Jurnal Sains Komputer & Informatika (J-SAKTI)
Volume 6 Nomor 1, Maret 2022, pp. 499-506
ISSN: 2548-9771/EISSN: 2549-7200
https://tunasbangsa.ac.id/ejurnal/index.php/jsakti

Perbandingan Response Time Penggunaan Index, Views, dan Materialized Views Database Mysql

Edi Witono[1], Parno[2]
[1,2]Universitas Budi Luhur, Indonesia
[1]2111600678@student.budiluhur.ac.id, [2]2111600660@student.budiluhur.ac.id

Abstract

The existence of information technology today can help the government in carrying out the task of providing services to the public that are efficient and transparent. One of them is the SisfoSPM application used by the Badan Litbang unit of the Ministry of Education and Culture's National Education and Culture Agency to verify spending budgets in the use of the State Revenue and Expenditure Budget. This web-based application has been using Mysql as its database since 2017. The amount of data from each year, makes this application run slowly, especially related to queries that display log data verification results of each employee. This study used experimental methods with employee log transaction data from 2017-2021 amounting to 780,212 records. The study uses three stages: preparation, experimentation and analysis of results. The results of this research, obtained the results of the main response time query module sisfoSPM application employee log calculation with the amount of data as much as 780,212 using the index method for 1.95 seconds, using the views method for 1.85 seconds and using the materialized views method for 0.01 seconds. With these results, researchers can draw the conclusion that the use of materialized views provides the fastest response time results.

Keywords: *Response time, MySQL, Index, Views, Materialized Views.*

Abstrak

Keberadaan teknologi informasi saat ini dapat membantu pemerintah dalam melaksanakan tugas memberikan pelayanan kepada publik yang efisien dan transparan. Salah satunya adalah aplikasi SisfoSPM yang digunakan oleh satuan kerja Badan Litbang Kementerian Pendidikan dan Kebudayaan Nasional untuk melakukan verifikasi tagihan belanja dalam penggunaan Anggaran Pendapatan dan Belanja Negara (APBN). Aplikasi berbasis web ini menggunakan Mysql sebagai databasenya sejak tahun 2017. Banyaknya data dari tahun ke tahun, membuat aplikasi ini berjalan lambat, terutama terkait query yang menampilkan log data hasil verifikasi masing-masing pegawai. Penelitian ini menggunakan metode eksperimental dengan data-data transaksi log pegawai dari tahun 2017-2021 sejumlah 780.212 record. Penelitian ini menggunakan tiga tahap yaitu persiapan, eksperimen dan analisis hasil. Hasil penelitian ini, diperoleh hasil response time query utama modul penghitungan log pegawai Aplikasi SisfoSPM dengan jumlah data sebanyak 780.212 menggunakan metode index selama 1,95 detik, menggunakan metode views selama 1,85 detik dan menggunakan metode materialized views selama 0,01 detik. Dengan hasil tersebut, peneliti dapat menarik kesimpulan bahwa penggunaan materialized views memberikan hasil response time yang paling cepat.

Kata kunci: *Response time, MySQL, Index, Views, Materialized Views.*

1. PENDAHULUAN

Implementasi teknologi informasi dalam sistem pemerintahan yang biasa disebut *E-Government* ditujukan untuk menjamin keterpaduan sistem pengelolaan dan pengolahan dokumen dan informasi elektronik dalam mengembangkan sistem pelayanan publik yang transparan [1]. Salah satu tujuan dari *E-Government* tersebut adalah agar pemerintah dapat menjalankan sistem yang secara lebih efisien dan transparan. Keberadaan teknologi informasi sangat membantu suatu organisasi dalam menjalankan aktifitas-aktifitas bisnis yang ada dalam organisasi [2]. Instansi Pemerintahan berlomba-lomba menggunakan aplikasi untuk

mempermudah pekerjaan mereka, tidak terkecuali satuan kerja Badan Penelitian dan Pengembangan Kementerian Pendidikan dan Kebudayaan Nasional (Balitbang Kemendikbud). Balitbang Kemendikbud menggunakan Aplikasi SisfoSPM untuk melakukan verifikasi tagihan belanja dalam penggunaan Anggaran Pendapatan dan Belanja Negara (APBN). Aplikasi SisfoSPM merupakan aplikasi berbasis *web* yang menggunakan database *MySQL* dan telah digunakan sejak tahun 2017. Seiring berjalan waktu, pertambahan data membuat aplikasi ini berjalan lambat, salah satunya adalah terkait *query* menampilkan log data hasil verifikasi masing-masing pegawai.

Dalam optimasi database *MySQL*, ada banyak cara yang dapat digunakan. Salah satu cara yang digunakan adalah penggunaan *index*. *Index* pada database digunakan untuk mencari nilai kolom pada tabel tertentu dengan cepat, tanpa membaca tabel mulai dari baris pertama [3]. Selain penggunaan *index*, penggunaan *views* dan *materialized views* juga lazim digunakan di database *Mysql*. *View* merupakan kumpulan *query* yang membentuk *virtual table* untuk menampilkan data dari satu tabel atau lebih [4], sehingga mempermudah pemanggilan dan juga mempercepat proses pengambilan data. Sedangkan *materialized view* biasanya digunakan untuk membuat *object view* dari suatu *query* dimana hasil *query* tersebut dibuat dalam sebuah tabel secara fisik [5].

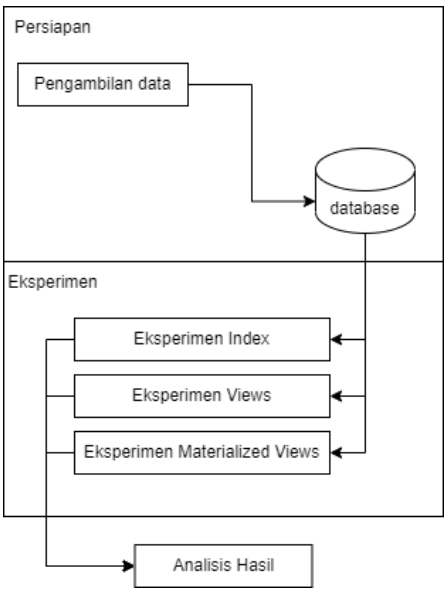
Terdapat beberapa penelitian sebelumnya yang dijadikan dasar sebagai referensi penelitian ini. Hery Siswanto [6] melakukan optimalisasi *query* pada sistem SISSDM atau HRIS pada Universitas "XYZ" dengan menggunakan metode *Index*, *Symbol Operator* dan *Wild Card*. Pada penelitian tersebut diperoleh hasil bahwa dengan optimasi diperoleh *response time* tercepat adalah 0,015 detik, sedangkan sebelumnya tanpa optimasi diperoleh waktu tercepat 0,023 detik. Selanjutnya Noviyanti [7] melakukan penelitian dengan membandingkan *query response time* menggunakan metode *cross product* dan *views* dengan sampel data penilaian perkuliahan sebanyak 100, 500 dan 1000 baris data. Pada penelitian tersebut disimpulkan bahwa penggunaan *views* memiliki *response time* yang lebih cepat dibandingkan dengan penggunaan *cross product* atau *query* biasa dengan selisih 0,0014 sampai 0,0040 detik. Penelitian senada dilakukan oleh Pipit Kurnia Safitri [8] pada data Sistem Informasi Penjadwalan Mata Pelajaran Sekolah menggunakan metode *views* yang menyimpulkan bahwa penggunaan *views* memiliki penulisan sintaks lebih sedikit dan *response time* yang lebih cepat. Dalam penelitian Febrianta Surya Nugraha [9] pada laporan transaksi penjualan menggunakan *where clause, views* dan *materialized views*, diperoleh kesimpulan bahwa penggunaan *materialized views* lebih cepat dibanding dengan *where clause* dan *views*.

Tujuan Penelitian ini adalah melakukan perbandingan response time penggunaan *index*, *views* dan *materialized views* pada database SisfoSPM Balitbang Kementerian Pendidikan dan Kebudayaan. Adapun Batasan Masalah yang ada dalam penelitian ini adalah hanya pada *query* penghitungan log data masing-masing pegawai di Aplikasi SisfoSPM. Perbedaan penelitian ini dengan penelitian-penelitian terdahulu adalah objek, metode, kombinasi data dan waktu penelitian.

Jurnal Sains Komputer & Informatika (J-SAKTI)
Volume 6 Nomor 1, Maret 2022, pp. 499-506
ISSN: 2548-9771/EISSN: 2549-7200
https://tunasbangsa.ac.id/ejurnal/index.php/jsakti

2. METODOLOGI PENELITIAN

Jenis penelitian ini adalah dengan menggunakan metode eksperimental dengan data-data transaksi log pegawai dari tahun 2017–2021 sejumlah 780.212 record pada aplikasi SisfoSPM. Penelitian ini menggunakan tiga tahap yaitu persiapan, eksperimen dan analisis hasil. Pada tahap persiapan, penulis mengumpulkan data pada database aplikasi SisfoSPM, kemudian di *copy* ke dalam *server* lokal. Pada tahap eksperimen, penulis membuat berbagai skenario penggunaan *index*, *views* dan *materialized views* kemudian dicatat *response time* pada masing-masing penggunaan metode. Pada tahap analisis hasil dilakukan perbandingan *response time* penggunaan *index*, *views* dan *materialized views* terhadap database aplikasi SisfoSPM. Penelitian ini dilakukan dengan menggunakan Laptop Matebook D14 dengan spesifikasi Intel Core I3-10110U, RAM 8GB dan *operating system* Windows 10 Home. Untuk *tools* yang digunakan untuk eksperimen adalah HeidiSQL versi 11.2.0.6213. Alur penelitian ini dapat dilihat pada gambar 1.



Gambar 1. Alur Penelitian

3. HASIL DAN PEMBAHASAN

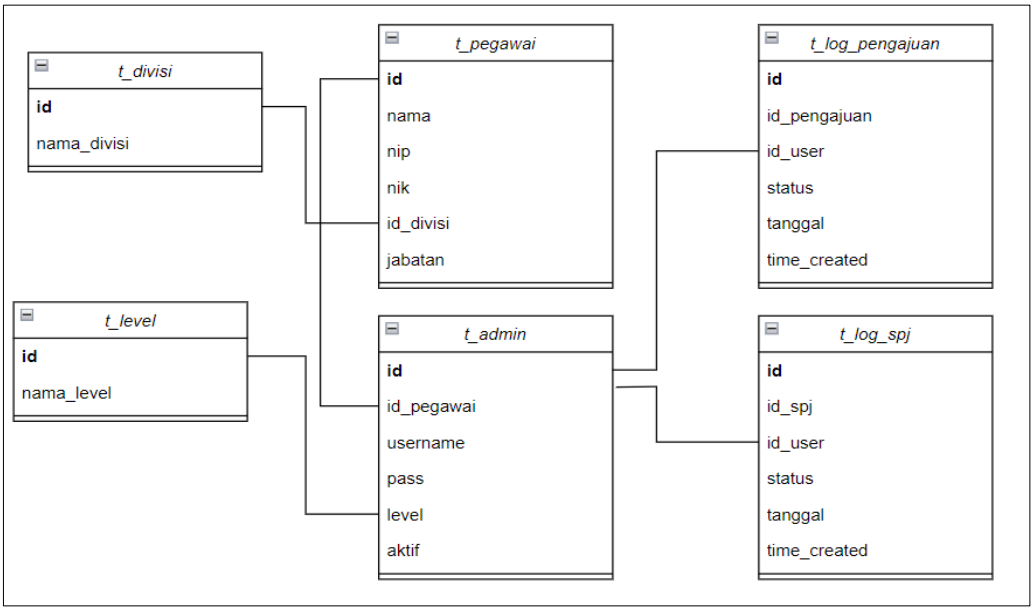
3.1. Tahap Persiapan

Pada tahap ini, peneliti mengambil data dari database *production* untuk modul penghitungan log pegawai pada aplikasi SisfoSPM. Metode pengambilan data yang digunakan adalah dengan menggunakan metode *MySQL Dump*. Pada metode *MySQL Dump*, akan dibuat salinan data berupa sebuah *file text* yang berisi perintah SQL setelah dilakukan proses dumping [10]. Selanjutnya *file text* yang berisi perintah-perintah *CREATE* dan *INSERT* tersebut jika dieksekusi akan membentuk struktur database beserta isinya secara otomatis [11]. *MySQL Dumping* dilakukan pada server database *production,* menggunakan perintah :

```
shell > mysqldump --all-databases > backup.sql
```

Jurnal Sains Komputer & Informatika (J-SAKTI)
Volume 6 Nomor 1, Maret 2022, pp. 499-506
ISSN: 2548-9771/EISSN: 2549-7200
https://tunasbangsa.ac.id/ejurnal/index.php/jsakti

Selanjutnya file hasil proses *MySQL Dumping* tersebut di-*restore* pada laptop peneliti. Terdapat 6 tabel yang digunakan, yaitu: tabel t_admin, t_pegawai, t_divisi, t_level, t_log_pengajuan dan t_log_spj. Skema tabel yang digunakan pada modul penghitungan log pegawai dapat dilihat pada gambar 2.



Gambar 2. Skema tabel modul penghitungan log pegawai pada Aplikasi SisfoSPM

3.2. Tahap Eksperimen

Pada tahap ini peneliti menjalankan perhitungan *response time* pada *query* awal sebelum optimasi, yaitu :

```
select id_user,nama,nip,nama_divisi,tanggal,sum(dok_spj) as dok_spj,sum(dok_usulan) as dok_usulan
FROM
(select a.id_user,c.nama,c.nip,d.nama_divisi,tanggal,count(*) as dok_spj,0 as dok_usulan
from t_log_spj  a left join t_admin b on a.id_user = b.id
left join t_pegawai c on b.id_pegawai=c.id left join t_divisi d on c.id_divisi=d.id group by a.id_user,tanggal
union all
select a.id_user,c.nama,c.nip,d.nama_divisi, tanggal,0 as dok_spj,count(*) as dok_usulan
from t_log_pengajuan  a left join t_admin b on a.id_user = b.id
left join t_pegawai c on b.id_pegawai=c.id left join t_divisi d on c.id_divisi=d.id
group by a.id_user,tanggal) a where year(tanggal)='2021' group by id_user,tanggal
```

Dengan jumlah data sebanyak 780.212 baris, *query* tersebut menghasilkan waktu *response time* 12,65 detik. Eksperimen pertama yang dilakukan adalah menggunakan metode *index*, peneliti terlebih dahulu melakukan identifikasi tabel mana saja yang akan ditambahkan *index*. Identifikasi tabel dilakukan dengan melihat jumlah baris data pada masing-masing tabel. Jumlah baris data pada masing-masing tabel dapat dilihat pada tabel 1.

Jurnal Sains Komputer & Informatika (J-SAKTI)
Volume 6 Nomor 1, Maret 2022, pp. 499-506
ISSN: 2548-9771/EISSN: 2549-7200
https://tunasbangsa.ac.id/ejurnal/index.php/jsakti

Tabel 1. Jumlah *row* data tabel Aplikasi SisfoSPM

No.	Nama Tabel	Tahun				
		2017	2018	2019	2020	2021
1	t_admin	97	97	97	97	97
2	t_pegawai	97	97	97	97	97
3	t_divisi	4	4	4	4	4
4	t_level	3	3	3	3	3
5	t_log_pengajuan	95,751	199,539	278,023	356,300	418,205
6	t_log_spj	85,566	180,774	243,726	306,486	361,806
	Jumlah Row	**181,518**	**380,514**	**521,950**	**662,987**	**780,212**

Dari informasi jumlah baris data tersebut di atas, maka ditambahkan *index* pada dua tabel utama yaitu t_log_pengajuan dan t_log_spj, dimana kolom yang dipanggil dengan perintah *where* dalam *query* utama di tambahkan sebagai *index*. Penambahan *index* pada tabel-tabel tersebut adalah menggunakan *query* sebagai berikut:

```
CREATE INDEX `idx_log_pengajuan` ON t_log_pengajuan(`id_user`, `tanggal`) USING BTREE

CREATE INDEX `idx_log_spj` ON t_log_spj(`id_user`, `tanggal`) USING BTREE
```

Selanjutnya, eksperimen dilanjutkan dengan menjalankan *query* utama pada tabel-tabel yang telah ditambahkan *index*. Dengan penambahan *index* tersebut, didapatkan hasil *response time* selama 1,95 detik.

Eksperimen berikutnya adalah dengan metode pembuatan *views*. Pembuatan *views* ini dilakukan dengan menduplikat *query* utama menjadi *query views* dan menyederhanakan *query* utama untuk memanggil *views* tersebut. *Query* pembuatan *views* adalah sebagai berikut:

```
CREATE VIEW `v_log_pegawai` AS
select id_user,nama,nip,nama_divisi,tanggal,sum(dok_spj) as dok_spj,sum(dok_usulan) as dok_usulan FROM
(select a.id_user,c.nama,c.nip,d.nama_divisi,tanggal,count(*) as dok_spj,0 as dok_usulan
from t_log_spj  a left join t_admin b on a.id_user = b.id
left join t_pegawai c on b.id_pegawai=c.id left join t_divisi d on c.id_divisi=d.id group by a.id_user,tanggal
union all
select a.id_user,c.nama,c.nip,d.nama_divisi, tanggal,0 as dok_spj,count(*) as dok_usulan
from t_log_pengajuan  a left join t_admin b on a.id_user = b.id
left join t_pegawai c on b.id_pegawai=c.id left join t_divisi d on c.id_divisi=d.id
group by a.id_user,tanggal) a where year(tanggal)='2021' group by id_user,tanggal
```

Selanjutnya, eksperimen dilanjutkan dengan menjalankan *query* utama yang telah dimodifikasi menggunakan metode *views*. Dengan penggunaan *views* tersebut didapatkan hasil *response time* selama 1,85 detik. E

ksperimen terakhir adalah dengan membuat *materialized views*. Pembuatan *materialized views* ini dilakukan dengan menduplikat *query views* menjadi sebuah *temporary* tabel dan menyederhanakan *query* utama untuk memanggil *materialized views* tersebut. *Query* pembuatan *materialized views* adalah sebagai berikut :

Jurnal Sains Komputer & Informatika (J-SAKTI)
Volume 6 Nomor 1, Maret 2022, pp. 499-506
ISSN: 2548-9771/EISSN: 2549-7200
https://tunasbangsa.ac.id/ejurnal/index.php/jsakti

```
CREATE TABLE `t_log_pegawai` AS
select id_user,nama,nip,nama_divisi,tanggal,sum(dok_spj) as dok_spj,sum(dok_usulan) as dok_usulan FROM
(select a.id_user,c.nama,c.nip,d.nama_divisi,tanggal,count(*) as dok_spj,0 as dok_usulan
from t_log_spj  a left join t_admin b on a.id_user = b.id
left join t_pegawai c on b.id_pegawai=c.id left join t_divisi d on c.id_divisi=d.id group by a.id_user,tanggal
union all
select a.id_user,c.nama,c.nip,d.nama_divisi, tanggal,0 as dok_spj,count(*) as dok_usulan
from t_log_pengajuan  a left join t_admin b on a.id_user = b.id
left join t_pegawai c on b.id_pegawai=c.id left join t_divisi d on c.id_divisi=d.id
group by a.id_user,tanggal) a where year(tanggal)='2021' group by id_user,tanggal
```

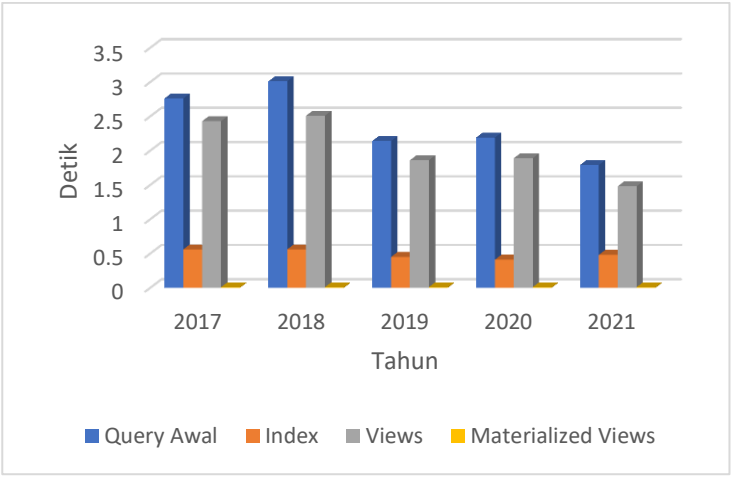
Selanjutnya, eksperimen dilanjutkan dengan menjalankan *query* utama yang telah dimodifikasi menggunakan metode *materialized views*. Pada eksperimen dengan metode *materialized views* ini dibutuhkan waktu pembuatan tabel *materialized views* selama 4,63 detik dan waktu *query* utama mendapatkan *response time* selama 0,01 detik. Selanjutnya, dengan teknik yang sama, peneliti melakukan simulasi untuk data Aplikasi SisfoSPM setiap tahun, mulai dari tahun 2017 sampai tahun 2021.

3.3. Tahap Analisis Hasil

Berdasarkan eksperimen yang telah dilakukan, untuk mengetahui performa kecepatan atau *response time query* utama, perlu dilakukan perbandingan hasil eksperimen baik menggunakan metode *Index*, *Views* dan *Materialized Views*. Perbandingan *response time query* utama modul penghitungan log pegawai pada aplikasi SisfoSPM dapat dilihat pada tabel 2 dan gambar 3.

Tabel 2. Hasil eksperimen

Tahun	**2017**	**2018**	**2019**	**2020**	**2021**
Jumlah Data	181.317	198.996	141.436	141.037	117.225
	Response Time (detik)				
Query Awal	2,76	3,01	2,14	2,19	1,79
Metode index	0,56	0,56	0,45	0,41	0,48
Metode views	2,43	2,51	1,86	1,89	1,48
Metode materialized views	0,01	0,01	0,01	0,01	0,01



Gambar 3. Grafik perbandingan *response time query* data per tahun

Berdasarkan hasil perbandingan *response time* dalam memproses *query* log transaksi pegawai per tahun mulai dari tahun 2017 sampai dengan tahun 2021, dengan jumlah data setiap tahun 141.037 – 198.996 baris data, diperoleh *response time* 0,41 – 0,56 detik menggunakan metode *index*, 1,48 – 2,51 detik menggunakan metode *views*, dan 0,01 detik menggunakan *materialized view*s. Dengan hasil perbandingan ini, penggunaan *materialized view*s jauh lebih cepat dari pada menggunakan *index* maupun *view*s.

4. SIMPULAN

Dari hasil eksperimen yang telah dilakukan, diperoleh hasil *response time query* utama modul penghitungan log pegawai Aplikasi SisfoSPM dengan jumlah data sebanyak 780.212 baris data, dengan menggunakan metode *index* selama 1,95 detik, menggunakan metode *views* selama 1,85 detik dan menggunakan metode *materialized views* selama 0,01 detik. Dengan hasil tersebut, peneliti dapat menarik kesimpulan bahwa penggunaan *materialized views* memberikan hasil *response time* yang paling cepat. Hal ini mendukung penelitian-penelitian sebelumnya yang menyatakan bahwa terdapat beberapa metode untuk meningkatkan performa kecepatan proses *query* dalam database *MySQL*, diantaranya adalah dengan metode *Index*, *Views* dan *Materialized Views*. Namun perlu diingat bahwa metode *materialized views* ini data di dalamnya tidak otomatis ter-*update* apabila terdapat penambahan dan pengurangan data pada tabel induk, sehingga perlu dipertimbangkan juga kebutuhan waktu untuk pembuatan *materialized view*s tersebut.

DAFTAR PUSTAKA

[1] Instruksi Presiden RI, "Instruksi Presiden RI No. 3 Tahun 2003 tentang Kebijakan dan Strategi Nasional Pengembangan E-Government," 2003.

[2] W. D. Novan, R. F. S. T, S. T. Falahah, F. R. Industri, and U. Telkom, "Analisis Dan Penyusunan Rancangan Enterprise Architecture Sistem Pemerintahan Berbasis Elektronik Menggunakan Framework Togaf Analysis and Drafting of Enterprise Architecture System of Electronic Based Government Using Togaf Framework on Construction Serv," *e-Proceeding Eng.*, vol. 8, no. 2, pp. 2633–2646, 2021.

[3] R. Pamungkas, "Optimalisasi Query Dalam Basis Data My Sql Menggunakan Index," *Res. Comput. Inf. Syst. Technol. Manag.*, vol. 1, no. 1, p. 27, 2018, doi: 10.25273/research.v1i1.2453.

[4] Y. Z. Ayık and F. Kahveci, "Materialized View Effect on Query Performance," vol. 11, no. 9, pp. 1070–1073, 2017.

[5] R. Elmasri and S. B. Navathe, *Fundamentals of Database Systems*, 7th ed. Pearson, 2015.

[6] H. Siswanto, T. Andi, and K. Kusrini, "Optimasi Query Pada Human Resource Information System (HRIS) di Universitas XYZ," *JMAI (Jurnal Multimed. Artif. Intell.*, vol. 2, no. 1, pp. 1–6, 2018, doi: 10.26486/jmai.v2i1.53.

[7] P. Noviyanti, A. Deolika, S. Hartinah, C. A. Haris, T. Maryana, and N. D. Sari, "Perbandingan Query Response Time pada Model Query View dan Cross Product Comparison of Query Respone Time in the Query View and Cross

Product," *e-Jurnal JUSITI*, vol. 7, no. 2, pp. 131–141, 2018, [Online]. Available: https://ejurnal.dipanegara.ac.id/index.php/jusiti/article/view/248.

[8] P. K. Safitri, W. W. Winarno, and E. Pramono, "Optimasi Query untuk Sistem Informasi Penjadwalan Mata Pelajaran Sekolah Menggunakan View (Studi Kasus : SMK VIP Purworejo)," vol. XIII, pp. 46–51, 2018, [Online]. Available: http://jti.respati.ac.id/index.php/jurnaljti/article/view/259.

[9] F. S. Nugraha, F. H. Purwanto, and F. E. N. Saputro, "Optimasi Query Pada Laporan Transaksi Penjualan Menggunakan Materialized View (Kasus : Moonly café)," *CCIT J.*, vol. 11, no. 1, pp. 1–14, 2018, doi: 10.33050/ccit.v11i1.552.

[10] Tawar and S. Wahyuningsih, "Pembandingan Metode Backup Database MySQL antara Replikasi dan MySQLDump," *Jusi*, vol. 1, no. 1, pp. 45–52, 2011.

[11] A. Solichin, "MySQL Dari Pemula Hingga Mahir," *Univ. Budi Luhur, Jakarta*, no. January 2010, pp. 1–117, 2010.