

SINKRONISASI TABEL BERBASIS *RECORD* MENGGUNAKAN SISTEM KEAMANAN *AUTHENTICATION, AUTHORIZATION, ACCOUNTING* (AAA) (STUDI KASUS DI STTA YOGYAKARTA)

Suwito, Hero Wintolo, Dwi Nugraheny
Teknik Informatika STTA Yogyakarta
informatika@stta.ac.id

Abstract

Synchronization is the process of setting the course of several processes at the same time. Synchronization builds upon students' academic value table STTA Yogyakarta by comparing records for the records in the database. If there is a difference in the academic value of the table the different values will be returned as the value of the original with comparing by using the master database. Record based table synchronization is expected to address the security system on the database and provide security guarantees a safe and comfortable.

Authentication is a process by which a user is identified by the server before the user using the network. In the process, the user is requesting access rights to the server to use a network. Authorization is the allocation of what services are entitled to be accessed by the user on the network. Authorization is done when the user has been declared eligible to use the network. Accounting is a process performed by the server to record all user activity in the network, such as when the user starts using the network, end user when the connection to the network, how long the user using the network, how much data is accessed from the network user.

The results of the implementation of synchronization using a table based record system security authentication, authorization, accounting changes that may prevent the occurrence of data in a database, optimizing the use of stored, and enhance security systems at the database STTA center of Yogyakarta.

Keywords: *table synchronization, security system authentication, authorization, accounting*

Abstrak

Sinkronisasi merupakan proses pengaturan jalannya beberapa proses pada saat yang bersamaan. Sinkronisasi ini dibangun berdasarkan tabel nilai akademik mahasiswa STTA Yogyakarta dengan membandingkan *record* demi *record* dalam *database*. Apabila terjadi perbedaan nilai pada tabel nilai akademik maka nilai yang berbeda tersebut akan dikembalikan seperti nilai semula dengan membandingkannya dengan menggunakan *database master*. Sinkronisasi tabel berbasis *record* ini diharapkan dapat mengatasi sistem keamanan pada *database* dan memberikan jaminan keamanan yang aman dan nyaman.

Authentication adalah suatu proses dimana *user* diidentifikasi oleh *server* sebelum *user* menggunakan jaringan. Pada proses ini, *user* meminta hak akses kepada *server* untuk menggunakan suatu jaringan. *Authorization* adalah pengalokasian layanan apa saja yang berhak diakses oleh *user* pada jaringan. *Authorization* dilakukan ketika *user* telah dinyatakan berhak untuk menggunakan jaringan. *Accounting* merupakan proses yang dilakukan oleh *server* yang mencatat semua aktivitas *user* dalam jaringan, seperti kapan *user* mulai

menggunakan jaringan, kapan *user* mengakhiri koneksinya dengan jaringan, berapa lama *user* menggunakan jaringan, berapa banyak data yang diakses *user* dari jaringan.

Hasil implementasi sinkronisasi tabel berbasis *record* menggunakan sistem keamanan *authentication*, *authorization*, *accounting* yaitu dapat mencegah terjadinya perubahan data pada *database*, mengoptimalkan penggunaan *stored*, dan meningkatkan sistem keamanan pada *database center* STTA Yogyakarta.

Kata kunci : sinkronisasi tabel, sistem keamanan *authentication*, *authorization*, *accounting*

1. Latar Belakang Masalah

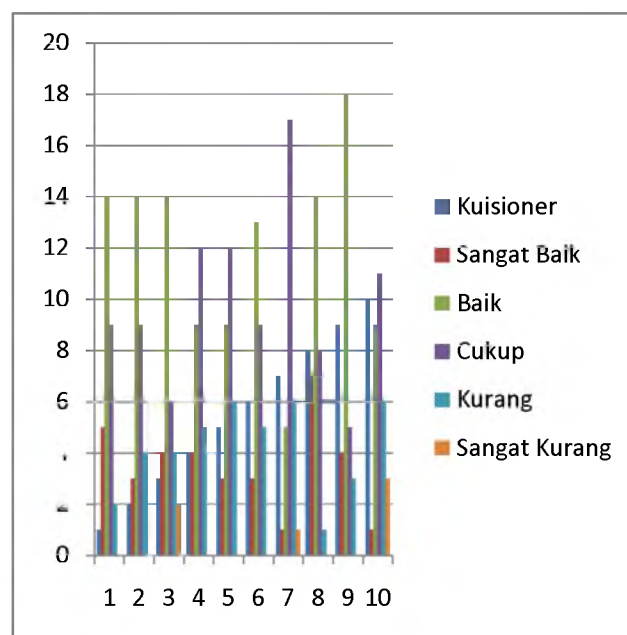
Sistem informasi akademik dapat diartikan sebagai sebuah sistem yang mampu mengintegrasikan proses-proses administrasi akademik perguruan tinggi ke dalam sebuah sistem informasi yang mampu menyimpan dan mengolah ribuan data akademik dari tahun ke tahun secara sistematis, dengan dukungan teknologi komputer dan perangkat lunak. Globalisasi dunia pendidikan menuntut perguruan tinggi untuk dapat mengelola informasi dengan baik, sehingga kebutuhan informasi masing-masing pihak yang berkepentingan dapat terpenuhi dengan cepat dan tepat. Oleh karena itu, proses penyampaian pesan, informasi, maupun pengetahuan dapat lebih cepat, mudah dan *up to date*. Salah satu sistem informasi yang ada di perguruan tinggi adalah layanan sistem informasi akademik. Layanan sistem informasi akademik di STTA (Sekolah Tinggi Teknologi Adisutjipto) Yogyakarta memberikan informasi dan layanan yang mencakup daftar nilai, jadwal kuliah, jadwal ujian, portal data diri, maupun KRS (Kartu Rencana Studi). Layanan sistem informasi ini diharapkan dapat memberikan informasi yang relevan dan *up to date* baik bagi calon mahasiswa, dosen, mahasiswa, bagian administrasi, maupun lembaga.

Salah satu sistem informasi akademik yang telah lama digunakan di perguruan tinggi adalah *software* EPSBED dari DIKTI. EPSBED atau Evaluasi Program Studi Berdasarkan Evaluasi Diri merupakan sistem evaluasi bagi program studi melalui pengumpulan data yang dilaporkan ke DIKTI pada setiap semester ganjil dan genap, didukung dengan program aplikasi EPSBED yang ringan atau program DOS (*Disk Operating System*) dan struktur data termasuk format data yang terus bertambah dari tahun ke tahun. *Database server* yang berisi data akademik di masing-masing perguruan tinggi dikelola sedemikian rupa guna mendukung layanan akademik. Salah satu contohnya adalah layanan sistem informasi akademik berbasis *web*. Layanan ini memberikan informasi seputar akademik yang dapat diakses melalui *website* resmi masing-masing perguruan tinggi menggunakan *browser*. Akan tetapi seiring dengan berkembangnya teknologi, dapat terjadi perubahan data akademik dalam *database*. Untuk mengetahui perubahan data tersebut dapat dilakukan sinkronisasi data.

Sinkronisasi adalah proses pengaturan jalannya beberapa proses pada saat yang bersamaan. Tujuan utama sinkronisasi adalah menghindari terjadinya inkonsistensi data karena pengaksesan oleh beberapa proses yang berbeda. Sinkronisasi ini dibangun berdasarkan tabel nilai akademik mahasiswa STTA Yogyakarta dengan membandingkan *record* demi *record* dalam *database*. Apabila terjadi perbedaan nilai pada tabel nilai akademik maka nilai yang berbeda tersebut akan dikembalikan seperti nilai semula dengan membandingkannya dengan menggunakan *database master*. Dengan adanya sinkronisasi

tabel berbasis *record* ini diharapkan dapat mengatasi sistem keamanan pada *database* dan memberikan jaminan keamanan yang aman dan nyaman.

Tingkat keamanan dan kenyamanan ini dilakukan analisis dengan menggunakan kuisioner. Berdasarkan hasil dari 30 lembar kuisioner yang terdiri dari 10 item pertanyaan yang telah dibagikan dan diisi oleh pengguna sistem informasi akademik STTA Yogyakarta, menyatakan bahwa tingkat akurasi informasi yang dihasilkan dari sistem informasi akademik adalah 46,6% baik. Dari sisi *interface*, pengguna menyatakan 46,6% baik. Dari sisi keamanan, pengguna menyatakan 46,6% baik. Dari sisi kelengkapan informasi yang dihasilkan, pengguna menyatakan 40% cukup. Dari sisi kenyamanan, pengguna menyatakan 40% cukup. Dari sisi partisipasi *user*, pengguna menyatakan 43,3% baik. Dari sisi pelayanan portal akademik, pengguna menyatakan 56,6% cukup. Dari sisi pemahaman *user*, pengguna menyatakan 46,6% baik.



Gambar 1 Grafik Hasil Kuisioner Persepsi Pengguna Terhadap Sistem Informasi Akademik STTA Yogyakarta

Dari sisi reliabilitas informasi yang dihasilkan terhadap informasi yang diharapkan, pengguna menyatakan 60% baik, dan dari sisi waktu akses terhadap sistem informasi akademik, pengguna menyatakan 36,6% cukup. Grafik hasil kuisioner persepsi pengguna terhadap sistem informasi akademik STTA Yogyakarta dapat dilihat pada gambar 1.

2. Landasan Teori

Sinkronisasi adalah komunikasi antara proses yang membutuhkan *place by calls* untuk mengirim dan menerima data *primitive*. Terdapat rancangan yang berbeda-beda dalam implementasi setiap *primitive*. Pengiriman pesan mungkin dapat diblok (*blocking*) atau tidak dapat diblok (*nonblocking*) juga dikenal dengan nama sinkron atau asinkron. Pengertian lain, sinkronisasi adalah proses pengaturan jalannya beberapa proses pada saat yang bersamaan. Sinkronisasi umumnya dilakukan dengan bantuan perangkat sinkronisasi. Penyelesaian terhadap masalah ini sangat penting karena perkembangan teknologi sistem

komputer menuju ke sistem *multiprocessing*, terdistribusi dan paralel yang mengharuskan adanya proses-proses konkuren.

Konkurensi adalah kondisi dimana pada saat yang bersamaan terdapat lebih dari satu proses disebut dengan konkurensi (proses-proses yang kongkuren). Proses-proses yang mengalami kongkuren dapat berdiri sendiri (*independent*) atau dapat saling berinteraksi, sehingga membutuhkan sinkronisasi atau koordinasi proses yang baik. Untuk penanganan konkuren, bahasa pemrograman saat ini telah memiliki mekanisme konkurensi dimana dalam penerapannya perlu dukungan sistem operasi dimana bahasa berada. Sinkronisasi diperlukan untuk menghindari terjadinya ketidak-konsistenan data akibat adanya akses data secara konkuren. Proses-proses disebut konkuren jika proses-proses itu ada dan berjalan pada waktu yang sama, proses-proses konkuren ini bisa bersifat *independent* atau bisa juga saling berinteraksi. Proses-proses konkuren yang saling berinteraksi memerlukan sinkronisasi agar terkendali dan juga menghasilkan *output* yang benar.

Mutual exclusion (pengeluaran timbal balik) merupakan kondisi dimana terdapat sumber daya yang tidak dapat dipakai bersama pada waktu yang bersamaan (misalnya : printer, *disk drive*). Kondisi demikian disebut sumber daya kritis, dan bagian program yang menggunakan sumber daya kritis disebut *critical region/ section*. Hanya satu program pada satu saat yang diijinkan masuk ke *critical region*. Pemrogram tidak dapat bergantung pada sistem operasi untuk memahami dan memaksakan batasan ini, karena maksud program tidak dapat diketahui oleh sistem operasi. Hanya saja, sistem operasi menyediakan layanan (*system call*) yang bertujuan untuk mencegah proses lain masuk ke *critical section* yang sedang digunakan proses tertentu. Pemrograman harus menspesifikasikan bagian-bagian *critical section*, sehingga sistem operasi akan menjaganya. Pentingnya *mutual exclusion* adalah jaminan hanya satu proses yang mengakses sumber daya pada suatu interval waktu. Mendasar untuk sistem terdistribusi adalah konkurensi dan kolaborasi di antara beberapa proses. Dalam banyak kasus, ini juga berarti bahwa proses akan perlu untuk secara simultan mengakses sumber daya yang sama. Untuk mencegah yang mengakses secara bersamaan seperti sumber daya yang korup, atau membuatnya tidak konsisten, solusi yang diperlukan untuk memberikan akses eksklusif bersama oleh proses (Tanenbaum, 2007). Pemaksaan atau pelanggaran *mutual exclusion* menimbulkan :

a. *Deadlock*

Deadlock atau pada beberapa buku disebut *Deadly Embrace* adalah keadaan dimana dua program memegang kontrol terhadap sumber daya yang dibutuhkan oleh program yang lain. Tidak ada yang bisa melanjutkan proses masing-masing sampai program yang lain memberikan sumber dayanya.

Berikut contoh ilustrasi *deadlock* :

1. Terdapat dua proses, yaitu P1 dan P2 dan dua sumber daya kritis, yaitu R1 dan R2.
2. Proses P1 dan P2 harus mengakses kedua sumber daya tersebut, dengan kondisi ini terjadi : R1 diberikan ke P1, sedangkan R2 diberikan ke P2.
3. Karena untuk melanjutkan eksekusi memerlukan kedua sumber daya sekaligus maka kedua proses akan saling menunggu sumber daya lain selamanya. Tak ada proses yang dapat melepaskan sumber daya yang telah dipegangnya karena menunggu sumber daya lain yang tak pernah diperolehnya. Kedua proses dalam kondisi *deadlock*, yang tidak dapat membuat kemajuan apapun dan *deadlock*

merupakan kondisi terparah karena dapat melibatkan banyak proses dan semuanya tidak dapat mengakhiri prosesnya secara benar.

b. *Starvation*

Starvation merupakan keadaan dimana pemberian akses bergantian terus-menerus, dan ada suatu proses yang tidak mendapatkan gilirannya.

Berikut contoh ilustrasi *starvation* :

1. Terdapat tiga proses, yaitu P1, P2 dan P3.
2. P1, P2 dan P3 memerlukan pengaksesan sumber daya R secara periodik skenario berikut terjadi.
3. P1 sedang diberi sumber daya R sedangkan P2 dan P3 diblok menunggu sumber daya R.
4. Ketika P1 keluar dari *critical section*, maka P2 dan P3 diijinkan mengakses R.
5. Asumsi P3 diberi hak akses, kemudian setelah selesai, hak akses kembali diberikan ke P1 yang saat itu kembali membutuhkan sumber daya R.
6. Jika pemberian hak akses bergantian terus-menerus antara P1 dan P3, maka P2 tidak pernah memperoleh pengaksesan sumber daya R. Dalam kondisi ini memang tidak terjadi *deadlock*, hanya saja P2 mengalami *starvation* (tidak ada kesempatan untuk dilayani).

c. *Critical Section*

Critical section adalah sebuah segmen kode di mana sebuah proses yang mana sumber daya bersama diakses, terdiri dari :

1. *Entry Section* : kode yang digunakan untuk masuk ke dalam *critical section*.
2. *Critical Section* : kode di mana hanya ada satu proses yang dapat dieksekusi pada satu waktu.
3. *Exit Section* : akhir dari *critical section*, mengizinkan proses lain.
4. *Remainder Section* : kode istirahat setelah masuk ke *critical section*.

3. Perancangan Aplikasi

Perancangan Jaringan Komputer

Sistem komputer adalah kumpulan dari elemen-elemen komputer yang terdiri dari *hardware, software, brainware* yang saling terintegrasi dan saling berinteraksi untuk melakukan pengolahan data dengan tujuan menghasilkan informasi sesuai dengan yang diharapkan. Ketiga elemen sistem komputer tersebut harus saling berhubungan dan membentuk kesatuan yang saling mendukung untuk bekerja sama. *Hardware* tidak akan berfungsi apabila tanpa *software*, demikian juga sebaliknya. Dan keduanya tidak akan bermanfaat apabila tidak ada manusia (*brainware*) yang mengoperasikan dan mengendalikannya.

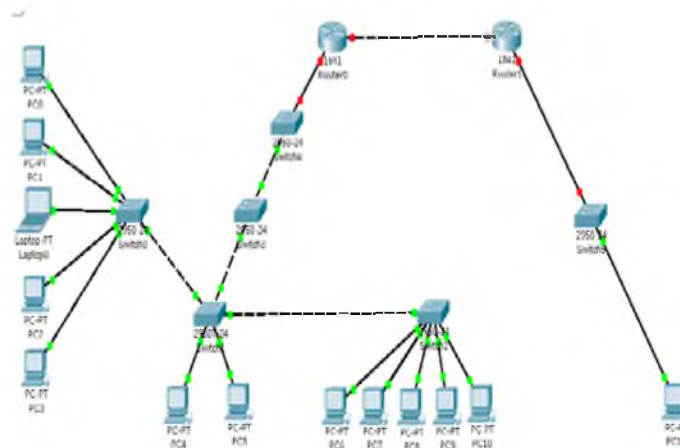
Sistem perangkat lunak sinkronisasi tabel berbasis *record* yang dibuat terdiri dari empat bagian yang saling berhubungan untuk dapat mengolah data dan menghasilkan informasi yang sesuai dengan yang diharapkan. Keempat bagian perangkat lunak tersebut penulis berikan nama sebagai berikut :

1. *Database PostgreSQL di Server*, merupakan *database server* yang menggunakan pengolahan *database PostgreSQL*. *Database* ini menyimpan informasi akademik berupa transkrip nilai dari mahasiswa STTA Yogyakarta.

2. *Database MySQL di Server*, merupakan *database server* yang menggunakan pengolahan *database MySQL*. *Database* ini menyimpan informasi akademik berupa transkrip nilai dari mahasiswa STTA Yogyakarta.
3. *Aplikasi Sinkronisasi*, aplikasi ini akan dijalankan pada salah satu komputer yang berada dalam satu jaringan dengan aplikasi *database server*. Aplikasi ini berguna untuk mensinkronisasi *database* antara *database PostgreSQL di server* dan *database MySQL di server*.

Pada perancangan infrastruktur jaringan komputer tidak hanya terpaku pada satu topologi jaringan saja. Tetapi dapat disesuaikan menurut kondisi lapangan dan kebutuhan sistem. Dalam penelitian ini menggunakan topologi star pada jaringan lokal. Media transmisi jaringan menggunakan kabel UTP.

Perancangan alur program memberikan gambaran secara visual melalui sebuah gambar untuk dapat memahami bagaimana sistem bekerja dalam sebuah area jaringan. Sistem berkolaborasi bersama dan berkomunikasi menggunakan sebuah jaringan komputer yang dihubungkan baik menggunakan media kabel maupun nirkabel dan peralatan-peralatan jaringan lainnya sebagai media komunikasi untuk dapat menghasilkan informasi yang diinginkan. Model penerapan aplikasi sinkronisasi dapat dilihat pada gambar 2.



Gambar 2 Model Penerapan Sistem Sinkronisasi

Perancangan Sistem Keamanan

Pesatnya perkembangan internet diiringi pula dengan mewabahnya berbagai *hacker*, *virus*, *spam*, *email bomb* dan sebagainya yang dapat mengganggu keamanan komputer pengguna (*user*). Oleh karena itu, pengguna perlu melakukan pencegahan terhadap masuknya hal-hal tersebut, misalnya dengan meningkatkan *security awareness* di komputer pengguna.

Pada dasarnya, sistem jaringan komputer merupakan sistem jaringan yang terbuka, artinya *user/pengguna* dalam jaringan tersebut dapat mengakses *device/resource* yang tersedia. *User* selalu ingin aman dan nyaman saat mengakses jaringan komputer. Oleh karena itu, diperlukan sebuah pengamanan jaringan yang akan melindungi aktivitas *user/ pengguna* selama di dalam jaringan.

Aplikasi sinkronisasi tabel berbasis *record* ini mengacu pada AAA (*Authentication, Authorization, Accounting*). AAA adalah sebuah model akses jaringan yang memisahkan tiga

macam fungsi kontrol, yaitu *Authentication, Authorization, dan Accounting* untuk diproses secara independen.

4. Pengujian dan Analisa

Tahap penjelasan program ini dilakukan untuk mengetahui apakah sistem yang telah dibuat benar-benar mencapai tujuan yang diinginkan dan siap untuk digunakan. Aplikasi sinkronisasi adalah sebuah aplikasi yang dibuat untuk mengecek perubahan nilai dan mengembalikan nilai tersebut menjadi nilai seperti semula apabila terjadi perubahan nilai pada *record* dalam *database* serta untuk meningkatkan keamanan pada sistem *database*. Penjelasan program aplikasi sinkronisasi meliputi menu *login user*, menu utama (*main menu*), menu pengguna (*register user*), menu sinkronisasi (*synchronization*), dan menu *about*.



Gambar 3 Tampilan Main Menu Aplikasi Sinkronisasi

Menu utama terdiri dari beberapa *sub menu* yang telah dijabarkan pada bab sebelumnya. Menu utama dapat diakses oleh *user* setelah *user* berhasil *login* dengan memasukkan *username* dan *password* yang benar. *Sub menu* yang terdapat pada *main menu* ini tidak semuanya bisa diakses oleh *user* secara bebas. Terdapat tiga hak akses yang bisa menggunakan aplikasi sinkronisasi, yaitu *admin*, *super user*, dan *user* biasa yang nanti akan dijelaskan pada *sub pembahasan manajemen user dan password* pada implementasi *form register user* untuk pengguna.

Pada menu utama ini terdapat dua tampilan menu yaitu, menu pertama merupakan menu bar yang terdiri dari menu *file*, *synchronization*, dan *about*. Sedangkan menu kedua adalah menu berupa *button* terdiri dari *synchronization*, *postgresql db*, *mysql db*, *register user*, dan *exit* yang diharapkan dapat memudahkan *user* dalam mengoperasikannya. Tampilan menu utama (*main menu*) dalam aplikasi sinkronisasi tabel berbasis *record* menggunakan sistem keamanan *authentication, authorization, accounting* (AAA) ditunjukkan pada gambar 3.

Analisa Keseluruhan

Aplikasi sinkronisasi ini bekerja dengan menggunakan dua *database* yang berbeda yaitu *database PostgreSQL* sebagai *database* utama (*primary database*) dan *database MySQL* sebagai *database* yang kedua (*secondary database*). Aplikasi sinkronisasi berjalan dengan membandingkan *field* tabel *d_transkrip* yang terdapat dalam *database PostgreSQL* dan *database MySQL*. Aplikasi akan mengecek *record per record* yang terdapat pada kedua *database*. Pertama, aplikasi mengecek apakah kedua *database* ada dan terkoneksi dengan aplikasi sinkronisasi. Apabila kedua *database* ada dan telah terkoneksi dengan aplikasi maka

selanjutnya adalah melakukan pengecekan apakah jumlah *record* antara kedua *database* sudah sama.

Apabila jumlah *record* ternyata tidak sama maka aplikasi akan menambahkan data yang belum ada tersebut ke dalam *database* dengan melakukan proses *insert* data sedangkan apabila data sudah ada tetapi berbeda maka aplikasi akan melakukan proses *update* terhadap data yang berbeda tersebut. Apabila *record* dalam kedua *database PostgreSQL* dan *database MySQL* sudah sama maka proses sinkronisasi sudah selesai dilakukan. Saat *superuser* menjalankan aplikasi sinkronisasi dengan menekan *button "Start Synchronization"* maka secara otomatis data yang diolah dalam proses sinkronisasi akan disimpan dalam sebuah *log*. *Log* merupakan berkas atau *cache* yang mampu menyimpan data-data *tracking* mengenai kinerja sebuah perangkat keras maupun perangkat lunak. *Log* ini dibuat dengan tujuan utama untuk evaluasi kinerja sebuah perangkat dalam *sub* sistem maupun untuk evaluasi kinerja sistem secara keseluruhan. *Log* pada aplikasi ini menyimpan data berupa *username*, jam, tanggal, dan operasi (berisi keterangan *field* nilai yang berubah).

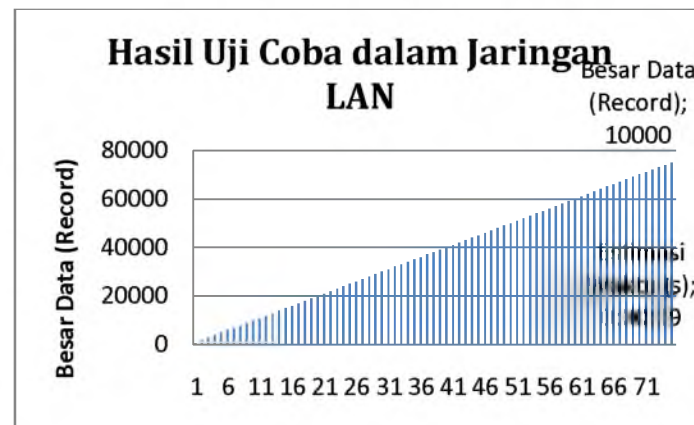
Pada aplikasi sinkronisasi ini terdapat *log security* yaitu *log* yang bekerja pada sistem keamanan *database*. *Log security* ini berfungsi sebagai media kontrol admin untuk mengetahui apakah terjadi perubahan data pada *field* dalam *database MySQL* atau tidak. *Log security* ini terdiri dari tabel (*datagrid*) yang berisi informasi berupa data nilai mahasiswa STTA Yogyakarta pada tabel *d_transkrip database MySQL* yang mengalami perubahan data. Sehingga dengan adanya *log security* ini, admin dapat mengetahui perubahan data yang terjadi pada *database MySQL* sebagai tindak lanjut admin dapat melakukan perintah untuk melakukan proses sinkronisasi *database* pada sistem. Apabila *log security* tidak memberikan informasi adanya perubahan data pada *database MySQL* maka admin tidak perlu melakukan perintah untuk melakukan proses sinkronisasi *database* pada sistem.

Tabel 1 Hasil Uji Coba dalam Jaringan LAN

No	Besar Data (Record)	Mulai	Selesai	Estima si Waktu (s)
1	1000	16:20:21	16:20:23	0:00:02
2	2000	16:23:35	16:23:39	0:00:04
3	3000	16:27:43	16:27:49	0:00:06
4	4000	16:31:04	16:31:11	0:00:07
5	5000	16:34:10	16:34:19	0:00:09
6	6000	16:37:02	16:37:13	0:00:11
7	7000	16:40:27	16:40:40	0:00:13
8	8000	16:51:32	16:51:48	0:00:16
9	9000	16:54:28	16:54:45	0:00:17
10	10000	16:58:04	16:58:23	0:00:19

Setelah uji coba ini dilakukan dalam jaringan LAN dengan menempatkan tiga komputer yang terdiri dari komputer yang berisi *database PostgreSQL*, komputer yang berisi *database MySQL*, dan komputer yang berisi aplikasi sinkronisasi yang sudah diberi alamat sehingga dapat berkomunikasi. Hasil uji coba selain dalam bentuk tabel seperti gambar tabel diatas

dari *record* 1000 sampai dengan *record* 10000, terdapat juga dalam bentuk grafik yang ditunjukkan pada gambar 4.



Gambar 4 Grafik Hasil Uji Coba Kenaikan Jumlah *Record* dengan Estimasi Waktu dalam Jaringan LAN

5. Kesimpulan

1. Perangkat lunak hasil dari rancangan dalam tugas akhir ini dapat digunakan untuk melakukan proses sinkronisasi data dengan tabel *d_transkrip* dengan dua *database* yang berbeda yaitu *database PostgreSQL* dan *database MySQL*.
2. Kecepatan untuk proses sinkronisasi data sangat tergantung dari koneksi jaringan yang dihubungkan dengan aplikasi sinkronisasi dan diperlukan *bandwidth* yang besar untuk koneksi aplikasi dengan jaringan Internet.
3. Aplikasi sinkronisasi yang dibangun menggunakan metode *Authentication, Authorization, Accounting* (AAA) setelah dilakukan proses uji coba dengan jaringan LAN dan jaringan Internet terbukti bisa digunakan untuk melakukan pengecekan data dan memperbaiki data yang mengalami perubahan pada *record* dalam *database*.

Saran

1. Penelitian dalam aplikasi sinkronisasi dapat diperluas, sebagai contoh tidak hanya tabel nilai akademik mahasiswa tetapi tabel-tabel lain seperti tabel administrasi, tabel bayar spp, tabel keuangan sehingga terpelihara keselarasan dan integritas data.
2. Pengembangan aplikasi ini sebaiknya menggunakan perangkat lunak yang berbasis *open source* sehingga penelitian dalam bidang sinkronisasi ini dapat berkembang dengan baik kedepannya.
3. Aplikasi sinkronisasi yang terkoneksi dengan jaringan Internet sebaiknya menggunakan IP dengan pengalamatan IP menggunakan sistem koneksi DHCP (*Dynamic Host Configuration Protocol*).

Referensi

- [1] Kusuma, Adi Wira, *Database Engine*, Gava Media, Yogyakarta, 2006.
- [2] Nurwono, Yuniarto, *Manajemen Database Identitas* (Terpusat, Kiasan, atau Bayangan), Andi Offset, Yogyakarta, 2005.
- [3] Sutedjo, Budi., *Konsep dan Aplikasi Pemrograman Client Server dan Sistem Terdistribusi*, Andi Offset, Yogyakarta, 2006.

- [4] Tanenbaum, Maarten., *Distributed System: Principles and Paradigms*, Second Edition, Person, 2007.
- [5] Utomo, Eko Priyo, *Pengantar Jaringan Komputer*, Yrama Widya, Bandung, 2006.
- [6] Wintolo, Hero., *Sinkronisasi Data pada Tabel yang Tersimpan di Dua Database Server yang Berbeda*, Jurnal Ilmiah Angkasa, Yogyakarta, 2010.