

KAJIAN PENGEMBANGAN SISTEM INFORMASI BERBASIS *OPEN SOURCE* JOOMLA DENGAN MENGGUNAKAN *AUTO GENERATE*

Muhammad Rifqi

Magister Ilmu Komputer Program Pascasarjana Universitas Budi Luhur
rifqi.muhammad@id.panasonic.com

ABSTRAK

Perkembangan pemrograman berorientasi obyek dewasa ini begitu pesat. Open source Joomla adalah satu contoh pemrograman berorientasi. Peneliti melakukan pengkajian Open Source Joomla pada tahap design dimana pada tahap design ini peneliti menemukan pada proses building yang mana antara modul dan component harus disesuaikan dengan letak block yang dibuat dalam suatu aplikasi namun secara parsial pada modul Front-end, Back-End, Add/Edit, Delete dan Preview. Dalam penelitian ini peneliti menggunakan metodologi useability sebagai konsep dasar model perencanaan strategis. Hasil dari penelitian ini menunjukkan bahwa auto generate membantu memetakan penempatan modul dan component yang telah disesuaikan dengan letak block serta menyederhanakan proses dan pengembangan pada aplikasi open source Joomla dalam satu kesatuan.

Kata Kunci : *Open source berbasis obyek, Joomla, object oriented, auto generate*

1. Latar Belakang

Open source merupakan alternatif yang signifikan [1], [11] untuk pengembangan produk komoditas perangkat lunak atau aplikasi layanan [2]. *Open source* jauh lebih cepat, lebih efektif, dan lebih rendah biayanya dibandingkan dengan *software engineering* lainnya, ini menunjukkan bahwa pendekatan baru untuk rekayasa, pengembangan produk dan inovasi lebih memungkinkan [2], [3], [12].

Ukuran atau indikator kualitas sebuah *system* meliputi kehandalan dan keamanan (13), waktu respon untuk mendiagnosa dan memperbaikinya [14], [15], serta peningkatan komunitas pengguna yang terkait dengan kemudahan dan teknik komunikasi yang efektif [4], membangun kepercayaan *public* [11], bertambahnya akumulasi sosial oleh pengembang dan kontributor [5].

Open source software memiliki fitur yang lebih stabil dan lebih tinggi kualitasnya

dibandingkan dengan sistem yang dikembangkan dengan teknik konvensional dalam pengaturan [6], [14], [15].

Open source software adalah alat dan sistem aplikasi yang murah/gratis untuk diperoleh, relatif mudah untuk dipelajari dan dapat di akses secara global dan dapat di transfer ke internet [1], [11] banyak *tools* yang diberikan oleh pengembang dengan cuma-cuma [5].

Joomla adalah contoh dari perangkat lunak *open source* [7]. Sebuah proyek *open source* dikembangkan oleh komunitas pengembang di seluruh dunia, semua sukarela meluangkan waktu mereka untuk itu.

Joomla adalah *powerful open source* [8] merupakan bagian dari CMS (*content management system*) yang tersedia di internet untuk di download gratis dan digunakan. Joomla digunakan di seluruh dunia dari *website* yang sederhana sampai yang kompleks, interaktif dan untuk aplikasi

perusahaan [7]. Bahkan bukan *programmer* pun dapat membuat *website* interaktif yang sangat baik dengan banyak fungsi dengan menggunakan Joomla [8]. Perubahan atau penambahan fungsi baru ke sebuah situs dapat dibuat dengan mudah, kemahiran dalam penggunaan setiap CMS membawa nilai nyata untuk sebuah perusahaan *web design*.

Joomla memiliki dua *site* yaitu situs umum (biasa disebut dengan *front end*) biasanya orang melihatnya pada alamat *website* sebagai contoh www.yoursite.com dan situs administrasi atau admin (biasa disebut dengan *back end*) yaitu URL www.yoursite.com/administrator [7].

2. Rumusan Permasalahan

Rumusan permasalahan dalam penelitian ini adalah bagaimana menyederhanakan proses dan pengembangan pada aplikasi *open source* Joomla menjadi satu kesatuan sehingga memudahkan dalam pembuatan suatu aplikasi pada Joomla [9].

3. Tujuan Penelitian

Tujuan penelitian ini adalah membantu memetakan penempatan modul dan *component* yang telah di sesuaikan dengan letak *block* serta menyederhanakan proses dan pengembangan pada aplikasi *open source* Joomla dengan memberikan alternatif dalam tahapan pengembangan sebuah sistem informasi agar pengembang dapat dengan mudah membuat sistem aplikasi.

4. Manfaat Penelitian

Manfaat yang diperoleh dari penelitian ini adalah sebagai berikut:

- 1) Dapat digunakan sebagai alternatif dalam mempermudah dan mempersingkat pembuatan sebuah sistem informasi.
- 2) Dengan penelitian ini dapat memberikan manfaat akademis yaitu sebagai tambahan referensi.

Manfaat praktis yang dapat diberikan dari penelitian ini berupa sumbangan pemikiran untuk pengembang lainnya lebih efisien dan efektif.

5. Landasan Teori

Peranan utama dalam kajian penelitian adalah untuk menyusun hipotesa [16]. Kajian diperlukan dalam upaya menemukan permasalahan yang tertera jelas di setiap "*historical background*" [17].

Pengembangan sistem informasi berguna untuk memperdalam dan memperluas pengetahuan yang telah ada [19]. Pengembangan sistem informasi adalah kegiatan ilmu pengetahuan dan teknologi yang bertujuan untuk memanfaatkan kaidah dan teori ilmu pengetahuan yang telah terbukti kebenarannya untuk meningkatkan fungsi, manfaat, dan aplikasi ilmu pengetahuan dan teknologi yang telah ada, atau menghasilkan teknologi baru (Undang-Undang Republik Indonesia Nomor 18 Tahun 2002).

"Sistem informasi adalah suatu jaringan kerja dari prosedur-prosedur yang saling berhubungan, berkumpul bersama-sama untuk melakukan suatu kegiatan atau menyelesaikan suatu sasaran tertentu" "Dalam arti yang sangat luas, istilah sistem informasi yang sering digunakan merujuk kepada interaksi antara orang, proses algoritmik, data, dan teknologi. Dalam pengertian ini, istilah ini digunakan untuk merujuk tidak hanya pada penggunaan organisasi teknologi informasi dan komunikasi (TIK), tetapi juga untuk cara di mana orang berinteraksi dengan teknologi ini dalam mendukung proses bisnis" [20]

"*Open source* adalah sebuah sistem baru dalam mendistribusikan *software* kepada pengguna dengan memberikan program dan *source code*-nya secara gratis. Bahkan pengguna dapat mempelajari dan melakukan modifikasi untuk membuat *software* tersebut sesuai dengan kebutuhan mereka" [18]

Joomla dirilis dibawah GNU *General Public License* (GNU GPL). GNU GPL lisensi memberikan banyak kebebasan kepada anda lakukan dengan perangkat lunak.

Joomla adalah sebuah *platform* yang dinamis dan berkembang, dapat digunakan untuk membangun sebuah Web untuk keperluan hobi maupun berbasis bisnis (profesional). Komunitas pengguna Joomla ada di seluruh dunia, dan memiliki para

profesional tentang Joomla yang dapat membantu mendapatkan hasil maksimal tentang Joomla [10].

Auto generate adalah aplikasi yang di gunakan oleh peneliti untuk mempermudah menempatkan modul dan komponen pada Joomla dengan menggunakan PHP.

Secara garis besar “Kajian Pengembangan Sistem Informasi Berbasis *Open source* Joomla Dengan Menggunakan *Auto Generate*” adalah penyusunan hipotesis tentang kelemahan yang terjadi pada modul *design* Joomla yang berbasis *open source* dengan mengembangkan penyerderhanaan dalam pemetaan modul dan komponen pada Joomla.

6. Metodologi Penelitian

Metodologi yang digunakan dalam mendukung pelaksanaan penelitian ini, terdiri dari dua jenis *metode*:

Metode Analisis, diantaranya :

1. *Trend* milis.
Tren Joomla saat ini pokok bahasan seputar bagaimana mempromosikan suatu produk dengan menggunakan Joomla, juga membuat suatu sistem aplikasi dengan menggunakan Joomla dan pokok bahasan lainnya.
2. *Review document*
Melakukan *review document* bagaimana membuat suatu sistem aplikasi dengan menggunakan Joomla dari beberapa sumber buku yang mendukung dalam pembuatan suatu sistem aplikasi, dengan cara menganalisa komponen-komponen apa-apa saja yang dibutuhkan dan mengidentifikasi prasyarat kebutuhan sistem aplikasi yang akan dibangun.
3. *Best practices*
Membuat aplikasi Joomla dengan lebih sederhana menyingkat waktu pengerjaan dan mengurangi resiko kesalahan dalam penempatan maupun pemberian nama pada modul dan komponen dalam suatu blok.

Metode Perancangan Sistem.

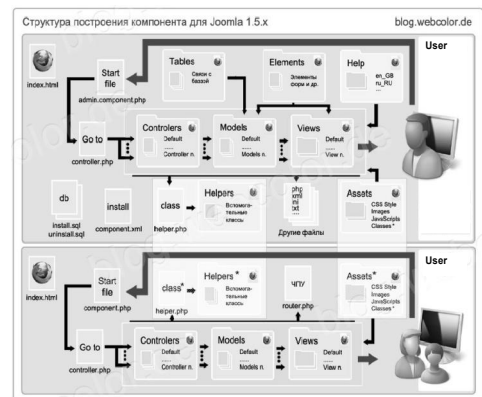
Pada tahapan perancangan sistem dilakukan perancangan model dan aplikasi berdasarkan hasil analisis terhadap

permasalahan yang dihadapi dengan merancang suatu sistem aplikasi *auto generate* yang dapat dengan mudah membantu menempatkan modul dan komponen dalam suatu blok.

Pada buku mastering Joomla dipaparkan bagaimana kita membuat sebuah aplikasi. Modul dan komponen merupakan bagian penting pada Joomla, *components* pada Joomla adalah ekstensi penting karena basic dari pembuatan *web design*. Setiap aplikasi Joomla dijalankan, komponen tersebut selalu digunakan untuk membuat dan menampilkan konten halaman utama [9].

Komponen Joomla terdiri dari 2 bagian utama: komponen *frontend* dan komponen *backend*. Komponen akan mengakses satu atau lebih *table database*, termasuk tabel inti serta tabel yang dibuat khusus untuk komponen yang digunakan. Komponen dasar yang digunakan dalam Joomla, yaitu *Model, View dan Controller*.

Detail komponen untuk aplikasi pada Joomla seperti pada gambar 1 berikut :



Gambar 1. Komponen untuk aplikasi Joomla

Mind Map Joomla ini akan membantu untuk lebih mudah di pahami. Pertama, cara membuat "hello world" di *Front End* dan *Back end*. Lalu bagaimana untuk mendaftar/*create* ke database. Setelah mendaftar, kita dapat menunjukkan link yang diakses komponen.

Setelah itu, kita memasuki *back end side*. dengan membuat *toolbar*. Lalu mempersiapkan tabel dalam database. Langkah selanjutnya: membuat *form* dan *list screen*, terakhir menuliskan kode untuk

menyimpan (*save*), mengedit (*edit*), dan menghapus data (*delete*) [9].

Peneliti mencoba untuk merangkum semua komponen-komponen yang dibutuhkan Joomla dalam membuat aplikasi seperti tersebut diatas dengan membuat variabel pada program PHP untuk lebih menyederhanakan dan mempermudah dalam pembuatan aplikasi dengan cara mendefinisikan setiap variabel yang dibutuhkan untuk komponen *view*, *controller* dan modul dalam satu program *auto generate*.

Seperti pemberian definisi untuk nama komponen, nama menu serta jumlah field yang dibutuhkan. Di bawah ini beberapa contoh program *Auto generate* tersebut :

```

72 $dir1 = $_POST['dir1'];
73 $numfield = $_POST['numfield'];
74 $menu = $_POST['menu'];
75 $field = $_POST['field'];
76 $typef = $_POST['typef'];
77 $sizef = $_POST['sizef'];
78 $descf = $_POST['descf'];
79 $tblnm = $_POST['tblnm'];
80 $fieldrel = $_POST['fieldrel'];
81 $fieldrel1 = $_POST['fieldrel1'];
82 $tblrelasi = $_POST['tblrelasi'];
    
```

Berikutnya membuat struktur komponen pada Joomla (*view*, *controller* dan modul) sebagai berikut:

```

119 mkdir($com, 0777);
120 mkdir($com."/admin", 0777);
121 mkdir($com."/site", 0777);
122 mkdir($com."/site/images", 0777);
123 mkdir($com."/admin/models", 0777);
124 mkdir($com."/admin/models", 0777);
125 mkdir($com."/admin/tables", 0777);
126 mkdir($com."/admin/views", 0777);
127 mkdir($com."/admin/views/".$dir1, 0777);
128 mkdir($com."/admin/views/".$dir1."/tmp", 0777);
129 mkdir($com."/admin/views/".$dir1."/s", 0777);
130 mkdir($com."/admin/views/".$dir1."/s/".$tmp, 0777);
131 mkdir($com."/site/controllers", 0777);
132 mkdir($com."/site/models", 0777);
133 mkdir($com."/site/views", 0777);
134 mkdir($com."/site/views/".$dir1, 0777);
135 mkdir($com."/site/views/".$dir1."/tmp", 0777);
136 mkdir($com."/site/views/".$dir1."/s", 0777);
137 mkdir($com."/site/views/".$dir1."/s/".$tmp, 0777);
    
```

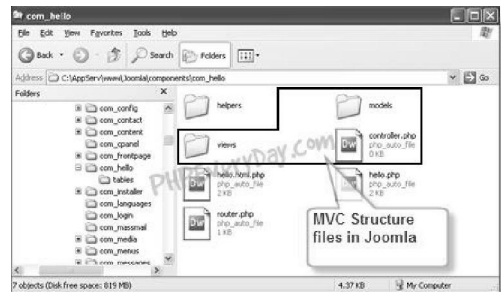
Dengan program *generate* secara otomatis, pembuatan modul aplikasi pada Joomla dapat menyingkat waktu pengerjaan.

7. Analisa Dan Implementasi Analisa

Sebuah kerangka kerja perangkat lunak adalah dasar dari sebuah aplikasi yang dapat digunakan oleh pengembang. Kerangka Joomla merilis banyak fitur untuk

pengembang. Joomla memandu pengembang melalui proses pengembangan komponen menggunakan *framework* [7].

Model-View-Controller (MVC) pada Joomla adalah pola desain perangkat lunak yang dapat digunakan untuk mengatur kode sedemikian rupa sehingga logika bisnis dan presentasi data terpisah [9]. Premis balik pendekatan ini adalah bahwa jika logika bisnis dikelompokkan menjadi satu bagian, maka *front-end* dan interaksi pengguna yang mengelilingi data dapat direvisi dan disesuaikan tanpa harus memprogram ulang logika bisnis. MVC pada awalnya dikembangkan untuk memetakan input, pengolahan, output kedalam sebuah arsitektur GUI. Ketiga peran utama adalah dasar untuk MVC Joomla [9].



Gambar 2. Struktur File MVC pada Joomla

Model

Model ini adalah bagian dari komponen yang merangkum data aplikasi. Model berisi metode untuk menambah, menghapus dan memperbarui informasi tentang data dalam database. Hal ini juga akan berisi metode untuk mengambil data dari database. Secara umum, teknik akses data yang mendasari harus berada di dalam model, model adalah satu-satunya elemen yang perlu diubah, bukan *viewer* atau *controller*.

Viewer

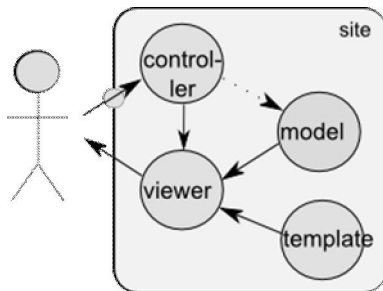
Viewer adalah bagian dari komponen yang digunakan untuk membuat data dari model dengan cara yang sesuai untuk interaksi. Untuk aplikasi berbasis web, *viewer* umumnya akan menjadi halaman HTML yang dikembalikan ke pengguna. Pandangan menarik data dari model (yang dilewatkan ke dari *controller*) dan *feed* data

ke dalam *template* yang dihuni dan disajikan kepada pengguna. *viewer* tidak menyebabkan data yang akan diubah dengan cara apapun, hanya menampilkan data yang diambil dari model.

Controller

Controller bertanggung jawab untuk menanggapi aksi pengguna. *Controller* akan menentukan permintaan yang sedang dibuat oleh pengguna dan merespon dengan tepat dengan mengacu kepada model untuk memanipulasi data secara tepat dan melewati model ke *view*. *Controller* tidak menampilkan data dalam model, hanya memicu metode dalam model yang telah di modifikasi.

Koneksi MVC



Gambar 3. Komponen dasar dalam Joomla

Gambar 3 menggambarkan komponen dasar yang digunakan dalam Joomla. Selain Model, *View* dan *Controller*, sebuah *Entry Point* telah ditambahkan yang digambarkan sebagai sebuah lingkaran kecil. Terlampir untuk penampil (*view*) *Template* telah ditambahkan. Dengan lima komponen tersebut pengembang dapat memahami tentang membuat Joomla dengan MVC komponen.

Implementasi

Aplikasi *Auto generate* dikembangkan pada flatform PHP dimana baik modul dan komponennya sudah ditempatkan pada *block* yang benar.

Pada Gambar 4 memperlihatkan variabel dari semua komponen yang diperlukan untuk sebuah aplikasi.

```

116 {
117 //echo "Doesn't exist" :
118 $com = $com . $dir1;
119 mkdir($com,0777);
120 mkdir($com."/admin",0777);
121 mkdir($com."/site",0777);
122 mkdir($com."/site/images",0777);
123 mkdir($com."/admin/controllers",0777);
124 mkdir($com."/admin/models",0777);
125 mkdir($com."/admin/tables",0777);
126 mkdir($com."/admin/views",0777);
127 mkdir($com."/admin/views/".$dir1,0777);
128 mkdir($com."/admin/views/".$dir1."/comp1",0777);
129 mkdir($com."/admin/views/".$dir1."/s",0777);
130 mkdir($com."/admin/views/".$dir1."/s/comp1",0777);
131 mkdir($com."/site/controllers",0777);
132 mkdir($com."/site/models",0777);
133 mkdir($com."/site/views",0777);
134 mkdir($com."/site/views/".$dir1,0777);
135 mkdir($com."/site/views/".$dir1."/comp1",0777);
136 mkdir($com."/site/views/".$dir1."/s",0777);
137 mkdir($com."/site/views/".$dir1."/s/comp1",0777);
138
139 //print "created":
140 $Handle = fopen($com."/admin/install.sql", 'w');
141 $Datasql = " CREATE TABLE IF NOT EXISTS #__".$dir1." (
142 id int(11) NOT NULL auto_increment,
143 ".$ssql."
144 PRIMARY KEY (`id`)
145 ) ENGINE=MyISAM AUTO_INCREMENT=0 DEFAULT CHARSET=utf8";
146 fwrite($Handle, $Datasql);
    
```

Gambar 4. Pemetaan komponen pada *auto generate*.

Setelah pemetaan modul dan komponen selanjutnya hasil pemetaan tersebut digunakan untuk menyimpan data berupa obyek. Gambar 5 memperlihatkan tampilan form *Auto generate* tersebut.

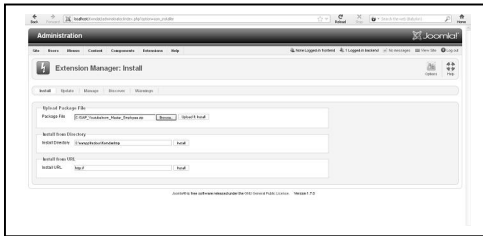
Gambar 5. *Form* Komponen *Generate*

Pada Gambar 6, *Component Name* digunakan untuk memberikan nama dari komponen yang akan dibuat, *Menu Name* untuk menamakan aplikasi menu sedangkan *Name of Field* adalah untuk menentukan berapa *field* yang akan digunakan untuk membuat suatu aplikasi. Gambar 6 menunjukkan salah satu contoh aplikasi.

Field Name	Type	Size	Description
UserID	varchar	15	UserID
StartDate	date		StartDated
Achievement	varchar	55	Achievement

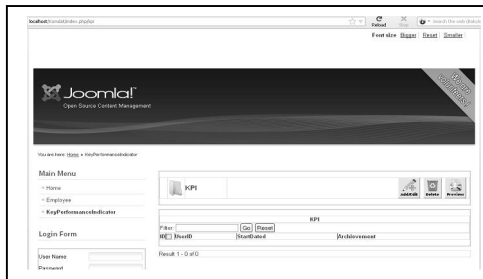
Gambar 6. Contoh Sebuah Aplikasi.

Setelah pengembang mengisikan form pada Gambar 6 tersebut dilanjutkan dengan menekan tombol “OK” untuk konfirmasi, maka system akan membuat satu modul yang berisikan komponen-komponen Joomla dalam bentuk zip (file kompres). Langkah selanjutnya pengembang melakukan *upload* dari *back-end* Joomla seperti Gambar 7.



Gambar 7. Upload Module Aplikasi

Gambar 8 menunjukkan menu dari *module* yang telah kita *upload* sebelumnya.



Gambar 8. Menu Key Performance Indicator (KPI)

Pada menu tersebut kita dapat melakukan proses penambahan/perubahan data dan menampilkan data yang telah kita input.

8. Kesimpulan

- Melalui aplikasi *auto generate* ini sebagai *tools* bagi orang yang berminat untuk belajar *open source* Joomla untuk membuat suatu aplikasi, baik sistem aplikasi bisnis pada perusahaan maupun pribadi.
- Sistem yang dikembangkan dalam bentuk menu yang interaktif, dengan menyediakan nama komponen, nama

menu juga *field* yang dibutuhkan dalam membuat sistem aplikasi dengan Joomla.

- Diperlukan penelitian lanjutan untuk menguji kinerja *web server* ketika menggunakan Joomla dengan versi terbaru.

Daftar Pustaka

- [1] DiBona, C., S. Ockman and M. Stone, *Open Sources: Voices from the Open Source Revolution*, O'Reilly Press, Sebastopol, CA, 1999.
- [2] Wheelwright, S.C. and Clark, K.B., *Revolutionizing Product Development: Quantum Leaps in Speed, Efficiency, and Quality*, Free Press, New York, 1992.
- [3] McCurdy, H.E., *Faster, Better, Cheaper: Low-Cost Innovation in the U.S. Space Program*, John Hopkins University Press, 2001.
- [4] Yamauchi, Y., M. Yokozawa, T. Shinohara, and T. Ishida, Collaboration with Lean Media: How Open-Source Software Succeeds, Proc. Computer Supported Cooperative Work Conf. (CSCW'00), 329-338, Philadelphia, PA, ACM Press, 2000.
- [5] Bergquist, M. and J. Ljungberg, The power of gifts: organizing social relationships in open source communities, *Information Systems Journals*, 11(4), 305-320, 2001.
- [6] Dinkelacker, J. P.K. Garg, R. Miller and D. Nelson, Progressive Open Source, *Proc. 24th Intern. Conf. Software Engineering*, Orlando, FL, 177-184, May, 2002.
- [7] North, B.M., Joomla 1.6 : A User's Guide. Building a Successful Joomla, Powered Website, 2011.
- [8] Bayross, I., Joomla 1.5 for Beginners : Opensourcevarsity learn without boundary, 2006.
- [9] Lanham, C. and Kennard, J., *Mastering Joomla! 1.5 Extension and Framework Development, Second Edition, The Professional Guide to Programming Joomla*, 2010.

- [10] Severdia, R. and Crowder, K., *Building Powerful and Efficient Websites, Using Joomla*, 2009.
- [11] Pavlicek, R. G., *Embracing Insanity: Open Source Software Development*. Indianapolis, IN: SAMS Publishing, 2000.
- [12] Voas, J., Faster, Better, and Cheaper, *IEEE Software*, 18(3), 96-97, May-June, 2001.
- [13] Sommerville, I., *Software Engineering (6th Edition)*, Addison-Wesley Pub Co., 2000.
- [14] Mockus, A., R.T. Fielding, and J. Herbsleb, Two case studies of open source software development: Apache and Mozilla, *ACM transactions on Software Engineering and Methodology*, 11(3), 309-346, July, 2002.
- [15] Zhao, L. and Elbaum, S., Quality assurance under the open source development model, *Journal Systems and Software*, 66, 65-75, 2003.
- [16] Long, Thomas J., Cs., *Completing Dissertation in the Behavioral Science and Education*, 1985.
- [17] Borg, Walter R., Cs., *Educational Research. An Introduction*, New York : Longman, 1983.
- [18] Stallman, Richard., *Free Software, Free Society, Selected Essays of Richard M. Stallman (2nd Edition)*, 2002.
- [19] Sugiyono, *Metode Penelitian Bisnis*, Cetakan keenam, Bandung, Penerbit : Alfabeta, 2004.
- [20] Kroenke, D.M., *Using MIS*, 2nd edition, Paperback, 2008