

Pertemuan 14

Lingkungan Basis Data

KONKURENSI

CONCURRENCY (KONKURENSI)

Ada 3 masalah yang disebabkan oleh Concurrency :

1. Masalah kehilangan modifikasi (*Lost Update Problem*)

Masalah ini timbul jika dua transaksi mengakses item database yang sama yang mengakibatkan nilai dari database tersebut menjadi tidak benar.

lanjutan

Transaksi A	Waktu	Transaksi B
=	↓	=
Baca R	t1	=
=	↓	=
=	t2	Baca R
=	↓	=
Modifikasi R	t3	=
=	↓	=
=	t4	Modifikasi R
=	↓	=

Contoh Lost Update problem

Data transaksi pada rekening bersama (Ika dan Susi)

Waktu	Transaksi Ika	Transaksi Susi	Saldo
T1	Read Saldo	1.000.000
T2	Read Saldo	1.000.000
T3	Saldo:=Saldo-50.000	1.000.000
T4	Write Saldo	950.000
T5	Saldo:= saldo+100.000	1.000.000
T6	Write Saldo	1.100.000

Nilai saldo menjadi tidak benar disebabkan transaksi Susi membaca nilai saldo sebelum transaksi Ika mengubah nilai tersebut dalam database, sehingga nilai yang sudah di update yang dihasilkan dari transaksi Ika menjadi hilang.

lanjutan

2. Masalah Modifikasi Sementara (*uncommitted Update Problem*)

Masalah ini timbul jika transaksi membaca suatu record yang sudah dimodifikasi oleh transaksi lain tetapi belum terselesaikan (*uncommitted*), terdapat kemungkinan kalau transaksi tersebut dibatalkan (*rollback*).

lanjutan

Transaksi A	Waktu	Transaksi B
-	↓	-
Baca R	t1	Modifikasi R
-	↓	-
-	t2	-
-	↓	-
Modifikasi R	t3	Rollback
-	↓	-

Contoh *uncommitted Update Problem*

Waktu	Transaksi Simpanan	Transaksi Bunga	Saldo
T1	Read Saldo	1.000.000
T2	Saldo:=saldo+1.000.0000	1.000.000
T3	Write Saldo	2.000.000
T4	Read Saldo	2.000.000
T5	Saldo:= saldo*0.15	2.000.000
T6	Write Saldo	2.300.000
T7	RollBack	2.300.000

Nilai saldo menjadi tidak benar disebabkan terjadi RollBack pada T7 yang membatalkan transaksi sebelumnya (T6), sehingga saldo seharusnya tetap 2.000.000

lanjutan

3. Masalah Analisa yang tidak konsisten (*Problem of inconsistency Analysis*)

Masalah ini timbul jika sebuah transaksi membaca suatu nilai tetapi transaksi yang kedua mengupdate beberapa nilai tersebut selama eksekusi transaksi pertama

Contoh Problem of Inconsistency Analysis

⊕ Nilai 1 = 40	Nilai 2 = 50	Nilai 3 = 30
Transaksi A	Waktu	Transaksi B
-	↓	-
Baca nilai 1(40)	t1	-
Jum=40	↓	-
-	↓	-
Baca nilai 2(50)	t2	-
Juml=90	↓	-
-	↓	-
-	↓	baca nilai 3(30)
-	↓	-
-	↓	modifikasi nilai 3
-	↓	30 → 20
-	↓	-
-	↓	baca nilai 1(40)
-	↓	-
-	↓	modifikasi nilai 1
-	↓	40 → 50
-	↓	-
-	↓	commit
Baca nilai 3(20)	t8	-
Juml=110(bukan 120)	↓	-
-	↓	-

Transaksi A menjumlahkan nilai 1, nilai 2 dan nilai 3
 Transaksi B → nilai 1 + 10, nilai 3 -10

Locking

LOCKING adalah salah satu mekanisme pengontrol concurrency

KONSEP DASAR :

Ketika sebuah transaksi memerlukan jaminan kalau record yang diinginkan tidak akan berubah secara mendadak, maka diperlukan kunci untuk record tersebut

FUNGSI

Locking berfungsi untuk menjaga record tersebut agar tidak dimodifikasi oleh transaksi lain.

lanjutan

Jenis- Jenis Lock :

1. Share (S)

Kunci ini memungkinkan pengguna dan para pengguna konkuren yang lain dapat membaca record tetapi tidak mengubahnya.

2. Exclusive (X)

Kunci ini memungkinkan pengguna untuk membaca dan mengubah record. Sedangkan pengguna konkuren lain tidak diperbolehkan membaca ataupun mengubah record tersebut.

	X	S	-
X	N	N	Y
S	N	Y	Y
-	Y	Y	Y

X = kunci X

S = kunci S

N = No

Y = Yes

Cara Kerja Locking

Masalah kehilangan modifikasi (Lost Update Problem)

Transaksi A	Waktu	Transaksi B
-		-
baca R11 (kunci S)		-
-	t2	baca R(kunci S)
-	↓	-
modifikasi R (kunci X)	t3	-
tunggu	↓	-
⋮	t4	modifikasi R (kunci X)
⋮	↓	tunggu
⋮	↓	⋮
⋮	↓	⋮
tunggu	↓	tunggu

lanjutan

Masalah Modifikasi Sementara (uncommitted Update Problem)

Transaksi A	Waktu	Transaksi B
-	↓	-
-	t1	modifikasi R (kunci X)
-	↓	-
baca R	t2	-
kunci (S)	↓	-
tunggu	↓	-
⋮	t3	synchpoint (kunci X dilepas)
⋮	↓	-
tunggu	t4	-
baca R kembali (Kunci S)	↓	-

lanjutan

Transaksi A	Waktu	Transaksi B
-	t1	modifikasi R
-	↓	(kunci X)
Modifikasi R	t2	-
Kunci (X)	↓	-
tunggu	↓	-
:	t3	synchpoint
:	↓	(kunci X dilepas)
tunggu	↓	-
modifikasi R	t4	-
(Kunci X)	↓	-

lanjutan

Masalah Analisa yang tidak konsisten (Problem of inconsistency Analisa)

⊕ Nilai 1 = 40	Nilai 2 = 50	Nilai 3 = 30
Transaksi A	Waktu	Transaksi B
-	↓	-
Baca nilai 1(40)	t1	-
(kunci S)	↓	-
Juml=40	↓	-
-	↓	-
Baca nilai 2(50)	t2	-
(kunci S)	↓	-
Juml=90	↓	-
-	t3	↓
-	↓	baca nilai 3(30)
-	↓	(kunci S)
-	t4	-
-	↓	modifikasi nilai 3
-	↓	(kunci X)
-	↓	30 → 20
-	t5	-
-	↓	baca nilai 1(40)
-	↓	(kunci S)
-	t6	-
-	↓	modifikasi nilai 1
-	↓	(kunci X)
modifikasi nilai 3	t7	tunggu
(kunci S)	↓	⋮
tunggu	↓	tunggu

Timestamping

TIMESTAMPING

Adalah salah satu alternatif mekanisme kontrol konkurensi yang dapat menghilangkan masalah dead lock

Dua masalah yang timbul pada Timestamping :

1. Suatu transaksi memerintahkan untuk membaca sebuah item yang sudah di update oleh transaksi yang belakangan.
2. Suatu transaksi memerintahkan untuk menulis sebuah item yang nilainya sudah dibaca atau ditulis oleh transaksi yang belakangan

Crash dan Recovery

PENGERTIAN :

Crash adalah suatu failure atau kegagalan dari suatu sistem

PENYEBAB DARI KEGAGALAN ADALAH :

1. Disk Crash yaitu informasi yang ada di disk akan hilang
2. Power failure yaitu informasi yang disimpan pada memori utama dan register akan hilang
3. Software Error yaitu output yang dihasilkan tidak betul dan sistem databasenya sendiri akan memasuki suatu kondisi tidak konsisten

Klasifikasi Failure

Berdasarkan Jenis storage

1. Volatile storage, biasanya informasi yang terdapat pada volatile akan hilang, jika terjadi kerusakan sistem (system crash) contoh: RAM
2. Non Volatile Storage, biasanya informasi yang terdapat pada non volatile storage tidak akan hilang jika terjadi kerusakan sistem contoh: ROM
3. Stable Storage, informasi yang terdapat dalam stable storage tidak pernah hilang. contoh: Harddisk RAID

Jenis-Jenis Kegagalan

1. Logical Error, program tidak dapat lagi dilaksanakan disebabkan oleh kesalahan input, data tidak ditemukan, over flow
2. System Error, sistem berada pada keadaan yang tidak diinginkan, seperti terjadi deadlock, sebagai akibat program tidak dapat dilanjutkan namun setelah beberapa selang waktu program dapat dijalankan kembali.
3. System Crash,kegagalan fungsi perangkat keras, menyebabkan hilangnya data pada volatile storage, tetapi data pada non volatile storage masih tetap ada.
4. Disk Failure, hilangnya data dari sebuah blok disk disebabkan oleh kerusakan head atau kesalahan pada waktu pengoperasian transfer data

Security

SECURITY adalah suatu proteksi data terhadap perusakan data dan pemakaian oleh pemakai yang tidak mempunyai ijin.

BEBERAPA MASALAH SECURITY SECARA UMUM :

1. Di dalam suatu perusahaan siapa yang diijinkan untuk mengakses suatu sistem
2. Bila sistem tersebut menggunakan password, bagaimana kerahasiaan dari password tersebut dan berapa lama password tersebut harus diganti
3. Di dalam pengontrolan hardware, apakah ada proteksi untuk penyimpanan data (data storage)

lanjutan

DUA KATAGORI PENYALAHGUNAAN DATABASE :

1. Katagori yang tidak disengaja
Contoh: Anomali yang disebabkan oleh pendistribusian data pada beberapa komputer
2. Katagori yang disengaja
Contoh: Insert, Delete & Update oleh pihak yang tidak berwenang

BEBERAPA TINGKATAN MASALAH SECURITY :

1. Phisical, berkaitan dengan pengamanan lokasi fisik database
2. Man, berkaitan dengan wewenang user
3. Sistem operasi, berkaitan dengan keamanan sistem operasi yang digunakan dalam jaringan
4. Sistem database, sistem dapat mengatur hak akses user







Pemberian Wewenang dan View

KONSEP VIEW adalah cara yang diberikan pada seorang pemakai untuk mendapatkan model database yang sesuai dengan kebutuhan perorangan

Database relational membuat pengamanan pada level :

- Relasi, seorang pemakai diperbolehkan atau tidak mengakses langsung suatu relasi
- View, seorang pemakai diperbolehkan atau tidak mengakses data yang terdapat pada view
- Read Authorization, data dapat dibaca tapi tidak boleh dimodifikasi
- Insert AuthORIZATION, pemakai boleh menambah data baru, tetapi tidak dapat memodifikasi data yang sudah ada

lanjutan

-  Update Authorization, pemakai boleh memodifikasi tetapi tidak dapat menghapus data
-  Delete Authorization, pemakai boleh menghapus data
-  Index Authorization, pemakai boleh membuat atau menghapus index
-  Resource Authorization, mengizinkan pembuatan relasi – relasi baru
-  Alternation Authorization, mengizinkan penambahan atau penghapusan atribut dalam satu relasi
-  Drop Authorization, pemakai boleh menghapus relasi yang ada

Integrity

Integrity

Berarti memeriksa keakuratan dan validasi data

BEBERAPA JENIS INTEGRITY :

1. **Integrity Konstains**, memberikan suatu sarana yang memungkinkan pengubahan database oleh pemakai berwenang sehingga tidak akan menyebabkan data inkonsistensi
2. **Integrity Rule** (pada basisdata relational), terbagi menjadi:
 - *Integrity Entity*, contoh: tidak ada satu komponen kunci primer yang bernilai kosong (null)
 - *Integrity Referensi*, suatu domain dapat dipakai sebagai kunci primer bila merupakan atribut tunggal pada domain yang bersangkutan