

BAB II

TINJAUAN PUSTAKA

2.1 Penelitian Terdahulu

Penelitian terdahulu ini menjadi salah satu acuan penulis dalam membuat laporan akhir sehingga dapat memperkaya teori yang digunakan dalam mengkaji penelitian yang dilakukan. Penelitian terdahulu berupa beberapa jurnal yang terkait dengan judul laporan akhir penulis, antara lain :

Penelitian Ismanto dan Pungkasanti (2021) yang berjudul “Aplikasi Perpustakaan Berbasis *Unified Modelling Language*” hasil dari penelitian ini adalah pembangunan sistem informasi perpustakaan menggunakan *Microsoft Visual Basic 2010* dan *Microsoft Access 2013* pada SMA Islam Nahdlatul Ulama. Sistem perpustakaan dapat membantu petugas perpustakaan untuk mencatat data transaksi peminjaman buku, mengelola daftar buku dan jumlah stok buku yang dimiliki SMA Islam Nahdlatul Ulama.

Penelitian Ismail dan Iahan (2021) yang berjudul “Aplikasi Peminjaman Buku Pada Perpustakaan Menggunakan Kartu Tanda Siswa (KTS)” ini membahas tentang petugas perpustakaan sekolah SMA Negeri 3 Parepare yang melakukan 2 kali pengerjaan, dimana aktivitasnya masih manual setelah peminjam buku tanda tangan dibuku besar petugas akan menginputkan data kedalam *Microsoft Excel*. Dengan buku besar ini sama saja menghabiskan kertas yang banyak dengan data yang banyak akan menumpuk nantinya, oleh karena itu digunakannya suatu sistem baru aplikasi desktop yang terintegrasi dengan RFID dan *Barcode* untuk mengurangi pemakaian kertas dan data yang tersimpan pun lebih terjamin aman.

Penelitian Yunus dan Adisal (2021) dengan judul “Aplikasi Perpustakaan Menggunakan *Quick Response Code*” hasil yang diperoleh dari penelitian ini adalah aplikasi yang dibuat mempermudah staf perpustakaan untuk melakukan *system* peminjaman. Aplikasi ini telah diimplementasi sebagai sumber informasi perpustakaan desa. Aplikasi yang dirancang terdiri dari beberapa halaman diantaranya *form login*, *form menu admin*, *form staff* perpustakaan, dan *form user* atau peminjaman. Teknologi yang digunakan untuk merancang aplikasi

perpustakaan adalah *Quick Response Code* berbasis *online* dan bahasa pemrograman PHP dan MySQL.

Penelitian yang berjudul “Implementasi RFID Dan Nodemcu Untuk Data Kunjungan Perpustakaan Berbasis IoT” oleh Khadafi, Darusalam dan Winarsih (2020) ini bertujuan untuk membuat perpustakaan sekolah berbasis web yang dapat memberikan informasi bagi para pengunjung. Agar memudahkan pencarian data perpustakaan pada sekolah SMK Yadika 13 Tambun Utara merancang website sistem informasi yang dapat mencari data melalui website tanpa mencari data secara manual. Supaya data yang masuk ke website diambil secara otomatis maka dirancang RFID dan NodeMCU sebagai mikrokontroler yang berbasis IoT. Dengan membangun sistem menggunakan bahasa pemrograman Javascript dan HTML system dapat dimonitor secara realtime. Penerapan RFID dan Web sangat menghemat waktu karena data yang telah diinput secara otomatis tersimpan dan dapat di baca kembali oleh system melalui tag RFID.

Penelitian Pratama dan Muludi (2021) yang berjudul “Aplikasi Perpustakaan Digital Pada Perpustakaan Jurusan Ilmu Komputer Universitas Lampung Berbasis Android” ini mendapatkan kesimpulan bahwa aplikasi yang dibuat dapat menampilkan informasi tentang data buku perpustakaan baik yang tersedia maupun yang masih dalam peminjaman. Sistem dari aplikasi perpustakaan digital mempermudah *admin* untuk manajemen data buku dan memudahkan pembuatan laporan perpustakaan. Aplikasi itu terdiri dari menu *login*, menu utama, menu peminjaman dan menu denda. Teknologi yang digunakan adalah basis data *online*, sehingga memerlukan jaringan internet dan berbasis android dapat dijalankan pada perangkat bergerak.

Dari 5 penelitian terdahulu tersebut belum ada penelitian yang membuat Aplikasi Peminjaman Buku menggunakan teknologi bahasa pemrograman Java Netbeans dan *database management system* MySQL yang terhubung dengan alat validasi keanggotaan perpustakaan. Kelebihan dari penelitian ini nanti adanya penerapan protokol kesehatan.

2.2 Aplikasi

Abdurahman dan Riswaya mengatakan aplikasi adalah program siap pakai yang dapat digunakan untuk menjalankan perintah-perintah dari pengguna aplikasi tersebut dengan tujuan mendapatkan hasil yang lebih akurat sesuai dengan tujuan pembuatan aplikasi tersebut. Pengertian aplikasi secara umum adalah alat terapan yang difungsikan secara khusus dan terpadu sesuai kemampuan yang dimilikinya, aplikasi merupakan suatu perangkat komputer yang siap pakai bagi *user*. (Simargolang & Nasution, 2018)

1.3 Basis Data (*Database*)

Stephens dan Plew mengatakan basis data adalah mekanisme yang digunakan untuk menyimpan informasi atau data. Informasi adalah sesuatu yang kita gunakan sehari-hari untuk berbagai alasan. Dengan basis data, pengguna dapat menyimpan data secara terorganisasi. Setelah data disimpan, informasi harus mudah diambil. Kriteria dapat digunakan untuk mengambil informasi. Cara data disimpan dalam basis data menentukan seberapa mudah mencari informasi berdasarkan banyak kriteria. Data pun harus mudah ditambahkan kedalam basisdata, dimodifikasi dan dihapus. (Supriyanto, 2015).

2.4 Java

Java adalah nama sebuah bahasa pemrograman yang sangat populer. Sebagai bahasa pemrograman, java bisa digunakan untuk menulis program. Bahasa java dikembangkan oleh *Sun Microsystem* dan mulai diperkenalkan kepada public pada tahun 1995. Java merupakan bahasa yang berorientasi objek. Java memiliki keunggulan yakni bersifat *universal*. Sebagai bahasa yang *universal*, java bisa dijumpai di berbagai *platform* (Linux, UNIX, Windows, Mac OS, dan lain-lain). Hasil kompilasi java yang dinamakan bytecode bisa dijalankan di berbagai *platform* sepanjang sistem memiliki *Java Runtime Environment (JRE)*. (Zefriyenni dalam Frakas, 2020)

1.5 Netbeans

Integrated Development Environment (IDE) NetBeans merupakan sebuah *tools* untuk programmer menulis, mengompilasi program. IDE NetBeans ditulis dalam Bahasa *Java*, namun dapat mendukung bahasa pemrograman lain. Terdapat banyak modul untuk memperluas IDE NetBeans. NetBeans mengacu pada dua hal, yakni *platform* untuk pengembangan *desktop java*, dan sebuah IDE yang dibangun menggunakan *platform NetBeans*. *Platform NetBeans* memungkinkan aplikasi yang dibangun dari sekumpulan komponen-komponen perangkat lunak modular yang disebut ‘modul’. Sebuah modul merupakan suatu *Arsip Java (Java Archive)* yang membuat kelas-kelas *java* untuk berinteraksi dengan *NetBeans Open API* dan *file manifestasi* yang mengidentifikasikannya sebagai modul. Aplikasi yang dibangun dengan modul-modul dapat dikembangkan dengan menambahkan modul-modul baru. Karena modul dapat dikembangkan secara independen, aplikasi berbasis *platform NetBeans* dapat dengan mudah dikembangkan oleh pihak ketiga secara mudah dan *powerful*. (Afrizal dalam Frakas, 2020)

1.6 MySQL

Fahrudin (dalam Siregar dan Sari, 2018) berpendapat *my structured query language* (MySQL) adalah sebuah program pembuat dan pengelola *database* atau yang sering disebut *Database Management System* (DBMS). Sifat dari DBMS ini adalah *open source*. MySQL juga merupakan program pengakses *database* yang bersifat jaringan, sehingga dapat digunakan untuk aplikasi *multi user* (banyak pengguna). Kelebihan lain dari MySQL adalah menggunakan bahasa *query* (permintaan) standar SQL. SQL adalah suatu bahasa permintaan yang terstruktur, SQL telah distandarkan untuk semua program pengakses *database*.

1.7 XAMPP

Dwi Priyanti mengatakan XAMPP adalah perangkat lunak bebas, yang mendukung banyak sistem operasi, merupakan kompilasi dari beberapa program. Fungsinya adalah sebagai *server* yang berdiri sendiri (*localhost*), yang terdiri atas program *apache HTTP Server*, *MySQL database*, dan penerjemah bahasa yang

ditulis dengan bahasa pemrograman PHP dan Perl. Nama XAMPP merupakan singkatan dari X (empat sistem operasi apapun), *Apache*, MySQL, PHP dan Perl. Program ini tersedia dalam GNU (*General Public License*), merupakan *web server* yang mudah digunakan yang dapat melayani tampilan halaman web yang dinamis. (Siregar dan Sari, 2018).

1.8 PhpMyAdmin

Saputro (dalam Kurniawan dan Fachrurrazi, 2019) mengatakan setiap RDBMS (*Relation Database Managemen Systems*) seperti Oracle, *SQL Server*, MySQL dan lain-lain, pasti memiliki tool yang dapat digunakan untuk mempermudah pengoperasian *database*. Oracle memiliki TOAD. *SQL Server* memiliki *Enterprise Manager* dan *SQL Query Analyzer*. Sedangkan MySQL memiliki tool atau aplikasi yang disebut PhpMyAdmin. PhpMyAdmin merupakan aplikasi berbasis web yang dikembangkan menggunakan bahasa pemrograman PHP. Melalui PhpMyAdmin, *user* dapat melakukan perintah Query tanpa harus mengetikkan seperti pada MS DOS. Perintah tersebut misalnya administrasi *User* dan *privileges*, *export* dan *import database*, manajemen *database*, manajemen label dan struktur *table*, dan sebagainya. PhpMyAdmin sangat *user friendly*, sangat mudah untuk digunakan walau pengguna baru (*newbie*)

2.9 Unified Modeling Language (UML)








Shalahuddin & Rosa mengemukakan bahwa *Unified Modeling Language* (UML) adalah standarisasi bahasa pemodelan buat membentuk suatu aplikasi yang dibangun dengan menggunakan teknik pemrograman berorientasi objek. (Aprianti dalam Frakas, 2020)

Diagram-diagram yang dipergunakan pada UML diantaranya ialah *uce case diagram*, *class diagram*, *activity diagram* serta *sequence diagram*.

1. Use Case Diagram

Use Case Diagram ialah sebuah interaksi antara satu atau lebih aktor menggunakan sistem yang akan dirancang.

Tabel 2.2 Simbol-Simbol *Use Case Diagram*

No	Simbol	Nama	Keterangan
1.		<i>Actor</i>	Orang, proses atau sistem yang berinteraksi dengan sistem informasi yang akan dibuat di luar sistem informasi.
2.		<i>Use Case</i>	Fungsionalitas yang disediakan sistem menjadi unit-unit yang saling bertukar pesan antar unit dan aktor.
3.		<i>Subsystem</i>	Menspesifikasikan paket yang akan ditampilkan oleh sistem secara terbatas.
4.		<i>Association</i>	Komunikasi antar aktor dan <i>use case</i> yang berpartisipasi
5.		<i>Dependency</i>	Hubungan dimana perubahan yang terjadi pada suatu elemen mandiri (<i>independent</i>) akan mempengaruhi elemen yang bergantung pada elemen yang tidak mandiri (<i>independent</i>).
6.		<i>Generalization</i>	Hubungan generalisasi dan spesialisasi antara dua buah <i>use case</i> yang mana fungsi yang satu lebih umum dari yang lainnya
7.		<i>Include</i>	Hubungan antara <i>use case</i> dengan <i>use case</i> , <i>include</i> menyatakan bahwa sebelum

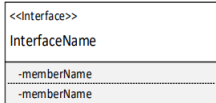

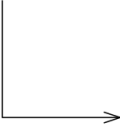
			pekerjaan dilakukan harus mengerjakan pekerjaan lain terlebih dahulu.
8.	--<<extend>>->	<i>Extend</i>	Hubungan antara <i>use case</i> , <i>extend</i> menyatakan bahwa jika pekerjaan yang dilakukan tidak sesuai atau terdapat kondisi khusus, maka lakukan pekerjaan itu.

(Sumber: Aprianti dalam Frakas, 2020)

2. Class Diagram

Class Diagram merupakan gambaran dari struktur sistem atau pendefinisian dari kelas-kelas yang akan dirancang untuk membentuk sebuah sistem.

Tabel 2.3 Simbol-Simbol *Class Diagram*

No	Simbol	Nama	Keterangan
1.		<i>Class</i>	Kelas pada struktur system.
2.		<i>Interface</i>	Sama dengan konsep <i>interface</i> dalam pemrograman berorientasi objek.
3.		<i>Association</i>	Relasi antar kelas dengan makna umum, asosiasi biasanya disertai dengan <i>multiplicity</i> .
4.		<i>Directed Association</i>	Relasi antar kelas dengan makna kelas yang satu digunakan oleh kelas yang lain, asosiasi biasanya juga disertai dengan <i>multiplicity</i> .

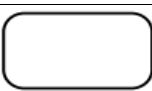
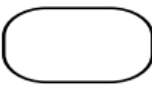
5.		<i>Aggregation</i>	Relasi antar kelas dengan makna semua bagian (<i>whole-part</i>)
6.		<i>Dependency</i>	Relasi antar kelas dengan makna kebergantungan antar kelas.
7.		<i>Interface Realization</i>	perasi yang benar-benar dilakukan oleh suatu objek.
8.		<i>Collaboration</i>	Deskripsi dari urutan aksi-aksi yang ditampilkan sistem yang menghasilkan suatu hasil yang terukur bagi suatu aktor.
9.		<i>Generalization</i>	Relasi antar kelas dengan makna generalisasi-spesialisasi (umum khusus).




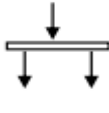
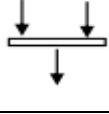
(Sumber: Aprianti dalam Frakas, 2020)

3. Activity Diagram

Activity Diagram adalah gambaran *workflow* (aliran kerja) dari sebuah sistem yang terdapat pada perangkat lunak.

Tabel 2.4 Simbol-Simbol *Activity Diagram*

No	Simbol	Nama	Keterangan
1.		<i>Activity</i>	Aktivitas yang dilakukan sistem, aktivitas biasanya diawali dengan kata kerja
2.		<i>Action</i>	State dari sistem yang mencerminkan eksekusi dari suatu aksi.


3.		<i>Decision</i>	Asosiasi percabangan dimana jika ada pilihan aktivitas lebih dari satu.
4.		<i>Initial Node</i>	Status awal aktivitas pada sebuah diagram aktivitas memiliki sebuah status awal.
5.		<i>Final Node</i>	Status akhir yang dilakukan sistem, sebuah diagram aktivitas memiliki sebuah status akhir.
6.		<i>Fork Node</i>	Satu aliran yang pada tahap tertentu berubah menjadi beberapa aliran.
7.		<i>Join Node</i>	Asosiasi penggabungan dimana lebih dari satu aktivitas digabungkan menjadi satu.

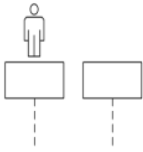


(Sumber: Aprianti dalam Frakas, 2020)

4. *Sequence Diagram*

Sequence Diagram merupakan gambaran dari kelakuan objek pada *use case* yang mendeskripsikan waktu hidup objek dan *message* yang dikirimkan serta diterima antar objek.

Tabel 2.5 Simbol-Simbol *Sequence Diagram*

No	Simbol	Nama	Keterangan
1.		<i>Activation</i>	Menunjukkan masa hidup dari objek.

2.		<i>Object Lifeline / Actor Lifeline</i>	Sebuah objek yang berasal dari kelas atau dapat dinamai dengan kelasnya saja. Aktor termasuk objek. Garis putus-putus menunjukkan garis hidup suatu objek.
3.		<i>Message</i>	Interaksi antara satu objek dengan objek lainnya. Objek dapat mengirimkan pesan ke objek lain. Interaksi antar objek ditunjukkan pada bagian operasi pada diagram kelas.
4.		<i>Return Message</i>	Pesan kembalian dari komunikasi objek.

(Sumber: Aprianti dalam Frakas, 2020)