Library Borrowing Tracker – Record books, borrowers, and loan histories.

Raditya Nathaniel Nugroho, Marco Christian, Gilbert Nathaniel
Basis Data Kelas KOMA
28 Desember 2025

I. Pendahuluan

Di era transformasi digital saat ini, efisiensi pengelolaan data menjadi aspek krusial dalam operasional institusi layanan publik, termasuk perpustakaan. Namun pada praktiknya, banyak perpustakaan tradisional masih mengandalkan sistem pencatatan manual berbasis buku fisik. Sistem konvensional ini memiliki keterbatasan signifikan, terutama dalam hal aksesibilitas dan keamanan data. Pengguna seringkali kesulitan mengecek ketersediaan koleksi secara *real-time* tanpa harus mencari secara fisik di rak buku. Selain itu, pendataan manual memiliki risiko tinggi terhadap kehilangan atau kerusakan data akibat faktor lingkungan, seperti usia kertas, kelembapan, hingga serangan rayap. Proses pengarsipan manual juga memakan waktu lama dan rentan terhadap kesalahan manusia (*human error*), yang dapat mengurangi integritas data.

Oleh karena itu, diperlukan sebuah solusi digital melalui perancangan sistem basis data modern menggunakan SQL (*Structured Query Language*). Masalah utama yang diangkat dalam penelitian ini adalah bagaimana merancang sistem basis data yang efisien dan terintegrasi, bagaimana sistem tersebut dapat mempermudah pencarian buku bagi pengguna, serta sejauh mana sistem ini mampu meningkatkan kecepatan dan akurasi pengelolaan data dibandingkan metode konvensional.

Tujuan utama dari penulisan ini adalah membangun sistem basis data digital menggunakan SQL yang mampu mengorganisir koleksi buku secara terstruktur, menyediakan mekanisme pencarian data yang cepat, serta mengoptimalkan kinerja manajemen perpustakaan melalui otomasi pengelolaan data yang akurat.

Secara praktis, pengembangan sistem ini memberikan kontribusi positif bagi berbagai pihak. Bagi pengelola, sistem ini mempermudah pemantauan stok dan riwayat peminjaman tanpa redundansi data. Bagi pengguna, sistem memberikan transparansi informasi ketersediaan koleksi secara cepat. Bagi institusi, teknologi ini meningkatkan standar pelayanan operasional sekaligus menjamin keamanan arsip data dalam jangka panjang.

Untuk menggambarkan interaksi antara aktor dan sistem, berikut adalah beberapa skenario penggunaan (*use cases*) utama:

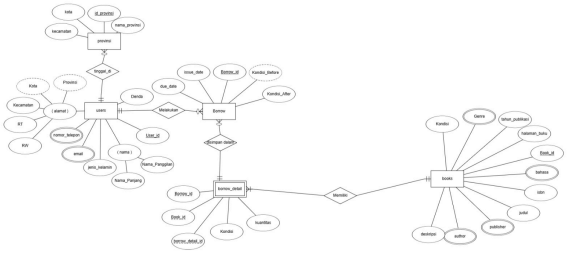
- Pencarian Koleksi: Pengguna mencari buku berdasarkan judul, penulis, atau penerbit melalui basis data untuk melihat status ketersediaan di rak.
- Manajemen Peminjaman: Pustakawan mencatat tanggal peminjaman dan menentukan tenggat waktu (*due date*) pada tabel *deposits* ketika pengguna meminjam buku.
- Pembaruan Inventaris: Admin melakukan operasi *update* pada jumlah stok (*copies*) setiap kali ada buku baru yang masuk atau buku yang rusak/hilang.
- Akses Digital: Pengguna dapat mengidentifikasi apakah sebuah judul memiliki versi digital melalui referensi tabel *ebooks*.

II. Database Design

Perancangan basis data ini bertujuan untuk menciptakan struktur yang sistematis guna mendukung operasional perpustakaan digital secara efisien, aman, dan meminimalkan redundansi data.

Sistem menggunakan model relasional dengan entitas yang saling terhubung untuk mencerminkan proses bisnis peminjaman buku. Berdasarkan diagram ERD yang dirancang,

berikut adalah rincian entitas dan atributnya:



- Users: Entitas untuk data anggota
  - Atribut: User_id (Primary Key), jenis_kelamin, Denda, nomor_telepon (Multivalued), email (Multivalued).
  - Atribut Komposit: nama (Nama_Panjang, Nama_Panggilan) dan alamat (Provinsi, Kota, Kecamatan, RT, RW).
- Books: Entitas untuk koleksi buku
  - Atribut: Book_id (Primary Key), isbn, judul, halaman_buku, tahun_publikasi, Genre, Kondisi, deskripsi.
  - Atribut Multivalued: author, publisher, dan bahasa.
- Borrow: Entitas transaksi peminjaman.
  - Atribut: Borrow_id (Primary Key), issue_date, due_date, Kondisi_Before (Derived), dan Kondisi_After.
- Borrow_detail (Weak Entity): Menghubungkan transaksi peminjaman dengan spesifik buku yang dipinjam.
  - Atribut: borrow_detail_id (Partial Key), Borrow_id (FK), Book_id (FK), Kuantitas, dan Kondisi.
- Provinsi: Entitas pendukung untuk standarisasi alamat.
  - Atribut: id_provinsi (PK), nama_provinsi, kota, kecamatan.

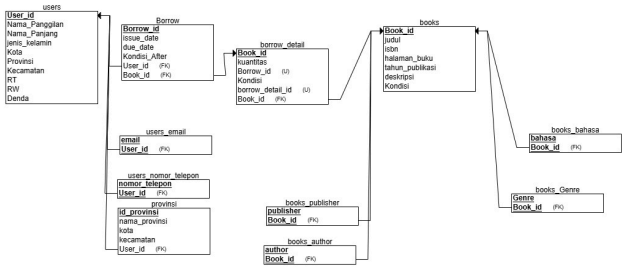
Untuk menjaga integritas data, dilakukan proses normalisasi sebagai berikut:

1) First Normal Form (1NF):
- Menghilangkan atribut bernilai banyak (multivalued) seperti nomor_telepon, email, author, dan publisher. Atribut-atribut ini dipindahkan ke tabel terpisah agar setiap sel berisi nilai atomik.
- Memecah atribut komposit nama dan alamat menjadi kolom mandiri.

2) Second Normal Form (2NF):
- Memastikan semua atribut non-kunci bergantung sepenuhnya pada Primary Key.
- Contoh: Detail provinsi dan kota dipisahkan dari tabel Users ke tabel Provinsi untuk menghindari redundansi data alamat pada setiap baris pengguna.

3) Third Normal Form (3NF):
- Menghilangkan dependensi transitif di mana atribut non-kunci bergantung pada atribut non-kunci lainnya.
- Informasi genre, penulis, dan penerbit dipisahkan ke tabel referensi masing-masing sehingga jika ada perubahan nama penerbit, sistem hanya perlu memperbarui satu record di tabel Publishers.

III. Database Design



Implementasi basis data dilakukan dengan menerjemahkan Entity-Relationship Diagram (ERD) ke dalam skema relasional formal dan perintah SQL untuk membangun struktur penyimpanan yang kokoh

Berdasarkan hasil normalisasi, skema relasional dipecah menjadi beberapa tabel untuk menangani atribut bernilai banyak (*multivalued*) dan memastikan integritas data.

1) Tabel Utama:
   a) users: Menyimpan data profil utama anggota.
   b) books: Menyimpan metadata inti buku (ISBN, judul, tahun, deskripsi).

c) provinsi: Tabel referensi wilayah yang terhubung ke pengguna.

2) Tabel Atribut Multivalued (Hasil Normalisasi):
   a) users_email & users_nomor_telepon: Menangani banyak kontak per satu pengguna.
   b) books_author, books_publisher, books_bahasa, & books_Genre: Menangani metadata buku yang dapat memiliki lebih dari satu nilai.

3) Tabel Transaksi:
   a) Borrow: Mencatat data utama peminjaman (siapa dan kapan).
   b) borrow_detail: Tabel antara yang merinci buku apa saja yang dipinjam dalam satu transaksi peminjaman (termasuk jumlah dan kondisi).

Sistem ini menerapkan batasan kunci yang ketat untuk menjaga kualitas data:

- Primary Key (PK): Setiap tabel memiliki pengidentifikasi unik (contoh: User_id, Book_id, Borrow_id) untuk mencegah duplikasi data.
- Foreign Key (FK): Digunakan untuk membangun relasi antar tabel (contoh: User_id pada tabel Borrow merujuk ke tabel users).
- Referential Integrity: Penggunaan batasan kunci tamu memastikan bahwa transaksi peminjaman tidak dapat dibuat untuk pengguna atau buku yang tidak terdaftar dalam sistem.

Contoh SQL query untuk mengimplementasikan beberapa tabel utama berdasarkan skema tersebut:

```
-- Pembuatan Tabel Utama Users
CREATE TABLE users (
   User_id INT PRIMARY KEY,
   Nama_Panggilan VARCHAR(50),
   Nama_Panjang VARCHAR(100),
   jenis_kelamin VARCHAR(10),
   Kota VARCHAR(50),
   Provinsi VARCHAR(50),
   Kecamatan VARCHAR(50),
   RT INT,
   RW INT,
   Denda DECIMAL(10, 2) DEFAULT 0
);

-- Pembuatan Tabel Books
CREATE TABLE books (
   Book_id INT PRIMARY KEY,
   judul VARCHAR(255) NOT NULL,
   isbn VARCHAR(20) UNIQUE,
   halaman_buku INT,
   tahun_publikasi YEAR,
   deskripsi TEXT,
   Kondisi VARCHAR(50)
);

-- Pembuatan Tabel Transaksi Borrow
CREATE TABLE Borrow (
   Borrow_id INT PRIMARY KEY,
   issue_date DATE NOT NULL,
   due_date DATE NOT NULL,
   Kondisi_After VARCHAR(50),
   User_id INT,
   Book_id INT,
    FOREIGN KEY (User_id) REFERENCES users(User_id),
    FOREIGN KEY (Book_id) REFERENCES books(Book_id)
);

-- Pembuatan Tabel Multivalued untuk Author
CREATE TABLE books_author (
   author VARCHAR(100),
   Book_id INT,
   PRIMARY KEY (author, Book_id),
    FOREIGN KEY (Book_id) REFERENCES books(Book_id)
);
```

IV. Application Implementation

Database disimpan di localhost myphpadmin XAMPP dengan detail entitas sebagai berikut :

1. books

#	Nama	Jenis	Penyortiran	Atribut	Tak Ternilai	Bawaan	Komentar	Ekstra	Tindakan
1	book_id	int(11)			Tidak	Tidak ada		AUTO_INCREMENT	Ubah Hap
2	judul	varchar(120)	utf8mb4_general_ci		Ya	NULL			Ubah Hap
3	isbn	varchar(100)	utf8mb4_general_ci		Tidak	Tidak ada			Ubah Hap
4	tahun_publikasi	date			Tidak	Tidak ada			Ubah Hap
5	deskripsi	text	utf8mb4_general_ci		Ya	NULL			Ubah Hap
6	stock	int(11)			Ya	NULL			Ubah Hap
7	bisa_dipinjam	enum('ya', 'tidak')	utf8mb4_general_ci		Tidak	Tidak ada			Ubah Hap
8	kondisi	enum('Sempurna', 'Baik', 'Lumayan', 'Buruk', 'Rusa...	utf8mb4_general_ci		Ya	NULL			Ubah Hap

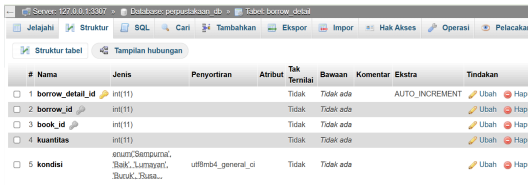
2. users

#	Nama	Jenis	Penyortiran	Atribut	Tak Ternilai	Bawaan	Komentar	Ekstra	Tindakan
1	user_id	int(11)			Tidak	Tidak ada		AUTO_INCREMENT	Ubah Hap
2	denda	int(11)			Ya	NULL			Ubah Hap
3	rt	smallint(6)			Tidak	Tidak ada			Ubah Hap
4	rw	smallint(6)			Tidak	Tidak ada			Ubah Hap
5	jenis_kelamin	enum('laki-laki', 'perempuan')	utf8mb4_general_ci		Tidak	Tidak ada			Ubah Hap
6	nama_panjang	varchar(120)	utf8mb4_general_ci		Tidak	Tidak ada			Ubah Hap
7	nama_panggilan	varchar(60)	utf8mb4_general_ci		Tidak	Tidak ada			Ubah Hap
8	id_provinsi	int(11)			Ya	NULL			Ubah Hap

3. Borrow :

#	Nama	Jenis	Penyortiran	Atribut	Tak Ternilai	Bawaan	Komentar	Ekstra	Tindakan
1	borrow_id	int(11)			Tidak	Tidak ada		AUTO_INCREMENT	Ubah Hap
2	issue_date	date			Tidak	Tidak ada			Ubah Hap
3	due_date	date			Tidak	Tidak ada			Ubah Hap
4	kondisi_after	enum('Sempurna', 'Baik', 'Lumayan', 'Buruk', 'Rusa...	utf8mb4_general_ci		Ya	NULL			Ubah Hap
5	book_id	int(11)			Ya	NULL			Ubah Hap
6	user_id	int(11)			Ya	NULL			Ubah Hap
7	kondisi_before	enum('Sempurna', 'Baik', 'Lumayan', 'Buruk', 'Rusa...	utf8mb4_general_ci		Tidak	Tidak ada			Ubah Hap

4. borrow_detail :



#	Nama	Jenis	Penyortiran	Atribut	Tak Ternilai	Bawaan	Komentar	Ekstra	Tindakan
1	borrow_detail_id	int(11)			Tidak	Tidak ada		AUTO_INCREMENT	Ubah Hap
2	borrow_id	int(11)			Tidak	Tidak ada			Ubah Hap
3	book_id	int(11)			Tidak	Tidak ada			Ubah Hap
4	kuantitas	int(11)			Tidak	Tidak ada			Ubah Hap
5	kondisi	enum('Sempurna', 'Baik', 'Lumayan', 'Buruk', 'Rusa...	utf8mb4_general_ci		Tidak	Tidak ada			Ubah Hap

5. Provinsi :

#	Nama	Jenis	Penyortiran	Atribut	Tak Ternilai	Bawaan	Komentar	Ekstra	Tindakan
1	id_provinsi	int(11)			Tidak	Tidak ada			Ubah Hapus Lainnya
2	nama_provinsi	varchar(100)	utf8mb4_general_ci		Tidak	Tidak ada			Ubah Hapus Lainnya
3	kota	varchar(100)	utf8mb4_general_ci		Tidak	Tidak ada			Ubah Hapus Lainnya
4	kecamatan	varchar(100)	utf8mb4_general_ci		Tidak	Tidak ada			Ubah Hapus Lainnya

Struktur atribut multivalued terdiri dari tiga bagian. Atribut multivalued_id adalah kunci primer. Atribut multivalued yang dimaksud menyimpan nilai data. Id_foreign key adalah kunci asing ke tabel induk.

Contoh: Users_Email

Contohnya adalah users_email. Atribut multivalued_id-nya adalah users_email_id. Atribut multivalued yang dimaksud adalah email_value. Id_foreign key-nya adalah user_id.

#	Nama	Jenis	Penyortiran	Atribut	Tak Ternilai	Bawaan	Komentar	Ekstra	Tindakan
1	email_id	int(11)			Tidak	Tidak ada		AUTO_INCREMENT	Ubah Hapus Lainnya
2	email	varchar(100)	utf8mb4_general_ci		Ya	NULL			Ubah Hapus Lainnya
3	user_id	int(11)			Ya	NULL			Ubah Hapus Lainnya

Integrasi antarmuka dibangun dengan PHP murni yang terkandung dalam HTML dan CSS, tanpa menggunakan JavaScript. Sistem ini terdiri dari 92 berkas kode.

Logika alur utamanya adalah: dimulai dari login_page.php, kemudian menuju dashboard.php. Dari dashboard, pengguna memilih salah satu entitas utama, yaitu books, users, borrow, atau borrow_detail. Setelah itu, pengguna dapat melihat data utama atau data atribut multivalue. Selanjutnya, pengguna menjalankan operasi CRUD. Operasi CRUD tersebut secara langsung memengaruhi entitas utama atau atribut multivalue yang bersangkutan.

Rincian Logika crud :

a. Logika create :

Logika operasi create dimulai dari halaman entitas utama. Pengguna memilih opsi untuk menambah data, yang akan membuka halaman tambah_entitas.php. Halaman tersebut berisi formulir yang mengumpulkan nilai untuk setiap variabel yang diperlukan dalam operasi query SQL. Data yang telah diisi kemudian dikirimkan ke berkas tambah_entitas_proses.php. Di dalam berkas proses tersebut, data yang

diperoleh dari pengguna digunakan untuk menjalankan query sebagai berikut:

```
$query = "INSERT INTO nama_tabel
(atribut1, atribut2, ……, atribut i)
//sesuai banyaknya atribut
VALUES (?, ?, ……, ? )";
$stmt = mysqli_prepare($conn,
$query);
mysqli_stmt_bind_param(
   $stmt,
   "Si, ……", //s untuk string dan i
untuk integer (datatype)
   $variabelatribut1,
$variabelatribut2, ……,
variabelatributi
);
```

b. Logika update :

Logika operasi update dimulai dari halaman entitas utama. Pengguna memilih opsi untuk memperbarui data, yang akan membuka halaman update_entitas.php. Halaman tersebut berisi tombol pada setiap baris data yang merupakan hidden submit form. Tombol ini berfungsi untuk mengirimkan primary key dari data yang dipilih. Primary key yang dikirim tersebut nantinya akan digunakan sebagai parameter untuk masuk ke halaman update_entitas_form.php.

```
   <form
action="update_entitas_form.php"
method="POST">
         <input type="hidden"
name="primary_key" value="<?=
$row['primary_key'] ?>">
         <button type="submit"
id="">UPDATE</button>
```

```
      </form>
```

Kemudian, update_entitas_form.php akan meminta input data dari pengguna. Data yang dimasukkan akan disimpan dalam variabel-variabel untuk digunakan dalam proses pembaruan data. Proses ini mirip dengan operasi create. Sebelum menampilkan formulir, berkas update_entitas_form.php akan terlebih dahulu menjalankan query untuk mengambil data yang ada. Query ini menggunakan parameter primary key yang disuplai oleh update_entitas.php pada tahap sebelumnya.

```
$query = "SELECT * FROM entitas
WHERE primary_key= ?";
```

Kemudian, proses dilanjutkan ke berkas update_entitas_proses.php. Di dalam berkas tersebut, dijalankan query SQL yang sesuai untuk memperbarui data berdasarkan variabel yang telah disiapkan.

```
$query = "UPDATE entitas SET
atribut1=?, atribut2=?, atribut3=?,
atribut4=?, atribut5=?;
$stmt = mysqli_prepare($conn,
$query);
mysqli_stmt_bind_param($stmt,
"ssssi", $atribut1, $atribut2,
$atribut3, $atribut4, $atribut5,
$bisa_dipinjam, $kondisi, $book_id);
//tipe data daan juga atributnya
mysqli_stmt_execute($stmt);
```

Maka, query tersebut akan memperbarui nilai dalam tabel entitas yang telah dipilih oleh pengguna

c. Logika delete :

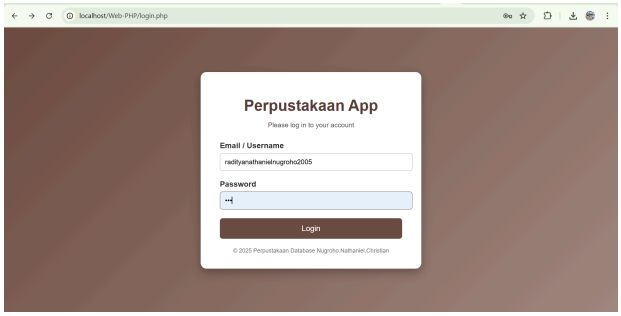
Logika operasi delete dimulai dari halaman entitas utama. Pengguna memilih opsi untuk menghapus data, yang akan membuka halaman hapus_entitas.php. Halaman tersebut berisi formulir yang akan mengisi variabel yang diperlukan untuk operasi query SQL di berkas PHP berikutnya. Data kemudian diproses di hapus_entitas_proses.php. Data yang diperoleh dari input pengguna digunakan untuk menjalankan query penghapusan.

```
$query = "DELETE FROM entitas
WHERE primary_key= ?";
$stmt = mysqli_prepare($conn,
$query);
mysqli_stmt_bind_param($stmt, "i",
$primary_key);
mysqli_stmt_execute($stmt);
```

Logika tersebut digunakan secara berulang untuk seluruh entitas kuat, entitas lemah, dan entitas atribut multivalue.

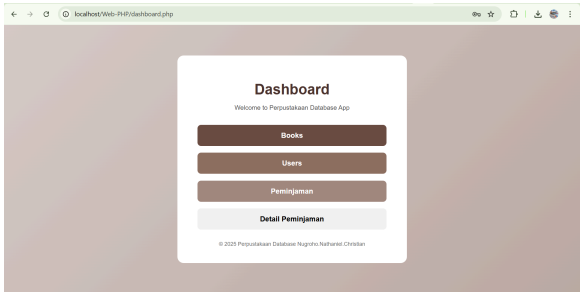
Contoh *running through* applikasi

Login_page.php :



Tampilan utama aplikasi perpustakaan memiliki dua bidang untuk mengisi nilai email dan nama pengguna. Karena fitur autentikasi belum tersedia, sistem hanya dapat mendeteksi apakah kedua bidang tersebut dalam keadaan kosong atau telah terisi. Pengguna hanya dapat masuk ke sistem jika kedua bidang tersebut telah terisi.

Dashboard.php :

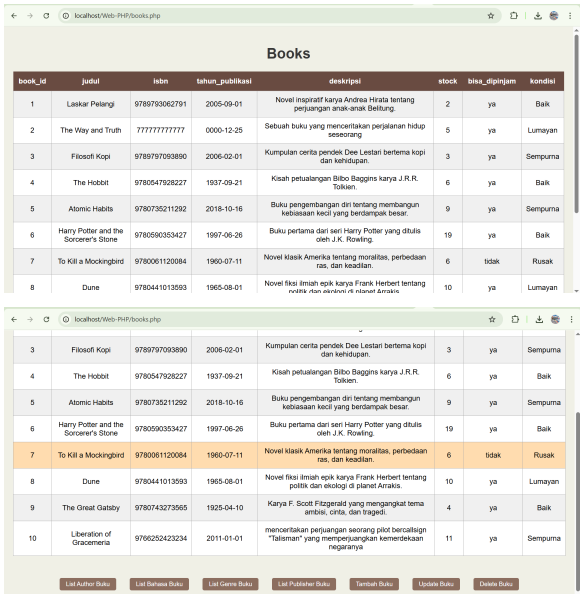


Tampilan dashboard setelah masuk ke sistem menampilkan entitas-entitas utama, yaitu books, users, borrow (peminjaman), dan detail peminjaman. Selain itu, terdapat pula entitas multivalue seperti books_author, books_genre, books_publisher, books_bahasa, users_email, users_nomor telepon, dan provinsi.

Contoh integrasi :

Untuk contohnya akan diambil dari entitas utama books :

Books.php



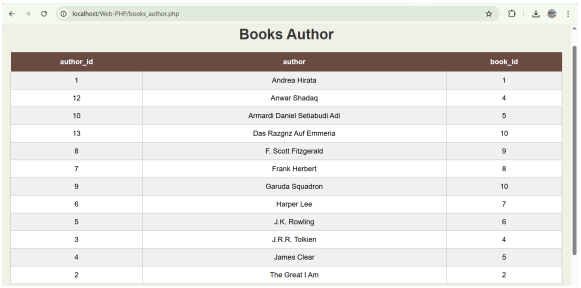
book_id	judul	isbn	tahun_publikasi	deskripsi	stock	bisa_dipinjam	kondisi
1	Laskar Pelangi	9789793062791	2005-09-01	Novel inspiratif karya Andrea Hirata tentang perjuangan anak-anak Belitung.	2	ya	Baik
2	The Way and Truth	7777777777777	0000-12-25	Sebuah buku yang menceritakan perjalanan hidup seseorang	5	ya	Lumayan
3	Filosofi Kopi	9789797093890	2006-02-01	Kumpulan cerita pendek Dee Lestari bertema kopi dan kehidupan.	3	ya	Sempurna
4	The Hobbit	9780547928227	1937-09-21	Kisah petualangan Bilbo Baggins karya J.R.R. Tolkien.	6	ya	Baik
5	Atomic Habits	9780735211292	2018-10-16	Buku pengembangan diri tentang membangun kebiasaan kecil yang berdampak besar.	9	ya	Sempurna
6	Harry Potter and the Sorcerer's Stone	9780590353427	1997-06-26	Buku pertama dari seri Harry Potter yang ditulis oleh J.K. Rowling.	19	ya	Baik
7	To Kill a Mockingbird	9780061120084	1960-07-11	Novel klasik Amerika tentang moralitas, perbedaan ras, dan keadilan.	6	tidak	Rusak
8	Dune	9780441013593	1965-08-01	Novel fiksi ilmiah epik karya Frank Herbert tentang politik dan ekologi di planet Arrakis.	10	ya	Lumayan

3	Filosofi Kopi	9789797093890	2006-02-01	Kumpulan cerita pendek Dee Lestari bertema kopi dan kehidupan.	3	ya	Sempurna
4	The Hobbit	9780547928227	1937-09-21	Kisah petualangan Bilbo Baggins karya J.R.R. Tolkien.	6	ya	Baik
5	Atomic Habits	9780735211292	2018-10-16	Buku pengembangan diri tentang membangun kebiasaan kecil yang berdampak besar.	9	ya	Sempurna
6	Harry Potter and the Sorcerer's Stone	9780590353427	1997-06-26	Buku pertama dari seri Harry Potter yang ditulis oleh J.K. Rowling.	19	ya	Baik
7	To Kill a Mockingbird	9780061120084	1960-07-11	Novel klasik Amerika tentang moralitas, perbedaan ras, dan keadilan.	6	tidak	Rusak
8	Dune	9780441013593	1965-08-01	Novel fiksi ilmiah epik karya Frank Herbert tentang politik dan ekologi di planet Arrakis.	10	ya	Lumayan

| 9 | The Great Gatsby | 9780743273565 | 1925-04-10 | Karya F. Scott Fitzgerald yang mengangkat tema ambisi, cinta, dan tragedi. | 4 | ya | Baik |
| 10 | Liberation of Gracemeria | 9766252423234 | 2011-01-01 | menceritakan perjuangan seorang pilot bercalsign "Talisman" yang memperjuangkan kemerdekaan negaranya | 11 | ya | Sempurna |

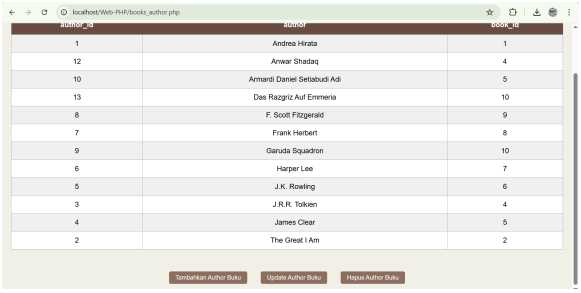
Sistem dilengkapi dengan tabel database dan fitur hover untuk mempermudah pembacaan data. Di bagian bawah antarmuka, terdapat daftar tabel dari atribut multivalue serta tombol untuk operasi CRUD.

Books_author.php :



Books Author

author_id	author	book_id
1	Andrea Hirata	1
12	Anwar Shadaq	4
10	Armardi Daniel Setiabudi Adi	5
13	Das Razgriz Auf Emmeria	10
8	F. Scott Fitzgerald	9
7	Frank Herbert	8
9	Garuda Squadron	10
6	Harper Lee	7
5	J.K. Rowling	6
3	J.R.R. Tolkien	4
4	James Clear	5
2	The Great I Am	2



author_id	author	book_id
1	Andrea Hirata	1
12	Anwar Shadaq	4
10	Armardi Daniel Setiabudi Adi	5
13	Das Razgriz Auf Emmeria	10
8	F. Scott Fitzgerald	9
7	Frank Herbert	8
9	Garuda Squadron	10
6	Harper Lee	7
5	J.K. Rowling	6
3	J.R.R. Tolkien	4
4	James Clear	5
2	The Great I Am	2

Tambahkan Author Buku | Update Author Buku | Hapus Author Buku

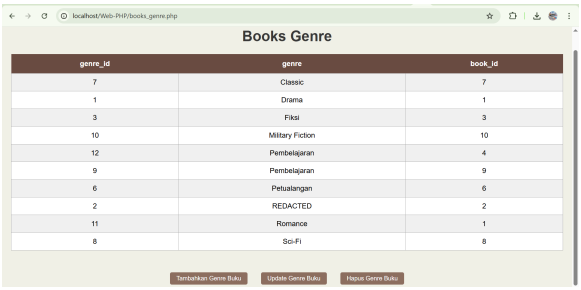
Books_bahasa.php :

Bahasa Buku

bahasa_id	bahasa	book_id
10	Emerian	10
5	English	5
7	English	7
8	English	8
9	English	9
6	High Valerian	6
2	Ibrani	2
1	Indonesia	1
3	Indonesia	3
11	Latin	2

Tambahkan Bahasa Buku | Update Bahasa Buku | Hapus Bahasa Buku

books_genre.php

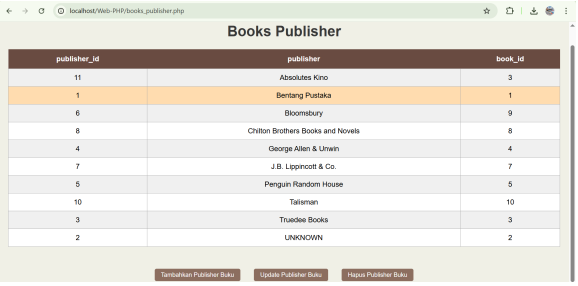


Books Genre

genre_id	genre	book_id
7	Classic	7
1	Drama	1
3	Fiksi	3
10	Military Fiction	10
12	Pembelajaran	4
9	Pembelajaran	9
6	Petualangan	6
2	REDACTED	2
11	Romance	1
8	Sci-Fi	8

Tambahkan Genre Buku | Update Genre Buku | Hapus Genre Buku

Books_publisher.php

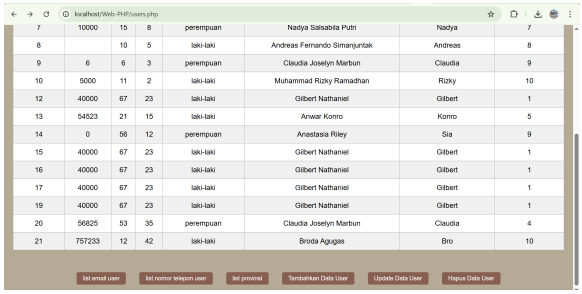


Books Publisher

publisher_id	publisher	book_id
11	Absolutes Kino	3
1	Bentang Pustaka	1
6	Bloomsbury	9
8	Chilton Brothers Books and Novels	8
4	George Allen & Unwin	4
7	J.B. Lippincott & Co.	7
5	Penguin Random House	5
10	Talisman	10
3	Truedee Books	3
2	UNKNOWN	2

Tambahkan Publisher Buku | Update Publisher Buku | Hapus Publisher Buku

Aplikasi ini telah mendukung operasi CRUD pada basis data. Sebagai contoh, operasi CRUD dapat dilakukan pada basis data users. Apabila pengguna membuka user.php dan menggulir ke bawah, akan terlihat serangkaian tombol sebagai berikut:

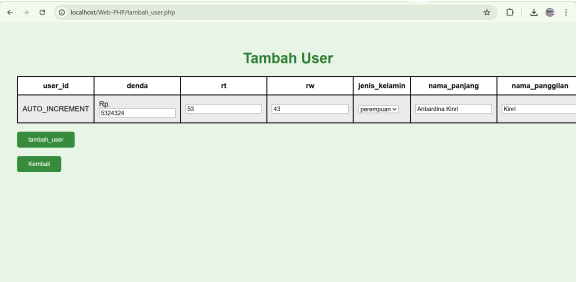


7	10000	15	8	perempuan	Nadya Salsabila Putri	Nadya	7
8		10	5	laki-laki	Andreas Fernando Simanjuntak	Andreas	8
9	6	6	3	perempuan	Claudia Joselyn Marbun	Claudia	9
10	5000	11	2	laki-laki	Muhammad Rizky Ramadhan	Rizky	10
12	40000	67	23	laki-laki	Gilbert Nathaniel	Gilbert	1
13	54523	21	15	laki-laki	Anwar Konro	Konro	5
14	0	56	12	perempuan	Anastasia Riley	Sia	9
15	40000	67	23	laki-laki	Gilbert Nathaniel	Gilbert	1
16	40000	67	23	laki-laki	Gilbert Nathaniel	Gilbert	1
17	40000	67	23	laki-laki	Gilbert Nathaniel	Gilbert	1
19	40000	67	23	laki-laki	Gilbert Nathaniel	Gilbert	1
20	56825	53	35	perempuan	Claudia Joselyn Marbun	Claudia	4
21	757233	12	42	laki-laki	Broda Agugas	Bro	10

list email user | list nomor telepon user | list provinsi | Tambahkan Data User | Update Data User | Hapus Data User

Jika pengguna mengklik tombol "Tambah Data User", maka sistem akan membuka halaman tambah_user.php.

tambah_user.php

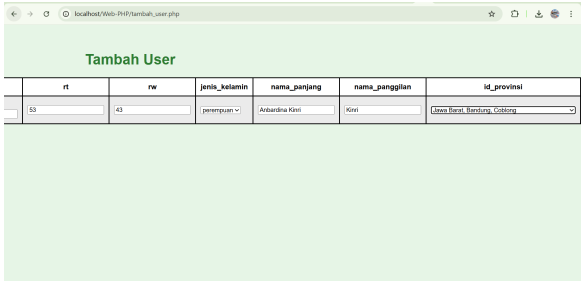


Tambah User

user_id	denda	rt	rw	jenis_kelamin	nama_panjang	nama_panggilan
AUTO_INCREMENT	Rp. 5324324	53	43	perempuan	Anbardina Kirri	Kirri

tambah_user

Kembali

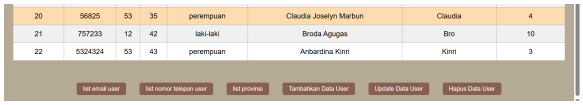


Nilai pada formulir dapat diisi oleh pengguna. Selanjutnya, dengan mengklik tombol "Tambah User", sistem akan menjalankan berkas tambah_user_proses.php dan menambahkan pengguna sesuai dengan data yang diinginkan.

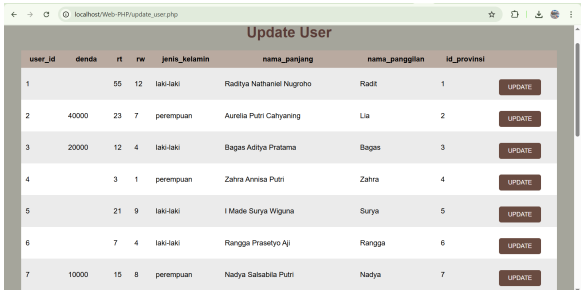
Nantinya menjalankan query :

```
$query = "INSERT INTO users
(denda, rt, rw, jenis_kelamin, nama_panjang,
nama_panggilan, id_provinsi)
VALUES (?, ?, ?, ?, ?, ?, ?)";
```

Users.php setelah create



Selanjutnya, jika pengguna mengklik "Update Data User", sistem akan menampilkan antarmuka sebagai berikut ketika memasuki halaman update_user.php:



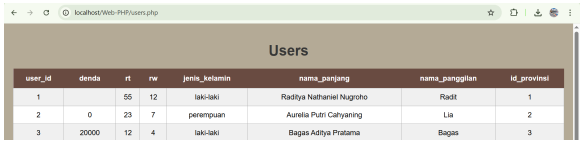
Setiap tombol tersebut merupakan hidden submit form. Tombol tersebut berfungsi untuk mengirimkan primary key dari entitas—dalam hal ini user_id—ke halaman update_entitas_form.php, yaitu update_user_form.php. Sebagai contoh, apabila pengguna ingin memperbarui data untuk user_id 2, misalnya mengubah nominal menjadi 0 rupiah, maka pengguna akan mengklik tombol update pada baris yang sesuai dengan user_id 2.

update_user_form.php

Kemudian, pengguna menyimpan perubahan tersebut. Tindakan ini akan mengirimkan nilai-nilai yang diperlukan untuk menjalankan berkas update_user_proses.php, yang selanjutnya akan mengeksekusi query pembaruan data.

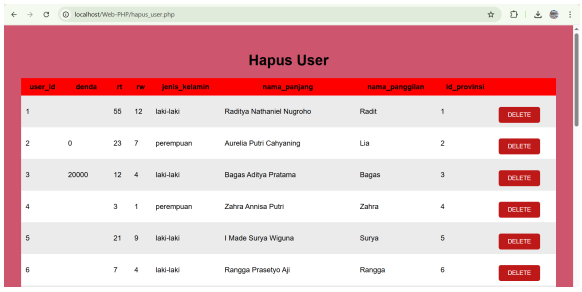
"UPDATE users SET denda=?, rt=?, rw=?, jenis_kelamin=?, nama_panjang=?, nama_panggilan=?, id_provinsi=? WHERE user_id=?"

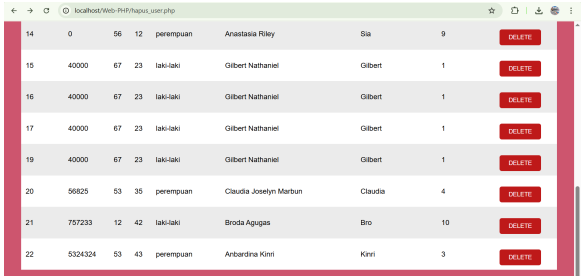
Users.php setelah update :



Perubahan dapat dilihat, dendanya telah menjadi 0. Selanjutnya, untuk menghapus data, pengguna mengklik tombol "Hapus Data User".

hapus_user.php





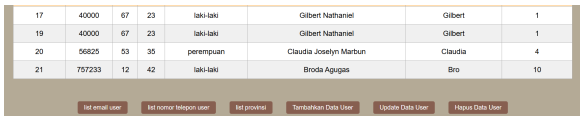
14	0	56	12	perempuan	Anastasia Riley	Sia	9	DELETE
15	40000	67	23	laki-laki	Gilbert Nathaniel	Gilbert	1	DELETE
16	40000	67	23	laki-laki	Gilbert Nathaniel	Gilbert	1	DELETE
17	40000	67	23	laki-laki	Gilbert Nathaniel	Gilbert	1	DELETE
19	40000	67	23	laki-laki	Gilbert Nathaniel	Gilbert	1	DELETE
20	56825	53	35	perempuan	Claudia Joselyn Marbun	Claudia	4	DELETE
21	757233	12	42	laki-laki	Broda Agugas	Bro	10	DELETE
22	5324324	53	43	perempuan	Anbardina Kinri	Kinri	3	DELETE

Model logika operasi delete ini mirip dengan operasi update. Setiap tombol delete mengirimkan hidden submission berupa primary key—dalam konteks ini user_id—ke berkas hapus_user_proses.php. Selanjutnya, dengan data user_id tersebut, berkas hapus_user_proses.php akan menjalankan query penghapusan yang sesuai.

```
$query = "DELETE FROM users WHERE user_id = ?";
```

Dengan demikian, data yang dipilih akan terhapus dari sistem. Sebagai contoh, jika pengguna ingin menghapus data milik Anbardina Kinri (dengan user_id = 22), maka langkah tersebut akan menghapus data tersebut.

Users.php setelah deletion :



17	40000	67	23	laki-laki	Gilbert Nathaniel	Gilbert	1
19	40000	67	23	laki-laki	Gilbert Nathaniel	Gilbert	1
20	56825	53	35	perempuan	Claudia Joselyn Marbun	Claudia	4
21	757233	12	42	laki-laki	Broda Agugas	Bro	10

list email user | list nomor telepon user | list provinsi | Tambahkan Data User | Update Data User | Hapus Data User

V. Conclusion and Reflection

Pengembangan sistem "Library Borrowing Tracker" berhasil membuktikan bahwa transisi dari manajemen perpustakaan tradisional ke sistem basis data digital berbasis SQL dapat mengatasi hambatan operasional yang signifikan. Sistem ini mampu meningkatkan efisiensi pencarian buku, menjamin keamanan arsip data dari kerusakan fisik, serta menyediakan transparansi ketersediaan koleksi secara *real-time* bagi pengguna. Melalui implementasi operasi CRUD dan integrasi antarmuka web, pengelolaan data peminjaman kini menjadi lebih terintegrasi, akurat, dan minim risiko kehilangan informasi dibandingkan metode konvensional.

Beberapa kendala yang dihadapi selama pengerjaan meliputi:

- Kompleksitas ERD: Memetakan atribut komposit seperti alamat dan atribut bernilai banyak (*multivalued*) agar sesuai dengan aturan normalisasi tanpa menghilangkan detail informasi asli.
- Sinkronisasi Data: Memastikan setiap aksi pada antarmuka aplikasi (*front-end*) melakukan pembaruan data yang tepat pada beberapa tabel terkait di *back-end* secara bersamaan.

Sistem ini memiliki potensi untuk dikembangkan lebih lanjut di masa depan melalui:

- Otomasi Denda: Mengintegrasikan logika penghitungan denda otomatis berdasarkan selisih tanggal pengembalian dan due_date.
- Fitur E-Book: Mengoptimalkan entitas ebooks agar pengguna dapat langsung mengunduh atau membaca versi digital melalui aplikasi.
- Sistem Notifikasi: Menambahkan fitur pengingat melalui email atau nomor telepon bagi anggota yang mendekati batas waktu pengembalian buku.

VI. Appendix

--Create (membuat)

```
create table users(
user_id int auto_increment,
denda int null,
provinsi varchar(120) not null,
kota varchar(100) not null,
kecamatan varchar(100) not null,
rt smallint not null,
rw smallint not null,
jenis_kelamin enum('laki-laki','perempuan') not null,
```

```
nama_panjang varchar(120) not null,
nama_panggilan varchar(60) not null,
primary key (user_id)
);

create table books(
book_id int auto_increment,
judul varchar(120),
isbn varchar(100) not null,
tahun_publikasi date not null,
deskripsi text,
stock int null,
bisa_dipinjam enum('ya', 'tidak') not null,
kondisi enum
('Sempurna','Baik','Lumayan','Buruk','Rusak','Hil
ang'),
primary key (book_id)
);

create table books_author(
author varchar(120),
book_id int,
foreign key (book_id) references
books(book_id)
);

create table books_bahasa(
bahasa varchar(120),
book_id int,
foreign key (book_id) references
books(book_id)
);

create table books_publisher(
publisher varchar(120),
book_id int,
foreign key (book_id) references
books(book_id)
);

create table books_genre(
genre varchar(80),
book_id int,
foreign key (book_id) references
books(book_id)
);

create table borrow(
borrow_id int auto_increment,
issue_date date not null,
due_date date not null,
kondisi_after enum
('Sempurna','Baik','Lumayan','Buruk','Rusak','Hil
ang'),
book_id int,
user_id int,
primary key (borrow_id),
foreign key (user_id) references users(user_id)
);

create table copies(
dimpinjam int,
borrow_id int,
book_id int,
foreign key (book_id) references
books(book_id),
foreign key (borrow_id) references
borrow(borrow_id)
);

create table users_email(
email varchar(100) null,
user_id int,
foreign key (user_id) references users(user_id)
);

create table users_nomor_Telepon(
nomor_telepon varchar(100) null,
user_id int,
foreign key (user_id) references users(user_id)
);

--Inserts
INSERT INTO users (denda, provinsi, kota,
kecamatan, rt, rw, jenis_kelamin, nama_panjang,
nama_panggilan)
VALUES
```

(NULL, 'DIY', 'Yogyakarta', 'Gondokusuman', 55, 12, 'laki-laki', 'Raditya Nathaniel Nugroho', 'Radit'),
(40000, 'Jawa Tengah', 'Semarang', 'Tembalang', 23, 7, 'perempuan', 'Aurelia Putri Cahyaning', 'Lia'),
(20000, 'Jawa Barat', 'Bandung', 'Coblong', 12, 4, 'laki-laki', 'Bagas Aditya Pratama', 'Bagas'),
(NULL, 'DKI Jakarta', 'Jakarta Selatan', 'Tebet', 3, 1, 'perempuan', 'Zahra Annisa Putri', 'Zahra'),
(NULL, 'Bali', 'Denpasar', 'Denpasar Timur', 21, 9, 'laki-laki', 'I Made Surya Wiguna', 'Surya'),
(NULL, 'DIY', 'Bantul', 'Kasihan', 7, 4, 'laki-laki', 'Rangga Prasetyo Aji', 'Rangga'),
(10000, 'Jawa Timur', 'Surabaya', 'Wonokromo', 15, 8, 'perempuan', 'Nadya Salsabila Putri', 'Nadya'),
(NULL, 'Sumatera Utara', 'Medan', 'Medan Kota', 10, 5, 'laki-laki', 'Andreas Fernando Simanjuntak', 'Andreas'),
(NULL, 'Kalimantan Timur', 'Balikpapan', 'Balikpapan Selatan', 6, 3, 'perempuan', 'Claudia Joselyn Marbun', 'Claudia'),
(5000, 'Sulawesi Selatan', 'Makassar', 'Panakkukang', 11, 2, 'laki-laki', 'Muhammad Rizky Ramadhan', 'Rizky');

INSERT INTO books
(judul, isbn, tahun_publikasi, deskripsi, stock, bisa_dipinjam, kondisi)
VALUES
('Laskar Pelangi', '9789793062791', '2005-09-01',
 'Novel inspiratif karya Andrea Hirata tentang perjuangan anak-anak Belitung.',
 12, 'ya', 'Baik'),

('The Way and Truth', '777777777777', '0000-12-25',
 'Sebuah buku yang menceritakan perjalanan hidup seseorang',
 7, 'ya', 'Lumayan'),

('Filosofi Kopi', '9789797093890', '2006-02-01',
 'Kumpulan cerita pendek Dee Lestari bertema kopi dan kehidupan.',
 5, 'ya', 'Sempurna'),

('The Hobbit', '9780547928227', '1937-09-21',
 'Kisah petualangan Bilbo Baggins karya J.R.R. Tolkien.',
 9, 'ya', 'Baik'),

('Atomic Habits', '9780735211292', '2018-10-16',
 'Buku pengembangan diri tentang membangun kebiasaan kecil yang berdampak besar.',
 14, 'ya', 'Sempurna'),

('Harry Potter and the Sorcerer''s Stone', '9780590353427', '1997-06-26',
 'Buku pertama dari seri Harry Potter yang ditulis oleh J.K. Rowling.',
 20, 'ya', 'Baik'),

('To Kill a Mockingbird', '9780061120084', '1960-07-11',
 'Novel klasik Amerika tentang moralitas, perbedaan ras, dan keadilan.',
 6, 'tidak', 'Rusak'),

('Dune', '9780441013593', '1965-08-01',
 'Novel fiksi ilmiah epik karya Frank Herbert tentang politik dan ekologi di planet Arrakis.',
 10, 'ya', 'Lumayan'),

('The Great Gatsby', '9780743273565', '1925-04-10',
 'Karya F. Scott Fitzgerald yang mengangkat tema ambisi, cinta, dan tragedi.',
 4, 'ya', 'Baik'),

('Liberation of Gracemeria', '9766252423234', '2011-01-01',
 'menceritakan perjuangan seorang pilot bercallsign "Talisman" yang memperjuangkan kemerdekaan negaranya',

```
 11, 'ya', 'Sempurna');


INSERT INTO books_author (author, book_id)
VALUES
('Andrea Hirata', 1),
('The Great I Am', 2),
('Dee Lestari', 3),
('J.R.R. Tolkien', 4),
('James Clear', 5),
('J.K. Rowling', 6),
('Harper Lee', 7),
('Frank Herbert', 8),
('F. Scott Fitzgerald', 9),
('Garuda Squadron', 10);

INSERT INTO books_bahasa (bahasa, book_id)
VALUES
('Indonesia', 1),
('Ibrani', 2),
('Indonesia', 3),
('English', 4),
('English', 5),
('High Valerian', 6),
('English', 7),
('English', 8),
('English', 9),
('Emerian', 10);

INSERT INTO books_publisher (publisher,
book_id) VALUES
('Bentang Pustaka', 1),
('UNKNOWN', 2),
('Truedee Books', 3),
('George Allen & Unwin', 4),
('Penguin Random House', 5),
('Bloomsbury', 6),
('J.B. Lippincott & Co.', 7),
('Chilton Books', 8),
('Charles Scribner\'s Sons', 9),
('Talisman', 10);

INSERT INTO books_genre (genre, book_id)
VALUES
('Drama', 1),
('REDACTED', 2),
('Fiksi', 3),
('Fantasy', 4),
('Self-Improvement', 5),
('Fantasy', 6),
('Classic', 7),
('Sci-Fi', 8),
('Classic', 9),
('Military Fiction', 10);

INSERT INTO users_email (email, user_id)
VALUES
('raditya.nugroho@mail.com', 1),
('aulia.salsabila@mail.com', 2),
('budi.santoso@mail.com', 3),
('siti.nuraini@mail.com', 4),
('michael.tan@mail.com', 5),
('linda.wijaya@mail.com', 6),
('agus.pratama@mail.com', 7),
('nina.rahma@mail.com', 8),
('eko.susanto@mail.com', 9),
('joko.hartono@mail.com', 10);

INSERT INTO users_nomor_telepon
(nomor_telepon, user_id) VALUES
('081234567801', 1),
('081234567802', 2),
('081234567803', 3),
('081234567804', 4),
('081234567805', 5),
('081234567806', 6),
('081234567807', 7),
('081234567808', 8),
('081234567809', 9),
('081234567810', 10);

INSERT INTO borrow (issue_date, due_date,
kondisi_after, book_id, user_id) VALUES
('2025-08-16', '2025-11-23', 'Lumayan', 1, 3),
('2025-08-16', '2025-11-23', 'Sempurna', 2, 3),
('2024-02-12', '2025-03-27', 'Rusak', 3, 6),
('2023-03-30', '2025-04-12', 'Hilang', 4, 2),
('2024-07-11', '2025-08-21', 'Baik', 7, 9),
```

```
('2022-10-15', '2025-11-23', 'Lumayan', 4, 5),
('2021-02-17', '2025-03-12', 'Rusak', 1, 3),
('2024-06-20', '2025-07-05', 'Baik', 10, 7),
('2025-03-21', '2025-04-09', 'Lumayan', 4, 3),
('2021-07-23', '2025-08-10', 'Baik', 5, 10);

INSERT INTO copies (dimpinjam, borrow_id,
book_id) VALUES
(1, 1, 1),
(1, 2, 2),
(1, 3, 3),
(1, 4, 4),
(1, 5, 7),
(1, 6, 4),
(1, 7, 1),
(1, 8, 10),
(1, 9, 4),
(1, 10, 5);


-- Update / Alter
ALTER TABLE borrow
ADD COLUMN kondisi_before
VARCHAR(50);

UPDATE borrow b
JOIN books bk ON b.book_id = bk.book_id
SET b.kondisi_before = bk.kondisi;

--tambahkan trigger untuk mengambil
kondisi_before untuk borrow dari books.kondisi
DELIMITER $$

CREATE TRIGGER set_kondisi_before
BEFORE INSERT ON borrow
FOR EACH ROW
BEGIN

DECLARE current_kondisi VARCHAR(50);

SELECT kondisi INTO current_kondisi
FROM books
WHERE book_id = NEW.book_id;
SET NEW.kondisi_before = current_kondisi;

END $$

DELIMITER ;


--buat copies menjadi lebih bermakna
ALTER TABLE copies
ADD COLUMN stock_before INT,
ADD COLUMN stock_after INT;


--Update sekali dulu agar datanya berubah
UPDATE copies c
JOIN books b ON c.book_id = b.book_id
SET
c.stock_before = b.stock,
c.stock_after = b.stock - c.dimpinjam;

--Pasang triger untuk kedepannya pengurangan
stock buku
DELIMITER $$

CREATE TRIGGER reduce_stock_on_borrow
AFTER INSERT ON borrow
FOR EACH ROW

BEGIN

UPDATE books
SET stock = stock - 1
WHERE book_id = NEW.book_id;

END $$

DELIMITER ;


-- Menyesuaikan stock dari perubahan stock di
copies
DELIMITER $$

CREATE TRIGGER copies_before_insert
```

```
BEFORE INSERT ON copies
FOR EACH ROW

BEGIN

DECLARE current_stock INT;

-- Ambil stok buku sebelum dipinjam
SELECT stock INTO current_stock
FROM books
WHERE book_id = NEW.book_id;

-- Set stock_before & stock_after otomatis
SET NEW.stock_before = current_stock;
SET NEW.stock_after = current_stock -
NEW.dimpinjam;

END $$

CREATE TRIGGER copies_after_insert
AFTER INSERT ON copies
FOR EACH ROW
BEGIN
-- Update stok buku agar mengikuti stock_after
(hasil peminjaman)
UPDATE books
SET stock = NEW.stock_after
WHERE book_id = NEW.book_id;
END $$

DELIMITER ;


-- Read

SELECT nama_panggilan
FROM users
WHERE provinsi = DIY;

SELECT book_id, judul, stock, kondisi
FROM books
WHERE bisa_dipinjam = 'ya' AND stock > 0;

SELECT b.borrow_id, b.issue_date, b.due_date,
bk.judul, b.kondisi_after
FROM borrow b
JOIN books bk ON b.book_id = bk.book_id
WHERE b.user_id = 3;

-- Delete
Delete from books_author
where book_id = 3;
```