

**PENGEMBANGAN SISTEM INFORMASI COMICREADER  
MENGUNAKAN KERANGKA KERJA YII**

**(Skripsi)**

**Oleh**

**SABILA RUSYDA**



**FAKULTAS MATEMATIKA DAN ILMU PENGETAHUAN ALAM  
UNIVERSITAS LAMPUNG  
BANDAR LAMPUNG  
2016**

## **ABSTRAK**

### **PENGEMBANGAN SISTEM INFORMASI COMICREADER MENGUNAKAN KERANGKA KERJA Yii**

**Oleh**

**SABILA RUSYDA**

Komik adalah istilah umum yang digunakan untuk menunjukkan fenomena gambar yang disandingkan secara berurutan dan bersebelahan. Biasanya komik tersedia dalam bentuk fisik berupa buku, namun demi mengikuti perkembangan digital maka lahirlah *webcomic* atau komik digital. ComicReader adalah aplikasi yang dibuat untuk mendistribusikan dan membaca komik secara digital. Aplikasi ini menggunakan Yii sebagai kerangka kerjanya. Yii adalah sebuah kerangka kerja berbasis PHP yang menggunakan pola arsitektur MVC.

Kata Kunci : Kerangka Kerja, Komik, Yii.

## **ABSTRACT**

### **DEVELOPMENT OF AN INFORMATION SYSTEM CALLED COMICREADER USING YII FRAMEWORK**

**By**

**SABILA RUSYDA**

Comic is a general term used to denote the phenomenon of images that is put together in sequence and placed side by side. It is usually available in the form of physical book, but in order to follow the development of digital technology, digital comic was born. ComicReader is an application that is created to digitally distribute and read comics. This application uses Yii as the framework. Yii is a PHP based framework that uses the MVC architectural pattern.

Keywords : Comic, Framework, Yii.

**PENGEMBANGAN SISTEM INFORMASI COMICREADER  
MENGUNAKAN KERANGKA KERJA YII**

**Oleh**

**SABILA RUSYDA**

**Skripsi**

**Sebagai Salah Satu Syarat untuk Memperoleh Gelar  
SARJANA KOMPUTER**

**Pada**

**Jurusan Ilmu Komputer  
Fakultas Matematika dan Ilmu Pengetahuan Alam**



**FAKULTAS MATEMATIKA DAN ILMU PENGETAHUAN ALAM  
UNIVERSITAS LAMPUNG  
BANDAR LAMPUNG  
2016**

**Judul Skripsi** : **PENGEMBANGAN SISTEM INFORMASI  
COMICREADER MENGGUNAKAN  
KERANGKA KERJA YII**

**Nama Mahasiswa** : **Sabila Rusyda**

**Nomor Pokok Mahasiswa** : **0817032006**

**Program Studi** : **Ilmu Komputer**

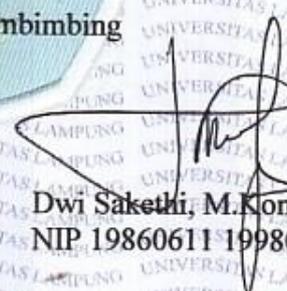
**Jurusan** : **Ilmu Komputer**

**Fakultas** : **Matematika dan Ilmu Pengetahuan Alam**

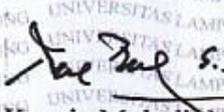


**1. Komisi Pembimbing**

  
**Aristoteles, M.Si.**  
**NIP 19810521 200604 1 002**

  
**Dwi Sakethi, M.Kom.**  
**NIP 19860611 199802 1 001**

**2. Ketua Jurusan Ilmu Komputer**

  
**Dr. Ir. Kurnia Muludi, M.S.Sc.**  
**NIP 1960616 198902 1 001**

**MENGESAHKAN**

**1. Tim Penguji**

**Pembimbing I : Aristoteles, M.Si.**

**Pembimbing II : Dwi Sakethi, M.Kom.**

**Pembahas : Ir. Machudor Yusman, M.Kom.**

**2. Dekan Fakultas Matematika dan Ilmu Pengetahuan Alam**



**Prof. Warsito, S.Si., DEA., Ph.D.**  
NIP. 19710212.199512.1.001

**Tanggal Lulus Ujian Skripsi: 7 Maret 2016**

## PERNYATAAN

Saya yang bertanda tangan di bawah ini, menyatakan bahwa skripsi saya yang berjudul "Pengembangan Sistem Informasi ComicReader Menggunakan Kerangka Kerja Yii" merupakan karya saya sendiri dan tidak dibuat oleh orang lain. Apabila dikemudian hari ditemukan penyimpangan, seperti penjiplakan, dan saya terbukti bersalah, maka saya bersedia menerima sanksi.



Dandar Lampung, 7 Maret 2016

Sabila Rusyda  
0817032006

## RIWAYAT HIDUP



Penulis dilahirkan di Bandar Lampung pada tanggal 02 September 1990, sebagai anak kedua dari empat bersaudara dari bapak Drs. Riyanto dan ibu Ir. Suhartini.

Pendidikan Taman Kanak-kanak (TK) Al-Kautsar Bandar Lampung diselesaikan tahun 1996, Sekolah Dasar (SD) diselesaikan di SD Al-Kautsar Bandar Lampung pada tahun 2002, Sekolah Menengah Pertama (SMP) di SMP Al-Kautsar Bandar Lampung pada tahun 2005, dan Sekolah Menengah Atas (SMA) di SMAN 9 Bandar Lampung pada tahun 2008.

Tahun 2008, penulis terdaftar sebagai mahasiswa Jurusan Ilmu Komputer FMIPA Unila melalui jalur PMKA. Pada tahun 2011, penulis melakukan Kerja Praktik (KP) di PT.PLN.

*This is for my family and friends who  
kept supporting me, Thank You.*

## **MOTO**

*Set rules for “Be You Self” and  
“Freedom”, because if not, ISIS can say  
“It’s our freedom to kill people, it’s us  
being ourselves” and that’s f\*cked up so  
let’s not promote that.*

## SANWACANA

Puji syukur penulis ucapkan kepada Allah SWT karena atas rahmat dan hidayah-Nya skripsi ini dapat diselesaikan. Skripsi berjudul “Pengembangan Sistem Informasi ComicReader Menggunakan Kerangka Kerja Yii” adalah salah satu syarat untuk memperoleh gelar Sarjana Komputer di Universitas Lampung. Dalam kesempatan ini, penulis mengucapkan terima kasih kepada:

1. Bapak Aristoteles, M.Si., selaku pembimbing satu dan juga pembimbing akademik saya, yang telah membimbing dan menyemangati selama penyusunan skripsi.
2. Bapak Dwi Sakethi, M.Kom., selaku pembimbing kedua yang telah membimbing dan menyemangati selama penyusunan skripsi.
3. Bapak Ir. Machudor Yusman, M.Kom., selaku pembahas yang telah memberikan masukan pada saat seminar dan ujian.
4. Bapak Febi Eka Febriansyah, M.T., selaku pembimbing akademik pertama.
5. Seluruh Dosen Jurusan Ilmu Komputer Universitas Lampung, terima kasih karena sudah membagi ilmunya.
6. Keluarga besar, terutama ayah Drs. Riyanto, ibu Ir. Suhartini, dan adik Shirtha El Rusyda, karena sudah menyemangati dan selalu mendoakan yang terbaik.

7. Seluruh teman-teman Ilmu Komputer 2008 yang sudah menyebar, mudah-mudahan bisa diberikan kesempatan untuk bertemu dan berkumpul lagi suatu hari nanti.
8. Seluruh pihak yang tidak dapat saya sebutkan satu persatu, namun ikut memberikan saya semangat, ide, dan ilmu.

Bandar Lampung, Maret 2016

**Sabila Rusyda**

## DAFTAR ISI

	<b>Halaman</b>
<b>DAFTAR TABEL</b> .....	viii
<b>DAFTAR GAMBAR</b> .....	ix
<b>I. PENDAHULUAN</b>	
1.1 Latar Belakang .....	1
1.2 Rumusan Masalah .....	2
1.3 Batasan Masalah .....	3
1.4 Tujuan .....	3
1.5 Manfaat .....	3
<b>II. TINJAUAN PUSTAKA</b>	
2.1 Komik .....	4
2.2 PHP .....	6
2.3 MVC .....	7
2.4 Yii .....	9
2.5 <i>Rational Unified Process</i> .....	11
2.6 <i>Unified Modeling Language</i> .....	19
<b>III. METODOLOGI PENELITIAN</b>	
3.1 Tempat dan Waktu Penelitian .....	22
3.2 Perangkat Keras .....	22
3.3 Perangkat Lunak .....	23
3.4 Perancangan Basis Data .....	23
3.5 Perancangan .....	28
3.5.1 Definisi Peran Pengguna .....	28

3.5.2	Diagram <i>Use Case</i> .....	28
3.5.3	Definisi <i>Use Case</i> .....	30
3.5.4	Skenario <i>Use Case</i> .....	31

#### **IV. HASIL DAN PEMBAHASAN**

4.1	Struktur Navigasi .....	38
4.2	<i>Coding</i> (Pengkodean) .....	39
4.2.1	Home .....	39
4.2.2	Top100 .....	43
4.2.3	Tags .....	45
4.2.4	Search .....	46
4.2.5	Register .....	48
4.2.6	Login .....	50
4.2.7	Subscriptions .....	53
4.2.8	Comic Information .....	55
4.2.9	Read .....	58
4.2.10	MyAdmin .....	60
4.2.11	Manage Comics .....	61
4.2.12	Manage Deposits .....	68
4.2.13	Manage Users .....	71
4.2.14	Manage Settings .....	72
4.3	<i>Testing</i> (Pengujian) .....	75
4.3.1	Tautan .....	75
4.3.2	Form Register .....	75
4.3.3	Form Login .....	77
4.3.4	Form Search .....	78
4.3.5	Form Add New Comic .....	78
4.3.6	Form Add New Chapter .....	81
4.3.7	Form Add New Page .....	83

#### **V. KESIMPULAN DAN SARAN**

5.1	Kesimpulan .....	86
-----	------------------	----

5.2 Saran .....	87
<b>DAFTAR PUSTAKA .....</b>	<b>88</b>
<b>LAMPIRAN .....</b>	<b>89</b>

## DAFTAR TABEL

<b>Tabel</b>	<b>Halaman</b>
3.1 Spesifikasi laptop .....	22
3.2 Tabel cr_users .....	23
3.3 Tabel cr_deposits .....	24
3.4 Tabel cr_comics .....	24
3.5 Tabel cr_tags .....	25
3.6 Tabel cr_chapters .....	25
3.7 Tabel cr_pages .....	26
3.8 Tabel cr_transactions .....	26
3.9 Definisi peran pengguna .....	28
3.10 Definisi <i>use case</i> .....	30
3.11 Skenario <i>use case login</i> .....	31
3.12 Skenario <i>use case</i> memasukkan data pengguna .....	31
3.13 Skenario <i>use case</i> mengubah data pengguna .....	31
3.14 Skenario <i>use case</i> menghapus data pengguna .....	32
3.15 Skenario <i>use case</i> memasukkan data komik .....	33
3.16 Skenario <i>use case</i> mengubah data komik .....	33
3.17 Skenario <i>use case</i> menghapus data komik .....	34
3.18 Skenario <i>use case</i> membeli komik .....	35
3.19 Skenario <i>use case</i> membaca komik .....	35
3.20 Skenario <i>use case</i> mencari komik .....	35
3.21 Skenario <i>use case</i> mengajukan penambahan saldo .....	36
3.22 Skenario <i>use case</i> melakukan konfirmasi <i>transfer</i> saldo .....	36
3.23 Skenario <i>use case</i> mendaftar jadi anggota .....	37
4.1 <i>Quick button</i> .....	41

## DAFTAR GAMBAR

<b>Gambar</b>	<b>Halaman</b>
2.1 Contoh panel dari komik <i>Peanuts</i> karangan Charles Schultz .....	5
2.2 Contoh kode PHP di dalam kode HTML .....	6
2.3 <i>Model-View-Controller</i> .....	8
2.4 Logo Yii .....	9
2.5 Grafik Model Iteratif yang menunjukkan bagaimana proses disusun secara dua dimensi .....	14
2.6 Sebuah deklarasi kelas dapat ditampilkan dengan menggunakan notasi UML .....	20
3.1 Skema basis data .....	28
3.2 Diagram <i>use case</i> tamu .....	28
3.3 Diagram <i>use case</i> anggota .....	29
3.4 Diagram <i>use case</i> admin .....	29
4.1 Struktur navigasi .....	38
4.2 Tampilan Home .....	42
4.3 Tampilan Top100 .....	44
4.4 Tampilan Tags .....	45
4.5 Tampilan Search .....	47
4.6 Pencarian melalui <i>link</i> pemilik komik .....	47
4.7 Pencarian melalui <i>link tag</i> .....	48
4.8 Tampilan <i>form</i> Register .....	50
4.9 Tampilan <i>form</i> Login .....	53
4.10 Tampilan Subscriptions .....	54
4.11 Tampilan Comic Information di <i>frontend</i> .....	57
4.12 Tampilan daftar <i>chapter</i> .....	58
4.13 Tampilan Read .....	59

4.14	Tampilan Dashboard pada modul MyAdmin .....	61
4.15	Tampilan Manage Comics .....	62
4.16	Tampilan <i>form</i> Add New Comic dan Edit Comic .....	64
4.17	Tampilan Comic Information di <i>backend</i> .....	65
4.18	Tampilan <i>form</i> Add New Chapter dan Edit Chapter .....	66
4.19	Tampilan Chapter Information .....	67
4.20	Tampilan Add New Page(s) .....	68
4.21	Tampilan Manage Deposits untuk admin .....	69
4.22	Tampilan Manage Deposits untuk anggota .....	69
4.23	Tampilan All Deposits .....	70
4.24	Tampilan Manage Users .....	71
4.25	Tampilan <i>form</i> Add New User dan Edit User .....	72
4.26	Tampilan Manage Settings .....	73
4.27	Tampilan <i>form</i> Edit Account .....	73
4.28	Tampilan <i>form</i> Change Password .....	74
4.29	Masukkan data registrasi .....	75
4.30	Pesan registrasi berhasil dilakukan .....	76
4.31	Data tersimpan di tabel <i>cr_users</i> .....	76
4.32	Pesan <i>error</i> registrasi .....	76
4.33	Masukkan data ke dalam <i>form</i> Login .....	77
4.34	Akun berhasil <i>login</i> .....	77
4.35	Pesan <i>error login</i> untuk kesalahan memasukkan <i>username</i> .....	77
4.36	Pesan <i>error login</i> untuk kesalahan memasukkan <i>password</i> .....	78
4.37	Pencarian menggunakan kata kunci .....	78
4.38	Masukkan data komik .....	79
4.39	Komik baru berhasil dibuat .....	79
4.40	Data tersimpan di tabel <i>cr_comics</i> .....	80
4.41	Lokasi <i>folder</i> dan <i>cover</i> komik .....	80
4.42	Pesan <i>error</i> pada Add New Comic .....	80
4.43	Masukkan data <i>chapter</i> .....	81
4.44	<i>Chapter</i> baru berhasil dibuat .....	82
4.45	Data tersimpan di tabel <i>cr_chapters</i> .....	82

4.46 Lokasi <i>folder</i> dan <i>cover chapter</i> .....	82
4.47 Pesan <i>error</i> pada Add New Chapter .....	83
4.48 Masukkan gambar untuk diunggah .....	84
4.49 <i>Page</i> baru berhasil dibuat .....	84
4.50 Data tersimpan di tabel <i>cr_pages</i> .....	84
4.51 Lokasi <i>page</i> dan <i>thumbnail</i> .....	85
4.52 Pesan <i>error</i> pada Add New Page .....	85

# **BAB I**

## **PENDAHULUAN**

### **1.1 Latar Belakang**

Ketika berbicara mengenai perkembangan budaya pop, maka komik, yang dalam bahasa Inggris disebut dengan *comic* dan dalam bahasa Jepang disebut dengan *manga*, adalah salah satu media yang tidak dapat dilupakan jasanya. Banyak tren yang lahir dan disebarkan melalui media ini. Salah satunya adalah tren *cosplay*. *Cosplay* adalah salah satu bentuk penampilan artistik dimana para *cosplayer*, sebutan bagi para aktor *cosplay*, akan berhias serta mengenakan kostum dan aksesoris yang menggambarkan karakter dalam komik dan film animasi. Bila novel biasa diadaptasi menjadi film layar lebar, maka komik merupakan salah satu media yang paling sering diadaptasi ke dalam bentuk film animasi atau biasa disebut juga dengan film kartun. Artis yang pekerjaannya adalah membuat komik disebut dengan komikus, ada komikus yang bekerja sendiri namun ada juga yang bekerja sebagai tim, misalnya, orang lain yang menulis jalan cerita sedangkan dia menggambar berdasarkan cerita tersebut.

Biasanya komik tersedia dalam bentuk fisik berupa buku, namun demi mengikuti perkembangan digital maka lahirlah *webcomic* atau komik digital, dimana

pembaca bisa membaca komik melalui media digital, seperti komputer dan tablet. Sedangkan komikus bisa memproduksi dan mendistribusikan komik secara lebih murah tanpa harus melalui penerbit dan membayar biaya cetak. Karena biaya produksi yang lebih murah itulah, secara tidak langsung *platform* komik digital akan memberikan peluang yang lebih luas bagi para komikus amatir yang ingin memulai karir dibidang ini namun memiliki anggaran terbatas.

ComicReader yang dikembangkan dalam penelitian ini adalah sebuah *platform* komik digital berupa aplikasi untuk mendistribusikan dan membaca komik secara digital, dengan komikus dan pembaca komik sebagai target pengguna utamanya. Aplikasi ini berguna sebagai alat manajemen komik yang akan atau sudah dipublikasikan kedalam media digital. ComicReader menggunakan Yii sebagai kerangka kerja pemrogramannya. Yii adalah sebuah kerangka kerja berbasis PHP yang dikembangkan oleh Qiang Xue, seorang pengembang sistem dan aplikasi yang memiliki pengalaman mengembangkan kerangka kerja lain bernama Prado.

## **1.2 Rumusan Masalah**

Dalam penelitian ini, terdapat beberapa masalah yang dibahas:

1. Bagaimana cara data komik dikelola menggunakan aplikasi yang dibuat dengan Yii?
2. Bagaimanakah cara komik ditampilkan menggunakan aplikasi yang dibuat dengan Yii?
3. Apa saja peran pengguna aplikasi?

4. Apa saja fitur yang didapatkan oleh pengguna aplikasi?

### **1.3 Batasan Masalah**

Berikut ini adalah beberapa batasan masalah dalam penelitian ini:

1. Aplikasi berbasis web.
2. Pengembangan aplikasi menggunakan kerangka kerja Yii.
3. Mata uang yang digunakan adalah rupiah.
4. File yang diunggah hanya file jpg, jpeg, dan png.

### **1.4 Tujuan**

Tujuan dari penelitian ini adalah:

1. Mengembangkan aplikasi berbasis web.
2. Mengembangkan aplikasi menggunakan kerangka kerja Yii.

### **1.5 Manfaat**

Manfaat dilakukannya penelitian ini adalah untuk:

1. Mengetahui cara membuat aplikasi menggunakan kerangka kerja Yii.
2. Mempelajari fitur-fitur di dalam kerangka kerja Yii.
3. Mengetahui cara membuat aplikasi untuk mengelola data-data komik.

## **BAB II**

### **TINJAUAN PUSTAKA**

#### **2.1 Komik**

Setelah pernah dianggap sebagai salah satu bentuk terendah dari hiburan massal, saat ini komik secara luas dianggap berpotensi mampu mengekspresikan secara kompleks dan mendalam sebuah bentuk seni sastra maupun bentuk seni visual. Pencarian definisi kata komik harus dimulai dengan menguraikan medium perantara buku komik dari kebingungan semantik kata komik. Pertama-tama, tidak ada medium perantara yang bernama komik. Komik adalah istilah umum yang digunakan untuk menunjukkan fenomena gambar yang disandingkan secara berurutan dan bersebelahan. Menurut ahli teori komik, Scott McCloud, kata ini bisa digunakan untuk *comic strips*, *comic books*, *cave paintings*, *Grecian urns* (guci Yunani), *tapestries* (permadani), *stained glass windows* (jendela gelas berwarna), dan seterusnya. Kata ini bisa digunakan untuk merepresentasikan banyak medium tersebut dikarenakan adanya kemiripan dalam hal bentuk. Salah satu kemiripan dari contoh-contoh tersebut adalah, proses penulisan ceritanya dibuat dalam bentuk presentasi dari momen-momen yang dipilih secara selektif, ke dalam panel-panel dengan ukuran berbeda. Panel adalah daerah yang dapat dilihat yang berisi momen dalam cerita. (Duncan, 2009).



**Gambar 2.1** Contoh panel dari komik *Peanuts* karangan Charles Schultz (Duncan, 2009)

Dibandingkan dengan karya sastra lain, komik memiliki beberapa kelebihan yaitu kepribadian penulis tidak terlihat menonjol. Penulis berusaha untuk lebih banyak mengungkapkan orisinalitasnya melalui gambar dan bukan melalui cerita. Ciri khas bentuk-bentuk kesenian ini adalah pembaca dibawa ke dalam alam imajinasi dimana alam tersebut bisa saja berbeda dari alam nyata, atau ke dalam lingkungan sosial yang tidak pernah dimasuki manusia, baik secara langsung maupun melalui tokoh-tokohnya, penulis membeberkan pandangan dunianya, menawarkan renungan yang mustahil disampaikan secara langsung. Penulis menempatkan tokoh-tokohnya di dalam situasi komunikasi, melalui perilaku verbal yang miskin nuansa karena adanya bingkai (ukuran balon). Komik juga mengandung aspek grafis, dimana gambar menggantikan penjelasan panjang, karena gambar dapat mengantarkan pembaca pada berbagai realitas yang terkadang sulit dibayangkan, karena itulah gambar komik cukup sederhana. Gambar juga mampu menyumbang beberapa unsur yang berguna untuk menjawab masalah yang lebih umum, misalnya gambaran yang khas Indonesia. (Bonneff, 2008).

## 2.2 PHP

Sub bab 2.2 sepenuhnya diambil dari buku *PHP: A Beginner's Guide* (Vaswani, 2009). PHP, tiga kata yang bila digabungkan akan membentuk nama salah satu bahasa pemrograman yang paling populer di dunia untuk pengembangan web, yaitu *PHP: Hypertext Preprocessor*. Selama beberapa tahun terakhir, PHP telah menjadi pilihan *de facto* untuk pengembangan aplikasi web berbasis data, terutama karena skalabilitas, kemudahan penggunaan, dan dukungan luas untuk *database* dan format data yang berbeda.

Ketika mengembangkan aplikasi PHP untuk web, pendekatan yang biasa digunakan adalah dengan menanamkan kode PHP ke dalam satu atau lebih dokumen HTML standar menggunakan "*tags*" khusus atau *delimiters* (pembatas) seperti pada Gambar 2.2.

```
<html>
<head></head>
<body>
<div>
  <?php echo sqrt(49); ?>
</div>
</body>
</html>
```

**Gambar 2.2 Contoh kode PHP di dalam kode HTML** (Vaswani, 2009)

Pada saat dokumen HTML diminta oleh pengguna, web server yang dapat membaca PHP, dapat mengenali dan mengeksekusi blok kode PHP lalu menyisipkan keluaran yang dihasilkan ke dokumen HTML sebelum mengembalikannya ke pengguna yang meminta. Hasilnya adalah sebuah halaman

web atau aplikasi yang tampak hidup, menanggapi secara cerdas semua tindakan pengguna berdasarkan logika program PHP yang tertanam di dalamnya.

### 2.3 MVC

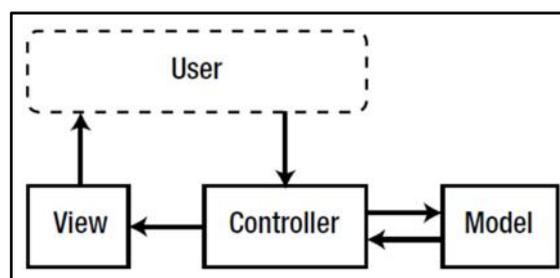
Sub bab 2.3 sepenuhnya diambil dari buku *Pro PHP MVP* (Pitt, 2012). MVC (*Model-View-Controller*) adalah pola desain *software* yang dibangun dari interkoneksi antara tiga jenis komponen utama dalam bahasa pemrograman, contohnya PHP. Ketiga jenis komponen tersebut adalah *model* (model), *view* (tampilan), dan *controller* (pengontrol). MVC biasanya memiliki fokus yang kuat pada paradigma perangkat lunak *object-oriented programming* (OOP).

*Model* adalah tempat dimana semua logika bisnis dari aplikasi disimpan. Logika bisnis bisa berupa hal-hal yang khusus berhubungan dengan bagaimana sebuah aplikasi menyimpan data, atau menggunakan jasa pihak ketiga, dalam rangka untuk memenuhi kebutuhan bisnisnya. Jika aplikasi harus mengakses informasi di dalam database, kode untuk melakukan hal itu akan disimpan dalam model. Jika diperlukan, misalnya, untuk mengambil data saham atau *tweet* tentang produk baru, kode yang digunakan juga akan disimpan di dalam model.

*View* adalah tempat dimana semua elemen *user interface*, atau tampilan pengguna, aplikasi disimpan. Hal ini dapat mencakup *markup* HTML, CSS *style sheets*, dan *file* JavaScript. Apa pun yang dapat dilihat atau berinteraksi dengan pengguna dapat disimpan dalam *view*, dan adakalanya apa yang pengguna lihat sebenarnya

merupakan kombinasi dari beberapa *view* berbeda yang dipanggil dalam permintaan yang sama.

*Controller* adalah komponen-komponen yang menghubungkan model dengan *view*. *Controller* mengisolasi logika bisnis milik model dari elemen antarmuka pengguna milik *view*, dan menangani bagaimana aplikasi akan merespon interaksi pengguna dalam *view*. *Controller* adalah titik pertama untuk masuk ke trio komponen ini, karena permintaan pertama akan melewati *controller*, yang kemudian memanggil model dan *view* yang diperlukan untuk memenuhi permintaan dari aplikasi. Lihat Gambar 2.2.



**Gambar 2.3 Model-View-Controller** (Pitt, 2012)

Tidak semua permintaan akan membutuhkan model atau *view*. Elemen yang dimuat tergantung pada jenis permintaan dan sumber daya yang diperlukan untuk memenuhinya. URL yang diminta lalu mendefinisikannya, dalam proses yang disebut *routing*. Sebuah *controller* dapat, misalnya, hanya berfungsi untuk mengganti keadaan aplikasi, atau untuk mengembalikan data yang belum diolah langsung dari layanan pihak ketiga. Dalam kasus-kasus tersebut, model atau *view* tidak diperlukan.

## 2.4 Yii

Sub bab 2.4 sepenuhnya diambil dari buku *The Yii Book, Revision: 0.9* (Ullman, 2014). Menulis segala sesuatu dari awal untuk setiap pengerjaan proyek merupakan suatu hal yang tidak praktis. Penggunaan kembali kode-kode yang pernah dibuat, akan lebih mempercepat pekerjaan pengkodean, lebih dapat diandalkan karena pernah digunakan sebelumnya, dan lebih aman karena sudah teruji. Maka dari itu, banyak pengembang yang menciptakan *framework* kecil untuk digunakan sendiri, seperti misalnya beberapa *functions* (fungsi atau metode) yang sering digunakan pada saat membuat perangkat lunak. *Framework* (kerangka kerja) sendiri adalah kumpulan kode yang dimaksudkan untuk mempercepat proses pembuatan atau pengembangan perangkat lunak. Tampilan logo Yii ditunjukkan pada Gambar 2.4.



**Gambar 2.4 Logo Yii** (yiiframework.com)

Yii diciptakan oleh Qiang Xue dan pertama kali dirilis pada tahun 2008. Yii merupakan singkatan dari “*Yes, it is!*”. Kata ini dianggap oleh pembuatnya sebagai respon yang paling akurat dan paling ringkas untuk menjawab pertanyaan dari mereka yang baru mengenal Yii, yaitu: “*Is it fast? ... Is it secure? ... Is it professional? ... Is it right for my next project? ... Yes, it is!*”.

Xue juga merupakan pendiri *framework* Prado, yang mengambil inspirasinya dari *framework* terkenal ASP.NET untuk pengembangan Windows. Dalam membuat Yii, Xue mengambil bagian terbaik dari Prado, Ruby on Rails, CakePHP, dan Symfony untuk menciptakan *framework* PHP yang modern, kaya fitur, dan sangat *useable* (bisa digunakan). Seperti kebanyakan *framework*, Yii menggunakan *Object-Oriented Programming* (OOP) murni. Tidak seperti kebanyakan *framework* lain, Yii mewajibkan penggunaan versi 5 dari PHP. Hal ini penting, karena PHP5 memiliki struktur objek yang jauh lebih baik dan maju dibandingkan dengan PHP4 yang lebih tua. Yii menggunakan pola arsitektur *de facto* standar *Model-View-Controller* (MVC).

Hampir semua aplikasi web yang ada sekarang ini, bergantung pada sebuah *database* atau basis data. Akibatnya, bagaimana *framework* mengelola interaksi *database* menjadi sangat penting. Yii dapat bekerja dengan *database* dalam beberapa cara yang berbeda, tetapi konvensi standarnya adalah dengan menggunakan *Object Relational Mapping* (ORM) melalui *Active Record* (AR). ORM menangani konversi data dari satu sumber ke sumber lain. Dalam kasus aplikasi berbasis Yii, data dipetakan dari variabel objek PHP ke dalam *database* dan sebaliknya.

Untuk interaksi *database* tingkat rendah, Yii menggunakan *PHP Data Objects* (PDO) milik PHP5. PDO menyediakan lapisan abstraksi akses data, yang memungkinkan penggunaan kode yang sama untuk berinteraksi dengan *database*, terlepas dari aplikasi *database* yang terlibat. Salah satu fitur terbaik Yii adalah

bahwa jika pengguna lebih memilih pendekatan yang berbeda, pengguna dapat menukar atau menambahkan alternatif baru. Sebagai contoh, pengguna dapat mengubah:

1. Dasar *database-specific library*.
2. Sistem template yang digunakan untuk membuat *output*.
3. Bagaimana *caching* dilakukan.

Alternatif yang bisa dimasukkan dapat berupa kode buatan sendiri, atau yang ditemukan di *library* pihak ketiga, termasuk kode dari *framework* lain. Walaupun banyak fleksibilitas yang ditawarkan, Yii masih sangat stabil, dan dengan menggunakan *caching* serta perangkat lainnya, Yii dapat tampil dengan cukup baik. Aplikasi Yii memiliki citra yang baik juga, karena telah diuji pada beberapa situs dengan permintaan tinggi, seperti Stay.com dan VICE.

## **2.5 Rational Unified Process**

Sub bab 2.5 sepenuhnya diambil dari buku *Rational Unified Process: Best Practices for Software Development Teams* (IBM, 2005). *Rational Unified Process* (RUP) adalah sebuah *Software Engineering Process*. RUP menyediakan pendekatan disiplin untuk menetapkan tugas dan tanggung jawab dalam suatu organisasi pengembang. Tujuannya adalah untuk memastikan diproduksi perangkat lunak berkualitas tinggi yang memenuhi kebutuhan pengguna akhir, dengan jangka waktu dan jumlah anggaran yang dapat diprediksi. RUP

merupakan produk pengolahan atau proses, yang dikembangkan dan dipelihara oleh Rational® Software.

*Rational Unified Process* dapat meningkatkan produktivitas tim, dengan memberikan akses mudah bagi setiap anggota tim ke basis pengetahuan yang berisi pedoman, contoh, dan alat mentor untuk semua hal penting dalam kegiatan pengembangan. Dengan memiliki semua anggota tim yang dapat mengakses basis pengetahuan yang sama, maka tidak akan ada masalah apabila mereka bekerja pada bagian persyaratan, desain, pengujian, manajemen proyek, ataupun manajemen konfigurasi, RUP akan memastikan semua anggota tim dapat berbagi bahasa, proses, dan penglihatan yang sama tentang bagaimana cara mengembangkan sebuah perangkat lunak.

Kegiatan RUP adalah tentang pembuatan dan pemeliharaan model. Daripada hanya berfokus pada penulisan dokumen yang tebal dan banyak, *Unified Process* lebih menekankan pada pengembangan dan pemeliharaan model – representasi yang kaya secara semantik dari sistem perangkat lunak yang sedang dikembangkan. Dengan begitu, *Rational Unified Process* merupakan panduan untuk bagaimana cara menggunakan *Unified Modeling Language* (UML) secara efektif. UML adalah bahasa standar industri yang memungkinkan pengembang untuk secara jelas mengkomunikasikan persyaratan, arsitektur, dan desain. UML sendiri dibuat oleh Rational® Software, dan sekarang berada di bawah naungan organisasi standar Object Management Group (OMG).

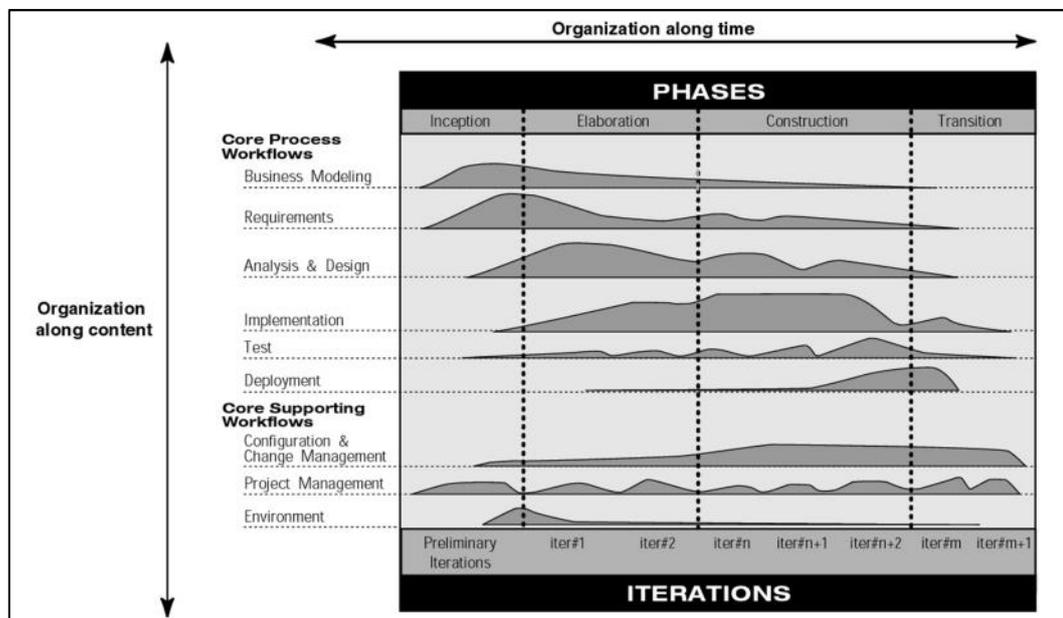
*Rational Unified Process* didukung oleh alat-alat, yang mengotomatisasi sebagian besar proses. Mereka digunakan untuk membuat dan memelihara berbagai artefak, terutama model, dari proses rekayasa perangkat lunak: permodelan visual, pemrograman, pengujian, dan lain-lain. Mereka tidak terhingga nilainya dalam mendukung semua pembukuan yang terkait dengan manajemen perubahan, serta manajemen konfigurasi yang menyertai setiap iterasi. RUP adalah proses yang dapat dikonfigurasi. Tidak ada proses tunggal yang cocok untuk semua pengembangan perangkat lunak. *Unified Process* dapat digunakan oleh tim pengembang kecil ataupun organisasi pengembang besar.

RUP menangkap banyak praktik terbaik dari pengembangan perangkat lunak modern dalam bentuk yang cocok untuk berbagai proyek dan organisasi. Dengan menyebarkan praktek-praktek terbaik menggunakan *Rational Unified Process* sebagai panduan, tim pengembang akan ditawarkan dengan sejumlah kunci keunggulan. Enam praktik terbaik yang biasa digunakan oleh organisasi adalah:

1. Kembangkan perangkat lunak secara *iterative* atau berulang.
2. Kelola kebutuhan dan persyaratan.
3. Gunakan arsitektur berbasis komponen.
4. Model perangkat lunak secara visual.
5. Uji kualitas perangkat lunak
6. Kontrol perubahan pada perangkat lunak

Jalannya proses yang ditunjukkan pada Gambar 2.5 bisa dijelaskan secara dua dimensi, atau melalui dua sumbu:

1. Sumbu horisontal mewakili waktu dan menunjukkan aspek dinamis dari proses saat dia diberlakukan, dan hal ini dinyatakan dalam *cycles* (siklus), *phases* (tahap atau fase), *iterations* (iterasi), dan *milestones* (tonggak atau target).
2. Sumbu vertikal mewakili aspek statis dari proses, bagaimana proses dijelaskan dalam hal *activities* (kegiatan), *artifacts* (artefak), *workers* (pekerja) dan *workflows* (alur kerja).



**Gambar 2.5 Grafik Model Iteratif yang menunjukkan bagaimana proses disusun secara dua dimensi (IBM, 2005)**

Pada penyusunan proses secara dinamis berdasarkan waktu. Siklus hidup perangkat lunak dipecah kedalam bentuk siklus-siklus, masing-masing siklus bekerja untuk sebuah generasi baru dari sebuah produk. RUP membagi satu siklus pengembangan menjadi empat fase berurutan, yaitu:

### 1. *Inception* (Permulaan)

Pengembang membangun kasus bisnis untuk sistem dan membatasi ruang lingkup proyek. Untuk mencapai ini, pengembang harus mengidentifikasi semua entitas eksternal yang berinteraksi dengan sistem (aktor) dan menentukan sifat interaksi ini pada tingkat tinggi. Ini melibatkan pengidentifikasian semua *use case* dan menggambarkan beberapa yang penting. Kasus bisnis meliputi kriteria keberhasilan, penilaian resiko, dan perkiraan sumber daya yang dibutuhkan, serta perencanaan fase yang menunjukkan tanggal pencapaian untuk semua target utama. Pada akhir fase ini, keputusan untuk meneruskan atau membatalkan proyek bisa dibuat bila target yang telah ditentukan gagal tercapai. Beberapa macam hasil yang bisa didapatkan dari fase ini berupa:

- a. Dokumen berisi kebutuhan proyek, fitur utama, dan batasan utama.
- b. Model *use case* awal (10% - 20% selesai).
- c. Kasus bisnis awal, yang di dalamnya terdapat konten bisnis, kriteria sukses dan ramalan finansial.
- d. Penilaian resiko awal.
- e. Rancangan proyek yang menunjukkan fase dan iterasi.
- f. Model bisnis, jika dibutuhkan.
- g. Satu atau beberapa prototipe.

### 2. *Elaboration* (Penguraian)

Tujuan dari fase ini adalah untuk menganalisa masalah utama, membangun dasar arsitektur suara, mengembangkan rancangan proyek, dan menghilangkan unsur-unsur dengan risiko tertinggi dari proyek.

Untuk mencapai tujuan ini, pengembang harus memiliki pandangan luas dan mendalam tentang sistem. Keputusan arsitektur harus dibuat dengan pemahaman dari keseluruhan sistem, seperti: ruang lingkup, fungsi utama dan persyaratan nonfungsional seperti persyaratan kinerja.

Pada tahap ini, prototipe arsitektur, yang dapat dieksekusi, dibangun dalam satu atau lebih iterasi, tergantung pada ruang lingkup, ukuran, resiko, dan kebaruan proyek. Upaya ini setidaknya akan mengatasi *use case* kritis yang teridentifikasi pada tahap awal, dan biasanya menunjukkan resiko teknis utama dalam proyek. Sementara itu, perubahan prototipe dari segi kualitas produksi komponen selalu menjadi tujuan, dengan tidak meniadakan pengembangan satu atau lebih prototipe sekali pakai, yang lebih diarahkan untuk penyelidikan, untuk mengurangi risiko tertentu seperti *trade-off* antara desain dan kebutuhan, studi kelayakan komponen, atau demonstrasi kepada investor, pelanggan, dan pengguna akhir. Pada akhir fase ini, keputusan untuk meneruskan atau membatalkan proyek bisa dibuat bila target yang telah ditentukan gagal tercapai. Beberapa macam hasil yang bisa didapatkan dari fase ini berupa:

- a. Model *use case* (sekitar 80% selesai), semua *use case* dan aktor telah diidentifikasi dan sebagian besar deskripsi *use case* telah dibuat.
- b. Persyaratan tambahan yang menangkap kebutuhan non-fungsional dan kebutuhan yang tidak berhubungan dengan *use case* tertentu.
- c. Deskripsi arsitektur perangkat lunak.

- d. Prototipe yang dapat dieksekusi.
- e. Revisi daftar resiko dan revisi kasus bisnis.
- f. Rancangan pengembangan untuk keseluruhan proyek, termasuk rancangan proyek secara kasar yang menunjukkan iterasi dan kriteria evaluasi untuk setiap iterasi.

### 3. *Construction* (Pembangunan)

Selama fase pembangunan, semua komponen yang tersisa dan fitur aplikasi dikembangkan dan diintegrasikan ke dalam produk, dan semua fitur telah diuji secara cermat. Tahap pembangunan, di satu sisi, adalah sebuah proses manufaktur, dimana penekanannya ditempatkan pada pengelolaan sumber daya dan pengendalian operasi untuk mengoptimalkan biaya, jadwal, dan kualitas. Dalam hal ini, pola pikir manajemen mengalami transisi dari pengembangan kekayaan intelektual selama fase permulaan dan penguraian, ke pengembangan produk yang bisa disebarakan selama fase pembangunan dan peralihan. Pada akhir fase ini, bila target yang ditentukan gagal dicapai maka mungkin fase transisi harus ditunda pada rilis berikutnya. Hasil akhir dari fase ini adalah sebuah produk yang siap diberikan kepada pengguna akhir, produk ini setidaknya memiliki beberapa hal seperti berikut ini:

- a. Produk perangkat lunak yang terintegrasi dengan *platform* yang memenuhi syarat dan kebutuhan.
- b. Panduan pengguna.
- c. Penjelasan mengenai rilis saat ini.

#### 4. *Transition* (Peralihan)

Tujuan dari fase peralihan adalah untuk mengalihkan produk perangkat lunak kepada komunitas pengguna. Setelah produk telah diberikan kepada pengguna akhir, masalah biasanya muncul yang mengharuskan pengembang untuk mengembangkan rilis baru, memperbaiki beberapa masalah, atau menyelesaikan fitur yang ditunda. Fase peralihan bisa dimasuki pada saat perangkat lunak dirasa sudah cukup matang untuk digunakan pada domain pengguna akhir. Fase ini biasanya membutuhkan beberapa bagian yang akan digunakan oleh sistem, agar dapat bekerja dengan normal, sudah selesai sampai dengan tingkat kualitas yang bisa diterima dan dokumentasi pengguna telah tersedia, sehingga transisi ke pengguna akan memberikan hasil yang positif bagi semua pihak.

Fase ini berfokus pada kegiatan yang diperlukan untuk menempatkan perangkat lunak ke tangan pengguna. Biasanya, fase ini meliputi beberapa iterasi, termasuk rilis beta, rilis yang tersedia secara umum, serta perbaikan *bug* dan rilis perangkat tambahan. Banyak upaya yang dikeluarkan dalam mengembangkan dokumentasi yang berorientasi pengguna, pelatihan pengguna, mendukung pengguna dalam menggunakan produk awal mereka, dan bereaksi terhadap umpan balik pengguna. Namun pada titik ini, umpan balik pengguna harus dibatasi terutama untuk penyetelan produk, konfigurasi, instalasi, dan masalah kegunaan.

Setiap fase disimpulkan dengan *milestone* atau target yang telah ditentukan. *Milestone* adalah titik dalam waktu di mana keputusan penting tertentu harus dibuat, dan oleh karena itu tujuan utama harus telah dicapai.

## ***2.6 Unified Modeling Language***

*Unified Modeling Language* (UML) adalah sebuah bahasa permodelan standar untuk pengembangan sistem dan perangkat lunak. Model adalah bentuk abstraksi dari hal yang sebenarnya. Pada saat pengembang memodelkan sistem, berarti mereka mengabstraksi rincian yang tidak relevan atau berpotensi membingungkan. Model adalah penyederhanaan dari sistem sebenarnya, sehingga memungkinkan desain dan kelayakan sistem untuk dipahami, dievaluasi, dan dikritisi lebih cepat daripada jika pengembang harus menggali melalui sistem itu sendiri. Untuk secara efektif memodelkan sistem, pengembang memerlukan satu hal yang sangat penting, yaitu: bahasa dimana model dapat digambarkan, dan di sini UML ikut berperan (Rumbaugh, 2005).

Sebuah bahasa pemodelan dapat terdiri dari *pseudo-code*, kode aktual, gambar, diagram, atau kalimat deskripsi yang panjang. Dengan kata lain, segala hal yang dapat membantu pengembang menjelaskan sistem mereka. Unsur-unsur yang membentuk bahasa pemodelan disebut dengan notasi. Gambar 2.6 menunjukkan contoh notasi UML (Hamilton, 2006).

Guitarist
-instrument: Instrument
+getInstrument(): Instrument +setInstrument(instrument: Instrument): void +play(): void +main(args: String[]): void

**Gambar 2.6** Sebuah deklarasi kelas dapat ditampilkan dengan menggunakan notasi UML (Hamilton, 2006)

Namun, notasi bukanlah keseluruhan cerita. Tanpa diberitahu bahwa salah satu kotak di Gambar 2.6 mewakili kelas, kebanyakan orang tidak akan tahu apakah gambar itu, meskipun mungkin bisa menebak. Deskripsi dari apa yang dimaksud oleh notasi disebut dengan semantik dari bahasa dan ditangkap di *meta-model* bahas. UML memiliki enam kelebihan, yaitu:

1. Bahasanya formal.

Setiap elemen dari bahasa memiliki makna yang terdefinisi, sehingga pengembang dapat dengan yakin bahwa ketika pengembang memodelkan segi tertentu dari sistem, tidak akan terjadi kesalahpahaman.

2. Ringkas.

Seluruh bahasa terdiri dari notasi yang sederhana dan terus terang.

3. Komprehensif, atau luas dan lengkap.

Dapat menggambarkan semua aspek penting dari sebuah sistem.

4. *Scalable*.

Pada saat dibutuhkan, bahasa ini cukup formal untuk menangani proyek-proyek pemodelan sistem besar, tetapi bahasa ini juga menurunkannya menjadi proyek-proyek kecil, menghindari terjadinya kehancuran masal.

5. Dibangun berdasarkan pengalaman.

UML adalah puncak dari praktek-praktek terbaik dalam komunitas berorientasi objek selama 15 tahun terakhir.

#### 6. Standar

UML dikendalikan oleh kelompok standar terbuka dengan kontribusi aktif dari kelompok vendor dan akademisi di seluruh dunia. Sebuah standar memastikan *transformability* dan *interoperabilitas* dari UML, yang berarti pengembang tidak terikat dengan suatu produk tertentu.

## **BAB III METODOLOGI PENELITIAN**

### **3.1 Tempat dan Waktu Penelitian**

Penelitian ini dilakukan di Jurusan Ilmu Komputer, Fakultas Matematika dan Ilmu Pengetahuan Alam, Universitas Lampung. Penelitian ini dilaksanakan pada semester genap tahun ajaran 2015/2016.

### **3.2 Perangkat Keras**

Perangkat keras yang digunakan dalam pengembangan adalah laptop dengan spesifikasi seperti pada Tabel 3.1.

**Tabel 3.1 Spesifikasi laptop**

Processor	4 cores, 4MB cache, 1,9GHz with Turbo Core up to 2,8GHz
GPU	1GB VRAM
Display	LED 14.0" HD (1366×768)
Memory	4GB DDR3
HDD	500GB 2,5" SATA HDD

### 3.3 Perangkat Lunak

Perangkat lunak yang digunakan dalam pengembangan aplikasi ini adalah sebagai berikut:

1. Sistem Operasi Window 7 Ultimate 64-bit (6.1, Build 7601)
2. WAMP Server Versi 2.5
3. NetBeans IDE 7.3.1
4. Chrome Browser Versi 51
5. Kerangka Kerja Yii
6. Kerangka Kerja Bootstrap

### 3.4 Perancangan Basis Data

Aplikasi ini menggunakan tujuh tabel yang masing-masing digunakan untuk menyimpan data yang berbeda namun saling terhubung satu sama lain. Tabel 3.2 s.d. Tabel 3.8 di bawah ini merupakan isi dari ketujuh tabel tersebut.

**Tabel 3.2 Tabel cr\_users**

Field	Type	Key	Default	Extra
id	int(11)	primary		auto_increment
username	varchar(255)	unique		
password	varchar(255)			
activation	varchar(255)			
email	varchar(255)	unique		
wallet	int(11)		0	
created_at	datetime		0000-00-00 00:00:00	
updated_at	datetime		0000-00-00 00:00:00	
updated_by	int(11)		null	
last_login	datetime		0000-00-00 00:00:00	
status	int(11)		0	

role	int(11)		9	
------	---------	--	---	--

**Tabel 3.3 Tabel cr\_deposits**

Field	Type	Key	Default	Extra
id	int(11)	primary		auto_increment
owner_id	int(11)	index		
sender_name	varchar(255)			
amount	int(11)			
payment_date	date			
filename	int(11)		null	
status	tinyint(11)		0	
created_at	datetime		0000-00-00 00:00:00	
updated_at	datetime		0000-00-00 00:00:00	
updated_by	int(11)	index	null	

**Tabel 3.4 Tabel cr\_comics**

Field	Type	Key	Default	Extra
id	int(11)	primary		auto_increment
uniqueid	varchar(100)	unique		
owner_id	int(11)	index		
cover	varchar(255)		null	
title	varchar(255)			
description	text		null	
tags	text		null	
rating	char(2)			
count	int(11)		0	
drafted_date	datetime		0000-00-00 00:00:00	
published_date	datetime		0000-00-00 00:00:00	
hidden_date	datetime		0000-00-00 00:00:00	
created_at	datetime		0000-00-00 00:00:00	
updated_at	datetime		0000-00-00 00:00:00	
updated_by	int(11)	index	null	
deleted	tinyint(1)		0	

Hubungan:

1. cr\_comics[owner\_id] = cr\_users[id]

*Note: Cascade on Delete, Cascade on Update*

2. cr\_comics[updated\_by] = cr\_users[id]

*Note: Cascade on Delete, Cascade on Update*

**Tabel 3.5 Tabel cr\_tags**

Field	Type	Key	Default	Extra
Id	int(11)	primary		auto_increment
Name	varchar(255)			
frequency	int(11)		1	

**Tabel 3.6 Tabel cr\_chapters**

Field	Type	Key	Default	Extra
Id	int(11)	primary		auto_increment
uniqueid	varchar(100)	unique		
comic_id	int(11)	index		
Cover	varchar(255)		null	
Title	varchar(255)			
Volume	int(11)		0	
Chapter	int(11)		0	
subchapter	int(11)		0	
Price	int(11)		0	
Count	int(11)		0	
drafted_date	datetime		0000-00-00 00:00:00	
published_date	datetime		0000-00-00 00:00:00	
hidden_date	datetime		0000-00-00 00:00:00	
created_at	datetime		0000-00-00 00:00:00	
updated_at	datetime		0000-00-00 00:00:00	
updated_by	int(11)	index	null	
deleted	tinyint(1)		0	

Hubungan:

1. cr\_chapters[comic\_id] = cr\_comics[id]

*Note: Cascade on Delete, Cascade on Update*

2. cr\_chapters[updated\_by] = cr\_users[id]

*Note: Cascade on Delete, Cascade on Update*

**Tabel 3.7 Tabel cr\_cpages**

Field	Type	Key	Default	Extra
id	int(11)	primary		auto_increment
chapter_id	int(11)	index		
filename	varchar(255)			
sort	int(11)		0	
mime	varchar(255)			
size	int(11)			
height	int(11)			
width	int(11)			
created_at	datetime		0000-00-00 00:00:00	
updated_at	datetime		0000-00-00 00:00:00	
updated_by	int(11)	index	null	

Hubungan:

1. cr\_cpages[chapters\_id] = cr\_chapters[id]

*Note: Cascade on Delete, Cascade on Update*

2. cr\_cpages[updated\_by] = cr\_users[id]

*Note: Cascade on Delete, Cascade on Update*

**Tabel 3.8 Tabel cr\_transactions**

Field	Type	Key	Default	Extra
id	int(11)	primary		auto_increment
chapter_id	int(11)	primary		
comic_id	int(11)	index		
type	varchar(100)			
wallet	int(11)			
price	int(11)			
total	int(11)			
created_at	datetime			

Hubungan:

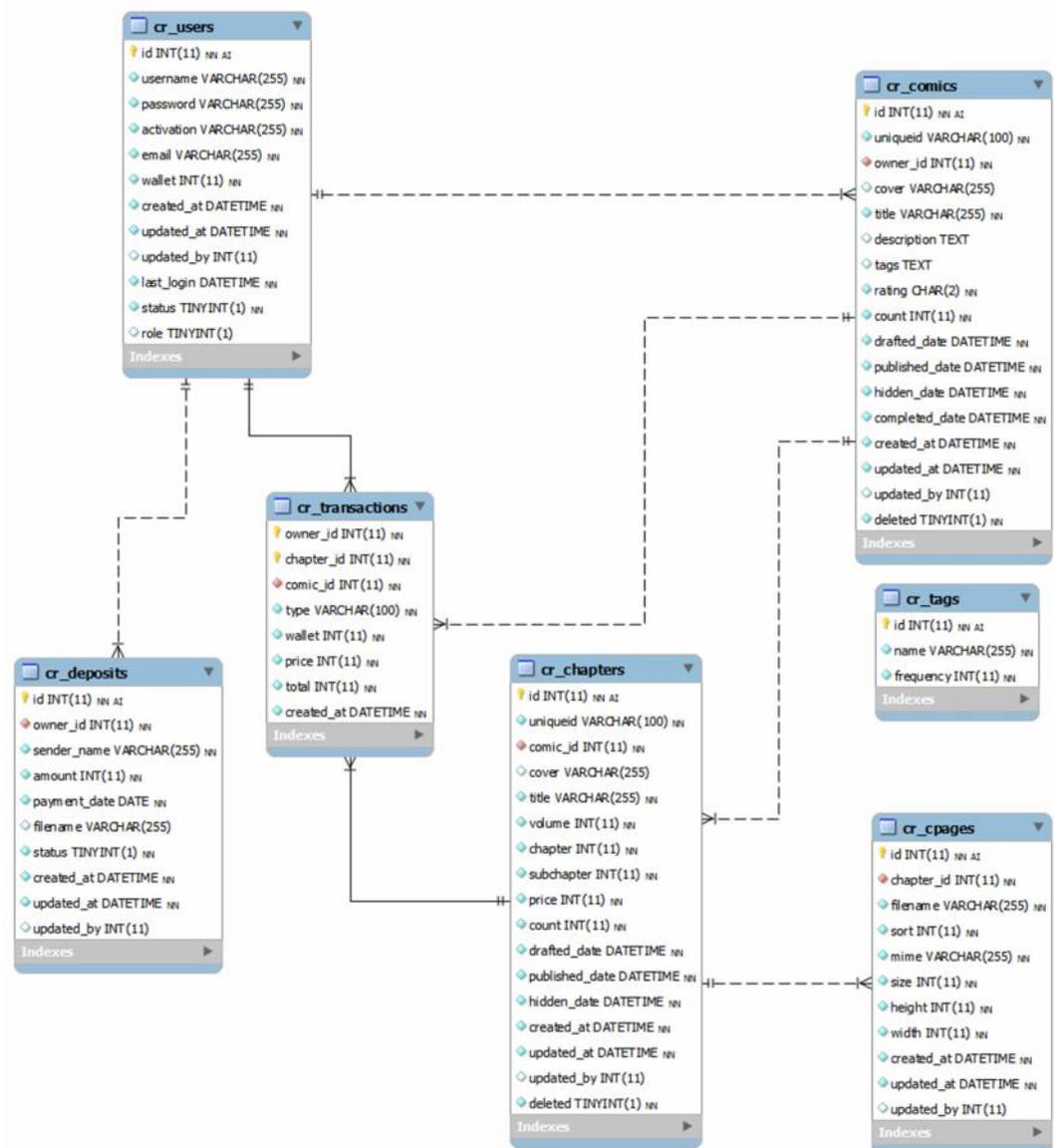
1. cr\_transactions[chapters\_id] = cr\_chapters[id]

*Note: Cascade on Delete, Cascade on Update*

2. cr\_transactions[comic\_id] = cr\_comics[id]

*Note: Cascade on Delete, Cascade on Update*

Desain skema basis data yang digunakan dalam pengembangan ditunjukkan pada Gambar 3.1.



Gambar 3.1 Skema basis data

### 3.5 Perancangan

#### 3.5.1 Definisi Peran Pengguna

Deskripsi peran pengguna ditunjukkan pada Tabel 3.9.

**Tabel 3.9 Definisi peran pengguna**

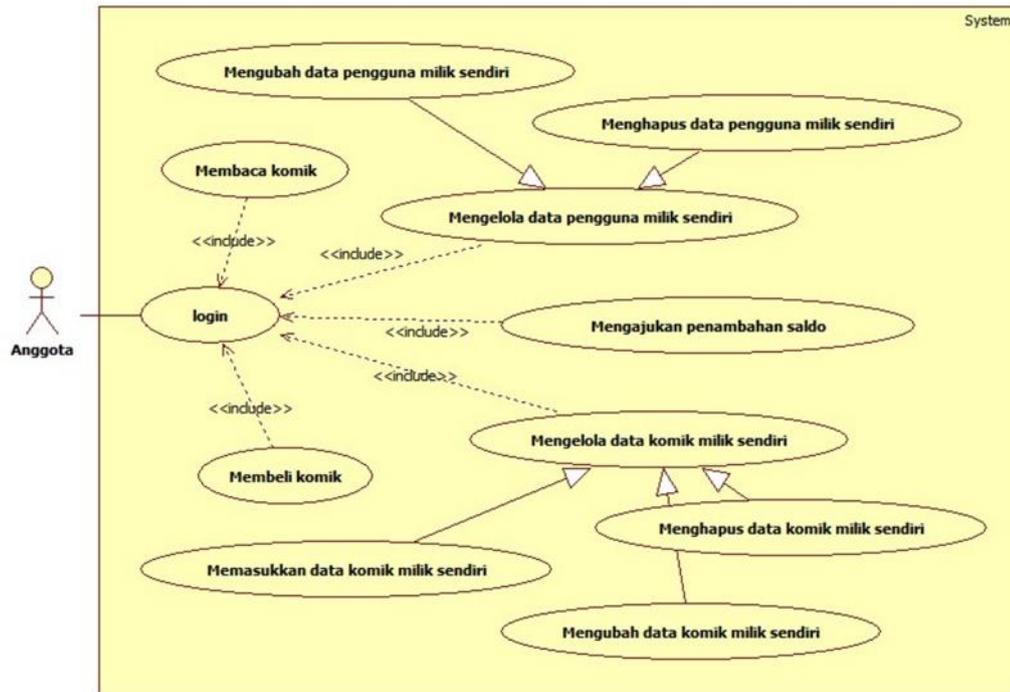
Peran	Deskripsi
<i>Guest</i> (Tamu)	Pengguna yang memiliki hak akses untuk melakukan proses <i>registrasi</i> dan <i>login</i> . Tamu termasuk dalam kategori ini
<i>Member</i> (Anggota)	Pengguna yang memiliki hak akses untuk mengelola data komik dan pengguna miliknya sendiri, misalnya membeli komik, mengunggah file komik dan mengganti password
<i>Admin</i>	Pengguna yang memiliki hak akses terluas. Mereka dapat mengelola semua data dalam aplikasi, serta bertugas untuk mengontrol jalannya aplikasi

#### 3.5.2 Diagram Use Case

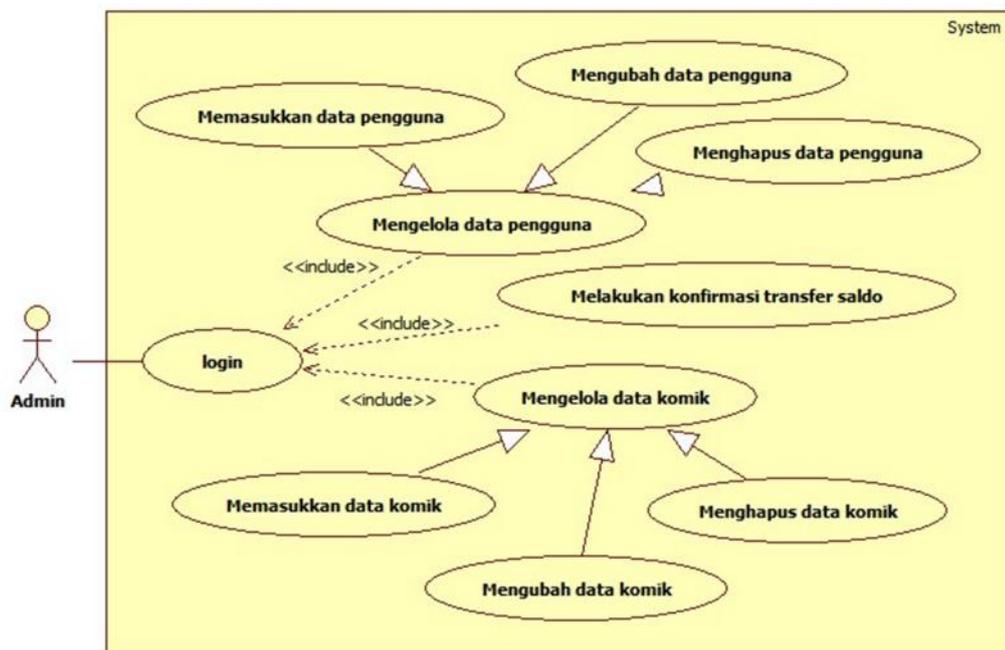
Diagram *use case* dapat dilihat pada Gambar 3.2, Gambar 3.3 dan Gambar 3.4.



**Gambar 3.2 Diagram use case tamu**



Gambar 3.3 Diagram *use case* anggota



Gambar 3.4 Diagram *use case* admin

### 3.5.3 Definisi Use Case

Deskripsi *use case* dapat dilihat di Tabel 3.10.

**Tabel 3.10 Definisi *use case***

No.	<i>Use Case</i>	Deskripsi
1	<i>Login</i>	Proses pengendalian akses pengguna ke aplikasi dengan cara, mencocokkan data yang dimasukkan oleh pengguna dengan data yang tersimpan di basis data
2	Mengelola data pengguna	Proses pengelolaan data pengguna, yaitu memasukkan data pengguna, mengubah data pengguna, dan menghapus data pengguna
3	Memasukkan data pengguna	Proses memasukkan data pengguna ke dalam basis data
4	Mengubah data pengguna	Proses mengubah data pengguna yang ada di dalam basis data
5	Menghapus data pengguna	Proses menghapus data pengguna yang ada di dalam basis data
6	Membaca data pengguna	Proses dimana aplikasi menampilkan data pengguna agar bisa dibaca oleh pengguna
7	Mengelola data komik	Proses pengelolaan data komik yaitu memasukkan data komik, mengubah data komik, dan menghapus data komik
8	Memasukkan data komik	Proses memasukkan data komik ke dalam basis data
9	Mengubah data komik	Proses mengubah data komik yang ada di dalam basis data
10	Menghapus data komik	Proses menghapus data komik yang ada di dalam basis data
11	Membeli komik	Proses memasukkan data transaksi dan saldo pengguna akan dikurangi
12	Membaca komik	Proses dimana aplikasi menampilkan data komik agar bisa dibaca oleh pengguna
13	Mencari komik	Proses dimana pengguna memasukkan kata kunci untuk mengumpulkan daftar komik yang sesuai dengan kata kunci
14	Mengajukan penambahan saldo	Proses dimana pengguna memasukkan data <i>transfer</i> saldo
15	Melakukan konfirmasi <i>transfer</i> saldo	Proses dimana pengguna melakukan pencocokan data transfer untuk proses konfirmasi dan penambahan saldo
16	Mendaftar jadi anggota	Proses pemasukan data calon anggota untuk mendapatkan akses ke dalam aplikasi

### 3.5.4 Skenario *Use Case*

Tabel 3.11 s.d. Tabel 3.23 merupakan skenario dari setiap *use case* di atas:

**Tabel 3.11 Skenario *use case login***

Aksi Pengguna	Reaksi Aplikasi
<b>Skenario Normal</b>	
1. Memasukkan <i>username</i> dan <i>password</i>	
	2. Mengecek validasi data yang dimasukkan
	3. Masuk ke aplikasi
<b>Skenario Alternatif</b>	
1. Memasukkan <i>username</i> dan <i>password</i>	
	2. Mengecek validasi data yang dimasukkan
	3. Menampilkan pesan validasi gagal

**Tabel 3.12 Skenario *use case memasukkan data pengguna***

Aksi Pengguna	Reaksi Aplikasi
<b>Skenario Normal</b>	
1. Memasukkan data pengguna sesuai kolom yang ada	
	2. Mengecek validasi data yang dimasukkan
	3. Menyimpan data pengguna ke basis data
	4. Menampilkan pesan data berhasil disimpan
<b>Skenario Alternatif</b>	
1. Memasukkan data pengguna yang tidak valid	
	2. Mengecek validasi data yang dimasukkan
	3. Menampilkan pesan validasi gagal

**Tabel 3.13 Skenario *use case mengubah data pengguna***

Aksi Pengguna	Reaksi Aplikasi
<b>Skenario Normal</b>	
1. Memilih data pengguna yang akan	

diubah	
	2. Mencari data pengguna di basis data
	3. Menampilkan data pengguna
4. Menekan tombol untuk mengubah data	
	5. Menampilkan kolom data dalam bentuk form
6. Memasukkan data baru pada kolom yang akan dirubah	
	7. Mengecek validasi data yang dimasukkan
	8. Menyimpan data ke basis data
	9. Menampilkan pesan data berhasil disimpan
<b>Skenario Alternatif</b>	
1. Memilih data pengguna yang akan diubah	
	2. Mencari data pengguna di basis data
	3. Menampilkan data pengguna
4. Menekan tombol untuk mengubah data	
	5. Menampilkan kolom data dalam bentuk form
6. Memasukkan data pengguna yang baru dan tidak valid pada kolom yang akan dirubah	
	7. Mengecek validasi data yang dimasukkan
	8. Menampilkan pesan validasi gagal

**Tabel 3.14 Skenario *use case* menghapus data pengguna**

Aksi Pengguna	Reaksi Aplikasi
<b>Skenario Normal</b>	
1. Memilih data pengguna yang akan dihapus	
	2. Mencari data pengguna di basis data
	3. Menampilkan data pengguna
4. Menekan tombol untuk menghapus data	
	5. Menampilkan pesan konfirmasi penghapusan data
6. Menekan tombol persetujuan untuk melanjutkan proses penghapusan	
	7. Menghapus data pengguna dari basis data
	8. Menampilkan pesan data berhasil dihapus
<b>Skenario Alternatif</b>	
1. Memilih data pengguna yang akan	

dihapus	
	2. Mencari data pengguna di basis data
	3. Menampilkan data pengguna
4. Menekan tombol untuk menghapus data	
	5. Menampilkan pesan konfirmasi penghapusan data
6. Menekan tombol penolakan untuk melanjutkan proses penghapusan	
	7. Kembali ke tampilan data pengguna

**Tabel 3.15 Skenario *use case* memasukkan data komik**

Aksi Pengguna	Reaksi Aplikasi
<b>Skenario Normal</b>	
1. Memasukkan data komik sesuai kolom yang ada	
	2. Mengecek validasi data yang dimasukkan
	3. Menyimpan data komik ke basis data
	4. Menampilkan pesan data berhasil disimpan
<b>Skenario Alternatif</b>	
1. Memasukkan data komik yang tidak valid	
	2. Mengecek validasi data yang dimasukkan
	3. Menampilkan pesan validasi gagal

**Tabel 3.16 Skenario *use case* mengubah data komik**

Aksi Pengguna	Reaksi Aplikasi
<b>Skenario Normal</b>	
1. Memilih data komik yang akan diubah	
	2. Mencari data komik di basis data
	3. Menampilkan data komik
4. Menekan tombol untuk mengubah data	
	5. Menampilkan kolom data dalam bentuk form
6. Memasukkan data komik yang baru pada kolom yang akan dirubah	
	7. Mengecek validasi data yang dimasukkan
	8. Menyimpan data ke basis data
	9. Menampilkan pesan data berhasil disimpan

<b>Skenario Alternatif</b>	
1. Memilih data komik yang akan diubah	
	2. Mencari data komik di basis data
	3. Menampilkan data komik
4. Menekan tombol untuk mengubah data	
	5. Menampilkan kolom data dalam bentuk form
6. Memasukkan data komik yang baru dan tidak valid pada kolom yang akan dirubah	
	7. Mengecek validasi data yang dimasukkan
	8. Menampilkan pesan validasi gagal

**Tabel 3.17 Skenario *use case* menghapus data komik**

Aksi Pengguna	Reaksi Aplikasi
<b>Skenario Normal</b>	
1. Memilih data komik yang akan dihapus	
	2. Mencari data komik di basis data
	3. Menampilkan data komik
4. Menekan tombol untuk menghapus data	
	5. Menampilkan pesan konfirmasi penghapusan data
6. Menekan tombol persetujuan untuk melanjutkan proses penghapusan	
	7. Menghapus data komik dari basis data
	8. Menampilkan pesan data berhasil dihapus
<b>Skenario Alternatif</b>	
1. Memilih data komik yang akan dihapus	
	2. Mencari data komik di basis data
	3. Menampilkan data komik
4. Menekan tombol untuk menghapus data	
	5. Menampilkan pesan konfirmasi penghapusan data
6. Menekan tombol penolakan untuk melanjutkan proses penghapusan	
	7. Kembali ke tampilan data komik

**Tabel 3.18 Skenario *use case* membeli komik**

Aksi Pengguna	Reaksi Aplikasi
<b>Skenario Normal</b>	
1. Menekan tombol untuk membeli	
	2. Mencari data komik di basis data
	3. Mencari data pengguna di basis data
	4. Mengecek validasi data
	5. Menyimpan data transaksi ke basis data
	6. Menampilkan pesan data berhasil disimpan
<b>Skenario Alternatif</b>	
1. Menekan tombol untuk membeli	
	2. Mencari data komik di basis data
	3. Mencari data pengguna di basis data
	4. Mengecek validasi data
	5. Menampilkan pesan validasi gagal

**Tabel 3.19 Skenario *use case* membaca komik**

Aksi Pengguna	Reaksi Aplikasi
<b>Skenario Normal</b>	
1. Memilih data komik yang akan dibaca	
	2. Mencari data komik di basis data
	3. Menampilkan data komik
<b>Skenario Alternatif</b>	
1. Memilih data komik yang akan dibaca	
	2. Mencari data komik di basis data
	3. Menampilkan pesan gagal menampilkan data komik

**Tabel 3.20 Skenario *use case* mencari komik**

Aksi Pengguna	Reaksi Aplikasi
<b>Skenario Normal</b>	
1. Memasukkan kata kunci	
	2. Mencari komik yang sesuai dengan kata kunci di basis data
	3. Menampilkan daftar komik yang memenuhi syarat dan sesuai dengan kata kunci
4. Memilih komik yang dicari	
	5. Mencari data komik di basis data
	6. Menampilkan komik

<b>Skenario Alternatif</b>	
1. Memasukkan kata kunci	
	2. Mencari komik yang sesuai dengan kata kunci di basis data
	3. Menampilkan pesan bahwa komik tidak ada di dalam basis data

**Tabel 3.21 Skenario *use case* mengajukan penambahan saldo**

Aksi Pengguna	Reaksi Aplikasi
<b>Skenario Normal</b>	
1. <i>Transfer</i> uang ke rekening	
2. Memasukkan data sesuai kolom yang ada	
	3. Mengecek validasi data yang dimasukkan
	4. Menyimpan data ke basis data
	5. Menampilkan pesan data berhasil disimpan
<b>Skenario Alternatif</b>	
1. <i>Transfer</i> uang ke rekening	
2. Memasukkan data sesuai kolom yang ada	
	3. Mengecek validasi data yang dimasukkan
	4. Menampilkan pesan validasi gagal

**Tabel 3.22 Skenario *use case* melakukan konfirmasi *transfer* saldo**

Aksi Pengguna	Reaksi Aplikasi
<b>Skenario Normal</b>	
1. Mengecek data <i>transfer</i>	
2. Menekan tombol untuk mengubah data	
	3. Mencari data di basis data
	4. Mengubah data di basis data
	5. Menampilkan pesan data diterima
<b>Skenario Alternatif</b>	
1. Mengecek data <i>transfer</i>	
2. Menekan tombol untuk mengubah data	
	3. Mencari data di basis data
	4. Mengubah data di basis data
	5. Menampilkan pesan data ditolak

**Tabel 3.23 Skenario *use case* mendaftar jadi anggota**

Aksi Pengguna	Reaksi Aplikasi
<b>Skenario Normal</b>	
1. Memasukkan data pada kolom-kolom yang tersedia di form	
	2. Mengecek validasi data yang dimasukkan
	3. Menyimpan data ke basis data
	4. Menampilkan pesan data berhasil disimpan
<b>Skenario Alternatif</b>	
1. Memasukkan data tidak valid pada kolom-kolom yang tersedia di form	
	2. Mengecek validasi data yang dimasukkan
	3. Menampilkan pesan validasi gagal

## **BAB V**

### **KESIMPULAN DAN SARAN**

#### **5.1 Kesimpulan**

Dari hasil penelitian yang telah dilakukan, ada beberapa kesimpulan yang di dapat:

1. Metodologi pengembangan sistem formal seperti RUP tidak cocok diterapkan oleh pengembang yang bekerja sendiri. Metodologi seperti ini lebih cocok diterapkan oleh sebuah tim pengembang, karena aturan yang ada, dibuat agar pekerjaan tim menjadi lebih efisien serta tugas dan tenggat waktu setiap anggota tim atau divisi menjadi lebih jelas, sehingga mengurangi resiko konflik dalam tim yang dapat menunda pekerjaan dan mengakibatkan kerugian materi bagi perusahaan. RUP sendiri dibuat oleh IBM, yang merupakan sebuah perusahaan.
2. *Framework* Yii mudah digunakan bila sudah dipelajari, namun butuh proses belajar yang panjang pula agar lebih mengenal segala fitur yang disediakan oleh *framework* ini. Yii memiliki banyak fitur dengan fungsi utamanya adalah untuk mempercepat kerja *coding*, maka dari itu, akan sangat disayangkan bila fitur tersebut tidak digunakan pada aplikasi berbasis Yii yang dikembangkan.

## 5.2 Saran

Dari hasil penelitian yang telah dilakukan, saran yang bisa disampaikan adalah sebagai berikut:

1. Penambahan fitur lupa *password* dan aktivasi akun melalui email. *Field* untuk menyimpan kunci aktivasi sudah tersedia di table *cr\_users*, namun belum dimanfaatkan secara optimal.
2. Pemanfaatan *Role-Based Access Control* (RBAC), bisa menggunakan RBAC bawaan Yii, ataupun menggunakan extensi yang dikembangkan oleh pengguna Yii lain. Saat ini, aplikasi ini masih menggunakan sistem otentikasi buatan sendiri yang simple karena peran pengguna yang digunakan sampai saat ini hanya dua jenis peran, yaitu admin dan anggota.
3. Penambahan fitur rating berupa *favorite* atau *like* untuk komik.
4. Penambahan fitur komen untuk komik.

## DAFTAR PUSTAKA

- Bonneff, Marcel. 1998. *Komik Indonesia*. Kepustakaan Populer Gramedia. Jakarta. 226 hlm.
- Duncan, Randy dan Mathew J. Smith. 2009. *The Power of Comics: History, Form and Culture*. The Continuum International Publishing Group. Amerika Serikat. 346 hlm.
- Hamilton, Kim dan Russell Miles. 2006. *Learning UML 2.0*. O'Reilly. Amerika Serikat. 286 hlm.
- IBM Staff. 2005. *Rational Unified Process: Best practices for software development teams*. Diakses 22 Januari 2014, 09:30 WIB. <http://www.ibm.com/developerworks/rational/library/253.html>
- Pitt, Chris. 2012. *Pro PHP MVC*. Apress. Amerika Serikat. 479 hlm.
- Rumbaugh, James, Ivar Jacobson dan Grady Booch. 2005. *The Unified Modeling Language Reference Manual, Second Edition*. Addison-Wesley. Amerika Serikat. 742 hlm.
- Ullman, Larry. 2014. *The Yii Book, Revision: 0.9*. 575 hlm.
- Vaswani, Vikram. 2009. *PHP: A Beginner's Guide*. The McGraw-Hill Companies. Amerika Serikat. 478 hlm.