

Laporan Akhir Proyek Basis Data Library Management System



Disusun Oleh Kelompok 1
Davi Ezra Syandana (24/538363/PA/22841)
Andra Kusnaedi Ilyaz (24/537757/PA/22793)

Bab I

Pendahuluan

1.1 Latar Belakang Masalah

Banyak perpustakaan berskala kecil, seperti perpustakaan sekolah, departemen, maupun perpustakaan kota daerah tertinggal, masih mengandalkan proses manual dalam pengelolaan koleksi buku dan kegiatan peminjaman. Proses manual tersebut tidak efisien, rentan terhadap hilangnya data, serta menyulitkan proses pelacakan buku yang terlambat dikembalikan. Selain itu, anggota perpustakaan tidak memiliki cara yang cepat untuk mengetahui ketersediaan suatu buku tanpa harus datang dan mencari secara langsung di lokasi.

Proyek ini bertujuan untuk mengatasi permasalahan tersebut melalui perancangan sistem basis data serta implementasi aplikasinya dalam bentuk web. Sistem Library Borrowing Tracker ini dirancang untuk mendigitalisasi proses inti perpustakaan, meliputi pencatatan data buku, data anggota, serta riwayat transaksi peminjaman. Dengan adanya sistem ini proses pengelolaan perpustakaan diharapkan menjadi lebih akurat, efisien, transparan, dan aman, sehingga dapat meningkatkan kualitas layanan serta kemudahan akses bagi seluruh pengguna.

1.2 System Users

Sistem ini akan memiliki dua jenis pengguna utama dengan hak akses yang berbeda:

1. Staff perpustakaan (Administrator) Memiliki hak akses penuh untuk mengelola seluruh data dalam system.

Tugas:

- Mengelola data buku dan anggota
- Mencatat transaksi peminjaman dan pengembalian buku.

2. Anggota Perpustakaan (pengguna) Memiliki hak akses terbatas.

Fitur:

- Melakukan pencarian buku untuk memeriksa ketersediaan.
- Melakukan peminjaman buku.
- Melihat Riwayat peminjaman.

1.3 System Objectives

Sesuai dengan tujuan pembelajaran proyek, system ini akan dikembangkan untuk mencapai target berikut:

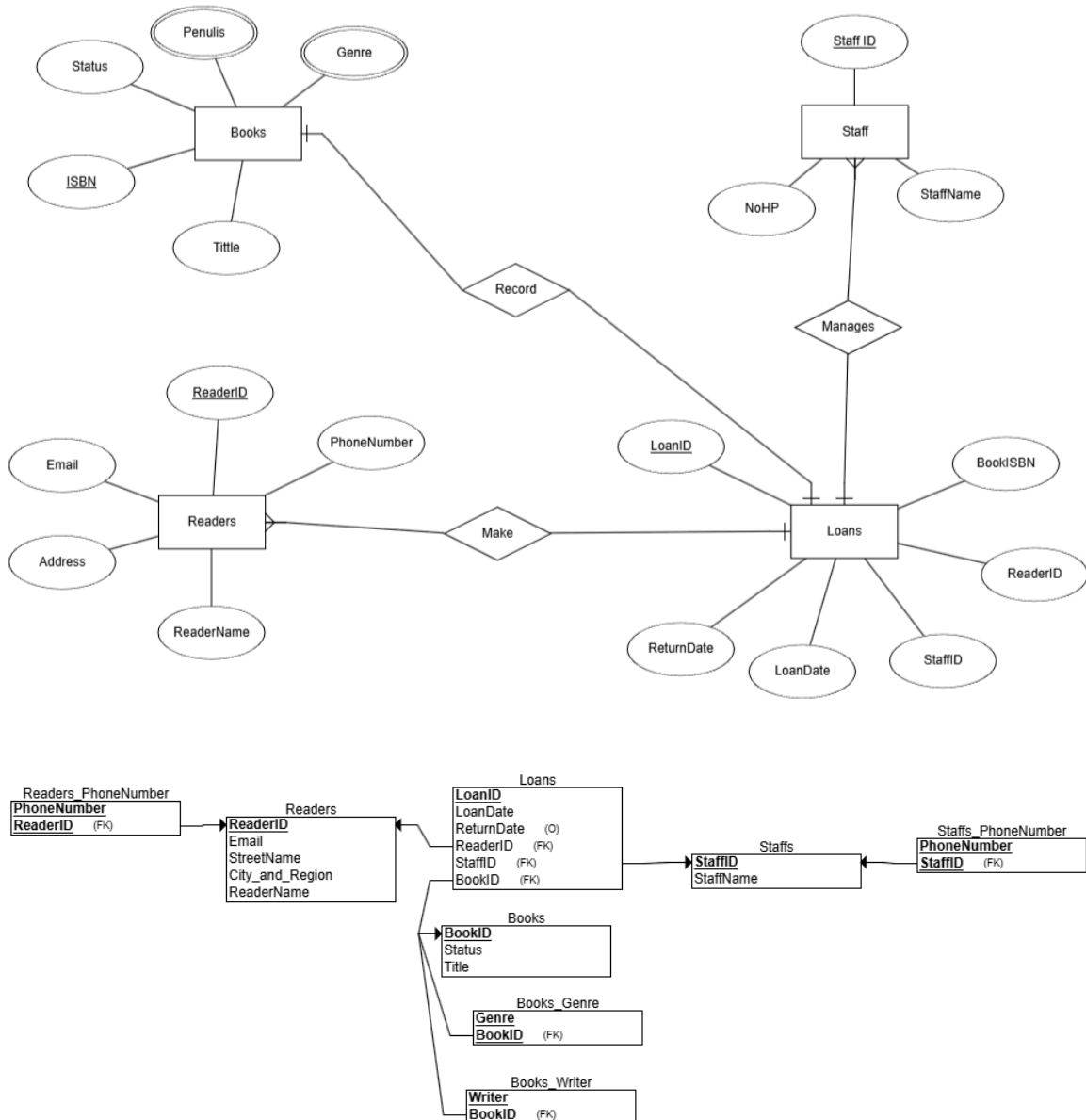
1. Merancang Model data Konseptual : Menganalisis kebutuhan data perpustakaan dan merancangnya ke dalam sebuah Entity-relationship Diagram (ERD).

2. Mengimplementasikan Basis Data Relasional: Mendesain skema database relasional berdasarkan ERD yang sudah dibuat, dan membangunnya menggunakan SQL.
3. Mengembangkan Fungsionalitas: Membangun sebuah aplikasi sederhana dalam bentuk web yang terhubung ke basis data dan memiliki fungsi-fungsi sebagai berikut:
 - Operasi CRUD lengkap.
 - Fitur pencarian.
 - Fitur menajukan pinjaman.

Bab II

Perancangan dan Implementasi Database

2.1 Visualisasi ERD dan Skema Relasional Logis (LRS)



Gambar di atas merepresentasikan Entity Relationship Diagram (ERD) dan Logical Relational Schema (LRS) yang berfungsi sebagai cetak biru arsitektur penyimpanan data untuk sistem perpustakaan ini. Transformasi dari desain konseptual (ERD) ke skema logis (LRS/Schema) dilakukan untuk memastikan data terstruktur dengan efisien dan siap diimplementasikan ke dalam *Database Management System* (DBMS).

Penjelasan utama dari transformasi skema ini adalah:

- Tabel Induk (Master): Tabel Readers, Staff, dan Books bertindak sebagai entitas utama yang menyimpan data referensi. Tabel-tabel ini berdiri secara independen dan *Primary Key* (PK) mereka menjadi acuan bagi tabel lain.
- Penanganan Atribut Multivalued (1NF): Berbeda dengan desain dasar, skema ini menerapkan normalisasi tingkat pertama secara ketat. Atribut yang bernilai ganda pada ERD (ditandai dengan garis oval ganda) seperti *PhoneNumber* pada *Readers*, *NoHP* pada *Staff*, serta *Genre* dan *Penulis* pada *Books*, telah dipisahkan menjadi tabel anak tersendiri (*Readers_PhoneNumber*, *Books_Genre*, dsb). Hal ini dilakukan untuk menghindari atomisitas data yang buruk.
- Tabel Transaksi (Loans): Tabel Loans berfungsi sebagai pusat aktivitas (transaksi) peminjaman. Tabel ini menghubungkan tiga entitas sekaligus: Buku (*Books*), Peminjam (*Readers*), dan Petugas (*Staff*) melalui mekanisme *Foreign Key* (FK).

2.2 Analisis Sintaks SQL per Tabel

Berdasarkan *Relation Schema* yang terbentuk, urutan pembuatan tabel dan analisis strukturnya disusun untuk menjaga *Referential Integrity*. Berikut adalah analisis mendalam untuk setiap kelompok tabel:

A. Tabel Induk (Master Data)

Tabel-tabel ini harus dibuat terlebih dahulu karena tidak memiliki ketergantungan *Foreign Key* ke tabel utama lain.

- Tabel Readers
Tabel ini menyimpan data fundamental anggota perpustakaan.
 - ReaderID (PK): Kolom identifikasi unik untuk setiap anggota.
 - Email: Atribut ini sebaiknya diberi batasan UNIQUE untuk mencegah pendaftaran ganda dengan email yang sama, mirip dengan logika bisnis pada sistem pendaftaran pengguna umumnya.
 - StreetName dan City_and_Region: Merupakan hasil dekomposisi dari atribut "Address" di ERD untuk memenuhi kaidah atomik data, sehingga pencarian berdasarkan kota atau wilayah menjadi lebih mudah.
- Tabel Books
Menyimpan inventaris buku.
 - BookID (PK): Menggantikan ISBN sebagai *system-generated ID* untuk efisiensi indeksasi database.
 - Status: Kolom penting untuk logika bisnis peminjaman, yang menentukan apakah buku sedang "Tersedia" atau "Dipinjam".
- Tabel Staff
Menyimpan data petugas perpustakaan yang berwenang mengelola peminjaman.

B. Tabel Penanganan Atribut Bernilai Ganda (Multivalued Attribute)

Kelompok tabel ini adalah implementasi teknis untuk menangani atribut yang memiliki lebih dari satu nilai (simbol oval ganda pada ERD).

- Tabel Books_Genre & Books_Writer
 - Karena satu buku bisa memiliki banyak genre (misal: Fiksi, Misteri) dan ditulis oleh lebih dari satu penulis, data ini dipisah.
 - Menggunakan *Composite Primary Key* atau kombinasi FK BookID dengan atribut nilainya (Genre/Writer) untuk memastikan relasi yang valid.
 - Menggunakan aturan ON DELETE CASCADE pada BookID, sehingga jika sebuah buku dihapus dari katalog, data genre dan penulisnya akan otomatis terhapus untuk menjaga kebersihan data³.
- Tabel Readers_PhoneNumber & Staffs_PhoneNumber
 - Memungkinkan satu orang (pembaca atau staf) memiliki lebih dari satu nomor telepon tanpa melanggar aturan Normalisasi Bentuk Pertama (1NF).

C. Tabel Transaksi (Loans)

Ini adalah tabel inti yang mencatat peristiwa peminjaman.

- LoanID (PK): Identitas unik setiap transaksi peminjaman.
- LoanDate & ReturnDate: Menyimpan durasi peminjaman. ReturnDate dapat bernilai NULL (nullable) saat peminjaman baru dibuat, dan diisi saat buku dikembalikan (logika serupa dengan waktu selesai acara yang belum ditentukan)⁴.
- Foreign Keys (Relasi):
 - ReaderID (FK): Menunjuk siapa yang meminjam.
 - StaffID (FK): Menunjuk siapa petugas yang melayani (Manages).
 - BookID (FK): Menunjuk buku apa yang dipinjam.Keberadaan tiga Foreign Key ini menegaskan relasi yang tergambar pada ERD di mana entitas Record, Make, dan Manages bermuara pada satu tabel Loans.

2.3 Verifikasi Normalisasi (3NF)

Desain skema database ini telah mematuhi aturan normalisasi hingga tingkat ketiga (3NF):

1. 1NF (First Normal Form): Terpenuhi. Semua atribut bernilai ganda pada desain konseptual (NoHP, Genre, Penulis) telah diekstraksi keluar dari tabel utama menjadi tabel terpisah (Books_Genre, Readers_PhoneNumber, dll). Tidak ada sel yang memuat banyak nilai.
2. 2NF (Second Normal Form): Terpenuhi. Tabel-tabel hasil pecahan (seperti Books_Writer) memiliki ketergantungan fungsional penuh terhadap kunci utamanya (kombinasi BookID dan Writer).

3. 3NF (Third Normal Form): Terpenuhi. Tidak ada ketergantungan transitif pada tabel utama. Atribut non-kunci (seperti StreetName pada Readers) bergantung langsung pada ReaderID, bukan pada atribut non-kunci lainnya. Data yang berulang (seperti detail buku) hanya direferensikan melalui BookID di tabel Loans, bukan ditulis ulang, sehingga mencegah redundansi data⁵.

Bab III

Implementasi CRUD

3.1 Arsitektur Aplikasi (Native PHP)

Berbeda dengan penggunaan kerangka kerja (*framework*) yang kompleks, pengembangan sistem backend ini mengadopsi pendekatan Native PHP dengan gaya pemrograman prosedural. Pendekatan ini dipilih untuk memberikan kontrol penuh terhadap alur eksekusi kode dan pemahaman mendalam mengenai siklus hidup permintaan HTTP (*Request Lifecycle*) tanpa lapisan abstraksi yang berlebihan.

Meskipun tidak menggunakan konsep *Model-View-Controller* (MVC) secara ketat, arsitektur sistem tetap dirancang secara modular untuk menjaga keterbacaan dan kemudahan pemeliharaan (*maintainability*). Sistem dibagi menjadi file-file fungsional terpisah berdasarkan tanggung jawabnya:

1. Konfigurasi Database (`connection.php`): Bertanggung jawab semata-mata untuk inisialisasi hubungan ke server database.
2. Logika Bisnis & Akses Data (`crud.php`): Bertindak sebagai "mesin" utama yang memuat kumpulan fungsi (*function library*) untuk memanipulasi data dan menangani logika routing sederhana.
3. Antarmuka Pengguna (File View/HTML): Bertindak sebagai lapisan yang dilihat oleh pengguna, memanggil & menggunakan fungsi-fungsi dari lapisan logika.

3.2 Implementasi Koneksi Database

Komunikasi antara aplikasi PHP dan basis data MySQL dikelola dalam file `connection.php`. Kami menggunakan ekstensi `mysqli` (MySQL Improved) yang merupakan standar modern untuk interaksi database prosedural di PHP.

```
<?php
//ini untuk koneksi antara database dan server web

$host = "localhost"; //nama host
$user = "root"; //nama user database
$pass = ""; //password database
$db = "libraryborrowingtracker"; //nama database
$conn = mysqli_connect($host, $user, $pass, $db); //fungsi untuk koneksi ke database

mysqli_select_db($conn, $db); //memilih database yang akan dipakai

?>
```

- Definisi Kredensial: Variabel `$host`, `$user`, `$pass`, dan `$db` didefinisikan secara eksplisit untuk memudahkan perubahan konfigurasi server di masa depan.
- Inisialisasi Koneksi: Fungsi `mysqli_connect()` digunakan untuk membuka jalur komunikasi ke server.

- Seleksi Database: `mysqli_select_db()` memastikan bahwa operasi query selanjutnya akan dieksekusi secara spesifik pada database `libraryborrowingtracker`.

3.3 Implementasi Logika CRUD (Create, Read, Update, Delete)

Inti dari fungsionalitas sistem terletak pada file `crud.php`. File ini tidak hanya berisi instruksi SQL, tetapi juga membungkus setiap operasi database ke dalam fungsi PHP (*User-Defined Functions*). Pendekatan ini memungkinkan kode digunakan kembali (*reusable*) di berbagai halaman tanpa perlu menulis ulang *query*.

A. Fungsi Penambahan Data (Create)

```
//fungsi untuk menambahkan data baru ke dalam database
function addBook($conn, $data) {
    $bookID = $data['bookID'];
    $tittle = $data['tittle'];
    $author = $data['author'];
    $borrowStatus = $data['borrowStatus'];

    $query = "INSERT INTO books (bookID, tittle, author, borrowStatus) VALUES ('$bookID', '$tittle', '$author', '$borrowStatus')";
    return mysqli_query($conn, $query);
}
```

Fungsi `addBook($conn, $data)` dirancang untuk menangani operasi penyisipan data buku baru.

- Ekstraksi Data: Data yang dikirim dari formulir (melalui variabel `$data`) dipecah menjadi variabel-variabel lokal (`$bookID`, `$tittle`, dst).
- Eksekusi Query: Perintah SQL `INSERT INTO books` disusun dan dijalankan menggunakan `mysqli_query`. Fungsi ini mengembalikan nilai boolean (`True/False`) yang menandakan keberhasilan operasi.

B. Fungsi Pembacaan Data (Read & Search)

```
//fungsi untuk menampilkan semua data dari database
function showBooks($conn) {
    $query = "SELECT * FROM books";
    $result = mysqli_query($conn, $query);
    $books = [];
    //ini untuk membaca loop, nantinya di html akan ada loop
    while ($row = mysqli_fetch_assoc($result)) {
        $books[] = $row;
    }
    return $books;
}
```

Sistem memiliki dua mekanisme pembacaan data:

1. Tampil Semua (`showBooks`): Menggunakan query `SELECT * FROM books` untuk mengambil seluruh katalog. Hasil query (*result set*) kemudian diiterasi menggunakan `while` dan `mysqli_fetch_assoc` untuk mengubah baris data database menjadi *associative array* PHP yang siap ditampilkan di tabel HTML.

2. Pencarian (cariBuku): Mengimplementasikan logika LIKE pada SQL (tittle LIKE ... OR author LIKE ...). Ini memungkinkan pengguna menemukan buku meskipun hanya memasukkan potongan kata kunci judul atau nama penulis.

C. Fungsi Pembaruan dan Penghapusan (Update & Delete)

```
//fungsi untuk mengupdate data berdasarkan bookID
function updateBook($conn, $data) {
    $bookID = $data['bookID'];
    $tittle = $data['tittle'];
    $author = $data['author'];
    $borrowStatus = $data['borrowStatus'];

    $query = "UPDATE books SET tittle='$tittle', author='$author',
    return mysqli_query($conn, $query);
}

//fungsi untuk menghapus data berdasarkan bookID
function deleteBook($conn, $bookID) {
    $query = "DELETE FROM books WHERE bookID='$bookID'";
    return mysqli_query($conn, $query);
}
```

- Update (updateBook): Fungsi ini menjalankan perintah UPDATE books SET ... WHERE bookID=.... Hal ini memungkinkan perubahan atribut buku (seperti judul atau status peminjaman) berdasarkan kunci primer bookID.
- Delete (deleteBook): Fungsi ini menghapus baris data secara permanen menggunakan perintah DELETE FROM yang ditargetkan spesifik pada bookID tertentu.

3.4 Logika Kontrol dan Routing Sederhana

Salah satu fitur krusial dalam crud.php adalah mekanisme penanganan formulir (*Form Handling*) yang terletak di bagian bawah file. Menggunakan logika kondisional `if(isset($_POST['action']))`, sistem dapat membedakan intensi pengguna

```

//logika control
if(isset($_POST['action'])) {

    // --- LOGIKA TAMBAH ---
    if ($_POST['action'] == "add") {
        if(addBook($conn, $_POST)) {
            header("location: adminpage.php");
            exit;
        } else {
            echo "Gagal menambah data.";
        }
    }

    // --- LOGIKA UPDATE ---
    } else if ($_POST['action'] == "update") {
        if(updateBook($conn, $_POST)) {
            header("location: adminpage.php");
            exit;
        } else {
            echo "Gagal mengupdate data.";
        }
    }
}
}

```

Pendekatan ini bertindak sebagai *Router* sederhana. Jika operasi berhasil, fungsi `header("location: adminpage.php")` akan secara otomatis mengarahkan pengguna kembali ke halaman admin, memberikan pengalaman pengguna (*User Experience*) yang mulus tanpa membiarkan pengguna "tersangkut" di halaman pemrosesan data. Terdapat juga logika lainnya yang terdapat pada file front end

Bab IV

Implementasi Front End

Sisi antarmuka (Frontend) dirancang untuk memberikan pengalaman visual yang intuitif sekaligus fungsional. Kami memadukan logika PHP *Native* dengan desain CSS kustom untuk menciptakan alur navigasi yang responsif. Fokus utama pada bab ini adalah implementasi mekanisme *Role-Based Access Control* (RBAC) yang memisahkan tampilan antara Pengguna Biasa (*User*) dan Administrator.

4.1 Implementasi Autentikasi (Login System)

Halaman Login berfungsi sebagai gerbang utama yang memverifikasi identitas pengguna sebelum memberikan akses ke dalam sistem. Berbeda dengan formulir statis, halaman ini mengimplementasikan logika keamanan berlapis.

A. Antarmuka Login

Seperti terlihat pada gambar implementasi, desain halaman login mengusung konsep minimalis modern dengan ornamen visual abstrak ("blobs") untuk memberikan kesan estetis yang tidak kaku. Formulir meminta input username dan password yang divalidasi secara *real-time* saat tombol "LOGIN" ditekan.

B. Logika "Role-Based Routing"

```
// 4. Redirect sesuai Role
if ($row['role'] == 'admin') {
    header("Location: adminpage/adminpage.php");
} else {
    header("Location: userpage/userpage.php"); // User biasa ke landing page
}
exit;
```

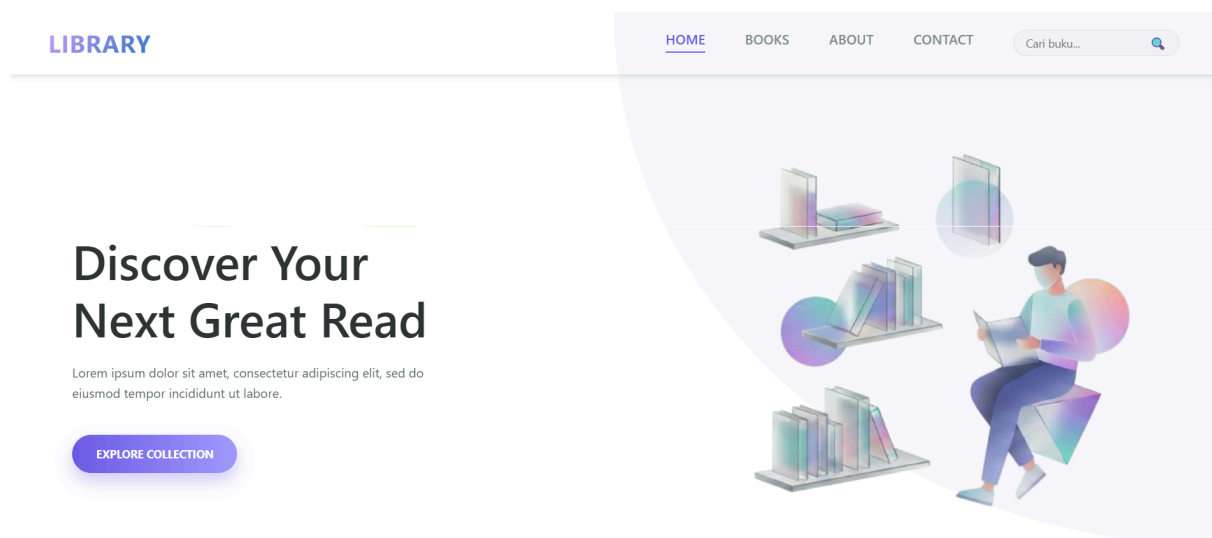
Fitur unggulan pada sistem autentikasi ini adalah kemampuan untuk mengarahkan pengguna ke halaman yang berbeda sesuai dengan hak aksesnya. Berdasarkan kode login.php, logika routing bekerja sebagai berikut:

1. Verifikasi Kredensial: Sistem menggunakan *Prepared Statement* (mysqli_prepare) untuk memeriksa kecocokan username dan password di database. Penggunaan metode ini sangat krusial untuk mencegah serangan *SQL Injection*.
2. Manajemen Sesi: Jika login berhasil, sistem menyimpan variabel \$_SESSION['role'] yang berisi status pengguna (apakah 'admin' atau 'user').
3. Pengarahan Kondisional (Redirection):
Sistem mengeksekusi logika percabangan untuk menentukan tujuan navigasi:
 - a. Admin: Diarahkan ke adminpage/adminpage.php untuk mengakses fitur pengelolaan data.
 - b. User: Diarahkan ke userpage/userpage.php untuk mengakses layanan peminjaman.

4.2 Implementasi Mode Pengguna (User Mode)

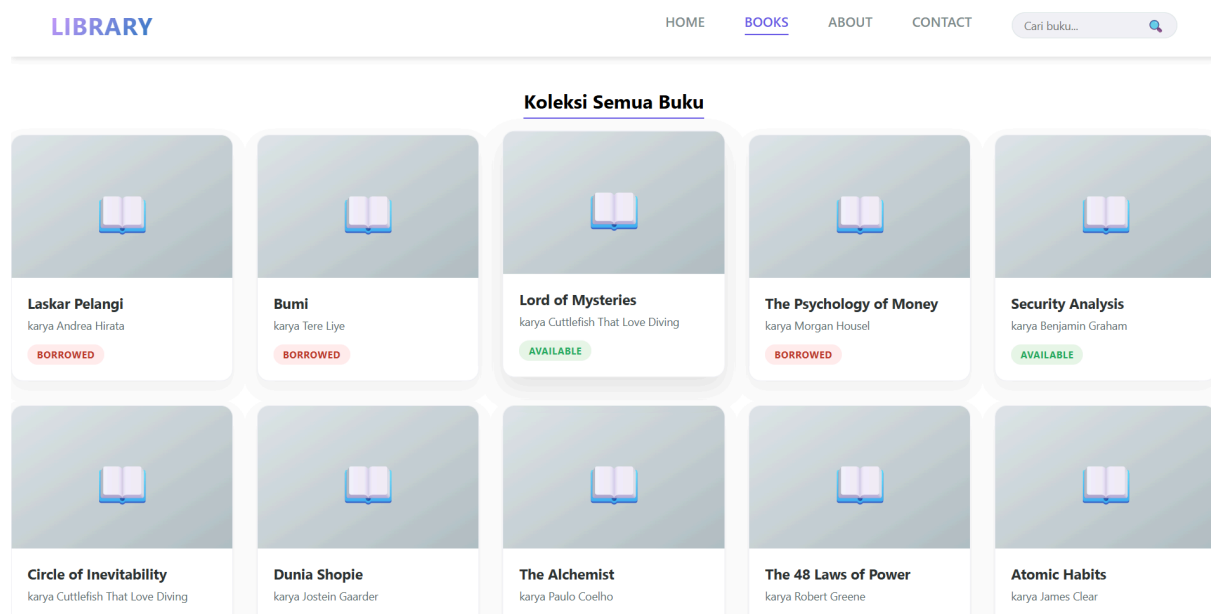
Mode ini dirancang dengan pendekatan *User-Centric*, fokus pada kemudahan eksplorasi dan pencarian buku bagi anggota perpustakaan.

A. Landing Page & Visualisasi



Saat pengguna berhasil masuk, mereka disambut dengan halaman muka (*Landing Page*) yang menampilkan ilustrasi visual dan *Call-to-Action* (CTA) "Explore Collection". Tampilan ini bertujuan untuk meningkatkan keterlibatan pengguna (*User Engagement*) sejak interaksi pertama.

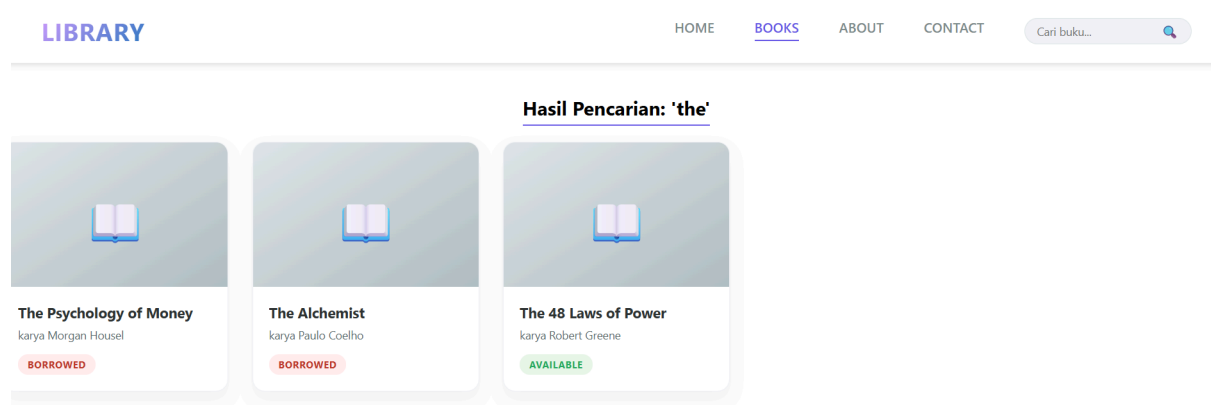
B. Katalog Buku Grid & Indikator Status



Berbeda dengan tampilan admin yang berbentuk tabel, katalog pengguna disajikan dalam format Grid Card. Setiap kartu buku menampilkan informasi esensial (Judul dan Penulis) serta indikator visual berupa "Badge Status":

- Badge Hijau (AVAILABLE): Menandakan buku tersedia dan dapat dipinjam.
- Badge Merah (BORROWED): Memberikan informasi instan bahwa buku sedang tidak tersedia, sehingga pengguna tidak perlu membuang waktu untuk mencoba meminjamnya.

C. Fitur Pencarian (Search)



Di bagian *header* navigasi, terdapat bilah pencarian (*Search Bar*) yang terintegrasi dengan fungsi cariBuku() di backend. Fitur ini memungkinkan pengguna memfilter ratusan koleksi buku hanya dengan kata kunci judul atau nama penulis secara instan.

4.3 Implementasi Mode Administrator (Admin Mode)

Mode administrator difokuskan pada efisiensi pengelolaan data operasional perpustakaan.

A. Dashboard CRUD

SELAMAT DATANG DI ADMIN PAGE

Disini kamu bisa melakukan add update dan delete buku dalam database

DAFTAR BUKU:

Add Book

ID	TITLE	AUTHOR	BORROWSTATUS	ACTION	
999-999-0001	Laskar Pelangi	Andrea Hirata	borrowed	Update	Delete
999-999-0002	Bumi	Tere Liye	borrowed	Update	Delete
999-999-0005	Lord of Mysteries	Cuttlefish That Love Diving	available	Update	Delete
999-999-0003	The Psychology of Money	Morgan Housel	borrowed	Update	Delete
999-999-0004	Security Analysis	Benjamin Graham	available	Update	Delete
999-999-0006	Circle of Inevitability	Cuttlefish That Love Diving	available	Update	Delete
999-999-0007	Dunia Shopie	Jostein Gaarder	available	Update	Delete

Halaman admin menyajikan data dalam format Tabel Relasional yang padat informasi. Desain ini dipilih untuk memudahkan administrator memindai data buku dalam jumlah besar (ID, Judul, Penulis, Status) dalam satu pandangan.

B. Kontrol Aksi (Action Controls)

Pada setiap baris data, terdapat tombol aksi yang terhubung langsung dengan fungsi logika di crud.php:

- Tombol Add Book: Terletak strategis di sudut kanan atas untuk menambah koleksi baru.

Tambah Data Buku

Kembali

ID Buku

Contoh: 999-999-0000

Judul Buku

Masukkan judul buku...

Penulis

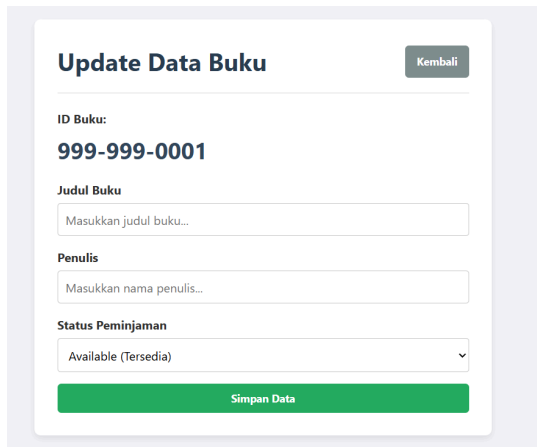
Masukkan nama penulis...

Status Peminjaman

Available (Tersedia)

Simpan Data

- Tombol Update (Kuning): Mengarahkan admin ke formulir penyuntingan data buku.



The screenshot shows a web form titled "Update Data Buku". In the top right corner, there is a yellow button labeled "Kembali". The form contains the following fields: "ID Buku:" with the value "999-999-0001"; "Judul Buku" with a text input field containing the placeholder "Masukkan judul buku..."; "Penulis" with a text input field containing the placeholder "Masukkan nama penulis..."; and "Status Peminjaman" with a dropdown menu currently showing "Available (Tersedia)". At the bottom of the form is a large green button labeled "Simpan Data".

- Tombol Delete (Merah): Berfungsi untuk menghapus data buku dari database. Pemisahan warna tombol (Kuning vs Merah) diterapkan untuk mencegah kesalahan klik (*human error*) pada operasi yang bersifat destruktif.

Bab V

Kesimpulan dan Refleksi

5.1 Kesimpulan

Pengembangan sistem informasi perpustakaan berbasis web ini telah berhasil mengintegrasikan desain basis data relasional dengan logika pemrograman prosedural PHP. Melalui tahapan perancangan dari Bab 2 hingga implementasi di Bab 4, dapat disimpulkan beberapa poin kunci:

1. Transisi Desain ke Implementasi: Skema basis data yang dirancang kompleks dengan normalisasi tingkat lanjut (pemisahan tabel *Books_Genre*, *Books_Writer*) berhasil diimplementasikan secara fisik menggunakan MySQL. Struktur ini terbukti mampu menjaga integritas data dan meminimalisir redundansi saat aplikasi dijalankan.
2. Efektivitas Arsitektur Native: Tanpa menggunakan *framework*, sistem berhasil menerapkan prinsip *Separation of Concerns* sederhana dengan memisahkan konfigurasi koneksi (*connection.php*), logika bisnis (*crud.php*), dan antarmuka pengguna. Hal ini membuktikan bahwa pemahaman mendalam tentang siklus hidup *request* PHP adalah fondasi vital sebelum melangkah ke penggunaan *framework*.
3. Mekanisme Kontrol Akses: Implementasi *Role-Based Access Control* (RBAC) pada sistem login berhasil mengamankan alur navigasi, memastikan pemisahan yang tegas antara hak akses Anggota (User) yang bersifat eksploratif dan Administrator yang bersifat manajerial.

5.2 Refleksi dan Pembelajaran

Selama proses pengembangan, terdapat beberapa pembelajaran teknis yang signifikan:

- Pentingnya Validasi Sisi Server: Kami menyadari bahwa validasi input tidak cukup hanya diandalkan pada sisi klien (HTML). Validasi di sisi server (PHP) sangat krusial untuk mencegah data kosong atau format yang tidak sesuai masuk ke dalam *query* basis data.
- Konsistensi Keamanan Data: Dalam implementasi fitur login, penggunaan *Prepared Statement* (*mysqli_prepare*) terbukti efektif menangani input pengguna. Namun, penerapan standar keamanan ini harus dilakukan secara konsisten di seluruh fitur CRUD untuk menutup celah kerentanan sepenuhnya.

5.3 Rencana Pengembangan Selanjutnya (Future Improvements)

Mengingat sistem ini masih dalam tahap pengembangan awal, terdapat dua fokus utama untuk peningkatan sistem agar siap digunakan dalam skala produksi, yaitu penguatan keamanan dan penambahan fitur interaksi peminjaman.

1. Peningkatan Keamanan Sistem (Security Hardening)

Keamanan data menjadi prioritas utama dalam rencana pengembangan berikutnya. Evaluasi terhadap kode crud.php menunjukkan adanya potensi celah yang perlu ditambah:

- Penerapan Prepared Statements Menyeluruh: Saat ini, fungsi penambahan data seperti addBook masih menggunakan metode penyisipan variabel langsung ke dalam string SQL (VALUES ('\$bookID', ...)). Pengembangan selanjutnya akan merefaktorkan seluruh fungsi CRUD untuk menggunakan *Prepared Statements* (bind param) guna memitigasi risiko serangan *SQL Injection* secara total.
- Enkripsi Kata Sandi (Password Hashing): Saat ini sistem membandingkan *password* secara langsung (\$password == \$row['password']). Ke depan, kami akan mengimplementasikan fungsi password_hash() saat registrasi dan password_verify() saat login, sehingga kata sandi tidak lagi tersimpan dalam bentuk teks biasa (*plain text*) di database.
- Perlindungan XSS (Cross-Site Scripting): Menambahkan fungsi sanitasi output seperti htmlspecialchars() pada setiap data yang ditampilkan ke pengguna untuk mencegah eksekusi skrip berbahaya.

2. Fitur "Request Order" (Peminjaman Mandiri)

Untuk meningkatkan efisiensi layanan dan mengurangi antrean fisik, kami merencanakan fitur peminjaman mandiri bagi anggota:

- Mekanisme Booking: Tombol pada katalog buku pengguna (User Mode) tidak hanya akan menampilkan status, tetapi juga berfungsi sebagai tombol "Ajukan Pinjaman" (Request Borrow).
- Tabel Reservasi: Kami akan menambahkan tabel baru (misalnya Reservations) di antara tabel Users dan Books. Saat pengguna menekan tombol pinjam, data akan masuk ke tabel ini dengan status "Pending".
- Persetujuan Admin: Di dashboard admin, akan ditambahkan menu "Daftar Permintaan" di mana admin dapat menyetujui atau menolak pengajuan tersebut. Jika disetujui, data akan otomatis berpindah ke tabel utama Loans dan stok buku akan berkurang secara otomatis.

