

# PENGUJIAN *BLACKBOX* APLIKASI *FORECASTING FTTH* BERBASIS *WEBSITE* (STUDI KASUS PADA PT YOFC INTERNATIONAL INDONESIA)

Moch Viqqy Fajar Aditya<sup>1\*</sup>, Agus Satrio Utomo<sup>1</sup>, Yusril Sadhanayoga<sup>1</sup>, Aries Saifudin<sup>1</sup>

<sup>1</sup>Fakultas Ilmu Komputer, Teknik Informatika, Universitas Pamulang, Jl. Raya Puspiptek No. 46, Kel. Buaran, Kec. Serpong, Kota Tangerang Selatan. Banten 15310, Indonesia

Email: <sup>1\*</sup>[fajaraaditya@gmail.com](mailto:fajaraaditya@gmail.com), <sup>2</sup>[satrio099u@gmail.com](mailto:satrio099u@gmail.com),

<sup>3</sup>[yusrilsadhanayoga15122000@gmail.com](mailto:yusrilsadhanayoga15122000@gmail.com), <sup>4</sup>[aries.saifudin@unpam.ac.id](mailto:aries.saifudin@unpam.ac.id)

(\* : coresponding author)

**Abstrak**– Aplikasi berbasis *website, desktop* ataupun *mobile* sebelum digunakan ke khalayak umum ataupun perusahaan tertentu harus melewati pengujian terhadap *software* atau perangkat lunaknya. Ada dua metode dalam pengujian yang biasa kita ketahui yaitu *Black Box Testing* dan *White Box Testing*. Untuk kali ini metode pengujian yang kita pilih adalah *Black Box Testing* dengan Teknik *Equivalence Partitions* bertujuan untuk menguji apakah ada kesalahan pada: fungsi sistem, *Interface*, struktur data atau akses data, performansi, inialisasi. Objek yang dilakukan dalam pengujian kali ini adalah aplikasi *Forecasting Material FTTH* Berbasis *Website*. Metode dan Teknik yang dipilih dalam pengujian ini adalah Teknik yang mudah dilakukan bagi *tester* pemula. Form yang diuji ada 4 yaitu *form login user* dan *admin*, *form* produk, *form* kategori, *form* proyek. Setiap *form* akan diuji dengan memasukan data input yang sudah biasa ditemukan dalam pengujian yang ada. Hasil penelitian menunjukkan bahwa pengujian *software* menggunakan metode *Black Box Testing* dengan Teknik *Equivalence Partitions* pada aplikasi masih ada *form* yang hasil atau output tidak sesuai dengan yang diharapkan dalam perancangan aplikasi tersebut karena terjadi error. Hasil perhitungan validitas system diperoleh nilai 100%.

**Kata Kunci:** *Blackbox Testing, Forecasting, Material, FTTH, Website*

**Abstract**– *Web-based, desktop or mobile applications before being used to the general public or certain companies must pass testing on the software. There are two methods in testing that we usually know, namely Black Box Testing and White Box Testing. For this time the testing method we chose was Black Box Testing with the Equivalence Partitions Technique which aims to test whether there are errors in: system functions, interfaces, data structures or data access, performance, initialization. The object carried out in this test is the Website-Based FTTH Material Forecasting application. The methods and techniques chosen in this test are techniques that are easy to do for novice testers. There are 4 forms tested, namely user and admin login forms, product forms, category forms, project forms. Each form will be tested by entering input data that is commonly found in existing tests. The results showed that software testing using the Black Box Testing method with the Equivalence Partitions Technique in the application there are still forms whose results or outputs do not match what is expected in the design of the application because an error occurs. The results of the calculation of system validity obtained a value of 100%.*

**Keywords:** *Blackbox Testing, Forecasting, Material, FTTH, Website*

## 1. PENDAHULUAN

### 1.1 Definisi Kualitas

Kualitas kata sering digunakan sebagai ekspresi mandiri tanpa kata sifat yang dilampirkan. Sebagian besar istilah “berkualitas tinggi” untuk mengungkapkan kepuasan dengan produk atau layanan yang digunakan. Ketika kita mengatakan bahwa suatu produk adalah produk yang berkualitas, itu berarti produk tersebut berkualitas baik. Di sisi lain, mereka kebanyakan menggunakan istilah “kualitas buruk” untuk mengungkapkan ketidakpuasan mereka terhadap produk atau layanan yang digunakan. Oleh karena itu, di benak kebanyakan orang, kata sifat “baik” secara implisit dikaitkan dengan kualitas kata

Definisi kualitas populer lainnya seperti yang didefinisikan oleh Joseph Moses Juran adalah kesesuaian untuk tujuan, dan kesesuaian serta penggunaan sangat penting untuk memahami kualitas. Kecuali kita mendefinisikan dua kata kunci ini, definisi kualitas sudah lengkap. Interpretasi kedua istilah ini oleh konsumen.

### 1.2 Pentingnya Pengujian Untuk Menjamin Kualitas

Pengujian perangkat lunak dianggap sebagai kegiatan penting dalam pengembangan perangkat lunak. Baru-baru ini, pentingnya pengujian independen (pengujian dilakukan oleh orang di luar perangkat lunak) menjadi lebih umum, sehingga perusahaan perangkat lunak yang berspesialisasi dalam pengujian perangkat lunak telah berkembang dan cukup berhasil. Meningkatnya kompleksitas dan ukuran perangkat lunak telah menyebabkan pengujian yang lebih kompleks dan jenis pengujian yang lebih beragam. Kualitas kepatuhan dibangun dengan mendefinisikan proses dan perbaikan terus menerus melalui seluruh program pengembangan dan penjaminan mutu. Kepatuhan dipastikan melalui pengukuran dan pengendalian.

**Tabel 1.** Merangkum Teknik yang Tersedia untuk Penjaminan Kualitas

Dimensi Kualitas	Bagaimana Membangun Kualitas	Teknik Untuk Memastikan Kualitas
Kualitas spesifikasi	Dokumentasi proses pengembangan spesifikasi; standard dan pedoman, format, dan pelat suhu untuk menentukan spesifikasi; dan daftar periksa	Ulasan ahli, ulasan sejawat, dan curah pendapat
Kualitas desain	Dokumentasi proses desain perangkat lunak; standar dan garis panduan, format, dan template untuk desain perangkat lunak; dan	Tinjauan ahli, tinjauan sejawat, tinjauan manajerial, dan curah pendapat
Kualitas pembangunan	Pedoman pengkodean, manajemen konfigurasi, dan manajemen perubahan	Ulasan sejawat dan perangkat lunak pengujian
Kesesuaian kualitas	Penerapan yang sungguh-sungguh dari semua aktivitas penjaminan mutu dalam organisasi, definisi proses, dan peningkatan	Audit, pengukuran, dan metrik untuk aktivitas jaminan kualitas, dan perbandingan metrik organisasi dengan metrik industri

### 1.3 Web Browser dan Web Server

*Web browser* atau sering juga disebut *Internet Browser* berfungsi sebagai jembatan bagi pengguna *computer* dalam menjelajah dunia maya. *Internet browser* merupakan sebuah aplikasi atau *software* yang digunakan untuk mengolah data yang ditransfer dari *world Wide Web* (lebih dikenal dengan istilah *www*) ke komputer dan menampilkannya secara visual agar mudah dimengerti oleh pengguna internet. (Setyaji dkk). *Browser web* adalah *software* yang digunakan untuk menampilkan informasi *web server*. Sedangkan aplikasi *web* adalah aplikasi yang diakses melalui jaringan internet. Dengan menggunakan aplikasi *web*, kita hanya perlu menempatkan aplikasi dalam sebuah server sehingga aplikasi tersebut dapat diakses dari manapun asal pengguna dapat mengakses *web-server*nya.

*Web server* adalah *server* yang melayani semua permintaan aplikasi *web*. Dalam hal ini untuk penulisan pemrogramannya adalah *specific* yaitu menggunakan Bahasa pemrograman *HTML (Hypertext Marku Language)* (Sidik, dkk).

Obyek yang akan diuji adalah Aplikasi *Forecasting FTTH Berbasis Website* (Studi Kasus Pada Pt YOFC International Indonesia). Aplikasi ini merupakan aplikasi buatan yang menitikberatkan pada penyimpanan database material hasil dari proses *Survey* dan *Design Review Meeting* pada project *FTTH (Fiber to the home)* MyRepublic Indonesia berdasarkan acuan *standard design pack* yang berlaku. Data ini berguna sebagai acuan *reconcile* pada saat akhir project telah terselesaikan yang berguna sebagai tolak ukur untuk menghitung *margin* sebuah perusahaan.

### 1.4 Masalah dan Kerugian

Adapun masalah yang diinginkan untuk tidak terjadi pada sebelum, saat, dan sesudah proses penginputan yaitu sebagai berikut :

1. Akses ke *web server* tidak bisa karena alasan teknik tertentu seperti *coding* ataupun *hosting* yang bermasalah.
2. Tidak muncul hasil editan ataupun save data yang telah diinput diberbagai form yang diuji ke dalam *database* ataupun *output dashboard*.
3. Tidak dapatnya akses sebuah user karena *database* tidak sync ataupun masalah teknikal lainnya

Apabila masalah diatas salah satunya atau semuanya terjadi maka perusahaan tidak dapat menggunakan aplikasi tersebut secara baik ataupun total. Karena diharapkan aplikasi ini hadir agar data asset perusahaan dengan berbagai tujuan nantinya dapat tersimpan dengan baik disebuah platform khusus yaitu aplikasi web server ini.

## 2. METODOLOGI PENELITIAN

Berdasarkan pemaparan oleh Jaya (2018), *black box testing* berkonsentrasi dari sisi kesesuaian perangkat lunak yang dikembangkan dengan kebutuhan pengguna yang telah didefinisikan pada saat awal perancangan. Pengujian dimaksudkan untuk mengetahui apakah fungsi-fungsi, masukan, dan keluaran dari perangkat lunak sesuai dengan spesifikasi yang dibutuhkan. Blackbox testing dilakukan sesuai dengan item uji yang telah dirancang. Adapun hasil blackbox testing adalah seluruh proses pada sistem telah berjalan dengan baik.

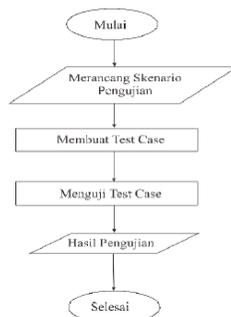
Keuntungan penggunaan metode *Blackbox Testing* adalah: (1) Penguji tidak perlu memiliki pengetahuan tentang bahasa pemrograman tertentu; (2) Pengujian dilakukan dari sudut pandang pengguna, ini membantu untuk mengungkapkan ambiguitas atau inkonsistensi dalam spesifikasi persyaratan; (3) *Programmer* dan *tester* keduanya saling bergantung satu sama lain. Kekurangan dari metode *Blackbox Testing* adalah : (1) Uji kasus sulit disain tanpa spesifikasi yang jelas; (2) Kemungkinan memiliki pengulangan tes yang sudah dilakukan oleh programmer; (3) Beberapa bagian *back end* tidak diuji sama sekali.

Saat ini terdapat banyak metode atau teknik untuk melaksanakan *Black Box Testing*, antara lain : *Equivalence Partitioning* merupakan suatu teknik pengujian yang berorientasi pada nilai masukan dan nilai keluaran yang dihasilkan pada suatu fungsi. *Equivalence Partitions* yang merupakan pengujian berdasarkan input pada setiap form pada aplikasi, dimana terdapat hasil yang diharapkan dan hasil aktualnya beserta penarikan kesimpulan berhasil atau gagal.

## 3. ANALISA DAN PEMBAHASAN

### 3.1 Proses Melakukan Pengujian

Dalam melakukan *software testing* ada beberapa tahapan-tahapan kegiatan yang harus di laksanakan. Hal ini dilukiskan melalui diagram alir pada gambar 1:



**Gambar 1.** Diaagram Alir Kegiatan *Software Testing*

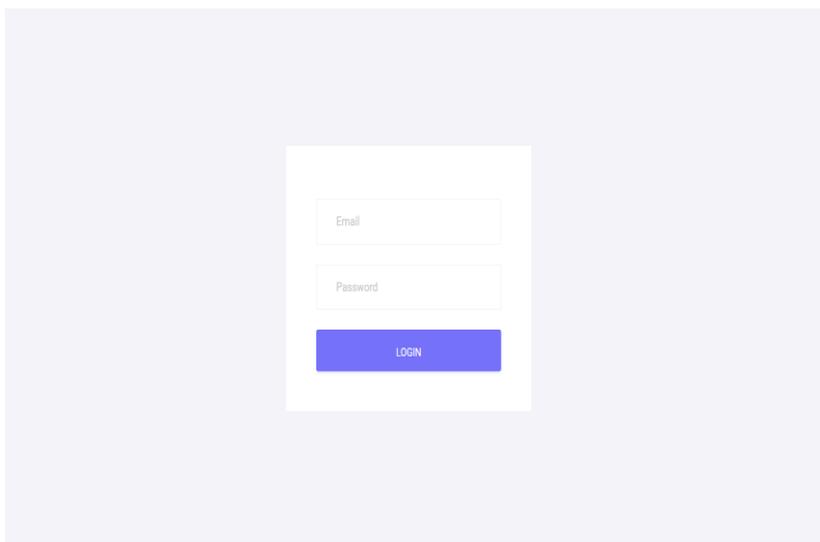
Keterangan:

1. Tahap perancangan scenario, merancang perintah pada form yang akan dilakukan pengujian.
2. Tahap membuat test case, yaitu mencatat kolom-kolom yang akan diuji pada form penginputan.
3. Tahap menguji test case, melakukan pengujian dengan melakukan perintah pada skenario pengujian.
4. Tahap hasil pengujian, menuliskan seluruh alur pengujian dalam bentuk laporan beserta metode yang digunakan.

### 3.2 Rancangan Test Case Form

Dalam penelitian dilakukkann pengujian terhadap 4 buah form yaitu: *form login*, *form produk*, *form kategori*, *form projek*. Adapun pertimbangan pengambilan form masing-masing dapat diterangkan sebagai berikut:

#### 1. Pengujian Form Login



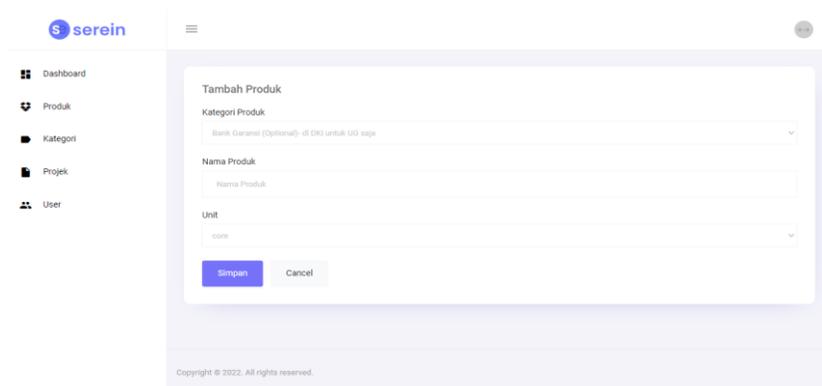
**Gambar 2.** Pengujian *Form Login*

Halaman ini diuji dikarenakan halaman awal sebagai portal *user* untuk menggunakan aplikasi tersebut di *web server*. Pada form ini akan dilakukan 4 pengujian pada tabel 2:

**Tabel 2.** Rancangan *Input Pada Data Form Login*

<b>Id</b>	<b>Deskripsi Pengujian</b>	<b>Hasil yang Diharapkan</b>
TEST-1-01	Mengosongkan field email dan password	Sistem akan menolak akses login dikarenakan field email dan password harus diisi
TEST-1-02	Mengisi field email dan mengosongkan field password	Sistem akan menolak akses login dikarenakan field password harus diisi
TEST-1-03	Mengisikan isian dengan salah satu data yang salah	Sistem akan menampilkan pemberitahuan bahwa username atau kata sandi salah
TEST-1-04	Mengisikan isian dengan data yang benar	Sistem akan menerima akses login dan akan mengarahkan langsung ke dashboard pengguna

## 2. Pengujian *Form* Produk



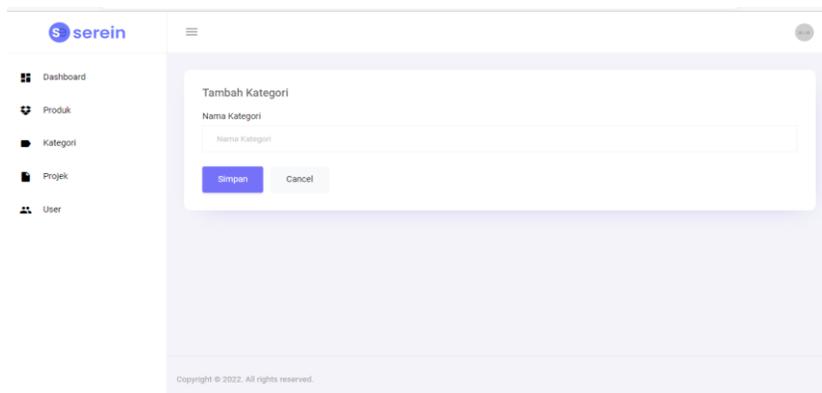
**Gambar 3.** Pengujian *Form* Produk

Halaman ini adalah halaman yang mungkin jadi paling banyak yang disesuaikan dengan kebutuhan *user* dalam menjalankan sebuah projek. Form ini memiliki 3 isian yang harus diberikan pengguna sesuai dengan Kategori, Nama Produk dan *Unit of Measure (UoM)* berdasarkan Design Pack yang menjadi acuan projek. Rancangan pengujiannya ialah dengan memasukkan 4 variasi input yang sering terjadi. Hal tersebut terlampir dalam tabel 3:

**Tabel 3.** Rancangan *Input* Pada Data *Form* Produk

<b>Id</b>	<b>Deskripsi Pengujian</b>	<b>Hasil yang Diharapkan</b>
TEST-2-01	Mengosongkan semua <i>field</i> dalam halaman produk	Sistem akan memberikan pesan error bahwa pengguna harus mengisi semua <i>field</i> sebagai <i>mandatory</i>
TEST-2-02	Mengisi <i>field</i> nama kategori dan mengosongkan <i>field</i> nama produk dan unit	Sistem akan memberikan pesan error bahwa pengguna harus mengisi <i>field</i> nama produk dan unit
TEST-2-03	Mengisi <i>field</i> nama produk dan unit tanpa mengisi <i>field</i> nama kategori	Sistem akan menampilkan pemberitahuan bahwa harus mengisi nama produk
TEST-2-04	Mengisi seluruh <i>field</i> dengan data yang benar	Sistem akan menyimpan nama produk ke dalam <i>database</i> sesuai data isian

## 3. Pengujian *Form* Kategori



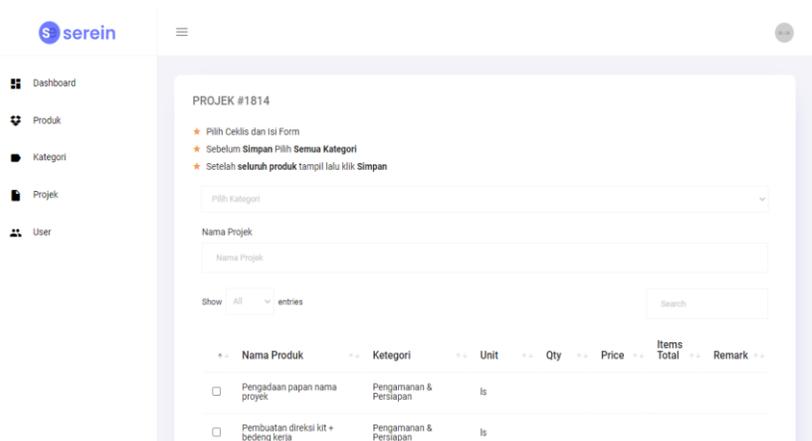
**Gambar 4.** Pengujian *Form* Kategori

Halaman ini diuji sebagai salah satu field mandatori yang kaitannya ada di dalam *form* proyek. Rancangan pengujiannya ialah dengan 3 variasi input. Hal tersebut terlampir dalam tabel 4:

**Tabel 4.** Rancangan *Input* Pada Data *Form* Kategori

<b>Id</b>	<b>Deskripsi Pengujian</b>	<b>Hasil yang Diharapkan</b>
TEST-3-01	Mengosongkan semua <i>field</i> dalam halaman kategori	Sistem akan memberikan pesan error bahwa pengguna harus mengisi field yang kosong
TEST-3-02	Mengisi <i>field</i> kategori dan melakukan cancel	Sistem tidak akan menyimpan database dan <i>refresh</i> ulang halaman
TEST-3-03	Mengisi <i>field</i> kategori sesuai kebutuhan	Sistem akan menampilkan pemberitahuan bahwa data berhasil disimpan di database

#### 4. Pengujian *Form* Proyek



**Gambar 5.** Pengujian *Form* Proyek

Halaman ini diuji sebagai halaman utama dari keseluruhan kategori, produk dan satuan unit yang telah didefinisikan pada *form* sebelumnya. Rancangan pengujiannya dengan melakukan 3 variasi input yang terlampir pada tabel 5 berikut:

**Tabel 5.** Rancangan *Input* Pada Data *Form* Proyek

<b>Id</b>	<b>Deskripsi Pengujian</b>	<b>Hasil yang Diharapkan</b>
TEST-4-01	Menceklis <i>form</i> tanpa memberikan input pada kolom <i>quantity</i> , <i>price</i> , dan <i>remark</i>	Sistem akan memberikan pesan error bahwa pengguna harus mengisi field <i>quantity</i> dan <i>price</i>
TEST-4-02	Menceklis <i>form</i> dan memberikan input pada kolom <i>quantity</i> saja tanpa <i>price</i>	Sistem tidak akan menyimpan database dan meminta untuk mengisi kolom <i>price</i> terlebih dahulu
TEST-4-03	Menceklis <i>form</i> dan menginput semua kolom sesuai kebutuhan	Sistem akan menampilkan pemberitahuan bahwa data berhasil disimpan di database

#### 4. IMPLEMENTASI

Setelah data input kita masukkan pada sistem, maka kita lihat bagaimana hasilnya apakah sesuai dengan yang diharapkan atau tidak. Dengan memberikan 3-4 variasi masukan pada masing-masing *form* yang diuji, hasilnya dapat dilihat pada tabel 2, 3, 4 dan 5.

Dari hasil pengujian untuk *Form Login* menunjukkan bahwa apa yang diharapkan dalam perancangan sesuai dengan hasil outputnya . Demikian pula untuk *Form Kategori, Produk, dan Proyek* buku hasilnya sesuai dengan yang diharapkan baik untuk masukan yang salah maupun masukan yang benar.

Secara keseluruhan dilakukan 14 skenario pengujian pada halaman yang berbeda. Pengujian tersebut menunjukkan hasil yang sangat memuaskan. Berdasarkan data tersebut didapatkan perhitungan sebagai berikut:

$$\text{Validitas sistem} = (JS - JTS) / TSP \times 100\%$$

$$\text{Validitas sistem} = (14 - 0) / 12 \times 100\% = 100\%$$

dengsn JS=jumlah skenario, J=jumlah tidak sesuai, TSP=total skenario pengujian

#### a. Hasil Pengujian *Form Login*

**Tabel 6.** Hasil Pengujian *Form Login*

<b>Id</b>	<b>Deskripsi Pengujian</b>	<b>Hasil yang Diharapkan</b>	<b>Kesimpulan</b>
TEST-1-01	Mengosongkan field email dan password	Sistem akan menolak akses login dikarenakan field email dan password harus diisi	Sesuai
TEST-1-02	Mengisi field email dan mengosongkan field password	Sistem akan menolak akses login dikarenakan field password harus diisi	Sesuai
TEST-1-03	Mengisikan isian dengan salah satu data yang salah	Sistem akan menampilkan pemberitahuan bahwa username atau kata sandi salah	Sesuai
TEST-1-04	Mengisikan isian dengan data yang benar	Sistem akan menerima akses login dan akan mengarahkan langsung ke dashboard pengguna	Sesuai

#### b. Hasil Pengujian *Form Produk*

**Tabel 7.** Hasil Pengujian *Form Produk*

<b>Id</b>	<b>Deskripsi Pengujian</b>	<b>Hasil yang Diharapkan</b>	<b>Kesimpulan</b>
TEST-2-01	Mengosongkan semua <i>field</i> dalam halaman produk	Sistem akan memberikan pesan error bahwa pengguna harus mengisi semua <i>field</i> sebagai <i>mandatory</i>	Sesuai
TEST-2-02	Mengisi <i>field</i> nama kategori dan mengosongkan <i>field</i> nama produk dan unit	Sistem akan memberikan pesan error bahwa pengguna harus mengisi <i>field</i> nama produk dan unit	Sesuai
TEST-2-03	Mengisi <i>field</i> nama produk dan unit tanpa mengisi <i>field</i> nama kategori	Sistem akan menampilkan pemberitahuan bahwa harus mengisi nama produk	Sesuai
TEST-2-04	Mengisi seluruh <i>field</i> dengan data yang benar	Sistem akan menyimpan nama produk ke dalam <i>database</i> sesuai data isian	Sesuai

**c. Hasil Pengujian Form Kategori**

**Tabel 8.** Hasil Pengujian *Form* Kategori

<b>Id</b>	<b>Deskripsi Pengujian</b>	<b>Hasil yang Diharapkan</b>	<b>Kesimpulan</b>
TEST-3-01	Mengosongkan semua <i>field</i> dalam halaman kategori	Sistem akan memberikan pesan error bahwa pengguna harus mengisi <i>field</i> yang kosong	Sesuai
TEST-3-02	Mengisi <i>field</i> kategori dan melakukan cancel	Sistem tidak akan menyimpan database dan <i>refresh</i> ulang halaman	Sesuai
TEST-3-03	Mengisi <i>field</i> kategori sesuai kebutuhan	Sistem akan menampilkan pemberitahuan bahwa data berhasil disimpan di database	Sesuai

**d. Hasil Pengujian Form Projek**

**Tabel 9.** Hasil Pengujian *Form* Projek

<b>Id</b>	<b>Deskripsi Pengujian</b>	<b>Hasil yang Diharapkan</b>	<b>Kesimpulan</b>
TEST-4-01	Menceklis <i>form</i> tanpa memberikan input pada kolom <i>quantity</i> , <i>price</i> , dan <i>remark</i>	Sistem akan memberikan pesan error bahwa pengguna harus mengisi <i>field</i> <i>quantity</i> dan <i>price</i>	Sesuai
TEST-4-02	Menceklis <i>form</i> dan memberikan input pada kolom <i>quantity</i> saja tanpa <i>price</i>	Sistem tidak akan menyimpan database dan meminta untuk mengisi kolom <i>price</i> terlebih dahulu	Sesuai
TEST-4-03	Menceklis <i>form</i> dan menginput semua kolom sesuai kebutuhan	Sistem akan menampilkan pemberitahuan bahwa data berhasil disimpan di database	Sesuai

**5. KESIMPULAN**

Pengujian perangkat lunak menggunakan Metode Black Box Testing dengan Teknik *Equivalence Partition* cocok bagi pemula karena lebih sederhana caranya. Hasil pengujian menunjukkan output sudah sesuai dengan yang diharapkan pada perancangan dengan validitas 100%. Untuk itu aplikasi ini sudah lulus uji secara testing dasar.

**REFERENCES**

Dioviando Putra Rakhmadani, B. A. (2021). Pengujian Blackbox Menggunakan Teknik *Equivalence Partitions* pada Aplikasi Petgram Mobile. *Jurnal ICTEE*, 10-16.

Jaya, M. S., Gumilang, P., Wati, T., Andersen, Y. (2019). Pengujian Black Box pada Aplikasi Sistem Penunjang Keputusan Seleksi Calon Pegawai Negeri Sipil Menggunakan Teknik *Equivalence Partitions*. *Jurnal Informatika Universitas Pamulang*, 131-136.

Jaya, T. S. (2018). Pengujian Aplikasi dengan Metode Blackbox Testing *Boundary Value Analysis* (Studi Kasus: Kantor Digital Politeknik Negeri Lampung). *Jurnal Pengembangan IT (JPIT)*, 1-4.

Rakasiwi, S. (2023). *TEKNIK MENJAMIN KUALITAS BAGI PENGEMBANG PERANGKAT LUNAK. SEMARANG: YAYASAN PRIMA AGUS TEKNIK.*

Suraya, U. M. (2022). PENGUJIAN FUNGSIONAL PERANGKAT LUNAK SISTEM INFORMASI PERPUSTAKAAN DENGAN METODE BLACK BOX TESTING. *Jurnal Ilmiah Teknik dan Ilmu Komputer*, 1-8.