

**IF2140 - Pemodelan Basis Data Milestone 2**  
**Transformasi ER ke Diagram Relasional &**  
**Implementasi Basis Data**



**Kelompok 11**

**18221042 Ghaylan M. Fatih**

**18221058 Marvel Subekti**

**18221100 Vincent Franstyo**

**18221122 Reswara Trista Aulia C**

**Sistem dan Teknologi Informasi**  
**Sekolah Teknik Elektro dan**  
**Informatika**  
**Institut Teknologi Bandung**  
**2022**

## Deskripsi Sistem :

Perpustakaan merupakan tempat yang berfungsi untuk melakukan penyimpanan berbagai jenis media cetak yang diatur dan disusun sedemikian rupa sehingga mudah untuk dicari dan dipergunakan apabila sewaktu-waktu diperlukan oleh pembaca (Sutarno NS, 2006:11). Perpustakaan juga sering digunakan oleh masyarakat luas untuk ‘berselancar’ menambah ilmu pengetahuan melalui buku secara gratis. Perpustakaan itu sendiri memiliki sebuah layanan utama yaitu peminjaman buku. Oleh sebab itu, tidak heran apabila perpustakaan merupakan salah satu fasilitas umum yang sangat bermanfaat bagi setiap lapisan masyarakat.

Kami sebagai masyarakat yang sering menggunakan fasilitas umum ini, sangat tertarik untuk meninjau bagaimana sistem dan *workflow* dari perpustakaan khususnya perpustakaan nasional Indonesia dimana banyak sekali data buku serta peminjam yang harus disimpan dan tidak boleh *redundant* karena akan menyebabkan suatu permasalahan di kemudian hari. Selain itu, sistem peminjaman buku yang dibentuk haruslah memiliki data yang konsisten serta lengkap agar ketersediaan dan keberadaan buku bisa terus di *track* dan bisa dimanfaatkan kembali oleh peminjam-peminjam selanjutnya. Oleh karena itu data-data yang masif harus diintegrasikan kedalam suatu sistem basis data yang akan kami analisis dan tinjau serta akan direpresentasikan dalam bentuk *Entity Relationship Diagram*. Basis data yang akan kami analisis mencakup beberapa entitas seperti data akun, kartu perpustakaan, buku, peminjaman buku, kategori buku, reservasi buku, pertemanan, status akun, sinopsis buku, *review* buku, catatan pribadi, serta data riwayat peminjaman buku. Berikut penjelasannya lebih lanjut.

Untuk setiap pendataan suatu entitas peminjaman buku, akan disimpan satu data buku yang terdiri dari ID Buku yang unik untuk setiap unit buku, *file size* dari *soft copy* buku, jumlah buku yang merupakan jumlah total dari satu judul buku yang ada di dalam *database*, judul buku, penulis, penerbit, ISBN, e-ISBN, tahun rilis, penilaian, dan sinopsis dari buku tersebut.

Selain data buku, akan disimpan juga data dari akun *user* yang meminjam buku yang terdiri atas ID akun yang unik, *username* dari akun, *e-mail*, *password*, nomor telepon dimana satu akun dapat memiliki lebih dari satu nomor telepon, biodata dari setiap akun, serta status akun yang terdiri dari 2 jenis status yaitu *bookworm* dan *socializer*. Setiap akun pasti memiliki *bookworm* atau *socializer*.

Kedua status tersebut akan dipilih oleh pengguna saat pertama kali membuat akun Ipusnas, dimana perbedaannya terletak pada atribut yang dimiliki kedua status tersebut. Untuk status *bookworm* maka setiap pengguna bisa memilih avatar dengan tipe tumbuhan, data dari status *bookworm* ini akan dicatat pada entitas Status\_BookWorm, sementara untuk status *socializer* maka avatar yang dapat dipilih adalah avatar yang bertipe hewan dan data-nya akan dicatat pada entitas Status\_Socializer. Selain itu perbedaan dari keduanya adalah, untuk status *bookworm*, maka nanti akan ada fitur tambahan yaitu fitur rekomendasi buku. Sementara untuk status *socializer*, maka nanti akan ada fitur tambahan yaitu fitur rekomendasi teman. Perlu dicatat bahwa satu akun hanya mampu memiliki satu status saja.

Pada pendataan entitas peminjaman buku juga akan disimpan data peminjaman dengan isi entitas berupa ID peminjaman yang unik, tanggal data peminjaman dibuat, dan juga tanggal buku harus dikembalikan.

Dalam entitas peminjaman buku, tidak semua buku dipinjam dan satu buku hanya dapat dilakukan satu peminjaman oleh satu akun dalam satu waktu tertentu. Setiap peminjaman pasti memiliki buku yang dipinjam dan setiap buku dapat dilakukan banyak peminjam sesuai dengan ketersediaan buku.

Pada setiap entitas kategori buku, akan disimpan data dari buku dan jenis kategori buku yang ada di *database*, dengan isi entitas berupa ID kategori yang unik dan nama kategorinya. Dimana seluruh buku yang ada didalam *database* pasti memiliki kategori, dan bisa memiliki

lebih dari satu kategori. Selain itu, setiap kategori buku pasti memiliki setidaknya satu buku yang termasuk dalam kategori tersebut.

Pada entitas *reservation* akan dicatat data mengenai ID reservasi, tanggal reservasi dilakukan, dan status reservasi. Dalam *database* ini 1 reservasi akan digunakan untuk 1 *copy* buku dengan setiap reservasi harus memiliki 1 buku dan buku tidak harus memiliki reservasi tetapi satu buku bisa memiliki banyak reservasi. Kemudian Tidak semua akun harus melakukan reservasi buku atau bisa melakukan banyak reservasi tetapi setiap reservasi harus tepat berelasi dengan 1 akun.

Setiap peminjaman akan tercatat dalam entitas *history*. Entitas *history* sendiri merupakan *weak entity* dari entitas peminjaman. Setiap peminjaman yang sudah berlalu akan dicatatkan status pengembalian dan tanggal pengembaliannya serta tanggal dipinjamnya buku tersebut yang mengacu kepada atribut *lendingId* pada entitas *lending* serta berfungsi sebagai acuan untuk atribut *getReturned*. Pencatatan peminjaman ini melibatkan entitas *lending* karena berisi buku yang sedang dipinjam serta berelasi secara tidak langsung dengan entitas *book* yang berisi informasi buku yang dipinjam dalam peminjaman tersebut serta *account* yang melakukan peminjaman buku tersebut melalui entitas peminjaman.

Setiap buku memiliki kesempatan untuk di-*review* oleh pengguna, namun tidak semua buku memiliki *review*. Satu buku juga memiliki kesempatan untuk memiliki banyak *review*. Setiap *review* ini memiliki ID *review*, jumlah komentar, jumlah *love*, jumlah *report*, dan waktu *review* tersebut diberikan.

Kemudian fitur lain yang bisa dipakai oleh setiap pengguna akun iPusnas adalah *personal note*. *Personal note* ini nantinya akan berfungsi untuk memberikan kemudahan bagi pengguna agar mampu mencatat hal-hal penting yang dirasa perlu dicatat oleh mereka. Oleh sebab itu, data yang akan disimpan dalam hal ini adalah *noteID*, judul *note*, deskripsi / teks *note*, dan waktu dibuatnya *note* tersebut. Perlu diperhatikan bahwa *personal note* bersifat opsional sehingga tidak semua akun perlu memiliki *personal note* tetapi setiap *note* yang dibuat oleh pengguna pasti dimiliki oleh suatu akun tertentu.

Selanjutnya, ada *library card* yang berfungsi sebagai kartu identitas pengguna fasilitas perpustakaan. *Library card* ini nanti akan dipakai setiap kali pengguna ingin meminjam buku fisik di perpustakaan. Oleh karena, itu di dalam *library card* ini dibutuhkan data pribadi pengguna, diantaranya adalah nama lengkap, alamat, nomor kartu, nomor *barcode*, keterangan aktif atau tidaknya kartu perpustakaan, serta tanggal terbentuknya kartu tersebut (*issuedAt*). Perlu dicatat bahwa pembuatan kartu perpustakaan dilakukan pada tahap awal registrasi akun sehingga setiap pengguna pasti memiliki sebuah kartu perpustakaan untuk dirinya sendiri dan setiap kartu perpustakaan pasti dimiliki oleh seorang pengguna.

Kemudian yang terakhir adalah relasi pertemanan memungkinkan setiap pengguna mampu melakukan pertemanan dengan teman-temannya. Untuk setiap pertemanan akan dicatat setiap waktu pertemanan (*follow*) dilakukan. Tetapi, hal ini bersifat opsional, artinya seorang pengguna bisa saja tidak melakukan pertemanan dengan pengguna lainnya tetapi untuk satu pengguna bisa melakukan banyak pertemanan dengan *user* lain.



### A. Entity Menjadi Relasional Model :

Untuk melakukan pemetaan entitas *Account* ke model relasional maka hal yang dilakukan adalah memasukan atribut *accountId*, *username*, *email*, *password*, *accountBio*, *cardNumber* ke dalam skema *Account*. *Primary key* dari relasi ini adalah *accountId*. Atribut *phoneNumber* tidak dimasukan secara langsung kedalam skema tersebut karena bersifat *multivalued* sehingga harus dibuat sebuah tabel baru dengan *primary key* dari entitas *Account* dan *phoneNumber*. Selain itu untuk atribut *accountStatus* ditentukan oleh *specialization entity* yaitu status *socializer* dan status *bookworm* sehingga untuk *accountStatus* tidak perlu dimasukan kedalam atribut skema *Account*. Perlu diperhatikan juga bahwa *cardNumber* pada skema *Account* merupakan *foreign key* dari skema *LibraryCard*.

Skema *PersonalNote* dibangun oleh atribut-atribut lokalnya seperti *noteId*, *noteTitle*, *description*, *noteCreationTime*, *accountId*, dimana *noteId* merupakan *primary key* dari skema *PersonalNote* dan *accountId* menjadi *foreign key* yang bergantung pada entitas *Account*. Karena hubungan antara entitas *Account* dengan *PersonalNote* bersifat *one to many*, maka *primary key* dari entitas *Account* perlu ditambahkan ke skema *PersonalNote*.

3. LibraryCard = (cardNumber, issuedAt, isActive, barcode, namaLengkap, alamat)

Skema *LibraryCard* dibangun oleh atribut-atribut *cardNumber*, *issuedAt*, *isActive*, *barcode*, *namaLengkap*, *alamat*, dimana *cardNumber* merupakan *primary key* dari entitas *LibraryCard*. Dalam kasus ini relasi antara entitas *LibraryCard* dengan *Account* bersifat *one to one* dengan *total participation* artinya tiap pengguna pasti memiliki tepat 1 kartu perpustakaan dan 1 kartu perpustakaan pasti dimiliki oleh tepat 1 pengguna. Oleh sebab itu, *primary key* dari *LibraryCard* atau *Account* bisa dimasukkan kedalam salah satu skema diantara keduanya untuk menjadi *foreign key*. Dalam hal ini saya memasukan *primary key* entitas *LibraryCard* yaitu *cardNumber* kedalam skema *Account*.

4. Review = (reviewId, comment, love, report, bookId)

Relasi *review* dibangun dengan hubungan *many to one* dengan relasi *Book* karena akan dilakukan *review* pada buku dimana tidak semua buku memiliki *review*, tetapi seluruh *review* pasti dimiliki buku. Relasi ini terdiri dari *reviewId*, *comment*, *love*, *report*, dan menyimpan *bookId* dari buku yang sedang di-review juga. *Primary key* dari relasi ini adalah *reviewId* dengan *bookId* adalah *foreign key* terhadap *bookId* dari relasi *Book*.

5. Book = (bookId, fileSize, isbn, eisbn, rate)

Untuk melakukan pemetaan entitas *Book* ke model relasional maka hal yang dilakukan adalah mengisi relasi *Book* dengan atribut *bookId*, *fileSize*, *isbn*, *eisbn*, dan *rate*. *Primary key* dari entity ini adalah atribut *bookId* dan entity ini memiliki *foreign key* terhadap relasi *Bdescription*. Hal ini disebabkan karena setiap buku pada perpustakaan diidentifikasi oleh *bookId* sehingga untuk jenis dan judul buku yang sama bisa saja memiliki *bookId* yang berbeda oleh sebab itu relasi akan bergantung pada relasi *BDescription*.

6. BDescription = (isbn, eisbn, author, publisher, title, releaseYear, synopsis)

Relasi *BDescription* terdiri dari atribut *isbn* dan *eisbn* sebagai *primary key*, serta atribut *author*, *publisher*, *title*, *releaseYear*, dan juga *synopsis*. Relasi ini berfungsi sebagai relasi yang mendeskripsikan informasi lengkap suatu buku oleh sebab itu pada tabel *book* memiliki *foreign key* ke atribut *isbn* dan *eisbn* pada relasi *Bdescription* ini

7. Category = (categoryId, categoryName)

Relasi *Category* terdiri dari atribut *categoryId* sebagai *Primary key* dan atribut *categoryName*. Pada relasi ini, isi atribut tidak mencapai 150 karena kategori buku pada umumnya hanya beberapa jenis saja. Relasi ini tidak memiliki *foreign key*.

8. Lending = (lendingId, creationDate, dueDate, bookId, accountId)

Relasi *Lending* terdiri dari atribut *lendingId*, *creationDate*, *dueDate*, *bookId*, *accountId* dengan *lendingId* sebagai *primary key*. Relasi ini berrelasi *one-to-many* dengan *Book* dan *Account*. Setiap *lendingId* hanya memiliki satu *bookId* dan *accountId*, tetapi satu *accountId* atau *bookId* bisa memiliki lebih dari satu *lendingId*. Relasi ini memiliki *foreign key* *bookId* terhadap relasi *Book* dan *accountId* terhadap relasi *Account*.

9. Reservation = (reservationId, reservationCreationDate, reservationStatus, bookId, accountId)

Relasi *Reservation* terdiri dari atribut *reservationId*, *reservationCreationDate*, *reservationStatus*, *bookId*, *accountId* dengan *reservationId* sebagai *primary key*. Relasi ini berrelasi *one-to-many* dengan *Book* dan *Account*. Setiap *reservationId* hanya memiliki satu

*bookId* dan *accountId*, tetapi satu *accountId* atau *bookId* bisa memiliki lebih dari satu *reservationId*. Relasi ini memiliki *foreign key bookId* terhadap relasi *Book* dan *accountId* terhadap relasi *Account*.

## **B. Relation Menjadi Relasional Model :**

1. Friendship = ( accountId , followerId , followedTime )

Skema *Friendship* timbul akibat relasi *many to many* pada entitas *Account*, dimana atributnya bergantung pada relasi antar *accountId* pengguna perpustakaan. Hal ini terjadi karena tiap pengguna iPusnas bisa memiliki banyak teman atau bahkan tidak memiliki teman sama sekali. Oleh sebab itu, dibutuhkan suatu tabel/skema baru yang memiliki atribut *primary key* dari relasi *Account*, kemudian atribut *accountId* pengguna lain yang diberi nama *followerId*. Perlu diperhatikan bahwa atribut *followerId* juga perlu menjadi *primary key* di skema ini karena sifatnya yang unik. Terakhir, atribut dari *followedTime* juga perlu ditambahkan yang menunjukkan waktu dilakukannya pertemanan oleh *user* dengan *accountId* yang bersangkutan.

2. BookCategory = ( categoryId , bookId )

Relasi *BookCategory* merupakan relasi yang terbentuk karena relasi *many-to-many* entitas *Book* dengan *Category*. Relasi ini terbentuk dengan isi atribut *bookId* yang merupakan *foreign key* dari relasi *Book* dan *categoryId* yang merupakan *foreign key* dari relasi *Category*.

## **C. Spesialization / Generalization Menjadi Relasional Model :**

1. Status\_BookWorm = ( accountId , plantAvatar )

Skema *Status\_BookWorm* dibangun oleh atribut *accountId* dan *plantAvatar* yang dimana *accountId* adalah *primary key* dari entitas *Status\_BookWorm*. Dalam kasus ini relasi antara *Account* dengan *Status\_BookWorm* adalah *specialization* dengan *disjoint* dan *total participation*, sehingga setiap *accountId* hanya bisa memiliki satu status dan *primary key* dari *Account(accountId)* ditambahkan ke skema *Status\_BookWorm*.

2. Status\_Socializer = ( accountId , animalAvatar )

Skema *Status\_Socializer* dibangun oleh atribut *accountId* dan *animalAvatar* yang dimana *accountId* adalah *primary key* dari skema *Status\_Socializer*. Dalam kasus ini relasi antara *Account* dengan *Status\_Socializer* adalah *specialization* dengan *disjoint* dan *total participation*, sehingga setiap *accountId* hanya bisa memiliki satu status dan *primary key* dari *Account(accountId)* ditambahkan ke skema *Status\_Socializer*.

## **D. Week Entity Menjadi Relasional Model :**

1. History = ( lendingId , dateReturned , getReturned )

Entitas *history* adalah *weak entity* dari entitas *lending* karena ia mencatat pengembalian sebuah buku yang telah tercatat dalam entitas *lending*. Karena ia adalah *weak entity*, ia berhubungan secara *many-to-one* kepada entitas *lending*. Pada *weak entity history* ini terkandung *dateReturned* dan *getReturned*. *dateReturned* akan bernilai *NULL* jika *getReturned* bernilai *FALSE*. Pada relasi *history* ini, ia memiliki diskriminator, yaitu *dateReturned* dan *primary key* yang dimiliki adalah *lendingId*.

### E. Multivalued Atribut Menjadi Relational Model :

1. Account\_PhoneNumber = ( accountId , phoneNumber )

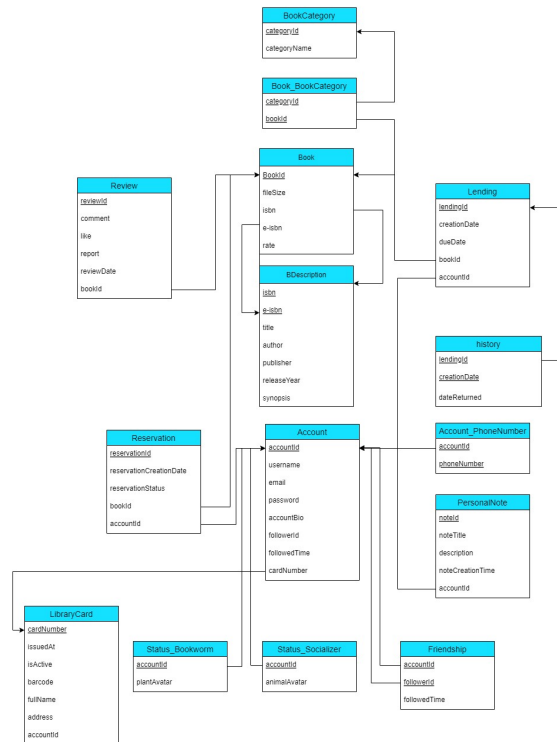
Berdasarkan keterangan dari langkah pemetaan entitas *Account* pada nomor 1, maka *Account\_PhoneNumber* merupakan skema yang muncul karena nilai dari atribut *phoneNumber* yang *multivalued*. Oleh sebab itu, diperlukan suatu skema/tabel baru yang atributnya merupakan gabungan dari *primary key* entitas *Account* yaitu *accountId* dengan *phoneNumber* itu sendiri yang juga turut menjadi *primary key* di skema/tabel tersebut.

### F. Derived Atribut Menjadi Relational Model :

Dalam melakukan transformasi derived atribut menjadi *relational model*, dibutuhkan atribut pendukung dari entitas yang sama. Dalam sistem iPusnas yang kami tinjau, kami memiliki 1 *derived attribute* yang berada pada *weak entity* yang bernama *history*. *Derived attribute* ada pada entitas ini adalah *getReturned* , dimana atribut ini bergantung pada atribut *lendingId* yang menjadi *primary key* sekaligus menjadi atribut *foreign key* yang mengacu kepada entitas *lending*. Selain itu, atribut *getReturned* juga dipengaruhi secara langsung oleh adanya atribut *dateReturned*. *dateReturned* akan bernilai tidak *NULL* apabila buku telah dikembalikan oleh peminjam buku di suatu waktu tertentu. Apabila *dateReturned* bernilai *NULL*, maka nilai dari *getReturned* akan menjadi *FALSE*, sementara apabila *dateReturned* tidak bernilai *NULL*, maka nilai dari *getReturned* akan menjadi *TRUE*. Berikut contohnya :

	<u>lendingId</u>	<u>dateReturned</u>	<u>getReturned</u>
▶	1	2020-10-13	TRUE
	2	2022-04-17	TRUE
	3	2022-07-27	TRUE
	4	2021-12-06	TRUE
	5	2022-08-19	TRUE

## GAMBAR DIAGRAM RELATIONAL MODEL



*gambar 1. 2. Relational Schema Diagram Ipusnas*



# Daftar relasi beserta penjelasan dan contoh tuple dalam relasi tersebut

## 1. Relasi Account

Account = (accountId, username, email, password, accountBio, cardNumber)

FK:

- Account(cardNumber) → LibraryCard(cardNumber)

KETERANGAN :

- AccountId tiap pengguna unik dengan format “acc” dan diikuti dengan 3 angka random dibelakangnya
- Card Number merupakan foreign key dari atribut Card Number di entitas Library Card

	accountId	username	email	password	accountBio	cardNumber
▶	acc001	lattenborough0	bdeleen0@va.gov	Q4HXIX	Duis bibendum, felis sed interdum venenatis, tu...	89931080721
	acc002	rblankley1	pdzenisenka1@newsvine.com	7LamHZX5t	Aliquam quis turpis eget elit sodales scelerisque....	9733300459
	acc003	emacartney2	aravelus2@sbwire.com	1yyAAYY	Integer ac leo. Pellentesque ultrices mattis odio...	35155267380
	acc004	pspear3	vkinsett3@pinterest.com	Cc4EXJBR97M	In sagittis dui vel nisl. Duis ac nibh. Fusce lacus ...	47931504909
	acc005	zwaddy4	yheddon4@a8.net	z0jGhdm2xcxv	Nulla ut erat id mauris vulputate elementum. Nul...	58213497782

## 2. Relasi Account\_PhoneNumber

Account\_PhoneNumber = (accountId, phoneNumber)

FK :

- Account\_PhoneNumber(accountId) → Account (accountId)

KETERANGAN :

- 1 Account yang sama bisa memiliki banyak nomor telepon

```
MariaDB [perpusnas]> select * from account_phonenumber limit 5;
+-----+-----+
| accountId | phoneNumber |
+-----+-----+
| acc001    | 6208077618936 |
| acc001    | 6289772367350 |
| acc002    | 6239631025197 |
| acc002    | 6287119968536 |
| acc003    | 6240668094655 |
+-----+-----+
5 rows in set (0.001 sec)
```

## 3. Relasi Friendship

Friendship = (accountId, followerId, followedTime)

FK :

- Friendship(accountId) → Account(accountId)
- Friendship(followerId) → Account(accountId)

KETERANGAN :

- 1 Akun bisa memiliki banyak *follower* dari akun yang berbeda

```
MariaDB [perpusnas]> select*from friendship limit 5;
```

accountId	followerId	followedTime
acc001	acc283	2022-08-29
acc002	acc061	2021-10-09
acc002	acc124	2021-12-16
acc004	acc001	2021-08-05
acc004	acc138	2021-03-29

```
5 rows in set (0.001 sec)
```

#### 4. Relasi LibraryCard

LibraryCard = (cardNumber, issuedAt, isActive, barcode, namaLengkap, alamat)

KETERANGAN :

- cardNumber setiap pengguna bersifat unik karena berfungsi untuk menyimpan informasi kartu perpustakaan setiap pengguna. Oleh sebab itu, satu pengguna hanya bisa untuk memiliki 1 kartu perpustakaan.

```
MariaDB [perpusnas]> select * from librarycard limit 5;
```

cardNumber	issuedAt	barcode	fullName	address
10166086739	2021-11-22	59315087340	Chester Sonnenschein	0 Hoffman Hill
1040283672	2020-10-31	14233745503	Inger Purnell	743 Golden Leaf Trail
10466608449	2022-04-29	97072209212	Maurie Shewring	88009 Manley Hill
11034142345	2021-10-20	71952591436	Eran Dimond	51 Derek Circle
1159897183	2021-12-06	62255527352	Lavinia Dillestone	77 Division Alley

```
5 rows in set (0.001 sec)
```

#### 5. Relasi PersonalNote

PersonalNote = (noteId, noteCreationTime, noteTitle, description, accountId)

FK:

- PersonalNote(accountId) → Account(accountId)

KETERANGAN :

- Setiap pengguna bisa memiliki beberapa catatan yang berbeda-beda. Oleh sebab itu, satu pengguna bisa saja memiliki satu atau lebih data catatan berbeda atau sama sekali tidak memiliki catatan didalam aplikasi iPusnas.

	noteId	noteTitle	description	noteCreationTime	accountId
►	0J94	maecenas ut massa quis augue	Integer ac leo. Pellentesque ultrices mattis odio...	2021-08-25	acc128
	0r83	sapien dignissim vestibulum	Fusce consequat. Nulla nisl. Nunc nisl. Duis bib...	2020-09-30	acc231
	0R93	metus sapien ut nunc vestibulum	In quis justo. Maecenas rhoncus aliquam lacus. ...	2022-08-16	acc295
	0w48	nulla nisl nunc nisl	Donec diam neque, vestibulum eget, vulputate ...	2022-10-15	acc226
	0Y65	quam sapien varius ut	In hac habitasse platea dictumst. Etiam faucibu...	2022-08-14	acc292

## 6.Relasi Status\_BookWorm

Status\_BookWorm = (accountId, plantAvatar)

FK :

- Status\_BookWorm(accountId) → Account(accountId)

KETERANGAN :

- Setiap pengguna hanya bisa menggunakan 1 *avatar* bertipe tumbuhan dalam waktu yang sama.

```
mysql> select * from status_bookworm limit 5;
+-----+-----+
| accountId | plantAvatar |
+-----+-----+
| acc001    | Strigula Lichen |
| acc002    | Mexican Saucerflower |
| acc003    | Mountain Fetterbush |
| acc004    | Ione Manzanita |
| acc005    | Montagne's Roccella Lichen |
+-----+-----+
5 rows in set (0.00 sec)
```

## 7.Relasi Status\_Socializer

Status\_Socializer = (accountId, animalAvatar)

FK :

- Status\_Socializer(accountId) → Account(accountId)

KETERANGAN :

- Setiap pengguna hanya bisa menggunakan 1 *avatar* bertipe hewan dalam waktu yang sama.

```
mysql> select * from status_socializer limit 5;
+-----+-----+
| accountId | animalAvatar |
+-----+-----+
| acc151    | Sloth, hoffman's |
| acc152    | Springbok      |
| acc153    | Common turkey  |
| acc154    | Kangaroo, jungle |
| acc155    | Badger, eurasian |
+-----+-----+
5 rows in set (0.00 sec)
```

## 8. Relasi History

History = (lendingId, dateReturned, getReturned)

FK:

History(lendingId) → Lending(lendingId)

KETERANGAN:

- *dateReturned* akan bernilai *NULL* jika *getReturned* bernilai *FALSE*

	<u>lendingId</u>	dateReturned	getReturned
▶	1	2020-10-13	TRUE
	2	2022-04-17	TRUE
	3	2022-07-27	TRUE
	4	2021-12-06	TRUE
	5	2022-08-19	TRUE

## 9. Relasi Review

Review = (reviewId, comment, love, report, bookId)

FK:

Review(bookId) → Book(bookId)

KETERANGAN:

- Tidak semua buku memiliki *review*, tetapi semua *review* pasti dimiliki oleh suatu buku.

```
mysql> select * from review limit 5;
+-----+-----+-----+-----+-----+
| bookId | reviewId | love | comment | report |
+-----+-----+-----+-----+-----+
| ID228  | acn5222  | 108  | 98      | 0      |
| ID231  | aek7643  | 109  | 78      | 0      |
| ID226  | aie1236  | 59   | 126     | 2      |
| ID149  | ajf6499  | 107  | 68      | 0      |
| ID179  | akv4698  | 52   | 117     | 2      |
+-----+-----+-----+-----+-----+
5 rows in set (0.00 sec)
```

## 10. Relasi Book

Book = (bookId, fileSize, isbn, eisbn, rate, synopsis)

FK: -

	bookId	fileSize	isbn	eisbn	rate	synopsis
►	ID001	49	341234703-5	005300415-9	3	Maecenas rhoncus aliquam lacus. Morbi quis tor...
	ID002	11	112992711-3	005916589-8	1	Curabitur convallis. Duis consequat dui nec nisi ...
	ID003	19	881574768-0	006896362-9	3	Fusce consequat. Nulla nisl. Nunc nisl. Duis bibe...
	ID004	32	280664525-5	007378086-3	1	Morbi porttitor lorem id ligula. Suspendisse orna...
	ID005	39	533154244-1	007500473-9	3	Donec odio justo, sollicitudin ut, suscipit a, feugi...

## 11. Relasi BDescription

BDescription = (isbn, eisbn, title, author, publisher, releaseYear, synopsis)

FK: -

	isbn	eisbn	author	publisher	title	releaseYear	synopsis
►	001612218-6	936470007-4	Howard Clashe	Kassulke and Sons	Sune på bilsemester	2008	In hac habitasse
	003051415-0	544492126-X	Stephen Piwell	Ernser-Blick	Modigliani	2005	In blandit ultrices
	012032840-2	708854002-9	Hortensia Stuckow	Mayer and Sons	Idle Hands	2009	In congue. Etiam
	012519825-6	581435380-5	Georgi Fardy	Balistreri Inc	Country	2008	Nam nulla. Integi
	014547811-4	120976899-2	Andris Sutton	Padberg-Grady	Land of the Pharaohs	1994	Nulla neque liber

## 12. Relasi Category

Category = (categoryId, categoryName)

FK: -

KETERANGAN :

- Kategori buku tidak memungkinkan untuk mendapatkan 150 data karena keterbatasan *source* sehingga untuk relasi ini kami menaruh sekitar 13 data buku dengan nama dan id yang masing-masing berbeda .

```
[mysql> select * from bookCategory limit 5;
```

categoryId	categoryName
CAT001	Fiction
CAT002	Non fiction
CAT003	Horror
CAT004	Comic
CAT005	Fable

```
5 rows in set (0.01 sec)
```

### 13. Relasi BookCategory

BookCategory = (categoryId, bookId)

FK:

BookCategory(bookId) → Book(bookId)

BookCategory(categoryId) → Category(categoryId)

```
[mysql> select * from book_bookCategory limit 5;
```

bookId	categoryId
ID023	CAT001
ID101	CAT001
ID102	CAT001
ID111	CAT001
ID201	CAT001

```
5 rows in set (0.00 sec)
```

### 14. Relasi Lending

Lending = (lendingId, creationDate, dueDate, bookId, accountId)

FK:

Lending(bookId) → Book(bookId)

Lending(accountId) → Account(accountId)

```
MariaDB [perpusnas]> select*from lending limit 5;
```

lendingId	creationDate	dueDate	bookID	accountId
1	2020-10-10	2020-10-17	ID001	acc001
2	2022-04-14	2022-04-21	ID002	acc002
3	2022-07-24	2022-07-31	ID003	acc003
4	2021-12-03	2021-12-10	ID004	acc004
5	2022-08-16	2022-08-23	ID005	acc005

```
5 rows in set (0.017 sec)
```

## 15. Relasi Reservation

Reservation = (reservationId, reservationCreationDate, reservationStatus, bookId, accountId)

FK:

Reservation(bookId) → Book(bookId)

Reservation(accountId) → Account(accountId)

```
MariaDB [perpusnas]> select*from reservation limit 5;
```

reservationId	reservationCreationDate	reservationStatus	bookId	accountId
1	2020-10-03	Ready	ID001	acc051
2	2022-04-07	Ready	ID002	acc259
3	2022-07-17	Ready	ID003	acc155
4	2021-11-26	Ready	ID004	acc050
5	2022-08-09	Ready	ID005	acc261

```
5 rows in set (0.010 sec)
```

# DAFTAR RELATIONAL SCHEMA DAN FOREIGN KEY

## 1.ACCOUNT :

LibraryCard = (cardNumber, issuedAt, isActive, barcode, namaLengkap, alamat)

PersonalNote = (noteId, noteCreationTime, noteTitle, description, accountId)

Account = (accountId, username, email, password, accountBio, cardNumber)

Account\_PhoneNumber = (accountId, phoneNumber)

Friendship = (accountId, followerId, followedTime)

## 2.ACCOUNT STATUS:

Status\_Bookworm = (accountId, plantAvatar)

Status\_Socializer = (accountId , animalAvatar)

## 3.LENDING :

Lending = (lendingId, creationDate, dueDate, accountId, bookId)

History = (lendingId, dateReturned, getReturned)

## 4.BOOK :

Book = (bookId, fileSize, isbn, eisbn, rate)

BDescription = (isbn, eisbn, title, author, publisher, releaseYear, synopsis)

Review = (reviewId, comment, love, report, bookId)

## 5.BOOK CATEGORY :

Category = (categoryId, categoryName)

BookCategory = (categoryId, bookId)

## 6. RESERVATION :

Reservation = (reservationId, reservationCreationDate, reservationStatus, accountId, bookId)

## FOREIGN KEY

### 1. ACCOUNT :

Account\_PhoneNumber(accountId) → Account (accountId)

Account(cardNumber) → LibraryCard(cardNumber)

PersonalNote(accountId) → Account(accountId)

Friendship(accountId) → Account(accountId)

### 2. STATUS :

Status\_Bookworm(accountId) → Account(accountId)

Status\_Socializer(accountId) → Account(accountId)

### 3. LENDING :

History(lendingId) → Lending(lendingId)

Lending(bookId) → Book(bookId)

Lending(accountId) → Account(accountId)

### 4. BOOK :

Review(bookId) → Book(bookId)

Book(isbn, eisbn) → BDescription(isbn, eisbn)



**5. BOOK CATEGORY :**

BookCategory(bookId) → Book(bookId)

BookCategory(categoryId) → Category(categoryId)

**6. RESERVATION :**

Reservation(bookId) → Book(bookId)

Reservation(accountId) → Account(accountId)

## Daftar Pustaka

*Csv to Sql – Convertcsv, n.d.;*

*Foreign Key Relationship with Composite Primary Keys in SQL Server 2005 - Stack Overflow, n.d.;*

*How Can I Find out What FOREIGN KEY Constraint References a Table in SQL Server? - Stack Overflow, n.d.;*

*Mockaroo - Random Data Generator and API Mocking Tool | JSON / CSV / SQL / Excel, n.d.;*

*Mysql - Error Code: 1062. Duplicate Entry “1” for Key “PRIMARY” - Stack Overflow, n.d.;*

*Mysql - SQL Keys, MUL vs PRI vs UNI - Stack Overflow, n.d.; Primary Key – MySQL*

*Duplicate Entry Error Even Though There Is No Duplicate Entry - Stack Overflow, n.d.;*

*[Solved] How to Solve MySQL Error Code: 1062 Duplicate Entry? ~ Technology Blog by*

*Rathish Kumar, n.d.; View Foreign Key Properties - SQL Server | Microsoft Learn, n.d.;*

*W3Schools Online Web Tutorials, n.d.*

*<https://dev.mysql.com/doc/refman/8.0/en/insert.html>*

## List Pembagian Tugas

Nama	Kontribusi / Tugas
Marvel Subekti	<ul style="list-style-type: none"> <li>- Membuat Relation Schema</li> <li>- Merevisi Relation Schema , ER Diagram , dan Deskripsi Sistem</li> <li>- Membuat Dataset Data Account , Account_PhoneNumber , Friendship , PersonalNote , dan LibraryCard</li> <li>- Mengkontak Asisten untuk menentukan jadwal asistensi</li> <li>- Melakukan Revisi , Manipulasi , dan Update semua dataset agar sesuai dengan spesifikasi tugas</li> <li>- Membuat database dengan mysql</li> <li>- Melakukan dump file</li> <li>- Memberikan Deskripsi sistem , penjelasan relasi , dan contoh tuple relasi Account , Account_PhoneNumber , Friendship , PersonalNote , dan LibraryCard</li> </ul>
Reswara Trista Aulia C.	<ul style="list-style-type: none"> <li>- Revisi dan saran skema relasional</li> <li>- Revisi relasi redundant</li> <li>- Membuat diagram skema relasional</li> <li>- Membuat dataset untuk relasiBook, BookCategory, Category, lending, reservation</li> <li>- Membuat langkah-langkah pemetaan untuk relasi Book, BookCategory, Category, lending, reservation</li> <li>- Membuat deskripsi sistem, penjelasan relasi, dan contoh tuple relasi Book, BookCategory, Category, lending, reservation</li> <li>- Membuat notulensi asistensi</li> </ul>
Ghaylan M. Fatih	<ul style="list-style-type: none"> <li>- Membuat dummy data untuk Status_BookWorm, Status_Socializer</li> <li>- Membuat langkah-langkah pemetaan untuk Status_BookWorm, Status_Socializer</li> <li>- Membuat revisi untuk Status_BookWorm, Status_Socializer</li> <li>- Memberikan Deskripsi sistem , penjelasan relasi , dan contoh tuple relasi Status_BookWorm, Status_Socializer.</li> </ul>
Vincent Franstyo	<ul style="list-style-type: none"> <li>- Membuat Cover Laporan</li> <li>- Membuat dummy data untuk Review dan History</li> <li>- Memperbaiki seluruh penggunaan kalimat atau kata yang kurang tepat dan melakukan pengecekan ulang terhadap tanda baca, huruf miring, dan kata-kata yang digunakan.</li> <li>- Membuat langkah-langkah pemetaan untuk Review dan History</li> <li>- Membuat deskripsi sistem, penjelasan relasi, dan contoh tuple relasi Review dan History.</li> <li>- Memperbaiki penulisan dalam daftar pustaka.</li> </ul>

## Moments of Meeting

### Tugas Besar IF2140 Pemodelan Basis Data

<b>Hari</b>	Sabtu	<b>Tanggal</b>	5/11/2022
<b>Kelas</b>	02	<b>Kelompok</b>	11
<b>Waktu</b>	<20.00-20.30>	<b>Tempat</b>	Zoom
<b>Anggota Kelompok</b>	<b>NIM</b>	<b>Nama</b>	
	18221042	Ghaylan Muhammad Fatih	
	18221058	Marvel Subekti	
	18221100	Vincent Franstyo	
	18221122	Reswara Trista Aulia Candrakanti	
<b>Milestone</b>	Milestone 2		
<b>Nama Asisten</b>	Dewa Ayu Mutiara Kirana P D		

Hasil Asistensi		
No.	Pertanyaan	Jawaban
1.	Revisi dari Milestone 2 ?	1.Tambahin penjelasan mengenai transformasi derived atribut 2.Transformasi ke relational model lebh baik dikelompokan 3.Tambahin List Relational skema beserta list foreign keynya
2	Transformasi Relational Model perlu di kasih gambarnya ga ?	Opsional

3.	<b>File Revisi milestone 1 apakah boleh mengambil deskripsi sistem dan ER dari laporan milestone 2 ?</b>	<b>Diperbolehkan karena memang diambil dari milestone 2</b>
4	<b>Untuk pengumpulan milestone 2 apakah hanya Laporan milestone 2 , file sql dump , dan laporan revisi milestone 1 ?</b>	<b>Iya</b>
5	<b>Kalau ada kata disadur dalam bahasa inggris , apakah harus di italic ? dan apakah semua paragrafnya harus di justify ?</b>	<b>Dianjurkan menggunakan italic supaya laporannya rapih dan untuk text paragraf sebaiknya menggunakan justify juga biar rapih</b>

### Dokumentasi

