



OPTIMASI BASIS DATA TERDISTRIBUSI DENGAN ALGORITMA PRIORITY SCHEDULING

Mutasar¹⁾ dan Chaeroen Niesa²⁾

^{1, 2)} Universitas Islam Kebangsaan Indonesia

e-mail: ¹ mutasarstmik@gmail.com

² jeumalaniesa@gmail.com

Abstract

[**Optimization of Distributed Database with Priority Scheduling Algorithm**] Optimation time access to distributed databases in the institution / company that has many branches need information systems that are perfectly adequate for easy data access and back-up data, to ensure the availability of data on an ongoing basis is critical, the agency / company that has many branches to be integration and prioritization of data in each branch are connected directly to the central part agency. The number of incoming data simultaneously cause the database is slow to process the data that should be the priority. The process of replication, distribution and priority data is the main problem in this case. With the use of tables status / temporary and priority table is a technique for optimizing time and priority of data replication process, use Priority Scheduling algorithm to accelerate the process of priority on the distribution data. By using this method the process of replication, distribution and priority data to any branch of the agency / company can access the data directly and the data in the priority would sooner go to a central server.

Keywords: database; *priority scheduling*; prioritas.

Abstrak

Optimasi waktu akses database terdistribusi pada instansi/perusahaan yang memiliki banyak cabang membutuhkan sistem informasi yang sangat memadai, untuk kemudahan akses data dan back-up data, untuk menjamin ketersediaan data secara berkelanjutan merupakan hal yang sangat penting, instansi/perusahaan yang memiliki banyak cabang untuk dapat melakukan integrasi dan prioritas data pada masing-masing cabang yang terhubung langsung dengan bagian pusat instansi. Banyaknya data yang masuk bersamaan menyebabkan database lambat untuk memproses data yang diprioritas. Proses replikasi, distribusi dan prioritas data menjadi permasalahan utama dalam kasus ini. Dengan Penggunaan tabel status/temporary dan tabel prioritas merupakan teknik untuk mengoptimalkan waktu proses replikasi dan prioritas data, penggunaan algoritma *Priority Scheduling* untuk mempercepat proses prioritas pada distribusi data. Dengan menggunakan metode ini proses replikasi, distribusi dan prioritas data ke setiap cabang instansi/perusahaan dapat mengakses data secara langsung dan data yang di prioritas akan lebih cepat masuk ke server pusat

Kata Kunci: database; *priority scheduling*; prioritas.

1. Pendahuluan

Permasalahan optimasi waktu akses dan data yang diprioritaskan pada database terdistribusi sangat penting, permasalahan ini tidak perlu terjadi seandainya terdapat beberapa backup data baru atau slave database sebelum sampai ke server pusat. Banyaknya data yang masuk bersamaan menyebabkan database lambat

untuk memproses data yang diprioritas, pemasalahan tidak mempunya database menerima data dalam jumlah sangat banyak serta berbagai macam prosesan menjadi suatu hal yang wajar terjadi pada basisdata[1] Dalam sistem basisdata terdistribusi yang tidak menggunakan algoritma *priority scheduling* kemungkinan akan mendapatkan waktu akses yang lambat dikarenakan banyaknya data-data yang masuk ke basisdata pusat tanpa mendapatkan prioritas data. Distribusi data dalam suatu instansi terkadang memerlukan prioritas untuk data-data tertentu, seperti data pegawai, Untuk memperlancar akses data tersebut maka digunakan algoritma *priority scheduling*[2] Penelitian ini dibuat untuk menganalisa waktu akses pada database terdistribusi yang nantinya berguna untuk pengelolaan yang lebih efisien terhadap waktu proses, proses pengelolaan database, dan ketersediaan data secara kontinyu.

Dalam replikasi, digunakan suatu perangkat lunak untuk mencari atau lebih tepatnya melacak perubahan yang terjadi di satu basis data. Setelah perubahan dalam satu basis data teridentifikasi dan diketahui, baru kemudian dilakukan perubahan agar semua basis data sama satu dengan yang lainnya. Proses replikasi memakan waktu yang lama dan membebani komputer karena kompleksitas prosesnya. Sementara itu, proses duplikasi tidak sama dan tidak sekompleks replikasi. Dalam proses ini, satu basis data dijadikan master, kemudian diperbanyak menjadi sejumlah duplikat. Selama proses duplikasi berlangsung, perubahan hanya boleh dilakukan pada basis data master agar data lokal tidak tertimpa [3] *Database Management System* (DBMS) merupakan software yang digunakan untuk membangun sebuah sistem basis data yang berbasis komputerisasi. DBMS membantu dalam pemeliharaan dan pengolahan kumpulan data dalam jumlah besar. Sehingga dengan menggunakan DBMS tidak menimbulkan kekacauan dan dapat digunakan oleh pengguna sesuai dengan kebutuhan. DBMS merupakan perantara bagi pemakai dengan database. Untuk berinteraksi dengan DBMS (basis data) menggunakan bahasa basis data yang telah ditentukan oleh perusahaan DBMS. Bahasa database biasanya terdiri atas perintah-perintah yang di formulasikan sehingga perintah tersebut akan diproses oleh DBMS. Perintah-perintah biasanya ditentukan oleh *user*. Pengembangan lebih lanjut dari DBMS ini menciptakan sistem distribusi database, yang nantinya data dapat terduplikasi dan tereplikasi agar database selalu ter *up to date*.

2. Metode

2.1. Database Terdistribusi

Database terdistribusi adalah sebuah database yang diatur oleh sebuah *Database Management System* (DBMS) yang tersimpan secara fisik pada beberapa komputer dan beberapa lokasi dengan terkoneksi pada jaringan [6] Terdapat dua macam sifat dari database terdistribusi yaitu *Heterogenous* dan *Homogenous*. *Homogenous* artinya suatu database terdistribusi dimana data didistribusikan pada beberapa komputer dengan menggunakan *Database Management System* (DBMS) yang sama. DBMS digunakan pada database terdistribusi untuk melakukan koordinasi. data pada beberapa node. Sedangkan *Heterogenous* adalah kebalikan dari homogenous dimana data di sebarakan dengan menggunakan DBMS yang berbeda.

Untuk memperoleh database yang handal perlu diperhatikan hal-hal seperti keamanan data, kualitas data, kemudahan akses, kemudahan pengolahan data, dan pengembangan database tersebut. Pada *Database Management System* (DBMS) kegagalan pada suatu site akan mematikan seluruh operasional DBMS. Namun pada *Distributed Database Management System* (DDBMS) kegagalan pada salah satu site, atau kegagalan pada hubungan komunikasi dapat membuat operasional DBMS tidak dapat dijalankan (Yuanyuan et al.,2010, jika terjadi kegagalan dalam pengaksesan data pada suatu site dikarenakan jaringan komunikasi terputus maka site yang ingin mengakses data tersebut dapat mengakses pada site yang tidak mengalami kerusakan.

Dalam sistem *Distributed Database Management System* (DDBMS) pengolahan dari permintaan atau transaksi mungkin memerlukan akses data yang berbeda di antara server. Hal ini membuat waktu komunikasi yang penting dari waktu respon query atau waktu transaksi[4]

Perancangan basis data lebih kompleks sebelumnya menjadi keuntungan, tetapi karena distribusi menyebabkan masalah sinkronisasi dan koordinasi, kontrol terdistribusi menjadi kerugian atau kekurangan di masalah ini.

Karakteristik database terdistribusi:

1. Komputer yang dihubungkan menggunakan jaringan komunikasi
2. Kumpulan data yang digunakan bersama secara logic tersebar pada sejumlah komputer yang berbeda.
3. Data pada masing-masing situs dibawah kendali satu DBMS
4. Masing-masing DBMS berpartisipasi dalam satu aplikasi global.
5. Data pada masing-masing situs dapat menangani aplikasi-aplikasi lokal secara otonom.

2.2. Replikasi Database

Replikasi adalah suatu teknik untuk melakukan copy dan pendistribusian data dan objek-objek database dari satu database ke database lain dan melaksanakan sinkronisasi antara database sehingga konsistensi data dapat terjamin.

Dengan menggunakan teknik replikasi ini, data dapat didistribusikan ke lokasi yang berbeda melalui koneksi jaringan lokal maupun internet. Replikasi juga memungkinkan untuk mendukung kinerja aplikasi, penyebaran data fisik sesuai dengan penggunaannya, seperti pemrosesan transaksi online dan DSS (Desicion Support System) atau pemrosesan database terdistribusi melalui beberapa server.

2.3. Temporary Table

Temporary table adalah table yang dibuat pada saat runtime dan dapat melakukan operasi yang dilakukan oleh tabel normal (seperti insert, update dan delete). Perbedaannya dengan tabel normal adalah ketika koneksi / session ke SQL Server selesai, maka tabel sementara akan otomatis terhapus, data yang sudah diinsert akan hilang.

Tipe tabel sementara dibagi menjadi 2, yaitu :

1. Local temp table : tabel hanya dapat digunakan oleh koneksi / session yang membuat table
2. Global temp table : tabel dapat digunakan oleh koneksi / session lain selama session yang membuat table masih ada.

2.4. Algoritma Priority Scheduling

Priority Scheduling merupakan algoritma penjadwalan yang mendahulukan proses yang memiliki prioritas tertinggi. Setiap proses memiliki prioritasnya masing-masing. Prioritas suatu proses dapat ditentukan melalui beberapa karakteristik antara lain:

1. Time limit.
2. Memory requirement.
3. Akses file.
4. Perbandingan antara burst M/K dengan CPU burst.
5. Tingkat kepentingan proses.

Priority scheduling juga dapat dijalankan secara *preemptive* maupun *non-preemptive*. Pada *preemptive*, jika ada suatu proses yang baru datang memiliki prioritas yang lebih tinggi daripada proses yang sedang dijalankan, maka proses yang sedang berjalan tersebut dihentikan, lalu CPU dialihkan untuk proses yang baru datang tersebut. Sementara itu, pada *non-preemptive*, proses yang baru datang tidak dapat mengganggu proses yang sedang berjalan, tetapi hanya diletakkan di depan *queue*. Kelemahan pada *priority scheduling* adalah dapat terjadinya *indefinite blocking* (*starvation*). Suatu proses dengan prioritas yang rendah memiliki kemungkinan untuk tidak dieksekusi jika terdapat proses lain yang memiliki prioritas lebih tinggi.

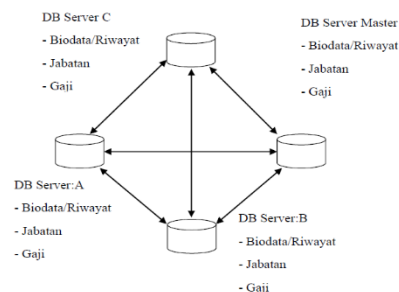
3. PEMBAHASAN

3.1. Data Yang Digunakan

Dalam penelitian ini penulis akan menganalisis waktu akses database terdistribusi dan data yang diprioritaskan menggunakan 4 database terdiri dari 1 database master dan 3 database slave yang tersebar disetiap client.

3.2. Analisa Data

Tahapan ini penulis akan menganalisa database pada Perusahaan X yaitu Database Pegawai sebagai Master, Database Biodata, Database Jabatan dan Database Gaji sebagai Slave, untuk lebih jelasnya dapat dilihat pada gambar dibawah ini:



Gambar 1. Perancangan Database Terdistribusi

3.3. Penentuan Data Prioritas

Data yang digunakan pada penelitian ini terbagi menjadi tiga jenis yaitu data biodata, jabatan dan gaji. Setiap data akan didistribusikan dari satu server menuju server yang lain. Pada distribusi konvensional atau Jurnal TIKFA Fakultas Ilmu Komputer Universitas Almuslim, Vol. 06 No. 02

distribusi biasa, tiap data didistribusikan secara bergantian tanpa memperdulikan prioritas atau tingkat kepentingan dari setiap data tersebut.

Prioritas data ditentukan oleh tingkat kebutuhan akan data tersebut dimana data tersebut diharapkan dapat tersedia dengan lebih cepat dibandingkan dengan data – data lainnya. Pada penelitian ini tiga jenis data yang digunakan memiliki tingkat kepentingan yang berbeda – beda. Tingkat kepentingan atau kebutuhan dari tiap data dapat dilihat pada tabel 1. Berikut:

Tabel 1. Tingkat Kepentingan Data

No.	Data	Tingkat Kepentingan / Kebutuhan
1	Biodata	Sangat Penting / Dibutuhkan Tersedia Secepatnya.
2	Jabatan	Penting / Dibutuhkan Tersedia Secepatnya.
3	Gaji	Penting / Dibutuhkan Tersedia Secepatnya.

Pada penelitian ini tingkat prioritas dari setiap jenis data dibagi menjadi rentang nilai antara 1 sampai 10, dimana data dengan nilai prioritas paling tinggi diberikan nilai 1 sedangkan data dengan nilai prioritas paling tinggi diberikan nilai 10, sedangkan data dengan tingkat prioritas diantaranya diberikan nilai 2 – 9 bergantung kepada kebutuhan data tersebut. Pemilihan nilai rentang 1 – 10 yang digunakan pada penelitian ini merupakan hasil perkiraan dimana data yang digunakan hanyalah tiga jenis data saja sehingga penggunaan rentang yang kecil dari 1 – 10 sudah cukup memenuhi untuk kebutuhan membedakan prioritas dari satu data dengan data yang lainnya. Nilai prioritas berlaku untuk setiap jenis data dan bukanlah setiap data, sebagai contoh jika jenis data biodata diberikan nilai prioritas 1 maka setiap data biodata yang masuk berapapun jumlahnya maka akan memiliki nilai prioritas 1 begitu juga dengan jenis data yang lain yang mana setiap data individu akan memiliki nilai prioritas sesuai dengan prioritas dari jenis datanya.

Tingkat kepentingan dipengaruhi oleh kebutuhan akan data tersebut bagi server – server lainnya. dapat dilihat pada table 1. data biodata memiliki tingkat kepentingan paling tinggi sehingga pada penelitian ini data biodata diberikan nilai prioritas 1. Data jabatan merupakan data yang paling rendah prioritas nya dan pada penelitian ini diberikan nilai 3, sedangkan data gaji memiliki tingkat kepentingan yang lebih tinggi dibandingkan dengan data jabatan sehingga diberikan nilai 2. Penentuan nilai prioritas dari data dapat dilihat pada tabel 2. berikut.

Tabel 2. Penentuan Nilai Prioritas Data

No.	Data	Tingkat Kepentingan / Kebutuhan	Nilai Prioritas
1	Biodata	Sangat Penting / Dibutuhkan Tersedia Secepatnya.	1
2	Jabatan	Penting / Dibutuhkan Tersedia Secepatnya.	3
3	Gaji	Penting / Dibutuhkan Tersedia Secepatnya.	2

3.4. Perhitungan Waktu Akses

Waktu akses merupakan hal yang paling penting dalam penelitian distribusi database. Dimana pengukuran kinerja dari teknik yang diterapkan pada sebuah distribusi database salah satunya adalah variabel waktu akses. Waktu akses dapat menggambarkan waktu yang dibutuhkan dari data tersebut di inputkan di server yang lain sampai data tersebut dapat di *query* di server lokal.

Secara umum waktu akses diperoleh dari penjumlahan dari waktu tunggu dengan waktu proses, dimana waktu tunggu adalah waktu yang dibutuhkan oleh data tersebut selama di antrian sedangkan waktu proses adalah waktu yang dibutuhkan dalam proses replikasi data tersebut dari server dimana data tersebut berada ke server lainnya.

Sebagai contoh, pada server A terdapat tiga jenis data biodata, gaji dan jabatan yang masing – masing memiliki jumlah *record* yang berbeda – beda yang akan di distribusikan ke server B. diketahui jumlah *record* pada data biodata adalah 10 *record*, data gaji adalah 34 *record* dan data jabatan 12 *record*. Waktu akses dari tiap jenis data memiliki perbedaan bergantung pada urutan replikasi yang digunakan. Pada distribusi konvensional proses distribusi dilakukan secara berurutan yang mana diketahui urutan pertama adalah biodata, di-ikuti data jabatan dan kemudian data gaji. Sehingga waktu akses dari tiap jenis data dapat dijabarkan sebagai berikut:

Diketahui :

Biodata = 40 *record*

Jabatan = 30 *record*

Gaji = 34 record

Urutan distribusi : Biodata, Jabatan, Gaji.

Asumsikan waktu proses adalah t untuk setiap record sehingga :

Waktu Proses Biodata = $40t$

Waktu Proses Jabatan = $30t$

Waktu Proses Gaji = $34t$

Waktu Akses :

Dikarenakan distribusi dilakukan secara berurutan mulai dari biodata maka waktu tunggu biodata adalah 0. Waktu tunggu data jabatan adalah sebanyak waktu proses data biodata yaitu $40t$ dan waktu tunggu data gaji adalah waktu proses data biodata ditambah waktu proses data jabatan yaitu $30t$.

Biodata

Waktu Proses = $40t$, Waktu Tunggu = $0t$, Sehingga waktu akses = $40t$

Jabatan

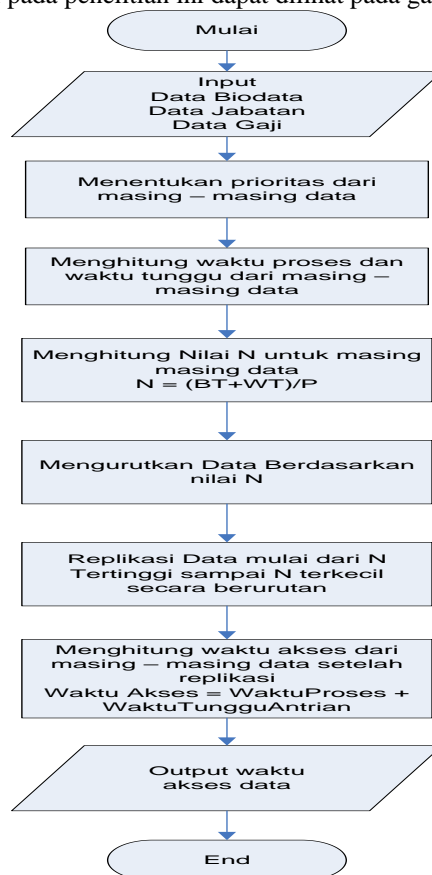
Waktu Proses = $30t$, Waktu Tunggu = $40t$, Sehingga waktu akses = $70t$

Gaji

Waktu Proses = $34t$, Waktu Tunggu = $70t$, Sehingga waktu akses = $104t$

3.5. Algoritma Priority Scheduling

Berdasarkan analisa terhadap proses *priority scheduling* yang telah dijabarkan, maka aliran proses *priority scheduling* yang akan digunakan pada penelitian ini dapat dilihat pada gambar 4. berikut ini :



Gambar 4. Penentuan waktu akses dan prioritas data

3.6. Analisis Waktu Akses

Pada distribusi database menggunakan *priority scheduling*, waktu akses dapat diminimalisir untuk data – data yang memiliki tingkat prioritas yang lebih tinggi, sehingga data dengan prioritas lebih tinggi dapat diakses lebih cepat dibandingkan dengan data dengan prioritas yang lebih rendah.

Sebagai contoh, prioritas data yang digunakan adalah prioritas data yang telah ditentukan yang dapat dilihat pada tabel 3.2. Dengan menggunakan jumlah *record* yang sama, maka waktu akses untuk tiap data dengan *priority scheduling* dapat dijabarkan sebagai berikut :

Diketahui :

Biodata = 40 *record*, waktu proses = 40t, waktu tunggu = 0

Jabatan = 30 *record*, waktu proses = 30t, waktu tunggu = 0

Gaji = 34 *record*, waktu proses = 34t, waktu tunggu = 0

Urutan Distribusi menggunakan sistem penilaian (N) dijabarkan sebagai berikut :

$$NBiodata = \frac{Waktu\ proses + Waktu\ Tunggu}{Prioritas} = \frac{40 + 0}{1} = 40$$

$$NJabatan = \frac{Waktu\ proses + Waktu\ Tunggu}{Prioritas} = \frac{30 + 0}{3} = 10$$

$$NGaji = \frac{Waktu\ proses + Waktu\ Tunggu}{Prioritas} = \frac{34 + 0}{2} = 17$$

Berdasarkan perhitungan diatas diperoleh bahwa NBiodata bernilai paling tinggi sehingga data Biodata diproses terlebih dahulu. Sehingga untuk data Jabatan dan data Gaji diperoleh waktu tunggu sebesar waktu proses data biodata :

Waktu Tunggu Gaji = 40t

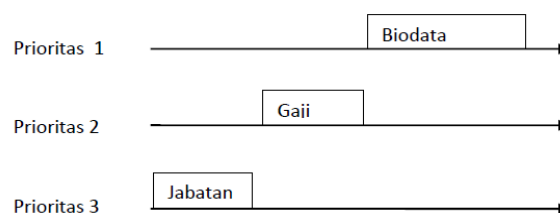
Waktu Tunggu Jabatan = 40t

Berikutnya menghitung nilai N untuk data gaji dan jabatan untuk memperoleh urutan kedua pada saat distribusi :

$$NJabatan = \frac{Waktu\ proses + Waktu\ Tunggu}{Prioritas} = \frac{30 + 40}{3} = 43$$

$$NGaji = \frac{Waktu\ proses + Waktu\ Tunggu}{Prioritas} = \frac{34 + 40}{2} = 54$$

Dari perhitungan diatas diperoleh data gaji memiliki nilai N yang lebih besar dari data jabatan sehingga diperoleh urutan distribusi :



Gambar 11. Prioritas Data

Sehingga berdasarkan komputasi *priority scheduling* diatas dapat dilihat pada tabel 3. sebagai berikut :

Tabel 3. Tabel Antrian Proses

No.	Nama	Predesesor	Lama Proses	Waktu Tunggu	Waktu Proses	Waktu Selesai
1	Biodata	-	40t	0t	0t	40t
2	Gaji	Biodata	34t	40t	40t	74t
3	Jabatan	Gaji	30t	74t	74t	104t

Berdasarkan komputasi *priority scheduling* pada Tabel 3. dapat digambarkan diagram antrian proses sebagai berikut :

Distribusi data <i>Priority Scheduling</i>																					
Waktu(t)	0	5	10	25	30	35	40	45	50	54	60	65	74	75	80	85	90	95	104		
Biodata	█																				
Gaji								█													
Jabatan																			█		

Gambar 12. Diagram distribusi data prioritas Scheduling

Berdasarkan urutan distribusi diatas maka dapat dihitung waktu akses dari tiap data adalah sebagai berikut :

Biodata
Waktu Proses = 40t, Waktu Tunggu = 0t, Sehingga waktu akses = 40t

Gaji
Waktu Proses = 34t, Waktu Tunggu = 40t, Sehingga waktu akses = 74t

Jabatan
Waktu Proses = 30t, Waktu Tunggu = 74t, Sehingga waktu akses = 104t

Berdasarkan simulasi diatas total waktu distribusi adalah sama dengan waktu distribusi konvensional yaitu **104t**, namun data biodata yang memiliki prioritas lebih tinggi dibandingkan data gaji dan data jabatan sehingga memiliki waktu akses yang jauh lebih cepat dibandingkan menggunakan distribusi konvensional yang mana perbandingannya dapat dilihat sebagai berikut :

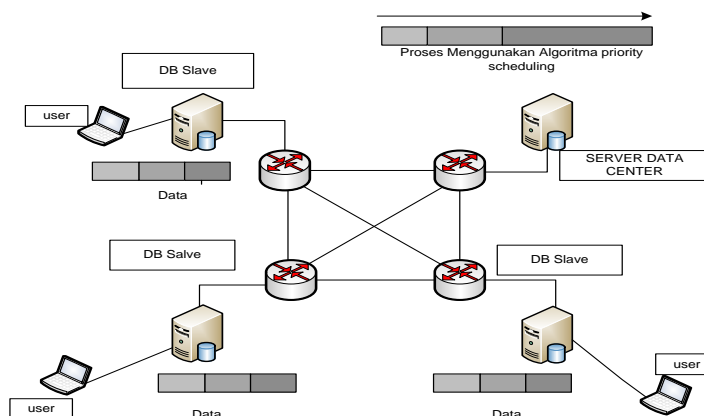
Tabel 4. Perbandingan Waktu Akses

No.	Data	Waktu Akses Distribusi	Waktu Akses Priority Scheduling
1	Biodata	40 t	44t
2	Gaji	34t	74t
3	Jabatan	30t	104t

Pada tabel 4. dapat dilihat bahwa data biodata memiliki waktu akses yang lebih cepat dimana di-distribusikan terlebih dahulu dibandingkan dengan data gaji. Namun data jabatan memiliki waktu akses yang jauh lebih lambat dikarenakan data biodata memiliki nilai waktu proses yang jauh lebih tinggi sehingga menyebabkan waktu tunggu yang cukup besar bagi data jabatan.

3.7. Alur Proses Scheduling

Pada Proses distribusi database dari server client ke server pusat menggunakan algoritma *Priority Scheduling*. Tiap data yang masuk dari masing-masing server akan diprioritaskan tingkat kepentingan data tersebut sebelum masuk ke server pusat, misalkan data biodata mendapatkan prioritas yang pertama masuk ke server pusat sedangkan prioritas kedua adalah data gaji dan ketiga data jabatan terakhir masuk. Untuk lebih jelasnya dapat dilihat pada gambar 14. dibawah ini:



Gambar 14. Alur Proses *Priority Scheduling*

Keterangan:

	: Data Biodata
	: Data Gaji
	: Data Jabatan

Data yang diinputkan oleh user seperti data biodata ,data gaji dan data jabatan akan masuk ke database slave, data-data tersebut akan direplikasi dan diprioritaskan untuk didistribusikan ke server pusat.

4.Hasil dan Pembahasan

4.1. Implementasi Distribusi Database Yang Dikembangkan

Distribusi database pada sistem yang telah dikembangkan pada penelitian ini dapat di-ilustrasikan menggunakan data – data sample. Informasi yang pertama sekali dibutuhkan adalah informasi prioritas dari tiap data yang mana dapat diasumsikan sebagai berikut :

Tabel 5. Contoh Prioritas Data

Data	Prioritas
Biodata	1
Jabatan	6
Gaji	3

Nilai prioritas menunjukkan tingkat kepentingan dari tiap jenis data dimana data dengan prioritas paling penting diberikan dengan nilai 1, sedangkan data dengan prioritas tidak penting diberikan dengan nilai 10. Berikutnya adalah menentukan waktu tunggu dan waktu proses dari tiap data yang akan didistribusikan, sebagai contoh data Biodata, jabatan dan gaji dapat dilihat pada tabel berikut :

Tabel 6. Biodata

nip	nama	kelamin	tanggal_lahir	tempat_lahir	agama	pendidikan	alamat
0102	Budi	Laki-Laki	5/26/2016	Medan	Islam	S1	asid
0103	Andi	Laki-Laki	5/26/2016	Medan	Islam	S1	Jl Pahlawan
0104	Joko	Laki-Laki	5/26/2016	Medan	Islam	D3	Jl Halat

Tabel 7. Jabatan

nip	tmt	golongan	jabatan
0102	5/26/2016	IIIA	Staf
0103	5/26/2016	IIIA	Staf IT
0104	5/26/2016	IID	Staf

Tabel 8. Gaji

nip	tahun	bulan	tanggal	gaji	tunjangan	total
0102	2016	5	5/26/2016	1500000	200000	1700000
0103	2016	5	5/26/2016	2000000	200000	2200000
0104	2016	5	5/26/2016	1500000	100000	1600000

Proses berikutnya adalah menentukan nilai N dari setiap data dengan kalkulasi sebagai berikut :

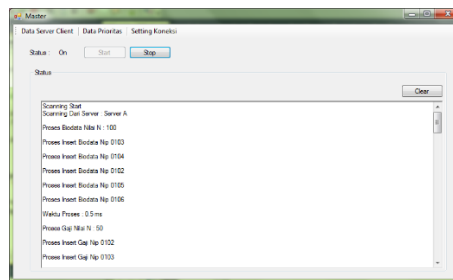
Tabel 9. menggunakan waktu proses

Data	Atribut	Nilai
Biodata	Prioritas	1
	Waktu Proses	3
	Waktu Tunggu	0
	Nilai N	$(3 + 0) / 1 = 3$
Jabatan	Prioritas	6
	Waktu Proses	3
	Waktu Tunggu	0
	Nilai N	$(3 + 0) / 6 = 0,5$
Gaji	Prioritas	3
	Waktu Proses	3
	Waktu Tunggu	0
	Nilai N	$(3 + 0) / 3 = 1$

Proses kalkulasi nilai N seperti yang dapat dilihat pada tabel 9. menggunakan waktu proses yang diperoleh dari jumlah record yang akan diproses. Waktu tunggu pada saat perhitungan awal adalah 0 karena semua data memiliki.

4.2. Hasil

Setelah hasil pengembangan dan simulasi yang dilakukan maka pada bagian ini akan dipaparkan hasil simulasi dan pengujian terhadap sistem distribusi dengan memanfaatkan *priority scheduling* yang dikembangkan pada penelitian ini. Untuk memperoleh hasil perbandingan kinerja maka perbandingan akan dilakukan dengan menggunakan waktu distribusi tanpa menggunakan *priority scheduling* dan waktu distribusi menggunakan *priority scheduling*. Analisa waktu distribusi database tanpa menggunakan *priority scheduling* dilakukan dengan menggunakan jumlah data yang sama dengan distribusi database yang menggunakan *priority scheduling*. Implementasi distribusi database tanpa menggunakan *priority scheduling* dapat dilihat pada gambar 15. berikut ini.

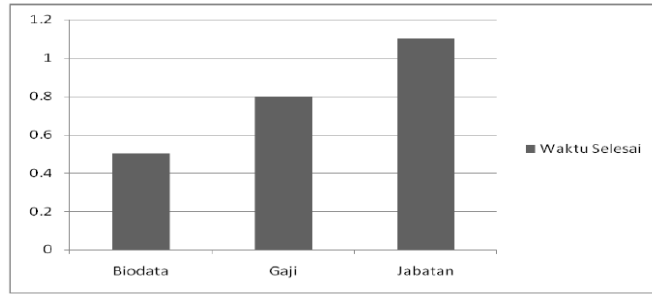


Gambar 15. Implementasi Distribusi Database Tanpa Menggunakan Priority Scheduling

Pada gambar 10 dapat dilihat proses distribusi database tanpa menggunakan *priority scheduling* dimana proses replikasi dilakukan secara bergantian tanpa memperhatikan prioritas dari masing – masing data, hasil analisa waktu proses replikasi tanpa menggunakan *priority scheduling* dapat dilihat pada tabel 10 berikut :

Tabel 10. Waktu Selesai Replikasi Data Tanpa Menggunakan Priority Scheduling

Data	Jumlah Data	Waktu Selesai
Biodata	5 Data	0.5 ms
Gaji	3 Data	0.8 ms
Jabatan	3 Data	1.1 ms

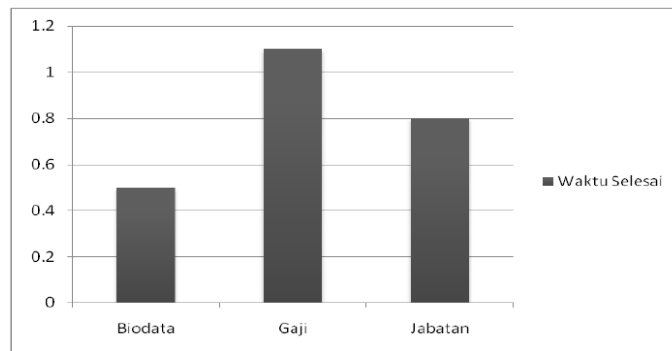


Gambar 16. Grafik Replikasi Data Tanpa Menggunakan Priority Scheduling

Hasil yang diukur pada distribusi database menggunakan priority scheduling berdasarkan implementasi yang telah dilakukan sebelumnya adalah sebagai berikut :

Tabel 11. Waktu Selesai Replikasi Data Menggunakan Priority Scheduling

Data	Jumlah Data	Prioritas	Waktu Selesai
Biodata	5 Data	1	0.5 ms
Gaji	3 Data	4	1.1 ms
Jabatan	3 Data	2	0.8 ms



Gambar 17. Grafik Replikasi Data Menggunakan Priority Scheduling

Berdasarkan waktu selesai replikasi yang diperoleh dari proses replikasi dengan atau tanpa menggunakan *priority scheduling*, dapat dilihat pada dasarnya waktu selesai keseluruhan proses adalah sama, namun perbedaan yang sangat terlihat adalah replikasi data menggunakan *priority scheduling* membuat data yang memiliki tingkat kebutuhan yang tinggi akan tersedia lebih cepat dibandingkan dengan data yang tingkat kebutuhannya lebih rendah. Perbandingan kedua implementasi yang berbeda dapat dilihat pada tabel 12. berikut :

Tabel 12. Perbandingan Waktu Distribusi

Data	Jumlah Data	Prioritas	Priority Scheduling	Normal
Biodata	5 Data	1	0.5 ms	0.5 ms
Gaji	3 Data	4	1.1 ms	0.8 ms
Jabatan	3 Data	2	0.8 ms	1.1 ms

Pada tabel 4.14 dapat dilihat bahwa waktu selesai keseluruhan proses replikasi adalah 1.1 ms untuk *priority scheduling* dan replikasi normal. Namun data jabatan memiliki waktu selesai yang lebih cepat dalam waktu 0.8 ms dibandingkan dengan distribusi normal dimana prioritas data jabatan lebih besar dibandingkan dengan

data gaji yang pada akhirnya di distribusikan pada urutan terakhir. Hal ini menunjukkan bahwa distribusi database menggunakan *priority scheduling* dapat memilih data – data dengan tingkat kebutuhan yang tinggi sehingga data – data tersebut memiliki waktu ketersediaan yang lebih cepat dibandingkan replikasi biasa.

IV. KESIMPULAN

Dari hasil penelitian yang dilakukan didapatkan kesimpulan sebagai berikut :

1. Proses distribusi data menggunakan algoritma *priority scheduling* yang telah dilakukan pada penelitian ini mampu menyeleksi data – data yang akan direplikasi berdasarkan prioritas dari masing – masing data tersebut. Pengujian yang dilakukan menunjukkan proses replikasi dilakukan secara berurutan diperoleh dari komputasi nilai N dari persamaan nilai prioritas dengan waktu proses dan waktu tunggu.
2. Proses *priority scheduling* pada dasarnya mengatur distribusi data keseluruhan dari proses replikasi data karena menentukan urutan data yang akan diproses sesuai dengan prioritas nya.
3. Proses *priority scheduling* meningkatkan ketersediaan data yang akan digunakan pada server – server yang lain dimana data – data dengan prioritas tinggi di replikasi lebih dahulu sehingga data – data tersebut tersedia dan dapat diakses dengan cepat pada server lain.
4. Proses replikasi menggunakan *priority scheduling* memberikan waktu tambahan tersendiri dikarenakan proses untuk menghitung urutan eksekusi replikasi data berdasarkan prioritasnya sehingga secara lokal data lebih cepat tersedia namun secara global proses replikasi menjadi sedikit lebih lambat karena tambahan proses tersebut.

Berdasarkan hasil analisis dengan algoritma apriori dan pengujian dengan program maka penulis menarik kesimpulan sehingga dapat bermanfaat dan berguna bagi pembaca, adapun kesimpulan sebagai berikut:

1. Dengan algoritma apriori dan pengujian dengan aplikasi program dapat menghasilkan pola kombinasi itemsets dan rules sebagai ilmu pengetahuan
2. Teknik data mining dengan algoritma apriori dapat diterapkan pada sistem penjualan.

DAFTAR PUSTAKA

- Ahmad, I., Karlapalem, K., Kwok, Y. K., & So, S. K. (2002). Evolutionary algorithms for allocating data in distributed database systems. *Distributed and Parallel Databases*, 11(1), 5-32.
- Abdalla, H. I., & Amer, A. A. (2012, March). Dynamic horizontal fragmentation, replication and allocation model in DDBSs. In *Information Technology and e-Services (ICITeS), 2012 International Conference on* (pp. 1-7). IEEE.
- Bhuyar, R. P., & Gawande, A. D. (2012). Distributed Database: Fragmentation and Allocation. *Journal of Data Mining and Knowledge Discovery*, 3(1), 58.[4] Bhura, S. A., Mahamune, A., & Alvi, A. S. (2015, February). Limited Preemptive Disk Scheduling for Real Time Database System. In *Computing Communication Control and Automation (ICCUBEA), 2015 International Conference on* (pp. 362-366). IEEE.
- Firoozy-Najafabadi, H. R., & Navin, A. H. (2012, October). Rule scheduling methods in active database systems: A brief survey. In *Application of Information and Communication Technologies (AICT), 2012 6th International Conference on* (pp. 1-5). IEEE.
- Furkhat, T. (2009). Network Issues in Clock Synchronization on Distributed Database. *Network*, 2(2).
- Jannah, M., Lolita, B., & Hustinawati, H. (2012). Analisis Metode Alokasi Fragmen Pada Sistem Basis Data Terdistribusi.
- Lehoczky, J. P. (1990, December). Fixed Priority Scheduling of Periodic Task Sets with Arbitrary Deadlines. In *RTSS* (Vol. 90, pp. 201-209).
- Mahmood, A., & Rashid, I. (2011, November). Comparison of load balancing algorithms for clustered web servers. In *Information Technology and Multimedia (ICIM), 2011 International Conference on* (pp. 1-6). IEEE.