

RANCANG BANGUN SISTEM INFORMASI PENJUALAN *HYBRID* BERBASIS WEBSITE (STUDI KASUS TOKO RIZKI PLASTIK)

Rizki Susilo Ramadhan, Apriade Voutama, Hannie

Program Studi Sistem Informasi S1, Fakultas Ilmu Komputer
Universitas Singaperbangsa Karawang, Jl. HS. Ronggo Waluyo Karawang, Indonesia
1910631250031@student.unsika.ac.id

ABSTRAK

Penelitian ini bertujuan untuk membuat sistem penjualan berbasis *website* yang dapat melayani penjualan *offline* dan *online (hybrid)* di Toko Rizki Plastik. Pengembangan sistem menggunakan metode SDLC model *waterfall*. Sistem yang dibuat dapat memproses penjualan *offline* dengan menggunakan modul kasir sedangkan untuk memproses penjualan *online* menggunakan modul toko *online*. Kedua modul tersebut dikelola oleh satu sistem yang dinamakan sistem informasi penjualan *hybrid* dengan tujuan agar data penjualannya terorganisir, terintegrasi, dan terkomputerisasi dengan baik. Dalam pengembangannya sistem akan dimodelkan dengan diagram UML dan diimplementasikan menggunakan bahasa pemrograman *Node.js* dan *PostgreSQL* sebagai *database*. Hasil dari penelitian ini menghasilkan 2 aktor *use case*, yaitu pelanggan dan *admin*. Pelanggan dapat melakukan belanja *online* melalui toko *online* dan *admin* dapat mengelola penjualan melalui *dashboard* penjualan pada sistem penjualan *hybrid*. Pengujian sistem informasi berhasil dilakukan menggunakan metode *blackbox testing* dengan hasil uji fungsionalitas yang sudah sesuai dengan harapan.

Kata kunci: Sistem Informasi, Penjualan Hybrid, *Node.js*, Website

1. PENDAHULUAN

Perkembangan teknologi telah mengakselerasi kemajuan sistem informasi dan transformasi digital yang terjadi di seluruh dunia. Dalam beberapa dekade terakhir, sistem informasi telah memainkan peran kunci dalam meningkatkan efisiensi, efektivitas, dan daya saing organisasi di berbagai sektor, seperti bisnis, pemerintahan, pendidikan, dan kesehatan.

Akselerasi digital yang telah terjadi memberikan dampak yang signifikan terhadap perilaku belanja konsumen di Indonesia. Seiring dengan kemajuan teknologi dan penyebaran akses internet yang lebih luas, kini konsumen memiliki lebih banyak pilihan untuk memilih cara mereka berbelanja, seperti melalui *e-commerce* dengan mengandalkan koneksi internet, langsung datang ke toko fisik dan mengkombinasikan keduanya seperti yang diterapkan pada metode *buy online pickup in store*. Tentunya setiap metode berbelanja yang ada memiliki kelebihan dan kekurangan yang berbeda-beda. Perkembangan perilaku belanja tersebut datang bukan untuk menggantikan salah satu metode yang ada, melainkan merupakan kesempatan atau peluang untuk pelaku usaha untuk memberikan pelayanan yang terbaik kepada pelanggan dan memudahkan mereka dalam berbelanja dengan cara mengintegrasikan metode penjualan *offline* dan *online* [1].

Dengan demikian penelitian ini memiliki tujuan yaitu mengembangkan sistem informasi penjualan *hybrid* berbasis *website*, sehingga pelaku usaha dapat menjual produknya kepada konsumen dengan beberapa metode penjualan seperti penjualan *online*, *offline*, dan menggabungkan keduanya. Integrasi dan sinkronisasi data penjualan merupakan hal yang penting dalam membangun sistem ini, oleh karena itu

didalam sistem ini akan menggunakan konsep *point of sale* / sistem kasir untuk memproses transaksi penjualan *offline* dan sistem *e-commerce* untuk transaksi penjualan *online*. Kedua sistem tersebut akan didistribusikan ke dalam satu sistem *dashboard* penjualan yang merupakan sistem informasi penjualan *hybrid*. Dengan begitu akan membantu meminimalisir *error* dalam memproses transaksi dari berbagai metode penjualan. Penelitian ini melakukan observasi yang dilakukan secara langsung pada Toko Rizki Plastik yang sedang menjalani bisnis model dengan model penjualan *hybrid*. Toko Rizki Plastik merupakan toko retail yang menjual berbagai macam kebutuhan mulai dari kemasan plastik, alat dan bahan baku untuk pembuatan kue dan roti. Saat ini sumber penjualan utama berasal dari transaksi penjualan *offline* dan *online*. Untuk transaksi *offline* di toko tersebut, pelanggan harus datang langsung ke lokasi toko untuk melakukan pembelian, sedangkan untuk transaksi *online*, pelanggan melakukan pemesanan barang dan konfirmasi pembayaran melalui *Whatsapp* kemudian konsumen mengambil barang pesannya di lokasi atau mendelegasikan pengambilan pesannya kepada ojek online atau juga bisa pesanan konsumen diantar ke rumah (*home service delivery*) dengan jarak tertentu.

2. TINJAUAN PUSTAKA

2.1. Rancang Bangun

Rancang bangun merujuk pada proses pengembangan suatu sistem, produk, atau teknologi baru. Rancang bangun melibatkan langkah-langkah yang terencana dan sistematis untuk merancang, mengembangkan, dan membangun solusi yang dari masalah dan tujuan yang sudah ditetapkan. Tujuan

utamanya adalah untuk memenuhi kebutuhan pengguna dengan cara yang efektif dan efisien. Dalam proses perancangan, fokus diberikan pada pemahaman yang mendalam tentang kebutuhan pengguna agar solusi yang dihasilkan dapat memenuhi kebutuhan tersebut secara optimal. Selain itu, tujuan lainnya adalah untuk meningkatkan kualitas perangkat lunak. Dengan merancang struktur yang baik, mengelola ketergantungan antar modul, dan mempertimbangkan aspek-aspek seperti keamanan dan kinerja, perangkat lunak dapat menjadi lebih stabil, mudah dipelihara, dan lebih dapat diandalkan [1].

2.2. Sistem Informasi

Sistem informasi merupakan sekumpulan sub-sistem yang saling terintegrasi atau terhubung dan bekerja sama, dimulai dari proses pengumpulan, penyimpanan, dan analisis sebuah data hingga menghasilkan sebuah informasi yang bermanfaat bagi penggunanya. Tujuan utama dari sistem informasi adalah menyediakan informasi yang relevan, akurat, dan tepat waktu kepada pengguna di dalam dan di luar organisasi, sehingga mendukung pengambilan keputusan yang baik, pengelolaan operasional yang efisien, dan pencapaian tujuan bisnis yang ditetapkan [2].

2.3. Penjualan

Penjualan merupakan suatu proses yang bertujuan untuk mengalihkan produk atau jasa kepada pelanggan dengan maksud memperoleh keuntungan atau profit dari transaksi penjualan. Kegiatan-kegiatan dalam penjualan meliputi kegiatan yang dimulai dari mencari pembeli, mempengaruhi pembeli hingga melakukan syarat-syarat pembayaran. Tujuan penjualan adalah untuk menghasilkan pendapatan dan meningkatkan pangsa pasar. Proses penjualan melibatkan langkah-langkah seperti identifikasi prospek, penawaran produk atau jasa kepada prospek, negosiasi harga atau syarat-syarat, penutupan penjualan, dan pengiriman produk atau penyediaan jasa kepada pelanggan [3].

2.4. Penjualan Hybrid

Penjualan *hybrid* adalah perpaduan antara sistem penjualan *online* dan *offline* yang menyebabkan makna penjualan *online* dan *offline* tidak bermakna untuk saling menggantikan tetapi memberikan *fleksibilitas* kepada konsumen dengan berbagai manfaat atau keuntungan yang diperoleh dari masing-masing sistem penjualan baik *offline* ataupun *online* [4].

Penjualan *offline* terjadi di dunia nyata dan mempertemukan kedua pihak yaitu penjual dan pembeli di suatu tempat seperti toko sehingga terjadi proses transaksi jual beli.

Penjualan *online* merupakan proses transaksi yang terjadi via *online* dan menggunakan platform seperti *website* yang terhubung dengan internet. Penjualan *online* menawarkan kemudahan kepada

pembeli dalam proses pencarian produk hingga pembayaran saat melakukan transaksi.

2.5. Website

Website pertama kali ditemukan oleh *computer scientist* yang berasal dari Inggris bernama Tim Berners-Lee pada saat ia bekerja sebagai *Software Engineer* di CERN. Awal dari tujuan pengembangan *website* oleh Tim Berners-Lee adalah untuk memudahkan para peneliti di tempatnya bekerja untuk bertukar informasi antar para peneliti yang tersebar di berbagai negara.

Website merupakan kumpulan halaman yang berisi sajian informasi yang dapat diakses oleh siapapun, kapanpun, dan dimanapun selama memiliki akses dengan internet dan memiliki perangkat lunak *web browser*. Dengan menggunakan konsep *hyperlink* (tautan), pengguna *website* dapat dengan mudah mendapatkan informasi berupa dokumen, gambar, audio, dan video [5].

2.6. Unified Modelling Language (UML)

UML merupakan metode pemodelan sistem yang digunakan dalam perancangan sistem untuk mengurangi kesalahan yang mungkin terjadi saat membuat program. Penerapan UML dapat menggambarkan aktor. Diagram UML mencakup berbagai jenis, seperti diagram *use case* yang menggambarkan interaksi antara aktor dan sistem, diagram kelas yang menunjukkan struktur kelas dan hubungan di antara mereka, diagram aktivitas yang menggambarkan alur kerja proses, dan masih banyak lagi. UML diagram membantu memperjelas kompleksitas sistem dan memungkinkan tim pengembang untuk memiliki pemahaman yang lebih baik tentang perangkat lunak yang akan mereka bangun [6].

2.7. Software Development Life Cycle (SDLC)

SDLC adalah proses untuk memahami bagaimana sebuah sistem informasi dapat membantu menyelesaikan masalah yang ada didalam proses bisnis, merancang dan membangun sistem, serta mengirimkan sistem kepada pengguna untuk digunakan. SDLC meliputi tahap perencanaan, analisis, desain, implementasi, pengujian, dan pemeliharaan. Tahap perencanaan melibatkan identifikasi kebutuhan bisnis, sedangkan tahap analisis menganalisis kebutuhan tersebut. Tahap desain melibatkan perancangan solusi perangkat lunak, sedangkan tahap implementasi berfokus pada penulisan kode. Setelah itu, perangkat lunak diuji secara menyeluruh sebelum diluncurkan. Tahap pemeliharaan terjadi setelah peluncuran dan melibatkan pemantauan, pemecahan masalah, dan pembaruan perangkat lunak. SDLC memberikan kerangka kerja yang efektif untuk mengelola proses pengembangan perangkat lunak dengan tujuan memastikan kualitas dan keberhasilan proyek [7].

2.8. Metode Waterfall

Metode *waterfall* adalah model di dalam Software Development Life Cycle (SDLC) yang mengikuti urutan linear dari tahap analisis, perancangan, pengembangan, pengujian, dan implementasi. Di dalam model *waterfall*, setiap tahapan harus diselesaikan secara keseluruhan sebelum melanjutkan ke tahapan berikutnya. Dalam hal ini, tahap-tahap tersebut saling terkait dan membentuk aliran seperti air terjun, di mana setiap tahap merupakan input bagi tahap berikutnya. Pendekatan ini menekankan perencanaan yang matang dan dokumentasi komprehensif sebelum melanjutkan ke langkah selanjutnya. Namun, kelemahan metode *waterfall* terletak pada kurangnya fleksibilitas dalam mengakomodasi perubahan kebutuhan dan umpan balik pengguna selama proses pengembangan [8].

2.9. Node.js

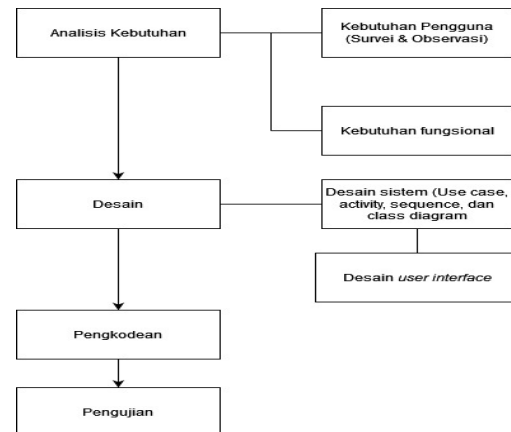
Node.js adalah platform pengembangan perangkat lunak berbasis JavaScript yang berjalan di sisi server. Dibangun di atas mesin JavaScript V8 dari Google Chrome, Node.js memungkinkan pengembang untuk menjalankan kode JavaScript di luar lingkungan browser. Dengan menggunakan Node.js, pengembang dapat membuat aplikasi jaringan yang efisien dan skalabel dengan kemampuan untuk menangani banyak koneksi bersamaan. Node.js juga memiliki ekosistem yang kaya dengan berbagai modul dan alat bantu yang mempercepat proses pengembangan. Dengan kecepatan dan skalabilitasnya, Node.js telah menjadi pilihan populer bagi para pengembang untuk membangun aplikasi server-side, aplikasi web real-time, dan aplikasi jaringan berkinerja tinggi. Node.js membuat *javascript* dapat dijalankan di sisi server seperti PHP, Ruby, Java, dan Go. Node.js tidak memerlukan perangkat lunak *web server* dikarenakan *Node.js* sudah memiliki *web server HTTP* sendiri untuk menjalankan program server untuk *web* [9].

2.10. Blackbox Testing

Blackbox testing dalam perangkat lunak adalah metode pengujian yang fokus pada perilaku dan fungsi eksternal perangkat lunak, tanpa memperhatikan struktur internal atau logika implementasinya. Dalam blackbox testing, pengujian dilakukan dari perspektif pengguna eksternal, di mana input diberikan ke perangkat lunak dan output yang dihasilkan dievaluasi untuk memastikan bahwa perangkat lunak berperilaku sesuai dengan harapan. Para pengujian tidak perlu mengetahui rincian implementasi perangkat lunak dan dapat menguji berbagai kasus penggunaan, input yang tidak valid, dan situasi yang berbeda untuk mengidentifikasi bug, kesalahan logika, atau masalah kinerja. Blackbox testing membantu memastikan bahwa perangkat lunak berfungsi dengan baik dari perspektif pengguna, memberikan kualitas yang diharapkan, dan memenuhi kebutuhan fungsionalitas yang telah ditetapkan [10].

3. METODE PENELITIAN

Penelitian ini menggunakan pendekatan studi kasus dan metode SDLC atau System Development Life Cycle dengan model *Waterfall* dengan urutan proses seperti dibawah ini :



Gambar 1. Alur penelitian

3.1. Analisis Kebutuhan

Pada tahap ini analisis kebutuhan mencakup juga pada analisis pengguna dan juga analisis kebutuhan fungsional yang menghasilkan penemuan-penemuan yang mana dibutuhkan sebagai pembangun fitur-fitur dalam sistem informasi yang akan dikembangkan.

3.2. Desain

Pada tahap ini yaitu setelah memperoleh hasil dari analisis kebutuhan adalah memodelkan data hasil analisis kebutuhan menjadi diagram UML yang dapat membantu untuk mengembangkan sistem.

3.3. Pengkodean

Tahapan ini merupakan tahap mengimplementasikan hasil pada tahap sebelumnya, yaitu tahap desain sistem ke dalam pengkodean menggunakan bahasa pemrograman *Node.js*.

3.4. Pengujian

Pada tahap ini adalah menguji hasil pekerjaan dari tahap pengkodean. Teknik pengujiannya menggunakan *blackbox testing*.

4. HASIL DAN PEMBAHASAN

4.1. Analisis Kebutuhan

Analisis dilakukan dengan cara mewawancarai owner atau pemilik dari Toko Rizki Plastik kemudian dari hasil wawancara tersebut mendapatkan informasi yang valid untuk menentukan kebutuhan pengguna sistem penjualan *online* berbasis *website*. Hasil analisis dari metode pengumpulan data yang sudah dilakukan adalah sebagai berikut:

1. Analisis Kebutuhan Pengguna

Pada sistem ini terdapat 2 pengguna, pelanggan yang melakukan transaksi secara *online* atau *offline* dan *admin* yang sebagai pengelola *dashboard* penjualan.

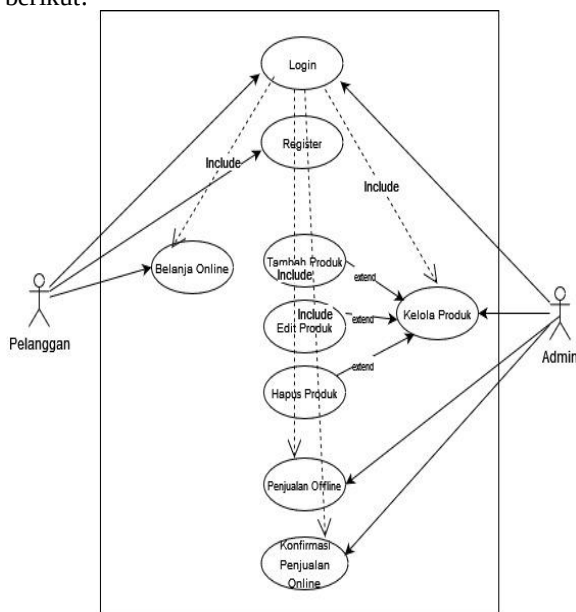
2. Analisis Fungsional

Analisis fungsional bertujuan untuk mengidentifikasi fitur-fitur yang wajib ada pada sistem dan memastikan agar sistem tersebut harus beroperasi untuk memenuhi kebutuhan pengguna. Berikut adalah beberapa kebutuhan fungsional yang diimplementasikan pada sistem ini:

- Sistem menggunakan konsep *e-commerce* atau toko *online* sebagai tempat untuk *user* melakukan transaksi secara *online*.
- Sistem menggunakan konsep kasir sebagai tempat untuk *user* melakukan transaksi secara *offline* yang akan diproses oleh *admin*.
- Pelanggan wajib melakukan *register* dan *login* pada sistem untuk belanja *online*.
- Admin dapat memproses transaksi didalam *dashboard* penjualan.
- Admin dapat mengatur produk-produk yang dijual pada *dashboard* penjualan.
- Pelanggan dapat membuat transaksi secara *online* melalui sistem penjualan.
- Admin dapat memproses transaksi pelanggan yang datang langsung ke toko fisik (*offline*).

4.2. Use Case Diagram

Use case diagram membantu dalam memahami kebutuhan pengguna, merencanakan dan mengelola proyek pengembangan perangkat lunak, serta berkomunikasi dengan pemangku kepentingan yang terlibat. Dibawah ini merupakan *use case diagram* yang sudah selesai dibuat untuk Sistem Informasi Penjualan *Hybrid* Berbasis *Website*, yaitu sebagai berikut:



Gambar 2. Usecase diagram

Hasil *use case diagram* pada gambar diatas merupakan hasil dari analisis kebutuhan pengguna

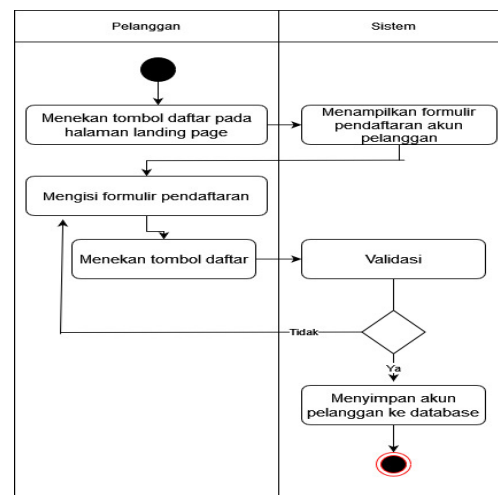
yang dilakukan pada tahap *analisis* sehingga menghasilkan *use case* beserta aktor yang terlibat pada Sistem Penjualan *Hybrid* Berbasis *Website*.

4.3. Activity Diagram

Activity diagram berperan dalam memvisualisasikan, menganalisis, memodelkan, mengkoordinasikan, berkomunikasi, dan memahami aliran informasi dalam suatu proses bisnis. Dibawah ini merupakan *activity diagram* yang telah dirancang untuk Sistem Informasi Penjualan *Hybrid* Berbasis *Website*, yaitu sebagai berikut:

4.4. Activity Diagram Register

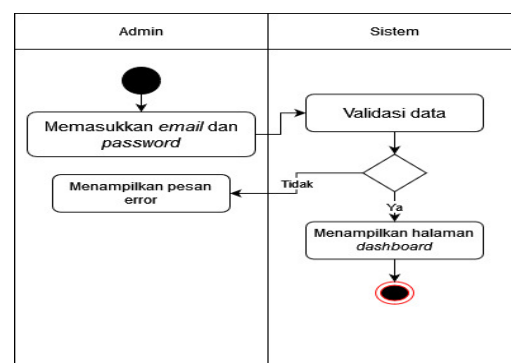
Pada gambar dibawah ini merupakan *activity diagram register* untuk aktor pelanggan sedangkan untuk aktor admin tidak ada karna akun untuk aktor *admin* secara *default* sudah dibuatkan oleh sistem. Data yang dibutuhkan untuk melakukan *register* adalah email dan *password*



Gambar 3. Activity diagram register

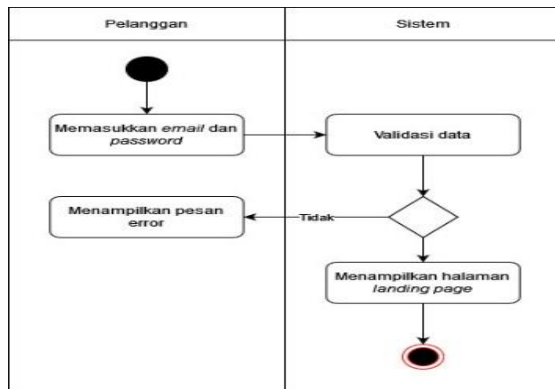
4.5. Activity Diagram Login

Pada gambar dibawah ini merupakan *activity diagram login* untuk aktor admin dan pelanggan. Untuk admin, jika berhasil login admin akan diarahkan ke halaman *dashboard* penjualan



Gambar 4. Activity diagram login admin

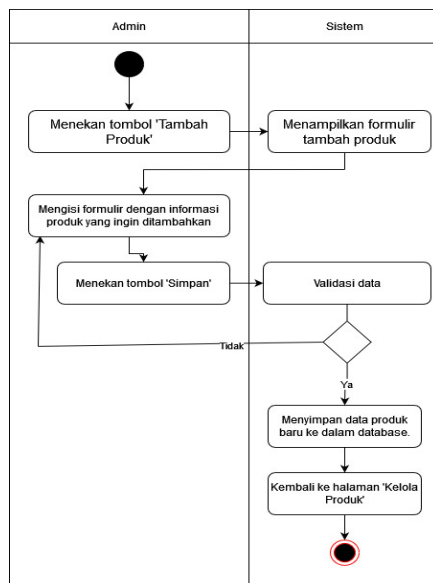
Untuk pelanggan, jika pelanggan berhasil login maka pelanggan akan diarahkan ke halaman *landing page* toko online setelah itu pelanggan bisa mulai untuk melakukan proses belanja online.



Gambar 5. Activity diagram login pelanggan

4.6. Activity Diagram Tambah Produk

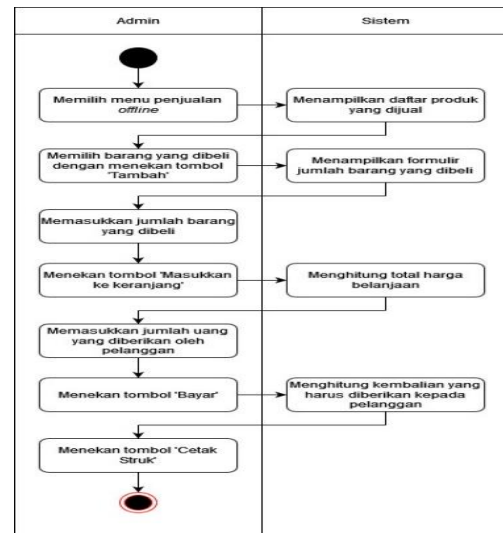
Pada gambar dibawah ini, admin dapat menambahkan atau mengakses fitur tambah produk yang berguna untuk menambahkan produk-produk yang dijual ke dalam sistem sehingga pelanggan dapat melakukan transaksi online ataupun offline. Produk yang ditambahkan memiliki informasi atau atribut seperti nama produk, stok, berat produk, foto produk, dan informasi lainnya yang bermanfaat untuk pelanggan mengetahui tentang informasi produk yang dijual.



Gambar 6. Activity diagram tambah produk

4.7. Activity Diagram Penjualan Offline

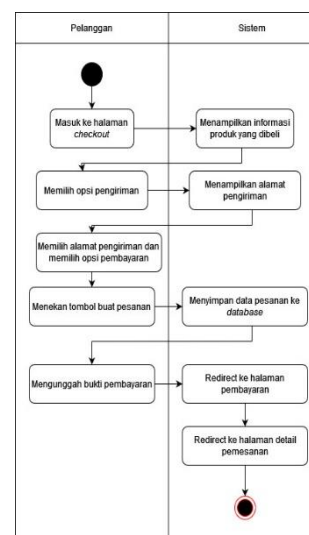
Pada gambar dibawah ini menggambarkan aktivitas yang terjadi ketika admin memproses transaksi penjualan yang terjadi secara offline atau saat pelanggan datang langsung ke toko fisik. Admin dapat menghitung total harga produk-produk yang dibeli oleh pelanggan pada *dashboard*.



Gambar 7. Activity diagram penjualan offline

4.8. Activity Diagram Belanja Online

Pada gambar dibawah ini merupakan *activity diagram* yang menggambarkan proses aktivitas pelanggan untuk melakukan belanja secara online melalui sistem penjualan. Transaksi yang masuk secara online kemudian akan dikonfirmasi oleh admin pada *dashboard*. Pelanggan dapat memilih opsi pengiriman ambil di toko atau *delivery order*. Untuk opsi pengiriman *delivery order*, pelanggan perlu memasukkan alamat pengiriman. Dan untuk opsi pembayaran tersedia *cash on delivery* atau transfer bank.

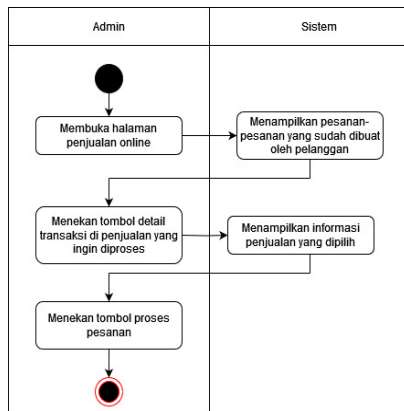


Gambar 8. Activity diagram belanja online

4.9. Activity Diagram Konfirmasi Penjualan Online

Pada gambar dibawah ini menggambarkan aktor admin yang dapat mengakses atau memproses transaksi pelanggan yang terjadi secara online. Setelah pelanggan melakukan pemesanan online melalui sistem. Aktivitas selanjutnya adalah admin melakukan konfirmasi pemesanan agar dapat segera diproses.

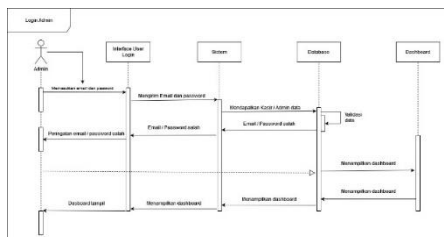
Sebelum melakukan konfirmasi admin dapat melihat informasi transaksi seperti metode pengiriman dan pembayaran, alamat pengiriman, resi pembayaran, serta total harga produk yang dibeli.



Gambar 9. Activity diagram konfirmasi penjualan online

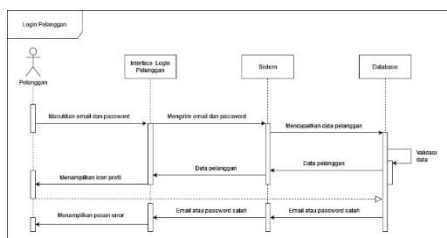
4.10. Sequence Diagram Login

Pada gambar dibawah ini merupakan *sequence diagram login* admin yang berfungsi untuk menggambarkan interaksi alur kerja yang dapat dilihat dari sisi admin, sistem, dan *database*.



Gambar 10. Sequence diagram login admin

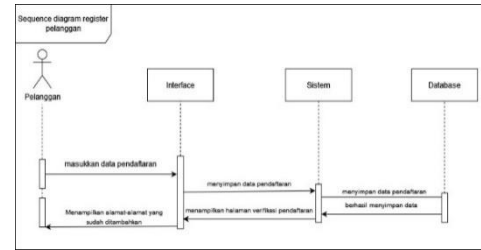
Sedangkan pada gambar dibawah ini merupakan *sequence diagram login* pelanggan yang berfungsi untuk menggambarkan interaksi alur kerja yang dapat dilihat dari sisi pelanggan, sistem, dan *database*.



Gambar 11. Sequence diagram login pelanggan

4.11. Sequence Diagram Register

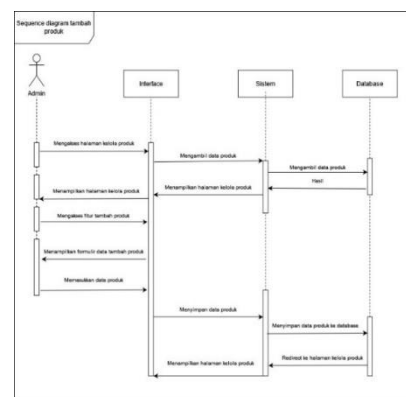
Pada gambar dibawah ini merupakan *sequence diagram register* yang berfungsi untuk menggambarkan interaksi alur kerja yang dapat dilihat dari sisi pelanggan, sistem, dan *database*.



Gambar 12. Sequence diagram register

4.12. Sequence Diagram Tambah Produk

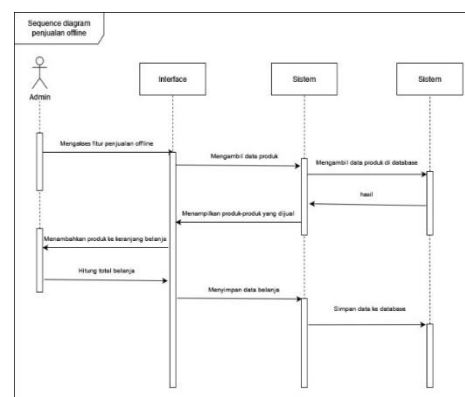
Pada gambar dibawah ini merupakan *sequence diagram tambah produk* yang berfungsi untuk menggambarkan interaksi alur kerja menambahkan produk yang dapat dilihat dari sisi admin, sistem, dan *database*.



Gambar 13. Sequence diagram tambah produk

4.13. Sequence Diagram Penjualan Offline

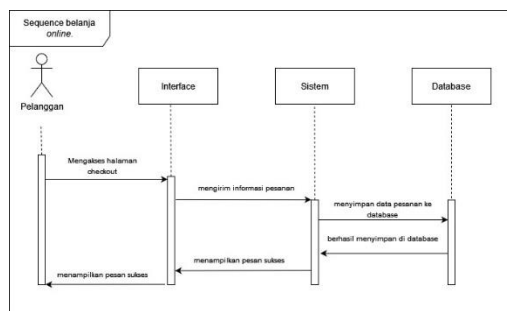
Pada gambar dibawah ini merupakan *sequence diagram penjualan offline* yang berfungsi untuk menggambarkan interaksi alur kerja dalam memproses transaksi penjualan *offline* yang dapat dilihat dari sisi admin, sistem, dan *database*.



Gambar 14. Sequence diagram penjualan offline

4.14. Sequence Diagram Belanja Online

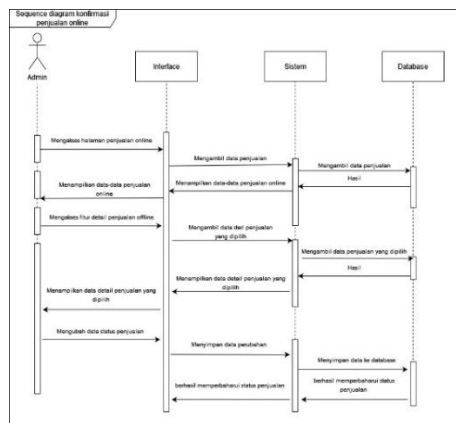
Pada gambar dibawah ini merupakan *sequence diagram belanja online* yang berfungsi untuk menggambarkan interaksi alur kerja dalam melakukan pemesanan atau belanja *online* yang dapat dilihat dari sisi admin, sistem, dan *database*.



Gambar 15. Sequence diagram belanja online

4.15. Sequence Diagram Konfirmasi Penjualan Online

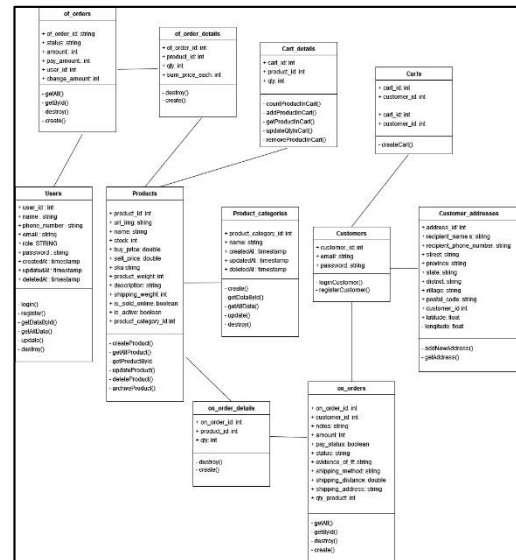
Pada gambar dibawah ini merupakan *sequence diagram* konfirmasi penjualan *online* yang berfungsi untuk menggambarkan interaksi alur kerja dalam memproses transaksi penjualan *online* yang dapat dilihat dari sisi admin, sistem, dan *database*.



Gambar 16. Sequence diagram konfirmasi penjualan online

4.16. Class Diagram

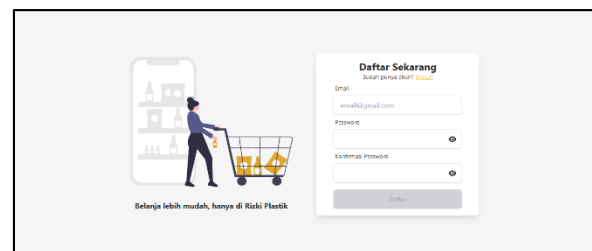
Class diagram berperan dalam representasi struktur sistem, pengorganisasian konsep objek, pengembangan kode yang efisien, komunikasi efektif, dan pemodelan relasi dan ketergantungan antara kelas-kelas dalam sistem. Dibawah ini merupakan gambar *class diagram* yang sudah selesai dibuat untuk sistem informasi penjualan *hybrid* berbasis *website*.



Gambar 17. Class diagram

4.17. Halaman Register

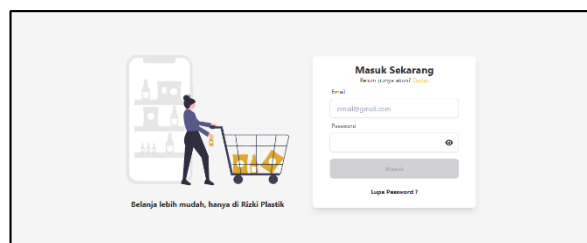
Gambar dibawah ini merupakan hasil tampilan dari halaman *register* yang berfungsi untuk mendaftarkan akun pelanggan kedalam sistem toko *online*. Untuk mendaftarkan akun, pelanggan perlu memasukkan *email* dan *password* yang valid.



Gambar 18. Halaman register pelanggan

4.18. Halaman Login Pelanggan

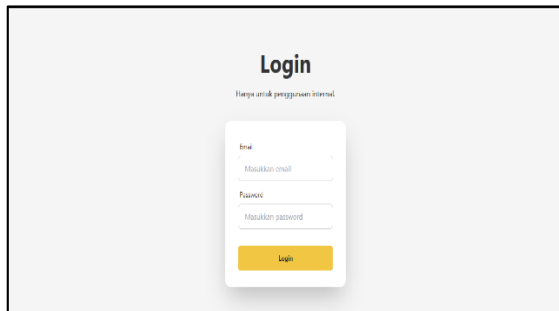
Gambar dibawah ini merupakan hasil tampilan dari halaman *login* yang berfungsi untuk akun pelanggan masuk ke dalam sistem toko *online*. Untuk masuk kedalam sistem, pelanggan perlu memasukkan *email* dan *password* yang sesuai.



Gambar 19. Halaman login pelanggan

4.19. Halaman Login Admin

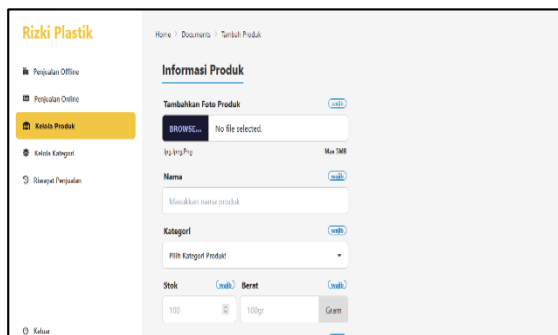
Gambar dibawah ini merupakan hasil tampilan dari halaman *login* admin yang berfungsi untuk akun admin masuk ke dalam sistem *dashboard* penjualan.



Gambar 20. Halaman *login* admin

4.20. Halaman Tambah Produk

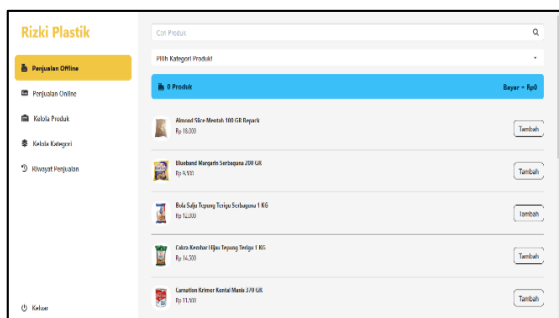
Gambar dibawah ini merupakan hasil tampilan dari halaman tambah produk, admin dapat menambahkan produk yang dijual dengan memasukkan data produk seperti nama, berat, harga produk dan sebagainya.



Gambar 21. Halaman tambah produk

4.21. Halaman Penjualan Offline

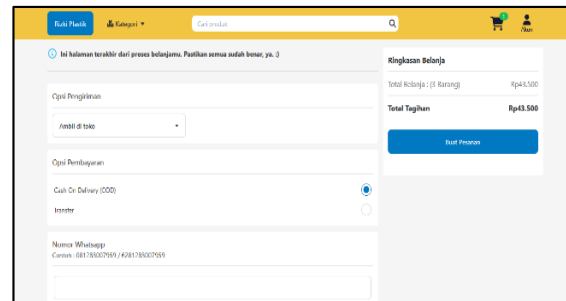
Gambar dibawah ini merupakan hasil tampilan dari halaman penjualan *offline* yang berfungsi untuk admin menghitung / memproses transaksi penjualan *offline* melalui sistem kasir yang ada pada *dashboard* penjualan.



Gambar 22. Halaman penjualan *offline*

4.22. Halaman Belanja Online

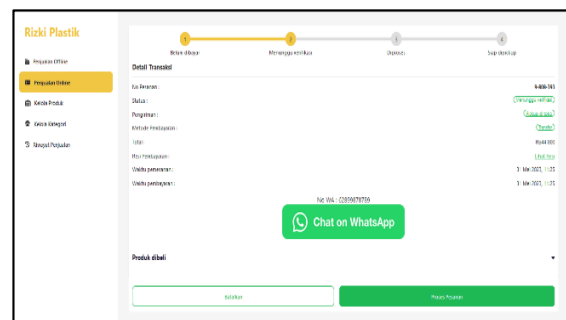
Gambar dibawah ini merupakan hasil tampilan dari halaman belanja *online* yang berfungsi untuk pelanggan melakukan pemesanan produk secara *online* melalui sistem toko *online*.



Gambar 23. Halaman belanja *online*

4.23. Halaman Konfirmasi Penjualan Online

Gambar dibawah ini merupakan hasil tampilan dari halaman konfirmasi penjualan *online* yang berfungsi untuk admin memproses pesanan yang masuk melalui toko *online*.



Gambar 24. Halaman konfirmasi penjualan *online*

4.24. Pengujian

Pada tahap ini, hasil yang diperoleh dari tahap pengkodean akan dilakukan pengujian fungsionalitasnya dengan metode *blackbox testing*. Blackbox testing melibatkan serangkaian langkah yang dilakukan untuk menguji perangkat lunak dari perspektif pengguna eksternal. Pertama, pengujian dimulai dengan memahami kebutuhan dan spesifikasi perangkat lunak yang akan diuji. Kemudian, skenario pengujian yang mencakup berbagai situasi, input valid dan tidak valid. Rencana pengujian dibuat untuk mengatur langkah-langkah pengujian dan hasil yang diharapkan. Data pengujian disiapkan sesuai dengan skenario yang ditentukan. Selanjutnya, kasus uji dijalankan satu per satu dengan memberikan input dan memeriksa keluaran yang dihasilkan. Hasil pengujian dibandingkan dengan hasil yang diharapkan untuk mengidentifikasi masalah atau kesalahan yang muncul. Temuan pengujian didokumentasikan dengan detail, dan langkah pengujian dapat diulang jika diperlukan.

Tabel 1. *Blackbox testing*

Halaman	Data Masukan	Hasil yang diharapkan	Hasil Pengujian	
			Benar	Salah
Register	Memasukkan email atau password yang valid.	Berhasil mendaftarkan akun kemudian diarahkan ke halaman <i>login</i> .	✓	
	Memasukkan email atau password yang tidak valid.	Gagal mendaftarkan akun.	✓	
Login	Memasukkan email dan password yang terdaftar.	Berhasil <i>login</i> .	✓	
	Memasukkan email dan password salah.	Gagal <i>login</i> .	✓	
Tambah produk	Memasukkan semua data produk.	Berhasil menambahkan produk kemudian diarahkan ke halaman list produk.	✓	
	Mengosongkan beberapa <i>field</i> data produk.	Gagal menambahkan produk.	✓	
Penjualan offline	Menambahkan produk dengan stok yang tersedia kedalam transaksi.	Berhasil melakukan transaksi penjualan <i>offline</i> .	✓	
	Menambahkan produk dengan stok yang kosong kedalam transaksi.	Gagal melakukan transaksi penjualan <i>offline</i> .	✓	
Belanja online	Menambahkan produk dengan stok yang tersedia kedalam transaksi.	Berhasil melakukan transaksi penjualan <i>online</i> .	✓	
	Menambahkan produk dengan stok yang kosong kedalam transaksi.	Gagal melakukan transaksi penjualan <i>online</i> .	✓	
Konfirmasi penjualan online	Menekan tombol proses pesanan.	Menampilkan pesan berhasil memproses pesanan.	✓	

5. KESIMPULAN DAN SARAN

Berdasarkan penelitian yang telah dilakukan dalam perancangan dan pengembangan sistem yang berjudul "Rancang Bangun Sistem Informasi Penjualan Hybrid Berbasis Website ", dapat disimpulkan bahwa UML (*Unified Modelling Language*) berhasil membantu untuk memodelkan sistem. Sistem berbasis *website* telah berhasil dibuat menggunakan bahasa pemrograman *Node.js* dan *PostgreSQL* sebagai *database management system*. Hasil dari pengujian *alpha testing* dengan metode *blackbox* membuat sistem dapat dijalankan sesuai kebutuhan tanpa ada *bug* sehingga sistem yang telah dibuat mampu mengatasi masalah proses penjualan *hybrid* dan dapat memenuhi kebutuhan pelanggan.

Dalam pengembangan berikutnya, disarankan untuk menambahkan fitur-fitur seperti fitur voucher, poin belanja, dan lainnya untuk meningkatkan loyalitas pelanggan.

DAFTAR PUSTAKA

- [1] Prabaningrum, B., Voutama, A. and Heryana, N., 2023. RANCANG BANGUN SISTEM MANAJEMEN KEUANGAN BERBASIS WEBSITE DALAM PENGELOLAAN LABA RUGI. *JATI (Jurnal Mahasiswa Teknik Informatika)*, 7(1), pp.671-680.
- [2] Tucunan, H.D.W. and Heryana, N., 2023. SISTEM INFORMASI PERSEDIAAN PRODUK MAKANAN RINGAN BERBASIS WEB. *JATI (Jurnal Mahasiswa Teknik Informatika)*, 7(1), pp.45-53
- [3] Safitri, R., 2019. Penerapan Metode Single Exponential Untuk Peramalan Volume Penjualan Minuman Kemasan (Studi Kasus: PT Coca Cola Amatil Indonesia Medan). *JURIKOM (Jurnal Riset Komputer)*, 6(1), pp.100-105
- [4] Dari Luring ke Daring, Sementara atau Selamanya Ding, K. and Jiang, P., 2017. RFID-based production data analysis in an IoT-enabled smart job-shop. *IEEE/CAA Journal of Automatica Sinica*.
- [5] Susilo, M., 2018. Rancang Bangun Website Toko Online Menggunakan Metode Waterfall. *InfoTekJar: Jurnal Nasional Informatika dan Teknologi Jaringan*, 2(2), pp.98-105.
- [6] Voutama, A., 2022. Sistem Antrian Cucian Mobil Berbasis Website Menggunakan Konsep CRM dan Penerapan UML. *Komputika: Jurnal Sistem Komputer*, 11(1), pp.102-111.
- [7] Rozaq, A., Hardinto, R.K. and Yunida, R., 2018. KAJIAN PENGEMBANGAN SISTEM INFORMASI PERIJINAN TRAYEK DINAS PERHUBUNGAN, KOMUNIKASI DAN INFORMATIKA KOTA BANJARMASIN.
- [8] Nur, H., 2019. Penggunaan Metode Waterfall Dalam Rancang Bangun Sistem Informasi Penjualan. *Generation Journal*, 3(1), pp.1-10.
- [9] Kurniawan, I. and Rozi, F., 2020. REST API Menggunakan NodeJS pada Aplikasi Transaksi Jasa Elektronik Berbasis Android. *JITSI: Jurnal Ilmiah Teknologi Sistem Informasi*, 1(4), pp.127-132.
- [10] Shadiq, J., Safei, A. and Loly, R.W.R., 2021. Pengujian Aplikasi Peminjaman Kendaraan Operasional Kantor Menggunakan BlackBox Testing. *Information Management For Educators And Professionals: Journal of Information Management*, 5(2), pp.97-110