DOKUMENTASI UJI KOMPETENSI KEAHLIAN
APLIKASI SIPALO (SISTEM PEMINJAMAN ALAT OLAHRAGA)

Disusun oleh:

FINELYA PUTRI ARDIANATASYA / 19

XII RPL 2

PROGRAM KEAHLIAN PENGEMBANGAN PERANGKAT LUNAK DAN GIM

KOMPETENSI KEAHLIAN REKAYASA PERANGKAT LUNAK

SMK BRANTAS KARANGKATES

2026

a. Analisis Kebutuhan

1. Admin: Admin adalah pengguna dengan hak akses tertinggi dalam sistem. Admin bertanggung jawab atas pengelolaan data master dan pengaturan sistem.
   Tanggung jawab Admin:
   - Login dan Logout ke sistem
   - Mengelola data user
   - Mengelola data alat
   - Mengelola kategori alat
   - Mengelola data peminjaman
   - Mengelola data pengembalian
   - Melihat log aktivitas
   - Mengelola laporan

2. Petugas: Petugas adalah pengguna yang menangani operasional peminjaman dan pengembalian alat di lapangan.
   Tanggung jawab Petugas:
   - Login dan Logout ke sistem
   - Menyetujui pengajuan peminjaman
   - Menolak pengajuan peminjaman
   - Memproses pengembalian alat
   - Memeriksa kondisi alat
   - Menghitung denda keterlambatan
   - Mencetak laporan

3. Peminjam: Peminjam adalah pengguna seperti siswa, guru, atau staf yang meminjam alat.
   Tanggung jawab Peminjam:
   - Login dan Logout ke sistem
   - Melihat daftar alat
   - Mengajukan peminjaman
   - Melihat status peminjaman
   - Melakukan pengembalian alat

4. Kebutuhan fungsional:
   1. Sistem melakukan autentikasi pengguna (login/logout) dan pengelolaan hak akses berdasarkan role dengan data yang tersimpan di database.
   2. Mengelola data master (user, alat, kategori) melalui operasi tambah, ubah, hapus, dan tampil yang tersimpan di database.
   3. Menampilkan daftar alat, detail alat, stok, dan kondisi alat kepada pengguna.
   4. Sistem memfasilitasi proses pengajuan peminjaman alat oleh peminjam melalui form peminjaman.
   5. Sistem memproses persetujuan atau penolakan pengajuan peminjaman oleh petugas.
   6. Memproses pengembalian alat termasuk pencatatan tanggal kembali dan kondisi alat.
   7. Sistem menghitung denda keterlambatan pengembalian sesuai aturan yang ditetapkan.

8. Sistem menyediakan pembuatan dan pencetakan laporan serta pencatatan log aktivitas.

5. Kebutuhan Non-Fungsional
   1. Tampilan mudah digunakan
   2. Navigasi jelas
   3. Form mudah dipahami
   4. Validasi input berjalan baik
   5. Hak akses berbasis role
   6. Stok alat selalu terupdate

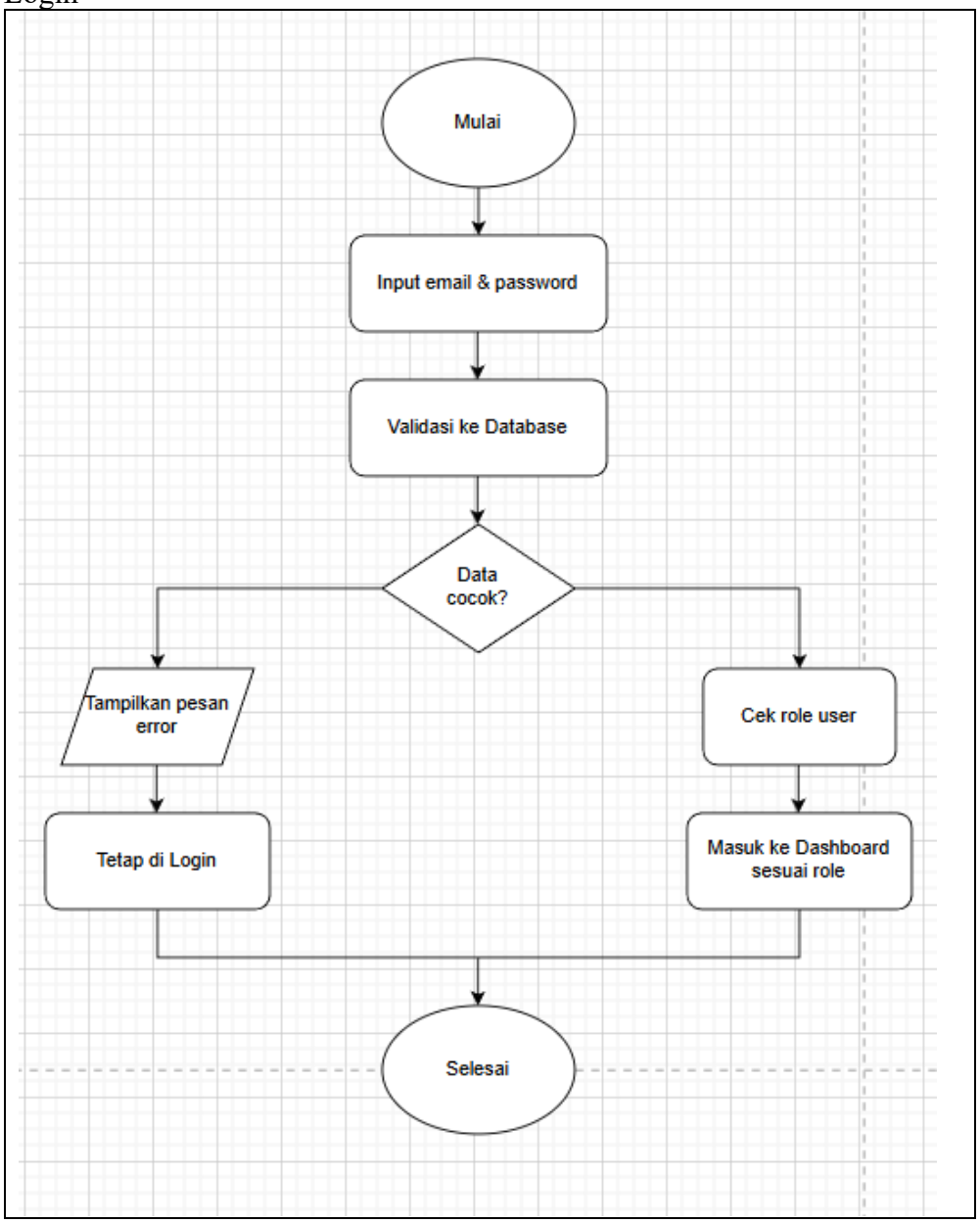
b. ERD

```
CREATE TABLE users (
  user_id UUID PRIMARY KEY
    REFERENCES auth.users(id) ON DELETE CASCADE,
  nama VARCHAR(100) NOT NULL,
  username VARCHAR(50) UNIQUE NOT NULL,
  role VARCHAR(20) NOT NULL
    CHECK (role IN ('admin', 'petugas', 'peminjam'))
);

CREATE TABLE kategori (
  kategori_id SERIAL PRIMARY KEY,
  nama_kategori VARCHAR(50) NOT NULL
);

INSERT INTO kategori (nama_kategori) VALUES
('Sepak Bola'),
('Basket'),
('Badminton'),
('Voli');


CREATE TABLE alat (
  alat_id SERIAL PRIMARY KEY,
  nama_alat VARCHAR(100) NOT NULL,
  kategori_id INT NOT NULL
    REFERENCES kategori(kategori_id),
```

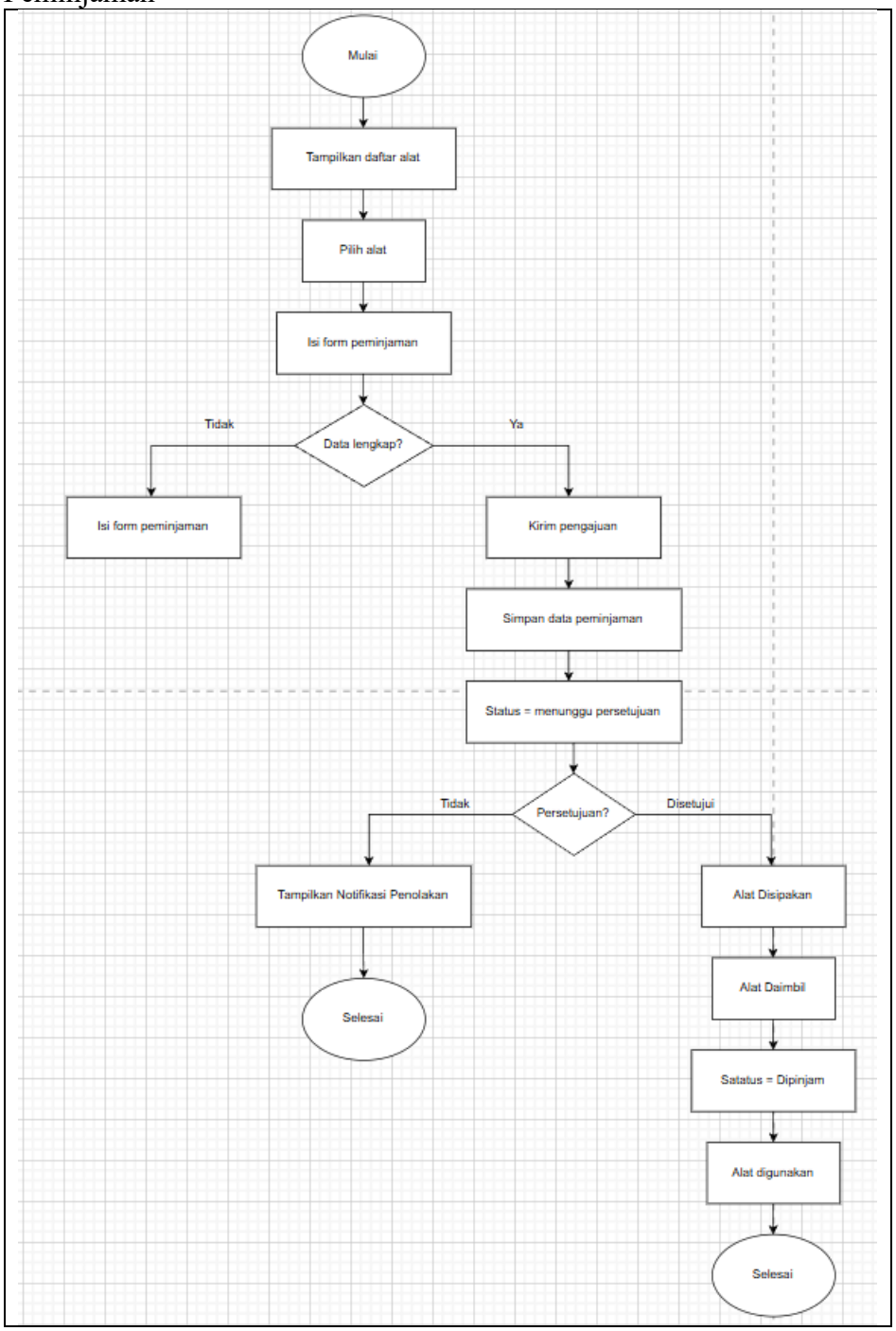
```
  stok INT NOT NULL,
  kondisi VARCHAR(50)
);

INSERT INTO alat (nama_alat, kategori_id, stok, kondisi) VALUES
('Bola Sepak', 1, 5, 'Baik'),
('Bola Basket', 2, 4, 'Baik'),
('Raket Badminton', 3, 10, 'Baik'),
('Bola Voli', 4, 2, 'Baik');


CREATE TABLE peminjaman (
  peminjaman_id SERIAL PRIMARY KEY,
  user_id UUID NOT NULL
    REFERENCES users(user_id),
  alat_id INT NOT NULL
    REFERENCES alat(alat_id),
  tgl_pinjam TIMESTAMP DEFAULT NOW(),
  lama_pinjam INT NOT NULL,
  status VARCHAR(20)
);


CREATE TABLE pengembalian (
  pengembalian_id SERIAL PRIMARY KEY,
  peminjaman_id INT NOT NULL
    REFERENCES peminjaman(peminjaman_id),
  tgl_kembali TIMESTAMP,
  denda NUMERIC,
  kondisi_alat VARCHAR(50)
);


CREATE TABLE log_aktivitas (
  log_id SERIAL PRIMARY KEY,
  user_id UUID NOT NULL
    REFERENCES users(user_id),
  aktivitas VARCHAR(100),
  tgl_aktivitas TIMESTAMP DEFAULT NOW()
);
```

c. Flowchart

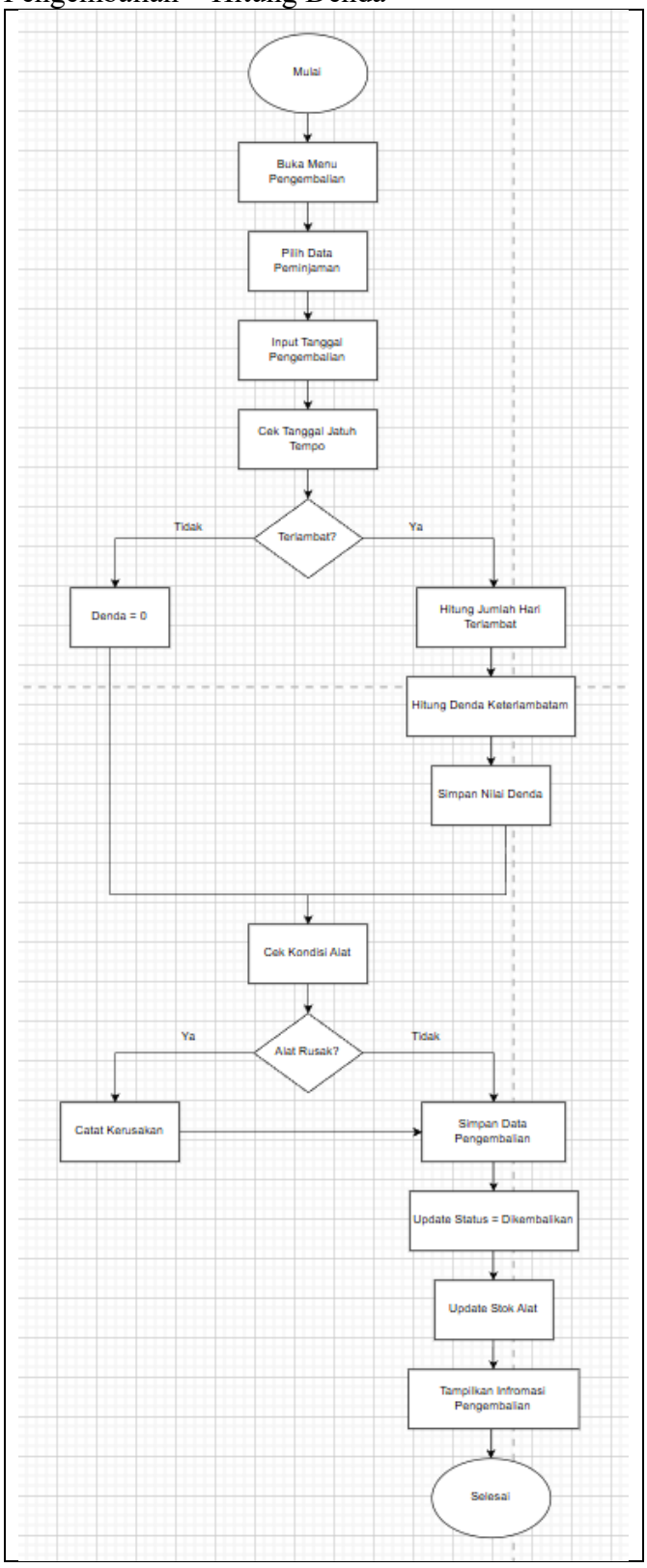
1. Login



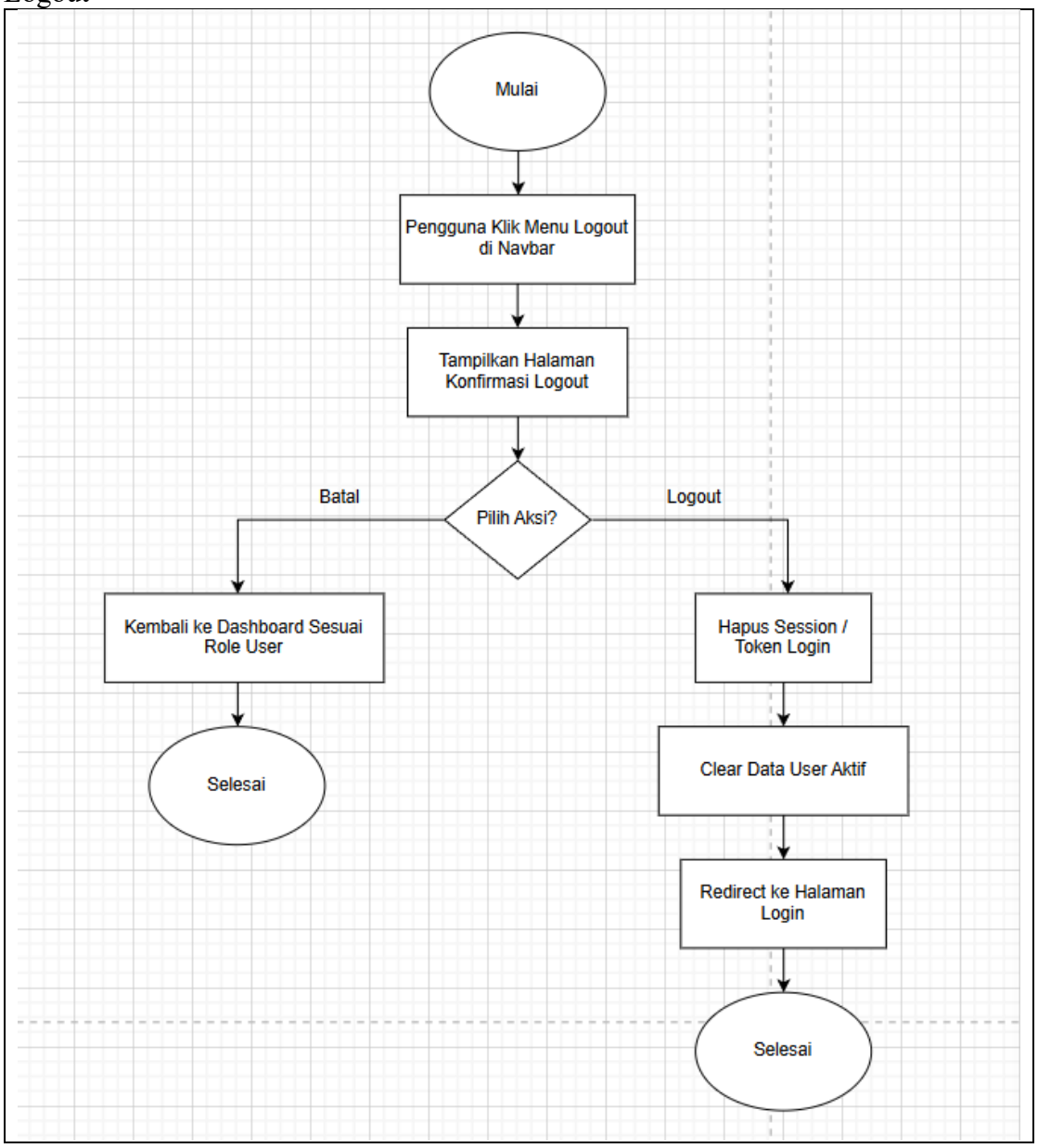
2. Peminjaman



3. Pengembalian + Hitung Denda



4. Logout



d. Dokumentasi fungsi / prosedur

1. Function & Trigger Peminjaman

```
Function:
        CREATE OR REPLACE FUNCTION proses_peminjaman_alat()
        RETURNS TRIGGER AS $$
        DECLARE
          stok_sekarang INT;
        BEGIN
          -- Ambil stok terakhir
            SELECT stok INTO stok_sekarang FROM alat WHERE alat_id =
        NEW.alat_id;

          -- Cek ketersediaan
          IF stok_sekarang <= 0 THEN
            RAISE EXCEPTION 'Stok alat sudah habis!';
```

```
    END IF;

    -- Kurangi stok
    UPDATE alat SET stok = stok - 1 WHERE alat_id = NEW.alat_id;

    RETURN NEW;
END;
$$ LANGUAGE plpgsql;
```

Trigger:

```
CREATE TRIGGER trg_peminjaman_otomatis
BEFORE INSERT ON peminjaman
FOR EACH ROW
EXECUTE FUNCTION proses_peminjaman_alat();
```

2. Function & Trigger Pengembalian

Function:

```
CREATE OR REPLACE FUNCTION proses_pengembalian_alat()
RETURNS TRIGGER AS $$
BEGIN
    -- Tambah stok kembali
    UPDATE alat
    SET stok = stok + 1
      WHERE alat_id = (SELECT alat_id FROM peminjaman WHERE peminjaman_id = NEW.peminjaman_id);

    -- Update status jadi kembali
    UPDATE peminjaman
    SET status = 'kembali'
    WHERE peminjaman_id = NEW.peminjaman_id;

    RETURN NEW;
END;
$$ LANGUAGE plpgsql;
```

Trigger:

```
CREATE TRIGGER trg_pengembalian_otomatis
AFTER INSERT ON pengembalian
FOR EACH ROW
EXECUTE FUNCTION proses_pengembalian_alat();
```

3. Function & Trigger Log_aktivitas

Function:

```
CREATE OR REPLACE FUNCTION catat_ke_log()
RETURNS TRIGGER AS $$
BEGIN
    INSERT INTO log_aktivitas (user_id, aktivitas, tgl_aktivitas)
    VALUES (NEW.user_id, 'Melakukan transaksi baru di sistem', NOW());
```

```
    RETURN NEW;
END;
$$ LANGUAGE plpgsql;
```

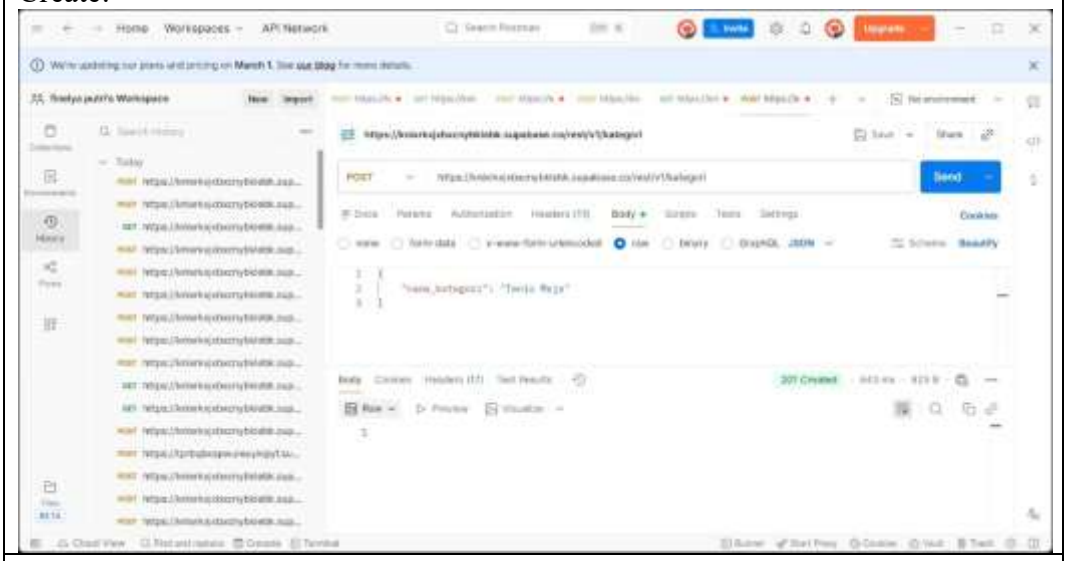
Trigger:

```
CREATE TRIGGER trg_buat_log
AFTER INSERT ON peminjaman
FOR EACH ROW
EXECUTE FUNCTION catat_ke_log();
```

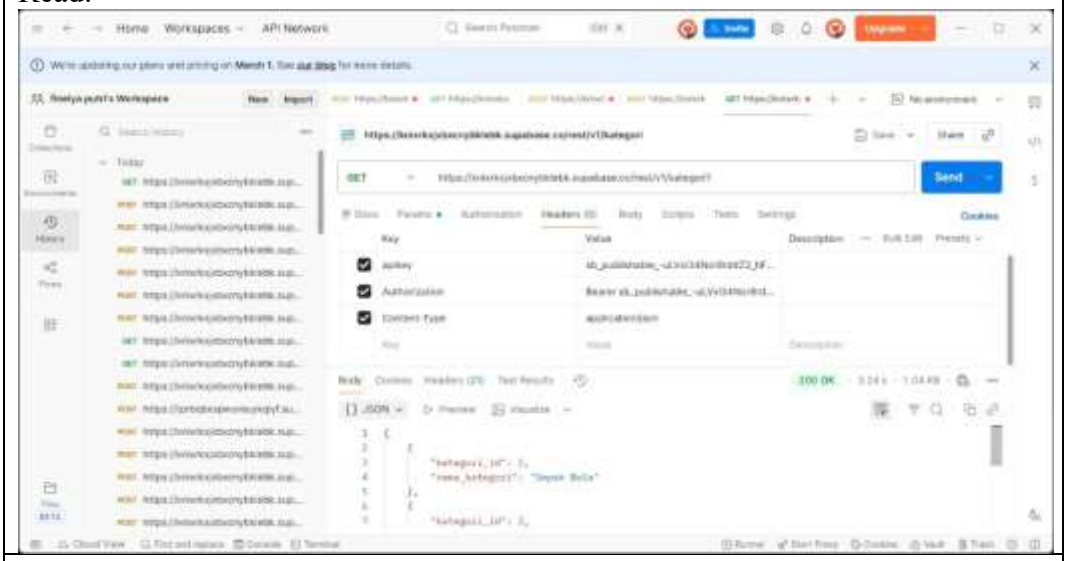
e. Test Postman

1. Kategori

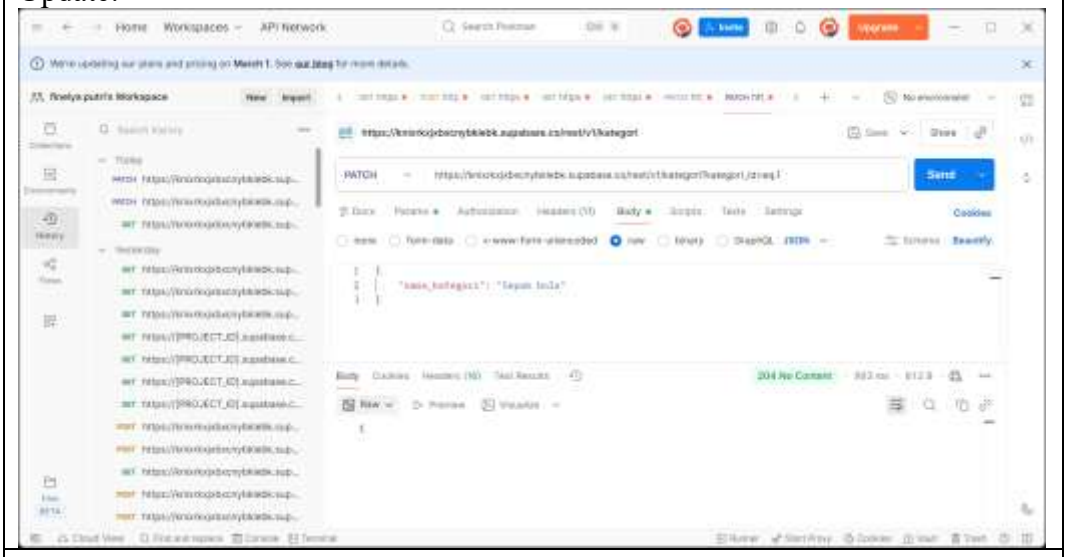
Create:



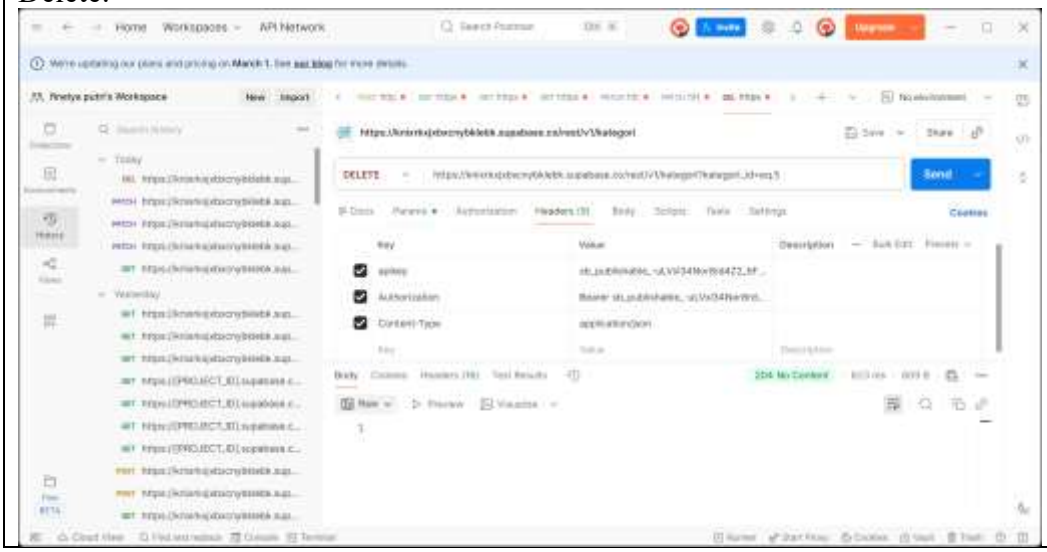
Read:



Update:

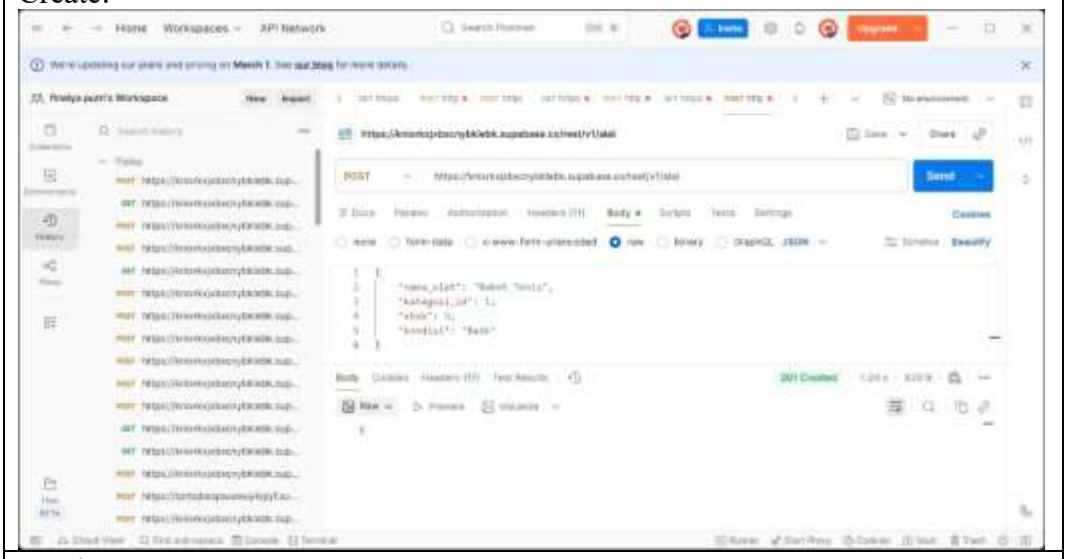


Delete:

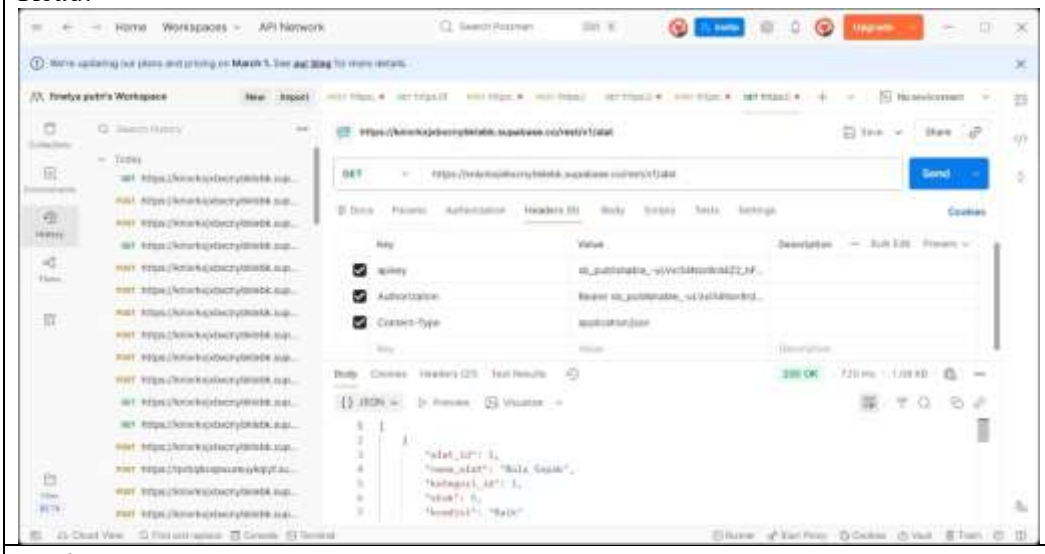


2. Alat

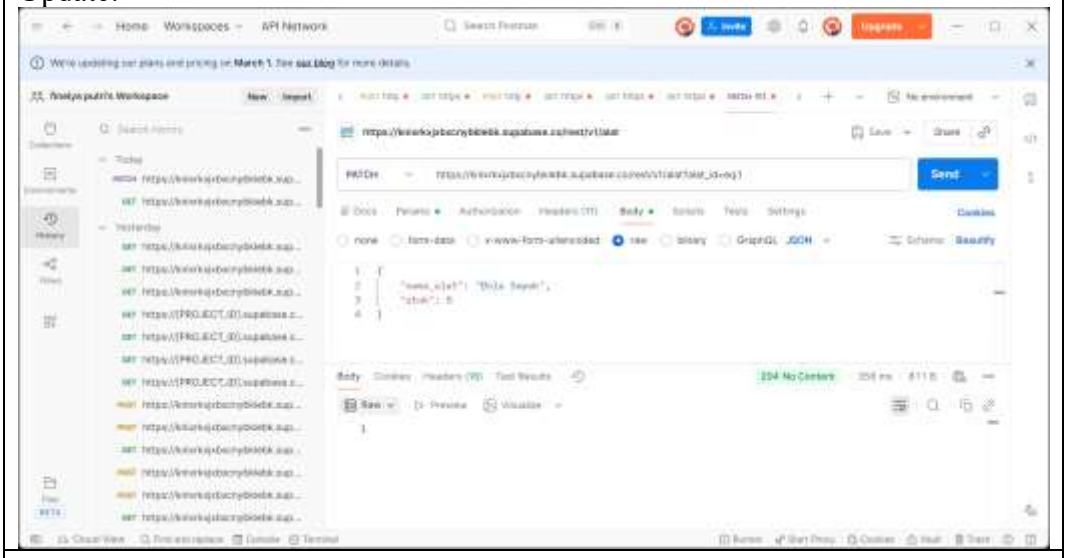
Create:



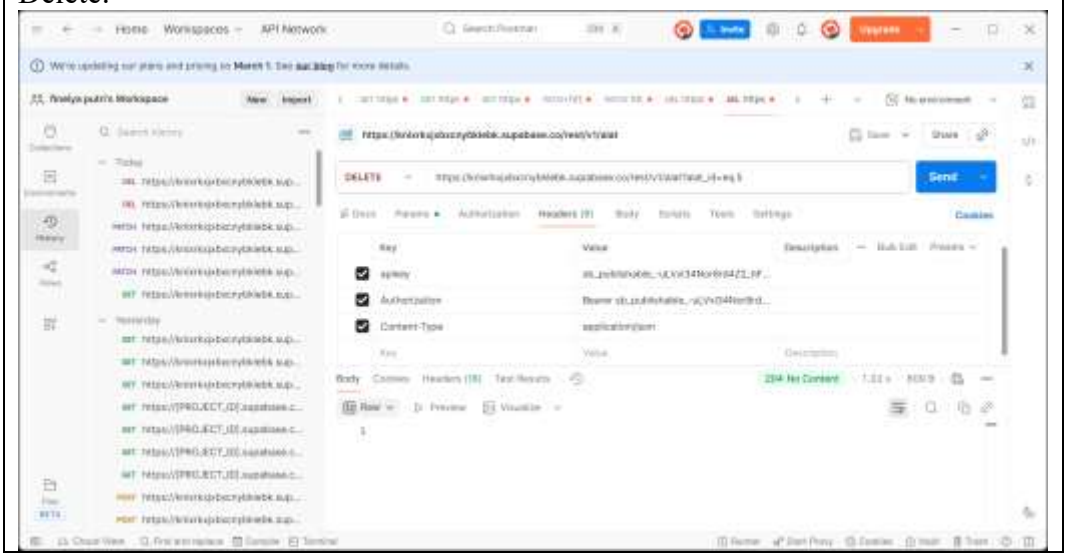
Read:



Update:

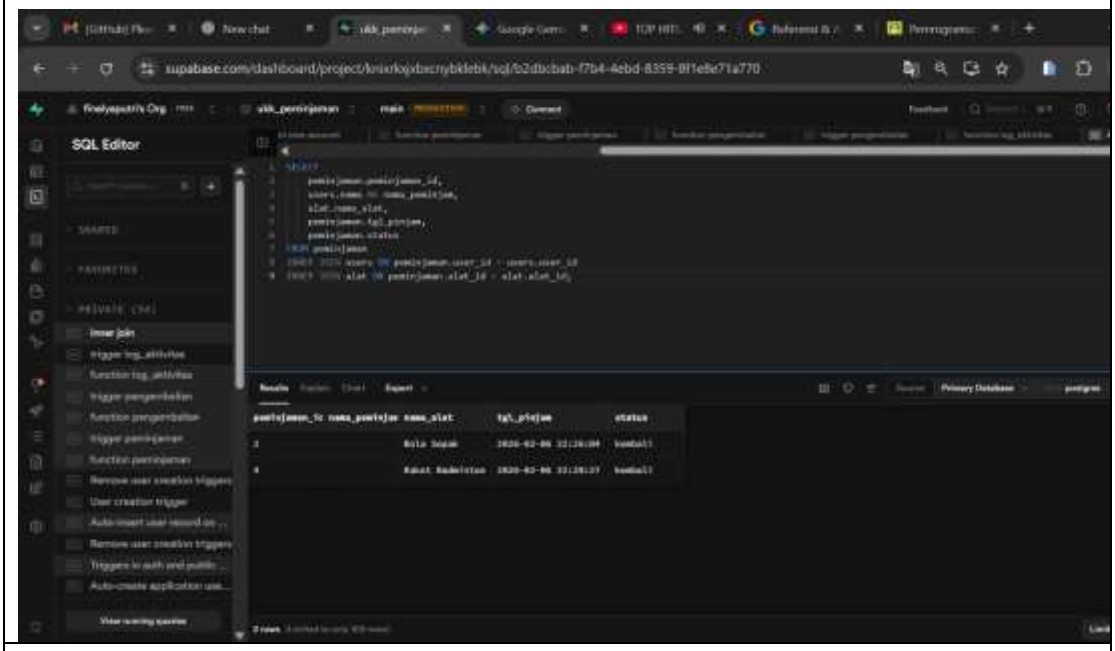


Delete:

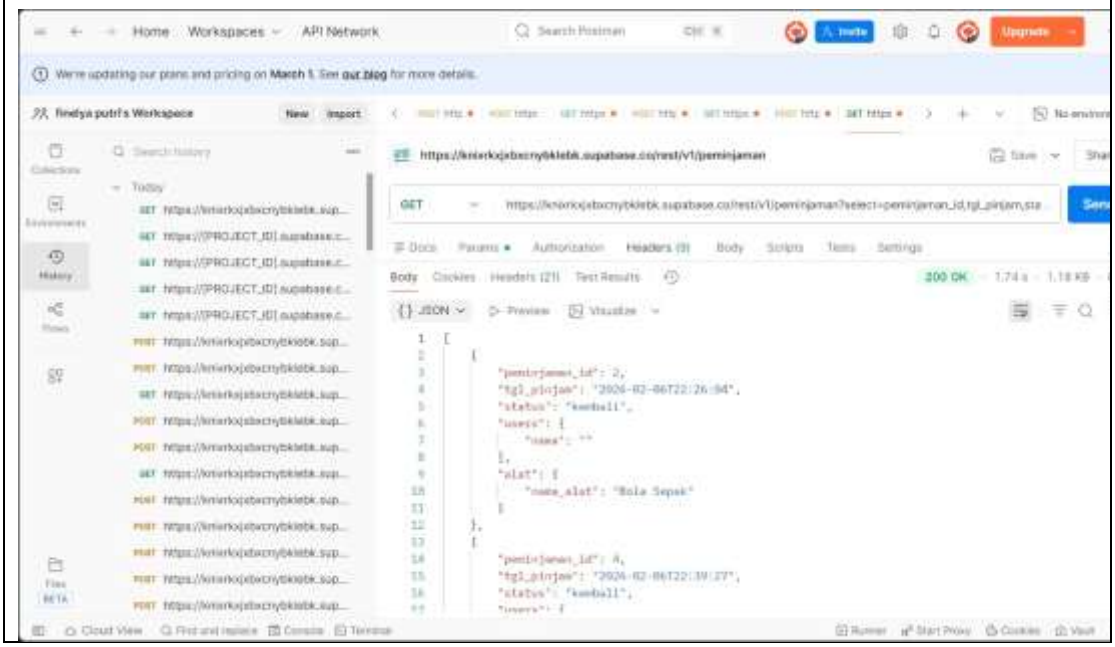


f. Inner Join

Inner Join supabase:

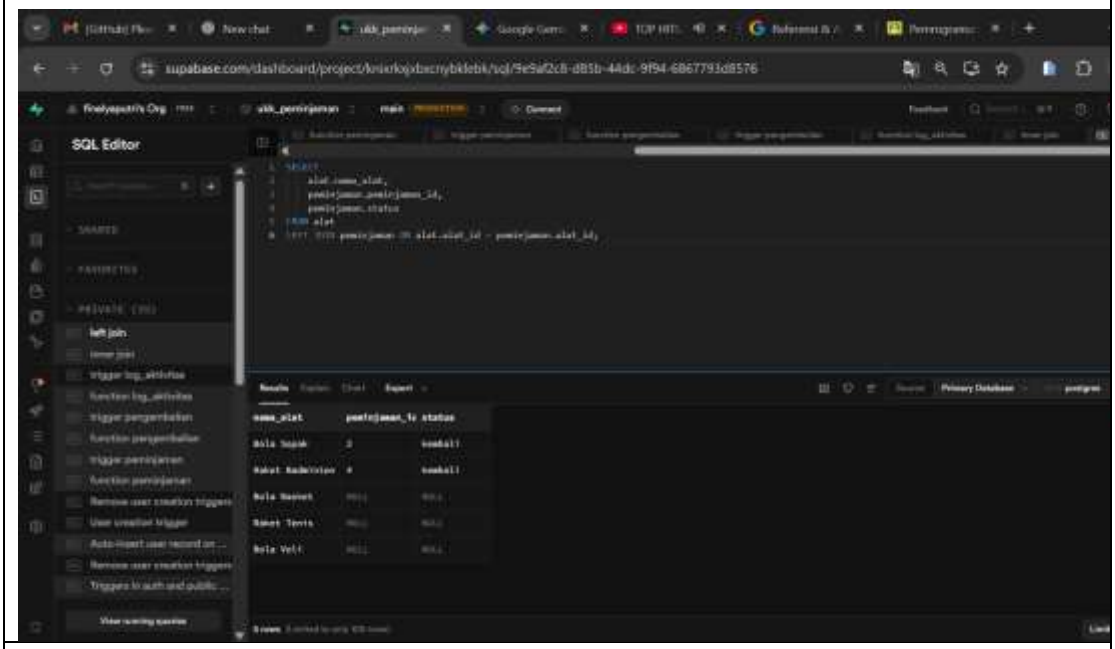


Inner Join Postman:

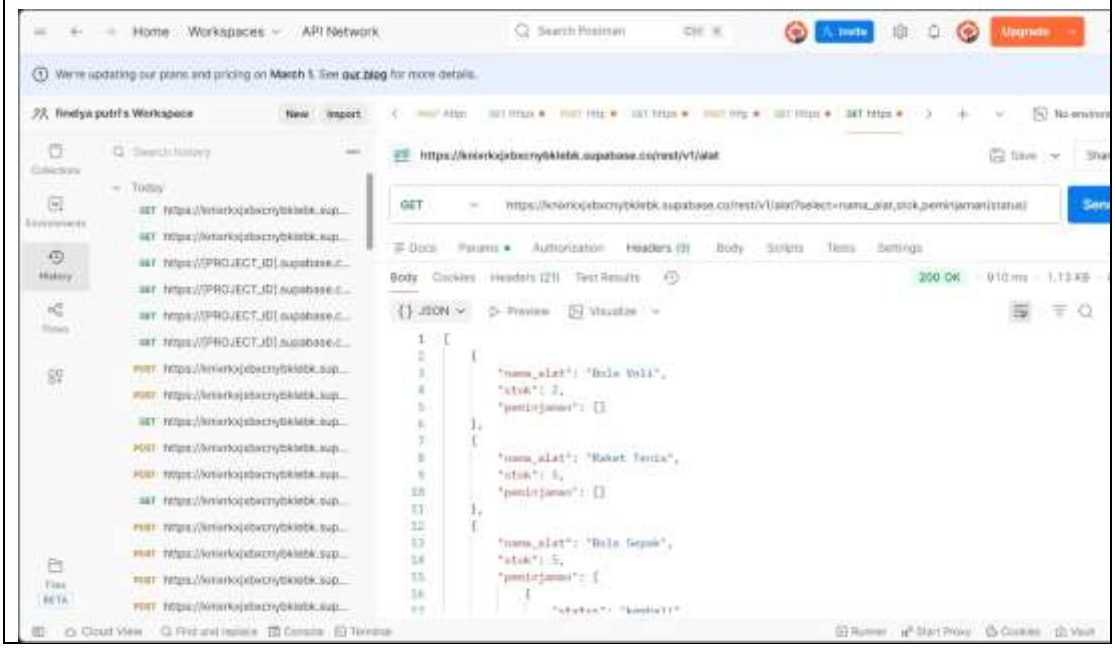


g. Left Join

Left Join Supabase:

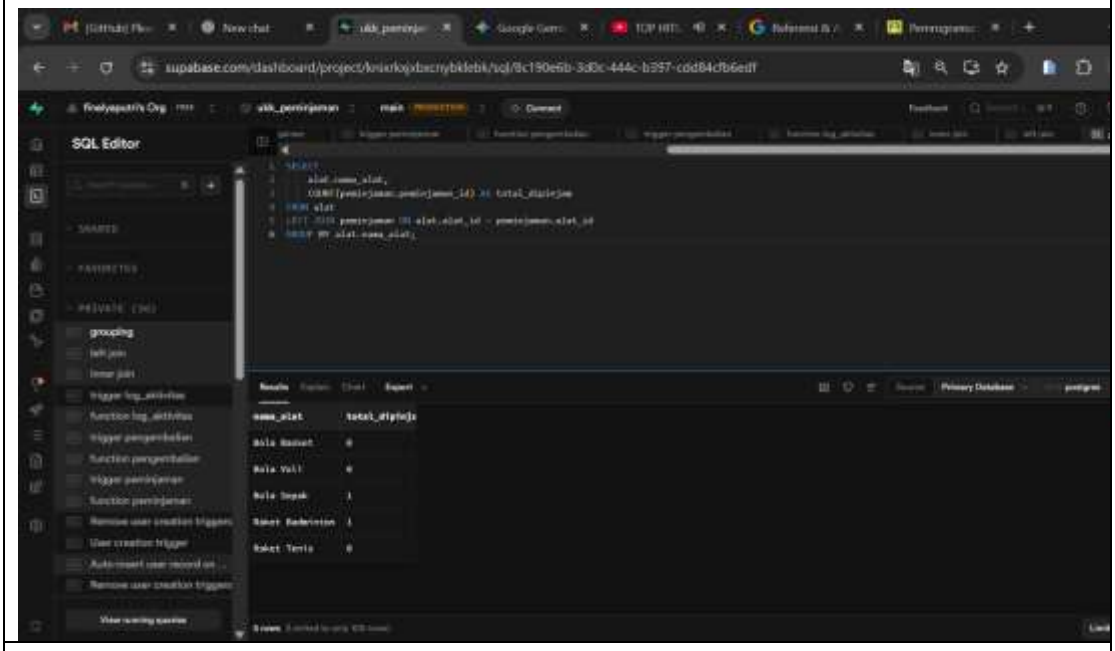


Left Join Postman:



h. Grouping

Grouping supabase:



Grouping Postman: