

Pembangkitan Formulir Web Berdasarkan Metadata SQL dan Spesifikasi W3C

Muhammad Fikry¹, Yusra², Taufik Hidayat³

^{1,2,3}Teknik Informatika UIN Sultan Syarif Kasim Riau

Jl. H.R. Soebrantas no. 155 KM. 18 Simpang Baru, Pekanbaru 28293

macfikry@gmail.com¹, usera84@yahoo.com², taufik.hidayat7@students.uin-suska.ac.id³

Abstrak - Dalam membangun sistem informasi berbasis web, terdapat di dalamnya sebuah tahapan membangun formulir data isian sebagai representasi basis data didalam sistem informasi. Formulir menjadi jalan utama untuk memasukkan data kedalam basis data melalui sistem informasi. Dalam membangun formulir sistem informasi berbasis web, seorang user interface programmer akan merancang formulir dengan elemen-elemen HTML yang sesuai dengan struktur basis data. Penelitian ini membahas tentang bagaimana membangun aturan-aturan pembangkitan formulir dan mengimplementasikan aturan-aturan tersebut kedalam aplikasi pembangkit formulir. penelitian dilakukan terhadap standar bahasa SQL dan standar penulisan tag HTML, kemudian dilakukan pemetaan elemen SQL menjadi elemen formulir HTML sebagai acuan dalam membangun aturan-aturan pembangkitan formulir. Setelah itu dilakukan analisa terhadap model hubungan data pada RDBMS serta menganalisa dampaknya terhadap formulir. Hasil analisa aturan-aturan pembangkitan formulir akan diimplementasikan kedalam aplikasi pembangkit formulir berbasis web berdasarkan metadata SQL. Berdasarkan hasil pengujian yang dilakukan menggunakan Black Box dan User Acceptance Test, aplikasi pembangkit dapat dibangun dan berjalan dengan baik dalam membangkitkan formulir HTML.

Kata kunci - Basis Data, HTML, Pembangkit Formulir Web, SQL, iraise, keluhan, klasifikasi, rapidminer, support vector machine

PENDAHULUAN

Antarmuka merupakan salah satu komponen penting dalam sebuah perangkat lunak, antarmuka merupakan jembatan utama antara pengguna dengan perangkat lunak untuk berinteraksi. Dalam membangun sebuah perangkat lunak berbasis web, tentunya akan dilakukan tahap pembuatan antarmuka, yaitu seorang User Interface (UI)

Programmer (selanjutnya disebut UI Programmer) akan menyajikan data yang disimpan di dalam basis data ke dalam bentuk antarmuka yang dilihat oleh pengguna. UI Programmer harus menerjemahkan elemen-elemen dalam sebuah skema basis data ke dalam kumpulan Tag HTML (Hypertext Markup Language) yang menjadi komponen dasar antarmuka perangkat lunak berbasis web, tentunya untuk membangun antarmuka sebuah perangkat lunak seorang UI Programmer harus paham dengan struktur basis data yang digunakan. Dalam membangun antarmuka pada umumnya UI Programmer akan melakukannya dengan mengetikkan sintaks HTML secara manual.

Jika dilihat dari fase siklus pengembangan perangkat lunak tradisional atau Software Development Life Cycle (SDLC) menurut Peter Rob dan Carlos Coronel pada awal fase implementasi, desain basis data sudah diimplementasikan ke dalam Database Management Systems (DBMS) dan pada fase inilah siklus pembuatan kode program dan pengujian akan terjadi. Rob dan Coronel juga menyatakan bahwa tahap implementasi dan pengujian ini dapat memakan waktu 50% sampai 60% dari total waktu keseluruhan pengembangan perangkat lunak dan dengan adanya generator aplikasi dapat mengurangi waktu dalam proses pembuatan kode program dan pengujian. Kemudian berdasarkan penelitian yang dilakukan oleh Sutton dan Sprague di diperoleh hasil yang memperkirakan bahwa 50% waktu desainer dihabiskan pada proses mendesain kode untuk antarmuka [1].

Seiring berjalannya waktu banyak dilakukan penelitian terkait otomatisasi dalam membangkitkan formulir web, seperti penelitian yang dilakukan oleh Rein Raudjärv yang membahas tentang membangkitkan web form berdasarkan skema dinamis yang ditulis dalam bahasa Java. Pada penelitian ini aplikasi pembangkit web form DynaForm menerima masukan berupa XML (Extensible Markup Language) skema yang menyediakan struktur form. Berdasarkan masukan yang diteima oleh DynaForm, maka akan dihasilkan keluran berupa web form dengan tambahan CSS. Dalam proses pembangkitan form dapat diberikan aturan tambahan menggunakan DynaData dan XML instance untuk menyediakan data nilai awal dari form [2].

Penelitian berikutnya adalah yang dilakukan oleh Atia Mahmud Albhah yang membahas tentang bagaimana membangun kerangka aturan dalam pembangkitan form, dalam penelitian ini dibahas tentang aturan-aturan yang digunakan dalam pembangkitan form berdasarkan metada dari skema basis data. Skema basis data diterjemahkan ke dalam format XML sebagai pusat aturan pembangkitan antarmuka form atau yang disebut dengan RuleML, berdasarkan aturan yang di tulis ke dalam format RuleML maka dapat dibangun antarmuka form dengan tambahan JavaScript dan CSS serta dapat dikembalikan ke dalam bentuk skema basis data [3]. Selanjutnya adalah penelitian yang dilakukan oleh Mohammed A. Mgheder dan M. J. Ridley yang membahas tentang generasi antarmuka berbasis web menggunakan metadata basis data yang ditulis dalam bahasa pemrograman PHP, penelitian ini menggunakan metadata basis data sebagai dasar dalam proses generasi antarmuka yang difokuskan dalam generasi formulir web [4].

Selain penelitian banyak juga bermunculan aplikasiaplikasi yang dibangun untuk membangkitkan formulir berbasis web, salah satunya adalah GII Generator yang dikeluarkan oleh YII Framework khusus untuk bahasa pemrograman PHP yang disertai pembangkit antarmuka CRUD yang disertai JavaScript dan CSS di dalamnya. Untuk melakukan pembangkitan antarmuka CRUD aplikasi ini harus melakukan pembangkitan Model dalam konsep Model View Controller (MVC) sebagai representasi basis data khususnya tabel dalam bentuk kode program, dimana aturan untuk pembangkitan form diletakan di dalam model tersebut [5]. Aplikasi pembangkit lainnya adalah ScriptCase yang merupakan alat RAD (Rapid Application Development) yang menghasilkan aplikasi PHP berbasis tabel basis data lengkap dengan query, filter, grafik, laporan PDF, forms, menu navigasi, export data ke XLS, DOC, XML dan CSV [6]. Pada hasil keluaran dari aplikasi ini termasuk juga antarmuka yang dibangkitkan berdasarkan tabel basis data dan proses pembangkitan antarmuka dapat dikontrol dengan memberikan aturan tambahan.

Kekurangan dari dua penelitian yang dilakukan oleh Rein Raudjärv dan Atia Mahmud Albhah adalah, untuk melakukan pembangkitan form, skema basis data haruslah dituliskan ke dalam format XML terlebih dahulu dan aturan tambahan untuk proses pembangkitan diletakan di dalam dokumen tersebut. Hal ini dianggap memperpanjang proses yang dilalui untuk melakukan pembangkitan. Sedangkan pada aplikasi GII Generator, proses pembangkitan model juga harus dilakukan sebelum melakukan pembangkitan antarmuka CRUD sedangkan kema basis data seharusnya dapat secara langsung menjadi masukan dalam proses

pembangkitan antarmuka. Jika dilihat dari aplikasi ScriptCase kekurangan yang ditemukan adalah pada hasil antarmuka yang dihasilkan salah satu diantaranya adalah tidak mampu secara otomatis menentukan field yang akan ditampilkan pada antarmuka terkait field yang merupakan foreign key. Selain itu, penelitianpenelitian sebelumnya dan aplikasi pembangkit formulir diatas tidak mampu mengenali relasi hubungan antar data dari basis data secara otomatis.

Berdasarkan uraian di atas, maka penulis ingin merancang sebuah aplikasi yang mampu membangkitkan elemen formulir berbasis web berdasarkan skema basis data secara langsung yang mampu mengenali hubungan antar data, tanpa harus melakukan pembangkitan dokumen lain untuk menjadi inputan dalam proses pembangkitan, serta aturan tambahan yang dibutuhkan untuk mengontrol proses pembangkitan diletakan didalam skema basis data tersebut.

LANDASAN TEORI

A. Basis Data

Menurut Thomas Connolly dan Carolyn Begg dalam bukunya menjelaskan bahwa basis data merupakan kumpulan data yang terhubung secara logis, yang digunakan secara bersama-sama dan deskripsi dari data tersebut yang dirancang untuk memenuhi kebutuhan informasi sebuah organisasi [2]. Model relasional atau Relational Database Management System (RDBMS) diusulkan oleh E. F. Codd pada tahun 1970. Pada model relasional ini seluruh data terstruktur dengan relasi (table) secara logis. Setiap relasi memiliki sebuah nama dan terdiri dari atribut (kolom) data. Setiap tuple (baris) memiliki satu nilai per-atribut [2]

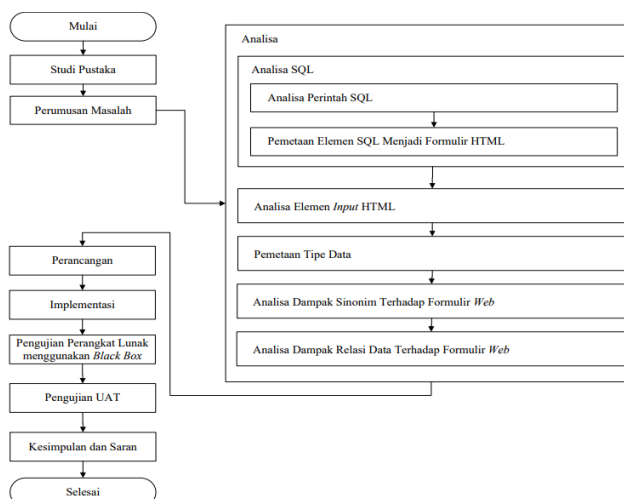
Sejarah dari model relasional (dan secara tidak langsung SQL) dimulai dengan publikasi E. F. Codd pada tahun 1970 ketika bekerja pada laboratorium IBM di San José. Pada tahun 1974, D. Chamberlin yang juga dari laboratorium IBM di San José mendefinisikan sebuah bahasa yang disebut Structured English Query Language (SEQUEL). Kemudian versi revisi SEQUEL/2 berikutnya dikeluarkan pada tahun 1976, tetapi dengan nama yang dirubah menjadi SQL karena alasan hukum. SQL (Structured Query Language) merupakan bahasa basis data yang memungkinkan pengguna untuk membuat basis data dan struktur relasional, melakukan tugas dasar manajemen manajemen data, seperti insertion, modification dan deletion data dari relasi serta melakukan query sederhana maupun query yang kompleks [2].

B. Teknologi Antarmuka Berbasis Web

Teknologi antarmuka berbasis web merupakan teknologi yang umum digunakan dalam membangun sebuah antarmuka berbasis web. Hypertext Markup Language (HTML) merupakan bahasa untuk menggambarkan struktur halaman web. Dengan HTML struktur halaman web digambarkan dengan menggunakan markup [7]. Versi HTML terakhir adalah HTML 5. HTML 5 merupakan versi lanjutan dari HTML 4 dan XHTML, pada versi ini HTML mendukung Mathematical Markup Language (MathML) dan Scalable Vector Graphics (SVG), serta penambahan elemen-elemen baru [7].

METODOLOGI PENELITIAN

Metodologi penelitian merupakan pedoman dalam melakukan penelitian. Di dalam metodologi penelitian terdapat langkah-langkah penelitian yang sudah ditentukan sebelumnya agar penelitian tidak menyimpang dari tujuan. Tahapan penelitian dapat dilihat pada gambar berikut.



Gambar 1 Metodologi Penelitian

ANALISA DAN PERANCANGAN

Berikut adalah beberapa tahapan analisis yang telah dilakukan.

A. Analisa SQL

Pada tahap ini dilakukan analisa terhadap RDBMS MariaDB untuk mengetahui elemen apa saja yang akan dilibatkan dalam pembangkitan formulir. Berdasarkan jenis perintah SQL dasar pada RDBMS MariaDB [8], jenis perintah SQL dasar tersebut dibagi ke dalam 3 jenis yaitu :

1. Database Definition Language (DDL)

DDL merupakan kumpulan perintah SQL yang digunakan dalam mendefinisikan bagaimana data disimpan dalam basis data. Contohnya CREATE DATABASE, DROP DATABASE, USE, CREATE TABLE, ALTER TABLE, DROP TABLE dan DESCRIBE.

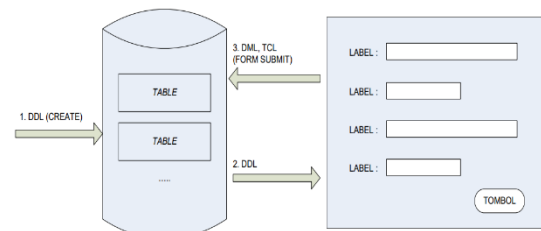
2. Database Manipulation Language (DML)

DML merupakan kumpulan perintah SQL yang digunakan untuk memanipulasi data. Adapun perintah SQL yang termasuk ke dalam kategori DML yaitu SELECT, INSERT, UPDATE, DELETE, REPLACE DAN TRUNCATE.

3. Transaction Control Statement (TCL)

TCL merupakan kumpulan perintah SQL yang digunakan untuk melakukan transaction. Adapun perintah SQL yang termasuk ke dalam kategori TCL yaitu START TRANSACTION, COMMIT dan ROLLBACK.

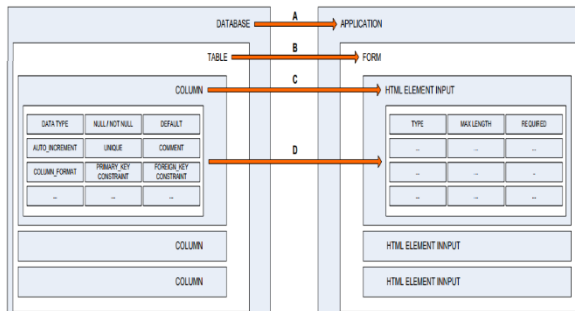
Berdasarkan jenis perintah SQL dasar diatas yang meliputi DDL, DML dan TCL dapat diambil kesimpulan perintah-perintah SQL dibagi kedalam 3 kategori tersebut untuk mengelompokan perintah SQL berdasarkan fungsinya. Dalam konteks fungsi dalam manajemen basis data, konsep CRUD (Create, Read, Update dan Delete) digunakan untuk menggambarkan fungsi dari tempat penyimpanan [9]. Kemudian Berdasarkan fase siklus pengembangan perangkat lunak tradisional atau Software Development Life Cycle (SDLC) menurut Peter Rob dan Carlos Coronel (2009, hal. 377) pada awal fase implementasi, desain basis data sudah diimplementasikan ke dalam Database Management Systems (DBMS) dan pada fase inilah siklus pembuatan kode program dan pengujian akan terjadi. Berdasarkan kutipan diatas maka akan digambarkan hubungan jenis perintah SQL dasar dengan formulir web untuk mengetahui kapan jenis perintah tersebut digunakan serta perannya dalam pembangkitan formulir. Gambar dibawah ini adalah alur perintah SQL dasar dalam proses olah data formulir web.



Gambar 2 Alur perintah SQL dasar dalam proses olah data formulir web

Didapatkan elemen-elemen SQL yang digunakan dalam perintah CREATE DATABASE dan CREATE TABLE, berdasarkan spesifikasi perintah SQL tersebut maka perintah-perintah

tersebut akan dipetakan dengan formulir HTML seperti gambar berikut.



Gambar 3 Pemetaan Elemen SQL Menjadi Formulir HTML

Pada gambar terlihat pemetaan elemen-elemen SQL menjadi formulir HTML sehingga didapatkan padanan elemen SQL pada formulir HTML. Berikut penjelasan gambar diatas :

1. Database dipetakan sebagai Application dalam artian satu aplikasi memiliki satu skema basis data.
2. Table dipetakan sebagai Form HTML dalam artian satu formulir merepresentasikan satu tabel.
3. Column dipetakan sebagai HTML Element Input dalam artian setiap elemen masukan HTML merepresentasikan satu column data.
4. Atribut pada Column direpresentasikan sebagai atribut pada HTML Element Input.

Dapat diambil kesimpulan bahwa dalam lingkup penelitian yang difokuskan dalam membangkitkan formulir maka objek utama yang dijadikan acuan dalam membangkitkan formulir adalah table dan column beserta atributnya. Maka akan dipetakan table dan column beserta atributnya menjadi elemen formulir HTML seperti pada tabel berikut.

Tabel 1. Pemetaan Tabel Menjadi Elemen Formulir HTML

No	Elemen SQL	Elemen Formulir HTML
1	Table	Tag <form>
2	Table name	Atribut "name" dan isi tag <label>
3	CONSTRAINT PRIMARY KEY (INDEX)	-
4	CONSTRAINT UNIQUE (INDEX)	-
5	CONSTRAINT FOREIGN KEY (INDEX)	HTML Element Input <select>
6	ENGINE	-
7	CHARACTER SET	-

No	Elemen SQL	Elemen Formulir HTML
8	CHECKSUM	-
9	COLLATE	-
10	COMMENT	-

Tabel 2. Pemetaan Column Menjadi Elemen Formulir HTML

No	Elemen SQL	Elemen Formulir HTML
1	Column	HTML Element Input <input> atau <textarea> (dibutuhkan analisa lebih lanjut tentang analisa elemen input HTML)
2	Column_name	Atribut name
3	Data_type	HTML Element Input Type (dibutuhkan analisa lebih lanjut tentang analisa elemen input HTML)
4	NULL atau NOT NULL	Atribut required
5	DEFAULT	Atribut value
6	AUTO INCREMENT	-
7	UNIQUE KEY	-
8	COMMENT	-
9	COLUMN FORMAT	-

Tabel 3. Pemetaan Tipe Data Terhadap Elemen Input Formulir Web

No	Tipe Data	Elemen HTML
1	TINYINT	Input (type=number)
2	BOOLEAN	Input (type=checkbox)
3	SMALLINT	Input (type=number)
4	MEDIUMINT	Input (type=number)
5	INT	Input (type=number)
6	INTEGER	Input (type=number)
7	BIGINT	Input (type=number)
8	DECIMAL	Input (type=number)
9	FLOAT	Input (type=number)
10	DOUBLE	Input (type=number)
11	DOUBLE PRECISION	Input (type=number)
12	BIT	Input (type=number)
13	CHAR	<= 78 karakter Input (type=text) > 78 karakter textarea
14	VARCHAR	<= 78 karakter Input (type=text) > 78 karakter textarea
15	BINARY	<= 78 karakter Input (type=text) > 78 karakter textarea
16	CHAR BYTE	<= 78 karakter Input (type=text) > 78 karakter textarea

No	Tipe Data	Elemen HTML
17	VARBINARY	<= 78 karakter Input (type=text) > 78 karakter textarea
18	TINYBLOB	Input (type=file)
19	BLOB	Input (type=file)
20	MEDIUMBLOB	Input (type=file)
21	LONGBLOB	Input (type=file)
22	TINYTEXT	<= 78 karakter Input (type=text) > 78 karakter textarea
23	TEXT	<= 78 karakter Input (type=text) > 78 karakter textarea
24	LONGTEXT	<= 78 karakter Input (type=text) > 78 karakter textarea
25	ENUM	Input (type=radio)
26	DATE	Input (type=date)
27	TIME	Input (type=time)
28	DATETIME	Input (type=datetime)
29	TIMESTAMP	-
30	YEAR	Input (type=number)
31	POINT	-
32	LINESTRING	-
33	POLYGON	-
34	MULTIPOINT	-
35	MULTILINESTRING	-
36	MULTIPOLYGON	-
37	GEOMETRYCOLLECTION	-
38	GEOMETRY	-

Berdasarkan tabel, tidak seluruhnya tipe data yang ada pada RDBMS MariaDB dapat dipetakan menjadi elemen formulir HTML, hal ini dikarenakan tidak ditemukannya elemen formulir HTML yang memenuhi spesifikasi untuk tipe data tersebut. Berdasarkan kondisi tersebut, untuk tipe data yang tidak memiliki pemetaan elemen HTML akan diperlakukan sebagai elemen input HTML dasar yaitu input (type=text) sebagai elemen penggantinya. Untuk tipe data TIMESTAMP tidak memiliki rancangan formulir, karena tipe data ini akan diberi nilai secara otomatis ketika perintah SQL Insert dan atau Update dieksekusi.

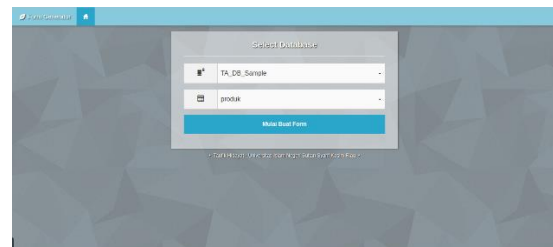
Ditemukan adanya sinonim atau alias merupakan istilah yang menyatakan tipe data tersebut merupakan nama lain dari suatu tipe data. Adapun sinonim atau alias yang digunakan pada tipe data RDBMS MariaDB adalah sebagai berikut.

1. Boolean merupakan sinonim atau alias dari TINYINT(1) yang dapat diartikan bahwa tipe data boolean yang digunakan pada saat mendefinisikan tipe data pada kolom tabel akan dijadikan sebagai tipe TINYINT(1). Berdasarkan permasalahan tersebut maka penentuan jenis input data yang akan digunakan untuk tipe data

- TINYINT(1) tersebut dikembalikan kepada UI Programmer untuk menentukannya.
2. INTEGER merupakan alias dari tipe data INT pada RDBMS MariaDB. INTEGER merupakan kepanjangan dari INT yang digunakan sebagai salah satu tipe data numerik yang disediakan pada RDBMS MariaDB
3. DOUBLE PRECISION merupakan tipe data yang dialiaskan kepada tipe data DOUBLE pada RDBMS MariaDB, yang merupakan tipe data untuk menampung data numerik dengan bilangan berkoma
4. CHAR BYTE merupakan alias dari tipe data BINARY pada RDBMS MariaDB, tipe data ini merupakan tipe data yang menampung data karakter string yang disimpan dalam bentuk biner

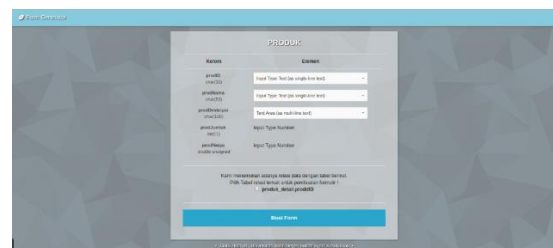
IMPLEMENTASI DAN PENGUJIAN

Tahapan implementasi aplikasi merupakan tahapan pembuatan kode program secara teknis berdasarkan perancangan prototype aplikasi. Adapun antarmuka dari hasil implementasi prototype aplikasi seperti pada gambar berikut. Gambar 4 memperlihatkan halaman pilih basis data dan tabel pada aplikasi pembangkit formulir, pada halaman ini pengguna diminta untuk memilih basis data dan tabel yang akan dibangkitkan formulirnya.



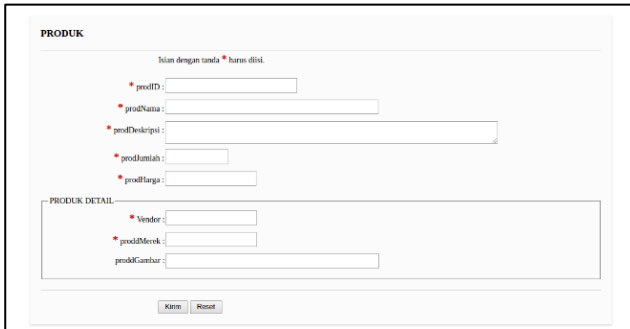
Gambar 4 Halaman pilih basis data dan tabel

Gambar 5 memperlihatkan halaman konfigurasi sebelum dilakukan pembangkitan formulir. Pada halaman ini pengguna dapat memilih konfigurasi dan relasi yang akan dilibatkan dalam pembangkitan formulir.



Gambar 5 Halaman konfigurasi dan relasi formulir

Gambar 6 memperlihatkan contoh gambar hasil formulir yang dibangkitkan dengan menggunakan aplikasi pembangkit formulir. Setiap elemen yang terdapat pada formulir dapat berubah berdasarkan spesifikasi tabel basis data dan konfigurasi serta relasi yang dipilih oleh pengguna.



Gambar 6 Halaman pembangkitan formulir

KESIMPULAN

Setelah melalui tahap analisa sampai dengan pengujian pada pembangkit formulir berbasis web berdasarkan metadata SQL, maka dapat diambil kesimpulan sebagai berikut.

1. Pemetaan tipe data menjadi elemen input formulir HTML berhasil dibuat.
2. Aplikasi pembangkit formulir berbasis web berdasarkan metadata SQL berhasil dirancang sesuai dengan hasil analisa dan perancangan.
3. Berdasarkan hasil pengujian Black Box disimpulkan bahwa aplikasi mampu membangun formulir berdasarkan spesifikasi SQL kedalam bentuk formulir sesuai dengan spesifikasi W3C.

REFERENSI

- [1] Bakar, M. A., Ismail, S., Idris, S., & Shukur, Z. (2015). seMeja API Design Based on CRUD+N Concept. Selangor, Malaysia.
- [2] Connolly, T., & Begg, C. (2005). Database System : A Practical Approach to Design, Implementation, and Management, Fourth Edition. United States: AddisonWesley.
- [3] RaudjÄrv, R. (2010). Dynamic Schema-Based Web Forms Generation in Java.
- [4] A. M. Albhbah, P. (2013). Dynamic Web Forms Development Using RuleML.
- [5] Winesett, J. (2012). Web Application Development with Yii and PHP Second

Edition. Birmingham - Mumbai: PACKT Publishing.

- [6] ScriptCase. (2016, April 04). FAQ. Retrieved from ScriptCase: <http://www.scriptcase.net/scriptcase-faq/>
- [7] W3C. (2016, April 04). HTML5. Retrieved from W3: <https://www.w3.org/TR/html5/>
- [8] MariaDB. (2017, 01 08). Basic SQL Statements. Retrieved from MariaDB: <https://mariadb.com/kb/en/mariadb/basic-sqlstatements/>
- [9] Bakar, M. A., Ismail, S., Idris, S., & Shukur, Z. (2015). seMeja API Design Based on CRUD+N Concept. Selangor, Malaysia.
- [10] ISO. (2015). Information technology - Database. INTERNATIONAL STANDARD ISO/IEC 9075-1 Fourth edition, 15.