

BAB 6 PENGUJIAN DAN ANALISIS HASIL

Bab ini akan menjelaskan tentang pengujian dan analisis dari hasil implementasi yang telah dilakukan pada bab sebelumnya. Proses pengujian dimulai dari tahap pengujian kode program yang akan dilakukan pada pengujian unit, kemudian dilanjutkan pengujian integrasi, setelah itu dilanjutkan dengan pengujian sistem atau fungsionalitas yang akan dilakukan pada pengujian validasi serta yang terakhir adalah pengujian menggunakan standar ISO 25010. Analisis hasil bertujuan untuk mendapatkan kesimpulan dari semua hasil pengujian yang telah dilakukan.

6.1 Pengujian

Pengujian yang dilakukan pada Sistem Ujian Harian Siswa ini terdiri dari pengujian unit, pengujian integrasi, pengujian validasi, serta pengujian berbasis ISO 25010. Pengujian unit dan pengujian integrasi menggunakan *white box testing* dengan menggunakan *basis path testing*. Pengujian ini dimulai dengan membuat *flowgraph* berdasarkan algoritme pada setiap fungsi. Setelah didapatkan *flowgraph* maka dilakukan pengujian *cyclomatic complexity* untuk mendapatkan jumlah *path*. Setelah mendapatkan jumlah *path* kemudian membuat jalur *independent path*, dan terakhir membuat *test case* dari *independent path* yang telah dibuat sebelumnya. Pengujian validasi dilakukan dengan menggunakan *black box testing*. Pengujian validasi dilakukan pada seluruh kebutuhan fungsional. Pengujian ISO 25010 dilakukan pada subkarakteristik *functional suitability*, *performance efficiency*, *compatibility*, *usability*, *reliability* dan *security*.

6.1.1 Pengujian Unit

Pengujian unit bertujuan untuk menguji unit-unit yang memiliki prioritas tinggi dalam sistem memperoleh hasil implementasi yang sesuai harapan di awal. Pengujian unit akan dilakukan pada seluruh operasi atau *method* yang terdapat pada seluruh *class*. Pengujian unit menggunakan model *basis path testing* dalam pengujiannya. Dalam pengujian unit ini hanya akan dilakukan pada 3 sampel *method* yaitu pada algoritme *method verifylogin()* pada *class Login*, *method attempt()* pada *class Quiz* dan *method view_result()* pada *class Result*.

1. Pengujian algoritme *method verifylogin()* pada *class Login*

Berikut merupakan algoritme dari *method verifylogin()* pada *class Login*.

Tabel 6.1 Algoritme *method verifylogin()* pada *class Login*

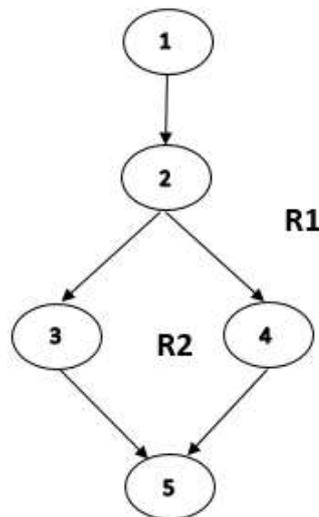
<code>username = POST[email]</code>	}	1
<code>password = POST[password]</code>		
<code>user = model(User_Model)</code>		
<code>IF(pengguna){</code>	→	2
<code>table = user</code>	→	3

Tabel 6.1 Algoritme *method verifylogin()* pada *class Login* (lanjutan)

<pre> where = array (email = username) data = model(User_Model) idUser = data SESSION[setdatauser] = logged_in sebagai user SESSION[setdatauser] = LoginPengguna sebagai username SESSION[setdatauser] = idUser sebagai idUser redirect dashboard </pre>	} 3
<pre> ELSE SESSION[setflashdata] = message sebagai invalidLogin redirect Login </pre>	} 4
<pre> ENDIF → 5 </pre>	

Berikut cara menentukan *Basis Path Testing* pada algoritme *method verifylogin()* pada *class Login*.

1) *Flowgraph*



2) *Cyclomatic Complexity*

- $V(G) = \text{Jumlah Region} = 2$
- $V(G) = E - N + 2 = 5 - 5 + 2 = 2$
- $V(G) = P + 1 = 1 + 1 = 2$

3) *Independent Path*

- Jalur 1 = 1 – 2 – 3 – 5
- Jalur 2 = 1 – 2 – 4 – 5

Berikut merupakan hasil pengujian dari algoritme *method verifylogin()* pada *class Login* berdasarkan jalur *independent path* yang didapat.

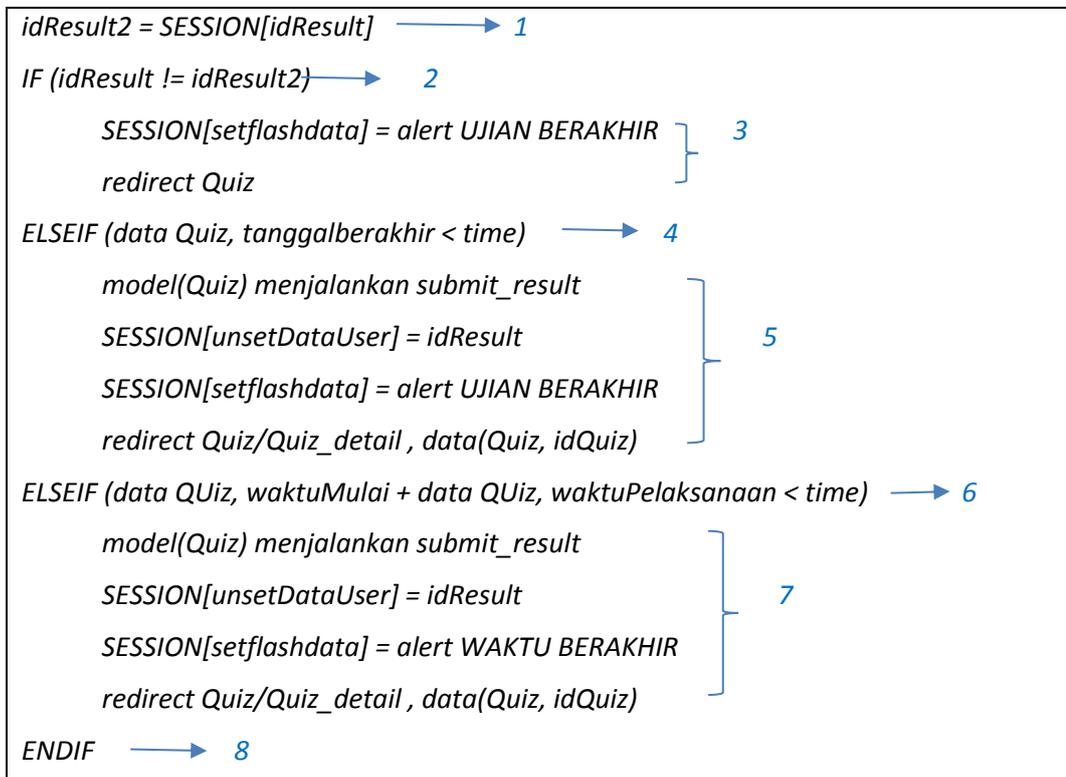
Tabel 6.2 Test case algoritme *method verifylogin()* pada *class Login*

No	No Jalur	Prosedur Uji	Expected Result	Result	Status
1.	1	Memanggil <i>method verifylogin()</i> dengan <i>username</i> "admin@example.com", <i>password</i> "admin" dan status = <i>user</i>	Menampilkan halaman <i>dashboard</i>	Menampilkan halaman <i>dashboard</i>	Valid
2.	2	Memanggil <i>method verifylogin()</i> dengan <i>username</i> "admin@example.com", <i>password</i> "123" dan status != <i>user</i>	Menampilkan pesan <i>invalidLogin</i> , menampilkan halaman <i>login</i>	Menampilkan pesan <i>invalidLogin</i> , menampilkan halaman <i>login</i>	Valid

2. Pengujian algoritme *method attempt()* pada *class Quiz*

Berikut merupakan algoritme dari *method attempt()* pada *class Quiz*.

Tabel 6.3 Algoritme *method attempt()* pada *class Quiz*

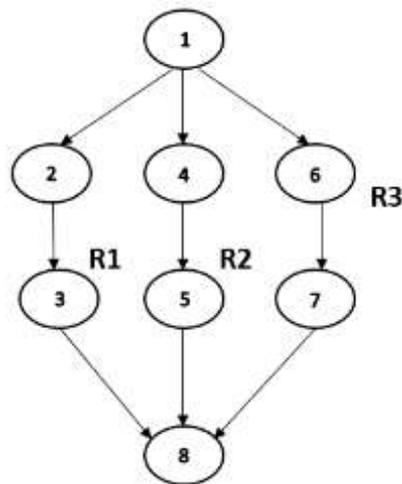


Tabel 6.3 Algoritme *method attempt()* pada *class Quiz* (lanjutan)

<code>data(seconds) = (data Quiz, waktuPelaksanaan*60) - (time - data Quiz, waktuMulai)</code> <code>data(question) = model(Quiz) menjalankan getQuestion</code> <code>data(option) = model(Quiz) memanggil getOption</code> <code>data(title) = data nama Quiz</code> <code>template (nama view, data)</code>	
--	--

Berikut cara menentukan *Basis Path Testing* pada algoritme *method attempt()* pada *class Quiz*.

1) *Flowgraph*



2) *Cyclomatic Complexity*

- $V(G) = \text{Jumlah Region} = 3$
- $V(G) = E - N + 2 = 9 - 8 + 2 = 3$
- $V(G) = P + 1 = 2 + 1 = 3$

3) *Independent Path*

- Jalur 1 = 1 - 2 - 3 - 8
- Jalur 2 = 1 - 4 - 5 - 8
- Jalur 3 = 1 - 6 - 7 - 8

Berikut merupakan hasil pengujian algoritme *method attempt()* pada *class Quiz* berdasarkan jalur *independent path* yang didapat.

Tabel 6.4 Test case algoritme *method attempt()* pada *class Quiz*

No	No Jalur	Prosedur Uji	Expected Result	Result	Status
1.	1	Memanggil <i>method attempt()</i> dengan nilai <i>idResult2 = 1</i> dan <i>idResult != idResult2</i>	Sistem menampilkan pemberitahuan "Anda telah mengerjakan ujian ini", menampilkan halaman ujian	Sistem menampilkan pemberitahuan "Anda telah mengerjakan ujian ini", menampilkan halaman ujian	Valid
2.	2	Memanggil <i>method attempt()</i> dengan nilai <i>idResult2 = 0</i> dan (data <i>Quiz</i> , tanggalberakhir < <i>time</i>)	Sistem menampilkan pemberitahuan "Ujian telah berakhir", menampilkan halaman detail ujian	Sistem menampilkan pemberitahuan "Ujian telah berakhir", menampilkan halaman detail ujian	Valid
3.	3	Memanggil <i>method attempt()</i> dengan nilai <i>idResult2 = 0</i> dan (data <i>Quiz</i> , waktuMulai + data <i>Quiz</i> , waktuPelaksanaan < <i>time</i>)	Sistem menampilkan pesan "waktu berakhir", menampilkan halaman daftar ujian	Sistem menampilkan pesan "waktu berakhir", menampilkan halaman daftar ujian	Valid

3. Pengujian algoritme *method view_result()* pada *class Result*

Berikut merupakan algoritme dari *method view_result()* pada *class Result*.

Tabel 6.5 Algoritme *method view_result()* pada *class Result*

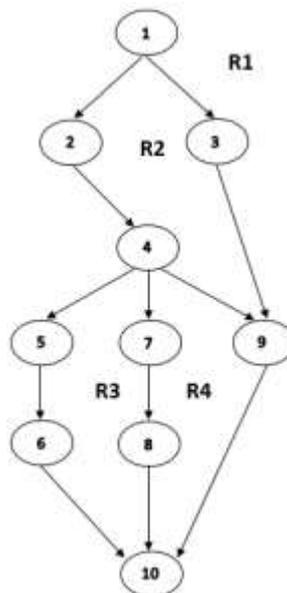
<pre> IF (logged_in sebagai siswa) → 1 data(getGroup) = model(Mapel) menjalankan getGroup → 2 ELSEIF (logged_in sebagai guru) data(getGroupGuru) = model(Mapel) menjalankan getGroupGuru } 3 ENDIF → 4 </pre>

Tabel 6.5 Algoritme *method view_result()* pada *class Result* (lanjutan)

<i>data(result) = model(Result) menjalankan getResult</i>	}	4
<i>data(attempt) = model(Result) menjalankan attempt (data result, idQuiz, data result, idUser)</i>		
<i>IF (data result, view_answers = 1) →</i>		5
<i> model(Quiz)</i>	}	6
<i> data(submit_answers) = model(Quiz) menjalankan submit_answers</i>		
<i> data(question) = model(Quiz) menjalankan getQuestion</i>		
<i> data(answers) = model(Quiz) menjalankan getAnswers</i>		
<i>ELSEIF (SESSION[userdata] = logged_in) →</i>		7
<i> template (nama view, data) →</i>		8
<i>ELSE</i>	}	9
<i> template (nama view, data)</i>		
<i>ENDIF →</i>		10

Berikut cara menentukan *Basis Path Testing* pada algoritme *method view_result()* pada *class Result*.

1) *Flowgraph*



2) *Cyclomatic Complexity*

- $V(G) = \text{Jumlah Region} = 4$
- $V(G) = E - N + 2 = 12 - 10 + 2 = 4$
- $V(G) = P + 1 = 3 + 1 = 4$

3) Independent Path

- Jalur 1 = 1 – 2 – 4 – 5 – 6 – 10
- Jalur 2 = 1 – 2 – 4 – 7 – 8 – 10
- Jalur 3 = 1 – 2 – 4 – 9 – 10
- Jalur 4 = 1 – 3 – 4 – 9 – 10

Berikut merupakan hasil pengujian algoritme *method view_result()* pada *class Result* berdasarkan jalur *independent path* yang didapat.

Tabel 6.6 Test case algoritme *method view_result()* pada *class Result*

No	No Jalur	Prosedur Uji	Expected Result	Result	Status
1.	1	Memanggil <i>method view_result()</i> dengan <i>logged_in</i> sebagai siswa dan <i>view_result = 1</i>	Sistem menampilkan hasil ujian siswa beserta jawaban yang benar dari setiap soal dalam ujian tersebut	Sistem menampilkan hasil ujian siswa beserta jawaban yang benar dari setiap soal dalam ujian tersebut	Valid
2.	2	Memanggil <i>method view_result()</i> dengan <i>logged_in</i> sebagai siswa dan <i>SESSION[userdata] = logged_in</i>	Sistem menampilkan hasil ujian	Sistem menampilkan hasil ujian	Valid
3.	3	Memanggil <i>method view_result()</i> dengan <i>logged_in</i> sebagai siswa dan tidak memenuhi seluruh kondisi sebelumnya	Sistem menampilkan hasil ujian	Sistem menampilkan hasil ujian	Valid
4.	4	Memanggil <i>method view_result()</i> dengan <i>logged_in</i> sebagai guru dan tidak memenuhi seluruh kondisi sebelumnya	Sistem menampilkan hasil ujian siswa yang dipilih oleh guru	Sistem menampilkan hasil ujian siswa yang dipilih oleh guru	Valid

6.1.2 Pengujian Integrasi

Pengujian integrasi bertujuan untuk menguji *class* yang saling berhubungan dengan menggunakan *basis path testing*. Pengujian integrasi akan dilakukan pada *class* Guru dengan *class* *Mapel_model*, *class* *Quiz* dengan *class* *Mapel_model* dan *class* *Result_model*, *class* *Quiz* dengan *class* *Quiz_model*. Pengujian yang pertama dilakukan pada algoritme operasi *insert_ajar()* pada *class* Guru sebagai operasi yang memanggil operasi *insert_ajar()* pada *class* *Mapel_model*. Pengujian yang kedua akan dilakukan pada algoritme operasi *result_quiz()* pada *class* *Quiz* sebagai operasi yang memanggil operasi *get_group_guru()* pada *class* *Mapel_model* dan memanggil operasi *get_result()* pada *class* *Result_model*. Pengujian yang terakhir akan dilakukan pada algoritme operasi *remove_quiz()* pada *class* *Quiz* sebagai operasi yang memanggil operasi *remove_quiz()* pada *class* *Quiz_model*.

1. Pengujian algoritme *method insert_ajar()* pada *class* Guru

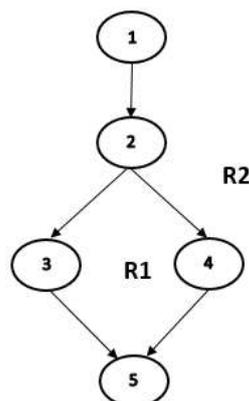
Berikut merupakan algoritme dari *method insert_ajar()* pada *class* Guru.

Tabel 6.7 Algoritme *method insert_ajar()* pada *class* Guru

<i>idUser</i> = POST[<i>idUser</i>]	}	1
<i>idMapel</i> = POST[<i>idMapel</i>]		
<i>idGroup</i> = POST[<i>idGroup</i>]		
IF(<i>Mapel_Model</i> menjalankan <i>insert_ajar</i>){	→	2
<i>SESSION</i> [<i>setFlashdata</i>] = message "data berhasil ditambahkan"	→	3
ELSE	}	4
<i>SESSION</i> [<i>setFlashdata</i>] = message "data error saat ditambahkan"		
redirect halaman gagal tambah mapel yang diajar		
ENDIF	}	5
redirect halaman tambah mapel&kelas		

Berikut cara menentukan *Basis Path Testing* pada algoritme *method insert_ajar()* pada *class* Guru.

1) Flowgraph



2) *Cyclomatic Complexity*

- $V(G) = \text{Jumlah Region} = 2$
- $V(G) = E - N + 2 = 5 - 5 + 2 = 2$
- $V(G) = P + 1 = 1 + 1 = 2$

3) *Independent Path*

- Jalur 1 = 1 – 2 – 3 – 5
- Jalur 2 = 1 – 2 – 4 – 5

Berikut merupakan hasil pengujian algoritme *method insert_ajar()* pada *class Guru* berdasarkan jalur *independent path* yang didapat.

Tabel 6.8 Test case algoritme *method insert_ajar()* pada *class Guru*

No	No Jalur	Prosedur Uji	Expected Result	Result	Status
1.	1	Memanggil <i>method insert_ajar()</i> dengan variabel <i>idUser = 47</i> , <i>idMapel = 2</i> , <i>idGroup = 19</i> , <i>idMapel</i> dan <i>idGroup</i> belum terpakai	Berhasil menambahkan <i>mapel&kelas</i> kedalam pengguna sebagai mata pelajaran dan kelas yang diampunya	Berhasil menambahkan <i>mapel&kelas</i> kedalam pengguna sebagai mata pelajaran dan kelas yang diampunya	Valid
2.	2	Memanggil <i>method insert_ajar()</i> dengan variabel <i>idUser = 45</i> , <i>idMapel = 2</i> , <i>idGroup = 19</i> , <i>idMapel</i> dan <i>idGroup</i> telah terpakai	Gagal menambahkan <i>mapel&kelas</i> kedalam pengguna sebagai mata pelajaran dan kelas yang diampunya	Gagal menambahkan <i>mapel&kelas</i> kedalam pengguna sebagai mata pelajaran dan kelas yang diampunya	Valid

2. Pengujian algoritme *method result_quiz()* pada *class Quiz*

Berikut merupakan algoritme dari *method result_quiz()* pada *class Quiz*.

Tabel 6.9 Algoritme *method result_quiz()* pada *class Quiz*

```

nmr_ajar = GET[nmr_ajar]
data[get_group_guru] = model(PenggunaModel) menjalankan
                        get_group_guru
data[result] = model(Result_Model) menjalankan getResult
load view result_quiz
    
```

Berikut cara menentukan *Basis Path Testing* pada algoritme *method result_quiz()* pada *class Quiz*.

a. *Flowgraph*



b. *Cyclomatic Complexity*

- $V(G) = \text{Jumlah Region} = 1$
- $V(G) = E - N + 2 = 0 - 1 + 2 = 1$
- $V(G) = P + 1 = 0 + 1 = 1$

c. *Independent Path*

- Jalur 1 = 1

Berikut merupakan hasil pengujian algoritme *method result_quiz()* pada *class Quiz* berdasarkan jalur *independent path* yang didapat.

Tabel 6.10 *Test case* algoritme *method result_quiz()* pada *class Quiz*

No	No Jalur	Prosedur Uji	Expected Result	Result	Status
1.	1	Memanggil <i>method result_quiz()</i> , variabel <i>nmr_ajar</i> bernilai = 32	Berhasil menampilkan hasil ujian siswa pada <i>mapel&kelas</i> yang diampu oleh guru	Berhasil menampilkan hasil ujian siswa pada <i>mapel&kelas</i> yang diampu oleh guru	Valid

3. Pengujian algoritme *method remove_quiz()* pada *class Quiz*

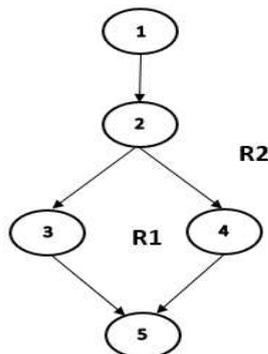
Berikut merupakan algoritme dari *method remove_quiz()* pada *class Quiz*.

Tabel 6.11 Algoritme *method remove_quiz()* pada *class Quiz*

<code>logged_in = SESSION[userdata]</code>	→	1
<code>IF (Quiz_Model menjalankan remove_quiz){</code>	→	2
<code>SESSION[setflashdata] = message "remove successfully"</code>	→	3
<code>ELSE</code>		} 4
<code>SESSION[setflashdata] = message "error to remove"</code>		
<code>ENDIF</code>		} 5
<code>redirect dashboard</code>		

Berikut cara menentukan *Basis Path Testing* pada algoritme *method remove_quiz()* pada *class Quiz*.

1) Flowgraph



2) Cyclomatic Complexity

- $V(G) = \text{Jumlah Region} = 2$
- $V(G) = E - N + 2 = 5 - 5 + 2 = 2$
- $V(G) = P + 1 = 1 + 1 = 2$

3) Independent Path

- Jalur 1 = 1 – 2 – 3 – 5
- Jalur 2 = 1 – 2 – 4 – 5

Berikut merupakan hasil pengujian algoritme *method remove_quiz()* pada *class Quiz* berdasarkan jalur *independent path* yang didapat.

Tabel 6.12 Test case algoritme method *remove_quiz()* pada class *Quiz*

No	No Jalur	Prosedur Uji	Expected Result	Result	Status
1.	1	Memanggil <i>method remove_quiz()</i> dengan status cek session = "logged_in", <i>idQuiz</i> = 1	Berhasil menghapus ujian	Berhasil menghapus ujian	Valid
2.	2	Memanggil <i>method remove_quiz()</i> dengan status cek session = "logged_in", <i>idQuiz</i> = 1	Gagal menghapus ujian	Gagal menghapus ujian	Valid

6.1.3 Pengujian Validasi

Pengujian validasi bertujuan untuk memastikan apakah sistem yang telah dibangun telah memenuhi kebutuhan yang sudah didefinisikan sebelumnya atau belum. Berikut ini adalah kasus uji yang dilakukan oleh pengguna sistem.

1. Kasus Uji *Login*

a. Kasus uji *username* dan *password* benar

Tabel 6.13 Pengujian *login* benar

Nama Kasus Uji	Uji <i>login</i> benar
Objek Uji	Uji tombol <i>login</i>
Tujuan Pengujian	Pengujian dilakukan untuk memastikan bahwa sistem dapat mengidentifikasi pengguna
Prosedur Uji	1.Memasukkan <i>username</i> dan <i>password</i> 2.Menekan tombol <i>login</i>
Hasil yang Diharapkan	Sistem menampilkan halaman dashboard sesuai dengan hak akses pengguna
Hasil Uji Coba	Valid

b. Kasus uji *username* dan/atau *password* salah

Tabel 6.14 Pengujian *login* salah

Nama Kasus Uji	Uji <i>login</i> salah
Objek Uji	Uji tombol <i>login</i>
Tujuan Pengujian	Pengujian dilakukan untuk memastikan bahwa sistem dapat mengidentifikasi pengguna
Prosedur Uji	1.Memasukkan <i>username</i> dan <i>password</i> 2.Menekan tombol <i>login</i>
Hasil yang Diharapkan	Sistem menampilkan tulisan “email atau password Anda salah”
Hasil Uji Coba	Valid

c. Kasus uji *username* dan/atau *password* kosong

Tabel 6.15 Pengujian *login* kosong

Nama Kasus Uji	Uji <i>login</i> kosong
Objek Uji	Uji tombol <i>login</i>
Tujuan Pengujian	Pengujian dilakukan untuk memastikan bahwa sistem dapat mengidentifikasi pengguna
Prosedur Uji	1.Memasukkan <i>username</i> dan <i>password</i> 2.Menekan tombol <i>login</i>
Hasil yang Diharapkan	Sistem menampilkan tulisan “Harap mengisi kolom ini”
Hasil Uji Coba	Valid

2. Kasus Uji Logout

Tabel 6.16 Pengujian *logout*

Nama Kasus Uji	Uji <i>logout</i>
Objek Uji	Uji tombol <i>logout</i>
Tujuan Pengujian	Pengujian dilakukan untuk memastikan bahwa sistem dapat mengeluarkan pengguna dari sistem
Prosedur Uji	Menekan tombol <i>logout</i>
Hasil yang Diharapkan	Sistem dapat mengeluarkan pengguna dari sistem
Hasil Uji Coba	Valid

3. Kasus Uji Melihat Profil Siswa

Tabel 6.17 Pengujian melihat profil siswa

Nama Kasus Uji	Uji Melihat Profil Siswa
Objek Uji	Uji Halaman Profil
Tujuan Pengujian	Pengujian dilakukan untuk memastikan bahwa sistem dapat menampilkan halaman profil
Prosedur Uji	Menekan menu profil
Hasil yang Diharapkan	Sistem dapat menampilkan halaman profil
Hasil Uji Coba	Valid

4. Kasus Uji Melihat Profil Guru

Tabel 6.18 Pengujian melihat profil guru

Nama Kasus Uji	Uji Melihat Profil Guru
Objek Uji	Uji Halaman Profil
Tujuan Pengujian	Pengujian dilakukan untuk memastikan bahwa sistem dapat menampilkan halaman profil
Prosedur Uji	Menekan menu profil
Hasil yang Diharapkan	Sistem dapat menampilkan halaman profil
Hasil Uji Coba	Valid

5. Kasus Uji Mengubah Profil Siswa

Tabel 6.19 Pengujian mengubah profil siswa

Nama Kasus Uji	Uji Mengubah Profil Siswa
Objek Uji	Uji Halaman Profil
Tujuan Pengujian	Pengujian dilakukan untuk memastikan bahwa sistem dapat mengupdate profil siswa
Prosedur Uji	1. Mengubah data siswa 2. Menekan tombol simpan
Hasil yang Diharapkan	Sistem dapat mengupdate profil siswa
Hasil Uji Coba	Valid

6. Kasus Uji Mengubah Profil Guru

Tabel 6.20 Pengujian mengubah profil guru

Nama Kasus Uji	Uji Mengubah Profil Guru
Objek Uji	Uji Halaman Profil
Tujuan Pengujian	Pengujian dilakukan untuk memastikan bahwa sistem dapat mengupdate profil guru
Prosedur Uji	1. Mengubah data guru 2. Menekan tombol simpan
Hasil yang Diharapkan	Sistem dapat mengupdate profil guru
Hasil Uji Coba	Valid

7. Kasus Uji Menambah Admin

- a. Kasus uji menambah admin dengan mengisi seluruh kolom profil

Tabel 6.21 Pengujian menambah admin dengan mengisi seluruh kolom profil

Nama Kasus Uji	Uji Menambah Admin Dengan Mengisi Seluruh Kolom Profil
Objek Uji	Uji Halaman Pengguna bagian Admin
Tujuan Pengujian	Pengujian dilakukan untuk memastikan bahwa sistem dapat menambahkan admin baru kedalam sistem
Prosedur Uji	1. Menekan tombol tambah admin 2. Mengisi data admin baru pada seluruh kolom profil 3. Menekan tombol simpan
Hasil yang Diharapkan	Sistem dapat menambahkan admin baru
Hasil Uji Coba	Valid

- b. Kasus uji menambah admin dengan hanya mengisi kolom nomor induk, email, *password*, nama depan

Tabel 6.22 Pengujian menambah admin dengan mengisi kolom profil nomor induk, email, *password*, nama depan

Nama Kasus Uji	Uji Menambah Admin Dengan Hanya Mengisi Kolom Nomor Induk, Email, <i>Password</i> , Nama Depan
Objek Uji	Uji Halaman Pengguna bagian Admin
Tujuan Pengujian	Pengujian dilakukan untuk memastikan bahwa sistem dapat menambahkan admin baru kedalam sistem
Prosedur Uji	1. Menekan tombol tambah admin

Tabel 6.23 Pengujian menambah admin dengan mengisi kolom profil nomor induk, email, *password*, nama depan (lanjutan)

	2.Mengisi data admin baru pada kolom nomor induk, email, <i>password</i> , nama depan 3.Menekan tombol simpan
Hasil yang Diharapkan	Sistem dapat menambahkan admin baru
Hasil Uji Coba	Valid

8. Kasus Uji Menambah Guru

- a. Kasus uji menambah guru dengan mengisi seluruh kolom profil

Tabel 6.24 Pengujian menambah guru dengan mengisi seluruh kolom profil

Nama Kasus Uji	Uji Menambah Guru Dengan Mengisi Seluruh Kolom Profil
Objek Uji	Uji Halaman Pengguna bagian Guru
Tujuan Pengujian	Pengujian dilakukan untuk memastikan bahwa sistem dapat menambahkan guru baru kedalam sistem
Prosedur Uji	1. Menekan tombol tambah guru 2. Mengisi data guru baru pada seluruh kolom profil 3. Menekan tombol simpan
Hasil yang Diharapkan	Sistem dapat menambahkan guru baru
Hasil Uji Coba	Valid

- b. Kasus uji menambah guru dengan hanya mengisi kolom nomor induk, email, *password*, nama depan

Tabel 6.25 Pengujian menambah guru dengan mengisi kolom nomor induk, email, *password*, nama depan

Nama Kasus Uji	Uji Menambah Guru Dengan Hanya Mengisi Kolom Nomor Induk, Email, <i>Password</i> , Nama Depan
Objek Uji	Uji Halaman Pengguna bagian Guru
Tujuan Pengujian	Pengujian dilakukan untuk memastikan bahwa sistem dapat menambahkan guru baru kedalam sistem
Prosedur Uji	a. Menekan tombol tambah guru b. Mengisi data guru baru pada kolom nomor induk, email, <i>password</i> , nama depan c. Menekan tombol simpan

Tabel 6.26 Pengujian menambah guru dengan mengisi kolom nomor induk, email, *password*, nama depan (lanjutan)

Hasil yang Diharapkan	Sistem dapat menambahkan guru baru
Hasil Uji Coba	Valid

9. Kasus Uji Menambah Siswa

a. Kasus uji menambah siswa dengan mengisi seluruh kolom profil

Tabel 6.27 Pengujian menambah siswa dengan mengisi seluruh kolom profil

Nama Kasus Uji	Uji Menambah Siswa Dengan Mengisi Seluruh Kolom Profil
Objek Uji	Uji Halaman Pengguna bagian Siswa
Tujuan Pengujian	Pengujian dilakukan untuk memastikan bahwa sistem dapat menambahkan siswa baru kedalam sistem
Prosedur Uji	1. Menekan tombol tambah siswa 2. Mengisi data siswa baru pada seluruh kolom profil 3. Menekan tombol simpan
Hasil yang Diharapkan	Sistem dapat menambahkan siswa baru
Hasil Uji Coba	Valid

b. Kasus uji menambah siswa dengan hanya mengisi kolom nomor induk, email, *password*, nama depan, kelas

Tabel 6.28 Pengujian menambah siswa dengan mengisi kolom nomor induk, email, *password*, nama depan, kelas

Nama Kasus Uji	Uji Menambah Siswa Dengan Hanya Mengisi Kolom Nomor Induk, Email, <i>Password</i> , Nama Depan, Kelas
Objek Uji	Uji Halaman Pengguna bagian Siswa
Tujuan Pengujian	Pengujian dilakukan untuk memastikan bahwa sistem dapat menambahkan siswa baru kedalam sistem
Prosedur Uji	a. Menekan tombol tambah siswa b. Mengisi data siswa baru pada kolom nomor induk, email, <i>password</i> , nama depan, kelas c. Menekan tombol simpan
Hasil yang Diharapkan	Sistem dapat menambahkan siswa baru
Hasil Uji Coba	Valid

10. Kasus Uji Mengubah Admin

Tabel 6.29 Pengujian mengubah admin

Nama Kasus Uji	Uji Mengubah Admin
Objek Uji	Uji Halaman Pengguna bagian Admin
Tujuan Pengujian	Pengujian dilakukan untuk memastikan bahwa sistem dapat memperbarui data profil admin
Prosedur Uji	1. Menekan tombol ubah data 2. Mengubah data profil admin 3. Menekan tombol simpan
Hasil yang Diharapkan	Sistem dapat memperbarui data profil admin
Hasil Uji Coba	Valid

11. Kasus Uji Mengubah Guru

Tabel 6.30 Pengujian mengubah guru

Nama Kasus Uji	Uji Mengubah Guru
Objek Uji	Uji Halaman Pengguna bagian Guru
Tujuan Pengujian	Pengujian dilakukan untuk memastikan bahwa sistem dapat memperbarui data profil guru
Prosedur Uji	1. Menekan tombol ubah data 2. Mengubah data profil guru 3. Menekan tombol simpan
Hasil yang Diharapkan	Sistem dapat memperbarui data profil guru
Hasil Uji Coba	Valid

12. Kasus Uji Mengubah Siswa

Tabel 6.31 Pengujian mengubah siswa

Nama Kasus Uji	Uji Mengubah Siswa
Objek Uji	Uji Halaman Pengguna bagian Siswa
Tujuan Pengujian	Pengujian dilakukan untuk memastikan bahwa sistem dapat memperbarui data profil siswa
Prosedur Uji	1. Menekan tombol ubah data 2. Mengubah data profil siswa 3. Menekan tombol simpan

Tabel 6.29 Pengujian mengubah siswa (lanjutan)

Hasil yang Diharapkan	Sistem dapat memperbarui data profil siswa
Hasil Uji Coba	Valid

13. Kasus Uji Menghapus Admin

Tabel 6.32 Pengujian menghapus admin

Nama Kasus Uji	Uji Menghapus Admin
Objek Uji	Uji Halaman Pengguna bagian Admin
Tujuan Pengujian	Pengujian dilakukan untuk memastikan bahwa sistem dapat menghapus admin
Prosedur Uji	Menekan tombol hapus
Hasil yang Diharapkan	Sistem dapat menghapus admin
Hasil Uji Coba	Valid

14. Kasus Uji Menghapus Guru

Tabel 6.33 Pengujian menghapus guru

Nama Kasus Uji	Uji Menghapus Guru
Objek Uji	Uji Halaman Pengguna bagian Guru
Tujuan Pengujian	Pengujian dilakukan untuk memastikan bahwa sistem dapat menghapus guru
Prosedur Uji	Menekan tombol hapus
Hasil yang Diharapkan	Sistem dapat menghapus guru
Hasil Uji Coba	Valid

15. Kasus Uji Menghapus Siswa

Tabel 6.34 Pengujian menghapus siswa

Nama Kasus Uji	Uji Menghapus Guru
Objek Uji	Uji Halaman Pengguna bagian Guru
Tujuan Pengujian	Pengujian dilakukan untuk memastikan bahwa sistem dapat menghapus guru
Prosedur Uji	Menekan tombol hapus

Tabel 6.35 Pengujian menghapus siswa (lanjutan)

Hasil yang Diharapkan	Sistem dapat menghapus guru
Hasil Uji Coba	Valid

16. Kasus Uji Menambah Mata Pelajaran

Tabel 6.36 Pengujian menambah mata pelajaran

Nama Kasus Uji	Uji Menambah Mata Pelajaran
Objek Uji	Uji Halaman Menu Daftar Mapel
Tujuan Pengujian	Pengujian dilakukan untuk memastikan bahwa sistem dapat menambah mata pelajaran baru kedalam sistem
Prosedur Uji	1. Memasukkan nama mata pelajaran baru ke kolom nama mata pelajaran 2. Menekan tombol tambah mata pelajaran
Hasil yang Diharapkan	Sistem dapat menyimpan mata pelajaran baru kedalam sistem
Hasil Uji Coba	Valid

17. Kasus Uji Mengubah Mata Pelajaran

Tabel 6.37 Pengujian mengubah mata pelajaran

Nama Kasus Uji	Uji Mengubah Mata Pelajaran
Objek Uji	Uji Halaman Menu Daftar Mapel
Tujuan Pengujian	Pengujian dilakukan untuk memastikan bahwa sistem dapat memperbarui nama mata pelajaran
Prosedur Uji	1. Mengubah nama mata pelajaran 2. Menekan enter pada keyboard
Hasil yang Diharapkan	Sistem dapat memperbarui nama mata pelajaran
Hasil Uji Coba	Valid

18. Kasus Uji Menghapus Mata Pelajaran

- a. Kasus uji berhasil menghapus mata pelajaran

Tabel 6.38 Pengujian berhasil menghapus mata pelajaran

Nama Kasus Uji	Uji Berhasil Menghapus Mata Pelajaran
Objek Uji	Uji Halaman Menu Daftar Mapel
Tujuan Pengujian	Pengujian dilakukan untuk memastikan bahwa sistem dapat menghapus mata pelajaran
Prosedur Uji	1. Menekan tombol silang 2. Menekan tombol OK pada halaman peringatan
Hasil yang Diharapkan	Sistem dapat menghapus nama pelajaran
Hasil Uji Coba	Valid

b. Kasus uji batal menghapus mata pelajaran

Tabel 6.39 Pengujian batal menghapus mata pelajaran

Nama Kasus Uji	Uji Batal Menghapus Mata Pelajaran
Objek Uji	Uji Halaman Menu Daftar Mapel
Tujuan Pengujian	Pengujian dilakukan untuk memastikan bahwa sistem batal menghapus mata pelajaran
Prosedur Uji	a. Menekan tombol silang b. Menekan tombol Cancel pada halaman peringatan
Hasil yang Diharapkan	Sistem batal menghapus nama pelajaran
Hasil Uji Coba	Valid

19. Kasus Uji Menambah Kelas

Tabel 6.40 Pengujian menambah kelas

Nama Kasus Uji	Uji Menambah Kelas
Objek Uji	Uji Halaman Menu Kelas
Tujuan Pengujian	Pengujian dilakukan untuk memastikan bahwa sistem dapat memenuhi kebutuhan fungsional untuk menambah kelas baru kedalam sistem
Prosedur Uji	1. Memasukkan nama kelas baru ke kolom nama kelas 2. Menekan tombol tambah kelas

Tabel 6.41 Pengujian menambah kelas (lanjutan)

Hasil yang Diharapkan	Sistem dapat menyimpan kelas baru kedalam sistem
Hasil Uji Coba	Valid

20. Kasus Uji Mengubah Kelas

Tabel 6.42 Pengujian mengubah kelas

Nama Kasus Uji	Uji Mengubah Kelas
Objek Uji	Uji Halaman Menu Kelas
Tujuan Pengujian	Pengujian dilakukan untuk memastikan bahwa sistem dapat memperbarui nama kelas
Prosedur Uji	1. Mengubah nama kelas 2. Menekan tombol enter
Hasil yang Diharapkan	Sistem dapat memperbarui nama kelas
Hasil Uji Coba	Valid

21. Kasus Uji Menghapus Kelas

a. Kasus uji berhasil menghapus kelas

Tabel 6.43 Pengujian berhasil menghapus kelas

Nama Kasus Uji	Uji Berhasil Menghapus Kelas
Objek Uji	Uji Halaman Menu Kelas
Tujuan Pengujian	Pengujian dilakukan untuk memastikan bahwa sistem dapat menghapus kelas
Prosedur Uji	1. Menekan tombol silang 2. Menekan tombol OK pada pesan peringatan
Hasil yang Diharapkan	Sistem dapat menghapus kelas
Hasil Uji Coba	Valid

b. Kasus uji batal menghapus kelas

Tabel 6.44 Pengujian batal menghapus kelas

Nama Kasus Uji	Uji Batal Menghapus Kelas
Objek Uji	Uji Halaman Menu Kelas

Tabel 6.45 Pengujian batal menghapus kelas (lanjutan)

Tujuan Pengujian	Pengujian dilakukan untuk memastikan bahwa sistem dapat membatalkan menghapus kelas
Prosedur Uji	1. Menekan tombol silang 2. Menekan tombol Cancel pada pesan peringatan
Hasil yang Diharapkan	Sistem batal menghapus kelas
Hasil Uji Coba	Valid

22. Kasus Uji Memasukkan Guru Ke Mata Pelajaran dan Kelas

a. Kasus uji berhasil memasukkan guru ke mata pelajaran dan kelas

Tabel 6.46 Pengujian memasukkan guru ke mata pelajaran dan kelas

Nama Kasus Uji	Uji Berhasil Memasukkan Guru Ke Mata Pelajaran dan Kelas
Objek Uji	Uji Halaman Tambah Mata Pelajaran dan Kelas
Tujuan Pengujian	Pengujian dilakukan untuk memastikan bahwa sistem dapat menambahkan mata pelajaran dan kelas yang diampu kedalam mata pelajaran dan kelas milik guru
Prosedur Uji	1. Memilih mata pelajaran dan kelas yang ada dalam <i>drop list</i> pilihan 2. Menekan tombol simpan
Hasil yang Diharapkan	Sistem dapat menyimpan mata pelajaran dan kelas yang diampu guru kedalam sistem
Hasil Uji Coba	Valid

b. Kasus uji gagal memasukkan guru ke mata pelajaran dan kelas

Tabel 6.47 Pengujian gagal memasukkan guru ke mata pelajaran dan kelas

Nama Kasus Uji	Uji Gagal Memasukkan Guru Ke Mata Pelajaran dan Kelas
Objek Uji	Uji Halaman Tambah Mata Pelajaran dan Kelas
Tujuan Pengujian	Pengujian dilakukan untuk memastikan bahwa sistem dapat menggagalkan menambahkan mata pelajaran dan kelas yang diampu kedalam mata pelajaran dan kelas milik guru karena telah terpakai
Prosedur Uji	1. Memilih mata pelajaran dan kelas yang ada dalam <i>drop list</i> pilihan 2. Menekan tombol simpan

Tabel 6.48 Pengujian gagal memasukkan guru ke mata pelajaran dan kelas (lanjutan)

Hasil yang Diharapkan	Sistem menampilkan halaman yang menerangkan bahwa mata pelajaran dan kelas telah terpakai
Hasil Uji Coba	Valid

23. Kasus Uji Menambah Soal dan Jawaban

Tabel 6.49 Pengujian menambah soal dan jawaban

Nama Kasus Uji	Uji Menambah Soal dan Jawaban
Objek Uji	Uji Halaman Tambah Soal
Tujuan Pengujian	Pengujian dilakukan untuk memastikan bahwa sistem dapat menambahkan soal dan jawaban baru kedalam sistem
Prosedur Uji	1. Memasukkan pilihan mata pelajaran, pilihan level, soal, deskripsi soal, pilihan jawaban dan pilihan jawaban yang benar 2. Menekan tombol simpan
Hasil yang Diharapkan	Sistem dapat menyimpan soal dan jawaban baru kedalam bank soal sistem
Hasil Uji Coba	Valid

24. Kasus Uji Mengubah Soal dan Jawaban

Tabel 6.50 Pengujian mengubah soal dan jawaban

Nama Kasus Uji	Uji Mengubah Soal dan Jawaban
Objek Uji	Uji Halaman Edit Soal
Tujuan Pengujian	Pengujian dilakukan untuk memastikan bahwa sistem dapat memperbarui soal dan jawaban
Prosedur Uji	1. Memasukkan pilihan mata pelajaran, pilihan level, soal, deskripsi soal, pilihan jawaban dan pilihan jawaban yang benar 2. Menekan tombol simpan
Hasil yang Diharapkan	Sistem dapat memperbarui soal
Hasil Uji Coba	Valid

25. Kasus Uji Menghapus Soal dan Jawaban

- a. Kasus uji berhasil menghapus soal dan jawaban

Tabel 6.51 Kasus uji berhasil menghapus soal dan jawaban

Nama Kasus Uji	Uji Berhasil Menghapus Soal dan Jawaban
Objek Uji	Uji Halaman Daftar Soal
Tujuan Pengujian	Pengujian dilakukan untuk memastikan bahwa sistem dapat menghapus soal dan jawaban
Prosedur Uji	1. Menekan tombol silang pada soal yang diinginkan 2. Menekan tombol OK pada pesan peringatan
Hasil yang Diharapkan	Sistem dapat menghapus soal
Hasil Uji Coba	Valid

b. Kasus uji batal menghapus soal dan jawaban

Tabel 6.52 Kasus uji batal menghapus soal dan jawaban

Nama Kasus Uji	Uji Batal Menghapus Soal dan Jawaban
Objek Uji	Uji Halaman Daftar Soal
Tujuan Pengujian	Pengujian dilakukan untuk memastikan bahwa sistem dapat membatalkan menghapus soal dan jawaban
Prosedur Uji	a. Menekan tombol silang pada soal yang diinginkan b. Menekan tombol Cancel pada pesan peringatan
Hasil yang Diharapkan	Sistem dapat membatalkan hapus soal
Hasil Uji Coba	Valid

26. Kasus Uji Melihat Mata Pelajaran dan Kelas

Tabel 6.53 Pengujian melihat mata pelajaran dan kelas

Nama Kasus Uji	Uji Melihat Mata Pelajaran dan Kelas
Objek Uji	Uji Menu Mapel&Kelas
Tujuan Pengujian	Pengujian dilakukan untuk memastikan bahwa sistem dapat menampilkan mata pelajaran dan kelas yang diampu oleh guru
Prosedur Uji	Menekan menu Mapel&Kelas
Hasil yang Diharapkan	Sistem dapat menampilkan mata pelajaran dan kelas yang diampu oleh guru
Hasil Uji Coba	Valid

27. Kasus Uji Membuat Ujian Harian

Tabel 6.54 Pengujian membuat ujian harian

Nama Kasus Uji	Uji Membuat Ujian Harian
Objek Uji	Uji Halaman Tambah Ujian
Tujuan Pengujian	Pengujian dilakukan untuk memastikan bahwa sistem dapat menambahkan ujian baru kedalam sistem
Prosedur Uji	a. Memasukkan nama ujian, pilihan mata pelajaran, deskripsi ujian, waktu mulai ujian, waktu selesai ujian, waktu mengerjakan ujian, mengerjakan ujian berapa kali, nilai minimal untuk bisa lulus, nilai benar, nilai salah, ijinan melihat jawaban saat lihat hasil, pilih kelas, pilih soal b. Menekan tombol simpan
Hasil yang Diharapkan	Sistem dapat menyimpan ujian baru kedalam sistem
Hasil Uji Coba	Valid

28. Kasus Uji Mengubah Ujian Harian

Tabel 6.55 Pengujian mengubah ujian harian

Nama Kasus Uji	Uji Mengubah Ujian Harian
Objek Uji	Uji Halaman Edit Ujian
Tujuan Pengujian	Pengujian dilakukan untuk memastikan bahwa sistem dapat memperbarui ujian
Prosedur Uji	1. Mengubah nama ujian, pilihan mata pelajaran, deskripsi ujian, waktu mulai ujian, waktu selesai ujian, waktu mengerjakan ujian, mengerjakan ujian berapa kali, nilai minimal untuk bisa lulus, nilai benar, nilai salah, ijinan melihat jawaban saat lihat hasil, pilih kelas, pilih soal 2. Menekan tombol simpan
Hasil yang Diharapkan	Sistem dapat memperbarui ujian
Hasil Uji Coba	Valid

29. Kasus Uji Menghapus Ujian Harian

- a. Kasus uji berhasil menghapus ujian harian

Tabel 6.56 Pengujian berhasil menghapus ujian harian

Nama Kasus Uji	Uji Berhasil Menghapus Ujian Harian
Objek Uji	Uji Halaman Daftar Ujian
Tujuan Pengujian	Pengujian dilakukan untuk memastikan bahwa sistem dapat menghapus ujian
Prosedur Uji	1. Menekan tombol silang pada ujian yang diinginkan 2. Menekan tombol OK pada pesan peringatan
Hasil yang Diharapkan	Sistem dapat menghapus ujian
Hasil Uji Coba	Valid

b. Kasus uji batal menghapus ujian harian

Tabel 6.57 Pengujian batal menghapus ujian harian

Nama Kasus Uji	Uji Batal Menghapus Ujian Harian
Objek Uji	Uji Halaman Daftar Ujian
Tujuan Pengujian	Pengujian dilakukan untuk memastikan bahwa sistem dapat membatalkan hapus ujian
Prosedur Uji	1. Menekan tombol silang pada ujian yang diinginkan 2. Menekan tombol Cancel pada pesan peringatan
Hasil yang Diharapkan	Sistem dapat membatalkan hapus ujian
Hasil Uji Coba	Valid

30. Kasus Uji Melihat Hasil Ujian Siswa

Tabel 6.58 Pengujian melihat hasil ujian siswa

Nama Kasus Uji	Uji Melihat Hasil Ujian Siswa
Objek Uji	Uji Halaman Daftar Nilai Ujian
Tujuan Pengujian	Pengujian dilakukan untuk memastikan bahwa sistem dapat menampilkan hasil ujian siswa
Prosedur Uji	Menekan tombol Lihat pada halaman Daftar Nilai Ujian
Hasil yang Diharapkan	Sistem dapat menampilkan hasil ujian siswa
Hasil Uji Coba	Valid

31. Kasus Uji Melihat Rekap Hasil Ujian Siswa

Tabel 6.59 Pengujian melihat rekap hasil ujian siswa

Nama Kasus Uji	Uji Melihat Rekap Hasil Ujian Siswa
Objek Uji	Uji Halaman Daftar Rekap Nilai Siswa
Tujuan Pengujian	Pengujian dilakukan untuk memastikan bahwa sistem dapat menampilkan rekap hasil ujian siswa
Prosedur Uji	1. Menekan menu Mapel&Kelas 2. Menekan Mapel&Kelas yang diinginkan
Hasil yang Diharapkan	Sistem dapat menampilkan rekap hasil ujian siswa pada Mapel&Kelas yang dituju
Hasil Uji Coba	Valid

32. Kasus Uji Melihat Daftar Mata Pelajaran

Tabel 6.60 Pengujian melihat daftar mata pelajaran

Nama Kasus Uji	Uji Melihat Daftar Mata Pelajaran
Objek Uji	Uji Menu Mata Pelajaran
Tujuan Pengujian	Pengujian dilakukan untuk memastikan bahwa sistem dapat menampilkan mata pelajaran yang terdapat ujian
Prosedur Uji	Menekan menu mata pelajaran
Hasil yang Diharapkan	Sistem dapat menampilkan mata pelajaran
Hasil Uji Coba	Valid

33. Kasus Uji Melihat Daftar Ujian

Tabel 6.61 Pengujian melihat daftar ujian

Nama Kasus Uji	Uji Melihat Daftar Ujian
Objek Uji	Uji Daftar Ujian
Tujuan Pengujian	Pengujian dilakukan untuk memastikan bahwa sistem dapat menampilkan daftar ujian pada mata pelajaran yang ditempuh oleh siswa
Prosedur Uji	1. Menekan menu Mata Pelajaran 2. Menekan submenu mata pelajaran yang diinginkan

Tabel 6.62 Pengujian melihat daftar ujian (lanjutan)

Hasil yang Diharapkan	Sistem dapat menampilkan daftar ujian pada mata pelajaran yang ditempuh
Hasil Uji Coba	Valid

34. Kasus Uji Mengerjakan Ujian

a. Kasus uji selesai mengerjakan ujian

Tabel 6.63 Pengujian selesai mengerjakan ujian

Nama Kasus Uji	Uji Selesai Mengerjakan Ujian
Objek Uji	Uji Halaman Mengerjakan Ujian
Tujuan Pengujian	Pengujian dilakukan untuk memastikan bahwa sistem dapat menampilkan halaman soal ujian dan menyimpan hasil ujian
Prosedur Uji	1. Siswa mengerjakan soal ujian 2. Menekan tombol selesai 3. Siswa menekan tombol selesai pada halaman peringatan
Hasil yang Diharapkan	Sistem dapat menyimpan hasil ujian siswa
Hasil Uji Coba	Valid

b. Kasus uji belum selesai mengerjakan ujian

Tabel 6.64 Pengujian belum selesai mengerjakan ujian

Nama Kasus Uji	Uji Belum Mengerjakan Ujian
Objek Uji	Uji Halaman Mengerjakan Ujian
Tujuan Pengujian	Pengujian dilakukan untuk memastikan bahwa sistem dapat menampilkan halaman soal ujian dan kembali ke halaman soal ujian saat siswa menekan tombol belum selesai mengerjakan ujian
Prosedur Uji	a. Siswa mengerjakan soal ujian b. Menekan tombol selesai c. Siswa menekan tombol belum pada halaman peringatan
Hasil yang Diharapkan	Sistem dapat menampilkan kembali halaman ujian
Hasil Uji Coba	Valid

35. Kasus Uji Melihat Hasil Ujian

Tabel 6.65 Pengujian melihat hasil ujian

Nama Kasus Uji	Uji Melihat Hasil Ujian
Objek Uji	Uji Halaman Daftar Ujian
Tujuan Pengujian	Pengujian dilakukan untuk memastikan bahwa sistem dapat menampilkan hasil ujian siswa
Prosedur Uji	Menekan tombol <i>click here</i>
Hasil yang Diharapkan	Sistem dapat menampilkan hasil ujian siswa
Hasil Uji Coba	Valid

36. Kasus Uji Melihat Rekap Hasil Ujian

Tabel 6.66 Pengujian melihat rekap hasil ujian

Nama Kasus Uji	Uji Melihat Rekap Hasil Ujian
Objek Uji	Uji Halaman Rekap Hasil Ujian
Tujuan Pengujian	Pengujian dilakukan untuk memastikan bahwa sistem dapat menampilkan rekap hasil ujian siswa
Prosedur Uji	Menekan menu rekap nilai
Hasil yang Diharapkan	Sistem dapat menampilkan rekap hasil ujian siswa
Hasil Uji Coba	Valid

6.1.4 Pengujian Kualitas ISO 25010

1. *Functional Suitability*

Pengujian ini bertujuan untuk memastikan kebutuhan fungsional yang telah didefinisikan sebelumnya telah diimplementasikan sesuai yang diharapkan. *Functional suitability* memiliki 3 subkarakteristik yaitu *functional completeness*, *functional correctness*, dan *functional appropriateness*. Seluruh subkarakteristik ini dapat diuji berdasarkan kebutuhan fungsional telah didefinisikan sebelumnya. Pengujian ini dapat mengacu pada pengujian validasi karena pengujian validasi juga menguji kebutuhan fungsional sistem. Pertama kebutuhan fungsional sistem telah diuji dan hasilnya seluruh kebutuhan fungsional sistem telah terimplementasi dengan baik. Kemudian dilakukan perhitungan persentase keberhasilan sebagai berikut :

$$\text{Persentase keberhasilan} = \frac{i}{r} \times 100\%$$

$$\text{Persentase keberhasilan} = \frac{36}{36} \times 100\%$$

$$\text{Persentase keberhasilan} = 100\%$$

Dari hasil persentase keberhasilan didapatkan bahwa sistem sangat baik dan memenuhi karakteristik *functional suitability*.

2. Performance Efficiency

a. Time Behavior

Pengujian pada subkarakteristik ini dilakukan dengan menggunakan *tool* WAPT (*Web Application Load, Stress and Performance Testing*). Beban sistem yang diujikan sebesar 20 *virtual user* dan setiap *user* melakukan satu *action* setiap 30 detik. Jika kriteria yang diuji telah didapatkan nilainya maka akan dibandingkan dengan standar waktu respon Jakob Nielsen. Jika sistem memiliki waktu respon kurang dari 10 detik maka sistem memenuhi subkarakteristik *time behaviour*. Namun jika sistem memiliki waktu respon lebih dari sama dengan 10 detik, maka proses atau *interface* pada sistem harus disederhanakan. Berikut Tabel 6.60 yang menjelaskan hasil pengujian *time behaviour*.

Tabel 6.67 Hasil pengujian *time behaviour*

URL	Rata-Rata Hasil (Detik)
http://local.ujianharian.com/	0,13
http://local.ujianharian.com/dashboard	0,18
http://local.ujianharian.com/admin/view_user	0,18
http://local.ujianharian.com/admin/tambah_user	0,12
http://local.ujianharian.com/admin/insert_user	0,36
http://local.ujianharian.com/user/edit_user/39	0,22
http://local.ujianharian.com/user/update_user/39	0,92
http://local.ujianharian.com/guru/add_category_group/39	0,24
http://local.ujianharian.com/guru/insert_ajar	0,36
http://local.ujianharian.com/admin/category_list	0,10
http://local.ujianharian.com/qbank/insert_category	0,85
http://local.ujianharian.com/qbank/update_category/10	0,11
http://local.ujianharian.com/qbank/remove_category/10	0,21
http://local.ujianharian.com/admin/group_list	0,15

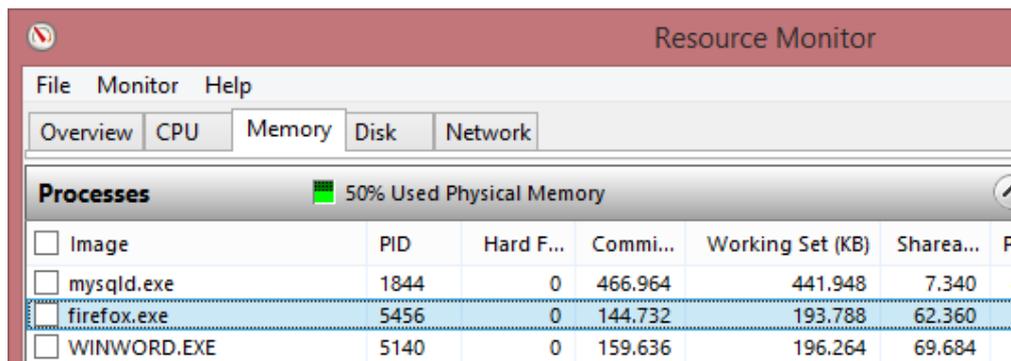
Tabel 6.60 Hasil pengujian time behavior (lanjutan)

http://local.ujianharian.com/admin/insert_group	0,73
http://local.ujianharian.com/admin/update_group/12	0,22
http://local.ujianharian.com/admin/remove_group/12	0,20
http://local.ujianharian.com/user/logout	0,16
http://local.ujianharian.com/quiz/view_quiz/23/2	0,15
http://local.ujianharian.com/quiz/result_quiz/2	0,15
http://local.ujianharian.com/result/view_result/1	0,20
http://local.ujianharian.com/quiz/download_rekap/23/2	0,09
http://local.ujianharian.com/quiz/add_quiz	0,16
http://local.ujianharian.com/quiz/insert_quiz	0,58
http://local.ujianharian.com/quiz/add_question/6	0,25
http://local.ujianharian.com/quiz	0,17
http://local.ujianharian.com/quiz/edit_quiz/6	0,31
http://local.ujianharian.com/quiz/update_quiz/6	0,78
http://local.ujianharian.com/qbank/new_question	0,15
http://local.ujianharian.com/qbank/edit_question/59	0,10
http://local.ujianharian.com/qbank	0,12
http://local.ujianharian.com/qbank/remove_question/59	0,28
http://local.ujianharian.com/quiz/quiz_detail/6	0,32
http://local.ujianharian.com/quiz/validate_quiz/6	1,09
http://local.ujianharian.com/index.php/quiz/save_answer/	0,25
http://local.ujianharian.com/index.php/quiz/submit_quiz/	0,41
http://local.ujianharian.com/result	0,15
Rata-Rata Seluruh URL (Detik)	0,30

Berdasarkan hasil pengujian *time behaviour* pada Tabel 6.60, nilai rata-rata waktu respon sistem seluruh URL adalah 0,30 detik.

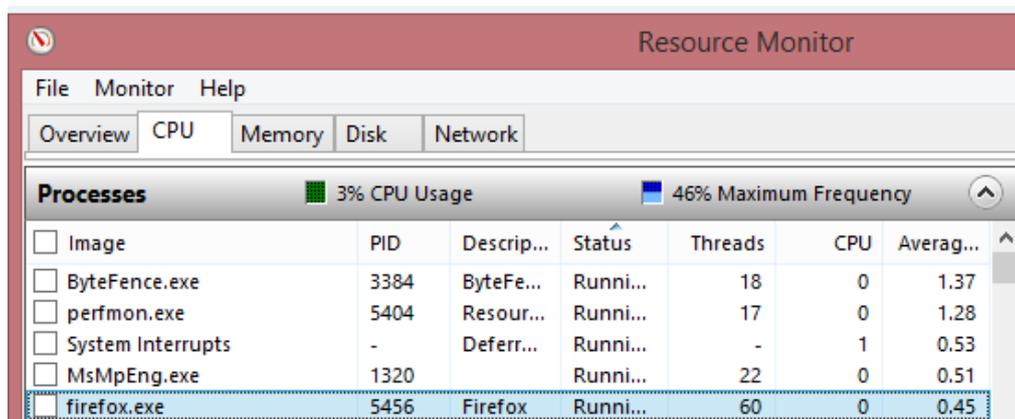
b. Resource Utilization

Pengujian pada subkarakteristik ini dilakukan dengan menggunakan *tool* Resource Monitor. Berikut Gambar 6.1 yang menunjukkan jumlah RAM yang digunakan oleh Mozilla Firefox tanpa membuka aplikasi berbasis *web* apapun.



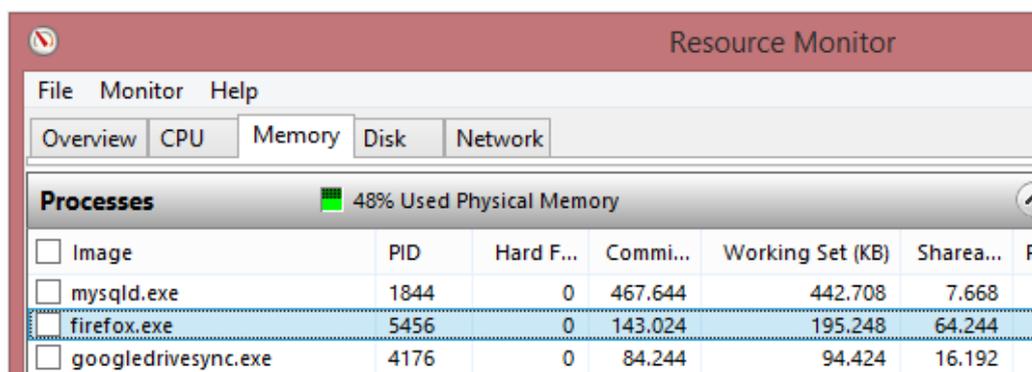
Gambar 6.1 RAM yang digunakan mozilla tanpa membuka aplikasi berbasis web apapun

Sedangkan pada Gambar 6.2 menunjukkan jumlah CPU yang digunakan oleh Mozilla Firefox tanpa membuka aplikasi berbasis web apapun.



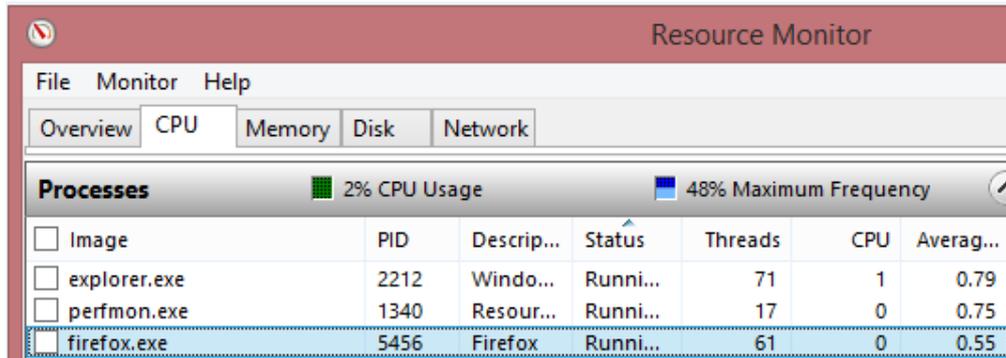
Gambar 6.2 CPU yang digunakan mozilla tanpa membuka aplikasi berbasis web apapun

Gambar 6.3 menunjukkan jumlah RAM yang digunakan oleh Mozilla Firefox dengan hanya membuka Sistem Ujian Harian Siswa.



Gambar 6.3 RAM yang digunakan mozilla dengan hanya membuka Sistem Ujian Harian Siswa

Gambar 6.4 menunjukkan jumlah CPU yang digunakan oleh Mozilla Firefox dengan hanya membuka Sistem Ujian Harian Siswa.



Gambar 6.4 CPU yang digunakan mozilla dengan hanya membuka Sistem Ujian Harian Siswa

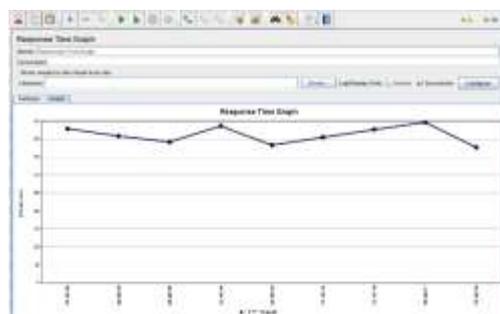
RAM yang dibutuhkan Sistem Ujian Harian Siswa = 195.248 KB – 193.788 KB
 = 1.460 KB
 = 1,460 MB

CPU yang dibutuhkan Sistem Ujian Harian Siswa = 0,55% - 0,45%
 = 0,10 %

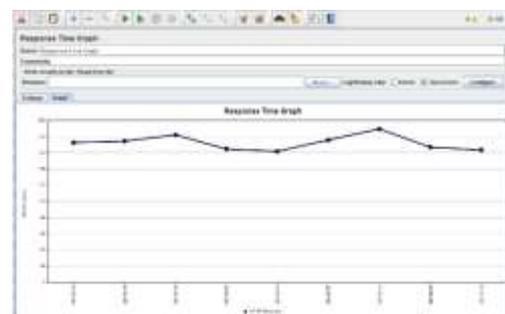
Jadi, jumlah RAM yang dibutuhkan oleh Sistem Ujian Harian Siswa adalah $\pm 1,460$ MB dan CPU yang dibutuhkan oleh Sistem Ujian Harian Siswa adalah $\pm 0,10\%$ dengan spesifikasi perangkat keras prosesor Intel Core i3 dan memori (RAM) sebesar 6GB.

c. Capacity

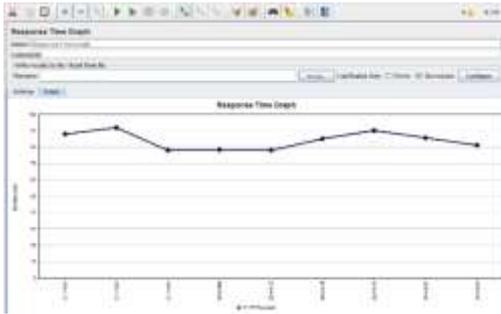
Pengujian pada subkarakteristik ini dilakukan dengan menggunakan *tool* Apache jMeter. Sistem Ujian Harian Siswa diuji dengan beban dimulai 30 *user* dan setiap *user* melakukan *action* setiap 30 detik sekali. 30 *user* dipilih karena pada saat ujian harian berlangsung, jumlah siswa yang melakukan ujian harian berjumlah 30 siswa pada waktu yang sama. Kemudian, jumlah *user* ditambah 1 secara terus menerus sampai batas stabil sistem. Jumlah *user* yang didapatkan pada batas stabil sistem adalah nilai dari *capacity*. Hasil pengujian *capacity* dapat dilihat secara lengkap pada Lampiran. Hasil pengujian *capacity* dapat dilihat pada Gambar 6.5.



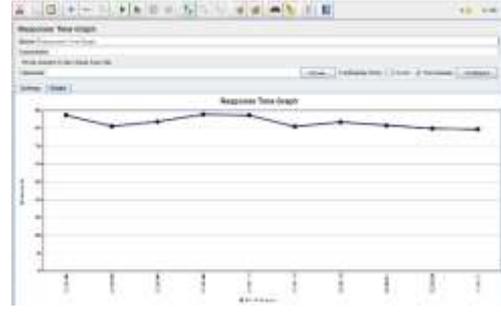
30 user



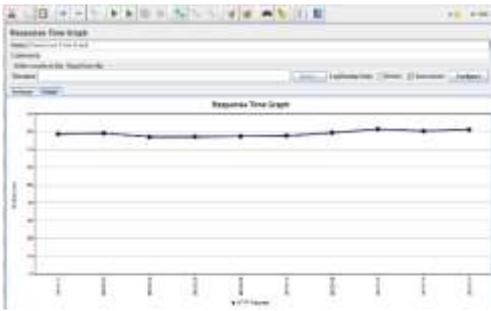
50 user



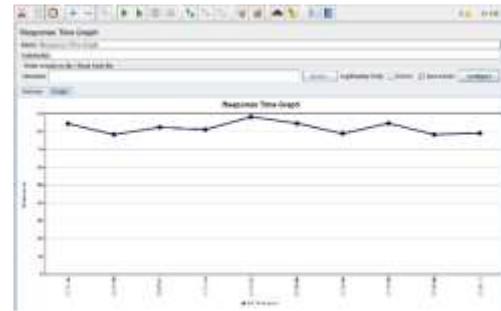
75 user



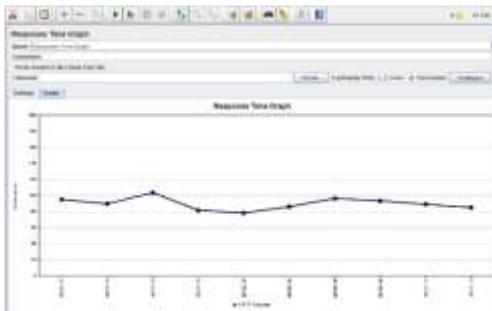
90 user



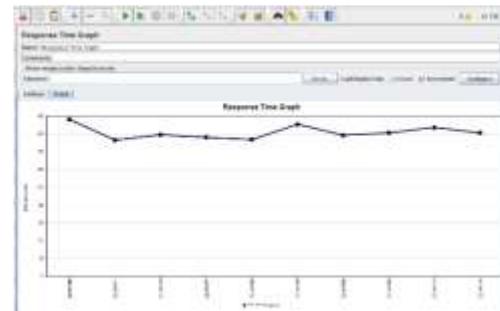
100 user



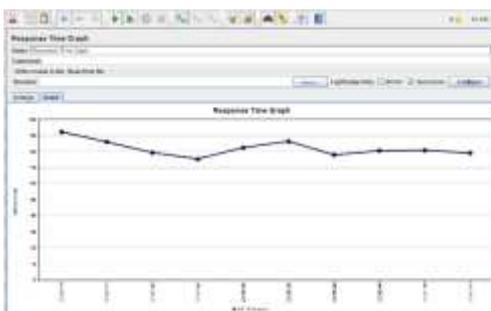
110 user



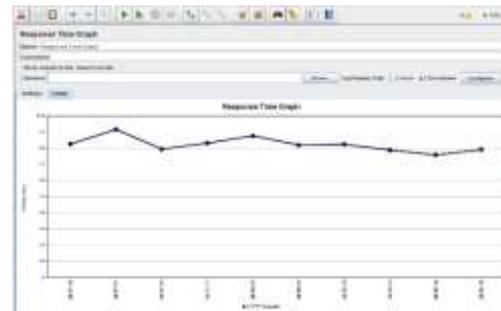
120 user



130 user



131 user



132 user



Gambar 6.5 Hasil pengujian *capacity*

3. *Compatibility*

Pengujian *compatibility* dilakukan dengan cara menjalankan sistem pada *browser* dalam hal ini menggunakan *browser* chrome yang dimana saat sistem ujian harian dijalankan bersama dengan sistem lain maka semua sistem yang berada dalam *browser* chrome ini dapat menjalankan fungsinya masing-masing dengan baik tanpa mengalami gangguan. Pada Gambar 6.6 menunjukkan sistem ujian harian dan sistem lain dapat berjalan baik dalam *browser* chrome. Kemudian dilakukan pengujian dengan menjalankan sistem pada perangkat keras dan sistem operasi yang berbeda. Pada Gambar 6.7, 6.8, 6.9 akan menunjukkan sistem dapat berjalan dengan baik pada *browser* chrome, mozilla dan opera pada perangkat Toshiba i3 dan Gambar 6.10 dan 6.11 yang menunjukkan sistem dapat berjalan dengan baik pada *browser* chrome dan mozilla pada perangkat Asus i5.



Gambar 6.6 Halaman *browser* chrome dengan menjalankan dua sistem yang berbeda



Gambar 6.7 Halaman *login browser* chrome pada laptop Toshiba i3



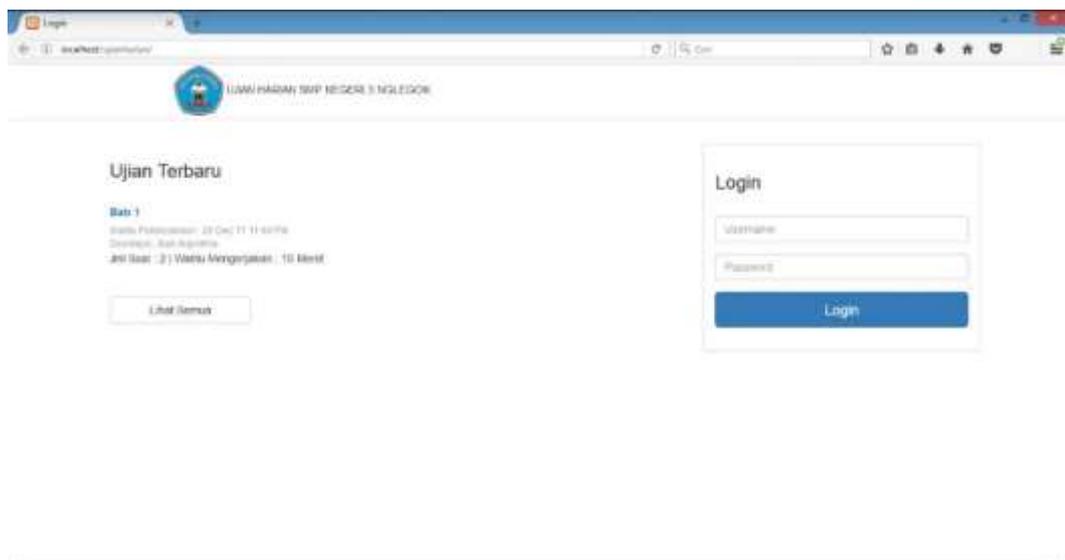
Gambar 6.8 Halaman *login browser* opera pada laptop Toshiba i3



Gambar 6.9 Halaman *login browser mozilla* pada laptop Toshiba i3



Gambar 6.10 Halaman *login browser chrome* pada laptop Asus i5



Gambar 6.11 Halaman *login browser mozilla* pada laptop Asus i5

4. Usability

Usability memiliki 6 subkarakteristik yaitu *appropriateness recognizability*, *learnability*, *operability*, *user error protection*, *user error protection*, *user error aesthetics* serta *accessibility*. Pengujian *usability* dilakukan dengan menggunakan kuesioner, yaitu *USE Questionnaire*. Penggunaan *USE Questionnaire* dikarenakan seluruh pertanyaan yang ada di dalamnya telah mencakup seluruh subkarakteristik yang ada pada karakteristik *Usability*. Minimum responden dari kuesioner *usability* adalah 5 responden (Nielsen, 2012). Yang menjadi responden dari kuesioner ini adalah 10 guru SMPN 3 Nglegok, dan 10 siswa SMPN 3 Nglegok. Rekap hasil pengisian kuesioner dapat dilihat pada Tabel 6.61 berikut.

Tabel 6.68 Rekap hasil pengisian kuesioner

Responden	Total Jawaban				
	STS	TS	RR	S	SS
Responden 1	0	0	0	16	14
Responden 2	0	0	3	16	11
Responden 3	0	0	1	24	5
Responden 4	0	0	0	15	15
Responden 5	0	0	0	30	0
Responden 6	0	0	0	19	11
Responden 7	0	0	0	19	11
Responden 8	0	0	2	15	13
Responden 9	0	0	3	17	10
Responden 10	0	0	0	9	21
Responden 11	0	0	2	17	11
Responden 12	0	0	0	25	5
Responden 13	0	0	2	21	7
Responden 14	0	0	2	19	9
Responden 15	0	0	0	12	18
Responden 16	0	0	0	16	14
Responden 17	0	0	0	9	21
Responden 18	0	0	0	11	19
Responden 19	0	0	0	16	14
Responden 20	0	0	0	13	17

Berdasarkan pada rekap hasil pengisian kuesioner pada Tabel 6.61, dapat diketahui skor total sebagai berikut.

$$\text{Skor}_{\text{total}} = (J_{\text{SS}} \times 5) + (J_{\text{S}} \times 4) + (J_{\text{RR}} \times 3) + (J_{\text{TS}} \times 2) + (J_{\text{STS}} \times 1)$$

$$\text{Skor}_{\text{total}} = (246 \times 5) + (339 \times 4) + (15 \times 3) + (0 \times 2) + (0 \times 1)$$

$$\text{Skor}_{\text{total}} = 1230 + 1356 + 45$$

$$\text{Skor}_{\text{total}} = 2631$$

Setelah mendapatkan skor total maka didapatkan persentase skor untuk mendapatkan interpretasi hasil pengujian *Usability* sebagai berikut.

$$P_{skor} = \frac{Skor_{total}}{i \times r \times 5} \times 100\%$$

$$P_{skor} = \frac{2631}{30 \times 20 \times 5} \times 100\%$$

$$P_{skor} = 87,7\%$$

5. Reliability

Reliability memiliki 4 subkarakteristik yaitu *maturity*, *availability*, *fault tolerance* dan *recoverability*. Namun pada pengujian ini hanya dilakukan pada subkarakteristik *maturity* dan *availability* berdasarkan kebutuhan nonfungsional yang telah didefinisikan sebelumnya.

a. Maturity

Pengujian pada subkarakteristik *maturity* menggunakan *Webserver Stress Testing*. Berikut hasil pengujian subkarakteristik *maturity* dengan masing-masing beban pengguna yang diberikan beserta waktu jeda perklik menunya.

1) Pengguna 30 *user* dengan jeda perklik 10 *second*

View Logfile Results						
Logfiles		Results per User (Complete Test)			Results per URL (Complete Test)	
URL No.	Name	Clicks	Errors	Errors [%]	Time Spent [ms]	Avg. Click Time [ms]
1	masuk sistem	35	0	0,00	8.190	234
2	mengerjakan ujian	32	0	0,00	13.346	417
3	melihat hasil ujian	31	0	0,00	13.289	429

Gambar 6.12 Hasil pengujian *maturity* dengan 30 *user* dengan jeda 10 *second*

2) Pengguna 30 *user* dengan jeda perklik 5 *second*

View Logfile Results						
Logfiles		Results per User (Complete Test)			Results per URL (Complete Test)	
URL No.	Name	Clicks	Errors	Errors [%]	Time Spent [ms]	Avg. Click Time [ms]
1	masuk sistem	65	0	0,00	26.604	409
2	mengerjakan ujian	61	0	0,00	38.095	625
3	melihat hasil ujian	56	0	0,00	35.437	633

Gambar 6.13 Hasil pengujian *maturity* dengan 30 *user* dengan jeda 5 *second*

3) Pengguna 30 *user* dengan jeda perklik 2 *second*

View Logfile Results						
Logfiles		Results per User (Complete Test)			Results per URL (Complete Test)	
URL No.	Name	Clicks	Errors	Errors [%]	Time Spent [ms]	Avg. Click Time [ms]
1	masuk sistem	134	0	0,00	89.424	667
2	mengerjakan ujian	111	0	0,00	121.451	1.094
3	melihat hasil ujian	105	0	0,00	115.297	1.098

Gambar 6.14 Hasil pengujian *maturity* dengan 30 *user* dengan jeda 2 *second*

4) Pengguna 50 *user* dengan jeda perklik 5 *second*

View Logfile Results						
Logfiles		Results per User (Complete Test)			Results per URL (Complete Test)	
URL No.	Name	Clicks	Errors	Errors [%]	Time Spent [ms]	Avg. Click Time [ms]
1	masuk sistem	102	0	0,00	68.777	674
2	mengerjakan ujian	93	0	0,00	68.111	732
3	melihat hasil ujian	90	0	0,00	67.003	744

Gambar 6.15 Hasil pengujian *maturity* dengan 50 *user* dengan jeda 5 *second*

5) Pengguna 90 *user* dengan jeda perklik 5 *second*

View Logfile Results						
Logfiles		Results per User (Complete Test)			Results per URL (Complete Test)	
URL No.	Name	Clicks	Errors	Errors [%]	Time Spent [ms]	Avg. Click Time [ms]
1	masuk sistem	168	0	0,00	162.509	967
2	mengerjakan ujian	156	0	0,00	209.858	1.345
3	melihat hasil ujian	155	0	0,00	200.634	1.294

Gambar 6.76 Hasil pengujian *maturity* dengan 90 *user* dengan jeda 5 *second*

6) Pengguna 100 *user* dengan jeda perklik 5 *second*

View Logfile Results						
Logfiles		Results per User (Complete Test)			Results per URL (Complete Test)	
URL No.	Name	Clicks	Errors	Errors [%]	Time Spent [ms]	Avg. Click Time [ms]
1	masuk sistem	187	0	0,00	191.684	1.025
2	mengerjakan ujian	171	0	0,00	222.753	1.303
3	melihat hasil ujian	168	0	0,00	216.856	1.291

Gambar 6.17 Hasil pengujian *maturity* dengan 100 *user* dengan jeda 5 *second*

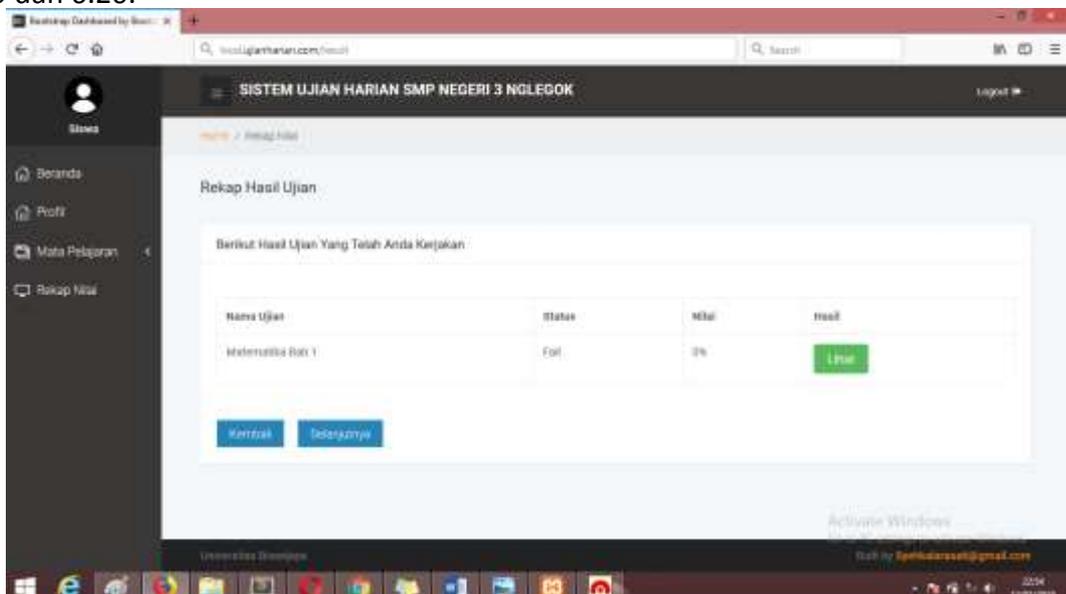
7) Pengguna 132 user dengan jeda perklik 5 second

Logfiles		Results per User (Complete Test)			Results per URL (Complete Test)	
URL No.	Name	Clicks	Errors	Errors [%]	Time Spent [ms]	Avg. Click Time [ms]
1	masuk sistem	214	0	0,00	332.463	1.554
2	mengerjakan ujian	193	0	0,00	410.047	2.125
3	melihat hasil ujian	190	0	0,00	418.215	2.201

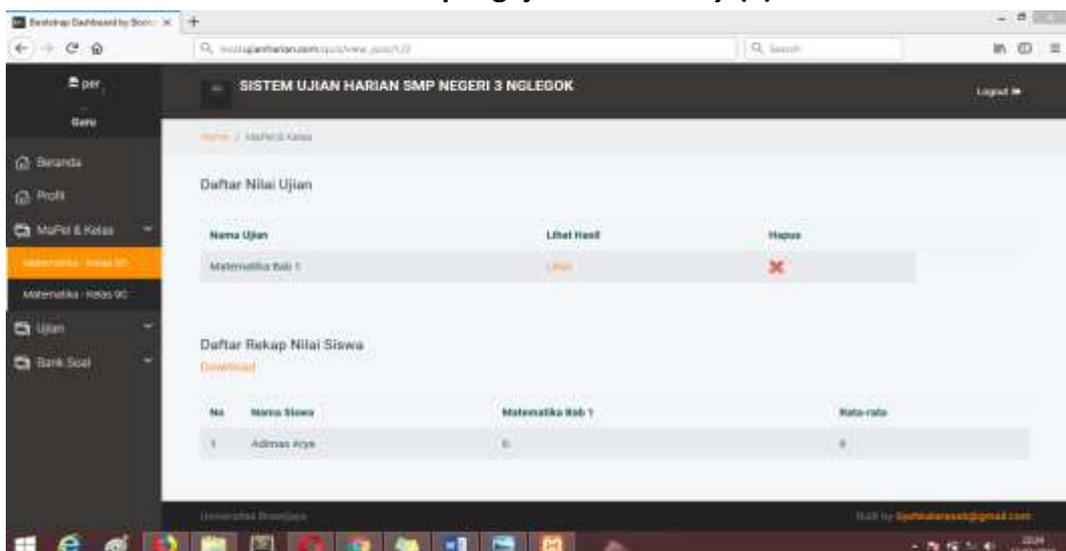
Gambar 6.18 Hasil pengujian *maturity* dengan 132 user dengan jeda 5 second

b. Availability

Hasil pengujian subkarakteristik *accountability* dapat dilihat pada Gambar 6.19 dan 6.20.



Gambar 6.89 Hasil pengujian *availability* (1)

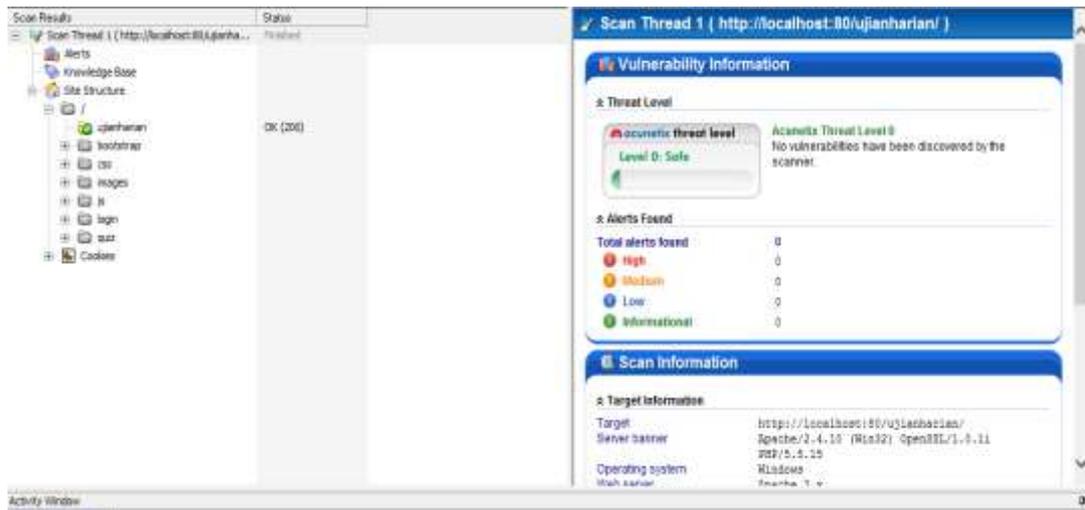


Gambar 6.20 Hasil pengujian *availability* (2)

6. Security

a. Confidentiality, Integrity, dan Non-Repudiation

Pengujian pada subkarakteristik *confidentiality*, *integrity* dan *non-repudiation* menggunakan aplikasi Acunetix Web Vulnerability Scanner. Berikut hasil pengujian subkarakteristik *confidentiality*, *integrity* dan *non-repudiation* dapat dilihat pada Gambar 6.21.



Gambar 6.21 Hasil pengujian *confidentiality*, *integrity*, dan *non-repudiation*

b. Accountability

Hasil pengujian subkarakteristik *accountability* dapat dilihat pada Gambar 6.22.



Gambar 6.22 Hasil pengujian *accountability*

c. Authenticity

Hasil pengujian subkarakteristik *authenticity* dapat dilihat pada Tabel 6.62 berikut.

Tabel 6.69 Hasil pengujian *authenticity*

No	Data Input	Hasil yang Diharapkan	Hasil yang Diperoleh	Status
1	Memasukkan email "admin@example.com" pada bagian <i>username</i> dan "admin" pada bagian <i>password</i> . <i>Username</i> dan <i>password</i> tersebut adalah milik Admin	Sistem dapat membuktikan bahwa <i>username</i> "admin@example.com" dan <i>password</i> "admin" adalah milik Admin dengan cara sistem dapat menampilkan halaman beranda admin.	Sistem dapat membuktikan bahwa <i>username</i> "admin@example.com" dan <i>password</i> "admin" adalah milik Admin dengan cara sistem dapat menampilkan halaman beranda admin.	Valid
2	Memasukkan email "guru@gmail.com" pada bagian <i>username</i> dan "guru" pada bagian <i>password</i> . <i>Username</i> dan <i>password</i> tersebut adalah milik Guru Utama	Sistem dapat membuktikan bahwa <i>username</i> "guru@gmail.com" dan <i>password</i> "guru" adalah milik Guru Utama dengan cara sistem dapat menampilkan halaman beranda Guru Utama	Sistem dapat membuktikan bahwa <i>username</i> "guru@gmail.com" dan <i>password</i> "guru" adalah milik Guru Utama dengan cara sistem dapat menampilkan halaman beranda Guru Utama	Valid
3	Memasukkan email "syehka@gmail.com" pada bagian <i>username</i> dan "123" pada bagian <i>password</i> . <i>Username</i> dan <i>password</i> tersebut adalah milik Syehka Sofia Arya Larasati	Sistem dapat membuktikan bahwa <i>username</i> "syehka@gmail.com" dan <i>password</i> "123" adalah milik Syehka Sofia Arya Larasati dengan cara sistem dapat menampilkan halaman beranda Syehka Sofia Arya Larasati	Sistem dapat membuktikan bahwa <i>username</i> "syehka@gmail.com" dan <i>password</i> "123" adalah milik Syehka Sofia Arya Larasati dengan cara sistem dapat menampilkan halaman beranda Syehka Sofia Arya Larasati	Valid

6.2 Analisis

Analisis dilakukan untuk mendapatkan kesimpulan dari hasil uji yang dilakukan oleh sistem, yakni pengujian unit, pengujian integrasi, pengujian validasi, dan pengujian berbasis ISO 25010. Proses analisis dilakukan sesuai dengan pengujian yang dilakukan pada sistem yakni analisis hasil uji pengujian unit, analisis hasil uji pengujian integrasi, analisis uji pengujian validasi dan analisis hasil uji pengujian berbasis ISO 25010.

6.2.1 Analisis Hasil Uji Pengujian Unit

Dari kasus uji yang telah dilaksanakan sesuai dengan prosedur pengujian basis *path* yang telah disebutkan dalam sub pokok bahasan 6.1.1 pada algoritme *Login*, mengerjakan ujian serta hasil ujian menunjukkan bahwa seluruh kasus uji memiliki hasil yang telah sesuai dengan yang diharapkan atau bernilai valid.

6.2.2 Analisis Hasil Uji Pengujian Integrasi

Dari kasus uji yang telah dilaksanakan sesuai dengan prosedur pengujian basis *path* yang telah disebutkan dalam sub pokok bahasan 6.1.2 pada *method* *insert_ajar()*, *result_quiz()*, dan *remove_quiz()* menunjukkan bahwa seluruh kasus uji memiliki hasil yang telah sesuai dengan yang diharapkan atau bernilai valid.

6.2.3 Analisis Hasil Pengujian Validasi

Dari kasus uji yang telah dilaksanakan sesuai dengan prosedur pengujian validasi yang telah disebutkan dalam sub pokok bahasan 6.1.3 maka didapatkan hasil seperti yang ditunjukkan dalam Tabel 6.63 berikut.

Tabel 6.70 Hasil pengujian validasi

No.	Kasus Uji	Hasil yang Didapatkan	Status
1.	Uji <i>Login</i>	Sistem dapat mengidentifikasi pengguna berdasarkan hak aksesnya	Valid
2.	Uji Logout	Sistem dapat mengeluarkan pengguna dari sistem	Valid
3.	Uji Melihat Profil Siswa	Sistem dapat menampilkan halaman profil	Valid
4.	Uji Melihat Profil Guru	Sistem dapat menampilkan halaman profil	Valid
5.	Uji Mengubah Profil Siswa	Sistem dapat mengupdate profil siswa	Valid
6.	Uji Mengubah Profil Guru	Sistem dapat mengupdate profil guru	Valid
7.	Uji Menambah Admin	Sistem dapat menambahkan admin baru	Valid

Tabel 6.63 Hasil pengujian validasi (lanjutan)

8.	Uji Menambah Guru	Sistem dapat menambahkan guru baru	Valid
9.	Uji Menambah Siswa	Sistem dapat menambahkan siswa baru	Valid
10.	Uji Mengubah Admin	Sistem dapat memperbarui data profil admin	Valid
11.	Uji Mengubah Guru	Sistem dapat memperbarui data profil guru	Valid
12.	Uji Mengubah Siswa	Sistem dapat memperbarui data profil siswa	Valid
13.	Uji Menghapus Admin	Sistem dapat menghapus admin	Valid
14.	Uji Menghapus Guru	Sistem dapat menghapus guru	Valid
15.	Uji Menghapus Siswa	Sistem dapat menghapus siswa	Valid
16.	Uji Menambah Mata Pelajaran	Sistem dapat menyimpan mata pelajaran baru kedalam sistem	Valid
17.	Uji Mengubah Mata Pelajaran	Sistem dapat memperbarui nama mata pelajaran	Valid
18.	Uji Menghapus Mata Pelajaran	Sistem dapat menghapus nama pelajaran	Valid
19.	Uji Menambah Kelas	Sistem dapat menyimpan kelas baru kedalam sistem	Valid
20.	Uji Mengubah Kelas	Sistem dapat memperbarui nama kelas	Valid
21.	Uji Menghapus Kelas	Sistem dapat menghapus kelas	Valid
22.	Uji Memasukkan Guru Ke Mata Pelajaran dan Kelas	Sistem dapat menyimpan mata pelajaran dan kelas yang diampu guru kedalam system	Valid
23.	Uji Menambah Soal dan Jawaban	Sistem dapat menyimpan soal dan jawaban baru kedalam bank soal sistem	Valid

Tabel 6.63 Hasil pengujian validasi (lanjutan)

24.	Uji Mengubah Soal dan Jawaban	Sistem dapat memperbarui soal	Valid
25.	Uji Menghapus Soal dan Jawaban	Sistem dapat menghapus soal	Valid
26.	Uji Melihat Mata Pelajaran dan Kelas	Sistem dapat menampilkan mata pelajaran dan kelas yang diampu oleh guru	Valid
27.	Uji Membuat Ujian Harian	Sistem dapat menyimpan ujian baru kedalam sistem	Valid
28.	Uji Mengubah Ujian Harian	Sistem dapat memperbarui ujian	Valid
29.	Uji Menghapus Ujian Harian	Sistem dapat menghapus ujian	Valid
30.	Uji Melihat Hasil Ujian Siswa	Sistem dapat menampilkan hasil ujian siswa	Valid
31.	Uji Melihat Rekap Hasil Ujian Siswa	Sistem dapat menampilkan rekap hasil ujian siswa pada Mapel&Kelas yang dituju	Valid
32.	Uji Melihat Daftar Mata Pelajaran	Sistem dapat menampilkan mata pelajaran	Valid
33.	Uji Melihat Daftar Ujian	Sistem dapat menampilkan daftar ujian pada mata pelajaran yang ditempuh	Valid
34.	Uji Mengerjakan Ujian	Sistem dapat menyimpan hasil ujian siswa	Valid
35.	Uji Melihat Hasil Ujian	Sistem dapat menampilkan hasil ujian siswa	Valid
36.	Uji Melihat Rekap Hasil Ujian	Sistem dapat menampilkan rekap hasil ujian siswa	Valid

Berdasarkan hasil pengujian validasi, seluruh kasus uji yang didefinisikan telah valid.

6.2.4 Analisis Hasil Uji Pengujian Berbasis ISO 25010

1. Analisis Hasil Uji Pengujian *Functional Suitability*

Berdasarkan hasil pengujian pada karakteristik *functional suitability* didapatkan hasil pengujian pada seluruh kebutuhan fungsional bernilai valid dengan persentase keberhasilan 100% atau sangat baik. Sehingga sistem ujian harian siswa telah memenuhi karakteristik *functional suitability*.

2. Analisis Hasil Uji Pengujian *Performance Efficiency*

a. Analisis Hasil Uji Pengujian *Time Behaviour*

Berdasarkan hasil pengujian pada subkarakteristik *time behaviour*, didapatkan rata-rata waktu respon sistem sebesar 0,30 detik. Kemudian rata-rata waktu respon tersebut dibandingkan dengan Tabel Standar Jakob Nielsen. Sehingga jika hasil tersebut dibandingkan dengan standar Jakob Nielsen, maka dapat dikatakan pengguna merasakan adanya jeda namun masih fokus dalam sistem. Dan untuk memenuhi standar, dibutuhkan nilai kurang dari 10 detik. Dengan demikian, dapat disimpulkan bahwa sistem ujian harian siswa telah memenuhi subkarakteristik *time behaviour*.

b. Analisis Hasil Uji Pengujian *Resource Utilization*

Berdasarkan hasil pengujian pada subkarakteristik *resource utilization*, jumlah RAM yang dibutuhkan oleh Sistem Ujian Harian Siswa adalah $\pm 1,460$ MB dan CPU yang dibutuhkan oleh Sistem Seleksi Penerimaan Siswa Baru SMA adalah $\pm 0,10$ % dengan spesifikasi perangkat keras prosesor Intel Core i3 dan memori (RAM) sebesar 6GB.

c. Analisis Hasil Uji Pengujian *Capacity*

Berdasarkan hasil pengujian pada subkarakteristik *capacity*, sistem yang diberi beban 133 *user* menunjukkan hasil *response time* yang tidak stabil. Ketidakstabilan *response time* apabila diberi beban 133 *user* dapat dilihat pada waktu 21:03:06 dengan *response time* sebesar 88 ms, pada waktu 21:03:09 dengan *response time* yang naik drastis menjadi sebesar 119 ms, waktu 21:03:12 dengan *response time* sebesar 120 ms dan kemudian pada waktu 21:03:16 *response time* yang turun drastis menjadi sebesar 89 ms. Dengan demikian, *capacity* dari Sistem Ujian Harian Siswa adalah 132 *user*. Gambar 6.23 menunjukkan hasil pengujian *capacity* dengan 133 *user*.

3. Analisis Hasil Uji Pengujian *Compatibility*

Berdasarkan hasil pengujian pada subkarakteristik *co-existence* didapatkan hasil bahwa Sistem Ujian Harian Siswa dapat berjalan dengan baik walau berdampingan dengan sistem lain dalam satu *browser* chrome. Hasil pengujian selanjutnya menunjukkan bahwa sistem dapat berjalan dengan baik pada *browser* chrome, mozilla dan opera di sistem operasi dan perangkat keras Toshiba i3 dan Asus i5.

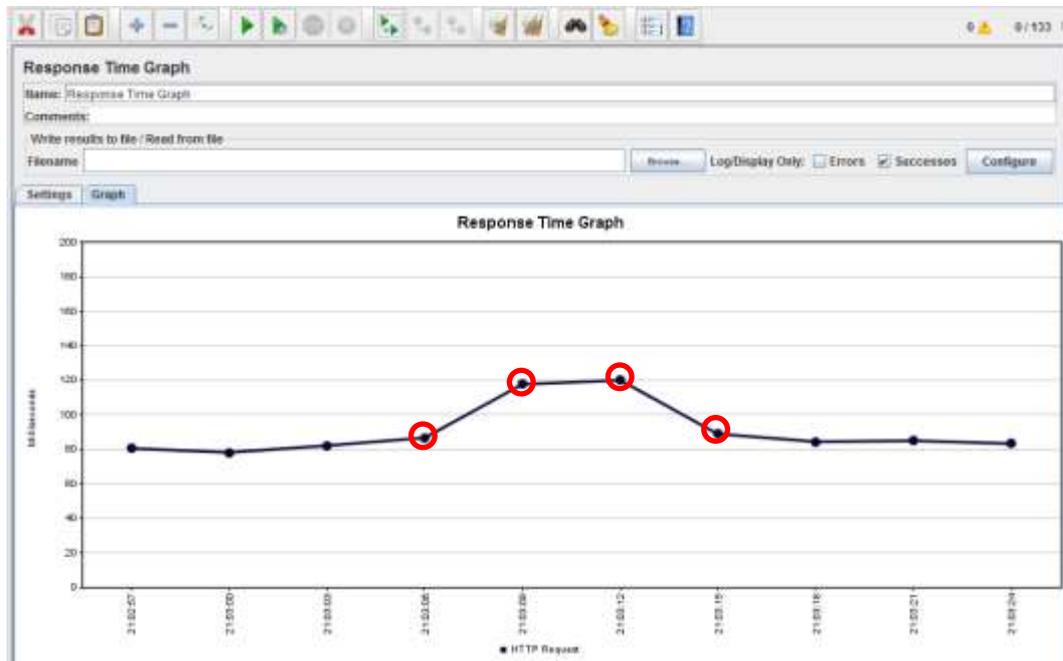
4. Analisis Hasil Uji Pengujian *Usability*

Berdasarkan hasil pengujian pada karakteristik *usability* didapatkan persentase skor sebesar 87,7%. Setelah didapatkan persentase skor maka hasil persentase skor tersebut akan dibandingkan dengan tabel kategori penilaian *usability*. Berikut Tabel 6.64 yang menunjukkan kategori penilaian *usability*.

Tabel 6.71 Kategori Penilaian *Usability*

Persentase Skor	Interpretasi
0% - 20%	Sangat Tidak Layak
21% - 40%	Tidak Layak
41% - 60%	Cukup Layak
61% - 80%	Layak
81% - 100%	Sangat Layak

Setelah dibandingkan dengan kategori penilaian *usability* maka Sistem Ujian Harian Siswa dapat dikatakan sangat layak. Sehingga dapat ditarik kesimpulan bahwa Sistem Ujian Harian Siswa telah memenuhi karakteristik *usability*.



Gambar 6.23 Hasil Pengujian *Capacity* Dengan 133 User

5. Analisis Hasil Uji Pengujian *Reliability*

a. Analisis Hasil Uji Pengujian *Maturity*

Berdasarkan hasil pengujian pada subkarakteristik *maturity* didapatkan hasil bahwa dengan sistem diberi uji berupa jumlah pengguna yang membeludak sampai 132 *user* namun sistem tetap bisa berjalan stabil dan dengan baik tanpa terjadi error. Hanya dengan semakin banyaknya pengguna yang mengakses sistem, waktu yang diperlukan untuk mengakses sistem akan sedikit lebih lama.

b. Analisis Hasil Uji Pengujian *Availability*

Berdasarkan hasil pengujian pada subkarakteristik *availability* didapatkan hasil bahwa Sistem Ujian Harian Siswa dapat dijalankan selama 24 jam. Terbukti bahwa sistem dapat diakses pada pukul 22:54 dan 22:23.

6. Analisis Hasil Uji Pengujian *Security*

a. Analisis Hasil Uji Pengujian *Confidentiality, Integrity, dan Non-repudiation*

Berdasarkan hasil pengujian pada subkarakteristik *confidentiality, integrity, dan non-repudiation*, didapatkan hasil yaitu tingkat keamanan level *safe* dimana tidak ditemukan kerentanan di dalam system. Dengan demikian, sistem seleksi memenuhi subkarakteristik *confidentiality, integrity, dan non-repudiation*.

b. Analisis Hasil Uji Pengujian *Accountability*

Berdasarkan hasil pengujian pada subkarakteristik *accountability*, didapatkan bahwa sistem dapat merekam aktivitas pengguna. Dengan adanya fitur yang dapat merekam aktivitas pengguna maka dapat diketahui pengguna mana saja yang telah melakukan interaksi dengan sistem pada waktu dan jam tertentu. Dengan demikian, sistem seleksi telah memenuhi subkarakteristik *accountability* karena sistem dapat merekam aktivitas pengguna.

c. Analisis Hasil Uji Pengujian *Authenticity*

Dari kasus uji yang telah dilakukan menunjukkan bahwa seluruh kasus uji memiliki hasil yang sesuai dengan yang diharapkan atau bernilai valid. Dengan demikian, Sistem Ujian Harian Siswa telah memenuhi subkarakteristik *authenticity*.