BAB III

IMPLEMENTASI DAN PEMBAHASAN

3.1 Implementasi dan Pembahasan Sistem

Setelah proses analisis dan perancangan sistem, proses selanjutnya adalah penerapan sistem.

1. Menghubungkan Program dengan Basis Data (*Data Base*)

   Sistem ini dibangun menggunakan *PHP – Yii Framework* versi 1.1.15. Untuk menghubungkan program dengan *data base*, diperlukan pengaturan di dalam file *main.php*. Cuplikan kode program penghubung *data base* sebagai berikut:

```
'db'=>array(
      'connectionString' =>
         'mysql:host=localhost;dbname=komik',
         //dbname adalah nama database
   'emulatePrepare' => true,
   'username' => 'root', //username database
   'password' => '', //password database
   'charset' => 'utf8',
),
```

   Bagian utama dari cuplikan program tersebut adalah `'connectionString'=>'mysql:host=localhost;dbname=komik'`, `'username'=>'root'`, dan `'password'=>''`. Bagian `'connectionString'` mendefinisikan jenis *data base* yang digunakan: *Mysql*; nama *host*: localhost; dan nama *data base*: komik. Bagian `'username'` mendefinisikan nama user dari *Mysql*,

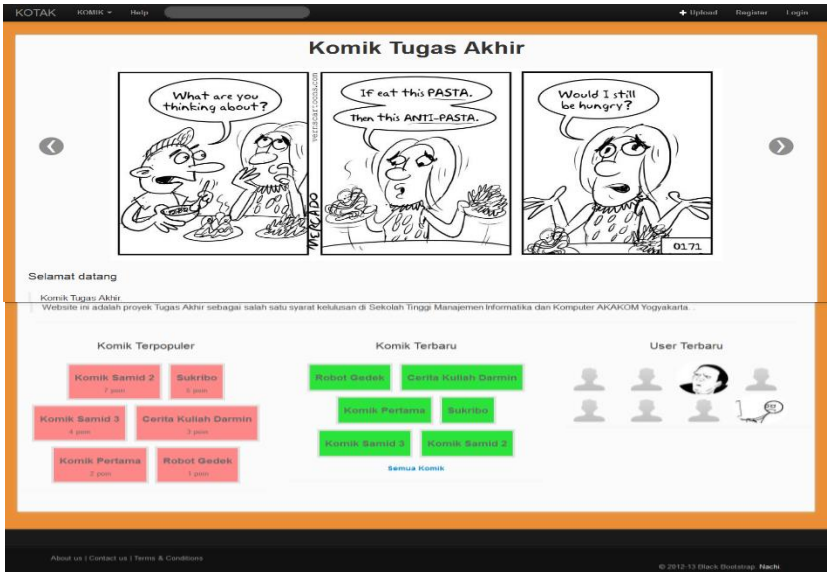
yaitu `'root'`. Bagian `'password'` mendifinisikan kata sandi dari *Mysql*.

2. Halaman Utama (*Index*)

Halaman *index* adalah halaman awal yang ditampilkan ketika pertama kali program dijalankan. Halaman ini dapat diakses oleh semua jenis pengguna. Untuk menampilkan halaman ini, program memanggil fungsi *actionIndex()* di dalam *controller file SiteController.php*. Berikut adalah cuplikan kode program dari fungsi *actionIndex()*.

```
public function actionIndex(){
$komikPoin = Yii::app()->db->createCommand()
->select('komik.ID as ID, komik.judul as judul,
  sum(jumlah) as total')->from('komik')
->join('chapter','komik.ID = chapter.id_komik')
->join('poin', 'chapter.ID = poin.id_chapter')
->where('komik.published=2')->group('komik.ID')
->order('total desc')->limit(6)->query();
$komikTerbaru = Komik::model()->findAllByAttributes
            (array('published'=>2),array('order'=>'id
            desc','limit'=>6));
$userTerbaru = User::model()->findAll
            (array('order'=>'id desc', 'limit'=>8));
//..... baris kode program sesudahnya
}
```

Dalam cuplikan kode program tersebut, terdapat tiga variabel yaitu `$komikPoin`, `$komikTerbaru`, dan `$userTerbaru`. Variabel `$komikPoin` berfungsi untuk menampilkan enam komik dengan poin tertinggi, varabel `$komikTerbaru` berfungsi untuk menampilkan enam komik terbaru, dan variabel `$userTerbaru` berfungsi untuk menampilkan delapan pengguna terbaru. Hasil dari cuplikan program tersebut ditunjukkan oleh Gambar 3.1



Gambar 3.1 Tampilan Halaman *Index*

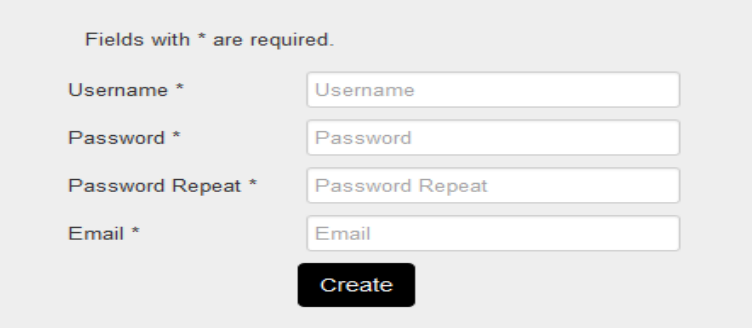
3. Halaman Registrasi

Halaman Registrasi adalah halaman yang disajikan untuk pengguna tamu yang ingin mendaftar sebagai anggota. Untuk memanggil halaman ini, program memanggil fungsi *actionRegister()* yang ada di dalam *controller file UserController.php*. Berikut adalah cuplikan kode program dari fungsi *actionRegister()*.

```
public function actionRegister(){
$model=new User('register');
$model->level = 2;
$model->avatar = 'default.png';
$this->layout = '//layouts/column1';
if (Yii::app()->user->id){$this->
redirect(array('site/index'));}
if(isset($_POST['User'])){
    $model->attributes=$_POST['User'];
  if($model->save())$this->redirect(array('site/login'));
        }
    //..... baris kode program sesudahnya
  }
```

Ketika fungsi tersebut dijalankan, objek dari kelas *User* dibuat sehingga program dapat mengakses atribut yang ada di kelas

User. Selanjutnya, variabel 'level' diisi nilai '2' dan variabel 'avatar' diisi dengan nilai 'default.png'. Setelah semua *field* telah diisi dan diverifikasi, nilai dari *field* disimpan ke *data base*. Tampilan halaman Registrasi ditunjukkan pada Gambar 3.2.



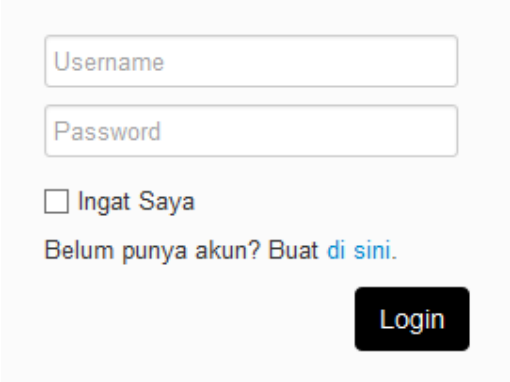
Gambar 3.2 Tampilan Halaman Registrasi

4. Halaman Login

Halaman *Login* adalah halaman yang disajikan untuk pengguna yang ingin melakukan proses autentikasi agar dapat melakukan proses tertentu berdasarkan hak aksesnya, '1' untuk *Admin* dan '2' untuk *Member*. Untuk menampilkan halaman ini, program memanggil fungsi *actionLogin()* dari *controller file SiteController*. Berikut cuplikan kode program dari fungsi *actionLogin()*.

```
//..... baris kode program sebelumnya
if(isset($_POST['LoginForm'])){
$model->attributes=$_POST['LoginForm'];
if($model->validate() && $model->login())
$this->redirect(Yii::app()->user->returnUrl);}
//..... baris kode program sesudahnya
```

Tampilan Halaman login ditunjukkan pada Gambar 3.3.

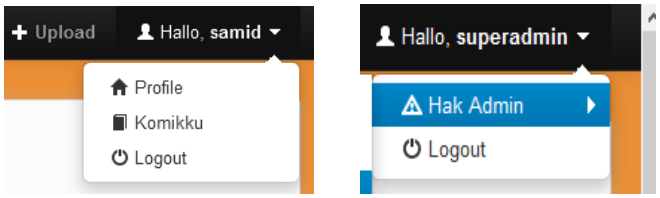


Gambar 3.3 Tampilan Halaman *Login*

Agar proses penentuan hak akses pengguna dapat berjalan, diperlukan suatu komponen untuk melakukannya. Komponen tersebut diberi nama *EWebUser.php* dan diletakkan pada *C:\xampp\htdocs\komik\protected\components*. Berikut kode program dari *EWebUser.php*.

```
<?php
class EWebUser extends CWebUser{
    protected $_model;
     function isAdmin(){ //fungsi untuk menentukan admin
        $user = $this->loadUser();
        if ($user) return $user->level==1;
        return false;
    }
    function isMember(){ //fungsi untuk menentukan member
        $user = $this->loadUser();
        if ($user)return $user->level==2;
        return false;
    }
     protected function loadUser(){
        if ($this->_model===null) {
      $this->_model = User::model()->findByPk($this->ID);
     }
        return $this->_model;
    }
}
```

Hasil penentuan hak akses untuk Member ditunjukkan pada Gambar 3.4 (a), sedangkan hak akses untuk Admin ditunjukkan pada Gambar 3.4 (b).

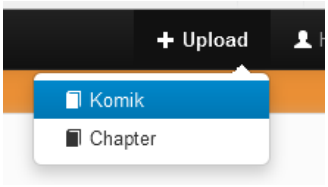


(a) (b)

Gambar 3.4 Hasil Penentuan Hak Akses

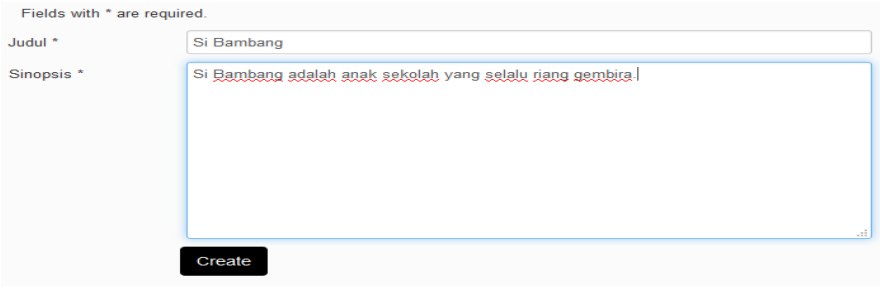
5. Membuat Komik Baru

Ketika telah berhasil melakukan *login*, pengguna dapat membuat komik baru dengan cara memilih menu *Upload* – Komik. Menu pilihan membuat komik baru ditunjukkan pada Gambar 3.5.



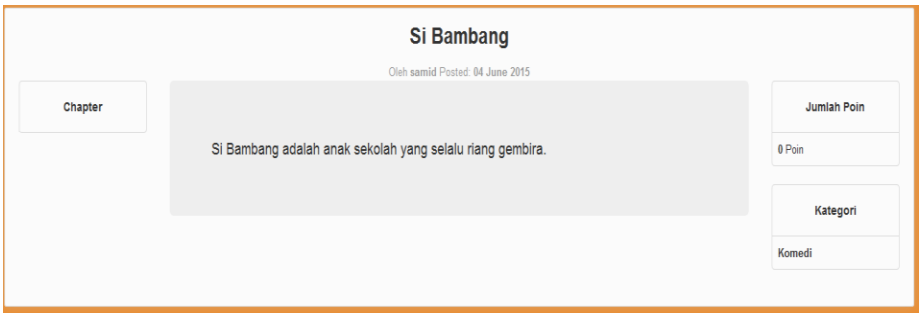
Gambar 3.5 Menu Membuat Komik Baru

Setelah menu membuat komik baru dipilih, muncul halaman untuk mengisikan Judul, Sinopsis, dan Kategori komik yang akan dibuat. Pengguna dapat memilih lebih dari satu kategori. Prosesnya ditunjukkan pada Gambar 3.6, Gambar 3.7, dan Gambar 3.8.



Gambar 3.6 Mengisi Judul dan Sinopsis Komik

Gambar 3.7 Memilih Kategori Komik



Gambar 3.8 Hasil Komik yang Dibuat

Untuk membuat komik baru, program memanggil menjalankan fungsi *actionCreate()* dari file *KomikController.php* dan *KomikkategoriController.php*. Berikut cuplikan kode program untuk membuat komik baru.

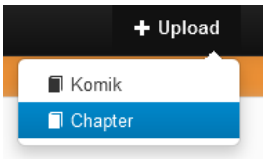
```
//KomikController.php
//..... baris kode program sebelumnya
$model->published = 1;
$model->id_user = Yii::app()->user->id;
if(isset($_POST['Komik'])){
$model->attributes=$_POST['Komik'];
if($model->save())
$this->redirect(array('komikkategori/create',
      'id'=>$model->ID
      ));
}
//..... baris kode program sesudahnya
```

```
// KomikkategoriController.php
  //..... baris kode program sebelumnya
  if(isset($_POST['Komikkategori'])){
$id_kategori= $_POST['Komikkategori']['id_kategori'];
if (isset($id_kategori) && count($id_kategori) > 0) {
  foreach ($id_kategori as $index => $value) {
  $model = new Komikkategori;
```

```
$model->attributes=$_POST['Komikkategori'];
$model->id_komik = $_GET['id'];
$model->id_kategori = (int)$value;
$model->save(); }}
if($model->save())$this->redirect(array('komik/view',
     'id'=>$model->id_komik));
}
```
//..... baris kode program sesudahnya

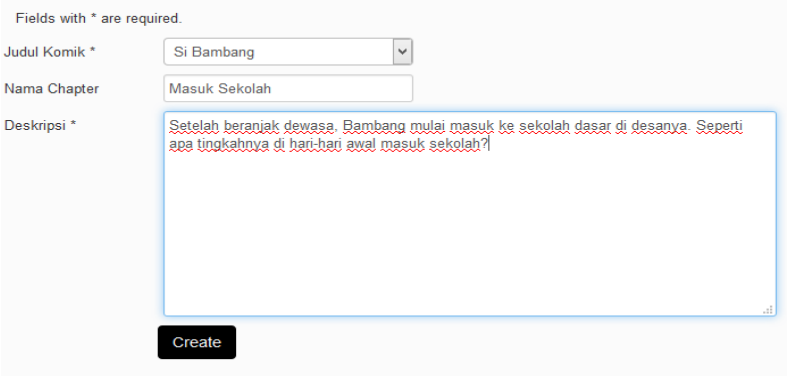
6. Membuat Chapter Baru

Setelah komik dibuat, pengguna dapat membuat Chapter baru melalui menu yang ditunjukkan pada Gambar 3.9.



Gambar 3.9 Menu Membuat Chapter Baru

Proses berikutnya adalah memilih Judul Komik yang telah dibuat, Nama Chapter, dan Deskripsi Chapter. Prosesnya ditunjukkan pada Gambar 3.10, sedangkan hasilnya ditunjukkan pada Gambar 3.11.



Gambar 3.10 Proses Pengisian Data Chapter

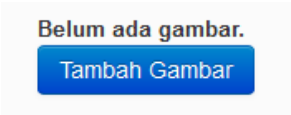
Gambar 3.11 Hasil Chapter yang Dibuat

Untuk melakukan proses diatas, program menjalankan fungsi *actionCreate()* dari *file ChapterController.php*. Berikut cuplikan kode program dari fungsi actionCreate().

```
//..... baris kode program sebelumnya
if($model->save())$this->redirect(array(
      'view','id'=>$model->ID,
      'komik'=>$model->id_komik));   }
//..... baris kode program sesudahnya
```

7. Menambahkan Gambar pada Chapter

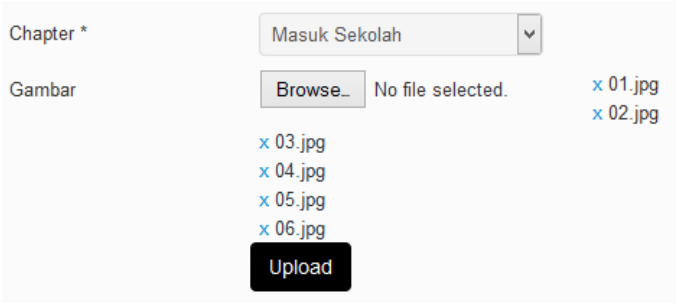
Setelah Chapter dibuat, pengguna dapat menambahkan gambar dengan cara menekan tombol 'Tambah Gambar' pada halaman *chapter* yang ditunjukkan pada Gambar 3.12.



Gambar 3.12 Tombol Tambah Gambar

Selanjutnya, pengguna memilih nama Chapter dan mulai mengunggah gambar. Gambar yang diunggah dapat berjumlah satu atau lebih. Gambar-gambar yang telah diunggah dikemas dalam satu halaman Chapter yang berkaitan. Gambar dengan

urutan pertama dalam proses pengunggahan menjadi gambar *preview* atau sering disebut *thumbnail*. Proses ditunjukkan pada Gambar 3.13 dan hasilnya ditunjukkan pada Gambar 3.14.



Gambar 3.13 Proses Mengunggah Gambar

Gambar 3.14 Hasil Gambar yang Diunggah

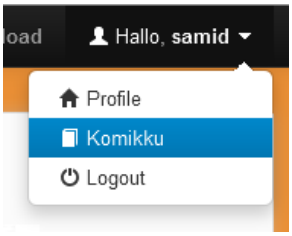
Untuk dapat melakukan proses tersebut, program menjalankan fungsi *actionCreate()* dari *file ChapterController.php*. Berikut cuplikan kode program dari fungsi *actionCreate()*.

```
    //..... baris kode program sebelumnya
    if(isset($_POST['Gambar'])){
    $images = CUploadedFile::getInstancesByName('gambar');
    if (isset($images) && count($images) > 0) {
    foreach ($images as $image => $pic) {
    $random_num =
rand(0,9999).'_'.rand(0,9999).'_'.rand(0,9999);
    $filename = $random_num.'_'.$image.'.jpg';
if ($pic->saveAs(
Yii::getPathOfAlias('webroot').'/images/komik/'.$filename))
{
          $model = new Gambar;
          $model->attributes=$_POST['Gambar'];
```

```
            $model->gambar = $filename;
            $model->save();
            }}
    }
     //..... baris kode program sesudahnya
```

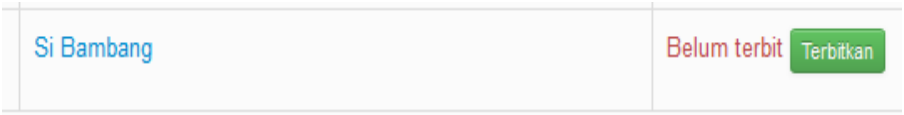
8. Menerbitkan Komik yang Telah Dibuat

Komik yang telah dibuat lengkap dengan *chapter* dan gambarnya dapat diterbitkan melalui menu 'Komikku' seperti ditunjukkan pada Gambar 3.15.



Gambar 3.15 Menu Melihat Kumpulan Karya Komik

Setelah menu tersebut dipilih, muncul daftar komik yang telah dibuat. Dari daftar komik yang ada, sebuah komik dapat diterbitkan dengan cara menekan tombol 'Terbitkan' yang ditunjukkan pada Gambar 3.16.



Gambar 3.16 Nama Komik dan Tombol Terbitkan

Untuk dapat melakukan proses tersebut, program menjalankan fungsi *actionTerbit()* dari file *KomikController.php*. Berikut cuplikan program dari fungsi actionTerbit().

```
    //..... baris kode program sebelumnya
$terbit = Yii::app()->db->createCommand()-> update(
'komik',array('published'=>2),'ID=:id',array(':id'=>$id)
);
    //..... baris kode program sesudahnya
```

9. Melihat Gambar Komik

Untuk dapat melihat semua gambar dari *Chapter* yang bersangkutan, pengguna dapat menekan tombol berwarna oren dengan ikon mata yang terletak di bawah kanan gambar. Gambar *preview* dan tombol tersebut ditunjukkan pada Gambar 3.17.

Gambar 3.17 Tombol Lihat Gambar *Chapter*

Setelah tombol diklik, akan muncul tampilan halaman melayang (*overlay*) berisi gambar-gambar yang telah diunggah sebelumnya dengan navigasi kanan kiri untuk melihat gambar selanjutnya. Tampilan halaman melayang ditunjukkan pada Gambar 3.18.

Gambar 3.18 Tampilan Halaman Melayang (Overlay)

Tampilan halaman melayang yang ditunjukkan oleh Gambar 3.18 menggunakan ekstensi (*extension*) dari *YII Framework*

bernama Magnicifig Popup. Dengan menggunakan ekstensi ini, program tidak perlu memuat gambar per halaman.

Untuk menggunakan ekstensi tersebut, diperlukan kode program seperti berikut yang diletakkan di dalam satu *file view*.

```
<?php
$src = array();
$root = Yii::app()->baseUrl;
$path = "/images/komik/";
foreach ($gambar as $key)
{$src[] = array('src' => $root.$path.$key->gambar);}
$this->widget("ext.magnific-popup.EMagnificPopup", array(
'target' => '.target', //nama class untuk tag <a>
'options' => array('gallery' => array('enabled' => true,),
'items' =>$src,),
));
?>
```

Selanjutnya nama target dijadikan nama *class* dalam tag <a> seperti kode program berikut.

```
<a class="target" href="#" ><!-- isi tag <a> --></a>
```

10. Menyukai Suatu Chapter

Setiap *Chapter* berhak mendapat apresiasi berupa poin dari pengguna yang telah membacanya. Pengguna yang dapat memberikan poin adalah pengguna terdaftar dan berhasil melakukan proses *login*. Poin yang diberikan hanya berjumlah satu poin setiap pengguna dan pengguna yang telah memberikan poin tidak bisa memberikan lagi. Untuk memberikan poin, klik tombol berwarna biru dengan ikon panah ke atas yang terletak di kiri bawah gambar *preview*. Tombol tersebut ditunjukkan pada Gambar 3.19.

Gambar 3.19 Tombol Memberikan Poin

Setelah tombol ditekan, muncul peringatan apakah pengguna pernah memberikan poin atau belum. Gambar 3.20(a) menunjukkan peringatan pengguna belum pernah memberikan poin sebelumnya, sedangkan Gambar 3.20(b) menunjukkan peringatan pengguna pernah memberikan poin sebelumnya.

(a) (b)

Gambar 3.20 Peringatan Setelah Memberikan Poin

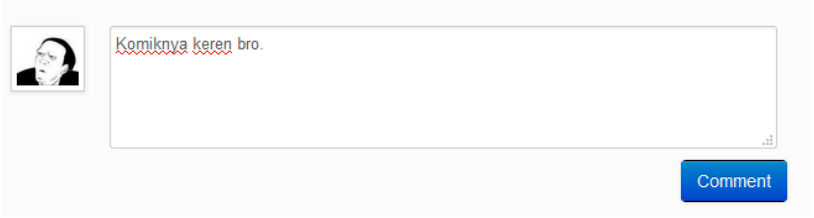
Untuk dapat memprosesnya, program memanggil *actionUpVote()* yang ada di *file ChapterController()*. Berikut adalah cuplikan kode program dari *actionUpVote()*.

```
//..... baris kode program sebelumnya
$poinChapter = Poin::model()-> findByAttributes(
array('id_chapter'=>$id,'id_user'=>Yii::app()->user->id));
//..... baris kode program sesudahnya
if (empty($poinChapter)) {
$poin->id_user=Yii::app()->user->id;
$poin->id_chapter=$id;
$poin->jumlah = 1;
$poin->save();
}
//..... baris kode program sesudahnya
```

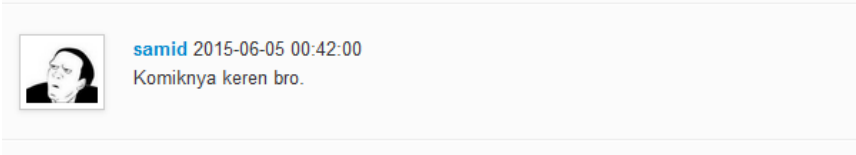
11. Memberikan Komentar pada Chapter

Pengguna terdaftar yang telah melakukan login juga dapat memberikan komentar pada suatu Chapter melalui *form* komentar yang ada di halaman Chapter. Tampilan *form*

ditunjukkan pada Gambar 3.21 dan hasil komentar ditunjukkan pada Gambar 3.22.



Gambar 3.21 Tampilan *Form* Komentar



Gambar 3.22 Hasil Komentar

Program menjalankan fungsi *actionView()* dari *ChapterContoller.php* agar muncul tampilan *form*. Berikut kode program untuk memasukkan isi komentar ke *data base*.

```
//.... baris kode program sebelumnya
if ($model->save()) $isi = $_POST['Chapter']['isi'];
   if (!empty($isi)){
         $komen = new KomentarKomik;
         $komen->id_user = Yii::app()->user->id;
         $komen->id_chapter = $model->ID;
         $komen->isi = $isi;
         $komen->save(false);
   } $this->refresh();
//..... baris kode program sesudahnya
```

3.2 Kelebihan dan Kekurangan Sistem

Kelebihan dari sistem ini adalah:

a) Menampilkan urutan komik berdasarkan jumlah semua poin dari *chapter* yang dimiliki.

b) Tampilan halaman melayang (*overlay*) untuk gambar komik.

c) Mengunggah lebih dari satu gambar komik dalam satu waktu.

d) Memberikan komentar dan poin di suatu chapter.

Kekurangan dari sistem ini adalah:

a) Belum ada sistem pelaporan data statistik secara fisik mengenai konten tertentu, misalnya jumlah pembaca komik dalam satu pekan.

b) Belum ada fasilitas untuk melaporkan konten yang bersifat tidak pantas ke admin. Akan tetapi, *admin* masih bisa menghapusnya.

3.3 Hasil Penyelesaian Kebutuhan

Tabel 3.1 Hasil Penyelesaian Kebutuhan

KODE	KEBUTUHAN	KETERANGAN	STATUS
F-01	Fasilitas registrasi (*register*)	Digunakan untuk menyimpan data dari pengguna agar dapat menjadi *member*.	Selesai
F-02	Fasilitas *login*	Setiap pengguna yang ingin mengunggah karya, memberi poin, atau mengomentari harus melewati proses autentikasi di fasilitas ini.	Selesai
F-03	Membuat, mengubah, dan menghapus komik	Membuat komik baru, mengubah, dan menghapus komik miliknya.	Selesai
F-04	Membuat, mengubah, dan menghapus *chapter*	Membuat *chapter* baru berdasarkan komik yang telah dibuat, mengubah, dan menghapusnya.	Selesai

KODE	KEBUTUHAN	KETERANGAN	STATUS
F-05	Fasilitas *upload* gambar (*create*)	Mengunggah gambar di dalam *chapter* miliknya yang telah dibuat sebelumnya.	Selesai
F-06	Fasilitas poin	Memberikan satu poin pada *chapter* miliknya sendiri atau pengguna lain.	Selesai
F-07	Fasilitas komentar	Memberikan komentar pada *chapter* miliknya atau pengguna lain.	Selesai
F-08	Fasilitas terbitkan komik	Fasilitas ini berfungsi untuk menerbitkan komik.	Selesai
F-09	Penentuan hak akses pengguna	Hak akses awal (*defaul*t) pengguna adalah *member* biasa. Akan tetapi *admin* dapat memberikan hak akses yang sama (hak akses *admin*).	Selesai
F-10	Pencarian	Fasilitas ini digunakan untuk melakukan pencarian komik	Selesai