

IMPLEMENTASI UNIT TESTING, INTEGRATION TESTING, SYSTEM TESTING, DAN VALIDATION TESTING PADA APLIKASI BERBASIS WEBSITE (STUDI KASUS: KAFE SATURDAYS)

Alvin Valerian, Delon Bernady, Ferry Wahyudi, Jonathan Pratama Dinatha, One Dhylan Barletyano

Teknik Informatika, Universitas Palangka Raya
Jalan Yos Sudarso Kota Palangka Raya, Indonesia
jonathanpratamad@mhs.eng.upr.ac.id

ABSTRAK

Perkembangan teknologi informasi mendorong adopsi aplikasi berbasis *website* dalam mendukung operasional bisnis, termasuk di sektor kuliner seperti Kafe *Saturdays*. Dalam implementasinya, sistem kafe ini menghadapi sejumlah permasalahan, seperti kesalahan dalam proses pemesanan, ketidaksesuaian data stok, serta gangguan saat sistem digunakan secara bersamaan. Penelitian ini bertujuan untuk menguji kualitas sistem melalui empat metode pengujian perangkat lunak, yaitu *unit testing*, *integration testing*, *system testing*, dan *validation testing*. *Unit testing* dilakukan dengan pendekatan *white box testing* untuk memastikan bahwa setiap fungsi CRUD berjalan dengan benar. Sementara itu, *integration testing*, *system testing*, dan *validation testing* menggunakan pendekatan *black box testing* untuk menguji integrasi antar modul, stabilitas sistem secara menyeluruh, dan kesesuaian fitur sistem dengan kebutuhan pengguna akhir. Proses pengujian dilakukan secara sistematis dengan skenario yang mewakili kondisi nyata penggunaan aplikasi. Hasil pengujian menunjukkan bahwa sebagian besar fitur berjalan sesuai spesifikasi, modul berfungsi dengan baik, dan data tetap konsisten. Namun, ditemukan kendala pada validasi input dan penurunan performa ketika sistem menerima beban tinggi. Penelitian ini menyimpulkan bahwa penerapan metode pengujian yang terstruktur mampu meningkatkan kualitas perangkat lunak serta memberikan pengalaman pengguna yang lebih optimal.

Kata kunci : *Unit Testing, Integration Testing, System Testing, Validation Testing, Black Box Testing, White Box Testing.*

1. PENDAHULUAN

Seiring dengan perkembangan pesat teknologi informasi, aplikasi berbasis *website* semakin banyak digunakan untuk mendukung berbagai aktivitas bisnis, termasuk di industri kuliner. Salah satu bisnis yang mengandalkan aplikasi berbasis *website* adalah Kafe *Saturdays* dengan menggunakan sistem digital untuk melakukan pemesanan, manajemen stok, dan interaksi dengan pelanggan. Agar aplikasi ini dapat berjalan dengan optimal dan memberikan pengalaman pengguna yang baik, diperlukan proses pengujian perangkat lunak yang sistematis dan menyeluruh.

Pengujian perangkat lunak (*software testing*) merupakan langkah yang krusial dalam siklus pengembangan perangkat lunak guna memastikan bahwa sistem yang dikembangkan berfungsi sesuai dengan spesifikasi yang telah ditetapkan [1]. Sebagai upaya untuk menjamin kualitas sistem, pengujian perangkat lunak menjadi aspek yang sangat penting. Pengujian ini bertujuan untuk menemukan dan memperbaiki *bug* atau kesalahan sebelum aplikasi digunakan oleh pengguna akhir [2]. Dengan pengujian yang tepat, risiko kegagalan sistem dapat diminimalkan sehingga aplikasi dapat berjalan secara efisien dan efektif.

Terdapat berbagai metode pengujian perangkat lunak yang dapat diterapkan untuk memastikan kualitas aplikasi berbasis *website*. Beberapa metode yang umum digunakan meliputi *unit testing*, *integration testing*, *system testing*, dan *validation testing* [3]. Masing-masing metode ini memiliki cakupan dan tujuan yang berbeda dalam menjamin

performa perangkat lunak agar sesuai dengan kebutuhan pengguna dan bisnis.

Kafe *Saturdays* yang menerapkan sistem digital dalam operasional bisnis menjadi aspek penting untuk meningkatkan efisiensi layanan dan kenyamanan pelanggan. Beberapa masalah teknis yang sering terjadi, seperti kesalahan dalam proses pemesanan, pelanggan yang tidak ingin menggunakan aplikasi, ketidaksesuaian data stok, serta gangguan saat mengakses sistem di jam sibuk [4]. Permasalahan ini dapat berdampak pada kepuasan pelanggan dan efektivitas operasional kafe secara keseluruhan.

Dalam upaya mengatasi permasalahan tersebut, diperlukan penerapan *unit testing* guna memastikan bahwa setiap komponen dalam sistem bekerja dengan baik. *Unit testing* merupakan salah satu metode pengujian perangkat lunak yang fokus pada pengujian bagian terkecil atau unit dari kode [5]. Pada Kafe *Saturdays*, *unit testing* diterapkan pada fitur utama seperti pengelolaan menu untuk memastikan setiap bagian dari sistem berjalan sesuai dengan fungsinya tanpa kesalahan.

Selain itu, *integration testing* juga penting untuk memastikan bahwa berbagai modul dalam sistem dapat bekerja secara harmonis. *Integration testing* meletakkan *unit-unit* pada *environment* semestinya, dan mengatur interaksi antar *unit* tersebut secara menyeluruh [6]. Pengujian ini bertujuan untuk memeriksa integrasi antara sistem kelola menu dan sistem tambah, hapus dan edit berjalan dengan baik sehingga data dapat diperbarui secara *real-time* dan

menghindari kesalahan seperti pelanggan dapat memesan menu yang tidak ada.

System testing kemudian dilakukan untuk menguji sistem secara keseluruhan, termasuk memastikan bahwa aplikasi berjalan stabil dalam berbagai skenario penggunaan. Tahap pengujian ini berguna untuk menemukan ketidaksesuaian ataupun *error* yang terjadi pada sistem [7]. Seperti dalam pengujian sistem Kafe *Saturdays*, perlu dipastikan bahwa aplikasi dapat menangani lonjakan pengelolaan menu secara bersamaan tanpa mengalami *crash* atau penurunan performa.

Terakhir, *validation testing* diperlukan untuk memastikan bahwa sistem sesuai dengan kebutuhan dan harapan pengguna. *Validation Testing* merupakan salah satu cara pengujian sistem yang berfokus yang bertujuan untuk memverifikasi bahwa tindakan yang terlihat oleh pengguna serta output yang dihasilkan oleh sistem sesuai dengan yang diharapkan dan dapat dikenali oleh pengguna [8]. Dalam hal ini pelanggan dan staf Kafe *Saturdays* dapat dilibatkan dalam pengujian guna mengidentifikasi apakah aplikasi mudah digunakan dan benar-benar membantu dalam proses operasional kafe.

Dengan menerapkan metode pengujian yang ada, diharapkan aplikasi berbasis *website* Kafe *Saturdays* dapat berjalan dengan stabil, minim kesalahan, dan memberikan pengalaman terbaik bagi pengguna. Penelitian ini juga bertujuan untuk mengevaluasi efektivitas masing-masing metode pengujian dalam meningkatkan kualitas perangkat lunak, serta memberikan rekomendasi terkait praktik terbaik dalam pengujian aplikasi berbasis *website* yang serupa.

2. TINJAUAN PUSTAKA

2.1. Penelitian Terdahulu

Penelitian pertama dilakukan oleh Angelia Brigitta Maharani, Dana Sulisty Kusumo dan Nungki Selviandro dalam penelitiannya yang dilakukan pada tahun 2024. Penelitian ini bertujuan untuk menguji keandalan aplikasi Cafeasy, yang dikembangkan untuk mendukung operasional kafe dalam proses pemesanan dan pencatatan inventaris secara otomatis. Pengujian yang dilakukan mencakup *unit testing* untuk memastikan setiap *unit* dalam sistem berfungsi sesuai spesifikasi, *integration testing* untuk menguji interaksi antar modul dalam aplikasi, serta *usability testing* untuk mengevaluasi pengalaman pengguna dalam menggunakan sistem. Hasil dari penelitian ini menunjukkan bahwa seluruh *unit* yang diuji dalam *unit testing* berhasil berjalan sesuai dengan *product backlog* yang telah ditetapkan, dengan tingkat keberhasilan 100%. Pada *integration testing*, modul-modul dalam sistem telah berfungsi dan saling berinteraksi dengan baik sesuai dengan alur bisnis yang dirancang. Selain itu, *usability testing* menunjukkan bahwa aplikasi memiliki tingkat efektivitas sebesar 94% dan skor *System Usability Scale* (SUS) sebesar 81,4, yang mengindikasikan bahwa sistem memiliki tingkat kegunaan yang tinggi

dan dapat memenuhi kebutuhan pengguna dengan baik [6].

Penelitian kedua dilakukan oleh Mohammad Iqbal dalam penelitiannya yang dilakukan pada tahun 2022. Penelitian ini bertujuan untuk meningkatkan efektivitas pelaporan dan monitoring pekerjaan, terutama dalam sistem kerja *Work From Home* (WFH) dan *Work From Office* (WFO). Penelitian ini mengidentifikasi berbagai permasalahan, seperti alokasi pekerjaan yang belum terstruktur, penyusunan jadwal yang masih manual, serta keterlambatan dalam pelaporan tugas harian. Untuk mengatasi masalah tersebut, dikembangkan sistem informasi berbasis web menggunakan metode *Waterfall* dan pemodelan *Unified Modelling Language* (UML), dengan fitur utama berupa *pengelolaan* alokasi pekerjaan, penyusunan jadwal otomatis, pelaporan tugas harian dengan batas waktu tertentu, serta monitoring laporan pegawai yang dilengkapi dengan notifikasi keterlambatan. Hasil pengujian menunjukkan bahwa sistem ini memiliki tingkat keberhasilan 100% berdasarkan *Black Box Testing*, serta memperoleh tingkat kepuasan pengguna yang tinggi melalui *User Acceptance Test* (UAT) dengan persentase 96,52% untuk kepala seksi, 80,55% untuk pegawai, dan 100% untuk kepala sub bagian tata usaha, sehingga sistem ini dinilai efektif dalam meningkatkan pengelolaan tugas dan monitoring pekerjaan di BPS Kabupaten Subang [9].

Penelitian ketiga dilakukan oleh Iqbal Fahrozi, Anzal Fadly Hadi Pratama, Yusri Nuraeni, dan Rendi Pratama Juniar dalam penelitiannya yang dilakukan pada tahun 2023. Penelitian ini bertujuan untuk menguji fungsionalitas aplikasi *Action & Strategy*, yang dikembangkan sebagai *media sharing knowledge* bagi anggota Brimob guna meningkatkan mutu dan kualitas personel. Pengujian dilakukan menggunakan metode *Black Box Testing* untuk mengevaluasi apakah fitur dalam aplikasi telah berjalan sesuai dengan kebutuhan pengguna serta memastikan tidak ada kesalahan dalam sistem. Pengujian dilakukan dalam dua tahap, yaitu pengujian fungsi aplikasi *Android* oleh sejumlah pengguna secara simultan dan pengujian fungsi web *admin* oleh seorang pengguna. Metode penelitian yang digunakan adalah metode kuantitatif dengan skala *Guttman*, di mana hasil pengujian menunjukkan bahwa semua fitur yang diuji, baik pada aplikasi *Android* maupun sistem web, berhasil berfungsi sesuai spesifikasi tanpa ditemukan kesalahan atau *bug*. Berdasarkan hasil pengujian *Black Box Testing*, aplikasi ini dinyatakan berjalan dengan baik, dan untuk menjaga stabilitas fungsinya, diperlukan prosedur penggunaan serta batasan akses pengguna yang jelas [10].

2.2. Website

Website merupakan kumpulan halaman web yang saling terhubung dan berada dalam satu *domain* atau *subdomain* di internet, dikembangkan pertama kali oleh Sir Timothy John "Tim" Berners-Lee sebagai bagian dari sistem *World Wide Web* (WWW).

Halaman-halaman tersebut ditulis dalam format *Hyper Text Markup Language* (HTML) dan diakses melalui protokol *Hypertext Transfer Protocol* (HTTP) menggunakan *browser*. *Website* dapat bersifat statis maupun dinamis, serta menyajikan berbagai konten seperti teks, gambar, video, hingga elemen interaktif. Fungsinya meliputi penyampaian informasi, layanan digital, promosi, hingga interaksi pengguna. Setiap *website* memiliki alamat unik yang disebut URL (*Uniform Resource Locator*) yang memungkinkan pengguna untuk mengakses informasi dari berbagai sumber secara mudah dan cepat melalui jaringan internet. Sebagai salah satu pilar utama dalam transformasi digital, *website* berperan penting dalam mendukung komunikasi, transaksi, dan integrasi layanan di berbagai sektor modern [11], [12].

2.3. Kafe

Kafe merupakan tempat yang menyediakan berbagai jenis minuman, terutama kopi, serta makanan ringan. Secara etimologi, istilah "*café*" berasal dari bahasa Prancis yang berarti kopi, yang kemudian diadaptasi dalam bahasa Indonesia menjadi "kafe". Kafe memiliki karakteristik yang menyerupai bar atau restoran tetapi berbeda dengan kafetaria. Beberapa kafe hanya menyediakan minuman dan makanan ringan, sementara yang lain menawarkan menu makanan utama, lebih mirip restoran. Selain untuk menikmati makanan dan minuman, kafe juga berfungsi sebagai tempat berkumpul, berdiskusi, atau bersantai. Umumnya, kafe fokus pada penjualan kopi, teh, dan makanan ringan seperti kue dan *sandwich*, serta tidak menyediakan minuman beralkohol [13].

2.4. Pengujian Perangkat Lunak

Pengujian perangkat lunak bertujuan untuk mengevaluasi apakah suatu sistem berfungsi sesuai dengan spesifikasi yang telah ditetapkan serta mengidentifikasi kesalahan atau cacat yang mungkin terjadi. Selain memastikan perangkat lunak berjalan sesuai harapan, pengujian juga memberikan gambaran mengenai kualitasnya secara keseluruhan dan berperan dalam manajemen risiko, terutama dalam lingkungan bisnis. Tujuan utama pengujian adalah menemukan serta memperbaiki *error*, *failure*, atau *bug* yang belum terdeteksi agar tidak berdampak negatif pada kinerja sistem. Oleh karena itu, pengembang dan tim penguji perlu mengalokasikan waktu khusus untuk melakukan pengujian secara sistematis dalam berbagai kondisi guna memastikan perangkat lunak dapat berjalan dengan baik di lingkungan yang sesuai [6], [14].

Dalam proses pengujian perangkat lunak, terdapat tiga aspek utama yang menjadi perhatian, yaitu [6] :

a. *Verification*, yaitu proses pengecekan atau pengujian perangkat lunak agar konsisten dengan spesifikasi yang telah ditetapkan. Evaluasi dilakukan dengan membandingkan hasil akhir sistem dengan persyaratan yang sebelumnya telah didefinisikan.

- b. *Error Finding*, yaitu pengujian yang disengaja untuk menemukan kesalahan dengan menempatkan perangkat lunak dalam berbagai kondisi tertentu untuk memastikan bahwa sistem dapat merespons secara benar terhadap masukan yang diberikan.
- c. *Validation*, yaitu proses untuk memastikan bahwa perangkat lunak telah sesuai dengan kebutuhan dan harapan pengguna berdasarkan spesifikasi yang telah ditentukan.

Spesifikasi perangkat lunak sangat penting dalam pengujian karena mendefinisikan fungsi dan perilaku yang harus dijalankan dengan benar. Jika terdapat penyimpangan dari spesifikasi yang ditetapkan, kesalahan tersebut bisa langsung terdeteksi dan diperbaiki. Kesalahan dalam perangkat lunak, yang dikenal sebagai *failure*, yang sering kali disebabkan oleh *bug* dalam kode sumber. Oleh karena itu, tim pengembang harus mampu mendiagnosis dan mengatasi bug tersebut untuk meningkatkan keandalan dan kualitas perangkat lunak sebelum diterapkan di lingkungan produksi. [6].

2.5. Black Box Testing

Black Box Testing adalah metode pengujian perangkat lunak yang berfokus pada validasi fungsionalitas suatu aplikasi tanpa memerlukan pemahaman tentang kode atau struktur internalnya. Metode ini mengevaluasi perangkat lunak berdasarkan input dan output yang diharapkan untuk memastikan bahwa sistem telah memenuhi persyaratan yang ditentukan [15]. *Black Box Testing* dilakukan untuk menemukan kesalahan perangkat lunak agar dapat diperbaiki sebelum digunakan secara luas. Pengujian ini sangat berguna dalam mengevaluasi antarmuka pengguna dan interaksi sistem karena berfokus pada hasil akhir dari eksekusi aplikasi [16].

Keuntungan utama dari *Black Box Testing* adalah kemampuannya dalam mendeteksi kesalahan fungsional tanpa perlu memahami struktur kode sumber. Namun, metode ini memiliki keterbatasan, seperti ketidakmampuannya dalam mengidentifikasi kesalahan pada tingkat kode, sehingga lebih efektif jika digabungkan dengan metode pengujian lain untuk memastikan evaluasi yang lebih menyeluruh [15]. Terdapat beberapa teknik dalam *Black Box Testing* yang digunakan untuk memastikan fungsionalitas aplikasi telah sesuai dengan spesifikasinya, antara lain [16] :

- a. Teknik *Equivalence Partitioning*, Membagi input data menjadi beberapa partisi atau kelompok yang setara untuk mengurangi jumlah kasus uji.
- b. Teknik *Boundary Value Analysis*, Berfokus pada pengujian batas nilai minimum dan maksimum untuk menemukan kesalahan yang terjadi di sekitar batas input.
- c. Teknik *Fuzzing*, Mendeteksi *bug* atau gangguan dalam perangkat lunak dengan menggunakan data yang tidak valid atau acak.

- d. Teknik *Cause-Effect Graph*, menggunakan grafik untuk menggambarkan hubungan antara penyebab dan efek dalam perangkat lunak.
- e. Teknik *Orthogonal Array Testing*, digunakan ketika domain input kecil tetapi kompleks, terutama untuk pengujian berskala besar.
- f. Teknik *All Pair Testing*, menghasilkan kombinasi pasangan kasus uji yang mencakup setiap kemungkinan kombinasi dari parameter input.

2.6. White Box Testing

White Box Testing merupakan metode pengujian perangkat lunak yang memerlukan pemahaman terhadap struktur internal dan implementasi sistem yang diuji. Pengujian ini sangat efektif dalam mendeteksi serta mengatasi permasalahan karena *bug* dapat ditemukan lebih cepat sebelum menyebabkan gangguan pada sistem. Dalam *White Box Testing*, *test case* dikembangkan berdasarkan analisis struktur kode dengan mempertimbangkan berbagai aspek, seperti *code coverage*, *branch coverage*, *path coverage*, dan *condition coverage*. Teknik-teknik utama dalam *White Box Testing* meliputi [6] :

a. Statement Coverage

Teknik ini mengukur persentase pernyataan (*statement*) dalam kode yang dieksekusi oleh *test case*. Jika nilai *statement coverage* kurang dari 100%, maka terdapat bagian kode yang belum diuji dan berpotensi menyebabkan kesalahan saat sistem berjalan.

b. Branch Coverage

Teknik ini dikenal juga sebagai *decision coverage*, teknik ini memastikan bahwa setiap percabangan dalam kode (*decision points*) telah dieksekusi setidaknya satu kali. *Branch coverage* bertujuan untuk memverifikasi bahwa semua kemungkinan jalur dalam alur kontrol kode telah diuji. Jika persentase *branch coverage* mencapai 100%, maka seluruh kontrol alur program telah diuji dengan baik.

c. Path Coverage

Teknik ini memastikan bahwa semua jalur eksekusi yang mungkin terjadi dalam kode telah diuji setidaknya satu kali. *Path testing* digunakan untuk mengidentifikasi semua kemungkinan jalur eksekusi dalam program guna menemukan potensi kesalahan dalam struktur kode yang kompleks.

2.7. Skenario Pengujian

Skenario pengujian merupakan serangkaian tindakan atau prosedur yang digunakan untuk menguji kinerja, fungsionalitas, dan keamanan perangkat lunak secara sistematis. Skenario ini dirancang agar dapat menemukan kesalahan dengan penggunaan waktu dan tenaga seminimal mungkin, sehingga tujuan pengujian dapat tercapai dengan baik [16]. Desain skenario pengujian dapat dibuat dengan berbagai teknik, di antaranya [16] :

- a. *White Box Testing*, yaitu metode pengujian yang memastikan perangkat lunak bekerja sesuai dengan spesifikasinya dengan menguji struktur internal, alur logika, dan kode sumbernya.
- b. *Black Box Testing*, yaitu metode pengujian yang memastikan bahwa fungsi perangkat lunak berjalan dengan benar sesuai dengan kebutuhan fungsional yang telah didefinisikan, tanpa memeriksa kode sumbernya.

2.8. Unit Testing

Unit testing merupakan sebuah praktik dalam sebuah pengembangan perangkat lunak, dimana pengujian ini bertujuan untuk memvalidasi bahwa sebuah unit dari sebuah kode akan berjalan sebagaimana mestinya dan kode yang diuji juga akan menghasilkan hasil yang sama setiap kali kode tersebut dieksekusi. Sebuah unit adalah bagian terkecil dari sebuah *software* yang dapat diuji, biasanya hanya memiliki satu atau beberapa input dan menghasilkan satu output. *Unit testing* memberikan banyak kelebihan, termasuk di dalamnya adalah menemukan *bug* lebih awal, mempercepat proses pengembangan perangkat lunak, mencegah regresi *bug*, serta memberikan pemahaman dari kode program [6].

2.9. Integration Testing

Integration testing merupakan tahap selanjutnya dari proses pengujian setelah *unit testing* selesai dilakukan. Setelah unit yang diuji perindividu berjalan dengan baik, ketika diuji dengan menggabungkan dua unit atau lebih yang telah diintegrasikan kebanyakan tidak berjalan dikarenakan perbedaan tipe *error* yang ditemukan pada perbedaan *level* pengujian. Tujuan dari *integration testing* adalah untuk meletakkan *unit-unit* pada *environment* semestinya, dan mengatur interaksi antar *unit* tersebut secara menyeluruh [6].

2.10. System Testing

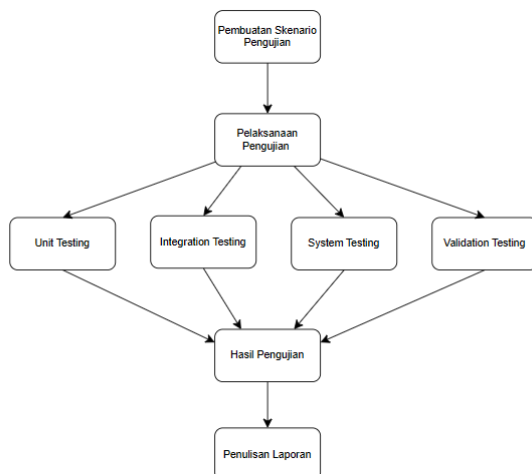
System testing merupakan salah satu metode pengujian dari *Black Box Testing* dan merupakan tahap pengujian perangkat lunak yang dilakukan pada sistem yang sudah sepenuhnya terintegrasikan untuk memastikan bahwa sistem memenuhi semua persyaratan yang ditetapkan. Pengujian ini berfokus pada fungsionalitas untuk memastikan setiap fitur dalam sistem berfungsi sesuai dengan spesifikasi awal. Selain itu, pengujian ini mencakup non-fungsionalitas sistem, yang berfokus pada performa, keamanan, dan kegunaan sistem. Tujuan utama *system testing* adalah mengenali kesalahan yang mungkin terlewat pada tahap sebelumnya, seperti *unit testing* atau *integration testing*. *System testing* memungkinkan pengembang sistem untuk memastikan bahwa sistem yang diproduksi memenuhi persyaratan pengguna dan bebas dari *bug* atau *error* yang dapat mempengaruhi kinerja sistem [17].

2.11. Validation Testing

Validation Testing merupakan metode pengujian sistem yang berfokus pada validasi aksi-aksi yang dapat diamati oleh pengguna serta hasil keluaran sistem yang dikenali oleh pengguna. Pengujian ini bertujuan untuk memastikan bahwa fitur-fitur yang disediakan oleh sistem berfungsi sesuai dengan harapan dan kebutuhan pengguna. Validasi dilakukan melalui serangkaian skenario pengujian untuk mengevaluasi kesesuaian fitur sistem dengan spesifikasi yang telah ditetapkan. Dalam proses ini, digunakan *use case scenario* sebagai dasar pengujian karena skenario tersebut mencakup berbagai interaksi yang mungkin terjadi dalam sistem, yang membantu memastikan keakuratan dan keandalan fungsionalitas yang diuji [8].

3. METODE PENELITIAN

Pada metodologi penelitian ini, diuraikan serangkaian langkah-langkah yang telah dilaksanakan dalam proses penelitian. Metode penelitian yang digunakan dalam pengujian sistem ini dapat dilihat pada Gambar 1.



Gambar 1. Alur penelitian

3.1. Pembuatan Skenario Pengujian

Pada tahap ini, dibuat rencana atau skenario pengujian yang mencakup berbagai aspek fungsionalitas dan non-fungsionalitas sistem. Skenario ini dirancang untuk menguji apakah fitur CRUD pada tabel menu dalam *website* sudah berjalan sesuai spesifikasi atau belum.

3.2. Pelaksanaan Pengujian

Setelah skenario pengujian dibuat, eksekusi pengujian dilakukan untuk memastikan kualitas sistem. *Unit Testing* dengan *White Box Testing* menguji fungsi CRUD pada tabel menu, sedangkan *Integration Testing* dengan *Black Box Testing* menguji interaksi antar modul. *System Testing* menguji keseluruhan sistem dalam berbagai skenario, dan *Validation Testing* melibatkan pengguna akhir untuk menilai kemudahan penggunaan serta keakuratan data.

3.3. Hasil Pengujian

Setelah semua pengujian dilakukan, hasilnya dikumpulkan dan dianalisis. Hasil yang diperoleh digunakan untuk menentukan apakah sistem berjalan sesuai harapan atau perlu perbaikan lebih lanjut.

3.4. Penulisan Laporan

Tahap terakhir adalah dokumentasi hasil pengujian dalam bentuk laporan. Laporan ini mencakup metode pengujian yang digunakan, hasil pengujian, perbaikan yang dilakukan, serta kesimpulan dari proses pengujian.

4. HASIL DAN PEMBAHASAN

4.1. Unit Testing

Unit testing dengan *White Box Testing* diterapkan untuk menguji setiap fungsi dalam fitur CRUD pada tabel menu sebelum diintegrasikan dengan modul lain. Pengujian ini memastikan operasi seperti *Create*, *Read*, *Update*, dan *Delete* berjalan sesuai spesifikasi. *PHPUnit* digunakan sebagai *framework* pengujian *unit* di PHP untuk menguji berbagai skenario, termasuk penambahan, pembaruan, penghapusan, dan pengambilan data menu.

```

1 <?php
2
3 use PHPUnit\Framework\TestCase;
4
5 class MenuTest extends TestCase {
6     private $db;
7
8     protected function setUp(): void {
9         // Koneksi ke database
10        $this->db = new mysqli("localhost", "root", "", "kafe_saturdays");
11
12        if ($this->db->connect_error) {
13            die("Koneksi database gagal: " . $this->db->connect_error);
14        }
15    }
16
17    /**Test Query Mengambil Data dari Tabel menu**
18    public function testFetchMenu() {
19        $query = "SELECT * FROM menu";
20        $result = $this->db->query($query);
21
22        $this->assertNotFalse($result, "Query gagal dijalankan");
23        $this->assertGreaterThan(0, $result->num_rows, "Data menu kosong");
24    }
  
```

Gambar 2. Inisialisasi *database* dan pengujian *query* pengambilan data

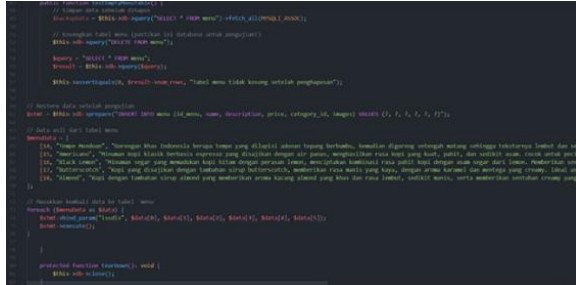
Kode pada Gambar 2 menghubungkan pengujian ke *database* MySQL dengan *mysqli*. Jika koneksi gagal, program menampilkan pesan kesalahan dan berhenti. Fungsi *testFetchMenu* menguji aksesibilitas tabel menu dan keberadaan data. Jika tabel kosong atau terjadi kesalahan *query*, pengujian dinyatakan gagal.

```

1 // **Test format data yang diambil**
2 public function testMenuDataFormat() {
3     $query = "SELECT * FROM menu LIMIT 1";
4     $result = $this->db->query($query);
5     $menu = $result->fetch_assoc();
6
7     $this->assertArrayHasKey('id_menu', $menu, "Kolom id_menu tidak ditemukan");
8     $this->assertArrayHasKey('name', $menu, "Kolom name tidak ditemukan");
9     $this->assertArrayHasKey('description', $menu, "Kolom description tidak ditemukan");
10    $this->assertArrayHasKey('price', $menu, "Kolom price tidak ditemukan");
11    $this->assertArrayHasKey('category_id', $menu, "Kolom category_id tidak ditemukan");
12    $this->assertArrayHasKey('images', $menu, "Kolom images tidak ditemukan");
13
14    $this->assertIsString($menu['name'], "Nama menu harus berupa string");
15    $this->assertIsString($menu['description'], "Deskripsi menu harus berupa string");
16    $this->assertIsNumeric($menu['price'], "Harga menu harus berupa angka");
17    $this->assertIsNumeric($menu['category_id'], "ID kategori harus berupa angka");
18    $this->assertIsString($menu['images'], "Nama file gambar harus berupa string");
19 }
  
```

Gambar 3. Pengujian format data tabel menu

Kode pada Gambar 3 menguji kesesuaian format data dari tabel menu. Tahap pertama mengambil satu data menu, tahap kedua memastikan kolom sesuai dengan struktur tabel, dan tahap ketiga memverifikasi tipe data: *name* dan *description* sebagai *string*, *price* dan *category_id* sebagai angka, serta *images* sebagai *string* (nama file gambar).



Gambar 4. Pengujian crud data tabel menu

Kode pada Gambar 4 menjalankan beberapa operasi pengujian. Pertama, data awal disimpan dengan mengambil seluruh isi tabel menu. Selanjutnya, tabel dikosongkan dan diverifikasi untuk memastikan penghapusan berhasil. Setelah pengujian, data yang telah disimpan sebelumnya dikembalikan ke tabel, mencakup beberapa *item* menu seperti "Tempe Mendoan", "Americano", dan lainnya, lengkap dengan deskripsi, harga, kategori, dan gambar. Setiap data

dimasukkan secara dinamis sebelum dieksekusi. Terakhir, fungsi penutup memastikan koneksi *database* ditutup setelah pengujian selesai.



Gambar 5. Hasil eksekusi unit testing

Pada gambar 5 hasil pengujian pada terminal menunjukkan tiga pengujian dengan total 14 asersi berhasil dijalankan tanpa kesalahan. Indikator 100% menandakan semua *test case* lolos. Informasi ini menandakan bahwa fitur CRUD dalam tabel menu berfungsi dengan baik tanpa adanya *error*.

4.2. Integration Testing

Integration Testing dilakukan setelah *unit testing* dilakukan, pengujian ini dilakukan dengan *Black Box* testing yang menguji fitur CRUD (*Create*, *Read*, *Update*, dan *Delete*) pada tabel menu dalam sistem kelola menu di Kafe Saturdays dengan tujuan untuk memastikan apakah fitur sudah terintegrasi dengan benar pada sistem baik melakukan berbagai skenario, termasuk penambahan, pembaruan, penghapusan, dan pengambilan data menu.

Tabel 1. Integration testing menu

Modul	Deskripsi	Hasil yang Diharapkan	Status Pass/ Fail
Halaman Menu	Menampilkan Halaman Menu	Menu Ditampilkan Dari Database	Pass
Menambah Menu	Menuju Site Penambahan data	Halaman Ditampilkan	Pass
	Masukan Nama Menu	Nama Menu Berhasil Diinput	Pass
	Masukan Deskripsi Menu	Deskripsi Menu Berhasil Diinput	Pass
	Masukan Harga Menu	Harga Menu Berhasil Diinput	Pass
	Pilih Kategori Menu	Kategori Berhasil Dipilih	Pass
	Pilih dan Unggah Gambar Menu	Gambar Berhasil Dipilih dan Diunggah	Pass
	Klik Tombol Tambah Menu	Menu Berhasil Ditambah	Pass
	Kembali ke Halaman Menu	Halaman Menu Terbuka	Pass
Menyunting Menu	Mencari Nama Menu	Menu yang dicari Ditampilkan	Pass
	Menekan Tombol Edit	Mengarahkan Pada Halaman Penyuntingan	Pass
	Memperbarui Data Menu	Data Menu Diperbarui	Pass
	Menekan Tombol Perbarui	Pembaruan Disimpan	Pass
Menghapus Menu	Menekan Tombol Hapus	Menu Dihapus	Pass

Pada Tabel 1 menunjukkan hasil pengujian *Integration Testing* pada sebagian fitur yaitu kelola menu dan didapati sudah terintegrasi sesuai dengan semestinya. Menu dapat ditampilkan dengan benar, pencarian juga menampilkan hasil yang diharapkan, penambahan menu, pembaruan menu dan penghapusan menu sudah terintegrasi dengan baik.

4.3. System Testing

Pengujian *System Testing* pada sistem CRUD Tabel Menu di *website* Kafe Saturdays bertujuan untuk memastikan admin dapat mengelola menu

dengan akurat. Pengujian mencakup proses login, dan fitur CRUD yang digunakan *admin*. Hasil yang diharapkan, sistem berfungsi dengan stabil, efisien, dan optimal. Pengujian *System Testing* pada sistem CRUD Tabel Menu di *website* Kafe Saturdays bertujuan untuk memastikan *admin* dapat mengelola menu dengan akurat. Pengujian mencakup proses *login*, dan fitur *CRUD* yang digunakan *admin*. Hasil yang diharapkan, sistem berfungsi dengan stabil, efisien, dan optimal.

Tabel 2. *System testing fungsional*

Modul	Deskripsi	Hasil yang Diharapkan	Status Pass/Fail
Login	Admin masuk di website dengan username dan password yang benar.	Proses login berhasil dan mengarahkan ke dashboard admin.	Pass
Menambah Menu	Admin menambahkan menu melalui Tabel Menu	Menu berhasil ditambahkan dan akan muncul pada dashboard	Pass
Menyunting Menu	Admin akan melakukan proses update pada menu.	Update berhasil dan data menu diperbarui.	Pass
Menghapus Menu	Admin akan menghapus menu yang ada.	Menu berhasil terhapus.	Pass

Pada Tabel 2 memperlihatkan hasil pengujian *system testing* fungsional yang menunjukkan setiap

langkah dalam alur kerja admin berjalan dengan baik dan memberikan hasil sesuai dengan yang diharapkan.

Tabel 3. *System testing non-fungsional*

Modul	Deskripsi	Hasil yang Diharapkan	Status Pass/ Fail
Security Testing	Menginput query SQL Injection ketika melakukan operasi insert dan update.	Query yang dimasukkan dikenali sebagai teks biasa dan tidak mempengaruhi database.	Pass
Usability Testing	Menguji kemudahan pengguna dalam mengoperasikan fitur CRUD melalui UI.	Pengguna berhasil mengoperasikan fitur CRUD tanpa kendala.	Pass
Load testing	menginput 50 data menu menggunakan JMeter.	Database dapat menangani insert 50 data tanpa kendala.	Pass
Stress testing	melakukan insert data menggunakan JMeter dari 100 hingga 500 data.	Database tidak mampu menangani beban insert dalam jumlah 500 data.	Fail

Pada Tabel 3 menunjukkan hasil pengujian *system testing* non-fungsional yang mencakup keamanan, kemudahan penggunaan, serta performa sistem dalam menangani berbagai skenario. Semua *testing* memberikan hasil yang sesuai kecuali *stress testing* yang mengungkapkan bahwa sistem mengalami *error* saat menangani 500 data sekaligus.

4.4. Validation Testing

Validation Testing dilakukan dengan metode *Black Box Testing* untuk memastikan fitur CRUD pada tabel menu dalam sistem Kafe Saturdays telah memenuhi kebutuhan pengguna serta sesuai dengan spesifikasi yang telah ditetapkan dalam perancangan

sistem. Pengujian ini melibatkan Admin sebagai pengguna utama yang bertugas mengelola menu pada website Kafe Saturdays.

Sebelum pengujian dilakukan, penting untuk memahami kebutuhan pengguna. Dalam pengujian ini, Admin mengharapkan fitur Kelola Menu dapat menambah, mengedit, mencari, dan menghapus menu dengan mudah serta memiliki validasi input. Beberapa aturan validasi diterapkan, seperti Nama Menu wajib diisi, Deskripsi diisi maksimal 255 karakter, Harga harus angka positif, Kategori dipilih dari daftar yang tersedia, dan Gambar hanya boleh berformat JPG, PNG, atau JPEG.

Tabel 4. *Validation testing menu*

Modul	Deskripsi	Hasil yang Diharapkan	Status Pass/Fail
Pencarian Menu	Cari menu berdasarkan nama yang ada	Menu yang dicari muncul dalam daftar hasil pencarian	Pass
	Cari menu berdasarkan nama yang tidak ada	Menu yang dicari tidak muncul dalam daftar hasil pencarian	Pass
Penambahan Menu	Tambah menu dengan data yang valid	Data berhasil ditambahkan ke database dan muncul di daftar menu	Pass
	Tambah menu tanpa mengisi kolom form	Muncul pesan error "Semua kolom wajib diisi" dan data tidak tersimpan	Pass
	Tambah menu dengan data tidak valid	Muncul pesan error yang menunjukkan kesalahan input	Fail
Pengeditan Menu	Edit nama menu	Nama menu berubah sesuai dengan input terbaru	Pass
	Edit harga menu dengan nilai negatif atau huruf	Muncul pesan error bahwa harga tidak boleh negatif atau menggunakan huruf	Fail
	Edit menu tanpa mengisi semua kolom	Muncul pesan error bahwa semua kolom wajib diisi	Pass
	Edit gambar dengan format tidak valid	Muncul pesan error "Format gambar tidak didukung. Harap unggah PNG, JPG, atau JPEG."	Fail
Penghapusan Menu	Hapus menu yang tersedia	Sebelum menghapus, sistem akan menampilkan pesan konfirmasi. Jika pengguna mengonfirmasi, menu akan dihapus dari daftar dan database	Pass

Tabel 4 menunjukkan hasil *Validation Testing*, di mana sebagian fitur telah berfungsi sesuai kebutuhan. Pencarian dan penghapusan menu berjalan baik, sementara penambahan dan pengeditan masih memiliki kekurangan dalam validasi input, seperti harga negatif atau huruf serta format gambar yang tidak sesuai tetap tersimpan.

5. KESIMPULAN DAN SARAN

Berdasarkan hasil pengujian, sistem telah memenuhi standar kualitas melalui *Unit Testing*, *Integration Testing*, *System Testing*, dan *Validation Testing*. *Unit Testing* memastikan fitur CRUD berfungsi tanpa kesalahan, sementara *Integration Testing* membuktikan interaksi antar modul berjalan baik. *System Testing* menguji stabilitas sistem, dan *Validation Testing* menunjukkan kemudahan penggunaan serta keakuratan data. Hasil pengujian mengonfirmasi bahwa sistem memiliki integritas data yang baik dan beroperasi optimal. Untuk penelitian selanjutnya, disarankan pengujian pada skala data lebih besar serta otomatisasi pengujian guna meningkatkan efisiensi dan akurasi.

DAFTAR PUSTAKA

- [1] R. Abdillah *et al.*, “Analisis dan Pengujian Perangkat Lunak Sistem Informasi Pembayaran Sekolah dengan Metode Pengukuran Kualitas SQuaRE6 Program Studi Teknologi Rekayasa Multimedia, Politeknik Bhakti Kartini, Indonesia,” *Jurnal Penelitian Rumpun Ilmu Teknik*, vol. 4, no. 1, pp. 208–217, doi: 10.55606/juprit.v4i1.4834.
- [2] A. Samdono, A. P. Sari, and F. P. Aditiawan, “PENGUJIAN BLACK BOX PADA SISTEM INFORMASI STOK DAN PENJUALAN BERBASIS WEBSITE MENGGUNAKAN METODE EQUIVALENCE PARTITIONING (STUDI KASUS: CV. ALGANI KARYA MANDIRI),” *Jurnal Mahasiswa Teknik Informatika*, vol. 8, no. 1, pp. 880–885, Mar. 2024, doi: <https://doi.org/10.36040/jati.v8i1.8893>.
- [3] W. Wibisono and F. Baskoro, “PENGUJIAN PERANGKAT LUNAK DENGAN MENGGUNAKAN MODEL BEHAVIOUR UML,” *JUTI: Jurnal Ilmiah Teknologi Informasi*, vol. 1, no. 1, 2002, doi: 10.12962/j24068535.v1i1.a95.
- [4] Wimber Jerry Panjaitan and Feri Lupiana, “Penerapan Transformasi Digital dan Hambatannya Pada Industri Kuliner di Indonesia,” *JURNAL RISET MANAJEMEN DAN EKONOMI (JRIME)*, vol. 1, no. 2, 2023, doi: 10.54066/jrime-itb.v1i2.650.
- [5] P. Razzaq Widodo and F. Fadlillah, “PENERAPAN METODE UNIT TESTING PADA PENGUJIAN APLIKASI PETSHOP BERBASIS WEB,” *Scientica: Jurnal Ilmiah Sains dan Teknologi*, vol. 1, no. 1, p. 1, Jan. 2023, Accessed: Mar. 20, 2025. [Online]. Available: <https://jurnal.kolibi.org/index.php/scientica/article/view/4086>
- [6] A. B. Maharani, D. S. Kusumo, and N. Selviandro, “Pengimplementasian Unit Testing, Integration Testing, dan Usability Testing pada Aplikasi Cafeasy Berbasis Website (Studi Kasus: Kafe di Daerah Bandung),” *e-Proceeding of Engineering*, vol. 11, no. 4, pp. 5160–5168, Oct. 2024, Accessed: Mar. 20, 2025. [Online]. Available: <https://openlibrarypublications.telkomuniversity.ac.id/index.php/engineering/article/view/24269>
- [7] A. Kartini and S. Hidayatulloh, “Aplikasi Sistem Pemesanan Menu Pada Kafe Nami Kopiminas Dengan Menggunakan Metode Waterfall,” *Jurnal Infortech*, vol. 5, no. 2, pp. 123–132, Dec. 2023, doi: <https://doi.org/10.31294/infortech.v5i2.1723>.
- [8] F. R. Kristiono, I. Arwani, and W. Purnomo, “Pengembangan Sistem Informasi Manajemen Registrasi Siswa Pada Bimbingan Belajar Lingua Institute,” *Jurnal Pengembangan Teknologi Informasi Dan Ilmu Komputer*, vol. 4, no. 8, pp. 2722–2730, Aug. 2020, Accessed: Mar. 20, 2025. [Online]. Available: <https://j-ptiik.ub.ac.id/index.php/j-ptiik/article/view/7756>
- [9] M. Iqbal, “SISTEM INFORMASI PELAPORAN DAN MONITORING PEKERJAAN DI BADAN PUSAT STATISTIK KABUPATEN SUBANG,” *JATI (Jurnal Mahasiswa Teknik Informatika)*, vol. 6, no. 2, pp. 420–429, Aug. 2022, doi: <https://doi.org/10.36040/jati.v6i2.4704>.
- [10] I. Fahrozi, A. Fadly, H. Pratama, Y. Nuraeni, and R. Pratama Juniar, “PENGUJIAN BLACK BOX TESTING PADA APLIKASI ACTION & STRATEGY BERBASIS ANDROID DENGAN TEKNOLOGI PHONEGAP,” *OKTAL: Jurnal Ilmu Komputer Dan Sains*, vol. 2, no. 5, p. 1347, Jun. 2023, Accessed: Mar. 20, 2025. [Online]. Available: <https://journal.mediapublikasi.id/index.php/oktal/article/view/2800>
- [11] N. L. M. E. Yuniawati, P. P. G. P. Pertama, and I. G. N. N. Bagiarta, “SISTEM INFORMASI PELAYANAN CUSTOMER PADA WAHYU SERVICE ELEKTRONIK BERBASIS WEB DENGAN MENGGUNAKAN FRAMEWORK LARAVEL,” *Naratif: Jurnal Nasional Riset, Aplikasi dan Teknik Informatika*, vol. 5, no. 2, pp. 186–197, Dec. 2023, doi: <https://doi.org/10.53580/naratif.v5i2.213>.
- [12] Mario and Ariessanti Hani Dewi, “PENGEMBANGAN PLATFORM DIGITAL PENYEWAAN TENDA PERNIKAHAN PT ANDREAS PESTA DENGAN MENGGUNAKAN METODE WATERFALL,” *Journal of Computer Science and Information Technology*, vol. 2, no. 1, pp. 26–8, Nov. 2024, doi: <https://doi.org/10.70248/jcsit.v2i1.1191>.

- [13] P. D. Anggraeni and T. A. Sabrina, "ANALISIS CUSTOMER ENGAGEMENT TERHADAP KAFE DI KOTA TEGAL," *Journal of Tourism and Economic*, vol. 4, no. 1, 2021, doi: 10.36594/jtec.v4i1.106.
- [14] Nabil Fahlevi Abdi and S.R. Candra Nursari, "Pengujian black box pada Website dengan Metode Robustness Testing (Studi kasus : Eiger Adventure)," *Journal of Informatics and Advanced Computing*, vol. 3, no. 2, 2022.
- [15] M. A. M. Ar Rachman *et al.*, "PENERAPAN BLACK BOX TESTING UNTUK EVALUASI FUNGSIONALITAS WEBSITE MAGGOPLAST," *Jurnal Mahasiswa Teknik Informatika*, vol. 9, no. 1, pp. 169–176, Dec. 2024, doi: <https://doi.org/10.36040/jati.v9i1.12177>.
- [16] F. Agil Sakinah, F. Prima Aditiawan, and A. Lina Nurlaili, "PENGUJIAN PADA APLIKASI MANAJEMEN ASET MENGGUNAKAN BLACK BOX TESTING," *Jurnal Mahasiswa Teknik Informatika*, vol. 8, no. 3, pp. 2814–2823, May 2024, doi: <https://doi.org/10.36040/jati.v8i3.9524>.
- [17] D. MARDIATI and Y. SAPUTRA, "IMPLEMENTASI SISTEM INFORMASI MANAJEMEN KLINIK MENGGUNAKAN METODE BLACK BOX TESTING," *Jurnal Informatika dan Teknik Elektro Terapan*, vol. 13, no. 1, pp. 1611–1618, Jan. 2025, doi: <http://dx.doi.org/10.23960/jitet.v13i1.6015>.