

***JOBSHEET***

# **BASIS DATA LANJUT**



**Jurusan Teknologi Informasi**

**POLITEKNIK NEGERI MALANG**

**TAHUN AJARAN 2025/2026**

## **PERTEMUAN 8**

### **Full-Text Search (FTS) dan JSONB**

Team Teaching Basis Data Lanjut:

- Candra Bella Vista, S.Kom., MT.
- Moch Zawaruddin Abdullah, S.ST., M.Kom.
- Yan Watequlis Syaifudin, ST., MMT., PhD.
- Yoppy Yunhasnawa, S.ST., M.Sc.

Mata Kuliah : Basis Data Lanjut  
Program Studi : D4 – Teknik Informatika / D4 – Sistem Informasi Bisnis  
Semester : 3 (tiga)  
Pertemuan ke- : 8

## **MATERI 8: FULL-TEXT SEARCH DAN JSONB**

### **Mata Kuliah: Basisdata Lanjut**

#### **Tujuan Pembelajaran**

Setelah menyelesaikan materi ini, mahasiswa diharapkan mampu:

1. Memahami konsep dasar pencarian teks penuh (*Full-Text Search*) dan penyimpanan data semi-terstruktur dengan JSONB di PostgreSQL.
2. Menjelaskan perbedaan antara pencarian teks biasa menggunakan LIKE dengan mekanisme FTS menggunakan `to_tsvector()` dan `to_tsquery()`.
3. Mengimplementasikan Full-Text Search untuk melakukan pencarian kata, frasa, maupun kombinasi logika pencarian.
4. Membuat dan mengoptimalkan indeks GIN untuk meningkatkan kinerja pencarian teks.
5. Menyimpan dan mengelola data fleksibel menggunakan tipe data JSONB.
6. Menggunakan berbagai operator JSONB (`->`, `->>`, `#>`, `@>`, `?`) untuk membaca, memfilter, dan memanipulasi data.
7. Menganalisis serta mengekstrak informasi dari data JSONB menggunakan *Common Table Expression* (CTE).

#### **Konsep Dasar**

Materi ini membahas dua fitur penting PostgreSQL yang memperluas kemampuannya melampaui database relasional konvensional, yaitu **Full-Text Search (FTS)** dan **JSONB (Binary JSON)**.

##### **1. Full-Text Search (FTS):**

FTS adalah mekanisme pencarian teks yang memahami struktur bahasa alami, bukan hanya pencocokan literal seperti pada LIKE.

PostgreSQL mengubah teks menjadi *vector of lexemes* menggunakan fungsi `to_tsvector()`, dan kueri menjadi *tsquery* melalui `to_tsquery()`. Operator `@@` digunakan untuk mencocokkan keduanya, sementara `ts_rank()` dan `ts_headline()` membantu memberikan peringkat dan menyoroti hasil pencarian.

##### **2. Indeks GIN:**

Indeks GIN (*Generalized Inverted Index*) digunakan untuk mempercepat pencarian teks besar. Dengan GIN, PostgreSQL tidak perlu memindai seluruh tabel saat mencari kata kunci tertentu, sehingga FTS dapat berjalan hingga puluhan kali lebih cepat dibanding pencarian biasa.

### 3. JSON dan JSONB:

JSON adalah format teks standar untuk menyimpan data semi-terstruktur, sedangkan JSONB adalah versi *binary* yang sudah diurai dan dapat diindeks. JSONB lebih efisien dalam hal kecepatan akses dan dukungan operator.

### 4. Operator JSONB:

PostgreSQL menyediakan beragam operator seperti `->` untuk mengambil objek JSON, `->>` untuk mengambil nilai teks, `#>` untuk navigasi jalur bersarang, serta `@>` dan `?>` untuk melakukan pencarian berdasarkan isi JSON.

### 5. CTE (Common Table Expression):

CTE digunakan untuk memecah data JSONB yang kompleks menjadi bentuk tabel sementara agar lebih mudah dianalisis. Misalnya, elemen array JSON dapat diubah menjadi baris tersendiri untuk dilakukan agregasi atau perhitungan statistik.

## Manfaat

Menguasai konsep Full-Text Search dan JSONB memberikan kemampuan strategis dalam pengelolaan serta analisis data modern, antara lain:

1. **Pencarian Data yang Cerdas:** Mahasiswa dapat membangun fitur pencarian teks yang lebih akurat dan cepat, mirip mesin pencari profesional, dengan dukungan bahasa alami.
2. **Performa Query Lebih Tinggi:** Indeks GIN mempercepat pencarian ribuan teks dalam waktu sangat singkat tanpa memindai seluruh tabel.
3. **Fleksibilitas Struktur Data:** JSONB memungkinkan penyimpanan data semi-terstruktur seperti spesifikasi produk, atribut dinamis, atau metadata penelitian tanpa perlu mengubah skema tabel.
4. **Kemampuan Analisis Terintegrasi:** Penggunaan CTE dan operator JSONB memungkinkan analisis data terstruktur dan tidak terstruktur dalam satu query SQL.
5. **Efisiensi Pengembangan Aplikasi:** Dengan dukungan FTS dan JSONB, PostgreSQL dapat menggantikan kebutuhan sistem NoSQL terpisah, memudahkan integrasi backend dan mengurangi kompleksitas arsitektur.

6. **Kesiapan untuk Big Data dan AI:** Kemampuan menyimpan teks panjang dan data fleksibel menjadikan mahasiswa siap menghadapi tantangan pengolahan data besar, text mining, dan machine learning berbasis SQL.

# Petunjuk Praktikum: Full-Text Search (FTS) dan JSONB

Penjelasan Topik-1 hingga Topik-4 terdapat pada slide materi presentasi.

## Topik-5: Latihan Dasar-dasar Index & Penggunaannya

### Tujuan

Mahasiswa memahami cara kerja Full-Text Search (FTS) di PostgreSQL menggunakan fungsi `to_tsvector()` dan `to_tsquery()`, serta cara mengoptimalkannya dengan indeks GIN.

---

### 1. Persiapan Awal

Buatlah sebuah database kosong baru. Pada database tersebut, buka Query Tool, lalu copy-paste SQL dari file bernama `latihan_fulltext_jsonb.sql`. Mintalah file ini ke pengampu.

Jika Anda berhasil mengeksekusi file SQL tersebut, maka di dalam database Ada akan terisi tabel `proposal` dengan kolom `judul_proposal` dan `deskripsi`. Data ini akan digunakan sebagai sumber teks untuk pencarian.

Selanjutnya, pada database Anda tersebut, tambahkan kolom baru di tabel `proposal` untuk menyimpan vektor pencarian:

```
ALTER TABLE proposal ADD COLUMN search_vector tsvector;
```

Isi kolom `search_vector` dengan teks gabungan dari `judul_proposal` dan `deskripsi`:

```
UPDATE proposal
```

```
SET search_vector =
```

```
    setweight(to_tsvector('indonesian', coalesce(judul_proposal, '')), 'A') ||
```

```
    setweight(to_tsvector('indonesian', coalesce(deskripsi, '')), 'D');
```

Buat indeks GIN agar pencarian teks lebih cepat:

```
CREATE INDEX idx_proposal_search
```

```
ON proposal
```

```
USING gin(search_vector);
```

---

### 2. Uji Pencarian Dasar

Jalankan query untuk mencari proposal yang mengandung kata "*keamanan*" dan "*jaringan*":

```
SELECT
```

```
    proposal_id,
```

```
    judul_proposal,
```

```
    ts_rank(search_vector, to_tsquery('indonesian', 'jaringan & keamanan')) AS rank
```

```
FROM proposal
WHERE search_vector @@ to_tsquery('indonesian', 'jaringan & keamanan')
ORDER BY rank DESC;
Catat hasil pencarian dan nilai rank yang muncul.
```

---

### 3. Pencarian Lanjutan dengan Operator

Uji perbedaan hasil pencarian berikut:

```
-- AND
SELECT judul_proposal FROM proposal
WHERE search_vector @@ to_tsquery('indonesian', 'sistem & rekomendasi');
```

```
-- OR
SELECT judul_proposal FROM proposal
WHERE search_vector @@ to_tsquery('indonesian', 'chatbot | rekomendasi');
```

```
-- Prefix
SELECT judul_proposal FROM proposal
WHERE search_vector @@ to_tsquery('indonesian', 'rekomendasi:*');
Catatan: operator :* digunakan untuk mencari awalan kata tertentu.
```

---

### 4. Ranking dan Highlight Hasil

Tambahkan kolom rank dan tampilkan highlight kata kunci:

```
SELECT
  proposal_id,
  judul_proposal,
  ts_rank(search_vector, to_tsquery('indonesian', 'jaringan & keamanan')) AS rank,
  ts_headline(
    'indonesian',
    deskripsi,
    to_tsquery('indonesian', 'jaringan & keamanan'),
    'StartSel=<b>,StopSel=</b>,MinWords=5,MaxWords=15,MaxFragments=2'
  ) AS snippet
FROM proposal
WHERE search_vector @@ to_tsquery('indonesian', 'jaringan & keamanan')
ORDER BY rank DESC
LIMIT 5;
Perhatikan bagaimana PostgreSQL menyorot (highlight) kata yang cocok di kolom deskripsi.
```

---

## Refleksi

1. Apa keunggulan `to_tsvector()` dibanding `LIKE '%...%'` dalam pencarian teks?
  2. Mengapa indeks GIN penting untuk Full-Text Search?
  3. Coba ubah konfigurasi bahasa dari 'indonesian' menjadi 'english' dan amati perbedaannya.
- 

## Topik-6: Latihan Manipulasi dan Query JSONB

### Tujuan

Mahasiswa dapat melakukan penyimpanan, pembacaan, dan filter data berbentuk JSONB serta memahami operator dan fungsi JSONB di PostgreSQL.

---

### 1. Persiapan Awal

Gunakan tabel **inventaris\_lab** yang berisi kolom spesifikasi JSONB. Periksa isi datanya:

```
SELECT nama_barang, spesifikasi FROM inventaris_lab LIMIT 3;
```

Contoh data:

```
{
  "merek": "MikroTik",
  "model": "CCR1009",
  "fitur": ["BGP", "VPN", "Firewall"],
  "port": {"ethernet": 8, "sfp": 1},
  "status": "aktif"
}
```

---

### 2. Membaca dan Mengakses Data JSONB

*-- Mengambil merek alat*

```
SELECT nama_barang, spesifikasi ->> 'merek' AS merek
FROM inventaris_lab;
```

*-- Mengambil jumlah port ethernet*

```
SELECT nama_barang, spesifikasi #> '{port, ethernet}' AS ethernet_port
FROM inventaris_lab;
```

---

### 3. Filtering Berdasarkan Isi JSONB

```
-- Cari alat dengan fitur VPN
SELECT nama_barang, spesifikasi->'fitur' AS fitur
FROM inventaris_lab
WHERE spesifikasi @> '{"fitur": ["VPN"]}';
```

```
-- Cari alat yang memiliki GPU
SELECT nama_barang, spesifikasi
FROM inventaris_lab
WHERE spesifikasi ? 'gpu';
```

---

### 4. Mengubah Nilai di JSONB

Tambahkan atau ubah elemen baru pada JSONB:

```
UPDATE inventaris_lab
SET spesifikasi = jsonb_set(spesifikasi, '{status}', '\"maintenance\"')
WHERE nama_barang ILIKE '%Router MikroTik%';
Periksa kembali hasilnya:

SELECT nama_barang, spesifikasi ->> 'status' AS status FROM inventaris_lab;
```

---

### 5. Analisis Data JSONB dengan CTE

```
WITH fitur AS (
  SELECT
    nama_barang,
    jsonb_array_elements_text(spesifikasi->'fitur') AS fitur
  FROM inventaris_lab
  WHERE spesifikasi ? 'fitur'
)
SELECT fitur, COUNT(*) AS jumlah
FROM fitur
GROUP BY fitur
ORDER BY jumlah DESC;
Tujuan: menghitung jumlah kemunculan setiap fitur pada seluruh alat laboratorium.
```

---

### Refleksi

1. Apa kelebihan menggunakan JSONB dibanding membuat banyak kolom tabel?
2. Jelaskan perbedaan antara operator -> dan ->>.



3. Bagaimana cara mempercepat pencarian data JSONB di kolom besar?
- 

## Tugas

1. Cobalah untuk menjalankan semua perintah SQL pada dokumen petunjuk praktikum ini, lalu buatlah laporan yang berisi screenshot serta penjelasan hasilnya.
  2. Pada laporan Anda tersebut, cantumkan juga jawaban dari semua poin **Refleksi** pada setiap sub-topik yang ada.
  3. Kerjakan juga **Soal Latihan Praktikum 8** di bagian bawah dokumen ini.
- 

## Penutup

Setelah menyelesaikan praktikum ini, mahasiswa diharapkan:

- Mampu menerapkan pencarian teks dengan `to_tsvector()` dan `to_tsquery()`.
- Memahami konsep GIN Index untuk optimasi FTS.
- Dapat melakukan penyimpanan dan filter data semi-terstruktur menggunakan JSONB.
- Menguasai kombinasi SQL + JSONB untuk eksplorasi dan analisis data modern.

## Soal Latihan Praktikum 8

### Soal-1: Full-Text Search Dasar

- Pada tabel `proposal`, telah ditambahkan kolom `search_vector` untuk menyimpan hasil `to_tsvector()` dari kolom `judul_proposal` dan `deskripsi`.
- Tuliskan SQL untuk menampilkan **judul proposal** yang mengandung kata “keamanan” dan “jaringan” sekaligus, beserta nilai ranking relevansinya!

### Soal-2: Full-Text Search (FTS) Lanjutan

- Dosen ingin mencari semua **proposal** yang membahas “chatbot” atau “rekomendasi” menggunakan Full-Text Search, bukan LIKE.
- Tuliskan query-nya, lalu jelaskan perbedaan hasilnya jika dibandingkan dengan ILIKE `'%chatbot%'`.

### Soal-3: JSONB (Akses dan Filter Data)

- Gunakan tabel `inventaris_lab` yang memiliki kolom spesifikasi JSONB.
- Tuliskan SQL untuk menampilkan **nama\_barang** dan **merek** dari semua alat yang memiliki fitur “VPN” di dalam JSONB.

### Soal-4: JSONB (Analisis Data Bersarang dengan CTE)

- Mahasiswa diminta menghitung **berapa kali setiap fitur muncul** di seluruh alat laboratorium yang memiliki kolom fitur pada JSONB.
- Tuliskan query SQL-nya menggunakan *Common Table Expression (CTE)*.

### Query Mahasiswa:

--

### Hasil Eksekusi:

--

## **Refleksi Pembelajaran**

**Nama Mahasiswa:** \_\_\_\_\_

**NIM:** \_\_\_\_\_

### **Hal yang sudah dipahami:**

- Mampu menerapkan pencarian teks dengan `to_tsvector()` dan `to_tsquery()`.
- Memahami konsep GIN Index untuk optimasi FTS.
- Dapat melakukan penyimpanan dan filter data semi-terstruktur menggunakan JSONB.
- Menguasai kombinasi SQL + JSONB untuk eksplorasi dan analisis data modern.

### **Kesulitan yang dialami:**