BAB I

PENDAHULUAN

A. Latar Belakang

Transaksi merupakan proses pertukaran antara dua pihak atau lebih dengan tujuan tertentu, yang saat ini semakin mudah dilakukan tanpa batasan ruang dan waktu berkat kemajuan teknologi digital (Nasution dkk., 2020). Pembeli dan penjual dapat melakukan transaksi kapan saja dan di mana saja, termasuk dalam konteks jual beli akun *game online* yang semakin populer. Menurut Wijaya dkk. (2021), meskipun kemajuan ini memberikan kemudahan, tantangan utama dalam transaksi digital, khususnya jual beli akun *game online*, adalah munculnya keraguan dan ketidakpastian antara kedua belah pihak. Risiko penipuan, seperti penjual yang tidak mengirimkan akun setelah menerima pembayaran atau pembeli yang membatalkan transaksi secara sepihak, menjadi permasalahan yang signifikan dalam ekosistem ini. Ketidakpastian ini menegaskan pentingnya transparansi dalam setiap transaksi.

Ditinjau dari sisi kerugian, telaah ilmiah menunjukkan tingginya tingkat menjadi korban dalam praktik perdagangan akun *game*. Survei representatif pada remaja melaporkan 35,8% dari mereka yang pernah berdagang akun atau item pernah menjadi korban *trade scam* (Kristiansen & Jensen, 2023). Studi pada komunitas *game*r menemukan 23% responden pernah mengalami kerugian properti *virtual*, dan 16% di antaranya terjadi saat proses perdagangan (Kristiansen, dkk, 2024). Rangkaian temuan ini menegaskan adanya risiko kerugian finansial dan

hilangnya aset yang signifikan dalam pasar jual-beli akun *game*, sehingga dibutuhkan mekanisme transaksi yang lebih tepercaya.

Dalam transaksi, baik yang daring maupun luring, sering timbul ketidakpastian yang dapat merugikan kedua belah pihak. Misalnya, pembeli mungkin tidak yakin apakah penjual akan mengirimkan akun setelah pembayaran dilakukan, sementara penjual khawatir tidak menerima pembayaran setelah mengirimkan akun (Yunitarini dkk., 2023). Hal ini menciptakan risiko yang dapat menghambat pertumbuhan pasar digital.

Teknologi *blockchain* menawarkan solusi yang ideal untuk permasalahan transaksi dengan sistem terdesentralisasi yang menjamin transparansi (Iqbal & Ahmad, 2024). Setiap transaksi dicatat secara permanen dalam blok-blok data yang terhubung dan terenkripsi, yang tidak dapat diubah setelah diverifikasi. Fitur ini efektif mencegah berbagai bentuk kecurangan, seperti manipulasi data dan pembatalan transaksi sepihak, serta menciptakan ekosistem yang lebih dapat dipercaya.

Riwayat transaksi disimpan dalam sistem terdistribusi yang terbuka, memungkinkan verifikasi oleh pihak berwenang. Mekanisme ini memastikan pertanggungjawaban melalui transparansi catatan transaksi. Dengan demikian, *blockchain* menawarkan solusi baru dalam sistem transaksi digital yang mengutamakan efisiensi (Sari & Nasution, 2024).

Penelitian ini bertujuan untuk mensimulasikan penggunaan *blockchain* dalam transaksi jual beli akun *game online* guna meningkatkan transparansi dan kepercayaan antara penjual dan pembeli. Dengan memanfaatkan teknologi ini, diharapkan transaksi dapat dilakukan secara aman dan terjamin, menurunkan risiko

penipuan, dan pada akhirnya mendorong pertumbuhan pasar jual beli akun *game online* yang lebih tepercaya.

B. Rumusan Masalah

Berdasarkan latar belakang yang telah diuraikan, dapat disimpulkan permasalahan yang dihadapi dalam implementasi teknologi *blockchain* ini yaitu:

1. Bagaimana membanguan desain *website* penjualan akun *game online* yang interaktif?
2. Bagaimana mengimplementasikan metode *blockchain* dengan *smart contract escrow* dalam *website* penjualan akun *game online* untuk transparansi transaksi penjualan?
3. Bagaimana mensimulasikan sistem ini agar bisa di implementasikan secara nyata?

C. Tujuan Penelitian

Tujuan dari penelitian ini adalah :

1. Membangun desain *website* jual beli akun *game online* yang interaktif.
2. Mengimplementasikan metode *blockchain* dengan *smart contract escrow* dalam *website* penjualan akun *game online* untuk transparansi transaksi penjualan.
3. Mensimulasikan sistem ini agar bisa di implementasikan secara nyata.

D. Batasan Masalah

Dalam Pembuatan "Simulasi *Blockchain* untuk Transparansi Transaksi Pada Jual Beli Akun *Game online*" ini mempunyai batasan masalah sebagai berikut

1. Penelitian ini hanya fokus pada transaksi jual beli akun *game online dengan* kategori *mobile,* yaitu *mobile legends: bang bang, free fire, pubg mobile* dan *genshin impact*, bukan pada jual beli item atau mata uang dalam *game*.
2. Simulasi *blockchain* yang dilakukan terbatas pada penggunaan *Smart contract escrow* untuk memverifikasi transaksi dan memastikan transparansi antara pembeli dan penjual.
3. Penelitian ini tidak membahas aspek hukum atau regulasi yang terkait dengan jual beli akun *game online*, melainkan hanya berfokus pada penerapan teknologi *blockchain* untuk memastikan transparansi dalam transaksi.
4. Sistem simulasi terbatas pada *tesnet ethereum sepolia*, tidak mencakup *mainnet ethereum* karena biaya transaksi.

E. Manfaat Penelitian

Adapun manfaat penelitian ini sebagai berikut:

1. Memberikan pemahaman tentang penerapan teknologi *blockchain* dalam transaks, khususnya untuk industri *game online*.
2. Meningkatkan transparansi dalam transaksi jual beli akun *game*, yang dapat mengurangi potensi penipuan dan ketidakpercayaan antara pembeli dan penjual.
3. Menyediakan solusi praktis untuk meningkatkan kepercayaan dan mempercepat pertumbuhan pasar jual beli akun *game online* yang lebih aman.

`

BAB II

TINJAUAN PUSTAKA

A. Kajian Teori

1. *Website*

Website merupakan sebuah kumpulan halaman *web* yang terhubung dan diakses melalui internet. Setiap halaman *web* biasanya berisi teks, gambar, video, dan elemen multimedia lainnya yang dirancang untuk memberikan informasi atau memberikan pengalaman interaktif kepada pengunjungnya. *Website* dapat diakses melalui peramban *web (web browser)* seperti *Google Chrome, Mozilla Firefox*, atau *Safari* dengan memasukkan alamat *URL* yang sesuai (Rina Noviana, 2022).

Menurut Soedewi (2022), *website* dapat diartikan sebagai sarana yang memenuhi kebutuhan pengguna untuk meningkatkan kepercayaan dalam mencari konten informasi. Pada dasarnya, *website* berfungsi sebagai penyimpanan informasi yang membantu pengguna menemukan apa yang mereka butuhkan, sekaligus meningkatkan pengalaman interaksi *online*.

2. *Solidity*

Solidity adalah bahasa pemrograman yang digunakan untuk menulis kontrak pintar (*smart contract*) pada platform *blockchain*, khususnya *ethereum*. *Solidity* merupakan bahasa yang dikembangkan untuk memudahkan pengembang dalam menulis kode yang akan dieksekusi di jaringan *blockchain*, sehingga memastikan transparansi dalam transaksi digital. Solidity sebagai bahasa untuk menulis *smart contract* (Buterin, 2014) relevan dalam penelitian ini karena memungkinkan

otomatisasi transaksi *escrow*. Dengan demikian, risiko penjual tidak menyerahkan akun setelah pembayaran dapat diminimalisir. Bahasa ini memiliki struktur sintaks yang mirip dengan *JavaScript*, sehingga memudahkan pengembang yang sudah familiar dengan *JavaScript* dalam mempelajari *solidity*.

3. *Blockchain*

Blockchain adalah teknologi basis data terdesentralisasi yang digunakan untuk mencatat transaksi secara transparan, aman, dan permanen tanpa brgantung pada otoritas pusat seperti bank atau pemerintahan. Teknologi ini pertama kali diperkenalkan oleh Satoshi Nakamoto dan diimplementasikan dalam Bitcoin (Nakamoto, 2008), sebagai mata uang digital berbasis *blockchain*.

Blockchain merupakan teknologi pencatatan data digital terdistribusi yang menyusun informasi dalam blok-blok terenkripsi yang saling terhubung membentuk rantai. Setiap blok menyimpan data transaksi, *timestamp*, dan *hash* yang mengaitkannya dengan blok sebelumnya, menciptakan struktur data yang bersifat permanen dan tidak dapat diubah (Supiandi, 2024). Sistem ini beroperasi secara terdesentralisasi tanpa otoritas pusat, di mana validasi transaksi dilakukan oleh jaringan node melalui mekanisme konsensus yang tercatat dalam salinan *ledger* (buku besar *blockchain*).

Cara kerja *blockchain*:

a. Pembuatan blok

   Pembuatan blok saat transaksi terjadi, data dikumpulkan dalam sebuah blok. Setiap blok memiliki header yang mencakup *hash* blok sebelumnya, *timpstamp*, *nonce* dan daftar transaksi.

b. Verfikasi dan Konsensus

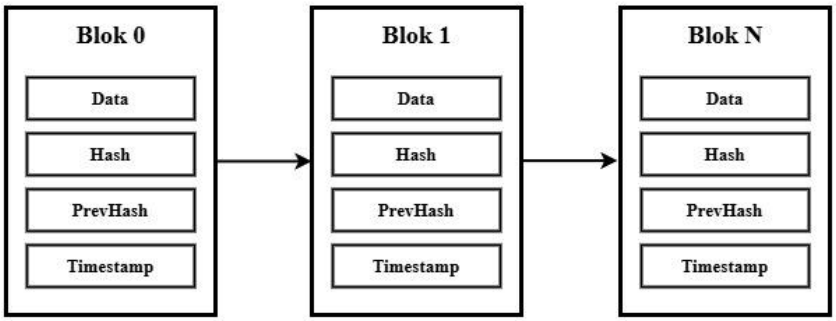
Verfikasi dan Konsensus blok baru diverfikasi oleh jaringan *node* melalui mekanisme konsensus, *seperti Proof of Stake (PoS)* di *ethereum* berdasarkan *aset* mereka.

c. Panambahan ke rantai

Penambahan ke rantai setelah diverfikasi, blok ditambahkan ke rantai *blockchain* yang ada. Setiap *node* memperbarui salinan *blockchain* mereka, menciptakan sistem terdistribusi.

d. Keamanan dan Transparansi

Keamanan dan Transparansi data dienkripsi menggunakan kriptografi, dan sekali ditambahkan, blok tidak bisa diubah tanpa mengubah seluruh rantai.



Gambar 2. 1 Tahap proses blok *blockchain*

4. *Smart contract*

Smart contract adalah program komputer yang secara otomatis mengeksekusi, mengontrol, atau mendokumentasikan perjanjian atau transaksi yang tercatat dalam *blockchain*. *Smart contract* memungkinkan dua pihak atau lebih untuk melakukan transaksi yang aman, terverifikasi. Kontrak ini berjalan berdasarkan kode yang telah diprogram, sehingga tidak ada ruang untuk kesalahan

manusia atau penipuan, dan semua ketentuan kontrak dieksekusi secara otomatis begitu kondisi tertentu dipenuhi (Kusyanti, dkk., 2023).

Keunggulan utama *smart contract* yaitu sifatnya yang tidak dapat diubah (*immutable*) dan transparan. Setelah disimpan di *blockchain*, kontrak tidak bisa dimodifikasi, sedangkan semua transaksinya dapat diverifikasi oleh pihak terkait (Buterin, 2014). Dengan sistem ini, proses bisnis menjadi lebih efisien karena mengurangi risiko kesalahan manusia. Teknologi ini memberikan solusi yang handal untuk berbagai kebutuhan transaksi digital.

Smart contract pada dasarnya terdiri dari beberapa komponen inti:

a. *state variables* untuk menyimpan keadaan/rekam jejak transaksi
b. *functions* yang mengeksekusi aturan-aturan kontrak
c. *events* yang memancarkan log ke *blockchain* agar setiap perubahan keadaan dapat ditelusuri
d. *modifiers* sebagai prasyarat eksekusi fungsi (misalnya hanya admin yang boleh memutus sengketa). Sifatnya yang *immutable* dan transparan memastikan seluruh ketentuan berjalan otomatis begitu syarat terpenuhi, serta jejaknya dapat diverifikasi publik.

Dalam konteks *escrow* jual beli akun *game*, siklus hidup kontrak dimulai dari *deploy* kontrak. Selanjutnya pembeli melakukan deposit (status *funded*), disusul penjual menyerahkan akun. Setelah itu pembeli mengonfirmasi atau mengajukan sengketa. Jika terjadi sengketa, admin memutus apakah dana dirilis ke penjual atau dikembalikan kepada pembeli.

5. *Ethereum*

Ethereum merupakan sebuah jaringan *blockchain* yang memungkinkan transaksi uang digital dilakukan tanpa melibatkan bank atau perantara lainnya. Setiap transaksi yang dilakukan dalam *ethereum* diproses oleh komputer yang tersebar di seluruh dunia, kemudian dicatat dalam buku besar digital yang bersifat transparan dan dapat diakses oleh semua pihak (Mabruroh, dkk, 2021). Sistem ini memastikan keterbukaan yang tinggi, menjadikannya tidak dapat untuk dimanipulasi.

Ethereum dikembangkan dengan menggunakan *solidity*, sebuah bahasa pemrograman yang dirancang khusus untuk membuat *smart contract*, yaitu kode yang berjalan di *blockchain* dan mengeksekusi perintah secara otomatis ketika syarat tertentu terpenuhi (Aji, dkk, 2023). Smart contract inilah yang menjadi inti *ethereum*, memungkinkan pembuatan aplikasi yang berjalan tanpa perantara karena aturannya telah ditetapkan dalam kode. Dengan demikian, *solidity* dan *smart contract* membentuk dasar dari ekosistem *ethereum*, memastikan bahwa segala operasi berjalan transparan, dan sesuai aturan yang telah diprogram.

Dalam proses pengembangan dan pengujian, *ethereum* juga menyediakan jaringan uji coba *(testnet)* yang berfungsi sebagai simulasi. Salah satu *testnet* yang saat ini digunakan secara luas adalah *sepolia*, yang memungkinkan pengembang untuk menguji transaksi, kontrak pintar, serta aplikasi terdesentralisasi Sepolia menyediakan *ethereum* versi simulasi yang bebas dari nilai finansial, sehingga sangat efektif digunakan dalam tahap uji coba. Hal ini memberi kesempatan bagi pengembang untuk evaluasi sistem secara lebih leluasa.

6. *Escrow*

Escrow adalah sistem yang melibatkan pihak ketiga untuk menyimpan sementara uang atau aset lainnya selama proses transaksi. Tujuan utama *escrow* memastikan bahwa kedua belah pihak, pembeli dan penjual, memenuhi kewajiban mereka sebelum transaksi selesai (Kusumawati & Aulawi, 2021). Sistem ini sering digunakan dalam transaksi *e-commerce*, terutama ketika kepercayaan antara pembeli dan penjual rendah atau ada risiko penipuan. Transfer antarbank menuntut kepercayaan tinggi dari pembeli karena dana dikirim sebelum barang diterima sebagaimana praktik pembayaran di muka pada jual beli online yang umum menggunakan transfer antarbank sehingga *non-delivery fraud* kerap terjadi (Supianti, 2020). Untuk mengurangi risiko ini, diperlukan mekanisme *escrow* yang menjaga dana hingga bukti penyerahan valid.

Cara kerja *escrow* dimulai dengan pembeli melakukan pembayaran ke *escrow* yang dikelola oleh pihak ketiga, bukan langsung ke penjual. Setelah itu, penjual mengirimkan produk atau melakukan layanan sesuai dengan kesepakatan yang telah dibuat. Begitu pembeli menerima produk atau layanan dan memeriksa bahwa semua persyaratan telah dipenuhi, perantara *escrow* akan melepaskan dana kepada penjual. Jika ada perselisihan, perantara *escrow* akan bertindak sebagai pihak netral yang memutuskan apakah dana akan dikembalikan kepada pembeli atau diberikan kepada penjual sesuai dengan perjanjian yang telah disepakati.



Gambar 2. 2 Tahapan proses *escrow*

Sumber: www. quietlight.com

7. Dompet kripto

Dompet kripto merupakan aplikasi atau alat yang digunakan pengguna untuk menyimpan, mengirim, dan menerima mata uang kripto seperti *bitcoin* atau *ethereum*. Berbeda dengan dompet fisik biasanya, dompet kripto menyimpan kunci kriptografi yang memberikan akses ke alamat *blockchain* tempat aset digital disimpan secara aman.

Teknologi *blockchain* yang mendasari mata uang kripto dapat diakses melalui berbagai antarmuka pengguna. Fitur-fitur *blockchain* seperti transaksi terdesentralisasi, *smart contract*, dan pencatatan yang tidak dapat diubah dapat dimanfaatkan melalui aplikasi *mobile*, platform berbasis *web*, aplikasi desktop, hingga *hardware wallet*. Keberagaman akses ini memungkinkan pengguna berinteraksi dengan jaringan *blockchain* secara fleksibel sesuai dengan kebutuhan dan preferensi mereka (Rufus & Purba, 2024).

8. *Web3.js*

Web3.js adalah komponen *JavaScript* yang menghubungkan antarmuka *web* dengan aset *ethereum* dan rantai *EVM* lain melalui provider yang mengikuti standar, sehingga aplikasi dapat membaca data rantai blok, memanggil fungsi kontrak pintar

secara baca maupun tulis, memantau *event log*, serta melakukan konversi satuan antara *wei* dan *ether* (Negara, dkk, 2021).

Dalam penelitian ini, Web3.js digunakan pada sisi klien untuk mengaktifkan koneksi dompet, memvalidasi *tesnet sepolia* melalui pemeriksaan *chainId*, membentuk objek kontrak *escrow* dari alamat, mengeksekusi operasi transaksi seperti *deposit*, *release*, *refund*, dan *dispute* melalui dompet pengguna, serta berlangganan event kontrak agar status pada antarmuka selalu selaras dengan catatan *on-chain* yang dapat ditelusuri melalui *etherscan sepolia* (Baiquni & Dirgahayu, 2023).

9. *Ethers.js*

Ethers.js adalah komponen *JavaScript* yang dirancang ringan dan modular. Pola kerjanya jelas yaitu provider sebagai sumber data, *Signer* sebagai penanda tangan transaksi dari akun pengguna, dan kontrak sebagai abstraksi alamat beserta pemanggilan fungsi serta pembacaan *event*. Desain ini membuat integrasi dompet pada *dApp* konsisten, mudah diaudit, dan menjaga ukuran bundel antarmuka tetap kecil sehingga sesuai untuk aplikasi *web*.

Dalam penelitian ini, *Ethers.js* digunakan pada sisi klien untuk mengeksekusi operasi tulis pada kontrak *escrow* yang meliputi *deposit*, *release*, *refund*, dan *dispute* melalui Signer dari dompet pengguna, membentuk objek kontrak dari alamat e*scrow*, memvalidasi *sepolia* melalui pemeriksaan, menautkan *hash* transaksi agar dapat ditelusuri di *etherscan sepolia*, serta memproses *event* kontrak agar status pada antarmuka selalu selaras dengan catatan *on-chain* (Saian, dkk, 2024).

10. *Harhadt*

Hardhat merupakan *framework* pengembangan *ethereum* yang dirancang untuk memudahkan proses penulisan, pengujian, dan *deployment smart contract* (ResearchGate, 2022). Alat ini menyediakan lingkungan lokal *blockchain* yang memungkinkan peneliti maupun pengembang untuk melakukan simulasi transaksi secara cepat tanpa harus langsung terhubung dengan jaringan utama (Ejournal UNSRAT, 2025).

Salah satu keunggulan Hardhat adalah fleksibilitasnya dalam mendukung plugin yang mempermudah integrasi dengan berbagai *library*, seperti *ethers.js* dan *web3.js*. Dengan fitur *hardhat*, pengembang dapat menjalankan *node* lokal yang merekam setiap transaksi, sehingga proses debugging menjadi lebih transparan (EatTheBlocks, 2023).

Dalam konteks penelitian ini, *Hardhat* digunakan untuk menyimulasikan skenario transaksi jual beli akun *game online* berbasis *smart contract escrow*. Penggunaan *hardhat* memastikan bahwa setiap interaksi, dapat diuji secara terkontrol sebelum diimplementasikan pada jaringan uji *ethereum* (*testnet*). Dengan demikian, *hardhat* memberikan jaminan bahwa *smart contract* yang dibangun telah melalui tahap validasi teknis dan fungsional sebelum diuji pada lingkungan yang lebih luas.

11. *React*

React adalah komponen JavaScript untuk membangun antarmuka pengguna pengguna secara deklaratif melalui unit komponen yang dapat digunakan ulang, dengan *JSX* untuk mendeskripsikan tampilan serta *hooks* seperti *state* dan *effect* untuk mengelola data dan interaktivitas pada tingkat komponen sehingga

pembaruan layar efisien, terukur. Adapun pendekatan berbasis komponen dan arsitektur sehingga mempermudah pemisahan logika presentasi, meningkatkan produktivitas tim, dan menjaga konsistensi antarmuka pada aplikasi *web* (Kroons & Dewi, 2023).

Dalam penelitian ini *react* digunakan untuk membangun antarmuka *escrow* di sisi klien yang mencakup halaman koneksi dompet, formulir transaksi *deposit* dan penyelesaian sengketa, tampilan status transaksi yang diperbarui secara reaktif dari *event* kontrak, serta penautan hash transaksi ke *etherscan sepolia*. Praktik ini selaras dengan panduan integrasi *dApp* berbasis *react* dan dompet kripto serta sejalan dengan temuan studi Indonesia yang menunjukkan adopsi *React* pada pengembangan aplikasi *web* (Suroyo, dkk, 2024)

12. *Game online*

Game online telah berkembang menjadi fenomena *global* yang menyatukan jutaan pemain melalui internet. Baik dimainkan sendirian maupun bersama teman dari berbagai belahan dunia, *game online* menawarkan pengalaman bermain yang seru dan interaktif. Data terbaru menunjukkan bahwa industri ini terus tumbuh pesat di berbagai platform, mulai dari komputer, konsol *game*, hingga *smartphone* (Mulachela, 2020). Daya tarik utamanya terletak pada kemampuannya menciptakan dunia virtual tempat pemain bisa berkompetisi, bekerja sama, dan membangun komunitas secara nyata.

Salah satu perkembangan menarik dalam dunia *game online* adalah munculnya pasar virtual dimana pemain bisa melakukan penjualan aset digital. Mulai dari penjualan akun *game* dan item dalam *game* memiliki nilai ekonomi nyata. Namun sayangnya, menurut penelitian (Putra, 2021), aktivitas jual beli ini

seringkali ditemukan masalah seperti ketidakpercayaan antara penjual dengan pembeli. Oleh karena itu, dibutuhkan platform perantara (*escrow*) berbasis *blockchain* yang dapat menjamin transparansi dan keamanan transaksi

Game mobile adalah permainan digital yang dirancang khusus untuk dijalankan pada perangkat bergerak seperti *smartphone* dan tablet. Keunggulan utama *game mobile* terletak pada aksesibilitas, portabilitas, dan interaktivitas, sehingga dapat dimainkan kapan saja dan di mana saja. Industri *game mobile* berkembang pesat seiring meningkatnya penetrasi internet dan penggunaan *smartphone,* menjadikannya salah satu sektor terbesar dalam industri hiburan digital (Newzoo, 2024).

Adapun *game mobile* populer saat ini jadi salah satu pengguna terbanyak di unduh aplikasi *Playstore* maupun *Appstore*:

a. *Mobile Legends: Bang Bang*



Gambar 2. 3 *Logo game mobile legends: bang bang*

Mobile Legends: Bang Bang telah diunduh ratusan juta kali di *google play* sebanyak 500 juta. *Game* ini bergenre *Multiplayer Online Battle Arena (MOBA)* yang menekankan strategi tim 5 *vs* 5 untuk menghancurkan *base* lawan. Popularitasnya sangat tinggi di Asia Tenggara karena durasi permainan singkat dan ekosistem *e-sports* yang besar.

b. *Free Fire*



Gambar 2. 4 *Logo game free fire*

Free Fire dikembangkan oleh Garena dan dirilis tahun 2017. *Game* bergenre *battle royale* ini memungkinkan 50 pemain bertarung dalam satu arena dengan durasi sekitar 10 menit. Keunggulannya adalah optimasi pada perangkat spesifikasi rendah, sehingga menjangkau basis pemain yang luas terutama di negara berkembang.

c. *PUBG Mobile*



Gambar 2. 5 *Logo game pubg mobile*

PUBG Mobile adalah adaptasi dari *PlayerUnknown's Battlegrounds* yang dikembangkan *Tencent*. *Game* ini juga bergenre *battle royale* dengan grafis realistis dan *gameplay* yang menekankan strategi serta keterampilan menembak. Sejak perilisannya, *PUBG Mobile* menjadi salah satu pelopor *genre battle royale* di perangkat mobile dengan komunitas global besar.

d. *Genshin Impact*



Gambar 2. 6 *Logo game genshin impact*

Genshin Impact merupakan *game Action RPG* dengan konsep dunia terbuka *(open world)* yang dikembangkan *miHoYo (HoYoverse). Game* ini dirilis tahun 2020 dengan keunggulan grafis berkualitas tinggi, sistem *gacha* karakter, dan mekanisme pertarungan berbasis elemen. *Genshin Impact* memperluas pengalaman *game mobile* dengan narasi mendalam serta eksplorasi dunia *virtual* yang imersif.

13. Arsitektur sistem

Arsitektur sistem Arsitektur sistem adalah model konseptual yang menetapkan struktur, perilaku, dan berbagai sudut pandang suatu sistem. Deskripsi arsitektur merupakan representasi formal yang disusun agar memudahkan penalaran terhadap struktur dan perilaku tersebut. Prinsip transformasi proses ini relevan untuk perancangan arsitektur sistem yang menekankan alur tugas yang jelas, keterlacakan, dan akuntabilitas landasan penting ketika merancang fitur-fitur transaksi yang sensitif terhadap kepercayaan (Hastuty, 2025). Sebuah arsitektur dapat mencakup beragam komponen dan subsistem yang saling berinteraksi untuk mewujudkan fungsi sistem secara utuh. Upaya memformalkan cara pendeskripsian ini melahirkan keluarga bahasa deskripsi arsitektur, dikenal sebagai *Architecture Description Language.*

Antarmuka dibangun dengan *react* untuk merender halaman secara deklaratif.

a. *Frontend*

Frontend mengelola koneksi dompet pengguna, membentuk objek kontrak, lalu mengeksekusi transaksi melalui dompet digital status UI disinkronkan dari event *on-chain* dan setiap transaksi ditautkan ke *Etherscan* untuk penelusuran. Implementasi memanfaatkan Web3.js dan Ethers.js di sisi klien agar pembacaan data rantai, pemanggilan fungsi kontrak, dan pemantauan event konsisten pada *dApp* berbasis web.

b. *Backend*

Layanan *backend* bertugas menyediakan API aplikasi web, otentikasi, serta operasi bisnis di luar *blockchain*. Server dibangun dengan Node.js dan Express.js, sedangkan data aplikasi disimpan pada database MySQL untuk keperluan pencatatan dan pelaporan yang tidak harus berada di rantai.

c. *Blockchain* (*On-Chain Layer*)

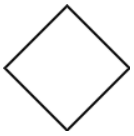
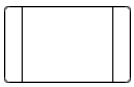
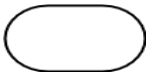
Logika inti *escrow* diimplementasikan sebagai *smart contract* pada Ethereum testnet Sepolia. Kontrak menangani penahanan dana, pelepasan ke penjual, pengembalian ke pembeli, serta penyelesaian seluruh jejak transaksi bersifat transparan dan dapat diverifikasi publik melalui Etherscan, sehingga meningkatkan kepercayaan dan menurunkan risiko kecurangan.

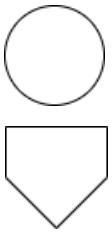
14. *Flowchart*

Menurut Sutanti, dkk (2020) *Flowchart* adalah representasi grafis yang menggambarkan langkah-langkah dan urutan prosedur suatu program. *Flowchart* membantu analis dan programmer memecah masalah menjadi segmen-segmen yang lebih kecil, serta mempermudah analisis terhadap alternatif solusi dalam proses operasional.

Flowchart berbentuk diagram atau gambar dengan aliran satu arah atau dua arah secara berurutan. Diagram ini digunakan untuk merepresentasikan atau mendesain program, membantu menyusun dan mendokumentasikan setiap langkah dalam suatu alur kerja.

Tabel 2. 1 *Symbol Use case diagram*

GAMBAR	NAMA	KETERANGAN
	Simbol input	Simbol jajargenjang, yang menyatakan proses input atau output tanpa tergantung.
	Simbol proses	Simbol proses, digambarkan dengan bentuk persegi panjang, digunakan untuk yang menyatakan suatu proses.
	Simbol Output	Simbol Output, digambarkan dengan bentuk jajar genjang terbalik, menggambarkan hasil yang menyatakan peralatan output yang digunakan.
	Simbol Percabangan	Simbol Percabangan, bentuk belah ketupat, menunjukkan kondisi tertentu menghasilkan dua kemungkinan jawaban, yaitu ya dan tidak.
	Simbol Prosedur	Simbol Prosedur, menggambarkan untuk pelaksanaan suatu bagian (sub-program) atau prosedur.
	Simbol Garis Air	Simbol Garis Air, digambarkan dengan anak panah digunakan untuk menggabungkan antara simbol yang satu dengan simbol yang lain.
	Simbol Terminator	Simbol Terminator, bentuk oval, yang menyatakan awal atau akhir suatu program.

	Simbol Konektor	Simbol Konektor: Bentuk lingkaran, digunakan untuk menghubungkan langkah-langkah dalam *Flowchart*, baik pada halaman yang sama maupun halaman yang berbeda.

15. *UML (Unified Modeling Language)*

UML (Unified Modeling Language) adalah bahasa pemodelan untuk sistem atau perangkat lunak yang berparadigma berorientasi objek. UML memiliki tujuan penggunaan yaitu memberikan bahasa pemodelan yang bebas dari berbagai bahasa pemrograman dan proses rekayasa, menyatukan praktek-praktek terbaik yang terdapat dalam pemodelan, dan memberikan model yang siap pakai, bahsa pemodelan visual yang ekspresif untuk mengembangkan dan saling menukar model dengan mudah dan dimengerti secara umum.

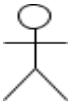
Pemodelan menggunakan UML bermanfaat untuk menggambarkan atau merancang sistem yang akan menyajikan informasi lebih rinci dalam proses perancangan suatu program. UML (Unified Modelling Language), sebagai standar industri, digunakan untuk menggambarkan, merencanakan, dan mendokumentasikan sistem perangkat lunak. UML menyediakan format standar untuk merancang model sistem.

UML mulai dikembangkan oleh Object Management Group sejak versi 1.0 dirilis pada bulan Januari 1997. Dalam rancang bangun berbasis objek, terdapat beberapa prinsip kunci yang harus dipahami, seperti Objek, Kelas, Abstraksi, Enkapsulasi, Pewarisan, dan Polimorfisme. Adapun jenis jenis UML yaitu :

a. *Use case*

Use case berfungsi untuk mendeskripsikan sebuah interaksi antara satu atau lebih actor dengan system informasi yang akan dibangun, dan dapat menggambarkan fungsi apa saja yang ada pada sebuah system informasi (Julianto & Setiawan, 2019).

Tabel 2. 2 *Symbol Use case diagram*

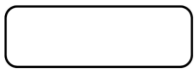
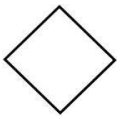
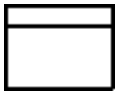
GAMBAR	NAMA	KETERANGAN
	*Actor*	Menspesifikasikan himpuan peran yang pengguna mainkan ketika berinteraksi dengan *Use case*.
	Dependency	Hubungan dimana perubahan yang terjadi pada suatu elemen mandiri (*independent*) akan mempengaruhi elemen yang bergantung padanya elemen yang tidak mandiri (*independent*).
	Generalization	Menspesifikasikan bahwa *use case* sumber secara eksplisit.
	Include	Menspesifikasikan bahwa *Use case* target memperluas perilaku dari *Use case* sumber pada suatu titik yang diberikan.
	Extend	Apa yang menghubungkan antara objek satu dengan objek lainnya.
	Association	Menspesifikasikan paket yang menampilkan sistem secara terbatas.

b. *Activity diagram*

Activity diagram menggambarkan workflow (aliran kerja) atau aktivitas dari sebuah system yang ada pada perangkat lunak.

Tabel 2. 3 *Symbol Activity diagram*

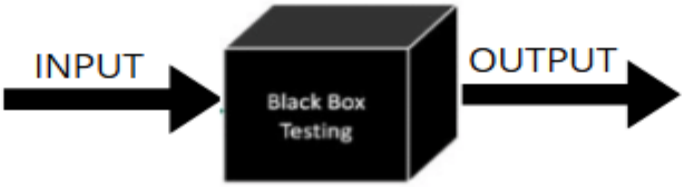
GAMBAR	NAMA	KETERANGAN
	*(initial Node) Status Awal*	Sebuah diagram aktivitas memiliki sebuah status awal.

GAMBAR	NAMA	KETERANGAN
	*Activity state*	Aktivitas yang dilakukan sistem, aktivitas biasanya diawali dengan kata kerja.
	Control Flow	Menjelaskan aliran kontrol dari satu tindakan ke tindakan lain.
	*Marge/Decision*	Percabangan dimana ada pilihan aktivitas yang lebih dari satu.
	*Final Node (Status Akhir)*	Status akhir yang dilakukan sistem, sebuah diagram aktivitas memiliki sebuah status akhir.
	Fork Node/Join Node	Pergabungan dimana leboh dari satu aktivitas lalu digabungkan jdi satu.
	*Swimlane*	Sebuah organisasi bisnis yang bertanggung terhadap aktivitas yang terjadi.

16. *Blackbox Testing*

Metode pengujian *Blackbox* adalah teknik pengujian yang berfokus pada spesifikasi fungsional perangkat lunak. Penguji dapat menentukan kumpulan kondisi input dan melakukan tes pada spesifikasi fungsional program. Uji *Blackbox* ini lebih fokus pada fungsionalitas sistem. Implementasi memiliki kelebihan dan kekurangan pengujian *Blackbox*. Salah satu keuntungannya adalah membantu Anda menemukan aspek-aspek yang tidak terpenuhi dari spesifikasi kebutuhan pengembangan perangkat lunak. Untuk kekurangan Blackbox, tes ini tidak sepenuhnya dilakukan dengan sejumlah terbatas penguji sehubungan dengan perangkat lunak yang diuji. (Fedianto, dkk, 2024). Pengujian Blackbox adalah metode pengujian perangkat lunak atau aplikasi yang berfokus pada fungsionalitas perspektif pengguna tanpa memeriksa struktur internal atau desain sistem. Sederhananya.

Pengujian Blackbox menilai apakah sistem dapat membuat edisi yang sesuai berdasarkan input yang ditentukan. Jika output cocok dengan ekspektasi, sistem dipertimbangkan dengan tepat. Analisis kebutuhan, perancangan terstruktur, dan pengujian fungsional (*black-box*) mampu meningkatkan efisiensi pencatatan, mengurangi kesalahan, dan memperbaiki akses informasi. Temuan ini menguatkan pemilihan arsitektur web-based pada sistem yang menuntut keandalan dan audit trail (Nasir, dkk, 2025). Namun, jika Langkah-langkah dilakukan, mereka perlu ditingkatkan. Metode ini juga dikenal sebagai tes perilaku, tes kotak tertutup, atau tes yang ditentukan.



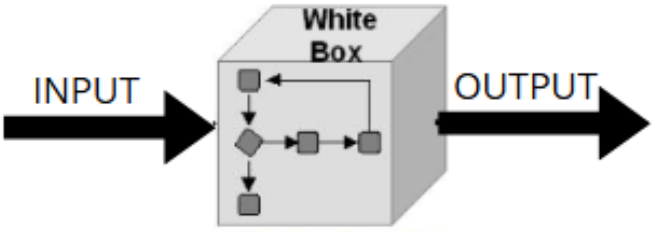
Gambar 2. 7 *Black box testing*

17. *Whitebox Testing*

Whitebox testing adalah teknik pengujian perangkat lunak yang menitikberatkan pada komponen internal, seperti desain, struktur kode, dan cara kerja perangkat lunak dari awal hingga akhir. Pengujian ini bertujuan untuk menganalisis alur data dari input ke *output*, menilai fungsionalitas, serta memastikan keamanan perangkat lunak. Dalam *white box testing*, penguji memiliki akses penuh untuk melihat dan menganalisis kode yang digunakan dalam pengembangan perangkat lunak. Oleh karena itu, metode ini juga dikenal dengan istilah *clear box testing*, *open box testing*, *code-based testing*, atau *structural testing*.

Whitebox testing dilakukan karena fokusnya pada logika internal dan struktur kode perangkat lunak. Uji kasus dirancang untuk menguji fungsi internal perangkat lunak dari sudut pandang pengembang. Teknik ini dilakukan ketika pengembang memiliki pemahaman menyeluruh tentang struktur program. Dengan menggunakan pendekatan ini, setiap cabang dan keputusan dalam program dapat diuj (Wintana & Nurhadi, 2022).

Whitebox *testing* memiliki beberapa kelebihan, di antaranya adalah kemampuannya untuk mengidentifikasi dan menghapus kode tersembunyi atau yang tidak diperlukan. Metode ini memungkinkan pengujian secara menyeluruh karena setiap bagian dari struktur atau logika program dieksplorasi secara detail. Selain itu, Whitebox *testing* membantu mengoptimalkan kode dan dapat dilakukan bahkan saat antarmuka pengguna grafis (GUI) masih dalam tahap pengembangan (Fedianto, dkk, 2024).



Gambar 2. 8 *White box testing*

18. *Functional Testing*

Functional testing merupakan metode pengujian perangkat lunak yang berfokus pada verifikasi fungsi sistem sesuai dengan spesifikasi yang telah ditentukan. Pengujian ini dilakukan untuk memastikan bahwa keluaran yang dihasilkan sudah sesuai dengan masukan yang diberikan, sehingga perangkat lunak dapat dipastikan berjalan sebagaimana mestinya. Dalam konteks aplikasi berbasis

blockchain, *functional testing* memiliki peran penting karena setiap tahapan transaksi, mulai dari pembuatan kontrak hingga penyelesaian transaksi, harus sesuai dengan alur yang dirancang agar sistem dapat diandalkan.

Menurut Staroletov dan Galkin (2023), *functional testing* pada aplikasi *blockchain* sebaiknya tidak hanya menilai fungsi tunggal, tetapi juga mencakup *integration testing*, yaitu pengujian antar modul yang saling berhubungan seperti dompet digital, *node blockchain*, dan antarmuka web. Pada penelitian ini, pengujian dilakukan dengan memanfaatkan jaringan uji (*testnet Ethereum*), *Etherscan*, serta integrasi dompet digital untuk memverifikasi alur transaksi secara menyeluruh sebelum sistem diterapkan secara nyata. Dengan cara ini, functional *testing* dapat menjamin bahwa sistem *blockchain* tidak hanya memenuhi kebutuhan fungsional dasar, tetapi juga mampu menjaga konsistensi integrasi antar komponen.

19. *Usability Testing*

Usability *Testing* adalah metode yang digunakan untuk menilai sejauh mana suatu perangkat lunak, situs web, atau produk dapat digunakan dengan efektif. Kemudahan akses ke situs web sangat penting, karena jika sebuah situs web sulit diakses atau digunakan, maka situs tersebut menjadi tidak layak untuk digunakan, karena tidak dapat memberikan informasi yang dibutuhkan oleh penggunanya.

Usability *Testing* melibatkan penentuan evaluator, pengumpulan data melalui survei dan kuisioner, pembuatan tugas-tugas usability, serta penyusunan skrip untuk pengujian usability. Pelaksanaan usability *testing* ini dilakukan dengan mempertimbangkan aspek *learnability, efficiency, memorability,* dan *satisfaction*.

Usability dapat diukur berdasarkan indikator-indikator berikut

a. Kemudahan (*learnability*) didefinisikan seberapa cepat pengguna mahir dalam menggunakan sistem serta kemudahan dalam penggunaan menjalankan suatu fungsi serta apa yang pengguna inginkan dapat meraka dapatkan.
b. Efisiensi (*efficiency*) didefenisikan sebagai sumber daya yang dikeluarkan guna mencapai ketepatan dan kelengkapan tujuan.
c. Mudah diingat (*memorability*) didefinisikan bagaimana kemapuan pengguna mempertahankan pengetahuannya setelah jangka waktu tertentu, kemampuan mengngat didapatkan dari peletakkan menu yang selalu tetap.
d. Kepuasan (*satisfaction*) didefinisikan kebebasan dari ketidaknyamanan, dan sikap positif terhadap penggunaan produk atau ukuran subjektif sebagaimana pengguna merasa tentang penggunaan sistem.

B. Kajian Hasil Penelitian Terdahulu

Penelitian sebelumnya sangat penting untuk digunakan sebagai acuan, pertimbangan, dan data pendukung dalam melaksanakan penelitian. Berikut ini adalah beberapa studi terdahulu yang telah dilakukan:

Tabel 2. 4 Kajian Hasil Penelitian terdahulu

No	Penelitian	Fokus Penelitian	Metode	Hasil
1	**Untung Rahardja, Qurotul Aini, M. Yusup, Aulia Edliyanti (2020)** *Penerapan Teknologi*	Penggunaan *blockchain* sebagai sistem keamanan dalam *e-*	Analisis *SWOT* dan implementasi sistem	*Blockchain* meningkatkan keamanan transaksi, memungkinkan pendistribusian

	Blockchain sebagai Media Pengamanan Proses Transaksi E-Commerce	*commerce* Pandawan		data secara transparan, serta mengurangi risiko kecurangan oleh pihak.
2	**Yoga Dwi Nugroho & I Made Suartana (2024)** *Penerapan Teknologi Blockchain pada Sistem Transaksi Aplikasi Point of Sale Berbasis Web*	Implementasi *Smart contract* berbasis *ethereum* dan metode konsensus *PoS* dalam aplikasi *POS*	Simulasi sistem & pengujian (*Smart contract* & *fungsional*)	Sistem berjalan dengan baik, mendukung transparansi dan keamanan transaksi, serta berhasil diuji dengan integrasi *React*.js dan *Web3.js*
3	**Annisa Lisdayanti & Oliver Hasan Padmanegara (2024)** *Peran Teknologi Blockchain dalam Meningkatkan Kepercayaan Konsumen dan Keamanan Data Privasi pada E-Commerce*	Peran *blockchain* dalam meningkatkan kepercayaan konsumen dan keamanan privasi data *e-commerce*	Tinjauan literatur sistematis (2017–2023)	*Blockchain* mampu meningkatkan transparansi dan keamanan data, namun masih menghadapi kendala teknis dan regulasi dalam implementasinya

Berdasarkan kajian terhadap penelitian terdahulu, penelitian ini memiliki persamaan dengan ketiga penelitian sebelumnya, yaitu sama-sama berfokus pada pemanfaatan teknologi *blockchain* untuk meningkatkan transparansi transaksi. Ketiganya juga menekankan pentingnya sistem terdesentralisasi dalam pengelolaan data transaksi.

Namun, terdapat perbedaan pada objek dan konteks transaksi yang diangkat dalam penelitian ini. Fokus penelitian ini adalah pada jual beli akun *game online*, yang berbeda dari konteks *point of sale* dalam penelitian sebelumnya. Selain itu,

sistem yang dikembangkan dalam penelitian ini mensimulasikan fitur *escrow* berbasis *smart contract*, yang belum dibahas dalam penelitian *terdahulu*.

Adapun hasil yang diharapkan dari penelitian ini adalah tersusunnya simulasi sistem transaksi jual beli akun *game online* berbasis *blockchain* dengan *smart contract ethereum* yang dilengkapi dengan fitur *escrow*, sehingga sistem mampu menerapkan transaksi real dan transparan.

`

BAB III

METODE PENELITIAN

A. Jenis Penelitian

Penelitian ini menggunakan metode kualitatif dengan pendekatan simulasi sistem untuk menguji penerapan teknologi *blockchain* dalam transaksi jual beli akun *game*. Pendekatan ini dipilih karena memungkinkan untuk mengkaji dampak *blockchain* terhadap transparansi transaksi digital dalam kondisi yang terkendali. Penelitian ini bertujuan untuk menganalisis bagaimana *blockchain* dapat mengatasi masalah yang sering terjadi pada sistem transaksi konvensional, seperti potensi kecurangan dan kurangnya kepercayaan antara pembeli dan penjual.

B. Lokasi dan Waktu

Penelitian ini dilakukan selama 5 (lima) bulan dimulai april hingga agustus 2025. Seluruh kegiatan dilakukan dalam bentuk simulasi sistem untuk menguji performa *blockchain*.

Tabel 3. 1 Lokasi dan Waktu

No	Uraian Kegiatan	Bulan 2025				
		April	Mei	Juni	Juli	Agustus
1	Studi literatur					
2	Membangun desain *website*					
3	Implementasi *blockchain* pada *website*					
4	Pengujian dan evaluasi simulasi sistem					
5	Analisis hasil simulasi					

C. Alat dan Bahan

Penelitian untuk pengembangan aplikasi ini hanya dapat dilakukan jika tersedia *hardware* dan *software* yang mendukung. Berikut adalah penjelasan mengenai *hardware* dan *software* yang akan digunakan:

1. *Hardware*

   Hardware yang digunakan ditunjukan pada Tabel 3.2.

   Tabel 3. 2 *Hardware*

Spesifikasi	
Merek Laptop	Lenovo IdeaPad 3 14ABA7
Processor Laptop	AMD Ryzen 5 5625U with Radeon Graphics (12 CPU), ~ 2.30 GHz
RAM Laptop	24 GB

2. *Software*

   Software yang digunakan ditunjukan pada Tabel 3.3.

   Tabel 3. 3 *Software*

Spesifikasi	
Sistem Operasi	Windows 11
Tool Pemrograman	*Visual Studio Code*
Bahasa Pemrograman	*JavaScript*, *Solidity*

D. Teknik Pengumpulan Data

Teknik pengumpulan data yang digunakan bertujuan untuk memperoleh informasi yang relevan dan akurat dalam rangka membangun sistem simulasi transaksi jual beli akun *game* berbasis *blockchain*. Adapun teknik pengumpulan data yang digunakan dalam penelitian ini yaitu:

1. Studi Literatur

   Studi literatur dilakukan dengan mengumpulkan dan menganalisis buku, jurnal, artikel ilmiah, dan publikasi *online* yang berkaitan dengan teknologi *blockchain* dan transaksi jual beli akun *game online*. Peneliti akan menelaah berbagai sumber informasi untuk memahami konsep dasar *blockchain*, karakteristik *smart contract*, dan implementasinya dalam transaksi digital. Selain itu, studi ini akan mencakup analisis dokumentasi teknis dari platform *blockchain* yang umum digunakan untuk mengidentifikasi fitur yang sesuai dengan kebutuhan sistem simulasi.

2. Kuesioner

   Kuesioner digunakan untuk mengumpulkan data dari pelaku transaksi jual beli akun *game online*, baik pembeli maupun penjual. Tujuannya adalah mengidentifikasi masalah umum dalam transaksi, menilai tingkat kepercayaan pengguna, dan mengetahui harapan terhadap sistem berbasis *blockchain*. Kuesioner disusun secara semi-terstruktur dan dibagikan melalui formulir digital agar mudah diisi oleh responden. Pertanyaannya mencakup pengalaman bertransaksi, kepuasan, serta tanggapan terhadap fitur seperti *escrow* dan riwayat transaksi.

3. Pengembangan Sistem

   Pengembangan sistem dilakukan melalui tahapan analisis kebutuhan, perancangan, implementasi, dan pengujian. Berdasarkan data yang diperoleh dari studi literatur, peneliti akan mengidentifikasi fitur-fitur utama yang diperlukan dalam simulasi sistem *blockchain* untuk transaksi jual beli akun *game online*. Perancangan sistem akan mencakup pembuatan alur

transaksi, struktur *smart contract* sederhana, dan antarmuka pengguna yang intuitif. Implementasi akan menggunakan *framework blockchain* yang tersedia untuk membuat simulasi yang dapat mendemonstrasikan proses transaksi dari awal hingga akhir. Pengembangan sistem tidak hanya mencakup analisis kebutuhan, perancangan, implementasi, dan pengujian, tetapi juga melibatkan proses *deployment smart contract* ke jaringan *sepolia* uji (*testnet*). Tahapan deployment ini penting agar simulasi tidak hanya dijalankan secara lokal, tetapi juga diuji dalam lingkungan *blockchain* publik yang menyerupai kondisi nyata. Pengujian akan dilakukan untuk memastikan sistem berjalan sesuai dengan yang diharapkan dan mampu mengatasi permasalahan yang telah diidentifikasi sebelumnya.

E. Teknik Analisis Data

Dalam penelitian ini, peneliti menggunakan analisis deskriptif kualitatif untuk mengevaluasi proses simulasi sistem transaksi berbasis *blockchain*. Teknik ini digunakan untuk mengkaji bagaimana sistem bekerja dalam meniru alur transaksi jual beli akun *game online* secara aman dan transparan menggunakan *Smart contract*.

Data yang dianalisis berasal dari hasil simulasi sistem, termasuk data transaksi dan keluaran dari *smart contract*. Analisis dilakukan dengan mengamati kesesuaian antara rancangan sistem dan output aktual pada platform simulasi seperti dompet digital.

Proses analisis mencakup :

1. Menguji fungsi fitur *smart contract escrow*.

2. Evaluasi alur transaksi mulai dari tahap *smart contract escrow* hingga pelepasan dana.
3. Menganalisis riwayat transaksi pada *blockchain* untuk menilai transparansi.
4. Selain analisis deskriptif dari hasil simulasi, peneliti juga menganalisis *log* transaksi yang tercatat di *testnet etherscan*.

Hal ini memberikan bukti nyata transparansi transaksi, karena setiap interaksi *smart contract* dapat ditelusuri secara publik. Data yang diamati meliputi hash transaksi, alamat pengirim dan penerima, status *escrow*, serta waktu konfirmasi blok. Teknik ini dipilih karena mampu memberikan gambaran menyeluruh terhadap kinerja sistem dan efektivitas penerapan *blockchain* dalam konteks jual beli akun *game online*.

F. *Algoritma* Sistem

Algoritma merupakan representasi langkah-langkah logis yang menggambarkan alur proses transaksi pada sistem yang dibangun. *Algoritma* sistem ini didasarkan pada penerapan *smart contract escrow* di jaringan *ethereum* (*testnet Sepolia*) untuk menjamin transparansi, keadilan, dan keamanan dalam jual beli akun *game* online.

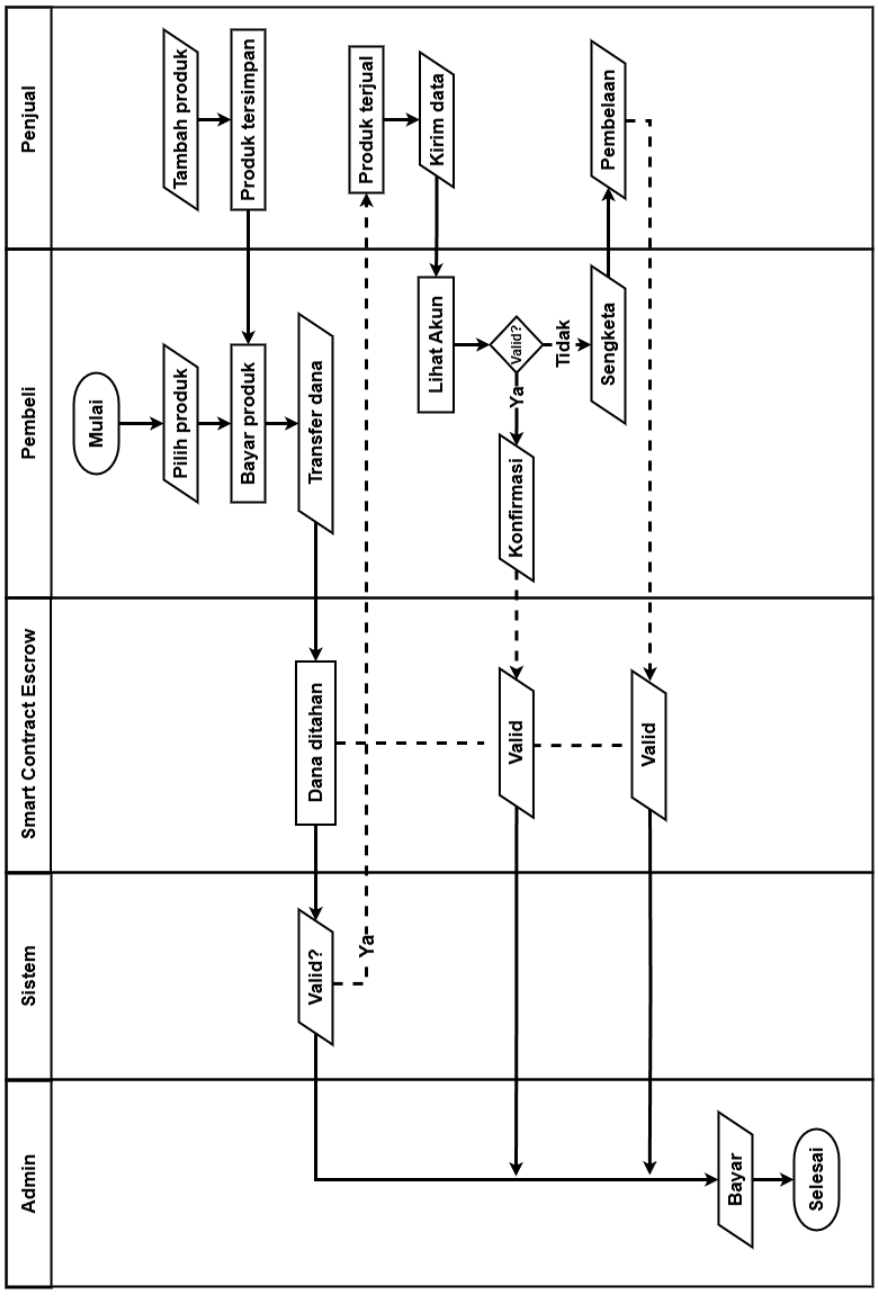
Secara umum, *algoritma* bekerja dengan menempatkan *escrow* sebagai pihak netral yang menahan dana hingga pembeli mengonfirmasi kesesuaian akun. Apabila terjadi perselisihan, admin bertugas sebagai arbitrator untuk memutuskan hasil sengketa berdasarkan bukti yang ada. Seluruh transaksi yang terjadi dicatat permanen pada *blockchain* dan dapat diverifikasi melalui *etherscan*, sehingga transparansi tetap terjaga.

Adapun langkah-langkah *algoritma* sistem dapat dijabarkan sebagai berikut:

1. Pembeli memilih produk yang ditawarkan oleh penjual melalui aplikasi.
2. Pembeli melakukan pembayaran (*deposit*) ke dalam *smart contract escrow*.
3. Blockchain mencatat transaksi secara permanen, yang dapat diverifikasi publik melalui *etherscan*.
4. Sistem memverifikasi validitas dana:
    a. Jika tidak *valid*, transaksi dibatalkan.
    b. Jika *valid*, lanjut ke langkah berikutnya.
5. Penjual mengirimkan data akun *game* kepada pembeli sesuai kesepakatan.
6. Pembeli melakukan pemeriksaan akun:
    a. Jika akun sesuai, pembeli memberikan konfirmasi, lalu admin memanggil fungsi melepaskan dana pada *smart contract*, dana dilepaskan ke penjual.
    b. Jika akun tidak sesuai, pembeli mengajukan sengketa.
7. Admin memverifikasi sengketa:
    a. Jika klaim pembeli *valid*, fungsi pengembalian dipanggil, dana dikembalikan ke pembeli.
    b. Jika klaim penjual *valid*, fungsi melepaskan dana dipanggil, dana dilepaskan ke penjual.
8. Transaksi selesai dan seluruh proses tercatat pada *blockchain* sebagai bukti permanen yang transparan dan dapat diaudit.

G. Diagram Alur

Flowchart

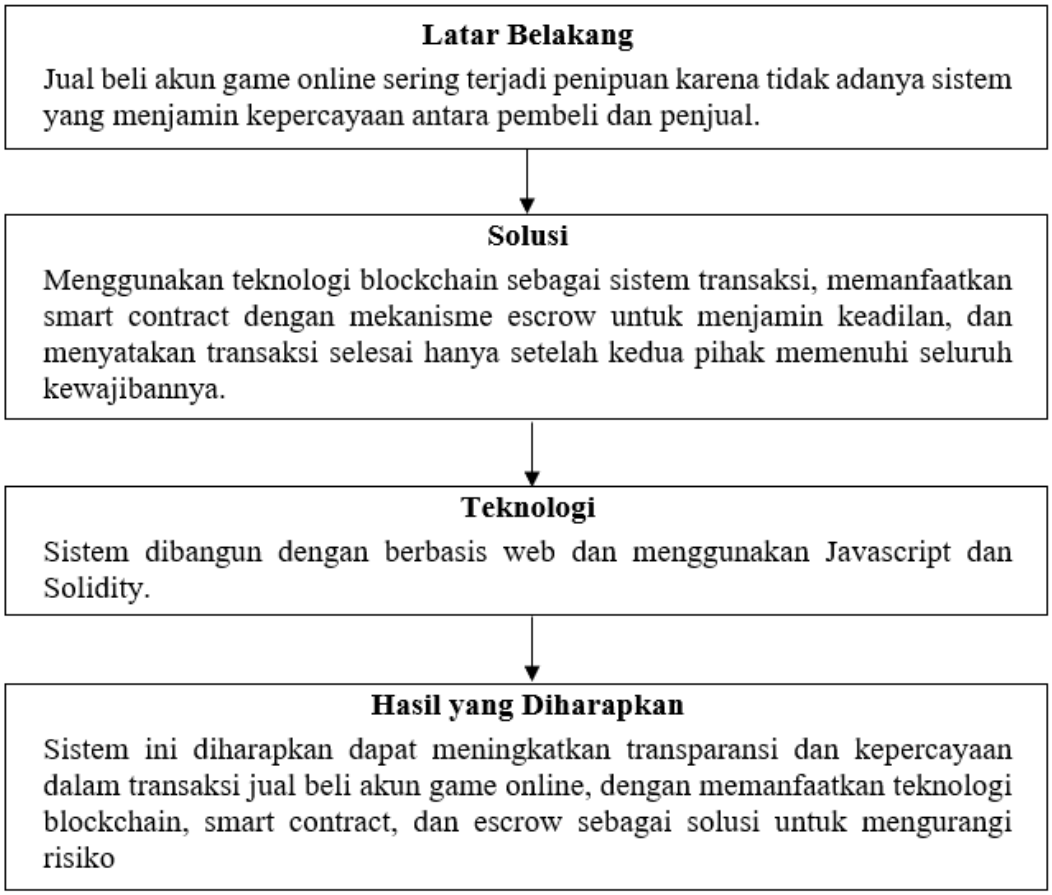


Gambar 3. 1 *Flowchart* transaksi

Proses transaksi jual beli akun *game online* berbasis *smart contract escrow* dimulai dari pembeli yang memilih produk yang ditawarkan penjual. Setelah

produk dipilih, pembeli melakukan pembayaran dengan mentransfer *ethereum* ke *smart contract escrow* sebagai pihak penengah, bukan langsung kepada penjual. *Smart contract* kemudian memvalidasi keabsahan dana yang masuk. Jika valid, status produk berubah menjadi terjual dan penjual mengirimkan data akun kepada pembeli. Selanjutnya, pembeli dapat memeriksa akun yang diterima dan memberikan konfirmasi. Apabila akun sesuai dengan kesepakatan, pembeli melakukan konfirmasi valid, lalu admin memanggil sistem untuk melepaskan dana dari *escrow* kepada penjual. Namun, jika akun yang diterima tidak sesuai, pembeli berhak mengajukan sengketa. Pada tahap ini, admin berperan sebagai mediator dengan memverifikasi bukti dari kedua belah pihak. Apabila pembelaan penjual terbukti benar, dana tetap dilepaskan kepada penjual, sedangkan jika klaim pembeli yang valid, dana akan dikembalikan kepada pembeli. Setelah keputusan akhir ditetapkan, sistem menutup proses transaksi dengan pelepasan dana kepada pihak yang berhak. Dengan alur tersebut, transparansi dan keamanan transaksi dapat terjaga melalui peran *smart contract* sebagai pengelola dana netral, serta admin sebagai pihak penentu dalam penyelesaian sengketa.

H. Kerangka Pikir



`

BAB IV

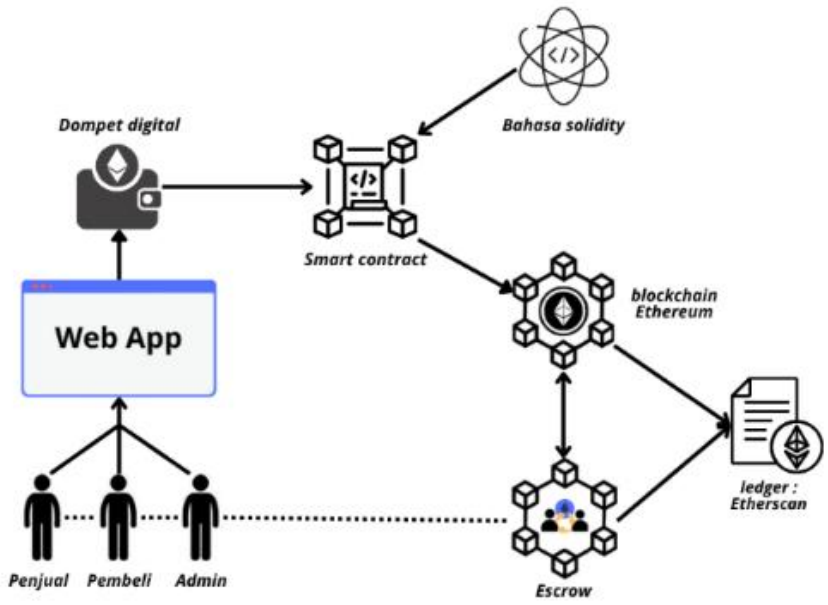
HASIL DAN PEMBAHASAN

A. Rancangan Sistem

Rancangan sistem adalah proses mendefinisikan arsitektur, komponen, modul, antarmuka, dan data dari sebuah sistem guna memenuhi kebutuhan tertentu. Proses perancangan ini mencakup berbagai aspek yang harus diperhatikan agar sistem yang dibangun dapat berjalan dengan efektif dan efisien. Salah satu aspek yang sangat penting dalam perancangan sistem adalah pemahaman yang mendalam terhadap kebutuhan pengguna, sehingga sistem yang dikembangkan dapat memberikan solusi yang relevan dan bermanfaat.

1. Arsitektur Rancangan Sistem

a. Arsitektur rancangan sistem *blockchain*



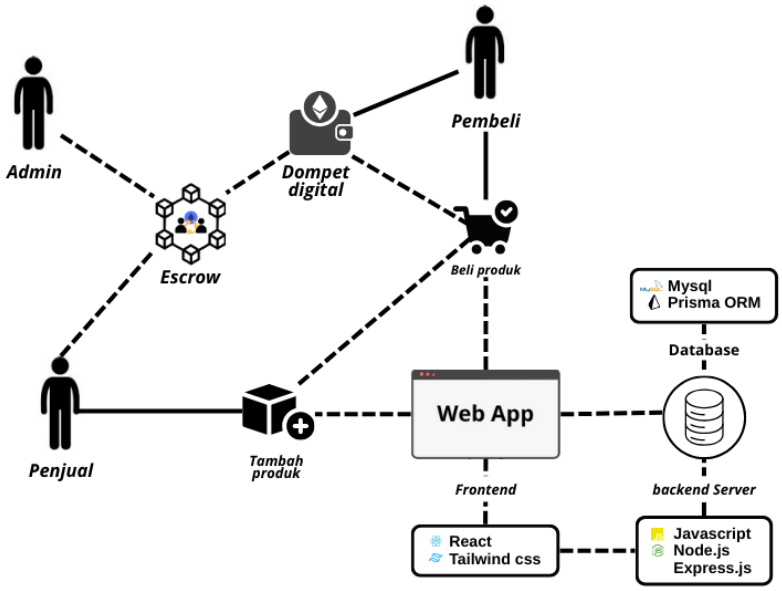
Gambar 4. 1 *Arsitektur* rancangan sistem *blockchain*

Gambar alur rancangan sistem ini menggambarkan bagaimana sistem akan diterapkan dalam *website* jual beli. Penjelasan lebih mendalam mengenai setiap komponen sistem disampaikan dalam bentuk tabel berikut, yang mencakup peran utama, fungsi, serta cara kerja sistem dalam menjamin transparansi, dan keandalan transaksi:

Tabel 4. 1 Alur rancangan sistem *blockchain*

Komponen	Deskripsi
Web App	Platform utama tempat interaksi antara pembeli, penjual, dan admin berlangsung. Pengguna dapat mengakses platform untuk melakukan transaksi digital.
Dompet Digital	Alat yang digunakan oleh pembeli dan penjual untuk melakukan transaksi dengan cryptocurrency.
Solidity	Bahasa pemrograman yang digunakan untuk menulis *smart contract* di *blockchain ethereum*.
Smart contract	Program yang dijalankan di *blockchain ethereum* untuk mengotomatiskan transaksi sesuai dengan ketentuan antara pembeli dan penjual.
Blockchain ethereum	Aset digital pembayaran yang mendukung transaksi terdesentralisasi, transparan, dan aman menggunakan *ethereum*.
Buku besar (*etherscan)*	Catatan untuk memverifikasi dan melacak transaksi yang terjadi di *blockchain ethereum*.
Escrow	Fitur pengamanan yang memastikan dana atau produk aman selama transaksi, hanya dilepaskan ketika semua syarat dipenuhi.
Penjual, Pembeli, Admin	Pengguna utama sistem: Penjual menjual produk, pembeli melakukan pembelian, admin mengelola dan memverifikasi transaksi

b. Arsitektur rancangan sistem aplikasi *website*



Gambar 4. 2 *Arsitektur* rancangan sistem aplikasi *website*

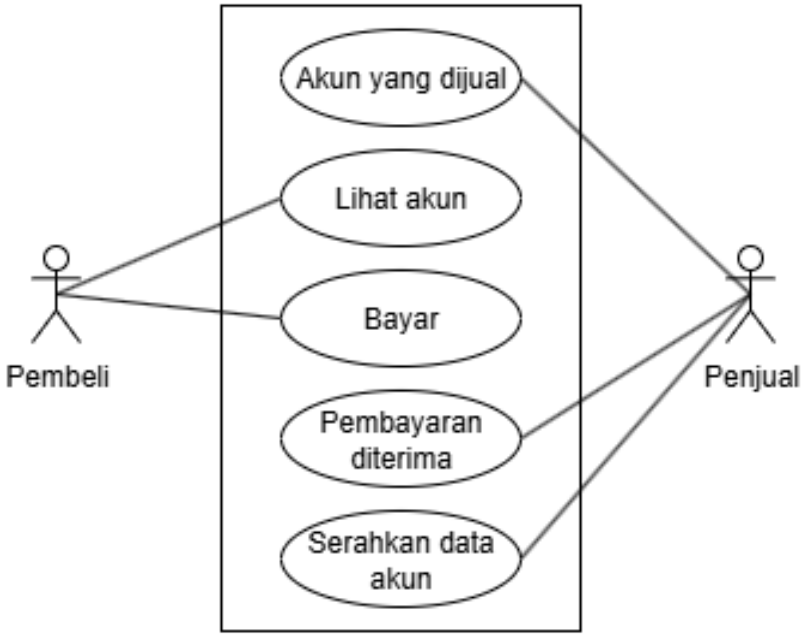
Arsitektur rancangan sistem ini menggambarkan bagaimana sistem akan diterapkan dalam *website* jual beli. Setiap komponen dalam sistem bekerja sama untuk menjamin kelancaran transaksi antara pembeli, penjual, dan pihak admin. Berikut adalah penjelasan lebih mendalam mengenai peran utama, fungsi, dan cara kerja sistem:

Tabel 4. 2 Alur rancangan sistem *website*

Komponen	Deskripsi
Admin	Administrator yang mengelola dan memonitor sistem transaksi.
Escrow	Sistem yang menyimpan dana sementara dalam transaksi hingga selesai.
Pembeli	Pengguna yang membeli produk melalui web app dan melakukan pembayaran.
Penjual	Penjual yang menawarkan produk dan mengelola penjualan melalui web app.

Web App	Aplikasi web yang memungkinkan interaksi pengguna dengan sistem.
Database	Tempat penyimpanan data transaksi, pengguna, produk, dan lainnya menggunakan MySQL.
Backend Server	Server yang menjalankan *backend* aplikasi, menggunakan Node.js dan Express.js.

2. *Use case* Diagram



Gambar 4. 3 *Use case* yang berjalan

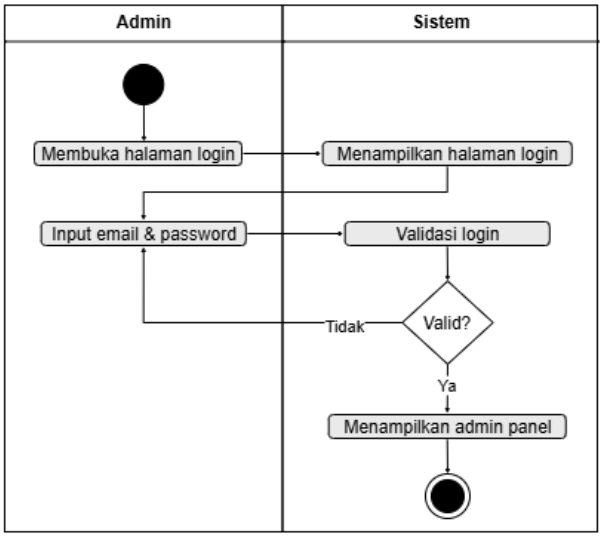
Dalam sistem berjalan sekarang dengan transaksi jual beli akun *game*, pembeli memilih akun yang dijual dan melakukan pembayaran sesuai kesepakatan. Setelah pembayaran diterima, penjual memberikan data akun kepada pembeli. Proses ini bergantung pada kepercayaan antara kedua pihak, dengan risiko ketidakpastian apakah pembayaran diterima atau data akun diberikan. Karena tidak ada sistem yang memastikan setiap langkah, transaksi ini berisiko jika salah satu pihak tidak memenuhi kewajibannya.

3. *Activity* Diagram

Activity diagram menggambarkan alur proses atau aktivitas yang terjadi dalam sistem perangkat lunak. Diagram ini berfungsi untuk memvisualisasikan urutan kerja (*workflow*) yang membantu memahami logika proses bisnis atau alur aktivitas sistem secara menyeluruh.

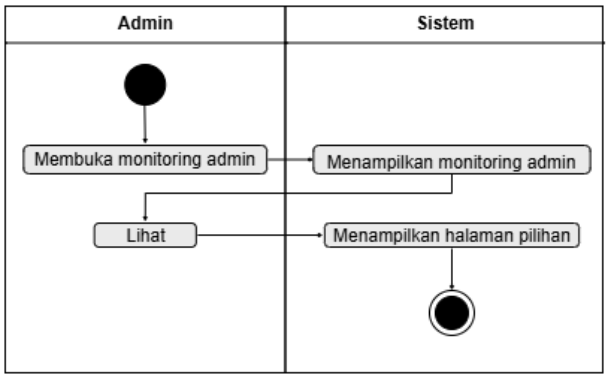
a. *Activity* diagram admin

1) *Activity* diagram *login*



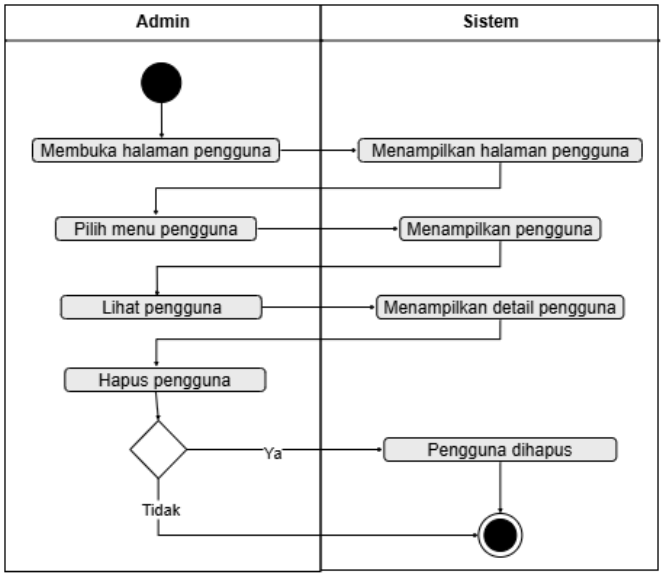
Gambar 4. 4 *Activity* diagram *login*

2) *Activity* diagram *monitoring* admin



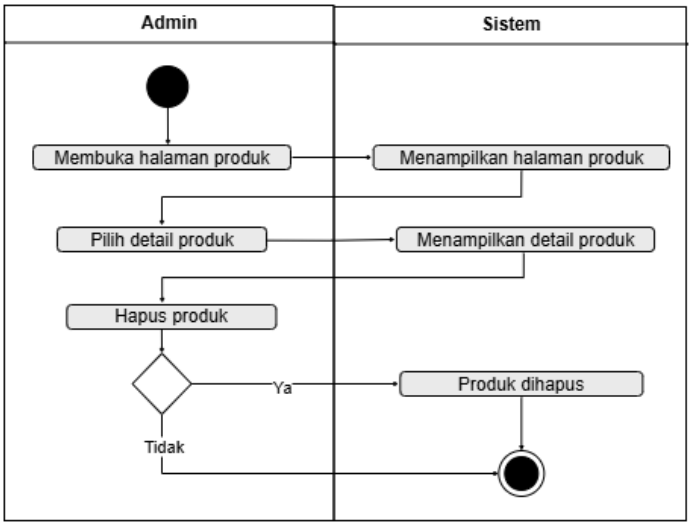
Gambar 4. 5 *Activity* diagram monitoring

3) *Activity* diagram hapus pengguna



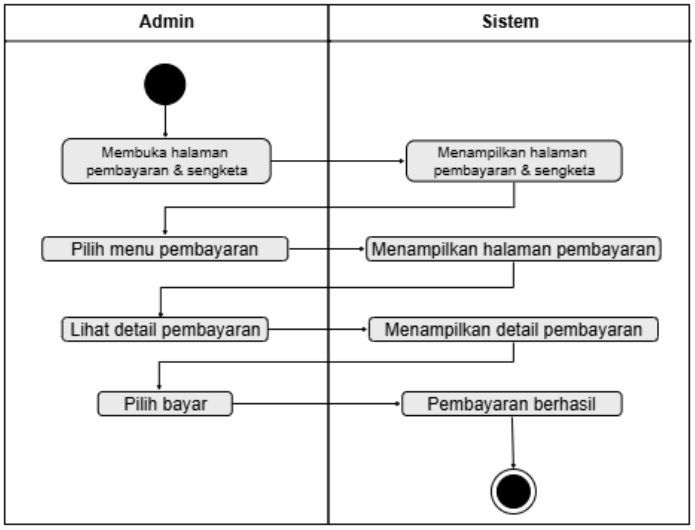
Gambar 4. 6 *Activity* diagram hapus pengguna

4) *Activity* diagram hapus produk



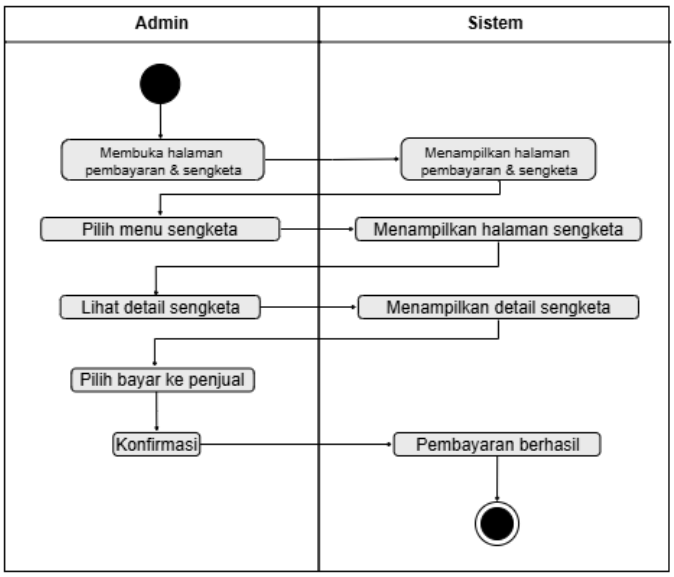
Gambar 4. 7 *Activity* diagram hapus produk

5) *Activity* diagram pembayaran



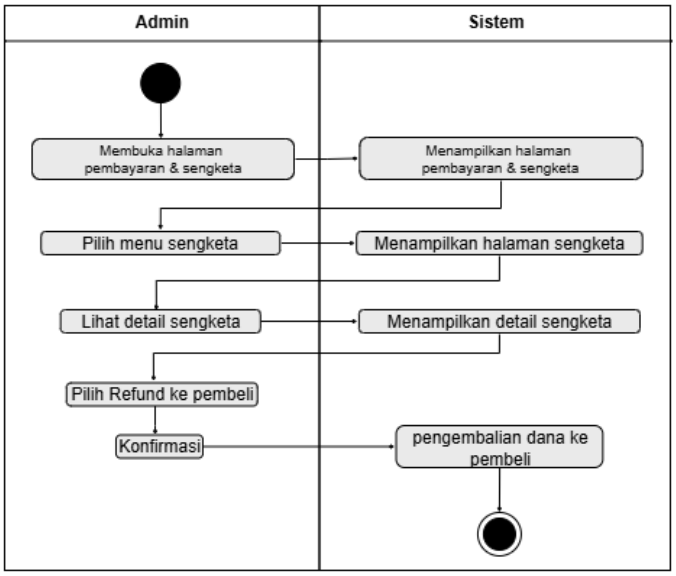
Gambar 4.8 *Activity* diagram pembayaran

6) *Activity* diagram sengketa bayar penjual



Gambar 4.9 *Activity* diagram sengketa bayar penjual

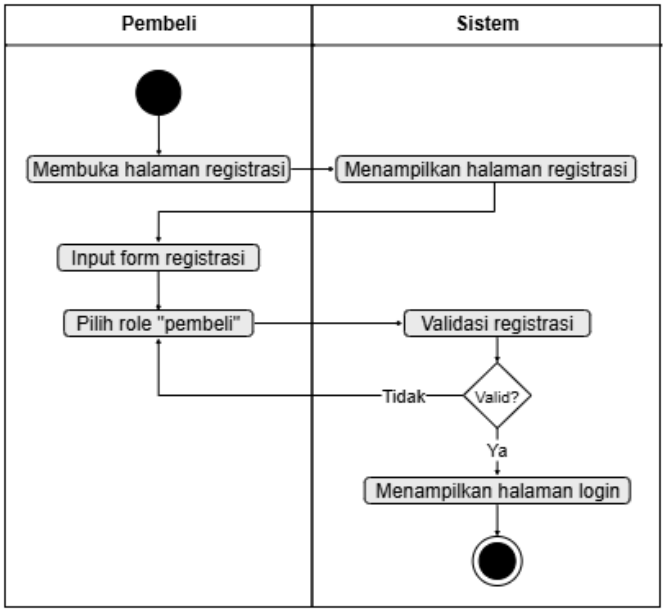
7) *Activity* diagram sengketa *refund* pembeli



Gambar 4. 10 *Activity* diagram sengketa *refund* pembeli

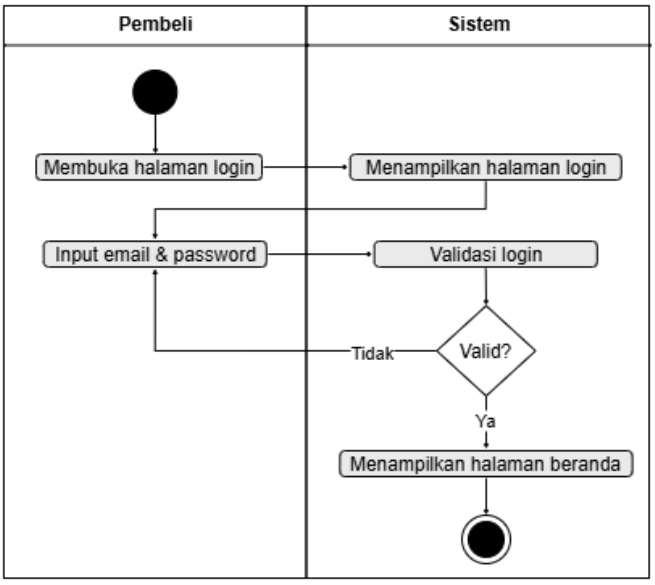
b. *Activity* diagram pembeli

1) *Activity* diagram registrasi



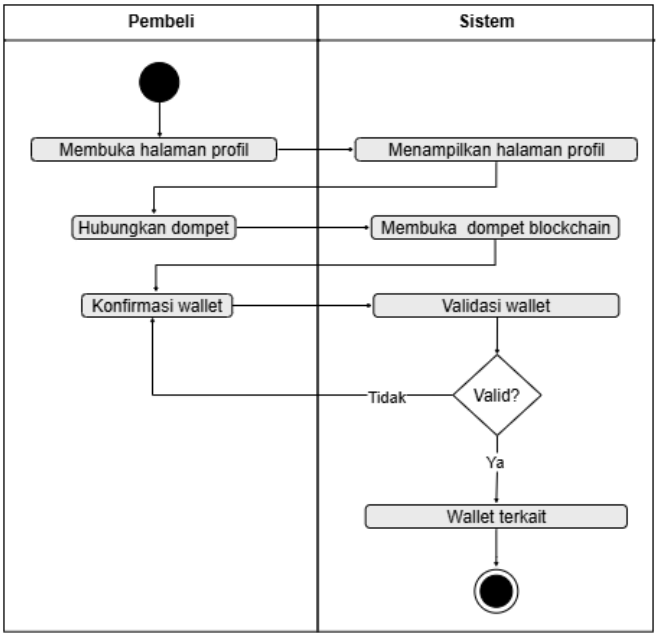
Gambar 4. 11 *Activity* diagram registrasi

2) *Activity* diagram *login*



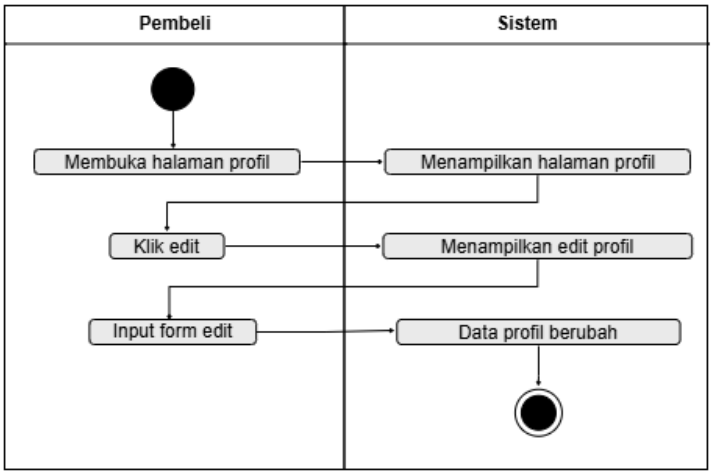
Gambar 4. 12 *Activity* diagram *login*

3) *Activity* diagram kaitkan hubungkan dompet



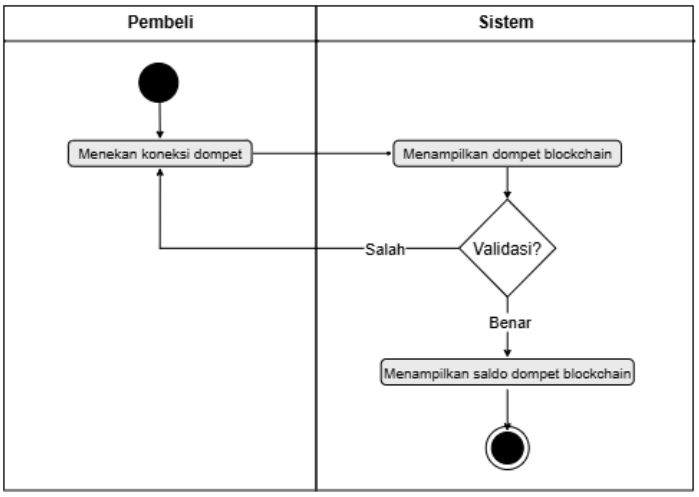
Gambar 4. 13 *Activity* diagram hubungkan dompet

4) *Activity* diagram edit profil



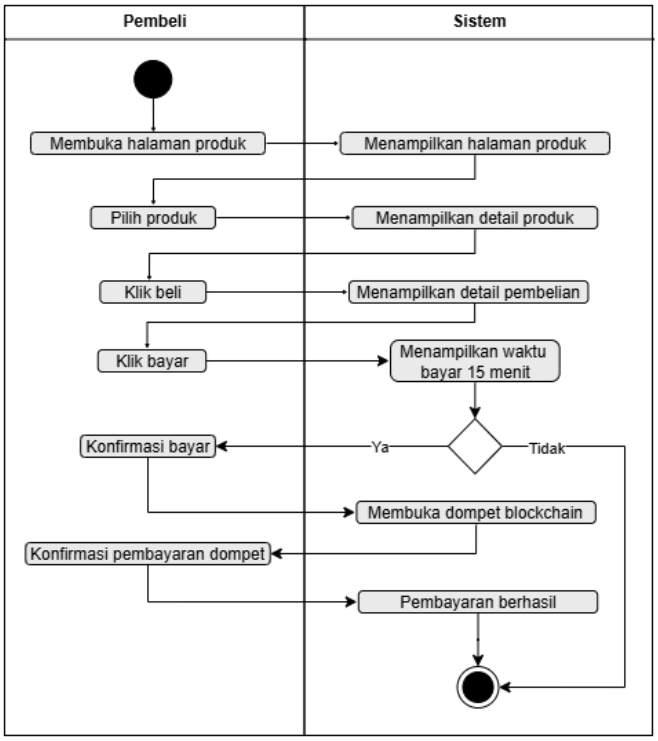
Gambar 4. 14 *Activity* diagram edit profil

5) *Activity* diagram koneksi hubungkan dompet



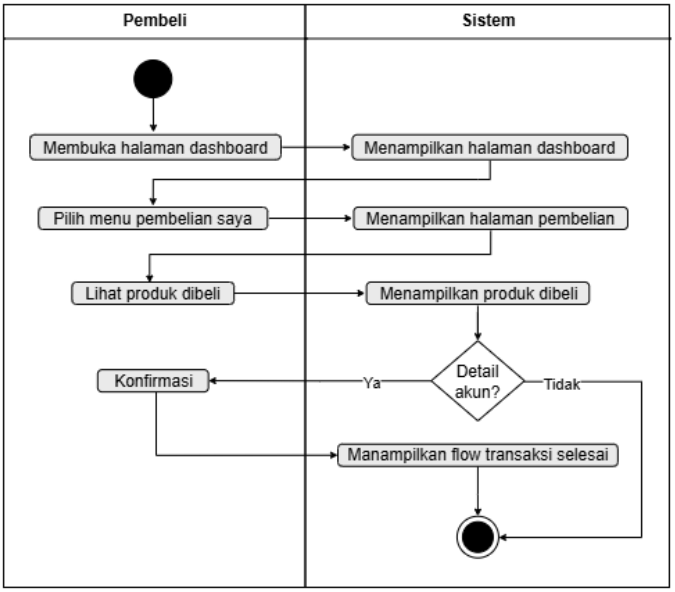
Gambar 4. 15 *Activity* diagram koneksi *hubungkan dompet*

6) *Activity* diagram beli produk



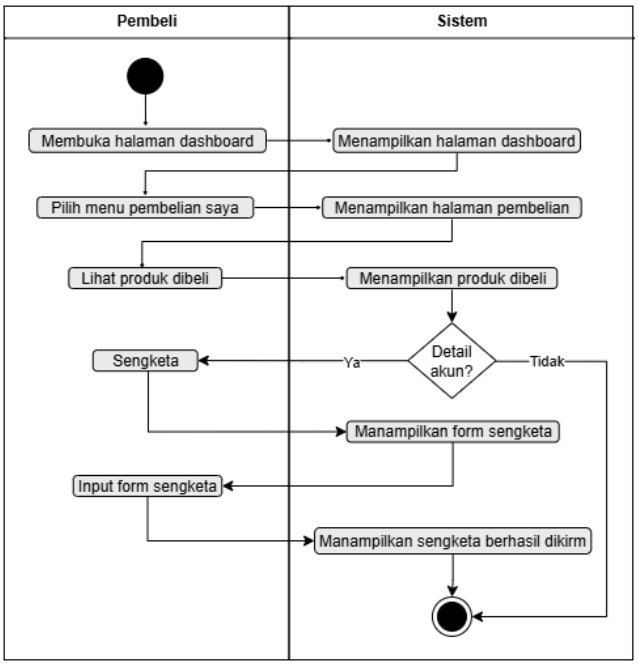
Gambar 4. 16 *Activity* diagram beli produk

7) *Activity* diagram produk sesuai



Gambar 4. 17 *Activity* diagram produk sesuai

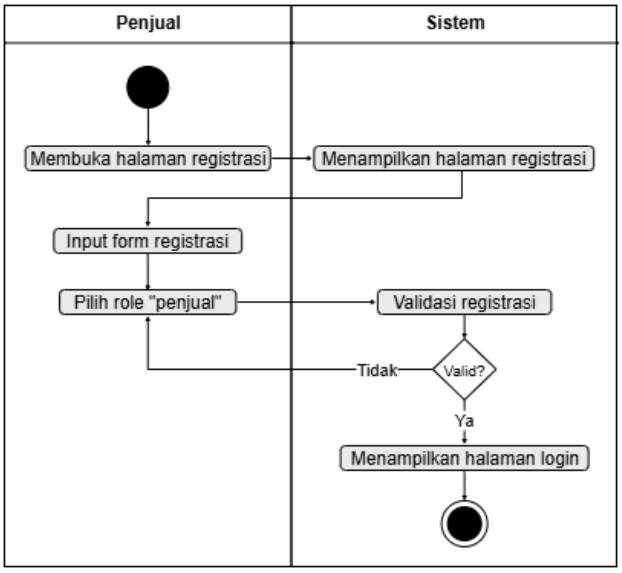
8) *Activity* diagram sengketa produk



Gambar 4. 18 *Activity* diagram sengketa produk

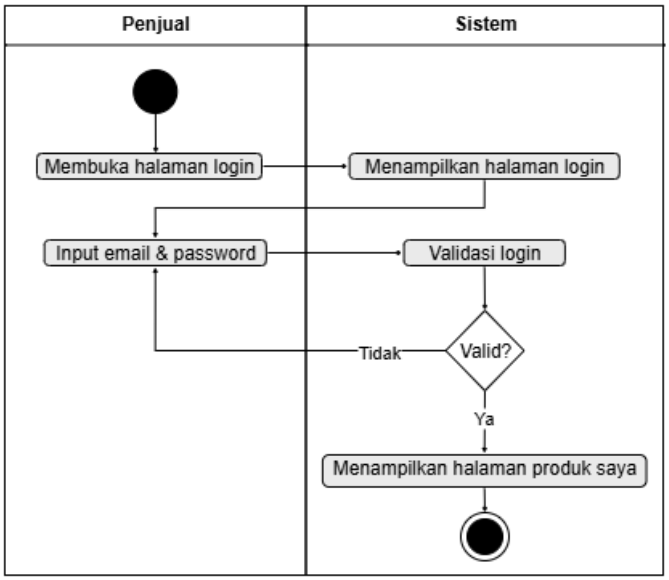
c. *Activity* diagram penjual

1) *Activity* diagram registrasi



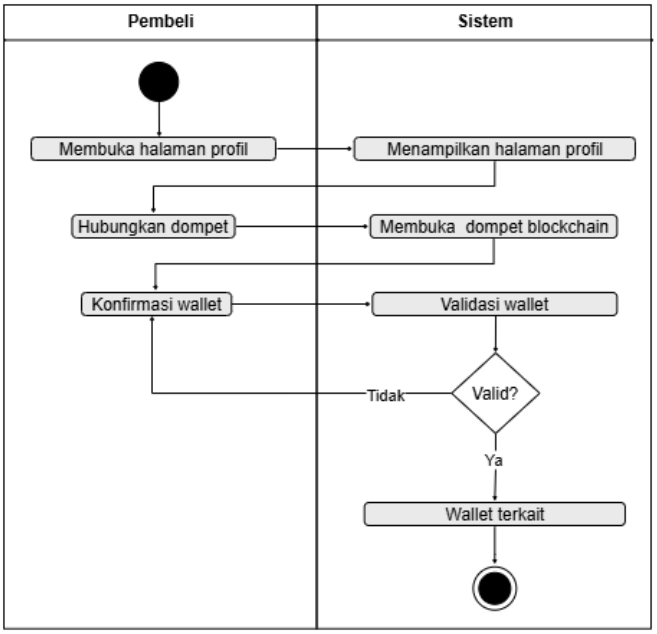
Gambar 4. 19 *Activity* diagram registrasi

2) *Activity* diagram *login*



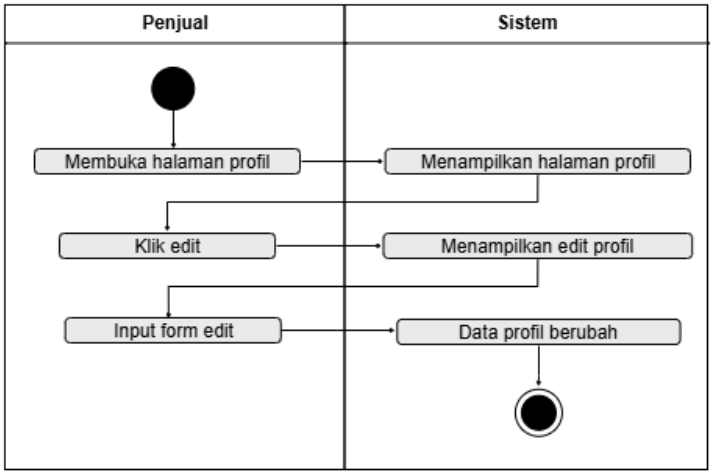
Gambar 4. 20 *Activity* diagram *login*

3) *Activity* diagram hubungkan hubungkan dompet



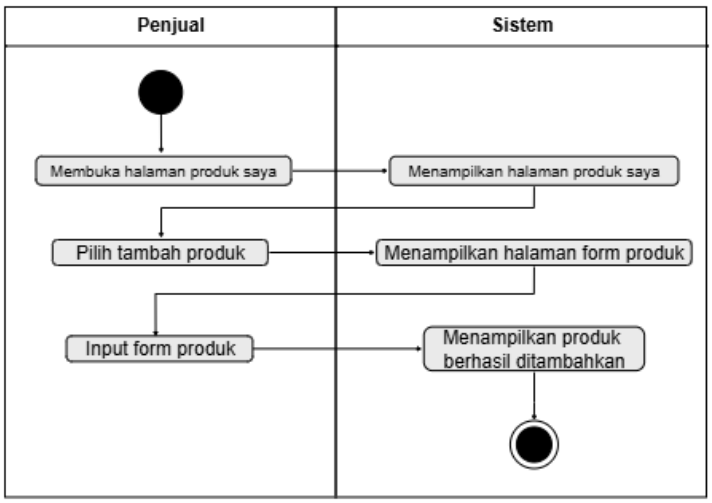
Gambar 4. 21 *Activity* diagram hubungkan hubungkan dompet

4) *Activity* diagram edit profil



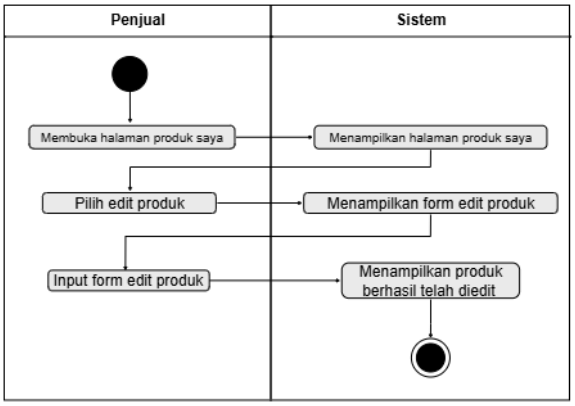
Gambar 4. 22 *Activity* diagram edit profil

5) *Activity* diagram tambah produk



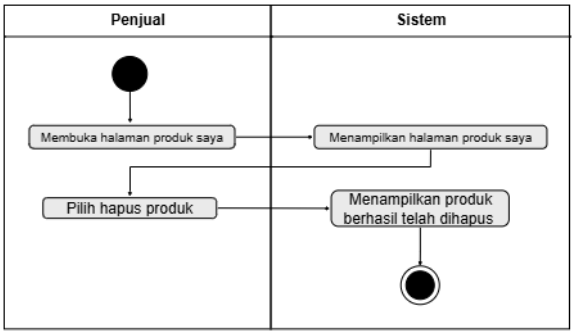
Gambar 4. 23 *Activity* diagram tambah produk

6) *Activity* diagram edit produk



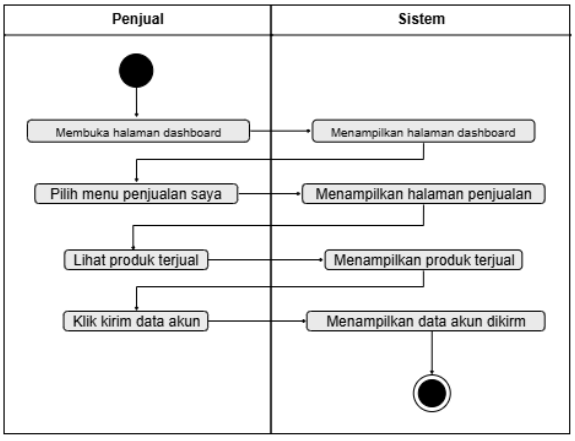
Gambar 4. 24 *Activity* diagram edit produk

7) *Activity* diagram hapus produk



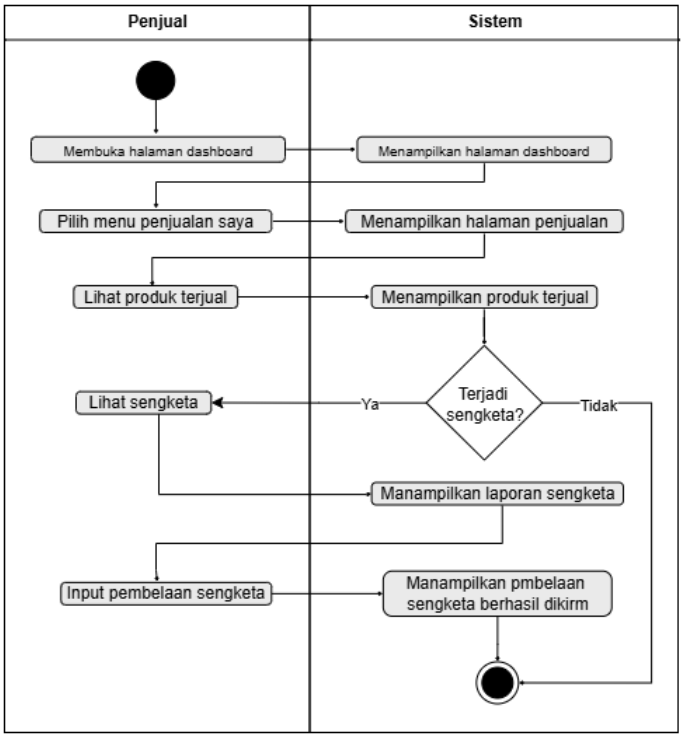
Gambar 4. 25 *Activity* diagram hapus produk

8) *Activity* diagram kirim detail akun



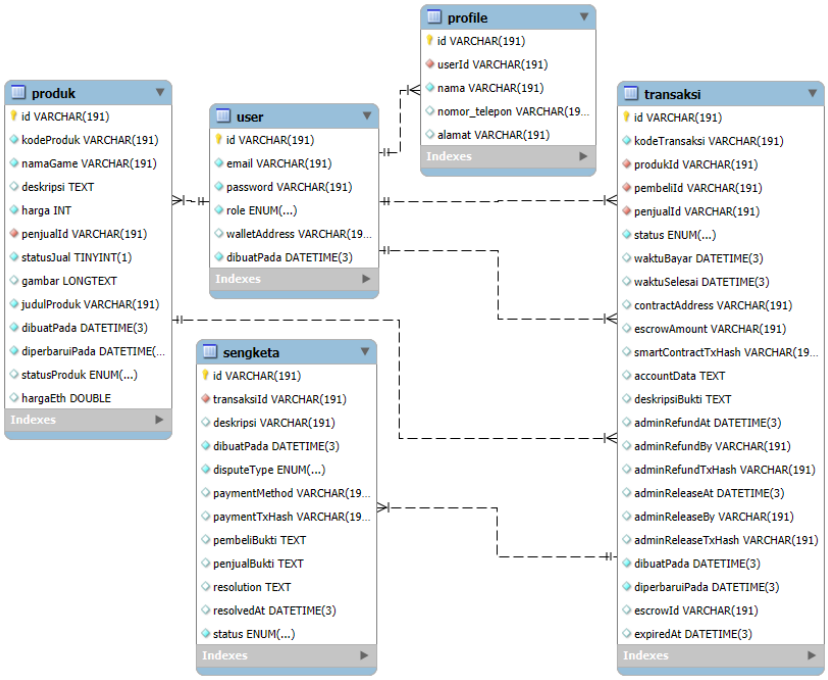
Gambar 4. 26 *Activity* diagram kirim detail akun

9) *Activity* diagram pembelaan sengketa



Gambar 4. 27 *Activity* diagram pembelaan sengketa

B. Diagram ERD



Gambar 4. 28 Diagram ERD

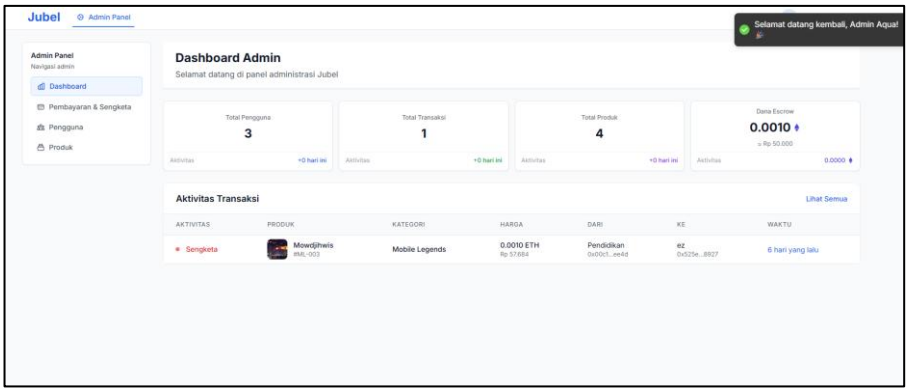
Diagram ERD tersebut menggambarkan hubungan antar tabel utama dalam sistem, yaitu tabel user yang menyimpan data pengguna sebagai aktor inti, tabel profile yang berisi informasi profil tambahan dari pengguna, tabel produk yang mencatat akun *game* yang diperjualbelikan, tabel transaksi yang merekam detail proses jual beli menggunakan mekanisme escrow, serta tabel sengketa yang mendokumentasikan proses penyelesaian konflik antara pembeli dan penjual dengan melibatkan admin sebagai penengah.

C. Pengujian Sistem

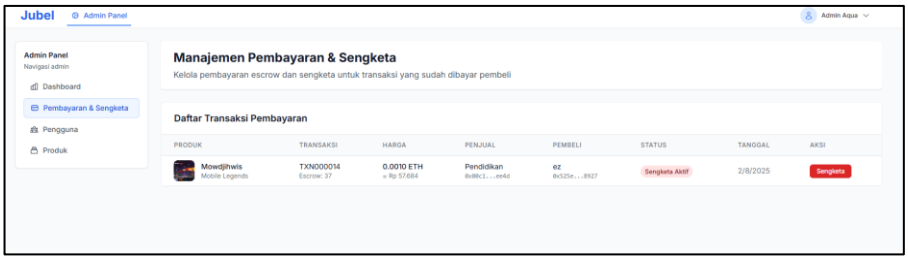
1. *Black box Testing*

a. Admin

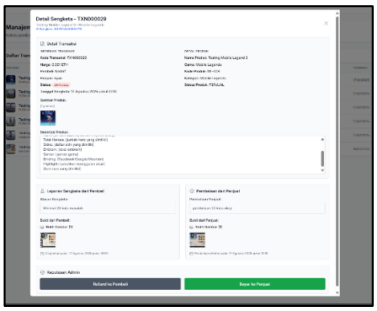
Tabel 4. 3 *Black box testing fitur login*

Tes Faktor	Hasil	Keterangan
Admin login menggunakan *email* dan *password* yang *valid*	✓	Berhasil, sistem menampilkan halaman admin panel bagian menu dashboard setelah *login* berhasil.
Screenshot		
		

Tabel 4. 4 *Black box testing* detail pembayaran dan sengketa

Tes Faktor	Hasil	Keterangan
Admin klik menu Pembayaran dan sengketa	✓	Berhasil, sistem menampilkan halaman pembayaran dan sengketa
Screenshot		
		

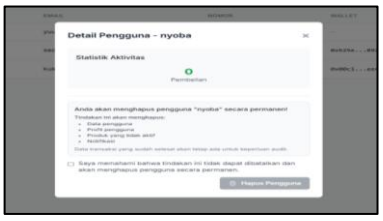
Tabel 4. 5 *Black box testing* klik lihat laporan sengketa

Tes Faktor	Hasil	Keterangan
Admin klik lihat laporan sengketa	✓	Berhasil, sistem menampilkan halaman laporan sengketa
Screenshot		
		

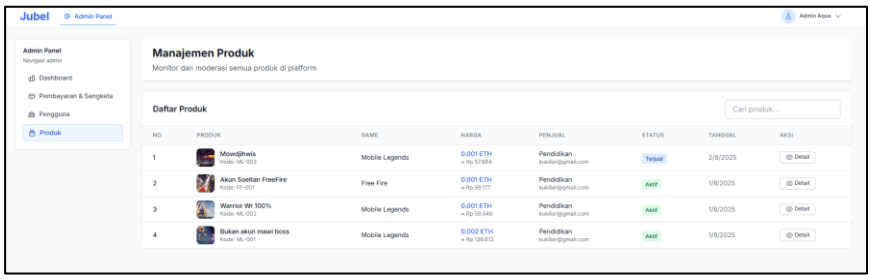
Tabel 4. 6 *Black box testing* menu manajemen pengguna

Tes Faktor	Hasil	Keterangan
Admin klik *menu* pengguna	✓	Berhasil, sistem menampilkan seluruh pengguna
Screenshot		
		

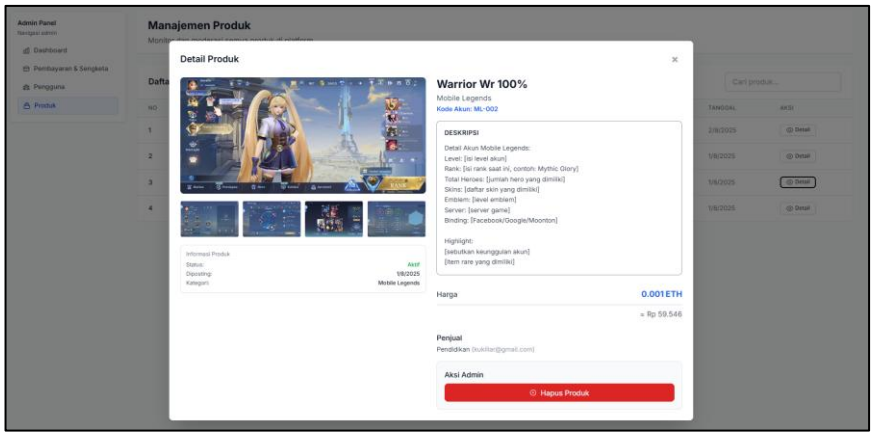
Tabel 4. 7 *Black box testing* detail aktivitas pengguna

Tes Faktor	Hasil	Keterangan
Admin klik *detail* pada pengguna	✓	Berhasil, sistem menampilkan aktivitas pengguna
Screenshot		
		

Tabel 4. 8 *Black box testing* menu manajemen produk

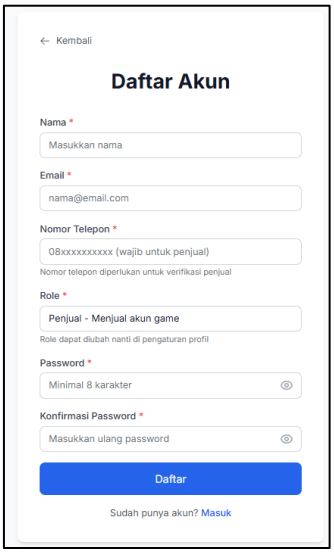
Tes Faktor	Hasil	Keterangan
Admin klik *menu* produk	✓	Berhasil, sistem menampilkan seluruh produk
Screenshot		
		

Tabel 4. 9 *Black box testing* detail *produk*

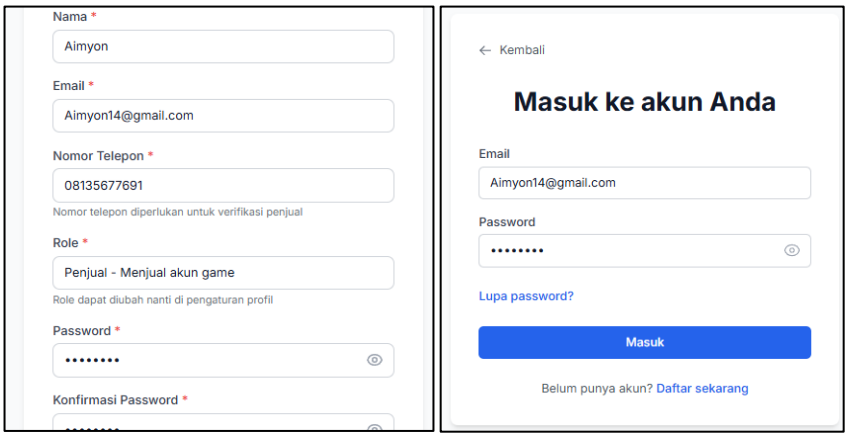
Tes Faktor	Hasil	Keterangan
Admin klik *detail* pada produk	✓	Berhasil, sistem menampilkan detail produk
Screenshot		
		

b. penjual

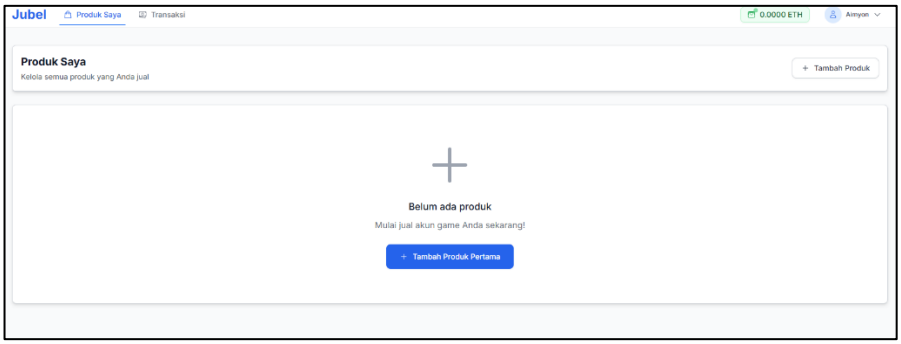
Tabel 4. 10 *Black box testing* registrasi penjual

Tes Faktor	Hasil	Keterangan
User pilih *registrasi*	✓	Berhasil, sistem menampilkan halaman registrasi
Screenshot		
		

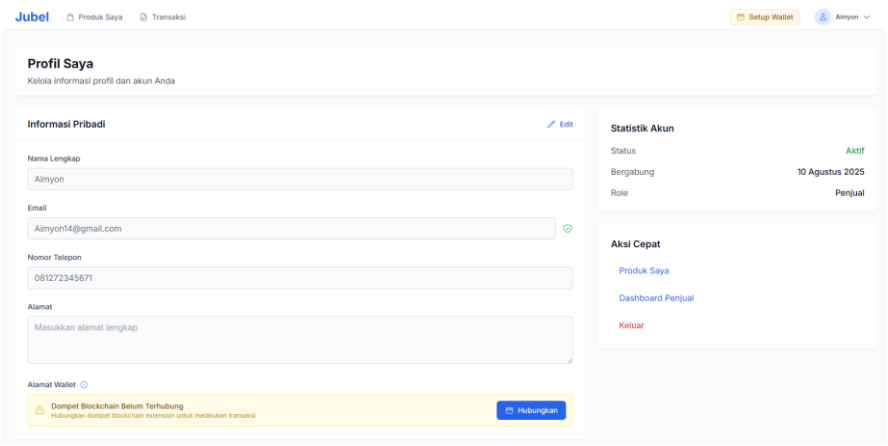
Lanjutan **Tabel 4.10**

Tes Faktor	Hasil	Keterangan
User mengisi *from registrasi* dengan *role* penjualan	✓	Berhasil, sistem berhasil menyimpan data dan role user menjadi penjual dan menampilkan halaman *login*.
Screenshot		
		

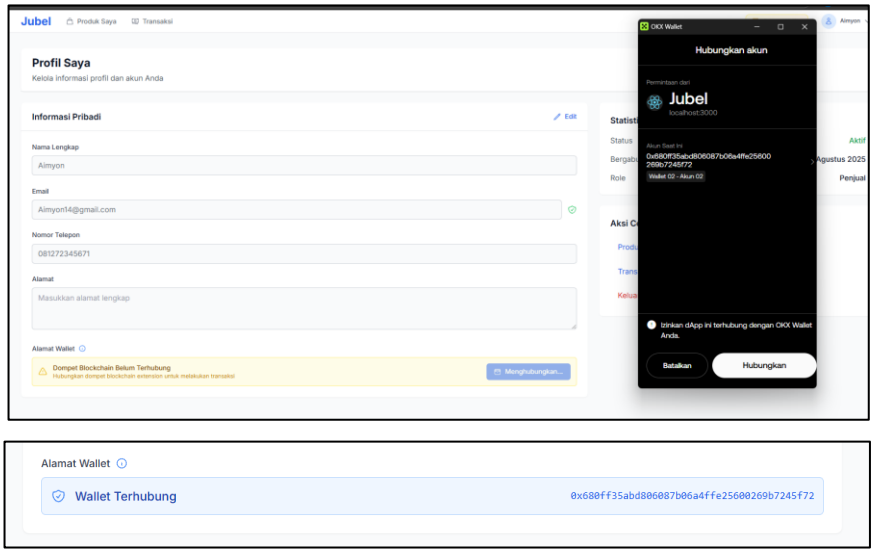
Tabel 4. 11 *Black box testing login*

Tes Faktor	Hasil	Keterangan
penjual klik masuk	✓	Berhasil, sistem menampilkan halaman produk saya pada role penjual
Screenshot		
		

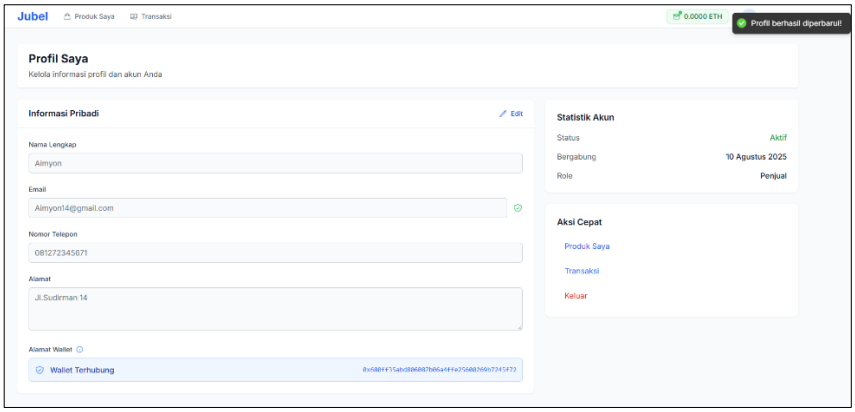
Tabel 4. 12 *Black box testing* profil

Tes Faktor	Hasil	Keterangan
penjual klik *menu* profil	✓	Berhasil, sistem menampilkan halaman profil
Screenshot		
		

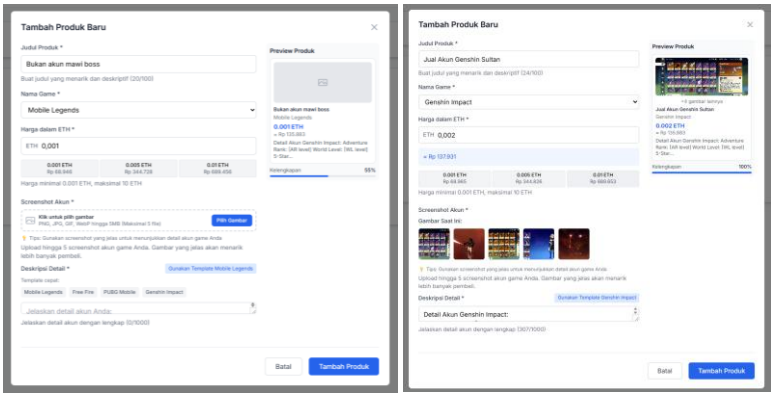
Tabel 4. 13 *Black box testing* hubungkan dompet

Tes Faktor	Hasil	Keterangan
Penjual klik hubungkan dompet *blockchain*	✓	Berhasil, sistem memanggil extension dompet *blockchain* dan terhubung dengan website
Screenshot		
		

Tabel 4. 14 *Black box testing* edit profil

Tes Faktor	Hasil	Keterangan
Penjual klik *edit* profil	✓	Berhasil, sistem menyimpan data profil yang telah diedit
		

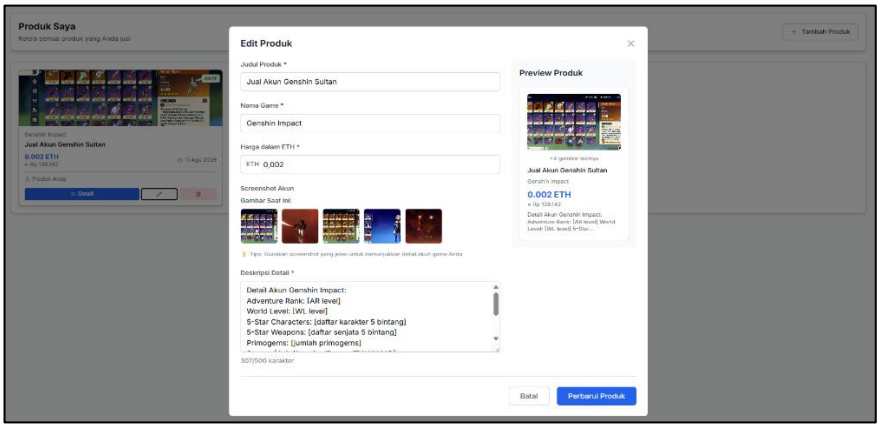
Tabel 4. 15 *Black box testing* menu tambah produk

Tes Faktor	Hasil	Keterangan
Penjual klik menu tambahkan produk	✓	Berhasil, sistem menampilkan form produk
Screenshot		
		

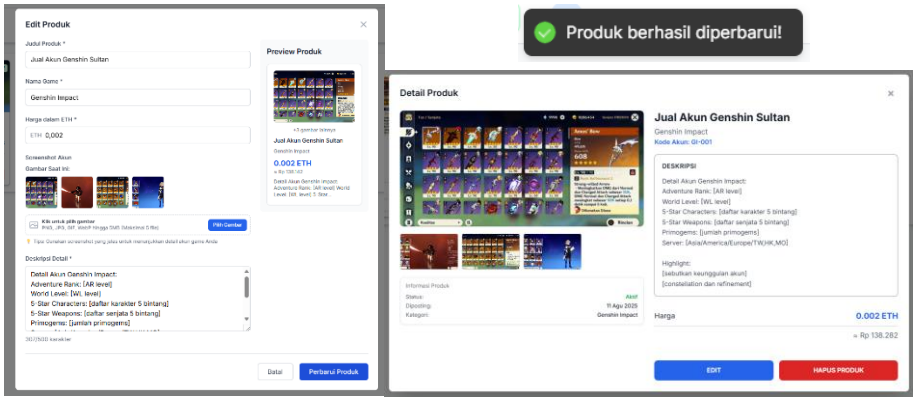
Tabel 4. 16 *Black box testing* tambah produk

Tes Faktor	Hasil	Keterangan
Penjual klik tambah produk	✓	Berhasil, sistem menampilkan produk baru
Screenshot		
		

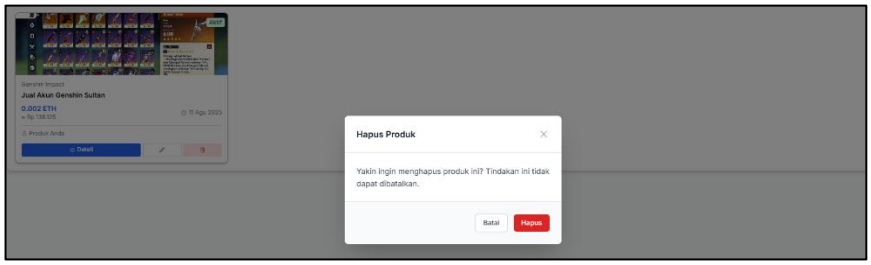
Tabel 4. 17 *Black box testing edit* produk

Tes Faktor	Hasil	Keterangan
Penjual klik *edit* produk	✓	Berhasil, sistem menampilkan form *edit* produk
Screenshot		
		

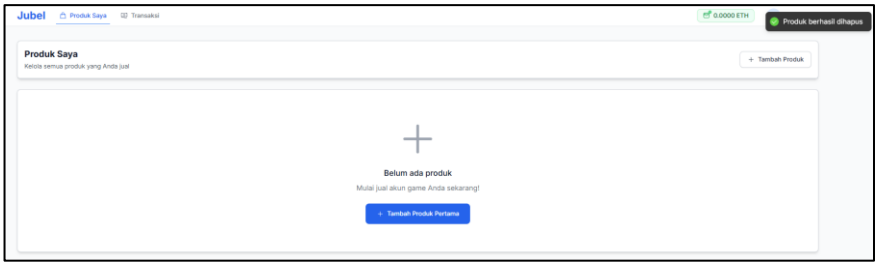
Tabel 4. 18 *Black box testing* perbarui produk

Tes Faktor	Hasil	Keterangan
Penjual klik perbarui produk	✓	Berhasil, sistem menyimpan perubahan produk
Screenshot		
		

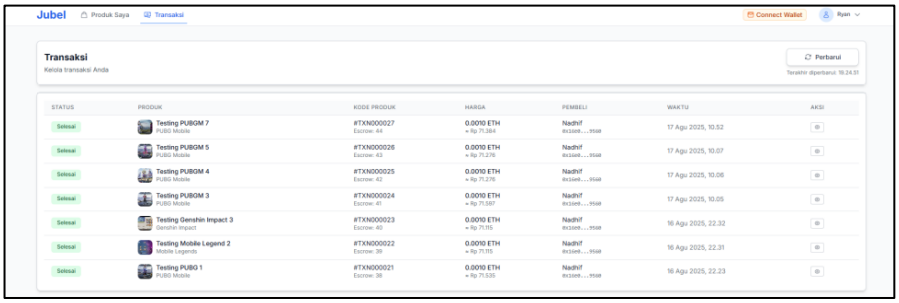
Tabel 4. 19 *Black box testing* hapus produk

Tes Faktor	Hasil	Keterangan
Penjual klik hapus produk	✓	Berhasil, sistem menampilkan pemberitahuan hapus produk
Screenshot		
		

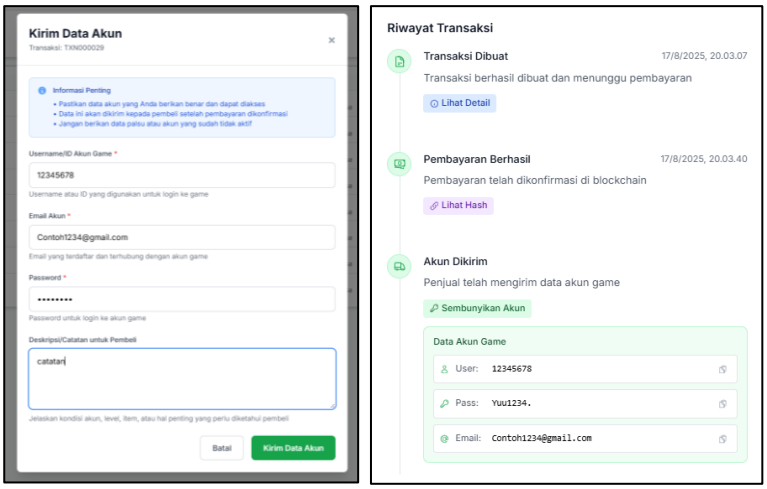
Tabel 4. 20 *Black box testing* produk dihapus

Tes Faktor	Hasil	Keterangan
Penjual klik hapus pada produk	✓	Berhasil, sistem menghapus produk
Screenshot		
		

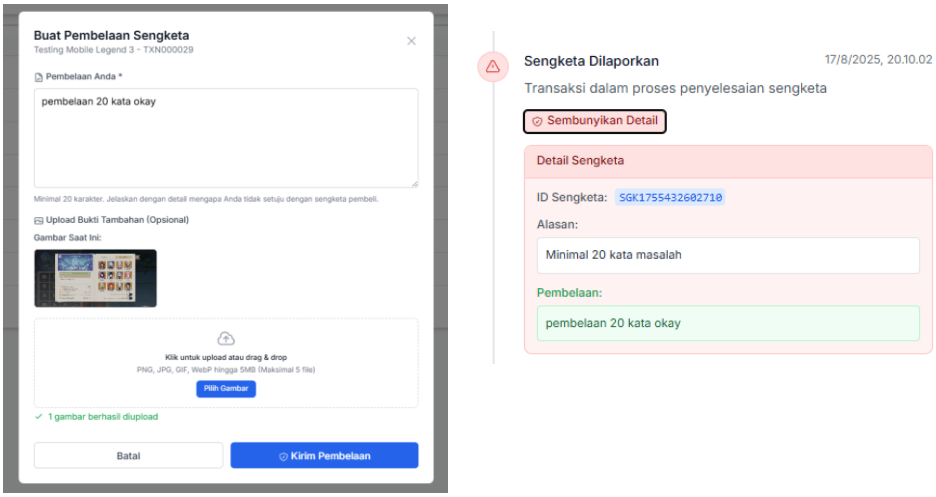
Tabel 4. 21 *Black box testing* transaksi

Tes Faktor	Hasil	Keterangan
Penjual klik menu transaksi	✓	Berhasil, sistem menampilkan halaman transaksi
Screenshot		
		

Tabel 4. 22 *Black box testing* kirim data akun

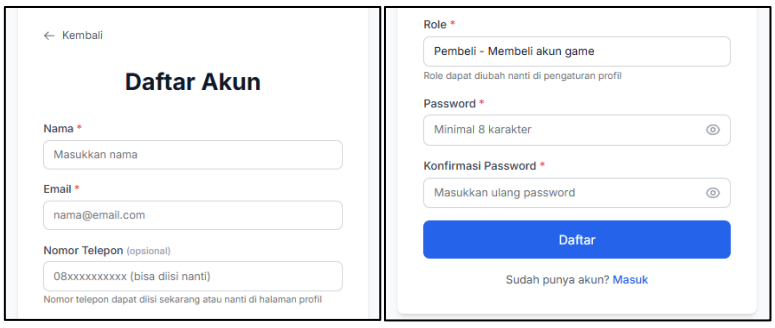
Tes Faktor	Hasil	Keterangan
Penjual klik kirim data akun	✓	Berhasil, sistem menampilkan data akun telah dikirim
Screenshot		
		

Tabel 4. 23 *Black box testing* kirim pembelaan

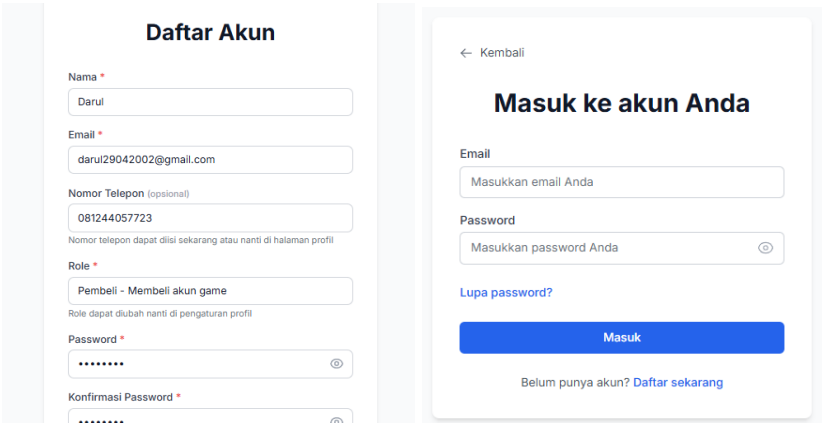
Tes Faktor	Hasil	Keterangan
Penjual klik tombol pembelaan	✓	Berhasil, sistem menampilkan pembelaan berhasil
Screenshot		
		

c. Pembeli

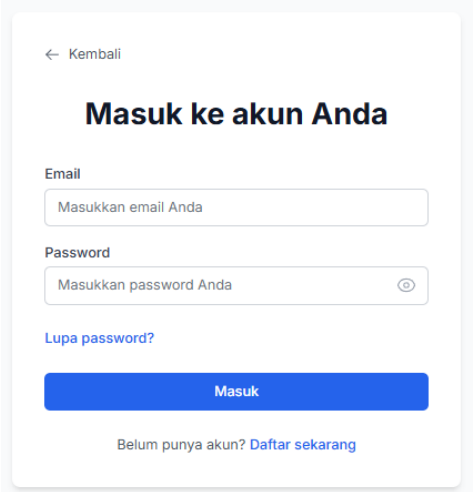
Tabel 4. 24 *Black box testing* registrasi pembeli

Tes Faktor	Hasil	Keterangan
User pilih *registrasi*	✓	Berhasil, sistem menampilkan halaman registrasi
Screenshot		
		

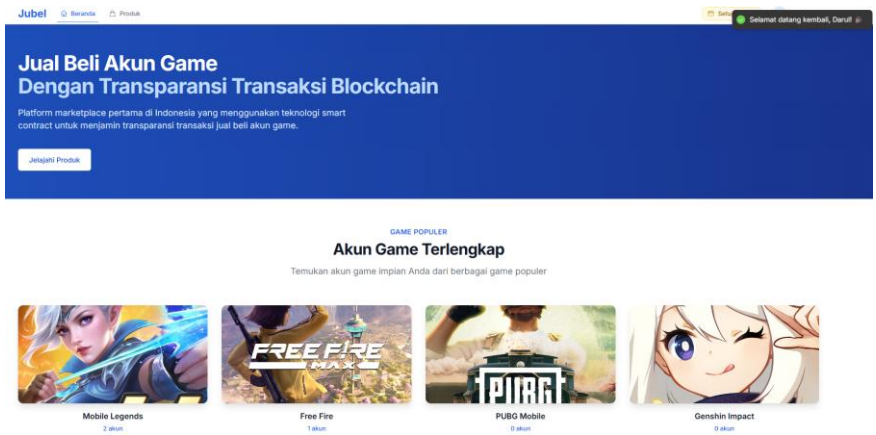
Lanjutan **Table 4.24**

Tes Faktor	Hasil	Keterangan
User mengisi from *registrasi* dengan role pembeli	✓	Berhasil, sistem berhasil menyimpan data dan role user menjadi pembeli dan menampilkan halaman *login*.
Screenshot		
		

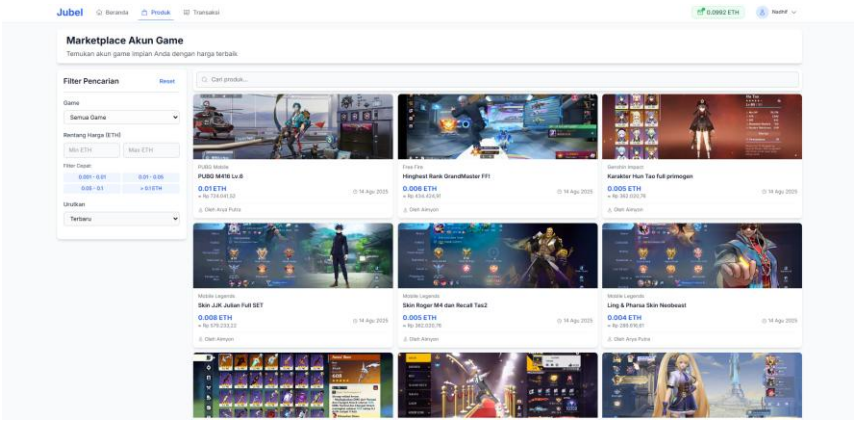
Tabel 4. 25 *Black box testing login*

Tes Faktor	Hasil	Keterangan
Pembeli pilih *login*	✓	Berhasil, sistem berhasil menampilkan halaman *login*
Screenshot		
		

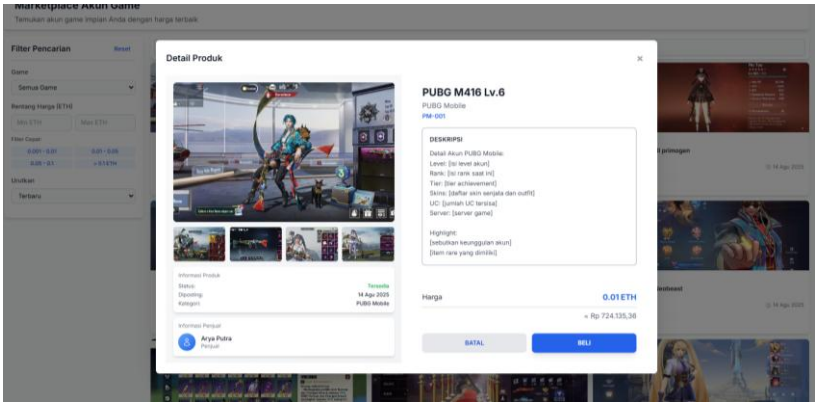
Tabel 4. 26 *Black box testing* klik masuk

Tes Faktor	Hasil	Keterangan
Pembeli klik masuk	✓	Berhasil, sistem menampilkan halaman beranda pada role pembeli
Screenshot		
		

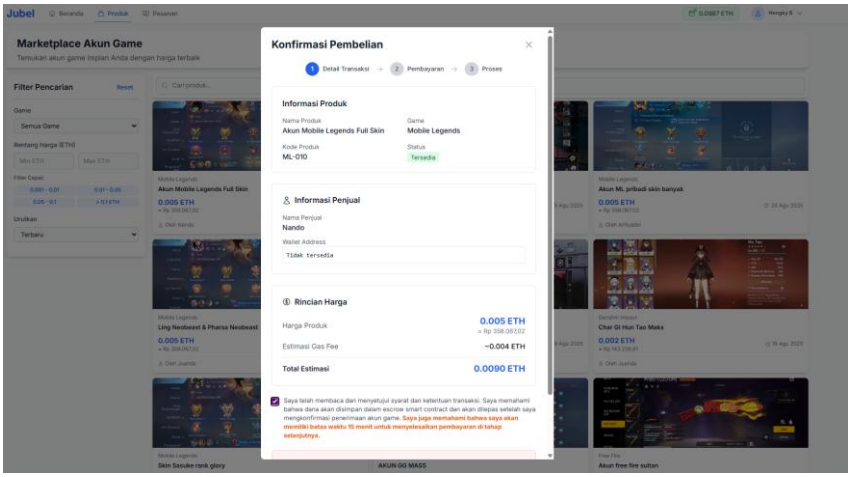
Tabel 4. 27 *Black box testing* menu produk

Tes Faktor	Hasil	Keterangan
Pembeli memilih klik menu produk	✓	Berhasil, sistem menampilkan halaman produk
Screenshot		
		

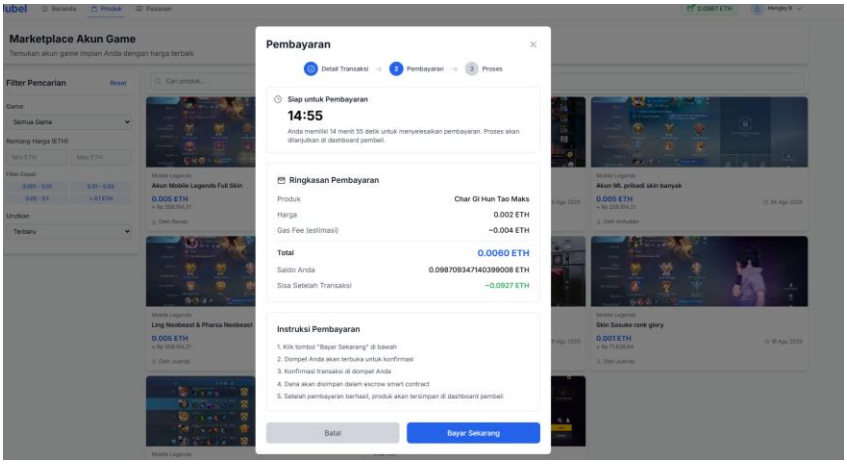
Tabel 4. 28 *Black box testing* pilih produk

Tes Faktor	Hasil	Keterangan
Pembeli memilih klik produk	✓	Berhasil, sistem menampilkan produk pilihan
Screenshot		
		

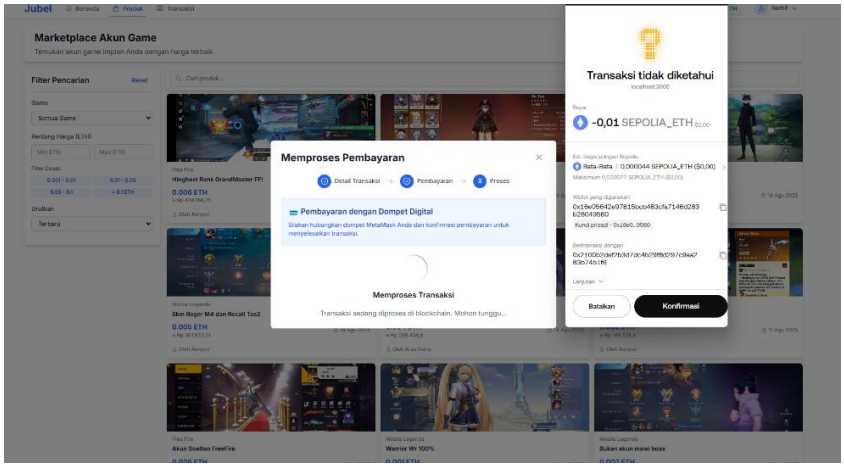
Tabel 4. 29 *Black box testing* beli produk

Tes Faktor	Hasil	Keterangan
Pembeli memilih klik beli produk	✓	Berhasil, sistem menampilkan detail transaksi produk
Screenshot		
		

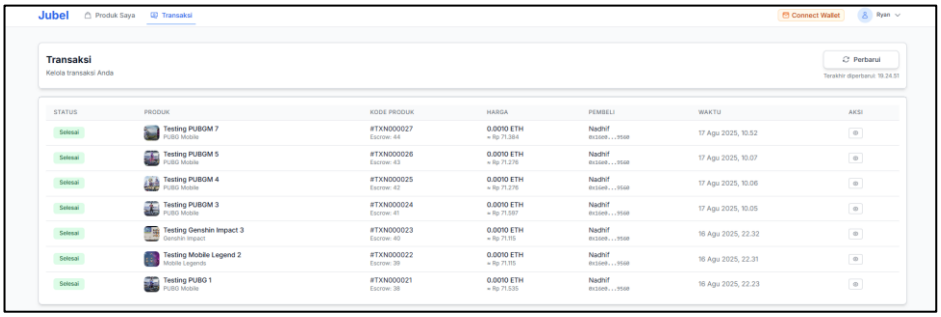
Tabel 4. 30 *Black box testing* lanjut bayar produk

Tes Faktor	Hasil	Keterangan
Pembeli memilih klik lanjut bayar produk	✓	Berhasil, sistem menampilkan timer pada pembayaran
Screenshot		
		

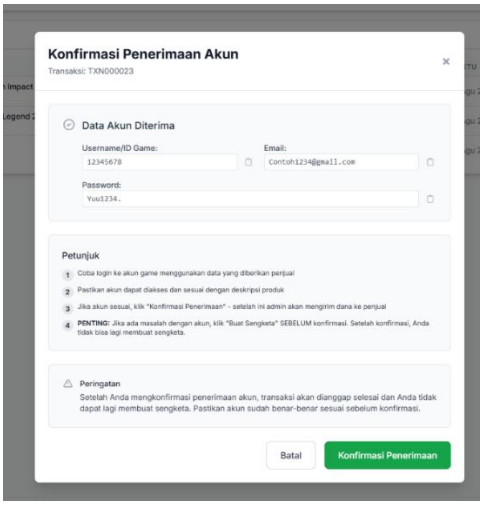
Tabel 4. 31 *Black box testing* lanjut bayar produk

Tes Faktor	Hasil	Keterangan
Pembeli memilih klik bayar sekarang	✓	Berhasil, sistem menampilkan proses transaksi dan memanggil extension dompet
Screenshot		
		

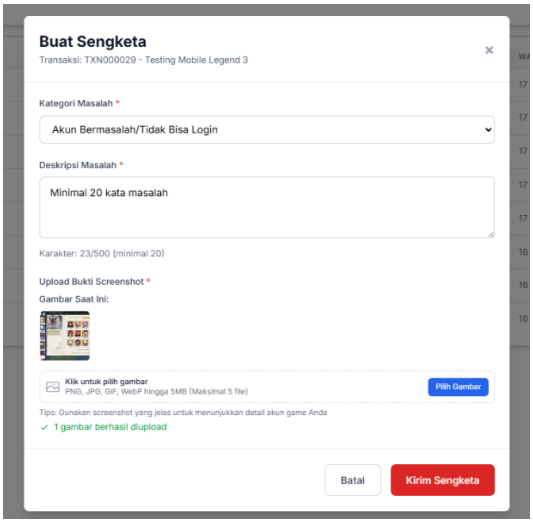
Tabel 4. 32 *Black box testing* pesanan

Tes Faktor	Hasil	Keterangan
pembeli klik menu pesanan	✓	Berhasil, sistem menampilkan halaman pesanan
Screenshot		
		

Tabel 4. 33 *Black box testing* konfirmasi akun sesuai

Tes Faktor	Hasil	Keterangan
pembeli klik menu pesanan	✓	Berhasil, sistem menampilkan halaman pesanan
Screenshot		
		

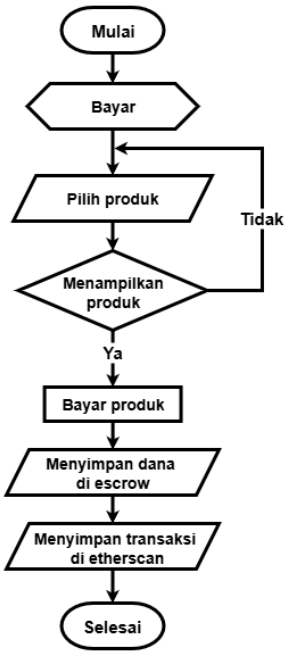
Tabel 4. 34 *Black box testing* melakukan sengketa

Tes Faktor	Hasil	Keterangan
pembeli klik menu pesanan	✓	Berhasil, sistem menampilkan halaman pesanan
Screenshot		
		

2. *White box Testing*

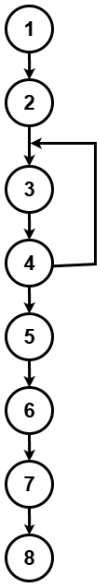
a. *White box testing simulasi blockchain*

1) *Flowchart* beli produk tercatat di *etherscan*



Gambar 4. 29 *Flowchart* beli produk tercatat di etherscan

2) *Flowgraph* beli produk tercatat di *etherscan*



Gambar 4. 30 *Flowgraph* beli produk tercatat di *etherscan*

Berdasarkan gambar diatas, dilakukan perhitungan sebagai berikut

a) Menghitung *Cyclomatic Complexity* V (G) = E – N + 2

N (node) = 8

E (edge) = 8

P (node) = 1

Penyelesaian V (G) = E – N + 2

= 8 – 8 + 2

= 2

Predikat Node (N) = P + 1

= 1 + 1

= 2

b) Berdasarkan perhitungan *Cyclomatic Complexity* dari *Flowgraph* diatas memiliki *Region* = 2

c) *Independent path* pada *Flowgraph* diatas adalah:

Path 1 = 1-2-3-4-5-6-7-8

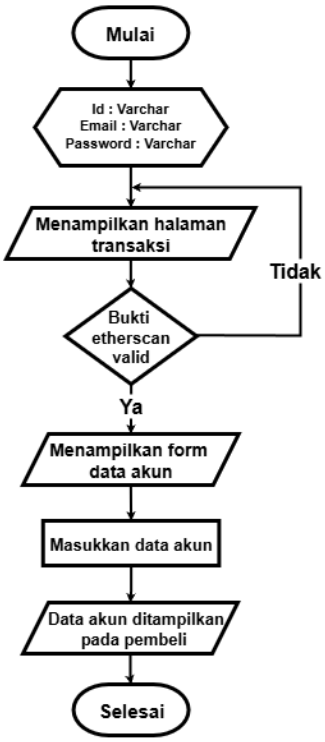
Path 2 = 1-2-3-4-3-4-5-6-7-8

d) Grafik Matriks

Tabel 4. 35 Grafik matriks beli produk tercatat di *etherscan*

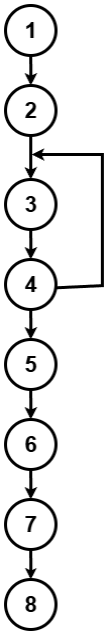
	1	2	3	4	5	6	7	8	E - 1
1		1							1 - 1 = 0
2			1						1 - 1 = 0
3				1					1 - 1 = 0
4			1		1				2 - 1 = 1
5						1			1 - 1 = 0
6							1		1 - 1 = 0
7								1	1 - 1 = 0
8									0
SUM (E + 1)									**1 + 1 = 2**

3) *Flowchart* data transaksi valid pada penjual



Gambar 4. 31 *Flowchart* data transaksi valid pada penjual

4) *Flowgraph* data transaksi valid pada penjual



Gambar 4. 32 *Flowgraph* data transaksi valid pada penjual

Berdasarkan gambar diatas, dilakukan perhitungan sebagai berikut

a) Menghitung *Cyclomatic Complexity* V (G) = E – N + 2

N (node) = 8

E (edge) = 8

P (node) = 1

Penyelesaian V (G) = E – N + 2

= 8 – 8 + 2

= 2

Predikat Node (N) = P + 1

= 1 + 1

= 2

b) Berdasarkan perhitungan *Cyclomatic Complexity* dari *Flowgraph* diatas memiliki *Region* = 2

c) *Independent path* pada *Flowgraph* diatas adalah:

Path 1 = 1-2-3-4-5-6-7-8

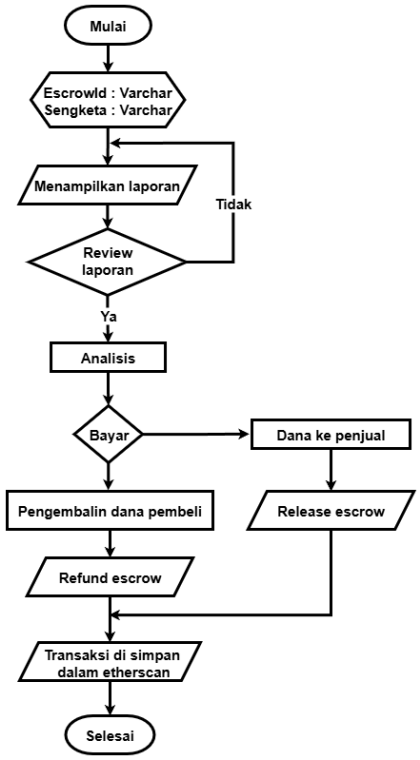
Path 2 = 1-2-3-4-3-4-5-6-7-8

d) Grafik Matriks

Tabel 4. 36 Grafik matriks data transaksi valid pada penjual

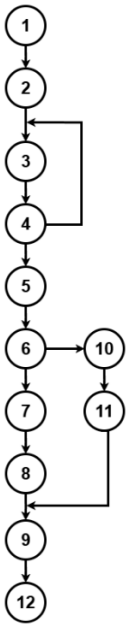
	1	2	3	4	5	6	7	8	E - 1
1		1							1 - 1 = 0
2			1						1 - 1 = 0
3				1					1 - 1 = 0
4			1		1				2 - 1 = 1
5						1			1 - 1 = 0
6							1		1 - 1 = 0
7								1	1 - 1 = 0
8									0
SUM (E + 1)									**1 + 1 = 2**

5) *Flowchart* admin laporan sengketa



Gambar 4. 33 *Flowchart* admin laporan sengketa

6) *Flowgraph* admin laporan sengketa



Gambar 4. 34 *Flowgraph* admin laporan sengketa

Berdasarkan gambar diatas, dilakukan perhitungan sebagai berikut

a) Menghitung *Cyclomatic Complexity* V (G) = E – N + 2

N (node) = 12

E (edge) = 13

P (node) = 2

Penyelesaian V (G) = E – N + 2

= 13 – 12 + 2

= 3

Predikat Node (N) = P + 1

= 2 + 1

= 3

b) Berdasarkan perhitungan *Cyclomatic Complexity* dari *Flowgraph* diatas memiliki *Region* = 3

c) *Independent path* pada *Flowgraph* diatas adalah:

Path 1 = 1-2-3-4-5-6-7-8-9-12

Path 2 = 1-2-3-4-3-4-5-6-7-8-9-12

Path 3 = 1-2-3-4-5-6-10-11-9-12

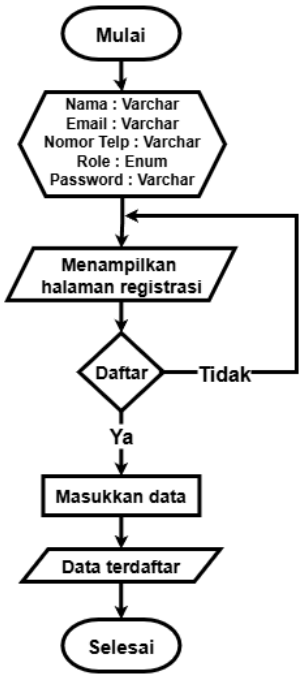
d) Grafik Matriks

Tabel 4. 37 Grafik matriks admin keputusan pembayaran

	1	2	3	4	5	6	7	8	9	10	11	12	E - 1
1		1											1 - 1 = 0
2			1										1 - 1 = 0
3				1									1 - 1 = 0
4			1		1								2 - 1 = 1
5						1							1 - 1 = 0
6							1			1			2 - 1 = 1
7								1					1 - 1 = 0
8									1				1 - 1 = 0
9												1	1 - 1 = 0
10											1		1 - 1 = 0
11									1				1 - 1 = 0
12													0
SUM (E + 1)													2 + 1 = 3

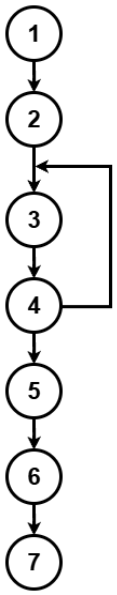
b. *White box testing* pembeli

1) Flowchart registrasi



Gambar 4. 35 *Flowchart* pembeli registrasi

2) Flowgraph registrasi



Gambar 4. 36 *Flowgraph* pembeli registrasi

Berdasarkan gambar diatas, dilakukan perhitungan sebagai berikut:

a) Menghitung *Cyclomatic Complexity* V (G) = E – N + 2

N (node) = 7

E (edge) = 7

P (node) = 1

Penyelesaian V (G) = E – N + 2

= 7 – 7 + 2

= 2

Predikat Node (N) = P + 1

= 1 + 1

= 2

b) Berdasarkan perhitungan *Cyclomatic Complexity* dari *Flowgraph* diatas memiliki *Region* = 2

c) *Independent path* pada *Flowgraph* diatas adalah:

Path 1 = 1-2-3-4-5-6-7

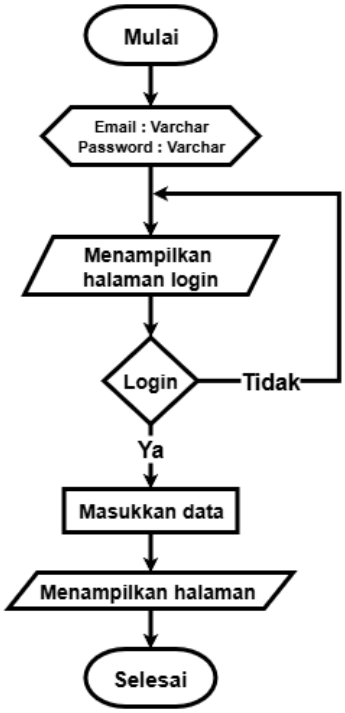
Path 2 = 1-2-3-4-3-4-5-6-7

d) Grafik Matriks

Tabel 4. 38 Grafik matriks registrasi

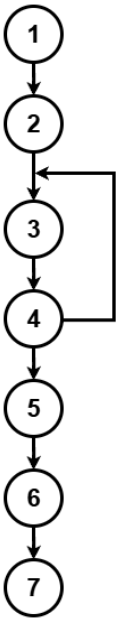
	1	2	3	4	5	6	7	E - 1
1		1						1 - 1 = 0
2			1					1 - 1 = 0
3				1				1 - 1 = 0
4			1		1			2 - 1 =1
5						1		1 - 1 = 0
6							1	1 - 1 = 0
7								1 - 1 = 0
SUM (E + 1)								1 + 1 = 2

3) Flowchart login



Gambar 4. 37 *Flowchart login*

4) Flowgraph login



Gambar 4. 38 *Flowgraph login*

Berdasarkan gambar diatas, dilakukan perhitungan sebagai berikut:

a) Menghitung *Cyclomatic Complexity* V (G) = E – N + 2

N (node) = 7

E (edge) = 7

P (node) = 1

Penyelesaian V (G) = E – N + 2

= 7 – 7 + 2

= 2

Predikat Node (N) = P + 1

= 1 + 1

= 2

b) Berdasarkan perhitungan *Cyclomatic Complexity* dari *Flowgraph* diatas memiliki *Region* = 2

c) *Independent path* pada *Flowgraph* diatas adalah:

Path 1 = 1-2-3-4-5-6-7

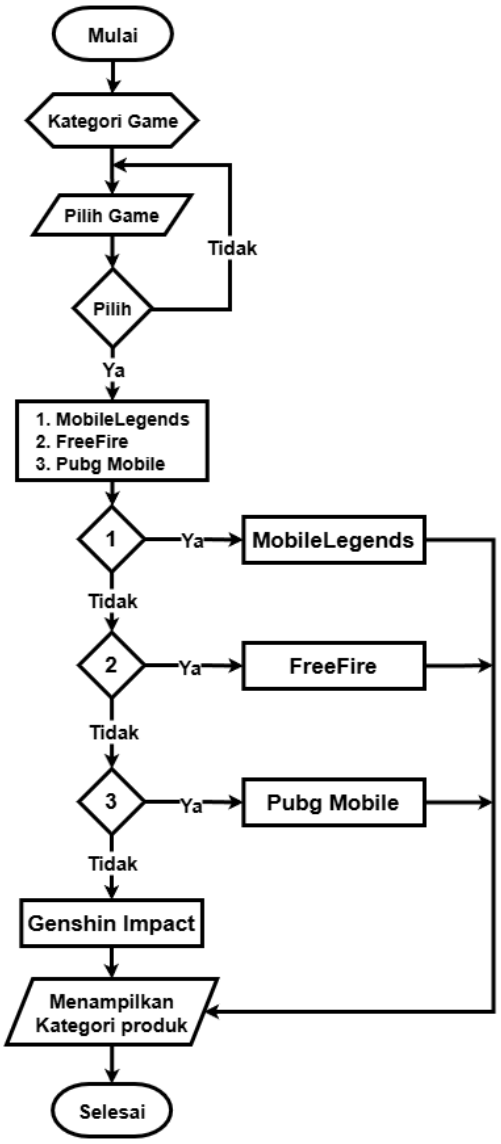
Path 2 = 1-2-3-4-3-4-5-6-7

d) Grafik Matriks

Tabel 4. 39 Grafik matriks *login*

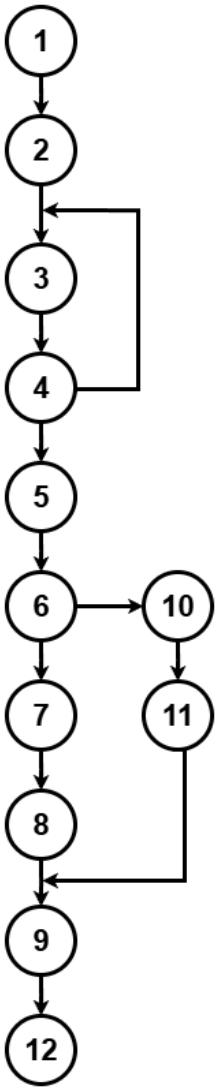
	1	2	3	4	5	6	7	E - 1
1		1						1 - 1 = 0
2			1					1 - 1 = 0
3				1				1 - 1 = 0
4			1		1			2 - 1 =1
5						1		1 - 1 = 0
6							1	1 - 1 = 0
7								1 - 1 = 0
SUM (E + 1)								1 + 1 = 2

5) *Flowchart* pilih kategori *game*



Gambar 4. 39 *Flowchart* pilih kategori *game*

6) *Flowgraph* pilih kategori *game*



Gambar 4. 40 *Flowgraph* pilih kategori *game*

Berdasarkan gambar diatas, dilakukan perhitungan sebagai berikut:

a) Menghitung *Cyclomatic Complexity* V (G) = E – N + 2

N (node) = 14

E (edge) = 17

P (node) = 4

Penyelesaian V (G) = E – N + 2

= 17 – 14 + 2

= 5

Predikat Node (N) = P + 1

= 4 + 1

= 5

b) Berdasarkan perhitungan *Cyclomatic Complexity* dari *Flowgraph* diatas memiliki *Region* = 5

c) *Independent path* pada *Flowgraph* diatas adalah:

Path 1 = 1-2-3-4-5-6-7-8-9-10-14

Path 2 = 1-2-3-4-3-4-5-6-7-8-9-10-14

Path 3 = 1-2-3-4-5-6-11-10-14

Path 4 = 1-2-3-4-5-6-7-12-10-14

Path 5 = 1-2-3-4-5-6-7-8-13-10-14

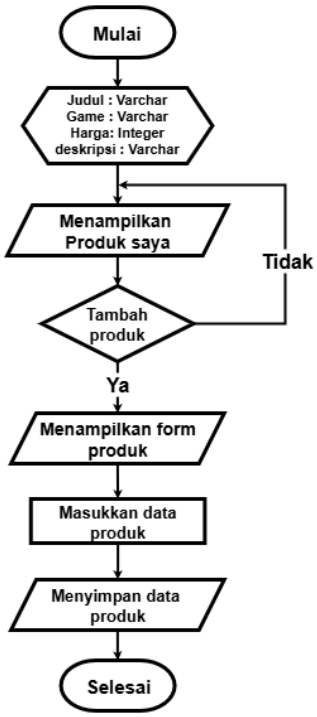
d) Grafik Matriks

Tabel 4. 40 Grafik matriks pilih kategori *game*

	1	2	3	4	5	6	7	8	9	10	11	12	13	14	E - 1
1		1													1 - 1 = 0
2			1												1 - 1 = 0
3				1											1 - 1 = 0
4			1		1										2 - 1 = 1
5						1									1 - 1 = 0
6							1				1				2 - 1 = 1
7								1				1			2 - 1 = 1
8									1				1		2 - 1 = 1
9										1					1 - 1 = 0
10														1	1 - 1 = 0
11										1					1 - 1 = 0
12										1					1 - 1 = 0
13										1					1 - 1 = 0
14															0
SUM (E + 1)															4 + 1 = 5

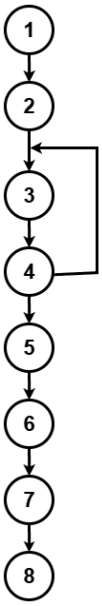
c. *White box testing* penjual

1) *Flowchart* tambah produk



Gambar 4. 41 *Flowchart* tambah produk

2) Flowgraph tambah produk



Gambar 4. 42 *Flowgraph* tambah produk

Berdasarkan gambar diatas, dilakukan perhitungan sebagai berikut:

a) Menghitung *Cyclomatic Complexity* V (G) = E – N + 2

N (node) = 8

E (edge) = 8

P (node) = 1

Penyelesaian V (G) = E – N + 2

= 8 – 8 + 2

= 2

Predikat Node (N) = P + 1

= 1 + 1

= 2

b) Berdasarkan perhitungan *Cyclomatic Complexity* dari *Flowgraph* diatas memiliki *Region* = 2

c) *Independent path* pada *Flowgraph* diatas adalah:

Path 1 = 1-2-3-4-5-6-7-8

Path 2 = 1-2-3-4-3-4-5-6-7-8

d) Grafik Matriks

Tabel 4. 41 Grafik matriks tambah produk

	1	2	3	4	5	6	7	8	E - 1
1		1							1 - 1 = 0
2			1						1 - 1 = 0
3				1					1 - 1 = 0
4			1		1				2 - 1 = 1
5						1			1 - 1 = 0
6							1		1 - 1 = 0
7								1	1 - 1 = 0
8									0
SUM (E + 1)									**1 + 1 = 2**

3. *Fungsional Testing*

a. *Deploy smart contract escrow* aset *ethereum sepolia*

Deploy smart contract pada aset *ethereum sepolia*. Potongan kode berikut menunjukkan implementasi dengan membuat smart contract escrow.

```
const { expect } = require("chai");
const { Ethers } = require("Hardhat");

describe("EscrowFixed", () => {
  let owner, buyer, seller, admin, other, escrow;
  const value = Ethers.parseEther("0.01");

  beforeEach(async () => {
    [owner, buyer, seller, admin, other] = await
Ethers.getSigners();

    const Escrow = await
Ethers.getContractFactory("EscrowFixed");
    escrow = await Escrow.deploy();
    await escrow.waitForDeployment();

    await escrow.connect(owner).addAdmin(admin.address);
  });

  it("createEscrow: menahan dana & status = FUNDED", async
() => {
    const tx = await escrow.connect(buyer).createEscrow(
      seller.address,
      { value }
    );
    await tx.wait();

    const escrowId = await escrow.escrowCounter();
    const data = await escrow.getEscrow(escrowId);

    expect(data.buyer).to.equal(buyer.address);
    expect(data.seller).to.equal(seller.address);
    expect(data.amount).to.equal(value);
       expect(data.status).to.equal(1n);
  });

  it("adminRefund: kembalikan dana ke buyer", async () =>
{
    await
escrow.connect(buyer).createEscrow(seller.address, "P001",
{ value });
    const id = await escrow.escrowCounter();

    const balBefore = await
Ethers.provider.getBalance(buyer.address);
```

```
      const tx = await
escrow.connect(admin).adminRefund(id, "refund reason!!");
    await tx.wait();
    const balAfter = await
Ethers.provider.getBalance(buyer.address);

    expect(balAfter).to.be.greaterThan(balBefore);

    const data = await escrow.getEscrow(id);
  });

  it("createDispute/resolveDispute: admin pilih pemenang",
async () => {
    await
escrow.connect(buyer).createEscrow(seller.address,{ value
});
    const id = await escrow.escrowCounter();

    await escrow.connect(buyer).createDispute(id);
    let data = await escrow.getEscrow(id);

    expect(data.status).to.equal(3n);

    await escrow.connect(admin).resolveDispute(id,
seller.address);
    data = await escrow.getEscrow(id);
    expect(data.status).to.equal(2n);
  });
});
```

b. Hasil dan pengujian

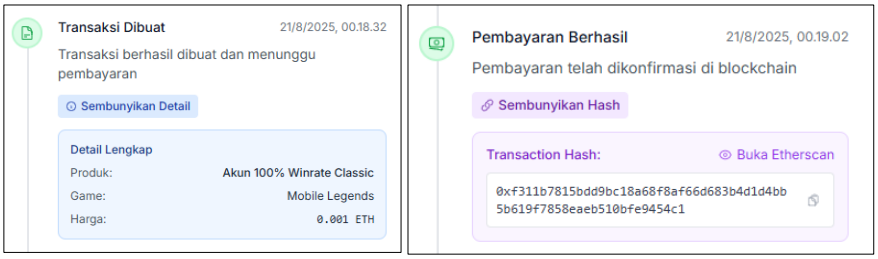
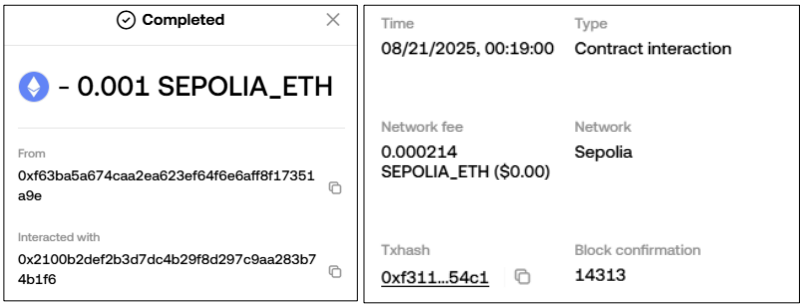
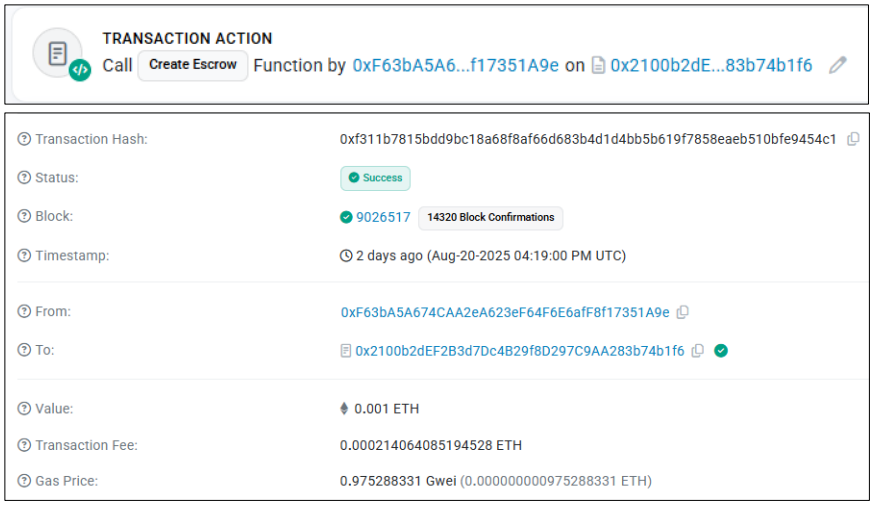
Tabel 4. 42 Hasil *deploy smart contrat*

Kategori	Hasil
Alamat Pembuat	0xE14fcb0fDb1256445DC6ddd876225a8fAd9D211F
Aset *token*	*Ethereum Sepolia (Tesnet)*
Toolkit	*Hardhat*
Alamat *Smart contract*	0x2100b2dEF2B3d7Dc4B29f8D297C9AA283b74b1f6

Smart contract escrow berhasil dideploy pada jaringan Ethereum Sepolia. Address Pembuat 0xE14fcb0fDb1256445DC6ddd876225a8fAd9D211F tercatat sebagai owner sekaligus admin awal kontrak. Alamat Kontrak yang terbentuk dan menjadi endpoint seluruh interaksi selanjutnya adalah 0x2100b2dEF2B3d7Dc4B29f8D297C9AA283b74b1f6.

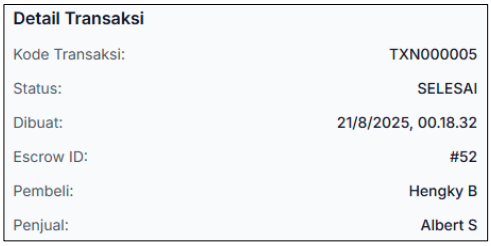
1) Pengujian transaksi

Tabel 4. 43 Pengujian transaksi

<table>
<tr><th rowspan="2">Tes faktor</th><th colspan="3">Hasil</th></tr>
<tr><th>Website</th><th>Dompet</th><th>Etherscan</th></tr>
<tr><td>Uji pembelian riwayat transaksi tercatat pada website, dompet, Etherscan</td><td>Berhasil</td><td>Berhasil</td><td>Berhasil</td></tr>
<tr><td colspan="4">Screenshot website</td></tr>
<tr><td colspan="4">
</td></tr>
<tr><td colspan="4">Screenshot dompet</td></tr>
<tr><td colspan="4">
</td></tr>
<tr><td colspan="4">Screenshot etherscan</td></tr>
<tr><td colspan="4">
</td></tr>
</table>

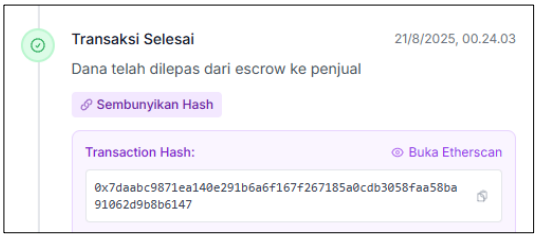
2) Verifikasi *event* dan status transaksi

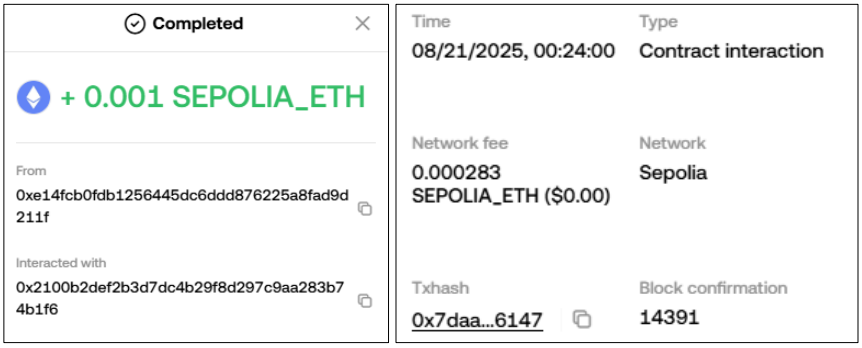
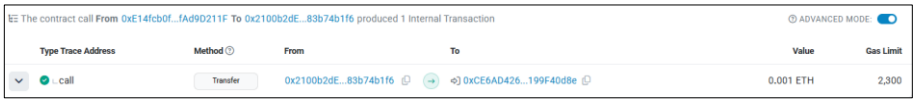
Tabel 4. 44 Verifikasi *event* dan status transaksi

Tes faktor	Hasil	
	Website	Etherscan
Uji riwayat verfikasi *event* dan status transaksi	Berhasil	Berhasil
Screenshot website		
		
Screenshot etherscan		
		

3) Pengujian pembayaran transaksi oleh admin

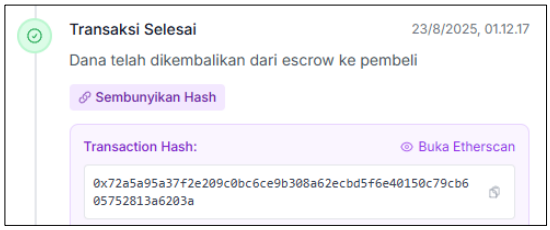
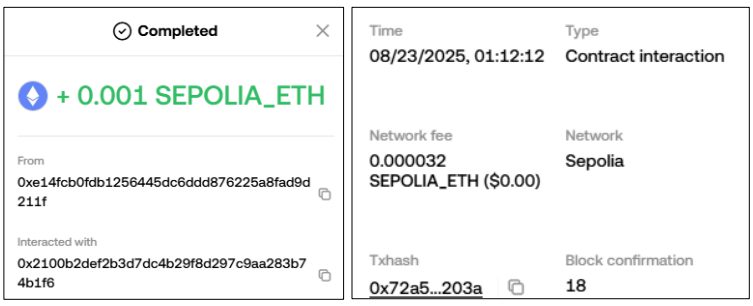
Tabel 4. 45 Pengujian pembayaran transaksi oleh admin

Tes faktor	Hasil		
	Website	Dompet	Etherscan
Uji pembayaran transaksi kepada penjual yang dilakukan oleh admin	Berhasil	Berhasil	Berhasil
Screenshot website			
			

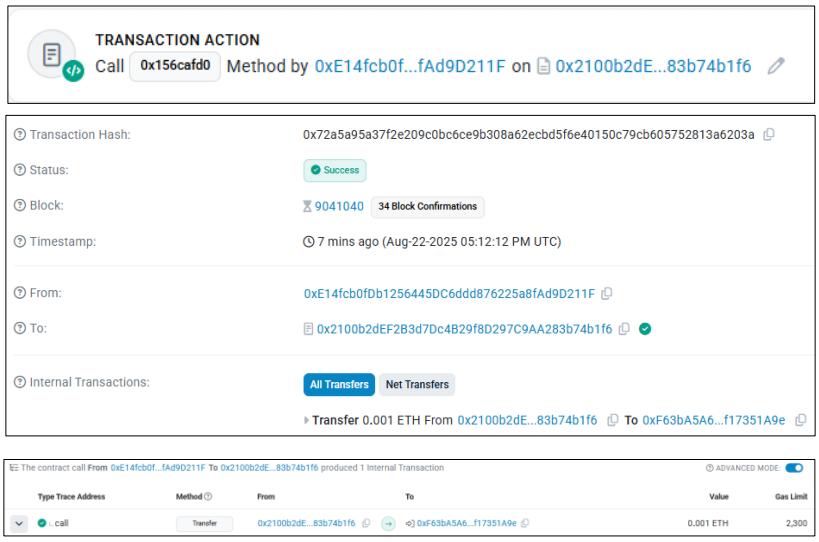
Screenshot dompet

Screenshot etherscan


4) Transaksi pembayaran *refund* oleh admin

Tabel 4. 46 Pengujian pembayaran *refund* oleh admin

Tes faktor	**Hasil**		
	Website	**Dompet**	**Etherscan**
Uji pembayaran transaksi pembayaran *refund* oleh admin	Berhasil	Berhasil	Berhasil
Screenshot website			
			
Screenshot dompet			
			

Screenshot etherscan



Tabel 4. 47 Hasil uji

No	Fitur	Peran	Hasil	TxHash	status
1	Create Escrow	Pembeli	Dana masuk ke escrow, dana tertahan dan tercatat dalam *etherscan*	0xf311b7815bdd9bc18a68f8af66d683b4d1d4bb5b619f7858eaeb510bfe9454c1	Berhasil
2	Admin Release Funds	Admin	Dana penjual, Admin kirim dana penjual tercatat *etherscan*	0x7daabc9871ea14091b6a6f167f267185acdb3058faa58ba91062d9b8b6147	Berhasil
3	Admin *Refund*	Admin	Dana pembeli, Admin *refund* dana pembeli tercatat *etherscan*	0x72a5a95a37f2e20c0bc6ce9b308a62ecb5f6e40150c79cb605752813a6203a	Berhasil

4. *Usability Testing*

Tabel 4. 48 Pertanyaan kuisioner

No	Pertanyaan
1	Antarmuka aplikasi mudah dipahami dan digunakan untuk melakukan transaksi
2	Proses registrasi dan *login* pada aplikasi berjalan dengan lancar dan cepat
3	Fitur transaksi (deposit, konfirmasi, dll.) pada aplikasi berfungsi dengan baik
4	Sistem escrow berbasis blockchain meningkatkan rasa aman dalam bertransaksi

5	Riwayat transaksi yang ditampilkan di aplikasi jelas dan transparan
6	Penerapan teknologi blockchain membuat saya lebih percaya diri dalam bertransaksi
7	Proses penyelesaian sengketa (dispute) pada aplikasi adil dan efisien
8	Aplikasi memberikan respons cepat saat melakukan transaksi atau interaksi
9	Informasi status transaksi (FUNDED, COMPLETED, DISPUTED) mudah dipahami
10	Secara keseluruhan, saya puas dengan pengalaman menggunakan aplikasi ini.

Seluruh pertanyaan dalam kuisioner ini memiliki indikator nilai yang dapat dilihat pada table 4.48.

Tabel 4. 49 Indikator Penilaian Kuisioner

Keterangan	Penilaian
Sangat Puas	5
Puas	4
Cukup	3
Tidak Puas	2
Sangat Tidak Puas	1

Hasil dari pengisian kuisioner dihitung menggunakan indikator kepuasan pengguna pada table 4.49.

Tabel 4. 50 Indikator Kepuasan Pengguna

Keterangan	Persentase
Sangat Puas	81 – 100 %
Puas	61 – 80 %
Cukup	41 – 60 %
Tidak Puas	21 – 40 %
Sangat Tidak Puas	1 – 20 %

Tabel 4. 51 hasil pengisian kuisioner

Nama	Jumlah Pertanyaan										Total Nilai
	1	2	3	4	5	6	7	8	9	10	
Darul Kaharman	4	5	5	5	5	5	4	4	4	4	45
Elawati Almauna	4	5	5	4	4	5	5	5	4	4	45

Anugrah	5	5	5	3	4	3	3	4	4	4	40
Ilham Idris	4	3	5	3	4	4	5	5	4	3	40
Maghfira Aulia R	4	5	3	4	4	5	5	4	4	4	42
A. Nurwafiq Azizah	5	5	3	4	3	4	4	5	5	5	43
Arifudin	4	4	4	5	5	4	4	5	4	4	43
Fajrin	4	4	5	5	3	3	5	5	3	5	42
Haris	4	3	4	4	3	4	4	4	5	4	39
Rina Amelia Sari	4	4	4	4	5	5	5	4	4	4	43
Fahmi	5	5	4	5	4	4	4	5	4	5	45
Dea Puspita	5	5	4	4	5	4	4	5	4	5	45
Abul A'la Al maududy	5	4	5	4	3	4	5	4	3	4	41
Nurul Azizah	4	4	4	3	5	5	5	5	4	5	44
Nur Ratna Sari	5	4	4	3	4	5	4	4	5	4	42
Erwin Gunawan	4	5	4	4	4	4	4	3	5	5	42
Ryan	4	5	5	5	4	3	5	4	4	4	43
Arsi Sanjaya	5	5	5	3	5	4	4	5	4	5	45
Nando Tri Armada	5	4	3	3	5	5	5	5	5	4	44
Olivia Damayanti	4	5	5	5	4	3	5	4	4	4	43
Jumlah Total Nilai											**856**

Jumlah nilai maksimal = 5 x 10 = 50

Jumlah responden = 20 orang

Jumlah kepuasan maksimal = 50 x 20 = 1.000

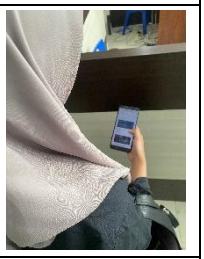
Hasil perhitungan kepuasan = $\frac{total\ nilai}{Jumlah\ kepuasan\ maksimal} x\ 100\%$

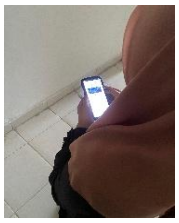
$= \frac{856}{1000} x\ 100\%$

$= 85{,}6\%$

Selain menghitung persentase kepuasan, analisis statistik deskriptif juga dilakukan untuk memperkuat interpretasi. Hasil menunjukkan rata-rata skor per responden setara dengan 85,6%. Dengan demikian, dapat disimpulkan bahwa sebagian besar responden merasa sistem yang dikembangkan mudah digunakan dan memberikan pengalaman transaksi yang transparan. Penambahan indikator statistik ini mempertegas bahwa tingkat kepuasan bukan hanya tinggi, tetapi juga stabil di antara responden.

Tabel 4. 52 Dokumentasi pengguna

No	Nama	Umur	Pendidkan	Pengguna	Dokumentasi
1	Darul Kaharman	23	S1 - Aktif	Menggunakan role admin	
2	Elawati Almauna	22	S1 - Aktif	Menggunakan role pembeli	
3	Anugrah Idam	29	S1 - Lulus	Menggunakan role penjual	
4	Ilham Idris	28	S1 - Lulus	Menggunakan role pembeli	
5	Maghfira Aulia Rahman	25	S1 – Aktif	Menggunakan role pembeli	
6	A. Nurwafiq Azizah	21	S1 – Aktif	Menggunakan role pembeli	

7	Arifudin	28	S1 – Lulus	Menggunakan role penjual	
8	Fajrin	20	S1 – Aktif	Menggunakan role pembeli	
9	Haris	30	S1 – Lulus	Menggunakan role penjual	
10	Rina Amelia Sari	26	S1 – Lulus	Menggunakan role penjual	
11	Fahmi	33	S1 – Lulus	Menggunakan role penjual	
12	Dea Puspita	31	S1- Lulus	Menggunakan role pembeli	

13	Abul A'la Al maududy	27	S1 – Lulus	Menggunakan role pembeli	
14	Nurul Azizah	24	S1- Lulus	Menggunakan role pembeli	
15	Nur Ratna Sari	19	S1 – Aktif	Menggunakan role pembeli	
16	Erwin Gunawan	35	SMA-Lulus	Menggunakan role pembeli	
17	Ryan	32	S1 – Lulus	Menggunakan role penjual	
18	Arsyi Sanjaya	16	SMA - Aktif	Menggunakan role pembeli	

19	Nando Tri Armada	17	SMA - Aktif	Menggunakan role penjual	
20	Olivia Damayanti	18	SMA - Aktif	Menggunakan role pembeli	

`

BAB V

PENUTUP

A. Kesimpulan

Penulis menyampaikan beberapa kesimpulan dari pembuatan *website* jual beli akun *game* menggunakan teknologi *blockchain* sebagai berikut:

1. *Website* penjualan akun *game online* yang dibangun menggunakan pemrograman seperti JavaScript, *solidity*, dan MySQL dan telah berhasil diuji menggunakan metode *blackbox, whitebox, fungsional* dan *usability*.
2. *Website* ini telah berhasil dengan menggunakan teknologi *blockchain* dengan *smart contract escrow*, agar terciptanya transaksi yang transparan. Sehingga meningkatkan kepercayaan pengguna dalam proses jual beli akun *game online*.
3. *Website* berhasil mensimulasikan sistem secara nyata dalam lingkungan yang menyerupai kondisi saat ini, menunjukkan bahwa sistem dapat berfungsi dengan baik secara operasional.

B. Saran

Penulis berharap kedepannya dapat menyempurnakan sistem jual beli akun *game* dengan penambahan beberapa token aset digital sebagai alat pembayaran transaksi dan membatasi keterbatasan *gas-fee* dengan rencana *mainnet* atau optimisasi biaya. Kemudian, menambahkan fitur *livechat* ke dalam *website* memudahkan para pengguna untuk berinteraksi agar *website* ini memberikan kenyamanan saat bertransaksi.

DAFTAR PUSTAKA

Aji, B. P., Wijayanto, H., & Mardiansyah, A. Z. (2023). *Implementasi Electronic Signature Pada Jaringan Privat Blockchain Implementation of Electronic Signature on the Ethereum Blockchain Private Network.*

Baiquni, M. F., & Dirgahayu, R. T. (2023). Aplikasi terdesentralisasi berbasis *blockchain* dan *smart contract* untuk pengelolaan zakat. *Automata*.

Buterin, V. (2014). *Ethereum: A Next-Generation Smart contract and Decentralized Application Platform. Ethereum* Foundation

EatTheBlocks. (2023). *Getting started with Hardhat smart contract framework*.

Ejournal UNSRAT. (2025). *Rancang bangun sistem e-voting menggunakan Hardhat blockchain*. Manado: Universitas Sam Ratulangi.

Fedianto, M. H. S., Aditiawan, F. P., & Al Haromainy, M. M. (2024). Pengujian Sistem Jaringan Dokumentasi Dan Informasi Menggunakan *Black box Testing* Dan *White box Testing*. *Jurnal Publikasi Sistem Informasi dan Manajemen Bisnis*, *3*(1), 213-221.

Hastuty, A. (2025). Hacking the future of college administration: Process transformation through AR/VR immersive technology. *EDUKASIA: Jurnal Pendidikan dan Pembelajaran, 6*(1), 95–106.

Iqbal, T., & Ahmad, L. (2024). Menerapkan *Blockchain* untuk Meningkatkan Transparansi dan Keamanan Rantai Pasokan: Studi Kasus di Industri Kelapa Sawit. *Jurnal Manajemen Dan Teknologi, 1*(1), 33–42.

Julianto, S., & Setiawan, S. (2019). Perancangan Sistem Informasi Pemesanan Tiket Bus Pada Po. Handoyo Berbasis *Online*. *Simatupang, Julianto Sianturi, Setiawan, 3*(2), 11–25.

Kristiansen, S. G., & Jensen, A. V. (2023). Victimization in online gaming-related trade scams: A study among young Danes. *Nordic Journal of Criminology, 24*(2), 1–17.

Kristiansen, S. G., Jensen, A. V., Nimgaard, M., & Ludvigsen, E. (2024). Virtual item trade scams: A typology of young victims. *International Review of Victimology* (advance online).

Kroons, A. A., & Dewi, C. (2023). Pengembangan dashboard Trivy berbasis website menggunakan *React* JS dan Golang. *Jurnal Indonesia: Manajemen Informatika dan Komunikasi*, 4(3), 1037–1049.

Kusumawati, R. D., & Aulawi, H. (2021). Peran Perceived Effectiveness of *Escrow* Services terhadap Satisfaction Jurnal Ekonomi Indonesia, 10(1), 13–24.

Kusyanti, A., Trisnawan, P. H., Brawijaya, U., Korespondensi, P., & Asuransi, K. (2023). *Implementation of Blockchain Technology Using Smart contract*. 10(7), 1563–1570.

Lisdayanti, A. (2024). *Peran Teknologi Blockchain dalam Meningkatkan Kepercayaan Konsumen dan Keamanan Data Privasi pada Platform E-Commerce di Indonesia. 5*(2), 347–361.

Mabruroh, A. M., Dewanta, F., & Wardana, A. A. (2021). Implementasi *Ethereum Blockchain* dan *Smart contract* Pada Jaringan Smart Energy Meter. *Multinetics*, *7*(1), 82–91.

Mulachela, A. (2020). Analisis Perkembangan Industri *Game* di Indonesia Melalui Pendekatan Rantai Nilai Global (Global Value Chain). *Indonesian Journal of Global Discourse, 2*(2), 32–51.

Nakamoto, S. (2008). *Bitcoin: A Peer-to-Peer Electronic Cash System.*

Newzoo. (2024). Global *game*s market report 2024. Diperoleh dari: https://newzoo.com/resources/blog/global-*game*s-market-revenue-estimates-and-forecasts-in-2024

Nasir, J., Hastuty, A., & Wafiah, A. (2025, July). Aplikasi layanan kesehatan di Posyandu berbasis website. *PROSISKO: Jurnal Pengembangan Riset dan Observasi Sistem Komputer*.

Nasution, E. Y., Hariani, P., Hasibuan, L. S., & Pradita, W. (2020). Perkembangan Transaksi Bisnis E-Commerce terhadap Pertumbuhan Ekonomi di Indonesia. *Jesya*, *3*(2), 506–519.

Negara, I. P. K., Adhi Putra, A. A. N. M., & Negara, I. B. K. D. S. (2021). Rancang bangun aplikasi e-wallet untuk jaringan *blockchain* Ethereum berbasis aplikasi mobile. *Journal of Informatics Engineering and Technology (JIETECH)*, 2(2), 11–22.

Rahardja, U., Aini, Q., Yusup, M., & Edliyanti, A. (2020). Penerapan Teknologi *Blockchain* Sebagai Media Pengamanan Proses Transaksi E-Commerce. *CESS (Journal of Computer Engineering, System and Science), 5*(1), 28.

ResearchGate. (2022). *Hardhat: Ethereum development framework for smart contracts*.

Rina Noviana. (2022). Pembuatan Aplikasi Penjualan Berbasis Web Monja Store Menggunakan Php Dan Mysql. *Jurnal Teknik Dan Science, 1*(2), 112–124

Rufus, C., & Purba, D. P. (2024). *Implementasi Teknologi Blockchain pada Real Estate Transaction.* 18221162

Saian, S. D. S., Sembiring, I., & Manongga, D. H. F. (2024). A prototype of decentralized applications population management system based on *blockchain* and *smart contract*. *JOIV: International Journal on Informatics Visualization*, 8(2), 845–853.

Sanka, A. I., Irfan, M., Huang, I., & Cheung, R. C. C. (2021). A survey of breakthrough in *Blockchain* technology: Adoptions, applications, challenges and future research. *Computer Communications, 169*, 179–201.

Sari, S. A., & Nasution, M. I. P. (2024). Implementasi Teknologi *Blockchain* Dalam Uang Digital: Potensi Dan Dampaknya terhadap Sistem Keuangan Global. *Jurnal Ilmiah Ekonomi Dan Manajemen, 2*(12), 259-267.

Satoshi Nakamoto. (2008). *Bitcoin: A Peer-to-Peer Electronic Cash System*. SSRN.

Soedewi, S. (2022). Penerapan Metode Design Thinking Pada Perancangan *Website* Umkm Kirihuci. *Visualita Jurnal Online Desain Komunikasi Visual, 10*(02), 17

Staroletov, S., & Galkin, D. (2023). *Testing methods for blockchain applications*.

Supiandi, G. (2024). Marketica: Jurnal Ilmiah Pemasaran. *Marketica: Jurnal Ilmiah Pemasaran*, *1*, 16–22.

Supianti, S. (2020). Analisis hukum ekonomi Islam tentang jual beli online (Studi di Jampue Kabupaten Pinrang). Skripsi tidak dipublikasikan. Parepare: Institut Agama Islam Negeri (IAIN) Parepare.

Suroyo, H., Dermawan, M. D., Wijaya, A., & Jemakmund. (2024). Pengembangan web admin aplikasi pulsa dengan *React*JS menggunakan metode Rapid Application Development. *Jurnal Teknologi dan Sistem Informasi Bisnis*, 6(2), 300–309.

Sutanti, A., MZ, M. K., Mustika, M., & Damayanti, P. (2020). Rancang Bangun Aplikasi Perpustakaan Keliling Menggunakan Pendekatan Terstruktur. *Komputa: Jurnal Ilmiah Komputer dan Informatika*, *9*(1), 1-8.

Wijaya, A. B., Lathif, T., Suryanto, M., Safitri, E. M., Harisno, H., Pujadi, T., sari puji rahayu, Laudon, K. C., Traver, C. G., Amalia, P., & Putra, Y. (2021). E-Commerce: Pasar Digital dan Barang Digital. *CommIT (Communication and Information Technology) Journal,* 3(2), 1–7.

Wintana, D., Pribadi, D., & Nurhadi, M. Y. (2022). Analisis Perbandingan Efektifitas White-Box *Testing* dan Black-Box *Testing*. *Jurnal Larik Ladang Artikel Ilmu Komputer*, *2*(1), 8-16.

Yogiyanti, E., & Suartana, I. M. (2024). Penerapan Teknologi *Blockchain* pada Sistem Laporan Keuangan Aplikasi Point of Sale. *Journal of Informatics and Computer Science, 06,* 179–188.

Yunitarini, R., Negara, Y. D. P., & Kamil, N. I. (2023). Aplikasi Rekening Bersama (Rekber) Terintegrasi Payment Gateway Menggunakan Metode Extreme Programming Dengan Arsitektur Monolitik. *Jurnal Sains, Nalar, Dan Aplikasi Teknologi Informasi, 3*(1), 29–41.