**LAPORAN TUGAS KECIL**

**IF3140 MANAJEMEN BASIS DATA**

**DATABASE PERFORMANCE TUNING POSTGRESQL**



Oleh

Wisnu 13513029

Alson Cahyadi 13514035

Ari Pratama Zhorifiandi 13514039

Geraldi Dzakwan 13514065

Ramos Janoah H. 13514089

Jovian Christianto 13514101

Alvin Junianto Lan 13514105

**PROGRAM STUDI INFORMATIKA**

**INSTITUT TEKNOLOGI BANDUNG**

**BANDUNG**

**2016**

# Dasar Teori

# Pembahasan Permasalahan dan Pembagian Kerja

Query Awal

1. **Tuning 1 : Collapsing Tables**

**Proses:**

Collapsing Tables dilakukan pada table genre dan genrefilm, dengan menyatukan nama genre ke table genrefilm.

**Alasan pemilihan cara:**

Pada query awal, terjadi beberapa kali natural join antara table genre dan genrefilm. Dengan melakukan collapsing table, akan meniadakan natural join antara table genre dan genrefilm.

**Hasil pemrosesan yang didapatkan:**

**Skema hasil perubahan:**

**Query Perubahan menjadi basis data baru:**

1. ALTER TABLE genrefilm

ADD COLUMN namagenre character varying(20);

1. UPDATE genrefilm

SET namagenre = genre.namagenre

FROM genre

WHERE genrefilm.genreid = genre.genreid;

1. ALTER TABLE genrefilm DROP CONSTRAINT genrefilm\_ibfk\_1;
2. DROP TABLE genre;
3. **Tuning 2: Adding Redundant Column**

**Proses:**

Adding Redundant Column menambahkan beberapa kolom atribut dari relasi Sutradara pada Relasi film.

**Alasan pemilihan cara:**

Sama seperti Collapsing tables, cara ini cukup efektif untuk mempercepat

query karena mengurangi natural join yang terjadi pada query.

- Menambahkan kolom NamaDepanStutradara dan NamaBelakangSutradara pada

tabel film agar mengurangi natural join dengan sutradara.

- Menambahkan kolom NamaDepanAktor dan NamaBelakangAktor pada tabel aktorfilm. Tujuannya adalah untuk mengurangi natural join yang terjadi.

**Hasil pemrosesan yang didapatkan:**

**Skema hasil perubahan:**

**Query Perubahan menjadi basis data baru:**

1. ALTER TABLE film

ADD COLUMN NamaDepanSutradara character varying(255),

ADD COLUMN NamaBelakangSutradara character varying(255);

2. Update film

SET NamaDepanSutradara = sutradara.NamaDepanSutradara,

NamaBelakangSutradara = sutradara.NamaBelakangSutradara

FROM sutradara

where film.sutradaraID = sutradara.sutradaraID;

3. ALTER TABLE aktorfilm

ADD COLUMN NamaDepanAktor character varying(255),

ADD COLUMN NamaBelakangAktor character varying(255);

4. Update aktorfilm

SET NamaDepanAktor = aktor.NamaDepanAktor,

NamaBelakangAktor = aktor.NamaBelakangAktor

FROM aktor

where aktor.AktorID = aktorfilm.AktorID;

1. **Tuning 3: Vertical Splitting**

**Proses:**

Memisahkan relasi film dari

film(FilmID, SutradaraID, NamaDepanSutradara, NamaBelakangSutradara, Judul, TanggalRilis, Rating, LamaFilm)

**menjadi**

film\_s(FilmID, SutradaraID, NamaDepanSutradara, NamaBelakangSutradara)

dan film\_data(FilmID, Judul, TanggalRilis, Rating, LamaFilm)

**Alasan pemilihan cara**

Cara ini dipilih karena dapat mempercepat query dengan cara mengurangi jumlah kolom yang diakses pada tabel film.

**Hasil pemrosesan yang didapatkan:**

**Skema hasil perubahan:**

**Query Perubahan menjadi basis data baru**

(1) ALTER TABLE film RENAME To film\_s;

(2) CREATE TABLE film\_data as TABLE film\_s;

(3) ALTER TABLE film\_s

DROP COLUMN judul,

DROP tanggalrilis,

DROP rating,

DROP lamafilm;

(4) ALTER TABLE film\_data

DROP COLUMN sutradaraid,

DROP COLUMN namadepansutradara,

DROP COLUMN namabelakangsutradara;

1. **Tuning 4: Adding Derived Attributes**

**Proses**

Menambahkan atribut turunan JumlahPenonton dari hasil perhitungan jumlah penonton

pada tabel mbd.film

**Alasan pemilihan cara**

Agar perhitungan untuk jumlah penonton langsung ada pada tabel dan tidak perlu lagi dihitung dari tabel penonton.

**Query perubahan untuk basis data baru**

1. ALTER TABLE film\_s

ADD COLUMN count\_penonton int;

1. UPDATE film

SET count\_penonton = T1.JumlahPenonton

FROM (SELECT filmid, count(penontonid) as JumlahPenonton

FROM penontonfilm group by filmid) as T1

WHERE film.filmid = T1.filmid;

# Kesimpulan dan Saran

# Referensi