

PERANCANGAN PENGUJIAN FUNGSIONAL DAN NON FUNGSIONAL APLIKASI SIAPPARA DI KABUPATEN HUMBANG HASUNDUTAN

Riyanthi Angrainy Sianturi¹, Arnaldo Marulitua Sinaga², Yohanssen Pratama³, Hotni Simatupang⁴, Julio Panjaitan⁵ dan Sandy Sihotang⁶

^{1,2,3,4,5,6} D4 Teknologi Rekayasa Perangkat Lunak, Institut Teknologi Del Laguboti, Indonesia

¹Email: riyanthi@del.ac.id

ABSTRAK

SIAPPARA adalah aplikasi yang dibangun untuk Dinas Koperasi, Perdagangan Dan Industri Kabupaten Humbang Hasundutan dalam melakukan administrasi dan penatausahaan retribusi di pasar rakyat. Aplikasi SIAPPARA terdiri dari dua platform yaitu aplikasi web yang digunakan oleh admin dan bendahara serta aplikasi *mobile* yang digunakan oleh petugas pasar. Saat ini aplikasi telah digunakan oleh petugas pasar di 12 pasar rakyat di Kabupaten Humbang Hasundutan. Untuk menjamin bahwa seluruh fungsi pada aplikasi telah berjalan sesuai dengan rancangan dan kebutuhan, maka perlu dilakukan pengujian kembali baik secara fungsional maupun secara non fungsional. Pengujian dilakukan dengan tujuan untuk menemukan ketidaksesuaian fungsi pada aplikasi yang sudah digunakan. Sebelum melakukan pengujian aplikasi, peneliti membuat rancangan pengujian mencakup pemilihan metode dan tahapan pengujian. Metode yang digunakan dalam penelitian ini adalah studi literatur dan eksperimen. Peneliti menggunakan metode studi literatur dalam mengumpulkan referensi teori dan informasi yang berhubungan dengan pengujian fungsional dan non fungsional pada aplikasi. Peneliti menggunakan metode eksperimen untuk memilih metode pengujian yang akan digunakan serta membuat perancangan pengujian untuk metode yang telah dipilih. Hasil dari penelitian ini adalah berupa rancangan skema kerja dari kegiatan pengujian yang akan dilakukan serta rancangan pengujian untuk metode pengujian *Category Based Partition*, metode pengujian *Usability Testing* dengan teknik *Heuristic Evaluation*, dan metode *Performance Testing*.

Kata kunci: Pengujian Perangkat Lunak, *Category Based Partition*, *Heuristic Evaluation*, *Performance Testing*

ABSTRACT

SIAPPARA is an application used to manage market retribution by Dinas Koperasi, Perdagangan Dan Industri kabupaten Humbang Hasundutan. The SIAPPARA application consists of two platforms, namely a web application used by the admin and treasurer and a mobile application used by market officers. The application has been used by market officers in 12 people's markets in Humbang Hasundutan. To ensure that all functions in the application have been running according to the design and requirements, it is necessary to test both functional and non functional features. Testing is carried out with the aim of finding failures in applications that have been used. Before testing the application, the researcher makes a test design including the selection of methods and stages of testing. The method used in this research is literature study and experiment. Researchers used the literature study method in collecting theoretical references and information related to functional and non functional testing of applications. Researchers use the experimental method to choose the test method to be used and make a test design for the method that has been selected. The results of this study is a design of testing activities using *Category Based Partition* testing method, the *Usability Testing* test method with the *Heuristic Evaluation* technique, and the *Performance Testing* method.

Keywords: Software Testing, *Category Based Partition*, *Heuristic Evaluation*, *Performance Testing*

1. PENDAHULUAN

Latar Belakang

Aplikasi SIAPPARA adalah aplikasi yang dibangun khusus untuk kabupaten Humbang Hasundutan. Aplikasi ini berfungsi untuk membantu petugas pasar dalam melakukan administrasi dan penatausahaan retribusi pasar. Aplikasi yang dibangun berbasis web dan *mobile*. Aplikasi berbasis web digunakan oleh admin dan bendahara, sementara aplikasi *mobile* digunakan oleh petugas pasar. Aplikasi ini digunakan di 12 pasar yang ada di kabupaten Humbang Hasundutan yang akan terintegrasi ke dalam

satu server. Aplikasi SIAPPARA digunakan oleh petugas pasar yang berumur 40 tahun ke atas, sehingga diharapkan aplikasi ini memiliki *user interface* yang *user friendly* sehingga dapat dengan mudah digunakan oleh petugas pasar.

Aplikasi SIAPPARA telah digunakan di pasar rakyat Humbang Hasundutan. Setelah penggunaan aplikasi secara langsung di pasar rakyat, aplikasi SIAPPARA belum pernah diuji kembali baik secara fungsional maupun secara non fungsional. Pengujian perlu dilakukan kembali untuk mengetahui apakah semua fungsi pada aplikasi ini sudah dapat berjalan dengan baik sesuai dengan kebutuhan pengguna, apakah aplikasi ini memberikan performa yang baik dan apakah *user interface* memberikan kemudahan bagi pengguna dalam menggunakannya.

Oleh karena itu, peneliti perlu melakukan pengujian pada aplikasi ini. Pengujian dilakukan untuk menemukan *failure* dan masalah yang terdapat pada aplikasi, memastikan semua fungsionalitas dan non fungsionalitas telah berjalan sesuai dengan *requirement* yang telah ditentukan oleh *developer* serta untuk menghilangkan *bugs* yang terdapat pada aplikasi.

Sebelum melakukan pengujian pada aplikasi, peneliti membutuhkan skema kerja dalam melaksanakan kegiatan pengujian serta rancangan pengujian yang akan digunakan. Dengan adanya skema kerja dan rancangan pengujian, peneliti akan memiliki gambaran yang jelas tentang pelaksanaan pengujian yang akan dilakukan dan peneliti akan memiliki pedoman dan panduan dalam menguji aplikasi. Tujuan dari pelaksanaan pengujian ini adalah untuk merancang skema kerja pada pelaksanaan pengujian aplikasi SIAPPARA serta untuk menghasilkan rancangan pengujian fungsional dan non fungsional yang dapat digunakan untuk menguji aplikasi SIAPPARA.

Tinjauan Pustaka

Perangkat Lunak

Perangkat lunak adalah suatu data yang diprogram sedemikian rupa dan disimpan dalam bentuk digital yang tidak terlihat secara fisik tetapi tersimpan dalam media penyimpanan komputer [1]. Perangkat lunak merupakan seluruh perintah yang digunakan untuk memproses informasi. Perangkat keras tidak akan berfungsi dengan optimal tanpa adanya keberadaan perangkat lunak [2].

Aplikasi

Aplikasi merupakan program siap pakai yang dapat digunakan untuk menjalankan perintah dari pengguna aplikasi untuk memperoleh hasil yang lebih akurat sesuai tujuan pembuatan aplikasi tersebut. Aplikasi merupakan pemecahan masalah menggunakan salah satu teknik pemrosesan data aplikasi yang biasanya berpacu pada sebuah komputasi yang diinginkan atau pemrosesan data yang diharapkan [3].

Pengujian Perangkat Lunak

Pengujian perangkat lunak merupakan salah satu teknik yang paling sering digunakan untuk melakukan verifikasi dan validasi kualitas perangkat lunak. *Software testing* adalah sekumpulan proses yang dirancang untuk memastikan kode program yang telah dirancang bekerja sesuai dengan rancangan yang telah dibuat dan tidak melakukan hal-hal lain diluar rancangan tersebut [4]. Pengujian perangkat lunak merupakan suatu prosedur untuk menjalankan program atau sistem dengan tujuan menemukan kesalahan [5].

Pengujian Fungsional

Pengujian fungsional adalah proses jaminan kualitas yang mendasarkan kasus pengujiannya pada spesifikasi komponen perangkat lunak yang diuji. Fungsi diuji dengan memberikan masukan data dan memeriksa hasil keluarannya, dan struktur program internal jarang dipertimbangkan [6]. Pengujian fungsional adalah pengujian *software* mengenai layanan yang harus diberikan oleh sistem tentang bagaimana sistem berinteraksi terhadap masukan data dan bagaimana sistem seharusnya berperilaku dalam suatu kondisi atau situasi tertentu [7].

Category Based Partition

Category Based Partition adalah pengujian berbasis spesifikasi yang dikembangkan oleh Ostrand dan Balcer untuk menghasilkan kasus uji dari spesifikasi fungsional. Metode ini efektif dalam menghasilkan pembuatan kasus uji fungsional. Tujuan dari metode ini adalah untuk menemukan semua kemungkinan pilihan diantara batasan secara lebih efektif dan menghasilkan kasus uji dengan lebih efisien.

Pengujian Non Fungsional

Pengujian non fungsional merupakan suatu pengujian yang tidak secara langsung menggambarkan layanan spesifik dalam sistem perangkat lunak, non fungsional sistem biasanya menggambarkan seberapa bagus suatu perangkat lunak atau sistem terhadap *development process*, standar dan waktu [7].

Usability Testing

Teknik *Usability Testing* merupakan salah satu cara untuk melakukan evaluasi terhadap sebuah produk [8]. *Usability Testing* adalah sebuah alat penelitian yang berakar pada metodologi eksperimental klasik [9].

Heuristic Evaluation

Heuristic Evaluation dilakukan dengan melihat suatu *interface* dan mencoba untuk mendapatkan pendapat mengenai *interface* tersebut, apakah baik atau tidak. *Heuristic Evaluation* biasanya dilakukan oleh *usability specialist* atau orang yang setidaknya memiliki keterlibatan pada proyek tersebut. Idealnya seorang ahli akan melakukan evaluasi atau ulasan berdasarkan prinsip kegunaan dari badan penelitian dan pengalaman profesional sebelumnya. Tujuan dari *Heuristic Evaluation* adalah untuk menemukan masalah *usability* dalam *user interface design* sehingga dapat dijadikan sebagai bagian dari proses desain berulang. Sudut pandangnya adalah target spesifik yang akan menggunakan produk tersebut.

Performance Testing

Performance Testing adalah jenis pengujian untuk memastikan perangkat lunak akan bekerja dengan baik di bawah beban kerja yang diharapkan. *Performance Testing* bertujuan untuk memverifikasi performa sistem secara spesifik, seperti waktu respon, ketersediaan layanan, dan jumlah halaman yang diakses. Pengujian dilakukan dengan cara simulasi oleh banyak pengguna secara serentak dengan rentang waktu yang telah ditentukan [10].

2. MATERI DAN METODE

Studi Literatur

Pada penelitian ini akan dilakukan studi literatur yaitu dengan cara mengumpulkan referensi teori dan informasi yang berhubungan dengan pengujian fungsional dan non fungsional pada aplikasi. Studi literatur yang digunakan dalam penelitian ini adalah jurnal, buku, laporan penelitian, dan bahan pustaka lainnya. Peneliti melakukan studi literatur ini dengan cara membaca, mempelajari dan mengkaji literatur-literatur yang relevan dengan metode pengujian fungsional dan non fungsional yang akan digunakan dalam pengujian aplikasi SIAPPARA. Tujuan utama peneliti melakukan studi literatur adalah untuk mencari dasar pijakan/fondasi dalam memperoleh dan membangun landasan teori, kerangka berpikir mengenai pengujian fungsional dan non fungsional, sehingga peneliti mempunyai pemahaman yang lebih luas dan mendalam terhadap topik yang diteliti.

Eksperimen

Pada penelitian ini juga menggunakan metode penelitian eksperimen dan percobaan untuk melihat hasil yang didapatkan. Metode eksperimen pada penelitian ini diterapkan dengan cara melakukan implementasi langsung serangkaian eksperimen dan pengamatan. Eksperimen dilakukan untuk memilih metode pengujian yang akan digunakan serta membuat perancangan pengujian untuk metode *Category Based Partition*, *Heuristic Evaluation*, dan *Performance Testing* yang telah dipilih. Eksperimen yang baik dilakukan untuk mendapatkan validasi terkait dengan metode yang diusulkan sehingga percobaan yang dilakukan dalam pembuatan perancangan pengujian akan memperoleh hasil yang efektif.

3. HASIL DAN PEMBAHASAN

Penelitian ini dilakukan untuk menghasilkan rancangan skema kerja dari kegiatan pengujian yang akan dilakukan dan menghasilkan rancangan pengujian untuk setiap metode yang akan digunakan untuk menguji aplikasi SIAPPARA. Pengujian yang akan dilakukan adalah pengujian fungsional dan non fungsional. Metode pengujian fungsional yang akan digunakan oleh peneliti adalah *Category Based Partition*, yaitu dengan menguji aplikasi berdasarkan kategori-kategori data masukannya.

Pada pengujian non fungsional, peneliti akan menggunakan pengujian *Performance Testing* dan juga *Usability Testing*. *Performance Testing* dilakukan untuk mengevaluasi kemampuan program untuk beroperasi dengan benar dipandang dari sisi acuan kebutuhan misalnya aliran data, ukuran pemakaian memori, kecepatan eksekusi. Pengujian ini mengevaluasi kemampuan program untuk beroperasi dengan benar dipandang dari aliran pemakaian memori [11].

Usability Testing merupakan salah satu cara untuk melakukan evaluasi sebuah produk [8]. *Usability Testing* dilakukan supaya peneliti dapat melihat dan mengetahui bagaimana *user interface* dari aplikasi SIAPPARA dan apakah *user interface* memberikan kemudahan bagi petugas pasar dalam melakukan tugas administrasi dan penatausahaan retribusi pasar di Kabupaten Humbang Hasundutan.

Sebelum melakukan pengujian, dibutuhkan rancangan pelaksanaan kegiatan pengujian yang jelas yang dapat digunakan sebagai panduan dalam melakukan pengujian. Rancangan skema kerja dalam pelaksanaan kegiatan pengujian pada Aplikasi SIAPPARA yang telah dibuat oleh peneliti adalah sebagai berikut:

1. Pengenalan Aplikasi

Pada tahap ini, peneliti akan melakukan pengenalan dan pembelajaran terhadap aplikasi SIAPPARA yang sudah dikembangkan dengan cara menjalankannya.

2. Analisis Aplikasi

Pada tahap ini, peneliti melakukan analisis terhadap aplikasi SIAPPARA yang telah dikembangkan. Hal-hal yang dianalisis oleh peneliti adalah fungsi-fungsi yang terdapat pada aplikasi web dan *mobile* SIAPPARA, alur kerja aplikasi.

3. Perancangan Pengujian

Pada tahap perancangan pengujian, peneliti akan merancang pengujian yang hendak dilakukan. Rancangan ini terdiri atas rancangan pengujian *Category Based Partition*, *Usability Testing*, dan *Performance Testing*.

4. Pengujian Tahap 1

Pada tahap Pelaksanaan pengujian, peneliti akan melakukan pelaksanaan pengujian sesuai rancangan yang telah didefinisikan sebelumnya. Peneliti akan menguji objek kaji sesuai dengan teknik pengujian yang sudah ditentukan.

5. Perbaikan Aplikasi

Setelah melakukan pengujian pada aplikasi SIAPPARA, maka peneliti akan melakukan perbaikan terhadap aplikasi sesuai dengan hasil pengujian yang sudah dilakukan sebelumnya.

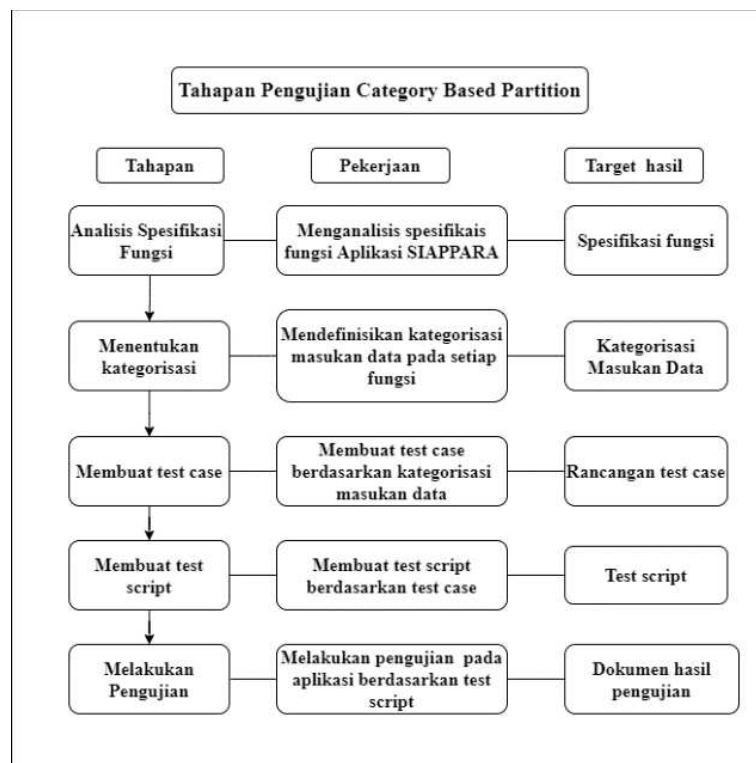
6. Pengujian Tahap 2

Setelah melakukan perbaikan aplikasi SIAPPARA, maka peneliti akan melakukan pengujian kembali. Pengujian ulang yang dilakukan adalah fungsional dan non fungsional untuk memeriksa apakah proses pengujian yang dilakukan sebelumnya sudah benar. Pengujian tahap kedua dilakukan dengan menguji aplikasi SIAPPARA menggunakan *test case* yang sama dengan yang digunakan pada pengujian tahap pertama.

Pada penelitian ini juga, peneliti telah membuat rancangan pengujian terhadap ketiga metode pengujian yang akan digunakan dalam menguji Aplikasi SIAPPARA. Rancangan untuk setiap pengujian yang akan dilakukan adalah sebagai berikut.

Rancangan Pengujian *Category Based Partition*

Rancangan pengujian *Category Based Partition* yang telah dibuat oleh peneliti dapat dilihat pada gambar 1.



Gambar 1. Hasil Rancangan Pengujian *Category Based Partition*

Penjelasan untuk rancangan pengujian *Category Based Partition* pada gambar 1 adalah:

1. Analisis spesifikasi fungsi

Pada tahap ini, peneliti akan menganalisis spesifikasi, menentukan unit fungsional yang terdapat pada aplikasi, dan menentukan parameter dari masing-masing fungsional. Salah satu contohnya adalah fungsi *Export Laporan Komoditi*.

2. Membuat Kategorisasi Masukan Data

Pada tahap ini, peneliti akan mendefinisikan kategorisasi untuk setiap masukan data pada setiap fungsi yang terdapat pada aplikasi SIAPPARA serta menentukan tipe kategori untuk parameter dari setiap masukan data. Setelah peneliti membuat partisi kategori, maka peneliti akan menggunakan kategori tersebut untuk membuat *test case*. Contoh kategorisasi masukan data pada fungsi *Export Laporan Komoditi* yang dapat dilihat pada tabel 1.

Tabel 1. Kategorisasi Masukan Data Fungsi *Export Laporan Komoditi*

Parameter	Kategori	Tipe Kategori
Tanggal awal	Tanggal, Bulan, Tahun	Enumerative
Tanggal akhir	Tanggal, Bulan, Tahun sebelum tanggal awal	Enumerative
	Tanggal, Bulan, Tahun sesudah tanggal awal	Enumerative

3. Membuat *Test Case*

Pada bagian ini peneliti akan merancang *test case* untuk setiap fungsi yang terdapat pada aplikasi SIAPPARA sesuai dengan kategorisasi masukan data yang telah dibuat sebelumnya. *Test case* akan di-generate dari kombinasi dari setiap kategori yang telah dibuat oleh peneliti. Contoh *test case* pada fungsi *Export Laporan Komoditi* dapat dilihat pada tabel 2.

Tabel 2. *Test Case* Fungsi *Export Laporan Komoditi*

No TC	Tanggal Awal	Tanggal Akhir
TC01	Tanggal, Bulan, Tahun	Tanggal, Bulan, Tahun sebelum tanggal awal
TC02	Tanggal, Bulan, Tahun	Tanggal, Bulan, Tahun sesudah tanggal awal

4. Membuat *Test Script*

Pada bagian ini peneliti akan membuat *test script* untuk setiap fungsi yang terdapat pada aplikasi SIAPPARA sesuai dengan *test case* yang telah dibuat sebelumnya. Pada skrip pengujian ini juga akan terdapat kolom untuk hasil yang diharapkan, dan hasil aktual dari pengujian yang akan dilakukan. Contoh *test script* pada fungsi *Export Laporan Komoditi* dapat dilihat pada tabel 3.

Tabel 3. *Test Script* Fungsi *Export Laporan Komoditi*

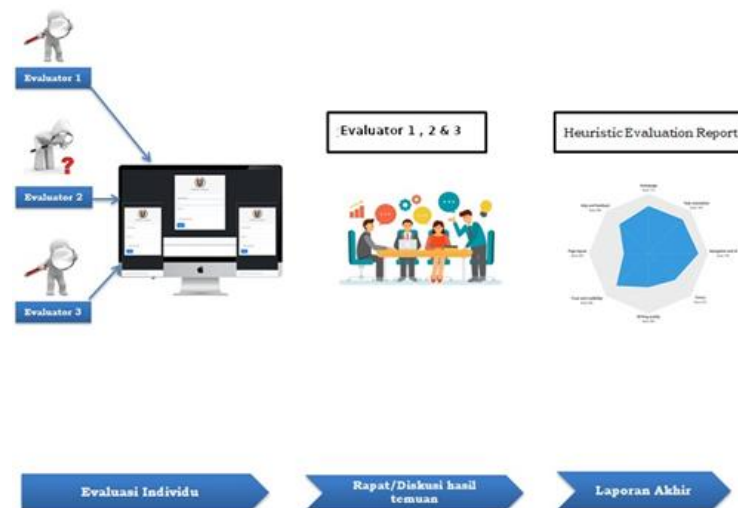
No TC	Tanggal Awal	Tanggal Akhir	Hasil yang diharapkan
TC01	20/04/2021	06/04/2021	Tidak dapat meng-export laporan komoditi, dan menampilkan message, "tanggal akhir tidak bisa lebih kecil dari tanggal awal"
TC02	06/04/2021	13/04/2021	Berhasil meng-export laporan komoditi

5. Jalankan Pengujian

Pada tahap ini, peneliti akan menjalankan pengujian terhadap aplikasi SIAPPARA sesuai dengan *test script* yang telah dibuat pada tahap keempat.

Rancangan Pengujian *Heuristic Evaluation*

Perancangan pelaksanaan pengujian *Heuristic Evaluation* yang akan dilakukan oleh peneliti dapat dilihat pada gambar 2.



Gambar 2. Hasil Rancangan Pelaksanaan Pengujian *Heuristic Evaluation*

Tahapan dalam pengujian *Heuristic Evaluation* adalah:

1. Evaluasi Individu

Evaluasi individu dilakukan oleh setiap *evaluator* untuk memeriksa *user interface* pada aplikasi SIAPPARA secara menyeluruh. Setiap *evaluator* bekerja sendiri-sendiri dan menuliskan semua temuan positif ataupun temuan negatif.

Rancangan format untuk membuat temuan positif dari aplikasi SIAPPARA berdasarkan 13 prinsip desain, dapat dilihat pada tabel 4.

Tabel 4. Rancangan Format Temuan Positif

1	<i>Visibility of System Status</i> Penjelasan:
2	<i>Match Between System and The Real World</i> Penjelasan:
3	<i>User Control and Freedom</i> Penjelasan:
4	<i>Consistency and Standard</i> Penjelasan:
5	<i>Error Prevention</i> Penjelasan:
6	<i>Recognition Rather Than Recall</i> Penjelasan:
7	<i>Flexibility and Efficiency of Use</i> Penjelasan:
8c	<i>Aesthetic and Minimalist Design</i> Penjelasan:
9	<i>Help Users Recognize, Diagnose, and Recover from Errors</i> Penjelasan:
10	<i>Help and Documentation</i> Penjelasan:
11	<i>Affordance</i> Penjelasan:
12	<i>Feedback</i> Penjelasan:
13	<i>Visibility</i> Penjelasan:

Rancangan format untuk membuat temuan negatif dari aplikasi SIAPPARA, dapat dilihat pada tabel 5.

Tabel 5. Rancangan Format Temuan Negatif

Tugas	Menjelaskan langkah-langkah yang digunakan oleh suatu <i>actor</i> tertentu, dalam melakukan suatu tugas, untuk dapat dilakukan evaluasi terkait dengan <i>usability</i> pada tugas tersebut.
Pengamatan	<i>Evaluator</i> akan menjelaskan temuan negatif yang ditemukan berdasarkan dengan pengamatan yang dilakukan
Rekomendasi	<i>Evaluator</i> akan mendefinisikan rekomendasi perbaikan terhadap temuan negatif tersebut
Kode	Membuat kode dari <i>heuristic</i> yang dilakukan contoh: H01T01/zz/02

Evaluator akan mendaftarkan semua temuan negatif yang diperoleh dengan mengisi tabel yang disediakan dengan format yang sama untuk 13 prinsip desain yang digunakan.

Kemudian, masing-masing *evaluator* akan membuat rekomendasi perbaikan aplikasi yang dibuat dalam tabel dengan menggunakan *Severity Rating* untuk pemeringkatan keseriusan (tingkat keparahan) masalah. *Severity Rating* bertujuan untuk memberikan nilai pada permasalahan *Usability* yang ditemukan selama proses evaluasi. *Severity Rating* pada bagian Temuan, diisi dengan pemberian kode angka warna berdasarkan tingkat keparahan masalah. Pola pemberian warna dapat dilihat pada tabel 6.

Tabel 6. *Severity Rating*

0	Bukan termasuk masalah <i>usability</i> Masalah ditemukan tetapi tidak termasuk masalah <i>Usability</i> .
1	<i>Cosmetic problem</i> Cukup mengganggu pengguna, tapi tidak terlalu menyebabkan masalah pada proses penyelesaian tugas-tugas. Perbaikan tidak terlalu dibutuhkan.
2	<i>Minor usability problem</i> Masalah yang berpotensi menyebabkan pengguna mengalami kesulitan dalam proses penyelesaian tugas. Dibutuhkan perbaikan, tingkat prioritas rendah.
3	<i>Major usability problem</i> Masalah yang sangat penting untuk diperbaiki, tingkat prioritas tinggi.
4	<i>Usability catastrophe</i> Masalah yang wajib diperbaiki sebelum aplikasi dapat digunakan.

2. Rapat/Diskusi hasil temuan

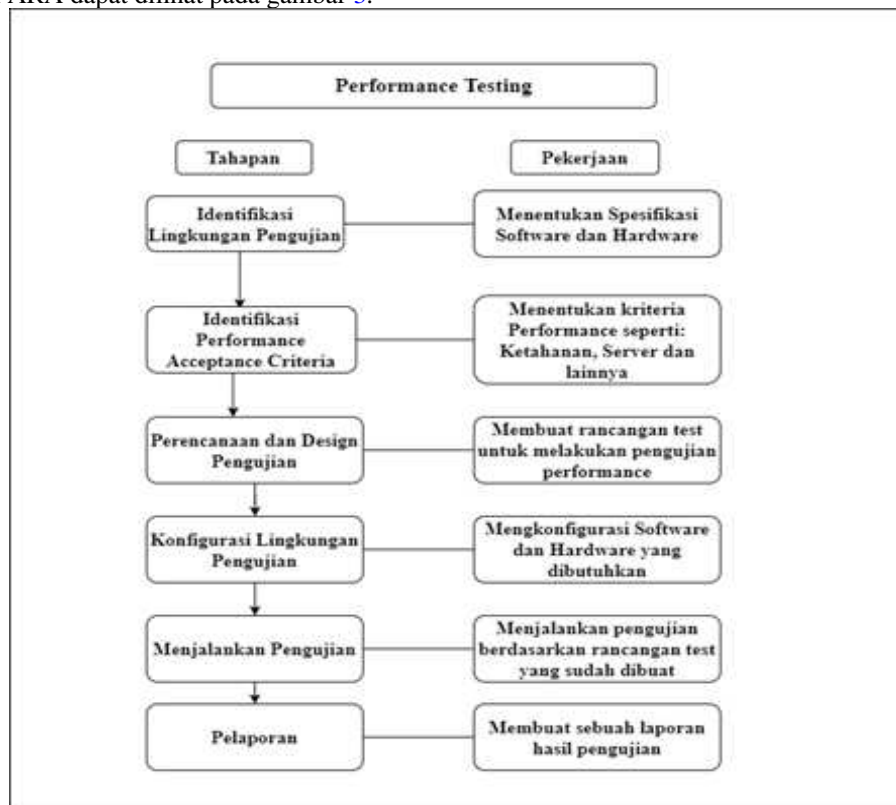
Setelah melakukan evaluasi mandiri pada aplikasi SIAPPARA, maka *evaluator* akan berkumpul kembali untuk melakukan rapat dan mendiskusikan hasil temuan masing-masing. Setiap *evaluator* akan menyampaikan temuan-temuan yang berhasil ditemukan. Setelah itu, *evaluator* akan membuat rangkuman dari keseluruhan temuan yang diperoleh.

3. Laporan Akhir

Evaluator yang adalah peneliti dalam penelitian ini akan membuat laporan akhir dari kegiatan pengujian *Heuristic Evaluation* pada aplikasi SIAPPARA yang dilakukan.

Rancangan Pengujian *Performance Testing*

Perancangan pelaksanaan pengujian *Performance Testing* yang akan dilakukan oleh peneliti pada aplikasi SIAPPARA dapat dilihat pada gambar 3.



Gambar 3. Hasil Rancangan Pelaksanaan *Performance Testing*

1. Identifikasi Lingkungan Pengujian

Berikut merupakan lingkungan pengujian yang akan digunakan oleh peneliti dalam melakukan pengujian *Performance Testing* pada aplikasi SIAPPARA.

- a. Spesifikasi *hardware* yang digunakan dalam pengujian *Performance Testing* adalah sebagai berikut:
 - Laptop : Lenovo
 - Processor : Intell, CoreTM i5 @2,50 GHz
 - Memory : 8192 MB RAM
 - Hardisk: 2 TB
- b. Spesifikasi *software* yang digunakan dalam pengujian *Performance Testing* adalah sebagai berikut:
 - Sistem Operasi: Windows 10 pro 64 Bit
 - Tools(*software* yang digunakan ketika melaksanakan pengujian): *Browser* (Google chrome), Apache JMeter.
 - DBMS: MySQL
2. Identifikasi *Performance Acceptance Criteria*
Aplikasi SIAPPARA digunakan untuk mensinkronisasikan data, sehingga diharapkan aplikasi mampu menahan beban yang diberikan serta memberikan waktu respon yang cepat. Kriteria *performance* yang akan diuji dari aplikasi SIAPPARA adalah: ketahanan server, *response time*, dan jumlah maksimum data yang dapat disinkronisasikan.
3. Perencanaan dan Desain Pengujian
Dalam melakukan *Performance Testing* peneliti menentukan apa saja yang akan diuji pada aplikasi SIAPPARA. Berikut merupakan rancangan *test case* untuk *Performance Testing* yang akan dilakukan.
 - a. Memeriksa berapa *user* yang dapat menambahkan data pasar dalam waktu satu detik secara bersamaan.
 - b. Memeriksa berapa *user* yang dapat menambahkan data tipe pasar dalam waktu satu detik secara bersamaan.
 - c. Memeriksa berapa *user* yang dapat menambahkan data jenis tempat dalam waktu satu detik secara bersamaan.
 - d. Memeriksa berapa *user* yang dapat menambahkan data biaya retribusi tempat dalam waktu satu detik secara bersamaan.
 - e. Memeriksa berapa *user* yang dapat menambahkan data komoditi dalam waktu satu detik secara bersamaan.
 - f. Memeriksa berapa banyak data *user* yang dapat ditambahkan dalam waktu satu detik secara bersamaan.
 - g. Memeriksa berapa banyak *user* yang dapat melakukan *export* laporan *Excel* dalam waktu satu detik secara bersamaan.
 - h. Memeriksa jumlah maksimum data yang dapat disinkronisasikan dalam waktu satu detik secara bersamaan.
4. Konfigurasi Lingkungan Pengujian
Peneliti akan melakukan konfigurasi lingkungan pengujian sebelum menjalankan *Performance Testing* pada aplikasi SIAPPARA. Konfigurasi dilakukan terhadap seluruh *hardware* dan *software* yang dibutuhkan selama proses pengujian berlangsung. Konfigurasi ini dilakukan untuk memastikan bahwa lingkungan pengujian sudah siap untuk digunakan.
5. Jalankan Pengujian
Peneliti akan menjalankan *Performance Testing* pada aplikasi SIAPPARA berdasarkan rancangan pengujian yang telah didefinisikan dan ditentukan.
6. Pelaporan
Setelah melakukan pengujian, maka peneliti akan membuat laporan hasil pengujian. Laporan hasil *Performance Testing* berisi jumlah maksimum *user* yang dapat mengakses aplikasi SIAPPARA secara bersamaan dalam 1 detik.

4. KESIMPULAN DAN SARAN

Berdasarkan penelitian yang dilakukan dengan judul Perancangan Pengujian Fungsional dan Non Fungsional Pada Aplikasi SIAPPARA, dapat disimpulkan bahwa peneliti berhasil membuat rancangan skema pengujian yang dapat digunakan dalam proses pengujian aplikasi SIAPPARA. Rancangan skema pengujian meliputi pengenalan aplikasi, analisis aplikasi, perancangan pengujian, pengujian tahap 1, perbaikan aplikasi, pengujian tahap 2. Peneliti juga telah membuat rancangan pengujian untuk ketiga metode pengujian yang dipilih. Rancangan pengujian untuk pengujian fungsional dengan metode *Category*

Based Partition terdiri dari kegiatan menganalisis spesifikasi fungsi, menentukan kategorisasi, membuat *test case*, membuat *test script*, dan melakukan pengujian. Rancangan pengujian non fungsional dengan metode *Heuristic Evaluation* yang telah dibuat adalah evaluasi individu, melakukan rapat untuk mendiskusikan hasil temuan, dan membuat laporan akhir pengujian. Rancangan pengujian non fungsional dengan metode *Performance Testing* adalah mengidentifikasi lingkungan pengujian, mengidentifikasi *performance acceptance criteria*, membuat perencanaan dan desain pengujian, melakukan konfigurasi lingkungan pengujian, menjalankan pengujian, serta membuat pelaporan hasil pengujian.

DAFTAR PUSTAKA

- [1] E. Ali, "Rekayasa Perangkat Lunak," vol. 205, no. 9, p. 61, 2019.
- [2] Y. I. Maulana, "Perancangan Perangkat Lunak Sistem Informasi Pendataan Guru Dan Sekolah (SINDARU) Pada Dinas Pendidikan Kota Tangerang Selatan," *J. Pilar Nusa Mandiri*, vol. 13, no. 1, pp. 21–27, 2017.
- [3] H. Abdurahman and A. R. Riswaya, "Aplikasi Pinjaman Pembayaran Secara Kredit Pada Bank Yudha BHakti," *Apl. Pinjaman Pembayaran Secara Kredit Pada Bank Yudha BHakti*, vol. 8, no. 2, pp. 61–69, 2014, [Online]. Available: <http://jurnal.stmik-mi.ac.id/index.php/jcb/article/view/114/138>.
- [4] G. J. Myers, *The Art of Software Tsting*, Third Edit., vol. 7, no. 2. Hoboken: John Wiley & Sons, Inc, 2012.
- [5] S. Nidhra, "Black Box and White Box Testing Techniques - A Literature Review," *Int. J. Embed. Syst. Appl.*, vol. 2, no. 2, pp. 29–50, 2012, doi: [10.5121/ijesa.2012.2204](https://doi.org/10.5121/ijesa.2012.2204).
- [6] H. Singh, M. Conrad, and S. Sadeghipour, "Test case design based on Z and the classification-tree method," *Proc. Int. Conf. Form. Eng. Methods, ICFEM*, pp. 81–90, doi: [10.1109/icfem.1997.630406](https://doi.org/10.1109/icfem.1997.630406).
- [7] A. C. Fatiyah, S. F. S. Gumilang, and D. Witarsyah, "Functional And Non-Functional Testing Of Borongajayuk Web Applications," vol. 44, no. 12, pp. 2–8, 2019.
- [8] Y. D. Wijaya, I. B. Kurniawan, and Z. J. Wahid, "Evaluasi Aplikasi Pencari Bakat 'APIK' Dengan Menggunakan Usability Testing," *J. IT CIDA*, vol. 4, no. 1, pp. 27–38, 2018.
- [9] J. Rubin and D. Chisnell, *Handbook of Usability Testing How to Plan, Design, and Conduct Effective Test*. Wiley Publishing, 2008.
- [10] D. Andriansyah, "Performance dan Stress Testing Dalam Mengoptimasi Website," *Comput. Based Inf. Syst. J.*, vol. 07, no. 1, pp. 23–28, 2019.
- [11] U. Hanifah, R. Alit, and S. Sugiarto, "Penggunaan Metode Black Box Pada Pengujian Sistem Informasi Surat Keluar Masuk," *SCAN - J. Teknol. Inf. dan Komun.*, vol. 11, no. 2, pp. 33–40, 2016, [Online]. Available: <http://ejournal.upnjatim.ac.id/index.php/scan/article/view/643>.