

## BAB II

### TINJAUAN PUSTAKA

#### 2.1 Aplikasi

Aplikasi ialah sebuah sistem atau perangkat lunak yang bangun untuk menjalankan proses tertentu pada sebuah perangkat elektronik, seperti komputer, *smartphone*, atau tablet. kata ini berasal dari bahasa Inggris *application software* yang berarti perangkat lunak aplikasi. Dalam konteks teknologi saat ini, aplikasi memiliki berbagai jenis dari berbagai sektor menyesuaikan kebutuhan. Fungsi utamanya adalah untuk mempermudah kehidupan manusia. Fungsinya juga dapat dibedakan berdasarkan tujuan penggunaannya, mulai dari aplikasi produktivitas, hiburan, hingga aplikasi pendidikan (Al Fatih, 2024). Jenis-jenis aplikasi diantaranya :

a. **Aplikasi *Mobile***

Jenis aplikasi ini dirancang khusus untuk perangkat *mobile*, seperti *Android* dan *IOS*. Contoh aplikasi *mobile* termasuk media sosial, permainan, dan alat produktivitas.

b. **Aplikasi *Web***

Aplikasi ini berjalan di *browser* dan tidak perlu diunduh. Contoh aplikasi *web* adalah platform *e-commerce* dan alat kolaborasi *online*.

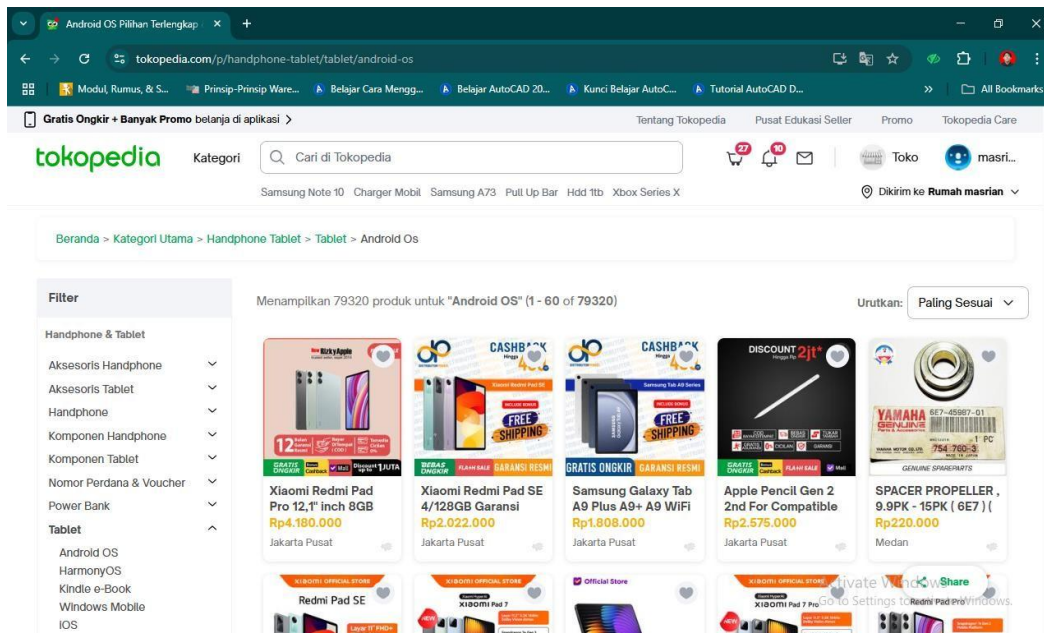
c. **Aplikasi *Desktop***

Dirancang untuk digunakan pada komputer, aplikasi desktop memiliki fitur lebih kompleks seperti *software* desain grafis dan pengolahan data.

Beberapa aplikasi populer saat ini adalah *WhatsApp*, *Instagram*, *Google Drive*, dan *Zoom*. Aplikasi-aplikasi ini sangat membantu dalam hal komunikasi, penyimpanan data, dan kolaborasi jarak jauh (Al Fatih, 2024)

#### 2.2 Aplikasi *Web*

Aplikasi *web* adalah program yang berjalan pada *server* dan diakses melalui jaringan internet melalui *browser* seperti *Google Chrome*, *Mozilla Firefox*, atau *Safari*. Dengan kata lain, aplikasi berbasis *web* adalah aplikasi yang tidak perlu diunduh ke perangkat pengguna, melainkan cukup diakses melalui URL (*Uniform Resource Locator*) atau alamat *web* tertentu. Biasanya, aplikasi ini memanfaatkan teknologi seperti HTML, CSS, dan *JavaScript* untuk tampilan dan fungsionalitas, serta *database* untuk menyimpan data (Prasetyo, 2024). Salah satu contoh aplikasi web dapat dilihat pada gambar 2.1



Gambar 2.1 Aplikasi Web

Aplikasi berbasis *website* sangat bermanfaat karena pengguna tidak perlu memasang *software* tambahan dan cukup membuka *browser* untuk mengaksesnya. Berbeda dengan aplikasi *desktop* atau *mobile* yang diinstal di perangkat, aplikasi berbasis *web* memungkinkan penghematan ruang penyimpanan dan pembaruan otomatis. Dalam pengembangannya, aplikasi *web* sering kali dibangun menggunakan *framework web* seperti *Laravel*, *Django*, atau *Ruby on Rails* yang membantu mempercepat proses *coding* dan mempermudah pengelolaan kode (Prasetyo, 2024)

## 2.3 Perancangan Sistem

Perancangan sistem merupakan rencana atau pendekatan yang digunakan untuk mengatasi masalah yang ada dengan tujuan mencapai solusi optimal serta mencapai tujuan tertentu (Rusmawan, 2019). Ini mencakup pemilihan strategi dalam merancang atau mengatur sistem secara optimal untuk memenuhi kebutuhan khusus dan mencapai tujuan yang diharapkan dengan efisiensi dan efektivitas. Pada fase desain perancangan sistem, akan disusun suatu bagan yang memvisualisasikan prosedur-prosedur dari sistem yang sedang dirancang menggunakan *Data Flow Diagram* (DFD) dan *flowchart*.

### 2.3.1 Data Flow Diagram (DFD)

*Data Flow Diagram* (DFD) adalah alat pemodelan yang digunakan untuk menggambarkan alur data dalam suatu sistem (Adani, 2021). Konsep DFD yaitu teknik visual yang diterapkan dalam analisis sistem untuk mengilustrasikan

perjalanan data dari satu proses ke proses lainnya, serta cara data tersebut diolah oleh sistem (Patria, 2023). *Data Flow Diagram* (DFD) terdiri dari sejumlah simbol yang menggambarkan entitas yaitu pengguna atau sistem eksternal, proses yaitu tugas atau aktivitas, aliran data dan penyimpanan data (Patria, 2023). Simbol DFD yang umum digunakan dapat dilihat pada gambar 2.2

| Keterangan                 | DeMarco and Yourdan<br>Simbol                                                       | Gane and Sarson<br>Simbol                                                             |
|----------------------------|-------------------------------------------------------------------------------------|---------------------------------------------------------------------------------------|
| Entitas Luar               |    |    |
| Proses                     |    |    |
| Aliran data<br>(data flow) |  |  |
| Simpan data                |  |  |

Gambar 2.2 Simbol-Simbol *Data Flow Diagram*

a. **Entitas Eksternal (Entitas Luar)**

Entitas eksternal dalam DFD adalah sumber data eksternal atau tujuan data eksternal yang berinteraksi dengan sistem yang sedang dianalisis. Entitas eksternal bisa berupa pengguna, sistem lain, atau perangkat lain yang terlibat dalam alur data sistem. Penggunaan entitas eksternal membantu membatasi ruang lingkup sistem yang sedang dianalisis dan membantu pemahaman tentang bagaimana data berinteraksi dengan dunia luar.

b. **Proses**

Dalam DFD, proses merupakan elemen penting yang merepresentasikan aktivitas atau tugas yang dilakukan pada data untuk mengubahnya menjadi informasi yang lebih berguna. Proses seringkali diberi nomor untuk mengidentifikasi urutan jalannya aliran data. Contoh dari proses ini bisa berupa perhitungan matematis, validasi data, pengolahan informasi, dan sebagainya. Proses ini menjadi “jantung” dari DFD karena mewakili bagaimana data mengalir dan diolah di dalam sistem.

c. **Aliran Data (*Data Flow*)**

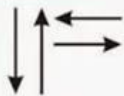
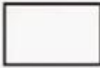
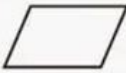
Aliran data dalam DFD digambarkan dengan garis berpanah yang menghubungkan proses, entitas eksternal, dan penyimpanan data. Aliran data menandakan pergerakan data dari satu proses ke proses lainnya atau dari proses ke penyimpanan data, serta dari entitas eksternal ke proses atau sebaliknya. Aliran data bisa berupa masukan ke proses atau keluaran dari proses, sehingga menggambarkan bagaimana data bergerak dan diproses dalam sistem.

d. **Penyimpanan data (*Arsip*)**

Penyimpanan data dalam DFD adalah elemen yang merepresentasikan tempat atau lokasi penyimpanan data dalam sistem, seperti basis data, berkas, atau memori. Penyimpanan data adalah wadah disimpannya data dan bisa diambil kembali pada waktu yang dibutuhkan. Dengan adanya simbol penyimpanan data dalam DFD, kita bisa mengenali dan memahami bagaimana data diolah dan atur dalam sistem, serta bagaimana data ini turut andil pada proses dan alur data yang ada.

### 2.3.2 *Flowchart*

*Flowchart* atau bagan alur adalah diagram yang menampilkan langkah-langkah dan keputusan untuk melakukan sebuah proses dari suatu program. Setiap langkah digambarkan dalam bentuk diagram dan dihubungkan dengan garis atau arah panah. *Flowchart* membantu dalam memahami, mendokumentasikan, merencanakan, dan mengkomunikasikan proses yang kompleks menjadi diagram yang lebih mudah dipahami oleh berbagai pihak, baik teknis maupun non-teknis (Telkomsel, 2023). Simbol *flowchart* bisa dilihat pada gambar 2.3

|                                                                                     |                                                                                                                                                                           |                                                                                      |                                                                                                                                     |
|-------------------------------------------------------------------------------------|---------------------------------------------------------------------------------------------------------------------------------------------------------------------------|--------------------------------------------------------------------------------------|-------------------------------------------------------------------------------------------------------------------------------------|
|    | <b>Flow Direction symbol</b><br>Yaitu simbol yang digunakan untuk menghubungkan antara simbol yang satu dengan simbol yang lain. Simbol ini disebut juga connecting line. |    | <b>Simbol Manual input</b><br>Simbol untuk memasukkan data secara manual on-line keyboard                                           |
|    | <b>Terminator Symbol</b><br>Yaitu simbol untuk permulaan (start) atau akhir (stop) dari suatu kegiatan                                                                    |    | <b>Simbol Preparation</b><br>Simbol untuk mempersiapkan penyimpanan yang akan digunakan sebagai tempat pengolahan di dalam storage. |
|    | <b>Connector Symbol</b><br>Yaitu simbol untuk keluar - masuk atau penyambungan proses dalam lembar / halaman yang sama.                                                   |    | <b>Simbol Predefine Proses</b><br>Simbol untuk pelaksanaan suatu bagian (sub-program)/prosedure                                     |
|    | <b>Connector Symbol</b><br>Yaitu simbol untuk keluar - masuk atau penyambungan proses pada lembar / halaman yang berbeda.                                                 |    | <b>Simbol Display</b><br>Simbol yang menyatakan peralatan output yang digunakan yaitu layar, plotter, printer dan sebagainya.       |
|    | <b>Processing Symbol</b><br>Simbol yang menunjukkan pengolahan yang dilakukan oleh komputer                                                                               |    | <b>Simbol disk and On-line Storage</b><br>Simbol yang menyatakan input yang berasal dari disk atau disimpan ke disk.                |
|    | <b>Simbol Manual Operation</b><br>Simbol yang menunjukkan pengolahan yang tidak dilakukan oleh computer                                                                   |    | <b>Simbol magnetik tape Unit</b><br>Simbol yang menyatakan input berasal dari pita magnetik atau output disimpan ke pita magnetik.  |
|  | <b>Simbol Decision</b><br>Simbol pemilihan proses berdasarkan kondisi yang ada.                                                                                           |  | <b>Simbol Punch Card</b><br>Simbol yang menyatakan bahwa input berasal dari kartu atau output ditulis ke kartu                      |
|  | <b>Simbol Input-Output</b><br>Simbol yang menyatakan proses input dan output tanpa tergantung dengan jenis peralatannya                                                   |  | <b>Simbol Dokumen</b><br>Simbol yang menyatakan input berasal dari dokumen dalam bentuk kertas atau output dicetak ke kertas.       |

Gambar 2.3 Simbol dan penjelasan *Flow Chart*

2.4 Metode *Agile*

Metode *Agile* adalah sebuah metode dalam pengembangan perangkat lunak dengan proses pengerjaan berulang yang memiliki aturan dan jalan keluar yang disetujui. *Agile* adalah metodologi perancangan sistem yang bersifat iteratif dengan lebih berfokus pada kolaborasi (Hendrawan dkk., 2020). Metode *Agile* sangat cocok untuk proyek jangka pendek. Dikarenakan metode ini dapat dengan mudah menyesuaikan kondisi apabila terjadi perubahan atau permasalahan dalam sebuah proyek. Dalam *Agile development*, nilai terpentingnya terdapat pada tim yang bisa memutuskan suatu hal dengan cepat dan tepat. Keputusan yang diambil tidak hanya cepat dari segi waktu, namun juga memiliki kualitas prediksi yang baik. Sehingga keputusan tersebut dapat mengatasi permasalahan yang ada tanpa menimbulkan masalah baru (Binar, 2024)

Konsep *Agile* muncul sebagai respon terhadap keterbatasan metode tradisional, seperti *waterfall*, yang sering kali dianggap kaku dan tidak mampu

beradaptasi dengan cepat terhadap perubahan kebutuhan dan kondisi pasar (Farras dkk., 2025). Pada dasarnya, metode *Agile workflow* adalah alternatif dari metode *waterfall* yang sudah ada sebelumnya. Metode *waterfall* sendiri bersifat linear sehingga tidak memungkinkan adanya perubahan di tengah proses pengembangan. Hal ini tentu mengkhawatirkan melihat situasi pasar yang dinamis. Proses *Agile development* ini mengacu pada konsep yang digagas oleh *Agile Manifesto*. Konsep tersebut diciptakan oleh 14 tokoh penting dalam dunia *software development*.

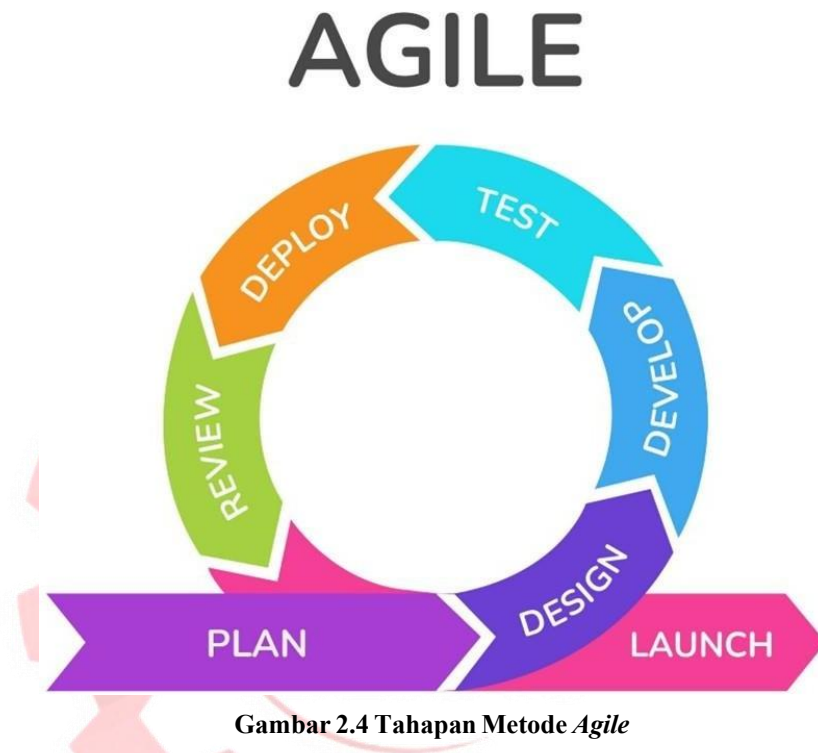
#### 2.4.1 Tahapan Metode *Agile*

Menurut (Codingstudio, 2024) tahapan metode *Agile* dirancang untuk meningkatkan kolaborasi dan efisiensi. Beberapa tahapan metode *Agile* mencakup:

- a. **Perencanaan**  
Tahapan ini berupa pemahaman tujuan proyek identifikasi kebutuhan dan penentuan prioritas fitur-fitur yang akan di kembangkan.
- b. **Desain**  
Tim pembuatan prototipe dan desain untuk fitur yang telah direncanakan. Ini termasuk pembuatan *wireframes*, *mockups* dan arsitektur sistem.
- c. **Pengembangan**  
Tim pengembangan menulis kode untuk mengimplementasikan fitur-fitur yang telah dirancang dan desain. Pengembangan ini dilakukan dalam iterasi singkat yang disebut sprint.
- d. **Pengujian**  
Memastikan fiturnya bekerja seperti harapan. Ini melibatkan pengujian unit, integrasi dan fungsional.
- e. **Peluncuran I**  
Peluncuran I disebut juga dengan *deployment*. Pada tahap ini, mereka menerapkan fitur yang telah diuji ke lingkungan produksi.
- f. **Evaluasi**  
Meninjau fitur yang telah diluncurkan dengan *stakeholder*, untuk mendapatkan umpan balik dan melakukan penyesuaian.
- g. **Peluncuran II**  
Peluncuran II atau *Launch* merupakan tahapan di mana fitur baru yang secara resmi diluncurkan dan tersedia untuk pengguna akhir.
- h. **Pemeliharaan**  
Setelah peluncuran, tim memantau fitur dan terus melakukan perbaikan atau peningkatan yang sesuai dengan kebutuhan.

Tahapan-tahapan tersebut dikerjakan secara berulang di setiap iterasi *Agile*. Setiap *sprint* biasanya berlangsung antara satu sampai empat minggu. Dalam setiap

sprint, tim *Agile* bekerja untuk menghasilkan bagian yang dapat berfungsi yang kemudian ditinjau dan dinilai oleh *stakeholder* (Codingstudio, 2024). Penggambaran Tahapan metode *Agile* dapat dilihat pada gambar 2.4



Gambar 2.4 Tahapan Metode *Agile*

*Agile workflow* mengacu pada manajemen dan penyelesaian proyek yang menekankan fleksibilitas, kolaborasi, dan peningkatan secara berkelanjutan. Hal ini didasarkan pada *agile software development methodology* yang ditandai dengan perencanaan yang adaptif dan penerapan yang iteratif serta berkembang secara sedikit demi sedikit. Tidak hanya itu, metode yang ada di dalam *Agile workflow* juga dapat dimanfaatkan sesuai kebutuhan dan urgensi yang dicocokkan dengan fungsinya. Hal ini membuat metode satu ini dinilai fleksibel dan efisien dalam beradaptasi dengan adanya perubahan yang cepat (Terralogiq, 2023).

Dengan menggunakan *Agile workflow*, biasanya proses pengembangan *software* dijalankan dengan durasi waktu yang berbeda-beda sesuai dengan tingkat kesulitan, besarnya pekerjaan, fitur yang dikembangkan, hingga metode yang digunakan. Namun, proses pengembangan *software* yang menerapkan *Agile* akan memakan waktu 1 sampai 4 minggu pada umumnya. Salah satu keuntungan dari menggunakan *Agile workflow* adalah dapat mengembangkan fitur-fitur terbaru dalam *software* secara cepat mengikuti perubahan situasi maupun kondisi serta kebutuhan dari pengguna (Terralogiq, 2023).

Salah satu contohnya adalah pada saat Anda sedang mengembangkan aplikasi berbasis *video-sharing*. Lalu, popularitas TikTok meningkat dan pengguna *smartphone* menjadi hanya menghabiskan waktu satu menit untuk menonton sebuah video. Melalui *Agile workflow* ini, Anda dapat beradaptasi menggunakan *insight* yang didapatkan dari situasi tersebut sehingga menambahkan fitur video pendek untuk aplikasi yang sedang dikembangkan. Melalui *Agile workflow* juga, proses pengembangan *software* menjadi lebih fleksibel dan efisien karena aplikasi tersebut sewaktu-waktu dapat diperbaharui dengan fitur-fitur terbaru yang sesuai dengan kondisi pasar dan kebutuhan pada setiap iterasinya. (Terralogiq, 2023)

## 2.5 Framework Laravel

*Framework Laravel* merupakan salah satu kerangka kerja berbasis PHP yang dirancang untuk membangun aplikasi *web* dengan lebih cepat dan efisien. *Laravel* menggunakan konsep *Model-View-Controller* (MVC) untuk memisahkan logika aplikasi, tampilan, dan kontrol data, sehingga memudahkan pengembang dalam menyusun dan memelihara kode. Dengan sintaksis yang elegan dan terstruktur, *Laravel* menyediakan berbagai fitur modern seperti *routing*, manajemen basis data dengan *Eloquent ORM*, serta pengelolaan *middleware* (Syaviqi & Bhakti, 2024).

### 2.5.1 Fitur utama Laravel

#### a. *Eloquent ORM*

*Eloquent ORM* adalah sistem ORM (*Object-Relational Mapping*) milik *Laravel* yang kuat dan mudah digunakan. *Eloquent* memungkinkan pengembang untuk berinteraksi dengan basis data menggunakan sintaksis yang elegan dan intuitif.

#### b. *Routing*

*Laravel* menyediakan sistem *routing* yang fleksibel dan mudah digunakan. Dengan *Laravel*, pengembang dapat dengan mudah menentukan rute untuk aplikasi mereka menggunakan metode yang sederhana dan mudah dibaca.

#### c. *Blade Templating Engine*

*Blade* adalah mesin *templating* *Laravel* yang ringan, tapi tetap kuat. *Blade* memungkinkan pengembang untuk menggunakan sintaksis sederhana untuk membuat tampilan dinamis dan dapat digunakan kembali.

#### d. *Middleware*

*Middleware* di *Laravel* memungkinkan pengembang untuk memfilter HTTP *request* yang masuk ke aplikasi. *Middleware* sangat berguna untuk tugas-tugas seperti otentikasi pengguna dan validasi input.

#### e. *Artisan CLI*

*Artisan* adalah *Command Line Interface* (CLI) milik *Laravel* yang menawarkan berbagai perintah bawaan untuk mempermudah pengembangan. Dengan *Artisan*, pengembang dapat membuat kontroler, model, dan komponen lainnya hanya dengan beberapa perintah.

f. ***Migration dan Seeders***

*Laravel* menyediakan sistem migrasi dan *seeder* yang kuat untuk mengelola skema basis data. Pengembang dapat dengan mudah membuat, mengubah, dan menghapus tabel serta mengisi tabel dengan data *dummy* untuk pengujian

### 2.5.2 Arsitektur MVC Pada *Laravel*

Arsitektur MVC (Model-View-Controller) adalah pola desain yang memisahkan aplikasi menjadi tiga komponen utama *model*, *view*, dan *controller* (MVC).

a. ***Model***

Komponen *model* bertanggung jawab untuk berinteraksi dengan basis data. Di *Laravel*, *Eloquent ORM* digunakan untuk mengelola operasi basis data. *Model* berisi logika bisnis dan aturan validasi data.

b. ***View***

Komponen *view* bertanggung jawab untuk menampilkan data kepada pengguna. *View* menerima data dari *controller* dan merendernya menjadi tampilan yang dapat dilihat oleh pengguna. *Laravel* menggunakan *Blade* sebagai mesin templating untuk membuat *views* yang dinamis dan mudah dikelola.

c. ***Controller***

Komponen *controller* bertindak sebagai perantara antara Model dan View. *Controller* menerima *input* dari pengguna, memprosesnya (dengan bantuan *model* jika diperlukan), dan menentukan *view* yang sesuai untuk ditampilkan. *Controller* juga mengelola logika aplikasi yang tidak terkait langsung dengan *model* atau *view*.

Arsitektur MVC membantu dalam memisahkan tanggung jawab di dalam aplikasi, membuat kode lebih terstruktur, mudah dipelihara, dan *scalable*. Dengan menggunakan MVC, pengembang dapat bekerja secara terpisah pada bagian-bagian yang berbeda dari aplikasi tanpa saling mengganggu sehingga meningkatkan efisiensi pengembangan.

## 2.6 MySQL

*MySQL* merupakan sistem manajemen *database* yang bersifat *open-source* yang menggunakan perintah dasar atau bahasa pemrograman yang berupa *Structured Query Language* (SQL) yang cukup populer di dunia teknologi. *MySQL* berguna sebagai *database*. *MySQL* memiliki dua bentuk lisensi, yaitu *free software* dan *shareware* atau perangkat lunak bermilik yang penggunaannya terbatas. Jadi, *MySQL* adalah *database server* gratis dengan lisensi GNU *General Public License* (GPL) yang bisa dipakai untuk keperluan pribadi atau komersil tanpa harus membayar lisensi yang ada (Biznetgio, 2022).

Dalam istilah pemrograman, SQL sendiri menjadi bahasa yang dipakai di dalam pengambilan data pada *relational database* atau *database* yang terstruktur. Dengan kata lain, *MySQL* merupakan *database management system* yang menggunakan bahasa SQL sebagai bahasa penghubung antara perangkat lunak aplikasi dengan *database server* (Biznetgio, 2022).

## 2.7 Pedagang kaki lima

Pedagang kaki lima merupakan salah satu lini usaha dagang yang melakukan kegiatan penjualan di daerah yang diperuntukkan untuk pejalan kaki atau bahu jalan. Beberapa orang menyebut atau mengelompokkan PKL untuk pedagang yang menggunakan gerobak. Istilah itu sering dimaksudkan karena total kaki pedagangnya berjumlah lima. Lima kaki yang dimaksud ialah dua kaki pedagang dan tiga "kaki" yang ada pada gerobak yang sebenarnya adalah tiga buah roda. Menghubungkan jumlah kaki dan roda dengan istilah kaki lima adalah pendapat yang mengada-ada dan tidak sesuai dengan sejarah. Pedagang bergerobak yang 'mangkal' secara statis di trotoar adalah fenomena yang cukup baru (sekitar 1980-an), sebelumnya PKL didominasi oleh pedagang pikulan (penjual cendol, pedagang kerak telur) dan gelaran (seperti tukang obat jalanan) (Wikipedia, 2024).

Sebenarnya istilah kaki lima berasal dari masa penjajahan Belanda. Peraturan pemerintahan saat itu mengatur bahwa setiap jalan raya yang dibangun dapat menyediakan sarana untuk perjalanan kaki. Lebar ruas untuk pejalan kaki adalah lima kaki atau sekitar satu setengah meter. Setelah itu, saat Indonesia sudah merdeka, ruas jalan untuk pejalan kaki banyak dimanfaatkan oleh para pedagang untuk berjualan. Jika merunut pada sejarahnya, seharusnya namanya adalah pedagang lima kaki (Wikipedia, 2024)

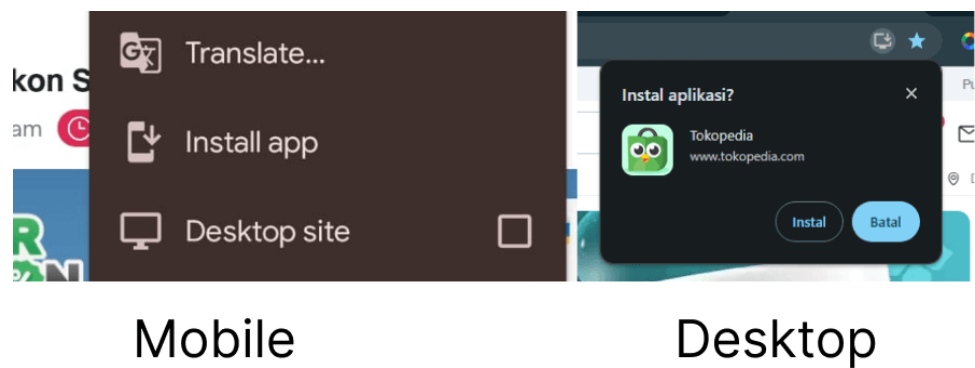
## 2.8 Sistem Kasir

Sistem kasir adalah sebuah alat yang digunakan oleh sebuah jenis usaha untuk mempermudah transaksi penjualan contohnya adalah mesin kasir tetapi fungsinya lebih dibandingkan mesin kasir biasa, sistem kasir biasanya digunakan sebuah kegiatan usaha untuk mencatat pesanan barang, mencatat penjualan, membantu menghitung laba tiap produk yang dijual, membeli barang, manajemen stok, dan lainnya. Aplikasi kasir adalah sistem digital yang digunakan untuk mencatat transaksi penjualan, mengelola stok, dan menghasilkan laporan keuangan. Dalam konteks usaha mikro seperti PKL, aplikasi kasir yang sederhana sangat dibutuhkan untuk menggantikan pencatatan manual yang rawan kesalahan. Aplikasi kasir berbasis desktop dan *web* telah terbukti membantu pelaku usaha kecil dalam mengelola usaha secara lebih efisien dan terstruktur (Afrizal dkk., 2024). Desain aplikasi kasir untuk PKL harus mempertimbangkan keterbatasan teknis pengguna, seperti perangkat keras yang sederhana, keterbatasan internet, serta rendahnya literasi digital. Oleh karena itu, tampilan antarmuka yang sederhana, navigasi intuitif, dan fitur yang fokus pada kebutuhan inti menjadi syarat mutlak

## 2.9 Progressive Web App (PWA)

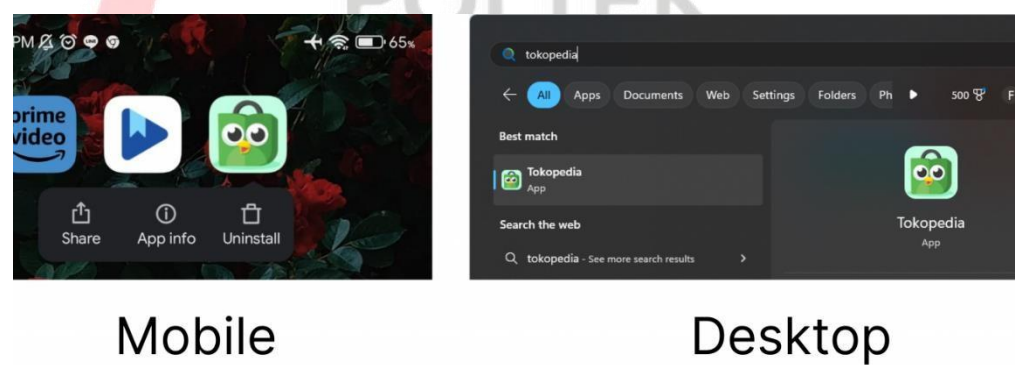
*Progressive Web App* (PWA) adalah sebuah aplikasi *web* yang dirancang atau dibangun dengan melakukan optimasi pada sebuah *website*. *Progressive Web App* (PWA) adalah aplikasi berbasis *web* yang menggabungkan fitur-fitur terbaik dari aplikasi *web* dan aplikasi *native* untuk menciptakan pengalaman pengguna yang cepat, andal, dan menarik. PWA menggunakan teknologi modern seperti *Service Workers*, *Web App Manifest*, dan *HTTPS* untuk memungkinkan fungsionalitas seperti akses *offline*, notifikasi *push*, dan instalasi langsung ke perangkat pengguna tanpa melalui toko aplikasi (Wikipedia, 2025).

Proses penerapan *progressive web apps* pada perangkat pun sangat mudah. Untuk mengetahuinya kunjungi halaman *website* yang ingin dikunjungi menggunakan *browser*. Lalu, perhatikan ikon seperti gambar di bawah ini yang dicontohkan pada salah satu *e-commerce* terbesar di Indonesia, yaitu Tokopedia. (Saepulloh, 2024). Salah satu bentuk penerapan fitur PWA dapat dilihat pada gambar 2.5



Gambar 2.5 Progressive Web App 1

Lalu, jika menekan tombol *install*, aplikasi tersebut akan terpasang dan langsung berfungsi layaknya aplikasi *native* (Saepulloh, 2024). Apabila aplikasi telah terpasang bentuknya akan terlihat seperti pada gambar 2.6



Gambar 2.6 Progressive Web App 2

2.10 Penelitian Terdahulu

Hasil dari penelitian terdahulu digunakan sebagai landasan untuk mengembangkan kerangka pemikiran dalam penelitian ini. Selain itu, informasi tersebut juga dimanfaatkan untuk membandingkan dan menganalisis persamaan serta perbedaan antara beberapa studi terdahulu, serta untuk mengeksplorasi potensi pengembangan lebih lanjut dalam penelitian ini. Ringkasan penelitian terdahulu dapat dilihat pada tabel 2.1

Tabel 2.1 Penelitian Terdahulu

| No | Penulis                    | Judul Penelitian                                     | Jenis Penelitian        | Hasil Penelitian                                                                                  |
|----|----------------------------|------------------------------------------------------|-------------------------|---------------------------------------------------------------------------------------------------|
| 1  | Syafriko Mutiyandani, dkk. | Perancangan Aplikasi Kasir (Studi Kasus: Rumah Makan | Metode <i>Waterfall</i> | Hasil yang dicapai pada penelitian ini adalah aplikasi kasir berbasis <i>Web</i> pada rumah makan |

|   |                          |                                                                                                |                  |                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                               |
|---|--------------------------|------------------------------------------------------------------------------------------------|------------------|-------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------|
|   |                          | Padang Anak Minang)<br>Menggunakan PHP Framework                                               |                  | Anak Minang yang memiliki <i>interface</i> yang sesuai dan <i>user friendly</i> , sehingga dapat memudahkan dan memberikan informasi secara detail pada transaksi penjualan                                                                                                                                                                                                                                                                                                                                                   |
| 2 | Asri Mulyani, dkk        | Rancang Bangun Aplikasi Kasir Penjualan pada Usaha Mikro Kecil Menengah 3Manstore Berbasis Web | Metode Waterfall | Hasil dari penelitian ini berupa Aplikasi Kasir Penjualan untuk Usaha Mikro Kecil Menengah Berbasis Web yang memilih hak akses <i>multi-user</i> serta sudah dilengkapi dengan fitur transaksi, manajemen hutang dan laporan laba rugi.                                                                                                                                                                                                                                                                                       |
| 3 | Aldhita Dwi Lestari, dkk | Rancang Bangun Aplikasi Kasir Berbasis Web pada Rumah Makan RR Chicken                         | Metode Waterfall | Peneliti berhasil membuat perancangan Sistem Aplikasi Kasir Berbasis Web dengan tampilan sistem yang sudah responsif dengan dua <i>role</i> akses, yaitu untuk <i>role admin</i> dan kasir. Adapun fitur yang dapat diakses pada <i>role admin</i> yaitu fitur <i>dashboard</i> , <i>management category</i> , <i>management product</i> , kelola keuangan, pengeluaran, <i>list transaction</i> , dan <i>management account</i> . Sedangkan pada bagian kasir hanya terdapat fitur <i>transaction</i> untuk melayani pesanan |

|   |                                     |                                                                                                                               |                           |                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                              |
|---|-------------------------------------|-------------------------------------------------------------------------------------------------------------------------------|---------------------------|--------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------|
|   |                                     |                                                                                                                               |                           | pembelian lengkap dengan perhitungan otomatis transaksinya, fitur <i>list transaction</i> disertai dengan fitur untuk pembuatan laporan harian dan fitur laporan pengeluaran yang dilakukan di bagian kasir.                                                                                                                                                                                                                                                                                                                                                                                                                                                 |
| 4 | Mahfudz<br>Alfanani<br>Syaviqi, dkk | Rancang Bangun Sistem E-Commerce Berbasis Website dengan <i>Framework Laravel</i> untuk Pemesanan Kasir Tunai di Fintechology | Metode <i>Prototyping</i> | Tujuan penelitian ini merancang dan membangun sebuah sistem <i>e-commerce</i> yang berbasis <i>web</i> menggunakan <i>Framework Laravel</i> . Sistem ini dibuat dengan menerapkan metode <i>Prototyping</i> untuk memenuhi kebutuhan pengguna. Prototipe sistem menghadirkan fitur seperti pemesanan, pengelolaan produk, dan pelaporan transaksi, yang dapat diakses melalui <i>user interface</i> yang mudah dipahami. Penggunaan <i>Laravel</i> memudahkan pengembangan aplikasi yang aman, responsif, dan terhubung dengan baik. Penelitian ini menjelaskan bahwa sistem yang dihasilkan mampu meningkatkan efisiensi operasional, mengurangi kekeliruan |

|   |                          |                                                                                                                                       |                     |                                                                                                                                                                                                                                                                                                                                                                                                              |
|---|--------------------------|---------------------------------------------------------------------------------------------------------------------------------------|---------------------|--------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------|
|   |                          |                                                                                                                                       |                     | pencatatan administrasi, dan memberikan pengalaman pengguna yang cukup baik.                                                                                                                                                                                                                                                                                                                                 |
| 5 | Aqil Farras,<br>dkk      | Pembuatan Aplikasi Inventaris Seal Menggunakan Metode <i>Agile</i> Berbasis <i>Appsheet</i>                                           | Metode <i>Agile</i> | Penelitian ini menghasilkan aplikasi inventaris yang ramah digunakan, meminimalisir potensi kesalahan manusia, serta meningkatkan efisiensi waktu dalam proses pengelolaan inventaris. Dengan demikian, penerapan <i>Agile</i> dalam pengembangan aplikasi <i>AppSheet</i> memberikan solusi yang efektif dalam manajemen inventaris.                                                                        |
| 6 | Steven Hendrawan,<br>dkk | Perancangan Sistem Informasi Permohonan Perizinan Penelitian dengan Metode <i>Agile</i> dan <i>Framework Laravel</i> Berbasis Website | Metode <i>Agile</i> | Peneliti membangun suatu sistem yang dapat mendukung kinerja karyawan di Badan Kesatuan Bangsa dan Politik (Bakesbangpol) Pemerintah Kota Salatiga dalam pengelolaan data permohonan dan pembuatan laporan. Peneliti menggunakan metode perancangan <i>Agile</i> dan <i>framework Laravel</i> dalam pembuatan website. Untuk menggambarkan kebutuhan pengguna, peneliti menggunakan <i>Unified Modelling</i> |

|  |  |  |  |                                                                                                                    |
|--|--|--|--|--------------------------------------------------------------------------------------------------------------------|
|  |  |  |  | <i>Language (UML).</i><br>Pengujian aplikasi<br>dilakukan dengan<br>menggunakan metode<br><i>White Box Testing</i> |
|--|--|--|--|--------------------------------------------------------------------------------------------------------------------|

