

Perancangan Sistem Transaksi Non-Tunai Berbasis QR Code di Platform Android

Jay Idoan Sihotang^{1*}, Rawlson Yoshua Limbong²

^{1,2}Fakultas Teknologi Informasi, Universitas Advent Indonesia
e-mail: ^{1*}jay.sihotang@unai.edu, ²1681009@unai.edu

Abstrak

Dewasa ini perkembangan teknologi mempengaruhi berbagai aspek dalam kehidupan, termasuk dalam melakukan transaksi finansial. Di kampus Universitas Advent Indonesia, pembayaran masih menggunakan cara konvensional alias menggunakan uang kertas untuk pembayaran. Sehingga memungkinkan terjadinya kesalahan dalam pembayaran, maupun pengguna tidak bisa melacak *history* pembayaran yang dilakukan. Sistem informasi dalam bentuk aplikasi pembayaran yang dibangun diharapkan dapat menjadi alternatif pembayaran di kampus. Aplikasi pemrograman bergerak yang dibangun berdasarkan Analisa kebutuhan terhadap sistem lama, dan dibangun menggunakan Android dengan *database* menggunakan *firebase*. Proses pembayaran menggunakan validasi *QR Code*, baik untuk melakukan transaksi antar penjual dengan pengguna, maupun sesama pengguna. Hasil dari penelitian adalah sebuah aplikasi pembayaran berbasis *QR Code* yang dapat berjalan dengan baik dan teruji secara fungsional dan tampilannya.

Kata Kunci: Pemrograman Bergerak, Sistem Informasi, Aplikasi Pembayaran, *QR Code*, *Android*

The Design of QR Code Based Cashless Transaction System on Android Platform

Abstract

Nowadays the development of technology affects many aspects in life, including financial transaction. In Universitas Advent Indonesia, the payment still using conventional method alias using cash for payments. That causes some possibility of error on payments, and the users can't trace their payments history. The information system in a form of payment application is expected to become a payments alternative in campus. The mobile programming application was developed based on old system requirement analysis and developed using Android with Firebase as a database. The payment process is validated using QR Code, both to make transactions between sellers and users, as well as fellow users. The results of the study are a QR Code-based payment application that can run well and is tested both functionally and in appearance.

Keywords: *Mobile programming, Information System, Payment Application, QR Code, Android*

1. Pendahuluan

Dewasa ini perkembangan teknologi Informasi sangat pesat dan mempengaruhi banyak aspek dalam kehidupan manusia. Salah satu dari penggunaan teknologi informasi tersebut adalah dalam melakukan sistem pembayaran. Sesuai dengan definisinya, maka aplikasi/sistem yang dibangun bertujuan untuk mendukung dan mempermudah pengolahan atau transmisi informasi antar berbagai pihak [1] [2]. Di Universitas Advent Indonesia (UNAI), sistem pembayaran dari mahasiswa/staf/dosen kepada *merchant* /*vendor* masih menggunakan metode konvensional alias menggunakan uang tunai. Di mana memungkinkan adanya terjadi ketidakselarasan nominal transaksi, kesulitan ketika mengembalikan uang

kembali, maupun pembeli tidak dapat mengetahui riwayat transaksi yang sudah dilakukan. Maka dibutuhkan sebuah sistem yang dapat mencegah kesalahan dalam pencatatan atau perhitungan dalam membuat laporan keuangan [3]. Aplikasi yang dibangun dapat memudahkan transaksional antar penjual dan pembeli, dan diharapkan dapat menjadi alternatif transaksi finansial [4]. Dengan adanya sistem pembayaran ini, maka diharapkan pengguna tidak perlu lagi melakukan pencatatan secara manual dan hasil dari transaksi dapat dilihat secara langsung.

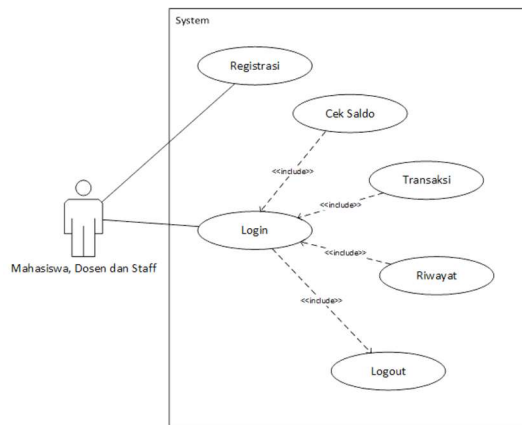
Pada sistem lama seluruh transaksi masih dilakukan secara manual. Manual yang dimaksud adalah seluruh pencatatan dan pembayaran dilakukan tidak menggunakan sistem yang terintegrasi langsung ke Universitas Advent. Sebagai contoh catatan masih dilakukan secara tertulis di buku dan setiap laporan masih diantar ke Biro Keuangan Mahasiswa. Untuk penjelasan detailnya, bisnis yang di analisa adalah *Laundry*, *Fotokopi*, *Bakery*, *OMI*, Klinik. Secara umum, semua pembayaran dilakukan secara manual seperti menotakan berapa biaya yang diberikan oleh kasir/pemilik, lalu membayarnya, memberikan bukti pembayaran dan mencatat transaksi yang dilakukan. Namun setiap departemen tidak memiliki alur yang sama. Hal inilah yang menyebabkan data yang ingin dimasukkan, mungkin tidak dicatat dengan baik atau laporan kurang lengkap

Aplikasi yang dibangun di atas sistem operasi Android menggunakan PHP sebagai *backend* yang memungkinkan terjadinya pertukaran data yang terintegrasi antara aplikasi dengan basis data [5]. Aplikasi *mobile* digunakan oleh lebih banyak pengguna, dan sistem operasi Android diyakini mumpuni dan aman [6] [7]. Penelitian ini menggunakan manajemen basis data *firebase* yang memungkinkan terjadinya pertukaran data secara *real-time* [8]. Dan untuk aplikasi *mobile*, sudah teruji bahwa *database Firebase* memiliki tingkat respon *real-time* yang lebih tinggi [9]. Dan penggunaan *QR Code* sebagai *validator* transaksi antara penjual dengan pembeli maupun antara sesama pengguna dikarenakan *QR Code* mampu menampung ukuran data yang lebih besar [10]. Pengembangan aplikasi pembayaran ini diharapkan dapat menjadi alternatif pembayaran di kampus, dan juga teruji secara fungsionalitas maupun tampilannya.

2. Metode Penelitian

Penelitian ini menggunakan metode *Waterfall* yang bersifat linear. Yaitu pengembangan aplikasi dimulai dari tahap perancangan sampai dengan tahap publikasi sebagai akhir dari pengembangan [11] [12]. Kebutuhan sistem yang dirancang bertujuan untuk memenuhi kebutuhan pengguna untuk dapat melihat saldo, melakukan transaksi, melakukan transfer, dan melihat riwayat transaksi. Aplikasi yang dibangun menggunakan perangkat keras *Intel Core i7* dengan RAM 16GB, dan lingkungan pengembangan aplikasi menggunakan *Android Studio*, dan dibantu dengan *Visual Studio Code*. Aplikasi dibangun dengan API Level Android di level 22, dengan tujuan agar banyak orang yang dapat menggunakannya.

Use Case Diagram

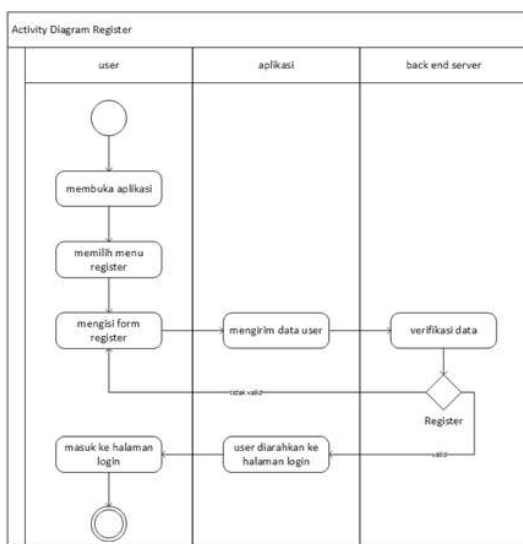


Gambar 1 Use Case Diagram

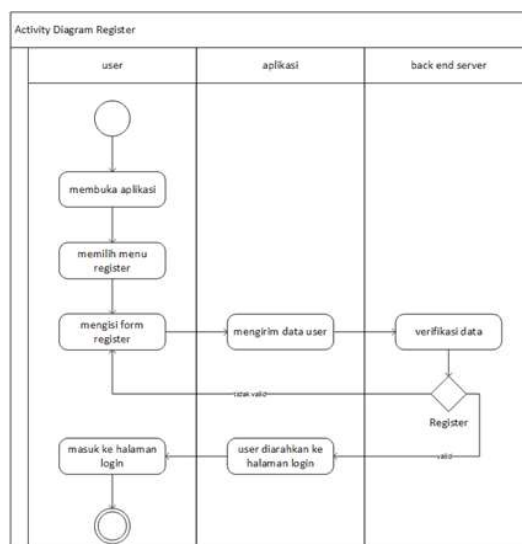
Gambar 1 menggambarkan interaksi antar pengguna dengan fungsi dalam aplikasi tersebut. Sistem baru akan mencakup beberapa fungsi. Di antaranya, fungsi register yang berguna untuk mendaftarkan pengguna, fungsi *login* yang memungkinkan pengguna masuk ke dalam aplikasi, fungsi cek saldo yang memberi informasi mengenai saldo yang dimiliki pengguna, fungsi transaksi untuk melakukan transaksi antara pengguna dengan vendor/penjual, fungsi transfer yang memungkinkan sesama pengguna saling bertukar atau transfer saldo, dan fungsi riwayat yang menampilkan informasi dari transaksi atau transfer yang dilakukan oleh pengguna.

Activity Diagram

Activity Diagram berguna untuk memodelkan interaksi antara pengguna dengan fungsi-fungsi yang ada secara bertahap. Dalam pengembangan aplikasi ini, terdapat beberapa *Activity Diagram* yaitu dalam kegiatan registrasi, *login*, cek saldo, transfer, transaksi, dan cek riwayat transaksi.



Gambar 2 Activity Diagram Registrasi

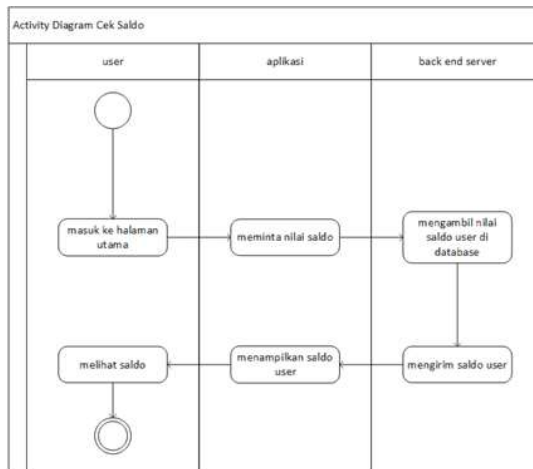


Gambar 3 Activity Diagram Login

Gambar 2 menjelaskan langkah-langkah yang harus dilakukan oleh pengguna sebelum dapat menggunakan aplikasi. Pengguna diharuskan untuk memiliki akun, dan pengguna dapat melakukan registrasi dengan memilih menu registrasi. Kemudian memasukkan data-data yang terdapat dalam *form*. Ketika data sudah selesai dimasukkan dan divalidasi oleh aplikasi, maka pengguna akan diarahkan ke halaman *login*. Namun ketika ditemukan suatu kesalahan/kekurangan dalam pengisian *form* registrasi, maka akan ditampilkan notifikasi *error*.

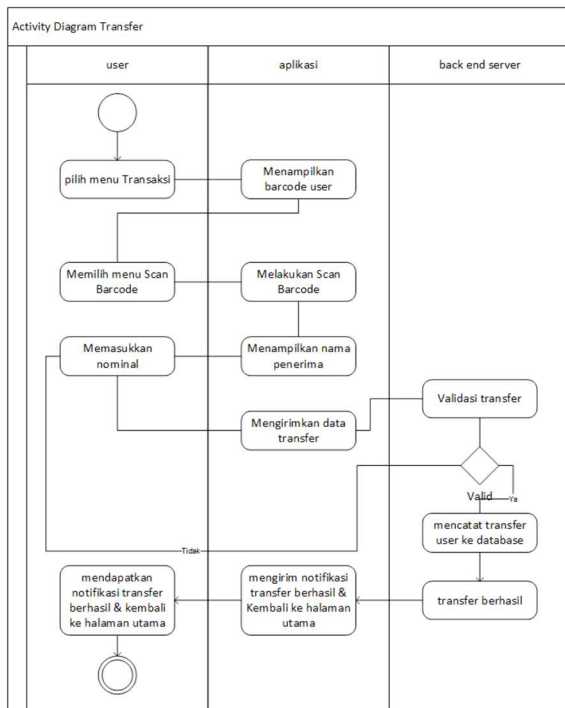
Activity Diagram Login (Gambar 3) menjelaskan mengenai tahapan yang dijalankan untuk memvalidasi apakah pengguna sudah memasukkan data *username* dan *password* yang benar. Jikalau ditemukan kesalahan dalam informasi yang dimasukkan, maka pengguna kembali diarahkan ke halaman *login*. Namun jikalau data yang dimasukkan benar, maka pengguna akan diarahkan ke halaman utama.

Di halaman utama, pengguna dapat berinteraksi dengan beberapa fungsi. Salah satunya adalah cek saldo. Di mana pengguna dapat melihat berapa sisa saldo yang dimiliki untuk dapat melakukan transaksi dengan penjual atau melakukan transfer dengan sesama pengguna. Mekanisme fungsi cek saldo diilustrasikan dalam Gambar 4 di bawah ini.

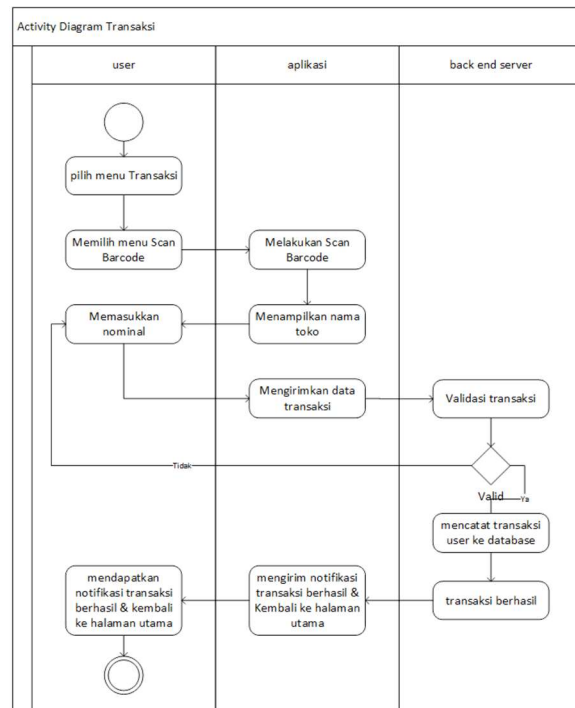


Gambar 4 Activity Diagram Cek Saldo

Terdapat dua buah *role* yang akan dilakukan oleh pengguna dalam menjalankan fungsi transfer yaitu sebagai pemberi atau sebagai penerima. Sebagai penerima, pengguna akan memilih menu transfer dan diarahkan ke halaman yang berisikan *QR Code* dari pengguna sebagai penerima, lalu menunjukkan *QR Code* tersebut kepada pengirim. Sebagai pengirim, pengguna akan memilih menu transfer dan tekan tombol *next*, kemudian akan masuk ke halaman transfer. Pengguna yang berperan sebagai pengirim memasukkan nominal yang akan dikirim, kemudian menekan tombol *scan barcode* untuk memindai *QR Code* dari penerima dana. Setelah memindai *QR Code* dari penerima, pengguna yang berperan sebagai pengirim menekan tombol konfirmasi untuk melakukan transfer. Secara otomatis sistem akan mencatat transaksi yang dilakukan, mengirimkan dana dari pengirim ke penerima, dan memberi informasi bahwa transfer yang dilakukan telah berhasil. Gambar 5 di bawah ini menggambarkan mekanisme transfer dana dari pengirim ke pengguna.

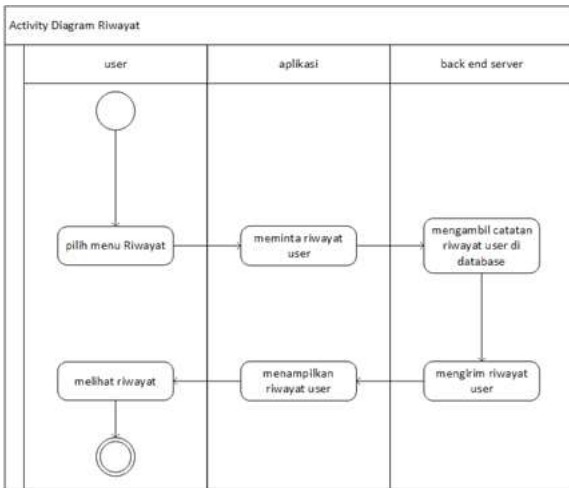


Gambar 5 Activity Diagram Transfer Dana

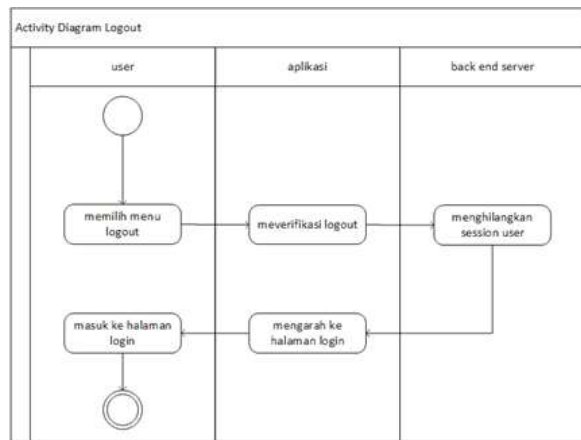


Gambar 6 Activity Diagram Transaksi

Gambar 6 mengilustrasikan mekanisme transaksi antara pengguna dengan penjual. Di mana untuk melakukan fungsi transaksi antara pengguna dengan penjual, pengguna dapat memilih menu transaksi di halaman utama. Kemudian pengguna akan diarahkan ke halaman yang berisikan *form* nominal yang akan dikirim. Untuk dapat memvalidasi pembayaran ke penjual, pengguna harus menekan tombol *scan barcode* yang membuka sebuah jendela pemindai yang digunakan untuk memindai *QR Code* dari penjual. Setelah *QR Code* penjual sudah dipindai dan nominal sudah dimasukkan oleh pengguna, maka pengguna perlu menekan tombol konfirmasi untuk melakukan transaksi. Aplikasi akan mengirimkan dana dari pengirim ke penjual, dan mencatat transaksi yang dilakukan. Pengguna akan mendapatkan notifikasi bahwa transaksi yang dilakukan berhasil.



Gambar 7 Activity Diagram Cek Riwayat

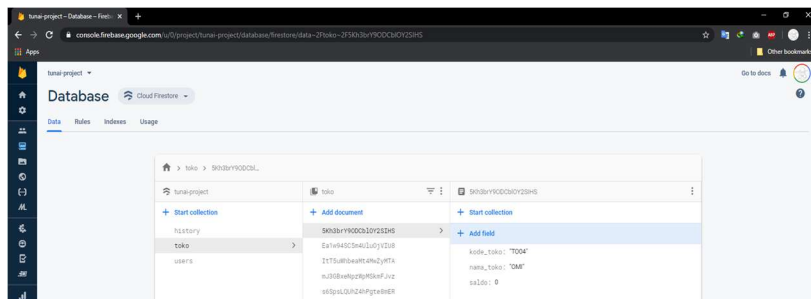


Gambar 8 Activity Diagram Logout

Pengguna dapat mengetahui riwayat transaksi/transfer yang dilakukan oleh pengguna dengan memilih menu riwayat pada halaman utama. Kemudian pengguna akan ditampilkan seluruh transaksi/transfer yang pengguna telah lakukan. *Activity Diagram* cek riwayat diilustrasikan dalam Gambar 7. Setelah pengguna selesai menggunakan aplikasi, pengguna dapat menjalankan fungsi *logout* sebagaimana diilustrasikan dalam Gambar 8. Pengguna akan memilih menu *logout* dari halaman utama, dan kemudian aplikasi akan memverifikasi *logout* dan menghapus sesi aktif dari pengguna. Pengguna akan kembali diarahkan ke halaman *login*.

Perancangan Database Firebase

Untuk dapat mengakses *database firebase*, dibutuhkan sebuah autentikasi terhadap pengguna yang aktif dalam sistem [13]. Data pengguna yang disimpan berupa *identifier* yang menjabarkan kapan pengguna melakukan registrasi, waktu terakhir *login*, serta UID dari pengguna. Dalam aplikasi ini autentikasi berguna untuk mengetahui apakah pengguna sudah terdaftar atau belum.

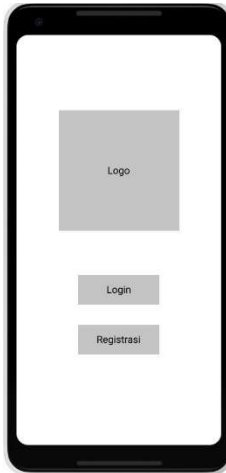


Gambar 9 Struktur Database Firebase

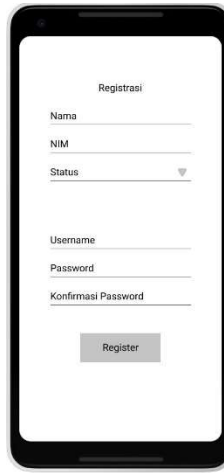
Database yang digunakan dalam penelitian ini adalah *Cloud Firestore* yang merupakan *database* non-SQL, dan dapat diakses di mana saja selama memiliki koneksi internet. *Database* ini menyimpan keseluruhan data dari aktivitas maupun entitas yang ada di dalam aplikasi. Mulai dari data pengguna, data toko, dan juga riwayat transaksi yang dilakukan di dalamnya.

Perancangan Antarmuka

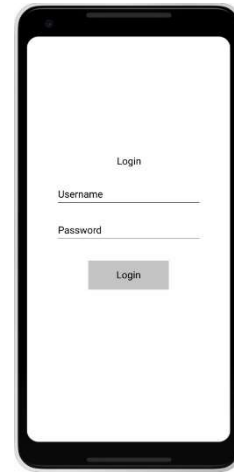
Perancangan Antarmuka (*User Interface / UI*) bertujuan untuk menggambarkan bagaimana tampilan-tampilan yang ada dalam aplikasi dimulai dari Halaman selamat datang sampai dengan menu di dalamnya. Tujuan ini dibuat agar dapat mempermudah pembaca mengerti tampilan aplikasinya.



Gambar 9 Perancangan Antarmuka Selamat Datang



Gambar 10 Perancangan Antarmuka Registrasi

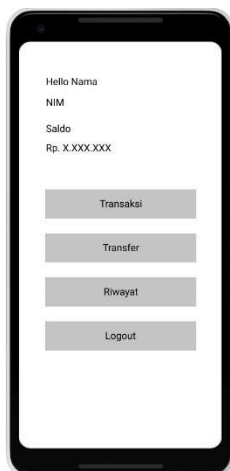


Gambar 11 Perancangan Antarmuka *Login*

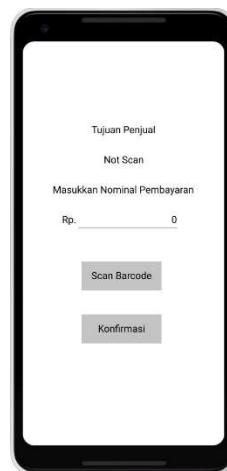
Pada bagian Selamat Datang (Gambar 9), *user* dapat memilih menu yang telah di tampilkan seperti *login* dan registrasi. Jika *user* belum pernah memakai atau terdaftar, maka *user* harus melakukan Registrasi. Jika *user* sudah terdaftar, maka *user* memilih tombol *login*. Ketika *user* memilih tombol registrasi, *user* harus mengisi *form* yang telah disediakan. Jika sudah terisi, maka *user* dapat menekan tombol *Register* untuk meng-*submit form* yang telah diisi. Sebagaimana diilustrasikan pada Gambar 10.

Gambar 11 mengilustrasikan halaman *login* yang muncul setelah *user* selesai melakukan registrasi. *user* harus mengisi bagian *username* dan *password* untuk masuk ke dalam Aplikasi. Jika sudah diisi oleh *user*, maka *user* dapat menekan tombol *login* untuk masuk ke dalam aplikasi.

Pada halaman Utama (Gambar 12), di dalam tampilan ini akan ada nama *user*, nomor induk, dan saldo dari *user*. Selain informasi *user* ada juga tombol seperti transaksi, transfer, riwayat, dan *logout*. *User* akan diarahkan ke halaman Transaksi (Gambar 13) setelah menekan tombol transaksi. Pada bagian ini, *user* dapat membuka *scanner QR Code* dan juga melakukan *input* nilai dari pembayaran yang akan dibayar. Setelah itu, *user* dapat menekan tombol konfirmasi untuk melakukan pembayaran.

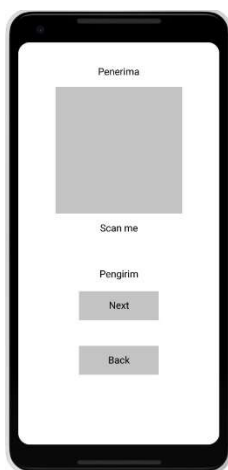


Gambar 12 Perancangan Antarmuka Halaman Utama

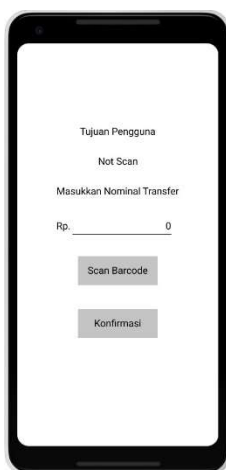


Gambar 13 Perancangan Antarmuka Transaksi

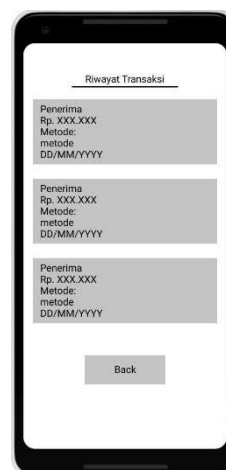
User dapat memilih tombol *Transfer* untuk melakukan transfer ke sesama *user* dan diarahkan ke halaman menu transfer (Gambar 14). Pada bagian ini, *user* dapat melihat *QR Code* yang dibuat oleh aplikasi dan selain itu tujuan dari *QR Code* yang ditampilkan oleh aplikasi bertujuan untuk menerima transferan dari *user* yang akan mengirim ke *penerima*. Selain itu pengirim dapat menekan tombol *next* untuk masuk ke halaman selanjutnya. Dan tombol *back* bertujuan untuk balik ke halaman sebelumnya. Setelah menekan tombol *next*, *user* pengirim akan mengarahkan *QR Codenya* ke *barcode* penerima yang ada pada menu transfer dan setelah itu memasukkan berapa nominal yang akan ditransfer. Setelah melakukan *input*, maka *user* dapat menekan tombol konfirmasi untuk melakukan transfer. Ilustrasi halaman transfer terdapat pada Gambar 15. Pengguna dapat melihat riwayat dengan menekan tombol riwayat pada halaman utama, dan diarahkan ke halaman Riwayat (Gambar 16). Pada bagian ini, *user* dapat melihat riwayat seperti pembayaran ke toko/vendor mana dan transfer ke *user* lain. Tombol *back* bertujuan untuk kembali ke halaman utama.



Gambar 14 Perancangan Antarmuka Menu Transfer



Gambar 15 Perancangan Antarmuka Transfer

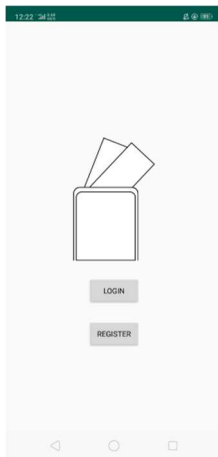


Gambar 16 Perancangan Antarmuka Riwayat

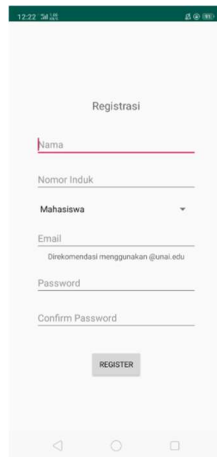
3. Hasil dan Pengujian Sistem

Saat pertama kali aplikasi dibuka, pengguna akan dihadapkan di halaman selamat datang (Gambar 17) dengan pilihan untuk *login* atau *register*. Jika pengguna pertama kali menggunakan aplikasi ini, maka diwajibkan untuk menekan tombol *register* untuk mendaftar. Jika pengguna sudah pernah menggunakan ini, maka pengguna dapat menekan tombol *login* untuk masuk ke dalam aplikasi.

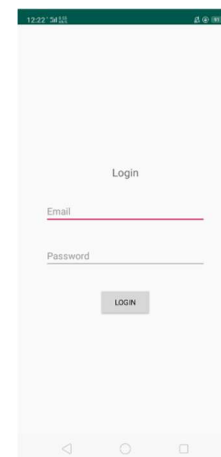
Ketika *user* memilih tombol *register*, pengguna akan mengisi *form* yang bertujuan untuk mendaftarkan diri ke dalam sistem di halaman registrasi (Gambar 18). Ada beberapa hal yang perlu di isi seperti nama, *username* dan lain sebagainya. Setelah meng-*input* data, maka pengguna dapat menekan tombol Registrasi untuk mendaftar dan akan diarahkan ke halaman *login*. Ketika masuk di halaman *login* (Gambar 19), pengguna akan dihadapkan dengan *form login*. Halaman *login* bertujuan untuk mengautentikasi pengguna yang akan masuk ke dalam sistem. Setelah pengguna memasukkan *username* dan *password*, maka aplikasi akan mengecek apakah pengguna sudah terdaftar atau belum. Jika salah satu dari *username* atau *password* yang di-*input* tidak terdaftar atau tidak sesuai dengan yang ada di dalam *database*, maka aplikasi akan tetap di halaman *login*. Namun, jika *username* dan *password* benar, maka pengguna akan masuk ke dalam halaman utama.



Gambar 17 Halaman Selamat Datang



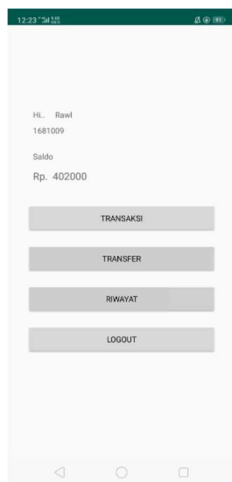
Gambar 18 Halaman Registrasi



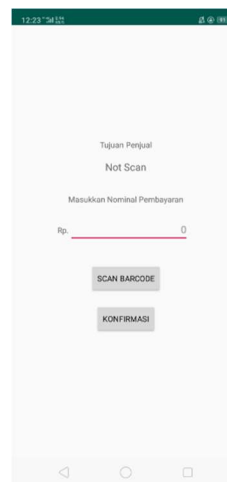
Gambar 19 Halaman *Login*

Halaman utama (Gambar 20) ditampilkan setelah pengguna masuk ke dalam sistem melalui halaman *login*. Halaman ini berisikan menu seperti transaksi, transfer, riwayat dan *logout*. Selain itu, pada tampilan ini akan ditampilkan saldo pengguna.

Ketika pengguna menekan tombol transaksi, pengguna akan diarahkan ke halaman transaksi (Gambar 21). Pada halaman ini, *user* dapat melakukan *scan QR Code* dari toko yang di inginkan. Setelah melakukan *scan* maka *user* harus meng-*input* berapa nilai yang nominal yang harus dibayar untuk melakukannya. Nominal akan diberikan oleh penjual untuk. Setelah itu, *user* dapat menekan tombol konfirmasi dan *user* akan diarahkan ke halaman utama.



Gambar 20 Halaman Utama

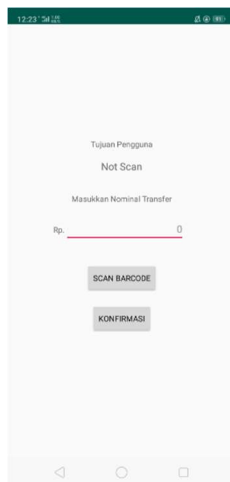


Gambar 21 Halaman Transaksi

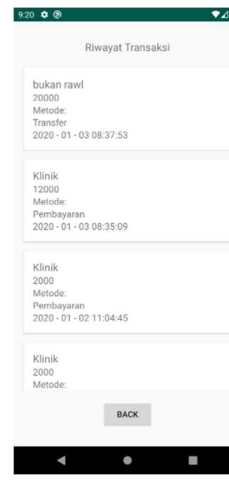
Dan pengguna dapat melakukan transfer dana antar sesama pengguna dengan menekan tombol transfer pada halaman utama, dan pengguna akan diarahkan ke halaman menu transfer (Gambar 22). Pada bagian ini, *user* akan ditampilkan sebuah *QR Code* yang telah dibuat oleh aplikasi. *Barcode* tersebut berasal dari *database* dan di bentuk oleh aplikasi untuk di tampilkan. Selain *QR Code* yang telah dibuat oleh aplikasi, ada tombol *next* yang bertujuan untuk ke menu selanjutnya dari transfer. Dan juga tombol *back* untuk kembali ke halaman sebelumnya. Dan pada halaman transfer (Gambar 23) yang muncul setelah pengguna menekan tombol *next* pada menu transfer, data yang ditampilkan tidak banyak yang berbeda dengan tampilan di halaman transaksi, perbedaan dari halaman transfer ini digunakan untuk melakukan transfer ke sesama *user*. Jadi jika di gunakan ke toko, maka aplikasi tidak menerima *barcode* toko. Pengguna dapat melihat riwayat transaksi di menu transaksi dengan cara menekan tombol riwayat, dan aplikasi akan mengarahkan tampilan ke halaman riwayat (Gambar 24).



Gambar 22 Halaman Menu Transfer



Gambar 23 Halaman Transfer



Gambar 24 Halaman Riwaya

Application Programming Interface (API)

Aplikasi ini menggunakan beragam *Application Programing Interface (API)* untuk membuatnya berjalan dengan baik. Sebagai contoh autentikasi dan *database*-nya. Karena aplikasi berjalan dalam sistem

Android maka bahasa yang digunakan adalah *Java*. *Firebase* memberikan bagaimana cara menggunakan *API* yang siap digunakan oleh pengembang untuk membuat aplikasi.

API yang akan digunakan untuk autentikasi ada beberapa dimulai dari registrasi, *login* dan juga selama berada dalam aplikasi. Berikut contoh – contoh dari inti penggunaannya.

```
// RegisterActivity.java

// Some Code
 mAuth.createUserWithEmailAndPassword(email,password)
 .addOnCompleteListener(new OnCompleteListener<AuthResult>() {
// Some Code
}
// Some Code
```

Pada bagian ini, *API* yang digunakan bertujuan untuk melakukan registrasi yang akan dicatat dalam *Authentication* yang disediakan oleh *Firebase*. Dengan menggunakan kata kunci "*createUserWithEmailAndPassword()*" maka pengguna yang akan terdaftar akan dicatat seperti *Identifier*, *Providers*, *Created*, *Signed In* dan *User UID*. Namun pada bagian *Signed In* tidak akan ada nilai yang dicatat karena pengguna belum melakukan *login*. Jika pengguna sudah terdaftar, maka *API* akan memberikan respon bahwa pengguna yang lain sudah menggunakan *Identifier* yang sama, atau gagal mengirimkan data dan lain sebagainya.

```
// LoginActivity.java

// Some Code
 mAuth.signInWithEmailAndPassword(email,password)
 .addOnCompleteListener(LoginActivity.this, new OnCompleteListener<AuthResult>() {
// Some Code
}
// Some Code
```

Pada bagian ini, *API* yang digunakan bertujuan untuk melakukan *login* yang akan dicek dalam *Authentication* yang disediakan oleh *Firebase*. Dengan menggunakan kata kunci "*signInWithEmailAndPassword()*" Jika pengguna sudah terdaftar, maka pada bagian *Signed In*, akan dimasukkan data kapan pengguna telah berhasil melakukan *login*. Namun jika ada kesalahan, maka *API* akan memberikan respon seperti sedang *offline*, kesalahan *email* atau *password* dan lain sebagainya.

```
// HomeActivity.java

// Some Code
 private FirebaseAuth currentUser;
 String currentUID;
// Some Code
 currentUser = FirebaseAuth.getInstance().getCurrentUser();
 currentUID = currentUser.getUid();
// Some Code
```

Pada bagian ini, *API* yang digunakan bertujuan untuk mengambil data apakah pengguna telah *login* atau tidak. Jika tidak ada, maka *API* akan memberhentikan proses karena tidak dapat mendapatkan hasil apakah pengguna sudah *login*. Setelah *API* mendapatkan apakah pengguna sudah *login* atau tidak, selanjutnya aplikasi akan meminta data dari *Authentication* untuk mengambil *User UID*. Tujuannya adalah untuk membantu pengambilan data dari *Database*.

API yang digunakan untuk *database* ada beberapa cara untuk menggunakannya. Berikut beberapa contoh dari penggunaannya.

```
// HomeActivity.java

// Some Code
 private FirebaseFirestore userState = FirebaseFirestore.getInstance();
// Some Code
```

```
DocumentReference getUserProfile = userState.collection("users").document(currentUID);
getUserProfile.get().addOnSuccessListener(new OnSuccessListener<DocumentSnapshot>() {
// Some Code
});
// Some Code
```

Pada bagian ini *API* yang digunakan bertujuan untuk mengambil data dari *database*. Jika berhasil, maka aplikasi dapat menampilkan data yang telah diambil dari *database*. Namun jika terjadi kesalahan, maka *API* akan memberikan respon sesuai dengan apa yang menjadi kendalanya.

Pengujian Sistem

Sistem yang telah dibentuk atau dibangun akan diuji untuk mengetahui apakah seluruh sistem dapat menghasilkan data yang diinginkan dan sesuai dengan tujuan sistem ini dibentuk. Berikut adalah metode pengujian dari masing-masing sistem yang diuji.

Tabel 1 Metode Pengujian Sistem

No.	Sistem yang Diuji	Keterangan	Metode Pengujian
1	Halaman selamat datang	Tombol Registrasi	<i>Black Box</i>
		Tombol <i>Login</i>	
2	Halaman Registrasi	Tombol Registrasi	<i>Black Box</i>
		Kesalahan data Registrasi	
3	Halaman <i>Login</i>	Tombol <i>Login</i>	<i>Black Box</i>
		Kesalahan data <i>Login</i>	
4	Halaman Utama	Data Pengguna	<i>Black Box</i>
		Tombol Transaksi	
		Tombol Transfer	
		Tombol Riwayat	
5	Halaman Menu Transfer	Informasi <i>Barcode</i> Pengguna	<i>Black Box</i>
		Tombol <i>Next</i>	
		Tombol <i>Back</i>	
6	Halaman Transfer	Tombol <i>Scan Barcode</i>	<i>Black Box</i>
		Tombol Konfirmasi	
7	Halaman Transaksi	Informasi <i>Scan Barcode</i>	<i>Black Box</i>
		Tombol <i>Scan Ulang</i>	
		Tombol <i>Text</i>	
		Tombol <i>Back</i>	
8	Halaman Riwayat	Informasi Riwayat	<i>Black Box</i>
		Tombol <i>Back</i>	

Dan berikut ini adalah hasil dari pengujian dari sistem yang menggunakan metode blackbox pada tiap sistem yang diuji.

Tabel 2 Hasil Pengujian Sistem

No.	Sistem yang Diuji	Proses Uji	Hasil yang Diharapkan	Hasil Pengujian
1	Halaman selamat datang	Menekan Tombol Registrasi	Jika ditekan, akan diarahkan ke halaman Registrasi	"Diterima"
		Menekan Tombol <i>Login</i>	Jika ditekan, maka diarahkan ke halaman <i>login</i>	"Diterima"
2	Halaman Registrasi	Menekan Tombol Registrasi	Jika ditekan, maka data akan di simpan ke dalam <i>database</i> dan pengguna dipindahkan ke halaman <i>login</i>	"Diterima"

		Kesalahan data Registrasi	Jika data yang di- <i>input</i> kosong atau kurang, akan muncul peringatan kesalahan	"Diterima"
3	Halaman <i>Login</i>	Menekan Tombol <i>Login</i>	Jika data yang di- <i>input</i> benar, maka akan diarahkan ke halaman utama	"Diterima"
		Kesalahan data <i>Login</i>	Jika data yang di- <i>input</i> kosong atau salah, maka akan muncul peringatan.	"Diterima"
4	Halaman Utama	Menampilkan Data Pengguna	Akan muncul data pengguna yang <i>login</i>	"Diterima"
		Menekan Tombol Transaksi	Jika ditekan, maka akan diarahkan ke halaman transaksi.	"Diterima"
		Menekan Tombol Transfer	Jika ditekan, maka akan diarahkan ke halaman transfer	"Diterima"
		Menekan Tombol Riwayat	Jika ditekan, maka akan diarahkan ke halaman riwayat.	"Diterima"
		Menekan Tombol <i>Logout</i>	Jika ditekan, maka pengguna yang sedang <i>login</i> , akan diarahkan ke halaman <i>login</i> .	"Diterima"
5	Halaman Menu Transfer	Informasi <i>Barcode</i> Pengguna	Akan menampilkan <i>barcode</i> yang sesuai dengan pengguna yang sedang <i>login</i>	"Diterima"
		Menekan Tombol <i>Next</i>	Jika ditekan, akan diarahkan ke halaman transfer selanjutnya	"Diterima"
		Menekan Tombol <i>Back</i>	Jika ditekan, akan diarahkan ke halaman sebelumnya.	"Diterima"
6	Halaman Transfer	Tombol <i>Scan Barcode</i>	Jika ditekan, maka akan melakukan <i>scan barcode</i> dan melakukan <i>scan</i>	"Diterima"
		Menekan Tombol Konfirmasi	Setelah memasukkan nominal yang akan ditransfer, maka data transfer akan dicatat dan kembali ke halaman utama.	"Diterima"
7	Halaman Transaksi	Tombol <i>Scan Barcode</i>	Jika ditekan, maka akan melakukan <i>scan barcode</i> dan melakukan <i>scan</i>	"Diterima"
		Menekan Tombol Konfirmasi	Setelah memasukkan nominal yang akan dibayar, maka data transaksi akan dicatat dan kembali ke halaman utama.	"Diterima"
8	Halaman Riwayat	Informasi Riwayat	Data riwayat yang telah dilakukan oleh pengguna, akan ditampilkan	"Ditolak"
		Menekan Tombol <i>Back</i>	Jika ditekan, maka akan diarahkan ke halaman sebelumnya	"Diterima"

Menurut hasil pengujian yang dilakukan, maka penulis mendapat kesimpulan bahwa secara dominan sistem berjalan sesuai dengan fungsinya dan dapat mengirim dan menerima data yang diminta.

4. Pembahasan/Kesimpulan

Berdasarkan analisa, perancangan dan pengujian yang telah dilakukan di Universitas Advent Indonesia berdasarkan bab sebelumnya, maka dapat disimpulkan bahwa: 1. Aplikasi pembayaran yang sudah dibuat

dapat digunakan melalui *Smartphone* berbasis Android. 2. Aplikasi pembayaran yang sudah dibuat dapat memenuhi kebutuhan yang ada seperti pembayaran, transfer dan melihat riwayat. 3. Aplikasi pembayaran yang sudah dirancang belum dapat memberikan beberapa data atau informasi yang holistik kepada pengguna dan terdapat beberapa fungsi yang sudah dibuat, namun belum dapat memberikan hasil yang sesuai dengan kebutuhan.

Untuk pembaca, pengguna dan pengembang sistem ke depannya agar dapat: 1. Diharapkan untuk pengembang selanjutnya, dapat mengembangkan *User Interface* dan *User Experience* yang lebih baik. 2. Karena aplikasi yang dibuat menggunakan *Database* NonSQL, maka banyak hal yang harus diubah seperti penstrukturan data dan jenis data yang akan disimpan. 3. Untuk penelitian selanjutnya diharapkan peneliti menggunakan *React-Native* atau *Expo-cli* hal ini disebabkan karena versi dari *React-Native* berkembang dengan pesat. Disarankan untuk peneliti selanjutnya, dapat memperhatikan setiap perubahan yang terjadi. 4. Menambahkan fungsi keamanan dari sistem transaksi agar semakin lengkap.

5. Referensi

- [1] J. Hutahean, *Konsep Sistem Informasi*, Yogyakarta: Deepublish Publisher, 2015.
- [2] T. Sutabri, *Konsep Sistem Informasi*, Bandung: Penerbit Andi, 2012.
- [3] S. F. Saragih and E. B. Wagiu, "Analisa Perencanaan Pembayaran Menggunakan Sistem *QR Code* di Industrial Universitas Advent Indonesia," *TeIka*, vol. 09, no. 01, pp. 15-29, Oktober 2019.
- [4] J. I. Sihotang, "Perancangan Sistem Transaksi Berbasis Near Field Communication (NFC) Dengan Sistem Operasi Android di Toko Virtual," in *Konferensi Nasional Sistem Informasi*, Makassar, 2014.
- [5] B. F. M. Sitompul, T. W. Krisna and V. W. J. Supit, *Perancangan Sistem Informasi MYUNAI Berbasis Mobile Application Android dan IOS Menggunakan Framework React Native*, Bandung: Fakultas Teknologi Informasi Universitas Advent Indonesia, 2019.
- [6] L. Collins and S. R. Ellis, *Mobile Devices: Tools and Technologies*, UK: CRC Press, 2015.
- [7] R. Singh, "An Overview of Android Operating System and Its Security," *Int. Journal of Engineering Research and Applications*, vol. 4, no. 2, pp. 519-521, 2014.
- [8] H. M. Christon, *Perancangan Sistem E-Learning Berbasis Android*, Bandung: Fakultas Teknologi Informasi Universitas Advent Indonesia, 2017.
- [9] M. Ohyver, J. V. Moniaga, I. Sungkawa, B. E. Subagyo and I. A. Chandra, "The Comparison *Firebase* Realtime *Database* and *MySQL Database* Performance using Wilcoxon Signed-Rank Test," *Procedia Computer Science*, vol. 157, pp. 396-405, 2019.
- [10] A. Qashlim and Hasruddin, "Implementasi Teknologi QR-Code untuk Kartu Identitas," *Jurnal Ilmiah Ilmu Komputer*, vol. 1, no. 2, pp. 1-6, 2015.
- [11] R. Susanto and A. Andriana, "PERBANDINGAN MODEL *WATERFALL* DAN *PROTOTYPING* UNTUK PENGEMBANGAN SISTEM INFORMASI," *Majalah Ilmiah UNIKOM*, vol. 14, no. 1, p. 174, 2016.
- [12] T. S. Jaya and D. Sahlinal, "Perancangan Kantor Digital Berbasis *Framework* dengan Metode *Waterfall* pada Politeknik Negeri Lampung," *Jurnal Informatika: Jurnal Pengembangan IT*, vol. 2 no. 2, pp. 14-17, 2017.
- [13] Google, "*Firebase*," Google, 2020. [Online]. Available: <https://firebase.google.com/>. [Accessed 2020 Januari 07].