

## BAB II

### TINJAUAN PUSTAKA DAN DASAR TEORI

#### 2.1 Tinjauan Pustaka

Pada penelitian ini digunakan tinjauan pustaka sebagai referensi dalam pembangunan rancangan hingga implementasi aplikasi yang akan dibuat. Daftar pustaka yang digunakan sebagai acuan serta referensi dari pembangunan penelitian tercantum pada tabel 2.1

**Tabel 2.1** Tabel Pustaka

<b>Nama Penulis</b>	<b>Topik</b>	<b>Metodologi</b>	<b>Hasil</b>
Iqbal Ramadhani Mukhlis, Deny Hermansyah, Vanessa Meilisa Lantang (2023)	Merancang basis data menggunakan <i>entity relationship diagram</i> dan <i>physical data model</i>	<i>Entity relationship database</i> dan <i>physical data model</i>	Penelitian ini mengungkap perancangan dengan <i>Entity Relationship Database</i> yang memberikan kemudahan dalam memahami relasi antar tabel, sedangkan perancangan dengan model <i>physical data model</i> memberikan kemudahan ketika ingin membangun basis data.
Sayyid Yakan Khomsi Pane, Nur Ghaniaviyanto Ramadhan, Faisal Dharma Adhinata (2022)	Perancangan basis data menggunakan normalisasi tabel	<i>Entity relationship database</i> dan normalisasi tabel	Penelitian ini mengungkap basis data berperan penting menghimpun dan mengelola data dalam rangka pelaksanaan rencana strategi sebuah instansi.
Mohammad Harry Khomas Saputra, Luthfi Muhammad	Penerapan arsitektur <i>microservice</i> pada sistem tata kelola	Arsitektur <i>microservice</i>	Penelitian ini mengungkapkan bahwa pengembangan aplikasi berbasis web menggunakan

Nabil (2021)	matakuliah		aritektur <i>microservice</i> lebih tidak mengganggu sistem lain saat pengembangan berlangsung
Alessandro Sinambela, Ernawati, Funny Farady Coastera (2021)	Implementasi arsitektur <i>microservices</i> pada rancang bangun aplikasi	Arsitektur <i>microservice</i>	Penelitian ini menghasilkan aplikasi marketplace menggunakan arsitektur <i>microservices</i> dan telah dilakukan uji kelayakan menggunakan kuisioner kepada pedagang Pasar Tradisional Bengkulu (PTM)
Eli Nurhayati dan Agussalim (2023)	Rancang bangun back-end API	<i>Waterfall</i> dan REST API	Penelitian ini menghasilkan back-end aplikasi ayamHub menggunakan <i>framework</i> ExpressJS
Raihan Rafid Adityo (2021)	Penggunaan AWS MinIO dalam Qiscus Meet SDK sebagai storage server lokal untuk melakukan penyimpanan hasil file recording	AWS MinIo sebagai tempat penyimpanan file hasil <i>recording</i>	Penelitian ini mengungkapkan dengan menggunakan AWS MinIo dapat mengurangi biaya sewa

Penelitian yang dilakukan oleh Iqbal Ramadhani Mukhlis, Deny Hermansyah dan Vanessa Meilisa Lantang (2023) adalah membuat “*Rancangan Basis Data Transaksi Pada PT. Bank Perkreditan Rakyat ABC Menggunakan MySQL Dengan Model Entity RelationshipDiagram (ERD) dan Physical Data Model (PDM)*”. Tujuan penelitian ini adalah untuk membuat perancangan basis data yang sesuai dengan alur transaksi online atau melalui E-Banking. Tahapan yang dilalui dalam merancang basis data dalam penelitian ini adalah membuat

*Entity Relationship Diagram (ERD)*, membuat *Physical Data Model (PDM)*, dan menormalisasi data sampai dengan 3NF. Kesimpulan yang didapat dari penelitian ini adalah perancangan dengan *Entity Relationship Database* yang memberikan kemudahan dalam memahami relasi antar tabel, sedangkan perancangan dengan model *physical data model* memberikan kemudahan ketika ingin membangun basis data.

Penelitian yang dibuat oleh Sayyid Yakan Khomsi Pane, Nur Ghaniaviyanto Ramadhan, Faisal Dharma Adhinata (2022) berjudul "*Perancangan Basis Data Menggunakan Normalisasi Tabel Pada Perusahaan Dagang Barokah Abadi*". Tahapan yang dilalui dalam merancang basis data dalam penelitian ini adalah membuat *Entity Relationship Diagram (ERD)*, dari hasil perancangan *ERD* tersebut dibuat *Physical Data Model (PDM)*, kemudian menormalisasi data sampai dengan 3NF dan tahapan yang terakhir adalah implementasi rancangan ke dalam basis data. Penelitian ini mengungkap basis data berperan penting dalam menghimpun dan mengelola data untuk pelaksanaan rencana strategi sebuah instansi.

Penelitian yang dibuat oleh Mohammad Harry Khomas Saputra, Luthfi Muhammad Nabil (2021) berjudul "*Penerapan Arsitektur Microservice Pada Sistem Tata Kelola Matakuliah Proyek Politeknik Pos Indonesia*". Sistem Informasi Tata Kelola Proyek Politeknik Pos Indonesia awalnya dibangun menggunakan arsitektur monolithic, namun karena diperlukan beberapa tambahan fitur maka arsitektur pada sistem informasi tersebut berubah menjadi arsitektur microservice. Arsitektur microservice dipilih karena dalam pengembangannya

sistem informasi ini dapat dipecah menjadi beberapa bagian dan membuat pengembang lebih mudah melakukan pengembangan sistem informasi seiringan dengan terpisahnya komponen - komponen dari aplikasi tanpa mengganggu komponen yang lain. Kesimpulan yang didapat dari penelitian ini adalah pengembangan aplikasi berbasis web menggunakan arsitektur *microservice* lebih tidak mengganggu sistem lain saat pengembangan berlangsung.

Penelitian yang dilakukan oleh Alessandro Sinambela, Ernawati, Funny Farady Coastera (2021) memiliki judul "*Implementasi Arsitektur Microservices Pada Rancang Bangun Aplikasi Marketplace Berbasis Web*". Hasil dari penelitian ini adalah aplikasi marketplace menggunakan arsitektur *microservices* yang telah dilakukan uji kelayakan menggunakan kuisioner kepada pedagang Pasar Tradisional Bengkulu (PTM).

Penelitian yang dilakukan oleh Eli Nurhayati dan Agussalim (2023) memiliki judul "*Rancang Bangun Back-end API pada Aplikasi Mobile AyamHub Menggunakan Framework Node JS Express*". Penelitian ini menggunakan metode *waterfall* dalam pengembangannya dan menggunakan metode *balckbox* dalam pengujiannya. Hasil dari penelitian ini adalah back-end aplikasi ayamHub menggunakan *framework* ExpressJS.

Penelitian yang dilakukan oleh Raihan Rafid Adityo (2021) memiliki judul "*Pengembangan Aplikasi Qiscus Meet Menggunakan Jitsi Open Source Framework*". Penelitian ini menggunakan AWS Minio sebagai tempat penyimpanan hasil *recording* yang dibuat oleh server Jibri. AWS MinIo dipilih sebagai *storage server* karena biaya sewanya lebih murah dibanding jenis server

lainnya.

## **2.2 DasarTeori**

### **2.2.1 *UMKM Financing Empowerment***

U-Fine (*UMKM Financing Empowerment*) adalah program pemberdayaan UMKM yang dilaksanakan oleh Direktorat Jenderal Perbendaharaan Kementerian Keuangan yang berfokus pada pelaku UMKM yang akan atau sudah menerima program pembiayaan pemerintah yang diharapkan dapat mendorong UMKM untuk naik kelas. Program ini dilaksanakan dalam bentuk pembinaan atau pendampingan terhadap UMKM terpilih untuk memperoleh akses pinjaman KUR maupun UMi, pelatihan pemasaran, dan kegiatan lainnya.

### **2.2.2 *Relational Database Management System* Atau Sistem Basis Data Relasional**

Data adalah semua fakta yang ada di dunia yang dapat direkam dan disimpan oleh komputer. Sedangkan basis data dapat diartikan sebagai kumpulan terorganisir data-data yang saling berhubungan sehingga dapat dengan mudah untuk disimpan, diolah, dimanipulasi dan dipanggil oleh pengguna, Adi Nugroho (2011).

Konsep pada basis data relasional menggambarkan relasi diantara data-data yang ada di dalam basis data. Sistem basis data tersebut digambarkan dalam bentuk tabel dua dimensi yang tersusun atas baris yang menggambarkan satu kesatuan data dan kolom yang menyatakan bagian-bagian yang menyusun

sebuah baris. Kumpulan data yang terdapat dalam basis data relasional dikelola dalam sebuah *software* yang disebut dengan *Relational Database Management System* (RDBMS), Adi Nugroho (2011).

Untuk mengelola kumpulan data pada basis data relasional diperlukan perintah SQL (*Structured Query Language*), SQL dapat digunakan untuk mendefinisikan struktur data, memodifikasi data pada basis data, menspesifikasi batasan keamanan, dan penentuan hak akses, Setiyowati dan Sri Iswanti (2021).

Perintah SQL dibagi menjadi beberapa jenis sesuai dengan fungsinya, yaitu :

1. *Data Definition Language* (DDL), merupakan perintah yang digunakan untuk membuat atau mengubah struktur objek basis data, diantaranya :
  - a. CREATE digunakan untuk membuat struktur objek basis data
  - b. ALTER digunakan untuk mengubah struktur objek basis data
  - c. DROP digunakan untuk menghapus objek basis data
  - d. TRUNCATE digunakan untuk mengosongkan atau menghapus data yang tersimpan pada basis data
  - e. RENAME digunakan untuk mengganti nama tabel pada basis data
2. *Data Control Language* (DCL), merupakan perintah untuk mengatur hak akses pengguna dalam basis data, diantaranya :
  - a. GRANT digunakan untuk memberi izin tertentu kepada pengguna
  - b. REVOKE digunakan untuk menarik izin tertentu dari pengguna
3. *Data Control Language* (DML), merupakan perintah untuk memanipulasi data, diantaranya:

- a. INSERT, digunakan untuk menambah baris ke tabel
  - b. DELETE, digunakan untuk menghapus baris dari tabel
  - c. UPDATE, digunakan untuk mengubah data pada tabel
  - d. MERGE, digunakan untuk menggabungkan data dari dua tabel
4. Perintah *Query*, merupakan perintah yang digunakan untuk menampilkan data dari tabel, yaitu SELECT
5. *Transaction Control Language* (TCL), merupakan perintah untuk melakukan penyimpanan atau perubahan data secara permanen dan melakukan pembatalan terhadap perubahan data yang disimpan, diantaranya :
- a. COMMIT, digunakan untuk melakukan penyimpanan atau perubahan data secara permanen
  - b. ROLLBACK, digunakan untuk membatalkan perubahan data yang disimpan
  - c. SAVEPOINT, digunakan untuk menentukan tahapan-tahapan dalam melakukan perubahan data sehingga jika ada kesalahan data bisa dikembalikan ke tahapan tertentu.

### 2.2.3 *Entity Relationship Diagram* (ERD)

*Entity Relationship Diagram* (ERD) adalah suatu model informasi untuk menjelaskan suatu hubungan antara data dan basis data yang digambarkan dengan sebuah grafik dan juga notasi dengan model data konseptual, Kurniawan (2023). Ada dua jenis model ERD, yaitu:

1. *Conceptual Data Model* (CDM)

CDM adalah suatu jenis model data yang menggambarkan hubungan antar tabel secara konseptual. CDM memodelkan struktur logis dari keseluruhan aplikasi data, tidak tergantung pada software atau pertimbangan model struktur data. CDM yang valid dapat dikonversi ke PDM, Mukhlis (2023).

## 2. *Physical Data Model (PDM)*

PDM adalah suatu jenis model data yang menggambarkan hubungan antar tabel secara fisikal. PDM merupakan representasi fisik dari basis data yang akan dibuat dengan mempertimbangkan DBMS yang akan digunakan, Mukhlis (2023). PDM dapat dihasilkan (*di-generate*) dari CDM yang valid. PDM dalam penerapannya dapat di samakan dengan skema relasi yang fungsinya adalah memodelkan struktur fisik dari suatu basis data.

### 2.2.4 Rest API

API adalah antarmuka yang digunakan untuk mengakses aplikasi atau layanan dari sebuah program. API memungkinkan pengembang untuk memakai fungsi yang sudah ada dari aplikasi lain sehingga tidak perlu membuat ulang dari awal. Tujuan penggunaan dari API untuk mempercepat proses pengembangan aplikasi dengan cara menyediakan sebuah *function* yang terpisah sehingga para developer tidak perlu lagi membuat fitur yang serupa.

Menurut Fielding (2000), REST adalah sebuah standar arsitektur web yang menggunakan protokol *Hypertext Transfer Protocol* dan juga *Uniform Resource Identifier* dalam berkomunikasi. REST memanipulasi sumber daya hanya dengan

melakukan pertukaran representasi. Enam batasan yang terdapat pada REST API yaitu :

1. *Client-Server*

Batasan *client-server* menjelaskan sebuah antarmuka yang terpisah antara client dan server. Pemisahan fungsionalitas tersebut berakibat pada penyederhanaan komponen server sehingga meningkatkan skalabilitas. Pemisahan juga memungkinkan kedua komponen untuk berkembang secara independen. Komponen server menyediakan sebuah layanan, mendengarkan *request* atas layanan tersebut, dan menolak atau menjalankan request serta mengirim kembali sebuah *response* kepada *client*. Sedangkan komponen *client* menginginkan layanan untuk dijalankan, dan mengirimkan *request* kepada server melalui sebuah koneksi.

2. *Stateless*

Batasan ini menjelaskan bahwa tidak ada status *session* yang diperbolehkan pada komponen server. Seluruh request dari client kepada server harus berisikan informasi yang cukup untuk dipahami dan konteks yang tersimpan di server tidak dapat dimanfaatkan. Oleh karena itu, status *session* sepenuhnya disimpan di client.

3. *Cacheable*

Batasan *cacheable* menjelaskan bahwa *response* dari server dapat bersifat *cacheable*. Oleh karena itu, response harus mendefinisikan secara implisit atau eksplisit dirinya sebagai cacheable maupun tidak.

#### 4. *Uniform Interface*

Batasan ini menekankan kepada komponen *interface* yang seragam. Dengan menerapkan prinsip rekayasa perangkat lunak ke komponen *interface*, sumberdaya dapat memanipulasi *create*, *read*, *update*, *delete* (CRUD) dengan memanfaatkan operasi HTTP yaitu POST, GET, PUT, dan DELETE.

#### 5. *Layered*

System Batasan ini memungkinkan arsitektur memiliki lapisan hierarki yang dapat membatasi komponen, sehingga masing-masing komponen tidak dapat “melihat” lapisan lain diluar tempat interaksi mereka.

#### 6. *Code on Demand* (opsional)

Batasan ini memungkinkan fungsionalitas dari client diperluas, dengan mengunduh dan mengeksekusi kode berupa skrip. Hal tersebut dapat menyederhanakan kerja client dengan mengurangi jumlah fitur yang harus dieksekusi.

### 2.2.5 Node Js Dan Express Js

Menurut Rafly, M. A (2020), Node.js adalah perangkat lunak yang didesain untuk mengembangkan aplikasi berbasis web dan ditulis dalam sintaks bahasa pemrograman JavaScript. Node.js melengkapi peran JavaScript sehingga bisa juga berlaku sebagai bahasa pemrograman yang berjalan di sisi server, seperti halnya PHP, Ruby, Perl, dan sebagainya. Node.js dapat berjalan di sistem operasi Windows, Mac OS X dan Linux tanpa perlu ada perubahan kode program,

ExpressJS merupakan sebuah *framework* yang dibangun dengan basis Javascript dan dikompilasi pada software NodeJS. ExpressJS biasanya digunakan untuk membangun API dan menampung layanan-layanan yang ada. *Framework* ini banyak digunakan untuk kalangan pengembang pemula karena mudahnya dalam membangun layanan-layanan dan mengintegrasikannya pada satu sistem yang dibangun pada ExpressJS.

### 2.2.6 MinIO AWS

Amazon Web Services atau biasa dikenal dengan AWS, merupakan salah satu platform cloud yang sangat komprehensif. Salah satu fitur yang diberikan oleh AWS yaitu *storage server*, biasanya jika ingin menggunakan fitur ini, pilihannya dapat menggunakan AWS S3. Namun jika kebutuhannya adalah sebuah penyimpanan objek yang dapat dijalankan di jaringan pribadi atau lokal, AWS juga menyediakan sebuah *service* yang bernama AWS MinIo.

Teknologi dari Minio.io sendiri dibuat *compatible* dengan layanan Amazon S3. Hal itu membuat Minio memiliki komunitas, support, serta dokumentasi yang cukup banyak.

### 2.2.7 Middleware

Middleware merupakan *function* yang memiliki akses terhadap objek *request* (req), objek *response* (res), dan *function* next yang digunakan untuk melanjutkan permintaan kepada *function* berikutnya. Middleware dapat melakukan beberapa tugas berikut :

1. Menjalankan kode program

2. Membuat perubahan pada objek *request* dan *response*
3. Mengakhiri siklus *request-response*
4. Memanggil *function* berikutnya pada stack

Jika *function* middleware tidak mengakhiri siklus *request-response*, maka *function* tersebut harus memanggil `next()` untuk melanjutkan kontrol ke *function* berikutnya, jika tidak maka *request* akan dibiarkan menggantung.

### 2.2.8 Pengujian Dengan Metode *White Box* dan *Black Box*

Menurut V. P. Katiyar and S. Patel (2019), pengujian White Box adalah metode pengujian aplikasi yang menggunakan penjelasan struktur kontrol sebagai bagian dari component-level design untuk membuat test cases. White Box sendiri mempunyai beberapa teknik di dalam pengujiannya, seperti : *Data Flow Testing*, *Control Flow Testing*, *Basic Path / Path Testing*, dan *Loop Testing*. Teknik pengujian *Basic Path* digunakan untuk mengukur kompleksitas logika dari rancangan prosedural. Teknik basis path testing lebih cocok digunakan dibandingkan dengan teknik lainnya, karena basis path testing akan menghasilkan jumlah test case dengan cakupan test yang lebih menyeluruh dibandingkan teknik lainnya.

Teknik basis path terdiri dari flow graph notation yang merupakan notasi sederhana yang menggambarkan alur kontrol program, cyclomatic complexity yang merupakan perhitungan untuk menentukan jumlah dari jalur pengujian, independent path yang merupakan penentuan jalur pengujian yang dilewati

setidaknya sekali, graph matrix yang merupakan matriks 2-dimensional yang bertujuan untuk membantu menghitung cyclomatic complexity.

Berbeda dengan *White Box*, pengujian *Black Box* berfokus pada fungsionalitas perangkat lunak. Pengujian black box bertujuan untuk menemukan fungsi yang tidak benar, kesalahan antarmuka, kesalahan pada struktur data, kesalahan performansi, kesalahan inisialisasi dan terminasi (F. C. Ningrum, et al., 2019). Teknik yang digunakan dalam pengujian blackbox ini adalah teknik *equivalence partitions* yang merupakan pengujian berdasarkan masukan data pada setiap form yang ada pada sistem.