



# Fullstack Web Challenge

**Gold** - Chapter 6

---



## SKILL METRICS

Kemampuan teknis  
yang akan kamu pelajari

01

## ACCEPTANCE CRITERIA

Kriteria yang harus  
kamu penuhi untuk  
dapat melewati chapter

02

## GAME DESIGN DOCUMENT

Hal yang akan kamu buat  
untuk dapat melewati  
chapter

03



### SKILL METRICS

1. Database
  - PostgreSQL
  - MongoDB (No SQL)
2. SQL
  - DDL (Data Definition Language)
  - DML (Data Manipulation Language)
3. ORM (Object Relational Mapping)
  - Migration
  - Model definition
  - Querying
  - Association
4. Implementasi di Express
  - View engine
  - Restful API (Json)

### DELIVERY

1. Melakukan operasi query didalam konsol DBMS, baik itu PostgreSQL (psql) maupun MongoDB (mongo shell)
2. Membandingkan MongoDB dengan SQL
3. Melakukan instalasi adapter dan melakukan konfigurasi untuk Postgresql (Sequelize) maupun MongoDB (mongoose)
4. Membuat model tanpa relasi dan menjelaskan hubungannya dengan table di database
5. Melakukan database migration dan menjelaskan apa kegunaanya
6. Membuat basic CRUD dengan menggunakan HTTP request
7. Membuat model baru yang memiliki relasi dengan model pertama
8. Membuat tampilan untuk menampilkan data berdasarkan model dengan menggunakan template engine (EJS)
9. Membuat tampilan untuk melakukan Create, Update, Delete dengan menggunakan template engine (EJS)

### CRITERIA

1. Mengetahui perbedaan arsitektur Monolitik dan Microservice secara garis besar
2. Menggunakan package yang diinstall dengan menggunakan NPM/YARN
3. Serving static HTML, static file dan json ke browser
4. Menggunakan metode routing untuk memisahkan tiap halaman
5. Menggunakan middleware untuk akun statis
6. Menerapkan view engine untuk menampilkan HTML File
7. Menyediakan endpoint dengan type data JSON dari data statis
8. Membuat dan melakukan import export module
9. Push ke git untuk di-review.



### Di challenge ini, yang bakal kamu lakuin adalah:

1. Buatlah monolith dashboard sederhana dengan menggunakan view engine.
2. Super User Authentication login dashboard menggunakan data statis.
3. Membuat minimal 3 tabel contohnya:
  - Tabel user\_game
  - Tabel user\_game\_biodata
  - Tabel user\_game\_history
4. Data pada monolith dashboard sudah terkoneksi dengan database SQL dan memiliki metode sebagai berikut:
  - Create data user game
  - Read data user game
  - Update data user game
  - Delete data user game
5. Merancang dan mengimplementasikan skema database yang mengaitkan ketiga tabel tersebut dengan user\_id sebagai Parent Key(PK).
6. Membuat RESTFUL API satu endpoint dengan methods CRUD (create, read, delete, update).
7. Database attribute ditentukan sendiri oleh siswa.

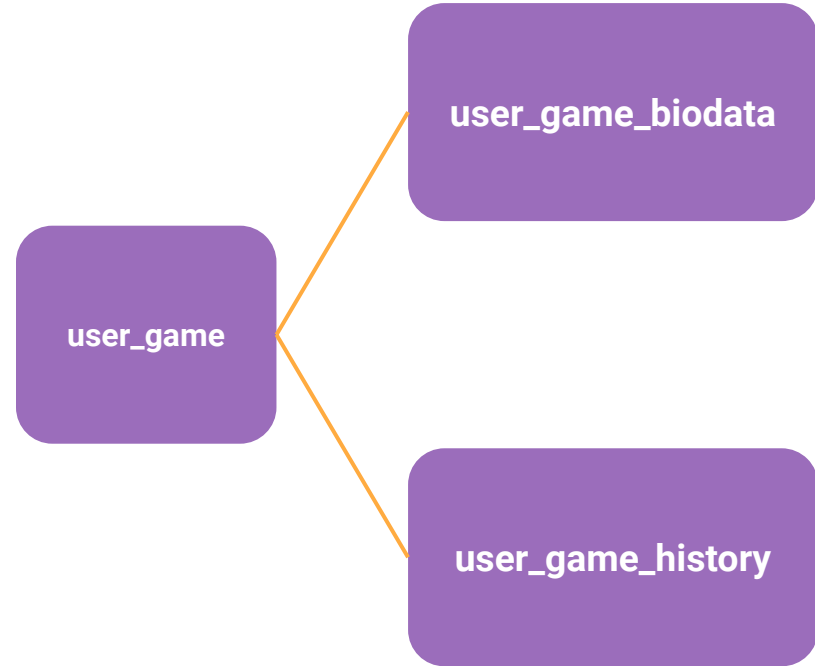
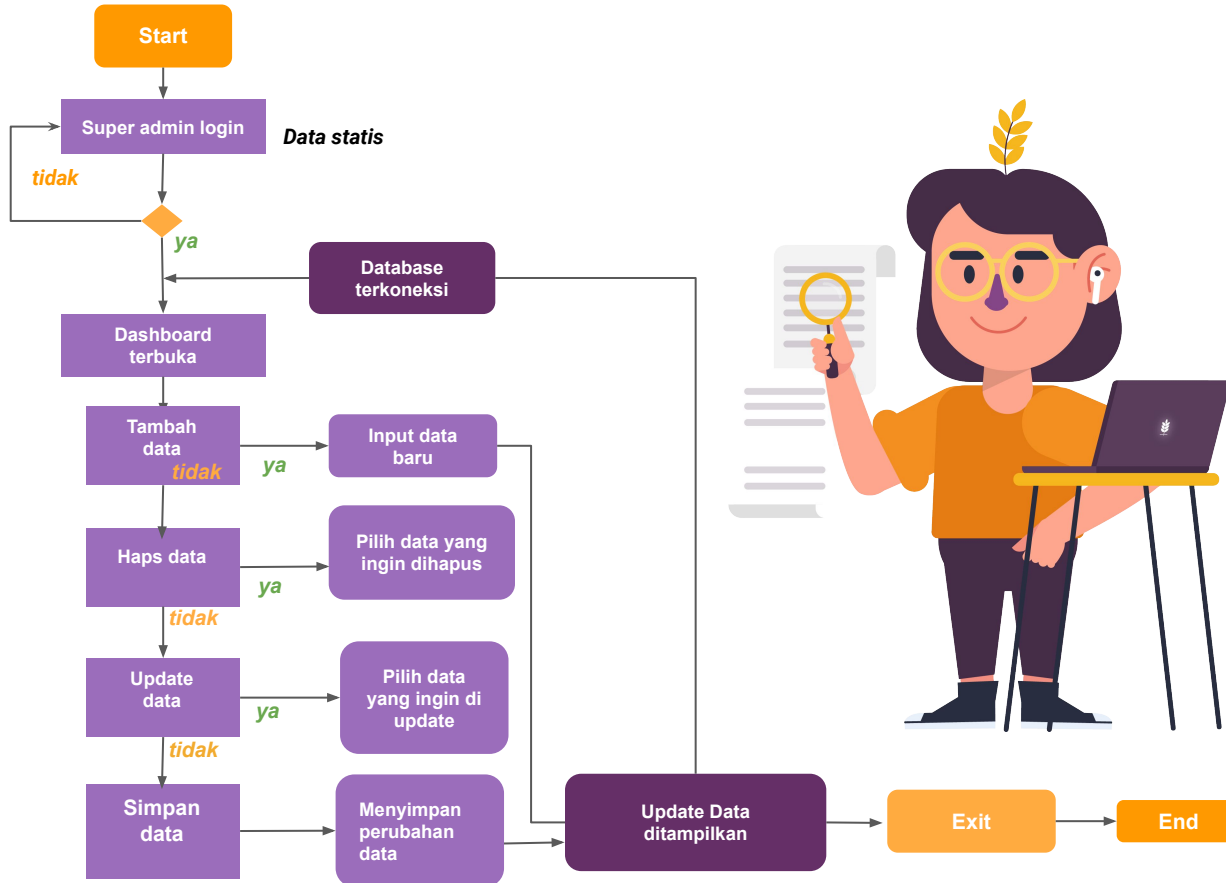




Table	Attribute	Detail
user_game	NOT NULL	Table untuk menyimpan data primer user contohnya ID, USERNAME, PASSWORD
user_game_biodata	NULL	Table untuk menyimpan biodata user
user_game_history	NULL	Table untuk menyimpan history dari aktifitas user game, contohnya list waktu dan skor dalam bermain game





Terima  
Kasih