

BAB I

PENDAHULUAN

A. Latar Belakang

Nusantara memiliki warisan budaya dan kuliner yang kaya. Salah satu kekayaan budaya Indonesia yang paling menonjol adalah keragaman kulinernya. Ada banyak sekali makanan nusantara yang lezat dan memikat dari Sabang sampai Merauke. Setiap daerah, suku dan budaya di Indonesia memiliki masakan unik yang mencerminkan keberagaman budaya dan tradisi lokal negara, penggunaan rempah, serta adat istiadat dan sejarah masyarakatnya. Makanan khas daerah di Indonesia merupakan warisan budaya yang sangat dihargai karena sudah ada sejak lama dan masih dibuat hingga sekarang.

Masakan tradisional nusantara menjadi semakin tidak populer di era globalisasi saat ini, karena masakan asing dari negara lain telah menggantikannya.. Sangatlah penting untuk melestarikan dan memperkenalkan keanekaragaman kuliner yang ada di seluruh nusantara kepada generasi berikutnya. Salah satu cara untuk melestarikan dan memperkenalkan kuliner nusantara adalah dengan menyebarkan informasi tentang masakan tersebut kepada masyarakat luas.

Aplikasi ensiklopedia digital masakan nusantara juga menjadi salah satu media yang berguna untuk menyebarkan pengetahuan atau informasi tentang masakan nusantara, mulai dari sejarah, bahan-bahan, cara membuat, hingga penyajiannya. Aplikasi ini juga dapat menjadi sarana untuk mengenalkan kuliner

nusantara kepada generasi muda yang mungkin masih belum mengenal lebih jauh berbagai macam masakan nusantara.

Pada era digital saat ini, aplikasi ensiklopedia masakan nusantara dapat dibuat menggunakan teknologi Aplikasi *Web Progressive (PWA)*. *PWA* merupakan teknologi *web* yang dapat diinstal di perangkat pengguna dan memberikan pengalaman pengguna yang mirip dengan aplikasi *native*. *PWA* memiliki potensi untuk menjadi solusi yang tepat bagi aplikasi ensiklopedia digital masakan nusantara. Dengan pengalaman pengguna yang lebih baik dari *PWA* dapat mendorong minat masyarakat untuk menemukan dan melestarikan tradisi kuliner nusantara.

Dalam konteks ini, membuat sebuah ensiklopedia digital masakan nusantara menggunakan teknologi *progressive web apps* bisa menjadi pilihan yang tepat. Ensiklopedia digital ini akan dapat diakses oleh pengguna melalui berbagai perangkat dan *platform* tanpa perlu menginstal perangkat lunak khusus. Hal ini membuat aplikasi ensiklopedia masakan nusantara dapat memungkinkan pengguna dengan cepat dan mudah mengakses ragam makanan Indonesia, baik dalam segi resep, maupun sejarah makanan khas Indonesia.

Dengan demikian, di rancanglah “**Rancang Bangun Ensiklopedia Digital Masakan Nusantara Menggunakan Teknologi *Progressive web apps***” dimana diharapkan dengan adanya aplikasi ini tidak hanya akan membantu dalam memperkenalkan dan melestarikan warisan kuliner nusantara tetapi juga berkontribusi pada peningkatan kesadaran dan apresiasi terhadap keberagaman budaya Indonesia dan potensi wisata kuliner nusantara.

B. Rumusan Masalah

Dengan latar belakang yang telah dijelaskan, dapat ditarik sebuah rumusan masalah yaitu bagaimana membuat aplikasi ensiklopedia digital masakan nusantara menggunakan teknologi *progressive web apps*.

C. Batasan Masalah

Berdasarkan dari rumusan masalah yang telah dipaparkan diatas, didapatkan batasan masalah dalam penelitian ini adalah sebagai berikut:

1. Aplikasi ini mencakup masakan- khas yang paling populer dan representatif dari setiap daerah nusantara dan bukan masakan modern.
2. Aplikasi ini memiliki minimal 1 sampel ragam makanan khas nusantara untuk setiap provinsi di Indonesia.
3. Aplikasi ini memiliki materi yang berisi teks, gambar, dan menampilkan *video* serta *audio*.

D. Tujuan Penelitian

Tujuan dari penelitian ini adalah membuat aplikasi ensiklopedia digital masakan nusantara menggunakan teknologi *progressive web apps* untuk memberikan kontribusi dalam mengenalkan dan menyebarluaskan informasi yang terfokus pada masakan nusantara.

E. Manfaat Penelitian

Berikut merupakan manfaat yang mampu di kontribusikan oleh penelitian ini adalah sebagai berikut:

1. Bagi akademik

Sebagai bahan referensi bagi penulis lain untuk mengembangkan kemampuan di bidang yang sama.

2. Bagi penulis

Penulis dapat menerapkan ilmu yang didapatkan dalam proses perkuliahan.

3. Bagi masyarakat

Sebagai media pelestarian budaya Indonesia melalui penyebaran informasi tentang masakan nusantara.

BAB II

TINJAUAN PUSTAKA

A. Kajian Teori

1. *Progressive web apps*

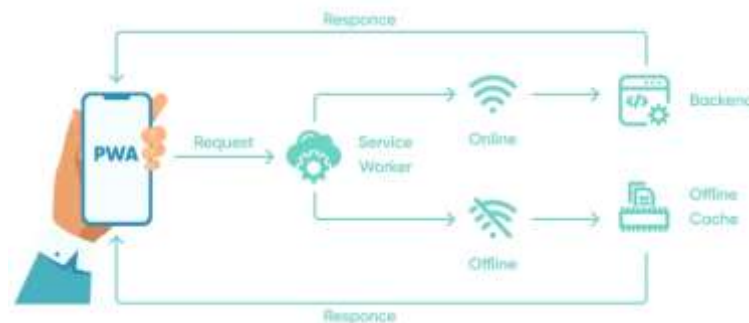


Gambar 2. 1 *Progressive web apps*

Progressive web apps adalah untuk teknologi untuk membuat aplikasi *web* yang memberikan tampilan dan nuansa aplikasi *web* konvensional pada aplikasi asli. Pengembang dapat menyempurnakan aplikasi *web* tradisional dengan semua komponen dan fungsionalitas aplikasi seluler dengan menggunakan aplikasi *web progressive*. Ini memperluas pengalaman pengguna (*UX*) yang luar biasa dan meningkatkan kegunaan. Pengembangan aplikasi ini menciptakan pengalaman *web* dengan fungsionalitas aplikasi. *PWA* adalah aplikasi *web* yang memanfaatkan kemampuan khusus *browser* untuk berperilaku lebih seperti aplikasi *native* atau *mobile app* saat digunakan dengan *browser* yang *compatible* (Wargo, 2020). Sedangkan menurut (Ater, 2019) *PWA* adalah aplikasi *web* generasi baru yang menggabungkan fitur dan sumber daya aplikasi *native* dengan aplikasi *web* standar. *PWA* adalah situs *web* standar yang dibuat dengan teknologi HTML, CSS dan Javascript yang kita kenal dan

sukai sebagai *web developer*, tetapi *PWA* selangkah lebih maju dan memberi pengguna pengalaman yang lebih baik (Alan Hume, 2021). Sebuah *PWA* harus dapat berjalan dengan baik di berbagai perangkat dan ukuran layar, memberikan pengalaman yang mirip dengan aplikasi *native*, selalu mendapatkan pembaruan terbaru, mudah ditemukan oleh pengguna, dapat diinstal di perangkat pengguna, dan dapat dibagikan dengan pengguna lain.

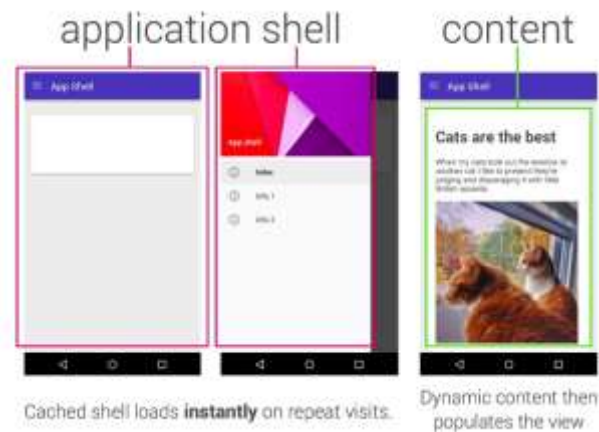
PWA dapat meningkatkan fungsionalitas *web app* yang sudah ada, baik itu dibangun menggunakan *framework* atau murni *JavaScript (Vanilla JS)*. *PWA* dapat diterapkan pada *website* yang dibangun menggunakan React, Angular, atau Vue. Anda bahkan dapat memasukkan fitur ini ke dalam *website* yang sudah aktif tanpa banyak mengubah kode.



Gambar 2. 2 Arsitektur Teknologi *PWA*

Ada beberapa komponen yang harus digunakan saat membangun *PWA*. Beberapa komponen harus digunakan setiap kali membuat *PWA*, sedangkan yang lain adalah opsional. Komponen wajib digunakan setiap kali membuat *PWA*, sedangkan yang opsional adalah komponen yang tidak mempengaruhi kinerja *PWA* tetapi dapat digunakan untuk meningkatkan fiturnya. Berikut komponen wajib yang dimaksud:

a. *Application shell*



Gambar 2. 3 *Application Shell*

Istilah "*Application Shell*" mengacu ke kerangka kerja untuk antarmuka aplikasi yang terdiri dari berbagai elemen halaman dan sumber daya lainnya. Karena sudah disimpan sebelumnya di *cache*, sumber daya dapat langsung dilihat ketika aplikasi diakses.

b. *Web app manifest*

Web app manifest adalah *file* JSON yang mengontrol tampilan sisi pengguna dan pengoperasian aplikasi. File ini biasanya disebut sebagai *manifest.json*. Ketika program pertama kali diluncurkan di *browser*, *browser* akan membaca file manifest, mengunduh sumber daya, dan menampilkan konten.

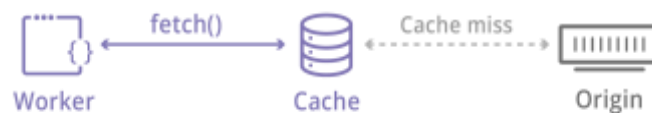
c. *Service worker*



Gambar 2. 4 Cara kerja *Service Worker*

Service worker adalah *file* yang ditulis dalam bahasa javascript yang berjalan di latar belakang *browser* dan terpisah dari *script* lain aplikasi web. Dengan adanya *service worker*, *browser* dapat menampilkan konten bahkan dalam mode jaringan terbatas.

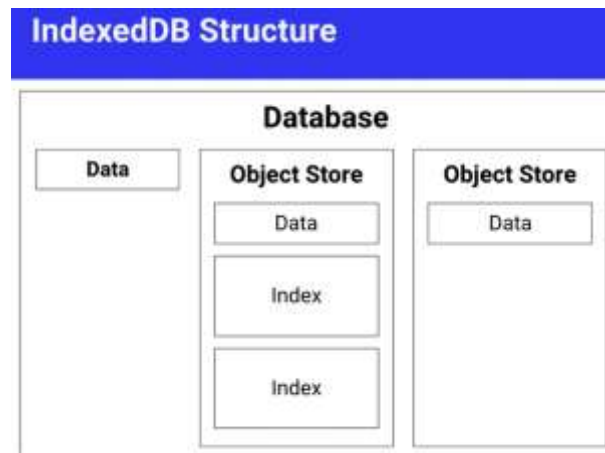
d. *Cache application programming interface*



Gambar 2. 5 *Cache API*

Cache Application Programming Interface (API) adalah ruang penyimpanan lokal yang digunakan *browser* untuk menyimpan sumber daya sehingga dapat digunakan kembali selama panggilan berikutnya tanpa membuat permintaan jaringan tambahan.

e. *IndexedDB*

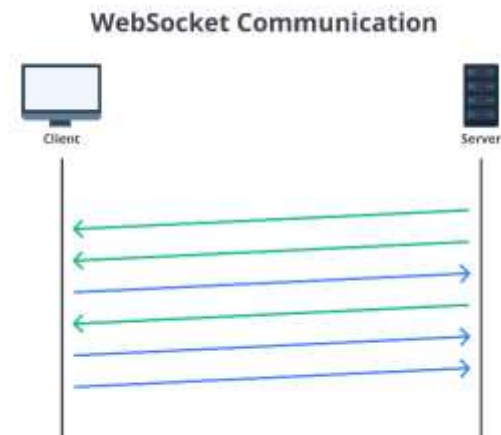


Gambar 2. 6 Struktur *IndexedDB*

Data apa pun dapat disimpan sebagai objek JavaScript di sistem penyimpanan lokal untuk *browser* yang disebut *IndexedDB*, yang berbasis NoSQL. Selain berperan sebagai *cache*, *IndexedDB* juga dapat digunakan untuk menyimpan *item* tertentu.

f. *Web Socket*

Standar baru untuk komunikasi *real-time* di *web* atau aplikasi seluler disebut *Web Socket* yang banyak digunakan saat ini untuk membuat aplikasi obrolan waktu nyata, umpan berita, dan bahkan *game* multipemain daring. *Web Socket* menjadi standar dan komponen HTML5 di dalam *browser* itu sendiri.



Gambar 2. 7 *Web Socket Communication*

Client dan *server* harus selalu terkoneksi agar dapat menerima sinyal komunikasi dari kedua belah pihak saat menggunakan pola komunikasi aktif dua arah (*full-duplex*) seperti ini. Kami dapat membangun komunikasi waktu nyata karena kami selalu terhubung. *Web Socket* dapat digunakan untuk komunikasi waktu nyata, *streaming* langsung, *game* multipemain, dan progres waktu nyata.

g. Notifications

Pesan *pop-up* yang dikenal sebagai notifikasi dapat ditampilkan di *gadget*. Situs *web* yang mendukung *PWA* mendukung tampilan notifikasi. Memberi tahu pengguna tentang informasi apa pun sangat penting. Metode efektif untuk mendorong orang kembali ke situs *web* adalah pemberitahuan.

2. Ensiklopedia digital

Ensiklopedia digital, sering dikenal sebagai ensiklopedia *online* atau ensiklopedia internet, adalah ensiklopedia yang dapat diakses melalui internet. Ini adalah kumpulan pengetahuan terorganisir dengan baik yang memberikan informasi luas tentang topik umum atau bidang tertentu. Ensiklopedia digital dapat diakses melalui berbagai *platform*, termasuk komputer, ponsel, maupun tablet. *Wikipedia*, *Encyclopaedia Britannica*, dan *Columbia Encyclopedia* adalah contoh ensiklopedia digital. Ensiklopedia adalah kumpulan referensi yang berisi informasi ilmiah yang mendasar dan komprehensif. Ada deskripsi artikel yang sederhana dan terpisah di dalamnya, serta deskripsi lengkap. Biasanya digunakan untuk menanggapi pertanyaan informasi umum, peristiwa, konsep, dan fakta. Berisi informasi tentang berbagai cabang ilmu pengetahuan atau mata pelajaran tertentu. Sebagai pernyataan pertanggungjawaban, nama penulis sering ditempatkan di akhir setiap artikel. Di akhir artikel/ensiklopedia, daftar bacaan yang direkomendasikan sering ditambahkan. Entri disusun menurut abjad dan sering diperbarui agar lebih teliti dan akurat. Berdasarkan (Citizendium, 2020) Ensiklopedia Digital merupakan Kumpulan pengetahuan terstruktur berbasis Internet yang terinspirasi oleh konsep ensiklopedia cetak lama.

Ensiklopedia (*/énsiklopédia/*) adalah kumpulan tulisan dengan anotasi yang memberikan informasi menyeluruh dan mudah dipahami tentang satu bidang ilmiah atau seluruh bidang ilmiah. Ensiklopedia dikategorikan menurut abjad, berdasarkan volume penerbitan atau genre, dan, tergantung pada volume

informasi yang disertakan, biasanya dicetak sebagai buku berseri. Buku ini dibagi menjadi beberapa bagian artikel, yang masing-masing membahas satu topik.

Ensiklopedia adalah karya referensi yang memberikan pengetahuan dasar dan luas tentang berbagai topik untuk membantu penelitian lebih lanjut. Sedangkan menurut (Yasin, 2020) Ensiklopedia adalah buku referensi atau karya ringkas yang menawarkan ringkasan pengetahuan dari semua disiplin ilmu atau dari subjek tertentu. Ensiklopedia adalah buku yang memiliki artikel individual yang *menggambarkan* berbagai cabang pengetahuan. Kata-kata atau kata kunci ensiklopedia disusun menurut abjad atau secara alfabetis. Dalam ensiklopedia, informasi penting dilengkapi dengan gambar pendukung yang sesuai dengan konteks informasi (Suwandi & Sutarmo, 2021).

Ensiklopedia dibuat untuk tujuan tertentu. Pada dasarnya ensiklopedia memiliki 3 tujuan utama, yaitu:

- a. *Source of answer to fact question*, yaitu sumber informasi untuk pertanyaan yang membutuhkan data dan fakta.
- b. *Source of background information*, yaitu sumber pengetahuan tentang subjek yang menyediakan subjek atau informasi mendasar dan bermanfaat untuk studi lebih lanjut.
- c. *Direction service*, menyediakan pembaca dengan sumber daya lebih lanjut pada subjek yang dibahas. Sumber informasi ini terletak di akhir artikel dan sering dimaksudkan untuk dibaca atau dipelajari.

3. Masakan nusantara

Masakan merupakan campuran bahan dari sumber hewan atau tumbuhan dan dikonsumsi makhluk hidup untuk mendapatkan energi dan nutrisi. Minuman adalah cairan yang dikonsumsi makhluk hidup dan berfungsi sama dengan makanan sebagai sumber energi dan nutrisi.(Ariyani et al., 2020). Masakan Indonesia merupakan jenis tradisi kuliner yang mewakili keragaman budaya yang ada di negara Indonesia. Masakan nusantara pada umumnya beraneka ragam rasa dan berlimpah rempah-rempah khas yang hanya berasal dari daerah masakan tersebut. (Dewi, 2020)

Masakan Indonesia terkenal penuh rempah-rempah dan menggunakan metode kuliner yang sejalan dengan bahan, tradisi, dan adat istiadat setempat. Selain itu, perdagangan dengan negara-negara seperti India, Cina, Timur Tengah, dan Eropa telah mempengaruhi perkembangan seni kuliner di wilayah tersebut. Karena banyaknya tradisi kuliner, setiap lokasi memiliki cita rasa tersendiri. Perdagangan tidak langsung antara Cina dan India melewati Selat Malaka sekitar awal Abad Bersama karena rute perdagangan beralih dari jalur darat (jalur sutra) ke jalur laut. Indonesia berpartisipasi aktif dalam perdagangan sebagai hasilnya. Karena hubungan perdagangan ini, Indonesia dan Cina serta Indonesia dan India memiliki kontak atau hubungan.

Inilah salah satu alasan mengapa budaya India atau Cina masuk ke Indonesia, sehingga menimbulkan pengaruh lintas budaya, khususnya di ranah kuliner. Penggunaan rempah-rempah seperti jintan, ketumbar, jahe, dan kari, yang sering disajikan dengan santan, adalah salah satu pengaruh awal budaya

dan masakan India di Indonesia. Setelah itu, Pengaruh pedagang Arab juga meningkatkan makanan Indonesia, terutama resep daging kambing dan sate. Masakan Arab, terutama kebab, menjadi inspirasi untuk hidangan sate. Pedagang Tionghoa membawa produk makanan dari negara mereka, seperti mie, kedelai, dan beragam jenis sayuran, bersama dengan pedagang Arab. Penjajah Belanda memperkenalkan cita rasa kuliner baru, termasuk lada Meksiko, kacang-kacangan Amerika untuk bumbu sate dan gado-gado, kentang Amerika Selatan, dan singkong Karibia. Selain itu, sayuran lain diimpor ke Indonesia, termasuk jagung, kacang panjang, wortel, dan kembang kol, yang memicu pengembangan sejumlah masakan baru. (Ikorasaki et al., 2022).

Masakan Nusantara merupakan masakan salah satu yang paling dinikmati di dunia, berkat penggunaan rempah-rempah yang begitu lengkap. Ragam makanan khas Indonesia sangat banyak sehingga melihat foto makanannya saja sudah membuat ngiler. Jenis makanan Nusantara dapat dikategorikan makanan berat dan makanan ringan. (Nulhakim & Firmansyah, 2020).

4. *HTML*



Gambar 2. 8 *Hypertext Markup Language*

Hypertext Markup Language atau *HTML*, adalah bahasa *markup* standar industri untuk membangun halaman *web*. Ini menawarkan kumpulan *tag* dan properti yang menentukan pengorganisasian halaman *web*, konten, dan format halaman *web*. Untuk merepresentasikan elemen pada halaman *web*, *HTML* menggunakan struktur hierarkis yang dikenal sebagai *Document Object Model (DOM)*. Teks, foto, tautan, multimedia, dan elemen lainnya semuanya dapat dimuat dalam setiap elemen, yang dibatasi oleh *tag* pembuka dan penutup.

Istilah "*HyperText*" mengacu pada kemampuan *HTML* untuk membuat tautan untuk mudah dan cepat beranjak dari satu situs *web* ke situs *web* lainnya. *HyperText Markup Language*, juga dikenal sebagai *HTML*, adalah bahasa global *World Wide Web*. Seperti yang tersirat dari namanya *HyperText Markup Language*, *HTML* adalah bahasa *markup* bukan bahasa pemrograman (Hartl & Donahoe, 2019).

Dokumen yang dibuat dengan menggunakan bahasa *markup* seperti *HTML* memiliki bagian yang dirender pada hasil akhir serta bagian yang menginstruksikan program rendering cara menginterpretasikan teks yang tersisa. Proses mengubah dokumen teks yang berisi teks *HTML* menjadi, misalnya, representasi visualnya di layar disebut sebagai "*rendering*" (Nematian, 2020).

Perkembangan *HTML* sangat pesat. Dari permulaan *HTML* 1.0 yang sederhana, yang mencakup elemen dasar seperti judul dan paragraf, hingga penambahan tabel dan gambar pada *HTML* 2.0, dan akhirnya ke *frame* dan

dukungan *form* yang disempurnakan pada *HTML* 3.2. Dengan *CSS*, konten dan presentasi sekarang dapat dipisahkan dalam *HTML* 4.01, yang juga meningkatkan komponennya. Dengan elemen semantiknya, dukungan multimedia, bentuk yang lebih baik, fitur aksesibilitas, dan kemampuan *mobile-friendly*, *HTML5* menandai periode transformasi.

Versi *HTML* terbaru, yang dikenal sebagai *HTML5*, digunakan untuk membuat dan mengatur dokumen di internet. Ini mencakup elemen dan fitur baru yang memungkinkan pembuatan halaman *web* yang lebih dinamis dan menarik, termasuk dukungan untuk audio dan video, kanvas untuk grafik dan animasi, dan peningkatan kompatibilitas dengan perangkat seluler.

Fungsionalitas baru yang diperkenalkan oleh *HTML5* memungkinkan pembuatan halaman *web* yang lebih dinamis dan interaktif, yaitu:

- a. Dukungan audio dan video.
- b. Kanvas visual dan animasi.
- c. Dukungan yang ditingkatkan untuk *mobile*..
- d. *Semantic Element* baru termasuk *nav*, *article*, *sections* dan *headers*.
- e. Kontrol untuk form, yaitu tanggal, waktu, *email*, pencarian, dll.
- f. Penyimpanan *web* untuk menyimpan informasi secara lokal.
- g. Fitur *drag* and *drop*.

5. CSS

CSS adalah bahasa yang terus berkembang untuk menggambarkan presentasi konten *web* di layar, printer, sintesis ucapan, pembaca layar, dan jendela obrolan. *CSS* digunakan oleh semua *browser* pada semua ukuran layar

di semua jenis perangkat *IoT*, termasuk ponsel, komputer, video game, televisi, jam tangan, dan konsol otomatis (Meyer & Weyl, 2020). Sedangkan menurut (Duckett, 2022) CSS adalah bahasa pemrograman website yang digunakan untuk menanggulangi tampilan halaman website. (Castro, 2022) menjelaskan bahwa CSS memiliki berbagai properti yang dapat digunakan untuk mengatur tampilan halaman *web*. CSS terdiri dari dua bagian utama, yaitu deklarasi dan selektor. Deklarasi adalah bagian yang menentukan properti dari suatu halaman *web*. Selektor adalah bagian yang menentukan elemen *web* mana yang akan diatur oleh deklarasi. CSS juga memiliki berbagai properti yang dapat digunakan untuk mengatur tampilan halaman *web*. Beberapa properti CSS yang umum digunakan antara lain:

- a. Ukuran: properti ini digunakan untuk mengatur ukuran halaman *web*, seperti ukuran font, ukuran gambar, dan ukuran jarak antar elemen.
- b. Warna: properti ini digunakan untuk mengatur warna halaman *web*, seperti warna font, warna latar belakang, dan warna *border*.
- c. Font: properti ini digunakan untuk mengatur font elemen *web*, seperti jenis font, ukuran font, dan gaya font.
- d. Tata letak: properti ini digunakan untuk mengatur tata letak elemen *web*, seperti posisi elemen, ukuran elemen, dan jarak antar elemen.

6. *PHP*



Gambar 2. 9 *Hypertext Preprocessor*

PHP adalah bahasa sisi server yang secara khusus digunakan untuk aplikasi website. Bahasa *PHP* berjalan di sisi *server*, artinya ia berjalan pada *server* dan bisa dimasukkan di antara bahasa *HTML* (*HyperText Markup Language*). Oleh karena itu, sintaks *PHP* tidak akan terlihat lagi dan hasil akhir dalam bentuk *HTML* akan dikirimkan ke *browser* (Sutarman, 2020). Karena *PHP* dijalankan di komputer *server*, itu dikenal sebagai bahasa pemrograman sisi *server*. Ini berbeda dengan bahasa pemrograman pada sisi klien, seperti JavaScript, yang beroperasi pada klien, yaitu *web browser*.

PHP merupakan singkatan dari *Personal Home Page*. PHP mendapatkan julukannya karena berguna untuk membangun halaman web pribadi. PHP menjadi bahasa pemrograman web yang sangat kuat dari waktu ke waktu, digunakan tidak hanya untuk membangun halaman website yang sederhana melainkan juga untuk situs website terkenal dengan jutaan pengunjung, seperti Wikipedia, WordPress, Joomla, dan lainnya.

7. Javascript



Gambar 2. 10 Javascript

JavaScript adalah bahasa pemrograman *open-source* untuk teknologi *web* yang dapat digunakan untuk memodifikasi *HTML* dan *CSS*. Biasanya disingkat menjadi *JS* saja. Sebagian besar *browser web*, termasuk *Internet Explorer*, *Google Chrome*, dan *Mozilla Firefox*, menggunakan *JavaScript* untuk operasi yang dilakukan di antarmuka pengguna. Karena kemampuannya untuk menginstruksikan *browser* untuk melakukan tugas, sekarang *JavaScript* menjadi *client-side scripting language* yang paling populer (Shute, 2020).

JavaScript dapat meningkatkan kegunaan halaman *web* dengan memberikan umpan balik yang cepat. Dengan bantuan bahasa pemrograman *JavaScript*, Anda dapat meningkatkan *HTML* Anda dengan menambahkan animasi, interaksi, dan efek visual yang dinamis untuk pengguna *browser*. (McFarland, 2021). Selain itu, *JavaScript* dapat digunakan pada *platform* apa pun dan merupakan alat yang berguna untuk mengembangkan *program* dan makro sementara. Anda dapat dengan cepat membuat perangkat lunak yang layak dengan menggunakan *text editor* dan *browser web* berkat ketersediaan *browser* yang tersebar luas (Patrick, 2019).

Saat pertama kali keluar, *JavaScript* hanya memiliki sedikit "bagian yang bagus". Saat ini, sebagian besar komponen sangat baik. Meskipun masih ada beberapa fitur lawas, fitur baru yang ditawarkan oleh standar *ES6/7/8* membantu Anda menulis kode yang lebih aman, mudah, dan singkat (Subramaniam, 2019) Kode JavaScript mengalami perubahan yang cukup signifikan saat *ES6* atau *ECMAScript 6* diterbitkan. Sekarang Anda akan dapat dengan cepat membedakan antara kode yang ditulis menggunakan sintaks sebelum *ES6* diterbitkan dan kode yang ditulis menggunakan sintaks *ES6* karena perubahannya sangat signifikan. *Javascript* modern saat ini sangat menyenangkan untuk ditulis. Saya telah lama menggeluti dunia pemrograman, tetapi saya akui bahwa kadang-kadang saya merasa sintaks sebelumnya canggung dan sulit dijelaskan daripada sintaks Javascript modern ini yang lebih unggul (Morgan, 2019).

Tidak seperti *ES5*, *ES6* menambahkan lebih dari sekadar sejumlah kecil *API* baru ke dalam bahasa. Ini mencakup banyak struktur sintaksis baru, beberapa di antaranya mungkin memerlukan waktu untuk membiasakan diri. Selain itu, ada banyak *organization form* baru dan dukungan *API* tambahan untuk berbagai jenis data. *ES6* adalah kemajuan yang signifikan untuk bahasa ini. *ES6* penuh dengan fitur baru yang belum Anda ketahui (Simpson, 2020).

8. *Framework laravel*



Gambar 2. 11 *Framework Laravel*

Laravel adalah aplikasi kerangka bangun untuk *PHP* yang gratis dan bisa digunakan oleh semua orang dengan kerangka bangun yang menggunakan konsep *Model-View-Controller (MVC)*. *Laravel* adalah kerangka kerja untuk pengembangan *web* yang menampilkan sintaks ekspresif dan elegan. Dengan kode program yang cepat dan terorganisir dengan baik, kerangka kerja ini meningkatkan produktivitas karyawan sekaligus meningkatkan kualitas aplikasi dan menurunkan biaya pengembangan. Hal ini juga membuat proses *development* menjadi lebih mudah dan fleksibel untuk para *developer* tanpa mempertaruhkan fungsionalitas aplikasi.

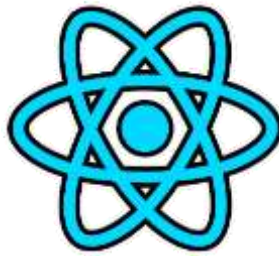
Beberapa fitur dari *framework laravel* termasuk menggunakan *Command Line Interface (CLI) Artisan*, menggunakan manajer paket *PHP Composer*, dan menulis kode yang cepat, bersih, dan mudah ditafsirkan. Informasi di bawah ini tentang *framework Laravel* dapat membantu pengembang mengelola *bug* ketika mereka menghasilkan kode dengan memungkinkan mereka untuk menulis kode sesuai dengan standar pengkodean dan tanpa mengorbankan fungsionalitas aplikasi.

9. *PhpMyAdmin*

Phpmyadmin adalah sebuah aplikasi gratis dan bisa digunakan oleh semua orang untuk dipakai dalam meng-administrasi *database MySQL* menjadi lebih mudah. Dengan *phpmyadmin*, anda dapat dengan mudah membuat *database*, tabel, menyisipkan, menghapus dan memperbarui *data* menggunakan antarmuka grafis, alih-alih mengetik instruksi SQL secara

manual. Selama sebuah OS dapat menjalankan server web dan MySQL, maka OS tersebut dapat menjalankan *PHPMyAdmin* karena berbasis web.

10. React



Gambar 2. 12 React

React.js adalah *javascript library* yang digunakan untuk membuat antarmuka pengguna atau *User interface*. *React.js* dibuat oleh Facebook dan pertama kali dirilis pada tahun 2013. *React.js* bukanlah sebuah *framework* atau kerangka kerja, tetapi *React* dapat digunakan bersama dengan *framework* lain untuk membuat lingkungan kerja yang lebih spesifik, seperti *React Native* untuk membuat aplikasi *mobile*. *React.js* lebih ke pembuatan elemen yang *reusable* yaitu bisa digunakan kembali untuk menampilkan data dinamis. Tujuannya adalah untuk mengurangi cacat dan menghindari masalah dalam pengembangan *User interface* aplikasi. *React* adalah *open-source framework* dengan ribuan komponen, modul, dan dukungan alat.

React adalah *library* yang paling banyak digunakan untuk mengembangkan antarmuka pengguna. *React* Digunakan untuk membangun aplikasi oleh perusahaan dan proyek besar. *React* populer karena faktanya *react* dibangun diatas kemampuan inti pengembangan *web* artinya setiap fungsi akan mudah dipahami dan dipelajari pada percobaan pertama (Morgan, 2021).

11. *Unified modelling language*

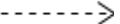
UML adalah kumpulan *support system* yang berguna untuk membuat perangkat lunak dan sistem berbasis objek menjadi lebih sederhana. *UML* juga membantu dalam *development* aplikasi yang berkelanjutan. Sistem atau aplikasi yang tidak didokumentasikan dengan baik seringkali menghambat proses pengembangan karena memaksa para pengembang untuk melakukan penyelidikan dan memahami kode program yang kompleks. *UML* juga berfungsi sebagai alat untuk men-*transfer* pengetahuan tentang suatu sistem atau aplikasi dari satu pengembang ke pengembang lainnya. Keberadaan *UML* tidak hanya mempermudah komunikasi antara para pengembang, tetapi juga memungkinkan orang di luar lingkaran pengembangan, seperti pihak bisnis atau siapapun, untuk memahami suatu sistem dengan lebih baik.

Sistem perangkat lunak direncanakan, didokumentasikan, dan dijelaskan menggunakan *UML (Unified Modeling Language)*, yang juga *UML* merupakan sebuah standar industri. *UML* menawarkan format umum untuk membuat model sistem. *UML* lebih layak diterapkan untuk membangun perangkat lunak yang menggunakan bahasa berbasis objek seperti C++, Java, C#, atau VB.NET, namun demikian, karena kelas dan operasi merupakan bagian dari ide dasarnya. *UML* masih dapat digunakan untuk memodelkan program prosedural yang ditulis dalam C# atau VB..

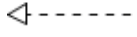
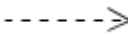
UML mulai dikembangkan oleh *Object Management Group* sejak versi 1.0 dirilis bulan Januari 1997. Beberapa konsep dasar dalam pemrograman

berbasis objek adalah Objek, Kelas, Abstraksi, Enkapsulasi, Pewarisan, dan Polimorfisme. Adapun daftar simbol *UML* yaitu:

Tabel 2. 1 *Symbol Use Case Diagram*

NO.	GAMBAR	NAMA	KETERANGAN
1		<i>Actor</i>	Menggambarkan berbagai peran yang dapat diambil pengguna dalam kasus penggunaan tertentu.
2		<i>Dependency</i>	Hubungan di mana modifikasi pada satu elemen independen berdampak pada elemen independen.
4		<i>Generalization</i>	Memperjelas di mana sumber kasus penggunaan berada.
5		<i>Include</i>	Menunjukkan bahwa pada titik tertentu perilaku <i>use case</i> sumber diperluas oleh <i>use case</i> target.
6		<i>Extend</i>	Apa yang mengikat sebuah item ke item lainnya.
7		<i>Association</i>	Mengidentifikasi paket yang memberikan tampilan terbatas pada sistem
8		<i>Use Case</i>	Hasil terukur dari seorang aktor dihasilkan melalui serangkaian aktivitas yang ditampilkan sistem yang dijelaskan.
9		<i>Collaboration</i>	Cara hukum dan komponen lain digabungkan untuk menghasilkan perilaku yang lebih efektif secara keseluruhan (sinergi).
10		<i>Note</i>	Komponen fisik yang mewakili sumber daya komputasi dan hadir ketika aplikasi dijalankan.

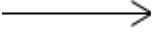
Tabel 2. 2 *Symbol Class Diagram*

NO.	GAMBAR	NAMA	KETERANGAN
1		<i>Generalization</i>	Hubungan di mana perilaku dan struktur data item di atas objek anak (<i>descendent</i>) digunakan bersama oleh objek induk (<i>ancestor</i>).
2		<i>Nary Association</i>	Upaya untuk menghindari hubungan dengan lebih baik dari dua objek.
3		<i>Class</i>	Kelompok objek dengan properti dan fungsi yang identik.
4		<i>Collaboration</i>	Penjelasan tentang serangkaian langkah yang diambil oleh sistem untuk memberikan hasil yang dapat diukur kepada aktor.
5		<i>Realization</i>	Tindakan yang benar-benar dilakukan oleh sebuah objek,
6		<i>Dependency</i>	Hubungan di mana modifikasi pada satu elemen independen berdampak pada elemen dependen
7		<i>Association</i>	Apa yang mengikat suatu objek dengan objek lainnya

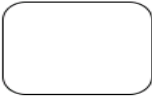
Tabel 2. 3 *Symbol Sequence diagram*

No	GAMBAR	NAMA	KETERANGAN
1		<i>LifeLine</i>	Objek entitas, antarmuka yang mampu melakukan interaksi timbal balik.
2		<i>Message</i>	Deskripsi tentang bagaimana informasi tentang aktivitas yang terjadi dikomunikasikan antar objek.
3		<i>Message</i>	Spesifikasi komunikasi antar objek yang berisi informasi informasi yang merinci tindakan yang terjadi.

Tabel 2. 4 *Symbol State Chart Diagram*

No	GAMBAR	NAMA	KETERANGAN
1		<i>State</i>	Nilai atribut dan nilai Tautan yang dimiliki suatu objek pada saat tertentu.
2		<i>Initial Pseudo State</i>	Bagaimana benda itu berasal atau terbentuk.
3		<i>Final State</i>	Bagaimana sesuatu diciptakan dan di-final kan.
4		<i>Transition</i>	Peristiwa yang mengubah satu atau beberapa nilai atribut objek yang menyebabkan objek berubah statusnya.
5		<i>Association</i>	Apa yang mengikat suatu item dengan item lainnya
6		<i>Node</i>	Komponen fisik yang mewakili sumber daya komputasi dan saat program dijalankan.

Tabel 2. 5 *Symbol Activity diagram*

No.	GAMBAR	NAMA	KETERANGAN
1		<i>Activity</i>	Mendemonstrasikan interaksi antara setiap kelas antarmuka.
2		<i>Action</i>	Status sistem yang merepresentasikan bagaimana suatu aktivitas dilakukan.
3		<i>Initial Node</i>	Bagaimana sesuatu dimulai atau dibuat.
4		<i>Activity Final Node</i>	Bagaimana sesuatu dibuat dan dihapus.
5		<i>Fork Node</i>	Aliran tunggal yang pada akhirnya terpecah menjadi beberapa aliran.

B. Kajian Hasil Penelitian Terdahulu

Untuk dapat digunakan sebagai data pendukung, diperlukan berbagai penelitian sebelumnya. Studi sebelumnya yang berkelindan dengan masalah yang dibahas dalam penelitian ini merupakan salah satu informasi pendukung yang peneliti butuhkan. Dalam hal ini, referensi yang berkaitan dengan tantangan teknologi informasi dari studi sebelumnya digunakan. Oleh karena itu penelitian dilakukan secara *online* dengan menggunakan berbagai temuan penelitian yang disajikan dalam makalah akhir, tesis, atau publikasi.

(Dewi, 2020) dengan judul “Rekayasa Perangkat Lunak Resep Kuliner Nusantara Berbasis *Android*”. Penelitian ini menghasilkan sebuah aplikasi Sistem Informasi Resep Kuliner Nusantara berbasis Android. Menu Provinsi yang terdiri dari 34 (tiga puluh empat) provinsi, bilah pencarian, dan menu kategori adalah 3 (tiga) menu utama yang ditampilkan pada halaman *homescreen* aplikasi. *Android Studio* dan bahasa pemrograman Java digunakan dalam pembuatan aplikasi Sistem Informasi Resep Masakan Nusantara ini.

(Ikorasaki et al., 2022) dengan judul “Rancang Bangun Aplikasi Pengenalan Masakan Nusantara Berbasis *Android*”. Di dalam melakukan pengembangan sistem, penulis menggunakan model waterfall atau siklus hidup perangkat lunak. Aplikasi ini terbagi menjadi 2 (dua) bagian yaitu *admin* dan *User*. Aplikasi ini meliputi resep masakan serta gambaran singkat tentang masakan yang dimaksud. Dibuat menggunakan *Android Studio* dengan bahasa pemrograman Java.

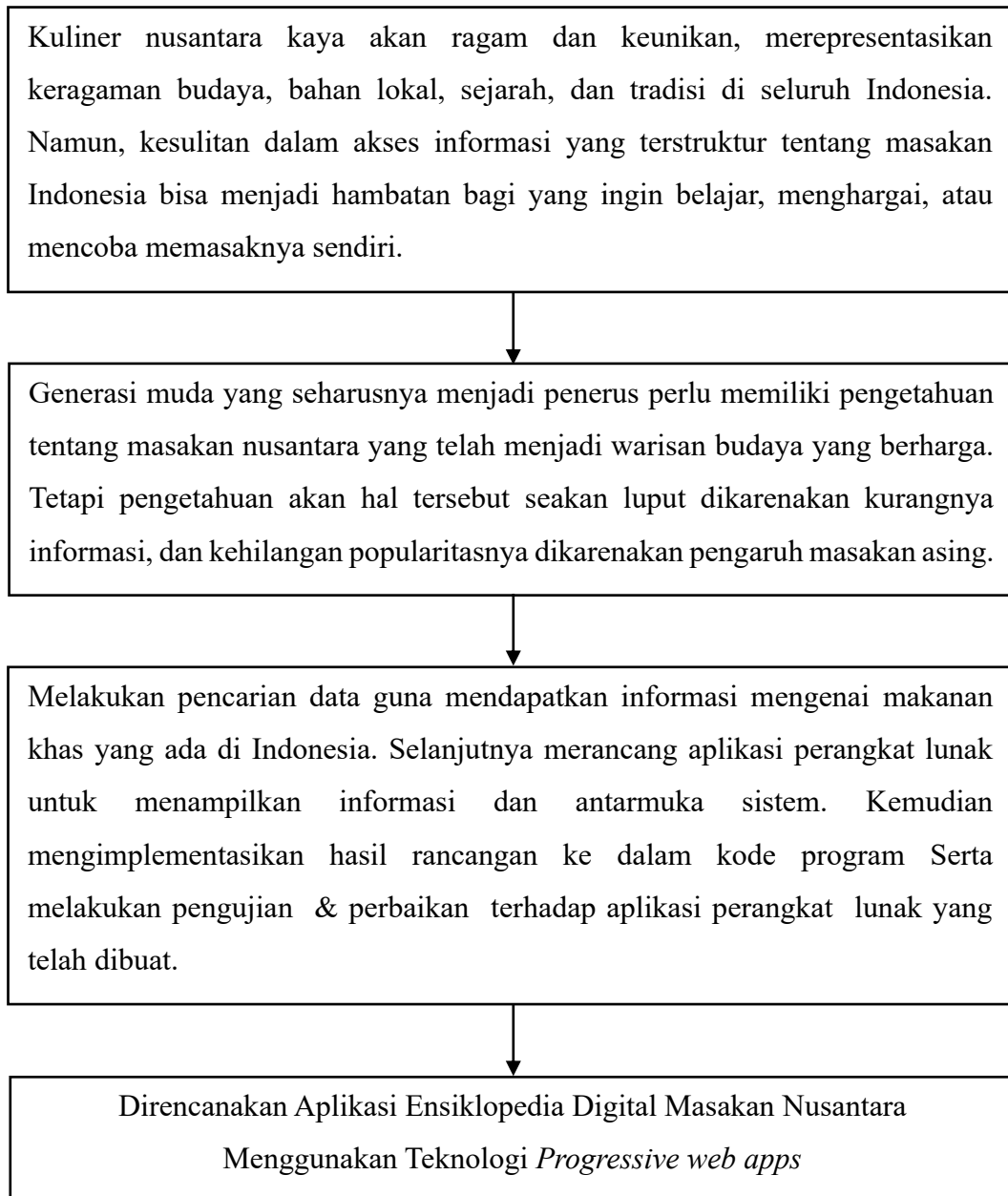
(Hasyim et al., 2022) dengan judul “Pengembangan Konsep *Game* Edukasi Tentang Pengenalan Makanan Tradisional Indonesia kepada Anak-anak Usia Dini”.

Penelitian ini menggunakan pendekatan kualitatif deskriptif, dengan ini dapat dijelaskan pengetahuan umum, jenis-jenis, bahan-bahan, dan pembuatan makanan tradisional. Penelitian ini menghasilkan sebuah *game* yang mengajarkan tentang pengenalan makanan tradisional Indonesia dan berisi berbagai langkah tindakan, termasuk *homescreen*, peta *menu* makanan, *reward*, dan *end*.

(Ariyani et al., 2020) dengan judul “Perancangan Aplikasi Pengenalan Masakan Khas Daerah di Indonesia Berbasis *Android*”. Metode penelitian yang digunakan adalah R&D (*Research and Development*). Hasil dari penelitian ini adalah Aplikasi Pengenalan Makanan Khas daerah di Indonesia. Aplikasi ini dilengkapi dengan tampilan *menu* utama yang menampilkan 7 (tujuh) pulau Indonesia yang berbeda. Setiap pilihan provinsi dalam menu pulau apabila di klik akan memberikan informasi tentang makanan khas daerah tersebut.

Novelty penelitian "Rancang Bangun Aplikasi Ensiklopedia Digital Masakan Nusantara Menggunakan Teknologi *Progressive web apps*" terletak pada penggunaan teknologi *PWA* untuk membangun aplikasi. *PWA* adalah teknologi *web* terbaru yang memungkinkan aplikasi web untuk diinstal di perangkat pengguna dan memiliki pengalaman pengguna yang mirip dengan aplikasi *native*. Hal ini merupakan hal yang baru karena belum ada penelitian sebelumnya yang menggunakan teknologi *PWA* untuk membuat aplikasi ensiklopedia digital masakan nusantara.

C. Kerangka Pikir



BAB III

METODE PENELITIAN

A. Jenis Penelitian

Jenis penelitian yang digunakan adalah penelitian *R&D (Research and Development)*.

B. Lokasi dan Waktu Penelitian

Penelitian dilakukan di Perpustakaan Umum Kota Parepare, Ujung Sabbang, Kecamatan Ujung, Kota Parepare, Sulawesi Selatan 91114. Penelitian ini akan berlangsung selama ± 2 (dua) bulan di tahun 2024.

C. Alat dan Bahan Penelitian

1. Perangkat keras

- a. Laptop Asus ROG Strix G512LI-I75TB6T dengan spesifikasi hardware:
 - *Processor* : Intel® Core™ i5-10300H CPU @ 2.50GHz
 - *Installed RAM* : 8GB
 - *SSD* : 512GB

2. Perangkat lunak

- a. Windows 10 Home Single Language
- b. Visual Studio Code
- c. Laragon
- d. PHP
- e. Javascript

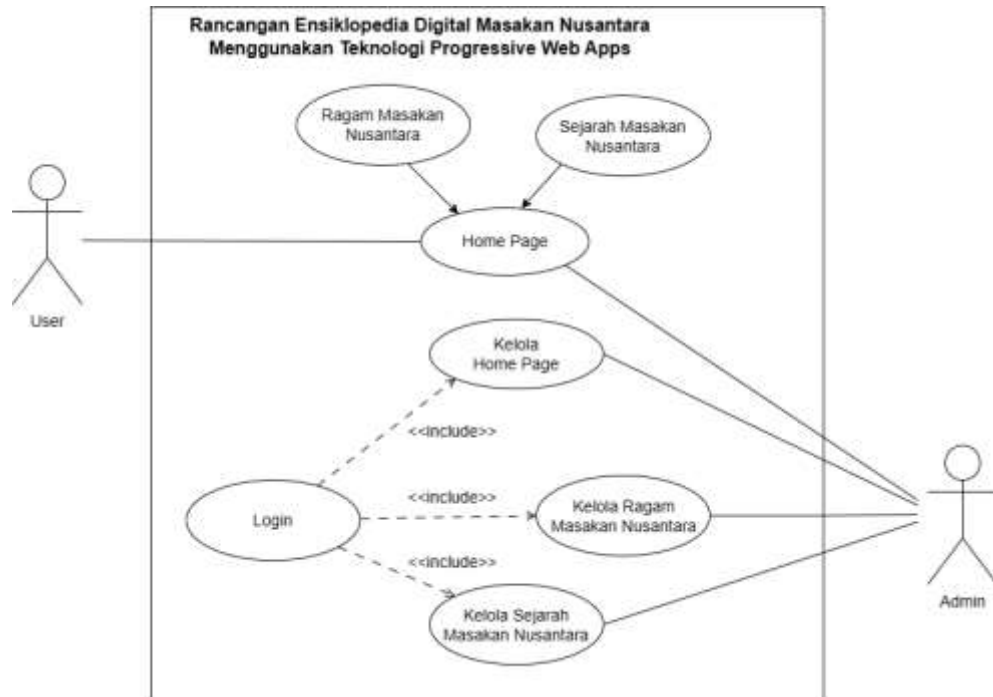
D. Rancangan Sistem

1. Sistem yang berjalan

Pada tahap awal penggunaan sistem yang telah berjalan, proses dimulai dengan pencarian sebuah buku dengan menggunakan kriteria yang telah ditetapkan, seperti judul, penulis, tahun terbit, atau kategori yang relevan. Setelah mengidentifikasi buku yang potensial. Pengguna kemudian melakukan seleksi atau pengambilan buku. Pengguna membuka buku yang telah dipilih dan melakukan pencarian informasi yang spesifik sesuai dengan kebutuhan. Pengguna melakukan pencarian data dengan menggunakan berbagai metode, seperti merujuk pada indeks, daftar isi, atau bagian-bagian khusus dari buku yang diambil.

Setelah melalui proses pencarian yang teliti, pengguna berhasil menemukan data yang sesuai dengan kebutuhan. Data yang ditemukan dapat berupa fakta, temuan, kutipan, atau referensi penting yang merujuk ke masakan nusantara. Sistem mencapai tahap selesai. Ini menandakan bahwa proses pencarian buku, pengambilan buku, pencarian data serta penemuan informasi dari buku telah berhasil diselesaikan.

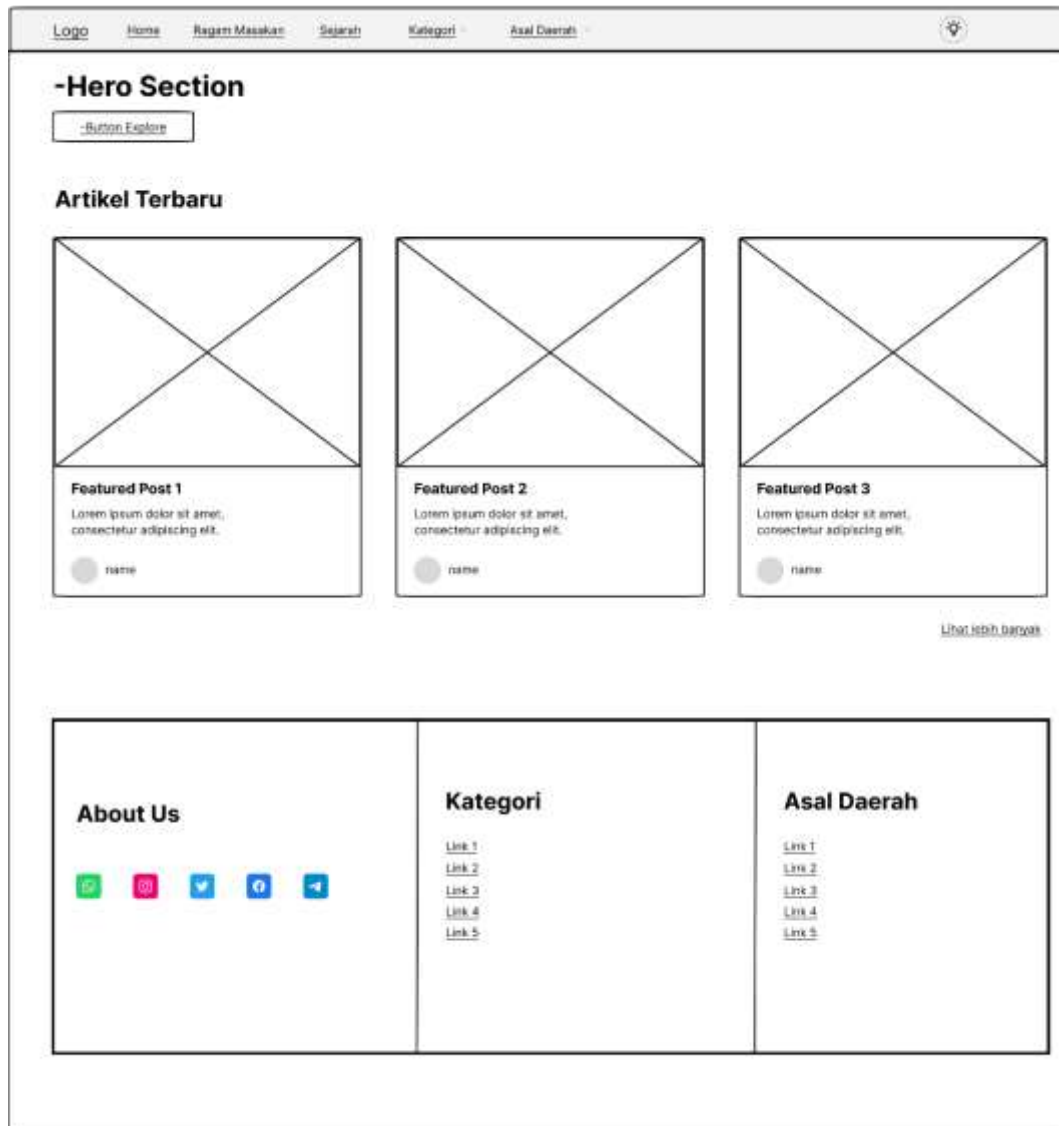
2. Sistem yang diusulkan



Gambar 3. 1 Sistem yang diusulkan

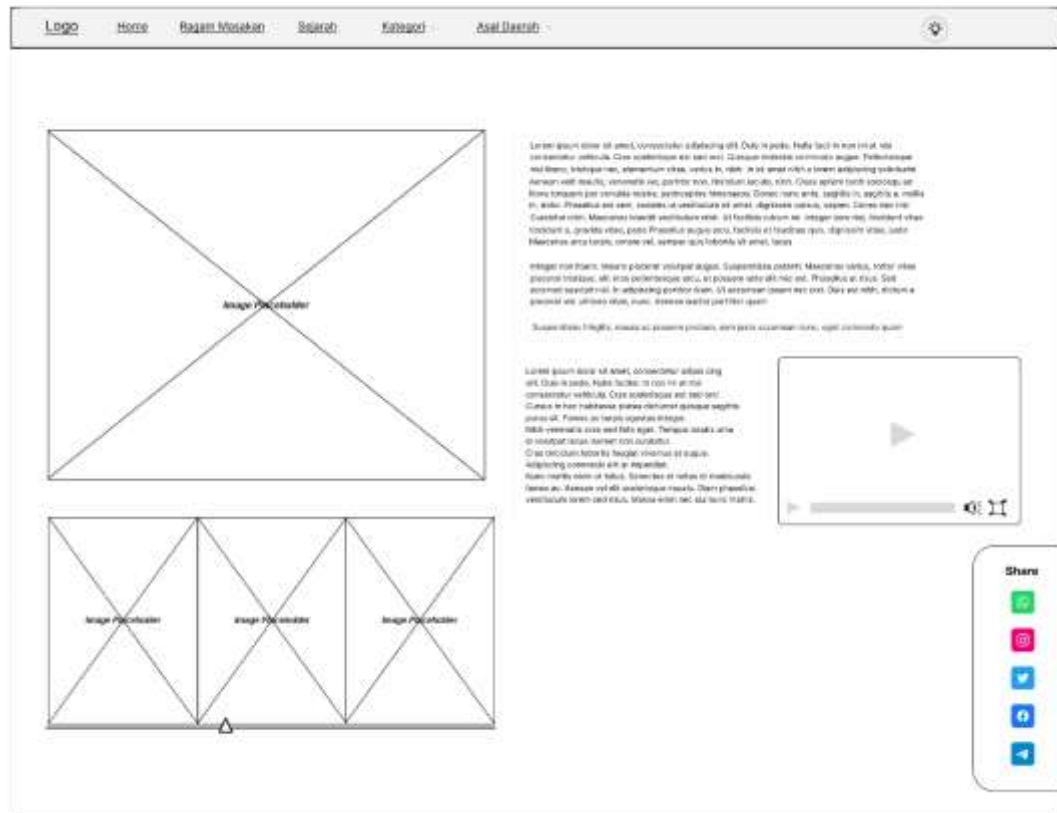
E. Rancangan Output

1. Halaman utama



Gambar 3. 2 Halaman utama

2. Halaman detail



Gambar 3. 3 Halaman detail

F. Teknik Pengumpulan Data

1. Studi literatur

Studi literatur yaitu metode untuk mengumpulkan informasi dari berbagai sumber, termasuk buku, jurnal, buku digital, video youtube, dan situs web yang berkaitan dengan masalah yang relevan.

2. Observasi

Salah satu metode untuk mengumpulkan data yang bisa dilakukan secara langsung ataupun tidak langsung. Observasi yang dilakukan secara tidak langsung dengan cara mengamati objek penelitian melalui media tertentu, seperti video atau foto.

3. Dokumentasi

Data-data terkait dikumpulkan dalam bentuk dokumen. Baik itu dokumen tertulis, elektronik, atau audiovisual.

BAB IV

HASIL DAN PEMBAHASAN

A. Analisis Aliran Data UML

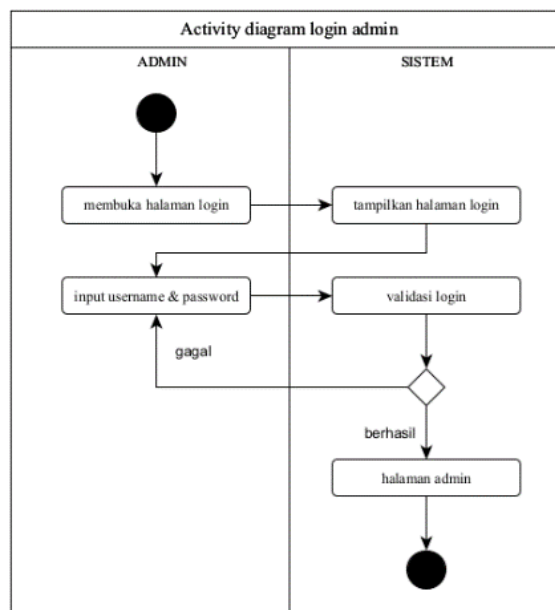
Dalam bagian ini, akan dibahas dua *Diagram UML* yang menggambarkan aliran data dan interaksi dalam aplikasi Ensiklopedia Masakan Nusantara, yaitu *Activity diagram* dan *Sequence diagram*.

1. Activity diagram

Diagram aktivitas berfungsi sebagai alat untuk menggambarkan rangkaian tindakan atau pengaturan alur dalam suatu prosedur organisasi atau skenario penggunaan sistem.

a. Activity diagram admin

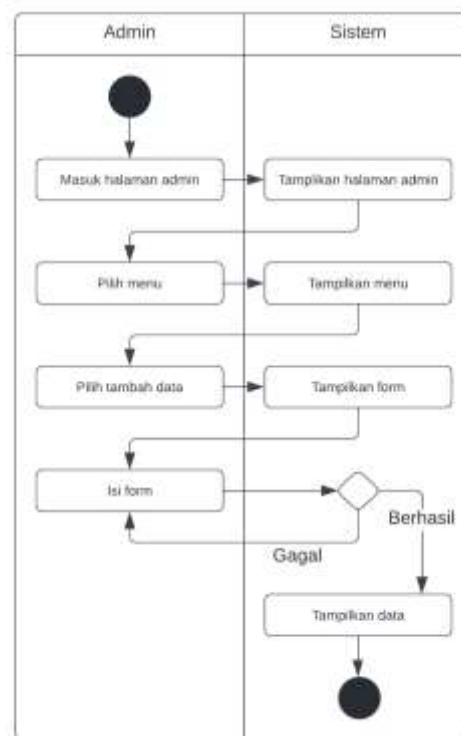
1) Activity diagram login



Gambar 4. 1 Activity diagram login

Pada gambar 4. 1 menjelaskan cara masuk sebagai *admin*. *Admin* harus terlebih dahulu mengakses situs web, setelah itu sistem akan menampilkan *form login* dan *admin* harus memasukkan nama pengguna dan kata sandi. Sistem kemudian akan melakukan validasi; jika informasi benar, maka akan masuk ke halaman *admin*; jika tidak, maka akan kembali ke halaman *login*.

2) Activity diagram tambah data

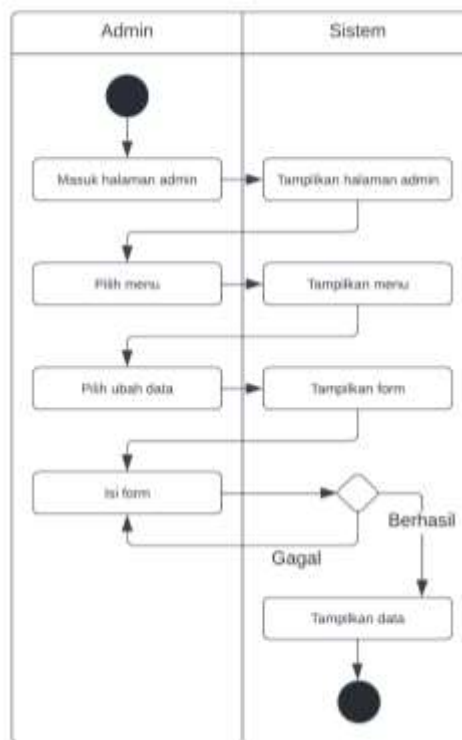


Gambar 4. 2 Activity diagram tambah data

Pada gambar 4. 2 menjelaskan langkah-langkah yang digunakan oleh *admin* untuk menambahkan data. Halaman *admin* ditampilkan oleh sistem ketika telah dibuka oleh *admin*. *Admin* kemudian memilih item menu tambah data, Selanjutnya, antarmuka sistem menampilkan opsi-opsi yang tersedia sesuai pilihan pengelola, kemudian pengelola

mengakses fitur untuk menambahkan informasi baru. Form tambah data kemudian akan ditampilkan oleh sistem. *Admin* selanjutnya diminta untuk melengkapi *form* tambah data. Setelah selesai, sistem akan mengecek data dan jika berhasil, sistem akan menampilkan halaman menu untuk data yang ditambahkan. Jika tidak berhasil, *administrator* harus mengisi formulir penambahan data dengan benar sekali lagi.

3) *Activity diagram* ubah data

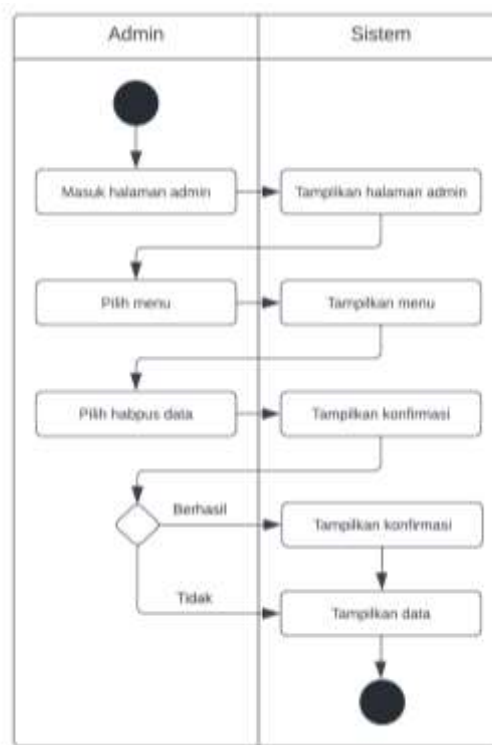


Gambar 4. 3 *Activity diagram* ubah data

Pada gambar 4. 3 menjelaskan prosedur yang perlu diikuti oleh pengelola sistem untuk melakukan perubahan pada informasi yang ada. Hal pertama yang dilakukan *admin* adalah membuat folder *admin*. Setelah itu, tampilan antarmuka akan memperlihatkan berbagai

pilihan fungsi untuk akun pengelola. Berikutnya, pengelola memilih kategori yang datanya hendak diperbarui, sistem akan menampilkan judul menu yang dipilih *admin*; terakhir, *admin* mengubah data. Sistem akan menampilkan data dalam bentuk tabel. Setelah *admin* mengisi *form* data, sistem akan memvalidasi informasi tersebut. Jika validasi berhasil, sistem akan menampilkan item menu dengan data yang dimasukkan, dan jika tidak, *administrator* disarankan untuk mengisi form data kembali dengan akurat.

4) Activity diagram hapus data

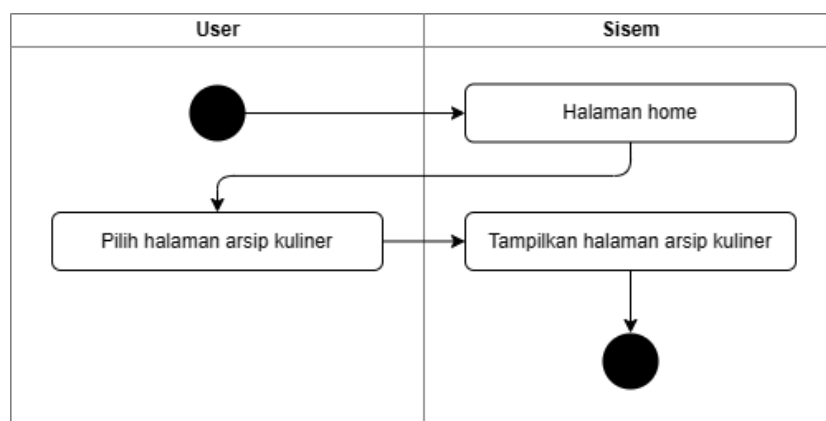


Gambar 4. 4 Activity diagram hapus data

Pada gambar 4. 4 menjelaskan prosedur yang digunakan *admin* untuk menghapus data. Halaman *admin* ditampilkan oleh sistem ketika telah dibuka oleh *admin*. *Admin* kemudian memilih menu yang datanya akan dimusnahkan, sistem menampilkan halaman menu yang telah dipilih oleh *admin*, dan *admin* mengklik pilihan hapus data. Sistem kemudian akan menampilkan konfirmasi penghapusan. *Admin* kemudian melakukan konfirmasi. Jika *admin* konfirmasi, data akan terhapus, jika tidak maka data batal dihapus.

b. Activity diagram User

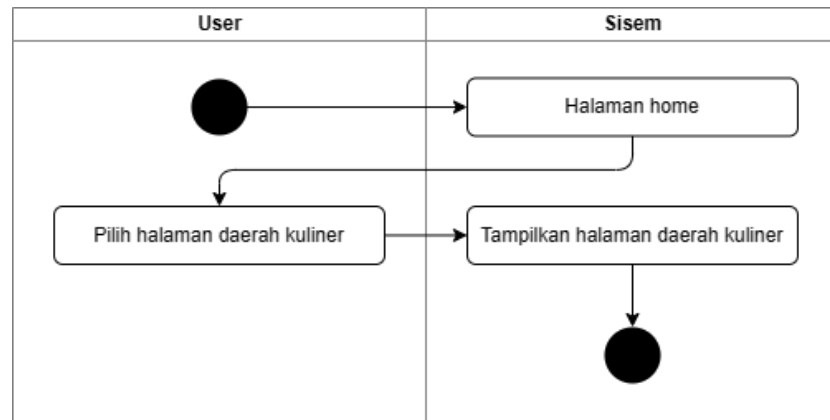
1) Activity diagram pilih arsip kuliner



Gambar 4. 5 Activity diagram pilih arsip kuliner

Pada gambar 4. 5 untuk melihat halaman arsip kuliner, pengguna harus membuka aplikasi terlebih dahulu. Tindakan ini akan meminta sistem untuk menampilkan halaman beranda. Kemudian, setelah pengguna memilih, sistem akan menampilkan halaman arsip kuliner.

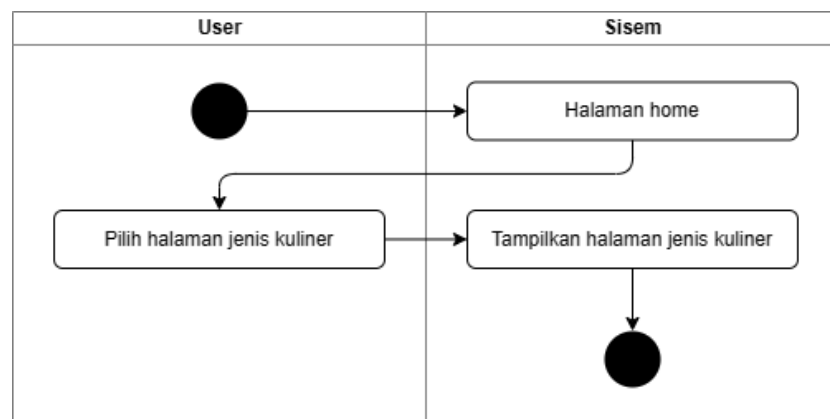
2) Activity diagram pilih daerah kuliner



Gambar 4. 6 Activity diagram pilih daerah kuliner

Pada gambar 4. 6 untuk melihat halaman daerah kuliner, Sistem menampilkan halaman beranda. Setelah pengguna memilih pilihan daerah kuliner, sistem akan menampilkan halaman yang sesuai.

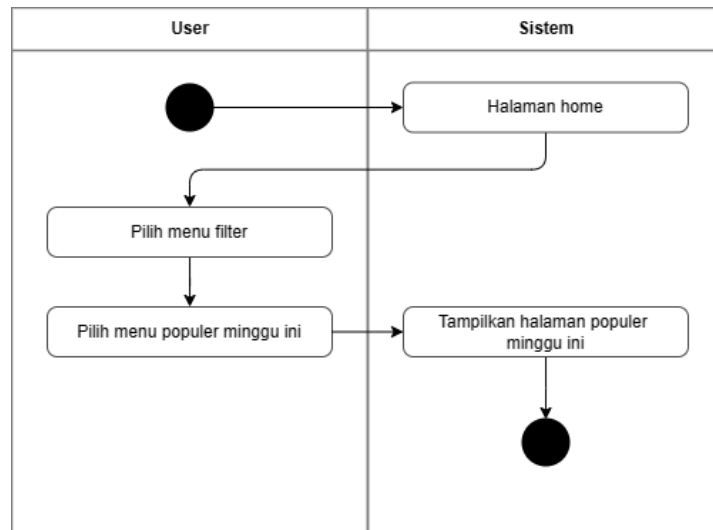
3) Activity diagram jenis kuliner



Gambar 4. 7 Activity diagram pilih jenis kuliner

Pada gambar 4. 7 untuk melihat halaman jenis kuliner, Sistem akan menampilkan halaman beranda. Setelah pengguna memilih pilihan jenis kuliner, sistem akan menampilkan halaman yang sesuai.

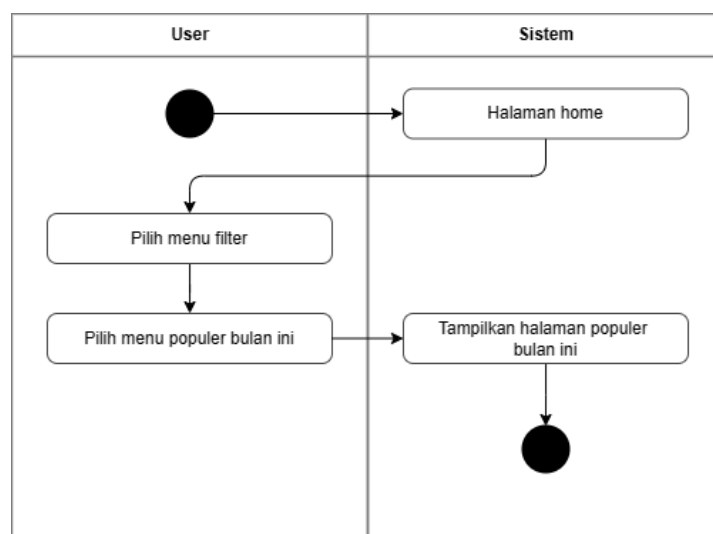
4) Activity diagram filter populer minggu ini



Gambar 4. 8 Activity diagram filter populer minggu ini

Pada gambar 4. 8 Sistem menampilkan halaman beranda. Setelah pengguna memilih pilihan *filter*, sistem akan menampilkan pilihan *filter*. Pengguna memilih *filter* populer minggu ini, sistem akan menampilkan halaman yang sudah di *filter*.

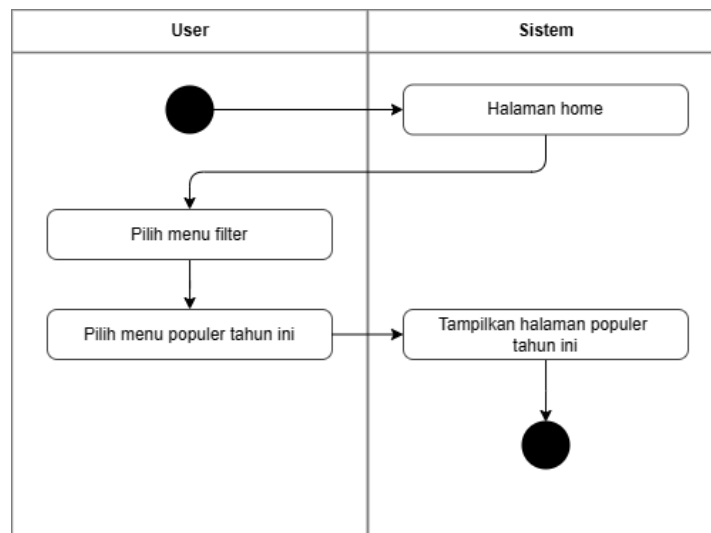
5) Activity diagram filter populer bulan ini



Gambar 4. 9 Activity diagram filter populer bulan ini

Pada gambar 4. 9 untuk melihat halaman kuliner yang sudah di *filter*, Langkah awal bagi pemakai adalah mengakses perangkat lunak. Selanjutnya, antarmuka program akan memunculkan tampilan utama. Setelah pengguna memilih pilihan *filter*, sistem akan menampilkan pilihan *filter*. Pengguna memilih *filter* populer bulan ini, sistem akan menampilkan halaman yang sudah di *filter*.

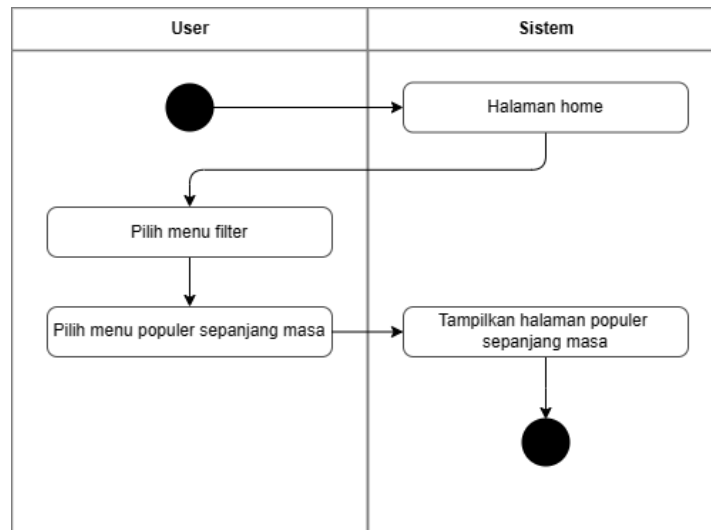
6) Activity diagram filter populer tahun ini



Gambar 4. 10 Activity diagram filter populer tahun ini

Pada gambar 4. 10 Sistem menampilkan halaman beranda. Setelah pengguna memilih pilihan *filter*, sistem akan menampilkan pilihan *filter*. Pengguna memilih *filter* populer tahun ini, sistem akan menampilkan halaman yang sudah di *filter*.

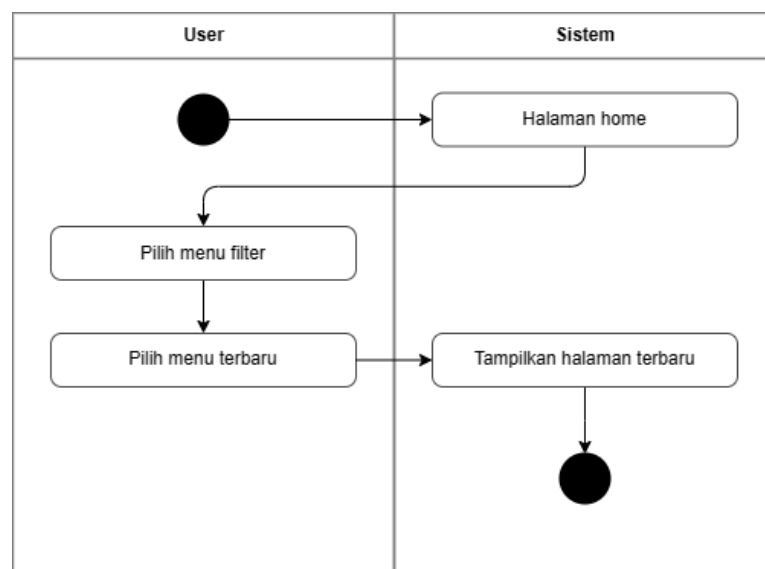
7) Activity diagram *filter* populer sepanjang masa



Gambar 4. 11 Activity diagram *filter* populer sepanjang masa

Pada gambar 4. 11 untuk melihat halaman kuliner yang sudah di *filter*, Sistem menampilkan halaman beranda. Setelah pengguna memilih pilihan *filter*, sistem akan menampilkan pilihan *filter*. Pengguna memilih *filter* populer sepanjang masa,

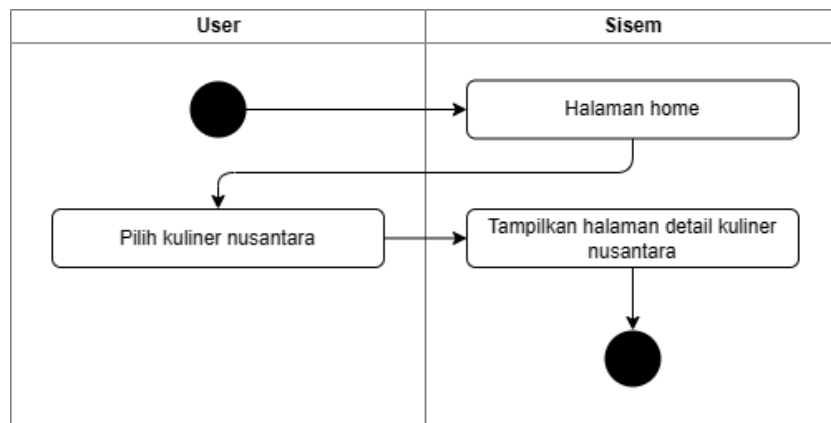
8) Activity diagram *filter* terbaru



Gambar 4. 12 Activity diagram *filter* terbaru

Pada gambar 4. 12 untuk melihat halaman kuliner yang sudah di *filter*, Langkah awal bagi pemakai adalah mengakses perangkat lunak. Selanjutnya, antarmuka program akan memunculkan tampilan utama. Setelah pengguna memilih pilihan *filter*, sistem akan menampilkan pilihan *filter*. Pengguna memilih *filter* terbaru, sistem akan menampilkan halaman yang sudah di *filter*.

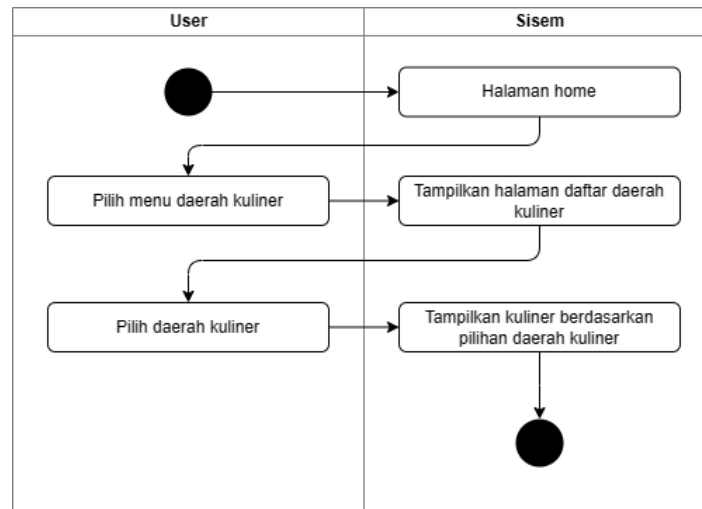
9) Activity diagram halaman detail kuliner



Gambar 4. 13 Activity diagram halaman detail kuliner

Pada gambar 4. 13 untuk melihat halaman detail kuliner, Langkah awal bagi pemakai adalah mengakses perangkat lunak. Selanjutnya, antarmuka program akan memunculkan tampilan utama. Setelah pengguna memilih kuliner, sistem akan menampilkan halaman detail kuliner.

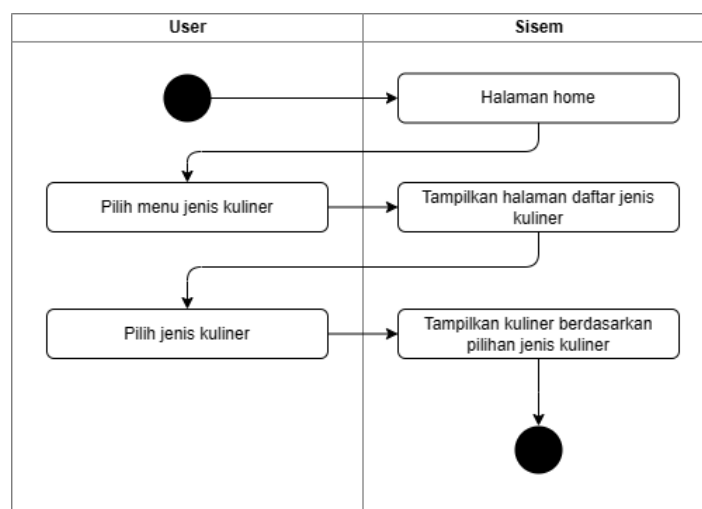
10) Activity diagram pilih daerah kuliner



Gambar 4. 14 Activity diagram pilih daerah kuliner

Pada gambar 4. 14 untuk melihat halaman daerah kuliner, pengguna harus membuka aplikasi terlebih dahulu. Sistem kemudian akan menampilkan halaman beranda. Pengguna akan memilih daerah kuliner lalu sistem menampilkan kuliner berdasarkan pilihan daerah kuliner pengguna.

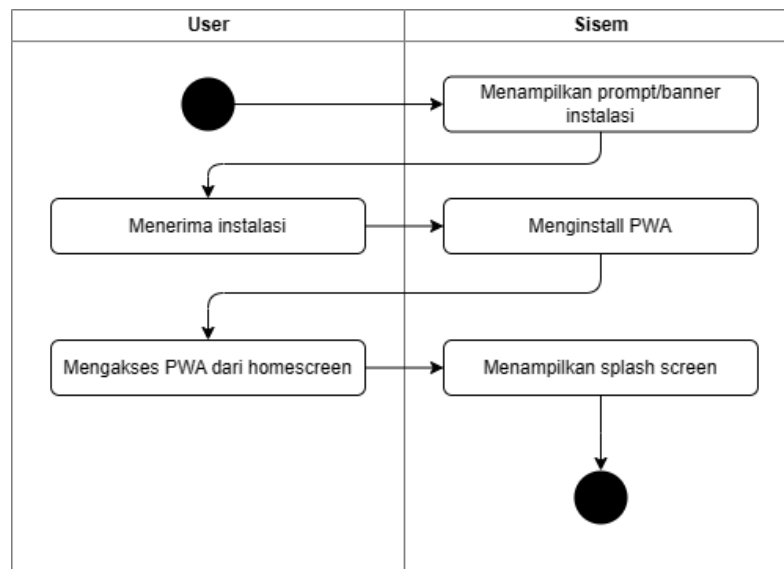
11) Activity diagram pilih jenis kuliner



Gambar 4. 15 Activity diagram pilih jenis kuliner

Pada gambar 4. 15 untuk melihat halaman jenis kuliner, Langkah awal bagi pemakai adalah mengakses perangkat lunak. Selanjutnya, antarmuka program akan memunculkan tampilan utama. Setelah pengguna memilih menu jenis kuliner, sistem akan menampilkan halaman daftar jenis kuliner. Pengguna akan memilih jenis kuliner lalu sistem menampilkan kuliner berdasarkan pilihan jenis kuliner pengguna.

12) Activity diagram instalasi *progressive web apps*



Gambar 4. 16 Activity diagram instalasi *progressive web apps*

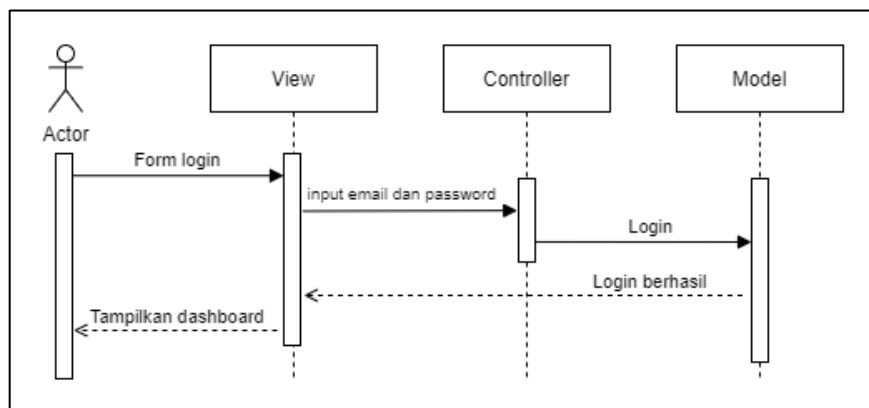
Pada gambar 4. 16 untuk melihat menginstall *progressive web apps*, pengguna harus membuka aplikasi terlebih dahulu. Sistem kemudian akan menampilkan *prompt/banner* instalasi. Pengguna akan menerima instalasi dan sistem akan menjalankan instalasi *progressive web apps*. Pengguna mengakses *progressive web apps* dari *homescreen* perangkat dan sistem menampilkan halaman *slash screen*.

2. Sequence diagram

Bagan alur interaksi atau *Sequence diagram* adalah representasi visual yang memperlihatkan komunikasi antar komponen dalam sebuah sistem secara kronologis. Ilustrasi ini menjabarkan cara berbagai elemen saling bertukar informasi atau data guna menuntaskan suatu aktivitas atau skenario penggunaan tertentu.

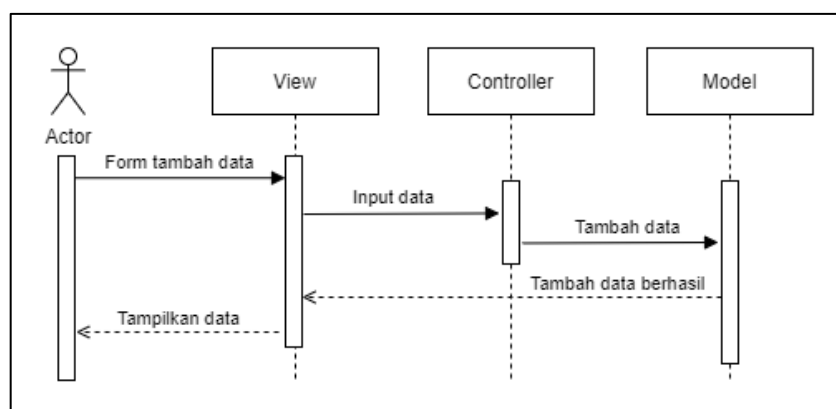
a. Sequence diagram admin

1) Sequence diagram login



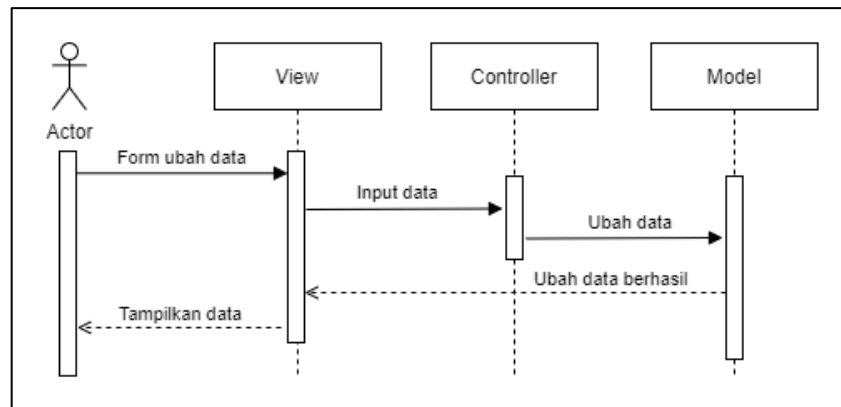
Gambar 4. 17 Sequence diagram login

2) Sequence diagram tambah data



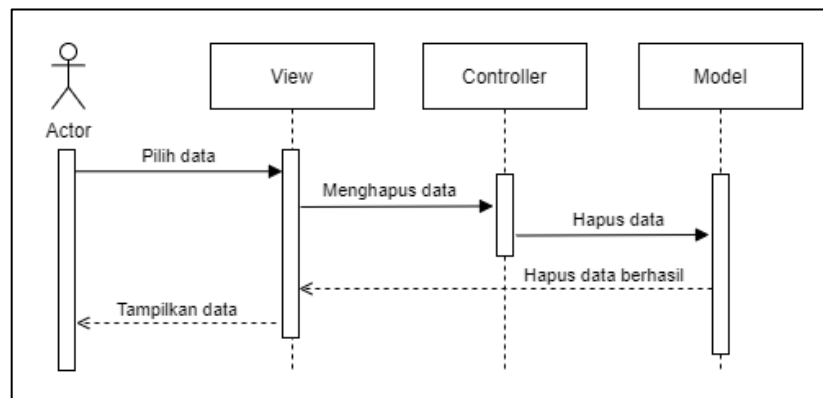
Gambar 4. 18 Sequence diagram tambah data

3) *Sequence diagram ubah data*



Gambar 4. 19 *Sequence diagram ubah data*

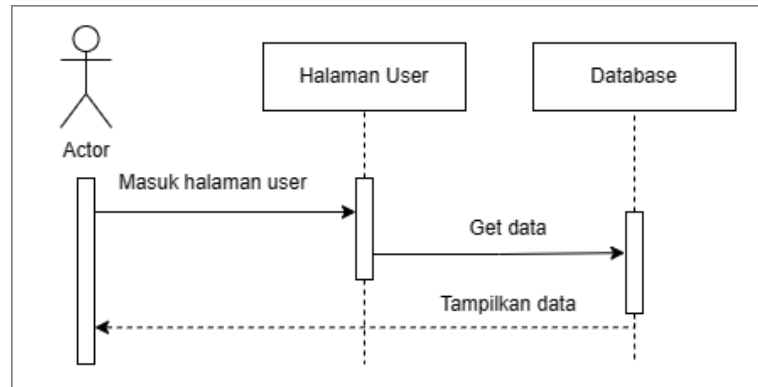
4) *Sequence diagram hapus data*



Gambar 4. 20 *Sequence diagram hapus data*

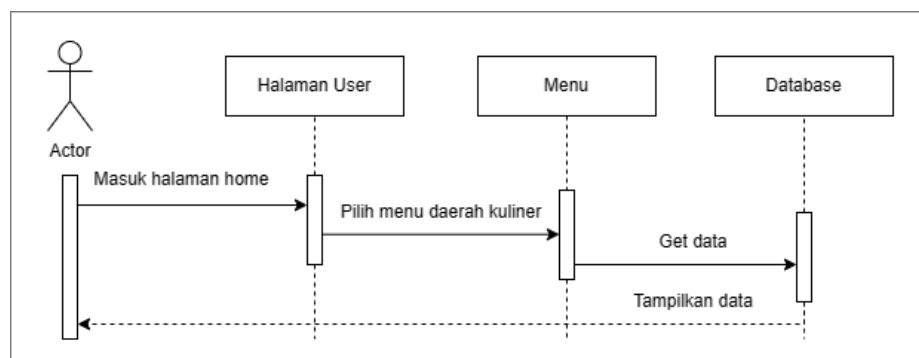
b. Sequence diagram User

1) Sequence diagram arsip



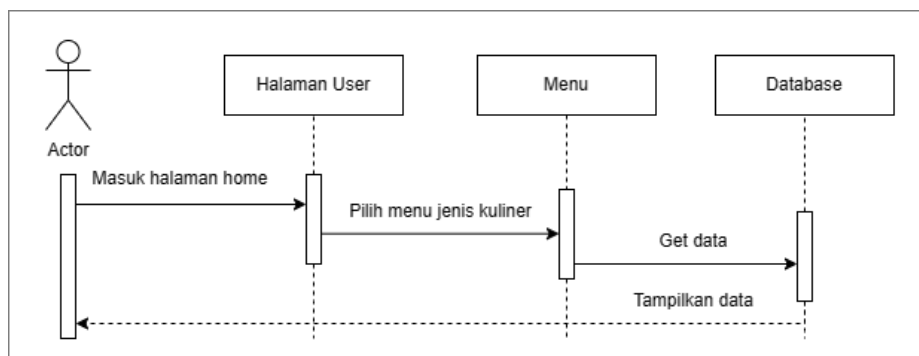
Gambar 4. 21 Sequence diagram arsip

2) Sequence diagram daerah kuliner



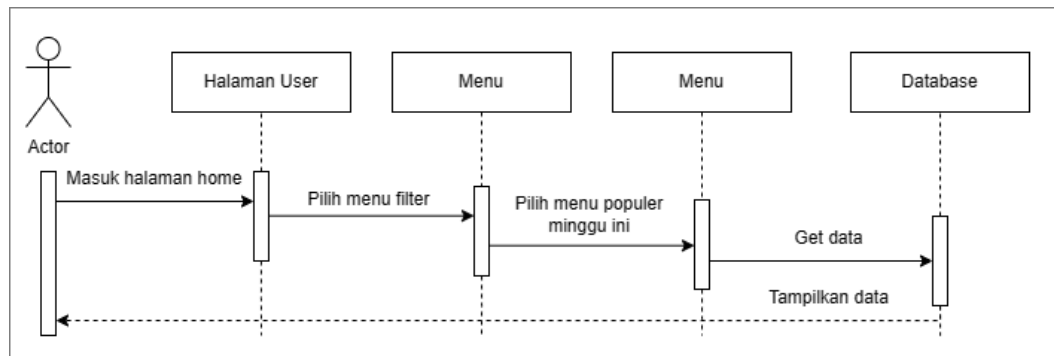
Gambar 4. 22 Sequence diagram daerah kuliner

3) Sequence diagram jenis kuliner



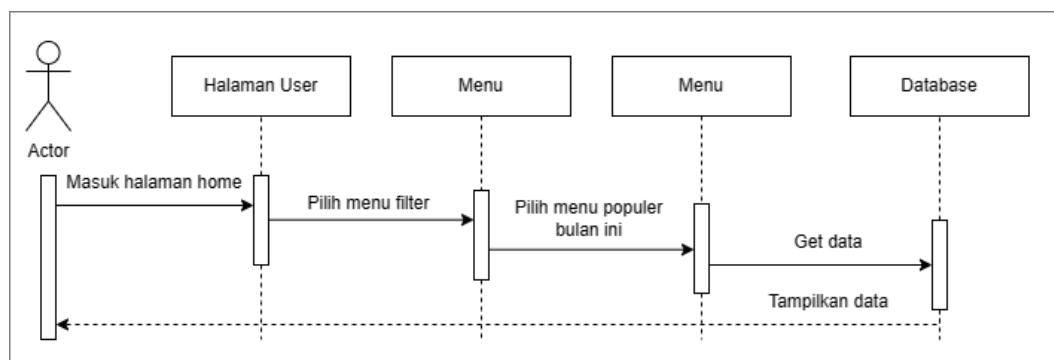
Gambar 4. 23 Sequence diagram jenis kuliner

4) *Sequence diagram filter populer minggu ini*



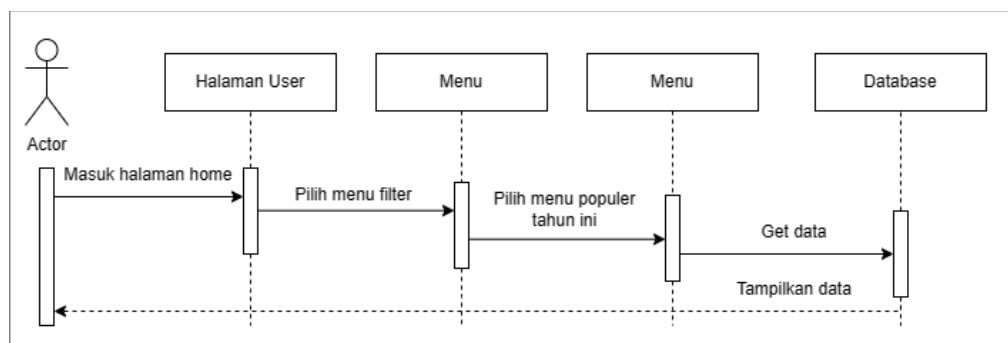
Gambar 4. 24 *Sequence diagram filter populer minggu ini*

5) *Sequence diagram filter populer bulan ini*



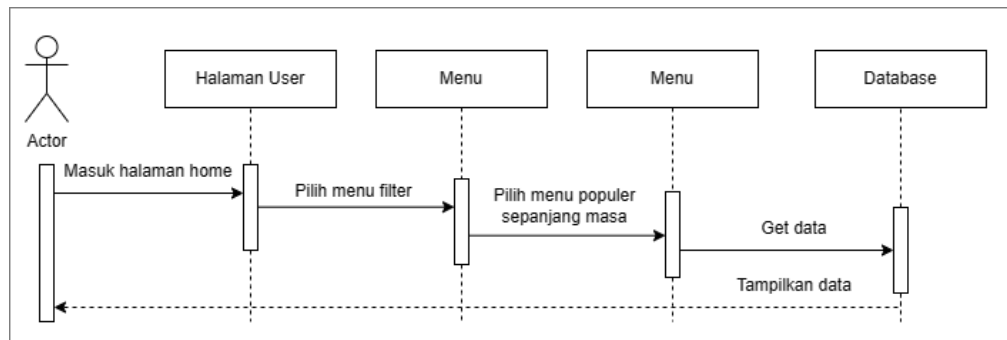
Gambar 4. 25 *Sequence diagram filter populer bulan ini*

6) *Sequence diagram filter populer tahun ini*



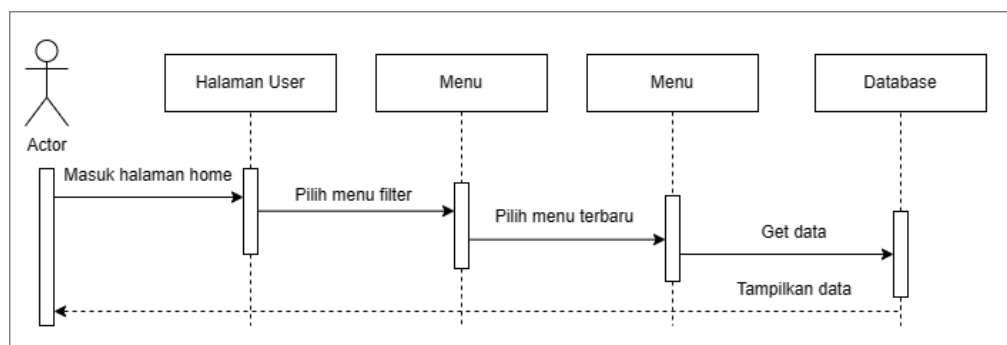
Gambar 4. 26 *Sequence diagram filter populer tahun ini*

7) *Sequence diagram filter populer sepanjang masa*



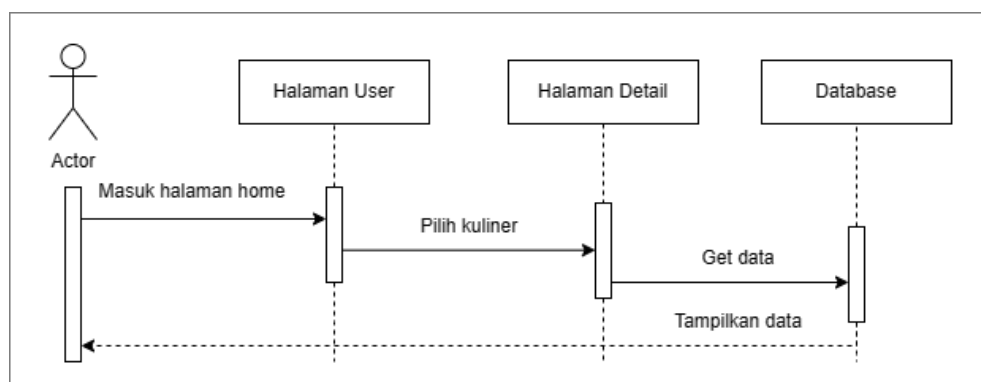
Gambar 4. 27 *Sequence diagram filter populer sepanjang masa*

8) *Sequence diagram filter terbaru*



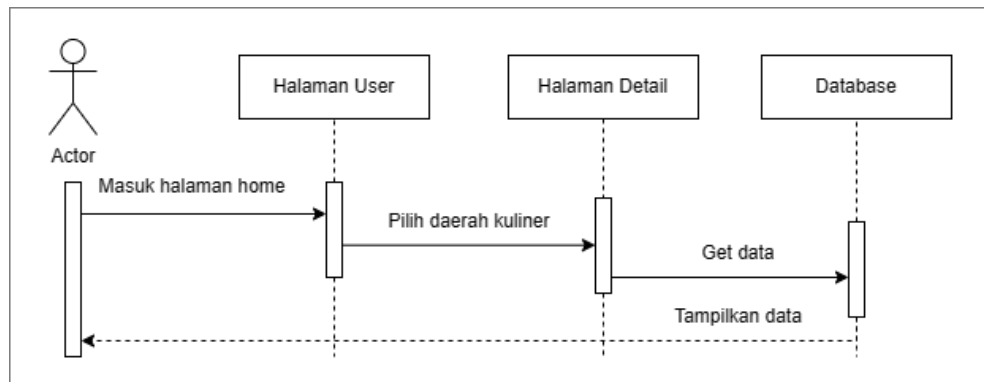
Gambar 4. 28 *Sequence diagram filter terbaru*

9) *Sequence diagram halaman detail*



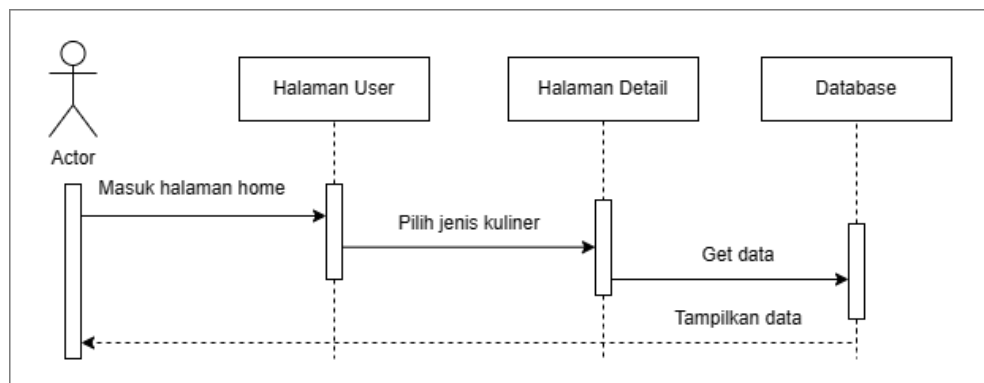
Gambar 4. 29 *Sequence diagram halaman detail*

10) *Sequence diagram* pilih daerah kuliner



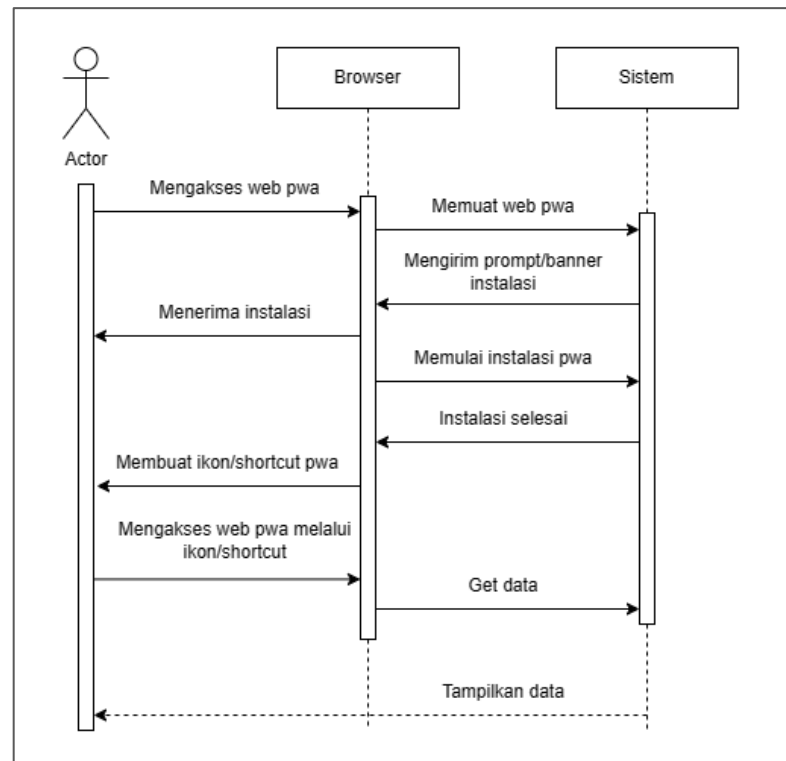
Gambar 4. 30 *Sequence diagram* pilih daerah kuliner

11) *Sequence diagram* pilih jenis kuliner



Gambar 4. 31 *Sequence diagram* pilih jenis kuliner

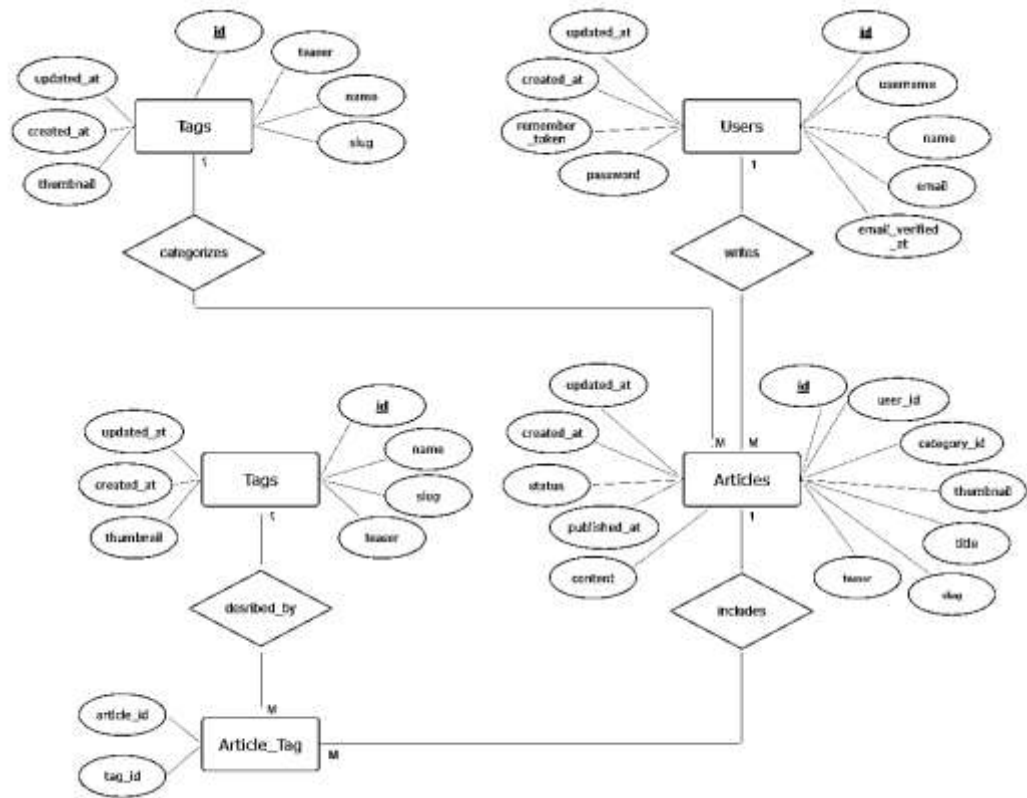
12) *Sequence diagram instalasi progressive web apps*



Gambar 4. 32 *Sequence diagram instalasi progressive web apps*

B. Rancangan Database

1. Diagram ERD



Gambar 4. 33 Diagram ER database

2. Relasi tabel



Gambar 4. 34 Relasi tabel *database*

Berikut adalah relasi tabel pada *database* Arsipkuliner:

a. *Users* dan *Articles*

Relasi: satu *user* dapat memiliki banyak *article*

Tipe relasi: *one-to-many*

Foreign key: 'user_id' di tabel *Articles* mengacu pada 'id' di tabel *users*

b. *Articles* dan *Categories*

Relasi: setiap *article* memiliki satu *category*

Tipe relasi: *many-to-one*

Foreign key: '*category_id*' di tabel *articles* mengacu pada '*id*' di tabel *categories*

c. *Categories* dan *Articles*

Relasi: Setiap *category* dapat memiliki banyak *article*

Tipe relasi: *one-to-many*

Foreign key: '*category_id*' di tabel *articles* mengacu pada '*id*' di tabel *categories*

d. *Articles* dan *Tags*

Relasi: Setiap *article* dapat memiliki banyak *tag* dan setiap *tag* dapat digunakan oleh banyak *article*

Tipe relasi: *many-to-many*

Tabel pivot: '*article_tag*' dengan kolom '*article_id*' dan '*tag_id*' sebagai *foreign key*

e. *Tags* dan *Articles*

Relasi: Setiap *tag* dapat digunakan oleh banyak *article*.

Tipe relasi: *many-to-many*

Tabel pivot: *article_tag* dengan kolom *article_id* dan *tag_id* sebagai *foreign key*

3. Kamus data

a. Tabel *articles*

Tabel 4. 1 Tabel *articles*

Column	Type	Attributes	Null	Default	Extra	Links to
id	bigint	UNSIGNED	No		Auto_increment	
user_id	bigint	UNSIGNED	No		Auto_increment	-> user.id ON UPDATE RESTRICT ON UPDATE RESTRICT
category_id	bigint	UNSIGNED	No			
thumbnail	varchar(255)		Yes	NULL		
title	varchar(255)		No			
slug	varchar(255)		No			
teaser	varchar(255)		No			
content	text		No			
published_at	timestamp		Yes	NULL		
status	varchar(255)		No	Pending		
created_at	timestamp		Yes	NULL		
updated_at	timestamp		Yes	NULL		

b. Tabel *categories*

Tabel 4. 2 Tabel *categories*

Column	Type	Attributes	Null	Default	Extra	Links to
id	bigint	UNSIGNED	No		Auto_increment	
name	varchar(255)		No			
slug	varchar(255)		No			
thumbnail	varchar(255)		Yes	NULL		
teaser	text		No			
created_at	timestamp		Yes	NULL		
updated_at	timestamp		Yes	NULL		

c. Tabel *tags*

Tabel 4. 3 Tabel *tags*

Column	Type	Attributes	Null	Default	Extra	Links to
id	bigint	UNSIGNED	No		Auto_increment	
name	varchar(255)		No			
slug	varchar(255)		No			
thumbnail	varchar(255)		Yes	NULL		
teaser	text		No			
created_at	timestamp		Yes	NULL		
updated_at	timestamp		Yes	NULL		

d. Tabel *users*

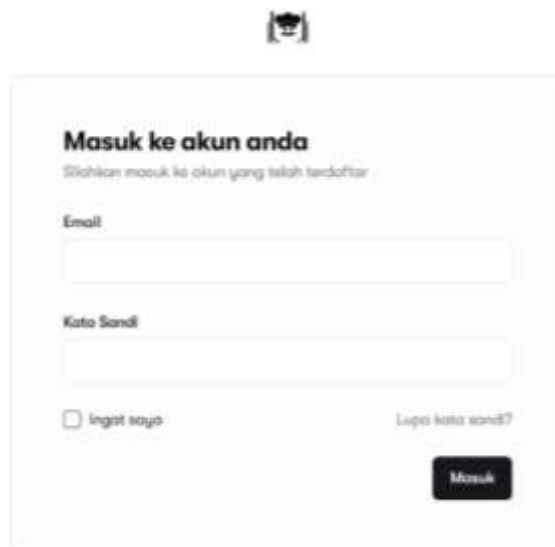
Tabel 4. 4 Tabel *users*

Column	Type	Attributes	Null	Default	Extra	Links to
id	bigint	UNSIGNED	No		Auto_increment	
username	varchar(255)		Yes	NULL		
name	varchar(255)		No			
<i>email</i>	varchar(255)		No			
<i>email_verified_at</i>	timestamp		Yes	NULL		
<i>password</i>	varchar(255)		No			
remember_token	varchar(100)		Yes	NULL		
created_at	timestamp		Yes	NULL		
updated_at	timestamp		Yes	NULL		

C. Detail Sistem

1. Halaman *login admin*

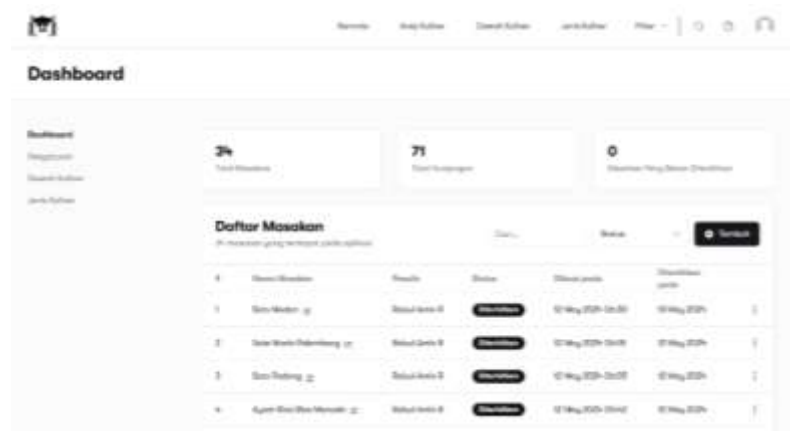
Antarmuka autentikasi pengelola sistem berfungsi sebagai pintu masuk utama yang dimanfaatkan oleh penanggung jawab untuk memperoleh izin akses ke area khusus pengelolaan.



Gambar 4. 35 Halaman *login admin*

2. Halaman *dashboard admin*

Laman *dashboard admin* merupakan tampilan yang menampilkan informasi mengenai jumlah data dari setiap tabel dalam aplikasi Ensiklopedia Masakan Nusantara. Pada halaman ini juga menampilkan informasi daftar masakan dan terdapat tombol untuk menambahkan data masakan.



Gambar 4. 36 Halaman *dashboard admin*

3. Halaman tambah data kuliner

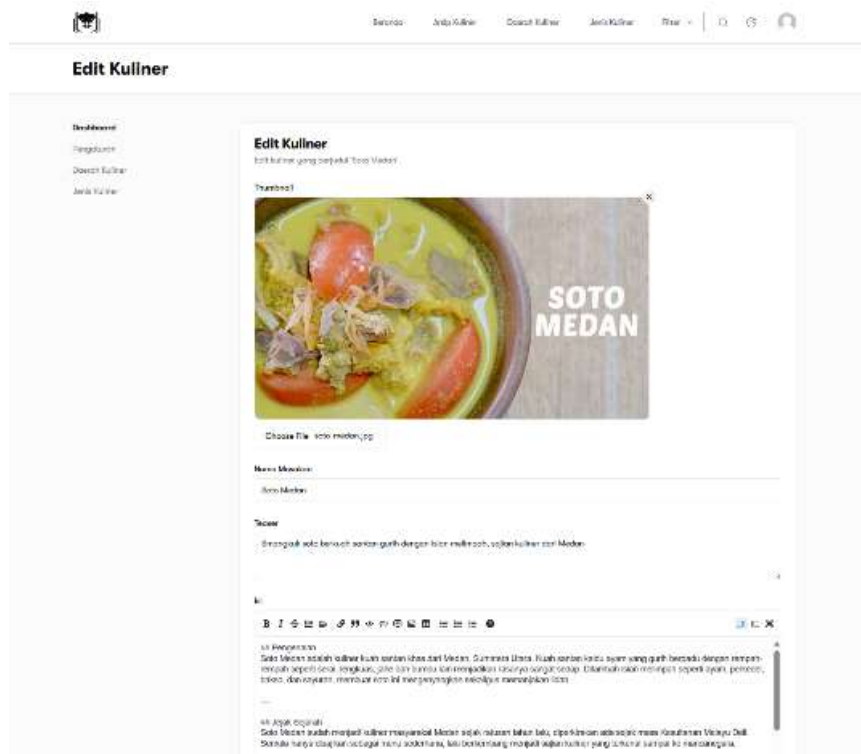
Halaman tambah data kuliner merupakan laman yang menyediakan formulir untuk memasukkan data kuliner.



Gambar 4. 37 Halaman tambah data kuliner

4. Halaman ubah data kuliner

Halaman ubah data kuliner merupakan laman yang menyediakan formulir untuk mengubah data kuliner.



Gambar 4. 38 Halaman ubah data kuliner

5. Halaman hapus data kuliner

Halaman hapus data kuliner merupakan laman yang menyediakan formulir untuk menghapus data kuliner.



Gambar 4. 39 Halaman hapus data kuliner

6. Halaman pengaturan *admin*

Halaman pengaturan *admin* merupakan laman yang menyediakan formulir untuk mengubah informasi pribadi *admin*, memperbarui kata sandi, dan menghapus akun.

Gambar 4. 40 Halaman pengaturan *admin*

7. Halaman daerah kuliner *admin*

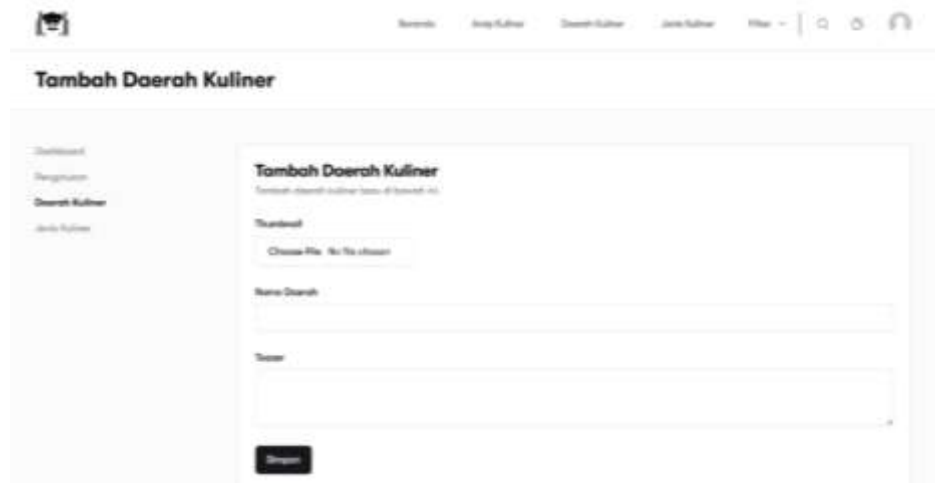
Halaman daerah kuliner *admin* merupakan halaman yang menampilkan informasi mengenai jumlah data dari tabel daerah kuliner dalam. Di laman ini tersedia tombol untuk memasukkan data daerah kuliner.

#	Nama	Slug	Masukan
1	Bali	bali	3
2	Bangka Belitung	bangka-belitung	2
3	Banten	banten	1

Gambar 4. 41 Halaman daerah kuliner *admin*

8. Halaman tambah data daerah kuliner

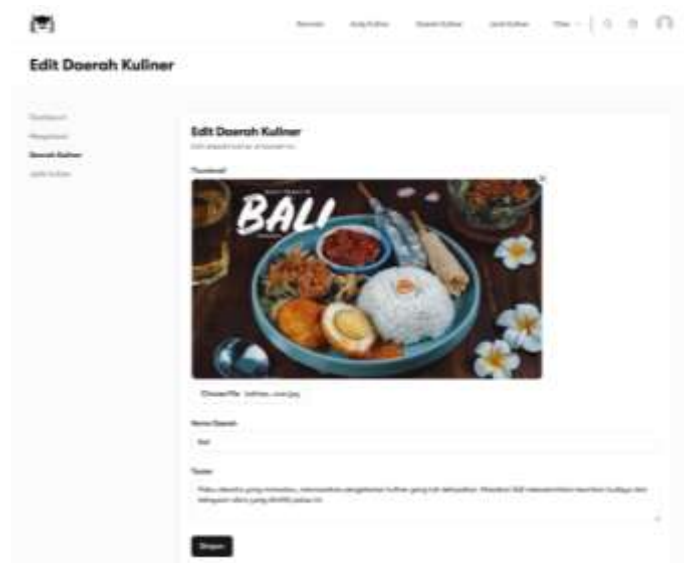
Halaman tambah data daerah kuliner merupakan laman yang menyediakan formulir untuk memasukkan data daerah kuliner.



Gambar 4. 42 Halaman tambah data daerah kuliner

9. Halaman ubah data daerah kuliner

Halaman ubah data daerah kuliner yaitu laman yang menyediakan formulir untuk mengubah data daerah kuliner.



Gambar 4. 43 Halaman ubah data daerah kuliner

10. Halaman hapus data daerah kuliner

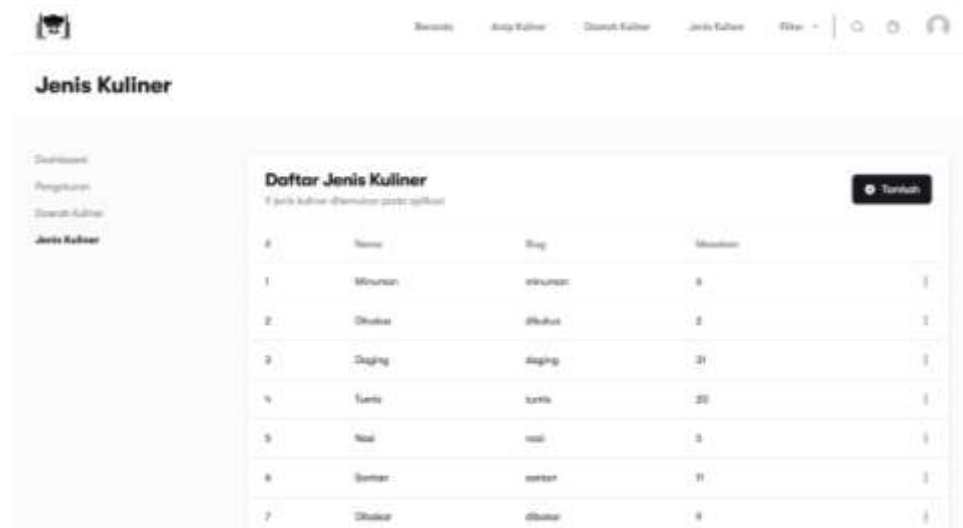
Halaman hapus data daerah kuliner merupakan halaman yang menampilkan peringatan untuk menghapus data daerah kuliner.



Gambar 4. 44 Halaman hapus data daerah kuliner

11. Halaman jenis kuliner *admin*

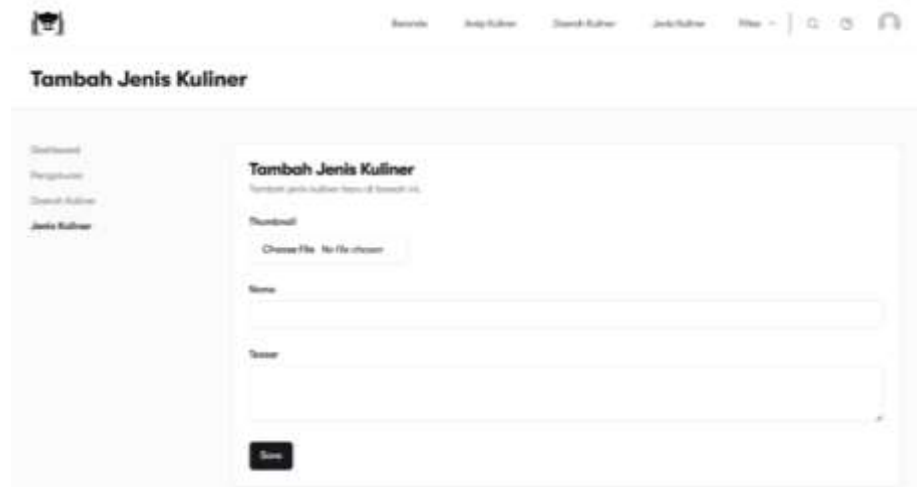
Halaman jenis kuliner *admin* merupakan halaman yang menampilkan informasi mengenai jumlah data dari tabel jenis kuliner. Di laman ini tersedia tombol untuk memasukkan data jenis kuliner.



Gambar 4. 45 Halaman jenis kuliner *admin*

12. Halaman tambah data jenis kuliner

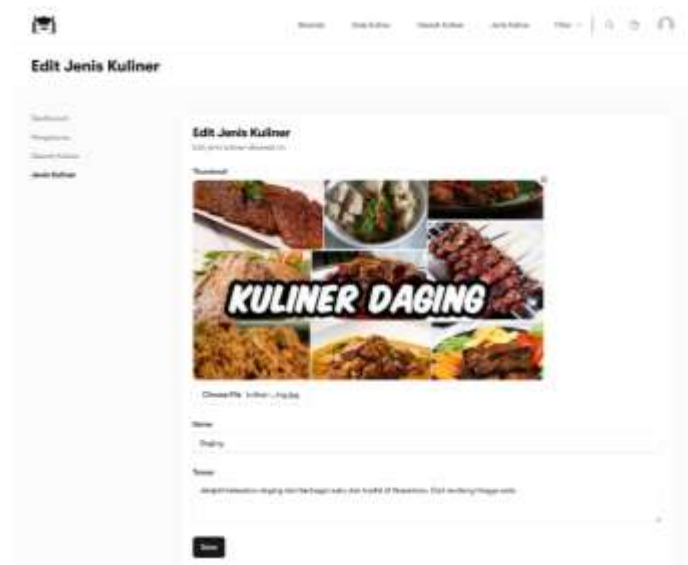
Halaman tambah data jenis kuliner yaitu laman yang menyediakan formulir untuk memasukan data jenis kuliner.



Gambar 4. 46 Halaman tambah data jenis kuliner

13. Halaman ubah data jenis kuliner

Halaman ubah data jenis kuliner merupakan laman yang menyediakan formulir untuk mengubah data jenis kuliner.



Gambar 4. 47 Halaman ubah data jenis kuliner

14. Halaman hapus data jenis kuliner

Halaman hapus data daerah kuliner merupakan halaman yang menampilkan peringatan untuk menghapus data jenis kuliner.

Hapus Jenis Kuliner

Apakah anda yakin ingin menghapus jenis kuliner ini?

Batal

Lanjutkan

Gambar 4. 48 Halaman hapus data jenis kuliner

15. Halaman *slash screen*

Halaman *slash screen* atau layar pembuka merupakan tampilan pertama yang muncul ketika sebuah aplikasi atau program diluncurkan, sebelum memasuki tampilan utama dari aplikasi tersebut.



Gambar 4. 49 Halaman *slash screen*

16. Halaman beranda

Laman utama merupakan antarmuka awal yang ditampilkan saat mengakses sistem.



Gambar 4. 50 Halaman beranda

17. Halaman beranda – *Progressive web apps*

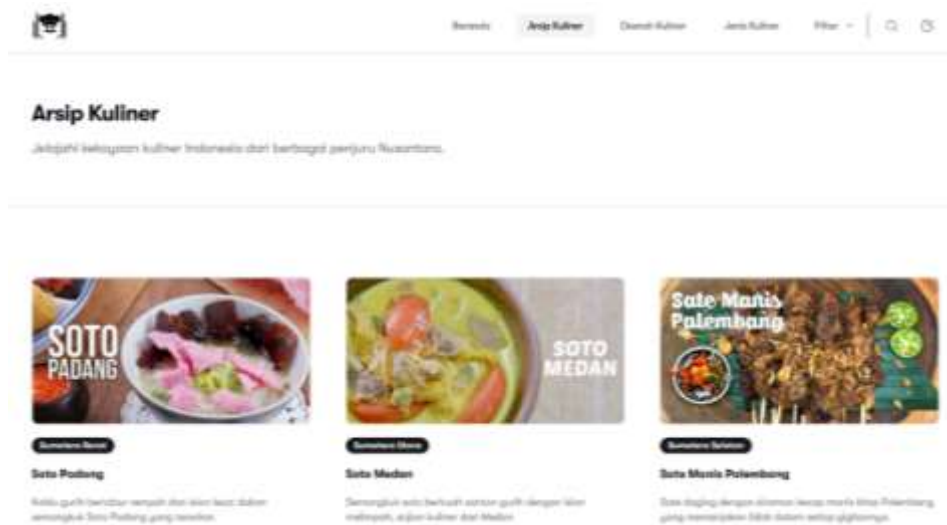
Ketika sebuah *Progressive Web App* (PWA) diakses melalui perangkat mobile, memiliki halaman yang responsif untuk menyesuaikan tampilan dengan beragam ukuran layar ponsel, dan aksesibilitas satu tangan.



Gambar 4. 51 Halaman beranda *progressive web apps*

18. Halaman arsip kuliner

Halaman arsip kuliner menampilkan berbagai pilihan masakan nusantara yang terdapat pada aplikasi ensiklopedia masakan nusantara.



Gambar 4. 52 Halaman arsip kuliner

19. Halaman arsip kuliner – *Progressive web apps*

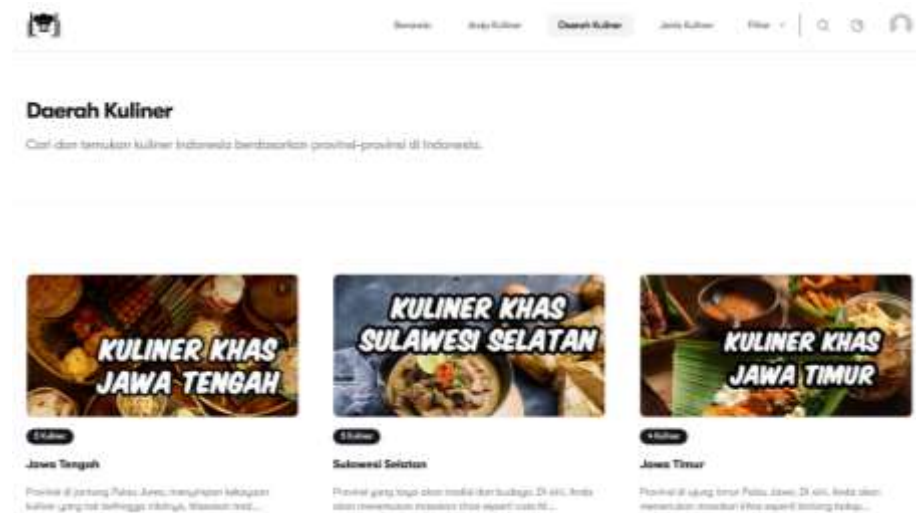
Halaman yang responsif untuk menyesuaikan tampilan dengan beragam ukuran layar ponsel, dan aksesibilitas satu tangan.



Gambar 4. 53 Halaman arsip kuliner *progressive web apps*

20. Halaman daerah kuliner

Halaman daerah kuliner menampilkan berbagai pilihan daerah masakan nusantara yang terdapat pada aplikasi.



Gambar 4. 54 Halaman daerah kuliner

21. Halaman daerah kuliner – *Progressive web apps*

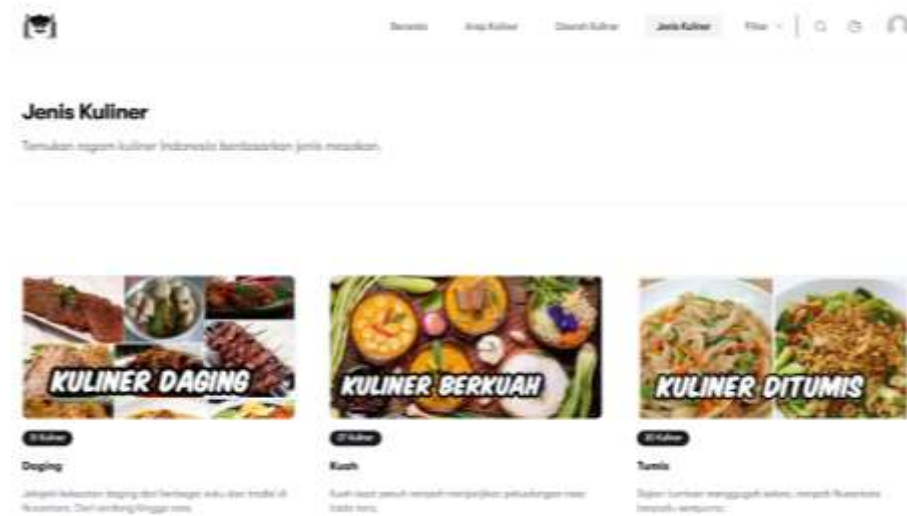
Halaman yang responsif untuk menyesuaikan tampilan dengan beragam ukuran layar ponsel, dan aksesibilitas satu tangan.



Gambar 4. 55 Halaman daerah kuliner *progressive web apps*

22. Halaman jenis kuliner

Halaman jenis kuliner menampilkan berbagai pilihan jenis hidangan masakan nusantara yang terdapat pada aplikasi.



Gambar 4. 56 Halaman jenis kuliner

23. Halaman jenis kuliner – *Progressive web apps*

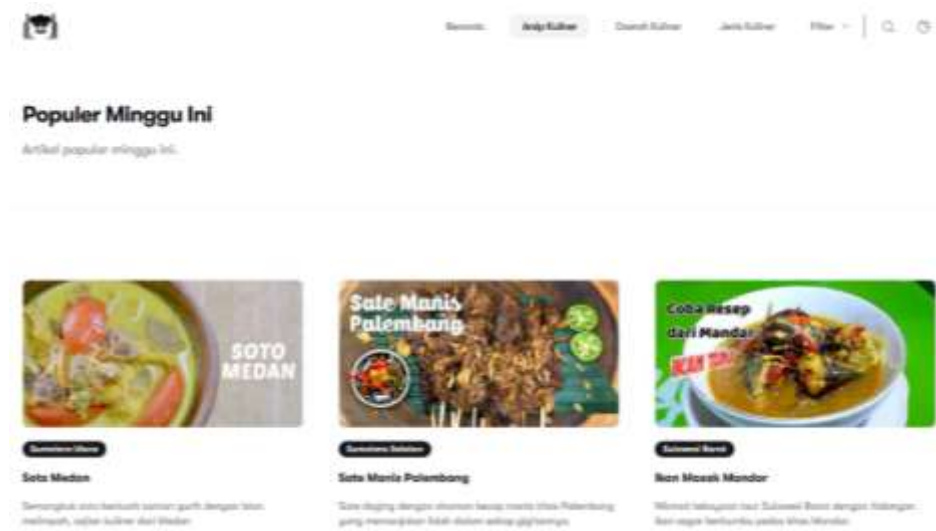
Halaman yang responsif untuk menyesuaikan tampilan dengan beragam ukuran layar ponsel, dan aksesibilitas satu tangan.



Gambar 4. 57 Halaman jenis kuliner *progressive web apps*

24. Halaman *filter* populer minggu ini

Halaman *filter* kuliner populer minggu ini menampilkan berbagai pilihan masakan nusantara yang populer minggu ini yang terdapat pada aplikasi.



Gambar 4. 58 Halaman *filter* populer minggu ini

25. Halaman *filter* populer minggu ini – *Progressive web apps*

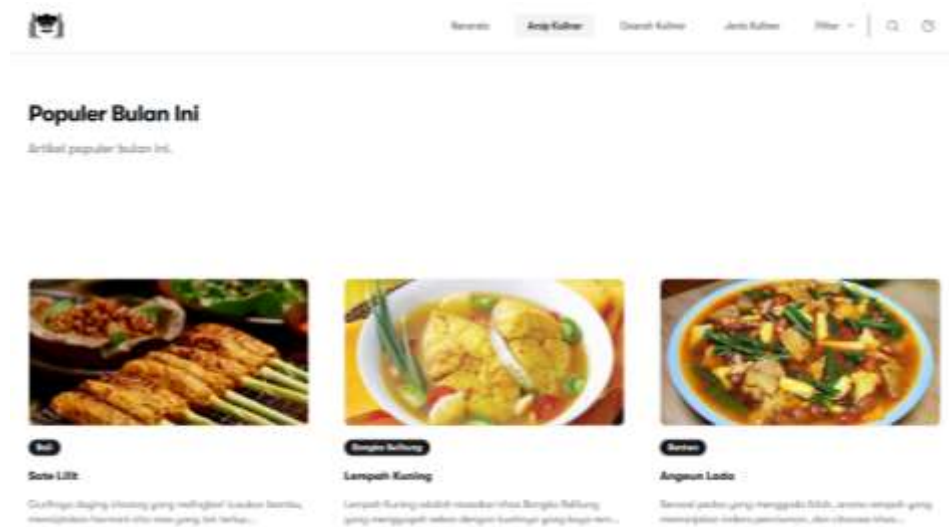
Halaman yang responsif untuk menyesuaikan tampilan dengan beragam ukuran layar ponsel, dan aksesibilitas satu tangan.



Gambar 4. 59 Halaman *filter* populer minggu ini *progressive web apps*

26. Halaman *filter* populer bulan ini

Halaman *filter* kuliner populer bulan ini menampilkan berbagai pilihan masakan nusantara yang populer bulan ini yang terdapat pada aplikasi.



Gambar 4. 60 Halaman *filter* populer bulan ini

27. Halaman *filter* populer bulan ini – *Progressive web apps*

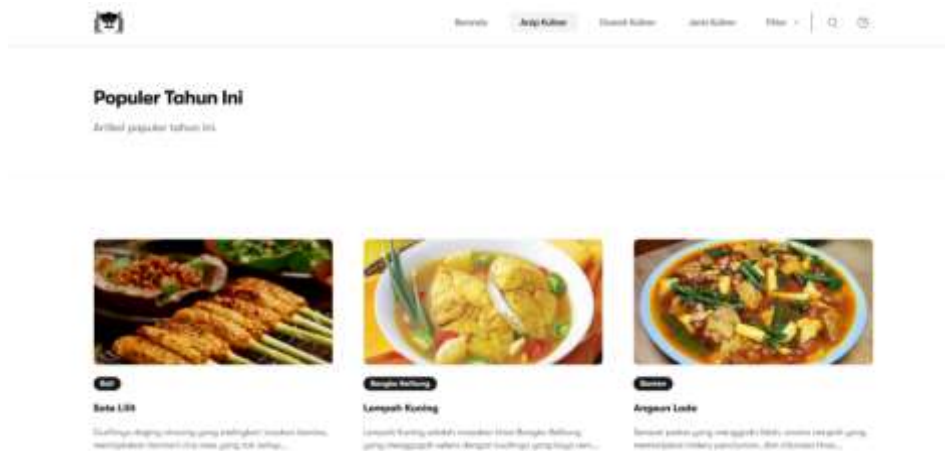
Halaman yang responsif untuk menyesuaikan tampilan dengan beragam ukuran layar ponsel, dan aksesibilitas satu tangan.



Gambar 4. 61 Halaman *filter* populer bulan ini *progressive web apps*

28. Halaman *filter* populer tahun ini

Halaman *filter* kuliner populer tahun ini menampilkan berbagai pilihan masakan nusantara yang populer tahun ini yang terdapat pada aplikasi.



Gambar 4. 62 Halaman *filter* populer tahun ini

29. Halaman *filter* populer tahun ini – *Progressive web apps*

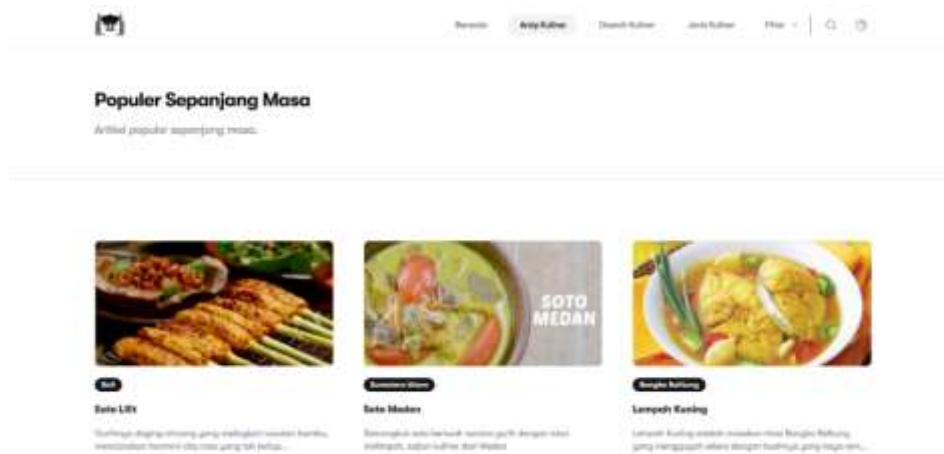
Halaman yang responsif untuk menyesuaikan tampilan dengan beragam ukuran layar ponsel, dan aksesibilitas satu tangan.



Gambar 4. 63 Halaman *filter* populer tahun ini *progressive web apps*

30. Halaman *filter* populer sepanjang masa

Halaman *filter* kuliner populer sepanjang masa menampilkan berbagai pilihan masakan nusantara yang populer sepanjang masa yang terdapat pada aplikasi ensiklopedia masakan nusantara.



Gambar 4. 64 Halaman *filter* populer sepanjang masa

31. Halaman *filter* populer sepanjang masa – *Progressive web apps*

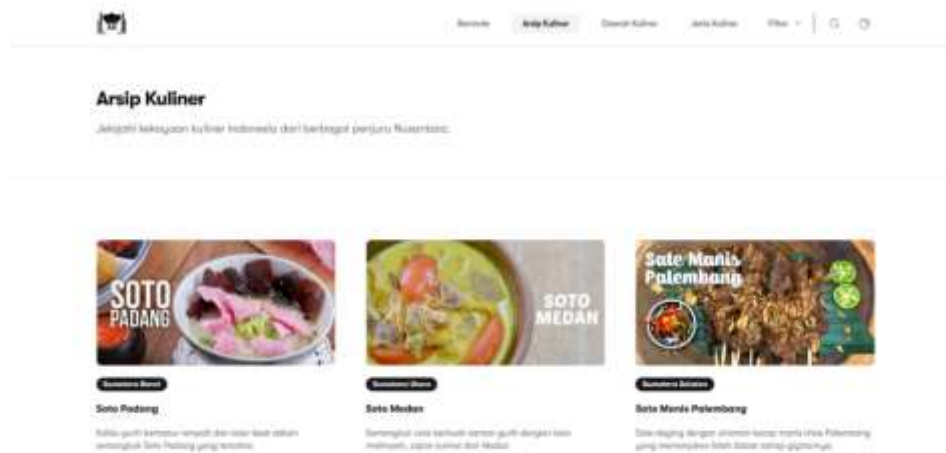
Halaman yang responsif untuk menyesuaikan tampilan dengan beragam ukuran layar ponsel, dan aksesibilitas satu tangan.



Gambar 4. 65 Halaman *filter* populer sepanjang masa *progressive web apps*

32. Halaman *filter* terbaru

Halaman *filter* kuliner terbaru menampilkan berbagai pilihan masakan nusantara yang populer baru ditambahkan oleh *admin* yang terdapat pada aplikasi ensiklopedia masakan nusantara.



Gambar 4. 66 Halaman *filter* terbaru

33. Halaman *filter* terbaru – *Progressive web apps*

Halaman yang responsif untuk menyesuaikan tampilan dengan beragam ukuran layar ponsel, dan aksesibilitas satu tangan.

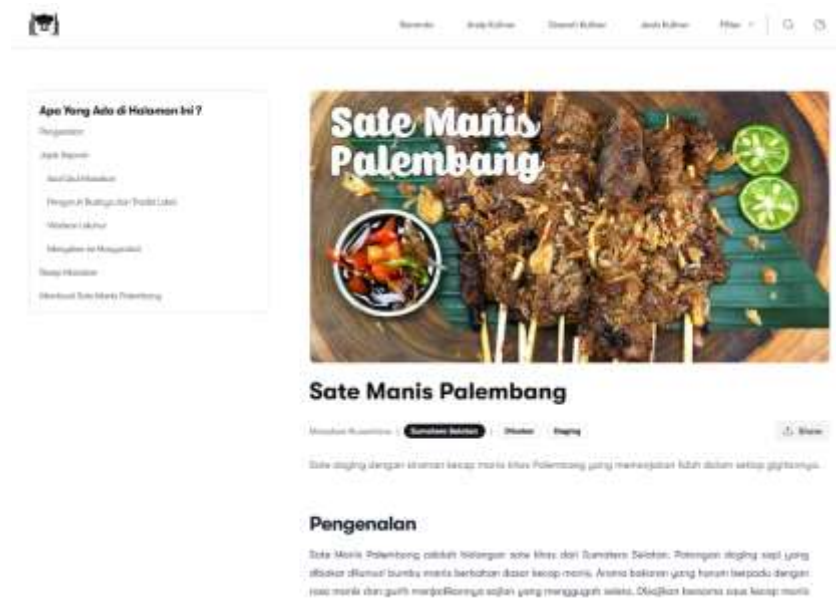


Gambar 4. 67 Halaman *filter* terbaru *progressive web apps*

34. Halaman detail masakan nusantara

Halaman detail masakan nusantara memberikan informasi lengkap tentang masakan seperti nama masakan, gambar masakan, informasi asal

masakan, jenis hidangan, deskripsi singkat, pengenalan masakan, sejarah masakan, resep masakan, dan cara membuat masakan.



Gambar 4. 68 Halaman detail masakan nusantara

35. Halaman detail masakan nusantara – *Progressive web apps*

Halaman yang responsif untuk menyesuaikan tampilan dengan beragam ukuran layar ponsel, dan aksesibilitas satu tangan.



Gambar 4. 69 Halaman detail masakan nusantara *progressive web apps*

D. Pengujian Sistem

Pengujian sistem bertujuan untuk mengidentifikasi kesalahan, kesenjangan, atau kekurangan dalam sistem sebelum digunakan oleh pengguna akhir. Ada dua pendekatan utama dalam pengujian sistem ini, yaitu pengujian *Black box* dan pengujian *white box*.

1. *Black box*

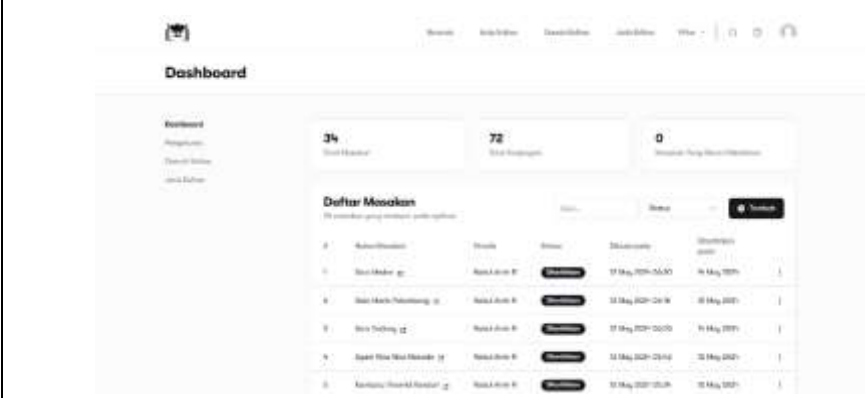
a. *Black box testing kesalahan email dan password*

Tabel 4. 5 *Black box testing kesalahan email dan password*

Tes faktor	Hasil	Keterangan
Memasukkan <i>email</i> atau <i>password</i> yang tidak sesuai	✓	Berhasil, ketika <i>email</i> atau <i>password</i> tidak sesuai tampil <i>login failed</i>
Screenshot		
		

b. *Black box testing login berhasil*

Tabel 4. 6 *Black box testing login berhasil*

Tes faktor	Hasil	Keterangan
Memasukkan <i>email</i> atau <i>password</i> yang benar	✓	Sistem berhasil menampilkan halaman <i>admin/dashboard</i> .
Screenshot		
		

c. *Black box testing tambah data*

Tabel 4. 7 *Black box testing tambah data*

Tes faktor	Hasil	Keterangan
<i>Admin</i> mengisi <i>form</i> tambah dan menekan tombol <i>simpan</i>	✓	Sukses, sebuah notifikasi muncul bahwa data telah berhasil ditambahkan.
Screenshot		
		

d. *Black box testing* ubah data

Tabel 4. 8 *Black box testing* ubah data

Tes faktor	Hasil	Keterangan
<i>Admin</i> mengisi <i>form</i> ubah data dan menekan tombol <i>simpan</i>	✓	Sukses, sebuah notifikasi muncul bahwa data telah berhasil diubah.
Screenshot		
 The screenshot shows a web interface with a table containing columns: 'Penulis', 'Status', 'Dibuat pada', and 'Dibuatkan pada'. The table lists several entries with the status 'Diterbitkan'. A dark notification box at the bottom right displays a green checkmark and the text 'Kuliner berhasil diubah.' (Cuisine successfully updated).		

e. *Black box testing* hapus data

Tabel 4. 9 *Black box testing* hapus data

Tes faktor	Hasil	Keterangan
<i>Admin</i> menghapus salah satu data	✓	Sukses, sebuah notifikasi muncul bahwa data telah berhasil dihapus.
Screenshot		
 The screenshot shows the same web interface as the previous one, but with a different notification box at the bottom right displaying a green checkmark and the text 'Kuliner berhasil dihapus.' (Cuisine successfully deleted).		

f. **Black box testing halaman splash screen**

Tabel 4. 10 *Black box testing* halaman *splash screen*

Tes faktor	Hasil	Keterangan
Pemakai melakukan inisiasi penggunaan perangkat lunak	✓	Sukses, tampil halaman beranda
Screenshot		
		

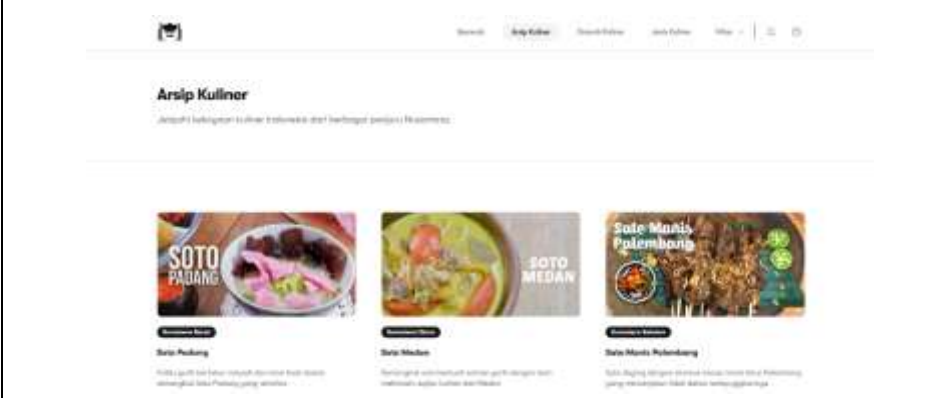
g. **Black box testing halaman beranda**

Tabel 4. 11 *Black box testing* halaman beranda

Tes faktor	Hasil	Keterangan
Pemakai melakukan inisiasi penggunaan perangkat lunak	✓	Sukses, tampil halaman beranda
Screenshot		
		

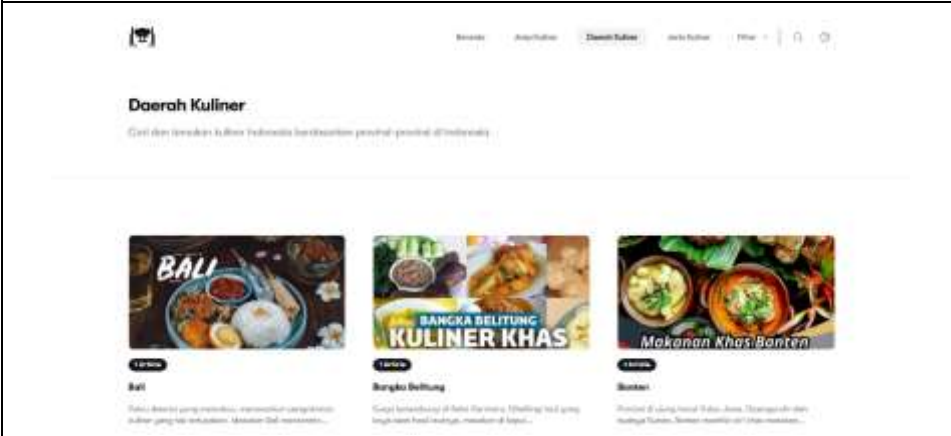
h. *Black box testing* halaman arsip kuliner

Tabel 4. 12 *Black box testing* halaman arsip kuliner

Tes faktor	Hasil	Keterangan
Pemakai menekan tombol navigasi arsip kuliner	✓	Sukses, tampil halaman arsip kuliner
Screenshot		
		

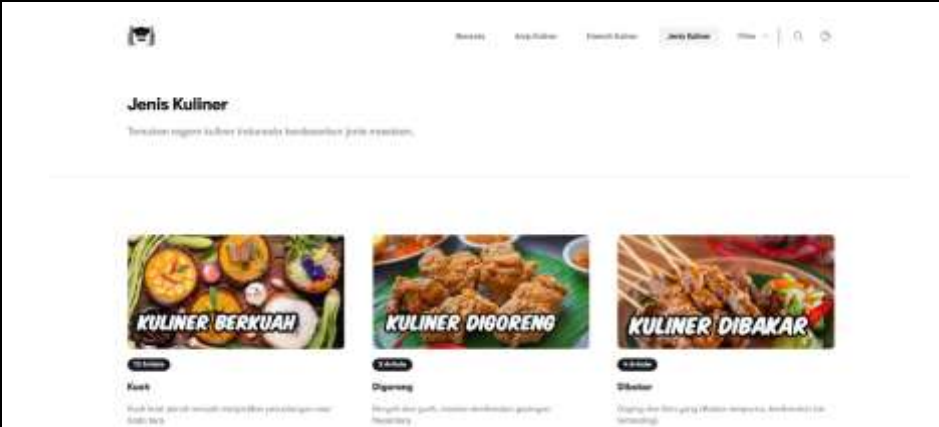
i. *Black box testing* halaman daerah kuliner

Tabel 4. 13 *Black box testing* halaman daerah kuliner

Tes faktor	Hasil	Keterangan
Pemakai menekan tombol navigasi daerah kuliner	✓	Sukses, tampil halaman daerah kuliner
Screenshot		
		

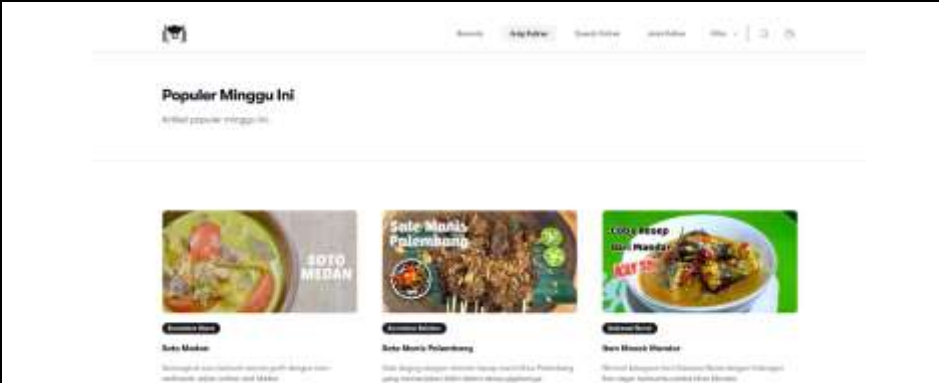
j. *Black box testing* halaman jenis kuliner

Tabel 4. 14 *Black box testing* halaman jenis kuliner

Tes faktor	Hasil	Keterangan
Pemakai mengaktifkan tombol navigasi jenis kuliner	✓	Sukses, tampil halaman jenis kuliner
Screenshot		
 The screenshot shows a web page titled 'Jenis Kuliner' with a subtitle 'Terdapat ragam kuliner Indonesia berdasarkan jenis masakan'. Below the title, there are three featured food items: 'KULINER BERKUAH' (Kuah), 'KULINER DIGORENG' (Digoreng), and 'KULINER DIBAKAR' (Dibakar). Each item has a corresponding image and a brief description.		

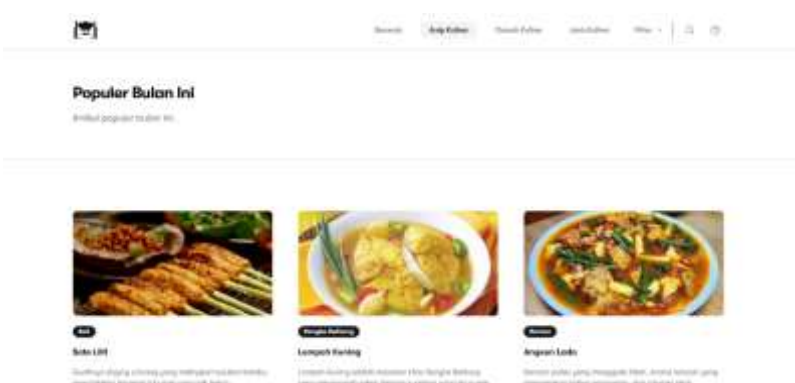
k. *Black box testing* filter populer minggu ini

Tabel 4. 15 *Black box testing* filter populer minggu ini

Tes faktor	Hasil	Keterangan
Pemakai mengaktifkan tombol navigasi <i>filter</i> dan memilih pilihan populer minggu ini	✓	Sukses, tampil halaman arsip kuliner yang telah di <i>filter</i> berdasarkan pilihan <i>User</i>
Screenshot		
 The screenshot shows a web page titled 'Populer Minggu Ini' with a subtitle 'Artikel populer minggu ini'. Below the title, there are three featured food items: 'SOTO MEDAN', 'Soto Mami Palembang', and 'Soto Hentak Damar'. Each item has a corresponding image and a brief description.		

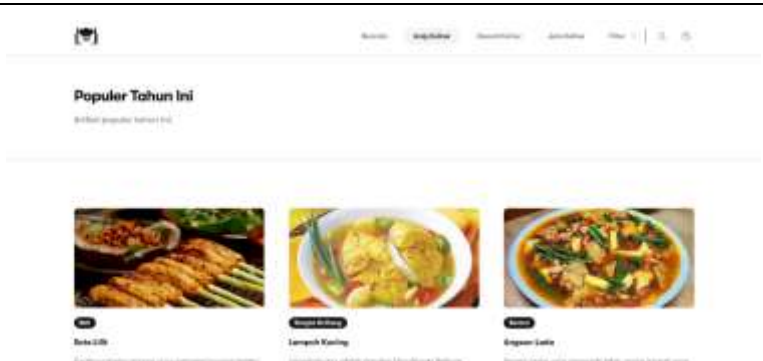
1. **Black box testing filter populer bulan ini**

Tabel 4. 16 Black box testing filter populer bulan ini

Tes faktor	Hasil	Keterangan
Pemakai mengaktifkan tombol navigasi <i>filter</i> dan memilih pilihan populer bulan ini	✓	Sukses, tampil halaman arsip kuliner yang telah di <i>filter</i> berdasarkan pilihan <i>User</i>
Screenshot		
		

m. **Black box testing filter populer tahun ini**

Tabel 4. 17 Black box testing filter populer tahun ini

Tes faktor	Hasil	Keterangan
Pemakai mengaktifkan tombol navigasi <i>filter</i> dan memilih pilihan populer tahun ini	✓	Sukses, tampil halaman arsip kuliner yang telah di <i>filter</i> berdasarkan pilihan <i>User</i>
Screenshot		
		

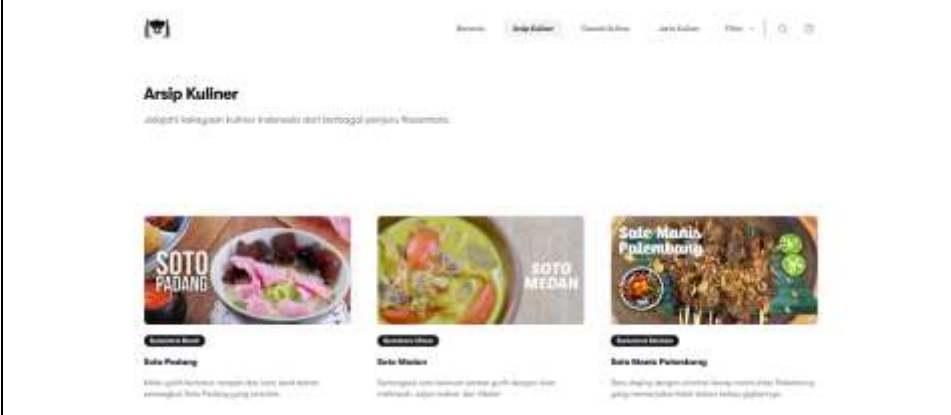
n. **Black box testing filter populer sepanjang masa**

Tabel 4. 18 *Black box testing filter populer sepanjang masa*

Tes faktor	Hasil	Keterangan
Pemakai mengaktifkan tombol navigasi <i>filter</i> dan memilih pilihan populer sepanjang masa	✓	Sukses, tampil halaman arsip kuliner yang telah di <i>filter</i> berdasarkan pilihan <i>User</i> .
Screenshot		
		

o. **Black box testing filter terbaru**

Tabel 4. 19 *Black box testing filter terbaru*

Tes faktor	Hasil	Keterangan
Pemakai mengaktifkan tombol navigasi <i>filter</i> dan memilih pilihan terbaru	✓	Sukses, tampil halaman arsip kuliner yang telah di <i>filter</i> berdasarkan pilihan <i>User</i>
Screenshot		
		

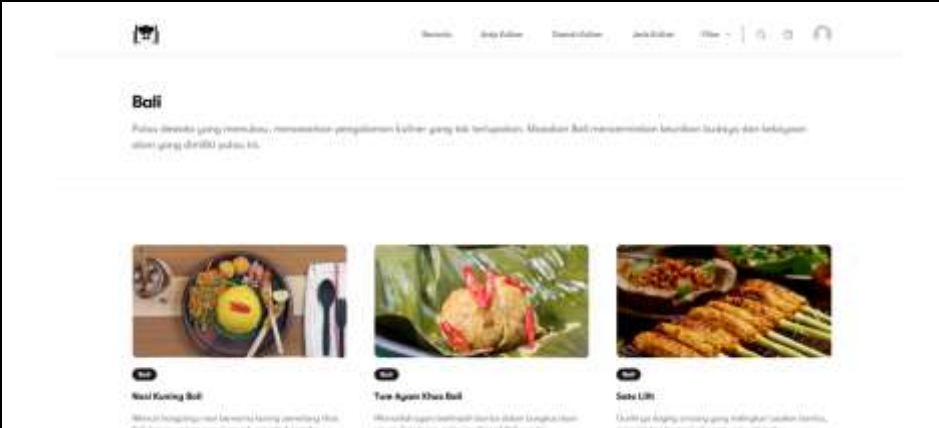
p. **Black box testing** halaman detail

Tabel 4. 20 *Black box testing* halaman detail

Tes faktor	Hasil	Keterangan
Pemakai mengaktifkan <i>card</i> kuliner nusantara	✓	Sukses, tampil halaman detail kuliner nusantara
Screenshot		
		

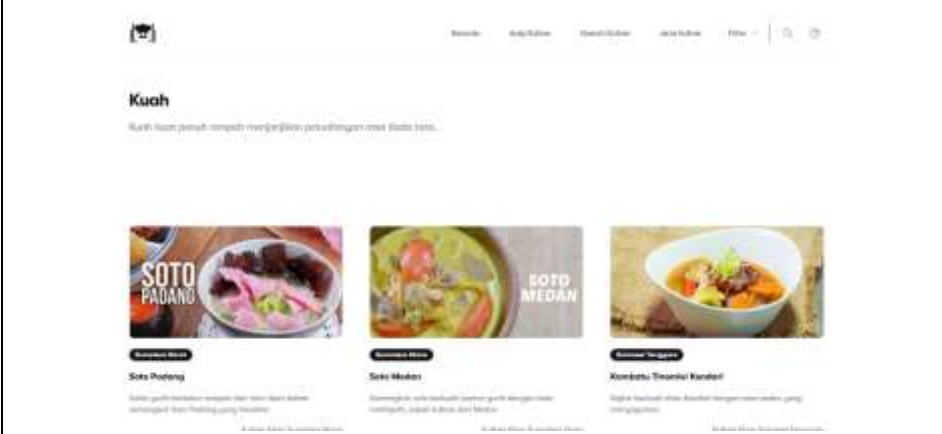
q. **Black box testing** pilih daerah kuliner

Tabel 4. 21 *Black box testing* pilih daerah kuliner

Tes faktor	Hasil	Keterangan
Pemakai mengaktifkan <i>card</i> daerah kuliner yang berada di halaman daerah kuliner.	✓	Sukses, tampil halaman daftar kuliner nusantara berdasarkan daerah yang dipilih <i>User</i> .
Screenshot		
		

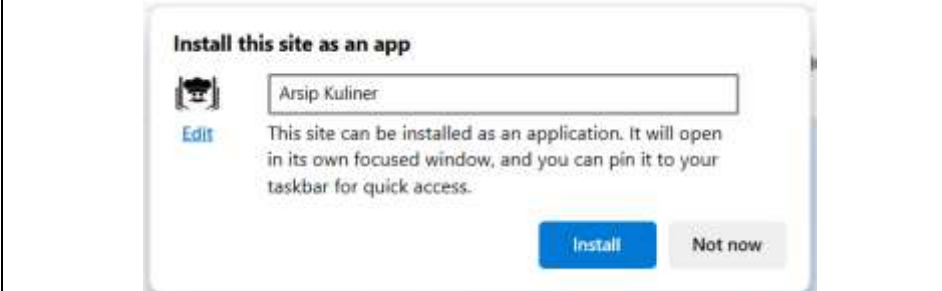
r. **Black box testing** pilih jenis kuliner

Tabel 4. 22 *Black box testing* pilih jenis kuliner

Tes faktor	Hasil	Keterangan
Pemakai mengaktifkan <i>card</i> jenis kuliner yang berada di halaman jenis kuliner.	✓	Sukses, tampil halaman daftar kuliner nusantara berdasarkan jenis yang dipilih <i>User</i> .
Screenshot		
		

s. **Black box testing** instalasi *progressive web apps*

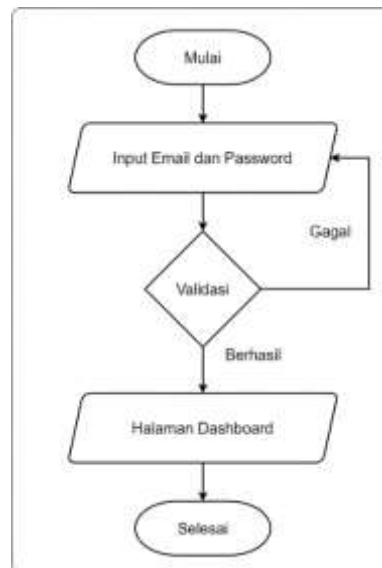
Tabel 4. 23 *Black box testing* instalasi *progressive web apps*

Tes faktor	Hasil	Keterangan
Pemakai mengaktifkan tombol install aplikasi ini	✓	Sukses, ikon/shortcut aplikasi ditambahkan di <i>homescreen</i>
Screenshot		
		

2. White box

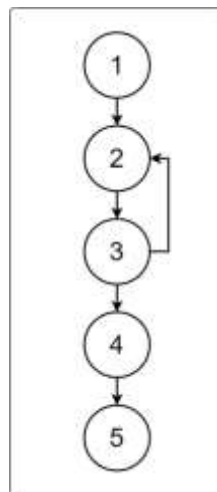
a. White box testing kesalahan email dan password

(1) Flowchart



Gambar 4. 70 Flowchart kesalahan email dan password

(2) Flowgraph



Gambar 4. 71 Flowgraph kesalahan email dan password

Berdasarkan gambar 4. 71 diatas, dilakukan perhitungan sebagai berikut:

(1) Menghitung *cyclomatic complexity* $V(G)$ pada *edge* dan *node*

$$\text{Pada rumus : } V(G) = E - N + 2$$

$$E \text{ (edge)} = 5$$

$$N \text{ (node)} = 5$$

$$P \text{ (Predikat node)} = 1$$

Penyelesaian :

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

$$= 5 - 5 + 2$$

$$= 2$$

$$\text{Predikat (P)} = P + 1$$

$$= 1 + 1$$

$$= 2$$

(2) Berdasarkan perhitungan *cyclomatic complexity* dari *flowgraph*

diatas memiliki *region* = 2

(3) *Independent path* pada *flowgraph* yaitu:

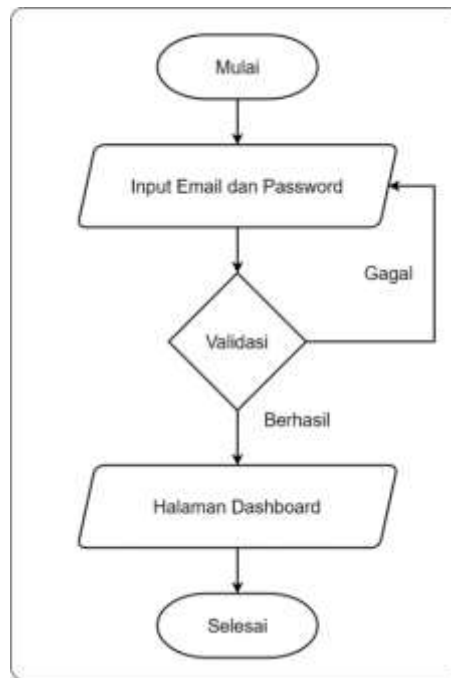
$$\text{Path 1} = 1 - 2 - 3 - 2$$

$$\text{Path 2} = 1 - 2 - 3 - 4 - 5$$

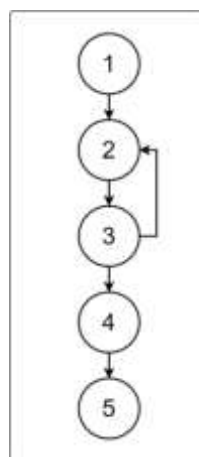
(4) Grafik matriks kesalahan *email dan password*

Tabel 4. 24 Grafik matriks kesalahan *email dan password*

	1	2	3	4	5	E-1
1		1				$1 - 1 = 0$
2			1			$1 - 1 = 0$
3		1		1		$2 - 1 = 1$
4					1	$1 - 1 = 0$
5						0
	SUM (E + 1)					$1 + 1 = 2$

b. White box testing login berhasil**1) Flowchart**

Gambar 4. 72 Flowchart login berhasil

2) Flowgraph

Gambar 4. 73 Flowgraph login berhasil

Berdasarkan gambar 4. 73 diatas, dilakukan perhitungan sebagai berikut:

(1) Menghitung *cyclomatic complexity* $V(G)$ pada *edge* dan *node*

Pada rumus : $V(G) = E - N + 2$

E (*edge*) = 5

N (*node*) = 5

P (predikat *node*) = 1

Penyelesaian :

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

$= 5 - 5 + 2$

$= 2$

Predikat (P) = $P + 1$

$= 1 + 1$

$= 2$

(2) Berdasarkan perhitungan *cyclomatic complexity* dari *flowgraph*

diatas memiliki *region* = 2

(3) *Independent path* pada *flowgraph* yaitu:

Path 1 = 1 – 2 – 3 – 2

Path 2 = 1 – 2 – 3 – 4 – 5

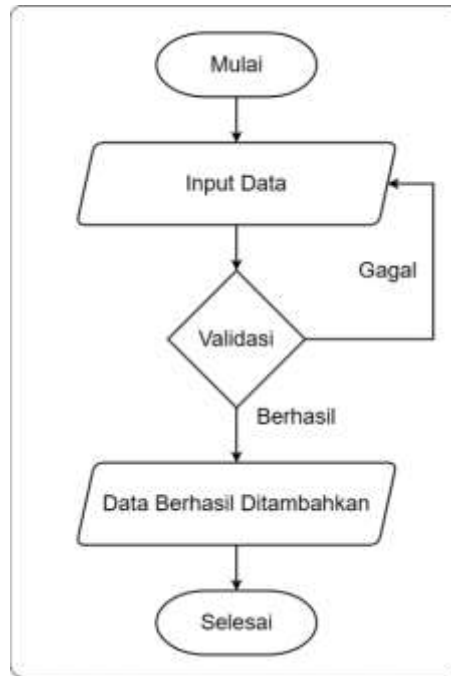
(4) Grafik matriks *login* berhasil

Tabel 4. 25 Grafik matriks *login* berhasil

	1	2	3	4	5	E-1
1		1				$1 - 1 = 0$
2			1			$1 - 1 = 0$
3		1		1		$2 - 1 = 1$
4					1	$1 - 1 = 0$
5						0
	SUM ($E + 1$)					$1 + 1 = 2$

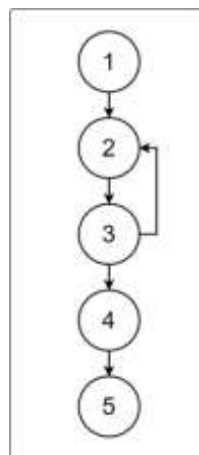
c. *White box testing* tambah data

1) *Flowchart*



Gambar 4. 74 *Flowchart* tambah data

2) *Flowgraph*



Gambar 4. 75 *Flowgraph* tambah data

Berdasarkan gambar 4. 75 diatas, dilakukan perhitungan sebagai berikut:

(1) Menghitung *cyclomatic complexity* $V(G)$ pada *edge* dan *node*

$$\text{Pada rumus : } V(G) = E - N + 2$$

$$E (\text{edge}) = 5$$

$$N (\text{node}) = 5$$

$$P (\text{Predikat } \textit{node}) = 1$$

Penyelesaian :

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

$$= 5 - 5 + 2$$

$$= 2$$

$$\text{Predikat (P)} = P + 1$$

$$= 1 + 1$$

$$= 2$$

(2) Berdasarkan perhitungan *cyclomatic complexity* dari *flowgraph*

diatas memiliki *region* = 2

(3) *Independent path* pada *flowgraph* yaitu:

$$\text{Path 1} = 1 - 2 - 3 - 2$$

$$\text{Path 2} = 1 - 2 - 3 - 4 - 5$$

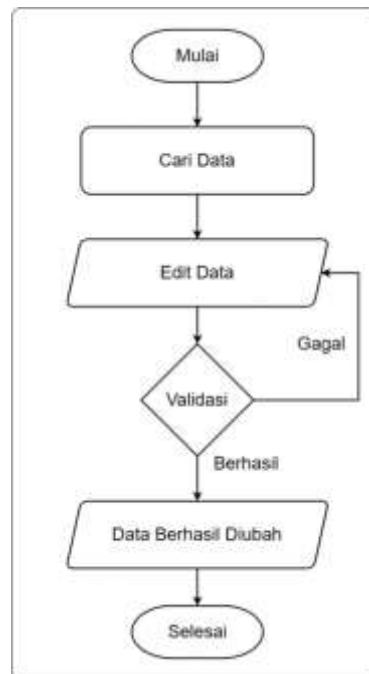
(4) Grafik matriks tambah data

Tabel 4. 26 Grafik matriks tambah data

	1	2	3	4	5	E-1
1		1				$1 - 1 = 0$
2			1			$1 - 1 = 0$
3		1		1		$2 - 1 = 1$
4					1	$1 - 1 = 0$
5						0
	SUM (E + 1)					$1 + 1 = 2$

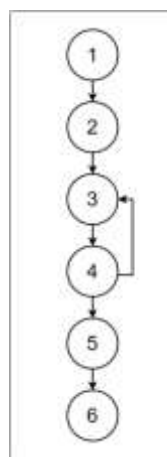
d. *White box testing* ubah data

1) *Flowchart*



Gambar 4. 76 *Flowchart* ubah data

2) *Flowgraph*

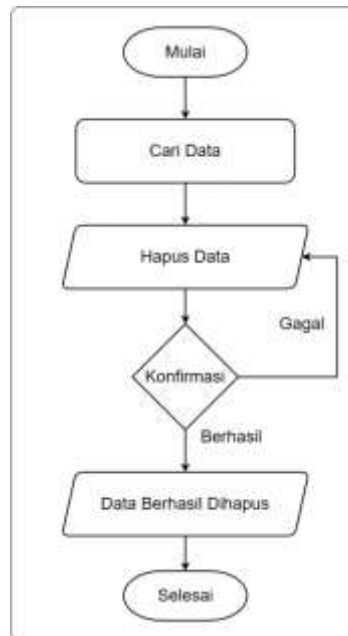


Gambar 4. 77 *Flowgraph* ubah data

Berdasarkan gambar 4. 77 diatas, dilakukan perhitungan sebagai berikut:

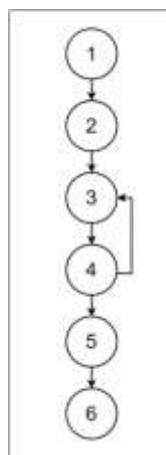
e. *White box testing* hapus data

1) *Flowchart*



Gambar 4. 78 *Flowchart* hapus data

2) *Flowgraph*

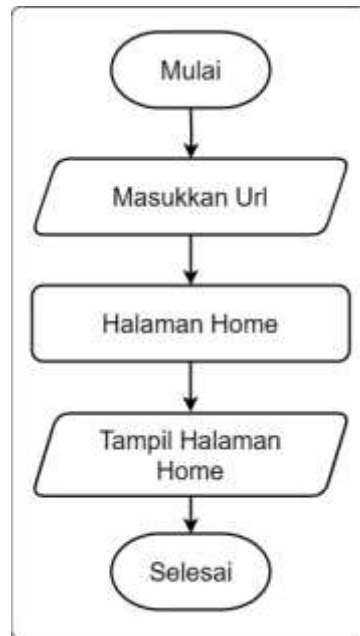


Gambar 4. 79 *Flowgraph* hapus data

Berdasarkan gambar 4. 79 diatas, dilakukan perhitungan sebagai berikut:

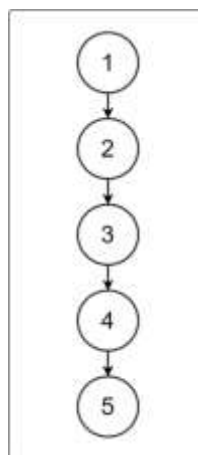
f. *White box testing* halaman beranda/home

1) *Flowchart*



Gambar 4. 80 *Flowchart* halaman beranda/home

2) *Flowgraph*



Gambar 4. 81 *Flowgraph* halaman beranda/home

Berdasarkan gambar 4. 81 diatas, dilakukan perhitungan sebagai berikut:

(1) Menghitung *cyclomatic complexity* $V(G)$ pada *edge* dan *node*.

$$\text{Pada rumus : } V(G) = E - N + 2$$

$$E \text{ (edge)} = 4$$

$$N \text{ (node)} = 5$$

$$P \text{ (Predikat node)} = 0$$

Penyelesaian :

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

$$= 4 - 5 + 2$$

$$= 1$$

$$\text{Predikat (P)} = P + 1$$

$$= 0 + 1$$

$$= 1$$

(2) Berdasarkan perhitungan *cyclomatic complexity* dari *flowgraph*

diatas memiliki *region* = 1

(3) *Independent path* pada *flowgraph* yaitu:

$$\text{Path 1} = 1 - 2 - 3 - 4 - 5$$

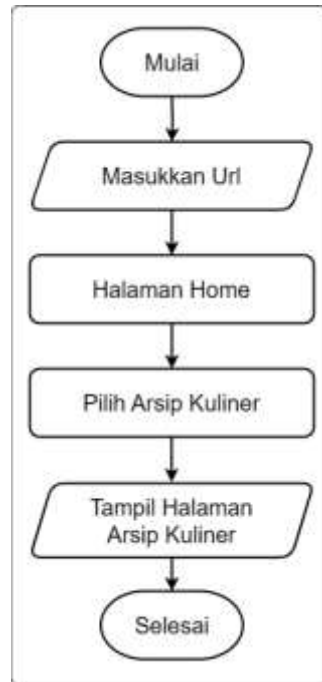
(4) Grafik matriks halaman beranda/*home*

Tabel 4. 29 Grafik matriks halaman beranda/*home*

	1	2	3	4	5	E-1
1		1				$1 - 1 = 0$
2			1			$1 - 1 = 0$
3				1		$1 - 1 = 0$
4					1	$1 - 1 = 0$
5						0
	SUM (E + 1)					$0 + 1 = 1$

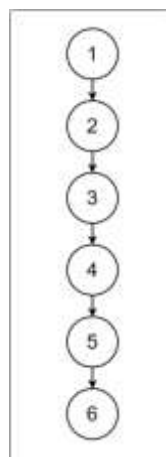
g. White box testing halaman arsip kuliner

1) Flowchart



Gambar 4. 82 *Flowchart* halaman arsip kuliner

2) Flowgraph

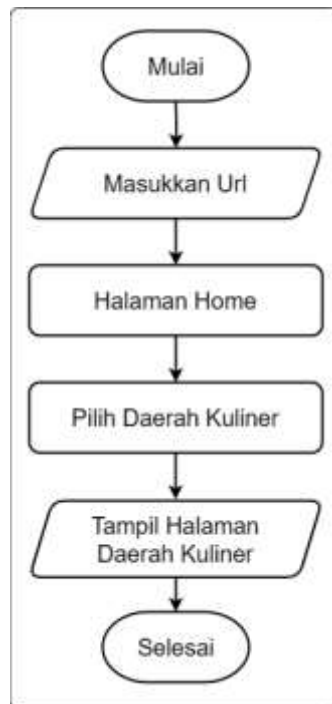


Gambar 4. 83 *Flowgraph* halaman arsip kuliner

Berdasarkan gambar 4. 83 diatas, dilakukan perhitungan sebagai berikut:

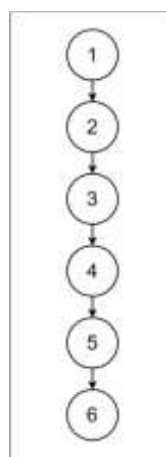
h. *White box testing* halaman daerah kuliner

1) *Flowchart*



Gambar 4. 84 *Flowchart* halaman daerah kuliner

2) *Flowgraph*

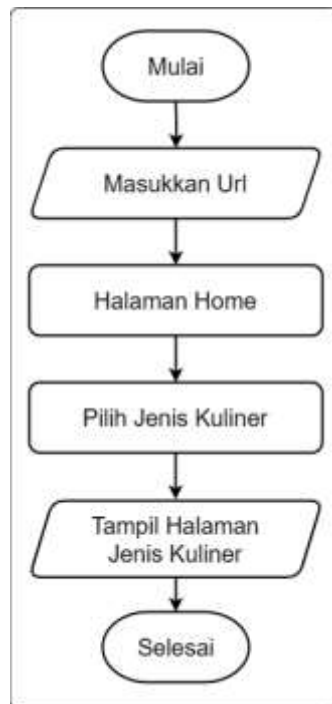


Gambar 4. 85 *Flowgraph* halaman daerah kuliner

Berdasarkan gambar 4. 85 diatas, dilakukan perhitungan sebagai berikut:

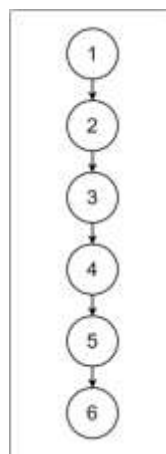
i. *White box testing* halaman jenis kuliner

1) *Flowchart*



Gambar 4. 86 *Flowchart* halaman jenis kuliner

2) *Flowgraph*

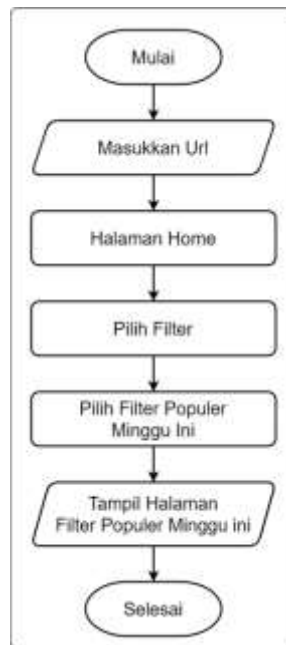


Gambar 4. 87 *Flowgraph* halaman jenis kuliner

Berdasarkan gambar 4. 87 diatas, dilakukan perhitungan sebagai berikut:

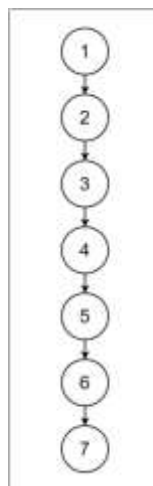
j. *White box testing* halaman *filter populer minggu ini*

1) *Flowchart*



Gambar 4. 88 *Flowchart* halaman *filter populer minggu ini*

2) *Flowgraph*

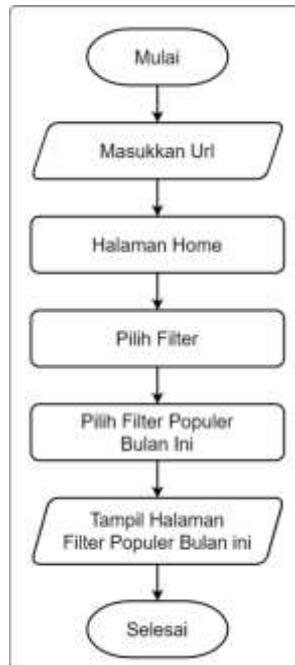


Gambar 4. 89 *Flowgraph* halaman *filter populer minggu ini*

Berdasarkan gambar 4. 89 diatas, dilakukan perhitungan sebagai berikut:

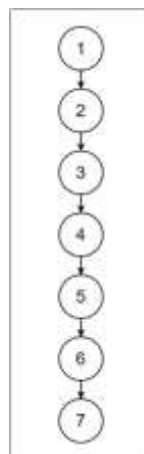
k. *White box testing* halaman *filter populer bulan ini*

1) *Flowchart*



Gambar 4. 90 *Flowchart* halaman *filter populer bulan ini*

2) *Flowgraph*



Gambar 4. 91 *Flowgraph* halaman *filter populer bulan ini*

Berdasarkan gambar 4. 91 diatas, dilakukan perhitungan sebagai berikut:

(1) Menghitung *cyclomatic complexity* $V(G)$ pada *edge* dan *node*.

$$\text{Pada rumus : } V(G) = E - N + 2$$

$$E \text{ (edge)} = 6$$

$$N \text{ (node)} = 7$$

$$P \text{ (Predikat node)} = 0$$

Penyelesaian :

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

$$= 6 - 7 + 2$$

$$= 1$$

$$\text{Predikat (P)} = P + 1$$

$$= 0 + 1$$

$$= 1$$

(2) Berdasarkan perhitungan *cyclomatic complexity* dari *flowgraph*

diatas memiliki *region* = 1

(3) *Independent path* pada *flowgraph* yaitu:

$$\text{Path 1} = 1 - 2 - 3 - 4 - 5 - 6 - 7$$

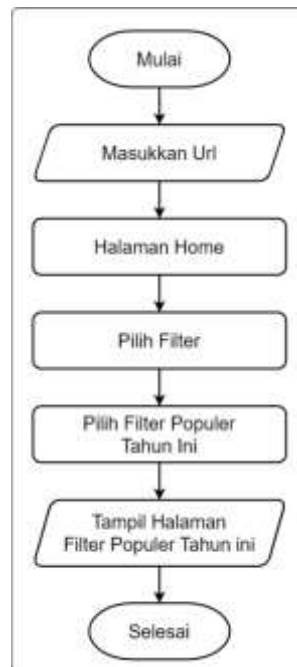
(4) Grafik matriks halaman *filter* populer bulan ini

Tabel 4. 34 Grafik matriks halaman *filter* populer bulan ini

	1	2	3	4	5	6	7	E-1
1		1						$1 - 1 = 0$
2			1					$1 - 1 = 0$
3				1				$1 - 1 = 0$
4					1			$1 - 1 = 0$
5						1		$1 - 1 = 0$
6							1	$1 - 1 = 0$
7								0
	SUM (E + 1)							$0 + 1 = 1$

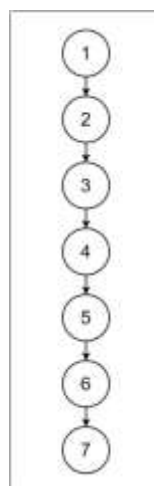
1. *White box testing filter populer tahun ini*

1) *Flowchart*



Gambar 4. 92 *Flowchart* halaman *filter* populer tahun ini

2) *Flowgraph*



Gambar 4. 93 *Flowgraph* halaman *filter* populer tahun ini

Berdasarkan gambar 4. 93 diatas, dilakukan perhitungan sebagai berikut:

(1) Menghitung *cyclomatic complexity* $V(G)$ pada *edge* dan *node*.

$$\text{Pada rumus : } V(G) = E - N + 2$$

$$E (\text{edge}) = 6$$

$$N (\text{node}) = 7$$

$$P (\text{Predikat } \textit{node}) = 0$$

Penyelesaian :

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

$$= 6 - 7 + 2$$

$$= 1$$

$$\text{Predikat (P)} = P + 1$$

$$= 0 + 1$$

$$= 1$$

(2) Berdasarkan perhitungan *cyclomatic complexity* dari *flowgraph*

diatas memiliki *region* = 1

(3) *Independent path* pada *flowgraph* yaitu:

$$\text{Path 1} = 1 - 2 - 3 - 4 - 5 - 6 - 7$$

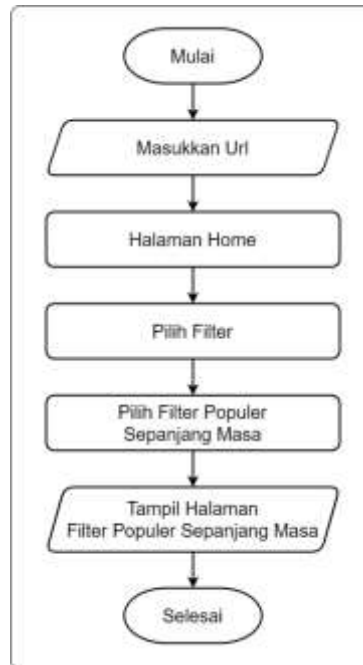
(4) Grafik matriks halaman *filter* populer tahun ini

Tabel 4. 35 Grafik matriks halaman *filter* populer tahun ini

	1	2	3	4	5	6	7	E-1
1		1						$1 - 1 = 0$
2			1					$1 - 1 = 0$
3				1				$1 - 1 = 0$
4					1			$1 - 1 = 0$
5						1		$1 - 1 = 0$
6							1	$1 - 1 = 0$
7								0
	SUM (E + 1)							$0 + 1 = 1$

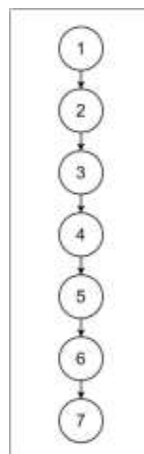
m. White box testing filter populer sepanjang masa

1) Flowchart



Gambar 4. 94 Flowchart halaman *filter* populer sepanjang masa

2) Flowgraph



Gambar 4. 95 Flowgraph halaman *filter* populer sepanjang masa

Berdasarkan gambar 4. 95 diatas, dilakukan perhitungan sebagai berikut:

(1) Menghitung *cyclomatic complexity* $V(G)$ pada *edge* dan *node*.

$$\text{Pada rumus : } V(G) = E - N + 2$$

$$E \text{ (edge)} = 6$$

$$N \text{ (node)} = 7$$

$$P \text{ (Predikat node)} = 0$$

Penyelesaian :

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

$$= 6 - 7 + 2$$

$$= 1$$

$$\text{Predikat (P)} = P + 1$$

$$= 0 + 1$$

$$= 1$$

(2) Berdasarkan perhitungan *cyclomatic complexity* dari *flowgraph*

diatas memiliki *region* = 1

(3) *Independent path* pada *flowgraph* yaitu:

$$\text{Path 1} = 1 - 2 - 3 - 4 - 5 - 6 - 7$$

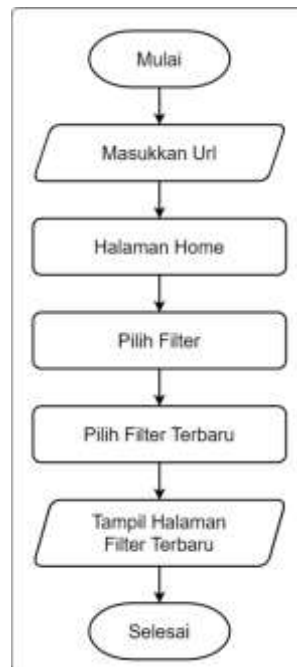
(4) Grafik matriks halaman *filter* populer sepanjang masa

Tabel 4. 36 Grafik matriks halaman *filter* populer sepanjang masa

	1	2	3	4	5	6	7	E-1
1		1						$1 - 1 = 0$
2			1					$1 - 1 = 0$
3				1				$1 - 1 = 0$
4					1			$1 - 1 = 0$
5						1		$1 - 1 = 0$
6							1	$1 - 1 = 0$
7								0
	SUM (E + 1)							$0 + 1 = 1$

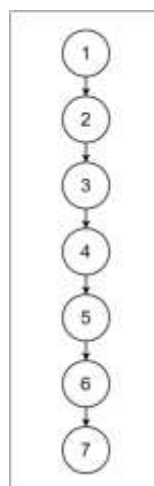
n. White box testing filter terbaru

1) Flowchart



Gambar 4. 96 Flowchart halaman *filter* terbaru

2) Flowgraph



Gambar 4. 97 Flowchart halaman *filter* terbaru

Berdasarkan gambar 4. 97 diatas, dilakukan perhitungan sebagai berikut:

(1) Menghitung *cyclomatic complexity* $V(G)$ pada *edge* dan *node*.

$$\text{Pada rumus : } V(G) = E - N + 2$$

$$E (\text{edge}) = 6$$

$$N (\text{node}) = 7$$

$$P (\text{Predikat } \textit{node}) = 0$$

Penyelesaian :

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

$$= 6 - 7 + 2$$

$$= 1$$

$$\text{Predikat (P)} = P + 1$$

$$= 0 + 1$$

$$= 1$$

(2) Berdasarkan perhitungan *cyclomatic complexity* dari *flowgraph*

diatas memiliki *region* = 1

(3) *Independent path* pada *flowgraph* yaitu:

$$\text{Path 1} = 1 - 2 - 3 - 4 - 5 - 6 - 7$$

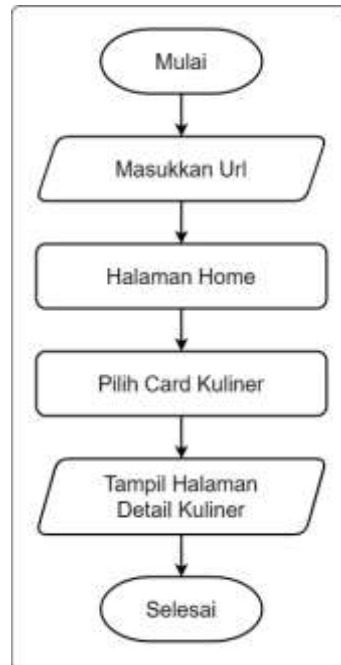
(4) Grafik matriks halaman *filter* terbaru

Tabel 4. 37 Grafik matriks halaman *filter* terbaru

	1	2	3	4	5	6	7	E-1
1		1						$1 - 1 = 0$
2			1					$1 - 1 = 0$
3				1				$1 - 1 = 0$
4					1			$1 - 1 = 0$
5						1		$1 - 1 = 0$
6							1	$1 - 1 = 0$
7								0
	SUM (E + 1)							$0 + 1 = 1$

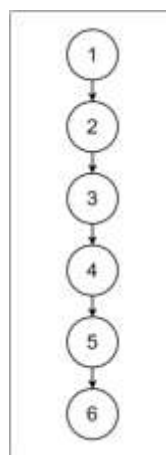
o. *White box testing* halaman detail

1) *Flowchart*



Gambar 4. 98 *Flowchart* halaman detail kuliner

2) *Flowgraph*

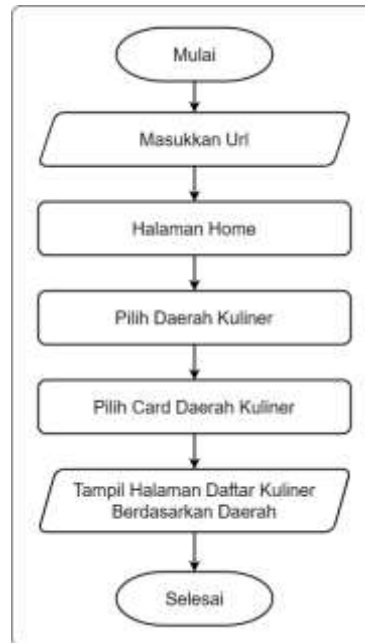


Gambar 4. 99 *Flowgraph* halaman detail kuliner

Berdasarkan gambar 4. 99 diatas, dilakukan perhitungan sebagai berikut:

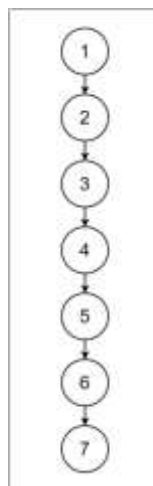
p. *White box testing* pilih daerah kuliner

1) *Flowchart*



Gambar 4. 100 *Flowchart* pilih daerah kuliner

2) *Flowgraph*



Gambar 4. 101 *Flowgraph* pilih daerah kuliner

Berdasarkan gambar 4. 101 diatas, dilakukan perhitungan sebagai berikut:

(1) Menghitung *cyclomatic complexity* $V(G)$ pada *edge* dan *node*.

$$\text{Pada rumus : } V(G) = E - N + 2$$

$$E \text{ (edge)} = 6$$

$$N \text{ (node)} = 7$$

$$P \text{ (Predikat node)} = 0$$

Penyelesaian :

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

$$= 6 - 7 + 2$$

$$= 1$$

$$\text{Predikat (P)} = P + 1$$

$$= 0 + 1$$

$$= 1$$

(2) Berdasarkan perhitungan *cyclomatic complexity* dari *flowgraph*

diatas memiliki *region* = 1

(3) *Independent path* pada *flowgraph* yaitu:

$$\text{Path 1} = 1 - 2 - 3 - 4 - 5 - 6 - 7$$

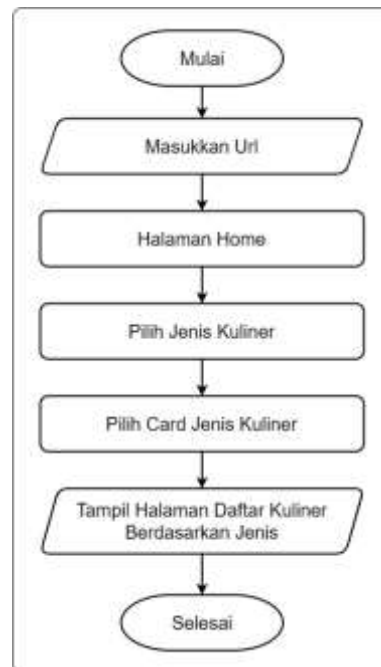
(4) Grafik matriks pilih daerah kuliner

Tabel 4. 39 Grafik matriks pilih daerah kuliner

	1	2	3	4	5	6	7	E-1
1		1						$1 - 1 = 0$
2			1					$1 - 1 = 0$
3				1				$1 - 1 = 0$
4					1			$1 - 1 = 0$
5						1		$1 - 1 = 0$
6							1	$1 - 1 = 0$
7								0
	SUM (E + 1)							$0 + 1 = 1$

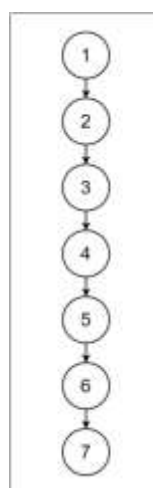
q. *White box testing* pilih jenis kuliner

1) *Flowchart*



Gambar 4. 102 *Flowchart* pilih jenis kuliner

2) *Flowgraph*



Gambar 4. 103 *Flowgraph* pilih jenis kuliner

Berdasarkan gambar 4. 103 diatas, dilakukan perhitungan sebagai berikut:

(1) Menghitung *cyclomatic complexity* $V(G)$ pada *edge* dan *node*.

$$\text{Pada rumus : } V(G) = E - N + 2$$

$$E (\text{edge}) = 6$$

$$N (\text{node}) = 7$$

$$P (\text{Predikat } \textit{node}) = 0$$

Penyelesaian :

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

$$= 6 - 7 + 2$$

$$= 1$$

$$\text{Predikat (P)} = P + 1$$

$$= 0 + 1$$

$$= 1$$

(2) Berdasarkan perhitungan *cyclomatic complexity* dari *flowgraph*

diatas memiliki *region* = 1

(3) *Independent path* pada *flowgraph* yaitu:

$$\text{Path 1} = 1 - 2 - 3 - 4 - 5 - 6 - 7$$

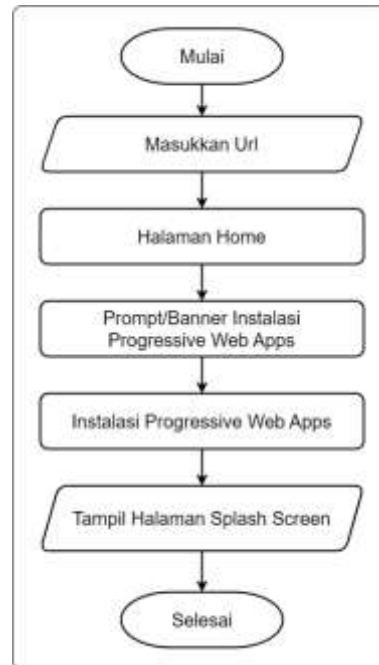
(4) Grafik matriks pilih jenis kuliner

Tabel 4. 40 Grafik matriks pilih jenis kuliner

	1	2	3	4	5	6	7	E-1
1		1						$1 - 1 = 0$
2			1					$1 - 1 = 0$
3				1				$1 - 1 = 0$
4					1			$1 - 1 = 0$
5						1		$1 - 1 = 0$
6							1	$1 - 1 = 0$
7								0
	SUM (E + 1)							$0 + 1 = 1$

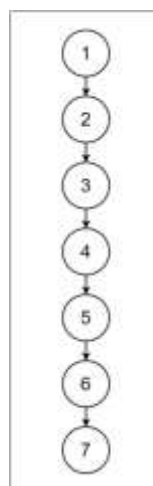
r. *White box testing instalasi progressive web apps*

1) *Flowchart*



Gambar 4. 104 *Flowchart instalasi progressive web apps*

2) *Flowgraph*



Gambar 4. 105 *Flowgraph instalasi progressive web apps*

Berdasarkan gambar 4. 105 diatas, dilakukan perhitungan sebagai berikut:

(1) Menghitung *cyclomatic complexity* $V(G)$ pada *edge* dan *node*.

$$\text{Pada rumus : } V(G) = E - N + 2$$

$$E \text{ (edge)} = 6$$

$$N \text{ (node)} = 7$$

$$P \text{ (Predikat node)} = 0$$

Penyelesaian :

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

$$= 6 - 7 + 2$$

$$= 1$$

$$\text{Predikat (P)} = P + 1$$

$$= 0 + 1$$

$$= 1$$

(2) Berdasarkan perhitungan *cyclomatic complexity* dari *flowgraph*

diatas memiliki *region* = 1

(3) *Independent path* pada *flowgraph* yaitu:

$$\text{Path 1} = 1 - 2 - 3 - 4 - 5 - 6 - 7$$

(4) Grafik matriks instalasi *progressive web apps*

Tabel 4. 41 Grafik matriks instalasi *progressive web apps*

	1	2	3	4	5	6	7	E-1
1		1						$1 - 1 = 0$
2			1					$1 - 1 = 0$
3				1				$1 - 1 = 0$
4					1			$1 - 1 = 0$
5						1		$1 - 1 = 0$
6							1	$1 - 1 = 0$
7								0
	SUM (E + 1)							$0 + 1 = 1$

BAB V

PENUTUP

A. Kesimpulan

Merujuk pada hasil kajian yang telah rampung, berikut ini adalah intisari yang dapat dirumuskan:

1. Dalam penelitian yang telah dilakukan, penulis berhasil mengembangkan sebuah sistem yang memudahkan pencarian informasi tentang Kuliner Nusantara.
2. Menggunakan teknologi *Progressive web apps* dalam pembuatan aplikasi *website* untuk memudahkan masyarakat dalam mengakses situs *web* tanpa perlu membuka *browser* atau mencari alamat situs, sehingga mempermudah pencarian informasi tentang Kuliner Nusantara.
3. Sistem ini dibangun dengan menggunakan *framework* Laravel untuk *backend* dan React untuk *frontend* yang diintegrasikan melalui Inertia.js. Visual Studio Code digunakan sebagai *text editor* dan MySQL sebagai sistem manajemen *database*.

B. Saran

Dalam proses studi ini, peneliti menyadari adanya sejumlah keterbatasan yang membutuhkan penyempurnaan dan elaborasi lebih lanjut. Oleh karena itu, peneliti mengajukan beberapa saran untuk kajian di masa mendatang, antara lain:

1. Mengembangkan aplikasi dengan opsi untuk mencetak atau mengunduh resep dalam format PDF atau format *file* lainnya untuk memudahkan pengguna dalam menyimpan dan menggunakan resep secara *offline*.
2. Menambahkan beberapa data atau melakukan pembaruan secara berkala terhadap basis data resep dengan menambahkan resep baru dari berbagai daerah di Indonesia.
3. Menyediakan opsi untuk menerjemahkan resep ke dalam bahasa lain untuk memudahkan pengguna yang tidak memahami bahasa Indonesia.

DAFTAR PUSTAKA

- Alan Hume, D. (2021). *Progressive web apps*. Manning Publications
- Ariyani, L., Sagita, S. M., & Mardiyati, S. (2020). *Perancangan Aplikasi Pengenalan Masakan Khas Daerah di Indonesia Berbasis Android*. *Jurnal Ilmiah Komputasi*, 18(4). <https://doi.org/10.32409/jikstik.18.4.2668>
- Ater, T. (2019). *Building Progressive web apps Bringing the Power of Native to the Browser* Tal Ater. O'Reilly Media
- Castro, E. (2022). *CSS: Visual Quickstart Guide* (5th ed.). Peachpit Press.
- Citizendium. (2020, April 9). *Online Encyclopedia Definition*. Diperoleh dari: https://en.citizendium.org/wiki/Online_encyclopedia/Definition. (Diakses 3 Desember 2023)
- Dewi, S. S. (2020). *Rekayasa Perangkat Lunak Resep Kuliner Nusantara Berbasis Android*. *Jurnal Riset Dan Aplikasi Mahasiswa Informatika (JRAMI)*, 1(03). <https://doi.org/10.30998/jrami.v1i03.363>
- Duckett, J. (2022). *CSS in Action*. Manning Publications.
- Hartl, M., & Donahoe, L. (2019). *Learn Enough HTML to Be Dangerous*. Learn Enough Society
- Hasyim, N., Senoprabowo, A., & Pramudya, E. R. (2022). *Pengembangan Konsep Game Edukasi Tentang Pengenalan Makanan Tradisional Indonesia kepada Anak-Anak Usia Dini*. *Prosiding Sains Nasional Dan Teknologi*, 12(1), 238. <https://doi.org/10.36499/psnst.v12i1.6891>
- Ikorasaki, F., Arwa, K., & Hrp, N. A. (2022). *Rancang Bangun Aplikasi Pengenalan Masakan Nusantara Berbasis Android*. *Blend Sains Jurnal Teknik*, 1(1). <https://doi.org/10.56211/blendsains.v1i1.78>
- McFarland, D. (2021). *JavaScript & jQuery: The Missing Manual, Second Edition* (Vol. 2). O'Reilly Media
- Meyer, E. A., & Weyl, E. (2020). *CSS: The Definitive Guide* (4th ed.). O'Reilly & Associates.
- Morgan, J. (2019). *Simplifying JavaScript: Writing Modern JavaScript with ES5, ES6, and Beyond*. O'Reilly Media
- Morgan, J. (2021). *How to code in React.js*. O'Reilly Media
- Nematrion, N. (2020). *HTML, CSS and JavaScript*. Nematrion