BAB II

TINJAUAN PUSTAKA

2.1. Teori Dasar

Dasar pemikiran konseptual yang berfungsi sebagai panduan untuk memahami dan menjelaskan fakta-fakta yang sedang diteliti dikenal sebagai teori dasar dalam penelitian. Teori ini membantu dalam merumuskan hipotesis, menentukan hubungan antara variabel, dan menangani isu-isu penelitian. Selain itu, teori fundamental berfungsi sebagai kerangka kerja yang rasional dan sistematis untuk mengevaluasi temuan penelitian dan mendukung klaim bahwa studi tersebut signifikan dan bahwa teknik yang digunakan relevan.

2.1.1. Software Development

Pengembangan suatu sistem aplikasi memerlukan suatu metode atau alur yang jelas agar sistem yang dibangun sesuai dengan apa yang sudah dirancang. Metode pengembangan *Software Development Life Cicle* (SDLC) merupakan salah satu metode dalam pengembangan *software* yang mengakomodasi kebutuhan pengguna berkaitan dengan sistem yang akan dikembangkan. Kebutuhan pengembangan sistem dapat berupa perubahan atau penciptaan aplikasi baru apakah secara secara modular maupun dengan proses instalasi baru (Silitonga & Purba, 2021).

Metode SDLC merupakan pendekatan sistematis untuk pengembangan perangkat lunak. SDLC menjelaskan tentang langkah- langkah atau tahapan yang harus diikuti oleh tim pengembangan perangkat lunak mulai dari pembuatan ide, pengiriman hingga pemeliharaan produk perangkat lunak. Setiap fase SDLC mempunyai tujuan dan aktivitas spesifik, yang membantu dalam mengelola risiko dan menjamin kualitas produk yang dihasilkan (Yusuf & Fauzi, 2024).

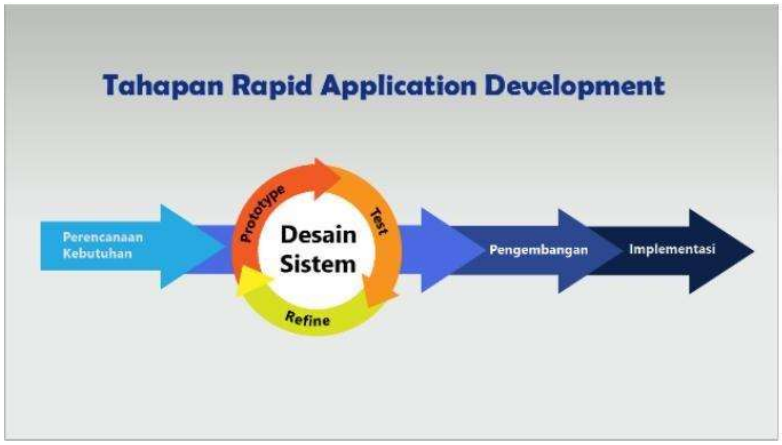
Konsep SDLC mendasari berbagai jenis metodologi pengembangan sistem. Metodologi-metodologi ini membentuk suatu kerangka kerja mulai dari proses perencanaan hingga proses pengendalian pembangunan atau pengembangan sistem informasi. Terdapat beberapa model SDLC yang dapat digunakan, salah satunya yaitu model *waterfall*. Model *waterfall* merupakan pendekatan klasik dalam pengembangan sistem yang menggambarkan metode pengembangan sistem secara linier dan berurutan (Annisa Tri Hidayati et al., 2023).

Selain *waterfall* SDLC juga memiliki metode *Rapid Application Development* (RAD) yang memiliki konsep yang hampir serupa tetapi memiliki efisiensi lebih cepat dari pada *waterfall*. RAD merupakan sebuah perancangan alur siklus hidup yang diperuntukan untuk menyediakan pengembangan yang perancangannya lebih cepat dan menghasilkan kualitas yang jauh lebih baik (P et al., 2022).

2.1.2. Model *Rapid Application Development* (RAD)

Mengembangkan suatu sistem atau aplikasi memerlukan metode *Software Development Life Cicle* (SDLC) salah satu yang dapat digunakan yaitu

menggunakan metode RAD. Metode *Rapid Application Development* (RAD) mampu mengembangkan sistem perangkat lunak yang menekankan pada sisklus pengembangan tahapan yang pendek dengan waktu yang relatif singkat. Metode pengembangan RAD merupakan adaptasi dari model *waterfall* versi kecepatan tinggi dengan menggunakan model *waterfall* untuk pengembangan setiap komponen perangkat lunak (Murdiani & Sobirin, 2022).



Gambar 2. 1 Metode *Rapid Application Development*
Sumber : (Nanda et al., 2025)

Pada gambar 2.1 memperlihatkan tahapan yang akan dilakukan ketika metode RAD diimplementasikan dimana dimulai dengan menganalisa perencanaan kebutuhan sistem, kemudian dilanjutkan dengan desain sistem, pengembangan dan terakhir implementasi sistem untuk menyesuaikan masalah. Penjelasan lebih rinci dari tahapan tersebut sebagai berikut (P et al., 2022) :

1. Perencanaan kebutuhan

   Dimana pada tahap ini merupakan awal dari satu pengembangan aplikasi sistem dengan melakukakn identifikasi permasalahan, pengumpulan data-data yang di peroleh dari perancang guna mengedintifikasi tujuan akhir dari sistem yang di butuhkan atau di rancang.

2. Desain sistem

   Pada tahapan ini perancang mulai mendesain sistemnya (*prototype*), dan kemudian di uji coba. Apabila yang di rencanakan tidak sesuai dengan yang di butuhkan maka dapat di diperbaiki. Pada tahapan ini terdapat spesifikasi *software* yang terdiri dari organisasi di dalam sistem, struktur data dan lain-lain.

3. Proses pengembangan

   Pada tahap ini peracangan desain sistem yang telah di rancang dan di aplikasikan ke versi *beta* sampai dengan versi akhir. Dimana pada proses ini sistem telah dirancang sebagaimana yang dibutuhkan.

4. Implementasi

   Tahapan ini merupakan tahap mengimplementasikan metode program sistem tersebut seperti kebutuhan sistem yang di butuhkan. Dimana pada metode akhir ini merupakan penerapan akhir dan dapat dijalankan.

2.1.3. Android

Aplikasi yang akan dibangun menggunakan basis android dimana android sendiri sudah hampir digunakan semua kalangan masyarakat, android merupakan sebuah sistem operasi untuk perangkat mobile berbasis Linux yang mencakup sistem operasi, *middleware* dan aplikasi. Android menyediakan *platform* terbuka bagi para pengembang untuk menciptakan aplikasi mereka. Pesatnya pertumbuhan android diantaranya karena android itu sendiri adalah *platform* yang sangat lengkap baik itu dari sisi sistem operasinya, aplikasi *DNA Tool* pengembangan, pasar

aplikasi android serta dukungan yang sangat tinggi dari komunitas *open source* di dunia (Rahmat Gunawan et al., 2021).

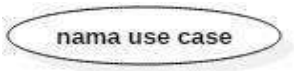
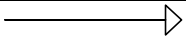
2.1.4. *Unified Modelling Language* (UML)

Mengembangkan sebuah aplikasi kita membutuhkan alat untuk melakukan desain sebuah sistem dimana desain tersebut akan memberikan gambaran alur dalam sebuah sistem yang kita rancang, salah satu yang dapat menggambarkan sebuah sistem dalam aplikasi yaitu *Unified Modelling Language* (UML) yang merupakan merupakan suatu teknik untuk memodelkan sistem. UML merupakan seperangkat aturan dan notasi untuk spesifikasi sistem *software*. Notasi ini menyediakan satu set elemen grafis untuk pemodelan sistem. Perancangan dan pembangunan aplikasi atau *software* berbasis objek atau *Object Oriented Analysis and Design* (OOAD) menganggap segala sesuatunya adalah objek serta sistem dipandang sebagai interaksi dari banyak objek yang dimodelkan menggunakan UML (Annisa Tri Hidayati et al., 2023).*Unified Modelling Language* (UML) memiliki berbagai macam bentuk dalam pemodelan suatu sistem yang dapat digunakan, diantaranya yaitu :

1. *Usecase Diagram*

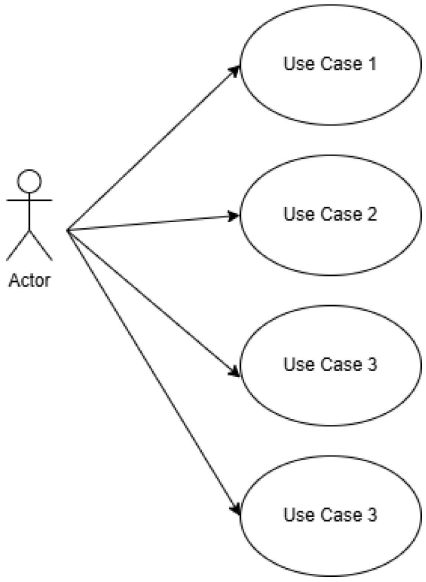
Dalam analisis dan desain perangkat lunak, diagram *use case* merupakan alat penting untuk menggambarkan bagaimana aktor dan sistem berinteraksi. Diagram ini secara efektif mempromosikan komunikasi dan kolaborasi antara pengembang dan pemangku kepentingan selama proses desain sistem dengan menggunakan simbol standar.

Tabel 2. 1 Simbol *Usecase Diagram*

Notasi	Keterangan
 *Use Case*	Hubungan yang ada antara seorang aktor dan sistem.
 *Actor*	Individu yang berinteraksi dengan sistem disebut sebagai pengguna sistem.
Association	Ini adalah hasil dari interaksi antara komponen-komponen.
Extend	Mengungkapkan niat atau tujuan yang dimiliki oleh sistem yang saat ini sedang beroperasi.
 *Generalization*	Ini adalah komponen yang memiliki makna yang berbeda dan unik.
<<include>>	Kriteria perilaku penting yang harus dipenuhi.

Sumber : (Ahmadar et al., 2021)

Pada tabel 2.1 merupakan simbol-simbol yang di atur ketika digunakan untuk merancang sebuah *use case*. Simbol ini akan mewakilkan sebuah atribut yang dapat dijelaskan. Dengan menggunakan simbol diatas, salah satu contoh bentuk perancangan *use case* tersebut menjadi sebuah *use case* dapat terlihat pada gambar di bawah ini :



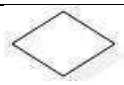
Gambar 2. 2 I *Use Case* diagram
Sumber : (Ahmadar et al., 2021)

Gambar 2.2 menggambarkan diagram *use case* yang mewakili aktor utama yang terhubung ke beberapa *use case*. Setiap *use case* ini menggambarkan sebuah interaksi atau fungsi yang dilakukan dalam sistem, yang mencakup berbagai proses yang dapat diakses oleh aktor tersebut.

2. *Activity Diagram*

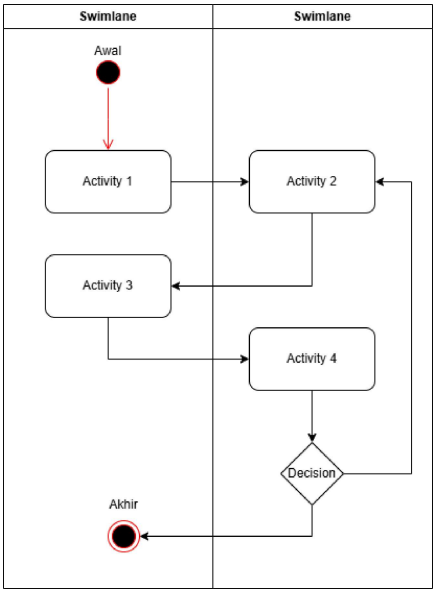
Activity diagram dalam UML adalah representasi visual yang menunjukkan urutan tugas, tindakan, dan alur kerja di dalam sebuah sistem atau perusahaan. Keputusan direpresentasikan dengan simbol berlian, alur kerja dengan panah, dan aktivitas dengan elips beserta deskripsi. Diagram ini sangat baik untuk memeriksa, merencanakan, dan mensimulasikan perangkat lunak serta proses bisnis, termasuk keputusan, alur kerja, dan tugas yang berjalan bersamaan.

Tabel 2. 2 Simbol *Activity Diagram*

Notasi	Keterangan
 *Swinlane*	Digunakan untuk memecah *activity* diagram untuk memecah tanggung jawab setiab colom.
 *Start*	Permulaan dari suatu aktivitas.
 *Activity*	Ketergantungan dalam suatu aktivitas.
 *Decision*	Pemisahan jalur berdasarkan kriteria tertentu.

Sumber : (Ahmadar et al., 2021)

Pada tabel 2.2 merupakan simbol-simbol yang dapat digunakan untuk merancang sebuah *activity diagram* dimana simbol-simbol tersebut mewakili atribut yang dapat dijelaskan. Simbol-simbol tersebut dapat digunakan untuk membangun *activity diagram* sebagaimana contoh dibawah ini :



Gambar 2. 3 *Activity* Diagram
Sumber : (Ahmadar et al., 2021)

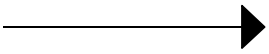
Gambar 2.3 merupakan contoh diagram alur aktivitas dalam format *swimlane,* yang membagi aktivitas menjadi dua *swimlane* yang berbeda, masing-masing untuk *entitas* yang terlibat dalam proses tersebut. Diagram ini menggambarkan aliran aktivitas, mulai dari *activity 1*, yang diikuti oleh *activity* 3 pada *swimlane* pertama. Di sisi *swimlane* kedua, ada *activity* 2 yang dilanjutkan dengan *activity* 4, dengan sebuah *decision* yang menandai titik percabangan dalam proses. Diagram ini digunakan untuk menunjukkan interaksi antara berbagai aktor atau *entitas* dalam suatu sistem yang melibatkan beberapa tahapan keputusan dan aktivitas.

3. *Sequence Diagram*

Sequence diagram digunakan dalam rekayasa perangkat lunak untuk menunjukkan bagaimana proses berinteraksi satu sama lain dari waktu ke waktu. Panah horizontal digunakan untuk menunjukkan pesan antara objek, dan garis vertikal digunakan untuk mewakili objek. Diagram ini, yang biasanya diambil dari

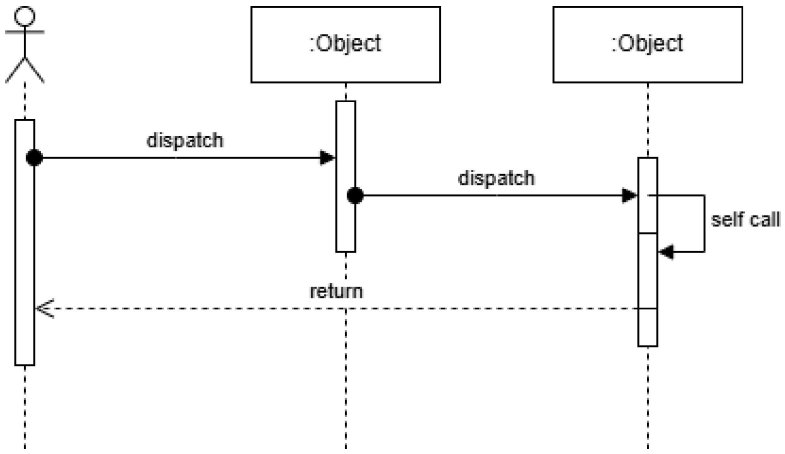
satu diagram kasus penggunaan, menunjukkan tindakan yang dilakukan sebagai respons terhadap suatu peristiwa untuk menghasilkan suatu kesimpulan.

Tabel 2. 3 Simbol *Sequence Diagram*

Notasi	Keterangan
 *Actor*	Orang-orang yang terlibat dalam memfasilitasi komunikasi di dalam sistem disebut pengguna.
Life line	Elemen ini berfungsi sebagai penghubung antara aktor dan *use case* dalam suatu sistem.
 *Object*	Komunikasi adalah proses yang terjadi antara dua entitas.
 *Uptime*	Gambaran ini menyoroti sifat berkelanjutan dari operasi, karena operasi tersebut berfungsi untuk menjaga hubungan antara entitas yang terlibat.
 *order type create*	Sebuah entitas menerima pernyataan, yang kemudian ditransfer ke entitas lain.

Sumber : (Ahmadar et al., 2021)

Tabel 2.3 menerangkan berbagai simbol yang dapat digunakan dalam merancang sebuah *sequence* diagram, dimana setiap simbol mewakili aktifitas yang terjadi pada sistem. Salah satu contoh *sequence* diagram yang dapat tergambar dari simbol-simbol tersebut sebagai berikut :



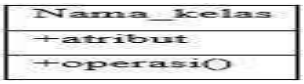
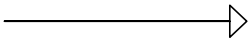
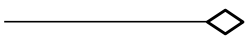
Gambar 2. 4 *Squence Diagram*
Sumber Data : (Silitonga & Purba, 2021)

Gambar 2.4 menggambarkan *sequence diagram* yang menggambarkan interaksi antara dua objek dalam sistem. Diagram ini menunjukkan berbagai pesan yang dikirimkan antar objek. *Dispatch* menunjukkan pengiriman pesan ke objek lain, *return* mengindikasikan kembalinya suatu nilai atau kontrol dari objek, dan *self call* menunjukkan panggilan metode dari objek itu sendiri. Diagram ini digunakan untuk memodelkan urutan peristiwa atau operasi dalam suatu sistem perangkat lunak, yang memungkinkan analisis interaksi waktu nyata antar objek atau komponen dalam sistem.

4. *Class Diagram*

Dalam sistem berorientasi objek, *class* diagram menampilkan karakteristik, metode, dan hubungan antar kelas serta hubungan dan struktur mereka. Hubungan statis antara kelas-kelas ini dijelaskan oleh diagram struktural UML ini, yang menekankan struktur kelas daripada interaksi objek.

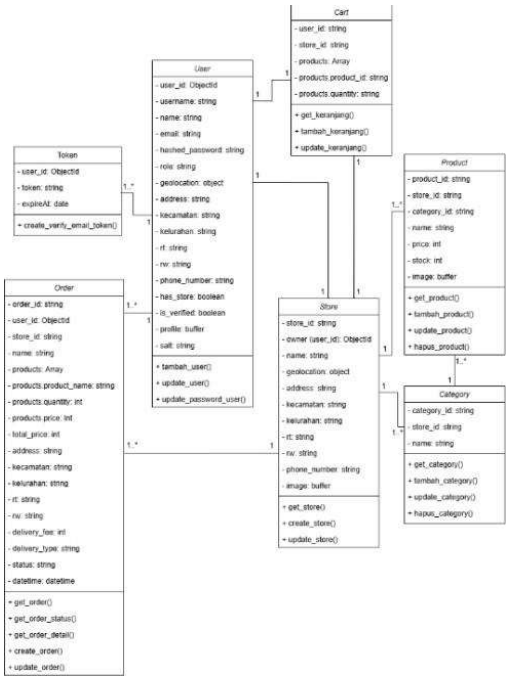
Tabel 2. 4 Simbol *Class diagram*

Notasi	Keterangan
 *Class*	Susunan kelas yang ada di dalam sistem
 *Interface*	Fokus utamanya adalah pada pemrograman berbasis objek.
Association	Atribut yang menunjukkan kesamaan atau kesetaraan yang lengkap.
Directed Association	Kehadiran dua *multiplisitas* memberikan arti padanya.
 *Generalization*	Hubungan dapat dilihat dalam konteks yang khusus maupun umum
 *Dependency*	Ketergantungan antara satu kelas dan kelas lainnya.
 *Agregation*	(Seluruh bagian)

Sumber : (Ahmadar et al., 2021)

Tabel 2.4 merupakan simbol-simbol yang dapat digunakan untuk membangun *class diagram*, setiap simbol akan membantu membangun *class*

diagram dimana menggambarkan alur data pada suatu sistem. Salah satu gambaran *class* diagram sebagai berikut:



Gambar 2. 5 Class Diagram
Sumber : (Kevin & Saragih, 2025)

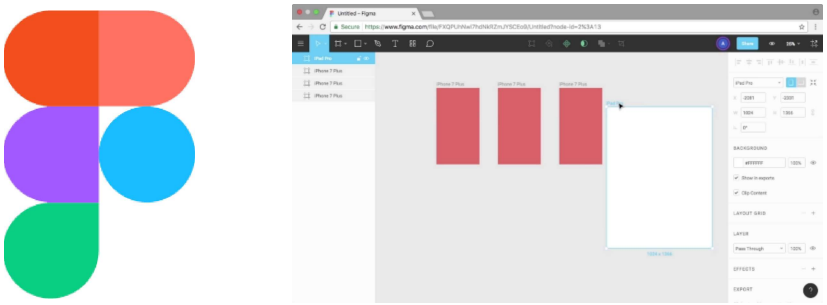
Class diagram yang digunakan untuk memodelkan struktur dan hubungan antar entitas dalam sistem perangkat lunak. Diagram ini menunjukkan bagaimana berbagai objek saling terhubung dan memiliki atribut serta metode yang mendefinisikan fungsionalitas mereka. Setiap entitas dalam sistem memiliki peran tertentu dan berinteraksi dengan entitas lainnya melalui hubungan yang jelas, yang memungkinkan pemrograman berbasis objek untuk mengatur aliran data dan operasi dengan efisien.

2.1.5. Software Pendukung

Software pendukung digunakan oleh periset untuk menjadi alat untuk memfasilitasi upaya penelitian mereka. Berikut adalah perangkat lunak yang digunakan dalam konteks ini.

1. *Figma*

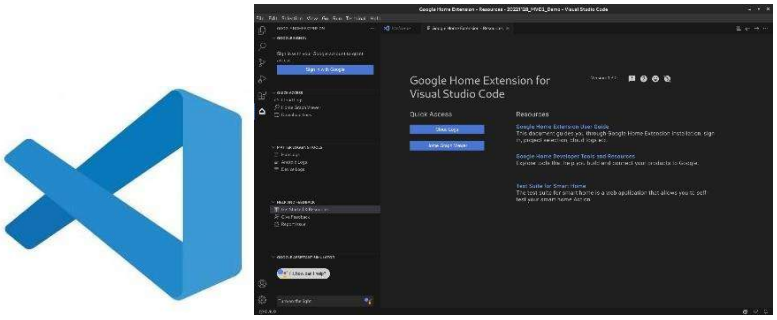
Figma merupakan salah satu design tool berbasis *cloud* dan alat *prototyping* untuk produk digital yangbiasanya digunakan untuk membuat desain aplikasi mobile, website, desktop dan sebagainya. *Figma* dapat digunakan pada sistem operasi mac, linux ataupun windows dengan menghubungkan perangkat ke internet (Febyla & Zubaidi, 2022).



Gambar 2. 6 Logo Figma dan Halaman Home
Sumber : (https://www.oreilly.com/content/how-do-i-define-my-workspace-in-figma/)

2. Visual Studio Code

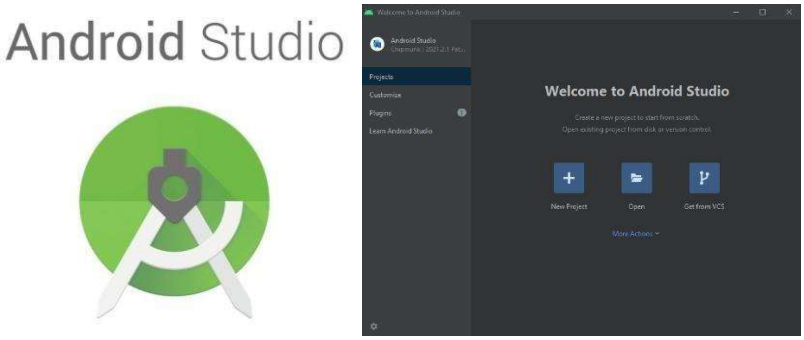
Visual Studio, juga dikenal sebagai VS Code, adalah editor perangkat lunak gratis yang dikembangkan oleh *Microsoft*. Ini dapat digunakan pada perangkat berbasis *Windows*, *Linux*, dan *macOS*, dan memungkinkan pengembang membuat perangkat lunak komputer, aplikasi seluler, sistem web, layanan web, dan berbagai sistem lainnya.



Gambar 2. 7 Logo VS Code dan Halaman Home Sumber : (https://vetores.org/visual-studio-code-svg-logo/)

3. *Android Studio*

Android studio yaitu *Integrated Development Environment* telah disahkan dalam meningkatkan aplikasi Android yang bersifat gratis. Android Studio ini diluncurkan dan diinformasikan oleh Google pada event Google I/O *Conference* untuk tahun 2013. (Rahmat Gunawan et al., 2021).

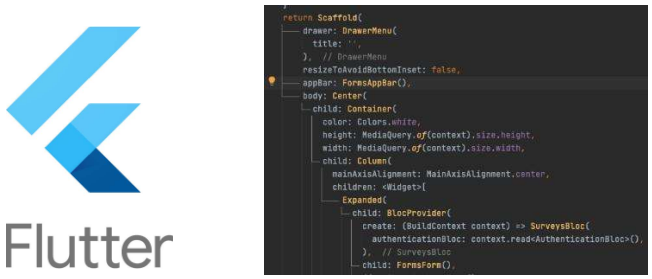


Gambar 2. 8 Logo dan Halaman Home Android Studio Sumber : (https://ccm.net/downloads/programming/8651-android-studio/)

4. *Flutter*

Flutter adalah sebuah *framework open-source* yang dikembangkan oleh Google untuk membangun antar muka (*user interface*) aplikasi android dan iOS. Jika kita berbicara bagaimana cara membuat aplikasi android maupun iOS, biasanya

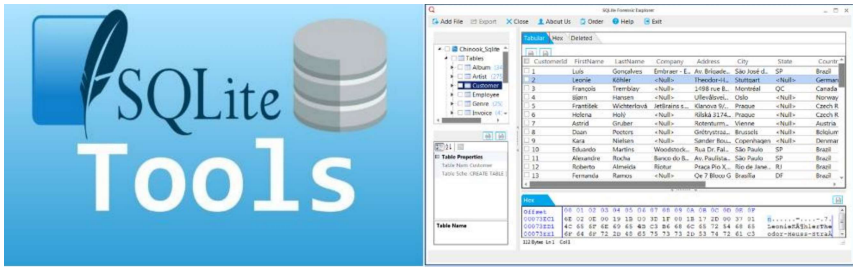
akan dihadapi dengan banyak pilihan kenapa aplikasi tersebut dapat dibangun.(Panji Rachmat Setiawan et al., 2022)



Gambar 2. 9 Logo dan Stuktur Flutter
Sumber : (https://codersera.com/blog/flutter-tutorial-how-to-create-your-first-flutter-app/)

5. *Sqlite*

Salah satu program *embedded* yang paling banyak digunakan adalah *SQLite*, yang menggabungkan antarmuka *SQL* dengan jejak memori yang kecil dan kecepatan yang sangat cepat. Setiap versi Android dapat menggunakan *SQLite* untuk membangun basis data karena itu adalah bagian dari *runtime* android (Nusantoro & Simanjutak, 2025).



Gambar 2. 10 Logo dan Home Sqlite
Sumber : https://www.testingdocs.com/wp-content/uploads/SQLite-Tools-768x480.png

2.2. Penelitian Terdahulu

Sebagai dasar untuk memahami topik yang dibahas, penting untuk mempelajari hasil penelitian terdahulu. Bagian ini akan mengulas penelitian terkait

yang relevan, guna memberikan gambaran yang lebih jelas dan mendalam tentang konteks penelitian ini. Berikut adalah beberapa penelitian sebegai referensi dalam penelitian ini :

1. Penelitian yang dilakukan oleh (Markovic et al., 2021) dengan judul penelitian "***Business-to-business open innovation: COVID-19 lessons for small and medium-sized enterprises from emerging markets***" mengangkat permasalahan dimana pelaku UMKM masih menggunakan metode penjualan dengan pencatatan fisik, dengan menggunakan teknologi aplikasi berbasis android untuk membantu dalam pelaku usaha tersebut.
2. Penelitian yang berjudul "**Inti Nusa Mandiri Model *Rapid Application Development* Untuk Rancang**" yang di lakukan oleh (Makmur, 2023) mengangkat permasalahan tentang pencatatan secara fisik dimana masih banyak kekurangan termasuk kerusakan pada benda fisik tersebut. Peneliti tersebut membuat sebuah rancangan aplikasi dengan menggunakan metode *Rapid application Development* (RAD) dalam perancangannya untuk membuat aplikasi dalam membantu mengurangi resiko akibat pencatatan secara fisik.
3. Penelitian yang dilakuka oleh (Siki et al., 2024) dengan judul "**Implementasi Model *Rapid Application Development* (RAD) dalam Pengembangan Website Penjualan Produk UMKM Bikomi Utara**" dimana peneliti ingin membantu pelaku UMKM agar menggunakan teknologi untuk meningkatkan efisiensi kinerja penjualan, maka peneliti

membuat aplikasi berbasis *website* dengan menggunakan metode *Rapid Application Development* (RAD)**.**

4. Penelitian yang berjudul "**Rancang Bangun Aplikasi *E-Vegetables* Menggunakan *Flutter* Di Kota Batam**" yang dilakukan oleh (Kevin & Saragih, 2025) mengankat permasalahan tentang kurangnya pemanfaatan teknologi dalam kegiatan jual beli, maka penelitian tersebut membuat aplikasi berbasis android dengan *flutter* menggunakan metode *Rapid Application Development* (RAD) yang menghasilkan rancangan aplikasi penjualan.
5. Penelitian yang dilakukan oleh (Yusuf & Fauzi, 2024) yang berjudul "**Digitalisasi Sistem Koperasi Karyawan Dengan Aplikasi Berbasis Android Di Pt Citra Tubindo Tbk Baharudin**" yaitu melakukan digitalisasi dalam pencatatan transaksi berbasis android untuk meningkatkan penggunaan teknologi dalam bidang usaha.
6. Penelitian yang berjudul "**Perancangaan Aplikasi Mobile *E-Commerce* Berbasis Android pada Natasya Butik**" yang di lakukan oleh (Sari & Andriasari, 2023) dimana mengangkat permasalahan dimana kegiatan transaksi masih secara manual sehingga akan memperlambat proses transaksi dan besar kemungkinan terjadinya kesalahan. Dengan permasalahan tersebut peneliti membuat media pemesanan berbasis androd dengan metode *waterfall*, Dengan adanya penerapan aplikasi mobile *commerce* dapat membantu penjual dalam melakukan pengelolahan data pemesanan barang, data pelanggan, dan konfirmasi pengiriman barang.

7. Penelitian yang dilakukan oleh (Yanuarsyah et al., 2024) dengan judul "**Dijitalisasi *E-Commerce* Warung Sayur Ibu Ecih Berbasis Android**" dimana peneliti ingin menarik minat pembeli lagi untuk berbelanja dengan membuat rancangan aplikasi berbasis androd dengan metode *prototype* dimana saat itu terjadi kendala yaitu pemberlakuan pembatasan kegiatan masyarakat.
8. Penelitian "**Perancangan *E-Commerce* Umkm Di Desa Kilangan Kecamatan Muara Bulian Kabupaten Batanghari**" yang di lakukan oleh (Yani et al., 2025) mengangkat permasalahan dimana kurangnya *efisiensi* dalam menggunakan media sosial sebagai media promosi bisnis UMKM di daerah tersebut. Dari permasalahan tersebut peneliti ingin membangun sistem informasi *E-Commerce* penjualan UMKM berbasis web dengan metode *Rapid Application Development* (RAD) dimana peneliti menghasilkan sistem informasi *E-Commerce* berbasis web yang dapat digunakan pihak Desa Kilangan untuk mempermudah *customer* dalam melihat informasi produk yang dibutuhkan, serta pemrosesan data tersimpan secara terpusat dan terintegrasi ke dalam database dan dapat memasarkan dan mempromosikan produk secara online.
9. Penelitian yang dilakukan (Hidayat & Ali, 2024) dengan judul penelitian "**Pengembangan Aplikasi Penjualan Tas Spundbond Dengan *Payment Gateway* Berbasis Web Menggunakan Metode RAD**" mengangkat permasalahan kurangnya *efisiensi* transaksi bisnis, dengan permasalahan tersebut peneliti melakukan pengembangan aplikasi *e-commers* dengan

menggunakan *framework* Laravel dan pendekatan *Rapid Application Developmen*t (RAD), dengan metode tersebut penelitian ini menghasilkan sistem *e-commerce* yang memungkinkan Tulip Craft Bogor untuk menjual produk tas spunbond secara online dengan cara yang mudah dan efektif, mengatasi tantangan pemasaran, dan meningkatkan efisiensi penjualan melalui platform web

10. Penelitian yang dilakukan (Nusantoro & Simanjutak, 2025) dengan judul "**Implementasi Aplikasi *E-Commerce* Berbasis Android Menggunakan Metode *Waterfall***" dimana mengangkat permasalahan kurangnya efisiensi dalam pemasaran dimana hanya menggunakan poster, dari masalah tersebut peneliti mengembangkan aplikasi berbasis Android dengan menggunakan metode *waterfall*, peneliti berhasil membangun aplikasi dimana mempermudah proses pemasaran, pemesanan, dan penjualan produk, serta meningkatkan pendapatan penjualan toko.

2.3. Kerangka Pemikiran

Masalah pengelolaan penjualan dan data transaksi pada Shakel Kebab sering kali menjadi tantangan dalam menjalankan operasional bisnis. Untuk mengatasi hal ini, diperlukan solusi teknologi yang dapat mempercepat dan mempermudah proses pengelolaan tersebut. Salah satu cara yang dapat dilakukan adalah dengan mengembangkan aplikasi berbasis Android yang dapat mengelola

transaksi secara lebih efisien. Berikut adalah kerangka pemikiran yang digunakan untuk merancang aplikasi tersebut.

Gambar 2. 11 Kerangka Pemikiran
Sumber : Penelitian 2025

Berdasarkan kerangka pemikiran di atas, permasalahan yang dihadapi dalam pengelolaan penjualan dan data transaksi pada Shakel Kebab dapat diselesaikan melalui penerapan metode *Rapid Application Development* (RAD). Dengan menggunakan *Android Studio* dan *Visual Studio*, aplikasi yang dapat mengelola penjualan kebab berbasis android dapat dirancang. Pengembangan aplikasi ini bertujuan untuk mempermudah proses transaksi, mempercepat pengolahan data, dan memberikan kemudahan bagi pemilik serta pelanggan dalam melakukan pemesanan, pembayaran, serta pencatatan transaksi secara efisien. Implementasi RAD memungkinkan pengembangan aplikasi yang lebih cepat dan efektif, sehingga dapat memenuhi kebutuhan bisnis dengan lebih baik.