

Pengujian Black Box pada Aplikasi Manajemen Persediaan Sparepart di Perusahaan CV. Anugerah Sejahtera Menggunakan Metode Equivalence Partitioning

Marsella Virgie¹, Cintiya Aulya Fransiska², Vannes Jhonatan³,
Antonius Wahyu S., Kom., M. T. I.⁴

^{1,2,3}Program Studi Sistem Informasi, Universitas Multi Data Palembang

E-mail : Marsellavirgie27@mhs.mdp.ac.id, cintiyaulya14@mhs.mdp.ac.id,
lukvj@mhs.mdp.ac.id, wahyu.sudrajat@mdp.ac.id.

Abstract

Software testing is an important component in the software development life cycle to ensure application quality and functionality. This article aims to explore the application of Black Box testing in website-based spare parts inventory management applications at CV companies. Anugerah Sejahtera using the Equivalence Partitioning method. This method was chosen to increase the efficiency and effectiveness of testing by dividing the input domain into partitions that are considered equivalent in testing. Through this case study, the test results will show how the Equivalence Partitioning method can be implemented, followed by analysis of the test results to assess the success of the application in meeting the company's business needs.

Article History

Submitted: 21 May 2024
Accepted: 30 May 2024
Published: 31 May 2024

Key Words

Black Box, Equivalence Partitioning, Inventory, Website.

Abstrak

Pengujian perangkat lunak merupakan komponen penting dalam siklus hidup pengembangan perangkat lunak untuk memastikan kualitas dan fungsionalitas aplikasi. Artikel ini bertujuan untuk mengeksplorasi penerapan pengujian *Black Box* pada aplikasi manajemen persediaan sparepart berbasis *website* di perusahaan CV. Anugerah Sejahtera dengan menggunakan metode *Equivalence Partitioning*. Metode ini dipilih untuk meningkatkan efisiensi dan efektivitas pengujian dengan membagi input domain menjadi partisi yang dianggap setara dalam pengujian. Melalui studi kasus ini, hasil pengujian akan menunjukkan bagaimana metode *Equivalence Partitioning* dapat diimplementasikan, diikuti dengan analisis hasil pengujian untuk menilai keberhasilan aplikasi dalam memenuhi kebutuhan bisnis perusahaan.

Sejarah Artikel

Submitted: 21 May 2024
Accepted: 30 May 2024
Published: 31 May 2024

Kata Kunci

Black Box, Equivalence Partitioning, Persediaan, Website.

Pendahuluan

Perusahaan CV. Anugerah Sejahtera adalah perusahaan yang bergerak di bidang distribusi dan penjualan *sparepart* otomotif. Seiring dengan pertumbuhan bisnis, manajemen persediaan menjadi semakin kompleks dan memerlukan sistem yang andal untuk mengelola stok, pemesanan, dan penjualan *sparepart*. Aplikasi manajemen persediaan berbasis *website* yang dikembangkan untuk perusahaan ini harus melalui serangkaian pengujian. Pengujian tersebut bertujuan untuk melihat kesesuaian fungsi-fungsi yang terdapat pada aplikasi [1], sehingga pengujian tersebut dapat digunakan untuk memperoleh pengukuran kualitas aplikasi [2] dan memastikan fungsionalitasnya berjalan dengan baik. Pengujian *black box* menjadi pilihan dengan melakukan pengujian pada batas nilai yang diisi pada aplikasi melalui serangkaian skenario kasus uji yang telah ditentukan untuk setiap fungsi [3].

Metode Penelitian

Tujuan dari penelitian ini adalah untuk menguji bagaimana fungsionalitas dari aplikasi manajemen persediaan *sparepart* di perusahaan CV. Anugerah Sejahtera dengan menggunakan metode *Equivalence Partitioning* dan pengujian *Black Box*. Dimana pada Pengujian *Black Box* berfokus pada memastikan bahwa setiap proses beroperasi sesuai dengan kebutuhan, sehingga pengujian adalah dengan menerapkan metode pelaksanaan program yang bertujuan untuk menemukan kesalahan dengan berfokus pada spesifikasi fungsional penggunaan aplikasi sehingga dapat segera dilakukan perbaikan agar sistem layak untuk digunakan [4].

Data masukan dari unit perangkat lunak dibagi menjadi beberapa partisi data menggunakan teknik pembagian kesesuaian. Teknik ini dapat digunakan untuk menurunkan kasus uji. Hasil ini bertujuan untuk mengetahui seberapa efektif dan efisien dari suatu aplikasi manajemen persediaan *sparepart* berbasis *website*. Jumlah tes kasus yang dibuat berdasarkan fungsi sistem yang ada, dan jumlah tes kasus yang dibuat menentukan seberapa cepat deteksi kesalahan pada *software* yang akan dikirim, dan hasil dan diskusi didasarkan pada daftar tes kasus yang dibuat. Semakin cepat deteksi kesalahan *software* dilakukan, semakin baik hasilnya [5].

Karena itu, melakukan pengujian sangat penting untuk mengurangi kesalahan tersebut. Dalam tahap pengujian *Equivalence Partitions* yang dilakukan adalah :

1. Membuat rancangan test case berdasarkan fungsi yang ada dalam pengujian perangkat lunak.
2. Membuat batasan pengujian *Equivalence Partitions*.
3. Membuat model pengujian dari skenario pengujian dan hasil yang diharapkan.
4. Melakukan pengujian berdasarkan model yang dirancang [6].

Pengujian menggunakan skenario pengujian acak untuk memastikan aksi yang diterima sistem baik dalam bentuk penolakan dan penerimaan yang dilakukan sistem pada *database* sehingga sistem dapat dinilai layak untuk digunakan [7].

A. Pengujian *Black Box*

Metode ini menguji berdasarkan dari fungsi yang disediakan oleh aplikasi yang akan disesuaikan dengan cara kerja sistem yang berjalan sesuai yang diinginkan perancangannya, apabila masukkan yang diisi tidak sesuai dengan aturan pengisian kolom maka data input tidak dapat tersimpan dalam *database* [8] dengan melalui banyaknya data uji dan banyak kolom data input yang akan diuji sesuai dengan aturan masukan sistem, untuk mengamati hasil eksekusi data dengan menemukan beberapa kesalahan pada pengujian [9].

B. Pengujian *Equivalence Partitions*

Pengujian ini dapat mengurangi potensi terjadinya cacat pada aplikasi dan memudahkan penilaian kelayakan aplikasi yang berfokus pada input data dimana setiap opsi input akan dianalisis kemudian dikelompokkan berdasarkan fungsinya sehingga didapat hasil dimana masukkan tersebut dianggap valid atau tidak valid [5]. Berikut pengujian sistem pada aplikasi manajemen persediaan CV Anugerah Sejahtera berbasis *website* yang kami implementasikan pada tabel rancangan *Test Case*.

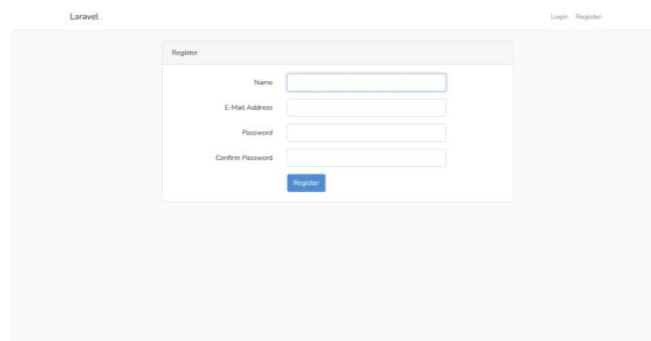


Gambar 1. Form Login

Berdasarkan pada Gambar 1 terdapat beberapa rencana pengujian yang dilakukan pada saat memasukkan data username mulai dari email dan password, validasi akan berhasil jika email dan password yang dimasukan sama dengan data yang sudah dibuat saat sign up dan pengguna akan masuk ke halaman utama. Sebaliknya jika user memasukan email atau password yang salah maka user akan mendapatkan pesan error atau gagal untuk login ke aplikasi tersebut.

Table 1. Rancangan Test Case dari Form Login

| ID | Deskripsi | Hasil yang diharapkan |
|-----|---|--|
| A01 | Memasukan email dan password yang sesuai dengan data yang dibuat sebelumnya dan menekan tombol 'Sign In'. | Berhasil login dan akan masuk dialihkan ke halaman <i>Dashboard</i> . |
| A02 | Memasukan email atau password yang tidak sesuai dengan data. | Gagal melakukan <i>login</i> dan tetap akan di layar <i>sign in</i> . |
| A03 | Melakukan daftar email dan password apabila data <i>user</i> belum terdaftar pada aplikasi dengan menekan tombol 'Sign Up'. | <i>User</i> akan diarahkan ke halaman <i>Sign Up</i> untuk mengisi data yang diperlukan untuk membuat akun pengguna. |



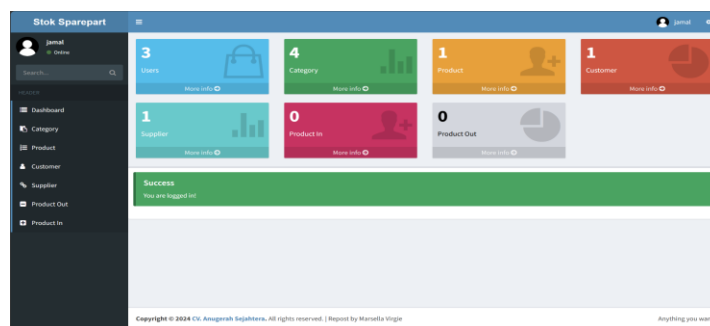
Gambar 2. Halaman Sign Up

Pada tampilan Gambar 2 terdapat *form* yang harus diisi user ketika menekan tombol *sign up*, yang pertama user harus mengisi nama, lalu *email address*, *password* dan *confirm password*.

Setelah mengisi semua data *form* dengan benar maka user dapat menekan tombol *register* dan data akan tersimpan di dalam *database*. Setelah itu *user* akan dialihkan ke halaman login kembali dan dapat memasukkan data yang sudah diisi saat di halaman registrasi.

Table 2. Rancangan Test Case dari Halaman Sign Up

| ID | Deskripsi | Hasil yang diharapkan |
|-----|---|---|
| B01 | Mengisi semua kolom yang tersedia sesuai dengan ketentuan dan mengklik tombol 'Register' | Data Berhasil terdaftar pada aplikasi. |
| B02 | Mengisi semua kolom tetapi pada kolom nama dibuat sama dengan data yang pernah disimpan. | Data Berhasil tersimpan di <i>database</i> . |
| B03 | Mengisi semua kolom tetapi pada kolom <i>email address</i> dibuat sama dengan data yang pernah disimpan. | Gagal menyimpan data dan muncul pesan kesalahan pada kolom email. |
| B04 | Mengisi semua kolom tetapi pada kolom <i>password</i> dibuat sama dengan data yang pernah disimpan. | Data Berhasil tersimpan di <i>database</i> . |
| B05 | Mengisi semua kolom tetapi pada bagian <i>confirm password</i> tidak sesuai dengan password yang dibuat. | Muncul pesan kesalahan di kolom <i>Confirm Password</i> , pengguna diminta untuk memasukkan kembali password. |
| B06 | Mengisi kolom nama dan <i>email</i> yang berbeda tetapi <i>password</i> yang dibuat sama dengan yang pernah disimpan. | Data Berhasil tersimpan di <i>database</i> . |



Gambar 3. Halaman Dashboard

Pada tampilan Gambar 3 halaman ini digunakan untuk menampilkan total input data yang dilakukan oleh pengguna yang berhasil mengakses aplikasi. Halaman ini akan menunjukkan seberapa banyaknya *user* yang mengakses, mengelola data kategori barang, mengelola data produk, mengelola data *customer*, mengelola data *suppliers*, mengelola persediaan masuk dan persediaan keluar. Pengujian dapat dilakukan dengan melakukan interaksi pada tombol halaman

yang disediakan atau dengan tombol *'more info'* yang akan mengarahkan pada halaman yang dipilih serta pengguna juga dapat melihat akun dengan klik ikon *user* dan melakukan *log out*.

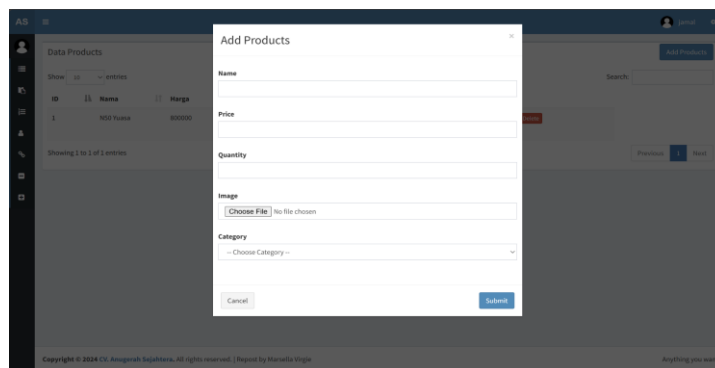
Table 3. Rancangan Test Case dari Halaman Dashboard

| ID | Deskripsi | Hasil yang diharapkan |
|-----|---|--|
| C01 | Mengklik tombol <i>'more info'</i> berdasarkan halaman yang dipilih pengguna. | Melakukan interaksi dengan tombol tersebut akan mengalihkan pengguna pada halaman yang telah dipilih pengguna. |
| C02 | Mengklik ikon <i>user</i> . | Menampilkan email pengguna dan tombol <i>log out</i> . |
| C03 | Mengklik tombol <i>'Logout'</i> . | Berhasil keluar dari akun. |

Tampilan halaman pada Gambar 4 ini digunakan agar bagian gudang dapat memasukkan nama kategori produk dengan mengklik tombol *'add categories'*, setelahnya pengguna dapat melihat deskripsi kategori, mengedit kategori, dan menghapus kategori. Data dapat unduh dalam bentuk file pdf maupun excel.

Table 4. Rancangan Test Case dari Halaman Data Kategori

| ID | Deskripsi | Hasil yang diharapkan |
|-----|--|---|
| D01 | Menambah data kategori dengan mengklik tombol <i>'Add Categories'</i> . | Menampilkan kolom input nama kategori untuk menambahkan kategori ke database, data yang tersimpan dapat di lihat, diubah, dan dihapus.. |
| D02 | Mengunduh laporan dengan mengklik tombol <i>'Export PDF'</i> , <i>'Export Excel'</i> . | Laporan terunduh sesuai dengan bentuk file yang dipilih. |



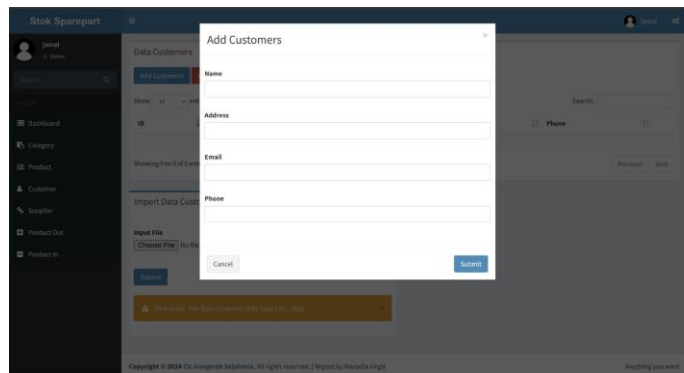
Gambar 5. Halaman Data Produk

Pada tampilan halaman Gambar 5 digunakan agar bagian gudang dapat memasukkan nama barang, harga, jumlah barang, tampilan barang, dan kategori barang dengan mengklik tombol *'Add*

Products apabila ingin menyimpan data produk ke *database* pengguna perlu mengklik tombol *Submit*.

Table 5. Rancangan Test Case dari Halaman Data Produk

| ID | Deskripsi | Hasil yang diharapkan |
|-----|---|--|
| E01 | Mengisi seluruh data produk yang diperlukan dan menekan tombol “ <i>Submit</i> ”. | Hasil data yang diinput akan masuk ke dalam <i>database</i> . |
| E02 | Mengisi kolom harga dengan menggunakan huruf dan tanda baca (.). | Muncul pesan kesalahan pada kolom harga, pengguna diharuskan untuk menginput harga. |
| E03 | Mengisi semua kolom yang ada tetapi kolom <i>quantity</i> tidak di isi. | Muncul pesan kesalahan pada kolom <i>quantity</i> , pengguna diharuskan untuk menginput jumlah barang. |
| E04 | Mengisi semua kolom kecuali <i>category</i> dan tidak memasukan gambar | Muncul pesan kesalahan pada kolom kategori dan gambar barang, pengguna diharuskan untuk menginput kategori barang dan gambar barang. |
| E05 | Mengisi semua kolom dengan data yang baru tetapi pada kolom nama dibuat sama dengan data yang sudah ada | Muncul pesan kesalahan pada kolom nama yang memberitahu pengguna bahwa nama tersebut sudah ada di <i>database</i> . |

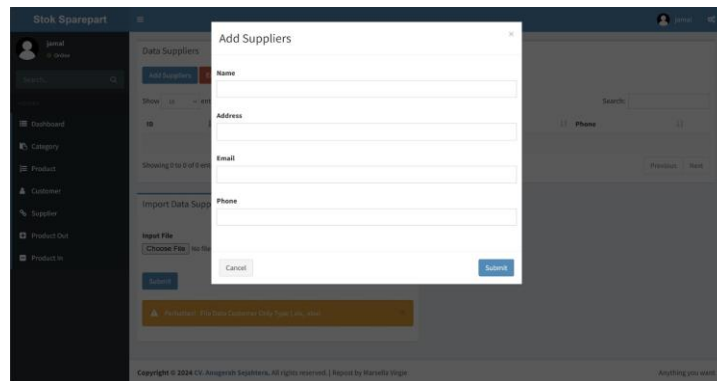


Gambar 6. Halaman Data Pelanggan

Pada tampilan Gambar 6, digunakan bagian penjualan untuk menambahkan data pelanggan yang baru, dengan memasukan nama, alamat, email serta nomor telepon. Apabila semua kolom tersebut sudah diisi maka selanjutnya dapat menekan tombol ‘*Submit*’ untuk menyimpan data pelanggan yang baru.

Table 6. Rancangan Test Case dari Halaman Data Pelanggan

| ID | Deskripsi | Hasil yang diharapkan |
|-----|--|--|
| F01 | Mengisi semua kolom dengan data baru dan menekan tombol 'Submit' | Data pelanggan berhasil tersimpan ke dalam <i>database</i> . |
| F02 | Mengisi semua kolom dengan data yang berbeda, tetapi data nama menggunakan data yang sudah ada sebelumnya. | Pengguna dapat melakukan <i>submit</i> dan data pelanggan berhasil tersimpan ke dalam <i>database</i> . |
| F03 | Mengisi kolom <i>email</i> dengan menggunakan email yang sudah ada sebelumnya. | Muncul pesan kesalahan pada kolom <i>email</i> , pengguna diminta untuk memasukkan <i>email</i> yang baru. |
| F04 | Mengisi semua kolom kecuali kolom <i>phone</i> dan 'Submit'. | Muncul pesan kesalahan pada kolom <i>phone</i> , pengguna diminta memasukkan nomor telepon. |
| F05 | Mengisi semua kolom kecuali kolom <i>email</i> dan 'Submit'. | Berhasil menyimpan data pelanggan ke <i>database</i> . |



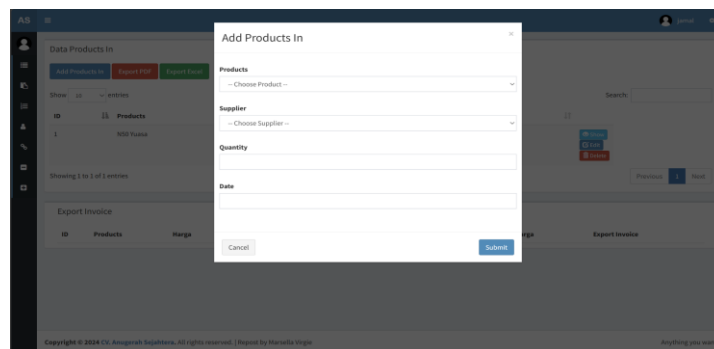
Gambar 7. Halaman Data Suppliers

Pada tampilan Gambar 7, digunakan saat bagian penjualan mencoba memasukan data *supplier* ke dalam *database*, dengan mengisi kolom nama, alamat, *email* serta nomor telepon. Apabila sudah mengisi semua kolom yang diminta maka dapat menekan tombol 'Submit' untuk menyimpan hasil input data *supplier* ke dalam *database*.

Table 7. Rancangan Test Case dari Halaman Data Suppliers

| ID | Deskripsi | Hasil yang diharapkan |
|-----|---|--|
| G01 | Mengisi semua kolom dengan data baru dan menekan tombol 'Submit' | Data <i>suppliers</i> berhasil tersimpan ke dalam <i>database</i> . |
| G02 | Mengisi semua kolom dengan data yang berbeda, tetapi kolom nama diisi menggunakan data yang sudah | Pengguna dapat melakukan <i>submit</i> dan data <i>suppliers</i> berhasil tersimpan ke dalam <i>database</i> . |

| | | |
|-----|--|--|
| | ada sebelumnya. | |
| G03 | Mengisi kolom <i>email</i> dengan menggunakan email yang sudah ada sebelumnya. | Muncul pesan kesalahan pada kolom <i>email</i> , pengguna diminta untuk memasukkan <i>email</i> yang baru. |
| G04 | Mengisi semua kolom kecuali kolom <i>phone</i> dan <i>'Submit'</i> . | Gagal menyimpan data dan muncul pesan kesalahan pada kolom <i>phone</i> , pengguna diminta memasukkan nomor telepon. |
| G05 | Mengisi semua kolom kecuali kolom <i>email</i> dan <i>'Submit'</i> . | Berhasil menyimpan data <i>suppliers</i> ke <i>database</i> . |



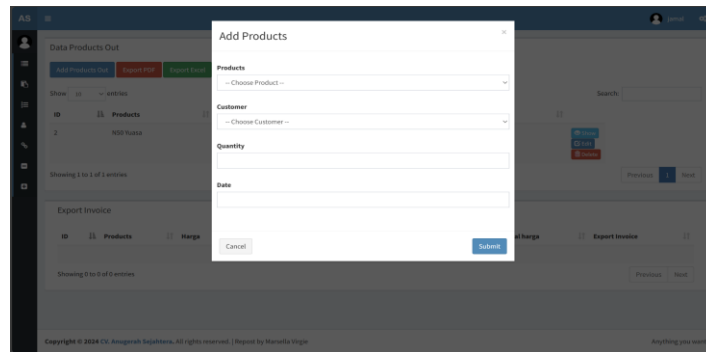
Gambar 8. Halaman Data Persediaan Masuk

Pada tampilan Gambar 8 akan digunakan oleh bagian gudang untuk menambahkan data persediaan yang masuk ke dalam database dengan mengisi semua kolom seperti kolom nama produk, nama supplier, jumlah persediaan yang masuk, dan tanggal persediaan masuk. Apabila sudah mengisi semua kolom yang diminta maka dapat menekan tombol *'Submit'* untuk menyimpan data ke dalam *database*.

Table 8. Rancangan Test Case dari Halaman Data Persediaan Masuk

| ID | Deskripsi | Hasil yang diharapkan |
|-----|--|--|
| H01 | Mengisi semua kolom dengan benar dan mengklik tombol <i>'Submit'</i> . | Data persediaan masuk berhasil tersimpan ke <i>database</i> . |
| H02 | Mengisi semua kolom kecuali kolom <i>'Products'</i> dan melakukan <i>'Submit'</i> . | Gagal menyimpan data dan muncul pesan kesalahan pada kolom <i>'products'</i> , pengguna diminta memilih produk yang ingin disimpan. |
| H03 | Mengisi semua kolom kecuali kolom <i>'Suppliers'</i> dan melakukan <i>'Submit'</i> . | Gagal menyimpan data dan muncul pesan kesalahan pada kolom <i>'suppliers'</i> , pengguna diminta memilih supplier yang ingin disimpan. |

| | | |
|-----|--|---|
| H04 | Mengisi semua kolom kecuali kolom 'Quantity' dan melakukan 'Submit'. | Gagal menyimpan data dan muncul pesan kesalahan pada kolom 'quantity', pengguna diminta mengisi jumlah barang yang ingin ditambahkan. |
| H05 | Mengisi semua kolom kecuali kolom 'Date' dan melakukan 'Submit'. | Gagal menyimpan data dan muncul pesan kesalahan pada kolom 'Date', pengguna diminta mengisi tanggal persediaan masuk. |



Gambar 9. Halaman Data Persediaan Keluar

Pada tampilan Gambar 9 akan digunakan oleh bagian penjualan untuk mengurangi persediaan yang sudah terjual dengan mengisi semua kolom seperti kolom nama produk, nama pembeli, jumlah barang yang dijual, dan tanggal barang tersebut dijual. Apabila sudah mengisi semua kolom yang diminta maka dapat menekan tombol 'Submit' untuk menyimpan data ke dalam database sehingga dapat dilakukan pengurangan persediaan otomatis dari sistem.

Table 9. Rancangan Test Case dari Halaman Data Persediaan Keluar

| ID | Deskripsi | Hasil yang diharapkan |
|-----|--|--|
| I01 | Mengisi semua kolom dengan benar dan mengklik tombol 'Submit'. | Data persediaan masuk berhasil tersimpan ke database. |
| I02 | Mengisi semua kolom kecuali kolom 'Products' dan melakukan 'Submit'. | Gagal menyimpan data dan muncul pesan kesalahan pada kolom 'products', pengguna diminta memilih produk yang ingin dilakukan pengurangan. |
| I03 | Mengisi semua kolom kecuali kolom 'Customer' dan melakukan 'Submit'. | Gagal menyimpan data dan muncul pesan kesalahan pada kolom 'customer', pengguna diminta memilih nama pelanggan yang membeli produk tersebut. |

| | | |
|-----|--|--|
| I04 | Mengisi semua kolom kecuali kolom 'Quantity' dan melakukan 'Submit'. | Gagal menyimpan data dan muncul pesan kesalahan pada kolom 'quantity', pengguna diminta mengisi jumlah barang yang dijual. |
| I05 | Mengisi semua kolom kecuali kolom 'Date' dan melakukan 'Submit'. | Gagal menyimpan data dan muncul pesan kesalahan pada kolom 'Date', pengguna diminta mengisi tanggal persediaan keluar. |

Hasil dan Pembahasan

Berdasarkan pengujian yang dilakukan pada aplikasi manajemen persediaan sparepart pada CV. Anugerah Sejahtera yang digunakan untuk mengetahui kekurangan dari sebuah aplikasi pada saat menggunakannya, hal ini juga dapat mengukur seberapa layak aplikasi sudah dapat digunakan untuk keperluan jangka panjang pada sebuah organisasi atau perusahaan.

Dengan melalui serangkaian test case dengan menggunakan pengujian black box dengan metode equivalence partitioning didapatkan hasil seperti yang ditunjukkan pada tabel 4 sebagai berikut :

Table 10. Hasil Test Case dari Aplikasi

| ID | Deskripsi | Hasil yang Diharapkan | Hasil Pengujian |
|-----|---|--|-----------------|
| A01 | Memasukan email dan password yang sesuai dengan data yang dibuat sebelumnya dan menekan tombol 'Sign In'. | Berhasil login dan akan masuk dialihkan ke halaman <i>Dashboard</i> . | Sesuai |
| A02 | Memasukan email atau password yang tidak sesuai dengan data. | Gagal melakukan <i>login</i> dan tetap akan di layar <i>sign in</i> . | Sesuai |
| A03 | Melakukan daftar email dan password apabila data <i>user</i> belum terdaftar pada aplikasi dengan menekan tombol 'Sign Up'. | <i>User</i> akan diarahkan ke halaman <i>Sign Up</i> untuk mengisi data yang diperlukan untuk membuat akun pengguna. | Sesuai |
| B01 | Mengisi semua kolom yang tersedia sesuai dengan ketentuan dan mengklik tombol 'Register' | Data Berhasil terdaftar pada aplikasi. | Sesuai |
| B02 | Mengisi semua kolom tetapi pada kolom nama dibuat sama dengan data yang pernah disimpan. | Data Berhasil tersimpan di <i>database</i> . | Sesuai |
| B03 | Mengisi semua kolom tetapi pada kolom <i>email address</i> dibuat sama dengan data yang pernah disimpan. | Gagal menyimpan data dan muncul pesan kesalahan pada kolom email. | Sesuai |
| B04 | Mengisi semua kolom tetapi pada | Data Berhasil tersimpan di | Sesuai |

| | | | |
|-----|---|---|--------|
| | kolom <i>password</i> dibuat sama dengan data yang pernah disimpan. | <i>database</i> . | |
| B05 | Mengisi semua kolom tetapi pada bagian <i>confirm password</i> tidak sesuai dengan <i>password</i> yang dibuat. | Muncul pesan kesalahan di kolom <i>Confirm Password</i> , pengguna diminta untuk memasukkan kembali <i>password</i> . | Sesuai |
| B06 | Mengisi kolom nama dan <i>email</i> yang berbeda tetapi <i>password</i> yang dibuat sama dengan yang pernah disimpan. | Data Berhasil tersimpan di <i>database</i> . | Sesuai |
| C01 | Mengklik tombol ' <i>more info</i> ' berdasarkan halaman yang dipilih pengguna. | Melakukan interaksi dengan tombol tersebut akan mengalihkan pengguna pada halaman yang telah dipilih pengguna. | Sesuai |
| C02 | Mengklik ikon <i>user</i> . | Menampilkan email pengguna dan tombol <i>log out</i> . | Sesuai |
| C03 | Mengklik tombol ' <i>Logout</i> '. | Berhasil keluar dari akun. | Sesuai |
| D01 | Menambah data kategori dengan mengklik tombol ' <i>Add Categories</i> '. | Menampilkan kolom input nama kategori untuk menambahkan kategori ke <i>database</i> , data yang tersimpan dapat di lihat, diubah, dan dihapus.. | Sesuai |
| D02 | Mengunduh laporan dengan mengklik tombol ' <i>Export PDF</i> ', ' <i>Export Excel</i> '. | Laporan terunduh sesuai dengan bentuk file yang dipilih. | Sesuai |
| E01 | Mengisi seluruh data produk yang diperlukan dan menekan tombol " <i>Submit</i> ". | Hasil data yang diinput akan masuk ke dalam <i>database</i> . | Sesuai |
| E02 | Mengisi kolom harga dengan menggunakan huruf dan tanda baca (.). | Muncul pesan kesalahan pada kolom harga, pengguna diharuskan untuk menginput harga. | Sesuai |
| E03 | Mengisi semua kolom yang ada tetapi kolom <i>quantity</i> tidak di isi. | Muncul pesan kesalahan pada kolom <i>quantity</i> , pengguna diharuskan untuk menginput jumlah barang. | Sesuai |
| E04 | Mengisi semua kolom kecuali <i>category</i> dan tidak memasukan gambar | Muncul pesan kesalahan pada kolom kategori dan gambar barang, pengguna diharuskan untuk menginput kategori barang dan gambar barang. | Sesuai |
| E05 | Mengisi semua kolom dengan data yang baru tetapi pada kolom nama | Muncul pesan kesalahan pada kolom nama yang memberitahu | Sesuai |

| | | | |
|-----|---|--|--------|
| | dibuat sama dengan data yang sudah ada | pengguna bahwa nama tersebut sudah ada di <i>database</i> . | |
| F01 | Mengisi semua kolom dengan data baru dan menekan tombol 'Submit' | Data pelanggan berhasil tersimpan ke dalam <i>database</i> . | Sesuai |
| F02 | Mengisi semua kolom dengan data yang berbeda, tetapi data nama menggunakan data yang sudah ada sebelumnya. | Pengguna dapat melakukan <i>submit</i> dan data pelanggan berhasil tersimpan ke dalam <i>database</i> . | Sesuai |
| F03 | Mengisi kolom <i>email</i> dengan menggunakan email yang sudah ada sebelumnya. | Muncul pesan kesalahan pada kolom <i>email</i> , pengguna diminta untuk memasukkan <i>email</i> yang baru. | Sesuai |
| F04 | Mengisi semua kolom kecuali kolom <i>phone</i> dan 'Submit'. | Muncul pesan kesalahan pada kolom <i>phone</i> , pengguna diminta memasukkan nomor telepon. | Sesuai |
| F05 | Mengisi semua kolom kecuali kolom <i>email</i> dan 'Submit'. | Berhasil menyimpan data pelanggan ke <i>database</i> . | Sesuai |
| G01 | Mengisi semua kolom dengan data baru dan menekan tombol 'Submit' | Data <i>suppliers</i> berhasil tersimpan ke dalam <i>database</i> . | Sesuai |
| G02 | Mengisi semua kolom dengan data yang berbeda, tetapi kolom nama diisi menggunakan data yang sudah ada sebelumnya. | Pengguna dapat melakukan <i>submit</i> dan data <i>suppliers</i> berhasil tersimpan ke dalam <i>database</i> . | Sesuai |
| G03 | Mengisi kolom <i>email</i> dengan menggunakan email yang sudah ada sebelumnya. | Muncul pesan kesalahan pada kolom <i>email</i> , pengguna diminta untuk memasukkan <i>email</i> yang baru. | Sesuai |
| G04 | Mengisi semua kolom kecuali kolom <i>phone</i> dan 'Submit'. | Gagal menyimpan data dan muncul pesan kesalahan pada kolom <i>phone</i> , pengguna diminta memasukkan nomor telepon. | Sesuai |
| G05 | Mengisi semua kolom kecuali kolom <i>email</i> dan 'Submit'. | Berhasil menyimpan data <i>suppliers</i> ke <i>database</i> . | Sesuai |
| H01 | Mengisi semua kolom dengan benar dan mengklik tombol 'Submit'. | Data persediaan masuk berhasil tersimpan ke <i>database</i> . | Sesuai |

| | | | |
|-----|---|--|--------|
| H02 | Mengisi semua kolom kecuali kolom 'Products' dan melakukan 'Submit'. | Gagal menyimpan data dan muncul pesan kesalahan pada kolom 'products', pengguna diminta memilih produk yang ingin disimpan. | Sesuai |
| H03 | Mengisi semua kolom kecuali kolom 'Suppliers' dan melakukan 'Submit'. | Gagal menyimpan data dan muncul pesan kesalahan pada kolom 'suppliers', pengguna diminta memilih supplier yang ingin disimpan. | Sesuai |
| H04 | Mengisi semua kolom kecuali kolom 'Quantity' dan melakukan 'Submit'. | Gagal menyimpan data dan muncul pesan kesalahan pada kolom 'quantity', pengguna diminta mengisi jumlah barang yang ingin ditambahkan. | Sesuai |
| H05 | Mengisi semua kolom kecuali kolom 'Date' dan melakukan 'Submit'. | Gagal menyimpan data dan muncul pesan kesalahan pada kolom 'Date', pengguna diminta mengisi tanggal persediaan masuk. | Sesuai |
| I01 | Mengisi semua kolom dengan benar dan mengklik tombol 'Submit'. | Data persediaan masuk berhasil tersimpan ke <i>database</i> . | Sesuai |
| I02 | Mengisi semua kolom kecuali kolom 'Products' dan melakukan 'Submit'. | Gagal menyimpan data dan muncul pesan kesalahan pada kolom 'products', pengguna diminta melakukan pengurangan. | Sesuai |
| I03 | Mengisi semua kolom kecuali kolom 'Customer' dan melakukan 'Submit'. | Gagal menyimpan data dan muncul pesan kesalahan pada kolom 'customer', pengguna diminta memilih nama pelanggan yang membeli produk tersebut. | Sesuai |
| I04 | Mengisi semua kolom kecuali kolom 'Quantity' dan melakukan 'Submit'. | Gagal menyimpan data dan muncul pesan kesalahan pada kolom 'quantity', pengguna diminta mengisi jumlah barang yang dijual. | Sesuai |
| I05 | Mengisi semua kolom kecuali kolom 'Date' dan melakukan 'Submit'. | Gagal menyimpan data dan muncul pesan kesalahan pada kolom 'Date', pengguna diminta mengisi tanggal persediaan keluar. | Sesuai |

Dari hasil pengujian yang dilihat dari tabel diatas, pengujian yang dilakukan semuanya sesuai dengan harapan penguji aplikasi. Pengujian ini memastikan bahwa aplikasi tersebut dapat berjalan dengan baik dan layak digunakan.

Kesimpulan

Berdasarkan hasil dari pengujian yang telah dilakukan dari aplikasi manajemen persediaan pada CV. Anugerah Sejahtera, maka kesimpulan yang didapatkan yaitu dengan pengujian menggunakan *Black Box* dengan metode *Equivalence Partitioning* memberikan kemudahan bagi penguji dalam mengukur seberapa layak aplikasi dapat dioperasikan, hal ini sekaligus menunjukkan bahwa aplikasi manajemen persediaan yang dibuat lolos dari tahapan pengujian dan layak untuk dioperasikan pada perusahaan.

Referensi

- [1] M. K. MZ, “PENGUJIAN PERANGKAT LUNAK METODE BLACK-BOX BERBASIS EQUIVALENCE PARTITIONS PADA APLIKASI SISTEM INFORMASI SEKOLAH,” *J. Mikrotik*, vol. 06, no. 3, pp. 02–16, 2016.
- [2] M. G. A. Sunarso and A. Saifudin, “Pengujian Black Box Pada Aplikasi System Inventory Warehouse Berbasis Desktop Menggunakan Metode Equivalence Partitioning,” *J. Ilmu Komput. dan Pendidik.*, vol. 2, no. 2, pp. 320–324, 2024, [Online]. Available: <https://journal.mediapublikasi.id/index.php/logic>
- [3] B. A. Maulana, E. Mawarni, M. Y. Hidayattuloh, V. Suryawijaya, and A. Saifudin, “Pengujian Black Box pada Sistem Informasi Barang Berbasis Web Menggunakan Metode Boundary Value Analysis,” *OKTAL J. Ilmu Komput. dan Sci.*, vol. 2, no. 6, pp. 1747–1753, 2023.
- [4] T. Hidayat and M. Muttaqin, “Pengujian Sistem Informasi Pendaftaran dan Pembayaran Wisuda Online menggunakan Black Box Testing dengan Metode Equivalence Partitioning dan Boundary Value Analysis,” *J. Tek. Inform. UNIS*, vol. 6, no. 1, pp. 2252–5351, 2018, [Online]. Available: www.ccssenet.org/cis
- [5] A. H. Albaqir, A. R. Putra, and A. Saifudin, “Pengujian Aplikasi Pembelian Tiket Online menggunakan Metode Pengujian Black Box Testing Berbasis Equivalence Partitions,” *J. Ris. Inform. dan Inov.*, vol. 1, no. 4, pp. 1–9, 2023.
- [6] A. Rosalina, A. A. G. R. Rassi, G. Y. Hadi, R. Ubaidillah, and T. Desyani, “Pengujian Black Box pada Sistem Informasi Penjualan HI Shoe Store Menggunakan Teknik Equivalence Partitions,” *J. Teknol. Sist. Inf. dan Apl.*, vol. 3, no. 1, p. 9, 2020, doi: 10.32493/jtsi.v3i1.4303.
- [7] M. S. A. Kesuma Jaya, P. Gumilang, Tresnawati, Y. P. Andersen, and T. Desyani, “Pengujian Black Box pada Aplikasi Sistem Penunjang Keputusan Seleksi Calon Pegawai Negeri Sipil Menggunakan Teknik Equivalence Partitions,” *J. Inform. Univ. Pamulang*, vol. 4, no. 4, p. 131, 2019, doi: 10.32493/informatika.v4i4.3834.
- [8] Uminingsih, M. Nur Ichsanudin, M. Yusuf, and Suraya, “PENGUJIAN FUNGSIONAL PERANGKAT LUNAK SISTEM INFORMASI PERPUSTAKAAN DENGAN METODE BLACK BOX TESTING BAGI PEMULA,” *STORAGE J. Ilm. Tek. dan Ilmu Komput.*, vol. 1, no. 2, pp. 1–8, 2022, doi: 10.55123/storage.v1i2.270.

- [9] D. Setiawan, M. A. Fadhillah, A. Wibawa, I. Sugiarto, A. Mulyana, and I. Kusyadi, “Pengujian Black Box pada Aplikasi Perpustakaan Berbasis Web Menggunakan Teknik Equivalence Partitioning,” *J. Teknol. Sist. Inf. dan Apl.*, vol. 3, no. 2, p. 95, 2020, doi: 10.32493/jtsi.v3i2.3955.