

SKRIPSI
PERANCANGAN SISTEM PEMBUKUAN DAN KASIR BERBASIS
WEB MENGGUNAKAN FRAMEWORK CODEIGNITER 4 (STUDI
KASUS TOKO SINAR SETUJU PALOPO)

“Diajukan untuk memenuhi salah satu persyaratan memperoleh Gelar Sarjana Komputer
pada Program Studi Informatika”



Disusun oleh :

NUR WILDAYANI
IK.22.11.023

FAKULTAS ILMU KOMPUTER
UNIVERSITAS MEGA BUANA PALOPO

2024

HALAMAN PERSETUJUAN

PROPOSAL SKRIPSI

**PERANCANGAN SISTEM PEMBUKUAN DAN KASIR BERBASIS WEB
MENGUNAKAN FRAMEWORK CODEIGNITER 4 (STUDI KASUS TOKO
SINAR SETUJU PALOPO)**

Diususikan oleh

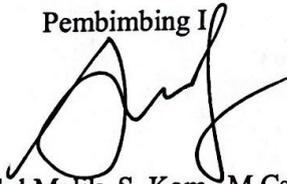


Nur Wildayani Kahar
IK.22.11.023

Telah di setujui

Pada tanggal 2024

Pembimbing I



Abdul Malik, S. Kom., M.Cs
NIDN : 0910049004

Pembimbing II



Muh. Hajar Harike, S. Kom., M.Si.
NIDN : 0912069201

HALAMAN PENGESAHAN

SKRIPSI

**PERANCANGAN SISTEM PEMBUKUAN DAN KASIR BERBASIS WEB
MENGUNAKAN FRAMEWORK CODEIGNITER 4 (STUDI KASUS TOKO
SINAR SETUJU PALOPO)**

Telah dipersiapkan dan disusun oleh

**NUR WILDAYANI KAHAR
NIM : IK.22.11.023**

Telah dipertahankan didepan Dewan Penguji
pada tanggal 8 Juni 2024.

Susunan Dewan Penguji

Pembimbing I



Muh. Hajar Harike, S. Kom., M.Si.
NIDN. 0912069201

Ketua Penguji



Abdul Malik, S. Kom., M.Cs
NIDN. 0910049004

Pembimbing II



M. Hasanuddin, S. Kom., M.Kom.
NIDN. 0927098905

Skripsi ini telah diterima sebagai salah satu persyaratan
untuk memperoleh gelar *Sarjana Komputer*
Tanggal, 8 Juni 2024

Dekan Fakultas Ilmu Komputer
Universitas Mega Buana Palopo



Abdul Malik, S. Kom., M.Cs
NIDN. 0910049004

PERNYATAAN BEBAS PLAGIASI

Saya yang bertanda tangan di bawah ini :

Nama : Nur Wildayani Kahar
NIM : IK. 22.11.023
Tahun terdaftar : 2022
Program Studi : S1 Informatika
Fakultas : Ilmu Komputer
Universitas : Universitas Mega Buana Palopo

Menyatakan bahwa dalam dokumen ilmiah Skripsi ini tidak terdapat bagian dari karya ilmiah lain yang telah diajukan untuk memperoleh gelar akademik di suatu Lembaga Pendidikan Tinggi, dan juga tidak terdapat karya atau pendapat yang pernah ditulis atau diterbitkan oleh orang/ lembaga lain, kecuali yang secara tertulis disitasi dalam rbitkan oleh orang/ lembaga lain, kecuali yang secara tertulis disitasi dalam dokumen ini dan disebutkan sumbernya secara lengkap dalam daftar Pustaka.

Dengan demikian saya menyatakan bahwa dokumen ilmiah ini bebas dari unsur-unsur plagiasi dan apabila dokumen ilmiah Skripsi ini di kemudian hari terbukti merupakan plagiasi dari hasil karya penulis lain dan/atau dengan sengaja mengajukan karya atau pendapat yang merupakan hasil karya penulis lain, maka penulis bersedia menerima sanksi akademik dan/atau sanksi hukum yang berlaku.

Palopo, 08 Juni 2024



Nur Wildayani Kahar
IK. 22.11.023

HALAMAN PERSEMBAHAN

Dengan segala puji syukur kepada Tuhan yang Maha Esa dan atas dukungan dan doa dari orang-orang tercinta, akhirnya skripsi ini dapat dirampungkan dengan baik dan tepat pada waktunya. Oleh karena itu, dengan rasa bangga dan bahagia saya ucapkan rasa syukur dan terimakasih saya kepada:

1. Tuhan YME, karena hanya atas izin dan karuniaNya maka skripsi ini dapat dibuat dan selesai pada waktunya. Puji syukur yang tak terhingga pada Tuhan penguasa alam yang meridhoi dan mengabulkan segala doa.
2. Orang tua tersayang Bapak Abdul Kahar dan Ibu Arniati yang telah memberikan dukungan moril maupun materi serta doa yang tiada henti untuk kesuksesan saya, karena tiada kata seindah lantunan doa dan tiada doa yang paling khusuk selain doa yang terucap dari orang tua. Ucapan terimakasih saja takkan pernah cukup untuk membalas kebaikan orang tua, karena itu terimalah persembahan bakti dan cinta ku untuk kalian bapak ibuku.
3. Adik-adik tersayang (St. Nahdatillah, St. Zahwa Muhliza dan Muh. Magrifatullah) terima kasih atas segala dukungan, bantuan dan pengertiannya selama saya (penulis) kuliah hingga menyelesaikan penyusunan skripsi ini.
4. Bapak dan Ibu dosen pembimbing, penguji dan pengajar, yang selama ini telah tulus dan ikhlas meluangkan waktunya untuk menuntun dan mengarahkan saya, memberikan bimbingan dan pelajaran yang tiada ternilai harganya, agar saya menjadi lebih baik. Terima kasih banyak Bapak dan Ibu dosen, jasa kalian akan selalu terpatrit di hati.
5. Rekan-rekan kantor saya, yang senantiasa memberikan dukungan, semangat, senyum dan doanya untuk keberhasilan ini, cinta kalian adalah memberikan kobaran semangat yang menggebu, terimakasih dan sayang ku untuk kalian.
6. Sahabat dan Teman Tersayang, tanpa semangat, dukungan dan bantuan kalian semua tak kan mungkin aku sampai disini, terimakasih untuk canda tawa, tangis, dan perjuangan yang kita lewati bersama dan terimakasih untuk kenangan manis yang telah mengukir selama ini. Dengan perjuangan dan kebersamaan kita pasti bisa! Semangat!!!

KATA PENGANTAR

Assalamu'alaikum Warahmatullahi Wabarokatuh

Alhamdulillah Rabbil 'Alamin. Segala puji dan syukur penulis panjatkan kepada Allah Subhanau wa Ta'ala karean atas rahmat dan karunia-Nya penulis dapat menyelesaikan Skripsi yang berjudul " Perancangan Sistem Pembukuan Dan Kasir Berbasis Web Menggunakan Framework Codeigniter 4 (Studi Kasus Sinar Setuju Palopo)". Shalawat serta salam semoga selalu tercurahkan kepada Rasulullah Shalallahu'alaihi wassalam yang telah membimbing umatnya dari zaman kegelapan ke zaman terang menderang seperti sekarang ini.

Skripsi ini disusun sebagai syarat yang harus dipenuhi menyelesaikan studi jenjang strata satu (S1) di Program Studi Teknik Informatika Universitas Mega Buana Palopo. Skripsi ini tidak akan berhasil tanpa peran serta dukungan dari berbagai pihak, yang senantiasa memberikan saran, dan motivasi yang sangat membantu untuk menyelesaikan skripsi ini, mengingat pengetahuan dan pengalaman penulis yang terbatas. Oleh karena itu, ucapan terimakasih disampaikan kepada:

1. Allah SWT yang telah mencurahkan rahmat dan hidayahnya hingga detik ini.
2. Kedua orang tua saya yaitu Bapak Abdul Kahar dan Ibu Arniati yang senantiasa memanjatkan do'a, mencurahkan kasih sayang, dan memberikan dukungan berupa moril maupun materil hingga menjadi penyemangat dalam menyelesaikan skripsi ini.
3. Bapak M. Hajar Harike, S. Kom, MM., dan Bapak M. Hasanuddin, S. Kom., M.Kom. selaku dosen pembimbing Penulis. Terimakasih karena telah banyak meluangkan waktu, tenaga, dan kesabaran dalam membimbing. Serta kebijaksanaan dalam memberikan saran dan nasihat yang sangat bermanfaat bagi penulis dalam mengerjakan skripsi ini.
4. Seluruh dosen dan staff Teknik Informatika yang telah memberikan pendidikan serta ilmu yang bermanfaat bagi penulis selama mengampu pendidikan di Universitas Mega Buana Palopo.
5. Semua pihak yang telah membantu baik secara langsung maupun tidak langsung yang tidak bisa penulis sebutkan satu persatu.

Keterbatasan kemampuan, pengetahuan dan pengalaman penulis dalam pembuatan skripsi ini masih jauh dari kata sempurna. Untuk itu penulis akan selalu menerima segala masukan yang ditujukan untuk menyempurnakan skripsi ini. Akhir kata semoga Allah SWT. membalas segala kebaikan yang penulis telah terima dan mengharapkan semoga skripsi ini dapat bermanfaat serta menambah wawasan pengetahuan baik bagi penulis sendiri maupun bagi pihak yang membutuhkan.

Skripsi ini masih jauh dari sempurna, oleh karena itu kritik dan saran yang bersifat membangun sangat kami harapkan guna perbaikan di masa yang akan datang. Semoga skripsi ini dapat memberikan manfaat dan menjadi salah satu sumbangan yang bermanfaat dalam pengembangan ilmu pengetahuan khususnya dalam bidang teknologi informasi.

Akhir kata, kami mohon maaf apabila terdapat kekurangan dalam penulisan skripsi ini.

Wassalamu`alaikum Warahmatullahi Wabarakatuh.

Palopo, 8 Juni 2024

Nur Wildayani Kahar

DAFTAR ISI

| | |
|--|------|
| HALAMAN PERSETUJUAN..... | ii |
| HALAMAN PENGESAHAN | iii |
| HALAMAN PERSEMBAHAN | v |
| KATA PENGANTAR | vi |
| DAFTAR ISI..... | viii |
| DAFTAR TABEL..... | x |
| INTISARI | xii |
| <i>ABSTRAC</i> | xiii |
| BAB I PENDAHULUAN | 1 |
| 1.1 Latar Belakang | 1 |
| 1.2 Rumusan Masalah | 1 |
| 1.3 Batasan Masalah..... | 2 |
| 1.4 Tujuan Penelitian..... | 2 |
| 1.5 Manfaat Penelitian..... | 2 |
| 1.6 Metode Penelitian..... | 2 |
| 1.6.1 Pengumpulan Data..... | 2 |
| 1.6.2 Metode Penelitian/Pengembangan Sistem | 3 |
| 1.7 Sistematika Penulisan | 5 |
| BAB II KAJIAN PUSTAKA | 7 |
| 2.1 Studi Literatur | 7 |
| 2.2 Dasar Teori..... | 10 |
| 2.2.1 Sistem..... | 10 |
| 2.2.2 Pembukuan | 10 |
| 2.2.3 Sistem Pembukuan..... | 10 |
| 2.2.4 Kasir | 11 |
| 2.2.5 Website | 11 |
| 2.2.6 <i>Framework Codeigniter 4</i> | 13 |
| 2.2.7 <i>Metode Rapid Application Development (RAD)</i> | 13 |
| 2.2.8 <i>MySQL</i> | 14 |
| 2.2.9 <i>HTML</i> | 15 |
| 2.2.10 <i>XAMPP</i> | 15 |
| 2.2.11 <i>Hypertext Preprocessor (PHP)</i> | 15 |
| 2.2.12 <i>Flowchart</i> | 15 |
| 2.2.13 <i>Unified Modelling Language (UML)</i> | 16 |
| 2.2.14 <i>Black Box Testing</i> | 22 |
| 2.2.15 <i>White Box Testing</i> | 23 |
| 2.2.16 Studi Kasus Toko Sinar Setuju Palopo..... | 24 |
| 2.3 Kerangka Pikir..... | 24 |
| BAB III ANALISIS DAN RANCANGAN SISTEM | 25 |
| 1.1 Analisis Sistem..... | 25 |
| 1.1.1 Analisis Sistem yang berjalan..... | 25 |
| 1.1.2 Sistem yang diusulkan..... | 25 |
| 1.1.3 Analisis Kebutuhan Sistem..... | 26 |
| 1.4 Rancangan Model Sistem | 29 |
| 1.4.1 Perancangan <i>UML (Unified Modeling Language)</i> | 29 |

| | |
|--|----|
| 1.5 Rancangan Pengujian Sistem | 37 |
| 1.6 Perancangan <i>Database</i> | 38 |
| 1.7 Perancangan Interface..... | 40 |
| BAB IV IMPLEMENTASI DAN PENGUJIAN | 44 |
| 4.1 Implementasi Sistem..... | 44 |
| 4.1.1 Batasan Implementasi | 44 |
| 4.1.2 Lingkungan Operasional | 44 |
| 4.1.3 Implementasi <i>Database</i> | 45 |
| 4.1.4 Implementasi Program | 46 |
| 4.2 Hasil Pengujian | 50 |
| 4.2.1 Hasil Pengujian Black Box..... | 50 |
| 4.2.2 Hasil Pengujian <i>White Box</i> | 52 |
| BAB V PENUTUP | 56 |
| 5.1 Kesimpulan | 56 |
| 5.2 Saran | 57 |
| DAFTAR PUSTAKA | 58 |

DAFTAR TABEL

| | |
|---|----|
| Tabel 2. 1 Studi Literatur | 9 |
| Tabel 2.2 <i>Simbol-Simbol Flowchart (Ii & Pustaka, 2002)(Ii & Pustaka, 2002)</i> | 16 |
| Tabel 2.3 <i>Simbol Use Case Diagram</i> | 17 |
| Tabel 2.4 <i>Simbol-Simbol Class Diagram (Indra Sukma, 2022)</i> | 18 |
| Tabel 2.5 <i>Simbol Sequncial Diagram</i> | 20 |
| Tabel 2.6 <i>Simbol Activity Diagram</i> | 21 |
| Tabel 2. 7 Simbol-Simbol Entity Relationship Diagram..... | 21 |
| Tabel 3.1 Sekenario <i>Use Case</i> Kategori | 30 |
| Tabel 3. 2 Sekenario <i>Use Case</i> Unit..... | 31 |
| Tabel 3.3 Sekenario <i>Use Case</i> Item | 32 |
| Tabel 3.4 Sekenario <i>Use Case Input</i> Penjualan Admin..... | 33 |
| Tabel 3.5 Sekenario <i>Use Case</i> Cetak Resi Admin | 33 |
| Tabel 3.6 Sekenario <i>Use Case Input</i> Penjualan Kasir | 34 |
| Tabel 3.7 Sekenario <i>Use Case</i> Cetak Resi Kasir | 34 |
| Gambar 4. 1 Implementasi Database..... | 45 |
| Gambar 4.2 Tampilan Halaman <i>Login</i> | 46 |
| Gambar 4.3 Tampilan Halaman <i>Dashboard</i> | 47 |
| Gambar 4.4 Tampilan Halaman Kategori | 47 |
| Gambar 4. 5 Tampilan Halaman Unit..... | 48 |
| Gambar 4.6 Tampilan Halaman Item | 48 |
| Gambar 4.7 Tampilan Halaman Penjualan | 49 |
| Gambar 4.8 Tampilan Halaman Penjualan setelah diisi | 49 |
| Gambar 4.9 Tampilan Halaman Resi..... | 50 |
| Gambar 4.10 <i>Source Code Validasi Login</i> | 53 |
| Gambar 4.11 <i>Flowgraph Form Login</i> | 54 |

DAFTAR GAMBAR

| | |
|---|----|
| Gambar 2. 1 Tahapan Rapid Application Development | 14 |
| Gambar 2.2 Black-Box Testing..... | 22 |
| Gambar 2.3 White-Box Testing | 23 |
| Gambar 2.4 Kerangka Pikir | 24 |
| Gambar 3.1 Analisis Sistem yang berjalan | 25 |
| Gambar 3.2 Analisis Sistem yang diusulkan | 26 |
| Gambar 3.3 Use Case Diagram Pembukuan | 29 |
| Gambar 3.4 Activity Diagram Data Barang..... | 35 |
| Gambar 3.5 Activity Diagram Penjualan..... | 36 |
| Gambar 3.6 Activity Diagram Cetak Resi | 36 |
| Gambar 3.7 Class Diagram Penjualan | 37 |
| Gambar 3.8 Interface Hakaman Login | 40 |
| Gambar 3.9 Interface Halaman Dashboard..... | 41 |
| Gambar 3.10 Interface Halaman Kategori..... | 41 |
| Gambar 3.11 Interface Halaman Unit | 42 |
| Gambar 3.12 Interface Halaman Item..... | 42 |
| Gambar 3.13 Interface Halaman Penjualan..... | 43 |
| Gambar 3.14 Interface Halaman Penjualan Resi | 43 |
| Gambar 4. 1 Implementasi Database..... | 45 |
| Gambar 4.2 Tampilan Halaman Login | 46 |
| Gambar 4.3 Tampilan Halaman Dashboard | 47 |
| Gambar 4.4 Tampilan Halaman Kategori | 47 |
| Gambar 4. 5 Tampilan Halaman Unit..... | 48 |
| Gambar 4.6 Tampilan Halaman Item | 48 |
| Gambar 4.7 Tampilan Halaman Penjualan | 49 |
| Gambar 4.8 Tampilan Halaman Penjualan setelah diisi | 49 |
| Gambar 4.9 Tampilan Halaman Resi..... | 50 |
| Gambar 4.10 Source Code Validasi Login..... | 53 |
| Gambar 4.11 Flowgraph Form Login | 54 |

Intisari

Dalam era digitalisasi yang berkembang pesat, peran teknologi informasi sangat krusial dalam mendukung operasional bisnis, terutama dalam manajemen pembukuan dan kasir yang efisien. Toko Sinar Setuju Palopo, sebagai salah satu toko retail yang menjual barang konsumen, masih mengandalkan metode manual dalam pencatatan transaksi, menghadapi potensi kesalahan, proses lambat, dan kesulitan dalam analisis data. Oleh karena itu, peneliti mengusulkan solusi dengan merancang sistem pembukuan dan kasir berbasis web menggunakan *Framework CodeIgniter 4*. Diharapkan sistem ini dapat meningkatkan produktivitas, mengurangi kesalahan, dan meningkatkan pengalaman pelanggan. Dalam pengembangan, metode *Extreme Programming* juga diterapkan. *CodeIgniter 4* dipilih karena kemampuannya dalam percepatan pengembangan, struktur yang terorganisir, dan dukungan komunitas yang luas. Melalui studi kasus ini, diharapkan tercipta sebuah sistem yang dapat membantu Toko Sinar Setuju Palopo meningkatkan efisiensi operasional dan memberikan layanan yang lebih baik kepada pelanggan.

Kata kunci : toko, web, *codeigniter*

Abstrac

In the rapidly evolving era of digitalization, the role of information technology is crucial in supporting business operations, particularly in efficient management of accounting and cashier systems. Store Sinar Setuju Palopo, as one of the retail stores selling consumer goods, still relies on manual methods for transaction recording, facing potential errors, slow processes, and difficulties in data analysis. Therefore, researchers propose a solution by designing a web-based accounting and cashier system using Framework CodeIgniter 4. It is anticipated that this system will enhance productivity, reduce errors, and improve customer experience. In development, the Extreme Programming method is also applied. CodeIgniter 4 is chosen for its ability to accelerate development, organized structure, and extensive community support. Through this case study, it is hoped that a system will be created to assist Store Sinar Setuju Palopo in enhancing operational efficiency and providing better service to customers.

Primary key : store, web, codeigniter

BAB I PENDAHULUAN

1.1 Latar Belakang

Dalam era digitalisasi yang terus berkembang, peran teknologi informasi menjadi semakin penting dalam mendukung kegiatan bisnis. Salah satu aspek penting dalam manajemen bisnis adalah sistem pembukuan dan kasir yang efisien dan akurat. Seiring dengan perkembangan teknologi, sistem ini pun berkembang dari yang awalnya manual menjadi berbasis komputer, dan kini beralih ke sistem berbasis web.

Toko Sinar Setuju Palopo, sebagai salah satu toko retail yang bergerak dalam bidang penjualan barang konsumen, memiliki kebutuhan akan sistem pembukuan dan kasir yang handal dan efisien. Saat ini, Toko Sinar Setuju Palopo masih menggunakan metode manual dalam pencatatan pembukuan dan transaksi kasirnya. Hal ini mengakibatkan potensi kesalahan manusia, lambatnya proses pencatatan, serta sulitnya dalam melakukan analisis data secara cepat dan akurat.

Oleh karena itu, untuk meningkatkan efisiensi dan akurasi dalam pengelolaan pembukuan dan kasir, serta untuk memenuhi tuntutan pasar yang semakin kompetitif, diperlukan sebuah sistem pembukuan dan kasir yang modern dan terintegrasi. Sehingga peneliti memberikan alternative dengan melakukan penelitian yang berjudul “**Perancangan Sistem Pembukuan Dan Kasir Berbasis Web Menggunakan *Framework Codeigniter 4***”. Dengan sistem ini, diharapkan Toko Sinar Setuju Palopo dapat meningkatkan produktivitas, mengurangi potensi kesalahan, serta memberikan pengalaman belanja yang lebih baik bagi pelanggan.

Sistem ini akan dirancang berbasis web dan menggunakan metode *Extreme Programming* dan *Framework CodeIgniter 4*. *CodeIgniter 4* dipilih karena kemampuannya dalam mempercepat proses pengembangan, menyediakan struktur yang terorganisir, serta dukungan komunitas yang luas. Dengan memanfaatkan keunggulan *CodeIgniter 4*, diharapkan pengembangan sistem ini dapat dilakukan dengan lebih efisien dan dapat disesuaikan dengan kebutuhan spesifik dari Toko Sinar Setuju Palopo (Julia, 2022).

1.2 Rumusan Masalah

Berdasarkan latar belakang masalah yang telah di uraikan sebelumnya, dapat di rumuskan bahwa permasalahan pada penelitian ini adalah bagaimana

mengimplementasikan sistem pembukuan dan kasir berbasis web menggunakan *Framework CodeIgniter 4*.

1.3 Batasan Masalah

Penulis menentukan batasan dan ruang lingkup permasalahan yang akan diterapkan pada penelitian ini, yaitu :

- a) Proposal ini akan fokus pada perancangan dan implementasi sistem pembukuan dan kasir berbasis web menggunakan *Framework CodeIgniter 4*.
- b) Penelitian akan difokuskan pada Pembukuan dan Kasir studi kasus Toko Sinar Setuju di Palopo.
- c) Sistem akan beroperasi melalui jaringan lokal.

1.4 Tujuan Penelitian

Berdasarkan uraian latar belakang dan rumusan masalah, maka dapat disampaikan tujuan dari penelitian ini adalah sebagai berikut :

- a) Membangun sebuah sistem pembukuan berbasis web yang memungkinkan pemilik toko untuk mengakses informasi keuangan secara online melalui web browser.
- b) Menghadirkan sistem kasir yang terintegrasi dalam platform web, dengan maksud meningkatkan pengelolaan kas agar lebih terstruktur dan akurat.
- c) Menciptakan sistem yang mendukung pertumbuhan bisnis toko dengan menyediakan informasi keuangan yang tepat dan mudah diakses dan lebih efektif.

1.5 Manfaat Penelitian

Manfaat yang diharapkan dari penelitian ini adalah :

1. Bagi Mahasiswa
 - a. Memperluas pengetahuan mahasiswa tentang bidang ilmu teknologi.
 - b. Memanfaatkan data dan temuan untuk mengembangkan proyek akhir.
2. Bagi Toko Sinar Setuju
Pihak pengelola akan mendapatkan aplikasi sistem informasi yang dapat meningkatkan efektivitas penjualan dan mengurangi kerugian secara signifikan.

1.6 Metode Penelitian

1.6.1 Pengumpulan Data

Metode pengumpulan data yang digunakan oleh penulis dalam melakukan pengumpulan data untuk pembuatan tugas akhir adalah :

1. Observasi yaitu data yang diperlukan untuk pembuatan sistem dikumpulkan melalui pengamatan langsung pada objek terkait. Observasi dilakukan di Toko Sinar Setuju untuk memperoleh informasi dan data yang diperlukan.
2. Wawancara untuk memperoleh informasi dan analisis yang akan menjadi acuan dalam pembuatan produk ini, kami menggunakan teknik pengumpulan data melalui wawancara dengan narasumber. Kami melakukan tanya jawab langsung dengan pegawai di Toko Sinar Setuju untuk mendapatkan data yang jelas dan akurat.
3. Studi Literatur melibatkan pencarian referensi teori yang relevan dengan kasus atau permasalahan yang dihadapi. Referensi tersebut berasal dari jurnal-jurnal yang mengkaji topik yang sama dengan permasalahan yang sedang diteliti.

1.6.2 Metode Penelitian/Pengembangan Sistem

Untuk memudahkan pemahaman, berikut adalah penjelasan mengenai tahapan pengembangan perangkat lunak menggunakan Metode RAD :

1. Perencanaan Kebutuhan

Ini adalah tahap awal dalam pengembangan sistem, di mana fokusnya adalah pada identifikasi masalah dan pengumpulan data dari pengguna atau pihak terkait untuk menentukan tujuan akhir sistem dan kebutuhan informasi yang dibutuhkan. Dalam tahap ini, keterlibatan aktif dari kedua belah pihak sangat penting untuk mengidentifikasi kebutuhan yang diperlukan untuk pengembangan sistem.

2. Desain Sistem

Dalam tahap desain sistem, keterlibatan aktif pengguna sangat penting untuk mencapai tujuan, karena pada tahap ini dilakukan proses perancangan dan literasi desain untuk memastikan kesesuaian dengan kebutuhan pengguna yang telah diidentifikasi sebelumnya. Hasil dari tahap ini adalah spesifikasi perangkat lunak yang mencakup struktur umum sistem, organisasi data, dan aspek lainnya.

3. Proses Pengembangan dan Pengumpulan Feedback

Pada tahap ini, desain sistem yang telah disetujui diubah menjadi aplikasi dalam bentuk versi beta hingga versi final. Selama tahap ini, para pengembang terus melakukan pengembangan dan integrasi dengan bagian-

bagian lainnya, sambil memperhatikan umpan balik dari pengguna atau klien. Jika proses berjalan lancar, pengembangan akan melanjutkan ke tahap berikutnya. Namun, jika aplikasi yang dikembangkan belum memenuhi kebutuhan, para pengembang akan kembali ke tahap desain sistem.

4. Implementasi atau Penyelesaian Produk

Ini adalah tahap di mana programmer menerapkan desain sistem yang telah disetujui sebelumnya. Sebelum implementasi, dilakukan pengujian terhadap program untuk mengidentifikasi kesalahan yang mungkin ada dalam sistem yang dikembangkan. Pada tahap ini, umumnya diberikan umpan balik tentang sistem yang telah dibuat dan mendapat persetujuan terkait implementasinya.

Sebagai suatu pendekatan dalam pengembangan perangkat lunak, tentu saja RAD memiliki sejumlah kelebihan.

Saat pengembangan aplikasi, ada 4 langkah yang harus dilalui yaitu :

a. Mengidentifikasi kebutuhan proyek

Dalam pendekatan RAD, langkah awalnya adalah mendefinisikan kebutuhan proyek. Pada tahap ini, tim harus mengidentifikasi kebutuhan proyek secara keseluruhan, yang mungkin tidak terlalu spesifik tetapi cukup umum dan bervariasi. Dari sini, tim akan memprioritaskan kebutuhan mana yang harus dipenuhi terlebih dahulu. Setelah kebutuhan dipastikan, tim akan menyusun rincian lebih lanjut, seperti tujuan yang harus dicapai, jadwal, dan anggaran yang diperlukan. Semua anggota tim juga harus mempertimbangkan potensi masalah selama pengembangan aplikasi dan merencanakan strategi yang sesuai untuk mengatasi tantangan tersebut. Secara keseluruhan, langkah awal ini bertujuan untuk memberikan gambaran umum tentang proyek yang akan dilakukan. Kemudian, tim dapat fokus pada detail yang lebih spesifik seiring berjalannya proses, membuat proses ini sangat fleksibel.

b. Prototyping

Langkah berikutnya adalah pembuatan prototipe. Pengembang akan segera membuat prototipe aplikasi yang diinginkan, lengkap dengan berbagai fitur dan fungsi. Tujuannya sederhana: memastikan bahwa prototipe yang dibuat memenuhi kebutuhan pelanggan. Namun, langkah

ini bisa diulang dan kadang-kadang melibatkan pengguna untuk menguji. Proses ini memungkinkan tim untuk mengeksplorasi potensi kegagalan di masa depan, membantu mengurangi kesalahan dan proses debugging. Selama fase ini, tim pengembangan dapat memastikan bahwa aplikasi yang dihasilkan mudah digunakan, stabil, bebas dari bug, dan memiliki desain yang baik.

c. Rapid Construction dan Feedback

Setelah menentukan jenis aplikasi yang ingin dibuat, pengembang mengubah prototipe menjadi versi beta aplikasi hingga selesai. Proses ini dapat dikatakan sebagai periode RAD yang sangat intens, di mana pengembang terus-menerus terlibat dalam pengkodean aplikasi, pengujian sistem, dan integrasi dengan pihak lain. Untuk mendukung kecepatan dalam proses ini, pengembang menggunakan alat dan kerangka kerja yang sesuai dengan pendekatan RAD. Implementasi dan penyelesaian produk

Di sini, fokus utama pengembang adalah mengatasi kekurangan yang mungkin muncul selama proses pengembangan aplikasi. Tugas ini meliputi peningkatan stabilitas aplikasi, perbaikan antarmuka pengguna, melakukan pemeliharaan rutin, dan menyusun dokumentasi yang diperlukan. Namun, tahap terakhir ini biasanya dilakukan sebelum aplikasi diserahkan kepada klien.

1.7 Sistematika Penulisan

Berdasarkan pedoman penulisan, penyusunan sistematika penulisan dapat dipilih menjadi 5 (Lima) Bab, sebagai berikut:

BAB I PENDAHULUAN

Pada bagian ini penelitian ini meliputi informasi secara umum tentang latar belakang masalah yang dihadapi oleh Toko Sinar Setuju ditempat penelitian dan mengapa harus dipecahkan.

BAB II TINJAUAN PUSTAKA

Pada bab ini membandingkan penelitian-penelitian sebelumnya yang terkait dengan penelitian, menjelaskan secara umum tempat penelitian skripsi dilaksanakan dan menjelaskan semua materi yang digunakan dalam menyusun skripsi.

BAB III ANALISIS DAN RANCANGAN SISTEM

Pada bab ini membahas secara bertahap tentang perencanaan dan analisis data saat pembuatan aplikasi sistem Pembukuan dan Kasir berbasis web menggunakan *Codeigniter 4*.

BAB IV IMPLEMENTASI

Pada bab ini diterangkan cara implementasi sistem yang digunakan untuk mengaplikasikan perancangan, dan uji coba sistem yang dilakukan oleh petugas pada Toko Sinar Setuju untuk mengetahui sejauh mana sistem tersebut berjalan.

BAB V KESIMPULAN DAN SARAN

Pada bab ini menyajikan kesimpulan secara keseluruhan dari penelitian yang dilakukan dan juga terdapat beberapa saran yang diajukan berkaitan dengan penelitian skripsi ini.

DAFTAR PUSTAKA

Pada bagian ini dipaparkan tentang sumber-sumber literatur yang digunakan dalam penyusunan penelitian Skripsi ini.

LAMPIRAN LAMPIRAN

Pada bagian ini berisi lembar lampiran-lampiran yang dibutuhkan dalam penelitian Skripsi ini.

BAB II

KAJIAN PUSTAKA

2.1 Studi Literatur

Penelitian tentang Sistem Pembukuan dan Kasir berbasis web menggunakan *framework Codeigniter 4*, berdasarkan tinjauan pustaka yang dilakukan peneliti *Codeigniter* adalah sebuah kerangka kerja (*framework*) PHP yang dikatakan memiliki kinerja yang lebih cepat daripada kerangka kerja lainnya.

1. Suminten, 2020, melakukan penelitian di Mart Serba Guna, yang masih mengandalkan penginputan data secara manual, menyebabkan antrean pelanggan yang panjang. Sistem manual juga rentan terhadap kesalahan pencatatan dan kesulitan dalam pencarian data, karena setiap kali mencari data penjualan harus dilakukan di buku besar. Selain itu, tidak ada informasi yang jelas tentang jumlah stok barang, sehingga sering kali pemilik toko baru menyadari kehabisan stok saat proses transaksi sedang berlangsung, yang dapat mengecewakan pelanggan. Untuk mengatasi masalah ini, diperlukan pengembangan aplikasi kasir berbasis web yang dapat menyederhanakan manajemen data oleh kasir. Metode pengembangan yang digunakan adalah metode waterfall, yang melibatkan tahap analisis, desain, pengkodean, pengujian, evaluasi, dan pemeliharaan.
2. Zahra Rashifah dan Eko Setia Budi, 2022, melakukan penelitian di 11Eleven adalah sebuah bisnis kuliner minuman yang menghadirkan produk dengan bahan dasar matcha dan milk tea, terletak di Ciputat, Kota Tangerang Selatan. Proses transaksi kasir di toko ini masih menggunakan metode konvensional, di mana transaksi dicatat manual pada kertas nota dan laporan transaksi kemudian disusun satu per satu. Penggunaan sistem manual ini mengakibatkan berbagai masalah seperti kesalahan harga, kesalahan perhitungan pembayaran, kesulitan dalam membaca pesanan, serta menurunkan efektivitas dan efisiensi pelayanan kepada pelanggan. Selain itu, terdapat risiko kehilangan nota transaksi, kerusakan pada nota penjualan, dan kesalahan dalam pembuatan laporan penjualan, menyebabkan pelaporan menjadi tidak efisien. Untuk mengatasi masalah ini, diperlukan penerapan sistem informasi kasir berbasis web. Tujuan dari pengembangan sistem informasi ini adalah untuk mempermudah kegiatan penjualan dan operasional toko 11Eleven. Metode prototipe digunakan dalam

pengembangan sistem informasi kasir berbasis web ini. Pengembangan *website* ini menggunakan *Framework CodeIgniter* dan Bootstrap. Dengan adanya sistem informasi kasir berbasis web ini, diharapkan dapat meningkatkan kemudahan dalam menjalankan kegiatan penjualan di toko 11Eleven.

3. Muchdi Noor Hidayat dan Muhammad Feizal, 2022, melakukan penelitian di Toko Nurhidayah adalah sebuah usaha yang menjual produk-produk pangan, berlokasi di Jl. Nangka Raya No 15 Parung Panjang Bogor. Proses transaksi di toko ini masih dilakukan secara manual, di mana pelanggan datang langsung ke toko untuk berbelanja, kemudian kasir memberikan struk pembayaran. Kadang-kadang terjadi kesalahan dalam pembuatan nota pembayaran yang bisa merugikan toko. Pendataan barang juga masih dilakukan secara manual oleh pemilik toko, sehingga seringkali terjadi kesalahan pencatatan. Toko Nurhidayah adalah salah satu toko yang bergerak dalam perdagangan sembako. Setiap hari, toko ini mencatat transaksi penjualan dan pembelian, namun belum memiliki sistem yang rinci untuk mengelola data tersebut, yang menyebabkan kesulitan dalam menyusun laporan transaksi dan sering terjadi kesalahan dalam pengolahan informasi.
4. Saputra et al., 2021, melakukan penelitian di perusahaan yang bergerak di bidang jasa percetakan masih mengandalkan proses manual dalam transaksi mereka, yang kurang efisien dan rentan terhadap kesalahan input data. Contohnya, pencatatan pemesanan menggunakan kertas, perhitungan menggunakan kalkulator, dan transaksi penjualan dicatat pada buku. Untuk mengatasi masalah ini, diperlukan sebuah aplikasi kasir berbasis web yang dapat mempermudah proses pemesanan, perhitungan, dan transaksi dengan akurasi. Aplikasi tersebut harus dilengkapi dengan fitur-fitur seperti login, transaksi penjualan, data produk, penambahan produk, pengaturan, dan keluar dari aplikasi. Pengembangan aplikasi ini dilakukan dengan metode waterfall, dan rancangan sistem dijelaskan menggunakan *Unified Modeling Language* (UML). Penggunaan metode *waterfall* memudahkan proses pengembangan aplikasi karena memiliki tahapan-tahapan yang terstruktur.
5. Julia, 2022, melakukan penelitian Sistem Informasi kasir dengan barcode cukup dibutuhkan oleh grosir atau supermarket yang dibutuhkan oleh masyarakat dengan cepat, akurat dan terintegrasi. Metodologi yang digunakan adalah UML

(*Unified Modeling Language*). Dengan menggunakan bahasa pemrograman PHP dengan Framework CodeIgniter dan database MySQL, sistem informasi kasir dengan fitur barcode telah dirancang untuk mengelola data secara berbasis web. Sistem ini memungkinkan produk atau barang dapat dengan cepat diproses dalam transaksi penjualan tanpa perlu menunggu waktu yang lama.

Tabel 2. 1 Studi Literatur

| No. | Referensi | Metode | Data set/Source | Keterangan | Perbedaan Penelitian |
|-----|-----------------------------|--|---|--|---|
| 1. | (Suminten, 2020) | <i>waterfall</i> | Mart Serbaguna Blora | Mempermudah pengelolaan data yang dilakukan kasir berbasis web dengan metode waterfall. | Metode yang digunakan berbeda. |
| 2. | (Rashifah & Budi, 2022) | <i>prototype</i> | Toko 11 Eleven | Menggunakan model prototype agar lebih mudah dalam pengembangan dan perancangan aplikasi. | Data dan metode yang digunakan berbeda. |
| 3. | (Muchdi Noor Hidayat, 2022) | <i>observasi, waterfall</i> | Toko Nurhidayah Parung Panjang | Metode yang digunakan para pengembang sistem membantu dalam mendefinisikan secara rinci kebutuhan sistem yang sesuai dengan keinginan klien. | Metode dan data yang diteliti berbeda. |
| 4. | (Saputra et al., 2021) | <i>waterfall</i> | Jasa Percetakan | Metode yang dilakukan secara bertahap dan meminimalisir terjadinya kesalahan. | Metode yang digunakan berbeda. |
| 5. | (Julia, 2022) | <i>UML (Unified Modeling Language)</i> | Supermarket, minimarket dan toko grosiran | Codeigniter merupakan framework yang diklaim memiliki eksekusi tercepat dibanding framework lainnya. | Data yang diteliti berbeda. |

2.2 Dasar Teori

2.2.1 Sistem

Sistem dapat dijelaskan sebagai sekelompok objek yang saling terkait dan berintegrasi atau bekerja bersama-sama, di mana hubungan antar objek tersebut membentuk satu kesatuan yang direncanakan untuk mencapai tujuan tertentu (Julia, 2022).

Pada dasarnya, sistem memberikan sejumlah keuntungan dalam proses pengambilan keputusan dan pemahaman tentang interaksi antarlingkungan yang berbeda-beda, yang secara bersama-sama berperan untuk mencapai tujuan spesifik. Sistem dapat dianggap sebagai sebuah entitas yang terdiri dari berbagai komponen, seperti manusia, teknologi, lingkungan, dan infrastruktur pendukung lainnya, yang saling terkait dan bekerja sama untuk mencapai tujuan yang telah ditetapkan (Suminten dan Ita Dewi Sintawati, 2020).

Sutabri dalam Suminten (2020) mendefinisikan sistem informasi sebagai suatu sistem yang berada di dalam suatu organisasi, yang menghubungkan kebutuhan pengolahan transaksi harian untuk mendukung fungsi operasional manajerial organisasi dengan kegiatan strategis organisasi untuk menghasilkan laporan-laporan yang dibutuhkan oleh pihak luar.

2.2.2 Pembukuan

Pembukuan adalah proses mencatat dan mengelola data keuangan di dalam suatu perusahaan. Ini mencakup pencatatan segala transaksi keuangan seperti pengeluaran, pemasukan, dan perubahan pada akun-akun tertentu. Tujuan pembukuan adalah memberikan catatan yang tepat dan terperinci tentang aktivitas keuangan perusahaan, yang nantinya digunakan untuk berbagai keperluan seperti pengambilan keputusan, pelaporan keuangan, dan pemantauan kinerja keuangan. Dengan kata lain, pembukuan membantu dalam mengawasi dan mengontrol keuangan suatu perusahaan.

2.2.3 Sistem Pembukuan

Definisi Sistem Pembukuan adalah proses dokumentasi keuangan yang mencakup mencatat semua arus keuangan, termasuk pengeluaran, pemasukan, dan transaksi keuangan lainnya yang terjadi dalam suatu perusahaan atau organisasi bisnis. Tujuan Sistem Pembukuan yaitu memberikan data keuangan

yang tepat dan dapat dipercaya untuk digunakan dalam pengambilan keputusan manajerial, pelaporan keuangan, dan evaluasi kinerja keuangan.

2.2.4 Kasir

Kasir adalah orang bertugas menerima pembayaran dari pelanggan atas barang atau jasa yang dibeli. Tugas utama kasir mencakup menghitung jumlah pembayaran yang akurat, menerima uang tunai atau pembayaran elektronik, memberikan kembalian jika diperlukan, dan mencatat transaksi penjualan. Selain itu, tanggung jawab kasir juga meliputi menjaga ketertiban di area kas dan memberikan layanan pelanggan yang baik. Dengan demikian, kasir memiliki peran yang signifikan dalam menjaga kelancaran proses penjualan serta memastikan pengalaman positif bagi para pelanggan. Fungsi Kasir dalam melaksanakan transaksi penjualan, menerima pembayaran dari pelanggan, menyalurkan uang tunai, dan mencatat semua transaksi keuangan yang terjadi.

2.2.5 Website

Website adalah sistem informasi yang terdiri dari sejumlah besar halaman web yang saling terhubung dan dapat diakses melalui internet. Halaman web ini dapat berisi teks, gambar, audio, video, dan berbagai jenis konten multimedia lainnya. Web memungkinkan pengguna untuk mengakses informasi, berinteraksi dengan konten, berkomunikasi, berkolaborasi, dan melakukan berbagai aktivitas lainnya secara daring. Ini telah menjadi salah satu sumber utama informasi dan komunikasi di era digital saat ini, memungkinkan akses ke berbagai layanan dan sumber daya dari mana saja di dunia (Student et al., 2021).

Sebuah situs web adalah kumpulan halaman web yang terhubung satu sama lain dan biasanya dihosting pada server yang sama. Situs web ini berisi informasi yang disediakan oleh individu, kelompok, atau organisasi. Umumnya, sebuah situs web dapat diakses melalui internet atau jaringan area lokal (LAN) menggunakan alamat internet yang dikenal sebagai *Uniform Resource Locator* (URL). Seluruh koleksi situs web yang dapat diakses secara publik di internet disebut sebagai *World Wide Web* (WWW). Sebuah halaman web awalnya dibuat sebagai berkas teks biasa yang diatur dengan instruksi *HTML*. Berkas ini kemudian diinterpretasikan oleh peramban web dan ditampilkan sebagai halaman di layar komputer. Pengguna mengakses halaman-halaman web ini melalui protokol komunikasi jaringan yang disebut *Hypertext Transfer Protocol*

(HTTP). Untuk meningkatkan keamanan dan privasi, situs web dapat mengimplementasikan protokol *Hypertext Transfer Protocol Secure (HTTPS)*.

Website memiliki berbagai fungsi tergantung pada kebutuhan dan tujuan penggunaannya. Berikut adalah beberapa fungsi umum dari *website*:

- a. Salah satu fungsi utama banyak *website* adalah menyediakan informasi kepada pengunjungnya. Informasi ini dapat berupa deskripsi produk atau layanan, profil perusahaan, berita, artikel, panduan, dan konten lainnya.
- b. *Website* digunakan sebagai alat pemasaran untuk mempromosikan produk, layanan, merek, atau acara. Ini dapat mencakup penempatan iklan, konten promosi, testimonial pelanggan, dan fitur penawaran khusus.
- c. Interaksi dengan Pelanggan: *Website* dapat berfungsi sebagai platform untuk berinteraksi dengan pelanggan. Ini termasuk formulir kontak, layanan obrolan langsung, forum diskusi, dan fitur komentar pada konten.
- d. Banyak *Website* berperan sebagai toko online di mana pengguna dapat membeli produk atau layanan secara langsung. Ini melibatkan proses penanganan pembayaran, pengelolaan pesanan, dan pengiriman barang.
- e. *Website* juga berfungsi sebagai sumber belajar dan informasi. Ini dapat berupa situs pendidikan, tutorial, panduan, e-book, dan konten edukatif lainnya.
- f. Sejumlah besar *website* berfungsi sebagai blog, di mana pemilik atau penulis konten dapat berbagi pemikiran, pengalaman, informasi, atau cerita dengan audiens mereka.
- g. *Website* sering digunakan sebagai platform untuk berbagi konten melalui media sosial. Ini memungkinkan pengguna untuk berbagi konten yang menarik dengan teman-teman mereka.
- h. *Website* sering digunakan oleh individu atau profesional kreatif untuk menampilkan portofolio karya mereka. Ini dapat berupa portofolio seni, desain grafis, fotografi, atau proyek-proyek profesional lainnya.
- i. Beberapa *website* bertujuan untuk membangun komunitas online di sekitar minat atau topik tertentu. Ini dapat berupa forum diskusi, grup Facebook, atau situs jejaring sosial khusus.

2.2.6 *Framework Codeigniter 4*

Framework adalah ide dan keunggulan yang digunakan dalam mengembangkan aplikasi perangkat lunak. *Codeigniter 4* adalah sebuah framework *PHP* yang diklaim memiliki kinerja eksekusi yang lebih cepat daripada framework lainnya. *Framework* ini mempunyai konsep *MVC*, yang merupakan singkatan dari Model, View, Controller, yang mengatur alur kerja dalam framework ini, khususnya dalam konteks *CodeIgniter*.

Struktur dan Komponen Codeigniter 4 yaitu penguraian mengenai struktur folder, konsep model-view-controller (MVC), pengaturan rute, serta berbagai fitur tambahan yang ada dalam CodeIgniter 4.

2.2.7 *Metode Rapid Application Development (RAD)*

Rapid Application Development (RAD) adalah salah satu metode populer dalam pengembangan aplikasi saat ini. Pendekatan ini menekankan pembuatan aplikasi melalui prototyping, iterasi, dan umpan balik yang berulang. *Rapid Application Development (RAD)* adalah model proses pengembangan perangkat lunak yang menekankan pada proses pengembangan yang singkat. RAD merupakan versi yang disesuaikan secara cepat dari model waterfall, dengan menerapkan pendekatan pembangunan komponen. RAD merupakan gabungan dari berbagai teknik prototyping dan metode pengembangan joint application untuk mempercepat pengembangan sistem. Dari definisi konsep RAD ini, dapat disimpulkan bahwa pengembangan aplikasi dengan menggunakan metode RAD dapat dilakukan dalam waktu yang relatif lebih cepat. Sesuai dengan metodologi RAD, tahap-tahapan pengembangan dapat dijelaskan sebagai berikut (Ii & Teori, 2018)

1. *Requirements Planning*

Pada tahap ini, pengguna dan analis bertemu untuk mengidentifikasi tujuan dari aplikasi atau sistem. Tahap ini melibatkan koordinasi antara kedua belah pihak untuk mencapai kesepakatan. Perencanaan persyaratan juga membutuhkan partisipasi pengguna dari berbagai level organisasi. Tahap ini berfokus pada pemecahan masalah bisnis yang dihadapi oleh organisasi.

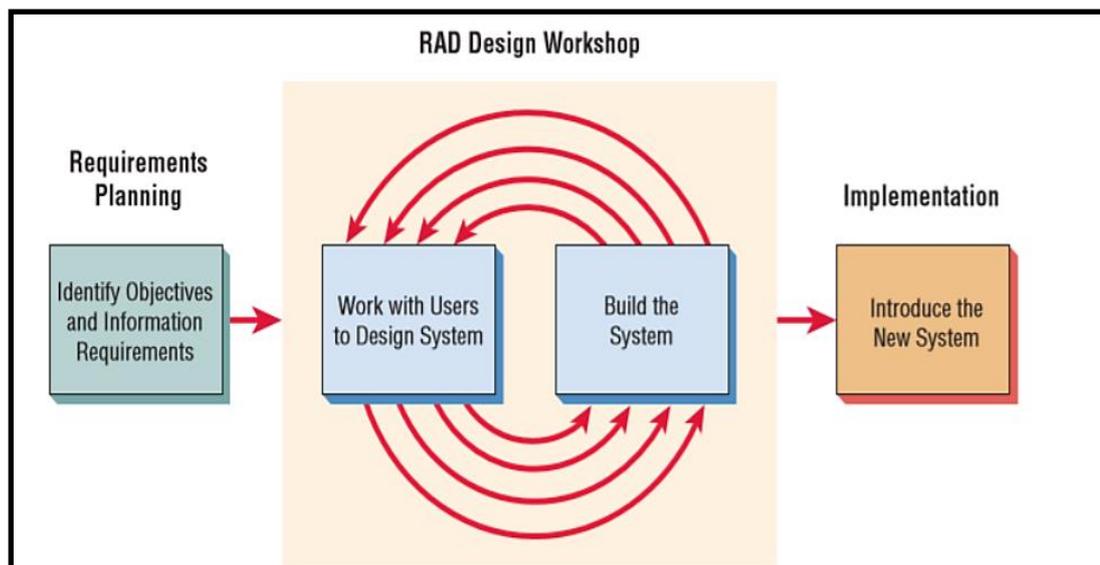
2. *Design Workshop*

Pada tahap ini, semua kegiatan dalam arsitektur direncanakan dan pemahaman atas masalah ditingkatkan berdasarkan analisis yang telah

dilakukan. Aktivitas dilakukan dengan merancang proses bisnis, model struktural, model perilaku, dan desain antarmuka pengguna. Hasilnya berupa pemodelan, perancangan basis data, dan desain antarmuka.

3. *Implementation*

Tahap implementasi melibatkan penerapan sistem dan metode pemrograman sesuai dengan kebutuhan sistem. Ini termasuk implementasi basis data dan penulisan program. Aktivitas yang dilakukan mencakup penentuan lingkungan implementasi perangkat lunak, perancangan basis data, pemrograman, dan antarmuka. Hasilnya adalah basis data dan kode program. Dengan menggunakan pendekatan ini, aplikasi dapat dikembangkan dan diperbaiki dengan cepat, sehingga sangat responsif terhadap kebutuhan dan perkembangan dunia digital yang bergerak sangat cepat. RAD berbeda secara signifikan dari metode pengembangan lainnya, seperti model *Waterfall* yang dianggap kurang efisien. Model *Waterfall* memerlukan perencanaan aplikasi secara menyeluruh sebelum pelaksanaan, yang seringkali memakan waktu lama sebelum aplikasi dapat diluncurkan.



Gambar 2. 1 *Tahapan Rapid Application Development*

2.2.8 *MySQL*

MySQL merupakan sistem manajemen database *SQL* atau *DBMS* yang bersifat multithreaded dan multi-pengguna. *MySQL* pada dasarnya merupakan turunan dari salah satu konsep utama basis data untuk pemilihan dan impor data, yang membuatnya mudah dan otomatis digunakan.

2.2.9 HTML

HTML adalah bahasa yang digunakan dalam World Wide Web. Semua dokumen web disusun dalam format *HTML*, termasuk dokumen berbagai format, hyperlink, gambar, multimedia, formulir yang dapat diisi, dan banyak lagi. Meskipun *HTML* bukanlah bahasa pemrograman, sehingga tidak memiliki kemampuan untuk menciptakan fungsionalitas yang dinamis, namun lebih tepatnya *HTML* dapat diklasifikasikan sebagai seperangkat aturan untuk pemformatan dokumen.

2.2.10 XAMPP

XAMPP adalah perangkat lunak yang tersedia secara gratis, kompatibel dengan berbagai sistem operasi, dan merupakan gabungan dari beberapa program. *XAMPP* berperan sebagai server lokal yang berdiri sendiri (localhost), terdiri dari server program Apache *HTTP*, database *MySQL*, dan penerjemah bahasa yang ditulis dalam PHP dan Perl. *XAMPP* merupakan singkatan dari X (merujuk pada sistem operasi apa pun), *Apache*, *MySQL*, *PHP*, Perl. Program ini adalah server web yang user-friendly yang mampu mengakomodasi halaman web dinamis, dan tersedia secara bebas di bawah Lisensi Publik Umum.

2.2.11 Hypertext Preprocessor (PHP)

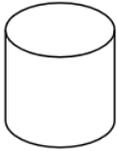
PHP (Hypertext Preprocessor) merupakan sebuah bahasa pemrograman server side scripting yang open source. Sebagai scripting language, PHP menjalankan instruksi pemrograman saat proses runtime. Output dari instruksi ini bervariasi tergantung pada data yang diproses. *PHP* adalah bahasa pemrograman server side, sehingga script *PHP* akan dieksekusi di server. Beberapa server yang sering digunakan bersama dengan *PHP* meliputi *Apache*, *Nginx*, dan *LiteSpeed*.

2.2.12 Flowchart

Flowchart (Diagram Alir) merupakan bagan yang mengarahkan alir di dalam prosedur sistem secara logika. *Flowchart* merupakan cara dalam menjelaskan tahap-tahap pemecahan masalah dengan merepresentasikan simbol-simbol tertentu yang mudah digunakan dan dipahami (Indra Sukma, 2022).

Simbol-simbol yang ada didalam *flowchart* dapat dilihat pada tabel berikut:

Tabel 2.2 *Simbol-Simbol Flowchart (Ii & Pustaka, 2002)(Ii & Pustaka, 2002)*

| No. | Simbol | Nama | Fungsi |
|-----|---|-------------------------|---|
| 1. |  | Proses | Digunakan untuk menunjukkan pengolahan yang dilakukan komputer |
| 2. |  | <i>Manual Operation</i> | Digunakan untuk menunjukkan pengolahan yang tidak dilakukan komputer |
| 3. |  | <i>Decision</i> | Digunakan untuk memilih proses berdasarkan kondisi |
| 5. |  | Terminal | Digunakan untuk memulai atau mengakhiri proses. |
| 6. |  | <i>Offline Storage</i> | Digunakan untuk memberitahukan bahwa data akan disimpan ke suatu media tertentu |
| 7. |  | Manual Input Simbol | Digunakan untuk input data secara manual dengan keyboard |
| 8. |  | <i>Input/Output</i> | Digunakan untuk Input dan Output. |
| 9. |  | Dokumen | Digunakan untuk data masukan dan keluaran dari dokumen |
| 10 |  | Database | Digunakan untuk menyimpan data. |

2.2.13 *Unified Modelling Language (UML)*

UML merupakan suatu bahasa berbasis grafik yang digunakan untuk memvisualisasikan, menspesifikasikan, membangun, dan mendokumentasikan sistem pengembangan perangkat melalui gambaran grafis. *UML* adalah salah satu alat yang berguna dalam pemrograman berbasis objek, dan saat ini, *UML*

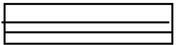
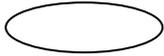
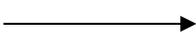
dianggap akan menjadi standar bagi industri pengembangan sistem atau perangkat lunak yang mengadopsi pendekatan berorientasi objek (Harike, 2019).

Ada beberapa macam diagram dalam *Unified Modeling Language (UML)*, yaitu:

1. *Use Case Diagram*

Diagram *Use Case* merupakan representasi abstrak dari interaksi antara sistem dan aktor. Diagram ini digunakan untuk mendeskripsikan bagaimana pengguna atau aktor berinteraksi dengan sistem melalui skenario penggunaan. *Diagram Use Case* sangat penting dalam menyusun kebutuhan sistem, berkomunikasi dengan klien, dan merancang kasus uji untuk fitur-fitur sistem. *Use case* adalah cara untuk menggambarkan bagaimana sistem akan dilihat oleh pengguna, sementara diagram use case membantu dalam komunikasi antara analis, pengguna, dan klien.

Tabel 2.3 *Simbol Use Case Diagram*

| No | Simbol | Nama | Fungsi |
|----|---|--------------------|---|
| 1 |  | <i>Actor</i> | Segala sesuatu yang berinteraksi dengan sistem aplikasi komputer. |
| 2 |  | <i>Class</i> | Pembentukan utama dari sistem berorientasi yang memiliki atribut dan operasional yang sama. |
| 3 |  | <i>Interface</i> | Kumpulan operasi tanpa implementasi dari suatu kelas. |
| 4 |  | <i>Use Case</i> | Menjelaskan yang dilakukan aktor dari sistem untuk mencapai tujuan tertentu. |
| 5 |  | <i>Interaction</i> | untuk menunjukkan baik alir pesan atau informasi antara objek maupun hubungan antar objek. |

| | | | |
|---|---|----------------|---|
| 6 |  | <i>Package</i> | Komentar atau wadah konseptual yang digunakan untuk mengelompokkan elemen-elemen dari sistem yang sedang dibangun, sehingga bisa dibuat model yang lebih sederhana. |
|---|---|----------------|---|

2. Class Diagram

Class adalah dekripsi kelompok obyek-obyek dengan *property*, perilaku (operasi) dan relasi yang sama. Sehingga dengan adanya class diagram dapat memberikan pandangan global atas sebuah sistem. Hal tersebut tercermin dari class-class yang ada dan relasinya satu dengan yang lainnya. Didalam sebuah sistem biasanya memiliki beberapa class diagram. *Class diagram* sangat membantu dalam visualisasi struktur kelas dari suatu system (Harike, 2019).

Tabel 2.4 Simbol-Simbol Class Diagram (Indra Sukma, 2022)

| No. | Simbol | Nama | Fungsi |
|-----|---|--------------------------------|---|
| 1. |  | <i>Class</i> | Kelas pada struktur sistem |
| 2. |  | Antarmuka/ <i>Interface</i> | Sama dengan konsep interface dalam Pemrograman berbasis objek. |
| 3. |  | Generalisasi | Relasi antar kelas dengan makna generalisasi (umum khusus) |
| 4. |  | Asosiasi Berarah | Relasi antar kelas dengan makna kelas yang satu digunakan oleh kelas lain |
| 5. |  | Asosiasi | Hubungan antar kelas dengan makna umum |

| | | | |
|----|---|----------------|---|
| 6. |  | kebergantungan | Hubungan antar kelas dengan makna kebergantungan antar kelas. |
| 7. |  | Agregasi | Hubungan antar kelas dengan makna semua bagian. |

3. *Component Software*

Component software merupakan bagian fisik dari sebuah sistem, karena menetap di komputer tidak berada di benak para analis. Komponen merupakan bagian implementasi software dari sebuah atau lebih class. Komponen dapat berupa *source code*, komponen biner, atau *executable component*. Sebuah komponen berisi informasi tentang *logic class* atau *class* yang diimplementasikan sehingga membuat pemetaan dari sebuah *logical view* ke komponen view. Sehingga komponen diagram dapat merepresentasikan dunia riil yaitu sebuah *component software* yang mengandung *component*, *interface* dan *relationship*.

4. *Deployment Diagram*

Menggambarkan tata letak sebuah system secara fisik, menampilkan bagian-bagian software yang berjalan pada bagian-bagian hardware, menunjukkan hubungan komputer dengan perangkat (*nodes*) satu sama lain dan jenis hubungannya. Didalam sebuah *nodes*, *executeable component* dan *object* yang dialokasikan untuk dapat memperlihatkan sebuah unit perangkat lunak yang telah dieksekusi oleh sebuah node tertentu dan ketergantungan komponen.

5. *State Diagram*

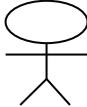
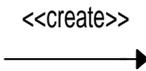
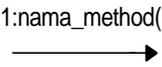
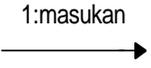
Menggambarkan semua *state* (kondisi). yang dimiliki oleh suatu object dari sebuah class dan keadaan yang menyebabkan state berubah. Kejadian ini dapat berupa *object* lain yang mengirim pesan. *State class* tidak digambarkan untuk semua class, hanya yang mempunyai sejumlah state yang terdefinisi dengan baik dan kondisi class berubah oleh state yang berbeda.

6. *Sequence Diagram*

Sequence Diagram digunakan untuk menggambarkan perilaku pada sebuah scenario. Kegunaannya untuk menunjukkan rangkaian pesan yang

dikirim antara object juga interaksi antara object, sesuatu yang terjadi pada titik tertentu dalam eksekusi sistem. *Sequencediagram* merupakan gambaran secara grafis sebuah skenario yang menunjukkan interaksi antar objek dalam sebuah urutan waktu, apa yang terjadi pertama kali dan apa yang terjadi berikutnya. *Sequencediagram* menggambarkan interaksi antar objek di dalam dan sekitar sistem (termasuk pengguna, tampilan, dan sebagainya), berupa pesan yang digambarkan terhadap waktu. *Sequencediagram* terdiri atas dimensi vertikal (waktu) dan dimensi horizontal (objek-objek yang terkait).

Tabel 2.5 *Simbol Sequncial Diagram*

| No | Simbol | Nama | Fungsi |
|----|---|--------------------------|--|
| 1 |  | <i>Actor</i> | Segala sesuatu yang berinteraksi dengan sistem aplikasi komputer. |
| 2 |  | <i>Garis hidup</i> | Menyatakan kehidupan suatu objek |
| 3 |  | <i>Pesan tipe create</i> | Objek yang lain, arah panah mengarah pada objek |
| 4 |  | <i>Pesan tipe call</i> | Menyatakan suatu objek memanggil operasi yang ada pada objek lain. |
| 5 |  | <i>Pesan tipe send</i> | Menyatakan bahwa suatu objek mengirimkan data/masukan ke objek lain. |

7. *Collaboration Diagram*

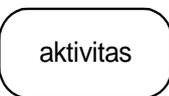
Menggambarkan kolaborasi dinamis seperti *sequence diagram*. Dalam menunjukkan proses pertukaran sebuah pesan, *collaboration diagram* menggambarkan *object* dan hubungannya (mengacu ke konteks). Jika

penekannya pada waktu atau urutan gunakan *sequence diagram*, tapi jika penekanannya pada konteks gunakan *collaboration diagram*.

8. Activity Diagram

Menggambarkan rangkaian aliran dari aktivitas, digunakan untuk mendeskripsikan aktifitas yang dibentuk dalam suatu operasi sehingga dapat juga digunakan untuk aktifitas lainnya seperti *use case* atau interaksi. *Activity* diagram menggambarkan *workflow* (aliran kerja) atau aktivitas dari sebuah sistem.

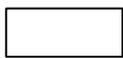
Tabel 2.6 Simbol Activity Diagram

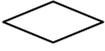
| No | Simbol | Nama | Fungsi |
|----|---|--------------|---|
| 1 |  | Status Awal | Status awal aktifitas sistem sebuah diagram aktivitas memiliki sebuah status awal |
| 2 |  | Aktivitas | Aktivitas yang dilakukan sistem, aktivitas biasanya diawali dengan kata kerja. |
| 3 |  | Percabangan | Asosiasi percabangan dimana jika ada pilihan aktivitas lebih dari satu |
| 4 |  | Status akhir | Status akhir yang dilakukan sistem |

9. Entity Relationship Diagram

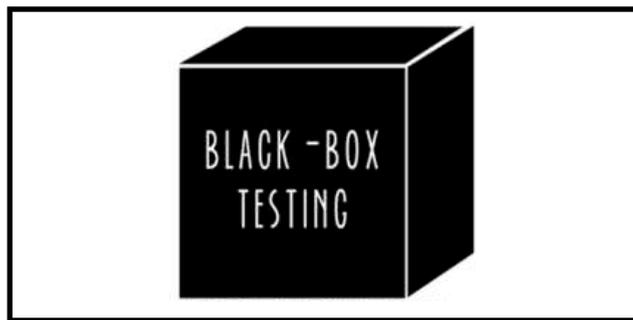
Entity Relationship Diagram atau *ERD* adalah alat perancangan untuk memodelkan suatu basis data. Tujuan dibuatnya *Entity Relationship Diagram* adalah untuk pemodelan yang menunjukkan hubungan antar data yang mempunyai relasi, juga mendokumentasikan data yang ada dengan cara menjelaskan tiap masing-masing data dan relasinya. Simbol *ERD* ditunjukkan pada Tabel 2.7 berikut ini (Indra Sukma, 2022).

Tabel 2. 7 Simbol-Simbol Entity Relationship Diagram

| No | Simbol | Nama | Keterangan |
|----|---|----------------|---|
| 1 |  | <i>Entitas</i> | Suatu objek yang dapat diidentifikasi dalam lingkungan pemakai. |

| | | | |
|---|---|-------------|---|
| 2 |  | Relasi | Menunjukkan adanya hubungan diantara sejumlah entitas yang berbeda. |
| 3 |  | Association | Penghubung antara relasi dengan entitas, relasi dan entitas dengan atribut. |
| 4 |  | Atribut | Mendeskripsikan karakter entitas (atribut yang berfungsi sebagai <i>key</i> yang diberi garis bawah). |

2.2.14 Black Box Testing



Gambar 2.2 Black-Box Testing

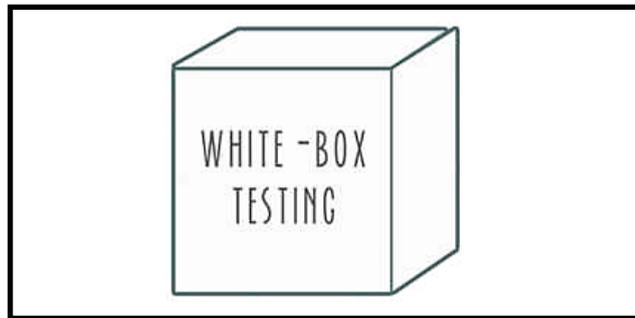
Metode uji "Kotak Hitam" atau "*Black Box*" melibatkan peninjauan fungsi-fungsi aplikasi tanpa memperhatikan cara kerjanya. Penguji hanya memperhatikan apakah aplikasi melakukan apa yang seharusnya dilakukan, tanpa memeriksa struktur atau mekanisme internalnya. Dalam metode ini, penguji memerlukan pemahaman tentang peran dan fungsi sistem tanpa perlu mengetahui detail internalnya. Mereka bertindak seperti pengguna biasa.

Tujuan metode ini adalah untuk memastikan bahwa setelah tahap akhir proyek, perangkat lunak atau aplikasi beroperasi dengan baik dan melayani pengguna dengan efisien. Penguji biasanya mencari fungsi yang hilang atau salah, masalah antarmuka, kinerja yang buruk, kesalahan program, atau masalah struktur data atau akses basis data eksternal.

Kejelasan: Persiapan tes ini cepat karena membutuhkan sedikit pengetahuan tentang sistem. Pembuatan dan pengujian skenario juga cepat karena mengikuti jalur pengguna yang relatif sedikit, tergantung pada kompleksitas sistem.

Ketidakterbukaan: Pendekatan ini mengambil sudut pandang "pengguna" daripada "pengembang". Hasil pengujian bersifat netral: sistem berfungsi atau tidak, tanpa ada ruang bagi preferensi subjektif. Tidak ada kemungkinan kontroversi, seperti pemilihan satu proses daripada yang lain berdasarkan opini pengembang.

2.2.15 *White Box Testing*



Gambar 2.3 *White-Box Testing*

Tes "Kotak Putih" melibatkan peninjauan fungsi aplikasi dan struktur internalnya, proses, dan komponen internal lainnya, bukan hanya fungsinya. Dalam metode ini, semua komponen internal perangkat lunak atau aplikasi diuji melalui kode sumber, yang menjadi fokus utama pengujian.

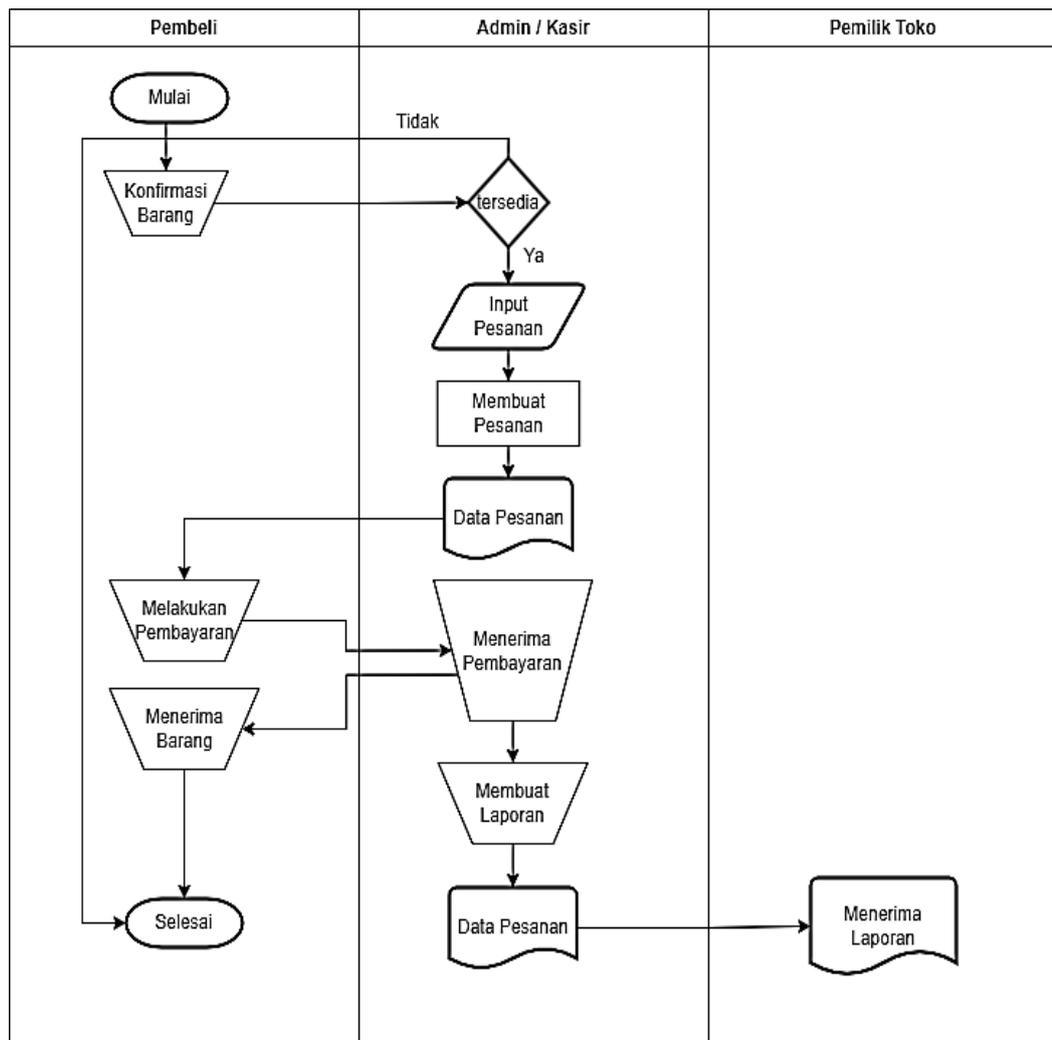
Dengan menggunakan tes "Kotak Putih", pengujian dapat melihat baris kode yang dipanggil untuk setiap fungsi, menguji aliran data, penanganan pengecualian dan kesalahan, ketergantungan sumber daya, serta logika dan kebenaran internal kode. Oleh karena itu, metode ini terutama bermanfaat selama pengembangan aplikasi, meskipun juga dapat dilakukan selama fase-fase lain dari proyek. Metode "Kotak Putih" dapat digunakan untuk tes unit, tes integrasi, dan tes sistem. Pengujian "Kotak Putih" melibatkan penggunaan skenario uji yang dibuat oleh pengujian berdasarkan pemahaman mereka tentang kode sumber dan lingkungan aplikasi. Tujuan dari menguji semua skenario ini adalah untuk memeriksa setiap baris kode, dan memastikan bahwa proses yang dilakukan oleh aplikasi setelah menerima input menghasilkan output yang diharapkan. Pengujian hanya mengevaluasi apakah kode menghasilkan hasil yang sesuai dengan yang diharapkan.

BAB III ANALISIS DAN RANCANGAN SISTEM

1.1 Analisis Sistem

1.1.1 Analisis Sistem yang berjalan

Pada tahap ini, penting untuk merencanakan langkah-langkah yang akan diambil untuk menemukan solusi atas masalah yang ada. Hal ini dilakukan dengan memahami secara mendalam sistem yang sedang diterapkan oleh Toko Sinar Setuju Palopo.

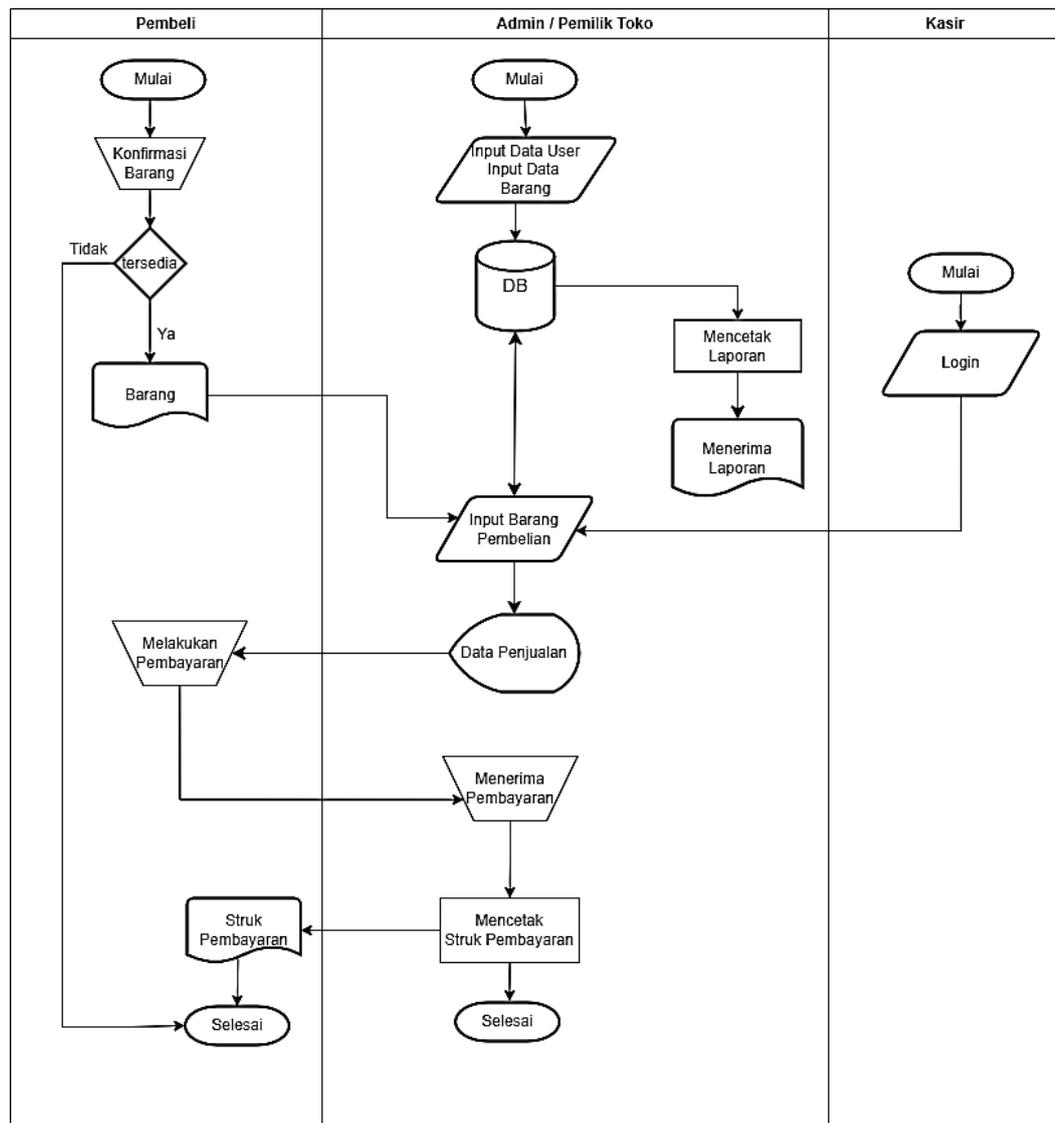


Gambar 3.1 Analisis Sistem yang berjalan

1.1.2 Sistem yang diusulkan

Berdasarkan hasil pengamatan di Toko Sinar Setuju, ditemukan kebutuhan akan perbaikan pada sistem penjualan. Selain itu, diperlukan pengembangan

aplikasi berbasis *web* untuk menangani masalah yang teridentifikasi. Rincian mengenai analisis sistem yang diusulkan dapat ditemukan pada gambar 3.2.



Gambar 3.2 Analisis Sistem yang diusulkan

1.1.3 Analisis Kebutuhan Sistem

Menurut penelitian di Toko Sinar Setuju Palopo, terdapat kebutuhan akan sebuah sistem pembukuan dan kasir berbasis web untuk mengatasi sejumlah masalah penjualan. Platform web ini dirancang untuk menyimpan serta menampilkan berbagai jenis dokumen multimedia seperti teks, gambar, dan animasi. Aplikasi ini menggunakan *protokol HTTP (Hypertext Transfer Protocol)* dan dapat diakses melalui perangkat lunak yang disebut browser.

Website dikelola oleh dua user yaitu admin/pemilik toko dan kasir dan bendahara barang :

a. Kebutuhan Sistem Admin

- 1) Admin/Pemilik Toko dapat melakukan login untuk dapat mengakses web, dengan menggunakan nama pengguna dan kata sandi.
- 2) Dapat mengelola data jenis dan kategori barang, dengan fitur tambah, hapus dan edit.
- 3) Dapat mengelolah data user dengan fitur tambah, hapus dan edit.
- 4) Dapat mengelola data item barang dengan fitur tambah, hapus dan edit.
- 5) Dapat melihat, mengelola data penjualan dan cetak resi penjualan.
- 6) Admin dapat logout setelah menggunakan web.

b. Kebutuhan Sistem Kasir

- 1) Kasir dapat melakukan login untuk dapat mengakses web, dengan menggunakan nama pengguna dan kata sandi.
- 2) Dapat melihat dan mengelola data penjualan dan cetak resi penjualan.
- 3) Kasir dapat logout setelah menggunakan web.

1.2 Deskripsi Data

Data yang digunakan dan diproses dalam penelitian ini yaitu Data User, Data Barang (Kategori, Unit, Item) dan Data Transaksi (Penjualan dan Resi).

1. Data *User*

Data *user* terdiri dari *user* Admin/Pemilik Toko dan *User* Kasir. Data *User* Admin/Pemilik Toko terdiri dari *field Id* Admin/Pemilik Toko, Data Barang (Kategori, Unit, Item), Data Transaksi (Penjualan dan Resi). Sedangkan Data *user* Kasir terdiri dari *field id Kasir*, Data Transaksi (Penjualan dan Resi).

2. Data Barang

Data Barang terdiri dari *field Id* Barang, terdiri dari *field Id* Kategori, *field Id* Unit dan *field Id* Item.

3. Data Transaksi

Data Transaksi terdiri dari *field Id* Transaksi, penjualan (untuk menginput penjualan) dan Resi (menampilkan semua resi yang dapat di cetak ulang).

4. Data Administrator

Administrator terdiri dari *field Id* Administrator, Profile, Pengguna/*User* (yang hanya bisa di ubah oleh Admin/Pemilik Toko) dan *Logout*.

1.3 Arsitektur Sistem

Arsitektur sistem yang dibangun dalam penelitian ini yaitu menentukan bagaimana komponen-komponen tersebut diorganisasikan dan bagaimana mereka saling berinteraksi untuk memastikan kinerja, skalabilitas, keandalan, dan keamanan sistem secara keseluruhan. Arsitektur sistem dalam penelitian ini terdiri dari 3 bagian yaitu *Client Side*, *Server Side* dan *Database*.

1. *Client Side (Frontend)* atau jenis website yang memproses data dan menampilkan halaman *web* melalui *browser* pengguna (*Users*) seperti Antarmuka Pengguna (*User Interface*) yang terdiri dari :
 - a) Desain antarmuka menggunakan HTML, CSS dan *JavaScript*.
 - b) Fitur utama: Data Barang, Data Transaksi dan Administrator.
2. *Server Side (Backend)* atau suatu istilah yang merujuk pada pemrosesan dan pengelolaan data yang terjadi di sisi server dalam sebuah aplikasi *web* atau sistem komputer. Dalam penelitian ini menggunakan *Framework Codeigniter 4* yang terdiri dari :
 - a) *Codeigniter 4* kerangka kerja PHP dengan model MVC (Model, View, Controller) untuk membangun situs web dinamis dengan menggunakan PHP.
 - b) Struktur MVC (*Model-View-Controller*) memisahkan logika bisnis, tampilan, dan pengelolaan data.
3. *Database* untuk data yang disimpan dan dikelola sedemikian rupa sehingga dapat diakses, diperbaharui, dan dikelola dengan mudah, *Database* yang digunakan dalam penelitian ini yaitu menggunakan *MySQL* yang terdiri dari:
 - a) *MySQL* digunakan untuk menyimpan data.
 - b) Struktur *database* relasional dengan tabel utama : *User* (Admin/Pemilik Toko) , *User Kasir*, *Transactions* (Data Transaksi), Data Barang yang meliputi *Categories* (Kategori), *Unit (Satuan)* dan *Item*, Administrator meliputi *Profile*.
 - c) ORM (*Object-Relational Mapping*) *Eloquent* dari *Codeigniter 4* yang digunakan untuk interaksi dengan *database*.

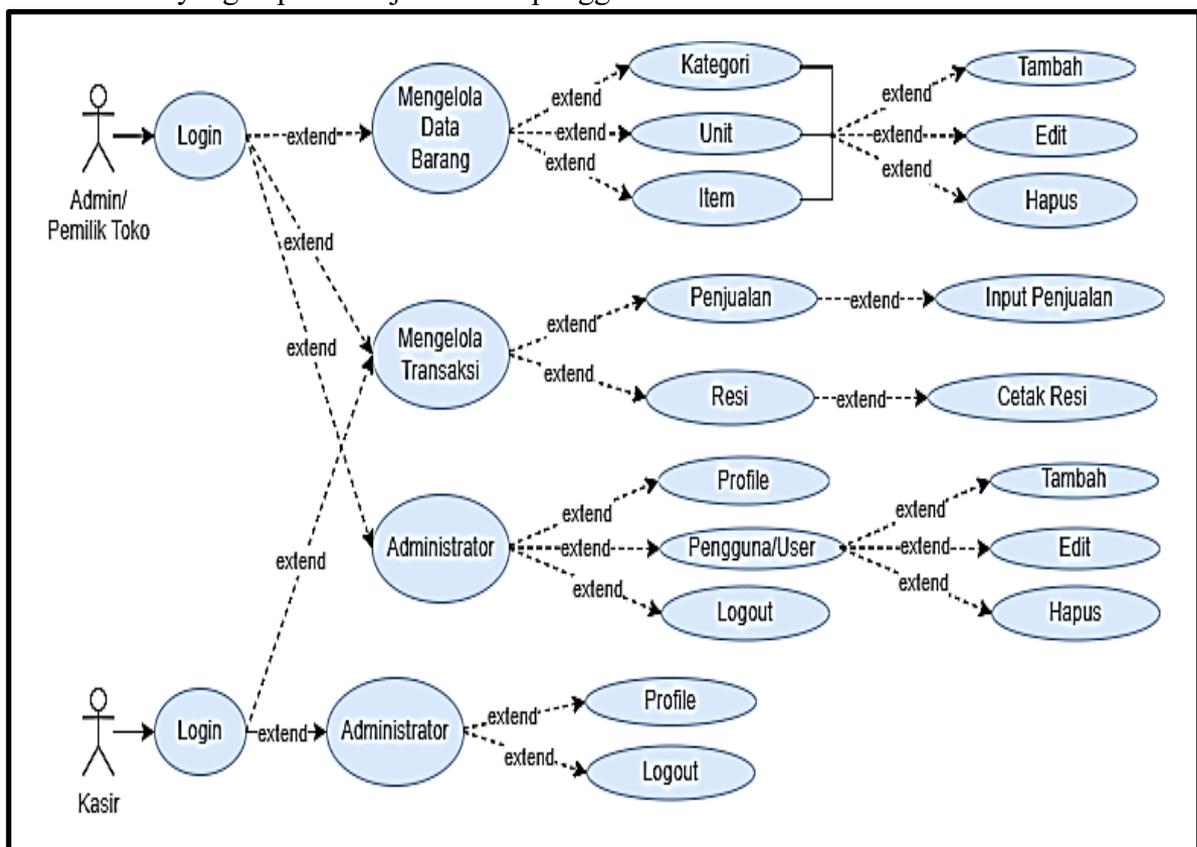
1.4 Rancangan Model Sistem

1.4.1 Perancangan UML (Unified Modeling Language)

Pada tahap perancangan ini, metode yang digunakan melibatkan UML (Unified Modeling Language) yang mencakup beberapa diagram, seperti *Use Case Diagram*, *Activity Diagram* dan *Class Diagram*. Diagram-diagram ini memungkinkan penggambaran operasional sistem dalam bentuk visual yang terstruktur.

1. Use Case Diagram

Use Case Diagram terdiri dari aktor atau *user*, *use case* dan hubungannya. Use Case Diagram digunakan untuk menjelaskan aktivitas yang dapat dikerjakan oleh pengguna sistem.



Gambar 3.3 Use Case Diagram Pembukuan

Pada gambar diatas menunjukkan bahwa sistem ini terdiri dari dua aktor yaitu admin/pemilik toko dan kasir. Hubungan actor (admin) dengan use case yang menjelaskan bahwa pada sistem admin yang mempunyai hak akses untuk input, tambah, hapus data, mengelolah transaksi, mengolah data dan cetak disemua fitur setelah melakukan login. Dan hubungan actor (kasir) dengan use case yang

menjelaskan bahwa pada system kasir yang mempunyai hak akses untuk input transaksi penjualan dan mencetak resi penjualan.

1. Skenario Use Case Mengelola Data Barang

- Nama Use Case : Kategori
 Aktor : Admin/Pemilik Toko
 Pre-Condition : Admin berhasil login
 Post-Condition : Admin berhasil melihat, menambahkan, memperbarui dan menghapus data kategori
 Deskripsi : Admin dapat mengelola data kategori

Tabel 3.1 Skenario *Use Case* Kategori

| Admin | Sistem |
|-------------------------------------|---|
| Membuka aplikasi | Menampilkan halaman kelola data kategori |
| Alternatif 1: tambah kategori | |
| Memilih tombol tambah data kategori | Menampilkan form tambah data kategori |
| Mengisi data kategori | Menampilkan form tambah data kategori |
| Memilih tombol simpan | Menampilkan notifikasi pesan “data berhasil ditambahkan” dan menampilkan halaman kelola data kategori |
| Alternatif 2: edit kategori | |
| Memilih tombol edit data kategori | Menampilkan form edit data kategori |
| Memperbarui data kategori | Menampilkan form tambah data kategori |
| Memilih tombol simpan | Menampilkan notifikasi pesan “data berhasil diubah” dan menampilkan halaman kelola data kategori |
| Alternatif 3: hapus kategori | |
| Memilih tombol hapus | Menampilkan konfirmasi hapus kategori |
| Memilih tombol ok | Menampilkan notifikasi pesan “data berhasil dihapus” dan menampilkan halaman kelola data kategori |

2. Skenario Use Case Mengelola Data Barang

| | |
|----------------|--|
| Nama Use Case | : Unit |
| Aktor | : Admin/Pemilik Toko |
| Pre-Condition | : Admin berhasil login |
| Post-Condition | : Admin berhasil melihat, menambahkan, memperbarui dan menghapus data Unit |
| Deskripsi | : Admin dapat mengelola data Unit |

Tabel 3. 2 Sekenario *Use Case* Unit

| Admin | Sistem |
|---------------------------------|---|
| Membuka aplikasi | Menampilkan halaman kelola data Unit |
| Alternatif 1: tambah Unit | |
| Memilih tombol tambah data Unit | Menampilkan form tambah data Unit |
| Mengisi data Unit | Menampilkan form tambah data Unit |
| Memilih tombol simpan | Menampilkan notifikasi pesan “data berhasil ditambahkan” dan menampilkan halaman kelola data Unit |
| Alternatif 2: edit Unit | |
| Memilih tombol edit data Unit | Menampilkan form edit data Unit |
| Memperbarui data Unit | Menampilkan form tambah data Unit |
| Memilih tombol simpan | Menampilkan notifikasi pesan “data berhasil diubah” dan menampilkan halaman kelola data Unit |
| Alternatif 3: hapus Unit | |
| Memilih tombol hapus | Menampilkan konfirmasi hapus Unit |
| Memilih tombol ok | Menampilkan notifikasi pesan “data berhasil dihapus” dan menampilkan halaman kelola data Unit |

3. Skenario Use Case Mengelola Data Barang

| | |
|---------------|--------|
| Nama Use Case | : Item |
|---------------|--------|

Aktor : Admin/Pemilik Toko
 Pre-Condition : Admin berhasil login
 Post-Condition : Admin berhasil melihat, menambahkan, memperbarui dan menghapus data Item
 Deskripsi : Admin dapat mengelola data Item

Tabel 3.3 Skenario *Use Case* Item

| Admin | Sistem |
|---------------------------------|---|
| Membuka aplikasi | Menampilkan halaman kelola data Item |
| Alternatif 1: tambah Item | |
| Memilih tombol tambah data Item | Menampilkan form tambah data Item |
| Mengisi data Item | Menampilkan form tambah data Item |
| Memilih tombol simpan | Menampilkan notifikasi pesan “data berhasil ditambahkan” dan menampilkan halaman kelola data Item |
| Alternatif 2: edit Item | |
| Memilih tombol edit data Item | Menampilkan form edit data Item |
| Memperbarui data Item | Menampilkan form tambah data Item |
| Memilih tombol simpan | Menampilkan notifikasi pesan “data berhasil diubah” dan menampilkan halaman kelola data Item |
| Alternatif 3: hapus Item | |
| Memilih tombol hapus | Menampilkan konfirmasi hapus Item |
| Memilih tombol ok | Menampilkan notifikasi pesan “data berhasil dihapus” dan menampilkan halaman kelola data Item |

4. Skenario Use Case Mengelola Data Transaksi

Nama Use Case : Input Penjualan
 Aktor : Admin/Pemilik Toko
 Pre-Condition : Admin berhasil login

Post-Condition : Admin berhasil melihat, menambahkan, memperbarui dan menghapus data penjualan

Deskripsi : Admin dapat mengelola data penjualan

Tabel 3.4 Skenario *Use Case Input* Penjualan Admin

| Admin | Sistem |
|---|--|
| Membuka aplikasi | Menampilkan halaman kelola data transaksi |
| Mengisi barcode/nama barang | |
| Mengisi jumlah barang, klik tombol tambah | Menampilkan list data keranjang belanja beserta harga Menampilkan notifikasi pesan “item berhasil ditambahkan ke keranjang” |
| Mengisi jumlah tunai | Menampilkan jumlah kembalian jika ada |
| Mengisi catatan, misal alamat pembeli dll | |
| Memilih tombol proses pembayaran | Menampilkan notifikasi pesan “yakin proses transaksi sudah benar?” |
| Memilih tombol konfirmasi | Menampilkan notifikasi pesan “SUKSES, transaksi berhasil” Menampilkan layout resi |

5. Skenario *Use Case* Mengelola Data Transaksi

Nama Use Case : Cetak Resi

Aktor : Admin/Pemilik Toko

Pre-Condition : Admin berhasil login

Post-Condition : Admin berhasil melihat dan mencetak resi

Deskripsi : Admin dapat mencetak resi penjualan

Tabel 3.5 Skenario *Use Case* Cetak Resi Admin

| Admin | Sistem |
|--|--|
| Membuka aplikasi | Menampilkan halaman resi |
| Klik tombol print data resi yang ingin dicetak | Menampilkan halaman cetak nota pada tab baru |

6. Skenario *Use Case* Mengelola Data Transaksi

Nama Use Case : Input Penjualan

Aktor : Kasir

Pre-Condition : Kasir berhasil login

Post-Condition : Kasir berhasil melihat, menambahkan, memperbarui dan menghapus data penjualan

Deskripsi : Kasir dapat mengelola data penjualan

Tabel 3.6 Skenario *Use Case Input* Penjualan Kasir

| Admin | Sistem |
|---|--|
| Membuka aplikasi | Menampilkan halaman kelola data transaksi |
| Mengisi barcode/nama barang | |
| Mengisi jumlah barang, klik tombol tambah | Menampilkan list data keranjang belanja beserta harga Menampilkan notifikasi pesan “item berhasil ditambahkan ke keranjang” |
| Mengisi jumlah tunai | Menampilkan jumlah kembalian jika ada |
| Mengisi catatan, misal alamat pembeli dll | |
| Memilih tombol proses pembayaran | Menampilkan notifikasi pesan “yakin proses transaksi sudah benar?” |
| Memilih tombol konfirmasi | Menampilkan notifikasi pesan “SUKSES, transaksi berhasil” Menampilkan layout resi |

7. Skenario Use Case Mengelola Data Transaksi

Nama Use Case : Cetak Resi

Aktor : Kasir

Pre-Condition : Kasir berhasil login

Post-Condition : Kasir berhasil melihat dan mencetak resi

Deskripsi : Kasir dapat mencetak resi penjualan

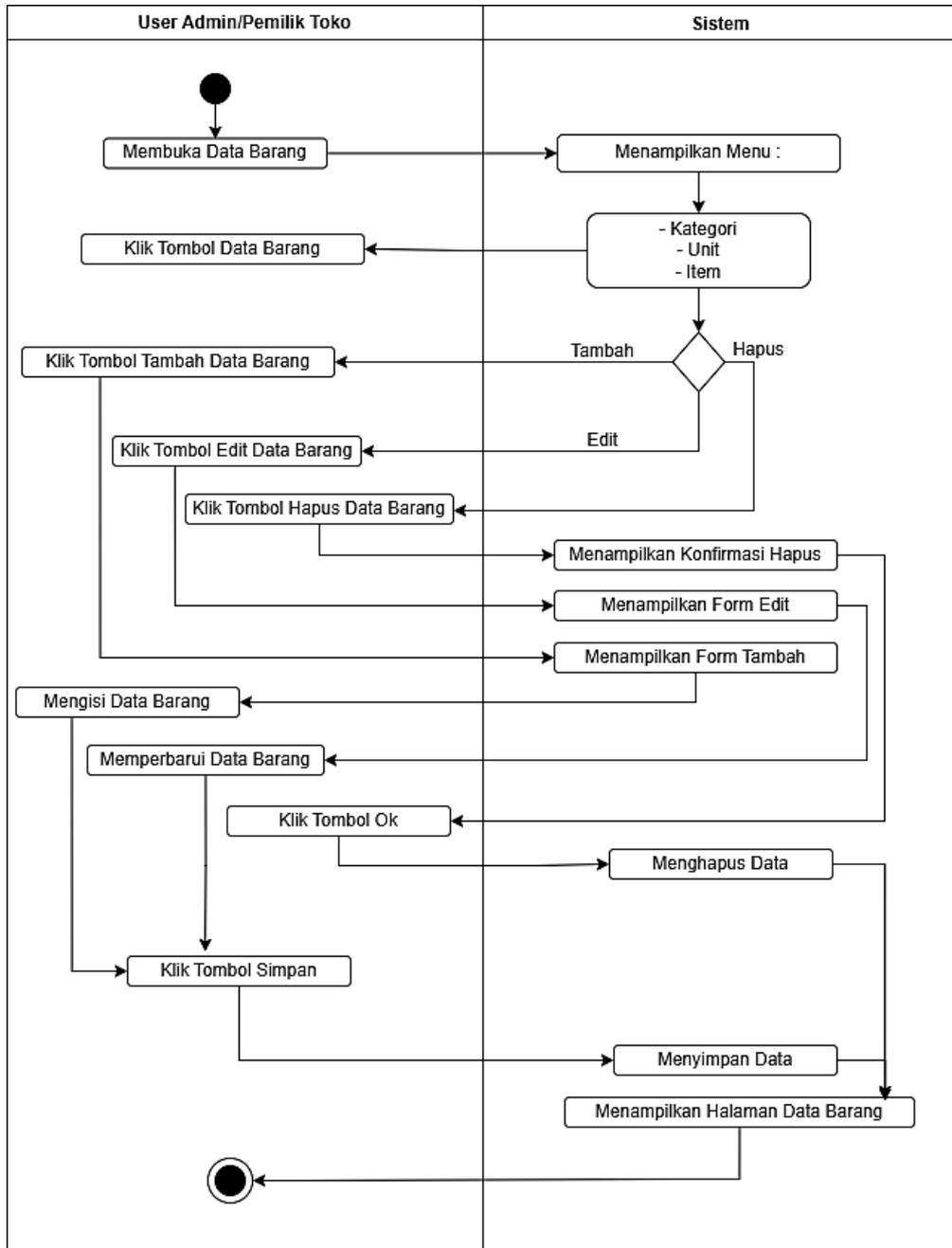
Tabel 3.7 Skenario *Use Case Cetak Resi* Kasir

| Admin | Sistem |
|--|--|
| Membuka aplikasi | Menampilkan halaman resi |
| Klik tombol print data resi yang ingin dicetak | Menampilkan halaman cetak nota pada tab baru |

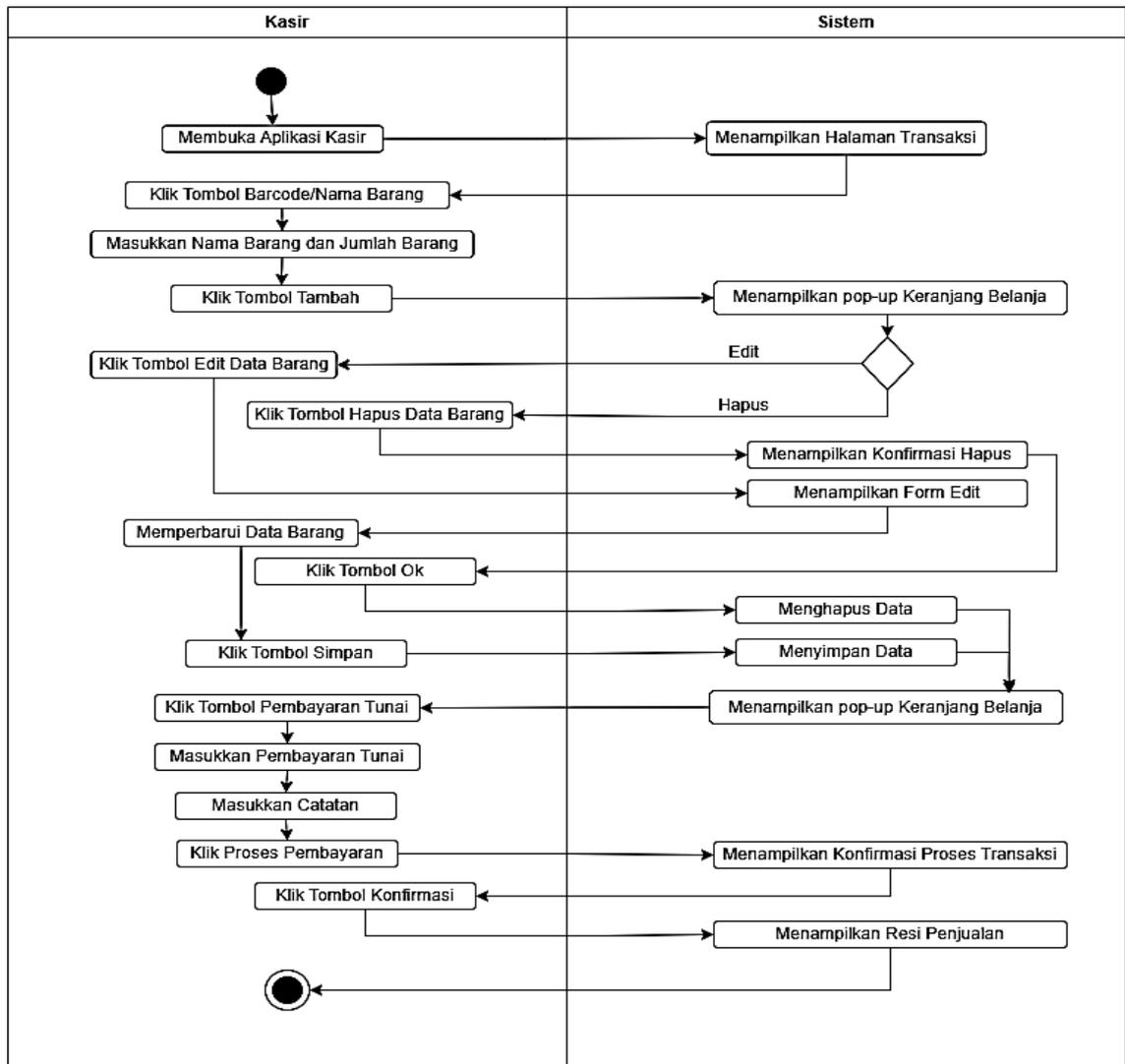
2. Activity Diagram

Activity diagram ini menggambarkan proses ketika pengguna melakukan pengelolaan terhadap data barang. *Activity Diagram* sangat berguna untuk menjelaskan aliran proses sistem dalam mengelola Data Barang.

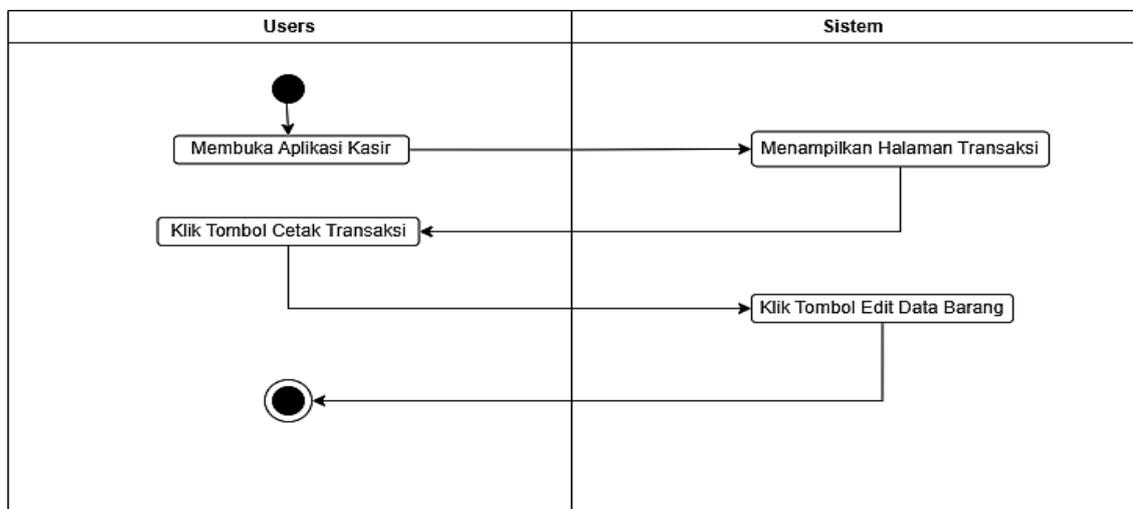
Activity Diagram menampilkan proses bisnis dan alur aktivitas dalam sebuah sistem yang akan dibangun. *Activity Diagram* dibuat untuk menjelaskan aktivitas aktor dan komputer dalam sistem. User Admin/Pemilik Toko harus melakukan login terlebih dahulu dengan username dan password yang telah dibuat.



Gambar 3.4 *Activity Diagram* Data Barang



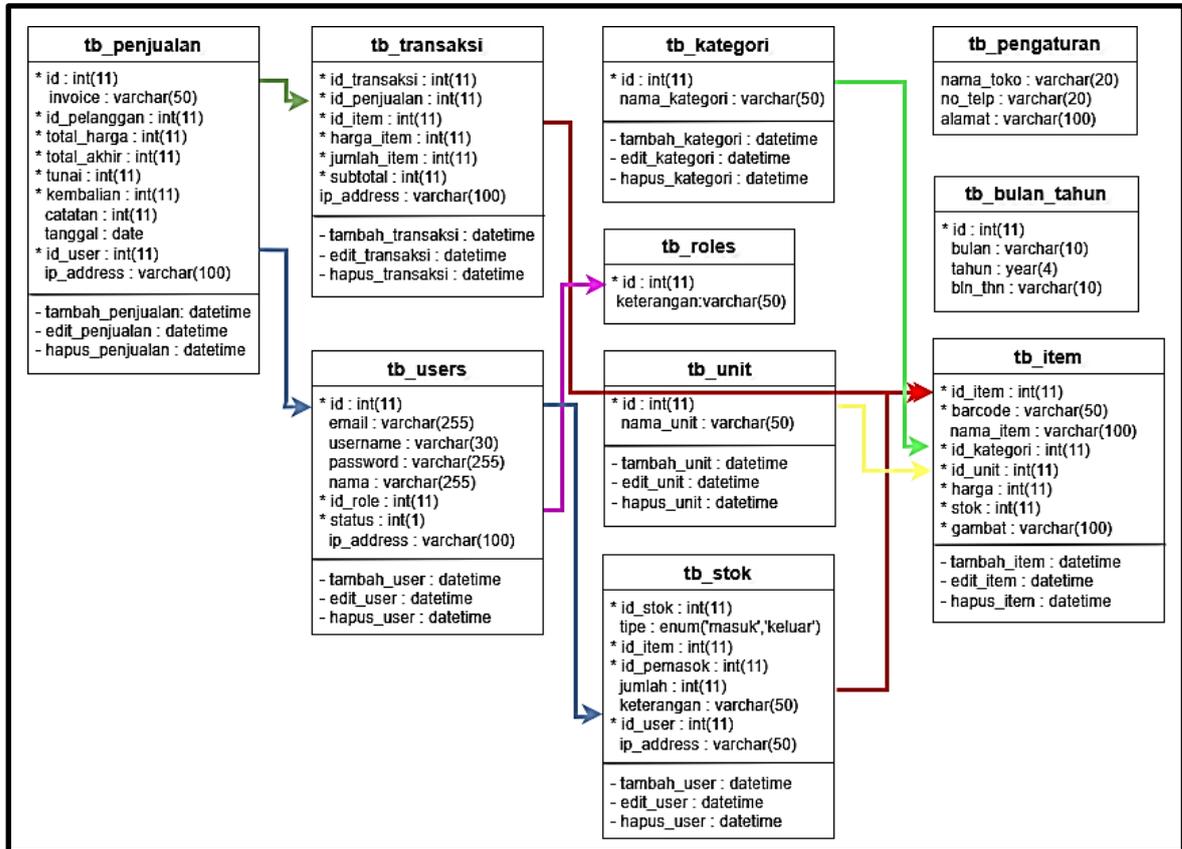
Gambar 3.5 Activity Diagram Penjualan



Gambar 3.6 Activity Diagram Cetak Resi

3. Class Diagram

Class Diagram *adalah* gambaran struktur sistem dari segi pendefinisian kelas-kelas yang akan dibuat untuk membangun aplikasi transaksi dan inventori ini.



Gambar 3.7 Class Diagram Penjualan

1.5 Rancangan Pengujian Sistem

Rancangan pada pengujian ini menggunakan *Blackbox* dan *Whitebox*.

1. Blackbox Testing

Metode uji "Kotak Hitam" atau "*Black Box*" melibatkan peninjauan fungsi-fungsi aplikasi tanpa memperhatikan cara kerjanya. Penguji hanya memperhatikan apakah aplikasi melakukan apa yang seharusnya dilakukan, tanpa memeriksa struktur atau mekanisme internalnya. Dalam metode ini, penguji memerlukan pemahaman tentang peran dan fungsi sistem tanpa perlu mengetahui detail internalnya. Mereka bertindak seperti pengguna biasa.

2. Whitebox Testing

Tes "Kotak Putih" melibatkan peninjauan fungsi aplikasi dan struktur internalnya, proses, dan komponen internal lainnya, bukan hanya fungsinya.

Dalam metode ini, semua komponen internal perangkat lunak atau aplikasi diuji melalui kode sumber, yang menjadi fokus utama pengujian.

1.6 Perancangan Database

Perancangan *database* yaitu proses penentuan konten dan struktur data yang diperlukan untuk mendukung berbagai rancangan sistem. Dalam perancangan *database* untuk pembuatan sistem ini, terdapat sepuluh tabel utama : tabel penjualan, tabel transaksi, tabel user, tabel roles, tabel stok, tabel kategori, tabel unit, tabel item, tabel pengaturan dan tabel bulan tahun. Berikut adalah penjelasan mengenai desain dari masing-masing tabel tersebut :

1. Struktur Tabel Penjualan

Fungsi : untuk mengelolah transaksi penjualan

Tabel 3.1 Struktur Tabel Penjualan

| <i>Field</i> | <i>Type Data</i> | <i>Size</i> | <i>Keterangan</i> |
|--------------|------------------|-------------|--------------------|
| id | <i>Integer</i> | 11 | <i>primary key</i> |
| id_pelanggan | <i>integer</i> | 11 | pelanggan |
| total_harga | <i>integer</i> | 11 | total harga |
| total_akhir | <i>integer</i> | 11 | total akhir |
| tunai | <i>integer</i> | 11 | tunai |
| kembalian | <i>integer</i> | 11 | kembalian |
| catatan | <i>text</i> | - | catatan |
| tanggal | <i>date</i> | - | tanggal |
| id_user | <i>integer</i> | 11 | user |
| ip_address | <i>varchar</i> | 100 | alamat |

2. Struktur Tabel Tansaksi

Fungsi : untuk melakukan transaksi penjualan dan resi

Tabel 3.2 Struktur Tabel Transaksi

| <i>Field</i> | <i>Type Data</i> | <i>Size</i> | <i>Keterangan</i> |
|-----------------|------------------|-------------|--------------------|
| id_transaksi | <i>integer</i> | 11 | <i>primary key</i> |
| id_penjualan | <i>integer</i> | 11 | penjualan |
| id_Transaksi | <i>integer</i> | 11 | transaksi |
| harga_Transaksi | <i>integer</i> | 11 | harga transaksi |

| | | | |
|------------------|----------------|----|------------------|
| jumlah_Transaksi | <i>integer</i> | 11 | jumlah transaksi |
| Subtotal | <i>integer</i> | 11 | subtotal |
| ip_address | <i>varchar</i> | 50 | alamat |

3. Struktur Tabel User

Fungsi : menambah, mengedit, menyimpan dan menghapus data user

Tabel 3.3 Struktur Tabel User

| <i>Field</i> | <i>Type Data</i> | <i>Size</i> | <i>Keterangan</i> |
|--------------|------------------|-------------|--------------------|
| id | <i>integer</i> | 11 | <i>primary key</i> |
| email | <i>varchar</i> | 50 | <i>email</i> |
| username | <i>varchar</i> | 30 | <i>Username</i> |
| password | <i>varchar</i> | 10 | <i>password</i> |
| nama | <i>varchar</i> | 25 | nama |
| alamat | <i>varchar</i> | 25 | alamat |
| id_role | <i>integer</i> | 50 | role |
| status | <i>integer</i> | 1 | status |
| ip_address | <i>varchar</i> | 25 | alamat |

4. Struktur Tabel Roles

Fungsi : Untuk memilih keterangan hak akses kepada pengguna

Tabel 3.4 Struktur Tabel Roles

| <i>Field</i> | <i>Type Data</i> | <i>Size</i> | <i>Keterangan</i> |
|--------------|------------------|-------------|--------------------|
| id | <i>integer</i> | 11 | <i>primary key</i> |
| keterangan | <i>varchar</i> | 50 | keterangan |

5. Struktur Tabel Stok

Fungsi : menambah, mengedit, menyimpan dan menghapus data stok

Tabel 3.5 Struktur Tabel Stok

| <i>Field</i> | <i>Type Data</i> | <i>Size</i> | <i>Keterangan</i> |
|--------------|------------------|-------------|--------------------|
| id_stok | <i>integer</i> | 11 | <i>primary key</i> |
| tipe | <i>enum</i> | - | 'masuk'_'keluar' |
| id_Transaksi | <i>integer</i> | 11 | transaksi |
| jumlah | <i>integer</i> | 11 | jumlah |
| keterangan | <i>varchar</i> | 50 | keterangan |
| id_user | <i>integer</i> | 11 | user |
| ip_address | <i>varchar</i> | 50 | alamat |

6. Struktur Tabel Kategori

Fungsi : menambah, mengedit, menyimpan dan menghapus data kategori

| <i>Field</i> | <i>Type Data</i> | <i>Size</i> | <i>Keterangan</i> |
|---------------|------------------|-------------|--------------------|
| id | <i>integer</i> | 11 | <i>primary key</i> |
| nama_kategori | <i>varchar</i> | 50 | nama kategori |

7. Struktur Tabel Unit

Fungsi : menambah, mengedit, menyimpan dan menghapus data unit

Tabel 3.6 Struktur Tabel Unit

| <i>Field</i> | <i>Type Data</i> | <i>Size</i> | <i>Keterangan</i> |
|--------------|------------------|-------------|--------------------|
| id | <i>integer</i> | 11 | <i>primary key</i> |
| nama_unit | <i>varchar</i> | 50 | nama unit |

8. Struktur Tabel Item

Fungsi : menambah, mengedit, menyimpan dan menghapus data item

Tabel 3.7 Struktur Tabel Item

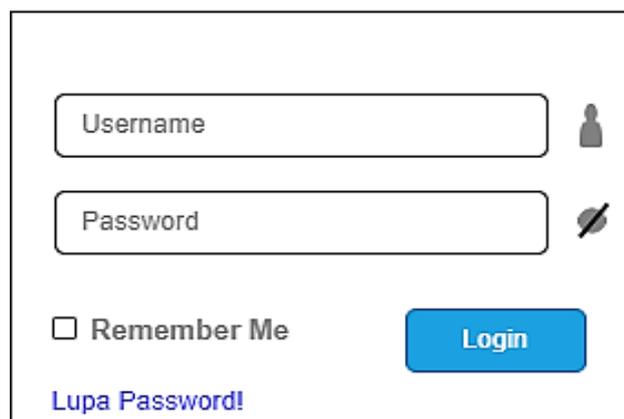
| <i>Field</i> | <i>Type Data</i> | <i>Size</i> | <i>Keterangan</i> |
|--------------|------------------|-------------|--------------------|
| id | <i>integer</i> | 11 | <i>Primary Key</i> |
| nama_item | <i>varchar</i> | 50 | nama item |

1.7 Perancangan Interface

Pada penelitian ini, perancangan antarmuka atau *interface* dari suatu sistem berfungsi untuk memberikan gambaran akan tampilan dari program. *Interface* yang bagus juga akan memberikan kemudahan bagi penggunaannya. Berikut adalah tampilan perancangan antarmuka :

1. *Interface* Halaman *Login*

Login

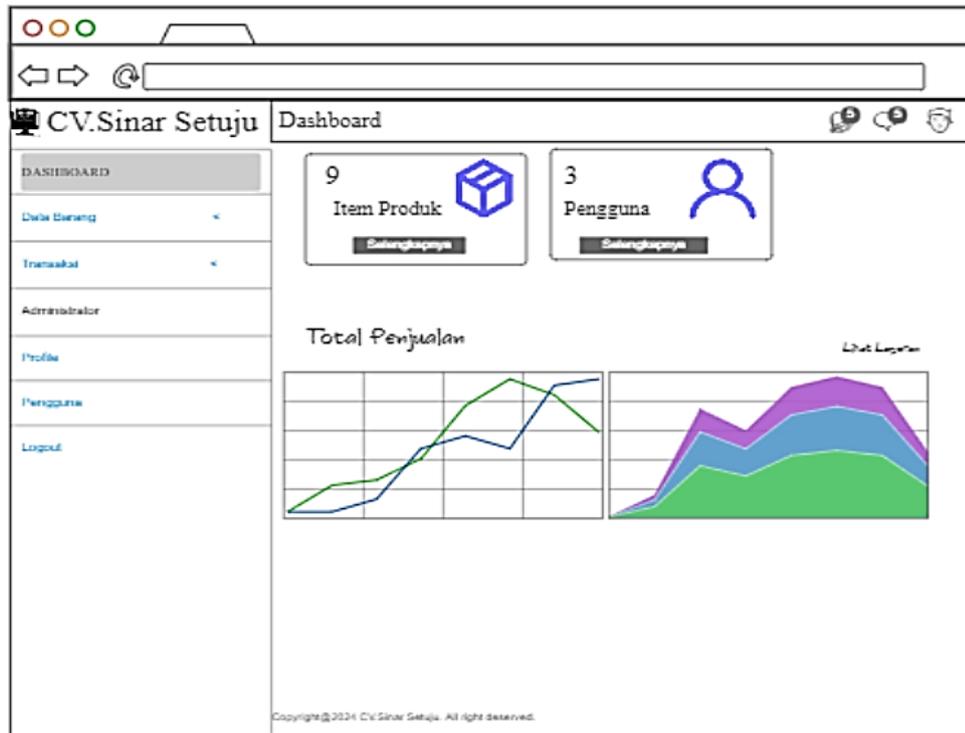


The image shows a login form with the following elements:

- A text input field labeled "Username" with a user icon to its right.
- A text input field labeled "Password" with a password icon to its right.
- A checkbox labeled "Remember Me".
- A blue button labeled "Login".
- A blue link labeled "Lupa Password!".

Gambar 3.8 *Interface* Hakaman *Login*

2. Interface Halaman Dashboard



Gambar 3.9 Interface Halaman Dashboard

3. Interface Halaman Data Barang

a. Interface Halaman Kategori

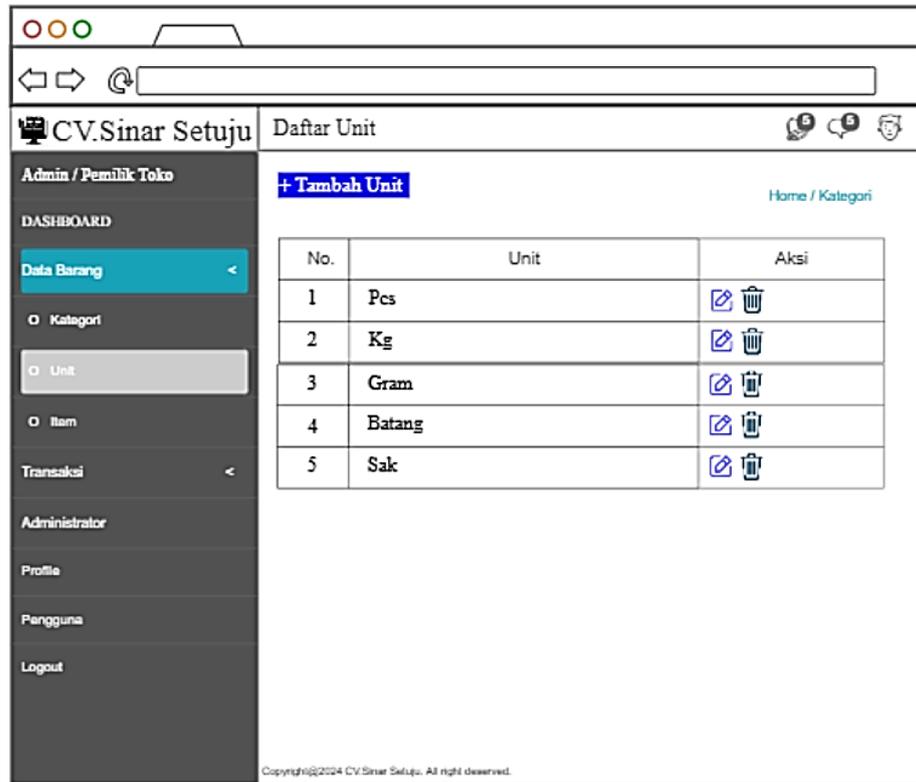
The screenshot shows the 'Daftar Kategori' page in the CV.Sinar Setuju system. The page title is 'Daftar Kategori' and it includes a breadcrumb 'Home / Kategori'. A '+ Tambah Kategori' button is located at the top left. The main content is a table with the following data:

| No. | Kategori | Aksi |
|-----|----------|------|
| 1 | Semen | |
| 2 | Pasir | |
| 3 | Tebel | |
| 4 | Paku | |
| 5 | Perkakas | |

A sidebar menu on the left lists: Admin / Pemilik Toko, DASHBOARD, Data Barang (selected), Kategori, Unit, Item, Transaksi, Administrator, Profile, Pengguna, and Logout. A copyright notice at the bottom reads 'Copyright © 2024 CV.Sinar Setuju. All right reserved.'

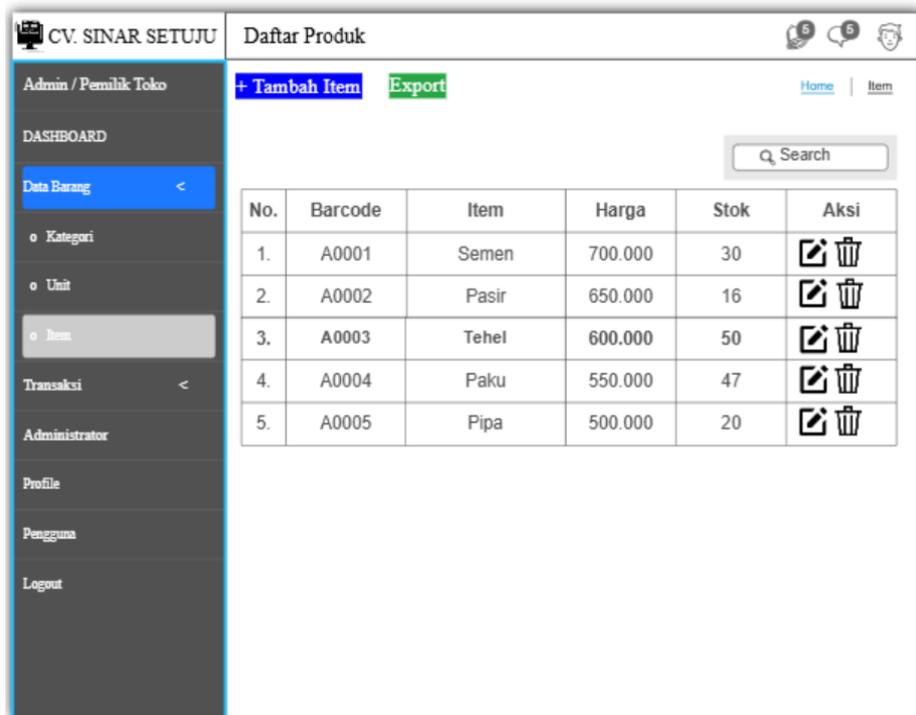
Gambar 3.10 Interface Halaman Kategori

b. *Interface Halaman Unit*



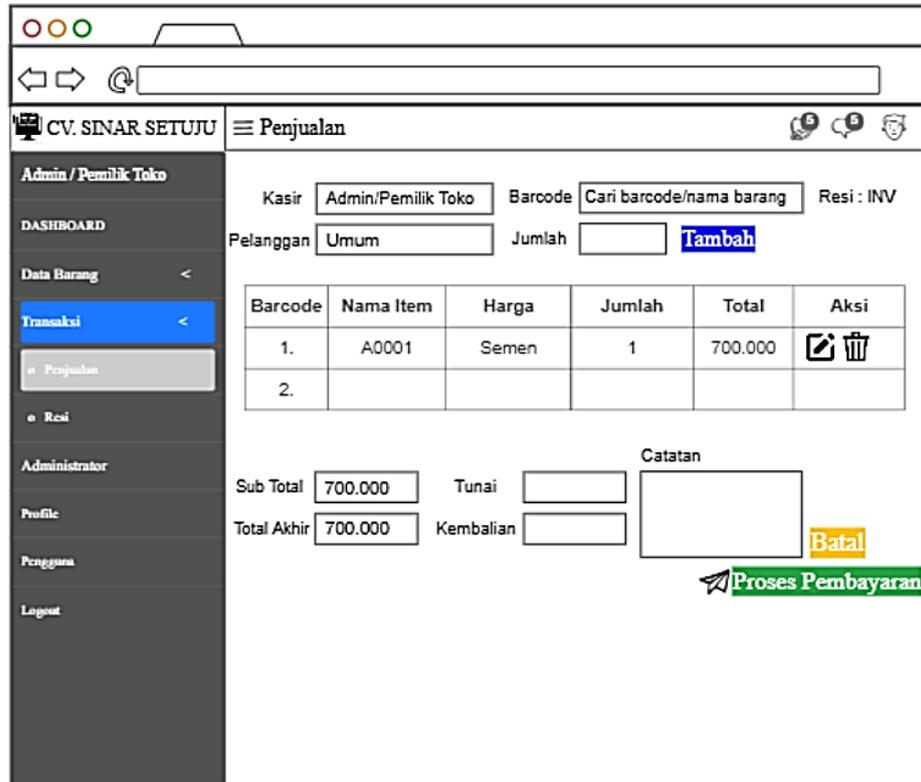
Gambar 3.11 *Interface Halaman Unit*

c. *Interface Halaman Item*



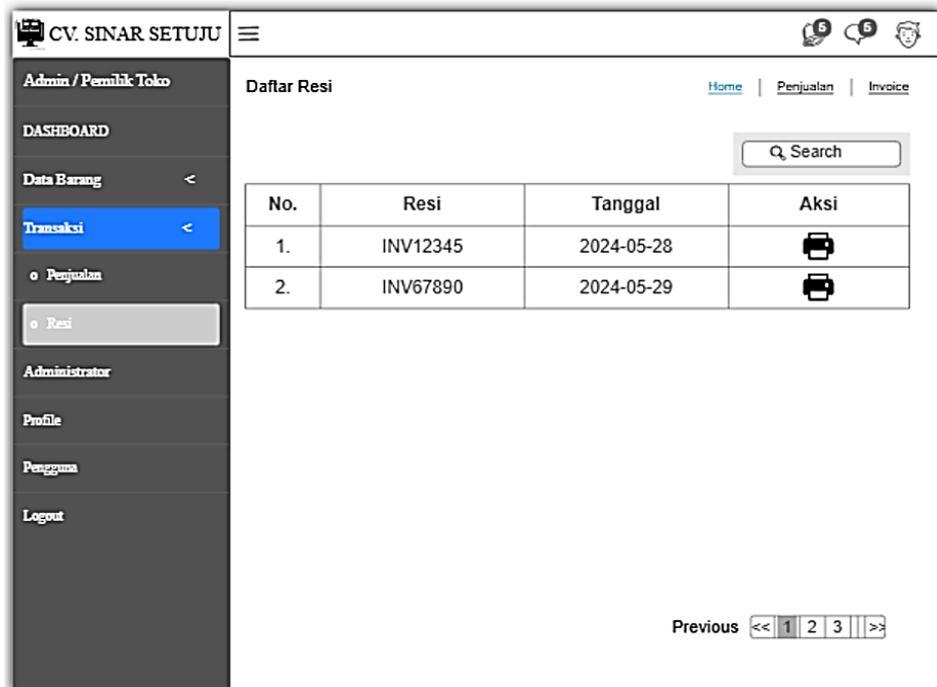
Gambar 3.12 *Interface Halaman Item*

4. *Interface* Halaman Transaksi
 - a. *Interface* Halaman Penjualan



Gambar 3.13 *Interface* Halaman Penjualan

- b. *Interface* Halaman Penjualan Resi



Gambar 3.14 *Interface* Halaman Penjualan Resi

BAB IV

IMPLEMENTASI DAN PENGUJIAN

4.1 Implementasi Sistem

Implementasi sistem melibatkan langkah-langkah untuk menerapkan desain yang telah dijelaskan dalam dokumen perancangan sistem yang telah disetujui, serta menguji, menginstal, memulai, dan menggunakan sistem yang baru atau yang sudah diperbaiki. Ini merupakan fase di mana aplikasi dibangun berdasarkan analisis dan desain sebelumnya untuk memastikan bahwa sistem yang dibuat dapat beroperasi dengan baik sesuai dengan tujuan yang telah ditetapkan. Tujuan dari proses implementasi ini meliputi:

1. Menyelesaikan desain sistem yang telah diatur dalam dokumen perancangan yang telah disetujui.
2. Melakukan uji coba dan dokumentasi terhadap program atau prosedur yang telah dijelaskan dalam dokumen perancangan sistem yang telah disetujui.
3. Memastikan bahwa pengguna sistem mampu mengoperasikan sistem tersebut dengan baik melalui kegiatan pelatihan.
4. Memastikan bahwa sistem memenuhi kebutuhan pengguna dengan melakukan uji coba secara menyeluruh.
5. Memastikan bahwa proses konversi ke sistem baru berjalan dengan lancar melalui perencanaan, pengawasan, dan instalasi sistem secara tepat.

4.1.1 Batasan Implementasi

Batasan implementasi dalam pengembangan sistem ini meliputi:

- a. Dikembangkan menggunakan bahasa pemrograman PHP versi 7.4.1
- b. DBMS yang digunakan adalah *MySQL*.
- c. Pengguna sistem adalah kasir dan admin selaku pemilik toko.

4.1.2 Lingkungan Operasional

Untuk menjalankan aplikasi ini, diperlukan beberapa komponen pendukung, termasuk perangkat keras dan perangkat lunak dengan spesifikasi sebagai berikut:

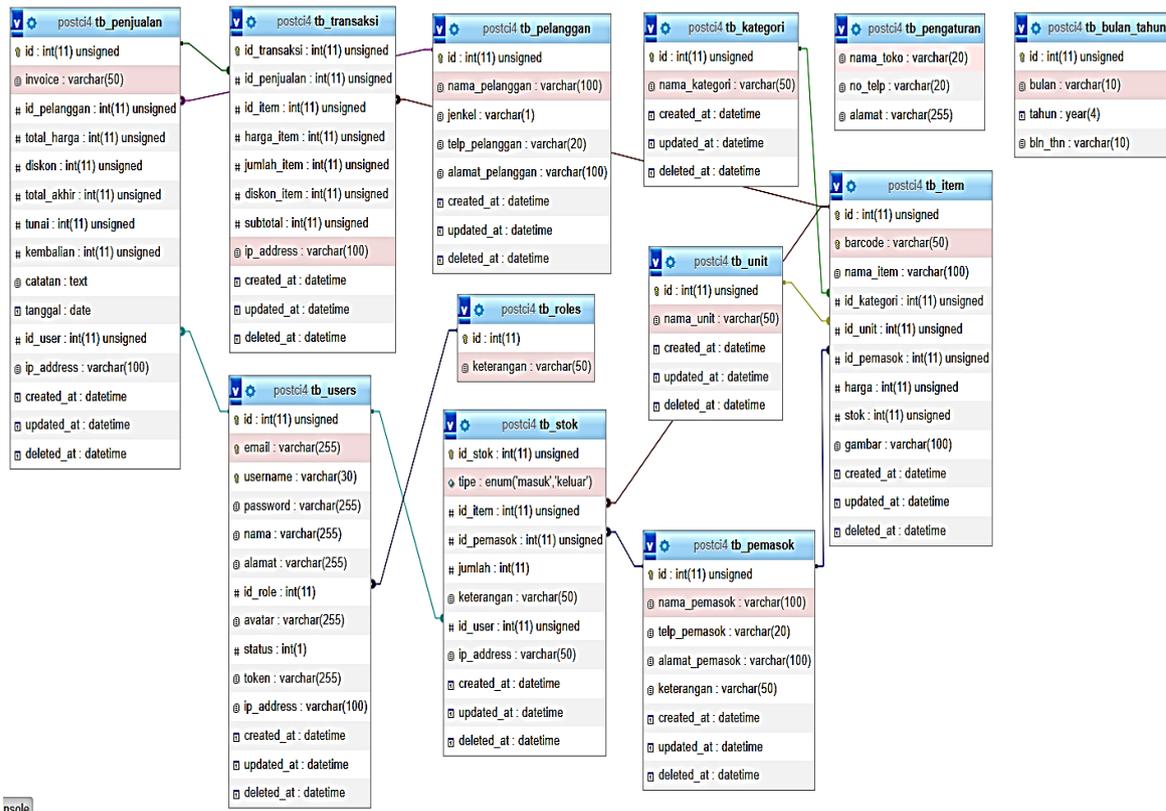
- a) Minimum perangkat keras yang dibutuhkan :
 1. *Processor* : Intel(R) Celeron(R) N4020 CPU @ 1.10GHz 1.10 GHz
 2. *Memory* : 4.00 GB RAM
 3. *SSD* : 250 GB

b) Perangkat lunak yang dibutuhkan:

1. Sistem Operasi : *Windows 11*
2. *Web Server* : *Apache*
3. *Browser* : *Chrome*
4. Bahasa Pemrograman : *HTML, PHP, Java Script*
5. *Tools* : *Visual Studio Code*
6. DBMS : *MySQL*
7. *Framework* : *Codeigniter 4*
8. *XAMPP* : *Versi 7.4.1*

4.1.3 Implementasi Database

Implementasi *database* atau proses lebih detail yang dapat mengubah desain *database* menjadi sistem operasional siap digunakan untuk mengelola dan menyimpan data. Proses ini melewati beberapa tahap dimulai dari perencanaan hingga pembukuan berlanjut. Adapun *database* dalam perancangan ini dapat dilihat dari gambar 4.1

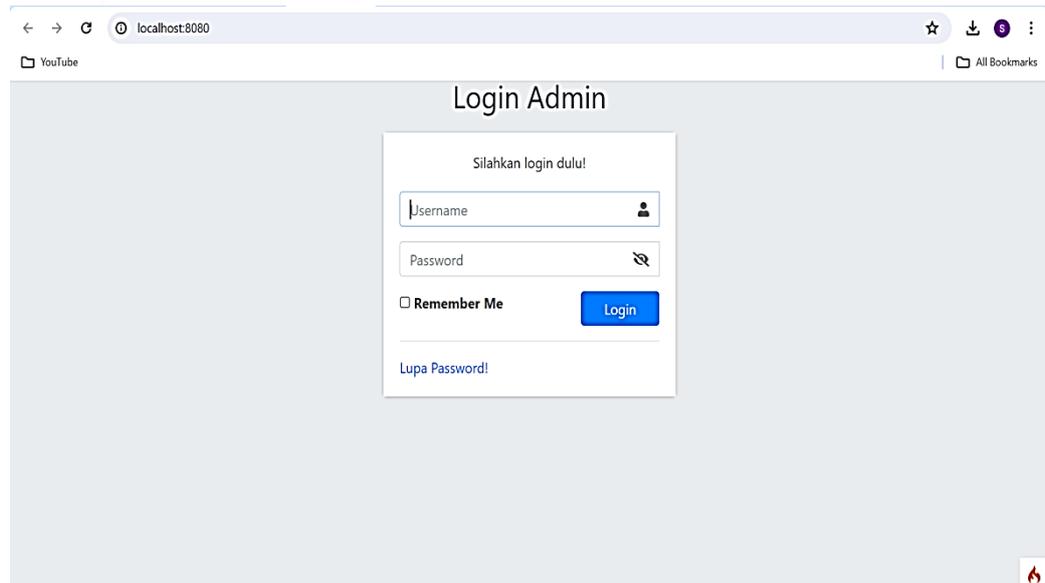


Gambar 4. 1 Implementasi Database

4.1.4 Implementasi Program

Implementasi program atau tahap dalam perancangan dan pengembangan perangkat lunak dimana rancangan dan spesifikasi yang telah dirancang sebelumnya dan diterapkan menjadi aplikasi yang siap digunakan oleh pengguna/*user*. Berikut implementasi program pada Perancangan Sistem Pembukuan dan Kasir berbasis *web* dengan menggunakan *Framework Codeigniter 4* adalah sebagai berikut :

1. Tampilan Halaman *Login*



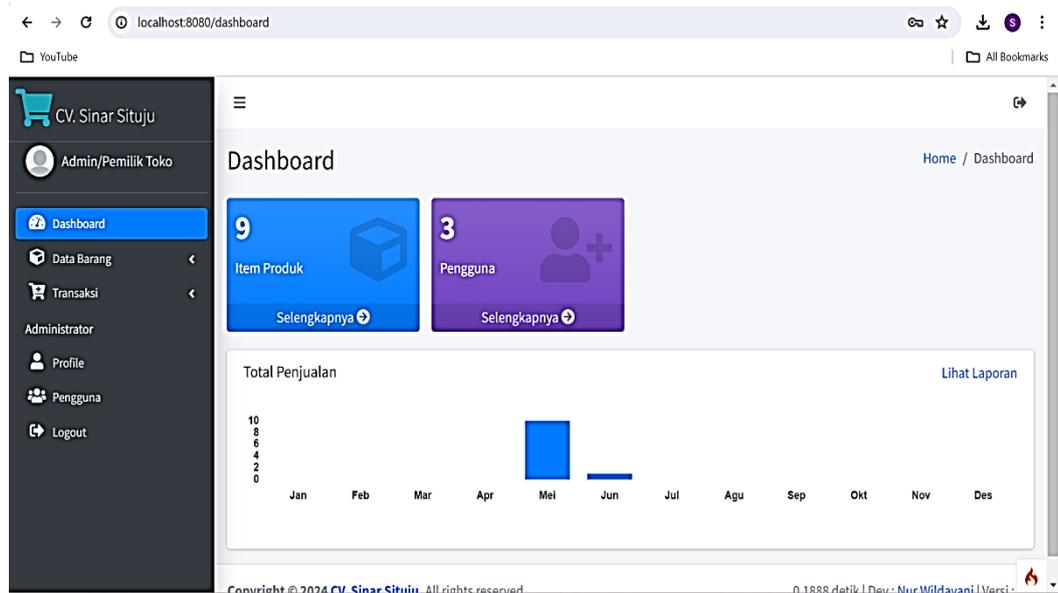
Gambar 4.2 Tampilan Halaman *Login*

Langkah awal ketika masuk ke sistem dengan menginput *username* dan *password* pada halaman *login*. Desain tampilan *login* atau syarat untuk menggunakan program secara penuh. Terdapat dua informasi yang harus dimasukkan pada menu *login* yaitu :

- a. Nama Pengguna/*User* : Identifikasi pengguna yang telah terdaftar dan diberikan izin untuk mengakses aplikasi.
- b. Kata Sandi/*Password* : Persyaratan mutlak untuk mengakses sistem/untuk dapat masuk ke dalam aplikasi.

2. Tampilan Halaman Utama

Halaman Dashboard untuk *user* dimana ada beberapa menu yaitu Dashboard, Data Barang yang meliputi Unit, Unit dan Transaksi, Transaksi yang meliputi Penjualan dan Resi. Pada menu Administrator terdapat Profile, Pengguna dan Logout.



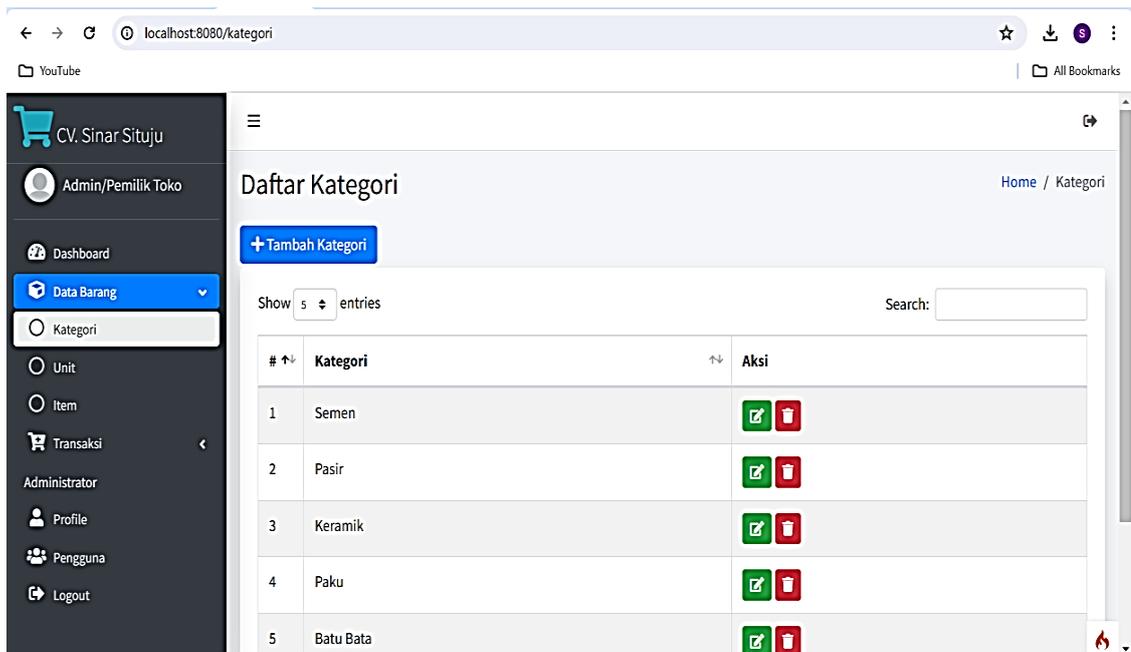
Gambar 4.3 Tampilan Halaman *Dashboard*

Setelah sukses *login*, maka tampil halaman utama/*dashboard* yang fungsinya untuk memberikan informasi *terupdate*.

3. Tampilan Halaman Data Barang

a. Halaman Kategori

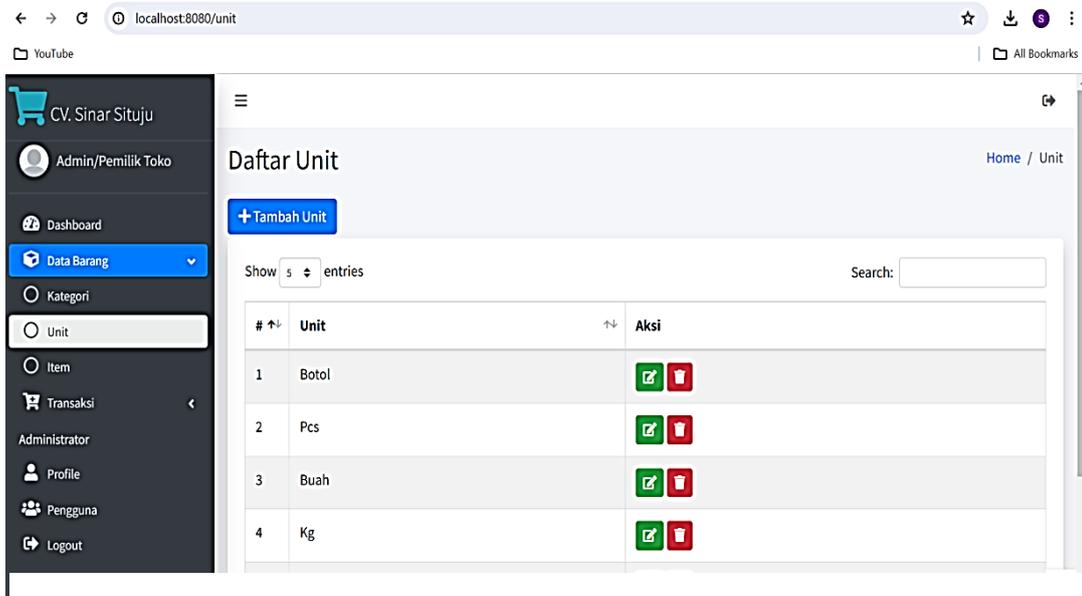
Pada data barang akan menampilkan menu Unit atau jenis barang yang tersedia pada toko bangunan yang dapat ditambah, edit dan menghapus Unit.



Gambar 4.4 Tampilan Halaman Kategori

b. Halaman Unit

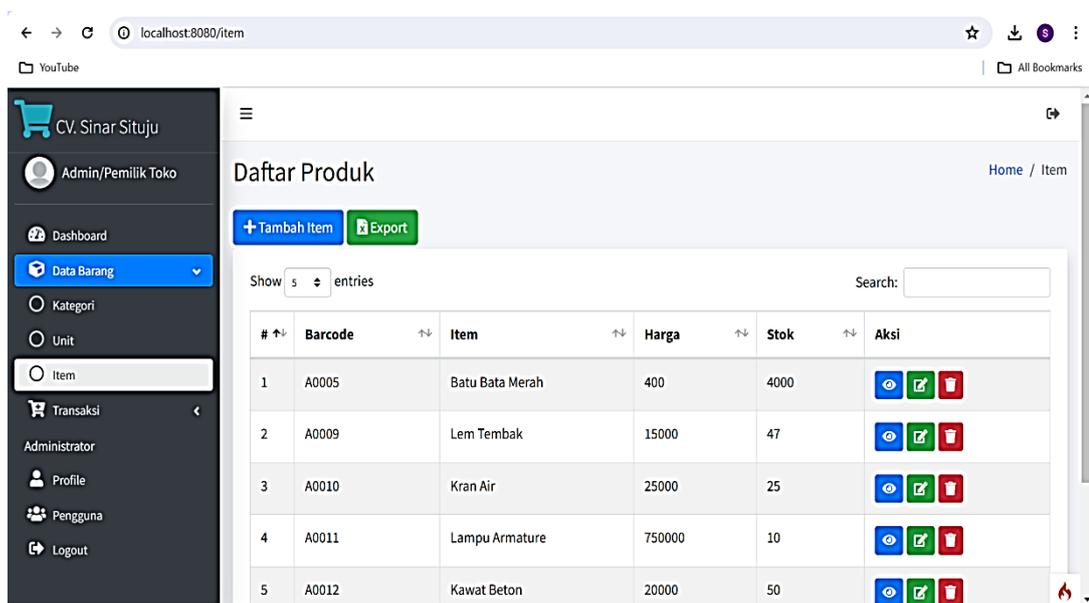
Pada data barang akan menampilkan menu unit atau satuan dari masing-masing barang yang dapat ditambahkan, mengedit dan menghapus unit.



Gambar 4.5 Tampilan Halaman Unit

d. Halaman Item

Pada data barang juga ada menu Transaksi yang dapat menampilkan semua yang ada pada menu Unit dan unit dimana *user* dapat melihat rincian, dapat diedit dan menghapus Transaksi.

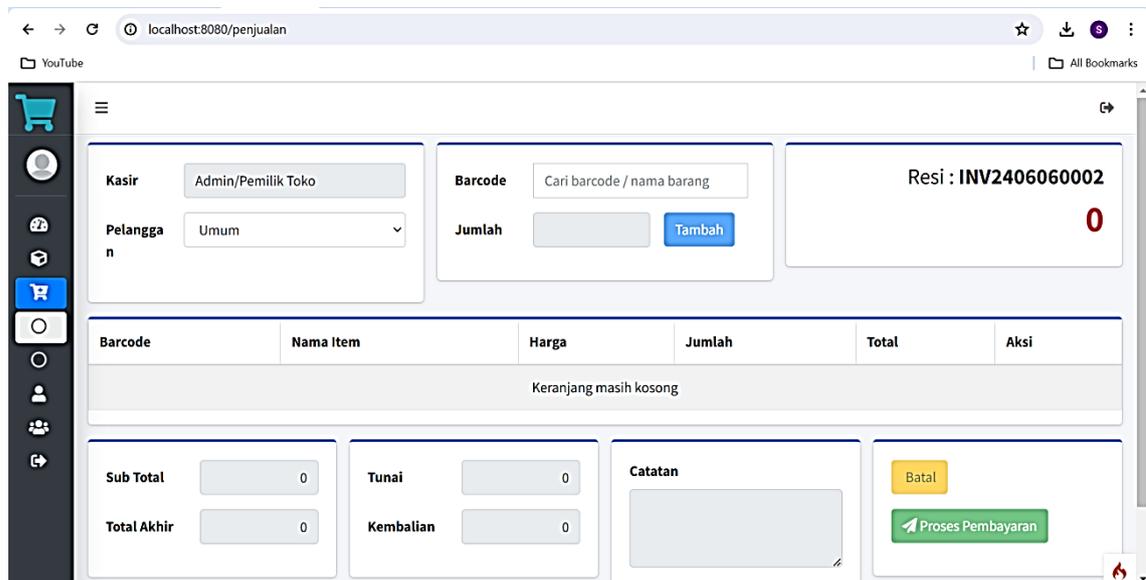


Gambar 4.6 Tampilan Halaman Item

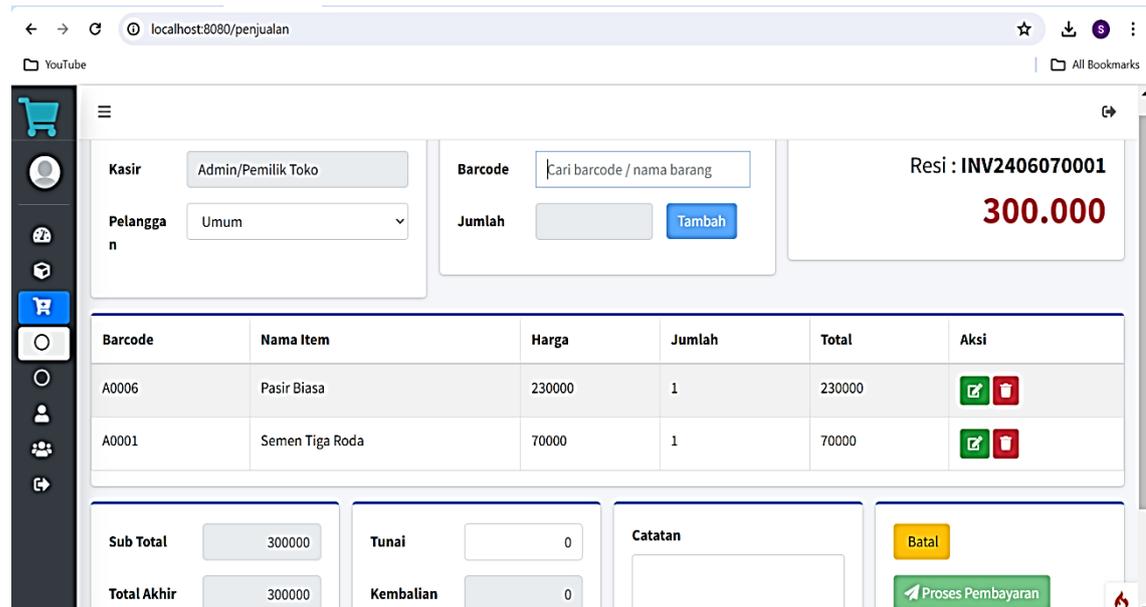
4. Tampilan Halaman Transaksi

a. Halaman Penjualan

Didalam menu penjualan, pengguna dapat melakukan transaksi jual beli dengan mengisi tabel barcode sampai tabel catatan.



Gambar 4.7 Tampilan Halaman Penjualan

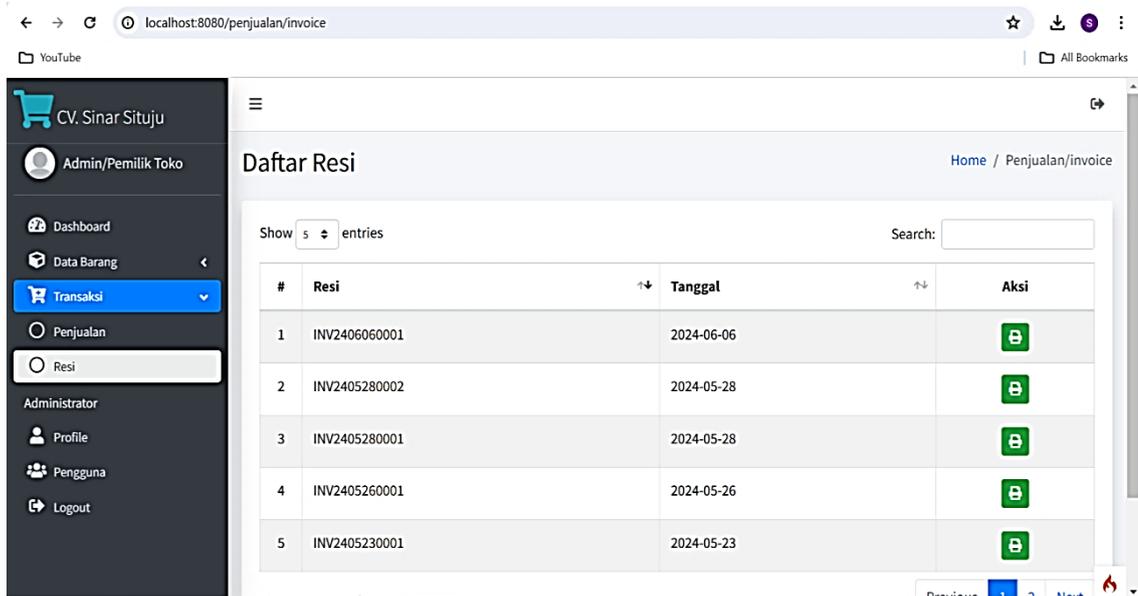


Gambar 4.8 Tampilan Halaman Penjualan setelah diisi

Tampilan pada saat semua tabel sudah terisi, dan barang yang diinginkan oleh pembeli sudah masuk dalam list keranjang belanja beserta total harga pembelian, pada tabel catatan bisa diisi oleh pengguna misal alamat pembeli atau keterangan lainnya.

b. Halaman Resi

Tampilan halaman resi menampilkan history struk belanja yang dapat dicetak sesuai tanggal yang diinginkan admin/pemilik toko maupun kasir.



Gambar 4.9 Tampilan Halaman Resi

4.2 Hasil Pengujian

4.2.1 Hasil Pengujian Black Box

Pengujian *blackbox* adalah metode yang difokuskan pada kebutuhan fungsional perangkat lunak. yang merujuk pada pendekatan bertujuan untuk memverifikasi fungsi-fungsi yang tersedia dalam perangkat lunak dan mengidentifikasi kesalahan dalam berbagai Unit, termasuk fungsi yang tidak tepat atau hilang, kesalahan antarmuka, masalah dalam struktur data atau akses database eksternal, ketidaksesuaian performa, serta kesalahan dalam inisialisasi dan terminasi. Pengujian sistem yang menggunakan metode *blackbox* ini dilakukan dengan :

Tabel 4.1 Pengujian *Black Box*

| No. | Kasus Uji | Langkah Uji | Hasil yang diharapkan | Hasil Aktual | Keterangan |
|-----|-------------------|---------------------------------------|-------------------------------|---|------------|
| 1. | Menu <i>Login</i> | Run program, tampil <i>form login</i> | Menampilkan <i>form login</i> | Tampilan <i>dashboard</i> /halaman utama. | Berhasil |

| | | | | | |
|----|---------------------|--------------------------|--|---|----------|
| 2. | Menu Data Barang | Klik menu Data Barang | Menampilkan pilihan Unit, Unit dan Transaksi | Tampilan Data Unit, Data Unit dan Data Transaksi. | Berhasil |
| 3. | Menu Data Unit | Klik menu Data Unit | Menampilkan <i>form</i> Data Unit | Tampilan Data Unit yang dapat ditambah, edit dan hapus. | Berhasil |
| 4. | Menu Data Unit | Klik menu Data Unit | Menampilkan <i>form</i> Data Unit | Tampilan Data Unit yang dapat ditambah, edit dan hapus. | Berhasil |
| 5. | Menu Data Transaksi | Klik menu Data Transaksi | Menampilkan <i>form</i> Data Transaksi | Tampilan Data Transaksi yang dapat ditambah, edit dan hapus. | Berhasil |
| 6. | Menu Data Transaksi | Klik menu Data Transaksi | Menampilkan pilihan Penjualan dan Resi | Tampilan Data Penjualan dan Data Resi | Berhasil |
| 7. | Menu Data Penjualan | Klik menu Data Penjualan | Menampilkan <i>form</i> Data Penjualan | Tampilan Data Penjualan yang dapat menginput transaksi Penjualan. | Berhasil |

| | | | | | |
|-----|--------------------|---------------------------|---------------------------------------|---|----------|
| 8. | Menu Data Resi | Klik menu Data Resi | Menampilkan <i>form</i> Data Resi | Tampilan history Data Resi yang dapat dicetak sesuai tanggal penjualan | Berhasil |
| 9. | Menu Data Profile | Klik menu Data Profile | Menampilkan <i>form</i> Data Profile | Tampilan Data Profile yang dapat diubah | Berhasil |
| 10. | Menu Data Pengguna | Klik menu Data Pengguna | Menampilkan <i>form</i> Data Pengguna | Tampilan Data Pengguna yang hanya dapat diakses oleh <i>admin/pemilik</i> toko dan dapat ditambahkan, diubah dan dihapus. | Berhasil |
| 11. | Menu <i>Logout</i> | Klik tombol <i>Logout</i> | Menampilkan konfirmasi <i>Logout</i> | <i>Logout</i> dari aplikasi dan menampilkan <i>form Login</i> . | Berhasil |

4.2.2 Hasil Pengujian *White Box*

White box testing untuk menguji perangkat lunak dari segi desain dan kode program apakah mampu menghasilkan fungsi-fungsi, masukan, dan keluaran yang sesuai dengan spesifikasi kebutuhan. Hasil pengujian *white box* dari perancangan Sistem Pembukuan dan Kasir berbasis *web* pada Toko Bangunan Sinar Setuju adalah sebagai berikut :

1. Pengujian White Box Login

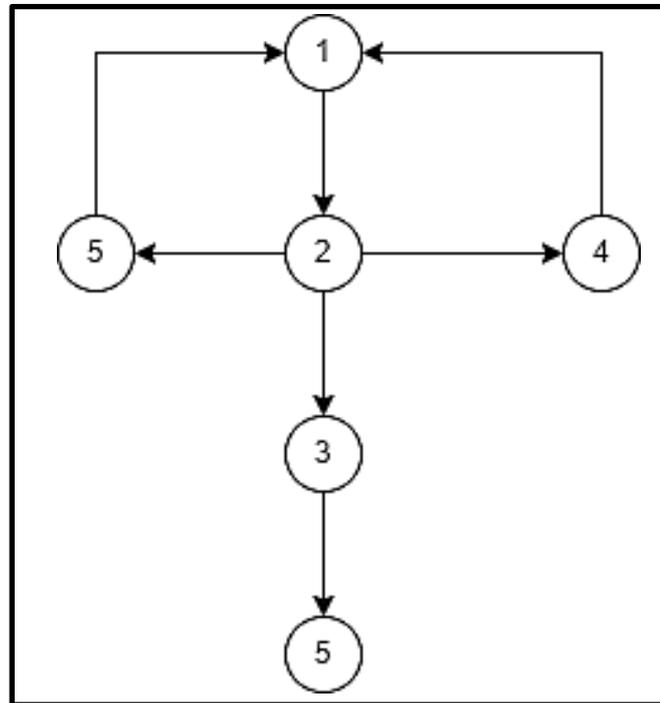
a. Source Code Validasi Login

```
public function proses()
{
    // jika methodnya post lanjut proses
    if ($this->request->getMethod() == 'post') {
        $rules = $this->validate([
            'username' => 'required|alpha_numeric',
            'password' => 'required'
        ]);
        if (!$rules) {
            $respon = [
                'validasi' => false,
                'error' => $this->validator->getErrors()
            ];
        } else {
            $user = $this->userModel->getUser($this->request-
>getPost('username', FILTER_SANITIZE_STRING));
            // jika username tidak terdaftar
            if (empty($user)) {
                $respon = [
                    'validasi' => true,
                    'error' => ['username' => 'Username tidak
terdaftar!']
                ];
            } else {
                // jika password tidak sama
                if (!verifikasi_password($this->request-
>getPost('password'), $user->password)) {
                    $respon = [
                        'validasi' => true,
                        'error' => ['password' => 'Password tidak
sesuai!']
                    ];
                } else {
                    // login sukses set session user
                    $data = [
                        'login' => true,
                        'id' => $user->id,
                    ];
                    $this->session->set($data); // set session
                    $respon = [
                        'validasi' => true,
                        'sukses' => true,
                        'aksi' => 'login',
                        'pesan' => 'Login berhasil!'
                    ];
                }
            }
        }
        return $this->response->setJSON($respon);
    }
}
```

Gambar 4.10 Source Code Validasi Login

b. *Flowgraph* Aplikasi *Form Login*

Tujuan dibuatnya *flowgraph* ini untuk menghitung *Cyclomatic Complexity* yang membutuhkan jumlah *node* dan jumlah *edge* dari suatu *flowgraph*.



Gambar 4.11 *Flowgraph* *Form Login*

c. *Cyclomatic Complexity* $V(G)$

Berdasarkan *flow graph* pada Gambar 4.10 diketahui bahwa jumlah *edge* (E) = 7 yang merupakan garis yang menghubungkan node, jumlah node (N) = 6 yang merupakan lingkaran yang menggambarkan suatu aktivitas, jumlah *predicate* (P) = 2 yang merupakan node bercabang, dan jumlah *region* (R) = 3 yang menandakan suatu area dalam *flow graph*, yang dapat dilihat dengan simbol R1 hingga R3 pada Gambar 4.10 sehingga jika dimasukkan ke dalam rumus perhitungan *cyclomatic complexity* dan juga bantuan *graph matrix* pada Tabel 4.2 dan Tabel 4.3, maka akan menghasilkan sebagai berikut :

Tabel 4. 2 *Cyclomatic Complexity* *Route Form Login*

| | | |
|--------------------|----------------|------------|
| $V(G) = E - N + 2$ | $V(G) = P + 1$ | $V(G) = R$ |
| $V(G) = 7 - 6 + 2$ | $V(G) = 2 + 1$ | $V(G) = 3$ |
| $V(G) = 3$ | $V(G) = 3$ | |

Tabel 4.3 *Graph Matrix Route Form Login*

| | 1 | 2 | 3 | 4 | 5 | 6 | Hasil |
|--------|---|---|---|---|---|---|-------|
| 1 | | I | | | | | 0 |
| 2 | | | I | I | | | 1 |
| 3 | | | | | I | I | 1 |
| 4 | I | | | | | | 0 |
| 5 | I | | | | | | 0 |
| 6 | | | | | | | 0 |
| Jumlah | | | | | | | 3 |

d. *Independent Path*

Dari hasil perhitungan tersebut didapatkan 6 jalur independen yaitu :

Path 1 : 1 – 2 – 3 – 4 – 5 – 6 (Skenario *login* berhasil)

Path 2 : 1 – 2 – 4 - 1

Path 3 : 1 – 2 – 5 - 1

e. *Tabel test case*

Tabel *test case* yang telah dibuat ditampilkan pada tabel 4.2

Tabel 4. 4 Tabel *Test Case* Aplikasi *Form Login*

| No | Kegiatan | Hasil yang diharapkan | Hasil | Keterangan |
|----|---|---|---|------------|
| 1. | <i>Input</i> nama <i>username</i> dan <i>password</i> yang benar | Masuk ke <i>Dashboard</i> | Masuk ke <i>Dashboard</i> | Berhasil |
| 2. | <i>Input</i> nama <i>username</i> dan <i>password</i> kosong | Muncul Pesan <i>Error</i> Redirect Ke <i>Login</i> | Muncul pesan <i>Error</i> Redirect Ke <i>Login</i> | Berhasil |
| 3. | <i>Input</i> nama <i>username</i> kosong sedangkan kolom <i>password</i> terisi | Muncul Pesan <i>Error</i> Redirect Ke <i>Login</i> | Muncul pesan <i>Error</i> Redirect Ke <i>Login</i> | Berhasil |

BAB V

PENUTUP

5.1 Kesimpulan

Penelitian ini menggambarkan implementasi CodeIgniter 4 sebagai kerangka kerja utama dalam pengembangan sistem pembukuan dan kasir berbasis web. Melalui implementasi ini, dapat dilihat bagaimana framework tersebut digunakan untuk membangun sistem yang fleksibel, efisien, dan mudah dikembangkan. Skripsi ini membahas desain arsitektur sistem secara rinci, yang meliputi desain database, logika bisnis, tata letak antarmuka pengguna, dan fungsionalitas sistem. Desain ini memberikan landasan yang kuat untuk pengembangan dan pemeliharaan sistem yang kompleks.

Penggunaan basis data dalam sistem ini sangat penting untuk menyimpan informasi pelanggan, transaksi, dan data penting lainnya. Dengan memanfaatkan basis data secara efektif, sistem dapat memberikan layanan yang cepat dan andal kepada pengguna. Sistem kasir yang dikembangkan dalam skripsi ini memungkinkan manajemen transaksi secara efisien. Hal ini dapat membantu pemilik bisnis dalam mengelola inventaris, melacak penjualan, dan menghasilkan laporan keuangan yang akurat. Aspek keamanan sistem menjadi prioritas utama dalam pengembangan aplikasi web. Dalam skripsi ini, upaya-upaya keamanan, seperti validasi input, enkripsi data sensitif, dan perlindungan terhadap serangan siber, telah diperhatikan untuk memastikan perlindungan yang optimal terhadap data pengguna.

Meskipun telah diimplementasikan dengan baik, sistem yang dikembangkan dalam skripsi ini mungkin memiliki keterbatasan tertentu. Misalnya, perlu diperhatikan pengoptimalan kinerja dan skalabilitas sistem saat digunakan dalam skala yang lebih besar. Selain itu, peluang pengembangan masa depan, seperti integrasi dengan teknologi baru atau peningkatan fungsionalitas, juga dapat dieksplorasi.

Kesimpulannya, skripsi ini telah berhasil merancang dan mengimplementasikan sistem pembukuan dan kasir berbasis web menggunakan CodeIgniter 4. Sistem ini dapat dijadikan referensi dan titik

awal bagi pengembang yang tertarik untuk membangun solusi serupa dalam konteks bisnis mereka sendiri.

5.2 Saran

Untuk meningkatkan sistem Anda, ada beberapa saran yang dapat Anda pertimbangkan. Pertama, Anda dapat mengembangkan modul tambahan seperti manajemen stok atau laporan keuangan yang lebih terperinci untuk memberikan nilai tambah kepada pengguna. Selain itu, melakukan optimalisasi kinerja sistem dengan teknik-teknik seperti caching dan pengindeksan database yang efisien akan meningkatkan responsivitas sistem. Pastikan juga untuk melakukan pengujian yang komprehensif termasuk pengujian fungsional, keamanan, dan kinerja untuk memastikan keandalan aplikasi sebelum diperkenalkan ke lingkungan produksi. Selanjutnya, penyempurnaan antarmuka pengguna yang ramah dan intuitif serta pengembangan aplikasi mobile akan meningkatkan pengalaman pengguna secara signifikan. Terakhir, terus perbarui praktik keamanan sistem seperti penerapan HTTPS dan perlindungan terhadap serangan injeksi SQL untuk melindungi sistem dari ancaman keamanan potensial. Melakukan kolaborasi dengan bisnis lokal juga dapat memberikan umpan balik berharga untuk mengarahkan peningkatan sistem sesuai dengan kebutuhan pasar lokal.

DAFTAR PUSTAKA

- Harike, M. H. (2019). Aplikasi E-voting Pemilihan OSIS pada SMA NEGERI 11 Luwu. *Seminar Nasional Teknologi Informasi Dan Komputer*, 175–183.
- Ii, B. A. B., & Pustaka, T. (2002). *BAB II Tinjauan Pustaka BAB II TINJAUAN PUSTAKA 2.1*. 1–64.
- Ii, B. A. B., & Teori, L. (2018). *BAB II LANDASAN TEORI 2.1 Rapid Application Development*. 7–11.
- Indra Sukma. (2022). BAB II Tinjauan Pustaka BAB II TINJAUAN PUSTAKA 2.1. 1–64. *Gastronomía Ecuatoriana y Turismo Local*, 1(69), 5–24.
- Julia, I. (2022). Perancangan Sistem Informasi Kasir Dengan Barcode Berbasis Web Dan Framework Codeigniter. *Informatika*, 10(2), 80–84.
<https://doi.org/10.36987/informatika.v10i2.2987>
- Muchdi Noor Hidayat. (2022). Perancangan dan Pengembangan Sistem Informasi Kasir Penjualan Sembako Berbasis Web Pada Toko Nurhidayah. *Prosiding SEINTEK Universitas Pamulang*, 148–161.
- Rashifah, Z., & Budi, E. S. (2022). Rancangan Sistem Informasi Pada Kasir Berbasis Web. *Jurnal Sistem Komputer Dan Informatika (JSON)*, 3(4), 529.
<https://doi.org/10.30865/json.v3i4.4241>
- Saputra, A., Safitri, C. F. I., Fitriyani, F., Gulo, Y., & Desyani, T. (2021). Pengembangan Aplikasi Kasir Menggunakan Model Waterfall. *Jurnal Teknologi Sistem Informasi Dan Aplikasi*, 4(2), 86. <https://doi.org/10.32493/jtsi.v4i2.10167>
- Student, M. T., Kumar, R. R., Ommments, R. E. C., Prajapati, A., Blockchain, T.-A., MI, A. I., Randive, P. S. N., Chaudhari, S., Barde, S., Devices, E., Mittal, S., Schmidt, M. W. M., Id, S. N. A., PREISER, W. F. E., OSTROFF, E., Choudhary, R., Bit-cell, M., In, S. S., Fullfillment, P., ... Fellowship, W. (2021). Jurnal Website. *Frontiers in Neuroscience*, 14(1), 1–13.
- Suminten, S. (2020). Sistem Informasi Penjualan Aplikasi Kasir Berbasis Website Pada Mart Serba Guna Blora. *PROSISKO: Jurnal Pengembangan Riset Dan Observasi Sistem Komputer*, 7(2), 102–107. <https://doi.org/10.30656/prosisko.v7i2.2320>