dipesan
$sikeranjang = isi_keranjang();
$jml      = count($sikeranjang);

// simpan data detail pemesanan
for ($i = 0; $i < $jml; $i++){
    mysql_query("INSERT INTO orders_detail(id_orders, id_produk, jumlah)
        VALUES('$id_orders', {$sikeranjang[$i]['id_produk']}, {$sikeranjang[$i]['jumlah']}");
}

// update/kurangi stok produk
for ($i = 0; $i < $jml; $i++) {
    mysql_query("UPDATE produk SET stok = stok - {$sikeranjang[$i]['jumlah']}
        WHERE id_produk = {$sikeranjang[$i]['id_produk']}");
}

```

```

// update/tambahkan produk yang dibeli (best seller)
for ($i = 0; $i < $jml; $i++) {
    mysql_query("UPDATE produk SET dibeli = dibeli + {$sikeranjang[$i]['jumlah']}
                WHERE id_produk = {$sikeranjang[$i]['id_produk']}");
}

// setelah data pemesanan tersimpan, hapus data pemesanan di tabel pemesanan sementara (orders_temp)
for ($i = 0; $i < $jml; $i++) {
    mysql_query("DELETE FROM orders_temp
                WHERE id_orders_temp = {$sikeranjang[$i]['id_orders_temp']}");
}

echo "<span class=judul_head>&#187; <b>Proses Transaksi Selesai</b></span><br /><br />
    Data pemesan beserta ordernya adalah sebagai berikut: <br />
    <table>
    <tr><td>Nama      </td><td> : <b>$_POST[nama]</b> </td></tr>
    <tr><td>Alamat Lengkap </td><td> : $_POST[alamat] </td></tr>
    <tr><td>Telpon    </td><td> : $_POST[telpon] </td></tr>
    <tr><td>E-mail    </td><td> : $_POST[email] </td></tr></table><hr /><br />

    Nomor Order: <b>$id_orders</b><br /><br />";

    $daftarproduk=mysql_query("SELECT * FROM orders_detail,produk
        WHERE orders_detail.id_produk=produk.id_produk
        AND id_orders='$id_orders'");

echo "<table cellpadding=5>
    <tr bgcolor=#D3DCE3><th>No</th><th>Nama Produk</th><th>Berat(Kg)</th><th>Jumlah</th><th>Harga</th><th>Sub
    Total</th></tr>";

$pesan="Terimakasih telah melakukan pemesanan online di GO_GAMING <br /><br />
    Nama: $_POST[nama] <br />
    Alamat: $_POST[alamat] <br />
    Telpon: $_POST[telpon] <br /><hr />

    Nomor Order: $id_orders <br />
    Data order Anda adalah sebagai berikut: <br /><br />";

$no=1;
while ($d=mysql_fetch_array($daftarproduk)){
    $subtotalberat = $d[berat] * $d[jumlah]; // total berat per item produk
    $totalberat = $totalberat + $subtotalberat; // grand total berat all produk yang dibeli

    $subtotal = $d[harga] * $d[jumlah];
    $total = $total + $subtotal;
    $subtotal_rp = format_rupiah($subtotal);
    $total_rp = format_rupiah($total);
    $harga = format_rupiah($d[harga]);

    echo "<tr bgcolor=#cccccc><td>$no</td><td>$d[nama_produk]</td><td align=center>$d[berat]</td><td
    align=center>$d[jumlah]</td><td>Rp. $harga</td><td>Rp. $subtotal_rp</td></tr>";

    $pesan.="<b>$d[jumlah] $d[nama_produk] -> Rp. $harga -> Subtotal: Rp. $subtotal_rp <br />";
    $no++;
}

$ongkos=mysql_fetch_array(mysql_query("SELECT ongkos_kirim FROM kota WHERE id_kota='$_POST[kota]'"));

```



```
$ongkos kirim1=$ongkos[ongkos_kirim];  
$ongkos kirim = $ongkos kirim1; // * $total berat;
```

```
$grandtotal = $total + $ongkos kirim;
```

```
$ongkos kirim_rp = format_rupiah($ongkos kirim);  
$ongkos kirim1_rp = format_rupiah($ongkos kirim1);  
$grandtotal_rp = format_rupiah($grandtotal);
```

```
$pesan.="<br /><br />Total : Rp. $total_rp  
<br />Ongkos Kirim untuk Tujuan Kota Anda : Rp. $ongkos kirim1_rp/Kg  
<br />Total Berat : $total berat Kg  
<br />Total Ongkos Kirim : Rp. $ongkos kirim_rp  
<br />Grand Total : Rp. $grandtotal_rp  
<br /><br />Silahkan lakukan pembayaran ke C [REDACTED] sebanyak Grand Total yang tercantum,  
nomor rekeningnya <b>[REDACTED] /b> a.n. Febry C [REDACTED] <br />";
```

```
$subjek="Pemesanan Online shinigami7 [REDACTED]";
```

```
// Kirim email dalam format HTML  
$dari = "From: shinigami [REDACTED] \n";  
$dari .= "Content-type: text/html \r\n";
```

```
// Kirim email ke kustomer  
mail($_POST[email],$subjek,$pesan,$dari);
```

```
// Kirim email ke pengelola toko online  
mail("shinigami7 [REDACTED]", $subjek, $pesan, $dari);
```

```
echo "<html>  
    <head>  
    <body>  
        <tr><td colspan=5 align=right>Total : Rp. </td><td align=right><b>$total_rp</b></td></tr>  
        <tr><td colspan=5 align=right>Ongkos Kirim untuk Tujuan Kota Anda: Rp. </td><td align=right><b>$ongkos kirim1_rp</b></td></tr>  
        <tr><td colspan=5 align=right>Total Berat : </td><td align=right><b>$total berat Kg</b></td></tr>  
        <tr><td colspan=5 align=right>Total Ongkos Kirim : Rp. </td><td align=right><b>$ongkos kirim_rp</b></td></tr>  
        <tr><td colspan=5 align=right>Grand Total : Rp. </td><td align=right><b>$grandtotal_rp</b></td></tr>  
    </table>";  
echo "<hr /><p>Data order dan nomor rekening transfer akan dikirim 1x24 jam ke email Anda atau via sms. <br />  
    Apabila Anda tidak melakukan transaksi dan konfirmasi dalam tempo 3 hari, maka data order Anda akan dianggap hangus  
dan melakukan pemesanan ulang. <br />
```

```
        <br /></p>  
    </body>  
    </head>  
    </html>";  
}  
}
```

```
elseif ($_GET[module]=='statorders'){
```

```
// tampilkan rincian produk yang di order  
$sql2=mysql_query("SELECT * FROM orders WHERE status_order= 'Baru'");
```

```

echo"<table align=center>";
echo "<tr align=center><td><strong>Nama
Kostumer</strong></td><td><strong>Alamat</strong></td><td><strong>Telepon</strong></td><td><strong>Email</strong></td>
<td><strong>Status order</strong></td><td><strong>Tanggal order</strong></td></tr>";

while($s=mysql_fetch_array($sql2)){

    echo "<tr
align=center><td>$s[nama_kustomer]</td><td>$s[alamat]</td><td>$s[telpon]</td><td>$s[email]</td><td>$s[status_order]</td><t
d>$s[tgl_order]</td></tr>";

}

// tampilkan rincian produk yang di order2
$sql3=mysql_query("SELECT * FROM orders WHERE status_order= 'Lunas'");

while($s=mysql_fetch_array($sql3)){

    echo "<tr
align=center><td>$s[nama_kustomer]</td><td>$s[alamat]</td><td>$s[telpon]</td><td>$s[email]</td><td>$s[status_order]</td><t
d>$s[tgl_order]</td></tr>";
}
echo "</table>";
}
?>

```

#### D. Cek\_login.php

```

<?php
include "../config/koneksi.php";
function antiinjection($data){
    $filter_sql = mysql_real_escape_string(stripslashes(strip_tags(htmlspecialchars($data,ENT_QUOTES))));
    return $filter_sql;
}

$username = antiinjection($_POST[username]);
$password = antiinjection(md5($_POST[password]));

$sqllogin=mysql_query("SELECT * FROM admins WHERE username='$username' AND password='$password' AND
blokir='N'");
$ketemu=mysql_num_rows($sqllogin);
$r=mysql_fetch_array($sqllogin);

// Apabila username dan password ditemukan
if ($ketemu > 0){
    session_start();
    session_register("namauser");
    session_register("namalengkap");
    session_register("passuser");
    session_register("leveluser");
}

```

```

$_SESSION[namauser] = $r[username];
$_SESSION[namalengkap] = $r[nama_lengkap];
$_SESSION[passuser] = $r[password];
$_SESSION[leveluser] = $r[level];

header('location:media.php?module=home');
}
else{
echo "<link href=../config/adminstyle.css rel=stylesheet type=text/css>";
echo "<center>LOGIN GAGAL! <br>
    Username atau Password Anda tidak benar.<br>
    Atau account Anda sedang diblokir.<br>";
echo "<a href=index.php><b>ULANGI LAGI</b></a></center>";
}
?>

```

#### E. Media.php admin

```

<?php
error_reporting(0);

session_start();

if (empty($_SESSION[username]) AND empty($_SESSION[passuser])){

echo "<link href='style.css' rel='stylesheet' type='text/css'>
<center>Untuk mengakses modul, Anda harus login <br>";

echo "<a href=index.php><b>LOGIN</b></a></center>";

}

else{

?>

<html>

<head>

<title>Welcome To Admin</title>

<script type="text/javascript" src="../nicEdit.js"></script>

<script type="text/javascript">

    bkLib.onDomLoaded(function() { nicEditors.allTextAreas() });

</script>

```

```
</script>

<link rel="shortcut icon" href="A.ico" />

<link href="style.css" rel="stylesheet" type="text/css" />

</head>

<body>

<div id="header">

    <div id="menu">

        <ul>

            <li><a href=?module=home>&#187; Home</a></li>

            <?php include "menu.php"; ?>

            <li><a href=logout.php>&#187; Logout</a></li>

        </ul>

        <p>&nbsp;</p>

    </div>

    <div id="content"><?php include "content.php"; ?>

</div>

    <div id="footer"><marquee direction="right" scrollamount="6" height="30" width="50%">Copyright &copy;
go_gaming</marquee></div>

</div>

</body>

</html>

<?php
}
?>
```

## LAMPIRAN B

### *CURRICULUM VITAE*



#### ✚ DATA PRIBADI

Nama Lengkap : Febry Gustiawan  
Tempat, Tanggal lahir : Merauke, 19 February 1989  
Jenis Kelamin : Laki-laki  
Agama : Islam  
Status : Belum Menikah  
Kewarganegaraan : WNI(Warga Negara Indonesia)  
Alamat : Panyileukan Citra AB 2 No.2, Bandung  
No. Telp/HP : 022-7835072/085860264235  
Email : febry.gustiawan@gmail.com

#### ✚ PENDIDIKAN FORMAL

Sekolah/Universitas	Fakultas/Jurusan	Kota	Tahun Lulus
SDN Panyileukan 01 Bandung	-	Bandung	2001
SMPN 18 Bandung	-	Bandung	2004
SMA BPI 1 Bandung	IPS	Bandung	2007
Univ. Widyatama Bandung	Sistem Informasi	Bandung	2012

#### ✚ PENGALAMAN ORGANISASI

Organisasi	Posisi	Tahun
Karang Taruna	Sekretaris	2007
Karang Taruna	Divisi Olahraga	2008
HIMASI (Himpunan Mahasiswa Sistem Informasi)	Anggota	2007 – 2008

MAPSI (Malam Perkenalan Sistem Informasi)	Wakil Ketua	2008
Senat Mahasiswa Fakultas Teknik	Departemen Sosial	2009 – 2010
ITCNT (Information Technology Competition and Talkshow)	Anggota	2010
WTE (Widyatama Technology Event)	Logistik	2010

Demikian *Cirriculum Vitae* ini saya buat dengan sebenar-benarnya.

Bandung, Mey 2012

**Febry Gustiawan**  
Penulis

