



Perancangan *E-Commerce* Berbasis Android dengan Menggunakan Metode Pembayaran *Blockchain*

Designing E-Commerce Based on Android Using Blockchain Payment Method

Pramudya Erviansyah^{*}, Ike Pertiwi Windasari, Rinta Kridalukmana

Program Studi Teknik Komputer, Fakultas Teknik, Universitas Diponegoro
Jl. Prof. Soedarto, SH, Kampus Undip Tembalang, Semarang, Indonesia 50275

How to cite: P. Erviansyah, I. P. Windasari, and R. Kridalukmana, "Perancangan *E-Commerce* Berbasis Android dengan Menggunakan Metode Pembayaran *Blockchain*", *Jurnal Teknik Komputer*, vol. 1, no. 3, pp. 101-111, Dec 2022, doi: 10.14710/jtk.v1i3.36844 [Online].

Abstract – *E-commerce is one of the platforms used for buying and selling transactions via the internet. E-commerce that is easily accessible to the public helps in a series of transaction processes quickly and practically. Along with its development, e-commerce is made in the form of mobile device applications, especially smartphones with various operating systems, for example, Android. Many e-commerce applications currently use conventional payments in form of interbank transfers, payments through supermarkets, digital wallets, and other services. This method is certainly less efficient because it takes a long time and costs a lot of money. Blockchain technology, for example, Stellar, can be used as an alternative payment method because it is relatively cheaper, safer, and faster. Based on this explanation, this research aims at designing an Android-based e-commerce application with the blockchain (Stellar) payment method. The results of the research are some conclusions that e-commerce applications using the blockchain payment method (Stellar) have been successfully created according to needs*

Kata kunci – *Android; MySQL; E-Commerce; Slim Microframework; RAD; Blockchain; Stellar; API*

Abstrak – *E-commerce merupakan salah satu platform yang digunakan untuk transaksi jual beli melalui internet. E-commerce yang mudah diakses oleh masyarakat membantu dalam serangkaian proses transaksi dengan cepat dan praktis. Seiring dengan perkembangannya, e-commerce dibuat dalam bentuk aplikasi perangkat mobile terutama smartphone dengan berbagai sistem operasi contohnya Android. Banyak aplikasi e-commerce pada saat ini masih menggunakan pembayaran konvensional berupa transfer antar bank, pembayaran melalui toko swalayan, digital wallet dan layanan lainnya. Cara itu tentu kurang efisien karena membutuhkan waktu yang lama dan biaya yang tidak sedikit. Teknologi blockchain contohnya Stellar dapat digunakan sebagai metode alternatif pembayaran karena relatif*

lebih murah, aman dan lebih cepat. Berdasarkan penjelasan tersebut, penelitian ini ditujukan untuk membuat rancangan aplikasi e-commerce berbasis Android dengan metode pembayaran blockchain. Hasil penelitian yang didapatkan berupa kesimpulan bahwa aplikasi e-commerce dengan menggunakan metode pembayaran blockchain berhasil dibuat sesuai dengan kebutuhan.

Kata kunci – *Android; MySQL; E-Commerce; Slim Microframework; RAD; Blockchain; Stellar; API*

I. PENDAHULUAN

Pada era yang sudah sangat berkembang terutama dibidang teknologi dan internet, segala sesuatu dapat dengan mudah diselesaikan menggunakan teknologi dan dapat dikerjakan melalui internet. Perkembangan teknologi dan internet memberi pengaruh yang besar terhadap gaya hidup masyarakat, termasuk transaksi jual beli yang dilakukan melalui internet atau biasa dikenal masyarakat sebagai *online shopping*. *E-commerce* adalah platform yang memungkinkan transaksi jual beli dilakukan secara *online* dari sudut tempat mana pun[1]. Konsep *online shopping* melalui *e-commerce* menyediakan berbagai kemudahan dan kelebihan jika dibandingkan dengan konsep belanja konvensional. *E-commerce* yang mudah diakses membantu masyarakat untuk melakukan serangkaian proses transaksi dengan cepat dan praktis. Selain proses transaksi yang lebih cepat, *e-commerce* juga menyediakan berbagai barang yang biasanya dibutuhkan oleh masyarakat mulai dari alat rumah tangga hingga alat elektronik. Didukung oleh perkembangan teknologi dan internet yang semakin canggih serta mudah digunakan, perkembangan *e-commerce* pun semakin bertambah dan semakin diminati oleh masyarakat maupun perusahaan.

Disisi lain, kemajuan teknologi yang paling pesat salah satunya adalah sstem operasi Android yang digunakan oleh berbagai perangkat *mobile* khususnya *smartphone*. Pada masa sekarang, perangkat *mobile* tidak hanya dapat digunakan untuk mengirim dan menerima panggilan, namun sudah dapat melakukan berbagai fungsi layaknya sebuah komputer. Seiring dengan perkembangan *smartphone* berbasis Android,

^{*}) Penulis Korespondensi (P. Erviansyah)
Email: perviansyah@gmail.com



pengguna sistem operasi ini juga semakin meningkat. Pada kuartal ketiga tahun 2019, Android merupakan sistem operasi *mobile* yang paling populer di Indonesia. Android sudah menguasai pangsa pasar di Indonesia yaitu sebesar 93,79%, jauh meninggalkan iOS yang hanya 5,32%, Windows phone 0,16%, Tizen 0,14% dan sistem operasi *mobile* lainnya 0,54%[2]. Berbagai aplikasi tersedia di Android seperti permainan, sosial media, pemutar musik, kamera, termasuk aplikasi *e-commerce*. Para pengembang dan perusahaan beramai-ramai membuat aplikasi berbasis Android selain karena penggunaannya yang banyak, aplikasi berbasis Android juga mudah untuk dikembangkan karena sifat sistem operasinya yang *open source* sehingga pengembang dapat dengan mudah melihat dan memahami *source code* sistem operasi Android secara bebas.

Dengan adanya *e-commerce* berbasis *mobile* yang dapat digunakan dimana pun oleh masyarakat selama memiliki koneksi internet. Kegiatan berbelanja masyarakat Indonesia mulai berubah dari *offline* ke *online*. Namun, belum diiringi perubahan metode pembayaran yang signifikan. Kebanyakan masyarakat masih memilih menggunakan metode transfer tunai melalui ATM, metode bayar di tempat (*Cash on Delivery* atau COD) atau kartu kredit[3]. Cara itu tentu kurang efisien karena membutuhkan waktu transaksi yang lama, biaya transaksi yang tidak sedikit dan tidak terlalu aman[4][5]. Blockchain dapat digunakan sebagai alternatif metode pembayaran secara *online* karena tingkat keamanan tinggi dan biaya transaksi rendah. Keamanan sangat diperlukan dalam *e-commerce* agar tidak ada penyadapan pada saat proses transaksi atau pertukaran data antara *client* dan *server*. Blockchain dirancang sebagai instrumen yang aman dan merupakan sistem komputasi terdistribusi yang memungkinkan proses transaksi berjalan secara *peer-to-peer* tanpa mengandalkan satu *server*. Blockchain saat ini juga telah diterapkan pada berbagai bidang selain bidang keuangan seperti ketahanan pangan, sistem pemungutan suara (*voting*) dan perencanaan kota[6][7]. Terdapat banyak platform berbasis teknologi Blockchain yang melayani transaksi pembayaran, seperti Bitcoin, Ripple, Ethereum dan Stellar (Lumens). Stellar merupakan platform berbasis Blockchain dengan alat pembayaran *cryptocurrency* bernama Lumens (XLM). Dibandingkan dengan platform lainnya, jaringan Stellar memiliki kecepatan lebih cepat, tingkat keamanan tinggi dan dengan biaya transaksi rendah[8][9].

Berdasarkan penjelasan diatas, penulis merasa tertarik untuk membahas teknologi Blockchain khususnya Stellar dan menerapkannya sebagai metode pembayaran ke dalam sebuah aplikasi *e-commerce* berbasis Android, sehingga proses pembayaran cepat, aman dan murah.

II. KAJIAN LITERATUR

A. Kajian Penelitian Terdahulu

Penelitian ini dilakukan tidak terlepas dari penelitian sebelumnya yang telah dilakukan sebagai referensi.

Hasil penelitian sebelumnya yang dijadikan referensi memiliki topik penelitian yang sejenis yaitu mengenai analisis dan perancangan aplikasi *e-commerce* berbasis Android dan penerapan teknologi Blockchain untuk metode pembayaran.

Telah banyak penelitian yang dilakukan untuk menggambarkan penerapan teknologi Blockchain pada era digital. *Cryptocurrency* yang telah dikenal luas hanyalah salah satu produk yang berbasis Blockchain. Saat ini Blockchain telah digunakan diberbagai bidang, seperti ketahanan pangan, tata kelola lingkungan, perencanaan kota, sistem pemungutan suara (*voting*) dan *smart contract*[6][7][10]. Pada penelitian Blockchain sebagai sistem *voting*, Blockchain digunakan sebagai jaringan penghubung antar *node* yang berlaku sebagai saksi yang memiliki kunci (*key*) yang diakui. Jaringan Blockchain ini menyediakan catatan dari semua saksi dan bertindak sebagai saluran komunikasi dan penyimpanan data^[8]. Pada *smart contract*, Blockchain digunakan sebagai rangkaian tanda tangan digital (*digital signature*) yang saling terhubung dengan data yang terenkripsi[10].

Berdasarkan hasil penelitian yang dilakukan, Blockchain dapat menyederhanakan berbagai proses panjang dan rumit yang melibatkan banyak pihak. Teknologi ini lebih unggul dibandingkan dengan teknologi konvensional karena dapat diimplementasikan di berbagai sektor industri, seperti rantai pasokan dan layanan keuangan. Seluruh data pada jaringan Blockchain dienkripsi dan disimpan diseluruh jaringan Blockchain dalam bentuk blok berdasarkan kesepakatan sehingga sulit untuk diretas karena membutuhkan waktu yang lama untuk memecah kode enkripsi dalam setiap blok tersebut. Diketahui juga bahwa Bank Dunia (World Bank) dan Commonwealth Bank of Australia (CBA) telah memelopori penerbitan obligasi yang dibuat dan dijalankan menggunakan teknologi Blockchain untuk memudahkan penghimpunan data dalam pasar modal[6].

Mengenai perancangan aplikasi *e-commerce* berbasis Android, beberapa penelitian yang telah dilakukan. Aplikasi *e-commerce* yang dibuat dapat digunakan pada berbagai toko antara lain toko baju, distro, elektronik dan komputer[12][13]. Menurut penelitian tersebut, perlu adanya aplikasi yang dapat memudahkan dalam transaksi pemesanan khususnya melalui perangkat bergerak sehingga lebih efektif dan efisien. Aplikasi ini dibuat dan dijalankan pada *smartphone* dengan sistem operasi Android. Android dipilih karena banyak digunakan dan mudah untuk dikembangkan. Android menjadi sistem operasi *mobile* yang paling banyak digunakan di Indonesia pada kuartal ketiga tahun 2019 dengan pangsa pasar sebesar 93,79%, iOS 5,32%, Windows phone 0,16%, Tizen 0,14% dan lainnya 0,54%[2].

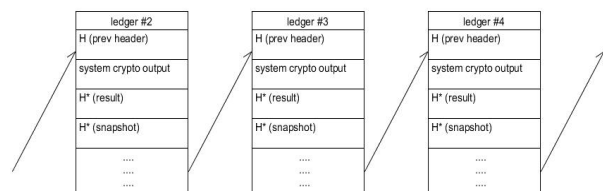
B. Blockchain

Blockchain atau dapat disebut juga sebagai teknologi pembukuan terdistribusi (*Distributed Ledger Technology*/DLT) merupakan sebuah konsep dimana

setiap peserta/pihak yang tergabung dalam jaringan terdistribusi memiliki hak akses terhadap pembukuan tersebut. Konsep yang dibawa oleh Blockchain merupakan penerapan konsep yang sudah ada, yaitu konsep basis data terdistribusi. Konsep ini lahir bersamaan dengan lahirnya Bitcoin sekaligus sebagai jawaban atas permasalahan tidak adanya pihak ke tiga (institusi finansial/pemerintah) untuk membangun kepercayaan diantara pihak-pihak yang melakukan transaksi di lingkungan yang tidak aman.

Pada basis data terdistribusi, informasi yang tercatat akan disimpan dan dibagikan kepada setiap anggota di jaringan tersebut. Teknologi ini juga yang mewujudkan penghilangan/ketidakhadiran pihak ke tiga (institusi finansial/pemerintah) bagi mata uang kripto. Selain itu, teknologi Blockchain juga dapat mencegah terjadinya transaksi ganda/*double-spending* dengan mengombinasikan teknologi jaringan *peer-to-peer* dan kunci publik kriptografi[12].

Fitur utama dari Blockchain adalah memeperbolehkan seseorang secara anonim untuk dapat berinteraksi dan melakukan transaksi kepada anggota lain dalam jaringan secara aman tanpa adanya tambahan pihak ketiga. Blockchain berisikan daftar informasi transaksi yang pada setiap blok yang disebut *ledger* diidentifikasi berdasarkan kriptografi campuran (*hash*). Setiap blok terhubung dan mereferensikan blok sebelumnya berdasarkan fungsi *hash* sehingga menghasilkan rantai blok (*block chain*)[11]. Ilustrasi rantai blok (*block chain*) dapat dilihat pada Gambar 2.2.



Gambar 1. Ilustrasi rantai blok (*blockchain*)

Dari ilustrasi pada Gambar 1, H merupakan informasi dari blok/*ledger* sebelumnya, H* (*result*) merupakan hasil *hash* dari data H dan H* (*snapshot*) berisikan sebagian informasi yang akan dikirim ke blok/*ledger* selanjutnya.

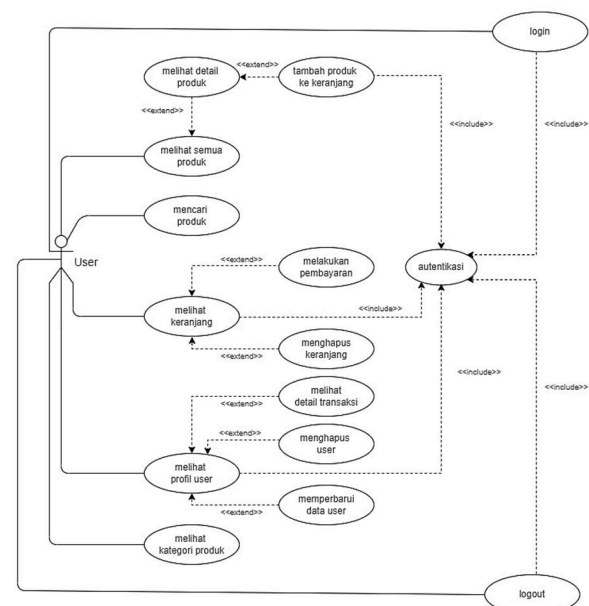
III. METODOLOGI PENELITIAN

Pada penelitian ini, Penulis menggunakan metode perancangan aplikasi *Rapid Application Development (RAD)*[15]. Dilakukan analisis perancangan dengan mengidentifikasi kebutuhan sistem melihat dari penelitian sebelumnya yang telah dilakukan mengenai perancangan *e-commerce* berbasis Android untuk mendapatkan kebutuhan sistem.

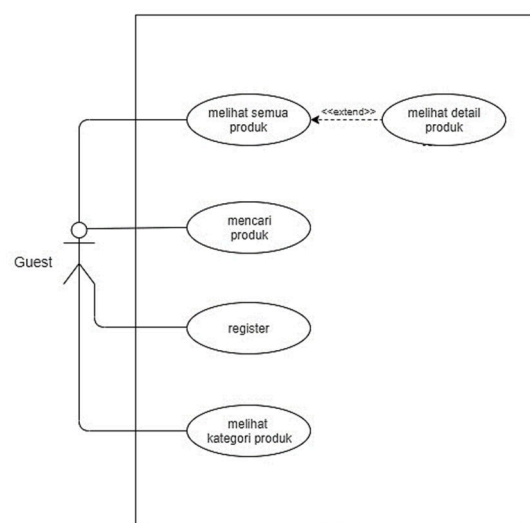
A. Kebutuhan Pengguna

Dari penelitian sebelumnya didapatkan kebutuhan pengguna dapat dipenuhi dengan membuat 2 tingkatan pengguna yang dapat berinteraksi dengan sistem yaitu *user* dan *guest*. *User* dapat melihat produk, melihat

detail sebuah produk, membeli produk, menambahkan produk ke dalam keranjang belanja, membatalkan keranjang belanja, melakukan pembayaran dan menampilkan transaksi pembayaran. *User* juga dapat mengubah data akun dan menghapus akun miliknya. Sedangkan *guest* hanya dapat melihat semua produk yang dijual, melihat detail produk dan melakukan pencarian produk berdasarkan kategori. Kebutuhan tersebut dapat dijelaskan lebih lanjut melalui diagram *usecase*. Diagram ini dapat memodelkan bisnis proses untuk menunjukkan hubungan antara fungsi yang dapat dilakukan aktor dalam sistem. Diagram *usecase user* dapat dilihat pada Gambar 2 dan diagram *usecase guest* dapat dilihat pada Gambar 3.



Gambar 2. Diagram *usecase user*



Gambar 3. Diagram *usecase guest*

Keterangan diagram *usecase user* dapat dilihat pada tabel 1 dan diagram *usecase guest* dapat dilihat pada tabel 2.



Tabel 1. Keterangan usecase user

No	Usecase	Deskripsi
1	Signin	Proses untuk masuk kedalam sistem bagi user sehingga dapat berinteraksi dengan sistem secara penuh.
2	Signout	Proses untuk keluar dari sistem bagi user yang telah masuk, sehingga akan beralih menjadi guest.
3	Autentikasi	Proses untuk mencocokkan data user pada basisdata.
4	Melihat semua produk	Proses untuk melihat semua produk pada aplikasi e-commerce yang dapat diakses oleh semua aktor.
5	Mencari produk	Proses untuk menampilkan produk berdasarkan kategori.
6	Melihat keranjang	Proses melihat keranjang belanja yang berisi produk yang akan dibeli oleh user.
7	Melihat profil user	Proses melihat detail informasi user yang tersimpan pada basis data.
8	Melihat detail produk	Proses melihat detail informasi produk yang diinginkan oleh user.
9	Tambah produk ke keranjang	Proses menambahkan produk diinginkan ke keranjang belanja ketika user masih akan membeli produk lain.
10	Menghapus keranjang	Proses untuk membatalkan produk pada keranjang belanja yang tidak diinginkan oleh user.
11	Memperbarui data user	Proses untuk memperbarui detail informasi user yaitu, email, kata sandi, alamat dan nomor telepon.
12	Menghapus user	Proses untuk menghapus akun user pada basisdata.
13	Melakukan pembayaran	Proses untuk melakukan pembayaran produk yang dibeli oleh user.
14	Memilih jasa pengiriman	Proses untuk memilih jasa pengiriman yang dipakai untuk mengirim produk.
15	Melihat kategori produk	Proses untuk melihat daftar kategori produk yang ada pada aplikasi.

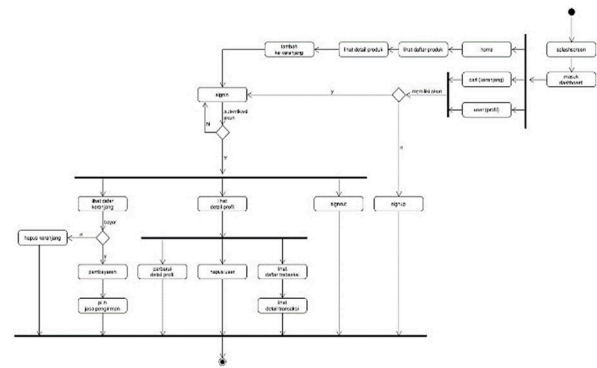
Tabel 2. Keterangan usecase guest

No.	Usecase	Deskripsi
1	Signup	Proses untuk menambah akun ke dalam basisdata sehingga pengguna dapat berinteraksi dengan sistem secara penuh.
2	Melihat semua produk	Proses untuk melihat semua produk pada aplikasi e-commerce yang dapat diakses oleh semua aktor.
3	Mencari produk	Proses untuk menampilkan produk berdasarkan kategori.
4	Melihat detail produk	Proses melihat detail informasi produk yang diinginkan oleh user.
5	Melihat kategori produk	Proses untuk melihat daftar kategori produk yang ada pada aplikasi.

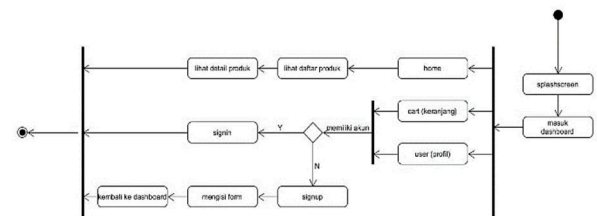
B. Perancangan Sistem dan Desain

Dalam melakukan perancangan sistem ini terdapat beberapa hal yang harus dibuat antara lain, perancangan proses kerja sistem, perancangan distribusi sistem dan

perancangan basis data. Perancangan kerja sistem digambarkan dalam diagram aktifitas. Perancangan distribusi sistem digambarkan dengan diagram deployment dan perancangan basisdata digambarkan dengan entity relationship diagram (ERD). Diagram aktifitas user dapat dilihat pada Gambar 4 dan diagram aktifitas guest pada Gambar 5.

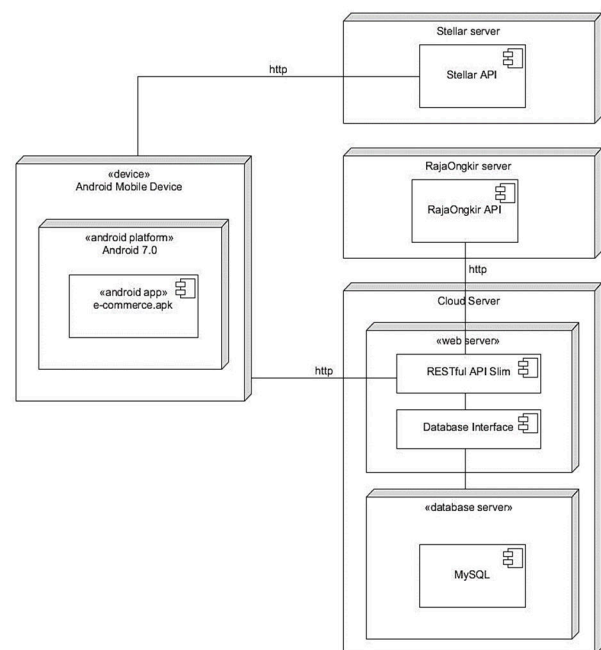


Gambar 2. Diagram aktivitas user



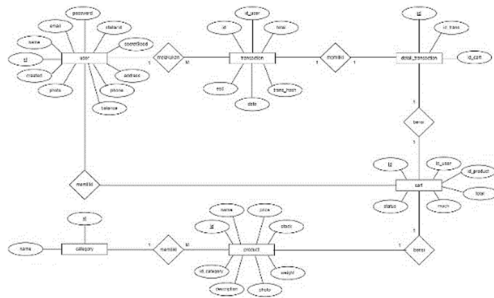
Gambar 3. Diagram aktivitas guest

Diagram deployment sistem dapat dilihat pada Gambar 6.



Gambar 4. Diagram deployment sistem

ERD sistem ditunjukkan pada Gambar 7.



Gambar 5. Entity Relationship Diagram (ERD) Sistem

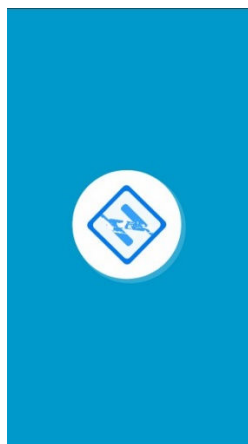
IV. HASIL DAN PEMBAHASAN

A. Implementasi Sistem

Implementasi aplikasi *e-commerce* ini menggunakan Android Studio dengan metode MVP (*Model, View, Presenter*) dijelaskan sebagai berikut:

1. Aktivitas *Splashscreen*

Pada aktivitas ini menampilkan logo sebelum menu utama aplikasi. Tampilan *splashscreen* ditunjukkan pada Gambar 8.



Gambar 6. Tampilan splashscreen

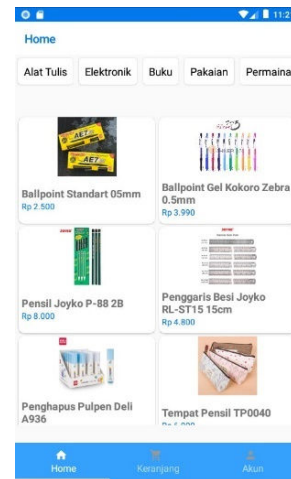
Berkas hubungan antara *model, view* dan *presenter* aktivitas ini ditunjukkan pada Tabel 3.

Tabel 3. Tabel MVP splashscreen

MVP	Nama Berkas	Fungsi
<i>Model</i>	-	-
<i>View</i>	Splashscreen.java	Menampilkan aktivitas <i>Splashscreen</i>
<i>Presenter</i>	-	-

2. Aktivitas *Dashboard Home*

Pada aktivitas ini menampilkan daftar seluruh produk dan kategori yang ada pada basisdata. Pengguna dapat melihat seluruh produk, melihat seluruh kategori produk dan mencari produk berdasarkan kategori. *Dashboard home* dapat diakses oleh *user* dan *guest*. Terdapat 3 buah *tab* pada *dashboard home*, yaitu *tab home* mengarah ke *dashboard home*, *tab keranjang* mengarah ke *dashboard cart* dan *tab profil* mengarah ke *dashboard user*. Tampilan *dashboard home* dapat dilihat pada Gambar 9.



Gambar 7. Tampilan dashboard home

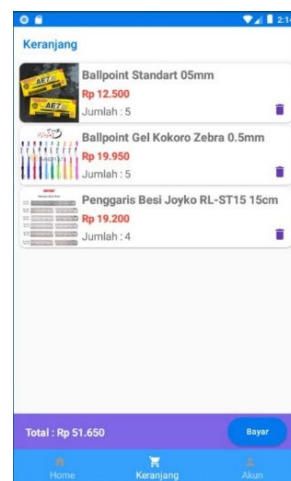
Berkas hubungan antara *model, view* dan *presenter* aktivitas ini ditunjukkan pada Tabel 4.

Tabel 4. Tabel MVP dashboard home

MVP	Nama Berkas	Fungsi
<i>Model</i>	ResponseListProducts.java DataProduct.java	Menentukan data produk yang ditampilkan pada <i>dashboard home</i>
<i>View</i>	HomeFragment.java	Menampilkan aktivitas <i>dashboard home</i>
<i>Presenter</i>	DashboardPresenter.java	Meminta dan menerima data produk dari basisdata

3. Aktivitas *Dashboard Cart* (Keranjang Belanja)

Pada aktivitas ini menampilkan keranjang belanja yang berisikan daftar produk yang dipilih oleh *user*. *Dashboard cart* hanya dapat diakses oleh *user* yang telah masuk menggunakan akun terautentikasi. *User* dapat melihat daftar produk, membatalkan produk yang tidak dipilih dan melihat harga total seluruh produk yang dipilih. *User* dapat membayar produk yang dipilih dengan cara menekan tombol bayar. Terdapat 3 buah *tab* pada *dashboard cart*, yaitu *tab home* mengarah ke *dashboard home*, *tab keranjang* mengarah ke *dashboard cart* dan *tab profil* mengarah ke *dashboard user*. Tampilan *dashboard cart* dapat dilihat pada Gambar 10.



Gambar 8. Tampilan dashboard cart

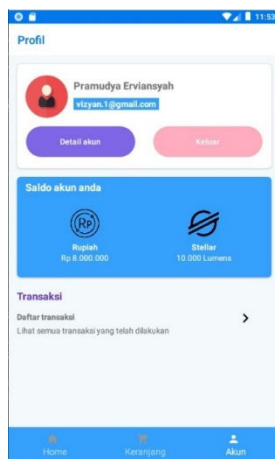
Berkas hubungan antara *model*, *view* dan *presenter* aktivitas ini ditunjukkan pada Tabel 5.

Tabel 5. Tabel MVP dashboard cart

MVP	Nama Berkas	Fungsi
<i>Model</i>	ResponseListCarts.java DataCart.java	Menentukan data produk yang ditampilkan pada keranjang
<i>View</i>	CartFragment.java	Menampilkan aktivitas <i>dashboard cart</i>
<i>Presenter</i>	DashboardPresenter.java	Meminta dan menerima data produk keranjang dari basisdata

4. Aktivitas *Dashboard User*

Pada aktivitas ini menampilkan data *user*, keterangan jumlah saldo, tombol detail *user*, tombol *signout* dan tombol daftar transaksi yang mengarah ke aktivitas daftar transaksi. *Dashboard user* hanya dapat diakses oleh *user* yang telah masuk dengan akun yang terautentikasi. Tampilan *dashboard user* dapat ditunjukkan pada Gambar 11.



Gambar 9. Tampilan dashboard user

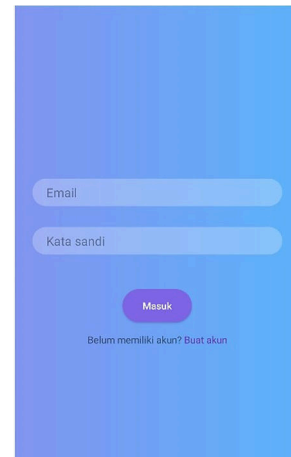
Berkas hubungan antara *model*, *view* dan *presenter* aktivitas ini ditunjukkan pada Tabel 6.

Tabel 6. Tabel MVP dashboard cart

MVP	Nama Berkas	Fungsi
<i>Model</i>	ResponseUser.java DataUser.java	Menentukan data user yang ditampilkan pada <i>dashboard user</i>
<i>View</i>	UserFragment.java	Menampilkan aktivitas <i>dashboard user</i>
<i>Presenter</i>	DashboardPresenter.java	Mengirim dan menerima data user dari basisdata

5. Aktivitas *Signin*

Pada aktivitas ini menampilkan formulir untuk autentikasi data *user* yang terdaftar pada basisdata. Tampilan *signin* dapat ditunjukkan pada Gambar 12.



Gambar 10. Tampilan signin

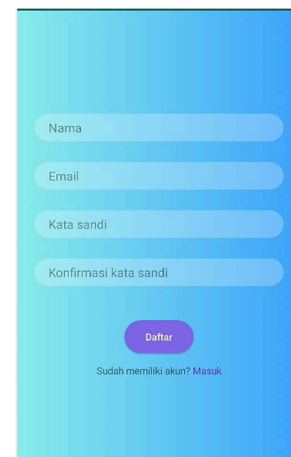
Berkas hubungan antara *model*, *view* dan *presenter* aktivitas ini ditunjukkan pada Tabel 7.

Tabel 7. Tabel MVP signin

MVP	Nama Berkas	Fungsi
<i>Model</i>	ResponseUser.java DataUser.java	Menentukan respon yang ditampilkan pada aktivitas <i>signin</i>
<i>View</i>	SigninActivity.java	Menampilkan aktivitas <i>signin</i>
<i>Presenter</i>	AuthPresenter.java	Mengirim, menerima dan mengautentikasi data <i>user</i> dari basisdata dan menerima respon dari <i>server</i>

6. Aktivitas *Signup*

Pada aktivitas ini menampilkan formulir untuk registrasi pengguna yang belum terdaftar pada basisdata. Tampilan *signup* dapat ditunjukkan pada Gambar 13.



Gambar 11. Tampilan signip

Berkas hubungan antara *model*, *view* dan *presenter* aktivitas ini ditunjukkan pada Tabel 8.

Tabel 8. Tabel MVP signin

MVP	Nama Berkas	Fungsi
<i>Model</i>	ResponseUser.java DataUser.java	Menentukan respon data yang ditampilkan pada aktivitas <i>signup</i>
<i>View</i>	SignupActivity.java	Menampilkan aktivitas <i>signup</i>
<i>Presenter</i>	Presenter.java	Mengirim data <i>user</i> yang akan disimpan pada basisdata dan menerima respon dari server



7. Aktivitas Detail Produk

Pada aktivitas ini menampilkan detail produk yang dipilih oleh pengguna dan tombol tambah ke keranjang. Tombol tambah ke keranjang akan mengarah ke aktivitas *signin* jika pengguna belum masuk (*signin*) ke dalam aplikasi. Setelah pengguna melakukan *signin*, tombol tambah ke keranjang akan mengarah ke aktivitas transaksi. Tampilan detail produk dapat ditunjukkan pada Gambar 14.



Gambar 12. Tampilan detail produk

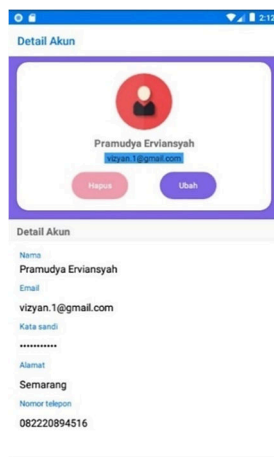
Berkas hubungan antara *model*, *view* dan *presenter* aktivitas ini ditunjukkan pada Tabel 9.

Tabel 9. Tabel MVP detail produk

MVP	Nama Berkas	Fungsi
<i>Model</i>	ResponseListProducts.java DataProduct.java	Menentukan data produk yang akan ditampilkan pada aktivitas detail produk
<i>View</i>	HomeFragment.java	Menampilkan aktivitas detail produk
<i>Presenter</i>	DashboardPresenter.java	Menerima data produk dari basisdata

8. Aktivitas Detail User

Pada aktivitas ini menampilkan detail akun milik *user*, tombol hapus untuk menghapus *user* dari basisdata, tombol ubah untuk mengubah data *user* serta tombol simpan untuk menyimpan data setelah diubah. Tampilan detail *user* ditunjukkan pada Gambar 15.



Gambar 13. Tampilan detail user

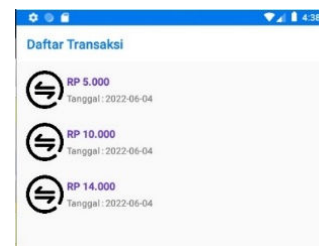
Berkas hubungan antara *model*, *view* dan *presenter* aktivitas ini ditunjukkan pada Tabel 10.

Tabel 10. Tabel MVP detail user

MVP	Nama Berkas	Fungsi
<i>Model</i>	ResponseUser.java ResponseData.java DataUser.java	Menentukan data user yang ditampilkan pada aktivitas detail user
<i>View</i>	UserActivity.java	Menampilkan aktivitas detail user
<i>Presenter</i>	UserPresenter.java	Mengirim dan menerima data user yang disimpan pada basisdata

9. Aktivitas Daftar Transaksi

Pada aktivitas ini menampilkan daftar transaksi yang telah dilakukan oleh *user*. Tampilan daftar transaksi ditunjukkan pada Gambar 16.



Gambar 14. Tampilan daftar transaksi

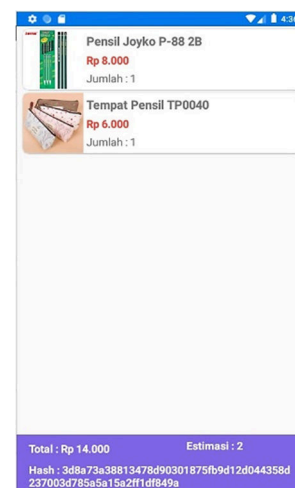
Berkas hubungan antara *model*, *view* dan *presenter* aktivitas ini ditunjukkan pada Tabel 11.

Tabel 11. Tabel MVP daftar transaksi

MVP	Nama Berkas	Fungsi
<i>Model</i>	ResponseListTrans.java DataTrans.java	Menentukan data transaksi yang ditampilkan pada aktivitas daftar transaksi
<i>View</i>	ListTransactionActivity.java	Menampilkan aktivitas daftar transaksi
<i>Presenter</i>	TransactionPresenter.java	Menerima data transaksi dari basisdata

10. Aktivitas Detail Transaksi

Pada aktivitas ini menampilkan detail transaksi yang dipilih oleh *user*. Tampilan detail transaksi ditunjukkan pada Gambar 17.



Gambar 15. Tampilan detail transaksi

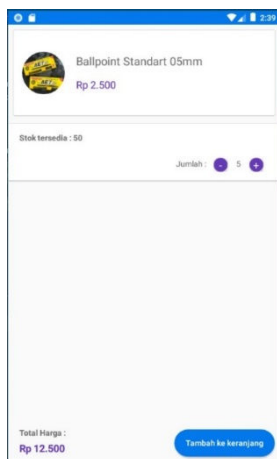
Berkas hubungan antara *model*, *view* dan *presenter* aktivitas ini ditunjukkan pada Tabel 12.

Tabel 12. Tabel MVP ddetail transaksi

MVP	Nama Berkas	Fungsi
<i>Model</i>	ResponseTrans.java DataTrans.java	Menentukan data yang ditampilkan pada aktivitas detail transaksi
<i>View</i>	DetailTransactionActivity.java	Menampilkan aktivitas detail transaksi
<i>Presenter</i>	TransactionPresenter.java	Menerima data detail transaksi yang ada pada sever stellar

11. Aktivitas Transaksi

Pada aktivitas ini menampilkan total harga yang harus dibayar serta pilihan pengiriman. Tampilan transaksi ditunjukkan pada Gambar 18.



Gambar 16. Tampilan transaksi

Berkas hubungan antara *model*, *view* dan *presenter* aktivitas ini ditunjukkan pada Tabel 13.

Tabel 13. Tabel MVP transaksi

MVP	Nama Berkas	Fungsi
<i>Model</i>	ResponseProducts.java ResponseData.java DataProduct.java	Menentukan data transaksi yang akan ditampilkan pada aktivitas transaksi
<i>View</i>	TransactionActivity.java	Menampilkan aktivitas transaksi
<i>Presenter</i>	TransactionPresenter.java	Mengirim data transaksi pada basisdata dan menerima respon dari server

B. Pengujian Sistem

Tahap pengujian bertujuan menguji sistem untuk mengetahui sistem sudah berjalan sesuai dengan rancangan saat pengembangan sebelum aplikasi dipublikasikan. Pengujian sistem pada aplikasi ini menggunakan metode *black box*. Pengujian dengan metode *black box* mempunyai tujuan untuk melakukan pengujian pada setiap fungsi yang berjalan pada aplikasi. Pengujian kebutuhan fungsional aplikasi ditunjukkan pada Tabel 15.

Tabel 14. Tabel Pengujian Fungsional Aplikasi

No	Pengujian Fungsional	Keterangan
1	Tersedia aktivitas <i>splashscreen</i> untuk menampilkan logo dan animasi loading sebelum masuk ke menu utama aplikasi	Tersedia
2	Tersedia aktivitas <i>dashboard home</i> untuk menampilkan daftar produk dan daftar kategori.	Tersedia
3	Tersedia aktivitas <i>dashboard cart</i> untuk menampilkan daftar produk pilihan <i>user</i> yang hanya dapat diakses oleh <i>user</i> yang telah melakukan <i>signin</i> .	Tersedia
4	Tersedia aktivitas <i>dashboard user</i> untuk menampilkan keterangan <i>user</i> secara umum.	Tersedia
5	Tersedia aktivitas <i>signin</i> untuk melakukan autentikasi pengguna dan mendapatkan hak akses pengguna dalam sistem.	Tersedia
6	Tersedia aktivitas <i>signup</i> untuk registrasi pengguna baru.	Tersedia
7	Tersedia aktivitas detail produk untuk menampilkan detail produk yang ingin dilihat oleh <i>user</i> maupun <i>guest</i> .	Tersedia
8	Tersedia aktivitas detail user untuk menampilkan data <i>user</i> secara detail yang hanya dapat diakses oleh <i>user</i> yang telah melakukan <i>signin</i> .	Tersedia
9	Tersedia aktivitas daftar transaksi untuk menampilkan daftar transaksi yang telah dilakukan oleh <i>user</i> .	Tersedia
10	Tersedia aktivitas detail transaksi untuk menampilkan transaksi yang dilakukan oleh <i>user</i> secara detail.	Tersedia
11	Tersedia aktivitas transaksi untuk menampilkan menu pembayaran produk yang dibeli oleh <i>user</i> .	Tersedia

C. Pengujian Transaksi Blockchain (Stellar)

Pengujian yang dilakukan dengan cara memanggil fungsi-fungsi SDK Stellar pada aplikasi *e-commerce* berbasis Android dan berfungsi dengan benar. Berikut adalah pengujian *blockchain (stellar)* dalam proses transaksi.

Setiap pengguna baru membuat akun pada aplikasi, aplikasi akan mendaftarkan akun pada server Stellar menggunakan *stellarId* dan *secretSeed*. *accountId* dan *secretSeed* merupakan satu pasang kata random yang dibuat oleh SDK Stellar. *stellarId* digunakan untuk mendapatkan *public key* dari server stellar. *secretSeed* digunakan untuk mendapatkan *private key* dari server Stellar. *Public key* digunakan sebagai alamat atau identitas pada saat transaksi, sedangkan *private key* digunakan untuk memverifikasi transaksi menggunakan tanda tangan digital. Pada pengujian didapatkan *stellarId* dan *secretSeed* seperti pada Tabel 16.

Tabel 15. Tabel pengujian stellarId dan secretSeed

stellar-Id	GCXL5K4JFCCLWLB2INCWYXNCFNPD5YIL2X4AG5SKUQWIARIK4234C
------------	---



secretSeed	SCRSIQFPMWL2T7Q5D4MCGKR5AJAITMHZ3FUXDCH3XQOUGWSS5LCQV6B
------------	---

```
I/System.out: {
  stellarId: GCXL5K4JFCCLWLBW2INCWYXNCFPNP05YL2X4AGC5KUQWIARIK4234C
  secretSeed: SCRSIQFPMWL2T7Q5D4MCGKR5AJAITMHZ3FUXDCH3XQOUGWSS5LCQV6B
}
```

Gambar 17. Tampilan stellarId dan secretSeed dari server Stellar

```
"id": 18,
"email": "perviansyah@gmail.com",
"name": "Pramudya Erviansyah",
"stellarId": "GCXL5K4JFCCLWLBW2INCWYXNCFPNP05YL2X4AGC5KUQWIARIK4234C",
"secretSeed": "SCRSIQFPMWL2T7Q5D4MCGKR5AJAITMHZ3FUXDCH3XQOUGWSS5LCQV6B",
"balance": "2000000",
"password": "$2y$10$30WcFzuIVMPIHfPoa2Er0BmW1IX8pJhmLXAAVZUcoiB590vXpTL6",
"photo": "http://10.0.2.2:8080/img/peison.webp",
"address": "",
"phone": "",
"created": "2022-06-06 21:48:35"
```

Gambar 18. Tampilan stellarId dan secretSeed dari lokal e-commerce

Saat proses transaksi diinisiasi, aplikasi akan bertransaksi dengan server Stellar. Transaksi dilakukan secara anonim antara *public key* pengirim dengan *public key* penerima. Transaksi Stellar dilakukan dengan menggunakan 3 parameter, yaitu

1. *Public key* merupakan alamat atau identitas yang digunakan dalam proses transaksi, didapatkan dari *accountId* pada SDK Stellar.
2. *Private key* merupakan kunci pasangan untuk tanda tangan digital, didapatkan dari *secretSeed* pada SDK Stellar.
3. *Amount* merupakan jumlah saldo yang dikirimkan.

Fungsi transaksi dalam aplikasi dapat dilihat pada Gambar 22.

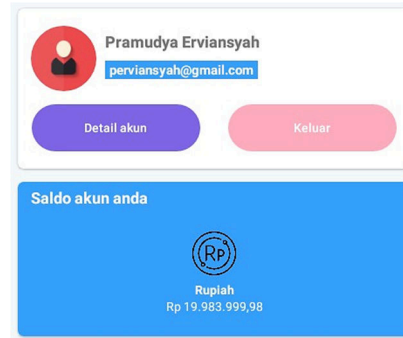
```
Transaction transaction = new Transaction.Builder(publicKey, Network.TESTNET)
    .addOperation(new PaymentOperation.Builder(Constant.ID_DESTINATION, new AssetTypeNative(), amount).build())
    .addMemo(Memo.txt(note))
    .setSourceID()
    .build();
transaction.sign(privateKey);
```

Gambar 19. Tampilan fungsi transaksi stellar pada Android Studio

Pada pengujian transaksi menggunakan akun *user* dengan saldo awal Rp 20.000.000 atau 10000 lumens. *User* melakukan transaksi pembelian produk dengan harga Rp 16.000. Sehingga saldo *user* setelah transaksi menjadi Rp 19.983.999,98. Saldo awal *user* dan saldo setelah transaksi dapat ditunjukkan oleh Gambar 4.23 dan 24.



Gambar 22. Saldo awal user



Gambar 20. Saldo akhir user

Saldo *user* setelah transaksi dapat dihitung melalui persamaan berikut,

Saldo akhir = saldo awal – (harga produk + biaya transaksi)

$$\begin{aligned} \text{Saldo akhir} &= 20.000.000 - (16.000 - 0,02) \\ &= 20.000.000 - (16.000,02) \\ &= 19.983.999,98 \end{aligned}$$

Setelah transaksi berhasil, server akan mengirimkan kembali data transaksi berhasil berupa kode *hash* transaksi. Kode *hash* dapat dilihat pada Gambar 25 dan 26.

```
I/System.out: DashboardPresenter->getStellarDetail: onResponse:
  Balance: 10000.000000
I/System.out: DashboardPresenter->addTransactionStellar(): onResponse:
  Transaction Success
  Hash :a727c7defb62d346566f81db63ae02b8840fa17cf10a7aae21b622c3e593f835
I/Choreographer: Skipped 314 Frames! The application may be doing too much work on its main thread.
I/System.out: DashboardPresenter->addTransaction(): onResponse: true
I/System.out: DashboardPresenter->editUser(): onResponse: true
I/System.out: DashboardPresenter->getStellarDetail: onResponse:
  Balance: 9991.999998
```

Gambar 21. Kode hash dari server stellar



Gambar 22. Kode hash pada aplikasi

D. Pembahasan

Aplikasi *e-commerce* menggunakan metode pembayaran Blockchain ini berbasis Android dibangun menggunakan Android Studio dengan API dari slim framework. Memiliki kelebihan dibangun dalam versi *mobile* Android yang dapat diakses dengan mudah dan menggunakan Blockchain sebagai sistem pembayarannya sehingga lebih aman.

Aplikasi ini diawali dengan *splashscreen* yang menampilkan logo aplikasi dilanjutkan dengan aktivitas *dashboard home* untuk menampilkan semua jenis



produk yang dijual pada *e-commerce*. Aktivitas *dashboard home* dapat diakses oleh semua orang baik *guest* maupun *user*. Terdapat 3 buah *tab* menu pada aplikasi, yaitu *tab home* yang mengarahkan ke *dashboard home*, *tab cart* yang mengarahkan ke aktivitas keranjang belanja dan *dashboard user* yang mengarah ke aktivitas *user*. Pada aktivitas keranjang belanja, hanya *user* yang telah melakukan *signin* yang dapat mengakses aktivitas tersebut. Aktivitas keranjang belanja berisikan produk – produk yang ingin dibeli oleh user. Saat user belum melakukan *signin*, maka user akan diarahkan ke aktivitas *signin* untuk proses autentikasi bagi user yang telah terdaftar. Sedangkan untuk user yang belum terdaftar, disediakan tombol *signup* pada aktivitas *signin* yang mengarahkan user pada aktivitas *signup* untuk mendaftar. Selanjutnya terdapat aktivitas *user* yang berisikan data user dan daftar transaksi yang dilakukan oleh user. Sama dengan aktivitas keranjang belanja, aktivitas *user* hanya bisa diakses oleh user yang telah melakukan *signin*.

User yang telah memilih produk yang ingin dibeli akan diarahkan ke aktivitas detail produk untuk melihat deskripsi produk, setelah itu *user* dapat menambahkan produk kedalam keranjang belanja. Untuk melakukan pembayaran *user* dapat melakukannya melalui tombol bayar pada aktivitas keranjang belanja. *User* akan diarahkan ke aktivitas pengiriman dan pembayaran.

Metode pembayaran pada aplikasi ini menggunakan bentuk *cryptocurrency* platform Stellar yaitu lumens. Saldo yang terdapat pada aplikasi akan dikonversi kedalam lumens dalam proses transaksi. Stellar akan memproses transaksi ke dalam rangkaian blok–blok data (*ledger*) pada *server*, rangkain tersebut menggunakan sistem keamanan blockchain. *Server localhost* akan menyimpan *stellarId* dan *secretSeed* pada saat registrasi untuk dicocokkan dengan *public key* dan *private key* dari server Stellar pada saat transaksi dilakukan. Setelah transaksi berhasil, server Stellar akan mengirimkan data kembali berupa kode hash transaksi sebagai bukti transaksi. Metode ini lebih aman dibandingkan dengan metode konvensional yang lain. Selain itu stellar hanya menarik biaya sebesar 0.00001 lumens atau Rp 0.02 (dengan kurs 1 lumens = Rp 2000) setiap transaksi. Perbandingan biaya transaksi dari berbagai metode lain ditunjukkan oleh Tabel 17.

Tabel 16. Tabel perbandingan biaya transaksi

Metode	Biaya
BI-FAST (Transfer Antar Bank)	Rp 3.500
Toko swalayan	Rp 2.500
Digital Wallet (Dompot Digital)	Rp 1.000
Stellar (lumens)	Rp 0,02

V. KESIMPULAN DAN SARAN

A. Kesimpulan

Dari hasil analisis dan pengujian Perancangan *E-commerce* Berbasis Android dengan Metode

Pembayaran Blockchain (Stellar), dapat disimpulkan bahwa :

1. Berdasarkan pengujian fungsional yang dilakukan terhadap sistem, semua fungsi dapat berjalan dengan semestinya
2. Berdasarkan pengujian sistem dengan metode *blackbox*, semua fungsi pada masing-masing aktivitas berjalan seperti yang diharapkan. Mulai dari menampilkan semua produk, menampilkan produk berdasarkan kategori, *signin*, *signup*, menampilkan detail akun, mengubah data akun menambahkan produk kedalam keranjang dan melakukan transaksi.
3. Berdasarkan pengujian transaksi blockchain, metode pembayaran blockchain dapat diimplementasikan dengan baik kedalam aplikasi *e-commerce* berbasis Android.
4. Transaksi menggunakan metode blockchain (Stellar) relatif lebih murah dibandingkan dengan metode transaksi konvensional lainnya yaitu 0,00001 lumens atau Rp 0,02. Transaksi stellar lebih aman karena telah menggunakan teknologi blockchain. Transaksi stellar dilakukan secara anonim antara *public key* dan diverifikasi melalui tanda tangan digital menggunakan *private key*.
5. Aplikasi mudah diakses karena dapat dijalankan pada perangkat *smartphone* berbasis Android yang banyak dipakai oleh masyarakat.

B. Saran

Dari hasil analisis dan pengujian Perancangan *E-commerce* Berbasis Android dengan Metode Pembayaran Blockchain (Stellar) dapat diberikan saran sebagai berikut:

1. Pengembangan aplikasi selanjutnya dapat dijalankan pada perangkat *smartphone* selain berbasis Android, seperti iOS.
2. Pengembangan aplikasi selanjutnya menggunakan *user interface* atau antarmuka yang lebih menarik sehingga pengguna tidak merasa bosan saat berinteraksi dengan aplikasi.
3. Pengembangan aplikasi selanjutnya berupa penambahan fitur-fitur yang dapat mempermudah proses transaksi seperti fitur *chat* dan pencarian produk berdasarkan nama atau harga.
4. Penggunaan metode berbeda dengan metode pembayaran yang lebih cepat, murah dan aman selain menggunakan Stellar.

DAFTAR PUSTAKA

- [1] H. Nawaf, S. Sinan, dan A. Mustafa, “*Pathfinding in Strategy Games and Maze Solving Using A* Search Algorithm*”, Baghdad: Journal of Computer and Communications, 2016.
- [2] F. Daniel, G. Alifio, dkk, “*A Systematic Literature Review of A* Pathfinding*”, Jakarta: Computer Science Department, School of Computer Science, Bina Nusantara University, 2020.



- [3] C. Ade, A. Mohammad, dkk, “*Application of A-Star Algorithm on Pathfinding Game*”, Sumatra Utara: Faculty of Computer Science and Information Technology, Universitas Sumatra Utara, 2020.
- [4] D. Silvester, B. Ketut, dkk, “*Comparative Analysis of Pathfinding Algoritma A**”, Medan: Journal of Physics: Conference Series, 2020.
- [5] M. Ghani, F. Juniardi, dkk, “*Implementasi Metode Pathfinding dengan Penerapan Algoritma A-Star untuk Mencari Jalur Terpendek pada Game Jumrah Launch Story*”, Walisongo Journal of Information Technology, 2021.
- [6] H. Zhonghua, S. Pengfei, dkk, “*Improved A-Star Algorithm for Long-Distance Off-Road Path Planning Using Terrain Data Map*”, ISPRS International Journal of Geoinformation, 2021.
- [7] D. Frantisek, dkk, “*Path planning with modified A-Star Algorithm for a Mobile Robot*”, Elsevier ltd., 2014.
- [8] H. Zhibo, L. Chenguang, dkk, “*Dynamic Anti-Collision A-Star Algorithm for Multiship Encounter Situations*”, Wuhan: Intelligent Transport System Research Center, Wuhan University of Technology, 2021.
- [9] W. Huanwei, L. Shangjie, dkk, “*The EBS-A* Algorithm: An Improved A* Algorithm for Path Planning*”, Zhengzhou: State Key Laboratory of Mathematical Engineering and Advanced Computing, 2022.
- [10] J. K. Haas, “*A History of the Unity Game Engine*”, Worcester Polytechnic Institute, 2014.
- [11] R. A. Krisdiawan, “*Implementasi Pengembangan Sistem GDLS dan Algoritma Linear Congruential Generator pada Game Puzzle*”, Kuningan: Fakultas Ilmu Komputer Universitas Kuningan, 2018.
- [12] R. Rido dan W. Yani, “*Game Development Life Cycle Guidelines*”, Bandung: School of Electrical Engineering and Informatics, Institut Teknologi Bandung, 2013.



©2022. This article is an open access article distributed under the terms and conditions of the [Creative Commons Attribution-ShareAlike 4.0 International License](https://creativecommons.org/licenses/by-sa/4.0/).