

Terbit online pada laman web jurnal : <http://teknosi.fti.unand.ac.id/>

## Jurnal Nasional Teknologi dan Sistem Informasi

| ISSN (Print) 2460-3465 | ISSN (Online) 2476-8812 |



Artikel Penelitian

# Rancang Bangun Aplikasi *Point of Sales Distro Management System* dengan Menggunakan *Framework React Native*

Putu Dika Arta Wiguna<sup>a</sup>, I Putu Agus Swastika<sup>a</sup>, I Putu Satwika<sup>a</sup><sup>a</sup>Jurusan Sistem Informasi, STMIK Primakara, Denpasar 80226, Indonesia

### INFORMASI ARTIKEL

#### Sejarah Artikel:

Diterima Redaksi: 14 November 2018

Revisi Akhir: 07 Januari 2018

Diterbitkan Online: 10 Januari 2018

### KATA KUNCI

Distro,

Point of Sale,

React Native

### KORESPONDENSI

Telepon: +62 878 60829433

E-mail: [putudikaaw@gmail.com](mailto:putudikaaw@gmail.com)

### A B S T R A C T

Perkembangan dunia fashion remaja selalu mengalami peningkatan dengan hadirnya berbagai jenis produk-produk baru yang selalu menarik dan inovatif. Berkembangnya fashion remaja saat ini seiring dengan bertambahnya permintaan dari konsumen. Pada proses bisnis distro ini, ternyata para pelaku bisnis distro ini menjalankan bisnisnya secara manual tanpa memanfaatkan perkembangan teknologi saat ini, sehingga mereka masih kesusahan dalam melakukan pencatatan transaksi, manajemen produk, dll. Berdasarkan permasalahan tersebut, maka perlu untuk membuat aplikasi Point of Sales untuk mempermudah pegawai distro dalam pekerjaannya. Metode penelitian yang digunakan pada penelitian ini adalah classic life cycle dan untuk pengumpulan data yaitu dengan melakukan wawancara, observasi dan dokumentasi pada distro yang dijadikan tempat penelitian. Berdasarkan hasil dari pengumpulan data, maka dibuatlah rancangan sistem, rancangan database dan rancangan antarmuka. Rancangan tersebut kemudian dituangkan kedalam coding untuk membuat aplikasi dengan menggunakan framework react native dan setelah aplikasi selesai dibuat, maka dilakukan pengujian terhadap aplikasi tersebut dengan metode black box testing. Hasil dari pengujian yang telah dilakukan sudah sesuai dengan yang diharapkan sehingga aplikasi Point of Sales ini berhasil dibuat dan sudah disesuaikan dengan kebutuhan pihak *distro*.

## 1. PENDAHULUAN

Seperti yang kita ketahui bersama, perkembangan dunia *fashion* saat ini semakin meningkat baik dari kualitas maupun kuantitasnya. Seriring dengan itu, saat ini banyak *distro-distro* yang hadir dan bersaing untuk menjual produk *fashion* anak muda, mulai dari baju, celana, topi dsbnya.

*Distro* atau *distribution store* yang merupakan sebuah toko yang menjual produk *fashion* anak muda yang sesuai dengan perkembangan tren atau mode *fashion* pada saat itu [1]. Produk *fashion* yang dijual oleh *distro* biasanya adalah produk dari suatu *clothing* (penyedia) atau dari suatu *band* musik dengan menggunakan *brand* yang mereka buat sendiri dan mereka promosikan sendiri, sehingga begitu *brand* tersebut mulai terkenal maka banyak anak muda yang tertarik dan membeli produk dari *clothing* atau *band* tersebut di *distro*. Kerjasama yang dilakukan oleh pihak *distro* dan pihak *clothing* adalah dalam

bentuk konsinyasi, yang dimana pihak *clothing* menitipkan produknya di beberapa *distro* dan baru menerima pembayaran setelah produknya terjual. Di beberapa *distro*, biasanya mengambil keuntungan sebesar 30% dari harga produk yang dititipkan oleh pihak *clothing*. Menurut pemilik *distro* Handmad di Denpasar, dalam proses bisnis *distro* mengalami beberapa kendala, diantaranya kesulitan dalam manajemen dan pengecekan stok produk, kesulitan dalam pencatatan transaksi penjualan, kesulitan dalam manajemen pembayaran *clothing* dan lain-lainnya. Menurut dia, hal tersebut terjadi karena semua proses yang dilakukan masih manual dengan menggunakan buku tulis atau menggunakan aplikasi *Microsoft Excel*.

Berdasarkan permasalahan di atas, maka perlu dibuatkan suatu sistem aplikasi yang dapat terintegrasi antara pihak *distro* sebagai penjual produk dan pihak *clothing* sebagai penyedia produk. Sistem aplikasi atau platform yang dapat digunakan untuk mengintegrasikan semuanya secara *realtime* adalah dengan menggunakan aplikasi atau platform *Distro Management System*

(DIMANS) berbasis web. Dengan menggunakan platform DIMANS, pihak *distro* nantinya akan mendapat kemudahan dalam melakukan manajemen dan pengecekan stok produk, pencatatan transaksi penjualan, manajemen pembayaran ke pihak *clothing* dan lain-lain. Dan platform dimans ini dikembangkan dalam aplikasi *point of sales* berbasis *mobile* yang nantinya khusus digunakan oleh pihak *distro* untuk mempermudah dalam proses penjualan di *distro* dan datanya otomatis terintegrasi dengan web platform dimans.

## 2. TINJAUAN PUSTAKA

### 2.1. Distro

*Distro* atau *Distributor Outlet* adalah tempat untuk mendistribusikan barang. Barang yang didistribusikan biasanya adalah baju, celana, tas, jaket dan lain-lain [2]. Produk yang ada di *distro* biasanya berasal dari komunitas musik *band* independen atau istilahnya *band indie*, komunitas *skateboard* dan merek anak muda masa kini. *Distro* pertama kali dibuka di kota Bandung untuk menjual produk dari *band* luar negeri khususnya *band underground* serta perlengkapan dan *apparel* untuk *skateboard*. Diawali dengan hadirnya *distro Reverse* di Jalan Sukasenang, yang menjual berbagai kaos musik luar dan *apparel skateboard*. *Distro Reverse* pada waktu itu juga dikenal sebagai markas musisi pelopor *indie label*. *Distro* yang menjual produk dari *clothing* lokal pertama kali sendiri adalah Anonim. Saat ini istilah *distro* kemudian dikenal sebagai toko atau *retail* yang khusus hanya menjual produk dari berbagai *clothing* lokal serta *merchandise band indie lokal*.

### 2.2. Point of Sale (POS)

Pengertian *Point of Sale* atau yang biasa yang disingkat POS yaitu merupakan kegiatan yang berorientasi pada penjualan serta sistem yang membantu proses transaksi [3]. *Point of sale* sendiri terdiri dari perangkat keras (*hardware*) dan perangkat lunak (*software*) yang dimana kedua komponen tersebut digunakan untuk setiap proses transaksi. *Software* POS merupakan komponen utama dari sistem POS yang akan menentukan jalannya proses, seperti apa yang harus dilakukan dan bagaimana harus melakukan. Sedangkan *hardware* POS diperlukan untuk menunjang *software* yang sedang berjalan dengan membantu proses pembayaran dan membuat struk belanja untuk pelanggan. Dalam hal pemilihan *hardware* ini, sebaiknya mencocokkan dengan lingkungan kerja, seperti yang akan digunakan oleh penulis pada laporan tugas akhir ini adalah *smartphone* dengan menggabungkan aplikasi *point of sales* dengan aplikasi *barcode scanner* pada *smartphone*, yang merupakan bagian terpenting untuk mempercepat proses pemasukkan barang dan proses pelayanan penjualan *distro*.

### 2.3. Framework Laravel

Laravel adalah sebuah *framework* pengembangan *web* dengan basis *Model View Controller* (MVC) yang ditulis dalam bahasa pemrograman PHP. Laravel telah dirancang untuk meningkatkan kualitas pengembangan *website* dengan mengurangi biaya pengembangan awal dan biaya perawatan berkelanjutan untuk meningkatkan pengalaman dalam pengembangan aplikasi dengan

menyediakan *sintaks ekspresif* yang jelas sehingga pembuatan *web* menjadi lebih efektif dan efisien [4].

Dalam penelitian ini, penulis menggunakan *framework* laravel untuk membuat *application program interface* (API) agar dapat menghubungkan aplikasi POS ini ke *database* MySQL.

### 2.4. Database MySql

MySQL merupakan sebuah *database* developer yang juga bersifat *free*, MySQL banyak digunakan sebagai *database* karena mudah digunakan dan juga sangat banyak tersedia [5]. MySQL sendiri menggunakan bahasa SQL yang saat ini sudah banyak digunakan. MySQL merupakan *software database* yang termasuk paling populer di lingkungan Linux atau Unix, kepopuleran ini ditunjang karena *query* dari basis data yang saat itu bisa dikatakan paling cepat dan juga memiliki sedikit permasalahan.

Beberapa keunggulan MySQL dibandingkan database lain [5] :

- Kemudahan dalam penggunaan MySQL adalah *simple database system* dengan performa tinggi dan tidak kompleks untuk proses instalasi dan administrasinya dibanding dengan sistem yang lebih besar.
- Mendukung bahasa *query* MySQL dapat menggunakan SQL, juga dapat diakses dengan menggunakan aplikasi ODBC.
- Kemampuan Banyak *client* dapat berhubungan dengan server pada saat bersamaan. *Clients* dapat menggunakan *multiple database* secara bersamaan.

### 2.5. React Native

React Native adalah *framework open source* besutan facebook yang dibuat setelah facebook sebelumnya membuat react.js, React.JS sendiri merupakan sebuah *library* dari facebook yang dapat digunakan untuk membangun antarmuka pengguna (UI). Jadi, react native adalah *framework open source* untuk membuat aplikasi *multi-plattform* (android, ios dan windows platform "dalam tahap pengembangan") dengan bahasa javascript, sesuai dengan deskripsi di situs resminya "Learn once, write anywhere" [6].

React Native adalah salah satu *framework* untuk membuat aplikasi *mobile* dengan menggunakan kode JavaScript. *Framework* React Native memiliki seperangkat komponen bagi platform iOS dan Android untuk membangun aplikasi *mobile* dengan tampilan yang benar-benar seperti native. Dengan menggunakan *Framework* React Native, kita dapat *render User Interface* untuk platform iOS dan Android. React Native ini adalah *framework open source*, yang bisa kompatibel dengan platform lain seperti Windows atau tvOS dalam waktu dekat [7].

Dengan *Framework* React Native, kita tidak membangun "aplikasi *mobile web*", "aplikasi HTML5", atau "aplikasi *hybrid*". Kita membangun aplikasi *mobile native* yang tidak dapat dibedakan dari aplikasi yang dibuat menggunakan Objective-C atau Java. React Native menggunakan blok bangunan *User Interface* dasar yang sama seperti aplikasi iOS dan Android biasa. Sehingga kita tinggal memasang blok-blok tersebut menggunakan JavaScript.

### 3. METODOLOGI

Adapun metodologi penelitian yang digunakan pada penelitian ini adalah *classic life cycle*. *Classic life cycle* adalah suatu paradigma perangkat lunak yang menuntut suatu sistem yang sistematis, mulai dari suatu level sistem kemudian terus maju ke level berikutnya. Adapun tahapan *Classic life cycle* adalah sebagai berikut: [8].

#### 1. *Analysis* (Analisis)

Analisis dilakukan terhadap data yang ada serta mengumpulkan kebutuhan-kebutuhan perangkat lunak yang akan dibangun. Merupakan tahap dimana inialisasi pendefinisian masalah untuk penyelesaian teknis pengembangan perangkat lunak mulai dilakukan. Dalam penelitian ini, data yang dikumpulkan oleh penulis yaitu dalam bentuk data kualitatif yang diperoleh dari hasil wawancara dengan pemilik atau penjaga *distro* dan hasil observasi pada *distro* yang menjadi tempat penelitian. Kemudian data tersebut penulis gunakan sebagai acuan dalam pembuatan alur sistem, rancangan antarmuka dan rancangan *database* aplikasi *point of sales* (POS). Selain itu, data tersebut juga penulis gunakan sebagai pertimbangan untuk *fitur* apa saja nantinya yang akan disediakan pada aplikasi POS yang disesuaikan dengan kebutuhan pada *distro*.

#### 2. *Design* (Desain)

Pada tahap desain dilakukan perubahan kebutuhan-kebutuhan menjadi bentuk karakteristik yang dimengerti perangkat lunak sebelum dimulai penulisan program. Adapun proses yang dilakukan pada tahap ini adalah mendekomposisi modul sistem yang akan dikembangkan, penetapan rancangan masukan dan keluaran yang diperlukan, penetapan struktur data yang dipilih, penetapan prosedur kerja internal, dalam penetapan formula pengolahan data dan representasi interface. Pada penelitian ini, penulis membuat rancangan antarmuka aplikasi POS dengan menggunakan *tools* Adobe Photoshop CS6 dan berpedoman pada aspek *usability* untuk perancangan antarmukanya. Selain itu, Penulis juga membuat rancangan sistem berupa *usecase diagram* dan rancangan *database* menggunakan ERD.

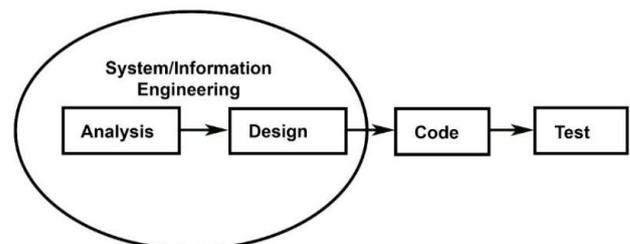
#### 3. *Coding* (Penulisan Program)

Penulisan program dilakukan setelah tahap desain selesai, yaitu dengan mengubah desain menjadi bentuk program yang dimengerti komputer yang dilakukan secara mekanis. Pada tahap ini dilakukan konversi dari hasil rancangan (spesifikasi program) menjadi *source code*. Pada penelitian ini, penulis menggunakan *framework* React Native untuk membuat aplikasi *mobile* POS karena *framework* React Native ini merupakan *framework semi native*, sehingga lebih mudah untuk membuat aplikasi yang nantinya dapat digunakan pada multi-platform seperti Android dan iOS. Untuk membuat aplikasi *mobile* pada *Framework* React Native ini menggunakan bahasa pemrograman Javascript dan bahasa pemrograman PHP untuk

membuat *application programming interface* (API) sebagai penghubung aplikasi *mobile* dengan *database*.

#### 4. *Testing* (Pengujian)

Setelah program dapat berjalan, selanjutnya dilakukan pengujian dengan memfokuskan pada logika internal dari perangkat lunak, fungsi eksternal dan mencari segala kemungkinan masalah. Selanjutnya memeriksa apakah perangkat lunak sudah sesuai dengan yang diharapkan atau belum. Pengujian merupakan proses untuk eksekusi program yang telah selesai dibuat untuk memeriksa apakah terdapat kesalahan atau tidak. Dalam penelitian ini, nantinya penulis akan melakukan testing dari program aplikasi *mobile* POS dengan menggunakan metode *testing "Black Box"* dengan cara langsung menggunakan aplikasi dengan mencoba memasukan inputan yang ada, apakah sudah sesuai dengan tujuan atau tidak. Adapun modul- modul yang akan di testing dalam aplikasi POS, yaitu seperti Modul *Login*, Transaksi Penjualan, Penambahan Stok, dll. Yang menjadi tester dalam melakukan testing menggunakan metode *Black Box* ini nantinya adalah penjaga/pemilik *distro* yang penulis pilih menjadi tempat penelitian.



Gambar 1. Model *Sekuensial Linier / Waterfall Model*

#### 3.1. *Instrumen Penelitian*

Sugiyono menyatakan bahwa “instrumen penelitian digunakan untuk mengukur nilai variabel yang diteliti”. Dengan demikian jumlah instrumen yang akan digunakan pada penelitian ini tergantung pada jumlah variabel yang diteliti. Adapun variabel yang akan diteliti menggunakan instrumen penelitian yaitu : bisnis proses yang ada pada *distro*. Dari variabel tersebut, maka terdapat tiga instrumen yang akan digunakan untuk mengumpulkan data yaitu wawancara, observasi/survey dan dokumentasi. Masing- masing instrumen akan diuraikan sebagai berikut: [9]

1. Dalam penelitian ini pengumpulan data dilakukan melalui wawancara. Wawancara dilakukan terhadap responden dalam hal ini adalah penjaga *distro* atau pemilik *distro* dengan metode *interview* dimana penulis bebas mengajukan pertanyaan terkait dengan data yang ingin dikumpulkan dan responden juga dapat memberikan jawaban dengan lebih jelas dan detail [10]. Wawancara tersebut dilakukan dengan tujuan untuk mendapatkan informasi terkait dengan pola penjualan dan manajemen produk sebagai acuan dalam pengembangan aplikasi *point of sales* pada *distro*. Wawancara yang dilakukan adalah wawancara tidak terstruktur yang dimana rincian topik pertanyaan disesuaikan dengan wawancara di lapangan.

Wawancara ini akan dilakukan pada 5 *Distro* berbeda yang ada di Kota Denpasar dengan jumlah responden 1 orang setiap *distro*.

2. Observasi/*survey* adalah pengambilan data yang dilakukan melalui pengamatan langsung terhadap suatu gejala atau pengumpulan informasi, baik dari populasi besar maupun kecil [10]. Dalam melakukan observasi dapat dilakukan dengan pengamatan secara langsung, observasi dilakukan dengan tes, kuisisioner, ragam gambar dan rekaman suara. Dari beberapa pilihan tersebut, penulis memilih untuk melakukan pengamatan secara langsung ke *distro* yang untuk melihat seperti apa kegiatan mereka dalam melakukan penjualan produk, manajemen stok produk, hingga pembayaran ke *clothing*. Observasi ini akan dilakukan selama 1 hari penuh pada 5 *distro* berbeda yang ada di Kota Denpasar.
3. Dokumentasi  
Dokumentasi adalah kegiatan khusus berupa pengumpulan, pengolahan, penyimpanan, penemuan kembali dan menelaah dokumen yang terdapat pada perusahaan [11]. Dalam penelitian ini, penulis mengumpulkan data-data penjualan di *distro*, data penerimaan barang dari *clothing*, laporan penjualan hingga data pembayaran *distro* ke *clothing*, yang nantinya data tersebut penulis gunakan sebagai acuan dalam perancangan aplikasi *point of sales*.

### 3.2. Jenis Data

Adapun data yang digunakan dalam penelitian ini adalah data kualitatif yaitu data yang disajikan dalam bentuk kata verbal bukan dalam bentuk angka. Yang termasuk data kualitatif dalam penelitian ini yaitu gambaran umum obyek penelitian, meliputi: hasil wawancara dengan penjaga *distro* atau pemilik *distro*, hasil observasi langsung pada *distro* dan hasil dokumentasi pada *distro*.

## 4. HASIL DAN PEMBAHASAN

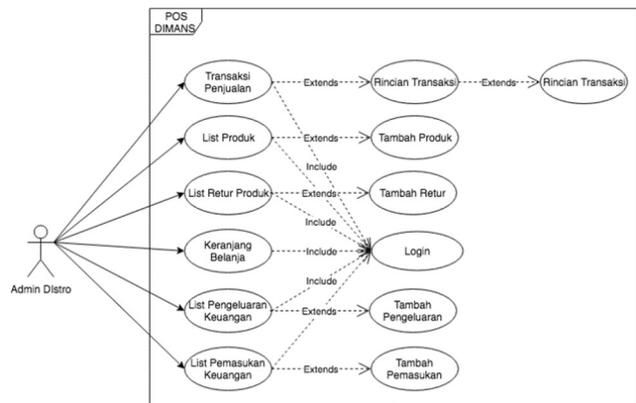
### 4.1. Hasil Penelitian

Dari penelitian yang sudah dilakukan, penulis mengumpulkan data dari melakukan wawancara dengan 5 *distro* yang ada di Kota Denpasar. Wawancara penulis lakukan dengan pemilik *distro* dan pegawai *distro*. Hasil dari wawancara yang penulis lakukan, digunakan untuk merancang bangun aplikasi *point of sales distro management system*, seperti bagaimana desain antarmuka aplikasi, *fitur* aplikasi, bagaimana alur sistem dan apa saja yang perlu ditampilkan pada aplikasi. Penulis juga melakukan observasi langsung ke *distro* untuk mengetahui seperti apa proses transaksi yang terjadi pada *distro* dan juga penulis melakukan studi pustaka untuk mendukung hasil dari wawancara. Kemudian, penulis juga mengumpulkan data seperti data pencatatan produk yang dimiliki oleh pihak *distro* untuk disesuaikan nantinya pada aplikasi yang dibuat. Selanjutnya dari data yang telah penulis kumpulkan, penulis gunakan untuk membuat atau merubah tampilan dan fungsi aplikasi *point of sales* sesuai dengan data yang sudah didapatkan. Adapun kesimpulan dari hasil wawancara dan observasi yang penulis lakukan adalah sebagai berikut :

1. Kasir *distro* dapat dengan mudah untuk melakukan transaksi, baik dengan metode *scan barcode* atau metode pilih produk secara manual.
2. Kasir *distro* dapat mencetak struk di akhir proses transaksi
3. Kasir *distro* dapat melihat riwayat transaksi yang sudah terjadi dan juga bisa mencetak *struk* kembali
4. Kasir *distro* dapat menyimpan atau menanggihkan transaksi sementara ke keranjang belanja
5. Petugas *distro* dapat menambahkan produk dengan mudah
6. Petugas *distro* dapat mencatatat pengeluaran keuangan dan pemasukan keuangan lainnya pada *distro*

#### 4.1.1. Hasil Rancangan Sistem

Rancang bangun aplikasi *point of sales* ini dirancang dengan menggunakan bahasa pemodelan *Unified Modelling Language* (UML). Menurut Nugroho, “UML (*Unified Modeling Language*) adalah ‘bahasa’ pemodelan untuk sistem atau perangkat lunak yang berparadigma ‘berorientasi objek’”. Pemodelan (*modeling*) sesungguhnya digunakan untuk penyederhanaan permasalahan-permasalahan yang kompleks sedemikian rupa sehingga lebih mudah dipelajari dan dipahami” [12]. Rancangan sistem *usecase* diagram Aplikasi *Point of Sales Distro Management System* (DIMANS) dapat dilihat pada gambar 4.1 berikut.

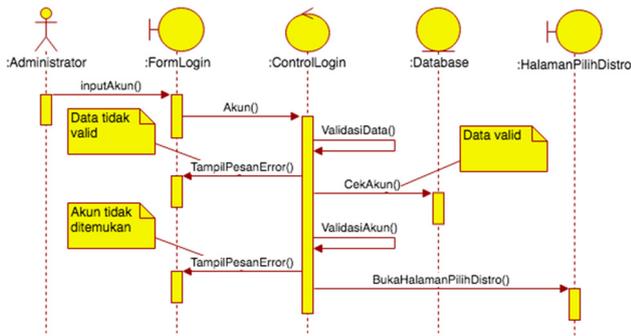


Gambar 2. Usecase Diagram Aplikasi *Point of Sales Distro Management System* (DIMANS)

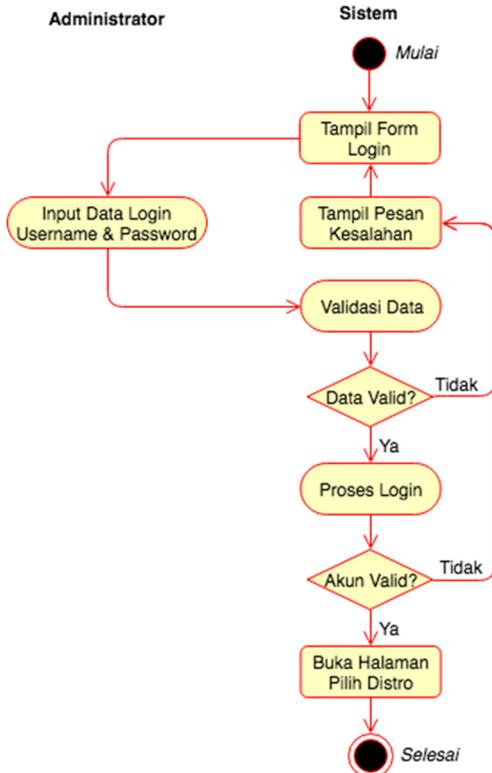
Berdasarkan diagram *usecase* pada gambar 2 diatas, dapat dilihat bahwa hanya terdapat satu pengguna aplikasi yaitu pihak *Distro*. Pihak *Distro* dapat melakukan aktifitas manajemen produk yang meliputi tambah produk dan retur produk, transaksi yang meliputi POS dan *list* transaksi, dan keuangan yang meliputi pengeluaran dan pemasukan lainnya, yang dimana seluruh proses tersebut dapat dilakukan oleh pihak *distro* jika sudah melakukan proses *login* terlebih dahulu.

#### A. Proses Login

Proses *login* merupakan sebuah proses untuk memverifikasi pengguna ketika menggunakan sebuah sistem. Dalam hal ini, pengguna perlu melakukan *login* terlebih dahulu dengan memasukan email dan *password* untuk dapat menggunakan aplikasi POS secara penuh. Adapun *Sequence Diagram* dan *Activity Diagram* proses *login* yaitu sebagai berikut:



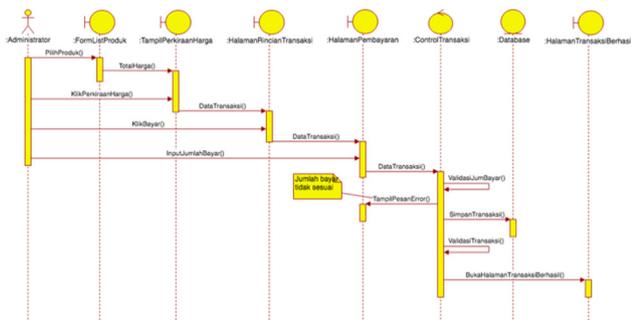
Gambar 3. Sequence Diagram Proses Login



Gambar 4. Activity Diagram Proses Login

B. Proses Transaksi

Proses transaksi merupakan sebuah proses bagi pengguna yang telah login ke sistem untuk melakukan transaksi penjualan pada aplikasi. Dalam proses transaksi ini, pengguna perlu memilih produk terlebih dahulu untuk dapat melakukan transaksi penjualan. Adapun Sequence Diagram dan Activity Diagram proses transaksi yaitu sebagai berikut:



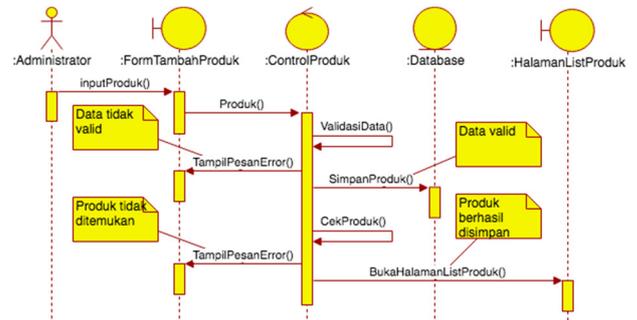
Gambar 5. Sequence Diagram Proses Transaksi



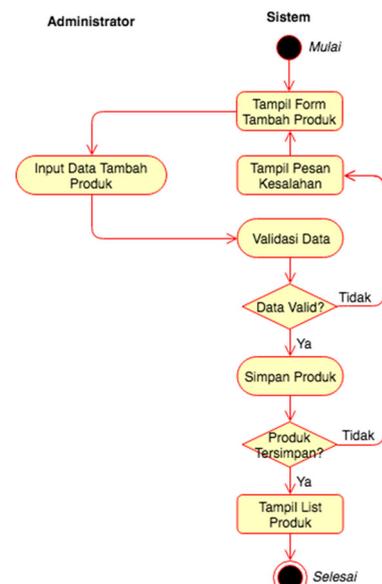
Gambar 6. Activity Diagram Proses Transaksi

C. Proses Tambah Produk

Proses tambah produk merupakan sebuah proses bagi pengguna yang telah login ke sistem untuk menambahkan produk baru. Adapun informasi yang dimasukkan ketika melakukan tambah produk seperti kode produk, jumlah produk dan biaya konsinyasi. Berikut merupakan Sequence Diagram dan Activity Diagram proses tambah produk:

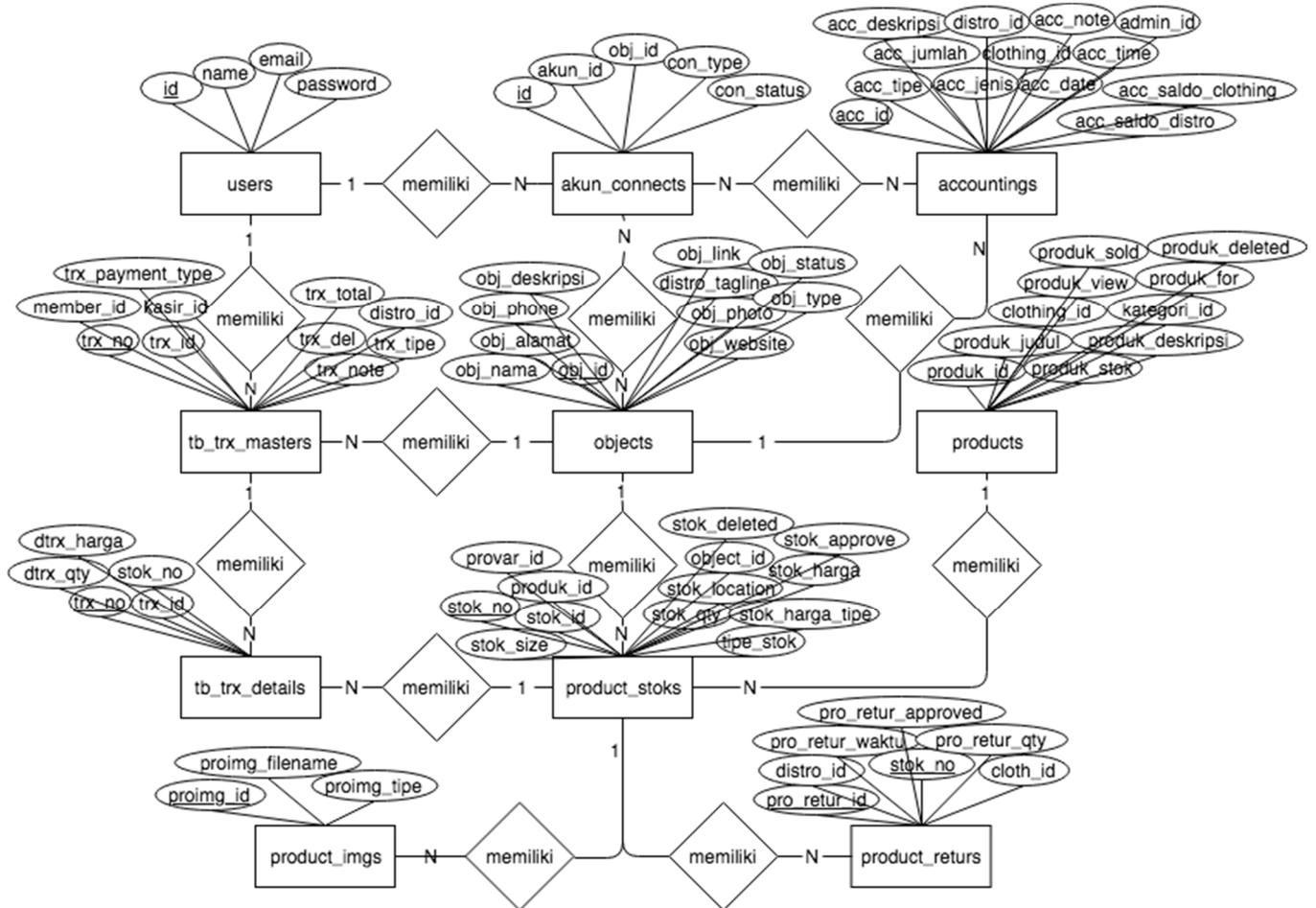


Gambar 7. Sequence Diagram Tambah Produk



Gambar 8. Activity Diagram Tambah Produk

4.1.2. Hasil Rancangan Basis Data



Gambar 9. Rancangan ERD Aplikasi Point of Sales Distro Management System

Perancangan *basis data* diperlukan untuk mempermudah ketika proses implementasi sistem. Ada banyak cara dalam menganalisis dan memodelkan suatu *basis data*, beberapa diantaranya adalah dengan menggunakan *Entity Relationship Diagram* (ERD). *Entity Relationship Diagram* (ERD) adalah sekumpulan cara atau peralatan untuk mendeskripsikan data-data atau objek-objek yang dibuat berdasarkan dan berasal dari dunia nyata yang disebut entitas (*entity*) serta hubungan (*relationship*) antar entitas-entitas tersebut dengan menggunakan beberapa notasi [13]. Adapun rancangan ERD aplikasi POS DIMANS dapat dilihat seperti gambar berikut.



Gambar 10. Rancangan Antarmuka Transaksi

4.1.3. Hasil Rancangan Antarmuka

A. Antarmuka Transaksi

Tampilan antarmuka aplikasi POS yang menjadi halaman utama ketika aplikasi ini dibuka. Pada halaman ini, ditampilkan daftar produk yang dimiliki *distro* dan pengguna langsung dapat melakukan transaksi penjualan pada halaman ini dengan cara menekan tombol (+) pada *list* produk untuk memilih produk. Tampilan rancangan antarmuka dapat dilihat pada gambar 10.

B. Antarmuka Pembayaran

Tampilan antarmuka "Pembayaran" akan tampil ketika setelah menekan tombol "lanjutkan" pada halaman "Rincian Transaksi" sebelumnya. Pada halaman ini, pengguna dapat memilih metode pembayaran yang akan digunakan dan memasukan jumlah pembayaran yang sesuai. Tampilan rancangan antarmuka dapat dilihat pada gambar 11.



Gambar 11. Rancangan Antarmuka Pembayaran

C. Antarmuka Pembayaran Sukses

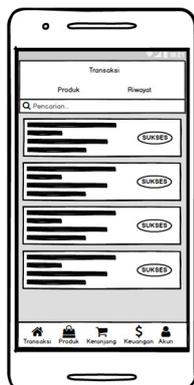
Rancangan antarmuka selanjutnya setelah memasukan jumlah pembayaran dan menekan tombol “Bayar” pada halaman sebelumnya, maka akan tampil halaman “Pembayaran Sukses” dan juga ditampilkan jumlah kembalian dari jumlah pembayaran yang dimasukan. Tampilan rancangan antarmuka dapat dilihat pada gambar 12.



Gambar 12. Rancangan Antarmuka Pembayaran Sukses

D. Antarmuka Riwayat Transaksi

Pada halaman ini, pengguna dapat melihat riwayat transaksi yang sudah pernah dilakukan sebelumnya. Tampilan rancangan antarmuka dapat dilihat pada gambar 13.



Gambar 13. Rancangan Antarmuka Riwayat Transaksi

E. Antarmuka Tambah Produk

Tampilan antarmuka “Tambah Produk” akan tampil ketika pengguna menekan tombol (+) untuk melakukan tambah produk. Pada halaman ini, pengguna bisa menginputkan kode produk,

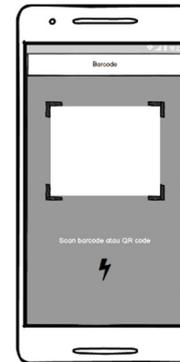
jumlah produk dan biaya konsinyasi untuk menambahkan produk. Tampilan rancangan antarmuka seperti pada gambar 14.



Gambar 14. Rancangan Antarmuka Tambah Produk

F. Antarmuka Transaksi Barcode

Tampilan antarmuka “Tambah Produk” akan tampil ketika pengguna menekan tombol (+) untuk melakukan tambah produk. Pada halaman ini, pengguna bisa menginputkan kode produk, jumlah produk dan biaya konsinyasi untuk menambahkan produk. Tampilan rancangan antarmuka dapat dilihat pada gambar 15.



Gambar 15. Rancangan Antarmuka Transaksi Barcode

4.2. Pembahasan

4.2.1. Aplikasi Point of Sales Dimans

Dari hasil penelitian dan rancangan yang telah penulis lakukan, maka dibuatlah sebuah aplikasi *Point of Sales* atau aplikasi penjualan yang dapat digunakan oleh pihak *distro* untuk mempermudah dalam memproses transaksi penjualan yang terjadi pada *distro*. Aplikasi POS ini penulis buat dengan menggunakan *framework* React Native yang saat penelitian ini dilakukan mulai banyak digunakan oleh para *apps developer*. Berikut merupakan pembahasan dari tampilan program aplikasi POS Dimans yang sudah dibuat :

A. Tampilan Transaksi

Tampilan transaksi menjadi tampilan pertama yang tampil ketika aplikasi pertama kali dibuka dan jika pengguna sudah melakukan *login* sebelumnya. Pada tampilan transaksi ini, penulis membuat desain antarmuka yang sederhana, sehingga dapat memudahkan pengguna dalam menggunakan aplikasi ini nantinya. Pada tampilan ini, penulis menampilkan *list* produk yang dimiliki oleh *distro* yang berisi nama produk, ukuran, kode, warna, harga dan gambar. Pegawai *distro* dapat langsung memilih produk-produk yang akan di transaksikan. Selain itu, juga terdapat *fitur* pencarian

produk dan *icon scan* untuk membuka halaman *barcode scan*. Tampilan antarmuka dapat dilihat pada gambar 16.

**B. Tampilan Pembayaran**

Pada tampilan pembayaran ini, pengguna diinfokan mengenai total harga dari transaksi, kemudian juga diberikan pilihan mengenai metode pembayaran dengan uang tunai atau kartu. Setelah itu, pengguna atau penjaga *distro* dapat langsung menekan tombol “Uang Pas” jika uang yang digunakan pembayaran oleh konsumen jumlahnya sesuai dengan total harga. Sedangkan jika uang yang dibayarkan lebih besar dari total harga, maka pengguna dapat memasukan jumlah pembayaran sesuai dengan yang dibayarkan oleh konsumen, kemudian tekan tombol “Bayar” untuk memproses transaksi. Tampilan antarmuka dapat dilihat pada gambar 17.

**C. Tampilan Transaksi Berhasil**

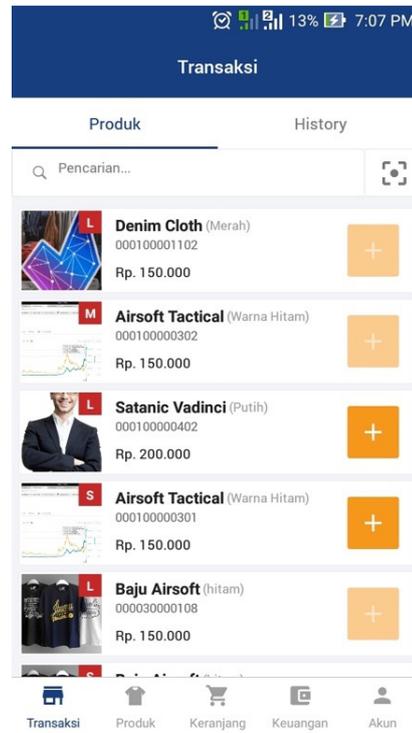
Ketika tampilan transaksi berhasil ini muncul, itu berarti transaksi yang dilakukan sebelumnya telah berhasil diproses dan disimpan. Pada halaman transaksi berhasil ini, ditampilkan jumlah uang kembalian jika konsumen memberikan uang pembayaran lebih dari total harga, jika uang yang diberikan sesuai, maka uang kembalian tampil angka 0 (nol) saja. Selain itu, pada tampilan ini juga terdapat dua tombol yaitu tombol “Transaksi Baru” untuk pergi ke halaman transaksi awal, dan tombol “Cetak Struk” untuk mencetak struk belanja yang dapat diberikan kepada konsumen sebagai bukti belanja nantinya. Tampilan antarmuka dapat dilihat pada gambar 18.

**D. Tampilan Riwayat Transaksi**

Pada halaman ini, pengguna dapat melihat riwayat transaksi yang sudah pernah dilakukan sebelumnya, yang berisi informasi nomor transaksi, metode pembayaran, total harga dan tanggal transaksi. Selain itu, juga ada *fitur* pencarian riwayat dan sinkron data riwayat dengan website jika pada saat transaksi dilakukan, pengguna dalam mode tanpa jaringan internet. Tampilan antarmuka dapat dilihat pada gambar 19.

**E. Tampilan Transaksi Barcode**

Tampilan transaksi *barcode* ini merupakan bagian dari menu transaksi awal. Pada halaman *barcode* ini pengguna dapat melakukan *scanning* pada kode *barcode* yang ada pada produk untuk mempermudah dan mempercepat dalam melakukan transaksi. Setelah scan berhasil, maka otomatis tampil *rincian* produk di bawah yang menginformasikan jumlah barang yang discan dan total harganya. Selain itu, juga ada *fitur torch* atau senter yang ada disebelah kanan atas yang berfungsi menghidupkan *flash* pada *smartphone* jika pada saat melakukan *scan barcode* dalam keadaan kekurangan cahaya. Tampilan antarmuka dapat dilihat pada gambar 20.



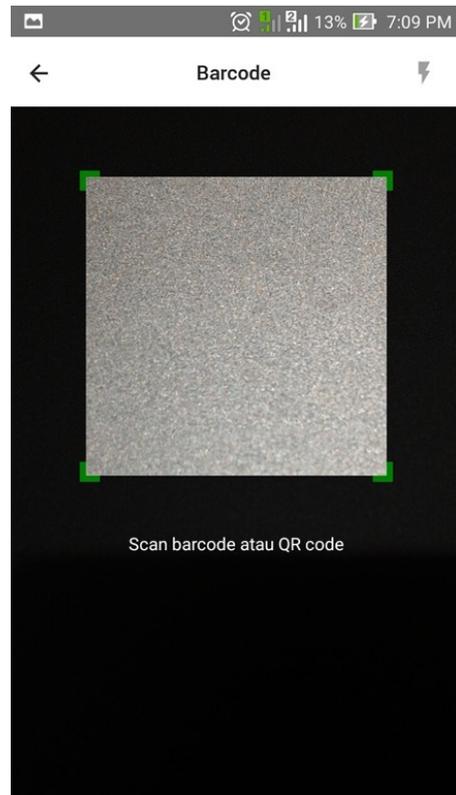
Gambar 16. Tampilan Halaman Transaksi



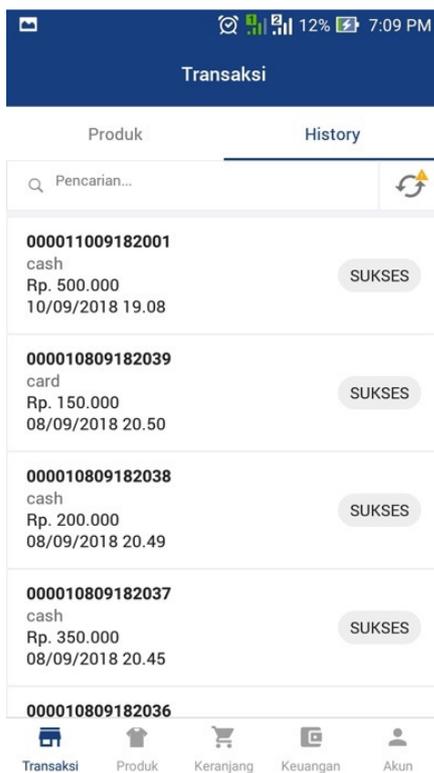
Gambar 17. Tampilan Halaman Pembayaran



Gambar 18. Tampilan Halaman Transaksi Berhasil



Gambar 20. Tampilan Halaman Barcode



Gambar 19. Tampilan Halaman Riwayat Transaksi



Gambar 21. Struk Belanja

#### 4.2.2. Pengujian Aplikasi dengan Metode Black Box

Berikut merupakan hasil pengujian aplikasi *point of sales distro management system* dengan menggunakan metode *black box testing* yang langsung diujikan ke pengguna yaitu pihak *distro* dan juga diuji oleh penulis sendiri. Adapun hasil ringkasan pengujian aplikasi *point of sales* yaitu dapat dilihat seperti pada tabel 1 sampai tabel 3.

Tabel 1. Pengujian Black Box pada Menu Sign In

Skenario Hasil Pengujian (Data Benar)	
Data Masukan	Memasukan email dan password yang sudah terdaftar Email : distro@dimans.co.id Password : 123456
Hasil yang diharapkan	Aplikasi akan menerima akses login
Pengamatan	Aplikasi menerima akses login dan diarahkan ke tampilan distro
Hasil	Sesuai
Skenario Hasil Pengujian (Data Kosong)	
Data Masukan	Password dikosongkan
Hasil yang diharapkan	Aplikasi akan menolak akses login
Pengamatan	Aplikasi menolak akses login dan menampilkan pesan "Peringatan: Password harus diisi"
Hasil	Sesuai
Skenario Hasil Pengujian (Data Salah)	
Data Masukan	Menginputkan email dan password yang tidak terdaftar Email : distro@123.com Password : 111111
Hasil yang diharapkan	Aplikasi akan menolak akses login
Pengamatan	Aplikasi menolak akses login dan menampilkan pesan "Peringatan: Email atau Password salah"
Hasil	Sesuai

Tabel 2. Pengujian Black Box pada Proses Transaksi

Skenario Hasil Pengujian (Data Benar)	
Data Masukan	Total belanja adalah Rp. 220.000. Jumlah Pembayaran : 250.000
Hasil yang diharapkan	Jumlah pembayaran >= total belanja, maka transaksi berhasil.
Pengamatan	Aplikasi menampilkan halaman transaksi berhasil
Hasil	Sesuai
Skenario Hasil Pengujian (Data Salah)	
Data Masukan	Total belanja adalah Rp. 220.000. Jumlah Pembayaran : 200.000
Hasil yang diharapkan	Jumlah pembayaran < total belanja, maka transaksi tidak dilanjutkan.
Pengamatan	Transaksi ditunda dan aplikasi menampilkan pesan "Jumlah pembayaran tidak sesuai"
Hasil	Sesuai

Tabel 3. Pengujian Black Box pada Menu Tambah Produk

Skenario Hasil Pengujian (Data Benar)	
Data Masukan	Kode IDS : 000100005102 Jumlah Stok : 20
Hasil yang diharapkan	Data tersimpan dalam aplikasi
Pengamatan	Data berhasil disimpan dan aplikasi menampilkan pesan "Produk berhasil ditambahkan"
Hasil	Sesuai
Skenario Hasil Pengujian (Data Kosong)	
Data Masukan	Jumlah stok dikosongkan
Hasil yang diharapkan	Data tidak dapat tersimpan
Pengamatan	Data tidak tersimpan dan aplikasi menampilkan pesan "Peringatan: Jumlah produk harus diisi"
Hasil	Sesuai

## 5. KESIMPULAN DAN SARAN

Berdasarkan seluruh tahapan hasil penelitian yang dilakukan untuk rancang bangun aplikasi *point of sales distro management system*, maka dapat disimpulkan rancang bangun aplikasi POS ini yang mulai dari melakukan pengumpulan data, perancangan sistem sampai ke penulisan program dan terakhir dengan melakukan pengujian menggunakan metode *black box testing* yang dimana aplikasi POS ini sudah berjalan sesuai dengan yang diharapkan, sehingga aplikasi POS ini berhasil dibuat dan juga sudah disesuaikan dengan kebutuhan pihak *distro* sehingga dapat membantu memudahkan pihak *distro* dalam melakukan transaksi penjualan dan manajemen stok produk.

Untuk pengembangan aplikasi *point of sales* ini selanjutnya dapat dilakukan dengan menambahkan metode pembayaran *payment gateway* seperti *midtrans*, *ovo* dan lain-lain, sehingga lebih memudahkan konsumen ketika ingin melakukan pembelian produk di *distro* karena terdapat banyak pilihan pembayaran.

## DAFTAR PUSTAKA

- [1] B. S. Rahardjo, *All About: Distro*. Jakarta: Elex Media Komputindo, 2009.
- [2] H. Susanto, "Perancangan Sistem Rekomendasi Pakaian Distro dengan Menggunakan Item Collaborative Filtering (Studi Kasus: The Jungle Distro Medan)," *Jurnal Pelita Informatika Budi Darma*, Volume VI, no. 3, 2014.
- [3] Hendry, *Membangun Aplikasi Point of Sale dengan VB 6.0, MySQL dan PHP*. Jakarta: Elex Meia Komputindo, 2010.
- [4] S. McCool, *Laravel Starter: The Definitive Introduction to the Laravel PHP Web Development Framework*. Birmingham: Publishing Ltd, 2012.
- [5] A. Handojo, Y. R. Otniel P and E. Kristianto J, "Pembuatan Aplikasi Sistem Pendaftaran Praktikum Online 71 pada Jurusan Teknik Informatika Universitas Kristen Petra dengan Menggunakan Php," *Jurnal Informatika*, vol. 4, no. 1, p. 10, 2003.

- [6] A. Hasyim, "Memulai Pengembangan Android dengan React Native di Windows." Internet: [www.codepolitan.com/memulai-pengembangan-android-dengan-react-native-di-windows-57b85678b26a9-17960](http://www.codepolitan.com/memulai-pengembangan-android-dengan-react-native-di-windows-57b85678b26a9-17960), Sep. 20, 2016 [Oct. 13, 2018].
- [7] B. Eisenman, *Learning React Native: Building Native Mobile Apps With Javascript*. California: O'Reilly Media Inc, 2015.
- [8] J. Kusnendar, "Perangkat Lunak untuk Mentransformasikan Model Entity Relationship ke Model Relational," *J. Pendidik. Teknol. Inf. dan Komun.*, vol. 2, no. 2, pp. 16–24, 2009.
- [9] Sugiono, *Metode Penelitian Pendidikan (Pendekatan Kuantitatif, Kualitatif dan R&D)*. Bandung: Alfabeta, 2011.
- [10] Sugiono, *Statistika untuk Penelitian*. Bandung: Alfabeta, 2013.
- [11] U. Narimawati, S. D. Anggadini and L. Ismawati, *Penulisan Karya Ilmiah: Panduan Awal Menyusun Skripsi dan Tugas Akhir Aplikasi pada Fakultas Ekonomi UNIKOM*. Bekasi: Genesis, 2010.
- [12] A. Nugroho, *Rekayasa Perangkat Lunak Berbasis Objek dengan Metode USDP*. Yogyakarta: Andi, 2010.
- [13] D. Edi and S. Betshani, "Analisis Data dengan Menggunakan ERD dan Model Konseptual Data Warehouse," *Jurnah Informatika*, vol. 5, no. 1, pp. 75-76, 2009.

#### BIODATA PENULIS



Putu Dika Arta Wiguna  
Mahasiswa Jurusan Sisem Informasi di Kampus  
STMIK Primakara.

I Putu Satwika  
Dosen di Kampus STMIK Primakara dan telah  
menyelesaikan program magister di Universitas  
Indonesia dengan bidang konsentrasi Ilmu  
Komputer.



I Putu Agus Swastika  
Dosen di Kampus STMIK Primakara dan telah  
menyelesaikan program magister di STIKOM  
Surabaya.

