

## MEMAHAMI PERAN DARI MEMBATALKAN DATA DALAM DATABASE ORACLE

### UNDERSTANDING THE ROLE OF UNDO DATA IN ORACLE DATABASE

Adibah Dewi Satriani<sup>1\*</sup>, Ida Ayu Ernawati<sup>1</sup>

\*E-mail: [dibah.satriani@gmail.com](mailto:dibah.satriani@gmail.com)

<sup>1</sup>Sistem Informasi, Fakultas Ilmu Komputer, UPN “Veteran” Jawa Timur

#### Abstrak

Artikel ini membahas peran penting *undo data* dalam *Oracle Database*. *Undo data* merupakan informasi yang diperlukan untuk membatalkan atau mengembalikan transaksi dalam basis data. *Oracle Database* menggunakan *undo data* untuk memastikan konsistensi dan pemulihan data saat terjadi *rollback* transaksi atau pemulihan setelah kegagalan sistem. Artikel ini mengungkapkan perbedaan antara manajemen undo otomatis serta manual, dengan penekanan pada manajemen otomatis yang memungkinkan *Oracle Database* secara otomatis mengelola *undo data* menggunakan *undo tablespace* serta *undo segments*. Penelitian ini juga membahas mode pembatalan otomatis (*Automatic Undo Mode*) dan penggunaan *automatic undo segments*, serta cara mengatasi masalah yang sering muncul seperti error ORA-1555 "*Snapshot too old*". Selain itu, pentingnya pemantauan *undo segments* dan ruang undo, serta pengukuran *undo tablespace* juga dibahas dalam artikel ini. Dengan pemahaman yang baik tentang *undo data*, pengguna *Oracle Database* dapat mengoptimalkan pengelolaan serta pemulihan data pada sistem basis data mereka.

**Kata kunci:** *undo data, oracle, database*

#### Abstract

*This article discusses the important role of undo data in Oracle Database. Undo data is the information needed to undo or roll back transactions in a database. Oracle Database uses undo data to ensure consistency and data recovery in case of transaction rollbacks or system failures. The article reveals the differences between automatic and manual undo management, with an emphasis on automatic management that allows Oracle Database to automatically manage undo data using undo tablespace and undo segments. The study also discusses Automatic Undo Mode and the use of automatic undo segments, as well as how to address common issues such as ORA-1555 "Snapshot too old" error. Furthermore, the article highlights the importance of monitoring undo segments and undo space, as well as measuring undo tablespace. With a good understanding of undo data, Oracle Database users can optimize data management and recovery in their database systems.*

**Keywords:** *undo data, oracle, database*

## 1. PENDAHULUAN

Pada era digital saat ini, penggunaan basis data menjadi sangat penting dalam berbagai bidang, termasuk dalam manajemen sistem informasi dan pemrosesan data. Banyak contoh permasalahan dalam suatu perusahaan yang mengalami kerusakan data sebelum melakukan backup, maka dari itu dalam penelitian ini menjelaskan beberapa bentuk dari *undo data* dan bagaimana cara menyelamatkan database atau mengembalikan database dengan melakukan *undo data*. Salah satu sistem basis data yang terkenal dan sering digunakan adalah *Oracle Database*. *Oracle Database*

memiliki berbagai fitur dan komponen yang membantu dalam pengelolaan dan pemulihan data. Salah satu komponen penting dalam *Oracle Database* adalah *undo data*. *Oracle Database* merupakan salah satu sistem manajemen basis data yang paling populer digunakan di berbagai perusahaan dan organisasi. Dalam lingkungan database ini, terdapat fitur yang penting dan kritis yang dikenal sebagai *undo data*. Dalam jurnal ini, kami akan membahas mengenai peran penting dari *undo data* dalam *Oracle Database*.

*Undo data* merujuk pada informasi yang diperlukan untuk membatalkan atau mengembalikan transaksi yang telah dilakukan dalam basis data. Dalam konteks *Oracle Database*, *undo data* digunakan untuk memastikan konsistensi dan pemulihan data pada saat terjadi *rollback* transaksi atau pemulihan setelah kegagalan sistem. Peran penting *undo data* dalam *Oracle Database* dapat lebih dipahami dengan mempelajari beberapa aspek terkait. Salah satu aspek yang perlu dipelajari adalah perbedaan antara manajemen undo otomatis (*Automatic Undo Management*) dan manual (*Manual Undo Management*). Dalam manajemen undo otomatis, *Oracle Database* secara otomatis mengelola *undo data* dengan mengalokasikan dan mengelola *undo space*. Sedangkan dalam manajemen *undo* manual, pengelolaan *undo data* dilakukan secara manual oleh pengguna, termasuk alokasi dan pemulihan *undo space*.

*Automatic Undo Mode* adalah fitur dalam *Oracle Database* yang memungkinkan pengguna untuk mengaktifkan manajemen undo otomatis. Dalam mode ini, *Oracle Database* secara otomatis mengelola *undo data* dengan menggunakan *undo tablespace* untuk menyimpan *undo segments*. *Undo tablespace* adalah ruang penyimpanan yang digunakan untuk menyimpan *undo segments*, yang merupakan unit penyimpanan *undo data*.

Penggunaan *automatic undo segments* memungkinkan *Oracle Database* untuk secara otomatis mengelola jumlah dan ukuran segmen undo yang dibutuhkan. Ini memudahkan pengguna dalam mengelola *undo data* tanpa harus melakukan tugas manual yang rumit. Pengguna juga dapat mengontrol penggunaan ruang undo dengan mengatur parameter dan konfigurasi yang sesuai.

Salah satu masalah yang sering muncul terkait *undo data* adalah *error* ORA-1555 "*Snapshot too old*". Untuk mengurangi kemungkinan terjadinya *error* ini, pengguna dapat mengatur parameter dan konfigurasi yang tepat, termasuk menyesuaikan ukuran *undo tablespace* dan memantau penggunaan *undo segments* dan ruang *undo* secara teratur.

Pemantauan *undo segments* dan ruang *undo* juga merupakan aspek penting dalam pengelolaan *undo data*. Pengguna perlu memastikan bahwa *undo segments* memiliki ukuran dan kapasitas yang memadai, serta memonitor penggunaan ruang *undo* untuk mencegah kehabisan ruang *undo*. Terakhir, pengguna perlu memperhatikan pengukuran (*sizing*) *undo tablespace* agar dapat menampung *undo data* yang cukup sesuai dengan kebutuhan aplikasi dan transaksi yang berjalan di *Oracle Database*. Pengukuran yang tidak tepat dapat menyebabkan masalah kinerja dan pemulihan data yang tidak efisien.

Dalam jurnal ini, kami akan membahas lebih detail terkait peran penting *undo data* dalam *Oracle Database*, termasuk perbedaan antara *automatic* dan *manual undo management*, *automatic undo mode*, *undo tablespace*, *automatic undo segments*, penggunaan *automatic undo segments*, cara meminimalkan *error* ORA-1555 *snapshot too old*, pengendalian penggunaan ruang *undo*, pemantauan *undo segments* dan *undo space*, serta pengukuran *undo tablespace*. Dengan pemahaman yang baik tentang *undo data*, pengguna *Oracle Database* dapat mengoptimalkan pengelolaan dan pemulihan data dalam sistem basis data mereka.

## 2. LANDASAN TEORI

Pada bab ini, kami akan menjelaskan landasan teori yang meliputi *oracle*, *database*, *rollback*, *commit*, dan *CMD*. *Oracle* merupakan DBMS terkemuka yang menyediakan fitur dan fungsionalitas untuk mengelola dan mengakses data secara efisien. Pengguna dapat menggunakan tools *CMD* dan *sqlplus* untuk mengubah atau membuat *database*. *Undo data* digunakan untuk

melacak perubahan dalam transaksi, sementara *rollback* dan *commit* digunakan untuk mengelola konsistensi dan keandalan data. Pemahaman tentang komponen-komponen ini akan membantu pengguna mengoptimalkan penggunaan *oracle* dan mengelola *database* dengan efektif.

### **2.1 Oracle**

*Oracle* adalah basis data relasional yang terdiri dari kumpulan data dalam suatu sistem manajemen basis data Relational Database Management System (RDBMS). Perusahaan perangkat lunak Oracle memasarkan jenis basis data ini untuk bermacam-macam aplikasi yang bisa berjalan pada banyak jenis dan merk perangkat keras komputer (platform). Basis data Oracle ini pertama kali dikembangkan oleh Larry Ellison, Bob Miner dan Ed Oates lewat perusahaan konsultasi nya bernama Software Development Laboratories (SDL) pada tahun 1977. Pada tahun 1983, perusahaan ini berubah nama menjadi Oracle Corporation sampai sekarang. Oracle menggunakan SQL (Structured query language) sebagai bahasa perantara antara user dan database. Kemudian Oracle Corp mengembangkan sebuah bahasa prosedural yang mereka sebut sebagai PL/SQL. Dengan menggunakan kedua bahasa ini, seorang user dapat mengoptimalkan penggunaan database Oracle. Oracle mendukung data yang sangat besar, menurut dokumentasi Oracle dapat menampung sampai 512 petabytes (1 petabytes = 1.000.000 gigabytes) atau sekitar 512.000.000 gigabytes[1].

### **2.2 Database**

Database atau basis data adalah kumpulan informasi yang disimpan dalam komputer secara sistematis sehingga dapat diperiksa menggunakan suatu program komputer untuk memperoleh informasi. Database adalah suatu aplikasi yang menyimpan sekumpulan data. Setiap database mempunyai perintah tertentu untuk membuat, mengakses, mengatur, mencari, dan menyalin data yang ada di dalamnya.

Maka dari beberapa pendapat dapat disimpulkan bahwa database atau basis data adalah suatu aplikasi yang berisi sekumpulan informasi yang tersimpan secara sistematis di komputer[2].

### **2.3 Undo Data**

*Undo data* merupakan informasi yang penting dalam basis data, yang digunakan untuk membatalkan atau mengembalikan perubahan yang terjadi. Undo data adalah informasi yang diperlukan untuk membatalkan atau mengembalikan perubahan dalam basis data[3]. Undo data juga digunakan untuk memastikan konsistensi dan pemulihan data setelah terjadi kegagalan sistem[4].

### **2.4 Rollback**

*Rollback* adalah mekanisme dalam database Oracle yang digunakan untuk membatalkan transaksi yang belum selesai atau tidak diinginkan. Rollback memungkinkan pengguna untuk mengembalikan database ke kondisi sebelumnya sebelum transaksi dilakukan.

Proses rollback melibatkan pengembalian database ke keadaan sebelum transaksi dimulai dan memastikan bahwa perubahan yang dibuat tidak berdampak pada data dalam database. Setiap transaksi di Oracle memiliki log transaksi, yang mencatat semua perubahan yang dibuat selama transaksi. Jika transaksi di-rollback, semua perubahan yang dilakukan selama transaksi akan dihapus dari log transaksi, sehingga database kembali ke keadaan sebelumnya[5].

### **2.5 Commit**

*Commit* biasa digunakan untuk mengakhiri transaksi, dan membuat transaksi yang terakhir kali dilakukan menjadi permanen. ketika transaksi dimulai, maka oracle akan mengelola perubahan data yang dilakukan oleh user, dan ketika dilakukan commit maka database akan memvalidasi perubahan yang telah dilakukan dan menyimpan transaksi secara permanen ke dalam file database. Commit memiliki peran yang cukup penting karena dapat menjaga ketahanan data secara permanen[6].

## 2.6 CMD

*Command Prompt* atau DOS prompt adalah sebuah *command line* (baris perintah) pada sebuah OS (*Operating System*) berbasis GUI untuk mengeksekusi file dengan cara menuliskan perintahnya pada jendela CMD[7].

Dalam artikel ini dilakukan proses transaksi pada oracle database, dengan menggunakan tools CMD yang ada didalam komputer pengguna. Dalam melakukan praktek menggunakan CMD serta tools dari sqlplus untuk melakukan perintah mengubah atau membuat database.

## 3. HASIL DAN PEMBAHASAN

### 3.1 Automatic VS Manual Undo Management

Automatic Undo Management (AUM) dan Manual Undo Management (MUM) adalah dua mode yang dapat digunakan dalam pengelolaan undo dalam database Oracle.

1. Automatic Undo Management (AUM):

Dalam mode AUM, oracle menggunakan mekanisme otomatis untuk mengelola undo data. Undo data adalah informasi yang diperlukan untuk membatalkan atau mengulangi transaksi dalam database. Dalam AUM, Oracle secara otomatis mengelola ukuran, alokasi, dan penghapusan undo data.

Keuntungan dari AUM termasuk:

- Kemudahan penggunaan: Oracle secara otomatis mengelola undo data, sehingga pengguna tidak perlu melakukan tindakan tambahan untuk mengelolanya.
- Performa yang lebih baik: AUM menggunakan tabel undo yang dioptimalkan dan algoritma manajemen yang efisien, yang dapat menghasilkan kinerja yang lebih baik dalam operasi rollback dan transaksi paralel.

2. Manual Undo Management (MUM):

Dalam mode MUM, pengguna secara manual mengelola undo data dalam database oracle. Dalam mode ini, pengguna perlu membuat dan mengelola segmen rollback secara manual. Beberapa hal yang perlu diperhatikan dalam MUM adalah:

- Pengaturan parameter inisialisasi: Saat beralih dari AUM ke MUM, parameter inisialisasi yang relevan perlu diubah, seperti 'UNDO\_MANAGEMENT' yang diatur menjadi 'MANUAL'. Perubahan ini memerlukan restart instance database.
- Pembuatan segmen rollback: Jika segmen rollback belum ada, pengguna harus membuatnya secara manual saat beralih ke mode MUM.

Pada umumnya, jika menggunakan alat GUI Database Creation Assistant (DBCA) untuk membuat database, Oracle secara default akan menggunakan AUM, kecuali jika secara eksplisit mengatur untuk menggunakan MUM. Mode manajemen undo dapat diubah dari satu ke yang lain, namun memerlukan perubahan parameter inisialisasi dan restart instance database.

### 3.2 Automatic Undo Mode

Automatic Undo Mode (AUM) adalah fitur dalam Oracle Database yang mengontrol dan mengelola segmen undo secara otomatis. Saat menggunakan AUM, parameter inisialisasi UNDO\_MANAGEMENT harus disetel ke AUTO. Secara default, nilai parameter ini adalah MANUAL. Dalam mode AUM, tidak diizinkan untuk membuat, mengalokasikan, atau mengelola segmen undo secara manual. Tanggung jawab ini diambil alih oleh Oracle Database. Segmen undo yang dibuat oleh Oracle Database disebut Automatic Undo Segments (AUS). Tidak diperlukan untuk mencantulkannya di file init.ora menggunakan parameter ROLLBACK\_SEGMENTS.

Oracle Database mengontrol karakteristik fisik dari segmen undo ini. Tidak dapat menentukan atau mengubah parameter penyimpanan AUS secara manual. Oracle Database membuat segmen undo dalam tipe tablespace khusus yang disebut UNDO Tablespace untuk mencegah dan

mempengaruhi parameter penyimpanan segmen undo ini. Serta tidak dapat mengubah karakteristik penyimpanan dari tablespaces undo.

Dengan menggunakan AUM, Oracle Database secara otomatis mengelola segmen undo, termasuk mengalokasikan ruang undo yang diperlukan, mengelola siklus reuse undo, dan memastikan konsistensi transaksi dengan menyediakan informasi undo saat diperlukan. Fitur AUM menyederhanakan pengelolaan undo dan memungkinkan pengguna fokus pada pengembangan aplikasi dan analisis hasil dan pembahasan daripada mengelola segmen undo secara manual.

### 3.3 Managing Undo Data

Managing Undo Data merupakan salah satu konsep yang digunakan dalam sistem database untuk mengelola dan mempertahankan data undo atau perubahan yang terjadi pada suatu transaksi. Undo data mencakup semua perubahan yang dilakukan oleh transaksi, sehingga memungkinkan untuk mengembalikan database ke keadaan sebelum transaksi dilakukan[8].

Pada contoh yang diberikan, terdapat beberapa perintah SQL yang berkaitan dengan managing undo data.

1. Commit: Perintah ini digunakan untuk mengakhiri transaksi dan mengkomitmen perubahan yang dilakukan. Setelah perintah ini dieksekusi, perubahan yang terjadi menjadi permanen dan tidak bisa di-rollback.
2. Rollback: Perintah ini digunakan untuk membatalkan transaksi yang belum di-commit. Dengan menjalankan perintah rollback, semua perubahan yang dilakukan dalam transaksi tersebut akan dibatalkan dan database akan dikembalikan ke keadaan sebelum transaksi dilakukan.

```
SQL> -- CREATE TABLE --
SQL> Create table mahasiswa(
  2 nim number(12) not null,
  3 nama varchar2(50) not null,
  4 jurusan varchar2(50) not null);

Table created.

SQL> -- MELIHAT DESKRIPSI TABEL --
SQL> Desc mahasiswa;
-----
Name                               Null?    Type
-----
NIM                                 NOT NULL NUMBER(12)
NAMA                                NOT NULL VARCHAR2(50)
JURUSAN                             NOT NULL VARCHAR2(50)
```

Gambar 1. Praktek Managing Undo Data-1

```
SQL> -- INPUT 3 DATA KE TABEL --
SQL> -- DATA PERTAMA --
SQL> Insert into mahasiswa (nim, nama, jurusan) values ('21082010001','wahyu', 'informatika');

1 row created.

SQL> -- DATA KEDUA --
SQL> Insert into mahasiswa (nim, nama, jurusan) values ('21082010002','pram', 'informatika');

1 row created.

SQL> -- DATA KETIGA --
SQL> Insert into mahasiswa (nim, nama, jurusan) values ('21082010003','dimas', 'informatika');

1 row created.

SQL> -- MELIHAT ISI TABEL --
SQL> Select * from mahasiswa;
-----
NIM NAMA
-----
JURUSAN
-----
2.1082E+10 wahyu
informatika
2.1082E+10 pram
informatika
2.1082E+10 dimas
informatika

SQL> COMMIT;

Commit complete.

SQL> ROLLBACK;

Rollback complete.
```

Gambar 2. Praktek Managing Undo Data-2

```
SQL> -- MELIHAT ISI TABEL --
SQL> Select * from mahasiswa;
-----
NIM NAMA
-----
JURUSAN
-----
2.1082E+10 wahyu
informatika
2.1082E+10 pram
informatika
2.1082E+10 dimas
informatika
```

Gambar 3. Praktek Managing Undo Data-3

```
SQL> -- INPUT 1 DATA KE TABEL --
SQL> -- DATA KEEMPAT --
SQL> Insert into mahasiswa (nim, nama, jurusan) values ('21082010004','Fahrizal', 'informatika')
2 ;
1 row created.

SQL> -- MELIHAT ISI TABEL --
SQL> Select * from mahasiswa;

      NIM NAMA
-----
JURUSAN
-----
2.1082E+10 wahyu
informatika
2.1082E+10 pram
informatika
2.1082E+10 dimas
informatika

      NIM NAMA
-----
JURUSAN
-----
2.1082E+10 Fahrizal
informatika

SQL> ROLLBACK;
Rollback complete.
```

**Gambar 4. Praktek Managing Undo Data-4**

```
SQL> -- MELIHAT ISI TABEL --
SQL> Select * from mahasiswa;

      NIM NAMA
-----
JURUSAN
-----
2.1082E+10 wahyu
informatika
2.1082E+10 pram
informatika
2.1082E+10 dimas
informatika
```

**Gambar 5. Praktek Managing Undo Data-5**

Dalam contoh yang diberikan, terdapat penggunaan managing undo data dengan contoh penggunaan commit dan rollback. Pada percobaan pertama, setelah dilakukan insert data ke tabel, dilakukan commit sehingga data menjadi permanen. Namun, ketika dilakukan rollback, perubahan yang terjadi dibatalkan dan data kembali ke keadaan sebelum insert dilakukan. Kemudian, dilakukan lagi insert data ke tabel dan dilakukan commit. Setelah itu, dilakukan rollback, tetapi kali ini perintah rollback tidak berpengaruh terhadap data yang sudah di-commit sebelumnya. Hal ini menunjukkan bahwa data yang sudah di-commit tidak dapat di-rollback. Secara umum, managing undo data sangat penting dalam sistem database karena memungkinkan pengelolaan transaksi dan pemulihan data ke keadaan sebelumnya jika diperlukan[9].

### 3.4 Automatic Undo Segment

Automatic Undo Segments (Segmen Pembatalan Otomatis) adalah fitur dalam Oracle Database yang digunakan untuk menyimpan informasi rollback (pembatalan) untuk transaksi yang telah dilakukan. Segmen undo digunakan untuk memastikan konsistensi data dan memungkinkan pemulihan (recovery) dalam keadaan yang tidak terduga, seperti rollback transaksi atau pemulihan setelah kegagalan sistem. Ketika membuat undo tablespace di Oracle, beberapa segmen undo otomatis akan dibuat secara otomatis oleh sistem. Jumlah segmen undo yang dibuat tergantung pada platform yang digunakan, tetapi pada umumnya 10 segmen undo dibuat di dalam undo tablespace. Nama segmen undo otomatis diawali dengan "\_SYSSMU\$" yang menunjukkan bahwa segmen tersebut adalah segmen undo sistem yang dikelola secara otomatis.

Setiap segmen undo dibuat dengan dua ekstensi berukuran 64KB. Nilai MINEXTENTS disetel ke 2, yang berarti setiap segmen undo akan memiliki setidaknya dua ekstensi awal. Nilai MAXEXTENTS disetel ke unlimited, yang berarti segmen undo dapat secara dinamis memperluas dirinya sendiri jika diperlukan. Nama segmen undo otomatis terdiri dari beberapa bagian. "\_SYSSMU" menunjukkan bahwa ini adalah segmen undo sistem yang dikelola oleh Oracle. N adalah nomor segmen undo (USN) berikutnya yang tersedia untuk database. Jadi, jika database memiliki lima segmen undo yang ada dan membuat undo tablespace baru, segmen undo baru akan diberi nama mulai dari "\_SYSSMU6\$" hingga "\_SYSSMU15\$".

Informasi mengenai USN (nomor segmen undo) dapat ditemukan dalam tampilan V\$ROLLNAME, V\$ROLLSTAT, dan sebagai SEGMENT\_ID dalam tampilan DBA\_ROLLBACK\_SEGS. Dalam beberapa situasi, jika diperlukan menggunakan nama segmen undo otomatis dalam skrip SQL, perlu untuk mengapitnya dengan tanda kutip ganda karena cara Oracle menamai segmen ini. Jika mengalami kerusakan pada segmen undo dan database tidak dapat memulai atau melaporkan masalah kerusakan blok segmen undo, disarankan untuk menghubungi dukungan Oracle. Ada parameter tersembunyi "\_corrupted\_rollback\_segments" yang masih tersedia, namun sebaiknya tidak menggunakannya tanpa persetujuan dari dukungan Oracle.

### **3.5 Penggunaan Segment Undo Otomatis**

Seperti judul materi diatas penggunaan segment undo sepenuhnya dilakukan secara otomatis. Oracle akan mengaktifkan sejumlah segment undo saat instance dimulai. Jumlah ini awalnya tergantung pada parameter SESSIONS dalam file init.ora. Saat jumlah transaksi aktif maka akan meningkat, dan Oracle akan mencoba mengaktifkan segmen undo yang tidak aktif. Jika semua segmen undo sudah aktif dan masih ada ruang yang cukup dalam tablespace undo, Oracle akan membuat segmen undo tambahan dan mengaktifkannya agar transaksi baru dapat menggunakannya. Tujuan utama adalah satu segmen undo per transaksi. Ketika ruang dalam tablespace undo untuk membuat segmen undo lebih lanjut sudah habis, maka Oracle akan mulai menugaskan transaksi baru ke segmen undo yang sudah ada. Dalam mode AUM, Oracle berusaha meminimalkan kesalahan yang terkait dengan segmen undo. Upaya dilakukan untuk meminimalkan error kehabisan ruang undo (ORA-1562) dan kesalahan "snapshot too old" (ORA-1555)[10]. Yang pertama dicapai dengan menggunakan mekanisme yang disebut Dynamic Extents Transfer yaitu sistem manajemen basis data (SMBD) seperti Oracle, yang terjadi ketika sebuah objek, seperti tabel atau indeks, memerlukan ruang tambahan untuk menyimpan data. Sementara yang terakhir dicapai dengan memberikan mekanisme kepada DBA untuk mempengaruhi periode retensi informasi yang tercommit dalam segmen undo yang dapat digunakan oleh query yang berjalan lama.

### **3.6 Meminimalkan Kesalahan ORA-1555 "Snapshot too old"**

Dengan ukuran yang tepat untuk undo tablespace dan nilai yang sesuai untuk parameter inialisasi UNDO\_RETENTION, maka dapat meminimalkan terjadinya kesalahan eror ini. Nilai parameter UNDO\_RETENTION menentukan durasi, dalam detik, untuk mempertahankan informasi yang sudah committed dalam segmen undo. Informasi ini akan digunakan oleh query saat mereka perlu membangun tampilan konsisten baca dari data yang telah berubah. Nilai default untuk UNDO\_RETENTION normalnya adalah 900 detik (15 menit). Nilai ini dapat diubah secara dinamis untuk mempengaruhi retensi informasi undo yang kemudian di-commit, serta informasi undo yang sudah ter-commit tetapi belum kadaluarsa. User juga dapat memantau penggunaan ruang undo dan lamanya query berjalan di database dengan menggunakan tampilan undo baru untuk mendapatkan nilai parameter ini yang dapat diterima.

Namun, apabila user menggunakan AUM, hal tersebut tidak akan menghilangkan error ORA-1555. Karena Dynamic extents transfer akan menimpa informasi undo yang mungkin diperlukan untuk membangun tampilan konsisten baca dari data yang telah berubah. Error juga dapat terjadi jika tablespace undo aktif diubah, karena segmen undo yang berisi informasi untuk membangun tampilan konsisten dari data yang berubah sekarang dalam kondisi offline.

Berikut adalah contoh entri dari log peringatan yang menunjukkan pernyataan SQL yang dilaporkan dieksekusi selama 10904 detik sebelum terjadi error ORA-1555.

```
Tue Dec 23 14:19:28 2003  
ORA-01555 caused by SQL statement below (Query Duration=10904 sec, SCN:  
0x056f.b98b7787):  
Tue Dec 23 14:19:28 2003  
SELECT yr, clshdg, SUM(billed_amt), SUM(item_cnt), SUM(cust_cnt), SUM(pd_adv_cnt),  
SUM(qce_tot) FROM ( SELECT con.DIR_PUB_YR yr, clhd.CLSHDG_STD_NAM clshdg, con.CUST_ID ||  
con.SOURCE_SYSTEM_CDE custid, . . .
```

**Gambar 6. Query Error ORA-1555**

*Sumber: Jurnal Oracle Understanding Undo Data*

Contoh query diatas merupakan cara meningkatkan periode UNDO\_RETENTION menjadi lebih dari 10904 detik untuk menghindari error ORA-1555.

### 3.7 Mengendalikan Penggunaan Ruang UNDO

Fitur Dynamic Extents Transfer juga memungkinkan sebuah transaksi berjalan secara liar dan menggunakan seluruh ruang UNDO. Hal ini dapat mempengaruhi transaksi lainnya secara negatif. Untuk mengendalikan penggunaan ruang UNDO, Oracle memperkenalkan sebuah direktif rencana sumber daya baru. Direktif UNDO\_POOL harus digunakan untuk mendefinisikan batas kuota ruang UNDO, dalam hal byte yang ditulis ke tablespace UNDO. Ketika sesi yang tergabung dalam kelompok konsumen melebihi batas kuota ruang UNDO mereka, mereka tidak akan diizinkan untuk melakukan aktivitas DML lebih lanjut dan sesi yang menyebabkan kuota UNDO melebihi batas akan diakhiri dengan kesalahan ORA-30027. Pelanggaran kuota UNDO gagal mendapatkan %s (byte). Nilai default untuk direktif UNDO\_POOL adalah UNLIMITED. Hal ini memungkinkan semua sesi untuk menggunakan sejumlah ruang UNDO yang tidak terbatas. Dengan hal ini, user harus menggunakan Resource Manager untuk memanfaatkan direktif UNDO\_POOL.

### 3.8 Pemantauan Segmen Undo dan Ruang Undo

Meskipun user menggunakan AUM, tetapi tetap harus melakukan pemantauan penggunaan tablespace undo, menyesuaikan waktu retensi undo, menentukan ukuran tablespace undo, dan sebagainya. Hal ini masih belum sepenuhnya otomatis untuk menghilangkan keterlibatan DBA. Oracle menyediakan beberapa alat untuk membantu user melaksanakan tugas-tugas ini secara efisien[11].

Dua tampilan baru, DBA\_UNDO\_EXTENTS, V\$UNDOSTAT, dan tampilan OEM akan memberikan informasi seperti penggunaan segmen undo dan waktu eksekusi query. Beberapa informasi ini akan diperlukan untuk menentukan ukuran yang tepat untuk tablespace undo dan mengatur waktu retensi undo guna meminimalkan error ORA-1555: snapshot too old.

User juga dapat menggunakan tampilan yang sudah ada untuk memantau penggunaan undo. Tampilan V\$ROLLSTAT memberikan informasi tentang perpanjangan, pemulihan, dan pengurangan segmen undo. Tampilan V\$TRANSACTION memberikan informasi tentang segmen undo yang digunakan oleh transaksi.

### 3.9 Mengukur Undo Tablespace

Seperti yang telah dijelaskan sebelumnya, tampilan V\$UNDOSTAT menyediakan informasi penting untuk memantau dan menentukan ukuran ruang undo. Tujuannya adalah memiliki cukup ruang undo untuk mendukung beban transaksi dan mempertahankan informasi yang sudah di commit dalam waktu yang cukup lama sehingga query tidak akan mengalami error ORA-1555.

Dengan memantau tampilan V\$UNDOSTAT, Anda dapat mengetahui jumlah transaksi, jumlah blok undo yang digunakan, dan waktu eksekusi query maksimum. Setelah mengetahui jumlah transaksi, user dapat menggunakan rumus berikut untuk mengetahui berapa banyak ruang undo yang akan dibutuhkan.

Dan user dapat menggunakan query berikut untuk mengetahui seberapa banyak ruang undo yang diperlukan dalam melakukan undo dan mengetahui waktu retensi undo:

```
SQL> SELECT rd AS "Retention",
2 (rd * (udbps * overhead) + overhead) as "Bytes"
3 FROM
4 (SELECT value AS RD
5 FROM v$parameter
6 WHERE name = 'undo_retention'),
7 (SELECT (sum (undoblks) /
8 sum ( ((end_time - begin_time) * 86400))) as UDBPS
9 FROM v$undostat),
10 (SELECT value AS OVERHEAD
11 FROM v$parameter
12 WHERE name = 'db_block_size');

Retention
-----
Bytes
-----
900
111740.045
```

**Gambar 7. Praktek Mengukur Undo Tablespace**

Dalam contoh di atas, 111740.045 byte, akan diperlukan untuk mendukung 900 detik waktu retensi pembatalan berdasarkan laju pembuatan undo. Dengan memanfaatkan Oracle Enterprise Manager (OEM) untuk memonitor tingkat generasi undo dan kebutuhan ruang undo untuk retensi undo dapat dilakukan dengan cepat dan mudah. Interface menampilkan informasi mengenai tablespace undo yang sedang digunakan, retensi undo, serta tingkat rata-rata dan maksimum generasi undo yang terjadi.

#### 4. KESIMPULAN DAN SARAN

Berdasarkan materi diatas, dapat disimpulkan bahwa *undo data* merupakan informasi penting dalam *database* yang digunakan untuk membatalkan atau mengembalikan perubahan yang terjadi. Ada dua mode dalam pengelolaan *undo*, yaitu Manajemen Pembatalan Otomatis (*Automatic Undo Management*) dan Manajemen Pembatalan Manual (*Manual Undo Management*). Mode otomatis (*Automatic Undo Mode*) memungkinkan *Oracle Database* mengelola *undo data* secara otomatis, sementara mode manual (*Manual Undo Management*) melibatkan pengelolaan *undo data* secara manual oleh pengguna. *Error ORA-1555 "Snapshot too old"* dapat terjadi jika *undo tablespace* tidak memiliki cukup ruang untuk menyimpan *undo data* yang diperlukan.

Saran untuk user ketika menggunakan *segment undo data*:

1. Penting untuk mempertimbangkan penggunaan Manajemen Pembatalan Otomatis (*Automatic Undo Management*) karena kemudahan penggunaannya dan performa yang lebih baik.
2. Jika menggunakan Manajemen Pembatalan Manual (*Manual Undo Management*), pastikan untuk mengatur parameter inisialisasi yang relevan dan membuat segmen *rollback* secara manual.
3. Dalam mode *Automatic Undo Mode*, perhatikan dan monitor penggunaan *undo segments*, ruang *undo*, dan ukuran *undo tablespace* untuk menghindari *error ORA-1555 "Snapshot too old"*.
4. Untuk menghindari *error ORA-1555 "Snapshot too old"*, pastikan ukuran *undo tablespace* dan nilai parameter *UNDO\_RETENTION* disetel dengan benar.
5. Penting untuk memantau penggunaan ruang *undo* dan lamanya *query* berjalan di *database* untuk mengendalikan penggunaan ruang *undo* dan mencegah dampak negatif pada transaksi lainnya.

Karena dalam pengelolaan *undo data*, perlu mempertimbangkan penggunaan mode otomatis, memonitor penggunaan *undo segments* dan ruang *undo*, serta mengatur ukuran *undo tablespace* dan nilai parameter *UNDO\_RETENTION* dengan baik dan benar. Karena dengan melakukan hal ini, dapat meminimalkan kesalahan dan memastikan pemulihan yang konsisten dalam basis data.

## 5. DAFTAR RUJUKAN

- [1] Armiami., 2016. *Basis Data Oracle Fundamental*. Yogyakarta: Deepublish.
- [2] Kurtz, D. (2012). Tablespace. In *PeopleSoft for the Oracle DBA* (pp. 191-200). Berkeley, CA: Apress.
- [3] Johan, I. (2004). Oracle Flashback-9i to 10g. New Jersey Oracle User Group.
- [4] Litchfield, D. (2007). *Oracle Forensics Part 6: Examining Undo Segments, Flashback and the Oracle Recycle Bin*. <http://www.ngssoftware.com>
- [5] Deshpande, K. (n.d.). *Oracle9i: Understanding Automatic Undo Management and Flashback Query*.
- [6] Alkhatib, G., & Labban, R. S. (n.d.). Transaction Management in Distributed Database Systems: the Case of Oracle's Two-Phase Commit. In *Journal of Information Systems Education* (Vol. 13, Issue 2).
- [7] Yunita, K., Steganografi dengan metode penggabungan file melalui Command Prompt serta Steganalisis hasil dengan metode pengenalan gambar, kultur gambar RGB 24 BIT dan rentang ukuran pada file JPEG,7,2015.
- [8] Sakethi, D., & Rico Andrian, dan. (2008). *Pengaturan temporary files dan Undo Segment pada Database Oracle siakad online* (Vol. 14, Issue 1).
- [9] Srinivas, K., Khan, T., & Siva, G. (2010). Hunting down undo issues with Oracle: a practical approach. In *IJCSE International Journal on Computer Science and Engineering* (Vol. 02, Issue 08).
- [10] Nindito, H., Sano, A. V. D., & Condrobimo, A. R. (2016, November). Comparative study of storing unstructured data type between BasicFile and SecureFile in Oracle Database 12c. In *2016 International Conference on Information Management and Technology (ICIMTech)* (pp. 146-149). IEEE.
- [11] Goldstein, P., Dodge, D., Firpo, M., Minner, L., Lee, W. H. K., Kanamori, H., ... & Kisslinger, C. (2003). SAC2000: Signal processing and analysis tools for seismologists and engineers. *The IASPEI international handbook of earthquake and engineering seismology*, 81, 1613-1620.
- [12] Yunita, K., Steganografi dengan metode penggabungan file melalui Command Prompt serta Steganalisis hasil dengan metode pengenalan gambar, kultur gambar RGB 24 BIT dan rentang ukuran pada file JPEG,7,2015.