

**RANCANG BANGUN APLIKASI *POINT OF SALES* BERBASIS
ANDROID MENGGUNAKAN *FRAMEWORK* REACT NATIVE**

SKRIPSI

Diajukan untuk Memenuhi Sebagian Syarat Guna Memperoleh
Gelar Sarjana Program Strata 1 (S.1)
Dalam Ilmu Teknologi Informasi



Oleh:

LINTANG AJI YOGA PRATAMA

NIM: 2008096059

**PROGRAM STUDI S-1 TEKNOLOGI INFORMASI
FAKULTAS SAINS DAN TEKNOLOGI
UNIVERSITAS ISLAM NEGERI WALISONGO SEMARANG
2024**

PERNYATAAN KEASLIAN

Yang bertandatangan di bawah ini:

Nama : Lintang Aji Yoga Pratama

NIM : 2008096059

Jurusan : Teknologi Informasi

Menyatakan bahwa skripsi yang berjudul:

RANCANG BANGUN APLIKASI *POINT OF SALES* BERBASIS ANDROID MENGGUNAKAN *FRAMEWORK* REACT NATIVE

Secara keseluruhan adalah hasil penelitian/karya saya sendiri,
kecuali bagian tertentu yang dirujuk sumbernya.

Semarang, 4 Juni 2024

Pembuat Pernyataan,

A handwritten signature in black ink is written over a 10,000 Indonesian postage stamp. The stamp features the Garuda Pancasila emblem and the text 'METRAL TEMPEL' and '10,000'. The serial number '5B4ALX16920117' is visible at the bottom of the stamp.

Lintang Aji Yoga Pratama

NIM: 2008096059

PENGESAHAN

Naskah skripsi berikut ini:

Judul : Rancang Bangun Aplikasi *Point of Sales* Berbasis Android
Menggunakan *Framework* React Native
Penulis : **Lintang Aji Yoga Pratama**
NIM : 2008096059
Jurusan : Teknologi Informasi

Telah diujikan dalam sidang *tugas akhir* oleh Dewan Penguji Fakultas Sains dan Teknologi UIN Walisongo Semarang dan dapat diterima sebagai salah satu syarat memperoleh gelar sarjana dalam program studi Teknologi Informasi.

Semarang, 4 Juni 2024

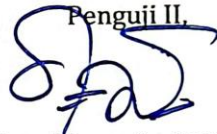
DEWAN PENGUJI

Penguji I,



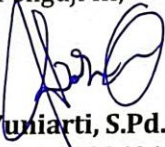
Masy Ari Ulinuha, M.T.
NIP. 198108122011011007

Penguji II,



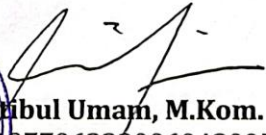
Hery Mustofa, M.Kom.
NIP. 198703172019031007

Penguji III,



Wenty Dwi Yuniarti, S.Pd., M.Kom.
NIP. 197706222006042005

Penguji IV,



Khoirul Umam, M.Kom.
NIP. 197706222006042005

Pembimbing I,



Hery Mustofa, M.Kom.
NIP. 198703172019031007

Pembimbing II,



Adzhal Arwani M., S.Kom. M.Kom.
NIP. 199107032019031006



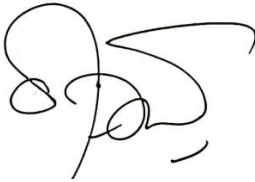
PERSETUJUAN PEMBIMBING

Proposal Skripsi ini telah disetujui oleh Pembimbing untuk dilaksanakan. Disetujui pada:

Hari : Selasa

Tanggal : 5 Desember 2023

Pembimbing I



Hery Mustofa, M.Kom.

NIP. 198703172019031007

Pembimbing II



Adzhal Arwani M., S,Kom M.Kom.

NIP. 199107032019031006

Mengetahui,

Ketua Jurusan Teknologi Informasi



Nur Cahyo Hendro Wibowo, S.T. M.Kom.

NIP. 197312222006041001

LEMBAR PENGESAHAN PROPOSAL SKRIPSI

Judul Skripsi : Rancang Bangun Aplikasi *Point of Sales* Berbasis Android
Menggunakan *Framework* React Native
Nama : Lintang Aji Yoga Pratama
NIM : 2008096059
Program Studi : Teknologi Informasi

Telah diujikan dalam sidang komprehensif oleh Dewan Penguji Fakultas Sains dan Teknologi UIN Walisongo Semarang dan dapat diterima sebagai salah satu syarat memperoleh gelar sarjana dalam program studi Teknologi Informasi.

Semarang, 22 Februari 2024

DEWAN PENGUJI

Penguji I



Masy Ari Ulinuha, M.T.
NIP. 198108122011011007

Penguji II



Hery Mustofa, M.Kom.
NIP. 198703172019031007

Penguji III



Wenty Dwi Yuniarti, S.Pd., M.Kom.
NIP. 197706222006042005

Penguji IV



Khotibul Umam, M.Kom.
NIP. 197706222006042005

NOTA DINAS

Semarang, 17/Mei/2024

Yth. Ketua Program Studi Teknologi Informasi
Fakultas Sains dan Teknologi UIN Walisongo
Semarang

Assalamu'alaikum. wr. wb.

Dengan ini diberitahukan bahwa saya telah melakukan bimbingan, arahan dan koreksi naskah skripsi dengan:

Judul : **Rancang Bangun Aplikasi *Point of Sales* Berbasis Android Menggunakan Framework React Native**

Nama : **Lintang Aji Yoga Pratama**

NIM : 2008096041

Jurusan : Teknologi Informasi

Saya memandang bahwa naskah skripsi tersebut sudah dapat diajukan kepada Fakultas Sains dan Teknologi UIN Walisongo untuk diujikan dalam Sidang Munaqosah.

Wassalamu'alaikum. wr. wb.

Pembimbing I,



Hery Mustofa, M.Kom.
NIP: 19810812 201101 1007

NOTA DINAS

Semarang, 17/Mei/2024

Yth. Ketua Program Studi Teknologi Informasi
Fakultas Sains dan Teknologi UIN Walisongo
Semarang

Assalamu'alaikum. wr. wb.

Dengan ini diberitahukan bahwa saya telah melakukan bimbingan, arahan dan koreksi naskah skripsi dengan:

Judul : **Rancang Bangun Aplikasi *Point of Sales* Berbasis Android Menggunakan *Framework* React Native**

Nama : **Lintang Aji Yoga Pratama**

NIM : 2008096041

Jurusan : Teknologi Informasi

Saya memandang bahwa naskah skripsi tersebut sudah dapat diajukan kepada Fakultas Sains dan Teknologi UIN Walisongo untuk diujikan dalam Sidang Munaqosah.

Wassalamu'alaikum. wr. wb.

Pembimbing II,



Adzhal Arwani Mahfudh, S.Kom., M.Kom.

NIP: 19910703 201903 1 006

RANCANG BANGUN APLIKASI ANDROID *POINT OF SALES* MENGUNAKAN *FRAMEWORK* REACT NATIVE

Oleh:

Lintang Aji Yoga Pratama

NIM. 2008096059

ABSTRAK

Bagi suatu bisnis, pencatatan transaksi, penyimpanan data pelanggan, dan juga perekapan laporan penjualan adalah hal yang penting dan dibutuhkan oleh pemilik bisnis. Pada awalnya, kebutuhan tersebut dapat dilakukan dengan pencatatan manual dengan menggunakan buku. Namun, seiring berjalannya waktu, muncul beberapa masalah yang sangat mungkin dijumpai saat melakukan pencatatan manual, seperti rentan rusaknya media pencatatan, sulitnya mencari suatu transaksi karena banyaknya transaksi yang ada, hingga ketidakamanan media pencatatan. Penelitian ini bertujuan untuk menjawab masalah semua masalah tersebut dengan membangun aplikasi *point of sales* (POS) berbasis Android dengan menggunakan *framework* React Native. Aplikasi ini dapat melakukan penghitungan secara cepat, menyimpan data transaksi dan data pelanggan, hingga mencetak *invoice* atau

setruk bagi para pelanggan dengan menggunakan *thermal printer*. Semua data transaksi dan data pengguna dapat disimpan dengan baik di *database*, sehingga dapat memudahkan pemilik bisnis untuk mencari suatu transaksi secara detail. Keamanan data juga dapat diatasi dengan metode *authentication* dan *authorization* dengan *token validation* pada setiap *Application Programming Interface* (API) dan juga *user interface* (UI) pada aplikasi *mobile* berbasis Android. Aplikasi ini berhasil dibangun menggunakan metode *Software Development Life Cycle* (SDLC) *Waterfall* dengan 5 tahapan yaitu *requirements analysis*, *system & software design*, *implementation*, *integration & testing*, dan *operation & maintenance*. Proses pengujian dilakukan dengan dua metode, yaitu metode *Black Box Testing* dan *User Acceptance Test* (UAT). Pada pengujian *Black Box Testing* didapatkan hasil bahwa fitur-fitur pada aplikasi telah berjalan sesuai dengan *requirement*. Pada pengujian *User Acceptance Test* (UAT) didapatkan skor total yaitu 84%. Dari hasil pengujian tersebut dapat disimpulkan bahwa aplikasi menunjukkan penerimaan yang baik dari pengguna.

Kata Kunci: *Point of Sales*, Android, React Native, *Software Development Life Cycle* (SDLC), *Waterfall*.

KATA PENGANTAR

Alhamdulillah, Puji dan Syukur penulis panjatkan kepada Allah Subhanahu Wata'ala. Zat yang hanya kepada-Nya memohon pertolongan. Alhamdulillah atas segala pertolongan, rahmat, dan kasih sayang-Nya sehingga penulis dapat menyelesaikan skripsi yang berjudul “Rancang Bangun Aplikasi Android *Point of Sales* Menggunakan *Framework* React Native”. Selawat dan salam kepada Rasulullah Shallallahu Alaihi Wasallam yang senantiasa menjadi sumber inspirasi dan teladan terbaik untuk umat manusia.

Penulis menyadari banyak pihak yang memberikan dukungan dan bantuan selama menyelesaikan studi dan tugas akhir ini. Oleh sebab itu, sudah sepantasnya penulis dengan penuh hormat mengucapkan terima kasih dan mendoakan semoga Allah memberikan balasan terbaik kepada:

1. Orang tua yang sangat penulis sayangi, Ibu Yani Setyaningsih dan Almarhum Ayah Heri Kuspartyanto, yang tanpa kenal waktu dan lelah selalu memberikan yang terbaik kepada penulis. Penulis sangat bersyukur dapat menjalani hidup bersama kalian, Ibu dan Ayah yang sangat

suportif yang dapat menjadi tempat berkeluh kesah dan teman bertukar pikiran yang baik.

2. Kedua adik yang penulis banggakan, Raka Alfathir Dwi Kayana dan Rangga El Rafif Adhiyatsa Rayyan. Terima kasih atas segala keceriaan dan kebahagiaan yang telah kalian berikan. Semoga kalian berdua dapat menggapai semua mimpi-mimpi kalian dan menjadi anak yang berbakti kepada Ibu dan Ayah.
3. Kakek dan Nenek penulis, Mamak Sundusiyah dan Almarhum Bapak Tukidjo Harjono, yang telah memberikan kenangan masa kecil yang menyenangkan dan dukungan yang sangat penulis butuhkan.
4. Dinda Khansa Berlian. Terima kasih atas segala waktu dan usaha yang telah diberikan kepada penulis.
5. Bapak Dr. Khothibul Umam, S.T., M. Kom., selaku Kepala Program Studi Teknologi Universitas Islam Negeri Walisongo Semarang.
6. Bapak Hery Mustofa, M.Kom. dan Bapak Adzhal Arwani Mahfudh, S.Kom., M. Kom, selaku pembimbing penulis dalam menyelesaikan skripsi.
7. Dosen-dosen Program Studi Teknologi Informasi UIN Walisongo Semarang. Terimakasih penulis juga haturkan untuk semua pihak yang telah membantu peneliti dalam

menyelesaikan skripsi ini yang tidak dapat peneliti sebutkan satu persatu.

8. Seluruh teman-teman yang penulis banggakan. Muhammad Abdur Rofi Maulidin, Ivan Rizky Saputra, Krisnanda Muhammad Iqbal, Alfian Hidayat, Faris Nur Rahman, Evan Puji Setyawan, Ibnu Alfian Rachmat, Aziz Maulana Rafiansyah, Rabbani Dwi Putra, M. Ali Ridho Fathoni, Wahyu Rizqy Saputra, Amayuda Safei, Akbar Sidiq, Lathif Milhas, Ahlis Aulia Rahman, dan semua teman-teman Angkatan 2020 Teknologi Informasi UIN Walisongo Semarang yang telah mendorong dan mendukung penulis untuk selalu lebih baik dari hari ke hari. Semoga kalian dapat mewujudkan angan dan impian kalian.

Akhir kata penulis menyadari bahwa tidak ada yang sempurna, penulis masih melakukan kesalahan dalam penyusunan skripsi. Oleh sebab itu, penulis meminta maaf yang sedalam-dalamnya atas kesalahan yang dilakukan penulis.

Penulis berharap semoga skripsi ini dapat bermanfaat bagi pembaca dan dapat dijadikan referensi demi pengembangan ke arah yang lebih baik.

Semarang, 17 Mei 2024
Lintang Aji Yoga Pratama

DAFTAR ISI

PERNYATAAN KEASLIAN	iii
PENGESAHAN	v
PERSETUJUAN PEMBIMBING.....	vii
LEMBAR PENGESAHAN PROPOSAL SKRIPSI.....	ix
NOTA DINAS.....	xii
ABSTRAK.....	xv
KATA PENGANTAR.....	xvii
DAFTAR ISI	xxi
DAFTAR GAMBAR.....	xxiii
DAFTAR TABEL.....	xxv
BAB I.....	2
A. Latar Belakang Masalah.....	2
B. Rumusan Masalah	6
C. Batasan Masalah	6
D. Tujuan Penelitian.....	7
E. Manfaat Penelitian	8
BAB II.....	10
A. Kajian Teori	10
1. Sistem <i>Point of Sales</i> (POS)	10
2. React Native	13
3. Express.js.....	16
4. NoSQL.....	16

5. MongoDB.....	18
B. Kajian Penelitian yang Relevan.....	19
BAB III.....	26
A. Metode Pengembangan Sistem	26
1. <i>Requirement Definition</i>	27
2. <i>System and Software Design</i>	33
3. <i>Implementation and Unit Testing</i>	38
4. <i>Integration and System Testing</i>	40
5. <i>Operation and Maintenance</i>	44
BAB IV	46
A. <i>Requirement Definition</i>	46
B. <i>System and Software Design</i>	47
C. <i>Implementation and Unit Testing</i>	50
D. <i>System Testing</i>	63
E. <i>Operation and Maintenance</i>	67
BAB V	68
A. Kesimpulan.....	68
B. Saran.....	69
DAFTAR PUSTAKA	70
LAMPIRAN	74
RIWAYAT HIDUP	86

DAFTAR GAMBAR

Gambar 2.1 Alur Pencetakan Setruk Pembelian	13
Gambar 3.1 Alur Metode Pengembangan Aplikasi Waterfall	27
Gambar 3.2 Data Flow Diagram Level 0	34
Gambar 3.3 Data Flow Diagram Level 1 Proses 5.....	36
Gambar 3.4 Entity Relationship Diagram Aplikasi POS	37
Gambar 4.1 Proses Pembuatan <i>Database</i>	48
Gambar 4.2 Pengisian Nama <i>Database</i>	48
Gambar 4.3 Skrip <i>Database Migration</i>	49
Gambar 4.4 Perintah Instalasi <i>Package Express.js</i>	50
Gambar 4.5 Daftar <i>Package</i> Pada <i>Web Service</i>	50
Gambar 4.6 Proses Routing Pada <i>Web Service</i>	52
Gambar 4.7 Proses Pembuatan <i>Handler</i> Pada <i>Web Service</i>	53
Gambar 4.8 Proses Pembuatan <i>Service Query</i>	54
Gambar 4.9 Inisialisasi <i>Project</i> React Native	55
Gambar 4.10 Daftar <i>Package</i> yang Digunakan Pada Aplikasi React Native.....	56
Gambar 4.11 Instalasi <i>Package</i> yang Dibutuhkan	57
Gambar 4.12 Project Structure Pada Folder <i>screens</i>	57
Gambar 4.13 Skrip <i>User Interface</i> Pada Halaman Produk.....	58
Gambar 4.14 Halaman Tambah Produk	59
Gambar 4.15 Integrasi <i>Web Service</i> dengan React Native	60
Gambar 4.16 Skrip <i>Trigger handleSubmit</i>	60

Gambar 4.17 Tombol Simpan Pada Halaman Tambah Produk	61
Gambar 4.18 Konfigurasi <i>Unit Testing</i> Pada Postman	62

DAFTAR TABEL

Tabel 2.1 Daftar Kajian Penelitian yang Relevan	20
Tabel 3.1 Daftar Menu pada UMKM Kedai A7	29
Tabel 3.2 Skenario <i>Black Box Testing</i>	41
Tabel 3.3 Daftar Pertanyaan Metode UAT	42
Tabel 3.4 Bobot Nilai <i>Mean Option Score</i> (MOS)	43
Tabel 3.5 Skala Likert.....	44
Tabel 4.1 Hasil <i>Black Box Testing</i> Pada Kedai A7	62
Tabel 4.2 Hasil <i>User Acceptance Testing</i> Pada Kedai A7	64
Tabel 4.3 Daftar Masukan Setelah Dilakukan <i>System Testing</i>	65

BAB I

PENDAHULUAN

A. Latar Belakang Masalah

Seiring perkembangan teknologi yang semakin pesat dari tahun ke tahun, perkembangan ilmu pengetahuan, dan digitalisasi yang masif mendorong semua sektor bisnis untuk terus beradaptasi dan berkembang dengan cepat. Perkembangan teknologi yang pesat telah membuat informasi mudah diakses melalui perangkat *mobile*, terutama *smartphone*. Perangkat *mobile*, termasuk *smartphone*, telah menjadi penting dalam kehidupan sehari-hari kita. Salah satunya adalah penggunaan *smartphone* dalam proses transaksi jual beli (Restiawan & Ramos, 2021).

Saat ini, pemanfaatan teknologi untuk bisnis terus berkembang dari bisnis UMKM (Usaha Mikro Kecil dan Menengah) yang memiliki lingkup kecil hingga perusahaan besar yang memiliki lingkup global. Hal ini disebabkan karena teknologi dapat membantu menyelesaikan permasalahan-permasalahan dalam bisnis dengan cepat dan efisien. Namun, dari sekitar 6,5 juta lebih UMKM, hanya 17,25 juta atau kurang lebih 26,5% UMKM yang

terhubung ke dalam ekosistem digital (Bank Indonesia, 2022). Bank Indonesia merasa bahwa Angka tersebut perlu ditingkatkan, seiring perkembangan akseptasi dan preferensi masyarakat dalam berbelanja daring yang semakin masif.

Pusat jajanan serba ada (Pujasera) Ngaliyan merupakan salah satu Pujasera yang terletak di Kecamatan Ngaliyan, Kota Semarang, Jawa Tengah. Pujasera Ngaliyan terletak di sebelah utara kantor Kecamatan Ngaliyan. Menurut pengurus Pujasera Ngaliyan, Mas Radit, Pujasera ini memiliki 22 *stand* UMKM dengan berbagai macam jajanan mulai dari makanan, minuman, hingga makanan ringan.

Sebagian besar proses transaksi UMKM di Pujasera Ngaliyan sudah mendukung pembayaran digital seperti Gopay, ShoppePay, hingga QRIS (Quick Response Code Indonesian Standard). Namun, untuk pencatatan transaksi penjualan dan list pesanan masih menggunakan pencatatan manual atau *paper-based*. Saat kondisi pelanggan sedang sepi, pencatatan transaksi masih dapat ditunjang dengan pencatatan manual, namun jika kondisi pelanggan sedang ramai, muncul suatu masalah yaitu tidak sempatnya untuk mencatat transaksi atau tidak jelasnya

tulisan karena terburu-buru saat melakukan pencatatan transaksi. Selain itu, tingginya risiko kerusakan atau kehilangan catatan transaksi, sulitnya mencari transaksi tertentu, dan ketidakamanan data yang disimpan juga menjadi permasalahan tambahan bagi pelaku usaha.

Di dalam Al-Qur'an dijelaskan bahwa Allah SWT menghalalkan manusia untuk melakukan jual beli sebagai salah satu cara untuk manusia memenuhi kebutuhan hidupnya. Di sisi lain, Allah SWT pun melarang riba karena transaksi riba dapat merugikan salah satu pihak yang melakukan transaksi. Hal-hal mengenai jual beli dan riba disebutkan dalam Surah Al-Baqarah ayat 275:

الَّذِينَ يَأْكُلُونَ الرِّبَا لَا يَقُومُونَ إِلَّا كَمَا يَقُومُ الَّذِي يَخْبِطُهُ الشَّيْطَانُ
مِنَ الْمَسِّ ذَلِكَ بِأَنَّهُمْ قَالُوا إِنَّمَا الْبَيْعُ مِثْلُ الرِّبَا وَأَحَلَّ اللَّهُ الْبَيْعَ وَحَرَّمَ
الرِّبَا فَمَنْ جَاءَهُ مَوْعِظَةٌ مِنْ رَبِّهِ فَانْتَهَى فَلَهُ مَا سَلَفَ وَأَمْرُهُ إِلَى اللَّهِ
وَمَنْ عَادَ فَأُولَئِكَ أَصْحَابُ النَّارِ هُمْ فِيهَا خَالِدُونَ

Artinya:

"Orang-orang yang makan (mengambil) riba tidak dapat berdiri melainkan seperti berdirinya orang yang kemasukan syaitan lantaran (tekanan) penyakit gila. Keadaan mereka yang demikian itu, adalah disebabkan mereka berkata (berpendapat), sesungguhnya jual beli itu sama dengan riba, padahal Allah telah menghalalkan jual beli dan mengharamkan riba."

Berdasarkan masalah yang ada pada Pujasera Ngaliyan mengenai pencatatan transaksi, penelitian ini menawarkan solusi untuk mengatasi masalah tersebut dengan merancang sebuah aplikasi *Point of Sales* (POS) berbasis Android dengan menggunakan *framework* React Native yang berbasis *cloud* atau internet. Berdasarkan data dari Hospitality Technology's 2018 Point of Sales (POS) Software Trend Reports, sebesar 61% *merchants* ingin menggunakan sistem POS berbasis *cloud* (Forbes, 2018). Aplikasi ini ditujukan untuk menjadi alat bantu bagi pelaku UMKM di Pujasera Ngaliyan untuk dapat mengelola kasir dan laporan transaksi dengan lebih mudah, efisien, dan aman. React Native merupakan *user interface* (UI) *framework* berbasis JavaScript yang digunakan untuk mengembangkan aplikasi *mobile* di dua sistem operasi yang berbeda, yaitu Android dan iOS. React Native merupakan salah satu proyek *open-source* yang pertama kali diluncurkan oleh Facebook pada tahun 2015. React Native telah dipakai oleh perusahaan-perusahaan besar seperti Facebook, Microsoft, Discord, Puma, hingga Tesla (React Native Documentation, 2023).

Menurut pemilik UMKM di Pujasera Ngaliyan, pencatatan secara digital akan mengurangi kesalahan-

kesalahan yang disebabkan oleh kelalaian sumber daya manusia (*human error*). Selain itu, dengan adanya digitalisasi pencatatan, akan membawa pengaruh positif pada sisi efisiensi pencatatan transaksi dan keamanan data.

Berdasarkan permasalahan yang telah dijabarkan, maka penelitian ini akan berfokus pada perancangan dan pengembangan aplikasi *Point of Sales* (POS) berbasis Android dengan menggunakan *framework* React Native kebutuhan bisnis berupa pengelolaan kasir dan pencatatan laporan transaksi bagi pemilik UMKM di Pujasera Ngaliyan.

B. Rumusan Masalah

Berdasarkan latar belakang masalah yang telah dipaparkan, maka rumusan masalah adalah sebagai berikut:

1. Bagaimana merancang aplikasi *Point of Sales* (POS) berbasis Android dengan menggunakan *framework* React Native di Pujasera Kec. Ngaliyan, Kota Semarang?
2. Bagaimana penerimaan pengguna dalam penerapan aplikasi *Point of Sales* (POS) berbasis Android dengan menggunakan *framework* React Native?

C. Batasan Masalah

Agar pembuatan aplikasi ini lebih terfokus pada topik yang diambil, maka masalah dibatasi pada hal-hal sebagai berikut:

1. Data mengenai pemakaian aplikasi *Point of Sales* (POS) dan kebutuhan fitur-fitur yang ada di dalamnya diambil di Pujasera Kec. Ngaliyan, Kota Semarang, Provinsi Jawa Tengah.
2. Metode pengembangan perangkat lunak yang digunakan pada penelitian ini adalah *Software Development Life Cycle* (SDLC) model *Waterfall*.
3. Penelitian ini tidak mendukung fitur *payment gateway* untuk pembayaran produk secara *online*.
4. Penelitian ini berfokus pada pembuatan aplikasi *Point of Sales* untuk pemilik bisnis.

D. Tujuan Penelitian

Adapun tujuan penelitian berdasarkan rumusan masalah adalah sebagai berikut;

1. Merancang dan membangun aplikasi *Point of Sales* (POS) berbasis Android dengan menggunakan *framework* React Native di Pujasera Kec. Ngaliyan, Kota Semarang.

2. Menguji penerimaan pengguna dalam penerapan aplikasi *Point of Sales* (POS) berbasis Android dengan menggunakan *framework* React Native.

E. Manfaat Penelitian

Manfaat yang didapat dari dirancangnya aplikasi *Point of Sales* (POS) ini adalah sebagai berikut:

1. Manfaat Teoritis

Penelitian ini diharapkan dapat menambah koleksi pustaka akademik dengan topik mengenai aplikasi *Point of Sales* (POS) berbasis Android dengan menggunakan *framework* React Native dan hasil penelitian dapat dijadikan bahan referensi oleh peneliti berikutnya yang akan membahas mengenai aplikasi *Point of Sales* berbasis Android dengan menggunakan *framework* React Native.

2. Manfaat Praktis

Penelitian ini diharapkan mampu memberikan kemudahan untuk UMKM di Pujasera Kec. Ngaliyan, Kota Semarang untuk meningkatkan efisiensi dalam proses transaksi dan pencatatan laporan penjualan.

BAB II

LANDASAN PUSTAKA

A. Kajian Teori

1. Sistem *Point of Sales* (POS)

Sistem *Point of Sales* (POS) adalah sebuah sistem aplikasi yang digunakan dalam bisnis untuk mengelola dan memproses transaksi penjualan dan pembelian barang dagang. Sistem ini mencakup pengolahan data transaksi pembelian, penjualan eceran, retur pembelian, dan pelaporan transaksi (Putra dkk., 2023).

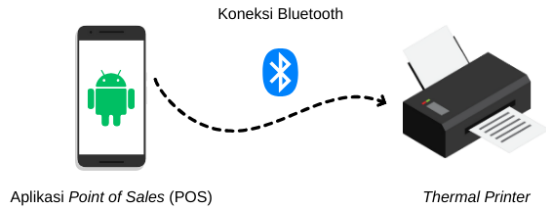
Dalam sistem POS, terdapat perangkat keras (hardware) dan perangkat lunak (software) yang digunakan untuk melakukan proses transaksi. Perangkat lunak POS merupakan komponen utama yang menentukan jalannya proses transaksi, sedangkan perangkat keras POS digunakan untuk membantu proses pembayaran dan mencetak setruk belanja. Sistem POS memiliki keunggulan dalam mempercepat proses transaksi, pengelolaan kategori barang, pengelolaan data *supplier*, manajemen pengguna, serta pembentukan laporan yang dapat membantu dalam pengembangan bisnis (Putra dkk., 2023).

Pada sistem POS, terdapat suatu fitur yaitu pencetakan setruk pembelian dengan menggunakan suatu perangkat keras *thermal printer*. *Thermal printer* adalah jenis *printer* yang menggunakan kepala cetak termal panas yang ditekan pada media untuk menghasilkan teks atau gambar hitam. Kepala cetak yang dipanaskan akan mengaktifkan kertas khusus yang diolah secara kimia (kertas termal) atau mentransfer tinta padat dari pita dan mengikatnya ke media, tergantung pada jenis printer termal. *Thermal printer* umumnya digunakan dalam memproduksi label, setruk, kuitansi, hingga *barcode* (CDW, 2022).

Sistem POS dapat berjalan pada berbagai macam sistem operasi, seperti Windows, MacOS, Linux pada perangkat *desktop*, hingga sistem operasi yang ada pada perangkat *mobile* seperti Android dan iOS. Menurut Cambridge Dictionary (2024), Perangkat *desktop* adalah komputer yang dirancang untuk ditempatkan di atas meja atau permukaan datar lainnya, biasanya digunakan di kantor, rumah, atau lingkungan bisnis. Sedangkan perangkat *mobile* merupakan perangkat elektronik yang dirancang untuk digunakan saat bergerak atau dalam perjalanan di mana pun pengguna berada.

Pada penelitian ini, perangkat *mobile* dipilih karena mobilitas perangkatnya yang tinggi karena ruang kosong *stand* UMKM di Pujasera Ngaliyan tidak memungkinkan untuk dipakainya perangkat *desktop*. Lalu, penulis membangun aplikasi POS pada sistem operasi Android karena menurut GlobalStats (2024), dari pasar perangkat lunak di dunia, sistem operasi Android digunakan oleh 69.94% pengguna perangkat *mobile* di seluruh dunia, disusul oleh iOS pada 29.32%, dan sistem operasi lainnya. Dapat disimpulkan bahwa pada tahun penelitian ini dibuat, sistem operasi Android masih menjadi sistem operasi yang paling populer di dunia.

Perpindahan data dari sistem POS ke perangkat *thermal printer* dilakukan dengan koneksi Bluetooth. Bluetooth adalah teknologi nirkabel yang digunakan untuk mentransfer data dan informasi antar perangkat elektronik dalam jarak pendek. Teknologi ini memungkinkan perangkat seperti smartphone, laptop, headset, speaker, printer, dan banyak perangkat lainnya untuk berkomunikasi dan bertukar data satu sama lain tanpa perlu kabel fisik (Bluetooth, 2023).



Gambar 2.1 Alur Pencetakan Setruk Pembelian

Pada penelitian ini, teknologi Bluetooth digunakan untuk mengkoneksikan perangkat *mobile* yang menggunakan Android API dengan perangkat *thermal printer* untuk mencetak setruk yang telah dihasilkan saat transaksi.

2. React Native

Menurut (Schmidt dkk., 2004), *application framework* merupakan sekumpulan artefak perangkat lunak yang terintegrasi (seperti kelas, objek, dan komponen) yang berkolaborasi untuk menyediakan arsitektur yang dapat digunakan kembali untuk rangkaian aplikasi terkait.

Dapat disimpulkan bahwa *framework* merupakan suatu kesatuan perangkat lunak dengan arsitektur tertentu yang dapat menjadi pondasi untuk mengembangkan suatu aplikasi perangkat lunak. *Framework* dapat mempermudah *developer* untuk mempersingkat proses pengembangan aplikasi perangkat lunak dengan menyediakan kode yang dapat digunakan kembali (*reusable code*), *library*, dan *best*

practices. Untuk membangun aplikasi Android, terdapat beberapa *framework* yang dapat digunakan, seperti Android Jetpack, Flutter, dan React Native.

Android Jetpack adalah kumpulan dari berbagai komponen, alat, dan panduan yang membantu pengembang dalam membangun aplikasi Android yang berkualitas dan efisien. Jetpack terdiri dari beberapa bagian seperti Room (untuk database), ViewModel (untuk mengelola data UI), LiveData (untuk membuat data yang dapat diamati).

Flutter adalah *framework* pengembangan aplikasi mobile yang dikembangkan oleh Google. Dengan menggunakan bahasa pemrograman Dart, Flutter memungkinkan pengembang untuk membuat aplikasi Android (dan juga iOS) dengan UI yang konsisten dan menarik. Flutter menggunakan pendekatan "Widget" yang kuat untuk membangun antarmuka pengguna.

React Native merupakan *user interface* (UI) *framework* berbasis JavaScript yang digunakan untuk mengembangkan aplikasi mobile di dua sistem operasi yang berbeda, yaitu Android dan iOS. React Native merupakan salah satu proyek *open-source* yang pertama kali diluncurkan oleh Facebook pada tahun 2015. React Native telah dipakai oleh perusahaan-perusahaan besar seperti Facebook, Microsoft,

Discord, Puma, hingga Tesla (React Native Documentation, 2023).

React Native menerjemahkan *markup language* menjadi elemen UI *mobile-native*. Selain itu, React Native bekerja secara terpisah dari thread UI utama, sehingga aplikasi dapat mempertahankan kinerja tinggi. Siklus pembaruan di React Native sama dengan di React, yaitu ketika *props* atau *state* berubah, React Native memuat ulang tampilannya.

Pada penelitian ini, React Native dipilih oleh penulis karena *framework* ini memiliki beberapa keunggulan dibandingkan dengan beberapa *framework* untuk membangun aplikasi Android lainnya. Pengembangan yang cepat dan efisien dapat dilakukan dengan React Native karena penggunaannya yang didasarkan pada JavaScript, yang lebih cepat daripada bahasa seperti Java atau Objective-C/Swift. Selain itu, kemampuan untuk menggunakan sebagian besar kode yang sama untuk platform Android dan iOS mengurangi kerja ganda dan mempercepat waktu pengembangan. Selain itu, React Native dikembangkan oleh Facebook (Meta) dan memiliki dukungan dari perusahaan tersebut serta kontribusi dari komunitas pengembang yang luas. Keberadaan dukungan dari perusahaan besar ini memberikan jaminan atas

kelangsungan dan perkembangan React Native di masa mendatang.

3. Express.js

Express merupakan *framework* untuk membangun suatu aplikasi web yang berjalan di *runtime* Node.js. *Framework* ini adalah *open-source framework* yang dibuat oleh TJ Holowaychuk pada tahun 2010. Express berfungsi untuk mengatur fungsionalitas *back-end* suatu *website* seperti autentikasi, koneksi *database*, *routing*, *middleware*, HTTP *request*, hingga *error handling* (Express.js Documentation, 2023).

4. NoSQL

Menurut (Elmasri & Natavhe, 2016), NoSQL atau *Not Only SQL* adalah basis data terdistribusi atau sistem penyimpanan terdistribusi, dengan fokus pada penyimpanan data semi terstruktur, kinerja tinggi, ketersediaan, replikasi data, dan skalabilitas dibandingkan dengan penekanan pada konsistensi data langsung, dan bahasa kueri yang kuat.

Menurut (AWS, 2023), NoSQL adalah teknologi basis data yang menyimpan data dalam skema fleksibel yang dapat diskalakan dengan mudah. NoSQL seringkali disebut sebagai basis data nonrelasional. Basis data NoSQL dibuat

husus untuk model data nonrelasional dan memiliki skema fleksibel untuk membangun aplikasi modern. Basis data NoSQL dikenal secara luas karena kemudahan pengembangan, fungsionalitas, dan performa dalam skala besar. Manfaat basis data NoSQL tercantum di bawah ini:

a. Skalabilitas

Basis data NoSQL umumnya didesain untuk menskalakan suatu aplikasi menggunakan kluster perangkat keras yang terdistribusi dibandingkan harus menaikkan skala dengan menambah server yang mahal dan andal.

b. Fleksibilitas

Basis data NoSQL umumnya menyediakan skema fleksibel yang memungkinkan pengembangan yang lebih cepat dan lebih berulang. Model data fleksibel membuat basis data NoSQL ideal untuk data yang semi terstruktur dan tidak terstruktur.

c. Performa Tinggi

Basis data NoSQL dioptimalkan untuk model data tertentu dan pola akses. Hal ini memungkinkan performa yang lebih tinggi daripada dibandingkan dengan fungsionalitas serupa dengan basis data relasional.

d. Fungsional

Basis data NoSQL menyediakan *application programming interface* (API) dan tipe data fungsional yang dibuat secara khusus untuk setiap model datanya.

5. MongoDB

MongoDB adalah sistem basis data berorientasi dokumen yang termasuk dalam kategori basis data NoSQL. Basis data NoSQL dirancang untuk mengelola, menyimpan, dan mengambil data yang bersifat semi-struktural atau tidak berstruktur dengan cara yang lebih fleksibel dibandingkan dengan sistem basis data relasional tradisional. Karakteristik utama MongoDB meliputi:

a. Dokumen

MongoDB menyimpan data dalam bentuk dokumen BSON (Binary JSON), yang memungkinkan penyimpanan data yang sangat fleksibel. Dokumen ini dapat berbeda dalam struktur, yang memudahkan pengembangan aplikasi yang membutuhkan evolusi skema data.

b. Skema Dinamis

MongoDB memiliki skema dinamis, yang berarti pengembang aplikasi dapat memasukkan data dengan skema yang berbeda ke dalam koleksi yang sama. Ini membuat MongoDB sangat sesuai untuk aplikasi yang berubah dan berkembang seiring waktu.

c. Skalabilitas Horizontal

MongoDB dirancang untuk skala horizontal, yang memungkinkan pengembang aplikasi untuk dengan mudah menambahkan *server* dan kapasitas penyimpanan saat aplikasi berkembang.

d. Indeks

MongoDB mendukung indeks yang dapat digunakan untuk mempercepat pencarian dan kueri data.

e. Replikasi dan Keandalan

MongoDB mendukung replikasi data untuk keandalan. Pengembang aplikasi dapat mengkonfigurasi *server* replika yang menyediakan salinan data yang sama, sehingga jika satu *server* mengalami gangguan, yang lainnya dapat menggantikannya.

f. Pemrosesan Paralel

MongoDB mendukung pemrosesan paralel untuk mempercepat kueri dan pengolahan data.

B. Kajian Penelitian yang Relevan

Tabel 2.1 di bawah ini merupakan beberapa penelitian terdahulu yang relevan berdasarkan dengan kesamaan topik atau penggunaan tools/teknologi yang diambil penulis dalam penelitian ini.

Tabel 2.1 Daftar Kajian Penelitian yang Relevan1

No.	Judul	Penulis	Hasil Penelitian
1.	Rancang Bangun Aplikasi <i>Point of Sales</i> Berbasis Android pada Sistem Basis Data	Vito Juliano (2021)	Penelitian ini telah berhasil Membangun sistem POS berbasis Android untuk PT Petra Surya Mandiri. Sistem ini dibangun menggunakan Android dengan bahasa pemrograman Java, Lumen PHP <i>framework</i> sebagai API, dan basis data MySQL sebagai tempat penyimpanan sekaligus logika pemrograman dan bisnis pada aplikasi.
2.	Perancangan Aplikasi <i>Point of Sales</i> (POS) Berbasis Android (Studi Kasus: Warkop	Restiawan Yoga Pratama, Ramos Somya (2021)	Jurnal penlitian ini menghasilkan aplikasi POS berbasis Android denga metode pengembangan aplikasi Spiral. Aplikasi dibuat dengan

	Vape Salatiga)		menggunakan dua <i>database</i> yaitu Firebase dan SQLite yang berbasis SQL server.
3.	Rancang Bangun Aplikasi <i>Point of Sales</i> (POS) dan Pengelolaan Stok Berbasis Web pada CV Daya Logam 2 Menggunakan Framework Laravel	Philip Shane Hardhianto Budiono (2023)	Hasil dari penelitian ini adalah sebuah aplikasi POS berbasis <i>website</i> yang dikembangkan menggunakan metode pengembangan aplikasi RAD (Rapid Application Development) dengan menggunakan <i>framework</i> Laravel.
4.	Rancang Bangun Aplikasi <i>Point of Sales</i> Berbasis Android Pada Mebel Selaparang	Muhammad Yahya Jaber, Misbahuddin, dan L. A. Syamsul Irfan Akbar. (2021)	Jurnal penelitian ini menghasilkan suatu aplikasi POS berbasis Android untuk usaha mebel Selaparang yang dibangun menggunakan Java. Berdasarkan pengujian <i>Mean Opinion Score</i>

			(MOS), aplikasi ini dinilai cukup memuaskan yang menghasilkan nilai 3.642 dari 5 dengan nilai persentase keberhasilan program sebesar 72.83%.
5.	Penerapan Teknologi <i>Point of Sales</i> (POS) Sebagai Media Informasi Penjualan Ikan Hias Berbasis Web Studi Kasus: King Koi Group.	Amanah Cetagati, Ade Surahman, Adi Sucipto. (2021)	Hasil penelitian dari jurnal ini merupakan suatu aplikasi POS berbasis <i>website</i> Bahasa pemrograman yang digunakan dalam pengembangan sistem adalah PHP, Visual Studio Code sebagai <i>editor</i> penulisan kode. Sedangkan disisi pengolahan database menggunakan MySQL. Pengujian yang dilakukan terhadap aplikasi memiliki persentase nilai rata-rata 97.74% berdasarkan

			aspek <i>suitability</i> , <i>accuracy</i> , <i>interoperability</i> , dan <i>security</i> .
--	--	--	---

Peneliti menggunakan acuan referensi yang telah disajikan pada tabel 2.1 di atas sebagai dasar perancangan aplikasi. Penelitian yang akan dilakukan memiliki perbedaan dari penelitian sebelumnya. Pada penelitian sebelumnya, aplikasi *Point of Sales* (POS) yang dikembangkan menghasilkan aplikasi *website* dan aplikasi Android dengan menggunakan bahasa pemrograman Java yang didukung oleh *database* MySQL dan Firebase. Sedangkan aplikasi yang akan dibuat merupakan sistem *Point of Sales* (POS) yang berbasis Android dengan menggunakan *framework* React Native yang dapat mengkompilasi kode menjadi aplikasi Android dan iOS (React Native Documentation, 2023). Lalu, aplikasi yang akan dibuat akan menggunakan *database* NoSQL MongoDB yang memiliki skema *database* yang dinamis sehingga memudahkan dalam segi skalabilitas aplikasi. Selain itu, aplikasi ini memiliki koneksi dengan *thermal printer* melalui Bluetooth yang akan mencetak struk untuk pelanggan saat transaksi telah selesai. Metode pengembangan sistem pada penelitian ini menggunakan

software development life cycle (SDLC) waterfall dengan sistem pengujian menggunakan metode *user acceptance test (UAT)*.

BAB III

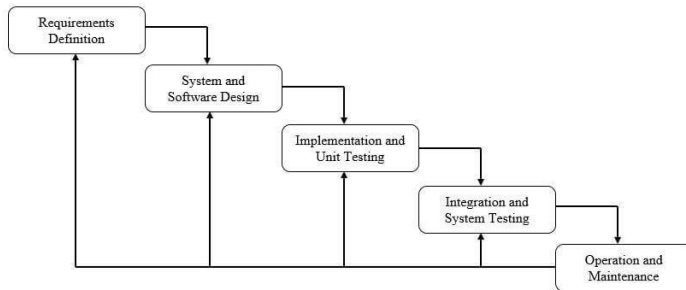
METODE PENELITIAN

A. Metode Pengembangan Sistem

Metode Waterfall adalah salah satu metode pengembangan perangkat lunak yang digunakan dalam perancangan suatu aplikasi. Metode ini merupakan metode pengembangan perangkat lunak yang linear dan terstruktur, yang setiap tahapannya harus selesai terlebih dahulu sebelum memasuki tahapan berikutnya. Tahapan pada metode Waterfall ini meliputi analisis kebutuhan, perancangan, implementasi, pengujian, dan pemeliharaan (Ariandi dkk., 2022).

Metode ini diambil oleh peneliti karena kebutuhan-kebutuhan dalam pembuatan sistem *Point of Sales* (POS) berbasis Android ini dapat ditunjang dengan metode Waterfall. Metode ini merupakan metode yang terstruktur dan terorganisir karena setiap fase memiliki tanggung jawabnya sendiri, sehingga memungkinkan pemahaman yang jelas tentang kemajuan proyek. Metode ini juga memudahkan pengembang untuk dapat memahami kebutuhan sistem di awal proses pengembangan, sehingga

memperkecil kemungkinan untuk melakukan perubahan besar di tengah proses pengembangan.



Gambar 3.1 Alur Metode Pengembangan Aplikasi Waterfall

Tahapan-tahapan pengembangan perangkat lunak pada penelitian ini dengan menggunakan metode Waterfall melibatkan langkah-langkah berikut:

1. *Requirement Definition*

Pada tahap ini, kebutuhan dan prasyarat sistem yang diinginkan dikumpulkan dari pengguna atau klien. Persyaratan ini dapat berupa fungsionalitas, antarmuka pengguna, dan fitur-fitur yang ada pada sistem yang diharapkan oleh pengguna atau klien. Pada tahap *requirement definition* ini, yang dilakukan adalah:

- a. Pengumpulan Data Primer

- 1) Observasi

Metode pengamatan langsung atau observasi dilakukan di lingkungan Pujasera Kecamatan Ngaliyan,

tepatnya pada *stand* Kedai A7 untuk memperoleh dan mengumpulkan data yang dibutuhkan. Berdasarkan observasi yang dilakukan diketahui bahwa UMKM yang ada di Pujasera Kecamatan Ngaliyan belum memiliki suatu sistem *Point of Sales* untuk mencatat transaksi-transaksi yang dilakukan.

2) Wawancara

Untuk memperoleh data yang lebih detail, dilakukan proses wawancara dengan pemilik *stand* Kedai A7, yaitu Ibu Ria dengan tujuan untuk memperoleh data-data yang dibutuhkan. Data-data ini nantinya akan dipakai sebagai dasar pembuatan fitur-fitur sesuai dengan apa yang dibutuhkan oleh pemilik usaha. Pada proses wawancara ini, diperoleh data penguat bahwa pada Kedai A7 Pujasera Kecamatan Ngaliyan masih menggunakan pencatatan konvensional untuk mencatat pemasukan dan pengeluaran berdasarkan transaksi yang dilakukan dan belum memiliki suatu sistem *Point of Sales* yang dapat membantu proses tersebut menjadi lebih efisien dan aman. Para pemilik UMKM juga mengaku bahwa mereka telah mendengar dan mengetahui adanya aplikasi POS atau kasir yang berbasis *online* namun belum pernah menggunakannya. Mereka berharap aplikasi ini dapat

mempermudah proses transaksi dan pencatatan laporan keuangan yang sebelumnya memiliki beberapa masalah jika dilakukan dengan pencatatan manual.

Selain itu, diambil pula data dari satu sampel menu produk UMKM di Pujasera Ngaliyan, yaitu Kedai A7, UMKM yang memiliki konsep restoran Jepang. Dari proses wawancara yang dilakukan, didapatkan data menu makanan dan minuman beserta harganya yang nantinya akan digunakan pada proses pembangunan aplikasi POS. Data menu yang digunakan ditunjukkan pada tabel di bawah ini.

Tabel 3.1 Daftar Menu pada UMKM Kedai A7

Kategori	Menu	Harga
Makanan	<i>Chicken Katsu + Nasi + Es Teh</i>	Rp14.000
	<i>Chicken Katsu + Kentang + Es Teh</i>	Rp16.000
	<i>Chicken Katsu (Matah) + Nasi + Es Teh</i>	Rp15.000
	<i>Chicken Katsu + Kentang + Es Teh</i>	Rp17.000
	<i>Chicken Katsu</i>	Rp10.000
	<i>Chicken Katsu (Matah)</i>	Rp11.000
	<i>Chicken Steak + Nasi + Es Teh</i>	Rp13.000
	<i>Chicken Steak + Kentang + Es Teh</i>	Rp15.000
	<i>Chicken Steak</i>	Rp9.000

	Ayam Bakar Bumbu Bali + Nasi + Es Teh	Rp13.000
	Ayam Bakar Bumbu Bali	Rp9.000
	Kentang Goreng	Rp7.000
	Nasi Putih	Rp3.000
Minuman	Es Teh Panas/Panas Manis	Rp3.000
	Es Teh Panas/Panas Tawar	Rp2.000
	Es Jeruk/Panas	Rp4.000
	Es Jeruk Selasih	Rp5.000
	<i>Milkshake</i> Coklat	Rp8.000
	<i>Milkshake</i> Stroberi	Rp8.000
	<i>Milkshake</i> Vanilla	Rp8.000

Dengan wawancara yang dilakukan, peneliti dapat mengetahui masalah yang dialami oleh pemilik UMKM sehingga proses penentuan fitur-fitur yang ada pada aplikasi *Point of Sales* (POS) dapat didefinisikan dengan baik.

b. Pengumpulan Data Sekunder

Metode yang dilakukan dalam pengumpulan data sekunder pada penelitian ini adalah metode studi literatur. Studi literatur adalah suatu metode penyelesaian masalah melalui tulisan-tulisan terdahulu. Melalui metode ini, dapat diperoleh data-data yang dibutuhkan melalui buku, jurnal, skripsi, dan juga media internet yang berhubungan dengan pengembangan sistem *Point of Sales* (POS) dengan

menggunakan *framework* berupa React Native untuk membangun aplikasi berbasis Android.

c. Analisa Sistem yang Berjalan

Pada saat ini, sistem pencatatan transaksi yang berjalan di Kedai A7 Pujasera Kecamatan Ngaliyan adalah pencatatan secara manual atau pencatatan dengan buku. Pada saat pesanan dari pelanggan masuk, pemilik Kedai A7 akan mencatat pesanan pelanggan yang berupa nama pesanan dan harga pesanan ke dalam buku catatan keuangan. Kemudian saat pesanan sudah ditutup, pemilik akan menghitung jumlah pesanan dan uang yang masuk dengan menggunakan kalkulator dan dicatatkan kembali ke dalam buku catatan keuangan.

d. Analisa Sistem Baru yang Diusulkan

Dilihat dari sistem yang telah berjalan, ditemukan adanya kekurangan berupa ketidakefektifan pencatatan dan kurangnya keamanan data transaksi yang dicatat. Selain itu, pemilik Kedai A7 juga mengakui bahwa terkadang jika banyak pesanan yang masuk, proses pencatatan lupa untuk dilakukan karena pemilik sibuk membuat pesanan pelanggan. Dari permasalahan tersebut, diusulkan suatu sistem baru berupa aplikasi

Point of Sales (POS) berbasis Android yang memiliki beberapa fitur yaitu:

- 1) Terdapat autentikasi pengguna dengan menggunakan email dan sandi. Sehingga data-data transaksi dapat aman diakses di beberapa *device* dengan menggunakan akun email yang telah dibuat.
- 2) Pemilik dapat menginputkan mengatur daftar menu, seperti menambahkan, mengubah, dan menghapus daftar menu.
- 3) Saat pesanan masuk, pemilik dapat dengan mudah mencatat pesanan dengan menekan daftar menu yang tampil di halaman utama aplikasi dan mengisi nama pemesan.
- 4) Setelah pesanan selesai dan pemesan telah membayar tagihan, pemilik dapat mencetak struk pesanan dengan menggunakan *printer* yang telah terkoneksi dengan aplikasi melalui koneksi Bluetooth.
- 5) Pemilik usaha dapat melihat riwayat transaksi yang telah terjadi. Pemilik dapat melihat tanggal transaksi, nama pelanggan, dan juga detail pembelian.
- 6) Laporan penjualan berupa rangkuman penjualan berdasarkan jangka waktu tertentu, misalkan harian, mingguan, hingga bulanan.

2. *System and Software Design*

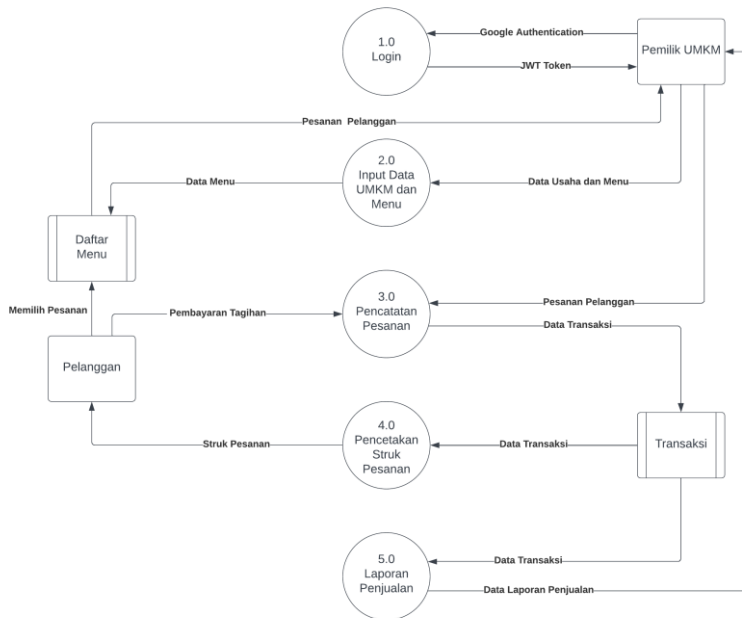
Berdasarkan persyaratan yang dikumpulkan, dibangunlah sebuah *system design*. Tahap desain ini mencakup desain *database*, arsitektur sistem, dan elemen-elemen lain yang relevan dengan perangkat lunak yang dikembangkan. Tahap ini membantu dalam menspesifikasikan kebutuhan sistem serta mendefinisikan arsitektur sistem secara keseluruhan (Harlinda & Nasir, 2020).

Pada tahap ini dilakukan proses perancangan berupa *data flow diagram* (DFD) untuk menggambarkan aliran data yang dari sistem, *entity relationship diagram* (ERD) untuk menggambarkan hubungan tiap entitas pada basis data yang digunakan, dan desain arsitektur aplikasi.

a. *Data Flow Diagram*

Data flow diagram (DFD) merupakan sebuah teknik grafis yang menggambarkan aliran data atau informasi yang diaplikasikan pada saat data bergerak dari input menjadi output.

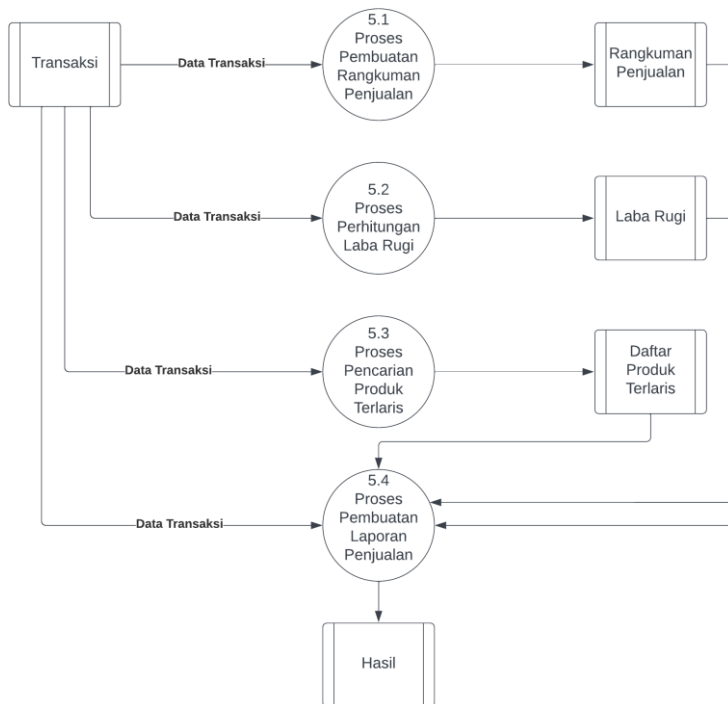
Data flow diagram (DFD) yang menggambarkan seluruh alur data pada aplikasi *Point of Sales* ditunjukkan oleh Gambar 3.1 di bawah ini.



Gambar 3.2 Data Flow Diagram Level 0

Terdapat dua entitas yaitu pemilik UMKM dan pelanggan yang memiliki akses terhadap proses-proses yang dijalankan oleh aplikasi. Pada awalnya, pemilik UMKM melakukan login dengan mengirimkan data Google Authentication, kemudian aplikasi memberikan *response* berupa token JWT yang berfungsi sebagai fitur keamanan berupa *authorization* (Proses 1.0). Setelah itu, data UMKM dan menu diinputkan yang nantinya pelanggan dapat memesan menu sesuai dengan daftar menu yang ada

(Proses 2.0). Data pesanan pelanggan kemudian diinputkan oleh pemilik UMKM dan menghasilkan suatu transaksi (Proses 3.0). Kemudian data transaksi tersebut dicetak ke dalam sebuah struk yang kemudian akan diberikan kepada pelanggan yang memesan (Proses 4.0). Selain itu, data transaksi tersebut juga disimpan pada *database* yang kemudian akan diolah menjadi laporan penjualan yang dapat diakses oleh pemilik UMKM (Proses 5.0). Selanjutnya, digambarkan proses 5.0 secara detail di dalam *data flow diagram* level 1. *Data flow diagram* level 1 pada proses 5.0 ditunjukkan pada gambar 3.2 berikut.



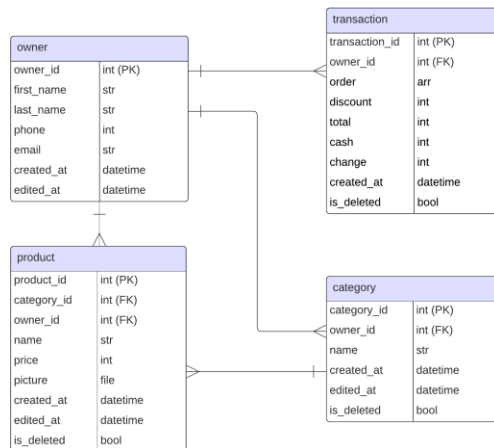
Gambar 3.3 Data Flow Diagram Level 1 Proses 5

Pada proses 5.0 terdapat beberapa subproses yaitu proses pembuatan rangkuman penjualan, proses perhitungan laba rugi, proses pencarian produk terlaris, dan proses pembuatan laporan penjualan. Seluruh proses ini mengolah data transaksi pelanggan yang kemudian menghasilkan rangkuman penjualan, laporan laba rugi,

dan daftar produk terlaris. Kemudian seluruh data tersebut dikumpulkan ke dalam laporan penjualan.

b. *Entity Relationship Diagram (ERD)*

Entity Relationship Diagram atau diagram hubungan antar entitas adalah model yang digunakan untuk menjelaskan hubungan antar tabel atau entitas dalam sistem basis data, berdasarkan konsep bahwa hubungan antar entitas tersebut sesuai dengan hubungan yang terjadi di dunia nyata (Kurniawan, 2023). Berikut ini rancangan *Entity Relationship Diagram (ERD)* pada aplikasi *Point of Sales (POS)* berbasis Android yang dapat dilihat pada gambar 3.3 berikut.



Gambar 3.4 Entity Relationship Diagram Aplikasi POS

3. *Implementation and Unit Testing*

Tahap ini melibatkan pengkodean dan pembuatan sistem berdasarkan desain *prototype* yang telah dibuat sebelumnya. Sistem dibuat dengan bahasa pemrograman, *framework*, dan *tools* sesuai dengan persyaratan yang telah ditentukan.

Proses pengembangan aplikasi *Point of Sales* (POS) berbasis Android ini dibagi menjadi dua tahapan, yaitu pengembangan aplikasi *front-end* Android dan pengembangan *web service*.

a. Pengembangan Aplikasi *Front-End* Android (*Client-Side*)

Pada proses ini, dibuat suatu aplikasi Android sebagai aplikasi *client-side* yang nantinya akan dipakai dan tampilan antarmukanya akan dirasakan langsung oleh pengguna. Aplikasi ini dibangun dengan menggunakan *framework* berbasis JavaScript yang berfungsi untuk membangun aplikasi *mobile* yaitu React Native. Di aplikasi ini, dibuat tampilan antarmuka pengguna (*user interface*) mulai dari *layouting*, *coloring*, dan *styling* setiap komponen pada halaman-halaman aplikasi. Selain itu, aplikasi React Native nantinya berfungsi untuk mengkoneksikan aplikasi *Point of Sales* (POS) berbasis

Android ini dengan Firebase untuk menerapkan autentikasi pengguna dengan Google Authentication.

Pengembangan berupa fitur-fitur di sisi klien juga dibangun di tahap ini, seperti penambahan transaksi secara manual di luar menu yang telah diinputkan, Selain itu, fungsi penyimpanan pesanan yang belum dibayar juga dibuat di sisi klien dengan menggunakan *state management* Redux, yang nantinya saat pelanggan membayar pesannya, data pesanan yang telah disimpan oleh Redux akan dikirimkan menuju *server*.

Proses integrasi API yang disediakan oleh *server* melalui *web service* juga dilakukan pada tahap ini. Hal tersebut memungkinkan aplikasi Android *client-side* dapat berkomunikasi dengan *web service* yang ada di *server-side*. Aplikasi *client-side* nantinya akan mengirim HTTP *request* dengan metode GET untuk membaca data, CREATE untuk menambah data, PUT untuk mengubah data, dan DELETE untuk menghapus data. *Request* tersebut kemudian akan diproses oleh *web service* kemudian mengirimkan *response* kembali ke aplikasi *client-side*.

b. Pengembangan Web Service (Server-Side)

Pada tahap ini, dibangun suatu *web service* dengan menggunakan *framework* Express.js yang berbasis

JavaScript. *Web service* ini nantinya akan memiliki koneksi ke basis data NoSQL yaitu MongoDB dengan menggunakan *package* berupa Mongoose. Selanjutnya, dilakukan definisi model atau skema basis data sesuai dengan desain ERD yang telah ditentukan. Kemudian setelah *web service* memiliki koneksi dengan *database* dibuatlah API dengan beberapa *endpoint* yang memiliki fungsinya masing-masing, seperti API *endpoint* untuk login, menambah menu, hingga mengakses laporan penjualan. Nantinya API *endpoint* tersebut akan dipakai oleh aplikasi *client-side* untuk melakukan *request* sesuai dengan fungsi yang dibutuhkan.

c. *Unit Testing*

Pada tahapan ini, dilakukan *testing* dengan mengecek apakah setiap *function* dalam sistem dapat berjalan sesuai dengan ekspektasi yang diinginkan atau tidak. Proses *unit testing* dilakukan dengan menggunakan Postman.

4. *Integration and System Testing*

Setelah sistem dibangun, tahap pengujian dilakukan untuk mengidentifikasi *bug* atau eror yang ada pada sistem. Pengguna atau klien dapat mencoba, menguji, dan memberikan umpan balik terhadap sistem untuk memastikan bahwa persyaratan telah terpenuhi. Pada

penelitian ini, dilakukan pengujian sistem menggunakan metode *Blackbox* dan *User Acceptance Test* (UAT).

Black box testing merupakan pengujian yang difokuskan pada spesifikasi fungsional atau fitur pada perangkat lunak dengan cara mendefinisikan kondisi-kondisi input yang memungkinkan dan akan melatih seluruh syarat-syarat fungsional suatu perangkat lunak (Cahyo, N. dkk., 2023). Skenario pengujian *black box* ditunjukkan pada table berikut.

Tabel 3.2 Skenario Black Box Testing

Menu	Aksi
<i>Login</i> Pengguna dengan Email	1. Email dan password benar 2. Email dan password salah
Mengisi Data Menu UMKM	1. Seluruh input kosong 2. Salah satu input kosong 3. Input nama menu kosong 4. Input harga kosong 5. Input kategori kosong 6. Seluruh input diisi dengan benar
Mengisi Daftar Kategori	1. Seluruh input kosong 2. Input nama kategori diisi dengan benar
Mencatat transaksi	1. Menu dipilih dari daftar menu

	2. Mengubah kuantitas menu 3. Input uang yang masuk
Mengecek laporan penjualan	1. Melihat laporan penjualan hari ini 2. Melihat laporan penjualan dengan menginputkan rentang waktu tertentu

User Acceptance Test (UAT) adalah metode pengujian aplikasi atau sistem yang dilakukan terhadap pengguna terakhir (*end user*) untuk mengetahui apakah aplikasi atau sistem yang dibangun sudah sesuai dengan kebutuhan pengguna (Wulandari dkk., 2023). Pengujian dengan metode UAT ini meninjau beberapa aspek yaitu aspek *functionality*, *reliability*, *usability*, dan *efficiency*.

Daftar pertanyaan yang digunakan dalam pengujian UAT ini ditunjukkan pada tabel 3.1 sebagai berikut.

Tabel 3.3 Daftar Pertanyaan Metode UAT

Kode	Pertanyaan
Aspek <i>Functionality</i> (Fungsionalitas)	
P1	Apakah pengguna dapat login dengan baik?
P2	Apakah pengguna dapat menginputkan data usaha dan menu dengan baik?

P3	Apakah pengguna dapat mencatat transaksi dengan baik?
P4	Apakah pengguna dapat mencetak struk transaksi dengan baik?
P5	Apakah pengguna dapat melihat laporan penjualan dengan baik?
Aspek Reliability (Keandalan)	
P6	Apakah data transaksi dapat disimpan dengan baik?
P7	Apakah aplikasi dapat menjalankan semua fungsi dengan cepat?
Aspek Usability (Kebergunaan)	
P8	Apakah aplikasi dapat membantu mengelola kasir dengan baik?
Aspek Efficiency (Efisiensi)	
P9	Apakah aplikasi ini dapat menghemat waktu pengguna dalam mencatat transaksi penjualan?
P10	Apakah aplikasi ini dapat menghemat waktu pengguna dalam membuat laporan penjualan?

Pertanyaan-pertanyaan tersebut kemudian diberikan kepada pemilik UMKM yang ada di Pujasera Kecamatan Ngaliyan. Pertanyaan-pertanyaan tersebut akan mengacu pada bobot *Mean Option Score* (MOS) yang ditunjukkan pada tabel 3.2 berikut.

Tabel 3.4 Bobot Nilai Mean Option Score (MOS)

MOS	Keterangan	Nilai
SS	Sangat Setuju	5

S	Setuju	4
KS	Kurang Setuju	3
TS	Tidak Setuju	2
TT	Tidak Tahu	1

Nilai-nilai hasil pengujian yang telah diberikan oleh pemilik UMKM kemudian dihitung persentasenya untuk mengetahui kriteria skala Likert dengan persamaan berikut (Wulandari dkk., 2023).

$$SkorTotal = \frac{SkorAktual}{SkorIdeal} \times 100\%$$

Skor total yang telah didapatkan pada setiap aspek pengujian kemudian akan dikonversikan menjadi kriteria skala Likert. Kriteria skala Likert ditunjukkan pada tabel berikut.

Tabel 3.5 Skala Likert

Skor Total (%)	Kriteria
20,00% – 36,00%	Tidak Baik
36,01% – 52,00%	Kurang Baik
52,01% – 68,00%	Cukup
68,01% – 84,00%	Baik
84,01% – 100%	Sangat Baik

5. *Operation and Maintenance*

Tahap ini berfokus pada proses pengoperasian sistem dan pemeliharaan sistem setelah sistem selesai dibuat

telah sesuai dengan kebutuhan yang diinginkan. Perbaikan *bug* baru yang muncul, pembaruan fitur, pengoptimalisasian fitur, dan peningkatan fitur yang telah ada dilakukan pada tahap ini. Proses pemeliharaan aplikasi *Point of Sales* (POS) ini dilakukan secara asinkronus oleh peneliti dengan bantuan pemilik UMKM yang berarti dapat terjadi dan dapat tidak terjadi tergantung dari pemilik UMKM yang menggunakan aplikasi *Point of Sales* (POS) ini, nantinya pemilik UMKM dapat menghubungi kontak peneliti jika terdapat permasalahan teknis pada aplikasi yang digunakan. Selain itu, pemeliharaan dari sisi *web service* juga akan dilakukan apabila terdapat eror yang muncul pada log aplikasi.

BAB IV

HASIL PENELITIAN DAN PEMBAHASAN

A. Requirement Definition

Pada tahapan ini dilakukan pengumpulan data dan informasi dengan mewawancarai *stand* Kedai A7 yang ada di Pujasera Kecamatan Ngaliyan. Data-data yang didapatkan kemudian dipakai proses pengembangan aplikasi. Dari proses pengumpulan data tersebut didapatkan data berupa kategori produk, daftar produk, dan harga produk yang ada pada *stand* Kedai A7. Data-data tersebut terdapat pada tabel di bawah ini.

Tabel 4.1 Daftar Menu pada UMKM Kedai A7

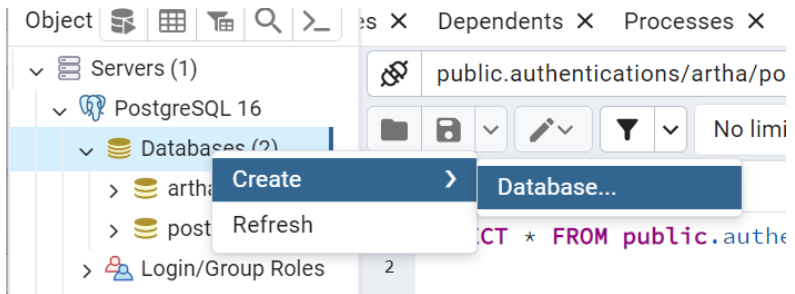
Kategori	Menu	Harga
Makanan	<i>Chicken Katsu</i> + Nasi + Es Teh	Rp14.000
	<i>Chicken Katsu</i> + Kentang + Es Teh	Rp16.000
	<i>Chicken Katsu</i> (Matah) + Nasi + Es Teh	Rp15.000
	<i>Chicken Katsu</i> (Matah) + Kentang + Es Teh	Rp17.000
	<i>Chicken Katsu</i>	Rp10.000
	<i>Chicken Katsu</i> (Matah)	Rp11.000
	<i>Chicken Steak</i> + Nasi + Es Teh	Rp13.000
	<i>Chicken Steak</i> + Kentang + Es Teh	Rp15.000

	<i>Chicken Steak</i>	Rp9.000
	Ayam Bakar Bumbu Bali + Nasi + Es Teh	Rp13.000
	Ayam Bakar Bumbu Bali	Rp9.000
	Kentang Goreng	Rp7.000
	Nasi Putih	Rp3.000
Minuman	Es Teh/Panas Manis	Rp3.000
	Es Teh/Panas Tawar	Rp2.000
	Es Jeruk/Panas	Rp4.000
	Es Jeruk Selasih	Rp5.000
	<i>Milkshake</i> Coklat	Rp8.000
	<i>Milkshake</i> Stroberi	Rp8.000
	<i>Milkshake</i> Vanilla	Rp8.000

B. *System and Software Design*

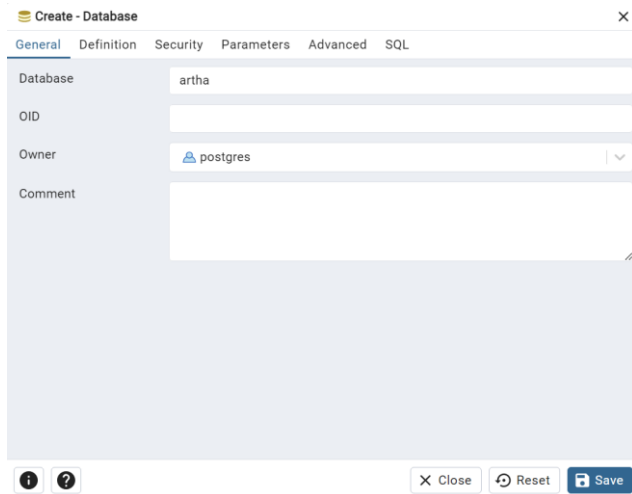
1. Membuat *Database*

Pembuatan *database* dilakukan pada aplikasi pgAdmin 4, yaitu aplikasi resmi dari PostgreSQL yang digunakan untuk *me-manage* kredensial *database*, membuat *database*, hingga menjalankan SQL *query*. Untuk membuat *database* PostgreSQL pada pgAdmin 4 adalah dengan mengklik kanan pada menu Databases, lalu klik Create, lalu klik “Database...” seperti pada gambar di bawah ini.



Gambar 4.1 Proses Pembuatan Database

Kemudian, akan muncul dialog Create – Database, lalu isi nama *database* dengan nama “artha” kemudian dikonfirmasi dengan mengklik tombol Save.



Gambar 4.2 Pengisian Nama Database

2. Membuat *Database Migration*

Pada proses ini dibuat migrasi *database* dengan tujuan untuk memindahkan struktur pada *database* dengan

efisien dan cepat tanpa perlu mendefinisikan kolom pada setiap table dalam *database*. Migrasi *database* pada aplikasi ini dibuat dengan menggunakan *library* node-pg-migrate yang berfungsi untuk mendefinisikan struktur tabel, kolom, relasi, dan tipe data dengan format JavaScript kemudian mengeksekusi perintah yang ada pada *file* JavaScript tersebut menjadi sebuah tabel beserta kolom, relasi, dan tipe data yang telah didefinisikan. Berikut ini merupakan contoh fungsi untuk melakukan migrasi database pada tabel *categories*.

```
// categories
pgm.createTable('categories', {
  id: {
    type: 'uuid',
    primaryKey: true,
  },
  name: {
    type: 'varchar(255)',
    notNull: true,
  },
  description: {
    type: 'text',
    notNull: false,
  },
  company_id: {
    type: 'uuid',
    notNull: true,
  },
  created_at: {
    type: 'timestamp without time zone',
    notNull: true,
    default: pgm.func('current_timestamp'),
  },
  updated_at: {
    type: 'timestamp without time zone',
    notNull: true,
    default: pgm.func('current_timestamp'),
  },
});
```

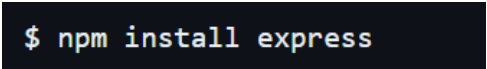
Gambar 4.3 Skrip Database Migration

C. *Implementation and Unit Testing*

1. Pengembangan *Web Service* (Express.js)

a. Menyiapkan *Working Environment* Express.js

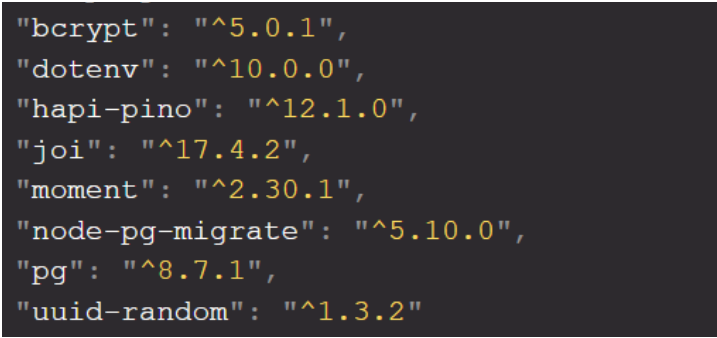
Untuk menginstal aplikasi Express, diperlukan JavaScript *runtime* yaitu Node.js. Jika Node.js telah diinstal pada server, jalankan perintah berikut untuk menginstal *framework* Express.js.



```
$ npm install express
```

Gambar 4.4 Perintah Instalasi *Web Service* Express.js

Kemudian, untuk mendukung pengembangan *web service* dengan Express.js, dibutuhkan beberapa *package* atau *library* seperti gambar di bawah ini.



```
"bcrypt": "^5.0.1",  
"dotenv": "^10.0.0",  
"hapi-pino": "^12.1.0",  
"joi": "^17.4.2",  
"moment": "^2.30.1",  
"node-pg-migrate": "^5.10.0",  
"pg": "^8.7.1",  
"uuid-random": "^1.3.2"
```

Gambar 4.5 Daftar *Package* Pada *Web Service*

Package-package pada gambar di atas digunakan untuk mempermudah pengembang dalam pengembangan aplikasi *web service*. Contohnya, seperti *bcrypt*

digunakan untuk melakukan enkripsi *password* saat pengguna mendaftar dan *login* akun, *dotenv* untuk membaca *environment file* yang berisi kredensial *database*, *ip port*, dan *host server*, *hapi-pino* digunakan untuk mencatat *error log*, *joi* digunakan sebagai *request validation*, *moment* untuk memformat *datetime*, *node-pg-migrate* dan *pg* untuk menjalankan *query database* dan migrasi *database*, dan *uuid-random* untuk *generate uuid*.

b. Membuat *Routes*

Routes merupakan jalur atau rute yang dapat diakses oleh *client* untuk menjalankan perintah tertentu. *Routes* pada dasarnya menyimpan beberapa hal dasar dari suatu *web service* seperti *method*, *path*, hingga *handler*. *Routes* ini dapat digambarkan sebagai pintu pada sebuah rumah. Setiap *request* yang masuk ke API harus melewati *routes* terlebih dahulu untuk melakukan pengalamatan, yaitu pengecekan *path* dan juga *method* yang diminta apakah terdapat pada API atau tidak, jika ada, maka *routes* akan memanggil *handler* untuk memproses *request* tersebut. Berikut ini merupakan contoh *file routes* pada *service products*.

```
const routes = (handler) => [
  {
    method: 'POST',
    path: '/products',
    handler: handler.postProductHandler,
    options: {
      auth: 'artha_jwt',
    },
  },
  {
    method: 'GET',
    path: '/products',
    handler: handler.getProductsHandler,
    options: {
      auth: 'artha_jwt',
    },
  },
  {
    method: 'GET',
    path: '/products/{id}',
    handler: handler.getProductByIdHandler,
    options: {
      auth: 'artha_jwt',
    },
  },
  {
    method: 'PUT',
    path: '/products/{id}',
    handler: handler.putProductsHandler,
    options: {
      auth: 'artha_jwt',
    },
  },
]
```

Gambar 4.6 Proses Routing Pada Web Service

Pada gambar di atas terdapat pendefinisian di mana *service* dapat diakses pada *path*, lalu *method* apa saja yang diterima, lalu *function* apa yang akan dijalankan pada *handler*.

c. Membuat *Handler*

Handler merupakan *function* yang akan dijalankan apabila *client* mengakses *route* di mana *handler* didefinisikan. *Handler* akan menerima *request body* yang dikirimkan oleh *client*, kemudian memprosesnya

dan mengirimkan Kembali hasil proses yang telah dijalankan kepada *client*. Berikut ini merupakan *file handler* pada *service products*.

```
async postProductHandler(request, h) {
  try {
    this._validator.validatePostProductPayload(request.payload);

    const { companyId } = request.auth.credentials;
    const {
      code, name, description, price, cost, stock, category_id: categoryId,
    } = request.payload;

    const productId = await this._service.addProduct({
      code, name, description, price, cost, stock, categoryId, companyId,
    });

    const response = h.response({
      status: 'success',
      message: 'Product berhasil ditambahkan',
      data: {
        productId,
        name,
      },
    });
    response.code(201);
    return response;
  } catch (error) {
    return error;
  }
}
```

Gambar 4.7 Proses Pembuatan *Handler* Pada *Web Service*

Pada *function postProductHandler*, *handler* menerima *request body* atau data yang dikirimkan oleh *client* dari parameter *request*, yang di dalamnya memuat data *companyId*, *code*, *name*, *description*, *price*, *cost*, *stock*, dan *categoryId*. Lalu, data-data *request* tersebut divalidasi oleh *validator validateProductPayload*. Lalu, jika data telah lolos validasi, *handler* akan menjalankan

function service addProduct yang akan melakukan *query* ke *database*.

d. Membuat *Service Query*

Pada tahap ini dilakukan pembuatan *service query* yang berfungsi untuk menjalankan *query database* dari data-data yang telah divalidasi. Berikut ini merupakan *service query* pada *service products*.

```
class ProductsService {
  async addProduct({
    code, name, description, price, cost, stock, categoryId, companyId,
  }) {
    const productId = uuid();
    const stockId = uuid();

    const productQuery = {
      text: `INSERT INTO products(id, code, name, description, price, cost, category_id,
        VALUES
        ($1, $2, $3, $4, $5, $6, $7, $8, (SELECT id FROM units WHERE company_id = $8
        values: [productId, code, name, description, price, cost, categoryId, companyId],
      );

    // update stock default warehouse default office
    const stockQuery = {
      text: `INSERT INTO
        stocks(id, product_id, stock, warehouse_id, sale, purchase)
        VALUES
        ($1, $2, $3, (
          SELECT id FROM warehouses WHERE office_id =
          (SELECT id FROM offices WHERE company_id = $4 LIMIT 1)
          LIMIT 1),
          0, 0)`,
      values: [stockId, productId, stock, companyId],
    };

    const client = await this._pool.connect();

    try {
      await client.query('BEGIN');
      await client.query(productQuery);
      await client.query(stockQuery);
      await client.query('COMMIT');
    } catch (err) {
      await client.query('ROLLBACK');
```

Gambar 4.8 Proses Pembuatan *Service Query*

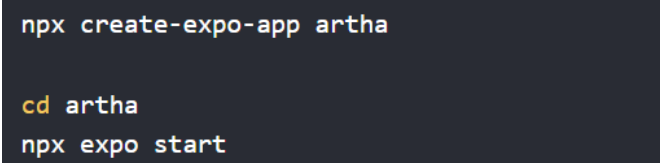
Pada *service addProduct* dilakukan dua manipulasi pada table *database* yaitu table *products* dan *stocks*.

Setelah data divalidasi oleh *handler* dijalankan proses *query database* berupa INSERT INTO yang berfungsi untuk menambahkan data (*row*) baru.

2. Pengembangan Aplikasi Android (React Native)

a. Menyiapkan *Working Environment* React Native

Tahap pertama pada proses pengembangan aplikasi React Native adalah dengan menginstal *framework* tersebut dengan perintah sebagai berikut.



```
npx create-expo-app artha  
  
cd artha  
npx expo start
```

Gambar 4.9 Inisialisasi *Project* React Native

Kemudian, dilanjutkan dengan menyiapkan *dependency* berupa *library* yang digunakan untuk menunjang fitur-fitur yang ada pada aplikasi. *Library-library* tersebut diatur oleh *package manager* yaitu Yarn. Yarn berfungsi untuk menginisialisasi versi dari setiap *library*, menambahkan, hingga menghapus *library* yang ada pada *project*. *Dependency* atau *library* ini didefinisikan pada file *package.json* seperti di bawah ini.

```

package.json > {} dependencies
4   "scripts": {
9     "eject": "expo eject"
10  },
11  "dependencies": {
12    "@expo/metro-runtime": "~3.1.3",
13    "react-native-async-storage/async-storage": "1.21.0",
14    "@react-navigation/native": "^6.0.13",
15    "@react-navigation/native-stack": "^6.9.0",
16    "axios": "^0.27.2",
17    "deprecated-react-native-prop-types": "^2.3.0",
18    "expo": "^50.0.0",
19    "expo-barcode-scanner": "~12.9.3",
20    "expo-splash-screen": "~0.26.4",
21    "expo-status-bar": "~1.11.1",
22    "is_js": "^0.9.0",
23    "native-base": "^3.4.16",
24    "react": "18.2.0",
25    "react-dom": "18.2.0",
26    "react-native": "0.73.6",
27    "react-native-daterange-picker": "^1.5.1",
28    "react-native-pager-view": "6.2.3",
29    "react-native-safe-area-context": "4.8.2",
30    "react-native-screens": "~3.29.0",
31    "react-native-snackbar-component": "^1.1.12",
32    "react-native-svg": "14.1.0",
33    "react-native-tab-view": "^3.1.1",
34    "react-native-web": "~0.19.6",
35    "use-debounce": "^8.0.4"
36  },

```

Gambar 4.10 Daftar *Package* yang Digunakan Pada Aplikasi React Native

Lalu, untuk menginstal *dependencies* yang telah didefinisikan adalah dengan menjalankan perintah ‘yarn install’ pada terminal. Pada proses ini, *yarn* akan meneginstal semua *library* yang telah diinisialisasi pada *property* ‘dependencies’ pada *file* package.json sesuai dengan versi yang telah didefinisikan. Jika semua *dependency* telah berhasil diinstal, maka akan muncul pesan seperti di bawah ini.

```

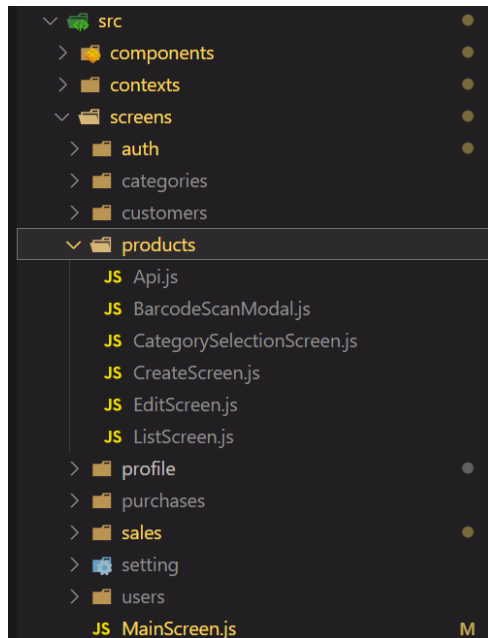
PS C:\Users\linta\Documents\artha> yarn install
yarn install v1.22.21
warning package-lock.json found. Your project contains lock files generated by tools other than Yarn. It is
advised not to mix package managers in order to avoid resolution inconsistencies caused by unsynchronized
lock files. To clear this warning, remove package-lock.json.
[1/4] Resolving packages...
success Already up-to-date.
Done in 0.49s.

```

Gambar 4.11 Instalasi Package yang Dibutuhkan

b. Pengkodean *User Interface*

Tahapan selanjutnya adalah proses pembuatan *component* tampilan per halaman aplikasi. *Script* dibuat pada folder *src* dan dalam *subfolder screens* seperti di bawah ini.



Gambar 4.12 Project Structure Pada Folder *screens*

Pada gambar di atas, dapat dilihat beberapa halaman yang ada pada aplikasi, seperti halaman *auth* hingga halaman *users* yang di masing-masing di dalamnya terdapat fitur-fitur khusus, misalnya pada *screens products*, terdapat *script CreateScreen.js* yang didalamnya berisi *component user interface* yang berfungsi untuk membuat produk. Di dalamnya terdapat *form* untuk mengisi data nama menu, harga, hingga stok menu seperti gambar di bawah ini.

```

91 export default function CreateScreen({ navigation }) {
92
93   </FormControl>
94   <FormControl isValid={isValid}>
95     <FormControl.Label>Nama Menu</FormControl.Label>
96     <Input
97       variant="outline"
98       placeholder="nama"
99       onChangeText={({text}) => setName(text)}
100       value={name}
101     />
102     <FormControl.ErrorMessage
103       leftIcon={<WarningOutlineIcon size="xs" />}
104     >
105       Nama wajib diisi
106     </FormControl.ErrorMessage>
107   </FormControl>
108   <FormControl>
109     <FormControl.Label>Deskripsi</FormControl.Label>
110     <Input
111       variant="outline"
112       placeholder="Deskripsi"
113       onChangeText={({text}) => setDescription(text)}
114       value={description}
115     />
116   </FormControl>

```

Gambar 4.13 Skrip User Interface Pada Halaman Produk

Dari skrip *user interface* di atas, dihasilkan tampilan halaman tambah produk seperti gambar di bawah ini.

14:47 77%

≡ **Buat Produk**

Kode

BR1715586469961

Kategori

Kategori

Nama Menu

nama

Deskripsi

Deskripsi

Harga Jual

0

Stok

0

Simpan

Gambar 4.14 Halaman Tambah Produk

c. Integrasi Aplikasi Android dengan *Web Service*

Setelah proses pengkodean *user interface* telah selesai dilakukan, proses selanjutnya adalah integrasi aplikasi Android dengan *web service* yang telah dibuat. Proses integrasi dilakukan dengan membuat fungsi-fungsi

HTTP *request* dengan menggunakan modul *axios*. Misalnya pada halaman *products*. Dalam *file Api.js* terdapat beberapa fungsi yang melakukan *request* ke *web service product* seperti pada gambar di bawah ini.

```
src > screens > products > JS Api.js > ...
36 export function createProduct(payload, token) {
37   return axios({
38     method: 'POST',
39     url: '/products',
40     data: payload,
41     headers: {
42       Authorization: 'Bearer ${token}',
43     },
44   })
45   .then((res) => res.data)
46   .catch((err) => {
47     throw err.response.data
48   })
49 }
50
51 export function getProduct(id, token) {
52   return axios({
53     method: 'GET',
54     url: '/products/${id}',
55     headers: {
56       Authorization: 'Bearer ${token}',
57     },
58   })
59   .then((res) => res.data.data)
60   .catch((err) => {
61     throw err.response.data
62   })
63 }
```

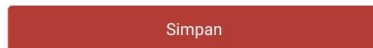
Gambar 4.15 Integrasi Web Service dengan React Native

Pada gambar di atas, terdapat fungsi *createProduct* dan *getProduct* yang masing-masing memiliki fungsi untuk membuat produk dan mendapatkan data produk. Setelah fungsi HTTP *request* telah didefinisikan, kemudian fungsi tersebut akan dipanggil dengan *trigger* tertentu, contohnya fungsi *createProduct* akan dipanggil pada saat pengguna menekan tombol *submit*

saat menambahkan menu baru seperti pada gambar berikut ini.

```
const handleSubmit = () => {
  if (name.length <= 0) {
    setIsInvalid(true)
    return
  }
  setLoading(true)
  createProduct({
    code,
    name,
    description,
    cost,
    price,
    stock,
    category_id: category ? category.id : '',
  }, user.accessToken)
    .then(() => {
      ToastAndroid.show('Produk ditambahkan', ToastAndroid.SHORT)
      navigation.goBack()
    })
    .catch((err) => {
      ToastAndroid.show(err?.message, ToastAndroid.SHORT)
    })
    .finally(() => setLoading(false))
}
```

Gambar 4.16 Skrip Trigger *handleSubmit*



Gambar 4.17 Tombol Simpan Pada Halaman Tambah Produk

Pada gambar di atas, terdapat fungsi *handleSubmit* yang akan dipanggil saat pengguna membuat produk baru dengan menekan tombol “Simpan”. Di dalam fungsi tersebut terdapat pemanggilan fungsi *createProduct* dengan beberapa parameter yang dikirimkan seperti

code, *name*, *price*, hingga *category_id*. Lalu, setelah proses berhasil atau gagal, akan ditampilkan *toast* sebagai notifikasi.

3. Unit Testing

Pada proses *unit testing*, dilakukan percobaan menggunakan *software* Postman. *Test case* yang dijalankan berupa pengaksesan *endpoint* pada *web service* dengan mengirimkan data *request* dan mengecek apakah hasilnya akan sukses atau gagal. Konfigurasi untuk melakukan *unit testing* pada Postman ada pada gambar di bawah ini.

```
"name": "login",
"event": [
  {
    "listen": "test",
    "script": {
      "exec": [
        "pm.test('test status code', () => {",
        "  pm.response.to.have.status(201)",
        "});",
        "",
        "pm.test('test login accesstoken', () => {",
        "  const responseJson = pm.response.json();",
        "",
        "  pm.expect(responseJson).to.haveOwnProperty('status');",
        "  pm.expect(responseJson.status).to.equals('success');",
        "",
        "  pm.environment.set('accessToken', responseJson.data.accessToken)",
        "  pm.environment.set('refreshToken', responseJson.data.refreshToken)",
        "  pm.environment.set('officeId', responseJson.data.user.officeid)",
        "});",
      ],
      "type": "text/javascript"
    }
  }
],
```

Gambar 4.18 Konfigurasi Unit Testing Pada Postman

D. System Testing

Proses *system testing* dilakukan pada hari Minggu, 12 Mei 2024 di Pujasera Ngaliyan, tepatnya di *stand* Kedai A7. Proses *testing* diuji langsung oleh pemilik Kedai A7, yaitu Ibu Ria. Pada tahapan ini, *system testing* dibagi menjadi dua bagian, yaitu *black box testing* dan *user acceptance test* (UAT).

1. *Black Box Testing*

Pada tahapan ini, dilakukan *black box testing* dengan tujuan untuk menguji fungsionalitas perangkat lunak yang dibangun, apakah pada aplikasi terdapat kesalahan antarmuka (*user interface*), struktur data, kesalahan performa, dan lain sebagainya. Tabel di bawah ini menunjukkan hasil *black box testing* yang telah dilakukan.

Tabel 4.1 Hasil *Black Box Testing* Pada Kedai A7

Menu	Aksi	Output yang Diharapkan	Hasil Testing
Login Pengguna dengan Email	Email dan password benar	Berhasil	Sesuai
	Email dan password salah	Gagal	Sesuai

Mengisi Data Menu UMKM	Seluruh input kosong	Gagal	Sesuai
	Salah satu input kosong	Gagal	Sesuai
	Input nama menu kosong	Gagal	Sesuai
	Input harga kosong	Gagal	Sesuai
	Input kategori kosong	Gagal	Sesuai
	Seluruh input diisi dengan benar	Berhasil	Sesuai
Mengisi Daftar Kategori	Seluruh input kosong	Gagal	Sesuai
	Input nama kategori diisi dengan benar	Berhasil	Sesuai
Mencatat transaksi	Menu dipilih dari daftar menu	Berhasil	Sesuai
	Input uang yang masuk	Berhasil	Sesuai

	Mengubah kuantitas menu	Berhasil	Sesuai
Mengecek laporan penjualan	Melihat laporan penjualan hari ini		Sesuai
	Melihat laporan penjualan dengan menginputkan rentang waktu tertentu		Sesuai

2. *User Acceptance Test (UAT)*

Pada tahap UAT, dilakukan pengujian untuk mengetahui apakah aplikasi telah sesuai dengan kebutuhan pengguna dengan meninjau dari beberapa aspek, seperti fungsionalitas, keandalan, kebergunaan, dan efisiensi. Tabel di bawah ini menunjukkan hasil UAT yang telah dilakukan.

Tabel 4.2 Hasil *User Acceptance Testing Pada Kedai A7*

Kode	Pertanyaan	Nilai
Aspek Functionality (Fungsionalitas)		
P1	Apakah pengguna dapat mendaftar dan login dengan baik?	5

P2	Apakah pengguna dapat menginputkan data usaha dan menu dengan baik?	4
P3	Apakah pengguna dapat mencatat transaksi dengan baik?	5
P4	Apakah pengguna dapat mencetak struk transaksi dengan baik?	2
P5	Apakah pengguna dapat melihat laporan penjualan dengan baik?	5
Aspek Reliability (Keandalan)		
P6	Apakah data transaksi dapat disimpan dengan baik?	4
P7	Apakah aplikasi dapat menjalankan semua fungsi dengan cepat?	3
Aspek Usability (Kebergunaan)		
P8	Apakah aplikasi dapat membantu mengelola kasir dengan baik?	4
Aspek Efficiency (Efisiensi)		
P9	Apakah aplikasi ini dapat menghemat waktu pengguna dalam mencatat transaksi penjualan?	5
P10	Apakah aplikasi ini dapat menghemat waktu pengguna dalam membuat laporan penjualan?	5

Kemudian, dilakukan perhitungan untuk mencari skor total dengan rumus berikut ini.

$$SkorTotal = \frac{5 + 4 + 5 + 2 + 5 + 4 + 3 + 4 + 5 + 5}{5.10} \times 100\%$$

$$SkorTotal = \frac{42}{50} \times 100\%$$

$$SkorTotal = 84\%$$

Skor total yang telah didapatkan, yaitu 84% kemudian dimasukkan ke dalam skala Likert, dan dihasilkan kriteria “Baik” dengan rentang skor 68,01% — 84%.

E. *Operation and Maintenance*

Pada proses ini, dilakukan *bug fixing* dan penyesuaian sesuai dengan permintaan dan saran dari pengguna setelah dilakukan proses *testing*. Berikut ini merupakan daftar saran yang diberikan oleh Ibu Ria, pemilik *stand* UMKM Kedai A7.

Tabel 4.3 Daftar Masukan Setelah Dilakukan *System Testing*

Daftar
1. <i>Form</i> kategori pada halaman tambah produk dibuat di atas nama produk karena membingungkan pengguna jika diletakkan di bagian bawah.
2. <i>Field</i> harga beli (modal) lebih baik dihilangkan karena modal fluktuatif (tidak konstan)
3. Pada menu keranjang di halaman kasir, dibuat kuantitas per item.
4. Mengimplementasikan fitur <i>share pdf</i> untuk nota pembelian dan laporan penjualan.

BAB V

KESIMPULAN DAN SARAN

A. Kesimpulan

Pembangunan aplikasi Android *point of sales* dengan menggunakan React Native yang diterapkan di *stand* Pujasera Kedai A7 dibutuhkan untuk mempermudah proses pencatatan pesanan pelanggan dan proses pembuatan laporan penjualan karena sistem yang berjalan sekarang masih dilakukan secara manual mulai dari proses pencatatan pesanan pelanggan dan proses pembuatan laporan penjualan dengan melakukan pencatatan di atas buku atau kertas yang masih menimbulkan banyak permasalahan. Berikut ini kesimpulan yang dapat diambil oleh penulis dari perancangan aplikasi Android *point of sales* dengan menggunakan React Native.

1. Aplikasi ini dibangun untuk memenuhi kebutuhan pemilik UMKM dalam proses pencatatan pesanan dan pembuatan laporan penjualan.
2. Membantu proses pencatatan laporan penjualan menjadi lebih aman dan optimal karena pengolahan data dilakukan dalam sistem dan penyimpanan data di dalam *database*.

3. Memberikan kemudahan dalam pencarian informasi penjualan yang telah lampau dengan efektif dan akurat.

B. Saran

Berikut saran yang diberikan penulis terhadap perancangan dan pembangunan aplikasi Android *point of sales* dengan menggunakan React Native yang telah dibuat.

1. Kedepannya diharapkan fitur cetak struk pembelian bisa ditambahkan dan diintegrasikan sehingga dapat berjalan dengan baik dan benar.
2. Meningkatkan fitur laporan penjualan dengan lebih detail, seperti fitur daftar produk terlaris dalam jangka waktu tertentu.
3. Menambahkan hasil *build* aplikasi pada React Native ke aplikasi iOS.

DAFTAR PUSTAKA

- Amazon Web Service. (2023). *NoSQL Article*.
<https://aws.amazon.com/id/nosql>
- Ariandi, F., Handayani, D., & Faozi, K. (2022). Perancangan Sistem Informasi Forum Paud Berbasis Web dengan Metode Waterfall. *Jurikom (Jurnal Riset Komputer)*, 9(6), 1857.
- Bank Indonesia. (2022). *Cerita BI*.
<https://www.bi.go.id/id/publikasi/ruang-media/cerita-bi/Pages/Konsumennya-Saja-Sudah-Digital-UMKM-nya-Juga-Dong.aspx>.
- Booch, G. (2007). *Object-Oriented Analysis and Design with Applications*. Addison-Wesley.
- Budiono, P. (2023). *Rancang Bangun Aplikasi Point of Sales (POS) dan Pengelolaan Stok Berbasis Web pada CV Daya Logam 2 Menggunakan Framework Laravel*. Universitas Multimedia Nusantara.
- CDW. (2022). *CDW Articles*.
<https://www.cdw.com/content/cdw/en/articles/hardware/what-is-a-thermal-printer-how-does-it-work.html>
- Cambridge Dictionary. (2024). *Dictionary*.
<https://dictionary.cambridge.org/dictionary/>
- Elmasari, R., & Navathe, S. B. (2016). *Fundamentals of Database System 7th edition* (7 ed.). Pearson.

- Express.js Documentation. (2023). *About Express.js*.
<https://expressjs.com>
- Forbes. (2018). *Forbes Innovation Newsletter*.
<https://www.forbes.com/sites/forbestechcouncil/2018/05/25/what-is-cloud-point-of-sale-and-how-is-it-different-from-traditional-pos/#25dad5201180>
- GlobalStats. (2024). Stat Counter.
<https://gs.statcounter.com/os-market-share/mobile/worldwide/#monthly-202301-202401>
- Harlinda, & Nasir. (2020). Perancangan Sistem Pendukung Keputusan Dalam Pengalokasian Dana Bantuan Sosial di Kabupaten Pinrang dengan Menggunakan Metode AHP. *Indonesian Journal of Data and Science (IJODAS)*, 1(2), 44–51.
- Jaber, Yahya, M., Misbahuddin, & Akbar, L. A. (2021). Rancang Bangun Aplikasi Point of Sales Berbasis Android Pada Mebel Selaparang. *Jurnal Teknik Elektro UNRAM*.
- Kurniawan, H. (2023). *Belajar Web Programming (Referensi Pengenalan Dasar Tahapan Belajar Pemrograman Web Untuk Pemula)* (1 ed.). PT. Sonpedia Publishing.
- Pressman, R. (2014). *Software Engineering A Practitioner's Approach 7th edition* (7 ed.). McGraw-Hill.
- Putra, A., Huddin, L., Akbar, F., & Thoyibah, T. (2023). PERANCANGAN SISTEM APLIKASI POINT OF SALES (POS) CV BAHRI BERKAH JAYA BERBASIS WEB. *Oktal: Jurnal Ilmu Komputer dan Sains*, 2(6), 1605–1615.
- React Native Documentation. (2023). *About React Native*.
<https://reactnative.dev>

- Restiawan, & Ramos. (2021). Perancangan Aplikasi Point Of Sales (POS) Berbasis Android (Studi Kasus: Warkop Vape Salatiga). *Jurnal Teknik Informatika dan Sistem Informasi*, 8(4), 1923–1938.
- Schmidt, C., Gokhale, A., & Natarajan, B. (2004). *Leveraging Application Frameworks*. ACM Queue.
- Sugiyono. (2017). *Metode Penelitian Kuantitatif, Kualitatif, dan R&D*. Alfabeta.
- Vito, J. (2021). *Rancang Bangun Aplikasi Point of Sales Berbasis Android pada Sistem Basis Data* [Universitas Multimedia Nusantara]. <https://kc.umn.ac.id/id/eprint/18007>
- Wibowo, N. C. H., Khaq, A. M. I., & Nur'aini, S. (2023). Pengembangan Sistem Layanan Administrasi Kependudukan Desa Papasan dengan QR Code sebagai Validasi Dokumen. *Walisongo Journal of Information Technology*, 5(1), 27–40. <https://doi.org/10.21580/wjit.2023.5.1.13936>
- Wulandari, Nofiyani, & Hasugian, H. (2023). USER ACCEPTANCE TESTING (UAT) PADA ELECTRONIC DATA PREPROCESSING GUNA MENGETAHUI KUALITAS SISTEM. *JMIK (Jurnal Mahasiswa Ilmu Komputer)*, 4(1), 20–27.

LAMPIRAN

Lampiran 1. Hasil *blackbox testing* di Kedai A7

KUISIONER PENELITIAN BLACK BOX TESTING RANCANG BANGUN APLIKASI *POINT OF SALES* BERBASIS ANDROID MENGGUNAKAN *FRAMEWORK* REACT NATIVE

Nama Penguji: Ria

Tempat Pengujian: Kedai A7, Pujasera Ngaliyan

Menu	Aksi	Hasil
Login Pengguna dengan Email	Email dan password benar	Berhasil
	Email dan password salah	Gagal
Mengisi Data Menu UMKM	Seluruh input kosong	Gagal
	Salah satu input kosong	Gagal
	Input nama menu kosong	Gagal
	Input harga kosong	Gagal
	Input kategori kosong	Gagal
	Seluruh input diisi dengan benar	Berhasil
Mengisi Daftar Kategori	Seluruh input kosong	Gagal

	Input nama kategori diisi dengan benar	Berhasil
Mencatat transaksi	Menu dipilih dari daftar menu	Berhasil
	Input uang yang masuk	Berhasil
	Mengubah kuantitas menu	Berhasil
Mengecek laporan penjualan	Melihat laporan penjualan hari ini	Berhasil
	Melihat laporan penjualan dengan menginputkan rentang waktu tertentu	Berhasil

Semarang, 12 Mei 2024



Pemilik Kedai A7 (Ria)

Lampiran 2. Hasil *user acceptance testing* (UAT) di Kedai A7

KUISIONER PENELITIAN UAT

RANCANG BANGUN APLIKASI *POINT OF SALES* BERBASIS ANDROID MENGGUNAKAN *FRAMEWORK* REACT NATIVE

Nama Penguji: Ria

Tempat Pengujian: Kedai A7, Pujasera Ngaliyan

Keterangan:

MOS	Keterangan	Nilai
SS	Sangat Setuju	5
S	Setuju	4
KS	Kurang Setuju	3
TS	Tidak Setuju	2
TT	Tidak Tahu	1

Kode	Pertanyaan	Nilai
Aspek <i>Functionality</i> (Fungsionalitas)		
P1	Apakah pengguna dapat mendaftar dan login dengan baik?	5
P2	Apakah pengguna dapat menginputkan data usaha dan menu dengan baik?	4
P3	Apakah pengguna dapat mencatat transaksi dengan baik?	5
P4	Apakah pengguna dapat mencetak struk transaksi dengan baik?	2

P5	Apakah pengguna dapat melihat laporan penjualan dengan baik?	5
Aspek Reliability (Keandalan)		
P6	Apakah data transaksi dapat disimpan dengan baik?	4
P7	Apakah aplikasi dapat menjalankan semua fungsi dengan cepat?	3
Aspek Usability (Kebergunaan)		
P8	Apakah aplikasi dapat membantu mengelola kasir dengan baik?	4
Aspek Efficiency (Efisiensi)		
P9	Apakah aplikasi ini dapat menghemat waktu pengguna dalam mencatat transaksi penjualan?	5
P10	Apakah aplikasi ini dapat menghemat waktu pengguna dalam membuat laporan penjualan?	5

Semarang, 12 Mei 2024

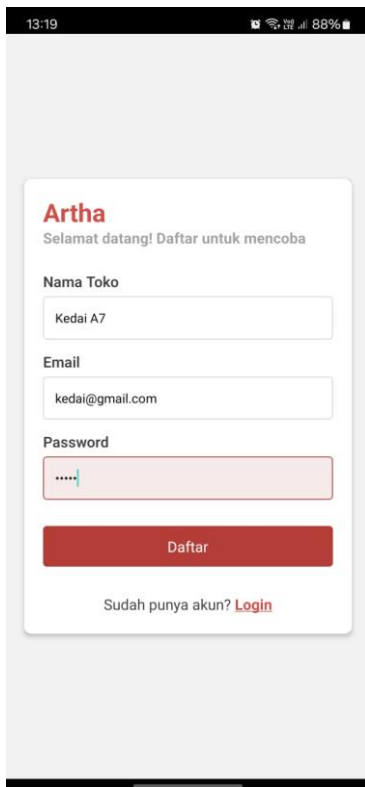


Pemilik Kedai A7 (Ria)

Lampiran 3. Dokumentasi proses *system testing* di Pujasera Ngaliyan



Lampiran 4. Tampilan *interface* aplikasi



13:19 88%

Artha
Selamat datang! Daftar untuk mencoba

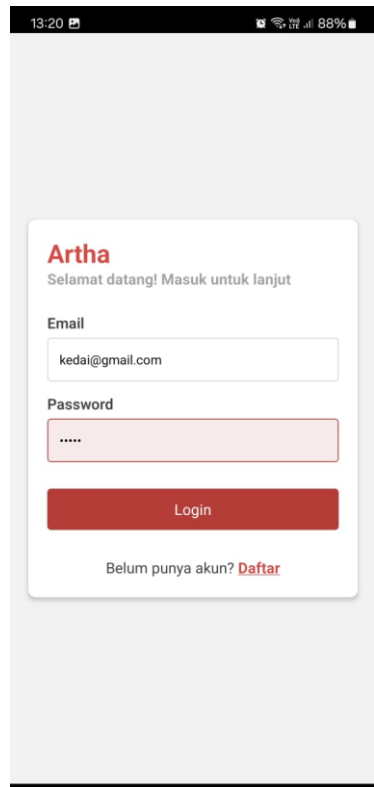
Nama Toko

Email

Password

Daftar

Sudah punya akun? [Login](#)



13:20 88%

Artha
Selamat datang! Masuk untuk lanjut

Email

Password

Login

Belum punya akun? [Daftar](#)

13:19 88%

Artha
Selamat datang! Daftar untuk mencoba

Nama Toko

Kedai A7

Email

kedai@gmail.com

Password

.....

Daftar

Sudah punya akun? [Login](#)

Pendaftaran berhasil silahkan login [LOGIN](#)

14:48 77%

Pelanggan

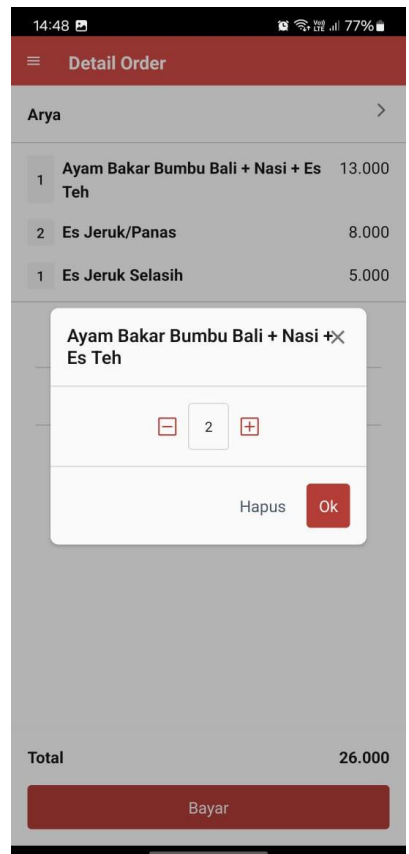
cari

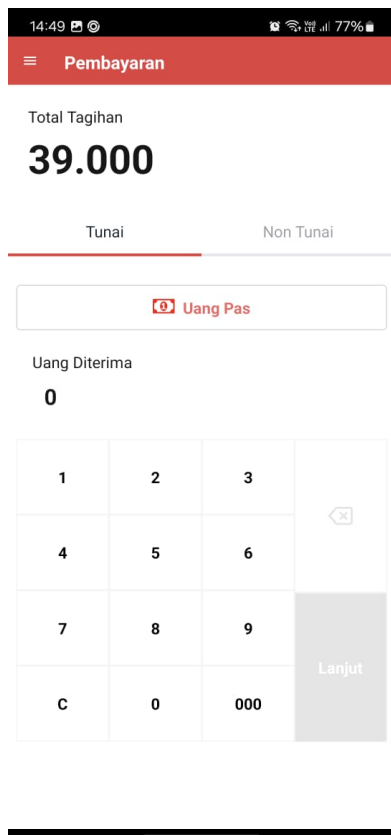
A Arya
087892928787

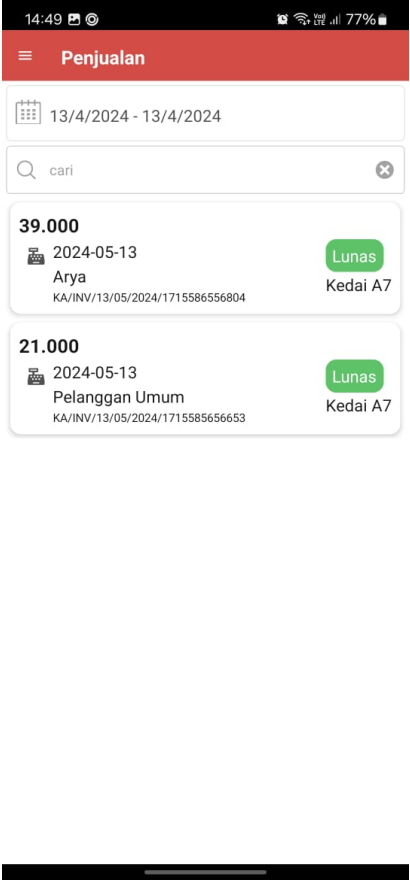
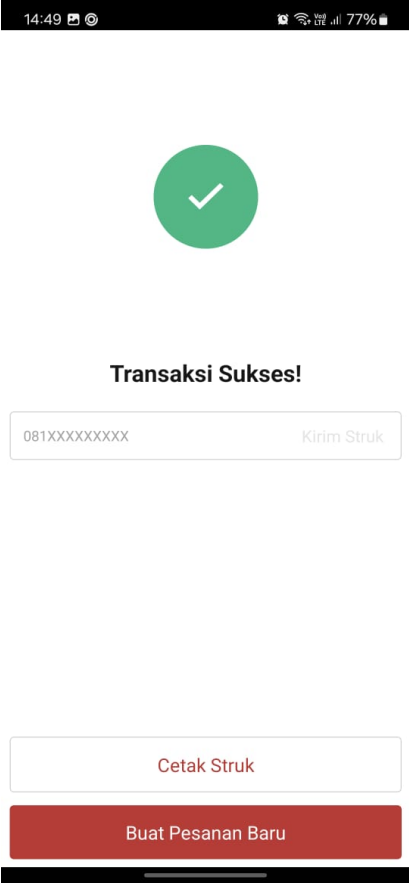
PU Pelanggan Umum

Pelanggan ditambahkan

+ Baru







14:49

77%

KA/INV/13/05/2024/1715586556804

Arya

13/4/2024

39.000

Kedai A7

5 barang

Lunas

2

Ayam Bakar Bumbu Bali + Nasi + Es Teh

26.000

2

Es Jeruk/Panas

8.000

1

Es Jeruk Selasih

5.000

Subtotal

39.000

Diskon

0

Total

39.000

Kirim

Cetak

Lampiran 5. Daftar Riwayat hidup

RIWAYAT HIDUP

A. Identitas Diri

1. Nama Lengkap : Lintang Aji Yoga Pratama
2. Tempat & Tanggal Lahir : Wonosobo, 28 Oktober 2002
3. Alamat Rumah : Talaga Bestari Blok E2 no. 17
RT 13 RW 01, Kecamatan
Sindang Jaya, Kabupaten
Tangerang, Banten.
4. HP : 085892942057
5. Email : lintangpratama@gmail.com

B. Riwayat Hidup

1. Madrasah Ibtidaiyah Negeri (MIN) 7 Kabupaten Tangerang
2. Sekolah Menengah Pertama Negeri (SMPN) 1 Balaraja
3. Sekolah Menengah Atas Negeri (SMAN) 1 Kabupaten
Tangerang

Semarang, 21 Mei 2024



Lintang Aji Yoga Pratama

NIM. 2008096059