

**DESAIN BASIS DATA**  
**“MINI PROJECT 2”**



Oleh:  
**Ikmal Ali Azhari**  
**(2309116087)**

**UNIVERSITAS MULAWARMAN**  
**FAKULTAS TEKNIK**  
**JURUSAN SISTEM INFORMASI**  
**2024**

## DAFTAR ISI

DAFTAR ISI .....	0
BAB 1 PENDAHULUAN.....	1
1.1. Latar Belakang .....	1
1.2. Tujuan.....	1
BAB 2 PEMBAHASAN .....	2
2.1. Supertype & Subtype .....	3
2.2. Optionality & Non - Optionality .....	3
2.3. Transferability & Non – Transferability.....	4
2.4. Arc, Hierarki, Rekursif.....	5
2.5. ERD, Entitas, Atribut, Kardinalitas, ERDish, Relational.....	7
BAB 3 PENUTUP .....	14
3.1. Kesimpulan .....	14

# **BAB 1**

## **PENDAHULUAN**

### **1.1. Latar Belakang**

Perkembangan teknologi informasi yang pesat telah membawa perubahan signifikan dalam berbagai aspek kehidupan, termasuk dalam industri permainan digital. Salah satu platform permainan yang populer adalah Roblox, yang memungkinkan pengguna untuk tidak hanya bermain, tetapi juga menciptakan konten mereka sendiri. Dalam konteks ini, pengelolaan basis data menjadi sangat penting untuk memastikan bahwa sistem dapat berjalan dengan efisien dan efektif.

Desain basis data yang baik merupakan fondasi utama dalam pengembangan sistem informasi yang andal. Hal ini melibatkan pengorganisasian data secara sistematis sehingga dapat mendukung kebutuhan operasional dan strategis organisasi. Konsep-konsep seperti supertype dan subtype, opsionalitas, transferabilitas, serta struktur hierarki dan rekursif adalah elemen kunci dalam merancang basis data yang kompleks namun terstruktur dengan baik.

Pada proyek mini ini, fokus diberikan pada penerapan konsep-konsep tersebut dalam desain basis data untuk sebuah platform permainan. Dengan memahami dan menerapkan konsep-konsep ini, diharapkan dapat dihasilkan sebuah model basis data yang tidak hanya memenuhi kebutuhan fungsional, tetapi juga fleksibel untuk dikembangkan lebih lanjut sesuai dengan perkembangan bisnis dan teknologi.

### **1.2. Tujuan**

Tujuan dari proyek mini ini adalah untuk mengimplementasikan konsep-konsep dasar dalam desain basis data, serta menggambarkan bagaimana konsep tersebut dapat diterapkan dalam sistem yang lebih besar. Dalam proyek ini, fokus diberikan pada pengembangan struktur basis data yang tidak hanya mendukung penyimpanan data yang efisien tetapi juga memungkinkan fleksibilitas dalam relasi antar entitas.

Salah satu tujuan utama adalah mengimplementasikan konsep supertype dan subtype, di mana entitas umum (supertype) memiliki atribut yang diwarisi oleh entitas spesifik (subtype). Penerapan konsep ini bertujuan untuk meminimalisir redundansi data dan memudahkan pengelolaan entitas yang memiliki karakteristik berbeda namun saling terkait.

Selain itu, konsep opsionalitas dan non-opsionalitas juga menjadi fokus dalam proyek ini. Opsionalitas merujuk pada apakah suatu entitas wajib memiliki relasi dengan entitas lain atau tidak. Dengan memahami opsionalitas, sistem dapat dirancang agar lebih fleksibel, memungkinkan beberapa entitas memiliki relasi yang tidak selalu harus ada.

Konsep transferabilitas dan non-transferabilitas juga diterapkan untuk menunjukkan bagaimana atribut atau relasi dapat dipindahkan antara entitas. Dalam beberapa kasus, atribut tertentu mungkin perlu ditransfer dari satu entitas ke entitas lain, sedangkan pada kasus lain, relasi tersebut harus tetap tidak dapat dipindahkan.

Akhirnya, proyek ini juga bertujuan untuk memvisualisasikan relasi-relasi yang kompleks dalam bentuk ERD (Entity-Relationship Diagram), yang merupakan alat yang efektif dalam mendesain dan memahami struktur basis data.

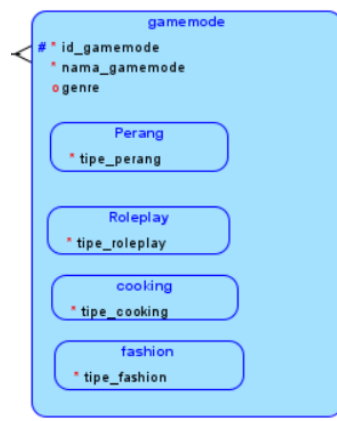
## **BAB 2**

### **PEMBAHASAN**

## 2.1. Supertype & Subtype

**Supertype** adalah entitas umum yang memiliki atribut dan relasi yang sama dengan beberapa entitas lain (subtype).

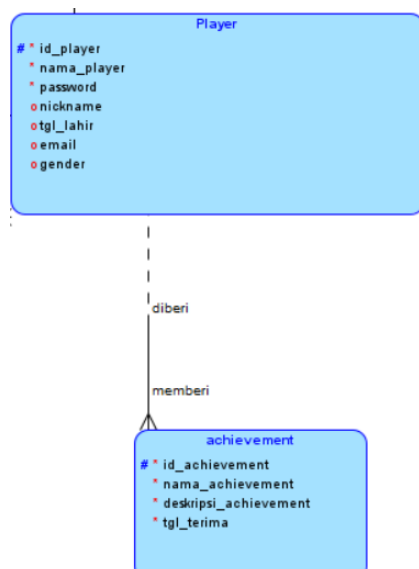
**Subtype** adalah entitas yang mewarisi atribut dan relasi dari supertype, tetapi juga memiliki atribut atau relasi tambahan yang khusus.



Gambar disamping adalah contoh dari supertype dan subtype yang Dimana supertype nya adalah gamemode dan subtypenya adalah Perang, Roleplay, Cooking, Fashion

## 2.2. Optionality & Non - Optionality

### Optionality

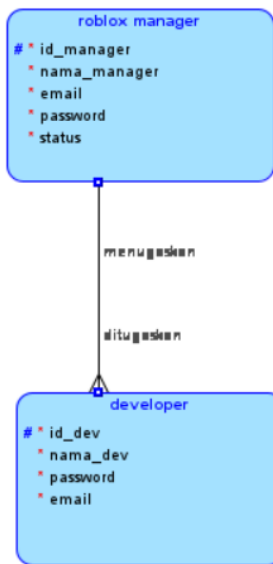


merujuk pada apakah keberadaan suatu relasi antara dua entitas adalah wajib (mandatory) atau tidak (optional). Jika suatu entitas bisa ada tanpa relasi dengan entitas lain, maka relasi tersebut bersifat optional.

Gambar disamping adalah contoh opsionalitas yaitu karena setiap satu atau banyak achievement harus memberi kepada player, tetapi setiap satu player tidak harus diberi achievement.

## Non-Optionality

adalah situasi di mana keberadaan relasi antara dua entitas adalah wajib. Contohnya, jika setiap **Order** harus memiliki **Customer**, maka relasi tersebut bersifat non-optional.

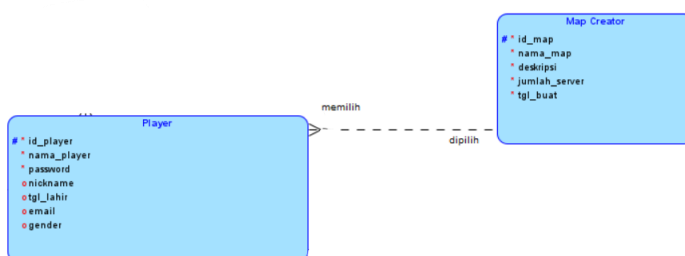


Gambar disamping adalah contoh non-optionality karena setiap satu manager harus menugaskan satu atau banyak developer, dan satu atau banyak developer harus ditugaskan oleh satu manager

## 2.3. Transferability & Non – Transferability

### Transferability

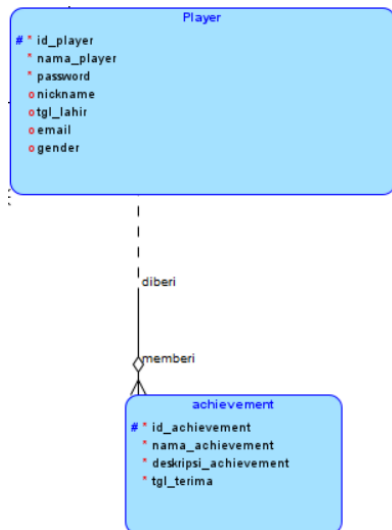
menunjukkan bahwa suatu atribut atau relasi dapat dipindahkan atau dialihkan dari satu entitas ke entitas lain. Dalam konteks hubungan antar entitas, ini berarti bahwa tanggung jawab atau atribut bisa berpindah antara entitas.



Gambar di atas adalah contoh transferability karena Player dapat dipindahkan ke Map Creator dengan menjadikan FK id\_player di entitas Map Creator sebagai id\_player yang membuat Map nya

### Non-Transferability

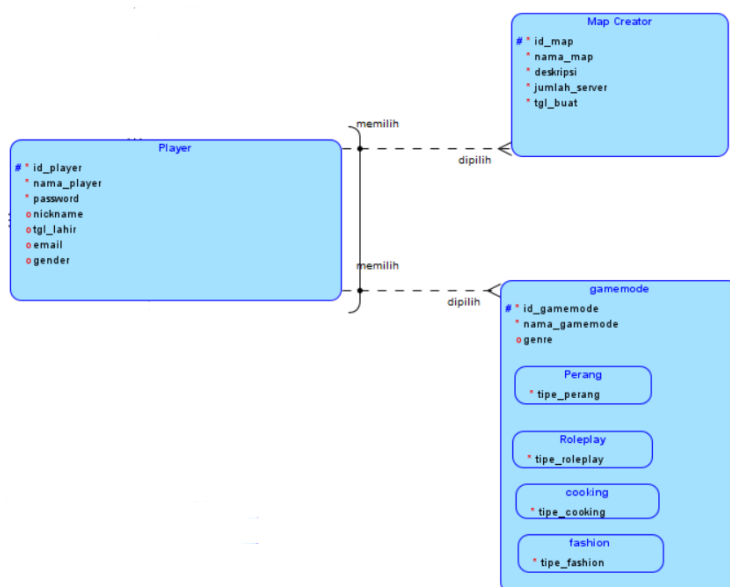
berarti bahwa atribut atau relasi tidak dapat dipindahkan atau dialihkan.



gambar disamping adalah contoh non transferability karena penghargaan yang diberikan kepada sang player adalah tetap untuk player itu sendiri, tidak bisa player memberi penghargaan yang sudah di capai ke player lain.

## 2.4. Arc, Hierarki, Rekursif

Arc dalam konteks ERD merujuk pada garis atau kurva yang menghubungkan dua entitas dan menunjukkan relasi antara keduanya. Arc dapat digunakan untuk menunjukkan relasi yang lebih kompleks, seperti hubungan antara entitas dalam struktur hierarki.

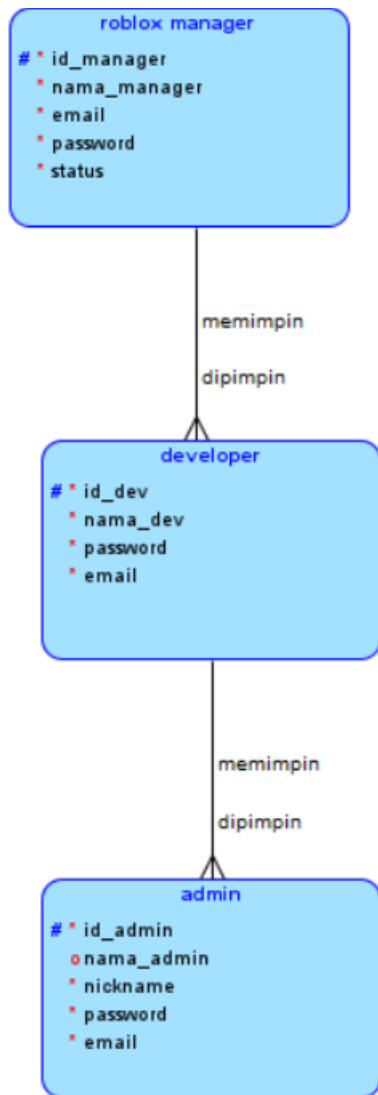


Gambar disamping adalah contoh Arc yang Dimana player terhubung dengan 2 relasi yaitu Map Creator dan Gamemode, tetapi tidak bisa memilih 2 pilihan, hanya bisa memilih 1 pilihan antara Map Creator atau gamemode.

## Hierarki

Hierarki merupakan sebuah kondisi yang merujuk pada struktur tata hubungan antar data yang diorganisir secara bertingkat, di mana data tingkat atas merepresentasikan konsep yang lebih

umum atau agregat, sementara data tingkat bawah adalah detail atau komponen spesifik dari konsep tersebut

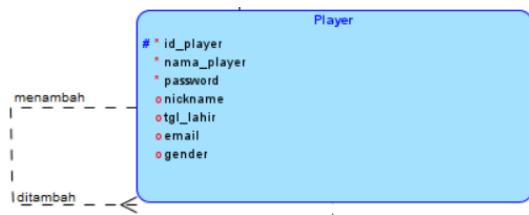


gambar disamping adalah contoh dari hierarki yang dimaksud adalah seperti disamping ini merupakan turunan tingkatan/jabatan dari tertinggi dan yang terendah

## Rekursif

Rekursif merupakan sebuah kondisi di mana data dapat merepresentasikan diri sendiri atau entitas serupa dalam struktur data. Kondisi ini memungkinkan hubungan yang kompleks seperti dalam organisasi hirarkis, jejaring, atau struktur yang melibatkan hubungan kompleks antar entitas dan fleksibel, di mana entitas dapat memiliki hubungan dengan dirinya sendiri atau entitas serupa.

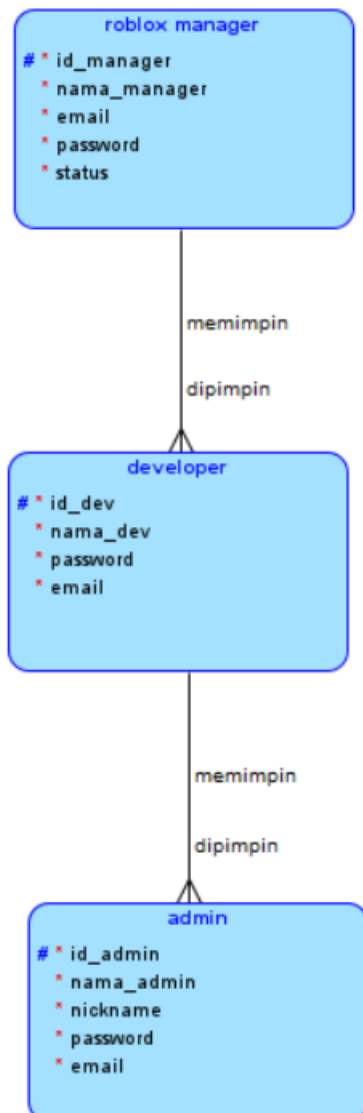


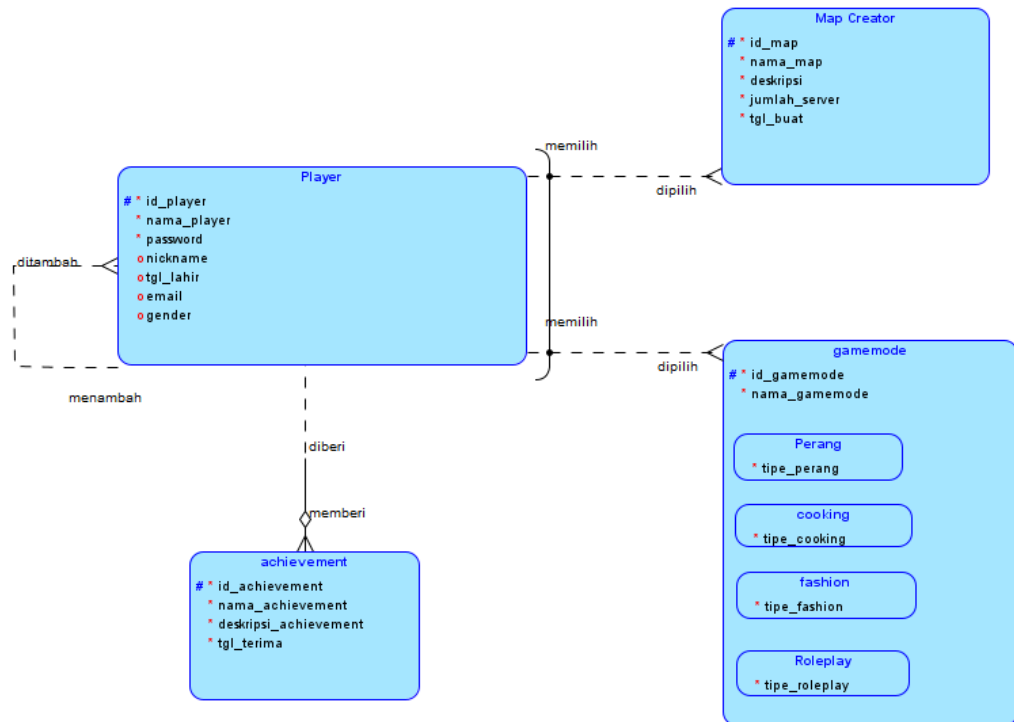


gambar diatas adalah contoh rekursif (relasi eksklusif) pada entitas player, di contoh konteks ini adalah Setiap satu Player(Anda) dapat menambah satu atau banyak Player(teman), dan satu atau banyak Player(teman) dapat ditambah satu Player(Anda)

## 2.5. ERD, Entitas, Atribut, Kardinalitas, ERDish, Relational

### ERD data modeler





## Entitas & Atribut

- Roblox manager:
  - **PK** Id\_manager NOT NULL CHAR10
  - Nama\_manager NOT NULL VARCHAR50
  - Email NOT NULL VARCHAR100
  - Password NOT NULL VARCHAR50
  - Status NOT NULL VARCHAR20
- Developer
  - **PK** Id\_dev CHAR10
  - Nama\_manager NOT NULL VARCHAR50
  - Email NOT NULL VARCHAR100
  - Password NOT NULL VARCHAR50
  - **FK** id\_manager NOT NULL CHAR10
- Admin
  - **PK** Id\_dev CHAR10
  - Nama\_admin NOT NULL VARCHAR50
  - Nickname NOT NULL VARCHAR50
  - Email NOT NULL VARCHAR100

- Password NOT NULL VARCHAR50
  - **FK** id\_dev NOT NULL CHAR10
- 
- Player
    - **PK** Id\_player NOT NULL CHAR10
    - Nama\_player NOT NULL VARCHAR50
    - Password NOT NULL VARCHAR50
    - nickname NULL VARCHAR50
    - tgl\_lahir NULL DATE
    - Email NOT NULL VARCHAR100
    - Gender NULL VARCHAR20
    - **FK** id\_dev NOT NULL CHAR10
    - **FK** id\_player NOT NULL CHAR10
- 
- Achievement
    - **PK** id\_achievement NOT NULL CHAR10
    - Nama\_Achievement NOT NULL VARCHAR50
    - Deskripsi\_Achievement NOT NULL VARCHAR50
    - Tgl\_terima NOT NULL DATE
    - **FK** id\_player NOT NULL CHAR10
- 
- Map Creator
    - **PK** id\_map NOT NULL CHAR10
    - Nama\_map NOT NULL VARCHAR50
    - Deskripsi NOT NULL VARCHAR 250
    - Jumlah\_server NOT NULL INTEGER
    - Tgl\_buat NOT NULL DATE
    - **FK** id\_player NOT NULL CHAR10
- 
- Gamemode
    - **PK** id\_gamemode NOT NULL CHAR10
    - Nama\_gamemode NOT NULL VARCHAR50

- SUBTYPE:
  - Perang: tipe\_perang NOT NULL VARCHAR30
  - Roleplay: tipe\_roleplay NOT NULL VARCHAR30
  - Cooking: tipe\_cooking NOT NULL VARCHAR30
  - Fashion: tipe\_fashion NOT NULL VARCHAR30

### **ERDish**

-Setiap satu dan hanya satu Roblox Manager harus memimpin satu atau banyak Developer, dan satu atau banyak Developer harus dipimpin oleh satu dan hanya satu Roblox Manager.

-Setiap satu dan hanya satu Developer harus memimpin satu atau banyak Admin, dan satu atau banyak Admin harus dipimpin oleh satu dan hanya satu Developer.

-Setiap satu dan hanya satu Player(Anda) dapat menambah satu atau banyak Player(teman), dan satu atau banyak Player(teman) dapat ditambah oleh satu dan hanya satu Player(Anda) (Rekursif).

-Setiap satu dan hanya satu Player dapat memilih satu atau banyak Map Creator, dan satu atau banyak Map Creator dapat dipilih oleh satu dan hanya satu Player

- Setiap satu dan hanya satu Player dapat memilih satu atau banyak Gamemode, dan satu atau banyak Gamemode dapat dipilih oleh satu dan hanya satu Player

-Setiap satu dan hanya satu player dapat diberi oleh satu atau banyak Achievement, dan satu atau banyak Achievement harus memberi satu dan hanya satu Player

### **Kardinalitas**

-Roblox manager **1 Mandatory : N Mandatory** Developer

-Developer **1 Mandatory : N Mandatory** Admin

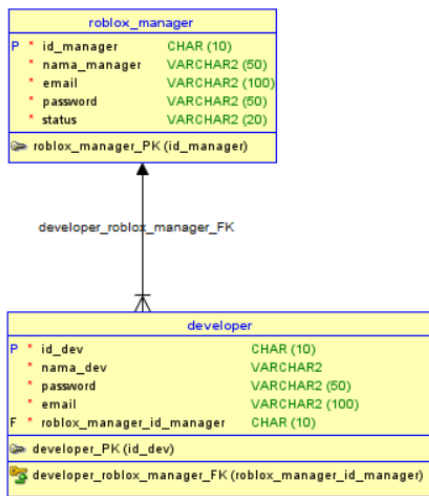
-Player **1 Optionality : N Optionality** Player (Rekursif)

-Player **1 Optionality : N Optionality** Map Creator (Arc)

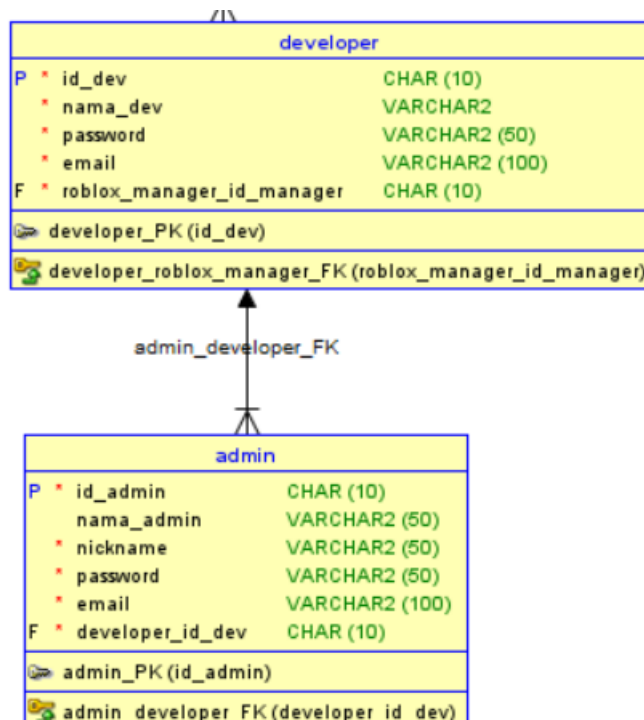
-Player **1 Optionality : N Optionality** Gamemode (Arc)

-Player **1 Optionality : N Optionality (Non-Transfer)** Achievement

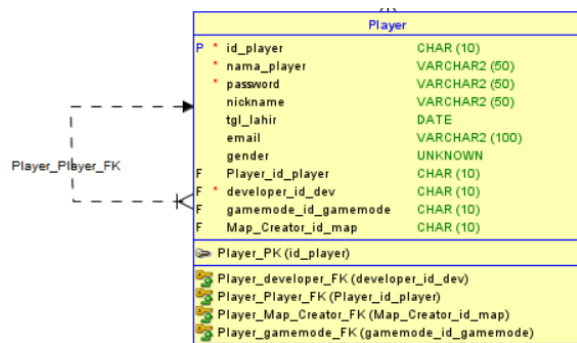
## Relational



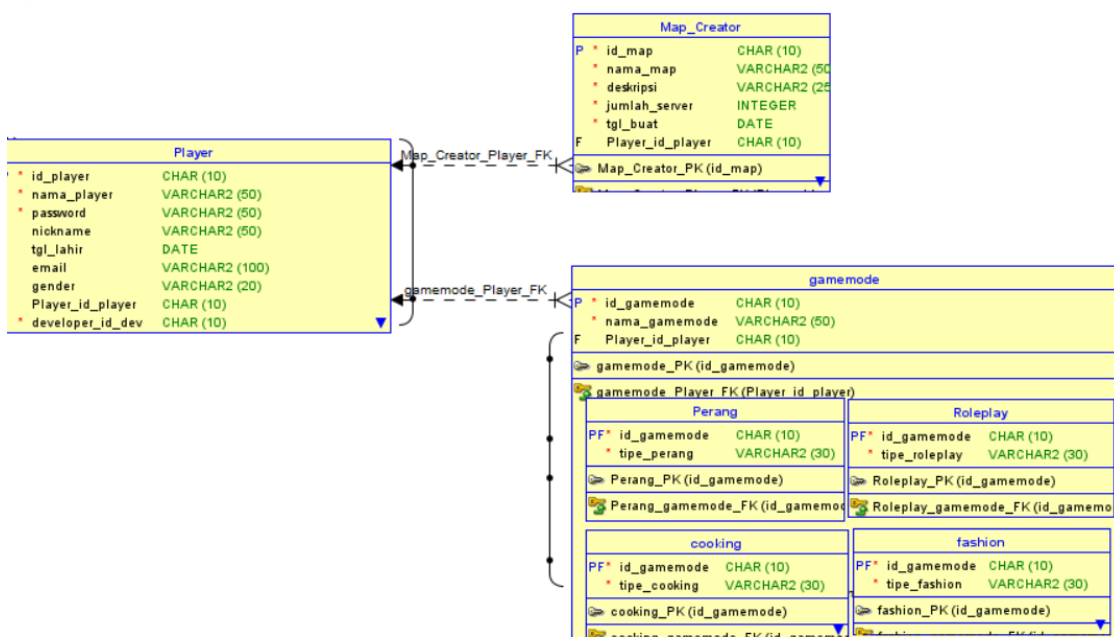
Roblox manager ke developer yaitu agar developer bisa memberi job ke developer dengan id\_manager di developer maka manager bisa melihat developer mana saja yang sudah di beri tugas oleh sang manager



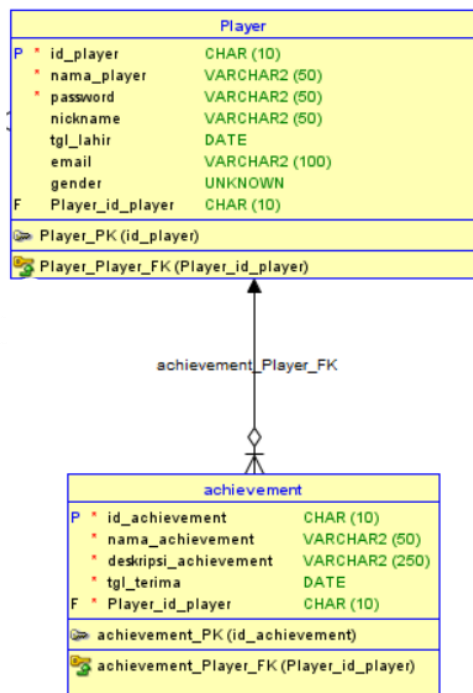
Developer ke admin, karena developer memberikan arahan kepada admin untuk memperbagus game roblox dan juga bisa untuk memantau para player yang menggunakan cheat dan exploiting



rekursif player ini konteksnya adalah seperti add friend pada umumnya yaitu dengan menggunakan rekursif maka ketika ada player yang ingin menambahkan pertemanan kepada player lain di system hanya perlu menggunakan dari id\_player itu sendiri.



Map Creator dan Gamemode adalah Arc dari player yang Dimana disini dimaksud adalah player hanya bisa memilih salah satu dari kedua pilihan tersebut yaitu Map Creator atau gamemode, jika memilih keduanya maka tidak bisa karena dari roblox sendiri tidak bisa membuka 2 pilihan secara bersamaan, dan Map Creator mengambil id\_player sebagai FK nya agar sebagai nama pembuat Mapnya, dan Gamemode mengambil id\_player sebagai FK nya untuk menampilkan nama player yang sedang bermain.



Relasi player ke achievement adalah agar bisa melihat player mana saja yang sudah menerima achievement yang sudah di sediakan dengan memasukkan `id_player` ke entitas achievement.

## **BAB 3**

### **PENUTUP**

#### **3.1. Kesimpulan**

Berdasarkan hasil proyek mini ini, dapat disimpulkan bahwa penerapan konsep-konsep dasar dalam desain basis data memberikan dampak yang signifikan terhadap efisiensi dan efektivitas pengelolaan data. Konsep supertype dan subtype memungkinkan pengelompokan entitas berdasarkan atribut umum, sehingga desain basis data menjadi lebih terorganisir dan mengurangi redundansi. Dengan menggunakan pendekatan ini, perancangan basis data tidak hanya menjadi lebih rapi tetapi juga lebih mudah untuk diperluas jika diperlukan.

Penggunaan opsionalitas dan non-opsionalitas dalam relasi antar entitas memberikan fleksibilitas dalam desain, memungkinkan hubungan antar entitas yang tidak selalu wajib, namun tetap konsisten dalam sistem. Hal ini penting untuk mendukung berbagai skenario operasional yang mungkin terjadi dalam sebuah sistem.

Selanjutnya, penerapan transferabilitas dan non-transferabilitas menunjukkan bagaimana desain basis data dapat mencerminkan aturan bisnis yang berlaku. Dalam beberapa kasus, atribut atau tanggung jawab dapat dipindahkan antara entitas, sementara dalam situasi lain, relasi tersebut harus tetap tidak dapat dipindahkan, menjaga integritas data dan proses yang terkait.

Dengan adanya ERD yang menggambarkan keseluruhan relasi antar entitas, desain basis data yang dihasilkan menjadi lebih mudah dipahami dan diimplementasikan. Secara keseluruhan, proyek ini berhasil mencapai tujuan yang ditetapkan dan memberikan pemahaman yang lebih baik tentang pentingnya desain basis data dalam pengembangan sistem informasi yang andal dan skalabel.