

Prosedur efektif pengembangan aplikasi basis data

Zulkarnaen Hatala¹

¹ 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 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 penelitian kali 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 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

I. Pendahuluan

A. Latar Belakang

Sering dijumpai beberapa kasus di mana aplikasi front-end basis data ingin dikembangkan secara cepat agar siap digunakan *end user* pada proses bisnis sehari-hari. Tetapi tabel yang akan ditampilkan banyak sekali hingga puluhan bahkan ratusan. Di sini dibutuhkan strategi pengembangan aplikasi basis data secara cepat dan efisien. Di samping itu sering kali lembaga membutuhkan data personal 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 atau mendatangi para personal. Atau sebaliknya para respondenlah yang mendatangi unit-unit administrasi untuk mengisikan mereka.

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.

Teknik ketiga adalah mengembangkan client yang berkomunikasi langsung dengan server database secara langsung misalnya aplikasi yang langsung mengakses Server mysql menggunakan OS native library. Tentunya hal ini tidak nyaman dibanding berbasis web karena harus mengembangkan aplikasi server dan client bersamaan.

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 web 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

Dalam mengembangkan aplikasi basis data sering dijumpai masalah-masalah 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

II. Tinjauan Pustaka

C. Basis Data Relasional

Basis data relasional (*relational database*) adalah teknik penyimpanan data atau informasi menggunakan struktur tertentu dimana sekelompok informasi berkaitan dikelompokkan dalam bentuk-bentuk tabular (*table*). Tabel-tabel ini saling mempunyai hubungan (*relation*). Contoh *table* misalnya tabel *user* yang diperuntukkan menyimpan data *login* dan *password* pengunjung suatu website. Setiap table dikelompokkan lagi ke dalam kolom-kolom (*columns*) atau *fields*. Contoh *fields* dari tabel *user* misalnya *login*, *password*, *email*, *phone number* dan *address*. Nama *table* beserta fieldnya disebut *structure* dari *table* tersebut. Sedangkan data atau *row* atau *record* atau isi dari tabel ditambahkan dengan cara dilajurkan sebagai baris-baris (*rows*) dengan *column header* berupa nama *fields*. Contoh *table* diberikan pada Gambar 1.

Id	login	password	email
1	admin	xzC82)]3fd	admin@test.ac.id
2	user 01	SdfsdfC33	user 01@test.edu

Gambar 1 : Table *user_login* untuk menyimpan data login dan password

Paket software aplikasi yang digunakan untuk/ mengelola *database* relasional dalam format aslinya disebut dengan *Relational Database Management System* (RDBMS). Sering juga disebut *database server*. Contoh dari RDBMS adalah MySQL[1], dan Microsoft Access[2].

Operasi *record* pada tabel ini didefinisikan sebagai berikut:

Create : membuat record baru di suatu tabel
Read : membaca record-record di tabel
Update : merubah/mengedit record yang sudah ada
Delete : menghapus record yang telah ada dari tabel

Ke empat operasi di atas biasa disingkat dengan istilah operasi CRUD.

D. Pola MVC untuk pengembangan Aplikasi web

Pada model paling dasar, bahasa pemrograman web terbagi atas dua bagian, yaitu Server side (PHP, ASP, CGI-x, Twig) dan pemrograman client side atau browser side yaitu HTML, Javascript dan CSS [1]. Peramban (internet browser) didesain untuk cuma bisa memahami sintaks bahasa client-side (HTML, CSS, JavaScript). Bahasa server-side ditujukan untuk melakukan pemrosesan di sisi server untuk menciptakan konten-konten *client-side* yang akan dikirim ke browser melalui jaringan. Pola pengembangan modular **Model View Controller** (MVC) bisa juga diterapkan disini dengan Twig sebagai 'V', *php-mysql* sebagai 'M', *twig* dan *php-silex-symfony* sebagai 'C'.

E. Protokol HTTP

Dalam model komunikasi client-server dalam skala global, HTTP digunakan di Internet untuk berkomunikasi antara browser dengan web server.

F. Membuat aplikasi basis data secara otomatis

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.

Untuk aplikasi basis data, apabila diotomatisasi maka bahasa pemrograman yang dihindari berupa:

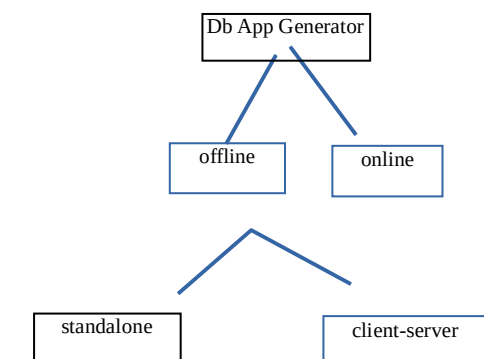
- 1) bahasa *SQL* (*structured query language*)

- 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 *pengembang*.

G. Platform Aplikasi basis data yang ada

Terdapat beberapa platform untuk membuat aplikasi basis data tanpa menulis kode. Pada gambar 2 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. Aplikasi yang dihasilkan juga bisa diakses secara online. Sedangkan generator offline tidak membutuhkan koneksi internet. Generator offline kemudian bisa dibedakan menjadi generator jenis *berdikari (standalone)* dan *client-server*. Untuk yang generator *berdikari* tidak bisa diakses melalui jaringan LAN sedangkan yang *client-server* bisa diakses dari jaringan LAN. Yang offline *client-server* bisa diupgrade untuk diletakkan secara online karena kompatibilitas protokol TCP/IP pada LAN dan internet.



Gambar 2 : Tipe DB App Generator

Tipe *online generator* contohnya adalah Knack, Caspio dan TrackVia [3]. Tipe generator yang *berdikari (standalone)* adalah seperti Microsoft Access (MA) [2]. Formulir untuk jumlah tabel yang sangat banyak bisa digenerate secara cepat menggunakan MA.

Bahkan dengan mengkonfigurasi *data-source*, MA bisa digunakan melalui jaringan LAN yang ada membuatnya dikategorikan sebagai tipe *client-server* atau tipe online.

III. Otomatisasi Aplikasi basis data berbasis web

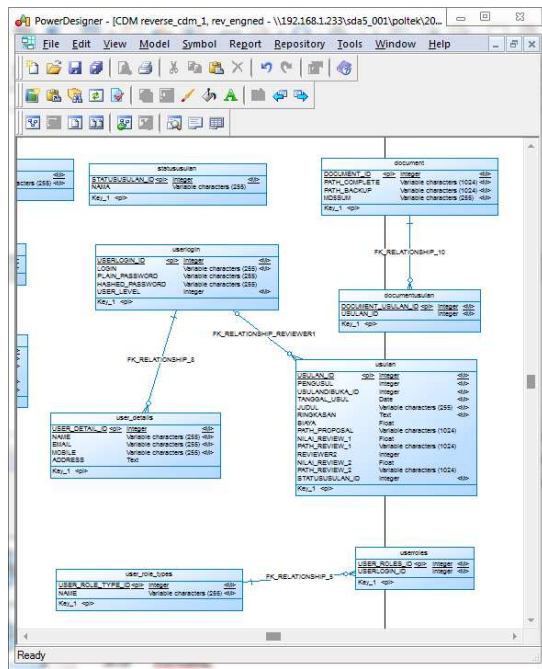
Langkah-langkah pembuatan aplikasi lengkap basis data secara otomatis yang diusulkan adalah sebagai berikut:

- 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 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 digenerate.
- 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.

H. Otomatisasi pembuatan basis data

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*). Disamping 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 3.

Apabila ingin membuat tabel dari basis data secara manual satu persatu, maka bisa menggunakan aplikasi seperti PHPMyAdmin atau MySQLFront. Form pembuatan tabel pada PHPMyAdmin 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 3 : Membuat ER-Diagram dengan Sybase Power Designer

I. Mengenerate formulir CRUD secara otomatis

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 CRUD bisa digenerate secara otomatis. Dalam hal ini bertujuan untuk menghindari pengkodean bahasa *server-side* PHP, Silex, Symfony, Twig dan bahasa *client side* HTML, CSS, JS.

J. 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 didalam suatu tabel. Prototipe fungsi hak akses didefinisikan sebagai berikut:

1) function *GrantCreateRecordPermission*(user, table);

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

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

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

K. Role

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

- *Administrator*
- *Guest*
- *Registered*

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.

L. Implementasi hak akses

Hak akses diimplementasikan pada 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.

M. Kustomisasi tampilan pengguna

Untuk mempermanis 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 HTML statis

IV. HASIL

Dalam penelitian ini diujicoba pengembangan aplikasi berbasis web dengan menggunakan Web Generator, SDbApp. Daftar semua software dan bahasa pemrograman yang terlibat dalam penelitian sebagai terlihat pada gambar 4:

No	Software	Versi
1	Sybase Power Designer	12.5.0.2169
2	XAmp	3.2.2
	Apache Web Server	2.4
	PHP	5.6.39
	MySQL	10.1.37
3	SDbApp	0.1
	PHP-Composer	-
	PHP-Silex	-
	Twig	-
	HTML,CSS, JS	-
4	Microsoft Windows 10 Pro	10.0.14393
5	Firefox	57.0(64-bit)

Gambar 4 : Software yang digunakan dalam penelitian

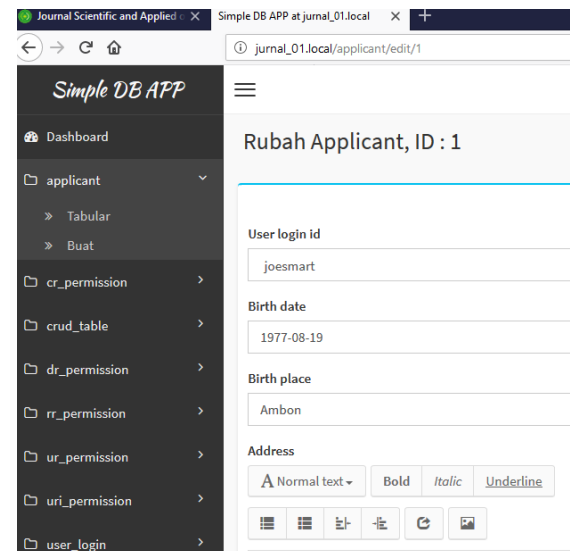
Di sini telah berhasil dikembangkan aplikasi registrasi mahasiswa baru dengan cepat mengikuti metoda yang diterangkan dalam bab III. Salah satu tampilan hasil generasi aplikasi adalah pada gambar 5. Sedangkan fungsionalitas sekuritas standar autentifikasi dan otorisasi bisa dicontohkan dalam gambar 6 dan gambar 7.

V. Kesimpulan dan Saran

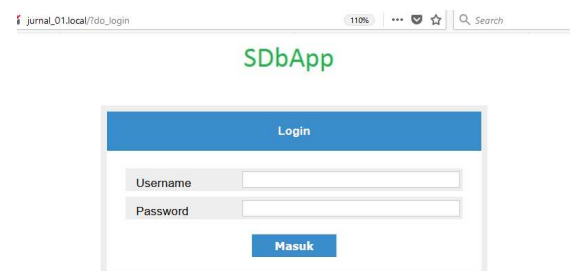
N. Kesimpulan

Mengembangkan basis data aplikasi menggunakan **SDbApp** [4] adalah lebih cepat dan efisien. Terutama dengan sistem basis data dengan banyak tabel. SDbApp sudah menyediakan keamanan standar berupa autentifikasi 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

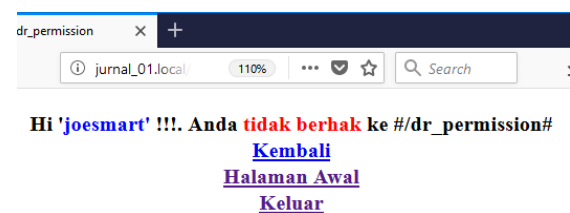
framework yang digunakan oleh SdbApp yaitu PHP, Silex, Symfony, HTML, JavaScript dan Cascade Style Sheet.



Gambar 5 : Tampilan hasil generasi aplikasi



Gambar 6 : Contoh form fungsi autentifikasi



Gambar 7 : Contoh tampilan fungsi otorisasi

O. Saran

Untuk pengembangan ke depan SdbApp harus menampilkan menu konfigurasi yang lebih *user friendly*, lebih banyak fungsionalitas dan lebih mudah digunakan.

Referensi

- [1] D. Wall, *Multi-Tier Application Programming with PHP: Practical Guide for Architects and*

- Programmers*. 2004.
- [2] J. Eckstein and B. R. Schultz, *Introductory Relational Database Design for Business, with Microsoft Access*. Wiley, 2018.
- [3] 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].
- [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].