

Junior Backend Assessment Test

- Buatlah sebuah service account yang berupa HTTP Server menggunakan REST API yang memiliki API sebagai berikut:
 1. `/daftar`: API untuk registrasi nasabah baru dengan payload JSON berisi field `nama`, `nik`, dan `no_hp`. API akan memberikan balikan dengan status 200 dan payload JSON berisi field `no_rekening` yang berisi data nomor rekening nasabah. Jika `nik` atau `no_hp` sudah digunakan, API akan memberikan balikan status 400 dan payload JSON berisi field `remark` yang berisi deskripsi kesalahan terkait data yg dikirim.
 2. `/tabung`: API untuk nasabah menabung dengan payload JSON berisi field `no_rekening`, dan `nominal`. API akan memberikan balikan dengan status 200 dan payload JSON berisi field `saldo` yang berisi data saldo nasabah saat ini. Jika `no_rekening` tidak dikenali, API akan memberikan balikan status 400 dan payload JSON berisi field `remark` yang berisi deskripsi kesalahan terkait data yg dikirim.
 3. `/tarik`: API untuk nasabah menarik dana nasabah dengan payload JSON berisi field `no_rekening` dan `nominal`. API akan memberikan balikan dengan status 200 dan payload JSON berisi field `saldo` yang berisi data saldo nasabah saat ini. Jika `no_rekening` tidak dikenali atau saldo tidak cukup, API akan memberikan balikan status 400 dan payload JSON berisi field `remark` yang berisi deskripsi kesalahan terkait data yg dikirim.
 4. `/saldo/{no_rekening}`: API untuk melihat data saldo nasabah dengan path parameter bernama `no_rekening` berisi nomor rekening nasabah. API akan memberikan balikan dengan status 200 dan payload JSON berisi field `saldo` yang berisi data saldo nasabah saat ini. Jika `no_rekening` tidak dikenali, API akan memberikan balikan status 400 dan payload JSON berisi field `remark` yang berisi deskripsi kesalahan terkait data yg dikirim.

Catatan:

- **WAJIB** membuat log yang jelas dan terstruktur. Sertai data dan context yang berkait dengan proses yang sedang terjadi pada log. Gunakan level log (INFO, WARNING, ERROR, CRITICAL) yang sesuai dengan keadaan.
- HTTP Server dibuat menggunakan Golang (`echo` atau `fiber`) atau Python (`fastapi`).
- Data disimpan dalam database PostgreSQL menggunakan docker container.
- Gunakan environment variable dan argument parser untuk konfigurasi semua service. Environment variable digunakan untuk konfigurasi sensitif seperti user dan password database. Argument parser digunakan untuk konfigurasi non-sensitif seperti REST API host dan port.
- Gunakan Dockerfile untuk membuat image untuk service account.
- Gunakan Docker Compose untuk deployment service account dan database.

Kriteria Penilaian:

- Log structure: Log yang terformat rapih dan memiliki informasi yang jelas
- Software Architecture: Variable naming yang jelas, separasi modul yang rapih
- Database schema: Penggunaan relasi antar table yang sesuai
- Database operation: Operasi database yang efektif
- REST API structure: Struktur endpoint, parameter, dan middleware yang sesuai
- Configuration management: Konfigurasi service yang fleksibel dan aman
- Deployment setup: Penggunaan Docker dalam deployment aplikasi