**Proposal Web Rekber (Rekening Bersama) dengan Fitur Chatting**

**1. Deskripsi Proyek**

Proyek ini adalah pengembangan sistem rekening bersama (rekber) yang memungkinkan transaksi aman antara buyer dan seller, dilengkapi dengan fitur realtime chat untuk komunikasi langsung. Sistem ini mencakup dashboard untuk user dan admin, serta perhitungan otomatis fee berdasarkan nominal transaksi.

**2. Teknologi yang Digunakan**

* **Backend:**

Django: Untuk manajemen user, transaksi, dan pengelolaan database.

Node.js + Socket.io: Untuk fitur realtime chat.

* **Database:**

PostgreSQL: Database utama untuk menyimpan data user, transaksi, room, dan fee.

* **Caching dan Messaging:**

Redis: Untuk caching dan sebagai message broker untuk komunikasi realtime pada chat.

* **Autentikasi:**

JWT (JSON Web Token): Untuk autentikasi user antara Django dan Node.js.

* **Server:**

Windows Server: Digunakan untuk pengembangan dan testing di localhost.

Linux: Untuk server produksi, memberikan stabilitas dan performa yang lebih baik.

**3. Struktur Utama Sistem**

**A. Register dan Login**

Register: User dapat membuat akun baru.

Login: User dapat masuk ke akun mereka. Sistem menghasilkan token autentikasi (JWT) yang digunakan untuk mengakses berbagai fitur, termasuk room chat.

**B. Dashboard User (Buyer/Seller)**

**Dashboard User:** Terintegrasi untuk buyer dan seller karena user bisa berperan sebagai buyer atau seller dalam transaksi yang berbeda.

**Form Transaksi:** User mengisi form untuk melakukan transaksi (nominal harga).

**Button "Buat Room":** Setelah mengisi form, user klik untuk memulai transaksi dan pembuatan room chat.

Popup Konfirmasi: Saat klik "Buat Room," muncul popup yang menampilkan:

Total Nominal Transaksi (Harga + Fee).

Fee yang dikenakan (dihitung otomatis berdasarkan nominal harga).

Dua pilihan: "Batal" atau "Lanjut".

Total Transaksi di Header Room Chat: Jika user lanjut, room chat dibuat, dan total nominal transaksi muncul di header room chat.

**C. Perhitungan Fee**

* Fee dihitung secara otomatis berdasarkan nominal transaksi:
* 10K 100K: Fee 5K
* 101K 300K: Fee 10K
* 301K 650K: Fee 15K
* 651K 950K: Fee 20K
* 951K 2Jt: Fee 25K
* 2Jt 3,5Jt: Fee 35K
* 3,5Jt 6,5Jt: 2,5%
* 6,5Jt 10Jt: 3,4%
* 10Jt 20Jt: 4,2%

**D. Room Chat**

Fitur realtime chat yang dihandle oleh Node.js dan Socket.io.

Saat user setuju dengan nominal dan fee, room chat otomatis dibuat, dan kedua belah pihak bisa berkomunikasi langsung untuk menyelesaikan transaksi.

**E. Dashboard Admin**

*Total Penghasilan dari Fee:* Admin bisa melihat total penghasilan yang dihasilkan dari fee pada hari tertentu.

*Jumlah Transaksi Hari Ini:* Admin bisa melihat jumlah transaksi yang berhasil dilakukan setiap harinya.

**Fitur Tambahan:**

1. Grafik jumlah transaksi atau fee per hari/bulan menggunakan Chart.js.
2. Daftar transaksi bermasalah atau pending yang perlu diatasi.
3. Data pengguna aktif hari ini.

**4. Alur Logika dan Flow**

**A. Flow User untuk Transaksi**

1. User login →

2. Mengisi form harga di dashboard →

3. Klik "Buat Room" →

4. Muncul popup total nominal + fee →

5. Pilih "Batal" atau "Lanjut" →

6. Jika lanjut, room chat dibuat, dan total nominal muncul di header room chat.

**B. Alur Penghitungan Fee**

1. Nominal transaksi dimasukkan oleh user.

2. Sistem menghitung fee berdasarkan nominal.

3. Total nominal transaksi (harga + fee) ditampilkan di popup konfirmasi.

**C. Alur Dashboard Admin**

1. Admin login.

2. Melihat total penghasilan fee hari ini.

3. Melihat jumlah transaksi hari ini.

4. Mengelola transaksi bermasalah dan pengguna aktif.

**5. Tools dan Fitur Tambahan**

1. nginx: Untuk reverse proxy antara Django dan Node.js.
2. Docker: Untuk containerization jika dibutuhkan deployment yang scalable.
3. Celery: Untuk job background, seperti penghitungan fee atau laporan transaksi.