

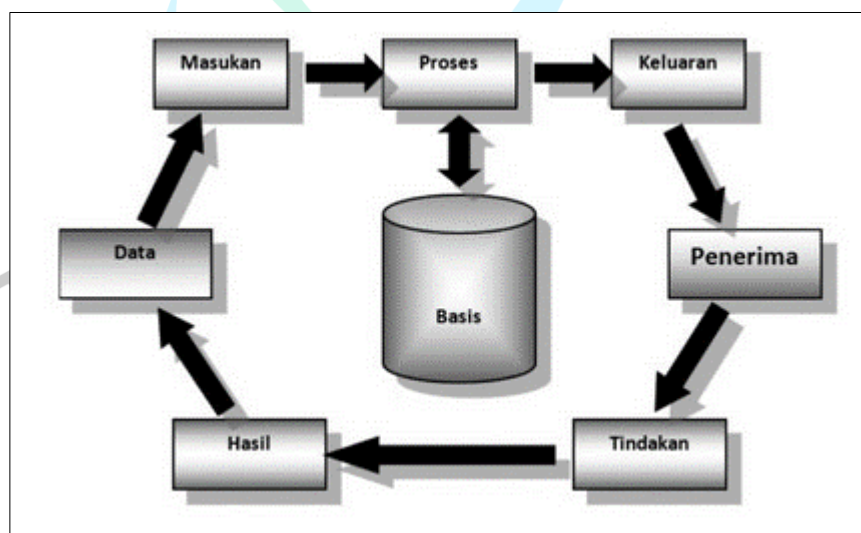
## BAB II

### TINJAUAN PUSTAKA

#### 2.1 Landasan Teori

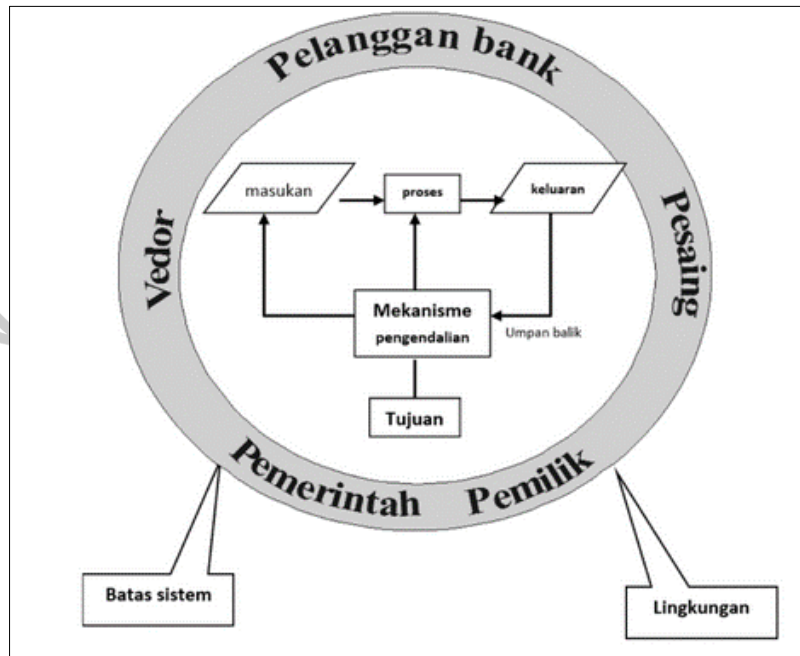
##### 2.1.1 Sistem Informasi (SI)

Anggraeni (2018) mengungkapkan bahwa sistem merupakan sekumpulan orang yang saling bekerja sama dengan aturan yang terstruktur untuk mencapai satu tujuan yang sama, sedangkan informasi adalah data yang diolah agar menjadi bentuk yang bermanfaat bagi penggunanya. Menggabungkan kedua definisi tersebut maka Sistem Informasi (SI) yaitu sebuah kombinasi teratur dari Sumber Daya Manusia (SDM), perangkat lunak, perangkat keras serta sumber daya data yang dikelola untuk dikumpulkan, diubah, dan disebarkan menjadi sebuah informasi di suatu organisasi. Informasi tersebut digunakan untuk mengambil tindakan keputusan sehingga menghasilkan hasil data sehingga menghasilkan sebuah siklus sistem informasi seperti pada Gambar 2.1.



Gambar 2. 1 Siklus Informasi  
Sumber: Pengantar Sistem Informasi, 2018.

Adapun karakteristik yang membentuk suatu sistem pada Gambar 2.2 antara lain:

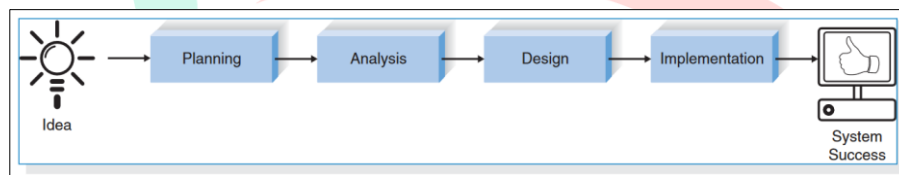


Gambar 2. 2 Karakteristik Sistem  
 Sumber: Pengantar Sistem Informasi, 2018.

- 1) **Input:** Energi yang memiliki atribut jumlah dan ongkos menjadi masukan ke sistem untuk memberi sinyal untuk berinteraksi dengan proses.
- 2) **Process:** Proses mengelola energi untuk mengubah masukan menjadi keluaran sesuai dengan tujuan yang ingin dicapai sebuah sistem.
- 3) **Output:** Energi yang dikelola proses diklasifikasi menjadi keluaran agar memberi hasil bermanfaat.
- 4) **Boundary:** Daerah yang menentukan apa yang termasuk dalam sistem atau tidak, serta memisahkan sistem dari lingkungan luar.
- 5) **Enviroment:** Lingkungan yang ada di luar sistem tetapi dapat mempengaruhi sistem seperti pelanggan, pesaing, jasa bisnis, pemerintah, dll.

## 2.1.2 System Development Life Cycle (SDLC)

Dennis (2015) mengungkapkan bahwa SDLC merupakan proses untuk menentukan bagaimana suatu Sistem Informasi (SI) agar dapat mendukung kebutuhan bisnis, merancang dan membangun sistem dari bisnis tersebut, lalu penggunanya dapat mengoperasikan sistem tersebut. Sistem Analisis umumnya menjadi kunci penting dalam menganalisis situasi bisnis, mencari perbaikan dan merancang SI pada proses SDLC. Membangun SI menggunakan SDLC mengikuti empat fase dasar yaitu perencanaan, analisis, desain dan implementasi seperti pada Gambar 2.3. Fase SDLC dijalankan menggunakan alur logis dari awal sampai akhir maupun secara, iteratif, inkremental dll, namun secara umum memiliki empat fase dasar sebelumnya. Berikut penjelasan dari fase dasar SDLC:



Gambar 2. 3 Fase Dasar SDLC

Sumber: *System Analysis and Design Fifth Edition*, 2015.

### 1) Planning

Fase *planning* atau perencanaan merupakan proses untuk memahami mengapa sebuah SI harus dibangun menggunakan proses inisiasi proyek dan manajemen proyek. Inisiasi proyek memerlukan analisa kelayakan atau *feasability analysis* mengenai aspek teknis (apakah bisa dibangun?), ekonomi (apakah memperoleh keuntungan?) dan organisasi (jika dibangun, apakah akan digunakan?). Setelah proyek disetujui berdasarkan analisa kelayakan, proses selanjutnya yaitu manajemen proyek. Manajemen proyek meliputi proses pembuatan rencana kerja, mengatur staf proyek dan mengarahkan proyek sesuai dengan SDLC. Hasil dari manajemen proyek berupa rencana proyek yang menjelaskan bagaimana tim akan mengembangkan sistem.

## 2) Analysis

Fase *analysis* atau analisis merupakan proses untuk menentukan apa saja yang bisa dilakukan oleh sistem, siapa yang menggunakannya serta dimana dan kapan sistem itu digunakan. Fase ini memiliki tiga langkah yaitu strategi analisis, pengumpulan kebutuhan serta pengembangan proposal sistem.

Langkah pertama yaitu strategi analisis dikembangkan untuk memberi panduan kepada staf proyek. Strategi analisis umumnya meliputi studi sistem berjalan atau saat ini (*as-is system*) dan membuat visualisasi baru yang dikembangkan (*to-be system*).

Langkah selanjutnya adalah pengumpulan kebutuhan, umumnya meliputi wawancara, workshop atau kuesioner. Melalui pengumpulan informasi tersebut dianalisis bersama beserta sponsor proyek untuk mengembangkan konsep sistem baru.

Langkah terakhir adalah pengembangan proposal sistem yaitu menggabungkan analisis sistem dan konsep sistem untuk dipresentasikan kepada sponsor proyek. Hal ini dilakukan untuk menentukan keputusan apakah proyek layak dilanjutkan.

## 3) Design

Fase *design* atau desain merupakan proses untuk memutuskan bagaimana sistem akan beroperasi dengan hardware, software dan infrastruktur jaringan. Selain itu juga memutuskan untuk menggunakan antarmuka, basis data dan form apa yang digunakan oleh sistem baru. Fase ini meliputi langkah seperti:

- a) Strategi desain menentukan apakah sistem dikembangkan oleh orang dalam perusahaan, *outsourcing* atau membeli perangkat lunak.
- b) Desain arsitektur sistem mendeskripsikan desain antarmuka untuk perangkat keras, perangkat lunak dan infrastruktur jaringan.

- c) Spesifikasi basis data dan spesifikasi file dikembangkan untuk mendefinisikan data apa yang disimpan dan dimana data tersebut tersimpan.
- d) Desain program dibuat oleh tim analis untuk mendefinisikan program apa yang ditulis beserta fungsinya.

#### 4) **Implementation**

Fase *implementation* atau implementasi merupakan fase terakhir dimana sistem sebenarnya dibangun atau dibeli dan diinstal. Fase ini umumnya menjadi perhatian lebih dikarenakan merupakan bagian terlama dan termahal dari proses pengembangan sistem. Fase ini memiliki tiga langkah yaitu:

- a) Kontruksi sistem yaitu proses membangun dan menguji sistem baru yang bertujuan untuk memastikan bahwa sistem tersebut berfungsi sesuai dengan desainnya.
- b) Instalasi sistem yaitu proses untuk mematikan sistem lama dan menghidupkan proses baru. Rencana pelatihan diperlukan pada proses ini untuk mengajarkan pengguna cara menggunakan sistem yang baru.
- c) Rencana dukungan sistem yaitu proses untuk melakukan evaluasi kembali untuk mengidentifikasi perubahan yang diperlukan pada sistem baru tersebut.

Fase SDLC memiliki empat fase dasar di atas sebagai pondasi dalam mengembangkan sistem informasi. Sebuah metodologi merupakan pendekatan formal yang digunakan untuk menerapkan SDLC. Metodologi pengembangan sistem bervariasi, maka dari itu terdapat kriteria untuk memilih metodologi pengembangan dapat dilihat pada Tabel 2.1.

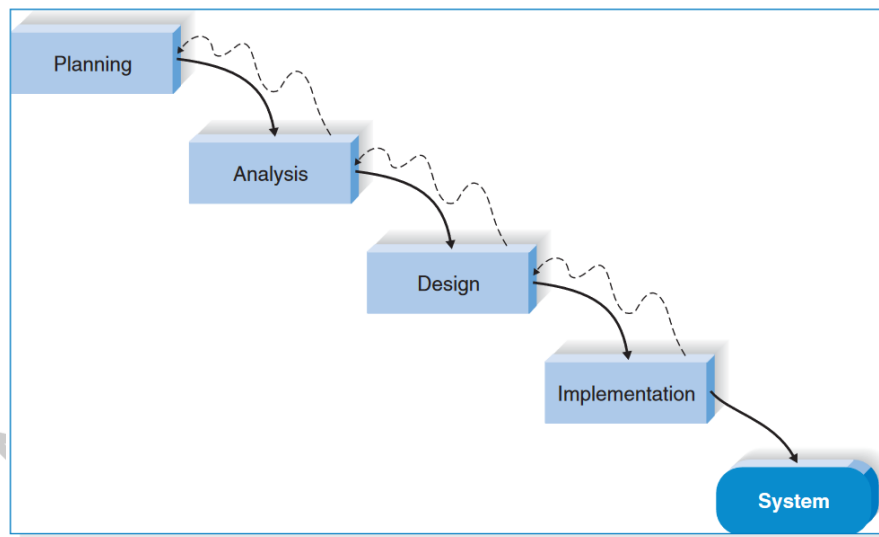
Tabel 2. 1 Kriteria dalam memilih Metodologi SDLC

Usefulness in Developing Systems	Waterfall	Pararell	V-Model	Iterative	System Prototyping	Throwaway Prototyping	Agile Development
Unclear User Requirements?	Poor	Poor	Poor	Good	Excellent	Excellent	Excellent
Unfamiliar Technology?	Poor	Poor	Poor	Good	Poor	Excellent	Poor
Are Complex?	Good	Good	Good	Good	Poor	Excellent	Poor
Are Reliable?	Good	Good	Excellent	Good	Poor	Excellent	Good
Short Time Schedule?	Poor	Good	Poor	Excellent	Excellent	Good	Excellent
With Schedule Visibility?	Poor	Poor	Poor	Excellent	Excellent	Good	Good

Sumber: *System Analysis and Design Fifth Edition*, 2015.

- 1) **Unclear User Requirements?:** kriteria ini menjelaskan se jelas apa kebutuhan pengguna dalam sistem yang dikembangkan.
- 2) **Unfamiliar Technology?:** kriteria ini menjelaskan sejauh mana pengguna mengenal metodologi sistem yang dikembangkan.
- 3) **Complex System?:** kriteria ini menjelaskan serumit apa sistem yang dikembangkan.
- 4) **Reliable System?:** kriteria ini menjelaskan seberapa handal sistem yang dikembangkan
- 5) **Short Time Schedule?:** kriteria ini menjelaskan apakah proyek ini cocok untuk jangka waktu yang pendek.
- 6) **Schedule Visibility?:** kriteria ini menjelaskan seberapa mudah untuk memantau jadwal proyek pada sistem yang dikembangkan.

Berdasarkan pertimbangan kriteria di atas metodologi yang cocok dalam laporan ini yaitu Waterfall dapat dilihat pada Gambar 2.4:



Gambar 2. 4 Metodologi SDLC Waterfall  
 Sumber: *System Analysis and Design Fifth Edition*, 2015.

Metodologi Waterfall merupakan pendekatan SDLC yang bersifat, sehingga analisis menyelesaikan tahapan pada satu fase terlebih dahulu sebelum menuju ke fase berikutnya. Oleh karena itu laporan ini menggunakan Waterfall:

- 1) **Unclear User Requirements?** Poor: dikarenakan kebutuhan pengguna sudah direncanakan dengan detail pada tahap awal pengembangan sistem sehingga tidak memerlukan *prototype*.
- 2) **Unfamiliar Technology?** Poor: dikarenakan pengembang sudah familiar dengan *Google Books API* dan *Barcode Scanner* serta *stakeholder* merupakan orang dari bidang pustakawan.
- 3) **Complex System?** Good: dikarenakan sistem ini memerlukan integrasi yang kompleks yaitu *Barcode Scanner* dengan *Google Books API* dengan *Framework CodeIgniter* untuk mengelola buku dengan modul pinjam.
- 4) **Reliable System?** Good: dikarenakan *stakeholder* memerlukan sistem yang siap untuk digunakan secara 24 jam dan mudah untuk diinstal.
- 5) **Short Time Schedule?** Poor: dikarenakan *stakeholder* memberikan jadwal waktu yang cukup lama sekitar enam bulan terhadap pengembangan sistem.
- 6) **Schedule Visibility?** Poor: dikarenakan keterlibatan *stakeholder* hanya memantau secara langsung pada saat fase perencanaan serta fase pengujian saat implementasi.

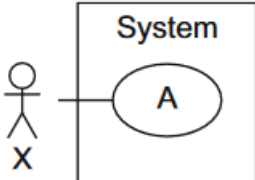
### 2.1.3 Object-Oriented Analysis Diagram (OOAD)

Martin (2004) mengungkapkan bahwa OOAD adalah metode perancangan sistem yang berfokus pada objek. Objek sendiri merupakan representasi dari entitas dunia nyata yang memiliki atribut dan operasi. Atribut adalah isi data yang mendeskripsikan sebuah objek. Operasi adalah fungsi atau tindakan yang dilakukan objek tersebut. Objek dikategorikan ke dalam kelas sesuai dengan masing-masing atribut dan operasi. Analisis dalam OOAD adalah memahami dan mendefinisikan tujuan yang akan dikerjakan. Desain OOAD dilakukan untuk mendefinisikan solusi terhadap tujuan yang ingin dicapai. OOAD juga memiliki enkapsulasi yang memungkinkan untuk membatasi akses kelas, objek dan atribut tertentu menggunakan *modifier* seperti *public*, *private* atau *protected*. OOAD umumnya divisualisasikan dengan UML atau *Unified Modeling Language*.

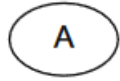
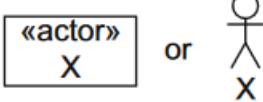
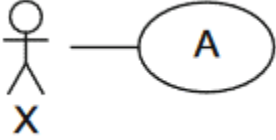
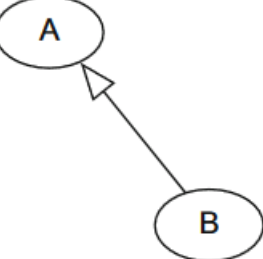
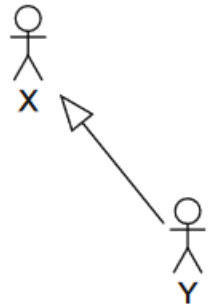
UML merupakan bahasa grafis berbentuk diagram yang digunakan untuk mengkomunikasikan alur dan perilaku antar objek. UML muncul atas penggabungan dari Notasi Booch oleh Grady Booch dan *Object Modeling Technique* (OMT) oleh James Rumbaugh sehingga muncul sekumpulan diagram yang umumnya digunakan untuk perancangan sistem antara lain:

- 1) *Use Case Diagram* yaitu diagram yang menggambarkan perilaku oleh entitas di luar sistem sehingga berguna untuk menjelaskan kebutuhan fungsional. Berikut adalah komponen utama yang membentuk sebuah diagram *use case* dapat dilihat di Tabel 2.2.

Tabel 2. 2 Komponen *Use Case Diagram*

No	Simbol	Keterangan
1		<p>Simbol persegi panjang yang menggambarkan boundary atau batas untuk mengelompokkan <i>use case</i> dalam lingkungan sistem tertentu. -</p> <p>Contoh system: administrasi mahasiswa</p>



2	<p><i>Use Case</i></p> 	<p>Simbol oval yang berisi perintah dari sistem yang dikendalikan oleh <i>actor</i>.</p> <p>Contoh <i>use case</i>: profesor dapat melihat data mahasiswa dan membuat jadwal ujian.</p>
3	<p><i>Actor</i></p> 	<p>Simbol <i>stick figure</i> atau persegi entitas yang berinteraksi dengan sistem.</p> <p>Entitas bisa berupa manusia (contoh: mahasiswa atau profesor) atau non-manusia (contoh: email atau server).</p>
4	<p><i>Association</i></p> 	<p>Simbol garis yang menggabungkan actor dengan use case dalam sistem.</p> <p>Contoh <i>association</i>: profesor dapat melihat data mahasiswa dan membuat jadwal ujian.</p>
5	<p><i>Use Case Generalization</i></p> 	<p>Simbol panah yang berfungsi untuk menurunkan perintah dari satu <i>use case</i> ke <i>use case</i> lainnya.</p> <p>Contoh: <i>use case</i> B memiliki perintah yang diturunkan dari <i>use case</i> A.</p>
6	<p><i>User Generalization</i></p> 	<p>Simbol panah yang berfungsi untuk menurunkan perintah dari satu <i>actor</i> ke <i>actor</i> lainnya.</p> <p>Contoh: <i>actor</i> Y memiliki perintah yang diturunkan dari <i>actor</i> X.</p>



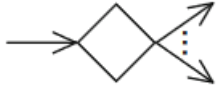
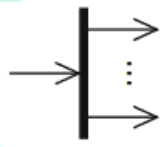
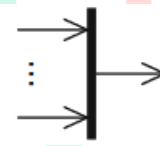
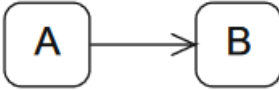
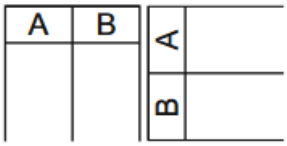
7	<p><i>Extend Relationship</i></p>	<p>Simbol panah dengan garis putus-putus dari <i>use case</i> B ke <i>use case</i> A mendeskripsikan bahwa suatu <i>use case</i> dapat menambah perintah dari <i>use case</i> lain secara opsional.</p> <p>Contoh: <i>actor</i> pelanggan dapat memasukan kupon jika ada sebelum melakukan check out pembayaran.</p>
8	<p><i>Include Relationship</i></p>	<p>Simbol panah dengan garis putus-putus dari <i>use case</i> A ke <i>use case</i> B mendeskripsikan bahwa suatu <i>use case</i> dependensi dari <i>use case</i> lainnya untuk menyelesaikan sebuah <i>task</i>.</p> <p>Contoh: <i>actor</i> pelanggan harus memasukan barang ke keranjang sebelum melakukan check out pembayaran.</p>

Sumber: UML @ Classroom, chapter 3, p.24.

- 2) *Activity Diagram* yaitu diagram yang menggambarkan aliran aktivitas secara rinci dengan skenario utama dan alternatif. Berikut adalah komponen utama yang membentuk sebuah diagram activity dapat dilihat di Tabel 2.3.

Tabel 2. 3 Komponen *Activity Diagram*

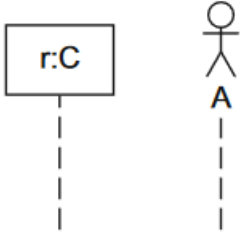

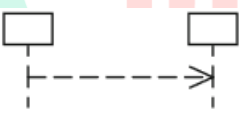

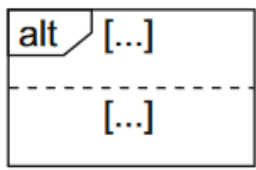
No	Simbol	Keterangan
1	<p><i>Action</i></p>	<p>Persegi dengan sudut melengkung menggambarkan task individual dalam sebuah aktivitas.</p> <p>Contoh: membaca, menulis, dll.</p>
2	<p><i>Activity</i></p>	<p>Persegi dengan sudut melengkung menggambarkan pengelompokkan action dalam satu aktivitas.</p> <p>Contoh: aktivitas menambah data buku berisi aksi mengisi form judul, pengarang, penerbit dll.</p>

3	<i>Initial</i> 	Lingkaran hitam berisi menggambarkan proses awal dari sebuah aktivitas.
4	<i>Activity Final</i> 	Lingkaran hitam dengan garis tepi menggambarkan proses akhir dari sebuah aktivitas.
6	<i>Decision</i> 	Belah ketupat yang bercabang memungkinkan action untuk memiliki beberapa kondisi alternatif.  Contoh: pelanggan reservasi restaurant, jika tempat tersedia maka pelayan memilih tempat. Jika tempat tidak tersedia maka pelanggan akan ditempatkan di <i>waiting list</i> .
7	<i>Parrarelization</i> 	Digambarkan dengan input arah panah dengan garis yang memiliki output bercabang dengan panah lainnya, sehingga memungkinkan aksi bisa dilakukan secara bersamaan.
8	<i>Synchronization</i> 	Digambarkan dengan input banyak arah panah dan memiliki output satu arah panah, sehingga memungkinkan menggabungkan banyak aksi untuk memiliki langkah lanjutan yang sama.
9	<i>Connector</i> 	Arah panah yang menggabungkan satu aksi dengan aksi lainnya.
10	<i>Partition</i> 	Partition digunakan untuk mengelompokkan aksi-aksi pada entitas yang relevan.  Contoh: pengurus taman baca dapat menambah data buku berisi aksi mengisi form judul, pengarang, penerbit dll.

Sumber: UML @ Classroom, chapter 7, p.143.

- 3) *Sequence Diagram* yaitu diagram yang menggambarkan perilaku dari waktu ke waktu dalam satu jalur atau skenario. Berikut adalah komponen utama yang membentuk sebuah diagram *sequence* dapat dilihat di Tabel 2.4.

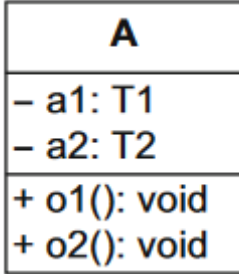
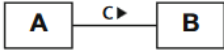
Tabel 2. 4 Komponen *Sequence Diagram*

No	Simbol	Keterangan
1	<p><i>Lifeline</i></p> 	<p>Persegi atau stick figur dengan garis titik-titik vertikal untuk merepresentasikan role atau objek pada <i>sequence diagram</i>.</p> <p>Contoh: Student, Professor, Database dll.</p>
2	<p><i>Synchronous Message</i></p> 	<p>Arah panah dengan segitiga penuh menggambarkan pesan yang dikirim secara <i>synchronous</i>. Pengguna harus menunggu response message sebelum membuat pesan selanjutnya.</p>
3	<p><i>Response Message</i></p> 	<p>Arah panah dengan siku menggambarkan pesan balasan dari perintah <i>synchronous message</i>.</p>
4	<p><i>Asynchronous Message</i></p> 	<p>Arah panah dengan siku menggambarkan pesan yang dikirim secara <i>asynchronous</i> sehingga tidak perlu menunggu <i>response message</i>.</p>
5	<p><i>Alternative Fragment</i></p> 	<p>Saat <i>sequence diagram</i> memiliki kondisi alternatif maka <i>fragment</i> ini digunakan di dalam diagram tersebut.</p> <p>Contoh: Pengguna aplikasi transportasi dapat memilih mode transportasi kendaraan roda dua atau roda empat.</p>

Sumber: UML @ Classroom, chapter 6, p.108.

- 4) *Class Diagram* yaitu diagram yang menggambarkan interaksi antar kelas dengan atribut dan operasi tertentu. Berikut adalah komponen utama yang membentuk sebuah diagram *class* dapat dilihat di Tabel 2.4.

Tabel 2. 5 Komponen *Class Diagram*

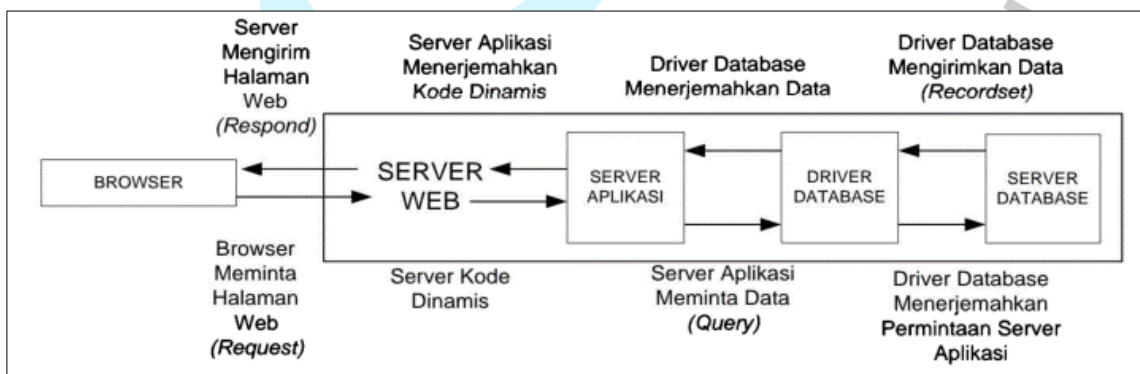
No	Simbol	Keterangan
1	<p><i>Class</i></p> 	<p>Kelas merupakan kategorisasi dari sebuah objek misalkan orang: guru, siswa, dll. serta bangunan: sekolah, gudang, dll.</p> <p>Kelas juga memiliki deskripsi struktur dari suatu objek seperti <i>attribute</i> dan <i>operations</i>.</p> <p>Contoh: Kelas siswa pada sistem akademik memiliki <i>attribute</i> dan tipe datanya seperti nama:string, alamat:string, umur:int, tanggal_lahir:date, dll serta <i>operations</i> siswa bisa mengumpulkan tugas dan melihat nilai.</p>
2	<p><i>Association</i></p> <p><i>Reading direction</i></p> 	<p>Asosiasi merupakan garis yang menghubungkan sebuah kelas dengan kelas lainnya. Garis tersebut juga memiliki <i>multiplicity</i> seperti <i>one to one</i> (1:1), <i>one to many</i>(1:N), dll.</p> <p>Contoh: Hubungan kelas guru dengan kelas siswa yaitu <i>one to many</i> (1:N), artinya sebuah guru memiliki banyak siswa dalam kegiatan belajar mengajar.</p>

Sumber: UML @ Classroom, chapter 4, p.84.

#### 2.1.4 Hypertext Preprocessor (PHP)

Supono (2018) mengungkapkan bahwa PHP merupakan bahasa pemrograman berbasis *server side*, sehingga memerlukan *web server* untuk menerjemahkan baris kode. Komponen yang terlibat dalam proses PHP yaitu: *Browser* agar pengguna dapat mengakses internet dan membuka *website*, *Web server* sebagai perangkat lunak yang menyimpan dan mengirimkan halaman website ke browser, *Server Aplikasi* sebagai interpreter PHP dan memproses kode PHP. *Driver database* sebagai perangkat lunak yang menghubungkan *server aplikasi* dengan *database* dan yang terakhir *Database* sebagai sistem penyimpanan data yang akan ditampilkan di *browser*.

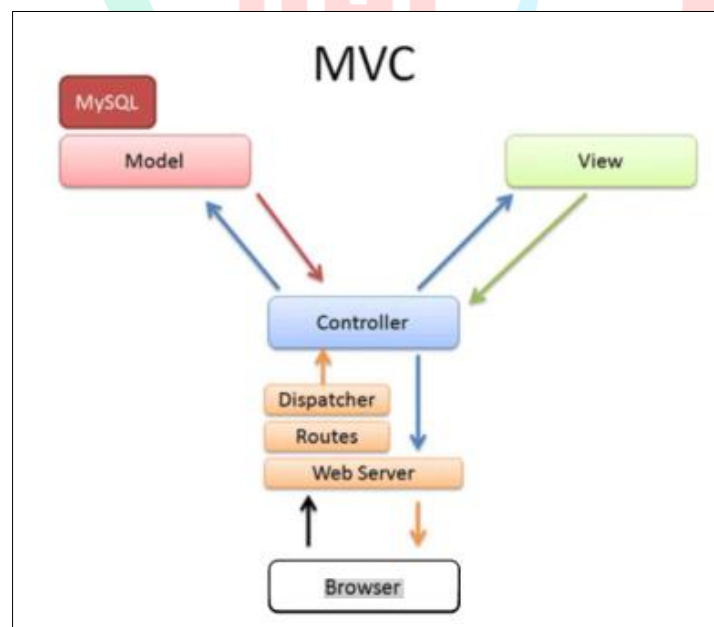
Adapun alur kerja dari PHP yaitu *client* menggunakan *browser* seperti Google Chrome, Mozilla Firefox, dll. untuk mengirimkan *file* PHP, lalu server web menerima *request* tersebut. Server web meneruskan request ke server aplikasi untuk memproses kode PHP pada halaman *website*. Server aplikasi meminta *query* data dari driver database agar diterjemahkan ke *server database*. Server aplikasi menggabungkan data dari *server database* dengan kode PHP Dinamis agar dapat ditampilkan ke browser. Berikut Gambar 2.5 yang menggambarkan alur dari proses PHP.



Gambar 2. 5 Prinsip Kerja PHP

Sumber: Pemrograman Web dengan PHP dan Framework CodeIgniter, 2018.

PHP juga memiliki *framework* untuk membantu pengembang program karena berisi sekumpulan fungsi dan class yang siap digunakan. *Framework* pada PHP seperti *CodeIgniter* merupakan *open source framework* berbasis MVC untuk memisahkan pengembangan *website* agar dinamis. *Framework* ini menyediakan *library* siap pakai seperti validasi, *pagination*, *session*, dll. *Framework* ini memiliki arsitektur pengembangan perangkat lunak MVC atau *Model*, *View* dan *Controller* yang memisahkan pengembangan perangkat lunak ke dalam tiga komponen utama. *Model* adalah komponen untuk mengelola *query* manipulasi data (*insert*, *update*, *delete* dll.). *View* adalah komponen untuk menampilkan *User Interface* (UI) atau tampilan pengguna program *Controller* adalah komponen untuk mengelola fungsi yang menjembatani *Model* dengan *View*. Berikut merupakan ilustrasi dari mekanisme MVC pada Gambar 2.6.



Gambar 2. 6 Mekanisme MVC

Sumber: Pemrograman Web dengan PHP dan Framework CodeIgniter, 2018.

### 2.1.5 Basis Data dan *Structured Query Language* (SQL)

Menurut Jamaludin et. al. (2022) basis data berasal dari kata “basis” yang berarti mengacu pada tempat tertentu serta dari kata “data” yaitu representasi dari objek di dunia nyata seperti manusia, barang, hewan dan lain-lain. Data sendiri diungkapkan dalam bentuk angka, huruf, simbol, bunyi, gambar atau kombinasi dari bentuk-bentuk tersebut. Secara keseluruhan basis data merupakan sekumpulan informasi yang tersimpan secara sistematis berada dalam komputer. Basis data dikelompokkan ke berbagai *Model* beserta fungsinya antara lain:

#### 1) Operational Database

Basis data yang memiliki fungsi untuk mengelola data dinamis secara *asynchronous*, sehingga pengguna bisa melihat, memanggil, mengubah dan menghapus data secara *real time*. Contoh: *Javascript Object Notation* (JSON) dan *Extensible Markup Language* (XML).

#### 2) Database Warehouse

Basis data ini merupakan *repository* sentral yang terintegrasi yaitu berasal dari satu atau lebih sumber yang berbeda. Basis data ini juga menyimpan riwayat data yang berfungsi untuk analisis dan pelaporan. Contoh: *Microsoft SQL Server*.

#### 3) Distributed Database

Basis data ini menyimpan data yang terdistribusi di berbagai perangkat komputer yang berbeda. Komunikasi antar sistem ini terjadi melalui jaringan yang terhubung. Contoh: *Microsoft Access*.

#### 4) Relational Database

Basis data ini berfungsi untuk mengorganisir data berdasarkan *Model* data relasi. Perangkat lunak yang mengelola basis data ini dengan mengatur hubungan antar data. Contoh: MySQL, PostgreSQL, MariaDB, MongoDB, Oracle Database, dll.



Relational Database atau *Model* data relasional digunakan dalam pengembangan sistem pada laporan ini. Bahasa yang digunakan untuk berinteraksi pada *Model* data relasional ini adalah SQL atau *Structured Query Language* sehingga memungkinkan pengguna untuk melakukan operasi seperti menambah, memperbarui, menghapus data, serta mengambil data dengan menggunakan Perangkat lunak yang digunakan untuk menerapkan bahasa tersebut yaitu MySQL.

MySQL memiliki intruksi yang digunakan untuk mengelola database dan tabel bernama *query*. Dalam MySQL terdapat tiga jenis query utama yaitu:

1) Query DDL (Data Definition Language)

Intruksi tersebut digunakan untuk mendefinisikan struktur database, seperti membuat database, tabel, mengubah struktur tabel, dan menghapus objek database.

Contoh: CREATE DATABASE, CREATE TABLE, ALTER TABLE, DROP DATABASE.

2) Query DML (Data Manipulation Language)

Intruksi ini digunakan untuk memanipulasi data dalam tabel, seperti menambahkan, mengubah, menghapus, dan mengambil data.

Contoh: INSERT INTO, UPDATE, DELETE, SELECT FROM.

3) Query DCL (Data Control Language)

Intruksi ini digunakan untuk mengatur hak akses pengguna terhadap database.

Contoh: Memberikan izin (GRANT) atau mencabut izin (REVOKE) pada user MySQL.

### 2.1.6 Barcode dan *International Standard Book Number (ISBN)*

Wahyono (2010) mengungkapkan bahwa *Barcode* atau kode batang adalah kode berbentuk garis berwarna hitam putih dengan ketebalan yang bervariasi, hal ini dirancang demikian agar bisa diterjemahkan oleh mesin pembaca. Secara umum *barcode* biasanya dicetak dan ditempel pada produk tertentu untuk merepresentasikan data produk tersebut. Salah satu pemanfaatan *barcode* pada kehidupan sehari-hari adalah pada cover belakang buku. *Barcode* pada buku berisi kode ISBN yang merupakan kode untuk mengidentifikasi buku yang berisi informasi buku terkait dengan judul, pengarang, penerbit, tahun terbit, deskripsi, dll.

Nomor ISBN terdiri dari 13 digit dan terdiri atas lima bagian dan dipisahkan dengan tanda strip (-) sebagai berikut, *Prefix identifier*: menunjukkan bahwa produk tersebut adalah buku. *Group identifier*: menunjukkan wilayah geografis penerbit. *Publisher prefix*: menunjukkan penerbit buku. *Title identifier*: menunjukkan judul buku. *Check digit*: digunakan untuk memverifikasi keabsahan kode ISBN. Contoh: ISBN 978-602-8519-93-9. (978) *Prefix identifier*, (602) *Group identifier*, (8519) *Publisher prefix*, (93) *Title identifier* dan (9) *Check digit*. (Perpusnas, 2019) Berikut adalah Gambar 2.4 dari struktur ISBN.



Gambar 2. 7 Struktur ISBN  
Sumber: Perpusnas

### 2.1.7 Application Programming Interface (API)

API adalah seperangkat metode yang digunakan membantu suatu pengembang perangkat lunak untuk mengakses sebuah data dari sumber luar pada lingkungan sistem yang tertutup. *Google Books* API merupakan salah satu pemanfaatan dari API yang memungkinkan pengembang untuk mendapatkan akses informasi buku seperti metadata buku, cover buku, format *e-books*, dll. *Google Books* API memungkinkan untuk ekstraksi data melalui *request Hypertext Transfer Protocol* (HTTP) dan mengembalikannya ke dalam *format JavaScript Object Notation* (JSON). (Michel, 2013).

Penggunaan *Google Books* API ini dengan menggunakan metode pencarian volume dengan mengirimkan *request* HTTP GET ke URL: “<https://www.googleapis.com/books/v1/volumes?q=kata+kunci>” *Request* ini memiliki satu parameter yang diperlukan yaitu “q” yang berisi *string* teks. Beberapa kata kunci khusus yang dapat dipergunakan adalah sebagai berikut:

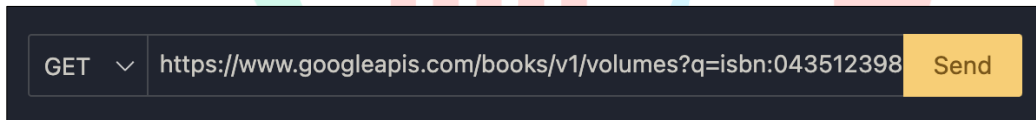
- **intitle:** Mengembalikan hasil dengan berdasarkan kata kunci di judul.
- **inauthor:** Mengembalikan hasil dengan berdasarkan kata kunci penulis.
- **inpublisher:** Mengembalikan hasil dengan berdasarkan kata kunci penerbit.
- **subject:** Mengembalikan hasil dengan kata kunci daftar kategori volume.
- **isbn:** Menampilkan hasil dengan kata kunci nomor ISBN (*International Standard Book Number*). (Google, 2024)

Dalam penerapannya, penggunaan kata kunci khusus dibatasi pada kata kunci “isbn:”, hal ini disesuaikan dengan tujuan semula yaitu pemanfaatan *barcode* ISBN yang berisi data nomor ISBN baik ISBN dengan format 10 angka maupun ISBN dengan format 13 angka.

Berikut ini adalah contoh penggunaan kode dengan AJAX (*Asynchronous JavaScript And XML*) pada *Javascript* dengan nomor ISBN 10 angka:

```
var api_link =  
"https://www.googleapis.com/books/v1/volumes?q=isbn:";  
var nomor_isbn = "043512398x";  
  
$.ajax({  
  url: api_link + nomor_isbn,  
  type: 'GET',  
  cache: false,  
  dataType: 'json',  
  success: function(res) {  
    /** decode object res **/  
  }  
});
```

Berikut ini adalah contoh hasil request pencarian dengan menggunakan kata kunci “isbn: 043512398x” berupa JSON dapat dilihat di Gambar 2.4 dan 2.5:



Gambar 2. 8 Request GET  
Sumber: Peneliti



```
1  {
2    "kind": "books#volumes",
3    "totalItems": 1,
4    "items": [
5      {
6        "kind": "books#volume",
7        "id": "ZoRAPgAACAAJ",
8        "etag": "LmOAdwvWeKE",
9        "selfLink": "https://www.googleapis.com/books/v1/volumes
10         /ZoRAPgAACAAJ",
11        "volumeInfo": {
12          "title": "Matilda",
13          "authors": [
14            "Roald Dahl",
15            "Quentin Blake"
16          ],
17          "publisher": "Heinemann Educational Publishers",
18          "publishedDate": "1992",
19          "description": "Chinese edition of Matilda, the Roald Dahl
20            classic. With unusual power, Matilda was able to rid of
21            her school with the vicious headmistress and help her
22            teacher Ms. Honey. Matilda is an unusual book. The last
23            book by Dahl, it is almost like his last words to his
24            readers.",
25          "industryIdentifiers": [
26            {
27              "type": "ISBN_10",
28              "identifier": "043512398X"
29            },
30            {
31              "type": "ISBN_13",
32              "identifier": "9780435123987"
33            }
34          ]
35        }
36      ]
37    }
```

Gambar 2. 9 Hasil JSON Request GET  
Sumber: Peneliti

### 2.1.8 Taman Bacaan Masyarakat (TBM)

Kementerian Pendidikan dan Kebudayaan (2021) mengungkapkan bahwa TBM adalah tempat dan layanan yang bertujuan untuk meningkatkan daya baca masyarakat dengan memanfaatkan sumber daya baik dari administrasi maupun sarana fisik. Upaya yang dilakukan TBM untuk mengembangkan minat baca adalah dengan menyediakan bahan bacaan seperti buku populer, fiksi, keterampilan, agama, dan cerita serta sarana pendukung seperti rak buku, meja, kursi, dll. Hal demikian juga didukung jika TBM memanfaatkan Information Technology (IT) seperti pengadaan jaringan *internet*, *e-book* atau buku digital, komputer/laptop dan sejenisnya. Indikator keberhasilan dari pengadaan TBM yaitu bertambahnya sarana dan prasarana, peningkatan jumlah/jenis layanan serta bertambahnya aktivitas literasi di area TBM.

Maka dari itu penerima manfaat dari penyelenggaraannya TBM yaitu masyarakat baik individu maupun lembaga idealnya menerima alokasi Bantuan Penguatan TBM sebesar dua puluh juta dengan rincian sebagai berikut:

- 1) Administrasi/Manajemen (maksimal 20%), mencakup pengadaan alat tulis kantor, inventaris dan label bahan bacaan, pelaporan, pajak, dan pengelolaan TBM.
- 2) Penambahan sarana pendukung (maksimal 20%), seperti lemari buku, meja, dan kursi.
- 3) Pengadaan buku/bahan bacaan (minimal 40%), yang disesuaikan dengan karakteristik lokal, dengan setiap judul buku maksimal 10 eksemplar, mencakup buku populer, fiksi, keterampilan, agama, dan cerita.
- 4) Peningkatan layanan bahan bacaan dan kegiatan literasi masyarakat (maksimal 20%), termasuk transportasi lokal, konsumsi rapat, dan retribusi.

### 2.1.9 Layanan Perpustakaan

Rahmah (2018) mengungkapkan bahwa layanan perpustakaan adalah penyediaan informasi dan fasilitas yang dapat dimanfaatkan baik dari pihak pengelola perpustakaan maupun pengunjung perpustakaan. Perpustakaan umumnya memiliki layanan sebagai berikut:

1) Layanan Pencarian

Layanan dimana pengelola perpustakaan membantu mencari informasi terkait buku yang dibutuhkan oleh pengunjung perpustakaan. Sistem pencarian buku dimanfaatkan dalam proses layanan ini untuk mencari buku berdasarkan kata kunci tertentu.

2) Layanan Sirkulasi

Sirkulasi adalah proses perputaran buku dari tangan ke tangan sehingga memungkinkan pengunjung perpustakaan untuk melakukan peminjaman buku untuk dibawa pulang serta buku tersebut akan dikembalikan dari jadwal yang telah ditentukan.

3) Layanan Ruang Baca

Layanan ini menyediakan tempat kondusif untuk membaca buku di tempat secara langsung dan ditujukan oleh pengunjung yang tidak membawa pulang buku yang dipinjam.

4) Layanan Penyebaran Informasi Baru

Layanan ini juga disebut dengan layanan kesiagaan informasi, dalam kata lain pengelola perpustakaan melakukan pemantauan terhadap terbitan terbaru bahan baca.

## 2.2 Tinjauan Studi

Berikut adalah ulasan jurnal-jurnal yang menjadi landasan teori:

- 1) Jurnal pada tahun 2023 yang berjudul “**Implementasi Barcode Scanner Berbasis Android untuk Otomatisasi Stock opname pada Library Management System (LMS) di Perpustakaan Muhammad Yusuf Ahmadi Tanjungpinang**” oleh Danandjaya Saputra membahas permasalahan yang dihadapi ketika melakukan proses *stock opname* tergolong lama yaitu memakan waktu satu sampai dua bulan karena proses pencocokan satu per satu kode buku dari koleksi buku di rak dengan database di LMS menggunakan *Microsoft Excel*. Hal ini bertujuan untuk memudahkan petugas *stock opname* melakukan pengecekan stok buku dengan efisien. Melalui metode pengumpulan data, implementasi aplikasi *barcode scanner* berbasis *android* bisa menjadi solusi untuk mewujudkan tujuan tersebut. Adapun hasil valid dari pengujian fungsional dengan *black box testing* terhadap komponen-komponen sistem yang dikembangkan seperti *login*, kamera untuk *scan barcode* buku dan memperbarui status buku saat *stock opname*.
- 2) Jurnal pada tahun 2023 yang berjudul “**Implementasi Web Service Restful API untuk Layanan Perpustakaan SMAS Daya Utama Bekasi**” oleh Briana Muham membahas permasalahan pencatatan peminjaman dan pengembalian buku serta lambatnya proses pencarian buku. Hal ini bertujuan untuk membantu petugas dalam proses rekapitulasi status buku seperti peminjaman dan pengembalian serta memperluas akses perpustakaan dengan *platform website* dan *android*. Melalui metode pengumpulan data, implementasi *web service restful API* bisa menjadi solusi untuk menjawab permasalahan dari layanan perpustakaan tersebut. Hasil pengujian *User Acceptance Testing (UAT)* menunjukkan tingkat kepuasan pengguna yang tergolong tinggi dengan nilai 78 dari 100.



- 3) Jurnal pada tahun 2022 yang berjudul “**Perancangan Aplikasi Android Katalog Buku dengan Fitur Barcode Scanner menggunakan Google ML Kit**” oleh Ryan Putrananda Kristianto membahas permasalahan berketergantungannya mesin pembaca untuk memindai kode batang (*barcode*). Maka dari itu perancangan sistem pencarian katalog buku berbasis *android* dengan fitur pemindaian *barcode* melalui pemanfaatan *Google Machine Learning Kit* bertujuan membaca gambar *barcode* yang diterjemahkan ke dalam kode *International Standard Book Number* (ISBN). Metode yang digunakan untuk mewujudkan tujuan tersebut ialah dengan memanfaatkan *Google Machine Learning Kit*. Teknologi tersebut menyediakan berbagai API umumnya untuk keperluan pemindaian *barcode*, pemrosesan gambar dan pemrosesan teks. Hasil dari analisis kebutuhan fungsional yaitu implementasi fitur pencarian *Barcode Scanner*, menu katalog buku, menu halaman *developer* dan menu utama.
- 4) Jurnal pada tahun 2022 yang berjudul “**Integrasi Barcode-QRCode pada Perpustakaan Universitas Bumigora Mataram dengan Konsep Sistem Terdistribusi Berbasis Mobile**” oleh Moch Syahrir membahas permasalahan sistem berjalan pada perpustakaan yang dibahas karena minimnya interaksi pengguna yang pada konteks ini yaitu mahasiswa terhadap sistem menyebabkan tidak efisiennya proses sirkulasi buku. Hal ini bertujuan untuk memberi solusi untuk memudahkan beban kerja operator pada proses sirkulasi buku yaitu peminjaman dan pengembalian karena pada sistem berjalan hanya operator saja yang menjadi pengguna. Maka dari itu pemanfaatan metode pemindaian *Barcode* dan *QR Code* dalam proses peminjaman dan pengembalian buku serta pengelolaan data buku dapat membantu meningkatkan proses operasional dari sistem berjalan sebelumnya. Hasil pengujian pada sistem tersebut menggunakan questioner dari 20 responden memiliki nilai 97,70 dari 100 sehingga memasuki kategori aplikasi yang sangat baik.

- 5) Jurnal pada tahun 2022 yang berjudul “**Penerapan Teknologi *Quick Response Code* dan *Application Programming Interface* pada Perancangan Aplikasi Perpustakaan (Studi Kasus : SMP Negeri 25 Surakarta)**“ oleh Annisa Nugraheni membahas permasalahan tidak adanya sistem proses pengelolaan bahan pustaka sehingga menimbulkan kesulitan mencari informasi pada banyaknya volume bahan pustaka. Hal ini bertujuan untuk memberikan solusi melalui pengembangan sistem yang yang dapat mengelola dan mencari informasi mengenai bahan pustaka. Metode yang digunakan ialah memanfaatkan fitur pemindaian *QR Code* berfungsi untuk memindai status buku apakah buku tersedia atau terpinjam, sedangkan fitur API digunakan untuk memudahkan menampilkan data pada *platform android mobile*. Hasil uji *System Usability Scale (SUS)* oleh 37 responden memiliki nilai rata-rata sebanyak 82,823 dari 100 sehingga tergolong *usable* atau berarti aplikasi dapat digunakan dengan baik.

Berdasarkan sekumpulan jurnal tinjauan studi sebelumnya terdapat poin-poin yang bisa disimpulkan dengan topik yang saat ini diteliti antara lain:

1) Perbedaan dari penelitian sebelumnya

Objek jurnal terdahulu memiliki konteks yang berbeda yaitu fokus pada Perpustakaan di Lembaga Pendidikan seperti sekolah, sedangkan penelitian ini fokus pada Taman Bacaan Masyarakat (TBM) di perumahan. Dari segi pengelolaan buku penelitian terdahulu fokus pada stok buku yang terdaftar di *database*, sedangkan pengelolaan buku di TBM didapatkan dari donasi masyarakat sekitar dan komunitas membaca.

2) Pengembangan dari penelitian sebelumnya

Penelitian sebelumnya hanya menggunakan API (Jurnal 2 dan 5) untuk meningkatkan aksesibilitas data serta QR atau *barcode scanner* untuk proses *stock opname* (Jurnal 1), pencarian katalog buku (Jurnal 3) dan pengelolaan status buku (Jurnal 4). Penelitian saat ini mengkombinasikan *library barcode scanner* dengan *Google Books API* memberikan kelebihan dalam otomatisasi proses *input data*. Informasi dari *Google Books* dapat langsung diambil tanpa memerlukan *input manual* dan mengurangi kesalahan pengetikan.