UNIVERSITAS GUNADARMA

**FAKULTAS INFORMATIKA**



**PENELITIAN ILMIAH**

**Pembuatan Website Pemesanan Masakan Melalui Barcode**

**Nama Npm Jurusan**

**: Ibnu Akmal**

**: 50421619**

**: Teknik Informatika**

**Pembimbing : Nurma Nugraha**

**Diajukan Guna Melengkapi Sebagian Syarat Dalam Mencapai Gelar Setara Sarjana Muda**

**JAKARTA 2024**

# DAFTAR ISI

Daftar Isi

[DAFTAR ISI ii](#_Toc175800147)

[DAFTAR GAMBAR v](#_Toc175800148)

[DAFTAR TABEL vi](#_Toc175800149)

[DAFTAR LAMPIRAN vii](#_Toc175800150)

[BAB I PENDAHULUAN 1](#_Toc175800151)

[1.1 Latar Belakang 1](#_Toc175800152)

[1.2 Batasan Masalah 2](#_Toc175800153)

[1.3 Tujuan Penulisan 2](#_Toc175800154)

[1.5 Sistematika Penulisan 3](#_Toc175800155)

[BAB 2 TINJAUAN PUSTAKA 4](#_Toc175800156)

[2.1 Sistem Informasi 4](#_Toc175800157)

[2.2 Pengertian Website 5](#_Toc175800158)

[2.2.1 Web Browser 5](#_Toc175800159)

[2.2.2 Web Server 5](#_Toc175800160)

[2.3 Bahasa Pemrograman 6](#_Toc175800161)

[2.3.2 PHP *(*Hypertext Preprocessor *)* 7](#_Toc175800162)

[2.3.3 CSS (Cascading Style Sheets) 7](#_Toc175800163)

[2.3.4 Javascript 8](#_Toc175800164)

[2.4 Pengertian XAMPP 8](#_Toc175800165)

[2.5 Unified Modeling Language (UML) 8](#_Toc175800166)

[2.6 Use Case Diagram 10](#_Toc175800167)

[2.7.1 Sequence Diagram 12](#_Toc175800168)

[2.7.2 Activity Diagram 13](#_Toc175800169)

[2.8 Struktur Navigasi 13](#_Toc175800170)

[2.8.1 Struktur Navigasi Linier 13](#_Toc175800171)

[2.8.2 Struktur Navigasi Non-Linier 14](#_Toc175800172)

[2.8.3 Struktur Navigasi Hirarki 14](#_Toc175800173)

[2.9 Pengujian Web (Black Box Testing) 15](#_Toc175800174)

[BAB 3 RANCANGAN & IMPLEMENTASI 21](#_Toc175800175)

[3.1 Gambaran Umum 21](#_Toc175800176)

[3.2 Tujuan Pembuatan Website 21](#_Toc175800177)

[3.3 Kebutuhan Perangkat 22](#_Toc175800178)

[3.4 Struktur Navigasi 22](#_Toc175800179)

[3.5 Use Case Diagram 23](#_Toc175800180)

[3.6 Tahapan Perancangan Tatap Muka 24](#_Toc175800181)

[3.6.1 Rancangan Halaman Home 24](#_Toc175800182)

[3.6.2 Rancangan Halaman Tentang Kami 25](#_Toc175800183)

[3.6.3 Rancangan Halaman Menu 26](#_Toc175800184)

[3.6.4 Rancangan Halaman Kontak 27](#_Toc175800185)

[3.6.5 Rancangan Halaman Keranjang (*cart)* 28](#_Toc175800186)

[3.6.6 Rancangan Halaman *Hambuerger Menu* 29](#_Toc175800187)

[3.6.7 Rancangan Halaman Admin Add Product 30](#_Toc175800188)

[3.7 Tahapan Implementasi Antarmuka 31](#_Toc175800189)

[3.7.1 Implementasi Halaman Utama 32](#_Toc175800190)

[3.7.2 Implementasi Halaman Tentang Kami 33](#_Toc175800191)

[3.7.3 Implementasi Halaman Menu 34](#_Toc175800192)

[3.7.4 Implementasi Halaman Kontak 35](#_Toc175800193)

[3.7.5 Implementasi Halaman Keranjang (*cart)* 36](#_Toc175800194)

[3.7.6 Implementasi Halaman *Hamburger Menu* 37](#_Toc175800195)

[3.7.7 Implementasi Halaman Pembayaran 38](#_Toc175800196)

[3.8 Tahap Uji Coba 39](#_Toc175800197)

[3.8.1 Blackbox Testing 39](#_Toc175800198)

[3.8.2 Tahap Uji Coba *User* (UAT) 39](#_Toc175800199)

[BAB IV PENUTUPAN 41](#_Toc175800200)

[4.1 Kesimpulan 41](#_Toc175800201)

[4.2 Saran 41](#_Toc175800202)

[DAFTAR PUSTAKA 42](#_Toc175800203)

[LAMPIRAN 1](#_Toc175800204)

Lampiran……..................................................................................................L-1

# DAFTAR GAMBAR

[Gambar 1. 1 *Struktur Navigasi Linier* 13](#_Toc175801358)

Gambar [1. 2 Struktur Navigasi Non-Linier 14](#_Toc175801359)

[Gambar 1. 3Struktur Navigasi Hirarki 15](#_Toc175801360)

[Gambar 3. 1Struktur Navigasi 22](#_Toc175801503)

[Gambar 3. 2*Use Case Diagram* 23](#_Toc175801504)

[Gambar 3. 3 Rancangan Tampilan Home 24](#_Toc175801505)

[Gambar 3. 4 Rancangan Tampilan Home 25](#_Toc175801506)

[Gambar 3. 5 Rancangan Menu. 26](#_Toc175801507)

[Gambar 3. 6 Rancangan Kontak. 27](#_Toc175801508)

[Gambar 3. 7Rancangan Keranjang 28](#_Toc175801509)

[Gambar 3. 8Rancangan *Hamburger Menu*. 29](#_Toc175801510)

[Gambar 3. 9Rancangan Pembayaran. 30](#_Toc175801511)

[Gambar 3. 10Halaman *Home* 32](#_Toc175801512)

[Gambar 3. 11Halaman Tentang Kami 33](#_Toc175801513)

[Gambar 3. 12Halaman Menu 34](#_Toc175801514)

[Gambar 3. 13Halaman Kontak 35](#_Toc175801515)

[Gambar 3. 14Halaman Keranjang 36](#_Toc175801516)

[Gambar 3. 15Halaman *Hamburger Menu* 37](#_Toc175801517)

[Gambar 3. 16Halaman Pembayaran 38](#_Toc175801518)

# DAFTAR TABEL

[Tabel 2. 1. Jenis *Relationship* 11](#_Toc175799840)

[Tabel 3. 1 Uji Coba Blackbox Testing 39](#_Toc175799841)

[Tabel 3. 2 Tabel Uji Coba *User* 40](#_Toc175799842)

# DAFTAR LAMPIRAN

index L1

Hamburger Menu L1

Shopping Cart L1

Tentang Kami L1

Home L1

Menu L1

Kontak L1

Style.css L1

App.js L1

# BAB I PENDAHULUAN

## Latar Belakang

Dalam menjalani kehidupan sehari-hari,tempat berkumpul ketika libur bersama teman,keluarga, dan pasangan merupakan hal yang sangat dibutuhkan untuk melepas penat selama hari kerja atau kuliah. Berbagai macam tempat berkumpul memungkinkan masyarakat memilih tempat apa yang diinginkan apalagi dengan pemesanan yang mudah. Oleh karena itu,inovasi ini untuk mempermudah pelanggan dengan hanya duduk di meja yang di inginkan dan memesan tampa harus beranjak.

ROOM COFFE & EATERY merupakan tempat berkumpul untuk semua kalangan yang ada di wilayah Stadion Pakansari . ROOM COFFEE & EATERY menjual berbagai macam makanan dan minuman dengan banyak pilihan dan yang paling menonjol disini adalah dibagian kopi nya.

Maka dari itu pembuatan *website* ini merupakan hal yang penting untuk membantu dalam pemesanan agar mempermudah pelanggan untuk memesan makanan atau minuman tanpa harus beranjak tempat yang sudah dipilih. Keberadaan *website* ini diharapkan agar para konsumen dapat memperoleh kemudahan dalam memilih dan memesan yang mereka inginkan, tanpa harus datang ke kasir

Dengan begitu diharapkan pembuatan website ini dapat memudahkan konsumen untuk melakukan pemesan minuman dan juga untuk mempermudah karyawan saat melakukan pekerjaan dan terfokus untuk mengantarkan pesanan.

## Batasan Masalah

Adapun batasan masalah dalam penulisan ini antara lain :

* + 1. *Website* bersifat *offline*.
    2. *Website* bersifat sementara.
    3. Konfirmasi dilakukan dengan menghubungi pelanggan untuk dimintai informasi terkait makanan yang dipesan melalui *website*.

## Tujuan Penulisan

Membuat *website* pemesanan melalui barcode pada ROOM COFFEE & EATERY dengan harapan agar dapat memudahkan para konsumen dalam memilih dan memesan minuman yang mereka inginkan tanpa harus beranjak dari tempat duduk.

* 1. **Metode Penelitian**

Untuk memecahkan suatu masalah digunakan cara atau metode tertentu agar penelitian dapat dipercaya kebenarannya . Metode yang di gunakan dalam penelitian ini adalah:

* + 1. Observasi : melakukan pengamatan dan pengumpulan data langsung dari ROOM COFFEE & EATERY.
    2. Wawancara : memperoleh informasi tentang harga dan produk yang akan dipasarkan.
    3. Analisis : menganalisis data informasi dan studi kasus yang telah didapat dari hasil observasi dan wawancara.

## Sistematika Penulisan

Berikut ini beberapa isi dari sistematika dari sistem yang dibuat, dibuat dalam 4 (empat bab).

* + 1. Pendahuluan, membahas tentang latar belakang, rumusan masalah, tujuan penulisan, metode penelitian, dan sistematika penulisan.
    2. Landasan Teori, membahas semua teori pengertian *website*, sistem informasi, xampp, basis data, PHP, MySQL dan segala tentang proses pembuatan *website* ini.
    3. Pembahasan, menyajikan hasil-hasil implementasi dari *website* yang sudah dibuat yaitu berupa rancangan, struktur, dan hasil.
    4. Penutup, menjelaskan kesimpulan yang sudah dibahas pada bab-bab sebelumnya serta saran dan masukan.

# BAB 2 TINJAUAN PUSTAKA

## Sistem Informasi

Sistem Informasi adalah kombinasi dari [teknologi informasi](https://id.wikipedia.org/wiki/Teknologi_informasi) dan aktivitas orang yang menggunakan teknologi itu untuk mendukung operasi dan manajemen. Dalam arti yang sangat luas, istilah sistem informasi yang sering digunakan merujuk kepada interaksi antara orang, proses algoritmik, data, dan teknologi. Dalam pengertian ini, istilah ini digunakan untuk merujuk tidak hanya pada penggunaan organisasi [Teknologi Informasi dan Komunikasi](https://id.wikipedia.org/wiki/Teknologi_informasi_dan_komunikasi) (TIK), tetapi juga untuk cara di mana orang berinteraksi dengan teknologi ini dalam mendukung proses bisnis.

Ada yang membuat perbedaan yang jelas antara sistem informasi, dan komputer sistem TIK, dan proses bisnis. Sistem informasi yang berbeda dari teknologi informasi dalam sistem informasi biasanya terlihat seperti memiliki komponen TIK. Hal ini terutama berkaitan dengan tujuan pemanfaatan teknologi informasi. Sistem informasi juga berbeda dari proses bisnis. Sistem informasi membantu untuk mengontrol kinerja proses bisnis.

Alter berpendapat untuk sistem informasi sebagai tipe khusus dari sistem kerja. Sistem kerja adalah suatu sistem di mana manusia dan/ atau mesin melakukan pekerjaan dengan menggunakan sumber daya untuk memproduksi produk tertentu dan/ atau jasa bagi pelanggan. Sistem informasi adalah suatu sistem kerja yang kegiatannya ditujukan untuk pengolahan (menangkap, transmisi, menyimpan, mengambil, memanipulasi dan menampilkan) informasi (Sutabri, 2005).

## Pengertian Website

*Website* atau situs dapat diartikan sebagai kumpulan halaman–halaman yang digunakan untuk menampilkan informasi teks, gambar diam atau gerak, animasi, suara dan atau gabungan dari semuanya, baik yang bersifat statis maupun dinamis yang membentuk satu rangkaian bangunan yang saling terkait, yang masing-masing dihubungkan dengan jaringan-jaringan halaman (Bekti, 2015).

## Web Browser

Menurut Sibero (2014:12) “Web Browser adalah aplikasi perangkat lunak yang digunakan untuk mengambil dan menyajikan sumber informasi web”. Sedangkan menurut Rudyanto Arief (2011:19) “web browser adalah program yang berfungsi untuk menampilkan dokumen-dokumen web dalam format HTML”. 2.1.4.

## Web Server

Menurut Sibero (2014:12) “Web Server adalah sebuah komputer yang terdiri dari perangkat keras dan perangkat lunak”. Web server adalah program aplikasi yang memiliki fungsi sebagai tempat menyimpan dokumen-dokumen web. Jadi semua dokumen web baik yang ditulis menggunakan Client scripting maupun server slide scripting tersimpan dalam direktori web server (document root) (Rudyanto Arief ,2011:19). Secara bentuk fisik dan cara kerjanya, perangkat keras web server tidak berbeda dengan komputer rumah atau PC, yang membedakan adalah kapasitas dan kapabilitasnya. Perbedaan tersebut dikarenakan web server bekerja sebagai penyedia layanan yang dapat diakses oleh banyak pengguna, sehingga dibutuhkan kapasitas dan kapabilitas yang besar dibandingkan PC.

## Bahasa Pemrograman

Bahasa pemograman suatu perangkat lunak yang menggunakan bahasa- bahasa pemograman yang digunakan untuk merancang tau membuat program sesuai keinginan dan kegunaan HTML *(Hypertext Markup Language*).

* + 1. **HTML ( *Hypertext Markup Language*)**

Pada umumnya HTML suatu bahasa yang digunakan untuk membuat halaman web. HTML juga dikenal sebagai aplikasi yang memiliki kemampuan browser .

Menurut Sutarman (2007:27)”HTML *Hypertext Markup Language)* adalah suatu bahasa yang digunakan untuk menulis halaman web”. Sedangkan menurut Larry (2012:3)”*Hypertext Markup Language* merupakan suatu metode untuk mengimplementasikan konsep hypertext dalam suatu naskah atau dokumen.

Jadi, dapat disimpulkan bahwa HTML adalah bahasa pemograman yang digunakan untuk menulis halaman web dengan metode untuk mengimplementasikan konsep hypertekt dalam suatu naskah atau dokumen.

## PHP *(*Hypertext Preprocessor *)*

Pada awalnya PHP digunakan sebagai bahasa pemrograman untuk sebuah *server-side* HTML*-embedded* dengan nama *Personal Home Page Tools*. Pertama sekali dibuat oleh Rasmus Lerdorf pada tahun 1994. Selanjutnya berganti nama menjadi FI ("Form Interpreter"), yang wujudnya berupa sekumpulan skrip yang digunakan untuk mengolah data form dari web. Selanjutnya Rasmus merilis kode sumber tersebut untuk umum pada 1995 dan menamakannya PHP/ FI, kependekan dari *Hypertext Preprocessing/ Form Interpreter*. Dengan perilisan kode sumber ini menjadi *open source*, maka banyak *programmer* yang tertarik untuk ikut mengembangkan PHP (Oktavian, 2010).

## CSS (Cascading Style Sheets)

CSS adalah salah satu bahasa desain web yang bisa dilakukan untuk mengatur sebuah halaman web menggunakan perancangan desain text berupa size, color, font, marginz, dan lain-lain. Menurut Sibero (2013:112) “CSS memiliki arti gaya menata halaman bertingkat, pada dasarnya setiap satu elemen yang sudah diformat dan memiliki anak dan sudah diformat, maka anak dari elemen tersebut secara otomatis akan mengikuti format elemen induknya”. Menurut Abdul Kadir dan Triwahyuni (2013:323) “ CSS adalah kode yang dimaksudkan untuk mengatur halaman web”. Sedangkan menurut Saputra dan Agustin (2013:6) “CSS atau yang memiliki kepanjangan Cascading Style Sheets merupakan suatu bahasa pemrograman web yang digunakan untuk megendalikan dan membangun berbagai web agar tampilan web terstruktur dan beragam”.

## Javascript

Pada tahun 1996, JavaScript secara resmi disebut sebagai ECMAScript, di mana ECMAScript 2 diluncurkan pada tahun 1998 dan ECMAScript 3 diperkenalkan pada tahun 1999. ECMAScript tersebut dikembangkan hingga akhirnya menjadi JavaScript sebagaimana yang kita kenal saat ini. JavaScript adalah bahasa pemrograman yang bersifat client side, JavaScript merupakan penyempurna tampilan dan sistem halaman web, sehingga menjadi sebuah halaman yang menarik.

## Pengertian XAMPP

XAMPP adalah perangkat lunak bebas yang mendukung banyak sistem operasi. XAMPP merupakan kompilasi dari beberapa program. Fungsinya adalah sebagai *server* yang berdiri sendiri (*localhost*) yang terdiri atas program Apache HTTP *Server*, MySQL *database* dan penerjemah bahasa yang ditulis dengan bahasa pemrograman PHP dan Perl. Nama XAMPP merupakan singkatan dari X (empat sistem operasi apapun, Apache, MySQL, PHP dan Perl). Program ini tersedia dalam GNU *General Public License* dan bebas. XAMPP merupakan web *server* yang mudah digunakan yang dapat melayani tampilan halaman web yang dinamis (Nugroho, 2013).

## Unified Modeling Language (UML)

Unified Modeling Language (UML) adalah sebuah bahasa yang telah menjadi standart dalam industri visualisasi, merancang dan mendokumentasikan sistem piranti lunak. UML dapat digunakan dalam pembuatan model untuk semua jenis aplikasi piranti lunak, dimana aplikasi tersebut dapat berjalan pada piranti keras, sistem oprasi dan jaringan apapun, serta ditulis dalam bahasa pemprograman apapun. Model yang dimaksud adalah proses merancang piranti lunak sebelum melakukan pengkodean (coding). UML mempunyai konsep dasar class dan operation untuk pemprograman software dengan menggunakan bahsa-bahsa pemprograman yang berorientasi obyek.

Diagram UML sendiri terdiri atas pengelompokan diagram-diagram sistem menurut aspek atau sudut pandang tertentu. Diagram adalah yang menggambarkan permasalahan maupun solusi dari permasalahan suatu model. Meskipun UML mempunyai banyak diagram, tetapi tidak semua diagram harus digunakan dalam pembuatan suatu sistem. Berikut ini penjelasan beberapa bagian dalam pembuatan pemodelan UML yang digunakan.

## Use Case Diagram

Merupakan gambaran graphical dari beberapa atau semua actor, karena diagram ini memberikan penjelasan umum antara sistem dengan dunia luar serta fitur-fitur apa yang harus dimiliki oleh sistem dipandang dari dunia luar tersebut. Suatu Use Case Diagram memiliki komponen berupa :

* + - * Stereotype

Menyediakan informasi tentang peranan dari elemen tanpa menyebutkan implementasinya. Diagram untuk menggambarkan use case dependency, calss, packagedan classifier. Stereotype merupakan notasi, dimana dalam diagram dituliskan dengan guillemet “<<...>>”.

* + - * Actor

Mewakili pihak yang terlibat dalam suatu use case yang terjadi. Actor dapat berupa manusia, sistem dan device yang memiliki peranan dalam keberhasilan operasi dari sistem.

* + - * Use Case

Mengidentifikasi fitur utama dari sistem. Tanpa use case sistem tidak dapat memenuhi permintaan actor. Use case mendefinisikan dan menggambarkan tujuan yang harus dicapai oleh sistem.

* + - * *Relationship*

*Relationship* berguna untuk menggambarkan hubungan antar *actor* dan *use case* dalam sistem. Jenis-jenis bentuk *relationship* seperti ditunjukkan pada Tabel 2.1.

Tabel 2. 1. Jenis *Relationship*

|  |  |  |
| --- | --- | --- |
| **Relationship** | **Fungsi** | **Notasi** |
| *Actor* | *Actor* adalah segala hal diluar sistem yang akan menggunakan sistem tersebut untuk melakukan  sesuatu. |  |
| *Usecase* | Mengidentifikasi fitur  kunci dari sistem. |  |
| *Association* | Jalur komunikasi antar aktor dengan *use case* yang saling berpartisipasi. |  |
| *Extend* | Penambahan perilaku ke dalam *use case* |  |
| *Usecase Generalization* | Hubungan antara *use case* umum dengan *use case* yang lebih spesifik, yang mewarisi dan menambah  fitur terhadapnya. |  |
| *Include* | Penambahan perilaku ke dalam *use case* dasar yang secara eksplisit | <<*include*>> |

## Sequence Diagram

Sequence Diagram mempresentasikan interaksi yang terjadi antara objek di sistem. Aspek penting dari diagram sequence adalah urutan waktu kejadian. Hal ini berarti bahwa urutan dari interaksi antara objek direpre- sentasikan secara langkah demi langkah. Objek-objek pada diagram sequ- ence saling berinteraksi satu sama lain dengan mengirim “pesan”. Suatu Sequence Diagram memiliki komponen berupa :

* + - * Object

Object adalah komponen berbentuk kotak yang mewakili sebuah class atau object. Mereka mendemonstrasikan bagaimana sebuah object berperilaku pada sebuah system.

* + - * Activation Boxes

Activation Boxes adalah komponen yang berbentuk persegi panjang yang menggambarkan waktu yang diperlukan sebuah object untuk menyelesaikan tugas. Lebih lama waktu yang dilakukan, maka activation boxes akan lebih panjang.

* + - * Actors

Actors adalah komponen yang berbentuk stick figure. Komponen yang mewakili seorang pengguna yang berinteraksi dengan system.

* + - * Lifeline

Lifeline adalah komponen yang berbentuk garis putus- putus. Lifeline biasanya memuat kotak yang berisi nama dari sebuah object. Berfungsi menggambarkan aktifitas dari object.

## Activity Diagram

Activitiy Diagram adalah diagram yang menggambarkan tentang aktifitas yang terjadi pada sistem. Dari pertama sampai akhir, diagram ini menunjukan langkah-langkah dalam proses kerja sistem yang di buat.

## Struktur Navigasi

Struktur navigasi adalah struktur alur cerita dan sebuah program. Sebelum, menyatukan elemen-elemen yang digunakan dalam aplikasi multimedia, sebaiknya mendefinisikan objek-objek dan merancang tampilan. Agar objek yang termasuk di dalam aplikasi tersebut tidak mengalami kerancuan informasi, dengan kata lain semua tampilan harus dapat memberikan informasi yang bulat dan utuh, sehinggan dapat tercapai suatu pembentukan aplikasi multimedia. Dalam pembuatan aplikasi multimedia ada empat macam struktur navigasi yang biasa digunakan.

## Struktur Navigasi Linier

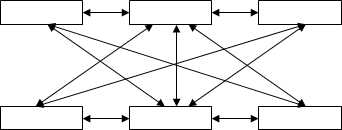
Struktur navigasi linier hanya mempunyai satu rangkaian cerita yang berurut, yang menampilkan satu demi satu tampilan layar secara berurut menurut urutannya. Tampilan yang dapat ditampilkan pada struktur jenis ini adalah satu halaman sebelumnya atau satu halaman sesudahnya, tidak dapat dua halaman sebelumnya atau dua halaman sesudahnya.



Gambar 1. 1 *Struktur Navigasi Linier*

## Struktur Navigasi Non-Linier

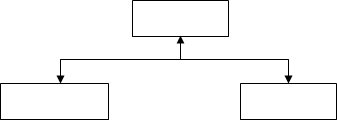
Struktur navigasi non-linier atau struktur tidak berurut merupakan pengembangan dari struktur navigasi linier. Pada struktur ini diperkenankan membuat navigasi bercabang. Percabangan yang dibuat pada struktur nonlinier ini berbeda dengan percabangan pada struktur hirarki, karena pada percabangan nonlinier ini walaupun terdapat percabangan, tetapi tiap-tiap tampilan mempunyai kedudukan yang sama yaitu tidak ada Master Page dan Slave Page.



Gambar 1. 2 Struktur Navigasi Non-Linier

## Struktur Navigasi Hirarki

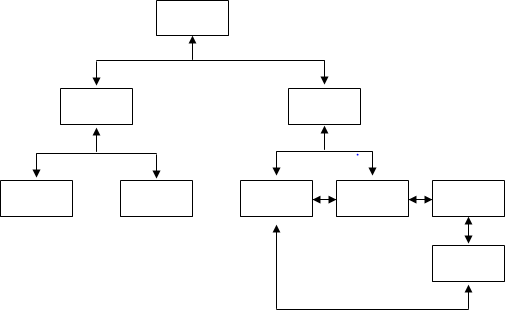
Struktur navigasi hirarki biasa disebut struktur bercabang, merupakan suatu struktur yang mengandalkan percabangan untuk menampilkan data berdasarkan kriteria tertentu. Tampilan pada menu pertama akan disebut sebagai Master Page (halaman utama pertama), halaman utama ini mempunyai halaman percabangan yang disebut Slave Page (halaman pendukung. Jika salah satu halaman pendukung dipilih atau diaktifkan, maka tampilan tersebut akan bernama Master Page (halaman utama kedua), dan seterusnya. Pada struktur navigasi ini tidak diperkenankan adanya tampilan secara linier.



Gambar 1. 3Struktur Navigasi Hirarki

* + 1. **Struktur Navigasi Campuran (*Composit*)**

Struktur navigasi campuran merupakan gabungan dari ketiga struktur sebelumnya yaitu linier, non-linier dan hirarki. Struktur navigasi ini juga biasa disebut dengan struktur navigasi bebas. Struktur navigasi ini banyak digunakan dalam pembuatan website karena struktur ini dapat digunakan dalam pembuatan website sehingga dapat memberikan ke- interaksian yang lebih tinggi.



*Gambar 2.5 Struktur Navigasi Campuran (Composit)*

## Pengujian Web (Black Box Testing)

Menurut Simarmata (2010d:316) klasifikasi black box testing mencakup beberapa pengujian, yaitu:

1. Pengujian Fungsional (Functional Testing)

Pengujian dilakukan dalam bentuk tertulis untuk memeriksa apakah aplikasi berjalan seperti yang diharapkan.

Pengujian fungsional meliputi 22 seberapa baik sistem melaksanakan fungsinya, termasuk perintah-perintah pengguna, manipulasi data, pencarian dan proses bisnis, pengguna layar, dan integrasi. Pengujian fumgsional juga meliputi permukaan yang jelas dari jenis fungsi-fungsi, serta operasi back-end (seperti, keamanan dan bagaimana meningkatkan sistem).

1. Pengujian Tegangan (Stress Testing)

Pengujian tegangan berkaitan dengan kualitas aplikasi di dalam lingkungan. Idenya adalah untuk menciptakan sebuah lingkungan yang lebih menuntut aplikasi, tidak seperti saat aplikasi dijalankan pada beban kerja normal. Pengujian ini adalah hal yang paling sulit, cukup komples dilakukan, dan memerlukan upaya bersama dari semua tim.

1. Pengujian Beban (Load Testing)

Pada pengujian beban, aplikasi yang terjadi pada pengujian situs web, untk mengetahui apakah aplikasi/situs gagal atau kinerjanya menurun. Pengujian beban beroperasi pada tingkat beban standar, biasanya beban tertinggi akan diberikan ketika sistem dapat menerima dan tetap berfungsi dengan baik. Perlu diketahui bahwa pengujian beban tidak bertujuan untuk merusak sistem dengan banyak hal, namun mencoba untuk menjaga agar sistem selalu kuat dan berjalan dengan lancar.

1. Pengujian Khusus (Ad-Hoc Testing)

Jenis pengujian ini dilakukan tanpa penciptaan rencana pengujian (test plan) atau kasus pengujian (test case). Pengujian khusus dapat menemukan lubang-lubang dalam pengujian strategi dan dapat mengekspos hubungan di antara subsistem lain yang tidak jelas. Dengan cara ini, pengujian 23 khusus berfungsi sebagai alat untuk memeriksa kelengkapan yang sedang diuji.

1. Pengujian Penyelidikan (Exploratory Testing)

Pengujian penyelidikan mirip dengan penguji khusus dan dilakukan untuk mempelajari/mencari aplikasi. Pengujian penyelidikan perangkat lunak ini merupakan pendekatan yang menyenangkan untuk pengujian.

1. Pengujian Usabilitas (Usability Testing)

Pengujian ini disebut juga sebagai pengujian untuk keakraban pengguna (testing for user-friendlines). Pengujian ini dilakukan jika antarmuka pengguna dari aplikasinya penting dan harus spesifik untuk jenis pengguna tertentu. Pengujian usability adalah proses yang bekerja dengan pengguna akhir secara langsung maupun tidak langsung untuk menilai bagaimana mereka berinteraksi dengannya.

Tujuan dari pengujian usability harus membatasi dan menghilangkan kesulitan bagi pengguna dan untuk memengaruhi area yang kuat untuk usabilitas maksimum.

1. Pengujian Asap (Smoke Testing)

Jenis pengujian ini disebut juga pengujian kenormalan (sanity testing). Pengujian ini dilakukan untuk memeriksa apakah aplikasi tersebut sudah siap untuk pengujian yang lebih besar dan bekerja dengan baik tanpa cela sampai tingkat yang paling diharapkan.

1. Pengujian Pemuliahan (Recovery Testing)

Pengujian pemuliahan (recovery testing) pada dasarnya dilakukan untuk memeriksa seberapa cepat dan baiknya aplikasi bis a pulih terhadap semua 24 jenis crash atau kegagalan hardware, masalah bencana, dan lain-lain. Jenis atau taraf pemulihan ditetapkan dalam persyaratan spesifikasi.

1. Pengujian Volume (Volume Testing)

Pengujian volume dilakukan terhadap efesiensi dari aplikasi. Jumlah data yang besar diproses melalui aplikasi (yang sedang diuji) untuk memeriksa keterbatasan ekstrem dari sistem. Pengujian volume, seperti namanya, adalah pengujian sebuah sistem (baik perangkat keras dan perangkat lunak) untuk serangkaian pengujian dengan volume data yang dapat diproses adalah subjek dari pengujian, seperti sistem yang dapat menangkap sistem pengolahan transaksi penjualan real-time atau dapat membarui basis data atau pengembalian data (data retrival).

1. Pengujian Domain (Domain Testing)

Pengujian domain merupakan penjelasan yang paling sering menjelaskan teknik pengujian. Dugaan dasarnya adalah bahwa anda mengambil ruang pengujian kemungkinan dari variabel individu dan membaginya lagi ke dalam subset (dalam beberapa cara) yang sama. Kemudia, anda menguji perwakilan dari ruang masing-masing subset.

1. Pengujian Skenario (Scenario Testing)

Pengujian skenario adalah pengujian yang realistis, kredibel dan memotivasi stakeholder, tantangan untuk program dan mempermudah penguji untuk melakukan evaluasi. Pengujian ini menyediakan kombinasi variabel-variabel dan fungsi yang sangat berarti dari pada kombinasi buatan yang anda dapatkan dengan pengujian domain atau desain pengujian kombinasi.

1. Pengujian Regresi (Regression Testing)

Pengujian regresi adalah gaya pengujian yang berfokus pada pengujian ulang (retesting) setelah ada perubahan. Pada pengujian regresi berorientasi risiko (risk-oriented regreession

testing), daerah yang sama yang sudah diuji, akan kita uji lagi dengan pengujian yang berbeda (semakin kompleks). Usaha regresi bertujuan untuk mengurangi risiko berikut ini: a. Perubahan yang dimaksudkan untuk memperbaiki bug yang gagal. b. Beberapa perubahan memiliki efek samping, tidak memperbaiki bug lama atau memperkenalkan bug baru.

1. Pengujian Pengguna (User Acceptance)

Pada jenis ini, perangkat lunak akan diserahkan kepada pengguna untuk mengetahui apakah perangkat lunak memenuhi harapan pengguna dan bekerja seperti yang diharapkan. Pada pengembangan perangkat lunak, user acceptance testing (UAT), juga disebut pengujian beta (beta testing), pengujian aplikasi (application testing), dan pengujian pengguna akhir (end user testing) adalah tahapan pengembangan perangkat lunak ketika perangkat lunak diuji pada “dunia nyata” yang dimaksud oleh pengguna. UAT dapat dilakukan dengan in-house testing dengan membayar relawan atau subjek pengujian menggunakan perangkat lunak atau biasanya mendistribusikan perangkat lunak secara gratis untuk diunduh melalui web.

1. Pengujian Alfa (alpha testing)

Pada jenis pengujian ini, pengguna akan diundang ke pusat pengembangan. Pengguna akan menggunakan aplikasi dan pengembangan mencatat setiap masukkan atau tindakan yang dilakukan oleh pengguna. Semua perilaku yang tidak normal dari sistem dan dikoreksi oleh para pengembang.

1. Pengujian Beta (beta testing)

Pada jenis pengujian ini, perangkat lunak mendistribusikan sebagai sebuah versi beta dengan pengguna yang menguji aplikasi di situs mereka. Pengecualian/catat yang terjadi akan dilaporkan kepada

pengembang. Penguji beta dilakukan setelah penguji alfa. Versi perangkat lunak yang dikenal dengan sebutan versi beta dirilis untuk pengguna yang terbatas diluar perusahaan. Perangkat lunak dilepaskan ke kelompok masyarakat agar dapat memastikan bahwa perangkat lunak tersebut memiliki beberapa kesalahan atau bug.

# BAB 3

# RANCANGAN & IMPLEMENTASI

## Gambaran Umum

Pemesanan minuman melalui *barcode* melibatkan pemindaian *QR code* yang terdapat pada meja pelanggan.Dengan menggunakan *smartphone,*pelanggan akan diarahkan ke halaman menu yang berada pada *website*, pelanggan memilih pesanan,dan pelanggan dapat memesan langsung melalui *website* tersebut.Pembayaran dilakukan secara digital,pesanan kemudian dikonfirmasi,diproses, dan diantar ke pelanggan.Sistem ini mempercepat proses pemesanan dan mengurangi kontak fisik.

## Tujuan Pembuatan Website

Tujuan dari *website* pemesanan melalui barcode pada ROOM COFFEE & EATERY:

1. Mendapatkan penghasilan : dengan dibuatnya *website* ini diharapkan mendapat penjualan yang tinggi dan luas, yang akhirnya mendapat penghasilan yang lebih tinggi.
2. Mendapatkan pengunjung : besar pengunjung suatu situs web berbanding lurus dengan uang yang didapatkan.
3. Berkomunikasi : harus mampu memastikan tampilan dan isi web kita dapat dimengerti dengan baik oleh *customer*. Begitu juga cara bekomunikasinya

## Kebutuhan Perangkat

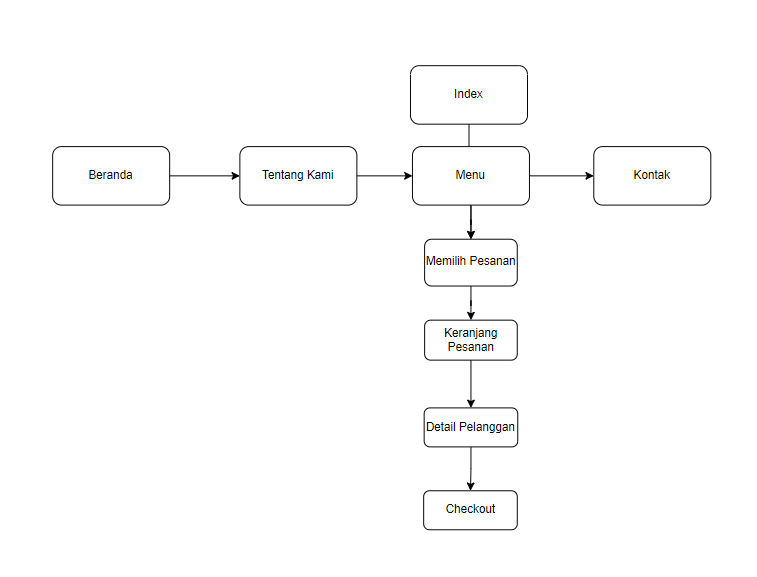
Dalam pembuatan aplikasi *website* ini, menggunakan perangkat lunak (*software*) dan perangkat keras (*hardware*) dengan rincian *hardware* yang digunakan yaitu :

1. *Int*el Pentium Gold
2. *Random Access Memory* (RAM) 4GB
3. *Video Graphic Audio* (VGA) NVIDIA GEFORCE 840M
4. *Harddisk* 500gb

Rincian untuk *software* yang digunakan yaitu :

1. Windows 11 64Bit *Operating System*
2. XAMPP *Control Panel* v3.3.0
3. Sublime *Text* 3/ Visual Studio Code
4. Firefox mozilla

## Struktur Navigasi

Gambar 3.1 berikut adalah struktur navigasi yang terdapat pada *website* pemesanan melalui barcode pada ROOM COFFEE & EATERY. Struktur navigasi termasuk struktur yang paling penting dalam pembuatan suatu aplikasi dan gambaran harus sudah ada pada tahap perencanaan. Peta navigasi merupakan rancangan hubungan dan rantai kerja dari beberapa area yang berbeda dan dapat membantu mengorganisasikan seluruh elemen aplikasi.

Gambar 3. 1Struktur Navigasi

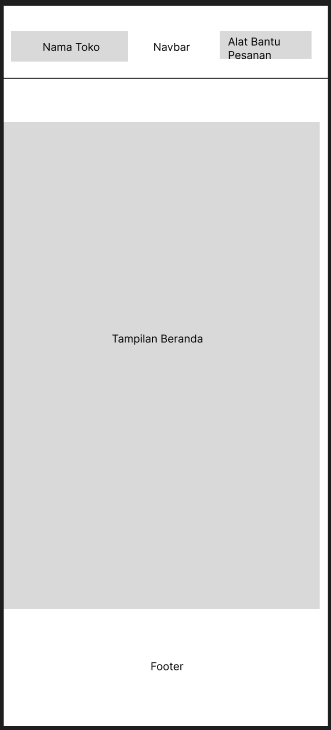
## Use Case Diagram

Gambar 3. 2*Use Case Diagram*

*Use case* di gambar 3.2 menunjukkan bahwa pemilik atau admin dapat mengelola data toko, pelanggan, *delete* data, *view* data, *insert* produk masakan, *view* produk, *delete* produk, dan *update* produk. Sedangkan pelanggan dapat melihat produk, menambah produk ke keranjang, dan melakukan order produk yang telah dimasukkan ke dalam keranjang.

## Tahapan Perancangan Tatap Muka

## Rancangan Halaman Home

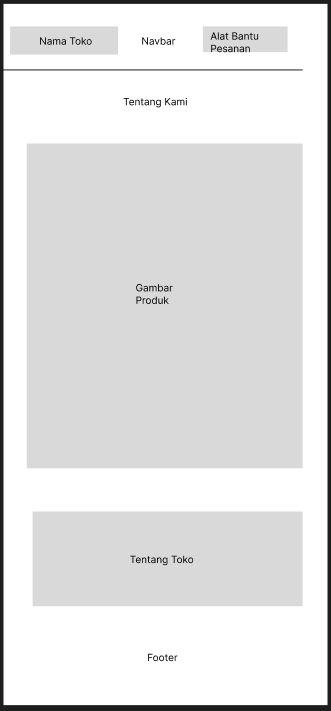
Antarmuka sistem diperlukan untuk memudahkan *user* dalam mengakses *website* atau sistem. Antarmuka menyediakan tampilan halaman yang digunakan untuk proses input hingga menghasilkan ouput yang sesuai dengan kebutuhan. Rancangan halaman *home* dapat dilihat di gambar 3.3 .

Gambar 3. 3 Rancangan Tampilan Home

## Rancangan Halaman Tentang Kami

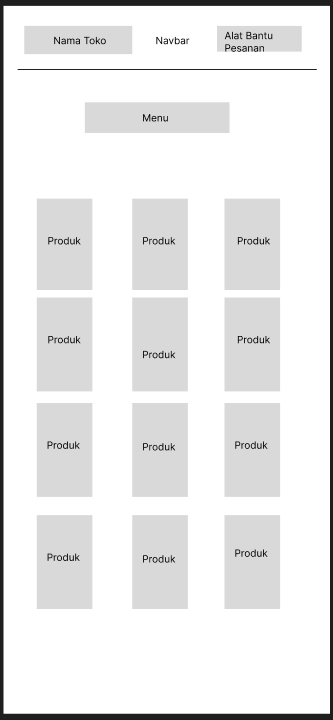
|  |
| --- |
|  |
|  |  |

Tampilan halaman pada gambar 3.4 Tentang Kami di gambar mendeskripsikan menu kopi *signature* yang khas di cafe tersebut.

****

Gambar 3. 4 Rancangan Tampilan Home

## Rancangan Halaman Menu

Tampilan Menu pada gambar 3.5 pelanggan untuk memilih pesanan apa yang ingin di pesan.

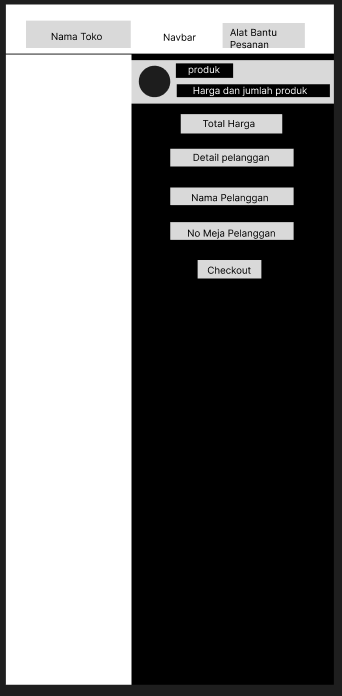
Gambar 3. 5 Rancangan Menu.

## Rancangan Halaman Kontak

Tampilan halaman kontak pada gambar 3.6 digunakan agar pelanggan dapat melihat posisi dimana tempat dan membagikan kepada teman atau kerabat pelanggan.

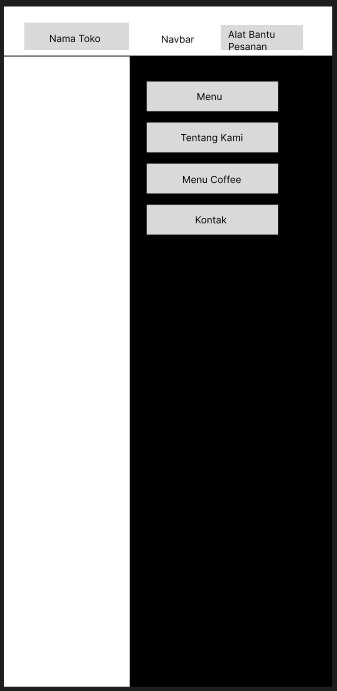
Gambar 3. 6 Rancangan Kontak.

## Rancangan Halaman Keranjang (*cart)*

Halaman Keranjang di gambar 3.7 untuk melihat pesanan,jumlah pesanan,total harga.Disini juga terdapat detail pelanggan untuk mesukan nama dan nomer meja.

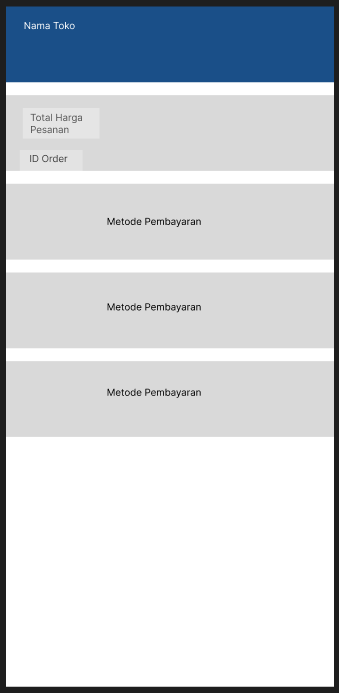
Gambar 3. 7Rancangan Keranjang

## Rancangan Halaman *Hambuerger Menu*

** Tampilan halaman *Hamburger Menu* pada gambar 3.8 ini digunakan untuk mempermudah pelanggan untuk mencari halaman yang langsung ingin dilihat

Gambar 3. 8Rancangan *Hamburger Menu*.

## Rancangan Halaman Admin Add Product

Tampilan halaman pada gambar 3.9 ini digunakan untuk pemyaran disni pelanggan dapat memilih metode pembayaran yang ingin digunakan, seperti gambar

Gambar 3. 9Rancangan Pembayaran.

## Tahapan Implementasi Antarmuka

Pada tahapan ini akan dilakukan implementasi pembuatan *website* berdasarkan rancangan yang telah dibuat sebelumnya, terdiri dari 7 halaman yaitu:

1. Pembuatan halaman *home*
2. Pembuatan halaman Tentang Kami
3. Pembuatan halaman Menu
4. Pembuatan halaman Kontak
5. Pembuatan halaman Keranjang
6. Pembuatan halaman *Hamburger Menu*
7. Pembuatan halaman Pembayaran

Pada implementasi pembuatan *website* ini digunakan bahasa pemrogramanHTML, PHP, CSS, JavaScript. Dan juga disertai *software* pendukung XAMPP untuk melakukan implementasi pada server local. Juga digunakan *Visual Studio Code* sebagai *code editor* untuk menulis dan mengedit program. Sebelum memulai implementasi beberapa hal yang harus disiapkan yaitu:

1. Membuat *folder* dengan nama “penjualan” di dalam *folder*

“C:/xampp/htdocs”.

1. Mengunduh dan instalasi aplikasi XAMPP dan *Visual Studio Code.*
2. Menginstalasi *browser firefox*.

## Implementasi Halaman Utama

Berikut ini adalah *source code* yang ada pada halaman home.

!DOCTYPE html>

<html lang="en">

  <head>

    <meta charset="UTF-8" />

    <meta name="viewport" content="width=device-width, initial-scale=1.0" />

    <title>ROOM COFFEE & EATERY</title>

…………



Gambar 3. 10Halaman *Home*

## Implementasi Halaman Tentang Kami

Berikut ini adalah potongan source code yang ada pada halaman Tentang Kami.

<section id="about" class="about">

      <h2>Tentang Kami</h2>

      <div class="row">

        <div class="about-img">

          <img src="img/Tentang-Kami.jpeg.jpg" alt="Tentang Kami" />

        </div>

        <div class="content">

......................



Gambar 3. 11Halaman Tentang Kami

## Implementasi Halaman Menu

Berikut ini adalah potongan *source code* yang ada pada halaman Menu.

<section class="products" id="products" x-data="products">

      <h2>Menu Coffee</h2>

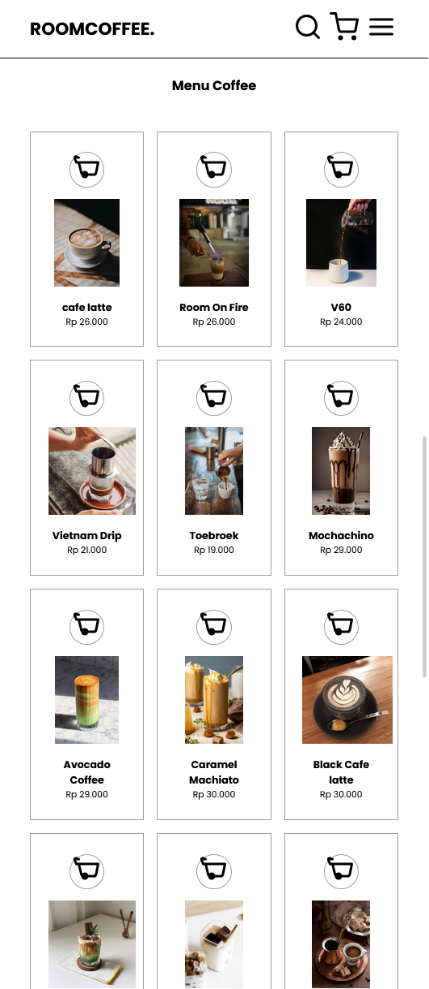
      <div class="row">

          <template x-for="(item, index) in items" :key="index">

              <div class="product-card">

                  <div class="product-icons">

                      <a href="#" @click.prevent="$store.cart.add(item)">



Gambar 3. 12Halaman Menu

## Implementasi Halaman Kontak

Berikut ini adalah potongan *source code* yang ada pada halaman kontak.

<section id="contact" class="contact">

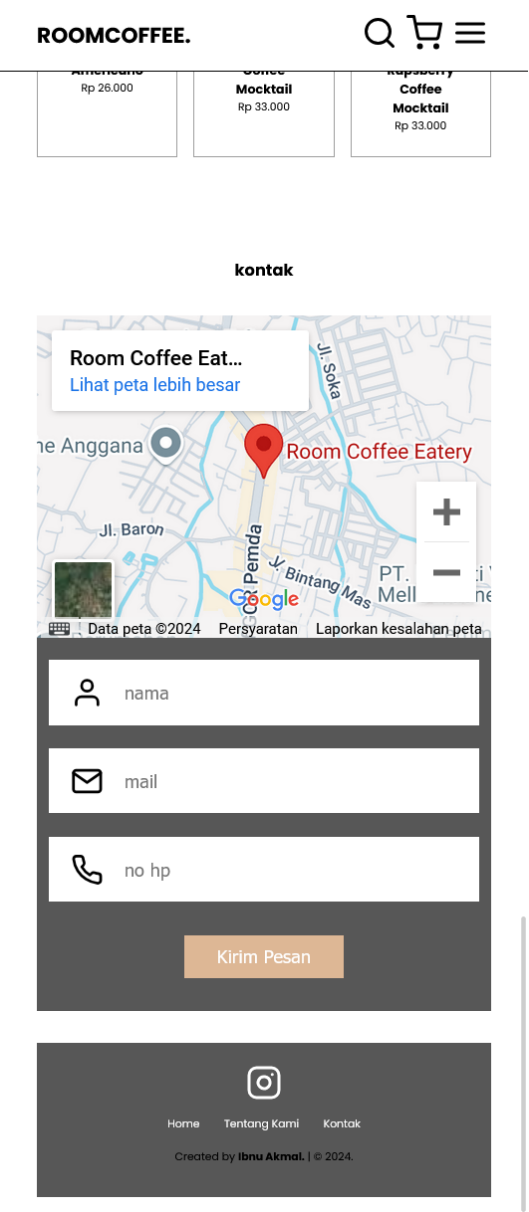
        <section id="menu" class="menu">

            <h2>kontak</h2>

            <p></p>

            <div class="row">

.................

****

Gambar 3. 13Halaman Kontak

## Implementasi Halaman Keranjang (*cart)*

Berikut ini adalah potongan *source code* yang ada pada halaman kontak.

<div class="shopping-cart">

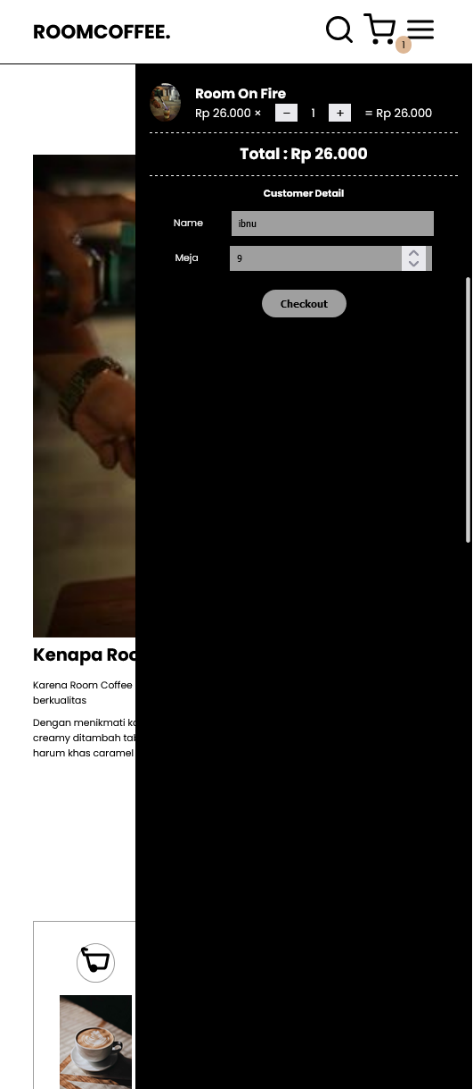
          <template x-for="(item, index) in $store.cart.items" x-key="index">

            <div class="cart-item">

                <img :src="`img/products/${item.img}`" :alt="item.name">

                <div class="item-detail">

.................



Gambar 3. 14Halaman Keranjang

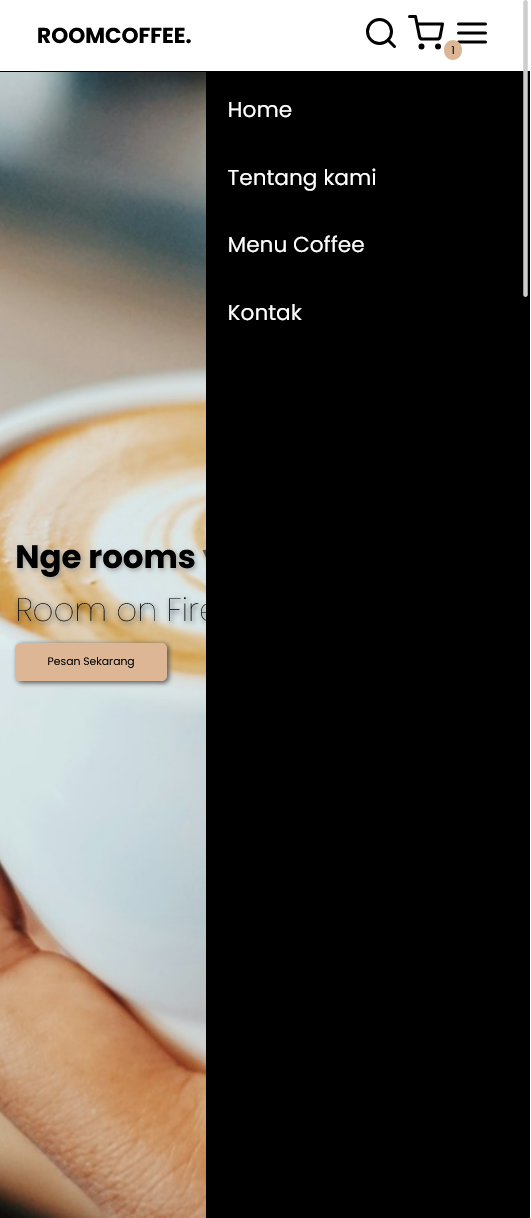
## Implementasi Halaman *Hamburger Menu*

Berikut ini adalah potongan *source code* yang ada pada halaman kontak.

<a href="#" id="hamburger-menu"><i data-feather="menu"></i></a>

      </div>

................



Gambar 3. 15Halaman *Hamburger Menu*

## Implementasi Halaman Pembayaran

Berikut ini adalah potongan *source code* yang ada pada halaman Pembayaran.

checkoutButton.addEventListener("click", async function (e) {

  e.preventDefault();

  const formData = new FormData(form);

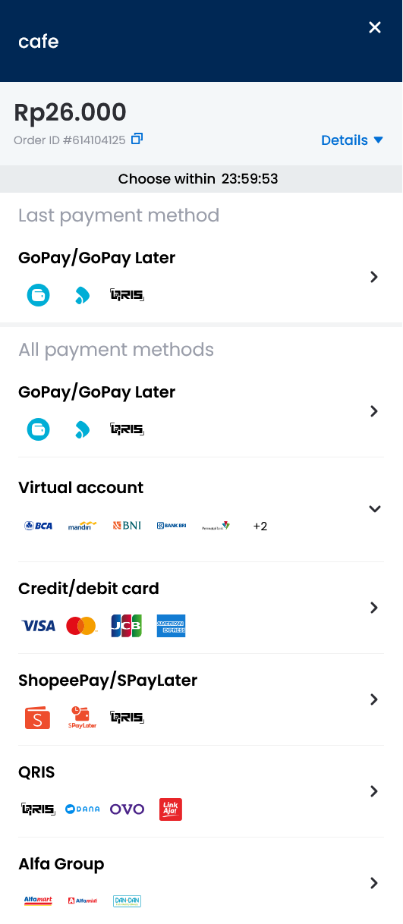
  const data = new URLSearchParams(formData);

  const objData = Object.fromEntries(data);

  console.log(objData);

....................

Untuk melihat program lengkap ada pada bagian Lampiran halaman L-28 dengan nama **add\_product.php.**



Gambar 3. 16Halaman Pembayaran

## Tahap Uji Coba

Pada tahap ini website akan diuji dengan menggunakan blackbox testing dan uji coba user.

### Blackbox Testing

*Blackbox testing* adalah uji coba yang bertujuan agar dapat mengetahui apakah fungsi-fungsi pada *website* dapat berjalan dengan benar sesuai dengan yang diharapkan, seperti pada tabel 3.6 berikut ini.

Tabel 2. 2 Uji Coba Blackbox Testing

|  |  |  |  |
| --- | --- | --- | --- |
| No | Skenario | Hasil yang diharapkan | Hasil yang terjadi |
| 1 | *User* membuka halaman indeks (menu  utama) | Halaman utama dapat terbuka | Halaman utama dapat terbuka |
| 2 | *User* menambahkan  produk ke keranjang | Produk yang dipilih  masuk ke keranjang | Produk yang dipilih  masuk ke keranjang |
| 3 | *User* membuka keranjang belanja | Halaman keranjang  dapat terbuka | Halaman keranjang dapat terbuka |
| 4 | *User* pembayaran berhasil | Data pembayaran berhasil | Data pembayaran berhasil |

## Tahap Uji Coba *User* (UAT)

Pada uji coba kali ini merupakan cara untuk mengevaluasi sebuah aplikasi dengan cara mengujinya kepada calon pengguna. Umumnya, selama pengujian, pengguna akan mencoba untuk menyelesaikan tugas

yang diberikan, sementara pemilik aplikasi akan mengamati, mendengar, dan mencatat hasil dari uji coba yang dilakukan oleh user.

STEP 1. User Menambahkan makanan yang dipilih kedalam keranjang.

STEP 2. User mengurangi, menambah, menghapus makanan yang telah dipilih di dalam keranjang.

STEP 3. User Melakukan Cekout makanan yang dipilih. STEP 4. User mengisi data order berupa identitas & alamat.

Tabel 2. 3 Tabel Uji Coba *User*

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
|  | Step 1 | Step 2 | Step 3 | Step 4 |
| User 1 |  |  |  |  |
| User 2 |  |  |  |  |
| User 3 |  |  |  |  |

 Lancar

 Cukup Lama Tidak Berjalan

# BAB IV PENUTUPAN

## Kesimpulan

Kesimpulan dari penelitian ini adalah *website* pemesanan melalui barcode pada ROOM COFFEE & EATERY ini mempermudah pelanggan dan membanyakan pelanggan,tanpa harus kontakfisik secara langsung dan tanpa ribet.

Dari hasil pengujian yang telah saya lakukan dapat ditarik kesimpulan bahwa *website* pemesanan melalui barcode pada ROOM COFFEE & EATERY ini layak untuk di gunakan oleh masyarakat umum secara masal karena sudah lolos uji kelayakan dengan menggunakan proses Uji Coba User Testing.

## Saran

Penulis menyarankan dalam pengembangan *website* pemesanan melalui barcode pada ROOM COFFEE & EATERY ini lebih lanjut yaitu dengan menambah fitur pelayanan pada website sehingga dapat menjadi lebih baik atau sesuai apa yang dibutuhkan oleh pengguna untuk mengikuti perkembangan yang ada, serta ada umpan balik atau ulasan supaya website yang telah dijalankan dan pelayanan yang akan diadakan selanjutnya dapat menjadi lebih baik.

# DAFTAR PUSTAKA

Abdul Kadir. 2002, *Dasar Pemrograman Web Dinamis Menggunakan PHP*, Andi Offset, Yogyakarta

AL-Bahra Bin Ladjamudin, *Analisis Dan Desain Sistem Informasi*

Supriyanto, A. 2005, Sistem Informasi

Hidayatullah, P. 2014. *Pemrograman Web*. Bandung : Informatika

Kadir, Abdul. 2014. *Pengenalan Sistem Informasi*. Yogyakarta : Andi.

Ladjamudin, Al-Bahra Bin. 2013. *Analisis dan Desain Sistem Informasi*. Yogyakarta: Graha Ilmu.

Nugroho, Bunafit. 2014. *Pemrograman Web :Membuat Sistem Informasi Akademik Sekolah dengan PHP-MySql dan Dreamweaver.* Yogyakarta: Gava Media.

# LAMPIRAN

##### Index.html

<!DOCTYPE html>

<html lang="en">

<head>

<meta charset="UTF-8" />

<meta name="viewport" content="width=device-width, initial-scale=1.0" />

<title>ROOM COFFEE & EATERY</title>

<!-- fonst -->

<link rel="preconnect" href="https://fonts.googleapis.com" />

<link rel="preconnect" href="https://fonts.gstatic.com" crossorigin />

<link

href="https://fonts.googleapis.com/css2?family=Poppins:ital,wght@0,100;0,200;0,400;0,700;1,700&display=swap"

rel="stylesheet"

/>

<!-- feather icons -->

<script src="https://unpkg.com/feather-icons"></script>

<!-- My Style -->

<link rel="stylesheet" href="css/style.css" >

<!-- alpineJS -->

<script defer src="https://cdn.jsdelivr.net/npm/alpinejs@3.x.x/dist/cdn.min.js"></script>

<!-- alpineJS -->

<!-- app -->

<script src="src/app.js" async></script>

<!-- app -->

<!-- midtrans -->

<script type="text/javascript"

src="https://app.sandbox.midtrans.com/snap/snap.js"

data-client-key="SB-Mid-client-wYs2xXmBj1v6XZMF">

</script>

</head>

<body>

**Hamburger Menu.html**

<!-- navbar start -->

<nav class="navbar" x-data>

<a href="#" class="navbar-logo">ROOM<span>COFFEE</span>.</a>

<div class="navbar-nav">

<a href="#home">Home</a>

<a href="#about">Tentang kami</a>

<a href="#products">Menu Coffee</a>

<a href="#contact">Kontak</a>

</div>

<div class="navbar-extra">

<a href="#" id="search-button"><i data-feather="search"></i></a>

<a href="#" id="shopping-cart-button">

<i data-feather="shopping-cart"></i>

<span class="quantity-badge" x-show="$store.cart.quantity" x-text="$store.cart.quantity"></span>

</a>

<a href="#" id="hamburger-menu"><i data-feather="menu"></i></a>

</div>

**Search Box.html**

<!-- SEARCH BOX start -->

<div class="search-form">

<input type="search" id="search-box" placeholder="search here...">

<label for="search-box"><i data-feather="search"></i></label>

</div>

<!-- SEARCH BOX end -->

**Shopping Card.html**

<!-- shopping card start -->

<div class="shopping-cart">

<template x-for="(item, index) in $store.cart.items" x-key="index">

<div class="cart-item">

<img :src="`img/products/${item.img}`" :alt="item.name">

<div class="item-detail">

<h3 x-text="item.name"></h3>

<div class="item-price">

<span x-text="rupiah(item.price)"></span> &times;

<button id="remove" @click="$store.cart.remove(item.id)">&minus;</button>

<span x-text="item.quantity"></span>

<button id="add" @click="$store.cart.add(item)">&plus;</button> &equals;

<span x-text="rupiah(item.total)"></span>

</div>

</div>

</div>

</template>

<h4 x-show="!$store.cart.items.length" style="margin-top: 1rem;">Cart is Empty</h4>

<h4 x-show="$store.cart.items.length">Total : <span x-text="rupiah($store.cart.total)"></span></h4>

<div class="form-container" x-show="$store.cart.items.length">

<form action="" id="checkoutForm">

<input type="hidden" name="items" x-model=" JSON.stringify($store.cart.items)">

<input type="hidden" name="total" x-model="$store.cart.total">

<h5>Customer Detail</h5>

<label for="name">

<span>Name</span>

<input type="text" name="name" id="name">

</label>

<label for="meja">

<span>Meja</span>

<input type="number" name="meja" id="meja" min="1">

</label>

<button class="checkout-button disabled" type="submit" id="checkout-button" value="checkout">Checkout</button>

</form>

</div>

</div>

</nav>

<script>

function validateForm() {

const mejaInput = document.getElementById('meja');

const mejaValue = mejaInput.value;

if (isNaN(mejaValue) || mejaValue <= 0) {

alert('Nomor meja harus berupa angka positif.');

return false;

}

return true;

}

</script>

<!-- shopping card end -->

<!-- navbar end -->

**Home.html**

<!-- hero section start -->

<section class="hero" id="home">

<main class="content">

<h1>Nge rooms <span>yukk</span></h1>

<p>Room on Fire!!!!</p>

<a href="#" class="cta">Pesan Sekarang</a>

</main>

</section>

<!-- hero section end -->

**Tentang Kami.html**

<!-- about section start -->

<section id="about" class="about">

<h2>Tentang Kami</h2>

<div class="row">

<div class="about-img">

<img src="img/Tentang-Kami.jpeg.jpg" alt="Tentang Kami" />

</div>

<div class="content">

<h3>Kenapa Room Coffee?</h3>

<p>

Karena Room Coffee menjadikan pilihan yang tepat untuk menikmati

waktu yang berkualitas

</p>

<p>

Dengan menikmati kopi signature "Room On Fire" kopi susu gula aren

yang sangat creamy ditambah taburan gula aren diatas foam lalu

dibakar dan mengeluarkan harum khas caramel

</p>

</div>

</div>

</section>

<!-- about section end -->

**Menu.html**

<!-- products start -->

<section class="products" id="products" x-data="products">

<h2>Menu Coffee</h2>

<div class="row">

<template x-for="(item, index) in items" :key="index">

<div class="product-card">

<div class="product-icons">

<a href="#" @click.prevent="$store.cart.add(item)">

<svg

width="24"

height="24"

fill="none"

stroke="currentColor"

stroke-width="2"

stroke-linecap="round"

stroke-linejoin="round">

<use xlink:href="#shopping-cart" />

</svg>

</a>

</div>

<div class="product-image">

<img :src="`img/products/${item.img}`" :alt="item.name">

</div>

<div class="product-content">

<h3 x-text="item.name"></h3>

<div class="product-price"><span x-text="rupiah(item.price)"></span></div>

</div>

</div>

</template>

</div>

<!-- SVG symbol definition for the shopping-cart icon -->

<svg style="display: none;">

<symbol id="shopping-cart" viewBox="0 0 24 24">

<path d="M6 6h15l-1.68 8.39a2 2 0 0 1-2 1.61H9.1a2 2 0 0 1-2-1.61L5 6m-2 0h2m-2 0l-1.34-3.34A1 1 0 0 1 3.61 2H5m-2 4h16m-7 10a2 2 0 1 1-4 0m4 0a2 2 0 1 0-4 0"/>

</symbol>

</svg>

</section>

<!-- products end -->

**Kontak.html**

<!-- contact section start-->

<section id="contact" class="contact">

<section id="menu" class="menu">

<h2>kontak</h2>

<p></p>

<div class="row">

<iframe src="https://www.google.com/maps/embed?pb=!1m18!1m12!1m3!1d3964.139317995322!2d106.834348!3d-6.504042199999989!2m3!1f0!2f0!3f0!3m2!1i1024!2i768!4f13.1!3m3!1m2!1s0x2e69c1a5c9aa0915%3A0x8b4d9bca1ee850c!2sRoom%20Coffee%20Eatery!5e0!3m2!1sid!2sid!4v1719665803766!5m2!1sid!2sid" allowfullscreen="" loading="lazy" referrerpolicy="no-referrer-when-downgrade" class="map"></iframe>

<form action=" ">

<div class="input-group">

<i data-feather="user"></i>

<input type="text" placeholder="nama">

</div>

<div class="input-group">

<i data-feather="mail"></i>

<input type="text" placeholder="mail">

</div>

<div class="input-group">

<i data-feather="phone"></i>

<input type="text" placeholder="no hp">

</div>

<button type="sumbit" class="btn">Kirim Pesan</button>

</form>

</div>

</section>

<!-- contact section end-->

Footer.html

<!-- footer star -->

<footer>

<div class="social">

<a href="#"><i data-feather="instagram"></i></a>

</div>

<div class="link">

<a href="#home">Home</a>

<a href="#about">Tentang Kami</a>

<a href="#contact">Kontak</a>

</div>

<div class="credit">

<p>Created by <a href="">Ibnu Akmal.</a> | &copy; 2024.</p>

</div>

</footer>

<!-- footer end -->

<!-- feather icons -->

<script>

feather.replace();

</script>

<!-- my javascrip-->

<script src="js/script.js"></script>

</body>

</html>

##### STYLE CSS

<?php

:root {

  --primary: #000000;

  --bg: #fff;

}

\* {

  margin: 0;

  padding: 0;

  box-sizing: border-box;

  outline: none;

  border: none;

  text-decoration: none;

}

html {

  scroll-behavior: smooth;

}

body {

  font-family: "poppins", sans-serif;

  background-color: var(--bg);

  color: #000000;

  min-height: 3000px;

}

/\*navbar\*/

.navbar {

  display: flex;

  justify-content: space-between;

  align-items: center;

  padding: 1.4rem 7%;

  background-color: #ffffff;

  border-bottom: 1px solid #000000;

  position: fixed;

  top: 0;

  left: 0;

  right: 0;

  z-index: 9999;

}

.navbar .navbar-logo {

  font-size: 2rem;

  font-weight: 700;

  color: #000000;

}

.navbar .navbar-logo span {

  color: var(--primary);

}

.navbar .navbar-nav a {

  color: #000000;

  display: inline-block;

  font-size: 1rem;

  margin: 0 0.8rem;

}

.navbar .navbar-nav a:hover {

  color: rgb(87, 87, 87);

}

.navbar .navbar-nav a::after {

  content: "";

  display: block;

  padding-bottom: 0.5rem;

  border-bottom: 0.1rem solid rgb(87, 87, 87);

  transform: scaleX(0);

  transition: 0.2s linear;

}

.navbar .navbar-nav a:hover::after {

  transform: scaleX(0.5);

}

.navbar .navbar-extra a {

  color: #000000;

  margin: 0 0.3rem;

}

.navbar .navbar-extra a:hover {

  color: rgb(87, 87, 87);

}

#hamburger-menu {

  display: none;

}

/\* navbar search form \*/

.navbar .search-form {

  position: absolute;

  top: 100%;

  right: 7%;

  background-color: #9f9f9f;

  width: 25rem;

  height: 5rem;

  display: flex;

  align-items: center;

  transform: scaleY(0);

  transform-origin: top;

  transition: 0.3s;

}

.navbar .search-form.active {

  transform: scaleY(1);

}

.navbar .search-form input {

  height: 100%;

  width: 100%;

  font-size: 1.2rem;

  color: #000000;

  padding: 1rem;

}

.navbar .search-form label {

  cursor: pointer;

  font-size: 2rem;

  margin-right: 1.5rem;

  margin-left: 1.5rem;

  color: #000000;

}

/\* shopping cart \*/

#shopping-cart-button {

  position: relative;

}

#shopping-cart-button .quantity-badge {

  display: inline-block;

  padding: 1px 5px;

  background-color: #ddb795;

  border-radius: 10px;

  font-size: 0, 5rem;

  position: absolute;

  top: 0;

  right: -10px;

}

.shopping-cart {

  position: absolute;

  top: 100%;

  right: -100%;

  height: 100vh;

  width: 35rem;

  padding: 0 1.5rem;

  background-color: #000000;

  color: #ffffff;

  transition: 0.3s;

}

.shopping-cart.active {

  right: 0;

}

.shopping-cart .cart-item {

  margin: 2rem 0;

  display: flex;

  align-items: center;

  gap: 1.5rem;

  padding-bottom: 1rem;

  border-bottom: 1px dashed;

  position: relative;

}

.shopping-cart img {

  height: 4rem;

  border-radius: 50%;

}

.shopping-cart h3 {

  font-size: 1.4rem;

}

.shopping-cart .item-price {

  font-size: 1.2rem;

}

.shopping-cart .cart-item #add,

.shopping-cart .cart-item #remove {

  display: inline-block;

  padding: 2px 5px;

  cursor: pointer;

  margin: 0 8px;

  background-attachment: white;

  color: black;

  font-weight: bold;

  font-size: 1.1rem;

}

.shopping-cart h4 {

  font-size: 1.6rem;

  margin-top: -1rem;

  text-align: center;

}

/\* form checkout \*/

.form-container {

  width: 100%;

  display: flex;

  justify-content: center;

  border-top: 1px dashed white;

  margin-top: 1rem;

  padding: 1rem;

}

.form-container h5 {

  text-align: center;

  font-size: 1rem;

}

.form-container form {

  width: 100%;

  text-align: center;

}

.form-container label {

  display: flex;

  justify-content: space-around;

  align-items: center;

  margin: 1rem 0;

}

.form-container span {

  text-align: right;

}

.form-container input {

  background-color: #9f9f9f;

  padding: 5px;

  font-size: 1rem;

  width: 70%;

}

.form-container .checkout-button {

  padding: 6px 14px;

  background-color: #ffffff;

  color: #000000;

  font-weight: bold;

  font-size: 1rem;

  border-radius: 20px;

  margin: 1rem auto;

  cursor: pointer;

}

.form-container .checkout-button.disabled {

  background-color: #9f9f9f;

  cursor: not-allowed;

}

/\* navbar search form \*/

/\*hero section \*/

.hero {

  min-height: 100vh;

  display: flex;

  align-items: center;

  background-image: url("../img/header-bg.jpeg.jpg");

  background-repeat: no-repeat;

  background-size: cover;

  background-position: center;

  position: relative;

}

.hero::after {

  content: "";

  display: block;

  position: absolute;

  width: 100%;

  height: 30%;

  bottom: 0;

  background: linear-gradient(0deg, rgb (#ffffff) 20%, rgb(87, 87, 87) 50%);

}

.hero .content {

  padding: 1.4rem;

  max-width: 60rem;

}

.hero .content h1 {

  font-size: 3em;

  color: #000000;

  text-shadow: 1px 1px 3px #9f9f9f;

  line-height: 1.2;

}

.hero .content h1 span {

  color: rgb(87, 87, 87);

}

.hero .content p {

  font-size: 3rem;

  margin-top: 1rem;

  line-height: 1.4;

  font-weight: 100;

  text-shadow: 1px 1px 3px #000000;

}

.hero .content .cta {

  margin-top: 1rem;

  display: inline-block;

  padding: 1rem 3rem;

  font-size: 1, 4rem;

  color: #000000;

  background-color: #ddb795;

  border-radius: 0.5rem;

  box-shadow: 1px 1px 3px rgb(87, 87, 87);

}

/\* About Section \*/

.about,

.products,

.contact {

  padding: 8rem 7% 1.4rem;

}

.about h2,

.products h2,

.contact h2 {

  text-align: center;

  font-size: 2, 6rem;

  margin-bottom: 3rem;

}

.about .row {

  display: flex;

}

.about .row .about-img {

  flex: 1 1 45rem;

}

.about .row .about-img img {

  width: 100%;

}

.about .row .content {

  flex: 1 1 35rem;

  padding: 0 1rem;

}

.about .row .content h3 {

  font-size: 1.8rem;

  margin-bottom: 1rem;

}

.about .row .content p {

  margin-bottom: 0.8rem;

  font-size: 1rem;

  font-weight: 500;

  list-style: 2rem;

}

/\* menu coffee \*/

.products .row {

  display: grid;

  grid-template-columns: repeat(auto-fit, minmax(10rem, 1fr));

  gap: 1.5rem;

  margin-top: 4rem;

}

.products .product-card {

  text-align: center;

  border: 1px solid #9f9f9f;

  padding: 2rem;

}

.products .product-icons {

  display: flex;

  justify-content: center;

  gap: 0.5rem;

}

.products .product-icons a {

  width: 4rem;

  height: 4rem;

  color: #000000;

  margin: 0.3rem;

  border: 1px solid #9f9f9f;

  border-radius: 50%;

  display: flex;

  justify-content: center;

  align-items: center;

}

.products .product-icons a:hover {

  background-color: rgb(178, 178, 178);

  border: 1px solid rgb(187, 187, 187);

}

.products .product-image {

  padding: 1rem 0;

}

.products .product-image img {

  height: 10rem;

}

/\* menu coffee \*/

/\* contact section \*/

.contact .row {

  display: flex;

  margin-top: 2rem;

  background-color: rgb(87, 87, 87);

}

.contact .row .map {

  flex: 1 1 30rem;

  width: 100%;

  object-fit: cover;

}

.contact .row form {

  flex: 1 1 30rem;

  padding: 3rem 1rem;

  text-align: center;

}

.contact .row form .input-group {

  display: flex;

  align-items: center;

  margin-top: 2rem;

  background-color: var(--bg);

  border: 1px solid rgb(87, 87, 87);

  padding-left: 2rem;

}

.contact .row form .input-group input {

  width: 100%;

  padding: 2rem;

  font-size: 1.7rem;

  background: none;

}

.contact .row form .btn {

  margin-top: 3rem;

  display: inline-block;

  padding: 1rem 3rem;

  font-size: 1.7rem;

  color: #fff;

  background-color: #ddb795;

  cursor: pointer;

}

/\* contact section \*/

/\* footer start \*/

footer {

  background-color: rgb(87, 87, 87);

  text-align: center;

  padding: 1rem 0 3rem;

  margin-top: 3rem;

}

footer .social {

  padding: 1rem 0;

}

footer .social a {

  color: #fff;

  margin: 1rem;

}

footer .social a:hover,

footer .link a:hover {

  color: #000000;

}

footer .link {

  margin-bottom: 1.4rem;

}

footer .link a {

  color: #fff;

  padding: 0.7rem 1rem;

}

footer .credit a {

  color: #000000;

  font-weight: 700;

}

/\* footer end \*/

/\*media queries\*/

/\*laptop\*/

@media (max-width: 1366px) {

  html {

    font-size: 75%;

  }

}

/\*tablet\*/

@media (max-width: 758px) {

  html {

    font-size: 62.5%;

  }

  #hamburger-menu {

    display: inline-block;

  }

  .navbar .navbar-nav {

    position: absolute;

    top: 100%;

    right: -100%;

    background-color: #000000;

    width: 30rem;

    height: 100vh;

    transition: 0.3s;

  }

  .navbar .navbar-nav.active {

    right: 0;

  }

  .navbar .navbar-nav a {

    color: #fff;

    display: block;

    margin: 1.5rem;

    padding: 0.5rem;

    font-size: 2rem;

  }

  .navbar .navbar-nav a::after {

    transform-origin: 0 0;

  }

  .navbar .navbar-nav a:hover::after {

    transform: scaleX(0.2);

  }

  .navbar .search-form {

    width: 90%;

    right: 2rem;

  }

  .about .row {

    flex-wrap: wrap;

  }

  .about .row .about-img img {

    height: 50rem;

    object-fit: cover;

    object-position: center;

  }

  .about .row .content {

    padding: 0;

  }

  .about .row.content p {

    font-size: 1.6rem;

  }

  .menu p {

    font-size: 1.2rem;

  }

  .contact .row .map {

    height: 30rem;

  }

}

.contact .row form {

  padding-top: 0;

}

.contact .row {

  flex-wrap: wrap;

}

/\*mobile phone\*/

@media (max-width: 450px) {

  html {

    font-size: 45%;

  }

}

##### App.js

document.addEventListener("alpine:init", () => {

  Alpine.data("products", () => ({

    items: [

      { id: 1, name: "cafe latte", img: "1.jpg.jpg", price: 26000 },

      { id: 2, name: "Room On Fire", img: "5.jpg.jpg", price: 26000 },

      { id: 3, name: "V60", img: "3.jpg.jpg", price: 24000 },

      { id: 4, name: "Vietnam Drip", img: "2.jpg.jpg", price: 21000 },

      { id: 6, name: "Toebroek", img: "4.jpg.jpg", price: 19000 },

      { id: 7, name: "Mochachino", img: "6.jpg.jpg", price: 29000 },

      { id: 8, name: "Avocado Coffee", img: "7.jpg.jpeg", price: 29000 },

      { id: 9, name: "Caramel Machiato", img: "8.jpg.jpeg", price: 30000 },

      { id: 10, name: "Black Cafe latte", img: "9.jpg.jpeg", price: 30000 },

      {

        id: 11,

        name: "Cafe Latte Nusantara",

        img: "10.jpg.jpeg",

        price: 32000,

      },

      {

        id: 12,

        name: "Ice Cube Coffee Milk",

        img: "11.jpg.jpeg",

        price: 32000,

      },

      { id: 13, name: "Turkis", img: "12.jpg.jpeg", price: 21000 },

      { id: 14, name: "Americano", img: "13.jpg.jpeg", price: 26000 },

      { id: 15, name: "Coffee Mocktail", img: "14.jpg.jpeg", price: 33000 },

      {

        id: 16,

        name: "Rapsberry Coffee Mocktail",

        img: "15.jpg.jpeg",

        price: 33000,

      },

    ],

  }));

  Alpine.store("cart", {

    items: [],

    total: 0,

    quantity: 0,

    add(newItem) {

      // cek barang yang sama

      const cartItem = this.items.find((item) => item.id === newItem.id);

      // jika belum ada

      if (!cartItem) {

        this.items.push({ ...newItem, quantity: 1, total: newItem.price });

        this.quantity++;

        this.total += newItem.price;

      } else {

        // cek barang sudah ada apa belum di cart

        this.items = this.items.map((item) => {

          // jika barang berbeda

          if (item.id !== newItem.id) {

            return item;

          } else {

            // jikabarang sudah ada, tambah quantity dan totalnya

            item.quantity++;

            item.total = item.price \* item.quantity;

            this.quantity++;

            this.total += item.price;

            return item;

          }

        });

      }

    },

    remove(id) {

      // item yang bakal di remove berdasarkan id

      const cartItem = this.items.find((item) => item.id === id);

      // jika lebih dari satu

      if (cartItem.quantity > 1) {

        // telusur

        this.items = this.items.map((item) => {

          // jika bukan barang yang di klick

          if (item.id !== id) {

            return item;

          } else {

            item.quantity--;

            item.total = item.price \* item.quantity;

            this.quantity--;

            this.total -= item.price;

            return item;

          }

        });

      } else if (cartItem.quantity === 1) {

        this.items = this.items.filter((item) => item.id !== id);

        this.quantity--;

        this.total -= cartItem.price;

      }

    },

  });

});

// validasi

const checkoutButton = document.querySelector(".checkout-button");

checkoutButton.disabled = true;

const form = document.querySelector("#checkoutForm");

form.addEventListener("keyup", async function () {

  for (let i = 0; i < form.elements.length; i++) {

    if (form.elements[i].value.length !== 0) {

      checkoutButton.classList.remove("disabled");

      checkoutButton.classList.add("disabled");

    } else {

      return false;

    }

  }

  checkoutButton.disabled = false;

  checkoutButton.classList.remove("disabled");

});

// kirim data ketika tombol di click

checkoutButton.addEventListener("click", async function (e) {

  e.preventDefault();

  const formData = new FormData(form);

  const data = new URLSearchParams(formData);

  const objData = Object.fromEntries(data);

  console.log(objData);

  try {

    const response = await fetch("php/placeOrder.php", {

      method: "POST",

      body: data,

    });

    const token = await response.text();

    // console.log(token);

    window.snap.pay(token);

  } catch (err) {

    console.log(err.message);

  }

});

// ajax/token

// rupiah

function rupiah(number) {

  return new Intl.NumberFormat("id-ID", {

    style: "currency",

    currency: "IDR",

    minimumFractionDigits: 0,

  }).format(number);

}