

# Pengembangan *E-marketplace In-game Currency* Menggunakan *Framework Laravel* dengan Metode *Extreme Programming*

<http://dx.doi.org/10.28932/jutisi.v7i2.3945>

Riwayat Artikel

Received: 31 Agustus 2021 | Final Revision: 25 November 2021 | Accepted: 25 November 2021

Daniel Apdianto Herman<sup>✉#1</sup>, Andy Kho<sup>\*2</sup>

<sup>#</sup> Program Studi Sistem Informasi, Universitas Internasional Batam  
Sei Ladi, Jl. Gajah Mada, Baloi Permai, Kec. Sekupang, Kota Batam, Kepulauan Riau 29442

<sup>1</sup>daniel.herman@uib.ac.id

<sup>\*</sup>Program Studi Sistem Informasi, Universitas Internasional Batam  
Sei Ladi, Jl. Gajah Mada, Baloi Permai, Kec. Sekupang, Kota Batam, Kepulauan Riau 29442

<sup>2</sup>1831089.andy@uib.edu

**Abstract** — In-game currency can be used to purchase various in-game items, especially in online games. In addition, in-game currency is also widely traded by players in certain online games. This eventually causes the potential for scam in buying and selling in-game currency to increase. This happened because there is no place that can be used by players to do transactions safely. The purpose of this research is to design and develop an e-marketplace for in-game currency using the Extreme Programming method which consists of planning, design, coding, and testing stages. This in-game currency e-marketplace application will be developed using the Laravel framework, which is one of the open-source PHP frameworks. The result of this research is an e-marketplace application that can be used by in-game currency buyers or sellers to do buying and selling easily and safely. With this e-marketplace, buying and selling in-game currency transactions will be easier, faster, and also scam-free.

**Keywords**— digital goods; e-marketplace; in-game currency.

## I. PENDAHULUAN

*Game online* merupakan permainan yang dapat dimainkan oleh banyak orang secara *online*. Ada *game online* yang memiliki cerita dan tantangan yang menarik, ada yang melibatkan emosi, dan ada juga yang terasa seperti terlibat langsung di dalamnya [1]. Pada *game online*, terdapat *virtual items* yang hanya dapat digunakan di dalam *game online* tersebut. *Virtual items* ini disebut sebagai *in-game items*. Dengan kata lain, *in-game items* bergantung pada *game online* tertentu. Keberadaan *in-game items* juga memiliki batas waktu, tergantung apakah *game online* tersebut masih aktif atau tidak. Salah satu *in-game items* yang terdapat pada hampir setiap *game online* adalah *in-game currency* [2].

*In-game currency* dapat digunakan untuk membeli berbagai *in-game items*. Karena itu, transaksi jual beli *in-game currency* menjadi marak terjadi. Baik transaksi langsung dengan *game online* tersebut, maupun transaksi dengan pemain lainnya. Namun, tentu saja hal ini menyebabkan potensi terjadinya penipuan semakin meningkat [3]. Salah satu contohnya adalah pada *game online* Growtopia. Pada *game online* ini, sangat banyak pemain yang melakukan transaksi jual beli *in-game currency*. Hal ini dikarenakan *in-game currency* pada *game online* ini cukup mudah didapatkan dan harganya juga lumayan tinggi. Akibatnya, penipuan di *game online* ini marak terjadi. Salah satu contoh kasus penipuan *game online* ini terjadi di Singapura. Pada kasus tersebut, seorang remaja berumur 16 tahun ditangkap karena menipu pemain lainnya dalam hal transaksi *in-game currency* pada *game online* Growtopia [4].

Penelitian yang dilakukan oleh Tristiyanto, Yunda Heningtyas, dan Hanan Risnawati adalah salah satu penelitian yang menjadi dasar utama dari penelitian ini. Pada penelitian tersebut, dirancang dan dikembangkan sebuah *e-marketplace* penyewaan untuk koperasi dengan tujuan untuk mempertemukan penyewa dan pemilik barang sehingga transaksi penyewaan dapat dilakukan dengan lebih efektif. Menurutnya, proses penyewaan yang dilakukan secara konvensional sering kali menyulitkan penyewa karena penyewa harus mengunjungi pemilik barang secara langsung untuk mengetahui informasi mengenai barang yang ingin disewa. Metode yang digunakan dalam penelitian tersebut adalah metode *Extreme*

*Programming* yang memiliki empat tahapan, yaitu *planning*, *design*, *coding*, dan *testing*. Pada akhir penelitian tersebut, berhasil dibangun sebuah *e-marketplace* penyewaan untuk koperasi di mana hasil yang didapatkan dari penelitian tersebut juga berhasil sesuai dengan yang diinginkan. Berdasarkan penelitian tersebut, didapatkan referensi mengenai metode dan cara perancangan serta pengembangan sebuah *e-marketplace* [5].

Penelitian lainnya adalah penelitian yang dilakukan oleh Erwin Yulianto dan Fajar Nurochim. Menurut penelitian tersebut, potensi ekonomi dari peluang usaha barang digital sangatlah tinggi, namun hanya sedikit masyarakat Indonesia yang sudah mengetahui hal tersebut. Karena itulah dalam penelitian tersebut, dibangun sebuah *e-marketplace* barang digital yang dapat mempermudah masyarakat, khususnya para talenta digital di Indonesia dalam mencari, menjual, maupun membeli produk-produk digital. Metode dalam penelitian tersebut dilakukan dengan menggunakan pendekatan *Case Study Research*. Kemudian, penelitian tersebut menggunakan metode *Software Development Life Cycle* untuk mengembangkan aplikasi *e-marketplace* barang digital tersebut, yaitu dimulai dengan pemetaan kebutuhan pengguna, lalu berlanjut ke tahapan perencanaan, pemodelan, konstruksi, dan penyerahan sistem ke pengguna. Pada akhir penelitian tersebut, berhasil dikembangkan sebuah *e-marketplace* barang digital yang mampu mempertemukan penjual dan pembeli barang digital serta mengkampanyekan barang digital di Indonesia sebagai peluang usaha baru yang potensial sesuai dengan tujuan dari penelitian tersebut. Berdasarkan penelitian tersebut, didapatkan referensi mengenai potensi dari usaha jual beli barang digital, terutama di Indonesia [6].

*E-marketplace* adalah salah satu sarana untuk mawadahi penjual dan pembeli dalam melakukan transaksi jual beli secara *online*. Alasan mengapa orang-orang lebih memilih untuk menggunakan *e-marketplace* dalam bertransaksi, antara lain bisa berkomunikasi secara langsung, adanya rekening bersama, dan juga transaksi yang lebih aman. Dengan memanfaatkan *e-marketplace*, penjual dan pembeli akan sangat terbantu. Hal ini dikarenakan tidak adanya batasan jarak dan waktu untuk saling berinteraksi dan bertransaksi [7]. Setiap *e-marketplace* memiliki bidangnya masing-masing. Kebanyakan dari *e-marketplace* yang sudah ada saat ini bergerak di bidang komoditi barang fisik. Namun seiring berjalannya waktu, kebutuhan akan barang non-fisik semakin meningkat, terutama barang digital seperti *in-game currency* [6].

Untuk merancang dan membangun aplikasi *e-marketplace in-game currency* ini, digunakan *framework* Laravel dengan metode *Extreme Programming*. *Framework* Laravel adalah *framework* PHP *open-source* yang menerapkan konsep MVC (*Model*, *View*, & *Controller*) yang banyak digunakan untuk mengembangkan aplikasi berbasis web. *Framework* Laravel pertama kali dikembangkan oleh Taylor Otwell pada tahun 2012.

Model MVC yang terdapat pada *framework* Laravel merupakan model pengembangan dengan memisahkan *model* atau data, *view* atau tampilan, dan *controller* atau pemroses. *Model* berperan untuk menghubungkan *controller* dengan *database*. Seluruh *query* yang ingin dilakukan akan diatur oleh *model* untuk kemudian dikirimkan ke *database*. *View* berperan untuk menampilkan data sesuai dengan yang diinginkan. *View* adalah bagian yang berinteraksi langsung kepada pengguna. Dan *controller* berperan untuk menghubungkan antara *model* dengan *view*, di mana fungsi dari *controller* adalah memproses perintah yang telah dibuat untuk menentukan proses berjalannya aplikasi. Dengan menggunakan *framework* Laravel, komponen pemrograman dapat menjadi *reusable*. Standar penulisan kode program pada *framework* Laravel juga wajib diikuti oleh setiap orang yang ada dalam tim, sehingga akan mempermudah dan mempercepat pengembangan aplikasi. Di samping itu, *framework* Laravel juga dapat memberikan keamanan dari serangan eksternal [8].

Selanjutnya, metode *Extreme Programming* adalah metode pengembangan perangkat lunak yang dapat mempercepat pembangunan suatu sistem, menyederhanakan tahapan pengembangan, dan mengutamakan fleksibilitas terhadap perubahan yang terjadi. *Extreme Programming* memiliki empat tahapan, yaitu *planning*, *design*, *coding*, dan *testing* [9]. Fakta bahwa metode ini disebut sebagai “*Extreme Programming*” juga merupakan keunggulan lain dari metode *Extreme Programming* ini, di mana metode ini merupakan metode ekstrim yang sangat membantu dalam mengembangkan aplikasi berkualitas tinggi dengan tingkat kepuasan konsumen yang tinggi, umpan balik yang cepat, dan biaya produksi yang lebih rendah. Selain hal tersebut, metode *Extreme Programming* juga menerapkan sistem *pair programming* yang dapat mempercepat dan meminimalisir terjadinya kesalahan dalam proses pengembangan aplikasi. Berikutnya, metode *Extreme Programming* juga memiliki lima nilai yang difokuskan, yaitu *simplicity*, *communication*, *feedback*, *courage*, dan *respect* [10]. Dengan menerapkan *pair programming* dan kelima nilai dalam metode *Extreme Programming*, maka proses pengembangan aplikasi akan menjadi jauh lebih cepat, efisien, dan berkualitas.

Berdasarkan latar belakang tersebut, maka akan dilakukan perancangan dan dikembangkan sebuah aplikasi *e-marketplace* untuk *in-game currency* berbasis *website*, yang dapat mempermudah proses transaksi antara penjual dan pembeli, terutama dalam hal transaksi *in-game currency*. *E-marketplace* ini akan memiliki beberapa fitur, seperti *market* dan *wallet*, serta dilengkapi juga dengan panel admin yang akan membantu proses yang berjalan pada *e-marketplace* ini.

## II. METODE PENELITIAN

Perancangan dan pengembangan aplikasi ini menggunakan metode *Extreme Programming*. *Extreme Programming* dikembangkan oleh Kent Beck pada tahun 1996 saat sedang mengembangkan aplikasi penggajian. Dalam bukunya yang berjudul “*Extreme Programming Explained*”, dijelaskan bahwa metode *Extreme Programming* merupakan sebuah metode

yang sangat ringan, fleksibel, dan memiliki resiko yang rendah, serta memiliki kemampuan untuk mengubah kebutuhan dari sistem dengan cepat [10].

Menurut Irmawati Carolina dan Adi Supriyatna, Extreme Programming (XP) merupakan sebuah proses rekayasa perangkat lunak yang cenderung menggunakan pendekatan berorientasi objek dan sasaran dari metode ini adalah tim yang dibentuk dalam skala kecil sampai medium serta metode ini juga sesuai jika tim dihadapkan dengan requirement yang tidak jelas maupun terjadi perubahan-perubahan requirement yang sangat cepat [11]. Dengan menggunakan metode *Extreme Programming*, proses perancangan dan pengembangan aplikasi ini akan menjadi lebih cepat dan mudah.

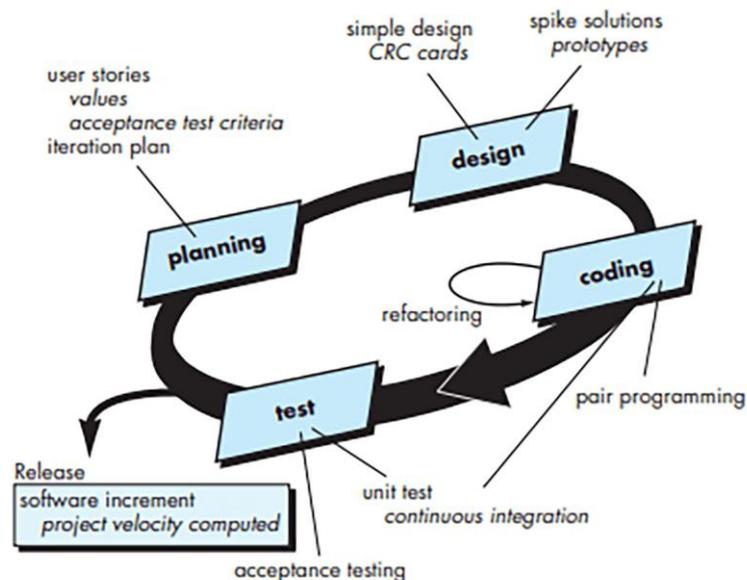
Metode *Extreme Programming* juga cocok digunakan untuk tim yang berukuran kecil hingga sedang. Selain itu, fakta bahwa metode ini disebut sebagai "*Extreme Programming*" juga merupakan keunggulan lain dari metode *Extreme Programming* ini, di mana metode ini merupakan metode ekstrim yang sangat membantu dalam mengembangkan aplikasi berkualitas tinggi dengan tingkat kepuasan konsumen yang tinggi, umpan balik yang cepat, dan biaya produksi yang lebih rendah [10].

Oleh karena itu, metode *Extreme Programming* dapat dikatakan cocok untuk digunakan pada perancangan dan pengembangan aplikasi *e-marketplace* ini, di mana untuk sekarang tim ini hanya beranggotakan dua orang. Selain itu, fleksibel terhadap perubahan-perubahan yang mungkin terjadi, juga dibutuhkan dalam perancangan dan pengembangan aplikasi yang sangat memungkinkan terjadinya perubahan-perubahan kebutuhan ini, baik secara mendadak ataupun tidak.

Pada metode *Extreme Programming* juga terdapat lima nilai yang difokuskan, yaitu *simplicity, communication, feedback, courage, dan respect*. Pada perancangan dan pengembangan aplikasi ini, *simplicity* dilakukan dengan tidak menambahkan fitur-fitur di luar kebutuhan pengguna. Selain itu, penggunaan *framework* dan *package* juga membantu dalam menyederhanakan penulisan *code* agar tidak banyak yang perlu dibuat secara manual. Metode *Extreme Programming* juga menekankan nilai *communication* pada pengguna. Pada pengembangan aplikasi ini, beberapa pengguna dan para pengembang akan bersama-sama akan melihat proses pengembangan dan memberikan saran secara langsung apabila ada yang terasa kurang, hal ini tentunya untuk mengurangi terjadinya ketidaksesuaian kebutuhan pengguna.

Selain *communication, feedback* dari pengguna juga penting, di mana pengguna akan melakukan pengujian secara langsung dan memberikan umpan balik mengenai aplikasi yang telah dikembangkan. Berikutnya, *courage* merupakan keberanian untuk mengambil keputusan dalam pengembangan aplikasi, contohnya seperti mengubah *coding* secara keseluruhan. Dan terakhir, *respect* merupakan sikap saling menghormati satu sama lain, baik pada pengguna maupun sesama tim.

Pada metode *Extreme Programming*, juga akan dilakukan *iteration to release phase*, yaitu sebuah metode di mana setiap tahapan dalam metode *Extreme Programming*, yaitu *planning, design, coding, dan testing* akan terus diiterasi atau diulangi berkali-kali hingga hasil akhir yang didapatkan sesuai dengan keinginan. Masing-masing iterasi memiliki aktivitas yang berbeda-beda, sesuai dengan *iteration planning* yang telah direncanakan. [10]. Model dari metode *Extreme Programming* dapat dilihat pada Gambar 1.



Gambar 1. Metode Extreme Programming [12]

Berikut ini adalah tahapan dari metode *Extreme Programming* yang digunakan pada penelitian ini:

#### A. Planning

Tahapan *planning* dimulai dengan mengumpulkan kebutuhan pengguna, lalu membuat *user stories* yang akan menggambarkan *output*, fitur, dan fungsionalitas yang diperlukan. Pengguna akan memberikan nilai prioritas ke *user stories* berdasarkan tingkat kebutuhan dari fitur. *User stories* akan direpresentasikan dengan pemodelan *diagram Unified Modeling Language* yang terdiri dari *Use Case Diagram*, *Activity Diagram*, dan *Entity Relationship Diagram*. *User stories* akan digunakan sebagai dasar dari perancangan dan pengembangan aplikasi e-marketplace ini [13].

#### B. Design

*Design* merupakan tahapan kedua dari metode *Extreme Programming*. Pada tahapan ini, akan dilakukan perancangan desain diagram sistem, seperti *use case diagram*, *activity diagram*, serta *entity relationship diagram* dan desain antarmuka sistem berupa *mockup* berdasarkan *user stories* yang telah ada. Desain yang dirancang pada tahapan ini nantinya akan digunakan pada tahapan *coding* [11].

#### C. Coding

Selanjutnya, pada tahapan *coding*, aplikasi e-marketplace ini akan mulai dikembangkan. Pengembangan dilakukan dengan menerjemahkan desain diagram sistem dan *mockup* yang telah dirancang pada tahapan sebelumnya ke dalam bahasa pemrograman. Penulisan kode program pada penelitian ini dilakukan dengan menggunakan bahasa pemrograman PHP, dengan *framework* Laravel [5]. Proses pengkodean akan menggunakan *text editor* Visual Studio Code dengan *internet browser* Opera GX serta *development server* Laragon.

Pada tahapan *coding*, juga digunakan *pair programming*, yaitu cara pemrograman di mana dua *programmer* bekerja pada satu komputer yang sama. Dengan begitu, maka biaya produksi dapat ditekan serendah mungkin dan kualitas aplikasi yang dikembangkan dapat maksimal dikarenakan setiap *error* yang muncul dapat dengan mudah diketahui dan diperbaiki oleh *programmer* yang satunya [10].

Dengan menggunakan *pair programming* ini, pengembangan aplikasi dapat didampingi oleh pengguna secara langsung sehingga pengguna dapat terlibat langsung dalam pemrograman sambil berkomunikasi dengan pengembang aplikasi. Tujuan dari hal ini adalah untuk membuat pandangan dari pengembang aplikasi sesuai dengan pandangan dari pengguna sistem.

#### D. Testing

Terakhir, setelah proses *coding* selesai, akan dilakukan tahapan *testing*. Tahapan *testing* atau pengujian pada metode *Extreme Programming* berfokus pada keseluruhan fungsi dan fitur dari sistem yang dapat digunakan oleh pengguna berdasarkan *user stories* yang ada. Metode pengujian yang digunakan pada penelitian ini adalah metode *Black-box Testing*. Kemudian, metode *Black-box Testing* yang digunakan adalah metode *Equivalence Partitioning*, yaitu membagi masukan dari program ke dalam kelas-kelas terpisah sehingga dapat diperoleh kasus uji [14].

### III. HASIL DAN PEMBAHASAN

Hasil dari penelitian ini berupa sebuah aplikasi *e-marketplace in-game currency* yang dirancang dan dikembangkan menggunakan *framework* Laravel dengan metode *Extreme Programming*. Aplikasi *e-marketplace* ini dapat mempermudah pelaku transaksi jual beli *in-game currency* dalam melakukan penjualan maupun pembelian *in-game currency* tertentu dengan mudah dan aman. Selain itu, terdapat juga fitur lainnya seperti *wallet* yang dapat digunakan untuk menyimpan atau mengirimkan *in-game currency* yang telah di-deposit-kan. Pengguna juga dapat melakukan *withdraw* terhadap *in-game currency* yang dimiliki ke akun *game online* masing-masing.

Pada perancangan dan pengembangan aplikasi *e-marketplace in-game currency* ini, proses iterasi yang dilakukan hanya sebatas iterasi pertama, karena selain untuk mempersingkat waktu, aplikasi ini sudah cukup baik dikembangkan untuk versi awalnya walau hanya dengan iterasi pertama. Selain itu, penggunaan *framework* Laravel dan berbagai *package* juga dilakukan untuk mempercepat proses pengembangan aplikasi ini. Berikut hasil perancangan dan pengembangan aplikasi serta hasil pengujian yang telah dilakukan berdasarkan metode *Extreme Programming*:

#### A. Planning

Pada tahapan *planning*, dilakukan analisis terhadap kebutuhan pengguna berdasarkan hasil wawancara dengan beberapa penjual maupun pembeli *in-game currency* yang ada di Facebook. Berdasarkan hasil wawancara tersebut, didapatkan berbagai informasi mengenai apa saja yang dibutuhkan oleh pengguna untuk *e-marketplace in-game currency* yang akan dikembangkan. Dari hasil wawancara tersebut, juga diketahui tingkat kebutuhan pengguna terhadap fitur berdasarkan *user stories* yang ada [13]. Pada Tabel 1 ditunjukkan beberapa *user stories* yang telah didapatkan.

TABEL 1  
USER STORIES

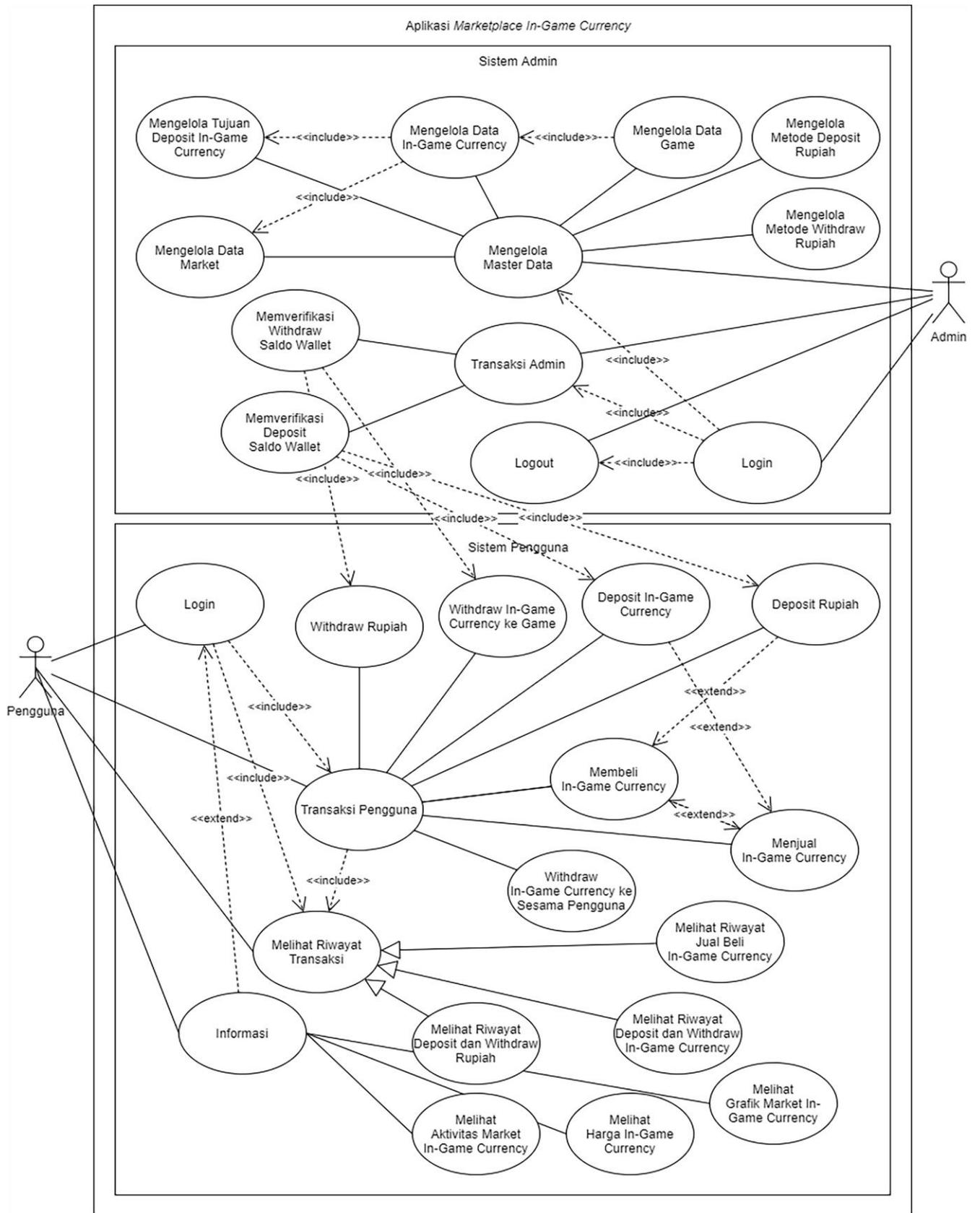
No.	Deskripsi	Prioritas
1	Sebagai pengguna, <i>login</i> atau <i>register</i> diperlukan sebelum menggunakan fitur-fitur di <i>e-marketplace</i> , sehingga data pribadi, rupiah, serta <i>in-game currency</i> dapat terjamin aman.	Tinggi
2	Sebagai pengguna, ingin bisa melakukan pembelian dan penjualan <i>in-game currency</i> dengan mudah dan aman, sehingga tidak perlu kesulitan dan takut tertipu.	Tinggi
3	Sebagai pengguna, ingin bisa melakukan pembelian dan penjualan <i>in-game currency</i> dengan harga tertentu, sehingga bisa mendapatkan harga terbaik.	Tinggi
4	Sebagai pengguna, ingin bisa membeli dan menjual <i>in-game currency</i> dengan harga pasar, sehingga bisa dengan cepat mendapatkan hasil pembelian atau penjualan.	Tinggi
5	Sebagai pengguna, ingin bisa mengirimkan <i>in-game currency</i> ke orang lain dengan mudah, sehingga tidak perlu repot-repot harus <i>online</i> di dalam <i>game</i> terlebih dahulu.	Sedang

### B. Design

Berikutnya, pada tahapan *design*, dilakukan pembuatan diagram-diagram sistem dalam bentuk *use case diagram*, *activity diagram*, serta *entity relationship diagram* dengan menggunakan draw.io dan desain antarmuka sistem dalam bentuk *mockup* dengan menggunakan Balsamiq Wireframes.

Pada tahapan ini, juga dilakukan komunikasi secara terus menerus dengan para pengguna mengenai desain sistem, baik itu diagram sistem maupun desain antarmuka sistem. Dengan begitu, didapatkan hasil desain yang cukup baik dan sesuai dengan kebutuhan dari pengguna. Pembuatan desain, baik *mockup* maupun diagram-diagram sistem ini juga dibuat agar mengacu pada *user stories* yang telah ditentukan sebelumnya [5].

- 1) *Use Case Diagram*: *Use case diagram* digambarkan dengan mengacu pada *user stories* yang telah dibuat sebelumnya. Dengan menggunakan *use case diagram*, dapat digambarkan apa saja yang dapat dilakukan oleh pengguna, baik pengguna biasa maupun admin dengan jelas. Admin dapat mengelola seluruh *master data* yang dibutuhkan pada aplikasi ini serta melakukan transaksi admin, seperti verifikasi *deposit* dan *withdraw* saldo *wallet*. Untuk melakukan hal tersebut, tentunya admin perlu melakukan *login* terlebih dahulu ke dalam panel admin yang telah disediakan. Selanjutnya, dari sisi pengguna, pengguna dapat melakukan berbagai macam bentuk transaksi, seperti *deposit* dan *withdraw* rupiah maupun *in-game currency*. Pengguna juga dapat membeli serta menjual *in-game currency* sesuai dengan market yang tersedia. Hasil dari penjualan maupun pembelian tersebut juga dapat ditarik atau di-*withdraw* sesuai dengan kebutuhan. Tentunya setelah melakukan *deposit* atau *withdraw*, admin harus memverifikasi *deposit* atau *withdraw* tersebut sebelum *deposit* atau *withdraw* tersebut dapat diproses. Untuk melakukan berbagai macam fitur transaksi yang disediakan tersebut, pengguna harus melakukan *login* terlebih dahulu ke dalam sistem, apabila pengguna tidak melakukan login, maka pengguna hanya dapat melihat informasi-informasi, seperti harga *in-game currency*, aktivitas *market*, dan grafik yang tersedia. Untuk lebih jelasnya, pada Gambar 2, ditunjukkan *use case diagram* dari aplikasi *e-marketplace in-game currency* ini. Sedangkan pada Tabel 2, ditunjukkan beberapa deskripsi singkat dari *use case diagram* pada Gambar 2 yang akan digunakan untuk pengujian nantinya.

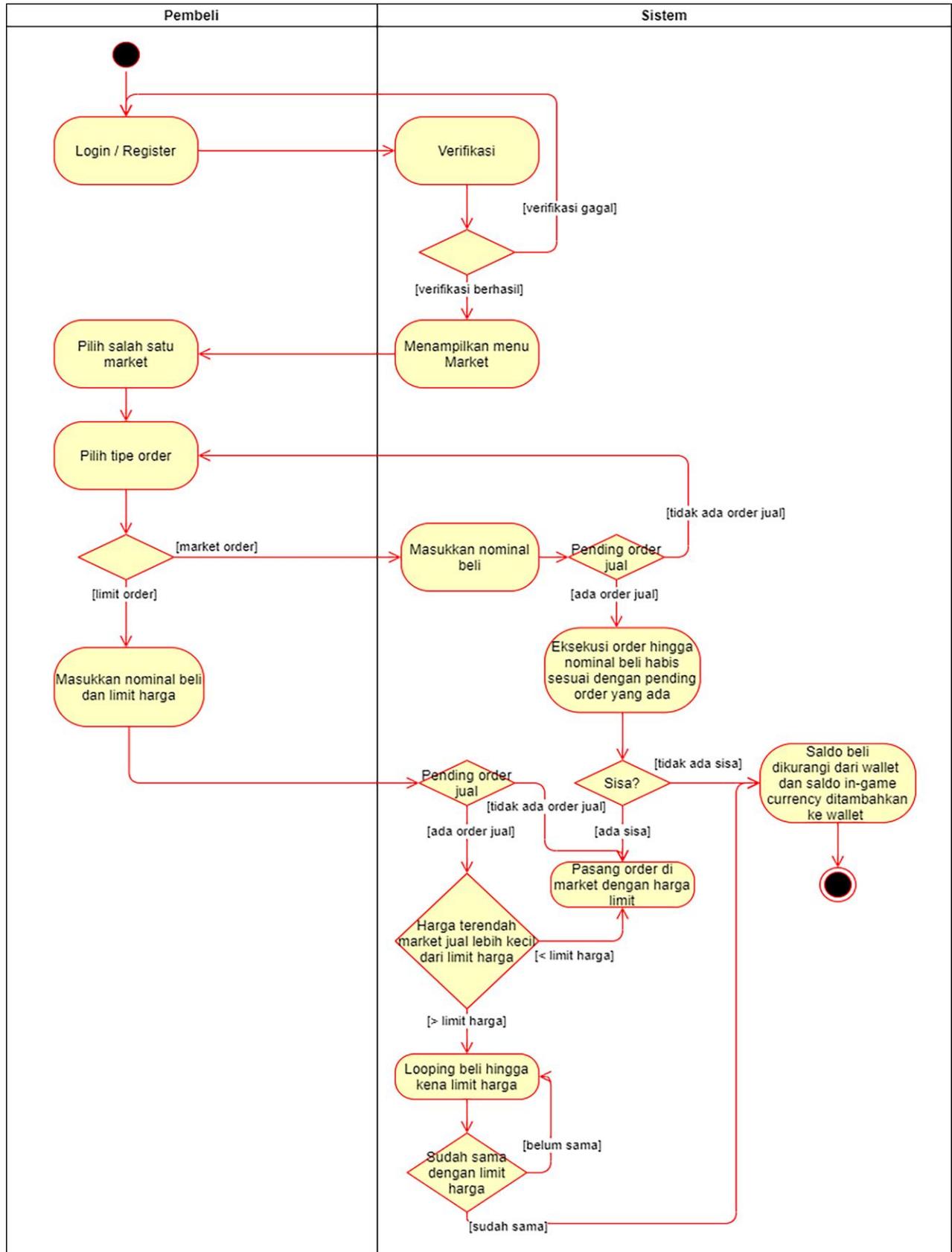


Gambar 2. Use Case Diagram

TABEL 2  
DESKRIPSI USE CASE

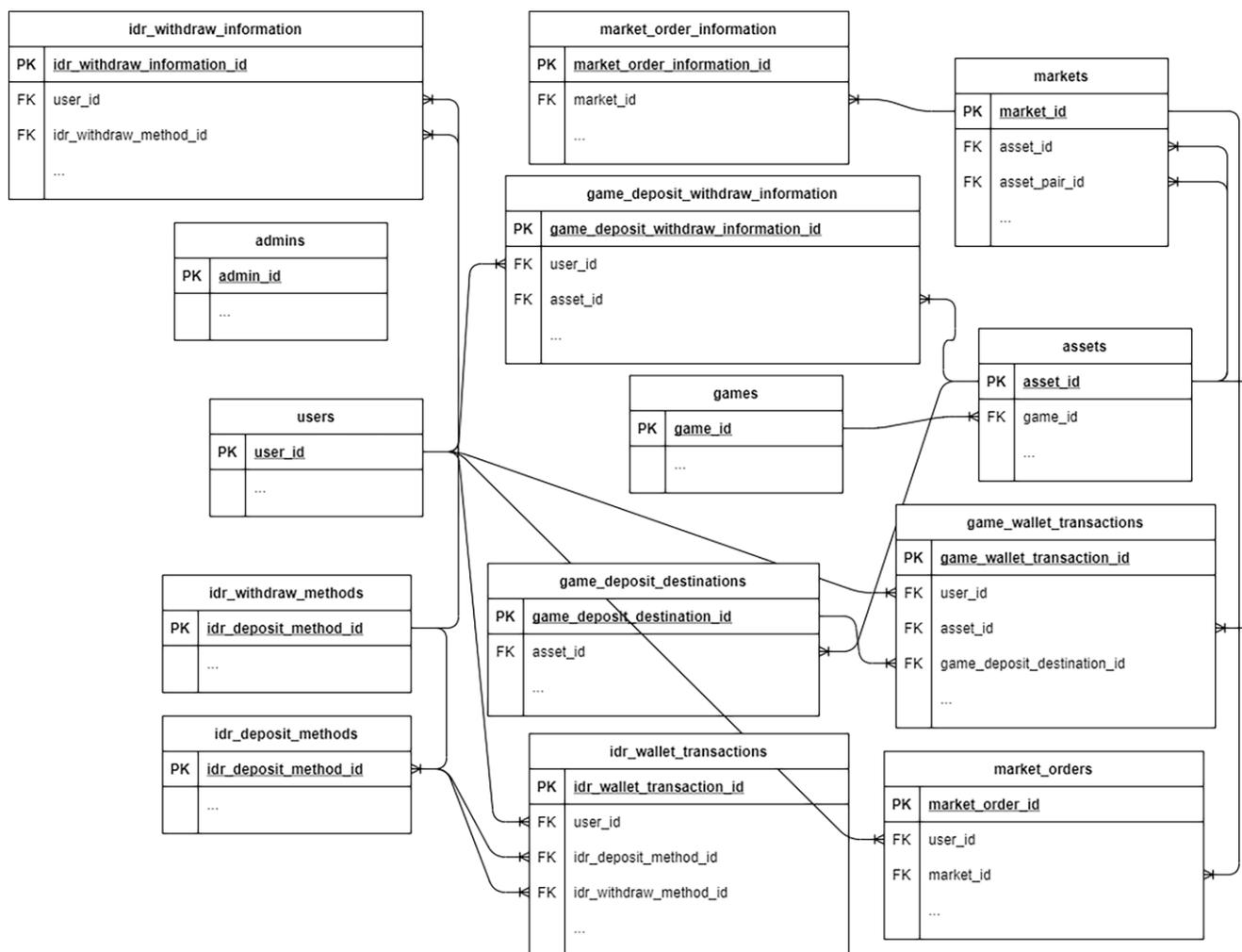
Use Case	Deskripsi
Register	Mendaftarkan diri ke dalam <i>e-marketplace</i> dengan cara memasukkan data diri, seperti nama lengkap, <i>username</i> , <i>email</i> , nomor <i>handphone</i> , dan <i>password</i> .
Login	Memasukkan <i>username</i> atau <i>email</i> dan <i>password</i> yang telah terdaftar untuk mendapat akses ke fitur yang telah disediakan.
Deposit In-game Currency	Melakukan <i>deposit in-game currency</i> melalui <i>game</i> yang terkait agar dapat digunakan untuk dijual atau dikirim lagi ke sesama pengguna.
Membeli In-game Currency	Melakukan pembelian <i>in-game currency</i> pada <i>market</i> yang telah disediakan. Pembelian dapat menggunakan tipe <i>limit</i> , yaitu membeli dengan batas harga tertentu maupun <i>market</i> , yaitu membeli langsung dari pasar.

- 2) *Activity Diagram*: *Activity diagram* adalah diagram yang menggambarkan kegiatan atau aktivitas yang terjadi antar aktor dengan sistem. *Activity diagram* dirancang dengan mengacu pada *use case* yang telah dirancang sebelumnya. Dengan adanya *activity diagram*, aktivitas yang dilakukan oleh pengguna dari sistem dapat terlihat dengan jelas. Contohnya, pada aktivitas pembelian *in-game currency*, pengguna terlebih dahulu harus melakukan *login* jika sudah punya akun atau *register* jika belum memiliki akun, jika pengguna melakukan *register*, maka pengguna harus melakukan verifikasi *email* terlebih dahulu sebelum akunnya aktif. Setelah sudah berhasil *login* atau *register*, selanjutnya pengguna akan masuk ke halaman *market*. Pada halaman ini, pengguna dapat melihat dan memilih salah satu *market* yang diinginkan serta memilih jenis *order* yang diinginkan sesuai dengan kebutuhan, yaitu *limit order* atau *market order*. *Limit order* akan mengeksekusi *order* sesuai dengan harga yang diinginkan dan sesuai dengan *order* yang tersedia di *market*. Apabila harga yang diinginkan tidak terdapat pada *order* yang tersedia, maka *limit order* akan secara otomatis membuat *order* baru di *market* dengan harga yang diinginkan tersebut. Sedangkan *market order* akan mengeksekusi *order* secara langsung berdasarkan *order* yang tersedia di *market* sesuai dengan berapa nominal beli yang diinginkan. Dengan menggunakan *market order*, *order* akan dieksekusi hingga *order* yang tersedia atau nominal beli dari pengguna habis. Setelah itu, maka saldo rupiah dari pengguna akan terpotong dan *in-game currency* akan secara otomatis masuk ke dalam *wallet* pengguna. Aktivitas pembelian *in-game currency* ini dapat dilihat pada Gambar 3.



Gambar 3. Activity Diagram Pembelian In-game Currency

- 3) *Entity Relationship Diagram*: *Entity relationship diagram* menggambarkan struktur *database* yang akan digunakan pada aplikasi *e-marketplace* ini. Desain *database* pada *entity relationship diagram* dirancang berdasarkan *use case* dan *activity* yang dibutuhkan oleh pengguna. *Entity relationship diagram* ini juga menggambarkan hubungan atau relasi antar tabel di dalam *database* agar dapat digunakan secara maksimal saat proses pengkodean nantinya. Tabel *admins* merupakan tabel yang berisi data admin yang akan digunakan oleh admin untuk *login* ke dalam panel admin. Kemudian, tabel *users* merupakan tabel yang berisi data pengguna yang sudah mendaftarkan diri ke dalam *e-marketplace* ini. Tabel *users* juga memiliki relasi *one to many* ke beberapa tabel, yaitu tabel *market\_orders*, *game\_deposit\_withdraw\_information*, *idr\_withdraw\_information*, *game\_wallet\_transactions*, dan *idr\_wallet\_transactions*. Relasi *one to many* dipilih karena satu pengguna dapat memiliki lebih dari satu informasi serta transaksi *deposit* maupun *withdraw in-game currency* dan rupiah. Selanjutnya, tabel *games* berisi data mengenai *game* yang aset atau *in-game currency*-nya akan diperdagangkan pada *e-marketplace in-game currency* ini. Tabel *games* juga memiliki relasi *one to many* ke tabel *assets* yang berisi data *in-game currency* tersebut. Relasi tersebut dipilih dikarenakan satu *game* dapat memiliki lebih dari satu *in-game currency*. Berikutnya, tabel *assets* juga berelasi *one to many* ke tabel *markets* karena satu *in-game currency* dapat memiliki lebih dari satu *market*, misalnya *market in-game currency A* dengan rupiah dan *market in-game currency A* dengan *in-game currency B*. Tabel *markets* berelasi *one to many* ke tabel *market\_order\_information* dan *market\_orders*. Tabel *market\_order\_information* adalah tabel yang berisi data mengenai informasi *order* dari *market* tertentu. Pada tabel ini terdapat kolom *open\_price*, *close\_price*, *low\_price*, *high\_price*, dan waktu data tersebut disimpan yang akan digunakan untuk keperluan pembentukan grafik nantinya dan akan diperbarui setiap 15 menit secara otomatis. Kemudian, tabel *market\_orders* berisi data *order* dari pengguna, baik *order* jual maupun beli dari pengguna. Salah satu kolom dari tabel ini adalah *executed\_amount*, di mana kolom ini berisi nominal jual atau beli yang sudah tereksekusi dan apabila masih belum tereksekusi sepenuhnya, maka *order* tersebut masih akan berstatus *pending*. Terdapat juga tabel *idr\_deposit\_methods* dan *game\_deposit\_destinations* yang berisi data-data yang diperlukan untuk keperluan *deposit* rupiah maupun *in-game currency* ke dalam aplikasi *e-marketplace* ini. Tabel *idr\_deposit\_methods* berisi data metode-metode *deposit* yang dapat digunakan oleh pengguna beserta rekening bank dari *e-marketplace* ini (jika tidak menggunakan *payment gateway*) untuk melakukan *deposit* rupiah. Dan untuk tabel *game\_deposit\_destinations* berisi data tujuan *deposit* dalam *game* atau akun *game* untuk *deposit* yang perlu pengguna kirimkan *in-game currency* agar *deposit in-game currency* dapat diproses. Kemudian, pada tabel *game\_deposit\_withdraw\_information* berisi data *game* yang dimiliki oleh pengguna untuk melakukan *deposit* atau *withdraw in-game currency*. Sedangkan tabel *idr\_deposit\_withdraw\_information* berisi data bank atau *e-wallet* yang dimiliki oleh pengguna untuk melakukan *deposit* atau *withdraw* rupiah. Berikutnya, tabel *game\_wallet\_transactions* berisi data *deposit* maupun *withdraw in-game currency* yang telah dilakukan oleh pengguna. Dan tabel *idr\_wallet\_transactions* tabel yang berisi data *deposit* maupun *withdraw* rupiah yang telah dilakukan oleh pengguna. Pada tabel *game\_wallet\_transactions* dan *idr\_wallet\_transactions* juga terdapat kolom *is\_internal* yang akan terisi apabila *deposit* atau *withdraw* yang dilakukan oleh pengguna bersifat internal atau dilakukan di dalam aplikasi *e-marketplace* ini, yaitu *deposit* atau *withdraw* antar pengguna. Untuk lebih lengkapnya mengenai gambaran dari struktur *database* aplikasi *e-marketplace* ini, dapat dilihat hasil perancangan *entity relationship diagram* yang ditunjukkan pada Gambar 4.

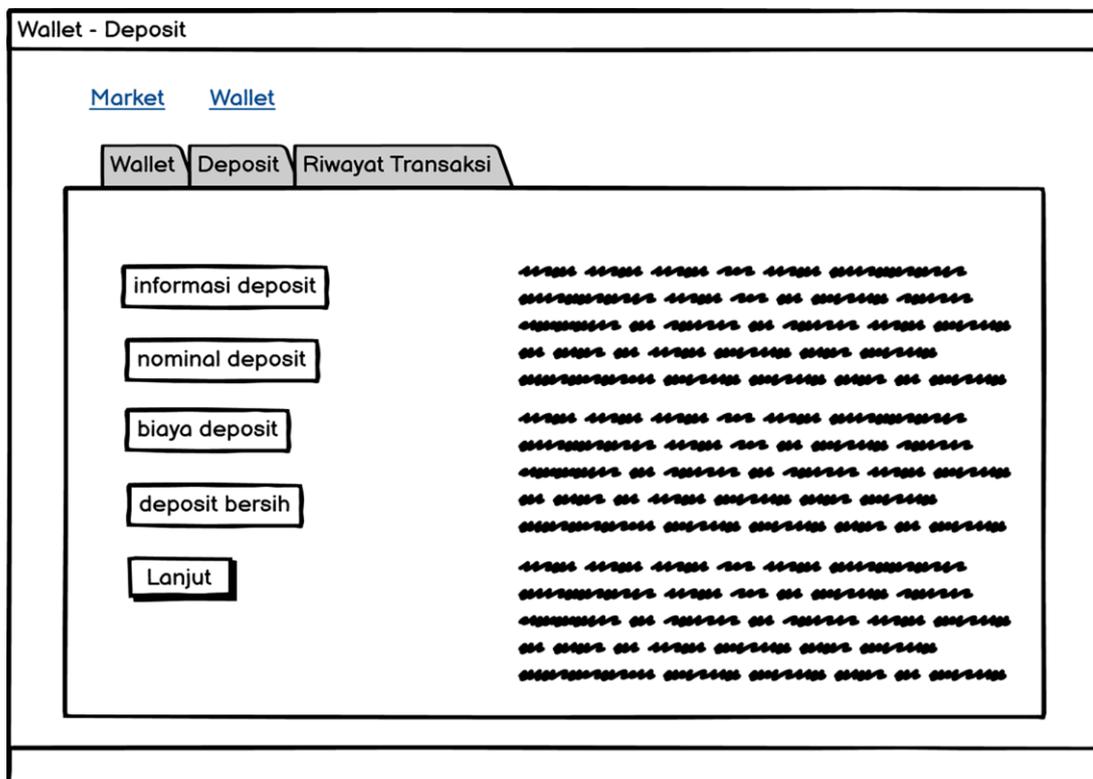


Gambar 4. Entity Relationship Diagram

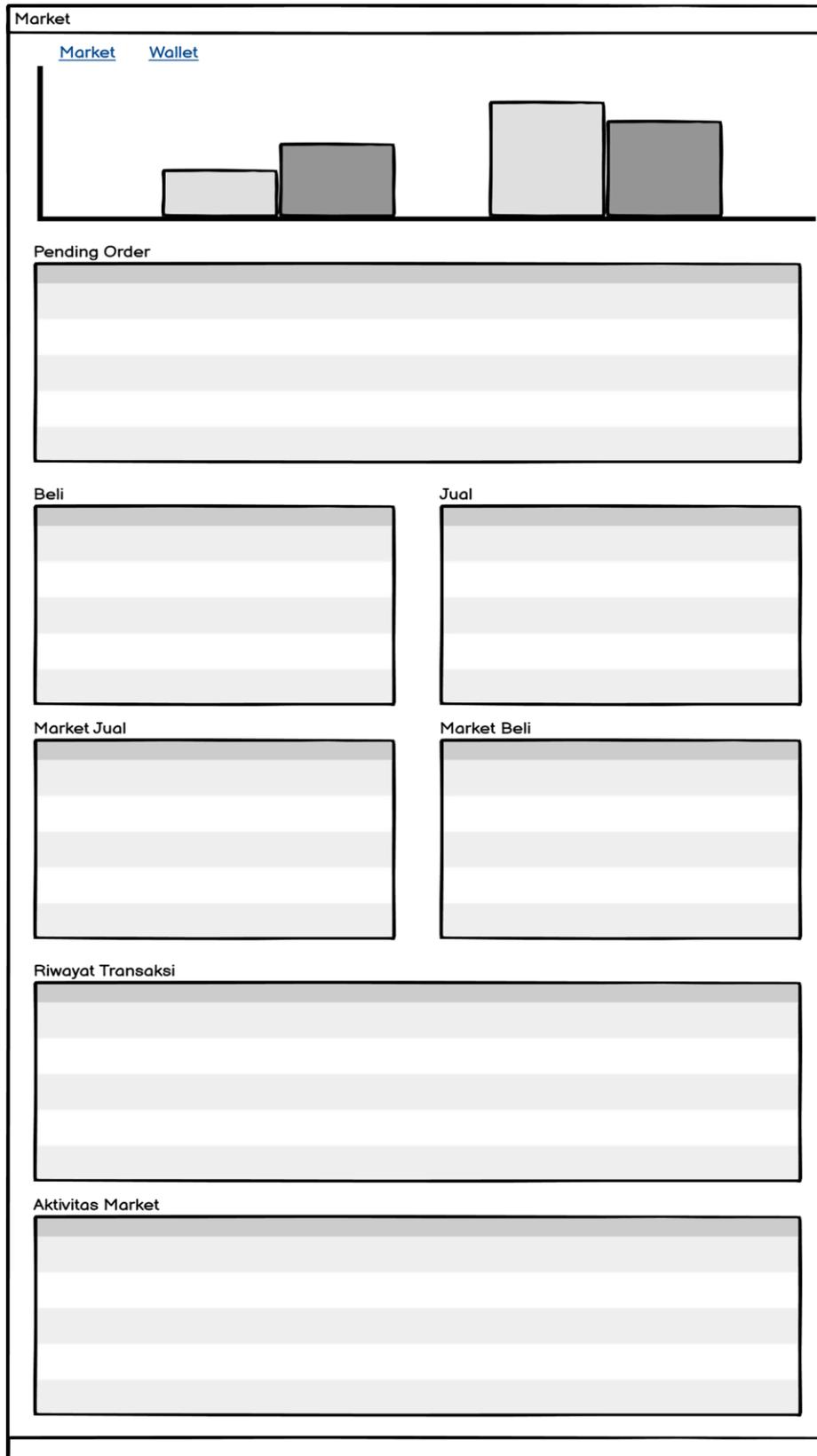
- 4) *Mockup*: Perancangan *mockup* dilakukan untuk menggambarkan tata letak masing-masing fitur yang ada pada *e-marketplace* ini berdasarkan struktur dan kebutuhan halaman yang ada. Selain itu, *mockup* ini juga dibuat agar dapat mempermudah dan mempercepat perancangan desain antarmuka dari *e-marketplace* ini. Berikut adalah beberapa *mockup* dari beberapa *mockup* yang telah dirancang, yaitu *mockup home*, *wallet*, dan *market*. Gambar 5 merupakan *mockup* dari halaman *home* yang akan tampil ketika pengguna maupun pengunjung pertama kali membuka situs *website e-marketplace* ini. Terdapat tombol untuk melakukan *login* atau *register* serta terdapat juga tombol untuk langsung mengunjungi *e-marketplace* tanpa melakukan *login* atau *register*. Pada bagian bawah, akan ditemukan berbagai informasi singkat mengenai *e-marketplace* ini, seperti fitur yang tersedia, *in-game currency* apa saja yang dapat diperjualbelikan, kontak yang dapat dihubungi, dan lainnya. Pada Gambar 6 merupakan *mockup* dari halaman *wallet*. Pada halaman ini, pengguna dapat memasukkan beberapa data yang diperlukan untuk melakukan *deposit* maupun *withdraw in-game currency*. Sedangkan Gambar 7 merupakan *mockup* dari halaman *market*. Pada halaman ini, pengguna dapat melakukan pembelian maupun penjualan *in-game currency* yang tersedia. Selain itu, terdapat juga fitur grafik dan berbagai data *market* yang dapat membantu pengguna dalam melakukan transaksi jual beli.



Gambar 5. Mockup Home



Gambar 6. Mockup Wallet



Gambar 7. Mockup Wallet

### C. Coding

Selanjutnya, pada tahapan *coding*, aplikasi *e-marketplace* ini akan dikembangkan dengan menggunakan *framework* Laravel. Pengembangan dilakukan dengan mengacu pada desain diagram sistem dan *mockup* yang telah dirancang pada tahapan sebelumnya. Proses *coding* dilakukan dengan menggunakan Visual Studio Code dengan Apache yang terintegrasi dengan Laragon sebagai *development server*-nya. Pada tahapan *coding*, *pair programming* digunakan untuk mempercepat proses pengembangan. *Pair programming* di sini dapat dilakukan dengan menggunakan *extension* dari Visual Studio Code, yaitu Live Share. Dengan *extension* ini, proses *pair programming* dapat dilakukan dan membuat proses pengembangan aplikasi *e-marketplace in-game currency* ini menjadi jauh lebih mudah dan cepat. Selain itu, *pair programming* juga dilakukan bersamaan dengan beberapa pengguna, di mana pengguna memberikan saran dan masukan secara langsung mengenai apa yang dibutuhkan dan apa yang tidak, sehingga mempermudah dan mengurangi terjadinya miskomunikasi atau ketidaksesuaian hasil akhir dengan yang dibutuhkan oleh pengguna.

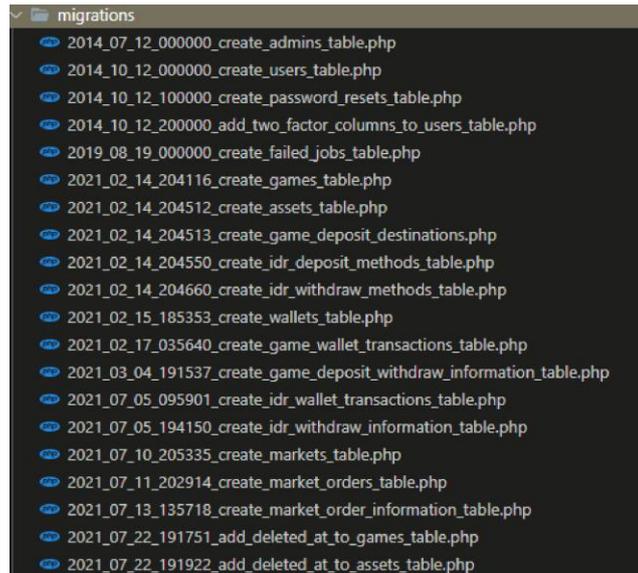
Proses ini dimulai dengan menyiapkan *project* Laravel baru dengan menggunakan Composer. Setelah itu, dilanjutkan dengan menginstal berbagai *package* yang dibutuhkan untuk *e-marketplace* ini. Kemudian, membuat dan menyusun *model*, *migration*, *route*, *controller*, dan *view* yang dibutuhkan. Pertama-tama, perlu diinstal beberapa *package* yang diperlukan. *Package* yang diinstal akan digunakan untuk menunjang aplikasi *e-marketplace* ini. Penggunaan *package* juga dilakukan dengan tujuan agar tidak terlalu lama ketika melakukan pengembangan aplikasi ini. Hal ini sesuai dengan nilai *simplicity* dari metode *Extreme Programming*.

*Package* pertama adalah PHP, yang merupakan bahasa pemrograman dari *framework* Laravel, sekaligus bahasa pemrograman yang digunakan pada aplikasi ini. Kemudian, *fideloper/proxy* merupakan *package* bawaan Laravel untuk mengatur pembuatan tautan yang benar, *redirecting*, *session*, dan sejenisnya. Selanjutnya, *fruitcake/laravel-cors* juga merupakan *package* bawaan Laravel untuk mengkonfigurasi pengaturan mengenai CORS pada aplikasi ini. Untuk *guzzlehttp/guzzle*, juga merupakan *package* bawaan Laravel, yang fungsinya adalah untuk mengirimkan *HTTP Request* ke luar aplikasi. Kemudian, *laravel/fortify* merupakan *package back-end* untuk membangun autentikasi dengan lebih mudah, cepat, dan mudah dikustomisasi dengan berbagai fitur tambahan lainnya. Untuk *laravel/framework* dan *laravel/tinker* juga merupakan bawaan Laravel, di mana *laravel/framework* adalah *framework* Laravel itu sendiri dan *laravel/tinker* adalah *package* untuk dapat menggunakan berbagai *code* Laravel pada *command line*, hal ini dapat mempermudah dan mempercepat proses pengembangan apabila membutuhkan data *dummy* ataupun ingin mengetes *coding*. Untuk *laravel/ui*, mirip seperti *laravel/fortify*, namun *laravel/ui* di sini hanya digunakan tampilan *view* untuk autentikasinya saja, namun fungsi autentikasi tetap diatur oleh *laravel/fortify*. Dan terakhir adalah *yajra/laravel-datatables-oracle*, yaitu *package datatables* yang dapat menampilkan data pada tabel dengan lebih ringan, berkat *server-side processing* yang digunakan, serta memiliki berbagai fitur-fitur yang tidak ada pada tabel biasa. Daftar *package* yang digunakan ditunjukkan pada Gambar 8.

```
"require": {  
    "php": "^7.3|^8.0",  
    "fideloper/proxy": "^4.4",  
    "fruitcake/laravel-cors": "^2.0",  
    "guzzlehttp/guzzle": "^7.0.1",  
    "laravel/fortify": "^1.7",  
    "laravel/framework": "^8.12",  
    "laravel/tinker": "^2.5",  
    "laravel/ui": "^3.2",  
    "yajra/laravel-datatables-oracle": "^9.0"  
},
```

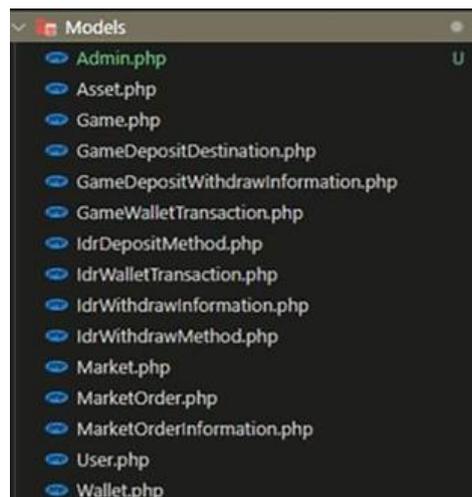
Gambar 8. Daftar *Package* yang Dibutuhkan

Proses *coding e-marketplace* ini dilanjutkan dengan membuat *migration* yang diperlukan untuk membuat tabel pada *database*. *Migration* dibuat sesuai dengan rancangan *entity relationship diagram* sebelumnya. Daftar *migration* yang dibuat dapat dilihat pada Gambar 9.



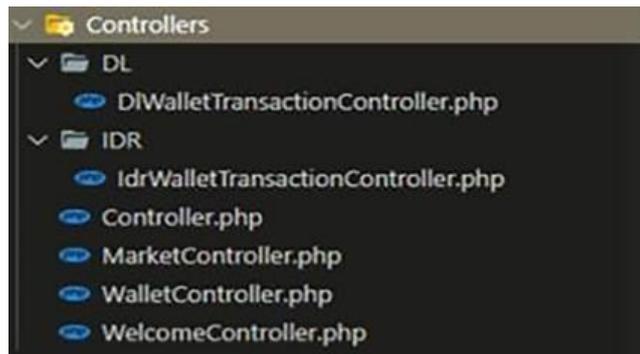
Gambar 9. Daftar Migration

Kemudian, dilanjutkan dengan membuat *model* yang akan digunakan untuk melakukan koneksi dari aplikasi *e-marketplace* ini ke tabel-tabel yang ada di *database*. Daftar *model* dapat dilihat pada Gambar 10.



Gambar 10. Daftar Model

Setelah itu, akan dibuat *controller* dan juga *route*. *Controller* akan digunakan untuk meletakkan berbagai *logic* dari aplikasi ini. Masing-masing *controller* akan dirancang agar berisi *logic* sesuai dengan halaman yang ada. Sedangkan *route* akan digunakan untuk mendefinisikan *controller* mana yang akan diakses ketika menerima sebuah *request* dengan URI tertentu. Daftar *controller* dan *route* dapat dilihat pada Gambar 11 dan Gambar 12.

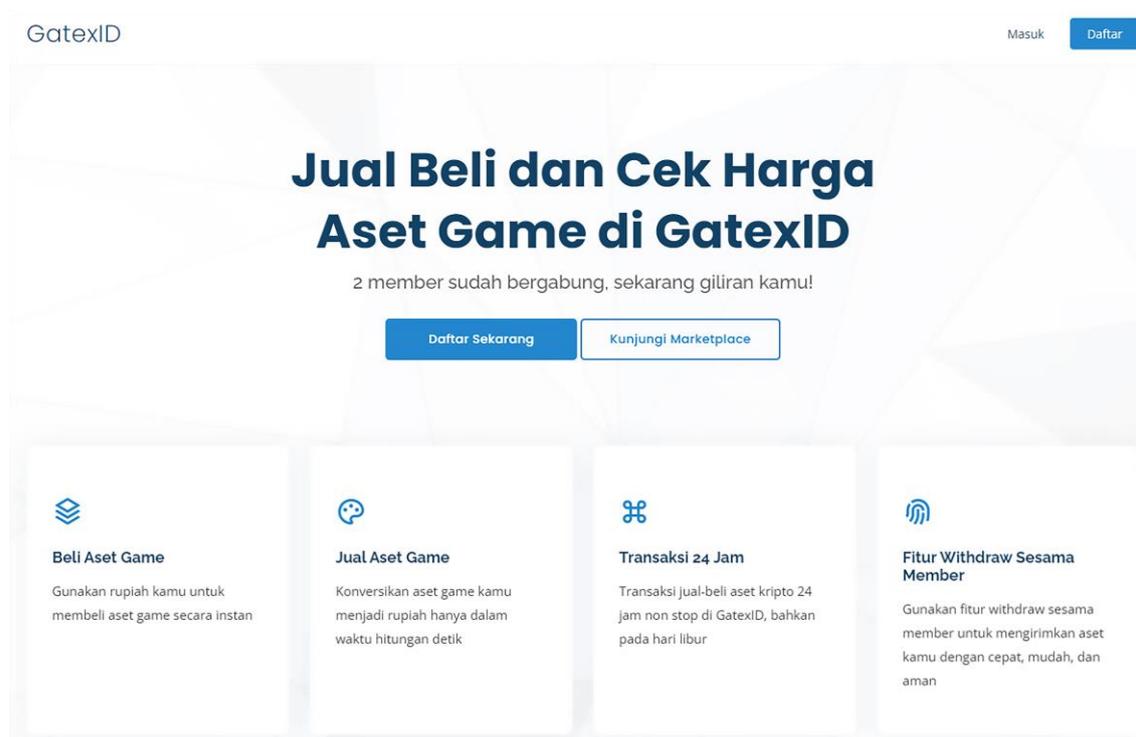


Gambar 11. Daftar Controller

```
19 // Market
20 Route::resource('market', 'MarketController');
21 Route::post('market/market-chart', 'MarketController@marketChart')->name('market.market-chart');
22 Route::post('market/market-title', 'MarketController@marketTitle')->name('market.market-title');
23
24 // Market Order
25 Route::post('market/market-order/count-buy-market', 'MarketController@countBuyMarket')->name('market-order.count-buy-market');
26 Route::post('market/market-order/count-sell-market', 'MarketController@countSellMarket')->name('market-order.count-sell-market');
27
28 Route::post('market/market-order/pending-buy-order', 'MarketController@pendingBuyOrder')->name('market-order.pending-buy-order');
29 Route::post('market/market-order/pending-sell-order', 'MarketController@pendingSellOrder')->name('market-order.pending-sell-order');
30 Route::post('market/market-order/order-activity', 'MarketController@orderActivity')->name('market-order.order-activity');
31
32 // Wallet
33 Route::post('wallet/idr-balance', 'WalletController@idrBalance')->name('wallet.idr-balance');
34 Route::post('wallet/asset-balance', 'WalletController@assetBalance')->name('wallet.asset-balance');
35
```

Gambar 12. Daftar Route

Dan terakhir adalah *view*. *View* merupakan tampilan dari aplikasi ini, di mana *view* merupakan bagian yang berinteraksi dengan pengguna dari aplikasi *e-marketplace* ini. Pada penelitian ini, akan digunakan *templating engine* dari *framework* Laravel yang bernama *Blade*. *Templating engine* dari Laravel ini memiliki berbagai fitur bantuan yang dapat membantu mempercepat proses implementasi, seperti *global helper*, *Blade's directives*, dan masih banyak lagi. *Mockup* yang telah dirancang sebelumnya akan diimplementasikan ke dalam *view* *Blade*. Beberapa hasil implementasi *mockup* yang telah dirancang sebelumnya ke dalam *view* *Blade* dapat dilihat pada Gambar 13, Gambar 15, dan Gambar 16.



Gambar 13. Tampilan Home

Grafik yang terdapat pada setiap *market* pada *e-marketplace in-game currency* ini dibuat berdasarkan harga pembukaan (*open price*), penutupan (*close price*), tertinggi (*high price*), dan terendah (*low price*) dalam periode waktu tertentu dari sebuah *market*. Grafik atau *chart* pada *e-marketplace in-game currency* ini dibuat dengan menggunakan *library* yang bernama Apache ECharts.

Apache ECharts merupakan sebuah *library* JavaScript *open-source* dan berbasis *website* yang mendukung visualisasi data dengan cepat dan interaktif. Apache ECharts dikembangkan oleh Baidu Inc. dan diinkubasi serta menjadi salah satu proyek *open-source* dari Apache pada tahun 2018. ECharts awalnya dikembangkan untuk memudahkan pengguna yang tidak terlalu ahli dalam pemrograman untuk tetap dapat memvisualisasikan data dengan mudah dan efisien. ECharts dibangun dengan menggunakan sistem manajemen dan *rendering* dengan kinerja tinggi dari kanvas HTML5 yang bernama ZRender. Karena hal itu, baik dari sisi performa maupun kustomisasi dari ECharts lebih baik dan lebih mudah dibandingkan *library* JavaScript untuk visualisasi data lainnya. Perbandingan performa ECharts terhadap beberapa *library* lainnya dapat dilihat pada Gambar 14 [15].

Library	Extended charts	Custom interactions	Canvas
HighCharts	X	X	X
Chart.js	X	X	✓
C3	X	X	X
ECharts	✓	✓	✓

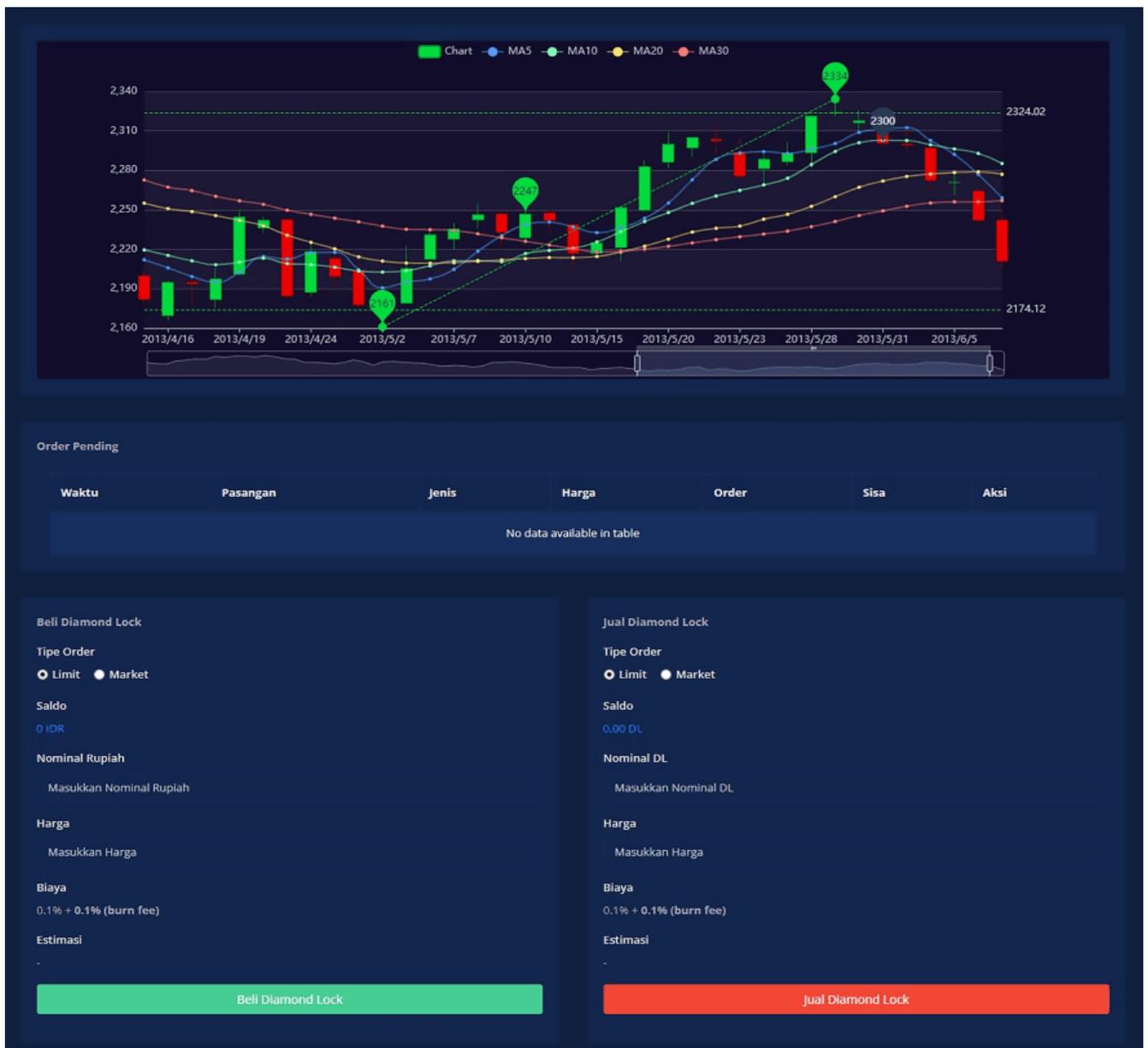
Gambar 14. Perbandingan Performa ECharts dengan *Library* Lainnya [15]

Pada aplikasi *e-marketplace in-game currency* ini, digunakan salah satu contoh yang disediakan pada *website* resmi dari Apache ECharts (<https://echarts.apache.org/en/index.html>), yaitu Shanghai *Index* pada bagian *Candlestick*. Shanghai *Index* ini sangat cocok untuk digunakan pada *e-marketplace in-game currency* ini, alasannya adalah contoh Shanghai *Index* yang

disediakan oleh ECharts memiliki *candle* hijau dan merah yang menandakan naik atau turunnya harga dari harga sebelumnya. Selain itu, juga terdapat fitur untuk mengatur periode grafik sesuai dengan kebutuhan. Di samping itu, penggunaan grafik Shanghai Index ini hanya membutuhkan *open price*, *close price*, *high price*, dan *low price*, serta waktu, di mana *e-marketplace* ini sudah didesain untuk mencatat data-data tersebut setiap 15 menit sekali berdasarkan data *order* yang sudah selesai di *market*. Selain itu, Shanghai Index dari ECharts ini juga menyediakan fitur MA atau *Moving Average* yang dapat membantu pengguna yang ingin menjual atau membeli *in-game currency* dalam membaca pergerakan *market*.

*Moving Average* adalah indikator untuk menganalisa kondisi pasar dengan menggunakan harga rata-rata dalam periode tertentu. Indikator ini sering digunakan oleh para *trader* untuk membaca *trend* yang telah terjadi, yang akan terjadi, serta yang akan berbalik arah. Shanghai Index yang disediakan ECharts ini menyediakan fitur *Moving Average* mulai dari MA5, MA10, MA20, dan MA30, serta masih dapat dikustomisasi sesuai kebutuhan. Angka di belakang MA tersebut adalah jumlah *candlestick* terakhir yang ingin dihitung nilai rata-ratanya.

Pada Gambar 15, dapat dilihat tampilan *market* serta ECharts yang telah diimplementasikan pada *e-marketplace in-game currency* ini.



Gambar 15. Tampilan Halaman Market

The screenshot shows a dark-themed interface for 'WALLET - DIAMOND LOCK'. At the top, there are navigation tabs: 'Deposit', 'Withdraw', and 'Riwayat Transaksi'. Below this, there are two main sections. The left section, titled 'DEPOSIT DIAMOND LOCK', contains a brief instruction, a 'Data Deposit' section with a dropdown menu, a 'Jumlah Deposit' input field, a 'Biaya Deposit' field showing '0', and a 'Jumlah Deposit Bersih' field also showing '0'. At the bottom of this section are 'Lanjut' and 'Kembali' buttons. The right section, titled 'DEPOSIT DIAMOND LOCK VIA PENGGUNA LAIN', contains an 'Alamat Email Anda' field with the value 'khoandy755@gmail.com' and a 'Salin' button. Below that is an 'Alamat Wallet DL Anda' field with the value 'DL-23812509-95da-449f-8265-e4ef96b39e0d' and another 'Salin' button. A 'NOTE:' section follows with four numbered points regarding wallet address usage, deposit timing, fees, and asset security.

Gambar 16. Tampilan Halaman *Wallet*

#### D. Testing

Setelah tahapan coding selesai, maka akan dilakukan tahapan testing. Pada tahapan testing, akan dilakukan pengujian sistem dengan menggunakan Black-box Testing dengan metode Equivalence Partitioning. Terdapat empat yaitu penentuan use case, penentuan kriteria, pendefinisian partisi, data uji, kasus uji, pengujian, dan evaluasi hasil pengujian. Pengujian dilakukan oleh kedua pihak, yaitu pengembang dan beberapa pengguna terpilih. Pengujian yang dilakukan menunjukkan hasil yang positif, artinya pengujian yang dilakukan berhasil dan sudah sesuai dengan tujuan dengan berdasarkan pada kasus uji yang telah ditentukan. Pada Tabel 3 ditunjukkan beberapa hasil pengujian yang telah dilakukan.

TABEL 3  
HASIL PENGUJIAN

Kasus Uji	Data Uji	Tipe Uji	Output	Hasil
<b>Registrasi</b>	Nama, username, email, nomor handphone, password, dan konfirmasi password tidak kosong serta pengguna belum terdaftar	<i>valid</i>	Tidak ada pesan. Registrasi berhasil dan langsung diarahkan untuk memverifikasi <i>email</i> .	Berhasil
	Nama, username, email, nomor handphone, password, atau konfirmasi password kosong sebagian, kosong semua, atau tidak sesuai jenis input	<i>invalid</i>	<i>This value is required. This value should be a valid email.</i>	Berhasil
	Username, email, atau nomor handphone sudah terdaftar	<i>invalid</i>	<i>The username has already been taken.</i>	Berhasil
<b>Login</b>	<i>Username</i> atau <i>email</i> dan <i>password</i> benar	<i>valid</i>	Tidak ada pesan. Login berhasil dan langsung diarahkan ke daftar <i>market</i> .	Berhasil
	<i>Username</i> , <i>email</i> atau <i>password</i> kosong semua, kosong sebagian, atau tidak sesuai jenis input	<i>invalid</i>	<i>This value is required. This value should be a valid email.</i>	Berhasil
	<i>Username</i> , <i>email</i> atau <i>password</i> salah	<i>invalid</i>	<i>These credentials do not match our records.</i>	Berhasil

<b>Deposit In-game Currency</b>	Data <i>deposit in-game currency</i> dan nominal <i>deposit in-game currency</i> tidak kosong	<i>valid</i>	Permintaan deposit berhasil dibuat.	Berhasil
	Data <i>deposit in-game currency</i> atau nominal <i>deposit in-game currency</i> kosong sebagian, kosong semua, atau tidak sesuai jenis <i>input</i>	<i>invalid</i>	Periksa kembali <i>input</i> -an anda.	Berhasil
	Ketika tombol salin email di klik, <i>email</i> pengguna akan tersalin	<i>valid</i>	Tidak ada pesan, email akan otomatis terpilih dan tersalin.	Berhasil
	Ketika tombol salin email di klik, <i>email</i> pengguna tidak tersalin	<i>invalid</i>	Tidak ada pesan, email tidak tersalin.	Berhasil
	Ketika tombol buat <i>wallet</i> di klik, alamat <i>wallet in-game currency</i> milik pengguna akan dibuat	<i>valid</i>	Alamat wallet Anda berhasil dibuat!	Berhasil
	Ketika tombol buat <i>wallet</i> di klik, alamat <i>wallet in-game currency</i> milik pengguna tidak terbuat	<i>invalid</i>	Terjadi masalah, silakan coba lagi.	Berhasil
	Ketika tombol salin <i>wallet</i> di klik, alamat <i>wallet in-game currency</i> milik pengguna akan tersalin	<i>valid</i>	Tidak ada pesan, alamat <i>wallet</i> akan otomatis terpilih dan tersalin.	Berhasil
	Ketika tombol salin <i>wallet</i> di klik, alamat <i>wallet in-game currency</i> milik pengguna tidak tersalin	<i>invalid</i>	Tidak ada pesan, alamat <i>wallet</i> tidak tersalin.	Berhasil
<b>Membeli In-game Currency</b>	Tipe <i>order</i> , nominal, dan harga tidak tidak kosong	<i>valid</i>	Order beli sukses.	Berhasil
	Tipe <i>order</i> , nominal, dan harga kosong sebagian, kosong semua, atau tidak sesuai jenis <i>input</i>	<i>invalid</i>	Periksa kembali <i>input</i> -an Anda.	Berhasil
	Saldo tidak cukup untuk melakukan pembelian	<i>invalid</i>	Nominal tersebut melebihi saldo Anda.	Berhasil

#### IV. SIMPULAN

Berdasarkan hasil penelitian tentang perancangan dan pengembangan *e-marketplace in-game currency* menggunakan *framework* Laravel dengan metode *Extreme Programming* ini, maka dapat disimpulkan bahwa *e-marketplace* ini dapat digunakan oleh pelaku transaksi jual beli *in-game currency* dalam melakukan jual beli *in-game currency* tertentu dengan mudah dan tentunya lebih aman. Metode *Extreme Programming* yang digunakan pada perancangan dan pengembangan aplikasi *e-marketplace in-game currency* ini juga cukup efektif karena dapat mempercepat proses pengembangan dan mengurangi terjadinya ketidaksesuaian kebutuhan pengguna dengan menerapkan metode *pair programming* dan nilai-nilai *Extreme Programming*, yaitu *simplicity*, *communication*, *feedback*, *courage*, dan *respect*. Hasil dari pengembangan aplikasi *e-marketplace in-game currency* menggunakan metode *Extreme Programming* ini juga sudah sesuai dengan kebutuhan pengguna dengan memiliki fitur-fitur penting, seperti fitur grafik pada *market* dan fitur *wallet* untuk membantu pengguna dalam mengelola *in-game currency* yang dimiliki juga sangat membantu. Dengan adanya *e-marketplace in-game currency* ini, kebutuhan transaksi *in-game currency* akan menjadi lebih mudah dilakukan, cepat, dan aman. Saran untuk pengembangan aplikasi ini adalah penambahan *payment gateway* untuk proses *deposit* dan *withdraw* rupiah dan juga menggunakan teknologi *real-time* pada *e-marketplace* ini untuk meningkatkan kenyamanan dari pengguna.

#### UCAPAN TERIMA KASIH

Puji dan syukur kepada Tuhan Yang Maha Esa atas segala rahmat dan karunia-Nya sehingga penelitian ini dapat diselesaikan. Penulisan karya tulis ini tidak terlepas dari bimbingan, bantuan, dan dorongan yang sangat berarti dari berbagai pihak. Oleh karena itu, rasa terima kasih sebesar-besarnya diucapkan kepada dosen pembimbing, teman-teman, dan juga Universitas Internasional Batam atas bantuannya dalam penulisan karya tulis ini.

#### DAFTAR PUSTAKA

- [1] M. A. Suplig, "Pengaruh Kecanduan Game Online Siswa SMA Kelas X Terhadap Kecerdasan Sosial Sekolah Kristen Swasta Di Makassar," *J. Jaffray*, vol. 15, no. 2, p. 177, 2017, doi: 10.25278/jj71.v15i2.261.
- [2] M. Wang, "Online Games Virtual Products-Transaction and Value," in *Proceedings of the 2018 International Conference on Management, Economics, Education and Social Sciences (MEESS 2018)*, 2018, vol. 236, pp. 340–347, doi: 10.2991/meess-18.2018.63.

- [3] I. Hughes, "Video Game Currency: Clever or Corrupting," *Itnow*, vol. 61, no. 4, pp. 60–63, Dec. 2019, doi: 10.1093/itnow/bwz117.
- [4] Channel NewsAsia, "Teenager Arrested in Cheating Case Involving Growtopia Accessories Buyers," [www.channelnewsasia.com](https://www.channelnewsasia.com/news/singapore/teenager-arrested-in-cheating-case-involving-growtopia-9953076), 2018. <https://www.channelnewsasia.com/news/singapore/teenager-arrested-in-cheating-case-involving-growtopia-9953076> (accessed Oct. 04, 2020).
- [5] T. Tristiyanto, Y. Heningtyas, and H. Risnawati, "Aplikasi Marketplace Penyewaan untuk Koperasi Menggunakan Laravel," *J. Komputasi*, vol. 8, no. 1, pp. 40–49, Apr. 2020, doi: 10.23960/komputasi.v8i1.2536.
- [6] E. Yulianto and F. Nurochim, "Pembangunan Marketplace Online Barang Digital Menggunakan Fitur Product Recommendation (Rating & Tag Analytics) dan Braintree Payment System," *Charity*, vol. 3, no. 2, pp. 1–13, 2020, doi: 10.25124/charity.v3i2.2712.
- [7] S. Susilowati and M. T. Negara, "Implementasi Model Rapid Application Development (RAD) dalam Perancangan Aplikasi e-Marketplace," *J. TECHNO Nusa Mandiri*, vol. 15, no. 1, pp. 25–30, 2018.
- [8] A. Waworuntu and E. Lumba, "Pengembangan Aplikasi Pembuatan Formulir Digital Menggunakan Laravel," in *SEMNATI 2019*, 2019, pp. 16–21.
- [9] M. Mansur, R. Yani, and K. Kasmawi, "Desain Sistem Aplikasi Les Privat Menggunakan Pendekatan Extreme Programming," *Digit. Zo. J. Teknol. Inf. dan Komun.*, vol. 11, no. 1, pp. 30–42, 2020, doi: 10.31849/digitalzone.v11i1.3862.
- [10] F. Anwer, S. Aftab, S. Shah Muhammad Shah, and U. Waheed, "Comparative Analysis of Two Popular Agile Process Models: Extreme Programming and Scrum," *Int. J. Comput. Sci. Telecommun.*, vol. 8, no. 2, pp. 1–7, 2017, [Online]. Available: [www.ijcst.org](http://www.ijcst.org).
- [11] I. Carolina and A. Supriyatna, "Penerapan Metode Extreme Programming dalam Perancangan Aplikasi Perhitungan Kuota Sks Mengajar Dosen," *J. IKRA-ITH Inform.*, vol. 3, no. 1, pp. 106–113, 2019, doi: 10.31227/osf.io/se6f9.
- [12] R. S. Pressman and B. R. Maxim, *Software Engineering: A Practitioner's Approach*, 8th ed. McGraw-Hill Education, 2015.
- [13] R. Priskila, "Perancangan Sistem Informasi Persediaan Barang pada Perusahaan Karya Cipta Buana Sentosa Berbasis Web dengan Metode Extreme Programming," *J. Comput. Eng. Syst. Sci.*, vol. 3, no. 2, pp. 94–99, 2018.
- [14] G. M. Sari, T. Tristiyanto, and M. Iqbal, "Sistem Informasi Simpan Pinjam Multikoperasi Menggunakan Framework Laravel," *J. Komputasi*, vol. 7, no. 2, pp. 1–7, 2019, doi: 10.23960/komputasi.v7i2.1902.
- [15] D. Li *et al.*, "ECharts: A Declarative Framework for Rapid Construction of Web-based Visualization," *Vis. Informatics*, vol. 2, no. 2, pp. 136–146, 2018.