








# RECOVERY SYSTEM

Sistem Basis Data

Gentisya Tri Mardiani, S.Kom., M.Kom

# Ilustrasi 1



Waktu	Kegiatan dan Kejadian	Ilustrasi
06:00	Bob membackup DATAFILE ke DVD Teknisi menonaktifkan UPS untuk di perbaiki	
08:00	Yuni melakukan penarikan uang Rp 2.000.000	 Pengambilan Rp 2.000.000
09:00	Perusahaan listrik memutuskan aliran listrik Server basis data bank hemat mati secara tiba-tiba.	
09:15	Aliran listrik kembali menyala. Bob menghidupkan kembali server Basis Data dan menemukan bahwa Hard Disk rusak, lalu Bob mentransfer data backup terakhir (jam 06:00) ke Hard Disk baru.	
Hari Esok	Yuni menemukan bahwa saldo tabungannya masih Rp. 5.000.000, bukan Rp 3.000.000 seperti seharusnya.	Inkonsistensi Data 



- Pada ilustrasi, merupakan salah satu contoh terbaiknya sifat durability yang harus dimiliki oleh transaksi.
- Sifat durability berarti “setelah suatu transaksi commit, perubahan yang terjadi harus tetap bertahan di dalam basis data”.
- Kejadian tersebut dapat diatasi dengan melakukan penelusuran catatan log transaksi.

# LOG



- Log adalah catatan transaksi yang berjalan pada server basis data secara mendetail.
- Setiap record pada log menggambarkan operasi tunggal transaksi yang berisi data:
  - Nama transaksi: nama unik transaksi yang menjalankan operasi write.
  - Nama data item yang dituliskan
  - Nilai lama: nilai data item sebelum operasi write
  - Nilai baru: nilai setelah operasi write

# Contoh isi Log



Record No.	Isi Redo Log Record	Status
Record 1	$\langle T_0, \text{begin} \rangle$	Berhasil
Record 2	$\langle T_0, \text{Saldo Yuni, 5.000.000, 3.000.000} \rangle$	Berhasil
Record 3	$\langle T_0, \text{commit} \rangle$	Berhasil

# Log dalam durability transaksi



- **Operasi Redo Transaksi**

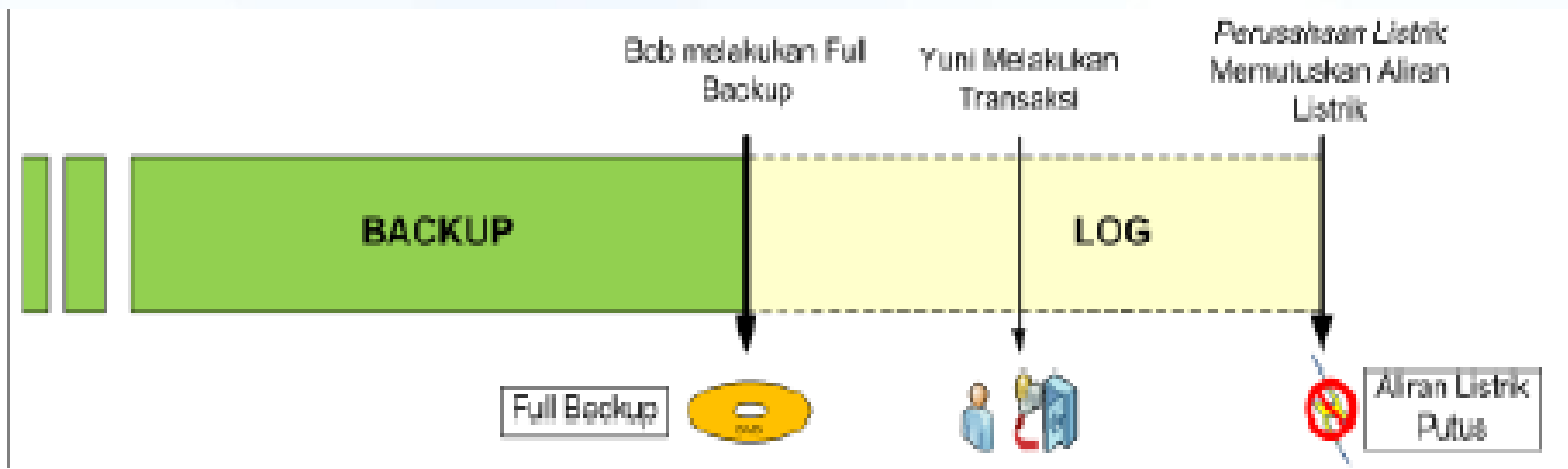
Proses dengan cara me-redone (mengulang) seluruh transaksi yang terekam di dalam log. Operasi ini juga biasa disebut “Forward Recovery”

operasi redo transaksi membutuhkan dua buah record yang berisi **<Ti begin>** dan **<Ti commit>**

# Log dalam durability transaksi

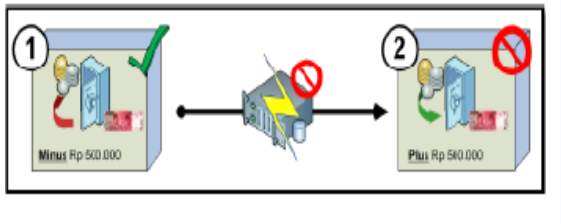


- Sistem melakukan penelusuran Log file dari backup terakhir sampai terjadinya kegagalan sistem. Seluruh transaksi yang terekam pada log file tersebut dieksekusi ulang, sehingga data dari transaksi tersebut kembali.



# Ilustrasi 2



Waktu	Kegiatan dan Kejadian	Ilustrasi
	<p>Ayu memiliki saldo Rp. 7.000.000 Tara memiliki saldo Rp. 45.000 Tara meminta dikirimkan uang sebesar Rp.500.000 dari Ayu.</p>	
09.00	<p>Ayu melakukan transaksi transfer uang ke saldo Tara, namun sebelum transaksi ini berhasil sepenuhnya, tiba-tiba aliran listrik terputus.</p>	



# Ilustrasi 2



- Ada dua operasi yang harus dilakukan, yaitu:
  - Mengurangi saldo pada nasabah Ayu sebesar Rp. 500.000
  - Menambah saldo pada nasabah Tara sebesar Rp.500.000
- Transaksi tersebut belum berhasil sempurna (*committed*), dan transaksi ini harus dibatalkan (*rollback*) sesuai dengan sifat atomik yang harus dimiliki transaksi.

# Log dalam keatomikan dan konsistensi transaksi



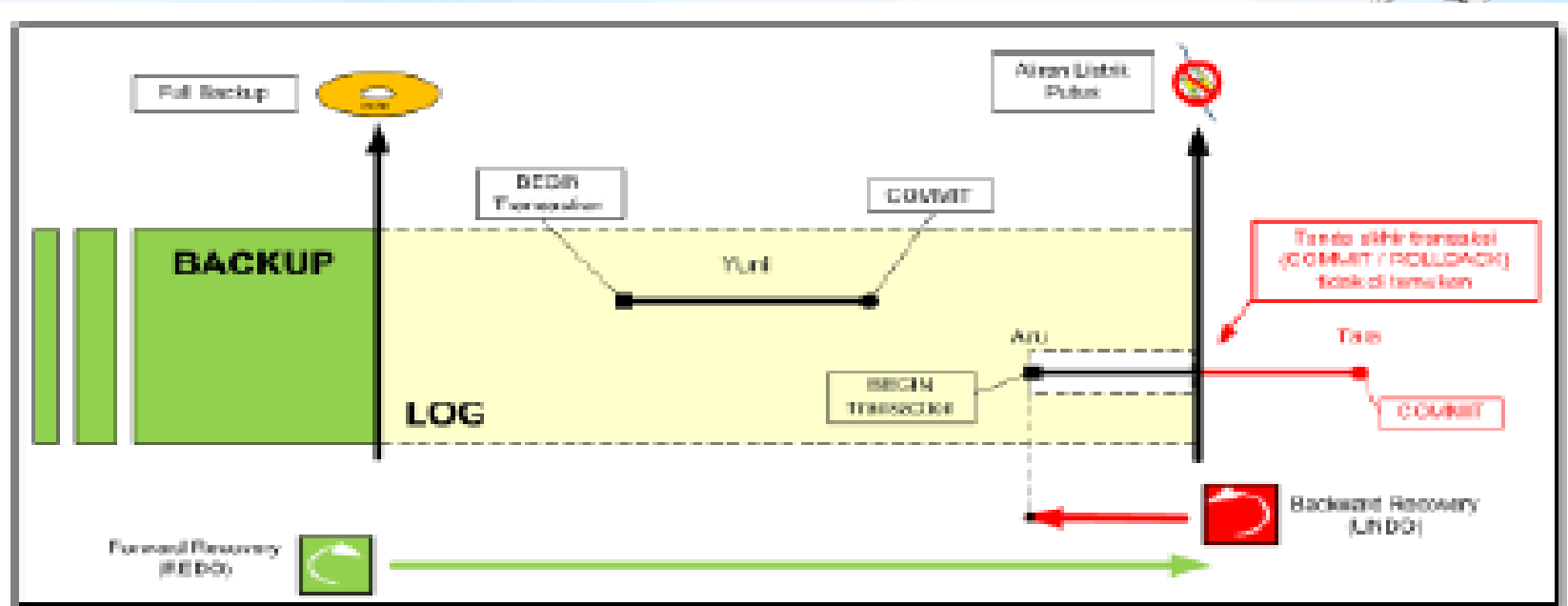
- **Operasi Undo transaksi**

proses dengan membatalkan (*un-done*) perubahan terakhir sampai titik ditemukannya catatan awal transaksi **<begin>** pada log.

Operasi ini biasa disebut “Backward Recovery”

- Operasi ini berjalan saat mekanisme recovery RDBMS menjalankan operasi redo dengan menelusuri log transaksi dari awal sampai menemukan transaksi yang tidak diakhiri dengan COMMIT atau ROLLBACK.

# Log dalam keatomikan dan konsistensi transaksi



- Secara otomatis transaksi transfer uang Ayu tersebut harus dibatalkan (*rollback*), sehingga saldo Ayu kembali utuh.

# Log pada storage



- Media penyimpanan sementara (volatile storage)
  - Data akan disimpan hanya jika ada aliran listrik, jika aliran listrik terputus maka data akan hilang.
  - Contoh: RAM, cache, dan register.
- Media penyimpanan permanen (nonvolatile storage)
  - Data akan tetap tersimpan walaupun aliran listrik terputus.
  - Contoh: harddisk, optical disk, dsb.



- Setiap record pada log menggambarkan operasi tunggal transaksi.
- Catatan transaksi (log) disimpan pada nonvolatile storage, agar saat terjadi putusnya aliran listrik data tidak hilang dan dapat digunakan untuk recovery.

# Contoh



Record No.	Isi Redo Log Record	Status
Record 1	$\langle T_0 \text{ begin} \rangle$	Berhasil
Record 2	$\langle T_0 \text{ Saldo Yuni, } 5.000.000, 3.000.000 \rangle$	Berhasil
Record 3	$\langle T_0 \text{ commit} \rangle$	Berhasil
Record 4	$\langle T_1 \text{ begin} \rangle$	Berhasil
Record 5	$\langle T_1 \text{ Saldo Ayu, } 7.000.000, 6.500.000 \rangle$	Berhasil
Record 6	$\langle T_1 \text{ Saldo Tara, } 45.000, 545.000 \rangle$	Gagal
Record 7	$\langle T_1 \text{ commit} \rangle$	Gagal



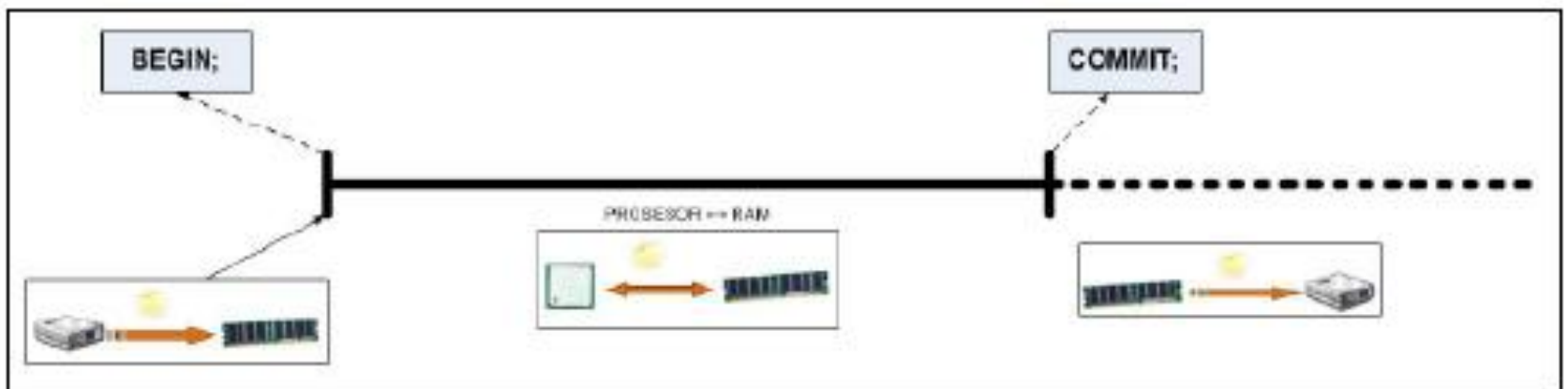
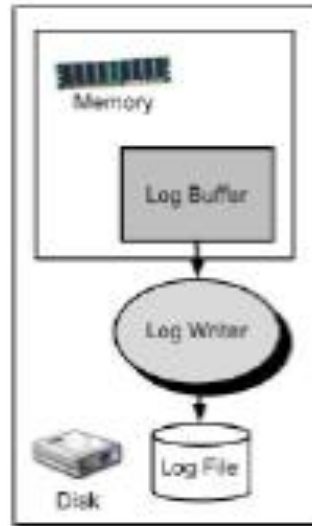
- Penyimpanan pada nonvolatile storage:
  - Kecepatan akses lambat.
  - Ukuran record log yang kecil, sehingga sistem terbebani dengan operasi write ke harddisk untuk setiap record.
- Gunakan volatile storage (RAM) dengan buffering, yaitu mengumpulkan record yang berukuran kecil tersebut sampai berukuran cukup besar, baru dituliskan ke harddisk.



- DBMS membuat proses *Log writer* yang menuliskan isi buffer dari memory ke harddisk.
- Transaksi yang belum diCOMMIT, data log akan berada pada RAM, setelah transaksi COMMIT, maka data log tersebut dituliskan ke dalam harddisk.



# Log pada storage



# DATAFILE



- DATAFILE biasa disebut DATABASE karena fungsinya untuk menyimpan data.
- Pada DBMS, database bukan hanya sebagai tempat menyimpan table, namun juga seluruh data yang dibutuhkan aplikasi DBMS untuk mengelola aplikasinya.
- Database yang disimpan pada harddisk disebut DATAFILE, database yang dibuffer pada RAM disebut *database buffer*.

# DATAFILE



- Datafile adalah file fisik yang menyimpan data yang telah dimasukkan ke dalam tabel pada DBMS.
- DATAFILE: tempat menyimpan data.
- LOG FILE: tempat menyimpan catatan bahwa data tersebut telah disimpan.

# DATAFILE



- Datafile menggunakan buffer untuk mempercepat pengaksesan data oleh banyak user.

Datafile suatu perusahaan berukuran ratusan hingga ribuan gigabyte, biasanya sering terjadi pengaksesan data bersamaan.

Kebutuhan akses multiuser datafile dapat diatasi dengan menggunakan buffer pada RAM.

# DATAFILE



- Penulisan perubahan pada *database buffer* di RAM ke dalam datafile di harddisk terjadi saat kapasitas *database buffer* penuh dan log telah dituliskan sebelumnya.

# DATAFILE



- Jika data transaksi yang telah diCOMMIT pada database buffer hilang (karena kegagalan sistem), maka transaksi tersebut dapat dipulihkan dengan menelusuri catatannya di Log file.

# Kegagalan Sistem



- Kegagalan sistem dapat disebabkan oleh putusnya aliran listrik, atau karena aplikasi DBMS harus di-restart, sehingga isi *data buffer* pada RAM hilang, baik *log buffer* maupun *database buffer*.

# Proses Recovery



- Pada saat sistem restart dan melakukan proses recovery, sistem akan membuat dua daftar:
  - [daftar undo] yang berisi transaksi yang harus dibatalkan (*rollback*)
  - [daftar redo] yang berisi transaksi yang harus dikerjakan ulang.



# Operasi Backup



- Berdasarkan waktu pelaksanaan atau strateginya, terdapat 2 operasi backup yang dapat dipilih:
  - **Backup statis**, backup dilakukan dengan lebih dahulu menonaktifkan basis data secara keseluruhan
  - **Backup dinamis**, backup dilakukan tanpa menonaktifkan basis data, sehingga user dapat tetap bekerja

# Backup statis



- Backup statis dapat dilakukan melalui program yang disediakan sistem operasi atau dengan menjalankan modul khusus yang disediakan DBMS.
- Proses backup dilakukan dengan penyalinan objek basis data secara keseluruhan, sehingga hasil operasi backup akan sama persis dengan basis data yang ada.

# Backup statis



- Pada saat kerusakan/ kehilangan basis data terjadi, maka salinan hasil operasi backup statis dapat langsung dilakukan proses *restore*.
- Namun, hasil dari restore ini belum tentu dapat mengembalikan kondisi basis data ke kondisi semula sebelum kerusakan terjadi, tetapi kondisi basis data pada saat backup dilakukan.

# Backup dinamis



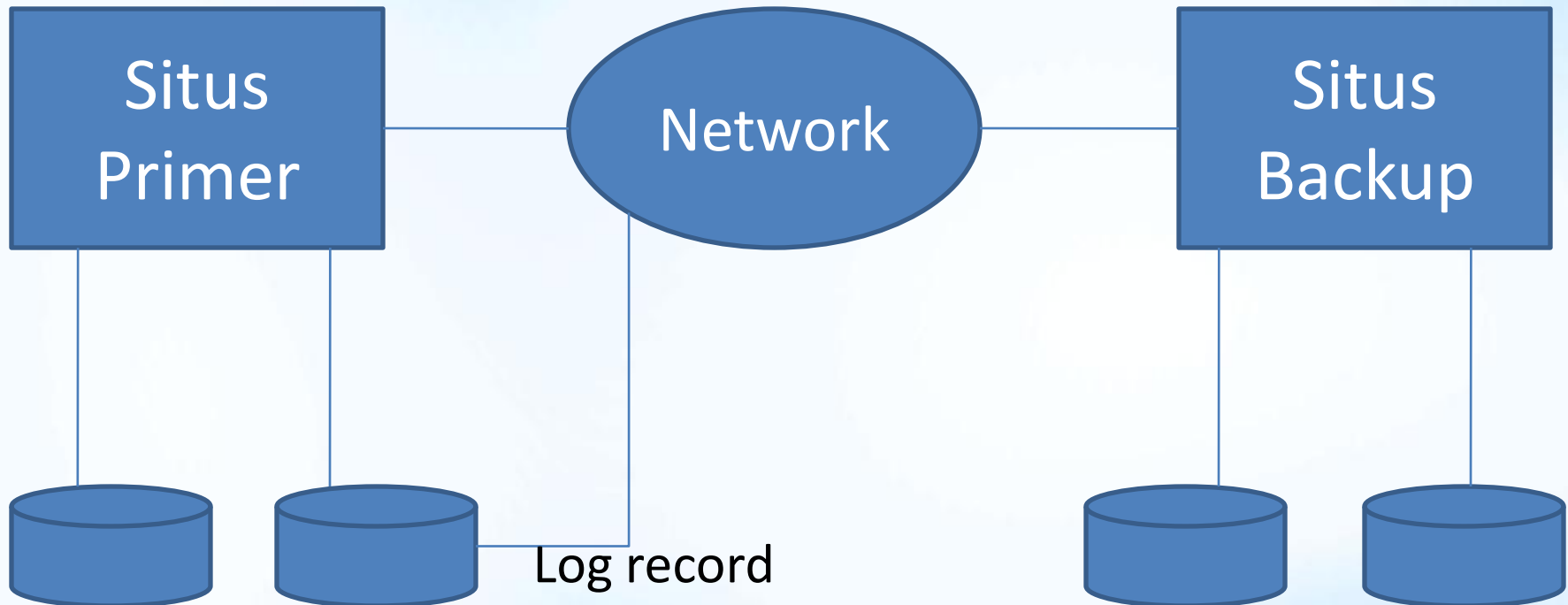
- Backup dinamis dilakukan selama basis data masih aktif, dan harus dilakukan secara hati-hati karena dapat mengganggu jalannya operasi utama terhadap basis data, (seperti manipulasi data)
- Operasi backup dilakukan secara selektif, yaitu hanya terhadap tabel-tabel yang mengalami perubahan data.

# Backup dinamis



- Backup dilakukan secara periodik, DBMS lebih dahulu melakukan pembentukan *file dump* di media penyimpanan stabil.
- File dump berisi salinan semua basis data sebelum terjadinya perubahan basis data.
- Pada saat kerusakan/ kehilangan basis data terjadi, maka file dump tersebut yang akan digunakan untuk operasi *restore*, kemudian dilanjutkan dengan operasi *recovery*.

# Backup jarak jauh (*remote backup system*)





- Situs remote backup tersebut harus terus disinkronisasi dengan situs primer ketika terjadi perubahan pada situs primer
- Sinkronisasi dilakukan dengan mengirim semua record log dari situs primer ke situs backup
- Ketika situs primer mengalami kerusakan, maka situs remote backup akan langsung mengambil alih pemrosesan.

# Pemilihan cara backup



- Jika basis data harus terus diaktifkan sepanjang waktu, maka dilakukan **backup dinamis**.
- Jika terdapat periode istirahat di mana basis data dapat dinonaktifkan sementara, maka **backup statis** dapat dilakukan secara reguler, kemudian ditambah dengan backup dinamis dari beberapa tabel yang sering digunakan.