

Generator Aplikasi Web dengan *GUI builder* dan Autogenerate Database untuk Mempercepat Proses Pengembangan

Fadilah Fahrul Hardiansyah^{a1}, Ahmad Syauqi Ahsan^{a2}, Faizal Amiruddin^{a3}, Mohammad Hasan Muktasyim Billah^{a4}

^aD4 Teknik informatika, Departemen Teknik Informatika dan Komputer, Politeknik Elektronika Negeri Surabaya, Jalan Raya ITS, Keputih, Kec. Sukolilo, Kota Surabaya, Jawa Timur 60111

¹ fahrul@pens.ac.id

² syauqi@pens.ac.id

³ f.a.faizal.amiruddin@gmail.com

⁴ moh.hasan.muktasyim.billah@gmail.com

Abstrak

Waktu pengerjaan (*deadline*) yang singkat dari proyek pembuatan aplikasi merupakan sebuah tantangan yang besar bagi para penyedia jasa pembuatan aplikasi. Dari waktu pengerjaan yang singkat ini, banyak masalah yang dapat timbul, dimulai dari proyek aplikasi yang tidak dapat selesai sempurna dan keterlambatan selesainya proyek aplikasi. Untuk mengatasi masalah-masalah tersebut penulis memberikan solusi dengan mengembangkan suatu Laravel package untuk mendesain dan generate aplikasi siap pakai. Package ini menyediakan *GUI builder* untuk mendesain aplikasi sesuai keinginan dengan melakukan *drag and drop* komponen-komponen HTML untuk dilakukan generate menjadi file-file *source code* dengan memanfaatkan Laravel Blade sebagai *template engine* untuk melakukan *render* template-template komponen HTML berdasarkan apa yang telah didesain pada *GUI builder*. Dalam ujicobanya menggunakan kasus operasi CRUD (*Create, Read, Update, and Delete*) sederhana, file-file *source code* hasil dari generate dapat berjalan dengan baik tanpa harus melakukan modifikasi pada file-file *source code* tersebut sebelum dijalankan.

Kata kunci: Framework, Laravel, *Software Design Pattern*.

Web Application Generator with GUI builder and Auto-Generated Database to Speed up the Development Process

Abstract

A short deadline for an application development project is a big challenge for programmers and application creation companies such as software houses. From this short processing time, many problems can arise, starting from application projects that cannot be completed perfectly, the delay in completing the application project, until programmers who ignore the rules of good code structure to pursue the target as soon as possible. To solve these problems, we provide solutions by developing a Laravel package to design and generate ready-to-use applications. This package provides a GUI builder for designing applications as desired by dragging and dropping HTML components to be generated into source code files. In generating source code files, this package utilizes the Laravel framework feature, namely Laravel Blade as a template engine for rendering HTML component templates based on what has been designed on the GUI builder. In the test using simple CRUD (*Create, Read, Update, and Delete*) operations, the generated source code files generated can run well without having to modify the source code files before they are run.

Keywords: Framework, Laravel, *Software Design Pattern*.

I. PENDAHULUAN

Perkembangan industri pengembangan aplikasi di Indonesia saat ini sedang tumbuh dengan pesat. Pesatnya perkembangan teknologi perangkat cerdas dan kemudahan akses internet sangat mendukung perkembangan industri pengembangan aplikasi. Kecepatan perkembangan industri pengembangan aplikasi ini ditandai dengan beberapa

perusahaan rintisan di bidang pengembangan aplikasi telah mendapatkan predikat sebagai “unicorn” yaitu perusahaan rintisan yang memiliki valuasi lebih besar dari 1 milyar Dolar Amerika. Perusahaan-perusahaan rintisan tersebut adalah Gojek, Traveloka, Tokopedia, dan Bukalapak. Untuk mendukung industri yang strategis ini, kemudahan dan kecepatan pengembangan aplikasi sangat dibutuhkan agar selalu dapat memenuhi kebutuhan pengguna.

Teknologi-teknologi yang digunakan dalam pengembangan aplikasi haruslah selalu mutakhir agar dapat selalu mengikuti perkembangan teknologi yang sangat pesat. Selain itu, kualitas kode harus sangat diperhatikan untuk menjaga agar aplikasi yang dikembangkan dapat selalu diperbaharui.

Kerap kali, untuk dapat segera memenuhi kebutuhan pengguna, perusahaan sering sekali memberikan *deadline* pengerjaan aplikasi dalam waktu yang sangat singkat. Hal ini berdampak besar dalam pengembangan aplikasi. Singkatnya waktu pengerjaan mengakibatkan tidak selesainya proyek aplikasi yang dikerjakan karena proses pengerjaan yang lama atau aplikasi selesai dengan tidak sempurna karena proses pengerjaan dilakukan secara terburu-buru. Pada akhirnya akibat dari keterlambatan selesainya proyek aplikasi, para penyedia jasa aplikasi sering harus membayar denda dari keterlambatan tersebut. Singkatnya permintaan waktu pengerjaan juga berimbas pada cara *programmer* dalam mengerjakan proyek aplikasi. Sering kali, dalam pengerjaannya *programmer* akan mengutamakan untuk mengejar target selesai secepat mungkin sehingga akan mengesampingkan kaidah-kaidah struktur kode yang baik.

Penelitian ini bertujuan untuk memberikan solusi baru untuk memudahkan para penyedia jasa pembuatan aplikasi dalam mempercepat penyelesaian proyek pembuatan aplikasi dengan pembuatan generator aplikasi web dengan *GUI builder* dan *auto-generated* database sebagai sebuah Laravel package. Hasil generate merupakan suatu aplikasi untuk menjalankan operasi CRUD (*Create, Read, Update, dan Delete*) untuk tabel-tabel yang tidak berelasi pada database. *Source code* hasil generate menerapkan fitur-fitur framework Laravel, *best practice, software design pattern* dan *clean code* pada hasil generate sehingga pembuatan aplikasi menjadi cepat dan menghasilkan struktur kode yang ringkas, mudah dipahami, dan mudah dikembangkan.

II. METODOLOGI

A. Metode Penelitian

Dalam penelitian ini dilakukan beberapa tahapan yaitu:

1) *Studi Literatur*: Pada tahapan ini dilakukan studi literatur terhadap konsep aplikasi yang akan dikembangkan. Penulis membaca literatur/paper/refrensi terkait untuk kemudian dijadikan pedoman dalam pembuatan package Laravel untuk website generator dan *auto-generated* database. Pada tahap ini juga dilakukan analisis komparasi produk yang serupa.

2) *Perancangan Sistem*: Pada tahapan ini, penulis menyusun rancangan desain sistem yang akan dikembangkan. Desain sistem yang dibuat didasarkan pada hasil studi literatur yang dilakukan sebelumnya. Perancangan sistem yang dibuat meliputi rencana alur jalannya aplikasi mulai dari *input* diberikan hingga mendapatkan output yang diinginkan.

3) *Pembuatan Kode Program*: Dalam tahap ini, penulis melakukan pembuatan kode program berdasarkan dari hasil rancangan sistem yang telah dibuat. Kode dibuat

dengan menggunakan PHP dengan menerapkan kaidah penulisan kode yang baik.

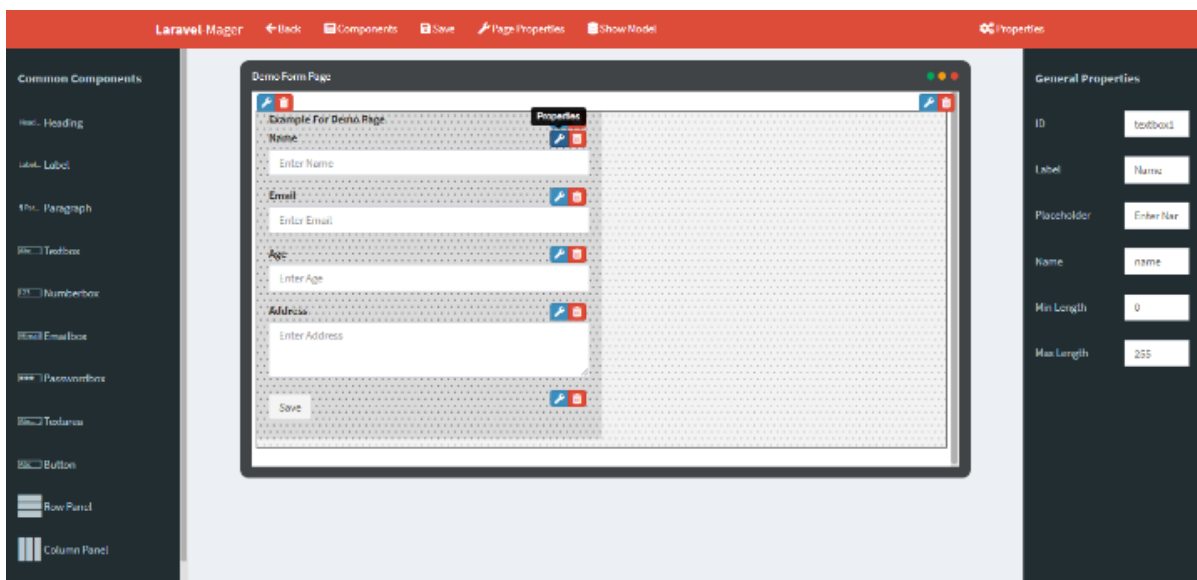
4) *Pengujian Sistem*: Pada tahap ini dilakukan pengujian untuk memastikan package bekerja dengan baik dan menghasilkan kualitas kode yang baik pula. Selain itu untuk mengetahui efektifitas dari package, dilakukan analisa langkah proses penggunaan dibandingkan dengan penggunaan package lain serta pengerjaan manual. Pengumpulan ulasan pengguna juga dilakukan untuk mengetahui apakah package yang dibuat dapat diterima dan digunakan dengan baik oleh pengguna.

B. Penelitian Terkait

Penerapan teknologi generator untuk dapat membuat aplikasi secara instan telah banyak digunakan pada berbagai macam bahasa pemrograman. Pada framework Laravel sendiri, teknologi generator telah diterapkan oleh banyak orang untuk dibuat sebagai package, contoh yang memiliki banyak fitur generate diantaranya adalah Infyom Laravel Generator, Craftable, Quick Admin Panel dan CRUD Booster. Selain itu, Yii Framework bahkan memiliki fitur generate bawaan bernama Gii. Infyom Laravel Generator menyediakan banyak fitur penting untuk melakukan generate suatu aplikasi secara lengkap, diantaranya adalah CRUD generator, test case generator, API generator lengkap dengan SwaggerUI documentation, dan database generator. Craftable memiliki fitur yang lebih sedikit dari Infyom Laravel Generator dengan mengedepankan fitur manajemen autentikasi pengguna dan media library untuk manajemen file-file yang diunggah pengguna. Dari empat generator Laravel yang dibahas penulis, Quick Admin Panel merupakan generator yang paling berbeda, apabila yang lain cukup mengakses generator secara lokal pada komputer penggunanya, Quick Admin Panel menyediakan sebuah halaman web yang dapat diakses melalui internet untuk menggunakan fitur-fitur generatornya. Quick Admin Panel ini memiliki fitur untuk menyusun rancangan database melalui form-form yang kemudian dari rancangan database tersebut akan dilakukan generate CRUD yang sesuai. CRUDBooster memiliki fitur-fitur yang banyak, paling banyak diantara keempat generator Laravel yang dibahas. CRUDBooster memiliki fitur-fitur untuk generate CRUD, API, validasi, eksport data, dan banyak fitur lain untuk melakukan generate. Infyom Laravel Generator dan Craftable menyediakan aplikasi *console (command line)* sebagai fitur utama untuk melakukan generate file-file kode. Quick Admin Panel, CRUD Booster dan Gii menerapkan tampilan web dengan menggunakan *input* form untuk memasukkan berbagai pengaturan untuk melakukan generate file-file kode sesuai keinginan.

Perbandingan antara sistem yang dikembangkan, berupa Laravel package untuk mendesain dan generate aplikasi siap pakai, dengan produk-produk yang terkait digunakan untuk memastikan keunikan dari sistem tersebut apabila dibandingkan dengan produk-produk lain. Terdapat beberapa aspek fitur yang akan dibandingkan, yaitu:

1) *Form Generator*: Fitur untuk melakukan generate sebuah form untuk melakukan operasi CRUD (Create, Read, Update, dan Delete) pada database.



Gambar. 1. Tampilan GUI Builder.

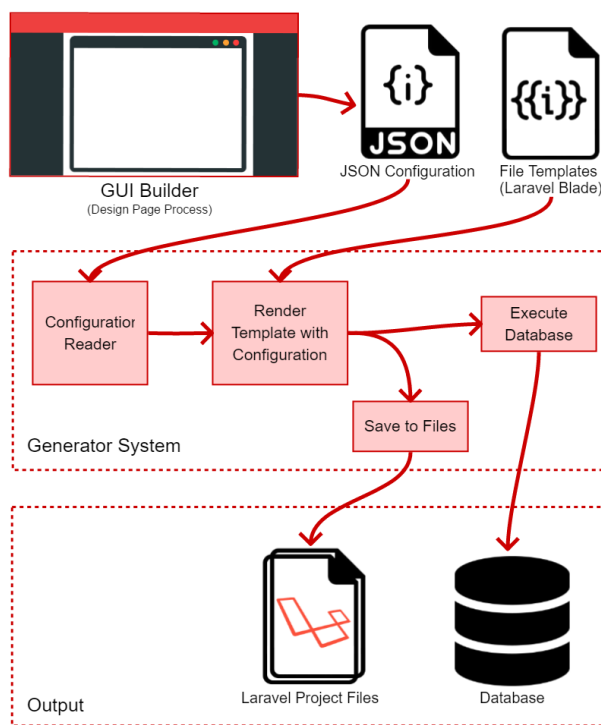
2) *Berbasis GUI*: Fitur dimana sistem menampilkan tampilan antarmuka grafis pada sebuah halaman web yang dapat ditampilkan pada browser untuk melakukan generate.

3) *Berbasis Command Line*: Fitur dimana sistem dapat dijalankan melalui aplikasi command line seperti terminal pada sistem operasi berbasis Linux dan Command Promt pada sistem operasi Windows untuk melakukan generate.

4) *GUI builder*: Fitur untuk membangun tampilan antarmuka dengan melakukan *drag and drop* komponen-komponen antarmuka ke dalam sebuah panel seperti yang biasa tersedia pada aplikasi IDE (*Integrated Development Environment*) seperti Netbeans dan Visual Studio.

5) *Form Generator*: Fitur untuk melakukan generate database secara otomatis sesuai dengan tampilan yang dibuat dari *GUI builder*.

Bagan desain sistem pada Gambar 1 menjelaskan tentang bagaimana package ini bekerja. Bagian-bagian dari sistem ini berupa *GUI builder* yang merupakan media untuk mendapatkan *input* dari pengguna, Generator System yang merupakan proses dari sistem, dan Output yang merupakan hasil keluaran dari sistem.



Gambar. 2. Desain Sistem.

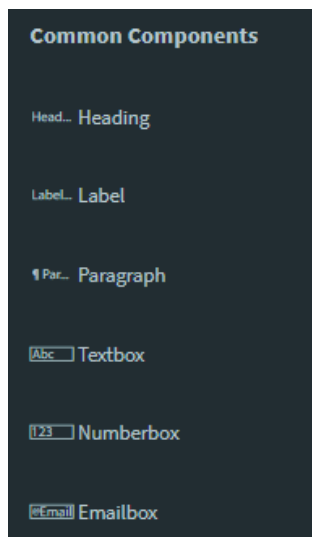
1) *GUI builder*: *GUI builder* digunakan untuk mendapatkan *input* berupa desain tampilan antarmuka web. *GUI builder* tersedia komponen-komponen HTML yang dapat di *drag and drop* untuk dapat disusun sesuai

TABEL I
PERBANDINGAN DENGAN SISTEM YANG DIAJUKAN.

Nama Produk	Form Generator	Berbasis GUI	Berbasis Command Line	GUI builder	Autogenerate Database
Infyom-Laravel Generator	✓	✓	✓	-	✓
Craftable	✓	-	✓	-	-
Quick Admin Panel	✓	✓	-	-	-
CRUDBooster	✓	✓	-	-	-
Yii2 Gii	✓	✓	-	-	-
Sistem yang diajukan	✓	✓	-	✓	✓

(✓) sudah diimplementasikan, (-) belum diimplementasikan

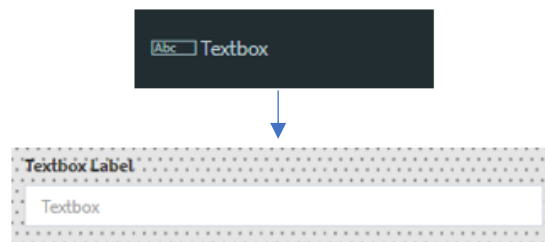
keinginan untuk menjadi suatu tampilan antarmuka web yang utuh. Komponen-komponen HTML yang tersedia untuk dilakukan *drag and drop*, berupa komponen-komponen HTML, baik untuk melakukan *input* ke dalam database maupun untuk menampilkan data dari database, seperti komponen *input* textbox, numberbox, emailbox, textarea, select, button, tabel, heading, paragraph, dan komponen-komponen lainnya. Untuk dapat melakukan generate database, jenis-jenis komponen tersebut menjadi acuan untuk menentukan tipe data yang dipakai. Namun untuk dapat menentukan tipe data, komponen yang bertipe *input* form yang hanya akan menjadi acuan dari tipe data. Sebagai contoh, komponen *input* textbox akan memiliki tipe data varchar dan komponen *input* numberbox akan memiliki tipe data integer. Tampilan *GUI builder* seperti terlihat pada Gambar 2. *GUI builder* memiliki tiga komponen utama untuk menyusun tampilan antarmuka web, yaitu panel Common Components, Drawing Area, dan panel General Properties. Pada penggunaannya, dalam menyusun tampilan antarmuka web, pengguna hanya perlu melakukan *drag and drop* komponen yang tersedia pada panel Common Components, yang ada pada sidebar sebelah kiri, ke Drawing Area, yang terletak pada bagian tengah antar muka, lalu mengisikasikan properties pada panel General Properties, yang ada pada sidebar sebelah kanan. Panel Common Components memuat komponen-komponen HTML yang tersedia untuk dapat dilakukan *drag and drop* dan disusun ke Drawing Area. Komponen ini ditampilkan dengan daftar *icon* dan nama komponen HTML untuk mewakili komponen-komponen HTML yang berkaitan. Panel Common Components adalah bagian yang seperti pada Gambar 3.



Gambar 3. Panel Common Components.

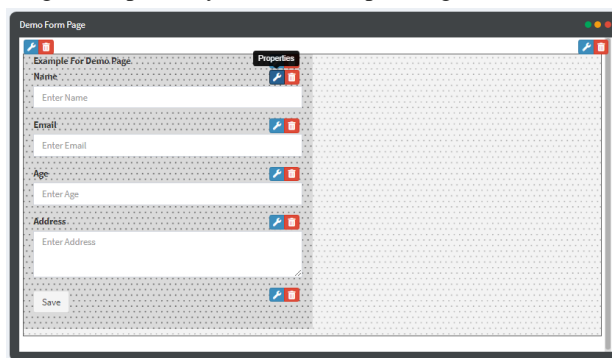
Gambar 4 menunjukkan tampilan komponen HTML *input* textbox pada panel Common Components dan ketika telah dilakukan *drag and drop* pada Drawing Area. Pada saat berada pada panel Common Components, komponen HTML *input* textbox akan hanya menampilkan *icon* dan kata “Textbox”, namun apabila telah dilakukan *drag and drop* pada Drawing Area, tampilan komponen ini akan

berubah menjadi tampilan textbox yang sesungguhnya serta dilengkapi dengan label.



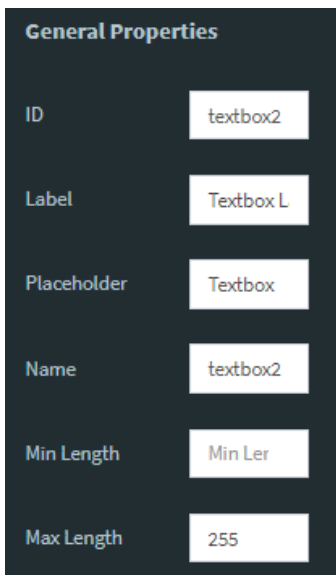
Gambar 4. Tampilan komponen textbox pada panel Common Components dan setelah dilakukan *drag and drop*.

Drawing Area seperti pada Gambar 5 digunakan untuk mengatur susunan penempatan tampilan komponen-komponen HTML sebagai acuan tampilan layout antarmuka halaman pada saat melakukan proses generasi. Pengguna dapat memindah dan menyusun komponen yang telah terdapat pada Drawing Area untuk menempatkannya sesuai keinginan. Drawing area akan menampilkan rancangan layout antarmuka halaman yang sama persis dengan tampilan layout hasil dari proses generasi.



Gambar 5. Drawing Area.

Panel General Properties pada Gambar 6 merupakan sebuah form yang digunakan untuk mengatur nilai-nilai dari atribut yang dimiliki oleh suatu komponen HTML.



Gambar. 6. Panel General Properties.

Misalkan pada textbox, komponen HTML ini memiliki atribut-atribut berupa id, label, placeholder, name, minlength, dan maxlength. Sebelumnya telah dijelaskan bahwa komponen yang bertipe *input* form yang hanya akan menjadi acuan dari tipe data saat proses generasi database. Selain itu, *input* name pada panel General Properties digunakan untuk menentukan nama kolom dari tabel pada database. Misalnya, pada suatu textbox, *input* name diisi dengan kata "nama_barang", maka nama kolom pada tabel di database juga akan memiliki nama "nama_barang". Komponen *GUI builder* ini akan menghasilkan file konfigurasi dengan tipe data JSON yang menyimpan konfigurasi yang telah diatur user pada tiap komponen yang dipakai. File konfigurasi yang dihasilkan tidak hanya satu melainkan beberapa file konfigurasi, setidaknya terdapat konfigurasi untuk *view* dan database. Contohnya, hasil konfigurasi sebuah textbox menghasilkan file-file JSON seperti berikut:

```
view.json
{
  "textbox1": {
    "type": "textbox",
    "id": "textbox-nama",
    "label": "Nama",
    "placeholder": "Masukkan Nama",
    "name": "nama_barang",
    "minlength": 0,
    "maxlength": 255
  }
}
```

```
database.json
{
  "nama_barang": {
    "type": "string",
    "length": 255,
    "label": "Nama",
    "input": "textbox"
  }
}
```

2) *Generator System*: Generator System merupakan komponen pemrosesan dari sistem ini. Generator System menjalankan proses-proses untuk mengolah data *input* dari *GUI builder* untuk mejadi output. Terdapat empat peroses yang berjalan pada generator system, yaitu Configuration Reader, Render Template with Configuration, Save to File, dan Execute Database. Configuration Reader adalah komponen untuk membaca file JSON Configuration untuk diterjemahkan ke dalam object-object atau array-array pada PHP. Proses 'Render Template with Configuration' dijalankan menggunakan fitur bawaan framework Laravel, yaitu Laravel Blade. Dalam penggunaan normal dari Laravel Blade ini adalah untuk me-render file *view* yang secara umum biasanya menggunakan tag-tag HTML menjadi halaman web yang dapat ditampilkan pada browser. Kemampuan Laravel Blade untuk me-render *view* dapat dimanfaatkan untuk melakukan generate, dengan melakukan *render* file template bertipe Laravel Blade yang digabungkan dengan hasil pembacaan file konfigurasi JSON. Hasil render template dengan konfigurasi akan menghasilkan variabel bertipe string. Dalam proses 'Save to File', variabel tersebut akan disimpan ke dalam file berdasarkan jenis file sesuai dengan konfigurasi, misalkan konfigurasi yang dilakukan render adalah konfigurasi controller, maka file akan disimpan ke dalam direktori controller framework Laravel dan dengan tipe file PHP. Dalam pengembangannya, proses 'Save to File' menggunakan komponen bawaan dari framework Laravel yaitu library facade File. Contoh proses dalam melakukan generate suatu textbox yang dibuat pada *GUI builder* adalah sebagai berikut:

- Tersedia file JSON untuk sebuah textbox.

```
view.json
{
  "textbox1": {
    "type": "textbox",
    "id": "textbox-nama",
    "label": "Nama",
    "placeholder": "Masukkan Nama",
    "name": "nama_barang",
    "minlength": 0,
    "maxlength": 255
  }
}
```

- Tersedia file template untuk textbox dalam bentuk Laravel Blade.

```
textbox.blade.php
<div class="form-group">
  <label>{{ $label }}</label>
  <input type="text" class="form-control"
    id="{{ $id }}" name="{{ $name }}"
    placeholder="{{ $placeholder }}"
    minlength="{{ $minlength }}"
    maxlength="{{ $maxlength }}">
</div>
```

- Menghasilkan satu file *view* pada Laravel untuk komponen textbox.

```
hasil
<div class="form-group">
  <label> Nama </label>
  <input type="text" class="form-control"
    id=" textbox-nama "
    name="nama_barang"
    placeholder="Masukkan Nama"
    minlength="0"
    maxlength="255">
</div>
```

Karena salah satu hasil dari *GUI builder* adalah file konfigurasi JSON untuk database, komponen ini juga menggunakan beberapa fitur PHP dan Laravel, yaitu PHP PDO, Laravel Migration, Laravel Artisan, dan Laravel Elequent untuk melakukan generate komponen-komponen database. Contoh dalam melakukan generate suatu kolom database dari textbox yang seperti pada contoh sebelumnya adalah sebagai berikut:

- Tersedia file JSON untuk sebuah kolom.

```
database.json
{
  "nama_barang": {
    "type": "string",
    "length": 255,
    "label": "Nama",
    "input": "textbox"
  }
}
```

- Tersedia file template untuk migration.

```
migration.blade.php
Schema::create('nama_tabel', function
(Blueprint $table) {
  $table->increments('id');
  $table->{{ $type }}('{{ $name }}');
  $table->timestamps();
});
```

- Hasil file migration yang akan dijalankan sehingga menjadi kolom-kolom pada database.

```
migration.php
Schema::create('nama_tabel', function
(Blueprint $table) {
  $table->increments('id');
  $table->string('nama_barang');
  $table->timestamps();
});
```

III. HASIL DAN PEMBAHASAN

A. Perbandingan Langkah Langkah Penggunaan

Berikut adalah perbandingan langkah-langkah antara pengerjaan proyek aplikasi dengan mengetik secara manual, pengerjaan proyek aplikasi menggunakan generate dengan sistem yang diajukan, dan pengerjaan proyek aplikasi menggunakan generate dengan package lain bernama

CRUDBooster untuk melihat seberapa lama masing-masing waktu pengerjaannya.

Aplikasi yang dikerjakan secara manual, mengadopsi dari laman tutorial pemrograman It Solution Stuff dengan url itsolutionstuff.com. Tutorial diambil dari tautan-tautan berikut:

- Laravel 5.7 CRUD (Create Read Update Delete) Tutorial Example dari <https://itsolutionstuff.com/post/laravel-57-crud-create-read-update-delete-tutorial-example-example.html>
- Laravel 5.7 - Create REST API with authentication using Passport Tutorial dari <https://itsolutionstuff.com/post/laravel-57-create-rest-api-with-authentication-using-passport-tutorial-example.html>

Tutorial-tutorial tersebut berisi tutorial pemrograman CRUD dasar menggunakan framework Laravel 5.7 dalam bentuk halaman web dan REST API.

Untuk membandingkan langkah-langkah dari ketiga cara tersebut, aplikasi yang dibuat akan menggunakan studi kasus berupa satu modul aplikasi untuk melakukan operasi CRUD data barang (product) dengan kolom berupa nama dan detail yang menyediakan tampilan web dengan tema AdminLTE dan tersedia pula REST API. Sehingga jumlah langkah-langkah dari pembuatan suatu aplikasi dari ketiga cara tersebut dapat dijabarkan dengan Tabel II berikut.

TABEL II
HASIL ANALISIS JUMLAH LANGKAH.

No.	Sistem	Jumlah Langkah
1.	Manual	24
2.	CRUDBooster	15
3.	Sistem yang Diajukan	17

Dari hasil perbandingan untuk ketiga cara dalam pembuatan sebuah modul CRUD aplikasi tersebut, jumlah langkah yang diperlukan dalam pengerjaan proyek aplikasi secara manual sesuai dengan tutorial akan lebih banyak dari pengerjaan proyek aplikasi dengan generate, baik menggunakan generator CRUDBooster maupun sistem yang diajukan.

Mayoritas langkah-langkah pengerjaan proyek secara manual adalah melakukan pengetikan kode. Pada dunia nyata tentu saja pengetikan kode akan jauh lebih memakan waktu dari pada mengisi form ataupun dengan *drag and drop* menggunakan *GUI builder*.

Dalam perbandingannya dengan CRUDBooster, sistem yang diajukan ini memiliki jumlah langkah yang hampir sama dalam pembuatan sebuah modul CRUD aplikasi, namun langkah-langkah yang dijalankan sangat berbeda. Penggunaan CRUDBooster hanya memerlukan proses pengisian form-form untuk menjadi suatu modul CRUD yang lengkap, namun hasil tampilan halaman yang dihasilkan oleh generator akan mengikuti tampilan yang disediakan oleh package CRUDBooster. Hasil tampilan dari generate CRUDBooster hanya dapat diubah dengan mengubah kode yang dihasilkan. Berbeda dengan menggunakan sistem yang diajukan, package ini terdapat

GUI builder yang disediakan untuk mendesain tampilan halaman sesuai keinginan.

B. Perbandingan Kualitas Kode

PHP Insights merupakan suatu library untuk menganalisa kode dari bahasa pemrograman PHP dan framework Laravel. Analisis dari PHP Insights ini dijalankan pada kode hasil pembuatan secara manual dan kode hasil generate, baik menggunakan CRUDBooster maupun sistem yang diajukan. Terdapat 4 aspek penilaian yang dinilai oleh PHP Insights, yaitu *code*, *complexity*, *architecture*, dan *style*.

Nilai *code* adalah nilai *clean code* atau seberapa bersih kode dari *code smell*, dimana *code smells* sendiri mengindikasikan adanya kelemahan struktur desain program yang dapat memperlambat development atau menambah resiko *bug* dan kegagalan di masa depan. Analisis nilai *code* ini dilakukan pada lingkup *comment*, *class*, *function*, dan *global variable*.

Nilai *complexity* merupakan tingkatan dimana sistem atau komponen memiliki desain atau implementasi yang sulit untuk dipahami dan diverifikasi. Berdasarkan nilai *complexity*, suatu software dapat dikategorikan sebagai sebuah software yang sederhana atau software yang kompleks.

Nilai *architecture* adalah seberapa baik penulisan stuktur kode sesuai kaidah pemrograman berbasis objek. Nilai *architecture* ini didapat dari analisis pada lingkup *class*, *interface*, *global variable*, dan *trait*.

Nilai *style* adalah seberapa baik kode dalam mentaati gaya penulisan kode yang baik, pada pemrograman PHP adalah gaya penulisan sesuai kaidah-kaidah PSR-2.

Selain itu, PHP Insights menyediakan skala baik tidaknya hasil analisis. Skala tersebut dibagi ke dalam tiga kategori, yaitu kategori buruk dengan nilai 1 sampai 49, kategori sedang dengan nilai 50 sampai 79, dan kategori baik dengan nilai 80 sampai 100.

TABEL III
HASIL ANALISIS KUALITAS KODE.

No.	Kategori	Manual	CRUDBooster	Sistem yang Diajukan
1.	Code	91.7%	93.1%	87.5%
2.	Complexity	92.6%	98.8%	95.0%
3.	Architecture	90.9%	86.4%	90.9%
4.	Style	86.7%	73.3%	86.7%

Dari hasil perbandingan ketiga cara untuk membuat aplikasi tersebut, didapatkan bahwa tidak selalu kode hasil generate akan lebih baik dari kode hasil melakukan penulisan secara manual. *Programmer* memiliki gaya masing-masing dalam menuliskan kode, sehingga dapat menghasilkan kode yang lebih baik, sama baiknya, atau lebih buruk dari hasil generate.

Di samping itu, hasil analisa menggunakan PHPInsights, didapatkan bahwa kode hasil generate menggunakan Laravel-Mager telah menghasilkan kode yang bagus dalam hal kualitas. Berbeda dengan CRUDBooster yang masih memiliki nilai dengan satu kategori sedang, yaitu pada nilai *style*.

C. Ulasan Pengguna dengan SUS

Ulasan pengguna digunakan untuk mengetahui apakah package Laravel-Mager dapat diterima dan digunakan dengan baik oleh pengguna. Ulasan pengguna ini didapatkan dengan membagikan kuesioner SUS (*System Usability Scale*). Kuesioner tersebut dibagikan melalui media sosial ke beberapa grup komunitas-komunitas *programmer*. Hasilnya mendapatkan 40 responden dengan latar belakang 52,5% pelajar atau mahasiswa dan 47,5% bekerja sebagai *programmer* atau dalam bidang IT.

TABEL IV
HASIL PERHITUNGAN SUS.

No.	Responden	Jumlah	Nilai (Jumlah x 2,5)
1	Responden 1	31	78
2	Responden 2	22	55
3	Responden 3	20	50
4	Responden 4	25	63
5	Responden 5	30	75
6	Responden 6	25	63
7	Responden 7	28	70
8	Responden 8	32	80
9	Responden 9	23	58
10	Responden 10	29	73
11	Responden 11	32	80
12	Responden 12	39	98
13	Responden 13	21	53
14	Responden 14	29	73
15	Responden 15	23	58
16	Responden 16	35	88
17	Responden 17	30	75
18	Responden 18	23	58
19	Responden 19	24	60
20	Responden 20	26	65
21	Responden 21	30	75
22	Responden 22	29	73
23	Responden 23	31	78
24	Responden 24	34	85
25	Responden 25	34	85
26	Responden 26	28	70
27	Responden 27	28	70
28	Responden 28	28	70
29	Responden 29	25	63
30	Responden 30	27	68
31	Responden 31	36	90
32	Responden 32	27	68
33	Responden 33	22	55

No.	Responden	Jumlah	Nilai (Jumlah x 2,5)
34	Responden 34	34	85
35	Responden 35	27	68
36	Responden 36	20	50
37	Responden 37	32	80
38	Responden 38	33	83
39	Responden 39	35	88
40	Responden 40	34	85
		Rata-Rata	71

Hasil rata-rata nilai yang didapat dari 40 responden tersebut adalah 71. Nilai 71 ini akan diproses menggunakan tabel penilaian SUS untuk menentukan seberapa baik dan menganalisis nilai SUS tersebut. Tabel penilaian SUS tersebut terdapat pada Tabel IV.

TABEL V
TABEL PENILAIAN SUS.

Grade	SUS	Percentile range	Adjective	Acceptable	NPS
A+	84.1 - 100	96-100	Best Imaginable	Acceptable	Promoter
A	80.8 - 84.0	90-95	Good	Acceptable	Promoter
A-	78.9 - 80.7	85-89		Acceptable	Promoter
B+	77.2 - 78.8	80-84		Acceptable	Promoter
B	74.1 - 77.1	70 - 79		Acceptable	Passive
B-	72.6 - 74.0	65 - 69		Acceptable	Passive
C+	71.1 - 72.5	60 - 64	OK	Acceptable	Passive
C	65.0 - 71.0	41 - 59		Marginal	Passive
C-	62.7 - 64.9	35 - 40		Marginal	Passive
D	51.7 - 62.6	15 - 34	OK	Marginal	Detractor
F	25.1 - 51.6	2- 14	Poor	Not Acceptable	Detractor
F	0-25	0-1.9	Worst Imaginable	Not Acceptable	Detractor

Sumber: <https://measuringu.com/interpret-sus-score>

Tabel V menunjukkan bahwa nilai 71 mendapatkan grade C dengan predikat good (baik). Nilai 71 tersebut mendapatkan skala *acceptable* dengan nilai *marginal* atau dapat diterima dan digunakan oleh pengguna. Juga dari

nilai *Net Promoter Score* (NPS) menunjukkan para responden rata-rata bersifat *passive* atau banyak yang ragu-ragu untuk mengisi.

IV. KESIMPULAN

Menggunakan generator untuk mengembangkan sebuah proyek aplikasi terbukti akan memudahkan proses pengerjaannya. Menggantikan proses penulisan kode dengan proses generate akan mempercepat proses selesainya pengerjaan proyek aplikasi. Dari hasil analisa kode menggunakan PHPInsights, terbukti akan menghasilkan kualitas kode yang bagus. Meskipun kualitas kode dari kode hasil generate menghasilkan kualitas yang bagus, tetapi apabila dibandingkan dengan kode hasil penulisan secara manual atau dari produk-produk generator yang lain, bisa jadi lebih baik, sama baiknya, atau lebih buruk kualitasnya. Selain itu, dari hasil penarikan kuesioner, package ini telah dapat diterima dan dapat digunakan dengan baik.

UCAPAN TERIMA KASIH

Penulis mengucapkan terima kasih kepada tim laboratorium Rekayasa Perangkat Lunak (RPL) Teknik Informatika PENS atas bantuannya untuk menjadi user dalam pengujian sistem, team teaching RPL Teknik Informatika PENS sebagai tim pakar yang telah memberikan masukan masukan dalam pengembangan sistem, serta tim pakar dari industri yang juga membantu memvalidasi dan menguji sistem. Tak lupa, kami ucapkan terima kasih kepada seluruh responden yang telah membantu memberikan penilaian pada sistem yang dikembangkan.

DAFTAR PUSTAKA

- [1] "PHP Standards Recommendations", PHP Framework Interopability Group, [Online]. Tersedia: www.php-fig.org/psr. [Diakses pada Juli 2018].
- [2] R. C. Martin, "The Principles of OOD", [Online]. Tersedia: butunclebob.com/ArticleS.UncleBob.PrinciplesOfOod. [Diakses pada Juli 2018].
- [3] A. Shvets, Design Patterns Explained Simply, Source Making, 2017.
- [4] "Package Development", Laravel, [Online]. Tersedia: <http://laravel.com/docs/5.7/packages>. [Diakses pada Januari 2019].
- [5] "Laravel Documentation", Laravel, [Online]. Tersedia: <http://laravel.com/docs/5.7>. [Diakses pada Januari 2019].
- [6] "Laravel Generator - Get your APIs and Admin Panel ready in minutes", InfyOmLabs, [Online]. Tersedia: <http://labs.infyom.com/laravelgenerator/>. [Diakses pada Mei 2018].
- [7] "Build administration interfaces at ease", Craftable, [Online]. Tersedia: <http://getcraftable.com/>. [Diakses pada Mei 2018].
- [8] "Laravel Admin Panel. Generated In Minutes.", Quick Admin Panel, [Online]. Tersedia: <http://quickadminpanel.com/>. [Diakses pada Mei 2018].
- [9] "CRUD Booster", CRUD Booster, [Online]. Tersedia: <http://crudbooster.com/>. [Diakses pada Mei 2018].
- [10] "Generating Code with Gii", Yii Framework, [Online]. Tersedia: <https://www.yiiframework.com/doc/guide/2.0/en/start-gii>. [Diakses pada Mei 2018].
- [11] B. Ajzele, Mastering PHP 7, Birmingham - Mumbai: Packt, 2017.
- [12] M. Noback, Principles of Package Design: Creating Reusable Software Components, Zeist, The Netherlands: Apress, 2018.
- [13] P. M. Jones, Modernizing Legacy Applications In PHP, Leampub, 2014.
- [14] "Packagist The PHP Package Repository", Packagist, [Online]. Tersedia: <https://packagist.org/>. [Diakses pada Januari 2019].

- [15] P. M. Jones, "PHP Package Development Standards," [Online].
Tersedia: <http://php-pds.com/>. [Diakses pada Januari 2019].