

# Penggunaan *Black-Box Testing* dalam *Integration Testing* Sistem Pembayaran PaDi UMKM

Michelle Angeline Aprila<sup>#1</sup>, Radiant Victor Imbar<sup>\*2</sup>

<sup>#</sup>Program Studi S1 Sistem Informasi, Fakultas Teknologi dan Rekayasa Cerdas, Universitas Kristen Maranatha  
Jl. Prof. drg. Surya Sumantri No. 65, Bandung 40164, Indonesia

<sup>1</sup>michelleangelineaprila@gmail.com

<sup>2</sup>radiant.vi@it.maranatha.edu

**Abstract** — This study aims to analyze the application of black box testing in integration testing for the payment method between PaDi UMKM and payment service providers. The research method used is qualitative, with a case study approach focused on the experiences during an internship as a Quality Assurance (QA) in the payment team. The findings of the study indicate that the implementation of integration testing using black box testing for the payment method between PaDi UMKM and payment service providers allows the QA team to conduct more detailed and accurate testing, based on the sprint retrospective results. The use of black box testing facilitates the testing process, enabling QA to perform testing more quickly and efficiently. Additionally, the application of black box testing provides added value in the form of user experience, offering clearer insights into user interactions with the features or application after its release. Overall, the application of this method plays a crucial role in improving the quality of testing and accelerating the evaluation and improvement process, ensuring that the released application has better quality and meets user expectations.

**Keywords**— Black Box Testing, Integration Testing, Quality Assurance (QA), User Experience.

## I. PENDAHULUAN

### A. Latar Belakang

Perkembangan teknologi informasi saat ini mengalami kemajuan yang sangat pesat, yang mempermudah berbagai aktivitas masyarakat, termasuk dalam sektor perdagangan. Di era modern ini, banyak individu yang lebih memilih untuk berbelanja melalui platform e-Commerce [1]. Oleh karena itu, pemilik usaha perlu memiliki media untuk melakukan penjualan secara daring, salah satunya adalah melalui media berjualan daring seperti PaDi UMKM. PaDi UMKM, yang merupakan Pasar Digital UMKM, diluncurkan oleh Kementerian BUMN dan berada di bawah pengawasan PT Telekomunikasi Indonesia. Tujuan dari platform ini adalah untuk mendukung usaha mikro, kecil, dan menengah (UMKM) dalam pendistribusian dan pemasaran produk secara digital, sehingga dapat memperluas pasar UMKM dan mempercepat pertumbuhan ekonomi Indonesia [2].

Berbagai fitur serta pengembangan yang diterapkan pada PaDi UMKM perlu melalui tahap pengembangan atau peningkatan, sehingga diperlukan *Quality Assurance* (QA) sebelum fitur dan pengembangan tersebut dirilis ke publik. Tujuan dari QA adalah untuk mencegah kesalahan saat aplikasi digunakan oleh masyarakat serta untuk meningkatkan kualitas aplikasi. Pengujian perangkat lunak memiliki dua metode utama, yaitu *black-box testing* dan *white-box testing*. *Black-box testing* dilakukan untuk menguji fungsionalitas aplikasi, dengan tujuan memastikan aplikasi berfungsi sesuai dengan harapan. Sementara itu, *white-box testing* dilakukan dengan menguji aplikasi berdasarkan cara kerja internalnya, yang bertujuan untuk menilai tingkat keakuratan metode yang diterapkan dalam pengembangan aplikasi tersebut [3]. *Black-box testing* mencakup beberapa jenis pengujian, di antaranya adalah *integration testing*, *functional testing*, *system testing*, *acceptance testing*, *beta testing*, dan *regression testing*. *White-box testing* juga meliputi beberapa jenis pengujian, di antaranya *unit testing*, *integration testing*, dan *regression testing* [4].

Pengujian yang dilakukan pada PaDi UMKM menggunakan metode *black-box testing* dengan teknik pengujian manual, yang bertujuan untuk memaksimalkan pengalaman pengguna (*user experience*) serta memberikan fleksibilitas dalam menangani perubahan skenario pengujian guna memastikan proses pembayaran berjalan dengan lancar. Mengingat banyaknya metode pembayaran yang tersedia serta beragamnya penyedia layanan pembayaran, diperlukan pengujian integrasi (*integration testing*) antara PaDi UMKM dengan pihak penyedia layanan pembayaran.

### B. Rumusan Masalah

Berdasarkan latar belakang di atas, pertanyaan yang timbul dalam penelitian ini antara lain:

1. Bagaimana proses pengujian dengan menggunakan metode *black-box testing* menggunakan *integration testing* antara PaDi UMKM dan penyedia layanan pembayaran?
2. Bagaimana hasil pengujian dengan menggunakan metode *black-box testing* menggunakan *integration testing* antara PaDi UMKM dan penyedia layanan pembayaran?

### C. Tujuan Penelitian

Berdasarkan latar belakang di atas, tujuan pembahasan dari laporan ini yaitu:

1. Menjelaskan proses pengujian *black-box testing* dengan menggunakan *integration testing* antara PaDi UMKM dengan penyedia layanan pembayaran.
2. Menjelaskan hasil pengujian dengan menggunakan metode *black-box testing* menggunakan *integration testing* antara PaDi UMKM dan penyedia layanan pembayaran.

## II. DESKRIPSI PEKERJAAN DAN TANGGUNG JAWAB

Sebagai *Quality Assurance (QA)* di tim *Payment and Financing* PaDi UMKM, penulis bertanggung jawab untuk menyusun *test case* sesuai dengan proyek yang sedang berjalan, melakukan pengujian pada aplikasi *staging* dan produksi, serta berkoordinasi dengan tim terkait. *Test case* disusun untuk mencegah kesalahan yang sama terjadi berulang kali. Pengujian dimulai pada *website staging*, dan apabila ditemukan *defect*, QA akan menghubungi tim terkait untuk melakukan perbaikan. Setelah perbaikan dilakukan, QA akan melaksanakan pengujian ulang. Apabila pengujian berhasil, tugas tersebut siap untuk dipindahkan ke *website production*.

## III. LANDASAN TEORI

### A. Metode Agile

Metode *Agile* merupakan suatu pendekatan pengembangan yang menekankan fokus pada individu, nilai bisnis, kemampuan beradaptasi terhadap perubahan, serta pembelajaran yang berkelanjutan. Metode ini bertujuan untuk meningkatkan kinerja proyek, mendukung tim yang mandiri, serta memperbaiki kualitas teknologi dan proses kolaborasi. *Agile* dirancang untuk mengatasi permasalahan yang sering ditemui dalam metode pengelolaan proyek tradisional, seperti tingginya kecepatan perubahan dan kompleksitas yang terus berkembang. Dengan melibatkan pengguna pada setiap tahap pengembangan, *Agile* menjadi lebih fleksibel dan efektif dalam memecahkan masalah [5] [6].

### B. SCRUM

*SCRUM* merupakan suatu kerangka kerja yang mendukung kolaborasi tim dalam mengelola tugas-tugas kompleks. *SCRUM* terdiri dari peran-peran seperti *Product Owner*, *Development Team*, dan *Scrum Master*. Tujuan dari *SCRUM* adalah untuk memaksimalkan nilai produk dan memastikan implementasi yang efektif serta efisien dalam pengelolaan proyek [7]. *SCRUM* mencakup beberapa jenis pertemuan, yaitu *Sprint*, *Sprint Planning*, *Daily Scrum*, *Sprint Review*, dan *Sprint Retrospective*. Setiap hari, tim melaksanakan *Daily Scrum* selama 15 menit untuk memberikan pembaruan mengenai progres yang telah dicapai dan mengidentifikasi serta mengatasi masalah yang muncul. Setelah *Sprint* selesai, diadakan *Sprint Review* untuk mengevaluasi hasil kerja yang telah dilakukan, serta *Sprint Retrospective* untuk melakukan introspeksi terhadap kinerja tim guna meningkatkan efektivitas kerja di masa mendatang [5].

### C. Software Testing Life Testing

*STLC (Software Testing Life Cycle)* terdiri dari enam tahap utama [8]:

1. Analisis Kebutuhan: Melakukan analisis terhadap jenis dan fokus pengujian yang akan dilaksanakan.
2. Perencanaan Pengujian: Menyusun rencana pengujian yang mencakup strategi pengujian dan memperkirakan biaya yang diperlukan.
3. Pengembangan *Test Case*: Membuat dan meninjau *test case* serta data pengujian yang relevan.
4. Persiapan Lingkungan: Menyiapkan lingkungan perangkat lunak dan perangkat keras yang diperlukan untuk mendukung pelaksanaan pengujian.
5. Eksekusi Pengujian: Melakukan pengujian sesuai dengan rencana yang telah disusun, mencatat hasil pengujian, dan melaporkan *bug* yang ditemukan selama pengujian.
6. Penutupan Siklus Pengujian: Melakukan pelaporan hasil pengujian dan evaluasi terhadap seluruh proses pengujian yang telah dilaksanakan.

Tahapan-tahapan tersebut memastikan bahwa pengujian dilakukan secara sistematis, terstruktur, dan terorganisir dengan baik.

#### D. Software Testing

Pengujian perangkat lunak (*software testing*) adalah proses untuk memeriksa sistem perangkat lunak guna mengidentifikasi adanya *bug* atau kesalahan sebelum digunakan secara luas. Kesalahan dapat terjadi akibat berbagai faktor, termasuk kesalahan dalam pengkodean atau ketidakjelasan informasi. Tujuan utama dari pengujian perangkat lunak adalah untuk mengidentifikasi dan memperbaiki *bug* yang ditemukan, sehingga sistem dapat berfungsi dengan baik dan sesuai harapan [9].

Kasus uji (*test case*) adalah dokumen yang merinci prosedur pengujian yang digunakan untuk mengevaluasi fungsionalitas suatu sistem. Setiap kasus uji mencakup ID, deskripsi, jenis skenario (positif atau negatif), langkah-langkah pengujian, hasil yang diharapkan, status (lulus/gagal), serta serangkaian panduan yang bertujuan untuk membantu pengembang dalam meningkatkan proses pengujian. Kasus uji juga mencakup komentar-komentar yang bermanfaat. Tujuan dari semua jenis pengujian ini adalah untuk memastikan bahwa perangkat lunak berfungsi sebagaimana mestinya dan bebas dari cacat yang dapat memengaruhi kinerjanya.[10].

#### E. Black-Box Testing

*Black-Box Testing* adalah metode pengujian yang menilai fungsionalitas dan kebutuhan suatu sistem tanpa memperhatikan kode program yang mendasarinya. Pengujian ini berfokus pada *input* yang diberikan kepada sistem dan *output* yang dihasilkan, serta apakah sistem berfungsi sesuai dengan kebutuhan pengguna, tanpa melibatkan pemahaman tentang struktur kode. Keuntungan utama dari *black-box testing* adalah pengujian tidak perlu mengetahui bahasa pemrograman atau implementasi teknis sistem, serta dapat membantu mengidentifikasi ketidakkonsistenan dalam spesifikasi persyaratan sistem [4].

#### F. Integration Testing

Beizer mendeskripsikan integrasi sebagai proses penggabungan berbagai komponen untuk membentuk suatu kesatuan komponen yang lebih besar [11]. Tujuan dari pengujian integrasi adalah untuk mengidentifikasi ketidaksesuaian atau masalah yang timbul antara dua komponen yang telah digabungkan [12].

#### G. Manual Testing

*Manual testing* adalah jenis pengujian perangkat lunak yang dilakukan secara manual oleh pengujian tanpa menggunakan alat otomatisasi. Pengujian ini memiliki peran yang penting dalam siklus pengembangan perangkat lunak untuk memastikan bahwa perangkat lunak memenuhi standar kualitas yang diharapkan. Pada pengujian *end-to-end*, *manual testing* digunakan untuk menjalankan perangkat lunak sesuai dengan *test case* yang telah disusun dan membandingkan *output* yang dihasilkan dengan hasil yang diharapkan [13] [14].

#### H. Use Case Diagram

Penerapan *use case* terinspirasi oleh *user stories* untuk mendukung pengelolaan *backlog* dalam *Scrum* dan sejalan dengan aliran Kanban, dengan memperkenalkan konsep baru yang penting, yaitu potongan kasus penggunaan. *Use case* merupakan urutan langkah yang dilakukan oleh sistem untuk menghasilkan hasil yang dapat diamati oleh pengguna tertentu. Untuk memahami sebuah *use case*, pendekatannya dilakukan melalui narasi. Cerita-cerita ini menggambarkan bagaimana tujuan dicapai dan bagaimana menangani masalah yang muncul selama proses tersebut. Cerita-cerita tersebut membantu pengembang memahami *use case* dan cara mengimplementasikannya secara bertahap [15].

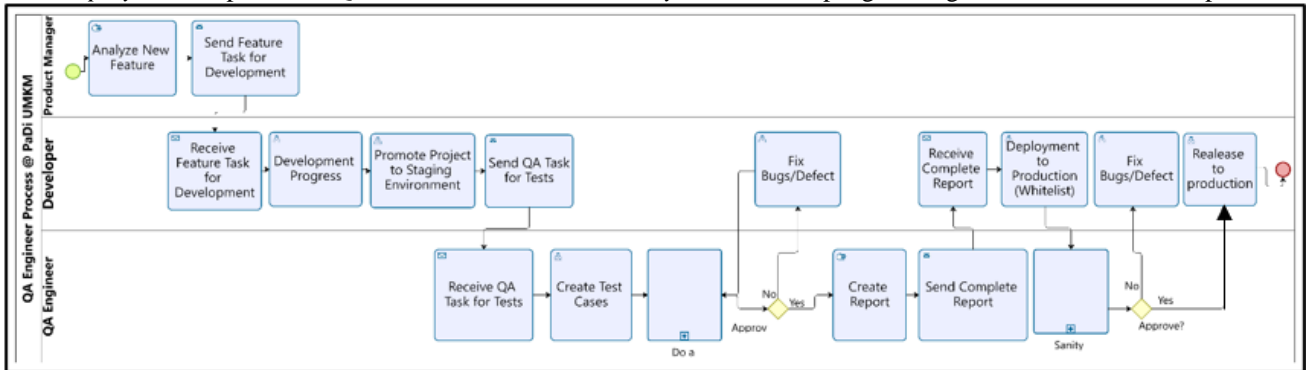
#### I. Flowchart

Analisis sistem menggunakan *flowchart* sebagai dokumentasi untuk menjelaskan diagram logis dari sistem yang akan dikembangkan kepada pengembang. Secara umum, *flowchart* terdiri dari simbol-simbol yang masing-masing mewakili suatu proses tertentu. Sementara itu, hubungan antar proses diwakili oleh jalur koneksi yang menghubungkan satu proses dengan proses lainnya [16].

IV. HASIL PEKERJAAN DAN PEMBAHASAN

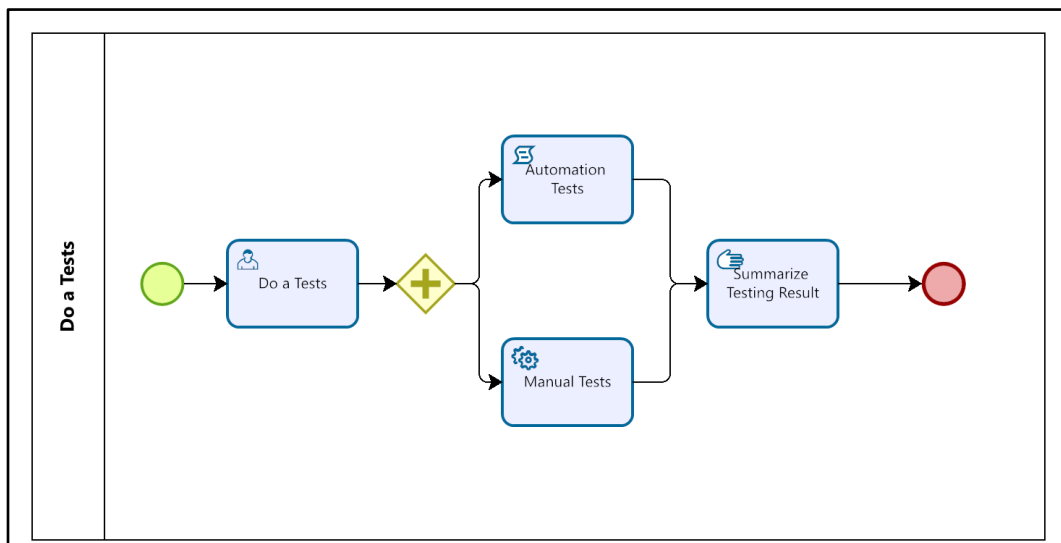
A. TAHAPAN IMPLEMENTASI

Gambar 1 menggambarkan alur kerja pada tim *Payment*. Tahap pertama dimulai dengan analisis fitur oleh *Project Manager* (PM) dan pembagian tugas kepada pengembang (developer). Setelah fitur selesai dikembangkan, *Quality Assurance* (QA) melakukan pengujian untuk memastikan fungsionalitasnya sesuai dengan harapan, dengan membuat *test case* dan menghubungi pengembang apabila ditemukan masalah. Jika semua fitur berfungsi dengan baik, QA menyusun laporan untuk proses *deployment* ke produksi. QA kemudian melakukan *sanity test* sebelum pengembang melakukan rilis untuk publik.



Gambar 1. Alur proses tim payment

Gambar 2 menggambarkan proses pengujian yang dilakukan oleh tim QA, yang mencakup pengujian otomatisasi dan manual. Pada tugas pembayaran yang dilakukan di PaDi UMKM, pengujian manual merupakan metode yang lebih dominan digunakan.



Gambar 2. Proses Testing

B. Testing Invoice dan Purchase Order Financing

PaDi UMKM menawarkan solusi pembiayaan melalui *PO Financing* dan *Invoice Financing* untuk mendukung kelancaran arus kas bisnis jangka pendek. *PO Financing* menggunakan *Purchase Order* (PO) sebagai jaminan untuk memperoleh pinjaman dari PNM, sementara *Invoice Financing* menggunakan *invoice* yang belum jatuh tempo sebagai jaminan kepada Pegadaian, PNM, Bank Mandiri, dan Bank Raya. Kedua jenis pembiayaan ini hanya dapat dilakukan oleh *buyer* B2B dengan pembayaran secara tempo, dan prosedur yang diterapkan pada keduanya hampir serupa.

Tabel I menunjukkan hasil pengujian pada fitur *invoice* dan *purchase order financing*. Berdasarkan pengujian tersebut, pengujian tidak menemukan kecacatan pada fungsionalitas fitur yang diuji.

TABEL I  
HASIL PENGUJIAN INVOICE DAN PURCHASE ORDER FINANCE

No	Fitur	Status	Catatan
1	<i>Seller</i> melakukan pinjaman <i>invoice financing</i>	<i>Pass</i>	
2	<i>Seller</i> melakukan <i>purchase order finance</i>	<i>Pass</i>	
3	<i>Seller</i> memilih PNM untuk melakukan pinjaman	<i>Pass</i>	
4	<i>Seller</i> memilih pegadaian untuk melakukan pinjaman	<i>Pass</i>	

### C. Testing Migration Library Lodash to non-Lodash

Tim Payment melakukan migrasi dari *lodash* ke *non-lodash*. Sebagai bagian dari proses QA, dilakukan pengujian untuk memastikan bahwa fitur pembayaran pada *website* tetap berfungsi dengan aman dan sesuai harapan setelah proses migrasi tersebut.

Tabel II menunjukkan hasil pengujian migrasi *library* dari *lodash* ke *non-lodash*. Selama pengujian, ditemukan *defect* pada tampilan transaksi *dashboard* dan tampilan *seller* pada bagian pinjaman tersedia. Kedua tampilan tersebut masih kosong (*blank*) dan tidak dapat diakses.

TABEL II  
HASIL PENGUJIAN MIGRATION LIBRARY LODASH TO NON-LODASH

No	Fitur	Status	Catatan
1	Invoice pada klpd	<i>Pass</i>	
2	Invoice pada buyer	<i>Pass</i>	
3	Instruksi pembayaran	<i>Pass</i>	
4	Tampilan jenis pembayaran	<i>Pass</i>	
5	Home top bar	<i>Pass</i>	
6	Transaksi dashboard	<i>Failed</i>	Masih blank
7	Seller pada pinjaman tersedia	<i>Failed</i>	Masih blank

### D. Testing E-Faktur

Sertifikat digital merupakan identitas wajib pajak yang dikeluarkan oleh Direktorat Jenderal Pajak dan hanya diberikan kepada *seller* yang terdaftar sebagai pengusaha kena pajak. Sertifikat ini diperlukan untuk pembuatan dan/atau pengajuan e-Faktur. Dengan adanya e-Faktur, *seller* tidak perlu mengunggah faktur pajak saat pembayaran dengan tempo pada status penagihan. Penggunaan e-Faktur juga mengeliminasi kebutuhan bagi *seller* untuk membuat faktur secara manual.

Tabel III menunjukkan hasil pengujian e-Faktur. Namun, pada fitur B2B non-WAPU dengan metode pembayaran langsung untuk produk PPN dan non-PPN, e-Faktur tidak terbuat secara otomatis. Seharusnya, dalam kasus tersebut, e-Faktur dapat dibuat secara otomatis.

TABEL III  
HASIL PENGUJIAN E-FAKTUR

No	Fitur	Status	Catatan
1	B2b Wapu pembayaran langsung produk PPN	<i>Pass</i>	
2	B2b Wapu pembayaran langsung produk PPN dan non PPN	<i>Pass</i>	
3	B2b Wapu pembayaran tempo produk PPN	<i>Pass</i>	
4	B2b Wapu pembayaran tempo produk PPN dan non PPN	<i>Pass</i>	
5	B2B non Wapu pembayaran langsung produk PPN	<i>Pass</i>	
6	B2B non Wapu pembayaran langsung produk PPN dan non PPN	<i>Failed</i>	E-faktur tidak terbuat otomatis
7	B2B non Wapu pembayaran tempo produk PPN	<i>Pass</i>	

8	B2B non Wapu pembayaran tempo produk PPN dan non PPN	Pass	
---	--	------	--

#### E. Testing Notifikasi

Notifikasi pada PaDi UMKM berfungsi sebagai pusat informasi penting, yang mencakup pembaruan pesanan, konfirmasi pembayaran, pengingat pengiriman, serta informasi mengenai promosi atau fitur baru dari platform PaDi UMKM. Notifikasi ini tersedia melalui web, aplikasi *mobile*, dan email, serta mencakup berbagai status, seperti menunggu pembayaran, kedaluwarsa, pembayaran diterima, dan penagihan.

TABEL IV dan TABEL V menunjukkan hasil pengujian untuk notifikasi dengan metode pembayaran langsung dan tempo. Untuk metode pembayaran langsung, semua fitur telah berhasil lolos pengujian. Namun, untuk metode pembayaran tempo, pada kasus H-7 sebelum *due date* dan H+8 setelah jatuh tempo, notifikasi tidak muncul sesuai yang diharapkan.

TABEL IV  
HASIL PENGUJIAN NOTIFIKASI UNTUK METODE LANGSUNG

No	Fitur	Status	Catatan
1	Ketika <i>Buyer</i> telah meng-klik tombol Bayar untuk menyelesaikan transaksi	Pass	
2	Ketika <i>Buyer</i> tidak menyelesaikan pembayaran dalam kurun waktu 1 minggu	Pass	
3	Ketika <i>Buyer</i> telah menyelesaikan pembayaran	Pass	

TABEL V  
HASIL PENGUJIAN NOTIFIKASI UNTUK METODE TEMPO

No	Fitur	Status	Catatan
1	Ketika Penagihan oleh <i>Seller</i> telah ter-submit secara otomatis ( <i>Seller Non PKP</i> ), Namun masih belum dilakukan <i>Approval</i> oleh <i>Buyer</i>	Pass	
2	Ketika Penagihan oleh <i>Seller</i> telah ter-submit, Namun masih belum dilakukan <i>Approval</i> oleh <i>Buyer</i>	Pass	
3	Ketika Perubahan Perpajakan telah di-submit ulang oleh <i>Seller</i>	Pass	
4	Ketika <i>Upload</i> Faktur Pajak telah di-submit ulang oleh <i>Seller</i>	Pass	
5	Ketika Tagihan sudah disetujui oleh <i>Buyer</i> , namun pembayaran belum dilakukan hingga H-7 ( <i>due date</i> pembayaran)	Failed	Tidak muncul notifikasinya
6	Ketika Tagihan sudah disetujui oleh <i>Buyer</i> , namun pembayaran belum dilakukan hingga H-3 ( <i>due date</i> pembayaran)	Pass	
7	Ketika Tagihan sudah disetujui oleh <i>Buyer</i> , namun pembayaran belum dilakukan hingga tanggal jatuh tempo	Pass	
8	Ketika Tagihan sudah disetujui oleh <i>Buyer</i> , namun pembayaran belum dilakukan hingga tanggal jatuh tempo (Reminder H+3 Due Date)	Pass	
9	Ketika Tagihan sudah disetujui oleh <i>Buyer</i> , namun pembayaran belum dilakukan hingga tanggal jatuh tempo (Reminder H+8 Due Date)	Failed	Tidak muncul notifikasi email

F. Testing Flow Pembayaran Bank BTN

Sebelumnya, Bank BTN menggunakan Xendit sebagai penyedia layanan *payment gateway*. Namun, saat ini dilakukan migrasi ke Finpay, yang merupakan salah satu produk *payment gateway* yang dikelola oleh PT Finnet Indonesia. Sehubungan dengan hal tersebut, dilakukan pengujian terhadap penggunaan Finpay untuk metode pembayaran pada PaDi UMKM melalui simulasi pembayaran.

TABEL VI menunjukkan hasil pengujian alur pembayaran Bank BTN setelah migrasi. Pada kasus *buyer* B2B dengan pembayaran tempo menggunakan *Virtual Account (VA)* Bank BTN, status pada halaman pembayaran masih belum berubah meskipun pembayaran telah dilakukan.

TABEL VI  
HASIL PENGUJIAN FLOW PEMBAYARAN BANK BTN

No	Fitur	Status	Catatan
1	Buyer B2B langsung gagal bayar, VA BTN expired lebih dari 7x24 jam	Pass	
2	Buyer B2B bayar langsung menggunakan VA BTN di PaDi UMKM	Pass	
3	Buyer KLPD bayar langsung menggunakan VA BTN di PaDi UMKM	Pass	
4	Buyer B2B bayar tempo menggunakan VA BTN expired tapi lebih dari 1 thn	Pass	
5	Buyer B2B bayar tempo menggunakan VA BTN di PaDi UMKM	Failed	Masih belum berubah statusnya di tampilan transaksi, tetapi sudah dibayar.
6	Buyer B2B bayar termin menggunakan VA BTN di PaDi UMKM	Pass	
7	Buyer KLPD bayar tempo menggunakan VA BTN di PaDi UMKM	Pass	

V. SIMPULAN DAN SARAN

A. Simpulan

Berdasarkan hasil penelitian, dapat disimpulkan bahwa pengujian menggunakan *black-box testing* dan *integration testing* dimulai dengan mempelajari *use case* yang ada, diikuti dengan penyusunan *test case* untuk memastikan semua skenario diuji dengan baik. Tim QA melakukan pengujian di *website staging* dan melaporkan *bug* kepada tim developer jika ditemukan. Setelah pengujian selesai, *release note* disusun untuk memberitahukan tim *payment* dan PM bahwa *task* siap untuk *dideploy*. Sebelum fitur digunakan secara langsung, dilakukan *sanity test* di lingkungan produksi untuk memastikan fitur berfungsi sesuai dengan harapan. Hasil pengujian disajikan dalam tabel, dan meskipun terdapat beberapa pengujian yang gagal, masalah tersebut telah diperbaiki oleh tim developer. Sebagian kegagalan pengujian disebabkan oleh faktor eksternal dan internal.

B. Saran

Tim *Payment*, khususnya *Project Manager*, perlu memastikan agar *backlog* tetap sesuai dengan target yang telah ditetapkan, agar tidak terus bergeser ke sprint berikutnya. Tim developer harus mempertimbangkan jumlah anggotanya agar setiap anggota dapat fokus pada tugas utama yang diberikan. Tim *Quality Assurance* QA sebaiknya memanfaatkan otomatisasi (*automation*) lebih sering untuk mempercepat proses pengujian serta mempelajarinya agar dapat bersaing dengan perusahaan lain. Meskipun pengujian manual tetap memiliki peran penting untuk memastikan tampilan sesuai dengan desain, penggunaan otomatisasi akan meningkatkan efisiensi dalam proses pengujian.

DAFTAR PUSTAKA

- [1] G. Nurcahyadi, "Survei : Sebagian Besar Masyarakat Indonesia Pilih Belanja Lewat E-commerce," Media Indonesia, 29 Agustus 2023. [Online]. Available: <https://mediaindonesia.com/ekonomi/608896/survei-sebagian-besar-masyarakat-indonesia-pilih-belanja-lewat-e-commerce>. [Accessed 12 10 2024].
- [2] "PaDi UMKM," 2022. [Online]. Available: <https://info.padiumkm.id/about-us>. [Accessed 18 Oktober 2024].
- [3] V. Febrian, M. R. Ramadhan, M. Faisal and A. Saifudin, "Pengujian pada Aplikasi Penggajian Pegawai dengan menggunakan Metode Blackbox," Jurnal Informatika Universitas Pamulang, vol. 5, pp. 61-66, 2020.

- [4] S. Nidhra and J. Dondeti, "BLACK BOX AND WHITE BOX TESTING TECHNIQUES –A LITERATURE REVIEW," *International Journal of Embedded Systems and Applications (IJESA)*, vol. 2, 2012.
- [5] S. Yordanova and K. Toshkov, "An Agile Methodology For Managing Business Processes In An It Company," 2019. [Online]. Available: <https://ideas.repec.org/a/dat/bmngmt/y2019i3p72-90.html>. [Accessed 24 October 2024].
- [6] K. C. Peitl and C. M. de Oliveira Baptista, "Agile Methodology: Benefits and Barriers on Its Initial Application," in *26th SAE BRASIL International Congress and Display*, 2017.
- [7] F. Sohail, S. S. Zia, R. Qureshi, M. Naseem and H. Haider, "Impact of Agile Methodology on Software Development Life Cycle," *Pakistan Journal of Engineering and Technology, PakJET*, vol. 4, no. 2, pp. 153-158, 2021.
- [8] T. Hamilton, "STLC (Software Testing Life Cycle)," Guru99, 2024.
- [9] J. Bach, "Explaining Testing to Anybody," [Online]. Available: [www.satisfice.com](http://www.satisfice.com). [Accessed 31 October 2024].
- [10] N. CHAUHAN, *Software Testing : Principles and Practices*, Oxford university press, 2010.
- [11] B. Beizer, *Software Testing Techniques*, New York: van nostrand reinhold, 1990.
- [12] S. Ali, L. C. Briand, M. J.-u. Rehman and H. Ashgar, "A state-based approach to integration testing based on UML models," *Information and Software Technology*, vol. 49, 2007.
- [13] K. . S. Thant and H. H. K. Tin, "THE IMPACT OF MANUAL AND AUTOMATIC TESTING ON SOFTWARE TESTING EFFICIENCY AND EFFECTIVENESS," *Indian journal of science and research*, pp. 88-93, 2023.
- [14] D. S. Prasetyo and W. Silfianti, "ANALISIS PERBANDINGAN PENGUJIAN MANUAL DAN AUTOMATION TESTING PADA WEBSITE E-COMMERCE," *Jurnal ilmiah teknik* , vol. 2, pp. 127-131, 2023.
- [15] I. Jacobson , I. Spence and B. Kerr, "The hub of software development. Use-case 2.0," *COMMUNICATIONS OF THE ACM*, vol. 59, 2016.
- [16] R. Rosaly, "Pengertian Flowchart Beserta Fungsi dan Simbol-simbol Flowchart yang Paling Umum Digunakan," *Jurnal libre*, 2019.