
Rancang Bangun *Admin Management* pada Aplikasi tetanggaBaik Sebagai Jejaring Sosial untuk Perumahan di Indonesia Berbasis PHP

Rico Nova Suprayoto, Undergraduate Student Teknik Informatika, Universitas Ciputra Surabaya, UC Town, CitraLand, Surabaya 60219

Adi Suryaputra Paramita, Dosen Teknik Informatika, Universitas Ciputra Surabaya, UC Town, CitraLand, Surabaya 60219

ABSTRAK

Seiring dengan berkembangnya pembangunan di Indonesia terutama pada bisnis *property*, semakin banyak juga orang yang mulai memilih untuk bertempat tinggal di sebuah kompleks perumahan. Para penghuni perumahan tersebut semakin bertambah akan tetapi banyak diantara penghuni tersebut saling tidak mengenal antara satu dengan yang lain. Selain itu para penghuni ketika mendapatkan masalah di suatu perumahan, sering kali juga para penghuni tersebut tidak tahu harus melapor kemana agar informasi yang mereka sampaikan dapat sampai dengan tepat dan diterima oleh pihak perumahan. Hal inilah yang menjadi tantangan seorang pengusaha *property* jika pengusaha tersebut memiliki keinginan untuk mewujudkan lingkungan perumahan yang harmonis dan aman. Sebuah aplikasi jejaring sosial berbasis *web* diharapkan mampu menjadi solusi permasalahan ini. Tujuannya adalah sebagai tempat penyaluran informasi antara pihak perumahan dengan penghuninya.

Untuk mengembangkan aplikasi ini digunakan beberapa teknologi penunjang yaitu *framework* laravel yang digunakan untuk membuat *web* dan *sms gateway api* yang digunakan untuk mengirimkan pemberitahuan melalui *sms*.

Metode pengembangan aplikasi diawali dengan mempelajari teknologi yang akan digunakan, mengumpulkan data dan menganalisa kebutuhan secara mendetail mengenai fitur utama, melakukan perancangan sistem dan melakukan pembuatan aplikasi sesuai dengan rancangan dan yang terakhir adalah melakukan pengujian fitur secara bertahap kemudian melakukan perbaikan jika terjadi *error*.

Aplikasi yang diuji coba berhasil berjalan dengan baik dan dapat disarankan untuk digunakan pada suatu lingkungan perumahan dalam bertukar informasi antar warganya atau dengan pihak perumahan.

Kata Kunci : sistem, jejaring sosial, *web*, pengusaha *property*, penghuni perumahan, tetangga

1. Pendahuluan

1.1. Latar Belakang

Perkembangan bisnis properti khususnya perumahan berkembang pesat di kota-kota besar di Indonesia. Dengan semakin banyaknya jumlah perumahan, semakin banyak pula orang-orang yang tinggal di perumahan. Arti kata perumahan adalah kumpulan beberapa rumah yang berfungsi sebagai lingkungan tempat tinggal atau lingkungan hunian yang dilengkapi dengan prasarana dan sarana lingkungan (Sugono, 2008). Banyak orang tinggal di perumahan dikarenakan adanya suasana dan lingkungan yang tenang, menyenangkan, dan juga rasa aman atau juga sebagai investasi. Walaupun pada suatu perumahan terdapat fasilitas-fasilitas yang lengkap dari perumahan untuk penghuninya tetapi sering kali terdapat suatu *gap* antara penghuni dan pihak perumahan yang terkadang dapat menjadi sumber masalah misalnya dalam hal penyampaian informasi dari pihak perumahan kepada penghuninya. Selain antara penghuni dan pihak perumahan, terkadang sesama penghuni atau tetangganya

bahkan tidak saling mengenal satu dengan yang lain. Dengan masalah tersebut, maka dibuatlah aplikasi jejaring sosial yang digunakan sebagai jembatan antara pihak perumahan dan penghuninya ataupun antar sesama penghuninya dalam bertukar informasi. Penggunaan aplikasi ini dari pihak perumahan ditujukan agar pihak perumahan dapat memberikan informasi kepada penghuni tentang hal-hal yang terjadi di lingkungan perumahan, seperti pemadaman listrik dan air secara mendadak, adanya penutupan jalan, pemberian informasi pembukaan sarana dan prasarana baru, pembukaan tipe-tipe rumah baru, dan lain-lainnya. Pihak penghunipun juga dapat menyampaikan informasi kepada pihak perumahan seperti adanya orang mencurigakan berkeliaran di lingkungan perumahan, ataupun terjadi kejadian yang membutuhkan penanganan cepat lainnya seperti kebakaran, banjir, pencurian, adanya sampah yang menumpuk, atau ada pohon tumbang, dan sebagainya. Selain itu sesama penghuni juga dapat saling berinteraksi, saling memberikan informasi misalnya menanyakan apakah ada tetangga yang tahu tentang anjingnya yang hilang, menanyakan informasi tentang tukang yang bagus di daerah

tersebut atau saling memberikan informasi klinik terdekat.

Metode yang digunakan untuk melakukan pembuatan perangkat lunak menggunakan metode *waterfall model*. Metode ini digunakan karena melakukan pendekatan secara sistematis dan urut mulai dari level kebutuhan sistem lalu menuju ke tahap analisis, perancangan, implementasi, pengujian, dan *maintenance*, sehingga aplikasi yang dihasilkan akan baik karena tahapan berikutnya dilakukan jika tahapan sebelumnya sudah benar. (Basuki, 2006).

1.2. Rumusan Masalah

Bagaimana merancang dan membangun *admin management* pada aplikasi tetanggaBaik sebagai jejaring sosial untuk perumahan di Indonesia berbasis PHP ?

1.3. Ruang Lingkup

- Perangkat lunak digunakan untuk perumahan yang ada *developer*-nya
- Perangkat lunak yang dibuat dapat saling melihat dan memberikan informasi baik antara pemilik perumahan dan juga pihak perumahan (bagian admin)
- Implementasi penggunaan perangkat lunak ini dengan menggunakan *web browser*.
- Pengguna perangkat lunak ini ditujukan untuk departemen perumahan dan penghuni yang tinggal di perumahan tersebut.
- Informasi yang bersifat darurat (*Alert*) akan langsung dikirimkan ke admin *posting* dengan menggunakan *email*, setelah di *approve* maka pesan tersebut akan di sebarkan ke penghuni yang berada dalam 1 komplek yang sama dengan pembuat pesan darurat (*Alert*) *via sms* dengan menggunakan *sms gateway* dan *email* setelah mendapatkan *approval* dari pihak admin.

1.4. Tujuan

Tujuan dari penelitian ini adalah merancang dan membangun *admin management* pada aplikasi tetanggaBaik sebagai jejaring sosial untuk perumahan di Indonesia berbasis PHP.

1.5. Manfaat

Manfaat yang dapat ditimbulkan adalah adanya *social network* yang dipakai pada suatu perumahan yang dapat membantu proses penyampaian informasi secara cepat antar pihak perumahan dan penghuninya. Dengan adanya aplikasi ini dapat menciptakan hubungan baik dan mempercepat dan mempermudah proses penyampaian informasi antar pihak perumahan dan penghuninya.

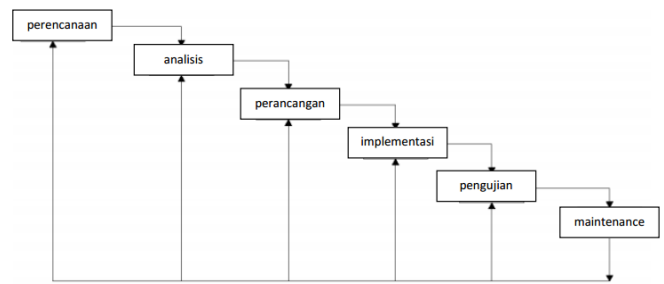
2. Landasan Teori

2.1. System Development Life Cycle dengan metode Waterfall

Dalam pembuatan sebuah sistem perlu dilakukan tahapan-tahapan pengembangan dalam prosesnya. Tahapan itu adalah perencanaan sistem, analisis sistem, desain dan perancangan sistem, implementasi sistem, pengujian sistem dan proses *maintenance*. Tahapan itu saling mempengaruhi dan membentuk sebuah siklus, yaitu jika pada suatu tahapan ternyata ada data-data yang seharusnya

dilakukan pada tahapan sebelumnya, maka dapat kembali ke tahapan sebelumnya. (Basuki, 2006)

Untuk melihat bentuk dari metode *waterfall* dapat dilihat pada Gambar 1.



Gambar 1. Tahapan dengan metode Waterfall

2.2. Jejaring Sosial

“Kita mendefinisikan *social network* sebagai *web-base service* yang memberikan kemampuan kepada seseorang untuk (1) membangun profil publik atau *semi-public* dalam sistem yang terikat, (2) artikulasi sekumpulan *user* dimana mereka memiliki suatu koneksi dan (3), menampilkan dan melintasi list koneksi mereka yang terjalin dalam sistem. Hubungan tersebut dapat berubah-ubah dari satu situs dan situs lainnya” (Boyd & Ellison, 2007)

Jejaring sosial secara sederhana dapat dijelaskan sebagai struktur relasi yang menghubungkan antara seseorang dalam suatu lingkungan atau perkumpulan. Jejaring sosial telah memanfaatkan sistem informasi yang telah berkembang sekarang ini agar setiap pengguna dapat menemukan teman baru tanpa dibatasi oleh jarak dan waktu.

2.3. Basis Data

Database adalah kumpulan *field-field* yang mempunyai kaitan antara satu *field* dengan *field* lain sehingga membentuk satu bangunan data untuk menyimpan informasi dalam batasan tertentu. (Pratama, 2013)

2.4. Pengembangan web

2.4.1. Web Server

Web server merupakan sebuah perangkat lunak dalam *server* yang berfungsi menerima permintaan (*request*) berupa halaman web melalui *HTTP* atau *HTTPS* dari klien yang dikenal dengan browser dan mengirimkan kembali (*response*) hasilnya dalam bentuk halaman-halaman *web* (Solichin, 2008).

2.4.2. PHP

“PHP digunakan sebagai *open source*, digunakan dengan tujuan bahasa *scripting*. Pada awal digunakan sebagai pembuatan *web site*. Kenyataannya, awal mula PHP digunakan sebagai alat pembuatan *personal home page* yang di buat oleh Rasmus Lerdorf untuk membantu pengguna dalam pembuatan *web page*-nya. PHP terbukti bermanfaat dan populer menjadikannya berkembang dengan cepat menjadi bahasa yang memiliki fitur yang lengkap, memperoleh nama PHP *Hypertext Preprocessor* untuk menunjukkan kemampuannya yang sudah diperluas dalam mengolah *web page* sebelum ditampilkan.” (Valade, 2004).

2.4.3. Framework Laravel

“Laravel adalah *framework* yang kuat, didesain untuk pembuatan aplikasi *web* dengan menggunakan *php*. Laravel akan memberikan jalan untuk memodulkan *code* kita dan membuatnya terlihat lebih bersih dan mudah untuk dibaca dan dipahami.” (Jain, 2012)

Laravel merupakan *framework* *php* dengan konsep MVC sehingga memudahkan untuk pembuatan program berbasis *php*.

2.4.4. Twitter Bootstrap

Twitter Bootstrap adalah sebuah alat bantu untuk membuat sebuah tampilan halaman *website* yang dapat mempercepat pekerjaan seorang pengembang *website* ataupun pendesain halaman *website*. (Ridha, 2013).

2.5. SMS Gateway API

SMS Gateway merupakan sebuah program aplikasi yang menghubungkan antara semua *SMS* yang diterima maupun dikirim ke sebuah *PC* dengan menggunakan jaringan GSM. (Utomo, 2006)

“*SMS Gateway API* adalah sebuah servis yang dapat digunakan untuk mengirim dan menerima *sms* secara terprogram dari *hp* android”. (<https://smsgateway.me/what-is-sms-gateway-api>).

Cara kerja dari *SMS Gateway API* adalah aplikasi mengirimkan *request* berupa nomor dan isi pesan *sms* kepada *server* milik *SMS Gateway API*, kemudian *server* mengirimkan *respond* ke *handphone* android yang telah diinstal aplikasi *SMS Gateway API*, kemudian aplikasi pada *handphone* akan mengirimkan *sms* ke nomor yang diberikan oleh aplikasi.

3. Desain Sistem

Membahas mengenai tahapan Analisis dan Perancangan dari Metode Pembuatan Perangkat Lunak *Waterfall Model*. Pada perancangan perangkat lunak ini menggunakan metode *waterfall model* karena fitur dari perangkat lunak sudah didefinisikan pada awal proses sehingga pengerjaannya akan menjadi lebih mudah, dan juga metode ini pengerjaannya dilakukan secara bertahap oleh karena itu jika perangkat lunak telah selesai dibuat maka dapat dipastikan bahwa perangkat lunak akan berjalan dengan baik sesuai dengan fitur awal aplikasi direncanakan.

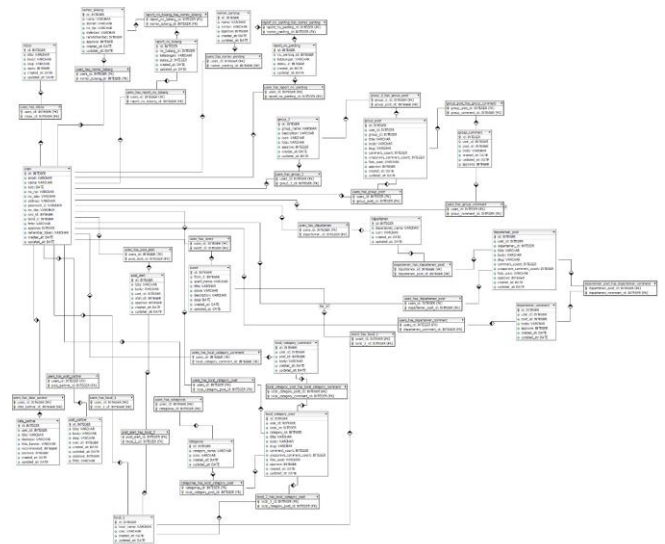
Pada tahapan analisis, dilakukan wawancara kepada 7 orang dari pihak perumahan di Citraland Surabaya. Urutan pertanyaan yang diajukan yaitu : (1) Cara pihak perumahan dan warga dalam saling menyampaikan informasi (2) Informasi-informasi apa saja yang disampaikan (3) Keluhan warga tentang penyampaian informasi (4) Pendapat tentang suatu produk baru yang dapat digunakan untuk bertukar informasi dengan cepat dan tepat yang dapat digunakan oleh perumahan kepada warganya atau sebaliknya. Untuk pengolahan datanya menggunakan teknik *Narative Analysis* yaitu untuk menganalisa pernyataan yang disampaikan oleh responden yang digunakan sebagai acuan penambahan fitur yang akan digunakan dalam aplikasi tetanggaBaik sebagai aplikasi jejaring sosial untuk perumahan.

Dari hasil penelitian, masing-masing responden memberikan jawaban yang hampir sama mengenai cara

memberikan informasi dari pihak perumahan kepada warganya, informasi yang disampaikanpun juga hampir sama. Tetapi yang berbeda adalah informasi yang diberikan oleh warga kepada pihak perumahan. Hal ini dapat disebabkan karena pihak perumahan walaupun yang memberikan informasi adalah orang yang berbeda tetapi informasi yang disampaikan adalah sama karena berkaitan dengan kepentingan perumahan, sedangkan untuk warga yang melaporkan sesuatu cenderung berbeda-beda karena sesuai dengan kepentingan masing-masing warga atau kejadian yang terjadi pada saat itu yang dialami oleh warga. Penyampaian informasi dari pihak perumahan kepada warganya disampaikan dengan menggunakan *SMS Blast* dan *posting* di akun media sosial perumahan, maka dapat dikatakan bahwa penggunaan *SMS Blast* merupakan fitur yang penting sehingga akan digunakan pada bagian *Menu Alert*, dimana informasi penting akan dengan segera di *sms*-kan dan di *email*-kan kepada penghuni di komplek tertentu. Penggunaan *SMS Blast* terdapat beberapa kekurangan yaitu jika nomor warga ganti dan belum melakukan pelaporan ganti nomor maka *SMS Blast* akan tetap mengirimkan pesan ke nomor yang lama sehingga pesan tidak sampai, oleh karena itu terdapat fitur untuk dapat melakukan perubahan data diri pada bagian *edit profile*, sehingga warga dapat mengganti nomor *hp* nya secara cepat tanpa harus melaporkannya melalui pihak marketing atau liaison officer

Pada bagian Perancangan dari metode *waterfall model* di bab ini terdiri dari rancangan *ER Diagram*, dan *Use Case Admin*.

3.1. Entity Relationship Diagram



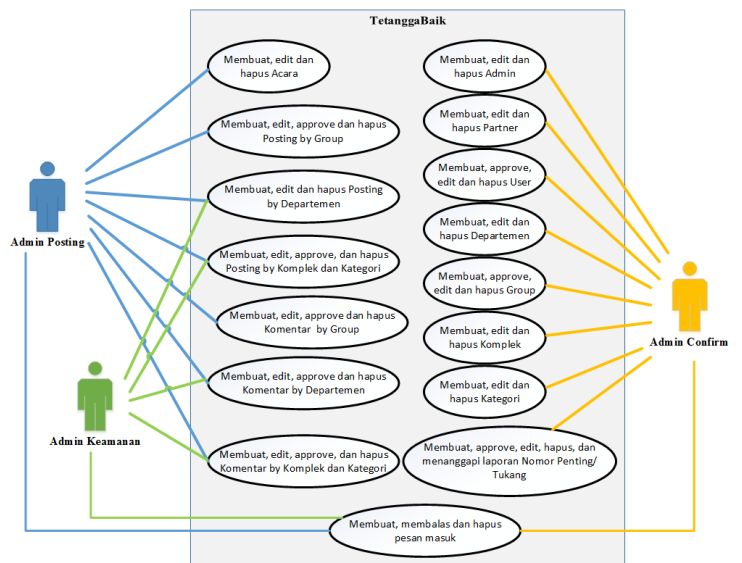
Gambar 2. ER Diagram tetangga Baik

Tabel 1. Penjelasan ER-Diagram

No	Nama Tabel	Keterangan	Hubungan
1	Users	Berisi data <i>email, password, name, dob, no_hp, no_telp, address, no_ktp, role_id, local</i> , foto, <i>approve</i>	Many to many dengan semua tabel
2	Inbox	Berisi <i>title, body, open, to_user_id</i>	Hubungan many to many dengan users
3	Nomor_Tukang	Berisi nama, alamat, no_hp, deskripsi, <i>approve</i>	Hubungan many to many dengan report_no_tukang
4	Report_no_tukang	Berisi no_tukang_id, keterangan, status	Hubungan many to many dengan nomor_tukang dan many to many dengan users
5	Nomor_Penting	Berisi nama, nomor, <i>approve</i>	Hubungan many to many dengan report_no_penting
6	Report_no_penting	Berisi no_penting_id, keterangan, status	Hubungan many to many dengan nomor_penting dan many to many dengan users
7	Data_partner	Berisi <i>title, deskripsi, foto_banner, approve</i>	Hubungan many to many dengan users
8	Post_Partner	Berisi <i>title, body, foto, approve</i>	Hubungan many to many dengan users
9	Local	Berisi <i>local_name, icon</i>	Hubungan many to many dengan user dan many to many dengan local_category_post
10	Local_category_post	Berisi <i>title, body, foto, approve, comment_count, unapprove_comment_count</i>	Hubungan many to many dengan local, many to many dengan user, dan many to many dengan local_category_comment
11	Local_category_comment	Berisi <i>body</i>	Hubungan many to many dengan users, dan many to many dengan local_category_post
12	Categories	Berisi <i>category_name, icon</i>	Hubungan many to many dengan user dan many to many dengan local_category_post
13	Group	Berisi <i>group_name, icon</i>	Hubungan many to many dengan user dan many to many dengan group_post
14	group_post	Berisi <i>title, body, foto, approve, comment_count, unapprove_comment_count</i>	Hubungan many to many dengan group, many to many dengan user, dan many to many dengan group_comment
15	group_comme	Berisi <i>body</i>	Hubungan many to

16	departemen	Berisi <i>departemen_name, icon</i>	many dengan users, dan many to many dengan group_post
17	departemen_post	Berisi <i>title, body, foto, approve, comment_count, unapprove_comment_count</i>	Hubungan many to many dengan user dan many to many dengan departemen_post
18	departemen_comment	Berisi <i>body</i>	Hubungan many to many dengan users, dan many to many dengan departemen_post
19	Post_alert	Berisi <i>title, body, approve</i>	Hubungan many to many dengan users dan many to many dengan local
20	Event	Berisi <i>event_name, title, place, description, date</i>	Hubungan many to many dengan users dan many to many dengan local

3.2. Use Case Diagram



Gambar 3 Use Case Admin TetanggaBaik

Tabel 2. Penjelasan Umum Pembagian Admin

Admin	Keterangan
Admin Posting	Bertugas untuk melakukan <i>posting</i> -an di Acara, Komplek, <i>Group</i> , Kategori, Alert dan menangani <i>approve posting</i>
Admin Keamanan	Bertugas untuk melakukan <i>posting</i> -an di Keamanan dan menerima laporan warga mengenai keamanan
Admin Confirm	Bertugas untuk menambahkan atau menghapus mengenai data <i>user, partner, kategori, komplek, group, nomor penting, nomor tukang, konten</i> pada halaman awal dan <i>approve</i> atau <i>decline</i> permintaan <i>user</i>

4. Implementasi dan Pengujian

Membahas mengenai tahapan Implementasi dan Pengujian dari Metode Pembuatan Perangkat Lunak *Waterfall Model*.

Pada tahapan implementasi terdiri dari lingkungan pengembangan perangkat lunak yaitu spesifikasi dari laptop yang digunakan untuk membuat, spesifikasi domain dan hosting yang digunakan untuk meng-*upload* dan sebagai tempat untuk mengakses/menggunakan aplikasi, spesifikasi dari modem yang digunakan pada saat uji coba aplikasi, implementasi penggunaan *framework* laravel, implementasi penggunaan *SMS Gateway API* sebagai *sms gateway* pada aplikasi.

Pada tahapan pengujian, aplikasi diujikan kepada 10 orang untuk mengetahui dan menguji apakah aplikasi dapat berjalan dengan lancar dan seharusnya tanpa adanya *error* atau *function* yang tidak bekerja dengan baik.

4.1. Spesifikasi dalam Pembuatan Aplikasi

Spesifikasi Laptop yang digunakan dalam pembuatan perangkat lunak :

- ASUS tipe N 55 SF
- Processor Intel® Core™ i7-2630QM CPU @ 2.00GHz
- RAM 8 GB
- Sistem Operasi Microsoft Windows 8 pro 64-bit
- XAMPP 5.6.3, Apache 2.4.10
- MySQL v 5.6.21
- PHP 5.6.3
- *Framework* Laravel v 4
- *Browser* Chrome v 42.0.2311.135 m

Spesifikasi *Domain* dan *Hosting* yang digunakan untuk menaruh dan akses perangkat lunak :

- *Domain* : tetanggabaik.com
- *Framework* Laravel v 4
- Bandwith 16 GB
- PHP 5.4.38
- MySQL v 5.5.42-cll
- Apache v 2.2.29
- cPanel v 11.48.3
- Sistem Operasi Linux

Spesifikasi *handphone* yang digunakan sebagai modem untuk *SMS Gateway API* :

- Samsung Galaxy Note 3 SM-N900
- Provider GSM INDOSAT
- Biaya per sms Rp 225

Sms Gateway menggunakan *SMS Gateway API service*

4.2. Implementasi Framework Laravel

Dalam pembuatan aplikasi tetanggaBaik menggunakan Laravel v.4 sebagai *framework*. Laravel bekerja dengan PHP 5.3+ sehingga dapat memanfaatkan kelebihan-kelebihan dari PHP versi terbaru. Dokumentasi laravel dapat dilihat di (<http://laravel.com/docs/4.2>).

4.2.1. Input-Output Database

Laravel 4 menggunakan *Eloquent ORM* untuk menghubungkan aplikasi dengan *database*, dan juga *Eloquent ORM* ini memungkinkan *server* untuk menggunakan berbagai macam *database* tanpa perlu

mengganti *query* yang kita gunakan. *Eloquent ORM* membuat *script query* menjadi lebih mudah untuk dipahami dan digunakan.

Contoh proses pengambilan data *user* dari *database*

```
$user = User::find($id)->delete();
```

Gambar 4. mengambil data *User* dari *database*

Pada Gambar 4 cara menggunakannya sebagai berikut :
\$(namaVariabel) = (NamaModel) :: (perintah query);

\$(namaVariabel) merupakan variabel yang digunakan untuk menampung hasil *query Eloquent ORM*.

(NamaModel) merupakan model yang digunakan untuk menghubungkan dengan tabel di *database*

(perintah query) merupakan bentuk *query* yang disederhanakan oleh *Eloquent ORM* yang merupakan perintah untuk melakukan sesuatu di *database*, pada contoh di atas pada baris pertama adalah perintah untuk mengambil semua *row* data di tabel *User*. Pada baris kedua adalah perintah untuk mencari data di tabel *User* dengan id = 1. Pada baris ketiga merupakan perintah untuk mengambil semua *row* data dari tabel *User* dengan *role_id* = 3.

Variasi perintah lainnya dapat dilihat pada dokumentasi Laravel *Eloquent ORM* (<http://laravel.com/docs/4.2/eloquent>)

Contoh proses *input, delete* dan *edit* data *User* kedalam *database*

```
$user = new User;  
$user->name = 'John';  
$user->save();
```

Gambar 5. input data *User* ke *database*

Pada Gambar 5 cara menggunakannya sebagai berikut:

\$(nama variabel) = new (Nama model);

Membuat sebuah variabel *class model* baru yang akan digunakan untuk menampung *value* yang akan dimasukkan kedalam *database*.

\$(nama variabel)->(nama tabel) = (value);

Menampung *value* sesuai dengan yang akan dimasukkan kedalam kolom di *database*, pada contoh diatas *value* 'John' dimasukkan dalam kolom *name*.

\$(nama variabel)->save();

Merupakan perintah *artisan* dari laravel untuk melakukan penyimpanan data ke dalam *database*.

```
$user = User::find($id)-  
>update(array('status' => 2));
```

Gambar 6. edit data *User* dari *database*

Pada Gambar 6 cara menggunakannya sebagai berikut:

\$(nama variabel) = (Nama model)::(Perintah query)->update((nama kolom => value));

Pada contoh gambar menggunakan perintah *find()* untuk mencari *user* sesuai dengan *id* yang di *inputkan* lalu

mengganti *value* pada kolom *status* dengan *value* baru yaitu 2 untuk *approved*.

```
$users = User::all();
return view::make('user.profile',
array('user' => $users));
```

Gambar 7. delete data *User* dari *database*

Pada Gambar 7 cara menggunakannya sebagai berikut:

```
$(nama variabel) = (Nama model)::(query)->delete();
```

Pada contoh gambar menggunakan perintah *find()* untuk mencari *user* sesuai dengan *id* yang diinputkan lalu menghapus *row* dari *user* tersebut.

Contoh proses mengirimkan data *User* ke *view*

```
$users = User::all();
$user = User::find(1);
$users = User::where('role_id', '=', 3)-
>get();
```

Gambar 8. mengirimkan data *User* ke *view*

Pada Gambar 8 cara menggunakannya sebagai berikut:

```
Return View::make((nama folder . nama file), (nama
variabel di view)=>(value));
```

`return view::make` merupakan perintah artisan laravel untuk membuat tampilan *view* di halaman *web*.

(nama folder . nama file) merupakan letak dan nama *file* di dalam *folder View*.

`array('user' => $users)` merupakan *value* yang akan dilempar ke *view*.

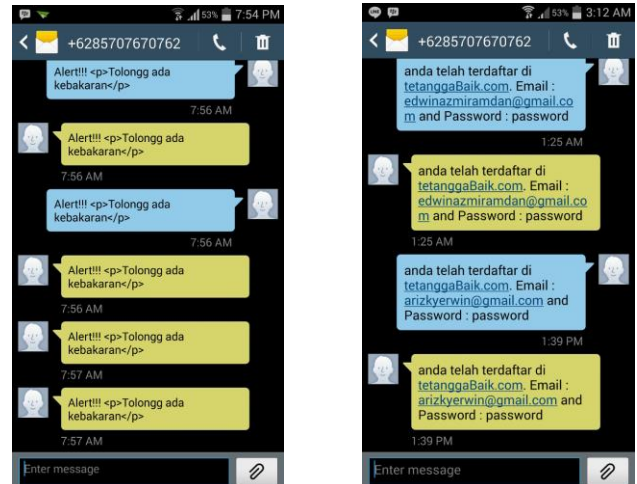
4.3. Implementasi SMS Gateway API

SMS Gateway API merupakan sebuah layanan yang dapat digunakan untuk merubah *handphone* android sebagai modem untuk melakukan pengiriman sms melalui *request* yang dibuat oleh aplikasi. Menggunakan *SMS Gateway API* karena *sms gateway* ini bisa dikatakan lebih fleksibel karena modem yang digunakan tidak harus dihubungkan ke *server* secara langsung tidak seperti *sms gateway* pada umumnya, sehingga jika menggunakan jasa *hosting* maka penggunaan *SMS Gateway API* ini lebih cocok.

✓ +6285707670762	anda telah terdaftar di tetanggaBaik.com. Email : edwinazmiramdan@gmail.com and Password : password	02/05/15 18:25:08	Msg #3808426 received on Device 7883
✓ 085707670762	anda telah terdaftar di tetanggaBaik.com. Email : edwinazmiramdan@gmail.com and Password : password	02/05/15 18:25:01	Msg #3808425 sent on Device 7883
✓ Google	Your Google verification code is 475980	02/05/15 18:03:16	Msg #3807116 received on Device 7883
✓ 0878512521	Alert!!! AWAS , POHON TUMBANG di DEPAN PINTU MASUK KOMPLEKS PURI WIDYA KENCANA	02/05/15 10:39:00	Msg #3801999 sent on Device 7883
✓ 08132125251	Alert!!! AWAS , POHON TUMBANG di DEPAN PINTU MASUK KOMPLEKS PURI WIDYA KENCANA	02/05/15 10:38:56	Msg #3801997 sent on Device 7883
✓ 0892525252	Alert!!! AWAS , POHON TUMBANG di DEPAN PINTU MASUK KOMPLEKS PURI WIDYA KENCANA	02/05/15 10:38:49	Msg #3801995 sent on Device 7883

Gambar 9. halaman *list* sms yang terkirim/pending/fail

Pada Gambar 9 menunjukkan *list* dari sms yang pernah dikirim / *pending* / gagal dengan menggunakan *SMS Gateway API*. Pada halaman *list* jika ada *status* sms yang *pending*, maka bisa dikirim secara manual dengan menekan pada tulisan *pending*-nya.



Gambar 10. halaman sms yang diterima di hp (kiri) sms alert, (kanan) sms daftar user baru

Pada Gambar 10 merupakan hasil dari *sms* yang dikirim dan diterima oleh *user* pada saat penggunaan *sms gateway*. Untuk pendaftaran *user* baru akan di kirimkan data *login* berupa *email* dan *password* yang dapat digunakan untuk *login* kedalam aplikasi. Sedangkan untuk *post Alert* akan memiliki tanda “Alert!!!” pada saat *sms* diterima

4.3.1. Penggunaan SMS Gateway API

```
// dapat di edit //
$nohp = $no_hp_penerima;
$baseUrl = "https://msgateway.me";
$url = "/api/v3/messages/send";

$emailmsggateway = 'riconova92@gmail.com!';
$passwordmsggateway = 'riconovasuprayoto!';
$deviceID = 7883;

$fields['number'] = $nohp;
$fields['message'] = $isiMessage;
$fields['device'] = $deviceID;
$fields['email'] = $emailmsggateway;
$fields['password'] = $passwordmsggateway;
// =====//
// perintah SMS Gateway API untuk mengirimkan
request ke Server SMS Gateway API
$url = $baseUrl.$url;
$fieldsString = http_build_query($fields);
$ch = curl_init();
curl_setopt($ch,CURLOPT_POST,
count($fields));
curl_setopt($ch,CURLOPT_POSTFIELDS,
$fieldsString);
curl_setopt($ch, CURLOPT_URL,$url);
curl_setopt($ch, CURLOPT_RETURNTRANSFER,1);
curl_setopt($ch, CURLOPT_HEADER, false);
curl_setopt($ch, CURLOPT_SSL_VERIFYPEER,
false);
$result = curl_exec ($ch);
$return['response'] =
json_decode($result,true);
if($return['response'] == false)
$return['response'] = $result;
$return['status'] = curl_getinfo($ch,
CURLINFO_HTTP_CODE);
curl_close ($ch);
```

Gambar 11. penggunaan sms gateway api untuk mengirimkan request

Pada Gambar 11 pada bagian *// dapat di edit //* merupakan bagian yang bisa di atur oleh untuk

menginputkan no *hp* penerima, *account* dari *sms gateway api*, *id device* (modem) yang digunakan untuk mengirimkan sms, dan isi dari *sms* yang dikirim.

Tabel 3. pembagian bagian *sms gateway api* yang dapat *diedit*

Code	Keterangan
<code>\$nohp = \$no_hp_penerima;</code> <code>\$fields['number'] =</code> <code>\$nohp;</code> <code>\$baseUrl =</code> <code>"https://smgateway.me";</code> <code>\$url =</code> <code>"/api/v3/messages/send";</code>	Uuntuk meng- <i>inputkan</i> no hp penerima sms <i>Url</i> untuk mengirimkan <i>request</i> dan <i>account</i> untuk <i>sms gateway api</i>
<code>\$emailsmgateway =</code> <code>'riconova92@gmail.com';</code> <code>\$passwordsmgateway =</code> <code>'riconovasuprayoto';</code> <code>\$fields['email'] =</code> <code>\$emailsmgateway;</code> <code>\$fields['password'] =</code> <code>\$passwordsmgateway;</code>	
<code>\$deviceID = 7883;</code> <code>\$fields['device'] =</code> <code>\$deviceID;</code> <code>\$fields['message'] =</code> <code>\$isiMessage;</code>	<i>Device</i> yang digunakan sebagai modem Isi dari <i>sms</i>

Tabel 4. perbandingan harga per *sms* beberapa operator di Indonesia

No	Nama Operator	Tarif Normal	Paket SMS
1	Indosat	Rp 225/sms	Paket Gratis 30 (gratis 30 sms tiap hari)
2	3	Rp 150/sms	Paket SMS++ (Rp 30.000 dapat 1000 sms)
3	Telkomsel	Rp 200/sms	Tidak ada
4	XL	Rp 165/sms	Kirim 10 sms gratis 100 sms tiap hari

Sumber : (<http://indosat.com/id/personal/prabayar/im3/im3-gratis-30>), (<http://tri.co.id/pulsa/harga>, <http://www.telkomsel.com/tarif-simpat>, <http://www.xl.co.id/id/prabayar/tarif?packedargs=param%3D1363668216810>)

Sangat disarankan untuk memilih jaringan operator kartu *simcard* GSM yang memiliki harga per *sms* nya murah atau memiliki paket *bundling* untuk *sms*. Jika dilihat pada tabel perbandingan di atas maka yang merupakan pilihan terbaik adalah menggunakan kartu *simcard* 3 dengan biaya Rp 150/sms dan memiliki paket *bundling sms* yaitu paket SMS++ Rp 30.000 untuk 1000 *sms*, jika dihitung harga normal untuk 1000 *sms* adalah $150 \times 1000 = \text{Rp } 150.000$.

4.4. Pengujian

Pada tahapan ini merupakan tahapan kelima dari *waterfall model*, pengujian dilakukan kepada 10 orang dengan tujuan dilakukan pengujian aplikasi ini adalah untuk mengetahui apakah aplikasi yang dibuat telah berjalan dengan.

4.4.1. Skenario Pengujian Admin Confirm Skenario Login

Pada skenario ini penguji membuka halaman tetanggabaik.com dan melakukan *login* kedalam halaman admin confirm dengan *email* dan *password* yang telah diberikan

Skenario User / Admin / Partner

Penguji diminta untuk membuat baru dengan memilih menu dan menginputkan data baru, sistem akan mengirimkan notifikasi sesuai dengan pilihan penguji (*sms* atau *email*), dan *user/admin/partner* baru menerima notifikasi pemberitahuan data *login*.

Kemudian penguji diminta untuk merubah/*edit* data utama dengan memilih/mencari menggunakan *search bar user/admin/partner* yang akan di *edit* lalu menekan tombol *edit*.

Kemudian penguji diminta untuk menghapus/*delete* data dengan memilih/mencari menggunakan *search bar user/admin/partner* yang akan di hapus lalu menekan tombol hapus.

Jika memilih *user*, penguji juga diminta *approve* permintaan *user* baru dan berhasil *login* dengan data *user* baru

Skenario Departemen / Kategori / Komplek / Helps & FAQ / Terms & Condition

Penguji diminta untuk membuat data baru dengan memilih menu dan menginputkan data baru. Lalu data baru akan ditampilkan dalam *list*.

Kemudian penguji diminta untuk merubah/*edit* data dengan memilih departemen/kategori/komplek/*helps & faq/terms & condition* yang akan di *edit* lalu menekan tombol *edit*.

Kemudian penguji diminta untuk menghapus/*delete* data dengan memilih departemen/kategori/komplek/*helps & faq/terms & condition* yang akan di hapus lalu menekan tombol hapus.

Skenario Group / Nomor Penting / Nomor Tukang

Penguji diminta untuk membuat data baru dengan memilih menu dan menginputkan data baru. Lalu data baru akan ditampilkan dalam *list*.

Kemudian penguji diminta untuk merubah/*edit* data dengan memilih *group / nomor penting / nomor tukang* yang akan di edit lalu menekan tombol *edit*.

Kemudian penguji diminta untuk menghapus/*delete* data dengan memilih *group / nomor penting / nomor tukang* yang akan di hapus lalu menekan tombol hapus.

Kemudian penguji diminta untuk *approve* data yang telah dibuat sebelumnya dengan memilih *group / nomor penting / nomor tukang* yang akan di *approve* lalu menekan tombol *approve*.

Kemudian penguji diminta untuk menanggapi laporan yang telah dibuat sebelumnya dari user dengan memilih laporan user nomor penting / laporan user nomor tukang dengan tanggapan terserah penguji untuk *edit* nomor atau hapus laporan.

Skenario Pesan

Penguji diminta membuat pesan baru kepada admin "admin posting" kemudian penguji diminta membalas pesan dari "admin posting" dan akhirnya penguji melihat "pesan keluar" apakah pesan telah terkirim atau belum. Berhasil menghapus 1 pesan masuk/pesan keluar.

Skenario Profil

Penguji diminta membuka halaman *profile* dan mengganti data *user* (boleh data akun, ataupun *password*)

Hasil Pengujian Admin Confirm**Tabel 5. Hasil Pengujian Admin Confirm**

No	Skenario	Hasil yang di harapkan	Hail yang terjadi	Berhasil	Gagal
1	<i>Login</i>	Penguji berhasil <i>login</i> dan masuk ke halaman Admin <i>Confirm</i>	Hasil yang didapatkan sesuai dengan yang di harapkan	5	0
2	<i>User</i> / <i>Admin</i> / <i>Partner</i>	Penguji berhasil membuat <i>user/admin/partner</i> baru kemudian mendapatkan <i>sms/email</i> notifikasi sesuai dengan pilihannya Dan <i>user/admin/partner</i> baru dapat <i>login</i> Berhasil merubah data utama <i>user/admin/partner</i> Berhasil menghapus <i>user/admin/partner</i> Berhasil <i>approve</i> permintaan <i>user</i> baru	Hasil yang didapatkan sesuai dengan yang di harapkan Kecuali pada saat pertama kalo mencoba terjadi <i>server down</i> sehingga pengiriman <i>sms gateway</i> dan <i>email</i> ke <i>user</i> baru tidak dapat dilakukan	4	1
3	Departemen / / Kategori / Komplek / Helps & FAQ / Terms & Condition	Penguji berhasil membuat departemen / kategori / komplek / <i>helps & faq / terms & condition</i> baru dan berhasil muncul dalam <i>list</i> Berhasil merubah departemen / kategori / komplek / <i>helps & faq / terms & condition</i> Berhasil menghapus departemen / kategori / komplek / <i>helps & faq / terms & condition</i>	Hasil yang didapatkan sesuai dengan yang di harapkan	5	0
4	Group / Nomor / Penting / Nomor / Tukang	Penguji berhasil membuat <i>group / nomor penting / nomor tukang</i> baru dan berhasil muncul dalam <i>list</i> Berhasil merubah <i>group / nomor penting / nomor tukang</i> Berhasil menghapus <i>group / nomor penting / nomor tukang</i> Berhasil <i>approve request-an group / nomor penting / nomor tukang</i> dari user dan dimunculkan dalam <i>list</i> di user Berhasil menanggapi laporan nomor penting / nomor tukang dari <i>user</i>	Hasil yang didapatkan sesuai dengan yang di harapkan	5	0
5	Pesan	Penguji berhasil membuat pesan baru kepada admin "admin posting" Berhasil melihat dan membalas pesan masuk dari "admin posting" Berhasil melihat pesan keluar apakah pesan telah terkirim atau belum Berhasil menghapus 1 pesan masuk/pesan keluar	Hasil yang didapatkan sesuai dengan yang di harapkan	5	0
6	Profil	Penguji berhasil membuka halaman <i>pofile</i> dan mengganti data diri / <i>password</i>	Hasil yang didapatkan sesuai dengan yang di harapkan	5	0

4.4.2. Skenario Pengujian Admin Posting dan Admin Keamanan**Skenario Login**

Pada skenario ini penguji membuka halaman tetanggabaik.com dan melakukan *login* kedalam halaman admin *posting / admin keamanan* dengan *email* dan *password* yang telah diberikan.

Skenario Pesan

Penguji diminta membuat pesan baru kepada admin "admin *confirm*" kemudian penguji diminta membalas pesan dari "admin *confirm*" dan akhirnya penguji melihat "pesan keluar" apakah pesan telah terkirim atau belum. Berhasil menghapus 1 pesan masuk/pesan keluar.

Skenario Profil

Peng uji diminta membuka halaman *profile* dan mengganti data user (boleh data akun, ataupun *password*).

Skenario Post Komplek / Departemen / Group / Alert

Peng uji diminta untuk membuat *posting*-an baru (terserah peng uji jenis *post*-nya), kemudian membuka *post*-nya kembali pada menu sesuai dengan jenis *post* yang dibuat dan merubah *post* yang baru dibuat, kemudian melakukan *approve post* yang dibuat oleh *user*, kemudian diminta untuk menghapus *post* yang dibuatnya tadi.

Skenario Acara

Peng uji diminta untuk membuat acara baru dengan syarat minimal mengajak 2 kompleks, kemudian peng uji diminta untuk melihat acara "testing aplikasi tetangga baik" yang

dibuat oleh "ryan surya", dan memberikan kepastian akan mengikuti acara atau tidak, setelah itu peng uji diminta untuk melihat list peserta pada acara "testing aplikasi tetangga baik".

Skenario Komentar

Peng uji diminta untuk membuka *posting*-an dengan kategori : "kehilangan" atau komplek : "puri widya kencana", judul : Kehilangan Sebuah berlian 100 Karat dengan merek "Bralianz" dan kemudian memberikan komentar pada *posting* itu. Lalu diminta untuk *approve* komentar dari *user* lain. Lalu diminta untuk menghapus komentar dari *user* lain.

Hasil Pengujian Admin Posting

Tabel 6. Hasil Pengujian Admin Posting

No	Skenario	Hail yang Diharapkan	Hasil yang Terjadi	Berhasil	Gagal
1	Login	Peng uji berhasil <i>login</i> dan masuk ke halaman Admin <i>Posting</i> / Admin Keamanan	Hasil yang didapatkan sesuai dengan yang di harapkan	5	0
2	Post Komplek / Departemen / Group / Alert	Peng uji berhasil membuat post baru di Departemen / Komplek / Group Peng uji berhasil membuat <i>post Alert</i> yang di sebarakan ke <i>user</i> melalui <i>sms</i> dan <i>email</i> dengan komplek yang sama pada waktu memilih komplek pada waktu membuat <i>post Alert</i> Peng uji berhasil menemukan dan melihat <i>post</i> nya Peng uji berhasil meng- <i>edit post</i> nya Peng uji berhasil <i>approve post</i> yang dibuat oleh <i>user</i> lain Peng uji berhasil menghapus <i>post</i> nya	Hasil yang didapatkan sesuai dengan yang di harapkan *2 kali <i>fail</i> pada waktu melakukan pengujian <i>post Alert</i> dan <i>servernya down</i> sehingga <i>sms</i> dan <i>email</i> tidak ada yang terkirim	3	2
3	Komentar	Peng uji berhasil memberikan komentar yang dapat langsung muncul di <i>user</i> tanpa harus di <i>approve</i> Peng uji berhasil <i>approve</i> komentar dari <i>user</i> lain Peng uji berhasil menghapus komentar dari <i>user</i> lain	Hasil yang didapatkan sesuai dengan yang di harapkan	5	0
4	Acara	Peng uji berhasil membuat acara baru Peng uji berhasil meng- <i>edit</i> acaranya yang baru dibuat Peng uji berhasil melihat dan memberikan kepastian mengikuti atau tidak Peng uji berhasil melihat list peserta acara	Hasil yang didapatkan sesuai dengan yang di harapkan	5	0
5	Pesan	Peng uji berhasil membuat pesan baru kepada admin "admin <i>confirm</i> " Berhasil melihat dan membalas pesan masuk dari "admin <i>confirm</i> " Berhasil melihat pesan keluar apakah pesan telah terkirim atau belum Berhasil menghapus 1 pesan masuk/pesan keluar	Hasil yang didapatkan sesuai dengan yang di harapkan	5	0
6	Profil	Peng uji berhasil membuka halaman <i>pofile</i> dan mengganti data diri / <i>password</i>	Hasil yang didapatkan sesuai dengan yang di harapkan	5	0

4.5. Pembahasan

Code dari *SMS Gateway API* digunakan oleh aplikasi untuk mengirimkan *request* pengiriman *sms* kepada nomor tujuan yang diambil dari *database* aplikasi ke *server SMS Gateway API* pada saat melakukan pengujian *server hosting* yang digunakan mendadak *down*, sehingga aplikasi tidak dapat mengirimkan *request* mengirimkan *sms* sehingga terjadi kegagalan total tiga kali yaitu dua kali pada waktu melakukan *posting Alert* (*posting* yang digunakan untuk menyebarkan berita penting melalui *sms* dan email kepada warga yang berada pada komplek yang sama dengan pembuat *posting Alert*) dan satu kali Daftar User Baru (karena notifikasi email dan *sms* tidak terkirim). Untuk uji coba fitur yang lain berhasil sesuai dengan hasil yang diharapkan.

5. Kesimpulan

Aplikasi yang diuji coba berhasil berjalan dengan baik dan dapat disarankan untuk digunakan pada suatu lingkungan perumahan dalam bertukar informasi antar warganya atau dengan pihak perumahan.

DAFTAR PUSTAKA

- Adi, W., & Biannto, I. (2010). Layanan Informasi Untuk Pasien Rumah Sakit Menggunakan SMS. *Jurnal Penelitian*, 13(2), 199-216. Retrieved May 18, 2014, from https://www.usd.ac.id/lembaga/lppm/jurnal.php?id=abstraksi&model=volume&id_j=14&id_m=58&id_k=271
- Basuki, A. (2006). Pengembangan Perangkat Lunak. Surabaya, Indonesia: PENS-ITS.
- Boyd, D. M., & Ellison, N. B. (2007). *Social Network Sites: Definition, History, and Scholarship*. *Journal of Computer-Mediated Communication*, 13(1), 210–230. Retrieved May 18, 2014, from <http://onlinelibrary.wiley.com/doi/10.1111/j.1083-6101.2007.00393.x/full>
- Jain, A. (2012). *An Introduction about Laravel Framework*. Retrieved May 18, 2014, from <http://mrbool.com/an-introduction-about-laravel-framework/26410>
- Laravel. (n.d.). *Documentation*. Retrieved from <http://laravel.com/docs/4.0>. Access date 1 May 2015
- Panji (2013). arti kata perumahan. Retrieved May 18, 2014, from <http://glosarium.org/arti/?k=perumahan>
- Pratama, A. (2013). Pengenalan MySQL. Retrieved May 18, 2014, from <http://ilmukomputer.org/2013/02/01/pengenalan-mysql/>
- Ridha, H. (2013). Implementasi Twitter Bootstrap pada Code Igniter. Retrieved May 18, 2014, from <http://ilmukomputer.org/2013/05/10/implementasi-twitter-bootstrap-pada-codeigniter/>
- Rozidi, R. I. (2004). Membuat Sendiri *SMS Gateway* (ESME) Berbasis Protokol SMPP. Yogyakarta, Indonesia: Andi Offset.
- SMS Gateway API. (n.d.). *Your Android phone, Your SMS Gateway*. Retrieved from <https://msgateway.me>. Access date 1 May 2015
- Solichin, A. (2008). Pemrograman *web* dengan PHP dan MySQL. Jakarta, Indonesia.
- Sugono, D. (2008). Perumahan. In Kamus Bahasa Indonesia (4th ed., p. 1228). Jakarta, Indonesia: Pusat Bahasa.
- Utomo, P. A. (2006). Membangun Aplikasi *SMS* Berbasis *Open Source*. Yogyakarta, Indonesia: Andi Offset.
- Valade, J. (2004). *PHP 5 for Dummies*. Indianapolis, Indiana: Willey Publising, Inc.