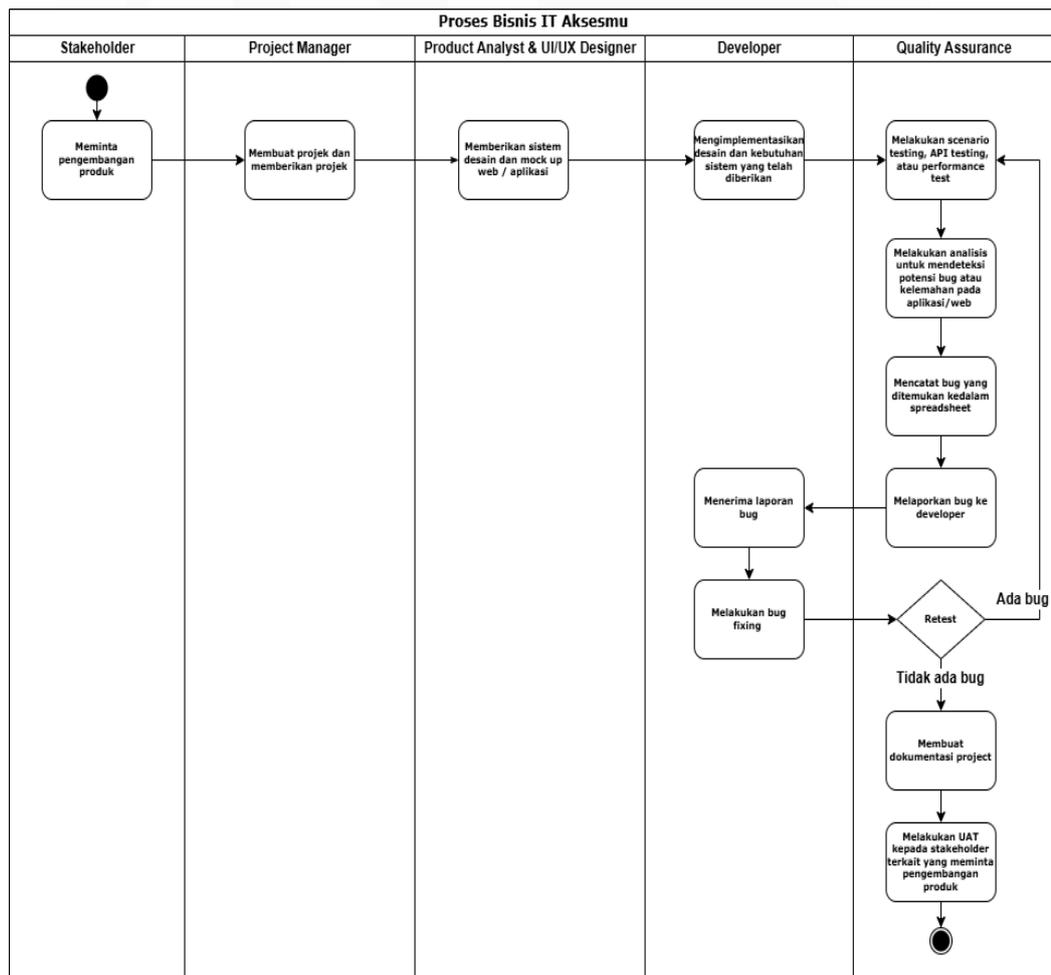


BAB III

PELAKSANAAN KERJA MAGANG

3.1 Kedudukan dan Koordinasi



Gambar 3. 1 Proses Bisnis IT Aksesmu

Gambar 3.1 menunjukkan proses bisnis divisi IT di Aksesmu yang menggambarkan alur kerja dan hubungan koordinatif antar tim dalam pelaksanaan suatu proyek pengembangan aplikasi atau website. Kedudukan masing-masing peran dalam tim IT memiliki fungsi yang saling berkaitan dan terkoordinasi untuk memastikan tercapainya hasil kerja yang sesuai dengan kebutuhan pengguna.

Proses dimulai dari *stakeholder*, baik internal maupun eksternal, yang

mengalami kendala atau memiliki masukan terhadap kinerja aplikasi atau website Aksesmu. *Stakeholder* menyampaikan keluhan atau kebutuhan tersebut kepada *Project Manager*, yang bertindak sebagai penghubung antara *stakeholder* dan tim teknis. *Project Manager* kemudian merangkum keluhan tersebut dan mengidentifikasi proyek-proyek pengembangan yang perlu dijalankan.

Selanjutnya, *Project Manager* melakukan koordinasi dengan *Product Analyst* dan *UI/UX Designer* untuk mendiskusikan solusi yang dapat diimplementasikan. *UI/UX Designer* bertugas menyusun rancangan antarmuka pengguna (user interface) yang lebih baik dan mudah digunakan, sementara *Product Analyst* membuat dokumen rencana pengembangan yang berisi gambaran umum sistem, kebutuhan fungsional, perancangan database, serta desain alur aplikasi.

Setelah dokumentasi dan rancangan disetujui, tugas berpindah ke *Developer*, yang akan mengimplementasikan rancangan ke dalam aplikasi staging. Tahap ini merupakan versi awal dari pengembangan, yang nantinya akan diuji oleh tim *Quality Assurance*. QA melakukan pengujian secara menyeluruh, baik melalui metode manual testing maupun *automation testing*, untuk memastikan bahwa sistem berjalan dengan baik dan sesuai standar.

Apabila ditemukan *bug* atau *error*, QA akan mencatat dan mendokumentasikannya untuk kemudian disampaikan kembali kepada *Developer*. *Developer* kemudian melakukan *bug fixing*, dan hasil perbaikan akan diuji kembali oleh QA hingga tidak ditemukan lagi kesalahan. Jika aplikasi sudah melewati semua tahapan pengujian dan dianggap stabil, maka QA akan melanjutkan proses ke tahap *User Acceptance Testing* (UAT).

UAT dilakukan bersama stakeholder dan tim IT untuk memastikan bahwa fitur atau perbaikan yang dikembangkan telah sesuai dengan ekspektasi dan kebutuhan pengguna akhir. Setelah UAT disetujui, fitur baru akan dirilis ke dalam aplikasi atau website versi produksi.

Dengan adanya koordinasi yang baik antara seluruh pihak yang terlibat

mulai dari *stakeholder*, *project manager*, *product analyst*, *UI/UX designer*, *developer*, hingga *quality assurance* maka proses pengembangan aplikasi dapat berjalan lebih efektif, efisien, dan menghasilkan produk yang berkualitas.

3.2 Tugas dan Uraian Kerja Magang

Pada langkah pengujian, tidak hanya produk yang harus diuji tetapi juga instruksi pengguna harus diuji. Penguji dapat membuat kasus uji baru atau memakai kasus uji yang sudah ada. Setelah pengujian, pengembang dan klien melakukan evaluasi bersama untuk memastikan bahwa semua kebutuhan telah dipenuhi. Dalam iterasi berikutnya, hasil evaluasi ini akan digunakan sebagai dasar untuk penyusunan dan pengembangan lebih lanjut [13]. Dalam pengembangan aplikasi digital untuk *quality assurance*, *jobdesk quality assurance* dapat mencakup desain sistem pelaporan terintegrasi, dashboard pemantauan penilaian, alur kerja validasi data, dan automasi proses audit [14]. Sistem digital ini mendukung efisiensi dan akuntabilitas dengan memastikan bahwa setiap data penilaian terdokumentasi, tervalidasi, dan dapat ditinjau kembali oleh auditor atau pemangku kepentingan secara real-time [15].

Kegiatan selama periode kerja magang sebagai *Quality Assurance (QA)* di Aksesmu menentukan dan melakukan pengujian aplikasi yang tepat agar hasil uji aplikasi yang maksimal, melakukan reporting dan tracking terhadap *bugs*, *help define*, *implement Quality Assurance (QA)* dengan cara mengembangkan aplikasi dengan tetap *on track* sehingga meminimalisir terjadinya *error*, berkolaborasi dengan tim *development* dan *product* untuk meningkatkan kualitas produk dan memastikan kebutuhan bisnis dapat terpenuhi, memastikan produk memenuhi standar kualitas yang ditetapkan termasuk *reliability*, *usability*, dan *performance* agar produk sesuai standar, mengembangkan dan membuat dokumentasi scenario test dengan baik, memberi bantuan atau arahan pada rekan tim ketika melaksanakan tanggung jawab dan menyelesaikan dengan tepat waktu dan benar, melaporkan kepada atasan ketika validasi proyek yang sudah selesai dan terdokumentasi dengan baik, berkoordinasi dengan tim

IT Infra untuk melakukan *performance test* menggunakan Jmeter dan Katalon terhadap aplikasi yang sedang di develop agar memiliki performa aplikasi yang sesuai dengan *requirement* yang ditetapkan, membuat skenario melalui spreadsheets, mencatat *buglist* melalui spreadsheets, menggunakan melakukan pengecekan data (Manual Testing) menggunakan DBeaver, melakukan *API Testing* menggunakan postman, melakukan cache data, menangkap data, memproses data *streaming* menggunakan redis insight, dan melakukan *User Acceptance Testing* (UAT) dengan stakeholder ketika test dan dokumentasi sudah selesai.



3.2.1 Tugas yang Dilakukan

Selama periode kegiatan magang di Aksesmu, penulis melakukan beberapa pekerjaan *quality assurance*, sebagai berikut:

Tabel 3.2. 1 Projek yang Dikerjakan

No.	Kegiatan	Tanggal Mulai	Tanggal Selesai
1.	Perkenalan lingkungan perusahaan		
1.a	Perkenalan terhadap lingkungan perusahaan di PT Sumber Trijaya Lestari (Aksesmu)	2/17/2025	2/17/2025
2.	<i>Transfer knowledge</i>		
2.a	Penalaran wawasan terkait perusahaan	2/17/2025	2/18/2025
2.b	Penalaran wawasan proses <i>testing</i> aplikasi	2/17/2025	2/18/2025
3.	<i>Training data untuk proses learning</i>		
3.a	Melakukan training <i>testing</i> pada <i>device registration</i>	2/18/2025	2/28/2025
3.b	Melakukan training <i>testing</i> pada AMS <i>Transactions</i>	2/26/2025	2/28/2025
4.	Validasi Input NIK KTP in Aksesmu		
4.a	Analisa sistem desain dan memahami struktur data	2/21/2025	3/21/2025
4.b	Membuat skenario	2/21/2025	3/21/2025
4.c	Testing aplikasi	2/21/2025	3/21/2025
4.d	Mencatat bug pada laporan <i>testing</i>	2/21/2025	3/21/2025
4.e	Melaporkan bug kepada <i>developer</i>	2/21/2025	3/21/2025
4.f	Membuat dokumentasi dan membuat flowchart	3/21/2025	3/21/2025
4.g	<i>Presentasi User Acceptance Testing (UAT)</i> dengan <i>user</i>	3/24/2025	3/24/2025
5.	API Tenor Tokomodan di Aksesmu		
5.a	Analisa sistem desain dan memahami	3/14/2025	3/20/2025

No.	Kegiatan	Tanggal Mulai	Tanggal Selesai
	struktur data		
5.b	Membuat skenario	3/14/2025	3/20/2025
5.c	API <i>Testing</i>	3/14/2025	3/20/2025
5.d	Mencatat <i>bug</i> pada laporan <i>testing</i>	3/14/2025	3/20/2025
5.e	Melaporkan <i>bug</i> kepada <i>developer</i>	3/14/2025	3/20/2025
5.f	Membuat dokumentasi	3/18/2025	3/20/2025
6.	<i>Enhance Perlindungan Data Pribadi di Aksesmu</i>		
6.a	Analisa sistem desain dan memahami struktur data	3/24/2025	3/26/2025
6.b	Membuat skenario	3/24/2025	3/26/2025
6.c	Testing aplikasi Aksesmu	3/24/2025	3/26/2025
6.d	Mencatat <i>bug</i> pada laporan <i>testing</i>	3/24/2025	3/26/2025
6.e	Melaporkan <i>bug</i> kepada <i>developer</i>	3/24/2025	3/26/2025
6.f	Membuat dokumentasi	3/24/2025	3/28/2025
7.	<i>Estimasi Pengiriman saat Checkout</i>		
7.a	Analisa sistem desain dan memahami struktur data	3/28/2025	3/28/2025
7.b	Membuat skenario	3/28/2025	3/28/2025
7.c	Testing aplikasi Aksesmu	3/28/2025	3/28/2025
7.d	Mencatat <i>bug</i> pada laporan <i>testing</i>	3/28/2025	3/28/2025
7.e	Melaporkan <i>bug</i> kepada <i>developer</i>	3/28/2025	3/28/2025
7.f	Membuat dokumentasi	3/28/2025	3/28/2025
8.	<i>Checkout H-3 dari GO SP</i>		
8.a	Analisa sistem desain dan memahami struktur data		
8.b	Membuat skenario	4/08/2025	4/16/2025
8.c	Testing aplikasi Aksesmu	4/08/2025	4/16/2025
8.d	Mencatat <i>bug</i> pada laporan <i>testing</i>	4/08/2025	4/16/2025

No.	Kegiatan	Tanggal Mulai	Tanggal Selesai
8.e	Melaporkan bug kepada <i>developer</i>	4/08/2025	4/16/2025
8.f	Membuat dokumentasi	4/08/2025	4/16/2025
8.g	Presentasi <i>User Acceptance Testing</i> (UAT) dengan <i>user</i>	4/16/2025	4/16/2025
9.	<i>Banner on AMS</i>		
9.a	Analisa sistem desain dan memahami struktur data	4/15/2025	4/16/2025
9.b	Testing aplikasi <i>Aksesmu Management System</i> (AMS) dan <i>website back office</i>	4/15/2025	4/16/2025
9.c	Mencatat bug pada laporan <i>testing</i>	4/15/2025	4/16/2025
9.d	Melaporkan bug kepada <i>developer</i>	4/15/2025	4/16/2025
9.e	Membuat dokumentasi	4/12/2025	4/16/2025
9.d	Presentasi <i>User Acceptance Testing</i> (UAT) dengan <i>user</i>	4/16/2025	4/16/2025
10.	<i>Automation Test menggunakan Katalon</i>		
10.a	Uji responsif UI/UX dari website dan aplikasi aksesmu berdasarkan variasi tipe handphone	4/16/2025	4/19/2025
11.	<i>Enhancement Sorting Number Payment Method</i>		
11.a	Analisa sistem desain dan memahami struktur data	4/15/2025	4/19/2025
11.b	Testing aplikasi Aksesmu	4/15/2025	4/19/2025
11.c	Melaporkan bug kepada <i>developer</i>	4/15/2025	4/19/2025
11.d	Membuat dokumentasi	4/19/2025	4/19/2025
12.	<i>Grading Member</i>		
12.a	Analisa sistem desain dan memahami struktur data	5/20/2025	6/19/2025
12.b	Testing aplikasi Aksesmu dan <i>Aksesmu Management System</i> (AMS)	5/20/2025	6/19/2025
12.c	Mencatat bug pada laporan <i>testing</i>	5/20/2025	6/19/2025
12.d	Melaporkan bug kepada <i>developer</i>	5/21/2025	6/19/2025

No.	Kegiatan	Tanggal Mulai	Tanggal Selesai
12.e	Membuat dokumentasi	5/30/2025	6/19/2025
12.f	Presentasi <i>User Acceptance Testing</i> (UAT) dengan <i>user</i>	6/19/2025	6/19/2025

3.2.2 Training data dari proses *learning*

Pengujian fungsional aplikasi Aksesmu dilakukan secara menyeluruh, dimulai dari pembukaan aplikasi, fitur *onboarding*, registrasi akun, aktivasi akun pengguna offline, hingga proses login dan akses ke halaman utama. Setiap elemen UI seperti tombol, form input, navigasi halaman, serta validasi data diuji untuk memastikan fungsinya berjalan dengan baik. Setelah login, pengguna dapat mengakses berbagai fitur utama seperti *search bar*, *scan barcode*, *swipe banner*, token pulsa, bayar tunda, catatan warung, edukasi, saldo virgo, aksesmu *star*, dan *voucher*. Selanjutnya, pengguna dapat memulai proses belanja dengan menambahkan barang ke keranjang, memilih metode pembayaran seperti COD, QRIS, atau transfer bank, yang datanya langsung tersinkronisasi ke *Aksesmu Management System* (AMS). AMS mendukung pengelolaan pesanan non-mandiri, mulai dari *picking* dan *packing* oleh tim gudang, pencetakan struk dan surat jalan, hingga pengiriman. Proses diakhiri dengan konfirmasi penerimaan melalui fitur *Proof of Delivery* (POD) yang mencatat nama penerima dan nominal pembayaran. Seluruh alur terintegrasi secara otomatis guna memastikan efisiensi dan keakuratan layanan.



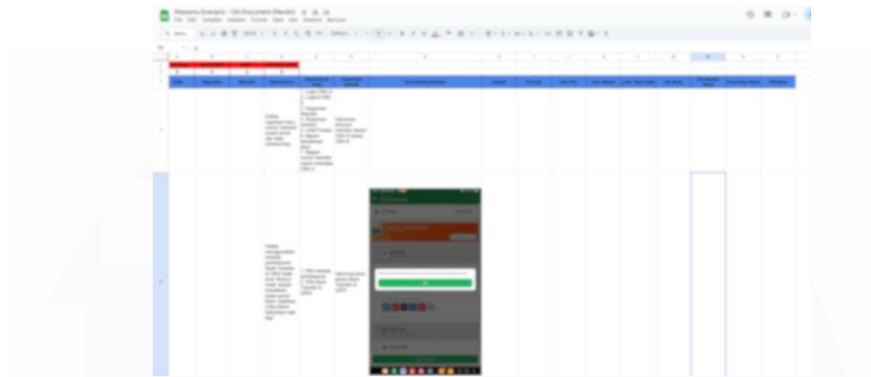
Gambar 3. 2 *Training Testing* Aplikasi

Gambar 3.2 menggambarkan proses pengujian aplikasi Aksesmu dalam konteks *training* mengenai *testing* yang dilakukan untuk memastikan setiap fitur berjalan sesuai fungsinya. Diagram atau ilustrasi dalam gambar ini menunjukkan alur dari fase awal hingga akhir, dimulai dari pembukaan aplikasi, registrasi, aktivasi akun, login, hingga akses ke fitur utama seperti pemindaian barcode, pengelolaan keranjang belanja, hingga metode pembayaran. Pengujian dilakukan tidak hanya pada sisi frontend (pengalaman pengguna), tetapi juga backend (sinkronisasi data ke AMS). Proses ini juga memperlihatkan integrasi antara aplikasi Aksesmu dengan *Aksesmu Management System (AMS)*, termasuk proses picking, packing, dan pencetakan struk di gudang hingga pengiriman dan konfirmasi *Proof of Delivery (POD)*. Tujuan utama dari testing ini adalah memastikan bahwa seluruh elemen sistem dari *User Interface (UI)* hingga sistem *backend* telah melewati validasi fungsional dan mampu mendukung operasional secara otomatis dan efisien.



Gambar 3.3 *Training* Membuat Skenario

Gambar 3.3 adalah skenario aplikasi *Aksesmu* yang dirancang dengan alur penggunaan yang jelas dan terstruktur mulai dari tahap awal pembukaan aplikasi hingga pengguna berhasil masuk ke halaman utama. Saat pertama kali dibuka, pengguna diberikan pilihan untuk login atau melakukan registrasi. Proses registrasi mencakup dua jalur, yaitu aktivasi bagi pengguna yang sudah terdaftar secara offline dan pendaftaran baru bagi pengguna yang belum memiliki akun. Setiap langkah dalam proses ini telah dilengkapi dengan elemen-elemen antarmuka seperti input data, tombol navigasi, dan konfirmasi persetujuan terhadap syarat dan ketentuan. Setelah berhasil mendaftar atau mengaktifkan akun, pengguna dapat langsung login dan mengakses halaman utama aplikasi. Di dalamnya, berbagai fitur utama seperti pencarian produk, promosi, pembayaran, edukasi, dan layanan lainnya dapat digunakan dengan mudah. Desain aplikasi menitikberatkan pada kemudahan navigasi dan aksesibilitas guna meningkatkan kenyamanan serta efisiensi bagi para penggunanya.



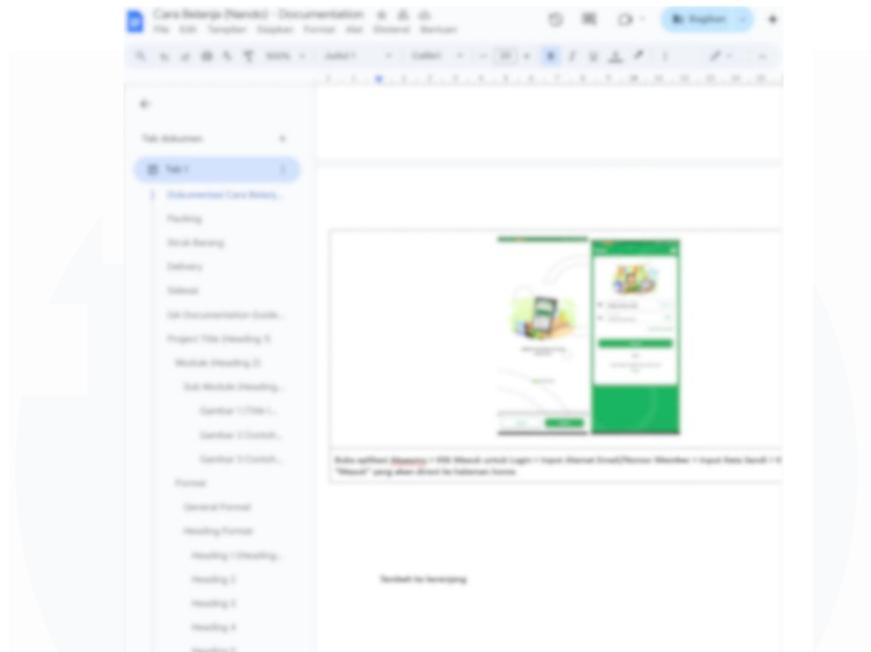
Gambar 3. 4 *Training* Mencatat *Buglist*

Gambar 3.4 merupakan proses pencatatan buglist dilakukan selama sesi pelatihan untuk memastikan setiap kendala atau kesalahan yang ditemukan dapat terdokumentasi dengan baik. Aktivitas ini bertujuan untuk mengidentifikasi masalah teknis maupun fungsional dalam aplikasi yang mungkin memengaruhi pengalaman pengguna. Setiap bug yang ditemukan dicatat secara sistematis, mencakup deskripsi masalah, langkah reproduksi, tingkat keparahan, serta status penanganannya. Pencatatan ini menjadi acuan penting bagi tim pengembang dalam melakukan perbaikan dan peningkatan aplikasi ke depannya. Dokumentasi buglist juga berfungsi sebagai bukti bahwa proses pelatihan tidak hanya fokus pada penggunaan aplikasi, tetapi juga pada evaluasi kualitas sistem secara menyeluruh.



Gambar 3. 5 *Training* Intervensi Data

Gambar 3.5 adalah intervensi data yang dilakukan dalam sesi pelatihan untuk memberikan pemahaman kepada pengguna terkait proses pengelolaan dan penyesuaian data yang diperlukan dalam sistem. Kegiatan ini mencakup langkah-langkah manipulasi data sesuai dengan kebutuhan operasional serta pengujian fungsi-fungsi yang berkaitan dengan pembaruan, koreksi, atau input ulang data. Tujuannya adalah untuk memastikan bahwa pengguna mampu melakukan intervensi secara tepat ketika terjadi ketidaksesuaian atau kesalahan data. Hasil dari proses ini menunjukkan bahwa pengguna mampu menjalankan prosedur intervensi dengan benar dan sistem dapat merespons perubahan data dengan baik tanpa menimbulkan gangguan pada proses lainnya, sehingga integritas dan konsistensi data tetap terjaga selama penggunaan aplikasi.



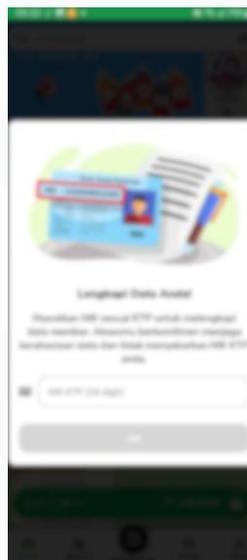
Gambar 3. 6 *Training* Dokumentasi Data

Gambar 3.6 adalah proses pembuatan dokumentasi data tahap terakhir dalam suatu proyek. Dokumentasi dibuat untuk memastikan aplikasi ketika kedepannya perusahaan membutuhkan kepastian dari tampilan aplikasi.

3.2.3 Validasi Input NIK KTP in Aksesmu

Validasi input *Nomor Induk Kependudukan (NIK) Kartu Tanda Penduduk (KTP)* di aplikasi Aksesmu, saat pengguna menuju halaman checkout, sistem akan memeriksa apakah member sudah memasukkan *Nomor Induk Kependudukan (NIK)*. Jika pengguna belum pernah mengisi *Nomor Induk Kependudukan (NIK)* maka akan muncul popup untuk memasukkan *Nomor Induk Kependudukan (NIK)*. Jika pengguna sudah mengisi tetapi masih menunggu verifikasi dan sudah terverifikasi, maka bisa langsung akses checkout. Jika *Nomor Induk Kependudukan (NIK)* tidak terverifikasi, maka akan muncul popup untuk input *Nomor Induk Kependudukan (NIK)* yang baru. Pada halaman akun terdapat

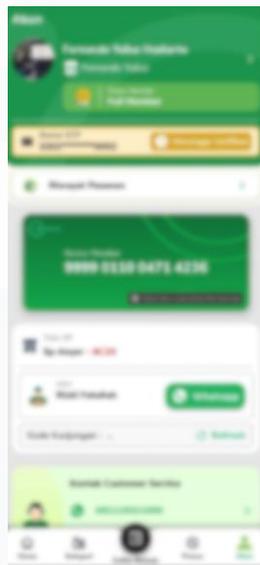
status belum input, menunggu verifikasi, *Kartu Tanda Penduduk* (KTP) tidak valid, dan terverifikasi. Jika pengguna belum input maka akan muncul popup untuk mengisi *Nomor Induk Kependudukan* (NIK) *Kartu Tanda Penduduk* (KTP). Jika sudah mengisi maka status menjadi menunggu verifikasi. Jika *Nomor Induk Kependudukan* (NIK) tidak valid maka status KTP tidak valid dan bisa edit menjadi *Nomor Induk Kependudukan* (NIK) *Kartu Tanda Penduduk* (KTP) yang baru. Jika sudah terverifikasi maka status menjadi terverifikasi dan tidak bisa edit *Nomor Induk Kependudukan* (NIK) *Kartu Tanda Penduduk* (KTP) lagi atau *view only*. Kegunaan dari validasi *Nomor Induk Kependudukan* (NIK) *Kartu Tanda Penduduk* (KTP) ini untuk meningkatkan keamanan dan kepercayaan pada sistem.



Gambar 3. 7 *Popup Input NIK KTP* (Home Page)

Gambar 3.7 adalah popup “Lengkapi Data Anda!” yang akan otomatis muncul saat pengguna menekan tombol *Checkout*, apabila sistem mendeteksi bahwa pengguna belum mengisi NIK KTP. Popup ini bersifat wajib dan menjadi langkah awal sebelum pengguna dapat melanjutkan ke proses pembelian. Hal ini sesuai dengan kebijakan bahwa identitas pembeli wajib dicantumkan dalam setiap transaksi. Dalam

popup tersebut, pengguna diminta untuk mengisi 16 digit NIK KTP dengan benar. Sistem akan melakukan pengecekan terhadap format dan kevalidan data yang diinput. Jika NIK yang dimasukkan tidak sesuai format (kurang dari 16 digit, angka berulang, atau angka berurutan), maka akan muncul pesan kesalahan dan pengguna perlu mengisi ulang. Jika NIK sudah valid, pengguna bisa melanjutkan ke proses *checkout*. Namun, apabila NIK yang dimasukkan ternyata tidak valid, maka akan muncul popup peringatan “Nomor KTP Anda Tidak Valid!”, dan pengguna hanya dapat mengganti dengan NIK baru melalui tombol Edit. Popup ini merupakan bagian dari fitur utama aplikasi Aksesmu, di mana proses *checkout* tidak dapat dilakukan sebelum NIK diisi dengan benar. Selain itu, status NIK akan ditampilkan pada halaman Akun agar pengguna bisa memantau dan memperbarui datanya sesuai ketentuan.



Gambar 3. 8 *Popup Input NIK KTP (Akun Page)*

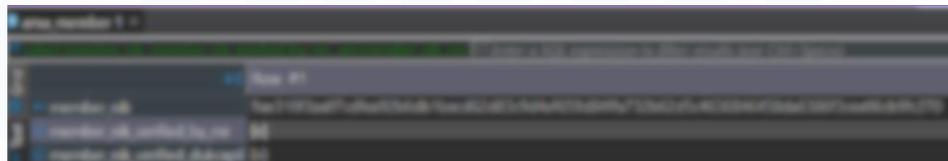
Gambar 3.8 adalah halaman akun yang terdapat status verifikasi NIK KTP seperti belum input, menunggu verifikasi, KTP tidak valid, dan KTP terverifikasi. Jika status belum input maka user diminta input NIK KTP dengan benar melalui popup NIK KTP. Jika status menunggu verifikasi maka user bisa edit NIK KTP, jika KTP tidak valid maka user

disuruh input NIK KTP yang baru dengan benar. Jika status sudah valid maka user tidak bisa edit NIK KTP dan hanya *view only*.



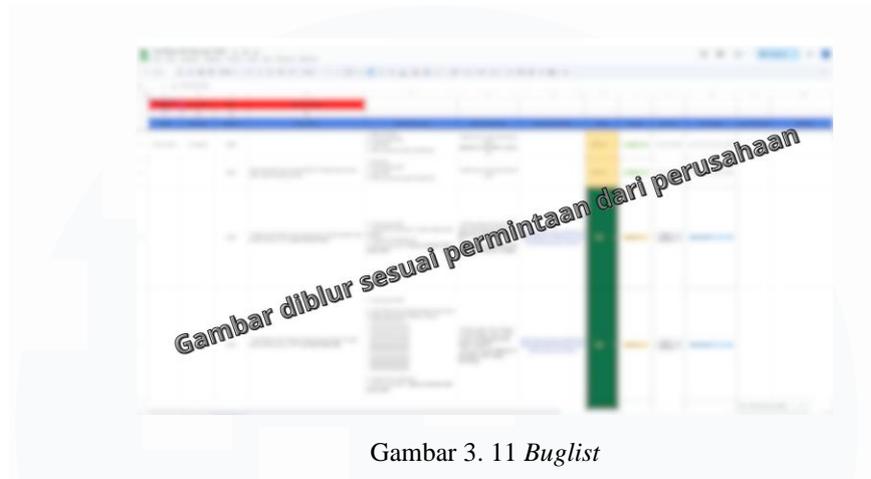
Gambar 3. 9 Skenario Validasi NIK KTP (Akun Page)

Gambar 3.9 adalah tahap testing yang dilakukan berdasarkan skenario yang dibuat dan skenario testing ini dilakukan pada halaman akun dan home serta memiliki beberapa tahap seperti ketika user belum input *Nomor Induk Kependudukan (NIK)*, menunggu verifikasi, *Kartu Tanda Penduduk (KTP)* tidak valid, dan terverifikasi.



Gambar 3. 10 Intervensi Database

Gambar 3.10 adalah tahap intervensi pada database dilakukan untuk melakukan testing pada skenario positif dan negatif yang dibuat. Hal ini dilakukan supaya menghindari terjadinya *bug* atau masalah pada elemen *user interface*.



Gambar 3. 11 Buglist

Gambar 3.11 merupakan hasil pengujian terhadap fitur input NIK pada aplikasi, ditemukan sejumlah *bug* terkait validasi dan tampilan informasi pada halaman akun dan checkout. Beberapa permasalahan utama meliputi ketidaksesuaian judul *popup*, tampilan teks status NIK, serta perilaku tombol edit yang tidak sesuai dengan desain fungsional yang diharapkan. Contohnya, tidak munculnya peringatan ketika NIK diisi dengan angka kurang dari 16 digit, angka yang sama berulang, atau angka berurutan. Selain itu, teks seperti *Empty*, *Checking*, *Invalid*, dan *Verified* yang muncul pada halaman akun belum sesuai dengan desain (*Figma*), yang seharusnya ditampilkan sebagai “Belum Input”, “Menunggu Verifikasi”, “KTP Tidak Valid”, dan “Terverifikasi”. Masalah lain yang ditemukan adalah ketika user melakukan edit NIK, tombol edit tetap muncul padahal seharusnya tidak ditampilkan jika status NIK masih dalam proses verifikasi atau sudah terverifikasi. Terakhir, ditemukan bahwa setelah input atau edit NIK, perubahan status tidak langsung muncul secara otomatis dan memerlukan refresh halaman, padahal seharusnya sistem dapat menampilkan status secara langsung. Seluruh *bug* tersebut telah diperbaiki sesuai dengan referensi dari *Figma* dan dokumen *System Design* (SD), dengan tingkat prioritas yang ditetapkan mulai dari *low* hingga *high* tergantung pada dampaknya terhadap pengalaman pengguna.



Gambar 3. 12 Dokumentasi

Gambar 3.12 merupakan dokumentasi validasi Nomor Induk Kependudukan (NIK) dibuat agar ke depannya ketika pengguna membutuhkan data yang berkaitan dengan proses verifikasi atau intervensi sistem, seluruh informasi dan langkah-langkah teknis yang telah dilakukan sebelumnya dapat dengan mudah diakses dan dijadikan referensi, baik untuk keperluan perbaikan, pengembangan fitur, ataupun dalam rangka memastikan konsistensi antara sistem yang berjalan dengan rancangan desain awal.



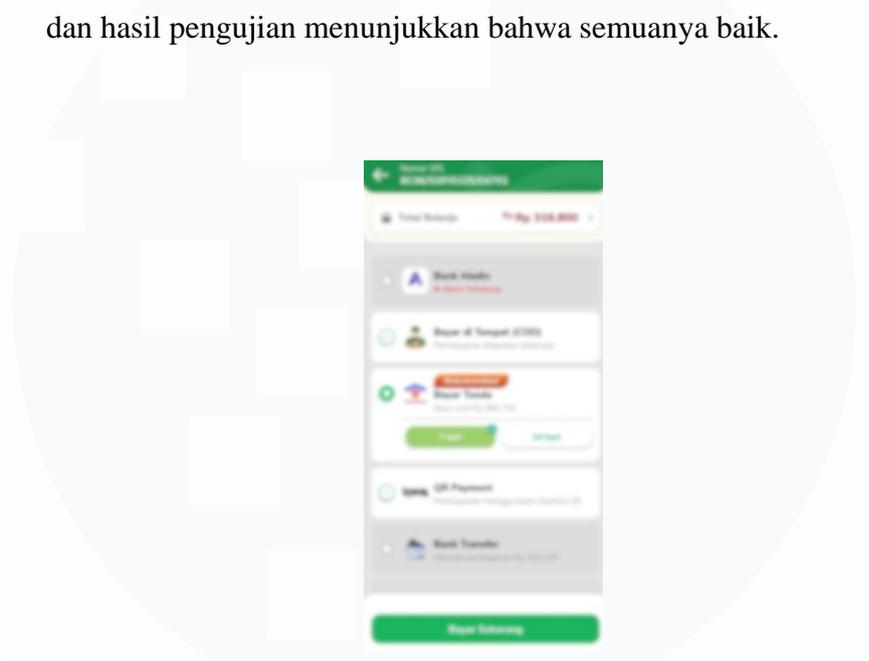
Gambar 3. 13 Membuat Flowchart

Gambar 3.13 adalah flowchart input *Nomor Induk Kependudukan* (NIK) dibuat agar kedepannya *user* bisa mengerti proses dari validasi *Nomor Induk Kependudukan* (NIK) pada halaman akun dan home. Flowchart ini juga dapat membantu *user* dalam mengikuti alur kerja sistem melalui flowchart yang informatif.

3.2.4 API Tenor Tokomodal in Aksesmu

Pengujian *Outlet Binaan Aksesmu* (OBA) untuk skenario dengan tenor 7 hari dan 7–14 hari menunjukkan bahwa alur pembayaran dengan metode Bayar Tunda berjalan sesuai ekspektasi. Pemilihan tenor berjalan normal, dengan tampilan checklist yang menunjukkan opsi yang dipilih, dan sistem menyesuaikan tenor berdasarkan batas yang tersedia. Pengguna juga dapat memilih metode pembayaran dengan tampilan yang sesuai, dan menerima notifikasi yang tepat saat nominal transaksi berada di luar batas yang ditentukan. Saat transaksi berhasil, proses pembayaran menampilkan *popup* konfirmasi dan validasi *One-Time Password* (OTP) dengan benar dan mengarahkan pengguna ke halaman konfirmasi.

Halaman *Dashboard* Bayar Tunda juga menampilkan semua informasi transaksi dan pinjaman, termasuk status pinjaman dan detail jatuh tempo sesuai tenor yang Anda pilih. Setiap skenario dijalankan dengan sukses, dan hasil pengujian menunjukkan bahwa semuanya baik.



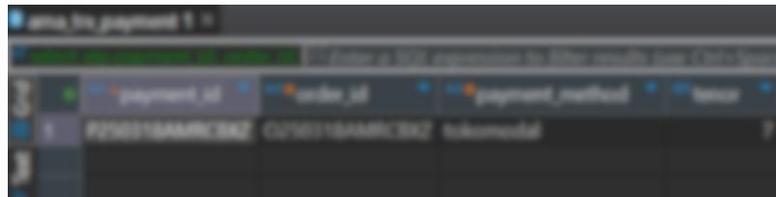
Gambar 3. 14 Tenor Bayar Tunda

Gambar 3.14 adalah tampilan testing tenor bayar tunda pada halaman metode pembayaran. Aksesmu adalah platform e-commerce yang menyediakan berbagai metode pembayaran guna memudahkan pengguna dalam bertransaksi. Salah satu metode yang ditawarkan adalah Pembayaran Bayar Tunda, yang memungkinkan pelanggan untuk berbelanja dengan membayar dalam jangka waktu tertentu. Fitur ini dirancang untuk meningkatkan fleksibilitas pembayaran bagi pelanggan yang membutuhkan tambahan waktu sebelum menyelesaikan transaksi. Dengan dukungan dari mitra keuangan seperti Tokomodal, pengguna dapat memilih tenor pembayaran tertentu sesuai dengan batas kredit yang diberikan. Keamanan transaksi juga menjadi prioritas utama dalam fitur ini, dengan penerapan *One Time Password* (OTP) menggunakan *POSTMAN* serta pembatasan limit transaksi agar pengguna dapat berbelanja secara bertanggung jawab.



Gambar 3. 15 Membuat Skenario

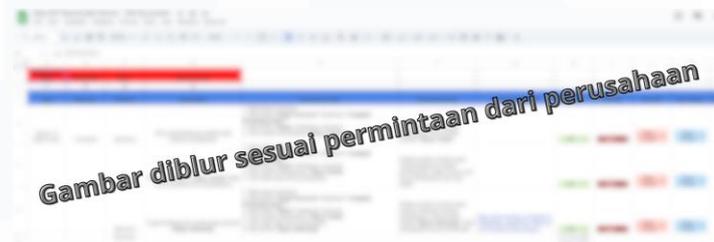
Gambar 3.15 adalah skenario API Tenor Tokomodal yang berhasil menampilkan tenor sesuai nominal dan batas transaksi. Jika nominal terlalu kecil, tenor tidak muncul. Jika melebihi batas per transaksi atau limit pengguna, sistem menolak dengan pesan yang sesuai. API juga menolak permintaan tidak valid. Secara keseluruhan, API berfungsi sesuai aturan bisnis. Skenario ini dibuat untuk memudahkan tugas dari *quality assurance* untuk memastikan aplikasi.



Gambar 3. 16 Intervensi Database

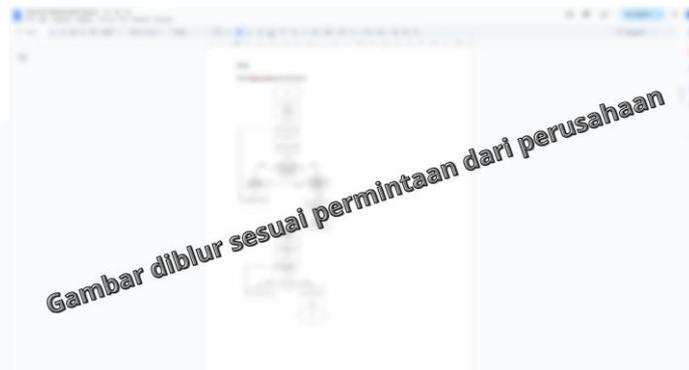
Gambar 3.16 merupakan proses pengujian fitur Bayar Tunda pada Aksesmu, dilakukan intervensi data secara langsung melalui database untuk mensimulasikan berbagai skenario positif dan negatif, seperti transaksi dengan tenor 7 hari, nominal melebihi limit, hingga status pinjaman yang sudah jatuh tempo. Intervensi ini bertujuan mempercepat proses uji validasi tanpa perlu menunggu kondisi real-time serta memastikan sistem mampu merespons skenario ekstrem sesuai aturan bisnis. Langkah ini juga membantu *quality assurance* dalam mengecek

konsistensi data yang ditampilkan di UI dengan data yang tersimpan di backend, serta mengidentifikasi bug yang mungkin tidak muncul dalam alur penggunaan normal.



Gambar 3. 17 Mencatat Buglist

Gambar 3.17 adalah hasil pengujian menunjukkan bahwa fitur metode pembayaran “Bayar Tunda” Tokomodal masih memiliki beberapa kendala. Awalnya, opsi Tokomodal tidak muncul saat memilih metode pembayaran, namun masalah tersebut telah diperbaiki. Setelah itu, ketika memilih metode “Bayar Tunda”, pilihan tenor pembayaran tidak muncul padahal seharusnya tersedia dan dapat dipilih. Terakhir, pada tahap akhir pembayaran, proses *booking* Tokomodal gagal saat menekan tombol “Bayar Sekarang”. Semua masalah ini telah dicatat dan memiliki prioritas tinggi untuk ditindaklanjuti oleh tim terkait.



Gambar 3. 18 Mencatat Dokumentasi

Gambar 3.18 merupakan dokumentasi ini memuat flowchart alur proses pemilihan tenor bayar tunda serta tampilan visual dari opsi tenor di aplikasi Aksesmu, yang dibuat untuk memudahkan pengguna dalam memahami langkah-langkah yang terjadi saat memilih tenor pembayaran. Flowchart disusun berdasarkan skenario aktual di aplikasi, dimulai dari pemilihan produk, masuk ke halaman pembayaran, memilih tenor, hingga konfirmasi transaksi dengan OTP. Setiap tahapan digambarkan secara sistematis untuk menggambarkan logika proses yang dijalankan sistem, termasuk pengecekan limit kredit, validasi nominal, serta penanganan error atau batasan transaksi. Dokumentasi ini penting sebagai panduan visual, terutama bagi pengguna atau tim internal, untuk memastikan bahwa tampilan dan alur yang muncul di aplikasi sesuai dengan desain dan aturan bisnis yang telah ditetapkan.

3.2.5 Enhance Perlindungan Data Pribadi in Aksesmu

Testing dari aplikasi Aksesmu yang menerapkan peningkatan *Perlindungan Data Pribadi* (PDP) dengan memastikan kepatuhan terhadap Syarat & Ketentuan serta Kebijakan Privasi, khususnya dalam pembatasan usia pengguna. Saat pendaftaran dan aktivasi akun, pengguna diwajibkan untuk mengisi data pribadi seperti nama toko, nama lengkap, nomor telepon, email, dan kata sandi, serta mencentang pernyataan bahwa mereka berusia 21 tahun ke atas. Jika persetujuan ini

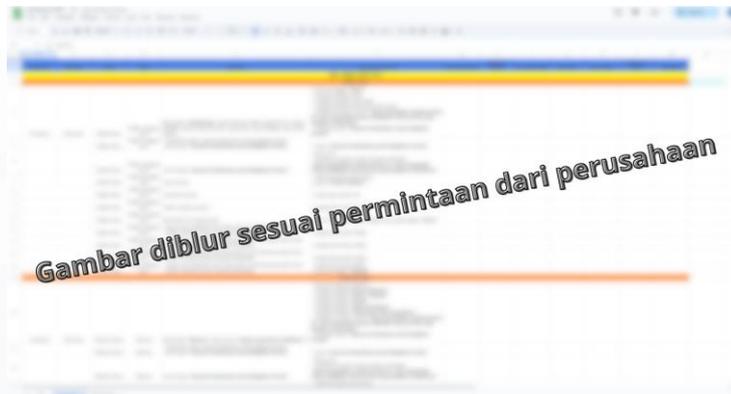
tidak diberikan, pengguna tidak dapat melanjutkan ke tahap berikutnya, termasuk mengakses layanan aplikasi.

Selain itu, jika pengguna belum menyetujui kebijakan sebelumnya, aplikasi akan menampilkan *popup* Syarat & Ketentuan serta Kebijakan Privasi pada halaman utama, yang hanya bisa ditutup setelah pengguna memberikan persetujuan. Dengan mekanisme ini, aplikasi memastikan bahwa hanya pengguna yang memenuhi batasan usia yang dapat mengakses layanan, terutama dalam pembelian produk dengan regulasi khusus seperti rokok dan barang lainnya yang memiliki batasan usia. Hal ini dilakukan untuk meningkatkan transparansi, perlindungan data, serta kepatuhan terhadap regulasi yang berlaku.



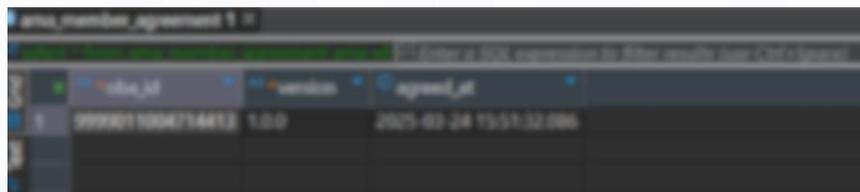
Gambar 3. 19 *Testing Enhance* Perlindungan Data Pribadi

Gambar 3.19 adalah tampilan dari pengujian *frontend* dan *backend* dilakukan untuk memastikan bahwa pengguna wajib menyetujui Syarat & Ketentuan serta Kebijakan Privasi sebelum melanjutkan ke layanan aplikasi. *Popup* muncul otomatis jika persetujuan belum diberikan. Sistem juga memblokir akses jika pengguna belum menyatakan berusia di atas 21 tahun.



Gambar 3. 20 Membuat Skenario

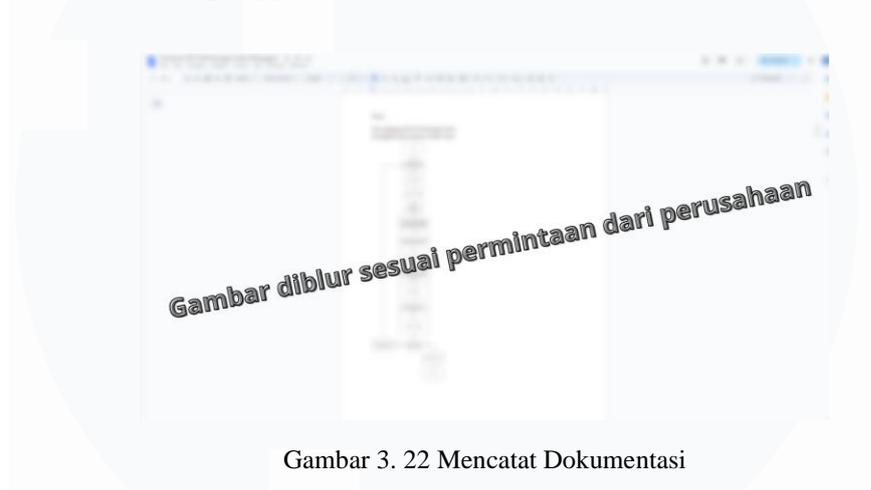
Gambar 3.20 adalah beberapa skenario pengujian disusun untuk memverifikasi alur perlindungan data pribadi secara menyeluruh. Skenario mencakup kondisi pengguna baru yang belum menyetujui kebijakan, pengguna yang mencoba mengakses layanan tanpa mencentang persetujuan, serta pengguna yang sudah menyetujui dan dapat lanjut ke layanan. Selain itu, diuji juga bagaimana sistem merespons ketika batas usia tidak terpenuhi, apakah popup muncul tepat waktu, serta apakah validasi benar-benar memblokir akses ke pembelian produk tertentu. Skenario ini membantu quality assurance dalam memastikan bahwa semua logika perlindungan data berjalan sesuai dengan regulasi dan kebijakan internal.



Gambar 3. 21 Intervensi Data

Gambar 3.21 adalah intervensi data dilakukan dengan cara memanipulasi nilai pada kolom persetujuan dan usia pengguna langsung dari database. Misalnya, status persetujuan diubah dari belum setuju menjadi setuju untuk melihat apakah sistem memperbarui tampilan sesuai kondisi terbaru. Pengujian juga dilakukan dengan menurunkan

nilai usia pengguna di bawah 21 tahun untuk melihat apakah sistem tetap menampilkan batasan akses. Intervensi ini sangat membantu dalam menguji skenario yang sulit dicapai secara alami dan mempercepat proses pengujian fitur-fitur penting tanpa harus menunggu interaksi manual dari pengguna.



Gambar 3. 22 Mencatat Dokumentasi

Gambar 3.22 adalah dokumentasi dibuat sebagai bukti implementasi dan referensi teknis yang menjelaskan bagaimana sistem Aksesmu menangani Perlindungan Data Pribadi, terutama dalam kaitannya dengan verifikasi usia dan persetujuan pengguna. Dokumentasi mencakup alur proses, daftar elemen yang divalidasi, hasil pengujian, serta rincian teknis terkait tampilan popup dan pembatasan akses. Hal ini bertujuan untuk memastikan pengembang, tim *quality assurance*, dan auditor memahami bahwa sistem telah memenuhi prinsip transparansi, legalitas, dan akuntabilitas dalam perlindungan data pribadi pengguna.

UNIVERSITAS
MULTIMEDIA
NUSANTARA



Gambar 3. 23 Membuat Flowchart

Gambar 3.23 adalah flowchart yang disusun untuk memberikan gambaran visual mengenai alur validasi Perlindungan Data Pribadi di Aksesmu. Diagram alur ini dimulai dari proses registrasi, di mana pengguna mengisi data pribadi, menyetujui syarat dan kebijakan, hingga proses pengecekan usia. Setelah itu, alur dilanjutkan dengan validasi sistem dan keputusan apakah pengguna dapat melanjutkan ke layanan utama atau tidak. Flowchart ini penting sebagai pedoman teknis dan dokumentasi visual agar seluruh tim memahami bagaimana proses berjalan secara sistematis, serta untuk memastikan sistem mematuhi peraturan yang berlaku terkait perlindungan data dan batasan usia.

3.2.6 Estimasi Pengiriman saat *Checkout*

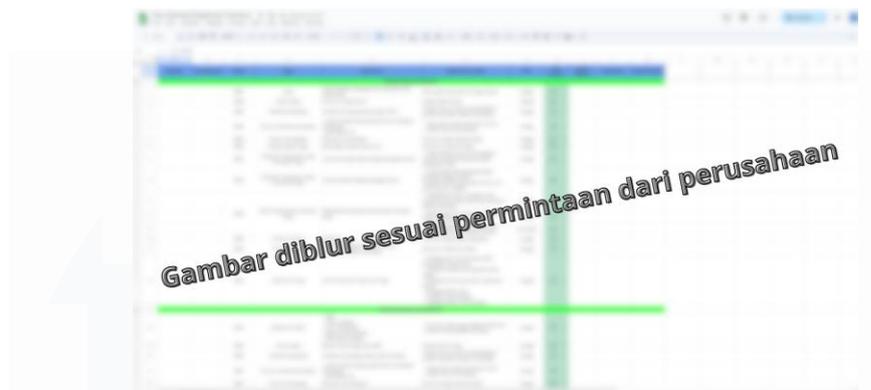
Testing skenario estimasi pengiriman saat checkout yang merupakan fitur yang menampilkan perkiraan waktu pengiriman produk di bagian atas halaman *checkout*. Estimasi ini mencakup pengiriman maksimal 2 hari setelah pembelian, dengan penyesuaian jika estimasi jatuh pada tanggal merah, yang akan memperpanjang pengiriman setelah

hari libur tersebut. Selain itu, jika stok produk tidak mencukupi, akan muncul pemberitahuan di bawah *flag* Tebus Murah dan pengguna tidak dapat melanjutkan ke proses pembayaran hingga stok tersedia. Fitur ini bertujuan memberikan transparansi dan pengalaman belanja yang lebih baik bagi pengguna.



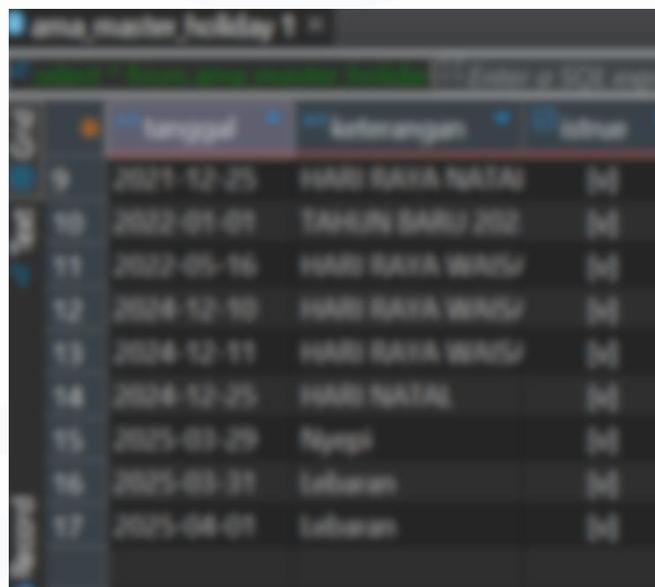
Gambar 3. 24 *Testing* Aplikasi

Gambar 3.24 adalah pengujian yang dilakukan untuk memastikan bahwa estimasi pengiriman ditampilkan dengan akurat di halaman *checkout*. Estimasi ini secara default menunjukkan maksimal 2 hari setelah tanggal pembelian, dan akan menyesuaikan secara otomatis jika estimasi pengiriman jatuh pada tanggal merah atau hari libur nasional. Sistem juga memblokir proses *checkout* jika stok produk tidak mencukupi, dengan menampilkan notifikasi tepat di bawah label Tebus Murah. Pengujian mencakup berbagai kombinasi tanggal, ketersediaan stok, dan hari libur untuk memastikan estimasi yang ditampilkan selalu relevan dan informatif bagi pengguna.



Gambar 3. 25 Membuat Skenario

Gambar 3.25 adalah skenario pengujian yang dibuat untuk mencakup kondisi normal (pengiriman dalam 2 hari), kondisi hari libur (estimasi bergeser), serta kondisi ketika stok tidak tersedia. Skenario juga menguji tampilan estimasi pada berbagai jenis produk dan lokasi pengiriman. Dengan skenario ini, *quality assurance* dapat mengonfirmasi bahwa sistem dapat secara dinamis menghitung dan menampilkan informasi pengiriman sesuai kondisi aktual.



Gambar 3. 26 Intervensi Data

Gambar 3.26 adalah intervensi dilakukan pada database untuk memanipulasi tanggal pembelian, data libur nasional, dan status stok

produk. Hal ini dilakukan untuk mempercepat pengujian terhadap berbagai skenario ekstrem, seperti pembelian di malam hari sebelum tanggal merah atau produk yang baru kembali tersedia. Intervensi ini membantu *quality assurance* dalam memvalidasi perhitungan estimasi yang tidak bisa diuji secara real-time.



Gambar 3. 27 Mencatat *Buglist*

Gambar 3.27 merupakan kumpulan *bug* pada saat pengujian, ditemukan beberapa *bug* seperti estimasi tanggal pengiriman yang tidak memperhitungkan hari libur, notifikasi stok kosong yang tidak muncul pada produk tertentu, serta kesalahan tampilan estimasi pada halaman *checkout* versi mobile. Selain itu, pada beberapa kasus estimasi pengiriman tetap muncul meski produk sudah tidak tersedia. Kumpulan *bug* ini telah dicatat, dianalisis, dan diperbaiki berdasarkan tingkat prioritasnya untuk memastikan pengalaman belanja pengguna tetap lancar



Gambar 3. 28 Membuat Dokumentasi

Gambar 3.28 merupakan dokumentasi disusun untuk menjelaskan alur kerja fitur estimasi pengiriman, termasuk cara sistem menghitung estimasi, bagaimana penyesuaian dilakukan terhadap hari libur, dan validasi ketersediaan stok. Dokumentasi ini juga mencakup tangkapan layar, referensi API terkait, serta hasil pengujian dari setiap skenario, yang berguna bagi tim pengembang dan *quality assurance* untuk pemeliharaan dan pengembangan fitur di masa depan.

3.2.7 Checkout H-3 dari *Grand Opening Stock Point*

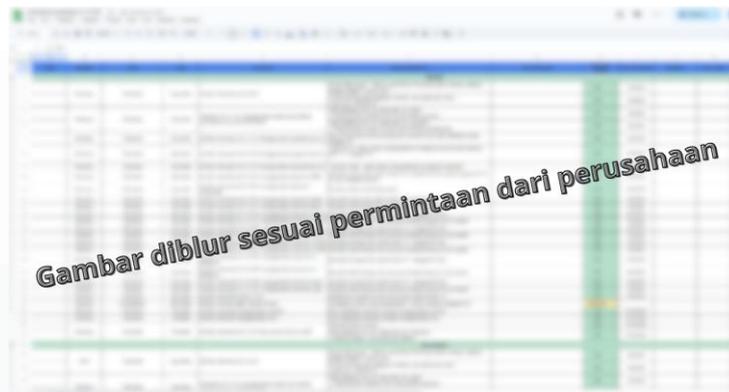
Checkout h-3 grand opening stock point diatas, sistem melakukan validasi berdasarkan tanggal *Grand Opening (GO)*. Jika transaksi dilakukan sebelum H-3 dari tanggal *grand opening*, akan muncul notifikasi *pop-up* yang menyatakan bahwa *Stock Point* belum dibuka dan menyarankan untuk menunggu hingga tanggal tertentu. Jika transaksi dilakukan antara H-3 dan H-1, sistem akan menampilkan kotak konfirmasi yang berisi informasi bahwa pesanan akan dikirim setelah *Stock Point* beroperasi pada tanggal *grand opening*. Jika member menyetujuinya maka bisa menggunakan semua metode pembayaran kecuali tokomodal, yang akan dinonaktifkan hingga tanggal *grand opening*, semua metode pembayaran tetap dapat digunakan. Dalam

transaksi yang dilakukan sebelum tanggal *grand opening*, sistem akan mengatur expired date pesanan menjadi satu hari setelah tanggal *grand opening*. Setelah tanggal *grand opening* tercapai, proses checkout akan dilakukan sesuai ketentuan yang berlaku.



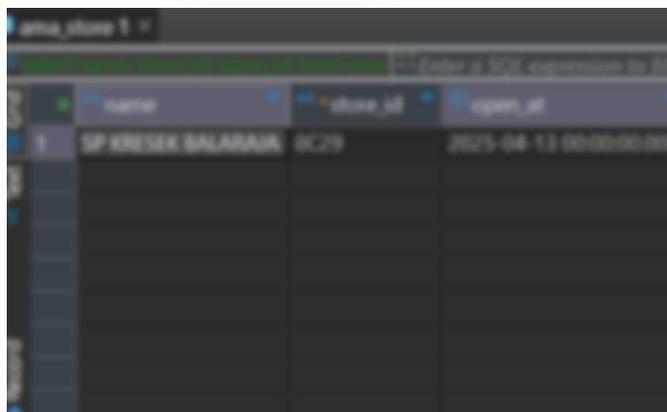
Gambar 3. 29 *Testing* Aplikasi

Gambar 3.29 merupakan pengujian dilakukan untuk memastikan sistem mampu melakukan validasi tanggal checkout terhadap tanggal *Grand Opening Stock Point*. Jika checkout dilakukan lebih awal dari H-3, sistem akan memblokir transaksi dengan popup pemberitahuan bahwa *Stock Point* belum dibuka. Saat transaksi dilakukan pada H-3 hingga H-1, sistem menampilkan konfirmasi yang menjelaskan bahwa pengiriman akan dilakukan setelah *Stock Point* resmi dibuka. Pengguna harus menyetujui informasi ini untuk melanjutkan. Semua metode pembayaran tersedia kecuali Tokomodal yang akan dinonaktifkan sampai tanggal GO. Setelah melewati tanggal GO, sistem menjalankan proses *checkout* secara normal sesuai ketentuan.



Gambar 3. 30 Mencatat Skenario

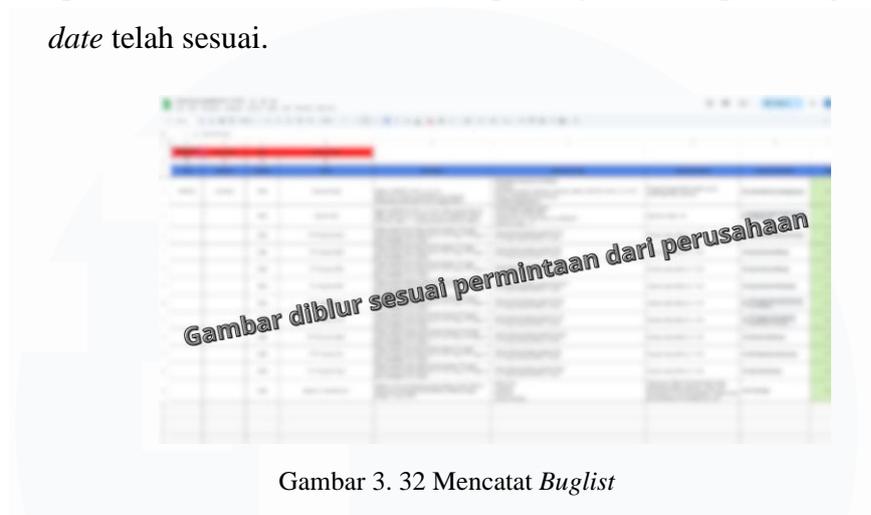
Gambar 3.30 merupakan skenario pengujian disusun berdasarkan kondisi waktu sebelum H-3, antara H-3 hingga H-1, dan setelah tanggal *Grand Opening*. Setiap skenario menguji tampilan popup, kondisi aktif/tidaknya metode pembayaran Tokomodal, serta validasi tanggal *expired* yang disesuaikan secara otomatis. Tujuan skenario ini adalah memastikan bahwa sistem bekerja sesuai logika bisnis dan ketentuan operasional yang telah ditetapkan.



Gambar 3. 31 Intervensi Data

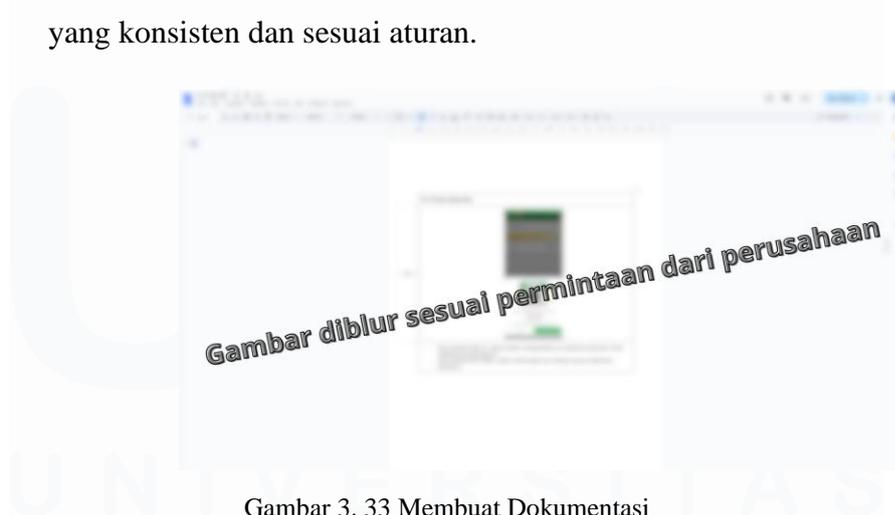
Gambar 3.31 adalah intervensi data dilakukan dengan mengatur tanggal sistem dan data *grand opening* secara manual di database untuk mensimulasikan kondisi checkout pada H-4, H-3, H-1, dan H+1. Intervensi ini membantu mempercepat pengujian tanpa harus menunggu waktu nyata, serta memungkinkan *quality assurance* memverifikasi

apakah notifikasi, batasan metode pembayaran, dan perhitungan *expired date* telah sesuai.



Gambar 3. 32 Mencatat *Buglist*

Gambar 3.32 merupakan beberapa bug ditemukan selama pengujian, di antaranya *popup* tidak muncul pada *checkout* sebelum H-3, metode pembayaran Tokomodal masih aktif saat seharusnya dinonaktifkan, dan *expired date* tidak sesuai dengan ketentuan (harus H+1 dari tanggal GO). Bug lain termasuk kesalahan tampilan pada informasi konfirmasi pengiriman di H-1. Semua *bug* dicatat dalam sistem pelaporan dan telah diperbaiki untuk memastikan pengalaman *checkout* yang konsisten dan sesuai aturan.



Gambar 3. 33 Membuat Dokumentasi

Gambar 3.33 adalah dokumentasi yang disusun untuk menjelaskan alur proses *checkout* yang terkait dengan tanggal *Grand Opening Stock Point*. Di dalamnya mencakup diagram alur, daftar validasi waktu *checkout*, pembatasan metode pembayaran, serta pengaturan tanggal

pengiriman dan *expired order*. Dokumentasi ini berfungsi sebagai referensi bagi tim pengembang dan QA untuk menjaga agar fitur tetap sesuai dengan kebutuhan bisnis dan tidak menimbulkan kebingungan bagi pengguna.

3.2.8 Banner on AMS

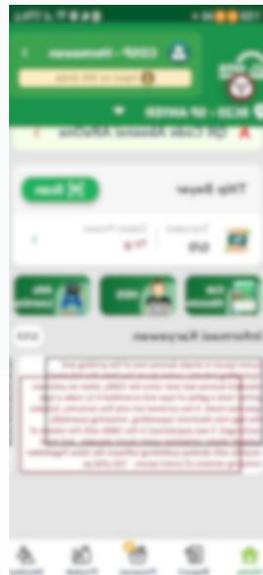
Banner pada sistem *Aksesmu Management System* (AMS) berfungsi sebagai media informasi visual yang ditujukan untuk tim *Stock Point*. Proses penginputan *banner* ini bertujuan untuk memastikan penyampaian informasi terbaru terkait Aksesmu dapat tersampaikan secara cepat dan efektif. Melalui *banner*, berbagai pengumuman penting seperti kebijakan cuti, update sistem, atau informasi internal lainnya dapat ditampilkan langsung di halaman utama *Aksesmu Management System* (AMS). Dengan adanya fitur ini, komunikasi internal menjadi lebih efisien dan terpusat, sehingga mendukung kelancaran operasional tim di lapangan.



Gambar 3. 34 *Testing Website Back Office*

Gambar 3.34 adalah pengujian yang dilakukan pada website *Back Office* Aksesmu untuk memastikan bahwa fitur input *banner* berfungsi dengan baik dan sesuai ketentuan. Proses *testing* mencakup validasi seluruh form *input* seperti unggah gambar banner, penentuan periode tayang (tanggal mulai dan berakhir), status aktif/non-aktif, serta urutan prioritas tampilan. Selain itu, dilakukan juga pengujian terhadap

behavior sistem saat banner diubah, dihapus, atau ditambahkan lebih dari satu secara bersamaan. Fokus utama pengujian adalah memastikan bahwa data yang dimasukkan melalui *Back Office* tersimpan dengan benar di database dan ditampilkan secara real-time di aplikasi *Aksesmu Management System (AMS)*. Respons sistem, kecepatan *load*, serta kesesuaian tampilan *preview* juga menjadi bagian penting dalam pengujian ini. Website *Back Office* harus mampu menangani berbagai skenario input agar dapat digunakan secara fleksibel oleh admin untuk mengelola informasi *banner* yang ditampilkan ke pengguna lapangan.



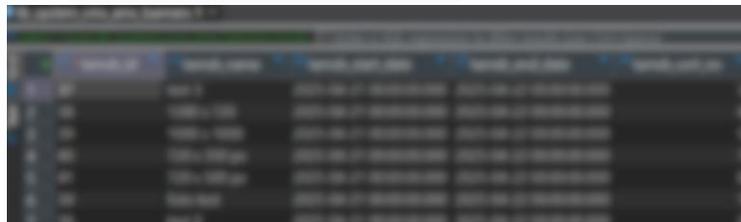
Gambar 3. 35 *Testing* Aplikasi AMS

Gambar 3.35 adalah pengujian yang dilakukan pada dua platform, yaitu website *Back Office* sebagai pusat *input* banner, dan aplikasi *Aksesmu Management System (AMS)* sebagai tempat tampilan banner kepada user. Pada *Back Office*, diuji proses input *banner* mulai dari upload gambar, penentuan tanggal tayang, serta aktivasi *banner*. Di AMS, diuji apakah *banner* tampil sesuai jadwal, apakah gambar termuat dengan baik, serta apakah tampilan menyesuaikan resolusi perangkat. Pengujian juga memastikan sinkronisasi antara sistem *backend* dan *frontend* berjalan *real-time* tanpa *delay*.



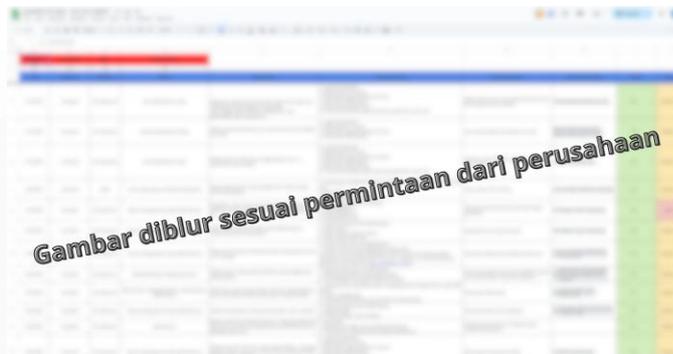
Gambar 3. 36 Membuat Skenario

Gambar 3.36 adalah skenario pengujian mencakup kondisi *banner* aktif dan tidak aktif, perubahan jadwal tayang, *banner* dengan format file tidak sesuai, serta pengujian *multiple banner* dengan urutan tampilan. Juga diuji kondisi ketika *banner* telah kedaluwarsa, untuk memastikan tidak lagi muncul di aplikasi AMS. Skenario ini bertujuan memastikan bahwa seluruh logika tampilan *banner* berjalan sesuai aturan yang telah ditentukan.



Gambar 3. 37 Intervensi Data

Gambar 3.37 adalah intervensi data yang dilakukan dengan mengatur tanggal mulai dan akhir penayangan *banner* di database, serta memanipulasi status aktif/tidak aktif. Ini dilakukan untuk menguji apakah sistem tetap konsisten dalam menampilkan *banner* sesuai kondisi *real-time*. Intervensi juga digunakan untuk menguji perilaku aplikasi saat menerima data *banner* yang tidak valid atau format gambar yang tidak sesuai spesifikasi.



Gambar 3. 38 Mencatat *Buglist*

Gambar 3.38 adalah beberapa bug yang ditemukan antara lain: banner tidak tampil meskipun status aktif, keterlambatan load gambar di aplikasi AMS, serta penjadwalan *banner* yang tidak tepat (terlalu cepat atau lambat tayang). Selain itu, ditemukan juga tampilan *banner* yang terpotong pada resolusi layar tertentu. Semua bug dicatat dalam sistem *tracking*, diberi label prioritas, dan diperbaiki secara bertahap untuk memastikan fitur berjalan optimal di berbagai kondisi perangkat.



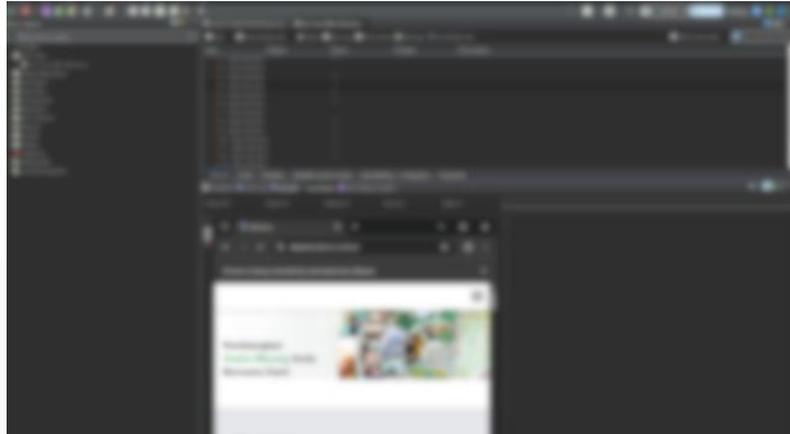
Gambar 3. 39 Membuat Dokumentasi

Gambar 3.39 adalah dokumentasi yang dibuat untuk menjelaskan alur kerja fitur *banner* mulai dari proses input, penjadwalan, hingga tampilan akhir di aplikasi. Disertakan pula panduan penggunaan fitur bagi admin *Back Office*, struktur data *banner*, serta referensi API jika ada. Dokumentasi ini penting sebagai referensi tim *developer* dan *quality*

assurance untuk menjaga konsistensi fungsi dan memudahkan perawatan sistem di masa mendatang.

3.2.9 Automation Test Menggunakan Katalon

Pada tahap ini, dilakukan dengan pengujian otomatisasi (automation testing) terhadap website dan aplikasi Aksesmu untuk memastikan bahwa tampilan dan fungsi berjalan konsisten di berbagai perangkat menggunakan. Pengujian difokuskan pada responsivitas UI dan performa fitur pada beberapa jenis handphone, termasuk perangkat seperti Redmi, yang dalam praktiknya ditemukan beberapa kendala seperti tombol tidak merespons atau tampilan tidak sesuai. Hal ini menunjukkan bahwa meskipun sistem berjalan dengan baik di sebagian besar perangkat, tetap diperlukan pengujian lintas perangkat untuk menjamin pengalaman pengguna yang optimal. Oleh karena itu, dilakukan automation testing berdasarkan skenario yang telah disusun, untuk mendeteksi *bug* yang muncul pada resolusi atau sistem operasi tertentu. Dokumentasi dibuat untuk mencatat hasil pengujian dan perbedaan perilaku antarfungsi pada masing-masing *device*, sehingga tim developer dapat segera menindaklanjuti perbaikannya.



Gambar 3. 40 Membuat Dokumentasi

Gambar 3.40 merupakan dokumentasi visual saat proses automation testing dilakukan pada website dan aplikasi Aksesmu menggunakan berbagai perangkat. Pengujian dilakukan untuk memastikan setiap elemen UI/UX berfungsi sebagaimana mestinya di berbagai jenis handphone. Salah satu temuan pada pengujian ini adalah tombol yang tidak merespons pada *device* tertentu seperti Redmi. Gambar ini menunjukkan secara langsung proses identifikasi masalah, yang nantinya akan dicatat dalam dokumentasi bug untuk ditindaklanjuti oleh tim *quality assurance* dan developer.

U M N
U N I V E R S I T A S
M U L T I M E D I A
N U S A N T A R A

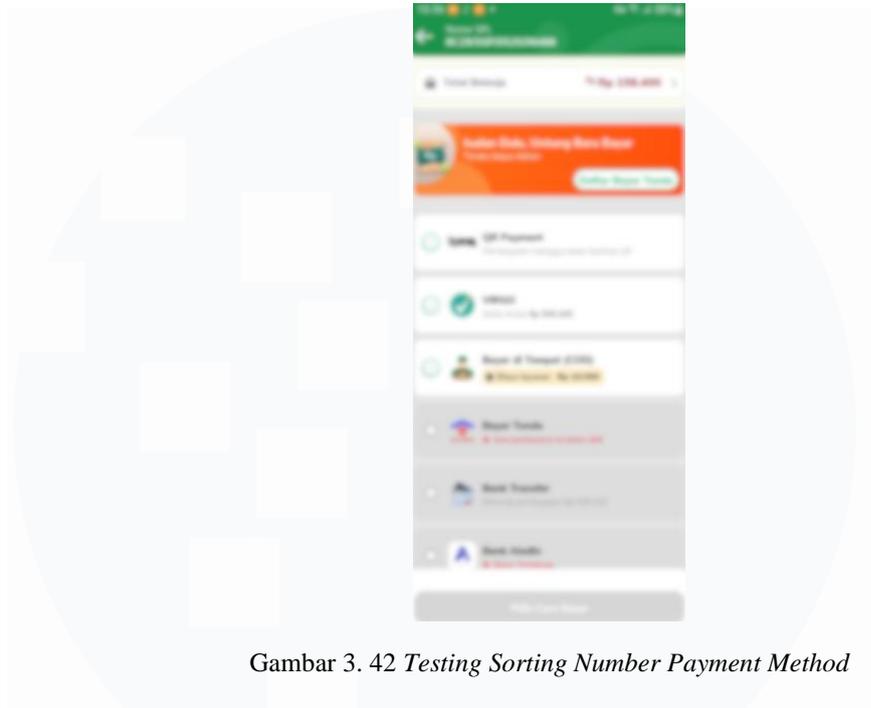
ID	Scenario	Steps	Status
1	Scenario 1	Step 1.1, Step 1.2, Step 1.3	Pass
2	Scenario 2	Step 2.1, Step 2.2, Step 2.3	Pass
3	Scenario 3	Step 3.1, Step 3.2, Step 3.3	Pass
4	Scenario 4	Step 4.1, Step 4.2, Step 4.3	Pass
5	Scenario 5	Step 5.1, Step 5.2, Step 5.3	Pass
6	Scenario 6	Step 6.1, Step 6.2, Step 6.3	Pass
7	Scenario 7	Step 7.1, Step 7.2, Step 7.3	Pass
8	Scenario 8	Step 8.1, Step 8.2, Step 8.3	Pass
9	Scenario 9	Step 9.1, Step 9.2, Step 9.3	Pass
10	Scenario 10	Step 10.1, Step 10.2, Step 10.3	Pass
11	Scenario 11	Step 11.1, Step 11.2, Step 11.3	Pass
12	Scenario 12	Step 12.1, Step 12.2, Step 12.3	Pass
13	Scenario 13	Step 13.1, Step 13.2, Step 13.3	Pass
14	Scenario 14	Step 14.1, Step 14.2, Step 14.3	Pass
15	Scenario 15	Step 15.1, Step 15.2, Step 15.3	Pass
16	Scenario 16	Step 16.1, Step 16.2, Step 16.3	Pass
17	Scenario 17	Step 17.1, Step 17.2, Step 17.3	Pass
18	Scenario 18	Step 18.1, Step 18.2, Step 18.3	Pass
19	Scenario 19	Step 19.1, Step 19.2, Step 19.3	Pass
20	Scenario 20	Step 20.1, Step 20.2, Step 20.3	Pass
21	Scenario 21	Step 21.1, Step 21.2, Step 21.3	Pass
22	Scenario 22	Step 22.1, Step 22.2, Step 22.3	Pass
23	Scenario 23	Step 23.1, Step 23.2, Step 23.3	Pass
24	Scenario 24	Step 24.1, Step 24.2, Step 24.3	Pass
25	Scenario 25	Step 25.1, Step 25.2, Step 25.3	Pass
26	Scenario 26	Step 26.1, Step 26.2, Step 26.3	Pass
27	Scenario 27	Step 27.1, Step 27.2, Step 27.3	Pass
28	Scenario 28	Step 28.1, Step 28.2, Step 28.3	Pass
29	Scenario 29	Step 29.1, Step 29.2, Step 29.3	Pass
30	Scenario 30	Step 30.1, Step 30.2, Step 30.3	Pass
31	Scenario 31	Step 31.1, Step 31.2, Step 31.3	Pass
32	Scenario 32	Step 32.1, Step 32.2, Step 32.3	Pass
33	Scenario 33	Step 33.1, Step 33.2, Step 33.3	Pass
34	Scenario 34	Step 34.1, Step 34.2, Step 34.3	Pass
35	Scenario 35	Step 35.1, Step 35.2, Step 35.3	Pass
36	Scenario 36	Step 36.1, Step 36.2, Step 36.3	Pass
37	Scenario 37	Step 37.1, Step 37.2, Step 37.3	Pass
38	Scenario 38	Step 38.1, Step 38.2, Step 38.3	Pass
39	Scenario 39	Step 39.1, Step 39.2, Step 39.3	Pass
40	Scenario 40	Step 40.1, Step 40.2, Step 40.3	Pass
41	Scenario 41	Step 41.1, Step 41.2, Step 41.3	Pass
42	Scenario 42	Step 42.1, Step 42.2, Step 42.3	Pass
43	Scenario 43	Step 43.1, Step 43.2, Step 43.3	Pass
44	Scenario 44	Step 44.1, Step 44.2, Step 44.3	Pass
45	Scenario 45	Step 45.1, Step 45.2, Step 45.3	Pass
46	Scenario 46	Step 46.1, Step 46.2, Step 46.3	Pass
47	Scenario 47	Step 47.1, Step 47.2, Step 47.3	Pass
48	Scenario 48	Step 48.1, Step 48.2, Step 48.3	Pass
49	Scenario 49	Step 49.1, Step 49.2, Step 49.3	Pass
50	Scenario 50	Step 50.1, Step 50.2, Step 50.3	Pass

Gambar 3. 41 *Testing* Berdasarkan Skenario

Gambar 3.41 menggambarkan aktivitas *automation testing* yang dilakukan berdasarkan skenario yang telah disusun sebelumnya. Skenario mencakup variasi tipe perangkat, resolusi layar, dan sistem operasi untuk memastikan aplikasi tetap berfungsi optimal dalam kondisi penggunaan yang berbeda. Dari skenario ini ditemukan bahwa beberapa perangkat seperti Redmi mengalami kendala pada elemen UI tertentu, sehingga diperlukan penyesuaian dan pengujian ulang secara spesifik.

3.2.10 *Enhancement Sorting Number Payment Method*

Salah satu tujuan dari fitur *enhancement* ini adalah untuk mengatur ulang tampilan urutan metode pembayaran di halaman *checkout* sesuai dengan preferensi pengguna sebelumnya. Dengan menggunakan logika pengaturan yang bergantung pada *sortNo*, sistem dapat menampilkan metode pembayaran terakhir yang digunakan pengguna di posisi paling atas, sementara metode lainnya mengikuti urutan *default* yang telah ditentukan sebelumnya. Peningkatan ini bertujuan untuk memberikan pengalaman pengguna yang lebih personal dan cepat serta meningkatkan efisiensi proses *checkout*.



Gambar 3. 42 Testing Sorting Number Payment Method

Gambar 3.42 dilakukan untuk memastikan bahwa sistem dapat menyimpan dan menampilkan metode pembayaran terakhir pengguna di urutan teratas saat transaksi berikutnya dilakukan. Proses dimulai dengan menguji metode pembayaran yang dipilih, seperti QRIS atau Aladin, setelah transaksi pertama. Setelah transaksi selesai, metode tersebut diuji ulang untuk memastikan bahwa metode tersebut muncul di posisi paling atas (*sortNo* = 1) pada checkout berikutnya. Selain itu, kami menguji skenario pergantian metode pembayaran secara berturut-turut untuk mengetahui apakah sistem benar-benar memperbarui urutan saat metode berubah. Untuk memastikan kembali ke urutan *default*, berbagai kondisi akun, *logout*, dan reset data preferensi diuji. Hasil pengujian menunjukkan bahwa sistem dapat secara dinamis mengatur dan mengubah urutan metode pembayaran sesuai dengan preferensi pengguna.



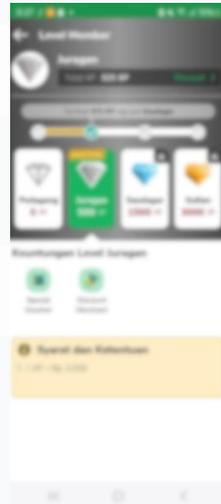
Gambar 3. 43 Membuat Dokumentasi

Gambar 3.43 logika pengurutan metode pembayaran berdasarkan *sortNo*, alur penyimpanan preferensi pengguna, dan skenario perubahan urutan dijelaskan. Diagram alur, struktur data penyimpanan preferensi metode pembayaran, dan contoh output tampilan disertakan untuk membuatnya lebih mudah dipahami oleh tim QA, *developer*, dan *stakeholder* terkait. Jika ada penyesuaian atau penambahan metode pembayaran baru di kemudian hari, dokumen ini dapat digunakan sebagai referensi.

3.2.11 Grading Member

Pada aplikasi Aksesmu, fitur *grading member* dapat digunakan untuk mengkategorikan anggota ke dalam tingkatan atau *level* tertentu berdasarkan aktivitas dan perilaku tertentu, seperti total transaksi, frekuensi belanja, dan perilaku pembayaran. Tujuan dari sistem *grading* ini adalah untuk meningkatkan pengelolaan loyalitas pelanggan, promosi yang lebih tepat sasaran, dan personalisasi layanan. Setiap anggota akan diberi kelas tertentu, seperti *Bronze*,

Silver, Gold, atau Platinum, yang diperbarui secara berkala berdasarkan data terbaru.



Gambar 3. 44 *Testing Grading Member*

Gambar 3.44 adalah proses pengujian dilakukan untuk memastikan bahwa sistem *grading* berjalan sesuai dengan rencana bisnis. Pengguna baru yang belum melakukan transaksi, pengguna aktif yang memiliki banyak transaksi, dan pengguna dengan pembayaran yang tidak lancar adalah beberapa skenario yang diuji. Pengujian memvalidasi apakah *grade* yang ditetapkan sesuai dengan parameter yang tepat dan apakah perubahan *grade* terjadi secara otomatis saat data transaksi berubah. Diuji juga tampilan *grade* di halaman akun pengguna dan apakah informasi *grade* memengaruhi akses ke fitur atau penawaran tertentu.



Gambar 3. 45 Skenario

Gambar 3.45 adalah pengujian fitur *Grading Member OBA* dilakukan untuk memastikan bahwa sistem pemberian XP dan peningkatan *grade* berjalan sesuai logika yang ditentukan. Pengujian mencakup skenario transaksi pertama kali setelah OBA menjadi *full member*, di mana pengguna mendapatkan bonus XP tambahan dan XP dasar berdasarkan nominal transaksi. Jika transaksi dibatalkan dan dilakukan ulang, hanya transaksi pertama yang mendapat bonus XP. Selain itu, sistem diuji terhadap batas *limit* XP harian, yang memastikan bahwa setelah batas tercapai, transaksi berikutnya tidak menghasilkan XP. Pengujian juga mencakup kondisi ketika *order* dibatalkan (*cancel by DRO*) dan dilakukan kembali, serta kombinasi penggunaan voucher dan kode PAS. Metode pembayaran seperti COD, VA, QRIS, Tokomodal, Virgo, dan Aladin juga diuji untuk memastikan tetap menghasilkan XP. Untuk member non-mandiri, sistem divalidasi agar tidak memberikan XP meskipun transaksi dilakukan. Proses update XP dan *grade* diuji berdasarkan total XP yang tercatat dalam histori setelah transaksi selesai dan berhasil (POD), termasuk validasi terhadap tanggal *expired* XP. Terakhir, elemen tampilan UI seperti halaman home, akun, dan detail *grade* juga diuji untuk memastikan informasi *grade*, XP, dan *benefit* ditampilkan sesuai desain. Semua skenario ini bertujuan memastikan

sistem *grading* akurat, adil, dan berjalan stabil dalam berbagai kondisi penggunaan.



Gambar 3. 46 Intervensi Data

Gambar 3.46 merupakan intervensi data yang dilakukan untuk mempercepat proses pengujian *grading* tanpa menunggu aktivitas transaksi nyata dari pengguna. Data transaksi, status pembayaran, dan frekuensi belanja dimodifikasi langsung di database untuk menghasilkan kondisi yang dibutuhkan sesuai skenario. Intervensi juga digunakan untuk menguji respons sistem terhadap data tidak normal atau ekstrem, seperti jumlah transaksi sangat tinggi atau status pembayaran gagal berulang kali. Hal ini membantu memastikan bahwa sistem *grading* tetap akurat dan tidak mudah terpengaruh oleh data yang tidak sesuai.

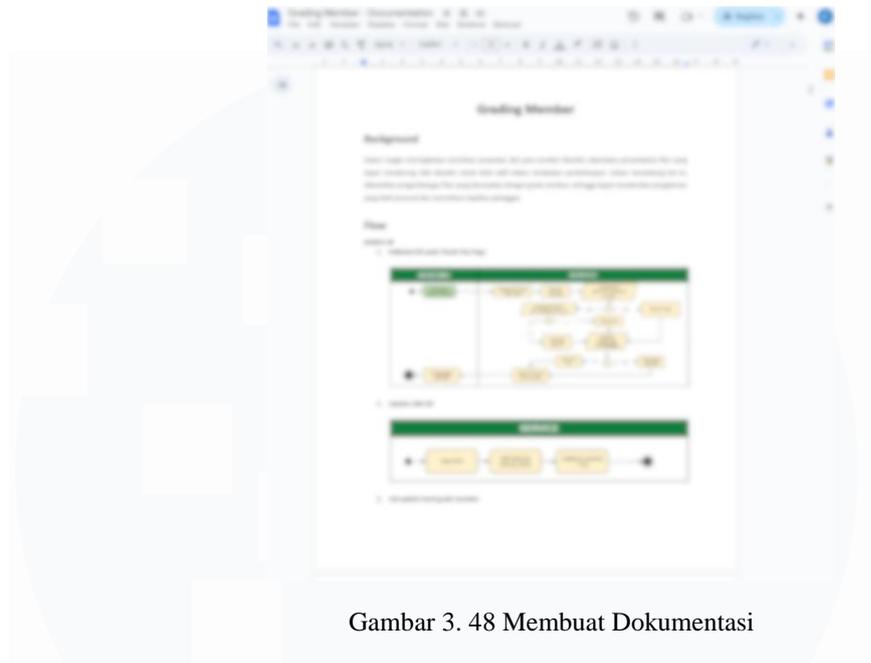
U N I V E R S I T A S
M U L T I M E D I A
N U S A N T A R A



ID	Description	Status	Priority
1	Value not matching actual value	Open	High
2	Update value not automatic after new transaction	Open	Medium
3	Data value not synchronized between frontend and backend	Open	High
4	User not meeting requirements still getting high value due to calculation error	Open	High
5	Calculation error in limit range	Open	High

Gambar 3. 47 Buglist

Gambar 3.47 merupakan tahapan selama pengujian, beberapa *bug* ditemukan, seperti tampilan nilai tidak sesuai dengan nilai aktual, *update* nilai tidak berjalan otomatis setelah transaksi baru, dan data nilai tidak sinkron antara *frontend* dan *backend*. Selain itu, terdapat kasus di mana pengguna yang tidak memenuhi syarat tetap mendapatkan nilai tinggi karena kesalahan dalam perhitungan ambang batas. Tim pengembang mencatat semua *bug* ini dalam sistem pelaporan dan kemudian membaginya berdasarkan tingkat kepentingan.



Gambar 3. 48 Membuat Dokumentasi

Gambar 3.48 adalah dokumentasi yang disusun untuk menjelaskan mekanisme grading, termasuk kriteria penilaian, logika penghitungan, dan frekuensi *update* data. Di dalamnya juga terdapat diagram alur proses grading, referensi data yang digunakan (seperti jumlah transaksi dan status pembayaran), serta integrasi dengan tampilan di aplikasi. Dokumentasi ini menjadi acuan penting bagi tim *quality assurance*, developer, dan *business* untuk memastikan konsistensi dan keakuratan dalam pengelolaan sistem grading ke depannya.

3.3 Kendala yang Ditemukan

Selama melakukan tugas selama melaksanakan magang sebagai *quality assurance* di Aksesmu, terdapat beberapa kendala seperti:

- 1) Perubahan kebutuhan yang mempengaruhi hasil seperti pengembangan yang sering kali terdapat perubahan dari *system design* awal. Hal ini sering terjadi ketika siklus pengembangan aplikasi dengan metode agile, yang menyebabkan ketidaksesuaian dalam system design dan pengujian sudah dilakukan sebelumnya.

- 2) Skenario yang dibuat terkadang tidak sepenuhnya terorganisir dengan baik sehingga menyebabkan keterlambatan pada publikasi hasil pengujian dan komunikasi dalam tim.
- 3) Kesulitan dalam mendeteksi *edge cases* dalam beberapa skenario pengujian yang tidak termasuk dalam *system design* dan UI/UX yang diberikan sehingga mengarah pada temuan bug yang tidak terduga yang membutuhkan penanganan lebih lanjut.

3.4 Solusi atas Kendala yang Ditemukan

Solusi yang dilakukan ketika terdapat kendala pada saat pelaksanaan magang meliputi:

- 1) Mengelola perubahan kebutuhan dengan menerapkan komunikasi didalam tim pengembang dan produk untuk memahami perubahan sejak awal dan melakukan perubahan skenario pengujian secara berkala supaya tetap sesuai dengan perubahan *system design*.
- 2) Meningkatkan kualitas skenario pengujian agar lebih terstruktur dan mudah dipahami. Selain itu, melakukan *review* bersama tim sebelum pengujian dimulai supaya bisa memastikan skenario yang dibuat.
- 3) Meningkatkan cakupan pengujian eksplorasi lebih luas, termasuk *edge cases* yang mungkin tidak terdeteksi dalam desain awal.
- 4) Meningkatkan koordinasi didalam tim dengan mengadakan sesi diskusi untuk memastikan semua anggota memiliki pemahaman yang sama mengenai perubahan yang terjadi dan mempercepat proses *debugging* serta publikasi produk.