

## BAB II

### TINJAUAN PUSTAKA

#### A. Hasil Penelitian Terdahulu

Menurut Penelitian (Alexander & Husufa, 2020) banyak bisnis yang sudah berkembang, khususnya industri restoran, tidak semuanya mampu bertahan lama karena sebagian besar gagal atau bangkrut karena berbagai alasan, termasuk manajemen yang buruk. Salah satunya adalah Olive Cafe dimana transaksinya masih menggunakan cara manual. Hal ini sering kali mengakibatkan penghitungan jumlah pembayaran yang tidak akurat, kesulitan menemukan tanda terima saat klien melakukan pembayaran, dan memerlukan banyak waktu untuk memberikan laporan yang tepat. Selain transaksi yang dilakukan secara manual, pemesanan sewa Olive Cafe juga tidak dicatat dengan benar.

Hasil Penelitian dari (Dika et al., 2018) menjelaskan mengelola dan memantau inventaris produk, mendokumentasikan transaksi penjualan, menangani pembelian garmen, dan tantangan terkait bisnis lainnya merupakan beberapa tantangan yang dihadapi pemilik distro Handmad di Denpasar. Ia mengaku hal itu terjadi akibat pengoperasiannya masih dilakukan secara manual dengan komputer notebook atau program *Microsoft Excel*. Masalah-masalah ini memerlukan pengembangan aplikasi titik penjualan seluler yang akan digunakan secara eksklusif oleh distro untuk menyederhanakan prosedur penjualan mereka. Data tersebut akan langsung terhubung dengan platform web *dimans*.

Hasil penelitian (Kurniati, 2021) menjelaskan Beberapa toko dan pedagang terus melakukan semua transaksinya menggunakan metode kuno. hal ini berdampak pada seberapa baik toko beroperasi, terutama dalam hal layanan pelanggan. Selain itu, tidak mungkin memperoleh berbagai informasi yang diperlukan dengan benar, termasuk stok barang dan laporan keuangan. *POS* dapat memberikan pelayanan yang lebih baik kepada pelanggan, karena

pembeli bisa mendapatkan nota transaksi. Selain itu, laporan keuangan yang menyajikan data produk, persediaan, transaksi dan pembelian dapat diakses lebih cepat dan akurat dibandingkan dengan sistem manual.

Penelitian dari (Cahyodi & Arifin, 2017) menjelaskan *POS* dapat memberikan dukungan yang lebih baik kepada pembeli, misalnya menghitung harga dan jumlah barang yang dibeli dapat lebih cepat dan jumlah produk tidak lagi bergantung pada pencatatan manual.

Menurut penelitian (Hasim et al., 2022) *Progressive Web App* hanyalah web biasa pada umumnya namun dapat memanfaatkan fitur tambahan seperti *service worker* dan *file manifest* agar dapat terlihat seperti aplikasi asli pada umumnya. Penerapan *Progressive Web App* menguntungkan toko Aneka Snack dalam hal penjualan dan jangkauan pelanggan dengan memungkinkan pengguna mengakses sistem penjualan *online* melalui berbagai perangkat dan platform dengan waktu pemuatan yang cepat. Aplikasi dapat diakses tanpa perlu mengetikkan alamat *URL*, serta dapat membagikan notifikasi sehingga pelanggan dapat mengetahui data terkini tentang Aneka Snack. Selain itu, *service worker* dapat menangani permintaan ketika jaringan tidak ada, yang berarti aplikasi dapat diakses lebih cepat ketika *offline* karena informasi dimuat dalam *cache*. Secara keseluruhan, fitur-fitur tersebut dapat membantu meningkatkan penjualan dan menjangkau pelanggan yang lebih luas.

Berdasarkan hasil penelitian (A. Kurniawan et al., 2018) Keuntungan *PWA* dapat memuat konten seketika bahkan dalam kondisi jaringan yang tidak baik. Ketika digunakan dari layar awal pengguna, maka *service worker* diaktifkan pada *PWA* untuk dimuat seketika. Dalam sistem monitoring keluhan sampah di Kota Makassar, *Progressive Web Application* digunakan untuk memudahkan masyarakat dalam melaporkan keluhan sampah secara *online*. *Service worker* digunakan untuk melakukan proses *caching* data pada jaringan aktif dan tidak aktif sehingga data keluhan tetap dapat diakses meskipun dalam keadaan *offline*. Dalam sistem ini, Informasi yang diperoleh dari *API* kemudian ditampilkan dalam antar muka web. Selama jaringan bersifat aktif, *service*

*workers* menyelesaikan kewajibannya dalam melakukan sistem penyimpanan. Sejak saat itu, informasi yang disimpan dapat diakses pada jaringan tidak aktif.

Hasil penelitan dari (Wirawan & Santosa, 2019) menjelaskan Dengan menggunakan aplikasi *Point of Sales*, pengguna bisa mendapatkan beberapa keuntungan manfaat dengan nilai tambah yang dapat dikelola dalam bentuk peningkatan kualitas pelayanan, peningkatan bisnis, keunggulan kompetitif dan proses pengambilan keputusan. Pemanfaatan teknologi yang baik dapat menjadi solusi dalam pencatatan dan pemrosesan transaksi sehingga dilakukan oleh sistem yang terkomputerisasi, karena akan mempercepat transaksi proses dan memberikan hasil yang akurat sehingga lebih efektif dan efisien. Transaksi terkomputerisasi menyediakan kenyamanan dalam bekerja dan hasil yang pasti bagi pembeli atau pelanggan.

Penelitian (Ramos & Castro, 2017) menjelaskan sebagian besar perusahaan diindustri ritel sudah menerapkan teknologi sistem *point of sales (POS)* dengan niat untuk membuat layanan mereka lebih cepat dan mendapatkan kontrol lebih besar atas beragam aspek operasional, banyak dari bisnis kekurangan staf yang mempengaruhi pengawasan yang baik terhadap penggunaan sistem. Perlu diperhatikan juga biaya-biaya lain seperti yang bersifat internal biaya yang terkait dengan penjadwalan dan pembayaran personel restoran saat mereka berlatih. Sama dengan biaya yang terkait dengan komitmen manajerial terhadap proyek sistem *POS*.

Menurut penelitian (Aminudin et al., 2019) masalah dari penelitian tersebut adalah kurangnya kemudahan dalam mengakses dan menyimpan dokumen digital tugas akhir mahasiswa di Fakultas Teknik Universitas Muhammadiyah Malang. Solusi yang ditawarkan adalah dengan membangun sistem penyimpanan tugas akhir berbasis *PWA* dengan penyelidikan kontekstual yang diambil di ruang baca Tim Desain Universitas Muhammadiyah Malang. Sistem ini memiliki kemampuan *offline mode*, *responsive* yang dapat menyesuaikan perangkat pengguna, dan dapat menampilkan *splashscreen* saat aplikasi pertama kali dibuka.

Berdasarkan hasil penelitian (Amrullah et al., 2021) *PWA* dapat meningkatkan aplikasi berbasis web dengan memberikan keterlibatan terbaik dalam memanfaatkan aplikasi web dalam keadaan apa pun, walaupun jaringan internet web diabaikan atau terputus. Pada *PWA* terdapat inovasi terbaru dari program seperti *Service Workers* dan *Application Manifest* yang memungkinkan aplikasi web diakses melalui ikon di *home screen* ponsel seperti aplikasi lokal. *PWA* juga mendukung layar splash dan pesan *pop-up*. Pengguna dapat mengakses aplikasi lebih cepat dan mudah..

## B. Landasan Teori

### 1. Progressive Web App (PWA)

*Progressive Web App (PWA)* merupakan aplikasi web yang dirancang untuk memberikan pengalaman pengguna yang serupa dengan aplikasi mobile atau native. *PWA* memanfaatkan teknologi web modern untuk menggabungkan keuntungan aplikasi web dan aplikasi *mobile*. *PWA* sebenarnya hanyalah aplikasi web biasa yang memanfaatkan fitur browser tampak seperti aplikasi asli. *PWA* digambarkan sebagai kumpulan inovasi, ide rencana, dan antarmuka *Pemrograman WEB (API)* yang bekerja sama untuk memberikan aplikasi web seperti aplikasi *mobile* (A. Kurniawan et al., 2018). *PWA* bukanlah teknologi atau kerangka kerja baru, melainkan praktik terbaik yang telah diadopsi oleh web untuk memberikan perasaan seperti aplikasi asli kepada pengguna. (Zait et al., 2018)

### 2. Point Of Sales (POS)

*Point Of Sales (POS)* merupakan suatu kegiatan yang berhubungan dengan penjualan dan pendataan barang pada sektor bisnis retail (Salim et al., 2023). *Point Of Sales* mengatur transaksi dan siapa yang bertanggung jawab atas setiap transaksi penjualan. *POS* biasanya terdiri dari (Komputer, Printer Resi, Lemari Uang, Pemindai Identifikasi Standar) dan *software* sebagai (Administrasi Stok, Perincian, Penjualan, Manajemen Klien, Penanganan Pengembalian) dimana kedua bagian tersebut digunakan untuk setiap interaksi pertukaran (Marisa & Yuarita, 2017).

### 3. *NextJS*

*NextJS* adalah kerangka kerja *React* yang populer untuk pengembangan aplikasi web modern. Dengan menyediakan fitur seperti rendering sisi server, *pre-fetching*, dan sistem routing yang mudah digunakan, *NextJS* mempermudah pengembang dalam membangun aplikasi web yang responsif dan efisien. Ia memungkinkan penggunaan *ReactJS* dengan lebih baik untuk keperluan pengembangan web, membantu dalam mengelola *state* aplikasi, dan menyediakan performa yang optimal dengan penanganan aspek teknis seperti *server-side rendering* (Sasikumar et al., 2022).

Salah satu keunggulan *NextJS* adalah kemudahan untuk mengembangkan aplikasi web menggunakan *React*. Adanya *server-side rendering (SSR)* dan *static site generation (SSG)*, memungkinkan aplikasi beroperasi dengan kinerja yang lebih baik dan waktu muat yang lebih cepat. Selain itu, *NextJS* memiliki integrasi *API backend* yang baik dan kemampuan *pre-fetching*, sehingga meningkatkan efisiensi penggunaan sumber daya. Namun, *NextJS* juga memiliki keterbatasan salah satunya dalam konfigurasi yang sangat kompleks, terutama untuk proyek-proyek besar. Hal ini membutuhkan pemahaman yang lebih mendalam tentang konsep *SSR* dan *SSG* untuk penggunaan yang efektif.

### 4. *Service Worker*

*Service Worker* adalah skrip yang mengubah fondasi program klien dan tidak memerlukan interaksi antar klien untuk menyelesaikan tugasnya, sehingga akan terus berjalan meskipun halaman situs tidak terbuka (A. Kurniawan et al., 2018). Dengan menggunakan *Service Worker*, halaman web dapat tetap tersedia bahkan dalam mode *offline*, konten web yang telah dicache memungkinkan halaman untuk memuat lebih cepat, bahkan dalam kondisi jaringan yang tidak stabil. Ini adalah salah satu cara untuk meningkatkan pengalaman pengguna yang lebih baik dan responsif dalam berbagai situasi (Zait et al., 2018).

## 5. Website

Website adalah sekelompok halaman yang menampilkan berbagai bentuk informasi seperti gambar diam atau bergerak, text, animasi, suara, dan kombinasi dari semua itu. Informasi ini dapat bersifat *statis*, yang berarti tidak berubah secara otomatis, atau *dinamis*, yang dapat mengubah tampilan berdasarkan permintaan atau interaksi pengguna. Semua halaman ini membentuk suatu struktur yang terkait dan terhubung satu sama lain melalui tautan yang disebut *hyperlink*. Teks yang digunakan sebagai jembatan penghubung antara halaman-halaman ini disebut *hypertext* (Elektro & Medan, 2012).

## 6. Typescript

*TypeScript* adalah ekstensi dari *JavaScript* yang dirancang untuk mengatasi beberapa kekurangan yang ada dalam bahasa *JavaScript*. Dalam hal sintaksis, *TypeScript* adalah suatu superset dari *EcmaScript 5*, yang berarti setiap program *JavaScript* juga merupakan program *TypeScript*. Salah satu kelebihan utama dari *TypeScript* adalah kemampuannya untuk memperkaya *JavaScript* dengan sistem modul, kelas, antarmuka, dan tipe statis. Dengan kata lain, *TypeScript* memungkinkan pengguna untuk mengembangkan aplikasi dengan lebih terstruktur dan aman. Sistem modul dan sistem tipenya sangat fleksibel dan mudah digunakan, dan mereka mendukung banyak praktik pemrograman umum yang ditemukan dalam bahasa *JavaScript* (Bierman et al., 2014).

## 7. Web App Manifest

*Web App Manifest* adalah sebuah berkas metadata dalam format *JSON* yang digunakan dalam pengembangan web untuk mendefinisikan karakteristik dan pengaturan suatu aplikasi web, terutama dalam konteks *Progressive Web Apps (PWA)*. *Web App Manifest* memberikan informasi penting kepada *browser* dan perangkat *mobile* tentang bagaimana sebuah aplikasi web harus berperilaku ketika diakses dari perangkat. Tujuan dari file manifest adalah mengekspos hal seperti pengaturan yang dimodifikasi untuk pengembang aplikasi *mobile*. seperti gambar logo, nama aplikasi,

*splash screen* dan banyak lagi. Singkatnya, Web App manifest dapat mengubah tampilan gaya aplikasi *PWA* (Sharma et al., 2019).

#### 8. *TailwindCSS*

*Tailwindcss* merupakan *Framework CSS* yang dirancang mempermudah pengembangan tampilan antar muka pada situs web atau aplikasi web. *Tailwindcss* sangat populer di kalangan pengembang web karena pendekatan yang berbeda dan konsep yang kuat dalam mendesain dan mengatur gaya tampilan. *TailwindCSS* menyediakan kelas *utilitas* yang dapat digunakan untuk membangun keseluruhan desain yang diinginkan pengguna tanpanya harus keluar dari file *HTML* (Rifandi et al., 2022).

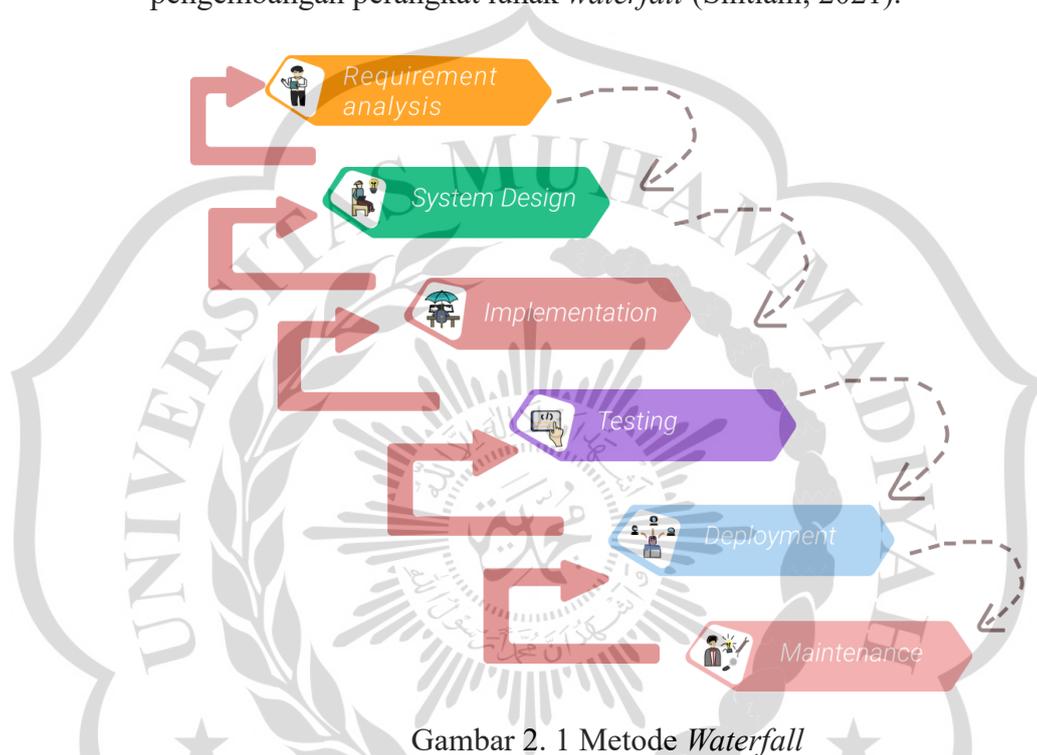
#### 9. *Single Page Application(SPA)*

*Single Page Application* adalah aplikasi web yang menyajikan konten kepada pengguna melalui satu halaman web tunggal, menghindari kebutuhan untuk memuat ulang seluruh halaman saat pengguna berpindah ke bagian yang berbeda. Dengan menggunakan teknik *AJAX* (*Asynchronous JavaScript and XML*), *SPA* memperbarui kontennya secara *dinamis* tanpa menyegarkan halaman, memberikan pengalaman pengguna yang lebih responsif dan mirip dengan aplikasi desktop. Teknologi seperti *React*, *Angular*, dan *Vue* sering digunakan untuk membangun *SPA*, memungkinkan pengembang untuk mengelola state aplikasi dengan lebih efisien dan memberikan navigasi yang lebih halus antar bagian tanpa perlu mengganti halaman secara keseluruhan. Keunggulan lainnya termasuk waktu pemuatan yang lebih cepat dan responsivitas yang meningkat, membuat *SPA* menjadi pilihan populer dalam pengembangan aplikasi web modern (Belluano, 2018).

#### 10. *Waterfall*

Metode *waterfall* adalah pendekatan pengembangan perangkat lunak yang bersifat linear dan berurutan. Metode ini menggambarkan proses pengembangan perangkat lunak sebagai serangkaian tahapan yang harus dilalui secara berurutan (H. Kurniawan et al., 2021). *Waterfall*

dianggap sebagai pendekatan yang kaku karena setiap tahap harus selesai sebelum memasuki tahap berikutnya. Meskipun demikian, model ini dapat berguna dalam proyek-proyek di mana kebutuhan dapat ditetapkan dengan jelas di awal, dan perubahan kebutuhan tidak terlalu sering muncul selama pengembangan (Pinontoan et al., 2019), berikut gambar 2. 1 metode pengembangan perangkat lunak *waterfall* (Sintiani, 2021).



Gambar 2. 1 Metode *Waterfall*

### 11. *Server Side Rendering*

*Server-Side Rendering* pada NextJS adalah metode rendering di mana halaman web dihasilkan di sisi server sebelum dikirimkan ke browser. Dengan *SSR*, konten utama termasuk data dinamis, diambil dan di-render di server, memungkinkan halaman untuk memiliki *SEO* yang lebih baik dan performa yang lebih cepat, karena pengguna menerima halaman yang sudah jadi dan siap ditampilkan sejak awal. Data yang diekspos melalui *API* dapat digunakan client untuk mendapatkan dan menampilkannya pada halaman web (Herman & Geovanny, 2022)

## 12. Client Side Rendering

*Client-side rendering* merupakan situs web yang dirender bukan di server melainkan di render browser menggunakan javascript, jadi browser klien hanya dapat menampilkan konten dari dokumen *HTML*, *CSS* dan *JS* yang akan dirender (Herman & Geovanny, 2022). Halaman web sebagian besar dibangun di sisi klien (*browser*) menggunakan *JavaScript* setelah halaman *HTML* awal diunduh. Ini berarti bahwa sebagian besar logika rendering dan data diambil oleh *JavaScript* di sisi klien.

## 13. PostgreSQL

*PostgreSQL* adalah sistem manajemen basis data (*DBMS*) yang kuat dan canggih yang bersifat sumber terbuka (*open source*). salah satu sistem manajemen basis data relasional (*RDBMS*) yang paling populer dan canggih yang tersedia secara bebas untuk digunakan oleh komunitas pengembang di seluruh dunia. Perbedaan utama antara *Postgres* dan Database Lainnya adalah teknologi *Postgres* yang memungkinkan pengguna untuk mengkarakterisasi *SQL* mereka sendiri, terutama saat membuat *function*. *Postgres* menyimpan informasi terkait tabel dan kolom serta tipe, fungsi, metode akses, dan banyak lagi. Hal ini memungkinkan semuanya dikumpulkan sebagai kelas yang dapat diubah oleh klien (Sugiana & Wiryana, 2001).