

KEUNTUNGAN PENGGUNAAN *EXTERNAL FUNCTION* PADA DATABASE POSTGRESQL

Nahrin Hartono

Fakultas Teknik Komputer Universitas Cokroaminoto Palopo

E-mail: nahrinhartono@gmail.com

Abstrak

Database merupakan pusat data bagi sebuah sistem informasi, telah banyak aplikasi sistem database yang beredar baik yang berbayar maupun yang gratis. PostgreSQL merupakan salah satu sistem database yang gratis dan memiliki fitur-fitur yang handal, salah satu fitur yang membedakan PostgreSQL dengan sistem basisdata yang lainnya adalah adanya fitur pembuatan Function, dimana pengguna dapat mendefinisikan sendiri Function-nya dengan menggunakan bahasa pemrograman tertentu, yang kemudian dapat dipanggil kembali untuk dieksekusi. Pada penelitian ini penulis mencoba melakukan pengujian penambahan data dengan menggunakan perintah INSERT dan perintah External Function yang dibuat sendiri pada terminal/command prompt dan web browser. Dari hasil uji coba yang dilakukan, dihasilkan rata-rata waktu proses penambahan data pada terminal/command prompt dengan menggunakan perintah INSERT adalah 185.293 ms dan rata-rata waktu proses external function adalah 129.52 ms, sedangkan rata-rata waktu proses penambahan data pada web browser dengan menggunakan perintah INSERT adalah 168,363 ms dan external function adalah 145,64 ms. Kedua perintah tersebut terdapat perbedaan dalam penulisan perintah, namun external function merupakan perintah yang didefinisikan secara mandiri oleh pengguna hal ini memungkinkan pengguna untuk menentukan bentuk perintah sesuai yang diinginkan pengguna.

Kata Kunci: INSERT, External Function, PostgreSQL

Abstract

Database is a central data for an information system, there were many database system applications that revolve both paid and free. PostgreSQL is one of the free database system and it has the powerful feature, one of the feature that make it different with others system is a function features, where the users can make their function by themselves by using specific programming language, then it can be called return for executed. In this research the writer tries for conducting additional data testing by using INSERT Query and External Function Query on the terminal/command prompt and web browser. From the result of the trial, the research found that the average of time in additional data processing on the terminal/command prompt by using INSERT Query was 185.293 ms and the average of time External Function Query process was 129.52 ms, while the average of time in additional data processing on web browser by using INSERT Query was 168.363 ms and External Function Query was 145.64 ms. There was a difference both of the query, but the important thing and should be noted that external function is a command was defined by stand alone by the user, this matter lets the user to determine a specify command that they need.

Key Words: INSERT, External Function, PostgreSQL.

1. PENDAHULUAN

Kebutuhan akan akses data yang cepat dan fleksibel merupakan hal yang sangat penting di era modern saat ini dimana semua kebutuhan manusia telah terdigitalisasi, manusia telah meminimalisir penggunaan kertas dan digantikan dengan data digital yang disimpan pada komputer sebagai pusat data. Sistem penyimpanan dan pengelolaan data sering disebut sebagai sistem basisdata. Sama halnya dengan data hardcopy maka sistem database juga ditujukan untuk menyimpan, mencari, penggunaan dan pembagian hak akses data [3]. Database memiliki peran yang sangat penting bagi Sistem Informasi Manajemen, salah satu peran sistem basisdata yaitu sebagai sumber informasi bagi Sistem Informasi Manajemen (SIM), mengimplikasikan bahwa basisdata yang digunakan harus mampu memenuhi kebutuhan berbagai informasi bagi para penggunanya [2].

Secara umum sistem database terbagi menjadi empat komponen : 1.) Data, merupakan informasi yang disimpan dalam suatu struktur yang terintegrasi, 2.) Hardware, merupakan perangkat keras berupa komputer dengan media penyimpanan sekunder yang digunakan untuk menyimpan data karena pada umumnya database memiliki ukuran besar, 3.) Software, Merupakan perangkat lunak yang digunakan untuk pengelolaan database. Perangkat lunak sering disebut Database Management System (DBMS), 4.) User, merupakan pengguna yang berupa seseorang yang mengelola database (database administrator) ataupun yang mengambil hasil pengelolaan database melalui bahasa query (end user) yang menggunakan data yang tersimpan dan dikelola [3].

Database Management System (DBMS) merupakan perangkat lunak yang dapat melakukan utilitas dan mengelola koleksi data dalam jumlah besar. DBMS pun dirancang agar dapat melakukan manipulasi data dengan lebih mudah [3]. Beberapa contoh DBMS adalah MySQL, DB2, Oracle, SQL Server, PostgreSQL dan lain lain. PostgreSQL merupakan salah satu DBMS yang menyediakan fitur-fitur yang handal selain itu PostgreSQL merupakan sebuah sistem basis data yang disebar luaskan secara bebas. Perbedaan penting antara PostgreSQL dengan sistem database lain adalah arsitektur PostgreSQL yang memungkinkan user untuk mendefinisikan sendiri SQL-nya, terutama pada pembuatan function atau biasa juga disebut *stored procedure* [5].

2. TINJAUAN PUSTAKA

PostgreSQL pada awalnya dikembangkan oleh mahasiswa dan staff programmer University of California, Berkeley, dibawah arahan Professor Michael Stonebraker [5]. PostgreSQL mendukung Standar Query Language (SQL) dengan kemampuan antara lain transaction, subquery, trigger, dan lain lain. perbedaan yang paling mendasar antara PostgreSQL dengan sistem basis data lain adalah kemampuan PostgreSQL yang memungkinkan user untuk mendefinisikan fungsi SQL-nya secara mandiri. Fungsi dalam PostgreSQL terbagi dalam tiga bagian yakni: fungsiSQL, Fungsi Procedural Language dan Bahasa C [3].

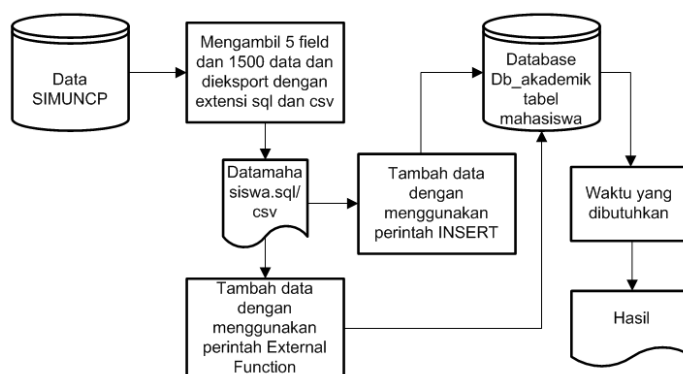
Fungsi SQL merupakan fungsi yang disusun dengan menggunakan bahasa SQL. fungsi SQL dapat berupa kumpulan query yang disimpan pada database yang jika diperlukan pengguna hanya perlu memanggil nama fungsi yang telah dibuat sebelumnya tanpa perlu lagi mengetikkan query secara detail. Fungsi yang telah dibuat akan tercatat dalam tabel pg_proc pada PostgreSQL, perintah \du digunakan melihat untuk fungsi-fungsi yang tersimpan pada tabel pg_proc [buku], data fungsi yang tersimpan dalam tabel tersebut sangat banyak maka untuk memudahkan menemukan fungsi yang dibuat perlu penyeleksian pada tabel berdasarkan atas kepemilikan (proowner) fungsi, perintah \du digunakan untuk melihat user id [3].

3. METODE PENELITIAN

Pada penelitian ini akan dilakukan pengujian pada database db_akademik yang berisi tabel mahasiswa, pada tabel tersebut akan dilakukan penginputan data dengan menggunakan *External Function* dan tanpa menggunakan *External Function* yaitu dengan menggunakan perintah *INSERT*. Uji coba pertama akan dilakukan dengan menjalankan query pada terminal/command prompt dan uji coba kedua akan dilakukan dengan menjalankan perintah query pada phpPgAdmin yang diakses dengan menggunakan *web browser* dengan alamat url: <http://localhost:8080/phpPgAdmin>, alamat tersebut akan mengakses postgresSQL dengan tampilan *Graphic User Interface (GUI)* yang ditampilkan melalui *web browser*. Spesifikasi hardware dan software yang akan digunakan pada pengujian adalah sebagai berikut :

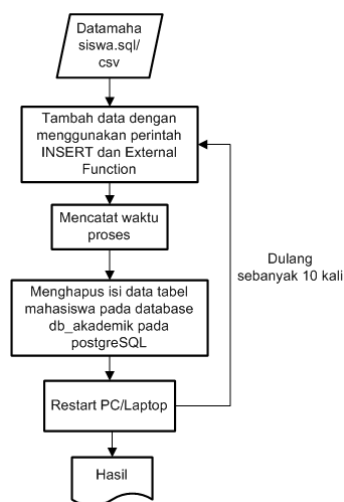
- 1.) Processor Intel Core i3 2.30. GHz
- 2.) RAM 2 GB
- 3.) Hard disk 500 GB
- 4.) Sistem Operasi Windows Seven 32-bit
- 5.) PostgreSQL 9.3
- 6.) Apache 2.2.6
- 7.) Web browser Mozilla Firefox Versi 30.0

Penelitian ini merupakan penelitian eksperimen, dimana pada penelitian akan dilakukan uji coba melakukan penambahan data pada tabel mahasiswa dengan menggunakan QUERY SQL dan dengan menggunakan *External Function*. Penambahan data dengan menggunakan kedua metode tersebut akan dilakukan penambahan data sebanyak 1500 data dan akan dilakukan sebanyak 10 kali, setiap sekali percobaan akan dilakukan restart pada portable computer, hal ini dilakukan untuk mendapatkan performance yang maksimal. adapun alur penelitian dapat dilihat pada gambar 1.



Gambar 1. Alur Kerja Penelitian

data yang digunakan pada penelitian ini adalah data yang diambil pada Sistem Informasi Manajemen Universitas Cokroaminto Palopo (SIMUNCP), dimana SIMUNCP menggunakan MySQL sebagai basis datanya. Data yang diambil pada SIMUNCP yaitu data mahasiswa yang terdapat pada tabel mhsw, untuk mendapatkan data tersebut maka dilakukan export database kedalam file dengan tipe data yang berextensi Comma Separated Value (CSV). File CSV tersebut dapat dibuka dengan menggunakan aplikasi Microsoft Office Excel [6]. Karena hasil ekspor tabel mhsw pada database SIMUNCP memiliki jumlah field yang tidak sesuai dengan field yang ada pada postgresSQL maka beberapa kolom pada hasil export akan dihapus. pada penelitian ini field yang dibutuhkan sebagai percobaan yaitu: field nim, field nama, field jenis_kelamin, field tempat_lahir dan field tgl_lahir. Dari data tersebut akan dilakukan percobaan menambahkan data dengan menggunakan perintah *INSERT* dan dengan menggunakan perintah *External Function*. Adapun alur kerja percobaan dapat dilihat pada gambar 2



Gambar 2. Alur Kerja Percobaan

Pada setiap percobaan akan dilakukan pencatatan waktu proses saat penambahan data. Berbeda dengan phpPgAdmin yang akan selalu menampilkan *time result* ketika satu perintah telah dieksekusi maka pada terminal/command prompt perlu menjalankan perintah */timing*, perintah ini akan menampilkan *time result* hasil pemrosesan dalam satuan waktu mili second.

4. HASIL DAN PEMBAHASAN

Untuk melakukan pengujian penambahan data dengan menggunakan perintah *INSERT* dan *External Function* maka pada postgresSQL dibuat database baru dengan nama *db_akademik* dan kemudian dibuat sebuah tabel dengan nama tabel mahasiswa yang mempunyai lima field yaitu : *nim*, *nama*, *jkel*, *tmplahir*, *tgllahir*. Struktur tabel mahasiswa dapat dilihat pada tabel 1

Tabel 1. Struktur Tabel Mahasiswa

Mahasiswa		
Field	Type	Attribute
Nim	Character (10)	Primary Key, CHECK
nama	Character Varying (40)	Not Null, CHECK
Jkel	Character (1)	Not Null, CHECK
tmplahir	Character Varying (40)	Not Null, CHECK
tgllahir	Date	Not Null

Pada tabel mahasiswa memiliki *PRIMARY KEY CONSTRAINT* pada field *nim*, *NOT NULL CONSTRAINT* pada field *nama*, *jkel*, *tmplahir*, *tgllahir* dan *CHECK CONSTRAINT* pada field *nama*, *jkel*, *tmplahir*. Pemberian *CHECK CONSTRAINT* pada tabel mahasiswa bertujuan agar pada

saat penambahan data tidak terjadi kesalahan peginputan [4]. Perintah SQL untuk membuat tabel mahasiswa dapat dilihat pada gambar 3

```
CREATE TABLE mahasiswa(  
  nim CHAR(10) PRIMARY KEY,  
  nama VARCHAR(40) NOT NULL,  
  jkel CHAR(1) NOT NULL,  
  tmplahir VARCHAR(40) NOT NULL,  
  tgllahir DATE NOT NULL,  
  CONSTRAINT check_nim  
    CHECK(nim ~*'^[0-9]+$'),  
  CONSTRAINT check_nama  
    CHECK(nama ~*'^[a-z,.,\,\,\ ]+$'),  
  CONSTRAINT check_jkel  
    CHECK(jkel IN ('L','P')),  
  CONSTRAINT check_tmplahir  
    CHECK (tmplahir ~*'^[a-z,.,\,\,\ ]+$'));
```

Gambar 3. Pembuatan tabel pada database

Pada database db_akademik selanjutnya akan diberikan *External Function* yang kemudian akan digunakan untuk melakukan *INSERT* data pada tabel mahasiswa, perintah *External Function* tersebut dapat dilihat pada gambar 4

```
CREATE FUNCTION tambahdata(  
  nim_arr1 text[],  
  nama_arr2 text[],  
  jkel_arr3 text[],  
  tmplahir_arr4 text[],  
  tgllahir_arr5 DATE[])  
RETURNS VOID AS  
$$  
  INSERT INTO mahasiswa  
  nim, nama, jkel, tmplahir, tgllahir)  
  SELECT  
    unnest(nim_arr1),  
    unnest(nama_arr2),  
    unnest(jkel_arr3),  
    unnest(tmplahir_arr4),  
    unnest(tgllahir_arr5);  
$$  
LANGUAGE sql STRICT;
```

Gambar 4. Pembuatan External Function

Uji coba pertama akan dilakukan menambahkan data pada database db_akademik tabel mahasiswa dengan menggunakan perintah *INSERT*, data yang ditambahkan sebanyak 1500 data, perintah untuk menambahkan data pada tabel mahasiswa dapat dilihat pada gambar 5

Gambar 5. Perintah Menambahkan Data dengan Menggunakan *INSERT*

```
(' (1301401001,1301401002,1301401003,1301401004,1301401005,1301401006,1301401007,1301401008,1301401009,1301401010,1301401011,1301401012,1301401013,1301401014,1301401015,1301401016,1301401017,1301401018,1301401019,1301401020,
```

Gambar 6. Perintah Menambahkan Data dengan Menggunakan *External Function*

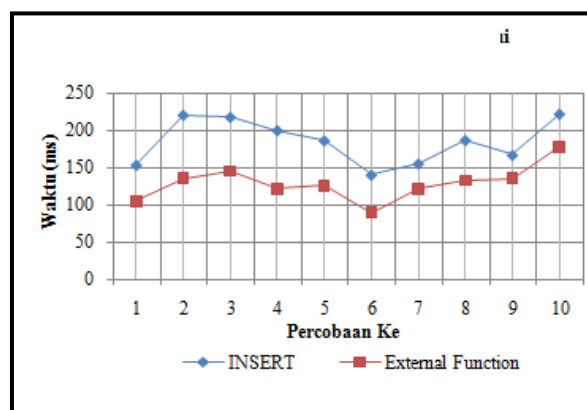
PostgreSQL merupakan sistem basisdata yang gratis tetapi memiliki fitur fitur yang handal salah satu fiturnya adalah pengguna dapat membuat sendiri perintah *function* pada basis datanya yang kemudian dapat dipanggil kembali untuk dieksekusi melalui terminal/command prompt maupun melalui phpPgAdmin yang diakses melalui *web browser*. Tapi yang perlu diketahui pembuatan fungsi fungsi yang rumit tidak dapat dibuat melalui phpPgadmin, fungsi fungsi yang rumit hanya dapat dibuat melalui terminal/command prompt. Dalam penelitian ini telah diuji coba

sebanyak 10 kali penambahan data dengan menggunakan perintah *INSERT* dan dengan menggunakan perintah *External Function* yang telah dibuat melalui terminal/command prompt. Hasil dari percobaan ditunjukkan pada tabel 2.

Tabel 2. Hasil dengan Menggunakan Perintah *INSERT* dan *External Function* pada Terminal/Command Prompt

Percobaan Ke	<i>INSERT</i> (ms)	<i>External Function</i> (ms)
1	153.83	105.28
2	220.99	135.77
3	218.19	145.98
4	200.26	122.24
5	186.69	126.02
6	140.59	89.66
7	155.62	122.22
8	187.23	133.24
9	167.20	135.91
10	222.33	178.901
Rata rata	185.293	129.52

Pada tabel 2 waktu proses tercepat dengan menggunakan perintah *INSERT* pada terminal/command prompt adalah 140,59 ms dan waktu proses tercepat dengan menggunakan perintah *External Function* pada terminal/command prompt adalah 89,66 ms, dari sepuluh kali percobaan didapatkan hasil rata rata proses penambahan data dengan menggunakan perintah *INSERT* terminal/command prompt yaitu 185,293 ms, sedangkan hasil rata rata proses penambahan data dengan menggunakan perintah *External Function* pada terminal/command prompt adalah 129,52 ms. Grafik hasil dari percobaan penambahan data dengan menggunakan perintah *INSERT* dan *External Function* pada terminal/command ditunjukkan pada gambar 7.



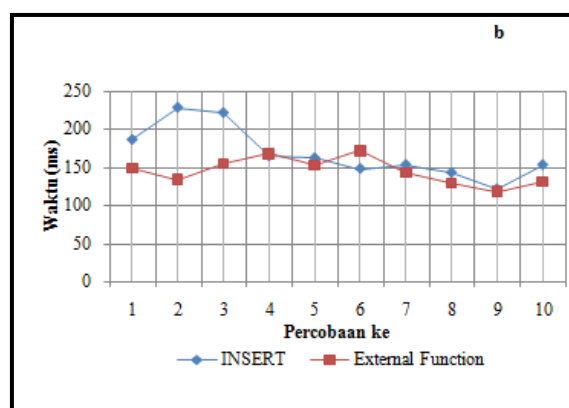
Gambar 7. Grafik Hasil Penambahan Data Melalui Terminal/Command Prompt

Pada percobaan selanjutnya penulis melakukan penambahan data dengan menggunakan perintah *INSERT* dan *External Function* pada *web browser* dengan mengakses url: <http://localhost:8080/phpPgAdmin>, hasil percobaan ditunjukkan pada tabel 3

Tabel 3. Hasil dengan Menggunakan Perintah *INSERT* dan *External Function* pada *web browser*

Percobaan Ke	<i>INSERT</i> (ms)	<i>External Function</i> (ms)
1	186.41	149.80
2	228.61	134.40
3	221.81	155.40
4	164.80	168.20
5	162.40	153.00
6	148.20	172.80
7	153.40	143.40
8	143.00	129.80
9	121.60	118.00
10	153.40	131.60
Rata rata	168.363	145.64

Pada tabel 3 menunjukkan waktu proses tercepat dengan menggunakan perintah *INSERT* melalui *web browser* adalah 121,60 ms dan waktu proses tercepat dengan menggunakan perintah *External Function* melalui *web browser* adalah 118,00 ms. hasil rata rata yang diperoleh dari sepuluh kali percobaan penambahan dengan menggunakan perintah *INSERT* pada *web browser* adalah 168,363 ms sedangkan penambahan data dengan menggunakan perintah *External Function* pada *web browser* adalah 145,64 ms. Grafik hasil percobaan penambahan dengan menggunakan *INSERT* dan *External Function* pada *web browser* ditunjukkan pada gambar 8.



Gambar 8. Grafik Hasil Penambahan Data Melalui *Web browser*

Dari kedua hasil percobaan yang telah dipaparkan diatas, maka ditemukan hasil bahwa perbedaan waktu proses penambahan dengan menggunakan perintah *INSERT* dan *External Function* melalui terminal/command prompt adalah 50,93 ms sedangkan penambahan data pada *web browser* dengan perintah *INSERT* dan *External Function* adalah 3,60 ms, maka bisa disimpulkan bahwa proses penambahan data dengan menggunakan *External Function* lebih cepat dibandingkan penambahan data dengan menggunakan perintah *INSERT*

5. KESIMPULAN

Penambahan data pada PostgreSQL dapat dilakukan dengan menggunakan perintah *INSERT* ataupun dengan *External Function* yang didefinisikan sendiri oleh pengguna, pada bab sebelumnya telah dijelaskan ada perbedaan waktu proses antara penggunaan penambahan data dengan perintah *INSERT* dan *External Function* baik melalui terminal/command prompt maupun melalui *web browser* phpPgAdmin. Namun hal yang pertama yang perlu diperhatikan bahwa penggunaan *External Function* tidak hanya sebatas penambahan data, *External Function* dapat juga melakukan perhitungan, percabangan, perulangan dengan menggunakan bahasa program tertentu [1]. Dalam penelitian ini dengan jelas diperlihatkan bahwa penambahan data dengan menggunakan perintah *External Function* memiliki proses kerja lebih cepat dibandingkan dengan menggunakan perintah *INSERT*, namun penulisan perintah yang berbeda sangat jelas ditunjukkan, pada perintah *External Function* penulisan perintah sangat tidak *user friendly* ini dikarenakan perintah yang diketikkan bukan merupakan *record* per *record*, seperti pada perintah *INSERT* namun data yang diketikkan berupa *field* per *field*, ditunjukkan pada gambar 5 dan gambar 6, namun penambahan data dengan menggunakan perintah *External Function* dapat didefinisikan sendiri yang berarti format pengetikan perintah juga dapat diatur sedemikian rupa untuk memudahkan dalam penginputan.

6. SARAN

Dari keseluruhan pemamparan pada penelitian ini sudah dapat disimpulkan bahwa peneliti ini terfokus pada penambahan data dengan menggunakan perintah *INSERT* dan *External Function*, sedangkan perintah *UPDATE* dan *DELETE* belum dilakukan uji coba, hal ini merupakan keterbatasan penulis baik itu keterbatasan pengetahuan penulis dan keterbatasan waktu sehingga dalam penelitian ini masih banyak memiliki kekurangan, penulis berharap dari penelitian menjadi acuan untuk penelitian yang akan datang yang menyangkut tentang sistem database PostgreSQL.

DAFTAR PUSTAKA

- [1] Raharjo, S. 2013. *Integrity Constraint Basis Data Relasional dengan Menggunakan PL/PGSQL dan CHECK Constraint*, SEMNAS Teknologi Informasi dan Multimedia, STMIK AMIKOM Yogyakarta.
- [2] Raharjo, S., Sutanta, E., dan Utami, E. 2007. *Analisi Aspek Aspek Kualitas Schema database (Studi Kasus pada Database Akademik Ista Yogyakarta)*. Seminar Nasional Teknologi 2007 (SNT 2007).
- [3] Utami, E., dan Raharjo, S. 2006, *RDBMS dengan PostgreSQL di GNU/Linux*, Andi Offset, Yogyakarta.
- [4] Utami, E. 2014, *The Advantages of Using CHECK Constraints in The Academic Database Table*, Journal Of Software, Vol. 9 No. 2, STMIK AMIKOM, Yogyakarta.
- [5] The PostgreSQL Global Development Group, *PostgreSQL 9.3.4 Documentation*, 2014.
- [6] W. Gilmore, *Beginning PHP and MySQL: From Novice to Professional*, ser. Aprespod Series. Apres, 2010.