
PERBANDINGAN EFISIENSI WAKTU PROSES PENGAKSESAN DATA ANTARA QUERY BERBENTUK JOIN DENGAN SUBSELECT

¹R Bagus Bambang Sumantri[✉], ²Guntur Subari, ³Fajar Mahardika, ¹Hadi Jayusman

¹Sistem Informasi, Fakultas Sains dan Teknologi, Universitas Harapan Bangsa, Banyumas, Indonesia

²Program Studi Ilmu Komputer, Universitas Ahmad Dahlan, Yogyakarta, Indonesia

³Teknologi Komputer, Institut Teknologi Sains NU Pekalongan, Indonesia

Email: bagusbambang@uhb.ac.id

DOI: <https://doi.org/10.46880/jmika.Vol7No1.pp25-33>

ABSTRACT

Queries in a database application system, have a great influence on the success of application development. Frequently used query forms include subselect and join. Choosing the right query form will affect the performance of the database application. Database applications are sometimes slow to respond because of slowness in their databases. The selection of the form of the query to be used is the first step of forming a query, because the resulting query will affect access from the database. The purpose of this study is to determine the form of SQL with a faster data access time, whether join or subselect. The subject of this study is queries made in the form of joins and subselects. The methods that will be used in this study are literature and experimental methods. The result of this study is a comparison table of access time between join and subselect queries. From the results of the query comparison, it can be concluded that the join form query is a faster SQL form compared to subselect. A faster query is determined from testing done by creating a simple application using both forms of the query.

Keyword: Database, SQL, Query, Join, Subselect, Query Optimization.

ABSTRAK

Query dalam sebuah sistem aplikasi basis data, memiliki pengaruh besar terhadap keberhasilan pengembangan aplikasi. Bentuk query yang sering digunakan diantaranya adalah subselect dan join. Pemilihan bentuk query yang tepat akan mempengaruhi kinerja aplikasi basis data. Aplikasi basis data kadang-kadang lambat memberikan respon karena terjadi kelambatan pada basisdatanya. Pemilihan bentuk query yang akan digunakan merupakan langkah awal dari pembentukan suatu query, karena query yang dihasilkan akan mempengaruhi akses dari basis data. Tujuan dari penelitian ini adalah menentukan bentuk SQL dengan waktu pengaksesan datanya lebih cepat, apakah join atau subselect. Subjek dari penelitian ini adalah query-query yang dibuat dalam bentuk join dan subselect. Metode yang akan digunakan dalam penelitian ini adalah metode kepustakaan dan eksperimen. Hasil dari penelitian ini berupa sebuah tabel perbandingan waktu akses antara query join dengan subselect. Dari hasil perbandingan query tersebut, dapat disimpulkan bahwa query bentuk join adalah bentuk SQL yang lebih cepat dibandingkan dengan subselect. Query yang lebih cepat ditentukan dari pengujian yang dilakukan dengan membuat sebuah aplikasi sederhana menggunakan kedua bentuk query tersebut.

Kata Kunci: Basis Data, SQL, Query, Join, subselect, Optimasi Query.

PENDAHULUAN

Data merupakan catatan fakta-fakta dalam kehidupan sehari-hari yang dapat menggambarkan suatu peristiwa tertentu yang sebenarnya (Lubis, 2016), data dapat berupa angka atau quantitative data yang menghasilkan informasi numerik dan juga dapat berupa kata atau qualitative data yang menghasilkan informasi deskriptif, kumpulan data-data dari berbagai peristiwa dalam kehidupan sehari-hari apabila dicari hubungannya antara satu dengan yang lain dapat menghasilkan suatu informasi penting yang dapat digunakan manusia ketika menjalani

kehidupannya sehari-hari untuk mengambil keputusan dalam berbagai persoalannya. Seiring dengan perkembangan zaman dimana manusia hidup dalam era teknologi saat ini banyak hal dapat dimudahkan dengan adanya teknologi komputerisasi, pengolahan data dapat dilakukan dengan lebih mudah dan cepat untuk menghasilkan informasi yang bermanfaat terutama dalam suatu kegiatan-kegiatan besar yang dilakukan kurun waktu yang lama dan berulang-ulang yang tentunya memiliki data yang besar pula untuk diolah. Dalam penyajian informasi saat ini aplikasi berbasis web yang semakin canggih dan

didukung dengan basis data yang baik mampu menjadikan dunia ini tidak terbatas ruang maupun waktu (Akbar, Syah Askar, 2012).

Komputer saat ini memainkan peran penting dalam semua lapisan masyarakat. Sebagian besar komputer ini digunakan untuk menyimpan dan mengambil data dalam jumlah besar secara efisien, seperti sistem yang disebut sistem basis data dan perangkat lunak yang mengontrol aliran data (input dan penghapusan data), sering disebut manajemen basis data. (DBMS) dan DBMS memungkinkan banyak pengguna untuk mengakses satu basis data pada saat yang bersamaan. Biasanya, antarmuka (utama) pertama ke DBMS adalah query tingkat tinggi atau DML (Bahasa Manipulasi Data) yang dapat digunakan dengan mudah. Contoh query tingkat lanjut adalah SQL (Structured Query Language).

Pernyataan-pernyataan dalam SQL dapat dihasilkan secara langsung oleh *user* dengan menggunakan perintah-perintah antar muka. Perintah-perintah dalam SQL biasa disebut dengan *query*. Untuk menghasilkan suatu *query* haruslah sesuai dengan kebutuhan dari *user*. *Query* dalam sebuah pengembangan sistem aplikasi sangat besar pengaruhnya untuk keberhasilan pengembangan aplikasi basis data tersebut. Keberhasilan pengembang aplikasi basis data yang besar sering kali menjadi terkesan kurang dihargai oleh pengguna apabila ternyata basis data tersebut relatif lambat memberikan data yang diperlukan. Pemilihan *query* mana yang akan digunakan merupakan langkah awal dari pembentukan *query* itu sendiri, karena *query* yang terbentuk akan mempengaruhi akses dari basis data. Salah satu fitur *query* yang paling berguna adalah kemampuannya untuk mengumpulkan dan memanipulasi data dari beberapa sumber. Tanpa adanya fitur ini, beberapa sumber harus menyimpan semua elemen data yang diperlukan untuk tiap aplikasi dalam satu tabel, bahkan mungkin dalam beberapa tabel tentu saja akan sangat menyusahakan untuk mengatur ulang, membuat ulang dan mengisi data berulang kali kedalam tabel-tabel dalam basis data dari sumber yang di miliki setiap kali pengguna membutuhkan *query* untuk mencari informasi yang baru. Dalam sistem penyimpanan data yang besar, penting untuk mengoptimalkan kerja *query* agar sistem dapat digunakan secara optimal dan waktu pemrosesan dapat diminimalkan. (Thangam & Peter, 2016). Optimasi *query* adalah proses menentukan strategi eksekusi yang optimal untuk pemrosesan *query* (Elmasri, R. and Navathe, 2011). Optimasi *query* menjadi salah satu cara yang dapat digunakan untuk melakukan sebuah proses pengolahan data dalam

mencari jalur tercepat Optimalisasi kueri adalah salah satu cara untuk menemukan cara tercepat untuk melakukan proses pemrosesan data pada database management system (DBMS) (Dewi & Rezeki, 2017) Optimalisasi basis data dapat dilakukan 1) sesuai struktur basis data yaitu dengan mengatur struktur tabel agar tidak terjadi redundansi yang dapat menghambat kinerja sistem, dan 2) dengan melakukan *query* tabel basis data untuk mengambil data. Teknik pengoptimalan *query* menyediakan beberapa desain *query* yang diuji untuk mengidentifikasi dan memilih *query* paling optimal yang meminimalkan penggunaan sumber daya data tanpa mengubah hasilnya (Muhammad, Zakariyau, & Ali, 2015).

Dengan statemen SQL memungkinkan untuk merancang tabel yang lebih kecil serta lebih spesifik, yang tentu saja akan lebih memudahkan dibanding mengatur tabel-tabel yang lebih besar. Contoh SQL untuk merancang tabel yang lebih spesifik adalah dengan menggunakan *Join* dan *Subselect* (Haryanto,2005). *Join* adalah mekanisme yang digunakan untuk menggabungkan beberapa tabel menggunakan pernyataan SQL. Menggunakan sintaks khusus, beberapa tabel dapat digabungkan sehingga satu hasil dapat dibuat sedangkan *Subselect* adalah *query* yang dilekatkan pada *query* lain. Baik *Join* maupun *Subselect* merupakan sintaks SQL yang dapat menggabungkan beberapa tabel. Namun yang menjadi pertanyaan adalah “antara *Join* dan *Subselect*, manakah yang lebih efisien dalam pemrosesan suatu permintaan jika diukur menggunakan waktu?”, dari pertanyaan tersebutlah latar belakang penelitian ini.

TINJAUAN PUSTAKA

Pada penelitian yang dilakukan, penulis mengacu pada studi literatur atau penelitian sebelumnya. Berikut ini kami uraikan penelitian terdahulu dengan judul Optimalisasi Query dalam Basis Data My SQL Menggunakan Index (Pamungkas, 2018) Sebuah studi yang dilakukan oleh Ridho Pamungkas menyelidiki optimasi *query* sistem informasi menggunakan indeks. Dalam database sistem informasi, indeks berguna untuk mempercepat pencarian jika indeks digunakan sesuai kebutuhan. Penggunaan indeks secara sewenang-wenang dalam database memperumit kinerja *query*. Metode yang dipilih dalam penelitian ini adalah metode eksperimen. Dengan metode eksperimen dimungkinkan untuk memanipulasi dan mengubah variabel serta mencari indeks yang optimal. Hasil penelitian menunjukkan bahwa indeks yang diusulkan dapat meningkatkan kinerja pencarian sejumlah besar baris data.

Kajian kedua adalah Analisis Kinerja Optimalisasi Query Cross Join, Natural Join dan Full Outer Join (Bismo Raharjo, Mantriwira, & Sumanto, 2018). Dalam penelitian Yohanes Aryo, Bismo Raharjo dkk menginvestigasi analisis performansi optimasi query pada sintak cross join, natural join dan full outer join. Optimalisasi survei memiliki 3 fase yaitu. tahap persiapan, tahap pengujian dan analisis hasil pengujian. Tahap persiapan dimulai dengan pengumpulan informasi dan desain database. Pada fase uji penelitian, tiga sintaks *queri* dioptimalkan. Algoritma optimisasi penelitian adalah tahap pengujian: Hash Join Query dan nested join query. Pengujian data dilakukan pada baris data 1, 2, 4, 8, 16, 32, 64, 128, 256, 512, 1024, 2048, 4096 dan 5000. Pada baris data 1, 4, 64, 128, 256, 512, 1024, 2048, 4096, dan 5000 menunjukkan respons time sintaks natural join lebih cepat dibanding sintaks *cross join* dan *full outer join*. Dan pada baris data 2, 8, 16, dan 5 32, sintaks cross join lebih unggul dibandingkan dari sintaks *natural join* dan full outer join.

Kajian ketiga adalah migrasi dan optimalisasi database program pembelajaran Web berbasis Sistem Informasi Manajemen Informasi Diploma III Universitas Muhammadiyah Metro (Sudarmaji, 2016). Studi ini termasuk studi kasus sistem informasi berbasis e-learning. Program Diploma III Universitas Muhammadiyah Metro Kota Metro perancangan baru sistem informasi perkuliahan online yang menggunakan database dan membuat interface perkuliahan antara dosen dan mahasiswa. Secara umum dapat penulis sampaikan bahwa beberapa sistem informasi yang digunakan saat ini kurang baik, karena sistem tersebut memiliki kekurangan yaitu sistem SIAKAD dan e-learning yang tidak terintegrasi sehingga dinilai kurang efektif. Dalam sistem perkuliahan Universitas Muhammadiyah Metro.

Definisi Data

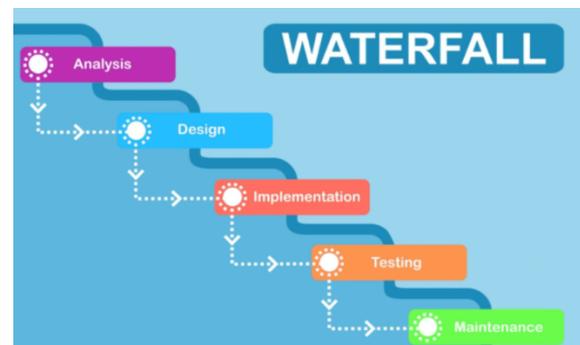
Data merupakan bahan dasar yang diolah sedemikian rupa sehingga menjadi informasi yang lebih bermanfaat dan berguna untuk mencapai tujuan yang telah ditetapkan bagi para pengguna informasi tersebut. Data adalah representasi objek, orang, peristiwa, aktivitas, konsep, dan objek penting lainnya dalam bahasa, matematika, dan simbol pengganti yang disepakati bersama. Data adalah bagian dari bahan mentah informasi yang dirumuskan sebagai kumpulan gelombang tidak acak yang menunjukkan kuantitas atau aktivitas dan hal-hal lain. (Raden Bagus Bambang Sumantri, Setiawan, & Sandi A, 2022)

SQL

SQL adalah bahasa khusus yang digunakan untuk mengakses dan mengelola RDBMS (Setiyadi, 2020). Aljabar relasional adalah sekumpulan operasi relasional dimana setiap operasi menggunakan satu atau lebih relasi untuk membuat relasi baru dan termasuk dalam kategori prosedur dan juga menyediakan sekumpulan operator untuk memanipulasi data (Setiyadi, 2020). Aljabar relasional terdiri dari kumpulan operasi relasional (relasional), dimana setiap operasi menciptakan relasi baru dari satu atau lebih relasi yang ada (Foster, 2016).

METODE PENELITIAN

Metode yang digunakan dalam penelitian ini adalah pendekatan alur hidup perangkat lunak secara sekuensial, sehingga model *waterfall* ini juga sering disebut sebagai model sekuensial linear (*Sequential Linear*) (Susilo & Kurniati, 2018). Berikut adalah tahapan pemodelan *Waterfall*:



Gambar 1. Model Waterfall

Sesuai dengan gambar di atas, tahapan dalam pemodelan *Waterfall* adalah sebagai berikut:

1. Analisis
Pada tahap ini, dilakukan pengidentifikasian permasalahan dan pembuatan perancangan solusi dari permasalahan tersebut (R Bagus Bambang Sumantri, Setiawan, & Triwibowo, 2022).
2. Desain
Pada tahap desain dibuat perancangan sistem berupa desain basis data menggunakan ERD, desain DFD, dan desain antarmuka.
3. Implementasi
Tahap implementasi dilakukan dengan merancang perangkat lunak yang diimplementasikan sebagai program atau unit program (Sasmito, 2017).
4. Testing
Dilakukan uji coba terhadap semua fungsi sistem dan pada tahap ini diperiksa apakah ada fungsi yang *error* atau tidak. Hasil uji coba seharusnya adalah

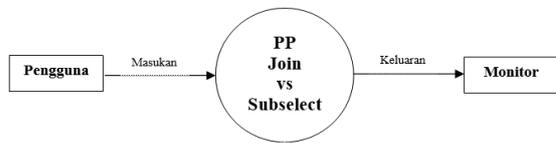
tidak ada fungsi yang *error* dan sudah sesuai dengan kebutuhan yang sudah diidentifikasi sebelumnya (Lumbanraja, 2018).

5. Maintenance

Tahap yang terakhir adalah tahap pemeliharaan, yaitu mengembangkan sistem sesuai dengan kebutuhan saat ini.

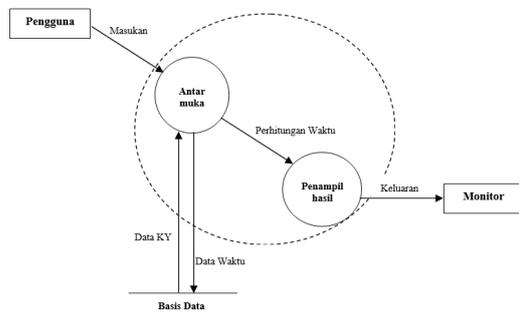
Perancangan Data Flow Diagram (DFD)

Perancangan DFD level 0 bertujuan untuk membuat skema aliran data dalam sistem dari dokumen dasar yang berada di luar lingkungan sistem, di proses dalam sistem hingga informasi yang dihasilkan oleh sistem untuk dipergunakan oleh lingkungan diluar sistem. Jadi perancangan diagram konteks memberikan gambaran interaksi antara sistem itu sendiri dengan lingkungan luar yang memberi pengaruh ke dalam sistem, termasuk di dalamnya proses *input* dan *output*, seperti terdapat dalam gambar 2 berikut ini.



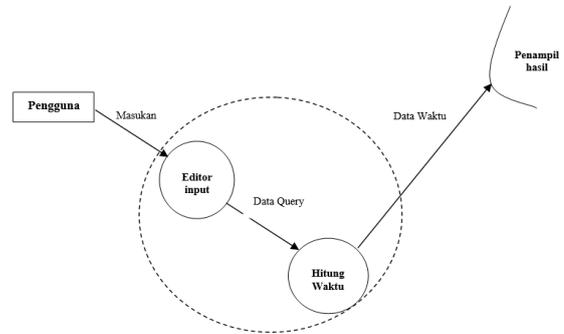
Gambar 2. DFD Level 0

DFD level 1 merupakan gambaran yang lebih detail dari diagram level 0. Level ini menjelaskan proses aliran data di dalam sistem dan interaksinya dengan entitas di luar sistem, termasuk proses penyimpanan data. Diagram aliran data level 1 ditunjukkan pada Gambar 3 seperti di bawah ini.



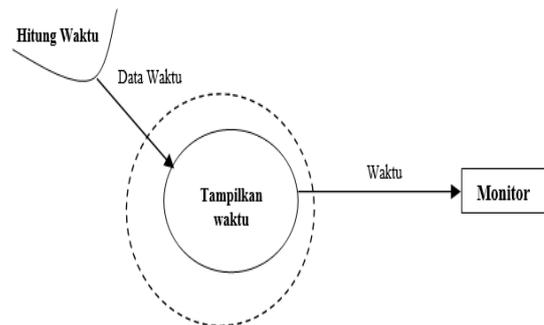
Gambar 3. DFD Level 1

DFD level 2 untuk proses 1 menggambarkan proses mulai dari memasukan *query* baik *join* maupun *subselect* sampai tahap proses untuk mendapatkan nilai rata-rata waktu proses *query*. Perancangan untuk proses 1 terlihat seperti gambar 4.



Gambar 4. DFD Level 2 untuk proses 1

DFD level 2 untuk proses 2 menggambarkan perancangan waktu yang akan ditampilkan setelah melalui proses eksekusi *query*. Berikut adalah gambar DFD level 2 untuk proses 2.



Gambar 5. DFD Level 2 untuk proses 2

HASIL & PEMBAHASAN

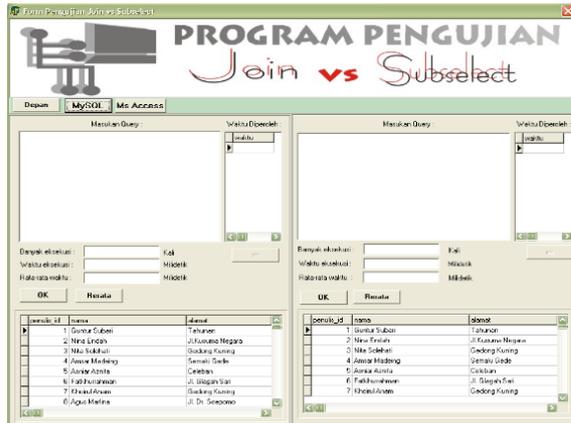
Sistem pengujian perbandingan *join* dan *subselect* ini terdiri atas satu menu pokok yang digunakan sebagai pembanding antara *query join* dengan *query* yang berbentuk *subselect*. Berikut ini adalah tampilan utama dari sistem pengujian ini.



Gambar 6. Tampilan Utama Sistem Pengujian

Dari tampilan utama sistem ini, pengguna dapat langsung menuju ke menu pokok yaitu menu pengujian. Pilih (klik) menu MySQL atau Ms Access maka akan

tampil halaman seperti yang tampak pada gambar 7. dibawah ini.



Gambar 7. Tampilan Menu Pengujian

Pada gambar diatas tampak dua bagian yang masing-masing akan digunakan untuk pengujian perbandingan *join* dan *subquery*. Untuk mengetahui proses penghitungan waktu eksekusi dari *query* yang dituliskan dapat dilihat dari potongan kode program dibawah ini.

```
[1]rerata := 0;
[2]ZTable_waktu1.First;
[3]for i := 1 to ZTable_waktu1.RecordCount do
[4] begin
[5] rerata :=
rerata+(ZTable_waktu1.fieldbyname('waktu').Value);
[7] hasil := rerata/i;
[8] Edit_rerata.Text := floatostr(hasil);
[9] ZTable_waktu1.Next;
[10]end;
[11]ZTable_waktu1.First;
```

Dari Potongan program diatas, nomor 1 sampai dengan nomor 3 adalah kode program untuk penghitungan waktu proses eksekusi *query*, dimana waktu (t1) diambil dari waktu sekarang dan *query* yang telah dimasukan kemudian dieksekusi, setelah mendapatkan data yang sesuai dengan *query* yang dituliskan, secara lansung waktu akan berhenti. Waktu (t2) terakhir yang tercatat akan dikurangkan dengan waktu mulai eksekusi (t1) dan akan ditampilkan. Potongan kode program nomor 4 akan menghitung berapa kali eksekusi dilakukan, potongan kode program nomor 5 digunakan untuk menampilkan nilai berapa banyaknya eksekusi dilakukan dan potongan program nomor 6 digunakan untuk menampilkan waktu eksekusi setiap kali eksekusi. Potongan program diatas digunakan untuk menghitung rata-rata waktu eksekusi dan berlaku untuk penghitungan waktu eksekusi dan rata-rata waktu eksekusi baik *join* maupun *subselect*.

Pengujian

Pada pengujian perbandingan query ini menggunakan dua DBMS (*Data Base Management System*) yaitu MySQL 8.0.31. dan Ms Access 2019, dengan menggunakan *query* yang sama dan data yang sama pula yaitu sebanyak 10000 data (*record*). Berikut adalah pernyataan-pernyataan yang digunakan untuk memperoleh *query* baik *query* berbentuk *join* maupun berbentuk *subselect*.

Tampilkan data penulis yang memiliki karya tulis tertentu. Bentuk *query*nya adalah:

Subselect :

```
[1] t1:=datetimetotimestamp(now).time;
[2] ZQuery_penulis.ExecSQL;
[3] t2:=datetimetotimestamp(now).time;
[4] proses1 := proses1 + 1;
[5] Edit_banyak.Text := FloatToStr(proses1);
[6] Edit_Waktu.Text := FloatToStr(t2-t1);
```

Join:

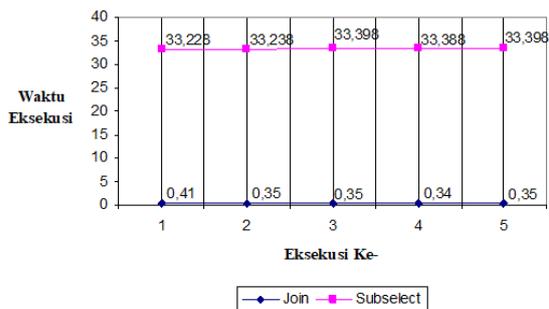
```
select nama,alamat,telp
from penulis,kol_penulis
where penulis.penulis_id = kol_penulis.penulis_id
and
kol_penulis.koleksi_id like 'KP%';
```

Berikut adalah hasil pengujian dari query diatas:

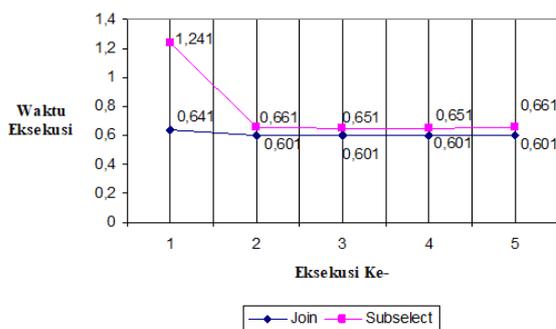
Tabel 1. Tabel Pengujian untuk Query Pernyataan 1

No	MySQL		MsAccess	
	Join	Subselect	Join	Subselect
1	0,41	33,228	0,641	1,241
2	0,35	33,238	0,601	0,661
3	0,35	33,398	0,601	0,651
4	0,34	33,388	0,601	0,651
5	0,35	33,398	0,601	0,661
Min	0,34	33,228	0,601	0,651
Max	0,41	33,398	0,641	1,241
Rearata	0,36	33,33	0,609	0,773

Tabel diatas menunjukkan waktu (dalam detik) tiap proses *query* yang berbentuk *join* dan *subselect*. Dari hasil diatas *query* yang berbentuk *join* jauh lebih cepat dalam pengaksesan data di *database MySQL* yaitu 32,97 detik atau 97.86 % lebih cepat dan pengaksesan data di *database Ms Access*, *join* lebih cepat 0,164 detik atau 11,86% lebih cepat. Grafik perbandingan *query* , seperti yang terlihat dibawah ini.



Gambar 8. Grafik Perbandingan untuk Query Pernyataan 1 di MySQL



Gambar 9. Grafik Perbandingan untuk Query Pernyataan 1 di Ms Access

Tampilkan semua karya ilmiah yang memiliki tema tertentu. Bentuk querynya adalah:

Subselect :

```
select judul,tahun
from koleksi
where koleksi_id in(select koleksi_id
from kol_tema
where tema_id = 'GIS');
```

Join :

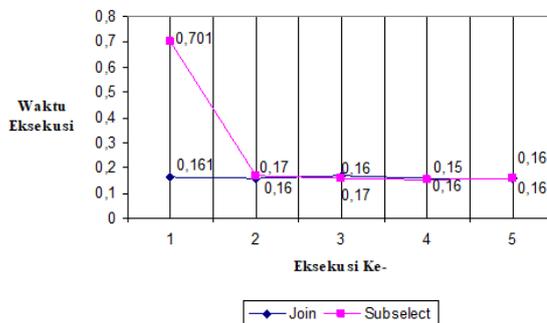
```
select judul,tahun
from koleksi,kol_tema
where koleksi.koleksi_id=kol_tema.koleksi_id and
kol_tema.tema_id = 'GIS';
```

Hasil pengujian dari query diatas yaitu:

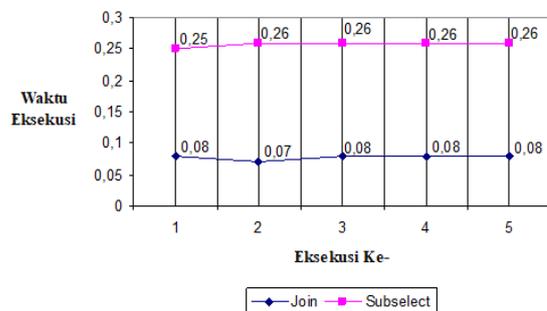
Tabel 2. Tabel Pengujian untuk Query Pernyataan 2

No	MySQL		MsAccess	
	Join	Subselect	Join	Subselect
1	0,08	0,25	0,161	0,701
2	0,07	0,26	0,16	0,17
3	0,08	0,26	0,17	0,16
4	0,08	0,26	0,16	0,16
5	0,08	0,26	0,16	0,16
Min	0,07	0,25	0,16	0,16
Max	0,08	0,26	0,17	0,701
Rearata	0,078	0,258	0,1622	0,2702

Tabel diatas menunjukkan waktu (dalam detik) tiap proses query yang berbentuk join dan subselect. Dari hasil diatas query yang berbentuk join jauh lebih cepat dalam pengaksesan data di database MySQL yaitu 0,18 detik atau 53,57 % lebih cepat dan pengaksesan data di database Ms Access, join lebih cepat 0,108 detik atau 24,97 % lebih cepat. Grafik perbandingan query, seperti yang terlihat dibawah ini.



Gambar 10. Grafik Perbandingan untuk Query Pernyataan 2 di MySQL



Gambar 11. Grafik Perbandingan untuk Query Pernyataan 2 di MS Access

Tampilkan data penulis (mahasiswa) tahun 2021 yang memiliki karya tulis. Bentuk querynya adalah:

Subselect :

```
select nama,alamat,telp
from penulis
where penulis_id in(select penulis_id
from kol_penulis
where koleksi_id like '%2021');
```

Join :

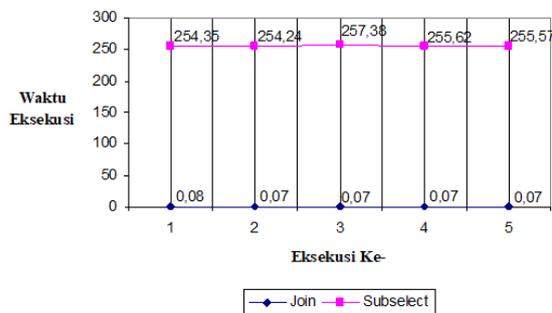
```
select nama,alamat,telp
from penulis,kol_penulis
where penulis.penulis_id=kol_penulis.penulis_id and
kol_penulis.koleksi_id like '%2021';
```

Hasil pengujian dari query diatas yaitu :

Tabel 3. Tabel Pengujian untuk Query Pernyataan3

No	MySQL		MsAccess	
	Join	Subselect	Join	Subselect
1	0,08	254,35	0,4	0,58
2	0,07	254,24	0,3	0,35
3	0,07	257,38	0,3	0,35
4	0,07	255,62	0,3	0,35
5	0,07	255,57	0,3	0,35
Min	0,07	254,24	0,3	0,35
Max	0,08	257,38	0,4	0,58
Rearata	0,072	255,432	0,32	0,396

Tabel diatas menunjukkan waktu (dalam detik) tiap proses query yang berbentuk join dan subselect. Dari hasil diatas query yang berbentuk join jauh lebih cepat dalam pengaksesan data di database MySQL yaitu 255,36 detik atau 99,94 % lebih cepat dan pengaksesan data di database Ms Access, join lebih cepat 0,076 detik atau 10,61 % lebih cepat. Grafik perbandingan query, seperti yang terlihat pada gambar dibawah ini.



Gambar 12. Grafik Perbandingan untuk Query Pernyataan 3 di MySQL

Cara Penghitungan Waktu Pengujian Query

Hasil pengujian dari query-query di atas, berdasarkan penghitungan data yang berupa waktu pengujian query-query tersebut. Berikut adalah cara penghitungan hasil pengujian query diatas.

- Selisih waktu join dengan subselect diperoleh dari pengurangan antara rata-rata waktu query dengan waktu terbesar dengan rata-rata waktu query terkecil. Contoh (diambil dari pernyataan query 1) :
 Diketahui : Rata-rata query join (Qj) = 0,36 detik
 Rata-rata query subselect (Qs) = 33,33 detik
 Penghitungan :
 Selisih = Qs – Qj
 = 33,33 detik – 0,36 detik
 = 32,97 detik
- Prosentase selisih waktu antara query join dengan subselect diperoleh dari pembagian selisih waktu

query dengan jumlah rata-rata waktu join dan subselect. Contoh (diambil dari pernyataan query 1):

Diketahui : Selisih waktu = 32,97 detik

Rata-rata Qj dan Qs = 33,69 detik

Penghitungan :

$$\text{Persentase} = (32,97/33,69) \times 100\% = 97,86 \%$$

- Rata-rata selisih waktu keseluruhan query yang diuji. Rata-rata selisih waktu keseluruhan diperoleh yaitu dengan penjumlahan keseluruhan waktu query kemudian dibagi dengan banyak data yang ada. Rata-rata waktu keseluruhan ini digunakan untuk menyimpulkan hasil penelitian ini. Data yang diperoleh dari hasil pengujian query ini yaitu sebagai berikut.

Tabel 4. Data pengujian query di MySQL

Eksekusi ke-	Waktu		Selisih
	Join	Subselect	
1	0,41	33,228	32,818
2	0,35	33,238	32,888
3	0,35	33,398	33,048
4	0,34	33,388	33,048
5	0,35	33,398	33,048
6	0,08	0,25	0,17
7	0,07	0,26	0,19
8	0,08	0,26	0,18
9	0,08	0,26	0,18
10	0,08	0,26	0,18
11	0,08	254,35	254,27
12	0,07	254,24	254,17
13	0,07	257,38	257,31
14	0,07	255,62	255,55
15	0,07	255,57	255,5
16	0,04	496,98	496,94
17	0,04	496,98	496,94
18	0,04	496,98	496,94
19	0,04	496,98	496,94
20	0,04	496,98	496,94
21	0,05	0,24	0,19
22	0,05	0,23	0,18
23	0,05	0,23	0,18
24	0,05	0,23	0,18
25	0,05	0,23	0,18
26	0,17	0,27	0,1
27	0,16	0,27	0,11
28	0,16	0,28	0,12
29	0,17	0,27	0,1
30	0,17	0,27	0,1
Rerata	0,13	131,08	130,96

Tabel diatas merupakan data hasil pengujian query yang dilakukan dengan menggunakan basis data MySQL. Dari tabel diatas diperoleh rata-rata waktu eksekusi pengujian *join* adalah 0,13 detik dan rata-rata eksekusi pengujian *subselect* adalah 131,08 detik, maka didapat selisih rata-rata waktu *join* dan *subselect* adalah 130,96 detik.

Berikut adalah data waktu pengujian yang dilakukan pada basis data Ms Access. Dari data yang ada diperoleh rata-rata waktu eksekusi pengujian *join* adalah 0,23 detik dan rata-rata waktu *subselect* adalah 0,32 detik, jadi diperoleh selisih rata-rata waktu *join* dan *subselect* adalah 0,09 detik. Data waktu yang diperoleh adalah seperti berikut.

Tabel 5. Data pengujian *query* di Ms Access

Eksekusi ke-	Waktu		selisih
	Join	Subselect	
1	0,641	1,241	0,6
2	0,601	0,661	0,06
3	0,601	0,651	0,05
4	0,601	0,651	0,05
5	0,601	0,661	0,06
6	0,161	0,701	0,54
7	0,16	0,17	0,01
8	0,17	0,16	0,01
9	0,16	0,16	0
10	0,16	0,16	0
11	0,40	0,58	0,18
12	0,30	0,35	0,05
13	0,30	0,35	0,05
14	0,30	0,35	0,05
15	0,30	0,35	0,05
16	0,13	0,23	0,1
17	0,14	0,22	0,08
18	0,13	0,21	0,08
19	0,13	0,22	0,09
20	0,13	0,21	0,08
21	0,08	0,16	0,08
22	0,07	0,08	0,01
23	0,08	0,08	0
24	0,07	0,08	0,01
25	0,08	0,08	0
26	0,08	0,27	0,19
27	0,07	0,14	0,07
28	0,07	0,14	0,07
29	0,08	0,15	0,07
30	0,08	0,14	0,06
Rerata	0,23	0,320	0,092

Pengujian Alfa Test

Pengujian menggunakan metode ini dilakukan dengan mengundang beberapa responden untuk menguji aplikasi dan memberikan daftar pertanyaan kepada masing-masing responden / aspek yang berhubungan dengan pengujian aplikasi untuk memberikan pendapatnya (sangat setuju, setuju, kurang setuju dan tidak setuju).

Berikut adalah hasil pengujian *alfa test* dari program pengujian perbandingan *join* dan *subselect*.

Tabel 6. Daftar Penilaian Responden

No	Pertanyaan	Penilaian			
		SS	S	K	T
1	Perancangan antarmuka program menarik	2	2		
2	Program mudah digunakan.	1	3		
3	Program berjalan dengan baik.	2	2		
4	Proses pengaksesan data berjalan dengan baik.	1	3		
5	Program sudah layak digunakan untuk membandingkan antara <i>query</i> berbentuk <i>join</i> dan <i>subselect</i>	3	1		
Jumlah		9	11		

Hasil perhitungan dari pengujian *alfa test* yang dilakukan dari 4 orang responden diperoleh sangat setuju = 9, setuju = 11, kurang setuju = 0, dan tidak setuju = 0. Untuk mengetahui prosentase dari masing-masing pilihan jawaban digunakan rumus sebagai berikut:

Prosentase

$$= \frac{\text{Hasil Penilaian}}{\text{Banyaknya Pertanyaan} \times \text{Jumlah Responden}} \times 100\%$$

Prosentase hasil penilaian yang memberikan jawaban sangat setuju, jawaban setuju dan kurang setuju adalah sebagai berikut:

$$\text{Sangat Setuju} = \frac{9}{20} \times 100\% = 45\%$$

$$\text{Setuju} = \frac{10}{20} \times 100\% = 55\%$$

Hasil responden yang memberikan penilaian tersebut diperoleh : sangat setuju 45%, setuju 55%, kurang setuju 0% dan tidak setuju 0%, jadi kesimpulannya bahwa program pengujian perbandingan *join* dan *subselect* sudah dapat berjalan dengan baik.

KESIMPULAN

Berdasarkan penelitian yang dilakukan yaitu pada pengujian *query* di DBMS yang berbeda, didapatkan selisih kecepatan akses data antara *join* dan *subselect* di MySQL dengan rata-rata waktu proses yaitu 130,95 detik dan di Ms Access yaitu 0,09 detik. Dari data yang diperoleh diatas dapat diambil kesimpulan bahwa *query* dengan bentuk *join* lebih cepat dari *query* dengan bentuk *subselect*, walaupun kedua *query* tersebut digunakan pada DBMS yang berbeda.

DAFTAR PUSTAKA

- Akbar, Syah Askar, A. S. (2012). Teknik Proteksi Terhadap Serangan SQL Injection Menggunakan Konsep Amnesia Pada Aplikasi WEB.
- Bismo Raharjo, Y. A., Mantriwira, D., & Sumanto, F. S. (2018). Analisis Kinerja Optimasi Query Cross Join, Natural Join Dan Full Outer Join. *CSRID (Computer Science Research and Its Development Journal)*, 9(2), 106. <https://doi.org/10.22303/csrid.9.2.2017.106-115>
- Dewi, M. M., & Rezeki, N. (2017). Analisis Perbandingan Optimasi Query Nasted Join dan Hash Join pada MySQL Server. *CSRID (Computer Science Research and Its Development Journal)*, 9(1), 31. <https://doi.org/10.22303/csrid.9.1.2017.31-42>
- Elmasri, R. and Navathe, S. B. (2011). *Fundamentals of Database Systems*. (A. Sixth Edition, Ed.). Wesley, United States of America.: Sixth Edition, Addison.
- Foster, E. C. (2016). *Database Systems, A Pragmatic Approach, A Texbook Introduction to Database Systems Theory and Practice*. USA: Apress.
- Lubis, A. (2016). *Basis Data Dasar*. Yogyakarta: Deepublish.
- Lumbanraja, H. D. (2018). Perancangan Sistem Informasi Akademik Online Menggunakan Black Box Testing Pada Sekolah Tinggi Ilmu Ekonomi Surya Nusantara. *TelKa*, 8(2), 9–18. <https://doi.org/10.36342/teika.v8i2.664>
- Muhammad, L. J., Zakariyau, Y. B., & Ali, A. G. (2015). Efficient Algorithm for Multi Query Optimization. *International Journal of Database Theory and Application*, 8(6), 133–138. <https://doi.org/10.14257/ijtda.2015.8.6.12>
- Pamungkas, R. (2018). Optimalisasi Query Dalam Basis Data My Sql Menggunakan Index. *RESEARCH: Computer, Information System & Technology Management*, 1(1), 27. <https://doi.org/10.25273/research.v1i1.2453>
- Sasmito, G. W. (2017). Penerapan Metode Waterfall Pada Desain Sistem Informasi Geografis Industri Kabupaten Tegal. *Jurnal Informatika: Jurnal Pengembangan IT (JPIT)*, 2(1), 6–12.
- Setiyadi, D. (2020). *Sistem Basis Data dan SQL (Structured Query Language)*. Bogor: Mitra Macana Media.
- Sudarmaji. (2016). Migrasi dan Optimalisasi Database Sistem Informasi berbasis E-Learning Program Diploma III Manajemen Informatika Universitas Muhammadiyah Metro. *MIKROTIK: Jurnal Manajemen Informatika*, 6(2), 8.
- Sumantri, R Bagus Bambang, Setiawan, W., & Triwibowo, D. N. (2022). RANCANG BANGUN APLIKASI MEDIA JASA DESAIN LOGO DENGAN, 6(2), 157–163.
- Sumantri, Raden Bagus Bambang, Setiawan, R. A., & Sandi A, A. S. (2022). Sistem Informasi Geografis Untuk Pemetaan Pariwisata Kabupaten Karanganyar Berbasis Web. *METHOMIKA Jurnal Manajemen Informatika Dan Komputerisasi Akuntansi*, 6(1), 1–9. <https://doi.org/10.46880/jmika.vol6no1.pp1-9>
- Susilo, M., & Kurniati, R. (2018). Rancang Bangun Website Toko Online Menggunakan Metode Waterfall, 2(2), 98–105.
- Thangam, R., & Peter, J. (2016). An Extensive Survey on Various Query Optimization Techniques. *International Journal of Computer Science and Mobile Computing*, 5(6), 233–237.