



JURNAL SISTEM BASIS DATA UNTUK APLIKASI KASIR BERBASIS MOBILE

Hana Habibah¹, Sephia Andini², Rahmat Friyady³

Universitas Pamulang, Tangerang

habibahhana4@gmail.com

Abstrak

Perkembangan teknologi informasi telah mendorong digitalisasi berbagai aktivitas bisnis, termasuk manajemen transaksi dan stok barang di kasir. Jurnal ini membahas perancangan sistem basis data untuk aplikasi kasir berbasis mobile menggunakan metode Waterfall. Sistem ini dirancang untuk mendukung pengelolaan pengguna, produk, transaksi, dan laporan penjualan secara efektif. Entity Relationship Diagram (ERD) dan struktur tabel yang diimplementasikan dalam phpMyAdmin dijelaskan secara detail untuk memastikan konsistensi dan integritas data.

Kata Kunci : Basis Data, Aplikasi Kasir, phpMyAdmin, ERD, Waterfall

Article History

Received: Desember 2024

Reviewed: Desember 2024

Published: Desember 2024

Plagiarism Checker No 234

Prefix DOI : Prefix DOI :

10.8734/Kohesi.v1i2.365

Copyright : Author

Publish by : Kohesi



This work is licensed

under a [Creative](https://creativecommons.org/licenses/by-nc/4.0/)

[Commons Attribution-](https://creativecommons.org/licenses/by-nc/4.0/)

[NonCommercial 4.0](https://creativecommons.org/licenses/by-nc/4.0/)

[International License](https://creativecommons.org/licenses/by-nc/4.0/)

PENDAHULUAN

Dengan berkembangnya teknologi informasi, banyak aspek kehidupan telah berubah, termasuk sektor bisnis dan perdagangan. Baik bisnis kecil maupun besar harus melakukan transformasi digital untuk meningkatkan efisiensi operasi dan memberikan layanan yang lebih baik kepada pelanggan. Sistem kasir berbasis digital yang menggantikan proses manual merupakan inovasi besar.

Banyak toko ritel menggunakan sistem kasir untuk mencatat transaksi penjualan, memantau stok barang, dan membuat laporan keuangan. Namun, masalah seperti kesalahan pencatatan, ketidaksesuaian stok, dan kehilangan data sering terjadi, yang dapat mengganggu kinerja bisnis.

Aplikasi kasir berbasis mobile dirancang untuk membantu pelanggan mengelola transaksi secara real-time, memantau stok barang, dan membuat laporan penjualan dengan cepat dan akurat. Aplikasi ini memiliki banyak keuntungan, seperti kemudahan akses, integrasi data yang lebih baik, dan efisiensi waktu.

Untuk proyek ini, model *Waterfall* digunakan sebagai metodologi pengembangan perangkat lunak karena pendekatannya yang terstruktur dan sistematis, yang memungkinkan setiap tahap pengembangan dilakukan dengan cermat sebelum melanjutkan ke tahap berikutnya, sehingga kebutuhan pengguna dapat diidentifikasi dan dipenuhi dengan lebih baik.



Basis data, yang dirancang untuk menjamin kebenaran, kesatuan, dan efisiensi dalam pengelolaan data, merupakan komponen utama dari aplikasi ini. *Entity Relationship Diagram (ERD)* menunjukkan bagaimana entitas dalam sistem berhubungan satu sama lain; ini termasuk pengguna, produk, transaksi, dan detail transaksi. Selain itu, untuk mempermudah pengujian dan pengelolaan, basis data diimplementasikan dengan *phpMyAdmin*.

Tujuan utama dari penelitian ini adalah untuk merancang dan menerapkan basis data yang optimal untuk mendukung aplikasi kasir berbasis ponsel. Dengan cara ini, diharapkan aplikasi kasir berbasis ponsel akan memberikan solusi yang andal bagi toko ritel untuk mengatasi masalah sehari-hari yang terkait dengan operasi mereka.

Jurnal ini akan membahas proses perancangan dan implementasi sistem basis data, termasuk penggunaan *ERD*, desain tabel, dan validasi relasi antar tabel. Hasilnya diharapkan dapat digunakan sebagai referensi untuk pengembangan aplikasi serupa di masa depan.

LANDASAN TEORI

1. Basis Data

Menurut Connolly dan Begg (2015), basis data adalah kumpulan data yang disusun secara sistematis yang memungkinkan akses, pengelolaan, dan pembaruan informasi. Sistem manajemen basis data (DBMS) adalah sistem yang mengelola pengambilan dan penyimpanan data. Aplikasi kasir menggunakan basis data untuk menyimpan informasi pengguna, produk, transaksi, dan detail transaksi.

Sistem Manajemen Basis Data (DBMS) adalah perangkat lunak yang dirancang untuk mengelola penyimpanan dan pengambilan data secara efektif. Connolly dan Begg (2015) menjelaskan bahwa DBMS memungkinkan integrasi data dan aksesibilitas yang lebih baik, sehingga mengurangi redundansi dan meningkatkan konsistensi data. Eriana (2020) juga menyoroti bahwa DBMS harus mendukung skalabilitas, terutama dalam aplikasi yang menangani volume data besar dan sering diakses secara real-time, seperti sistem kasir.

Aplikasi kasir modern memanfaatkan basis data untuk mencatat transaksi penjualan, memantau stok barang, dan menghasilkan laporan keuangan. Connolly dan Begg (2015) menegaskan pentingnya pengelolaan data yang terstruktur untuk memastikan kelancaran operasional. Dalam penelitian yang sama, Eriana (2020) menekankan bahwa sistem kasir yang dirancang dengan baik dapat memberikan solusi andal bagi pelaku bisnis untuk mengatasi masalah seperti kesalahan pencatatan dan kehilangan data.

Entity Relationship Diagram (ERD) adalah alat konseptual yang digunakan untuk menggambarkan hubungan antar-entitas dalam basis data. Elmasri dan Navathe (2010) menyebutkan bahwa ERD membantu dalam pemetaan entitas dan atributnya, sehingga memudahkan proses desain dan implementasi basis data. Eriana (2020) menambahkan bahwa penggunaan ERD memungkinkan pengelolaan hubungan antar-entitas seperti pengguna, produk, dan transaksi secara lebih efisien, serta memastikan bahwa atribut kunci utama dan kunci asing didefinisikan dengan baik.

Implementasi basis data biasanya dilakukan menggunakan alat seperti *phpMyAdmin*, yang memungkinkan pengelolaan tabel dan hubungan antar-entitas secara visual. Welling dan Thomson (2009) menjelaskan bahwa *phpMyAdmin* adalah alat yang populer karena antarmukanya yang user-friendly dan mendukung berbagai operasi MySQL. Eriana (2020) mencatat bahwa pengujian integritas data, termasuk validasi relasi antar-tabel, adalah langkah penting dalam memastikan keberhasilan implementasi.



Desain dan pengelolaan basis data yang baik adalah fondasi utama bagi keberhasilan aplikasi kasir berbasis mobile. Connolly dan Begg (2015) menyimpulkan bahwa sistem basis data yang dirancang secara sistematis dapat memenuhi kebutuhan bisnis yang semakin kompleks. Eriana (2020) mendukung pandangan ini dengan menekankan pentingnya memastikan integritas dan konsistensi data untuk meningkatkan keandalan aplikasi dan pengalaman pengguna secara keseluruhan.

2. *Entity Relationship Diagram (ERD)*

Elmasri dan Navathe (2010) menyatakan bahwa ERD adalah alat pemodelan konseptual yang digunakan untuk menunjukkan entitas dalam basis data serta hubungan yang ada di antara mereka. Dalam studi ini, ERD membantu memetakan hubungan antara entitas seperti pengguna, produk, transaksi, dan detail transaksi, sehingga desain basis data menjadi lebih terstruktur dan efisien.

ERD terdiri dari beberapa komponen utama, yaitu entitas, atribut, dan relasi. Entitas merepresentasikan objek atau konsep yang memiliki data yang relevan dalam basis data, seperti pengguna, produk, transaksi, dan detail transaksi dalam sistem kasir. Atribut adalah informasi atau karakteristik yang dimiliki oleh entitas, seperti nama pengguna, harga produk, dan jumlah barang dalam transaksi. Sementara itu, relasi menggambarkan hubungan antar-entitas, seperti hubungan antara pengguna yang melakukan transaksi dengan detail barang yang dibeli. Dengan mendefinisikan komponen-komponen ini secara jelas, desain basis data menjadi lebih terstruktur, sehingga meminimalkan kemungkinan terjadinya redundansi atau inkonsistensi data.

Keunggulan utama ERD adalah kemampuannya dalam memberikan representasi yang sederhana namun informatif terhadap struktur basis data yang kompleks. Diagram ini membantu pengembang dan pemangku kepentingan non-teknis untuk memahami logika di balik sistem yang dirancang. Dalam studi ini, ERD digunakan untuk memetakan hubungan antara pengguna, produk, transaksi, dan detail transaksi. Sebagai contoh, pengguna dapat terhubung dengan transaksi melalui atribut ID pengguna, sementara transaksi dapat terhubung dengan produk melalui detail transaksi. Relasi ini membantu mengelola data dengan lebih efisien, memastikan integritas referensial, dan mempermudah proses pembaruan data.

Selain itu, ERD juga berfungsi sebagai panduan dalam implementasi basis data. Diagram ini dapat diterjemahkan menjadi tabel-tabel dalam basis data relasional, di mana setiap entitas diwakili oleh tabel, atribut menjadi kolom, dan relasi menjadi kunci asing. Dalam sistem kasir berbasis mobile, misalnya, tabel pengguna akan berisi informasi seperti nama dan ID pengguna, sementara tabel transaksi akan mencatat ID transaksi, ID pengguna, dan total harga. Relasi antara tabel ini dapat diimplementasikan menggunakan kunci asing, sehingga memastikan data antar-tabel saling terhubung dengan baik.

Secara keseluruhan, ERD adalah alat yang sangat penting dalam desain basis data, terutama dalam aplikasi kasir yang membutuhkan pengelolaan data secara sistematis dan efisien. Dengan memanfaatkan ERD, pengembang dapat memastikan bahwa basis data dirancang dengan struktur yang optimal untuk mendukung operasi sistem secara real-time. Elmasri dan Navathe (2010) menekankan bahwa representasi konseptual yang disediakan oleh ERD tidak hanya mempermudah proses desain tetapi juga mengurangi potensi



kesalahan selama pengembangan sistem, sehingga menghasilkan aplikasi yang lebih andal dan efisien.

3. Elemen-elemen dalam ERD

1) Entitas

Entitas adalah elemen dasar dalam pemodelan data yang merujuk pada objek atau konsep yang dapat diidentifikasi secara unik dan berbeda satu sama lain. Dalam konteks basis data atau diagram Entity-Relationship (ER), entitas biasanya digambarkan dengan simbol persegi panjang. Setiap entitas memiliki atribut yang menggambarkan karakteristiknya, yang membedakannya dari entitas lainnya. Contohnya, dalam sebuah sistem manajemen sekolah, entitas dapat berupa "Siswa," "Guru," atau "Mata Pelajaran," yang masing-masing memiliki atribut seperti nama, alamat, atau kode mata pelajaran.

Selain entitas biasa, ada juga yang disebut dengan "entitas lemah." Entitas lemah ini digambarkan dengan lambang persegi panjang kecil di dalam persegi panjang yang lebih besar. Perbedaan utama antara entitas biasa dan entitas lemah adalah bahwa entitas lemah tidak dapat diidentifikasi secara unik tanpa bergantung pada entitas lain. Dengan kata lain, entitas lemah membutuhkan entitas lain untuk dapat diidentifikasi, karena entitas lemah tidak memiliki kunci utama yang membedakannya secara independen. Sebagai contoh, dalam sistem basis data rumah sakit, "Tindakan Medis" bisa menjadi entitas lemah yang tidak dapat berdiri sendiri tanpa entitas lain seperti "Pasien" atau "Dokter," karena tindakan medis hanya dapat diidentifikasi jika ada hubungan dengan pasien yang menjalani tindakan tersebut.

Entitas lemah dinamakan demikian karena tidak memiliki keberadaan atau identitas yang jelas tanpa asosiasi dengan entitas lain yang lebih kuat. Mereka bergantung pada entitas yang lebih kuat atau independen untuk dapat dibedakan dan diidentifikasi. Dalam konteks pemodelan ER, hubungan antara entitas lemah dan entitas lainnya biasanya digambarkan dengan garis ganda, yang menunjukkan ketergantungan kuat antara keduanya. Misalnya, dalam sistem basis data perusahaan, "Tugas Karyawan" bisa menjadi entitas lemah yang terkait erat dengan entitas "Karyawan." Tanpa adanya hubungan dengan entitas "Karyawan," entitas "Tugas Karyawan" tidak akan memiliki identitas yang jelas.

Pada dasarnya, entitas lemah digunakan untuk menggambarkan objek yang eksistensinya bergantung pada objek lainnya, sehingga mereka harus terhubung langsung dengan entitas lain dalam diagram ER. Hal ini memungkinkan pemodelan hubungan antar objek atau entitas yang lebih kompleks dalam sebuah sistem basis data, di mana entitas lemah berfungsi untuk mencatat informasi tambahan yang tidak bisa berdiri sendiri tanpa entitas utama yang lebih kuat.

2) Atribut

Atribut kunci atau Key Attributes adalah atribut yang berfungsi untuk menentukan data yang bersifat penting. Biasanya atribut kunci ini berbentuk angka atau numerik. Contoh dari atribut ini adalah No. KTP, NIM (Nomor Induk Mahasiswa), dan lain-lain. Atribut kunci ini dilambangkan dengan lingkaran lonjong dengan keterangan di dalamnya yang diberi garis bawah.



Berikutnya adalah atribut simpel. Atribut simpel adalah atribut yang tidak dapat dipecah lagi dan bernilai tunggal. Contoh dari atribut ini adalah alamat kantor, nama penerbit, dan lain-lain.

Atribut multinilai atau Multivalued Attributes adalah atribut yang memiliki atribut lebih dari satu nilai. Contoh dari atribut ini adalah sebuah website artikel yang memiliki beberapa penulis.

Selanjutnya adalah atribut gabungan atau Composite Attributes. Atribut gabungan adalah atribut yang terdiri dari beberapa atribut yang berukuran lebih kecil dan memiliki arti tertentu. Contoh dari atribut ini adalah sebuah nama yang terdiri atas nama depan, nama tengah, dan nama belakang.

Yang terakhir adalah atribut derivatif. Atribut derivatif adalah atribut yang dihasilkan dari atribut lain dan atributnya tidak wajib untuk ditulis dalam Entity Relationship Diagram. Contoh dari atribut ini adalah selisih harga, usia, dan kelas.

3) Relasi

Relasi atau relation. Relasi dalam ERD adalah hubungan yang terjadi antara satu atau lebih entitas. Relasi sendiri sering disebut dengan proses. Komponen ini digambarkan dengan lambang belah ketupat. Terdapat tiga jenis relasi yang digunakan dalam ERD

Setiap relasi dalam ERD dapat diartikan sebagai proses yang terjadi antara entitas-entitas yang saling berhubungan. Sebagai contoh, dalam aplikasi kasir, relasi dapat menggambarkan bagaimana entitas "Pengguna" terhubung dengan entitas "Transaksi," atau bagaimana entitas "Transaksi" terhubung dengan entitas "Produk." Relasi ini memastikan bahwa data yang dihasilkan dan disimpan dalam basis data memiliki konteks yang jelas, sehingga memudahkan proses pengambilan data untuk berbagai keperluan, seperti analisis atau pelaporan.

Terdapat tiga jenis relasi utama yang biasa digunakan dalam ERD: relasi satu ke satu (1:1), relasi satu ke banyak (1:N), dan relasi banyak ke banyak (M:N). Relasi satu ke satu terjadi ketika satu entitas berhubungan langsung dengan satu entitas lainnya, seperti hubungan antara entitas "Pengguna" dengan "Profil Pengguna." Relasi satu ke banyak menggambarkan situasi di mana satu entitas dapat berhubungan dengan banyak entitas lainnya, seperti hubungan antara "Pengguna" dengan "Transaksi," di mana satu pengguna dapat melakukan banyak transaksi. Sementara itu, relasi banyak ke banyak menggambarkan hubungan di mana banyak entitas dalam satu tabel dapat berhubungan dengan banyak entitas di tabel lainnya, seperti hubungan antara "Produk" dan "Transaksi," yang sering kali memerlukan tabel penghubung untuk mencatat detail transaksi.

Relasi dalam ERD juga melibatkan atribut tertentu yang memperjelas sifat hubungan tersebut. Sebagai contoh, pada relasi antara "Pengguna" dan "Transaksi," atribut seperti "ID Pengguna" dapat digunakan untuk mengidentifikasi pengguna yang melakukan transaksi tertentu. Dengan menggunakan atribut-atribut ini, relasi dalam basis data dapat diimplementasikan menggunakan kunci asing (foreign key) yang menjaga integritas referensial dan memastikan konsistensi data antar-tabel.

Selain mendefinisikan hubungan antar-entitas, relasi dalam ERD juga membantu dalam perancangan basis data yang efisien. Dengan memahami relasi yang ada, pengembang dapat mengidentifikasi kebutuhan akan tabel penghubung atau indeks



untuk mempercepat pencarian data. Relasi yang dirancang dengan baik tidak hanya meningkatkan efisiensi basis data tetapi juga meminimalkan redundansi data yang dapat menyebabkan masalah dalam pengelolaan data di masa depan.

Secara keseluruhan, relasi adalah elemen kunci dalam ERD yang memungkinkan penggambaran hubungan antar-entitas secara sistematis dan terstruktur. Dengan memahami jenis-jenis relasi dan bagaimana mereka diterapkan dalam basis data, pengembang dapat merancang sistem yang lebih efisien, konsisten, dan relevan dengan kebutuhan pengguna. Dalam aplikasi kasir, pemanfaatan relasi yang tepat memastikan bahwa data seperti pengguna, transaksi, dan produk dapat diintegrasikan dengan baik untuk mendukung operasional bisnis yang optimal.

4. *PhpMyAdmin*

phpMyAdmin adalah alat berbasis web yang digunakan untuk mengelola basis data MySQL. Welling dan Thomson (2009) menjelaskan bahwa phpMyAdmin menawarkan antarmuka grafis yang memudahkan pengguna dalam mengelola basis data. Sebelumnya, banyak operasi yang harus dilakukan melalui command line, yang bisa membingungkan bagi pengguna yang tidak terlalu familiar dengan perintah teks. Dengan hadirnya phpMyAdmin, proses tersebut menjadi jauh lebih mudah dan dapat dilakukan dengan antarmuka visual yang intuitif. Hal ini membuat phpMyAdmin menjadi pilihan populer di kalangan pengembang web, administrator sistem, dan bahkan pengguna non-teknis yang ingin mengelola basis data MySQL mereka.

Salah satu keunggulan utama phpMyAdmin adalah kemudahan penggunaan yang ditawarkannya. Pengguna dapat melakukan hampir semua operasi yang diperlukan untuk mengelola basis data MySQL tanpa harus menulis kode SQL secara manual. PhpMyAdmin menyediakan antarmuka grafis yang memungkinkan pengguna untuk membuat, mengedit, dan menghapus basis data, tabel, dan kolom dengan mudah. Dengan navigasi yang sederhana dan banyaknya opsi yang jelas terlihat, phpMyAdmin mengurangi kesulitan dalam mengelola basis data yang besar dan kompleks. Ini adalah salah satu alasan mengapa phpMyAdmin sangat populer di kalangan pengembang dan administrator basis data di seluruh dunia.

Fungsi-fungsi dasar yang dapat dilakukan melalui phpMyAdmin meliputi pembuatan dan pengelolaan basis data, tabel, kolom, serta indeks. Dalam proses pembuatan basis data, pengguna dapat menentukan nama dan karakter set basis data dengan memilih opsi yang sudah tersedia melalui menu drop-down. Begitu basis data dibuat, pengguna dapat mulai membuat tabel yang akan menampung data. PhpMyAdmin menyediakan antarmuka untuk menambahkan kolom-kolom dalam tabel, menentukan tipe data untuk setiap kolom, serta menetapkan kunci utama dan hubungan antar kolom. Proses ini yang sebelumnya memerlukan pengetahuan teknis tentang SQL menjadi lebih sederhana dan dapat dilakukan dengan mudah melalui antarmuka grafis.

Selain itu, phpMyAdmin memungkinkan pengguna untuk mengelola relasi antar tabel dalam basis data. Relasi antara tabel sangat penting dalam basis data relasional karena memungkinkan data dihubungkan satu sama lain secara logis. phpMyAdmin mempermudah pembuatan relasi ini dengan menyediakan antarmuka grafis untuk menetapkan kunci asing antara tabel yang berbeda. Hal ini memungkinkan pengguna untuk memanipulasi dan menghubungkan data dengan cara yang efisien, tanpa harus menulis



perintah SQL secara manual. Dengan menggunakan phpMyAdmin, pengelolaan relasi antar tabel menjadi lebih mudah dipahami dan lebih cepat untuk dilaksanakan.

phpMyAdmin juga mendukung pengelolaan indeks, yang sangat penting untuk meningkatkan kinerja query pada basis data yang besar. Indeks berfungsi untuk mempercepat pencarian data dalam tabel dengan menyediakan struktur yang memungkinkan sistem basis data untuk menemukan baris tertentu dengan lebih cepat. Melalui phpMyAdmin, pengguna dapat membuat, mengedit, dan menghapus indeks dengan mudah, sehingga kinerja query pada basis data yang besar dapat ditingkatkan secara signifikan. Hal ini memungkinkan phpMyAdmin untuk digunakan tidak hanya oleh pengembang yang bekerja dengan basis data kecil, tetapi juga oleh mereka yang mengelola basis data yang lebih besar dan lebih kompleks.

Selain fitur-fitur di atas, phpMyAdmin juga memungkinkan pengguna untuk mengelola pengguna dan perizinan dalam basis data. Pengelolaan pengguna adalah fitur penting dalam setiap sistem basis data, terutama ketika ada banyak orang yang terlibat dalam pengelolaan dan pemeliharaan basis data. Dengan phpMyAdmin, administrator basis data dapat membuat dan mengelola akun pengguna, menetapkan hak akses untuk setiap pengguna, serta mengontrol jenis operasi yang dapat dilakukan oleh masing-masing pengguna. Pengguna dapat diberi izin untuk membaca, menulis, mengedit, atau menghapus data dalam basis data, sesuai dengan peran dan tanggung jawab mereka dalam organisasi. Hal ini meningkatkan kontrol dan keamanan dalam pengelolaan basis data, memastikan bahwa hanya pengguna yang berwenang yang dapat melakukan perubahan pada data.

Keamanan dalam phpMyAdmin juga menjadi perhatian penting. Mengingat phpMyAdmin sering digunakan untuk mengelola basis data secara jarak jauh melalui web, sangat penting untuk memastikan bahwa alat ini aman dari potensi ancaman keamanan. PhpMyAdmin mendukung berbagai metode autentikasi yang dapat digunakan untuk mengamankan akses, termasuk autentikasi berbasis username dan password. Selain itu, phpMyAdmin juga memungkinkan administrator untuk mengonfigurasi akses hanya dari alamat IP tertentu, meningkatkan lapisan keamanan lebih lanjut. Fitur keamanan ini sangat penting terutama dalam pengelolaan basis data yang menyimpan informasi sensitif atau kritis.

PhpMyAdmin juga menyediakan kemampuan untuk melakukan impor dan ekspor data dalam berbagai format. Fitur impor memungkinkan pengguna untuk mengunggah data dari file eksternal, seperti file CSV atau SQL, ke dalam basis data MySQL. Sebaliknya, fitur ekspor memungkinkan pengguna untuk mengambil data dari basis data dan menyimpannya dalam format yang dapat digunakan di aplikasi lain. Ini sangat berguna untuk migrasi data, cadangan data, atau berbagi data antar sistem. PhpMyAdmin mendukung berbagai format ekspor, termasuk SQL, CSV, Excel, dan lainnya, yang mempermudah integrasi data dengan aplikasi pihak ketiga atau untuk kebutuhan analisis lebih lanjut.

Dalam hal pemeliharaan basis data, phpMyAdmin juga menawarkan beberapa alat yang dapat membantu dalam memeriksa dan memperbaiki masalah yang mungkin terjadi pada basis data. Misalnya, phpMyAdmin dapat digunakan untuk memeriksa integritas basis data, memeriksa apakah tabel-tabel dalam basis data mengandung data yang rusak atau tidak konsisten. PhpMyAdmin juga memungkinkan pengguna untuk mengoptimalkan tabel, yang dapat meningkatkan kinerja basis data secara keseluruhan dengan mengurangi



fragmentasi data. Fitur pemeliharaan ini membantu menjaga basis data tetap dalam kondisi optimal dan berfungsi dengan baik seiring waktu.

Sebagai alat berbasis web, phpMyAdmin dapat diakses dari berbagai perangkat yang terhubung ke jaringan. Hal ini memungkinkan administrator basis data untuk mengelola basis data mereka dari jarak jauh, yang sangat berguna terutama ketika mengelola basis data yang terletak di server jarak jauh. Dengan phpMyAdmin, pengguna tidak lagi perlu mengakses server fisik untuk melakukan perubahan pada basis data. Antarmuka berbasis web ini memberikan kenyamanan dan fleksibilitas lebih bagi pengguna untuk mengelola basis data mereka kapan saja dan di mana saja, selama mereka memiliki akses ke internet.

Secara keseluruhan, phpMyAdmin adalah alat yang sangat berguna dan serbaguna untuk mengelola basis data MySQL. Keunggulan utama phpMyAdmin adalah kemudahan penggunaan dan antarmuka grafis yang memungkinkan pengguna dari berbagai tingkat keahlian untuk mengelola basis data dengan lebih mudah. Dukungan phpMyAdmin untuk berbagai operasi MySQL, termasuk pengelolaan basis data, tabel, kolom, relasi, indeks, pengguna, dan perizinan, menjadikannya pilihan populer bagi banyak pengembang dan administrator basis data di seluruh dunia. Kemudahan akses dan fleksibilitas yang ditawarkan oleh phpMyAdmin semakin memperkuat posisinya sebagai salah satu alat manajemen basis data yang paling banyak digunakan.

5. Metode Waterfall

Metode Waterfall adalah salah satu pendekatan klasik dalam pengembangan perangkat lunak yang memfokuskan pada proses bertahap dan linier. Dalam model ini, setiap tahap pengembangan perangkat lunak harus diselesaikan sebelum tahap berikutnya dapat dimulai. Secara garis besar, model ini mencakup serangkaian langkah yang terstruktur dan jelas, mulai dari tahap analisis kebutuhan hingga pemeliharaan produk akhir. Hal ini memungkinkan tim pengembang untuk mengikuti alur yang sistematis, memastikan setiap tahapan dilakukan dengan cermat dan tidak ada langkah yang terlewatkan.

Menurut Pressman (2015), pendekatan Waterfall sangat efektif untuk proyek yang memiliki persyaratan yang sudah terdefinisi dengan jelas dan tidak cenderung berubah. Misalnya, dalam pembuatan aplikasi kasir berbasis ponsel, proyek ini biasanya memiliki spesifikasi yang sudah ditentukan di awal, sehingga model Waterfall menjadi sangat relevan. Dalam konteks ini, pengembangan dimulai dengan analisis kebutuhan yang mendalam, yang akan menggambarkan secara rinci semua fitur dan fungsi yang diinginkan oleh aplikasi. Setelah kebutuhan dipahami dan disepakati, langkah-langkah selanjutnya dapat dimulai.

Tahap pertama dalam metode Waterfall adalah requirements gathering atau pengumpulan kebutuhan. Pada tahap ini, pengembang bekerja sama dengan klien untuk mengidentifikasi dan mendokumentasikan semua kebutuhan fungsional dan non-fungsional aplikasi yang akan dibangun. Dalam pembuatan aplikasi kasir berbasis ponsel, misalnya, hal ini bisa meliputi kebutuhan fitur seperti pemrosesan pembayaran, integrasi dengan sistem inventaris, dan kemampuan untuk menghasilkan laporan transaksi secara real-time. Pengumpulan kebutuhan yang jelas dan komprehensif sangat penting untuk mencegah adanya perubahan besar selama proses pengembangan yang dapat mengganggu kelancaran proyek.

Setelah tahap pengumpulan kebutuhan selesai, proses berikutnya adalah desain sistem. Pada tahap ini, pengembang merancang arsitektur perangkat lunak, menentukan



struktur data, dan membuat skema antarmuka pengguna. Desain ini harus mengakomodasi semua persyaratan yang telah dikumpulkan sebelumnya. Untuk aplikasi kasir berbasis ponsel, desain dapat mencakup keputusan terkait pemilihan platform pengembangan, antarmuka pengguna yang ramah dan intuitif, serta cara aplikasi akan berinteraksi dengan sistem backend yang menyimpan data transaksi. Desain yang matang akan mempermudah proses pengembangan dan meminimalkan potensi kesalahan pada tahap berikutnya.

Setelah desain selesai, langkah selanjutnya adalah tahap implementasi atau pengkodean. Pada tahap ini, pengembang perangkat lunak mulai menulis kode sumber untuk aplikasi sesuai dengan desain yang telah disepakati. Pengembang akan membagi pekerjaan pengkodean ke dalam modul-modul yang lebih kecil, sehingga pengembangan dapat dilakukan secara bertahap. Dalam pembuatan aplikasi kasir, pengkodean dapat mencakup penulisan fungsi untuk menangani pembayaran, memproses item yang dibeli, serta mengelola transaksi dan laporan keuangan. Selama tahap ini, komunikasi yang baik antara anggota tim pengembang sangat penting untuk memastikan integrasi antar modul berjalan dengan lancar.

Setelah tahap pengkodean selesai, aplikasi memasuki tahap pengujian. Pada tahap ini, pengembang dan penguji perangkat lunak melakukan berbagai uji coba untuk memastikan aplikasi berfungsi sesuai dengan yang diharapkan. Uji coba dilakukan untuk menemukan dan memperbaiki bug atau masalah yang dapat mempengaruhi kinerja aplikasi. Pada aplikasi kasir, pengujian bisa mencakup verifikasi bahwa transaksi dihitung dengan benar, pembayaran diproses sesuai dengan metode yang dipilih, dan laporan yang dihasilkan sesuai dengan standar yang ditetapkan. Pengujian yang menyeluruh akan memastikan bahwa aplikasi siap digunakan oleh pengguna akhir tanpa gangguan.

Tahap berikutnya dalam model Waterfall adalah implementasi atau peluncuran produk. Setelah aplikasi melewati pengujian dan dinyatakan bebas dari masalah besar, aplikasi siap untuk diimplementasikan ke lingkungan produksi. Pada tahap ini, aplikasi dipasang di server atau perangkat pengguna dan siap untuk digunakan dalam situasi nyata. Untuk aplikasi kasir, ini berarti aplikasi tersebut akan mulai digunakan di tempat penjualan, dan sistem akan mulai memproses transaksi yang sebenarnya. Proses implementasi ini juga dapat mencakup pelatihan untuk pengguna akhir, seperti kasir atau staf toko, untuk memastikan mereka dapat menggunakan aplikasi dengan efektif dan efisien.

Tahap terakhir dari model Waterfall adalah pemeliharaan. Setelah aplikasi diimplementasikan, pemeliharaan dilakukan untuk memastikan aplikasi tetap berjalan dengan baik dan dapat beradaptasi dengan perubahan kebutuhan pengguna atau perbaikan bug yang ditemukan setelah penggunaan. Pemeliharaan melibatkan pemantauan aplikasi secara terus-menerus dan pembaruan yang diperlukan untuk menjaga aplikasi tetap up-to-date dan bebas dari masalah. Dalam konteks aplikasi kasir, pemeliharaan dapat mencakup pembaruan untuk menambah fitur baru, memperbaiki bug yang muncul, atau menyesuaikan aplikasi dengan perubahan kebijakan atau regulasi yang berlaku.

Keuntungan utama dari metode Waterfall adalah struktur yang jelas dan terdefinisi dengan baik, yang memudahkan pengelolaan proyek, terutama untuk proyek-proyek yang persyaratannya sudah stabil dan tidak banyak berubah. Namun, kelemahannya adalah ketergantungan pada tahap sebelumnya yang harus diselesaikan sepenuhnya sebelum melanjutkan ke tahap berikutnya. Hal ini membuat model ini kurang fleksibel dalam menangani perubahan yang tidak terduga selama proses pengembangan. Oleh karena itu,



meskipun metode Waterfall sangat sesuai untuk proyek yang stabil dan terdefinisi dengan jelas, model ini mungkin kurang ideal untuk proyek yang memerlukan fleksibilitas dan penyesuaian berkelanjutan.

6. Sistem Kasir

Perangkat lunak yang disebut sistem kasir digunakan untuk mengelola stok, mencatat transaksi, dan membuat laporan keuangan. Lauton dan Laudon (2020) menyatakan bahwa aplikasi kasir dan jenis sistem informasi lainnya sangat penting untuk meningkatkan efisiensi dan ketepatan proses bisnis, terutama di industri ritel.

Salah satu fungsi utama dari sistem kasir adalah mengelola stok barang. Sistem ini memungkinkan pemilik bisnis untuk memantau jumlah barang yang tersedia secara real-time, mengidentifikasi barang yang membutuhkan restocking, dan mengurangi risiko kekurangan atau kelebihan stok. Informasi stok yang akurat sangat penting untuk memastikan ketersediaan produk bagi pelanggan sekaligus mengoptimalkan inventaris. Lauton dan Laudon (2020) menekankan bahwa fitur ini memberikan keuntungan kompetitif karena memungkinkan bisnis untuk merespons kebutuhan pasar dengan cepat.

Selain itu, sistem kasir berperan penting dalam pencatatan transaksi penjualan. Setiap transaksi yang dilakukan, baik secara tunai maupun non-tunai, direkam secara otomatis dalam basis data. Proses ini tidak hanya meminimalkan kesalahan pencatatan tetapi juga memberikan transparansi dalam pelaporan keuangan. Data transaksi yang tercatat dapat digunakan untuk menganalisis tren penjualan, membantu pemilik bisnis dalam pengambilan keputusan strategis, seperti penentuan harga, promosi, atau pengadaan barang.

Fungsi lain dari sistem kasir adalah pembuatan laporan keuangan yang akurat dan terperinci. Sistem ini mampu menghasilkan laporan penjualan harian, mingguan, atau bulanan, yang memudahkan pemilik usaha dalam memantau performa bisnisnya. Lauton dan Laudon (2020) mencatat bahwa laporan keuangan yang dihasilkan oleh sistem kasir membantu memastikan kepatuhan terhadap regulasi perpajakan serta memberikan wawasan mendalam tentang keuntungan dan kerugian usaha.

Sistem kasir modern sering kali berbasis cloud, yang memungkinkan akses data secara real-time dari berbagai perangkat. Teknologi ini memberikan fleksibilitas bagi pemilik bisnis untuk memantau operasional usahanya kapan saja dan di mana saja. Lauton dan Laudon (2020) menyoroti bahwa sistem berbasis cloud juga menawarkan tingkat keamanan data yang lebih tinggi karena dilengkapi dengan fitur backup otomatis dan perlindungan dari kehilangan data akibat kerusakan perangkat keras.

Selain efisiensi dan akurasi, sistem kasir juga dirancang untuk meningkatkan pengalaman pelanggan. Dengan kemampuan untuk memproses transaksi dengan cepat dan menyediakan informasi yang akurat, sistem ini membantu mengurangi waktu tunggu pelanggan di kasir. Hal ini berkontribusi pada kepuasan pelanggan, yang merupakan faktor penting dalam mempertahankan loyalitas mereka terhadap bisnis.

Integrasi dengan alat pembayaran digital adalah salah satu fitur unggulan dari sistem kasir modern. Lauton dan Laudon (2020) menyatakan bahwa integrasi ini memungkinkan bisnis untuk menerima berbagai metode pembayaran, seperti kartu kredit, e-wallet, atau transfer bank, yang semakin diminati oleh pelanggan di era digital. Dengan menyediakan opsi pembayaran yang beragam, bisnis dapat menjangkau lebih banyak pelanggan dan meningkatkan volume penjualan.



Secara keseluruhan, sistem kasir bukan hanya alat untuk mencatat transaksi tetapi juga solusi teknologi yang menyeluruh untuk manajemen bisnis. Dengan kemampuannya untuk mengelola stok, mencatat transaksi, menghasilkan laporan, dan meningkatkan pengalaman pelanggan, sistem kasir menjadi elemen kunci dalam mendukung pertumbuhan bisnis. Lauton dan Laudon (2020) menegaskan bahwa implementasi sistem kasir yang efektif dapat membantu bisnis menghadapi tantangan operasional sekaligus menciptakan peluang baru untuk berkembang di pasar yang kompetitif.

7. Foreign Key Dalam Basis Data

Foreign key merupakan atribut dalam sebuah tabel yang digunakan untuk menjalin hubungan antar tabel dalam basis data. Korth dan Silberschatz (2011) mengemukakan bahwa peran utama dari foreign key adalah untuk menjaga integritas referensial, yang memastikan bahwa nilai dalam satu tabel hanya dapat mengacu pada nilai yang valid dan ada di tabel lain. Dalam konteks basis data relasional, foreign key berfungsi untuk memastikan konsistensi data antar tabel yang saling terkait, menghindari munculnya data yang tidak valid atau tidak terhubung dengan baik. Oleh karena itu, foreign key memainkan peran penting dalam menjaga keakuratan dan integritas data dalam sistem basis data.

Dalam implementasinya, foreign key sering kali digunakan dalam berbagai aplikasi yang membutuhkan hubungan antar tabel, seperti dalam aplikasi kasir. Misalnya, dalam aplikasi kasir, terdapat beberapa tabel yang harus saling terhubung, seperti tabel transaksi, pengguna, dan produk. Foreign key digunakan untuk menghubungkan tabel transaksi dengan tabel pengguna, yang berisi informasi pelanggan, serta tabel produk, yang menyimpan informasi tentang barang yang dijual. Dengan adanya foreign key, setiap transaksi yang tercatat dalam aplikasi kasir dapat dengan mudah dikaitkan dengan pengguna dan produk yang sesuai, memastikan bahwa setiap transaksi memiliki data yang valid dan terstruktur dengan baik.

Penggunaan foreign key dalam aplikasi kasir tidak hanya memastikan keterhubungan antar data, tetapi juga menjaga integritas dan akurasi data transaksi yang tercatat. Jika terjadi upaya untuk menambahkan data yang tidak valid, seperti memasukkan ID pengguna atau produk yang tidak ada dalam tabel terkait, sistem akan menolak tindakan tersebut, mencegah terjadinya inkonsistensi data. Hal ini sangat penting dalam operasi harian sebuah aplikasi kasir, karena setiap transaksi harus memiliki data yang valid dan terhubung dengan entitas yang sesuai. Dengan demikian, foreign key memastikan bahwa aplikasi kasir dapat berfungsi dengan lancar dan data yang dihasilkan dapat dipercaya.

Konsep foreign key sangat penting untuk menghindari inkonsistensi data yang dapat terjadi akibat kesalahan input atau penghapusan data yang tidak terkoordinasi. Sebagai contoh, jika sebuah transaksi tercatat di tabel transaksi tetapi pengguna yang melakukan transaksi tersebut dihapus dari tabel pengguna, maka akan terjadi ketidaksesuaian data. Dengan adanya foreign key, sistem akan mencegah penghapusan data di tabel pengguna selama masih ada referensi aktif di tabel transaksi. Hal ini memberikan jaminan bahwa data antar-tabel tetap konsisten dan saling terhubung secara logis.

Penggunaan foreign key juga mempermudah pengelolaan data dalam aplikasi kasir. Dalam tabel transaksi, misalnya, foreign key yang merujuk ke tabel pengguna memungkinkan sistem untuk mengidentifikasi pengguna yang melakukan transaksi tertentu. Selain itu, foreign key yang menghubungkan tabel transaksi dengan tabel produk



memungkinkan sistem untuk mencatat detail barang yang dibeli dalam setiap transaksi. Relasi ini memastikan bahwa semua informasi.

Foreign key memiliki beberapa aturan yang harus diikuti untuk menjaga integritas referensial. Salah satunya adalah aturan cascade delete atau cascade update, di mana perubahan pada data di tabel induk secara otomatis diterapkan pada data yang terkait di tabel anak. Misalnya, jika pengguna memperbarui informasi identitasnya di tabel pengguna, maka informasi tersebut secara otomatis diperbarui di tabel transaksi. Korth dan Silberschatz (2011) menyebutkan bahwa aturan ini membantu mengurangi redundansi data dan memastikan bahwa semua tabel dalam basis data tetap sinkron.

Selain itu, foreign key juga berfungsi untuk mencegah duplikasi data yang tidak diperlukan. Sebagai contoh, tanpa foreign key, informasi produk yang sama mungkin harus dicatat berulang kali di setiap transaksi. Dengan adanya foreign key, data produk hanya perlu dicatat satu kali di tabel produk, sementara tabel transaksi cukup mencantumkan referensi ke ID produk. Pendekatan ini tidak hanya menghemat ruang penyimpanan tetapi juga meningkatkan efisiensi sistem secara keseluruhan.

Dalam aplikasi kasir, implementasi foreign key memerlukan desain basis data yang matang. Entity Relationship Diagram (ERD) sering digunakan untuk memvisualisasikan relasi antar-tabel sebelum diimplementasikan. ERD ini menggambarkan bagaimana foreign key menghubungkan entitas-entitas seperti pengguna, produk, transaksi, dan detail transaksi. Visualisasi ini memudahkan pengembang untuk memahami struktur relasional basis data dan mengidentifikasi potensi masalah yang mungkin muncul dalam proses desain.

Proses implementasi foreign key dalam basis data biasanya dilakukan dengan menggunakan alat manajemen basis data seperti phpMyAdmin atau bahasa SQL. Dalam SQL, foreign key didefinisikan pada saat pembuatan tabel dengan menggunakan perintah FOREIGN KEY yang diikuti oleh referensi ke tabel lain. Misalnya, perintah SQL untuk menghubungkan tabel transaksi dengan tabel pengguna akan mencantumkan sintaks seperti FOREIGN KEY (user_id) REFERENCES users(id). Sintaks ini memastikan bahwa ID pengguna dalam tabel transaksi harus sesuai dengan ID pengguna yang valid di tabel pengguna.

Sistem kasir modern yang berbasis cloud juga memanfaatkan foreign key untuk mendukung pengelolaan data secara real-time. Dengan teknologi ini, data transaksi yang direkam di satu lokasi dapat segera diakses dan diverifikasi di lokasi lain tanpa risiko inkonsistensi. Korth dan Silberschatz (2011) mencatat bahwa foreign key sangat penting dalam sistem berbasis cloud karena membantu memastikan bahwa data tetap konsisten meskipun diakses oleh banyak pengguna secara bersamaan.

Secara keseluruhan, foreign key adalah elemen kunci dalam desain basis data yang memungkinkan pengelolaan data yang efisien, konsisten, dan dapat diandalkan. Dalam aplikasi kasir, penggunaan foreign key membantu mengintegrasikan data transaksi, pengguna, dan produk, sehingga mendukung operasional bisnis yang lebih efektif. Korth dan Silberschatz (2011) menegaskan bahwa penerapan foreign key yang tepat tidak hanya meningkatkan efisiensi teknis tetapi juga memberikan nilai tambah strategis bagi pengelolaan bisnis secara keseluruhan.

Penggunaan foreign key dalam aplikasi kasir memiliki manfaat yang sangat besar, terutama dalam hal integrasi data yang melibatkan transaksi, pengguna, dan produk. Setiap transaksi yang tercatat dalam sistem kasir dapat dihubungkan dengan pengguna yang



melakukan pembelian dan produk yang dibeli, berkat penerapan foreign key. Dengan demikian, basis data dapat mengelola dan menyajikan informasi yang relevan dan akurat dengan cara yang efisien, memungkinkan bisnis untuk mendapatkan wawasan yang lebih jelas dan mudah dalam pengambilan keputusan. Selain itu, integrasi data yang baik juga memungkinkan pemantauan dan analisis yang lebih efektif terhadap operasional harian aplikasi kasir.

Selain efisiensi operasional, penggunaan foreign key juga sangat penting dalam menjaga konsistensi dan akurasi data yang terkait. Dalam aplikasi kasir, setiap transaksi yang melibatkan produk dan pengguna harus disertai dengan informasi yang sah, seperti ID pengguna yang valid dan produk yang tersedia di sistem. Foreign key memastikan bahwa tidak ada transaksi yang dapat dibuat tanpa adanya data yang tepat dan sah dari tabel terkait. Misalnya, jika seorang pengguna mencoba untuk melakukan transaksi dengan ID produk yang tidak ada di dalam tabel produk, sistem akan menolak transaksi tersebut, sehingga menghindari inkonsistensi dan kesalahan data.

Korth dan Silberschatz (2011) menjelaskan bahwa penerapan foreign key yang tepat dapat memberikan nilai tambah strategis dalam pengelolaan basis data. Dengan struktur yang jelas dan hubungan antar tabel yang kuat, data dapat dikelola lebih mudah dan lebih cepat. Dalam aplikasi kasir, penerapan foreign key memungkinkan pengelolaan informasi transaksi menjadi lebih efisien, memungkinkan pencarian data yang cepat dan terorganisir, serta mempermudah laporan dan analisis yang dilakukan secara berkala. Hal ini pada akhirnya akan berkontribusi pada penghematan waktu dan pengurangan biaya operasional, yang sangat penting dalam mendukung kelancaran bisnis.

Penerapan foreign key juga memainkan peran krusial dalam meningkatkan keamanan dan mencegah kesalahan input data. Ketika data pada tabel terkait tidak valid, sistem akan memblokir proses input yang tidak tepat, seperti transaksi dengan referensi data yang tidak ada atau salah. Misalnya, jika seorang kasir mencoba untuk memasukkan transaksi dengan ID pengguna yang tidak terdaftar atau ID produk yang tidak tersedia, foreign key akan memastikan bahwa hanya data yang sah yang diterima oleh sistem. Hal ini tidak hanya meningkatkan efisiensi operasional tetapi juga memastikan bahwa data yang diproses aman dan tidak mengarah pada kesalahan yang dapat merugikan bisnis.

Secara keseluruhan, foreign key bukan hanya alat teknis untuk menghubungkan data dalam sebuah basis data, tetapi juga merupakan elemen strategis yang mendukung pengelolaan data yang lebih baik dan pengambilan keputusan yang lebih informasional. Dalam aplikasi kasir, penerapan foreign key memberikan manfaat besar dengan mengintegrasikan berbagai aspek penting dari transaksi, pengguna, dan produk, serta membantu menjaga integritas dan akurasi data. Korth dan Silberschatz (2011) menegaskan bahwa penggunaan foreign key yang tepat bukan hanya berkontribusi pada efisiensi teknis, tetapi juga memberikan keuntungan kompetitif bagi pengelolaan bisnis secara keseluruhan, menjadikannya sebagai bagian yang tidak terpisahkan dari desain basis data yang efektif dan berkelanjutan.

METODOLOGI

Metode pengembangan perangkat lunak model Waterfall digunakan dalam penelitian ini. Menurut Pressman (2015), model Waterfall adalah pendekatan pengembangan perangkat lunak yang terstruktur dengan tahapan yang jelas. Tahapan-tahap ini termasuk:



1. Analisis kebutuhan sistem

Tahap analisis kebutuhan sistem merupakan langkah awal yang sangat penting dalam pengembangan perangkat lunak, termasuk aplikasi berbasis kasir. Tujuan utama dari tahap ini adalah untuk memahami kebutuhan pengguna secara mendalam serta mengidentifikasi elemen-elemen yang harus ada dalam aplikasi dan basis data yang akan dibangun. Dalam proses ini, pengembang harus memastikan bahwa sistem yang dirancang mampu memenuhi harapan dan kebutuhan operasional pengguna secara efektif. Langkah ini tidak hanya berfokus pada aspek teknis tetapi juga pada bagaimana sistem dapat memberikan solusi praktis terhadap permasalahan yang dihadapi oleh calon pengguna.

Pengumpulan data adalah komponen utama dalam analisis kebutuhan sistem. Wawancara dengan calon pengguna merupakan metode yang sering digunakan untuk mendapatkan informasi langsung tentang kebutuhan mereka. Dalam wawancara ini, pengembang dapat menggali informasi terkait alur kerja pengguna, jenis data yang sering dikelola, dan fitur-fitur yang diinginkan dalam aplikasi. Informasi ini kemudian dianalisis untuk menentukan komponen utama yang harus ada dalam sistem, seperti pengelolaan stok barang, pencatatan transaksi, dan pembuatan laporan penjualan. Hasil dari wawancara ini menjadi dasar bagi desain sistem yang akan dibuat.

Selain wawancara, tinjauan literatur juga menjadi bagian penting dari analisis kebutuhan. Literatur yang relevan memberikan panduan tentang praktik terbaik dalam desain aplikasi kasir dan basis data. Studi kasus dari aplikasi serupa dapat memberikan wawasan tentang fitur yang efektif dan potensi masalah yang perlu diantisipasi. Tinjauan ini juga membantu pengembang memahami standar industri, seperti bagaimana integritas data dapat dijaga atau bagaimana sistem dapat dioptimalkan untuk mendukung operasional bisnis secara real-time.

Hasil dari tahap analisis kebutuhan tidak hanya mencakup daftar kebutuhan fungsional tetapi juga kebutuhan non-fungsional. Kebutuhan fungsional meliputi fitur yang harus ada dalam sistem, seperti kemampuan mencatat transaksi atau mengelola stok barang. Sementara itu, kebutuhan non-fungsional mencakup aspek-aspek seperti kecepatan akses, skalabilitas, dan keamanan data. Dalam aplikasi kasir, kebutuhan non-fungsional seperti respons cepat terhadap transaksi pelanggan dan perlindungan data sensitif menjadi sangat krusial untuk memberikan pengalaman pengguna yang optimal.

Dokumentasi kebutuhan adalah langkah berikutnya setelah pengumpulan data. Dalam proses ini, kebutuhan pengguna yang telah diidentifikasi disusun secara terstruktur dan sistematis. Dokumentasi ini menjadi panduan bagi pengembang dalam tahap-tahap berikutnya, seperti desain dan implementasi sistem. Selain itu, dokumentasi ini juga mempermudah komunikasi antara pengembang dan calon pengguna, sehingga memastikan bahwa sistem yang dirancang sesuai dengan ekspektasi mereka. Dokumentasi kebutuhan sering kali mencakup diagram alur kerja, deskripsi fitur, dan spesifikasi teknis lainnya.

Secara keseluruhan, analisis kebutuhan sistem adalah fondasi utama dalam pengembangan aplikasi berbasis kasir. Tahap ini memastikan bahwa sistem dirancang untuk memenuhi kebutuhan spesifik pengguna sambil tetap sesuai dengan standar industri. Dengan memahami kebutuhan pengguna melalui wawancara dan tinjauan literatur, pengembang dapat menciptakan aplikasi yang tidak hanya fungsional tetapi juga memberikan nilai tambah bagi operasional bisnis. Keberhasilan tahap ini menjadi penentu



utama untuk memastikan bahwa sistem yang dibangun dapat beroperasi dengan efisien dan memberikan manfaat maksimal bagi penggunaanya.

2. Desain sistem

Alat pemodelan seperti Entity Relationship Diagram (ERD) digunakan untuk menentukan struktur data dan hubungannya. Diagram use case juga digunakan untuk menunjukkan interaksi antara pengguna dan sistem.

3. Implementasi basis data dalam phpMyAdmin

Untuk menerapkan desain ini, phpMyAdmin digunakan untuk membuat tabel basis data dan mendefinisikan relasi antar tabel. Bahasa pemrograman yang digunakan oleh aplikasi mobile dapat mengakses basis data ini melalui API.

4. Pengujian aplikasi

Sistem diuji untuk memastikan bahwa itu bekerja sesuai dengan spesifikasi. Pengujian ini meliputi validasi relasi antarkunci asing, pengujian integritas data, dan pengujian fungsionalitas aplikasi..

5. Pemeliharaan sistem

Setelah implementasi sistem selesai, tahap pemeliharaan dilakukan untuk memastikan aplikasi berjalan dengan lancar dan memperbaiki bug..

PERANCANGAN SISTEM

1. Entity Relationship Diagram (ERD)

1) ERD menggambarkan relasi antar entitas dalam sistem:

Users: Menyimpan informasi pengguna (admin dan kasir).

Produk: Menyimpan data produk, termasuk nama, kategori, harga, dan stok.

Transaksi: Mencatat data transaksi, termasuk pengguna yang melakukan transaksi dan total harga.

Detail Transaksi: Mencatat detail barang yang dibeli dalam setiap transaksi.

2) Struktur Basis Data di phpMyAdmin

Tabel-tabel berikut dirancang untuk mengimplementasikan ERD:

Tabel User

	#	Nama	Jenis	Penyortiran
☐	1	id_transaksi	int(11)	
☐	2	id_user	int(11)	
☐	3	tanggal	datetime	
☐	4	total_harga	decimal(10,2)	

Gambar 1. Tabel User



Tabel Produk

#	Nama	Jenis	Penyortiran	Atri
1	id_produk	int(11)		
2	nama_produk	varchar(100)	utf8mb4_general_ci	
3	kategori	varchar(50)	utf8mb4_general_ci	
4	harga	decimal(10,2)		
5	stock	int(11)		

Gambar 2. Tabel Produk

Tabel Transaksi

#	Nama	Jenis	Penyortiran
1	id_user	int(11)	
2	nama	varchar(100)	utf8mb4_general_ci
3	email	varchar(100)	utf8mb4_general_ci
4	password	varchar(255)	utf8mb4_general_ci
5	role	enum('admin', 'kasir')	utf8mb4_general_ci

Gambar 3. Tabel Transaksi

Tabel Detail Transaksi

#	Nama	Jenis	Penyortiran	Atribut
1	id_detail_transaksi	int(11)		
2	id_transaksi	int(11)		
3	id_produk	int(11)		
4	jumlah	int(11)		
5	subtotal	decimal(10,2)		

Gambar 4. Tabel Detail Transaksi

DAFTAR PUSTAKA

- Connolly, T., & Begg, C. (2015). Database Systems: A Practical Approach to Design, Implementation, and Management (6th ed.). Pearson Education.
- Elmasri, R., & Navathe, S. B. (2010). Fundamentals of Database Systems (6th ed.). Addison-Wesley.
- Eriana, E. S. (2020). Desain Basis Data dalam Aplikasi Kasir Berbasis Mobile. [Dapat ditemukan di Google Scholar, jurnal atau repository lokal universitas terkait].
- Pressman, R. S. (2014). Rekayasa Perangkat Lunak: Pendekatan Praktisi.
- Welling, L., & Thomson, L. (2009). PHP and MySQL Web Development (4th ed.). Addison-Wesley