

APLIKASI PEMESANAN BERBASIS WEBSITE PADA DOBRO COFFEE DENGAN MENERAPKAN *PAYMENT GATEWAY MIDTRANS*

Fadya Angelina Raharjo^{1*}, Intan Purnamasari², Sofi Defiyanti³

^{1,2,3} Program Studi Informatika, Fakultas: Ilmu Komputer, Universitas Singaperbangsa Karawang
Institusi

Keywords:

Pemesanan Daring;
Website;
Midtrans;
Payment Gateway;
Waterfall;
F&B.

Correspondent Email:

2110631170123@student.unsika.ac.id

Abstrak. Pertumbuhan kedai kopi di Bekasi (120 unit tahun 2021) memperkuat persaingan. Dobro Coffee menghadapi masalah antrian (10-15 menit), kesalahan input pesanan, yang berdampak pada penurunan penjualan (Rp65.000 Oktober 2024 menjadi Rp49.000 Januari 2025). Penelitian ini mengusulkan aplikasi pemesanan website terintegrasi Payment Gateway Midtrans, dikembangkan dengan metode SDLC Waterfall (analisis kebutuhan, perancangan UML & basis data, implementasi PHP, MySQL, Bootstrap, Laragon, pengujian fungsional). Hasilnya adalah prototipe dengan modul pelanggan (registrasi, katalog menu, pembayaran Midtrans) dan admin (dasbor, kelola menu/pesanan). Pengujian SUS pada 25 responden menghasilkan skor 82,5 ('Good' dan 'Acceptable'). Sistem diproyeksikan mengurangi antrian, kesalahan input, dan meningkatkan efisiensi.



Copyright © [JITET](#) (Jurnal Informatika dan Teknik Elektro Terapan). This article is an open access article distributed under terms and conditions of the Creative Commons Attribution (CC BY NC)

Abstract. The growth of coffee shops in Bekasi (120 units in 2021) has intensified competition. Dobro Coffee faces problems of long queues (10-15 minutes), order input errors, which impact sales decline (Rp65,000 in October 2024 to Rp49,000 in January 2025). This research proposes a website-based ordering application integrated with Midtrans Payment Gateway, developed using the Waterfall SDLC method (needs analysis, UML & database design, implementation with PHP, MySQL, Bootstrap, Laragon, functional testing). The result is a prototype with customer modules (registration, menu catalog, Midtrans payment) and admin modules (dashboard, menu/order management). SUS testing on 25 respondents resulted in a score of 82.5 ('Good' and 'Acceptable'). The system is projected to reduce queues, input errors, and improve efficiency.

1. PENDAHULUAN

Bisnis kuliner, yang meliputi berbagai usaha makanan dan minuman kini semakin menyadari betapa pentingnya peran teknologi dalam mendukung dan meningkatkan efisiensi kegiatan operasional mereka sehari-hari, terutama di tengah persaingan yang semakin ketat di sektor kuliner, di mana kedai kopi menjadi salah satu segmen yang paling berkembang pesat di kalangan masyarakat Indonesia, salah satunya di Kota Bekasi, Jawa Barat. Terdapat peningkatan jumlah usaha 120 unit *Coffee Shop* (kedai kopi) yang beroperasi di Kota Bekasi [1].

Dalam bisnis yang sangat kompetitif ini, penggunaan *website* yang efektif dapat menjadi kunci untuk menarik pelanggan dan meningkatkan penjualan. Banyak kedai kopi kini mengadopsi sistem pemesanan *online* yang memungkinkan pelanggan untuk melihat menu, dan melakukan pemesanan hanya beberapa klik. Hal ini dapat meningkatkan kenyamanan bagi pelanggan, hal itu juga mengurangi waktu tunggu dan antrian. Namun masih banyak juga kedai kopi yang masih belum mempunyai *website*, salah satunya Dobro Coffee.

Dobro Coffee merupakan kedai kopi yang berlokasi di Pejuang Kota Bekasi. Dobro Coffee merupakan salah satu bisnis kuliner yang menyediakan berbagai cita rasa *coffee*, *non-coffee*, *main course* dan *snack* bagi pelanggan. Berdasarkan pengamatan dan wawancara dengan pihak manajemen Dobro Coffee, saat ini bisnis tersebut belum memiliki *website*, selain itu berdasarkan hasil survei dan wawancara peneliti kepada tim Dobro Coffee dan kepada beberapa pelanggannya, permasalahan yang ada di Dobro Coffee saat ini terkait dengan sistem pemesanan dan pengelolaan informasi yang masih konvensional.

Saat ini, proses pemesanan dilakukan secara manual melalui kasir, di mana pelanggan harus mengantri hanya untuk melakukan pemesanan. Sistem ini menyebabkan antrian yang terlalu panjang, terutama pada jam-jam sibuk pelanggan harus menunggu 10 sampai 15 menit untuk melakukan pemesanan yang tidak hanya mengakibatkan ketidaknyamanan bagi pelanggan tetapi juga berpotensi kehilangan konsumen yang memilih untuk pergi ke tempat lain yang lebih efisien. Selain itu, proses manual ini sering kali mengakibatkan kesalahan dalam penginputan pesanan, di mana pesanan yang diterima oleh karyawan Dobro Coffee tidak sesuai dengan yang diminta oleh pelanggan. Kesalahan ini dapat merugikan reputasi Dobro Coffee dan mengurangi kepuasan pelanggan. Di samping itu, karyawan Dobro Coffee juga mengalami kesulitan dalam memantau status pesanan secara *real-time*, yang dapat menyebabkan kebingungan dan keterlambatan dalam penyajian produk kepada pelanggan. Salah satu penyebab terjadinya masalah ini adalah karena belum adanya sistem pemesanan *online*.

Karena salah satu dampak dari permasalahan tersebut, Dobro Coffee mengalami penurunan dalam penjualannya, khususnya dalam jangka waktu lima bulan terakhir, tepatnya dari bulan September 2024 hingga Januari 2025, di mana penurunan tersebut menjadi perhatian serius bagi manajemen untuk segera mencari solusi yang efektif. Berikut adalah data penjualan selama 5 bulan terakhir di Dobro Coffee dari September 2024 hingga Januari 2025.

Masalah ini penting untuk segera diperbaiki, mengingat banyaknya penurunan pada penjualan dan konsumen yang mengeluh mengenai keterlambatan dalam pemesanan dan ketidaknyamanan dalam berkomunikasi. Oleh sebab itu,

dibutuhkan solusi yang lebih efisien agar proses pemesanan dan transaksi dapat dikelola dengan lebih terstruktur. Salah satu solusi yang bisa diimplementasikan adalah membangun sistem informasi pemesanan berbasis *website*, dengan memanfaatkan teknologi untuk menyederhanakan proses pemesanan sekaligus meningkatkan efisiensi layanan. Sistem informasi ini dapat menggantikan metode konvensional yang ada, serta memungkinkan pemesanan dilakukan secara lebih praktis dan terorganisir [2].

2. TINJAUAN PUSTAKA

2.1 Aplikasi

Aplikasi merupakan perangkat lunak atau program yang dibuat untuk menjalankan tugas pada perangkat seperti komputer, laptop, maupun *smartphone*. Kata "aplikasi" sendiri berasal dari bahasa Inggris "*application*," yang berarti penerapan atau penggunaan [3]. Secara teknis, bagian belakang (*back-end*) aplikasi dikembangkan oleh *programmer* dengan menggunakan bahasa pemrograman tertentu, sedangkan bagian depan (*front-end*) nantinya dapat diakses oleh pengguna setelah aplikasi diinstal di perangkat elektronik. Aplikasi ini dapat digunakan untuk berbagai tujuan, seperti pengolahan dokumen, komunikasi, desain grafis, pengelolaan perangkat keras, bermain *game*, dan lain sebagainya.

2.2 Website

Website merupakan layanan informasi yang menggunakan konsep *clientserver* dimana antara pengguna dengan *administrator* saling memberikan data atau informasi yang dapat memudahkan keduanya. Informasi dan data yang disajikan dalam halaman mempunyai teknologi layanan informasi, multimedia (gambar, suara, animasi, *video*). *Website* berkembang sangat cepat dipengaruhi oleh faktor jaringan internet, dimana *website*

tidak akan berfungsi jika tidak adanya internet [4].

2.3 Payment Gateway

Payment Gateway adalah sistem elektronik yang memfasilitasi transaksi pembayaran antara pembeli, penjual, dan bank. *Payment Gateway* bertindak sebagai jembatan, menghubungkan ketiga pihak untuk memproses pembayaran secara elektronik menggunakan berbagai metode, seperti kartu kredit, uang elektronik, atau metode lainnya. Banyak penyedia layanan *web hosting e-commerce* mengintegrasikan *Payment Gateway* ke dalam paket mereka. Sistem ini memanfaatkan internet dan dirancang untuk otomatisasi dan keamanan transaksi, menawarkan akurasi dan otorisasi pembayaran yang efisien [5].

2.4 Midtrans

Midtrans merupakan *payment gateway* terbesar di Indonesia, *midtrans* ini sudah banyak digunakan oleh toko-toko *online* besar bahkan Bukalapak, Airy, Tokopedia, JD.ID, Akulaku, dsb. Tetapi akan dikenai biaya potongan untuk setiap transaksi yang dilakukan dalam *payment gateway midtrans*, namun biayanya masih tergolong murah. Dengan menggunakan *midtrans* ini dapat memudahkan penjualan dengan cara konfirmasi pesanan otomatis, serta 21 metode pembayaran yang beragam seperti Gopay, Ovo, Dana, dll [6].

2.5 System Development Life Cycle

System Development Life Cycle (SDLC) pengembangan sistem programmer yang sudah ada. Tahapan SDLC-nya terdiri dari lima tahapan yaitu penelitian, analisis, desain, pengkodean, dan pemeliharaan[7].

2.6 Model Waterfall

Metode *Waterfall* adalah pendekatan pengembangan perangkat lunak tradisional dan sistematis yang berjalan secara berurutan melalui tahapan-tahapan yang berbeda [8]. Model ini dimulai dengan

analisis kebutuhan pengguna, dilanjutkan dengan perancangan sistem (perangkat keras dan lunak), implementasi (pengkodean), pengujian, dan akhirnya pemeliharaan sistem setelah peluncuran.

2.7 Hyper Text Preprocessor (PHP)

PHP (*Hypertext Preprocessor*) adalah bahasa pemrograman *server-side* yang populer dan banyak digunakan untuk membangun situs *web*, sering dipadukan dengan HTML [9]. Awalnya dikenal sebagai *Personal Home Page*, PHP kini telah berkembang menjadi bahasa yang kuat, mampu membangun situs *web* besar dan kompleks seperti Wikipedia dan WordPress. PHP dieksekusi di *server* dan memungkinkan pembuatan situs *web* dinamis dengan efisien.

2.8 Cascading Style Sheet (CSS)

CSS memiliki kepanjangan yaitu Cascading Style Sheet yang merupakan sebuah media yang digunakan untuk mempermudah pengelolaan pada suatu halaman website. Suatu halaman pada sebuah website dapat dimodifikasi tanpa diharuskannya mengganti file HTML-nya. CSS biasanya digunakan pada file HTML berguna untuk membuat penyajian tampilan yang dapat meningkatkan kualitas penggunaan HTML itu sendiri menjadi lebih luas. Selain itu, penyajian tampilan juga dikelola pada dokumen terpisah yang tidak meningkatkan kesulitan pada sebuah pengelolaan dokumen tersebut [10].

2.9 BlackBox

Blackbox testing dilakukan meliputi *input* dan *output* sistem, serta memastikan bahwa seluruh fitur yang ada pada aplikasi dapat bekerja sesuai harapan. Pada konteks aplikasi pemesanan menu ini, *blackbox* akan memastikan bahwa seluruh proses pemesanan menu, pembayaran, dan dapat berhasil seluruhnya. Prinsip dasar yang dimiliki oleh pengujian kotak

hitam ini adalah untuk mendeteksi dari berbagai jenis kesalahan yang ada pada sistem aplikasi yang meliputi fungsi sistem yang tidak berjalan dengan baik, salah serta kesalahan pada tampilan antarmuka, serta performa yang kurang optimal [11].

2.10 Laragon

Laragon adalah aplikasi server lokal yang mudah digunakan dan bebas, kompatibel dengan berbagai sistem operasi seperti Windows, MacOS, dan Linux [12]). Aplikasi ini menyediakan berbagai layanan, alat, dan fitur seperti Apache, MySQL, PHP Server, Memcached, Redis, Composer, Xdebug, PhpMyAdmin, Cms, dan Laravel. Keunggulan Laragon terletak pada fitur-fiturnya yang lengkap, antarmuka *grafis* (GUI) yang ramah pengguna, dan dukungan untuk berbagai bahasa pemrograman dan teknologi termasuk Git, Node.js/MongoDB, Python/Django/Flask/Postgre, Ruby, Java, dan Go.

2.11 White Box Testing

White Box Testing adalah sebuah Teknik pengujian aplikasi yang melibatkan pemahaman terhadap struktur kontrol dalam level komponen desain, dengan tujuan untuk menghasilkan kasus uji. *White box testing* memiliki beberapa metode dalam pengujian, seperti: *Data Flow Tesing*, *Control Flow Testing*, *Basic Path/Path Testing*, dan *Loop Testing*. Dalam *White Box Testing*, seorang penguji harus mengetahui kode sumber yang diuji. *White Box Testing* memiliki potensi untuk menemukan kekurangan dalam implementasi aplikasi. Tes ini dapat diterapkan pada Tingkat integrasi, unit dan system [13].

2.12 Visual Studio Code

Visual Studio Code adalah versi ringan dari lingkungan pengembangan Microsoft resmi yang secara eksklusif yang hanya berfokus pada kode editor. *VSCode* bersifat

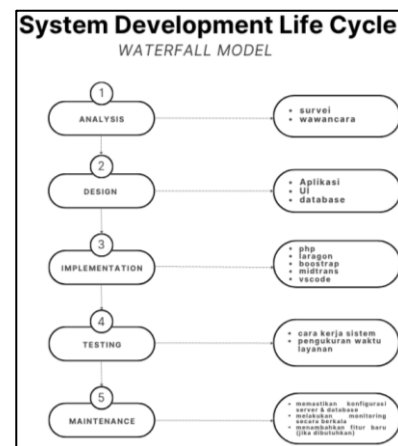
Multiplatform dan mendukung sintaksis dalam banyak bahasa pemrograman. *VSCode* menyediakan dukungan banyak bahasa berbeda seperti HTML, CSS, Java Script, C, JSON, PHP, SQL, Ruby, Visual Basic, PHO dan lain sebagainya. Mendukung highlight, snippet, autoindentasi dan autokomplet. VS Code menyediakan fleksibilitas dan kesederhanaan dalam mendukung kompatibilitas antar platform yang ditawarkan, diluar dukungan repositori Git atau kemampuan untuk membaja beberapa iterasi berkas dalam satu jendela [14].

2.13 *Unified Modelling Language (UML)*

UML adalah bahasa yang digunakan untuk menggambarkan, menentukan, membangun dan mencatat komponen-komponen pada perangkat lunak. Artefak-artefak ini mencakup seperti model-model, penjelasan, atau perangkat dalam sistem perangkat lunak, seperti bisnis dan yang lainnya. Disamping itu, UML merupakan suatu bentukbahasa permodelan yang menerapkan prinsip orientasi objek. Grady Booch, James Rumbaugh, dan Ivar Jacobson menciptakan UML dalam kerangka Rational Software Corps. UML sendiri menawarkan seperti simbol-simbol untuk membantu dalam menggambarkan sistem dari sudut pandang yang berbeda. UML juga digunakan bukan hanya dalam permodelan perangkat lunak, tetapi juga dalam bidang lain yang sangat memerlukan permodelan[15].

3. METODE PENELITIAN

Metode penelitian yang digunakan untuk menyelesaikan masalah yaitu SDLC model waterfall. Metode ini digunakan karena menyediakan proses secara sekuensial linier atau berurutan yang mulai dari analisis, perancangan, implementasi, pengujian, dan evaluasi. Berikut penjelasan mengenai alur penelitian pada Gambar 1.



Gambar 1. Alur Penelitian

- a. Tahap Analisis
Tahap awal analisis data dilakukan dengan cara wawancara, observasi dan pengamatan secara langsung mengenai permasalahan dan proses bisnis pada Dobro Coffee.
- b. Tahap Perancangan
Tahap perancangan sistem adalah proses perancangan desain aplikasi seperti *Use Case Diagram*, *database* dan perancangan desain *interface*.
- c. Implementasi
Tahap ini akan dilaksanakan pembuatan kode program aplikasi yang ada pada tahap perancangan sistem untuk menjadi aplikasi.
- d. Pengujian
Tahap pengujian ini menggunakan blackbox testing pengujian yang dilakukan tanpa melihat struktur internal kode program.
- e. Maintenance
Tahapan ini nantinya akan melakukan pemeliharaan sistem yang sudah dipakai oleh pengguna agar sistem dapat dilakukan perbaikan, penambahan fungsi lain bisa dilakukan dalam aplikasi maka akan diperbaharui atau dilakukan analisis ulang.

4. HASIL DAN PEMBAHASAN

4.1. Hasil

Dalam penelitian yang dilakukan terhadap system pemesanan pada Dobro Coffee,

didapatkan hasil berdasarkan pengamatan dan intervensi. Dalam membangun aplikasi berbasis website ini, akan menggunakan Bahasa pemrograman PHP kemudian mengikuti metode System Development Life Cycle (SDLC) dengan model waterfall. Proses aplikasi berbasis website ini akan dilakukan dengan tahapan-tahapan berikut:

Analysis

1. Analisis Kebutuhan User

Dalam aplikasi yang dibuat, terdapat dua peran pengguna yaitu admin dan pelanggan. Admin bertanggung jawab memeriksa pesanan yang masuk serta menambahkan data menu, sedangkan pelanggan mampu melihat menu yang tersedia dan memesan melalui aplikasi tersebut.

2. Analisis Kebutuhan Aplikasi

Penelitian ini dilakukan untuk mendapatkan gambaran yang jelas tentang jenis-jenis kebutuhan yang dibutuhkan pengguna dalam aplikasi tersebut. Melalui wawancara yang sesuai dengan analisis tersebut, ditemukan beberapa hal yang dibutuhkan pengguna:

- Memudahkan pelanggan melakukan pemesanan via aplikasi berbasis web.
- Memudahkan admin melihat pesanan yang masuk dan menampilkan menu yang tersedia di web.

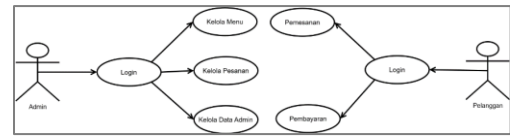
System Design

Dalam merancang desain ini, menggunakan permodelan UML yang meliputi: use case diagram, activity diagram dan sequence diagram. Pada desain database menggunakan class diagram.

Use Case Diagram

Berikut ini merupakan *use case diagram* yang telah dirancang dalam membuat Sistem Pemesanan

online Berbasis Website pada Dobro Coffee. Sebagai berikut:

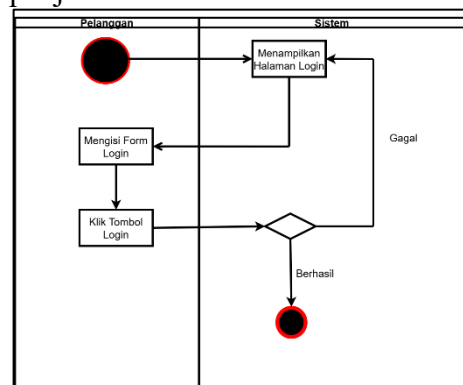


Gambar 2. Use Case Diagram

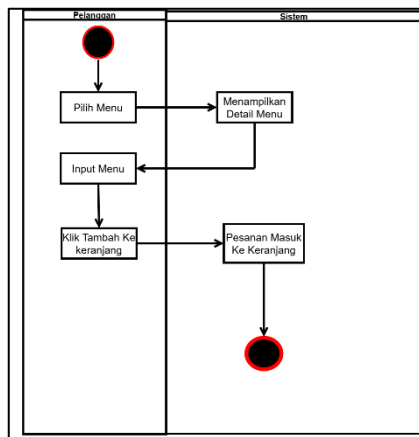
Dapat dilihat bahwa terdapat penggambaran aktivitas, Admin memiliki hak akses login sistem. Kemudian admin dapat mengakses mengolah kategori menu, data pesanan, dan data admin. Sedangkan user atau *customer* memiliki hak akses untuk melihat menu, melakukan pemesanan, dan melakukan transaksi.

Activity Diagram

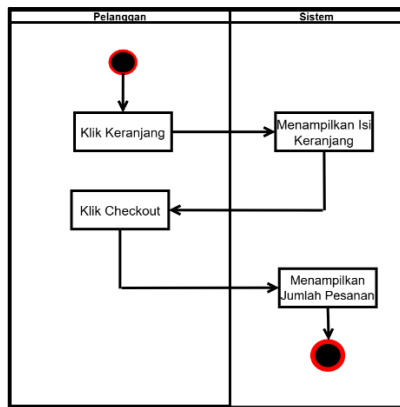
Untuk memahami interaksi antara aplikasi dan pengguna, dibutuhkan sebuah diagram yang bisa menjelaskan tahapan aktivitas dari berbagai skenario yang ada di diagram *use case*. Berikut adalah *activity diagram* yang diperlukan untuk sistem pemesanan online berbasis web, yang dibuat dalam rangka merencanakan strategi penjualan di Dobro Coffee.



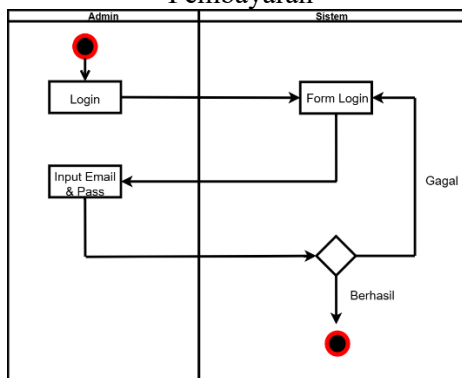
Gambar 3. Activity Diagram Login



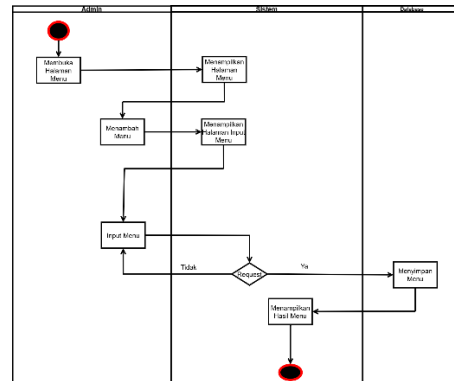
Gambar 4. Activity Diagram Pemesanan



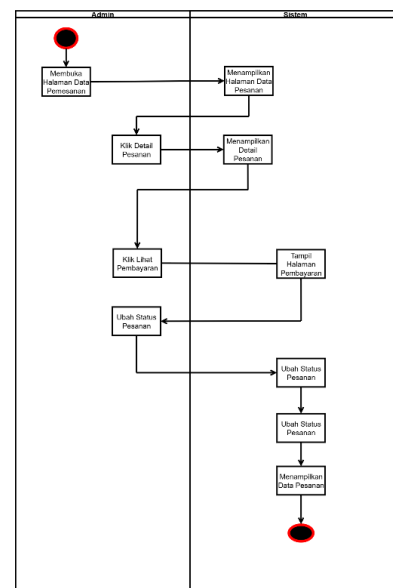
Gambar 5. Activity Diagram Pembayaran



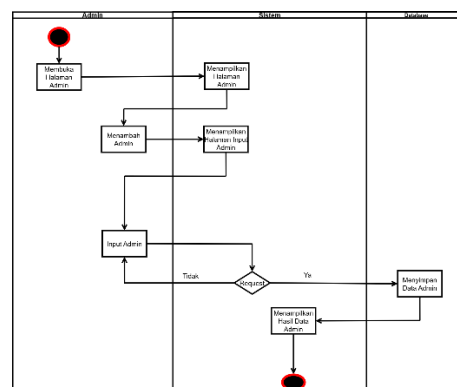
Gambar 6. Activity Diagram Login – Admin



Gambar 7. Activity Diagram Kelola Menu



Gambar 8. Activity Diagram Kelola Pesanan



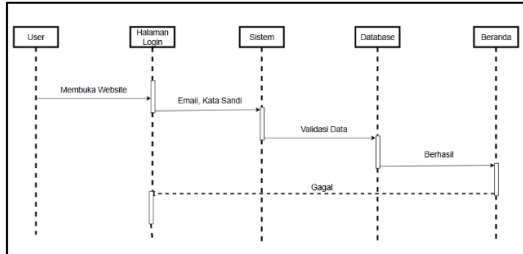
Gambar 9. Activity Diagram Kelola Data Admin

Sequence Diagram

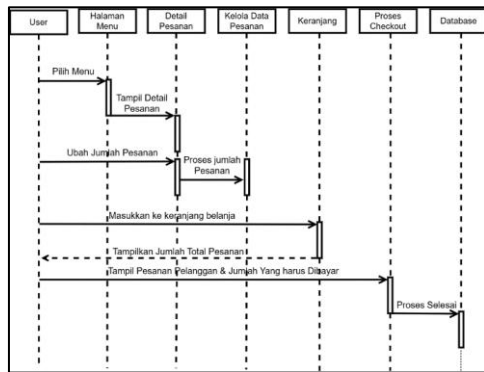
Sequence Diagram menggambarkan urutan interaksi objek dengan menggambarkan masa hidup objek dan

message yang dikirim dan diterima oleh objek.

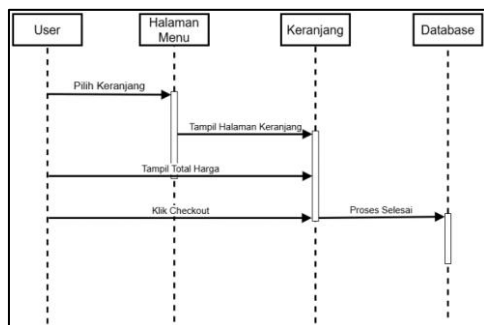
Pada *sequence diagram*, sistem ini memiliki beberapa bagian yang akan jabarkan pada gambar-gambar berikut:



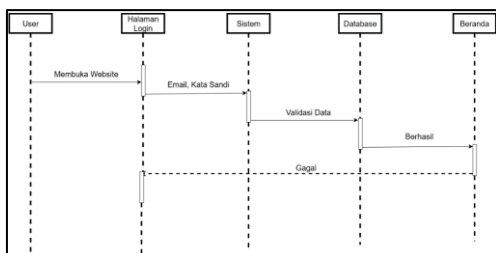
Gambar 10. *Sequence Diagram* Login



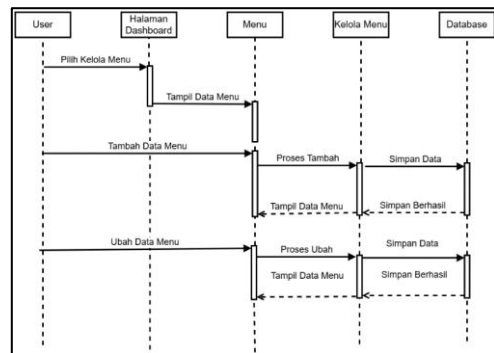
Gambar 11. *Sequence Diagram* Pemesanan



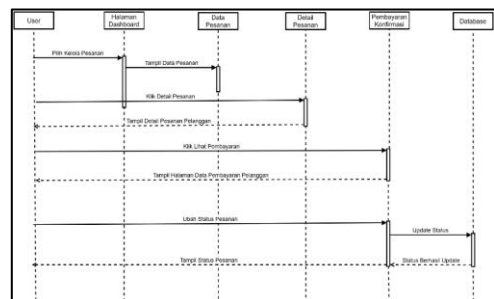
Gambar 12. *Sequence Diagram* Pembayaran



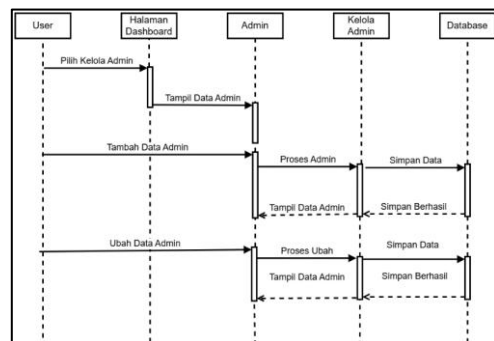
Gambar 13. *Sequence Login-Admin*



Gambar 14. *Sequence Diagram* Kelola Menu



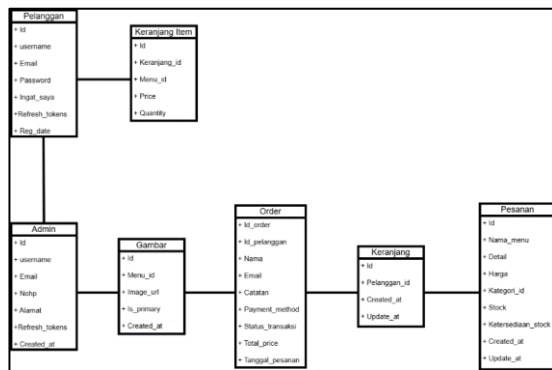
Gambar 15. *Sequence Diagram* kelola Pesanan



Gambar 16. *Sequence Diagram* Kelola Admin

Class Diagram

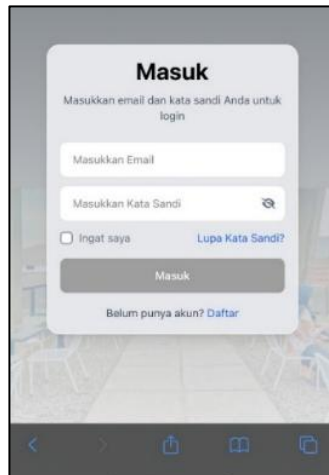
Dibawah ini merupakan gambar dari *class diagram* yang mana di dalamnya terdapat hubungan antar objek dari setiap kelas yang memiliki atribut yang berbeda.



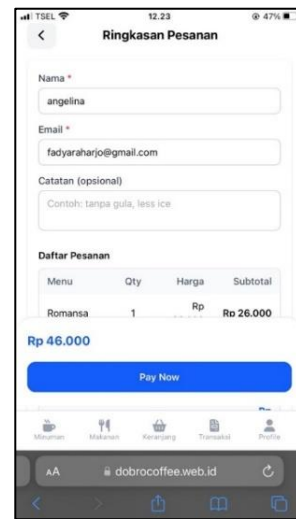
Gambar 17. Class Diagram

Implementasi

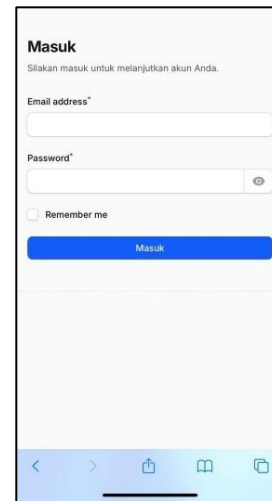
Berikut Berikut adalah *interface* pada sistem pemesanan *online* berbasis *website* dalam upaya perencanaan strategi penjualan di Dobro Coffee. Sebagai berikut:



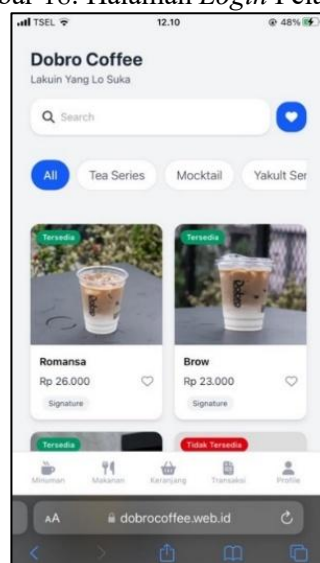
Gambar 18. Halaman Login Pelanggan



Gambar 20. Halaman Pembayaran



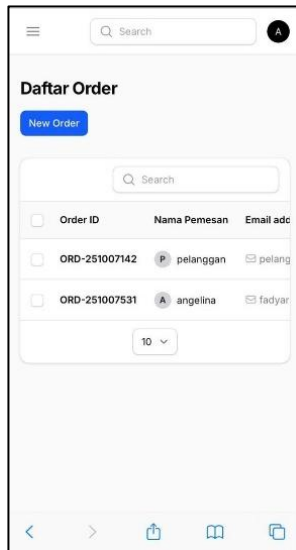
Gambar 21. Halaman Login – Admin



Gambar 19. Halaman Menu



Gambar 22. Halaman Dashboard



Gambar 23. Halaman Kelola Pesanan

Testing

Pengujian dilakukan bertujuan untuk mengecek Kembali masing-masing fitur dan fungsi yang terdapat pada sistem serta mengetahui jika terdapat kesalahan. Pengujian atau testing aplikasi yang telah dibuat menggunakan pengujian *Black Box Testing* dan *White Box Testing*.

a. BlackBox Testing

Berikut hasil Black Box Testing pada aplikasi sistem pemesanan telah diperoleh.

Tabel 1. *BlackBox Testing Website Pemesanan*

No .	Penguji an	Test Case	Hasil Penguji an	Kesimpul an
1.	Saat Mengisi Bagian Login Web Pemesan an	Klik Login & Mengisi	Sesuai Harapan	Valid
2.	Saat Mengisi Bagian Daftar Akun Web Pemesan an	Klik Daftar & Mengisi	Sesuai Harapan	Valid
3.	Saat Klik Profile	Klik Profile &	Sesuai Harapan	Valid

No .	Penguji an	Test Case	Hasil Penguji an	Kesimpul an
	Custom er	Menguba h Profile		
4.	Saat Klik Menu Minuma n Web Pemesan an	Memilih Menu Pada Minuman	Sesuai Harapan	valid
5.	Saat Klik Menu Makanan Web Pemesan an	Memilih Menu Pada Makanan	Sesuai Harapan	valid
6.	Saat Klik Detail Menu Minuma n & Makanan	Melihat Detail Menu Minuman & Makanan	Sesuai Harapan	valid
7.	Saat Klik Tambah Keranja n g	Melakuka n Checkout Pemesana n	Sesuai Harapan	valid
8.	Saat Klik Transaks i	Memilih Metode Pembayar an	Sesuai Harapan	valid

Tabel 2. Tabel *Blackbox Testing Pada Website Admin*

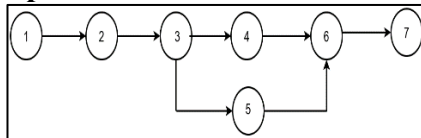
No .	Pengujian	Test Case	Hasil Pengujian	Kesimpula n
1.	Saat Bagian Login Web Admin	Mengisi Bagian Login	Sesuai Harapan	Valid
2.	Saat Klik Bagian Sidebar	Muncul List Pilihan Yang Ada Di Sidebar	Sesuai Harapan	Valid
3.	Saat Bagian Kelola Admin	Klik Bagian Kelola Admin, Klik Pada Bagian Tambah Admin.	Sesuai Harapan	Valid
4.	Saat Bagian Kelola Jenis	Klik Bagian Kelola Jenis, dan Menambahk an Kategori	Sesuai Harapan	Valid
5.	Saat Bagian Kelola Menu	Klik Bagian Kelola Menu dan Menambahk an Menu	Sesuai Harapan	Valid

No	Pengujian	Test Case	Hasil Pengujian	Kesimpulan
6.	Saat Bagian Kelola Order	Klik Bagian Kelola Order dan Menambahk an Order	Sesuai Harapan	Valid
7.	Saat Bagian Kelola Customer	Klik Bagian Kelola Customer dan Menambahk an Customer.	Sesuai Harapan	Valid

b. WhiteBox Testing

Berikut hasil *WhiteBox Testing* pada aplikasi sistem pemesanan telah diperoleh.

Tampilan Menu



Gambar 24. Flowgraph Menu

Pada Gambar 24 jumlah jalur yang terbentuk dapat dikalkulasikan menggunakan rumus perhitungan *cyclomatic complexity*.

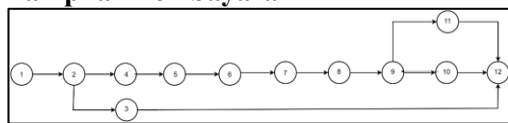
$$V(G) = E - N + 2$$

$$V(G) = 7 - 7 + 2 = 2$$

Berdasarkan hasil perhitungan diatas, diperoleh jalur sebagai berikut:

- Jalur 1 : 1 – 2 – 3 – 4 – 6 – 7
- Jalur 2 : 1 – 2 – 3 – 5 – 6 – 7

Tampilan Pembayaran



Gambar 25. Flowgraph Pembayaran

Pada Gambar 25 jumlah jalur yang terbentuk dapat dikalkulasikan menggunakan rumus perhitungan *cyclomatic complexity*.

$$V(G) = E - N + 2$$

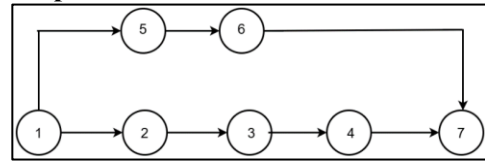
$$V(G) = 13 - 12 + 2 = 3$$

Berdasarkan hasil perhitungan diatas, diperoleh jalur sebagai berikut:

- Jalur 1 : 1 – 2 – 4 – 5 – 6 – 7 – 8 – 9 – 10 – 12
- Jalur 2 : 1 – 2 – 4 – 5 – 6 – 7 – 8 – 9 – 11 – 12

- Jalur 3 : 1 – 2 – 3 – 12

Tampilan Dashboard



Gambar 26. Flowgraph Dashboard

Pada Gambar 26 jumlah jalur yang terbentuk dapat dikalkulasikan menggunakan rumus perhitungan *cyclomatic complexity*.

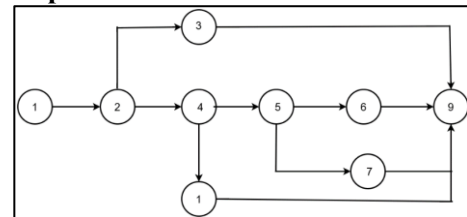
$$V(G) = E - N + 2$$

$$V(G) = 7 - 7 + 2 = 2$$

Berdasarkan hasil perhitungan diatas, diperoleh jalur sebagai berikut:

- Jalur 1 : 1 – 2 – 3 – 4 – 7
- Jalur 2 : 1 – 5 – 6 – 7

Tampilan Kelola Order



Gambar 27. Flowgraph Kelola Order

Pada Gambar 27 jumlah jalur yang terbentuk dapat dikalkulasikan menggunakan rumus perhitungan *cyclomatic complexity*.

$$V(G) = E - N + 2$$

$$V(G) = 11 - 9 + 2 = 4$$

Berdasarkan hasil perhitungan diatas, diperoleh jalur sebagai berikut:

- Jalur 1 : 1 – 2 – 3 – 4 – 5 – 9
- Jalur 2 : 1 – 2 – 3 – 4 – 6 – 9
- Jalur 3 : 1 – 2 – 7 – 9
- Jalur 4 : 1 – 8 – 9

4.2. Pembahasan

Hasil Aplikasi Pemesanan Berbasis Website Pada Dobro Coffee dengan integrasi payment gateway Midtrans, serta mengevaluasi kinerja dan kegunaan sistem menggunakan metode System Usability Scale (SUS). Sekaligus mengkaitkannya dengan rumusan masalah dan tujuan penelitian yang telah ditetapkan.

Berdasarkan Pada tahap analisis, wawancara dengan manajer Dobro Coffee Insan Ady Nugroho (28 tahun) dan observasi terhadap staff seperti Affan (Head Kitchen, 25 tahun), Ranga (Waiters, 24 tahun), Naya (Head Barista, 23 tahun), serta Kanya (Kasir, 23 tahun) mengungkap masalah pemesanan manual yang memakan waktu 10-15 menit dan mengakibatkan antrian yang panjang, dan salah input pesanan.

Tahap desain menghasilkan Diagram Use Case, pada interaksi aktor (Pelanggan, Owner/Staff) dengan fungsi login, browse menu, pesan, bayar Midtrans, dan kelola pesanan, dan didukung Diagram Activity untuk alur pemesanan, Diagram Class/Sequence untuk struktur database MySQL (tabel pengguna, menu, pesanan, transaksi), serta antarmuka responsif Bootstrap secara optimal

Implementasi membangun aplikasi menggunakan PHP versi 3.8.0, MySQL via PHPMyAdmin di Laragon, Bootstrap HTML/CSS untuk UI, dengan fitur dashboard admin tambah/edit menu kopi, katalog pelanggan, keranjang, checkout terintegrasi Midtrans mendukung metode pembayaran Cashless, mengikuti model Waterfall hingga siap deploy hosting.

Pada pengujian testing BlackBox memverifikasi semua fungsi input-output (login, pesan, bayar) sesuai spesifikasi tanpa error, sedangkan WhiteBox memastikan logika PHP/MySQL bebas bug dan System Usability Scale (SUS) dari 25 responden skor 82,5 ("good", acceptable) mengonfirmasi kemudahan penggunaan dengan waktu pemesanan turun menjadi 3-5 menit.

Pada tahap terakhir yaitu maintenance meliputi monitoring performa server/hosting, backup MySQL harian, perbaikan bug, update keamanan Midtrans/PHP untuk cegah kelemahan atau celah pada website.

5. KESIMPULAN

Berdasarkan penelitian yang telah dilakukan maka didapatkan sebuah kesimpulan sebagai berikut:

- a. Aplikasi pemesanan berbasis website untuk Dobro Coffee yang terintegrasi *payment gateway midtrans* telah berhasil dirancang dan dikembangkan menggunakan metode SDLC model *waterfall*, meliputi analisis kebutuhan, perancangan UML (*use case, activity, class, sequence*), perancangan basis data, implementasi dengan PHP-MySQL, bootstrap, dan laragon, serta pengujian fungsional.
- b. Website yang dihasilkan untuk customer Dobro Coffee mencakup *registrasi/login*, katalog & pencarian menu, keranjang, *checkout*, pembayaran *Midtrans* dan untuk admin *dashboard*, kelola menu/pesanan/admin, yang dirancang untuk mengatasi masalah proses pemesanan manual.
- c. Sistem aplikasi website yang diusulkan mampu mengurangi antrian yang pada awalnya 10-15 menit menjadi 3-5 menit pemesanan dan dapat meminimalkan kesalahan input pemesanan, dan meningkatkan efisiensi layanan.

UCAPAN TERIMA KASIH

Penulis mengucapkan terima kasih kepada pihak-pihak terkait yang telah memberi dukungan terhadap penelitian ini.

DAFTAR PUSTAKA

- [1] Satu Data Indonesia, "Jumlah Usaha Coffee Shop di Kota Bekasi, Jawa Barat." 2024. [Online]. Available: <http://katalog.data.go.id>
- [2] Delia, "Sistem Informasi Pemesanan Online Berbasis Website," 2021.
- [3] Pane, Zamzam, and Fadillah, *Aplikasi Mobile: Konsep dan Pengembangan*. 2020.
- [4] Abdul and et al., *Pemrograman Web: Membangun Website Dinamis dengan PHP dan MySQL*. 2019.
- [5] Rahardika, *Payment Gateway: Solusi Pembayaran Elektronik yang Aman dan*

- Praktis*. 2020.
- [6] M. Fadillah, “Midtrans: Payment Gateway Terbesar di Indonesia,” 2021.
 - [7] Salam, I. A., Prihandani, K., & Purnamasari, I. (2023). *Rancang Bangun Aplikasi Profit Penjualan Motor Berbasis Desktop Konsep Arsitektur Model View Controller (Mvc)*. Jurnal Informatika Dan Teknik Elektro Terapan, 11(3s1). <https://doi.org/10.23960/jitet.v11i3s1.3495>.
 - [8] Wahid, *Model Waterfall: Metodologi Pengembangan Sistem Informasi yang Sistematis*. 2020.
 - [9] Muchlis, Suhartini, and Fajriyah, *Pemrograman Web dengan PHP: Membuat Website Interaktif dan Dinamis*. 2023.
 - [10] D, Puspitasari. (2020). *Sistem Informasi Perpustakaan Sekolah Berbasis Web*. Jurnal Pilar Nusa Mandiri, XII(1), 227–240. <https://doi.org/10.31328/ciastech.v6i1.5363>.
 - [11] C. Christian and A. Voutama, “RANCANG BANGUN APLIKASI SISTEM INFORMASI INVENTARIS BERBASIS WEBSITE,” Jurnal Informatika dan Teknik Elektro Terapan, vol. 12, no. 2, pp. 1500–1509, Apr. 2024, doi: 10.23960/jitet.v12i2.4259.
 - [12] Sahat and et al., *Laragon: Panduan Lengkap Mengatur Server Lokal untuk Website*. 2024.
 - [13] Londjo, M. F. (2021). Implementasi White Box Testing Dengan Teknik Basis Path Pada Pengujian Form Login. Jurnal Siliwaangi, 7(2), 35–40.
 - [14] Roni, Aditya, and Echa, *Visual Studio Code: Editor Kode yang Ramah Pengguna*. 2020.
 - [15] Prasetya, A. F., Sintia, & Putri, U. L. D. (2022). *Perancangan Aplikasi Rental Mobil Menggunakan Diagram UML (Unified Modelling Language)*. Jurnal Ilmiah Komputer Terapan Dan Informasi, 1(1), 14–18.