

# Migrasi Database Mysql Ke Postgresql Pada Aplikasi Sistem Evaluasi Dosen Oleh Mahasiswa (EDOM) Jurusan Ekonomi Dan Bisnis

Muhammad Irfan Lutfi <sup>1</sup>, Imam Asrowardi <sup>2</sup> Agiska Ria Supriyatna<sup>3</sup>

Jurusan Ekonomi dan Bisnis, Program Studi Manajemen Informatika, Politeknik Negeri Lampung, Kota Bandar Lampung, Lampung 35141

## INFORMASI ARTIKEL

Diterima 2022/10/11  
Direvisi 2022/10/31  
Diterbitkan 2022/11/02

### Kata kunci:

Migrasi Database;  
MySQL;  
PostgreSQL;  
EDOM;  
SDLC;

## ABSTRAK

Pelaksanaan Pengisian Evaluasi Dosen Oleh Mahasiswa (EDOM) yang dilakukan setiap akhir semester ketika kegiatan Ujian Akhir Semester (UAS), telah dikembangkan menjadi sistem informasi terkomputerisasi secara online menggunakan database MySQL. Seiring bertambahnya jumlah data pada Sistem informasi EDOM, dibutuhkan penyimpanan data yang memiliki kompleksitas yang tinggi, sehingga dari sisi database harus memiliki kinerja yang tinggi pula dan bisa menangani jumlah data yang besar. Untuk kebutuhan pada sistem informasi EDOM, MySQL sudah tidak mampu memberikan performa terbaiknya dalam menangani jumlah data yang besar, sehingga diperlukan database yang memiliki kinerja lebih baik lagi. Tujuan dari tugas akhir ini adalah Memigrasikan Database MySQL ke PostgreSQL pada Sistem Aplikasi (Evaluasi Dosen Oleh Mahasiswa) EDOM Jurusan Ekonomi dan Bisnis, agar mengatasi masalah pada database MySQL yang sudah tidak mampu memberikan performa terbaiknya dalam menangani jumlah data yang besar. Metode pengembangan sistem yang digunakan adalah metode SDLC. Berdasarkan hasil pengujian yang dilakukan dengan cara menguji kinerja query pada kedua dbms, PostgreSQL mempunyai performa yang stabil dibandingkan dengan MySQL, pada jumlah data yang sama MySQL unggul pada update dan select biasa tanpa klausa tambahan. Sementara itu PostgreSQL lebih unggul pada create, delete data dan klausa join yang kompleks. Pengujian selanjutnya dilakukan pada Aplikasi menggunakan function php yaitu *microtime* dengan menulis script pada awal body dan sebelum tutup body halaman html pada aplikasi yang menggunakan dbms MySQL dan PostgreSQL, dilakukan guna menguji kecepatan dalam memuat data dalam aplikasi sistem EDOM, hasilnya penggunaan dbms PostgreSQL lebih cepat dalam memuat data, karena data disajikan menggunakan klausa join yang kompleks. Hal ini dikarenakan PostgreSQL mendukung *locking* di level yang lebih rendah, yaitu row sehingga menjadi salah satu keuntungan dalam dilakukannya migrasi database karena proses bisnis pada Aplikasi Sistem EDOM yang dilakukan menjadi lebih efektif.

# Database Migration MySQL to PostgreSQL In System Application Lecturer Evaluation By Student (EDOM) Departement of Economics and Business

## ARTICLE INFO

Received 2022/10/11  
Revised 2022/10/31  
Published 2022/11/02

### Keyword:

Database Migration;  
MySQL;  
PostgreSQL;  
EDOM;  
SDLC;

## ABSTRACT

Implementation of Filling Out Lecturer Evaluations by Students (EDOM) which is carried out at the end of each semester during the Semester Final Examination (UAS) activities, has been developed into an online computerized information system using a MySQL database. As the amount of data in the EDOM information system increases, data storage that has a high complexity is needed, so that from the database side it must have high performance and can handle large amounts of data. For the needs of the EDOM information system, MySQL is no longer able to provide its best performance in handling large amounts of data, so a database that has better performance is needed. The purpose of this final project is to Migrate MySQL Database to PostgreSQL on Application System (Evaluation of Lecturers by Students) EDOM Department of Economics and Business, in order to overcome the problem of MySQL database which is no longer able to provide its best performance in handling large amounts of data. The system development method used is the SDLC method. Based on the results of tests carried out by testing query performance on both dbms, PostgreSQL has a stable performance compared to MySQL, at the same amount of data MySQL excels in regular updates and selects without additional clauses. Meanwhile PostgreSQL excels at creating, deleting data and complex join clauses. The next test is carried out on the application using the php function, namely microtime by writing a script at the beginning of the body and before closing the body of the html page on applications that use MySQL and PostgreSQL dbms, carried out to test the speed in loading data in EDOM system applications, the result is that the use of PostgreSQL dbms is faster in load data, because the data is presented using complex join clauses. This is because PostgreSQL supports locking at a lower level, namely row so that it becomes one of the advantages in doing database migration because the business processes in the EDOM System Application are made more effective.

## 1. PENDAHULUAN

Dosen merupakan tenaga pendidik profesional yang mempunyai tanggung jawab mentransformasikan, mengembangkan ilmu pengetahuan, dan memperluas ilmu tersebut melalui pendidikan dan pengabdian kepada masyarakat. Tugas umum dari dosen adalah mengajar dan membimbing mahasiswa sesuai kompetensi dan keahliannya sebagaimana yang terdapat pada Undang-undang Nomor 14 pada tahun 2005 (Lesmana & Irawan, 2018). Kinerja dosen merupakan elemen penting dalam suatu institusi sistem pendidikan yang ada di perguruan tinggi negeri. Dosen harus memiliki kompetensi pedagogik, kepribadian, sosial, dan profesional (Nento, 2018).

Evaluasi kinerja dosen diperlukan dalam proses terciptanya peningkatan kualitas pengajaran, pengembangan diri pada dosen, peningkatan kepuasan mahasiswa terhadap kinerja dosen, upaya pencapaian tujuan program studi/jurusan/institusi pendidikan, peningkatan penilaian masyarakat terhadap institusi pendidikan. Evaluasi kinerja dosen dapat membantu

mahasiswa dalam mendapatkan pengajaran yang berkualitas. Sistem dan kualitas mutu pengajaran perlu dievaluasi dan ditingkatkan karena pendidikan adalah suatu kegiatan yang mahal upaya pencapaian taraf hidup yang lebih baik (Harisantoso, 2012).

Pelaksanaan pengisian Evaluasi Dosen Oleh Mahasiswa (EDOM) dilakukan setiap akhir semester ketika kegiatan Ujian Akhir Semester (UAS). Pengisian EDOM pada mulanya dilakukan secara manual dengan membagikan form kuesioner kepada setiap mahasiswa dengan bentuk lembaran kertas, setelah itu hasil dari kuesioner dikumpulkan dan disimpan dalam aplikasi *Microsoft Excel*. Hasil dari EDOM yang sudah direkap, dibagikan kepada setiap dosen sesuai dengan matakuliah yang diampunya. Saat ini, EDOM sudah dikembangkan menjadi sistem informasi yang terkomputerisasi secara *online* (Lesmana & Irawan, 2018).

Pengembangan sistem informasi EDOM saat ini menggunakan database MySQL yang merupakan database paling populer di dunia dan bersifat *open source* (Warman & Ramdaniansyah, 2018). Seiring bertambahnya jumlah data pada Sistem informasi EDOM, dibutuhkan penyimpanan data yang memiliki kompleksitas yang tinggi, sehingga dari sisi database harus memiliki kinerja yang tinggi pula dan bisa menangani jumlah data yang besar. Untuk kebutuhan pada sistem informasi EDOM, MySQL sudah tidak mampu memberikan performa terbaiknya dalam menangani jumlah data yang besar, sehingga diperlukan database yang memiliki kinerja lebih baik lagi. PostgreSQL mengalahkan MySQL untuk query dengan klausa *join* yang kompleks. Hal ini dikarenakan PostgreSQL mendukung *locking* di level yang lebih rendah, yaitu row (Wibowo & Wiguna, 2019).

Berdasarkan permasalahan yang ada pada sistem informasi EDOM saat ini, solusi yang tepat adalah dengan pengembangan sistem menggunakan database PostgreSQL yang dinilai mempunyai kelebihan saat memuat data yang tinggi. Selain itu PostgreSQL memiliki arsitektur multiproses yang berarti memiliki tingkat stabilitas yang lebih tinggi dibandingkan MySQL. Database PostgreSQL dapat mengatasi kompleksitas yang cukup besar, karena setiap akhir semester pertambahan data bisa membuat sistem semakin berat. (Wibowo & Wiguna, 2019).

## 2. METODE

*System Development Life Cycle (SDLC)* adalah metodologi klasik yang digunakan dalam pengembangan, pemeliharaan, dan penggunaan sistem informasi. Metodologi dalam sistem ini berupa siklus hidup, tetapi modelnya lebih dipengaruhi oleh kebutuhan untuk mengembangkan sistem yang lebih cepat, yang bisa dicapai dengan meningkatkan siklus hidup dan menggunakan peralatan pengembangan berbasis komputer (Wahyudi, 2018). Terlihat pada Gambar 1 dibawah ini



**Gambar 1.** Siklus SDLC (System Development Life Cycle)

### 3. HASIL DAN PEMBAHASAN

Metode pengembangan Sistem yang digunakan dalam Migrasi Database MySQL ke PostgreSQL pada Aplikasi Sistem Evaluasi Dosen Oleh Mahasiswa Jurusan Ekonomi dan Bisnis yaitu metode SDLC. Tahapan yang dilakukan pada metode SDLC dalam pengembangan sistem yaitu dengan tahap perencanaan, analisis, desain, implementasi.

#### 1. Perencanaan

Berdasarkan studi literatur yang dilakukan terkait jurnal yang membahas tentang migrasi database MySQL ke PostgreSQL, perancangan database baru sebagai target migrasi diperlukan guna memperbaiki performa Aplikasi Sistem EDOM Jurusan Ekonomi dan Bisnis. Seperti yang dijelaskan sebelumnya bahwa seiring bertambahnya jumlah data pada Aplikasi Sistem EDOM, MySQL sudah tidak mampu memberikan performa terbaiknya dalam menangani jumlah data yang besar, sehingga diperlukan database yang memiliki kinerja lebih baik lagi seperti PostgreSQL. Pada Aplikasi Sistem EDOM terdapat banyak penyajian data yang menggunakan klausa join, PostgreSQL mengalahkan MySQL untuk query dengan klausa *join*.

#### 2. Analisis

##### a. Analisis Proses Migrasi

Tahap analisis pada proses migrasi database adalah proses untuk mengkonversi data dari satu database ke database yang lain dalam *environment* yang sama maupun *environment* yang berbeda. *Environment* yang dimaksud adalah *hardware* dan *software* yang digunakan bersamaan dengan database. Di dalam proses ini, tidak hanya data saja yang dipindahkan, tetapi juga objek-objek yang di dalam database tersebut termasuk tabel, *view*, *trigger*, *stored procedure*, dan objek-objek lainnya. Alasan mengapa melakukan proses migrasi database adalah sebagai berikut:

1. Database PostgreSQL memiliki penyimpanan yang lebih besar dibanding MySQL, sesuai dengan kebutuhan pada Aplikasi Sistem EDOM
2. Database PostgreSQL mengalahkan MySQL untuk query dengan klausa *join* yang mana pada Aplikasi Sistem EDOM terdapat banyak penyajian data menggunakan klausa *join*.

3. Aplikasi Sistem EDOM menjadi lambat karena ukuran database sudah terlalu besar sehingga proses bisnis pun menjadi lambat

Proses migrasi dilakukan dengan beberapa tahapan yang harus dilakukan seperti penyesuaian tipe data antara database lama (MySQL) ke database yang baru (PostgreSQL) begitu pula dengan *environment* database seperti *hardware* dan *software* yang bekerja sama dengan database tersebut.

- b. Replikasi

Proses replikasi database adalah proses menggandakan semua perubahan yang terjadi pada suatu database. Dalam proses ini dilakukan membuat replika dari database yang sedang dipakai pada Aplikasi Sistem EDOM sebagai backup bilamana terjadi kegagalan pada sistem baik yang disengaja maupun yang tidak, agar tetap memiliki salinan dari server utama untuk menjalankan *reporting* dan analisis kerja tanpa mengganggu jalannya proses bisnis pada Aplikasi Sistem EDOM. Beberapa alasan melakukan replikasi pada database utama adalah sebagai berikut:

1. Dengan melakukan replikasi, ketersediaan data sehingga proses bisnis pun lebih terjamin.
2. Memiliki *backup* yang dapat berguna bilamana terjadi kegagalan yang disengaja maupun yang tak disengaja.

Sama halnya seperti proses migrasi, proses replikasi juga memerlukan penyesuaian antara database lama dengan database yang baru. Apalagi jika *environment*, tempat database lama dan database baru beroperasi berbeda. Perlu adanya konfigurasi khusus pada kedua database tersebut agar proses replikasi bisa berjalan lancar.

3. Desain

Pada tahap ini dibuat rancangan perbandingan database lama dan yang baru berdasarkan analisis yang telah dilakukan pada tahap sebelumnya. Aplikasi Sistem EDOM mempunyai 14 tabel diantaranya:

**Tabel 1.** Tabel jadwal kuliah

Struktur tabel jadwalkuliah			
	MySQL		PostgreSQL
id_jadwal	Int(11) PK	id_jadwal	Integer PK
tahunajaran_id	Int(11)	tahunajaran_id	Integer
jurusan_id	Varchar(3)	jurusan_id	character varying (3)
prodi_id	Varchar(5)	prodi_id	character varying (5)
semester_id	Varchar(4)	semester_id	character varying (4)
KodeKelas	Varchar(20)	kodekelas	character varying (20)
Ruang_id	Varchar(6)	ruang_id	character varying (6)
KodeMk	Varchar(10)	kode_mk	character varying (10)
sks	Varchar(2)	sks	character varying (2)
keterangan_jadwal	Enum('teori', 'praktik')	keterangan_jadwal	Enum('teori', 'praktik')
hari	Varchar(10)	hari	character varying (10)
jam_mulai	time	jam_mulai	Time without time zone
jam_selesai	time	jam_selesai	Time without time zone
dosenpi_nip	Varchar(20)	dosenpi_nip	character varying (20)
dosenanggota1_nip	Varchar(20)	dosenanggota1_nip	character varying (20)
dosenanggota2_nip	Varchar(20)	dosenanggota2_nip	character varying (20)
dosenanggota3_nip	Varchar(20)	dosenanggota3_nip	character varying (20)
teknisi1_id	Varchar(20)	teknisi1_id	character varying (20)
teknisi2_id	Varchar(20)	teknisi2_id	character varying (20)
teknisi3_id	Varchar(20)	teknisi3_id	character varying (20)
teknisi4_id	Varchar(20)	teknisi4_id	character varying (20)
status_jadwal	Enum('Aktif', 'Tidak')	status_jadwal	Enum('Aktif', 'Tidak')

Tabel 2. Tabel dosen

Struktur tabel dosen			
	MySQL		PostgreSQL
NIP	Varchar(18) PK	nip	character varying (18) PK
NamaDosen	Varchar(50)	namadosen	character varying (50)
jurusan_id	Varchar(6)	jurusan_id	character varying (6)
Prodi_id	Varchar(6)	prodi_id	character varying (6)
ttd	Varchar(50)	ttd	character varying (50)
email	Varchar(50)	email	character varying (50)
publish	Enum('T', 'F')	publish	Enum('T', 'F')

Tabel 3. Tabel masterkelas

Struktur tabel masterkelas			
	MySQL		PostgreSQL
masterkelas_id	Int(11) PK	masterkelas_id	Integer PK
KodeKelas	Varchar(35)	kodekelas	character varying (35)
KodeProdi	Varchar(7)	kodeprodi	character varying (7)
TahunAjaran	Varchar(20)	tahunajaran	character varying (20)
Semester	Varchar(1)	semester	character varying (1)
Kelompok	Enum('A', 'B', 'C', 'D')	kelompok	Enum('A', 'B', 'C', 'D')
Status	Enum('Aktif', 'Nonaktif')	status	Enum('Aktif', 'Nonaktif')

Tabel 4. Table Kelas

Struktur tabel kelas			
	MySQL		PostgreSQL
kelas_id	Int(11) PK	kelas_id	Integer PK
nama_kelas	Varchar(255)	nama_kelas	character varying (255)

Tabel 5. Tabel detail\_kelas

Struktur tabel detail_kelas			
	MySQL		PostgreSQL
detail_id	Int(11) PK	detail_id	Integer PK
KodeKelas	Varchar(35)	kodekelas	character varying (35)
npmmahasiswa	Varchar(10)	npmmahasiswa	character varying (10)

**Tabel 6.** Tabel matakuliah

Struktur tabel matakuliah			
	MySQL		PostgreSQL
Matakuliah_id	Int(11) PK	matakuliah_id	Integer PK
KodeProdi	Varchar(10)	kodeprodi	character varying (10)
KodeMk	Varchar(10)	kodemk	character varying (10)
NamaMK	Varchar(100)	namamk	character varying (100)
SksTeori	Int(1)	sksteori	Integer
SksPraktik	Int(1)	skspraktik	Integer
Semester	Varchar(4)	semester	character varying (4)
Dasar	Varchar(50)	dasar	character varying (50)
link	Varchar(100)	link	character varying (100)

**Tabel 7.** Tabel programstudi

Struktur tabel programstudi			
	MySQL		PostgreSQL
KodeJurusan	Varchar (10) PK	kodejurusan	character varying (10)PK
KodeProdi	Varchar(10)	kodeprodi	character varying (10)
NamaProdi	Varchar(50)	namaprodi	character varying (50)
Jenjang	Varchar(2)	jenjang	character varying (2)
TahunBerdiri	Varchar(4)	tahunberdiri	character varying (4)
SKPendirian	Varchar(20)	skpendirian	character varying (20)
PictureType	Varchar(50)	picturetype	character varying (50)
PictureName	Varchar(50)	picturename	character varying (50)

**Tabel 8.** Mahasiswa

Struktur tabel mahasiswa			
	MySQL		PostgreSQL
Npm	int (11) PK	Npm	integer PK
nama	Varchar(255)	nama	character varying (255)
email	Varchar(255)	email	character varying (255)
jenis_kelamin	Enum('l', 'p')	jenis_kelamin	Enum('l', 'p')
no_hp	Varchar(16)	no_hp	character varying (16)
tanggal_lahir	date	tanggal_lahir	date
tempat_lahir	Varchar(255)	tempat_lahir	character varying (255)
foto	Varchar(255)	foto	character varying (255)
alamat	Varchar(255)	alamat	character varying (255)
asal_sekolah	Varchar(255)	asal_sekolah	character varying (255)
tahun_lulus_sma	Varchar(4)	tahun_lulus_sma	character varying (4)
type_sma	Enum('SMA', 'MA', 'SMK', 'Paket')	type_sma	Enum('SMA', 'MA', 'SMK', 'Paket')
jurusan_sma	Varchar(255)	jurusan_sma	character varying (255)
nama_ayah	Varchar(255)	nama_ayah	character varying (255)
pekerjaan_ayah	int (11)	pekerjaan_ayah	integer
penghasilan_ayah	int (11)	penghasilan_ayah	integer
no_hp_ayah	Varchar(16)	no_hp_ayah	character varying (255)
pekerjaan_ibu	int (11)	pekerjaan_ibu	integer
penghasilan_ibu	int (11)	penghasilan_ibu	integer
nama_ibu	Varchar(255)	nama_ibu	character varying (255)
no_hp_ibu	Varchar(16)	no_hp_ibu	character varying (16)
tahun_masuk	Varchar(4)	tahun_masuk	character varying (4)
jalur_masuk_id	Varchar(11)	jalur_masuk_id	character varying (11)
pembiayaan_id	Varchar(11)	pembiayaan_id	character varying (11)
jurusan_id	Varchar(10)	jurusan_id	character varying (10)
prodi_id	Varchar(10)	prodi_id	character varying (10)
created_at	Timestamp	created_at	Timestap with time zone
update_at	Timestamp	update_at	Timestamp with time zone
provinsi_id	int (11)	provinsi_id	integer
kabupaten_id	int (11)	kabupaten_id	integer
kecamatan_id	int (11)	kecamatan_id	integer
keluarahan_id	int (11)	keluarahan_id	character varying (50)
status	Enum('Aktif', 'Cuti', 'DO', 'Lulus')	status	

**Tabel 9.** Table Tahun Ajaran

Struktur tabel tahunajaran			
	MySQL		PostgreSQL
Id	Int(11) PK	id	Integer PK
TahunAjaran	Varchar(20)	tahunajaran	character varying (20)
Aktif	Enum('Y', 'N')	aktif	Enum('Y', 'N')

**Tabel 10.** Tabel semester

Struktur tabel semester			
	MySQL		PostgreSQL
Id	Int(11) PK	id	Integer PK
Semester	Varchar(20)	semester	character varying (20)
Aktif	Enum('Y', 'N')	aktif	Enum('Y', 'N')

**Tabel 11.** Tabel versi kuesioner

Struktur tabel quesversi			
	MySQL		PostgreSQL
versiid	Int(11) PK	versiid	Integer PK
nama_versi	Varchar(50)	nama_versi	character varying (50)
keterangan	Varchar(50)	keterangan	character varying (50)
enable	Enum('ves', 'no')	enable	Enum('Y', 'N')

**Tabel 12.** Tabel kategori kuesioner

Struktur tabel queskategori			
	MySQL		PostgreSQL
quesKat_id	Int(11) PK	queskat_id	Integer PK
versiid	int(3)	versiid	Integer
namaKategori	Varchar(50)	namakategori	character varying (50)
enable	Enum('yes', 'no')	enable	Enum('yes', 'no')

**Tabel 13.** Tabel detail kuisioner

Struktur tabel quesdetail			
	MySQL		PostgreSQL
quesDetail_id	Int(11) PK	quesdetail_id	Integer PK
quesKat_id	int(11)	queskat_id	Integer
namaKategori	Varchar(250)	namakategori	character varying (250)

**Tabel 14.** Tabel kuesioner master

Struktur tabel quesmaster			
	MySQL		PostgreSQL
idquesmaster	Int(11) PK	idquesmaster	Integer PK
tahunajaran	Varchar(50)	tahunajaran	character varying (50)
semester	Varchar(4)	semester	character varying (4)
prodi_id	Varchar(5)	prodi_id	character varying (5)
KodeMk	Varchar(10)	kodemk	character varying (10)
npm	Varchar(10)	npm	character varying (10)
KodeKelas	Varchar(35)	kodekelas	character varying (35)
NIP	Varchar(20)	nip	character varying (20)
tgl	date	tgl	date

**Tabel 15.** Tabel kuesioner master detail

Struktur tabel quesmasterdetail			
	MySQL		PostgreSQL
idquesmaster	Int(11) PK	idquesmaster	Integer PK
idquesmaster	Int(11)	idquesmaster	Integer
detailid	Int(11)	detailid	Integer
nilai	Int(11)	nilai	Integer

Berdasarkan perbandingan design tabel di atas ada sedikit perbedaan tipe data dan penggunaan huruf pada nama kolom. Tipe data yang terdapat pada MySQL tidak semuanya dapat berjalan normal ketika diimplementasikan ke PostgreSQL, walaupun PostgreSQL hampir mendukung semua tipe data dan lebih kaya dibandingkan MySQL, namun ada perbedaan nama dan karakteristik pada beberapa tipe data. Penggunaan tipe data *enum* pada PostgreSQL harus didaftarkan dahulu untuk menentukan tipe data kita sendiri sedangkan kita bisa langsung menggunakan tipe data *enum* pada MySQL.

#### 4. Impelemntasi

##### a. Tahap Migrasi Database

##### 1. Export

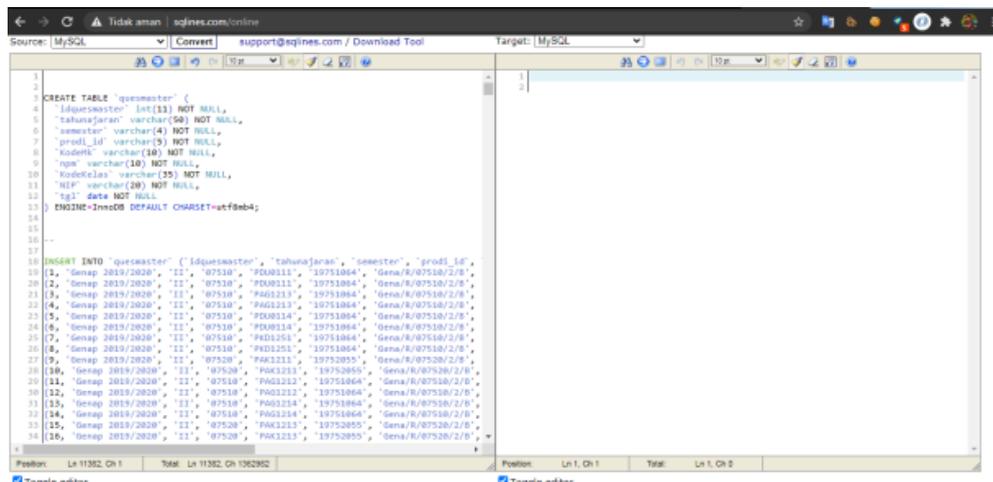
Export data pada database MySQL adalah proses backup untuk memperoleh file database MySQL dalam format yang diinginkan (SQL, CSV, DII). Proses Export terlebih dahulu dilakukan dengan memilih table yang diinginkan.



Gambar 2. Export database MySQL

##### 2. Proses Migrasi

File yang berhasil diexport akan menghasilkan query DDL dan DML. Kemudian query tersebut dipindah ke dalam text editor pada tool sqllines.com, dengan source yang dipakai adalah MySQL dan target yang dipakai adalah PostgreSQL, klik convert untuk mengeksekusi proses konversi database.



Gambar 3. Konversi database MySQL ke PostgreSQL

### 3. Import

Query yang berhasil dikonversi kemudian dijalankan pada Tool database PostgreSQL (PgAdmin)

```

Query Editor  Query History
38 CREATE TABLE quesmaster (
39     idquesmaster int NOT NULL,
40     tahunajaran varchar(50) NOT NULL,
41     semester varchar(4) NOT NULL,
42     prodi_id varchar(5) NOT NULL,
43     KodeMk varchar(10) NOT NULL,
44     npm varchar(10) NOT NULL,
45     KodeKelas varchar(35) NOT NULL,
46     NIP varchar(20) NOT NULL,
47     tgl date NOT NULL
48 );
49
50 ALTER TABLE quesmaster
51     ADD PRIMARY KEY (idquesmaster);
52
53 CREATE SEQUENCE sec_quesmaster;
54 ALTER TABLE quesmaster ALTER COLUMN idquesmaster SET DEFAULT NEXTVAL('sec_quesmaster');
55
56 CREATE UNIQUE INDEX CONCURRENTLY unique_quesmaster
57 ON quesmaster (KodeKelas, tahunajaran, semester, prodi_id, KodeMk, npm, NIP);
58
59 INSERT INTO quesmaster (idquesmaster, tahunajaran, semester, prodi_id, KodeMk, npm, KodeKelas, NIP, tgl) VALUES
60 (1, 'Genap 2019/2020', 'II', '07510', 'PDU0111', '19751064', 'Gena/R/07510/2/B', '197703132015041001', '2020-06-12');

```

**Gambar 4.** Proses import database PostgreSQL

#### b. Pengujian Kinerja Query Pada DBMS

Dalam pengujian kinerja *query* database MySQL dan PostgreSQL dilakukan dengan mengimplementasikan (CRUD) pada masing-masing DBMS dengan jumlah data yang sama yaitu 1, 100, dan 100 dilakukan 3 kali percobaan *query*. Selanjutnya untuk mendapatkan jumlah hasil kecepatan dilakukan perhitungan menggunakan rumus pengujian *query*, kemudian akan terlihat selisih waktu eksekusi *query* pada kedua DBMS sehingga kecepatan *query* pada masing-masing DBMS dapat dianalisa

##### 1. Insert

*Query insert* berguna untuk memasukan atau menyimpan data baru ke dalam database.

Adapun contoh *query insert* adalah sebagai berikut:

```

INSERT INTO `quesmaster` (`idquesmaster`, `tahunajaran`, `semester`, `prodi_id`,
`KodeMk`, `npm`, `KodeKelas`, `NIP`, `tgl`) VALUES (1, 'Genap 2019/2020', 'II',
'07510', 'PDU0111', '19751064', 'Gena/R/07510/2/B', '197703132015041001', '2020-06-12');

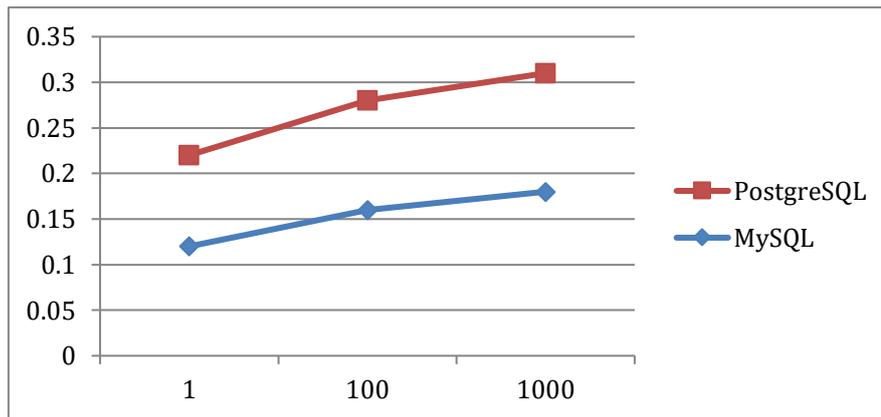
```

**Gambar 5.** Query Insert pada MySQL

Dilakukan perbandingan *insert* 1000 data pada masing masing DBMS.

**Tabel 16.** Perbandingan Insert MySQL dan PostgreSQL

Database	1	100	1000
MySQL (sec)	0.12	0.16	0.18
PostgreSQL (sec)	0.10	0.12	0.13



Gambar 6. Grafik pengujian insert

2. Update

Query Update berguna untuk merubah data yang ada pada database. Adapun contoh query update adalah sebagai berikut:

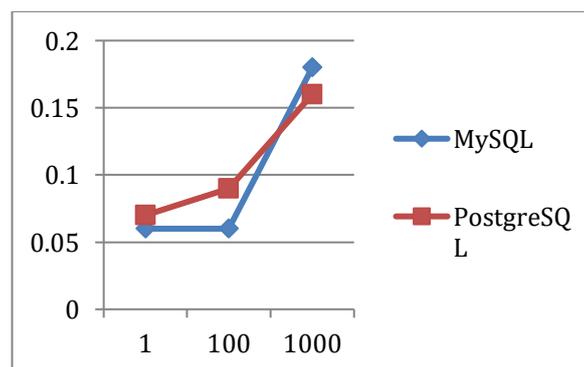
```
UPDATE `quesmaster` SET `idquesmaster`= 'III' WHERE idquesmaster = '1'
```

Gambar 7. Query update pada PostgreSQL

Dilakukan perbandingan Update sebuah data pada masing-masing DBMS.

Tabel 17. Perbandingan Update MySQL dan PostgreSQL

Database	1	100	1000
MySQL (sec)	0.06	0.06	0.18 Sec
PostgreSQL (sec)	0.07	0.09	0.16



Gambar 8. Grafik pengujian update

3. Read

Query Read berguna untuk menampilkan data yang ada pada database. Adapun contoh *query read* adalah sebagai berikut:

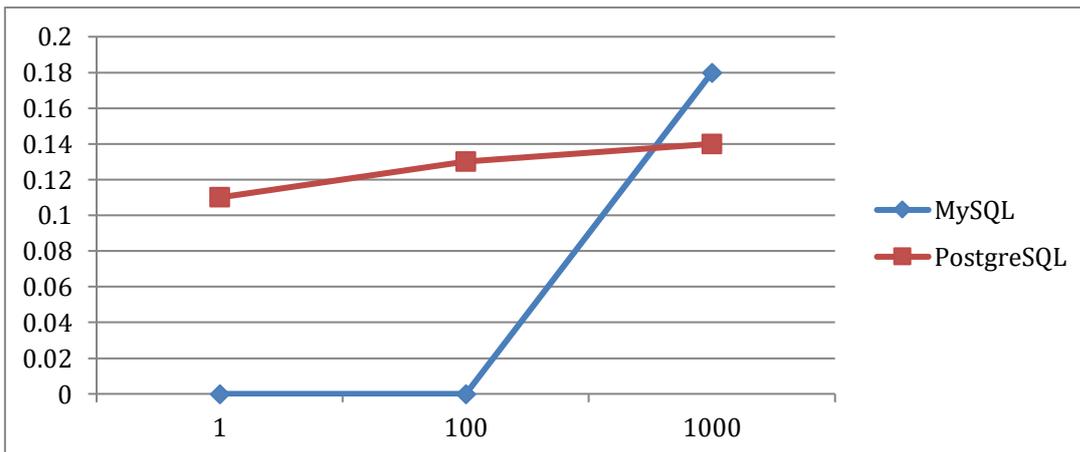
```
SELECT * FROM `quesmaster` WHERE LIMIT 1000
```

Gambar 9. Query read 1000 tabel pada PostgreSQL

Pada pengujian ini dilakukan Read 1000 data yang sama pada masing-masing DBMS

Tabel 18. Perbandingan Read MySQL dan PostgreSQL

Database	1	100	1000
MySQL (sec)	0.00	0.00	0.18
PostgreSQL (sec)	0.11	0.13	0.14



Gambar 10. Grafik pengujian read

4. Delete

Query delete berguna untuk menghapus data yang ada pada database. Adapun contoh *query delete* adalah sebagai berikut:

```
Delete * FROM `quesmaster`
```

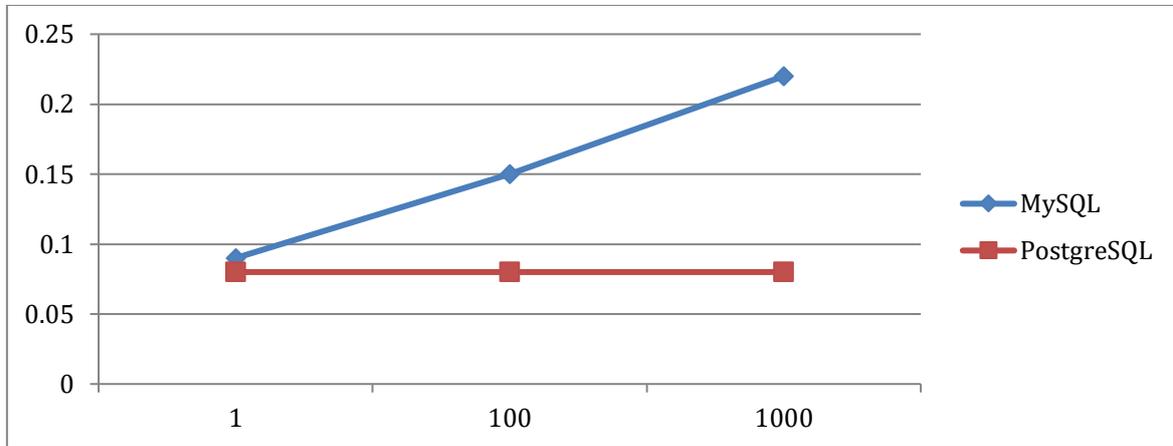
Gambar 11. Query delete tabel pada PostgreSQL

Pada pengujian ini dilakukan penghapusan 1000 data pada masing

Tabel 19. Perbandingan Delete MySQL dan PostgreSQL.

Database	1	100	1000
----------	---	-----	------

MySQL	0.09	0.15	0.22
PostgreSQL	0.08	0.08	0.08



**Gambar 12.** Grafik pengujian delete

## 5. JOIN

*Query Join* merupakan gabungan dari beberapa tabel yang memiliki kolom yang sama. Adapun contoh query join adalah sebagai berikut:

```
SELECT
AVG(quesmasterdetail.nilai) as ratapoin, dosen>NamaDosen, quesmaster.tahunajaran,
quesmaster.NIP, quesmaster.idquesmaster, COUNT(DISTINCT quesmaster.npm) as jumlahresponden,
quesmaster.tahunajaran, quesmaster.prodi_id, programstudi>NamaProdi
FROM quesmasterdetail
JOIN quesmaster
    ON quesmasterdetail.idquesmaster = quesmaster.idquesmaster
JOIN dosen
    ON quesmaster.NIP = dosen.NIP
JOIN programstudi
    ON quesmaster.prodi_id = programstudi.KodeProdi
WHERE programstudi.KodeJurusan = '$jur'
GROUP BY quesmaster.NIP, quesmaster.prodi_id, quesmaster.tahunajaran
```

**Gambar 13.** Query join tabel pada PostgreSQL

Pada pengujian ini dilakukan *join* dari 4 tabel dengan data yang sama pada masing masing DBMS. Terdapat fungsi aggregate *avg* dan juga *count*, dan klausa pengelompokan (*group by*) berdasarkan 3 kolom

✓ Menampilkan baris 0 - 24 (total 139, Pencarian dilakukan dalam 107,0000 detik.)

**Gambar 14.** Kinerja Join pada MySQL



5	Destika Ahyani S.P., M.S. Agribisnis Pangan Genap 2019/2020	33	4.89	4.90	4.88	4.87	4.88
6	Nurmalia S.E., M.M., M.A., CA Akuntansi Genap 2019/2020	70	4.80	4.91	4.89	4.89	4.86
7	Ii Teguh Budi Triyanto, M.Si Agribisnis Pangan Genap 2019/2020	32	4.82	4.87	4.89	4.88	4.85
8	Depta Anggraini S.E., M.SiAk. Akuntansi Genap 2019/2020	32	4.84	4.87	4.83	4.88	4.85
9	Lihan Rini Puspita Wijaya S.E., M.Si. Akuntansi Genap 2019/2020	97	4.82	4.87	4.88	4.88	4.85
10	Dr. Imayanti Nover S.P., M.Si. Agribisnis Pangan Genap 2019/2020	33	4.79	4.89	4.81	4.90	4.84

Showing 1 to 10 of 139 entries

Page generated in : 443521.37 -ms

**Gambar 17.** Kode pengujian pada aplikasi menggunakan dbms MySQL

5	Destika Ahyani S.P., M.Si Agribisnis Pangan Genap 2019/2020	33	4.89	4.90	4.88	4.87	4.88
6	Nurmalia S.E., M.M., M.A., CA Akuntansi Genap 2019/2020	70	4.80	4.91	4.89	4.89	4.86
7	Ii Teguh Budi Triyanto, M.Si Agribisnis Pangan Genap 2019/2020	32	4.82	4.87	4.89	4.88	4.85
8	Depta Anggraini S.E., M.SiAk. Akuntansi Genap 2019/2020	32	4.84	4.87	4.83	4.88	4.85
9	Lihan Rini Puspita Wijaya S.E., M.Si. Akuntansi Genap 2019/2020	97	4.82	4.87	4.88	4.88	4.85
10	Dr. Imayanti Nover S.P., M.Si. Agribisnis Pangan Genap 2019/2020	33	4.79	4.89	4.81	4.90	4.84

Showing 1 to 10 of 139 entries

Page generated in : 2118.12 -ms

**Gambar 18.** Kode pengujian pada aplikasi menggunakan dbms PostgreSQL

Melalui pengujian tersebut dihasilkan nilai 2118.12 -ms pada aplikasi yang menggunakan postgresQL dan 443521.37 -ms pada aplikasi yang menggunakan MySQL. Perbandingan yang dihasilkan cukup signifikan, untuk kebutuhan pada Aplikasi Sistem Edom yang menjadi salah satu keuntungan dalam dilakukannya migrasi database karena proses bisnis yang dilakukan menjadi lebih efektif.

#### d. Penyesuaian Aplikasi Pada Database PostgreSQL

##### 1. Case insensitive

Penggunaan huruf besar kecil pada beberapa kolom tabel di database sebelumnya (MySQL) ditujukan untuk mempermudah dalam mengidentifikasi nama tabel, namun saat dikonversi ke database yang baru (PostgreSQL) terjadi perubahan, penggunaan huruf besar berubah jadi kecil, sehingga terjadi penyesuaian nama kolom pada *query* pada program dan juga nama kolom yang masih menggunakan huruf besar diganti jadi huruf kecil.

PostgreSQL pada penggunaan nama entitas (tabel, tampilan, prosedur, dll) yang tidak dikutip diartikan menjadi *case-insensitive* artinya huruf besar dan kecil diartikan sama. Untuk membuat *case-sensitive* yang artinya penamaan huruf besar atau perpaduan huruf besar dan kecil dianggap berbeda, harus diberi tanda kutip untuk mengidentifikasi *case-sensitive*.

```

if ($this->input->post('NIP1')) {
    $data = [
        "tahunajaran" => $this->input->post('tahunajaran'),
        "semester" => $this->input->post('semester'),
        "prodi_id" => $this->input->post('prodi_id'),
        "KodeMk" => $this->input->post('KodeMk'),
        "npm" => $this->input->post('npm'),
        "KodeKelas" => $this->input->post('KodeKelas'),
        "NIP" => $this->input->post('NIP1'),
        "tgl" => date('Y/m/d', time())
    ];
    $quesDetail_ida = $this->input->post('quesDetail_ida');
    $quesa = $this->input->post('quesa');
    $jumlah = count($quesDetail_ida);
    $this->db->trans_begin();
    $query = $this->db->insert('quesmaster', $data);
    $this->db->trans_complete();
    if ($this->db->trans_status() == FALSE) {
        $this->db->trans_rollback();
    } else {
        $this->db->trans_commit();
    }
}

```

Gambar 19. Aplikasi Sistem EDOM sebelum migrasi database

```

if ($this->input->post('NIP1')) {
    $data = [
        "tahunajaran" => $this->input->post('tahunajaran'),
        "semester" => $this->input->post('semester'),
        "prodi_id" => $this->input->post('prodi_id'),
        "kodeMk" => $this->input->post('KodeMk'),
        "npm" => $this->input->post('npm'),
        "kodekelas" => $this->input->post('KodeKelas'),
        "nip" => $this->input->post('NIP1'),
        "tgl" => date('Y/m/d', time())
    ];
    $quesDetail_ida = $this->input->post('quesDetail_ida');
    $quesa = $this->input->post('quesa');
    $jumlah = count($quesDetail_ida);
    $this->db2->trans_begin();
    $query = $this->db2->insert('quesmaster', $data);
    $this->db2->trans_complete();
    if ($this->db2->trans_status() == FALSE) {
        $this->db2->trans_rollback();
    } else {
        $this->db2->trans_commit();
    }
}

```

Gambar 20. Aplikasi Sistem EDOM setelah migrasi database

## 2. Query

Pada Sistem Aplikasi Sistem Evaluasi Dosen Oleh Mahasiswa banyak digunakan penyajian data yang memungkinkan untuk mengelompokan data. Pada database MySQL penggunaan *group by* bisa dilakukan tanpa adanya fungsi *aggregate* namun pada postgresQL tidak dapat dilakukan seperti itu. Pada postgresQL harus dilakukan menggunakan klausa *distinct*, karena *group by* difungsikan untuk fungsi *aggregate*. Pada penyesuaian *query* dilakukan untuk mengganti *query* pada MySQL yang tidak dapat dijalankan pada PostgreSQL

```

public function kajur_detail()
{
    $jur = $this->session->userdata('jurusan_id');
    $query = "
    SELECT AVG(quesmasterdetail.nilai) as ratapoin,
    dosen>NamaDosen, quesmaster.tahunajaran, quesmaster.NIP,
    quesmaster.idquesmaster, count(DISTINCT quesmaster.npm) as jumlahresponden,
    quesmaster.tahunajaran, quesmaster.prodi_id, programstudi>NamaProdi
    FROM quesmasterdetail
    JOIN quesmaster
    ON quesmasterdetail.idquesmaster = quesmaster.idquesmaster
    JOIN dosen
    ON quesmaster.NIP = dosen.NIP
    JOIN programstudi
    ON quesmaster.prodi_id = programstudi.KodeProdi
    WHERE programstudi.KodeJurusan = '$jur'
    GROUP BY quesmaster.NIP, quesmaster.prodi_id, quesmaster.tahunajaran
    ORDER BY AVG(quesmasterdetail.nilai) DESC
    ";
    return $this->db->query($query)->result_array();
}

```

**Gambar 21.** Query Aplikasi Sistem EDOM sebelum migrasi database

```

public function kajur_detail()
{
    $jur = $this->session->userdata('jurusan_id');
    $query = "
    SELECT DISTINCT ON (quesmaster.nip, quesmaster.prodi_id, quesmaster.tahunajaran)
    quesmaster.nip, quesmaster.prodi_id, quesmaster.tahunajaran,
    AVG(quesmasterdetail.nilai) OVER(PARTITION BY quesmaster.nip,
    quesmaster.prodi_id, quesmaster.tahunajaran) ratapoin,
    dosen.namadosen, quesmaster.prodi_id, programstudi.namaprodi
    FROM quesmasterdetail
    JOIN quesmaster
    ON quesmasterdetail.idquesmaster = quesmaster.idquesmaster
    JOIN dosen
    ON quesmaster.nip = dosen.nip
    JOIN programstudi
    ON quesmaster.prodi_id = programstudi.kodeProdi
    WHERE programstudi.kodejurusan = '$jur'
    ";
    return $this->db2->query($query)->result_array();
}

```

**Gambar 22.** Query Aplikasi Sistem EDOM setelah migrasi database

#### 4. KESIMPULAN

Kesimpulan dari tugas akhir ini adalah telah berhasil dimigrasikan database dari MySQL ke PostgreSQL yang digunakan untuk menyimpan data pada Aplikasi Sistem Evaluasi Dosen Oleh Mahasiswa (EDOM) Jurusan Ekonomi dan Bisnis. Tujuan dari dilakukannya Migrasi database MySQL ke PostgreSQL untuk mengatasi masalah pada database MySQL yang sudah tidak mampu memberikan performa terbaiknya dalam menangani jumlah data yang besar.

#### DAFTAR PUSTAKA

- Andi, J. (2015). Pembangunan Aplikasi Child Tracker Berbasis Assisted – Global Positioning System ( A-GPS ) Dengan Platform Android. *Jurnal Ilmiah Komputer Dan Informatika (KOMPUTA)*, 1(1), 1-8.
- Bafadal, M. M., Basisdata, M., Sistem, P., Akademik, I., & Bafadal, M. M. (2012). *Migrasi basisdata pada sistem informasi akademik universitas tanjungpura*. 2(2), 71-80.
- Harisantoso, J. (2012). Pengukuran Kinerja Dosen Melalui EKD (Evaluasi Kinerja Dosen) STKIP PGRI Situbondo Berdasarkan Persepsi Mahasiswa. *SAINTEKS Universitas Muhammadiyah Purwokerto*, 9(2), 64-71.
- Lesmana, I., & Irawan, D. (2018). Aplikasi Penilaian Kinerja Dosen Berbasis Android. *Nuansa Informatika*, 12(2). <https://doi.org/10.25134/nuansa.v12i2.1355>
- Nento, S. (2018). Analisis Kompetensi Profesional dan Kinerja Dosen. *Jurnal Ilmiah Iqra'*, 6(1). <https://doi.org/10.30984/jii.v6i1.619>

- 
- Palit, R. V, Rindengan, Y. D. Y., & Lumenta, A. S. M. (2015). Rancangan Sistem Informasi Keuangan Gereja Berbasis Web Di Jemaat Gmim Bukit Moria Malalayang. *Jurnal Teknik Elektro Dan Komputer*, 4(7), 1-7.
- Pradiyawati, N. L. D., Kenali, E. W., & Zuriati. (2017). *telah dirancang serangkaian diimplementasikan dengan program pengkodean dengan dreamweaver , bahasa pemograman PHP dan javascript serta menggunakan aplikasi implementasi notepad ++ , sublime teks3 dan MySQL.*
- Putri Andhikha, I. (2017). Implementasi Dan Analisis Kinerja Mysql Cluster Menggunakan Metode Load Balancing. *Jurnal Manajemen Informatika*, 8(1).
- Rochsas, J. F., Putra, S. D., & Sahlinal, D. (2017). *Aplikasi Monitoring Sprint Backlog Proyek Pada PT . Eannovate Creative Technology Berbasis Web.*
- Sutanta, E. (2004). *Sistem Basis Data*. xxviii+354.
- Wahyudi, A. (2018). Perancangan sistem menggunakan metode sdlc. *Jurnal Dinamika Informatika*, 4(2), 1-11.
- Warman, I., & Ramdaniansyah, R. (2018). Analisis Perbandingan Kinerja Query Database Management System (Dbms) Antara Mysql 5.7.16 Dan Mariadb 10.1. *Jurnal Teknoif*, 6(1), 32-41.
- Wibowo, A. T., & Wiguna, A. S. (2019). Pemanfaatan Teknologi Single Page Application (SPA) dalam Pembuatan Aplikasi Feedback Dosen dari Mahasiswa Sebagai Bentuk Pengawasan Lembaga Terhadap Kinerja Dosen di Bidang Pengajaran. *SMARTICS Journal*, 5(1), 34-43.