1. BCM dari Distribusi Digital Game:

A screenshot of a computer screen

Description automatically generated

1. Dari BCM Distribusi digital game, saya mengambil Digital Storefront Operations sebagai domain
2. Dengan subdomain dan *functions* sebagai berikut:

**3.1 Core Capability: Family Sharing**

* **Family Game Library:** Fitur ini memungkinkan satu anggota keluarga yang membeli game untuk berbagi game tersebut dengan anggota keluarga lainnya. Setiap anggota keluarga bisa mengakses perpustakaan game yang sama tanpa harus membeli ulang game untuk setiap akun. Fitur ini memberikan kemudahan dan penghematan biaya bagi keluarga yang ingin bermain game bersama.
* **Account Permissions & Restrictions:** Fungsi ini memberikan kontrol penuh kepada pemegang akun utama (misalnya, orang tua) untuk mengatur izin dan batasan pada game yang bisa diakses oleh anggota keluarga lain, terutama anak-anak. Dengan fitur ini, orang tua dapat memastikan bahwa hanya game yang sesuai dengan usia atau yang mereka setujui yang dapat dimainkan oleh anak-anak mereka.
* **Shared Game Progress:** Fitur ini memungkinkan setiap anggota keluarga memiliki progres game tersendiri meskipun mereka memainkan game yang sama. Hal ini mencegah konflik data penyimpanan dan memungkinkan semua orang menikmati pengalaman bermain yang dipersonalisasi, tanpa khawatir progres mereka terganggu oleh orang lain.
* **Multi-Device Sharing:** Anggota keluarga dapat mengakses game yang dibagikan melalui berbagai perangkat, seperti PC, konsol, atau perangkat mobile, yang terhubung dalam satu akun keluarga. Ini memungkinkan fleksibilitas bermain di berbagai perangkat, baik di rumah maupun saat bepergian.
* **Simultaneous Play Limitations:** Fungsi ini mengelola dan membatasi jumlah anggota keluarga yang dapat memainkan game yang sama secara bersamaan. Batasan ini diterapkan untuk mencegah penyalahgunaan seperti banyak orang bermain secara simultan dengan satu lisensi, sehingga tetap sesuai dengan kebijakan lisensi game.

**3.2 Generic Capability**

* **User Registration & Login**: Proses dasar pendaftaran dan login akun untuk pengguna. Ini termasuk kemampuan untuk membuat akun keluarga yang terhubung, di mana satu akun utama bisa mengelola akun-akun anak atau anggota keluarga lainnya. Fungsi ini adalah fitur yang sangat standar pada hampir semua platform digital.
* **Search & Filtering:** Fungsi pencarian dan penyaringan memungkinkan pengguna mencari game berdasarkan berbagai kriteria seperti genre, harga, rating, dan kategori ramah keluarga. Meskipun penting, ini adalah kapabilitas umum yang diharapkan pada semua platform distribusi digital.
* **Checkout & Payment Integration:** Proses pembayaran dan checkout yang terintegrasi untuk pembelian game. Mendukung berbagai metode pembayaran seperti kartu kredit, PayPal, dan lainnya. Fitur ini membantu menyederhanakan proses pembelian game, tetapi bukan sesuatu yang unik bagi platform.
* **Game Recommendations:** Mesin rekomendasi yang memberikan saran game kepada pengguna berdasarkan tren umum, penjualan populer, dan ulasan dari pengguna lain. Namun, ini tidak secara spesifik disesuaikan untuk kebutuhan keluarga, jadi sifatnya umum.
* **Reviews & Ratings:** Fitur yang memungkinkan pengguna memberikan ulasan dan rating pada game yang mereka mainkan. Fungsi ini juga membantu pengguna lain untuk memilih game berdasarkan pengalaman dan pendapat dari komunitas. Namun, ini adalah fungsi standar di hampir setiap platform distribusi game.

**3.3 Supporting Capability**

* **Game Library Management:** Fungsi ini memastikan game yang tersedia di platform diatur dengan baik, termasuk menampilkan kategori yang tepat, melabeli game baru, dan menonjolkan game ramah keluarga. Ini adalah proses back-end yang mendasar dalam memastikan perpustakaan game tetap terorganisir dan mudah diakses oleh pengguna.
* **Catalog Updates:** Fungsi ini bertugas menambahkan game baru ke platform secara teratur, memperbarui informasi game yang sudah ada, seperti harga, deskripsi, dan ketersediaan. Ini adalah bagian dari pemeliharaan platform agar konten tetap relevan dan terkini.
* **Localization & Multilingual Support:** Fitur yang mendukung berbagai bahasa dan penyesuaian harga berdasarkan wilayah geografis. Ini penting untuk menarik audiens global, namun tidak memberikan keunggulan yang signifikan dibandingkan platform lain yang sudah menyediakan fitur ini.
* **Technical Support:** Layanan pelanggan yang membantu pengguna dalam menyelesaikan masalah teknis terkait dengan akun, fitur berbagi keluarga, akses game, atau kesalahan dalam platform. Dukungan ini biasanya dilakukan melalui berbagai saluran seperti chat, email, atau telepon.
* **Family Setup Assistance:** Fitur ini memberikan panduan atau tutorial kepada pengguna untuk membantu mereka menyiapkan fitur berbagi keluarga, mengelola izin akun, dan memanfaatkan kapabilitas platform dengan lebih baik. Ini adalah layanan tambahan yang bermanfaat, namun tidak memerlukan inovasi teknis yang tinggi.

1. **Proposed Architecture**

A diagram of a business process

Description automatically generated

A screenshot of a computer

Description automatically generatedSequence diagram yang dibuat adalah seperti berikut:

A screenshot of a computer

Description automatically generatedA diagram of a company

Description automatically generated

1. **Teknologi yang Diajukan**

Presentation layer : Next.js karena framework Next.js berbasis react untuk membangun web interface yang interaktif dan responsif serta untuk meningkatkan optimisasi pada web app

Business logic & API : Node.js karena merupakan bagian dari ekosistem JavaScript yang mendukung konkurensi untuk penggunaan beberapa user dalam satu family secara bersamaan

Database : PostgreSQL karena skalabilitas dan integritas data yang tinggi, mendukung JSON, dan dapat menangani relasi yang kompleks

External Service : Stripe API untuk proses pembayaran yang aman dan andal

Autentikasi : Supabase karena skabalitas dan kinerja tinggi, juga integrasi dengan layer lain yang lebih mudah seperti dengan PostgreSQL pada database dan Stripe API pada external service

1. **A. Development Environment**

**IDE: Visual Studio Code dengan extensions:**

* ESLint
* Prettier
* Docker
* GitLens
* PostgreSQL
* React Developer Tools

**Version control:**

* Git dengan GitHub untuk repository
* Branch strategy: main, develop, feature branches

**Local Development server:**

* Node.js v18+ untuk runtime
* npm untuk package management
* nodemon untuk auto-reload development

**Database**

* PostgreSQL 14+
* pgAdmin 4 untuk database management

**Langkah-langkah implementasi:**

**A diagram of a software development process

Description automatically generated**

**Timeline:**

diperkirakan untuk dibuat dalam 2 bulan (November 2024-Januari 2025)

* Setup & Initial Development: 1-2 minggu
* Feature Development: 4 minggu
* Testing & Deployment: 1-2 minggu

**Initial Setup**

npx create-next-app@latest Game-Dist-App

cd Game-Dist-App

npm install @supabase/supabase-js

Berikut hasil frontend

A screenshot of a computer

Description automatically generated

Setelah menekan tombol “Sign in with google”A screenshot of a computer

Description automatically generated

Bearer token didapat setelah login



Bearer token digunakan untuk autentikasi setiap layanan di web.

Dan hasil di database setelah login:

A screenshot of a computer

Description automatically generated

Terlihat bahwa user tertambahkan di database secara otomatis setelah melakukan OAuth.

**Containerization:**

Containerization dilakukan menggunakan Dockerfile untuk membangun image Docker yang berisi aplikasi Next.js Anda, dimulai dari menginstal dependensi, membangun aplikasi, hingga menghasilkan image produksi yang siap dijalankan. Sementara itu, docker-compose.yml digunakan untuk menjalankan container aplikasi berdasarkan image tersebut, dengan konfigurasi tambahan seperti port mapping, volume, dan environment variables (termasuk Supabase URL dan key). Kombinasi keduanya memungkinkan aplikasi Anda berjalan dalam container yang terisolasi, efisien, dan mudah dideploy ke lingkungan apa pun.

A screenshot of a computer program

Description automatically generated

**Penjelasan Tahapan dalam Dockerfile**

1. **Build Stage** (FROM node:18 AS builder):
   * **Base Image:** Menggunakan node:18 sebagai base image (versi Node.js 18).
   * **Set Working Directory:** WORKDIR /app menetapkan direktori kerja di dalam container ke /app.
   * **Install Dependencies:**
     + COPY package.json package-lock.json ./ menyalin file package.json dan package-lock.json ke direktori kerja.
     + RUN npm install --legacy-peer-deps menjalankan instalasi dependensi Node.js.
   * **Environment Variables:**
     + ARG NEXT\_PUBLIC\_SUPABASE\_URL dan ARG NEXT\_PUBLIC\_SUPABASE\_ANON\_KEY mendefinisikan argumen build yang akan dipakai untuk Supabase.
     + ENV menetapkan variabel lingkungan di container untuk digunakan aplikasi.
   * **Copy Project Files:**
     + COPY . . menyalin semua file proyek ke dalam direktori kerja di container.
   * **Build Next.js App:**
     + RUN npm run build membangun aplikasi Next.js untuk produksi, menghasilkan folder .next.
2. **Production Stage** (FROM node:18-alpine):
   * **Base Image:** Menggunakan node:18-alpine (versi ringan Node.js 18).
   * **Set Working Directory:** WORKDIR /app menetapkan direktori kerja di dalam container.
   * **Copy Artifacts:**
     + Menyalin hasil build (.next), file publik (public), dan dependensi (node\_modules) dari tahap build ke tahap produksi.
   * **Expose Port:**
     + EXPOSE 3000 memberitahukan bahwa container mendengarkan pada port 3000.
   * **Start Server:**
     + CMD ["npm", "start"] menjalankan perintah untuk memulai server Next.js dalam mode produksi.

**Main Business Logic:**

**Core Business Implementation:**

Family Sharing:

A screenshot of a computer

Description automatically generated

**Pembuatan Grup Keluarga**

* Pengguna dapat membuat satu grup keluarga.
* Pembuat grup secara otomatis menjadi pemilik grup.

**Manajemen Anggota**

* Pemilik dapat mengundang anggota melalui email.
* Anggota dapat dihapus oleh pemilik grup.
* Maksimal 5 anggota per grup (termasuk pemilik).

**Berbagi Game**

* Game yang dibeli oleh salah satu anggota dapat diakses oleh semua anggota grup.
* Pemilik asli game tetap memiliki hak kepemilikan.
* Fitur berbagi dapat diaktifkan atau dinonaktifkan untuk setiap game.

Untuk setiap api endpoint dan implementasi main business logic dapat diakses melalui Readme.md yang tertera untuk penjelasan yang lebih lengkap. Secara umum, berikut hal-hal yang diimplementasikan:

* User authentication with Google OAuth
* Game library management
* Family group creation and management
* Game sharing between family members

**Service Integration:**

**A screenshot of a computer

Description automatically generated**

Integrasi ini melibatkan penggunaan layanan chatbot dari pihak ketiga (dalam kasus ini service teman sekelas) ke dalam aplikasi **Game Distribution App**. Tujuan utama dari integrasi ini adalah memberikan fitur chatbot yang dapat membantu pengguna menjawab pertanyaan atau mengatasi kesulitan yang mereka hadapi saat menggunakan aplikasi.

**Langkah-Langkah Implementasi**

1. **Request Access**  
   Aplikasi meminta izin kepada penyedia layanan chatbot untuk memanfaatkan layanannya. Hal ini biasanya dilakukan dengan mengirimkan informasi seperti URL aplikasi atau API key yang digunakan untuk mengakses layanan tersebut.
2. **Script Injection**  
   Layanan chatbot diintegrasikan ke dalam aplikasi dengan cara menyisipkan script pihak ketiga. Berikut adalah contoh potongan kode yang digunakan:

window.MUSICMATE\_API\_KEY = "your\_api\_key\_here";

<script src="https://spotify-bot.azurewebsites.net/static/js/widget-loader.js"></script>

Dalam contoh ini, file eksternal widget-loader.js digunakan untuk menangani logika chatbot, sementara window.MUSICMATE\_API\_KEY berfungsi sebagai mekanisme otentikasi.

**Cara Kerja**

* Ketika aplikasi dijalankan, script chatbot akan di-load secara otomatis, menghubungkan aplikasi dengan server chatbot pihak ketiga.
* API key yang telah disetel memungkinkan aplikasi untuk menggunakan layanan chatbot dengan aman dan sesuai dengan batasan akses yang diberikan oleh penyedia layanan.