

# SKRIPSI

## KUSTOMISASI SHARIF JUDGE UNTUK KEBUTUHAN PROGRAM STUDI TEKNIK INFORMATIKA



Stillmen Vallian

NPM: 2014730083

PROGRAM STUDI TEKNIK INFORMATIKA  
FAKULTAS TEKNOLOGI INFORMASI DAN SAINS  
UNIVERSITAS KATOLIK PARAHYANGAN

«tahun»



# UNDERGRADUATE THESIS

«JUDUL BAHASA INGGRIS»



Stillmen Vallian

NPM: 2014730083

DEPARTMENT OF INFORMATICS  
FACULTY OF INFORMATION TECHNOLOGY AND SCIENCES  
PARAHYANGAN CATHOLIC UNIVERSITY

«tahun»



# LEMBAR PENGESAHAN

**KUSTOMISASI SHARIF JUDGE UNTUK KEBUTUHAN PROGRAM STUDI TEKNIK  
INFORMATIKA**

**Stillmen Vallian**

**NPM: 2014730083**

**Bandung, «tanggal» «bulan» «tahun»**

**Menyetujui,**

**Pembimbing Utama**

**Pembimbing Pendamping**

**«pembimbing utama/1»**

**«pembimbing pendamping/2»**

**Ketua Tim Penguji**

**Anggota Tim Penguji**

**«penguji 1»**

**«penguji 2»**

**Mengetahui,**

**Ketua Program Studi**

**Mariskha Tri Adithia, P.D.Eng**



## PERNYATAAN

Dengan ini saya yang bertandatangan di bawah ini menyatakan bahwa skripsi dengan judul:

### **KUSTOMISASI SHARIF JUDGE UNTUK KEBUTUHAN PROGRAM STUDI TEKNIK INFORMATIKA**

adalah benar-benar karya saya sendiri, dan saya tidak melakukan penjiplakan atau pengutipan dengan cara-cara yang tidak sesuai dengan etika keilmuan yang berlaku dalam masyarakat keilmuan.

Atas pernyataan ini, saya siap menanggung segala risiko dan sanksi yang dijatuhkan kepada saya, apabila di kemudian hari ditemukan adanya pelanggaran terhadap etika keilmuan dalam karya saya, atau jika ada tuntutan formal atau non-formal dari pihak lain berkaitan dengan keaslian karya saya ini.

Dinyatakan di Bandung,  
Tanggal «tanggal» «bulan» «tahun»

Meterai Rp. 6000
---------------------

Stillmen Vallian  
NPM: 2014730083





## ABSTRAK

«Tuliskan abstrak anda di sini, dalam bahasa Indonesia»

Lorem ipsum dolor sit amet, consectetur adipiscing elit. Ut purus elit, vestibulum ut, placerat ac, adipiscing vitae, felis. Curabitur dictum gravida mauris. Nam arcu libero, nonummy eget, consectetur id, vulputate a, magna. Donec vehicula augue eu neque. Pellentesque habitant morbi tristique senectus et netus et malesuada fames ac turpis egestas. Mauris ut leo. Cras viverra metus rhoncus sem. Nulla et lectus vestibulum urna fringilla ultrices. Phasellus eu tellus sit amet tortor gravida placerat. Integer sapien est, iaculis in, pretium quis, viverra ac, nunc. Praesent eget sem vel leo ultrices bibendum. Aenean faucibus. Morbi dolor nulla, malesuada eu, pulvinar at, mollis ac, nulla. Curabitur auctor semper nulla. Donec varius orci eget risus. Duis nibh mi, congue eu, accumsan eleifend, sagittis quis, diam. Duis eget orci sit amet orci dignissim rutrum.

**Kata-kata kunci:** «Tuliskan di sini kata-kata kunci yang anda gunakan, dalam bahasa Indonesia»



## ABSTRACT

«Tuliskan abstrak anda di sini, dalam bahasa Inggris»

Nam dui ligula, fringilla a, euismod sodales, sollicitudin vel, wisi. Morbi auctor lorem non justo. Nam lacus libero, pretium at, lobortis vitae, ultricies et, tellus. Donec aliquet, tortor sed accumsan bibendum, erat ligula aliquet magna, vitae ornare odio metus a mi. Morbi ac orci et nisl hendrerit mollis. Suspendisse ut massa. Cras nec ante. Pellentesque a nulla. Cum sociis natoque penatibus et magnis dis parturient montes, nascetur ridiculus mus. Aliquam tincidunt urna. Nulla ullamcorper vestibulum turpis. Pellentesque cursus luctus mauris.

**Keywords:** «Tuliskan di sini kata-kata kunci yang anda gunakan, dalam bahasa Inggris»



*«kepada siapa anda mempersembahkan skripsi ini...?»*



## KATA PENGANTAR

«Tuliskan kata pengantar dari anda di sini ...»

Nulla malesuada porttitor diam. Donec felis erat, congue non, volutpat at, tincidunt tristique, libero. Vivamus viverra fermentum felis. Donec nonummy pellentesque ante. Phasellus adipiscing semper elit. Proin fermentum massa ac quam. Sed diam turpis, molestie vitae, placerat a, molestie nec, leo. Maecenas lacinia. Nam ipsum ligula, eleifend at, accumsan nec, suscipit a, ipsum. Morbi blandit ligula feugiat magna. Nunc eleifend consequat lorem. Sed lacinia nulla vitae enim. Pellentesque tincidunt purus vel magna. Integer non enim. Praesent euismod nunc eu purus. Donec bibendum quam in tellus. Nullam cursus pulvinar lectus. Donec et mi. Nam vulputate metus eu enim. Vestibulum pellentesque felis eu massa.

Quisque ullamcorper placerat ipsum. Cras nibh. Morbi vel justo vitae lacus tincidunt ultrices. Lorem ipsum dolor sit amet, consectetur adipiscing elit. In hac habitasse platea dictumst. Integer tempus convallis augue. Etiam facilisis. Nunc elementum fermentum wisi. Aenean placerat. Ut imperdiet, enim sed gravida sollicitudin, felis odio placerat quam, ac pulvinar elit purus eget enim. Nunc vitae tortor. Proin tempus nibh sit amet nisl. Vivamus quis tortor vitae risus porta vehicula.

Bandung, «bulan» «tahun»

Penulis





# DAFTAR ISI

KATA PENGANTAR	xv
DAFTAR ISI	xvii
DAFTAR GAMBAR	xix
DAFTAR TABEL	xxi
<b>1 PENDAHULUAN</b>	<b>1</b>
1.1 Latar Belakang . . . . .	1
1.2 Rumusan Masalah . . . . .	1
1.3 Tujuan . . . . .	1
1.4 Batasan Masalah . . . . .	2
1.5 Metodologi . . . . .	2
1.6 Sistematika Pembahasan . . . . .	2
<b>2 LANDASAN TEORI</b>	<b>3</b>
2.1 <i>CodeIgniter</i> . . . . .	3
2.1.1 Fitur-fitur CodeIgniter . . . . .	4
2.1.2 <i>Flow Chart</i> Aplikasi . . . . .	5
2.1.3 <i>Model-View-Controller</i> . . . . .	5
2.1.4 Desain dan Tujuan Arsitektur . . . . .	6
2.2 <i>Sharif Judge</i> . . . . .	6
2.2.1 Instalasi . . . . .	7
2.2.2 <i>Clean URLs</i> . . . . .	8
2.2.3 <i>Users</i> . . . . .	8
2.3 L <sup>A</sup> T <sub>E</sub> X . . . . .	9
2.4 Template Skripsi FTIS UNPAR . . . . .	9
2.4.1 Tabel . . . . .	10
2.4.2 Kutipan . . . . .	10
2.4.3 Gambar . . . . .	11
DAFTAR REFERENSI	15
A KODE PROGRAM	17
B HASIL EKSPERIMEN	19



## DAFTAR GAMBAR

2.1	<i>Flow Chart</i> Aplikasi . . . . .	5
2.2	Gambar <i>Serpentes</i> dalam format png . . . . .	12
2.3	Ular kecil . . . . .	12
2.4	<i>Serpentes</i> betina . . . . .	13
B.1	Hasil 1 . . . . .	19
B.2	Hasil 2 . . . . .	19
B.3	Hasil 3 . . . . .	19
B.4	Hasil 4 . . . . .	19



## DAFTAR TABEL

2.1	<i>User Roles Table</i>	8
2.2	<i>Permission Table</i>	9
2.3	Tabel contoh	10
2.4	Tabel bewarna(1)	10
2.5	Tabel bewarna(2)	10



# BAB 1

## PENDAHULUAN

### 1.1 Latar Belakang

*Sharif Judge* adalah *grader* otomatis yang mampu menilai ketepatan serta performansi program yang dikumpulkan mahasiswa. Perangkat lunak ini diciptakan oleh Mohammad Javad Naderi dan bersifat *open source*. *Web Interface* perangkat lunak ini dibuat menggunakan *framework CodeIgniter* dan *backend* menggunakan *BASH* [1]. Selain sebagai *grader*, perangkat lunak ini memiliki beberapa fungsi seperti deteksi plagiarisme jawaban para peserta. Cara kerja perangkat lunak ini dimulai dari dosen memasukkan data yang dibutuhkan berupa soal, peserta, dan kunci jawaban. Data yang dimasukan tersebut dapat diakses oleh para peserta. Peserta dapat mengumpulkan jawaban dalam bentuk kode program ke dalam *Sharif Judge*. *Sharif Judge* akan menjalankan kode program dan menyesuaikan dengan kunci jawaban, lalu *grader* akan menilai jawaban para peserta.

*Sharif Judge* digunakan oleh Jurusan Teknik Informatika Universitas Katolik Parahyangan pada mata kuliah seperti Algoritma dan Struktur Data serta Desain Analisis dan Algoritma. Perangkat lunak ini sangat membantu dosen dan mahasiswa dalam bidang akademik. Sistem penilaian otomatis merupakan salah satu fitur yang sering digunakan oleh para dosen. Dengan memanfaatkan fitur di atas, dosen dapat dengan mudah memberikan nilai tugas, kuis, atau ujian ke mahasiswa. Mahasiswa juga dapat melihat nilai secara langsung setelah jawaban dikumpulkan. Jika masih ada waktu, mahasiswa masih dapat memperbaiki jawaban yang salah. Ketika waktu sudah habis, jawaban terakhir yang dikumpulkan akan diambil sebagai jawaban final mahasiswa.

Pada prakteknya *Sharif Judge* masih butuh pengembangan, karena Jurusan Teknik Informatika memiliki kebutuhan yang lebih spesifik seperti *login* yang terintegrasi dengan *password* pada Teknik Informatika. Selain itu *Sharif Judge* terakhir dicommit pada *Github* pada bulan Juli 2015, dan masih ada beberapa bug yang belum diperbaiki. Hal tersebut menyebabkan *Sharif Judge* kurang memenuhi kebutuhan program studi Teknik Informatika.

Pada skripsi ini, peneliti akan mengembangkan *Sharif Judge* agar sesuai dengan kebutuhan yang disebutkan diatas. Dari kebutuhan yang disebutkan diatas, akan dirancang fitur-fitur untuk diimplementasikan pada *Sharif Judge*. Dengan pengimplementasian fitur yang baru, diharapkan kebutuhan mahasiswa dan dosen dapat terpenuhi.

### 1.2 Rumusan Masalah

1. Fitur-fitur apa saja yang dibutuhkan oleh Teknik Informatika?
2. Bagaimana mengembangkan *Sharif Judge* sehingga memenuhi kebutuhan Teknik Informatika?

### 1.3 Tujuan

1. Menganalisa dan mengetahui fitur-fitur yang dibutuhkan Teknik Informatika.
2. Mengimplementasi kebutuhan program studi Teknik Informatika pada *Sharif Judge*.

## 1.4 Batasan Masalah

Batasan masalah yang dibuat terkait dengan pengerjaan skripsi ini adalah sebagai berikut:

1. Perangkat lunak akan dikembangkan sesuai dengan kebutuhan yang telah dianalisis oleh para dosen pengguna dan daftar isu pada repositori *Sharif Judge*.

## 1.5 Metodologi

Metodologi yang dilakukan dalam pengerjaan skripsi ini adalah sebagai berikut:

1. Studi literatur mengenai :
  - *CodeIgniter* sebagai framework untuk mengembangkan perangkat lunak.
  - Dokumentasi *Sharif Judge* sebagai panduan untuk mengembangkan perangkat lunak.
2. Menganalisis kebutuhan-kebutuhan dari para dosen pengguna *Sharif Judge* dan daftar isu pada repositori *Sharif Judge* pada *Github*.
3. Merancang dan menentukan fitur yang akan diimplementasi.
4. Mengimplementasikan fitur terhadap perangkat lunak.
5. Mengujikan perangkat lunak ke mata kuliah selama satu semester.
6. Membuat dokumentasi perangkat lunak.

## 1.6 Sistematika Pembahasan

Setiap bab dalam skripsi ini memiliki sistematika penulisan yang dijelaskan kedalam poin-poin sebagai berikut:

1. Bab 1 : Pendahuluan  
Bab 1 membahas mengenai gambaran umum penelitian ini. Berisi tentang latar belakang, rumusan masalah, tujuan, batasan masalah, metode penelitian, dan sistematika penulisan.
2. Bab 2 : Dasar Teori  
Bab 2 membahas mengenai teori-teori yang mendukung berjalannya penelitian ini. Berisi tentang *CodeIgniter* dan dokumentasi *Sharif Judge*.
3. Bab 3 : Analisis  
Bab 3 membahas mengenai analisa masalah.
4. Bab 4 : Perancangan  
Bab 4 membahas mengenai perancangan yang dilakukan sebelum melakukan tahapan implementasi.
5. Bab 5 : Implementasi dan Pengujian  
Bab 5 membahas mengenai implementasi dan pengujian yang telah dilakukan.
6. Bab 6 : Kesimpulan dan Saran  
Bab 6 membahas hasil kesimpulan dari keseluruhan penelitian ini dan saran-saran yang dapat diberikan untuk penelitian berikutnya.



## BAB 2

### LANDASAN TEORI

Bab ini membahas tentang landasan teori yang akan digunakan dalam skripsi ini yang diambil dari dua sumber, yaitu "CodeIgniter Documentation" karya British Columbia Institute of Technology [2] dan "Sharif Judge Documentation" karya Mohammad Javad Naderi [3].

#### 2.1 *CodeIgniter*

*CodeIgniter* merupakan sebuah *framework* bagi pengguna yang ingin membangun aplikasi web menggunakan PHP. Tujuan utamanya adalah memungkinkan para pengguna mengembangkan proyek-proyek menjadi lebih cepat dibandingkan menulis kode dari awal. *Framework* ini memiliki banyak library untuk tugas-tugas yang biasa diperlukan, serta antarmuka dan struktur logis yang sederhana untuk mengakses library ini. *CodeIgniter* membuat para pengguna lebih fokus pada proyek dengan meminimalkan jumlah kode yang dibutuhkan untuk tugas yang diberikan [2].

Beberapa keunggulan dari *CodeIgniter* yaitu:

- *Framework* yang Ringan  
Inti dari sistem *CodeIgniter* hanya membutuhkan *library* yang kecil. Hal ini sangat berbeda dengan *framework* lain yang membutuhkan *resource* yang lebih. *Library* tambahan dimuat secara dinamis atau sesuai dengan permintaan sehingga sistem dapat berjalan cepat.
- Menggunakan Konsep M-V-C  
*CodeIgniter* menggunakan pendekatan *Model-View-Controller* yang memungkinkan pemisahan anatara logika dan presentasi.
- Menghasilkan *Clean URLs*  
*URL* yang dihasilkan oleh *CodeIgniter* bersih dan *search-engine friendly*. *CodeIgniter* menggunakan pendekatan *segment-based* seperti:  
example.com/news/article/345
- *Packs a Punch*  
*CodeIgniter* dilengkapi dengan *library* yang umumnya diperlukan untuk tugas pengembangan web seperti mengakses database, mengirim email, memvalidasi data *form*, menjaga *session*, memanipulasi gambar, bekerja dengan *XML-RPC* data dan masih banyak lagi.
- *Extensible*  
Sistem dapat dengan mudah diperluas dengan menggunakan *library* pengguna, *helper*, atau melalui *class extensions* dan *system hooks*.
- Tidak Membutuhkan *Template Engine*  
*CodeIgniter* dilengkapi dengan *template parser* sederhana yang dapat digunakan secara opsional. *Template Engine* tidak dapat menandingi kinerja dari *native* PHP. Sintak yang harus dipelajari untuk menggunakan *Template Engine* biasanya lebih mudah dari mempelajari

dasar-dasar PHP. Perhatikan potongan kode PHP di bawah ini:

```
<ul>
<?php foreach ($addressbook as $name):?>
<li><?=$name?></li>
<?php endforeach; ?>
</ul>
```

Sangat berlawanan dengan *pseudo-code* yang digunakan oleh *Template Engine*:

```
<ul>
{foreach from=$addressbook item="name"}
<li>{$name}</li>
{/foreach}
</ul>
```

Terlihat *Template Engine* sedikit lebih bersih, namun harus ditukar dengan performa yang kurang baik karena *pseudo-code* harus dikonversi kembali menjadi PHP. Salah satu tujuan dari *CodeIgniter* adalah performa maksimal, oleh karena itu *CodeIgniter* tidak menggunakan *Template Engine*.

- Dokumentasi yang Baik  
Dokumentasi merupakan salah satu bagian terpenting dari kode itu sendiri. *CodeIgniter* berkomitmen membuat kode yang sangat bersih dan terdokumentasi dengan baik.

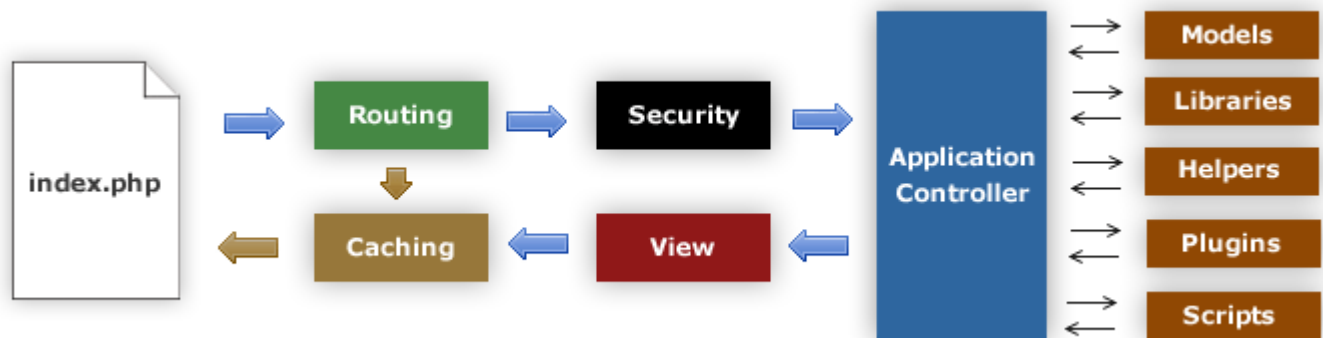
### 2.1.1 Fitur-fitur CodeIgniter

Berikut beberapa fitur utama yang terdapat pada *framework CodeIgniter* seperti:

- Sistem berbasis *MVC*
- *Framework* yang ringan
- *Database Class* yang lengkap dengan dukungan untuk beberapa platform
- Dukungan *query builder* untuk database
- *Form* dan validasi data
- Keamanan dan *XSS Filtering*
- *Session Management*
- *Email Sending Class*
- *Image Manipulation Library*
- *File Uploading Class*
- *Calendaring Class*
- *Unit Testing Class*

### 2.1.2 *Flow Chart Aplikasi*

Gambar 2.1 menunjukkan bagaimana data mengalir ke seluruh sistem [2]:



Gambar 2.1: *Flow Chart Aplikasi*

1. File *index.php* berfungsi sebagai *front controller* dan menginisialisasi *resource* utama yang dibutuhkan untuk menjalankan *CodeIgniter*.
2. *Router* memeriksa *HTTP request* untuk menentukan apa yang harus dilakukan.
3. Jika terdapat *file cache*, maka akan langsung dikirimkan ke *browser*.
4. *HTTP request* dan data pengguna yang dikirim akan terlebih dahulu disaring untuk alasan keamanan. *Application controller* akan dimuat setelah proses penyaringan selesai.
5. *Controller* akan memuat *model*, *core libraries*, *helpers* dan *resource* lain yang dibutuhkan untuk memproses permintaan khusus.
6. *View* akan di *render* kemudian dikirim ke web *browser*. Jika proses *caching* diaktifkan, maka *View* akan di *cache* terlebih dahulu sehingga permintaan berikutnya dapat dilayani.

### 2.1.3 *Model-View-Controller*

*CodeIgniter* merupakan *framework* yang menggunakan pola pengembangan *Model-View-Controller*. *MVC* adalah pendekatan perangkat lunak yang memisahkan logika aplikasi dari presentasi. Hal tersebut memungkinkan halaman web pengguna memiliki *scripting* yang minimal karena presentasi terpisah dari *scripting* PHP [2].

- *Model* merepresentasikan bagian struktur data pengguna. Biasanya kelas *Model* akan berisikan fungsi-fungsi yang membantu pengguna untuk mengambil, menyimpan, dan memperbarui informasi pada database.
- *View* merupakan informasi yang akan ditampilkan kepada pengguna. Umumnya *View* merupakan sebuah halaman web, namun pada *CodeIgniter*, *View* dapat berupa bagian-bagian halaman seperti *header* atau *footer*. Selain itu *View* juga dapat berupa halaman *RSS* atau jenis "halaman" lainnya.
- *Controller* berfungsi sebagai perantara antara *Model*, *View*, dan *resource* lain yang dibutuhkan untuk memproses *HTTP request* dan menghasilkan halaman web.

*CodeIgniter* memiliki pendekatan yang cukup fleksibel terhadap *MVC* karena *Model* tidak selalu diperlukan. Para pengguna dapat membangun aplikasi minimal menggunakan *Controller* dan *View*. Hal tersebut dapat dilakukan jika pengguna tidak memerlukan adanya pemisahan tambahan atau pengguna merasa bahwa mempertahankan sebuah *Model* membutuhkan kompleksitas yang lebih tinggi [2].

### 2.1.4 Desain dan Tujuan Arsitektur

Dari sudut pandang teknis, dan arsitektural, *CodeIgniter* dibuat dengan tujuan sebagai berikut:

- *Dynamic Instation*  
Dalam *CodeIgniter*, komponen dimuat dan rutinitas dieksekusi hanya jika diminta. Tidak ada asumsi yang dibuat oleh sistem tentang apa yang mungkin diperlukan di luar resource utama, sehingga sistem ini sangat ringan secara *default*. *Event*, *Controller* dan *View* yang pengguna rancang akan menentukan apa yang dipanggil.
- *Loose Coupling*  
*Coupling* adalah sejauh mana komponen-komponen dari sistem saling mengandalkan satu sama lain. Semakin sedikit komponen yang bergantung satu sama lain, maka komponen tersebut lebih dapat digunakan kembali dan sistem menjadi fleksibel. Tujuan dari *framework* ini adalah sistem yang sangat longgar (*very loosely coupled system*).
- *Component Singularity*  
*Singularity* adalah sejauh mana komponen memiliki tujuan yang difokuskan secara sempit. Dalam *CodeIgniter*, setiap kelas dan fungsinya sangat otonom. Hal tersebut memungkinkan fungsi dapat berjalan secara maksimal.

*CodeIgniter* merupakan sistem yang *loosely coupled* dengan singularitas komponen yang tinggi (*dynamically instantiated*). *Codeigniter* berusaha untuk sederhana, fleksible, dan kinerja tinggi dengan paket yang sekecil mungkin [2].

## 2.2 Sharif Judge

*Sharif Judge* adalah *grader* otomatis yang mampu menilai ketepatan serta performansi program yang dikumpulkan mahasiswa. Perangkat lunak ini diciptakan oleh Mohammad Javad Naderi dan bersifat *open source*. *Web Interface* perangkat lunak ini ditulis menggunakan PHP (*framework CodeIgniter*) dan *backend* menggunakan *BASH* [1]. Selain sebagai *grader* otomatis, *Sharif Judge* juga memiliki beberapa fitur lainnya seperti:

- Beberapa peran pengguna (*admin*, *head instructor*, *instructor*, *student*)
- Sandboxing (belum diterapkan untuk *python*)
- Deteksi kecurangan (mendeteksi kode yang mirip) menggunakan *Moss*
- Pengaturan untuk menilai keterlambatan pengiriman
- Antrian Pengiriman
- Mengunduh hasil dalam bentuk *file excel*
- Mengunduh kode yang telah dikirim dalam bentuk *file zip*
- Metode "*Output Comparison*" dan "*Tester Code*" untuk memeriksa kebenaran dari hasil keluaran.

- Menambahkan beberapa pengguna sekaligus
- Diskripsi Masalah (*PDF/Markdown/HTML*)
- Penilaian ulang (*rejudge*)
- Papan nilai
- Notifikasi

### 2.2.1 Instalasi

Untuk menjalankan Sharif Judge membutuhkan sebuah *server Linux* dengan persyaratan berikut:

- *Webserver* menjalankan PHP versi 5.3 atau yang lebih baru
- Pengguna harus dapat menjalankan php dari *command line*. Pada *Ubuntu*, pengguna perlu menginstal paket *php5-cli*
- *Mysql database* (dengan ekstensi *mysqli* untuk PHP) atau *PostgreSQL database*
- PHP harus memiliki akses untuk menjalankan *shell commands* menggunakan fungsi *shell\_exec*. Contohnya seperti *command* di bawah ini:

```
echo shell_exec("php -v");
```

- Perangkat yang digunakan untuk *compiling* dan menjalankan kode yang dikumpulkan (*gcc, g++, javac, java, python2, python3 commands*)
- *Perl* lebih baik diinstal untuk ketepatan waktu, batas memori dan memaksimalkan batas ukuran pada *output* kode yang dikirimkan

Jika persyaratan di atas telah terpenuhi, maka akan masuk tahap instalasi sebagai berikut:

- Mengunduh versi terakhir dari Sharif Judge dan *unpack* hasil download di direktori *public html*
- (Pilihan) Pindahkan folder *system* dan *application* keluar dari *public directory* dan masukan *path* lengkap di *file index.php*

```
$system_path = '/home/mohammad/secret/system';
$application_folder = '/home/mohammad/secret/application';
```

- Buat sebuah *Mysql* atau *PostgreSQL* database untuk Sharif Judge. Jangan menginstall paket koneksi database untuk *C/C++, Java*, atau *Python*
- Atur pengaturan koneksi database di file *application/config/database.php*. Pengguna dapat menggunakan awalan untuk nama tabel.

```
/* Enter database connection settings here: */
'dbdriver' => 'postgre', // database driver (mysqli, postgre)
'hostname' => 'localhost', // database host
'username' => '', // database username
'password' => '', // database password
'database' => '', // database name
'dbprefix' => 'shj_', // table prefix
/*****/
```

- Buat *application/cache/Twig* dapat ditulis oleh PHP
- Buka halaman utama *Sharif Judge* pada web *browser* dan ikuti proses instalasi berikutnya
- *Log in* menggunakan akun *admin*
- Pindahkan folder *tester* dan *assigments* di luar *public directory* lalu simpan *path* lengkap di halaman *Settings*. Dua folder tersebut harus dapat ditulis oleh PHP. File-file yang diserahkan akan disimpan di folder *assigments* sehingga tidak dapat diakses publik.

### 2.2.2 Clean URLs

Secara *default*, *index.php* merupakan bagian dari seluruh *urls* yang ada pada *Sharif Judge* seperti:

```
http://example.mjnaderi.ir/index.php/dashboard
http://example.mjnaderi.ir/index.php/users/add
```

Pengguna dapat menghilangkan *index.php* dan memiliki *urls* yang baik jika sistem pengguna mendukung aturan *rewrite* seperti:

```
http://example.mjnaderi.ir/dashboard
http://example.mjnaderi.ir/users/add
```

Untuk memungkinkan *clean urls*, ubah isi file *.htaccess2* menjadi *.htaccess* yang berlokasi di direktori utama *Sharif Judge*. Berikut isi file *.htaccess2*:

```
# You also need to change
# $config['index_page'] = 'index.php';
# to
# $config['index_page'] = '';
# in application/config/config.php
# in order to enable clean urls.

RewriteEngine on
RewriteCond $1 !^(index\.php|assets|robots\.txt)
RewriteRule ^(.*)$ index.php?/$1 [L]
```

Lalu buka file *application/config/config.php* dan ubah

```
$config['index_page'] = 'index.php';
```

menjadi

```
$config['index_page'] = '';
```

### 2.2.3 Users

Pada *Sharif Judge*, pengguna dikelompokkan menjadi 4 yaitu *Admins*, *Head Instructor*, *Instructor*, dan *Students* Tabel 2.1 menunjukan *level* para pengguna.

Tabel 2.1: User Roles Table	
User Role	User Level
<i>Admin</i>	3
<i>Head Instructor</i>	2
<i>Instructor</i>	1
<i>Student</i>	0

Setiap pengguna memiliki aksi yang berbeda-beda. Aksi yang dapat dilakukan para pengguna akan diseaikan dengan level masing-masing. Perhatikan tabel 2.2 berikut:

Tabel 2.2: *Permission Table*

Aksi	<i>Admin</i>	<i>Head Instructor</i>	<i>Instructor</i>	<i>Student</i>
Mengubah <i>Settings</i>	BISA	GB	GB	GB
Mengubah <i>Settings</i>	BISA	GB	GB	GB
Mengubah <i>Settings</i>	BISA	GB	GB	GB
Mengubah <i>Settings</i>	BISA	GB	GB	GB

Sed feugiat. Cum sociis natoque penatibus et magnis dis parturient montes, nascetur ridiculus mus. Ut pellentesque augue sed urna. Vestibulum diam eros, fringilla et, consectetuer eu, nonummy id, sapien. Nullam at lectus. In sagittis ultrices mauris. Curabitur malesuada erat sit amet massa. Fusce blandit. Aliquam erat volutpat. Aliquam euismod. Aenean vel lectus. Nunc imperdiet justo nec dolor.

Etiam euismod. Fusce facilisis lacinia dui. Suspendisse potenti. In mi erat, cursus id, nonummy sed, ullamcorper eget, sapien. Praesent pretium, magna in eleifend egestas, pede pede pretium lorem, quis consectetuer tortor sapien facilisis magna. Mauris quis magna varius nulla scelerisque imperdiet. Aliquam non quam. Aliquam porttitor quam a lacus. Praesent vel arcu ut tortor cursus volutpat. In vitae pede quis diam bibendum placerat. Fusce elementum convallis neque. Sed dolor orci, scelerisque ac, dapibus nec, ultricies ut, mi. Duis nec dui quis leo sagittis commodo.

## 2.3 L<sup>A</sup>T<sub>E</sub>X

Mengapa menggunakan L<sup>A</sup>T<sub>E</sub>X untuk buku skripsi dan apa keunggulan/kerugiannya bagi mahasiswa dan pembuat template.

Aliquam lectus. Vivamus leo. Quisque ornare tellus ullamcorper nulla. Mauris porttitor pharetra tortor. Sed fringilla justo sed mauris. Mauris tellus. Sed non leo. Nullam elementum, magna in cursus sodales, augue est scelerisque sapien, venenatis congue nulla arcu et pede. Ut suscipit enim vel sapien. Donec congue. Maecenas urna mi, suscipit in, placerat ut, vestibulum ut, massa. Fusce ultrices nulla et nisl.

Etiam ac leo a risus tristique nonummy. Donec dignissim tincidunt nulla. Vestibulum rhoncus molestie odio. Sed lobortis, justo et pretium lobortis, mauris turpis condimentum augue, nec ultricies nibh arcu pretium enim. Nunc purus neque, placerat id, imperdiet sed, pellentesque nec, nisl. Vestibulum imperdiet neque non sem accumsan laoreet. In hac habitasse platea dictumst. Etiam condimentum facilisis libero. Suspendisse in elit quis nisl aliquam dapibus. Pellentesque auctor sapien. Sed egestas sapien nec lectus. Pellentesque vel dui vel neque bibendum viverra. Aliquam porttitor nisl nec pede. Proin mattis libero vel turpis. Donec rutrum mauris et libero. Proin euismod porta felis. Nam lobortis, metus quis elementum commodo, nunc lectus elementum mauris, eget vulputate ligula tellus eu neque. Vivamus eu dolor.

## 2.4 Template Skripsi FTIS UNPAR

Akan dipaparkan bagaimana menggunakan template ini, termasuk petunjuk singkat membuat referensi, gambar dan tabel. Juga hal-hal lain yang belum terpikir sampai saat ini.

Nulla in ipsum. Praesent eros nulla, congue vitae, euismod ut, commodo a, wisi. Pellentesque habitant morbi tristique senectus et netus et malesuada fames ac turpis egestas. Aenean nonummy magna non leo. Sed felis erat, ullamcorper in, dictum non, ultricies ut, lectus. Proin vel arcu a odio lobortis euismod. Vestibulum ante ipsum primis in faucibus orci luctus et ultrices posuere cubilia Curae; Proin ut est. Aliquam odio. Pellentesque massa turpis, cursus eu, euismod nec,

tempor congue, nulla. Duis viverra gravida mauris. Cras tincidunt. Curabitur eros ligula, varius ut, pulvinar in, cursus faucibus, augue.

Nulla mattis luctus nulla. Duis commodo velit at leo. Aliquam vulputate magna et leo. Nam vestibulum ullamcorper leo. Vestibulum condimentum rutrum mauris. Donec id mauris. Morbi molestie justo et pede. Vivamus eget turpis sed nisl cursus tempor. Curabitur mollis sapien condimentum nunc. In wisi nisl, malesuada at, dignissim sit amet, lobortis in, odio. Aenean consequat arcu a ante. Pellentesque porta elit sit amet orci. Etiam at turpis nec elit ultricies imperdiet. Nulla facilisi. In hac habitasse platea dictumst. Suspendisse viverra aliquam risus. Nullam pede justo, molestie nonummy, scelerisque eu, facilisis vel, arcu.

### 2.4.1 Tabel

Berikut adalah contoh pembuatan tabel. Penempatan tabel dan gambar secara umum diatur secara otomatis oleh  $\text{\LaTeX}$ , perhatikan contoh di file bab2.tex untuk melihat bagaimana cara memaksa tabel ditempatkan sesuai keinginan kita.

Perhatikan bawa berbeda dengan penempatan judul gambar gambar, keterangan tabel harus diletakkan di atas tabel!! Lihat Tabel 2.3 berikut ini:

Tabel 2.3: Tabel contoh

	$v_{start}$	$\mathcal{S}_1$	$v_{end}$
$\tau_1$	1	12	20
$\tau_2$	1		20
$\tau_3$	1	9	20
$\tau_4$	1		20

Tabel 2.4 dan Tabel 2.5 berikut ini adalah tabel dengan sel yang berwarna dan ada dua tabel yang bersebelahan.

Tabel 2.4: Tabel bewarna(1)

	$v_{start}$	$\mathcal{S}_2$	$\mathcal{S}_1$	$v_{end}$
$\tau_1$	1	5	12	20
$\tau_2$	1	8		20
$\tau_3$	1	2/8/17	9	20
$\tau_4$	1			20

Tabel 2.5: Tabel bewarna(2)

	$v_{start}$	$\mathcal{S}_1$	$\mathcal{S}_2$	$v_{end}$
$\tau_1$	1	12	5	20
$\tau_2$	1		8	20
$\tau_3$	1	9	2/8/17	20
$\tau_4$	1			20

### 2.4.2 Kutipan

Berikut contoh kutipan dari berbagai sumber, untuk keterangan lebih lengkap, silahkan membaca file referensi.bib yang disediakan juga di template ini. Contoh kutipan:

- Buku: [4]
- Bab dalam buku: [5]
- Artikel dari Jurnal: [6]
- Artikel dari prosiding seminar/konferensi: [7]
- Skripsi/Thesis/Disertasi: [8] [9] [10]
- Technical/Scientific Report: [11]



- RFC (Request For Comments): [12]
- Technical Documentation/Technical Manual: [13] [14] [15]
- Paten: [16]
- Tidak dipublikasikan: [17] [18]
- Laman web: [19]
- Lain-lain: [20]

### 2.4.3 Gambar

Pada hampir semua editor, penempatan gambar di dalam dokumen L<sup>A</sup>T<sub>E</sub>X tidak dapat dilakukan melalui proses *drag and drop*. Perhatikan contoh pada file bab2.tex untuk melihat bagaimana cara menempatkan gambar. Beberapa hal yang harus diperhatikan pada saat menempatkan gambar:

- Setiap gambar **harus** diacu di dalam teks (gunakan *field* LABEL)
- *Field* CAPTION digunakan untuk teks pengantar pada gambar. Terdapat dua bagian yaitu yang ada di antara tanda [ dan ] dan yang ada di antara tanda { dan }. Yang pertama akan muncul di Daftar Gambar, sedangkan yang kedua akan muncul di teks pengantar gambar. Untuk skripsi ini, samakan isi keduanya.
- Jenis file yang dapat digunakan sebagai gambar cukup banyak, tetapi yang paling populer adalah tipe PNG (lihat Gambar 2.2), tipe JPG (Gambar 2.3) dan tipe PDF (Gambar 2.4)
- Besarnya gambar dapat diatur dengan *field* SCALE.
- Penempatan gambar diatur menggunakan *placement specifier* (di antara tanda [ dan ] setelah deklarasi gambar. Yang umum digunakan adalah **H** untuk menempatkan gambar **sesuai** penempatannya di file .tex atau **h** yang berarti "kira-kira" di sini.  
Jika tidak menggunakan *placement specifier*, L<sup>A</sup>T<sub>E</sub>X akan menempatkan gambar secara otomatis untuk menghindari bagian kosong pada dokumen anda. Walaupun cara ini sangat mudah, hindarkan terjadinya penempatan dua gambar secara berurutan.
  - Gambar 2.2 ditempatkan di bagian atas halaman, walaupun penempatannya dilakukan setelah penulisan 3 paragraf setelah penjelasan ini.
  - Gambar 2.3 dengan skala 0.5 ditempatkan di antara dua buah paragraf. Perhatikan penulisannya di dalam file bab2.tex!
  - Gambar 2.4 ditempatkan menggunakan *specifier* **h**.

Curabitur tellus magna, porttitor a, commodo a, commodo in, tortor. Donec interdum. Praesent scelerisque. Maecenas posuere sodales odio. Vivamus metus lacus, varius quis, imperdiet quis, rhoncus a, turpis. Etiam ligula arcu, elementum a, venenatis quis, sollicitudin sed, metus. Donec nunc pede, tincidunt in, venenatis vitae, faucibus vel, nibh. Pellentesque wisi. Nullam malesuada. Morbi ut tellus ut pede tincidunt porta. Lorem ipsum dolor sit amet, consectetur adipiscing elit. Etiam congue neque id dolor.

Donec et nisl at wisi luctus bibendum. Nam interdum tellus ac libero. Sed sem justo, laoreet vitae, fringilla at, adipiscing ut, nibh. Maecenas non sem quis tortor eleifend fermentum. Etiam id tortor ac mauris porta vulputate. Integer porta neque vitae massa. Maecenas tempus libero a libero posuere dictum. Vestibulum ante ipsum primis in faucibus orci luctus et ultrices posuere cubilia Curae; Aenean quis mauris sed elit commodo placerat. Class aptent taciti sociosqu ad litora torquent per conubia nostra, per inceptos hymenaeos. Vivamus rhoncus tincidunt libero. Etiam



Gambar 2.2: Gambar *Serpentes* dalam format png

elementum pretium justo. Vivamus est. Morbi a tellus eget pede tristique commodo. Nulla nisl. Vestibulum sed nisl eu sapien cursus rutrum.

Nulla non mauris vitae wisi posuere convallis. Sed eu nulla nec eros scelerisque pharetra. Nullam varius. Etiam dignissim elementum metus. Vestibulum faucibus, metus sit amet mattis rhoncus, sapien dui laoreet odio, nec ultricies nibh augue a enim. Fusce in ligula. Quisque at magna et nulla commodo consequat. Proin accumsan imperdiet sem. Nunc porta. Donec feugiat mi at justo. Phasellus facilisis ipsum quis ante. In ac elit eget ipsum pharetra faucibus. Maecenas viverra nulla in massa.

Nulla ac nisl. Nullam urna nulla, ullamcorper in, interdum sit amet, gravida ut, risus. Aenean ac enim. In luctus. Phasellus eu quam vitae turpis viverra pellentesque. Duis feugiat felis ut enim. Phasellus pharetra, sem id porttitor sodales, magna nunc aliquet nibh, nec blandit nisl mauris at pede. Suspendisse risus risus, lobortis eget, semper at, imperdiet sit amet, quam. Quisque scelerisque dapibus nibh. Nam enim. Lorem ipsum dolor sit amet, consectetur adipiscing elit. Nunc ut metus. Ut metus justo, auctor at, ultrices eu, sagittis ut, purus. Aliquam aliquam.



Gambar 2.3: Ular kecil

Etiam pede massa, dapibus vitae, rhoncus in, placerat posuere, odio. Vestibulum luctus commodo lacus. Morbi lacus dui, tempor sed, euismod eget, condimentum at, tortor. Phasellus aliquet odio ac lacus tempor faucibus. Praesent sed sem. Praesent iaculis. Cras rhoncus tellus sed justo ullamcorper sagittis. Donec quis orci. Sed ut tortor quis tellus euismod tincidunt. Suspendisse congue nisl eu elit. Aliquam tortor diam, tempus id, tristique eget, sodales vel, nulla. Praesent tellus mi, condimentum sed, viverra at, consectetur quis, lectus. In auctor vehicula orci. Sed pede sapien, euismod in, suscipit in, pharetra placerat, metus. Vivamus commodo dui non odio. Donec et felis.

Etiam suscipit aliquam arcu. Aliquam sit amet est ac purus bibendum congue. Sed in eros. Morbi non orci. Pellentesque mattis lacinia elit. Fusce molestie velit in ligula. Nullam et orci vitae

nibh vulputate auctor. Aliquam eget purus. Nulla auctor wisi sed ipsum. Morbi porttitor tellus ac enim. Fusce ornare. Proin ipsum enim, tincidunt in, ornare venenatis, molestie a, augue. Donec vel pede in lacus sagittis porta. Sed hendrerit ipsum quis nisl. Suspendisse quis massa ac nibh pretium cursus. Sed sodales. Nam eu neque quis pede dignissim ornare. Maecenas eu purus ac urna tincidunt congue.



Gambar 2.4: *Serpentes* jantan



## DAFTAR REFERENSI

- [1] Naderi, M. J. (2014) Sharif judge. <https://github.com/mjnaderi/Sharif-Judge/>. 6 Oktober 2017.
- [2] of Technology, B. C. I. (2017) Codeigniter documentation. [https://codeigniter.com/user\\_guide/overview/index.html](https://codeigniter.com/user_guide/overview/index.html). 6 Oktober 2017.
- [3] Naderi, M. J. (2014) Sharif judge documentation. <https://github.com/mjnaderi/Sharif-Judge/tree/docs/v1.4>. 6 Oktober 2017.
- [4] de Berg, M., Cheong, O., van Kreveld, M. J., dan Overmars, M. (2008) *Computational Geometry: Algorithms and Applications*, 3rd edition. Springer-Verlag, Berlin.
- [5] van Kreveld, M. J. (2004) Geographic information systems. Bagian dari Goodman, J. E. dan O'Rourke, J. (ed.), *Handbook of Discrete and Computational Geometry*. Chapman & Hall/CRC, Boca Raton.
- [6] Buchin, K., Buchin, M., van Kreveld, M. J., Löffler, M., Silveira, R. I., Wenk, C., dan Wiratma, L. (2013) Median trajectories. *Algorithmica*, **66**, 595–614.
- [7] van Kreveld, M. J. dan Wiratma, L. (2011) Median trajectories using well-visited regions and shortest paths. *Proceedings of the 19th ACM SIGSPATIAL International Conference on Advances in Geographic Information Systems*, Chicago, USA, 1-4 November, pp. 241–250. ACM, New York.
- [8] Lionov (2002) Animasi algoritma sweepline untuk membangun diagram voronoi. Skripsi. Universitas Katolik Parahyangan, Indonesia.
- [9] Wiratma, L. (2010) Following the majority: a new algorithm for computing a median trajectory. Thesis. Utrecht University, The Netherlands.
- [10] Wiratma, L. (2022) Coming Not Too Soon, Later, Delay, Someday, Hopefully. Disertasi. Utrecht University, The Netherlands.
- [11] van kreveld, M., van Lankveld, T., dan Velthkamp, R. (2013) Watertight scenes from urban lidar and planar surfaces. Technical Report UU-CS-2013-007. Utrecht University, The Netherlands.
- [12] Rekhter, Y. dan Li, T. (1994) A border gateway protocol 4 (bgp-4). RFC 1654. RFC Editor, <http://www.rfc-editor.org>.
- [13] ITU-T Z.500 (1997) *Framework on formal methods in conformance testing*. International Telecommunications Union. Geneva, Switzerland.
- [14] Version 9.0.0 (2016) *The Unicode Standard*. The Unicode Consortium. Mountain View, USA.
- [15] Version 7.0 Nougat (2016) *Android API Reference Manual*. Google dan Open Handset Alliance. Mountain View, USA.

- [16] Webb, R., Daruca, O., dan Alfadian, P. (2012) *Method of optimizing a text message communication between a server and a secure element*. Paten no. EP2479956 (A1). European Patent Organisation. Munich, Germany.
- [17] Wiratma, L. (2009) Median trajectory. Report for GMT Experimentation Project at Utrecht University.
- [18] Lionov (2011) Polymorphism pada C++. Catatan kuliah AKS341 Pemrograman Sistem di Universitas Katolik Parahyangan, Bandung. <http://tinyurl.com/lionov>. 30 September 2016.
- [19] Erickson, J. (2003) CG models of computation? <http://www.computational-geometry.org/mailling-lists/compgeom-announce/2003-December/000852.html>. 30 September 2016.
- [20] AGUNG (2012) Menjajal tango 12. Majalah HAI no 02, Januari 2012.

# LAMPIRAN A

## KODE PROGRAM

Listing A.1: MyCode.c

```
1 // This does not make algorithmic sense,
2 // but it shows off significant programming characters.
3
4 #include<stdio.h>
5
6 void myFunction( int input, float* output ) {
7     switch ( array[i] ) {
8         case 1: // This is silly code
9             if ( a >= 0 || b <= 3 && c != x )
10                 *output += 0.005 + 20050;
11             char = 'g';
12             b = 2^n + ~right_size - leftSize * MAX_SIZE;
13             c = (--aaa + &daa) / (bbb++ - ccc % 2 );
14             strcpy(a,"hello_$@?");
15         }
16         count = ~mask | 0x00FF00AA;
17     }
18 }
19
20 // Fonts for Displaying Program Code in LATEX
21 // Adrian P. Robson, nepsweb.co.uk
22 // 8 October 2012
23 // http://nepsweb.co.uk/docs/progfonts.pdf
```

Listing A.2: MyCode.java

```
1 import java.util.ArrayList;
2 import java.util.Collections;
3 import java.util.HashSet;
4
5 //class for set of vertices close to furthest edge
6 public class MyFurSet {
7     protected int id; //id of the set
8     protected MyEdge FurthestEdge; //the furthest edge
9     protected HashSet<MyVertex> set; //set of vertices close to furthest edge
10    protected ArrayList<ArrayList<Integer>> ordered; //list of all vertices in the set for each trajectory
11    protected ArrayList<Integer> closeID; //store the ID of all vertices
12    protected ArrayList<Double> closeDist; //store the distance of all vertices
13    protected int totaltrj; //total trajectories in the set
14
15    /*
16     * Constructor
17     * @param id : id of the set
18     * @param totaltrj : total number of trajectories in the set
19     * @param FurthestEdge : the furthest edge
20     */
21    public MyFurSet(int id,int totaltrj,MyEdge FurthestEdge) {
22        this.id = id;
23        this.totaltrj = totaltrj;
24        this.FurthestEdge = FurthestEdge;
25        set = new HashSet<MyVertex>();
26        ordered = new ArrayList<ArrayList<Integer>>();
27        for (int i=0;i<totaltrj;i++) ordered.add(new ArrayList<Integer>());
28        closeID = new ArrayList<Integer>(totaltrj);
29        closeDist = new ArrayList<Double>(totaltrj);
30        for (int i = 0;i <totaltrj;i++) {
31            closeID.add(-1);
32            closeDist.add(Double.MAX_VALUE);
33        }
34    }
35
36 }
```





## LAMPIRAN B

### HASIL EKSPERIMEN

Hasil eksperimen berikut dibuat dengan menggunakan TIKZPICTURE (bukan hasil excel yg diubah ke file bitmap). Sangat berguna jika ingin menampilkan tabel (yang kuantitasnya sangat banyak) yang datanya dihasilkan dari program komputer.



Gambar B.1: Hasil 1



Gambar B.2: Hasil 2



Gambar B.3: Hasil 3



Gambar B.4: Hasil 4