

EVALUASI DATABASE SISTEM INFORMASI ALUMNI UNIVERSITAS MUHAMMADIYAH BENGKULU

¹⁾Yetman Erwadi, ²⁾Ema Utami, ³⁾Hanif Al Fatta

^{1),2),3)} Program Studi MTI Universitas AMIKOM Yogyakarta

Jalan Ring Road, Condong Catur, Depok, Sleman. Telp. : 0274-884201

Email : ¹⁾abangyetman25@gmail.com, ²⁾emm@nrar.net, ³⁾hanif.a@amikom.ac.id

ABSTRACT

Database is an important part of information system, one of the role of database system that is as source of information for information system. Information system alumni of Muhammadiyah University of Bengkulu is a web-based application that uses MySQL as its database system. Evaluation is done to identify the errors that exist in the database SIAUMB, after the evaluation on the database there are some errors in the database SIAUMB, including errors in the relation, field, key, constraint, record, query, adding data (insert), change data (update) and deletion of data (delete). Future databases can produce an alumni information system capable of providing exploratory information quickly, easily and accurately and able to assist the parties concerned in making management decisions alumni Muhammadiyah University of Bengkulu. Then can be done new database design with reference to result of evaluation which have been done. After that, data quality testing on aspects of accuracy and non-duplicate aspects, testing is done by executing query, each query that executed will generate record and result column then compared between new database with old database before evaluation.

Keywords : Database, Alumnus, Evaluation

1. PENDAHULUAN

Database sampai saat ini menjadi bagian penting bagi sistem informasi. Kebutuhan informasi sekarang ini telah tergitalisasi, manusia telah meminimalisir penggunaan kertas dan digantikan dengan data-data digital yang disimpan pada komputer sebagai pusat data. Sistem penyimpanan dan pengolahan data dalam komputer tersebut sering disebut sebagai *database*. *Database* merupakan kumpulan data yang umumnya menggambarkan aktifitas-aktifitas dan pelakunya dalam suatu organisasi. *Database* berperan sebagai sumber informasi bagi sebuah sistem informasi, *database* harus mampu memenuhi kebutuhan akan informasi bagi penggunaannya. Secara umum sistem *database* terbagi menjadi beberapa komponen yaitu data, *hardware*, *software*, dan *user*. *Database* dalam perancangannya memiliki aturan tersendiri dan menggunakan bahasa pemrograman tersendiri untuk mengolahnya. SQL atau disebut dengan SEQUEL (*Strucured English Query Language*) merupakan bahasa pemrograman

memiliki tujuan khusus dan dirancang untuk mengelola data dalam sistem manajemen basis data (SMBD). Penelitian yang dilakukan Hartono et al (2016). Membahas perancangan sebuah *database* baru sebagai target migrasi dengan memperbaiki kesalahan-kesalahan yang ada pada *database* lama, melakukan pengukuran kualitas data pada *database* setelah migran dilakukan, dan memaksimalkan kinerja *database* dengan mengoptimalkan *query* yang digunakan pada aplikasi sistem informasi manajemen Universitas Cokroaminoto Palopo.

Dari pembahasan di atas maka hasil peneliti tersebut menghasilkan informasi yaitu pada *database* yang lama tidak terdapat *foreign key*, maka dari itu untuk mengetahui relasi dari tabel-tabelnya yaitu dengan melakukan pengecekan setiap *join query* yang ada pada *source code* aplikasi, selain itu untuk mengetahui relasi dari setiap tabel juga dilakukan dengan menganalisis informasi yang dihasilkan dari setiap tabel. Berdasarkan hasil evaluasi basis data lama maka dapat di ketahui bahwa terdapat beberapa

kesalahan-kesalahan pada basis data tersebut, yaitu ada beberapa tabel pada *database* yang tidak menggunakan *primary key*, kolom yang menjadi referensi bagi tabel lain tidak unik, dapat kolom yang menghasilkan informasi yang sama pada tabel yang berelasi. Perancangan *database* baru sebagai target migrasi dilakukan dengan mengacu pada hasil evaluasi *database* lama.

Pengukuran kualitas data dilakukan setelah migrasi, hasil pengukuran kualitas data pada *database* baru adalah baik terutama pada aspek non duplikat dengan nilai persentase 100% untuk aspek akurasi tidak mencapai nilai persentase 100% disebabkan oleh beberapa faktor yaitu data pada tabel yang saling berelasi tidak memiliki referensi dan perubahan data pada saat proses migrasi dan perubahan kolom. Optimalisasi dilakukan pada *query-query* yang ada pada *source code* aplikasi, dari hasil pengujian dari lima kelompok *query* dihasilkan *query* yang telah dioptimasi dan eksekusi pada *database* baru lebih cepat dibanding *query* yang belum teroptimasi.

Naoyuki Miyamoto, Ken Higuchi dan Tatsou Tsuji dalam penelitiannya membahas tentang migrasi *database* pada *multi-database system* MySQL dengan *spider storage engine*. Pada penelitian tersebut penulis merancang *multi-database system* dengan menggunakan *engine spider storage*. Dalam penelitian tersebut terdiri dari 5 atribut, dan berisi lima ratus *record*. Dengan menerapkan skenario pengujian, migrasi *database* dengan mengganti tipe *engine* pada *database* dapat berjalan maksimal. (Miyamoto et al, 2014).

Penelitian Qian Wang Chun Yu dan Naijia Liu, membahas tentang migrasi *database* Universitas Tsinghua. Dalam penelitian tersebut dijelaskan bahwa hal penting dalam migrasi data adalah integritas data, akurasi data dan keberlangsungan aplikasi. Penelitian tersebut menghasilkan bahwa metode *transportable tablespaces* membutuhkan waktu proses yang lebih singkat namun kecenderungan akan hilangnya data lebih besar, metode *export* dan *import* membutuhkan waktu yang cukup lama dalam proses migrasi, namun kecenderungan akan hilangnya data lebih kecil. Penulis merekomendasikan dalam penelitiannya bahwa jika kapasitas *database*

yang akan dimigrasi berukuran besar maka sebaiknya menggunakan metode *transportable tablespaces* (Wang et al, 2012).

Mengying Zhang dalam penelitiannya membahas tentang migrasi *stored procedure database oracle* ke *postgresql*, dalam penelitiannya dijelaskan bahwa ada dua cara melakukan migrasi *procedure database*, yang pertama adalah dengan melakukan modifikasi *source code* pada *postgresql* agar dapat didukung oleh *Oracle*, namun metode ini akan memberikan beban pekerjaan dan administrator *database* harus paham betul dengan *source code* pada *postgresql*. Cara yang kedua yaitu dengan membangun *middleware*, *middleware* digunakan untuk digunakan untuk menulis ulang prosedur pada *database oracle* agar dapat dikenali pada *database postgresql*, dalam kasus ini digunakan *tool antlr parser* (Zhang, 2013).

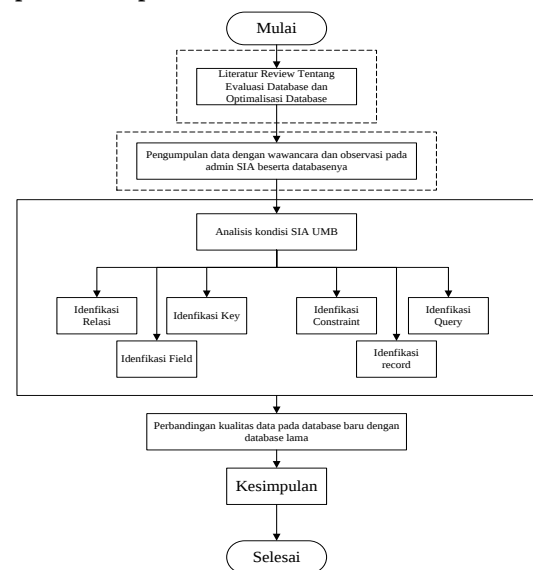
SQL memiliki tiga bagian utama yaitu bahasa pemrograman untuk mendefinisikan data (*Data Definition Language-DDL*), untuk manipulasi dan akses data (*Data Manipulation Language-DML*), dan digunakan untuk bagian pengawasan/kontrol pemakai (*Data Control Language*). tabel, kolom dan *record* merupakan elemen penting dalam *database*, ketiga elemen tersebut tidak dapat dipisahkan satu sama lain, dan tidak menutup kemungkinan dalam suatu *database* data terdapat banyak tabel, didalam tabel memiliki banyak kolom dan kolom-kolom tersebut berisi *record-record* (data). Sistem informasi alumni Universitas Muhammadiyah Bengkulu (SIAUMB) aplikasi *web* yang diimplementasikan pada jaringan *Local Area Network* (LAN) dan menggunakan MySQL sebagai sistem manajemen *databases*nya. Sistem informasi alumni UMB ini belum dilakukan evaluasi sama sekali semenjak di bangun dan dioperasikan pada tahun 2012, salah satu hal yang menjadi masalah pada *database* SIAUMB yang digunakan saat ini adalah persoalan penginputan data dimana data yang digunakan saat ini tidak

menyediakan fitur *constraint check*. Fitur *constraintn check* sendiri berfungsi untuk membatasi penginputan data pada *database*, *user* akan dipaksa memasukan data sesuai dengan format/aturan pada *constraint check* yang telah dibuat. Sebagai contoh adalah penggunaan *contraint check* untuk membatasi nim mahasiswa, dimana nim mahasiswa pada Universitas Muhammadiyah Bengkulu sepuluh karakter angka, dengan menggunakan *contraint check* maka *user* akan dipaksa mengikuti aturan tersebut, selain itu pembatasan penginputan data juga mencegah terjadinya eksploitasi antar muka aplikasi ke suatu *database*. Kesalahan dalam pembuatan program antar muka yang dilakukan komunikasi ke *server database* dapat menyebabkan munculnya kerawanan keamanan terhadap data yang disimpan. Injeksi SQL melalui program antar muka dapat dilakukan apabila terjadi kesalahan program. Akar penyebab dari injeksi SQL adalah ketidakcukupan validasi masukan. Evaluasi sangat perlukan untuk mengetahui kesalahan-kesalahan apa saja yang ada pada *database* SIAUMB, hasil evaluasi bisa dijadikan untuk merancang *database* yang baru untuk memaksimalkan sistem informasi alumni Universitas Muhammadiyah Bengkulu.

2. METODE PENELITIAN

Evaluasi dilakukan dengan menganalisis struktur *datbase* SIAUMB untuk mendapatkan informasi hasil dari identifikasi relasi, *field*, *key*, *constraint*, *record* dan *query* yang digunakan oleh aplikasi SIAUMB. Hasil evaluasi kemudian dijadikan rujukan untuk merancang *database* baru dengan melakukan perbaikan-perbaikan pada *database* lama. Hasil dari rancangan *database* baru setelah dievaluasi kemudian dibandingkan dengan *database* lama sebelum di evaluasi dengan mengeksekusi *query*, adapun aspek yang dibandingkan adalah aspek non duplikat dan aspek akurasi. Aspek non duplikat dilakukan dengan membandingkan hasil eksekusi *query SELECT* yang akan menghasilkan data pada kolom yang terseleksi dan hasil eksekusi *query SELECT* dengan klausa *GROUP BY* yang akan menghasilkan data

yang sebenarnya pada kolom yang terseleksi, *query* tersebut dieksekusi pada *database* setelah dievaluasi dan *database* lama sebelum evaluasi. Adapun tabel yang dilakukan pengujian adalah tabel yang menyimpan data mahasiswa yaitu pada tabel alumni dan tabel prodi, kedua tabel tersebut dipilih karena diasumsikan bahwa seharusnya tabel tersebut tidak terdapat data-data duplikat. Aspek kedua adalah aspek akurasi, untuk aspek akurasi pengujian dilakukan pada akurasi baris dan akurasi kolom hal tersebut dilakukan dengan mengesekusi join *query* yang didapat dari *soure code* aplikasi SIAUMB, setiap *query* yang dieksekusi akan menghasilkan jumlah baris record dan jumlah kolom, hasil tersebut dilakukan perbandingan antara *database* lama sebelum evaluasi dengan *database* baru setelah evaluasi. Berikut ini alur penelitian pada Gambar 1.



Gambar 1. Alur Penelitian

3. HASIL DAN PEMBAHASAN

Pada penelitian ini dilakukan evaluasi *database* sistem informasi alumni Universitas Muhammadiyah Bengkulu yang akan menjadi rujukan untuk perancangan *database* baru. Setelah dilakukan pengujian aspek akurasi dan pengujian aspek non duplikat. Berdasarkan hasil analisis identifikasi relasi, *field*, *key*, *constraint*, *record* dan

query pada *database* SIAUMB maka dapat diketahui : 1). Terdapat satu tabel pada *database* yang tidak menggunakan *primary key*. *Primary key* sendiri merupakan satu atau lebih nilai pada kolom untuk membuat sebuah *record* unik. 2). Tidak adanya penggunaan *foreign key* pada semua tabel. *Foreign key* berguna untuk menjamin integritas referensial dan untuk memastikan nilai pada suatu kolom (atau kumpulan kolom) sesuai dengan nilai yang ada pada tabel 3. Berdasarkan *query* yang ada pada *source code* aplikasi SIAUMB maka di ketahui bahwa terdapat 14 tabel belum digunakan.

Hasil evaluasi yang dilakukan sebelumnya, menunjukkan perbedaan struktur *database* sebelum evaluasi dan sesudah evaluasi ditunjukkan pada tabel 1 berikut ini.

Tabel 1. Perbedaan database lama dengan database baru setelah evaluasi

No	Perbedaan	Evaluasi		Keterangan
		Se-sudah	Se-belum	
1	Primary Key	Ya	Ya	Sebelum evaluasi terdapat 1 tabel yang tidak menggunakan <i>primary key</i>
2	Foreign Key	Ya	Tidak	
3	Check Constraint	Ya	Tidak	
4	Jumlah Tabel	15	16	Beberapa tabel setelah di evaluasi tidak digunakan karena tidak di sebutkan pada <i>source code</i> aplikasi
5	Jumlah Kolom	136	168	Beberapa kolom tidak digunakan

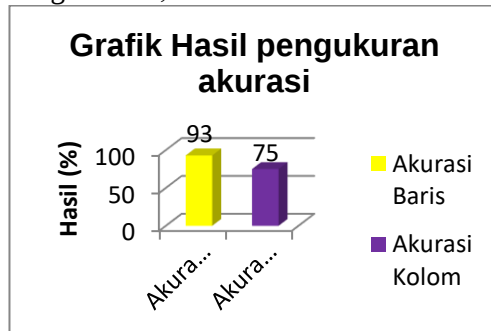
				atau dirancang karena tidak disebutkan pada <i>source code</i> aplikasi dan memiliki informasi Yang sama pada saat berrelasi
--	--	--	--	--

Dapat di lihat bahwa terdapat perbedaan antara *database* sebelum di evaluasi dengan *database* setelah dilakukan evaluasi, *database* baru di rancang dengan menggunakan parameter *database*.

Check constraint sendiri digunakan untuk membatasi penambahan data pada *database*. Setelah melakukan perancangan *database* baru, hal selanjutnya yang dilakukan adalah memindahkan data-data yang ada ke *database* yang baru dirancang, pemindahan data dilakukan dengan *eksport* dan *import*. Setelah semua data berhasil dipindahkan selanjutnya dilakukan pengukuran kualitas data.

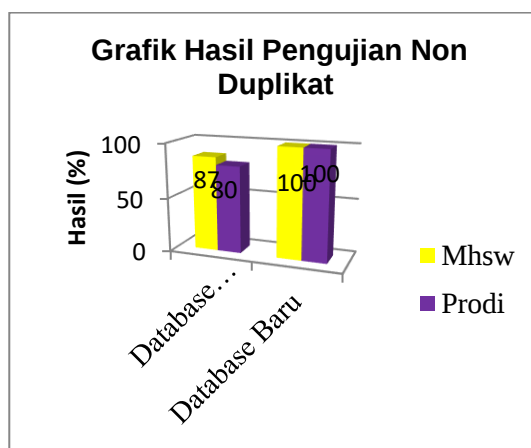
Pengukuran kualitas data non duplikat dilakukan dengan membandingkan hasil pengujian pada *database* lama sebelum di evaluasi dengan *database* setelah evaluasi, sedangkan untuk aspek akurasi hanya dilakukan pada *database* baru setelah evaluasi. Hal ini disebabkan karena tidak adanya pembandingan akurasi pada *database* data sebelum evaluasi, sedangkan pada *database* baru setelah evaluasi dapat diukur tingkat akurasi dengan menjadikan *database* lama sebelum evaluasi sebagai pembandingan. *Query* pengujian aspek akurasi akan berbeda antara *database* lama sebelum evaluasi dengan *database* baru setelah evaluasi, hal ini disebabkan karena adanya perubahan struktur pada *database* yang baru, namun perubahan *query* tersebut tidak akan mempengaruhi informasi yang dihasilkan. Pengukuran kualitas data aspek akurasi dilakukan

dengan mengeksekusi 16 *query* yang di eksekusi pada *database* sebelum dan setelah evaluasi, hasil pengujian kualitas data aspek akurasi baris dan akurasi kolom ditunjukkan pada gambar 2, berikut ini.



Gambar 2. Hasil pengukuran kualitas data aspek akurasi

Pada gambar 2 ditunjukkan grafik hasil pengujian untuk aspek akurasi baris pada *database* baru setelah evaluasi sebesar 93% sedangkan untuk aspek akurasi kolom pada



Gambar 3. Kualitas pengukuran non duplikat

Pada gambar 3 dapat dilihat bahwa pada data mahasiswa untuk *database* lama sebelum evaluasi dihasilkan nilai sebesar 87% dan untuk data prodi 80% yang berarti terdapat data yang duplikat pada data mahasiswa dan data dosen sedangkan pada database baru setelah evaluasi untuk data mahasiswa dihasilkan nilai 1 atau dengan nilai persentase 100% dan untuk data prodi juga dihasilkan nilai 1 dengan nilai persentase 100% yang berarti pada *database* baru setelah evaluasi dilakukan tidak terdapat lagi data mahasiswa dan data prodi yang duplikat.

database baru setelah evaluasi adalah sebesar 75%.

Pengukuran kualitas aspek non duplikat dilakukan pada data mahasiswa yang terdapat pada tabel mahasiswa dan data alumni terdapat pada tabel prodi, yang seharusnya data mahasiswa dan data dan data prodi tidak terjadi duplikasi. Untuk mengukur aspek non duplikat pada data tersebut dilakukan dengan mengeksekusi *query*, adapun *query* yang digunakan untuk pengujian kualitas data aspek non duplikat yaitu dengan mengeksekusi *query SELECT* tanpa klausa *GROUP BY* dan *query SELECT* dengan klausa *GROUP BY*. Hasil pengukuran kualitas data aspek non duplikat ditunjukkan pada gambar 3, berikut ini.

4. KESIMPULAN

4.1 Kesimpulan

Evaluasi *database* untuk mendapatkan kemungkinan kesalahan-kesalahan yang dapat terjadi pada *database* sistem informasi alumni Universitas Muhammadiyah Bengkulu (SIAUMB), evaluasi dilakukan dengan mengidentifikasi struktur *database* SIAUMB meliputi identifikasi relasi, identifikasi *field*, identifikasi *key*, identifikasi *constraint*, identifikasi *record* serta mengidentifikasi *query* pada *source code* aplikasi untuk mengetahui tabel dan kolom apa saja pada *database* yang digunakan oleh aplikasi SIAUMB. Setelah dilakukan evaluasi maka dapat disimpulkan :

1. *Database* tersebut sangat memungkinkan terjadinya kesalahan dalam penambahan data (*insert*), kesalahan dalam pengubahan data (*update*), serta kesalahan penghapusan data (*delete*). Hasil evaluasi kemudian dijadikan rujukan untuk merancang *database* baru untuk memperbaiki kesalahan-kesalahan tersebut. Setelah *database* baru dirancang selanjutnya

memindahkan data yang ada di *database* lama, lalu dilakukan pengujian kualitas data.

2. Hasil pengujian kualitas data pada *database* baru setelah dievaluasi untuk aspek akurasi baris adalah 99% hasil ini tidak mencapai 100% disebabkan karena ada perubahan struktur pada *database* baru setelah evaluasi sehingga beberapa data perlu dilakukan penyusunan saat proses perbaikan data ke *database* baru setelah evaluasi. Sama halnya dengan aspek akurasi kolom yang tidak mencapai nilai 100% hal tersebut disebabkan terjadinya perubahan jumlah kolom pada *database* baru setelah evaluasi, perubahan tersebut disebabkan karena beberapa kolom pada tabel tertentu di *database* lama sebelum evaluasi memiliki nilai yang sama pada tabel ini, sehingga kolom-kolom tersebut tidak dirancang.
3. Untuk hasil pengujian kualitas data untuk aspek non duplikat pada *database* lama untuk mahasiswa dan prodi yaitu senilai 96% dan 99% yang berarti masih terdapat data yang duplikat pada data mahasiswa dan prodi hal ini disebabkan tidak adanya *constraint unique* pada tabel yang menyimpan data mahasiswa dan data prodi, sedangkan pada *database* baru setelah evaluasi diberikan *constraint unique* pada tabel yang menyimpan data mahasiswa dan prodi sehingga telah dilakukan pengujian dihasilkan nilai 100% yang berarti tidak ada lagi data mahasiswa dan data prodi yang duplikat.

4.2 Saran

Dari keseluruhan pemaparan pada penelitian ini dapat disarankan yakni :

1. Pada penelitian selanjutnya evaluasi *database* dengan melakukan identifikasi kesalahan-kesalahan yang ada pada *database*, hasil dari evaluasi tersebut kemudian dijadikan acuan untuk merancang *database* baru.
2. Pengujian kualitas data aspek akurasi

dan aspek duplikat, untuk aspek-aspek lain seperti integritas turunan, kelengkapan data dan konsistensi data.

3. Membuat *tools* yang menyangkut tentang evaluasi *database* dan pengukuran kualitas data.

DAFTAR PUSTAKA

- Darmanto, E., 2015, Analisis Optimalisasi Bahasa SQL Berdasarkan Relational Algebra Pada Kasus Rekapitulasi Mahasiswa Layak Wisuda, Jurnal SIMETRIS, 6 (2), 405-414.
- Fachrurrozi, M., 2013, Peningkatan Fungsionalitas Perangkat Lunak Melalui Restrukturisasi Data (Studi Kasus: Sistem Akademik Fakultas Ilmu Komputer UNSRI), Jurnal GENERIC, 4 (1), 33-38.
- Hartono, N., Utami, E., & Amborowati, A., 2016, Migrasi dan Optimalisasi Database Sistem Informasi Universitas Cokroaminoto Palopo, Jurnal Buana Informatika.
- Kiruthika, J., & Souheil, K., 2014, *Performance Issues and Query Optimization in Big Multidimensional Data*, DCABES, 24-28.
- Miyamoto, N., Higuchi, K., & Tsuji, T., 2014, *Incremental data Migration for Multi-Database System Based on MySQL with Spider Storage Engine*, IIAIAAI, 745750.
- Nogroho, B., 2008, Panduan lengkap Menguasai perintah SQL, Jakarta selatan, MediaKita.
- Raharjo, S., Sutanta, E., & Utami, E., 2007, Analisis Aspek-Aspek Schema Database (Studi Kasus Pada Database Akademik ISTA Yogyakarta), Seminar Nasional Teknologi (pp.D1 –D13), Yogyakarta: SNT AMIKOM.
- Raharjo, S., 2013, *Integrity Constraint Basis Data Relasional dengan menggunakan PL/PGSQL dan CHECK Constraint*, Seminar Nasional Teknologi Informasi dan Multimedia, Yogyakarta.

Wang, Q., Liu, N., & Yu, C., 2012, *Research And Practice Of University Database Migration*, ITME, !, 31-34.

Zhang, M., 2013, *PMT: A Procedure Migration Tool From Oracle to PostgeSQL*, ICSSC, 391-396.