

📌 **Tujuan:** Mengembangkan sistem backend aplikasi **Bantu** yang dapat menangani **pengguna, pencarian pekerjaan, transaksi, dan komunikasi** dalam platform freelance ini.

📌 **Tantangan:** Setiap tim bertugas untuk membuat **arsitektur backend yang scalable dan efisien**, mencakup database, API, serta autentikasi pengguna.

📌 **Deadline:** 7 Hari

🔧 Tugas Tim Backend Developer:

Setiap tim harus membangun **backend MVP (Minimum Viable Product)** untuk aplikasi "**Bantu**", mencakup:

1. Arsitektur Backend & Database

- Desain **database schema** menggunakan PostgreSQL/MySQL/MongoDB.
- Struktur tabel utama: **User, Perusahaan, Freelancer, Pekerjaan, Transaksi, Chat, Review**.
- Relasi antara entitas (misal, **satu user bisa menjadi freelancer & klien**).

2. Autentikasi & Manajemen Pengguna

- **JWT/OAuth** untuk login & registrasi pengguna (Freelancer & Perusahaan).
- Middleware untuk **otorisasi role-based** (admin, freelancer, perusahaan).

3. API Development

- RESTful API atau GraphQL untuk mengelola fitur utama:
 - ✅ **CRUD Pekerjaan** (post, update, delete, apply job).
 - ✅ **Pencarian & Filter Pekerjaan** berdasarkan kategori, lokasi, harga.
 - ✅ **Sistem Proposal & Negosiasi** antara freelancer & perusahaan.
 - ✅ **Sistem Pembayaran** menggunakan **Midtrans/Xendit** (opsional).

4. Real-time Chat & Notifikasi

- **Socket.io** atau **Firebase** untuk fitur komunikasi antar pengguna.
- **Push Notification API** untuk status pekerjaan dan update transaksi.

5. Keamanan & Optimasi Backend

- **Proteksi API** (Rate Limiting, CORS, Input Validation).

- **Caching dengan Redis** untuk mempercepat respons pencarian.
 - **Deployment & CI/CD pipeline** (opsional).
-

Output yang Diharapkan:

- ✓ **Database Schema & API Documentation (Swagger/Postman)**
 - ✓ **Backend Service (Node.js/Express, Django, Laravel, atau lainnya)**
 - ✓ **Authentication & User Role Management**
 - ✓ **Deployment di Cloud (Heroku, AWS, GCP, atau lainnya) (opsional)**
 - ✓ **Presentasi & Demo API (10-15 menit)**
-

Kriteria Penilaian:

- ◆ **Struktur Backend yang Scalable** – Apakah desainnya efisien untuk aplikasi besar?
- ◆ **Keamanan & Best Practices** – Apakah sistem autentikasi & API sudah aman?
- ◆ **Kinerja API & Responsiveness** – Seberapa cepat dan efisien data diambil?
- ◆ **Implementasi Fitur Inti** – Apakah CRUD pekerjaan, transaksi, dan chat sudah berjalan?
- ◆ **Presentasi & Kemampuan Problem-Solving** – Seberapa baik tim menjelaskan solusi backend mereka?