

GL03

DOKUMEN UJI PERANGKAT LUNAK

STEVENT

untuk:

<nama pelanggan>

Dipersiapkan oleh:

Muhammad Alfhi Saputra 1301171205

Athallah Tsany Rakha Dzaky 1301174026

Kartika Putri Dwi Atmojo 1301170084

Indri Islamiawati Putri 1301170458

Prodi Teknik Informatika - Universitas Telkom

	Prodi Teknik Informatika Universitas Telkom	Nomor Dokumen		Halaman
		<i>DUPL-STEVENT 001</i>		<#>/<jml #
		Revisi	<nomor revisi>	Tgl: <isi tanggal>

Daftar Isi

1	Pendahuluan	6
1.1	Tujuan Pembuatan Dokumen	6
1.2	Deskripsi Umum Sistem	6
1.3	Definisi dan Singkatan	6
1.4	Referensi	6
2	Lingkungan Pengujian Perangkat Lunak	6
2.1	Perangkat Lunak Pengujian	6
2.2	Perangkat Keras Pengujian	6
2.3	Material Pengujian	6
2.4	Sumber Daya Manusia	6
2.5	Metode Pengujian	6
2.6	Tujuan Pengujian	6
2.7	Rencana Pengujian	7
2.8	Pelaksanaan Pengujian	7
2.8.1	Pengujian DUPL-01 Login User	7
2.9	Kesimpulan Pengujian	8
3	Lampiran	8

Daftar Gambar

Hanya dicantumkan dan diisi jika ada gambar pada badan dokumen

Daftar Tabel

Hanya dicantumkan dan diisi jika ada tabel pada badan dokumen

Daftar Lampiran

Hanya dicantumkan dan diisi jika ada lampiran setelah badan dokumen

1 Pendahuluan

1.1 Tujuan Pembuatan Dokumen

Pembuatan dokumen ini bertujuan sebagai panduan pengujian terhadap perangkat lunak Website Penjualan dan Pengelolaan tiket Event dengan melihat kesesuaian hasil uji perangkat lunak dengan perancangan yang telah dilakukan. Dokumen ini ditujukan pada pengembang sebagai bahan pertimbangan pengembangan perangkat lunak agar memenuhi target perancangan.

1.2 Ruang Lingkup Pengujian

Ruang lingkup pengujian meliputi Perangkat Lunak Pengujian, Perangkat Keras Pengujian, Material Pengujian, Metode Pengujian, Sumber Daya Manusia, Tujuan Pengujian, Rencana Pengujian, Pelaksanaan Pengujian dan Kesimpulan Pengujian

1.3 Referensi

Referensi yang digunakan

Menjelaskan referensi SKPL & DPPL yang akan digunakan sebagai rujukan pembuatan dokumen ini

1.4 Overview Sistem & Fitur Utamanya

Secara keseluruhan, sistem yang dibangun merupakan sistem penjualan dan pengelolaan tiket secara online. Pengguna dibagi menjadi dua, yaitu admin dan user biasa. Di bagian user, terdapat beberapa fungsi, diantaranya: register, login, membeli tiket, membuat event, mengedit event, menghapus event dan memvalidasi peserta. Kemudian, pada pihak admin memiliki fungsi-fungsi berikut: register, login, validasi event,

Menjelaskan deskripsi umum dari sistem/perangkat lunak yang diuji.

1.5 Overview Pengujian

1.5.1 Perangkat Keras Pengujian

Perangkat keras yang dilakukan dalam pengujian ini adalah:

- Laptop ASUS ZenBook UX410UQ dengan spesifikasi:
 - CPU. Intel Core i7-7500U 89.
 - GPU. NVIDIA GeForce 940MX (2GB DDR3) 131.
 - HDD/SSD. 256GB M.2 SATA SSD.
 - RAM. 8GB DDR4, 2400 MHz.

1.5.2 Sumber Daya Manusia

Sumber daya manusia yang diperlukan untuk pengujian perangkat lunak ini adalah pengembang yang memahami:

- Keseluruhan konsep sistem penjualan dan pengelolaan tiket event
- Konsep bahasa pemrograman PHP

- Konsep basis data menggunakan XAMPP

1.5.3 Perangkat Lunak Pengujian

Perangkat Lunak Website Penjualan dan Pengelolaan Tiket Event ini diuji menggunakan beberapa perangkat lunak, yaitu:

1. Sistem operasi: Windows 10
2. Bahasa pemrograman: PHP
3. Web server: XAMPP ver 7.0 untuk database
4. Editor: Visual Studio Code 1.40.2
5. Browser: Google Chrome ver 70

1.5.4 Material Pengujian

Pengguna	Modul Uji	Keterangan
Admin	Login	Login sebagai admin untuk masuk ke dalam sistem dan mendapatkan hak akses berupa menu validasi peserta, daftar absen peserta dan*
	Validasi event	Dalam menu ini, admin mampu menentukan event-event mana yang sesuai dan asli. ketika event telah divalidasi oleh admin, barulah event akan muncul pada laman utama dari user.
	Validasi user	Dalam menu ini, admin mampu melakukan blokir pada user-user yang dianggap bermasalah atau sudah tidak lagi aktif agar tidak dapat melakukan update, pembuatan event ataupun melakukan pembelian.
User	Login	Login sebagai user untuk masuk ke dalam sistem untuk mendapat hak akses untuk membeli tiket event, dan membuat event.
	Pembuatan Event	Pada menu ini, user dapat membuat event dengan menginputkan harga tiket,

		poster, serta deskripsi tentang event yang akan dibuat.
	Pembelian tiket	Pada menu ini. user bertindak sebagai peserta dari event. user mampu melihat deskripsi dari event, membeli tiket event dan melakukan pembayaran, serta melakukan validasi dengan mengirimkan bukti pembayaran.
	Edit event	Pada menu ini, user dapat melakukan pergantian pada bagian-bagian event yang dikira kurang sesuai dengan data atau deskripsi yang kurang sesuai dengan preferensi user.
	Registrasi	Pada menu ini, user menginputkan data-data yang dibutuhkan, seperti nama, e-mail dan password.

1.5.5 Strategi dan Metode Pengujian

Strategi dan metode pengujian yang digunakan ada 2, yaitu:

- Unit testing: Pengujian fungsi, kelas, interface, maupun prosedur dari sebuah aplikasi menggunakan code untuk menentukan apakah sebuah fungsi berjalan dengan baik atau tidak.
- White box testing: Pengujian untuk memastikan detail perancangan yang telah dibuat menggunakan struktur kontrol desain program yang dibagi menjadi beberapa kasus pengujian untuk mendapatkan logika program yang benar.

1.5.6 Jadwal Pengujian

Use Case	PIC	Butir Uji	Jenis Pengujian	Jadwal pengujian
Login admin	Admin	Dapat melakukan login	unit	25 November 2019
Login user	User	Dapat	unit	25 November 2019

		melakukan login		
Create event	User	Event berhasil dibuat	unit	25 November 2019
Edit event	User	Event dapat di edit user	unit	25 November 2019
Logout	User	User berhasil melakukan logout	unit	25 November 2019
Beli tiket	User	Pembelian tiket berhasil dilakukan	unit	25 November 2019

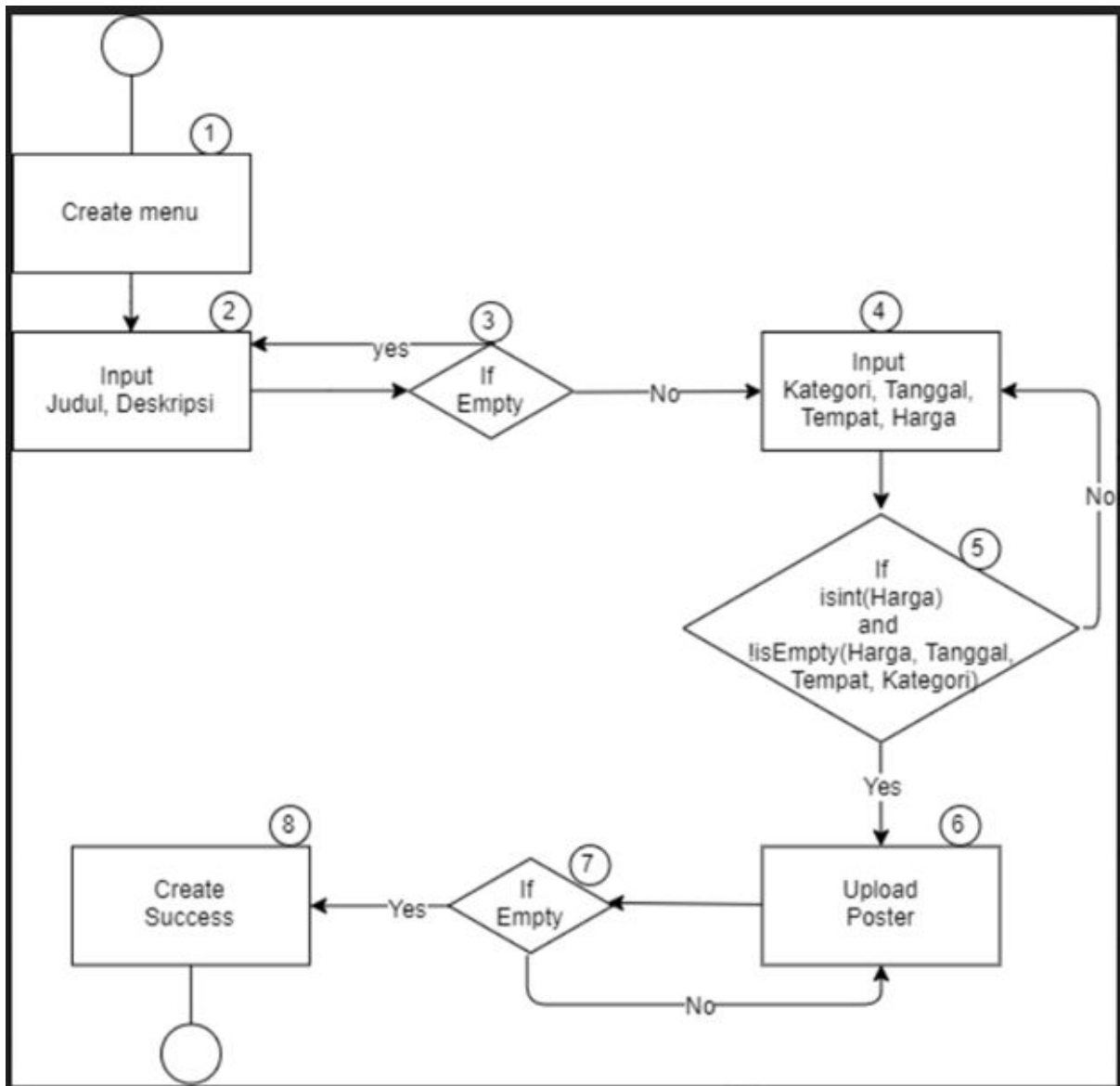
2 Pelaksanaan Pengujian

Use Case	PIC	Butir Uji	Jenis Pengujian	Jadwal pengujian
Login admin	Admin	Dapat melakukan login	unit	25 November 2019
Login user	User	Dapat melakukan login	unit	25 November 2019
Create event	User	Event berhasil dibuat	unit	25 November 2019
Edit event	User	Event dapat di edit user	unit	25 November 2019
Logout	User	User berhasil melakukan logout	unit	25 November 2019
Beli tiket	User	Pembelian tiket berhasil dilakukan	unit	25 November 2019

2.1 Pengujian UNIT

2.1.1 Pengujian White Box Method

Create Event



Gambar 1. Diagram create event

Create Event Cyclometric complexity:

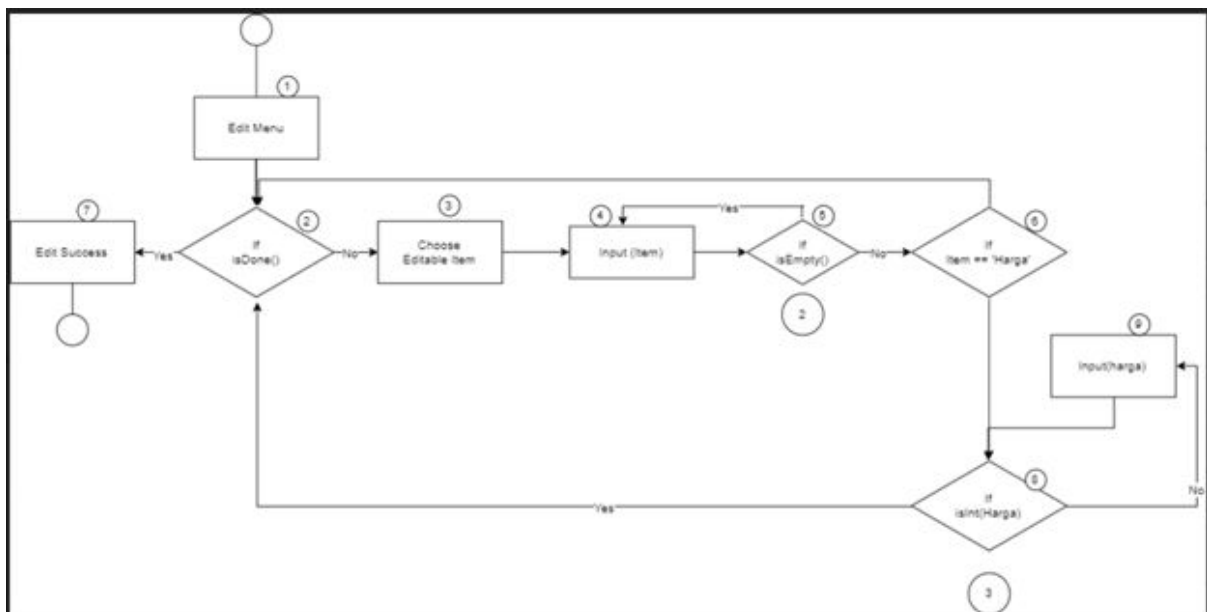
1. $3+1 = 4$

1. 1,2,3,4,5,6,7,8
2. 1,2,3,2,3,4,5,6,7,8
3. 1,2,3,4,5,4,5,6,7,8
4. 1,2,3,4,5,6,7,6,7,8

2. Test Case:

- X = 'event1', 'Final Competition Programming Telkom 2020', 'Technology', '24/06/2020', 'Telkom University Convention Hall', "", 540000, Post.img
- Path: 1,2,3,4,5,4,5,6,7,8

Edit Event



Gambar 2. Diagram edit event

Create Event Cyclometric complexity:

1. $4+1 = 5$

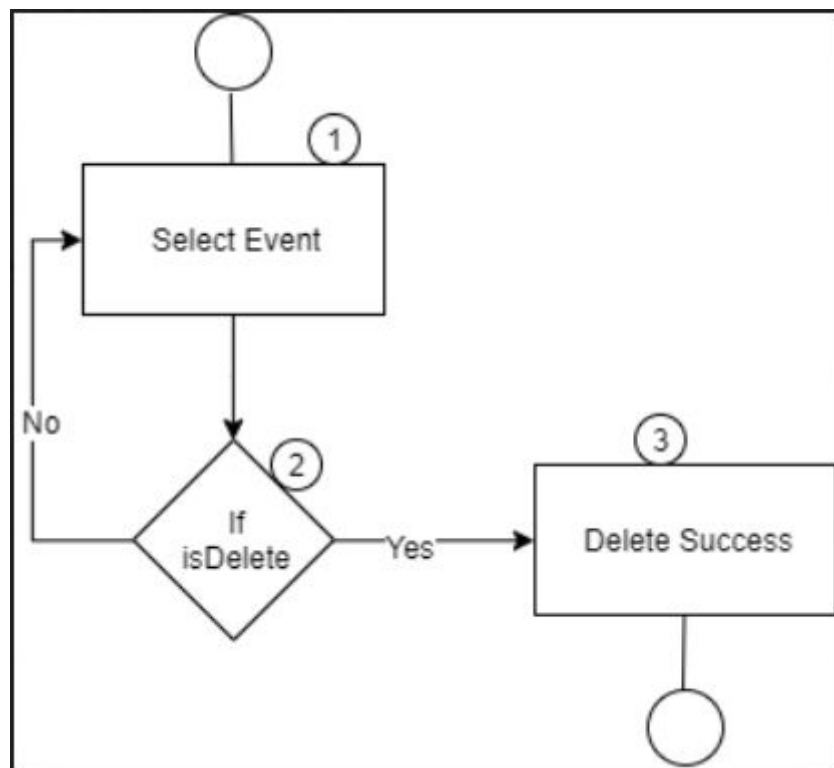
1. 1, 2, 3, 4, 5, 6, 2, 7
2. 1, 2, 3, 4, 5, 4, 5, 6, 2, 7
3. 1, 2, 3, 4, 5, 4, 5, 6, 8, 2, 7
4. 1, 2, 3, 4, 5, 6, 8, 9, 8, 2, 7

5. 1,2,3,4,5,4,5,6,8,9,8,2,7

2. Test Case:

- X = Harga(), "", Harga(), "Abjad", 320000
- Path = 1,2,3,4,5,4,5,6,8,9,8,2,7

Delete event



Gambar 3. Diagram delete event

Delete Event Cyclomatic complexity:

a. $1+1 = 2$

1. 1,2,3

2. 1,2,1,2,3

b. Test Case:

- X = Event('event1'), Delete(), 'No', Event(event2), Delete(), 'Yes'
- Path: 1,2,1,2,3

2.1.2 Pengujian Class dengan JUnit/PHPUnit

Jelaskan di sini contoh pengujian sebuah class.

Setiap methodnya perlu diuji, dengan data uji yang membuat VALID atau yang membuat FAIL.

Tabel 2 Pengujian Class

CLASS	Method	Kasus dan Hasil Uji (Data normal)			
		Data Masukan	Yang diharapkan	Pengamatan*	Kesimpulan
User	login	email : username@example.com password: secret	Masuk ke page home	Dapat melakukan login	[X] diterima [] ditolak
	register		User terdaftar di databse	Dapat melakukan registrasi	
Event	create event	judul : "Judul Event" tempat : "Lokasi event" deskripsi : "Deskripsi event" harga : 10000 jadwal: "14/11/19" sampul : "images.jpg"	Data tersimpan di database	Data berhasil disimpan	[X] diterima
	update event		Data di database terupdate berdasarkan id event	data berhasil diupdate	[] ditolak
	Hapus event		Data terhapus	data berhasil dihapus	
	view event		Data dapat terlihat oleh	Data dapat terlihat oleh user	

			user		
Ticket	Buy event view ticket	id_event	Membuat data tiket berdasarkan id event dan id user User dapat melihat data event yang telah dibeli	Data berhasil dibuat User dapat melihat data event yang telah dibeli	[X] diterima [] ditolak

* Contoh pengujian dengan JUnit/PHPUnit dilampirkan

A. Contoh Code JUnit/PHPUnit untuk pengujian Class : (Sebutkan NamaClassnya)

```
public function user_can_create_event(){
    $data = [
        'judul' => "Judul event",
        'tempat' => "Lokasi event",
        'deskripsi' => "Deskripsi event",
        'harga' => 10000,
        'jadwal' => "14/11/19",
        'sampul' => "images.jpg",
    ];
    $user = factory(\App\User::class)->create();
    $response = $this->actingAs($user, 'api')->json('POST', '/api/products',$data);
    $response->assertStatus(200);
    $response->assertJson(['status' => true]);
    $response->assertJson(['message' => "Event Created!"]);
    $response->assertJson(['data' => $data]);
}
```

```
public function user_can_update_event(){
    $response = $this->json('GET', '/api/products');
    $response->assertStatus(200);
    $product = $response->getData()[0];
    $user = factory(\App\User::class)->create();
    $update = $this->actingAs($user, 'api')->json('PATCH', '/api/products/'.$product->id,['name' => "Changed for test"]);
    $update->assertStatus(200);
    $update->assertJson(['message' => "Event Updated!"]);
}
```

```

public function user_can_see_all_event(){
    $response = $this->json('GET', '/api/products');
    $response->assertStatus(200);
    $response->assertJsonStructure(
        [
            [
                'id_event',
                'judul',
                'tempat',
                'deskripsi',
                'id_user',
                'created_at',
                'updated_at',
                'harga',
                'jadwal',
                'sampul'
            ]
        ]
    );
}

```

```

public function user_can_see_the_detail_of_event(){
    $response = $this->json('GET', '/api/products');
    $response->assertStatus(200);
    $response->assertJsonStructure(
        [
            [
                [
                    'id_event',
                    'judul',
                    'tempat',
                    'deskripsi',
                    'id_user',
                    'created_at',
                    'updated_at',
                    'harga',
                    'jadwal',
                    'sampul'
                ]
            ]
        ]
    );
}

```

```
public function testHomePage()
{
    $response = $this->call('GET', '/');
    $this->assertRegexp('/Laravel/', $response->getContent());
}
```

```
public function user_can_view_login_form() {
    $response = $this->get('/login');
    $response->assertSuccessful();
    $response->assertViewIs('auth.login');
}
```

```
public function user_can_view_register_form(){
    $response = $this->get('/register');
    $response->assertSuccessful();
    $response->assertViewIs('auth.register');
}
```

```
public function registered_user_can_login()
{
    $user = factory(User::class)->create([
        'email' => 'username@example.net',
        'password' => 'secret',
    ]);
    $response = $this->get('/login');
    $response = $this->submitForm('Login', [
        'email' => 'username@example.net',
        'password' => 'secret',
    ]);
    $this->seePageIs('/home');
}
```

```
public function logged_in_user_can_logout()
{
    $user = factory(User::class)->create([
        'email' => 'username@example.net',
        'password' => 'secret',
    ]);
    $this->actingAs($user);
    $this->visit('/home');
    $this->post('/logout');
    $this->visit('/home');
    $this->seePageIs('/login');
}
```

```

public function user_can_buy_ticket(){
    $ticket = factory(Tickets::class)->event([
        'id_event' => 'test',
    ]);
    $response = $this->belongsTo('App\Events','id_event');
    $this->seePageIs('/my-ticket');
}

```

```

public function user_can_see_ticket_they_buy(){
    $response = $this->json(['GET', '/api/products']);
    $response->assertStatus(200);
    $response->assertJsonStructure([
        [
            [
                'id_event',
                'judul',
                'tempat',
                'deskripsi',
                'id_user',
                'harga',
                'jadwal',
                'sampul'
            ]
        ]
    ]);
}

```

Gambar 4. Kumpulan unit testing

B. Screenshoot hasil pengujian JUnit

```

C:\xampp\htdocs\Stevent-master\Stevent-master>"vendor/bin/phpunit"
PHPUnit 8.3.5 by Sebastian Bergmann and contributors.

.....                                                    11 / 11 (100%)

Time: 438 ms, Memory: 20.00 MB

OK (11 tests, 11 assertions)

```

Gambar 5. Hasil unit testing

2.2 Kesimpulan Pengujian

(Berisi laporan dari pengujian yang telah dilakukan, dengan menyampaikan informasi status dari setiap fungsional yang diuji apakah telah berhasil/tidak)

Kelas Uji	Butir Uji	Kesimpulan pengujian
<i>Login User</i>	Pendaftaran <i>User</i> baru (kasus Uji : Data Normal)	Diterima
	Pendaftaran <i>User</i> baru (kasus Uji : Data Salah)	Diterima
	Pengecekan <i>User</i> yang telah terdaftar (kasus Uji : Data Normal)	Diterima
	Pengecekan <i>User</i> yang telah terdaftar (kasus Uji : Data Salah)	Diterima

3 Lampiran

- A. Capture /screenshot hasil pengujian modul-modul penting
- B. Hasil pengukuran OOMetric aplikasi yang telah berhasil dibangun dengan software (tool) pengukuran OOMetric (lihat <http://www.virtualmachinery.com/jhawkmetricsclass.htm>)