

Week 02

Pendahuluan PostgreSQL 17

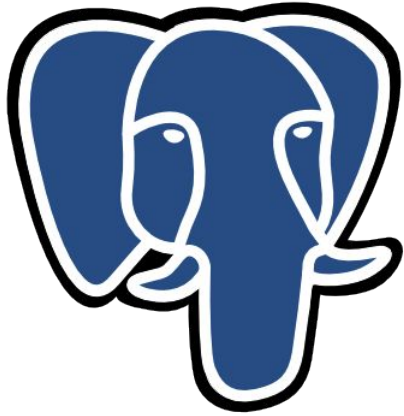
Arif Wicaksono Septyanto, S.Kom.,M.Kom

Sistem Informasi, Institut Teknologi Kalimantan

08/09/2025



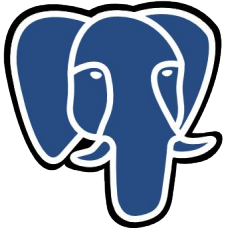
Pendahuluan PostgreSQL 17



PostgreSQL

- PostgreSQL bersifat **open source**, artinya Anda bebas menginstal, menggunakan, dan menyebarkan PostgreSQL tanpa harus membayar biaya atau royalti.
- Basis data stabil mampu berjalan lama tanpa dimatikan.
- Dikembangkan lebih dari 30 tahun, dibuat oleh kelompok penelitian basis data di **University of California, Barkeley**.
- Kini dikembangkan oleh komunitas besar pengembang dan kontributor.

Fitur Utama PostgreSQL



PostgreSQL

Fitur utama postgresQL :

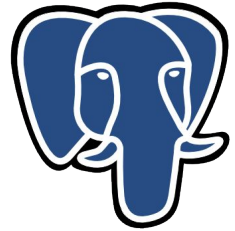
1. Kompatibilitas tinggi dengan **standar SQL**,
2. Arsitektur **Client - Server**
3. Dirancang dengan konkurensi tinggi, dimana **pembaca dan penulis tidak saling menghalangi**.
4. Sangat **fleksibel** dan dapat dikembangkan, cocok **berbagi jenis aplikasi**.
5. Skalabilitas dan kinerja tinggi, dilengkapi berbagai opsi **penyetelan (tuning)**
6. Dukungan berbagai model data, seperti relasi (tabel biasa), **Data pasca relasional (array, relasi bersarang)**, data dokumen (**JSON, XML**), data kunci (key/value)

Pembeda PostgreSQL

PostgreSQL fokus pada 4 tujuan utama :

1. Menghasilkan perangkat lunak yang **kuat dan berkualitas**.
2. **Murah perawatan**, baik pengguna kecil maupun skala besar
3. Mengikuti standar **SQL dan kompatibilitas**.
4. **Keamanan tinggi**.

Fitur-fitur PostgreSQL lebih mirip **Oracle atau SQL Server** daripada MySQL

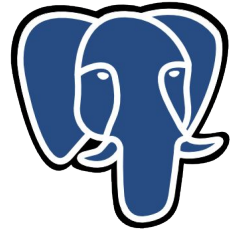


PostgreSQL

Pembeda PostgreSQL

PostgreSQL memberikan banyak pilihan :

1. Model basis data terstruktur (normalisasi) dengan tambahan fitur seperti array dan tipe rekaman.
2. Gunakan skema fleksibel dengan **JSONB** atau ekstensi **hstore**.
3. Buat fungsi server-side berbagai bahasa, seperti Python, Perl, JavaScript, dan lainnya.



PostgreSQL

Contoh Penggunaan JSONB dan hstore

JSONB merupakan tipe data biner untuk menyimpan dokumen JSON secara **efisien**, mendukung indeks dan pencarian cepat. Contoh : Menyimpan data profil pengguna yang fleksibel tanpa membuat kolom baru setiap kali ada atribut tambahan.

```
#membuat tabel
```

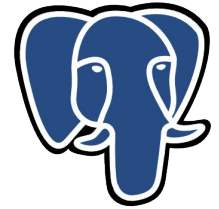
```
CREATE TABLE pengguna (  
    id serial PRIMARY KEY,  
    nama text,  
    data_lain jsonb  
);
```

```
#masukkan data
```

```
INSERT INTO pengguna (nama, data_lain) VALUES  
(  
    'Andi', '{  
        "hobi": ["membaca", "bersepeda"],  
        "alamat": "Jakarta",  
        "notifikasi": true,  
        "umur": 28  
    }'  
);
```

```
-- Cari pengguna yang hobi-nya bersepeda
```

```
SELECT nama FROM pengguna  
WHERE data_lain->'hobi' ? 'bersepeda';
```



PostgreSQL

Contoh Penggunaan JSONB dan hstore

hstore adalah ekstensi PostgreSQL untuk menyimpan pasangan kunci-nilai (key => value), mirip dengan kamus (dictionary).

```
#aktifkan ekstensi
```

```
CREATE EXTENSION IF NOT EXISTS hstore;
```

```
#buat tabel
```

```
CREATE TABLE produk (  
    id serial PRIMARY KEY,  
    nama text,  
    atribut hstore  
);
```

```
-- Cari produk berwarna hitam
```

```
SELECT nama FROM produk  
WHERE atribut->'warna' = 'hitam';
```

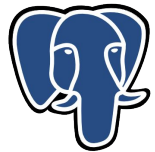
```
-- Tampilkan semua atribut laptop
```

```
SELECT atribut FROM produk WHERE nama = 'Laptop';
```

Hasil: "warna"=>"hitam", "ram"=>"16GB", "tipe"=>"gaming"

```
#masukkan data
```

```
INSERT INTO produk (nama, atribut) VALUES  
(  
    'Laptop', ('warna' => "hitam", "ram" => "16GB", "tipe" => "gaming")  
,  
    'Mouse', ('warna' => "putih", "baterai" => "rechargeable")  
);
```



PostgreSQL

Contoh Penggunaan JSONB dan hstore

Fitur	JSONB	hstore
Format	JSON (struktur bersarang)	Kunci-nilai sederhana
Tipe Data	String, angka, array, boolean, objek	Hanya string
Indeks	Didukung (GIN), sangat cepat	Didukung
Fleksibilitas	Tinggi, cocok untuk data kompleks	Rendah, cocok untuk metadata sederhana

Kapan Menggunakan JSONB atau hstore

- Gunakan JSONB jika:
Perlu menyimpan data bersarang seperti profil, pengaturan, atau dokumen dengan struktur kompleks.
- Gunakan hstore jika:
Butuh menyimpan pasangan kunci-nilai sederhana, seperti tag atau konfigurasi cepat.

Ekstensibilitas (Kemampuan Dikembangkan)

PostgreSQL dirancang sangat ekstensibel:

1. Menambahkan tipe data, operator, atau indeks baru.
2. Membuat bahasa pemrograman server sendiri.
3. Menggunakan plugin untuk mengubah cara eksekusi perintah atau menambah optimizer kueri.

Fitur CREATE EXTENSION memudahkan penambahan ekstensi dengan otomatisasi versi dan ketergantungan. Contoh ekstensi populer:

➔ PostGIS: Sistem Informasi Geografis (GIS) kelas dunia untuk aplikasi peta dan analisis lokasi.

Ekstensi PostGIS di PostgreSQL

Sebelum menggunakan PostGIS, PostgreSQL tidak bisa memahami tipe data seperti "titik di peta" atau "garis lintasan". Tapi dengan ekstensibilitasnya, kita bisa menambahkan kemampuan itu hanya dengan menginstal dan mengaktifkan ekstensi.

```
-- Aktifkan ekstensi PostGIS di database Anda  
CREATE EXTENSION IF NOT EXISTS postgis;
```

Perintah ini secara otomatis:

- Menambahkan tipe data baru (seperti `geometry`, `geography`)
- Menambahkan fungsi baru (seperti `ST_Distance`, `ST_Buffer`)
- Mengatur ketergantungan dan versi secara internal

Membuat Tabel dengan Kolom Geospasial

```
CREATE TABLE lokasi_penting (  
    id SERIAL PRIMARY KEY,  
    nama VARCHAR(100),  
    koordinat GEOMETRY(POINT, 4326) -- Titik (latitude/longitude) dalam sistem  
WGS84  
);
```

📌 GEOMETRY(POINT, 4326) adalah tipe data baru yang ditambahkan oleh PostGIS.
Tanpa ekstensi ini, PostgreSQL tidak mengenal GEOMETRY.

```
INSERT INTO lokasi_penting (nama, koordinat)  
VALUES  
    ('Monas', ST_SetSRID(ST_MakePoint( 106.8272, -6.1754), 4326)),  
    ('GBK', ST_SetSRID(ST_MakePoint( 106.7925, -6.2259), 4326));
```

ST_MakePoint, ST_SetSRID → fungsi-fungsi baru dari PostGIS

Ini adalah fungsi query spasial yang tidak ada di PostgreSQL dasar.

4326 adalah kode EPSG (European Petroleum Survey Group) untuk sistem koordinat geografis

WGS84 (World Geodetic System 1984)

Contoh Gunakan Fungsi Spasial (Analisis Lokasi)

```
SELECT
    l1.nama AS lokasi1,
    l2.nama AS lokasi2,
    ST_Distance(l1.koordinat::geography, l2.koordinat::geography) AS jarak_meter
FROM
    lokasi_penting l1,
    lokasi_penting l2
WHERE
    l1.nama = 'Monas' AND l2.nama = 'GBK';
```

Hasil: Misalnya 9850.23 meter (sekitar 9.85 km)

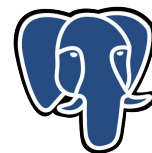
Ini adalah analisis geospasial yang sangat kompleks, tapi dijalankan langsung di database!



PostgreSQL

Manfaat Nyata dari Ekstensibilitas

Fitur	Disediakan oleh
Tipe data <code>GEOMETRY</code>	Ekstensi PostGIS (ditambahkan ke PostgreSQL)
Fungsi <code>ST_Distance</code>	Ekstensi PostGIS
Optimasi kueri spasial	PostGIS + optimizer PostgreSQL yang bisa diperluas
Manajemen versi & dependensi	<code>CREATE EXTENSION</code> — fitur bawaan PostgreSQL



PostgreSQL

Analogi sederhana

Bayangkan PostgreSQL seperti smartphone Android:

- Bawaan: telepon, kamera, browser (seperti tipe data `INT`, `TEXT`, `DATE`)
- Tapi kamu bisa pasang aplikasi (ekstensi) seperti:
 - Google Maps → seperti `PostGIS`
 - WhatsApp → seperti `pg_cron` (untuk job scheduling)
 - Adblock → seperti `pg_qualstats` (untuk analisis kueri)

Dan semua ini terintegrasi langsung ke "sistem operasi"-nya (PostgreSQL).



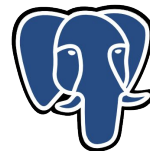
PostgreSQL

Contoh Ekstensi Lain (Selain PostGIS)

Ekstensi	Fungsi
<code>pg_cron</code>	Menjadwalkan kueri seperti cron job
<code>uuid-oss</code>	Generate UUID otomatis
<code>hstore</code>	Menyimpan data key-value di dalam kolom
<code>timescaledb</code>	Database time-series (untuk IoT, monitoring)
<code>pg_trgm</code>	Pencarian teks mirip (fuzzy search)

Semuanya diaktifkan dengan:

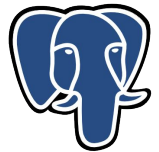
```
CREATE EXTENSION nama_ekstensi;
```



PostgreSQL

Kinerja dan Konkurensi Tingkat Tinggi

- Lebih dari 1 juta operasi pembacaan per detik pada server dengan 4 soket CPU.
- Lebih dari 30.000 transaksi penulisan per detik, dengan jaminan durability — artinya data tetap aman meskipun terjadi kegagalan sistem.
- Optimizer kueri canggih yang memilih jalur eksekusi tercepat.
- MVCC (Multi-Version Concurrency Control) yang memisahkan proses baca dan tulis.
- Transaksi harian (OLTP) – cepat dan andal
- Analisis data (OLAP) – query kompleks dan agregasi besar
- Pencarian teks dan data semi-terstruktur – seperti JSON, XML, atau data geospasial



PostgreSQL

Populer di Seluruh Dunia

PostgreSQL digunakan oleh banyak perusahaan besar, seperti:

- Apple, Skype, McAfee, Heroku, IMDB, NTT, BASF, Genentech
- UK Met Office, US National Weather Service

Pada 2010, PostgreSQL telah diunduh lebih dari 1 juta kali per tahun. Komisi Eropa menyatakan bahwa "PostgreSQL dianggap sebagai alternatif yang bisa diandalkan."

Dr. Werner Vogels, CTO Amazon Web Services (AWS), menyebut PostgreSQL sebagai "**database yang luar biasa**" dan "pilihan utama bagi pengembang perusahaan dan startup." AWS juga mengumumkan bahwa PostgreSQL adalah layanan mereka yang paling cepat berkembang.



PostgreSQL

Dukungan Komersial

Meski open source, PostgreSQL memiliki dukungan profesional global dari berbagai perusahaan, seperti:

- EnterpriseDB (EDB): Perusahaan terbesar penyedia dukungan komersial, tempat penulis buku ini bekerja. Menyediakan dukungan 24/7 dalam bahasa Inggris.
- Banyak perusahaan lain menyediakan dukungan lokal, vertikal, atau teknis khusus.

PostgreSQL juga tersedia sebagai layanan cloud (managed service) dari AWS, Google Cloud, Azure, dan penyedia lainnya.



PostgreSQL

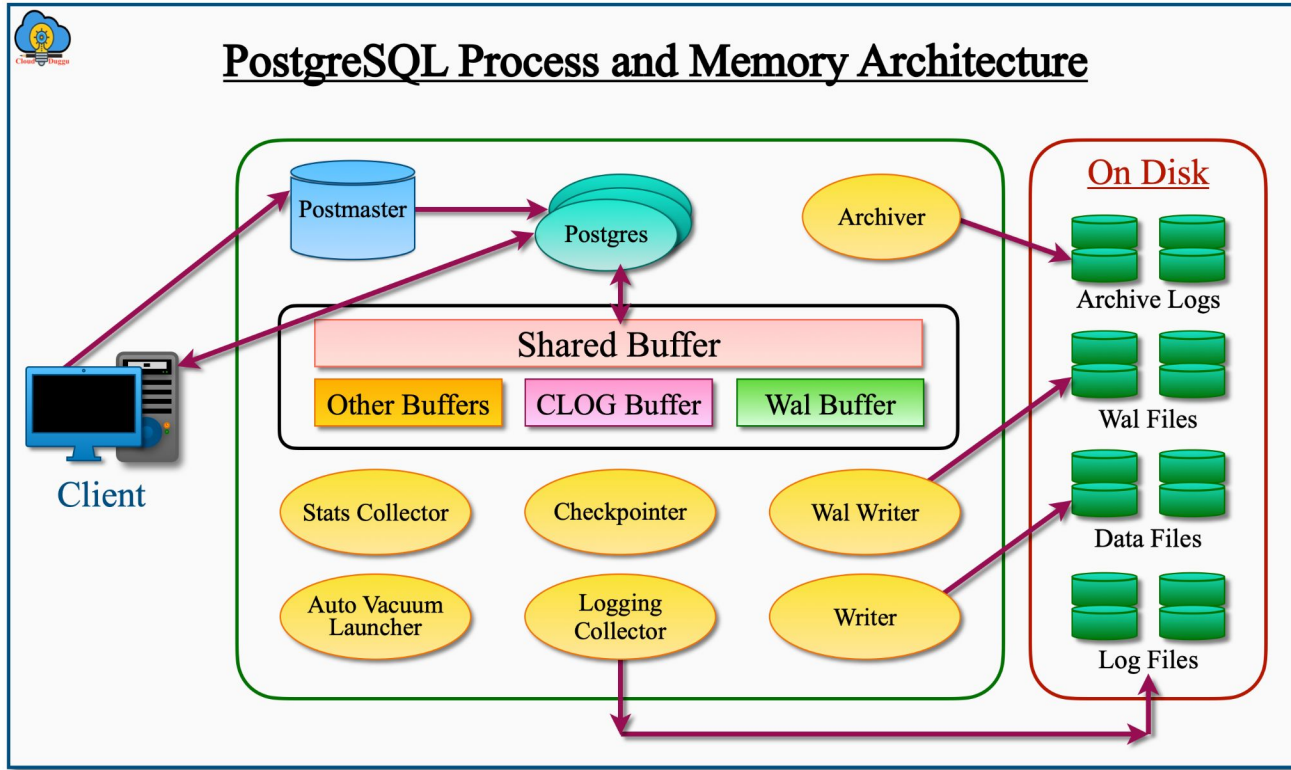
Arsitektur PostgreSQL

PostgreSQL adalah sistem basis data open source yang canggih, pertama kali dikembangkan pada 8 Juli 1996 di University of California, Berkeley. Sejak itu, terus diperbarui dan ditingkatkan oleh komunitas pengembang aktif dari seluruh dunia.

PostgreSQL menggunakan model client-server, artinya:

- Anda (pengguna) mengirim permintaan lewat aplikasi atau alat (client).
- Server PostgreSQL menerima permintaan, memprosesnya, lalu mengirim hasilnya kembali ke client.
- Setiap koneksi dari client akan membuat proses baru di server yang disebut postgres.
- Port default untuk koneksi ini adalah 5432, tapi bisa diubah lewat file konfigurasi postgresql.conf.

Arsitektur PostgreSQL



1. Postmaster (Pengawas Utama)

Fungsi : Proses pertama saat PostgreSQL dijalankan, Bertindak sebagai receptionist atau pengawas. Tugas utama :

- Menerima Koneksi client.
- Memeriksa login dan izin (otentikasi).
- Membuat proses baru (postgres) untuk setiap koneksi.
- Memantau proses-proses tersebut dan menjalankannya kembali.

2. Shared Memory – Memori Bersama

- Tempat penyimpanan sementara di memori (RAM) untuk data yang sering digunakan.
- Menyimpan salinan dari: Baris data tabel, Indeks (yang membantu pencarian lebih cepat)

Analogi aplikasi meminta data:

- PostgreSQL cek dulu di Shared Buffer.
- Kalau datanya ada → ambil dari sana (cepat!).
- Kalau tidak ada → baru baca dari disk (lebih lambat).
- Diatur lewat parameter: `shared_buffers` di file `postgresql.conf`.

3. WAL Buffer (Write-Ahead Log Buffer)

- Menyimpan catatan perubahan data sebelum benar-benar disimpan di disk.
- Fungsinya: jika server mati tiba-tiba, data tidak hilang.
- Dikelola oleh parameter: `wal_buffers`.

4. CLOG Buffer (Commit Log)

- Menyimpan informasi: apakah suatu transaksi sudah selesai (commit) atau belum.
- Tidak perlu diatur manual — sistem mengelolanya otomatis.

5. Work Memory

- Digunakan untuk operasi seperti:
 - ◆ Mengurutkan data (ORDER BY)
 - ◆ Menggabungkan tabel (JOIN)
 - ◆ Menghitung grup (GROUP BY)

6. Maintenance Work Memory

- Digunakan untuk tugas pemeliharaan seperti:
 - ◆ Membuat indeks (CREATE INDEX)
 - ◆ Membersihkan data (VACUUM)
 - ◆ Diatur lewat: `maintenance_work_mem`

7. Temp Buffers

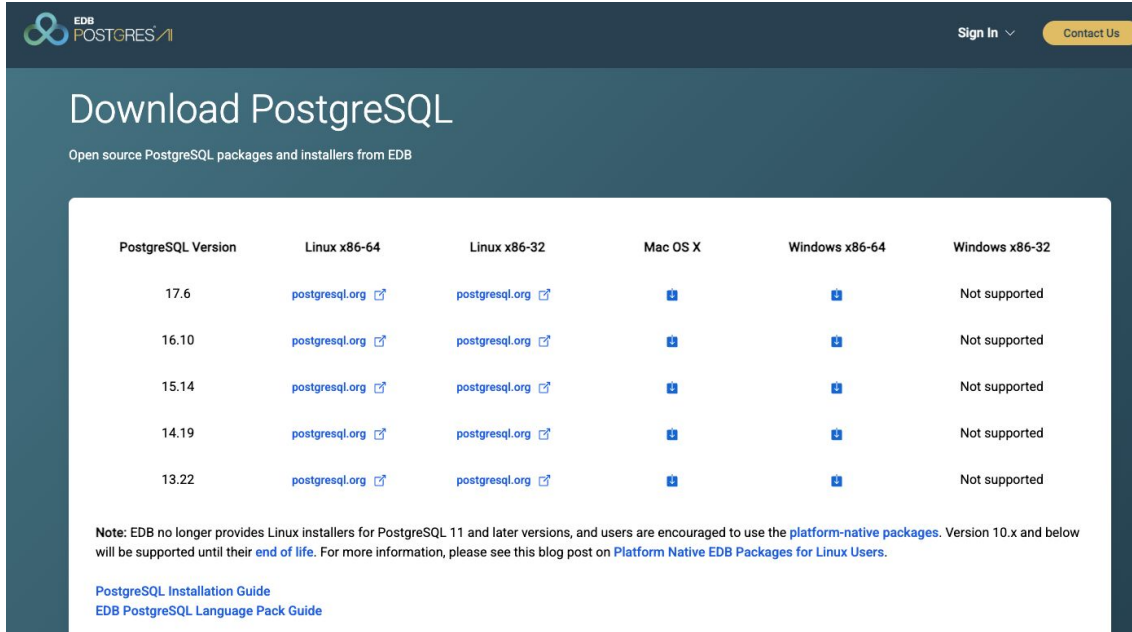
- Digunakan saat Anda membuat tabel sementara.
- Hanya berlaku untuk satu sesi (koneksi).

Arsitektur PostgreSQL

Proses Latar Belakang (Background Processes)

1. Background Writer : Menulis data yang sudah diubah (dirty pages) dari shared buffer ke file di disk, Agar selalu ada ruang kosong di memori untuk data baru.
2. Checkpointer : PostgreSQL menyimpan semua perubahan dari memori ke disk — ini disebut checkpoint. Tujuannya: mempercepat pemulihan jika terjadi crash.
3. Autovacuum Launcher : Membersihkan data yang sudah dihapus agar tidak memenuhi ruang. Contoh: Jika Anda DELETE data, PostgreSQL tidak langsung menghapusnya dari file, Autovacuum yang membersihkannya.
4. WAL Writer : Menulis isi WAL Buffer ke file WAL di disk setelah COMMIT. Memastikan perubahan tersimpan permanen.
5. Statistics Collector : Mengumpulkan data statistik: Berapa banyak baris di tabel?, Indeks mana yang sering digunakan?, Bagaimana pola akses pengguna?
6. Logging Collector : Mengumpulkan semua pesan log (error, peringatan, info) dan menyimpannya di file log.
7. Archiver : Jika mode archive aktif, proses ini menyalin file WAL ke lokasi cadangan.

Instalasi



The screenshot shows the EDB PostgreSQL download page. It features a dark header with the EDB logo and navigation links for 'Sign In' and 'Contact Us'. The main heading is 'Download PostgreSQL' with the subtitle 'Open source PostgreSQL packages and installers from EDB'. Below this is a table listing PostgreSQL versions and their corresponding download links for different operating systems.

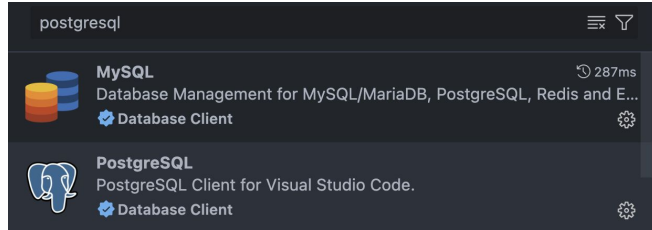
PostgreSQL Version	Linux x86-64	Linux x86-32	Mac OS X	Windows x86-64	Windows x86-32
17.6	postgresql.org	postgresql.org			Not supported
16.10	postgresql.org	postgresql.org			Not supported
15.14	postgresql.org	postgresql.org			Not supported
14.19	postgresql.org	postgresql.org			Not supported
13.22	postgresql.org	postgresql.org			Not supported

Note: EDB no longer provides Linux installers for PostgreSQL 11 and later versions, and users are encouraged to use the [platform-native packages](#). Version 10.x and below will be supported until their [end of life](#). For more information, please see this blog post on [Platform Native EDB Packages for Linux Users](#).

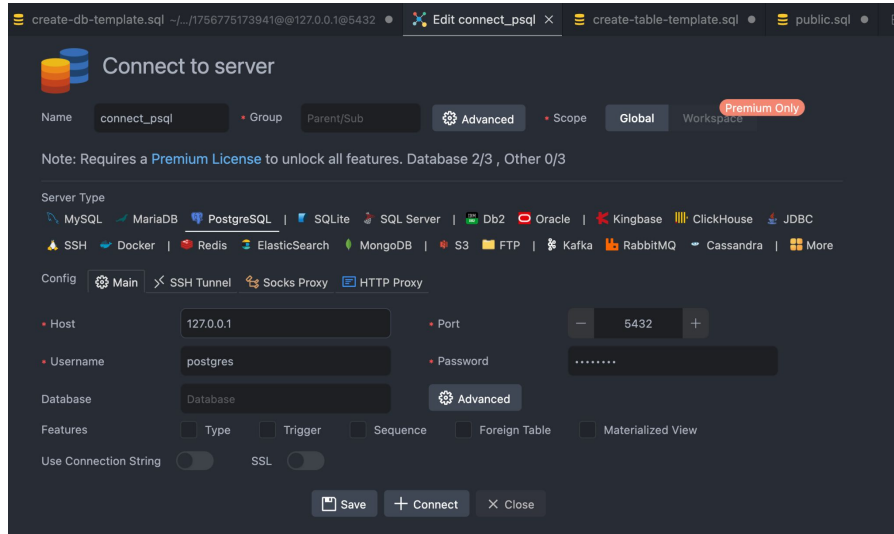
[PostgreSQL Installation Guide](#)
[EDB PostgreSQL Language Pack Guide](#)

<https://www.enterprisedb.com/downloads/postgres-postgresql-downloads>

Instalasi Database Client



Host : Localhost / 127.0.0.1
Username : postgres (default)
Port : 5432 (default)



Penggunaan Database Client

```

create-db-template.sql ~/.../1756775173941@127.0.0.1@5432 • Edit connect_psql create-table-template
Users > arifwicaksonoseptyanto > .dbclient > storage > 1756775173941@127.0.0.1@5432@penjualan_db@public > c
1 -- Buat tabel customers di PostgreSQL
  >Run | >Reset | >Select
2 CREATE TABLE IF NOT EXISTS customers (
3     customer_id SERIAL PRIMARY KEY,
4     name VARCHAR(100) NOT NULL,
5     email VARCHAR(100) UNIQUE NOT NULL,
6     phone VARCHAR(15),
7     address TEXT,
8     birthdate DATE -- Tanggal lahir pelanggan
9 );
10
11 >Run | >Reset | >Select
12 CREATE TABLE IF NOT EXISTS products (
13     product_id SERIAL PRIMARY KEY,
14     name VARCHAR(100) NOT NULL,
15     description TEXT,
16     price DECIMAL(10, 2) NOT NULL,
17     stock INT NOT NULL
18 );
19 >Run | >Reset | >Select
20 CREATE TABLE IF NOT EXISTS orders (
21     order_id SERIAL PRIMARY KEY,
22     customer_id INT NOT NULL,
23     order_date TIMESTAMP DEFAULT CURRENT_TIMESTAMP,
24     total_amount DECIMAL(10, 2) NOT NULL,
25     FOREIGN KEY (customer_id) REFERENCES customers(customer_id)
26     ON DELETE CASCADE
27 );
  
```

create-db-template.sql ~/.../1756775173941@127.0.0.1@5432 • Edit connect_psql create-table-template.sql customers x pul

Properties DATA Log ER Monitor

SELECT * FROM customers LIMIT 100

	customer_id integer	name varchar(100)	email varchar(100)	phone varchar(15)	address text	birthdate date
1		Andi Pratama	andi.pratama@email.com	081234567890	Jl. Merdeka No. 10, Jakarta	1990-05-15
2		Budi Santoso	budi.santoso@email.com	081234567891	Jl. Sudirman No. 25, Bandung	1985-08-22
3		Citra Dewi	citra.dewi@email.com	081234567892	Jl. Diponegoro No. 33, Suraba	1992-11-30
4		Dewi Lestari	dewi.lestari@email.com	081234567893	Jl. Gatot Subroto No. 45, Yogy	1988-03-10
5		Eko Wijaya	eko.wijaya@email.com	081234567894	Jl. Ahmad Yani No. 18, Medan	1995-07-25

Keunggulan penggunaan Database client :

1. Lebih ringan dibanding pgAdmin4
2. Lebih mudah memonitoring data.
3. Di Dalam environment VS Code

ERD Refresher

Notasi Arti



• ENTITY



• WEAK ENTITY



• RELATIONSHIP



• IDENTIFYING
RELATIONSHIP



• ATRIBUT
DERIVATIF

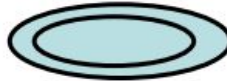
Notasi Arti



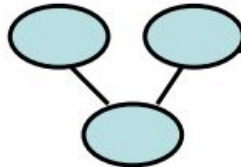
• ATRIBUT



• ATRIBUT
PRIMARY KEY

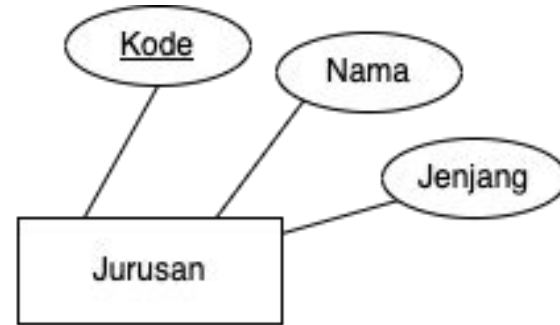
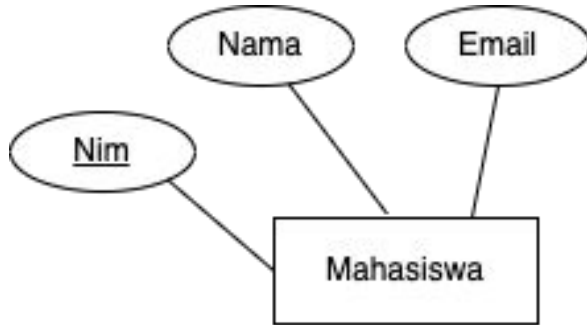


• ATRIBUT MULTI
VALUE

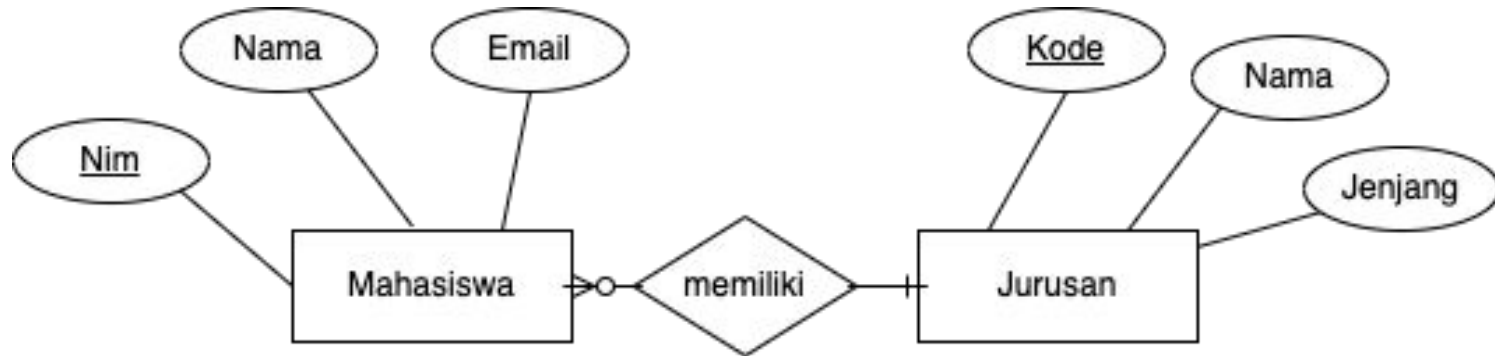


• ATRIBUT
COMPOSITE

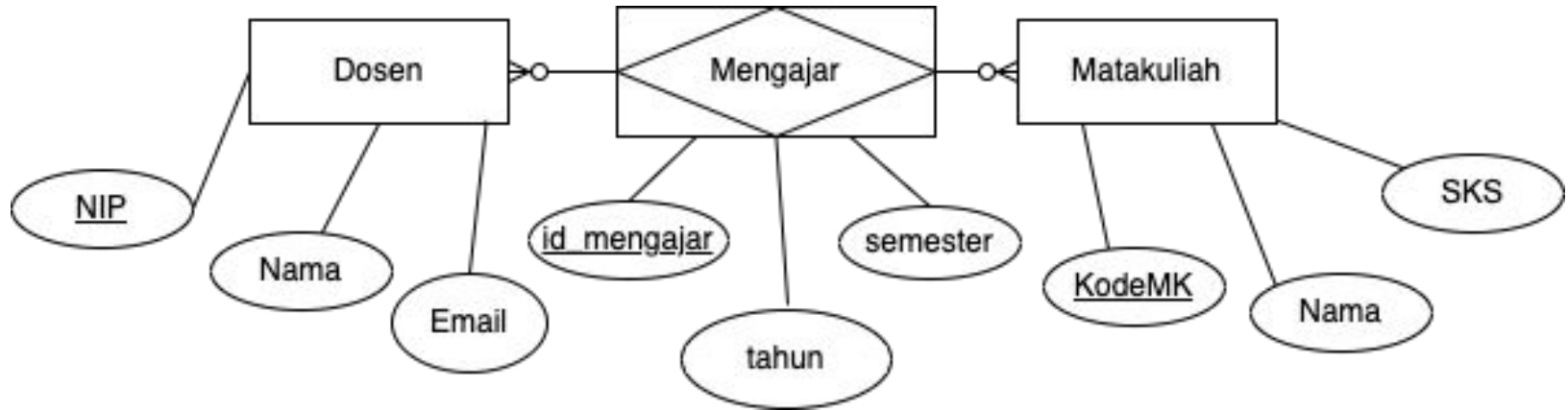
Simbol ERD (Entity Relationship Diagram)



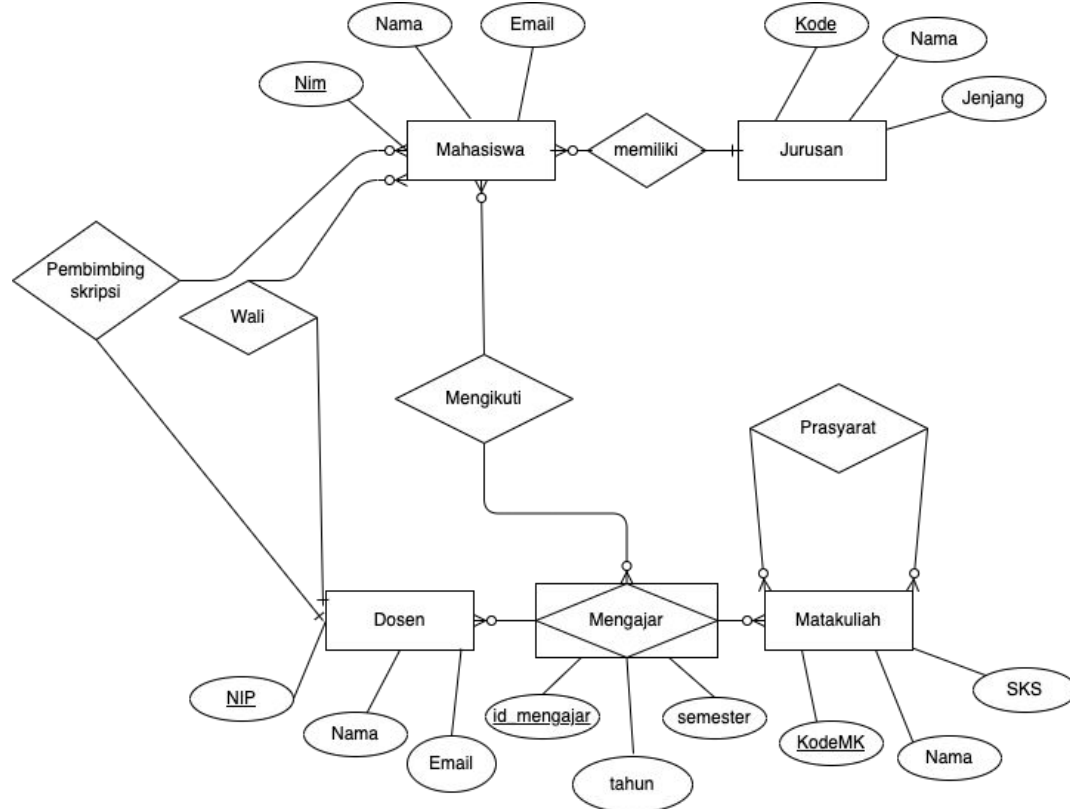
Simbol ERD (Entity Relationship Diagram)



Simbol ERD (Entity Relationship Diagram)



Simbol ERD (Entity Relationship Diagram)



PostgreSQL CLI VS GUI

```

> psql --host=localhost --port=5432 --dbname= --username=postgres --password
Password:
psql (17.6)
Type "help" for help.

postgres=# exit;
> psql --host=localhost --port=5432 --dbname=penjualan_db --username=postgres --password
Password:
psql (17.6)
Type "help" for help.

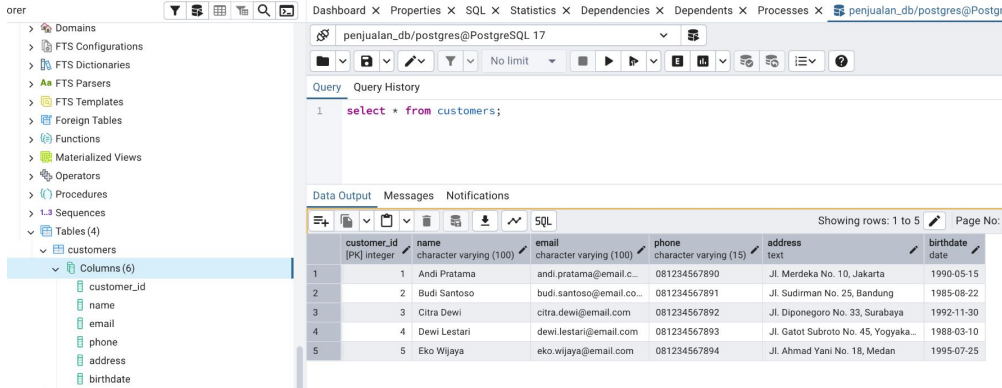
penjualan_db=# select * from customers;
 customer_id |   name   |      email       |   phone   |      address      | birthdate
-----+-----+-----+-----+-----+-----
      1 | Andi Pratama | andi.pratama@email.com | 081234567890 | Jl. Merdeka No. 10, Jakarta | 1990-05-15
      2 | Budi Santoso | budi.santoso@email.com | 081234567891 | Jl. Sudirman No. 25, Bandung | 1985-08-22
      3 | Citra Dewi | citra.dewi@email.com | 081234567892 | Jl. Diponegoro No. 33, Surabaya | 1992-11-30
      4 | Dewi Lestari | dewi.lestari@email.com | 081234567893 | Jl. Gatot Subroto No. 45, Yogyakarta | 1988-03-10
      5 | Eko Wijaya | eko.wijaya@email.com | 081234567894 | Jl. Ahmad Yani No. 18, Medan | 1995-07-25
(5 rows)

penjualan_db=# SELECT version();
              version
-----
PostgreSQL 17.6 on x86_64-apple-darwin23.6.0, compiled by Apple clang version 16.0.0 (clang-1600.0.26.6), 64-bit
(1 row)

```

PostgreSQL CLI VS GUI

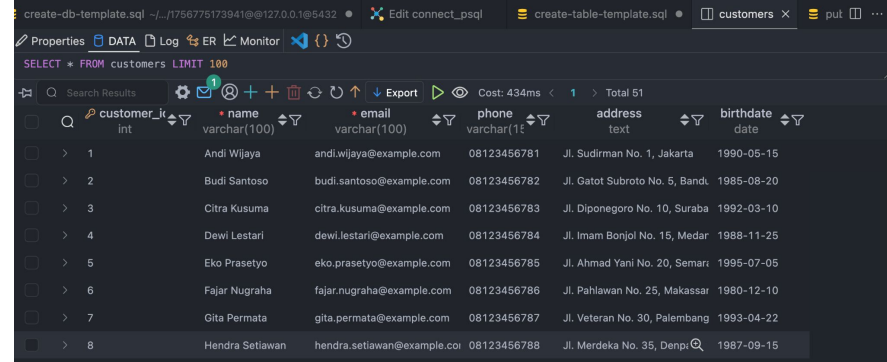
pgAdmin 4



pgAdmin 4 interface showing the 'customers' table in the 'penjualan_db' database. The table structure and data are as follows:

customer_id	name	email	phone	address	birthdate
1	Andi Pratama	andi.pratama@email.co...	081234567890	Jl. Merdeka No. 10, Jakarta	1990-05-15
2	Budi Santoso	budi.santoso@email.co...	081234567891	Jl. Sudirman No. 25, Bandung	1985-08-22
3	Citra Dewi	citra.dewi@email.com	081234567892	Jl. Diponegoro No. 33, Surabaya	1992-11-30
4	Dewi Lestari	dewi.lestari@email.com	081234567893	Jl. Gatot Subroto No. 45, Yogyakarta	1988-03-10
5	Eko Wijaya	eko.wijaya@email.com	081234567894	Jl. Ahmad Yani No. 18, Medan	1995-07-25

Database Client



Database Client interface showing the 'customers' table. The table structure and data are as follows:

customer_id	name	email	phone	address	birthdate
1	Andi Wijaya	andi.wijaya@example.com	08123456781	Jl. Sudirman No. 1, Jakarta	1990-05-15
2	Budi Santoso	budi.santoso@example.com	08123456782	Jl. Gatot Subroto No. 5, Band.	1985-08-20
3	Citra Kusuma	citra.kusuma@example.com	08123456783	Jl. Diponegoro No. 10, Surabaya	1992-03-10
4	Dewi Lestari	dewi.lestari@example.com	08123456784	Jl. Imam Bonjol No. 15, Medan	1988-11-25
5	Eko Prasetyo	eko.prasetyo@example.com	08123456785	Jl. Ahmad Yani No. 20, Semarang	1995-07-05
6	Fajar Nugraha	fajar.nugraha@example.com	08123456786	Jl. Pahlawan No. 25, Makassar	1980-12-10
7	Gita Permata	gita.permata@example.com	08123456787	Jl. Veteran No. 30, Palembang	1993-04-22
8	Hendra Setiawan	hendra.setiawan@example.com	08123456788	Jl. Merdeka No. 35, Denpasar	1987-09-15

Apa Itu Tipe Data dalam Database

Tipe data adalah jenis nilai yang bisa disimpan dalam sebuah kolom di tabel database.

Bayangkan kamu membuat formulir:

- Kolom "Nama" → harus berisi teks (misal: "Andi")
- Kolom "Usia" → harus berisi angka (misal: 25)
- Kolom "Tanggal Lahir" → harus berisi tanggal (misal: 1999-05-15)

THANKS!

arif.wicaksono@lecturer.itk.ac.id
+62 852 1308 1309