EKSEKUSI PARALEL OPERATOR RELASI DARI TRANSAKSI PADA BASIS DATA MODEL RELASI

*Dr. Ing. Ir. Benhard Sitohang**

SARI

Waktu tanggap SMBDR dicoba diperbaiki dengan mengantisipasi eksekusi seluruh operator relasi dari transaksi **secara paralel**. Pada makalah ini, dijelaskan pendekatan yang memungkinkan eksekusi secara paralel (berupa usulan pendefinisian **operator bebas**), uji-coba, serta analisis untuk mengetahui sejauh mana perbaikan tersebut dapat dicapai. Hasil uji-coba pada sistem komputer PDP-11/44, sistem operasi Xenix, digunakan mengidentifikasi kendala-kendala yang berperan dalam menentukan perbaikan termaksud. Waktu tanggap beberapa transaksi uji-coba yang dieksekusi secara paralel ternyata lebih baik daripada eksekusi secara sekuensial (reduksi : 11,36 %). Beberapa transaksi uji coba lainnya kurang memberikan hasil yang mendukung hipotesa (reduksi : -60,87 %).

ABSTRACT

Improvement of RDBMS response-time was tried by anticipating to execute all relational operators of the transaction as a parallel process. This paper explains an approach which enables parallel execution (by defining "free operators"), test-case and analysis to know how far the improvement can be attained. Test-case result on PDP-11/44, under Xenix, was used to identify others constraints which plays role to the improvement. Response-time of test-case wich parallel-execution performed better then sequential-execution (reduction : 11,36 %). Others test-case transactions don't present satisfying result to the hypotheses (reduction : -60,87 %).

LABORATORIUM BASIS DATA & SISTEM MANAJEMEN INFORMASI
Jurusan Teknik Informatika, FTI-ITB, Jl. Ganesa No.10 - Bandung 40132

I. PENDAHULUAN

I.1. WAKTU TANGGAP SMBDR

Waktu tanggap (time response) dari Sistem Manajemen Basis Data Relasional [SMBDR] relatif lambat. Bahkan lebih lambat lagi, bila SMBDR digunakan oleh beberapa pemakai sekaligus (multi-users). Degradasi waktu tanggap tersebut, sebagai akibat dari ketidak sesuaian antara karakteristik SMBDR dengan sistem komputer yang umum digunakan.

Beberapa perbedaan karakteristik tersebut, adalah :

1. Konsep proses pada prosesor (termasuk mikro komputer) didasarkan pada operasi numerik, yang biasa disebut sebagai "Numerical Processor". Pada sisi lain, SMBDR melakukan operasi teks, dapat disebut sebagai "Text Processor". Sebagai akibat dari perbedaan ini, operasi pada SMBDR tidak dikenal sebagai operasi primitif. Dengan demikian operasi teks harus terlebih dahulu ditransformasi menjadi operasi numerik.
2. Metoda akses (pencarian) memori sekunder dan memori utama tidak didasarkan atas semantik dari data, tetapi dilakukan dengan menggunakan alamat fisik data. Hal ini menimbulkan kesulitan pada akses data pada memori, yang selanjutnya berakibat pada operasi Masukan / Keluaran yang relatif dominan.

I.2. USAHA UNTUK MEMPERBAIKI WAKTU TANGGAP SMBDR

Beberapa penelitian yang sedang dilaksanakan (merupakan topik penelitian yang dominan saat ini dalam aspek Basis Data, disamping Sistem Manajemen Basis Data Terdistribusi), dimaksudkan untuk memperbaiki kedua kelemahan tersebut diatas. Beberapa alternatif pendekatan yang sedang dilakukan adalah penerapan [STA 88] :

1. Intelligent Secondary Storage Device
   Contoh : CASSM, RAP, RARES
2. Data Base Filter on I/O channel
   Contoh : CAPS, SURE, VERSO
3. Multi processor Data Base Computer
   Contoh : MICRONET, XDMS, EDC
4. Text Processors
   Contoh : GESCAN, EUREKA
5. Associative Memory System
   Contoh : LOGIC MEMORY, ASP, STARAN

I.3. HIPOTESA DASAR

Sesuai dengan prinsip dan konsep basis data model relasi, pemrosesan transaksi dilaksanakan dengan terlebih dahulu menjabarkannya dalam bentuk deretan (struktur pohon) operator relasi (unary : projection, selection, dan binary: produk kartesian, join, quotient, natural join, union, pengurangan, interseksi). Dengan adanya beberapa operator yang tidak saling bergantungan secara langsung (mis.: operator relasi pada daun struktur pohon), dibuat satu hipotesa sebagai berikut :

> **operator relasi dari traksaksi dapat dieksekusi secara paralel (paralel processing), jika setiap operator relasi dari transaksi didefinisikan sebagai satu proses.** Dampak positif yang dapat diharapkan, adalah perbaikan waktu tanggap (time response) dari eksekusi transaksi termaksud [SIT 85]. Tergantung dari struktur transaksi, beberapa operator (sebagai terjemahan dari transaksi) dapat dieksekusi secara tidak bergantungan.

Didasarkan pada hipotesa ini, semua operator dari satu transaksi (khususnya operator yang tidak saling bergantungan) dapat dieksekusi secara paralel dengan memanfaatkan sistem operasi multi-programming, atau lebih baik lagi, dengan sistem komputer yang dibangun dengan pemroses banyak (multi processor).

I.4. ALTERNATIF PERBAIKAN & KRITERIA LAIN

Uji-coba dan analisis telah dilaksanakan, yang dimaksudkan untuk lebih memperbaiki waktu tanggap (sebagai objektif akhir), yang didasarkan atas hipotesa tersebut diatas. Aktifitas awal yang telah dilakukan, adalah mendefinisikan ketidak tergantungan antara operator relasi pada transaksi (untuk itu, didefinisikan **operator bebas, kelompok operator bebas** dan **derajat kebebasan**), dan kemudian mencoba (dengan implementasi) eksekusi secara paralel semua operator yang tidak bergantungan (operator bebas), dengan memanfaatkan sistem operasi multi programming. Hasil dari penelitian tersebut adalah merupakan pokok bahasan pada tulisan ini.

Dengan hasil positif yang didapatkan dari penelitian tersebut diatas, maka prinsip eksekusi paralel yang dimaksudkan juga diperluas, dengan menerapkan konsep multi processor yang dibangun dari mikro komputer (penelitian lanjutan, yang tidak dibahas dalam tulisan ini). Untuk pendekatan terakhir ini, dilakukan dengan mengantisipasi kriteria tambahan yang dititik beratkan dari faktor biaya, yaitu :

> **Menggunakan perangkat keras yang sudah umum dikenal, dan dapat digunakan dengan investasi yang relatif murah.**

Sebagai kesimpulan, akan dijelaskan hasil dan kendala yang dihadapi, beserta evaluasi untuk pengembangan lebih lanjut.

II. TRANSAKSI PADA BASIS DATA

Transaksi pada basis data adalah merupakan penjabaran dari kebutuhan akan pengaksesan basis data. Transaksi biasanya dijabarkan dalam sintaks tertentu (tergantung dari SMBD), dan diaktifkan secara sendiri, atau diintegrasikan dalam bahasa pemrograman (host language).

Beberapa sintaks transaksi pada SMBDR yang sudah dikenal [ULL 88], antara lain Algebric language, Structure Query Language / SQL (pada saat ini secara de-facto dianggap sebagai standard untuk klassifikasi non-procedural language), Query By Example [QBE], QUEry Language (QUEL), dll. Keseluruhan sintaks yang disebutkan diatas, secara prinsip menekankan karakteristik "user-friendly", dan secara keseluruhan dapat dianggap sama, jika dilihat dari sisi pemrosesan secara internal pada SMBDR yang dimaksudkan.

Sebagaimana disebutkan diatas, perbedaan hanya terletak pada sintaks, kata kunci yang digunakan, cara penjabaran kebutuhan (SQL, QUEL, dll. dengan kalimat, sedangkan QBE dengan tabel, dan Algebric Language dengan mengunakan simbol operator relasi), dan lain-lain perbedaan, yang pada dasarnya dimaksudkan untuk lebih memberikan kemudahan bagi pemakai. Seluruh sintaks yang disebutkan diatas termasuk dalam kelompok non procedural.

III. PENJABARAN TRANSAKSI DENGAN OPERATOR RELASI

Pada pemrosesannya, seluruh sintaks yang disebutkan terdahulu [secara internal] dijabarkan dalam bentuk struktur pohon dari operator relasi untuk transaksi, dengan kodifikasi yang disesuikan dengan kebutuhan proses.

Berikut ini adalah contoh penjabarannya, sedangkan untuk mendukung pemahaman terhadap strutur pohon, maka deskripsi basis data yang digunakan sebagai contoh, dijelaskan pada Lampiran-1.

Contoh-1 :

SQL : SELECT SNAME, ITEM, PRICE FROM SUPPLIERS, ORDERS
WHERE NAME = 'Ali Baba' AND SUPPLIERS.ITEM = ORDERS.ITEM

Arti : Tuliskan semua SNAME, ITEM & PRICE dari setiap produk yang dipesan oleh pemesan bernama 'Ali Baba'.

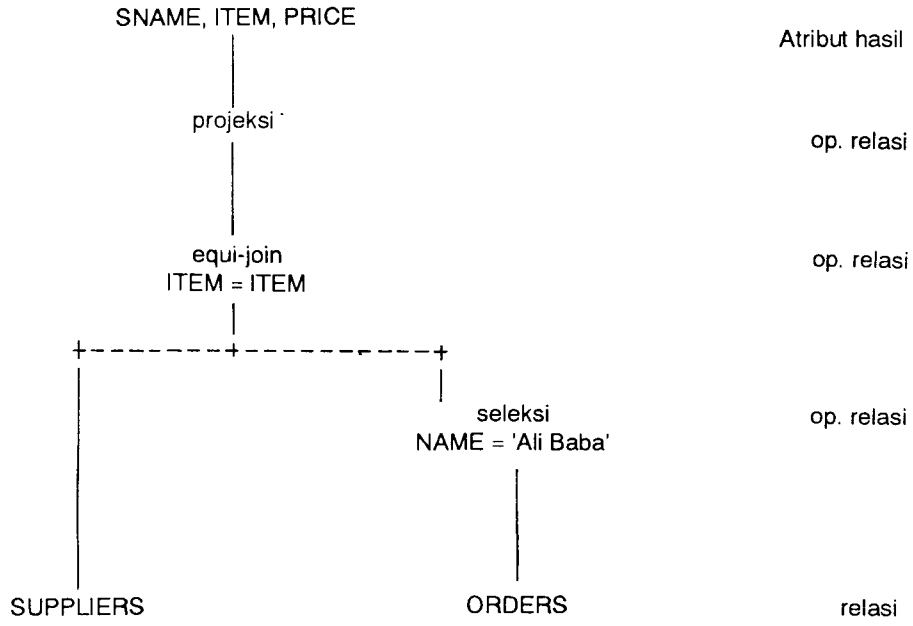
Penjabaran dengan struktur pohon operator relasi adalah seperti pada Gambar-1.

Contoh-2 :

SQL : SELECT MEMBER_CODE, NAME FROM MEMBERS WHERE 10 <=
SELECT SUM(QUANTITY) FROM ORDERS
WHERE MEMBER_CODE = MEMBERS.

Arti : Tuliskan semua MEMBER_CODE dan NAME dari pemesan, yang mempunyai pesanan lebih besar atau sama dengan 10 unit.

Penjabaran dengan struktur pohon operator relasi adalah seperti pada Gambar-2.



Gambar-1 Struktur Pohon Transaksi Contoh-1

IV. OPERATOR BEBAS, KELOMPOK OPERATOR BEBAS SERTA DERAJAT KEBEBASAN

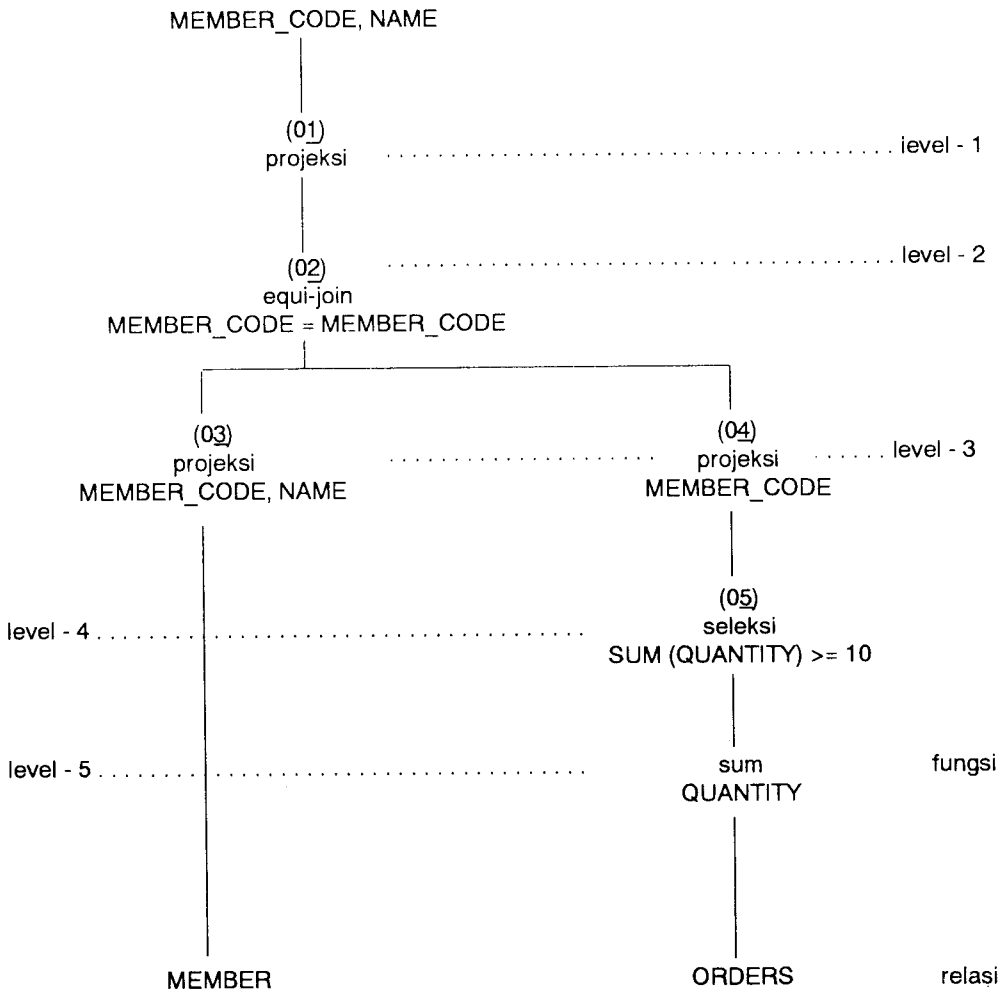
IV.1. OPERATOR BEBAS [SIT 85]

– **Operator bebas** (pengertian bebas adalah relatif terhadap operator relasi lainnya dalam satu transaksi) adalah operator relasi, dimana hasil (output) dan masukan (input) dari operator termaksud tidak menentukan eksekusi dari operator lainnya.

Pada Gambar-2, operator projeksi O3 disebut sebagai operator bebas, relatif terhadap O4 maupun O5.

IV.2. KELOMPOK OPERATOR BEBAS [BUT 87]

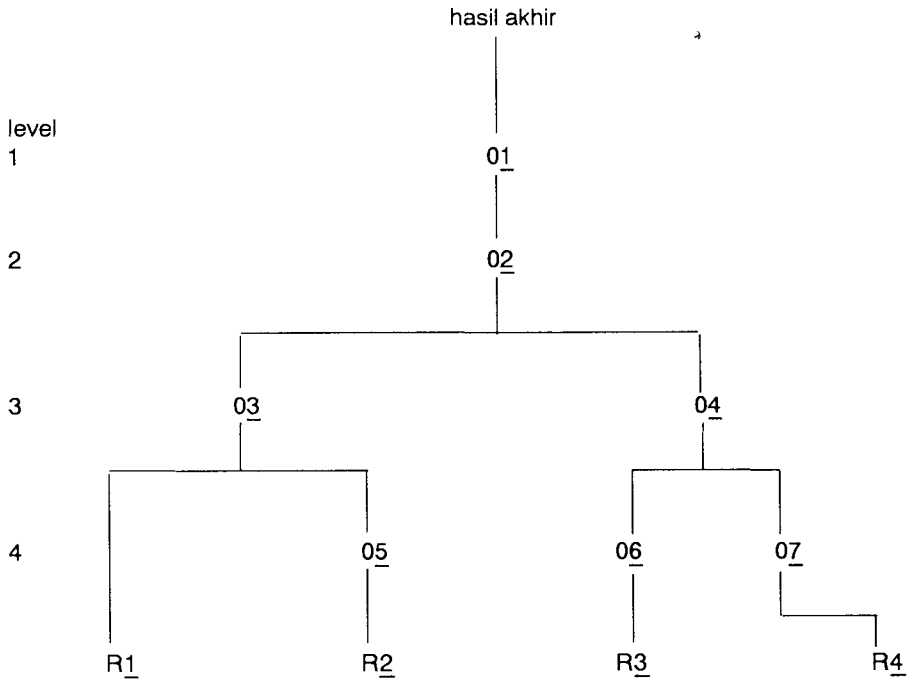
– **Kelompok operator bebas** adalah merupakan kumpulan dari beberapa operator relasi pada satu transaksi, dimana definisi **Operator bebas** dipenuhi oleh sesama operator relasi tersebut.



Gambar-2 Struktur Pohon Transaksi Contoh-2

Pada Gambar-2, maka kumpulan operator relasi O3 dan O4 adalah satu kelompok operator bebas. Sedangkan O3 dan O5 adalah kelompok operator bebas lain. Dengan demikian, pada Contoh-2, terdapat dua kelompok operator bebas. Untuk lebih mempermudah pemahaman, akan digunakan contoh transaksi teoritis, dengan struktur pohon seperti pada Gambar-3.

Pada Gambar-3, terdapat beberapa kelompok operator bebas (untuk mempermudah representasi, maka kelompok operator bebas dinyatakan dengan "panah dua arah"), yaitu :



Gambar-3 Struktur Pohon Transaksi

1. 03 dan 04 (03 ⟷ 04)
2. 04 dan 05 (04 ⟷ 05)
3. 03, 06 dan 07
   atau : 3.1. 03 dan 06 (03 ⟷ 06),
   3.2. 03 dan 07 (03 ⟷ 07),
   3.3. 06 dan 07 (06 ⟷ 07),

4. 04 dan 05 (04 ⟷ 05)
5. 05, 06 dan 07
   atau : 5.1. 05 dan 06 (05 ⟷ 06)
   5.2. 05 dan 07 (05 ⟷ 07)
   5.3. 06 dan 07 (06 ⟷ 07)

Jika dilakukan optimasi, maka akan didapatkan kelompok operator bebas berikut :

1. 03 ⟷ 04
2. 04 ⟷ 05
3. 03 ⟷ 06
4. 03 ⟷ 07
5. 06 ⟷ 07
6. 05 ⟷ 06
7. 05 ⟷ 07

IV.3. DERAJAT KEBEBASAN [BUT 87]

- **Derajat kebebasan** dari operator bebas adalah tingkat prioritas eksekusi operator bebas, relatif terhadap semua operator relasi pada transaksi.
- **Derajat kebebasan suatu operator bebas** ditentukan berdasarkan tingkatan (level) operator bebas tersebut pada struktur pohon operator relasi.

Pada Gambar-3, maka :

O3, O4 : mempunyai derajat kebebasan 3

O5, O6, O7 : mempunyai derajat kebebasan 4

V. EKSEKUSI OPERATOR BEBAS & KELOMPOK OPERATOR BEBAS

V.1. EKSEKUSI SECARA BERURUTAN (SEKUENSIAL)

Prinsip eksekusi kumpulan operator relasi yang merupakan penjabaran dari transaksi pada SMBDR dilakukan secara **berurutan** (sequential), meskipun SMBDR termaksud dioperasikan pada sistem operasi yang mendukung proses paralel (seperti multi-task). Dengan demikian, waktu eksekusi (waktu tanggap) untuk satu transaksi tertentu, adalah merupakan akkumulasi waktu tanggap eksekusi setiap operator relasi yang dimaksudkan sebagai penjabaran transaksi. Untuk Gambar-3, secara teoritis waktu tanggap transaksi (ts) adalah :

$$ts = \sum_{i=1}^{7} to_i$$

dimana : to_i : waktu eksekusi untuk operator relasi Oi

V.2. EKSEKUSI SECARA PARALEL

Dengan menerapkan definisi operator bebas dan kelompok operator bebas, maka eksekusi operator relasi secara paralel akan dapat diterapkan, dengan prinsip sebagai berikut:

- Seluruh operator relasi yang didefinisikan dalam satu kelompok operator bebas (misalnya : O3 dengan O4 dari Gambar-3) dapat dieksekusi secara bersama (paralel) dan tidak saling bergantungan satu sama lainnya.
- Prioritas pemilihan operator bebas yang akan dieksekusi, ditentukan berdasarkan derajat kebebasan dari setiap operator bebas. Operator bebas dengan derajat kebebasan paling besar akan mendapat prioritas pertama (hal ini adalah merupakan penjabaran prinsip urutan (sequence) dari struktur pohon.
- Eksekusi operator relasi dimulai dari level (tingkatan) yang paling tinggi (untuk Gambar-3, maka dimulai dari level 4, level 3 dan seterusnya sampai dengan level 1).
- Setiap operator relasi dianggap sebagai satu proses. Pada eksekusi sekuensial, seluruh operator relasi dari transaksi dianggap sebagai satu proses.

Berdasarkan prinsip ini, maka waktu eksekusi transaksi untuk Gambar-3 secara paralel (tp) secara teoritis akan lebih kecil dari ts, atau tp < ts.

Secara teoritis, besarnya tp akan ditentukan oleh beberapa hal berikut :

1. Jumlah proses yang dapat aktif secara bersama (paralel). Hal ini ditentukan oleh kapasitas sistem komputer yang digunakan.
2. Jumlah operator bebas pada setiap saat.
3. Waktu eksekusi terbesar dari setiap operator bebas yang terdapat pada satu kelompok operator bebas.
4. Waktu tanggap adalah sama dengan waktu untuk eksekusi seluruh operator relasi yang berada pada jalur terpanjang (jalur pada pohon transaksi adalah mulai dari akar pohon transaksi, sampai dengan daun pohon transaksi).

Secara praktis, hal tersebut diatas masih perlu dikoreksi, dengan beberapa beban (overhead) berikut :

1. Waktu yang dibutuhkan untuk pengendalian (sinkronisasi) proses yang paralel (manajemen untuk proses yang paralel).
2. Karena transaksi pada SMBDR biasanya melibatkan data dengan volume yang relatif besar, maka ada kecenderungan bahwa pemrosesan secara paralel akan menuntut penggunaan memori sekunder (untuk penyimpanan sementara) relatif lebih besar daripada eksekusi secara berurutan. Hal ini akan mengakibatkan naiknya beban (overhead) dari aktifitas Masukan / Keluaran.

V.3. ALGORITMA PENGENDALIAN EKSEKUSI PARALEL

Algoritma pengendalian eksekusi secara paralel dimaksudkan untuk mengendalikan pengaktifan proses untuk setiap operator relasi, agar terdapat sinkronisasi, sesuai dengan urutannya pada pohon transaksi.

Beberapa strategi yang mungkin diterapkan sebagai prinsip algoritma pengendali eksekusi paralel telah dibahas dan dianalisa secara rinci, yaitu penggunaan **satu variabel bersama** dan **dua variabel bersama** dengan beberapa alternatif algoritmanya [BUT 87]. Berikut ini dijelaskan salah satunya, untuk memberikan gambaran tentang pemecahan yang mungkin diterapkan.

Untuk penyimpanan data pengendalian proses paralel, digunakan satu tabel (dengan 4 arrays), yaitu :

- KONDISI : jumlah operator (proses) yang ditunggu, untuk menyatakan kondisi pengaktifan operator tersebut.
  0 : tidak ada proses yang ditunggu
  n : terdapat n proses yang harus ditunggu
  * : proses sudah aktif
- PROSES : Proses yang dimaksudkan
- DERAJAT : Derajat kebebasan proses, relatif terhadap struktur pohon yang dimaksudkan. Hal ini dapat diinterpretasikan sebagai tingkat (level) operator relasi yang dimaksudkan pada pohon transaksi.
- POINTER : Pointer yang menuju ke proses yang menunggu.

Untuk Gambar-3, tabel yang sesuai adalah sebagai berikut :

No.	K	P	D	PN
1	0	5	4	4
2	0	6	4	5
3	0	7	4	5
4	1	3	3	6
5	2	4	3	6
6	2	2	2	7
7	1	1	1	–

K : kondisi
p : proses (operator)
D : derajat kebebasan
PN : pointer

1. Algoritma pengaktifan proses (AKTIF) :

   PROSEDUR AKTIF (aktif secara permanen)

   BILA processor tersedia MAKA
   Cari proses Px dengan K = 0 dan D paling besar
   Untuk Px, maka
   K = * (identifikasi bahwa proses sudah aktif)
   Aktifkan proses Px

   SELESAI

   AKHIR PROSEDUR AKTIF;

2. Algoritma modifikasi tabel (MOD_TABEL) :

   PROSEDUR MOD_TABEL (Untuk optimasi processor, maka prosedur tersebut hanya aktif, jika ada proses yang sudah selesai).

   BILA ada proses Px sudah selesai MAKA
   m = PN dari Px
   PN dari Px = 0
   K dari baris ke-m dikurangi satu
   (Km = Km – 1)

   SELESAI

   AKHIR PROSEDUR MOD_TABEL;

VI. HASIL IMPLEMENTASI PADA SISTEM OPERASI MULTI-PROGRAMMING

Implementasi dari prinsip yang telah dijelaskan sebelumnya (eksekusi operator relasi secara paralel) telah dilakukan, dan telah diuji-coba dengan berbagai keterbatasan yang ada.

Uji-coba dilakukan dengan dua cara, yaitu eksekusi secara sekuensial dan eksekusi secara paralel, dengan menggunakan data dan metoda eksekusi masing-masing operator yang sama untuk kedua cara eksekusi. Dengan demikian, secara garis besar bahwa perbedaan waktu tanggap untuk transaksi yang sama, hanya akan timbul sebagai akibat dari :

1. Beban (overhead) pengendalian eksekusi paralel (hanya untuk eksekusi secara paralel)
2. Beban (overhead) aktifitas Masukan / Keluaran yang timbul sebagai akibat keterbatasan memori utama.
3. Jumlah kanal Masukan / Keluaran yang tersedia untuk eksekusi secara paralel.

VI.1. FASILITAS KOMPUTER

Sistem komputer yang digunakan adalah :

- Processor : PDP 11/44
- Sistem Operasi : XENIX
- Pemrograman : C-Language
- Kanal I/O : 2 unit
- Disk : 2 unit

VI.2. BASIS DATA & TRANSAKSI

Basis Data yang digunakan sebagai contoh dalam pelaksanaan uji-coba adalah basis data PERSONALIA, yang terdiri dari 10 relasi. Struktur dan contoh data disertakan pada Lampiran-1.

Transaksi untuk basis data termaksud didefinisikan secara khusus (dengan kode). Hal ini dilakukan untuk menghindari semaksimal mungkin adanya beban (overhead) kompilasi transaksi. Pada lampiran-2, disertakan 6 transaksi yang digunakan pada uji-coba.

VI.3. HASIL UJI-COBA

Dalam pelaksanaan uji-coba, kondisi yang diterapkan adalah sebagai berikut :

1. Pada saat percobaan, sistem komputer tidak digunakan untuk proses yang lain.
2. Jumlah tupple yang digunakan dalam uji-coba adalah 1000 dan 10.000. Hal ini dipenuhi dengan data dummy.
3. Pengamatan dilakukan terhadap waktu : waktu mulai eksekusi serta saat hasil akhir eksekusi didapatkan. Dengan demikian, dalam waktu yang diamati telah tercakup seluruh beban (overhead) nyata yang digunakan dalam rangka eksekusi transaksi.

Untuk kondisi tersebut, didapatkan hasil sesuai dengan isi tabel- 1 (untuk 1000 tupple) dan tabel-2 (untuk 10.000 tupple). Untuk informasi lebih rinci, lihat [BUT 87].

Tabel-1 : Waktu tanggap (detik) untuk 1000 tupple :

TRANSAKSI	EKSEKUSI SEKUENSIAL	EKSEKUSI PARALEL			
		1 disk		2 disk	
		waktu	reduksi	waktu	reduksi
T1	9,20	14,80	–60,87 %		
T2	28,00	36,33	–29,75 %		
T3	15,33	17,38	–13,37 %	15,60	–1,76 %

Tabel-2 : Waktu tanggap (detik) untuk 10.000 tupple :

TRANSAKSI	EKSEKUSI SEKUENSIAL	EKSEKUSI PARALEL			
		1 disk		2 disk	
		waktu	reduksi	waktu	reduksi
T1	70,33	68,80	+0,75 %		
T2	306,00	328,00	–7,19 %	355,00	–16,01 %
T3	192,50	181,50	+5,71 %	177,55	+ 7,79 %
T4	254,50	241,50	+5,11 %	234,00	+ 8,06 %
T5	1967,50	1818,0	+7,60 %	1744,00	+11,36 %
T6	538,00			529,00	+ 1,67 %

VI.4. EVALUASI HASIL UJI-COBA

Dari hasil uji-coba (disertakan pada Tabel-1 dan Tabel-2) terlihat beberapa hal yang penting, dan beberapa keanehan :

1. Waktu tanggap eksekusi paralel dengan 1000 tupple ternyata lebih besar dari waktu tanggap eksekusi sekuensial. Sebaliknya, untuk 10.000 tupple, maka keadaan akan terbalik. Dari uji-coba ini dapat dinyatakan bahwa :
   - Pada prinsipnya, perbaikan waktu tanggap dapat terjadi dengan melakukan eksekusi paralel, meskipun secara nyata hal ini tidak terlihat pada uji-coba dengan 1000 tupple.
   - beban (overhead) pengendalian eksekusi paralel adalah **lebih besar dari perbaikan** waktu tanggap untuk 1000 tupple.
2. Reduksi waktu tanggap ternyata lebih baik jika menggunakan dua disk dan dua kanal. Dengan demikian dapat dinyatakan bahwa, jika jumlah kanal dan disk membesar, maka waktu tanggap akan makin membaik (tentunya ini berlaku, jika transaksi yang dimaksudkan menggunakan banyak relasi).

3. Waktu eksekusi T4 secara paralel ternyata lebih baik dari eksekusi sekuensial. Hal ini tidak mungkin, karena pada T4 tidak terdapat operator bebas. Seharusnya, waktu eksekusi secara paralel akan lebih besar dari waktu eksekusi secara sekuensial, karena adanya beban (overhead) pengendalian eksekusi paralel.
4. Sesuai dengan transaksi dan lingkungan uji-coba, maka reduksi waktu tanggap maksimum yang dapat dicapai adalah 11,36 %. Hal ini dapat diamati dari eksekusi paralel untuk transaksi T5 (menggunakan 2 disk). Dengan reduksi hanya 11,36 %, berarti beban (overhead) untuk pengendalian eksekusi paralel relatif cukup dominan di dalam waktu tanggap.
5. Mengingat keterbatasan fasilitas yang dapat digunakan, khususnya yang berkaitan dengan sistem operasi, beberapa hasil uji-coba tidak dapat dijelaskan (misalnya hasil uji-coba dari transaksi T2). Kemungkinan, hal ini terjadi sebagai akibat dari tidak idealnya kondisi percobaan. Jika akan dianalisa, seharusnya hal ini akan dapat dijawab dengan mempertimbangkan beberapa aspek dari sisi sistem operasi, yaitu :
   - strategi pengelolaan sumber (resource) oleh sistem operasi
   - operasi masukan/keluaran (I/O)
   - memori primer dan strategi pengalokasian untuk setiap proses pada eksekusi paralel.
6. Reduksi waktu tanggap akan membaik, jika struktur pohon dari transaksi memiliki banyak cabang dan seimbang. Hal ini didukung oleh makin banyaknya jumlah operator bebas.

VII. KESIMPULAN

Masalah yang dibahas dalam tulisan ini berangkat dari suatu hipotesa. Dari analisa teoritis terhadap ide yang dimaksudkan pada hipotesa, telah dijelaskan bahwa hipotesa tersebut benar, dengan dikembangkannya definisi **operator bebas**, **kelompok operator bebas** dan **derajat kebebasan**.

Berbagai aspek yang mengakibatkan meningkatnya kompleksitas persoalan dalam rangka implementasi (uji-coba) juga telah dibahas, walaupun belum mencakup seluruh aspek, khususnya dari sisi sistem operasi (pengelolaan eksekusi paralel). Sebagai dampak langsung dari kompleksitas tersebut, adanya beberapa hasil pengamatan yang belum dapat dijelaskan secara tuntas.

Dengan keterbatasan yang disebutkan diatas, hasil uji-coba setidak-tidaknya secara global sudah dapat digunakan sebagai dasar untuk menyatakan kebenaran hipotesa yang disebutkan di awal tulisan ini, yaitu :

- Operator-operator dari transaksi pada basis data model relasi dapat dieksekusi secara paralel, dengan menganggap bahwa setiap operator relasi adalah satu proses.
- Eksekusi operator relasi dari transaksi secara paralel akan memperbaiki waktu tanggap.

Berbagai hal masih dianggap menjadi penghalang dari keberhasilan uji-coba (minimal, belum memberikan reduksi waktu tanggap yang relatif baik, khususnya pada eksekusi untuk jumlah tupple yang terbatas). Hal ini terjadi sebagai akibat keterbatasan dari berbagai sisi.

Dua hal penting di bawah ini yang sangat berperan dalam meningkatkan reduksi waktu tanggap, yaitu :

1. Meningkatkan derajat paralelisme pada processor. Hal ini mungkin dilakukan dengan pengembangan metoda optimasi transaksi, sehingga struktur pohon operator relasi dapat diusahakan agar semaksimal mungkin bercabang banyak dan seimbang. Struktur pohon operator relasi seperti ini akan berarti memperpendek jalur operator relasi dan sekaligus meningkatkan jumlah operator bebas.
2. Meningkatkan derajat paralelisme pada aktifitas (operasi) Masukan / Keluaran.

Reduksi waktu tanggap maksimum (teoritis) yang dapat dicapai dengan peningkatan pada dua hal tersebut diatas, adalah mendekati prosentase beban Masukan / Keluaran pada SMBDR pada umumnya, yang berkisar antara 60 - 80 % [DON 85] .

Untuk tujuan peningkatan reduksi waktu tanggap yang dimaksudkan diatas, beberapa hal yang masih perlu dikembangkan lebih lanjut, antara lain :

1. Optimasi algoritma yang digunakan, yang berkaitan dengan eksekusi setiap operator relasi, khususnya yang menyangkut **pengendalian eksekusi secara paralel.**
2. Pengembangan metoda optimasi transaksi, sehingga struktur pohon operator relasi dapat diusahakan agar semaksimal mungkin bercabang banyak dan seimbang.
3. Melaksanakan uji coba pada sistem komputer yang memberikan keleluasaan lebih besar dalam hal kemungkinan konfigurasi perangkat keras dan sistem operasi yang dapat digunakan.

UCAPAN TERIMA KASIH

Keberhasilan uji-coba eksekusi paralel operator relasi yang dijelaskan pada tulisan ini tidak lepas dari ketersediaan komputer yang dimiliki oleh PT.INTI, dan peran Sdr.Ir.Manonton Butarbutar dalam pengembangan berbagai program yang digunakan untuk uji-coba. Untuk itu, penulis mengucapkan terima kasih atas dukungannya.

DAFTAR PUSTAKA

[BUT 87] Butarbutar, M., EKSEKUSI PARALEL OPERATOR RELASI SMBDR, Tugas akhir program S-1, Jurusan Teknik Informatika ITB, Bandung, Pebruari 1987.

[DON 85] Donovan, J.J. & Stuart, E.M., OPERATING SYSTEM, Mc.Graw-Hill, Tokyo, 1985.

[SIT 85] Sitohang,B., EKSEKUSI OPERATOR RELASI RDBMS PADA SISTEM OPERASI MULTI-PROGRAMMING, KKN - IPKIN, Jakarta, September 1985.

[STA 88] Stanley, Y.W. Su, DATABASE COMPUTERS : Principles, Architectures & Techniques, Mc.Graw-Hill, 1988.

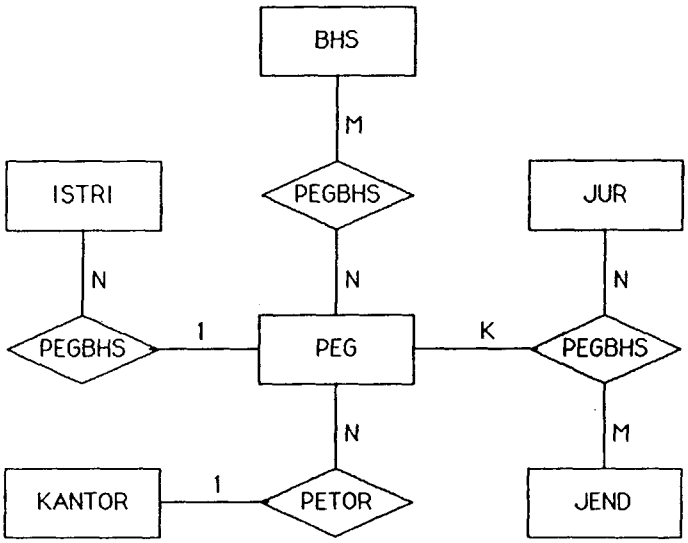
[ULL 82] Ullman, J.D., PRINCIPLES OF DATABASE SYSTEMS, Computer Science Press, Maryland, 1982.

[ULL 88] Ullman, J.D., DATABASE AND KNOWLEDGE - BASE SYSTEM, Volume I, Computer Science Press, Maryland, 1988.

Lampiran-1

STRUKTUR BASIS DATA

1. Model E-R



2. Skema logik :

PEG (NIP , NAMA, UMUR)
JEN(KJEN, NJEN)
JUR(KJUR, NJUR)
BHS(KBHS, NBHS)
ISTRI (NIT , PEK)
KANTOR(KTOR, NTOR)
PEND(NIP , KJEN, KJUR)
PEGBHS(NIP, KBHS, KET)
PETRI(NIP, NIT)
PETOR(NIP, KTOR, TGL)

Lampiran-1

3. Contoh Data :

PEG

NIP	NAMA	UMUR
8701	Ali	40
8702	Budi	30
8703	Charles	27
8704	Daniel	25
8705	Efendi	29

JEN

KJEN	NJEN
KR	Kursus
S0	Diploma
S1	Sarjana
S2	Master
S3	Doktor

JUR

KJUR	NJUR
IF	Informatika
EL	Elektro
MA	Matematika
BI	Biologi
TA	Tambang

BHS

KBHS	NBHS
IG	Inggris
PR	Perancis
JR	Jerman
JP	Jepang
BL	Belanda

ISTRI

NIT	PEK
Ani	PT. Ganesha
Tuti	PT. Ganesha
Betty	Ikut Suami
Ati	PT. Ganesha
Susi	PT. Dago
Nelly	Ikut Suami
Teti	PT. Dago

KANTOR

KTOR	NTOR
JK	Jakarta
BD	Bandung
MD	Medan
SB	Surabaya
SM	Semarang

PEND

NIP	KJEN	KJUR
8701	S1	BI
8701	S2	MA
8701	S3	IF
8702	S1	EL
8702	S2	IF
8703	S1	TA
8703	S2	IF
8704	S1	IF
8705	S1	IF

PEGBHS

NIP	KBHS	KET
8701	IG	A
8701	PR	A
8701	JR	P
8702	IG	P
8702	PR	A
8703	IG	P
8703	PR	A
8704	IG	A
8705	IG	P

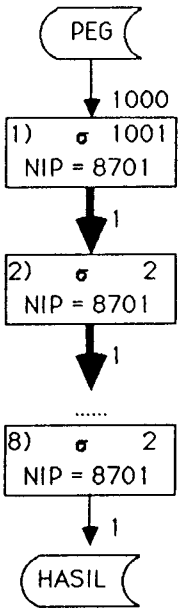
PETRI

NIP	NIT
8701	Teti
8702	Ani
8702	Tuti
8702	Betty
8703	Susi
8704	Ati
8705	Nelly

LAMPIRAN-2

TRANSAKSI UJI-COBA

1. **Transaksi T1**
   Makna : Dapatkan data pribadi pegawai dengan NIP adalah 8701.
2. **Transaksi T2**
   Makna : Dapatkan NIP dan NAMA pegawai yang memenuhi syarat berikut :
   kode jurusan adalah IF, kode jenjang S1, dan umur lebih kecil dari 30.
3. **Transaksi T3**
   Makna : Dapatkan NIP pegawai yang memenuhi syarat berikut :
   nama jurusan adalah Informatika, dengan jenjang S1, dan pernah bekerja di Bandung.
4. **Transaksi T4**
   Makna : Dapatkan data pribadi, pendidikan, penguasaan bahasa, istri, kantor dan jenjnag pendidikan dari pegawai dengan NIP adalah 8704.
5. **Transaksi T5**
   Makna : Siapa saja pegawai (NIP) yang tercatat pada relasi PEG, PEND, PEGBHS, PETRI, dan PETOR. Daftarkan kode kantor, bahasa dan jenjang pendidikannya.
6. **Transaksi T6**
   Makna : Dapatkan data pribadi dan pendidikan dari pegawai bernama ALI.



pipa = 7
file = 2
deskriptor file = 16

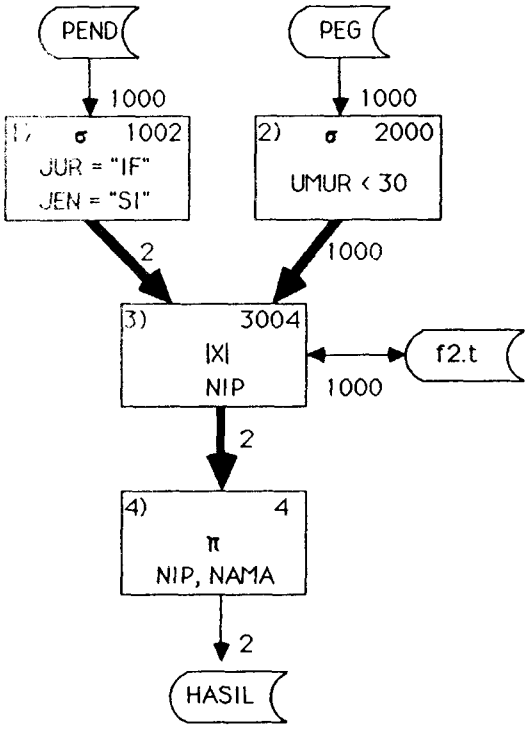
Keterangan :

 masukan/keluaran dari/ke file

masukan/keluaran dari/ke pipa

*Angka di samping  menyatakan jumlah record file
*Angka di samping menyatakan jumlah record melalui pipa
*Angka di kiri atas kotak menyatakan nomor operator
*Angka dikanan atas kotak menyatakan jumlah operasi M/K.

Struktur Pohon Transaksi T1

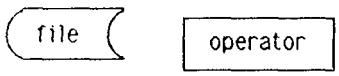


pipa = 3
file = 4
deskriptor file = 10

Keterangan :

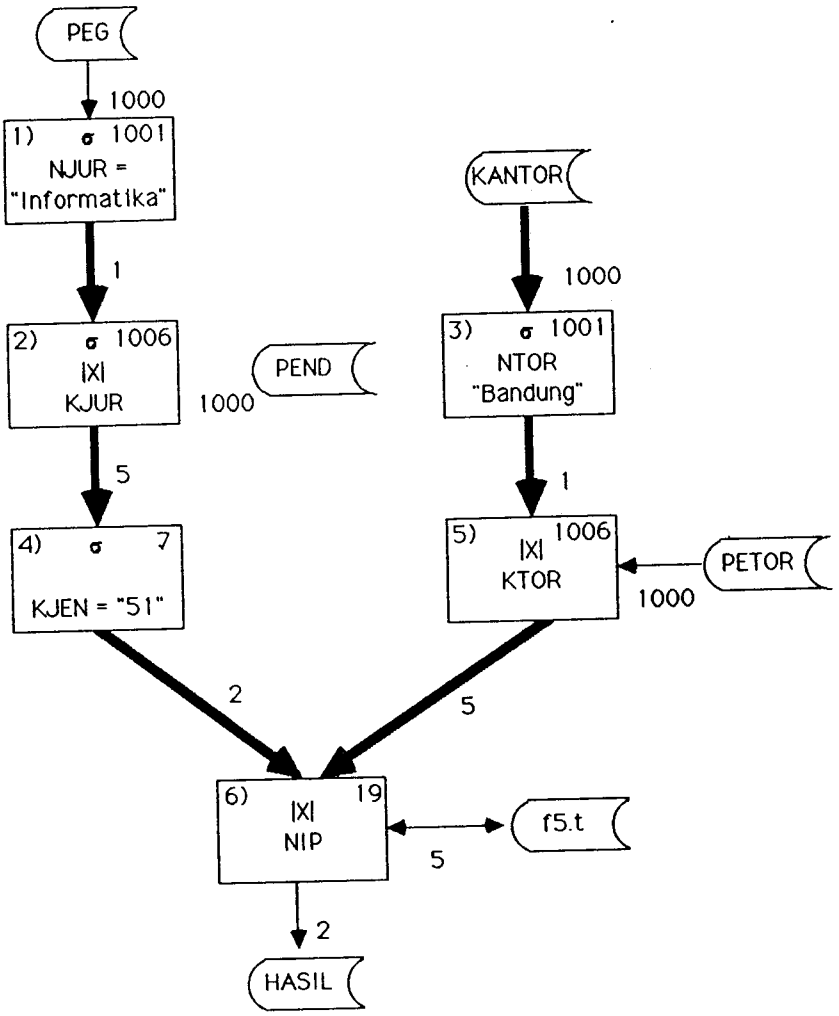
masukan/keluaran dari/ke file

masukan/keluaran dari/ke pipa



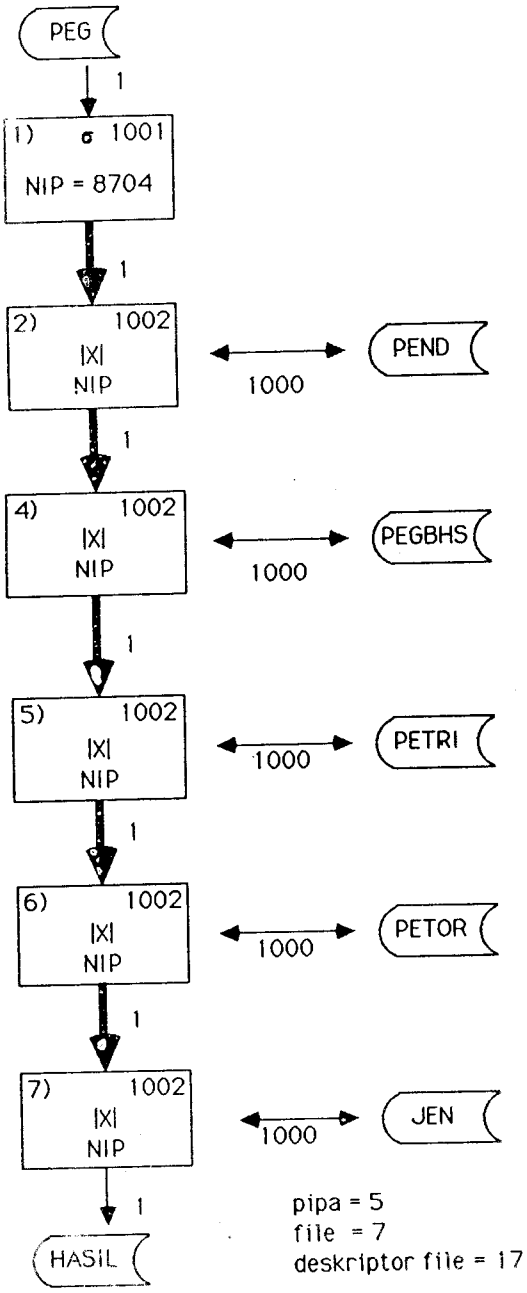
*Angka di samping ⟶ menyatakan jumlah record file
*Angka di samping ➡ menyatakan jumlah record melalui pipa
*Angka di kiri atas kotak menyatakan nomor operator
*Angka dikanan atas kotak menyatakan jumlah operasi M/K.

Struktur Pohon Transaksi T2

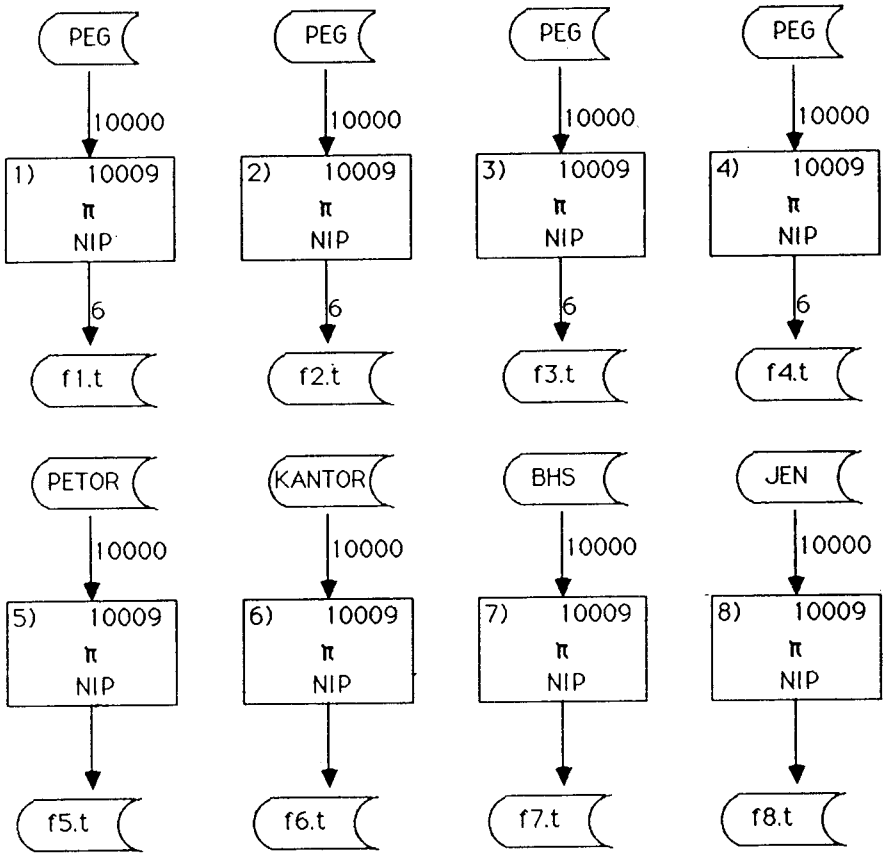


pipa = 5
file = 6
deskriptor file = 16

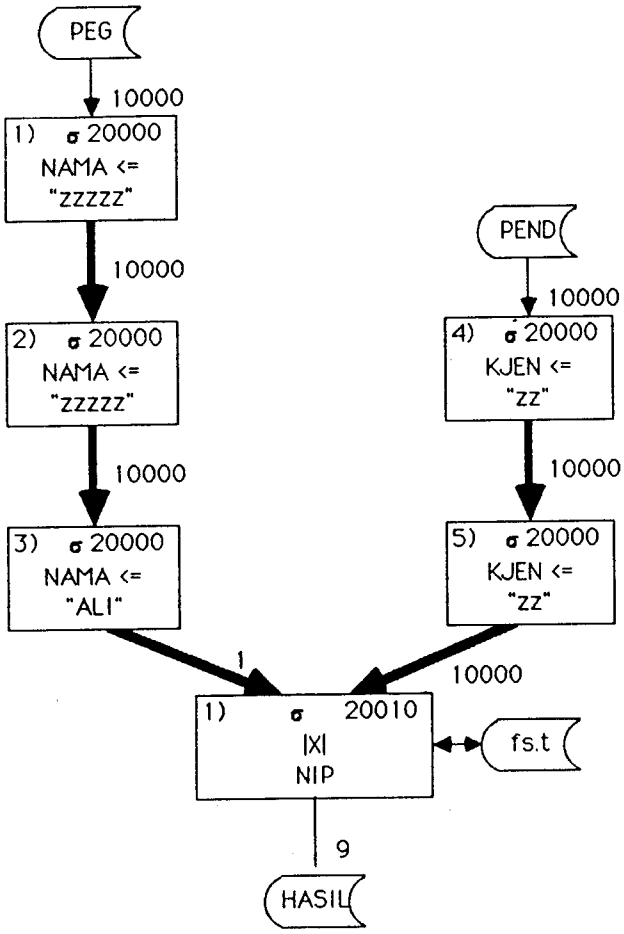
Struktur Pohon Transaksi T3



Struktur Pohon Transaksi T4



Struktur Pohon Transaksi T5



Struktur Pohon Transaksi T6