

SISTEM INFORMASI PENJUALAN POINT OF SALE MENGGUNAKAN FRAMEWORK REACT NATIVE PADA TOKO IBNU ALI KECAMATAN BAWANG BANJARNEGARA

Rizki Aulia Ramadhan¹, Khalimaturofiah²

Teknik Informatika
STIMIK Tunas Bangsa Banjarnegara
stbbanjarnebara@gmail.com
rofi@stb.ac.id

Abstract

The sales system is one of the important business activities carried out by business people. The sales system at Toko Ibnu Ali is still simple and has several problems including the transaction process still using a calculator which sometimes inputs wrong numbers which can harm sellers and buyers, transaction notes that are still written manually on note books which are prone to errors in writing and the data is inaccurate, then there is no sales note bookkeeping which results in no sales reports. The implementation of a Point of Sale sales information system using the React Native framework was created to help the cashiers of the Ibnu Ali Store in solving problems that exist in this store. In this study, the method used is the waterfall method which starts from needs analysis, design, implementation, testing and finally maintenance. The results of system testing conducted by 5 respondents with 8 questions obtained an overall score of 82.5%. This result indicates that the point of sale sales system application can help and overcome the existing sales system problems at the Ibnu Ali store.

Keywords: Sales Information System Point of Sale, Framework React Native, Waterfall.

Pendahuluan

Toko Ibnu Ali adalah toko yang bergerak dibidang penjualan buku Islam, pakaian muslim dan herbal yang berlokasi di Blambangan, kecamatan Bawang, Banjarnegara. Toko Ibnu Ali menyediakan buku dan pakaian muslim mulai dari anak – anak hingga remaja serta menyediakan berbagai herbal yang sering dikonsumsi oleh masyarakat pada umumnya.

Sistem penjualan yang ada di toko ini masih sederhana dan memiliki beberapa permasalahan diantaranya:

1. Proses transaksi masih menggunakan kalkulator. Meski telah menggunakan kalkulator, perhitungan transaksi terkadang kasir salah *input* angka yang dapat merugikan pembeli dan penjual.
2. Nota transaksi yang masih ditulis manual pada buku nota. Semakin banyak transaksi yang berlangsung maka semakin banyak pula nota yang ditulis secara manual dan rentan terjadinya kesalahan dalam penulisan nota transaksi sehingga hal tersebut mengakibatkan data yang ditulis pada buku nota transaksi tidak akurat dan dapat merugikan pembeli. Nota transaksi yang ditulis dalam buku nota transaksi sekitar 50 – 100 transaksi perbulannya.
3. Belum adanya pembukuan nota penjualan yang menjadikan tidak adanya laporan penjualan.

Berdasarkan uraian permasalahan tersebut, maka peneliti mengambil judul “Sistem Informasi Penjualan *Point of Sale* menggunakan *framework React Native* Pada Toko Ibnu Ali Kecamatan Bawang Banjarnegara”. Aplikasi dari sistem ini diberi nama “YukKasir” dan aplikasi ini berbasis Android dan IOS dikarenakan *framework React Native* dapat berjalan di kedua sistem operasi tersebut.

Landasan Teori

Sistem Informasi Penjualan

Sistem informasi penjualan adalah sebuah prosedur yang melaksanakan, mencatat, mengalkulasi, membuat dokumen, dan informasi penjualan untuk keperluan manajemen. Sistem informasi penjualan merupakan sub sistem informasi bisnis, sub sistem bisnis lainnya bisa berupa pemasaran, sumber daya manusia, keuangan akuntansi dan manufaktur produksi (Rifal Nurjamil & Falentino Sembring, 2021).

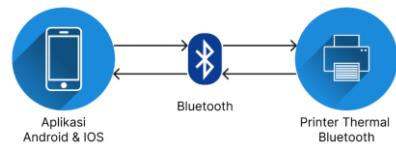
Sistem ini biasanya terdiri dari perangkat lunak (*software*) dan perangkat keras (*hardware*) yang digunakan untuk mengumpulkan, menyimpan, dan menganalisis data penjualan. Biasanya, sistem informasi penjualan juga menyediakan laporan penjualan yang dapat digunakan oleh perusahaan untuk mengambil keputusan strategis. Sistem informasi penjualan dapat juga digunakan untuk mengelola stok barang, mengelola transaksi

pembayaran, dan mengelola hubungan dengan pelanggan.

Point of Sale

Point of Sale adalah sebuah sistem yang bertujuan melakukan pengelolaan transaksi yang didalamnya termasuk penggunaan mesin kasir. Dalam lingkup *POS*, sebuah mesin kasir tidak independen (berdirisendiri), namun termasuk didalamnya *software* penunjang dan piranti lain. Sistem *POS* lebih mengacu kepada penggunaan teknologi untuk melakukan efisiensi transaksi, ketelitian dan kelengkapan pelaporan transaksi (Bilqis Nabila & Filda Angellia, 2020).

Pada penelitian ini, peneliti memanfaatkan koneksi *bluetooth* untuk menghubungkan perangkat lunak (*software*) yang sudah ter-*install* di *smartphone* dengan perangkat keras (*hardware*) *bluetooth Thermal printer*.



Gambar 1. Ilustrasi *Point of Sale*

Gambar diatas menjelaskan aplikasi *Point of Sale* yang saling terhubung dengan koneksi *Bluetooth* antara *software* aplikasi *mobile* (Android dan IOS) dan *hardware printer Thermal Bluetooth*. Aplikasi yang sudah terhubung dengan *printer Bluetooth* dapat melakukan berbagai perintah seperti *print* strok transaksi, *barcode* dan fungsi lainnya yang berkaitan dengan *Point of Sale*.

Framework React Native

Framework React Native adalah sebuah *framework* JavaScript yang dapat berjalan pada perangkat *mobile* IOS dan Android yang didirikan oleh Facebook pada tahun 2015 (Novrina, Ratih Nurdiani & Arya Rangga, 2022). Menurut *official React Native*, *React Native* adalah *framework* untuk membuat aplikasi *Native* untuk Android, IOS dan lainnya menggunakan *library React*. *React Native* menggabungkan bagian terbaik dari pengembangan *Native* dengan *library React* Pustaka Javascript terbaik dikelasnya untuk membangun antarmuka pengguna (Official React Native, 2023).



Gambar 2. React Native (sumber: React Native Official)

Dalam membuat sebuah aplikasi *mobile*, developer harus mempelajari spesifik bahasa pemrograman dari setiap *platform* seperti Android dengan bahasa pemrograman Kotlin dan IOS dengan bahasa pemrograman Swift. Dengan adanya React Native menjadikan para developer hanya mempelajari bahasa satu pemrograman saja yakni JavaScript, sehingga dengan satu bahasa pemrograman, dapat dijalankan pada perangkat Android dan IOS.

JavaScript

JavaScript adalah bahasa pemrograman *web* yang digunakan untuk memanipulasi elemen *HTML* dan membuat interaksi (Pratama, 2016). Sebagai contoh, apa yang terjadi ketika sebuah tombol diklik? Bagaimana membuat gambar muncul bergantian secara otomatis (*slider*), atau bagaimana cara mengubah warna sebuah kotak dari merah ke biru setelah halaman tampil selama 1 menit? Inilah yang bisa kita lakukan dengan JavaScript. JavaScript menambahkan aspek “pemrograman” ke dalam *HTML* dan *CSS*. Misalnya jika sebuah *tag* `<h1>` diklik, tampilkan isi seluruh artikel yang terdiri dari 10 *tag* `<p>`, atau jika sebuah gambar diklik, kita bisa mengubah *class* *CSS*-nya dari normal menjadi *warning*. JavaScript memiliki fitur dinamis, tidak bertipe dan diproses secara *interpreted*. JavaScript mirip dengan bahasa PHP dimana kita tidak perlu menetapkan sebuah *variable* harus bertipe *integer*, *float*, maupun *string*. Setiap *variable* di dalam JavaScript bisa diisi dengan tipe data apa saja dan kapan saja sepanjang kode program (bersifat dinamis).

NodeJs

Nodejs adalah *platform runtime* JavaScript yang dibangun di atas *engine* JavaScript V8 milik Google (Dicoding Indonesia, 2021). Nodejs menyediakan lingkungan *runtime* yang efisien dan mudah digunakan untuk menjalankan aplikasi *server-side* dengan JavaScript. Nodejs sangat populer di kalangan pengembang *web* karena memungkinkan mereka menulis *server-side code* dalam bahasa yang sama dengan yang digunakan untuk menulis *client-side code* (JavaScript), sehingga memudahkan pengembangan aplikasi web secara keseluruhan.

Nodejs sering digunakan dalam lingkungan pengembangan aplikasi React Native karena beberapa alasan:

- a. Nodejs diperlukan untuk menjalankan perintah-perintah dari *command line interface (CLI)* React Native, seperti `"react-native init"` untuk membuat proyek baru.
- b. Nodejs digunakan oleh *bundler* JavaScript (seperti *Metro* atau *Webpack*) yang digunakan oleh React Native untuk mengelompokkan file JavaScript menjadi sebuah *bundle* yang bisa dijalankan di aplikasi *mobile*.
- c. Nodejs juga dapat digunakan untuk menjalankan *server-side code* (misalnya, untuk mengakses *database* atau menangani *request API*) yang diintegrasikan dengan aplikasi *mobile* React Native.

Android

Android adalah sistem operasi berbasis Linux yang dirancang untuk perangkat bergerak layar sentuh seperti telepon pintar dan komputer tablet (Yudhanto, 2017). Android awalnya dikembangkan oleh Android Inc., dengan dukungan finansial dari Google, yang kemudian membelinya pada tahun 2005. Sistem operasi Android dirilis secara resmi pada tahun 2007 bersamaan dengan didirikannya *Open Handset Alliance (OHA)*, konsorsium dari perusahaan-perusahaan perangkat keras, perangkat lunak, dan telekomunikasi yang bertujuan untuk memajukan standar terbuka perangkat seluler.

Xcode

Xcode adalah lingkungan pengembangan terintegrasi Apple untuk macOS, digunakan untuk mengembangkan perangkat lunak untuk macOS, iOS, iPadOS, watchOS, dan juga tvOS. Ini awalnya dirilis pada akhir 2003. Lalu dirilis lagi pada tanggal 26 September 2022, melalui Mac App Store dengan macOS Monterey. Mereka juga masih menggunakan bahasa pemrograman C (Syahvira Nuzulia, 2022).

Xcode juga mendukung beberapa bahasa pemrograman seperti C, C++, Objective C dan juga yang terakhir mendukung bahasa Swift yang tersedia pada Xcode 6. Seperti pada umumnya IDE, maka Xcode menyediakan juga semua kebutuhan pemrograman seperti *editor*, *compiler*, *debugger* dan juga *run-time simulator*. Selain itu Xcode juga dilengkapi dengan fasilitas *management* pengembangan aplikasi seperti *repository* dan *version control*. Xcode ini juga sudah terintegrasi dengan aplikasi *git* dapat memudahkan kalian untuk melakukan kegiatan *software configuration management* untuk kerja *team*.

Firebase

Firebase adalah suatu layanan dari Google untuk memberikan kemudahan para *developer* aplikasi dalam mengembangkan aplikasinya. Dengan menggunakan Firebase, *developer* aplikasi

bisa fokus mengembangkan aplikasinya tanpa memikirkan *backend* yang terlalu dalam (Official firebase, 2022).

Firebase memiliki beberapa fitur diantaranya :

1. Firebase Authentication

Firebase Authentication adalah salah satu layanan *backend*, fitur Android dan iOS, *SDK* yang mudah digunakan, dan tampilan *interface* yang siap pakai untuk meng-*autentikasi* pengguna ke aplikasi yang kamu buat. *Firebase Authentication* mendukung *autentikasi* menggunakan nomor telepon, sandi, penyedia identitas gabungan populer seperti seperti Google, Facebook, dan sebagainya.

2. Firebase Realtime Database

Firebase Realtime Database adalah *database* yang di-*host* melalui *cloud*. Data disimpan dan dieksekusi dalam bentuk *JSON* dan disinkronkan secara *realtime* ke setiap *user* yang terkoneksi. Hal ini berfungsi memudahkan kamu dalam mengelola suatu *database* dengan skala yang cukup besar. Ketika kamu membuat aplikasi *lintas-platform multiplatform* menggunakan *SDK* Android, iOS, dan juga JS (JavaScript), semua pengguna akan berbagi sebuah *instance Realtime Database* dan menerima *update* -an data secara serentak dan otomatis.

3. Firebase Cloud Firestore

Firebase Cloud Firestore adalah *database* yang bersifat fleksibel dan terukur untuk pengembangan perangkat seperti *mobile*, *web*, dan *server* di *Firebase* dan *Google Cloud Platform*. *Firebase Cloud Firestore* merupakan *database NoSQL* yang *di-hosting* di *cloud* dan dapat diakses melalui *SDK real* oleh aplikasi iOS, Android dan *web*.

4. Firebase Cloud Messaging

FCM (Firebase Cloud Messaging) yaitu menyediakan koneksi yang handal dan tentunya hemat baterai antar *server* maupun antar *device*. Sehingga *user* dapat mengirim dan menerima pesan serta notifikasi di Android, iOS, dan *web* tanpa perlu biaya. Pesan notifikasi ini terintegrasi sepenuhnya dengan *Google Analytics*, sehingga *user* memiliki akses secara detail untuk interaksi dan *tracking* konversi.

Figma

Figma adalah salah satu *design tool* yang biasanya digunakan untuk membuat tampilan aplikasi *mobile*, *desktop*, *website* dan lain-lain (Muhyidin, 2020). Figma merupakan salah satu *web apps design* sehingga dapat diakses lewat *browser* dan dapat berjalan di sistem operasi Windows, Linux dan Mac. Umumnya Figma banyak digunakan oleh seseorang yang bekerja dibidang *UI/UX*, *web design* dan bidang lainnya yang sejenis.

Barcode

Barcode merupakan kode buat mendefinisikan huruf serta angka yang terdiri dari campuran garis dengan pengaturan jarak yang berbeda-beda (Hasanah & Harani & Riza, 2020). Data pada *Barcode* berisi enkripsi dari beberapa digit angka. Pada saat *Barcode* tersebut di *scan* dengan menggunakan alat *barcode scanner*, hingga kode tersebut secara otomatis tersambung ke informasi produk yang telah ditaruh dalam *Database*. Hasil dari pemindaian tersebut berisi data- data dari bermacam produk semacam nama produk, harga serta informasi yang lain cocok dengan apa yang telah dimasukkan pada *Database*. Pada penelitian kali ini, peneliti men-*scan barcode* menggunakan kamera *smartphone* Android dan IOS.

Unified Modeling Language (UML)

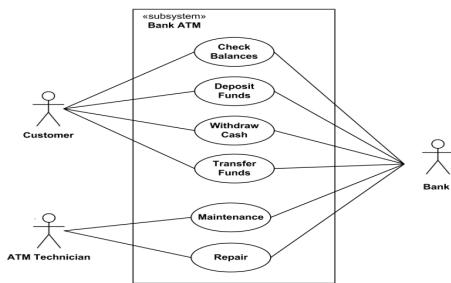
Unified Modeling Language (UML) adalah salah satu metode dalam teknik rekayasa perangkat lunak yang digunakan untuk menggambarkan alur dan cara kerja sistem, fungsi, tujuan dan mekanisme kontrol sistem tersebut (Abdillah, 2019). *UML* diciptakan oleh *Object Management Group* dengan versi awal 1.0 pada bulan Januari 1997. *UML* juga dapat didefinisikan sebagai suatu bahasa standar untuk penulisan *blueprint* sebuah *software*. Adapun tujuan dan fungsi *UML* yaitu sebagai berikut:

- a. Dapat memberikan bahasa pemodelan *visual* atau gambar kepada para pengguna dari berbagai macam pemrograman maupun proses umum rekayasa.
- b. Menyatukan informasi-informasi terbaik yang ada dalam pemodelan.
- c. Memberikan suatu gambaran model atau sebagai bahasa pemodelan visual yang ekspresif dalam pengembangan sistem.
- d. Tidak hanya menggambarkan model sistem *software* saja, namun dapat memodelkan sistem berorientasi objek atau yang biasa disebut *object oriented programming (OOP)*.

Unified Modeling Language (UML) memiliki berbagai model, berikut ini contoh dari model *UML*:

a. Model Use Case Diagram

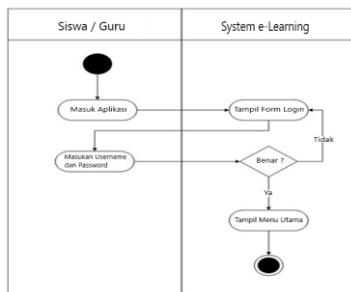
Use Case diagram digunakan untuk mengkomunikasikan interaksi manusia (*actor*) dengan apa yang bisa dilakukan oleh sistem. Sebuah *Use Case* dapat mewakili beberapa jalur interaksi manusia dengan sistem dan setiap jalur disebut sebagai skenario.



Gambar 3. Contoh *Use Case Diagram*

b. Model *Activity Diagram*

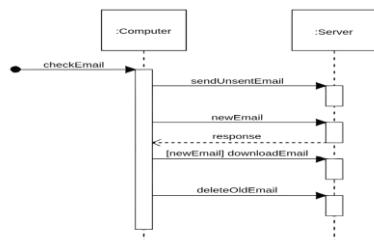
Activity Diagram atau dalam bahasa Indonesia berarti diagram aktivitas, merupakan sebuah diagram yang dapat memodelkan berbagai proses yang terjadi pada sistem. Seperti layaknya runtutan proses berjalannya suatu sistem dan digambarkan secara vertikal. *Activity Diagram* adalah salah satu contoh diagram dari *UML* dalam pengembangan dari *Use Case*.



Gambar 4. Contoh *Activity Diagram*

c. Model Sequence Diagram

Sequence Diagram mengilustrasikan objek-objek yang terdapat pada *Use Case* dan menggambarkan arus pesan antara satu sama lain pada Objek *Use Case*. *Sequence Diagram* bersifat dinamis dan lebih banyak menampilkan aktivitas objek berdasarkan urutan waktu.

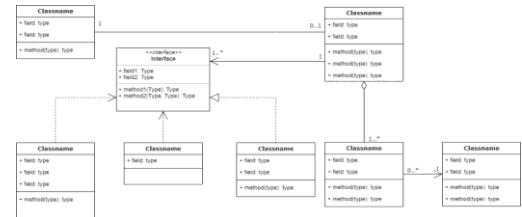


Gambar 5. Contoh Model *Sequence Diagram*

d. Model *Class Diagram*

Class Diagram adalah model statis yang mendukung tampilan data dan informasi dari keseluruhan sistem. Penggunaan *Class Diagram* dikaitkan dengan struktur basis data sistem atau dapat menggantikan *ERD* pada proses

penggambaran diagram rekayasa perangkat lunak yang konvensional.



Gambar 6. Contoh *Class Diagram*

Waterfall

Metode *Waterfall* adalah pengembangan perangkat lunak yang dibuat secara berurutan, dengan melakukan pekerjaan dari atas sampai kebawah seperti air terjun dengan melewati tahapan *Requirement, Design, Implementation, Verification* dan *Maintenance* (Driyani, 2018). Fase-fase dalam Metode *Waterfall* menurut referensi Driyani adalah sebagai berikut :

1. *Requirement* (analisis kebutuhan)

Pada fase ini dilakukan Analisa untuk kebutuhan sistem, pengumpulan data dengan cara melakukan penelitian, wawancara atau studi literatur.

2. *Design user interface* (desain tampilan)

Pada fase ini dilakukan pembuatan desain *UI* bedasarkan kebutuhan perangkat lunak untuk dapat diimplementasi.

3. *Implementation* (penerapan)

Fase inilah merupakan tahapan nyata untuk mengerjakan sistem. Tahapan dalam pembuatan desain diubah menjadi program diintegrasikan menjadi sistem secara keseluruhan serta perangkat lunak seutuhnya.

4. *Verification* (integrasi dan pengetesan)

Pada fase ini, merupakan final untuk pembuatan sebuah sistem. Yaitu tahapan verifikasi oleh pengguna. Pengguna akan menguji aplikasi sudah sesuai dengan kebutuhan pengguna.

5. *Maintenance* (Pemeliharaan)

Fase akhir yaitu pemeliharaan berupa instalasi dan proses perbaikan sistem berdasarkan kebutuhan pengguna.

Pengujian Sistem

Merupakan kegiatan yang dilakukan untuk mengetahui apakah pemrograman telah dilakukan dengan benar sehingga menghasilkan fungsi yang dikehendaki. Pengujian juga dimaksudkan untuk mengetahui keterbatasan dan kelemahan sistem yang dibuat agar sebisa mungkin dilakukan penyempurnaan.

Peneliti mengambil pengujian sistem dengan menggunakan metode pengujian kotak hitam (*black box*) dan kotak putih (*White box*) :

a. *Black box Testing*

Black box testing atau dapat disebut juga *Behavioral Testing* adalah pengujian yang dilakukan untuk mengamati hasil *input* dan *output* dari perangkat lunak tanpa mengetahui struktur kode dari perangkat lunak. Pengujian ini dilakukan di akhir pembuatan perangkat lunak untuk mengetahui apakah perangkat lunak dapat berfungsi dengan baik (Setiawan, 2021).

b. *White box Testing*

White box testing atau yang dapat diartikan menjadi “pengujian kotak putih” adalah pengujian yang dilakukan untuk menguji perangkat lunak dengan cara menganalisa dan meneliti struktur *internal* dan kode dari perangkat lunak. Lain halnya dengan *black box testing* yang hanya melihat hasil *input* dan *output* dari perangkat lunak, pengujian *white box testing* berfokus pada aliran *input* dan *output* dari perangkat lunak (Setiawan, 2021).

Skala Likert

Skala likert digunakan untuk mengukur sikap, pendapat dan persepsi seseorang atau sekelompok orang tentang kejadian atau gejala sosial yang telah ditetapkan secara spesifik oleh peneliti (selanjutnya disebut *variabel* penelitian). Dengan menggunakan *skala likert*, *variabel* yang akan diukur dijabarkan menjadi dimensi. Dimensi lalu dijabarkan menjadi *subvariabel*. Kemudian, *subvariabel* dijabarkan menjadi indikator-indikator yang dapat diukur. Akhirnya, indikator-indikator terukur dapat dijadikan titik tolak untuk membuat item instrumen yang berupa pertanyaan atau pertanyaan yang perlu dijawab oleh responden penelitian (Sudaryono, 2015). Setiap jawaban dihubungkan dengan bentuk pertanyaan atau dukungan sikap yang diungkapkan dengan kata-kata sebagai berikut :

- | | |
|------------------------------|-----|
| a. Sangat Setuju (SS) | = 5 |
| b. Setuju (S) | = 4 |
| c. Netral (N) | = 3 |
| d. Tidak Setuju (TS) | = 2 |
| e. Sangat Tidak Setuju (STS) | = 1 |

Metode Penelitian

Metode penelitian yang digunakan oleh peneliti adalah metode kualitatif. Menurut Ramdhani (2021:6-7) menyatakan bahwa jenis penelitian kualitatif memiliki sifat deskriptif dan cenderung menggunakan analisis. Proses dan makna lebih ditonjolkan dalam jenis penelitian ini dengan landasan teori yang dimanfaatkan sebagai pemandu agar fokus penelitian sesuai dengan fakta di lapangan. Hasil dari riset kualitatif memerlukan kedalaman analisis dari peneliti.

Metode Pengembangan Sistem

Pada penelitian ini, peneliti menggunakan metode *waterfall*. Metode ini sering juga disebut

model sequensial linear (*sequential linear*) atau alur hidup klasik (*classic life cycle*). Metode ini dalam pengembangan perangkat lunak dibuat secara berurutan dengan melakukan pekerjaan dari atas sampai kebawah seperti air terjun dengan melewati tahapan *Requirement*, *Design*, *Implementation*, *Verification* dan *Maintenance*.

Fase – fase dalam metode Waterfall yang peneliti lakukan adalah sebagai berikut :

1. Analisis kebutuhan (*Requirement*)
2. Pembuatan kode program (*Implementation*)
3. Pengujian (*Verification*)
4. Pemeliharaan (maintenance)

Metode Pengumpulan Data

1. Wawancara
2. Observasi
3. Dokumen

Metode Analisis Data

Metode analisis data yang peneliti lakukan diantaranya :

1. Analisis kebutuhan: Menentukan kebutuhan sistem yang diperlukan oleh toko Ibnu Ali dari hasil wawancara dan analisis dokumen yang ada. Berdasarkan hasil wawancara yang peneliti lakukan, toko Ibnu Ali membutuhkan sebuah sistem informasi untuk mengatasi permasalahan yang ada di toko ini.
2. Analisis Sistem Yang dibutuhkan: Mengidentifikasi sistem yang akan dibuat sesuai dengan kebutuhan. Sistem yang dibutuhkan pada toko ini adalah sistem yang dapat melakukan proses transaksi mulai dari perhitungan hingga cetak nota transaksi secara otomatis.
3. Analisis data : Menganalisis data yang dibutuhkan oleh sistem. Data yang dibutuhkan oleh sistem pada toko Ibnu Ali adalah data produk di toko ini. Setelah melakukan observasi, produk di toko Ibnu Ali terdiri dari 4 kategori diantaranya buku, pakaian, herbal dan parfum.
4. Analisis kebutuhan teknis : Menentukan spesifikasi yang diperlukan untuk membangun sistem informasi ini dan kompatibilitas dengan perangkat keras. Berikut ini tabel spesifikasi yang diperlukan dalam pembuatan sistem informasi dan kompatibilitas dengan perangkat keras.

Analisis Dan Pembahasan

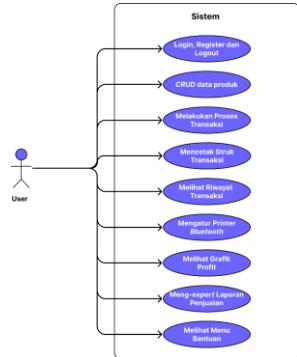
Rancangan Sistem

Pada tahap proses perancangan sistem ini akan digambarkan secara garis besar tentang program sistem informasi penjualan *Point of Sale* menggunakan *framework* React Native pada toko Ibnu Ali. Berikut gambaran proses rancangan sistem yang akan dibuat :

1. Menentukan desain proses masukan dan keluaran program menggunakan *Use Case Diagram*, *Activity Diagram*, *Sequence Diagram*, *Class Diagram* dan *database*.
2. Menentukan desain *user interface* menggunakan *software Figma*.
3. Implementasi pembuatan aplikasi yang sesuai dengan desain proses dan desain *user interface*

a. *Use Case Diagram*

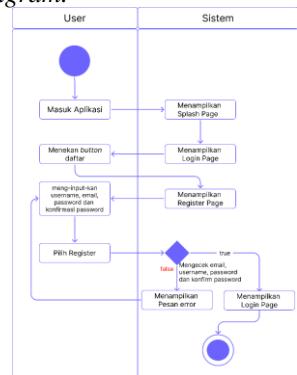
Use Case Diagram sistem informasi penjualan *Point of Sale* menggunakan *framework React Native*.



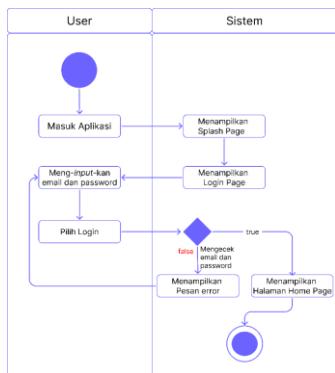
Gambar 7. *Use Case Diagram*

b. *Activity Diagram*

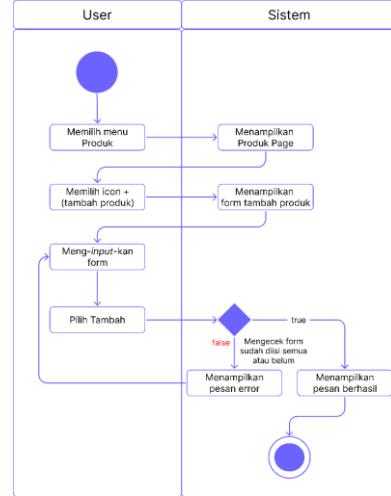
Activity Diagram merupakan diagram yang menggambarkan aliran kerja atau aktivitas dari sebuah sistem. Pada sistem ini terdapat 12 proses *Activity Diagram*.



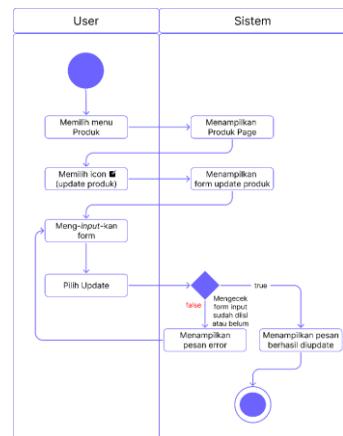
Gambar 8. *Activity Diagram Register*



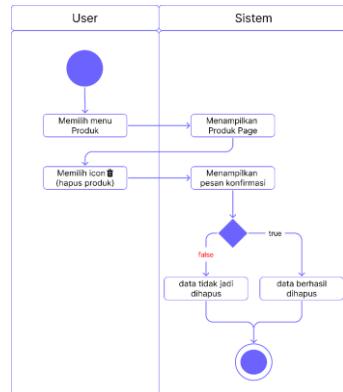
Gambar 9. *Activity Diagram Login*



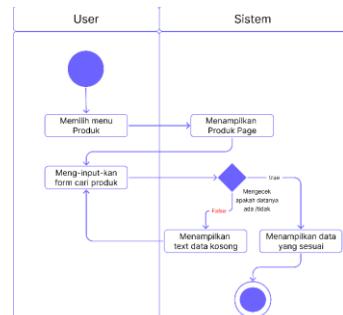
Gambar 10. *Activity Diagram* tambah produk



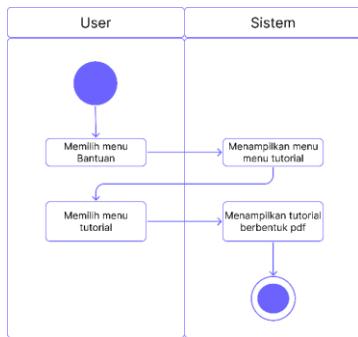
Gambar 11. *Activity Diagram* update produk



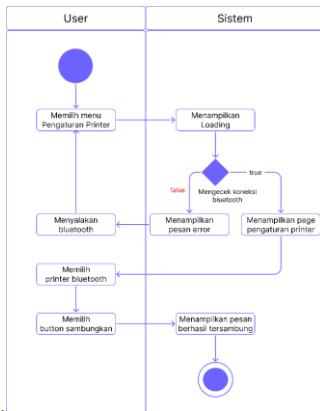
Gambar 12. *Activity Diagram* delete produk



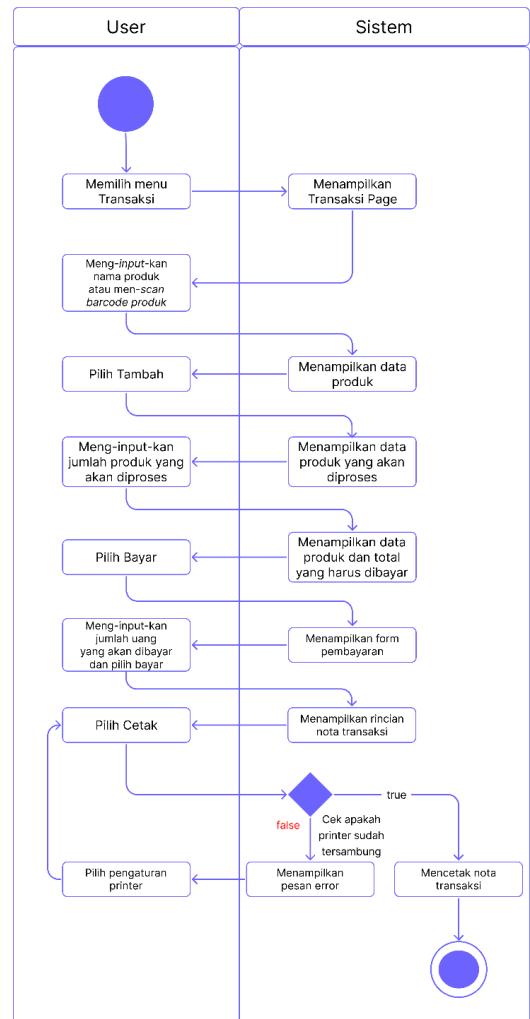
Gambar 13. *Activity Diagram* search produk



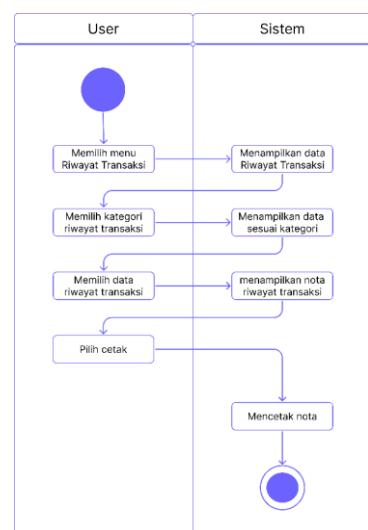
Gambar 14. *Activity Diagram* menu bantuan



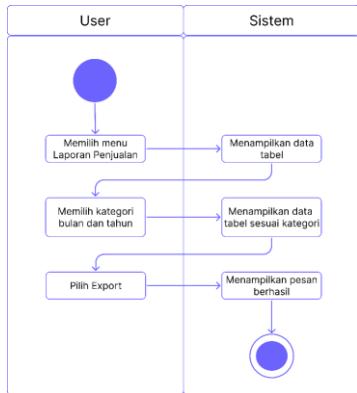
Gambar 15. *Activity Diagram* menu pangaturan printer



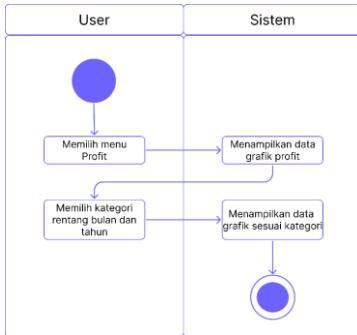
Gambar 16. *Activity Diagram* menu transaksi



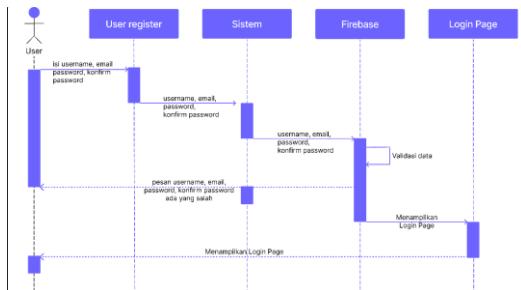
Gambar 17. *Activity Diagram* menu riwayat transaksi



Gambar 18. *Activity Diagram* menu laporan penjualan

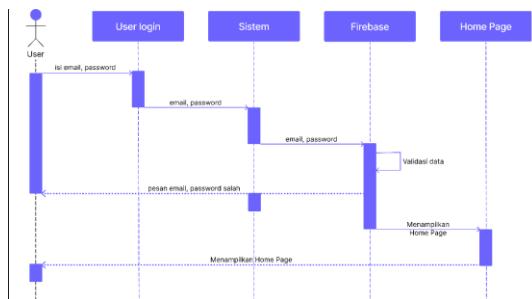


Gambar 19 *Activity Diagram* menu profit



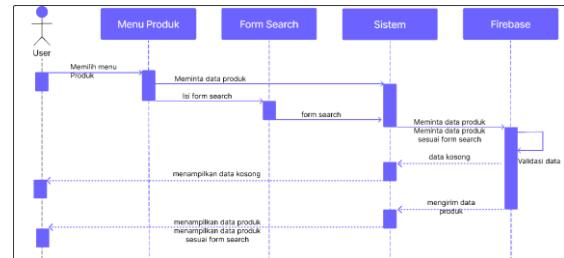
Gambar 20 *Sequence Diagram* Register

Sequence Diagram pada gambar terdiri dari *user* dan 4 objek yaitu *user register*, sistem, *Firebase* dan *login page*.



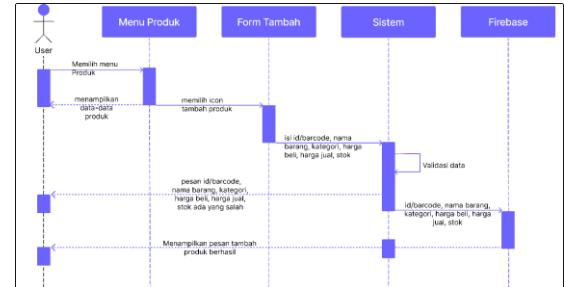
Gambar 21. *Sequence Diagram* login

Sequence Diagram pada gambar terdiri dari *user* dan 4 objek yaitu *user login*, sistem, *Firebase* dan *login page*.



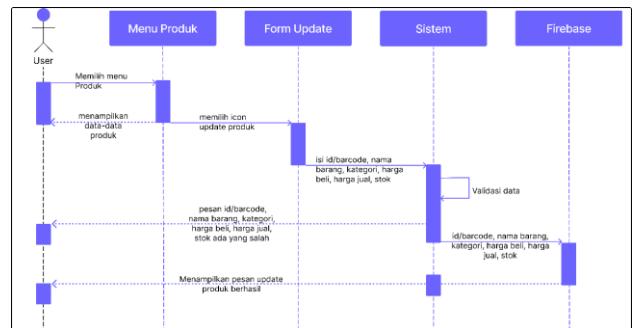
Gambar 22. *Sequence Diagram* read dan search data produk

Sequence Diagram pada gambar terdiri dari *user* dan 4 objek yaitu *menu produk*, *form search*, sistem dan *Firebase*.



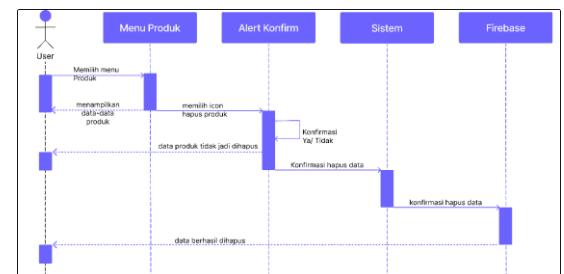
Gambar 23 *Sequence Diagram* create produk

Sequence Diagram pada gambar terdiri dari *user* dan 4 objek yaitu *menu produk*, *form tambah*, sistem dan *Firebase*.



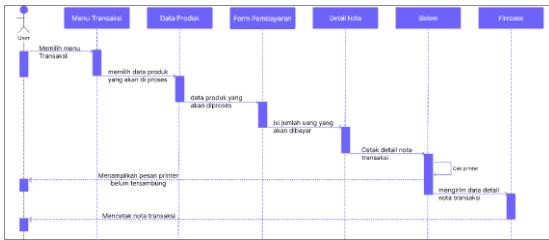
Gambar 24. *Sequence Diagram* update data produk

Sequence Diagram pada gambar terdiri dari *user* dan 4 objek yaitu *menu produk*, *form update*, sistem dan *Firebase*.



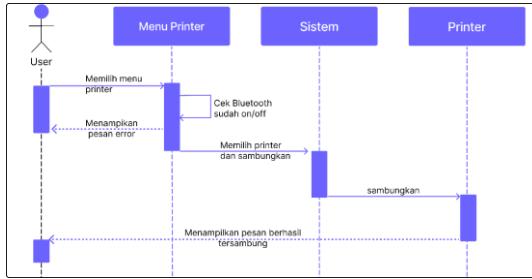
Gambar 25. *Sequence Diagram* delete data product

Sequence Diagram pada gambar terdiri dari *user* dan 4 objek yaitu *menu produk*, *alert konfirmasi*, sistem dan *Firebase*.



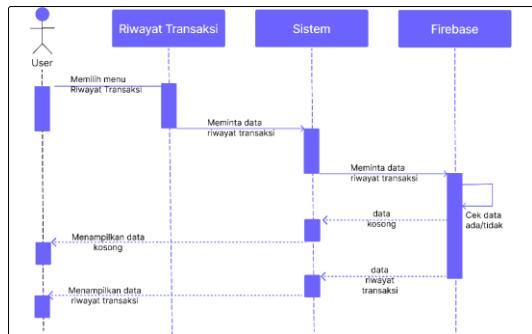
Gambar 26. Sequence Diagram Transaksi

Sequence Diagram pada gambar terdiri dari *user* dan 6 objek yaitu menu *transaksi*, data produk, *form pembayaran*, detail nota, sistem dan Firebase.



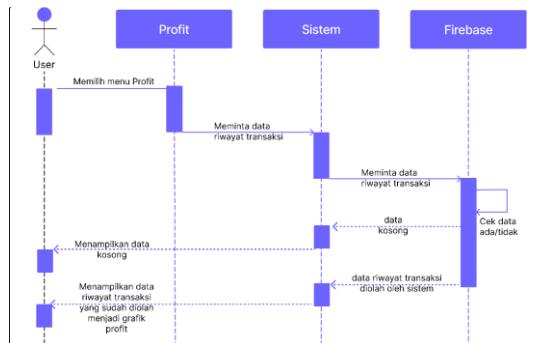
Gambar 27. Sequence Diagram pengaturan *printer*

Sequence Diagram pada gambar terdiri dari *user* dan 3 objek yaitu menu *printer*, sistem dan *printer*.



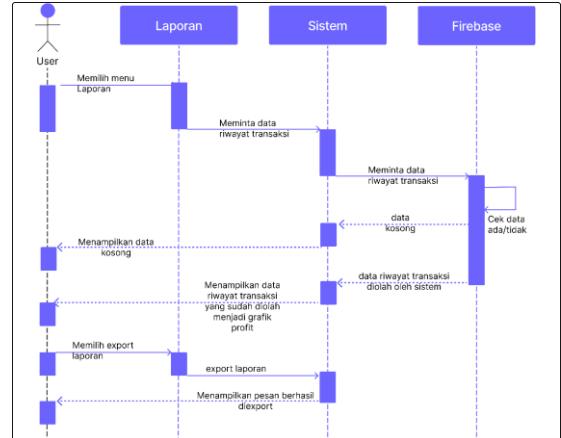
Gambar 28. Sequence Diagram riwayat transaksi

Sequence Diagram pada gambar terdiri dari *user* dan 3 objek yaitu menu riwayat transaksi, sistem dan Firebase.



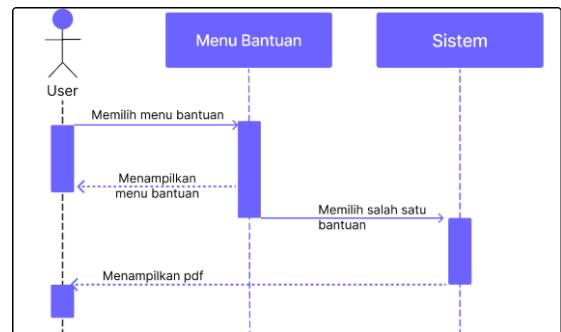
Gambar 29. Sequence Diagram profit transaksi

Sequence Diagram pada gambar terdiri dari *user* dan 3 objek yaitu menu *profit*, sistem dan Firebase.



Gambar 30. Sequence Diagram laporan

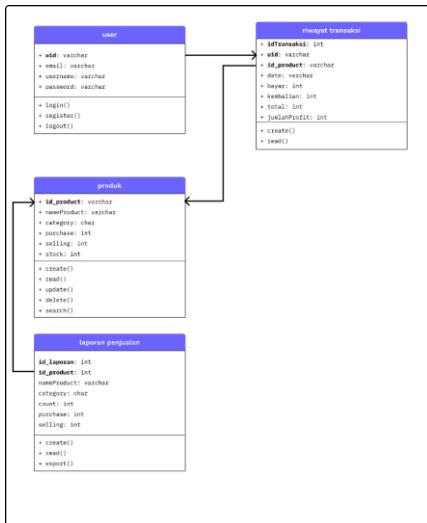
Sequence Diagram pada gambar terdiri dari *user* dan 3 objek yaitu menu laporan, sistem dan Firebase.



Gambar 31. Sequence Diagram bantuan

Class Diagram

Class Diagram merupakan *diagram* yang digunakan untuk menampilkan beberapa kelas serta paket-paket yang ada pada sistem aplikasi. Model Class Diagram sistem ini dapat dilihat pada Gambar .



Gambar 32. Model Class Diagram

Gambar menjelaskan bahwa *Class Diagram* dari sistem ini memiliki 4 kelas yang terdiri dari *user*, *riwayat transaksi*, *produk* dan *laporan penjualan*.

Rancangan Database

Database adalah sekumpulan data yang dikelola bedasarkan ketentuan tertentu yang saling berkaitan sehingga memudahkan dalam pengelolaannya.

Tabel 1. Database *users*

No	Name	Type	Length	Keterangan
1.	<i>uid</i>	<i>uuid</i>	28	<i>Primary key</i>
2.	<i>email</i>	<i>varchar</i>	20	
3.	<i>username</i>	<i>varchar</i>	20	
4.	<i>password</i>	<i>varchar</i>	10	

Tabel 2. Database *produk*

No	Name	Type	Length	Keterangan
1.	<i>Id_product</i>	<i>varchar</i>	13	<i>Primary key</i>
2.	<i>nameProduct</i>	<i>varchar</i>	50	
3.	<i>category</i>	<i>char</i>	7	
4.	<i>purchase</i>	<i>int</i>	9	
5.	<i>selling</i>	<i>int</i>	9	
6.	<i>stock</i>	<i>int</i>	4	

Tabel 3. Database *riwayat transaksi*

No	Name	Type	Length	Keterangan
1.	<i>idTransaksi</i>	<i>int</i>	12	<i>Primary key</i>
2.	<i>uid</i>	<i>uuid</i>	28	<i>Foreign key</i>

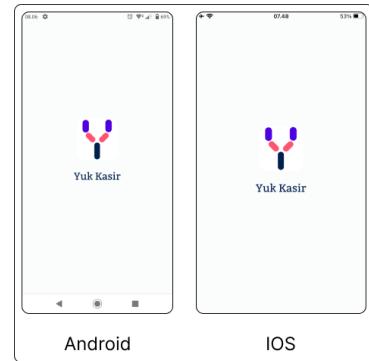
3.	<i>Id_product</i>	<i>varchar</i>	13	<i>Foreign key</i>
4.	<i>date</i>	<i>date</i>		
5.	<i>bayar</i>	<i>int</i>	9	
6.	<i>kembalian</i>	<i>int</i>	9	
7.	<i>total</i>	<i>int</i>	9	
8.	<i>jumlahProfit</i>	<i>int</i>	9	

Tabel 4. Database laporan penjualan

No	Name	Type	Length	Keterangan
1.	<i>Id_laporan</i>	<i>int</i>	13	<i>Primary key</i>
2.	<i>Id_product</i>	<i>varchar</i>	13	<i>Foreign key</i>
3.	<i>nameProduct</i>	<i>varchar</i>	50	
4.	<i>category</i>	<i>char</i>	7	
5.	<i>count</i>	<i>int</i>	4	
6.	<i>purchase</i>	<i>int</i>	9	
7.	<i>selling</i>	<i>int</i>	9	

Implementasi

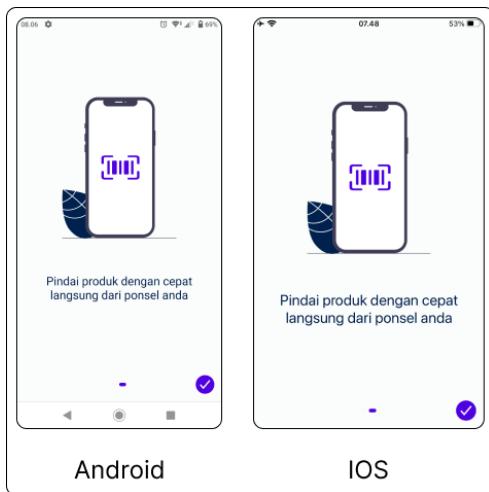
Bagian ini berisi tentang implementasi sistem secara menyeluruh sesuai dengan hasil analisis dan perancangan sistem. Adapun implementasi pada aplikasi “YukKasir”.



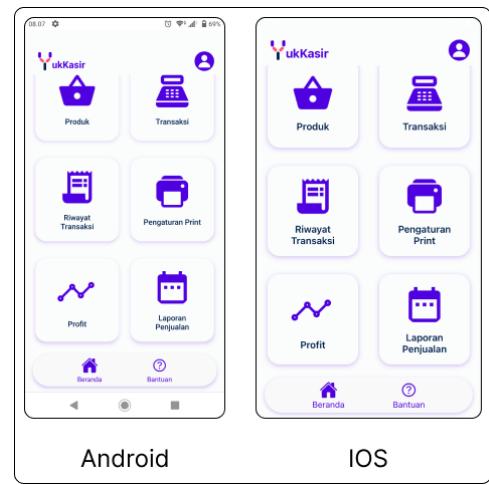
Gambar 33. Implementasi *splash page*



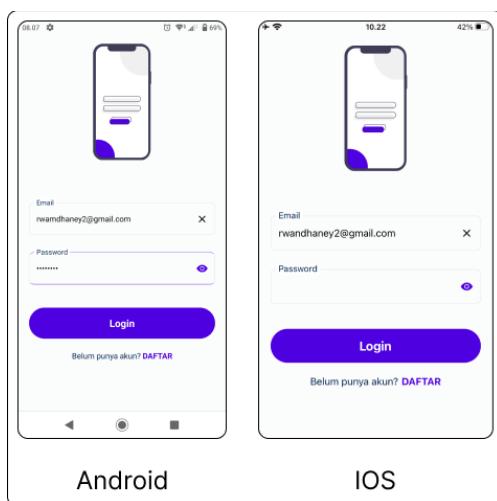
Gambar 34. Implementasi halaman *onboarding*



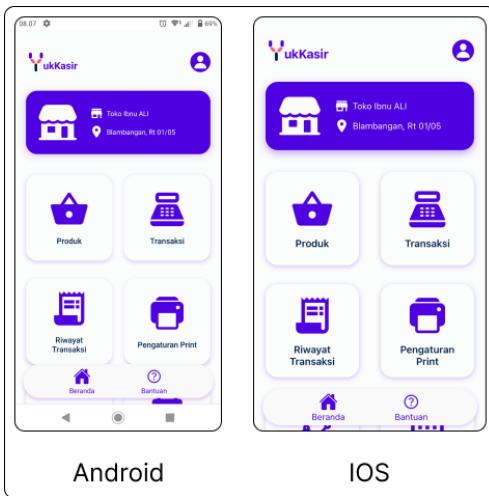
Gambar 35. Implementasi halaman *onboarding* lanjutan



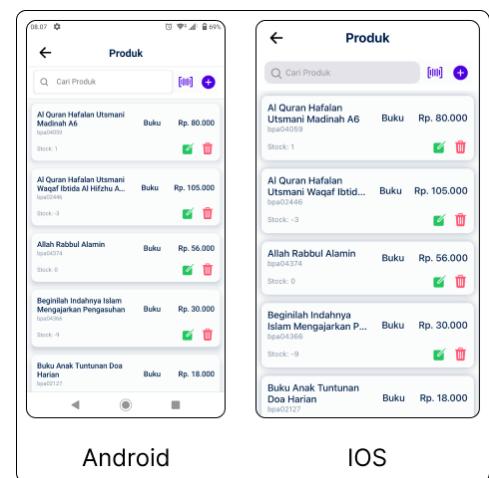
Gambar 38. Implementasi halaman *home* lanjutan



Gambar 36. Implementasi halaman *login*



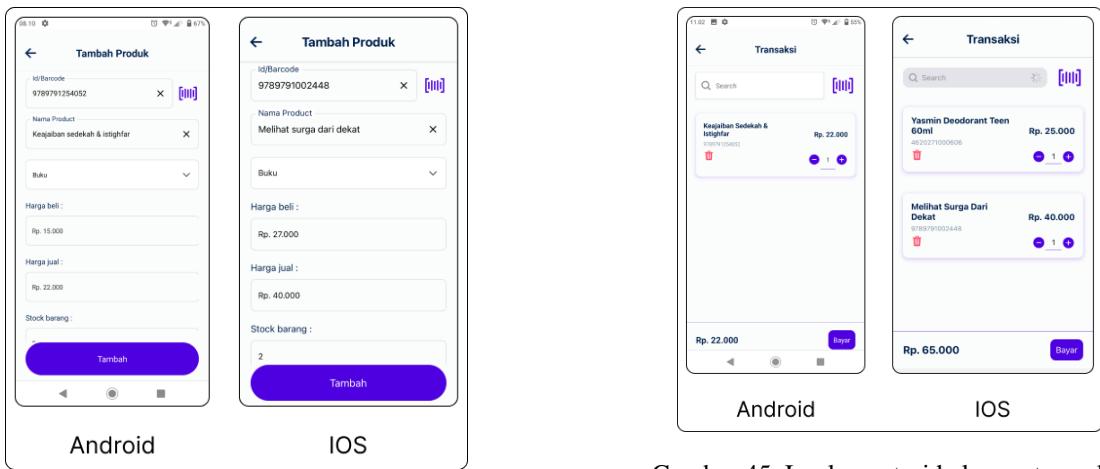
Gambar 37. Implementasi halaman *home*



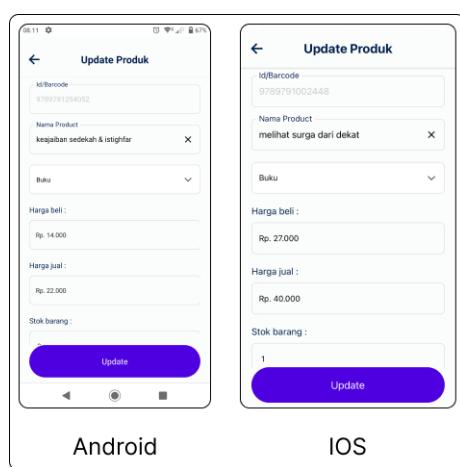
Gambar 39. Implementasi halaman produk



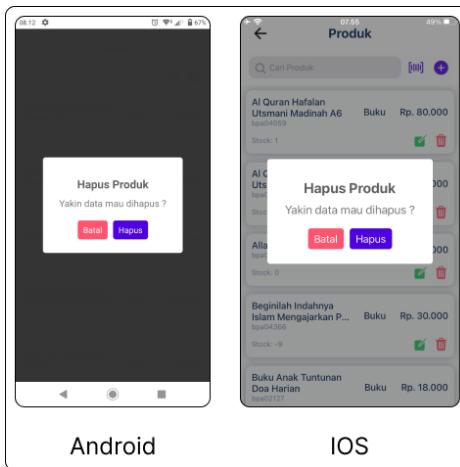
Gambar 40. Implementasi *scan barcode*



Gambar 41. Implementasi tambah produk

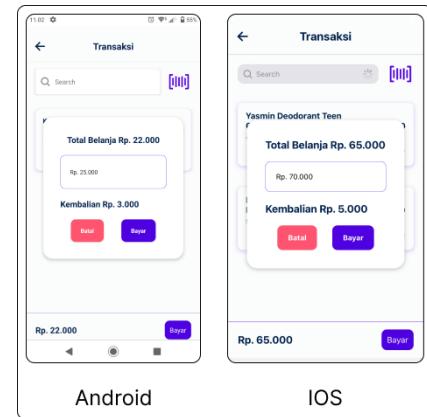


Gambar 42. Implementasi update produk

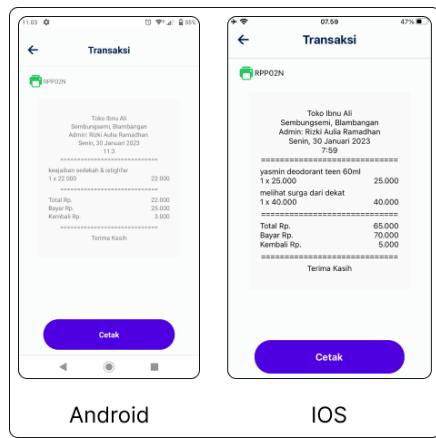


Gambar 43. Implementasi hapus produk

Gambar 45. Implementasi halaman transaksi



Gambar 46. Implementasi form pembayaran



Gambar 47. Implementasi halaman detail transaksi

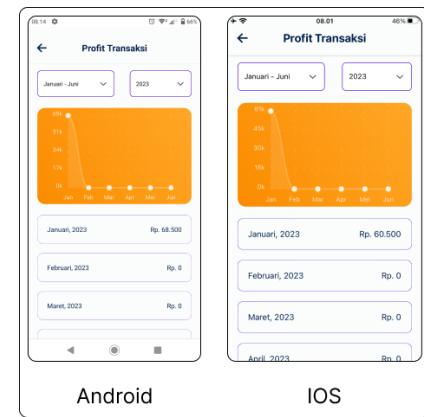




Android

IOS

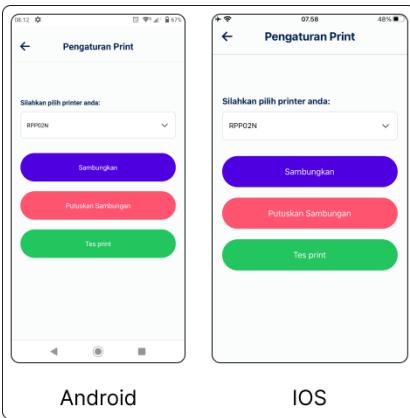
Gambar 48. Implementasi cetak nota transaksi



Android

IOS

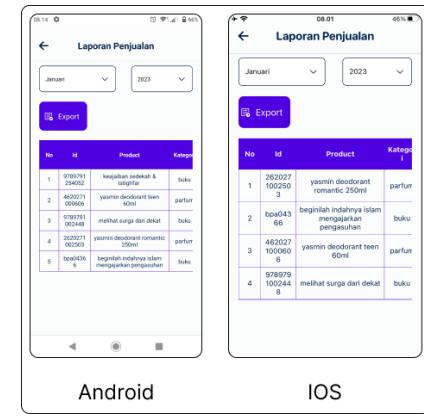
Gambar 51. Implementasi halaman *profit transaksi*



Android

IOS

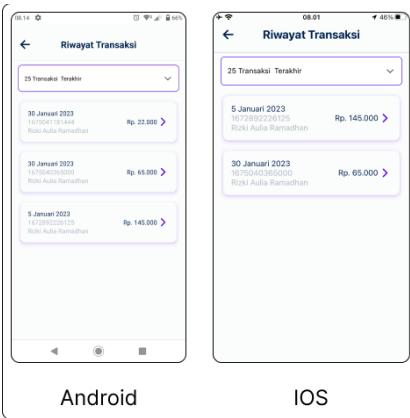
Gambar 49. Implementasi halaman pengaturan *printer*



Android

IOS

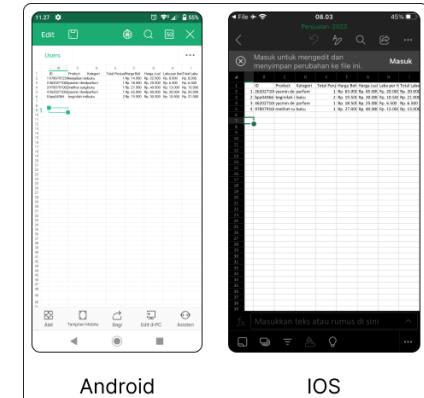
Gambar 52. Impelentasi halaman laporan



Android

IOS

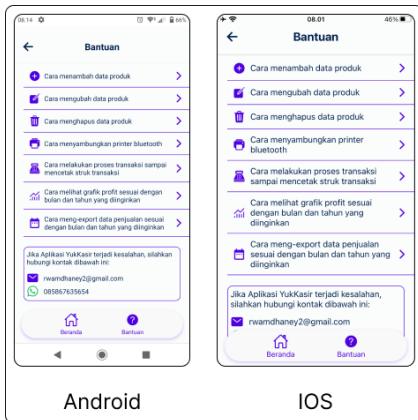
Gambar 50. Implementasi halaman riwayat transaksi



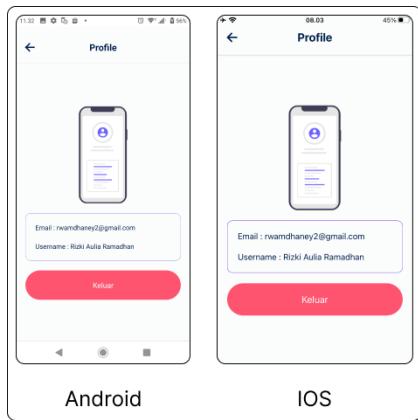
Android

IOS

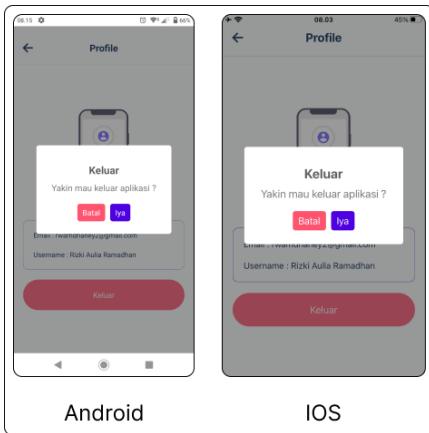
Gambar 53. Implementasi *export* data penjualan



Gambar 54. Implementasi halaman bantuan



Gambar 55. Implementasi halaman *profile*



Gambar 56. Implementasi *logout*

Pengujian Sistem

Pengujian sistem merupakan kegiatan yang dilakukan untuk mengetahui apakah aplikasi yang peneliti buat telah dilakukan dengan benar sehingga menghasilkan fungsi yang diinginkan. Pengujian ini juga bermaksud untuk mengetahui keterbatasan dan kelemahan sistem yang dibuat agar sebisa mungkin dilakukan perbaikan. Dari hasil perhitungan *skala likert* kuesioner secara keseluruhan dari 5 responden

dengan 8 pertanyaan yang peneliti ajukan maka skor keseluruhannya adalah 82,5%.

Kesimpulan

Berdasarkan hasil penelitian serta pembahasan yang dilakukan oleh peneliti, dapat diambil kesimpulan bahwa hasil pengujian sistem secara keseluruhan yang dilakukan oleh 5 responden dengan 8 pertanyaan diperoleh skor rata – rata 82,5% yang menandakan aplikasi sistem penjualan *Point of Sale* tersebut dapat membantu mengatasi masalah proses transaksi, nota transaksi dan pembukuan nota penjualan yang ada pada toko Ibnu Ali.

Saran

Saran yang dapat peneliti sampaikan yang dapat digunakan sebagai bahan pertimbangan adalah sebagai berikut:

1. Aplikasi ini masih bisa dikembangkan pada *platform website*, khususnya *user* dengan role admin yang bertujuan mengelola semua data yang ada pada aplikasi YukKasir.
2. Aplikasi ini belum menerapkan *design thinking*, agar para *user* nyaman dalam menggunakan aplikasi.
3. Belum adanya fitur *generate barcode* untuk produk yang tidak ada *barcode*-nya.

Daftar Pustaka

Abdillah, R.;, Kuncoro, A.;, Kurniawan, I., 2019, Analis Aplikasi Pembelajaran Matematika Berbasis Android dan Desain Sistem Menggunakan *UML* 2.0, Jurnal The Original Research of Mathematics (THEOREMS), Universitas Majalengka, Majalengka, ISSN: 2541-4321, Vol.4 No.1 Juli 2019

Driyani, D., 2018, Perancangan Media Pembelajaran Sekolah Dasar Berbasis Android Menggunakan Metode Rekayasa Perangkat Lunak Air Terjun (*Waterfall*), Satuan Tulisan Riset dan Inovasi Teknologi (STRING), Universitas Indraprasta PGRI, Jakarta Selatan, ISSN: 2549-2837, Vol.3 No.1 Agustus 2018

Hasanah, M.;, Harani, NH.;, Riza, N., 2020, Implementasi *Barcode* dan Algoritma Regresi Linear Untuk Memprediksi Data Persediaan Barang, Kreatif Industri Nusantara, Bandung

Indonesia, D., 16 Januari 2023, Tutorial Node.js: Apa itu Node.js dan dasar-dasar Node.js <https://www.dicoding.com/blog/apa-itu-node-js/>

- Johnson, B. (2019). Visual Studio Code. In Visual Studio Code. <https://doi.org/10.1002/9781119588238>
- Muhyidin, MA.; Sulhan, MA.; Sevtiana, A., 2018, Perancangan *UI/UX* Aplikasi MY CIC Layanan Informasi Akademik Mahasiswa Menggunakan Aplikasi Figma, *Digital Of Information Technology* (DIGIT), LPPM Universitas Catur Insan Cedekia, Cirebon, ISSN : 2088-589x, Vol.10 No.2 November 2020
- Nurjamil, R.;, Sembiring, F., 2021, Perancangan Sistem Informasi Penjualan Jasa Pembuatan Furniture Berbasis Web (Studi Kasus Design Interior Concept Modern), Jurnal LARIK, Universitas Gunadarma, Jakarta Pusat, ISSN: 2808-1730, Vol.2 No.1 Juli 2021
- Novrina, Sari, R.N.;, 2022, Pemanfaatan React Native dan Firebase untuk Pembuatan Aplikasi Jual Beli Item Game Onlie, SISMATIK (Seminar Nasional Sistem Informasi dan Manajemen Informasi), Universitas Nusa Putra, Sukabumi, Vol.1 No.1 Agustus 2021
- Pamungkas, G.;, Yuliansyah, H., 2017, Rancang Bangun Aplikasi Android POS (*Point of Sale*) Kafe untuk Kasir *Portable* dan *Bluetooth Printer*, Jurnal Sains dan Teknologi, Universitas Pendidikan Ganesha, Sumatera Selatan, e-ISSN: 2548-8570, Vol.6 No.1 April 2017
- Prabowo, M., 2020, Metode Pengembangan Sistem Informasi, Lembaga Penelitian dan Pengabdian kepada Masyarakat (LP2M) IAIN Salatiga, Salatiga
- Pratama, A., 2016, *Javascript uncover*, Leanpub, British
- Pratama, R. Y.;, Somya, R., 2021, Perancangan Aplikasi *Point of Sales* (POS) Berbasis Android Studi Kasus Warkop Vape Salatiga, Jurnal Teknik Informatika dan Sistem Informasi (JATISI), STMIK Global Informatika MDP, Sumatera Selatan, ISSN: 2407-4322, Vol.8 No.4 Desember 2021
- Ramdhani, M., 2021, Metode Penelitian, Cipta Media Nusantara, Surabaya
- Satgani, B. N.;, Angellia, F., 2020, Analisis Sistem Penjualan dengan *Point of Sales* (POS) Berbasis Web di Gudang Kopi Oncak, Jurnal Sistem Informasi Bisnis (JUNSIBI),
- Institut Bisnis dan Informatika Kosgoro, Jakarta Selatan, e-ISSN: 2774-3454, Vol.1 No.1 April 2020
- Setiawan, R., 2021, *Black box* Testing untuk Menguji Perangkat Lunak, <https://www.dicoding.com/blog/black-box-testing>
- Setiawan, R., 2021, *White box* Testing untuk Menguji Perangkat Lunak, <https://www.dicoding.com/blog/white-box-testing>
- Sudaryono, 2015. Metodologi Riset di Bidang TI (Panduan Praktis, Toeri dan Contoh Kasus). Yogyakarta: Andi.
- Widianto, S. R., 2017, Rancang Bangun Aplikasi Telemedika untuk Pasien Diabetes Berbasis *Platform IOS*, Jurnal MULTINETICS, Politeknik Negeri Jakarta, Jakarta Selatan, e-ISSN: 2443-2334, Vol.2 No.1 Mei 2017
- Yudhanto, Y.;, Wijayanto, A., 2017, Mudah Membuat Aplikasi Android dengan Android Studio, PT Elex Media Komputindo, Jakarta
- Zulkifli, M.;, Wibowo, V., 2019, Perancangan Perangkat Lunak *Point of Sale* Berbasis Web pada Toko Banlie Mart Di Kabupaten Bengkayang, Jurnal ENTER, STMIK Pontianak, Pontianak, Vol.2 Agustus 2019