

# Laporan Pengujian Performa Query

Misi 2 kelompok 6. LPMPP

## A. Skenario Pengujian

Pengujian dilakukan untuk membandingkan efisiensi query pada Data Warehouse setelah penerapan strategi Indexing dan Partitioning. Pengujian menggunakan parameter SET STATISTICS IO dan SET STATISTICS TIME.

## B. Hasil Pengujian 1: Query Agregasi dengan Index

The screenshot shows the SQL Server Management Studio interface. The top window displays the SQL script being executed:

```
-- 1. Aktifkan Mode Pencatatan Waktu & I/O  
SET STATISTICS TIME ON; -- Buat list durasi (ms)  
SET STATISTICS IO ON; -- Buat list seberapa berat dia baca disk  
GO  
  
--PRINT '--- TEST 1: QUERY AGREGASI JOIN (Menguji Index) ---';  
-- Query ini memaksa database pakai Index di Foreign Key (StatusKey & JenisKlKey)  
-- Harusnya cepet banget karena udah di-index di Step 4.  
  
SELECT  
    j.Nama_JenisKl,  
    s.Nama_Status,  
    SUM(f.JumlahPengajuan) AS Total_Pengajuan,  
    SUM(f.BiayaPengajuan) AS Total_Biaya  
FROM dbo.Fact_Pengajuan f  
JOIN dbo.Dim_JenisKl j ON f.JenisKlKey = j.JenisKlKey  
JOIN dbo.Dim_Status s ON f.StatusKey = s.StatusKey  
GROUP BY j.Nama_JenisKl, s.Nama_Status  
ORDER BY Total_Pengajuan DESC;  
GO
```

The bottom window shows the execution results and messages. The results grid contains the following data:

	Nama_JenisKl	Nama_Status	Total_Pengajuan	Total_Biaya
1	Paten	Diberi (Granted)	241	312500000.00
2	Desain Industri	Diberi (Granted)	233	268000000.00
3	Hak Cipta	Ditolak (Rejected)	232	295000000.00
4	Desain Industri	Publikasi	231	269500000.00
5	Paten Sederhana	Publikasi	228	276000000.00
6	Merek	Diberi (Granted)	228	287500000.00
7	Paten	Pemeriksaan Formalitas	227	281000000.00
8	Merek	Pemeriksaan Formalitas	225	273000000.00
9	Rahasia Dagang	Ditolak (Rejected)	225	285000000.00
10	Paten	Publikasi	223	291500000.00
11	Desain Industri	Diajukan (Submitted)	222	277500000.00
12	Rahasia Dagang	Pemeriksaan Formalitas	222	286000000.00
13	Rahasia Dagang	Pemeriksaan Substantif	219	267500000.00
14	Merek	Diajukan (Submitted)	218	275500000.00
15	Hak Cipta	Pemeriksaan Substantif	218	273500000.00

**Analisis:**

Query melakukan agregasi data (SUM) dengan JOIN ke tabel dimensi. Hasil menunjukkan waktu eksekusi total 19 ms dengan CPU time 15 ms. Penggunaan *Non-Clustered Index* pada Foreign Key (JenisKIKKey, StatusKey) terbukti sangat efektif, ditandai dengan jumlah *Logical Reads* yang sangat rendah (hanya 2 reads pada tabel dimensi).

## C. Hasil Pengujian 2: Query Filter Partisi (Partition Pruning)

```
SQLQuery1.sql - C:\CAELAN\kn409 (57)* ↴ X
SET STATISTICS IO ON; -- Buat liat seberapa berat dia baca disk
GO

PRINT '---- TEST 2: QUERY FILTER PARTISI (Menguji Partition Pruning) ----';

SELECT
    DateKey_Daftar,
    COUNT(PengajuanKey) AS Jumlah_Harian
FROM dbo.Fact_PengajuanT
WHERE DateKey_Daftar BETWEEN 20240101 AND 20241231 -- Filter 1 Tahun
GROUP BY DateKey_Daftar
ORDER BY DateKey_Daftar;
GO

-- Matikan Mode Statistik
SET STATISTICS TIME OFF;
SET STATISTICS IO OFF;
GO

91 % ▾
Results Messages


|    | DateKey_Daftar | Jumlah_Harian |
|----|----------------|---------------|
| 1  | 20240101       | 4             |
| 2  | 20240102       | 2             |
| 3  | 20240103       | 15            |
| 4  | 20240104       | 5             |
| 5  | 20240105       | 6             |
| 6  | 20240106       | 2             |
| 7  | 20240107       | 11            |
| 8  | 20240108       | 3             |
| 9  | 20240109       | 3             |
| 10 | 20240110       | 5             |
| 11 | 20240111       | 4             |
| 12 | 20240112       | 5             |
| 13 | 20240113       | 4             |
| 14 | 20240114       | 6             |
| 15 | 20240115       | 4             |


```

```
SQLQuery1sql - C:\ELAN\kn409 (57)* + X
SET STATISTICS IO ON; -- Buat list seberapa berat dia baca disk
GO
--PRINT '*** TEST 2: QUERY FILTER PARTISI (Menguji Partition Pruning) ***';

--SELECT
--    Datekey_Daftar,
--    COUNT(PengajuanKey) AS Jumlah_Harian
--FROM dbo.Fact_PengajuanK1
--WHERE Datekey_Daftar BETWEEN 20240101 AND 20241231 -- Filter 1 Tahun
--GROUP BY Datekey_Daftar
--ORDER BY Datekey_Daftar;
--GO

-- Matikan Mode Statistik
SET STATISTICS TIME OFF;
SET STATISTICS IO OFF;
GO

91 %
```

Results of Messages

```
SQL Server parse and compile time: 
CPU time = 0 ms, elapsed time = 0 ms.

SQL Server Execution Times: 
CPU time = 0 ms, elapsed time = 0 ms.
SQL Server waits and queue times: 
CPU time = 0 ms, elapsed time = 0 ms.

SQL Server Execution Times: 
CPU time = 0 ms, elapsed time = 0 ms.

SQL Server Execution Times: 
CPU time = 0 ms, elapsed time = 0 ms.
SQL Server parse and compile time: 
CPU time = 0 ms, elapsed time = 0 ms.

*** TEST 2: QUERY FILTER PARTISI (Menguji Partition Pruning) ***

SQL Server Execution Times: 
CPU time = 0 ms, elapsed time = 0 ms.

(84 rows affected)
Table 'Fact_PengajuanK1'. Scan count 1, logical reads 9, physical reads 1, page server reads 0, read-ahead reads 7, page server read-ahead reads 0, lob logical reads 0, lob physical reads 0, lob page server reads 0, lob page read-ahead reads 0.
```

## **Analisis:**

Query melakukan filter data khusus tahun 2024. Strategi partisi terbukti berhasil melakukan *Partition Pruning*, di mana SQL Server hanya membaca 1 partisi (Scan count = 1) dan mengabaikan partisi tahun lainnya. Hal ini menurunkan *Logical Reads* secara signifikan menjadi 9 page reads dengan waktu eksekusi 0 ms (instant).

#### D. Query Optimization Recommendations (Rekomendasi)

Rekomendasi Optimasi Lanjutan:

1. Maintenance Statistics Rutin:

Disarankan untuk menjadwalkan *Update Statistics* secara berkala (mingguan) agar Query Optimizer selalu memiliki metadata terbaru tentang distribusi data, memastikan rencana eksekusi (Execution Plan) tetap optimal seiring bertambahnya data.

2. Index Maintenance (Rebuild/Reorganize):

Seiring berjalannya waktu, proses ETL (Insert/Delete) dapat menyebabkan *fragmentasi index*. Disarankan membuat *Maintenance Plan* untuk melakukan *Rebuild Index* jika fragmentasi > 30% atau *Reorganize* jika fragmentasi antara 5-30%.

3. Penerapan Columnstore Index (Opsional):

Jika volume data tabel Fakta (Fact\_PengajuanKI) mencapai jutaan baris di masa depan, disarankan untuk mengganti *Row-Store Index* dengan Clustered Columnstore Index (CCI). CCI dapat mengkompresi data hingga 10x dan mempercepat query agregasi analitik secara signifikan.

4. Manajemen Partisi Tahunan:

Karena tabel menggunakan strategi partisi tahunan (PF\_TahunPengajuan), administrator database harus menyiapkan *Filegroup* dan *Scheme* baru sebelum tahun berganti (contoh: menyiapkan partisi 2026 sebelum Januari 2026) menggunakan fungsi SPLIT RANGE.