

# SKRIPSI

## KEPATUHAN DAN REKOMENDASI PERBAIKAN WEB CONTENT ACCESSIBILITY GUIDELINE UNTUK APLIKASI SHARIF JUDGE



Amabel Levint

NPM: 2016730013

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

«tahun»



# UNDERGRADUATE THESIS

## CONFORMANCE AND IMPROVEMENT RECOMMENDATIONS OF WEB CONTENT ACCESSIBILITY GUIDELINES FOR SHARIF JUDGE APPLICATION



Amabel Levint

NPM: 2016730013

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

«tahun»



# LEMBAR PENGESAHAN

## KEPATUHAN DAN REKOMENDASI PERBAIKAN WEB CONTENT ACCESSIBILITY GUIDELINE UNTUK APLIKASI SHARIF JUDGE

Amabel Levint

NPM: 2016730013

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:

### **KEPATUHAN DAN REKOMENDASI PERBAIKAN WEB CONTENT ACCESSIBILITY GUIDELINE UNTUK APLIKASI SHARIF JUDGE**

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

Amabel Levint  
NPM: 2016730013





## 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

<b>KATA PENGANTAR</b>	<b>xv</b>
<b>DAFTAR ISI</b>	<b>xvii</b>
<b>DAFTAR GAMBAR</b>	<b>xxi</b>
<b>DAFTAR TABEL</b>	<b>xxiii</b>
<b>1 PENDAHULUAN</b>	<b>1</b>
1.1 Latar Belakang . . . . .	1
1.2 Rumusan Masalah . . . . .	1
1.3 Tujuan . . . . .	2
1.4 Batasan Masalah . . . . .	2
1.5 Metodologi . . . . .	2
1.6 Sistematika Pembahasan . . . . .	2
<b>2 LANDASAN TEORI</b>	<b>3</b>
2.1 WCAG 2.1 . . . . .	3
2.1.1 Kriteria Sukses 1.1.1 Non-text Content . . . . .	4
2.1.2 Kriteria Sukses 1.2.1 Audio-only dan Video-only (Prerecorded) . . . . .	4
2.1.3 Kriteria Sukses 1.2.2 Captions (Prerecorded) . . . . .	5
2.1.4 Kriteria Sukses 1.2.3 Audio Descriptive atau Media Alternative (Prerecorded)	5
2.1.5 Kriteria Sukses 1.2.4 Captions (Live) . . . . .	5
2.1.6 Kriteria Sukses 1.2.5 Audio Description (Prerecorded) . . . . .	5
2.1.7 Kriteria Sukses 1.2.6 Sign Language (Prerecorded) . . . . .	5
2.1.8 Kriteria Sukses 1.2.7 Extended Audio Description (Prerecorded) . . . . .	5
2.1.9 Kriteria Sukses 1.2.8 Media Alternative (Prerecorded) . . . . .	5
2.1.10 Kriteria Sukses 1.2.9 Audio-only (Live) . . . . .	6
2.1.11 Kriteria Sukses 1.3.1 Info dan Relationships . . . . .	6
2.1.12 Kriteria Sukses 1.3.2 Meaningful Sequence . . . . .	6
2.1.13 Kriteria Sukses 1.3.3 Sensory Characteristics . . . . .	6
2.1.14 Kriteria Sukses 1.3.4 Orientation . . . . .	6
2.1.15 Kriteria Sukses 1.3.5 Identify Input Purpose . . . . .	6
2.1.16 Kriteria Sukses 1.3.6 Identify Purpose . . . . .	6
2.1.17 Kriteria Sukses 1.4.1 Use of Color . . . . .	7
2.1.18 Kriteria Sukses 1.4.2 Audio Control . . . . .	7
2.1.19 Kriteria Sukses 1.4.3 Contrast (Minimum) . . . . .	7
2.1.20 Kriteria Sukses 1.4.4 Resize text . . . . .	7
2.1.21 Kriteria Sukses 1.4.5 Images of Text . . . . .	7
2.1.22 Kriteria Sukses 1.4.6 Contrast (Enhanced) . . . . .	7
2.1.23 Kriteria Sukses 1.4.7 Low atau No Background Audio . . . . .	8
2.1.24 Kriteria Sukses 1.4.8 Visual Presentation . . . . .	8
2.1.25 Kriteria Sukses 1.4.9 Images of Text (No Exception) . . . . .	8

2.1.26	Kriteria Sukses	1.4.10 Reflow	8
2.1.27	Kriteria Sukses	1.4.11 Non-text Contrast	9
2.1.28	Kriteria Sukses	1.4.12 Text Spacing	9
2.1.29	Kriteria Sukses	1.4.13 Content on Hover or Focus	9
2.1.30	Kriteria Sukses	2.1.1 Keyboard	9
2.1.31	Kriteria Sukses	2.1.2 No Keyboard Trap	10
2.1.32	Kriteria Sukses	2.1.3 Keyboard (No Exception)	10
2.1.33	Kriteria Sukses	2.1.4 Character Key Shortcuts	10
2.1.34	Kriteria Sukses	2.2.1 Timing Adjustable	10
2.1.35	Kriteria Sukses	2.2.2 Pause, Stop, Hide	11
2.1.36	Kriteria Sukses	2.2.3 No Timing	11
2.1.37	Kriteria Sukses	2.2.4 Interruptions	11
2.1.38	Kriteria Sukses	2.2.5 Re-authenticating	11
2.1.39	Kriteria Sukses	2.2.6 Timeouts	11
2.1.40	Kriteria Sukses	2.3.1 Three Flashes or Below Threshold	11
2.1.41	Kriteria Sukses	2.3.2 Three Flashes	12
2.1.42	Kriteria Sukses	2.3.3 Animation from Interactions	12
2.1.43	Kriteria Sukses	2.4.1 Bypass Blocks	12
2.1.44	Kriteria Sukses	2.4.2 Page Titled	12
2.1.45	Kriteria Sukses	2.4.3 Focus Order	12
2.1.46	Kriteria Sukses	2.4.4 Link Purpose (In Context)	12
2.1.47	Kriteria Sukses	2.4.5 Multiple Ways	12
2.1.48	Kriteria Sukses	2.4.6 Headings and Labels	12
2.1.49	Kriteria Sukses	2.4.7 Focus Visible	13
2.1.50	Kriteria Sukses	2.4.8 Location	13
2.1.51	Kriteria Sukses	2.4.9 Link Purpose (Link Only)	13
2.1.52	Kriteria Sukses	2.4.10 Section Headings	13
2.1.53	Kriteria Sukses	2.5.1 Pointer Gestures	13
2.1.54	Kriteria Sukses	2.5.2 Pointer Cancellation	13
2.1.55	Kriteria Sukses	2.5.3 Label in Name	14
2.1.56	Kriteria Sukses	2.5.4 Motion Actuation	14
2.1.57	Kriteria Sukses	2.5.5 Target Size	14
2.1.58	Kriteria Sukses	2.5.6 Concurrent Input Mechanisms	14
2.1.59	Kriteria Sukses	3.1.1 Language of Page	14
2.1.60	Kriteria Sukses	3.1.2 Language of Parts	15
2.1.61	Kriteria Sukses	3.1.3 Unusual Words	15
2.1.62	Kriteria Sukses	3.1.4 Abbreviations	15
2.1.63	Kriteria Sukses	3.1.5 Reading Level	15
2.1.64	Kriteria Sukses	3.1.6 Pronunciation	15
2.1.65	Kriteria Sukses	3.2.1 On Focus	15
2.1.66	Kriteria Sukses	3.2.2 On Input	15
2.1.67	Kriteria Sukses	3.2.3 Consistent Navigation	16
2.1.68	Kriteria Sukses	3.2.4 Consistent Identification	16
2.1.69	Kriteria Sukses	3.2.5 Change on Request	16
2.1.70	Kriteria Sukses	3.3.1 Error Identification	16
2.1.71	Kriteria Sukses	3.3.2 Labels or Instructions	16
2.1.72	Kriteria Sukses	3.3.3 Error Suggestion	16
2.1.73	Kriteria Sukses	3.3.4 Error Prevention (Legal, Financial, Data)	16
2.1.74	Kriteria Sukses	3.3.5 Help	17
2.1.75	Kriteria Sukses	3.3.6 Error Prevention (All)	17
2.1.76	Kriteria Sukses	4.1.1 Parsing	17

2.1.77	Kriteria Sukses 4.1.2 Name, Role, Value	17
2.1.78	Kriteria Sukses 4.1.3 Status Messages	17
2.2	SharIF Judge	18
2.2.1	Instalasi	18
2.2.2	<i>Clean URLs</i>	19
2.2.3	Users	20
2.2.4	Menambah Assignment	21
2.2.5	<i>Sample Assignment</i>	24
2.2.6	<i>Test Structure</i>	25
2.2.7	Deteksi Kecurangan	28
2.2.8	Keamanan	29
2.2.9	<i>Sandbox</i>	31
2.2.10	<i>Shield</i>	31
<b>3</b>	<b>ANALISIS</b>	<b>35</b>
3.1	Tingkat Kepatuhan <i>SharIF Judge</i>	35
3.1.1	<i>Perceivable</i>	37
3.1.2	<i>Operable</i>	51
3.1.3	<i>Understandable</i>	62
3.1.4	<i>Robust</i>	66
3.2	Peningkatan ke <i>Level A</i>	67
<b>4</b>	<b>IMPLEMENTASI</b>	<b>69</b>
4.1	Implementasi Ke <i>Level A</i>	69
4.1.1	Implementasi Kriteria Sukses 1.1.1 Non-text Content	69
4.1.2	Implementasi Kriteria Sukses 1.3.1 Info dan Relationships	70
4.1.3	Implementasi Kriteria Sukses 2.1.1 Keyboard	73
4.1.4	Implementasi Kriteria Sukses 2.1.2 No Keyboard Trap	73
4.1.5	Implementasi Kriteria Sukses 2.4.1 Bypass Blocks	73
4.1.6	Implementasi Kriteria Sukses 2.4.4 Link Purpose (In Context)	73
4.1.7	Implementasi Kriteria Sukses 3.1.1 Language of Page	76
4.1.8	Implementasi Kriteria Sukses 3.3.2 Labels or Instructions	77
4.1.9	Implementasi Kriteria Sukses 4.1.1 Parsing	77
<b>A</b>	<b>KODE PROGRAM</b>	<b>79</b>
<b>B</b>	<b>HASIL EKSPERIMEN</b>	<b>81</b>



## DAFTAR GAMBAR

2.1	Tampilan Halaman <i>Assignments</i> . . . . .	21
3.1	Kriteria Sukses 1.3.1 - Elemen yang Tidak Diberi Label Pada Halaman <i>Add Users</i>	39
3.2	Kriteria Sukses 1.3.1 - Elemen yang Tidak Diberi Label Pada Halaman <i>Problems</i> Bagian <i>Edit Markdown</i> . . . . .	40
3.3	Kriteria Sukses 1.3.1 - Elemen yang Tidak Diberi Label Pada Halaman <i>Problems</i> Bagian <i>Edit Plain Html</i> . . . . .	40
3.4	Kriteria Sukses 1.4.3 - Kontras Elemen Navigasi Atas . . . . .	42
3.5	Kriteria Sukses 1.4.3 - Kontras Elemen Dashboard . . . . .	43
3.6	Kriteria Sukses 1.4.3 - Kontras Elemen Settings . . . . .	43
3.7	Kriteria Sukses 1.4.3 - Kontras Elemen Notifications . . . . .	44
3.8	Kriteria Sukses 1.4.3 - Kontras Elemen Assignments . . . . .	44
3.9	Kriteria Sukses 1.4.3 - Kontras Elemen <i>Add Assignments</i> . . . . .	45
3.10	Kriteria Sukses 1.4.4 - Sidenav . . . . .	46
3.11	Kriteria Sukses 1.4.6 - Kontras Elemen Navigasi Atas . . . . .	47
3.12	Kriteria Sukses 1.4.6 - Kontras Elemen Dashboard . . . . .	47
3.13	Kriteria Sukses 1.4.6 - Kontras Elemen Settings . . . . .	48
3.14	Kriteria Sukses 1.4.6 - Kontras Elemen Notifications . . . . .	48
3.15	Kriteria Sukses 1.4.6 - Kontras Elemen Assignments . . . . .	49
3.16	Kriteria Sukses 1.4.6 - Kontras Elemen Add Assignments . . . . .	49
3.17	Kriteria Sukses 1.4.8 - Notifications . . . . .	50
3.18	Kriteria Sukses 1.4.10 - Horizontal <i>Scroll</i> . . . . .	50
3.19	Kriteria Sukses 1.4.11 - Ikon Pada Halaman <i>Assignments</i> . . . . .	51
3.20	Kriteria Sukses 1.4.12 - Halaman <i>Add Assignment</i> . . . . .	51
3.21	Kriteria Sukses 2.1.1 - <i>Collapse Menu</i> . . . . .	52
3.22	Kriteria Sukses 2.1.1 - <i>Tools</i> . . . . .	52
3.23	Kriteria Sukses 2.1.2 - <i>Settings</i> . . . . .	53
3.24	Kriteria Sukses 2.1.2 - <i>Add User</i> . . . . .	53
3.25	Kriteria Sukses 2.1.2 - <i>Add Assignment</i> . . . . .	54
3.26	Kriteria Sukses 2.1.2 - <i>Problems</i> Bagian <i>Edit Markdown</i> . . . . .	54
3.27	Kriteria Sukses 2.1.3 - <i>Collapse Menu</i> . . . . .	55
3.28	Kriteria Sukses 2.1.3 - <i>Tools</i> . . . . .	55
3.29	Kriteria Sukses 2.4.5 - Menemukan Halaman Web Pada Navigasi . . . . .	59
3.30	Kriteria Sukses 2.5.5 - Bidang Masukan Berukuran 30 CSS Piksel . . . . .	62
3.31	Kriteria Sukses 3.2.4 - <i>Assignments</i> . . . . .	64
3.32	Kriteria Sukses 3.2.4 - <i>Users</i> . . . . .	64
3.33	Kriteria Sukses 3.2.4 - <i>Notifications</i> . . . . .	65
3.34	Kriteria Sukses 3.3.2 - <i>Edit Markdown</i> Tidak Diberi Label Atau Instruksi . . . . .	65
B.1	Hasil 1 . . . . .	81
B.2	Hasil 2 . . . . .	81
B.3	Hasil 3 . . . . .	81
B.4	Hasil 4 . . . . .	81



## DAFTAR TABEL

2.1	<i>User Roles Table</i> . . . . .	20
2.2	<i>Permission Table</i> . . . . .	20
2.3	<i>Problem 1 (Penjumlahan)</i> . . . . .	24
2.4	<i>Problem 2 (Max)</i> . . . . .	24
3.1	Tabel kepatuhan <i>SharIF Judge</i> terhadap prinsip <i>Perceivable</i> . . . . .	35
3.2	Tabel kepatuhan <i>SharIF Judge</i> terhadap prinsip <i>Operable</i> . . . . .	36
3.3	Tabel kepatuhan <i>SharIF Judge</i> terhadap prinsip <i>Understandable</i> . . . . .	37
3.4	Tabel kepatuhan <i>SharIF Judge</i> terhadap prinsip <i>Robust</i> . . . . .	37





# BAB 1

## PENDAHULUAN

### 1.1 Latar Belakang

*SharIF Judge* [?] adalah sebuah aplikasi gratis dan *open source* untuk menilai code berbahasa *C* , *C++*, *Java* dan *Python*. *SharIF Judge* adalah pencabangan dari *Sharif Judge* yang telah dibuat oleh Mohammed Javad Naderi. Versi dari pencabangan ini memuat fitur baru yang diperlukan oleh jurusan teknik informatika UNPAR. Aplikasi ini dibuat menggunakan PHP (*CodeIgnitor framework*) dan bagian backendnya dibuat dengan BASH.

Pada zaman perkembangan teknologi saat ini, banyak sekali orang yang mengakses teknologi. Beberapa diantara mereka memiliki disabilitas. Sejumlah individu dan organisasi dari berbagai negara bekerja sama membuat standar untuk *Web content accessibility* yang memenuhi kebutuhan individu, organisasi, dan pemerintah secara internasional. Melalui perundingan tersebut terbentuklah *Web Content Accessibility Guidelines* (*WCAG*) 2.1 [?] yang memuat rekomendasi untuk membuat konten web lebih mudah diakses. Pedoman-pedoman ini akan membuat konten lebih mudah diakses untuk orang disabilitas termasuk akomodasi untuk kebutaan dan penglihatan rendah, ketulian dan gangguan pendengaran, gerakan terbatas, fotosensitif, atau kombinasinya, dan beberapa akomodasi untuk kesulitan belajar dan keterbatasan kognitif; tetapi tidak akan memenuhi setiap kebutuhan pengguna dengan disabilitas.

Di dalam *WCAG* 2.1 ada 78 kriteria sukses. Kriteria sukses adalah pedoman untuk membuat konten lebih mudah diakses. Ada 3 tingkat kepatuhan yaitu A (terkecil), AA, AAA (terbesar). Tingkat kepatuhan A adalah tingkat kepatuhan terkecil yang diperoleh jika seluruh kriteria sukses tingkat A terpenuhi atau versi alternatifnya tersedia. Tingkat kepatuhan AA adalah tingkat kepatuhan yang diperoleh jika seluruh kriteria sukses tingkat A dan AA terpenuhi atau versi alternatif tingkat AA tersedia. Tingkat kepatuhan AAA adalah tingkat kepatuhan yang diperoleh jika seluruh kriteria sukses tingkat A, AA, dan AAA terpenuhi atau versi alternatif tingkat AAA tersedia.

Pada skripsi ini, akan dilakukan analisis tingkat kepatuhan dan rekomendasi perbaikan aplikasi *SharIF Judge* berdasarkan *Web Content Accessibility Guideline* 2.1. Selain itu, aplikasi *SharIF Judge* juga akan diuji dengan beberapa kondisi keterbatasan seperti keterbatasan visual, keterbatasan gerak, keterbatasan pendengaran. Dengan perbaikan ini diharapkan aplikasi *SharIF Judge* dapat diakses oleh banyak kalangan.

### 1.2 Rumusan Masalah

- Bagaimana tingkat kepatuhan *SharIF Judge* terhadap *WCAG* 2.1 ?
- Rekomendasi perbaikan apa saja yang perlu dilakukan terhadap *SharIF Judge* untuk menaikkan tingkat kepatuhannya ?

### 1.3 Tujuan

- Mengetahui tingkat kepatuhan *SharIF Judge* terhadap *WCAG 2.1*.
- Membuat rekomendasi perbaikan yang perlu dilakukan terhadap *SharIF Judge* untuk menaikkan tingkat kepatuhannya.

### 1.4 Batasan Masalah

Batasan masalah pada skripsi ini adalah sebagai berikut:

1. content

### 1.5 Metodologi

Metodologi yang dilakukan pada skripsi ini adalah sebagai berikut:

1. Studi literatur mengenai *WCAG 2.1* dan *SharIF Judge*
2. Mengukur tingkat kepatuhan *SharIF Judge* terhadap *WCAG 2.1*
3. Memberikan rekomendasi perbaikan pada setiap kriteria kesuksesan.
4. Mengimplementasikan rekomendasi perbaikan.
5. Menguji hasil perbaikan.

### 1.6 Sistematika Pembahasan

Setiap bab dalam skripsi ini memiliki sistematika penulisan ke dalam poin-poin sebagai berikut:

1. Bab 1: Pendahuluan, akan membahas gambaran umum dari skripsi ini. Bab ini berisi latar belakang, rumusan masalah, tujuan, batasan masalah, metode penelitian, dan sistematika pembahasan.
2. Bab 2: Landasan Teori, akan membahas dasar teori yang menjadi acuan dalam pembuatan skripsi ini. Dasar teori yang digunakan yaitu *WCAG 2.1* dan *SharIF Judge*.
3. Bab 3: Analisis, akan membahas hasil analisis mengenai tingkat kepatuhan situs web *SharIF Judge* terhadap *WCAG 2.1*.
4. Bab 4: Perancangan, akan membahas mengenai perubahan-perubahan yang dapat dilakukan untuk meningkatkan kepatuhan situs web *SharIF Judge* terhadap *WCAG 2.1*.
5. Bab 5: Implementasi dan Pengujian, akan membahas hasil implementasi dan pengujian yang telah dilakukan pada situs web *SharIF Judge*.
6. Bab 6: Kesimpulan dan saran, akan berisi kesimpulan dari hasil penelitian yang telah dilakukan dan saran yang dapat diberikan untuk penelitian berikutnya.

## BAB 2

### LANDASAN TEORI

#### 2.1 WCAG 2.1

*Web Content Accessibility Guidelines (WCAG) 2.1* memuat rekomendasi untuk membuat konten web lebih mudah diakses. Pedoman-pedoman ini akan membuat konten lebih mudah diakses untuk orang disabilitas termasuk akomodasi untuk kebutaan dan penglihatan rendah, ketulian dan gangguan pendengaran, gerakan terbatas, fotosensitif, atau kombinasinya, dan beberapa akomodasi untuk kesulitan belajar dan keterbatasan kognitif; tetapi tidak akan memenuhi setiap kebutuhan pengguna dengan disabilitas. *WCAG* dikembangkan oleh *World Wide Web Consortium* melalui kerja sama dengan individu dan organisasi di seluruh dunia dengan tujuan memberikan standar bersama untuk aksesibilitas konten web yang memenuhi kebutuhan individu, organisasi, dan pemerintah internasional. *WCAG 2.1* merupakan pembaruan dari *WCAG 2.0* yang dibuat pada 11 Desember 2008. Ada 78 kriteria sukses dalam *WCAG 2.1*. Kriteria sukses adalah pedoman untuk membuat konten lebih mudah diakses. Kriteria Sukses *WCAG 2.1* ditulis sebagai pernyataan yang dapat diuji yang tidak teknologi spesifik. Pedoman ini mencakup aksesibilitas konten web di desktop, laptop, tablet, dan perangkat bergerak. Dengan mengikuti pedoman ini juga akan sering membuat konten web lebih bermanfaat bagi pengguna secara umum.

Ada beberapa kondisi yang harus dipenuhi untuk sebuah Kriteria Sukses yaitu:

1. Semua Kriteria Sukses harus menjadi masalah akses penting bagi orang disabilitas yang mengatasi masalah di luar masalah kegunaan yang dihadapi oleh semua pengguna. Dengan kata lain, masalah akses harus menyebabkan masalah yang lebih besar bagi orang disabilitas daripada orang yang tidak disabilitas agar dianggap sebagai masalah aksesibilitas.
2. Semua Kriteria Sukses harus dapat diuji. Hal ini penting karena jika tidak, maka tidak mungkin untuk menentukan apakah suatu halaman memenuhi Kriteria Sukses. Kriteria Sukses dapat diuji dengan kombinasi evaluasi mesin dan manusia selama pengujian dapat menentukan apakah sebuah Kriteria Sukses terpenuhi dengan tingkat kepercayaan yang tinggi.

Kriteria Sukses memiliki tiga tingkat kesesuaian yaitu tingkat A (terkecil), AA, AAA (terbesar). Ada beberapa faktor yang menentukan tingkat tersebut. Faktor tersebut termasuk:

1. Apakah Kriteria Sukses esensial (dalam kata lain, jika Kriteria Sukses tidak terpenuhi maka teknologi bantuan juga tidak dapat membuat konten dapat diakses).
2. Apakah mungkin untuk memenuhi Kriteria Sukses untuk semua situs web dan jenis konten yang akan diterapkan Kriteria Sukses.
3. Apakah Kriteria Sukses membutuhkan keterampilan yang dapat dicapai secara wajar oleh pembuat konten (Pengetahuan dan keterampilan untuk memenuhi Kriteria Sukses dapat diperoleh dalam pelatihan seminggu atau kurang).
4. Apakah Kriteria Sukses dapat memaksakan batasan tampilan dan fungsi dari halaman web (batasan dari fungsi, presentasi, kebebasan berekspresi, desain atau estetika)

## 5. Apakah tidak ada solusi jika Kriteria Sukses tidak terpenuhi

Kriteria sukses dibagi menjadi 4 berdasarkan prinsip *accessibility*, yaitu:

- *Perceivable*: Informasi dan antarmuka pengguna harus ditampilkan kepada pengguna dengan cara yang dapat dipahami.
- *Operable*: Komponen antarmuka pengguna dan navigasi harus dapat dioperasikan.
- *Understandable*: Informasi dan pengoperasian antarmuka pengguna harus dapat dimengerti.
- *Robust*: Konten harus cukup handal sehingga dapat ditafsirkan oleh berbagai agen pengguna, termasuk teknologi alat bantu.

Berikut adalah uraian kriteria sukses *WCAG 2.1*:

### 2.1.1 Kriteria Sukses 1.1.1 Non-text Content

(Level A)

Semua konten bukan teks yang ditampilkan ke pengguna memiliki teks alternatif yang tujuannya sama. Kecuali pada kondisi sebagai berikut:

- Kontrol, dan masukan: Jika konten bukan teks adalah kontrol atau masukan user maka konten tersebut harus memiliki nama yang menjelaskan tujuannya.
- Media berbasis waktu: Jika konten bukan teks adalah media berbasis waktu, maka setidaknya disediakan alternatif berupa teks untuk identifikasi deskriptif dari konten bukan teks.
- Tes: Jika konten bukan teks adalah tes atau latihan yang tidak valid jika ditampilkan dalam teks, maka setidaknya disediakan teks alternatif untuk identifikasi deskriptif dari konten bukan teks.
- Indra: Jika konten bukan teks digunakan untuk menciptakan pengalaman indra tertentu, maka setidaknya disediakan teks alternatif untuk identifikasi deskriptif dari konten bukan teks.
- *CAPTCHA*: Jika tujuan dari konten bukan teks digunakan untuk memastikan apakah konten tersebut diakses oleh manusia dan bukannya komputer, maka disediakan teks alternatif untuk mengidentifikasi dan menjelaskan tujuan dari konten bukan teks, dan disediakan bentuk alternatif dari *CAPTCHA* menggunakan mode keluaran untuk berbagai jenis persepsi indra untuk mengakomodasi berbagai disabilitas.
- Dekorasi, pemformatan, tak kentara: Jika konten bukan teks adalah dekorasi saja, digunakan untuk pemformatan, atau tidak ditampilkan kepada pengguna, maka konten tersebut diterapkan dengan cara yang dapat diacuhkan oleh teknologi bantuan.

### 2.1.2 Kriteria Sukses 1.2.1 Audio-only dan Video-only (Prerecorded)

(Level A)

Untuk rekaman audio saja dan rekaman video saja, berikut ini benar, kecuali ketika audio atau video adalah media alternatif untuk teks dan diberi label dengan jelas:

- Rekaman audio saja: Tersedia alternatif untuk media berbasis waktu yang isinya mewakili informasi yang sama dengan konten rekaman audio saja.
- Rekaman video saja: Tersedia alternatif untuk media berbasis waktu atau trek audio yang isinya mewakili informasi yang sama dengan konten rekaman video saja.

### **2.1.3 Kriteria Sukses 1.2.2 Captions (Prerecorded)**

(Level A)

Caption disediakan untuk semua konten rekaman audio di media berbasis waktu kecuali medianya adalah media alternatif untuk teks dan diberi label dengan jelas.

### **2.1.4 Kriteria Sukses 1.2.3 Audio Descriptive atau Media Alternative (Prerecorded)**

(Level A)

Tersedianya alternatif untuk media berbasis waktu atau deskripsi audio dari konten rekaman video untuk media yang disinkronkan, kecuali medianya adalah media alternatif untuk teks dan diberi label dengan jelas.

### **2.1.5 Kriteria Sukses 1.2.4 Captions (Live)**

(Level AA)

Keterangan tersedia untuk semua konten audio yang disiarkan langsung di media yang disinkronkan.

### **2.1.6 Kriteria Sukses 1.2.5 Audio Description (Prerecorded)**

(Level AA)

Deskripsi audio disediakan untuk semua konten rekaman video di media yang disinkronkan.

### **2.1.7 Kriteria Sukses 1.2.6 Sign Language (Prerecorded)**

(Level AAA)

Intepretasi bahasa isyarat disediakan untuk semua konten rekaman audio di media yang disinkronkan.

### **2.1.8 Kriteria Sukses 1.2.7 Extended Audio Description (Prerecorded)**

(Level AAA)

Ketika keheningan di audio tidak memadai untuk menyampaikan maksud video tersebut, deskripsi audio tambahan disediakan untuk semua konten rekaman video pada media yang disinkronkan.

### **2.1.9 Kriteria Sukses 1.2.8 Media Alternative (Prerecorded)**

(Level AAA)

Tersedia alternatif untuk media berbasis waktu untuk semua rekaman media yang disinkronkan dan untuk semua rekaman media video saja.

### **2.1.10 Kriteria Sukses 1.2.9 Audio-only (Live)**

(Level AAA)

Tersedia alternatif untuk media berbasis waktu yang menampilkan informasi yang setara untuk konten siaran langsung audio saja.

### **2.1.11 Kriteria Sukses 1.3.1 Info dan Relationships**

(Level A)

Informasi , struktur, dan hubungan yang ditampilkan melalui presentasi dapat ditentukan secara pemrograman atau tersedia dalam teks.

### **2.1.12 Kriteria Sukses 1.3.2 Meaningful Sequence**

(Level A)

Ketika urutan konten disajikan memengaruhi maknanya, urutan bacaan yang benar dapat ditentukan secara pemrograman.

### **2.1.13 Kriteria Sukses 1.3.3 Sensory Characteristics**

(Level A)

Petunjuk yang diberikan untuk memahami dan mengoperasikan konten tidak hanya bergantung pada komponen karakteristik sensorik seperti bentuk, warna, ukuran, lokasi visual, orientasi, atau suara.

### **2.1.14 Kriteria Sukses 1.3.4 Orientation**

(Level AA)

Tampilan dan pengoperasian konten tidak bergantung pada satu orientasi tampilan , seperti *portrait* atau *landscape*, kecuali jika orientasi tampilan tertentu esensial.

### **2.1.15 Kriteria Sukses 1.3.5 Identify Input Purpose**

(Level AA)

Tujuan untuk setiap bidang masukan yang digunakan untuk mendapatkan informasi pengguna dapat ditentukan secara pemrograman ketika:

- Bidang masukan menyajikan tujuan yang diidentifikasi di bagian tujuan masukan untuk komponen antarmuka pengguna.
- Konten diimplementasikan menggunakan teknologi dengan dukungan untuk mengidentifikasi makna yang diharapkan masukan data formulir.

### **2.1.16 Kriteria Sukses 1.3.6 Identify Purpose**

(Level AAA)

Dalam konten yang diimplementasi dengan bahasa *markup*, tujuan dari komponen antarmuka, ikon, dan bidang dapat ditentukan secara pemrograman.

### 2.1.17 Kriteria Sukses 1.4.1 Use of Color

(Level A)

Warna tidak hanya digunakan sebagai satu-satunya cara visual untuk menyampaikan informasi, menunjukkan aksi, menampilkan respon, atau membedakan elemen visual.

### 2.1.18 Kriteria Sukses 1.4.2 Audio Control

(Level A)

Jika ada audio yang diputar secara otomatis di halaman web yang berdurasi lebih dari 3 detik, maka setidaknya ada mekanisme untuk menjeda atau menghentikan audio, atau ada mekanisme untuk mengontrol volume audio secara independen dari tingkat volume sistem secara keseluruhan.

### 2.1.19 Kriteria Sukses 1.4.3 Contrast (Minimum)

(Level AA)

Presentasi visual dari teks, gambar teks, memiliki rasio kontras setidaknya 4.5:1, kecuali jika:

- Teks besar: Teks berukuran besar dan gambar teks berukuran besar memiliki rasio kontras setidaknya 3:1.
- Insidental: Teks atau gambar teks yang merupakan bagian dari komponen antarmuka yang tidak aktif, atau hanya dekorasi saja, atau yang tidak tampak, atau bagian dari gambar yang memiliki konten visual yang signifikan, tidak memiliki syarat kontras.
- Logo: Teks yang merupakan bagian dari logo atau nama merek tidak memiliki syarat kontras.

### 2.1.20 Kriteria Sukses 1.4.4 Resize text

(Level AA)

Teks dapat diubah ukurannya tanpa teknologi bantuan hingga 200 persen tanpa kehilangan konten atau fungsionalitasnya kecuali untuk keterangan dan gambar teks.

### 2.1.21 Kriteria Sukses 1.4.5 Images of Text

(Level AA)

Jika suatu teknologi yang digunakan dapat mencapai presentasi visual, maka teks digunakan untuk menyampaikan informasi daripada gambar teks kecuali dalam kondisi berikut:

1. Customizable: Gambar teks dapat di sesuaikan dengan kebutuhan pengguna.
2. Esensial: Tampilan teks esensial untuk menyampaikan informasi.

### 2.1.22 Kriteria Sukses 1.4.6 Contrast (Enhanced)

(Level AAA)

Presentasi visual dari teks, gambar teks, memiliki rasio kontras setidaknya 7:1, kecuali jika:

1. Teks besar: Teks berukuran besar dan gambar teks berukuran besar memiliki rasio kontras setidaknya 4.5:1.

2. Insidental: Teks atau gambar teks yang merupakan bagian dari komponen antarmuka yang tidak aktif, atau hanya dekorasi saja, atau yang tidak tampak, atau bagian dari gambar yang memiliki konten visual yang signifikan, tidak memiliki syarat kontras.
3. Logo: Teks yang merupakan bagian dari logo atau nama merek tidak memiliki syarat kontras.

### 2.1.23 Kriteria Sukses 1.4.7 Low atau No Background Audio

(Level AAA)

Untuk konten rekaman audio saja yang berisi pidato di latar depan, bukan audio *CAPTCHA* atau audio logo, dan bukan suara musik atau rap, setidaknya salah satu kondisi berikut ini benar:

- Tidak ada latar belakang: Audio tidak berisi suara di latar belakangnya.
- Mematikan suara: Suara pada latar belakang dapat dimatikan
- 20 Desibel: Suara pada latar belakang setidaknya 20 desibel lebih kecil daripada pidato pada latar depan, dengan pengecualian untuk suara yang berdurasi satu atau dua detik.

### 2.1.24 Kriteria Sukses 1.4.8 Visual Presentation

(Level AAA)

Untuk presentasi visual dari blok teks, sebuah mekanisme harus ada untuk mencapai kondisi berikut:

- Warna pada latar belakang dan latar depan dapat dipilih oleh pengguna.
- Lebar tidak lebih dari 80 karakter.
- Teks tidak diratakan (sejajar dengan margin kiri dan kanan)
- Jeda baris dalam paragraf setidaknya satu setengah spasi, dan jeda paragraf setidaknya 1.5 lebih besar dari jeda baris.
- Teks dapat diubah ukurannya tanpa teknologi bantuan hingga 200 persen dengan cara pengguna tidak perlu gulir secara horizontal untuk membaca teks dalam mode *full-screen*.

### 2.1.25 Kriteria Sukses 1.4.9 Images of Text (No Exception)

(Level AAA)

Gambar teks hanya digunakan untuk dekorasi murni atau dimana presentasi teks tertentu esensial untuk informasi yang disampaikan.

### 2.1.26 Kriteria Sukses 1.4.10 Reflow

(Level AA)

Konten dapat ditampilkan tanpa kehilangan informasi atau fungsionalitasnya, dan tanpa memerlukan gulir di 2 dimensi untuk:

- Gulir vertikal untuk konten yang lebarnya setara dengan 320 piksel *css*.
- Gulir horizontal untuk konten yang tingginya setara dengan 256 piksel *css*.

Kecuali bagian konten yang memerlukan tata letak dua dimensi untuk keperluan dan tujuannya.



### 2.1.27 Kriteria Sukses 1.4.11 Non-text Contrast

(Level AA)

Presentasi visual berikut harus memiliki rasio kontras setidaknya 3:1 terhadap warna yang bedekatan:

- Komponen antarmuka: Informasi visual diperlukan untuk mengidentifikasi komponen antarmuka dan statusnya, kecuali antarmuka yang tidak aktif atau tampilan komponen ditentukan oleh agen pengguna dan tidak diubah oleh pembuat web.
- Objek grafik: Bagian dari grafik diperlukan untuk menjelaskan kontennya, kecuali tampilan grafik esensial untuk informasi yang disampaikan.

### 2.1.28 Kriteria Sukses 1.4.12 Text Spacing

(Level AA)

Dalam konten yang diimplementasikan menggunakan bahasa *markup* yang mendukung properti gaya teks berikut, tidak ada kehilangan konten atau fungsionalitas ketika mengatur setelan semua hal berikut dan tidak mengganti properti tampilannya:

- Tinggi baris setidaknya 1.5 kali ukuran teks.
- Spasi antar paragraf setidaknya 2 kali ukuran teks.
- Spasi antar huruf setidaknya 0.12 kali ukuran teks
- Spasi antar kata setidaknya 0.16 kali ukuran teks.

### 2.1.29 Kriteria Sukses 1.4.13 Content on Hover or Focus

(Level AA)

Ketika mendapatkan dan kemudian menghapus *pointer hover* atau fokus keyboard memicu konten tambahan untuk menjadi terlihat dan kemudian disembunyikan, berikut ini benar:

- Dapat disingkirkan: Tersedianya mekanisme untuk menyingkirkan konten tambahan tanpa menggerakkan *pointer hover* atau fokus keyboard, kecuali konten tambahan menunjukkan kesalahan input, atau tidak mengganggu atau menggantikan konten lain.
- *Hoverable*: Jika *pointer hover* dapat memicu konten tambahan, maka penunjuk dapat digerakkan di atas konten tambahan tanpa konten tambahan tersebut hilang.
- Persisten: Konten tambahan dapat terlihat sampai penunjuk atau fokus dihilangkan, pengguna menyingkirkannya, atau informasinya sudah tidak valid.

Kecuali: Presentasi visual dari konten tambahan dikontrol oleh agen pengguna dan tidak diubah oleh pembuat web.

### 2.1.30 Kriteria Sukses 2.1.1 Keyboard

(Level A)

Semua fungsionalitas konten dapat dioperasikan melalui antarmuka keyboard tanpa memerlukan waktu spesifik untuk penekanan tombolnya, kecuali jika fungsi yang mendasarinya membutuhkan input yang bergantung pada pergerakan pengguna dan bukan hanya pada titik akhir.

### 2.1.31 Kriteria Sukses 2.1.2 No Keyboard Trap

(Level A)

Jika fokus keyboard dapat dipindahkan ke komponen tertentu pada halaman menggunakan antarmuka keyboard, maka fokus dapat dipindahkan dari komponen itu hanya menggunakan antarmuka keyboard, dan jika memerlukan lebih dari sekadar penekanan tombol panah atau tombol *tab* atau metode keluar standar lainnya, pengguna diberikan informasi tentang metode tersebut untuk memindahkan fokus.

### 2.1.32 Kriteria Sukses 2.1.3 Keyboard (No Exception)

(Level AAA)

Semua fungsionalitas konten dapat dioperasikan melalui antarmuka keyboard tanpa memerlukan waktu spesifik untuk penekanan tombolnya.

### 2.1.33 Kriteria Sukses 2.1.4 Character Key Shortcuts

(Level A)

Jika pintasan keyboard diimplementasi dengan menggunakan huruf (termasuk huruf besar dan kecil), tanda baca, angka, atau karakter simbol, maka setidaknya salah satu dari yang berikut ini benar:

- Matikan: Tersedia mekanisme untuk mematikan pintasan.
- Dipetakan kembali: Tersedia mekanisme untuk memetakan kembali pintasan untuk menggunakan satu atau lebih karakter keyboard yang tidak dapat dicetak.
- Hanya aktif saat fokus: Pintasan keyboard untuk komponen antarmuka pengguna hanya aktif saat komponen tersebut memiliki fokus.

### 2.1.34 Kriteria Sukses 2.2.1 Timing Adjustable

(Level A)

Untuk setiap batasan waktu yang ditentukan oleh konten, setidaknya salah satu dari yang berikut ini benar:

- Matikan: Pengguna dapat mematikan batas waktu sebelum mencapai batas tersebut; atau
- Sesuaikan: Pengguna dapat menyesuaikan batas waktu sebelum mencapai batas tersebut, dengan waktu tambahan yang setidaknya sepuluh kali lebih panjang dari setelan standar; atau
- Perpanjangan: Pengguna diperingati ketika batas waktu habis dan diberikan waktu setidaknya 20 detik untuk menambah batas waktu dengan perintah sederhana (misalnya, tekan tombol spasi), dan pengguna dapat menambah batas waktu setidaknya sepuluh kali lipat; atau
- Perkecualian waktu riil: Batas waktu diperlukan sebagai bagian dari kejadian waktu riil (misalnya, lelang), dan mustahil untuk menyediakan alternatif untuk batas waktu; atau
- Perkecualian esensial: Batas waktu esensial dan perpanjangan batas ini menyalahi inti dari kegiatan tersebut; atau
- Perkecualian 20 jam: Batas waktu yang diberikan lebih dari 20 jam.

### **2.1.35 Kriteria Sukses 2.2.2 Pause, Stop, Hide**

(Level A)

Untuk informasi yang bergerak, berkedip, bergulir, atau diperbarui secara otomatis, semua hal berikut ini benar:

- Bergerak, berkedip, bergulir: Untuk informasi yang bergerak, berkedip, bergulir yang (1) mulainya otomatis, (2) berlangsung lebih dari lima detik, dan (3) ditampilkan paralel dengan konten lain, ada mekanisme bagi pengguna untuk memberi jeda, memberhentikan, atau menyembunyikan informasi tersebut; kecuali jika aktivitas bergerak, berkedip, atau bergulir tersebut merupakan bagian dari aktivitas yang esensial; dan
- Diperbarui otomatis: Untuk informasi mana pun yang diperbarui secara otomatis, yaitu yang (1) mulainya otomatis dan (2) ditampilkan paralel dengan konten lain, ada mekanisme bagi pengguna untuk memberi jeda, memberhentikan, atau menyembunyikan informasi tersebut; kecuali jika pembaruan otomatis tersebut merupakan bagian dari aktivitas yang esensial; dan

### **2.1.36 Kriteria Sukses 2.2.3 No Timing**

(Level AAA)

Waktu bukanlah bagian esensial dari kejadian atau aktivitas yang disajikan oleh konten, kecuali untuk media yang disinkronisasi non-interaktif dan kejadian waktu riil.

### **2.1.37 Kriteria Sukses 2.2.4 Interruptions**

(Level AAA)

Interupsi dapat ditunda oleh pengguna, kecuali interupsi yang melibatkan keadaan darurat.

### **2.1.38 Kriteria Sukses 2.2.5 Re-authenticating**

(Level AAA)

Ketika sesi autentikasi berakhir, pengguna dapat melanjutkan aktivitas tanpa kehilangan data setelah autentikasi ulang.

### **2.1.39 Kriteria Sukses 2.2.6 Timeouts**

(Level AAA)

Pengguna diperingatkan tentang waktu ketidakaktifan yang dapat menyebabkan kehilangan data, kecuali jika data tersebut disimpan lebih dari 20 jam ketika pengguna tidak melakukan tindakan apapun.

### **2.1.40 Kriteria Sukses 2.3.1 Three Flashes or Below Threshold**

(Level A)

Halaman web tidak mengandung apapun yang berkedip lebih dari tiga kali dalam periode satu detik, atau kelipan berada dibawah batas umum kelipan dan kelipan merah

**2.1.41 Kriteria Sukses 2.3.2 Three Flashes**

(Level AAA)

Halaman web tidak mengandung apapun yang berkelip lebih dari tiga kali dalam periode satu detik

**2.1.42 Kriteria Sukses 2.3.3 Animation from Interactions**

(Level AAA)

Animasi gerak yang dipicu oleh interaksi dapat dinonaktifkan, kecuali jika animasi itu penting untuk fungsionalitas atau informasi yang sedang disampaikan.

**2.1.43 Kriteria Sukses 2.4.1 Bypass Blocks**

(Level A)

Tersedianya sebuah mekanisme untuk meloncati area konten yang diulang-ulang pada berbagai halaman web.

**2.1.44 Kriteria Sukses 2.4.2 Page Titled**

(Level A)

Halaman web memiliki judul yang menggambarkan topik atau tujuan.

**2.1.45 Kriteria Sukses 2.4.3 Focus Order**

(Level A)

Jika halaman web dapat dinavigasi secara berurutan dan urutan navigasi memengaruhi makna atau operasi, maka komponen yang dapat menerima fokus akan menerima fokus dalam urutan yang menjaga makna dan pengoperasian.

**2.1.46 Kriteria Sukses 2.4.4 Link Purpose (In Context)**

(Level A)

Tujuan setiap tautan dapat ditentukan dari teks tautan saja atau dari kombinasi teks tautan beserta konteks tautan yang ditentukan secara pemrograman, kecuali jika tautan tersebut bersifat ambigu bagi pengguna secara umum.

**2.1.47 Kriteria Sukses 2.4.5 Multiple Ways**

(Level AA)

Ada berbagai cara untuk menemukan halaman web dalam satu set halaman web kecuali halaman web adalah hasil dari, atau langkah dalam suatu proses.

**2.1.48 Kriteria Sukses 2.4.6 Headings and Labels**

(Level AA)

Judul dan label menjelaskan topik atau tujuan.

### 2.1.49 Kriteria Sukses 2.4.7 Focus Visible

(Level AA)

Setiap antarmuka pengguna yang dapat dioperasikan dengan keyboard memiliki mode pengoperasian di mana indikator fokus keyboard terlihat jelas.

### 2.1.50 Kriteria Sukses 2.4.8 Location

(Level AAA)

Informasi mengenai lokasi pengguna dalam satu set halaman web tersedia.

### 2.1.51 Kriteria Sukses 2.4.9 Link Purpose (Link Only)

(Level AAA)

Tersedianya sebuah mekanisme untuk memungkinkan tujuan setiap tautan diidentifikasi dari teks tautan saja, kecuali jika tujuan tersebut ambigu bagi pengguna secara umum.

### 2.1.52 Kriteria Sukses 2.4.10 Section Headings

(Level AAA)

Judul bagian digunakan untuk mengatur konten.

### 2.1.53 Kriteria Sukses 2.5.1 Pointer Gestures

(Level A)

Semua fungsionalitas yang menggunakan gerakan *multipoint* atau berbasis jalur untuk operasi dapat dioperasikan dengan *pointer* tunggal tanpa gestur berbasis jalur, kecuali jika *multipoint* atau gestur berbasis jalur sangat penting.

### 2.1.54 Kriteria Sukses 2.5.2 Pointer Cancellation

(Level A)

Untuk fungsionalitas yang dapat dioperasikan menggunakan pointer tunggal, setidaknya salah satu dari yang berikut ini benar:

- Tidak Ada *Down-Event*: *Down-Event* dari *pointer* tidak dipakai untuk eksekusi bagian dari fungsi;
- Gagal atau Batalkan: Keberhasilan fungsi pada *up-event*, dan tersedia mekanisme untuk mengagalkan fungsi sebelum berhasil atau membatalkan fungsi setelah berhasil;
- *Up Reversal*: *Up-event* akan membalikkan setiap hasil dari *down-event* sebelumnya;
- Esensial: Keberhasilan fungsi saat *down-event* esensial.

### 2.1.55 Kriteria Sukses 2.5.3 Label in Name

(Level A)

Untuk komponen antarmuka pengguna dengan label yang menyertakan teks atau gambar teks, nama tersebut berisi teks yang ditampilkan secara visual.

### 2.1.56 Kriteria Sukses 2.5.4 Motion Actuation

(Level A)

Fungsi yang dapat dioperasikan oleh gerakan perangkat atau gerakan pengguna juga dapat dioperasikan oleh komponen antarmuka pengguna dan merespons gerakan dapat dinonaktifkan untuk mencegah pergerakan tidak disengaja, kecuali ketika:

- Antarmuka yang Didukung: Gerakan digunakan untuk mengoperasikan fungsi melalui antarmuka yang mendukung aksesibilitas;
- Esensial: Gerakan sangat esensial untuk fungsi dan jika menonaktifkannya maka membuat aktivitas tersebut tidak valid.

### 2.1.57 Kriteria Sukses 2.5.5 Target Size

(Level AAA)

Ukuran target untuk *pointer* masukan setidaknya 44 *css* piksel kecuali jika:

- Setara: Target tersedia melalui tautan atau kontrol yang setara pada halaman yang sama berukuran setidaknya 44 *css* piksel;
- *Inline*: Target terdapat pada kalimat atau blok teks;
- Kontrol Agen Pengguna: Ukuran dari target ditentukan oleh agen pengguna dan tidak diubah oleh pembuat web;
- Esensial: Presentasi visual dari target sangat penting untuk informasi yang disampaikan.

### 2.1.58 Kriteria Sukses 2.5.6 Concurrent Input Mechanisms

(Level AAA)

Konten web tidak membatasi penggunaan modalitas masukan yang tersedia pada platform kecuali jika pembatasan itu penting, diperlukan untuk memastikan keamanan konten, atau diperlukan untuk menghormati pengaturan pengguna.

### 2.1.59 Kriteria Sukses 3.1.1 Language of Page

(Level A)

Standar bahasa manusia dari setiap halaman web dapat ditentukan secara pemrograman.

**2.1.60 Kriteria Sukses 3.1.2 Language of Parts**

(Level AA)

Bahasa manusia dari setiap bait atau frasa dalam konten dapat ditentukan secara pemrograman kecuali untuk nama diri, istilah teknis, kata-kata dari bahasa yang tidak ditentukan, dan kata-kata atau frasa yang telah menjadi bagian dari bahasa sehari-hari dari teks yang ada di sekelilingnya.

**2.1.61 Kriteria Sukses 3.1.3 Unusual Words**

(Level AAA)

Tersedianya mekanisme untuk mengidentifikasi definisi kata atau frasa tertentu yang digunakan dengan cara yang tidak biasa atau terbatas, termasuk idiom dan jargon.

**2.1.62 Kriteria Sukses 3.1.4 Abbreviations**

(Level AAA)

Tersedianya mekanisme untuk mengidentifikasi kepanjangan dari singkatan.

**2.1.63 Kriteria Sukses 3.1.5 Reading Level**

(Level AAA)

Ketika teks membutuhkan kemampuan membaca lebih maju daripada tingkat pendidikan menengah bawah setelah penghapusan nama diri dan jabatan, konten tambahan atau versi yang tidak memerlukan kemampuan membaca lebih maju daripada tingkat pendidikan menengah bawah harus tersedia.

**2.1.64 Kriteria Sukses 3.1.6 Pronunciation**

(Level AAA)

Tersedianya mekanisme untuk mengidentifikasi pengucapan kata tertentu ketika makna kata tersebut, dalam konteksnya, bersifat ambigu tanpa mengetahui pengucapannya.

**2.1.65 Kriteria Sukses 3.2.1 On Focus**

(Level A)

Tidak ada perubahan konteks jika komponen antarmuka pengguna menerima fokus.

**2.1.66 Kriteria Sukses 3.2.2 On Input**

(Level A)

Mengubah pengaturan komponen antarmuka pengguna mana pun tidak otomatis menyebabkan perubahan konteks kecuali jika pengguna telah diperingati tentang hal ini sebelum menggunakan komponen.

**2.1.67 Kriteria Sukses 3.2.3 Consistent Navigation**

(Level AA)

Mekanisme navigasi yang diulang pada beberapa halaman web dalam satu set halaman web muncul dalam urutan relatif yang sama setiap kali diulang, kecuali jika perubahan dilakukan oleh pengguna.

**2.1.68 Kriteria Sukses 3.2.4 Consistent Identification**

(Level AA)

Komponen yang memiliki fungsi yang sama dalam satu set halaman web diidentifikasi secara konsisten.

**2.1.69 Kriteria Sukses 3.2.5 Change on Request**

(Level AAA)

Perubahan konteks hanya terjadi oleh permintaan pengguna atau ada mekanisme tersedia untuk menonaktifkan perubahan tersebut.

**2.1.70 Kriteria Sukses 3.3.1 Error Identification**

(Level A)

Jika kesalahan masukan terdeteksi secara otomatis, *item* yang salah diidentifikasi dan kesalahan tersebut dijelaskan kepada pengguna dalam teks.

**2.1.71 Kriteria Sukses 3.3.2 Labels or Instructions**

(Level A)

Label atau instruksi diberikan saat konten membutuhkan masukan pengguna.

**2.1.72 Kriteria Sukses 3.3.3 Error Suggestion**

(Level AA)

Jika kesalahan masukan terdeteksi secara otomatis dan saran untuk mengoreksi diketahui, maka saran tersebut diberikan kepada pengguna, kecuali jika itu akan membahayakan keamanan atau tujuan konten.

**2.1.73 Kriteria Sukses 3.3.4 Error Prevention (Legal, Financial, Data)**

(Level AA)

Untuk halaman web yang menyebabkan komitmen legal atau transaksi finansial bagi pengguna, dan memodifikasi atau menghapus data yang dapat dikontrol pengguna dalam sistem penyimpanan data, atau yang mengirimkan tanggapan tes pengguna, setidaknya salah satu dari yang berikut ini benar:

- Bisa dibatalkan: Data yang dikirim bisa dibatalkan;



- Diperiksa: Data yang dimasukkan oleh pengguna dapat diperiksa untuk eror masukan dan pengguna dipersilakan untuk mengoreksinya;
- Dikonfirmasi: Tersedianya mekanisme untuk meninjau, mengonfirmasi, dan mengoreksi informasi sebelum informasi tersebut dikirim.

#### 2.1.74 Kriteria Sukses 3.3.5 Help

(Level AAA)

Tersedianya bantuan pada konteks yang sedang berjalan.

#### 2.1.75 Kriteria Sukses 3.3.6 Error Prevention (All)

(Level AAA)

Untuk halaman web yang mengharuskan pengguna untuk mengirimkan informasi, setidaknya salah satu dari yang berikut ini benar:

- Bisa dibatalkan: Data yang dikirim bisa dibatalkan;
- Diperiksa: Data yang dimasukkan oleh pengguna dapat diperiksa untuk eror masukan dan pengguna dipersilakan untuk mengoreksinya;
- Dikonfirmasi: Tersedianya mekanisme untuk meninjau, mengonfirmasi, dan mengoreksi informasi sebelum informasi tersebut dikirim.

#### 2.1.76 Kriteria Sukses 4.1.1 Parsing

(Level A)

Dalam konten yang diimplementasikan menggunakan bahasa *markup*, elemen memiliki tag awal dan akhir yang lengkap, elemen disusun berlapis sesuai dengan spesifikasinya, elemen tidak mengandung atribut duplikat, dan setiap ID unik, kecuali jika spesifikasi mengizinkan fitur ini.

#### 2.1.77 Kriteria Sukses 4.1.2 Name, Role, Value

(Level A)

Untuk semua komponen antarmuka pengguna (termasuk tetapi tidak terbatas pada: elemen formulir, tautan, dan komponen yang dihasilkan oleh skrip), nama dan peran dapat ditentukan secara pemrograman; *states*, properti, dan nilai-nilai yang dapat diatur oleh pengguna dapat diatur secara terprogram; dan pemberitahuan perubahan pada *item-item* ini tersedia untuk agen pengguna, termasuk teknologi bantu.

#### 2.1.78 Kriteria Sukses 4.1.3 Status Messages

(Level AA)

Dalam konten yang diimplementasikan menggunakan bahasa *markup*, pesan status dapat ditentukan secara terprogram melalui peran atau sifat sedemikian rupa sehingga dapat disajikan kepada pengguna dengan teknologi bantuan tanpa menerima fokus.

## 2.2 SharIF Judge

*SharIF Judge* adalah sebuah aplikasi gratis dan *open source* untuk menilai code berbahasa *C*, *C++*, *Java* dan *Python*. *SharIF Judge* adalah pencabangan dari *Sharif Judge* yang telah dibuat oleh Mohammed Javad Naderi. Versi dari pencabangan ini memuat fitur baru yang diperlukan oleh jurusan teknik informatika UNPAR. Aplikasi ini dibuat menggunakan PHP (*CodeIgnitor framework*) dan bagian backendnya dibuat dengan *BASH*.

### 2.2.1 Instalasi

Berikut adalah persyaratan dan langkah-langkah instalasi yang perlu dilakukan untuk menjalankan *SharIF Judge*:

#### Persyaratan

Untuk menjalankan *SharIF Judge*, diperlukan sebuah *Linux server* dengan persyaratan berikut:

- *Webserver* menjalankan PHP versi 5.3 atau lebih baru
- Pengguna dapat menjalankan PHP melalui *command line*. Pada *Ubuntu*, pengguna perlu menginstal paket *php5-cli*
- *MySql database* (dengan ekstensi untuk PHP) atau *PostgreSql database*
- PHP harus diberikan akses untuk menjalankan perintah menggunakan fungsi *shell\_exec*. Contoh seperti perintah berikut:

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

- Perkakas yang digunakan untuk melakukan proses kompilasi dan menjalankan kode yang dikumpulkan (*gcc, g++, javac, java, python2, python3 commands*)
- *Perl* lebih baik diinstall untuk alasan ketepatan waktu, batas memori dan memaksimalkan batas ukuran pada *output* kode yang dikirim.

#### Instalasi

- Mengunduh versi terakhir dari *SharIF Judge* dan *unpack* hasil unduhan di direktori *public html*
- Memindahkan folder *system* dan *application* ke luar *public directory*, dan masukkan *path* lengkap di *file index.php*

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

- Membuat sebuah database *Mysql* atau *PostgreSql* untuk *Sharif Judge*. Jangan menginstall paket koneksi database untuk *C/C++*, *Java*, atau *Python*
- Mengatur 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
/*****/

```

- Mengatur *RADIUS* server dan *mail server* pada file *application/config/secrets.example.php* dan simpan dengan nama *secrets.php*
- Membuat direktori *application/cache/Twig* agar dapat ditulis oleh PHP
- Membuka halaman utama *SharIF Judge* pada *web browser* dan ikuti proses instalasi berikutnya
- *Log in* menggunakan akun *admin*
- Memindahkan folder *tester* dan *assigments* ke luar public directory lalu simpan path lengkap di halaman *Settings*. Dua folder tersebut harus dapat ditulis oleh PHP. File-file yang diunggah akan disimpan di folder *assigments* sehingga tidak dapat diakses publik.

### 2.2.2 Clean URLs

Secara standar, *index.php* merupakan bagian dari seluruh *URLs* yang ada pada *SharIF Judge*. Berikut contoh *URLs SharIF Judge*:

- <http://example.mjnaderi.ir/index.php/dashboard>
- <http://example.mjnaderi.ir/index.php/users/add>

Pengguna dapat menghilangkan *index.php* di atas dan memiliki *URLs* yang baik jika sistem pengguna mendukung *URL rewriting*. *URL rewriting* merupakan proses mengubah parameter yang terdapat pada *URL*. Berikut adalah contoh *URL* yang telah melewati proses *URL rewriting*:

- <http://example.mjnaderi.ir/dashboard>
- <http://example.mjnaderi.ir/users/add>

#### Cara memakai Clean URLs

Pengguna dapat melakukan langkah berikut untuk menggunakan *clean URL*:

- Mengubah nama file *.htaccess2* menjadi *.htaccess* yang ada di direktori utama *SharIF Judge*. *.htaccess* merupakan sebuah file konfigurasi yang digunakan pada *web server*
- Mengubah *\$config['index\_page'] = 'index.php'*; menjadi *\$config['index\_page'] = ''*; pada file *application/config/config.php*

### 2.2.3 Users

Pada *SharIF Judge*, *users* dibagi menjadi 4 *role*. Keempat *role* tersebut adalah *Admins*, *Head Instructor*, *Instructor*, dan *Students*. Tabel 2.1 menunjukkan *level* yang ada untuk setiap *role*.

Tabel 2.1: *User Roles Table*

<i>User Role</i>	<i>User Level</i>
<i>Admin</i>	3
<i>Head Instructor</i>	2
<i>Instructor</i>	1
<i>Student</i>	0

Setiap *users* dapat melakukan aksi yang berbeda-beda. Aksi yang dapat dilakukan para *users* akan disesuaikan pada tingkat masing-masing. Tabel 2.2 menunjukkan aksi yang dapat dilakukan pada setiap *role*.

Tabel 2.2: *Permission Table*

Aksi	<i>Admin</i>	<i>Head Instructor</i>	<i>Instructor</i>	<i>Student</i>
Mengubah <i>Settings</i>	✓	×	×	×
Menambah/Menghapus <i>users</i>	✓	×	×	×
Mengubah Peran <i>users</i>	✓	×	×	×
Menambah/Menghapus/Mengubah <i>Assignment</i>	✓	✓	×	×
Mengunduh <i>Test</i>	✓	✓	×	×
Menambah/Menghapus/Mengubah Notifikasi	✓	✓	×	×
<i>Rejudge</i>	✓	✓	×	×
Melihat/ <i>Pause</i> /Melanjutkan/ <i>Submission Queue</i>	✓	✓	×	×
Mendeteksi Kode yang Mirip	✓	✓	×	×
Melihat Semua Kode	✓	✓	✓	×
Mengunduh Kode Final	✓	✓	✓	×
Memilih <i>Assignment</i>	✓	✓	✓	✓
<i>Submit</i>	✓	✓	✓	✓

Pengguna dapat menambahkan *users* dengan masuk ke bagian *Add Users* di halaman *Users*. Pengguna harus mengisi semua informasi yang ada pada *text area*. Baris yang diawali dengan # adalah komentar. Setiap baris lainnya mewakili pengguna dengan sintaksis berikut:

```

USERNAME,EMAIL,DISPLAY-NAME,PASSWORD,ROLE

* Usernames may contain lowercase letters or numbers and must be between 3 and 20
  characters in length.
* Passwords must be between 6 and 30 characters in length.
* You can use RANDOM[n] for password to generate random n-digit password.
* ROLE must be one of these: 'admin', 'head_instructor', 'instructor', 'student'

```

Berikut adalah contohnya:

```

# This is a comment!
# This is another comment!
instructor instructor@sharifjudge.ir 123456 head_instructor
instructor2 instructor2@sharifjudge.ir random[7] instructor
student1 st1@sharifjudge.ir random[6] student
student2 st2@sharifjudge.ir random[6] student
student3 st3@sharifjudge.ir random[6] student

```

```
student4 st4@sharifjudge.ir random[6] student
student5 st5@sharifjudge.ir random[6] student
student6 st6@sharifjudge.ir random[6] student
student7 st7@sharifjudge.ir random[6] student
```

### 2.2.4 Menambah Assignment

Pengguna dapat menambahkan *assignment* dengan cara masuk ke bagian *Add* di halaman *assignments*. Gambar 2.1 menunjukkan contoh tampilan halaman untuk menambah *assignment*.

Gambar 2.1: Tampilan Halaman *Assignments*

Berikut ini adalah pengaturan yang terdapat pada halaman *Add Assignments*:

- *Assignment Name*  
Nama *assignment* yang akan ditampilkan dalam daftar *assignment*.
- *Start Time*  
User tidak dapat mengumpulkan assignment sebelum "*Start Time*". Format yang digunakan untuk pengaturan start time adalah *MM/DD/YYYY HH:MM:SS*. Contohnya: 08/31/2013 12:00:00
- *Finish Time, Extra Time*  
Peserta tidak dapat mengumpulkan *assignment* setelah *Finish Time* + *Extra Time*. *Assignment* yang telat akan dikalikan dengan koefisien tertentu. Pengguna harus menulis *script* PHP untuk menghitung koefisien pada bidang "*Coefficient Rule*". Format yang digunakan untuk pengaturan *finish time* adalah *MM/DD/YYYY HH:MM:SS*. Contoh: 08/31/2013 23:59:59. "*Extra Time*" akan terhitung dalam satuan menit. Pengguna juga dapat menggunakan operator aritmatika seperti \*, -, +, /. Contohnya 120 (2 jam) atau 48\*60 (2 hari).
- *Participants*  
Pengguna dapat menggunakan kata kunci *ALL* pada kolom *Participants* untuk mengizinkan

seluruh peserta agar dapat mengumpulkan *assignment*. Untuk membatasi peserta tertentu, pengguna dapat memasukkan *username* peserta pada kolom *Participants*. Setiap *username* dapat dipisahkan menggunakan tanda koma. Contoh: *admin, instructor1, instructor2, student1, student2, student3, student4*.

- *Test*  
Pengguna dapat mengunggah *test cases* dalam bentuk *zip file* dengan struktur yang sudah ditentukan.
- *Open*  
Pengguna dapat membuka atau menutup *assignment* untuk *user students* menggunakan pilihan ini. Jika pengguna menutup *assignment*, *non-student users* masih dapat mengumpulkan *assignment*.
- *Scoreboard*  
Pengguna dapat mengaktifkan atau menonaktifkan papan nilai dengan menggunakan pilihan ini.
- *Java Exceptions*  
Pengguna dapat mengaktifkan dan menonaktifkan *java exception* yang ditunjukan kepada *students*. Perubahan pada pilihan ini tidak memengaruhi kode yang sebelumnya sudah dinilai. Nama *exception* akan muncul jika pada *file* `pathtester/java_exceptions_list` berisikan nama *exception* tersebut. Berikut adalah hasil *exception* yang ditunjukan jika pengguna mengaktifkan pengaturan *Java Exceptions*:

```
Test 1
ACCEPT
Test 2
Runtime Error (java.lang.ArrayIndexOutOfBoundsException)
Test 3
Runtime Error (java.lang.ArrayIndexOutOfBoundsException)
Test 4
ACCEPT
Test 5
ACCEPT
Test 6
ACCEPT
Test 7
ACCEPT
Test 8
Runtime Error (java.lang.ArrayIndexOutOfBoundsException)
Test 9
Runtime Error (java.lang.StackOverflowError)
Test 10
Runtime Error (java.lang.ArrayIndexOutOfBoundsException)
```

- *Archieved Assignment*  
Jika fitur ini diaktifkan maka *assignment* akan dibuat dengan *Finish Time* 2038-01-18 00:00:00 (UTC+ 7) dalam kata lain pengguna memiliki waktu yang tidak terbatas untuk mengumpulkan *assignment* tersebut.

- *Coefficient Rule*

Pengguna dapat menulis *script* PHP pada bagian ini. Pengguna harus memasukkan koefisien (dari 100) pada variabel *\$coefficient*. Pengguna dapat menggunakan variabel *\$extra\_time* dan *\$delay*. *\$extra\_time* merupakan total waktu ekstra yang diberikan kepada *users* dalam satuan detik dan *\$delay* merupakan jumlah detik berlalu dari waktu selesai (dapat negatif). *Script* PHP pada bagian ini tidak boleh mengandung *tags* `<?php` , `<?` , `?>`. Berikut contoh *\$extra\_time* 172800 (2 hari):

```
if ($delay<=0)
// no delay
$coefficient = 100;

elseif ($delay<=3600)
// delay less than 1 hour
$coefficient = ceil(100-((30*$delay)/3600));

elseif ($delay<=86400)
// delay more than 1 hour and less than 1 day
$coefficient = 70;

elseif (($delay-86400)<=3600)
// delay less than 1 hour in second day
$coefficient = ceil(70-((20*($delay-86400))/3600));

elseif (($delay-86400)<=86400)
// delay more than 1 hour in second day
$coefficient = 50;

elseif ($delay > $extra_time)
// too late
$coefficient = 0;
```

- *Time Limit*

Pengguna dapat mengatur batas waktu untuk menjalankan kode dalam satuan milisekon. Program yang ditulis menggunakan *Python* dan *Java* biasanya lebih lambat dari *C/C++*. Oleh karena itu *Python* dan *Java* membutuhkan waktu yang lebih.

- *Memory Limit*

Pengguna dapat mengatur batas memori dalam satuan *kilobytes*, namun penggunaan *Memory Limit* tidak terlalu akurat.

- *Allowed Languages*

Pengguna dapat mengatur bahasa untuk setiap *problem* (dipisahkan menggunakan koma). Bahasa yang tersedia seperti *C*, *C++*, *Java*, *Python 2*, *Python 3*, *zip*, *PDF*. Pengguna dapat menggunakan *zip* atau *PDF* jika mengaktifkan pilihan *Upload Only*. Contoh: *C*, *C++* , *zip* atau *Python 2*, *Python 3* atau *Java* , *C*.

- *Diff Command*

*Command* ini digunakan untuk membandingkan keluaran dengan keluaran yang benar. Secara standar *SharIF Judge* menggunakan *diff*, namun pengguna dapat mengubah *command* pada bagian ini. Bidang ini tidak boleh mengandung spasi.

- *Diff Arguments*

Pengguna dapat mengatur argumen dari *Diff Command* disini. Pengguna dapat melihat *man diff* untuk melihat daftar lengkap *diff* argumen. *SharIF Judge* menambahkan dua pilihan baru yaitu *ignore* dan *identical*. *Ignore* akan menghiraukan semua baris baru dan spasi. *Identical* tidak akan menghiraukan apapun namun keluaran dari file yang dikumpulkan harus identik dengan keluaran *test case* agar dapat diterima.

- *Upload Only*

Jika pengguna mengatur *problem* sebagai *Upload-Only*, maka *SharIF Judge* tidak akan menilai *assignment* pada *problem* tersebut. Pengguna dapat menggunakan *zip* dan *PDF* pada *allowed languages* jika mengaktifkan pilihan ini.

### 2.2.5 Sample Assignment

Berikut adalah contoh *assignment* untuk testing pada *SharIF Judge*. Untuk menambah *assignment* pengguna dapat menekan *Add* pada halaman *Assignments*.

#### 1. Problem 1 (Penjumlahan)

Program pengguna akan menerima bilangan integer  $n$ , kemudian menerima masukkan sebanyak  $n$  buah bilangan integer dan akan menampilkan hasil penjumlahan  $n$  buah bilangan integer tersebut. Tabel 2.3 menunjukkan contoh hasil masukkan dan keluaran *problem 1*.

Tabel 2.3: *Problem 1 (Penjumlahan)*

Sample Input	Sample Output
5 54 78 0 4 9	145

#### 2. Problem 2 (Max)

Program pengguna akan menerima bilangan integer  $n$ , kemudian menerima masukkan sebanyak  $n$  buah bilangan integer dan akan menampilkan hasil penjumlahan dari dua nilai tertinggi. Tabel 2.4 menunjukkan contoh hasil masukkan dan keluaran *problem 2*.

Tabel 2.4: *Problem 2 (Max)*

Sample Input	Sample Output
7 162 173 159 164 181 158 175	356

#### 3. Problem 3 Upload

Program pengguna akan menerima sebuah *file C* atau *zip*. *Problem* ini menggunakan pilihan *Upload Only* sehingga tidak dinilai oleh *SharIF Judge*.

Pengguna dapat menemukan *file zip* pada folder *Assignment*. Berikut susunan pohon dari tiga *problem* di atas:

```
.
|-- p1
|   |-- in
|       |-- input1.txt
|       |-- input2.txt
|       |-- input3.txt
|       |-- input4.txt
```



```

| | |-- input5.txt
| | |-- input6.txt
| | |-- input7.txt
| | |-- input8.txt
| | |-- input9.txt
| | --- input10.txt
| |-- out
| | --- output1.txt
| |-- tester.cpp
| --- desc.md
|-- p2
| |-- in
| | |-- input1.txt
| | |-- input2.txt
| | |-- input3.txt
| | |-- input4.txt
| | |-- input5.txt
| | |-- input6.txt
| | |-- input7.txt
| | |-- input8.txt
| | |-- input9.txt
| | --- input10.txt
| |-- out
| | |-- output1.txt
| | |-- output2.txt
| | |-- output3.txt
| | |-- output4.txt
| | |-- output5.txt
| | |-- output6.txt
| | |-- output7.txt
| | |-- output8.txt
| | |-- output9.txt
| | --- output10.txt
| |-- desc.md
| --- Problem2.pdf
|-- p3
| --- desc.md
--- SampleAssignment.pdf

```

*Problem 1* menggunakan metode "*Tester*" untuk memeriksa keluaran, sehingga memiliki *file tester.cpp* (*Tester Script*). *Problem 2* menggunakan metode *Output Comparison* untuk memeriksa keluaran, sehingga memiliki dua folder (*in* dan *out*) yang berisi *test case*. *Problem 3* menggunakan pilihan *Upload-Only*.

### 2.2.6 Test Structure

Saat menambah *assignments*, pengguna harus menyediakan *file zip* yang berisi *test cases*. *File zip* ini berisi folder-folder untuk setiap *problem*. Pengguna harus memberi nama folder sesuai aturan seperti *p1*, *p2*, *p3*, dst. *Assignment* yang menggunakan pilihan *Upload-Only* tidak membutuhkan *folder*.

## Metode Pengecekan

*SharIF Judge* memiliki dua metode pengecekan untuk setiap *problem* yaitu metode "*Input/Output Comparison*" dan metode "*Tester*".

### 1. Metode *Input/Output Comparison*

Dalam metode ini pengguna harus memasukkan beberapa *file input* dan beberapa *output* ke dalam folder *problem*. *SharIF Judge* akan memasukkan nilai dari *file input* ke kode *users* dan membandingkan hasil keluaran dari kode *users* dengan *file output*. *Input files* harus berada dalam folder "*in*" dengan nama *input1.txt*, *input2.txt*, dst. *Output files* harus berada dalam folder "*out*" dengan nama *output1.txt*, *output2.txt*, dst.

### 2. Metode *Tester*

Dalam metode ini pengguna harus memasukkan beberapa *file input*, sebuah *file C++* (*tester.cpp*), dan beberapa *file output*. *SharIF Judge* akan memasukkan nilai dari *file input* ke kode *users* dan mengambil keluaran dari kode *users*. *tester.cpp* akan mengambil nilai dari *file input*, *file output*, dan keluaran *users*. Jika keluaran dari kode *users* benar maka akan mengembalikan nilai 0, sedangkan jika salah akan mengembalikan nilai 1.

Pengguna dapat menggunakan kode pada listing 2.1 sebagai templat untuk menulis *tester.cpp*:

Listing 2.1: Contoh kode *tester.cpp*

```
/*
 * tester.cpp
 */

#include <iostream>
#include <fstream>
#include <string>
using namespace std;
int main(int argc, char const *argv[])
{
    ifstream test_in(argv[1]);    /* This stream reads from test's input
    file */
    ifstream test_out(argv[2]);   /* This stream reads from test's
    output file */
    ifstream user_out(argv[3]);   /* This stream reads from user's
    output file */

    /* Your code here */
    /* If user's output is correct, return 0, otherwise return 1 */

    ...

}
```

## Contoh *File*

Pengguna dapat menemukan contoh *file* penguji dalam folder *Assignments*. Berikut contoh susunan pohon dari *file* tersebut:

```

.
|-- p1
|   |-- in
|   |   |-- input1.txt
|   |   |-- input2.txt
|   |   |-- input3.txt
|   |   |-- input4.txt
|   |   |-- input5.txt
|   |   |-- input6.txt
|   |   |-- input7.txt
|   |   |-- input8.txt
|   |   |-- input9.txt
|   |   --- input10.txt
|   |-- out
|   |   --- output1.txt
|   --- tester.cpp
--- p2
|-- in
|   |-- input1.txt
|   |-- input2.txt
|   |-- input3.txt
|   |-- input4.txt
|   |-- input5.txt
|   |-- input6.txt
|   |-- input7.txt
|   |-- input8.txt
|   |-- input9.txt
|   --- input10.txt
--- out
|-- output1.txt
|-- output2.txt
|-- output3.txt
|-- output4.txt
|-- output5.txt
|-- output6.txt
|-- output7.txt
|-- output8.txt
|-- output9.txt
--- output10.txt

```

*Problem 1* menggunakan metode "*Tester*" untuk mengecek hasil keluarannya sehingga memiliki file *tester.cpp*. Listing 2.2 menunjukkan isi dari file *tester.cpp* untuk *Problem 1*:

Listing 2.2: Isi File *tester.cpp* untuk *Problem 1*

```

/*
 * tester.cpp
 */

#include <iostream>
#include <fstream>
#include <string>

```

```

using namespace std;
int main(int argc, char const *argv[])
{
    ifstream test_in(argv[1]);    /* This stream reads from test's input file
    */
    ifstream test_out(argv[2]);    /* This stream reads from test's output
    file */
    ifstream user_out(argv[3]);    /* This stream reads from user's output
    file */

    /* Your code here */
    /* If user's output is correct, return 0, otherwise return 1      */
    /* e.g.: Here the problem is: read n numbers and print their sum: */

    int sum, user_output;
    user_out >> user_output;

    if ( test_out.good() ) // if test's output file exists
    {
        test_out >> sum;
    }
    else
    {
        int n, a;
        sum=0;
        test_in >> n;
        for (int i=0 ; i<n ; i++){
            test_in >> a;
            sum += a;
        }
    }

    if (sum == user_output)
        return 0;
    else
        return 1;
}

```

*Problem 2* menggunakan metode "*Input/Output Comparison*" untuk mengecek hasil keluarannya sehingga memiliki dua folder *in* dan *out* yang berisi *test cases*. *Problem 3* menggunakan pilihan *Upload-Only* sehingga tidak memiliki folder apapun.

### 2.2.7 Deteksi Kecurangan

*SharIF Judge* menggunakan *Moss* untuk mendeteksi kode yang mirip. *Moss (Measure Of Software Similarity)* merupakan sistem otomatis untuk menentukan kemiripan program. Pada saat ini, aplikasi utama *Moss* telah digunakan untuk mendeteksi plagiarisme pada kelas *programming*. Pengguna dapat mengirimkan kode final (yang dipilih oleh *students* sebagai *Final Submission*) ke server *Moss* dengan satu klik.

Sebelum menggunakan *Moss* ada beberapa hal yang perlu diperhatikan yaitu:

- Pengguna harus mendapatkan *Moss user id* dan mengaturnya di *SharIF Judge*. Untuk mendapatkan *Moss user id*, pengguna harus terlebih dahulu mendaftar pada halaman <http://theory.stanford.edu/~aiken/moss/>. Pengguna akan menerima sebuah *email* yang berisi *script perl*. Dalam *script* tersebut terdapat *Moss user id*.

Listing 2.3 menunjukkan potongan *script perl* yang mengandung *user id*:

Listing 2.3: Potongan *script perl*

```
...

$server = 'moss.stanford.edu';
$port = '7690';
$noreq = "Request not sent.";
$usage = "usage: moss [-x] [-l language] [-d]
[-b basefile1] ... [-b basefileN] [-m #]
[-c \"string\"] file1 file2 file3 ...";

#
# The userid is used to authenticate your queries to the server;
# don't change it!
#
$userid=YOUR_MOSS_USER_ID;

#
# Process the command line options. This is done in a non-standard
# way to allow multiple -b's.
#
$opt_l = "c"; # default language is c
$opt_m = 10;
$opt_d = 0;

...
}
```

*User id* yang terdapat pada potongan *script perl* di atas, dapat digunakan pada *SharIF Judge* pada halaman *Moss*. *SharIF Judge* akan menggunakan *user id* tersebut di *Moss perl script*. *Perl* harus terinstal pada server agar dapat menggunakan *Moss*.

- Pengguna dianjurkan untuk mendeteksi kode yang mirip setelah waktu *assignment* selesai, karena para peserta masih dapat mengubah *Final Submissions* masing-masing sebelum waktu habis. Dengan cara tersebut, *SharIF Judge* dapat mengirimkan *Final submissions* para peserta ke server *Moss*.

### 2.2.8 Keamanan

Pengguna dapat mengatur keamanan *SharIF Judge* dengan beberapa langkah. Berikut adalah langkah-langkah yang dapat dilakukan.

#### Memakai *Sandbox*

Pengguna harus memakai *Sandbox* untuk *C/C++* dan mengaktifkan *Java Security Manager (Java Policy)* untuk *Java*. Penjelasan lebih lanjut tentang *Sandbox* ada pada point 2.2.9.

## Memakai *Shield*

*Shield* bukan merupakan pertahanan yang asli, tetapi lebih baik ada daripada tidak sama sekali. Pengguna dapat mengaktifkan *Shield* untuk *C*, *C++*, *Python*. Penjelasan lebih lanjut tentang *Shield* ada pada point [2.2.10](#).

## Jalankan sebagai *Non-Privileged User*

Pengguna diharapkan menjalankan kode yang sudah dikumpulkan sebagai *non-privileged user* - *user* yang tidak memiliki akses jaringan, tidak dapat menulis *file* apapun, dan tidak dapat membuat banyak proses.

Pengguna harus menjalankan *PHP* pada server sebagai *user www-data*. Membuat *user* baru dengan nama *restricted\_user* dan passwordnya. Jalankan *sudo visudo* dan tambahkan baris berikut pada akhir file *sudoers*:

```
www-data ALL=(restricted_user) NOPASSWD: ALL
```

Pada file *tester/runcode.sh* ganti

```
if $TIMEOUT_EXISTS; then
    timeout -s9 $((TIMELIMITINT*2)) $CMD <$IN >out 2>err
else
    $CMD <$IN >out 2>err
fi
```

menjadi

```
if $TIMEOUT_EXISTS; then
    sudo -u restricted_user timeout -s9 $((TIMELIMITINT*2)) $CMD <$IN >out 2>
    err
else
    sudo -u restricted_user $CMD <$IN >out 2>err
fi
```

dan *uncomment* baris berikut:

```
sudo -u restricted_user kill -9 -u restricted_user
```

- Menonaktifkan jaringan pada *restricted\_user*  
*restricted\_user* seharusnya tidak dapat mengakses jaringan. Pengguna dapat menonaktifkan jaringan untuk sebuah *user* pada *Linux* dengan menggunakan *iptables*. Setelah dinonaktifkan, uji dengan *ping* sebagai *restricted\_user*.
- Menolak izin menulis  
Pastikan tidak ada *file* atau *directory* yang dapat ditulis oleh *restricted\_user*.
- Membatasi jumlah proses  
Pengguna dapat membatasi jumlah proses dengan cara menambahkan baris berikut pada file */etc/security/limits.conf*:

<i>restricted_user</i>	<i>soft</i>	<i>nproc</i>	3
<i>restricted_user</i>	<i>hard</i>	<i>nproc</i>	5

## Memakai Dua *Server*

Pengguna dapat memakai sebuah *server* untuk antarmuka web dan menangani permintaan web, dan memakai *server* lainnya untuk menjalankan kode yang dikumpulkan. Hal ini mengurangi risiko menjalankan kode yang dikumpulkan. Untuk memakai dua *server* pengguna harus mengganti *source code SharIF Judge*.

### 2.2.9 *Sandbox*

*SharIF Judge* menjalankan banyak *arbitrary codes* yang *user* kumpulkan sehingga memerlukan perangkat untuk kode *sandbox* yang sudah dikumpulkan. Pengguna dapat meningkatkan keamanan dengan cara mengaktifkan *shield* dan *sandbox* secara bersamaan.

#### *C/C++ Sandboxing*

*SharIF Judge* menggunakan *EasySandbox* untuk *sandboxing* kode *C/C++*. *EasySandbox* membatasi jalannya kode mengguna *seccomp*, mekanisme *sandboxing* pada *Linux kernel*.

Secara standar, *EasySandbox* dinonaktifkan pada *SharIF Judge*. Pengguna dapat mengaktifkannya pada halaman *Settings*. Pengguna harus "*Build EasySandbox*" sebelum mengaktifkannya.

**Build EasySandbox** *EasySandbox files* terletak pada *tester/easysandbox*. Untuk membangun *EasySandbox* jalankan:

```
$ cd tester/easysandbox
$ chmod +x runalltests.sh
$ chmod +x runtest.sh
$ make runtests
```

Jika pesan yang ditampilkan adalah "*All tests passed!*", maka *EasySandbox* telah berhasil dibangun dan dapat dipakai pada sistem. Pengguna dapat mengaktifkan *EasySandbox* pada halaman *Settings*.

#### *Java Sandboxing*

*Java Sandbox* dapat diaktifkan secara standar menggunakan *Java Security Manager*. Pengguna dapat mengaktifkan/menonaktifkan hal ini pada halaman *Settings*.

### 2.2.10 *Shield*

*Shield* merupakan mekanisme sederhana yang melarang jalannya kode yang berbahaya. *Shield* bukan solusi *sandboxing*. *Shield* hanya menyediakan sebagian pertahanan terhadap serangan sepele. Pertahanan yang asli terhadap kode yang tidak terpercaya hanya didapatkan dengan mengaktifkan *Sandbox*.

#### *Shield untuk C/C++*

Dengan mengaktifkan *Shield* untuk *C/C++*, *SharIF Judge* hanya menambahkan *#define* pada bagian awal kode *C/C++* sebelum menjalankannya.

Pengguna dapat melarang menggunakan *goto* dengan cara menambahkan baris berikut pada bagian awal kode yang dikumpulkan:

```
#define goto YouCannotUseGoto
```

Dengan baris ini pada awal *files*, semua kode yang sudah dikumpulkan yang menggunakan *goto* akan mendapatkan *compilation error*.

Jika pengguna menggunakan *Shield*, semua kode yang berisi *#undef* akan mendapatkan *compilation error*.

### Mengaktifkan *Shield* untuk *C* atau *C++*

Pengguna dapat mengaktifkan dan menonaktifkan *Shield* pada halaman *Settings*.

### Menambahkan aturan untuk *C/C++*

Daftar aturan *#define* terletak pada file *tester/shield/defc.h* (untuk *C*) dan *tester/shield/defcpp.h* (untuk *C++*). Pengguna dapat menambahkan aturan *#define* baru pada file ini. Konten dari file ini dapat diganti pada halaman *Settings*.

Berikut adalah sintaksis yang digunakan pada *files* tersebut:

```
/*

@file defc.h
There should be a newline at end of this file.
Put the message displayed to user after // in each line

e.g. If you want to disallow goto, add this line:
#define goto errorNo13    //Goto is not allowed

*/

#define system errorNo1    //"system" is not allowed
#define freopen errorNo2   //File operation is not allowed
#define fopen errorNo3     //File operation is not allowed
#define fprintf errorNo4   //File operation is not allowed
#define fscanf errorNo5    //File operation is not allowed
#define feof errorNo6      //File operation is not allowed
#define fclose errorNo7    //File operation is not allowed
#define ifstream errorNo8  //File operation is not allowed
#define ofstream errorNo9  //File operation is not allowed
#define fork errorNo10     //Fork is not allowed
#define clone errorNo11    //Clone is not allowed
#define sleep errorNo12    //Sleep is not allowed
```

Pada akhir *files defc.h* dan *defcpp.h* harus ada baris baru. Ada banyak aturan yang tidak dapat dipakai pada *g++*. Sebagai contoh pengguna tidak dapat memakai *#define fopen errorNo3* pada *C++* karena hasilnya *compile error*.

### *Shield* untuk *Python*

Dengan mengaktifkan *Shield* untuk *Python*, *SharIF Judge* hanya menambahkan beberapa kode pada bagian awal kode *Python* yang telah dikumpulkan sebelum dijalankan untuk mencegah penggunaan fungsi yang berbahaya. Kode tersebut terletak pada *files tester/shield/shield\_py2.py* dan *tester/shield/shield\_py3.py*. Pengguna dapat mengaktifkan/menonaktifkan *Shield* untuk *Python* pada halaman *Settings*.

Berikut adalah cara untuk keluar dari *Shield* dan menggunakan fungsi yang telah di *blacklist*:



```
# @file shield_py3.py

import sys
sys.modules['os']=None

BLACKLIST = [
    '__import__', # deny importing modules
    'eval', # eval is evil
    'open',
    'file',
    'exec',
    'execfile',
    'compile',
    'reload',
    'input'
]

for func in BLACKLIST:
    if func in __builtins__.__dict__:
        del __builtins__.__dict__[func]
```



## BAB 3

### ANALISIS

#### 3.1 Tingkat Kepatuhan *SharIF Judge*

Kesuksesan aplikasi *SharIF Judge* dalam mematuhi kriteria sukses *WCAG* 2.1 ditulis dalam tabel-tabel berikut.

Tabel 3.1: Tabel kepatuhan *SharIF Judge* terhadap prinsip *Perceivable*

Kriteria Sukses	Tingkat Kepatuhan	Hasil
1.1.1	A	Tidak Sukses
1.2.1	A	Sukses
1.2.2	A	Sukses
1.2.3	A	Sukses
1.2.4	AA	Sukses
1.2.5	AA	Sukses
1.2.6	AAA	Sukses
1.2.7	AAA	Sukses
1.2.8	AAA	Sukses
1.2.9	AAA	Sukses
1.3.1	A	Tidak Sukses
1.3.2	A	Sukses
1.3.3	A	Sukses
1.3.4	A	Sukses
1.3.5	AA	Tidak Sukses
1.3.6	AAA	Tidak Sukses
1.4.1	A	Sukses
1.4.2	A	Sukses
1.4.3	AA	Tidak Sukses
1.4.4	AA	Tidak Sukses
1.4.5	AA	Sukses
1.4.6	AAA	Tidak Sukses
1.4.7	AAA	Sukses
1.4.8	AAA	Tidak Sukses
1.4.9	AAA	Sukses
1.4.10	AA	Tidak Sukses
1.4.11	AA	Diabaikan
1.4.12	AA	Sukses
1.4.13	AA	Sukses
Tingkat kepatuhan tertinggi yang dicapai		-

Tabel 3.2: Tabel kepatuhan *SharIF Judge* terhadap prinsip *Operable*

Kriteria Sukses	Tingkat Kepatuhan	Hasil
2.1.1	A	Tidak Sukses
2.1.2	A	Tidak Sukses
2.1.3	AAA	Tidak Sukses
2.1.4	A	Sukses
2.2.1	A	Sukses
2.2.2	A	Sukses
2.2.3	AAA	Sukses
2.2.4	AAA	Sukses
2.2.5	AAA	Tidak Sukses
2.2.6	AAA	Tidak Sukses
2.3.1	A	Sukses
2.3.2	AAA	Sukses
2.3.3	AAA	Sukses
2.4.1	A	Tidak Sukses
2.4.2	A	Sukses
2.4.3	AA	Sukses
2.4.4	A	Tidak Sukses
2.4.5	AA	Tidak Sukses
2.4.6	AA	Tidak Sukses
2.4.7	AA	Tidak Sukses
2.4.8	AAA	Sukses
2.4.9	AAA	Tidak Sukses
2.4.10	AAA	Tidak Sukses
2.5.1	A	Sukses
2.5.2	A	Sukses
2.5.3	A	Sukses
2.5.4	A	Sukses
2.5.5	AAA	Tidak Sukses
2.5.6	AAA	Sukses
Tingkat kepatuhan tertinggi yang dicapai		-

Tabel 3.3: Tabel kepatuhan *SharIF Judge* terhadap prinsip *Understandable*

Kriteria Sukses	Tingkat Kepatuhan	Hasil
3.1.1	A	Tidak Sukses
3.1.2	AA	Tidak Sukses
3.1.3	AAA	Sukses
3.1.4	AAA	Sukses
3.1.5	AAA	Sukses
3.1.6	AAA	Sukses
3.2.1	A	Sukses
3.2.2	A	Sukses
3.2.3	AA	Sukses
3.2.4	AA	Tidak Sukses
3.2.5	AAA	Sukses
3.3.1	A	Sukses
3.3.2	A	Tidak Sukses
3.3.3	AA	Sukses
3.3.4	AA	Sukses
3.3.5	AAA	Tidak Sukses
3.3.6	AAA	Sukses
Tingkat kepatuhan tertinggi yang dicapai		-

Tabel 3.4: Tabel kepatuhan *SharIF Judge* terhadap prinsip *Robust*

Kriteria Sukses	Tingkat Kepatuhan	Hasil
4.1.1	A	Tidak Sukses
4.1.2	A	Sukses
4.1.3	AA	Tidak Sukses
Tingkat kepatuhan tertinggi yang dicapai		-

### 3.1.1 *Perceivable*

#### Kriteria Sukses 1.1.1 Non-text Content

(Tidak Sukses)

Kriteria ini tidak sukses dipatuhi karena:

- Gambar logo *SharIF Judge* tidak memiliki alternatif teks. Kesalahan dapat dilihat pada potongan kode 3.1.

Listing 3.1: Kriteria Sukses 1.1.1 - Logo SharIF Judge Tidak Diberi Teks Alternatif

```

...
<a href="{{ site_url('/') }}">
    
    <h1 class="shjlogo-text">SharIF <span>Judge</span></h1>
</a>
...

```

- Gambar pdf pada halaman *Assignment* tidak memiliki teks alternatif. Kesalahan dapat dilihat pada potongan kode 3.2.

Listing 3.2: Kriteria Sukses 1.1.1 - Gambar PDF Tidak Diberi Teks Alternatif

```
...  
<td>  
  <a href="{{ site_url('assignments/pdf/~item.id) }}"></a>  
</td>  
...
```

**Kriteria Sukses 1.2.1 Audio-only dan Video-only (Prerecorded)**

(Sukses)

Kriteria ini sukses dipatuhi karena pada aplikasi *SharIF Judge* tidak terdapat konten berbasis waktu.

**Kriteria Sukses 1.2.2 Captions (Prerecorded)**

(Sukses)

Kriteria ini sukses dipatuhi karena pada aplikasi *SharIF Judge* tidak terdapat konten berbasis waktu.

**Kriteria Sukses 1.2.3 Audio Descriptive atau Media Alternative (Prerecorded)**

(Sukses)

Kriteria ini sukses dipatuhi karena pada aplikasi *SharIF Judge* tidak terdapat konten berbasis waktu.

**Kriteria Sukses 1.2.4 Captions (Live)**

(Sukses)

Kriteria ini sukses dipatuhi karena pada aplikasi *SharIF Judge* tidak terdapat konten berbasis waktu.

**Kriteria Sukses 1.2.5 Audio Description (Prerecorded)**

(Sukses)

Kriteria ini sukses dipatuhi karena pada aplikasi *SharIF Judge* tidak terdapat konten berbasis waktu.

**Kriteria Sukses 1.2.6 Sign Language (Prerecorded)**

(Sukses)

Kriteria ini sukses dipatuhi karena pada aplikasi *SharIF Judge* tidak terdapat konten berbasis waktu.

**Kriteria Sukses 1.2.7 Extended Audio Description (Prerecorded)**

(Sukses)

Kriteria ini sukses dipatuhi karena pada aplikasi *SharIF Judge* tidak terdapat konten berbasis waktu.

**Kriteria Sukses 1.2.8 Media Alternative (Prerecorded)**

(Sukses)

Kriteria ini sukses dipatuhi karena pada aplikasi *SharIF Judge* tidak terdapat konten berbasis waktu.

**Kriteria Sukses 1.2.9 Audio-only (Live)**

(Sukses)

Kriteria ini sukses dipatuhi karena pada aplikasi *SharIF Judge* tidak terdapat konten berbasis waktu.

**Kriteria Sukses 1.3.1 Info dan Relationships**

(Tidak Sukses)

Kriteria ini tidak sukses dipatuhi karena:

- Ada elemen dalam form *Add Users* yang tidak diberi label. Tampilan halaman web dapat dilihat pada gambar 3.1.

Gambar 3.1: Kriteria Sukses 1.3.1 - Elemen yang Tidak Diberi Label Pada Halaman *Add Users*

- Dalam halaman *Add Assignment* ada elemen dalam form *problems* tidak diberi label. Selain itu masih ada juga bidang masukan *required* tetapi tidak diberikan tanda *asterisk*(\*) dan diberi warna merah. Kesalahan dapat dilihat pada potongan kode 3.3.

Listing 3.3: Kriteria Sukses 1.3.1 - Elemen Dalam Form yang Tidak Diberi Label

```

...
<td><input type="text" name="name[]" class="sharif_input short"
value="Problem "/></td>\
...

```

- Dalam halaman *Problem* terdapat elemen form yang tidak diberi label.



Gambar 3.2: Kriteria Sukses 1.3.1 - Elemen yang Tidak Diberi Label Pada Halaman *Problems* Bagian *Edit Markdown*



Gambar 3.3: Kriteria Sukses 1.3.1 - Elemen yang Tidak Diberi Label Pada Halaman *Problems* Bagian *Edit Plain Html*

### Kriteria Sukses 1.3.2 Meaningful Sequence

(Sukses)



Kriteria ini sukses dipatuhi karena setiap halaman pada *SharIF Judge* memiliki urutan baca dan navigasi yang benar.

### Kriteria Sukses 1.3.3 Sensory Characteristics

(Sukses)

Kriteria ini sukses dipatuhi karena petunjuk yang diberikan pada aplikasi *SharIF Judge* tidak hanya bergantung pada komponen karakteristik sensorik seperti bentuk, warna, ukuran, lokasi visual, orientasi, atau suara.

### Kriteria Sukses 1.3.4 Orientation

(Sukses)

Kriteria ini sukses dipatuhi karena konten dari *SharIF Judge* dapat ditampilkan dalam orientasi *portrait* atau *landscape*.

### Kriteria Sukses 1.3.5 Identify Input Purpose

(Tidak Sukses)

Kriteria ini tidak sukses karena masih ada bidang masukan yang tidak memiliki label, salah satunya pada halaman *Problems* bagian *Edit Markdown*. Kesalahan dapat dilihat pada potongan kode 3.4.

Listing 3.4: Kriteria Sukses 1.3.5 - Elemen Tidak Diberi Label Pada Halaman *Problems* Bagian *Edit Markdown*

```
...
<p class="input_p">
    <textarea name="text">{{ problem.description }}</textarea>
</p>
...
```

### Kriteria Sukses 1.3.6 Identify Purpose

(Tidak Sukses)

Kriteria ini tidak sukses dipatuhi karena ada elemen *HTML* yang seharusnya dapat dipakai tetapi tidak digunakan, seperti bagian navigasi dan judul bagian. Kesalahan dapat dilihat pada potongan kode 3.5.

Listing 3.5: Kriteria Sukses 1.3.6 - *Sidebar*

```
...
<div id="side_bar" class="sidebar_open">
    <ul>
        <li class="color-dashboard{{ selected=='dashboard' ? ' selected'
        }}">
    ...
```

### Kriteria Sukses 1.4.1 Use of Color

(Sukses)

Kriteria ini sukses dipatuhi karena warna tidak digunakan sebagai satu-satunya cara untuk menyampaikan informasi, menunjukkan aksi, menampilkan respon, atau membedakan elemen visual. Selain warna, ada juga teks yang menjelaskan informasi yang ditampilkan.

### Kriteria Sukses 1.4.2 Audio Control

(Sukses)

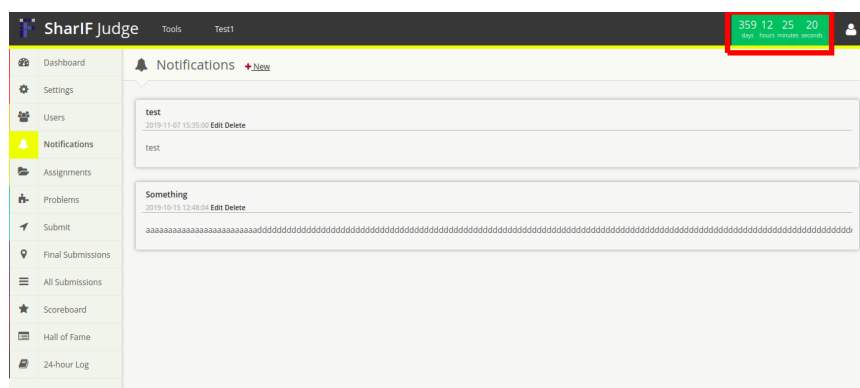
Kriteria ini sukses dipatuhi karena pada aplikasi *SharIF Judge* tidak terdapat konten berbasis waktu.

### Kriteria Sukses 1.4.3 Contrast (Minimum)

(Tidak Sukses)

Kriteria ini tidak sukses dipatuhi karena pada aplikasi *SharIF Judge* terdapat beberapa teks dengan rasio kontras kurang dari 4.5:1, antara lain:

- Navigasi Atas: Teks tempat waktu yang tersisa untuk mengumpulkan suatu *Assignment* memiliki rasio kontras 2.52:1 terhadap warna latar belakangnya. Tampilan halaman web dapat dilihat pada gambar 3.4.



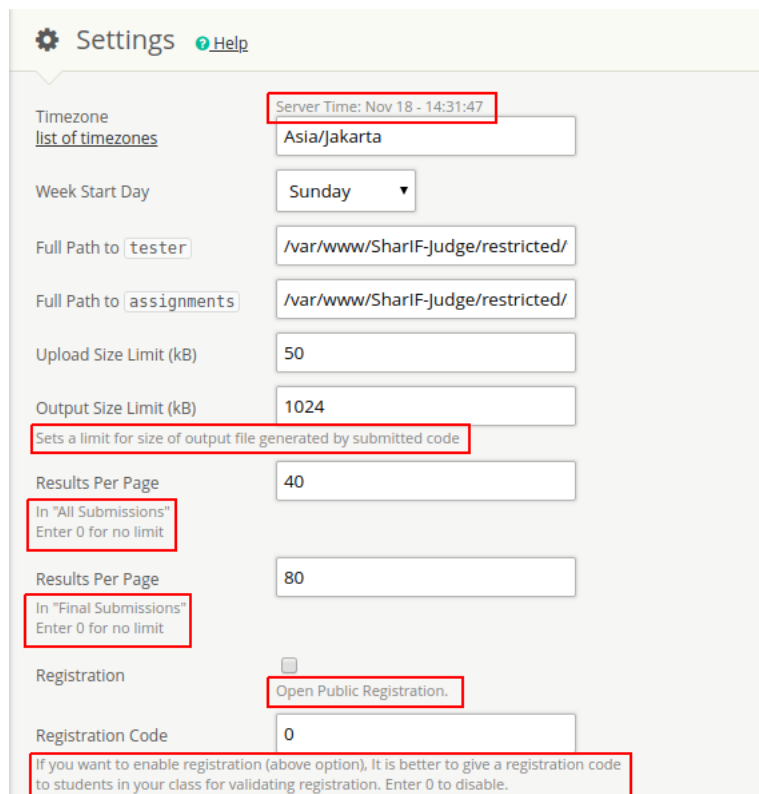
Gambar 3.4: Kriteria Sukses 1.4.3 - Kontras Elemen Navigasi Atas

- Halaman *Dashboard*: Teks pada tombol *today* memiliki rasio kontras 4.01:1 terhadap warna latar belakangnya. Teks pada kalender memiliki rasio kontras 2.69:1 terhadap warna latar belakangnya. Teks pada kalender yang menunjukkan tanggal memiliki rasio kontras 1.57:1 terhadap warna latar belakangnya. Teks yang menunjukkan waktu notifikasi dibuat memiliki rasio kontras 2.07:1 terhadap warna latar belakangnya. Tampilan halaman web dapat dilihat pada gambar 3.5.



Gambar 3.5: Kriteria Sukses 1.4.3 - Kontras Elemen Dashboard

- Halaman *Settings*: Informasi tambahan pada bidang masukkan memiliki rasio kontras 3.55:1 terhadap warna latar belakangnya. Informasi peringatan pada bidang masukkan memiliki rasio kontras 3.69:1 terhadap warna latar belakangnya. Tampilan halaman web dapat dilihat pada gambar 3.6.



Gambar 3.6: Kriteria Sukses 1.4.3 - Kontras Elemen Settings

- Halaman *Notifications*: Teks yang menunjukkan waktu notifikasi dibuat memiliki rasio kontras 1.91:1 terhadap warna latar belakangnya. Tampilan halaman web dapat dilihat pada gambar 3.7.



Gambar 3.7: Kriteria Sukses 1.4.3 - Kontras Elemen Notifications

- Halaman *Assignment*: Teks yang berwarna merah pada tabel *Assignment* memiliki rasio kontras 3.89:1 terhadap warna latar belakangnya. Tampilan halaman web dapat dilihat pada gambar 3.8.

Select	Name	Problems	Submissions	Coefficient	Start Time	Finish Time	Status	PDF	Actions
<input type="checkbox"/>	a	1 problem	0 submissions	Finished	2019-09-08 00:00:00	2019-10-06 00:00:00	Close		
<input checked="" type="checkbox"/>	Test1	1 problem	0 submissions	100 %	2019-09-09 00:00:00	2020-11-12 00:00:00	Open		
<input type="checkbox"/>	Test	1 problem	0 submissions	Finished	2019-02-04 00:00:00	2019-06-18 00:00:00	Close		

Gambar 3.8: Kriteria Sukses 1.4.3 - Kontras Elemen Assignments

- Halaman *Add Assignment*: Informasi tambahan pada bidang masukkan memiliki rasio kontras 3.55:1 terhadap warna latar belakangnya. Informasi peringatan pada bidang masukkan memiliki rasio kontras 3.69:1 terhadap warna latar belakangnya. Tampilan halaman web dapat dilihat pada gambar 3.9.

**Add Assignment** [Help](#)

Assignment Name

Start Time

Finish Time

Extra Time (minutes)   
Extra time for late submissions.

Participants   
Enter username of participants here (comma separated). Only these users are able to submit. You can use keyword "ALL".

Tests and Descriptions (zip file)  No file chosen  
[Use this structure](#)

PDF File  No file chosen  
PDF File of Assignment

☐ Open [Open or close this assignment](#)

☐ Scoreboard [Check this to enable scoreboard](#)

☐ Java Exceptions [Check this to show Java exceptions to users](#)

☐ Archived Assignment [Check this to make an archived assignment](#)

Coefficient rule (?)  
[PHP script without <?php ?> tags](#)

```
/*
 * Put coefficient (from 100) in variable $coefficient.
 * You can use variables $extra_time and $delay.
 * $extra_time is the total extra time given to users
 * (in seconds) and $delay is number of seconds passed
 * from finish time (can be negative).
 * In this example, $extra_time is 172800 (2 days):
 */

if ($delay <= 0)
// no delay
$coefficient = 100;

elseif ($delay <= 3600)
// delay less than 1 hour
$coefficient = ceil(100 - ((30 * $delay) / 3600));

elseif ($delay <= 86400)
// delay more than 1 hour and less than 1 day
$coefficient = 70;
```

Problems [+](#)

	Name	Score	Time Limit (ms)			Memory Limit (kB)	Allowed Languages (?)	Diff Command (?)	Diff Argument (?)	Upload Only (?)
			C/C++	Python	Java					
1	Problem	100	500	1500	2000	50000	C,C++,Python 2,Python 3,J	diff	-bB	<input type="checkbox"/> <a href="#">x</a>

Gambar 3.9: Kriteria Sukses 1.4.3 - Kontras Elemen *Add Assignments***Kriteria Sukses 1.4.4 Resize text**

(Tidak Sukses)

Kriteria ini tidak sukses dipatuhi karena fungsionalitas *navigation bar* tidak berjalan pada pembesaran 125 persen. Tampilan halaman web dapat dilihat pada gambar 3.10.



Gambar 3.10: Kriteria Sukses 1.4.4 - Sidenav

### Kriteria Sukses 1.4.5 Images of Text

(Sukses)

Kriteria ini sukses dipatuhi karena semua gambar teks yang ada pada aplikasi *SharIF Judge* penting untuk informasi yang disampaikan.

### Kriteria Sukses 1.4.6 Contrast (Enhanced)

(Tidak Sukses)

Kriteria ini tidak sukses dipatuhi karena pada aplikasi *SharIF Judge* terdapat beberapa teks dengan rasio kontras kurang dari 4.5:1, antara lain:

- Navigasi Atas: Teks tempat waktu yang tersisa untuk mengumpulkan suatu *Assignment* memiliki rasio kontras 2.52:1 terhadap warna latar belakangnya. Tampilan halaman web dapat dilihat pada gambar 3.11.







Select	Name	Problems	Submissions	Coefficient	Start Time	Finish Time	Status	PDF	Actions
<input type="checkbox"/>	a	1 problem	0 submissions	Finished	2019-09-08 00:00:00	2019-10-06 00:00:00	Close		
<input checked="" type="checkbox"/>	Test1	1 problem	0 submissions	100 %	2019-09-09 00:00:00	2020-11-12 00:00:00	Open		
<input type="checkbox"/>	Test	1 problem	0 submissions	Finished	2019-02-04 00:00:00	2019-06-18 00:00:00	Close		

Gambar 3.15: Kriteria Sukses 1.4.6 - Kontras Elemen Assignments

- Halaman *Add Assignment*: Informasi tambahan pada bidang masukkan memiliki rasio kontras 3.55:1 terhadap warna latar belakangnya. Informasi peringatan pada bidang masukkan memiliki rasio kontras 3.69:1 terhadap warna latar belakangnya. Tampilan halaman web dapat dilihat pada gambar 3.16.

**Add Assignment** [Help](#)

Assignment Name:

Start Time:

Finish Time:

Extra Time (minutes):  Extra time for late submissions.

Participants:  Enter username of participants here (comma separated). Only these users are able to submit. You can use keyword "ALL".

Tests and Descriptions (zip file):  No file chosen Use this structure

PDF File:  No file chosen PDF File of Assignment

**Coefficient rule (?)** PHP script without <?php ?> tags

```
/*
 * Put coefficient (from 100) in variable $coefficient.
 * You can use variables $extra_time and $delay.
 * $extra_time is the total extra time given to users
 * (in seconds) and $delay is number of seconds passed
 * from finish time (can be negative).
 * In this example, $extra_time is 172800 (2 days):
 */

if ($delay<=0)
// no delay
$coefficient = 100;

elseif ($delay<=3600)
// delay less than 1 hour
$coefficient = ceil(100-((30*$delay)/3600));

elseif ($delay<=86400)
// delay more than 1 hour and less than 1 day
$coefficient = 70;
```

**Problems** [+](#)

	Name	Score	Time Limit (ms)			Memory Limit (kB)	Allowed Languages (?)	Diff Command (?)	Diff Argument (?)	Upload Only (?)
			C/C++	Python	Java					
1	Problem	100	500	1500	2000	50000	C,C++,Python 2,Python 3,J	diff	-bB	<input type="checkbox"/>

Gambar 3.16: Kriteria Sukses 1.4.6 - Kontras Elemen Add Assignments

### Kriteria Sukses 1.4.7 Low atau No Background Audio

(Sukses)

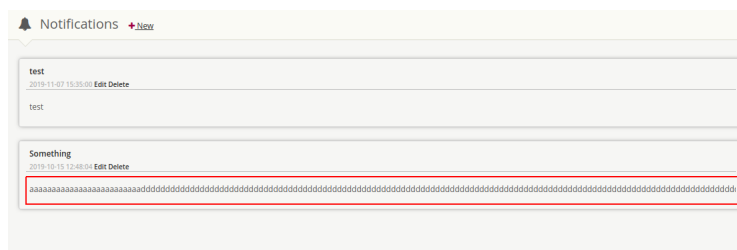
Kriteria ini sukses dipatuhi karena pada aplikasi *SharIF Judge* tidak terdapat konten berbasis waktu.

### Kriteria Sukses 1.4.8 Visual Presentation

(Tidak Sukses)

Kriteria ini tidak sukses dipatuhi karena pada halaman *Dashboard* dan *Notifications* teks yang menampilkan isi notifikasi memiliki lebar lebih dari 80 karakter. Tampilan halaman web ditampilkan

pada gambar 3.17.



Gambar 3.17: Kriteria Sukses 1.4.8 - Notifications

### Kriteria Sukses 1.4.9 Images of Text (No Exception)

(Sukses)

Kriteria ini sukses dipatuhi karena semua gambar teks yang ada pada aplikasi *SharIF Judge* penting untuk informasi yang disampaikan.

### Kriteria Sukses 1.4.10 Reflow

(Tidak Sukses)

Kriteria ini tidak sukses dipatuhi karena pada setiap halaman aplikasi *SharIF Judge* memerlukan *scroll* secara horizontal ketika ditampilkan pada resolusi layar 1280 pixel dengan pembesaran 400 persen. Tampilan halaman web dapat dilihat pada gambar 3.18

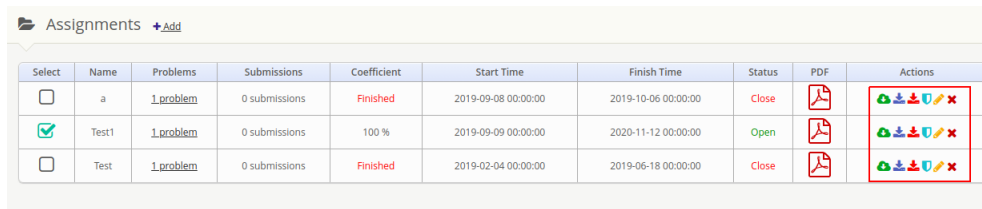
Select	Name	Problems	Submissions	Coefficient	Time	Time	St
<input type="checkbox"/>	a	1 <u>problem</u>	0 submissions	Finished	2019-09-08 00:00:00	2019-10-06 00:00:00	Cl
<input checked="" type="checkbox"/>	Test1	1 <u>problem</u>	0 submissions	100 %	2019-09-09 00:00:00	2020-11-12 00:00:00	O
<input type="checkbox"/>	Test	1 <u>problem</u>	0 submissions	Finished	2019-02-04 00:00:00	2019-06-18 00:00:00	Cl

Gambar 3.18: Kriteria Sukses 1.4.10 - Horizontal *Scroll*

### Kriteria Sukses 1.4.11 Non-text Contrast

(Diabaikan)

Kriteria ini diabaikan karena pengukuran kontras sulit untuk dilakukan. Tampilan halaman web dapat dilihat pada gambar 3.19.



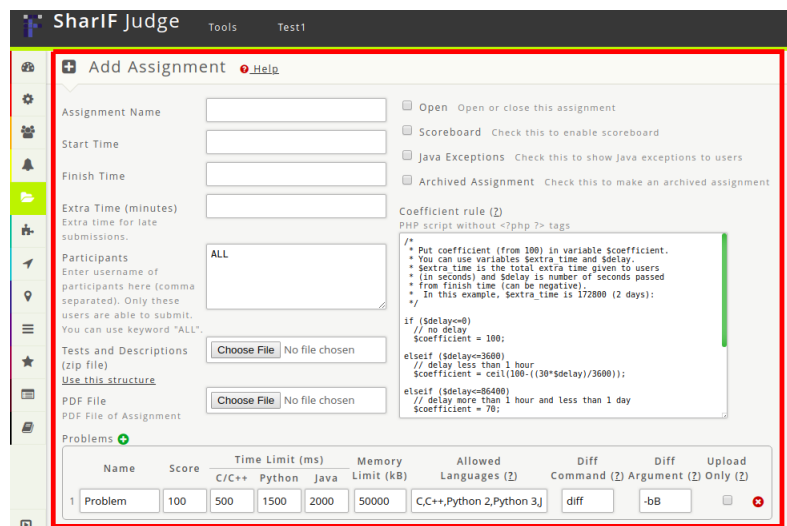
Select	Name	Problems	Submissions	Coefficient	Start Time	Finish Time	Status	PDF	Actions
<input type="checkbox"/>	a	1 problem	0 submissions	Finished	2019-09-08 00:00:00	2019-10-06 00:00:00	Close		
<input checked="" type="checkbox"/>	Test1	1 problem	0 submissions	100 %	2019-09-09 00:00:00	2020-11-12 00:00:00	Open		
<input type="checkbox"/>	Test	1 problem	0 submissions	Finished	2019-02-04 00:00:00	2019-06-18 00:00:00	Close		

Gambar 3.19: Kriteria Sukses 1.4.11 - Ikon Pada Halaman *Assignments*

### Kriteria Sukses 1.4.12 Text Spacing

(Sukses)

Kriteria ini sukses dipatuhi karena penulis telah melakukan uji coba pada salah satu halaman agar kriteria ini terpenuhi dan hasilnya tidak terjadi masalah. Oleh karena itu, penulis menyimpulkan bahwa jika perubahan dilakukan di halaman lain akan menghasilkan hal yang sama. Tampilan halaman web dapat dilihat pada gambar 3.20.



**SharIF Judge** Tools Test1

**Add Assignment** [Help](#)

Assignment Name:

Start Time:

Finish Time:

Extra Time (minutes):

Participants:

Tests and Descriptions (zip file):  No file chosen

PDF File:  No file chosen

Open: ☐ Open or close this assignment

Scoreboard: ☐ Check this to enable scoreboard

Java Exceptions: ☐ Check this to show Java exceptions to users

Archived Assignment: ☐ Check this to make an archived assignment

Coefficient rule (2):  
 PHP script without <?php ?> tags  

```

/*
 * Put coefficient (from 100) in variable $coefficient.
 * You can use variables $extra_time and $delay.
 * $extra_time is the total extra time given to users
 * (in seconds) and $delay is number of seconds passed
 * from finish time (can be negative).
 * In this example, $extra_time is 172800 (2 days):
 */
if ($delay==0)
// no delay
$coefficient = 100;
elseif ($delay<3600)
// delay less than 1 hour
$coefficient = ceil(100-((30*$delay)/3600));
elseif ($delay<86400)
// delay more than 1 hour and less than 1 day
$coefficient = 70;

```

Name	Score	Time Limit (ms)	Memory Limit (kB)	Allowed Languages (2)	Diff Command (2)	Diff Argument (2)	Upload Only (2)
1 Problem	100	500	1500	2000	50000	C,C++,Python 2,Python 3,J	diff -bB

Gambar 3.20: Kriteria Sukses 1.4.12 - Halaman *Add Assignment*

### Kriteria Sukses 1.4.13 Content on Hover or Focus

(Sukses)

Kriteria ini sukses dipatuhi karena setiap konten tambahan yang muncul sesaat ketika suatu elemen menerima penunjuk kursor atau fokus *keyboard*, konten tambahan tersebut dapat disingkirkan, dapat ditunjuk, dan persisten.

### 3.1.2 Operable

#### Kriteria Sukses 2.1.1 Keyboard

(Tidak Sukses)

Kriteria ini tidak sukses dipatuhi karena:

- Tombol *Collapse Menu* pada *sidebar* tidak dapat dioperasikan dengan keyboard. Tampilan halaman web dapat dilihat pada gambar 3.21.



Gambar 3.21: Kriteria Sukses 2.1.1 - *Collapse Menu*

- Tombol *Tools* pada *Menu* tidak dapat dioperasikan dengan keyboard. Tampilan halaman web dapat dilihat pada gambar 3.22.



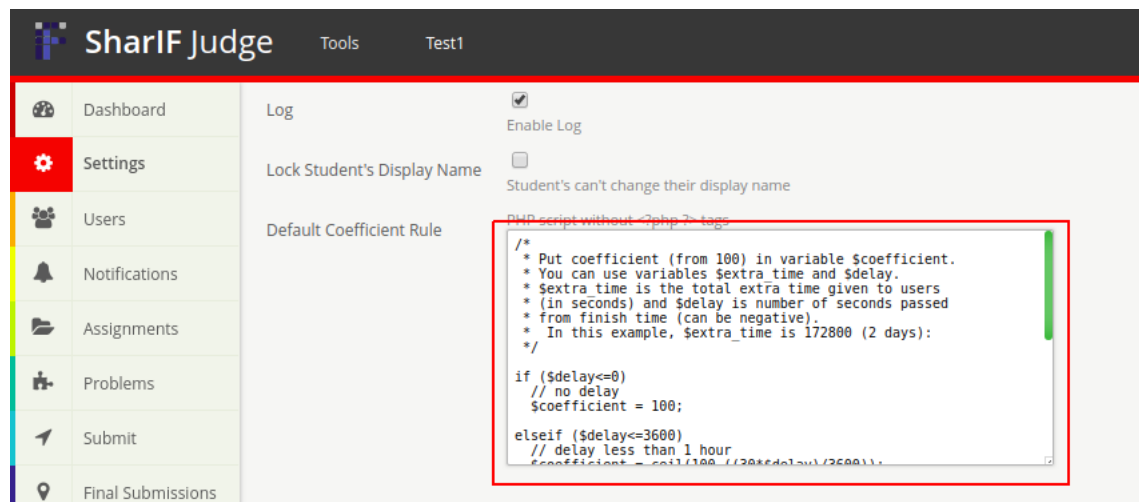
Gambar 3.22: Kriteria Sukses 2.1.1 - *Tools*

### Kriteria Sukses 2.1.2 No Keyboard Trap

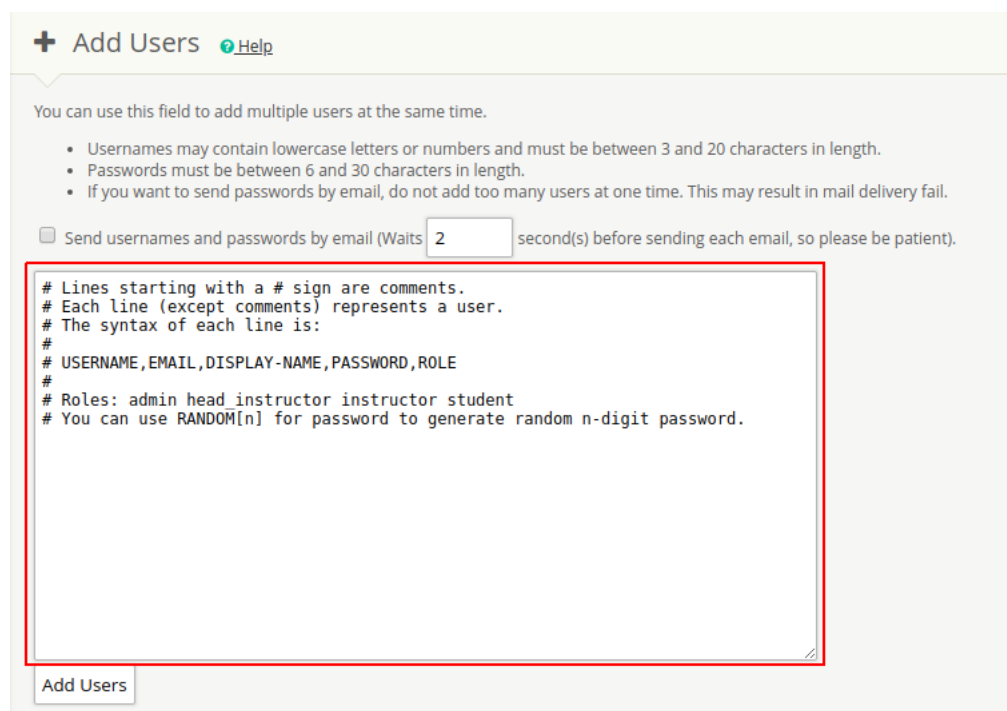
(Tidak Sukses)

Kriteria ini tidak sukses dipatuhi karena:

- Dalam halaman *Settings* ada *input field* yang fokusnya tidak dapat dipindahkan menggunakan antarmuka *keyboard*. *Input field* yang dimaksud dapat dilihat pada gambar 3.23.

Gambar 3.23: Kriteria Sukses 2.1.2 - *Settings*

- Dalam halaman *Add User* ada *input field* yang fokusnya tidak dapat dipindahkan menggunakan antarmuka *keyboard*. *Input field* yang dimaksud dapat dilihat pada gambar 3.24.

Gambar 3.24: Kriteria Sukses 2.1.2 - *Add User*

- Dalam halaman *Add Assignment* ada *input field* yang fokusnya tidak dapat dipindahkan menggunakan antarmuka *keyboard*. *Input field* yang dimaksud dapat dilihat pada gambar 3.25.

**Add Assignment** [Help](#)

Assignment Name

Start Time

Finish Time

Extra Time (minutes)   
Extra time for late submissions.

Participants   
Enter username of participants here (comma separated). Only these users are able to submit. You can use keyword "ALL".

Tests and Descriptions (zip file)  No file chosen  
[Use this structure](#)

PDF File  No file chosen  
PDF File of Assignment

☐ Open Open or close this assignment

☐ Scoreboard Check this to enable scoreboard

☐ Java Exceptions Check this to show java exceptions to users

☐ Archived Assignment Check this to make an archived assignment

Coefficient rule (?)  
PHP script without `<?php ?>` tags

```
/*
 * Put coefficient (from 100) in variable $coefficient.
 * You can use variables $extra time and $delay.
 * $extra time is the total extra time given to users
 * (in seconds) and $delay is number of seconds passed
 * from finish time (can be negative).
 * In this example, $extra_time is 172800 (2 days):
 */

if ($delay<=0)
// no delay
$coefficient = 100;

elseif ($delay<=3600)
// delay less than 1 hour
$coefficient = ceil(100-((30*$delay)/3600));

elseif ($delay<=86400)
// delay more than 1 hour and less than 1 day
$coefficient = 70;
```

Problems [+](#)

	Name	Score	Time Limit (ms)			Memory Limit (kB)	Allowed Languages (?)	Diff Command (?)	Diff Argument (?)	Upload Only (?)
			C/C++	Python	Java					
1	Problem	100	500	1500	2000	50000	C,C++,Python 2,Python 3,J	diff	-bB	<input type="checkbox"/>

Gambar 3.25: Kriteria Sukses 2.1.2 - *Add Assignment*

- Dalam halaman *Problems*(*Edit Markdown*) ada *input field* yang fokusnya tidak dapat dipindahkan menggunakan antarmuka *keyboard*. *Input field* yang dimaksud dapat dilihat pada gambar 3.26.

**Edit Problem Description (Markdown)**

Assignment 2 (Test1)  
Problem 1

Gambar 3.26: Kriteria Sukses 2.1.2 - *Problems* Bagian *Edit Markdown*

**Kriteria Sukses 2.1.3 Keyboard (No Exception)**

(Tidak Sukses)

Kriteria ini tidak sukses dipatuhi karena:

- Tombol *Collapse Menu* pada *sidebar* tidak dapat dioperasikan dengan keyboard. Tampilan halaman web dapat dilihat pada gambar 3.27.



Gambar 3.27: Kriteria Sukses 2.1.3 - *Collapse Menu*

- Tombol *Tools* dan *Assignments* pada *Menu* tidak dapat dioperasikan dengan keyboard. Tampilan halaman web dapat dilihat pada gambar 3.28.



Gambar 3.28: Kriteria Sukses 2.1.3 - *Tools*

**Kriteria Sukses 2.1.4 Character Key Shortcuts**

(Sukses)

Kriteria ini sukses dipatuhi karena pada aplikasi *SharIF Judge* tidak terdapat pintasan *keyboard* untuk konten yang ditampilkan.

**Kriteria Sukses 2.2.1 Timing Adjustable**

(Sukses)

Kriteria ini sukses dipatuhi karena pada aplikasi *SharIF Judge* batas waktu untuk mengumpulkan *Assignment* merupakan hal yang esensial dan perpanjangan batas waktu menyalahi inti dari kegiatan tersebut.

**Kriteria Sukses 2.2.2 Pause, Stop, Hide**

(Sukses)

Kriteria ini sukses dipatuhi karena pada aplikasi *SharIF Judge* satu-satunya informasi yang diperbarui secara otomatis adalah batas waktu pengumpulan *Assignment* yang merupakan bagian dari aktivitas yang esensial.

**Kriteria Sukses 2.2.3 No Timing**

(Sukses)

Kriteria ini sukses dipatuhi karena pada aplikasi *SharIF Judge* waktu pengumpulan *Assignment* bukanlah bagian esensial, pengguna dapat mendapatkan nilai yang sama selama mengumpulkan *Assignment* dalam batas waktu yang telah ditentukan.

**Kriteria Sukses 2.2.4 Interruptions**

(Sukses)

Kriteria ini sukses dipatuhi karena pada aplikasi *SharIF Judge* tidak ada interupsi.

**Kriteria Sukses 2.2.5 Re-authenticating**

(Tidak Sukses)

Kriteria ini tidak sukses dipatuhi karena pada aplikasi *SharIF Judge* ketika sesi autentikasi berakhir, data yang belum disimpan akan hilang.

**Kriteria Sukses 2.2.6 Timeouts**

(Tidak Sukses)

Kriteria ini tidak sukses dipatuhi karena pada aplikasi *SharIF Judge* waktu ketidakaktifan yang dapat menyebabkan kehilangan data kurang dari 20 jam. Pada potongan kode 3.6 ditunjukkan bahwa sesi autentikasi hanya berlangsung selama 2 jam ketika user tidak melakukan tindakan apa pun.

Listing 3.6: Kriteria Sukses 2.2.6 - Sesi Autentikasi

```
...  
$config['sess_cookie_name']      = 'shjsession';  
$config['sess_expiration']      = 7200;  
...
```



**Kriteria Sukses 2.3.1 Three Flashes or Below Threshold**

(Sukses)

Kriteria ini sukses dipatuhi karena pada aplikasi *SharIF Judge* tidak ada konten yang berkedip lebih dari detik dalam periode satu detik.

**Kriteria Sukses 2.3.2 Three Flashes**

(Sukses)

Kriteria ini sukses dipatuhi karena pada aplikasi *SharIF Judge* tidak ada konten yang berkedip lebih dari tiga detik dalam periode satu detik.

**Kriteria Sukses 2.3.3 Animation from Interactions**

(Sukses)

Kriteria ini sukses dipatuhi karena satu-satunya animasi yang terdapat di aplikasi *SharIF Judge* adalah tampilan batas waktu pengumpulan *Assignment*, animasi ini penting untuk informasi yang disampaikan.

**Kriteria Sukses 2.4.1 Bypass Blocks**

(Tidak Sukses)

Kriteria ini tidak sukses dipatuhi karena tidak ada mekanisme untuk meloncati menu pada *sidebar*.

**Kriteria Sukses 2.4.2 Page Titled**

(Sukses)

Kriteria ini sukses dipatuhi karena semua halaman pada aplikasi *SharIF Judge* memiliki judul yang menggambarkan topik atau tujuan.

**Kriteria Sukses 2.4.3 Focus Order**

(Sukses)

Kriteria ini sukses dipatuhi karena semua halaman pada aplikasi *SharIF Judge* yang memiliki urutan navigasi dapat menerima fokus dalam urutan yang menjaga makna dan pengoperasiannya.

**Kriteria Sukses 2.4.4 Link Purpose (In Context)**

(Tidak Sukses)

Kriteria ini tidak sukses dipatuhi karena:

- Pada halaman *Assignment* ada gambar yang memiliki tautan tetapi tidak ada teks yang menjelaskan tujuan dari tautan tersebut. Pada potongan kode 3.7 ditunjukkan bahwa gambar pdf memiliki tautan yang tidak memiliki teks yang menjelaskan tujuannya.

Listing 3.7: Kriteria Sukses 2.4.4 - Gambar PDF

```
...  
<td>  
    <a href="{{ site_url('assignments/pdf/~item.id) }}"></a>  
</td>  
...
```

- Pada menu *Top Bar* ada gambar yang memiliki tautan tetapi tidak ada teks yang menjelaskan tujuan dari tautan tersebut. Pada potongan kode 3.8 ditunjukkan bahwa gambar logo profil memiliki tautan yang tidak memiliki teks yang menjelaskan tujuannya.

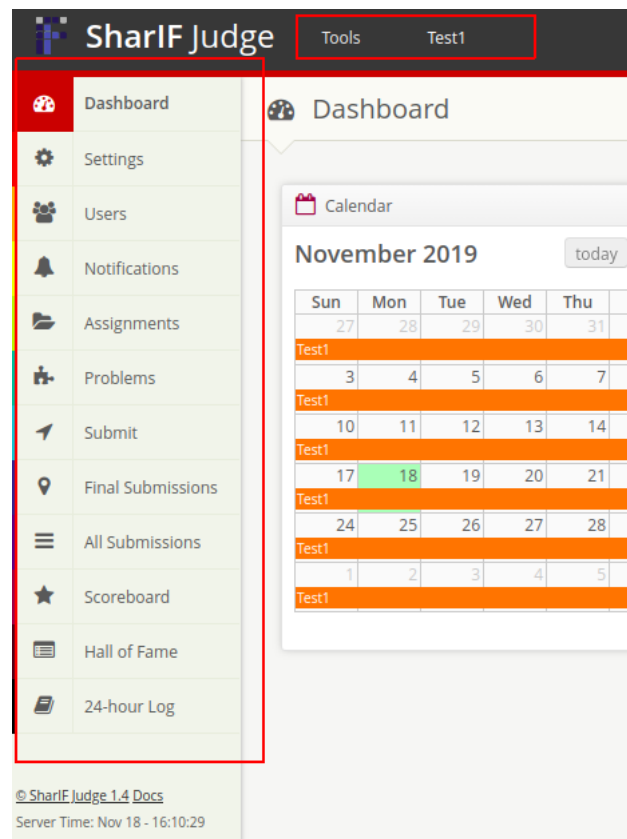
Listing 3.8: Kriteria Sukses 2.4.4 - Gambar Logo Profile

```
...  
<a href="{{ site_url('profile') }}" id="profile_link"><i class="fa  
fa-user"></i></a>  
...
```

### Kriteria Sukses 2.4.5 Multiple Ways

(Tidak Sukses)

Kriteria ini tidak sukses dipatuhi karena pada halaman aplikasi *SharIF Judge* satu-satunya cara untuk menemukan halaman web yang tersedia melalui *sidebar*. Tampilan halaman web dapat dilihat pada gambar 3.29.



Gambar 3.29: Kriteria Sukses 2.4.5 - Menemukan Halaman Web Pada Navigasi

**Kriteria Sukses 2.4.6 Headings and Labels**

(Tidak Sukses)

Kriteria ini tidak sukses dipatuhi karena:

- Pada halaman *Add User* ada *field input* yang tidak memiliki label yang menjelaskan tujuannya. Pada potongan kode 3.9 ada bidang masukan yang tidak memiliki label.

Listing 3.9: Kriteria Sukses 2.4.6 - Halaman *Add User*

```
...
<p class="input_p">
    <textarea name="new_users" id="new_users" rows="20" cols
="80" class="sharif_input">
    # Lines starting with a # sign are comments.
...

```

- Pada halaman *Add Assignment* ada *field input* yang tidak memiliki label yang menjelaskan tujuannya. Pada potongan kode 3.10 ada bidang masukan yang tidak memiliki label.

Listing 3.10: Kriteria Sukses 2.4.6 - Halaman *Add Assignment*

```
...
<td><input type="text" name="name[]" class="sharif_input short"
value="{{ problem.name }}" /></td>
...

```

- Pada halaman *Problems* bagian *Edit Markdown* dan *Edit Plain HTML* memiliki *field input* yang tidak memiliki label yang menjelaskan tujuannya. Pada potongan kode 3.11 ada bidang masukan yang tidak memiliki label.

Listing 3.11: Kriteria Sukses 2.4.6 - Halaman *Problems* bagian *Edit Markdown*

```
...
<p class="input_p">
    <textarea name="text">{{ problem.description }}</textarea>
</p>
...
```

### Kriteria Sukses 2.4.7 Focus Visible

(Tidak Sukses)

Kriteria ini tidak sukses dipatuhi karena fokus tidak tampak ketika fokus sedang berada pada menu *sidebar*.

### Kriteria Sukses 2.4.8 Location

(Sukses)

Kriteria ini sukses dipatuhi karena pada setiap halaman aplikasi *SharIF Judge* terdapat judul yang menjelaskan lokasi pengguna.

### Kriteria Sukses 2.4.9 Link Purpose (Link Only)

(Tidak Sukses)

Kriteria ini tidak sukses dipatuhi karena:

- Pada halaman *Assignment* ada gambar yang memiliki tautan tetapi tidak ada teks yang menjelaskan tujuan dari tautan tersebut. Pada potongan kode 3.12 ditunjukkan bahwa gambar pdf memiliki tautan yang tidak memiliki teks yang menjelaskan tujuannya.

Listing 3.12: Kriteria Sukses 2.4.9 - Gambar PDF

```
...
<td>
    <a href="{{ site_url('assignments/pdf/~item.id) }}"></a>
</td>
...
```

- Pada menu *Top Bar* ada gambar yang memiliki tautan tetapi tidak ada teks yang menjelaskan tujuan dari tautan tersebut. Pada potongan kode 3.13 ditunjukkan bahwa gambar logo profil memiliki tautan yang tidak memiliki teks yang menjelaskan tujuannya.

Listing 3.13: Kriteria Sukses 2.4.9 - Gambar Logo Profile

```
...
<a href="{ site_url('profile') }" id="profile_link"><i class="fa
fa-user"></i></a>
...
```

### Kriteria Sukses 2.4.10 Section Headings

(Tidak Sukses)

Kriteria ini tidak sukses dipatuhi karena pada aplikasi *SharIF Judge* judul bagian tidak dipakai untuk mengatur konten. Pada potongan kode 3.14 bagian judul bagian tidak menggunakan tag *heading*.

Listing 3.14: Kriteria Sukses 2.4.10 - Title Heading

```
...
<div id="page_title">
  <i class="fa {% block icon %}{% endblock %}"></i>
  <span dir="auto">{% block title %}{% endblock %}</span>
  {% block title_menu %}{% endblock %}
</div>
...
```

### Kriteria Sukses 2.5.1 Pointer Gestures

(Sukses)

Kriteria ini sukses dipatuhi karena pada aplikasi *SharIF Judge* tidak ada fungsionalitas yang harus dijalankan dengan menggunakan *multi-point* atau gestur berbasis *path*.

### Kriteria Sukses 2.5.2 Pointer Cancellation

(Sukses)

Kriteria ini sukses dipatuhi karena pada aplikasi *SharIF Judge* tidak ada fungsionalitas yang dapat dioperasikan dengan pointer tunggal dijalankan dengan menggunakan *down-event*.

### Kriteria Sukses 2.5.3 Label in Name

(Sukses)

Kriteria ini sukses dipatuhi karena pada aplikasi *SharIF Judge* komponen dengan label yang menyertakan teks atau gambar teks, nama tersebut berisi teks yang ditampilkan secara visual.

### Kriteria Sukses 2.5.4 Motion Actuation

(Sukses)

Kriteria ini sukses dipatuhi karena pada aplikasi *SharIF Judge* tidak terdapat fungsionalitas yang dapat dioperasikan oleh gerakan perangkat atau gerakan pengguna.

### Kriteria Sukses 2.5.5 Target Size

(Tidak Sukses)

Kriteria ini tidak sukses dipatuhi karena pada aplikasi *SharIF Judge* masih banyak target yang dapat difokus berukuran kurang dari 44 piksel *css*. Salah satu contohnya adalah bidang masukan *Upload Size Limit* yang berukuran 30 piksel *css*. Tampilan halaman web dapat dilihat pada gambar 3.30.



Gambar 3.30: Kriteria Sukses 2.5.5 - Bidang Masukan Berukuran 30 CSS Piksel

### Kriteria Sukses 2.5.6 Concurrent Input Mechanisms

(Sukses)

Kriteria ini sukses dipatuhi karena pada aplikasi *SharIF Judge* tidak membatasi modalitas masukan yang tersedia pada platform.

#### 3.1.3 Understandable

##### Kriteria Sukses 3.1.1 Language of Page

(Tidak Sukses)

Kriteria ini tidak sukses dipatuhi karena elemen *html* tidak memiliki atribut *lang*. Pada potongan kode 3.15 ditunjukkan bahwa elemen *html* tidak memiliki atribut *lang*.

Listing 3.15: Kriteria Sukses 3.1.1 - Elemen *HTML*

```
...
<html>
<head>
...
```

##### Kriteria Sukses 3.1.2 Language of Parts

(Tidak Sukses)

Kriteria ini tidak sukses dipatuhi karena elemen *html* tidak memiliki atribut *lang*. Semua halaman pada aplikasi *SharIF Judge* memakai bahasa Inggris, jika elemen *html* memiliki atribut *lang* maka kriteria ini akan terpenuhi.

#### **Kriteria Sukses 3.1.3 Unusual Words**

(Sukses)

Kriteria ini sukses dipatuhi karena pada aplikasi *SharIF Judge* tidak ada kata atau frasa tertentu yang digunakan dengan cara tidak biasa atau terbatas.

#### **Kriteria Sukses 3.1.4 Abbreviations**

(Sukses)

Kriteria ini sukses dipatuhi karena singkatan yang ada pada *SharIF Judge* lazim digunakan oleh kalangan informatika, singkatan yang dimaksud adalah IP, PDF.

#### **Kriteria Sukses 3.1.5 Reading Level**

(Sukses)

Kriteria ini sukses dipatuhi karena pada aplikasi *SharIF Judge* tidak ada teks yang membutuhkan kemampuan membaca lebih maju daripada tingkat pendidikan menengah bawah.

#### **Kriteria Sukses 3.1.6 Pronunciation**

(Sukses)

Kriteria ini sukses dipatuhi karena pada halaman aplikasi *SharIF Judge* setiap kata dapat dimengerti artinya tanpa pengguna perlu mengetahui cara mengucapkan kata tersebut.

#### **Kriteria Sukses 3.2.1 On Focus**

(Sukses)

Kriteria ini sukses dipatuhi karena pada halaman aplikasi *SharIF Judge* setiap komponen antarmuka yang menerima fokus tidak menyebabkan perubahan konteks.

#### **Kriteria Sukses 3.2.2 On Input**

(Sukses)

Kriteria ini sukses dipatuhi karena pada setiap pengguna mengubah setelan komponen antarmuka tidak menyebabkan perubahan konteks.

#### **Kriteria Sukses 3.2.3 Consistent Navigation**

(Sukses)

Kriteria ini sukses dipatuhi karena bagian navigasi menu yang muncul berulang pada tiap halaman aplikasi *SharIF Judge*, muncul dalam urutan relatif yang sama setiap kali terlihat.

### Kriteria Sukses 3.2.4 Consistent Identification

(Tidak Sukses)

Kriteria ini tidak sukses dipatuhi karena ada komponen yang memiliki fungsi yang sama tetapi tidak diidentifikasi secara konsisten. Komponen tersebut antara lain:

- Pada halaman *Assignments* istilah untuk menambah *Assignment* diberi nama *Add*. Tampilan halaman web dapat dilihat pada gambar 3.31.



Gambar 3.31: Kriteria Sukses 3.2.4 - *Assignments*

- Pada halaman *Users* istilah untuk menambah *user* diberi nama *Add User*. Tampilan halaman web dapat dilihat pada gambar 3.32.



Gambar 3.32: Kriteria Sukses 3.2.4 - *Users*

- Pada halaman. *Notifications* istilah untuk menambah *Notification* diberi nama *New*. Tampilan halaman web dapat dilihat pada gambar 3.33.



Gambar 3.33: Kriteria Sukses 3.2.4 - *Notifications***Kriteria Sukses 3.2.5 Change on Request**

(Sukses)

Kriteria ini sukses dipatuhi karena perubahan konteks hanya terjadi bila dilakukan oleh pengguna.

**Kriteria Sukses 3.3.1 Error Identification**

(Sukses)

Kriteria ini sukses dipatuhi karena kesalahan masukan terdeteksi secara otomatis, *item* yang salah diidentifikasi dan kesalahan tersebut dijelaskan kepada pengguna dalam teks.

**Kriteria Sukses 3.3.2 Labels or Instructions**

(Tidak Sukses)

Kriteria ini tidak sukses karena pada halaman problem bagian edit html, edit markdown, dan edit plain html tidak ada label atau instruksi untuk pengisiannya. Salah satu contoh tampilan halaman web dapat dilihat pada gambar 3.34.

Gambar 3.34: Kriteria Sukses 3.3.2 - *Edit Markdown* Tidak Diberi Label Atau Instruksi**Kriteria Sukses 3.3.3 Error Suggestion**

(Sukses)

Kriteria ini sukses dipatuhi karena ketika kesalahan masukan terdeteksi secara otomatis dan saran untuk mengoreksi diketahui, maka saran tersebut diberikan kepada pengguna.

### Kriteria Sukses 3.3.4 Error Prevention (Legal, Financial, Data)

(Sukses)

Kriteria ini sukses dipatuhi karena pada halaman yang mengirim tanggapan pengguna, data yang dimasukkan oleh pengguna diperiksa terkait kesalahan masukan dan pengguna dipersilakan untuk mengoreksinya.

### Kriteria Sukses 3.3.5 Help

(Tidak Sukses)

Kriteria ini tidak sukses dipatuhi karena pada aplikasi *SharIF Judge* masih ada konten yang tidak diberi label. Pengguna harus dapat mengetahui tujuan dari konten melalui labelnya.

### Kriteria Sukses 3.3.6 Error Prevention (All)

(Sukses)

Kriteria ini sukses dipatuhi karena pada halaman yang mengharuskan pengguna untuk mengirimkan informasi, data yang dimasukkan oleh pengguna diperiksa terkait kesalahan masukan dan pengguna dipersilakan untuk mengoreksinya.

## 3.1.4 Robust

### Kriteria Sukses 4.1.1 Parsing

(Tidak Sukses)

Kriteria ini tidak sukses dipatuhi karena pada aplikasi *SharIF Judge* masih banyak elemen yang tidak memiliki tag awal dan akhir yang lengkap, elemen tidak disusun berlapis sesuai dengan spesifikasinya, elemen mengandung atribut duplikat, dan ada ID yang unik. Pada potongan kode 3.16 ditunjukkan bahwa masih ada elemen yang memiliki atribut yang duplikat.

Listing 3.16: Kriteria Sukses 4.1.1 - Elemen Memiliki Atribut Duplikat

```
...
<input id="form_extra_time" type="text" name="extra_time" id="extra_time" class="
    sharif_input_medium" value="{ edit ? edit_assignment.extra_time|
    extra_time_formatter : set_value('extra_time') }" />
...
```

### Kriteria Sukses 4.1.2 Name, Role, Value

(Sukses)

Kriteria ini sukses dipatuhi karena program yang dibuat menggunakan standar *HTML* sudah memenuhi kriteria ini jika digunakan sesuai spesifikasi.

### Kriteria Sukses 4.1.3 Status Messages

(Tidak Sukses)

Kriteria ini tidak sukses dipatuhi karena pada aplikasi *SharIF Judge* tidak ada notifikasi saat melakukan aksi.

## 3.2 Peningkatan ke *Level A*

Pada bagian ini akan dijelaskan secara singkat untuk meningkatkan kepatuhan aplikasi *SharIF Judge* menjadi *level A* dengan cara memenuhi tingkat kepatuhan *level A* pada bagian 3.1.

### Kriteria Sukses 1.1.1 Non-text Content

Kriteria ini dapat dipatuhi dengan cara memberikan teks alternatif untuk konten *non-text* yang memberikan informasi yang sama seperti konten *non-text*. Konten yang dimaksud adalah:

- Logo *SharIF Judge* pada bagian menu atas.
- Gambar *PDF* pada halaman *Assignment*.

### Kriteria Sukses 1.3.1 Info dan Relationships

Tanya : Apakah isi tabel harus bisa difokus agar screen reader dapat membaca isi tabel.

Kriteria ini dapat dipatuhi dengan cara memberikan label yang sesuai pada elemen setiap form masukan. Elemen yang dimaksud adalah:

- Semua elemen bagian tabel *Problems* pada halaman *Add Assignment*.
- *Checkbox* pada halaman *Add Users*.
- *Textarea* pada halaman *Add Users*.
- *Textarea* pada halaman *Edit Problem Markdown*.
- *Textarea* pada halaman *Edit Problem Plain HTML*.
- *Dropdown* pada halaman *Problems*.
- Masukan *Upload File* pada halaman *Problems*.

### Kriteria Sukses 2.1.1 Keyboard

Kriteria ini dapat dipatuhi dengan cara memastikan semua fungsionalitas konten dapat dioperasikan dengan keyboard. Elemen yang harus diperbaiki adalah:

- Menu *Tools* pada bagian menu atas.
- *Dropdown* pada logo *Profile* pada bagian menu atas.
- Aksi *Delete User* dan *Delete Submissions* pada halaman *Users*.

### Kriteria Sukses 2.1.2 No Keyboard Trap

TODO Cari Solusinya

- Seluruh *Textarea* pada halaman *Settings*.
- *Textarea Participants* dan *Coefficient rule* pada halaman *Add Assignment*.
- *Textarea* pada halaman *Edit Problem Markdown*.

### Kriteria Sukses 2.4.1 Bypass Blocks

Kriteria ini dapat dipatuhi dengan cara menambahkan mekanisme untuk meloncati menu navigasi ketika pengguna bernavigasi menggunakan *keyboard*. Dengan cara ini pengguna dapat langsung menuju isi konten tanpa harus melewati navigasi menu.

### Kriteria Sukses 2.4.4 Link Purpose (In Context)

Kriteria ini dapat dipatuhi dengan cara menambahkan teks alternatif pada gambar yang memiliki tautan yang menjelaskan tujuan tautan tersebut. Gambar yang dimaksud adalah:

- Seluruh tautan pada *sidebar*.
- Tautan *Profile* pada bagian menu atas.
- Tautan *PDF* pada halaman *Assignment*.

### Kriteria Sukses 3.1.1 Language of Page

Kriteria ini dapat dipatuhi dengan cara memberikan atribut *lang* pada setiap halaman untuk menunjukkan bahasa yang digunakan pada halaman tersebut.

### Kriteria Sukses 3.3.2 Labels or Instructions

Kriteria ini dapat dipatuhi dengan cara memberikan label yang sesuai pada setiap bidang masukan agar pengguna dapat mengerti maksud dan tujuan bidang masukan tersebut. Bidang masukan yang dimaksud adalah:

- Semua elemen bagian tabel *Problems* pada halaman *Add Assignment*.
- *Checkbox* pada halaman *Add Users*.
- *Textarea* pada halaman *Add Users*.
- *Textarea* pada halaman *Edit Problem Markdown*.
- *Textarea* pada halaman *Edit Problem Plain HTML*.
- *Dropdown* pada halaman *Problems*.
- Masukan *Upload File* pada halaman *Problems*.

### Kriteria Sukses 4.1.1 Parsing

Kriteria ini dapat dipatuhi dengan cara memastikan setiap elemen memiliki *tag* buka dan tutup yang lengkap, elemen disusun berlapis sesuai spesifikasinya, elemen tidak memiliki atribut duplikat, dan setiap elemen memiliki ID yang unik. Elemen yang belum memenuhi kriteria ini adalah:

- *Form* masukan *Extra Time* pada halaman *Add Assignment*.

## BAB 4

### IMPLEMENTASI

Pada Bab ini akan dijelaskan bagaimana implementasi untuk menaikkan tingkat kepatuhan aplikasi *SharIF Judge*. Bagian ini akan dibagi menjadi tiga bagian yaitu implementasi ke *level A*, *AA*, *AAA*.

#### 4.1 Implementasi Ke *Level A*

TODO

##### 4.1.1 Implementasi Kriteria Sukses 1.1.1 Non-text Content

Berikut adalah perubahan yang perlu dilakukan untuk memenuhi Kriteria Sukses 1.1.1:

- Pada bagian menu atas logo *SharIF Judge* tidak memiliki teks alternatif yang menjelaskan gambar dari logo tersebut sehingga perlu ditambahkan sebuah label untuk menjelaskan gambar tersebut. Perubahan yang terjadi ada pada file `/application/views/templates/top_bar.twig`, potongan kode dapat dilihat pada 4.1.

Listing 4.1: Perubahan pada file `top_bar.twig`

```
@@ -41,7 +41,7 @@
</div>
<div id="shj_logo" class="top_left">
<a href="{{ site_url('/') }}">
- 
+ 
<h1 class="shjlogo-text">SharIF <span>Judge</span></h1>
</a>
</div>
```

- Pada halaman *Assignments* terdapat gambar *PDF* di daftar *Assignment* yang tidak memiliki teks alternatif yang menjelaskan gambar tersebut sehingga perlu ditambahkan sebuah label untuk menjelaskan gambar tersebut. Perubahan yang terjadi ada pada file `/application/views/pages/assignments.twig`, potongan kode dapat dilihat pada 4.2.

Listing 4.2: Perubahan pada file `assignments.twig`

```
@@ -69,7 +69,7 @@
{% endif %}
</td>
<td>
- <a href="{{ site_url('assignments/pdf/'~item.id) }}"></a>
```

```
+ <a href="{ site_url('assignments/pdf/~item.id) }"></a>
</td>
{% if user.level > 0 %}
<td>
```

#### 4.1.2 Implementasi Kriteria Sukses 1.3.1 Info dan Relationships

Berikut adalah perubahan yang perlu dilakukan untuk memenuhi Kriteria Sukses 1.3.1:

- Semua elemen bagian tabel *Problems* pada halaman *Add Assignment* perlu diberi label untuk memberikan informasi sesuai elemennya. Perubahan yang terjadi ada pada file */application/views/pages/admin/add\_assignment.twig*, potongan kode dapat dilihat pada 4.3.

Listing 4.3: Perubahan pada file *add\_assignment.twig*

```
@@ -201,16 +201,16 @@
{% for problem in problems %}
<tr>
<td>{{ problem.id }}</td>
- <td><input type="text" name="name[]" class="sharif_input short"
  value="{{ problem.name }}" /></td>
- <td><input type="text" name="score[]" class="sharif_input tiny2"
  value="{{ problem.score }}" /></td>
- <td><input type="text" name="c_time_limit[]" class="sharif_input
  tiny2" value="{{ problem.c_time_limit }}" /></td>
- <td><input type="text" name="python_time_limit[]" class="
  sharif_input tiny2" value="{{ problem.python_time_limit }}" /></td>
- <td><input type="text" name="java_time_limit[]" class="sharif_input
  tiny2" value="{{ problem.java_time_limit }}" /></td>
- <td><input type="text" name="memory_limit[]" class="sharif_input
  tiny" value="{{ problem.memory_limit }}" /></td>
- <td><input type="text" name="languages[]" class="sharif_input short2
  " value="{{ problem.allowed_languages }}" /></td>
- <td><input type="text" name="diff_cmd[]" class="sharif_input tiny"
  value="{{ problem.diff_cmd }}" /></td>
- <td><input type="text" name="diff_arg[]" class="sharif_input tiny"
  value="{{ problem.diff_arg }}" /></td>
- <td><input type="checkbox" name="is_upload_only[]" class="check"
  value="{{ problem.id }}" {{ problem.is_upload_only ? 'checked' }} /></td>
+ <td><input type="text" name="name[]" class="sharif_input short"
  value="{{ problem.name }}" aria-label="Problem Name" /></td>
+ <td><input type="text" name="score[]" class="sharif_input tiny2"
  value="{{ problem.score }}" aria-label="Score" /></td>
+ <td><input type="text" name="c_time_limit[]" class="sharif_input
  tiny2" value="{{ problem.c_time_limit }}" aria-label="Time Limit for C
  " /></td>
+ <td><input type="text" name="python_time_limit[]" class="
  sharif_input tiny2" value="{{ problem.python_time_limit }}" aria-label="
  Time Limit for Python" /></td>
```

```

+         <td><input type="text" name="java_time_limit[]" class="sharif_input
tiny2" value="{{ problem.java_time_limit }}" aria-label="Time Limit for
Java"/></td>
+         <td><input type="text" name="memory_limit[]" class="sharif_input
tiny" value="{{ problem.memory_limit }}" aria-label="Memory Limit"/></td>
+         <td><input type="text" name="languages[]" class="sharif_input short2
" value="{{ problem.allowed_languages }}" aria-label="Allowed Languages
"/></td>
+         <td><input type="text" name="diff_cmd[]" class="sharif_input tiny"
value="{{ problem.diff_cmd }}" aria-label="Diff Command"/></td>
+         <td><input type="text" name="diff_arg[]" class="sharif_input tiny"
value="{{ problem.diff_arg }}" aria-label="Diff Argument"/></td>
+         <td><input type="checkbox" name="is_upload_only[]" class="check"
value="{{ problem.id }}" {{ problem.is_upload_only ? 'checked' }} aria-
label="Upload Only"/></td>
<td><i class="fa fa-times-circle fa-lg color1 delete_problem pointer"></i></
td>
</tr>
{% endfor %}

```

- *Checkbox* dan *Textarea* pada halaman *Add Users* perlu diberi label untuk memberikan informasi elemen tersebut. Perubahan yang terjadi ada pada file `/application/views/pages/admin/add_user.twig`, potongan kode dapat dilihat pada 4.4.

Listing 4.4: Perubahan pada file `add_user.twig`

```

@@ -56,10 +56,10 @@
<li>If you want to send passwords by email, do not add too many users at one
time. This may result in mail delivery fail.</li>
</ul>
<p class="input_p">
-         <input type="checkbox" name="send_mail" id="send_mail" /> Send
usernames and passwords by email (Waits <input type="text" name="delay"
id="delay" class="sharif_input tiny" value="2"/> second(s) before sending
each email, so please be patient).
+         <input type="checkbox" name="send_mail" id="send_mail" aria-label="
Send mail"/> Send usernames and passwords by email (Waits <input type="
text" name="delay" id="delay" class="sharif_input tiny" value="2" aria-
label="Delay"/> second(s) before sending each email, so please be patient
).
</p>
<p class="input_p">
- <textarea name="new_users" id="new_users" rows="20" cols="80" class="
sharif_input">
+ <textarea name="new_users" id="new_users" rows="20" cols="80" class="
sharif_input" aria-label="Command for creating new user">
# Lines starting with a # sign are comments.
# Each line (except comments) represents a user.
# The syntax of each line is:

```

- *Textarea* pada halaman *Edit Problem Markdown* perlu diberi label untuk memberikan informasi elemen tersebut. Perubahan yang terjadi ada pada file `/application/views/pages/admin/edit_problem_md.twig`, potongan kode dapat dilihat pada 4.5.

Listing 4.5: Perubahan pada *file edit\_problem\_md.twig*

```

@@ -177,7 +177,7 @@ Violets are blue.
</p>
{{ form_open("problems/edit/md/#{description_assignment.id}/#{problem.id}")
}}
<p class="input_p">
-     <textarea dir="auto" name="text" rows="30" cols="75" class="
      sharif_input" id="md_editor">{{ problem.description }}</textarea>
+     <textarea dir="auto" name="text" rows="30" cols="75" class="
      sharif_input" id="md_editor" aria-label="Markdown Editor">{{ problem.
      description }}</textarea>
</p>
<p class="input_p">
<input type="submit" value="Save" class="sharif_input"/>

```

- *Textarea* pada halaman *Edit Problem Plain HTML* perlu diberi label untuk memberikan informasi elemen tersebut. Perubahan yang terjadi ada pada *file /application/views/pages/admin/edit\_problem\_plain.twig*, potongan kode dapat dilihat pada 4.6.

Listing 4.6: Perubahan pada *file edit\_problem\_plain.twig*

```

@@ -33,7 +33,7 @@
</p>
{{ form_open("problems/edit/html/#{description_assignment.id}/#{problem.id}")
}}
<p class="input_p">
-     <textarea name="text" rows="30" cols="80" class="sharif_input" id="
      html_editor">{{ problem.description }}</textarea>
+     <textarea name="text" rows="30" cols="80" class="sharif_input" id="
      html_editor" aria-label="HTML Editor">{{ problem.description }}</textarea
      >
</p>
<p class="input_p">
<input type="submit" value="Save" class="sharif_input"/>

```

- *Dropdown* pada halaman *Problems* perlu diberi label untuk memberikan informasi elemen tersebut. Perubahan yang terjadi ada pada *file /application/views/pages/problems.twig*, potongan kode dapat dilihat pada 4.7.

Listing 4.7: Perubahan pada *file problems.twig* untuk *Dropdown Select Language*

```

@@ -96,7 +96,7 @@
$(document).ready(function(){
<input type="hidden" name="problem" value="{{ problem.id }}" />

<p class="input_p">
-     <select id="languages" name="language" class="sharif_input full-
      width">
+     <select id="languages" name="language" class="sharif_input full-
      width" aria-label="Select Language">
<option value="0" selected="selected">-- Select Language --</option>
{% for l in problem.allowed_languages %}
<option value="{{ l }}">{{ l }}</option>

```



- Masukan *Upload File* pada halaman *Problems* perlu diberi label untuk memberikan informasi elemen tersebut. Perubahan yang terjadi ada pada file `/application/views/pages/problems.twig`, potongan kode dapat dilihat pada 4.8.

Listing 4.8: Perubahan pada file *problems.twig* untuk masukan *Upload file*

```

@@ -104,7 +104,7 @@
$(document).ready(function(){
</select>
</p>
<p class="input_p">
-     <input type="file" id="file" class="sharif_input full-width" name="
+     <input type="file" id="file" class="sharif_input full-width" name="
+     userfile" aria-label="Upload File"/>
</p>
<p class="input_p">
<input type="submit" value="Submit" class="sharif_input"/>

```

#### 4.1.3 Implementasi Kriteria Sukses 2.1.1 Keyboard

TODO

#### 4.1.4 Implementasi Kriteria Sukses 2.1.2 No Keyboard Trap

TODO

#### 4.1.5 Implementasi Kriteria Sukses 2.4.1 Bypass Blocks

TODO

#### 4.1.6 Implementasi Kriteria Sukses 2.4.4 Link Purpose (In Context)

Berikut adalah perubahan yang perlu dilakukan untuk memenuhi Kriteria Sukses 2.4.4:

- Seluruh link pada *sidebar* perlu diberikan label yang menjelaskan link tersebut. Perubahan yang terjadi ada pada file `/application/views/templates/side_bar.twig`, potongan kode dapat dilihat pada 4.9.

Listing 4.9: Perubahan pada file *side\_bar.twig*

```

@@ -6,78 +6,78 @@
<div id="side_bar" class="sidebar_open">
<ul>
<li class="color-dashboard{{ selected=='dashboard' ? ' selected' }}">
-     <a href="{{ site_url('dashboard') }}">
+     <a href="{{ site_url('dashboard') }}" aria-labelledby="dashboard-
+     label">
<i class="fa fa-dashboard fa-lg"></i>
-     <span class="sidebar_text">Dashboard</span>
+     <span class="sidebar_text" id="dashboard-label">Dashboard</
+     span>
</a>
</li>
{% if user.level == 3 %}

```

```

<li class="color-settings{{ selected=='settings' ? ' selected' }}">
-     <a href="{{ site_url('settings') }}">
+     <a href="{{ site_url('settings') }}" aria-labelledby="settings-label
    ">
<i class="fa fa-gear fa-lg"></i>
-         <span class="sidebar_text">Settings</span>
+         <span class="sidebar_text" id="settings-label">Settings</
    span>
</a>
</li>
<li class="color-users{{ selected=='users' ? ' selected' }}">
-     <a href="{{ site_url('users') }}">
+     <a href="{{ site_url('users') }}" aria-labelledby="users-label">
<i class="fa fa-users fa-lg"></i>
-         <span class="sidebar_text">Users</span>
+         <span class="sidebar_text" id="users-label">Users</span>
</a>
</li>
{% endif %}
<li class="color-notifications{{ selected=='notifications' ? ' selected' }}">
-     <a href="{{ site_url('notifications') }}">
+     <a href="{{ site_url('notifications') }}" aria-labelledby="
    notifications-label">
<i class="fa fa-bell fa-lg"></i>
-         <span class="sidebar_text">Notifications</span>
+         <span class="sidebar_text" id="notifications-label">
    Notifications</span>
</a>
</li>
<li class="color-assignments{{ selected=='assignments' ? ' selected' }}">
-     <a href="{{ site_url('assignments') }}">
+     <a href="{{ site_url('assignments') }}" aria-labelledby="assignments-
    label">
<i class="fa fa-folder-open fa-lg"></i>
-         <span class="sidebar_text">Assignments</span>
+         <span class="sidebar_text" id="assignments-label">
    Assignments</span>
</a>
</li>
<li class="color-problems{{ selected=='problems' ? ' selected' }}">
-     <a href="{{ site_url('problems') }}">
+     <a href="{{ site_url('problems') }}" aria-labelledby="problems-label
    ">
<i class="fa fa-puzzle-piece fa-lg"></i>
-         <span class="sidebar_text">Problems</span>
+         <span class="sidebar_text" id="problems-label">Problems</
    span>
</a>
</li>
<li class="color-submit{{ selected=='submit' ? ' selected' }}">

```

```

-         <a href="{ { site_url('submit') } }">
+         <a href="{ { site_url('submit') } }" aria-labelledby="submit-label">
<i class="fa fa-location-arrow fa-lg"></i>
-             <span class="sidebar_text">Submit</span>
+             <span class="sidebar_text" id="submit-label">Submit</span>
</a>
</li>
<li class="color-final_submissions{ { selected=='final_submissions' ? '
selected' } }">
-         <a href="{ { site_url('submissions/final') } }">
+         <a href="{ { site_url('submissions/final') } }" aria-labelledby="final-
submission-label">
<i class="fa fa-map-marker fa-lg"></i>
-             <span class="sidebar_text">Final Submissions</span>
+             <span class="sidebar_text" id="final-submission-label">Final
Submissions</span>
</a>
</li>
<li class="color-all_submissions{ { selected=='all_submissions' ? ' selected'
} }">
-         <a href="{ { site_url('submissions/all') } }">
+         <a href="{ { site_url('submissions/all') } }" aria-labelledby="all-
submission-label">
<i class="fa fa-bars fa-lg"></i>
-             <span class="sidebar_text">All Submissions</span>
+             <span class="sidebar_text" id="all-submission-label">All
Submissions</span>
</a>
</li>
<li class="color-scoreboard{ { selected=='scoreboard' ? ' selected' } }">
-         <a href="{ { site_url('scoreboard') } }">
+         <a href="{ { site_url('scoreboard') } }" aria-labelledby="scoreboard-
label">
<i class="fa fa-star fa-lg"></i>
-             <span class="sidebar_text">Scoreboard</span>
+             <span class="sidebar_text" id="scoreboard-label">Scoreboard</
span>
</a>
</li>
<li class="color-halloffame{ { selected=='halloffame' ? ' selected' } }">
-         <a href="{ { site_url('halloffame') } }">
+         <a href="{ { site_url('halloffame') } }" aria-labelledby="hall-of-fame-
label">
<i class="fa fa-list-alt fa-lg"></i>
-             <span class="sidebar_text">Hall of Fame</span>
+             <span class="sidebar_text" id="hall-of-fame-label">Hall of
Fame</span>
</a>
</li>
{% if user.level == 3 %}
<li class="color-logs{ { selected=='logs' ? ' selected' } }">

```

```

-      <a href="{{ site_url('logs') }}">
+      <a href="{{ site_url('logs') }}" aria-labelledby="24-hour-log-label">

<i class="fa fa-book fa-lg"></i>
-      <span class="sidebar_text">24-hour Log</span>
+      <span class="sidebar_text" id="24-hour-log-label">24-hour
      Log</span>
</a>
</li>
{% endif %}

```

- Tautan pada gambar *Profile* yang ada di menu atas perlu diberi label untuk menjelaskan tujuan dari tautan tersebut. Perubahan yang terjadi ada pada `/application/views/templates/top_bar.twig`, potongan kode dapat dilihat pada 4.10.

Listing 4.10: Perubahan pada file `top_bar.twig`

```

@@ -5,7 +5,7 @@
#}
<div id="top_bar" class="color-{{ selected }}">
<div class="top_object shj_menu" id="user_top">
-      <a href="{{ site_url('profile') }}" id="profile_link"><i class="fa
      fa-user"></i></a>
+      <a href="{{ site_url('profile') }}" id="profile_link" aria-label="
      Profile"><i class="fa fa-user"></i></a>
<div class="top_menu user-menu">
<div class="gravatar"></div>
<div class="name"><i class="fa fa-user"></i> {{ user.username }}</div>

```

- Tautan pada gambar *PDF* perlu diberi label untuk menjelaskan tujuannya. Perubahan untuk tautan pada gambar *PDF* yang ada di halaman *Assignments* sudah dilakukan pada 4.2

#### 4.1.7 Implementasi Kriteria Sukses 3.1.1 Language of Page

Seluruh halaman aplikasi *SharIF Judge* menggunakan Bahasa Inggris sehingga perlu diberikan atribut *lang* yang bernilai *"en"* pada awal *tag html* untuk menunjukkan bahwa bahasa yang dipakai di halaman tersebut adalah Bahasa Inggris. Setiap kali halaman dibuka, aplikasi *SharIF Judge* akan menjalankan file *base.twig* sebagai dasarnya sehingga perubahan hanya perlu dilakukan pada file tersebut. Perubahan dapat dilihat pada `/application/views/templates/base.twig`, potongan kode dapat dilihat pada 4.11.

Listing 4.11: Perubahan pada file `base.twig`

```

@@ -4,7 +4,7 @@
# author: Mohammad Javad Naderi <mjnaderi@gmail.com>
#}
<!DOCTYPE html>
- <html>
+ <html lang="en">
<head>
<title>{% block head_title %}{% endblock %} - SharIF Judge</title>
<meta content="text/html" charset="UTF-8">

```

#### 4.1.8 Implementasi Kriteria Sukses 3.3.2 Labels or Instructions

Berikut adalah perubahan yang perlu dilakukan untuk memenuhi Kriteria Sukses 3.3.2:

- Semua elemen bagian tabel *Problems* pada halaman *Add Assignment* perlu diberi label untuk menjelaskan tujuan dari elemen tersebut. Perubahan sudah dilakukan pada 4.3.
- *Checkbox* dan *Textarea* pada halaman *Add Users* perlu diberi label yang menjelaskan tujuan dari elemen tersebut. Perubahan sudah dilakukan pada 4.4.
- *Textarea* pada halaman *Edit Problem Markdown* perlu diberi label yang menjelaskan tujuan dari elemen tersebut. Perubahan sudah dilakukan pada 4.5.
- *Textarea* pada halaman *Edit Problem Plain HTML* perlu diberi label yang menjelaskan tujuan dari elemen tersebut. Perubahan sudah dilakukan pada 4.6.
- *Dropdown* pada halaman *Problems* perlu diberi label yang menjelaskan tujuan dari elemen tersebut. Perubahan sudah dilakukan pada 4.7.
- Masukan *Upload File* pada halaman *Problems* perlu diberi label yang menjelaskan tujuan dari elemen tersebut. Perubahan sudah dilakukan pada 4.8.

#### 4.1.9 Implementasi Kriteria Sukses 4.1.1 Parsing

*Form* masukan *Extra Time* pada halaman *Add Assignment* memiliki atribut *id* yang duplikat. Atribut yang dibuang yaitu atribut *id="extra\_time"* karena sudah dipakai pada elemen *extra time* pada menu bagian atas. Perubahan dapat dilihat pada file */application/views/pages/admin/add\_assignment.twig*, potongan kode dapat dilihat pada 4.12.

Listing 4.12: Perubahan pada file *add\_assignment.twig*

```
@@ -119,7 +119,7 @@
Extra Time (minutes)<br>
<span class="form_comment">Extra time for late submissions.</span>
</label>
-     <input id="form_extra_time" type="text" name="extra_time" id="extra_time"
-       class="sharif_input medium" value="{ { edit ? edit_assignment.extra_time |
-       extra_time_formatter : set_value('extra_time') } }" />
+     <input id="form_extra_time" type="text" name="extra_time" class="
+       sharif_input medium" value="{ { edit ? edit_assignment.extra_time |
+       extra_time_formatter : set_value('extra_time') } }" />
{ { form_error('extra_time', '<div class="shj_error">', '</div>') } }
</p>
<p class="input_p clear">
```



# 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
5 #include<stdio.h>
6
7 void myFunction( int input, float* output ) {
8     switch ( array[i] ) {
9         case 1: // This is silly code
10             if ( a >= 0 || b <= 3 && c != x )
11                 *output += 0.005 + 20050;
12             char = 'g';
13             b = 2^n + ~right_size - leftSize * MAX_SIZE;
14             c = (--aaa + &daa) / (bbb++ - ccc % 2);
15             strcpy(a,"hello_$@?");
16         }
17         count = ~mask | 0x00FF00AA;
18     }
19
20 // Fonts for Displaying Program Code in LATEX
21 // Adrian P. Robson, nepswb.co.uk
22 // 8 October 2012
23 // http://nepswb.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
11    trajectory
12    protected ArrayList<Integer> closeID; //store the ID of all vertices
13    protected ArrayList<Double> closeDist; //store the distance of all vertices
14    protected int totaltrj; //total trajectories in the set
15
16    /*
17     * Constructor
18     * @param id : id of the set
19     * @param totaltrj : total number of trajectories in the set
20     * @param FurthestEdge : the furthest edge
21     */
22    public MyFurSet(int id,int totaltrj,MyEdge FurthestEdge) {
23        this.id = id;
24        this.totaltrj = totaltrj;
25        this.FurthestEdge = FurthestEdge;
26        set = new HashSet<MyVertex>();
27        ordered = new ArrayList<ArrayList<Integer>>();
28        for (int i=0;i<totaltrj;i++) ordered.add(new ArrayList<Integer>());
29        closeID = new ArrayList<Integer>(totaltrj);
30        closeDist = new ArrayList<Double>(totaltrj);
31        for (int i = 0; i < totaltrj; i++) {
32            closeID.add(-1);
33            closeDist.add(Double.MAX_VALUE);
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