

BLACK BOX TESTING APLIKASI POINT OF SALES POST

Danendra Khansa Pallas Wahyudi ¹⁾

Sistem Informasi Universitas Ma Chung, Villa Puncak Tidar N-1, Malang, Indonesia
email : 321710002@student.machung.ac.id¹⁾

Abstrak

Black box testing terhadap aplikasi POST dilakukan untuk melihat apakah aplikasi POST yang masih terbilang baru ini layak digunakan, terutama karyawan kasir yang lebih berpotensi melakukan kesalahan yang disengaja maupun tidak disengaja. Dipilihnya aplikasi POST karena tersedia pilihan paket gratis yang memiliki fitur yang lebih banyak dibanding kompetitor kebanyakan. Dipilihnya metode black box testing karena tidak memungkinkan untuk menggunakan metode white box testing yang memerlukan pengetesan sampai dengan pemrograman yang digunakan dalam aplikasi. Teknik yang digunakan dalam pengujian pada aplikasi POST adalah equivalence partitioning. Dipilihnya equivalence partitioning karena teknik tersebut adalah teknik yang paling mudah dan paling banyak digunakan dalam melakukan testing. Hasil penelitian menunjukkan bahwa aplikasi POST dapat dikatakan layak karena tingkat severity yang diperoleh adalah S2 yang masih dapat ditoleransi. Dari 256 test case, yang dinyatakan gagal adalah 24 test case. Total halaman yang di tes berjumlah 100 halaman dan test case yang dinyatakan gagal terdapat pada 19 halaman aplikasi.

Kata Kunci :

Testing, Black box testing, Equivalence partitioning, Point of sales

Abstract

Black box testing of the POST application is carried out to see whether the POST application which is still relatively new is suitable for use, especially cashier employees who have the potential to make mistakes, intentional or unintentional. The POST application was chosen because there was a choice of free packages that had more features than most competitors. The black box testing method was chosen because it was not possible to use the white box testing method which required testing up to the programming used in the application. The technique used in testing the POST application is equivalence partitioning. Equivalence partitioning was chosen because this technique is the easiest technique and the most widely used in testing. The output from this research showed that the POST application can be said to be feasible because the severity level obtained is S2 which can still be tolerated. From 256 test cases, 24 test cases failed. The total number of pages tested is 100 pages and test cases that have failed are found on 19 application pages.

Keywords :

Testing, Black box testing, Equivalence partitioning, Point of sales

1. PENDAHULUAN

Aplikasi POST diluncurkan pada tanggal 17 Januari 2020 oleh PT Payfazz Teknologi Nusantara yang berlokasi di Jakarta. Berdasarkan halaman *website* resmi PT Payfazz Teknologi Nusantara atau biasa yang dikenal sebagai PAYFAZZ merupakan perusahaan pengembang teknologi finansial [1]. Kelebihan aplikasi POST dibandingkan dengan aplikasi *point of sales* yang lain adalah pilihan paket gratis yang memiliki fitur yang lebih banyak dibanding kompetitor kebanyakan, harga POST Premium terbaru yang dicantumkan di *website* resminya seharga Rp 58.000 per bulan dan gratis pemakaian selama 30 hari dan tersedia juga versi *lite* yang gratis. Kelebihan lain dari aplikasi POST yaitu hanya dengan paket yang gratis pengguna mendapatkan fitur penyimpanan data yang tidak terbatas sehingga pengguna dapat memasukkan data yang sangat banyak seperti data produk, transaksi, dan data pelanggan [2].

Sistem informasi merupakan sistem yang memiliki tujuan untuk memberikan informasi kepada kelompok orang pada suatu bagian dalam sebuah perusahaan [3]. Sedangkan menurut sumber lain, Sistem informasi merupakan kumpulan prosedur dalam organisasi yang pada saat dijalankan akan memberikan informasi kepada pengambil keputusan atau untuk mengendalikan organisasi [4].

Testing merupakan tahap evaluasi yang memiliki tujuan melihat level kemudahan pengguna saat menggunakan sebuah perangkat lunak yang berdasarkan indikator yang sudah ditentukan[5]. Sedangkan sumber lain mengatakan bahwa pengujian atau testing sangat penting dilakukan karena memiliki tujuan untuk menjamin kualitas dari perangkat lunak, dan juga bisa menjadi peninjauan terakhir terhadap pengkodean, desain, dan spesifikasi[6].

Testing yang dilakukan mencakup keseluruhan modul dari aplikasi *owner* dan kasir sehingga ruang lingkup pengetesan sesuai dengan metode *black box* testing yang hanya menguji tampilan dan fungsi dari aplikasi tidak seperti *white box* testing yang menguji sampai dengan pemrograman aplikasi dan basis data yang digunakan.

Sehingga tujuan dilakukannya proyek ini adalah untuk melakukan *black box* testing terhadap aplikasi POST dan membuat dokumen berdasarkan hasil testing yang telah dilakukan. Selain itu testing juga bertujuan untuk melihat apakah aplikasi POST yang masih terbilang baru ini layak digunakan, terutama karyawan kasir yang lebih berpotensi melakukan kesalahan yang disengaja maupun tidak disengaja.

2. METODE / ALGORITMA

Metode yang digunakan dalam melakukan testing terhadap aplikasi POST adalah metode *black box* testing. Dipilihnya metode *black box* testing karena tidak memungkinkan untuk menggunakan metode *white box* testing yang memerlukan pengetesan sampai dengan pemrograman yang digunakan dalam aplikasi. *Black box* testing disebut juga dengan teknik pada testing perangkat lunak yang bertujuan menguji spesifikasi fungsional dari sebuah perangkat lunak. Teknik *black box* testing ini tidak melihat struktur kontrol dari perangkat lunak dalam penerapannya sehingga berfokus pada informasi di domain perangkat lunak [7]. Sedangkan sumber lain mengatakan bahwa *black box* testing merupakan pengujian pada fungsi operasional dalam perangkat lunak [6].

Teknik yang digunakan dalam pengujian pada aplikasi POST adalah *equivalence partitioning* yang melakukan pengujian dengan cara mencoba berbagai masukan dan melihat apakah data yang diterima oleh sistem seharusnya valid atau tidak. *Equivalence partitioning* merupakan salah satu metode dalam *black box* testing yang membagi domain masukan suatu program menjadi beberapa kelas data di mana sebuah *test case* dapat diturunkan [8]. Sedangkan sumber lain mengatakan bahwa *Equivalence partitioning* merupakan teknik untuk menguji data yang bisa diaplikasikan di mana data masukan atau data keluaran dari program dibagi menjadi kategori atau partisi yang berbeda sehingga *test case* bisa dikembangkan untuk setiap kategori data masukan atau keluaran [9].

Standar yang digunakan dalam melakukan testing adalah standar ISO/IEC/IEEE 29119-3:2013. *Test case* hasil pengujian terdiri dari *test data*, *expected result*, *actual result*, status, catatan, dan tampilan.

Data hasil pengujian diperoleh dari hasil testing pada aplikasi POST. Data yang digunakan untuk menguji aplikasi bersifat acak seperti penggunaan simbol, angka, dan huruf untuk melihat apakah aplikasi berjalan dengan semestinya atau tidak. Pada hasil testing juga disertakan gambar pada saat melakukan testing untuk memperjelas apa yang terjadi pada saat melakukan testing.

Setelah testing dilakukan maka akan ditentukan tingkat *severity* yang diperoleh. Tingkat *severity* digunakan untuk memberitahukan seberapa parah dampak *bug* bagi aplikasi yang

ditemukan selama testing. *Severity* didefinisikan atas dasar bagaimana pelanggan akan terpengaruh nantinya jika *bug* terjadi selama produksi, *severity* diindikasikan sebagai S1, S2, S3, dan S4 di mana semakin kecil maka semakin parah *bug* yang ditemukan [9]. Sedangkan sumber lain mengatakan bahwa *severity* mendeskripsikan seberapa parah kesalahan dari hasil akhir untuk dilakukan perbaikan yang perlu dilakukan untuk sistem [10].

Permasalahan yang terjadi pada saat melakukan testing adalah adanya pembaruan aplikasi selama pengerjaan sehingga ada berbagai perubahan pada aplikasi, untuk mengatasi masalah tersebut maka pada pengetesan dicantumkan versi aplikasi sehingga hasil pengetesan sesuai dengan versi saat itu.

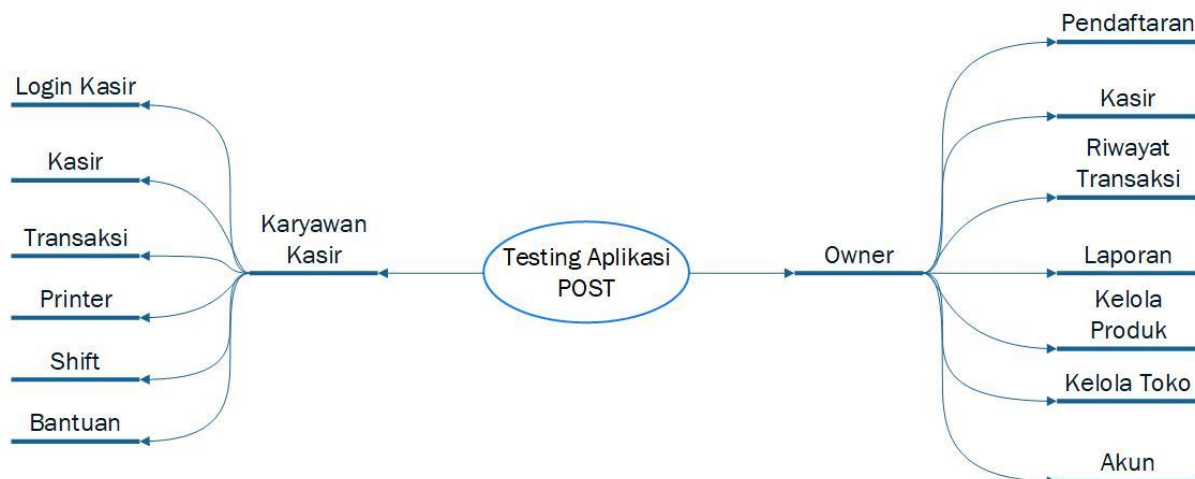
Berdasarkan penelitian terdahulu yang berjudul “Implementasi Black Box Testing Pada Sistem Informasi Pendaftaran Santri Berbasis Web Dengan Menggunakan Php Dan Mysql”, mekanisme atau kriteria pengujian perangkat lunak terbagi menjadi tiga bagian [11]. Berikut adalah bagian-bagian dari mekanisme pengujian perangkat lunak berdasarkan penelitian yang telah disebutkan.

- a. Pengetesan program aplikasi dengan cara menjalankan aplikasi.
- b. Pengetesan pemasukan data, perubahan data, dan penghapusan data.
- c. Pengetesan terhadap fungsi tombol yang terdapat pada masing-masing halaman apakah berfungsi dengan baik.

Jika berdasarkan salah satu dari ketiga kriteria tersebut aplikasi tidak menunjukkan hasil yang semestinya maka *test case* dinyatakan gagal.

Terdapat dua opsi *login* pada aplikasi POST yaitu sebagai *owner* dan sebagai kasir, oleh karena itu hasil pengetesan dibagi menjadi dua bagian yaitu aplikasi *owner* dan aplikasi kasir. Tidak semua menu yang ada di aplikasi kasir akan dilakukan pengetesan karena ada beberapa fitur yang sudah dilakukan pengetesan pada aplikasi *owner* dan memiliki fungsi yang sama.

Berikut adalah *mind map* dari struktur menu aplikasi *owner* dan karyawan kasir pada aplikasi POST.



Gambar 2.1 *Mind map* Struktur Menu Aplikasi POST

3. HASIL DAN PEMBAHASAN

Hasil pengetesan tidak ditampilkan secara keseluruhan dalam penelitian karena hasil pengetesan terdiri dari 256 *test case*, oleh karena itu hasil pengetesan yang ditampilkan hanyalah beberapa bagian yang paling penting untuk diperbaiki dari aplikasi POST.

Berikut adalah keterangan dari kriteria yang terdapat pada setiap *test case*.

A = Pengetesan aplikasi dengan melihat apakah aplikasi berjalan dengan baik.

B = Pengetesan pemasukan data, perubahan data, dan penyimpanan data.

C = Pengetesan terhadap fungsi tombol yang terdapat pada masing-masing halaman apakah berfungsi dengan baik.

3.1 Aplikasi Owner

Project Name : POST
Module Name : Owner
Test Designed date : 23-04-2020
Test Executed by : Danendra
Test Execution Date : 23-04-2020

3.1.1 Pendaftaran Akun

Release Version : APP POST. V1.4.0 (17)
Pre-condition : Aplikasi sudah terinstal pada perangkat android
Test priority : High
Test title : Pendaftaran akun
Test summary : Pengguna tetap dapat mendaftar meskipun data yang dimasukkan pengguna palsu.

Test Case : 1
Test Data : Password: qwertyui (password hanya berisi huruf)
Expected Result : Tidak diperbolehkan
Actual result : Diperbolehkan
 Status : Gagal
 Catatan : *Password* setidaknya harus terdiri dari huruf dan angka atau simbol
 Kriteria : B
 Tampilan :

The screenshot shows a mobile application interface for creating an account. At the top, there is a back arrow and the title 'Buat Akun'. Below the title, there are three input fields: 'Email' with a placeholder '+-&%\$#@123.com', 'Nomor Handphone Akun' with a placeholder '085431659739', and 'Password' with a placeholder 'qwertyui' and a visibility toggle icon. At the bottom of the form, there is a dark blue button labeled 'LANJUT'.

Gambar 3.1. Password Hanya Berisi Huruf

3.1.2 Kasir (Pesanan)

Release Version : APP POST. V1.4.0 (17)

Pre-condition : Data produk sudah tersimpan dalam aplikasi

Test priority : High

Test title : Kasir (Pesanan)

Test summary : Setelah menyimpan pesanan sebaiknya diberikan notifikasi bahwa pesanan telah tersimpan untuk memberitahukan kepada pengguna apakah pesanan telah tersimpan

Test Case : 1

Test Data : Menekan tombol “Simpan”

Expected Result : Mengubah data Pesanan Tersimpan dan Kembali ke tampilan Semua Produk dengan menampilkan pesan bahwa pesanan telah tersimpan

Actual result : Mengubah data Pesanan Tersimpan dan Kembali ke tampilan Semua Produk

Status : Gagal

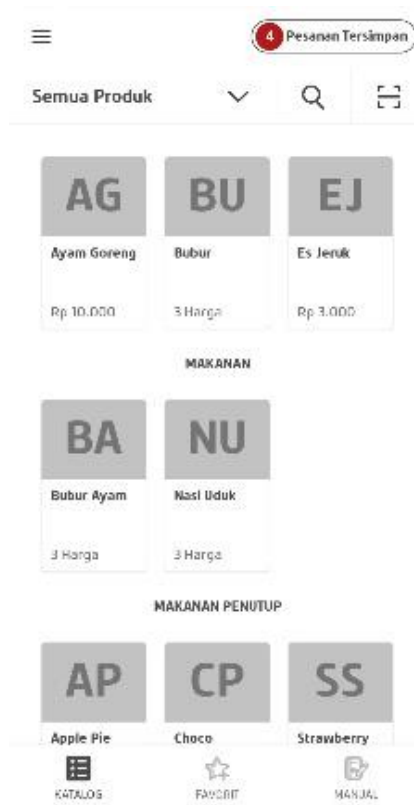
Catatan : Sebaiknya setelah menyimpan data pesanan, aplikasi memberikan pesan bahwa data telah tersimpan untuk memberitahukan pengguna bahwa pesanan telah tersimpan

Kriteria : C

Tampilan :



Gambar 3.2 Tampilan Sebelum Menyimpan Pesanan



Gambaran 3.3 Tampilan Setelah Menyimpan Pesanan

3.1.3 Riwayat Transaksi

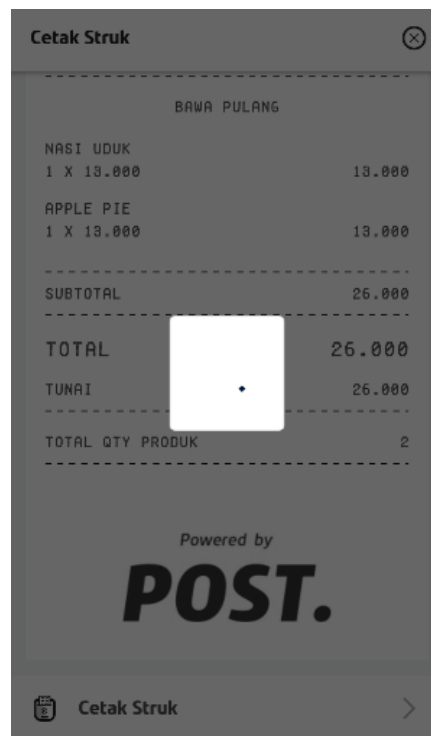
- Release Version* : APP POST. V1.4.0 (17)
- Pre-condition* : Sudah pernah melakukan transaksi
- Test priority* : *Medium*
- Test title* : Riwayat transaksi
- Test summary* : Sistem tidak dapat mencetak setruk bukti pembayaran, kemungkinan disebabkan perangkat yang digunakan tidak saling mendukung

- Test Case* : 1
- Test Data* : Membuka aplikasi POST
- Expected Result* : Aplikasi terbuka
- Actual result* : Aplikasi hanya terbuka pada tampilan loading screen
- Status* : Gagal
- Catatan* : Setelah aplikasi tidak dibuka selama beberapa hari, saat dibuka aplikasi hanya menampilkan loading screen dan tidak berubah setelah ditunggu beberapa saat, setelah menutup aplikasi dan membukanya kembali aplikasi berjalan dengan normal
- Kriteria* : A
- Tampilan* :



Gambar 3.4 Aplikasi POST Hanya Menampilkan *Loading Screen*

- Test Case : 2
- Test Data : Menekan tombol “Cetak setruk”
- Expected Result : Sistem mencetak setruk
- Actual result : Sistem tidak mencetak setruk
- Status : Gagal
- Catatan : Setruk tidak dapat tercetak, mungkin dikarenakan perangkat yang digunakan tidak saling mendukung
- Kriteria : C
- Tampilan :



Gambar 3.5 Setruk Tidak Dapat Tercetak

3.1.5 Kelola Toko

Release Version : APP POST. V1.4.0 (17)

Pre-condition : Pengguna sudah masuk menggunakan akun *owner* dan berada pada halaman kelola toko

Test priority : *Medium*

Test title : Kelola toko

Test summary : Ada beberapa hal yang perlu diperbaiki pada menu kelola toko, salah satunya yang terpenting adalah pengaturan penghitungan pajak & *service charge* terbalik antara yang termasuk pajak & *service charge* dan yang tidak termasuk pajak & *service charge*.

Test Case : 1

Test Data : Memilih pengaturan perhitungan “harga tidak termasuk pajak & *service charge*” dan menyimpan pengaturan

Expected Result : Pajak & *service charge* tidak ditambahkan dalam pesanan

Actual result : Pajak & *service charge* ditambahkan dalam pesanan

Status : Gagal

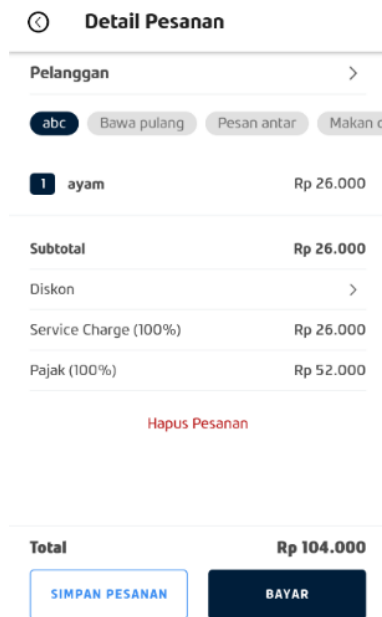
Catatan : Jika memilih “harga tidak termasuk pajak & *service charge*” semestinya Pajak & *service charge* tidak ditambahkan dalam pesanan, tetapi aplikasi malah menambahkannya pada pesanan

Kriteria : C

Tampilan :

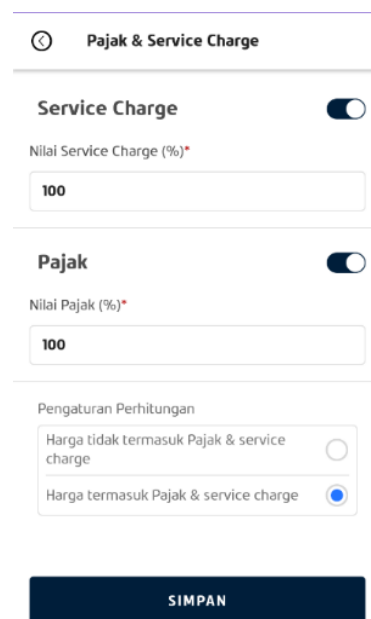
The screenshot shows a mobile application interface for managing taxes and service charges. At the top, there is a back arrow and the title 'Pajak & Service Charge'. Below this, there are two main sections. The first section is 'Service Charge', which has a toggle switch turned on and an input field containing the number '100'. The second section is 'Pajak', also with a toggle switch turned on and an input field containing '100'. Below these sections is a 'Pengaturan Perhitungan' (Calculation Settings) section with two radio button options: 'Harga tidak termasuk Pajak & service charge' (which is selected) and 'Harga termasuk Pajak & service charge'. At the bottom of the screen is a dark blue button labeled 'SIMPAN' (Save).

Gambar 3.6 Menyimpan Pengaturan Yang Tidak Termasuk Pajak & Service



Gambar 3.7 Aplikasi Menambahkan Pajak & Service Charge Pada Pesanan

<i>Test Case</i>	: 2
<i>Test Data</i>	: Memilih pengaturan perhitungan “harga termasuk pajak & service charge” dan menyimpan pengaturan
<i>Expected Result</i>	: Pajak & service charge ditambahkan dalam pesanan
<i>Actual result</i>	: Pajak & service charge tidak ditambahkan dalam pesanan
<i>Status</i>	: Gagal
<i>Catatan</i>	: Jika memilih “harga termasuk pajak & service charge” semestinya pajak & service charge ditambahkan dalam pesanan, tetapi aplikasi malah tidak memasukkan pajak & service charge
<i>Kriteria Tampilan</i>	: C



Gambar 3.8 Menyimpan Pengaturan Yang Termasuk Pajak & Service

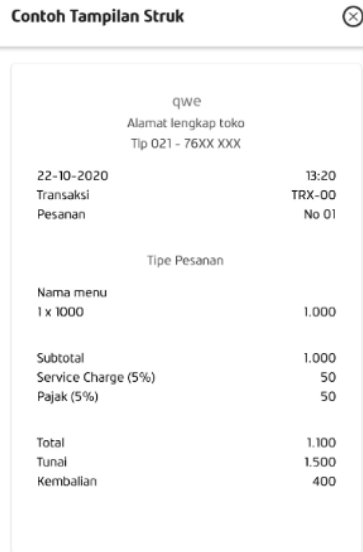


Gambar 3.9 Aplikasi Menambahkan Pajak & Service Charge Pada Pesanan

<i>Test Case</i>	: 3
<i>Test Data</i>	: Menekan tombol “contoh tampilan struk” tanpa menyimpan terlebih dahulu gambar yang telah di unggah dan catatan kaki setruk yang telah dimasukkan
<i>Expected Result</i>	: Menampilkan tampilan setruk dengan gambar yang telah di unggah dan catatan kaki setruk yang telah dimasukkan
<i>Actual result</i>	: Tidak menampilkan gambar dan catatan kaki
Status	: Gagal
Catatan	: Pengguna biasanya ingin melihat contoh tampilan dengan gambar dan catatan sebelum disimpan
Kriteria	: C
Tampilan	:

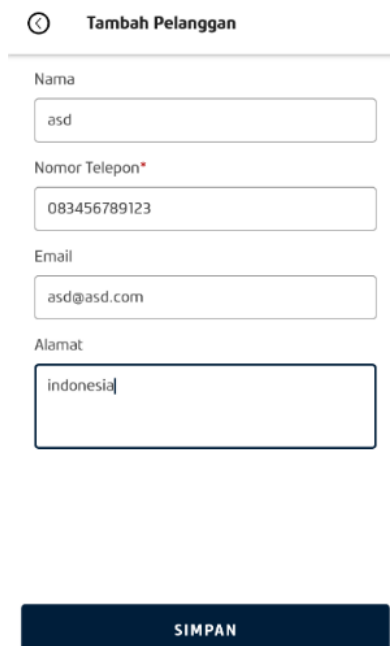


Gambar 3.10 Tampilan Sebelum Menekan Tombol “Contoh Tampilan Struk”

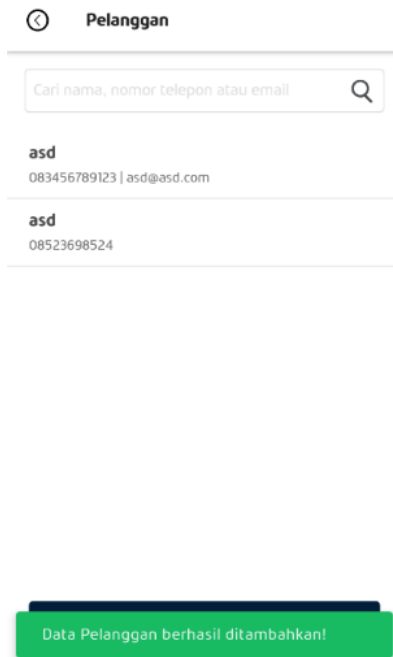


Gambar 3.11 Contoh Tampilan Struk Tanpa Gambar Dan Catatan

- Test Case* : 4
- Test Data* : Menyimpan data dengan nama pelanggan yang masih sama dengan nama pelanggan lain
- Expected Result* : Tidak diperbolehkan
- Actual result* : Tersimpan
- Status : Gagal
- Catatan : Nama pelanggan seharusnya tidak boleh benar-benar sama agar tidak terjadi kesalahan dalam pemesanan
- Kriteria : B
- Tampilan :

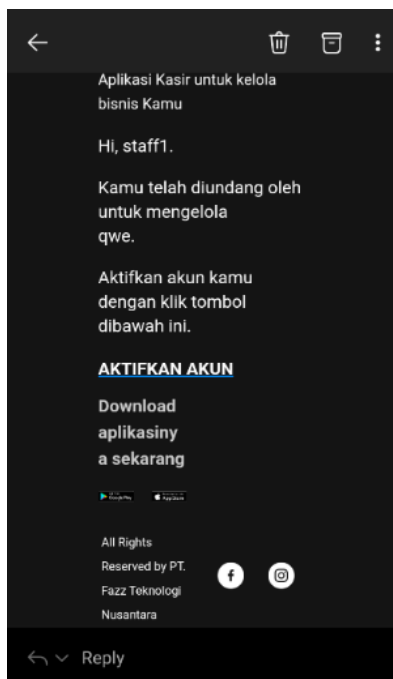


Gambar 3.12 Nama Pelanggan Sama Seperti Nama Pelanggan Lain

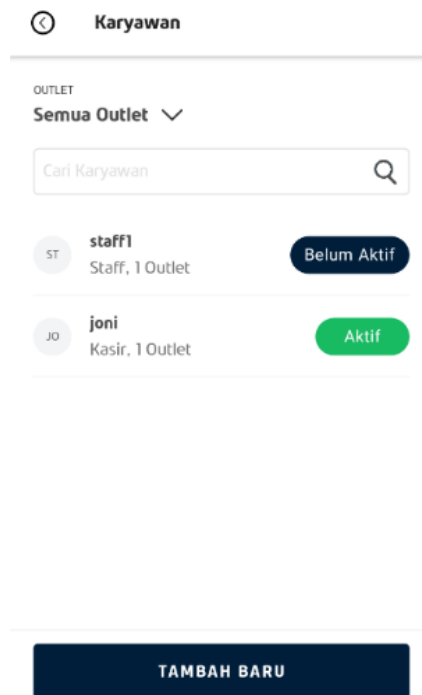


Gambar 3.13 Data Nama Pelanggan Sama Seperti Nama Pelanggan Lain Tersimpan

- Test Case : 5
- Test Data : Mengaktifkan akun staff melalui email
- Expected Result : Akun staff aktif
- Actual result : Akun staff tidak aktif
- Status : Gagal
- Catatan : Setelah menekan link pada email aplikasi POST hanya terbuka dan tidak mengaktifkan akun staff
- Kriteria : A
- Tampilan :

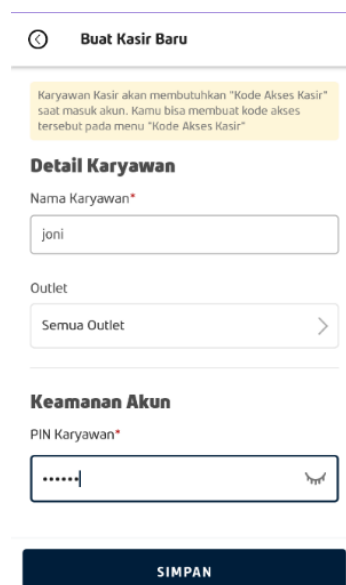


Gambar 3.14 Email Pengaktifan Akun Staff

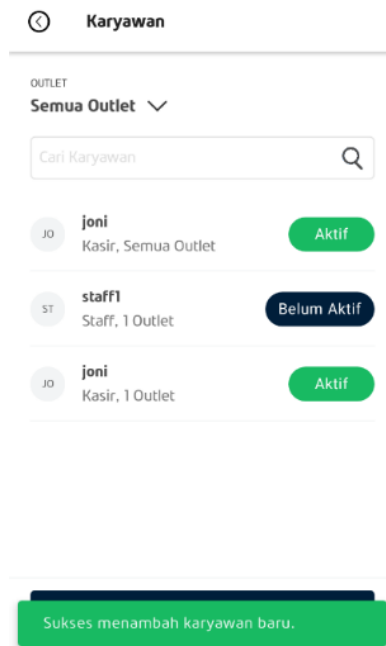


Gambar 3.15 Akun Staff Masih Tidak Aktif

<i>Test Case</i>	: 6
<i>Test Data</i>	: Mengisi data karyawan dengan nama kasir sama seperti yang sudah pernah ditambahkan dan menyimpan data
<i>Expected Result</i>	: Data tidak berhasil disimpan
<i>Actual result</i>	: Data berhasil disimpan
Status	: Gagal
Catatan	: Nama karyawan kasir seharusnya tidak boleh benar-benar sama
Kriteria	: B
Tampilan	:



Gambar 3.16 Menambahkan Data Karyawan Kasir



Gambar 3.17 Karyawan Kasir Berhasil Ditambahkan

3.2 Aplikasi Kasir

Project Name : POST

Module Name : Kasir

Test Designed date : 23-04-2020

Test Executed by : Danendra

Test Execution Date : 12-10-2020

Release Version : v1.16.0

Pre-condition : Pengguna sudah masuk sebagai kasir

Test priority : *Medium*

Test title : *Shift*

Test summary : Setelah *shift* diakhiri kemudian *login* kembali dengan akun karyawan untuk melihat riwayat *shift* hilang termasuk riwayat di hari sebelumnya, selain itu sebaiknya *owner* juga dapat karyawan karena saat ini *owner* tidak bisa melihat riwayat *shift* karyawan.

Test Case : 1

Test Data : Memeriksa apakah *shift* tersimpan di riwayat shift

Expected Result : Menampilkan data *shift* sebelumnya

Actual result : Semua data *shift* hilang

Status : Gagal

Catatan : Riwayat shift karyawan tidak tampil termasuk riwayat yang ditambahkan beberapa hari sebelumnya

Kriteria : A

Tampilan :



Gambar 3.18 Semua Riwayat Shift Hilang

3.3 Penghitungan Hasil Testing

Dari 256 *test case* yang telah dilakukan, kategori pengetesan aplikasi dengan melihat apakah aplikasi berjalan dengan baik berjumlah 7 *test case* dan 3 *test case* dinyatakan gagal. Sedangkan kategori pengetesan pemasukan data, perubahan data, dan penyimpanan data berjumlah 108 *test case* dan 16 *test case* dinyatakan gagal. Kategori pengetesan terakhir yaitu pengetesan terhadap fungsi tombol yang terdapat pada masing-masing halaman berjumlah 141 *test case* dan 5 *test case* dinyatakan gagal. Dari semua *test case* yang telah dilakukan, *test case* yang dinyatakan gagal berjumlah 24 dan yang dinyatakan berhasil berjumlah 232 *test case*.

3.4 Tingkat Severity

Tingkat *severity* yang ditentukan dari skala S1 sampai dengan S3 berdasarkan *bug* atau kesalahan yang ditemui selama melakukan testing pada aplikasi POST masih pada tingkat S2 (*Severity* 2) yaitu kesalahan yang ditemukan masih dapat ditoleransi. Kesalahan yang ditemukan masih dapat ditoleransi karena beberapa kesalahan tidak mengganggu penggunaan aplikasi secara keseluruhan seperti pengaturan yang terbalik antara mengaktifkan dan menonaktifkan pajak & *service charge*, tetapi ada juga kesalahan yang dapat menyusahkan pengguna seperti laporan *shift* yang tiba-tiba hilang.

4. KESIMPULAN

Kesimpulan dari dilakukannya testing pada aplikasi POST adalah walaupun aplikasi POST masih terbilang cukup baru dibanding kompetitor yang lain, POST sudah cukup layak untuk digunakan oleh para pemilik usaha sebagai aplikasi *point of sales* pada usahanya. Aplikasi POST dapat dikatakan layak karena berdasarkan tingkat *severity* yang diperoleh adalah S2 yang masih dapat ditoleransi, akan tetapi masih ada beberapa hal yang perlu diperbaiki pada aplikasi POST agar kualitas semakin meningkat dan pengalaman pengguna menjadi lebih baik. Dari

256 *test case*, yang dinyatakan gagal adalah 24 *test case*. Total halaman yang di tes berjumlah 100 halaman dan *test case* yang dinyatakan gagal terdapat pada 19 halaman aplikasi.

Berikut ini adalah beberapa saran agar aplikasi POST menjadi lebih baik lagi.

1. Sebaiknya fitur yang diperoleh pengguna menyesuaikan kategori bisnis yang dipilih oleh pengguna, karena saat ini kategori bisnis yang dipilih tidak berpengaruh pada aplikasi.
2. Beberapa saran sudah tertera pada catatan yang ada di beberapa tabel *test case* saat melakukan testing sehingga mungkin pihak pengembang dapat mempertimbangkan catatan-catatan tersebut.
3. Penggunaan kata perlu diperhatikan agar sesuai dengan Bahasa Indonesia yang benar, seperti contohnya dalam aplikasi dituliskan struk sedangkan menurut Kamus Besar Bahasa Indonesia (KBBI) penulisannya adalah setruk.

5. REFERENSI

- [1] “Tentang PAYFAZZ – Informasi Lengkap Seputar Aplikasi dan Perusahaan.” <https://www.payfazz.com/tentang-payfazz> (diakses Jul 16, 2020).
- [2] “POST. | Aplikasi Kasir Online Lengkap POST - Harga Baru – POST.” <https://post.app/new-pricing/> (diakses Agu 19, 2020).
- [3] E. Yunaeti dan R. Irvani, *Pengantar Sistem Informasi - Elisabet Yunaeti Anggraeni*. Penerbit Andi, 2017.
- [4] A.-B. Bin Ladjamudin, *Analisis dan Desain Sistem Informasi*. Tangerang: Graha Ilmu, 2005.
- [5] J. Rubin dan D. Chiisnell, *Handbook of usability testing: How to plan, design, and conduct effective tests*, Second., vol. 17, no. 2. Wiley Publishing, 2008.
- [6] F. H. Utami dan Asnawati, *Rekayasa Perangkat Lunak*. Deepublish, 2015.
- [7] J. Watkins dan S. Mills, *Testing IT: An off-the-shelf software testing process*. Cambridge University Press, 2011.
- [8] R. S. Pressman, *Software Quality Engineering: A Practitioner’s Approach*. The McGraw-Hill Companies, 2010.
- [9] M. G. Limaye, *Software Testing*. Tata McGraw-Hill Education, 2009.
- [10] B. S. Ainapure, *Software Testing and Quality Assurance*. Technical Publications, 2013.
- [11] H. B. I. Alfaris, C. Anam, dan A. Masy’an, “Implementasi Black Box Testing Pada Sistem Informasi Pendaftaran Santri Berbasis Web Dengan Menggunakan PHP Dan MYSQL,” *J. Sains dan Teknol.*, vol. 6, no. 1, hal. 23–38, 2016, Diakses: Nov 11, 2020. [Daring]. Tersedia pada: <http://ejournal.unwaha.ac.id/index.php/saintek/article/view/64/64>.