

DOKUMEN TEKNIS

“DSL WEB GALLERY GENERATOR”

IF4150 Rekayasa Perangkat Lunak Spesifik Domain

Kelas 2 - Kelompok 9 :

Steve Immanuel Harnadi	13512035
Jacqueline	13512074
Dariel Valdano	13512079
Rikysamuel	13512089



**Program Studi Teknik Informatika
Sekolah Teknik Elektro dan Informatika
Institut Teknologi Bandung
Jl. Ganesha 10, Bandung 40132**

1 Pendahuluan

1.1 Tujuan Penulisan Dokumen

Dokumen ini dibuat sebagai dokumentasi teknis terhadap aplikasi *Web Gallery Generator* berbasis *Domain-Specific Language* (DSL). Dokumen ini juga berupaya menjelaskan cara kerja aplikasi *Web Gallery Generator* beserta arsitektur program di dalamnya. Pada akhirnya, dokumen ini akan memberikan gambaran kepada *Software Engineer* dalam mengembangkan perangkat lunak secara lebih efisien menggunakan *tools* DSL, dengan studi kasus aplikasi *Web Gallery Generator*.

1.2 Lingkup Pengembangan Proyek

Aplikasi *Web Gallery Generator* secara umum berfungsi sebagai *template generator files* berbasis web. *Files* keluaran yang dihasilkan dari *template generator* ini berupa *file* php yang didukung oleh *resource files* lainnya seperti css, javascript, dan lain-lain. Untuk menghasilkan *files* keluaran tersebut, aplikasi menerima konfigurasi input dalam bentuk *script* DSL yang dimasukkan oleh pengguna. Kemudian, *script* DSL tersebut di-*parsing* dengan *library* yang mendukung DSL (dalam hal ini *Groovy Library*). Hasil *parsing* merupakan *source code* php yang sekaligus adalah *files* keluaran yang dapat diakses melalui *localhost* (*Apache HTTP Server*).

Source code program yang dihasilkan dari *web gallery generator* ini kurang lebih memiliki fungsionalitas seperti pada referensi <http://bestwebgallery.com/> (tanggal akses : 1 Desember 2015, pukul 00.30). Fitur-fitur utama *web gallery* yang dikembangkan pada kasus ini di antaranya fitur upload gambar (opsional), konfigurasi gambar *profile picture* (berdasarkan lokasi gambar yang ditentukan pengguna), deskripsi *header* dan *footer* (ditentukan pengguna) serta warnanya, dan konfigurasi *template web gallery* yang hendak digunakan (desain *web gallery* yang diinginkan pengguna). Adapun fitur-fitur tersebut dikembangkan secara terpisah untuk setiap pengguna (*independen per pengguna*).

1.3 Definisi, Akronim, dan Singkatan

Berikut adalah tabel yang berisi singkatan / akronim / istilah beserta definisinya.

Tabel 1 Daftar Istilah / Akronim / Singkatan dan Definisinya

<i>Singkatan / Akronim / Istilah</i>	<i>Definisi / Kepanjangan</i>
DSL	<i>Domain-Spesific Language</i> , bahasa pemrograman yang dapat digunakan untuk mengembangkan aplikasi / perangkat lunak domain spesifik tertentu
<i>Groovy</i>	Suatu <i>extension library</i> di lingkungan Java yang biasa digunakan untuk mengembangkan aplikasi spesifik domain dengan kaidah DSL.
<i>Grammar</i>	<i>Grammar</i> adalah suatu kumpulan <i>production rules</i> yang bersifat rekursif yang biasanya digunakan untuk mengidentifikasi konten string
CSS	<i>Cascading Style Sheet</i> , suatu jenis <i>resource file</i> pada halaman web yang berisi <i>script</i> tampilan web.
jQuery	Suatu javascript <i>library</i> yang berfungsi mempersingkat fungsi-fungsi yang biasa dilakukan javascript meliputi manipulasi konten HTML, <i>event handling</i> , animasi, atau AJAX <i>handler</i> .
PHP	<i>Hypertext Preprocessor</i> , suatu Bahasa pemrograman yang digunakan untuk mendefinisikan fungsi-fungsi pada server side
<i>Apache</i>	Suatu layanan yang disediakan <i>localhost</i> untuk protokol <i>http request</i> , biasanya terpasang di port 8080.
<i>Localhost</i>	Server lokal terintegrasi yang menyediakan berbagai macam protokol, contoh : XAMPP, WAMPP, dan lain-lain.
IDE	<i>Integrated Development Environment</i> , suatu aplikasi <i>software</i> yang menyediakan fasilitas bagi <i>programmer</i> untuk mengembangkan dan mengetes <i>source code</i> sekaligus.
SDK	<i>Software Development Kit</i> , suatu <i>tools</i> pengembangan <i>software</i> yang memuat paket <i>software</i> untuk pembuatan aplikasi tertentu.

1.4 Referensi

Desain (dan implementasi) aplikasi / perangkat lunak *web gallery generator* dikembangkan dalam lingkungan bahasa pemrograman Java, dengan IDE IntelliJIDEA

versi 14.0 dan *Groovy* SDK versi 2.4.5. Dokumentasi terkait kedua *tools* / *library* tersebut dapat dilihat pada tautan berikut ini.

1. IntelliJIDEA 14.0 : <https://www.jetbrains.com/idea/> (tanggal akses : 30 November 2015, pukul 23.00)
2. *Groovy Library* : <http://www.groovy-lang.org/learn.html> (tanggal akses : 30 November 2015, pukul 23.00)

Adapun referensi terkait pengembangan *web gallery generator* secara spesifik mengacu ke referensi <https://dzone.com/articles/groovy-dsl-simple-example> (tanggal akses : 15 November 2015, pukul 21.00) dengan beberapa modifikasi kode dari referensi tersebut.

1.5 Deskripsi Dokumen

Dokumen ini secara garis besar berisi 3 bagian utama : deskripsi, desain aplikasi, dan contoh implementasi aplikasi. Bagian deskripsi dokumen ini dibagi ke dalam deskripsi domain aplikasi *web gallery* dan deskripsi terkait *grammar* yang digunakan. Deskripsi domain dan *grammar* penting kaitannya dengan *web gallery generator* yang hendak dikembangkan, karena bagian tersebut menjelaskan *rules* DSL secara lebih rinci. Bagian desain aplikasi meliputi alur kerja dan arsitektur program. Bagian ini merupakan inti utama dari dokumen ini meliputi proses *parsing* DSL dengan *rules* yang sudah dijelaskan di bagian deskripsi beserta implementasinya dalam program. Bagian terakhir merupakan contoh implementasi aplikasi berupa studi kasus untuk setiap skenario input DSL yang mungkin dapat dilakukan beserta kesimpulan akhir penggunaan aplikasi.

2 Deskripsi Domain

Domain-Specific Language (DSL) adalah bahasa komputer yang khusus untuk suatu domain tertentu. DSL menggunakan bahasa yang lebih ekspresif dibandingkan dengan bahasa pemrograman biasa, sebab DSL dapat memisahkan logika bisnis dengan kode aplikasi. Karena DSL bersifat ekspresif, maka DSL banyak dikembangkan untuk aplikasi yang bersifat spesifik domain. Salah satu contoh aplikasi berbasis DSL yang kami pakai adalah *Groovy Library*. *Groovy* adalah bahasa dinamis untuk Java Virtual Machine (JVM) dan menggunakan *Meta Object Protocol*. *Groovy* di-*compile* langsung ke dalam *bytecode* dan *grammar*-nya fleksibel, sebagai penurunan dari Java 5.

Web Gallery adalah salah satu bentuk web yang menampilkan gambar-gambar dari sumber yang dimiliki pengguna. Salah satu contoh aplikasi yang dapat melakukan *auto-generate* gambar milik pengguna ke dalam bentuk tampilan web misalnya Adobe Photoshop. Proses *auto-generate* dari gambar (dan deskripsi) menjadi *Web Gallery* dirasa perlu karena bermanfaat bagi pengguna yang ingin mengelola *web gallery*-nya sendiri secara lebih otonom. Untuk mendukung hal tersebut, maka teknologi DSL diperlukan karena teknologi tersebut mampu mendukung pembuatan *web gallery* yang *scalable* sesuai kebutuhan spesifik pengguna.

3 Deskripsi DSL

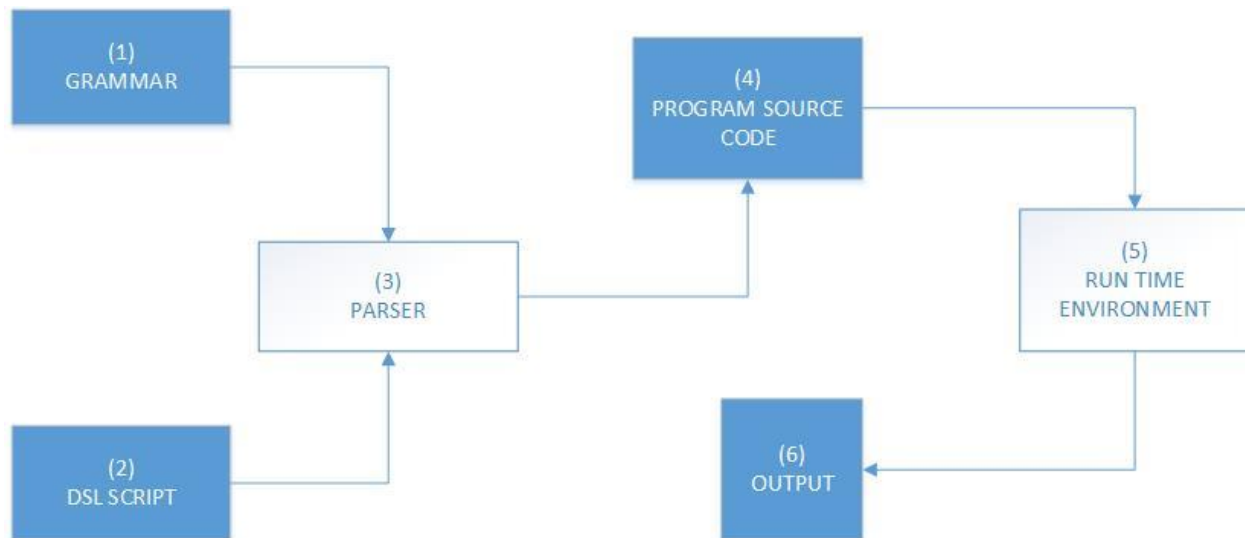
Berikut adalah *rules* yang digunakan dalam proses *parsing* *Web Gallery* DSL. Penjelasan lebih detail terkait *grammar* ini akan diuraikan lebih lanjut pada bagian 4 (alur kerja pemakaian DSL).

```
<WebGallery> ::= HTMLGenerator.make { <gallery> }
<gallery> ::= <name> | <color> | <background> | <picture> |
               <description> | <upload> | <footer> | <directory> |
               <output>
<name> ::= <string>
<color> ::= <string>
<background> ::= <string>
<picture> ::= <string>
<description> ::= <string>
<upload> ::= <boolean>
<footer> ::= <string>
<directory> ::= <string>
<output> ::= <string>
<boolean> ::= true | false
<string> ::= "(a-z, A-Z, 0-9)"
```

Kode 1 Grammar Rules DSL pada Web Gallery

4 Alur Kerja Pemakaian DSL

Berikut adalah arsitektur yang menggambarkan alur kerja pemakaian *web gallery* DSL secara umum. Kotak biru pada gambar di bawah adalah data yang dihasilkan atau dimasukkan, sedangkan kotak putih adalah kakas menampilkan output.



Gambar 1 Alur Kerja Pemakaian DSL Secara Umum

Berikut adalah penjelasan setiap komponen alur kerja pemakaian *web gallery* DSL seperti yang tertuang pada gambar di atas.

(1) Grammar :

Grammar yang digunakan pada *web gallery* DSL ini sudah digambarkan secara rinci pada bab deskripsi DSL. Secara umum, *web gallery* DSL yang dibuat mem-*parsing script* yang dimasukkan pengguna ke dalam 9 konfigurasi pilihan : *name*, *color*, *background*, *picture*, *description*, *upload*, *footer*, *directory*, dan *output*. Konfigurasi *name*, *color*, *background*, *picture*, *description*, *footer*, *directory*, dan *output* harus dimasukkan dalam format *ASCII String*. Adapun *upload* dimasukkan dalam bentuk *Boolean* dan *template* dalam bentuk angka 1-3 bergantung dari opsi *template* yang ingin dipakai.

(2) DSL Script :

Berdasarkan *grammar* yang sudah dijelaskan pada butir (1), maka pemakai *web gallery* DSL memasukkan *script* dengan format sebagai berikut. *Script* ini dijalankan terpisah dari kelas Java / *Groovy* dan berfungsi sebagai *main* dari eksekusi DSL.

```

HTMLGenerator.make {
    name <nama pengguna web gallery>
    color <warna untuk tampilan nama pengguna web gallery>
    background <warna latar belakang dari web gallery>
  }

```

```

picture <gambar pada folder assets yang hendak dipakai sebagai
      profpic>
description <deskripsi gambar yang hendak dimasukkan>
upload <konfigurasi apakah pemilik web gallery memakai fitur
      upload gambar atau tidak (nilai true / false)>
footer <kalimat footer yang ingin ditampilkan pemilik web
      gallery>
directory <lokasi tempat aplikasi akan di-install>
output <nama file output yang ingin di-generate>
templateOne/templateTwo <token wajib yang harus dipanggil pada
      script ini (harus diletakkan di baris terakhir script
      ini)>
}

```

Kode 2 Format Input Script DSL yang Dimasukkan Pengguna

Berikut adalah contoh pemakaian *script* di atas pada *main program* Java.

```

HTMLGenerator.make {
    name "Lorem Ipsum"
    color "red"
    background "#000000"
    picture "D://img//landscape0.jpg"
    description "Lorem ipsum dolor sit amet, consectetur adipiscing"
    upload false
    footer "Online Gallery Copyright 2015."
    directory "F:\\xampp\\htdocs\\Gallery"
    output "index"
    templateTwo
}

```

Kode 3 Contoh Input Script DSL yang Dimasukkan Pengguna

(3) Parser :

Pada tahap ini, *parser* (dalam hal ini adalah *Groovy Library*) melakukan validasi terhadap *script* DSL. Jika *script* lolos validasi, maka opsi / data yang sudah dimasukkan pengguna melalui *script* tersebut akan disimpan dalam model kelas *Groovy* yang sudah dibuat. Sebaliknya, jika tidak lolos validasi maka program akan menampilkan pesan *error* ke pengguna. Kemudian, model kelas *Groovy* tersebut mengeksekusi XML *builder* yang men-*generate template* HTML berdasarkan opsi yang sudah diterimanya.

(4) Program Source Code :

Hasil *auto-generate* dari *script* DSL yang dimasukkan adalah *source code* berbasis html dan *resource files* lain yang mendukung tampilan html tersebut. Secara umum, *source code* tersebut dibagi ke dalam folder / *file* dengan struktur direktori berikut.

- a) Folder *css* yang berisi kode yang mengatur tampilan web
- b) Folder *images* yang nantinya akan berisi *resource* gambar dan deskripsi gambar yang terkandung di konten web
- c) Folder *jQuery* berisi *library* dan file *javascript* untuk mendukung fungsionalitas *client-side* seperti *lightbox* (saat satu gambar diperbesar)
- d) File *avatar.jpg* yang merupakan *profile picture* dari pemilik web gallery
- e) File *php* yang berisi fungsi-fungsi *server-side* yang akan memproses request dari *client* terkait fungsionalitas upload gambar (jika fitur dinyalakan)

Dalam implementasinya, struktur direktori di atas dapat disimpan di lokasi apa saja sesuai dengan *path* direktori yang diinginkan pengguna. Berikut adalah *screenshot* struktur direktori *source code* berbasis HTML yang dihasilkan dari program ini.

Name	Date modified	Type	Size
css	11/30/2015 7:14 PM	File folder	
images	11/30/2015 7:14 PM	File folder	
jquery	11/30/2015 7:14 PM	File folder	
php	11/30/2015 7:14 PM	File folder	
avatar.jpg	11/30/2015 7:14 PM	JPG File	1,908 KB
index.php	11/30/2015 7:14 PM	PHP File	2 KB

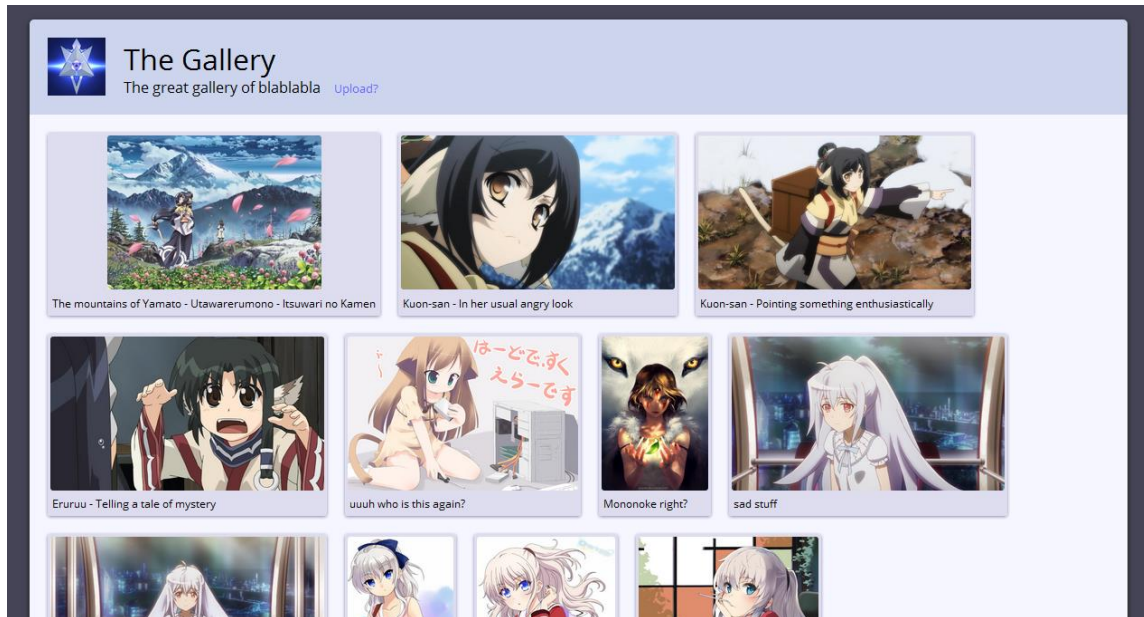
Gambar 2 Struktur Direktori HTML Web Gallery yang Dihasilkan dari DSL

(5) Runtime Environment :

Source Code hasil *auto-generate* dapat dijalankan pada lingkungan *Apache Local HTTP Server* XAMPP, karena dibutuhkan kode PHP yang meng-*handle* upload gambar pada *web gallery* (jika pengguna memilih opsi *true*).

(6) Output Source Code :

Berikut adalah salah satu contoh tampilan *web gallery* yang dihasilkan setelah memasukkan direktori web html di gambar 2 ke *Apache Local HTTP Server XAMPP*.

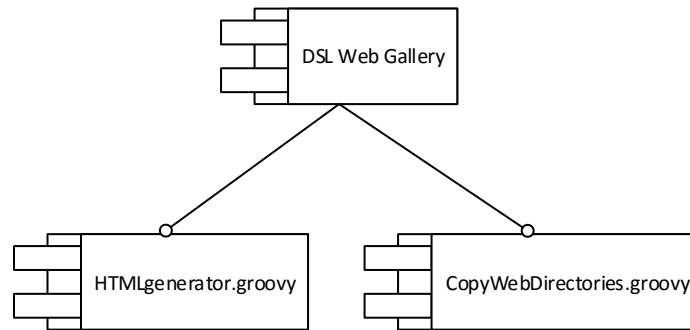


Gambar 3 Contoh hasil web gallery yang dihasilkan untuk seorang client

The screenshot shows a web page titled "The Gallery Uploader" with the subtitle "Upload Images here" and a link "back?". The page contains a form for uploading images. It states "Only .PNG or .JPG files are accepted." Below this, there is an "Input File:" label, a "Browse..." button, and the text "No file selected." There is also a "Description:" label followed by a text input field. At the bottom of the form is a "Submit Query" button. The footer of the page says "Online Gallery copyright BlaBlaBla 2015."

Gambar 4 Contoh hasil upload form untuk menambah gambar

5 Arsitektur Perangkat Lunak dan Deskripsi Modul



Gambar 5 Arsitektur Perangkat Lunak DSL Web Gallery

DSL Web Gallery terdiri dari 2 modul utama:

- HTMLGenerator.groovy adalah parser DSL utama yang akan membaca script DSL dan membuat HTML berdasarkan dari script tersebut.
- CopyWebDirectories.groovy adalah komponen pembantu yang digunakan untuk meng-copy file file web lainnya seperti font, javascript, css dan sebagainya.

6 Contoh Kasus

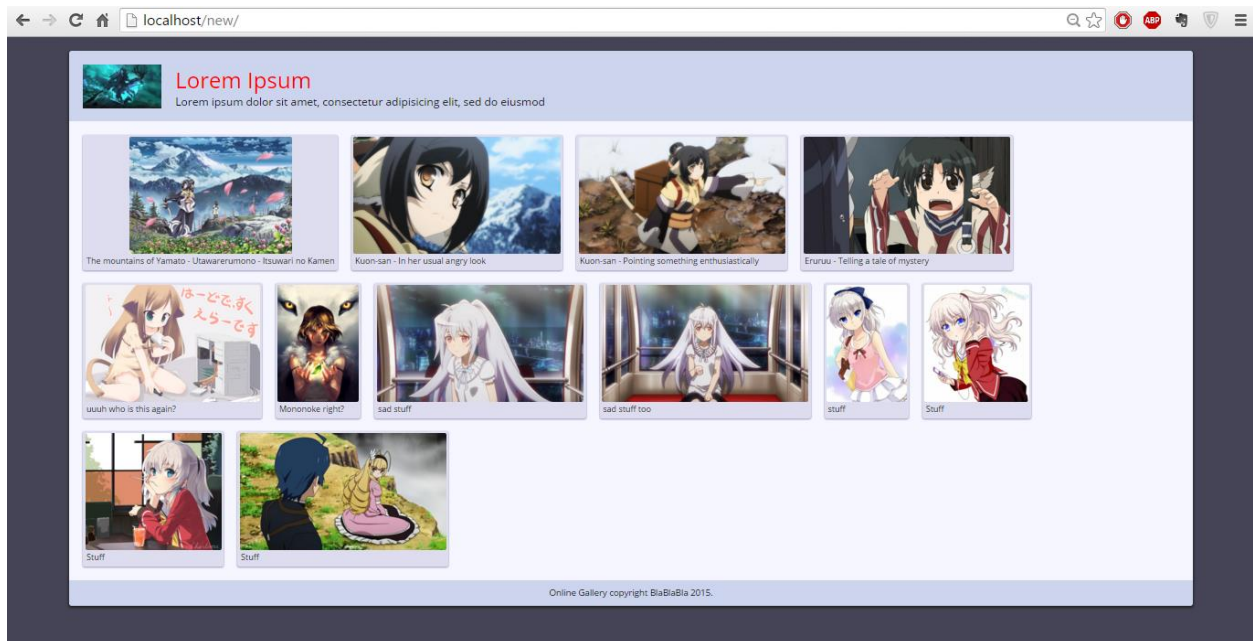
6.1 Contoh Kasus 1

Berikut adalah contoh kasus dimana *user* yang bersangkutan memasukan *script* yang lengkap.

```
HTMLGenerator.make {
    name "Lorem Ipsum"
    color "red"
    picture "E:\\abadon.png"
    description "Lorem ipsum dolor sit amet, consectetur adipiscing elit, sed do eiusmod"
    upload false
    footer "Online Gallery copyright BlaBlaBla 2015."
    directory "F:\\xampp\\htdocs\\new"
    output "index"
    templateTwo
}
```

Gambar 6 Kode Contoh Kasus 1

Hasilnya adlaah judul *Web Gallery* yang berjudul Lorem Ipsum dengan font berwarna merah. Dalam *folder images/* sudah terdapat beberapa gambar sehingga *web gallery* me-load gambar yang tersedia



Gambar 7 Hasil Contoh Kasus 1

6.2 Contoh Kasus 2

Contoh kasus kedua, sama seperti contoh kasus 1 tapi tanpa menggunakan *syntax* “color”

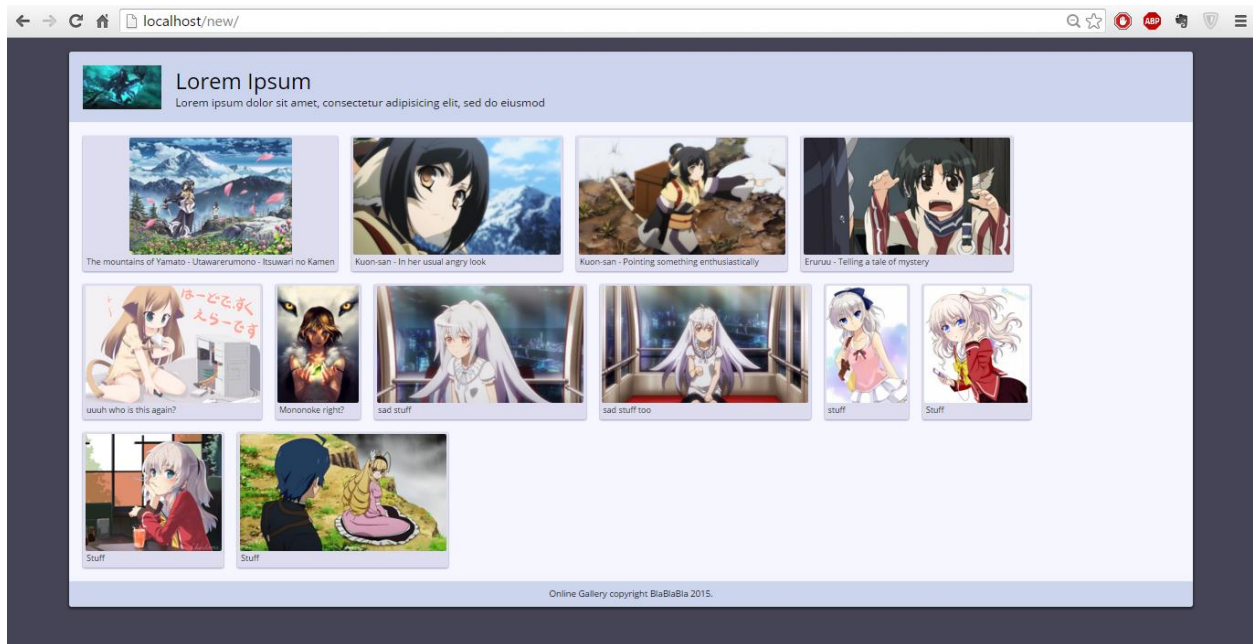
```

H\Generator.make {
  name "Lorem Ipsum"
  picture "E:\\abadon.png"
  description "Lorem ipsum dolor sit amet, consectetur adipiscing elit, sed do eiusmod"
  upload false
  footer "Online Gallery copyright BlaBlaBla 2015."
  directory "F:\\xampp\\htdocs\\new"
  output "index"
  templateTwo
}

```

Gambar 8 Kode Contoh Kasus 2

Hasil nya adalah *Web Gallery* dari template 2 dengan konfigurasi *default*.



Gambar 9 Hasil Contoh Kasus 2

6.3 Contoh Kasus 3

Contoh kasus 3 jika user memasukan *syntax* tidak dikenal “*unknownSyntax*”

```
HTMLGenerator.make {
  name "Lorem Ipsum"
  picture "E:\\abaddon.png"
  description "Lorem ipsum dolor sit amet, consectetur adipisicing elit, sed do eiusmod"
  upload false
  footer "Online Gallery copyright BlaBlaBla 2015."
  directory "F:\\xampp\\htdocs\\new"
  output "index"
  unknownSyntax "12345"
  templateTwo
}
```

Gambar 10 Contoh Kasus 3

Hasilnya mengeluarkan pesan “*Wrong Syntax*”

```
Wrong Syntax

Process finished with exit code 0
```

Gambar 11 Hasil Contoh Kasus 3

7 Kesimpulan

Web Gallery DSL yang kami kembangkan secara keseluruhan masih berfungsionalitas sederhana. Meskipun demikian, memodifikasi tampilan *web gallery* satu per satu sesuai keinginan pengguna adalah hal yang naif. Sebab, kakas *Groovy Library* terbukti sangat mempermudah proses modifikasi tampilan *web gallery* dari manual menjadi semiotomatis. Dengan demikian, pengguna itu sendiri dapat mengatur tampilan *web gallery* sesuai dengan desain / *layout* yang diinginkannya bahkan tanpa bantuan dari *programmer* langsung.