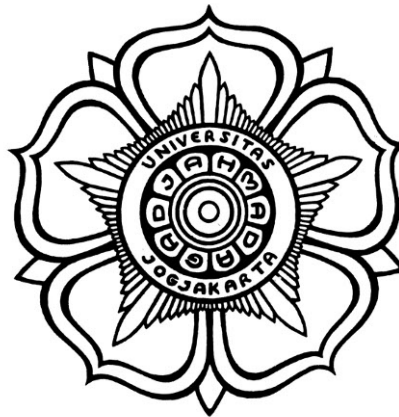


**LAPORAN PROYEK
BASIS DATA**

**DESAIN DAN IMPLEMENTASI BASIS DATA “INFINIREAD” UNTUK
SISTEM PERPUSTAKAAN DIGITAL**

***DESIGN AND IMPLEMENTATION OF THE “INFINIREAD” DATABASE
FOR DIGITAL LIBRARY SYSTEM***



Kukuh Agus Hermawan (24/533395/PA/22573)
Aulia Fathus Tsani (24/534388/PA/22661)
Alifa Batrisyia Nariswari (24/535256/PA/22707)

**PROGRAM STUDI ILMU KOMPUTER
DEPARTEMEN ILMU KOMPUTER DAN ELEKTRONIKA
FAKULTAS MATEMATIKA DAN ILMU PENGETAHUAN ALAM
UNIVERSITAS GADJAH MADA
YOGYAKARTA**

2025

DAFTAR ISI

DAFTAR ISI	ii
I Introduction	1
1.1 Problem Background	1
1.2 Problem Statement	2
1.3 Project Objectives	2
1.4 Project Scope	2
II Database Design	4
2.1 Problem Domain	4
2.2 System Users	4
2.3 System Objectives	4
2.4 System Overview	5
2.5 Desain Database dan ERD	6
2.5.1 Entitas dan Atribut	6
2.5.2 Entity Relationship Diagram (ERD)	7
2.5.3 Relasi Tabel dan Atribut	8
2.5.4 Database Normalization	8
2.5.4.1 Initial Relational Schema	9
2.5.4.2 First Normal Form (1NF)	9
2.5.4.3 Second Normal Form (2NF)	10
2.5.4.4 Third Normal Form (3NF)	10
2.5.4.5 Relational Schema Result (3NF)	11
III Implementation and Testing	12
3.1 Database Implementation	12
3.1.1 Relational Schema	12
3.2 Key Constraints and Table Relationships	13
3.2.1 Primary Key	13
3.2.2 Foreign Key	13
3.2.3 Table Relationships	14
3.3 SQL Implementation	14
3.3.1 Table Creation (DDL)	14

3.3.2	Transaction Table Implementation	17
3.3.3	Data Manipulation (DML)	17
3.4	Pengujian Sistem dengan CRUD Data	18
3.4.1	Metode Pengujian	18
3.4.2	Setup Data Minimal (Agar FK Tidak Error)	18
3.4.3	Skenario Uji: CRUD Lengkap pada Book	19
3.4.3.1	Create, Read, Update, Delete (Book)	19
3.4.4	Skenario Uji: Read pada Member (dengan 1 Data Uji)	20
3.4.4.1	Create Singkat dan Read (Member)	20
3.4.5	Skenario Uji: Transaksi Borrowing	21
3.4.5.1	Create dan Update Status (Borrowing)	21
3.5	Implementasi Website	23
3.5.1	Teknologi yang Digunakan	23
3.5.2	Arsitektur	23
3.5.3	Modul Utama Website	24
3.5.3.1	Modul Manajemen Buku	24
3.5.3.2	Modul Peminjaman Buku	24
3.5.3.3	Modul Manajemen Member	25
3.5.3.4	Modul Manajemen Alamat	26
3.5.4	Keamanan dan Autentikasi	26
3.5.5	Alur Login dan Hak Akses Pengguna	26
3.5.5.1	Login sebagai Member	27
3.5.5.2	Login sebagai Staff	28
3.5.5.3	Akun Uji (Demo Account)	29
3.6	Pembahasan	30
3.6.1	Kesesuaian Sistem dengan Tujuan	30
3.6.2	Kelebihan Sistem	30
3.6.3	Keterbatasan Sistem	30
IV	Kesimpulan	32
4.1	Kesimpulan	32
4.2	Saran Pengembangan	32
	DAFTAR PUSTAKA	33
A	Script SQL Basis Data InfiniRead	34

B Dokumentasi Antarmuka dan Fitur Website InfiniRead**37**

BAB I

Introduction

1.1 Problem Background

Perkembangan teknologi informasi telah mendorong perubahan besar dalam sistem pengelolaan dan layanan perpustakaan di lingkungan perguruan tinggi. Pemanfaatan teknologi digital memungkinkan perluasan jangkauan akses informasi, peningkatan efisiensi layanan, serta kemudahan dalam pengelolaan koleksi secara terintegrasi. Pengembangan koleksi digital di universitas juga terbukti meningkatkan ketersediaan sumber belajar dan mempercepat proses distribusi informasi kepada pengguna (Saputra, 2025).

Penerapan sistem perpustakaan digital di Indonesia terus berkembang melalui berbagai inisiatif seperti IndonesiaDLN dan IOS. Meskipun demikian, masih terdapat sejumlah kendala yang berkaitan dengan infrastruktur, aksesibilitas, dan kesiapan sumber daya manusia dalam pengelolaan sistem yang efektif (Saputra, 2025). Kondisi ini menunjukkan pentingnya penerapan sistem yang tidak hanya berfokus pada teknologi, tetapi juga pada kemudahan penggunaan, keamanan data, dan integrasi layanan.

Dari sisi teknis, rancangan sistem perpustakaan digital modern umumnya menggunakan basis data relasional untuk mendukung pengelolaan data anggota, koleksi buku, dan transaksi peminjaman. Agar keamanan data dan hak akses pengguna tetap terjaga, sistem ini menerapkan mekanisme *access control* berbasis peran sesuai kebijakan institusi (Ray and Chakraborty, 2006).

Selain itu, sistem otomatisasi layanan juga menjadi bagian penting dalam pengembangan perpustakaan digital. Beberapa perpustakaan universitas di luar negeri telah menerapkan kebijakan pengembalian otomatis tanpa denda untuk meningkatkan kepatuhan pengguna dan mengurangi beban administratif pustakawan (Taylor's University Library, 2024).

Berdasarkan kebutuhan tersebut, proyek *InfiniRead* dikembangkan sebagai sistem perpustakaan digital berbasis web dengan rancangan basis data terintegrasi, kontrol akses pengguna, serta fitur penarikan otomatis buku digital setelah masa pinjam berakhir. Sistem ini diharapkan dapat meningkatkan efisiensi pengelolaan, keamanan data, serta kenyamanan layanan bagi sivitas akademika.

1.2 Problem Statement

Berdasarkan uraian pada bagian latar belakang, permasalahan yang diangkat dalam proyek ini dapat dirumuskan sebagai berikut:

- Bagaimana merancang skema basis data yang efisien untuk menyimpan dan mengelola data anggota, koleksi buku, serta transaksi peminjaman pada sistem perpustakaan digital?
- Bagaimana menerapkan mekanisme kontrol akses berbasis aturan atau atribut agar setiap pengguna memperoleh hak akses sesuai kebijakan, termasuk pembatasan usia?
- Bagaimana membangun fitur otomasi yang mampu menarik kembali buku secara otomatis setelah masa pinjam berakhir guna mencegah keterlambatan dan denda?

1.3 Project Objectives

Tujuan utama dari proyek *InfiniRead* adalah:

- Menghasilkan rancangan dan implementasi basis data terintegrasi untuk entitas anggota, buku, dan transaksi peminjaman.
- Menerapkan mekanisme kontrol akses pada tingkat aplikasi maupun basis data agar sejalan dengan kebijakan institusi.
- Mengembangkan fitur otomasi berupa penarikan otomatis buku digital setelah masa pinjam berakhir untuk meningkatkan efisiensi dan kepatuhan pengguna.

1.4 Project Scope

Ruang lingkup proyek *InfiniRead* dibatasi pada perancangan dan implementasi sistem basis data serta aplikasi pendukung untuk sistem perpustakaan digital sesuai dengan kebutuhan tugas mata kuliah. Cakupan proyek ini meliputi aspek-aspek berikut:

1. Database Design

Proyek ini mencakup identifikasi *problem domain*, penentuan entitas, atribut,

dan relasi yang terlibat dalam sistem perpustakaan digital. Perancangan dilakukan menggunakan *Entity Relationship Diagram* (ERD) sebagai model konseptual, yang selanjutnya dinormalisasi hingga mencapai *Third Normal Form* (3NF) untuk memastikan konsistensi dan efisiensi penyimpanan data.

2. Database Implementation

Desain basis data yang telah dirancang kemudian diimplementasikan ke dalam skema relasional menggunakan SQL. Implementasi meliputi pembuatan tabel, penetapan *primary key* dan *foreign key*, serta penerapan *constraint* yang diperlukan untuk menjaga integritas data.

3. Application Implementation

Proyek ini mencakup pengembangan aplikasi berbasis web yang terhubung dengan basis data untuk mendukung pengelolaan data anggota, koleksi buku, dan transaksi peminjaman. Aplikasi menyediakan fungsi dasar *Create, Read, Update, Delete* (CRUD) sesuai dengan peran pengguna.

4. System Testing and Results

Pengujian dilakukan untuk memastikan bahwa setiap fungsi sistem berjalan sesuai dengan kebutuhan yang telah ditentukan. Pengujian meliputi pengelolaan data, relasi antar tabel, serta validasi hasil melalui eksekusi query dan tampilan aplikasi.

5. Project Limitations

Proyek ini dikembangkan sebagai prototipe akademik dan tidak mencakup pengembangan sistem berskala industri. Aspek seperti optimasi performa tingkat lanjut, keamanan lanjutan, dan integrasi dengan sistem eksternal berada di luar cakupan proyek ini.

BAB II

Database Design

2.1 Problem Domain

Perpustakaan konvensional masih menghadapi berbagai kendala seperti keterbatasan akses, proses administrasi manual yang lambat, serta kesulitan dalam pelacakan data buku dan anggota. Dalam era digitalisasi kampus, dibutuhkan sistem yang mampu mengelola data secara otomatis, efisien, dan terintegrasi. Sistem *InfiniRead* dikembangkan sebagai solusi berbasis web untuk mendigitalisasi seluruh proses pengelolaan perpustakaan, mulai dari pendaftaran anggota, pengaturan keanggotaan, hingga peminjaman dan pengembalian buku digital. Sistem ini juga menerapkan filterisasi berdasarkan tanggal lahir agar setiap anggota hanya dapat mengakses koleksi sesuai kelompok usia. Dengan rancangan basis data relasional yang terstruktur, *InfiniRead* mampu meningkatkan efisiensi, keamanan, dan akurasi pengelolaan data perpustakaan.

2.2 System Users

1. Staff

Bertugas mengelola data buku, member, dan transaksi peminjaman. Staff juga berperan dalam memastikan ketersediaan koleksi serta memantau aktivitas peminjaman agar sistem berjalan sesuai kebijakan perpustakaan.

2. Member

Pengguna yang terdaftar dalam sistem dan memiliki status keanggotaan aktif. Member dapat melihat katalog buku, meminjam buku digital, memperbarui langganan, serta memantau riwayat peminjaman dan keanggotaan secara mandiri.

2.3 System Objectives

1. **Membangun basis data terintegrasi** yang mengelola seluruh entitas utama seperti *Book*, *Member*, *Borrowing*, *Membership*, dan *Payment* secara konsisten dan efisien.

2. **Mengoptimalkan efisiensi layanan perpustakaan digital** melalui fitur otomatisasi, termasuk *auto-return* untuk pengembalian buku tanpa denda setelah masa pinjam berakhir.
3. **Meningkatkan akurasi dan keamanan data** dengan penerapan filterisasi berdasarkan tanggal lahir guna menyesuaikan akses koleksi sesuai batas usia.
4. **Menyediakan sistem yang mudah digunakan** oleh staff dan member dengan proses administrasi yang cepat, transparan, dan terpusat dalam satu platform digital.

2.4 System Overview

InfiniRead merupakan platform perpustakaan digital berbasis web yang memudahkan anggota untuk meminjam dan membaca buku secara daring. Sistem ini dikembangkan sebagai solusi modern terhadap keterbatasan sistem perpustakaan konvensional, yakni kendala akses, waktu, maupun pengelolaan data peminjaman yang kurang efisien.

Seluruh proses administrasi dalam *InfiniRead* dilakukan secara digital, mencakup pendaftaran anggota, pengelolaan koleksi buku, peminjaman, hingga pengembalian otomatis. Seluruh aktivitas tersebut tercatat dalam basis data sehingga mendukung efisiensi dan akurasi pengelolaan perpustakaan.

Berbeda dengan sistem perpustakaan konvensional, *InfiniRead* tidak menerapkan denda keterlambatan. Setiap buku yang mencapai batas waktu peminjaman akan dikembalikan secara otomatis oleh sistem sehingga mempermudah pengelolaan stok buku. Batas waktu peminjaman buku ditentukan berdasarkan jumlah halaman sehingga memberikan fleksibilitas bagi pengguna.

Sistem *InfiniRead* juga menerapkan filterisasi konten berdasarkan usia pengguna. Data tanggal lahir yang diinput saat proses pendaftaran digunakan untuk menentukan kelompok usia pengguna. Data ini akan digunakan untuk memfilter koleksi buku sesuai rating usia yang telah ditetapkan pada setiap buku untuk menjaga keamanan dan kesesuaian bacaan bagi setiap kelompok umur.

2.5 Desain Database dan ERD

2.5.1 Entitas dan Atribut

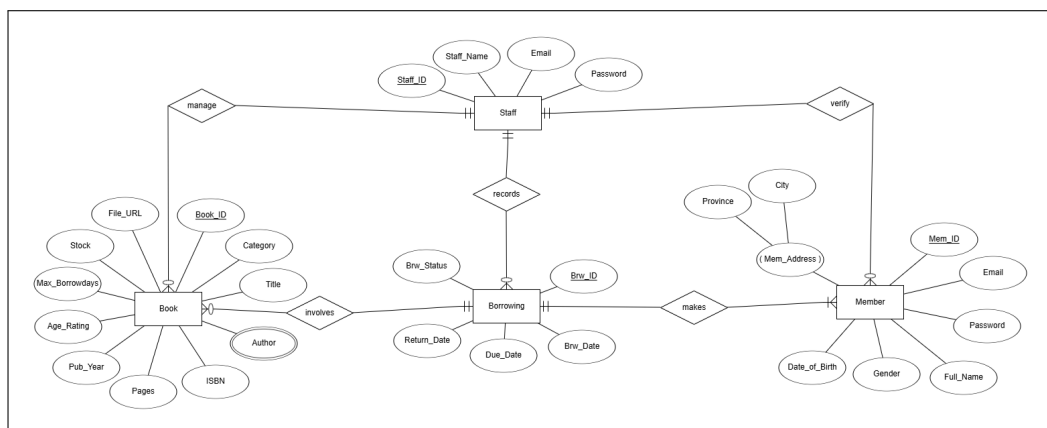
Tabel 2.1: Entitas dan Atribut

Entitas	Atribut	Deskripsi
Book	Book_ID (PK)	ID unik buku dengan format XX – 000000 2 huruf pertama: penanda kategori 6 digit terakhir: ID judul
	Category	Nama Kategori
	Title	Judul buku
	Author	Nama penulis
	ISBN	Nomor ISBN
	Pages	Jumlah halaman
	Pub_Year	Tahun terbit
	Age_Rating	Rating umur
	Max_Borrowdays	Lama maksimal peminjaman buku Dengan aturan ≤ 100 halaman \rightarrow 3 hari 101–300 halaman \rightarrow 7 hari > 300 halaman \rightarrow 14 hari
	Stock	Jumlah ketersediaan buku
	File _{URL}	Link buku
Borrowing	Brw_ID (PK)	ID unik transaksi peminjaman
	Brw_Date	Tanggal peminjaman
	Due_Date	Batas tanggal peminjaman
	Return_Date	Tanggal pengembalian
	Brw_Status	Status pengembalian (true: sudah dikembalikan, false: masih dipinjam)
Member	Mem_ID (PK)	ID unik member

Entitas	Atribut	Deskripsi
	Full_Name	Nama lengkap
	Gender	Jenis kelamin
	Date_of_Birth	Tanggal lahir
	Email	Akun email
	Password	Password akun
	Mem_Address	Alamat member
Staff	Staff_ID (PK)	Id unik staff
	Staff_Name	Nama staff
	Email	Akun email
	Password	Password akun

2.5.2 Entity Relationship Diagram (ERD)

ERD sistem *InfiniRead* menggambarkan hubungan antar entitas utama: *Book*, *Borrowing*, *Member*, dan *Staff*.



Gambar 2.1: Entity Relationship Diagram (ERD) Sistem *InfiniRead*

Entity Relationship Diagram (ERD) pada sistem *InfiniRead* merepresentasikan model konseptual yang digunakan untuk menggambarkan entitas, atribut, dan relasi utama dalam sistem perpustakaan digital. Pada tahap ini, beberapa data seperti *City* dan *Province* masih direpresentasikan sebagai atribut dalam entitas *Member*.

Selanjutnya, ERD tersebut dikonversi ke dalam skema relasional dan dilakukan proses normalisasi hingga mencapai Third Normal Form (3NF). Proses ini menghasilkan pemisahan entitas *City* dan *Province* menjadi tabel tersendiri untuk menghilangkan ketergantungan transitif serta menjaga konsistensi dan integritas data pada basis data sistem *InfiniRead*.

2.5.3 Relasi Tabel dan Atribut

Tabel 2.2: Relasi Tabel dan Atribut

No	Relasi	Entitas 1	Kardinalitas 1	Entitas 2	Kardinalitas 2	Keterangan
1	involves	Book	Mandatory One	Borrowing	Optional Many	Sekali peminjaman bisa dilakukan dengan banyak buku.
2	makes	Member	Mandatory One	Borrowing	Mandatory Many	Satu member bisa memiliki banyak transaksi peminjaman.
3	records	Staff	Optional Many	Borrowing	Mandatory One	Setiap staff bisa mencatat banyak transaksi peminjaman
4	manage	Staff	Optional Many	Book	Mandatory One	Setiap staff bisa mengatur banyak buku
5	verify	Staff	Optional Many	Member	Mandatory One	Setiap staff bisa memverifikasi banyak member

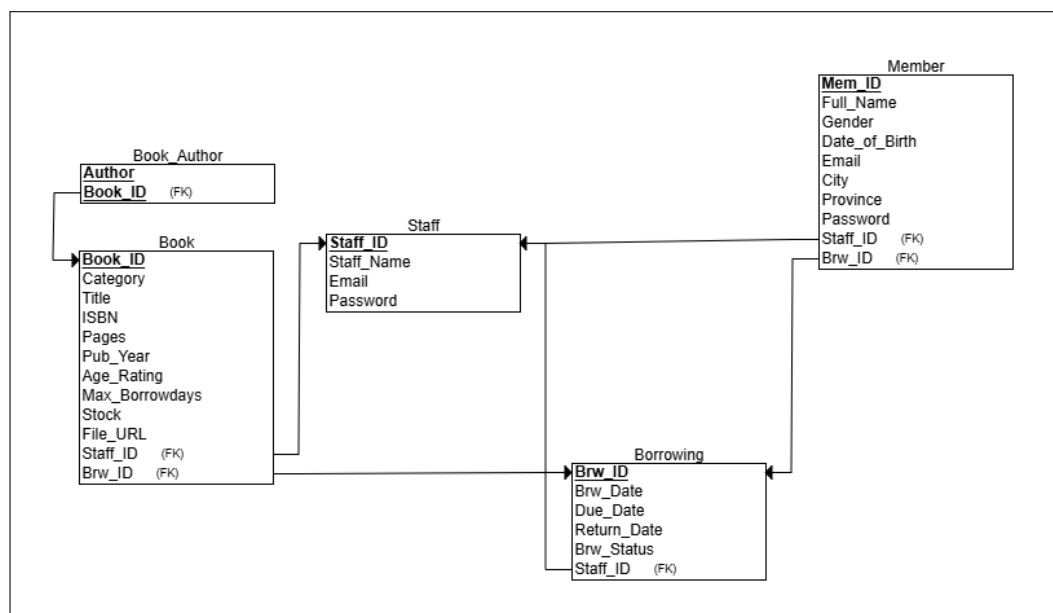
2.5.4 Database Normalization

Normalisasi basis data dilakukan untuk memastikan bahwa struktur basis data yang dirancang bebas dari redundansi, inkonsistensi, serta anomali manipulasi data. Pada sistem *InfiniRead*, proses normalisasi dilakukan secara bertahap mulai dari pe-

memeriksa *First Normal Form* (1NF) hingga perbaikan struktur untuk mencapai *Third Normal Form* (3NF).

2.5.4.1 Initial Relational Schema

Gambar berikut menunjukkan hasil konversi awal dari *Entity Relationship Diagram* (ERD) ke dalam skema relasional sebelum dilakukan proses normalisasi lanjutan.



Gambar 2.2: Hasil Konversi Awal ERD ke Skema Relasional Sistem *InfiniRead*

2.5.4.2 First Normal Form (1NF)

Suatu tabel dikatakan memenuhi *First Normal Form* apabila setiap atribut memiliki nilai atomik (tidak dapat dibagi lagi) dan tidak terdapat kelompok atribut berulang.

Berdasarkan rancangan awal sistem *InfiniRead*, seluruh entitas utama seperti *Book*, *Member*, *Borrowing*, dan *Staff* telah menggunakan atribut dengan nilai tunggal, antara lain *Book_ID*, *Title*, *ISBN*, *Email*, dan *Date_of_Birth*. Kondisi ini menunjukkan bahwa secara umum struktur tabel telah memenuhi ketentuan 1NF.

Namun, pada entitas *Book* terdapat atribut *Author* yang berpotensi menyimpan lebih dari satu nilai untuk satu buku. Keadaan ini melanggar prinsip 1NF karena satu atribut menyimpan data multivalued. Untuk mengatasi permasalahan tersebut, atribut

Author dipisahkan ke dalam tabel tersendiri, yaitu *Book_Author*. Dengan pemisahan ini, setiap kombinasi penulis dan buku direpresentasikan dalam satu baris data, sehingga seluruh tabel memenuhi ketentuan 1NF.

2.5.4.3 Second Normal Form (2NF)

Second Normal Form mensyaratkan bahwa tabel telah memenuhi 1NF dan setiap atribut non-kunci harus bergantung sepenuhnya pada *primary key*.

Pada sistem *InfiniRead*, sebagian besar tabel menggunakan *primary key* tunggal, seperti *Staff_ID*, *Mem_ID*, dan *Book_ID*. Oleh karena itu, tidak terdapat ketergantungan parsial antara atribut non-kunci terhadap sebagian dari kunci utama.

Tabel *Book_Author* menggunakan *composite primary key* yang terdiri dari *Author* dan *Book_ID*. Seluruh atribut pada tabel tersebut sepenuhnya bergantung pada kombinasi kedua atribut kunci tersebut dan tidak terdapat atribut non-kunci yang bergantung hanya pada salah satu bagian kunci. Dengan demikian, seluruh tabel dalam sistem telah memenuhi ketentuan *Second Normal Form* (2NF).

2.5.4.4 Third Normal Form (3NF)

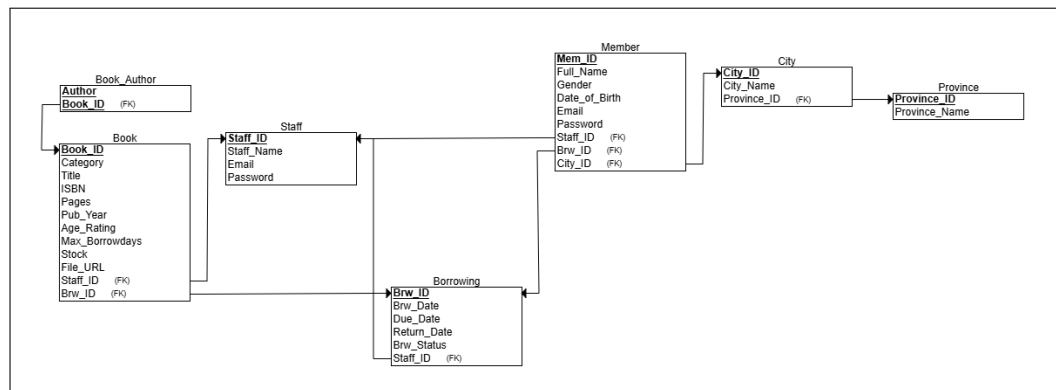
Third Normal Form (3NF) mengharuskan bahwa setiap tabel telah memenuhi ketentuan *Second Normal Form* (2NF) serta tidak mengandung ketergantungan transitif, yaitu kondisi ketika atribut non-kunci bergantung pada atribut non-kunci lainnya.

Pada rancangan awal sistem *InfiniRead*, tabel *Member* menyimpan informasi lokasi berupa atribut *City* dan *Province* secara langsung. Dalam struktur tersebut, atribut *Province* tidak bergantung langsung pada *Mem_ID*, melainkan bergantung pada atribut *City*. Ketergantungan ini menimbulkan ketergantungan transitif yang melanggar prinsip 3NF.

Untuk menghilangkan ketergantungan transitif tersebut, data lokasi dipisahkan ke dalam dua entitas terpisah, yaitu *City* dan *Province*. Tabel *City* menyimpan informasi nama kota dan memiliki *foreign key* yang mereferensikan tabel *Province*. Selanjutnya, tabel *Member* hanya menyimpan *City_ID* sebagai *foreign key*. Dengan perbaikan ini, setiap atribut non-kunci pada tabel *Member* bergantung langsung pada *primary key*.

2.5.4.5 Relational Schema Result (3NF)

Gambar berikut menunjukkan hasil akhir skema relasional setelah proses normalisasi hingga mencapai *Third Normal Form* (3NF).



Gambar 2.3: Skema Relasional Sistem *InfiniRead* Setelah Normalisasi 3NF

Setelah seluruh tahapan normalisasi dilakukan, struktur basis data sistem *InfiniRead* telah memenuhi *Third Normal Form* (3NF). Proses normalisasi menghasilkan tujuh tabel utama, yaitu:

1. *Staff*
2. *Member*
3. *Book*
4. *Borrowing*
5. *Book_Author*
6. *City*
7. *Province*

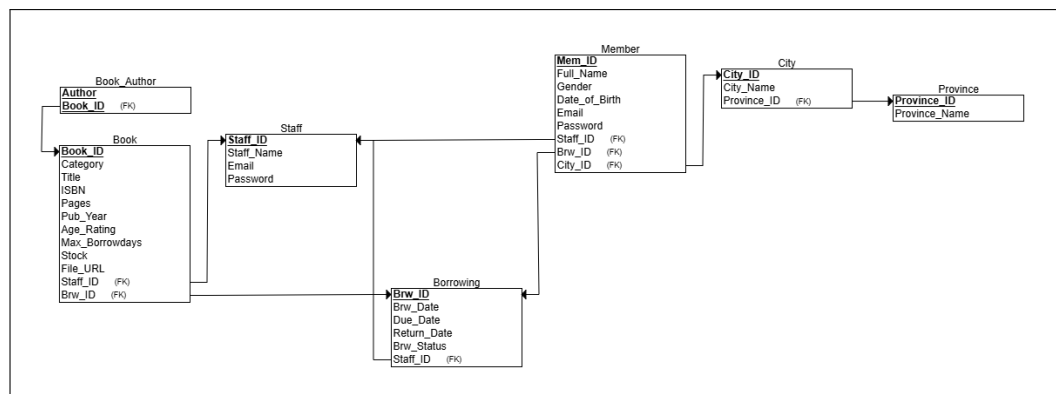
Struktur basis data hasil normalisasi ini mampu meminimalkan redundansi data, menghilangkan ketergantungan parsial dan transitif, serta menjaga konsistensi dan integritas relasi antar entitas melalui penerapan *primary key* dan *foreign key* secara tepat.

BAB III

Implementation and Testing

3.1 Database Implementation

3.1.1 Relational Schema



Gambar 3.1: Skema Relasional Sistem *InfiniRead* Setelah Normalisasi 3NF

Berdasarkan hasil perancangan dan normalisasi hingga *Third Normal Form* (3NF), sistem *InfiniRead* diimplementasikan menggunakan tujuh tabel utama, yaitu *Staff*, *Member*, *Book*, *Borrowing*, *Book_Author*, *City*, dan *Province*. Skema relasional ini dirancang untuk merepresentasikan hubungan antar entitas secara efisien dan mendukung kebutuhan operasional sistem perpustakaan digital.

Tabel *Staff* menyimpan data akun petugas yang bertanggung jawab dalam pengelolaan sistem, termasuk pengelolaan data buku, verifikasi member, dan pencatatan transaksi peminjaman.

Tabel *Book* menyimpan informasi detail koleksi buku digital. Setiap buku dikelola oleh seorang staff yang direpresentasikan melalui atribut *Staff_ID* sebagai *foreign key*.

Relasi antara buku dan penulis direalisasikan melalui tabel *Book_Author*. Tabel ini memungkinkan satu buku memiliki lebih dari satu penulis tanpa menimbulkan redundansi data.

Tabel *Member* menyimpan data identitas pengguna sistem. Informasi domisili member direpresentasikan melalui *City_ID*, sementara *Staff_ID* menunjukkan petugas yang melakukan verifikasi akun.

Tabel *Borrowing* menyimpan data transaksi peminjaman buku, termasuk tanggal peminjaman, tanggal jatuh tempo, tanggal pengembalian, dan status peminjaman. Setiap transaksi dicatat oleh staff dan dikaitkan dengan satu member serta satu buku.

Tabel *City* dan *Province* digunakan untuk menyimpan data alamat. Pemisahan kedua tabel ini dilakukan untuk menghilangkan ketergantungan transitif dan menjaga konsistensi data wilayah.

3.2 Key Constraints and Table Relationships

Penerapan *key constraints* bertujuan untuk menjaga integritas data dan memastikan hubungan antar tabel berjalan sesuai dengan skema relasional hasil normalisasi.

3.2.1 Primary Key

Setiap tabel memiliki *primary key* yang bersifat unik, yaitu:

- *Staff*: *Staff_ID*
- *Member*: *Mem_ID*
- *Book*: *Book_ID*
- *Borrowing*: *Brw_ID*
- *City*: *City_ID*
- *Province*: *Province_ID*
- *Book_Author*: (*Author*, *Book_ID*)

3.2.2 Foreign Key

Relasi antar tabel dibangun menggunakan *foreign key* sebagai berikut:

- *Book.Staff_ID* → *Staff.Staff_ID*
- *Member.Staff_ID* → *Staff.Staff_ID*
- *Member.City_ID* → *City.City_ID*
- *City.Province_ID* → *Province.Province_ID*

- *Book_Author.Book_ID* → *Book.Book_ID*
- *Borrowing.Staff_ID* → *Staff.Staff_ID*
- *Borrowing.Mem_ID* → *Member.Mem_ID*
- *Borrowing.Book_ID* → *Book.Book_ID*

3.2.3 Table Relationships

Berdasarkan *key constraints* yang diterapkan, hubungan antar tabel dapat dirangkum sebagai berikut:

- Satu *Staff* dapat mengelola banyak *Book* dan memverifikasi banyak *Member*
- Satu *Member* dapat melakukan banyak transaksi *Borrowing*
- Satu *Book* dapat terlibat dalam banyak transaksi *Borrowing*
- Relasi *Book* dan *Author* bersifat *many-to-many*
- Satu *Province* memiliki banyak *City*

3.3 SQL Implementation

Implementasi basis data sistem *InfiniRead* dilakukan menggunakan *Structured Query Language* (SQL) pada sistem manajemen basis data relasional. Tahapan ini mencakup pembuatan tabel menggunakan *Data Definition Language* (DDL), penerapan *primary key* dan *foreign key*, serta pengisian data awal untuk keperluan pengujian sistem.

3.3.1 Table Creation (DDL)

Setiap tabel dibuat menggunakan perintah `CREATE TABLE` dan disesuaikan dengan skema relasional hasil normalisasi hingga *Third Normal Form* (3NF).

Tabel *Province* dan *City* digunakan untuk menyimpan data wilayah secara terpisah guna menjaga konsistensi dan menghindari redundansi data alamat.

Listing III.1: Implementasi Tabel Province dan City

```
1 CREATE TABLE Province (
  Province_ID INT NOT NULL,
```

```

3 Province_Name VARCHAR(100) NOT NULL,
   PRIMARY KEY (Province_ID)
5 );

7 CREATE TABLE City (
   City_ID INT NOT NULL,
   Province_ID INT NOT NULL,
   City_Name VARCHAR(100) NOT NULL,
11  PRIMARY KEY (City_ID),
   FOREIGN KEY (Province_ID)
13 REFERENCES Province(Province_ID)
   );

```

Tabel *Staff* menyimpan data akun petugas sistem yang bertanggung jawab terhadap pengelolaan data dan pencatatan transaksi.

Listing III.2: Implementasi Tabel Staff

```

CREATE TABLE Staff (
2  Staff_ID INT NOT NULL AUTO_INCREMENT,
   Staff_Name VARCHAR(100) NOT NULL,
4  Email VARCHAR(100) NOT NULL,
   Password VARCHAR(255) NOT NULL,
6  PRIMARY KEY (Staff_ID)
   );

```

Tabel *Member* menyimpan data identitas pengguna sistem. Setiap member dikaitkan dengan satu kota domisili dan diverifikasi oleh seorang staff.

Listing III.3: Implementasi Tabel Member

```

1 CREATE TABLE Member (
   Mem_ID INT NOT NULL AUTO_INCREMENT,
3  Full_Name VARCHAR(100) NOT NULL,
   Gender ENUM('Man', 'Woman') NOT NULL,
5  Date_of_Birth DATE NOT NULL,
   Email VARCHAR(50) NOT NULL,
7  Password VARCHAR(255) NOT NULL,
   City_ID INT NOT NULL,
9  Staff_ID INT NOT NULL,
   Is_Verified TINYINT(1) NOT NULL DEFAULT 0,
11 PRIMARY KEY (Mem_ID),
   UNIQUE (Email),
13 FOREIGN KEY (City_ID)

```

```

REFERENCES City(City_ID),
15 FOREIGN KEY (Staff_ID)
REFERENCES Staff(Staff_ID)
17 );

```

Tabel *Book* menyimpan informasi koleksi buku digital dan dikaitkan dengan staff yang mengelola data buku tersebut.

Listing III.4: Implementasi Tabel Book

```

1 CREATE TABLE Book (
  Book_ID VARCHAR(20) NOT NULL,
3 Category VARCHAR(20) NOT NULL,
  Title VARCHAR(100) NOT NULL,
5 ISBN VARCHAR(50) NOT NULL,
  Pages INT NOT NULL,
7 Pub_Year INT NOT NULL,
  Age_Rating INT NOT NULL,
9 Max_Borrowdays INT NOT NULL,
  Stock INT NOT NULL,
11 Staff_ID INT NOT NULL,
  File_URL VARCHAR(255) NOT NULL,
13 PRIMARY KEY (Book_ID),
  FOREIGN KEY (Staff_ID)
15 REFERENCES Staff(Staff_ID)
);

```

Relasi antara buku dan penulis direalisasikan melalui tabel *Book_Author* menggunakan *composite primary key* untuk mendukung relasi *many-to-many*.

Listing III.5: Implementasi Tabel Book_Author

```

CREATE TABLE Book_Author (
2 Author VARCHAR(100) NOT NULL,
  Book_ID VARCHAR(20) NOT NULL,
4 PRIMARY KEY (Author, Book_ID),
  FOREIGN KEY (Book_ID)
6 REFERENCES Book(Book_ID)
);

```

3.3.2 Transaction Table Implementation

Tabel *Borrowing* digunakan untuk menyimpan data transaksi peminjaman buku. Tabel ini terhubung langsung dengan tabel *Staff*, *Member*, dan *Book* melalui *foreign key*.

Listing III.6: Implementasi Tabel Borrowing

```

1 CREATE TABLE Borrowing (
    Brw_ID INT NOT NULL AUTO_INCREMENT,
3    Brw_Date DATE NOT NULL,
    Due_Date DATE NOT NULL,
5    Return_Date DATE DEFAULT NULL,
    Brw_Status ENUM('BORROWED', 'RETURNED', 'OVERDUE')
7        NOT NULL DEFAULT 'BORROWED',
    Staff_ID INT NOT NULL,
9    Mem_ID INT NOT NULL,
    Book_ID VARCHAR(20) NOT NULL,
11   PRIMARY KEY (Brw_ID),
    FOREIGN KEY (Staff_ID)
13       REFERENCES Staff (Staff_ID),
    FOREIGN KEY (Mem_ID)
15       REFERENCES Member (Mem_ID),
    FOREIGN KEY (Book_ID)
17       REFERENCES Book (Book_ID)
);

```

3.3.3 Data Manipulation (DML)

Untuk keperluan pengujian sistem, dilakukan pengisian data awal menggunakan perintah INSERT guna mensimulasikan proses operasional sistem.

Listing III.7: Contoh Insert Data Awal

```

INSERT INTO Province
2 VALUES (1, 'Daerah_Istimewa_Yogyakarta');

4 INSERT INTO City
VALUES (1, 1, 'Yogyakarta');

6
8 INSERT INTO Staff (Staff_Name, Email, Password)
VALUES ('Admin', 'admin@infiniread.id', 'password123');

```

```

10 INSERT INTO Book
VALUES ('BK-000001', 'Computer', 'Database_Systems',
12      '9781234567890', 450, 2023, 18,
      14, 5, 1, 'file/book1.pdf');

```

Implementasi SQL ini memastikan struktur basis data sesuai dengan hasil perancangan, menjaga integritas referensial, serta mendukung operasional sistem *Infini-Read*.

3.4 Pengujian Sistem dengan CRUD Data

3.4.1 Metode Pengujian

Pengujian dilakukan pada lingkungan lokal menggunakan XAMPP dengan DBMS MariaDB. Seluruh pengujian dieksekusi melalui MariaDB Command Line (*mysql client*). Fokus pengujian adalah validasi operasi basis data menggunakan **sampel data**. Operasi CRUD **diwakilkan** pada tabel *book* (CRUD lengkap), sedangkan tabel *member* dan *borrowing* diuji pada operasi yang paling relevan dengan proses bisnis sistem (contoh: pengecekan data *member* dan transaksi *peminjaman*). Karena terdapat *foreign key constraint*, dilakukan *setup* data minimal (*province*, *city*, *staff*) sebelum pengujian.

3.4.2 Setup Data Minimal (Agar FK Tidak Error)

Query SQL:

```

1  -- 1) PROVINCE (parent)
INSERT INTO province (Province_ID, Province_Name)
3  VALUES (99, 'TEST PROVINCE');

5  -- 2) CITY (child of province)
INSERT INTO city (City_ID, Province_ID, City_Name)
7  VALUES (9901, 99, 'TEST CITY');

9  -- 3) STAFF (dipakai book/member/borrowing)
INSERT INTO staff (Staff_Name, Email, Password)
11 VALUES ('Staff Test', 'stafftest@infiniread.test',
      'masuk123');
SET @staff_id := LAST_INSERT_ID();
13 SELECT @staff_id AS staff_id_uji;

```

```

MariaDB [infiniread_xampptest]> INSERT INTO province (Province_ID, Province_Name)
-> VALUES (99, 'TEST PROVINCE');
Query OK, 1 row affected (0.010 sec)

MariaDB [infiniread_xampptest]>
MariaDB [infiniread_xampptest]> INSERT INTO city (City_ID, Province_ID, City_Name)
-> VALUES (9901, 99, 'TEST CITY');
Query OK, 1 row affected (0.005 sec)

MariaDB [infiniread_xampptest]>
MariaDB [infiniread_xampptest]> INSERT INTO staff (Staff_Name, Email, Password)
-> VALUES ('Staff Test', 'stafftest@infiniread.test', 'masuk123');
Query OK, 1 row affected (0.004 sec)

MariaDB [infiniread_xampptest]>
MariaDB [infiniread_xampptest]> SET @staff_id := LAST_INSERT_ID();
Query OK, 0 rows affected (0.002 sec)

MariaDB [infiniread_xampptest]> SELECT @staff_id AS staff_id_uji;
+-----+
| staff_id_uji |
+-----+
|              |
+-----+
|              |
+-----+
1 row in set (0.001 sec)

```

Gambar 3.2: Setup data minimal pada MariaDB CLI (province, city, staff).

3.4.3 Skenario Uji: CRUD Lengkap pada Book

3.4.3.1 Create, Read, Update, Delete (Book)

Query SQL:

```

1  -- CREATE
   INSERT INTO book
3  (Book_ID, Category, Title, ISBN,
   Pages, Pub_Year, Age_Rating,
5  Max_Borrowdays, Stock, Staff_ID, File_URL)
   VALUES
7  ('TST001', 'Academic', 'Testing Book
   CRUD', '9789999999999',
   200, 2024, 18,
9  14, 3, @staff_id, 'https://example.com/tst001.pdf');

11 -- READ (verifikasi insert)
   SELECT Book_ID, Title, Stock, Staff_ID
13 FROM book
   WHERE Book_ID='TST001';

15
   -- UPDATE
17 UPDATE book
   SET Stock=5, Title='Testing Book CRUD (Updated)'
19 WHERE Book_ID='TST001';

21
   -- READ (verifikasi update)
   SELECT Book_ID, Title, Stock

```

```

23 FROM book
WHERE Book_ID='TST001';

25 -- DELETE

27 DELETE FROM book WHERE Book_ID='TST001';

29 -- READ (verifikasi delete)
SELECT Book_ID FROM book WHERE Book_ID='TST001';

```

```

MariaDB [infiniread_xampptest]> -- CREATE
MariaDB [infiniread_xampptest]> INSERT INTO book
-> (Book_ID, Category, Title, ISBN,
-> Pages, Pub_Year, Age_Rating,
-> Max_Borrowdays, Stock, Staff_ID, File_URL)
-> VALUES
-> ('TST001','Academic','Testing Book CRUD','9789999999999',
-> 200,2024,18,
-> 14,3,@staff_id,'https://example.com/tst001.pdf');
Query OK, 1 row affected (0.006 sec)

MariaDB [infiniread_xampptest]>
MariaDB [infiniread_xampptest]> -- READ (verifikasi insert)
MariaDB [infiniread_xampptest]> SELECT Book_ID, Title, Stock, Staff_ID
-> FROM book
-> WHERE Book_ID='TST001';
+-----+-----+-----+-----+
| Book_ID | Title           | Stock | Staff_ID |
+-----+-----+-----+-----+
| TST001  | Testing Book CRUD | 3     | 1        |
+-----+-----+-----+-----+
1 row in set (0.001 sec)

MariaDB [infiniread_xampptest]>
MariaDB [infiniread_xampptest]> -- UPDATE
MariaDB [infiniread_xampptest]> UPDATE book
-> SET Stock=5, Title='Testing Book CRUD (Updated)'
-> WHERE Book_ID='TST001';
Query OK, 1 row affected (0.016 sec)
Rows matched: 1  Changed: 1  Warnings: 0

MariaDB [infiniread_xampptest]>
MariaDB [infiniread_xampptest]> -- READ (verifikasi update)
MariaDB [infiniread_xampptest]> SELECT Book_ID, Title, Stock
-> FROM book
-> WHERE Book_ID='TST001';
+-----+-----+-----+-----+
| Book_ID | Title           | Stock |
+-----+-----+-----+-----+
| TST001  | Testing Book CRUD (Updated) | 5     |
+-----+-----+-----+-----+
1 row in set (0.000 sec)

MariaDB [infiniread_xampptest]>
MariaDB [infiniread_xampptest]> -- DELETE
MariaDB [infiniread_xampptest]> DELETE FROM book WHERE Book_ID='TST001';
Query OK, 1 row affected (0.003 sec)

MariaDB [infiniread_xampptest]>
MariaDB [infiniread_xampptest]> -- READ (verifikasi delete)
MariaDB [infiniread_xampptest]> SELECT Book_ID FROM book WHERE Book_ID='TST001';
Empty set (0.001 sec)

```

Gambar 3.3: Pengujian CRUD lengkap pada tabel book melalui MariaDB CLI.

3.4.4 Skenario Uji: Read pada Member (dengan 1 Data Uji)

3.4.4.1 Create Singkat dan Read (Member)

Query SQL:

```
-- CREATE
```



```

2  INSERT INTO member
   (Full_Name, Gender, Date_of_Birth, Email, Password,
4   City_ID, Staff_ID, Is_Verified)
VALUES
6  ('Member Test', 'Man', '2006-01-01',
   'membertest@infiniread.test', 'dummyhash',
8   9901, @staff_id, 1);

10 SET @mem_id := LAST_INSERT_ID();

12 -- READ (verifikasi data member uji)
SELECT Mem_ID, Full_Name, Email, City_ID, Staff_ID,
      Is_Verified
14 FROM member
WHERE Mem_ID=@mem_id;

```

```

MariaDB [infiniread_xampptest]> -- CREATE
MariaDB [infiniread_xampptest]> INSERT INTO member
-> (Full_Name, Gender, Date_of_Birth, Email, Password,
-> City_ID, Staff_ID, Is_Verified)
-> VALUES
-> ('Member Test', 'Man', '2006-01-01',
-> 'membertest@infiniread.test', 'dummyhash',
-> 9901, @staff_id, 1);
Query OK, 1 row affected (0.006 sec)

MariaDB [infiniread_xampptest]>
MariaDB [infiniread_xampptest]> SET @mem_id := LAST_INSERT_ID();
Query OK, 0 rows affected (0.001 sec)

MariaDB [infiniread_xampptest]>
MariaDB [infiniread_xampptest]> -- READ (verifikasi data member uji)
MariaDB [infiniread_xampptest]> SELECT Mem_ID, Full_Name, Email, City_ID, Staff_ID, Is_Verified
-> FROM member
-> WHERE Mem_ID=@mem_id;
+-----+-----+-----+-----+-----+-----+
| Mem_ID | Full_Name | Email | City_ID | Staff_ID | Is_Verified |
+-----+-----+-----+-----+-----+-----+
| 1 | Member Test | membertest@infiniread.test | 9901 | 1 | 1 |
+-----+-----+-----+-----+-----+-----+
1 row in set (0.001 sec)

```

Gambar 3.4: Pengujian pada tabel member: create dan read.

3.4.5 Skenario Uji: Transaksi Borrowing

3.4.5.1 Create dan Update Status (Borrowing)

Query SQL:

```

1  -- CATATAN: Borrowing butuh Staff, Member, dan Book
   yang valid.

3  -- CREATE (peminjaman)
INSERT INTO borrowing
5  (Brw_Date, Due_Date, Return_Date, Brw_Status,
   Staff_ID, Mem_ID, Book_ID)
7  VALUES

```

```

('2025-12-22','2026-01-05',NULL,'BORROWED',
9  @staff_id,@mem_id,'TST001');

11 SET @brw_id := LAST_INSERT_ID();

13 -- READ (verifikasi transaksi)
SELECT Brw_ID, Brw_Status, Staff_ID, Mem_ID, Book_ID
15 FROM borrowing
WHERE Brw_ID=@brw_id;

17 -- UPDATE (pengembalian)
19 UPDATE borrowing
SET Return_Date='2025-12-22', Brw_Status='RETURNED'
21 WHERE Brw_ID=@brw_id;

23 -- READ (verifikasi update)
SELECT Brw_ID, Return_Date, Brw_Status
25 FROM borrowing
WHERE Brw_ID=@brw_id;

```

```

MariaDB [infiniread_xamptest]> -- CREATE (peminjaman)
MariaDB [infiniread_xamptest]> INSERT INTO borrowing
-> (Brw_Date, Due_Date, Return_Date, Brw_Status,
-> Staff_ID, Mem_ID, Book_ID)
-> VALUES
-> ('2025-12-22','2026-01-05',NULL,'BORROWED',
-> @staff_id,@mem_id,'TST001');
Query OK, 1 row affected (0.004 sec)

MariaDB [infiniread_xamptest]>
MariaDB [infiniread_xamptest]> SET @brw_id := LAST_INSERT_ID();
Query OK, 0 rows affected (0.000 sec)

MariaDB [infiniread_xamptest]>
MariaDB [infiniread_xamptest]> -- READ (verifikasi transaksi)
MariaDB [infiniread_xamptest]> SELECT Brw_ID, Brw_Status, Staff_ID, Mem_ID, Book_ID
-> FROM borrowing
-> WHERE Brw_ID=@brw_id;
+-----+-----+-----+-----+-----+
| Brw_ID | Brw_Status | Staff_ID | Mem_ID | Book_ID |
+-----+-----+-----+-----+-----+
| 2 | BORROWED | 1 | 1 | TST001 |
+-----+-----+-----+-----+-----+
1 row in set (0.000 sec)

MariaDB [infiniread_xamptest]>
MariaDB [infiniread_xamptest]> -- UPDATE (pengembalian)
MariaDB [infiniread_xamptest]> UPDATE borrowing
-> SET Return_Date='2025-12-22', Brw_Status='RETURNED'
-> WHERE Brw_ID=@brw_id;
Query OK, 1 row affected (0.002 sec)
Rows matched: 1 Changed: 1 Warnings: 0

MariaDB [infiniread_xamptest]>
MariaDB [infiniread_xamptest]> -- READ (verifikasi update)
MariaDB [infiniread_xamptest]> SELECT Brw_ID, Return_Date, Brw_Status
-> FROM borrowing
-> WHERE Brw_ID=@brw_id;
+-----+-----+-----+
| Brw_ID | Return_Date | Brw_Status |
+-----+-----+-----+
| 2 | 2025-12-22 | RETURNED |
+-----+-----+-----+
1 row in set (0.000 sec)

```

Gambar 3.5: Pengujian transaksi peminjaman: create dan update status pada borrowing.

3.5 Implementasi Website

3.5.1 Teknologi yang Digunakan

Website *InfiniRead* dikembangkan sebagai sistem perpustakaan digital berbasis web dengan memanfaatkan teknologi berikut:

- **Backend:** PHP digunakan sebagai bahasa pemrograman sisi server untuk mengelola logika Website, autentikasi pengguna, validasi data, serta interaksi dengan basis data.
- **Database:** MySQL digunakan sebagai sistem manajemen basis data relasional untuk menyimpan dan mengelola data buku, member, staff, peminjaman, serta data alamat (province dan city).
- **Frontend:** HTML digunakan untuk struktur halaman, CSS untuk tampilan antarmuka pengguna, dan JavaScript untuk interaksi dinamis seperti pemuatan data city berdasarkan province secara asinkron.
- **Server:** Website di-deploy menggunakan layanan hosting InfinityFree yang menyediakan dukungan PHP dan MySQL.
- **Version Control:** Git dan GitHub digunakan untuk mengelola versi kode sumber selama proses pengembangan.

3.5.2 Arsitektur

InfiniRead menerapkan arsitektur *client-server* dengan pemisahan yang jelas antara:

- **Frontend** sebagai antarmuka pengguna (member dan staff).
- **Backend** sebagai pengelola logika dan akses basis data.

Backend diorganisasikan ke dalam beberapa komponen utama, yaitu *controller*, *model*, dan *middleware*. Controller bertanggung jawab menangani permintaan dari frontend, model berfungsi untuk mengakses basis data, dan middleware digunakan untuk mengatur hak akses berdasarkan peran pengguna (member atau staff).

3.5.3 Modul Utama Website

Website InfiniRead memiliki beberapa modul utama sebagai berikut:

3.5.3.1 Modul Manajemen Buku

Modul ini digunakan untuk mengelola data buku yang tersedia di perpustakaan. Data buku disimpan pada tabel `book` dan mencakup informasi seperti judul, kategori, ISBN, jumlah halaman, tahun terbit, batas maksimal hari peminjaman, stok, serta `Staff_ID` yang menunjukkan staff pengelola buku tersebut.

Relasi penulis buku dikelola melalui tabel `book_author` untuk mendukung satu buku memiliki lebih dari satu penulis.

The screenshot displays the 'Kelola Buku' (Manage Books) interface. At the top, there's a search bar with the placeholder 'Cari judul, kategori, ID, atau author...' and a 'Search' button. Below this is the 'Tambah Buku Baru' (Add New Book) form, which includes fields for Book ID, Judul (Title), Kategori (Category), ISBN, Jumlah Halaman (Number of Pages), Tahun Terbit (Year Published), Age Rating (with a minimum value), Maksimal Hari Peminjaman (Maximum Borrowing Days), Stok (Stock), File URL (Link to book), and Penulis (Author). A 'Simpan Buku' (Save Book) button is at the bottom of the form. Below the form is a 'Daftar Buku' (Book List) table with columns for ID, Judul, Kategori, Stok, Penulis, and Aksi (Actions). The table contains two entries: one for 'Advanced Calculus' by Tom M. Apostol and another for 'Advanced Linear Models' by John A. Nelder and Peter McCullagh.

ID	Judul	Kategori	Stok	Penulis	Aksi
8025	Advanced Calculus	Academic	1	Tom M. Apostol	Edit Hapus
8082	Advanced Linear Models	Academic	0	John A. Nelder, Peter McCullagh	Edit Hapus

Gambar 3.6: Tampilan halaman modul manajemen buku pada website *InfiniRead*

3.5.3.2 Modul Peminjaman Buku

Modul peminjaman memungkinkan member melakukan peminjaman buku yang tersedia. Setiap transaksi peminjaman dicatat pada tabel `borrowing` yang menyimpan informasi tanggal pinjam, tanggal jatuh tempo, tanggal kembali, status peminjaman, serta relasi ke `member`, `staff`, dan `book`.

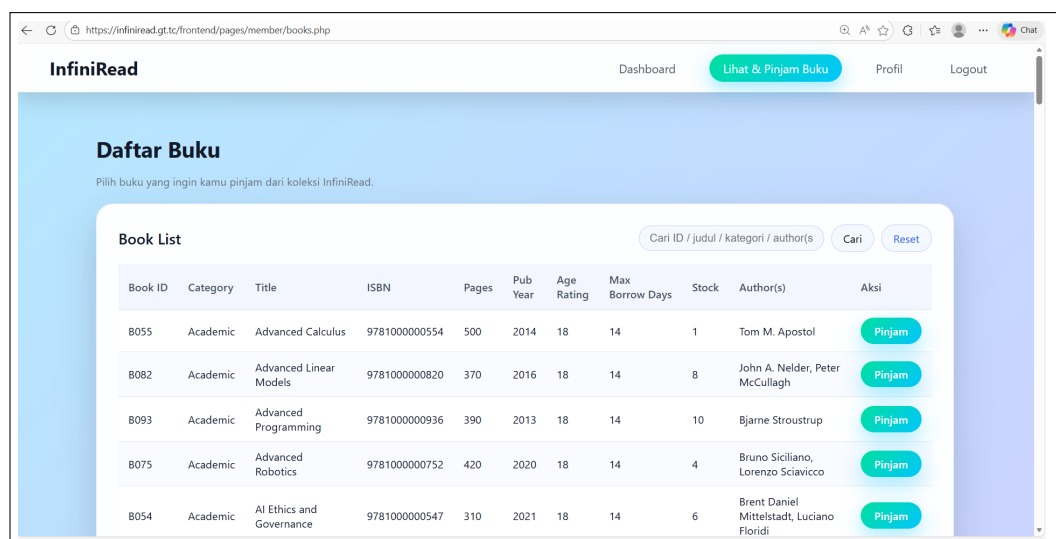
Saat member meminjam buku:

- Sistem otomatis mencatat tanggal pinjam sebagai tanggal saat transaksi dilakukan.

- Tanggal jatuh tempo dihitung berdasarkan batas maksimal hari peminjaman buku.
- Stok buku akan berkurang secara otomatis.

Saat buku dikembalikan:

- Sistem mencatat tanggal kembali secara otomatis.
- Status peminjaman diubah menjadi RETURNED.
- Stok buku akan bertambah kembali.

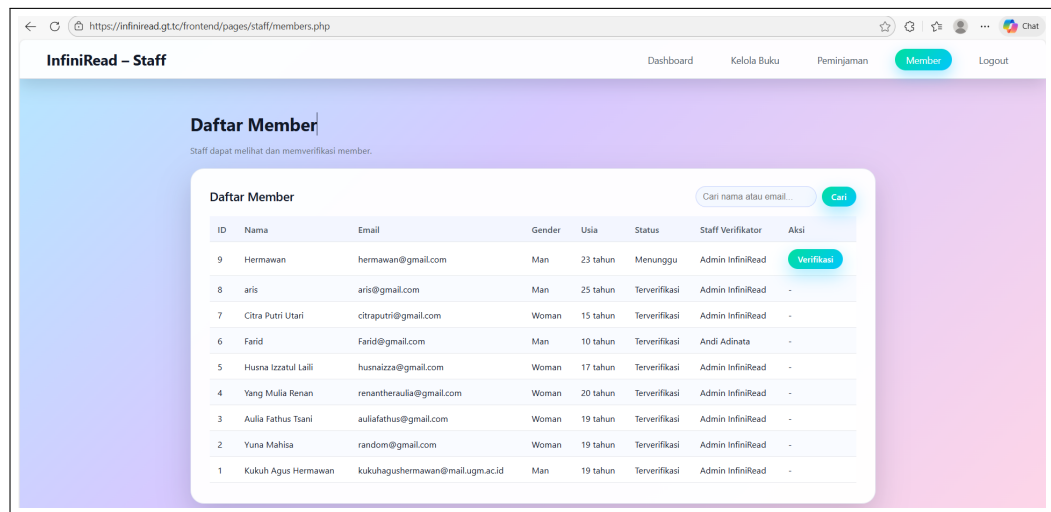


Gambar 3.7: Tampilan halaman modul peminjaman buku pada website *InfiniRead*

3.5.3.3 Modul Manajemen Member

Modul ini digunakan untuk mengelola data member. Data member disimpan pada tabel `member` dan mencakup informasi identitas, email, kata sandi terenkripsi, domisili (city dan province), serta `Staff_ID` sebagai penanda staff yang melakukan verifikasi member.

Staff memiliki akses untuk melihat daftar member dan melakukan verifikasi sesuai kebutuhan sistem.



Gambar 3.8: Tampilan halaman modul manajemen member pada website *InfiniRead*

3.5.3.4 Modul Manajemen Alamat

Data alamat disimpan pada tabel `province` dan `city`. Modul ini digunakan pada proses registrasi dan pembaruan profil member. Pemilihan city dilakukan secara dinamis berdasarkan province menggunakan JavaScript dan API backend.

3.5.4 Keamanan dan Autentikasi

Sistem autentikasi diterapkan menggunakan mekanisme sesi (*session*) pada PHP. Middleware digunakan untuk membatasi akses halaman berdasarkan peran pengguna:

- Member hanya dapat mengakses fitur peminjaman dan profil pribadi.
- Staff hanya dapat mengakses fitur pengelolaan buku, peminjaman, dan member.

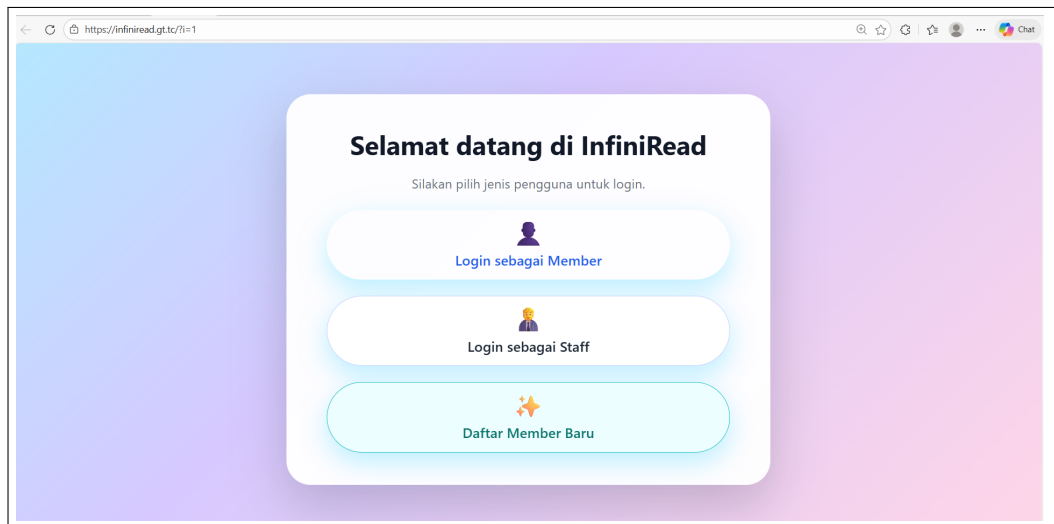
Kata sandi pengguna disimpan dalam bentuk terenkripsi untuk menjaga keamanan data.

3.5.5 Alur Login dan Hak Akses Pengguna

Website *InfiniRead* mendukung dua jenis pengguna, yaitu **Member** dan **Staff**, yang masing-masing memiliki hak akses dan fitur yang berbeda. Proses autentikasi dilakukan melalui halaman login utama yang dapat diakses melalui alamat:

`https://infiniread.gt.tc`

Pada halaman login, pengguna memasukkan *username* (email) dan *password*. Sistem kemudian melakukan validasi kredensial terhadap basis data dan menentukan peran pengguna berdasarkan data akun.



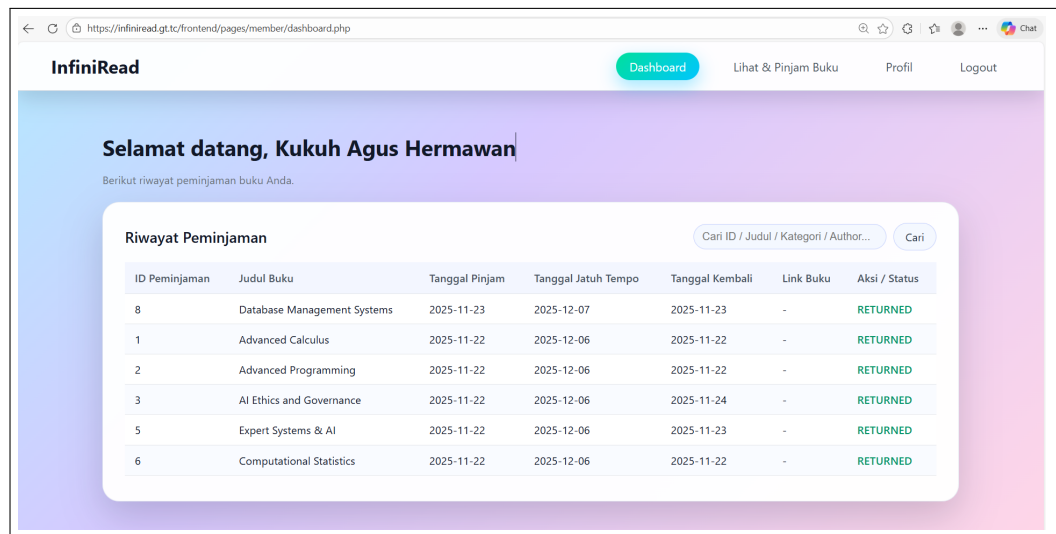
Gambar 3.9: Tampilan pilihan role login utama web *InfiniRead* untuk menentukan hak akses sebagai *Member* atau *Staff*

3.5.5.1 Login sebagai Member

Setelah berhasil login sebagai member, pengguna akan diarahkan ke halaman *Dashboard Member*. Pada halaman ini, member dapat:

- Melihat ringkasan aktivitas peminjaman buku.
- Mengakses daftar buku yang tersedia.
- Melakukan peminjaman buku.
- Melihat riwayat peminjaman dan status pengembalian.
- Mengelola dan memperbarui data profil pribadi.

Setiap peminjaman yang dilakukan oleh member akan otomatis tercatat pada tabel *borrowing* dengan relasi ke *Mem_ID*, *Book_ID*, dan *Staff_ID*.



Gambar 3.10: Tampilan halaman Dashboard Member pada website *InfiniRead*

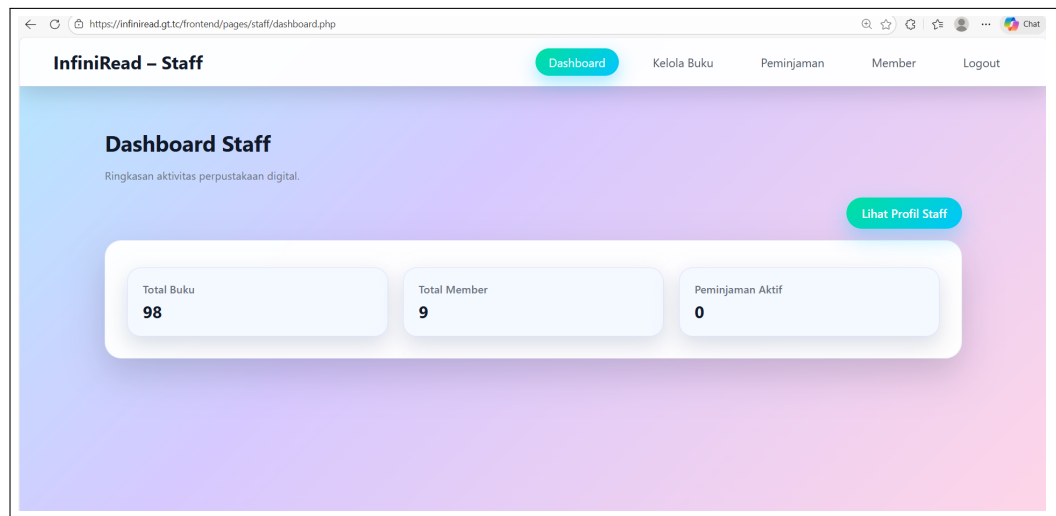
3.5.5.2 Login sebagai Staff

Apabila pengguna login sebagai staff, sistem akan mengarahkan ke halaman *Dashboard Staff*. Staff memiliki hak akses yang lebih luas dibandingkan member, yaitu untuk mengelola data sistem perpustakaan secara keseluruhan.

Fitur utama yang dapat diakses oleh staff meliputi:

- Mengelola data buku (tambah, ubah, dan hapus buku).
- Melihat dan mengelola seluruh data peminjaman.
- Melakukan konfirmasi atau pencatatan pengembalian buku.
- Melihat dan memverifikasi data member.

Setiap aktivitas pengelolaan oleh staff dicatat melalui atribut `Staff_ID` pada tabel terkait untuk menjaga konsistensi dan akuntabilitas data.



Gambar 3.11: Tampilan halaman Dashboard Staff pada website *InfiniRead*

3.5.5.3 Akun Uji (Demo Account)

Untuk keperluan pengujian dan evaluasi sistem, Website *InfiniRead* menyediakan akun uji (*demo account*) yang dapat digunakan oleh penguji atau dosen untuk mencoba seluruh fitur yang tersedia pada sistem. Akun ini disediakan khusus untuk kebutuhan akademik dan pengujian fungsionalitas Website.

Akun Member

Akun member digunakan untuk menguji fitur peminjaman buku, pengelolaan profil, serta riwayat peminjaman. Kredensial akun member uji adalah sebagai berikut:

- **Email:** cobamember@gmail.com
- **Password:** masuk123

Dengan akun ini, pengguna dapat masuk sebagai member dan mencoba seluruh fitur yang tersedia untuk pengguna member, termasuk melakukan peminjaman dan pengembalian buku secara otomatis.

Akun Staff

Akun staff digunakan untuk menguji fitur administrasi sistem perpustakaan. Kredensial akun staff uji adalah sebagai berikut:

- **Email:** admin@infiniread.test
- **Password:** masuk123

Melalui akun staff, penguji dapat mengakses halaman *Dashboard Staff* dan mencoba fitur pengelolaan buku, pengelolaan data peminjaman, serta verifikasi data member.

Catatan: Kredensial akun uji ini disediakan hanya untuk keperluan pengujian akademik dan tidak digunakan pada lingkungan produksi.

3.6 Pembahasan

3.6.1 Kesesuaian Sistem dengan Tujuan

Berdasarkan hasil implementasi dan pengujian, sistem *InfiniRead* telah memenuhi tujuan utama yang ditetapkan, yaitu menyediakan sistem perpustakaan digital yang mampu mengelola data buku, member, staff, serta proses peminjaman dan pengembalian buku secara terintegrasi. Seluruh relasi antar entitas pada basis data telah diimplementasikan sesuai dengan desain ERD yang dirancang sebelumnya.

3.6.2 Kelebihan Sistem

Sistem *InfiniRead* memiliki beberapa kelebihan, antara lain:

- Struktur basis data yang terorganisir dan terhubung dengan baik melalui relasi antar tabel.
- Pemisahan peran antara staff dan member sehingga hak akses lebih terkontrol.
- Proses peminjaman dan pengembalian buku dilakukan secara otomatis dan konsisten.
- Antarmuka pengguna yang sederhana dan mudah digunakan.

3.6.3 Keterbatasan Sistem

Meskipun sistem telah berjalan dengan baik, masih terdapat beberapa keterbatasan, antara lain:

- Sistem belum mendukung notifikasi keterlambatan pengembalian buku.

- Fitur laporan statistik peminjaman masih terbatas.
- Sistem belum dioptimalkan untuk penggunaan berskala besar atau akses bersamaan dalam jumlah tinggi.

Keterbatasan tersebut dapat dijadikan dasar pengembangan lebih lanjut pada versi sistem berikutnya.

BAB IV

Kesimpulan

4.1 Kesimpulan

Pengembangan sistem perpustakaan digital *InfiniRead* berhasil menghasilkan sebuah aplikasi berbasis web yang mampu mengelola data perpustakaan secara terstruktur. Sistem ini mendukung pengelolaan buku, anggota, staf, serta transaksi peminjaman dan pengembalian dengan basis data relasional yang dirancang sesuai prinsip normalisasi. Implementasi sistem telah berjalan sesuai dengan perancangan awal dan kebutuhan fungsional.

4.2 Saran Pengembangan

Untuk pengembangan selanjutnya, sistem dapat ditingkatkan dengan menambahkan fitur notifikasi keterlambatan, perhitungan denda otomatis, laporan peminjaman dalam bentuk grafik, serta peningkatan aspek keamanan sistem. Selain itu, pengembangan antarmuka pengguna yang lebih responsif untuk perangkat mobile juga dapat menjadi fokus pengembangan di masa mendatang.

DAFTAR PUSTAKA

- Ray, I. and Chakraborty, S. (2006). A framework for flexible access control in digital library systems. In *IFIP Annual Conference on Data and Applications Security*, pages 234–247, Cambridge, UK. Springer. IFIP/DBSec.
- Saputra, A. (2025). Policies, methods, and challenges in the developing digital collections in university libraries. *EDULIB Journal of Library and Information Science*, 15(1):95–106. EDULIB UPI.
- Taylor’s University Library (2024). Facilities – automated book returns. Technical report, Taylor’s University, Subang Jaya, Malaysia. Official library facility information.

LAMPIRAN A

Script SQL Basis Data InfiniRead

Lampiran ini berisi satu kesatuan script SQL yang digunakan untuk membangun basis data InfiniRead, mulai dari pembuatan database hingga seluruh tabel beserta relasi *foreign key*nya.

```
-- =====
2  -- CREATE DATABASE
-- =====
4  CREATE DATABASE infiniread;
   USE infiniread;
6
-- =====
8  -- TABLE: Province
-- =====
10 CREATE TABLE Province
   (
12   Province_ID INT NOT NULL,
     Province_Name VARCHAR(100) NOT NULL,
14   PRIMARY KEY (Province_ID)
   );
16
-- =====
18 -- TABLE: City
-- =====
20 CREATE TABLE City
   (
22   City_ID INT NOT NULL,
     Province_ID INT NOT NULL,
24   City_Name VARCHAR(100) NOT NULL,
     PRIMARY KEY (City_ID),
26   FOREIGN KEY (Province_ID) REFERENCES
       Province(Province_ID)
   );
28
-- =====
30 -- TABLE: Staff
-- =====
32 CREATE TABLE Staff
   (
```

```

34  Staff_ID INT NOT NULL AUTO_INCREMENT,
    Staff_Name VARCHAR(100) NOT NULL,
36  Email VARCHAR(100) NOT NULL,
    Password VARCHAR(255) NOT NULL,
38  PRIMARY KEY (Staff_ID)
);

40
-- =====
42 -- TABLE: Member
-- =====

44 CREATE TABLE Member
(
46  Mem_ID INT NOT NULL AUTO_INCREMENT,
    Full_Name VARCHAR(100) NOT NULL,
48  Gender ENUM('Man', 'Woman') NOT NULL,
    Date_of_Birth DATE NOT NULL,
50  Email VARCHAR(50) NOT NULL,
    Password VARCHAR(255) NOT NULL,
52  City_ID INT NOT NULL,
    Staff_ID INT NOT NULL,
54  Is_Verified TINYINT(1) NOT NULL DEFAULT 0,
    PRIMARY KEY (Mem_ID),
56  UNIQUE (Email),
    FOREIGN KEY (City_ID) REFERENCES City(City_ID),
58  FOREIGN KEY (Staff_ID) REFERENCES Staff(Staff_ID)
);

60
-- =====
62 -- TABLE: Book
-- =====

64 CREATE TABLE Book
(
66  Book_ID VARCHAR(20) NOT NULL,
    Category VARCHAR(20) NOT NULL,
68  Title VARCHAR(100) NOT NULL,
    ISBN VARCHAR(50) NOT NULL,
70  Pages INT NOT NULL,
    Pub_Year INT NOT NULL,
72  Age_Rating INT NOT NULL,
    Max_Borrowdays INT NOT NULL,
74  Stock INT NOT NULL,
    Staff_ID INT NOT NULL,

```

```

76     File_URL VARCHAR(255) NOT NULL,
      PRIMARY KEY (Book_ID),
78     FOREIGN KEY (Staff_ID) REFERENCES Staff(Staff_ID)
    );

80
-- =====
82 -- TABLE: Book_Author
-- =====
84 CREATE TABLE Book_Author
(
86     Author VARCHAR(100) NOT NULL,
      Book_ID VARCHAR(20) NOT NULL,
88     PRIMARY KEY (Author, Book_ID),
      FOREIGN KEY (Book_ID) REFERENCES Book(Book_ID)
90 );

92 -- =====
-- TABLE: Borrowing
-- =====
94
CREATE TABLE Borrowing
96 (
      Brw_ID INT NOT NULL AUTO_INCREMENT,
98      Brw_Date DATE NOT NULL,
      Due_Date DATE NOT NULL,
100      Return_Date DATE DEFAULT NULL,
      Brw_Status ENUM('BORROWED', 'RETURNED', 'OVERDUE')
102      NOT NULL DEFAULT 'BORROWED',
      Staff_ID INT NOT NULL,
104      Mem_ID INT NOT NULL,
      Book_ID VARCHAR(20) NOT NULL,
106      PRIMARY KEY (Brw_ID),
      FOREIGN KEY (Staff_ID) REFERENCES Staff(Staff_ID),
108      FOREIGN KEY (Mem_ID) REFERENCES Member(Mem_ID),
      FOREIGN KEY (Book_ID) REFERENCES Book(Book_ID)
110 );

```


LAMPIRAN B

Dokumentasi Antarmuka dan Fitur Website InfiniRead

Fitur untuk Member

(a) Halaman Login Member

(b) Halaman Registrasi Member

ID Peminjaman	Judul Buku	Tanggal Pinjam	Tanggal Jatuh Tempo	Tanggal Kembali	Link Buku	Aksi / Status
8	Database Management Systems	2025-11-23	2025-12-07	2025-11-23	-	RETURNED
1	Advanced Calculus	2025-11-22	2025-12-06	2025-11-22	-	RETURNED
2	Advanced Programming	2025-11-22	2025-12-06	2025-11-22	-	RETURNED
3	AI Ethics and Governance	2025-11-22	2025-12-06	2025-11-24	-	RETURNED
5	Expert Systems & AI	2025-11-22	2025-12-06	2025-11-23	-	RETURNED
6	Computational Statistics	2025-11-22	2025-12-06	2025-11-22	-	RETURNED

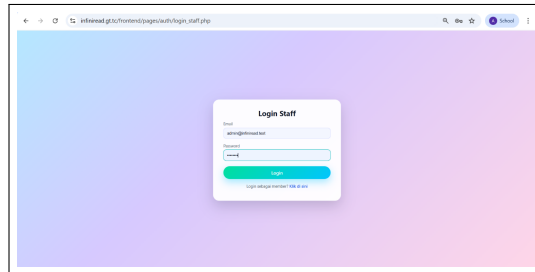
(c) Dashboard Member

Book ID	Category	Title	ISBN	Pages	Pub Year	Age Rating	Max Borrow Days	Stock	Author(s)	Aksi
B005	Academic	Advanced Calculus	9781000000054	500	2014	18	14	1	Tom M. Apostol	Pinjam
B002	Academic	Advanced Linear Models	9781000000020	370	2016	18	14	8	John A. Nelder, Peter McCullagh	Pinjam
B003	Academic	Advanced Programming	9781000000036	390	2013	18	14	10	Stuart Seidenberg	Pinjam
B005	Academic	Advanced Robotics	9781000000072	430	2020	18	14	4	Bruce Schlemmer, Lorenzo Sciavicco	Pinjam
B004	Academic	AI Ethics and Governance	9781000000047	310	2021	18	14	6	Brent Daniel Mitchell, Luciano Floridi	Pinjam

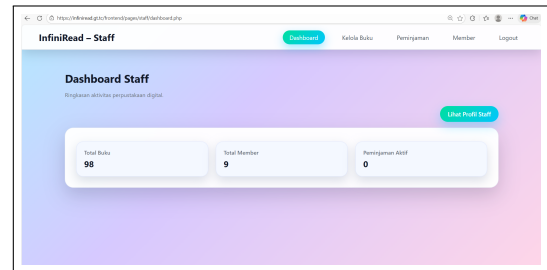
(d) Daftar Buku untuk Member

(e) Profil Member

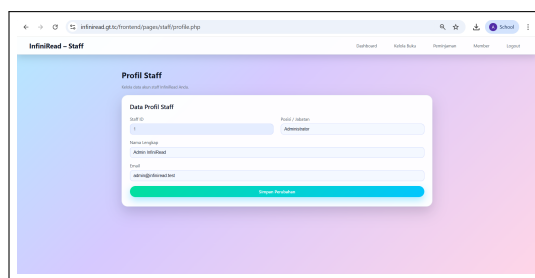
Fitur untuk Staff



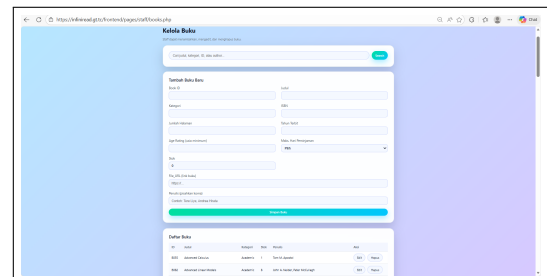
(a) Halaman Login Staff



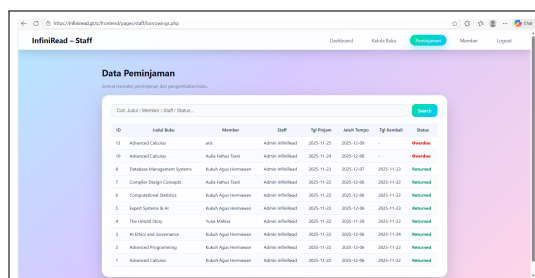
(b) Dashboard Staff



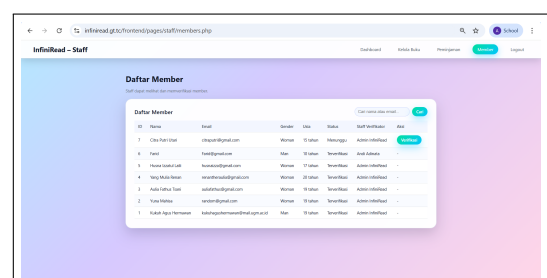
(c) Profil Staff



(d) Kelola Data Buku



(e) Kelola Data Peminjaman



(f) Kelola dan Verifikasi Data Member