

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
2020

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
2020**

LEMBAR PENGESAHAN

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

Amabel Levint

NPM: 2016730013

Bandung, 12 Juli 2020

Menyetujui,

Pembimbing

Pascal Alfadian, Nugroho, M.Comp.

Ketua Tim Penguji

Anggota Tim Penguji

Raymond Chandra Putra, M.T.

Dr. Veronica Sri Moertini

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 12 Juli 2020

Amabel Levint
NPM: 2016730013

ABSTRAK

SharIF Judge (dengan huruf kapital "IF") adalah sebuah aplikasi gratis dan *open source* untuk menilai *code* berbahasa *C*, *C++*, *Java* dan *Python*. *SharIF Judge* adalah pencabangan(*fork*) dari *Sharif Judge* (dengan huruf kecil "if") yang telah dibuat oleh Mohammed Javad Naderi pada tahun 2014. Versi dari pencabangan ini memuat fitur baru yang diperlukan oleh jurusan teknik informatika UNPAR. Aplikasi ini dapat digunakan oleh mahasiswa dan dosen. Banyaknya pengguna yang mengakses aplikasi *SharIF Judge* menjadikan kemudahan menggunakan aplikasi ini menjadi hal yang penting.

Web Content Accessibility Guidelines (WCAG) 2.1 telah membuat pedoman untuk mengukur kemudahan sebuah situs dapat diakses. Hasil pengukurannya disebut tingkat kepatuhan. Aplikasi *SharIF Judge* telah diukur tingkat kepatuhannya terhadap pedoman WCAG 2.1. Beberapa bagian dalam aplikasi *SharIF Judge* yang belum memenuhi syarat akan diperbaiki untuk meningkatkan tingkat kepatuhan *SharIF Judge* sampai tingkat *A*.

Pengujian aplikasi *SharIF Judge* dilakukan dengan menggunakan fitur yang ada dalam kondisi seakan-akan penguji tidak dapat melihat. Pengujian ini menggunakan teknologi alat bantu pembaca layar. Berdasarkan pengujian yang telah dilakukan, penguji berhasil menggunakan fitur yang ada pada aplikasi *SharIF Judge* dengan mengandalkan teknologi alat bantu, pendengaran, dan *keyboard* sebagai alat bantu navigasi.

Kata-kata kunci: *SharIF Judge*, aksesibilitas, aplikasi *web*, WCAG 2.1

ABSTRACT

SharIF Judge is a free and open source online judge for C, C++, Java and Python programming course. SharIF Judge is a branch(fork) of the original Sharif Judge created by Mohammed Javad Naderi in 2014. This version contains many improvements due to the needs of the Computer Science department of UNPAR. This application can be used by students and lecturers. The many amount of users that use SharIF Judge makes accessibility for this application important.

Web Content Accessibility Guideline(WCAG) 2.1 has made guidelines to measure accessibility for websites. The measurement result is called conformance level. SharIF Judge's accessibility will be measured using the WCAG 2.1 guidelines. Parts of the SharIF Judge application that have not met requirements determined by WCAG 2.1 will be fixed.

SharIF Judge will be tested by using features in which the tester is conditioned as if they cannot see. This test is done using a screen reader technology. Based on the results, testers are able to use SharIF Judge by utilizing the screen reader, hearing, and keyboard as navigation tools.

Keywords: SharIF Judge, accessibility, web application, WCAG 2.1

*Dipersembahkan untuk Tuhan YME, keluarga tercinta, Bapak
Pascal Alfadian sebagai dosen pembimbing, teman-teman yang
telah membantu dan memberi semangat dalam penyusunan skripsi
ini, dan diri sendiri*

KATA PENGANTAR

Puji syukur penulis panjatkan kepada Tuhan Yang Maha Esa karena atas karunia-Nya, penulis dapat menyelesaikan penyusunan skripsi yang berjudul "Kepatuhan dan Rekomendasi Perbaikan *Web Content Accessibility Guideline* untuk Aplikasi *SharIF Judge*". Penulis menyadari bahwa penyusunan skripsi ini tidak terlepas dari bantuan berbagai pihak. Pada kesempatan kali ini penulis ingin mengucapkan terima kasih kepada:

1. Keluarga yang sudah memberikan dukungan kepada penulis.
2. Bapak Pascal Alfadian, M.Comp. selaku dosen pembimbing yang telah membimbing penulis dan memberikan dukungan serta bantuan kepada penulis dalam proses penyusunan skripsi ini.
3. Bapak Raymond Chandra Putra, M.T. selaku dosen penguji yang telah memberikan kritik dan saran sehingga penelitian ini menjadi lebih baik.
4. Teman-teman seperjuangan kuliah dan skripsi, khususnya Hizkia Steven dan Cahyadi Hartanto Sulaiman yang sudah memberikan bantuan, kritik, dan saran untuk penyusunan skripsi ini.
5. Pihak-pihak lain yang terlalu banyak untuk disebutkan satu per satu yang telah membantu skripsi ini.

Penulis menyadari bahwa penelitian ini masih jauh dari kata sempurna. Oleh karena itu, penulis memohon maaf jika terjadi kesalahan. Semoga penelitian ini dapat memberikan informasi yang bermanfaat dan inspirasi untuk penelitian berikutnya.

Bandung, Juli 2020

Penulis

DAFTAR ISI

KATA PENGANTAR	xv
DAFTAR ISI	xvii
DAFTAR GAMBAR	xix
DAFTAR TABEL	xxi
1 PENDAHULUAN	1
1.1 Latar Belakang	1
1.2 Rumusan Masalah	2
1.3 Tujuan	2
1.4 Batasan Masalah	2
1.5 Metodologi	2
1.6 Sistematika Pembahasan	2
2 LANDASAN TEORI	5
2.1 WCAG 2.1	5
2.1.1 <i>Perceivable</i>	6
2.1.2 <i>Operable</i>	12
2.1.3 <i>Understandable</i>	16
2.1.4 <i>Robust</i>	18
2.2 <i>SharIF Judge</i>	19
2.2.1 Instalasi	19
2.2.2 <i>Clean URLs</i>	20
2.2.3 Users	20
2.2.4 Menambah <i>Assignment</i>	21
2.2.5 <i>Sample Assignment</i>	24
2.2.6 <i>Test Structure</i>	26
2.2.7 Deteksi Kecurangan	29
2.2.8 Keamanan	30
2.2.9 <i>Sandbox</i>	31
2.2.10 <i>Shield</i>	32
3 ANALISIS	35
3.1 Metodologi Pengecekan	35
3.2 Tingkat Kepatuhan <i>SharIF Judge</i>	35
3.2.1 <i>Perceivable</i>	38
3.2.2 <i>Operable</i>	52
3.2.3 <i>Understandable</i>	60
3.2.4 <i>Robust</i>	64
3.3 Peningkatan ke <i>Level A</i>	65
3.3.1 Kriteria Sukses 1.1.1 <i>Non-text Content</i>	65

3.3.2	Kriteria Sukses 1.3.1 <i>Info and Relationships</i>	65
3.3.3	Kriteria Sukses 2.1.1 <i>Keyboard</i>	65
3.3.4	Kriteria Sukses 2.1.2 <i>No Keyboard Trap</i>	66
3.3.5	Kriteria Sukses 2.4.1 <i>Bypass Blocks</i>	66
3.3.6	Kriteria Sukses 2.4.4 <i>Link Purpose (In Context)</i>	66
3.3.7	Kriteria Sukses 3.1.1 <i>Language of Page</i>	66
3.3.8	Kriteria Sukses 3.3.2 <i>Labels or Instructions</i>	66
3.3.9	Kriteria Sukses 4.1.1 <i>Parsing</i>	66
4	IMPLEMENTASI DAN PENGUJIAN	67
4.1	Implementasi	67
4.1.1	Implementasi Kriteria Sukses 1.1.1 <i>Non-text Content</i>	67
4.1.2	Implementasi Kriteria Sukses 1.3.1 <i>Info and Relationships</i>	67
4.1.3	Implementasi Kriteria Sukses 2.1.1 <i>Keyboard</i>	68
4.1.4	Implementasi Kriteria Sukses 2.1.2 <i>No Keyboard Trap</i>	69
4.1.5	Implementasi Kriteria Sukses 2.4.1 <i>Bypass Blocks</i>	69
4.1.6	Implementasi Kriteria Sukses 2.4.4 <i>Link Purpose (In Context)</i>	69
4.1.7	Implementasi Kriteria Sukses 3.1.1 <i>Language of Page</i>	69
4.1.8	Implementasi Kriteria Sukses 3.3.2 <i>Labels or Instructions</i>	70
4.1.9	Implementasi Kriteria Sukses 4.1.1 <i>Parsing</i>	70
4.2	Pengujian	70
4.2.1	Skenario dan Hasil Pengujian	70
4.2.2	Kesimpulan	82
5	KESIMPULAN DAN SARAN	83
5.1	Kesimpulan	83
5.2	Saran	83
	DAFTAR REFERENSI	85
	A KODE PROGRAM	87

DAFTAR GAMBAR

2.1	Pembagian Kriteria Sukses	6
2.2	Tampilan Halaman <i>Assignments</i>	22
3.1	Pembagian Kriteria Sukses	35
3.2	Pelanggaran Kriteria Sukses 1.3.1 - Elemen yang Tidak Diberi Label Pada Halaman <i>Add Users</i>	40
3.3	Pelanggaran Kriteria Sukses 1.3.1 - Elemen yang Tidak Diberi Label Pada Halaman <i>Problems</i> Bagian <i>Edit Markdown</i>	41
3.4	Pelanggaran Kriteria Sukses 1.3.1 - Elemen yang Tidak Diberi Label Pada Halaman <i>Problems</i> Bagian <i>Edit Plain Html</i>	41
3.5	Pelanggaran Kriteria Sukses 1.4.3 - Kontras Elemen Navigasi Atas	43
3.6	Pelanggaran Kriteria Sukses 1.4.3 - Kontras Elemen <i>Dashboard</i>	43
3.7	Pelanggaran Kriteria Sukses 1.4.3 - Kontras Elemen <i>Settings</i>	44
3.8	Pelanggaran Kriteria Sukses 1.4.3 - Kontras Elemen <i>Notifications</i>	44
3.9	Pelanggaran Kriteria Sukses 1.4.3 - Kontras Elemen <i>Assignments</i>	45
3.10	Pelanggaran Kriteria Sukses 1.4.3 - Kontras Elemen <i>Add Assignments</i>	45
3.11	Pelanggaran Kriteria Sukses 1.4.4 - <i>Sidenav</i>	46
3.12	Pelanggaran Kriteria Sukses 1.4.6 - Kontras Elemen Navigasi Atas	47
3.13	Pelanggaran Kriteria Sukses 1.4.6 - Kontras Elemen <i>Dashboard</i>	47
3.14	Pelanggaran Kriteria Sukses 1.4.6 - Kontras Elemen <i>Settings</i>	48
3.15	Pelanggaran Kriteria Sukses 1.4.6 - Kontras Elemen <i>Notifications</i>	48
3.16	Pelanggaran Kriteria Sukses 1.4.6 - Kontras Elemen <i>Assignments</i>	49
3.17	Pelanggaran Kriteria Sukses 1.4.6 - Kontras Elemen <i>Add Assignments</i>	49
3.18	Pelanggaran Kriteria Sukses 1.4.8 - Lebar <i>Notifications</i>	50
3.19	Pelanggaran Kriteria Sukses 1.4.10 - <i>Horizontal Scroll</i>	50
3.20	Pelanggaran Kriteria Sukses 1.4.11 - Ikon Pada Halaman <i>Assignments</i>	50
3.21	Ikon <i>Download Test and Description</i>	51
3.22	Pelanggaran Kriteria Sukses 1.4.12 - <i>Text Spacing</i> Halaman <i>Add Assignment</i>	52
3.23	Pelanggaran Kriteria Sukses 2.1.1 - Tombol <i>Tools</i>	53
3.24	Pelanggaran Kriteria Sukses 2.1.2 - <i>Keyboard Trap</i> Pada Halaman <i>Settings</i>	53
3.25	Pelanggaran Kriteria Sukses 2.1.2 - <i>Keyboard Trap</i> Pada Halaman <i>Add User</i>	53
3.26	Pelanggaran Kriteria Sukses 2.1.2 - <i>Keyboard Trap</i> Pada Halaman <i>Add Assignment</i>	54
3.27	Pelanggaran Kriteria Sukses 2.1.2 - <i>Keyboard Trap</i> Pada Halaman <i>Problems</i> Bagian <i>Edit Markdown</i>	54
3.28	Pelanggaran Kriteria Sukses 2.1.1 - Tombol <i>Tools</i>	54
3.29	Pelanggaran Kriteria Sukses 2.4.5 - Menemukan Halaman Web Pada Navigasi	57
3.30	Pelanggaran Kriteria Sukses 2.5.5 - Bidang Masukan Berukuran 30 <i>CSS Pixel</i>	60
3.31	Pelanggaran Kriteria Sukses 3.2.4 - <i>Assignments</i>	62
3.32	Pelanggaran Kriteria Sukses 3.2.4 - <i>Users</i>	63
3.33	Pelanggaran Kriteria Sukses 3.2.4 - <i>Notifications</i>	63
3.34	Pelanggaran Kriteria Sukses 3.3.2 - <i>Edit Markdown</i> Tidak Diberi Label Atau Instruksi	64

DAFTAR TABEL

2.1	<i>User Roles Table</i>	20
2.2	<i>Permission Table</i>	21
2.3	<i>Problem 1 (Penjumlahan)</i>	25
2.4	<i>Problem 2 (Max)</i>	25
3.1	Kepatuhan <i>SharIF Judge</i> terhadap prinsip <i>Understandable</i>	36
3.2	Kepatuhan <i>SharIF Judge</i> terhadap prinsip <i>Robust</i>	36
3.3	Kepatuhan <i>SharIF Judge</i> terhadap prinsip <i>Perceivable</i>	37
3.4	Kepatuhan <i>SharIF Judge</i> terhadap prinsip <i>Operable</i>	37
4.1	Hasil pengujian <i>Login</i>	71
4.2	Hasil pengujian <i>Settings</i>	71
4.3	Hasil pengujian <i>Add Users</i>	72
4.4	Hasil pengujian <i>Delete User</i>	73
4.5	Hasil pengujian <i>Delete User Submissions</i>	73
4.6	Hasil pengujian <i>Edit User</i>	74
4.7	Hasil pengujian <i>Add Notification</i>	74
4.8	Hasil pengujian <i>Add Assignment</i>	75
4.9	Hasil pengujian <i>Delete Assignment</i>	76
4.10	Hasil pengujian <i>Edit Assignment</i>	77
4.11	Hasil pengujian <i>Edit Problem Description (Markdown)</i>	78
4.12	Hasil pengujian <i>Edit Problem Description (HTML)</i>	79
4.13	Hasil pengujian <i>Edit Problem Description (Plain HTML)</i>	80
4.14	Hasil pengujian <i>Submit</i>	81
4.15	Hasil pengujian <i>Final Submission</i>	81

BAB 1

PENDAHULUAN

Bab ini membahas tentang latar belakang, rumusan masalah, tujuan, batasan masalah, metodologi, serta sistematika pembahasan pada skripsi ini.

1.1 Latar Belakang

Pada zaman perkembangan teknologi saat ini, banyak sekali orang yang mengakses halaman web. Beberapa diantara mereka memiliki disabilitas. Masih banyak web yang masih belum dapat diakses oleh kalangan disabilitas. Menurut Brian Sierkowski [1], aksesibilitas web penting karena beberapa hal. Pertama, semua orang memiliki hak untuk mengakses internet sehingga akses web harus dapat diakses oleh berbagai kalangan. Kedua, dengan membuat sebuah web mudah diakses maka pengguna web tersebut akan bertambah. Ketiga, munculnya *browser* yang bukan tradisional seperti pada perangkat *smartphone* dan komputer tablet yang jumlahnya tiga kali lebih banyak daripada *personal computer* (PC) dalam beberapa tahun mendatang. Keempat, aksesibilitas berawal dari teknologi dan strategi desain yang membuat konten mudah dikelola dan dimigrasi daripada desain lama yang mencampurkan konten dengan pemformatannya.

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 [2] yang memuat rekomendasi untuk membuat konten web lebih mudah diakses. Konten web yang telah mengikuti pedoman-pedoman ini akan 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.

SharIF Judge [3] (dengan huruf kapital "IF") adalah sebuah aplikasi gratis dan *open source* untuk menilai *code* berbahasa C, C++, Java dan Python. *SharIF Judge* adalah pencabangan(*fork*) dari *Sharif Judge* [4] (dengan huruf kecil "if") 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*)[5] dan bagian *backend* dibuat dengan BASH.

Pada skripsi ini, telah 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 kondisi keterbatasan visual. Dengan perbaikan ini diharapkan aplikasi *SharIF Judge* dapat diakses oleh banyak kalangan.

1.2 Rumusan Masalah

Rumusan masalah pada skripsi ini adalah sebagai berikut:

- Bagaimana tingkat kepatuhan *SharIF Judge* terhadap *WCAG 2.1* ?
- Rekomendasi perbaikan apa saja yang perlu dilakukan terhadap *SharIF Judge* untuk meningkatkan tingkat kepatuhannya ?
- Bagaimana pengaruh peningkatan tingkat kepatuhan *WCAG 2.1* terhadap penggunaan aplikasi *SharIF Judge* oleh pengguna dengan keterbatasan penglihatan ?

1.3 Tujuan

Tujuan pada skripsi ini adalah sebagai berikut:

- Mengukur tingkat kepatuhan *SharIF Judge* terhadap *WCAG 2.1*.
- Membuat rekomendasi perbaikan yang perlu dilakukan terhadap *SharIF Judge* untuk meningkatkan tingkat kepatuhannya.
- Mengetahui pengaruh peningkatan tingkat kepatuhan *WCAG 2.1* terhadap penggunaan aplikasi *SharIF Judge* oleh pengguna dengan keterbatasan penglihatan.

1.4 Batasan Masalah

Batasan masalah pada skripsi ini adalah sebagai berikut:

1. Pengujian hanya dilakukan pada perangkat komputer.
2. Pengujian hanya dilakukan dalam kondisi keterbatasan visual.
3. Peningkatan tingkat kepatuhan hanya dilakukan sampai tingkat *A*.
4. Untuk beberapa kriteria sukses tidak dapat diukur karena keterbatasan alat, waktu, dan sumber daya (akan dibahas pada subbab 3.2.1)

1.5 Metodologi

Metodologi yang dilakukan pada skripsi ini adalah sebagai berikut:

1. Studi literatur mengenai *SharIF Judge* dan bagaimana cara mematuhi kriteria-kriteria *WCAG 2.1*.
2. Mengukur tingkat kepatuhan *SharIF Judge* terhadap kriteria-kriteria *WCAG 2.1*.
3. Memberikan rekomendasi perbaikan terhadap setiap kriteria sukses *WCAG 2.1* yang belum terpenuhi pada aplikasi *SharIF Judge*.
4. Mengimplementasikan rekomendasi perbaikan pada aplikasi *SharIF Judge*.
5. Menguji hasil perbaikan dengan cara mengoperasikan fitur-fitur aplikasi *SharIF Judge* dalam kondisi tidak dapat melihat.
6. Menulis dokumen skripsi.

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* dan perubahan yang dapat dilakukan untuk meningkatkan kepatuhan situs web *SharIF Judge* terhadap *WCAG 2.1*. Beberapa pengecekan kepatuhan kriteria sukses akan dibantu dengan *tools*.
 4. Bab 4: Implementasi dan Pengujian, akan membahas hasil implementasi dan pengujian yang telah dilakukan pada situs web *SharIF Judge*. Pengujian hanya dilakukan dalam kondisi keterbatasan visual.
 5. Bab 5: Kesimpulan dan saran, akan berisi kesimpulan dari hasil penelitian yang telah dilakukan dan saran yang dapat diberikan untuk penelitian berikutnya.

BAB 2

LANDASAN TEORI

Bab ini membahas tentang landasan teori yang digunakan dalam skripsi ini.

2.1 WCAG 2.1

Web Content Accessibility Guidelines (WCAG) 2.1 memuat rekomendasi untuk membuat konten web lebih mudah diakses. Konten web yang telah mengikuti pedoman-pedoman ini akan 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 yang penting bagi orang disabilitas diluar 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.

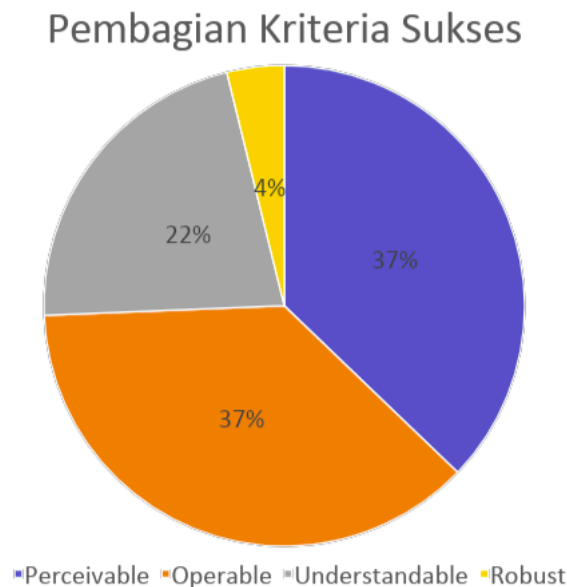
Kriteria sukses memiliki tiga tingkat kesesuaian yaitu tingkat *A* (terkecil), *AA*, *AAA* (terbesar). Ada beberapa faktor yang menentukan tingkat tersebut. Faktor tersebut termasuk:

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.

Ada 78 kriteria sukses yang dibagi menjadi 3 tingkatan yaitu tingkat *A*, *AA*, dan *AAA*. Pembagian kriteria sukses berdasarkan prinsip *accessibility* dapat dilihat pada gambar 2.1.



Gambar 2.1: Pembagian Kriteria Sukses

Berikut adalah uraian kriteria sukses *WCAG 2.1*:

2.1.1 *Perceivable*

Pada subbab ini dibahas kriteria sukses yang termasuk ke dalam kategori *perceivable*. Kategori ini mensyaratkan agar informasi dan antarmuka pengguna harus ditampilkan kepada pengguna dengan cara yang dapat dipahami. Terdapat dua puluh sembilan poin kriteria sukses.

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 merupakan kontrol atau masukan user maka konten tersebut harus memiliki nama yang menjelaskan tujuannya.
- Media berbasis waktu: jika konten bukan teks merupakan media berbasis waktu, maka setidaknya disediakan alternatif berupa teks untuk identifikasi deskriptif dari konten bukan teks.
- Tes: jika konten bukan teks merupakan 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, *invisible*: 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.

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.

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.

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.

Kriteria Sukses 1.2.4 *Captions (Live)* (Level AA)

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

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

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

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

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

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.

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.

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.

Kriteria Sukses 1.3.1 *Info and Relationships* (Level A)

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

Kriteria Sukses 1.3.2 *Meaningful Sequence* (Level A)

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

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.

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.

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.

Kriteria Sukses 1.3.6 *Identify Purpose* (Level AAA)

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

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.

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.

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.

Untuk menghitung rasio kontras, *relative luminance* warna yang lebih muda(L1) akan dibandingkan dengan *relative luminance* warna yang lebih tua(L2). Dari hasil perbandingan tersebut hasilnya berkisar 1:1(tidak ada kontras) sampai 21:1(kontras paling tinggi). Berikut adalah rumus untuk perbandingan tersebut:

$$\text{Contrast Ratio} = (L1 + 0.05) : (L2 + 0.05) \quad (2.1)$$

Untuk menghitung *relative luminance* sebuah warna *sRGB*(*standard Red Green Blue*) digunakan rumus sebagai berikut:

$$L = 0.2126 \times R + 0.7152 \times G + 0.0722 \times B \quad (2.2)$$

Variabel R , G , dan B didefinisikan sebagai berikut:

$$R = \begin{cases} R_{sRGB} \div 12.92 & \text{if } R_{sRGB} \leq 0.03928, \\ ((R_{sRGB} + 0.055) \div 1.055)^{2.4} & \text{otherwise} \end{cases} \quad (2.3)$$

$$G = \begin{cases} G_{sRGB} \div 12.92 & \text{if } G_{sRGB} \leq 0.03928, \\ ((G_{sRGB} + 0.055) \div 1.055)^{2.4} & \text{otherwise} \end{cases} \quad (2.4)$$

$$B = \begin{cases} B_{sRGB} \div 12.92 & \text{if } B_{sRGB} \leq 0.03928, \\ ((B_{sRGB} + 0.055) \div 1.055)^{2.4} & \text{otherwise} \end{cases} \quad (2.5)$$

Variabel R_{sRGB} , G_{sRGB} , dan B_{sRGB} didefinisikan sebagai berikut:

$$R_{sRGB} = R_{8bit} \div 255 \quad (2.6)$$

$$G_{sRGB} = G_{8bit} \div 255 \quad (2.7)$$

$$B_{sRGB} = B_{8bit} \div 255 \quad (2.8)$$

Sebagai contoh perhitungan, berikut adalah perhitungan kontras untuk teks berwarna hitam di atas latar belakang berwarna putih. Teks hitam memiliki spesifikasi warna sebagai berikut:

$$R_{8bit} = 0$$

$$G_{8bit} = 0$$

$$B_{8bit} = 0$$

Sedangkan warna latar belakang putih memiliki spesifikasi warna sebagai berikut:

$$R_{8bit} = 255$$

$$G_{8bit} = 255$$

$$B_{8bit} = 255$$

Berikut adalah contoh perhitungan *relative luminance* untuk teks berwarna hitam:

$$R_{sRGB} = 0 \div 255 = 0$$

$$G_{sRGB} = 0 \div 255 = 0$$

$$B_{sRGB} = 0 \div 255 = 0$$

$$R = 0 \div 12.92 = 0$$

$$G = 0 \div 12.92 = 0$$

$$B = 0 \div 12.92 = 0$$

$$L = 0.2126 \times 0 + 0.7152 \times 0 + 0.0722 \times 0 = 0$$

Berikut adalah contoh perhitungan *relative luminance* untuk latar belakang berwarna putih:

$$R_{sRGB} = 255 \div 255 = 1$$

$$G_{sRGB} = 255 \div 255 = 1$$

$$B_{sRGB} = 255 \div 255 = 1$$

$$R = ((1 + 0.055) \div 1.055)^{2.4} = 1$$

$$G = ((1 + 0.055) \div 1.055)^{2.4} = 1$$

$$B = ((1 + 0.055) \div 1.055)^{2.4} = 1$$

$$L = 0.2126 \times 1 + 0.7152 \times 1 + 0.0722 \times 1 = 1$$

Hasil perbandingan kontras teks hitam di atas latar belakang putih sebagai berikut:

$$\begin{aligned} \text{Contrast Ratio} &= (1 + 0.05) : (0 + 0.05) \\ &= 21 : 1 \end{aligned}$$

Kriteria ini dapat diperiksa dengan menggunakan alat bantu *axe - Web Accessibility Testing*¹. Alat bantu ini tersedia pada *Chrome Web Store* sebagai ekstensi untuk *Google Chrome*. Untuk menggunakan alat bantu ini pengguna harus menavigasi seluruh halaman *website* dan menjalankan aplikasi *axe - Web Accessibility Testing*. Setelah selesai dijalankan, aplikasi *axe - Web Accessibility Testing* akan memberikan hasil pengujian *accessibility* salah satunya kontras.

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.

¹<https://www.deque.com/axe/>

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:

- *Customizable*: gambar teks dapat disesuaikan dengan kebutuhan pengguna.
- Esensial: tampilan teks esensial untuk menyampaikan informasi.

Kriteria Sukses 1.4.6 *Contrast (Enhanced)* (Level AAA)

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

- Teks besar: teks berukuran besar dan gambar teks berukuran besar memiliki rasio kontras setidaknya 4.5:1.
- 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. Rumus perhitungan kontras dapat dilihat pada subbab [2.1.1](#).

Kriteria Sukses 1.4.7 *Low or 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.

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*.

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.

Kriteria Sukses 1.4.10 *Reflow* (Level AA)

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

- *Scroll vertical* untuk konten yang lebarnya setara dengan 320 *pixel css*.
- *Scroll horizontal* untuk konten yang tingginya setara dengan 256 *pixel css*.

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

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.

Rumus perhitungan kontras dapat dilihat pada subbab 2.1.1.

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.

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.

2.1.2 *Operable*

Pada subbab ini dibahas kriteria sukses yang termasuk ke dalam kategori *operable*. Kategori ini mensyaratkan agar komponen antarmuka pengguna dan navigasi harus dapat dioperasikan. Terdapat dua puluh sembilan poin kriteria sukses.

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.

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.

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.

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.

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.

Kriteria Sukses 2.2.2 *Pause, Stop, Hide* (Level A)

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

- Bergerak, berkedip, *scrolling*: 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

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.

Kriteria Sukses 2.2.4 *Interruptions* (Level AAA)

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

Kriteria Sukses 2.2.5 *Re-authenticating* (Level AAA)

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

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.

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

Kriteria Sukses 2.3.2 *Three Flashes* (Level AAA)

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

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.

Kriteria Sukses 2.4.1 *Bypass Blocks* (Level A)

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

Kriteria Sukses 2.4.2 *Page Titled* (Level A)

Halaman web memiliki judul yang menggambarkan topik atau tujuan.

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.

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.

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.

Kriteria Sukses 2.4.6 *Headings and Labels* (Level AA)

Judul dan label menjelaskan topik atau tujuan.

Kriteria Sukses 2.4.7 *Focus Visible* (Level AA)

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

Kriteria Sukses 2.4.8 *Location* (Level AAA)

Informasi mengenai lokasi pengguna dalam satu set halaman web tersedia.

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.

Kriteria Sukses 2.4.10 *Section Headings* (Level AAA)

Judul bagian digunakan untuk mengatur konten.

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 *gesture* berbasis jalur, kecuali jika *multipoint* atau *gesture* berbasis jalur sangat penting.

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.

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.

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.

Kriteria Sukses 2.5.5 *Target Size* (Level AAA)

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

- Setara: Target tersedia melalui tautan atau kontrol yang setara pada halaman yang sama berukuran setidaknya 44 *css pixel*;
- *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.

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.

2.1.3 *Understandable*

Pada subbab ini dibahas kriteria sukses yang termasuk ke dalam kategori *understandable*. Kategori ini mensyaratkan agar informasi dan pengoperasian antarmuka pengguna harus dapat dimengerti. Terdapat tujuh belas poin kriteria sukses.

Kriteria Sukses 3.1.1 *Language of Page* (Level A)

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

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.

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 *idioms* dan *jargon*.

Kriteria Sukses 3.1.4 *Abbreviations* (Level AAA)

Tersedianya mekanisme untuk mengidentifikasi kepanjangan dari singkatan.

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.

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.

Kriteria Sukses 3.2.1 *On Focus* (Level A)

Tidak ada perubahan konteks jika komponen antarmuka pengguna menerima fokus.

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.

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.

Kriteria Sukses 3.2.4 *Consistent Identification* (Level AA)

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

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.

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.

Kriteria Sukses 3.3.2 *Labels or Instructions* (Level A)

Label atau instruksi diberikan saat konten membutuhkan masukan pengguna.

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.

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.

Kriteria Sukses 3.3.5 *Help* (Level AAA)

Tersedianya bantuan pada konteks yang sedang berjalan.

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.

2.1.4 *Robust*

Pada subbab ini dibahas kriteria sukses yang termasuk ke dalam kategori *robust*. Kategori ini mensyaratkan agar konten harus cukup handal sehingga dapat ditafsirkan oleh berbagai agen pengguna, termasuk teknologi alat bantu. Terdapat tiga poin kriteria sukses.

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.

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.

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 [3] (dengan huruf kapital "IF") adalah sebuah aplikasi gratis dan *open source* untuk menilai code berbahasa *C*, *C++*, *Java* dan *Python*. *SharIF Judge* adalah pencabangan(*fork*) dari *Sharif Judge* [4] (dengan huruf kecil "if") 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*) [5] dan bagian *backend* dibuat dengan *BASH*.

2.2.1 Instalasi

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

Persyaratan

Untuk menjalankan aplikasi *SharIF Judge*, diperlukan satu *Linux server* dengan syarat berikut:

- *Webserver* dapat 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:

```
1 echo shell_exec("php -v");
2
```

- *Tools* yang digunakan untuk dapat 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 terbaru 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*.

```
1 $system_path = '/home/mohammad/secret/system';
2 $application_folder = '/home/mohammad/secret/application';
3
```

- 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.

```

1  /* Enter database connection settings here: */
2  'dbdriver' => 'postgre',    // database driver (mysqli, postgre)
3  'hostname' => 'localhost', // database host
4  'username' => ' ',         // database username
5  'password' => ' ',         // database password
6  'database' => ' ',         // database name
7  'dbprefix' => 'shj_',      // table prefix
8  /*****
9

```

- 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 mengikuti 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

Umumnya *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 menghapus *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 oleh *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 *textarea*. Baris yang diawali dengan *#* adalah komentar. Setiap baris lainnya mewakili pengguna dengan sintaksis berikut:

```

1 USERNAME,EMAIL,DISPLAY-NAME,PASSWORD,ROLE
2
3 * Usernames may contain lowercase letters or numbers and must be between 3 and 20
  characters in length.
4 * Passwords must be between 6 and 30 characters in length.
5 * You can use RANDOM[n] for password to generate random n-digit password.
6 * ROLE must be one of these: 'admin', 'head_instructor', 'instructor', 'student'
```

Berikut adalah contohnya:

```

1 # This is a comment!
2 # This is another comment!
3 instructor instructor@sharifjudge.ir 123456 head_instructor
4 instructor2 instructor2@sharifjudge.ir random[7] instructor
5 student1 st1@sharifjudge.ir random[6] student
6 student2 st2@sharifjudge.ir random[6] student
7 student3 st3@sharifjudge.ir random[6] student
8 student4 st4@sharifjudge.ir random[6] student
9 student5 st5@sharifjudge.ir random[6] student
10 student6 st6@sharifjudge.ir random[6] student
11 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 2.2 menunjukkan contoh tampilan halaman untuk menambah *assignment*.

Sharif Judge Tools Sample 7 5 15 36

Dashboard Settings Users Notifications **Assignments** Problems Submit Final Submissions All Submissions Scoreboard

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

Problems: [Use this structure](#)

Name	Score	Time Limit (ms)			Memory Limit (KB)	Allowed Languages	Diff	Diff Command	Diff Argument	Upload
		C/C++	Python	Java						
1 Problem	100	500	1500	2000	50000	C,C++Python 2,Python 3,js	diff	-b8		<input type="button" value="Upload"/>

© Sharif Judge Ltd Coza
Server Time: Mar 19 - 17:44:23

Gambar 2.2: Tampilan Halaman *Assignments*

Berikut ini adalah pengaturan yang ada 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 ditunjukkan 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 ditunjukkan jika pengguna mengaktifkan pengaturan *Java Exceptions*:

```

1      Test 1
2      ACCEPT
3      Test 2
4      Runtime Error (java.lang.ArrayIndexOutOfBoundsException)
5      Test 3
6      Runtime Error (java.lang.ArrayIndexOutOfBoundsException)
7      Test 4
8      ACCEPT
9      Test 5
10     ACCEPT
11     Test 6
12     ACCEPT
13     Test 7
14     ACCEPT
15     Test 8
16     Runtime Error (java.lang.ArrayIndexOutOfBoundsException)
17     Test 9
18     Runtime Error (java.lang.StackOverflowError)
19     Test 10
20     Runtime Error (java.lang.ArrayIndexOutOfBoundsException)
21

```

- *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):

```

1      if ($delay<=0)
2      // no delay
3      $coefficient = 100;
4
5      elseif ($delay<=3600)
6      // delay less than 1 hour
7      $coefficient = ceil(100-((30*$delay)/3600));
8
9      elseif ($delay<=86400)
10     // delay more than 1 hour and less than 1 day

```

```

11     $coefficient = 70;
12
13     elseif (($delay-86400)<=3600)
14         // delay less than 1 hour in second day
15         $coefficient = ceil(70-((20*($delay-86400))/3600));
16
17     elseif (($delay-86400)<=86400)
18         // delay more than 1 hour in second day
19         $coefficient = 50;
20
21     elseif ($delay > $extra_time)
22         // too late
23         $coefficient = 0;
24

```

- *Time Limit*

Pengguna dapat menentukan batas waktu untuk menjalankan kode dalam satuan *millisecond*. 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 menentukan batas memori dalam satuan *kilobytes*, namun penggunaan *Memory Limit* tidak terlalu akurat.

- *Allowed Languages*

Pengguna dapat menentukan 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 pengujian 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:

```

1 .
2 |-- p1
3 |   |-- in
4 |   |   |-- input1.txt
5 |   |   |-- input2.txt
6 |   |   |-- input3.txt
7 |   |   |-- input4.txt
8 |   |   |-- input5.txt
9 |   |   |-- input6.txt
10 |   |   |-- input7.txt
11 |   |   |-- input8.txt
12 |   |   |-- input9.txt
13 |   |   --- input10.txt
14 |   |-- out
15 |   |   --- output1.txt
16 |   |-- tester.cpp
17 |   --- desc.md
18 |-- p2
19 |   |-- in
20 |   |   |-- input1.txt
21 |   |   |-- input2.txt
22 |   |   |-- input3.txt
23 |   |   |-- input4.txt
24 |   |   |-- input5.txt
25 |   |   |-- input6.txt
26 |   |   |-- input7.txt
27 |   |   |-- input8.txt
28 |   |   |-- input9.txt
29 |   |   --- input10.txt
30 |   |-- out

```

```

31 | | | -- output1.txt
32 | | | -- output2.txt
33 | | | -- output3.txt
34 | | | -- output4.txt
35 | | | -- output5.txt
36 | | | -- output6.txt
37 | | | -- output7.txt
38 | | | -- output8.txt
39 | | | -- output9.txt
40 | | --- output10.txt
41 | | -- desc.md
42 | --- Problem2.pdf
43 | -- p3
44 | --- desc.md
45 --- 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

Ketika menambah *assignments*, pengguna harus menyediakan *file zip* yang berisi *test cases*. *File zip* ini berisi folder-folder untuk setiap *problem*. Pengguna harus memberikan 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*:

Kode 2.1: Contoh kode *tester.cpp*

```

1 /*
2 * tester.cpp
3 */
4
5 #include <iostream>
6 #include <fstream>

```



```

7 | #include <string>
8 | using namespace std;
9 | int main(int argc, char const *argv[])
10 | {
11 |
12 |     ifstream test_in(argv[1]);    /* This stream reads from test's input
13 |     file */
14 |     ifstream test_out(argv[2]);   /* This stream reads from test's output
15 |     file */
16 |     ifstream user_out(argv[3]);   /* This stream reads from user's output
17 |     file */
18 |
19 |     /* Your code here */
20 |     /* If user's output is correct, return 0, otherwise return 1 */
21 |     ...
22 | }

```

Contoh *File*

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

```

1 | .
2 | |-- p1
3 | |   |-- in
4 | | |   |-- input1.txt
5 | | |   |-- input2.txt
6 | | |   |-- input3.txt
7 | | |   |-- input4.txt
8 | | |   |-- input5.txt
9 | | |   |-- input6.txt
10 | | |  |-- input7.txt
11 | | |  |-- input8.txt
12 | | |  |-- input9.txt
13 | | |  --- input10.txt
14 | |   |-- out
15 | | |   --- output1.txt
16 | |   --- tester.cpp
17 | --- p2
18 | |-- in
19 | |   |-- input1.txt
20 | |   |-- input2.txt
21 | |   |-- input3.txt
22 | |   |-- input4.txt
23 | |   |-- input5.txt
24 | |   |-- input6.txt
25 | |   |-- input7.txt
26 | |   |-- input8.txt
27 | |   |-- input9.txt

```

```

28 | --- input10.txt
29 |--- out
30 |-- output1.txt
31 |-- output2.txt
32 |-- output3.txt
33 |-- output4.txt
34 |-- output5.txt
35 |-- output6.txt
36 |-- output7.txt
37 |-- output8.txt
38 |-- output9.txt
39 |--- output10.txt

```

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

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

```

1  /*
2  * tester.cpp
3  */
4
5  #include <iostream>
6  #include <fstream>
7  #include <string>
8  using namespace std;
9  int main(int argc, char const *argv[])
10 {
11
12     ifstream test_in(argv[1]);    /* This stream reads from test's input file
13     */
14     ifstream test_out(argv[2]);    /* This stream reads from test's output file
15     */
16     ifstream user_out(argv[3]);    /* This stream reads from user's output file
17     */
18
19     /* Your code here */
20     /* If user's output is correct, return 0, otherwise return 1      */
21     /* e.g.: Here the problem is: read n numbers and print their sum: */
22
23     int sum, user_output;
24     user_out >> user_output;
25
26     if ( test_out.good() ) // if test's output file exists
27     {
28         test_out >> sum;
29     }
30     else
31     {
32         int n, a;
33         sum=0;
34         test_in >> n;
35         for (int i=0 ; i<n ; i++){

```

```

33         test_in >> a;
34         sum += a;
35     }
36 }
37
38 if (sum == user_output)
39     return 0;
40 else
41     return 1;
42
43 }

```

Problem 2 menggunakan metode "*Input/Output Comparison*" untuk memeriksa 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*:

Kode 2.3: Potongan *script perl*

```

1  ...
2
3  $server = 'moss.stanford.edu';
4  $port = '7690';
5  $nreq = "Request not sent.";
6  $usage = "usage: moss [-x] [-l language] [-d]
7  [-b basefile1] ... [-b basefilen] [-m #]
8  [-c \"string\"] file1 file2 file3 ...";
9
10 #
11 # The userid is used to authenticate your queries to the server;
12 # don't change it!
13 #
14 $userid=YOUR_MOSS_USER_ID;
15
16 #
17 # Process the command line options. This is done in a non-standard
18 # way to allow multiple -b's.
19 #
20 $opt_l = "c"; # default language is c
21 $opt_m = 10;

```

```

22     $opt_d = 0;
23
24     ...
25 }
26
27

```

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 dapat memakai *Sandbox* untuk *C/C++* dan mengaktifkan *Java Security Manager (Java Policy)* untuk *Java*. Penjelasan lebih lanjut tentang *Sandbox* ada pada subbab 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 subbab 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*:

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

Pada file *tester/runcode.sh* ganti

```

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

```

menjadi

```

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

```

```
4 sudo -u restricted_user $CMD <$IN >out 2>err
5 fi
```

dan *uncomment* baris berikut:

```
1 sudo -u restricted_user pkill -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*:

```
1 restricted_user soft nproc 3
2 restricted_user hard nproc 5
3
```

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*² yang *user* kumpulkan sehingga memerlukan *tools* 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*.

Umumnya *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:

```
1 $ cd tester/easysandbox
2 $ chmod +x runalltests.sh
3 $ chmod +x runtest.sh
4 $ 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*.

²kode yang dapat dieksekusi

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 ringan. 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:

```
1 #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:

```
1 /*
2
3 @file defc.h
4 There should be a newline at end of this file.
5 Put the message displayed to user after // in each line
6
7 e.g. If you want to disallow goto, add this line:
8 #define goto errorNo13    //Goto is not allowed
9
10 */
11
12 #define system errorNo1    //"system" is not allowed
13 #define freopen errorNo2   //File operation is not allowed
14 #define fopen errorNo3     //File operation is not allowed
15 #define fprintf errorNo4   //File operation is not allowed
16 #define fscanf errorNo5    //File operation is not allowed
17 #define feof errorNo6      //File operation is not allowed
18 #define fclose errorNo7    //File operation is not allowed
```

```

19 #define ifstream errorNo8      //File operation is not allowed
20 #define ofstream errorNo9      //File operation is not allowed
21 #define fork errorNo10         //Fork is not allowed
22 #define clone errorNo11        //Clone is not allowed
23 #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*:

```

1 # @file shield_py3.py
2
3 import sys
4 sys.modules['os']=None
5
6 BLACKLIST = [
7     '__import__', # deny importing modules
8     'eval', # eval is evil
9     'open',
10    'file',
11    'exec',
12    'execfile',
13    'compile',
14    'reload',
15    'input'
16 ]
17 for func in BLACKLIST:
18     if func in __builtins__.__dict__:
19         del __builtins__.__dict__[func]

```


BAB 3

ANALISIS

Bab ini membahas tentang analisis tingkat kepatuhan *SharIF Judge* terhadap *WCAG 2.1*.

3.1 Metodologi Pengecekan

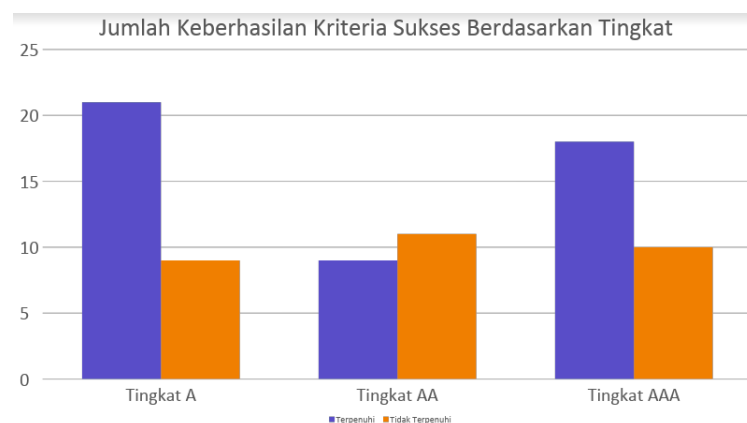
Proses pengecekan kepatuhan setiap kriteria sukses pada aplikasi *SharIF Judge* dilakukan dengan tahap-tahap sebagai berikut:

1. Menggunakan *tools axe - Web Accessibility Testing* dan *Accessibility Insight for Web* pada seluruh halaman web untuk mendeteksi kepatuhan kriteria sukses.
2. Jika kepatuhan kriteria sukses tidak terdeteksi dengan *tools*, maka dilakukan pengecekan manual pada *source code*.

Tools axe - Web Accessibility Testing dan *Accessibility Insight for Web* dapat memeriksa kepatuhan kriteria sukses 2.1.1, 2.1.1, 2.1.1, 2.1.2, 2.1.3, 2.1.3, dan 2.1.4.

3.2 Tingkat Kepatuhan *SharIF Judge*

Aplikasi *SharIF Judge* sudah mematuhi 21 dari 30 kriteria sukses tingkat A, 9 dari 20 kriteria sukses tingkat AA, dan 18 dari 28 kriteria sukses tingkat AAA. Jumlah keberhasilan kriteria sukses berdasarkan tingkat dapat dilihat pada gambar 3.1.



Gambar 3.1: Pembagian Kriteria Sukses Berdasarkan Tingkat

Untuk membantu pemeriksaan kepatuhan *SharIF Judge* terhadap *WCAG 2.1* digunakan alat bantu seperti *axe - Web Accessibility Testing*¹ dan *Accessibility Insight for Web*², tetapi alat bantu tersebut tidak dapat memeriksa seluruh kriteria sukses *WCAG 2.1*. Kesuksesan aplikasi *SharIF Judge* dalam mematuhi kriteria sukses *WCAG 2.1* ditulis dalam tabel 3.1, 3.2, 3.3, dan 3.4.

¹<https://www.deque.com/axe/>

²<https://accessibilityinsights.io/>

Penjelasan untuk analisis setiap kriteria sukses dapat dilihat pada subbab 3.2.1, 3.2.2, 3.2.3, dan 3.2.4.

Tabel 3.1: 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 3.2: 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		-

Tabel 3.3: 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 3.4: 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
Dilanjutkan pada halaman berikutnya		

Tabel 3.4 – lanjutan dari halaman sebelumnya

Kriteria Sukses	Tingkat Kepatuhan	Hasil
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		-

3.2.1 *Perceivable*

Berikut adalah kepatuhan *SharIF Judge* berdasarkan prinsip *perceivable*.

Text Alternatives

Pada subbab ini dibahas kriteria sukses yang termasuk kedalam kategori *text alternatives*. Kategori ini mensyaratkan agar konten yang bukan teks memiliki teks alternatif.

Kriteria Sukses 1.1.1 *Non-text Content* (Tidak Sukses)

Kriteria ini tidak sukses dipatuhi karena:

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

Kode 3.1: Pelanggaran Kriteria Sukses 1.1.1 - Logo *SharIF Judge* Tidak Diberi Alternatif Teks

```

1  ...
2  <a href="{{ site_url('/') }}">
3      
4      <h1 class="shjlogo-text">SharIF <span>Judge</span></h1>
5  </a>
6  ...
7  
```

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

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

```

1  ...
2  <td>
3  <a href="{{ site_url('assignments/pdf/~item.id') }}"></a>

```

4	</td>
5	...
6	

Time-based Media

Pada subbab ini dibahas kriteria sukses yang termasuk kedalam kategori *time-based media*. Kategori ini mensyaratkan agar konten berbasis waktu memiliki versi alternatifnya.

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.

Adaptable

Pada subbab ini dibahas kriteria sukses yang termasuk kedalam kategori *adaptable*. Kategori ini mensyaratkan agar konten dapat ditampilkan dengan berbagai cara (contohnya tata letak yang lebih sederhana) tanpa kehilangan informasi atau struktur konten tersebut.

Kriteria Sukses 1.3.1 *Info and 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 3.2.

Secure | sharif-judge/users/add

ge Tools Test1

+ Add Users Help

You can use this field to add multiple users at the same time.

- 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.
- If you want to send passwords by email, do not add too many users at one time. This may result in mail delivery fail.

☐ Send usernames and passwords by email (Wait 2 second(s) before sending each email, so please be patient).

Lines starting with a # sign are comments.
 # Each line (except comments) represents a user.
 # The syntax of each line is:
 #
 # USERNAME, EMAIL, DISPLAY-NAME, PASSWORD, ROLE
 #
 # Roles: admin head instructor instructor student
 # You can use RANDOM[n] for password to generate random n-digit password.

Add Users

Gambar 3.2: Pelanggaran 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. Kesalahan dapat dilihat pada potongan kode 3.3.

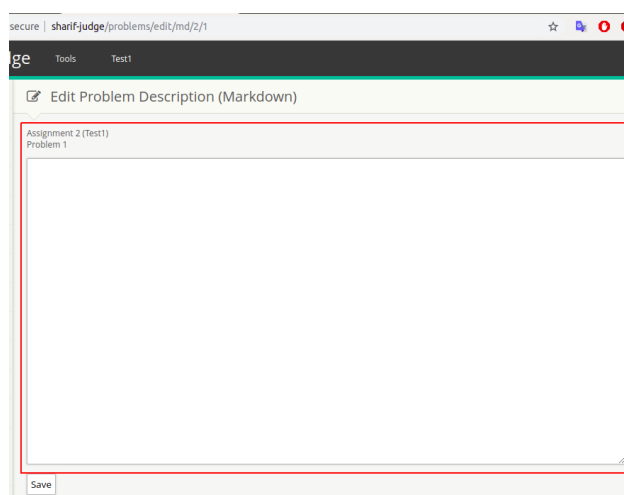
Kode 3.3: Pelanggaran Kriteria Sukses 1.3.1 - Elemen Dalam Form yang Tidak Diberi Label

```

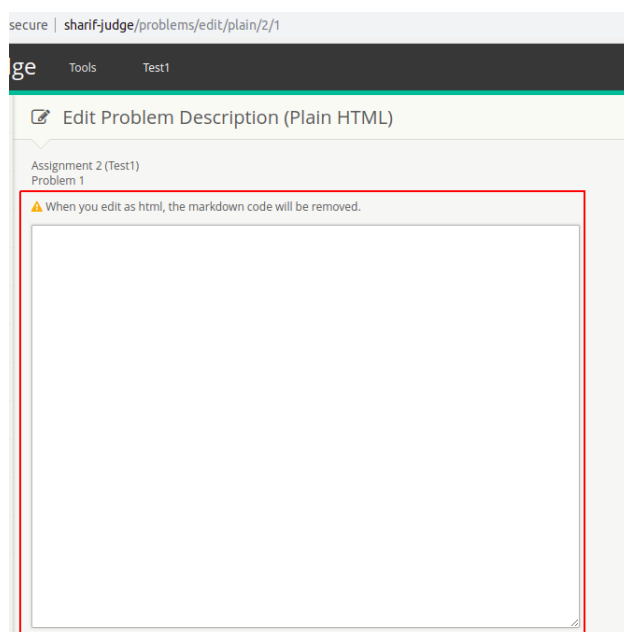
1    ...
2    <td><input type="text" name="name[]" class="sharif_input short" value="
    Problem "></td>\
3    ...
4

```

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



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



Gambar 3.4: Pelanggaran 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.

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

```
1 ...
2 <p class="input_p">
3   <textarea name="text">{{ problem.description }}</textarea>
4 </p>
5 ...
```

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.

Kode 3.5: Pelanggaran Kriteria Sukses 1.3.6 - *Sidebar* Tidak Memakai *Tag Nav*

```
1 ...
2 <div id="side_bar" class="sidebar_open">
3   <ul>
4     <li class="color-dashboard{{ selected=='dashboard' ? ' selected' }}">
5 ...
```

Distinguishable

Pada subbab ini dibahas kriteria sukses yang termasuk kedalam kategori *distinguishable*. Kategori ini mensyaratkan agar konten dapat dilihat dan didengar dengan mudah, termasuk memisahkan latar depan dari latar belakang.

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.

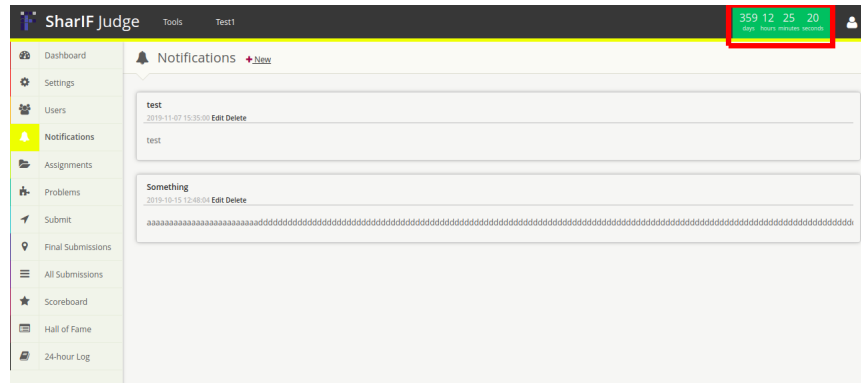
Kriteria Sukses 1.4.2 *Audio Control* (Sukses)

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

Kriteria Sukses 1.4.3 *Contrast (Minimum)* (Tidak Sukses)

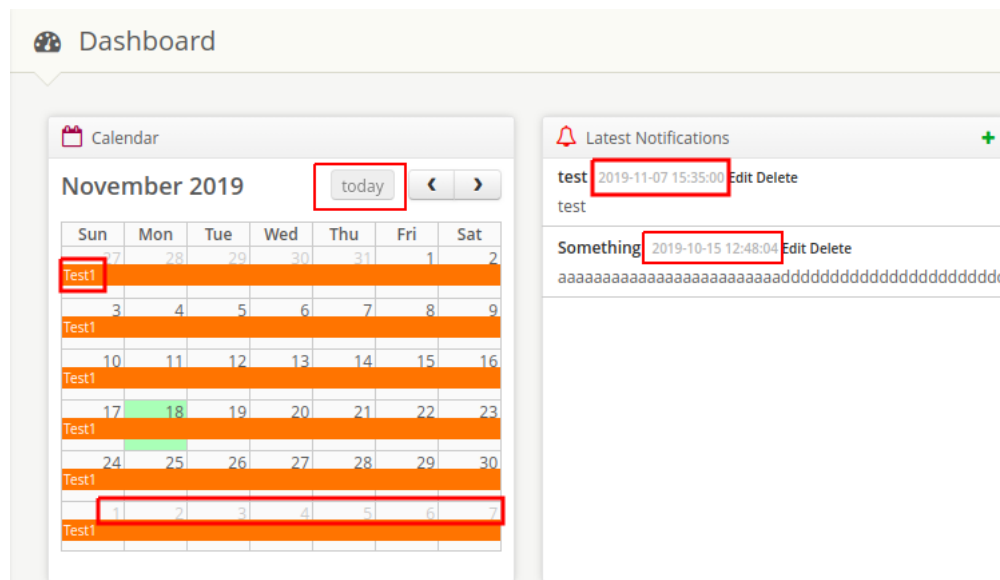
Kriteria ini tidak sukses dipatuhi karena pada aplikasi *SharIF Judge* terdapat beberapa teks dengan rasio kontras kurang dari 4.5:1, antara lain:

- 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 3.5.



Gambar 3.5: Pelanggaran 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 3.6.



Gambar 3.6: Pelanggaran 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 3.7.

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 3.9: Pelanggaran 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 3.10.

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;
```

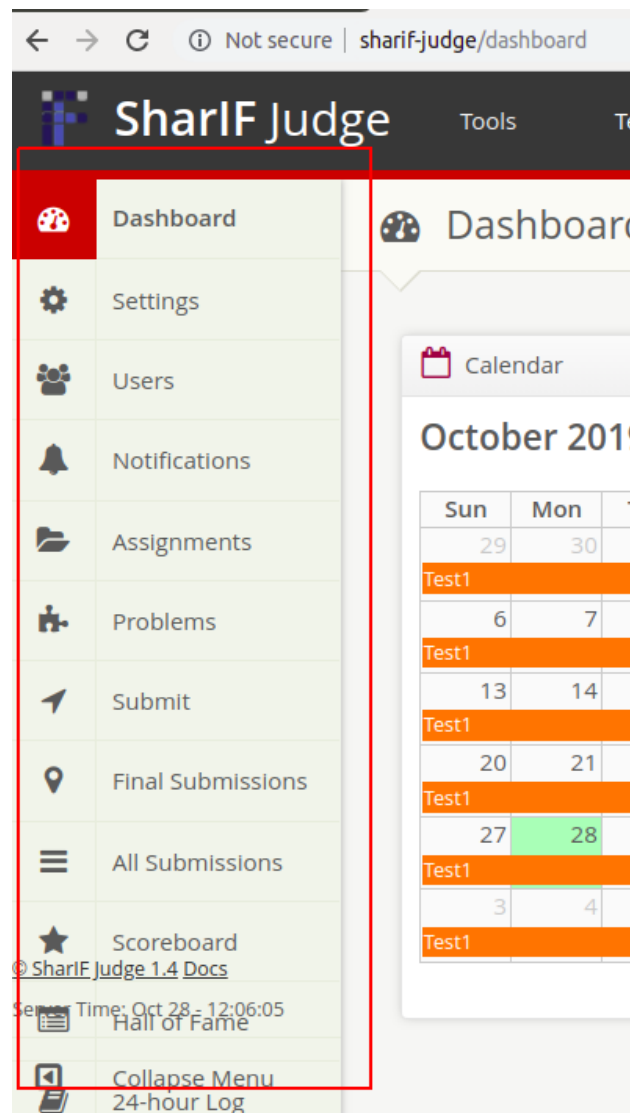
Problems [+](#)

	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 3.10: Pelanggaran Kriteria Sukses 1.4.3 - Kontras Elemen *Add Assignments*

Kriteria Sukses 1.4.4 *Resize text* (Tidak Sukses)

Kriteria ini tidak sukses dipatuhi karena konten *navigation bar* akan hilang pada pembesaran 125 persen. Tampilan halaman web dapat dilihat pada gambar 3.11.



Gambar 3.11: Pelanggaran Kriteria Sukses 1.4.4 - *Sidenav*

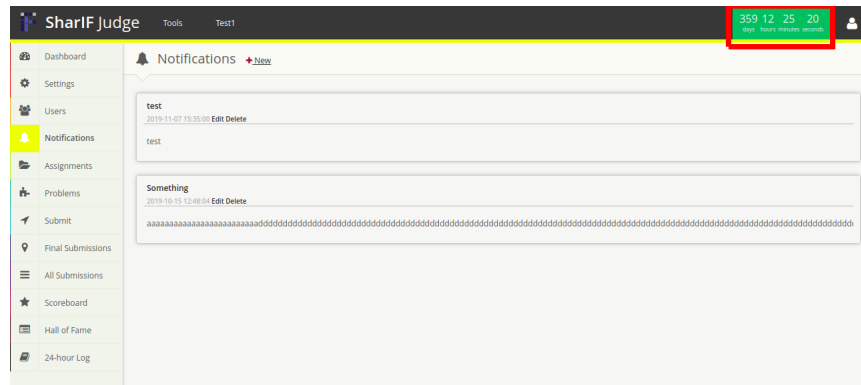
Kriteria Sukses 1.4.5 *Images of Text* (Sukses)

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

Kriteria Sukses 1.4.6 *Contrast (Enhanced)* (Tidak Sukses)

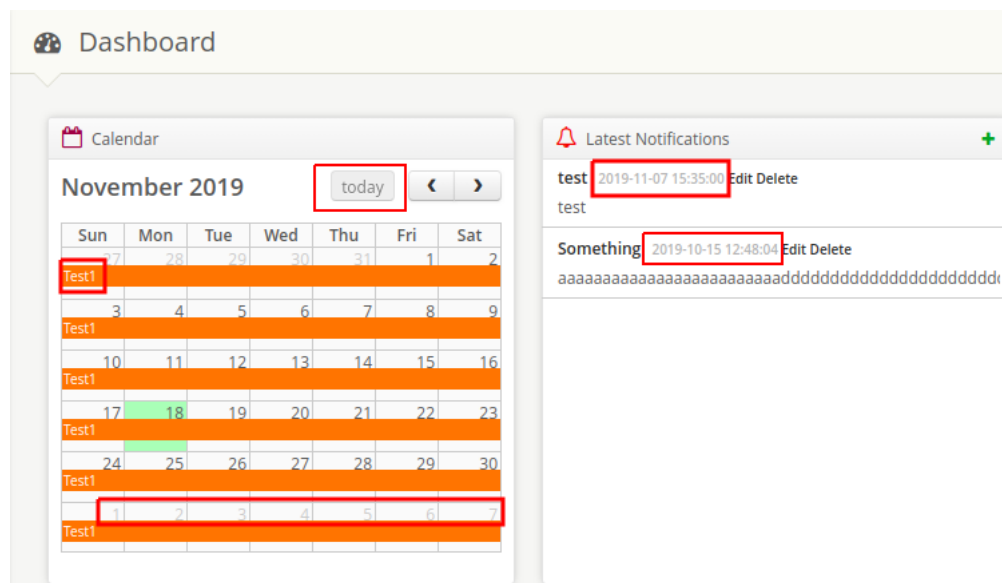
Kriteria ini tidak sukses dipatuhi karena pada aplikasi *SharIF Judge* terdapat beberapa teks dengan rasio kontras kurang dari 4.5:1, antara lain:

- 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 3.12.



Gambar 3.12: Pelanggaran Kriteria Sukses 1.4.6 - 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 3.13.

Gambar 3.13: Pelanggaran Kriteria Sukses 1.4.6 - 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 3.14.

Settings [Help](#)

Timezone [list of timezones](#) Server Time: Nov 18 - 14:31:47
 Asia/Jakarta

Week Start Day Sunday

Full Path to tester /var/www/SharIF-judge/restricted/

Full Path to assignments /var/www/SharIF-judge/restricted/

Upload Size Limit (kB) 50

Output Size Limit (kB) 1024
 Sets a limit for size of output file generated by submitted code

Results Per Page 40

In "All Submissions" Enter 0 for no limit

Results Per Page 80

In "Final Submissions" Enter 0 for no limit

Registration ☐ Open Public Registration.

Registration Code 0
 If you want to enable registration (above option), It is better to give a registration code to students in your class for validating registration. Enter 0 to disable.

Gambar 3.14: Pelanggaran Kriteria Sukses 1.4.6 - 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 3.15.

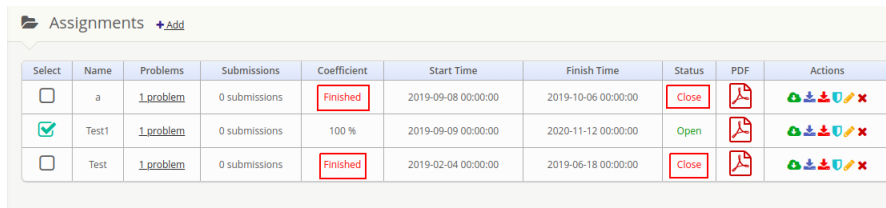
Notifications [+ New](#)

test
 2019-11-07 15:35:00 [Edit](#) [Delete](#)
 test

Something
 2019-10-15 12:48:04 [Edit](#) [Delete](#)
 aaaaaaaaaaaaaaaaaaaaaaaaaaddd

Gambar 3.15: Pelanggaran Kriteria Sukses 1.4.6 - Kontras Elemen *Notifications*

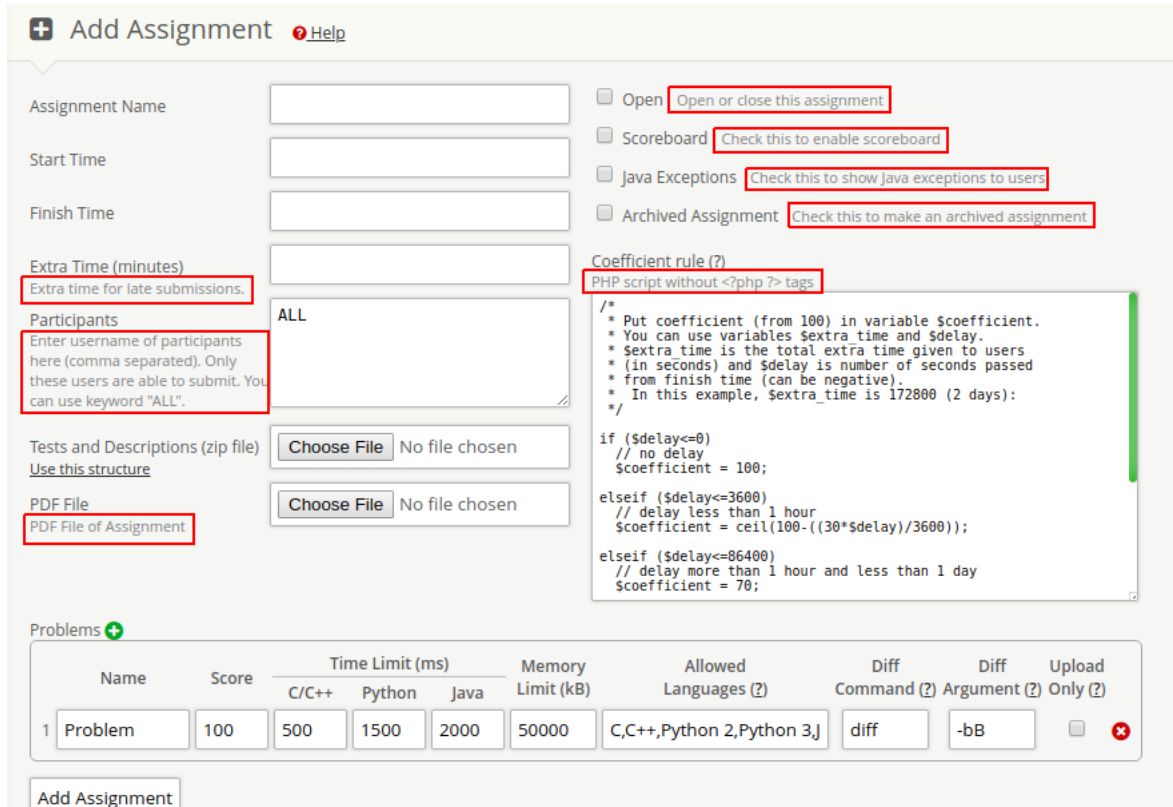
- Halaman *Assignment*: teks yang berwarna merah pada tabel *Assignment* memiliki rasio kontras 3.89:1 terhadap warna latar belakangnya. Tampilan halaman web dapat dilihat pada gambar 3.16.



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 3.16: Pelanggaran 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 3.17.



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 [+](#)

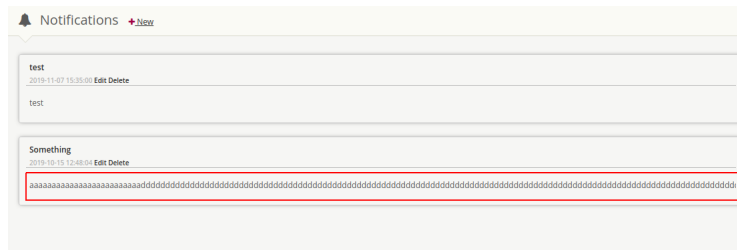
	Name	Score	Time Limit (ms)			Memory Limit (kB)	Allowed Languages (?)	Diff Command (?)	Diff Argument (?)	Upload Only (?)
			C/C++	Python	Java					
1	Problem	100	500	1500	2000	50000	C,C++,Python 2,Python 3,J	diff	-bB	<input type="checkbox"/>

Gambar 3.17: Pelanggaran Kriteria Sukses 1.4.6 - Kontras Elemen *Add Assignments***Kriteria Sukses 1.4.7 *Low or 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 3.18.

Gambar 3.18: Pelanggaran 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 3.19

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

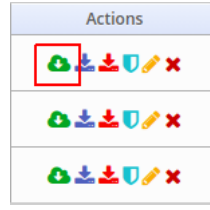
Gambar 3.19: Pelanggaran Kriteria Sukses 1.4.10 - *Horizontal Scroll***Kriteria Sukses 1.4.11 *Non-text Contrast* (Diabaikan)**

Kriteria ini diabaikan karena pengujiannya terlalu banyak dan tidak ada alat bantu untuk mempermudah pengujian ini. Tampilan halaman web dapat dilihat pada gambar 3.20.

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 3.20: Pelanggaran Kriteria Sukses 1.4.11 - Ikon Pada Halaman *Assignments*

Sebagai contoh perhitungan berikut adalah gambar ikon *Download Test and Description*.

Gambar 3.21: Ikon *Download Test and Description*

Ikon *Download Test and Description* memiliki spesifikasi warna sebagai berikut:

$$R_{8bit} = 3$$

$$G_{8bit} = 166$$

$$B_{8bit} = 34$$

Sedangkan warna latar belakangnya memiliki spesifikasi warna sebagai berikut:

$$R_{8bit} = 252$$

$$G_{8bit} = 252$$

$$B_{8bit} = 252$$

Rumus perhitungan kontras dapat dilihat pada subbab 2.1.1. Berikut adalah contoh perhitungan *relative luminance* untuk ikon *Download Test and Description* pada gambar 3.21:

$$R_{sRGB} = 3 \div 255 = 0.01176$$

$$G_{sRGB} = 166 \div 255 = 0.65098$$

$$B_{sRGB} = 34 \div 255 = 1.33333$$

$$R = 0.01176 \div 12.92 = 0.00091$$

$$G = ((0.65098 + 0.055) \div 1.055)^{2.4} = 0.38132$$

$$B = ((1.33333 + 0.055) \div 1.055)^{2.4} = 1.93275$$

$$L = 0.2126 \times 0.00091 + 0.7152 \times 0.38132 + 0.0722 \times 1.93275 = 0.41245$$

Sedangkan *relative luminance* latar belakang pada ikon *Download Test and Description* di gambar 3.21 sebagai berikut:

$$R_{sRGB} = 252 \div 255 = 0.98823$$

$$G_{sRGB} = 252 \div 255 = 0.98823$$

$$B_{sRGB} = 252 \div 255 = 0.98823$$

$$R = ((0.98823 + 0.055) \div 1.055)^{2.4} = 0.97343$$

$$G = ((0.98823 + 0.055) \div 1.055)^{2.4} = 0.97343$$

$$B = ((0.98823 + 0.055) \div 1.055)^{2.4} = 0.97343$$

$$L = 0.2126 \times 0.97343 + 0.7152 \times 0.97343 + 0.0722 \times 0.97343 = 0.97343$$

Hasil perbandingan kontras ikon *Download Test and Description* dengan warna latar belakangnya

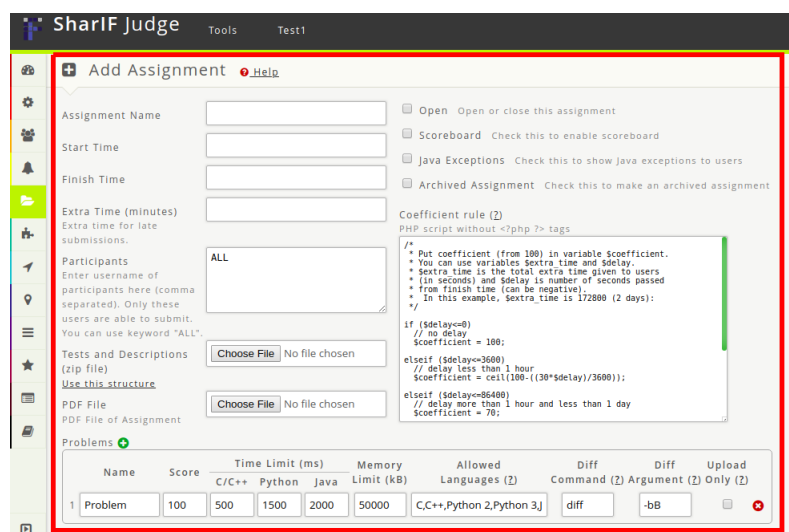
pada gambar 3.21 sebagai berikut:

$$\begin{aligned} \text{Contrast Ratio} &= (0.97343 + 0.05) : (0.41245 + 0.05) \\ &= 2.21306 : 1 \end{aligned}$$

Dari hasil perhitungan di atas ditemukan bahwa perbandingan rasio kontras ikon *Download Test and Description* dengan warna latar belakangnya pada gambar 3.21 bernilai 2.21306 : 1.

Kriteria Sukses 1.4.12 *Text Spacing* (Sukses)

Kriteria ini sukses dipatuhi karena penulis telah melakukan uji coba dengan cara mengubah seluruh *text spacing* 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 3.22.



Gambar 3.22: Pelanggaran Kriteria Sukses 1.4.12 - *Text Spacing* 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.

3.2.2 *Operable*

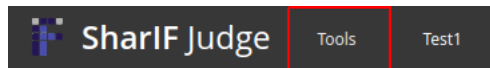
Berikut adalah kepatuhan *SharIF Judge* berdasarkan prinsip *operable*.

Keyboard Accessible

Pada subbab ini dibahas kriteria sukses yang termasuk kedalam kategori *Keyboard Accessible*. Kategori ini mensyaratkan semua fungsionalitas dapat diakses dengan *keyboard*.

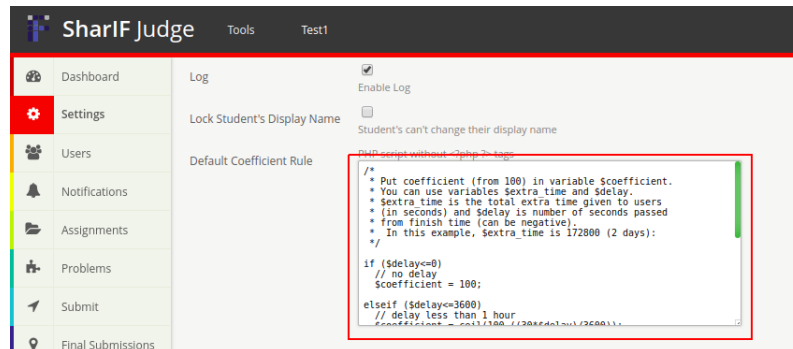
Kriteria Sukses 2.1.1 *Keyboard* (Tidak Sukses)

Kriteria ini tidak sukses dipatuhi karena masih banyak elemen yang tidak dioperasikan dengan keyboard, salah satunya tombol *Tools* pada *Menu*. Tampilan halaman web dapat dilihat pada gambar 3.23.

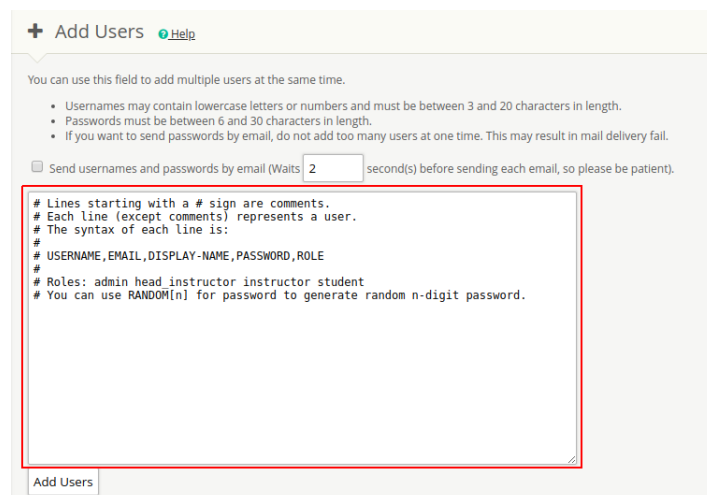
Gambar 3.23: Pelanggaran Kriteria Sukses 2.1.1 - Tombol *Tools***Kriteria Sukses 2.1.2 *No Keyboard Trap*** (Tidak Sukses)

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 3.24.

Gambar 3.24: Pelanggaran Kriteria Sukses 2.1.2 - *Keyboard Trap* Pada Halaman *Settings*

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

Gambar 3.25: Pelanggaran Kriteria Sukses 2.1.2 - *Keyboard Trap* Pada Halaman *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 3.26.

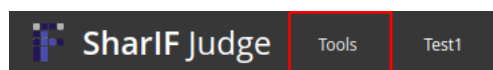
Gambar 3.26: Pelanggaran Kriteria Sukses 2.1.2 - *Keyboard Trap* Pada Halaman *Add Assignment*

- Dalam halaman *Problems (Edit Markdown)* ada *input field* yang fokusnya tidak dapat dipindahkan menggunakan antarmuka *keyboard*. *Input field* yang dimaksud dapat dilihat pada gambar 3.27.

Gambar 3.27: Pelanggaran Kriteria Sukses 2.1.2 - *Keyboard Trap* Pada Halaman *Problems* Bagian *Edit Markdown*

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

Kriteria ini tidak sukses dipatuhi karena masih banyak elemen yang tidak dioperasikan dengan keyboard, salah satunya tombol *Tools* pada *Menu*. Tampilan halaman web dapat dilihat pada gambar 3.28.



Gambar 3.28: Pelanggaran Kriteria Sukses 2.1.1 - Tombol *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.

Enough Time

Pada subbab ini dibahas kriteria sukses yang termasuk kedalam kategori *enough time*. Kategori ini mensyaratkan agar tersedianya waktu yang cukup untuk pengguna membaca dan memanfaatkan konten.

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.

Kode 3.6: Pelanggaran Kriteria Sukses 2.2.6 - Sesi Autentikasi

```
1 ...  
2 $config['sess_cookie_name']      = 'shjsession';  
3 $config['sess_expiration']       = 7200;  
4 ...
```

Seizures and Physical Reactions

Pada subbab ini dibahas kriteria sukses yang termasuk kedalam kategori *seizures and physical reactions*. Kategori ini mensyaratkan agar desain konten dirancang dan ditampilkan tidak menyebabkan kejang atau reaksi fisik bagi pengguna.

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 tiga 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.

Navigable

Pada subbab ini dibahas kriteria sukses yang termasuk kedalam kategori *navigable*. Kategori ini mensyaratkan agar tersedianya cara untuk membantu pengguna bernavigasi, mencari konten, dan menentukan di mana mereka berada.

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.

Kode 3.7: Pelanggaran Kriteria Sukses 2.4.4 - Gambar PDF Tidak Diberi Teks Alternatif

```

1    ...
2    <td>
3        <a href="{{ site_url('assignments/pdf/'~item.id) }}"></a>
5    </td>
6    ...

```

- 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.

Kode 3.8: Pelanggaran Kriteria Sukses 2.4.4 - Gambar Logo *Profile* Tidak Diberi Teks Alternatif

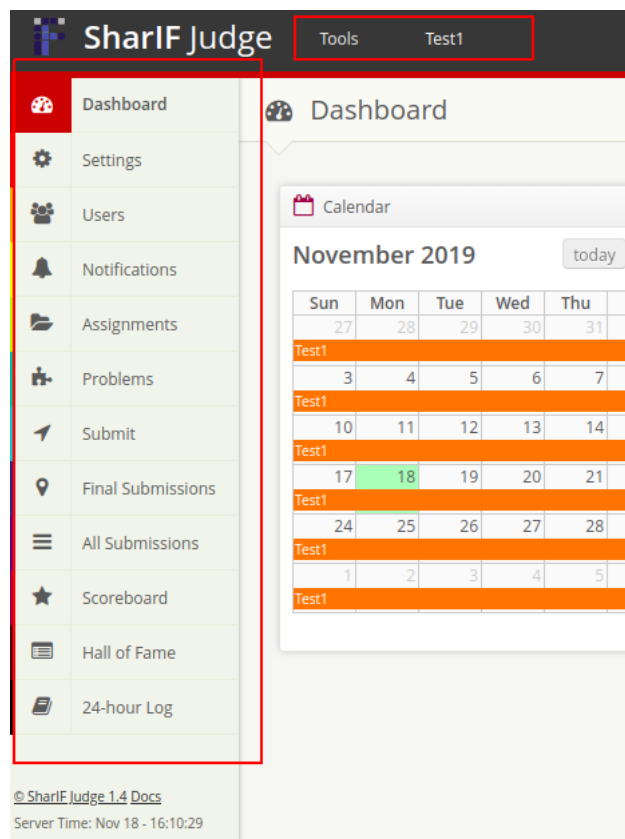
```

1    ...
2    <a href="{{ site_url('profile') }}" id="profile_link"><i class="fa fa-
3    user"></i></a>
4    ...

```

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: Pelanggaran 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.

Kode 3.9: Pelanggaran Kriteria Sukses 2.4.6 - Halaman *Add User* Tidak Memiliki Label

```

1  ...
2  <p class="input_p">
3      <textarea name="new_users" id="new_users" rows="20" cols="80" class
4      ="sharif_input">
5          # Lines starting with a # sign are comments.
6  ...

```

- 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.

Kode 3.10: Pelanggaran Kriteria Sukses 2.4.6 - Halaman *Add Assignment* Tidak Memiliki Label

```

1  ...
2  <td><input type="text" name="name[]" class="sharif_input short" value="{{
3  problem.name }}" /></td>
4  ...

```

- 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.

Kode 3.11: Pelanggaran Kriteria Sukses 2.4.6 - Halaman *Problems* bagian *Edit Markdown* Tidak Memiliki Label

```

1  ...
2  <p class="input_p">
3      <textarea name="text">{{ problem.description }}</textarea>
4  </p>
5  ...
6

```

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.

Kode 3.12: Pelanggaran Kriteria Sukses 2.4.9 - Gambar PDF Tidak Diberi Teks Alternatif

```

1  ...
2  <td>
3  <a href="{ site_url('assignments/pdf/'~item.id) }"></a>
4  </td>
5  ...
6

```

- 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.

Kode 3.13: Pelanggaran Kriteria Sukses 2.4.9 - Gambar Logo *Profile* Tidak Diberi Teks Alternatif

```

1  ...
2  <a href="{ site_url('profile') }" id="profile_link"><i class="fa fa-
user"></i></a>
3  ...
4

```

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*.

Kode 3.14: Pelanggaran Kriteria Sukses 2.4.10 - *Title Heading* Tidak Dipakai

```

1  ...
2  <div id="page_title">
3    <i class="fa {% block icon %}{% endblock %}"></i>
4    <span dir="auto">{% block title %}{% endblock %}</span>
5    {% block title_menu %}{% endblock %}
6  </div>
7  ...

```

Input Modalities

Pada subbab ini dibahas kriteria sukses yang termasuk kedalam kategori *input modalities*. Kategori ini mensyaratkan agar pengguna dapat mengoperasikan fungsionalitas dengan mudah melalui berbagai *input* selain *keyboard*.

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 *multipoint* 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)

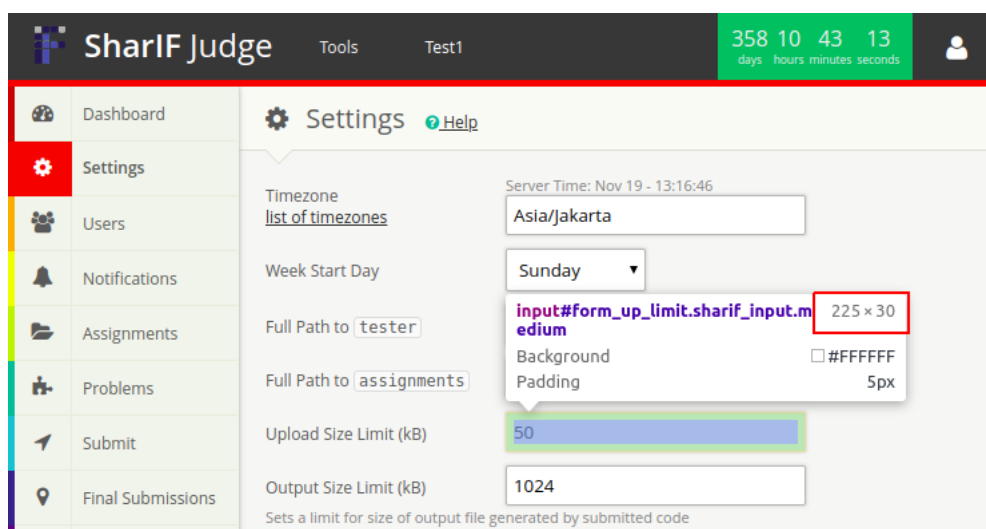
Kriteria ini sukses dipatuhi karena pada aplikasi *SharIF Judge* komponen dengan label yang menyertakan teks atau gambar teks, nama tersebut berisi teks yang ditampilkan secara visual.

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: Pelanggaran Kriteria Sukses 2.5.5 - Bidang Masukan Berukuran 30 *CSS Pixel*

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.

3.2.3 *Understandable*

Berikut adalah kepatuhan *SharIF Judge* berdasarkan prinsip *understandable*.

Readable

Pada subbab ini dibahas kriteria sukses yang termasuk kedalam kategori *readable*. Kategori ini mensyaratkan agar konten teks dapat dibaca dan dipahami.

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*.

Kode 3.15: Pelanggaran Kriteria Sukses 3.1.1 - Elemen *HTML* Tidak Memiliki Atribut *lang*

```
1 ...  
2 <html>  
3 <head>  
4 ...
```

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.

Predictable

Pada subbab ini dibahas kriteria sukses yang termasuk kedalam kategori *predictable*. Kategori ini mensyaratkan agar halaman web ditampilkan dan dapat dioperasikan dengan cara yang mudah ditebak.

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.

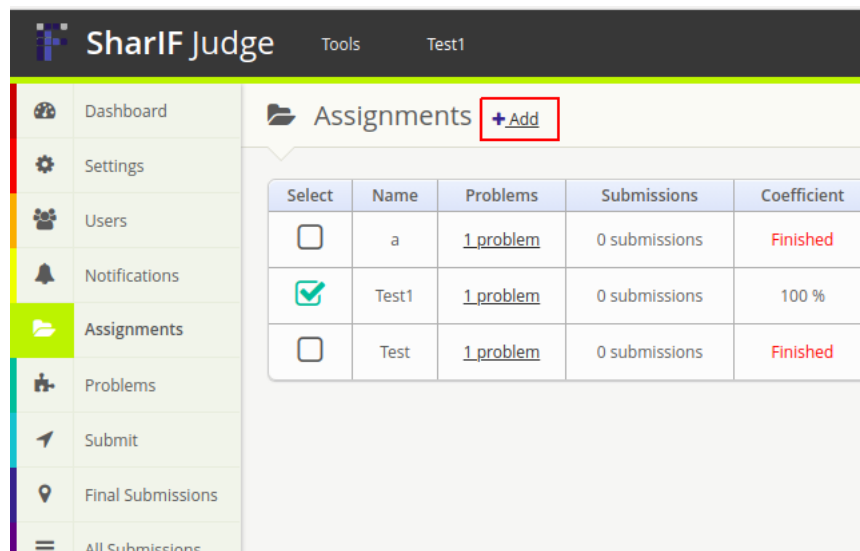
Kriteria Sukses 3.2.3 *Consistent Navigation* (Sukses)

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)

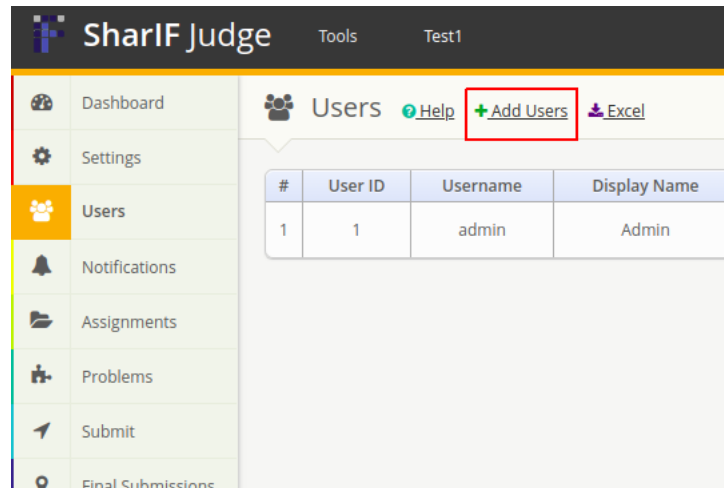
Kriteria ini tidak sukses dipatuhi karena ada komponen yang memiliki fungsi yang sama tetapi tidak diidentifikasi secara konsisten. Komponen tersebut antara lain:

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

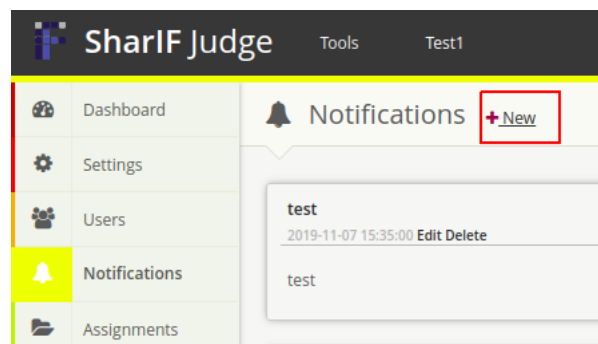


Gambar 3.31: Pelanggaran Kriteria Sukses 3.2.4 - *Assignments*

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

Gambar 3.32: Pelanggaran 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: Pelanggaran 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.

Input Assistance

Pada subbab ini dibahas kriteria sukses yang termasuk kedalam kategori *input assistance*. Kategori ini mensyaratkan agar tersedianya bantuan untuk pengguna menghindari kesalahan dan mengoreksi kesalahan tersebut.

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: Pelanggaran 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.

3.2.4 *Robust*

Berikut adalah kepatuhan *SharIF Judge* berdasarkan prinsip *robust*.

Compatible

Pada subbab ini dibahas kriteria sukses yang termasuk kedalam kategori *compatible*. Kategori ini mensyaratkan agar konten yang dibuat memiliki kompatibilitas dengan *user agent* saat ini maupun saat yang akan datang, termasuk teknologi alat bantu.

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.

Kode 3.16: Pelanggaran Kriteria Sukses 4.1.1 - Elemen Memiliki Atribut Duplikat

```

1 ...
2 <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') }}" />
3 ...

```

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.

3.3 Peningkatan ke *Level A*

Pada bagian ini dijelaskan secara singkat untuk meningkatkan kepatuhan aplikasi *SharIF Judge* menjadi *level A* dengan cara memenuhi 9 kriteria sukses *level A* yang belum terpenuhi dari total 30 kriteria sukses *level A*.

3.3.1 Kriteria Sukses 1.1.1 *Non-text Content*

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

- Logo *SharIF Judge* pada bagian menu atas. Listing dapat dilihat pada 3.1.
- Gambar *PDF* pada halaman *Assignment*. Listing dapat dilihat pada 3.2.

3.3.2 Kriteria Sukses 1.3.1 *Info and Relationships*

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*.
- Pemakaian *header* pada setiap judul halaman.

3.3.3 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.
- Menu pada logo *Profile* pada bagian menu atas.
- Memilih *Assignment* pada bagian menu atas.
- Aksi *Delete User* dan *Delete Submissions* pada halaman *Users*.

- *Checkbox* untuk memilih *Assignment* pada halaman *Assignments*.
- *Add* dan *Delete Problems* pada halaman *Add Assignment*.
- Memilih *Submission* pada halaman *All Submissions*.
- Melihat kode pada halaman *Submission* dan *Final Submission*.
- Melihat status pada halaman *Submission* dan *Final Submission*.
- Melihat *log* pada halaman *Submission* dan *Final Submission*.
- Aksi *rejudge* pada halaman *Submission* dan *Final Submission*.

3.3.4 Kriteria Sukses 2.1.2 *No Keyboard Trap*

Kriteria ini dapat dipatuhi dengan cara memastikan tidak ada *keyboard trap* di semua halaman. Elemen yang harus diperbaiki adalah:

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

3.3.5 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.

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

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

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

3.3.7 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.

3.3.8 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* dan *Edit Problem Plain HTML*.
- *Dropdown* pada halaman *Problems*.
- Masukan *Upload File* pada halaman *Problems*.

3.3.9 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 DAN PENGUJIAN

Bab ini membahas bagaimana implementasi untuk menaikkan tingkat kepatuhan aplikasi *SharIF Judge*.

4.1 Implementasi

Pada bagian ini akan dibahas perbaikan apa saja yang dilakukan untuk membuat kriteria sukses pada subbab 3.3 yang belum sukses dipatuhi menjadi sukses dipatuhi. Setiap perubahan akan ditampilkan dalam format *diff* dengan jarak *indentation* sudah dimodifikasi untuk menghemat ruang.

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

Semua perubahan kode dapat dilihat pada listing A.1.

4.1.2 Implementasi Kriteria Sukses 1.3.1 *Info and 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`.
- *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`.
- *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`.
- *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`.
- *Dropdown* pada halaman *Problems* perlu diberi label untuk memberikan informasi elemen tersebut. Perubahan yang terjadi ada pada file `/application/views/pages/problems.twig`.
- 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`.

- Setiap judul bagian pada halaman *SharIF Judge* tidak memakai *tag heading*. Perubahan yang perlu dilakukan adalah dengan cara memakai *tag heading* pada setiap judul bagian. Perubahan yang terjadi ada pada file `/application/views/templates/base.twig`. Semua perubahan kode dapat dilihat pada listing [A.2](#).

4.1.3 Implementasi Kriteria Sukses 2.1.1 *Keyboard*

Berikut adalah perubahan yang perlu dilakukan untuk memenuhi kriteria sukses 2.1.1:

- Fokus *keyboard* tidak dapat terfokus pada menu *Tools* pada menu bagian atas. Hal ini dapat diperbaiki dengan cara menambahkan *tabindex* yang bernilai 0 sehingga fokus *keyboard* dapat terfokus di elemen tersebut. Perubahan dapat dilihat pada file `/application/views/templates/top_bar.twig`.
Menu *Tools* pada bagian menu atas memiliki *sub-menu* yang dapat muncul ketika *pointer cursor* berada di atasnya, menu ini tidak muncul ketika fokus *keyboard* berada pada menu *Tools*. Perlu ditambahkan fungsi untuk memunculkan *sub-menu* ketika fokus *keyboard* berada pada menu *Tools*. *Sub-menu* akan hilang jika fokus *keyboard* terfokus pada *sidebar*. Perubahan dapat dilihat pada file `/assets/js/shj_functions.js`.
- *Sub-menu Profile* pada bagian menu atas tidak muncul ketika fokus *keyboard* berada pada gambar *Profile*. Gambar *Profile* memiliki kelas yang sama seperti menu *Tools* sehingga perubahan yang dilakukan pada kelas tersebut dapat memperbaiki masalah ini.
- *Sub-menu* memilih *Assignment* pada bagian menu atas tidak muncul ketika fokus *keyboard* berada pada tautan *Assignment* sekarang. Tautan *Assignment* memiliki kelas yang sama seperti menu *Tools* sehingga perubahan yang dilakukan pada kelas tersebut dapat memperbaiki masalah ini.

Sub-menu memilih *Assignment* pada bagian menu atas tidak dapat difokuskan dengan *keyboard* sehingga perlu ditambahkan atribut *tabindex* yang bernilai 0 agar fokus *keyboard* dapat terfokus pada elemen tersebut. Perubahan dapat dilihat pada file `/application/views/templates/top_bar.twig`.

Pada *Sub-menu* memilih *Assignment* pada menu atas, pengguna tidak dapat memilih *Assignment* dengan menggunakan *keyboard*. Hal ini dapat diperbaiki dengan cara menambahkan fungsi yang jika dijalankan akan menjalankan fungsi yang sama ketika pengguna menekan salah satu dari *list Assignment*. Pengguna harus menekan tombol "enter" pada *keyboard* ketika fokus *keyboard* berada pada elemen ini untuk menjalankan fungsi tersebut. Perubahan dapat dilihat pada file `/assets/js/shj_functions.js`.

- Aksi untuk *Delete User* dan *Delete Submissions* pada halaman *Users* tidak dapat difokuskan dengan *keyboard* sehingga perlu diubah menjadi *button* agar dapat difokuskan dengan *keyboard*. Perubahan dapat dilihat pada file `/application/views/pages/admin/users.twig`.
- *checkbox* untuk memilih *Assignment* pada halaman *Assignments* tidak dapat difokuskan dengan *keyboard* sehingga perlu diberi atribut *tabindex* yang bernilai 0 agar fokus *keyboard* dapat terfokus pada elemen tersebut. Perubahan dapat dilihat pada file `/application/views/pages/assignments.twig`.
- Tombol *Add* dan *Delete Problems* pada halaman *Add Assignment* tidak dapat difokuskan *keyboard* sehingga perlu diubah menjadi *button* agar dapat difokuskan dengan *keyboard*. Perubahan dapat dilihat pada file `/application/views/pages/admin/add_assignment.twig`.
- *checkbox* pada halaman *All Submissions* tidak dapat difokuskan dengan *keyboard* sehingga perlu ditambahkan atribut *tabindex* yang bernilai 0 agar elemen dapat difokuskan dengan *keyboard*. Perubahan dapat dilihat pada file `/application/views/pages/submissions.twig`.
Aksi yang ada di *checkbox* pada halaman *All Submissions* tidak dapat dijalankan dengan *keyboard* sehingga perlu ditambahkan fungsi yang jika dijalankan akan menjalankan fungsi yang sama ketika pengguna menekan *checkbox* tersebut. Fungsi ini hanya bisa dijalankan jika fokus sedang berada pada *checkbox* dan pengguna menekan tombol "spacebar" pada *keyboard*. Perubahan dapat dilihat pada file `/assets/js/shj_submissions.js`.

Aksi melihat status, melihat kode, melihat *log* dan aksi *rejudge* pada halaman *All Submission* dan *Final Submissions* tidak dapat difokuskan dengan *keyboard* sehingga perlu ditambahkan atribut *tabindex* yang bernilai 0 agar elemen dapat difokuskan dengan *keyboard*. Selain itu fungsi dari elemen tersebut tidak dapat dijalankan ketika pengguna menekan tombol "enter" pada *keyboard* sehingga perlu ditambahkan sebuah fungsi yang menjalankan fungsi yang sama ketika pengguna menekan tombol tersebut. Perubahan dapat dilihat pada file */assets/js/shj_submissions.js*.

Semua perubahan kode dapat dilihat pada listing [A.3](#).

4.1.4 Implementasi Kriteria Sukses 2.1.2 *No Keyboard Trap*

Kriteria 2.1.2 dapat dipenuhi dengan cara menambahkan mekanisme untuk mengaktifkan dan mematikan fitur *indent* yang menggunakan tombol "tab" pada setiap halaman yang memakai fitur tersebut sehingga pengguna dapat melanjutkan navigasi pada aplikasi menggunakan tombol "tab". Mekanisme baru ini dapat diaktifkan/dimatikan dengan cara menekan tombol "esc". Perubahan kode dapat dilihat pada listing [A.4](#).

4.1.5 Implementasi Kriteria Sukses 2.4.1 *Bypass Blocks*

Kriteria 2.4.1 dapat dipenuhi dengan cara menambahkan tautan di awal setiap halaman untuk meloncati menu navigasi. Tautan hanya dapat dijalankan jika pengguna mengoperasikannya dengan *keyboard*. Perubahan dapat dilihat pada file */application/views/templates/base.twig*. Selain itu ada juga penambahan untuk *style* baru yang dapat dilihat pada file */assets/styles/main.css*.

Tidak semua *browser* dapat menangani *in-page links*. Secara visual fokus *keyboard* sudah berada pada lokasi target, tetapi sebenarnya fokus *keyboard* belum diset ke lokasi target. Masalah ini dapat diselesaikan dengan menggunakan *Javascript* untuk set fokus *keyboard* ke lokasi target. Perubahan ada pada file */assets/js/shj_functions.js*.

Semua perubahan kode dapat dilihat pada listing [A.5](#).

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*.
- 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*.
- Tautan pada gambar *PDF* perlu diberi label untuk menjelaskan tujuannya. Perubahan untuk tautan pada gambar *PDF* yang ada di halaman *Assignments* sudah dilakukan pada listing [A.1](#).
- Tautan yang ada pada tabel *Problems* pada halaman *Add Assignment* tidak memiliki label yang menjelaskan tujuannya, untuk itu perlu diberi label yang menjelaskan tujuannya. Perubahan dapat dilihat pada file */application/views/pages/admin/add_assignment.twig*.

Semua perubahan kode dapat dilihat pada listing [A.6](#).

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 listing [A.7](#).

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 listing A.2.
- *checkbox* dan *textarea* pada halaman *Add Users* perlu diberi label yang menjelaskan tujuan dari elemen tersebut. Perubahan sudah dilakukan pada listing A.2.
- *textarea* pada halaman *Edit Problem Markdown* perlu diberi label yang menjelaskan tujuan dari elemen tersebut. Perubahan sudah dilakukan pada listing A.2.
- *textarea* pada halaman *Edit Problem Plain HTML* perlu diberi label yang menjelaskan tujuan dari elemen tersebut. Perubahan sudah dilakukan pada listing A.2.
- *Dropdown* pada halaman *Problems* perlu diberi label yang menjelaskan tujuan dari elemen tersebut. Perubahan sudah dilakukan pada listing A.2.
- Masukan *Upload File* pada halaman *Problems* perlu diberi label yang menjelaskan tujuan dari elemen tersebut. Perubahan sudah dilakukan pada listing A.2.

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 listing A.8.

4.2 Pengujian

Pada bagian ini ditulis skenario pengujian dan hasil yang didapatkan dari setiap skenario pengujian. Tujuan dari pengujian ini untuk melihat perbaikan yang dilakukan pada subbab 4.1 berhasil atau tidak. Pengujian dilakukan dengan menggunakan perangkat komputer berupa laptop dengan sistem operasi Ubuntu, browser *Google Chrome*, dan *screen reader ChromeVox* sebagai teknologi alat bantu. Pengujian dilakukan pada server lokal milik penguji dan akun yang digunakan oleh penguji memiliki hak akses tak terbatas sehingga dapat menggunakan semua fitur yang terdapat pada halaman web *SharIF Judge*. Selain itu pengujian juga dilakukan dengan kondisi seakan-akan memiliki keterbatasan visual.

4.2.1 Skenario dan Hasil Pengujian

Pada bagian ini akan ditulis skenario dan hasil pengujian yang dilakukan untuk menguji perbaikan yang telah dilakukan. Setiap skenario pengujian ditulis dalam bentuk poin-poin yang menjelaskan cara untuk menggunakan fitur-fitur yang ada pada aplikasi *SharIF Judge* dan hasil pengujian dituliskan dalam bentuk tabel yang berisi langkah skenario pengujian, hasil pengujian, dan aksi yang dilakukan dalam pengujian.

Login

Berikut adalah langkah-langkah yang perlu dilakukan untuk *Login*:

1. Memasukkan alamat <http://sharif-judge/login>¹ pada *address bar browser*.
2. Mengisi *Username* pada bidang masukan yang sudah disediakan.
3. Mengisi *Password* pada bidang masukan yang sudah disediakan.
4. Menekan tombol *Login*.

Hasil pengujian *Login* dapat dilihat pada tabel 4.1.

¹url yang dipakai dalam sistem milik penguji

Tabel 4.1: Hasil pengujian *Login*

Langkah	Hasil	Aksi
1	Sukses	Menekan tombol <i>Tab</i> sampai <i>ChromeVox</i> membacakan " <i>address and search bar</i> " kemudian pengguna memasukkan alamat http://sharif-judge/login .
2	Sukses	Menekan tombol <i>Tab</i> sampai <i>ChromeVox</i> membacakan bidang masukan <i>Username</i> . Setelah itu pengguna memasukkan <i>Username</i> .
3	Sukses	Menekan tombol <i>Tab</i> sampai <i>ChromeVox</i> membacakan bidang masukan <i>Password</i> . Setelah itu pengguna memasukkan <i>Password</i> .
4	Sukses	Menekan tombol <i>Tab</i> sampai <i>ChromeVox</i> membacakan tombol <i>Login</i> . Setelah itu pengguna menekan tombol <i>Enter</i> pada <i>keyboard</i> .

Settings

Berikut adalah langkah-langkah yang perlu dilakukan untuk sunting *Settings*:

1. *Login* pada aplikasi *SharIF Judge*.
2. Memilih *Settings* pada menu *sidebar*.
3. Mengisi bidang masukan *Timezone* dengan nilai "*Asia/Jakarta*".
4. Mengubah nilai *Week Start Day* menjadi "*Sunday*".
5. Mengubah nilai *Registration* menjadi "*checked*".
6. Mengubah isi *Default Coefficient Rule* sesuai aturan yang sudah disediakan.
7. Menekan tombol *Save Changes* untuk menyimpan perubahan.

Hasil pengujian *Settings* dapat dilihat pada tabel 4.2.

Tabel 4.2: Hasil pengujian *Settings*

Langkah	Hasil	Aksi
1	Sukses	Langkah <i>Login</i> dapat dilihat pada subbab 4.2.1.
2	Sukses	Menekan tombol <i>Tab</i> untuk navigasi ke <i>sidebar</i> sampai <i>ChromeVox</i> membacakan tautan untuk menu <i>Settings</i> . Setelah itu pengguna menekan tombol <i>Enter</i> pada <i>keyboard</i> .
3	Sukses	Menekan tombol <i>Tab</i> sampai <i>ChromeVox</i> membacakan bidang masukan <i>Timezone</i> . Setelah itu pengguna mengisi dengan nilai " <i>Asia/Jakarta</i> ".
4	Sukses	Menekan tombol <i>Tab</i> sampai <i>ChromeVox</i> membacakan bidang masukan <i>Week Start Day</i> . Setelah itu pengguna mengubah isi <i>combobox</i> dengan menekan tombol panah atas atau bawah pada <i>keyboard</i> sampai <i>ChromeVox</i> membacakan " <i>Sunday</i> ".
5	Sukses	Menekan tombol <i>Tab</i> sampai <i>ChromeVox</i> membacakan bidang masukan <i>Default Coefficient Rule</i> . Setelah itu pengguna mengisi sesuai spesifikasi yang sudah ditentukan. Pengguna dapat melanjutkan navigasi dengan menekan tombol <i>Tab</i> setelah mematikan fitur <i>tab indent</i> pada <i>textarea</i> dengan menekan tombol <i>Esc</i> pada <i>keyboard</i> .
6	Sukses	Menekan tombol <i>Tab</i> sampai <i>ChromeVox</i> membacakan tombol <i>Save Changes</i> kemudian menekan tombol <i>Enter</i> pada <i>keyboard</i> .

Add Users

Berikut adalah langkah-langkah yang perlu dilakukan untuk menambah *User*:

1. *Login* pada aplikasi *SharIF Judge*.
2. Memilih *Users* pada menu *sidebar*.

3. Menekan tombol *Add Users*.
4. Mengubah nilai *Send mail* menjadi *"checked"*.
5. Mengubah interval untuk mengirim pesan menjadi *"2"*.
6. Mengisi perintah untuk membuat *User* pada bidang masukan yang sudah disediakan.
7. Menekan tombol *Add Users* untuk menambah *User*.

Hasil pengujian *Add Users* dapat dilihat pada tabel 4.3.

Tabel 4.3: Hasil pengujian *Add Users*

Langkah	Hasil	Aksi
1	Sukses	Langkah <i>Login</i> dapat dilihat pada subbab 4.2.1.
2	Sukses	Menekan tombol <i>Tab</i> untuk navigasi ke <i>sidebar</i> sampai <i>ChromeVox</i> membacakan tautan untuk menu <i>Users</i> . Setelah itu pengguna menekan tombol <i>Enter</i> pada <i>keyboard</i> .
3	Sukses	Menekan tombol <i>Tab</i> sampai <i>ChromeVox</i> membacakan tautan <i>Add Users</i> kemudian pengguna menekan tombol <i>Enter</i> pada <i>keyboard</i> .
4	Sukses	Menekan tombol <i>Tab</i> sampai <i>ChromeVox</i> membacakan <i>checkbox</i> untuk <i>Send mail</i> . Kemudian pengguna menekan <i>space bar</i> pada <i>keyboard</i> untuk mengubah nilainya menjadi <i>"checked"</i> .
5	Sukses	Menekan tombol <i>Tab</i> sampai <i>ChromeVox</i> membacakan <i>"Delay"</i> . Setelah itu pengguna mengisi dengan nilai <i>"2"</i> .
6	Sukses	Menekan tombol <i>Tab</i> sampai <i>ChromeVox</i> membacakan <i>"Command for creating new user"</i> . Setelah itu pengguna mengisi sesuai spesifikasi yang sudah ditentukan. Pengguna dapat melanjutkan navigasi dengan menekan tombol <i>Tab</i> setelah mematikan fitur <i>tab indent</i> pada <i>textarea</i> dengan menekan tombol <i>Esc</i> pada <i>keyboard</i> .
7	Sukses	Menekan tombol <i>Tab</i> sampai <i>ChromeVox</i> membacakan tombol <i>Add Users</i> kemudian menekan tombol <i>Enter</i> pada <i>keyboard</i> .

Delete User

Berikut adalah langkah-langkah yang perlu dilakukan untuk menghapus *User*:

1. *Login* pada aplikasi *SharIF Judge*.
2. Memilih *Users* pada menu *sidebar*.
3. Mencari nama *User* yang akan dihapus pada tabel *Users*.
4. Menekan tombol *Delete User* pada kolom *action* di baris yang sesuai.
5. Menekan tombol *Yes, Delete* pada *popup* yang muncul.

Hasil pengujian *Delete User* dapat dilihat pada tabel 4.4.

Tabel 4.4: Hasil pengujian *Delete User*

Langkah	Hasil	Aksi
1	Sukses	Langkah <i>Login</i> dapat dilihat pada subbab 4.2.1.
2	Sukses	Menekan tombol <i>Tab</i> untuk navigasi ke <i>sidebar</i> sampai <i>ChromeVox</i> membacakan tautan untuk menu <i>Users</i> . Setelah itu pengguna menekan tombol <i>Enter</i> pada <i>keyboard</i> .
3	Sukses	Menavigasikan <i>ChromeVox</i> sampai membacakan baris <i>User</i> yang akan dihapus pada tabel.
4	Sukses	Menekan tombol navigasi maju <i>ChromeVox</i> sampai membacakan <i>Delete User</i> kemudian pengguna menekan tombol <i>Enter</i> pada <i>keyboard</i> .
5	Sukses	Menekan tombol navigasi maju <i>ChromeVox</i> sampai membacakan tombol <i>Yes, Delete</i> kemudian pengguna menekan tombol <i>Enter</i> pada <i>keyboard</i> .

Delete User Submissions

Berikut adalah langkah-langkah yang perlu dilakukan untuk menghapus *User submissions*:

1. *Login* pada aplikasi *SharIF Judge*.
2. Memilih *Users* pada menu *sidebar*.
3. Mencari nama *User* yang akan dihapus *Submissions* pada tabel *Users*.
4. Menekan tombol *Delete Submissions* pada kolom *action* di baris yang sesuai.
5. Menekan tombol *Yes, Delete* pada *popup* yang muncul.

Hasil pengujian *Delete User Submissions* dapat dilihat pada tabel 4.5.

Tabel 4.5: Hasil pengujian *Delete User Submissions*

Langkah	Hasil	Aksi
1	Sukses	Langkah <i>Login</i> dapat dilihat pada subbab 4.2.1.
2	Sukses	Menekan tombol <i>Tab</i> untuk navigasi ke <i>sidebar</i> sampai <i>ChromeVox</i> membacakan tautan untuk menu <i>Users</i> . Setelah itu pengguna menekan tombol <i>Enter</i> pada <i>keyboard</i> .
3	Sukses	Menavigasikan <i>ChromeVox</i> sampai membacakan baris <i>User</i> yang akan dihapus <i>Submissions</i> pada tabel.
4	Sukses	Menekan tombol navigasi maju <i>ChromeVox</i> sampai membacakan <i>Delete User Submissions</i> kemudian pengguna menekan tombol <i>Enter</i> pada <i>keyboard</i> .
5	Sukses	Menekan tombol navigasi maju <i>ChromeVox</i> sampai membacakan tombol <i>Yes, Delete</i> kemudian pengguna menekan tombol <i>Enter</i> pada <i>keyboard</i> .

Edit User

Berikut adalah langkah-langkah yang perlu dilakukan untuk sunting *User*:

1. *Login* pada aplikasi *SharIF Judge*.
2. Memilih *Users* pada menu *sidebar*.
3. Mencari nama *user* yang di *edit* pada tabel *Users*.
4. Menekan tombol *Edit* pada kolom *action* di baris yang sesuai.
5. Mengisi isi bidang masukan *Name* menjadi "*student*".
6. Mengubah isi *User Role* menjadi "*student*".
7. Menekan tombol *Save* untuk menyimpan perubahan.

Hasil pengujian *Edit User* dapat dilihat pada tabel 4.6.

Tabel 4.6: Hasil pengujian *Edit User*

Langkah	Hasil	Aksi
1	Sukses	Langkah <i>Login</i> dapat dilihat pada subbab 4.2.1.
2	Sukses	Menekan tombol <i>Tab</i> untuk navigasi ke <i>sidebar</i> sampai <i>ChromeVox</i> membacakan tautan untuk menu <i>Users</i> . Setelah itu pengguna menekan tombol <i>Enter</i> pada <i>keyboard</i> .
3	Sukses	Menavigasikan <i>ChromeVox</i> sampai membacakan baris <i>User</i> yang akan dihapus <i>Submissions</i> pada tabel.
4	Sukses	Menekan tombol navigasi maju <i>ChromeVox</i> sampai membacakan <i>Edit</i> kemudian pengguna menekan tombol <i>Enter</i> pada <i>keyboard</i> .
5	Sukses	Menekan tombol <i>Tab</i> sampai <i>ChromeVox</i> membacakan bidang masukan <i>Name</i> . Setelah itu pengguna mengisi dengan nilai " <i>student</i> ".
6	Sukses	Menekan tombol <i>Tab</i> sampai <i>ChromeVox</i> membacakan bidang masukan <i>User Role</i> . Setelah itu pengguna mengubah isi <i>combobox</i> dengan menekan tombol panah atas atau bawah pada <i>keyboard</i> sampai <i>ChromeVox</i> membacakan " <i>student</i> ".
7	Sukses	Menekan tombol <i>Tab</i> sampai <i>ChromeVox</i> membacakan tombol <i>Save</i> kemudian pengguna menekan tombol <i>Enter</i> pada <i>keyboard</i> .

Add Notification

Berikut adalah langkah-langkah yang perlu dilakukan untuk menambah notifikasi:

1. *Login* pada aplikasi *SharIF Judge*.
2. Memilih *Notifications* pada menu *sidebar*.
3. Menekan tautan *New*.
4. Mengisi judul pada bidang masukan *Title*.
5. Mengisi notifikasi pada bidang masukan *Text*.
6. Menekan tombol *Add* untuk menambah notifikasi.

Hasil pengujian *Add Notification* dapat dilihat pada tabel 4.7.

Tabel 4.7: Hasil pengujian *Add Notification*

Langkah	Hasil	Aksi
1	Sukses	Langkah <i>Login</i> dapat dilihat pada subbab 4.2.1.
2	Sukses	Menekan tombol <i>Tab</i> untuk navigasi ke <i>sidebar</i> sampai <i>ChromeVox</i> membacakan tautan untuk menu <i>Notifications</i> . Setelah itu pengguna menekan tombol <i>Enter</i> pada <i>keyboard</i> .
3	Sukses	Menekan tombol <i>Tab</i> sampai <i>ChromeVox</i> membacakan tautan <i>New</i> kemudian pengguna menekan tombol <i>Enter</i> pada <i>keyboard</i> .
4	Sukses	Menekan tombol <i>Tab</i> sampai <i>ChromeVox</i> membacakan bidang masukan <i>Title</i> . Setelah itu pengguna mengisi dengan nilai " <i>Judul</i> ".
5	Sukses	Menekan tombol <i>Tab</i> sampai <i>ChromeVox</i> membacakan bidang masukan <i>Text</i> . Setelah itu pengguna mengisi dengan nilai " <i>Isi Notifikasi</i> ".
6	Sukses	Menekan tombol <i>Tab</i> sampai <i>ChromeVox</i> membacakan tombol <i>Add</i> kemudian pengguna menekan tombol <i>Enter</i> pada <i>keyboard</i> .

Add Assignment

Berikut adalah langkah-langkah yang perlu dilakukan untuk menambah *Assignment*:

1. *Login* pada aplikasi *SharIF Judge*.

2. Memilih *Assignments* pada menu *sidebar*.
 3. Menekan tautan *Add*.
 4. Mengisi bidang masukan *Assignment Name* dengan nilai "*Assignment 1*".
 5. Mengisi bidang masukan *Participants* dengan nilai "*ALL*".
 6. Mengunggah *PDF File* dengan *file PDF* yang sesuai.
 7. Mengubah nilai *Scoreboard* menjadi "*checked*".
 8. Menekan tombol *Add Assignment* untuk menambah *Assignment*.
- Hasil pengujian *Add Assignment* dapat dilihat pada tabel 4.8.

Tabel 4.8: Hasil pengujian *Add Assignment*

Langkah	Hasil	Aksi
1	Sukses	Langkah <i>Login</i> dapat dilihat pada subbab 4.2.1.
2	Sukses	Menekan tombol <i>Tab</i> untuk navigasi ke <i>sidebar</i> sampai <i>ChromeVox</i> membacakan tautan untuk menu <i>Assignments</i> . Setelah itu pengguna menekan tombol <i>Enter</i> pada <i>keyboard</i> .
3	Sukses	Menekan tombol <i>Tab</i> sampai <i>ChromeVox</i> membacakan tautan <i>Add</i> kemudian pengguna menekan tombol <i>Enter</i> pada <i>keyboard</i> .
4	Sukses	Menekan tombol <i>Tab</i> sampai <i>ChromeVox</i> membacakan bidang masukan <i>Assignment Name</i> . Setelah itu pengguna mengisi dengan nilai " <i>Assignment 1</i> ".
5	Sukses	Menekan tombol <i>Tab</i> sampai <i>ChromeVox</i> membacakan bidang masukan <i>Participants</i> . Setelah itu pengguna mengisi dengan nilai " <i>All</i> ". Pengguna dapat melanjutkan navigasi dengan menekan tombol <i>Tab</i> setelah mematikan fitur <i>tab indent</i> pada <i>textarea</i> dengan menekan tombol <i>Esc</i> pada <i>keyboard</i> .
6	Sukses	Menekan tombol <i>Tab</i> sampai <i>ChromeVox</i> membacakan <i>PDF File</i> kemudian pengguna menekan tombol <i>Enter</i> pada <i>keyboard</i> . Setelah itu pengguna memilih <i>file PDF</i> yang akan diunggah.
7	Sukses	Menekan tombol <i>Tab</i> sampai <i>ChromeVox</i> membacakan <i>checkbox</i> untuk <i>Scoreboard</i> . Kemudian pengguna menekan <i>space bar</i> pada <i>keyboard</i> untuk mengubah nilainya menjadi " <i>checked</i> ".
8	Sukses	Menekan tombol <i>Tab</i> sampai <i>ChromeVox</i> membacakan tombol <i>Add Assignment</i> kemudian pengguna menekan tombol <i>Enter</i> pada <i>keyboard</i> .

Delete Assignment

Berikut adalah langkah-langkah yang perlu dilakukan untuk menghapus *Assignment*:

1. *Login* pada aplikasi *SharIF Judge*.
2. Memilih *Assignments* pada menu *sidebar*.
3. Mencari nama *Assignment* yang akan di hapus pada tabel *Assignments*.
4. Menekan tombol *Delete* pada kolom *action* di baris yang sesuai.
5. Menekan tombol *Delete this assignment* untuk menghapus *Assignment*.

Hasil pengujian *Delete Assignment* dapat dilihat pada tabel 4.9.

Tabel 4.9: Hasil pengujian *Delete Assignment*

Langkah	Hasil	Aksi
1	Sukses	Langkah <i>Login</i> dapat dilihat pada subbab 4.2.1 .
2	Sukses	Menekan tombol <i>Tab</i> untuk navigasi ke <i>sidebar</i> sampai <i>ChromeVox</i> membacakan tautan untuk menu <i>Assignments</i> . Setelah itu pengguna menekan tombol <i>Enter</i> pada <i>keyboard</i> .
3	Sukses	Menavigasikan <i>ChromeVox</i> sampai membacakan baris <i>Assignment</i> yang akan dihapus pada tabel.
4	Sukses	Menekan tombol navigasi maju <i>ChromeVox</i> sampai membacakan <i>Delete</i> kemudian pengguna menekan tombol <i>Enter</i> pada <i>keyboard</i> .
5	Sukses	Menekan tombol <i>Tab</i> sampai <i>ChromeVox</i> membacakan tombol <i>Delete this assignment</i> kemudian pengguna menekan tombol <i>Enter</i> pada <i>keyboard</i> .

Edit Assignment

Berikut adalah langkah-langkah yang perlu dilakukan untuk sunting *Assignment*:

1. *Login* pada aplikasi *SharIF Judge*.
2. Memilih *Assignments* pada menu *sidebar*.
3. Mencari nama *Assignment* yang akan disunting pada tabel *Assignments*.
4. Menekan tombol *Edit* pada kolom *action* di baris yang sesuai.
5. Mengisi bidang masukan *Extra Time* dengan nilai *"0*60"*.
6. Mengisi bidang masukan *Participants* dengan nilai *"ALL"*.
7. Mengunggah *PDF File* dengan file *PDF* yang sesuai.
8. Mengubah nilai *Open* menjadi *"checked"*.
9. Menekan tombol *Edit Assignment* untuk menyimpan perubahan.

Hasil pengujian *Edit Assignment* dapat dilihat pada tabel [4.10](#).

Tabel 4.10: Hasil pengujian *Edit Assignment*

Langkah	Hasil	Aksi
1	Sukses	Langkah <i>Login</i> dapat dilihat pada subbab 4.2.1.
2	Sukses	Menekan tombol <i>Tab</i> untuk navigasi ke <i>sidebar</i> sampai <i>ChromeVox</i> membacakan tautan untuk menu <i>Assignments</i> . Setelah itu pengguna menekan tombol <i>Enter</i> pada <i>keyboard</i> .
3	Sukses	Menavigasikan <i>ChromeVox</i> sampai membacakan baris <i>Assignment</i> yang akan disunting pada tabel.
4	Sukses	Menekan tombol navigasi maju <i>ChromeVox</i> sampai membacakan <i>Edit</i> kemudian pengguna menekan tombol <i>Enter</i> pada <i>keyboard</i> .
5	Sukses	Menekan tombol <i>Tab</i> sampai <i>ChromeVox</i> membacakan bidang masukan <i>Extra Time</i> . Setelah itu pengguna mengisi dengan nilai "0*60".
6	Sukses	Menekan tombol <i>Tab</i> sampai <i>ChromeVox</i> membacakan bidang masukan <i>Participants</i> . Setelah itu pengguna mengisi dengan nilai "All". Pengguna dapat melanjutkan navigasi dengan menekan tombol <i>Tab</i> setelah mematikan fitur <i>tab indent</i> pada <i>textarea</i> dengan menekan tombol <i>Esc</i> pada <i>keyboard</i> .
7	Sukses	Menekan tombol <i>Tab</i> sampai <i>ChromeVox</i> membacakan <i>PDF File</i> kemudian pengguna menekan tombol <i>Enter</i> pada <i>keyboard</i> . Setelah itu pengguna memilih <i>file PDF</i> yang akan diunggah.
8	Sukses	Menekan tombol <i>Tab</i> sampai <i>ChromeVox</i> membacakan <i>checkbox</i> untuk <i>Open</i> . Kemudian pengguna menekan <i>space bar</i> pada <i>keyboard</i> untuk mengubah nilainya menjadi "checked".
9	Sukses	Menekan tombol <i>Tab</i> sampai <i>ChromeVox</i> membacakan tombol <i>Edit Assignment</i> kemudian pengguna menekan tombol <i>Enter</i> pada <i>keyboard</i> .

Edit Problem Description(Markdown)

Berikut adalah langkah-langkah yang perlu dilakukan untuk sunting deskripsi masalah dalam format *markdown*:

1. *Login* pada aplikasi *SharIF Judge*.
2. Memilih *Assignment* pada menu bagian atas.
3. Memilih *Problems* pada menu *sidebar*.
4. Memilih *Problem* pada tabel *Problems*.
5. Menekan tautan *Edit Markdown*.
6. Mengisi deskripsi *Problem* pada *textarea*.
7. Menekan tombol *Save* untuk menyimpan perubahan.

Hasil pengujian *Edit Problem Description (Markdown)* dapat dilihat pada tabel 4.11.

Tabel 4.11: Hasil pengujian *Edit Problem Description (Markdown)*

Langkah	Hasil	Aksi
1	Sukses	Langkah <i>Login</i> dapat dilihat pada subbab 4.2.1.
2	Sukses	Menekan tombol <i>Tab</i> untuk navigasi ke <i>topbar</i> sampai <i>ChromeVox</i> membacakan <i>Assignment</i> yang akan disunting deskripsinya. Setelah itu pengguna menekan tombol <i>Enter</i> pada <i>keyboard</i> .
3	Sukses	Menekan tombol <i>Tab</i> untuk navigasi ke <i>sidebar</i> sampai <i>ChromeVox</i> membacakan tautan untuk menu <i>Problems</i> . Setelah itu pengguna menekan tombol <i>Enter</i> pada <i>keyboard</i> .
4	Sukses	Menekan tombol <i>Tab</i> sampai <i>ChromeVox</i> membacakan <i>Problem</i> pada tabel yang akan disunting deskripsinya kemudian pengguna menekan tombol <i>Enter</i> pada <i>keyboard</i> .
5	Sukses	Menekan tombol <i>Tab</i> sampai <i>ChromeVox</i> membacakan tautan <i>Edit Markdown</i> kemudian pengguna menekan tombol <i>Enter</i> pada <i>keyboard</i> .
6	Sukses	Menekan tombol <i>Tab</i> sampai <i>ChromeVox</i> membacakan bidang masukan <i>Markdown Editor</i> . Setelah itu pengguna mengisi deskripsi sesuai dengan spesifikasinya. Pengguna dapat melanjutkan navigasi dengan menekan tombol <i>Tab</i> setelah mematikan fitur <i>tab indent</i> pada <i>textarea</i> dengan menekan tombol <i>Esc</i> pada <i>keyboard</i> .
7	Sukses	Menekan tombol <i>Tab</i> sampai <i>ChromeVox</i> membacakan tombol <i>Save</i> kemudian pengguna menekan tombol <i>Enter</i> pada <i>keyboard</i> .

Edit Problem Description(HTML)

Berikut adalah langkah-langkah yang perlu dilakukan untuk sunting deskripsi masalah dalam format *HTML*:

1. *Login* pada aplikasi *SharIF Judge*.
2. Memilih *Assignment* pada menu bagian atas.
3. Memilih *Problems* pada menu *sidebar*.
4. Memilih *Problem* pada tabel *Problems*.
5. Menekan tautan *Edit HTML*.
6. Mengisi deskripsi *Problem* pada *text editor*.
7. Menekan tombol *Save* untuk menyimpan perubahan.

Hasil pengujian *Edit Problem Description (HTML)* dapat dilihat pada tabel 4.12.

Tabel 4.12: Hasil pengujian *Edit Problem Description (HTML)*

Langkah	Hasil	Aksi
1	Sukses	Langkah <i>Login</i> dapat dilihat pada subbab 4.2.1.
2	Sukses	Menekan tombol <i>Tab</i> untuk navigasi ke <i>topbar</i> sampai <i>ChromeVox</i> membacakan <i>Assignment</i> yang akan disunting deskripsinya. Setelah itu pengguna menekan tombol <i>Enter</i> pada <i>keyboard</i> .
3	Sukses	Menekan tombol <i>Tab</i> untuk navigasi ke <i>sidebar</i> sampai <i>ChromeVox</i> membacakan tautan untuk menu <i>Problems</i> . Setelah itu pengguna menekan tombol <i>Enter</i> pada <i>keyboard</i> .
4	Sukses	Menekan tombol <i>Tab</i> sampai <i>ChromeVox</i> membacakan <i>Problem</i> pada tabel yang akan disunting deskripsinya kemudian pengguna menekan tombol <i>Enter</i> pada <i>keyboard</i> .
5	Sukses	Menekan tombol <i>Tab</i> sampai <i>ChromeVox</i> membacakan tautan <i>Edit HTML</i> kemudian pengguna menekan tombol <i>Enter</i> pada <i>keyboard</i> .
6	Sukses	Menekan tombol <i>Tab</i> sampai <i>ChromeVox</i> membacakan bidang masukan <i>Edit Text</i> .
7	Sukses	Menekan tombol <i>Tab</i> sampai <i>ChromeVox</i> membacakan tombol <i>Save</i> kemudian pengguna menekan tombol <i>Enter</i> pada <i>keyboard</i> .

Edit Problem Description(HTML)

Berikut adalah langkah-langkah yang perlu dilakukan untuk sunting deskripsi masalah dalam format *plain HTML*:

1. *Login* pada aplikasi *SharIF Judge*.
2. Memilih *Assignment* pada menu bagian atas.
3. Memilih *Problems* pada menu *sidebar*.
4. Memilih *Problem* pada tabel *Problems*.
5. Menekan tautan *Edit Plain HTML*.
6. Mengisi deskripsi *Problem* pada *textarea*.
7. Menekan tombol *Save* untuk menyimpan perubahan.

Hasil pengujian *Edit Problem Description (Plain HTML)* dapat dilihat pada tabel 4.13.

Tabel 4.13: Hasil pengujian *Edit Problem Description (Plain HTML)*

Langkah	Hasil	Aksi
1	Sukses	Langkah <i>Login</i> dapat dilihat pada subbab 4.2.1.
2	Sukses	Menekan tombol <i>Tab</i> untuk navigasi ke <i>topbar</i> sampai <i>ChromeVox</i> membacakan <i>Assignment</i> yang akan disunting deskripsinya. Setelah itu pengguna menekan tombol <i>Enter</i> pada <i>keyboard</i> .
3	Sukses	Menekan tombol <i>Tab</i> untuk navigasi ke <i>sidebar</i> sampai <i>ChromeVox</i> membacakan tautan untuk menu <i>Problems</i> . Setelah itu pengguna menekan tombol <i>Enter</i> pada <i>keyboard</i> .
4	Sukses	Menekan tombol <i>Tab</i> sampai <i>ChromeVox</i> membacakan <i>Problem</i> pada tabel yang akan disunting deskripsinya kemudian pengguna menekan tombol <i>Enter</i> pada <i>keyboard</i> .
5	Sukses	Menekan tombol <i>Tab</i> sampai <i>ChromeVox</i> membacakan tautan <i>Edit Plain HTML</i> kemudian pengguna menekan tombol <i>Enter</i> pada <i>keyboard</i> .
6	Sukses	Menekan tombol <i>Tab</i> sampai <i>ChromeVox</i> membacakan bidang masukan <i>HTML Editor</i> .
7	Sukses	Menekan tombol <i>Tab</i> sampai <i>ChromeVox</i> membacakan tombol <i>Save</i> kemudian pengguna menekan tombol <i>Enter</i> pada <i>keyboard</i> .

Submit File

Berikut adalah langkah-langkah yang perlu dilakukan untuk mengumpulkan *file*:

1. *Login* pada aplikasi *SharIF Judge*.
2. Memilih *Assignment* pada menu bagian atas.
3. Memilih *Submit* pada menu *sidebar*.
4. Memilih *Problem* pada *dropdown Problem*.
5. Memilih bahasa pemrograman pada *dropdown Language*.
6. Memilih *file* yang akan dikumpulkan.
7. Menekan tombol *Submit* untuk mengumpulkan *file*.

Hasil pengujian untuk mengumpulkan *file* dapat dilihat pada tabel 4.14.

Tabel 4.14: Hasil pengujian *Submit*

Langkah	Hasil	Aksi
1	Sukses	Langkah <i>Login</i> dapat dilihat pada subbab 4.2.1.
2	Sukses	Menekan tombol <i>Tab</i> untuk navigasi ke <i>topbar</i> sampai <i>ChromeVox</i> membacakan <i>Assignment</i> yang sesuai. Setelah itu pengguna menekan tombol <i>Enter</i> pada <i>keyboard</i> .
3	Sukses	Menekan tombol <i>Tab</i> untuk navigasi ke <i>sidebar</i> sampai <i>ChromeVox</i> membacakan tautan untuk menu <i>Submit</i> . Setelah itu pengguna menekan tombol <i>Enter</i> pada <i>keyboard</i> .
4	Sukses	Menekan tombol <i>Tab</i> sampai <i>ChromeVox</i> membacakan bidang masukan <i>Problem</i> . Setelah itu pengguna mengubah isi <i>combobox</i> dengan menekan tombol panah atas atau bawah pada <i>keyboard</i> sampai <i>ChromeVox</i> membacakan " <i>Problem 1</i> ".
5	Sukses	Menekan tombol <i>Tab</i> sampai <i>ChromeVox</i> membacakan bidang masukan <i>Language</i> . Setelah itu pengguna mengubah isi <i>combobox</i> dengan menekan tombol panah atas atau bawah pada <i>keyboard</i> sampai <i>ChromeVox</i> membacakan " <i>Java</i> ".
6	Sukses	Menekan tombol <i>Tab</i> sampai <i>ChromeVox</i> membacakan bidang masukan <i>File</i> . Setelah itu pengguna menekan tombol <i>Enter</i> pada <i>keyboard</i> . Pengguna harus memilih <i>file</i> yang akan diunggah dengan benar.
7	Sukses	Menekan tombol <i>Tab</i> sampai <i>ChromeVox</i> membacakan tombol <i>Submit</i> kemudian pengguna menekan tombol <i>Enter</i> pada <i>keyboard</i> .

Final Submission

Berikut adalah langkah-langkah yang perlu dilakukan untuk memilih *Final Submission*:

1. *Login* pada aplikasi *SharIF Judge*.
 2. Memilih *Assignment* pada menu bagian atas.
 3. Memilih *All Submission* pada menu *sidebar*.
 4. Memilih *Submission* yang akan dijadikan *Final Submission* pada tabel *All Submissions*
 5. Menekan tombol pada kolom *Final* untuk menjadikan *Submission* menjadi *Final Submission*.
- Hasil pengujian untuk memilih *Final Submission* dapat dilihat pada tabel 4.15.

Tabel 4.15: Hasil pengujian *Final Submission*

Langkah	Hasil	Aksi
1	Sukses	Langkah <i>Login</i> dapat dilihat pada subbab 4.2.1.
2	Sukses	Menekan tombol <i>Tab</i> untuk navigasi ke <i>topbar</i> sampai <i>ChromeVox</i> membacakan <i>Assignment</i> yang sesuai. Setelah itu pengguna menekan tombol <i>Enter</i> pada <i>keyboard</i> .
3	Sukses	Menekan tombol <i>Tab</i> untuk navigasi ke <i>sidebar</i> sampai <i>ChromeVox</i> membacakan tautan untuk menu <i>All Submissions</i> . Setelah itu pengguna menekan tombol <i>Enter</i> pada <i>keyboard</i> .
4	Sukses	Menavigasikan <i>ChromeVox</i> sampai membacakan baris <i>Submission</i> yang akan menjadi <i>Final Submission</i> pada tabel.
5	Sukses	Menavigasikan <i>ChromeVox</i> sampai membacakan tombol <i>Set Final</i> pada <i>Submission</i> yang akan menjadi <i>Final Submission</i> pada tabel. Kemudian pengguna menekan tombol <i>Spacebar</i> pada <i>keyboard</i> .

4.2.2 Kesimpulan

Semua perubahan yang dilakukan pada aplikasi *SharIF Judge* telah meningkatkan tingkat kepatuhannya sampai tingkat *A*. Perubahan ini memungkinkan pengguna untuk memakai aplikasi *SharIF Judge* menggunakan *keyboard* dalam kondisi tidak dapat melihat dengan bantuan *screen reader ChromeVox*. Dengan perubahan yang dilakukan pada aplikasi *SharIF Judge* diharapkan pengguna yang memiliki disabilitas keterbatasan penglihatan dapat memakai aplikasi *SharIF Judge* tanpa menghadapi kesulitan.

BAB 5

KESIMPULAN DAN SARAN

Bab ini membahas tentang kesimpulan berdasarkan hasil dari analisis, implementasi dan pengujian, serta saran-saran untuk pengembangan selanjutnya.

5.1 Kesimpulan

Dari hasil penelitian yang dilakukan, maka diperoleh kesimpulan sebagai berikut:

- Aplikasi *SharIF Judge* dapat diperbaiki sehingga tingkat kepatuhannya terhadap *WCAG 2.1* mencapai tingkat *A*.
- Perbaikan pada aplikasi *SharIF Judge* memberi aksesibilitas untuk pengguna yang memiliki disabilitas kebutaan atau penglihatan rendah.
- Perbaikan yang dilakukan pada aplikasi *SharIF Judge* memungkinkan pengguna dapat menggunakan fitur-fitur yang ada dalam kondisi tidak dapat melihat.
- Untuk dapat menggunakan fitur-fitur yang ada pada aplikasi *SharIF Judge* dalam kondisi tidak dapat melihat, pengguna harus mengerti bagaimana cara menggunakan teknologi alat bantu yang digunakan.

5.2 Saran

Berdasarkan hasil penelitian yang dilakukan, berikut adalah saran-saran untuk pengembangan selanjutnya:

- Memperbaiki aplikasi *SharIF Judge* sehingga tingkat kepatuhan terhadap *WCAG 2.1* mencapai tingkat *AA* dan *AAA*.
- Melakukan pengujian aplikasi *SharIF Judge* terhadap pengguna yang benar-benar memiliki disabilitas.
- Melakukan pengujian aplikasi *SharIF Judge* untuk keterbatasan pendengaran, keterbatasan gerak, keterbatasan berbicara, dan fotosensitif.

DAFTAR REFERENSI

- [1] Sierkowski, B. (2002) Achieving web accessibility. *SIGUCCS '02: Proceedings of the 30th annual ACM SIGUCCS conference on User services*, Providence, USA, November, pp. 288–291. Association for Computing Machinery, New York.
- [2] Version 2.1 (2018) *Web Content Accessibility Guidelines*. W3C Web Accessibility Initiative (WAI). Massachusetts, USA.
- [3] Commit 02ce9a0 (2019) *SharIF-Judge*. Fakultas Teknologi Informasi dan Sains Universitas Katolik Parahyangan. Bandung, Indonesia.
- [4] Commit 642120b (2015) *Sharif-Judge*.
- [5] Version 3.0 (2013) *Codeigniter*. British Columbia Institute of Technology. Burnaby, Canada.

LAMPIRAN A

KODE PROGRAM

Kode A.1: Perubahan untuk mematuhi kriteria 1.1.1

```
1 diff --git a/application/views/pages/assignments.twig b/application/views/pages/
  assignments.twig
2 index f4091385..5d4bafa3 100644
3 --- a/application/views/pages/assignments.twig
4 +++ b/application/views/pages/assignments.twig
5 @@ -69,7 +69,7 @@
6 {% endif %}
7 </td>
8 <td>
9 -   <a href="{ { site_url('assignments/pdf/~item.id) } }"></a>
10 +   <a href="{ { site_url('assignments/pdf/~item.id) } }"></a>
11 </td>
12 {% if user.level > 0 %}
13 <td>
14
15 diff --git a/application/views/templates/top_bar.twig b/application/views/
  templates/top_bar.twig
16 index ccb67df9..57ed6ca4 100644
17 --- a/application/views/templates/top_bar.twig
18 +++ b/application/views/templates/top_bar.twig
19 @@ -41,7 +41,7 @@
20 </div>
21 <div id="shj_logo" class="top_left">
22 <a href="{ { site_url('/') } }">
23 -   
24 +   
25 <h1 class="shjlogo-text">SharIF <span>Judge</span></h1>
26 </a>
27 </div>
```

Kode A.2: Perubahan untuk mematuhi kriteria 1.3.1

```
1 diff --git a/application/views/pages/admin/add_assignment.twig b/application/
  views/pages/admin/add_assignment.twig
2 index 13a0ddea..0d3d0cb3 100644
```

```

3 --- a/application/views/pages/admin/add_assignment.twig
4 +++ b/application/views/pages/admin/add_assignment.twig
5 @@ -22,16 +22,16 @@
6 <script>
7 shj.num_of_problems={{ problems|length }};
8 shj.row='<tr><td>PID</td>\
9 -   <td><input type="text" name="name[]" class="sharif_input short" value="
   Problem "/></td>\
10 -   <td><input type="text" name="score[]" class="sharif_input tiny2" value
   ="100"/></td>\
11 -   <td><input type="text" name="c_time_limit[]" class="sharif_input tiny2"
   value="500"/></td>\
12 -   <td><input type="text" name="python_time_limit[]" class="sharif_input tiny2"
   value="1500"/></td>\
13 -   <td><input type="text" name="java_time_limit[]" class="sharif_input tiny2"
   value="2000"/></td>\
14 -   <td><input type="text" name="memory_limit[]" class="sharif_input tiny" value
   ="50000"/></td>\
15 -   <td><input type="text" name="languages[]" class="sharif_input short2" value
   ="C,C++,Python 2,Python 3,Java"/></td>\
16 -   <td><input type="text" name="diff_cmd[]" class="sharif_input tiny" value="
   diff"/></td>\
17 -   <td><input type="text" name="diff_arg[]" class="sharif_input tiny" value="-
   bB"/></td>\
18 -   <td><input type="checkbox" name="is_upload_only[]" class="check" value="PID
   "/><td><i class="fa fa-times-circle fa-lg color1 delete_problem pointer"></i
   ></td></td>\
19 +   <td><input aria-label="Problem Name" type="text" name="name[]" class="
   sharif_input short" value="Problem "/></td>\
20 +   <td><input aria-label="Score" type="text" name="score[]" class="sharif_input
   tiny2" value="100"/></td>\
21 +   <td><input aria-label="Time Limit for C" type="text" name="c_time_limit[]"
   class="sharif_input tiny2" value="500"/></td>\
22 +   <td><input aria-label="Time Limit for Python" type="text" name="
   python_time_limit[]" class="sharif_input tiny2" value="1500"/></td>\
23 +   <td><input aria-label="Time Limit for Java" type="text" name="
   java_time_limit[]" class="sharif_input tiny2" value="2000"/></td>\
24 +   <td><input aria-label="Memory Limit" type="text" name="memory_limit[]" class
   ="sharif_input tiny" value="50000"/></td>\
25 +   <td><input aria-label="Allowed Languages" type="text" name="languages[]"
   class="sharif_input short2" value="C,C++,Python 2,Python 3,Java"/></td>\
26 +   <td><input aria-label="Diff Command" type="text" name="diff_cmd[]" class="
   sharif_input tiny" value="diff"/></td>\
27 +   <td><input aria-label="Diff Argument" type="text" name="diff_arg[]" class="
   sharif_input tiny" value="-bB"/></td>\
28 +   <td><input aria-label="Upload Only" type="checkbox" name="is_upload_only[]"
   class="check" value="PID"/><td><i class="fa fa-times-circle fa-lg color1
   delete_problem pointer"></i></td></td>\
29 </tr>';
30 $(document).ready(function(){
31 $("#add").click(function(){

```

```

32 @@ -201,16 +201,16 @@
33 {% for problem in problems %}
34 <tr>
35 <td>{{ problem.id }}</td>
36 - <td><input type="text" name="name[]" class="sharif_input short" value="{{
  problem.name }}" /></td>
37 - <td><input type="text" name="score[]" class="sharif_input tiny2" value="{{
  problem.score }}" /></td>
38 - <td><input type="text" name="c_time_limit[]" class="sharif_input tiny2"
  value="{{ problem.c_time_limit }}" /></td>
39 - <td><input type="text" name="python_time_limit[]" class="sharif_input tiny2"
  value="{{ problem.python_time_limit }}" /></td>
40 - <td><input type="text" name="java_time_limit[]" class="sharif_input tiny2"
  value="{{ problem.java_time_limit }}" /></td>
41 - <td><input type="text" name="memory_limit[]" class="sharif_input tiny" value
  ="{{ problem.memory_limit }}" /></td>
42 - <td><input type="text" name="languages[]" class="sharif_input short2" value
  ="{{ problem.allowed_languages }}" /></td>
43 - <td><input type="text" name="diff_cmd[]" class="sharif_input tiny" value="{{
  problem.diff_cmd }}" /></td>
44 - <td><input type="text" name="diff_arg[]" class="sharif_input tiny" value="{{
  problem.diff_arg }}" /></td>
45 - <td><input type="checkbox" name="is_upload_only[]" class="check" value="{{
  problem.id }}" {{ problem.is_upload_only ? 'checked' }} /></td>
46 + <td><input aria-label="Problem Name" type="text" name="name[]" class="
  sharif_input short" value="{{ problem.name }}" /></td>
47 + <td><input aria-label="Score" type="text" name="score[]" class="sharif_input
  tiny2" value="{{ problem.score }}" /></td>
48 + <td><input aria-label="Time Limit for C" type="text" name="c_time_limit[]"
  class="sharif_input tiny2" value="{{ problem.c_time_limit }}" /></td>
49 + <td><input aria-label="Time Limit for Python" type="text" name="
  python_time_limit[]" class="sharif_input tiny2" value="{{ problem.
  python_time_limit }}" /></td>
50 + <td><input aria-label="Time Limit for Java" type="text" name="
  java_time_limit[]" class="sharif_input tiny2" value="{{ problem.
  java_time_limit }}" /></td>
51 + <td><input aria-label="Memory Limit" type="text" name="memory_limit[]" class
  ="sharif_input tiny" value="{{ problem.memory_limit }}" /></td>
52 + <td><input aria-label="Allowed Languages" type="text" name="languages[]"
  class="sharif_input short2" value="{{ problem.allowed_languages }}" /></td>
53 + <td><input aria-label="Diff Command" type="text" name="diff_cmd[]" class="
  sharif_input tiny" value="{{ problem.diff_cmd }}" /></td>
54 + <td><input aria-label="Diff Argument" type="text" name="diff_arg[]" class="
  sharif_input tiny" value="{{ problem.diff_arg }}" /></td>
55 + <td><input aria-label="Upload Only" type="checkbox" name="is_upload_only[]"
  class="check" value="{{ problem.id }}" {{ problem.is_upload_only ? 'checked'
  }} /></td>
56 <td><i class="fa fa-times-circle fa-lg color1 delete_problem pointer"></i></td>
57 </tr>
58 {% endfor %}
59

```

```

60 diff --git a/application/views/pages/admin/add_user.twig b/application/views/
    pages/admin/add_user.twig
61 index 7f7ef43d..0b85aeb6 100644
62 --- a/application/views/pages/admin/add_user.twig
63 +++ b/application/views/pages/admin/add_user.twig
64 @@ -56,10 +56,10 @@
65 <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>
66 </ul>
67 <p class="input_p">
68 -     <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).
69 +     <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).
70 </p>
71 <p class="input_p">
72 -     <textarea name="new_users" id="new_users" rows="20" cols="80" class="
        sharif_input">
73 +     <textarea name="new_users" id="new_users" rows="20" cols="80" class="
        sharif_input" aria-label="Command for creating new user">
74 # Lines starting with a # sign are comments.
75 # Each line (except comments) represents a user.
76 # The syntax of each line is:
77
78 diff --git a/application/views/pages/admin/edit_problem_md.twig b/application/
    views/pages/admin/edit_problem_md.twig
79 index a9bed8fb..e6fd7923 100644
80 --- a/application/views/pages/admin/edit_problem_md.twig
81 +++ b/application/views/pages/admin/edit_problem_md.twig
82 @@ -177,7 +177,7 @@ Violets are blue.
83 </p>
84 {{ form_open("problems/edit/md/#{description_assignment.id}/#{problem.id}") }}
85 <p class="input_p">
86 -     <textarea dir="auto" name="text" rows="30" cols="75" class="sharif_input" id
        ="md_editor">{{ problem.description }}</textarea>
87 +     <textarea dir="auto" name="text" rows="30" cols="75" class="sharif_input" id
        ="md_editor" aria-label="Markdown Editor">{{ problem.description }}</textarea>
88 </p>
89 <p class="input_p">
90 <input type="submit" value="Save" class="sharif_input"/>
91
92 diff --git a/application/views/pages/admin/edit_problem_plain.twig b/application/
    views/pages/admin/edit_problem_plain.twig
93 index 2693e63d..cf8b0e76 100644
94 --- a/application/views/pages/admin/edit_problem_plain.twig
95 +++ b/application/views/pages/admin/edit_problem_plain.twig
96 @@ -33,7 +33,7 @@

```



```

97 </p>
98 {{ form_open("problems/edit/html/#{description_assignment.id}/#{problem.id}") }}
99 <p class="input_p">
100 -   <textarea name="text" rows="30" cols="80" class="sharif_input" id="
      html_editor">{{ problem.description }}</textarea>
101 +   <textarea name="text" rows="30" cols="80" class="sharif_input" id="
      html_editor" aria-label="HTML Editor">{{ problem.description }}</textarea>
102 </p>
103 <p class="input_p">
104 <input type="submit" value="Save" class="sharif_input"/>
105
106 diff --git a/application/views/pages/problems.twig b/application/views/pages/
      problems.twig
107 index 4f66153e..d1c119fc 100644
108 --- a/application/views/pages/problems.twig
109 +++ b/application/views/pages/problems.twig
110 @@ -96,7 +96,7 @@ $(document).ready(function(){
111 <input type="hidden" name="problem" value="{{ problem.id }}" />
112
113 <p class="input_p">
114 -   <select id="languages" name="language" class="sharif_input full-width">
115 +   <select id="languages" name="language" class="sharif_input full-width" aria-
      label="Select Language">
116 <option value="0" selected="selected">-- Select Language --</option>
117 {% for l in problem.allowed_languages %}
118 <option value="{{ l }}">{{ l }}</option>
119 @@ -104,7 +104,7 @@ $(document).ready(function(){
120 </select>
121 </p>
122 <p class="input_p">
123 -   <input type="file" id="file" class="sharif_input full-width" name="userfile
      "/>
124 +   <input type="file" id="file" class="sharif_input full-width" name="userfile"
      aria-label="Upload File"/>
125 </p>
126 <p class="input_p">
127 <input type="submit" value="Submit" class="sharif_input"/>
128
129 diff --git a/application/views/templates/base.twig b/application/views/templates/
      base.twig
130 index e580f5de..047dc432 100644
131 --- a/application/views/templates/base.twig
132 +++ b/application/views/templates/base.twig
133 @@ -58,7 +58,7 @@ shj.color_scheme = 'github';
134
135 <div id="page_title">
136 <i class="fa {% block icon %}{% endblock %}"></i>
137 -   <span dir="auto">{% block title %}{% endblock %}</span>
138 +   <h1 dir="auto">{% block title %}{% endblock %}</h1>
139 {% block title_menu %}{% endblock %}
140 </div>

```

```

141 |
142 | diff --git a/assets/styles/main.css b/assets/styles/main.css
143 | index a0e61502..97fb8672 100644
144 | --- a/assets/styles/main.css
145 | +++ b/assets/styles/main.css
146 | @@ -571,6 +571,14 @@ div#page_title > i + span {
147 | margin-left: 8px;
148 | }
149 |
150 | + div#page_title > h1 {
151 | +   margin-left: 8px;
152 | +   display: inline !important;
153 | +   font-size: 20px !important;
154 | +   margin-block-start: 0em !important;
155 | +   margin-block-end: 0em !important;
156 | + }
157 | +
158 | span.title_menu_item {
159 | font-size: 12px;
160 | margin-left: 10px;

```

Kode A.3: Perubahan untuk mematuhi kriteria 2.1.1

```

1 |
2 | diff --git a/application/views/pages/admin/add_assignment.twig b/application/
  |   views/pages/admin/add_assignment.twig
3 | index 13a0ddea..57e753c4 100644
4 | --- a/application/views/pages/admin/add_assignment.twig
5 | +++ b/application/views/pages/admin/add_assignment.twig
6 | @@ -31,7 +31,8 @@
7 | <td><input type="text" name="languages[]" class="sharif_input short2" value="C,C
  |   ++,Python 2,Python 3,Java"/></td>\
8 | <td><input type="text" name="diff_cmd[]" class="sharif_input tiny" value="diff
  |   "/></td>\
9 | <td><input type="text" name="diff_arg[]" class="sharif_input tiny" value="-bB
  |   "/></td>\
10 | -   <td><input type="checkbox" name="is_upload_only[]" class="check" value="PID
  |   "/><td><i class="fa fa-times-circle fa-lg color1 delete_problem pointer"></i
  |   ></td></td>\
11 | +   <td><input type="checkbox" name="is_upload_only[]" class="check" value="PID
  |   "/></td>\
12 | +   <td><button class="delete_problem" type="button" aria-label="Delete Problem
  |   "><i class="fa fa-times-circle fa-lg color1 pointer"></i></button></td>\
13 | </tr>';
14 | $(document).ready(function(){
15 | $("#add").click(function(){
16 | @@ -178,7 +179,7 @@
17 | {{ form_error('late_rule', '<div class="shj_error">', '</div>')}}
18 | </p>
19 | </div>
20 | -   <p class="input_p" id="add_problems">Problems <i class="fa fa-plus-circle fa-
  |   lg color11 pointer" id="add"></i>

```

```

21 + <p class="input_p" id="add_problems">Problems <button type="button" id="add"
    aria-label="Add Problem"><i class="fa fa-plus-circle fa-lg color11 pointer
    "></i></button>
22 <table id="problems_table">
23 <thead>
24 <tr>
25 @@ -211,7 +212,7 @@
26 <td><input type="text" name="diff_cmd[]" class="sharif_input tiny" value="{{
    problem.diff_cmd }}" /></td>
27 <td><input type="text" name="diff_arg[]" class="sharif_input tiny" value="{{
    problem.diff_arg }}" /></td>
28 <td><input type="checkbox" name="is_upload_only[]" class="check" value="{{
    problem.id }}" {{ problem.is_upload_only ? 'checked' }} /></td>
29 - <td><i class="fa fa-times-circle fa-lg color1 delete_problem pointer"></i></
    td>
30 + <td><button class="delete_problem" type="button" aria-label="Delete Problem
    "><i class="fa fa-times-circle fa-lg color1 pointer"></i></td>
31 </tr>
32 {% endfor %}
33 </tbody>
34 diff --git a/application/views/pages/admin/users.twig b/application/views/pages/
    admin/users.twig
35 index 21d51fe2..d8bf0858 100644
36 --- a/application/views/pages/admin/users.twig
37 +++ b/application/views/pages/admin/users.twig
38 @@ -52,10 +52,10 @@
39 <td>{{ user.first_login_time ? user.first_login_time : 'Never' }}</td>
40 <td>{{ user.last_login_time ? user.last_login_time : 'Never' }}</td>
41 <td>
42 - <a title="Edit" href="{{ site_url('profile/'~user.id) }}"><i class="fa fa-
    pencil fa-lg color9"></i></a>
43 - <a title="Submissions" href="{{ site_url('submissions/all/user/'~user.
    username) }}"><i class="fa fa-bars fa-lg color12"></i></a>
44 - <span title="Delete User" class="delete_user pointer"><i title="Delete User"
    class="fa fa-times fa-lg color2"></i></span>
45 - <span title="Delete Submissions" class="delete_submissions pointer"><i class
    ="fa fa-times-circle fa-lg color1"></i></span>
46 + <a title="Edit User {{ user.username }}" href="{{ site_url('profile/'~user.
    id) }}"><i class="fa fa-pencil fa-lg color9"></i></a>
47 + <a title="Submissions User {{ user.username }}" href="{{ site_url('
    submissions/all/user/'~user.username) }}"><i class="fa fa-bars fa-lg color12
    "></i></a>
48 + <button class="delete_user"><span title="Delete User {{ user.username }}"
    class="pointer"><i title="Delete User" class="fa fa-times fa-lg color2"></i></
    span></button>
49 + <button class="delete_submissions" ><span title="Delete Submissions {{ user.
    username }}" class="pointer"><i class="fa fa-times-circle fa-lg color1"></i></
    span></button>
50 </td>
51 </tr>
52 {% endfor %}

```

```

53 diff --git a/application/views/pages/assignments.twig b/application/views/pages/
    assignments.twig
54 index f4091385..e6b13589 100644
55 --- a/application/views/pages/assignments.twig
56 +++ b/application/views/pages/assignments.twig
57 @@ -48,7 +48,7 @@
58 </thead>
59 {% for item in all_assignments|reverse %}
60 <tr>
61 -     <td><i class="pointer select_assignment fa {{ item.id == user.
        selected_assignment.id ? 'fa-check-square-o color6' : 'fa-square-o' }} fa-2x"
        data-id="{{ item.id }}"></i></td>
62 +     <td><i tabindex="0" class="pointer select_assignment fa {{ item.id == user.
        selected_assignment.id ? 'fa-check-square-o color6' : 'fa-square-o' }} fa-2x"
        data-id="{{ item.id }}"></i></td>
63 <td dir="auto">{{ item.name }}</td>
64 <td><a href="{{ site_url('problems/'~item.id) }}">{{ item.problems }} problem{{
        item.problems != 1 ? 's' }}</a></td>
65 <td>{{ item.total_submits }} submission{{ item.total_submits != 1 ? 's' }}</td>
66 diff --git a/application/views/pages/submissions.twig b/application/views/pages/
    submissions.twig
67 index 2f58f9cf..607c8370 100644
68 --- a/application/views/pages/submissions.twig
69 +++ b/application/views/pages/submissions.twig
70 @@ -108,7 +108,7 @@
71 <tr data-u="{{ submission.username }}" data-a="{{ submission.assignment }}" data-
    p="{{ submission.problem }}" data-s="{{ submission.submit_id }}" {% if view ==
        'final' and j is even %}class="hl"{% endif %}>
72 {% if view == 'all' %}
73 <td>
74 -     <i class="pointer set_final fa {{ submission.is_final ? 'fa-check-circle-o
        color11' : 'fa-circle-o' }} fa-2x"></i>
75 +     <i tabindex="0" aria-label="Set Final" role="checkbox" aria-checked="{{
        submission.is_final ? 'true' : 'false' }}" class="pointer set_final fa {{
        submission.is_final ? 'fa-check-circle-o color11' : 'fa-circle-o' }} fa-2x"></
        i>
76 </td>
77 {% endif %}
78 {% if user.level > 0 %}
79 @@ -148,7 +148,7 @@
80 {% else %}
81 {% set submission_class = 'btn shj-blue' %}
82 {% endif %}
83 -     <div class="{{ submission_class }}" data-type="result" >
84 +     <div tabindex="0" class="{{ submission_class }}" data-type="result" >
85 {% if submission.status == 'SCORE' %}
86 {{ submission.final_score }}
87 {% else %}
88 @@ -159,9 +159,9 @@
89 </td>
90 <td>

```

```

91 {% if submission.file_type == 'zip' or submission.file_type == 'pdf' or
    submission.file_type == 'txt' %}
92 - <div class="btn shj-orange" data-type="download">Download</div>
93 + <div tabindex="0" class="btn shj-orange" data-type="download">Download</div>
94 {% else %}
95 - <div class="btn shj-orange" data-type="code" >Code</div>
96 + <div tabindex="0" class="btn shj-orange" data-type="code" >Code</div>
97 {% endif %}
98 </td>
99 {% if user.level > 0 %}
100 @@ -169,13 +169,13 @@
101 {% if submission.status == 'Uploaded' %}
102 ---
103 {% else %}
104 - <div class="btn" data-type="log" >Log</div>
105 + <div tabindex="0" class="btn" data-type="log" >Log</div>
106 {% endif %}
107 </td>
108 {% endif %}
109 {% if user.level >= 2 %}
110 <td>
111 - <div class="shj_rejudge pointer"><i class="fa fa-refresh fa-lg color10"></i>
    </div>
112 + <div tabindex="0" class="shj_rejudge pointer"><i class="fa fa-refresh fa-lg
    color10"></i></div>
113 </td>
114 {% endif %}
115 </tr>
116 diff --git a/application/views/templates/top_bar.twig b/application/views/
    templates/top_bar.twig
117 index ccb67df9..d6b4fc0d 100644
118 --- a/application/views/templates/top_bar.twig
119 +++ b/application/views/templates/top_bar.twig
120 @@ -46,7 +46,7 @@
121 </a>
122 </div>
123 {% if user.level >= 2 %}
124 - <div class="top_object shj_menu top_left" id="admin_tools_top">
125 + <div class="top_object shj_menu top_left" tabindex="0" id="admin_tools_top">
126 Tools
127 <ul class="top_menu">
128 <li><a href="{ { site_url('rejudge') } }">Rejudge</a></li>
129 @@ -59,7 +59,7 @@
130 <a href="{ { site_url('assignments') } }"><span dir="auto" class="assignment_name
    ">{ { user.selected_assignment.name|length > 30 ? user.selected_assignment.name
    |slice(0, 30) ~ '...' : user.selected_assignment.name } }</span></a>
131 <ul class="top_menu" id="select_assignment_menu">
132 {% for assignment_item in all_assignments|reverse[:5] %}
133 - <li class="assignment_block select_assignment">
134 + <li class="assignment_block select_assignment" tabindex="0">
135 <i class="fa { { assignment_item.id == user.selected_assignment.id ? 'fa-check-
```

```

    square-o color6' : 'fa-square-o' }}" data-id="{{ assignment_item.id }}"></i>
136 <span class="assignment_item" dir="auto">{{ assignment_item.name }}</span>
137 </li>
138 diff --git a/assets/js/shj_functions.js b/assets/js/shj_functions.js
139 index c7b7140f..18c162d1 100644
140 --- a/assets/js/shj_functions.js
141 +++ b/assets/js/shj_functions.js
142 @@ -341,6 +341,16 @@ $(document).ready(function () {
143 * Top Bar
144 */
145 $(document).ready(function () {
146 +   $('shj_menu').on('focusin',function(e){
147 +       $(this).children("top_menu").show();
148 +       $(this).addClass('shj_white');
149 +       $("#side_bar").on('focusin', function (e) {
150 +           if(!$(e.target).parent().is('top_menu')){
151 +               $('shj_menu').children("top_menu").hide();
152 +               $('shj_menu').removeClass('shj_white');
153 +           }
154 +       })
155 +   });
156 $("#top_bar").hoverIntent({
157 over: function () {
158 $(this).children("top_menu").show();
159 @@ -352,6 +362,11 @@ $(document).ready(function () {
160 },
161 selector: '.top_object.shj_menu'
162 });
163 +   $(".select_assignment").on('keyup', function (e) {
164 +       if(e.which==13){
165 +           $(this).trigger("click");
166 +       }
167 +   })
168 $(".select_assignment").click(
169 function () {
170 var id = $(this).children('i').addBack('i').data('id');
171 diff --git a/assets/js/shj_submissions.js b/assets/js/shj_submissions.js
172 index e92c3f01..ba4d15f5 100644
173 --- a/assets/js/shj_submissions.js
174 +++ b/assets/js/shj_submissions.js
175 @@ -12,6 +12,11 @@ $(document).ready(function () {
176 e.preventDefault();
177 $('code-column').selectText();
178 });
179 +   $(".btn").on('keyup', function (e) {
180 +       if(e.which==13){
181 +           $(this).trigger("click");
182 +       }
183 +   })
184 $(".btn").click(function () {
185 var button = $(this);

```

```

186 var row = button.parents('tr');
187 @@ -62,6 +67,11 @@ $(document).ready(function () {
188
189 });
190 $(".shj_rejudge").attr('title', 'Rejudge');
191 +   $(".shj_rejudge").on('keyup', function (e) {
192 +       if(e.which==13){
193 +           $(this).trigger("click");
194 +       }
195 +   })
196 $(".shj_rejudge").click(function () {
197 var row = $(this).parents('tr');
198 $.ajax({
199 @@ -87,6 +97,11 @@ $(document).ready(function () {
200 }
201 });
202 });
203 +   $(".set_final").on('keyup', function (e) {
204 +       if(e.which==32){
205 +           $(this).trigger("click");
206 +       }
207 +   })
208 $(".set_final").click(
209 function () {
210 var row = $(this).parents('tr');
211 @@ -107,6 +122,8 @@ $(document).ready(function () {
212 error: shj.loading_error,
213 success: function (response) {
214 if (response.done) {
215 +   $(".tr[data-u='" + username + "'][data-p='" + problem + "'] i.set_final").
216 +       attr("aria-checked", false);
217 +   $(".tr[data-u='" + username + "'][data-p='" + problem + "'][data-s='" +
218 +       submit_id + "'] i.set_final").attr("aria-checked", true);
219 $(".tr[data-u='" + username + "'][data-p='" + problem + "'] i.set_final").
220     removeClass('fa-check-circle-o color11').addClass('fa-circle-o');
221 $(".tr[data-u='" + username + "'][data-p='" + problem + "'][data-s='" + submit_id
222 + "'] i.set_final").removeClass('fa-circle-o').addClass('fa-check-circle-o
223     color11');
224 }
225 }

```

Kode A.4: Perubahan untuk mematuhi kriteria 2.1.2

```

1 diff --git a/application/views/pages/admin/add_assignment.twig b/application/
  views/pages/admin/add_assignment.twig
2 index 13a0ddea..5953bd21 100644
3 --- a/application/views/pages/admin/add_assignment.twig
4 +++ b/application/views/pages/admin/add_assignment.twig
5 @@ -16,6 +16,13 @@
6 <script>
7 $(document).ready(function(){
8 tabOverride.set(document.getElementsByTagName('textarea'));
9 +   $switch = false;

```

```

10 +     $("textarea").keyup(function (e) {
11 +         if (e.which==27){
12 +             tabOverride.set(document.getElementsByTagName('textarea'),$switch);
13 +             $switch = !$switch;
14 +         }
15 +     });
16 });
17 </script>
18 <script type="text/javascript" src="{ { base_url('assets/js/jquery-ui-timepicker-
    addon.js') } }"></script>
19 @@ -126,6 +133,7 @@
20 <label for="form_participants">Participants<br>
21 <span class="form_comment">Enter username of participants here (comma separated).
22 Only these users are able to submit. You can use keyword "ALL".</span>
23 +     <span class="form_comment clear">Press "esc" to enable/disable tabindent</
    span>
24 </label>
25 <textarea id="form_participants" name="participants" rows="5" class="sharif_input
    medium">{{ edit ? edit_assignment.participants : set_value('participants', '
    ALL') }}</textarea>
26 </p>
27 @@ -174,6 +182,7 @@
28 <p class="input_p">
29 <label for="form_late_rule">Coefficient rule (<a target="_blank" href="https://
    github.com/ifunpar/Sharif-Judge/blob/docs/v1.4/add_assignment.md#coefficient-
    rule">?</a>)</label><br>
30 <span class="form_comment medium clear" style="display: block;">PHP script
    without &lt;?php ?&gt; tags</span>
31 +     <span class="form_comment clear">Press "esc" to enable/disable tabindent</
    span><br>
32 <textarea id="form_late_rule" name="late_rule" rows="20" class="sharif_input
    add_text">{{ edit ? edit_assignment.late_rule : set_value('late_rule',
    default_late_rule) }}</textarea>
33 {{ form_error('late_rule', '<div class="shj_error">', '</div>')}}
34 </p>
35
36 diff --git a/application/views/pages/admin/add_user.twig b/application/views/
    pages/admin/add_user.twig
37 index 7f7ef43d..a0e91e60 100644
38 --- a/application/views/pages/admin/add_user.twig
39 +++ b/application/views/pages/admin/add_user.twig
40 @@ -22,6 +22,13 @@
41 <script>
42 $(document).ready(function(){
43 tabOverride.set(document.getElementsByTagName('textarea'));
44 +     $switch = false;
45 +     $("textarea").keyup(function (e) {
46 +         if (e.which==27){
47 +             tabOverride.set(document.getElementsByTagName('textarea'),$switch);
48 +             $switch = !$switch;
49 +         }

```



```

50 +   });
51 });
52 </script>
53 <script>
54 @@ -59,6 +66,7 @@
55 <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).
56 </p>
57 <p class="input_p">
58 +   <span class="form_comment clear">Press "esc" to enable/disable tabindent</
    span><br>
59 <textarea name="new_users" id="new_users" rows="20" cols="80" class="sharif_input
    ">
60 # Lines starting with a # sign are comments.
61 # Each line (except comments) represents a user.
62
63 diff --git a/application/views/pages/admin/edit_problem_md.twig b/application/
    views/pages/admin/edit_problem_md.twig
64 index a9bed8fb..a0bc3110 100644
65 --- a/application/views/pages/admin/edit_problem_md.twig
66 +++ b/application/views/pages/admin/edit_problem_md.twig
67 @@ -15,8 +15,15 @@
68 <script type='text/javascript' src="{{ base_url('assets/js/taboverride.min.js')
    }}"></script>
69 <script>
70 $(document).ready(function(){
71 -   tabOverride.set(document.getElementById('md_editor'));
72 -   });
73 +   tabOverride.set(document.getElementById('md_editor'));
74 +   $switch = false;
75 +   $("textarea").keyup(function (e) {
76 +       if (e.which==27){
77 +           tabOverride.set(document.getElementById('md_editor'),$switch);
78 +           $switch = !$switch;
79 +       }
80 +   });
81 +   });
82 </script>
83 {% endblock %}
84
85 @@ -176,6 +183,7 @@ Violets are blue.
86 Problem {{ problem.id }}
87 </p>
88 {{ form_open("problems/edit/md/#{description_assignment.id}/#{problem.id}") }}
89 +   <span class="form_comment clear">Press "esc" to enable/disable tabindent</
    span><br>
90 <p class="input_p">
91 <textarea dir="auto" name="text" rows="30" cols="75" class="sharif_input" id="
    md_editor">{{ problem.description }}</textarea>

```

```

92 </p>
93
94 diff --git a/application/views/pages/admin/settings.twig b/application/views/
    pages/admin/settings.twig
95 index 56bbe8cb..827955cc 100644
96 --- a/application/views/pages/admin/settings.twig
97 +++ b/application/views/pages/admin/settings.twig
98 @@ -15,8 +15,15 @@
99 <script type='text/javascript' src="{ { base_url('assets/js/taboverride.min.js')
    }}"></script>
100 <script>
101 $(document).ready(function(){
102 -     tabOverride.set(document.getElementsByTagName('textarea'));
103 - });
104 +     tabOverride.set(document.getElementsByTagName('textarea'));
105 +     $switch = false;
106 +     $("textarea").keyup(function (e) {
107 +         if (e.which==27){
108 +             tabOverride.set(document.getElementsByTagName('textarea'),$switch);
109 +             $switch = !$switch;
110 +         }
111 +     });
112 + });
113 </script>
114 {% endblock %}
115
116 @@ -122,6 +129,7 @@ $(document).ready(function(){
117 <p class="input_p">
118 <label for="form_late_rule">Default Coefficient Rule</label>
119 <span class="form_comment clear">PHP script without &lt;?php ?&gt; tags</span><br
    >
120 +     <span class="form_comment clear">Press "esc" to enable/disable tabindent</
    span><br>
121 <textarea id="form_late_rule" name="default_late_rule" rows="15" class="
    sharif_input add_text clear">{{ default_late_rule }}</textarea>
122 </p>
123
124 @@ -140,11 +148,13 @@ $(document).ready(function(){
125 <p class="input_p">
126 <label for="form_mail_reset">Password Reset Email</label>
127 <span class="form_comment">You can use {SITE_URL}, {RESET_LINK} and {VALID_TIME
    }</span><br>
128 +     <span class="form_comment clear">Press "esc" to enable/disable tabindent</
    span><br>
129 <textarea id="form_mail_reset" name="reset_password_mail" rows="15" class="
    sharif_input add_text clear">{{ reset_password_mail }}</textarea>
130 </p>
131 <p class="input_p">
132 <label for="form_mail_add">Add User Email</label>
133 <span class="form_comment clear">You can use {SITE_URL}, {LOGIN_URL}, {ROLE}, {
    USERNAME} and {PASSWORD}</span><br>

```

```

134 +   <span class="form_comment clear">Press "esc" to enable/disable tabindent</
      span><br>
135 <textarea id="form_mail_add" name="add_user_mail" rows="15" class="sharif_input
      add_text clear">{{ add_user_mail }}</textarea>
136 </p>
137
138 @@ -203,18 +213,22 @@ $(document).ready(function(){
139 </p>
140 <p class="input_p">
141 <label for="form_def_c">Shield Rules (for C)</label>
142 +   <span class="form_comment clear">Press "esc" to enable/disable tabindent</
      span><br>
143 <textarea id="form_def_c" name="def_c" rows="15" class="sharif_input add_text
      clear">{{ defc }}</textarea>
144 </p>
145 <p class="input_p">
146 <label for="form_def_cpp">Shield Rules (for C++)</label>
147 +   <span class="form_comment clear">Press "esc" to enable/disable tabindent</
      span><br>
148 <textarea id="form_def_cpp" name="def_cpp" rows="15" class="sharif_input add_text
      clear">{{ defcpp }}</textarea>
149 </p>
150 <p class="input_p">
151 <label for="form_shield_py2">Shield (for Python 2)</label>
152 +   <span class="form_comment clear">Press "esc" to enable/disable tabindent</
      span><br>
153 <textarea id="form_shield_py2" name="shield_py2" rows="15" class="sharif_input
      add_text clear">{{ shield_py2 }}</textarea>
154 </p>
155 <p class="input_p">
156 <label for="form_shield_py3">Shield (for Python 3)</label>
157 +   <span class="form_comment clear">Press "esc" to enable/disable tabindent</
      span><br>
158 <textarea id="form_shield_py3" name="shield_py3" rows="15" class="sharif_input
      add_text clear">{{ shield_py3 }}</textarea>
159 </p>
160 <p class="input_p">

```

Kode A.5: Perubahan untuk mematuhi kriteria 2.4.1

```

1 diff --git a/application/views/templates/base.twig b/application/views/templates/
    base.twig
2 index e580f5de..c0b6a8d0 100644
3 --- a/application/views/templates/base.twig
4 +++ b/application/views/templates/base.twig
5 @@ -51,6 +51,8 @@ shj.color_scheme = 'github';
6 {% block other_assets %}{% endblock %}
7
8 <body id="body">
9 +   <a href="#page_title" class="skip">Skip to content</a>
10 +
11 {% include 'templates/top_bar.twig' %}

```

```
12 {% include 'templates/side_bar.twig' %}
13 <div id="main_container" class="scroll-wrapper">
14
15 diff --git a/assets/js/shj_functions.js b/assets/js/shj_functions.js
16 index c7b7140f..9899d077 100644
17 --- a/assets/js/shj_functions.js
18 +++ b/assets/js/shj_functions.js
19 @@ -555,3 +555,26 @@ $(document).ready(function(){
20 $(document).ready(function(){
21 $('input').attr('dir', 'auto');
22 });
23 +
24 + /**
25 +  * Skip links
26 +  */
27 + $(document).ready(function(){
28 + // bind a click event to the 'skip' link
29 + $(".skip").click(function(event){
30 +
31 + // strip the leading hash and declare
32 + // the content we're skipping to
33 + var skipTo="#" + this.href.split('#')[1];
34 +
35 + // Setting 'tabindex' to -1 takes an element out of normal
36 + // tab flow but allows it to be focused via javascript
37 + $(skipTo).attr('tabindex', -1).on('blur focusout', function () {
38 +
39 + // when focus leaves this element,
40 + // remove the tabindex attribute
41 + $(this).removeAttr('tabindex');
42 +
43 + }).focus(); // focus on the content container
44 + });
45 + });
46
47 diff --git a/assets/styles/main.css b/assets/styles/main.css
48 index a0e61502..333bd795 100644
49 --- a/assets/styles/main.css
50 +++ b/assets/styles/main.css
51 @@ -146,6 +146,26 @@ blockquote {
52 margin-left: 7px!important;
53 }
54
55 + .skip {
56 + position: absolute;
57 + top: -1000px;
58 + left: -1000px;
59 + height: 1px;
60 + width: 1px;
61 + text-align: left;
62 + overflow: hidden;
```

```

63 +     }
64 +
65 +     a.skip:active,
66 +     a.skip:focus,
67 +     a.skip:hover {
68 +         left: 0;
69 +         top: 0;
70 +         width: auto;
71 +         height: auto;
72 +         overflow: visible;
73 +     }
74 +
75 /*****
76 div#top_bar {
77 top: 0;

```

Kode A.6: Perubahan untuk mematuhi kriteria 2.4.4

```

1 diff --git a/application/views/templates/side_bar.twig b/application/views/
  templates/side_bar.twig
2 index da069f2e..ce5a0e22 100644
3 --- a/application/views/templates/side_bar.twig
4 +++ b/application/views/templates/side_bar.twig
5 @@ -6,78 +6,78 @@
6 <div id="side_bar" class="sidebar_open">
7 <ul>
8 <li class="color-dashboard{{ selected=='dashboard' ? ' selected' }}">
9 -     <a href="{{ site_url('dashboard') }}">
10 +     <a href="{{ site_url('dashboard') }}" aria-labelledby="dashboard-label">
11 <i class="fa fa-dashboard fa-lg"></i>
12 -     <span class="sidebar_text">Dashboard</span>
13 +     <span class="sidebar_text" id="dashboard-label">Dashboard</span>
14 </a>
15 </li>
16 {% if user.level == 3 %}
17 <li class="color-settings{{ selected=='settings' ? ' selected' }}">
18 -     <a href="{{ site_url('settings') }}">
19 +     <a href="{{ site_url('settings') }}" aria-labelledby="settings-label">
20 <i class="fa fa-gear fa-lg"></i>
21 -     <span class="sidebar_text">Settings</span>
22 +     <span class="sidebar_text" id="settings-label">Settings</span>
23 </a>
24 </li>
25 <li class="color-users{{ selected=='users' ? ' selected' }}">
26 -     <a href="{{ site_url('users') }}">
27 +     <a href="{{ site_url('users') }}" aria-labelledby="users-label">
28 <i class="fa fa-users fa-lg"></i>
29 -     <span class="sidebar_text">Users</span>
30 +     <span class="sidebar_text" id="users-label">Users</span>
31 </a>
32 </li>
33 {% endif %}

```

```

34 <li class="color-notifications{{ selected=='notifications' ? 'selected' }}">
35 -   <a href="{{ site_url('notifications') }}">
36 +   <a href="{{ site_url('notifications') }}" aria-labelledby="notifications-
      label">
37 <i class="fa fa-bell fa-lg"></i>
38 -   <span class="sidebar_text">Notifications</span>
39 +   <span class="sidebar_text" id="notifications-label">Notifications</span>
40 </a>
41 </li>
42 <li class="color-assignments{{ selected=='assignments' ? 'selected' }}">
43 -   <a href="{{ site_url('assignments') }}">
44 +   <a href="{{ site_url('assignments') }}" aria-labelledby="assignments-label">
45 <i class="fa fa-folder-open fa-lg"></i>
46 -   <span class="sidebar_text">Assignments</span>
47 +   <span class="sidebar_text" id="assignments-label">Assignments</span>
48 </a>
49 </li>
50 <li class="color-problems{{ selected=='problems' ? 'selected' }}">
51 -   <a href="{{ site_url('problems') }}">
52 +   <a href="{{ site_url('problems') }}" aria-labelledby="problems-label">
53 <i class="fa fa-puzzle-piece fa-lg"></i>
54 -   <span class="sidebar_text">Problems</span>
55 +   <span class="sidebar_text" id="problems-label">Problems</span>
56 </a>
57 </li>
58 <li class="color-submit{{ selected=='submit' ? 'selected' }}">
59 -   <a href="{{ site_url('submit') }}">
60 +   <a href="{{ site_url('submit') }}" aria-labelledby="submit-label">
61 <i class="fa fa-location-arrow fa-lg"></i>
62 -   <span class="sidebar_text">Submit</span>
63 +   <span class="sidebar_text" id="submit-label">Submit</span>
64 </a>
65 </li>
66 <li class="color-final_submissions{{ selected=='final_submissions' ? 'selected'
      }}">
67 -   <a href="{{ site_url('submissions/final') }}">
68 +   <a href="{{ site_url('submissions/final') }}" aria-labelledby="final-
      submission-label">
69 <i class="fa fa-map-marker fa-lg"></i>
70 -   <span class="sidebar_text">Final Submissions</span>
71 +   <span class="sidebar_text" id="final-submission-label">Final Submissions</
      span>
72 </a>
73 </li>
74 <li class="color-all_submissions{{ selected=='all_submissions' ? 'selected' }}">
75 -   <a href="{{ site_url('submissions/all') }}">
76 +   <a href="{{ site_url('submissions/all') }}" aria-labelledby="all-submission-
      label">
77 <i class="fa fa-bars fa-lg"></i>
78 -   <span class="sidebar_text">All Submissions</span>
79 +   <span class="sidebar_text" id="all-submission-label">All Submissions</span>

```

```

80 </a>
81 </li>
82 <li class="color-scoreboard{{ selected=='scoreboard' ? ' selected' }}">
83 -   <a href="{{ site_url('scoreboard') }}">
84 +   <a href="{{ site_url('scoreboard') }}" aria-labelledby="scoreboard-label">
85 <i class="fa fa-star fa-lg"></i>
86 -   <span class="sidebar_text">Scoreboard</span>
87 +   <span class="sidebar_text" id="scoreboard-label">Scoreboard</span>
88 </a>
89 </li>
90 <li class="color-halloffame{{ selected=='halloffame' ? ' selected' }}">
91 -   <a href="{{ site_url('halloffame') }}">
92 +   <a href="{{ site_url('halloffame') }}" aria-labelledby="hall-of-fame-label">
93 <i class="fa fa-list-alt fa-lg"></i>
94 -   <span class="sidebar_text">Hall of Fame</span>
95 +   <span class="sidebar_text" id="hall-of-fame-label">Hall of Fame</span>
96 </a>
97 </li>
98 {% if user.level == 3 %}
99 <li class="color-logs{{ selected=='logs' ? ' selected' }}">
100 -   <a href="{{ site_url('logs') }}">
101 +   <a href="{{ site_url('logs') }}" aria-labelledby="24-hour-log-label">
102 <i class="fa fa-book fa-lg"></i>
103 -   <span class="sidebar_text">24-hour Log</span>
104 +   <span class="sidebar_text" id="24-hour-log-label">24-hour Log</span>
105 </a>
106 </li>
107 {% endif %}
108
109 diff --git a/application/views/templates/top_bar.twig b/application/views/
    templates/top_bar.twig
110 index ccb67df9..c49c8ab7 100644
111 --- a/application/views/templates/top_bar.twig
112 +++ b/application/views/templates/top_bar.twig
113 @@ -5,7 +5,7 @@
114 #}
115 <div id="top_bar" class="color-{{ selected }}">
116 <div class="top_object shj_menu" id="user_top">
117 -   <a href="{{ site_url('profile') }}" id="profile_link"><i class="fa fa-user
    "></i></a>
118 +   <a href="{{ site_url('profile') }}" id="profile_link" aria-label="Profile"><
    i class="fa fa-user"></i></a>
119 <div class="top_menu user-menu">
120 <div class="gravatar"></div>
121 <div class="name"><i class="fa fa-user"></i> {{ user.username }}</div>

```

Kode A.7: Perubahan untuk mematuhi kriteria 3.1.1

```

1 diff --git a/application/views/templates/base.twig b/application/views/templates/
    base.twig
2 index e580f5de..91b720a3 100644

```

```

3 --- a/application/views/templates/base.twig
4 +++ b/application/views/templates/base.twig
5 @@ -4,7 +4,7 @@
6 # author: Mohammad Javad Naderi <mjnaderi@gmail.com>
7 #}
8 <!DOCTYPE html>
9 -     <html>
10 +     <html lang="en">
11 <head>
12 <title>{% block head_title %}{% endblock %} - SharIF Judge</title>
13 <meta content="text/html" charset="UTF-8">

```

Kode A.8: Perubahan untuk mematuhi kriteria 4.1.1

```

1 diff --git a/application/views/pages/admin/add_assignment.twig b/application/
  views/pages/admin/add_assignment.twig
2 index 13a0ddea..a8bb55fd 100644
3 --- a/application/views/pages/admin/add_assignment.twig
4 +++ b/application/views/pages/admin/add_assignment.twig
5 @@ -119,7 +119,7 @@
6 Extra Time (minutes)<br>
7 <span class="form_comment">Extra time for late submissions.</span>
8 </label>
9 -     <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') }}" />
10 +     <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') }}" />
11 {{ form_error('extra_time', '<div class="shj_error">', '</div>')}}
12 </p>
13 <p class="input_p clear">

```