

Penerapan Model View Controller Berbasis Framework Codeigniter Pada Sistem Penjualan (Studi Kasus : Toko Jaya Motor)

Maria Sri Wulandari, Rahayu Noveandini, dan Febriansyah

Sistem Informasi, STMIK Jakarta STI&K
Jl. BRI Radio Dalam Kebayoran Baru Jakarta Selatan
E-mail : mswuland@yahoo.com, ayu.noveandini@gmail.com, febri030412@gmail.com

Abstrak

Perkembangan teknologi informasi telah mengubah paradigma transaksi jual beli, yang sebelumnya mengharuskan interaksi langsung, kini dapat dilakukan secara online. Teknologi internet memberikan dampak signifikan pada dunia bisnis, memungkinkan calon pembeli untuk melihat produk, mengakses informasi, melakukan pemesanan, dan pembayaran melalui layar komputer dari rumah atau kantor. Kemudahan ini tidak hanya menghemat waktu dan biaya, tetapi juga memungkinkan pengambilan keputusan yang cepat tanpa harus mengunjungi toko fisik. Toko Jaya Motor, yang saat ini masih menerapkan sistem penjualan konvensional di mana pelanggan harus datang langsung ke toko, memerlukan pembaruan sistem untuk mengikuti tren bisnis modern. Oleh karena itu, dirancang dan dibangun sistem penjualan berbasis web menggunakan metode pengembangan sistem model System Development Life Cycle (SDLC) yang menekankan tahapan pengembangan yang berurutan dan sistematis. Sistem ini dirancang dengan pola arsitektur Model View Controller (MVC) menggunakan framework CodeIgniter dan Bootstrap untuk menciptakan website yang dinamis, responsif, dan user-friendly. Perancangan sistem menggunakan Unified Modeling Language (UML) untuk memodelkan struktur dan perilaku sistem, serta Class Diagram untuk menggambarkan relasi antar data dalam basis data. Selain itu, fitur keamanan yang andal juga diterapkan untuk melindungi data pelanggan dan transaksi. Dengan implementasi sistem penjualan berbasis web ini, diharapkan Toko Jaya Motor dapat meningkatkan efisiensi operasional dan memberikan pelayanan yang lebih baik kepada pelanggan. Selain itu, sistem ini diharapkan dapat meningkatkan daya saing Toko Jaya Motor di era digital yang semakin kompetitif, memenuhi kebutuhan dan ekspektasi pelanggan yang semakin mengutamakan kemudahan dan kecepatan dalam bertransaksi. Implementasi sistem ini juga membuka peluang bagi Toko Jaya Motor untuk memperluas jangkauan pasar dan menjangkau lebih banyak pelanggan potensial secara global.

Kata kunci : Sistem Penjualan, WEB, Model View Controller, SDLC, UML

Pendahuluan

Toko Jaya Motor saat ini menggunakan sistem manual dalam penyampaian informasi dan proses penjualan produk. Sistem ini menimbulkan beberapa masalah, seperti minimnya penjualan karena pelanggan harus datang langsung ke toko, pencatatan transaksi yang kurang efisien, dan komunikasi dengan pelanggan yang tidak optimal. Selain itu, proses transaksi sering tertunda karena pelanggan harus menunggu pelayanan, yang dapat menyebabkan penundaan atau pembatalan pembelian. Mengingat perkembangan teknologi dan tren bisnis modern yang semakin mengutamakan kemu-

dahan dan kecepatan dalam bertransaksi, diperlukan solusi yang lebih efisien untuk meningkatkan penjualan dan pelayanan pelanggan di Toko Jaya Motor. Oleh karena itu, dirancang dan dibangun sistem penjualan berbasis web yang dapat mengatasi masalah-masalah tersebut dan mendukung operasional toko secara lebih efektif. Tujuan dari penelitian ini adalah untuk merancang dan mengembangkan sistem penjualan berbasis web untuk Toko Jaya Motor. Sistem ini diharapkan dapat meningkatkan efisiensi operasional dan memberikan pelayanan yang lebih baik kepada pelanggan. Tujuan spesifik dari penelitian ini meliputi: Mengotomatisasi proses pendaf-

taran akun, pemesanan barang, dan sistem pembayaran; Mengelola pengiriman barang dengan menggunakan jasa pengiriman; Menyediakan fitur pembuatan laporan transaksi dan update status informasi kepada pelanggan; Memastikan keamanan data pelanggan dan transaksi; Meningkatkan daya saing Toko Jaya Motor di era digital.

Suendri dari Universitas Islam Negeri Sumatera Utara (2018), tujuan dari penelitian ini untuk melakukan pencatatan seluruh produk software yang dikembangkan oleh perusahaan yang bergerak dibidang perangkat lunak. Semakin banyaknya pelanggan yang menggunakan produk tersebut akan lebih memudahkan jika perusahaan mempunyai sebuah sistem yang bisa diakses secara realtime dan mencatat seluruh data-data penting software. Konsep Model View Controller (MVC) merupakan sebuah arsitektur dimana proses pada sistem dipisah menjadi tiga bagian dengan tugas-tugasnya. Penelitian ini diharapkan bisa membantu perusahaan perangkat lunak dalam melakukan pencatatan produk yang telah dijual kepada pelanggan[1].

Khana Wijaya, Andi Christian dari STMIK Prabumulih (2019), tujuan dari penelitian ini pembangunan website SMK Yayasan Bakti Prabumulih. Situs sekolah memiliki peran yang sangat penting dalam pendidikan. Yayasan SMK Bakti Prabumulih adalah salah satu sekolah swasta yang memiliki dua jurusan, sepeda motor otomotif dan teknik komputer serta teknik jaringan, tetapi sekolah ini tidak memiliki situs web sehingga banyak orang yang belum tahu mengingat sekolah ini masih baru. Dalam pembuatan situs web ini penulis juga menerapkan metode MVC sebagai opsi dalam pengembangan sistem[2].

Syaiful Anwar dari Universitas Bina Sarana Informatika (2020), penelitian ini menggunakan framework Codeigniter perancangan aplikasi penjualan ini dapat disusun menjadi lebih mudah dan lebih terarah pembagian struktur pemrogramannya. Konsep Model View Controller (MVC) memberikan penyusunan coding program lebih mudah dibuat konsepnya dan mempermudah pemeriksaan bagian-bagian dari coding yang dibuat. Desain tampilan aplikasi (View) dapat dibuat lebih dinamis dan menarik dengan konsep MVC ini[3].

Metode Penelitian

Metode pengembangan sistem menggunakan System Development Life Cycle (SDLC). Berikut tahapan sistem SDLC:

1. Tahap Perencanaan

Pada tahap ini dilakukan identifikasi sistem dan prosedur yang ingin dibuat, dengan menetapkan hasil seperti apa yang diharapkan dari sistem dan prosedur yang ingin dirancang[3].

2. Analisis Masalah

Mempelajari data yang diperoleh dari sistem yang sedang beroperasi, kemudian melakukan analisa sistem secara keseluruhan, serta permasalahan yang terjadi untuk menemukan jawaban apa penyebab yang sebenarnya dari masalah yang terjadi.

3. Perancangan Sistem

Pada tahap ini menjelaskan langkah-langkah perancangan aplikasi dimulai dari Struktur Navigasi, Use Case Diagram, dan Rancangan Input/Output[4]. Pada proses ini dimulai dengan mendesain tampilan di dalam komputer sebagai gambaran awal yang memperhatikan tampilan program.

4. Pembuatan Aplikasi

Pada tahap ini desain program diterjemahkan di dalam kode dengan menggunakan bahasa pemrograman web dengan menggunakan PHPMySQL[5]. Aplikasi yang dibuat langsung diuji, apakah sudah bekerja dengan sesuai yang diharapkan.

5. Uji Coba

Pada bagian ini dilakukan beberapa uji coba terhadap aplikasi dengan menggunakan metode Black Box Testing. Black Box Testing merupakan teknik pengujian perangkat lunak yang berfokus pada spesifikasi fungsional dari perangkat lunak. Black Box Testing memiliki tiga tipe pengujian yaitu Functional testing, Non-functional testing, dan Regression testing[6]. Dalam pengujian sistem ini menggunakan tipe Functional testing. Functional testing adalah proses pengujian terhadap fungsi atau fitur spesifik sebuah software. Functional testing berfokus pada pengujian aspek-aspek paling penting dari sebuah software dan integrasi antara komponen-komponen utamanya, untuk menguji sistemnya secara keseluruhan.

6. Implementasi

Mengidentifikasi kebutuhan pengguna terkait dengan sistem yang dibuat dan mengimplementasikan rancangan dari tahap-tahap sebelumnya seperti pembuatan database yang sesuai dengan skema rancangan, pembuatan aplikasi berdasarkan desain sistem, dan pengujian serta perbaikan aplikasi (debugging).

7. Pemeliharaan

Pada tahap ini dilakukan suatu upaya untuk memperbaiki, menjaga, menanggulangi, dan mengembangkan sistem yang ada. Pemeliharaan ini di perlukan untuk meningkatkan efisiensi dan efektivitas kinerja sistem agar dalam penggunaannya dapat optimal. Sistem perlu dipelihara karena beberapa hal, seperti

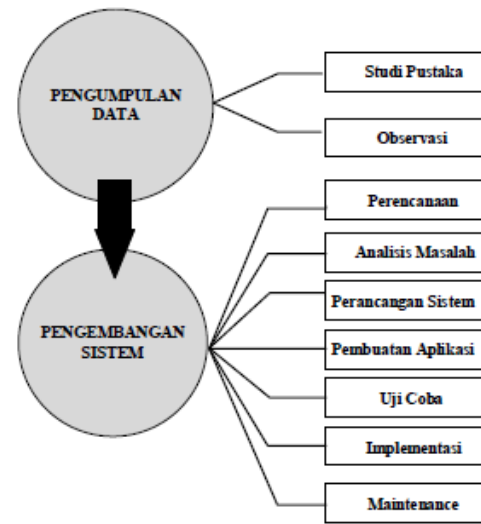
adanya kesalahan yang belum terdeteksi sehingga kesalahan tersebut perlu diperbaiki dan mengalami perubahan karena permintaan dari pengguna .

MVC atau *Model View Controller* dengan cepat menjadi praktik standar industri yang digunakan di setiap lingkungan pengembangan yang modern. MVC adalah salah satu pola desain arsitektur paling populer untuk aplikasi web. Model adalah lapisan abstraksi, yang menyajikan data dalam suatu aplikasi. Secara umum, biasanya sesuai dengan catatan dalam database DBMS atau semua jenis mesin penyimpanan seperti NoSQL atau Microsoft Access.

View adalah lapisan yang menyajikan data dalam format yang sesuai dan dalam keinginan pengguna, dan biasanya disajikan sebagai halaman web HTML. Ini adalah presentasi visual akhir dari data dalam aplikasi web yang *Controller* telah terima berdasarkan perubahan dari Model. Pada dasarnya *Controller* adalah layer yang menangani permintaan dan mengembalikan respons yang sesuai ke *View* yang dituju. Oleh karena itu, *Controller* bertindak sebagai koordinator antara *View* dan *Model*. Dengan kata lain, itu adalah lapisan logis dari aplikasi. Misalnya, *Controller* dapat menerima pengiriman formulir dari *View* dan melakukan validasi, menulis data yang valid ke database, dan mengarahkan pengguna ke route lain. MVC merangkul data bersama dengan pemrosesan (Model) dan mengisolasinya dari proses manipulasi (Controller) dan presentasi (*View*) untuk diwakili pada *User Interface*.

MVC mengikuti pendekatan *Layering* yang paling umum. *Layering* tidak lain adalah pemisahan logis dari kode kita ke fungsi di kelas yang berbeda. Pendekatan ini dikenal dan paling banyak diterima. *Model* artinya logika bisnis dari aplikasi dan merupakan inti dari aplikasi. *View* adalah antarmuka pengguna dari controller dan wajah public dari respon peristiwa pengguna. Banyak tampilan untuk berbagai tujuan dirancang dan dikembangkan dengan HTML, *Cascading style sheets* (CSS), Javascript, dan lain-lain. Komponen *Controller* mengimplementasikan alur *Control* antara *View* dan *Model*.

Unified Modelling Language (UML) bukanlah suatu proses melainkan bahasa permodelan secara grafis untuk menspesifikasikan, memvisualisasikan, membangun dan mendokumentasikan seluruh artefak sistem perangkat lunak. Penggunaan model ini bertujuan mengidentifikasi bagian – bagian yang termasuk dalam lingkup sistem yang dibahas dan bagaimana hubungan antara sistem dengan subsistem maupun sistem lain diluarnya. Tahapan Metode Penelitian dapat dilihat pada Gambar 1.



Gambar 1: Tahapan Metode Penelitian

Hasil dan Pembahasan

Model View Controller (MVC)

Model View Controller (MVC) adalah konsep yang terpisah menjadi tiga bagian, yakni model, view, controller. Tiga komponen ini sebenarnya mempunyai pengertian dan tugasnya masing-masing dalam mendukung pembangunan web. Komponen model berhubungan dengan database dan interaksi[7]. Sebagai contoh dalam web ini mempunyai fitur keranjang belanja. Model yang akan menentukan apa saja data yang terdapat pada keranjang belanja tersebut. Komponen selanjutnya *view* yang berkaitan langsung dengan pengguna. *View* akan menentukan bagaimana daftar disajikan pada pengguna dan menerima data yang perlu ditampilkan dari model. Konsep terakhir dari bagian MVC adalah controller. *Controller* berisikan logika yang dapat memperbarui model atau tampilan sebagai respon dari tindakan pengguna web. *Controller* menjadi bagian yang bertugas untuk menghapus atau menambahkan suatu barang ke dalam fitur keranjang belanja pada web ini. *Model View Controller* (MVC) adalah sebuah pola desain perangkat lunak yang memisahkan aplikasi ke dalam tiga komponen utama: *Model*, *View*, dan *Controller*. Ini membantu dalam memisahkan logika bisnis dari tampilan antarmuka pengguna dan memungkinkan pengembangan, pengujian, dan pemeliharaan sistem yang lebih mudah.

Berikut adalah analisis penerapan MVC pada sistem penjualan:

1. Model

Deskripsi: Model mewakili data dan logika bisnis aplikasi. Pada sistem penjualan, Model akan mencakup semua data yang berhubungan dengan produk, pelanggan, pesanan, inventaris, dan transaksi. Implementasi: Data Produk memberikan informasi tentang produk seperti nama, deskripsi, harga, dan

stok. Data Pelanggan memberikan informasi pelanggan seperti nama, alamat, dan riwayat pembelian. Data Pesanan menjelaskan data mengenai pesanan termasuk item yang dipesan, jumlah, dan status pesanan. Data Inventaris menyampaikan informasi status stok produk dan manajemen inventaris. Data Transaksi adalah data transaksi pembayaran dan metode pembayaran. Manfaat: Pemisahan data dari antarmuka pengguna memungkinkan pengelolaan data yang lebih efektif dan pengujian yang lebih mudah. Perubahan pada data atau logika bisnis tidak mempengaruhi tampilan antarmuka pengguna.

2. View

Deskripsi: View adalah komponen yang bertanggung jawab untuk menampilkan data dari Model kepada pengguna. Pada sistem penjualan, View akan mencakup antarmuka pengguna seperti halaman produk, keranjang belanja, formulir pesanan, dan riwayat transaksi. Implementasi: pada Halaman Produk menampilkan daftar produk, detail produk, dan opsi pencarian. Pada Keranjang Belanja menampilkan produk yang dipilih oleh pelanggan dan total harga. Halaman Formulir Pesanan memungkinkan pelanggan untuk memasukkan informasi pengiriman dan pembayaran. Pada Riwayat Transaksi menampilkan riwayat pembelian pelanggan. Manfaat: Pemisahan tampilan dari logika bisnis memudahkan pengembangan dan modifikasi antarmuka pengguna. Desainer dapat bekerja pada antarmuka tanpa mempengaruhi logika bisnis.

3. Controller

Deskripsi: Controller adalah komponen yang mengelola aliran data antara Model dan View. Controller menerima input dari pengguna melalui View, memprosesnya (dengan bantuan Model), dan kemudian memperbarui View sesuai dengan hasilnya. Implementasi: Proses Pengolahan Pesanan adalah mengambil data pesanan dari View, memvalidasinya, dan memperbarui Model. Proses pengelolaan Inventaris adalah memperbarui stok produk setelah transaksi selesai. Autentikasi dan Otorisasi adalah mengelola proses login dan hak akses pengguna. Interaksi Pelanggan adalah mengelola interaksi pelanggan seperti pencarian produk dan penambahan produk ke keranjang belanja. Manfaat: Pemisahan logika kontrol dari tampilan dan data memungkinkan kode yang lebih bersih dan terorganisir. Memungkinkan pengembangan fitur baru tanpa harus merombak seluruh sistem.

Studi Kasus Implementasi MVC dalam Sistem Penjualan.

Implementasi MVC dalam sistem penjualan terdiri dari beberapa tahap, yakni :

1. Pembuatan Pesanan

View: Pelanggan memasukkan informasi pesanan melalui antarmuka pengguna. *Controller:* Mengambil data dari View, memvalidasi data, dan mengirimkannya ke Model. *Model:* Menyimpan data pesanan ke database dan memperbarui status stok. *Controller:* Mengambil respon dari Model dan memperbarui View untuk menampilkan konfirmasi pesanan.

2. Pengelolaan Inventaris

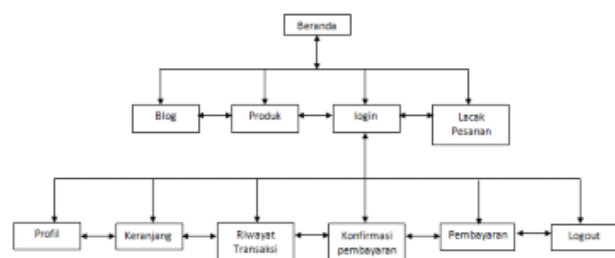
View: Admin melihat dan mengedit informasi produk melalui antarmuka pengguna. *Controller:* Mengambil data yang dimasukkan oleh admin, memvalidasi, dan mengirimkannya ke Model. *Model:* Memperbarui informasi produk di database. *Controller:* Mengambil respon dari Model dan memperbarui View untuk menampilkan informasi produk yang diperbarui.

3. Riwayat Transaksi

View: Pelanggan meminta riwayat transaksi melalui antarmuka pengguna. *Controller:* Mengambil permintaan dari View dan meminta data dari Model. *Model:* Mengambil data transaksi dari database dan mengembalikannya ke Controller. *Controller:* Mengirimkan data transaksi ke View untuk ditampilkan kepada pelanggan. Dengan mengadopsi MVC, sistem penjualan dapat lebih terstruktur, mudah dikelola, dan skalabel, sehingga memudahkan pengembangan dan pemeliharaan jangka panjang

Struktur Navigasi Pengguna

Struktur navigasi pengguna dapat dilihat pada Gambar 2, dimana pengguna dapat langsung mengakses menu yang terdapat pada web, jika pengguna ingin melakukan transaksi, pengguna harus melakukan registrasi kemudian login untuk menyelesaikan proses transaksi. Jenis struktur navigasi yang digunakan yaitu struktur navigasi campuran [8].

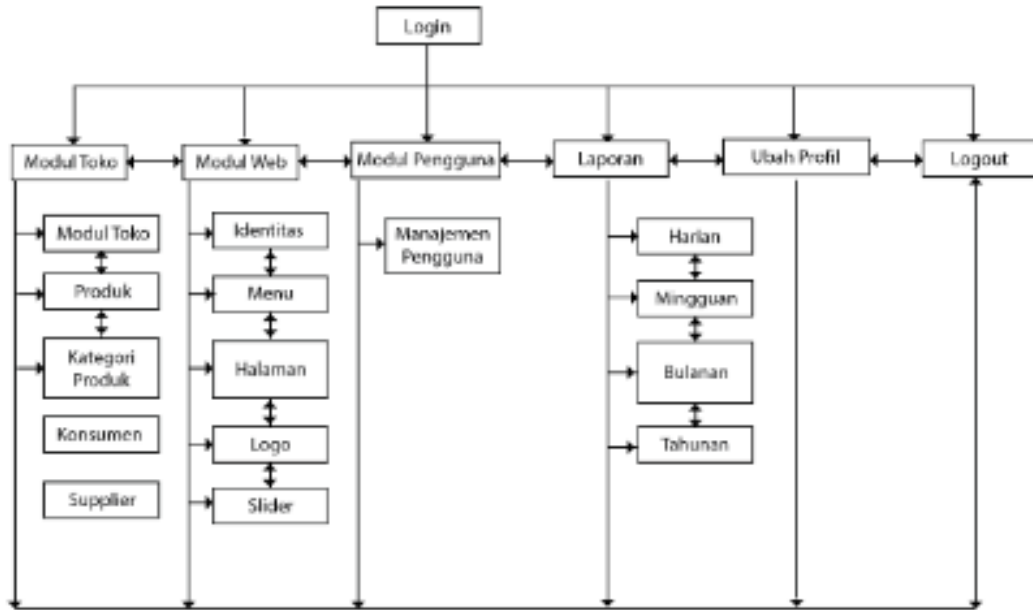


Gambar 2: Struktur Navigasi Pengguna

Struktur Navigasi Admin

Struktur navigasi admin dapat dilihat pada Gambar 3, admin dapat langsung mengakses halaman-

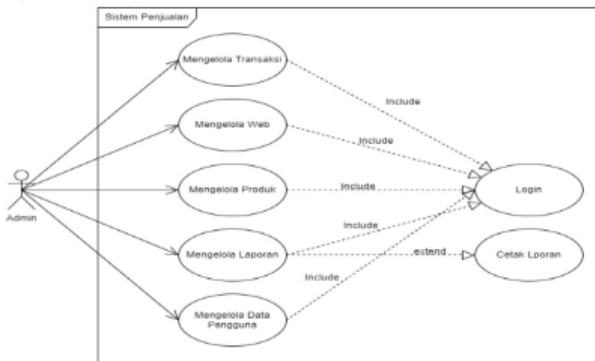
halaman untuk mengelola web dan transaksi yang terjadi.



Gambar 3: Struktur Navigasi Admin

Use Case Diagram Admin

Use case diagram admin adalah kegiatan interaksi antara admin dengan sistem. Sesuai dengan Gambar 3.4 use case diagram, Aktor sebagai admin harus masuk dan login ke panel administrator, setelah itu admin dapat melihat bagian dashboard ataupun beranda, transaksi yang terdiri dari detail transaksi, cetak transaksi dan update transaksi. Menu produk yang terdiri tambah, ubah, dan menghapus data kategori produk dan data produk. Selanjutnya ada menu data pengguna, dan juga ada laporan serta menu keluar. Use case diagram pada admin dapat dilihat pada Gambar 4.

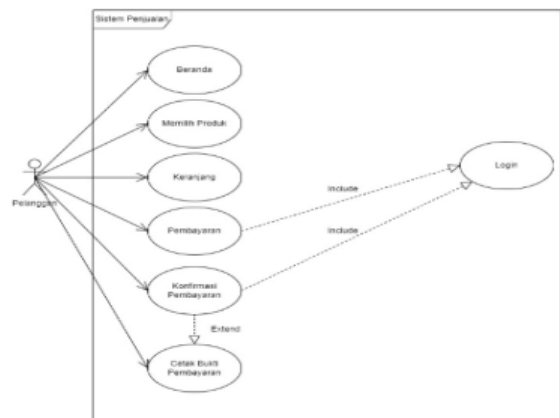


Gambar 4: Use Case Diagram Admin

Use Case Diagram Pengguna

Use case diagram pengguna adalah kegiatan interaksi antara pengguna dengan sistem. User dapat

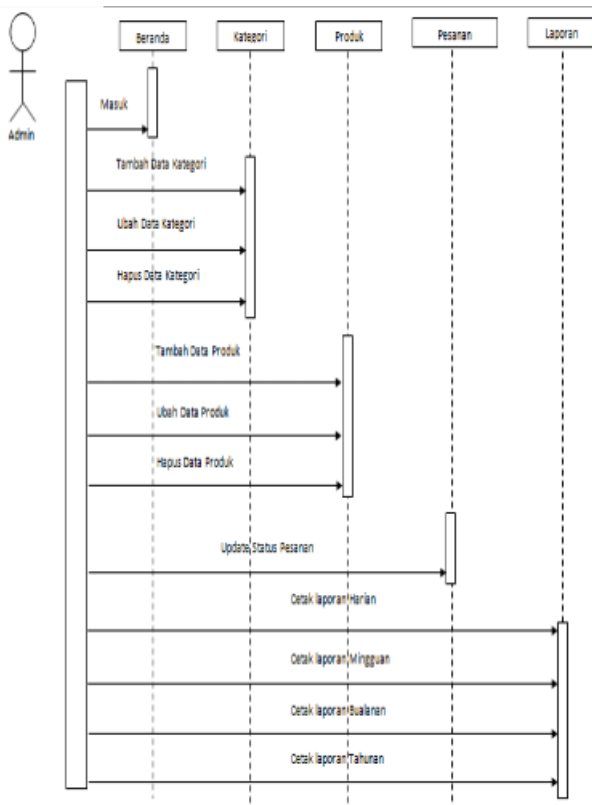
pat mengakses langsung ke dalam halaman beranda, produk, tentang, kontak, keranjang dan login. Kemudian pengguna dapat melakukan login lalu dapat mengakses menu profil, keranjang belanja, riwayat belanja dan pembayaran. Use case diagram pada pengguna dapat dilihat pada Gambar 5.



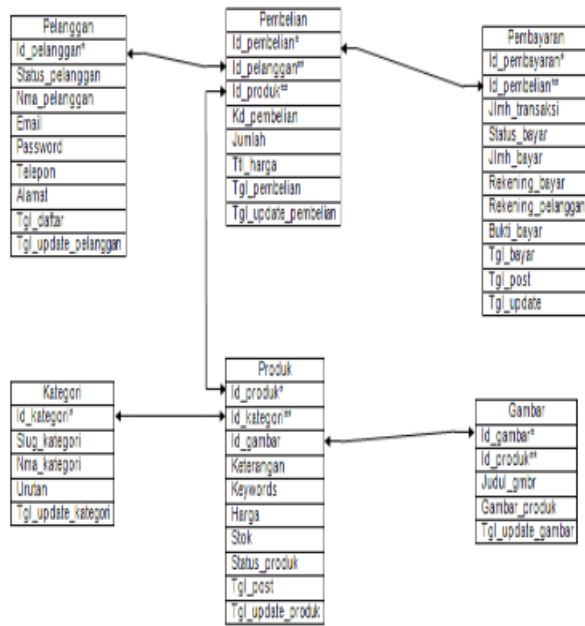
Gambar 5: Use Case Diagram Pengguna

Sequence Diagram Admin

Sequence Diagram Admin menjelaskan bagaimana interaksi-interaksi yang dapat dilakukan oleh admin setelah admin login pada sistem[9]. Sequence Diagram Admin dapat dilihat pada Gambar 6.



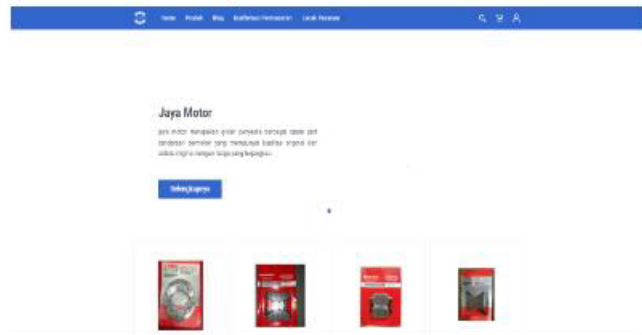
Gambar 6: Sequence Diagram Admin



Gambar 8: Class Diagram

User Interface

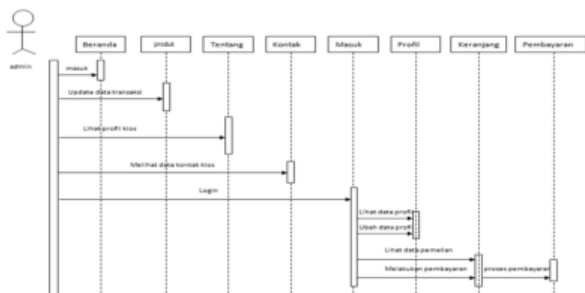
Pada halaman ini merupakan halaman utama dimana setiap pengguna akan secara otomatis memasuki halaman ini disaat pertama kali menggunakan sistem ini. Gambar 9 adalah Tampilan halaman utama.



Gambar 9: Tampilan Halaman Utama

Sequence Diagram Pengguna

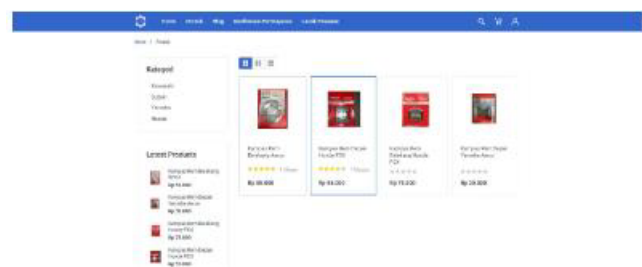
Sequence Diagram Pengguna menjelaskan bagaimana interaksi-interaksi yang dapat dilakukan oleh pengguna setelah login pada sistem. Sequence Diagram Pengguna dapat dilihat pada Gambar 7.



Gambar 7: Sequence Diagram Pengguna

Class Diagram

Class diagram adalah sebuah class yang menggambarkan struktur dan penjelasan class, paket, dan objek serta hubungan satu sama lain seperti containment, pewarisan, asosiasi, dan lain-lain[10]. Class diagram juga menjelaskan hubungan antar class dalam sebuah sistem yang sedang dibuat dan bagaimana caranya agar mereka saling berkolaborasi untuk mencapai sebuah tujuan. Class diagram dapat dilihat pada Gambar 8.

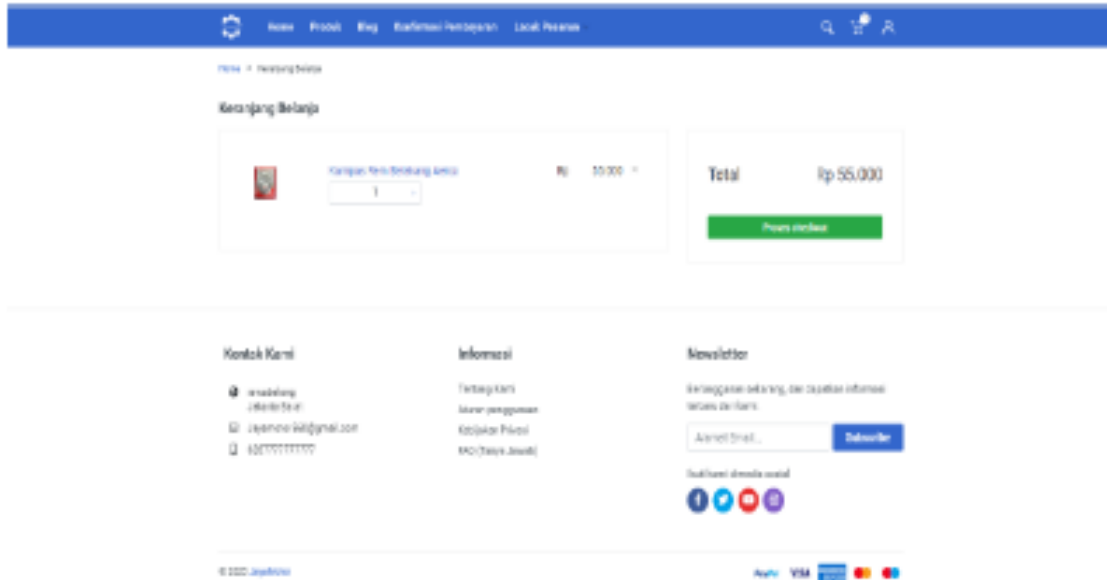


Gambar 10: Tampilan Halaman Kategori

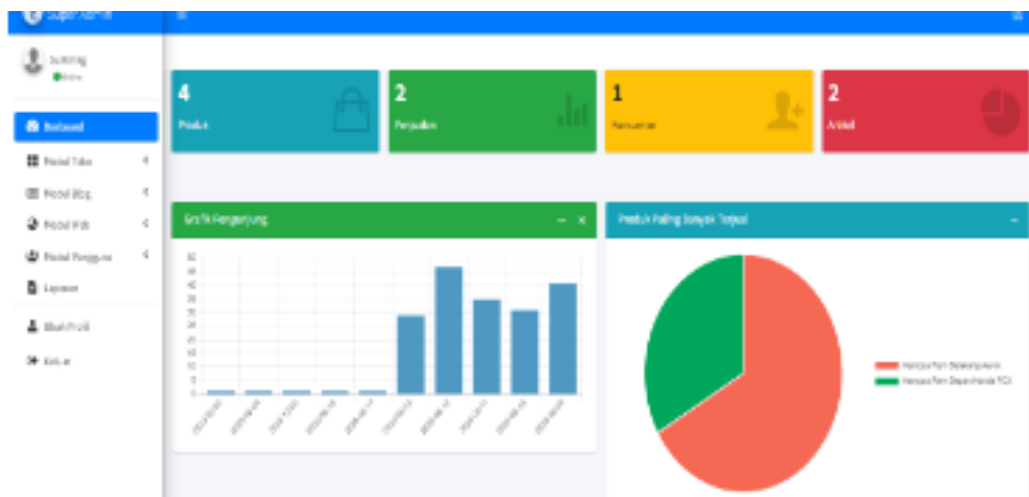
Halaman keranjang akan menampilkan list Produk yang akan mau di beli oleh user dan bisa di hapus kembali maupun lanjut belanja kembali. Gambar 11 adalah Tampilan halaman keranjang.

Pada halaman utama panel admin khusus untuk login admin untuk masuk panel admin. Gambar 12

adalah Tampilan utama panel admin. Pada halaman daftar penjualan, admin dapat menginput resi barang yang sudah dikirim maupun melihat detail barang yang belum di checkout ataupun sudah di checkout.



Gambar 11: Tampilan Halaman Kategori



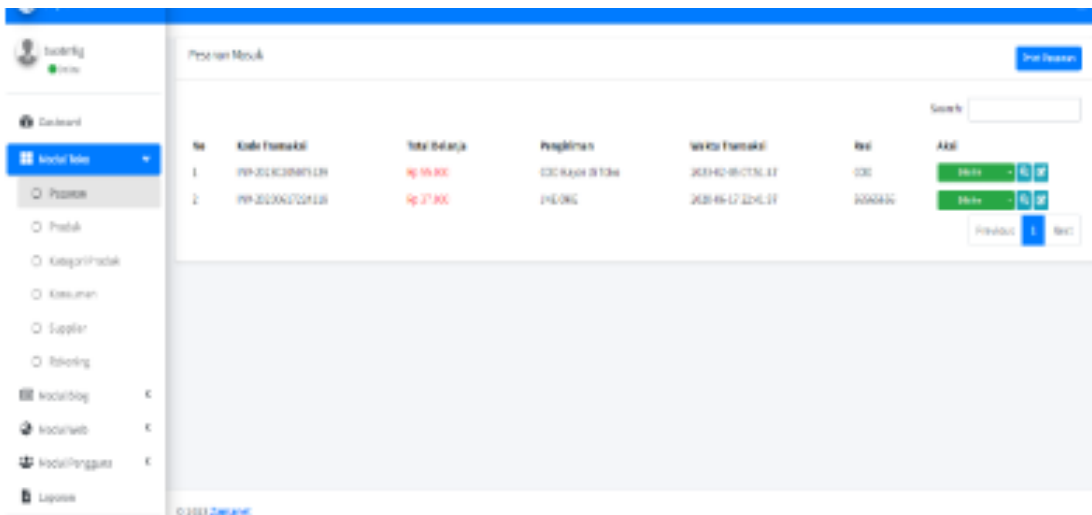
Gambar 12: Tampilan utama panel admin

Gambar 13 adalah tampilan halaman daftar penjualan. Pada halaman daftar produk, admin dapat menambah produk baru, mencari, mengubah dan menghapus produk. Gambar 14 adalah Tampilan halaman daftar produk.

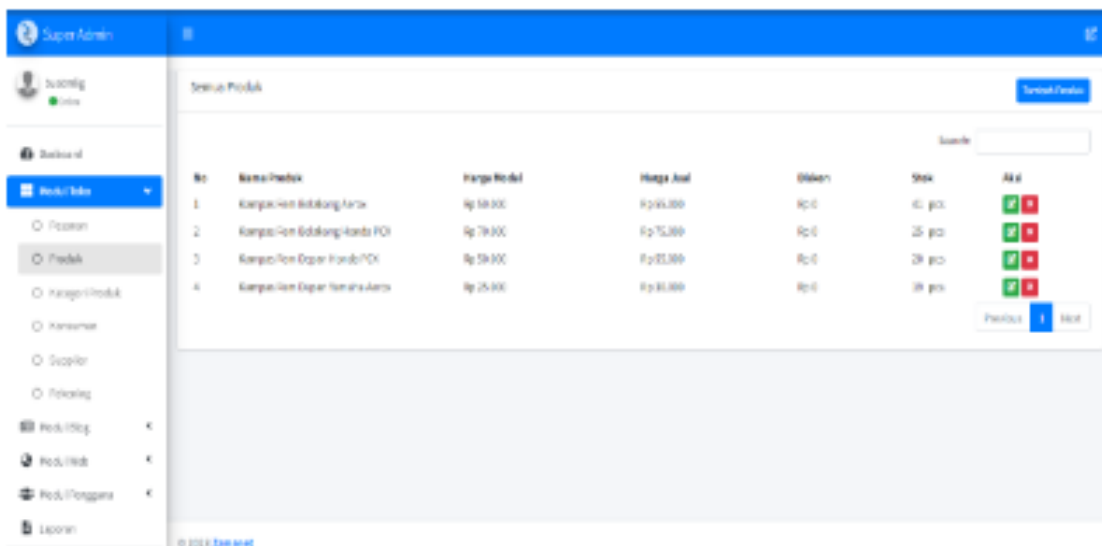
Uji Penilaian Pengguna

Untuk mengetahui penilaian pengguna, maka diberikan kuesioner kepada pengguna website dan kuesioner diberikan kepada 20 orang responden

yang menggunakan website ini. Responden dipersilahkan untuk menguji jalannya website untuk nantinya memberikan penilaian melalui kuesioner yang diberikan dan dari hasil kuesioner tersebut akan dilakukan perhitungan agar dapat diambil kesimpulan terhadap penilaian sistem ini[11]. Untuk mengetahui penilaian pengguna, maka diberikan kuesioner kepada pengguna website dan kuesioner diberikan kepada 20 orang responden yang menggunakan website ini.



Gambar 13: Tampilan halaman daftar penjualan



Gambar 14: Tampilan halaman daftar produk

Responden dipersilahkan untuk menguji jalannya website untuk nantinya memberikan penilaian melalui kuesioner yang diberikan dan dari hasil kuesioner tersebut akan dilakukan perhitungan agar dapat diambil kesimpulan terhadap penilaian sistem ini. Hasil kuesioner akan dihitung menggunakan metode pengukuran Skala Likert dengan rentang jarak (interval) sebanyak 20 yang didapat dari 100 / jumlah skor Likert (5). Berikut kriteria interpretasi skor berdasarkan interval yang sudah dibuat:

- a. Angka 0% – 20% = Sangat tidak setuju
- b. Angka 21% – 40% = Tidak setuju
- c. Angka 41% – 60% = Kurang setuju
- d. Angka 61% – 80% = Setuju
- e. Angka 81% – 100% = Sangat setuju

Perhitungan dilakukan dengan cara sebagai berikut:

Diketahui :

SS = Pilihan jawaban sangat setuju bernilai 5.

S = Pilihan jawaban setuju bernilai

4. KS = Pilihan jawaban kurang bernilai setuju

3. TS = Pilihan jawaban tidak bernilai setuju

2. STS = Pilihan jawaban sangat bernilai tidak

setuju

1. X = Jumlah responden.

Y = Total Skor.

Mencari nilai Y (total skor) = Jumlah pilihan jawaban*nilai. Mencari hasil akhir = $Y / (X*SS)$

Mencari terlebih dahulu total score yaitu jawaban dari responden dikali skor kemudian dijumlahkan maka akan dapat total skor. Untuk mencari hasil akhir perhitungan dengan cara jumlah responden dikalikan skor tertinggi kemudian total skor dibagi dengan hasil tersebut. Perhitungan dilakukan dalam sebuah tabel seperti Tabel 1.

Tabel 1: Hasil Kuesioner Terhadap Penilaian Pengguna

NO	PERNYATAAN	SS (5)	S (4)	KS (3)	TS (2)	STS (1)	RESPONDEN (X)	TOTAL SKOR (Y)	Y/(X *SS)	HASIL AKHIR
1	Mudah dalam mengakses website	7	13				20	87	0,870	87%
2	Pilihan menu sesuai yang dibutuhkan	8	9	3			20	85	0,850	85%
3	Menu dari keseluruhan halaman mudah dimengerti	7	10	3			20	84	0,840	84%
4	Informasi yang diberikan mudah dimengerti	4	13	2			20	82	0,820	82%
5	Gambar dan teks dapat dilihat dengan jelas	8	9	3			20	85	0,850	85%
6	Tampilan dari keseluruhan halaman menarik	7	13				20	87	0,870	87%
7	Website sesuai dengan fungsi yang dibutuhkan	6	12	2			20	84	0,840	84%
8	Mudah dalam mengoperasikan website	8	10	2			20	86	0,860	86%
9	Website membuat pembelian barang menjadi efektif	8	9	3			20	85	0,850	85%
10	Proses daftar menjadi member, lebih mudah dilakukan dan dimengerti	9	8	3			20	86	0,860	86%
11	Menu halaman cara order membantu pengguna untuk mengerti cara order	5	10	5			20	80	0,800	80%
12	Pengguna merasa terbantu dalam mencari produk berdasarkan kategori	8	9	3			20	85	0,850	85%
13	Pengguna dapat mengetahui detail produk dengan jelas	8	12				20	88	0,880	88%
14	Pengguna merasa diuntungkan dengan adanya detail tentang produk yang dicari	9	8	3			20	86	0,860	86%
15	Pengguna mendapat kemudahan dalam membeli produk	8	10	2			20	86	0,860	86%
16	Pengguna dapat mengetahui informasi mengenai daftar order	8	12				20	88	0,880	88%
17	Pengguna dapat mengubah biodata dan mengubah password	7	13				20	87	0,870	87%
18	Pengguna dapat mengecek no resi pengiriman	10	9	1			20	89	0,890	89%
19	Website sangat dibutuhkan dan sangat membantu	11	8	1			20	90	0,900	90%
20	Perlu dibuatkan aplikasi mobile android dan IOS	7	10	1	1	1	20	81	0,810	81%

Penutup

Hasil perhitungan kuesioner disimpulkan bahwa sebanyak 87% sangat setuju bahwa website dapat digunakan dengan baik dan mudah dalam mengakses website, selain itu responden juga sangat setuju bahwa perlu dibuat sistem mobile Android dan IOS. Dengan terbentuknya sistem informasi penjualan berbasis web maka dapat menampilkan secara online tentang layanan dan produk. Setiap fungsi fitur yang terdapat pada sistem dapat berfungsi benar

dan sesuai dengan hasil rancangan. Data transaksi pembayaran dan penjualan barang yang diterima tersimpan didalam sebuah database. Pencarian terhadap data akan lebih mudah, dan pengolahan data transaksi lebih optimal dan terstruktur. Pengembangan sistem yang dapat dilakukan adalah dengan menambahkan menu yang sesuai dengan kebutuhan pelanggan agar memudahkan pelanggan dalam mendapatkan informasi. Dan menjadikan website ini menjadi flexibel dapat dibuka di mobile.

Ucapan Terimakasih

Terima kasih kepada para pimpinan STMIK Jakarta STI&K yang telah memberikan dukungan terhadap pelaksanaan penelitian. Dan para rekan dosen yang telah membantu dalam proses pelaksanaan penelitian, kami mengucapkan terimakasih

Daftar Pustaka

- [1] Suendri, "Penerapan Konsep Model View Controller Pada Perancangan Sistem Manajemen", vol. 3, no. 2, pp. 36–45, 2018.
- [2] K. Wijaya dan A. Christian, "Implementasi Metode Model View Controller (MVC) Dalam Rancang Bangun Website SMK Yayasan Bakti Prabumulih", *Paradig. - J. Komput. dan Inform.*, vol. XXI, no. 1, pp.1–8, doi: 10.31294/p.v20i2, 2019.
- [3] Syaiful Anwar, "Rancangan Aplikasi Penjualan Perlengkapan Dan Mainan Anak Menggunakan Konsep Model View Controller (MVC)", *JURSISTEKNI (Jurnal Sistem Informasi dan Teknologi Informasi)* Vol. 2, No. 1, Januari 2020: Hal. 33 – 45 ISSN. P: 2715-1875, E: 2715-1883, 2020.
- [4] Saad, M.I., "Otodidak Web Programming: Membuat Website Edutainment:", Elex Media Komputindo, 2020.
- [5] Eddy Suprihadi , *Sistem Informasi Bisnis Dunia*, AndiPublisher, Yogyakarta, 2020.
- [6] Mochahmad Wahyudi, *Fullstack Developer Aplikasi Penjualan Tiket Bioskop*, Graha Ilmu, Yogyakarta, 2021.
- [7] W.N. Cholifah, Y. Yulianingsih, dan S.M. Sagita, "Pengujian black box testing pada aplikasi action & strategy berbasis android dengan teknologi phonegap", *STRING (Satuan Tulisan Riset dan Inovasi Teknologi)*, 3(2), pp.206-210, DOI: <http://dx.doi.org/10.30998/string.v3i2.3048>, 2018.
- [8] Rohi Abdullah, *Membuat Toko Online dengan Teknik OOP, MVC, dan AJAX*, PT. Elex Media Komputindo, Jakarta, 2017.
- [9] K. Nistrina, dan L. Sahidah, *Unified Modelling Language (Uml) Untuk Perancangan Sistem Informasi Penerimaan Siswa Baru Di Smk Marga Insan Kamil*, *J-SIKA| Jurnal Sistem Informasi Karya Anak Bangsa*, 4(1), pp.17-23, 2022.
- [10] Anto Subagia, "Membangun Aplikasi dengan Codeigniter dan Database SQL Server", PT. Elex Media Komputindo, Jakarta, 2017.
- [11] Dheni Apriantsani Budiman dan Decky Maulana Nugraha, "Aplikasi Raport Online Berbasis Web Menggunakan Framework CodeIgniter", *Jurnal Computech & Bisnis STMIK Mardira Indonesia Bandung*, Vol. 13 No. 2, 2019.