



MOBILE-BASED BUS RAPID TRANSIT (BRT) PAYMENT INFORMATION SYSTEM

Andi Muh. Akbar Saputra^a, Ashabul Taufik^b

^a Fakultas Ilmu Keguruan dan Ilmu Pendidikan / Pendidikan Teknologi Informasi, akbarsaputra@uim-makassar.ac.id, Universitas Islam Makassar

^b Fakultas Ilmu Keguruan dan Ilmu Pendidikan / Pendidikan Teknologi Informasi, ashabultaufik@uim-makassar.ac.id, Universitas Islam Makassar

ABSTRACT

With current technological developments, mobile phones can provide Bus Rapid Transit (BRT) transportation payment systems that still use ineffective tickets where passengers have to queue to buy ticket tickets. As well as the physical condition of the ticket which is easily wet and torn. In fact, currently there are many payment systems that use technology assistance as a payment method that can minimize queues for purchasing transportation tickets. Such as smart cards used by Trans Jakarta transportation and the use of smartphones as a means of payment for Go-Jek and Grab transportation. So we need an innovation that can replace the BRT payment system which still uses tickets using the current technology. For this reason, a mobile-based Bus Rapid Transit (BRT) payment system was created so that passengers can easily make transportation payments. This system can validate voucher codes and make payments.

Keywords: *Payment System, BRT, Android, PHP, Web monitoring.*

ABSTRAK

Dengan perkembangan teknologi saat ini, mobile phone dapat menyajikan Sistem pembayaran transportasi Bus Rapi Transit (BRT) yang masih menggunakan karcis tidak efektif dimana penumpang harus mengantri untuk membeli tiket karcis. Serta kondisi fisik karcis yang mudah basah dan sobek. Padahal, saat ini sudah banyak sistem pembayaran yang menggunakan bantuan teknologi sebagai metode pembayaran yang dapat meminimalisir antrian pembelian tiket transportasi. Seperti smart card yang digunakan oleh transportasi Trans Jakarta dan penggunaan smartphone sebagai alat pembayaran transportasi Go-Jek dan Grab. Maka diperlukan sebuah inovasi yang dapat menggantikan sistem pembayaran BRT yang masih menggunakan karcis dengan menggunakan teknologi yang ada saat ini. Untuk itu dibuatlah sebuah sistem pembayaran Bus Rapi Transit (BRT) berbasis mobile agar penumpang dengan mudah melakukan pembayaran transportasi. Sistem ini dapat melakukan validasi kode voucher serta melakukan pembayaran..

Kata Kunci: Sistem Pembayaran, BRT, Android, PHP, Web monitoring

1. PENDAHULUAN

Keberhasilan pembangunan sangat dipengaruhi oleh peran transportasi sebagai urat nadi kehidupan politik, ekonomi, sosial budaya, dan pertahanan keamanan. Sistem jaringan transportasi dapat dilihat dari segi efektifitas, dalam arti selamat, aksesibilitas tinggi, terpadu, kapasitas mencukupi, teratur, lancar dan cepat, mudah dicapai, tepat waktu, nyaman, tarif terjangkau, tertib, aman, rendah polusi serta dari segi efisiensi dalam arti beban publik rendah dan rutinitas tinggi dalam satu kesatuan jaringan sistem transportasi. Dalam menunjang dan menggerakkan dinamika pembangunan, Transportasi berfungsi sebagai katalisator dalam mendukung pertumbuhan ekonomi dan pengembangan wilayah. Transportasi juga memiliki fungsi strategis dalam merekat integritas satu wilayah. Jika dilihat dari aspek kepentingan publik, sistem transportasi yang meliputi transportasi darat mengemban fungsi pelayanan publik dalam suatu wilayah.

Pengembangan Transportasi harus didasarkan pada pengembangan yang berkelanjutan yaitu melihat jauh ke depan, berdasarkan perencanaan jangka panjang yang komprehensif dan berwawasan lingkungan. Perencanaan jangka pendek harus didasarkan pada pandangan jangka panjang, sehingga tidak

terjadi perencanaan bongkar pasang. Untuk menanggulangi kemacetan lalu lintas, pemerintah telah melakukan berbagai langkah baik berupa menyusun kebijakan, tindakan, dan aspek hukum.

Hasilnya berupa pembangunan dan pengembangan prasarana jalan. Untuk mengurangi masalah transportasi yang ada di Indonesia khususnya di Kota Makassar, yang dilakukan pemerintah adalah memperbaiki sarana transportasi massal yang ada dengan transportasi baru yang bernama Bus Rapid Transportasi (BRT). Jika pelayanan transportasi massal sudah baik maka orang akan beralih ke transportasi massal dari pada transportasi pribadi, dengan peralihan tersebut maka secara tidak langsung kebutuhan mengenai informasi dan tiket sangat dibutuhkan, baik masyarakat Makassar atau pun masyarakat pendatang.

2. TINJAUAN PUSTAKA

2.1. Aplikasi *mobile*

Aplikasi *mobile* adalah sebuah aplikasi yang memungkinkan Anda melakukan mobilitas dengan menggunakan perlengkapan seperti PDA, telepon seluler atau Handphone. Dengan menggunakan aplikasi *mobile*, Anda dapat dengan mudah melakukan berbagai macam aktifitas mulai dari hiburan, berjualan, belajar, mengerjakan pekerjaan kantor, browsing dan lain sebagainya. Pemanfaatan aplikasi *mobile* untuk hiburan paling banyak digemari oleh hampir 70% pengguna telepon seluler, sampai video player membuat kita menjadi semakin mudah menikmati hiburan kapan saja dan dimanapun.

2.2. Bahasa Pemrograman

Bahasa Pemrograman (*programming language*) adalah sebuah instruksi standar untuk memerintah komputer agar mempunyai fungsi tertentu. Bahasa pemrograman ini merupakan suatu himpunan dari aturan sintaks dan semantik yang dipakai untuk mendefinisikan program komputer. Bahasa ini memungkinkan seorang programmer dapat menentukan secara persis data mana yang akan diolah oleh komputer, bagaimana data ini akan disimpan/diteruskan, dan jenis langkah apa secara persis yang akan diambil dalam berbagai situasi.

2.2.1 PHP

PHP atau Hypertext Preprocessor adalah pemrograman interpreter yaitu proses penerjemahan baris kode sumber menjadi kode mesin yang dimengerti komputer secara langsung pada saat baris kode dijalankan. PHP disebut juga pemrograman server side programming, hal ini dikarenakan seluruh prosesnya dijalankan pada server. Salah satu kelebihan dari PHP adalah mampu untuk berkomunikasi dengan database. Dengan demikian, menampilkan data yang bersifat dinamis, yang diambil dari database, merupakan hal yang mudah untuk diimplementasikan. Itulah sebabnya sering dikatakan bahwa PHP sangat cocok untuk membangun halaman-halaman web dinamis. Beberapa kelebihan PHP dibandingkan dengan bahasa pemrograman yang lain, yaitu :

1. Cara koneksi dan query database yang sederhana
2. Dapat bekerja pada sistem operasi berbasis Windows, Linux, Mac OS, dan kebanyakan varian UNIX.
3. Biaya yang dibutuhkan untuk menggunakan PHP tidak mahal bahkan gratis.
4. Mudah dibangun karena memiliki fitur dan fungsi khusus untuk membuat web dinamis. Bahasa pemrograman PHP dirancang untuk dapat dimasukkan dalam HTML (*embedded script*)
5. Security system yang cukup tinggi.
6. Waktu eksekusi yang lebih cepat dibandingkan dengan bahasa pemrograman web lainnya yang berorientasi pada *serverside scripting*.
7. Akses ke sistem database yang lebih fleksibel dan mudah, seperti pada MySQL.

2.2.2 Java

Java adalah bahasa pemrograman yang dapat membuat seluruh bentuk aplikasi, desktop, web, mobile dan lainnya, sebagaimana dibuat dengan menggunakan bahasa pemrograman konvensional yang lain. Bahasa Pemrograman Java ini berorientasi objek (OOP-Object Oriented Programming), dan dapat dijalankan pada berbagai platform sistem operasi. Perkembangan Java tidak hanya terfokus pada satu sistem operasi, tetapi dikembangkan untuk berbagai sistem operasi dan bersifat open source. Dengan slogannya “Write once, run anywhere”. Bahasa ini banyak mengadopsi sintaksis yang terdapat pada C dan C++ namun dengan sintaksis model objek yang lebih sederhana. Aplikasi-aplikasi berbasis java umumnya dikompilasi ke dalam p-code (bytecode) dan dapat dijalankan pada berbagai Mesin Virtual Java (JVM). Java merupakan bahasa pemrograman yang bersifat umum/non-spesifik (general purpose).

Paradigma OOP menyelesaikan masalah dengan merepresentasikan masalah ke model objek. Contoh Pemisalan Objek dalam OOP : Objek-objek dalam dunia nyata, mempunyai 2 karakteristik khusus : Status dan Perilaku. Contohnya, Laptop punya status (jumlah baterai, jumlah ram, sebuah layar) dan perilaku (mengeksekusi, menonton, mengerjakan). Bahasa yang berorientasi pada objek pun mempunyai karakteristik yang sama dengan objek-objek di dunia nyata. Yaitu status yang dalam bahasa pemrograman biasanya disimpan sebagai Variabel dan perilaku yang diimplementasikan sebagai Method.

2.3 Database

Database adalah kumpulan data yang terintegrasi dan diatur sedemikian rupa sehingga data tersebut dapat dimanipulasi, diambil, dan dicari secara cepat. database sebagai kumpulan informasi yang bermanfaat dan terorganisir. Database merupakan suatu kumpulan dari data – data yang saling berhubungan satu dengan yang lainnya, tersimpan di perangkat keras (hardware) komputer dan digunakan perangkat lunak (software) untuk memanipulasinya. Data disimpan dalam database untuk keperluan penyediaan informasi, diorganisasikan untuk efisiensi kapasitas penyimpanan supaya informasi yang dihasilkan berkualitas. Database diakses atau dimanipulasikan dengan menggunakan paket software Database Manajemen System (DBMS). Dalam praktek, penggunaan istilah database lebih dibatasi pada arti implisit yang khusus, yaitu :

- a. Basis data merupakan penyajian suatu aspek dari dunia nyata.
- b. Basis data merupakan kumpulan data dari berbagai sumber yang secara logika mempunyai arti implisit , sehingga data yang terkumpul secara acak dan tanpa mempunyai arti, tidak dapat disebut sebagai database.
- c. Database perlu dirancang,dibangun, dan data dikumpulkan untuk suatu tujuan. Database dapat digunakan oleh beberapa pemakai dan beberapa aplikasi yang sesuai dengan kepentingan pemakai.

Dari batasan tersebut diatas, database mempunyai berbagai sumber data dalam pengumpulan data, bervariasi derajat interaksi kejadian dari dunia nyata. Dirancang dan dibangun agar dapat digunakan oleh beberapa pemakai untuk berbagai kepentingan. Penyusunan database bisa dilakukan secara sederhana dengan menuliskan data-datanya, atau bisa juga dengan cara modern dengan menggunakan teknologi komputer. Cara apapun yang digunakan, untuk memudahkan penyusunan dan penampilan data, maka digunakan bentuk tabel. Sebuah tabel terdiri dari baris dan kolom. Dalam dunia database, baris distilahkan sebagai Record atau Tupple dan kolom distilahkan sebagai Field atau Attribute. Saat ini tersedia banyak perangkat lunak yang ditujukan untuk mengelola database. Perangkat lunak seperti itu biasa dinamakan DBMS (database management system). Access, MS SQL Server, dan MySQL merupakan beberapa contoh produk pengelola database. Beberapa diantaranya berkelas database server, yaitu jenis yang secara aktif memantau permintaan akses terhadap data. Dalam hal ini, database server akan segera menanggapi permintaan data.

2.3.1 MySQL

MySQL merupakan software relational database management system(RDBMS) atau database server yang dapat mengelola database dengan sangat cepat, dapat menampung data dalam jumlah yang sangat besar, dapat diakses oleh banyak user (multi-user), dan dapat melakukan suatu proses secara sinkron atau bersamaan (multi-threaded). Lisensi MySQL terbagi menjadi dua, yaitu MySQL sebagai produk open source di bawah GNU General Public License (gratis) dan produk komersial yang tentu memiliki nilai lebih atau kemampuan yang tidak disertakan pada versi gratis. Pada kenyataannya, untuk keperluan industri menengah ke bawah, produk open source masih dapat digunakan dengan baik. Open source menyatakan bahwa software ini dilengkapi dengan source code (kode yang dipakai untuk membuat MySQL), selain tentu saja bentuk executable-nya atau kode yang dapat dijalankan secara langsung dalam sistem operasi, dan dapat diperoleh dengan cara mengunduh di Internet secara gratis. Pengaksesan data dalam database dapat dilakukan dengan mudah melalui SQL (Structured Query Language). Data dalam database bisa

diakses melalui aplikasi non-web (misalnya dengan Visual Basic) maupun aplikasi web misalnya dengan PHP.

2.3.2 Normalisasi

Normalisasi adalah suatu proses yang digunakan untuk menentukan pengelompokan atribut-atribut dalam sebuah relasi sehingga diperoleh relasi yang berstruktur baik. Dalam hal ini yang dimaksud dengan relasi berstruktur baik adalah relasi yang memenuhi dua kondisi berikut :

1. mengandung redundansi sesedikit mungkin, dan
2. memungkinkan baris-baris dalam relasi disisipkan, dimodifikasi, dan dihapus tanpa menimbulkan kesalahan atau ketidakkonsistenan.

Model berfungsi untuk mengelola database seperti MYSQL. Model berhubungan dengan database sehingga biasanya dalam model akan berisi class ataupun fungsi untuk membuat (create), melakukan pembaruan (update), menghapus data (delete), mencari data (search), dan mengambil data (select) pada database. Selain itu juga model akan berhubungan dengan perintah-perintah query sebagai tidak lanjut dari fungsi – fungsi (create, update, delete, select).

2.4 Use-case diagram

Use-case diagram adalah gambaran graphical dari beberapa atau semua *actor*, *use-case*, dan interaksi diantara komponen-komponen tersebut yang memperkenalkan suatu sistem yang akan dibangun. *Use-case diagram* menjelaskan manfaat suatu sistem jika dilihat menurut pandangan orang yang berada di luar sistem. Diagram ini menunjukkan fungsionalitas suatu sistem atau kelas dan bagaimana sistem tersebut berinteraksi dengan dunia luar.


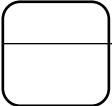



Use-case diagram dapat digunakan selama proses analisis untuk menangkap *requirement* sistem dan untuk memahami bagaimana sistem seharusnya bekerja. Selama tahap desain, *use-case diagram* berperan untuk menetapkan perilaku (*behavior*) sistem saat diimplementasikan. Dalam sebuah model mungkin terdapat satu atau beberapa *use-case diagram*. Kebutuhan atau *requirements system* adalah fungsionalitas apa yang harus disediakan oleh sistem kemudian didokumentasikan pada model *use-case* yang menggambarkan fungsi sistem yang diharapkan (*use-case*), dan yang mengelilinginya (*actor*), serta hubungan antara actor dengan *use-case* (*use-case diagram*) itu sendiri. *Actor* tersebut mempresentasikan seseorang atau sesuatu (seperti perangkat, sistem lain) yang berinteraksi dengan sistem. Sebuah *actor* mungkin hanya memberikan informasi inputan pada sistem, hanya menerima informasi dari sistem atau keduanya menerima, dan memberi informasi pada sistem. Actor hanya berinteraksi dengan *use-case*, tetapi tidak memiliki kontrol atas *use-case*. *Actor* digambarkan dengan *stick man*. *Actor* dapat digambarkan secara umum atau spesifik, dimana untuk membedakannya dapat digunakan *relationship*.

2.5 Data Flow Diagram (DFD)

Desain arsitektur menentukan hubungan di antara elemen-elemen struktural utama dari program. Desain arsitektur sistem ini di sajikan ke dalam diagram yang dikenal dengan DFD (Data Flow Diagram). Data flow diagram adalah sebuah teknik grafis yang menggambarkan aliran informasi yang diaplikasikan pada saat data bergerak dari input menjadi output. DFD juga dikenal sebagai grafik aliran data atau bubble chart. DFD terdiri dari level level, masing-masing level menjelaskan level sebelumnya.

Tingkatan pada Data Flow Diagram di mulai pada DFD tingkat 0 yang merepresentasikan seluruh elemen sistem sebagai sebuah bubble tunggal dengan data input dan output yang ditunjukkan oleh anak panah yang masuk dan keluar secara berurutan. Untuk tingkatan berikutnya adalah DFD tingkat I yang merupakan dekomposisi dari diagram level zero. Pada tingkatan ini proses yang berada pada level 0 di pecah menjadi sub-sub proses. Jika proses pada level 1 masih bisa untuk dipecah ke dalam sub lagi maka dilakukan lagi proses dekomposisi. Proses dekomposisi dilakukan sampai dengan proses siap dituangkan ke dalam program Berikut merupakan notasi dari DFD.




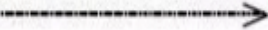

Tabel 2.1 Notasi DFD

NOTASI	NAMA	ARTI
	Entitas	Entitas eksternal, yaitu entitas luar yang berhubungan dengan sistem
	Data Proses	Menyatakan proses-proses yang terdapat di dalam sistem
	Data Store	Merupakan penyimpanan data yang digunakan oleh sistem, namanya terdapat diantara dua garis
 	Konektor	Penghubung, untuk mengalirkan data dari satu bagian ke bagian lain sesuai arah panah

2.6 Class diagram

Class diagram adalah diagram yang digunakan untuk menampilkan beberapa kelas serta paket-paket yang ada dalam sistem/perangkat lunak yang sedang digunakan. *Class diagram* memberikan gambaran (diagram statis) tentang sistem/perangkat lunak dan relas-relasi yang ada didalamnya. Elemen-elemen class diagram dalam pemodelan UML terdiri dari: *Class-class*, *Structure class*, sifat *class (class behavior)*, perkumpulan/gabungan (*association*), pengumpulan/kesatuan (*agregation*), ketergantungan (*dependency*), relasi-relasi turunannya, keberagaman dan indikator navigasi, dan *role name* (peranan/tugas nama).

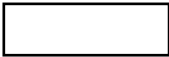
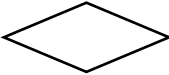
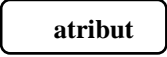
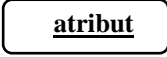
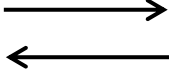
Tabel 2.2 Notasi Class Diagram

NOTASI	NAMA	ARTI
Nama Class <hr/> + atribut + atribut + atribut <hr/> + <i>method</i> + <i>method</i>	<i>Class</i>	<i>Class</i> blok adalah blok pembangun pada pemrograman berorientasi objek. Bagian atas adalah nama dari <i>class</i> . Bagian tengah mendefinisikan <i>property/atribut class</i> . Bagian bawah mendefinisikan <i>method</i> dari sebuah <i>class</i> .
	Asosiasi	relasi antar kelas dengan makna umum, asosiasi biasanya juga disertai dengan <i>multiplicity</i> .
	Asosiasi berarah	Relasi antar kelas dengan makna kelas yang satu digunakan oleh kelas yang lain, asosiasi biasanya juga disertai dengan <i>multiplicity</i> .
	Generalisasi	Relasi antar kelas dengan makna generalisasi – spesialisasi (umum khusus)
	Kebergantungan	Relasi antar kelas dengan makna kebergantungan antar kelas
	Agregasi	Relasi antar kelas dengan makna semua-bagian

2.7 Entity relation Diagram (ERD)

ERD merupakan suatu model untuk menjelaskan hubungan antar data dalam basis data berdasarkan objek-objek dasar data yang mempunyai hubungan antar relasi. ERD untuk memodelkan struktur data dan hubungan antar data, untuk menggambarannya digunakan beberapa notasi dan simbol.

Tabel 2.3 Simbol Notasi ERD

NOTASI	NAMA	ARTI
	Entitas	Entitas eksternal, yaitu entitas luar yang berhubungan dengan sistem
	Relasi	Dalam ER-Diagram notasi ini berarti relasi yang menghubungkan dua atau lebih entitas
1/N, 1/1	Kardinalitas	Menunjukkan kardinalitas relasi antar entitas, 1/n berarti hubungan <i>one to many</i> , 1/1 berarti hubungan <i>one to one</i>
	Atribut	Merupakan atribut dari suatu entitas atau relasi
	Kunci Atribut	Merupakan atribut dari suatu entitas atau relasi yang menjadi primary key.
	Konektor	Penghubung, untuk mengalirkan data dari satu bagian ke bagian lain sesuai arah panah

Menurut salah satu para ahli, Brady dan Loonam (2010), Entity Relationship diagram (ERD) merupakan teknik yang digunakan untuk memodelkan kebutuhan data dari suatu organisasi, biasanya oleh System Analyst dalam tahap analisis persyaratan proyek pengembangan system. Sementara seolah-olah teknik diagram atau alat peraga memberikan dasar untuk desain database relasional yang mendasari sistem informasi yang dikembangkan. ERD bersama-sama dengan detail pendukung merupakan model data yang pada gilirannya digunakan sebagai spesifikasi untuk database.

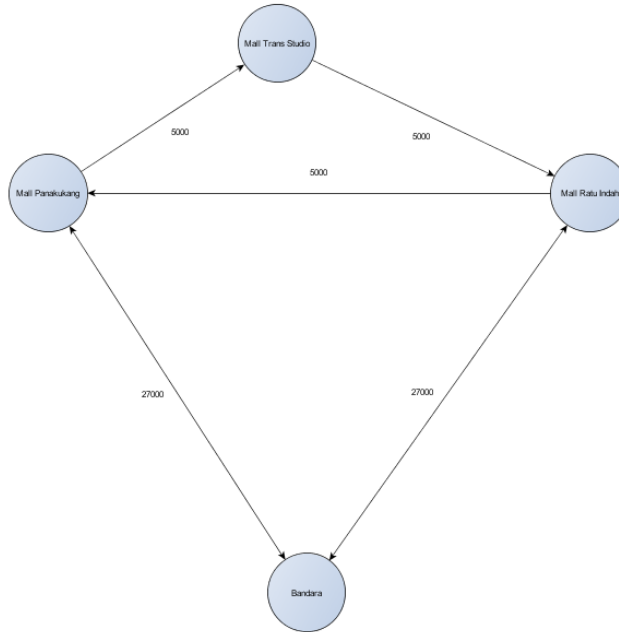
3. METODOLOGI PENELITIAN

3.1 Jenis Penelitian

Penelitian dan pengembangan (research and development-R&D) berasal dari dua kata yaitu penelitian (research) dan pengembangan (development). Frase ini merupakan gabungan 2 (dua) kata kerja yang memiliki tujuan aktivitas. Penelitian (research) merupakan suatu mekanisme atau kegiatan ilmiah dengan mengikuti aturanaturan atau norma-norma penelitian yang sudah standar dan diakui secara universal; sedangkan pengembangan (development) berarti suatu aktivitas yang merujuk pada penambahan, peningkatan, baik dari segi kuantitas maupun kualitas dari suatu kegiatan atau objek yang menjadi kegiatan. Tahapan yang dilalui dalam penelitian, pembangunan konsep, atau penyelesaian kasus, dituliskan pada bagian metodologi.

3.2 Rancangan Struktural

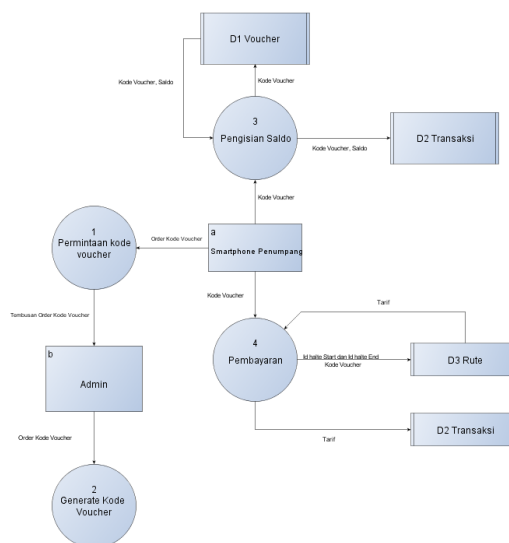
Pada tahap ini, dilakukan penentuan tariff tiap rute antar halte yang bertujuan mengetahui tarif dari tiap-tiap halte serta mengetahui arah jalur bus BRT. Ada 2 tipe tarif yang disediakan oleh bus rapid transit yaitu tarif halte mall dan tarif halte bandara. Untuk setiap halte mall akan dikenakan tarif sebesar RP.5000, sedangkan untuk tarif menuju halte bandara dikenakan biaya RP.27000 karena jarak halte bandara sangat jauh dari halte mall.



Gambar 3.1 Tarif Rute Halte

3.2.1 Data Flow Diagram

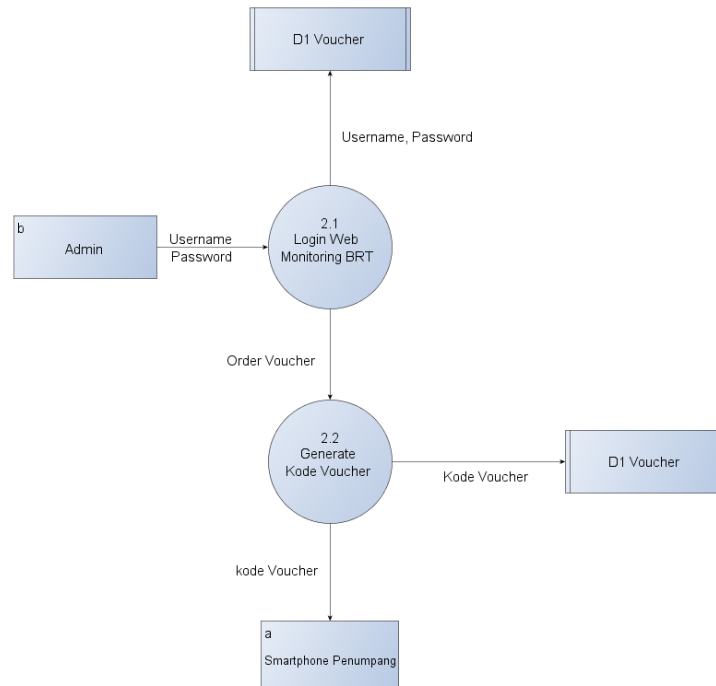
DFD (Data Flow Diagram) adalah diagram yang menggunakan notasi-notasi untuk menggambarkan arus dari data sistem. DFD sering digunakan untuk menggambarkan suatu system yang telah ada atau system baru yang akan dikembangkan secara logika tanpa mempertimbangkan lingkungan fisik dimana data tersebut mengalir (misalnya lewat telpon, surat dan sebagainya) atau lingkungan fisik dimana data tersebut akan disimpan. Pada Level 0 ini terdapat 2 entitas luar yaitu smartphone sebagai media pembayaran serta pengisian saldo. Sedangkan Admin sebagai monitoring dan penyedia kode voucher.



Gambar 3.2 DFD Level 0

Pada Level 1 untuk proses nomor 2 dijelaskan bagaimana proses generate kode voucher. Untuk generate kode voucher, Admin melakukan login pada website BRT. Kemudian admin akan membuka

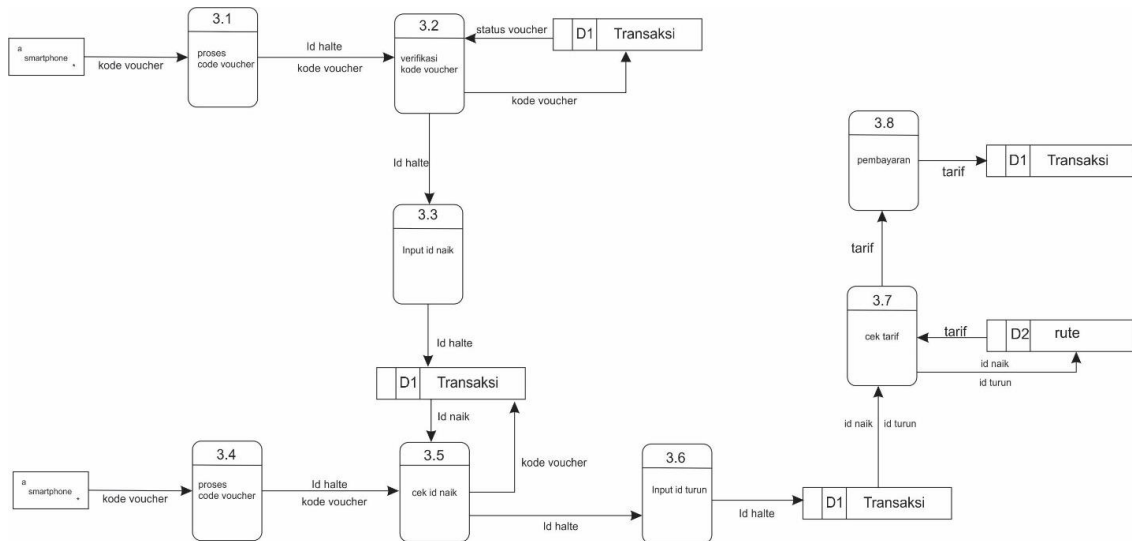
halaman generate kode voucher dan menekan tombol generate. Website akan menampilkan form kode voucher dan admin akan mengisi form tersebut untuk membuat kode voucher. Setelah mengisi form, admin akan menekan



Gambar 3.3 DFD Level 1 Proses 1

Pada Level 1 untuk proses nomor 2 dijelaskan bagaimana proses pembayaran yang terjadi di dalam system. Pertama-tama user harus menscan smarthphone pada scanner halte sebelum masuk. Scanner akan mengambil data smarthphone dan mengirimnya ke system beserta id halte. System akan mengecek terlebih dahulu kode voucher yang dikirim oleh scanner. Jika kode voucher terdaftar pada database code_voucher system akan meneruskan proses pembayaran, jika tidak terdapat pada database code_voucher system akan menghentikan proses pembayaran. Setelah proses pengecekan selesai system akan meneruskan dengan memasukkan id halte pertama pada database code_voucher dan system akan membuka palang pintu masuk. Setelah user turun dari bis user harus melakukan scan pada scanner halte. Scanner akan menerima data berupa kode voucher dan meneruskannya pada system.

Sistem akan menerima data dari scanner berupa kode voucher dan id halte kemudian melakukan pengecekan id naik pada database code_voucher. Jika terdapat id_naik pada code_voucher berdasarkan kode voucher yang ada pada database system akan memasukkan id halte pada field id_turun berdasarkan kode voucher user. System akan melakukan pembayaran dengan mencocokkan tarif halte pada database tarif berdasarkan berdasarkan data yang dibawa oleh system yaitu id_naik dan id_turun. Setelah mendapatkan tarif system akan melakukan pengurangan saldo pada database code_voucher berdasarkan kode voucher yang diterima.

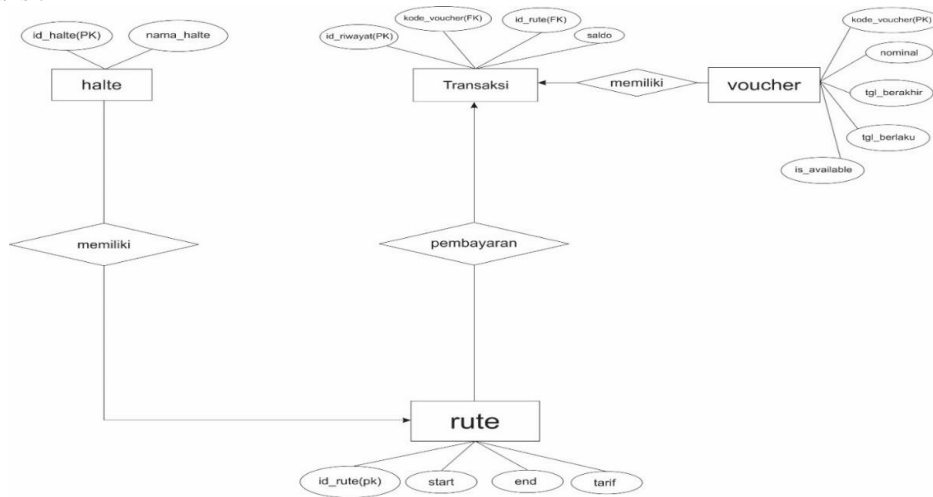


Gambar 3.4 DFD Level 1 Proses 3

3.3 Rancangan Basis Data

3.3.1 Rancangan ERD

Dalam sistem yang dibuat terdapat 4 tabel database yang digunakan yaitu tabel voucher, halte, rute, dan transaksi.



Gambar 3.5 Diagram ERD

Pada gambar di atas, di jelaskan tabel rute memiliki hubungan dengan tabel halte karena pada kolom start dan kolom end berisikan id halte yang ada pada tabel halte. tabel rute memiliki relasi terhadap tabel transaksi dengan nama pembayaran, maksud dari kata “pembayaran” berarti tabel rute melakukan pembayaran terhadap tabel transaksi dengan cara mengirim id rute yang berisi nominal tarif terhadap kolom saldo yang ada pada tabel transaksi. Sedangkan pada twabel voucher memiliki hubungan dengan tabel transaksi karena kode_voucher yang ada pada tabel voucher akan di masukkan juga kedalam tabel transaksi sebagai id penumpang.

3.3.2 Penerapan Normalisasi

Normalisasi adalah melakukan dekomposisi atau penguraian tabel beserta datanya, menjadi tabel yang normal menurut konsep RDBMS (Relational Database Management System). Dekomposisi diawali dengan melakukan analisis pada suatu tabel atau beberapa contoh formulir yang sudah memiliki data lengkap untuk suatu basis data.

1. Bentuk normal pertama (1NF)

Pada tahap normalisasi pertama adalah merubah atribut multivalued menjadi atribut single value, dengan cara menghilangkan repeating group pada tabel dan melengkapi data field yang kosong.

id_riwayat	code_voucher	nominal	id_halte	nama_halte	id_rute	start	end	tarif	saldo
1	19010121558337	10000	1	Mall Panakukang	1	1	1	5000	20000
2	19092825431337	10000	2	Mall Ratu indah	2	2	1	5000	15000
3	45286766013623	20000	3	Trans Studio	3	3	1	5000	10000
4	45447828703723	20000	4	Bandara	4	4	1	27000	10000

Gambar 3.6 Histori Transaksi

2. Bentuk normal kedua (2NF)

Setelah memenuhi Bentuk Normal Pertama (1NF). Pada tahap bentuk normal kedua, yaitu menghilangkan ketergantungan secara parsial, yaitu dengan cara melakukan dekomposisi tabel, setiap tabel memiliki record yang tergantung pada satu field key saja.

id_halte	nama_halte
1	Mall Panakukang
2	Mall Ratu indah
3	Trans Studio
4	Bandara

Gambar 3.7 List Halte

code_voucher	nominal	tgl_voucher	tgl_berakhir	is_available
19010121558337	10000	2016-04-11	2016-05-07	10000
19092825431337	10000	2016-04-11	2016-05-07	10000
45286766013623	20000	2016-01-01	2016-12-31	20000
45447828703723	20000	2016-01-01	2016-12-31	20000

Gambar 3.8 Voucher

id_riwayat	code_voucher	id_rute	start	end	tarif	saldo
1	19010121558337	1	1	1	5000	20000
2	19092825431337	2	2	1	5000	15000
3	45286766013623	3	3	1	5000	10000
4	45447828703723	4	4	1	27000	10000

Gambar 3.9 Transaksi

3. Bentuk normal ketiga (3NF)

Setelah memenuhi Bentuk Normal Kedua (2NF). Pada Gambar 3 Tabel Transaksi terdapat data-data yang masih harus dilakukan dekomposisi. Oleh sebab itu, di lakukan proses normalisasi ketiga (3NF). Dekomposisi dilakukan pada field *id_rute*, *start* dan *end*, yang harus dipisahkan menjadi tabel yang terpisah untuk menghilangkan ketergantungan transitif.

id_rute	start	end	tarif
1	1	1	5000
2	2	1	5000
3	3	1	5000
4	4	1	27000

Gambar 3.10 List Rute

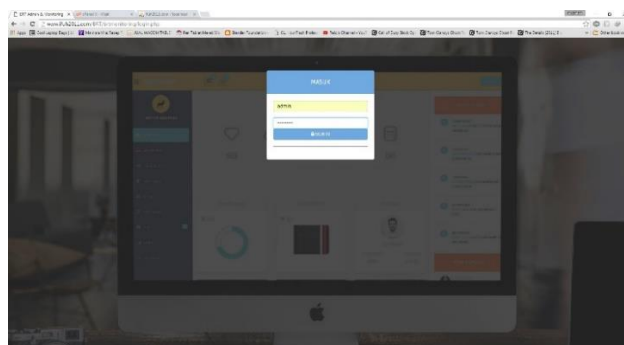
id_riwayat	code_voucher	id_rute	saldo
1	19010121558337	1	20000
2	19092825431337	2	15000
3	45286766013623	3	10000
4	45447828703723	4	10000

Gambar 3.11 List Transaksi

4. HASIL DAN PEMBAHASAN

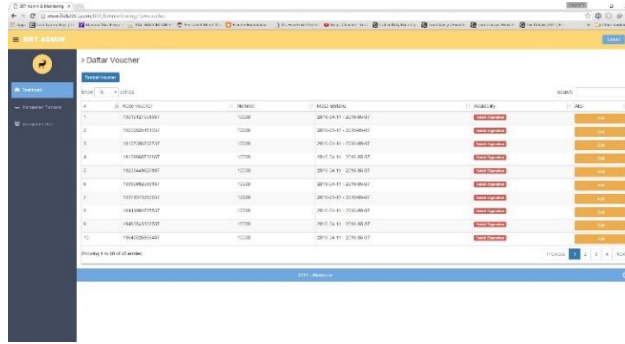
4.1 Hasil

- 4.1.1 Berikut akan dijelaskan proses pembuatan kode voucher serta monitoring transaksi yang akan dilakukan pada web BRT oleh admin.



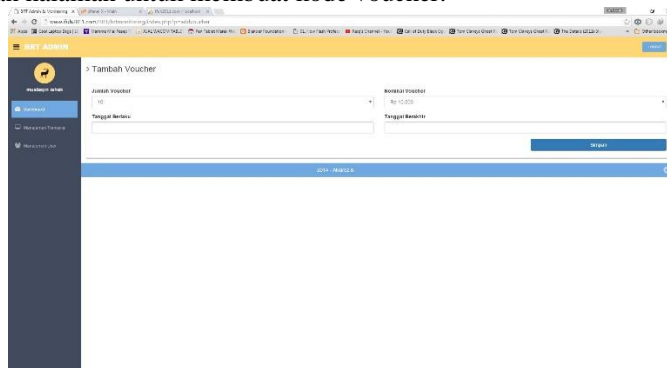
Gambar 4.1 Tampilan awal Web BRT

- 4.1.2 Admin harus melakukan login pada web BRT, jika login berhasil web akan menampilkan daftar kode voucher.



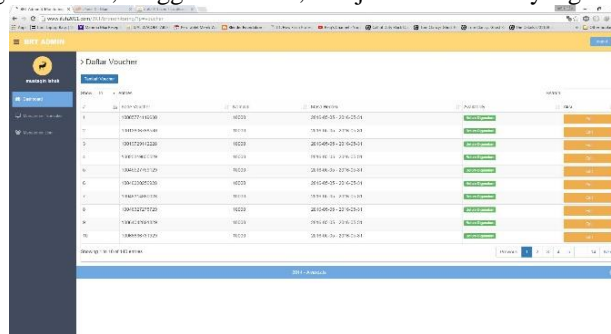
Gambar 4.2 Tampilan Web BRT Setelah berhasil login

4.1.3 Untuk melakukan generate voucher admin harus menekan tombol tambah voucher. Web akan menampilkan halaman untuk membuat kode voucher.



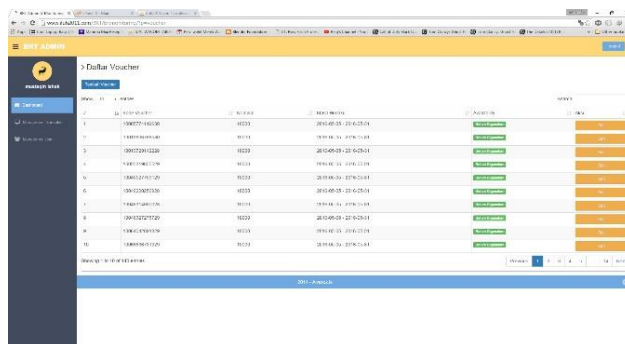
Gambar 4.3 Tampilan halaman tambah voucher

4.1.4 Admin harus terlebih dahulu mengisi form yang ada pada halaman tambah voucher. Nilai voucher, tanggal berlaku, tanggal berakhir, dan jumlah voucher yang akan di buat.



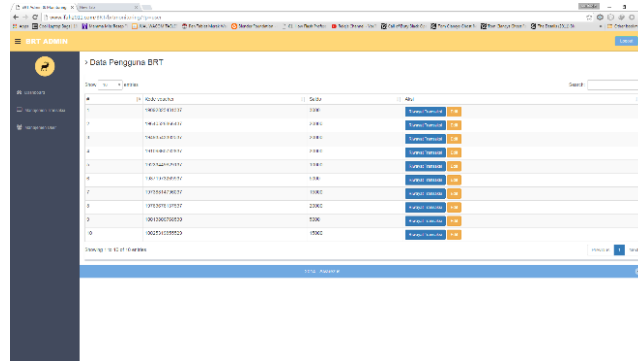
Gambar 4.4 Tampilan ketika form telah di isi

4.1.5 Setelah form di isi admin tinggal menekan tombol simpan. Dan voucher akan tergenerate sesuai permintaan admin.

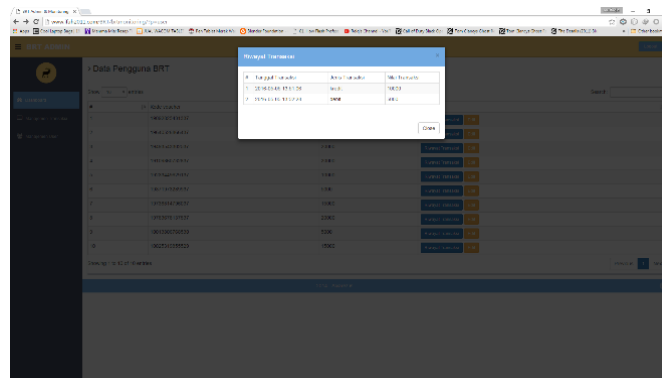


Gambar 4.5 Tampilan hasil tambah voucher

- 4.1.6 Admin telah mendapatkan stock voucher yang akan digunakan oleh penumpang yang ingin mengisi saldo.
- 4.1.7 Pada web BRT, admin dapat memonitoring setiap transaksi baik dalam pengisian saldo maupun transaksi yang pembayaran yang dilakukan oleh penumpang. Berikut tampilan halaman monitoring :



Gambar 4.6 Tampilan monitoring penumpang

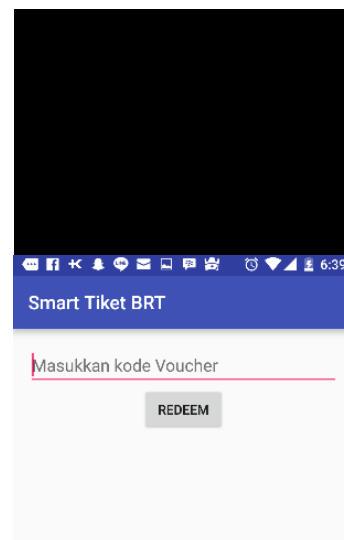


Gambar 4.7 Tampilan detail transaksi penumpang

- 4.1.8 Berikut ini tampilan aplikasi BRT pada smartphone serta dijelaskan proses pengisian saldo dan proses pembayaran.



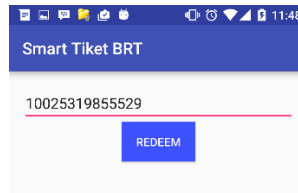
Gambar 4.8 Tampilan hasil tambah voucher



Gambar 4.9 Tampilan Pengisian Saldo

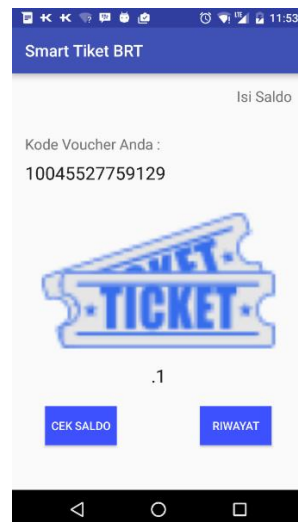
- 4.1.9 Sebelum dapat digunakan, penumpang harus mengaktifkan NFC pada smartphone.
- 4.1.10 Setelah NFC di aktifkan, penumpang harus mengisi saldo pada aplikasi BRT.

- 4.1.11 Penumpang hanya perlu memasukkan kode voucher yang didapatkan dari admin untuk melakukan pengisian saldo.



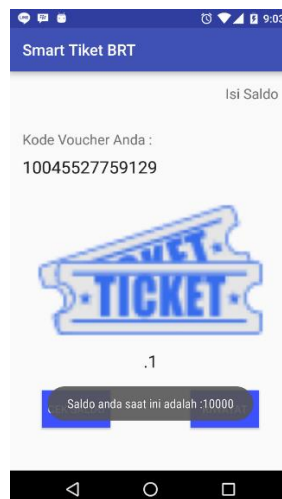
Gambar 4.10 Tampilan pengisian saldo

- 4.1.12 Setelah penumpang memasukkan kode voucher, penumpang tinggal menekan tombol redeem. jika berhasil akan tertampil kode voucher yang sudah dimasukkan di laman depan aplikasi.



Gambar 4.11 Tampilan halaman depan

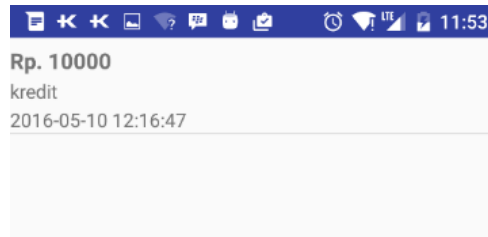
- 4.1.13 Untuk memastikan saldo sudah terisi, penumpang dapat melakukan cek saldo dengan menekan tombol "cek saldo" .



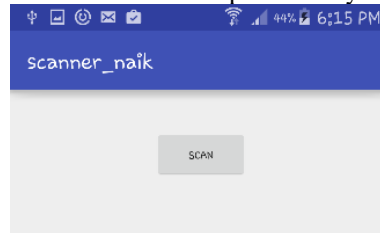
Gambar 4.12 Tampilan cek saldo

- 4.1.14 Setelah pengisian saldo berhasil, penumpang sudah bisa menggunakan aplikasi tersebut. Untuk menggunakan aplikasi tersebut penumpang tinggal menempelkan smartphone pada scanner sebelum naik bus. Penumpang dapat melihat riwayat transaksi yang dapat dilihat

dengan menekan tombol riwayat. Setelah menekan tombol riwayat, maka muncul tampilan riwayat penumpang seperti pada gambar di bawah ini.

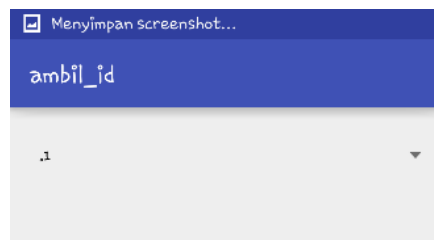


Gambar 4.13 Tampilan Riwayat



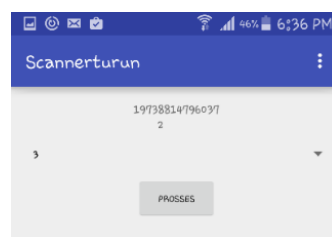
Gambar 4.14 Tampilan Scanner naik

- 4.1.15 Scanner akan menerima data berupa kode voucher yang akan di cek apakah terdapat kode voucher pada database. Jika sukses scanner akan mengirim id halte pada smartphone.



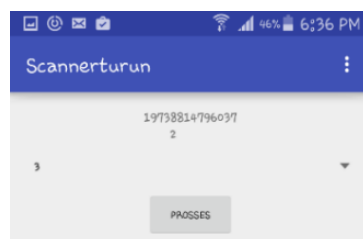
Gambar 4.15 Tampilan Pengimputan Id Halte

- 4.1.16 Pada saat turun dari bus penumpang akan melakukan scan pada pintu keluar untuk melakukan proses pembayaran. Scanner akan mengambil data berupa id halte yang didapatkan pada saat penumpang naik dan kode voucher.



Gambar 4.16 Tampilan Scanner turun

- 4.1.17 Setelah data didapatkan sistem akan mengecek tarif berdasarkan id halte yang berasal dari penumpang dengan id halte yang ada pada scanner pada database. Setelah tarif didapatkan sistem akan melakukan pembayaran pada saldo kode voucher.



Gambar 4.18 Tampilan transaksi berhasil

4.2 Pembahasan

Sistem pembayaran BRT berbasis mobile ini digunakan untuk memudahkan penumpang dalam pembayaran transportasi. Sistem ini akan dipasang pada smartphone penumpang berupa aplikasi sebagai alat pembayaran transportasi yang akan digunakan pada saat hendak masuk halte maupun keluar dari halte. Untuk menggunakannya, penumpang harus melakukan pengisian saldo yang bisa didapatkan pada kode voucher. Kode voucher bisa didapatkan dari admin BRT yang berlokasi di loket halte. Setelah melakukan pengisian saldo, penumpang dapat menggunakan aplikasi tersebut dengan melakukan scan pada saat masuk halte. Jika sistem menerima data yang diterima dari smartphone, pintu masuk akan terbuka dan penumpang bisa masuk ke dalam halte. Begitu pula pada saat keluar halte, penumpang harus melakukan scan sebelum keluar. Sistem akan menerima data dari smartphone dan memproses pembayaran. Setelah sistem selesai memproses, pintu akan terbuka dan penumpang bisa meninggalkan halte.

5. KESIMPULAN DAN SARAN

Berdasarkan hasil penelitian, maka dapat disimpulkan beberapa hal yaitu hasil pengujian fungsional dari sistem pembayaran BRT berbasis mobile menunjukkan bahwa sistem dapat berjalan dengan baik, yaitu sistem dapat menggenerate kode voucher, mengisi saldo, dan melakukan pembayaran. Hasil pengujian pada beberapa smartphone yang memiliki fitur NFC yaitu NEXUS 5, Samsung Galaxy Note 4, dan Samsung Galaxy S6 dapat berjalan dengan baik dan tidak terdapat error saat dilakukan pengujian. Sistem dapat mengenal kode voucher yang valid dengan kode voucher yang tidak valid. Sistem mampu memproses pembayaran berdasarkan kode voucher dan id halte yang didapatkan dari smartphone user. Dalam penggunaan sistem pembayaran BRT berbasis mobile, user harus menggunakan smartphone yang mendukung teknologi NFC untuk melakukan pembayaran serta koneksi internet pada saat melakukan pengisian saldo dan melihat riwayat. Jika kedua hal tersebut tidak terpenuhi maka sistem tidak akan berjalan dengan baik.

UCAPAN TERIMA KASIH

Terima kasih dan penghargaan kami berikan kepada pihak fakultas, universitas, dan SMK Gunung Sari 1 Makassar atas segala dukungan dan motivasinya dalam penyelesaian penelitian ini, sehingga dapat terselesaikan dengan tepat waktu.

DAFTAR PUSTAKA

- [1] Jogyanto H.M. 1989. Analisis & Desain Sistem Informasi: pendekatan terstruktur teori dan praktisi aplikasi bisnis, [e-book], ANDI Yogyakarta, 699-713.
- [2] Priyadi, Yudi. 2013. Kolaborasi SQL & ERD dalam Implementasi Database. Bandung: ANDI Yogyakarta.
- [3] Nazruddin Safaat H.MT.2014. Aplikasi Berbasis Android. INFORMATIKA Bandung.
- [4] Developers, NFC Basics,<http://developer.android.com/guide/topics/connectivity/nfc/nfc.html>. Diakses pada tanggal 14 May 2016.
- [5] Nazaruddin, Safaat. 2012. Pemrograman Aplikasi Mobile, Smartphone, Dasar. Jakarta: Gunadarma.
- [6] Imansyah, Muhammad. 2003. PHP dan MySQL Untuk Orang Awam. Palembang: Maxikom.