

BAB II

TINJAUAN PUSTAKA

2.1. Penelitian Sebelumnya

Proses penelitian diawali dengan melakukan kajian pustaka terhadap penelitian terdahulu dengan tujuan untuk memahami penerapan konsep yang akan digunakan dalam penelitian. Referensi yang digunakan yaitu jurnal asli, terbaru, dan relevan dengan penelitian selanjutnya. Kajian pustaka untuk mencari perbandingan dan inspirasi ide gagasan baru. Perbandingan dilakukan untuk mengetahui perbedaan antara penelitian sebelumnya dengan penelitian selanjutnya. Adapun perbedaan penelitian sebelumnya dengan penelitian yang akan dilakukan terdapat pada tabel 2.1.

Tabel 2.1 Penelitian Sebelumnya.

No	Judul	Penulis	Kesamaan	Perbedaan	Kesimpulan
1.	Pengembangan Aplikasi <i>Mobile</i> Pencatatan Transaksi Penjualan Olahan Mangga Terintegrasi Berbasis <i>Android</i> [7].	Adi Suheryadi, Kurnia Adi Cahyanto, M Lukman Sifa, Diyanto, Ade Diana P.	Sistem yang dibangun memiliki kesamaan dalam penerapannya yaitu berbasis <i>Android</i> .	Penggunaan <i>database</i> yang digunakan pada penelitian sebelumnya menggunakan <i>MYSQL database</i> , sedangkan yang peneliti lakukan menggunakan non SQL yaitu <i>Firestore</i> .	Penggunaan aplikasi pencatatan transaksi telah di terapkan oleh UMKM Kenanga Mandiri. Sistem menerapkan <i>client-server</i> , dengan adanya penerapan sistem tersebut memudahkan proses transaksi yang terjadi.
2.	Rancang Bangun Sistem Informasi Penjualan Dan Laporan Keuangan Berbasis Web (Studi Kasus UD. Taru Lestari Desa Sibetan Kecamatan Bebandem) [3].	I Gusti Ayu Tania Pratini, I Gede Putu Krisna Juliharta, I Nyoman Yudi Anggara Wijaya	Permasalahan yang dihadapi hampir cenderung sama dimana terjadi kesalahan dalam pencatatan penjualan dan pembuatan laporan.	Sistem yang dibangun pada penelitian sebelumnya berbasis <i>web</i> , sedangkan penulis berbasis <i>android</i> .	Perancangan dan pengembangan sistem pelaporan penjualan dan laporan keuangan memiliki beberapa fitur diantaranya: transaksi, penambahan stok barang, data pengeluaran, dan pelaporan penjualan dan keuangan. Dari hasil uji, bisa di simpulkan bahwa perlu sebuah sistem guna membuat laporan penjualan dan keuangan.

No	Judul	Penulis	Kesamaan	Perbedaan	Kesimpulan
					Penerapan sistem tersebut membantu mempermudah proses pencatatan dan pelaporan transaksi.
3	Aplikasi <i>Point Of Sale</i> (POS) Dengan Metode <i>Model View Controller</i> Studi Kasus Geprek Rame [6].	Aulia Hamdi, Toni Anwar, dan Irfan Santiko	Mengembangkan Aplikasi <i>Point of Sale</i> untuk menangani masalah dalam hal transaksi dan pelaporan penjualan.	Dalam penelitian sebelumnya menggunakan metode <i>Model View Controler</i> (MVC), sedangkan penulis menggunakan <i>Prototyping untuk</i>	Penggunaan metode MVC sangat baik untuk mengembangkan sistem karena memisahkan antara tampilan/ <i>user interface</i> , data, dan proses sehingga mengefektifkan dalam proses pengembangan dan perawatan sistem.

No	Judul	Penulis	Kesamaan	Perbedaan	Kesimpulan
4	Aplikasi Pemandu Desa Wisata Berbasis <i>Android</i> Menggunakan <i>Firebase Realtime Database</i> [9].	Aldy Firmansyah, Hari Murti	Penelitian ini memiliki kesamaan pada penggunaan <i>database</i> yaitu menggunakan <i>database</i> dari <i>Firebase</i> .	Sistem yang dibangun memiliki perbedaan, dimana penelitian sebelumnya membuat sistem untuk pemandu desa wisata, sedangkan penelitian ini membangun sistem <i>Point of Sale</i> . Kemudian metode yang digunakan dalam penelitian sebelumnya menggunakan <i>Rapid Application</i>	Perubahan data yang dilakukan akan langsung berubah tanpa perlu melakukan tindakan perbaruan, semua itu karena menggunakan <i>real-time database</i> dari <i>Firebase</i>
5	Pengembangan Teknologi <i>Mobile</i> Untuk Sistem Kasir Rumah Makan Di Kota Sampit Menggunakan <i>Firebase Realtime Database</i> [8].	Kurniawati, dan Lukman Bachtiar	Terdapat kesamaan masalah dengan penelitian sebelumnya, yaitu sering terjadi kesalahan dalam dalam perhitungan total transaksi, kesulitan dalam pembuatan laporan. Kemudian sama-sama menggunakan <i>database</i> <i>Firebase</i> .	Studi kasus yang diteliti memiliki perbedaan, dimana penelitian sebelumnya berada di “Warung Ayam Panggang Martapura” di Sampit, penelitian yang akan dilakukan berada di “KINI <i>Cheese Tea</i> Berkoh” Purwokerto.	Dengan menerapkannya sistem kasir dapat membantu dan mempermudah proses transaksi, dan membantu pengguna dalam proses pencatatan laporan.

No	Judul	Penulis	Kesamaan	Perbedaan	Kesimpulan
6	Pengembangan Aplikasi <i>Point Of Sale</i> Berbasis <i>Android</i> Menggunakan Metode <i>Rapid Application Development</i> [10].	Eko Junirianto, Nia Kurniadin	Penelitian sebelumnya memiliki kesamaan yaitu penerapan sistem <i>Point of Sale</i> untuk menyelesaikan permasalahan yang ada.	Penelitian sebelumnya memiliki perbedaan dengan penelitian yang dilakukan penulis, dimana metode yang digunakan pada penelitian sebelumnya menggunakan metode <i>Rapid Application Development</i> (RAD) sedangkan penulis menggunakan metode <i>Prototyping</i> .	Penggunaan metode RAD dapat dibilang membutuhkan waktu yang sebentar dalam membangun sistem.
7	Rancang Bangun Aplikasi <i>Point Of Sales</i> Pada Apotek Kamila Berbasis <i>Web</i> [11].	Wulan damayanti, Jemakmun, Suyanto	Membangun sistem <i>Point of sale</i> untuk menangani permasalahan yang ada.	Perbedaan dengan penelitian sebelumnya adalah dalam penelitian ini membuat sistem dengan berbasis android, sedangkan penelitian sebelumnya berbasis website. Kemudian penelitian sebelumnya menggunakan <i>barcode</i> , sedangkan penelitian ini tidak.	Penelitian sebelumnya menjabarkan bahwa penggunaan <i>barcode</i> dapat mempermudah dan mempercepat proses transaksi.

No	Judul	Penulis	Kesamaan	Perbedaan	Kesimpulan
8.	Pemanfaatan Realtime Database di Platform Firebase Pada Aplikasi E-Tourism Kabupaten Nabire [12].	Edwin Adrin Wihelmus Sanadi, Andani Achmad, Dewiani	Persamaan pada penelitian sebelumnya adalah sama-sama menggunakan <i>database</i> dari Firebase untuk menyimpan data.	Perbedaan terletak pada studi kasus dan sistem yang dibangun, dimana penelitian sebelumnya membuat sistem <i>E-Tourism</i> di Kabupaten Nabire, sedangkan dalam penelitian ini membuat POS pada “KINI <i>Cheese Tea</i> Berkoh” di Purwokerto.	Dengan menggunakan <i>realtime database</i> dari Firebase, memungkinkan dalam perubahan data secara langsung ketika melakukan perubahan tanpa harus melakukan <i>refresh</i> pada halaman.

2.2. Dasar Teori

A. *Point of Sale*

Point of Sale merupakan sistem yang memungkinkan untuk melakukan transaksi, yang mana di dalamnya memuat mesin kasir. Sistem POS tidak hanya sekedar melakukan transaksi penjualan, didalamnya juga terintegasi perhitungan akuntansi, manajemen barang dan stok, modul pegawai, modul utang piutang dan berbagai macam fungsi lainnya [6].

Sebelum sistem *Point of Sale* dikembangkan, kebanyakan orang menggunakan *cash register*, dan bahkan secara manual untuk menangani transaksi. Fungsi-fungsi dari *cash register* sendiri sudah kurang memadai, hal itu menjadikan dikembangkannya POS pada zaman sekarang ini [13].

B. *System Development Life Cycle (SDLC)*

System Development Life Cycle atau yang biasa disebut siklus hidup sistem merupakan proses pembuatan ataupun perubahan suatu sistem serta model yang digunakan untuk mengembangkan sistem tersebut. SDLC terdiri dari beberapa tahapan: *planning* (rancangan), *analysis* (analisis), *design* (desain), *implementation* (implementasi), *testing* (ujicoba), dan *maintenance* (pengelolaan). Model-model pengembangan perangkat lunak lainnya juga berdasarkan pada konsep SDLC. Model tersebut diantaranya adalah *Prototyping*, *Waterfall*, *Rapid Application Development (RAD)*, dan lain sebagainya [14].

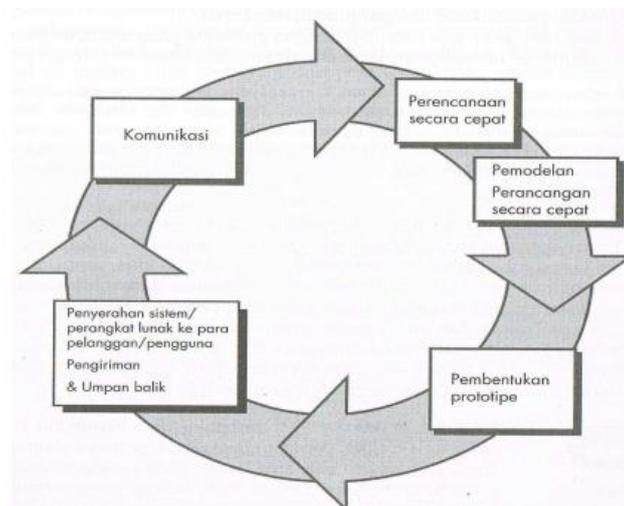
C. *Prototyping*

Metode *prototyping* merupakan model proses yang diterapkan dalam berkomunikasi dengan pelanggan untuk membuat sebuah aplikasi atau sistem. Metode *prototyping* digunakan karena metode ini mengacu pada kepuasan pelanggan secara keseluruhan [15].

Metode *prototyping* digunakan untuk melakukan perancangan sistem yang akan dikembangkan. Metode ini cocok digunakan untuk mengembangkan perangkat yang akan dikembangkan kembali, dan yang belum memiliki kejelasan akan kebutuhan penggunaannya. Pada *prototype* segala perubahan dapat terjadi guna untuk memenuhi kebutuhan pengguna dan memungkinkan

pengembang agar dapat memahami kebutuhan dari pengguna dengan lebih baik [16]. Berikut merupakan tahapan dalam metode *prototyping* :

- a. Komunikasi dan pengumpulan data awal, fase ini merupakan tahap awal dalam metode *prototyping*. Fase ini bertujuan untuk mengidentifikasi kebutuhan dalam pembuatan aplikasi. Pelanggan berperan sangat penting pada fase ini agar hasil perancangan aplikasi sesuai dengan keinginan pelanggan.
- b. *Quick design* (desain cepat), pada tahap ini permintaan pengguna yang didapatkan dari fase komunikasi akan dikembangkan menjadi rancangan desain *interface* sesuai kebutuhan utama dan pendukung.
- c. Permodelan perancangan secara cepat, pada tahap ini dilakukan permodelan aplikasi sesuai kebutuhan pelanggan berupa UML dan kebutuhan lain sesuai kebutuhan pelanggan.
- d. Pembentukan *prototype*, pada tahap ini dilakukan pembangunan aplikasi sesuai data-data pada tahap sebelumnya.
- e. Penyerahan *prototype* dan umpan balik, *prototype* yang telah dibangun pada tahap ini diserahkan kepada pengguna. Hal ini ditujukan agar pengguna dapat memberikan umpan balik tentang aplikasi yang dibuat. Umpan balik tersebut akan digunakan sebagai dasar acuan untuk melakukan perbaikan sesuai dari spesifikasi dari pengguna.



Gambar 2.1 Paradigma pembuatan Prototyping [16].

D. Android

Android merupakan sistem operasi perangkat mobile berbasis *linux* yang didalamnya menyangkut sistem operasi, *middleware*, dan aplikasi. *Android* merupakan *platform open source* dimana para pengembang bebas menciptakan aplikasi mereka sendiri [17]. Arsitektur android dapat dijelaskan sebagai berikut :

1. *Applications* dan *Widgets*, yaitu layar yang berhubungan dengan aplikasi saja, seperti pada saat mengunduh hingga menjalankan aplikasi.
2. *Applications framework* merupakan layer pembuat aplikasi melakukan pengembangan atau pembuatan aplikasi yang akan dijalankan pada sistem operasi android.
3. *Libraries* merupakan layer dimana fitur-fitur berada. Biasanya para pengembang mengakses *libraries* untuk menjalankan aplikasi. *Layer* ini meliputi berbagai *library C/C++* inti seperti *Libc* [17].

E. Unified Modeling Language (UML)

Unified Modeling Language (UML) merupakan metode permodelan secara visual yang digunakan untuk merancang dan atau membuat *software* berorientasi objek. UML juga memberikan standar penulisan sebuah sistem *blue print*, meliputi konsep proses bisnis, penulisan kelas-kelas dalam bahasa pemrograman spesifik, skema *database*, dan komponen yang diperlukan dalam sistem *software* [18]. UML merupakan salah satu standar bahasa yang banyak digunakan dalam dunia industri untuk mendefinisikan *requerement*, membuat analisis dan desain, serta menggambarkan arsitektur dalam pemrograman berorientasi objek [19].

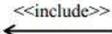
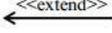
UML merupakan bahasa spesifikasi yang digunakan untuk menspesifikasikan, mendokumentasikan, dan membangun perangkat lunak. UML merupakan bahasa untuk memvisualisasikan, menspesifikasikan, membangun, dan mendokumentasikan berdasarkan grafik atau gambar dari sebuah sistem pengembangan *software* berbasis *object-oriented*. UML memberikan standar penulisan sebuah sistem, meliputi konsep bisnis proses, penulisan kelas-kelas dalam bahasa pemrograman yang spesifik, skema *database*, dan komponen yang diperlukan dalam sistem *software*. Diagram UML antara lain [20]:

1. Use Case Diagram

Use case merupakan abstraksi dari interaksi antara aktor dan sistem. *Use case* bekerja dengan mendeskripsikan tipe interaksi antara pengguna dengan sistem itu sendiri melalui sebuah cerita bagaimana sistem itu digunakan. Diagram *use case* berguna dalam hal menjalankan fasilitas yang ada (*requirements*) dimana *use case* baru selalu menghasilkan fasilitas baru ketika sistem dianalisis dan desain menjadi lebih jelas, penggunaan notasi dan simbol pada diagram membuat pengembang lebih mudah berkomunikasi dengan kliennya [21].

Use case menggambarkan *external view* dari suatu sistem yang akan buat kedalam model. Model *use case* dapat dijabarkan kedalam *use case*, akan tetapi diagram tidak identik dengan model karena diagram lebih sempit daripada model [20]. Tabel 2.2 merupakan simbol yang digunakan pada *use case* diagram :

Tabel 2.2 Diagram Use Case [22].

Simbol	Keterangan
	Aktor : Mewakili peran orang, sistem yang lain, atau alat ketika berkomunikasi dengan <i>use case</i>
	<i>Use case</i> : Abstraksi dan interaksi antara sistem dan aktor
	<i>Association</i> : Abstraksi dari penghubung antara aktor dengan <i>use case</i>
	<i>Generalisasi</i> : Menunjukkan spesialisasi aktor untuk dapat berpartisipasi dengan <i>use case</i>
	Menunjukkan bahwa suatu <i>use case</i> seluruhnya merupakan fungsionalitas dari <i>use case</i> lainnya
	Menunjukkan bahwa suatu <i>use case</i> merupakan tambahan fungsional dari <i>use case</i> lainnya jika suatu kondisi terpenuhi

2. Class Diagram

Class diagram sebagai suatu objek yang memiliki atribut dan perlakuan yang sama, kelas juga terkadang disebut objek [20]. *Class diagram*

merupakan penggambaran dari struktur kelas yang terdapat pada suatu sistem. Tabel 2.3 merupakan simbol yang digunakan pada *class diagram*.

Tabel 2.3 Simbol *Class Diagram*

NO	GAMBAR	NAMA	KETERANGAN
1		<i>Generalization</i>	Hubungan dimana objek anak (<i>descendent</i>) berbagi perilaku dan struktur data dari objek yang ada di atasnya objek induk (<i>ancestor</i>).
2		<i>Nary Association</i>	Upaya untuk menghindari asosiasi dengan lebih dari 2 objek.
3		<i>Class</i>	Himpunan dari objek-objek yang berbagi atribut serta operasi yang sama.
4		<i>Collaboration</i>	Deskripsi dari urutan aksi-aksi yang ditampilkan sistem yang menghasilkan suatu hasil yang terukur bagi suatu actor
5		<i>Realization</i>	Operasi yang benar-benar dilakukan oleh suatu objek.
6		<i>Dependency</i>	Hubungan dimana perubahan yang terjadi pada suatu elemen mandiri (<i>independent</i>) akan memengaruhi elemen yang bergantung padanya elemen yang tidak mandiri
7		<i>Association</i>	Apa yang menghubungkan antara objek satu dengan objek lainnya

3. Activity Diagram

Activity diagram menunjuk kepada suatu aktivitas didalam sistem dengan bentuk kumpulan aksi-aksi, bagaimana dimulai, keputusan yang mungkin dapat terjadi, hingga berakhirnya aksi. *Activity diagram* merupakan aktifitas, objek, transisi *state* dan *event* dengan kata lain kegiatan pada diagram alur kerja menggambarkan perilaku dari sistem untuk aktifitas [20]. Tabel 2.4 merupakan simbol yang digunakan pada *activity diagram*.

Tabel 2.4 *Activity Diagram*

Nama	Simbol	Fungsi
Initial State		Menggambarkan awal dimulainya suatu aliran aktivitas
Final State		Menggambarkan berakhirnya suatu aliran aktivitas
Activity		Menggambarkan aktivitas yang dilakukan dalam suatu aliran aktivitas
Decision		Menggambarkan pilihan kondisi atau cabang-cabang aktivitas tertentu
Transition		Berguna untuk menghubungkan satu komponen dengan komponen lainnya.

4. Sequence Diagram

Sequence diagram merupakan gambaran tahap demi tahap, termasuk urutan perubahan secara logis yang seharusnya dilakukan untuk menghasilkan sesuatu sesuai dengan *use case diagram* [20]. Tabel 2.5 merupakan simbol yang digunakan pada *sequence diagram*.

Tabel 2.5 Simbol *Sequence Diagram*

NO	GAMBAR	NAMA	KETERANGAN
1		<i>Actor</i>	Menggambarkan orang yang sedang berinteraksi dengan sistem.
2		<i>Entity Class</i>	Menggambarkan hubungan yang akan dilakukan
3		<i>Boundary Class</i>	Menggambarkan sebuah gambaran dari foem
4		<i>Control Class</i>	Menggambarkan penghubung antara boundary dengan tabel
5		<i>A focus of Control & A Life Line</i>	Menggambarkan tempat mulai dan berakhirnya message
6		<i>A message</i>	Menggambarkan Pengiriman Pesan

F. Firebase Realtime Database

Firebase merupakan *backend as a Service* yang dimiliki oleh Google saat ini. *Firebase* merupakan solusi yang ditawarkan oleh Google untuk mempermudah pekerjaan pengembangan aplikasi *Mobile*. *Firebase* membuat para pengembang aplikasi tetap fokus tanpa harus memberikan *effort* yang besar dalam urusan *backend*. *Firebase Realtime Database* merupakan *database* berbasis *cloud NoSQL* yang menyingkronkan data pada semua klien secara *realtime* dan menyediakan fungsional secara *realtime*. Data yang telah diinputkan kedalam *database Realtime* akan disimpan dalam bentuk *JSON* [8].

G. Black Box Testing

Black box testing merupakan metode yang digunakan untuk menguji perangkat lunak. Metode pengujian ini dapat diterapkan ke berbagai tingkat pengujian perangkat lunak antara lain : fungsional, integrasi, unit, sistem dan

penerimaan. Pengujian ini memilih *input* yang valid dan tidak kemudian menentukan *output* yang benar [23].

Black box testing adalah pengujian untuk menunjukkan kesalahan pada suatu sistem seperti kesalahan dalam fungsi sistem. Pengujian dilakukan menggunakan masukan data acak dengan tujuan untuk mendapatkan hasil yang pasti. Pasti dalam hal ini artinya ketika terjadi kesalahan, maka sistem akan melakukan penolakan, sedangkan ketika data inputan benar maka dapat diterima atau masuk ke *database* [24].

a. Cara Kerja *Black Box Testing*

Pengujian *black box testing* dilakukan secara *relativ* bertentangan dengan kebutuhan yang ada dan memastikan sistem dapat menangani segala masukan yang tidak sesuai, sehingga *user* dapat memasukan data yang sesuai. Pengujian bertujuan untuk menemukan kesalahan misalnya : fungsi yang tidak benar, kesalahan dalam *interface*, kesalahan dalam struktur data maupun akses database eksternal, dan kesalahan kinerja.

b. Teknik *Black Box Testing*

1. *Descision Table*

Descision table merupakan cara yang digunakan untuk memodelkan logika yang rumit, seperti diagram alur.

2. *All-Pairs Testing*

All-pairs testing merupakan metode pengujian kombinator yang digunakan untuk setiap pasangan parameter termasuk ke dalam sistem atau algoritma suatu perangkat lunak

3. *State Transition Table*

State transition table merupakan suatu teori automata dan logika skuensial, pada suatu tabel yang menunjukkan *state* dalam pengujian. Teknik ini berguna untuk melakukan pengetesan terhadap kondisi mesin dan navigasi [25].

4. *Equivalence Partitioning*

Equivalence partitioning merupakan teknik yang membagikan data masukan dari beberapa unit ke dalam perangkat lunak menjadi

beberapa partisi data dari *test case* dapat diturunkan. Teknik ini dilakukan dengan cara melakukan *partition* atau membagi menjadi beberapa partisi dari data input [26].

5. *Boundry Values Analysis*

Teknik ini merupakan pengujian yang dirancang untuk mencakup perwakilan dari setiap batas nilai. Teknik ini dilakukan dengan cara mencari adakah eror dari luar sistem atau sisi dalam *software*, minimum atau maksimal nilai dari eror yang telah ditemukan [24].