

FINAL PROJECT AoL
DATA AND INFORMATION MANAGEMENT
“SISTEM INFORMASI MANAJEMEN BIOSKOP”

Class: LI11



Lecturer Name:

Harkat Christian Zamasi, S.Kom., MMSI

Group J:

2440007081 - Ewaldo Samuel

2440010076 - Emily Indrakusuma

2440012535 - Andrew Duarte Winoto

2440039051 - Gabriela Azzahra Ferdy

2440068972 - Adham Miftah

Daftar Isi

CHAPTER 1: <i>Company Profile</i>	1
CHAPTER 2: <i>Company Business Process</i>	3
CHAPTER 3: <i>Problem Statements</i>	4
CHAPTER 4: <i>Database Planning</i>	5
4. 1. <i>Mission Statement</i>	5
4. 2. <i>Mission Objective</i>	5
CHAPTER 5: <i>System Definition</i>	7
CHAPTER 6: <i>Requirement Collection and Analysis</i>	8
CHAPTER 7: <i>Database Design</i>	11
7. 1. <i>Conceptual Design</i>	11
7. 2. <i>Arcs, Hierarchies, and Recursive Modeling</i>	22
7. 3. <i>Changes and Historical Modeling</i>	24
7. 4. <i>Logical Design</i>	24
7. 5. <i>Normalization</i>	31
7. 6. <i>Physical Design Database</i>	41
7.6.1. Step 3: Translate Logical Data Model for Target DBMS	41
7.6.2. Step 4: Design File Organizations and indexes	54
7.6.3. Step 5: Design User View	59
7.6.1. Step 6: Design Security Mechanisms	66
CHAPTER 7: <i>User Interface</i>	68

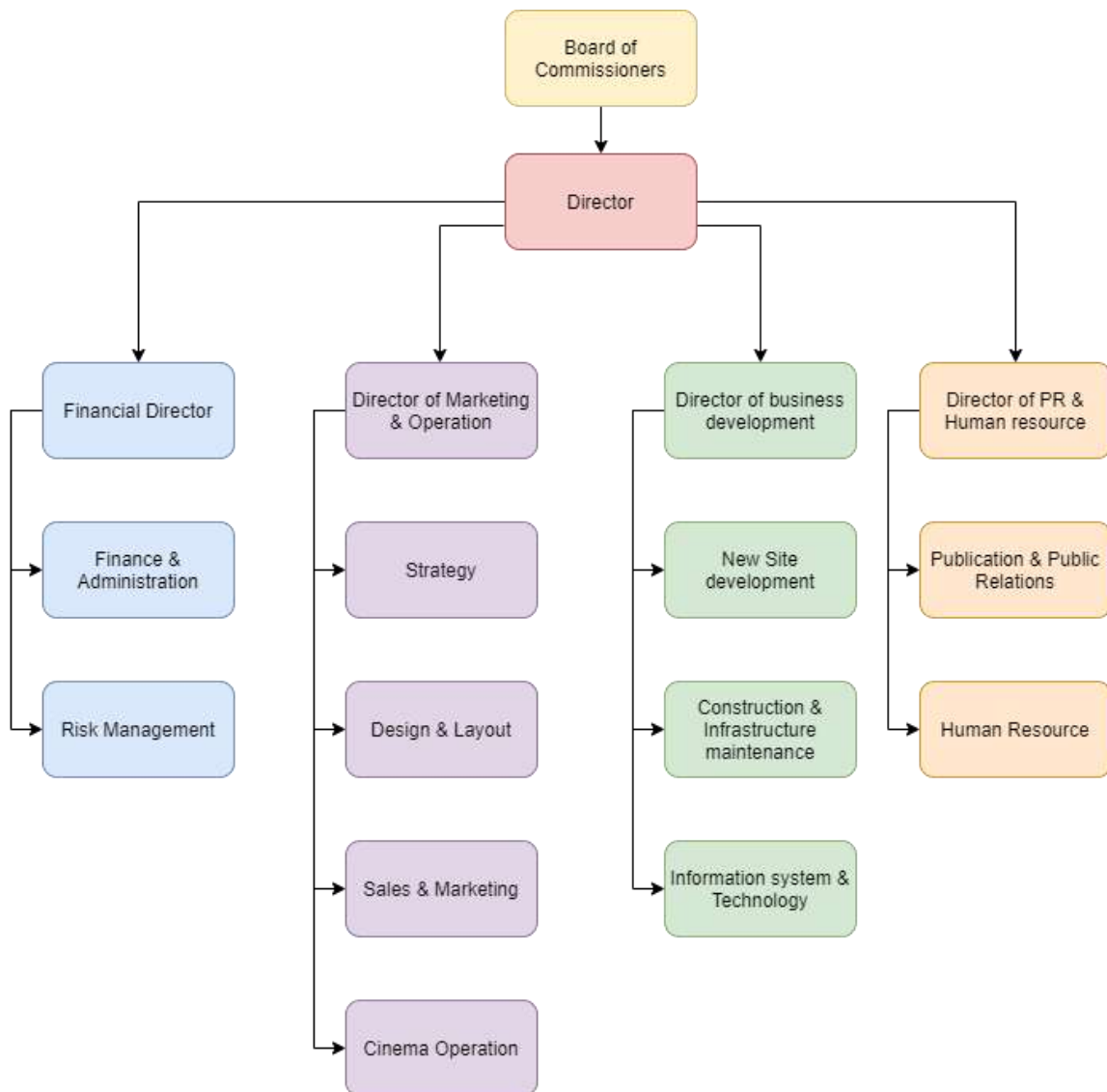
CHAPTER 1: *Company Profile*

ABC Cinema adalah sebuah perusahaan yang bergerak pada bisnis bioskop yang memelopori jaringan cineplex di seluruh Indonesia. ABC Cinema telah berdiri kurang lebih selama 10 tahun dari tahun 2011. ABC Cinema sekarang ini memiliki 35 cabang di seluruh Indonesia. 20 diantaranya berada di pulau Jawa, 5 di Kalimantan, 5 di Sulawesi, dan 5 lagi berada di Papua. Hingga kini kurang lebih ABC Cinema memiliki total staff yaitu sekitar 1000 staff di seluruh Indonesia. ABC Cinema sendiri didirikan oleh 5 orang yaitu Ewaldo Samuel, Emily Indrakusuma, Andrew Duarte, Gabriela Azzahra, dan Adham Miftah. Semenjak didirikannya ABC Cinema, ABC Cinema telah mendapatkan berbagai penghargaan. Pada tahun 2017 hingga tahun 2019, ABC Cinema telah mendapatkan penghargaan sebagai bioskopp terbaik di Indonesia. Pada tahun 2020 ABC Cinema masuk salah satu nominasi bioskop terbaik di dunia.

ABC Cinema didirikan dengan tujuan agar bioskop di Indonesia lebih dapat tersebar dan masyarakat Indonesia lebih dapat menikmati film-film yang ada dengan harga yang terjangkau. Selain itu, mereka memiliki prinsip yaitu kepuasan konsumen adalah hal yang paling utama, dimana service yang mereka berikan akan dijanjikan untuk memuaskan seluruh konsumennya. ABC Cinema memiliki visi dan misi yang jelas. Visi dari ABC Cinema adalah memiliki bioskop yang futuristik dimana semua dapat dimaksimalkan dengan menggunakan teknologi. Selain itu ABC Cinema ingin konsumen merasakan bahwa ABC Cinema dapat memberikan rasa kepuasan yang ekstra bagi konsumennya. Misi dari ABC Cinema adalah agar masyarakat di seluruh Indonesia dapat menikmati seluruh film dengan cara yang sangat praktis.

ABC Cinema juga menyediakan berbagai fasilitas yang memanjakan konsumen dan juga film-film yang menarik dan eksklusif hanya terdapat pada ABC Cinema. Hingga saat ini, ABC Cinema menjadi salah satu bioskop paling ternama di Indonesia terutama pada generasi milenial karena fasilitasnya yang sangat menarik dan berbagai film yang hanya ditayangkan pada ABC Cinema.

Organizational Structure ABC Cinema



CHAPTER 2: *Company Business Process*

ABC Cinema memiliki proses bisnis yang kurang lebih sama seperti bioskop pada umumnya. *Customer* dapat melakukan pembelian tiket film dengan langsung mendatangi bioskop yang berada pada kota masing-masing.

Untuk menjelaskan lebih detail dari proses bisnis ABC Cinema, berikut dibawah ini akan dijelaskan secara detail:

Menyediakan layanan penayangan film baik film local maupun internasional.

Saat ini, Cinema ABC melayani pembelian tiket secara *walk-in*. *Customer* yang ingin membeli tiket dapat datang langsung ke bioskop di kota masing-masing secara langsung. Nantinya, saat masuk ke dalam gedung bioskop yang akan menyediakan layar lebar terkait *film* apa saja yang sedang ditayangkan dalam bioskop itu beserta studio dan jam tayang masing-masing *film* tersebut.

Setelah menemukan *film* yang cocok, *customer* dapat melakukan antre pada loket bioskop tersebut. Staff dari masing-masing loket akan melayani pembelian tiket untuk *customer*. Nantinya, *customer* bisa menyampaikan judul film dan jam tayang beserta studio yang diinginkan oleh *customer*. Staff akan menyajikan pilihan tempat duduk yang tersedia yang dapat dipilih oleh *customer*. Pada bioskop ABC, *customer* dapat langsung memesan beberapa judul film dalam satu kali pemesanan.

Setelah semua pesanan dirasa selesai, nantinya *customer* dapat memilih metode pembayaran yang ia inginkan, antara *card*, *ewallet*, atau tunai. Setelah itu, staff akan memproses pembayaran reservasi film tersebut dan akan mencetak *receipt* dan tiket untuk *customer* bawa.

Setelah proses *booking* diselesaikan, kursi yang dibeli oleh *customer* akan langsung *ter-update* menjadi tidak tersedia lagi. Sehingga, *customer* lain tidak akan bisa membeli kursi tersebut lagi.

CHAPTER 3: *Problem Statements*

Terdapat berbagai macam masalah yang dihadapi oleh *system database* yang ada di bioskop-bioskop sekarang ini. Salah satunya adalah terkadang terdapat kursi yang telah dibeli oleh pelanggan, namun masih dapat dibeli lagi oleh pelanggan lainnya. Hal tersebut mengakibatkan kesalah pahaman antar pelanggan dimana mereka sama-sama memiliki tiket dengan nomor bangku yang sama. Hal tersebut menandakan bahwa *system database* yang digunakan oleh bioskop-bioskop masih rentan akan terjadinya kesalahan. Hal tersebut dapat dipicu karena salah seorang pelanggan pesan ditempat dan yang lain pesan melalui aplikasi.

Selain itu, data *staff/cashier* yang melayani pembeli terkadang tidak terekam dengan baik karena masih adanya sistem penyimpanan manual. Sehingga perlu dibuat suatu sistem basis data antar bioskop yang ada untuk menghindari data *staff* yang tidak terekam. Dengan melalui aplikasi dan system yang kami buat, kami dapat mengatasi permasalahan yang ada pada *system database* dari bioskop-bioskop yang ada sekarang ini.

CHAPTER 4: *Database Planning*

Aktivitas perencanaan sistem database untuk Sistem Informasi Manajemen Bioskop ABC dapat terealisasi dengan baik dan efisien dengan melakukan dua langkah dalam *database planning*, yaitu mendefinisikan *mission statement* dan *mission objectives*.

4. 1. *Mission Statement*

Tujuan dari pembuatan sistem basis data pada Sistem Informasi Manajemen Bioskop adalah untuk mempermudah dalam mengorganisir dan memproses dalam melakukan pengelolaan data pada setiap cabang bioskop, seperti transaksi penjualan tiket pada setiap bioskop. Sehingga nantinya tiket yang terjual dapat *ter-update* secara *real-time*.

Hal tersebut dilakukan agar untuk mempercepat *performance* pada bioskop dan menghindari kehilangan dan keterlambatan penyimpanan data. Selain itu, data akan menjadi lebih konsisten dan validitas data dapat dipercaya.

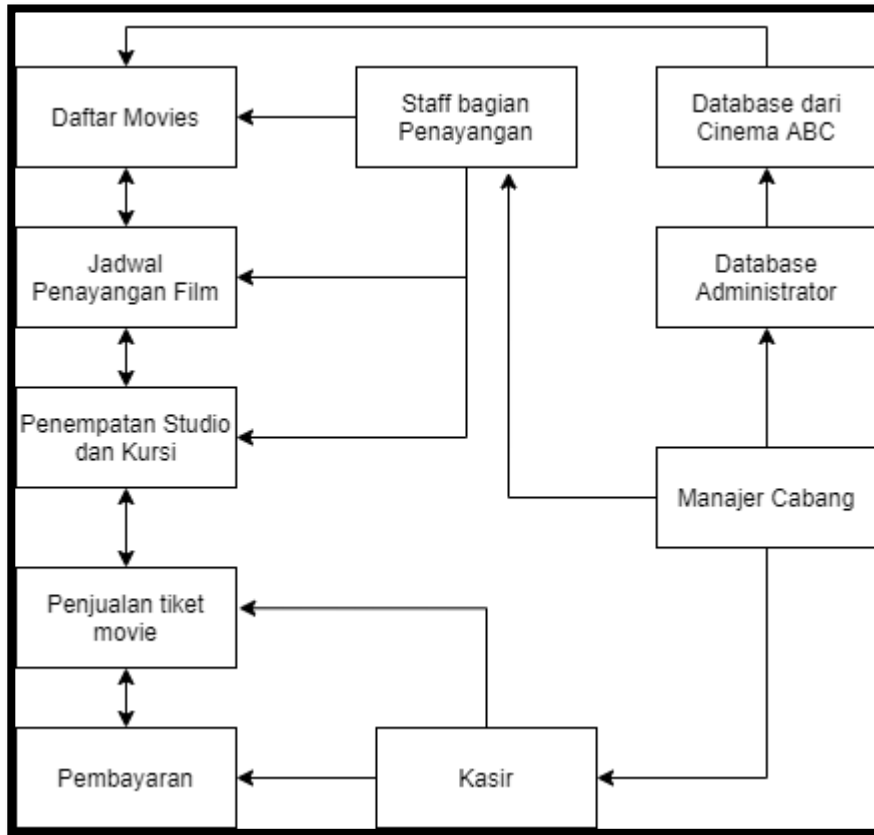
4. 2. *Mission Objective*

1. Untuk mengelola (*insert, update, delete*) data staff yang bekerja
2. Untuk mengelola (*insert, update, delete*) data *Cinemas*
3. Untuk mengelola (*insert, update, delete*) data Studio pada setiap bioskop
4. Untuk mengelola (*insert, update, delete*) data Tipe *Studio*
5. Untuk mengelola (*insert, update, delete*) data bangku bioskop
6. Untuk mengelola (*insert, update, delete*) data *Movie*
7. Untuk mengelola (*insert, update, delete*) data *Scheduled Movie*
8. Untuk mengelola (*insert, update, delete*) data *Booking Movie*
9. Untuk mengelola (*insert, update, delete*) data *Payment*
10. Untuk mengelola (*insert, update, delete*) data *Payment Type*
11. Untuk menampilkan pencarian data terkait staff yang bekerja
12. Untuk menampilkan pencarian terkait data *Cinema*
13. Untuk menampilkan pencarian terkait data Studio pada setiap bioskop
14. Untuk menampilkan pencarian terhadap data Tipe *Studio*
15. Untuk menampilkan pencarian terhadap data bangku bioskop
16. Untuk menampilkan pencarian terhadap data *Movie*
17. Untuk menampilkan pencarian terhadap data *Scheduled Movie*

18. Untuk menampilkan pencarian terhadap data *Booking Movie*
19. Untuk menampilkan pencarian terhadap data *Payment*
20. Untuk menampilkan pencarian terhadap data *Payment Type*

CHAPTER 5: *System Definition*

Perancangan basis data ini meliputi sistem bioskop dari ABC Cinema yang telah kami rancang. Berikut merupakan definisi sistem perancangan basis data pada ABC Cinema:



Gambar *System Definition*

CHAPTER 6: *Requirement Collection and Analysis*

Untuk perancangan sistem informasi manajemen bioskop ini, kami menggunakan *fact-finding technique observation*, di mana kami melihat langsung bagaimana proses bisnis sebuah bioskop berjalan. Kami memilih teknik ini karena teknik *observation* merupakan teknik yang efektif dan tidak memakan waktu banyak, khususnya untuk kasus kami yaitu bioskop yang proses bisnisnya dapat dengan mudah terlihat jelas dari luar karena para penonton bioskop berpartisipasi dalam sebagian besar proses bisnisnya.

Data	Tipe Akses	Kasir	Bagian Penayangan	Manajer Cabang	Database Administrator
Staff	Insert			X	X
	Update			X	X
	Delete			X	X
	View			X	X
	Print			X	X
Cinemas	Insert			X	X
	Update			X	X
	Delete			X	X
	View			X	X
	Print			X	X
Studios	Insert			X	X
	Update	X		X	X
	Delete			X	X
	View	X		X	X

	Print			X	X
StudioTypes	Insert			X	X
	Update			X	X
	Delete			X	X
	View	X		X	X
	Print			X	X
Seats	Insert				X
	Update	X			X
	Delete				X
	View	X		X	X
	Print				X
Movies	Insert		X		X
	Update		X		X
	Delete		X		X
	View	X	X	X	X
	Print		X		X
ScheduledMovie	Insert		X		X
	Update		X		X
	Delete		X		X
	View		X	X	X
	Print		X		X
BookingMovie	Insert	X			X
	Update	X			X

	Delete				X
	View	X		X	X
	Print	X			X
Payment	Insert	X			X
	Update				X
	Delete				X
	View	X		X	X
	Print	X			X
PaymentTypes	Insert				X
	Update				X
	Delete				X
	View	X		X	X
	Print				X

CHAPTER 7: Database Design

7.1. Conceptual Design

Berikut adalah langkah-langkah perancangan basis data konseptual, antara lain sebagai berikut:

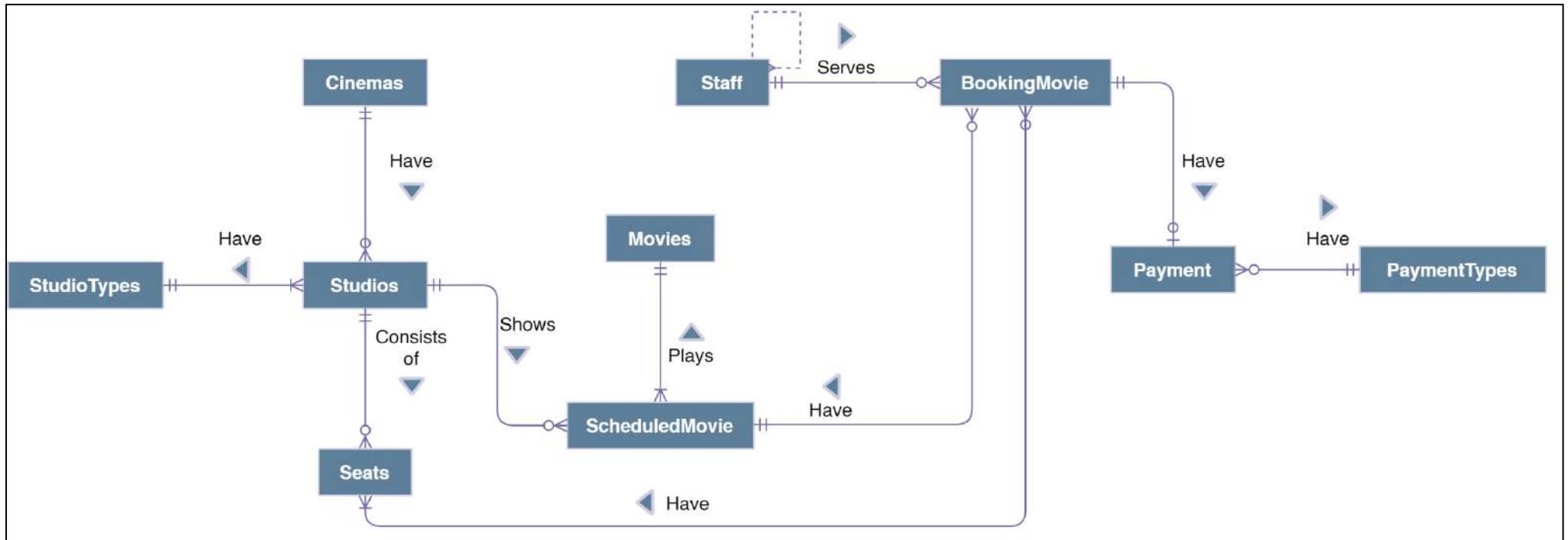
7.1.1. Identifikasi Tipe Entitas

<i>Entity Name</i>	<i>Description</i>	<i>Aliases</i>	<i>Occurence</i>
<i>Staff</i>	Mendeskripsikan mengenai staff yang bekerja	Karyawan	Satu staff bekerja hanya pada 1 gedung bioskop. Satu staff bisa melayani pemesanan movie lebih dari satu
<i>Cinemas</i>	Mendeskripikan mengenai seluruh <i>general information</i> pada bioskop	Gedung Bioskop	Setiap cinema bisa memiliki lebih dari satu studio bioskop. Setiap cinema bisa memiliki lebih dari satu karyawan
<i>Studio</i>	Mendeskripsikan mengenai detail informasi dalam studio	Studio	Satu studio hanya memiliki satu studio type. Satu studio bisa memiliki lebih dari satu movie showing. Satu studio akan dimiliki oleh hanya 1 cinema Satu studio bisa memiliki lebih dari 1 kursi

<i>StudioTypes</i>	Mendeskripsikan mengenai detail dari setiap tipe studio	Tipe Studio Bioskop	Satu Studio Type bisa dimiliki oleh satu studio.
<i>Seats</i>	Mendeskripsikan nomor bangku dalam studio.	Kursi penonton	Satu kursi penonton hanya bisa dimiliki oleh 1 studio
<i>Movies</i>	Mendeskripsikan mengenai informasi dari masing-masing movie.	Film	Satu film bisa ditayangkan lebih dari satu studio
<i>ScheduledMovie</i>	Mendeskripsikan jadwal movie yang sedang tayang.	Film dan jam tayang dari film yang sedang tayang	Satu film dapat tayang pada lebih dari satu studio dan memiliki waktu dan tanggal tayang yang berbeda-beda Dalam satu film yang tayang, merupakan 1 judul film yang ada.
<i>BookingMovie</i>	Mendeskripsikan pemesanan tiket film.	Pemesanan tiket film	Satu pemesanan tiket akan dilayani oleh satu staf. Satu pemesanan tiket akan bisa memiliki 1 pembayaran
<i>Payment</i>	Mendeskripsikan kode unik	Pembayaran	Satu pembayaran hanya berlaku untuk

	pembayaran dari setiap pemesanan tiket.		satu kali transaksi pemesanan tiket film
<i>PaymentTypes</i>	Mendeskripsikan tipe pembayaran yang digunakan user.	Metode pembayaran	Satu metode pembayaran hanya bisa digunakan untuk satu pembayaran pemesanan tiket film

First-Cut Entity Relationship Diagram



7.1.2. Identifikasi Tipe Hubungan

<i>Entity Name</i>	<i>Multiplicity</i>	<i>Relationship</i>	<i>Multiplicity</i>	<i>Entity Name</i>
<i>Staff</i>	0..*	Bekerja	1..1	<i>Cinemas</i> <i>BookingMovies</i> <i>Staff</i>
	1..1	Melayani	1..*	
	0..1	Supervise	0..*	
<i>Cinema</i>	1..1	Memiliki	0..*	<i>Studios</i>
<i>Studios</i>	1..1	Terdiri atas	0..*	<i>Seats</i> <i>StudioTypes</i> <i>ScheduledMovie</i>
	0..*	Memiliki	1..1	
	1..1	Menayangkan	0..*	
<i>Movies</i>	1..*	Menayangkan	1..1	<i>ScheduledMovie</i>
<i>BookingMovie</i>	1..1	Memiliki	0..1	<i>Payment</i> <i>ScheduledMovie</i> <i>Seats</i>
	0..*	Memiliki	1..1	
	0..*	Memiliki	1..*	
<i>Payment</i>	0..*	Memiliki	1..1	<i>PaymentTypes</i>

7.1.3. Mengidentifikasi dan Mengaitkan Atribut dengan Tipe Entitas

<i>Entity Name</i>	<i>Attributes</i>	<i>Description</i>	<i>Data types & Length</i>	<i>Nulls</i>	<i>Multi-valued</i>
<i>Cinemas</i>	1. CinemaID	1. Pengidentifikasi unik dari setiap bioskop	1. CHAR (7)	No	No
	2. CinemaName	2. Nama bioskop	2. VARCHAR (100)	No	No
	3. CinemaLocation	3. Lokasi bioskop	3. VARCHAR (100)	No	No
	4. StudioQty	4. Jumlah Studio di setiap bioskop	4. INT	No	No

<i>Studios</i>	1. StudioID 2. StudioName 3. CinemaID 4. StudioTypeID 5. Capacity	1. Pengidentifikasi unik setiap studio 2. Nama studio 3. Pengidentifikasi unik dari setiap bioskop 4. Pengidentifikasi unik dari setiap jenis studio 5. Kapasitas tiap studio	1. CHAR (10) 2. VARCHAR (30) 3. CHAR (7) 4. CHAR (6) 5. INT	No No No No No	No No No No No
<i>StudioTypes</i>	1. StudioTypeID 2. StudioTypeName	1. Pengidentifikasi unik dari setiap jenis studio 2. Nama tipe studio	1. CHAR (6) 2. VARCHAR (30)	No No	No No
<i>Seats</i>	1. SeatID 2. StudioID 3. SeatNumber	1. Pengidentifikasi unik setiap kursi 2. Pengidentifikasi unik setiap studio 3. Jumlah kursi di studio	1. CHAR (8) 2. CHAR (10) 3. CHAR(3)	No No No	No No No
<i>Staff</i>	1. StaffID 2. StaffName 3. StaffType 4. StaffPhoneNumber 5. StaffGender	1. Pengidentifikasi unik untuk setiap staff 2. Nama dari setiap staff 3. Tipe dari staff (full-time/part-time) 4. Nomor Telepon dari setiap staff 5. Jenis kelamin dari setiap staff	1. CHAR(7) 2. VARCHAR (30) 3. VARCHAR (10) 4. VARCHAR (15) 5. CHAR(1)	No No No No No	No No No Yes No

Booking Movie	<ol style="list-style-type: none"> 1. BookingID 2. StaffID 3. ScheduledMovieID 4. DateOfBooking 5. StatusOfPayment 	<ol style="list-style-type: none"> 1. Pengidentifikasi unik untuk setiap booking 2. Staff mana yang mengurus booking 3. Film mana yang dipesan 4. Tanggal booking dilakukan 5. Status dari pembayaran 	<ol style="list-style-type: none"> 1. CHAR(8) 2. CHAR(7) 3. CHAR(7) 4. DATE 5. VARCHAR(9) 	No	No
Movies	<ol style="list-style-type: none"> 1. MovieID 2. MovieName 3. Genre 4. ProductionHouse 5. ViewingCategory 6. MovieImage 	<ol style="list-style-type: none"> 1. Pengidentifikasi unik untuk setiap 2. Nama filmnya 3. Genre filmnya 4. <i>Production house</i> mana yang membuat film tersebut 5. Tentukan kategori usia mana yang cocok untuk ditonton 6. Sebagai directory gambar yang digunakan pada film tersebut 	<ol style="list-style-type: none"> 1. CHAR(7) 2. VARCHAR(100) 3. VARCHAR(20) 4. VARCHAR(20) 5. VARCHAR(5) 6. VARCHAR(100) 	No	No
Schedule dMovie	<ol style="list-style-type: none"> 1. ScheduledMovieID 2. MovieID 	<ol style="list-style-type: none"> 1. Pengidentifikasi unik dari setiap waktu tayang 2. Identifikasi film mana yang disebutkan 	<ol style="list-style-type: none"> 1. CHAR(7) 2. CHAR(7) 	No	No

	3. StudioID 4. TimeStart 5. ShowStartDate 6. ShowEndDate	3. Identifikasi di mana film akan ditayangkan 4. Waktu mulai setiap pemutaran film 5. Tanggal kapan film akan mulai tayang 6. Tanggal kapan film berakhir tayang	3. CHAR(10) 4. TIME 5. DATE 6. DATE	No No No No	No No No No
Payment	1. PaymentID 2. PaymentTypeID 3. BookingID 4. Nominal	1. Pengidentifikasi untuk setiap pembayaran 2. Pengidentifikasi unik dari tipe pembayaran 3. Pengidentifikasi unik dari setiap booking 4. Nominal dari pembayaran	1. CHAR(8) 2. CHAR(7) 3. CHAR(8) 4. INT	No No No No	No No No No
Payment Types	1. PaymentTypeID 2. PaymentTypeName	1. Pengidentifikasi unik dari setiap jenis pembayaran 2. Nama jenis pembayaran	1. CHAR (7) 2. VARCHAR (30)	No No	No No

7.1.4. Menentukan Domain Attribute

<i>Entity Name</i>	<i>Attribute</i>	<i>Attribute Domain</i>
Cinemas	1. CinemaID 2. CinemaName 3. CinemaLocation 4. StudioQty	1. 'CNM[0-9][0-9][0-9][0-9]' 2. VARCHAR (100) 3. VARCHAR (100) 4. INT

<i>Studios</i>	<ol style="list-style-type: none"> 1. StudioID 2. StudioName 3. CinemaID 4. StudioTypeID 5. Capacity 	<ol style="list-style-type: none"> 1. '[A-Z][A-Z][A-Z]STD[0-9][0-9][0-9][0-9]' 2. VARCHAR (30) 3. 'CNM[0-9][0-9][0-9][0-9]' 4. 'STP[0-9][0-9][0-9]' 5. INT
<i>StudioTypes</i>	<ol style="list-style-type: none"> 1. StudioTypeID 2. StudioTypeName 	<ol style="list-style-type: none"> 1. 'STP[0-9][0-9][0-9]' 2. 'Regular' OR 'Premiere' OR 'Platinum'
<i>Seats</i>	<ol style="list-style-type: none"> 1. SeatID 2. StudioID 3. SeatNumber 	<ol style="list-style-type: none"> 1. 'SET[A-Z][A-Z][0-9][0-9][0-9]' 2. '[A-Z][A-Z][A-Z]STD[0-9][0-9][0-9][0-9]' 3. '[A-Z][0-9][0-9]'
<i>Staff</i>	<ol style="list-style-type: none"> 1. StaffID 2. StaffName 3. StaffType 4. StaffPhoneNumber 5. StaffGender 	<ol style="list-style-type: none"> 1. 'STF[0-9][0-9][0-9][0-9]' 2. VARCHAR (30) 3. 'Full-time' OR 'Part-time' 4. VARCHAR(15) 5. 'M' OR 'F'
<i>BookingMovie</i>	<ol style="list-style-type: none"> 1. BookingID 2. StaffID 3. ScheduledMovieID 4. DateOfBooking 5. StatusOfPayment 	<ol style="list-style-type: none"> 1. 'BOK[0-9][0-9][0-9][0-9]' 2. 'STF[0-9][0-9][0-9][0-9]' 3. 'SCM[0-9][0-9][0-9][0-9]' 4. DATE 5. 'Pending' or 'Cancelled' or 'Success'
<i>Movies</i>	<ol style="list-style-type: none"> 1. MovieID 2. MovieName 3. Genre 4. ProductionHouse 	<ol style="list-style-type: none"> 1. 'MOV[A-Z0-9][A-Z0-9][A-Z0-9][A-Z0-9]' 2. VARCHAR (100) 3. VARCHAR (20) 4. VARCHAR (20)

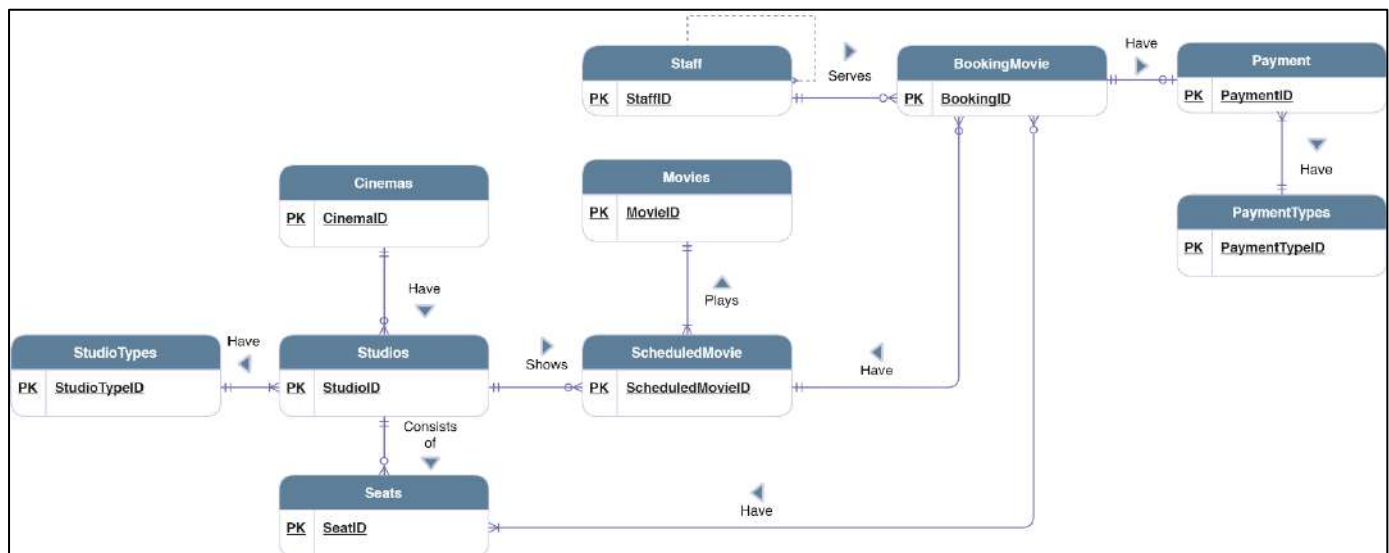
	5. ViewingCategory 6. MovieImage	5. 'G' or 'PG' or 'PG-13' or 'R' or 'NC-17' 6. VARCHAR (100)
ScheduledMovie	1. ScheduledMovieID 2. MovieID 3. StudioID 4. TimeStart 5. ShowStartDate 6. ShowEndDate	1. 'SCM[0-9][0-9][0-9][0-9]' 2. 'MOV[A-Z0-9][A-Z0-9][A-Z0-9][A-Z0-9]' 3. '[A-Z][A-Z][A-Z]STD[0-9][0-9][0-9][0-9]' 4. TIME 5. DATE 6. DATE
Payment	1. PaymentID 2. PaymentTypeID 3. BookingID 4. Nominal	1. 'PYM[A-Z0-9][A-Z0-9][A-Z0-9][A-Z0-9]' 2. 'PYT[0-9][0-9][0-9][0-9]' 3. 'BOK[A-Z0-9][A-Z0-9][A-Z0-9][A-Z0-9]' 4. INT
PaymentTypes	1. PaymentTypeID 2. PaymentTypeName	1. 'PYT[0-9][0-9][0-9][0-9]' 2. VARCHAR (30)

7.1.5. Menentukan Atribut *Candidate Key*, *Primary Key*, dan *Alternate Key*

Entity Name	Candidate Keys	Primary Keys	Alternate Keys
Cinemas	CinemaID CinemaName	CinemaID	CinemaName
Studios	StudioID	StudioID	-
StudioTypes	StudioTypeID	StudioTypeID	-

<i>Seats</i>	SeatID	SeatID	-
<i>Staff</i>	StaffID	StaffID StaffPhoneNumber	StaffPhoneNumber
<i>BookingMovie</i>	BookingID	BookingID	-
<i>Movies</i>	MovieID	MovieID	-
<i>ScheduledMovie</i>	ScheduledMovieID	ScheduledMovieID	-
<i>Payment</i>	PaymentID	PaymentID	-
<i>PaymentTypes</i>	PaymentTypeID	PaymentTypeID	-

7.1.6. Perancangan Entity Relationship Diagram dengan Primary Key



7.2. Arcs, Hierarchies, and Recursive Modeling

7.2.1. Arcs

Arcs merupakan *relationship* yang memiliki nama lain **exclusive OR relationship**.

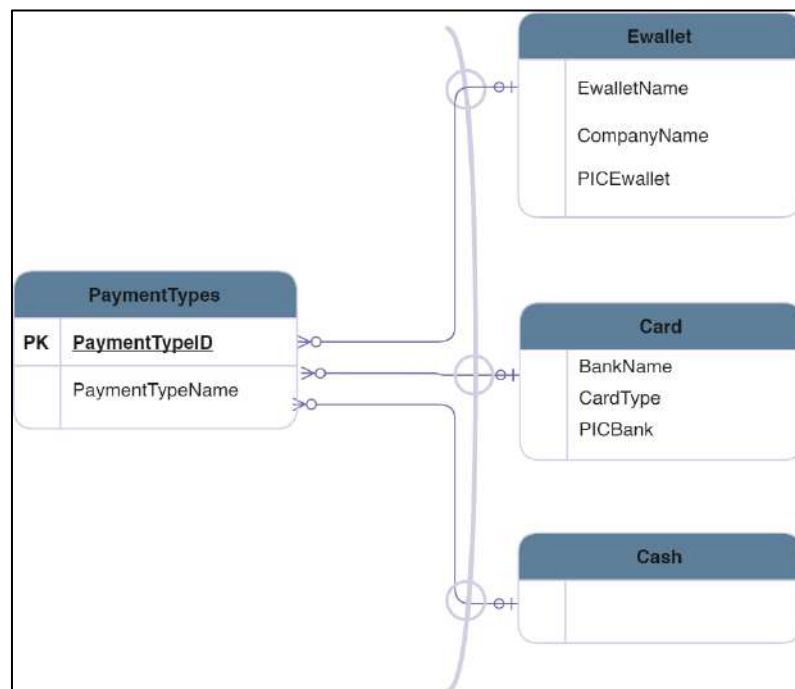
Exclusive OR relationship adalah hubungan antara satu entitas dan dua (atau lebih) entitas lain di mana hanya satu dari hubungan yang dapat ada pada suatu waktu.

Dalam perancangan basis data sistem informasi manajemen bioskop terdapat beberapa *relationship* yang termasuk dalam **exclusive OR relationship**.

a. Hubungan antara pembayaran dengan tipe pembayarannya

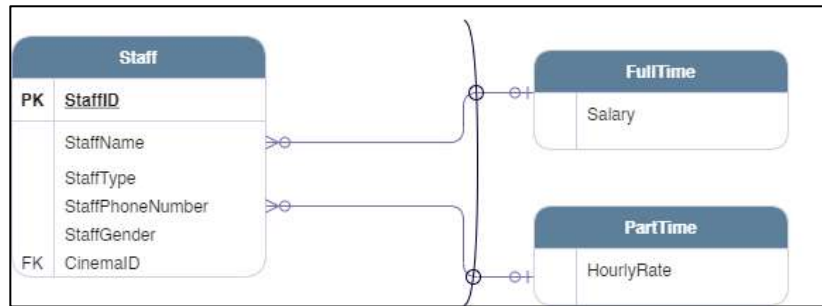
Di mana dalam satu pembayaran hanya boleh memiliki satu tipe pembayaran saja dan tidak boleh memiliki tipe pembayaran lebih dari satu.

Berikut adalah gambaran *relationship* menggunakan *arcs*:



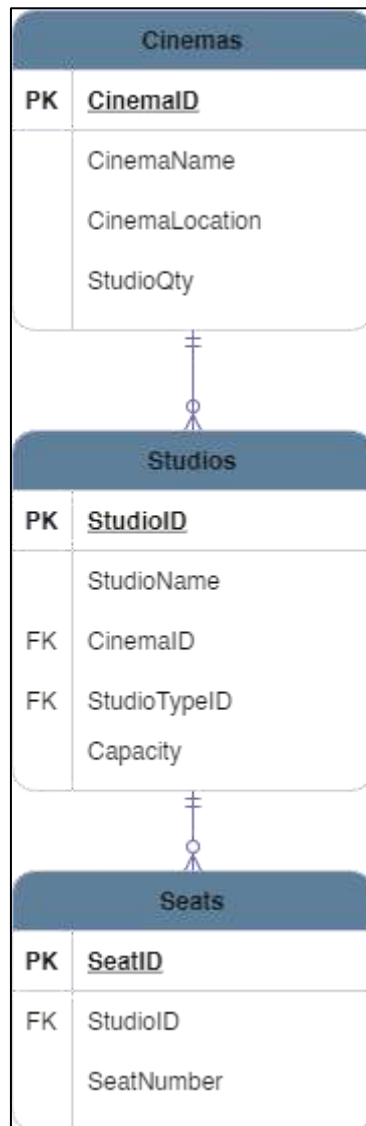
b. Hubungan antara jenis-jenis karyawan

Satu karyawan **wajib** menjadi karyawan tetap atau part-time
Namun, dalam hal ini, nantinya *part—time* dan *full-time* akan dijadikan sebagai entitas **subclass** dari staff karena memiliki **kemiripan**.



7.2.2. Hierarchies

Hierarchies modeling adalah jenis relasi yang turun temurun dari entitas satu yang lainnya.



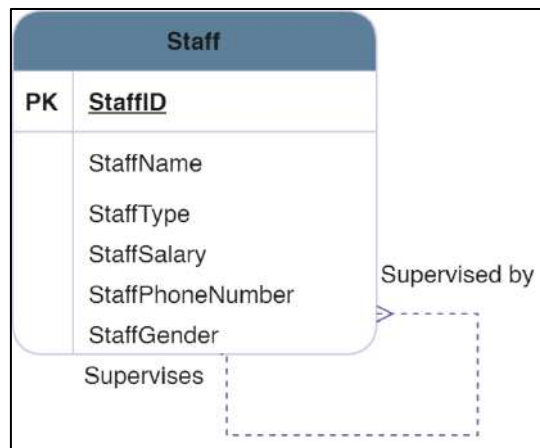
Dalam perancangan sistem basis data sistem manajemen bioskop, terdapat *hierarchies modeling*, yaitu, Cinema sebagai gedung besarnya, Studio

sebagai ruangan-ruangan yang ada di dalam satu *cinema* dan Seats yang ada di dalam setiap studio.

7.2.3. *Recursive Modeling*

Recursive modeling adalah jenis relasi yang terjadi pada satu entitas terhadap entitas itu sendiri (relasinya hanya pada 1 entitas saja)

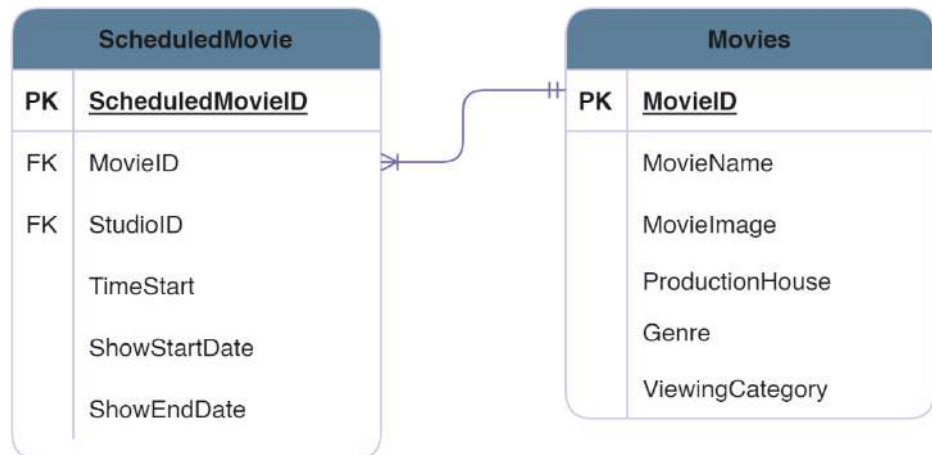
Seperti halnya:



Pada bagian tersebut, staff bisa menjadi *supervisor* yang akan melakukan supervisi karyawan lainnya.

7.3. *Changes and Historical Modeling*

7.3.1. *Historical Modeling*



7.4. *Logical Design*

Berikut adalah langkah-langkah perancangan basis data konseptual, antara lain sebagai berikut:

7.4.1. Turunkan relasi untuk model data logis: *Strong Entity*

- **Cinemas** (CinemaID, CinemaName, CinemaLocation, StudioQty)
Primary key (PK) CinemaID.
- **Staff** (StaffID, StaffName, StaffType, StaffPhoneNumber, StaffGender).
Primary key (PK) StaffID.
- **Movies** (MovieID, MovieName, Genre, ProductionHouse, ViewingCategory, MovieImage). **Primary key (PK)** MovieID

7.4.2. Turunkan relasi untuk model data logis: *Weak Entity*

- **Studios** (StudioID, StudioName, CinemaID, StudioTypeID, Capacity) **Primary key (PK)** StudioID.
- **StudioType** (StudioTypeID, StudioTypeName, Price).
Primary Key (PK) StudioTypeID
- **Seats** (SeatID, StudioID, SeatNumber).
Primary key (PK) SeatID.
- **Payment** (PaymentID, PaymentTypeID, BookingID, Nominal).
Primary key (PK) PaymentID
- **PaymentTypes** (PaymentTypeID, PaymentTypeName).
Primary key (PK) PaymentTypeID
- **ScheduledMovie** (ScheduledMovieID, MovieID, StudioID, TimeStart, ShowStartDate, ShowEndDate)
Primary key (PK) ScheduledMovieID
- **BookingMovie** (BookingID, StaffID, ScheduledMovieID, ScheduledMovieID DateOfBooking, StatusOfPayment).
Primary key (PK) BookingID

7.4.3. Turunkan relasi untuk model data logis: *one-to-many (1..*) binary relationship types*

- a. Hubungan *One-To-Many* antara StudioTypes dengan Studio Penempatan StudioTypeID ke Studio untuk hubungan memiliki:
 - **StudioTypes** (StudioTypeID, StudioTypeName, Price)
 - o **Primary Key:** StudioTypeID
 - **Studio**(StudioID, StudioName, CinemaID, StudioTypeID, Capacity)
 - o **Primary Key:** StudioID

- o **Foreign Key:** CinemaID, StudioTypeID **References** Cinemas (CinemaID), StudioTypes (StudioTypeID)
- b. Hubungan *One-to-Many* Studios dengan ScheduledMovie:
 Penempatan StudioID ke ScheduledMovie untuk hubungan memiliki:
- Studio (StudioID, StudioName, CinemaID, StudioTypeID, Capacity)
 - o **Primary Key:** StudioID
 - o **Foreign Key:** CinemaID, StudioTypeID **References** Cinemas (CinemaID), StudioTypes (StudioTypeID)
 - ScheduledMovie (ScheduledMovieID, MovieID, StudioID, TimeStart, ShowStartDate, ShowEndDate)
 - o **Primary key (PK)** ScheduledMovieID
 - o **Foreign Key:** MovieID, StudioID **References** Movie (MovieID), Studio (StudioID)
- c. Hubungan *One-to-Many* Studios dengan Seats
 Penempatan StudioID ke Seats untuk hubungan memiliki:
- Studio(StudioID, StudioName, CinemaID, StudioTypeID, Capacity)
 - o **Primary Key** StudioID
 - o **Foreign Key** CinemaID, StudioTypeID **References** Cinemas (CinemaID), StudioTypes (StudioTypeID)
 - Seats(SeatID, StudioID, SeatNumber)
 - o **Primary Key** SeatID
 - o **Foreign Key** StudioID **References** Studio(StudioID)
- d. Hubungan *One-to-Many* PaymentTypes dengan Payment
 Penempatan PaymentTypeID ke Payment memiliki:
- **PaymentTypes**(PaymentTypeID, PaymentTypeName)
 - o **Primary Key:** PaymentTypeID
 - **Payment**(PaymentID, PaymentTypeID, BookingID, Nominal)
 - o **Primary Key:** PaymentID

- o **Foreign Key:** PaymentTypeID, BookingID **References** PaymentTypes (PaymentTypeID), BookingMovie (BookingID)
- e. Hubungan *One-to-Many* Staff dengan BookingMovie
Penempatan StaffID ke BookingMovie memiliki:
- **Staff**(StaffID, StaffName, StaffType, StaffPhoneNumber, StaffGender, CinemaID)
 - o **Primary Key:** StaffID
 - o **Foreign Key:** CinemaID **References** Cinemas(CinemaID)
 - **BookingMovie**(BookingID, StaffID)
 - o **Primary Key:** BookingID
 - o **Foreign Key:** StaffID **References** Staff(StaffID)

ScheduledMovieID	References
ScheduledMovieID (ScheduledMovieID)	
- f. Hubungan *One-to-Many* ScheduledMovie dengan BookingMovie
Penempatan ScheduledMovieID ke BookingMovie memiliki:
- **ScheduledMovieID** (ScheduledMovieID, MovieID, StudioID, TimeStart, ShowStartDate, ShowEndDate)
 - o **Primary Key:** ScheduledMovieID
 - o **Foreign Key:** MovieID, StudioID **References** Movie (MovieID), Studio (StudioID)
 - **BookingMovie**(BookingID, StaffID, DateOfBooking, StatusOfPayment)
 - o **Primary Key:** BookingID
 - o **Foreign Key:** StaffID **References** Staff(StaffID)

ScheduledMovieID	References
ScheduledMovieID (ScheduledMovieID)	

7.4.4. Turunkan relasi untuk model data logis: *one-to-one (1..1) binary relationship types*

- a. Hubungan *One-to-One* BookingMovie dengan Payment
Penempatan BookingID ke Payment memiliki:
- **BookingMovie**(BookingID, StaffID, DateOfBooking, StatusOfPayment)

- o **Primary Key:** BookingID
- o **Foreign Key:** StaffID **References** Staff(StaffID)
- **Payment**(PaymentID, PaymentTypeID, BookingID, Nominal)
 - o **Primary Key:** PaymentID
 - o **Foreign Key:** PaymentTypeID, BookingID **References** PaymentTypes (PaymentTypeID), BookingMovie (BookingID)

7.4.5. Turunkan relasi untuk model data logis: *one-to-one (1..1) recursive relationship types*

Tidak terdapat *one-to-one (1:1) recursive relationship types* pada perancangan sistem basis data sistem manajemen bioskop ini.

7.4.6. Turunkan relasi untuk model data logis: *superclass/subclass relationship types*

Option 3: Mandatory, disjoint (or)

Superclass: PaymentTypes; **Subclass:** Cash, Ewallet, Card

- **Card** (PaymentTypeID, PaymentTypeName, BankName, CardType, PICBank)

Primary Key: PaymentTypeID

- **Ewallet** (PaymentTypeID, PaymentTypeName, EWalletName, CompanyName, PICEwallet)

Primary Key: PaymentTypeID

- **Cash** (PaymentTypeID, PaymentTypeName)

Primary Key: PaymentTypeID

Superclass: Staff; **Subclass:** PartTime, FullTime

- **PartTime** (StaffID, StaffName, StaffType, StaffPhoneNumber, StaffGender, CinemaID, HourlyRate)

Primary Key: StaffID

- **FullTime** (StaffID, StaffName, StaffType, StaffPhoneNumber, StaffGender, CinemaID, Salary)

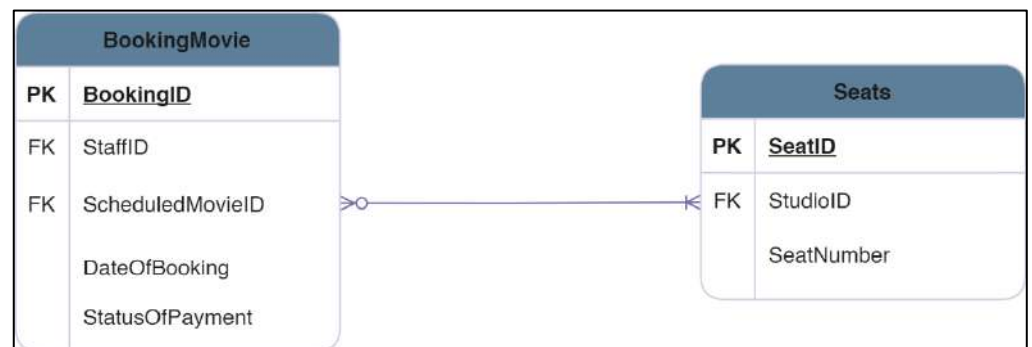
Primary Key: StaffID

7.4.7. Turunkan relasi untuk model data logis: *many-to-many (*:*) binary relationship types*

Pada bagian ini, *relation* dari *many-to-many* harus dihilangkan, sehingga yang terjadi adalah:

<p>Seats (SeatID, StudioID, SeatNumber) PrimaryKey: SeatID ForeignKey StudioID references Studio (StudioID)</p>	<p>BookingMovie (BookingID, StaffID, ScheduledMovieID, DateOfBooking, StatusOfPayment) ForeignKey StaffID, ScheduledMovieID references Staff(StaffID), ScheduledMovie (ScheduledMovieID)</p>
<p>BookingMovieDetails (BookingID, SeatID) PrimaryKey: BookingID, SeatID ForeignKey: BookingID, SeatID references BookingMovie (BookingID), Seats(SeatID)</p>	

Hubungan awal [pada conceptual]



Hubungan after



7.4.8. Turunkan relasi untuk model data logis: **complex relationship types**

BookingMovieDetails (BookingID from BookingMovie, SeatID from Seats, ScheduledMovieID from ScheduledMovie)

Primary Key: BookingID, SeatID, ScheduledMovieID

Foreign Key: BookingID references BookingMovie (BookingID)
 SeatID references Seats (SeatID)
 ScheduledMovieID references ScheduledMovie
 (ScheduledMovieID)

7.4.9. Turunkan relasi untuk model data logis: *multi-valued attribute*

Atribut *PhoneNumber* pada Staff bisa memiliki banyak nilai, sehingga perlu dibuat menjadi:

Staff (StaffID, StaffName, StaffType, StaffGender, CinemaID (FK))

Primary Key: StaffID

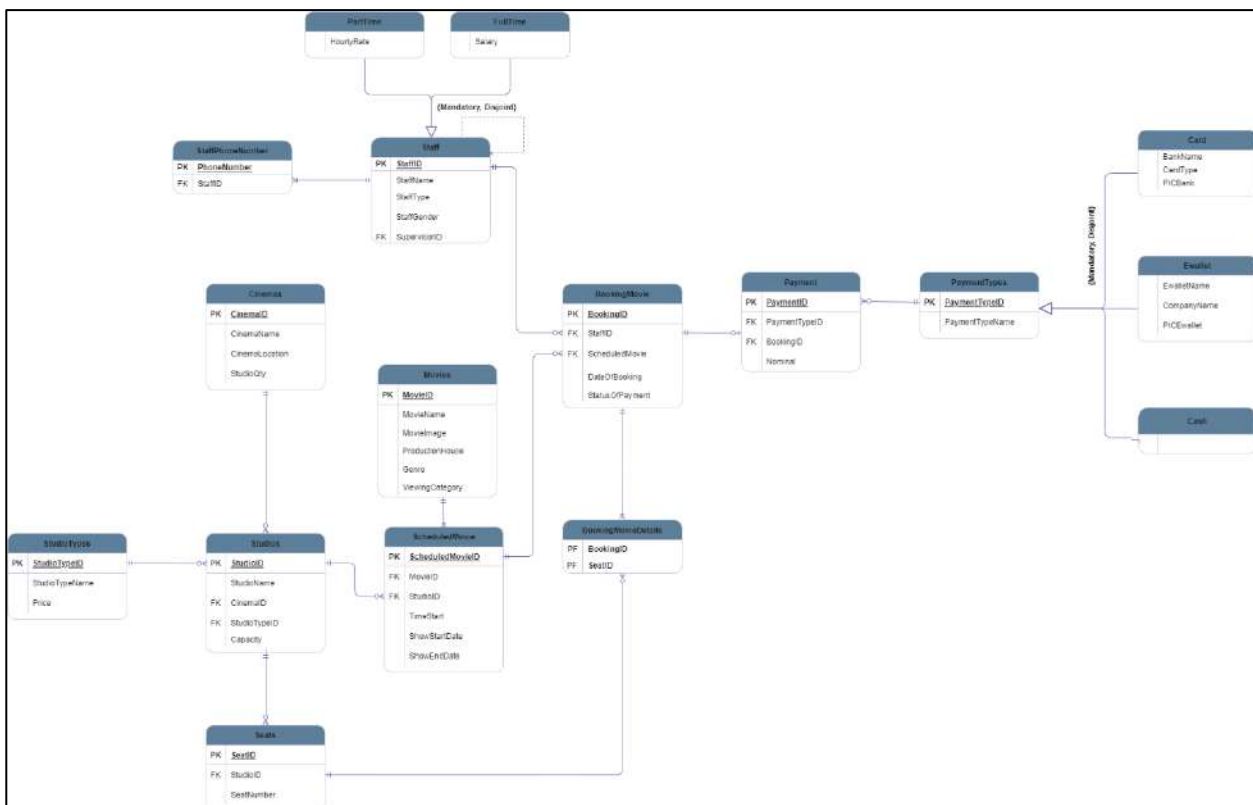
Post StaffID into StaffPhoneNumber

StaffPhoneNumber (PhoneNumber, StaffID)

Primary Key: PhoneNumber

Foreign Key: StaffID references Staff(StaffID)

7.4.10. ERD Logical Design



Link:

<https://drive.google.com/file/d/1hWZVIANM8AbL7Jv9mB5100AFQZNs3V7Z/view?usp=sharing>

7. 5. Normalization

Memvalidasi Hubungan menggunakan *normalization*

Entity	Normalisasi
Cinema	<p><u>UNF</u></p> <p>Cinema: <u>CinemaID</u>, CinemaName, CinemaLocation, StudioQty, StudioID, StudioName, Capacity</p>
	<p><u>1NF</u></p> <p>Cinema: <u>CinemaID</u>, CinemaName, CinemaLocation, StudioQty</p> <p>Studio: StudioID, StudioName, Capacity, CinemaID</p>
	<p><u>2NF</u></p> <p>Cinema: <u>CinemaID</u>, CinemaName, CinemaLocation, StudioQty</p> <p>CinemaStudio: <u>CinemaID</u>, StudioID</p> <p>Studio: StudioID, StudioName, Capacity</p>
	<p><u>3NF</u></p> <p>Cinema: <u>CinemaID</u>, CinemaName, CinemaLocation, StudioQty</p> <p>CinemaStudio: <u>CinemaID</u>, StudioID</p> <p>Studio: StudioID, StudioName, Capacity</p>
Studios	<p><u>UNF</u></p> <p>Studio: <u>StudioID</u>, StudioName, Capacity, CinemaID, CinemaName, CinemaLocation, StudioQty, StudioTypeID, StudioTypeName, Price, SeatID, SeatNumber</p>

	<p><u>1NF</u></p> <p>Studio: CinemaID, <u>StudioID</u>, StudioName, StudioTypeID, StudioTypeName, Price, Capacity</p> <p>Cinema: <u>CinemaID</u>, CinemaName, CinemaLocation</p> <p>Seat: SeatID, SeatNumber, StudioID</p> <p><u>2NF</u></p> <p>CinemaStudio: <u>CinemaID</u>, StudioID</p> <p>Studio: <u>StudioID</u>, StudioName, StudioTypeID, StudioTypeName, Price, Capacity</p> <p>Cinema: <u>CinemaID</u>, CinemaName, CinemaLocation</p> <p>Seat: SeatID, SeatNumber, StudioID</p> <p><u>3NF</u></p> <p>CinemaStudio: <u>CinemaID</u>, StudioID</p> <p>Studio: <u>StudioID</u>, StudioName, StudioTypeID, Capacity</p> <p>StudioTypes: <u>StudioTypeID</u>, StudioTypeName, Price</p> <p>Cinema: <u>CinemaID</u>, CinemaName, CinemaLocation, StudioQty</p> <p>Seat: SeatID, SeatNumber, StudioID</p>
StudioTypes	<p><u>UNF</u></p> <p>StudioTypes: <u>StudioTypeID</u>, StudioTypeName, Price, StudioID, StudioName, Capacity</p>

	<p><u>1NF</u></p> <p>StudioTypes: <u>StudioTypeID</u>, StudioTypeName, Price, StudioID, StudioName, Capacity</p> <p><u>2NF</u></p> <p>StudioTypes: <u>StudioTypeID</u>, StudioTypeName, Price, StudioID, StudioName, Capacity</p> <p><u>3NF</u></p> <p>StudioTypes: <u>StudioTypeID</u>, StudioTypeName, Price</p> <p>Studio: <u>StudioID</u>, StudioName, StudioTypeID, Capacity</p>
Staff	<p><u>UNF</u></p> <p>Staff: <u>StaffID</u>, StaffName, StaffType, StaffPhoneNumber, StaffGender, BookingID, DateOfBooking</p> <p><u>1NF</u></p> <p>Staff: <u>StaffID</u>, StaffName, StaffType, StaffPhoneNumber, StaffGender</p> <p>BookingMovie: BookingID, DateOfBooking, StaffID</p> <p><u>2NF</u></p> <p>Staff: <u>StaffID</u>, StaffName, StaffType, StaffPhoneNumber, StaffGender</p>

	<p>BookingMovie: BookingID, DateOfBooking, StaffID</p> <p><u>3NF</u></p> <p>Staff: <u>StaffID</u>, StaffName, StaffType, StaffPhoneNumber, StaffGender</p> <p>BookingMovie: BookingID, DateOfBooking, StaffID</p>
<p>Movies</p>	<p><u>UNF</u></p> <p>Movie: <u>MovieID</u>, MovieName, ProductionHouseID, ProductionHouse, Genre, ViewingCategoryID, ViewingCategory, ViewingCategoryDesc, MovieImage</p> <p><u>1NF</u></p> <p>Movie: <u>MovieID</u>, MovieName, ProductionHouseID, ProductionHouse Genre, ViewingCategoryID, ViewingCategory, ViewingCategoryDesc, MovieImage</p> <p><u>2NF</u></p> <p>Movie: <u>MovieID</u>, MovieName, ProductionHouseID, Genre, ViewingCategoryID, MovieImage</p> <p>ProductionHouse: <u>ProductionHouseID</u>, ProductionHouse</p> <p>ViewingCategory: <u>ViewingCategoryID</u>, ViewingCategory, ViewingCategoryDesc</p> <p><u>3NF</u></p> <p>Movies: <u>MovieID</u>, MovieName, Genre, ProductionHouseID, ViewingCategoryID, MovieImage</p>

	<p>ProductionHouse: <u>ProductionHouseID</u>, ProductionHouse</p> <p>ViewingCategory: <u>ViewingCategoryID</u>, ViewingCategory, ViewingCategoryDesc</p>
<p>ScheduledMovie</p>	<p><u>UNF</u></p> <p><u>ScheduledMovieID:</u> <u>ScheduledMovieID</u>, MovieID, StudioID, TimeStart, ShowStartDate, ShowEndDate, MovieName, ProductionHouse, Genre, ViewingCategory, StudioName, CinemaID, StudioTypeID, Capacity</p> <p><u>1NF</u></p> <p><u>ScheduledMovieID:</u> <u>ScheduledMovieID</u>, MovieID, StudioID, TimeStart, ShowStartDate, ShowEndDate, MovieName, ProductionHouse, Genre, ViewingCategory, StudioName, CinemaID, StudioTypeID, Capacity</p> <p><u>2NF</u></p> <p><u>ScheduledMovie:</u> <u>ScheduledMovieID</u>, MovieID, StudioID, TimeStart, ShowStartDate, ShowEndDate,</p> <p>Movie: <u>MovieID</u>, MovieName, ProductionHouseID, Genre, ViewingCategoryID</p> <p>Studio: <u>StudioID</u>, StudioName, CinemaID, StudioTypeID</p> <p><u>3NF</u></p> <p><u>ScheduledMovie:</u> <u>ScheduledMovieID</u>, MovieID, StudioID, TimeStart, ShowStartDate, ShowEndDate</p>

	<p>Movie: <u>MovieID</u>, MovieName, ProductionHouse, Genre, ViewingCategoryID</p> <p>Studio: <u>StudioID</u>, StudioName, CinemaID, StudioTypeID</p> <p>Cinema: <u>CinemaID</u>, CinemaName, CinemaLocation, StudioQty</p> <p>StudioTypes: <u>StudioTypeID</u>, StudioTypeName, Price</p>
<p>Seats</p>	<p><u>UNF</u></p> <p>Seats: <u>SeatID</u>, StudioID, SeatNumber, StudioName, CinemaID, CinemaName, StudioTypeID, Capacity</p> <p><u>1NF</u></p> <p>Seats: <u>SeatID</u>, StudioID, SeatNumber</p> <p>Studio: StudioID, StudioTypeID, StudioTypeName, CinemaID, CinemaName, Capacity</p> <p><u>2NF</u></p> <p>Seats: <u>SeatID</u>, StudioID, SeatNumber</p> <p>Studio: <u>StudioID</u>, StudioTypeID, StudioTypeName, CinemaID, CinemaName, Capacity</p> <p><u>3NF</u></p> <p>Seats: <u>SeatID</u>, SeatNumber, StudioID</p> <p>Studio: <u>StudioID</u>, StudioTypeID, CinemaID, Capacity</p> <p>Cinema: CinemaID, CinemaName</p> <p>StudioType: StudioTypeID, StudioTypeName</p>

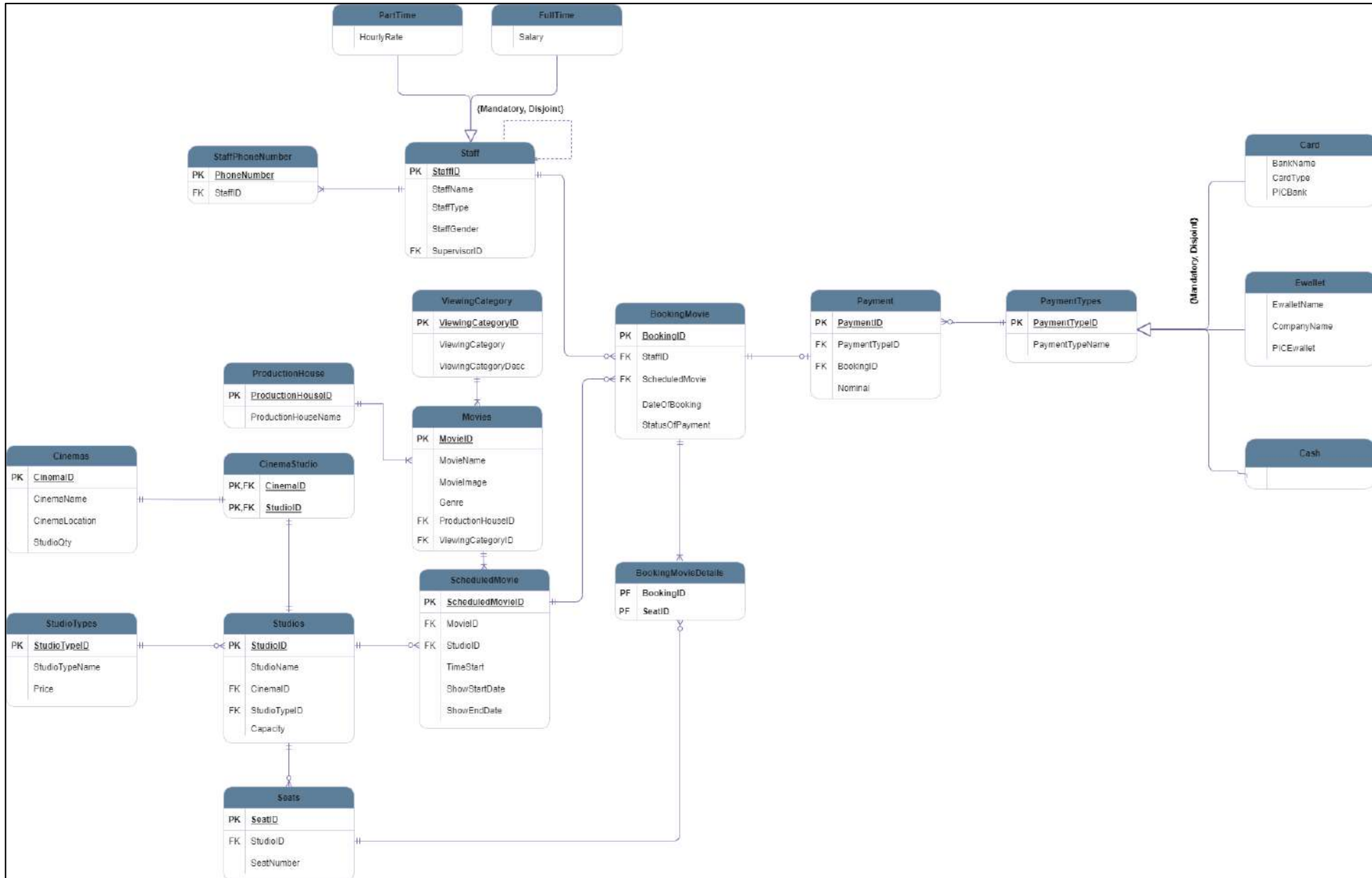
BookingMovie	<p><u>UNF</u></p> <p>BookingMovie: <u>BookingID</u>, StaffID, ScheduledMovieID, DateOfBooking, StatusOfPayment, StaffName, StaffType, StaffGender, MovieID, StudioID, TimeStart, ShowStartDate, ShowEndDate, SeatID, SeatNumber</p> <p><u>1NF</u></p> <p>BookingMovie: <u>BookingID</u> , ScheduledMovieID, DateOfBooking, StatusOfPayment, MovieID, StudioID, TimeStart, ShowStartDate, ShowEndDate, StaffID, StaffName, StaffType, StaffGender</p> <p>BookingMovieDetails: BookingID, SeatID, SeatNumber</p> <p><u>2NF</u></p> <p>BookingMovie: <u>BookingID</u>, , ScheduledMovieID, DateOfBooking, StatusOfPayment, MovieID, StudioID, TimeStart, ShowStartDate, ShowEndDate, StaffID, StaffName, StaffType, StaffGender</p> <p>BookingMovieDetails: BookingID, SeatID</p> <p>Seat: SeatID, SeatNumber</p> <p><u>3NF</u></p> <p>BookingMovie: <u>BookingID</u>, ScheduledMovieID, DateOfBooking, StatusOfPayment, StaffID</p> <p>BookingMovieDetails: BookingID, SeatID</p> <p>Seat: SeatID, SeatNumber</p> <p>Staff: <u>StaffID</u>, StaffName, StaffType, StaffGender</p>

	<p>ScheduledMovie: ScheduledMovieID, MovieID, StudioID, TimeStart, ShowStartDate, ShowEndDate</p>
Payment	<p><u>UNF</u></p> <p>Payment: <u>PaymentID</u>, PaymentTypeID, BookingID, Nominal, StaffID, StaffName, StaffType, StaffGender, ScheduledMovieID, BookingID, DateOfBooking, StatusOfPayment, PaymentTypeName</p> <p><u>1NF</u></p> <p>Payment: <u>PaymentID</u>, PaymentTypeID, BookingID, Nominal, StaffID, StaffName, StaffType, StaffGender, ScheduledMovieID, BookingID, DateOfBooking, StatusOfPayment, PaymentTypeName</p> <p><u>2NF</u></p> <p>Payment: <u>PaymentID</u>, PaymentTypeID, BookingID, Nominal</p> <p>BookingMovie: BookingID, StaffID, ScheduledMovie, DateOfBooking, StatusOfPayment</p> <p><u>3NF</u></p> <p>Payment: <u>PaymentID</u>, PaymentTypeID, Nominal</p> <p>BookingMovie: BookingID, StaffID, ScheduledMovie, DateOfBooking, StatusOfPayment</p> <p>PaymentTypes: PaymentTypeID, PaymentTypeName</p>
PaymentTypes	<p><u>UNF</u></p> <p>PaymentType: <u>PaymentTypeID</u>, PaymentTypeName, PaymentId, Nominal</p>

	<p><u>1NF</u></p> <p>PaymentType: <u>PaymentTypeID</u>, PaymentTypeName</p> <p>Payment: <u>PaymentID</u>, PaymentTypeID, Nominal</p>
	<p><u>2NF</u></p> <p>PaymentType: <u>PaymentTypeID</u>, PaymentTypeName</p> <p>Payment: <u>PaymentID</u>, PaymentTypeID, Nominal</p>
	<p><u>3NF</u></p> <p>PaymentType: <u>PaymentTypeID</u>, PaymentTypeName</p> <p>Payment: <u>PaymentID</u>, PaymentTypeID, Nominal</p>

Entity Relationship Diagram setelah Normalization (link:

https://drive.google.com/file/d/1aqXhdb9TQbCv_yYXCaMZxKcw0GQkMDr0/view?usp=sharing)



7. 6. *Physical Design Database*

7.6.1. **Step 3: Translate Logical Data Model for Target DBMS**

a. Design Base Relations

Table 1: Cinemas (DBDL for Cinemas Relation)

1. Domain CinemaID CHAR(7); must be in 'CNM[0-9][0-9][0-9][0-9]'
2. Domain CinemaName VARIABLE LENGTH CHARACTER STRING, length 100
3. Domain CinemaLocation VARIABLE LENGTH CHARACTER STRING, length 100
4. Domain StudioQty integer, in the range of 0-15

Cinemas(

CinemaID	CinemaIdentifier	NOT NULL,
CinemaName	CinemaName	NOT NULL,
CinemaLocation	LocationOfCinema	NOT NULL,
StudioQty	NumberOfStudio	NOT NULL,

PRIMARY KEY (CinemaID)

)

Table 2: Studios (DBDL for Studios Relation)

1. Domain StudioID CHAR(10); must be in '[A-Z][A-Z][A-Z]STD[0-9][0-9][0-9][0-9]'
2. Domain StudioName VARIABLE LENGTH CHARACTER STRING, length 30
3. Domain CinemaID CHAR(7); must be in 'CNM[0-9][0-9][0-9][0-9]'
4. Domain StudioTypeID CHAR(6); must be in 'STP[0-9][0-9][0-9]'
5. Domain Capacity integer, in the range of 20-250

Studios(

StudioID	StudioIdentifier	NOT NULL,
----------	------------------	-----------

StudioName	StudioName	NOT NULL,
CinemaID	CinemaIdentifier	NOT NULL,
StudioTypeID	TypeOfStudio	NOT NULL,
Capacity	StudioCapacity	NOT NULL,

PRIMARY KEY (StudioID)
FOREIGN KEY (CinemaID) REFERENCES Cinema(CinemaID) ON
UPDATE CASCADE ON DELETE CASCADE
FOREIGN KEY (StudioTypeID) REFERENCES
StudioTypes(StudioTypeID) ON UPDATE CASCADE ON DELETE
CASCADE
)

Table 3: CinemaStudio (DBDL for CinemaStudio Relation)

1. Domain StudioID CHAR(10); must be in '[A-Z][A-Z][A-Z]STD[0-9][0-9][0-9][0-9]'
2. Domain CinemaID CHAR(7); must be in 'CNM[0-9][0-9][0-9][0-9]'

CinemaStudio(

StudioID	StudioIdentifier	NOT NULL,
CinemaID	CinemaIdentifier	NOT NULL,

PRIMARY KEY (StudioID, CinemaID)
FOREIGN KEY (CinemaID) REFERENCES Cinema(CinemaID) ON
UPDATE CASCADE ON DELETE CASCADE
FOREIGN KEY (StudioTypeID) REFERENCES
StudioTypes(StudioTypeID) ON UPDATE CASCADE ON DELETE
CASCADE
)

Table 4: StudioTypes (DBDL for StudioTypes Relation)

1. Domain StudioTypeID CHAR(6); must be in 'STP[0-9][0-9][0-9]'

2. Domain StudioTypeName VARIABLE LENGTH CHARACTER STRING, length 30, must be in 'Regular' OR 'Premiere' OR 'Platinum'

StudioTypes(

 StudioTypeID StudioTypeIdentifier NOT NULL,

 StudioTypeName StudioTypeName NOT NULL,

 PRIMARY KEY (StudioTypeID)

)

Table 5: Seats (DBDL for Seats Relation)

1. Domain SeatID CHAR(8); must be in 'SET[A-Z][A-Z][0-9][0-9][0-9]'
2. Domain StudioID CHAR(10); must be in '[A-Z][A-Z][A-Z]STD[0-9][0-9][0-9][0-9]'
3. Domain SeatNumber CHAR(3)

Seats(

 SeatID SeatIdentifier NOT NULL,

 StudioID StudioIdentifier NOT NULL,

 SeatNumber SeatNumber NOT NULL

 PRIMARY KEY (SeatID)

 FOREIGN KEY (StudioID) REFERENCES Studio(StudioID) ON UPDATE

 CASCASE ON DELETE CASCADE

)

Table 6: Staff (DBDL for Staff Relation)

1. Domain StaffID CHAR(7); must be in 'STF[0-9][0-9][0-9][0-9]'
2. Domain StaffName VARIABLE LENGTH CHARACTER STRING; length 30
3. Domain StaffType VARIABLE LENGTH CHARACTER STRING, length 10, must be in 'Full-time' OR 'Part-time'
4. Domain StaffPhoneNumber VARIABLE LENGTH CHARACTER STRING, length 15, must be start with '+62' OR '08'

5. Domain StaffGender SINGLE CHARACTER, must be in 'F' or 'M'
6. Domain SupervisorID CHAR(7); must be in 'STF[0-9][0-9][0-9][0-9]'

Staff(

StaffID	StaffIdentifier	NOT NULL,
StaffName	StaffName	NOT NULL,
StaffType	TypeOfStaff	NOT NULL,
StaffPhoneNumber	StaffPhoneNumber	NOT NULL,
StaffGender	StaffGender	NOT NULL,
SupervisorID	SupervisorIdentifier	NOT NULL

PRIMARY KEY (StaffID)

)

Table 7: BookingMovie (DBDL for BookingMovie Relation)

1. Domain BookingID CHAR(8) must be in BOK[0-9][0-9][0-9][0-9]
2. Domain StaffID CHAR(7) must be in 'STF[0-9][0-9][0-9][0-9]'
3. Domain ScheduledMovieID CHAR(7) must be in 'SCM[0-9][0-9][0-9][0-9]'
4. Domain DateOfBooking DATE
5. Domain StatusOfPayment VARIABLE LENGTH CHARACTER STRING, length 9, must be in 'Pending' or 'Cancelled' or 'Success'

BookingMovie(

BookingID	BookingIdentifier	NOT NULL,
StaffID	StaffIdentifier	NOT NULL,
ScheduledMovieID	MovieShowNowID	NOT NULL,
DateOfBooking	BookingDate	NOT NULL,
StatusOfPayment	BookingStatusPayment	NOT NULL

PRIMARY KEY (BookingID)

FOREIGN KEY (StaffID) REFERENCES Staff (StaffID) ON UPDATE
 CASCADE ON DELETE CASCADE

)

Table 8: BookingMovieDetails (DBDL for BookingMovieDetails Relation)

1. Domain BookingID CHAR(8) must be in 'BOK[0-9][0-9][0-9][0-9]'
2. Domain SeatID CHAR(8); must be in 'SET[A-Z][A-Z][0-9][0-9][0-9]'

BookingMovieDetails(
 BookingID BookingIdentifier NOT NULL,
 SeatID SeatIdentifier NOT NULL,
 PRIMARY KEY (BookingID, SeatID)

FOREIGN KEY (BookingID) REFERENCES BookingMovie (BookingID)
 ON UPDATE CASCADE ON DELETE CASCADE

FOREIGN KEY (SeatID) REFERENCES Seats (SeatID) ON UPDATE
 CASCADE ON DELETE CASCADE

)

Table 9: Movies (DBDL for Movies Relation)

1. Domain MovieID CHAR(8); must be in BOK[0-9][0-9][0-9][0-9]'
2. Domain MovieName VARIABLE LENGTH CHARACTER STRING;
 length 100
3. Domain Genre CHAR(7) must be in 'SCM[0-9][0-9][0-9][0-9]'
4. Domain ProductionHouseID CHAR(7) must be in 'PHO[0-9][0-9][0-9]'
5. Domain ViewingCategoryID CHAR(4) must be in 'VC[0-9][0-9]'
6. Domain MovieImage VARIABLE LENGTH CHARACTER STRING;
 length 100

Movies(
 MovieID MovieIdentifier NOT NULL,
 MovieName NameOfMovie NOT NULL,

```

Genre                GenreOfMovie          NOT NULL,
ProductionHouseID    ProdHouseIdentifier   NOT NULL,
ViewingCategoryID    ViewingCatIdentifier  NOT NULL
MovieImage           DirectoryMovieImage   NOT NULL

PRIMARY KEY (MovieID)

FOREIGN KEY (ProductionHouseID) REFERENCES
ProductionHouse (ProductionHouseID) ON UPDATE
CASCADE ON DELETE CASCADE
FOREIGN KEY (ViewingCategoryID) REFERENCES
ViewingCategory(ViewingCategoryID) ON UPDATE
CASCADE ON DELETE CASCADE
)

```

Table 10: ProductionHouse (DBDL for ProductionHouse Relation)

1. Domain ProductionHouseID CHAR(7); must be in 'PHO[0-9][0-9][0-9]'
2. Domain ProductionHouseName VARIABLE LENGTH CHARACTER STRING, length 100

```

ProductionHouse(
    ProductionHouseID ProdHouseIdentifier NOT NULL,
    MovieName         NameOfProdHouse     NOT NULL,
    PRIMARY KEY (ProductionHouse)
)

```

Table 11: ViewingCategory (DBDL for ViewingCategory Relation)

1. Domain ViewingCategoryID CHAR(4); must be in 'VC[0-9][0-9][0-9]'

2. Domain ViewingCategory VARIABLE LENGTH CHARACTER STRING, length 100
3. Domain ViewingCategoryDesc VARIABLE LENGTH CHARACTER STRING, length 255

ViewingCategory (

ViewingCategoryID	ViewingCatIdentifier	NOT NULL,
ViewingCategory	ViewingCategoryName	NOT NULL,
ViewingCategoryDesc	DescriptionOfViewingCategory	NOT NULL,

PRIMARY KEY (ViewingCategoryID)

)

Table 12: ScheduledMovie (DBDL for ScheduledMovie Relation)

1. Domain ScheduledMovieID CHAR(7); must be in 'SCM[0-9][0-9][0-9][0-9]'
2. Domain MovieID CHAR(7); must be in 'MOV[A-Z0-9][A-Z0-9][A-Z0-9][A-Z0-9]'
3. Domain StudioID CHAR(7); must be in '[A-Z][A-Z][A-Z]STD[0-9][0-9][0-9][0-9]'
4. TimeStart TIME
5. ShowStartDate DATE
6. ShowEndDate DATE

ScheduledMovie(

ScheduledMovieID	SchedMovieIdentifier	NOT NULL,
MovieID	MovieIdentifier	NOT NULL,
StudioID	StudioIdentifier	NOT NULL,
TimeStart	MovieTimeStart	NOT NULL,
ShowStartDate	MovieStartDate	NOT NULL

```

        ShowEndDate      MovieEndDate      NOT NULL
PRIMARY KEY (ScheduledMovieID)
FOREIGN KEY MovieID REFERENCES Movie(MovieID) ON
UPDATE CASCADE ON DELETE CASCADE
FOREIGN KEY StudioID REFERENCES Studio(StudioID) ON
UPDATE CASCADE ON DELETE CASCADE
)

```

Table 13: Payment (DBDL for Payment Relation)

1. Domain PaymentID CHAR(8); must be in 'PYM[A-Z0-9][A-Z0-9][A-Z0-9][A-Z0-9][A-Z0-9]
2. Domain PaymentTypeID CHAR(7) must be in 'PYT[0-9][0-9][0-9][0-9]
3. Domain BookingID CHAR(8) must be in 'BOK[0-9][0-9][0-9][0-9]
4. Nominal INT

Payment (

```

        PaymentID      PaymentIdentifier      NOT NULL,
        PaymentTypeID  TypePayment            NOT NULL,
        BookingID      BookingIdentifier      NOT NULL,
        Nominal         NominalOfPayment      NOT NULL
PRIMARY KEY (PaymentID)
FOREIGN KEY (PaymentTypeID) REFERENCES
PaymentTypes(PaymentTypeID) ON UPDATE CASCADE ON
DELETE NO ACTION
FOREIGN KEY (BookingID) REFERENCEES BookingMovies (BookingID)
ON UPDATE CASCADE ON DELETE NO ACTION
)

```

Table 14: PaymentType (DBDL for PaymentType Relation)

1. Domain PaymentTypeID CHAR(7); must be in 'PYT[0-9][0-9][0-9][0-9]
2. Domain PaymentTypeName VARIABLE LENGTH CHARACTER STRING, length 30

ProductionHouse(

PaymentTypeID PaymentTypeIdentifier NOT NULL,

PaymentTypeName NameOfPayment NOT NULL

PRIMARY KEY (PaymentTypeID)

)

b. Design Representation of Derived Data

Staff

StaffID	StaffName	Staff Type	Staff Gender	Supervisor ID	Number ofBooking
STF001	Andrew Duarte	Full-Time	M	STF005	2
STF002	Sulaiman	Full-Time	M	STF005	0
STF003	Britney Spears	Part-Time	F	STF004	1
STF005	Hariyatno	Full-Time	M	STF003	0

BookingMovie

BookingID	StaffID	Scheduled Movie	DateOf Booking	Status OfPayment
BOKA0121	STF001	SCM001	2021-12-12	Success
BOKA0122	STF002	SCM002	2021-12-12	Cancelled
BOKA0123	STF001	SCM002	2021-12-12	Success

BOKA0124	STF003	SCM003	2021-12-12	Success
----------	--------	--------	------------	---------

Penjelasan:

Derived Attribute → **NumberOfBooking**

Attribut StatusOfPayment pada **BookingMovie** yang sudah berubah status nya menjadi “Success” akan mempengaruhi attribut NumberOfBooking pada **Staff** dimana nantinya data akan terupdate sesuai dengan status “Success” yang ada pada StatusOfPayment.

BookingMovie

BookingID	StaffID	ScheduledMovie	DateOfBooking	Status OfPayment	Total Seats
BOKA0121	STF001	SCM001	2021-12-12	Success	2
BOKA0122	STF002	SCM002	2021-12-12	Cancelled	1
BOKA0123	STF001	SCM002	2021-12-12	Success	2
BOKA0124	STF003	SCM003	2021-12-12	Success	3

BookingMovieDetail

BookingID	SeatID
BOKA0121	SETAB011
BOKA0121	SETAB012
BOKA0122	SETBB013
BOKA0123	SETBB014
BOKA0123	SETBB015
BOKA0124	SETCB016

BOKA0124	SETCB017
BOKA0124	SETCB018

Penjelasan:

Derived Attribute → **TotalSeats**

c. Design General Constraints

1. **Atribut CinemaID pada entitas Cinema harus diisi dengan format 'CNM[0-9][0-9][0-9][0-9]'**

```
CONSTRAINT CheckCinemaID CHECK(CinemaID LIKE 'CNM[0-9][0-9][0-9][0-9]')
```

2. **Atribut StudioID pada entitas Studio harus memiliki format '[A-Z][A-Z][A-Z]STD[0-9][0-9][0-9][0-9]'**

```
CONSTRAINT CheckStudioID CHECK(StudioID LIKE '[A-Z][A-Z][A-Z]STD[0-9][0-9][0-9][0-9]')
```

3. **Atribut StudioTypeID pada entitas StudioType harus memiliki format 'STP[0-9][0-9][0-9]'**

```
CONSTRAINT CheckStudioTypeID CHECK (StudioTypeID LIKE 'STP[0-9][0-9][0-9]')
```

4. **Atribut SeatID pada entitas Seats harus memiliki format 'SET[A-Z][A-Z][0-9][0-9][0-9]'**

```
CONSTRAINT CheckSeatID CHECK(SeatID LIKE 'SET[A-Z][A-Z][0-9][0-9][0-9]')
```

5. **Atribut StaffID pada entitas Staff harus memiliki format 'STF[0-9][0-9][0-9][0-9]'**

```
CONSTRAINT CheckStaffID CHECK(StaffID LIKE 'STF[0-9][0-9][0-9][0-9]')
```

6. **Atribut BookingID pada entitas BookingMovie harus memiliki format 'BOK[0-9][0-9][0-9][0-9]'**

```
CONSTRAINT CheckBookingMovieID CHECK BookingMovieID LIKE 'BOK[0-9][0-9][0-9][0-9]')
```

7. **Atribut MovieID pada entitas Movie harus memiliki format 'MOV[A-Z0-9][A-Z0-9][A-Z0-9][A-Z0-9]'**
 CONSTRAINT CheckMovieID CHECK MovieID LIKE 'MOV[A-Z0-9][A-Z0-9][A-Z0-9][A-Z0-9]'
8. **Atribut ScheduledMovieID pada entitas ScheduledMovie harus memiliki format 'SCM[0-9][0-9][0-9][0-9]'**
 CONSTRAINT CheckScheduledMovieID CHECK (ScheduledMovieID LIKE 'SCM[0-9][0-9][0-9][0-9]')
9. **Atribut PaymentID pada entitas Payment harus memiliki format 'PYM[A-Z0-9][A-Z0-9][A-Z0-9][A-Z0-9][A-Z0-9]'**
 CONSTRAINT CheckPaymentID CHECK (PaymentID LIKE 'PYM[A-Z0-9][A-Z0-9][A-Z0-9][A-Z0-9][A-Z0-9]')
10. **Atribut PaymentTypeID pada entitas PaymentType harus memiliki format 'PYT[0-9][0-9][0-9][0-9]'**
 CONSTRAINT CheckPaymentTypeID CHECK (PaymentTypeID LIKE 'PYT[0-9][0-9][0-9][0-9]')
11. **Atribut ViewingCategory pada entitas Movies hanya bisa di isi dengan 'G' atau 'PG' atau 'PG-13' atau 'R' atau'NC-17'**
 CONSTRAINT CheckViewingCategory CHECK (ViewingCategory IN ('G', 'PG', 'PG-13', 'R', 'NC-17'))
12. **Atribut StaffType pada entitas Staff hanya bisa diisi dengan 'Full-time' atau 'Part-time'**
 CONSTRAINT CheckStaffType CHECK (StaffType IN ('Full-time', 'Part-time'))
13. **Atribut StaffPhoneNumber pada Staff harus diawali dengan '+62' atau '08'**
 CONSTRAINT CheckStaffPhone CHECK (StaffPhoneNumber LIKE '+62%' OR StaffPhoneNumber LIKE '08%')
14. **Atribut StaffGender pada entitas Staff hanya bisa diisi dengan 'M' atau 'F'**

CONSTRAINT CheckStaffGender CHECK StaffGender IN ('M',
'F')

15. **Satu staff dalam satu hari hanya boleh melayani maksimal 100 booking**

CONSTRAINT StaffNotHandlingTooMuchBooking
CHECK (NOT EXISTS (SELECT StaffID,
FROM BookingMovie bm, Staff s
WHERE bm.StaffID = s.StaffID
GROUP BY StaffID
Having COUNT(BookingID)>100))

16. **Atribut StatusOfPayment pada entitas Payment hanya bisa diisi dengan 'Pending', 'Cancelled' atau 'Success'**

CONSTRAINT CheckStatusOfPayment CHECK(StatusOfPayment IN
('Pending', 'Cancelled', 'Success'))

17. **Atribut StudioTypeName pada entitas StudioType hanya bisa diisi dengan 'Regular' atau 'Premiere' atau 'Platinum'**

CONSTRAINT CheckStudioTypeName CHECK StudioTypeName IN
('Regular', 'Premiere', 'Platinum')

18. **Atribut Capacity pada entitas Studios hanya bisa diisi dengan jarak 20 sampai 250**

CONSTRAINT CheckStudioCapacity CHECK Capacity >= 20 AND
Capacity <= 250

19. **Satu studio maksimal menayangkan 2 scheduled movie**

CONSTRAINT ShowingMovieInStudio
CHECK (NOT EXISTS (SELECT StudioID
FROM ScheduledMovie sm, Studio s
WHERE sm.StudioID = s.StudioID
GROUP BY StudioID
HAVING COUNT(*) > 2))

7.6.2. Step 4: Design File Organizations and indexes

a. Analyze Transaction

- (a) Memasukkan dan menampilkan data *movie* beserta detail *movie*
- (b) Memasukkan dan menampilkan *moviescheduled* sesuai dengan *movie* yang tersedia
- (c) Menampilkan data *staff*
- (d) Menampilkan dan memasukkan data *bookingmovie*
- (e) Menampilkan dan memasukkan data *detail bookingmovie*
- (f) Menampilkan dan memasukkan data *payment*
- (g) Menampilkan kursi yang tersedia pada suatu *scheduledmovie*
- (h) Menampilkan data studio
- (i) Menampilkan data *cinema*
- (j) Menampilkan data *studiotype*
- (k) Menampilkan data *paymenttype*
- (l) Mengidentifikasi jumlah staff yang melayani pemesanan tiket
- (m) Mengidentifikasi jumlah *scheduledmovie* dalam Studio
- (n) Mengidentifikasi total studio dalam *cinema*
- (o) Mengidentifikasi total *booking* dalam setiap *scheduledmovie*

Transaction/Relation	(a)				(b)				(c)				(d)				(e)			
	I	R	U	D	I	R	U	D	I	R	U	D	I	R	U	D	I	R	U	D
Cinemas																				
Studios														✓						
CinemaStudios																				
StudioTypes																				
Seats														✓				✓		
Staff									✓	✓	✓			✓						
BookingMovie													✓	✓	✓			✓		
BookingMovieDetails																	✓	✓	✓	
Movies	✓	✓	✓			✓														
ProductionHouse		✓																		
ViewingCategory		✓																		
ScheduledMovie					✓	✓	✓	✓						✓						
Payment																				
PaymentTypes																				

Transaction/Relation	(f)				(g)				(h)				(i)				(j)			
	I	R	U	D	I	R	U	D	I	R	U	D	I	R	U	D	I	R	U	D
Cinemas										✓			✓	✓	✓					
Studios									✓	✓	✓									
CinemaStudios										✓										
StudioTypes																		✓		
Seats						✓														
Staff																				
BookingMovie		✓				✓														
BookingMovieDetails																				
Movies																				
ProductionHouse																				
ViewingCategory																				
ScheduledMovie						✓														
Payment	✓	✓	✓																	
PaymentTypes		✓																		

Transaction/Relation	(k)				(l)				(m)				(n)				(o)			
	I	R	U	D	I	R	U	D	I	R	U	D	I	R	U	D	I	R	U	D
Cinemas																				
Studios										✓										
CinemaStudios														✓						
StudioTypes																				
Seats																				
Staff						✓														
BookingMovie						✓												✓		
BookingMovieDetails																		✓		
Movies																				
ProductionHouse																				
ViewingCategory																				
ScheduledMovie										✓								✓		
Payment																				
PaymentTypes		✓																		

b. Choose Indexes

Tabel	Index	Nama Index
Cinema	CinemaID	CinemaIdx
Studio	StudioID	StudioIdx
CinemaStudio	CinemaID, StudioID	CinemaStudioIdx
StudioType	StudioTypeID	StudioTypeIdx
Seat	SeatID	SeatIdx
Staff	StaffID	StaffIdx
BookingMovie	BookingID	BookingMovieIdx
BookingMovieDetails	BookingID, SeatID	BookingMovieDetailsIdx
Movies	MovieID	MoviesIdx
ProductionHouse	ProductionHouseID	ProductionHouseIdx
ViewingCategory	ViewingCategoryID	ViewingCategoryIdx
ScheduledMovie	ScheduledMovieID	ScheduledMovieIdx
Payment	PaymentID	PaymentIdx
PaymentType	PaymentTypeID	PaymentTypeIdx

1. CREATE UNIQUE INDEX CinemaIdx ON Cinema(CinemaID)
2. CREATE UNIQUE INDEX StudioIdx ON Studio(StudioID)
3. CREATE UNIQUE INDEX CinemaStudioIdx ON
CinemaStudio(CinemaID, StudioID)
4. CREATE UNIQUE INDEX StudioTypeIdx ON
StudioTypes(StudioTypeID)
5. CREATE UNIQUE INDEX SeatIdx ON Seat(SeatID)
6. CREATE UNIQUE INDEX StaffIdx ON Staff(StaffID)
7. CREATE UNIQUE INDEX BookingMovieIdx ON
BookingMovie(BookingID)
8. CREATE UNIQUE INDEX BookingMovieDetailsIdx ON
BookingMovieDetails(BookingID, SeatID)
9. CREATE UNIQUE INDEX MoviesIdx ON Movies(MovieID)
10. CREATE UNIQUE INDEX ProductionHouseIdx ON
ProductionHouse(ProductionHouseID)

11. CREATE UNIQUE INDEX ViewingCategoryIdx ON
ViewingCategory(ViewingCategoryID)
12. CREATE UNIQUE INDEX ScheduledMovieIdx ON
ScheduledMovie(ScheduledMovieID)
13. CREATE UNIQUE INDEX PaymentIdx ON Payment(PaymentID)
14. CREATE UNIQUE INDEX PaymentTypeIdx ON
PaymentType(PaymentTypeID)

7.6.3. Step 5: Design User View

Tahapan ini dilakukan untuk merancang *user view* yang telah diidentifikasi selama tahapan *requirement collection and analysis* dari proses *database system development lifecycle*.

<u>User View</u>	<u>View</u>
Kasir User View	<i>ScheduledMovie View</i> <u>ScheduledMovie</u> ScheduledMovieID TimeStart ShowStartDate ShowEndDate <u>Movies</u> MovieID MovieName <u>Studios</u> StudioID StudioName
	<i>Seat View</i> <u>Seats</u> SeatID SeatNumber <u>Studios</u> StudioID StudioName
	<i>Movie View</i>

	<p><u>Movies</u> MovieID MovieName Genre</p> <p><u>ProductionHouse</u> ProductionHouseID ProductionHouseName</p> <p><u>ViewingCategory</u> ViewingCategoryID ViewingCategory ViewingCategoryDesc</p>
	<p><i>Booking Movie View</i></p> <p><u>BookingMovie</u> BookingID DateOfBooking StatusOfPayment</p> <p><u>BookingMovieDetails</u> BookingID SeatID</p>
	<p><i>Payment View</i></p> <p><u>Payment</u> PaymentID Nominal</p> <p><u>PaymentType</u> PaymentTypeID PaymentTypeName</p>
Bagian Penayangan <i>User View</i>	<p><i>Movie View</i></p> <p><u>Movies</u> MovieID MovieName Genre</p>

	<p><u>ProductionHouse</u> ProductionHouseID ProductionHouseName</p> <p><u>ViewingCategory</u> ViewingCategoryID ViewingCategory ViewingCategoryDesc</p> <hr/> <p><i>ScheduledMovie View</i></p> <p><u>ScheduledMovie</u> ScheduledMovieID TimeStart ShowStartDate ShowEndDate</p> <p><u>Movies</u> MovieID MovieName</p> <p><u>Studios</u> StudioID StudioName</p>
<p><i>Manajer User View</i></p>	<p><i>ScheduledMovie View</i></p> <p><u>ScheduledMovie</u> ScheduledMovieID TimeStart ShowStartDate ShowEndDate</p> <p><u>Movies</u> MovieID MovieName</p> <p><u>Studios</u> StudioID StudioName</p>

	<p><i>Seat View</i></p> <p><u>Seats</u></p> <p>SeatID</p> <p>SeatNumber</p> <p><u>Studios</u></p> <p>StudioID</p> <p>StudioName</p>
	<p><i>Movie View</i></p> <p><u>Movies</u></p> <p>MovieID</p> <p>MovieName</p> <p>Genre</p> <p><u>ProductionHouse</u></p> <p>ProductionHouseID</p> <p>ProductionHouseName</p> <p><u>ViewingCategory</u></p> <p>ViewingCategoryID</p> <p>ViewingCategory</p> <p>ViewingCategoryDesc</p>
	<p><i>Payment View</i></p> <p><u>Payment</u></p> <p>PaymentID</p> <p>Nominal</p> <p><u>PaymentType</u></p> <p>PaymentTypeID</p> <p>PaymentTypeName</p>
	<p><i>Staff View</i></p> <p><u>Staff</u></p> <p>StaffID</p> <p>StaffName</p>

	StaffType StaffGender SupervisorID StaffPhoneNumber StaffHourlyRate StaffSalary
	<i>Cinema View</i> <u>Cinema</u> CinemaID CinemaName CinemaLocation StudioQty
	<i>Studio and Type View</i> <u>Studio</u> StudioID StudioName Capacity <u>StudioType</u> StudioTypeID StudioTypeName Price <u>Cinema</u> CinemaID
	<i>Booking Movie View</i> <u>BookingMovie</u> BookingID DateOfBooking StatusOfPayment <u>BookingMovieDetails</u> BookingID SeatID

<p>Database Administrator User</p> <p>View</p>	<p><i>ScheduledMovie View</i></p> <p><u>ScheduledMovie</u></p> <p>ScheduledMovieID</p> <p>TimeStart</p> <p>ShowStartDate</p> <p>ShowEndDate</p> <p><u>Movies</u></p> <p>MovieID</p> <p>MovieName</p> <p><u>Studios</u></p> <p>StudioID</p> <p>StudioName</p>
	<p><i>Seat View</i></p> <p><u>Seats</u></p> <p>SeatID</p> <p>SeatNumber</p> <p><u>Studios</u></p> <p>StudioID</p> <p>StudioName</p>
	<p><i>Movie View</i></p> <p><u>Movies</u></p> <p>MovieID</p> <p>MovieName</p> <p>Genre</p> <p><u>ProductionHouse</u></p> <p>ProductionHouseID</p> <p>ProductionHouseName</p> <p><u>ViewingCategory</u></p> <p>ViewingCategoryID</p> <p>ViewingCategory</p>

	ViewingCategoryDesc
	<i>Payment View</i> <u>Payment</u> PaymentID Nominal <u>PaymentType</u> PaymentTypeID PaymentTypeName
	<i>Staff View</i> <u>Staff</u> StaffID StaffName StaffType StaffGender SupervisorID StaffPhoneNumber StaffHourlyRate StaffSalary
	<i>Cinema View</i> <u>Cinema</u> CinemaID CinemaName CinemaLocation StudioQty

	<p><i>Studio and Type View</i></p> <p><u>Studio</u></p> <p>StudioID</p> <p>StudioName</p> <p>Capacity</p> <p><u>StudioType</u></p> <p>StudioTypeID</p> <p>StudioTypeName</p> <p>Price</p> <p><u>Cinema</u></p> <p>CinemaID</p>
	<p><i>Booking Movie View</i></p> <p><u>BookingMovie</u></p> <p>BookingID</p> <p>DateOfBooking</p> <p>StatusOfPayment</p> <p><u>BookingMovieDetails</u></p> <p>BookingID</p> <p>SeatID</p>

7.6.1. Step 6: Design Security Mechanisms

Data	Kasir				Bagian Penayangan				Manajer Cabang				Database Administrator			
	I	R	U	D	I	R	U	D	I	R	U	D	I	R	U	D
Staff									X	X	X	X	X	X	X	X
Cinemas									X	X	X	X	X	X	X	X
Studios		X	X						X	X	X	X	X	X	X	X
StudioTypes		X							X	X	X	X	X	X	X	X
Seats		X	X							X			X	X	X	X
Movies		X			X	X	X	X		X			X	X	X	X
ScheduledMovie					X	X	X	X		X			X	X	X	X

BookingMovie	X	X	X							X			X	X	X	X
Payment	X	X								X			X	X	X	X
PaymentTypes		X											X	X	X	X
CinemaStudio		X						X	X	X	X	X	X	X	X	X
BookingMovieDetails	X	X	X					X	X				X	X	X	X
ProductionHouse					X				X				X	X	X	X
ViewingCategory		X							X				X	X	X	X

CHAPTER 7: User Interface

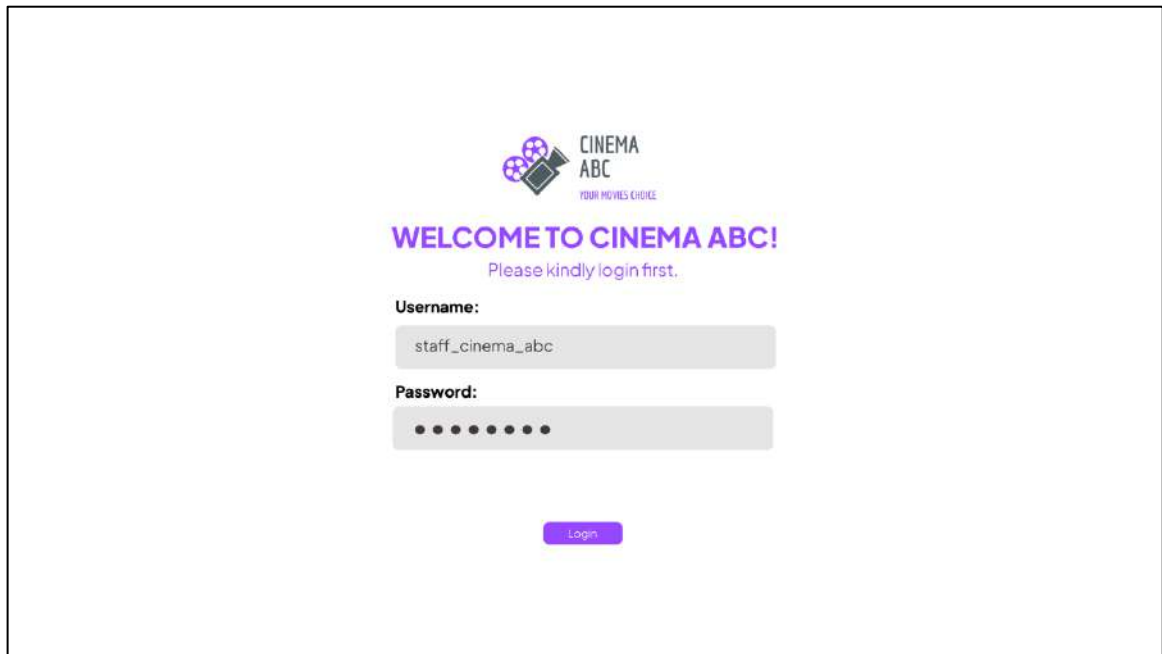


Figure 1. Halaman Login

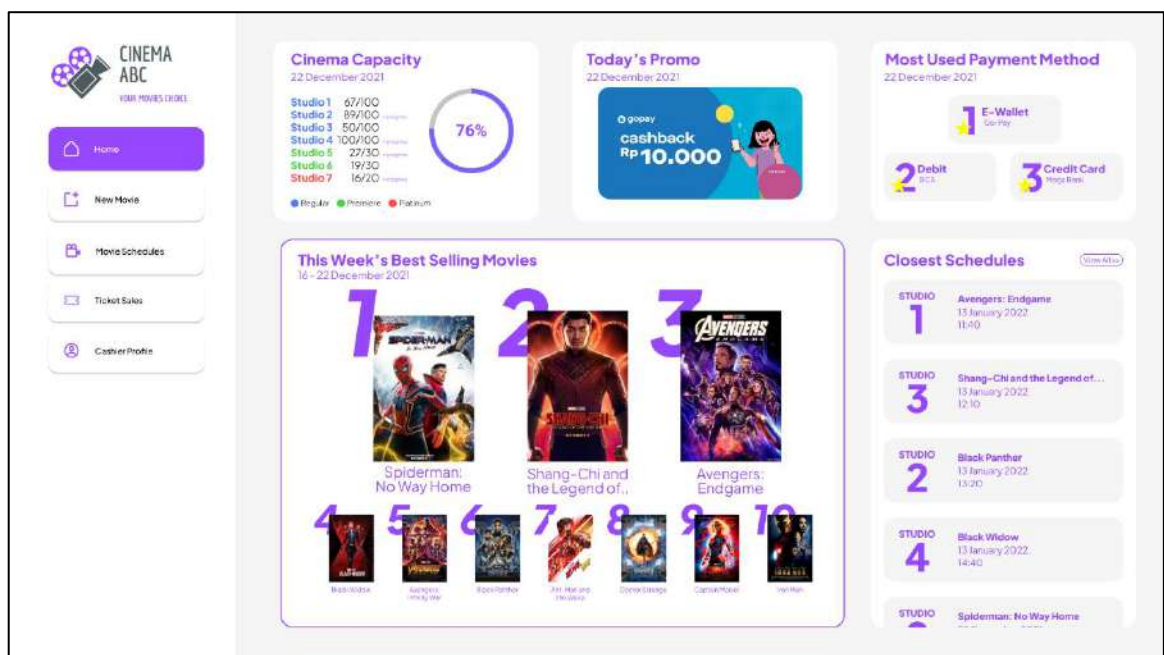


Figure 2. Halaman Home (Dashboard)

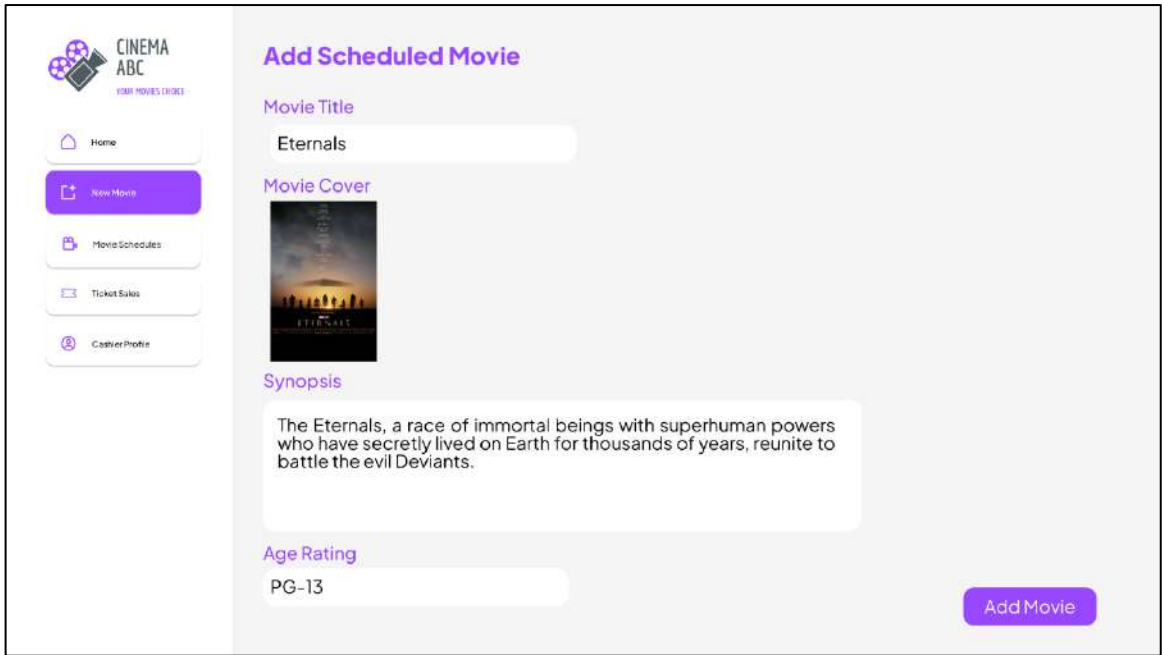


Figure 3. Add Scheduled Movie

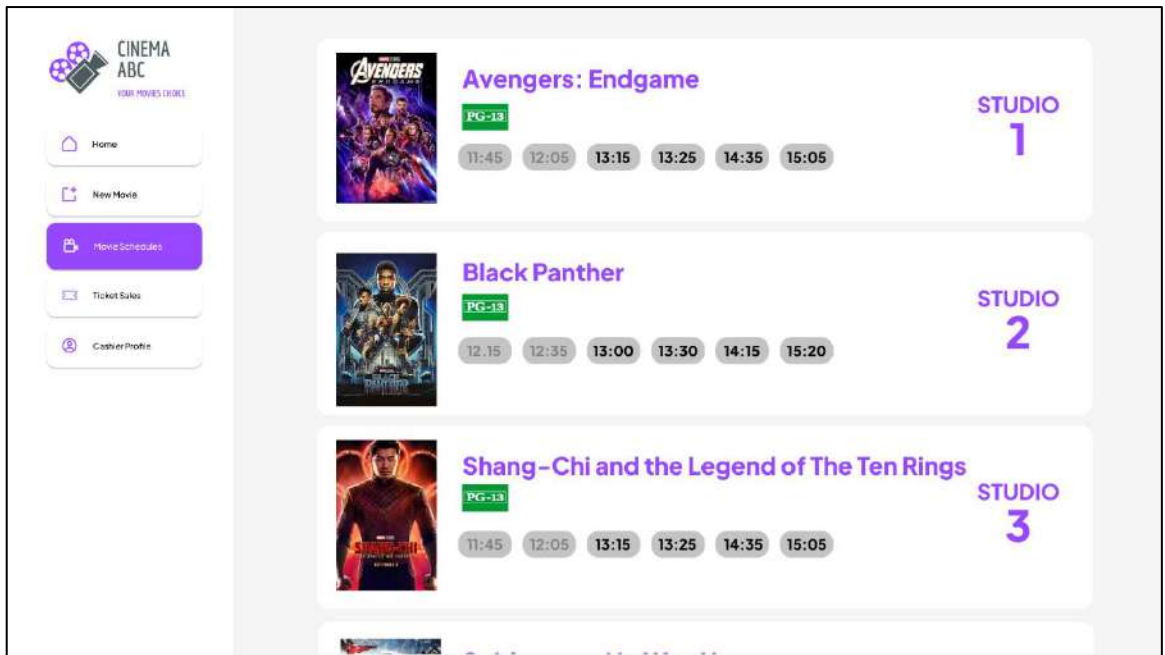


Figure 4. View Scheduled Movie

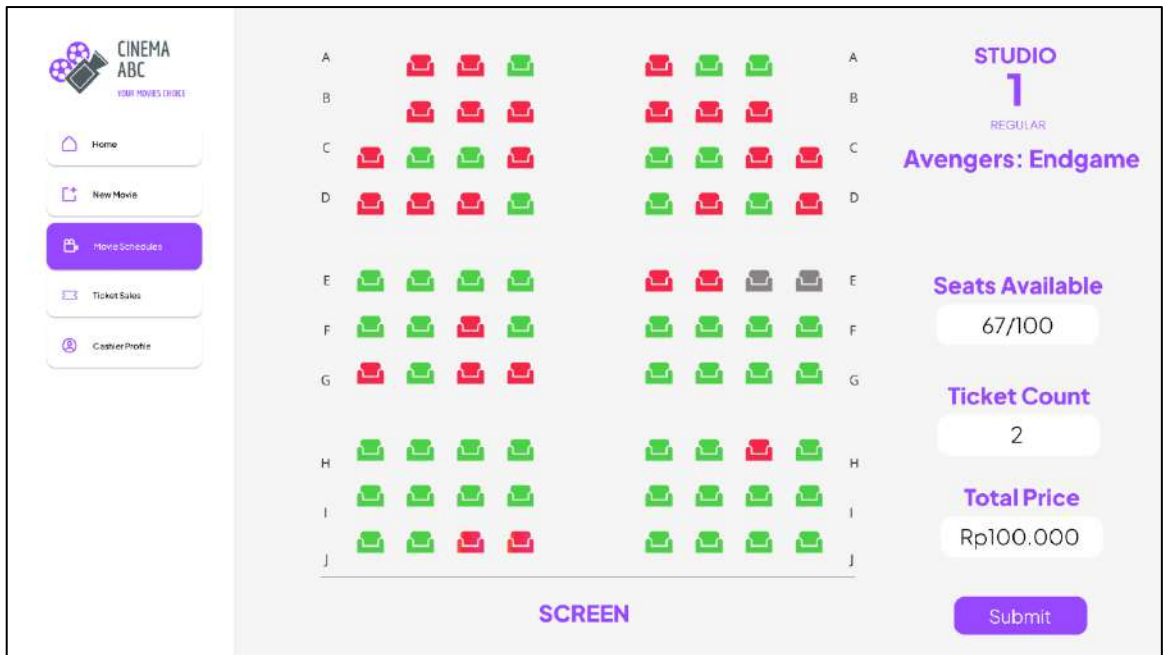


Figure 5. Seat Selection

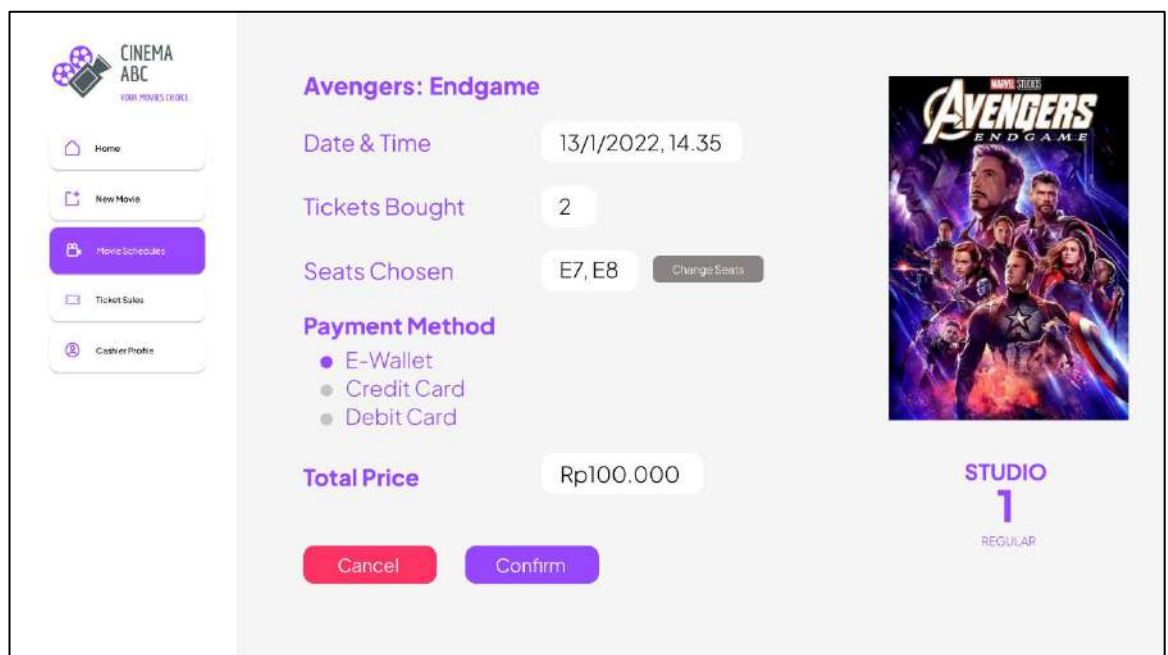


Figure 6. Payment

No.	Booking ID	Movie Title	Ticket Count	Date	Status	Staff in Charge
1	BOK4829	Avengers: Endgame	2	13/1/2022	Success	Rahmat
2	BOK4818	Avengers: Endgame	4	13/1/2022	Pending	Joni
3	BOK4816	Avengers: Endgame	1	13/1/2022	Cancelled	Desi
4	BOK4807	Avengers: Endgame	3	13/1/2022	Cancelled	Rahmat
5	BOK4806	Avengers: Endgame	2	13/1/2022	Success	Desi
6	BOK4801	Avengers: Endgame	2	13/1/2022	Pending	Kristanto
7	BOK4798	Avengers: Endgame	4	13/1/2022	Success	Joni
8	BOK4795	Avengers: Endgame	1	13/1/2022	Cancelled	Rahmat
9	BOK4794	Avengers: Endgame	3	13/1/2022	Success	Rosi
10	BOK4651	Avengers: Endgame	4	13/1/2022	Success	Kristanto

Figure 7. View Transaction

Avengers: Endgame

Booking ID: BOK4829

Studio: 1

Seats Chosen: E7, E8

Date of Booking: 13/1/2022

Payment Status: **Success** Go-Pay 082126*****




Figure 8. View Transaction Detail