

Tugas 2 Basis Data Lanjut

1. [25] Dari schedule berikut ini yang melibatkan 3 transaksi pada data item A, B, dan C:

S1: r1(A); r2(C); r1(C); r3(A); r3(B); w1(A); c1; w3(B); c3; r2(B); w2(C); w2(B); c2;

S2: r1(A); r2(C); r1(C); r3(A); r3(B); w1(A); w3(B); r2(B); w2(C); w2(B); c1; c2; c3;

S3: r1(A); r2(C); r3(A); r1(C); r2(B); r3(B); w1(A); c1; w2(C); w3(B); w2(B); c3; c2;

S1)

No	T1	T2	T3	B)
1	R1(A)			I. Ya, karena tidak ada transaksi yang mengubah/write apapun yang T1 baca. Jadi, T2 dan T3 tidak perlu di rollback
2		R2(C)		
3	R1(C)			
4			R3(A)	II. Ya, karena T3 yang W(B) sudah di commit duluan. Jadi, T1 dan T3 tidak perlu di rollback
5			R3(B)	
6	W1(A)			III. Ya, karena T3 tidak membaca apapun yang diupdate atau ditulis oleh T1 atau T2. Jadi, T1 dan T2 tidak perlu di rollback
7	Com1			
8			W3(B)	
9			Com3	C) Ya cascadeless, karena setiap transaksi yang membaca nilai yang diupdate oleh transaksi lain, transaksi pengupdate itu di commit sebelum nilai itu dibaca. Dalam schedule ini, R2(B) (no. 10) membaca nilai B yang diupdate oleh T3, tapi T3 sudah di commit duluan.
10		R2(B)		
11		W2(C)		
12		W2(B)		S1 juga strict karena W2(B) dilakukan setelah T3 commit
13		Com2		

S2)

No	T1	T2	T3	B)
1	R1(A)			I. Ya, karena tidak ada yang baca A setelah W1(A) (no. 6) II. Ya, karena tidak ada yang baca C setelah no. 9 dan B setelah no.10 III. Tidak, karena T2 baca B (no. 8) setelah ditulis oleh T3 (no. 7) dan T2 sudah commit duluan (no. 12)
2		R2(C)		
3	R1(C)			
4			R3(A)	C)

Nama: Zidan Kharisma Adidarma

NPM: 2006463881

Tugas 2 Basis Data Lanjut

5			R3(B)	Tidak cascadeless, karena transaksi 2 membaca B yang ditulis dari T3 yang belum di commit. Bukan strict juga karena bukan cascadeless
6	W1(A)			
7			W3(B)	
8		R2(B)		
9		W2(C)		
10		W2(B)		
11	Com1			
12		Com2		
13			Com3	

S3)

No	T1	T2	T3	B) I. Ya, karena tidak ada yang baca A setelah W1(A) (no. 7) II. Ya, karena tidak ada yang baca C setelah no. 9 dan tidak ada yg baca B setelah no. 11 III. Ya, karena tidak ada yang baca B setelah no. 10 C) Ya cascadeless, karena setiap transaksi yang membaca nilai yang diupdate oleh transaksi lain, transaksi pengupdate itu di commit sebelum nilai itu dibaca. Dalam schedule ini, tidak ada yang membaca nilai yang sudah diupdate. Akan tetapi, ini tidak strict karena T2 mengupdate nilai B (no. 11) padahal T3 yang sebelumnya mengupdate B (no. 10) belum commit
1	R1(A)			
2		R2(C)		
3			R3(A)	
4	R1(C)			
5		R2(B)		
6			R3(B)	
7	W1(A)			
8	Com1			
9		W2(C)		
10			W3(B)	
11		W2(B)		
12			Com3	
13		Com2		

Nama: Zidan Kharisma Adidarma

NPM: 2006463881

Tugas 2 Basis Data Lanjut

2. [15] Dari schedule berikut ini, tentukan apakah serializable dan berikan alasan singkat! Jika schedule serializable, serial schedule yang ekuivalen dengan schedule yang bersangkutan!

a. $r_1(A)$; $r_3(A)$; $w_1(A)$; $r_2(A)$; $w_3(A)$;

Precedence graph:

from\to	T1	T2	T3	Schedule tidak serializable karena ada siklus dalam graph, sebagai contoh: $T1 \rightarrow T2 \rightarrow T3 \rightarrow T1$
T1		A	A	
T2			A	
T3	A			

b. $r_1(A)$; $r_3(A)$; $w_3(A)$; $w_1(A)$; $r_2(A)$;

Precedence graph:

from\to	T1	T2	T3	Schedule tidak serializable karena ada siklus dalam graph, sebagai contoh: $T1 \rightarrow T3 \rightarrow T1$
T1		A	A	
T2				
T3	A	A		

c. $r_3(A)$; $r_2(A)$; $w_3(A)$; $r_1(A)$; $w_1(A)$;

Precedence graph:

from\to	T1	T2	T3	Schedule serializable karena tidak ada siklus dalam graph. Schedule yang ekuivalen: $r_2(A)$; $r_3(A)$; $w_3(A)$; $r_1(A)$; $w_1(A)$;
T1				
T2	A		A	
T3	A			

d. $r_3(A)$; $r_2(A)$; $r_1(A)$; $w_3(A)$; $w_1(A)$;

Precedence graph:

from\to	T1	T2	T3	Schedule tidak serializable karena ada siklus dalam graph, sebagai contoh: $T1 \rightarrow T3 \rightarrow T1$
T1			A	
T2	A		A	
T3	A			

Tugas 2 Basis Data Lanjut

3. [10] Di bawah ini merupakan penjadwalan yang menggunakan binary locking. Tentukan apakah penjadwalan di bawah ini well-formed transaction, legal scheduler, dan/atau memenuhi two phase locking.

1) S1: L1(A), R1(A), L2(C), W2(C), U1(A), L3(B), R3(B), U3(B), R1(A), U2(C)

Tidak well formed, karena R1(A) dilakukan padahal T1 sudah melepas lock A dan T1 belum melock A lagi. Memenuhi legal scheduler. Tidak 2PL karena tidak well-formed

2) S2: L2(A), W2(A), R2(A), L1(B), R1(B), W2(A), U1(B), U2(A)

Well-formed, karena R/W dilakukan setelah ada lock. Legal scheduler, karena 1 lock dipegang at most 1 transaksi at any given point. Dan 2PL, karena di T1 tidak di unlock sampai operasi R/W selesai, dan di T2 tidak di unlock sampai operasi R/W selesai (dgn kata lain, lock tidak dibutuhkan lagi sama transaksi)

4. [20] Diberikan schedule seperti di bawah ini.

S1: r1(x); r2(z); r1(x); r3(x); r3(y); w1(x); w3(y); r2(y); w2(z); w2(y)

S2: r1(x); r2(z); r3(x); r1(z); r2(y); r3(y); w1(x); w2(z); w3(y); w2(y)

Untuk masing-masing schedule:

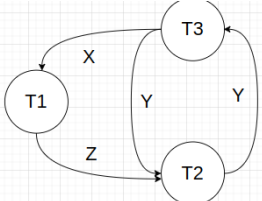
- [5] Gambarkan precedence graph schedule tersebut.
- [10] Dari precedence graph tersebut tentukan apakah schedule serializable atau tidak serta berikan alasan singkat.
- [5] Susunlah S1 sehingga mengikuti protokol 2PL

S1)

from\to	T1	T2	T3	(T2) <-- Y -- (T3) -- X --> (T1)
T1				Berdasarkan graf, schedule S1 bisa diserializable karena tidak ada siklus
T2				Schedule S1 yang mengikuti protokol 2PL: r3(X); r3(Y); w3(Y); r1(X); r1(X); w1(X); r2(Z); r2(Y); w2(Z); w2(Y); atau T3 -> T1 -> T2
T3	X	Y		(bila lockingnya diperlihatkan): l3(X); l3(Y); r3(X); r3(Y); w3(Y); u3(X); u3(Y); l1(X); r1(X); r1(X); w1(X); u1(X); l2(Z); l2(Y); r2(Z); r2(Y); w2(Z); w2(Y); u2(Y); u2(Z)

Tugas 2 Basis Data Lanjut

S2)

from\to	T1	T2	T3	
T1		Z		 <p>Berdasarkan graf, schedule S2 tidak bisa diserializable karena ada siklus. Sebagai contoh: T3->T2->T3 T2->T3->T2 T3->T1->T2->T3 T2->T3->T1->T2</p>
T2			Y	
T3	X	Y		

5. [30] Terapkan timestamp ordering pada S1 dan S2 di nomor 4 dengan menggunakan format seperti berikut ini dan tentukan apakah schedule selesai dijalankan!

Jika schedule tidak selesai dijalankan, apa yang terjadi dengan tiap transaksi di dalamnya, serta jelaskan singkat!

Asumsi:

- data item X, Y, dan Z belum dibaca/tulis oleh transaksi lain.
- Timestamp tiap transaksi di-set dengan nilai urutan operasi pertamanya pada suatu schedule. Misalnya: S: R1(X), R1(Y), R2(Y), maka timestamp T1 adalah 1 (karena operasi pertamanya dimulai pada no. urut 1) dan timestamp T2 adalah 3 (karena operasi pertamanya dimulai pada no. urut 3).
- Jika ada transaksi yang abort, cascading rollback diterapkan.

S1) : r1(x); r2(z); r1(x); r3(x); r3(y); w1(x); w3(y); r2(y); w2(z); w2(y)

T1: 1	T2: 2	T3: 4	X	Y	Z
R1(X)			tr=1 tw=0	tr=0 tw=0	tr=0 tw=0
	R2(Z)		tr=1 tw=0	tr=0 tw=0	tr=2 tw=0
R1(X)			tr=1 tw=0	tr=0 tw=0	tr=2 tw=0
		R3(X)	tr=4 tw=0	tr=0 tw=0	tr=2 tw=0
		R3(Y)	tr=4 tw=0	tr=4 tw=0	tr=2 tw=0

Nama: Zidan Kharisma Adidarma

NPM: 2006463881

Tugas 2 Basis Data Lanjut

W1(X)ABORT			tr=4 tw=0	tr=4 tw=0	tr=2 tw=0
		W3(Y)	tr=4 tw=0	tr=4 tw=4	tr=2 tw=0
	R2(Y)ABORT		tr=4 tw=0	tr=4 tw=4	tr=2 tw=0
	W2(Z)		tr=4 tw=0	tr=4 tw=4	tr=2 tw=0
	W2(Y)		tr=4 tw=0	tr=4 tw=4	tr=2 tw=0
<p>Kesimpulan: T1 dan T2 abort. Sementara T3 selesai. Schedule tidak selesai dijalankan karena terdapat transaksi pada T2 yang tidak dijalankan akibat sudah abort</p>					

S2) r1(x); r2(z); r3(x); r1(z); r2(y); r3(y); w1(x); w2(z); w3(y); w2(y)

T1: 1	T2: 2	T3: 3	X	Y	Z
R1(X)			tr=1 tw=0	tr=0 tw=0	tr=0 tw=0
	R2(Z)		tr=1 tw=0	tr=0 tw=0	tr=2 tw=0
		R3(X)	tr=3 tw=0	tr=0 tw=0	tr=2 tw=0
R1(Z)			tr=3 tw=0	tr=0 tw=0	tr=2 tw=0
	R2(Y)		tr=3 tw=0	tr=2 tw=0	tr=2 tw=0
		R3(Y)	tr=3 tw=0	tr=3 tw=0	tr=2 tw=0
W1(X)ABORT			tr=3 tw=0	tr=3 tw=0	tr=2 tw=0
	W2(Z)		tr=3 tw=0	tr=3 tw=0	tr=2 tw=2
		W3(Y)	tr=3 tw=0	tr=3 tw=3	tr=2 tw=2
	W2(Y)ABORT		tr=3 tw=0	tr=3 tw=3	tr=2 tw=2
<p>Kesimpulan: Schedule selesai dijalankan, tapi T1 dan T2 abort</p>					