

# Prosedur efektif pengembangan aplikasi basis data

Zulkarnaen Hatala<sup>1</sup>

<sup>1</sup> Politeknik Negeri Ambon

Jl. Ir. Putuhena Wailela Rumah Tiga Ambon, 97234, Indonesia

*Abstract*—Efficient and quick procedure to build a web application is presented. The steps are intended to build a database application system with hundreds of tables. The procedure can minimize tasks needed to write code and doing manual programming line by line. The intention also to build rapidly web-based database application. In this method security concerning authentication and authorization already built in ensuring the right and eligible access of the user to the system. The end result is ready to use the web-based 3-tier application. Moreover, the application generated is still flexible to be customized and to be enhanced to suit more specific requirement in part of each module of the software both the server-side and client-side programming codes.

*Abstrak*—Pada tulisan ini diusulkan prosedur cepat dan efisien pengembangan aplikasi basis data menggunakan generator aplikasi. Bertujuan untuk meminimalisir penulisan bahasa pemrograman. Keuntungan dari prosedur ini adalah bisa digunakan untuk mengembangkan aplikasi basis data secara cepat terutama dengan sistem basis data yang terdiri dari banyak tabel. Hak akses dan prosedur keamanan standar telah disediakan sehingga setiap user terjamin haknya terhadap entitas tertentu di basis data. Hasil generasi adalah aplikasi basis data 3-tier berbasis web yang siap pakai. Sistem aplikasi yang terbentuk masih sangat lentur untuk dilakukan penyesuaian setiap komponen aplikasi baik di sisi server maupun di sisi client.

*Keywords*—Web Programming, Application Builder, Rapid Application Generator, 3-tier web application

## I. Pendahuluan

### A. Latar Belakang

Sering dijumpai beberapa kasus di mana aplikasi atau sistem informasi dengan komponen basis data ingin dikembangkan secara cepat untuk segera digunakan *end user* pada proses bisnis. Tetapi tabel yang akan ditampilkan banyak sekali hingga puluhan bahkan ratusan. Di samping itu sering kali lembaga membutuhkan data personal dalam dalam waktu cepat. Misalkan perusahaan ingin mendapatkan data dari pelanggan dan pegawai. Perguruan Tinggi menginput data para mahasiswa dan dosen pengajar. Lembaga pemerintah ingin mendapatkan informasi sosial personal dari masyarakat. Lembaga survei menginginkan tanggapan dari para responden. Maka cara konvensional yang digunakan yaitu dengan mendata langsung ke lapangan dengan kata lain mendatangi para personal. Atau sebaliknya para respondenlah yang mendatangi unit-unit administrasi untuk memberikan data mereka. Di sini dibutuhkan strategi pengembangan aplikasi basis data secara cepat dan efisien.

Metoda lebih baru adalah memanfaatkan infrastruktur jaringan LAN atau internet untuk akuisisi data tanpa bertemu langsung antara personal pemilik data dengan petugas administrasi penginput data.

Untuk metoda tidak langsung ini maka diperlukan membuat aplikasi komputer yang bisa menangkap data melalui jaringan internet atau LAN yang ada. Salah satu solusi umum adalah menggunakan aplikasi *client-server* berbasis **web** HTTP. Di mana personal dan responden menginputkan data menggunakan aplikasi komputer yang disebut peramban atau browser untuk menghubungi server web yang disediakan oleh peminta data. Server web ini terpisah dari komputer responden dan dihubungkan dengan infrastruktur jaringan komputer.

Selain menggunakan model *browser-web server* cara lain adalah menggunakan aplikasi khusus *client-server* tidak berbasis web HTTP. Keuntungan menggunakan model berbasis web adalah karena aplikasi *client* yaitu browser sudah tersedia secara umum di komputer-komputer desktop maka cukup mengembangkan aplikasi di sisi server saja. Sedangkan jika menggunakan model *client-server* tidak berbasis web, maka pengembangan aplikasi selain dilakukan di sisi *client* harus juga dilakukan di sisi server. Tentunya hal ini tidak nyaman dibanding berbasis web, karena harus mengembangkan aplikasi server dan *client* bersamaan. Teknik ketiga adalah mengembangkan *client* yang berkomunikasi langsung dengan server database secara langsung. Misalnya aplikasi yang langsung mengakses Server

mysql menggunakan *OS native library*. Teknik ini menimbulkan resiko dalam hal keamanan di mana *client* dibebaskan untuk mengakses langsung ke *server database*. Terdapat kemungkinan kode aplikasi *client* dirubah dan dikompilasi ulang untuk kepentingan *hacker*.

Walaupun model-model pengembangan aplikasi web sudah mapan. Tetapi dari sisi sumber daya manusia proses pengembangan aplikasi web membutuhkan level-level tertentu penguasaan di teknologi pemrograman baik di sisi *client (client-side)* maupun sisi *server (server-side)*. Di sini salah satu solusi yang ditawarkan adalah generator aplikasi di mana tugas pemrograman bisa diminimalisir bahkan dihindari.

**B. Permasalahan**

Jadi dalam mengembangkan aplikasi basis data diidentifikasi permasalahan sebagai berikut:

- 1) Sumber daya manusia yang terbatas baik secara kualitas dan kuantitas.
- 2) Waktu pengembangan aplikasi yang terbatas
- 3) Jumlah entitas dan tabel yang sangat banyak menyebabkan pemrograman formulir yang sangat banyak.

**C. Teori Pendukung**

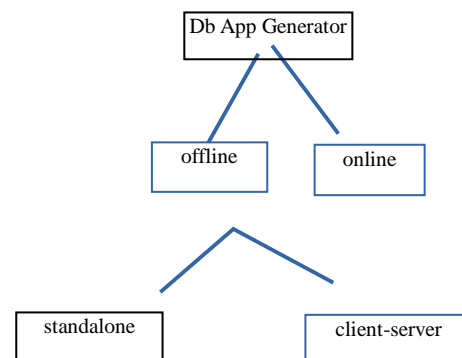
Membuat aplikasi secara *otomatis* bertujuan untuk menghindari atau meminimalisir penulisan bahasa pemrograman secara manual. Hal ini bukan berarti aplikasi tidak menggunakan bahasa pemrograman, melainkan bahwa rutin, source code dan listing pemrograman akan dibuat (*generated*) secara otomatis. Dengan demikian persyaratan pemahaman, penguasaan dan kemahiran terhadap bahasa pemrograman tersebut menjadi berkurang atau diminimalisir. Dalam hal tabel dan formulir yang banyak

Untuk aplikasi basis data baik *standalone* hingga 3-tier [1], apabila diotomatisasi maka bahasa pemrograman yang diminimalisir penulisan kode manual termasuk:

- 1) bahasa *SQL (structured query language)*
- 2) bahasa *server side (JSP, ASP, CGI, Twig)*
- 3) bahasa *client side (HTML, Java Script, CSS)*
- 4) bahasa khusus platform OS seperti *Basic, Pascal, Java, C, C++, C# dan lain-lain*.

Setelah proses generasi otomatis, maka rutin yang tercapai biasanya masih ingin dirubah-rubah sesuai keinginan.

Terdapat beberapa *platform* untuk membuat aplikasi basis data tanpa menulis kode. Pada gambar 1 dibedakan antara yang *online* dan yang *offline*. Untuk menggunakan generator aplikasi online harus tersedia koneksi internet karena generator aplikasi diletakkan di server internet secara online oleh vendor. Aplikasi yang dihasilkan juga hanya bisa diakses secara online. Sedangkan generator offline tidak membutuhkan koneksi internet. Generator offline kemudian bisa dibedakan menjadi generator jenis berdikari (*standalone*) dan *client-server*. Hasil dari generator berdikari tidak bisa diakses melalui jaringan LAN sedangkan hasil dari generator *client-server* bisa diakses dari jaringan LAN. Generator yang offline *client-server* bisa ditingkatkan untuk diletakkan secara online karena kompatibilitas protokol TCP/IP antara LAN dengan internet.



Gambar 1 : Tipe DB App Generator

Tipe *online generator* contohnya adalah Knack, Caspio dan TrackVia [2]. Tipe generator yang berdikari (*standalone*) adalah seperti Microsoft Access (MA) [3]. Formulir untuk jumlah tabel yang sangat banyak bisa di-*generated* secara mudah menggunakan MA. Bahkan dengan mengkonfigurasi *data-source*, MA bisa digunakan melalui jaringan LAN yang ada, dalam hal ini membuat MA bisa dikategorikan sebagai tipe *client-server* atau tipe online.

**II. Metodologi**

Untuk menyelesaikan permasalahan yang telah dipaparkan sebelumnya, maka penelitian ini mengusulkan langkah-langkah umum (*main strategy*) untuk pengembangan aplikasi basis data dengan menggunakan generator aplikasi. Dikembangkan juga suatu generator aplikasi yang akan membuat aplikasi 3-tier berbasis web secara otomatis. Presedur yang diusulkan telah diujicobakan untuk contoh kasus proses “registrasi mahasiswa secara online” di perguruan tinggi.

Langkah-langkah pembuatan aplikasi lengkap basis data secara otomatis yang diusulkan adalah sebagai berikut:

1) *Otomatisasi pembuatan basis data relasional*, bertujuan untuk menghindari atau meminimalisir pengkodean dalam bahasa SQL Langkah ini tidak diperlukan jika basis data sudah tersedia.

2) *Otomatisasi aplikasi basis data 3-tier berbasis web*, bertujuan untuk menghindari atau meminimalisir pengkodean dalam bahasa pemrograman web server-side dan client-side. *Simple DB Application Generator, SDbApp* [4] adalah generator *backend* dan *frontend* aplikasi basis data yang dibuat dalam bahasa PHP. Untuk menggunakan SDbApp maka harus terinstall interpreter PHP di Sistem Operasi yang diinginkan.

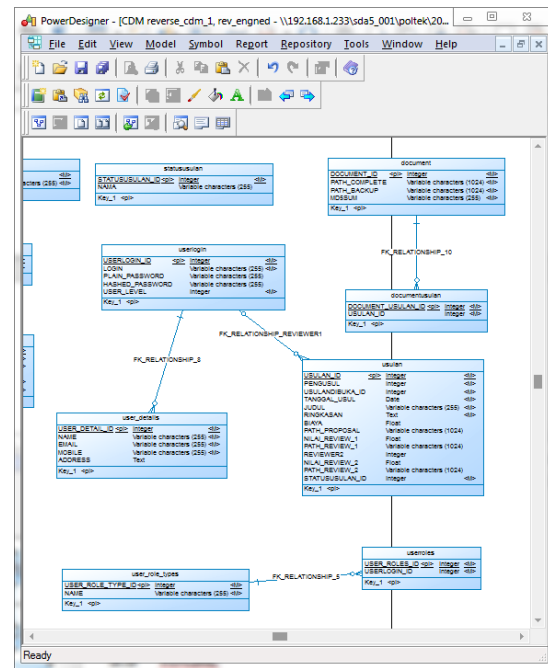
3) *Kustomisasi hak akses*. Langkah ini dibutuhkan jika diinginkan membatasi hak akses terhadap aplikasi yang di-generated.

4) *Kustomisasi Antar muka grafis*. Langkah ini dibutuhkan jika tampilan antar muka grafis ingin dirubah-rubah dan disesuaikan. Langkah ini bisa dilakukan secara bebas tanpa harus mengulangi langkah 1 dan 2.

**D. Otomatisasi pembuatan basis data pada MySQL**

Membuat *Entity Relationship Diagram, ERD* [5] adalah tahap pertama yang harus digunakan jika basis data belum tersedia. ER dibuat dengan aplikasi pemodelan ER, *Sybase Power Designer, PD*[6]. Keuntungan menggunakan PD adalah model analisa basis data semakin mudah dan mengurangi kesalahan manusia (*human error*). Di samping itu kelebihan lainnya adalah kode SQL untuk struktur basis data bisa langsung dibuat secara otomatis. Tentunya kelebihan PD akan semakin dirasakan jika tabel yang akan dibuat sangat banyak dalam jumlah waktu yang singkat. Contoh tampilan desain ER dengan Power Designer sebagaimana pada Gambar 2.

Apabila ingin membuat tabel tabel dari basis data secara manual satu persatu, maka bisa menggunakan aplikasi seperti *PHPMysqlAdmin* atau *MySQLFront*. Form pembuatan tabel pada *PHPMysqlAdmin* dan *MySQLFront* masih lebih mudah digunakan jika dibandingkan dengan memprogram manual menggunakan bahasa SQL. Kelemahan cara manual adalah jika tabel yang ingin dibuat banyak sekali mencapai puluhan, ratusan hingga ribuan tabel. Keuntungan lainnya menggunakan Power Designer adalah terlihatnya secara visual hubungan relasional antar tabel-tabel yang terkait.



Gambar 2 : Membuat ER-Diagram dengan Sybase Power Designer

**E. Otomatisasi aplikasi basis data 3-tier berbasis web**

Setelah basis data selesai dibuat maka aplikasi web bisa dibuat secara otomatis menggunakan *Simple Database Application Generator, SDbApp* [4]. Disini kode-kode *server side* yang berhubungan ke RDBMS MySQL maupun kode-kode *client side* form web untuk tampilan operasi *Create, Read, Update* dan *Delete (CRUD)* bisa di-generated secara otomatis. Dalam hal ini bertujuan untuk menghindari pengkodean bahasa/framework *server-side* antara lain PHP, Silex, Symfony, Twig dan juga bahasa *client side* antara lain AdminLTE, HTML, CSS, JS.

**F. Kustomisasi Hak akses**

Permission atau hak akses menentukan operasi CRUD tertentu yang boleh dilakukan oleh pengguna terhadap objek basis data dan membatasi serta menyaring operasi lainnya. SdbApp telah menyediakan fungsi autentifikasi dan otorisasi [7]. Dengan SDbApp maka hak akses diberikan per *record level* dari suatu tabel. Prototipe fungsi hak akses antara user terhadap *record* di *table* didefinisikan dalam *pseudocode* Java berikut:

- 1) *function GrantCreateRecordPermission(user, table);*
- 2) *function GrantReadRecordPermission(user, table, records []);*

3) *function GrantUpdateRecordpermission(user, table, records[]);*

4) *function GrantDeleteRecordPermission(user, table, records[]);*

**G. Role**

Dalam kaitannya dengan keamanan, maka Role adalah bagaimana mengelompokkan beberapa pengguna ke suatu pola hak akses yang mirip. Misalkan pada studi kasus pada proses registrasi mahasiswa baru melalui website online. Maka dalam hal ini sebut saja 3 *role* yang terkait:

- *Administrator*
- *Guest*
- *Registered/Applicant*

SdbApp[4] telah menyediakan role standar Administrator dan Guest. Administrator adalah *role* dengan akses penuh terhadap sistem. Dalam level *record*, maka para Administrator mempunyai akses penuh (CRUD) terhadap semua *record* di semua tabel. Guest pada satu sisi tidak punya akses apapun terhadap salah satu tabel. Tetapi *Guest* mungkin saja mendaftar untuk menjadi calon mahasiswa. Setelah pendaftaran, seorang Guest berubah role menjadi Registrar. Di level *record*, registrar atau calon mahasiswa baru bisa mempunyai akses RU tetapi tidak mempunyai akses CD di tabel registration. Hal ini karena seorang mahasiswa yang sudah terdaftar tidak boleh mendaftar lagi (CREATE), dan tidak boleh menghapus data pendaftarannya (DELETE). Seorang registrar tentu bisa mengubah data-data pendaftarannya (UPDATE,READ), mungkin karena salah pengetikan atau karena dirasa belum lengkap.

*Role* kadang mengharuskan mekanisme yang tidak bisa diakomodasi di level akses per record. Hal ini karena untuk satu tabel role mengharuskan hak akses terhadap kolomnya berbeda-beda. Contoh kasus pada tabel registrasi calon mahasiswa, setelah pendaftaran maka di tabel tersebut mahasiswa bisa merubah data pendaftarannya, tetapi tidak bisa merubah data-data tertentu seperti primary key di tabel tersebut. Bahkan kadangkala akses UPDATE terhadap record bisa dibatasi oleh sesuatu yang sangat tergantung proses bisnisnya. Misalnya saja tiap calon mahasiswa cuma bisa merubah data sampai waktu tertentu, setelah waktu tersebut hak akses ke record di tabel dibatasi cuma READ.

**H. Implementasi hak akses**

Hak akses standar sudah diimplementasikan oleh SdbApp berupa rutin-rutin PHP di *server side*

untuk menyaring akses ke database MySQL. Rutin-rutin PHP ini bisa ditulis secara terpusat pada satu file tunggal ataupun disebar secara modular di file-file berukuran kecil.

**I. Kustomisasi tampilan pengguna**

Untuk menyesuaikan tampilan maka bisa dilakukan tanpa harus memahami pemrograman HTML secara intensif, menggunakan aplikasi seperti *Macromedia Dreamweaver* (MD). MD juga bisa mempercepat waktu kustomisasi tampilan halaman hasil generasi SdbApp.

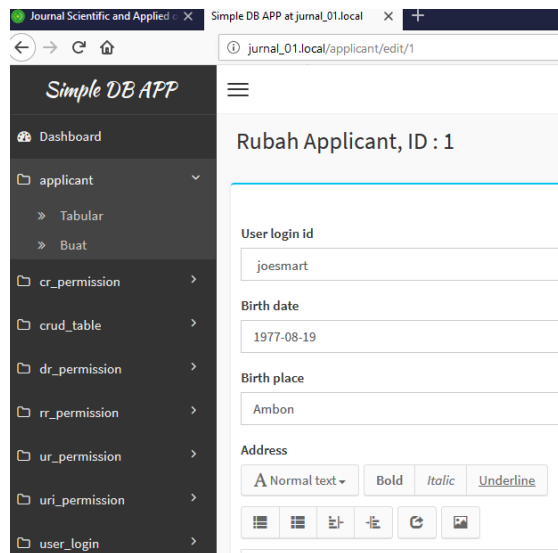
**III. Hasil**

Penelitian ini telah berhasil mengembangkan suatu generator aplikasi 3-tier berbasis web yaitu SdbApp[4]. Penelitian ini mengusulkan prosedur efisien untuk mengembangkan aplikasi basis data 3-tier berbasis web. Prosedur ini telah berhasil digunakan untuk membuat aplikasi registrasi mahasiswa baru pada perguruan tinggi. Generasi dihasilkan pada lingkungan software dan bahasa pemrograman yang sebagaimana terdaftar pada gambar 3:

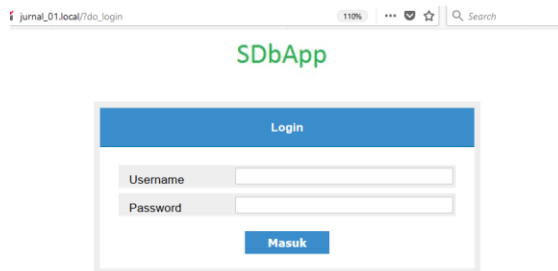
No	Software	Versi
1	Sybase Power Designer	12.5.0.2169
2	Xampp	3.2.2
	Apache	2.4
	PHP	5.6.39
	MySQL	10.1.37
3	SdbApp	0.1
	PHP-Composer	1.8.0
	PHP-Silex	1.2
	Twig	1.8.0
4	Microsoft Windows	Pro 10.0.14393
5	Firefox	57.0(64-bit)

Gambar 3 : Software yang digunakan dalam penelitian

Setelah berhasil di-generated maka aplikasi *back-end* dihosting di server Apache-mysql-Xampp. Tampilan aplikasi *front-end* bisa dilihat di *browser* seperti Firefox. Salah satu contoh tampilan hasil generasi aplikasi adalah pada gambar 4. Sedangkan fungsionalitas sekuritas[7] standar autentifikasi dan otorisasi bisa dicontohkan dalam gambar 5 dan gambar 6.



Gambar 4 : Tampilah hasil generasi aplikasi



Gambar5 : Contoh form fungsi autentikasi



Gambar 6 : Contoh tampilan fungsi otorisasi

#### IV. Kesimpulan dan Saran

##### J. Kesimpulan

Mengembangkan basis data aplikasi menggunakan prosedur yang diusulkan dengan menggunakan SdbApp[4] adalah lebih cepat dan efisien. Terutama dengan sistem basis data dengan banyak tabel. SDbApp sudah menyediakan keamanan standar berupa autentikasi dan otorisasi. Hasil generasi dari SdbApp masih bisa dikustomisasi di setiap komponen dan modul untuk beradaptasi dengan kebutuhan khusus yang diinginkan. Untuk mengkustomisasi hasil generasi dari SdbApp mensyaratkan pemahaman bahasa pemrograman dan

framework yang digunakan oleh SdbApp yaitu PHP-Silex, dan AdminLTE.

##### K. Saran

Untuk pengembangan ke depan SdbApp harus menampilkan menu konfigurasi yang lebih *user friendly*, lebih banyak fungsionalitas dan lebih mudah digunakan. Tentu juga harus menghasilkan aplikasi yang bisa beradaptasi dengan cara dikonfigurasi dan lebih meminimalisir lagi penulisan code bahasa pemrograman.

#### Referensi

- [1] D. Wall, *Multi-Tier Application Programming with PHP: Practical Guide for Architects and Programmers*. 2004.
- [2] J. Maria, "Top 10 Online Database App Builders to Create your PM Tool." [Online]. Available: <https://project-management.com/top-10-database-application-builders-for-creating-your-own-pm-tool/>. [Accessed: 26-Dec-2018].
- [3] J. Eckstein and B. R. Schultz, *Introductory Relational Database Design for Business, with Microsoft Access*. Wiley, 2018.
- [4] Z. Hatala, "SDBApp, Simple Database Application Generator," 2018. [Online]. Available: <https://github.com/dzhatala/simpledbappgenerator>. [Accessed: 24-Dec-2018].
- [5] I. Song, M. Evans, and E. K. Park, "A Comparative Analysis of Entity-Relationship Diagrams," *J. Comput. Softw. Eng.*, vol. 3, no. 4, pp. 427–459, 1995.
- [6] W. Xiao-Yun, "PowerDesigner Chief Architect." 2005.
- [7] "Web Application Security Fundamentals | Microsoft Docs." [Online]. Available: [https://docs.microsoft.com/en-us/previous-versions/msp-n-p/ff648636\(v=pandp.10\)](https://docs.microsoft.com/en-us/previous-versions/msp-n-p/ff648636(v=pandp.10)). [Accessed: 26-Dec-2018].