



# Dasar-Dasar SQL dan Penggunaan DBeaver

## Panduan Lengkap

Menguasai SQL dan DBeaver untuk Analisis Data Perizinan Berusaha

# Pentingnya Menguasai SQL untuk SPB

Bagi Direktorat **Sistem Perizinan Berusaha (SPB)**, kemampuan mengakses dan menganalisis data secara mandiri adalah **keterampilan krusial**. Ini bertujuan untuk **triangulasi informasi** yang diberikan Pengembang, yaitu **memverifikasi data secara independen** untuk memastikan akurasi dan menghindari kesalahpahaman.

Dengan menguasai SQL, tim dapat melakukan verifikasi data sendiri, memastikan informasi yang diterima valid dan dapat dipertanggungjawabkan, serta tidak bergantung sepenuhnya pada penjelasan dari tim pengembang.



## Verifikasi Mandiri

Memverifikasi klaim Pengembang dengan data aktual.



## Audit Independen

Melakukan audit data sistem secara berkala.



## Deteksi Inkonsistensi

Mengidentifikasi ketidaksesuaian data dengan cepat.



## Keputusan Berbasis Data

Membuat keputusan berdasarkan data yang terverifikasi.

# Konsep Dasar Database

**Database** adalah kumpulan data terorganisir yang tersimpan secara elektronik, seperti lemari arsip digital yang informasinya mudah diakses.

Kita sering berinteraksi dengan database tanpa menyadarinya: buku telepon, katalog produk, dan daftar karyawan adalah contoh database sehari-hari yang memudahkan pencarian data.

## Database (DB)

Kumpulan data terorganisir dalam satu sistem. Contoh: Database OSS-RBA.

## Skema (Schema)

Pengelompokan tabel-tabel seperti folder. Contoh: oss\_rba\_master.

## Tabel (Table)

Data tersusun dalam baris dan kolom. Contoh: m\_kbli (mirip Excel).

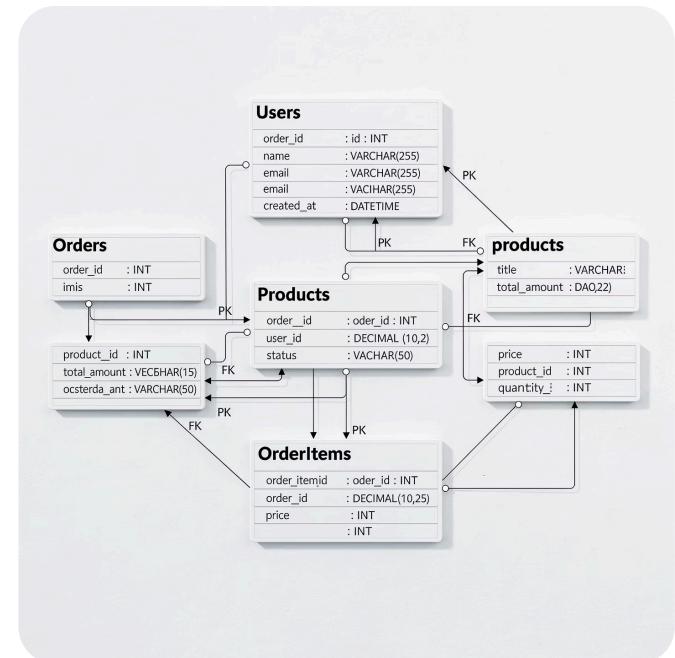
## Komponen Tabel Database

- Kolom (Field):** Jenis data dalam tabel. Contoh: kode, judul.
- Baris (Record):** Satu data lengkap. Contoh: Satu KBLI.

## Relasi Antar Tabel

Tabel-tabel terhubung membentuk sistem informasi:

- One-to-Many:** Satu KBLI bisa memiliki banyak izin.
- Many-to-Many:** Banyak KBLI bisa punya banyak izin.
- One-to-One:** Satu data hanya punya satu pasangan.



❑ **Contoh Relasi di OSS-RBA:** Tabel m\_kbli dan m\_kbli\_izin. Satu kode KBLI dapat memiliki banyak izin berbeda.

# Pengenalan SQL: Bahasa untuk Berkommunikasi dengan Database

**SQL** (Structured Query Language) adalah bahasa standar untuk berinteraksi dengan database. Dengan SQL, Anda dapat mengambil (SELECT), menambah (INSERT), mengubah (UPDATE), menghapus (DELETE), dan membuat/mengubah struktur database (CREATE, ALTER).

SQL adalah **bahasa universal** yang dipakai di banyak sistem database seperti PostgreSQL, MySQL, dan Oracle. Menguasai SQL memudahkan Anda bekerja di berbagai platform.



1

## DDL - Data Definition Language

Perintah untuk membuat atau mengubah struktur database:

- **CREATE** - membuat tabel, database, dll
- **ALTER** - mengubah struktur tabel
- **DROP** - menghapus tabel, database, dll

## Syntax Dasar SQL

SQL menggunakan **statement** yang diakhiri dengan tanda ; (semicolon). Format dasar query SQL adalah:

```
SELECT kolom1, kolom2  
FROM nama_tabel  
WHERE kondisi  
ORDER BY kolom1;
```

### • Aturan Penting SQL

SQL **case-insensitive** untuk keyword (SELECT = select = Select), namun nama tabel/kolom bisa case-sensitive tergantung database

### • Penggunaan Semicolon

Gunakan tanda ; (semicolon) untuk mengakhiri setiap statement SQL

### • Komentar dalam SQL

Bisa menggunakan komentar dengan -- untuk single line atau /\* \*/ untuk multi-line

# Instalasi dan Setup DBeaver

DBeaver adalah aplikasi gratis, open source untuk mengakses dan mengelola database. Berfungsi seperti "browser" database, memudahkan pengelolaan data tanpa command line rumit, dan mendukung berbagai jenis database (PostgreSQL, MySQL, Oracle, dll).



## Gratis & Open Source

DBeaver Community Edition gratis, didukung komunitas aktif.



## Multi-Database Support

Mendukung lebih dari 80 jenis database populer (PostgreSQL, MySQL, Oracle, dll).



## Interface User-Friendly

Antarmuka grafis intuitif untuk navigasi dan visualisasi data.



## Visual Query Builder

Bangun query SQL dengan drag-and-drop, tanpa perlu coding manual.

## Langkah-Langkah Instalasi DBeaver

01

### Download Installer

Kunjungi [dbeaver.io/download/](https://dbeaver.io/download/), unduh Community Edition sesuai OS Anda.

02

### Instalasi Aplikasi

Ikuti wizard instalasi setelah mengunduh file.

03

### Jalankan DBeaver

Buka DBeaver dari Start Menu/Applications.

04

### First Time Setup

Buat koneksi baru atau jelajahi antarmuka saat pertama dibuka.

# Koneksi ke Database dengan DBeaver

Langkah pertama adalah membuat koneksi database di DBeaver untuk mengakses struktur, menjalankan query, dan mengelola data Anda.



## Buka New Connection

Klik ikon "New Database Connection" atau pilih **Database → New Database Connection**.



## Pilih Tipe Database

Pilih **PostgreSQL** (digunakan OSS-RBA) dari daftar. Klik **Next**.



## Isi Detail Koneksi

Lengkapi **Host**, **Port**, **Database**, **Username**, **Password**. Centang "Save password".



## Test & Download Driver

Klik "Test Connection". Jika diminta, download driver PostgreSQL. Tes ulang hingga muncul "**Connected**".



## Simpan Koneksi

Klik **Finish**. Koneksi muncul di panel kiri (Database Navigator).



## Tips Mengelola Koneksi

**Mengedit:** Klik kanan koneksi, pilih "Edit Connection".

**Menghapus:** Klik kanan koneksi, pilih "Delete".

**Keamanan Password:** DBeaver mengenkripsi password. Untuk keamanan ekstra, jangan centang "Save password".

# Menggunakan DBeaver untuk Query SQL

SQL Editor adalah inti DBeaver untuk menulis dan menjalankan query. Interface intuitif ini mendukung syntax highlighting, auto-complete, dan formatting otomatis.

## Cara 1: SQL Editor untuk Database

Klik kanan koneksi di Database Navigator → pilih **SQL Editor** → **New SQL Script**. SQL Editor baru akan terbuka.

## Cara 2: SQL Editor untuk Tabel Spesifik

Klik kanan tabel tertentu → pilih **View Data** atau **Edit Data**. SQL Editor akan terbuka di bawah, terhubung ke tabel tersebut.

## Cara 3: Menggunakan Shortcut

Gunakan shortcut **Ctrl+Shift+Enter** (Win/Linux) atau **Cmd+Shift+Enter** (Mac) untuk membuka SQL Editor baru.

## Menjalankan Query SQL

### Execute Statement

Highlight query, lalu klik **Execute SQL Statement** (ikon play hijau) atau **Ctrl+Enter**.

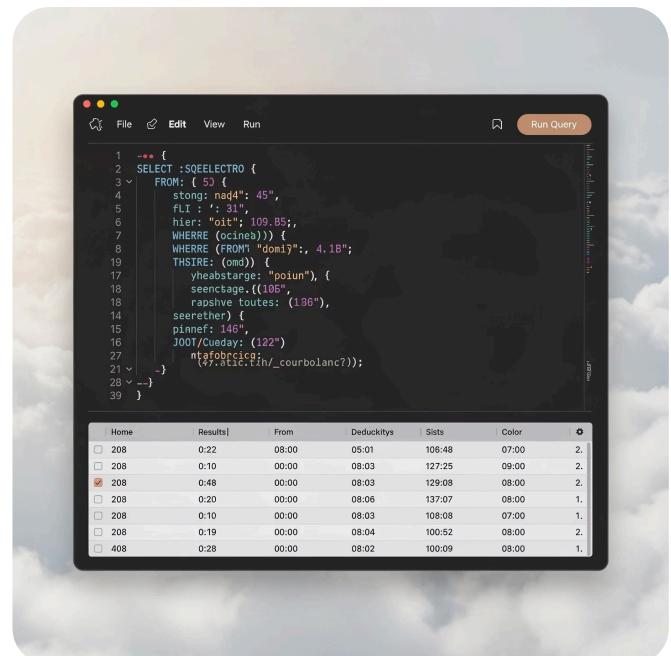
### Execute Script

Untuk menjalankan beberapa query, gunakan **Execute SQL Script** (**Ctrl+Alt+X**). Semua query akan dijalankan berurutan.

### Execute Current

Letakkan kursor di dalam query dan tekan **Ctrl+Enter**. DBeaver akan otomatis menjalankan query tersebut.

## Interface SQL Editor



1. **Toolbar:** Tombol untuk execute, format, save, dll.
2. **Editor Area:** Area utama untuk mengetik SQL query dengan syntax highlighting.
3. **Results Tab:** Menampilkan hasil query dalam tabel interaktif.
4. **Log Tab:** Menampilkan log eksekusi, error, dan informasi debugging.

# Perintah SQL Dasar: SELECT dan WHERE

**SELECT** adalah perintah fundamental SQL untuk mengambil data. Anda bisa memilih kolom spesifik atau semua kolom dari tabel.

**WHERE** digunakan untuk memfilter data berdasarkan kondisi tertentu, membuat query lebih spesifik dan efisien.

## Format Dasar SELECT

```
SELECT kolom1, kolom2, ...
FROM nama_tabel;
```

### Contoh 1: Semua Kolom

```
-- Semua kolom dari m_kbli
SELECT * FROM m_kbli;
```

Mengambil seluruh data dari tabel m\_kbli.

### Contoh 2: Kolom Spesifik

```
-- Kolom kode dan judul
SELECT kode, judul FROM m_kbli;
```

Hanya mengambil kolom kode dan judul.

## Menggunakan WHERE untuk Filter Data

Operator	Fungsi	Contoh
=	Sama dengan	WHERE kategori = 'A'
!=	Tidak sama dengan	WHERE kategori != 'A'
>	Lebih besar	WHERE kode > '01000'
LIKE	Pencarian teks	WHERE judul LIKE '%industri%'

### Filter Kategori Spesifik

```
SELECT kode, judul, kategori
FROM m_kbli
WHERE kategori = 'A'
LIMIT 3;
```

**Hasil:** KBLI dengan kategori A (Pertanian, Kehutanan, Perikanan).

### Pencarian Teks (LIKE)

```
SELECT kode, judul
FROM m_kbli
WHERE judul LIKE '%industri%'
LIMIT 3;
```

**Hasil:** Semua KBLI mengandung kata "industri". Tanda % berarti "karakter apapun".

# Perintah SQL: ORDER BY dan GROUP BY

## ORDER BY - Mengurutkan Data

**ORDER BY** mengurutkan hasil query berdasarkan satu atau lebih kolom, secara ascending (naik) atau descending (turun).

### Format Dasar:

```
SELECT kolom1, kolom2  
FROM nama_tabel  
ORDER BY kolom1 ASC; -- atau DESC
```

- **ASC (Ascending)**

Urutkan nilai naik (A-Z, 1-9, tanggal lama)

- **DESC (Descending)**

Urutkan nilai turun (Z-A, 9-1, tanggal baru)

### Contoh ORDER BY:

```
-- Urutkan naik berdasarkan kode  
SELECT * FROM m_kbli  
ORDER BY kode ASC;
```

### ❑ Perbedaan WHERE dan HAVING

**WHERE:** Filter baris individu sebelum GROUP BY.

**HAVING:** Filter hasil agregat setelah GROUP BY.

**Contoh:** WHERE kategori = 'A' vs  
HAVING COUNT(\*) > 100

## GROUP BY - Mengelompokkan Data

**GROUP BY** mengelompokkan baris dengan nilai sama dalam kolom tertentu. Sering digunakan dengan fungsi agregat (COUNT, SUM, AVG) untuk analisis data.

### Format Dasar:

```
SELECT kolom1, fungsi_agregat(kolom2)  
FROM nama_tabel  
GROUP BY kolom1;
```

## COU... SUM()

Menghitung

Jumlah baris

Menjumlahkan

Total nilai

## AVG()

Rata-rata

Nilai rata-rata

### Contoh GROUP BY:

```
-- Hitung jumlah per kategori  
SELECT kategori, COUNT(*) as jumlah  
FROM m_kbli  
GROUP BY kategori;
```

# JOIN - Menggabungkan Data dari Multiple Tabel

**JOIN** adalah fitur SQL powerful untuk menggabungkan data dari dua atau lebih tabel berdasarkan kolom yang berhubungan. Ini penting dalam database relasional seperti OSS-RBA, di mana data efisinya tersebar di beberapa tabel.

Misalnya, menggabungkan tabel KBLI dan tabel mapping izin untuk melihat KBLI beserta izin-izin terkait dalam satu query.

## Contoh Praktis JOIN di OSS-RBA

```
-- Mendapatkan KBLI beserta semua izinnya
SELECT
    k.kode,
    k.judul,
    ki.id_lic,
    ki.nama_izin
FROM m_kbli k
INNER JOIN m_kbli_izin ki ON k.kode = ki.kbli
WHERE k.kategori = 'C'
ORDER BY k.kode, ki.id_lic;
```

Query ini menampilkan semua KBLI kategori 'C' dan izin terkaitnya, diurutkan berdasarkan kode KBLI dan ID izin.

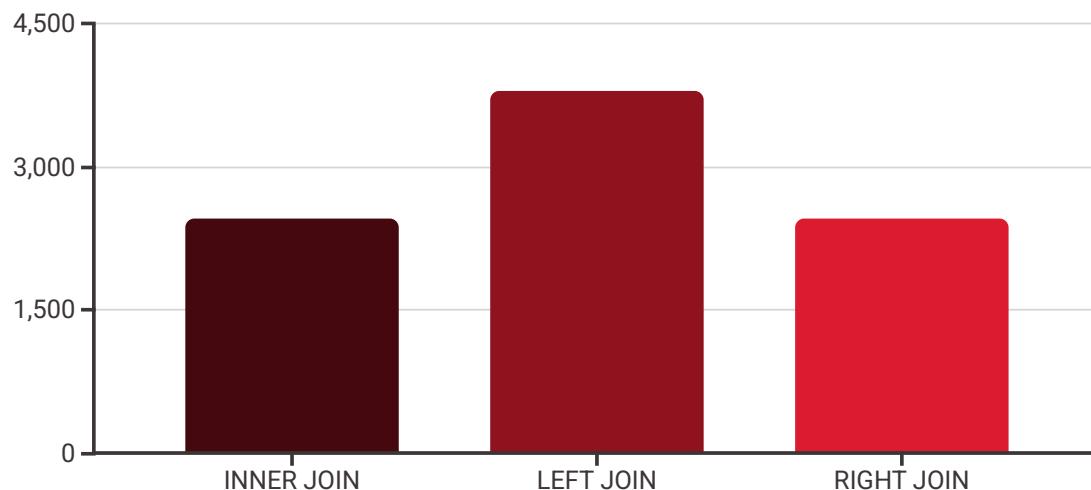


Chart di atas menunjukkan jumlah baris hasil dari berbagai tipe JOIN saat menggabungkan m\_kbli (3793 baris) dan m\_kbli\_izin. INNER JOIN hanya mengambil KBLI yang punya izin, sedangkan LEFT JOIN mengambil semua KBLI, termasuk yang tidak punya izin.

# Fungsi-Fungsi SQL yang Berguna

SQL menyediakan berbagai fungsi built-in untuk memanipulasi dan menganalisis data, dibagi menjadi kategori seperti string, date, dan aggregate functions.

## String Functions

Fungsi untuk manipulasi teks:

- **UPPER()/LOWER():** Ubah huruf
- **LENGTH():** Hitung panjang
- **SUBSTRING():** Ambil bagian

```
SELECT UPPER(judul) FROM  
m_kbli;
```

## Date Functions

Fungsi untuk operasi tanggal/waktu:

- **CURRENT\_DATE:** Tanggal hari ini
- **CURRENT\_TIMESTAMP:** Tanggal & waktu
- **EXTRACT():** Ambil bagian tanggal

```
SELECT EXTRACT(YEAR  
FROM created_at) FROM  
t_nib;
```

## Aggregate Functions

Fungsi kalkulasi statistik:

- **COUNT():** Hitung baris
- **SUM():** Jumlah nilai
- **AVG():** Rata-rata

```
SELECT COUNT(*) FROM  
m_kbli;
```

# Contoh Praktis: Kombinasi Fungsi SQL

## Analisis Data NIB per Bulan

```
SELECT
    DATE_TRUNC('month', created_at) as bulan,
    COUNT(*) as jumlah_nib,
    COUNT(DISTINCT pelaku_usaha_id) as jumlah_pelaku
FROM t_nib
WHERE EXTRACT(YEAR FROM created_at) = 2024
GROUP BY DATE_TRUNC('month', created_at)
ORDER BY bulan;
```

Analisis NIB per bulan di 2024, termasuk jumlah pelaku usaha unik.

## Pencarian KBLI dengan Text Processing

```
SELECT
    kode,
    UPPER(judul) as judul_upper,
    LENGTH(judul) as panjang_judul,
    SUBSTRING(kode, 1, 2) as kategori_kode
FROM m_kbli
WHERE LOWER(judul) LIKE '%manufaktur%'
ORDER BY panjang_judul DESC
LIMIT 10;
```

Mencari KBLI dengan kata 'manufaktur' (case-insensitive) dan manipulasi string.

# Tips & Trik Produktivitas DBeaver

DBeaver memiliki banyak fitur untuk meningkatkan produktivitas dalam mengelola database. Kuasai tips ini untuk bekerja lebih cepat dan efisien.



## Cari Database (Ctrl+Shift+D)

Temukan tabel atau kolom di seluruh database dengan cepat.



## Lihat Data Cepat

Klik kanan tabel → **View Data** untuk pratinjau instan.



## Eksport Mudah

Eksport hasil query ke Excel/CSV langsung.

### Visual Query Builder

Buat query via drag-and-drop; generate SQL otomatis. Ideal untuk pemula & query kompleks.

### Generate SQL Otomatis

Hasilkan template SELECT/INSERT/UPDATE/DELETE dari tabel. Hemat waktu penulisan.

### Bookmarks & Snippets

Simpan query favorit (bookmark) atau buat template (snippets) untuk akses cepat.

### Analisis Execution Plan

Periksa tab Execution Plan untuk optimasi performa query.

## Keyboard Shortcuts Esensial

Aksi	Windows/Linux	Mac
Eksekusi Query	Ctrl+Enter	Cmd+Enter
Format SQL	Ctrl+Shift+F	Cmd+Shift+F
Auto-complete	Ctrl+Space	Cmd+Space
Cari Database	Ctrl+Shift+D	Cmd+Shift+D
Skrip SQL Baru	Ctrl+Shift+Enter	Cmd+Shift+Enter

# Troubleshooting: Mengatasi Masalah Umum

Atasi masalah umum DBeaver dengan solusi cepat berikut.



## Masalah Koneksi

"**Connection refused**": Cek host/port, firewall, status server.

"**Authentication failed**": Verifikasi kredensial & izin.



## Masalah Query SQL

**Query lambat**: Gunakan LIMIT/WHERE, cek execution plan.

"**Column not exist**": Cek nama kolom (case-sensitive).



## Masalah Performa DBeaver

**Lambat/hang**: Tutup koneksi, restart DBeaver, pakai LIMIT.

**Auto-complete error**: Refresh metadata.

## Best Practices untuk Menghindari Masalah

01

### Test dengan LIMIT

Selalu gunakan LIMIT pada query data besar sebelum eksekusi penuh.

```
SELECT * FROM table LIMIT  
10;
```

02

### Backup Sebelum Perubahan

Backup data dengan SELECT sebelum UPDATE/DELETE.

```
-- Backup data  
SELECT * FROM table  
WHERE id = 'xyz';
```

03

### Gunakan Transaction

Gunakan BEGIN, COMMIT/ROLLBACK untuk perubahan data penting.



### Tips Keamanan Data

**PENTING:** Tim audit sebaiknya gunakan akses **READ-ONLY**. Hindari UPDATE, DELETE, INSERT di database production tanpa izin. Verifikasi izin dengan administrator.

# Latihan Praktis dan Studi Kasus

Praktik adalah kunci menguasai SQL dan DBeaver. Bagian ini menyediakan latihan hands-on untuk mengaplikasikan pengetahuan ke skenario nyata OSS-RBA.

Latihan dirancang bertahap, dari setup dasar hingga query kompleks (JOIN, analisis data). Cobalah setiap latihan dan bandingkan hasilnya.

## Studi Kasus: Verifikasi Data NIB

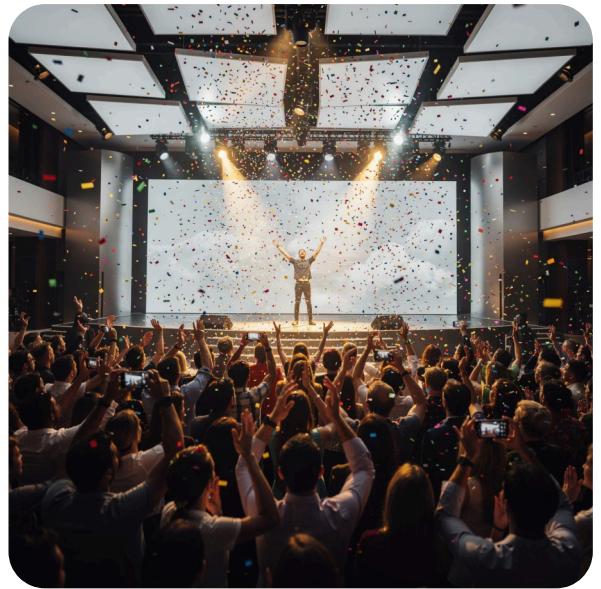
**Skenario:** Pengembang melaporkan 1.250 NIB kategori KBLI 'C' dibuat Januari 2024. Tim SPB perlu verifikasi data ini secara independen.

```
-- Query untuk verifikasi
SELECT
    COUNT(*) as total_nib,
    COUNT(DISTINCT pelaku_usaha_id) as total_pelaku_usaha,
    DATE_TRUNC('month', created_at) as bulan
FROM t_nib n
INNER JOIN t_nib_kbli nk ON n.id = nk.nib_id
INNER JOIN m_kbli k ON nk.kbli_code = k.kode
WHERE k.kategori = 'C'
    AND EXTRACT(YEAR FROM n.created_at) = 2024
    AND EXTRACT(MONTH FROM n.created_at) = 1
GROUP BY DATE_TRUNC('month', n.created_at);
```

Query ini menghitung NIB kategori KBLI tertentu di periode spesifik. Bandingkan hasilnya dengan laporan Pengembang untuk akurasi data.

# Kesimpulan dan Langkah Selanjutnya

Selamat! Anda kini memiliki **kemampuan kritis** dalam SQL Dasar dan DBeaver, memungkinkan Anda mengakses, menganalisis, dan memverifikasi data secara mandiri. Ini akan meningkatkan independensi peserta dalam triangulasi informasi, verifikasi klaim Pengembang, dan pengambilan keputusan berbasis data yang akurat.



## Langkah Selanjutnya untuk Pengembangan Skill



### Praktik Konsisten

Gunakan SQL & DBeaver sehari-hari untuk memperkuat pemahaman.

### Tingkat Lanjut

Pelajari subquery, window functions, & optimasi query untuk analisis kompleks.

### Berbagi Pengetahuan

Bagikan pembelajaran & bangun *knowledge base* tim.

## Referensi dan Sumber Belajar

### Dokumentasi

- [DBeaver Documentation](#)
- [PostgreSQL Official Docs](#)
- [W3Schools SQL Tutorial](#)

### Penutup

Dokumen ini dibuat khusus untuk membantu Peserta Training memahami dasar-dasar SQL dan menggunakan DBeaver dengan efektif.