

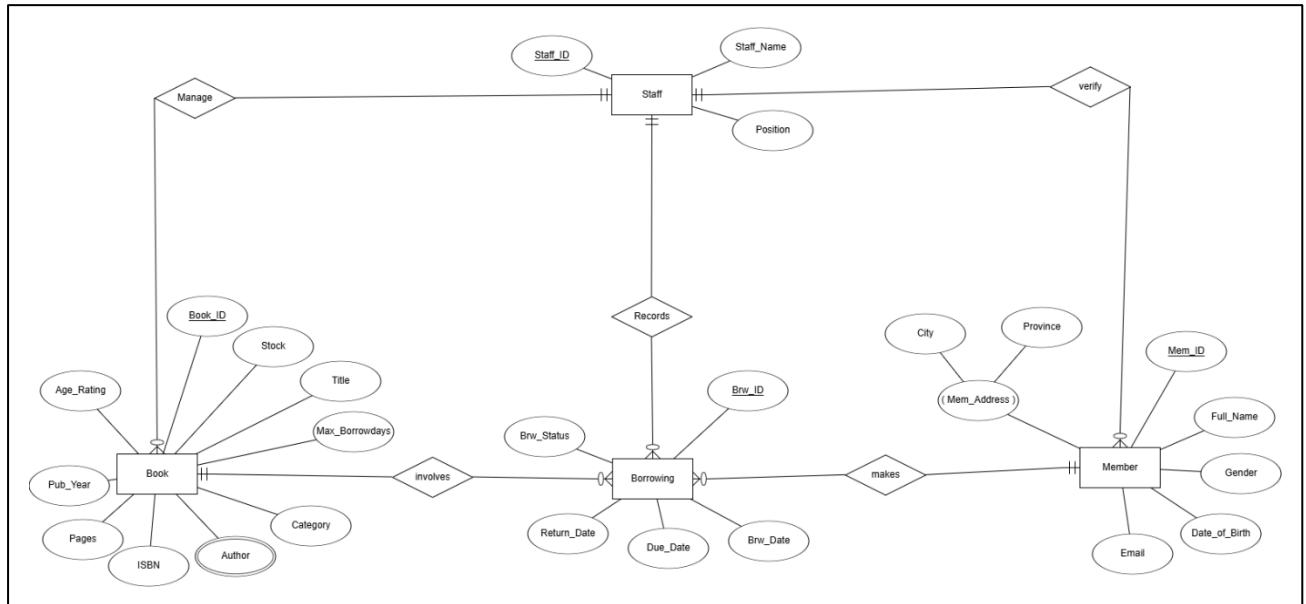
## PDF EXPLANATION

### Anggota Kelompok 06:

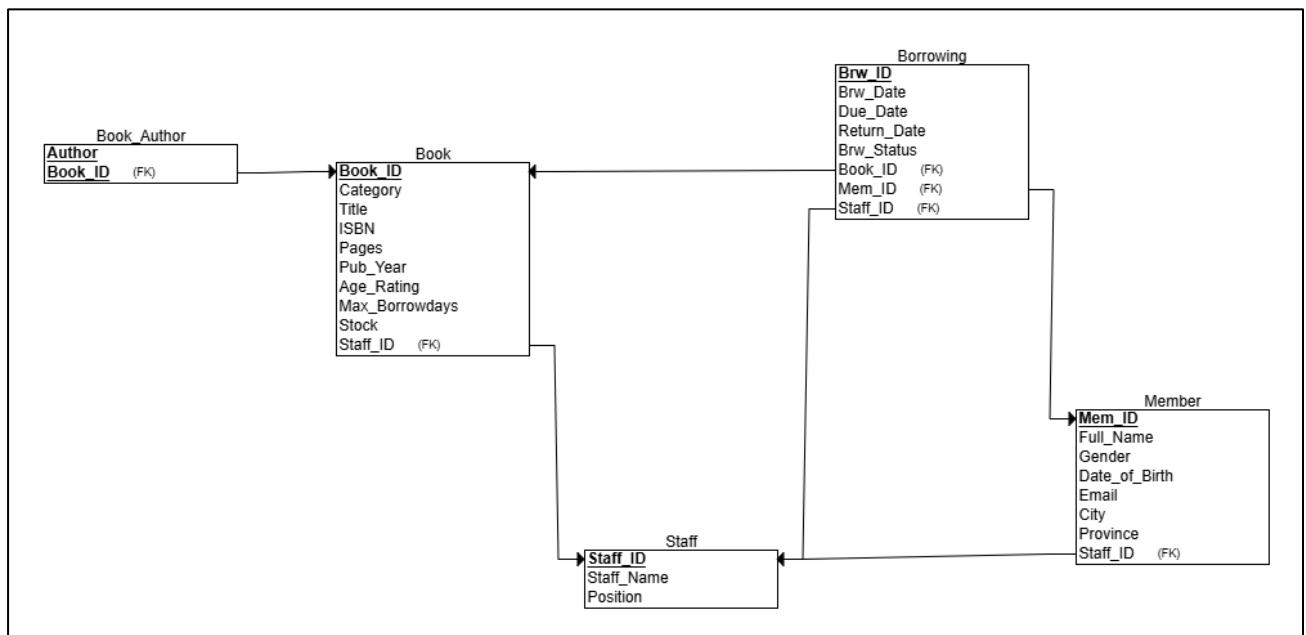
- 1) Kukuh Agus Hermawan (24/533395/PA/22573)
- 2) Aulia Fathus Tsani (24/534388/PA/22661)
- 3) Alifa Batrisyia Nariswari (24/535256/PA/22707)

### A. Convert ERD into a relational schema

**ERD:**



**Skema basisdata:**



**Penjelasan:****a. STAFF****Atribut:**

Staff\_ID (PK), Staff\_Name, Position

**Penjelasan:**

Tabel Staff menyimpan data petugas perpustakaan yang bertugas mengelola sistem InfiniRead.

Setiap staf memiliki ID unik sebagai primary key.

Mereka berperan dalam tiga hal utama:

1. Mengelola buku (melalui relasi dengan tabel *Book*),
2. Mencatat transaksi peminjaman (relasi dengan *Borrowing*), dan
3. Memverifikasi atau mendaftarkan anggota baru (relasi dengan *Member*).

Relasinya bersifat one-to-many (1:N) terhadap ketiga entitas tersebut, artinya satu staf dapat mengelola banyak buku, memproses banyak transaksi, dan memverifikasi banyak anggota.

**b. MEMBER****Atribut:**

Mem\_ID (PK), Full\_Name, Gender, Date\_of\_Birth, Email, City, Province, Staff\_ID (FK)

**Penjelasan:**

Tabel Member menyimpan informasi pengguna atau anggota yang terdaftar di sistem perpustakaan digital.

Atribut Staff\_ID (FK) menunjukkan petugas (staf) yang memverifikasi pendaftaran anggota tersebut.

Atribut alamat (City, Province) berasal dari pemisahan atribut komposit Mem\_Address agar data menjadi atomik sesuai 1NF.

Satu anggota dapat melakukan banyak transaksi peminjaman buku, yang direkam dalam tabel Borrowing melalui relasi one-to-many (1:N).

**c. BOOK****Atribut:**

Book\_ID (PK), Category, Title, ISBN, Pages, Pub\_Year, Age\_Rating, Max\_Borrowdays, Stock, Staff\_ID (FK)

**Penjelasan:**

Tabel Book menyimpan data koleksi buku yang tersedia di sistem InfiniRead. Foreign key Staff\_ID menghubungkan buku dengan petugas yang menginput atau mengelolanya.

Relasi one-to-many (1:N) antara *Staff* dan *Book* menunjukkan bahwa satu staf dapat mengelola banyak buku.

Selain itu, setiap buku dapat terlibat dalam banyak transaksi peminjaman di tabel Borrowing (1 Book → N Borrowing).

Atribut Author tidak disimpan langsung di tabel ini karena bersifat multivalued; oleh karena itu dipisahkan menjadi tabel Book\_Author.

**d. BORROWING****Atribut:**

Brw\_ID (PK), Brw\_Date, Due\_Date, Return\_Date, Brw\_Status, Book\_ID (FK), Mem\_ID (FK), Staff\_ID (FK)

**Penjelasan:**

Tabel Borrowing merepresentasikan setiap transaksi peminjaman buku yang terjadi di sistem.

Setiap record mencatat:

- Buku yang dipinjam (Book\_ID),
- Anggota yang meminjam (Mem\_ID),
- Petugas yang memproses transaksi (Staff\_ID),
- Tanggal peminjaman, batas waktu pengembalian, tanggal pengembalian aktual (bisa NULL bila belum dikembalikan),
- Dan status peminjaman (Borrowed atau Returned).

Relasinya bersifat many-to-one (N:1) ke *Staff*, *Member*, dan *Book*. Dengan demikian, setiap transaksi peminjaman hanya melibatkan satu petugas, satu anggota, dan satu buku.

**e. BOOK\_AUTHOR****Atribut:**

Author, Book\_ID (FK)

**Penjelasan:**

Tabel Book\_Author digunakan untuk menyimpan daftar penulis buku.

Satu buku dapat memiliki lebih dari satu penulis (relasi one-to-many (1:N) dari *Book* ke *Book\_Author*).

Primary key dari tabel ini adalah gabungan (Author, Book\_ID) agar setiap kombinasi penulis dan buku bersifat unik.

Pemisahan tabel ini dilakukan untuk memenuhi First Normal Form (1NF) karena atribut *Author* bersifat multivalued.

## B. Normalize up to 3NF

### 1. 1NF (First Normal Form)

#### Aturan:

- Setiap tabel harus memiliki primary key.
- Semua atribut harus berisi nilai tunggal (bersifat atomik).
- Tidak boleh ada grup berulang atau daftar nilai dalam satu kolom artinya setiap kolom hanya boleh menyimpan satu nilai untuk setiap baris (record).

#### Analisis:

Pada rancangan awal ERD sistem InfiniRead, terdapat:

- 1) Atribut Author pada entitas Book bersifat *multivalued*, karena satu buku bisa memiliki lebih dari satu penulis. Untuk memenuhi 1NF, atribut ini dipisahkan ke dalam tabel baru Book\_Author, dengan atribut Author dan Book\_ID sebagai komponennya.
- 2) Atribut Mem\_Address pada entitas Member bersifat *komposit* (terdiri dari City dan Province). Untuk membuat datanya atomik, atribut ini dipecah menjadi dua atribut terpisah, yaitu City dan Province.

#### Hasil:

Setelah pemisahan atribut *multivalued* dan *komposit*, semua tabel hanya berisi nilai tunggal di setiap kolom dan telah memiliki *primary key* masing-masing. Dengan demikian, seluruh tabel dalam sistem InfiniRead telah memenuhi 1NF.

### 2. 2NF (Second Normal Form)

#### Aturan:

- Tabel harus sudah memenuhi 1NF.
- Semua atribut non-key harus bergantung sepenuhnya pada *primary key* (tidak boleh ada *partial dependency*).

**Analisis:**

- 1) Tabel Staff, Member, Book, dan Borrowing memiliki *primary key tunggal* (Staff\_ID, Mem\_ID, Book\_ID, dan Brw\_ID), sehingga tidak mungkin terjadi ketergantungan parsial, karena setiap atribut non-key langsung bergantung pada satu primary key tersebut.
- 2) Tabel Book\_Author memiliki *primary key komposit* (Author, Book\_ID). Tidak terdapat atribut non-key lain selain komponen kunci tersebut, sehingga semua atribut bergantung sepenuhnya pada keduanya.

**Contoh:**

- Dalam tabel Borrowing, atribut seperti Brw\_Date, Due\_Date, dan Return\_Date bergantung langsung pada Brw\_ID sebagai primary key.
- Dalam tabel Book\_Author, setiap kombinasi (Author, Book\_ID) bersifat unik dan tidak menimbulkan ketergantungan sebagian terhadap salah satu komponen kunci.

**Hasil:**

Semua tabel di sistem InfiniRead telah memiliki ketergantungan penuh terhadap kunci primernya, sehingga telah memenuhi bentuk 2NF.

### 3. 3NF (Third Normal Form)

**Aturan:**

- Tabel harus sudah memenuhi 2NF.
- Tidak boleh ada *transitive dependency* artinya, atribut non-key tidak boleh bergantung pada atribut non-key lainnya.

**Analisis:**

- 1) Dalam tabel Member, seluruh atribut (Full\_Name, Gender, Date\_of\_Birth, Email, City, Province) bergantung langsung pada Mem\_ID sebagai primary key. Tidak ada atribut non-key yang bergantung pada atribut non-key lain seperti Staff\_ID.
- 2) Dalam tabel Book, atribut seperti Category, Title, ISBN, Pages, Pub\_Year, Age\_Rating, Max\_Borrowdays, dan Stock semuanya bergantung langsung pada Book\_ID dan tidak saling bergantung antaratribut.
- 3) Dalam tabel Borrowing, atribut Brw\_Date, Due\_Date, Return\_Date, dan Brw\_Status bergantung langsung pada Brw\_ID sebagai kunci utama, bukan pada atribut lain seperti Book\_ID atau Mem\_ID.
- 4) Dalam tabel Book\_Author, tidak ada atribut non-key, sehingga otomatis lolos 3NF.

**Hasil:**

Tidak ditemukan adanya *transitive dependency* di seluruh tabel.

Setiap atribut non-key hanya bergantung pada primary key masing-masing tabel.

Dengan demikian, struktur tabel sistem InfiniRead telah memenuhi Third Normal

Form (3NF) sepenuhnya.

**C. Implement with SQL ( CREATE TABLE , keys, constraints)**

```
CREATE TABLE Staff
(
    Staff_ID INT NOT NULL AUTO_INCREMENT,
    Staff_Name VARCHAR(100) NOT NULL,
    Position VARCHAR(50) NOT NULL,
    PRIMARY KEY (Staff_ID)
);

CREATE TABLE Member
(
    Mem_ID INT NOT NULL AUTO_INCREMENT,
    Full_Name VARCHAR(100) NOT NULL,
    Gender ENUM('Man', 'Woman') NOT NULL,
    Date_of_Birth DATE NOT NULL,
    Email VARCHAR(50) NOT NULL,
    City VARCHAR(50),
    Province VARCHAR(50),
    Staff_ID INT NOT NULL,
    PRIMARY KEY (Mem_ID),
    FOREIGN KEY (Staff_ID) REFERENCES Staff(Staff_ID)
);

CREATE TABLE Book
(
    Book_ID VARCHAR(20) NOT NULL,
    Category VARCHAR(20) NOT NULL,
```

```
Title VARCHAR(100) NOT NULL,  
ISBN VARCHAR(50) NOT NULL,  
Pages INT NOT NULL,  
Pub_Year INT NOT NULL,  
Age_Rating INT NOT NULL,  
Max_Borrowdays ENUM('3', '7', '14') NOT NULL,  
Stock INT NOT NULL,  
Staff_ID INT NOT NULL,  
PRIMARY KEY (Book_ID),  
FOREIGN KEY (Staff_ID) REFERENCES Staff(Staff_ID)  
);
```

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

```
CREATE TABLE Borrowing  
(  
Brw_ID INT NOT NULL AUTO_INCREMENT,  
Brw_Date DATE NOT NULL,  
Due_Date DATE NOT NULL,  
Return_Date DATE,  
Brw_Status ENUM('Borrowed', 'Returned') NOT NULL,  
Staff_ID INT NOT NULL,  
Mem_ID INT NOT NULL,  
Book_ID VARCHAR(20) NOT NULL,  
PRIMARY KEY (Brw_ID),  
FOREIGN KEY (Staff_ID) REFERENCES Staff(Staff_ID),  
FOREIGN KEY (Mem_ID) REFERENCES Member(Mem_ID),  
FOREIGN KEY (Book_ID) REFERENCES Book(Book_ID)
```

);

### **Penyesuaian tipe data dan constrain:**

- a. Atribut teks seperti nama, judul, kategori, posisi, alamat (City, Province), dan ID buku menggunakan VARCHAR agar dapat diisi dengan kombinasi huruf dan angka.
- b. Atribut tanggal seperti Brw\_Date, Due\_Date, Return\_Date, dan Date\_of\_Birth menggunakan DATE.
- c. Atribut status seperti Brw\_Status, Gender, dan Max\_Borrowdays menggunakan ENUM untuk membatasi nilai input hanya pada opsi tertentu yang diizinkan sistem.
- d. Atribut numerik seperti Pages, Pub\_Year, Age\_Rating, dan Stock menggunakan INT karena merepresentasikan angka bulat.
- e. Semua relasi antar tabel dijaga dengan FOREIGN KEY (...) REFERENCES ... untuk memastikan integritas data antar entitas tetap konsisten (Staff → Member/Book/Borrowing, Member → Borrowing, Book → Borrowing, Book → Book\_Author).