

# Perbandingan Tingkat Efisiensi Waktu *Query SELECT* pada *Database Interface* Navicat dan SQLYog di MySQL DBMS

Ahnaf Hadi Fathulloh<sup>1</sup>, Hofifah Isma Adauwiyah<sup>2</sup>

**Abstrak**—Memudahkan dalam mengoperasikan database MySQL tentunya diperlukan aplikasi pendukung yang memiliki query response time yang cepat agar dapat mengefisienkan waktu. Pada penelitian ini dilakukan pengukuran performa query select pada database MySQL dengan aplikasi database Navicat dan juga SQLYog dengan jumlah record data 50, 100, 500, 1000, dan 5000. Query yang diuji adalah DML (Data Manipulation Language), yaitu select. Penelitian ini bertujuan untuk memberikan analisis perbandingan efisiensi waktu dari query select yang dilakukan pada aplikasi database Navicat dan SQLYog. Hasil yang diperoleh menunjukkan bahwa query select pada aplikasi database Navicat terbukti mempunyai tingkat efisiensi waktu yang lebih tinggi (response time yang lebih cepat) dibandingkan dengan SQLYog dalam pengujian record data 50 yaitu sebesar 0.0028 detik, pada record data 100 yaitu sebesar 0.0041 detik, pada record data 500 yaitu 0.0153 detik, pada record data 1000 yaitu 0.0274 detik, dan pada record data 5000 yaitu 0.1318 detik.

**Keywords**—Navicat, SQLYog, Query, Efisiensi Waktu, MySQL.

## I. PENDAHULUAN

Seiring dengan pesatnya kemajuan teknologi, penggunaan *database* adalah hal yang lumrah, ini dikarenakan *database* adalah komponen penting bagi suatu organisasi maupun perusahaan dalam menyimpan data dengan jumlah yang sangat banyak, baik di bidang teknologi informasi, bisnis, kedokteran, bahkan hampir seluruh bidang memerlukan *database*. Untuk *manage database*, tentunya dibutuhkan *Database Management System* (DBMS) yang cocok digunakan untuk sebuah perusahaan tertentu.

*Database Management System* sendiri, yaitu sebuah perangkat lunak dengan fungsinya yang terkenal untuk mengelola data yang kapasitasnya sangat besar. Penggunaan

DBMS di zaman sekarang sangat dibutuhkan di berbagai aspek,

baik skala luas, menengah maupun skala kecil. Salah satu jenis DBMS yang banyak diterapkan, yaitu *Relational Database Management System* karena konsepnya yang sederhana

sehingga mudah untuk dipelajari, yang mana di dalam RDBMS berisi tabel-tabel yang saling memiliki hubungan dan terkoneksi melalui *primary key*. RDBMS mulai digunakan dari tahun 1970-an karena RDBMS memiliki syarat-syarat penting seperti *confidentiality*, *integrity*, serta *availability*. Dan juga transaksi di RDBMS sifatnya *Atomic*, *Consistency*, *Isolation*, dan *Durability* atau yang disingkat dengan ACID [1]. Salah satu contoh dari RDBMS, yaitu MySQL. Untuk memudahkan dalam mengoperasikan *database* MySQL diperlukan aplikasi pendukung yang memiliki *query response time* yang cepat sehingga dapat mengefisienkan waktu. Saat ini, sudah banyak aplikasi untuk memudahkan dalam mengoperasikan *database* MySQL, contohnya seperti phpMyAdmin, MySQL Workbench, SQLYog, Valentina Studio, Navicat, MySQL Front, dan lain-lain.

Berdasarkan permasalahan tersebut, maka akan dilakukan pengujian terhadap perbandingan tingkat efisiensi waktu *query select* antara aplikasi *database* Navicat dan SQLYog.

## II. KAJIAN PENELITIAN

### A. *Database dan DBMS*

Basis data, yaitu kumpulan dari berbagai data yang tersimpan dan tersusun dengan teratur di dalam komputer agar dapat dikelola dengan suatu *software* sehingga menghasilkan informasi yang berguna bagi penggunaannya. Jadi, kehadiran basis data di sini sangatlah penting sebagai tempat atau gudangnya data-data yang ada untuk kemudian diolah lebih lanjut. Dengan adanya basis data, kita dapat mengelompokkan

Received: 27 November 2020; Revised: 10 July 2021; Accepted: 15 July 2021

<sup>1</sup>A.H Fathulloh, Prodi Sistem Informasi Fakultas Sains dan Teknologi UIN Syarif Hidayatullah Jakarta (e-mail: ahnaf.hadi19@mhs.uinjkt.ac.id).

<sup>2</sup>H.I Adauwiyah, Prodi Sistem Informasi Fakultas Sains dan Teknologi UIN Syarif Hidayatullah Jakarta (e-mail: hofifah.isma19@mhs.uinjkt.ac.id).

data dengan mudah, menghindari adanya data ganda, dan juga menghindari hubungan yang tidak jelas antara satu data dengan data lainnya. Dalam kegiatan pemeliharaan, pengambilan, penginputan suatu data dengan mudah dan efektif tentunya kita membutuhkan perangkat lunak, yaitu *Database Management System (DBMS)*.

*Database Management System* sendiri, yaitu sebuah *software* atau sistem yang dibuat dengan tujuan untuk membantu mengolah basis data dan mengeksekusi suatu operasi apabila diminta oleh banyak pengguna basis data (*database user*). Memastikan supaya basis data selalu terorganisir secara stabil dan bisa diakses dengan mudah adalah tugas dari DBMS yang merupakan sebagai penghubung antara basis data dengan *application program*. Jadi, dapat dikatakan bahwa DBMS akan menangani semua akses ke basis data yang dilakukan oleh *user*. Selain itu, DBMS juga berfungsi sebagai alat untuk mendefinisikan data, melakukan *check* terhadap keamanan dan integritas data yang diterjemahkan oleh DBA (*Database Administrator*), mengatasi hal-hal yang dianggap gagal dalam melakukan akses data, dan lain-lain [2].

#### B. Database SQL (Relasional) dan RDBMS

"relasional" dapat diartikan sebagai tabel-tabel yang saling terkoneksi dalam basis data relasional. Tabel yang dimaksudkan di sini adalah tabel dua dimensi yang terdiri dari baris (tupel) dan kolom (atribut). Tabel-tabel ini dihubungkan dengan *primary key* dan *foreign key*.

*Relational Database Management System*, yaitu program yang melayani *database system*, di mana entitas utamanya adalah tabel-tabel yang saling berelasi. Dalam hal ini, data akan disimpan pada suatu tabel, lalu tabel lain bisa menghubungkan melalui *foreign key* jika terdapat data yang dibutuhkan di tabel tersebut. Dengan adanya RDBMS, apabila ditemukan relasi yang salah terhadap data, RDBMS secara otomatis tidak menerima data itu, sehingga mempermudah *software developer* untuk membuat *coding* karena RDBMS membantu pengecekan tersebut. Jadi, RDBMS bertugas untuk melindungi data-data yang relasinya benar-benar tepat atau berkaitan satu sama lain, baik berupa *primary key* maupun *foreign key* nya. RDBMS memungkinkan untuk menyimpan data-data bagian *field* pada tiap-tiap tabel. RDBMS juga memberikan fasilitas *relational operators* untuk melakukan pembaruan (*updating*) dan juga *delete* informasi yang disimpan ke dalam tabel *database* [3].

#### C. MySQL

MySQL adalah RDBMS sumber terbuka dan memiliki kemampuan untuk mengerjakan pekerjaan secara sekaligus. Pencipta MySQL, yaitu Michael "Monty" Widenius tahun 1995. Di tahun 2000, MySQL diumumkan dengan lisensi ganda, di mana mengizinkan semua orang menggunakannya

secara *free* di bawah Lisensi Publik Umum GNU, sehingga kepopulerannya semakin memuncak. MySQL AB (AB = *aktiebolag*) merupakan perusahaan pemilik dan pengembang MySQL, di mana perusahaan tersebut saat ini menjadi anak perusahaan dari Sun Microsystems. MySQL bisa menjadi basis data sepopuler seperti sekarang adalah bukan karena tanpa alasan, melainkan karena mempunyai kelebihan terhadap fitur-fiturnya yang sangat beragam, salah satu fitur yang paling terkenal adalah masalah *speed* atau kecepatannya. E-Week melakukan perbandingan terhadap sejumlah *database* seperti MySQL, MS SQL, Oracle, Sybase ASE, dan IBM DB2. Dibuktikan, yaitu MySQL dan Oracle memiliki kemampuan terbaik dalam hal *performance* dan *scalability*. MySQL dapat dengan lancar dan cepat mengatasi tabel yang berjumlah puluhan ribu serta *record* data yang berjumlah miliaran.

MySQL menyediakan berbagai *storage machine* yang pastinya memiliki kelebihannya masing-masing. Contohnya, yaitu *storage machine* atau mesin penyimpanan yang mengizinkan mengembalikan data ke kondisi sebelumnya (*rollback*) yang dinamakan *transaction-safe*. Dan masih beragam keunggulan lain yang dipegang oleh MySQL [1],[2]

#### D. Navicat

Navicat adalah aplikasi besutan PremiumSoft CyberTech Ltd yang berfungsi manajemen *database*. Pengguna aplikasi ini bisa menggunakan berbagai jenis *database* seperti (MySQL, Oracle, PostgreSQL, SQLi, dll) dan bisa diimplementasikan di beragam *operating system* seperti Linux, Windows, dan Macintosh. *User Interface* pada Navicat mengadopsi hierarki *database* dari phpMyAdmin dan memuat beberapa kustomisasi agar mempermudah pengguna dalam mengonfigurasi *database*-nya. Navicat merupakan *software* yang premium, jadi sebelum Anda menggunakan aplikasi ini, Anda diharuskan membeli lisensi resminya dari Navicat. Dan pada instalasinya, akan diminta konfirmasi pembelian dengan memasukkan kode lisensi saat melakukan instalasi *software* tersebut [4].

Sebagai *database interface* Navicat sangat memudahkan pengguna dalam mengeksekusi *Data Definition Language* dan dalam mengeksekusi *Data Manipulation Language* dari berbagai jenis *Database Management System (DBMS)*. Dengan menggunakan aplikasi ini, Anda tidak diharuskan menghafal semua *syntax* dari *query* karena Navicat akan mengeksekusi semua *query* yang diperlukan hanya dengan melalui klik pada aplikasinya [5].

#### E. SQLYog

SQLYog merupakan aplikasi administrasi pada *Database* basis besar di Indonesia. SQLYog adalah aplikasi yang dikembangkan oleh Webyog yang berbasis di kota Bangalore, India yang memiliki versi gratis atau *open source*, dan ada juga yang berbayar. SQLYog memiliki sangat beragam alat yang

bisa membantu para penggunanya dalam mengelola *database*. Pada aplikasi ini dilengkapi dengan kelengkapan seperti *Store Procedure, Function, dan Trigger* dengan sangat mudah [6].

#### F. XAMPP

XAMPP merupakan *software* yang namanya diambil dari potongan beberapa kata seperti Apache, MySQL atau MariaDB, PHP dan Perl. Sedangkan “X” di ujung kiri asal muasalnya dari sebuah terma *cross platform* yang menerangkan bahwa aplikasi ini bisa digunakan di berbagai macam *platform operating system* seperti Windows, Linux, Mac, dan Solaris. Pencipta pertama *software* ini, yaitu tim Apache Friends dan hingga sekarang sudah mencapai 7.4.11 yang bisa digunakan secara bebas dan gratis karena sudah memiliki lisensi GNU (*General Public License*). *Software* XAMPP memiliki fungsi yang membuat *server* lokal dengan mengaktifkan beberapa modul dan biasanya digunakan dalam proses pengembangan *website*. XAMPP digunakan dalam menguji aplikasi yang sedang dalam proses pengembangan ataupun sekadar menampilkan konten *website* tanpa harus terkoneksi dengan internet karena agar XAMPP berjalan, tidak harus menggunakan internet. Dan bisa membuat *web hosting* di lokal *server* tetapi hanya bisa diakses terbatas [7].

#### G. Query

*Query* merupakan perintah dalam sistem *database* yang berguna untuk meng-*access* dan men-*display* data pada DBMS (*Database Management System*). Dengan *query*, kita dapat mendefinisikan data, mengontrol data, dan juga memanipulasi pada *database*. *Query* juga digunakan untuk membuat koleksi data bisa saling berinteraksi. Bahasa *query* atau *query language* memiliki beberapa tipe bahasa, dan SQL termasuk kategori bahasa yang terkenal sekali di kalangan *Database Administrator*. Dalam kegunaannya *query* dibagi ke dalam tiga jenis *database*, di antaranya:

##### 1). Data Definition Language

DDL merupakan *query* dari SQL yang digunakan dalam mendefinisikan data. *Query* bisa membuat tabel baru, mengubah isi tabel, mendefinisikan *index*, menyusun bentuk tabel dan lain sebagainya. Perintah *query* yang tersedia di DDL, yaitu : *DROP, ALTER, dan CREATE*.

##### 2). Data Manipulation Language

DML, yaitu metode yang digunakan dalam SQL untuk mengeksekusi *manipulation* data pada *database*. Perintah yang tersedia di DML, antara lain : *SELECT, UPDATE, DELETE, dan INSERT*.

##### 3). Data Control Language

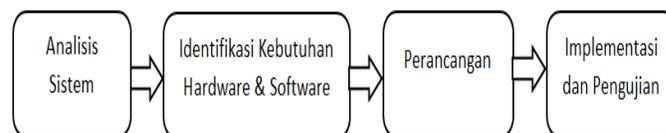
DCL merupakan metode yang digunakan dalam SQL dalam

mengatur otoritas akses *database*, dalam mengaudit *database*, mengalokasikan ruang dalam *database*, dan juga mendefinisikan ruang dalam *database*. Perintah yang terdapat pada DCL ialah : *GRANT, REVOKE COMMIT, dan ROLLBACK*.

### III. METODE PENELITIAN

Pelaksanaan dalam penelitian di sini menggunakan metode kuantitatif, di mana data yang akan digunakan sebagai objek penelitian adalah berupa angka. Sedangkan untuk mengumpulkan data-data dan bahan penelitian, penulis memanfaatkan teknik dengan observasi, di mana dalam proses pengumpulan data penulis berfungsi sebagai pengamat proses penelitian dengan mengumpulkan sumber-sumber data yang valid. Data yang diamati, yaitu yang berkaitan dengan proses pengukuran *query select* berdasarkan tingkat efisiensi waktu pada aplikasi *database* Navicat dan SQLYog.

Tahapannya terbagi ke dalam empat langkah, yaitu: menganalisis pada sistem, mengidentifikasi spesifikasi *software* dan *hardware* yang digunakan, perancangan percobaan, dan juga pengimplementasian dan pengujian, seperti yang diperlihatkan pada Gambar 1.



Gambar 1. Metode Penelitian

#### A. Menganalisis Sistem

Di bagian ini akan dilaksanakan analisis terhadap cakupan *system* secara global untuk percobaan yang dikembangkan. Uji coba *query select* berdasarkan tingkat efisiensi waktu ini memerlukan perangkat keras (*hardware*), perangkat lunak (*software*) serta sampel data untuk diinput pada MySQL *database*.

#### B. Mengidentifikasi Hardware dan Software

Di bagian ini akan dilaksanakan identifikasi perangkat keras dan perangkat lunak sebagai kebutuhan saat proses uji coba *query select* berdasarkan efisiensi waktu. Spesifikasi *hardware* dan *software* yang digunakan ditampilkan pada Tabel 1 dan Tabel 2.

Tabel 1.  
Software yang digunakan

No	Software	Version
1.	Microsoft Windows	10

2.	MySQL	5.7
3.	Navicat Premium	15.0.18
4.	SQLYog	13.1.1

Tabel 2.  
Hardware yang digunakan

No	Item	Description
1.	Processor	Intel® Core™ i5-5200U CPU @2.20GHz (4 CPUs)
2.	Memory	4GB
3.	Graphic	AMD Radeon R5 M330
4.	Harddisk	500 GB

### C. Perancangan

Di bagian ini akan dilaksanakan pembuatan rancangan sebagai bentuk persiapan yang diperlukan ketika di tahap percobaan nanti. Adapun hal-hal yang harus disiapkan sebelum dilakukan proses pengukuran *query select* berdasarkan tingkat efisiensi waktu, yaitu:

#### 1). Mempersiapkan Data

Data yang dijadikan uji coba pada penelitian kali ini adalah *database game* yang sudah tidak aktif dan berisi lebih dari 5.000 *record* pada tabel *accounts* dan lebih dari 50.000 *record* jika dijumlahkan bersama seluruh tabel lainnya.

#### 2). Rancangan Percobaan

Percobaan dilakukan dengan mengeksekusi *query* yang akan mengeluarkan *record* sebanyak 50, 100, 500, 1000, dan 5000. *Query* yang akan dieksekusi pada percobaan kali ini, yaitu *select*.

#### 3). Pengukuran

Setiap kali *query* dijalankan pada masing-masing batas *record* akan diulang sebanyak 10 kali, dan pengukuran *query* berdasarkan tingkat efisiensi waktu yang digunakan untuk perbandingan akan diambil nilai rata-ratanya. Setelah itu, hasilnya akan diinput ke dalam tabel serta direpresentasikan dalam bentuk grafik.

### D. Pengimplementasian dan Pengujian

Pada bagian implementasi dan pengujian ini dilakukan instalasi *software* XAMPP yang akan digunakan untuk mengaktifkan modul *MySQL server* pada Windows, lalu aplikasi *Navicat Premium*, dan *SQLYog* sebagai *software* yang akan dibandingkan kecepatan *query*-nya pada percobaan kali

ini. Kemudian dilakukan konfigurasi agar *Navicat Premium* dan *SQLYog* terkoneksi dengan *MySQL server*.

## IV. HASIL

### A. Pengimplementasian Query

Di tahapan pertama, yaitu memulai membuat *query* sederhana yang digunakan untuk proses uji coba. *Output* yang diharapkan akan mengeluarkan *record* berjumlah 50, 100, 500, 1000, dan 5000. Maka dari itu pada *query* ditambahkan batasan atau *limit* untuk membatasi *output* yang dihasilkan dari *query*.  
 SELECT \* FROM accounts WHERE 1 LIMIT 50;  
 SELECT \* FROM accounts WHERE 1 LIMIT 100;  
 SELECT \* FROM accounts WHERE 1 LIMIT 500;  
 SELECT \* FROM accounts WHERE 1 LIMIT 1000;  
 SELECT \* FROM accounts WHERE 1 LIMIT 5000;

### B. Eksekusi Query

Dari hasil eksekusi *query select* yang tertera pada A. *Pengimplementasian Query* dengan batasan-batasan yang telah ditentukan, didapatkan data pengukuran *query select* berdasarkan tingkat efisiensi waktu dari 10 kali percobaan seperti yang ditunjukkan pada Tabel 3 untuk *software* *Navicat Premium* dan Tabel 4 untuk *software* *SQLYog*.

Tabel 3.  
Hasil pengukuran *query select* untuk *software* *Navicat Premium*.

*	1	2	3	4	5	6	7	8	9	10
<b>LIMIT (50)</b>	0.0 04	0.0 02	0.0 04	0.0 03	0.0 03	0.0 02	0.0 02	0.0 03	0.0 03	0.0 02
<b>LIMIT (100)</b>	0.0 04	0.0 05	0.0 04	0.0 04						
<b>LIMIT (500)</b>	0.0 16	0.0 14	0.0 17	0.0 15	0.0 14	0.0 17	0.0 15	0.0 15	0.0 14	0.0 16
<b>LIMIT (1000)</b>	0.0 3	0.0 23	0.0 27	0.0 27	0.0 25	0.0 33	0.0 26	0.0 32	0.0 24	0.0 27
<b>LIMIT (5000)</b>	0.1 38	0.1 39	0.1 31	0.1 23	0.1 35	0.1 39	0.1 33	0.1 37	0.1 28	0.1 15

Tabel 4.  
Hasil pengukuran *query select* untuk *software* *SQLYog*.

*	1	2	3	4	5	6	7	8	9	10
<b>LIMIT (50)</b>	0.0 03	0.0 03	0.0 02	0.0 03	0.0 03	0.0 02	0.0 03	0.0 04	0.0 03	0.0 03
<b>LIMIT (100)</b>	0.0 04	0.0 05	0.0 04	0.0 05	0.0 04	0.0 07	0.0 04	0.0 04	0.0 04	0.0 05
<b>LIMIT (500)</b>	0.0 16	0.0 16	0.0 17	0.0 17	0.0 15	0.0 17	0.0 15	0.0 16	0.0 15	0.0 15
<b>LIMIT (1000)</b>	0.0 31	0.0 29	0.0 3	0.0 31	0.0 32	0.0 3	0.0 31	0.0 32	0.0 32	0.0 31
<b>LIMIT (5000)</b>	0.1 42	0.1 43	0.1 46	0.1 49	0.1 56	0.1 45	0.1 43	0.1 49	0.1 45	0.1 44

### C. Perbandingan Query Berdasarkan Efisiensi Waktu

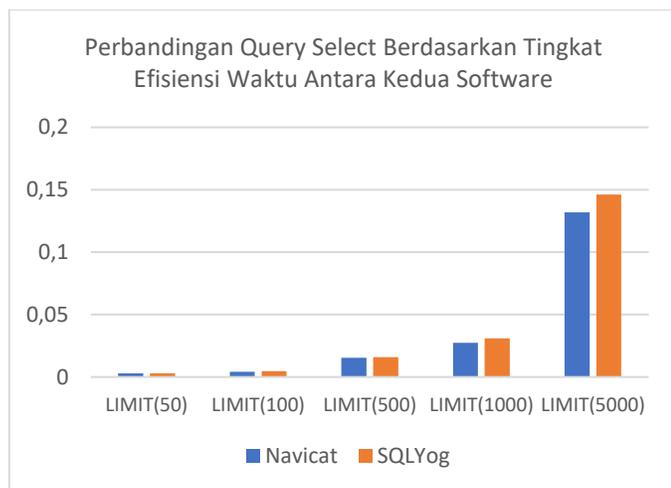
Setelah dilakukan eksekusi pada *query* dan telah didapatkan hasil pengukuran *query select* berdasarkan tingkat efisiensi waktu dari 10 kali percobaan, pada tahap ini diambil nilai rata-rata dari setiap batasan (*limit*) kedua *software* tersebut, sehingga didapatkan data seperti yang diperlihatkan pada tabel berikut.

Tabel 5.

Perbandingan *query select* berdasarkan tingkat efisiensi waktu antara kedua *software*.

	LIMIT (50)	LIMIT (100)	LIMIT (500)	LIMIT (1000)	LIMIT (5000)
<b>Navicat</b>	0.0028	0.0041	0.0153	0.0274	0.1318
<b>SQLYog</b>	0.0029	0.0046	0.0159	0.0309	0.1462

Jika dari Tabel 5 divisualisasikan menjadi bentuk diagram, maka akan menghasilkan perbandingan seperti yang diperlihatkan pada gambar diagram berikut.



Gambar 2. Perbandingan *query select* berdasarkan tingkat efisiensi waktu antara kedua *software*.

### V. KESIMPULAN

Dari percobaan yang sudah dilaksanakan terhadap penelitian kali ini, maka diambil kesimpulan bahwa perbandingan nilai rata-rata yang diperoleh dari proses pengukuran *query select* berdasarkan tingkat efisiensi waktu dari 10 kali percobaan terhadap aplikasi Navicat dan SQLYog, yaitu pada aplikasi Navicat memiliki tingkat efisiensi waktu yang lebih tinggi (*response time* yang lebih cepat) dibandingkan dengan aplikasi SQLYog, baik dalam pengujian *record* data dengan batasan 50,

100, 500, 1000, maupun 5000.

### REFERENSI

- [1] M. Silalahi dan D. Wahyudi. (2018, Februari). Perbandingan Performansi Database MongoDB dan MySQL dalam Aplikasi File Multimedia Berbasis Web. *Computer Based Information System Journal*. [Online]. 6(1), pp. 63-63. Available: <http://ejournal.upbatam.ac.id/index.php/cbis>
- [2] TermasMedia. (2020, September). Pengertian Database. [Online]. Available: <https://www.termasmedia.com/lainnya/software/69-pengertian-database.html>
- [3] W. Pratama. (2019). Relational Database Management System. [Online]. Available: [http://wahyu\\_pratama.staff.gunadarma.ac.id/Downloads/files/51353/SB D+1+Pertemuan+8+-+RDBMS.pdf](http://wahyu_pratama.staff.gunadarma.ac.id/Downloads/files/51353/SB D+1+Pertemuan+8+-+RDBMS.pdf)
- [4] Kurniawan. (2018, September). Tutorial Konfigurasi Database Menggunakan Navicat. [Online]. Available: <https://www.kaskus.co.id/thread/5b95e08d98e31b09388b4567/tutorial-konfigurasi-database-menggunakan-navicat/>
- [5] D. Suprianto, "Membuat Aplikasi Desktop Menggunakan MySQL & VB.NET Secara Profesional," Cet. ke-1. Jakarta Selatan: Mediakita, 2010, pp. 21.
- [6] N. E. Widianingsih. (2019). Aplikasi yang Digunakan untuk Database MySQL. [Online]. Available: <https://docplayer.info/143140689-Aplikasi-yang-digunakan-untuk-database-mysql.html>
- [7] Andy. (2019, September). Pengertian XAMPP Lengkap dengan Fungsi dan Cara Instalasi. [Online]. Available: <https://qwords.com/blog/pengertian-xampp/>
- [8] V D. Kurniawan. (2020, Juni). Apa itu Query SQL?. [Online]. Available: <https://www.niagahoster.co.id/blog/apa-itu-query-sql/>