

## **BAB II**

### **TINJAUAN PUSTAKA**

#### **II.1. Penelitian Terkait**

Penelitian terkait bertujuan untuk mengambil beberapa referensi jurnal terkait yang digunakan dalam mendukung penelitian publikasi ilmiah dalam jurnal lokal.

Menurut penelitian yang dilakukan oleh Hendra Purnama, dkk (2015) melakukan penelitian yang berjudul “Perancangan Program Aplikasi Tabungan Siswa Sekolah Dasar Negeri Cipancar IV Dengan Menggunakan Pendekatan Metodologi *Rapid Application Development*”. Berdasarkan hasil perancangan sistem, yaitu data siswa untuk menyimpan data identitas, sedangkan untuk menyimpan data transaksi disimpan dalam entitas tabungan, penyetoran serta pengambilan dan untuk menyimpan laporan kas masuk dan keluar disimpan dalam entitas laporan.

Penelitian yang dilakukan Septika Hapsari, (2011) juga melakukan penelitian yang berjudul “Rancang Bangun Sistem Informasi Pembayaran Dan Tabungan Siswa Pada Bank Mini Artha Mandiri Sekolah Menengah Kejuruan (SMK) Negeri Pringkuku Pacitan”. Untuk menghasilkan aplikasi diimplementasikan Sistem Informasi Pembayaran Dan Tabungan Siswa Pada Bank Mini Artha Mandiri SMK Negeri Pringkuku terbukti bahwa sistem tersebut memberikan efisiensi waktu dalam menangani masalah pembuatan laporan keuangan.

Penelitian yang dilakukan Nisa Nur Pradipta, dkk (2017) juga melakukan penelitian yang berjudul “Aplikasi Tabungan Dan Pinjaman Berbasis Akuntansi Perbankan”. Berdasarkan pembahasan hasil penelitian aplikasi mampu menangani transaksi tabungan yang terdiri dari tarikan dan setoran, pinjaman, pembayaran angsuran beserta bunga serta menghasilkan laporan keuangan berupa laporan laba rugi dan neraca sesuai dengan standar akuntansi.

## **II.2. Uraian Teoritis**

Adapun teori-teori yang digunakan sebagai landasan dalam penulisan proposal skripsi ini adalah sebagai berikut:

### **II.2.1. Rancang Bangun**

Rancang Bangun (desain) adalah tahap setelah analisis dari siklus pengembangan sistem yang merupakan pendefinisian dari kebutuhan fungsional, serta menggambarkan bagaimana suatu sistem dibentuk yang dapat berupa penggambaran, perencanaan dan pembuatan sketsa atau pengaturan dari beberapa elemen yang terpisah ke dalam satu kesatuan yang utuh dan berfungsi, termasuk menyangkut mengkonfigurasi dari komponen-komponen perangkat keras dan perangkat lunak dari suatu sistem (Muhammad Ridwan, dkk ; 2016 : 23).

### **II.2.2. Aplikasi**

Secara istilah pengertian aplikasi adalah suatu program yang siap untuk digunakan yang dibuat untuk melaksanakan suatu fungsi bagi pengguna jasa aplikasi serta penggunaan aplikasi lain yang dapat digunakan oleh suatu sasaran yang akan dituju. Menurut kamus *computer* eksekutif, aplikasi mempunyai arti

yaitu pemecahan masalah yang menggunakan salah satu tehnik pemrosesan data aplikasi yang biasanya berpacu pada sebuah komputansi yang diinginkan atau diharapkan maupun pemrosesan data yang di harapkan.

Pengertian aplikasi menurut Kamus Besar Bahasa Indonesia, “Aplikasi adalah penerapan dari rancang sistem untuk mengolah data yang menggunakan aturan atau ketentuan bahasa pemrograman tertentu” (Andi Juansyah ; 2015 : 2).

### **II.2.3. Tabungan**

Menurut UU Perbankan No.10 Tahun 1998 tabungan adalah simpanan yang pada penarikannya hanya dapat dilakukan menurut syarat tertentu yang telah disepakati, namun tidak dapat ditarik dengan cek, bilyet giro atau alat lainnya yang dipersamakan dengan itu (Akh. Fawaid dan Moh. Holis ; 2019 : 88).

Menurut Soemitro Djojohadikusumo (1954) tabungan didefinisikan sebagai kemampuan dan kesediaan untuk menahan napsu konsumsi selama beberapa waktu agar dimasa yang depan terbuka kemungkinan konsumsi yang memuaskan, Kemudian Pengertian Tabungan menurut Simorangkir (1991 : 47) tabungan diartikan sebagai bagian derajat pendapatan nasional pertahunnya yang tidak dikonsumsi (Akh. Fawaid dan Moh. Holis ; 2019 : 88).

Menurut teori klasik tabungan adalah fungsi dari tingkat bunga. Makin tinggi tingkat bunga makin tinggi pula keinginan masyarakat untuk menabung. Artinya pada tingkat bunga yang lebih tinggi masyarakat akan lebih terdorong untuk mengorbankan atau mengurangi pengeluaran untuk konsumsi guna menambah tabungan (Akh. Fawaid dan Moh. Holis ; 2019 : 88).

#### II.2.4. Pengertian SQLite

Saputro (2013), SQLite merupakan sebuah basis data yang bersifat ACID *compliant* dan memiliki ukuran pustaka kode yang relatif kecil, ditulis dalam bahasa C. SQLite merupakan proyek yang bersifat *public domain* yang dikerjakan oleh D. Ricard Hipp. SQLite memiliki fitur relasional database, hampir sama dengan SQL pada desktop hanya saja SQLite membutuhkan memori sedikit. Pada prinsipnya SQLite merupakan library C yang diimplementasikan secara *embeddable* (tertanam) sebagai SQL *database engine*. Kemasan praktis inilah yang memberikan banyak sekali keuntungan dimana tidak perlu melakukan manajemen *database server* terpisah. Selain itu, ukuran kecil yang sengaja didesain pada SQLite memungkinkan kita untuk membuat aplikasi yang ringan dengan kinerja tinggi (Tuji Rochma Wati dan Heri Sismoro ; 2014 : 62).

#### II.2.5. Pengertian Pemrograman Java

Java menurut defenisi dari *Sun* adalah nama untuk sekumpulan teknologi untuk membuat dan menjalankan perangkat lunak pada komputer personal ataupun pada lingkungan jaringan. Java2 adalah generasi kedua dari *java platform* (generasi awalnya adalah *Java Development Kit*). Java berdiri di atas mesin interpreter yang diberi nama *Java Virtual Machine* (JVM). JVM inilah yang akan membaca *bytecode* dalam *file.class* dari suatu program sebagai sebuah representasi langsung program yang berisi bahasa mesin. Oleh karena itu, bahasa java disebut sebagai bahasa pemrograman yang portable karena dapat dijalankan pada berbagai sistem operasi, asalkan pada sistem operasi tersebut terdapat JVM. *Platform java* memiliki tiga buah edisi yang berbeda, yaitu J2EE (*Java2*

*Enterprise Edition*), J2ME (*Java2 Micro Edition*), dan J2SE (*Java2 Second Edition*) (Satriawaty Mallu ; 2015 : 38).

Java adalah bahasa pemrograman yang dapat dijalankan di berbagai komputer termasuk telepon genggam. Bahasa ini awalnya dibuat oleh James Gosling saat masih bergabung di Sun Microsystems saat ini merupakan bagian dari Oracle dan dirilis tahun 1995 (Satriawaty Mallu ; 2015 : 38).

### **II.2.6. Sejarah Android**

Android adalah sebuah sistem operasi perangkat mobile berbasis linux yang mencakup sistem operasi, middleware dan aplikasi. Android menyediakan platform terbuka bagi para pengembang untuk menciptakan aplikasi mereka.

Awalnya, Google Inc. membeli Android Inc. yang merupakan pendatang baru yang membuat peranti lunak untuk ponsel atau smartphone. Kemudian untuk mengembangkan Android, dibentuklah *Open Handset Alliance*, konsorsium dari 34 perusahaan peranti keras, peranti lunak dan telekomunikasi, termasuk Google, HTC, Intel, Motorola, Qualcomm, T-Mobile, dan Nvidia.

Pada saat perilisan perdana Android, 5 November 2007, Android bersama Open Handset Alliance menyatakan mendukung pengembangan open source pada perangkat mobile. Di lain pihak, Google merilis kode-kode Android di bawah lisensi Apache, sebuah lisensi perangkat lunak dan open platform perangkat seluler (Andi Juansyah ; 2015 : 2).

### II.2.7. Android Studio

Android studio adalah IDE (*Integrated Development Environment*) resmi untuk pengembangan aplikasi Android dan bersifat *open source* atau gratis. Peluncuran Android Studio ini diumumkan oleh Google pada 16 mei 2013 pada *event* Google I/O *Conference* untuk tahun 2013. Sejak saat itu, Android Studio menggantikan Eclipse sebagai IDE resmi untuk mengembangkan aplikasi Android (Andi Juansyah ; 2015 : 2-3). Adapun gambar Android Studio yang dapat dilihat pada gambar II.1.



**Gambar II.1. Android Studio**  
(*Sumber: Andi Juansyah, 2015*)

Android studio sendiri dikembangkan berdasarkan IntelliJ IDEA yang mirip dengan Eclipse disertai dengan ADT *plugin* (*Android Development Tools*).

Android studio memiliki fitur :

- a. Projek berbasis pada Gradle Build
- b. Refactory dan pembenahan bug yang cepat
- c. Tools baru yang bernama “Lint” dikalim dapat memonitor kecepatan, kegunaan, serta kompetibelitas aplikasi dengan cepat.
- d. Mendukung Proguard And App-signing untuk keamanan.
- e. Memiliki GUI aplikasi android lebih mudah.

- f. Didukung oleh *Google Cloud Platform* untuk setiap aplikasi yang dikembangkan (Andi Juansyah ; 2015 : 3).

## II.2.8. *Unified Modeling Language (UML)*

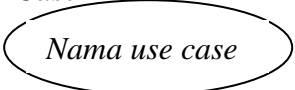
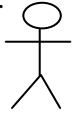
*Unified Modeling Language (UML)* adalah standarisasi bahasa pemodelan untuk membangun perangkat lunak yang dibangun dengan menggunakan teknik pemrograman berorientasi objek.


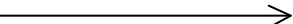

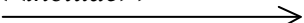

UML merupakan metodologi dalam pengembangan system berorientasi objek dan juga merupakan alat untuk mendukung pengembangan sistem. UML saat ini banyak dipergunakan dalam dunia industri yang merupakan standar bahasa pemodelan umum dalam industry pengembangan sistem (Gellysa Urva dan Helmi Fauzi Siregar; 2015 : 93-94)

### 1. *Use Case Diagram*

*Use Case* mendeskripsikan sebuah interaksi antara satu atau lebih aktor dengan sistem informasi yang akan dibuat. Simbol-simbol *Use Case Diagram* dapat dilihat pada Tabel II.2.

**Tabel II.2. Simbol *Use Case Diagram***

Simbol	Deskripsi
<p><i>Use Case</i></p> 	Fungsionalitas yang disediakan sistem sebagai unit-unit yang saling bertukar pesan antar unit dan aktor.
<p>Aktor/<i>Actor</i></p> 	Orang, proses, atau sistem lain yang berinteraksi dengan sistem informasi yang akan dibuat diluar sistem informasi.


Asosiasi/ <i>association</i> 	Komunikasi antar aktor dan <i>Use Case</i> yang berpartisipasi.
Ekstensi/ <i>extend</i> << <i>extend</i> >> 	Relasi <i>Use Case</i> tambahan ke sebuah <i>Use Case</i> dimana <i>Use Case</i> yang ditambah dapat berdiri sendiri walau tanpa <i>Use Case</i> tambahan.
Generalisasi/ <i>generalization</i> 	Hubungan generalisasi dan spesialisasi antara dua buah <i>Use Case</i> yang mana fungsi yang satu lebih umum dari yang lainnya.
Menggunakan <i>include/Use Case</i> << <i>include</i> >>  uses 	Relasi <i>Use Case</i> tambahan ke sebuah <i>Use Case</i> dimana <i>Use Case</i> yang ditambahkan memerlukan <i>Use Case</i> ini untuk menjalankan fungsinya.


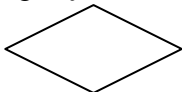


(Sumber: Winda Aprianti dan Umi Maliha, 2016)

## 2. Activity Diagram

*Activity Diagram* menggambarkan *workflow* (aliran kerja) dari sebuah sistem atau proses bisnis atau menu yang ada pada perangkat lunak. Simbol-simbol *Activity Diagram* dapat dilihat pada Tabel II.3.

**Tabel II.3. Simbol Activity Diagram**

Simbol	Deskripsi
Status awal 	Status awal aktivitas pada sebuah diagram aktivitas memiliki sebuah status awal.



Aktivitas 	Aktivitas yang dilakukan sistem, aktivitas biasanya diawali dengan kata kerja.
Percabangan/ <i>join</i> 	Asosiasi percabangan dimana jika ada pilihan aktivitas lebih dari satu.
Penggabungan/ <i>join</i> 	Asosiasi penggabungan dimana lebih dari satu aktivitas digabungkan menjadi satu.
Status akhir 	Status akhir yang dilakukan sistem, sebuah diagram aktivitas memiliki sebuah status akhir.

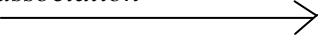

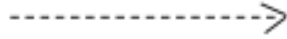

(Sumber: Winda Aprianti dan Umi Maliha, 2016)

### 3. Class Diagram

*Class diagram* menggambarkan struktur sistem dari segi pendefinisian kelas-kelas yang akan di buat untuk membangun sistem. Kelas memiliki apa yang disebut atribut dan metode atau operasi. Simbol-simbol yang ada pada *Class diagram* ditunjukkan oleh Tabel II.1.

**Tabel II.1. Simbol Class Diagram**

Simbol	Deskripsi
Kelas 	Kelas pada struktur system
Antarmuka / <i>interface</i>  <b>Nama <i>interface</i></b>	Sama dengan konsep <i>interface</i> dalam pemrograman berorientasi objek.
<u>Asosiasi/association</u>	Relasi antar kelas dengan makna umum, asosiasi biasanya disertai dengan <i>multiplicity</i> .

<p>Asosiasi Berarah/<i>directed association</i></p> 	<p>Relasi antar kelas dengan makna kelas yang satu digunakan oleh kelas yang lain, asosiasi biasanya juga disertai dengan <i>multiplicity</i>.</p>
<p>Generalisasi</p> 	<p>Relasi antar kelas dengan makna generalisasi spesialisasi (umum khusus).</p>
<p>Kebergantungan/<i>dependency</i></p> 	<p>Relasi antar kelas dengan makna kebergantungan antarkelas.</p>
<p>Agregasi/ <i>aggregation</i></p> 	<p>Relasi antar kelas dengan makna semua-bagian.</p>

(Sumber: Winda Aprianti dan Umi Maliha, 2016)