# ORACLE DATABASE ADMINISTRATION

1. Syawal Saputra

# KATA PENGANTAR

Pertama-tama saya panjatkan

DAFTAR ISI

**[ORACLE DATABASE ADMINISTRATION 1](#_Toc2515)**

**[KATA PENGANTAR 2](#_Toc11491)**

**[PENDAHULUAN 4](#_Toc11145)**

[1.1 Latar Belakang 4](#_Toc6467)

[1.2 Rumusan Masalah 4](#_Toc28546)

**[PREPARING LAB ENVIRONMENT 5](#_Toc14643)**

[2.1 Lab Requirements 5](#_Toc14019)

[2.2 Installation Oracle Linux 8 5](#_Toc11868)

**[INTRODUCTION 10](#_Toc13861)**

[3.1 Introduction DBMS 10](#_Toc21693)

[3.2 Introduction Oracle Database 10](#_Toc4622)

[3.3 Oracle Database Architecture 11](#_Toc7678)

**[BASIC ADMINISTRATION 14](#_Toc13098)**

[4.1 Instalasi Oracle Database 19c 14](#_Toc29178)

[4.2 Create Listener 14](#_Toc29013)

[4.3 Create Database 18](#_Toc32655)

[4.4 Connect to Database 26](#_Toc12527)

[4.5 Tablespace 28](#_Toc10023)

[4.6 Start and Shutdown Instance 28](#_Toc16181)

[4.7 Database Initialization Parameters 30](#_Toc25374)

[4.8 Data Dictionary 32](#_Toc8690)

**[FUNDAMENTAL ORACLE DATABASE MULTITENANCY 34](#_Toc22927)**

[5.1 CDB & PDB 34](#_Toc8291)

[5.2 Container Identification 34](#_Toc6892)

[5.3 Local Users and Common Users 35](#_Toc6004)

[5.4 Create CDB dan PDB Using DBCA 36](#_Toc11138)

[5.5 Create PDB Using PDB SEED 43](#_Toc8224)

[5.6 Clone PDB 43](#_Toc4825)

**[MANAGING USERS & SECURITY 46](#_Toc17061)**

[6.1 Database Users 46](#_Toc28233)

[6.2 Schema 47](#_Toc374)

[6.3 User Privileges 47](#_Toc28695)

[6.4 User Roles 48](#_Toc31874)

[6.5 User Profiles 48](#_Toc9565)

**[MANAGING DATABASE CONNECTIVITY 49](#_Toc23432)**

[7.1 Oracle Net Listener 49](#_Toc7173)

[7.2 Oracle Net Service 49](#_Toc4612)

[7.3 Service Registration 50](#_Toc10320)

[7.4 Naming Methods 50](#_Toc24706)

[7.5 Database Links 51](#_Toc23254)

[7.6 Database Link Users 52](#_Toc1677)

[7.7 Naming Database Links 52](#_Toc26885)

[7.8 Lab 52](#_Toc12220)

**[UNDO & FLASHBACK 53](#_Toc29801)**

[8.1 Fast Recovery Area 53](#_Toc29271)

[8.2 Redo Log 54](#_Toc4440)

[8.3 Archive Log 56](#_Toc13289)

[8.4 Undo 57](#_Toc12203)

[8.5 Flashback 57](#_Toc11119)

[8.6 Restore Point 59](#_Toc28077)

**[BACKUP & RECOVERY 61](#_Toc30800)**

[9.1 Database Backup and Recovery 61](#_Toc26871)

[9.2 RMAN 61](#_Toc19602)

[9.3 P 62](#_Toc29373)

[9.4 p 62](#_Toc30771)

[9.5 P 62](#_Toc32542)

**[DATA GUARD 63](#_Toc26340)**

[10.1 Data Guard 63](#_Toc12974)

[10.2 Prepare Primary Server Database 63](#_Toc7146)

[10.3 Prepare Standby Server Database 66](#_Toc14489)

[10.4 Testing 67](#_Toc24278)

[10.5 Switchover 68](#_Toc9973)

[10.6 p 69](#_Toc20225)

# PENDAHULUAN

## Latar Belakang

Dalam era digital yang semakin berkembang pesat, pengelolaan basis data menjadi krusial bagi perusahaan-perusahaan yang bergantung pada informasi untuk mengelola operasi bisnis mereka. Oracle Database telah lama menjadi salah satu platform database paling andal dan populer di dunia. Sebagai inti dari infrastruktur data bagi banyak organisasi besar dan kecil di seluruh dunia, keberhasilan pengelolaan Oracle Database menjadi kunci bagi kesuksesan operasional dan strategis suatu perusahaan.

"Oracle Database Administration" adalah buku panduan yang dirancang untuk membantu para Database Administrator dalam memahami dan menguasai berbagai aspek administrasi Oracle Database. Buku ini menyediakan fondasi bagi pembaca yang ingin mempelajari lebih lanjut tentang cara mengelola dan mengoptimalkan database Oracle. Pada buku ini, pembaca akan diperkenalkan dengan konsep dasar basis data, arsitektur Oracle Database, dan peran seorang Database Administrator dalam menjaga kesehatan dan kinerja sistem.

Dengan memahami Administrasi Oracle Database yang disajikan dalam Buku ini, diharap pembaca akan siap untuk menjelajahi topik-topik yang lebih mendalam dalam buku ini, termasuk instalasi, konfigurasi, pemeliharaan, dan pemecahan masalah dalam administrasi Oracle Database. Dengan demikian, pembaca dapat membentuk fondasi yang kokoh dalam perjalanan menuju penguasaan administrasi Oracle Database.

## Rumusan Masalah

# PREPARING LAB ENVIRONMENT

## Lab Requirements

Sebelum masuk ke materi, pastikan dulu laptop/komputer teman-teman memenuhi persyaratan yang ada. Agar nantinya ketika melakukan lab tidak ada kendala dalam hardware komputer/laptopnya. Disini kita akan menggunakan VM (Virtual Machine) untuk menginstall oracle databasenya. Berikut ini beberapa requirements/persyaratan untuk hardware laptop/komputer yang diperlukan:

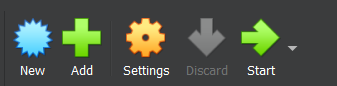
* RAM 4GB atau lebih (8GB Reccomended)
* Disk space 30GB atau lebih
* Processor 2 core atau lebih
* Software virtualbox

## Installation Oracle Linux 8

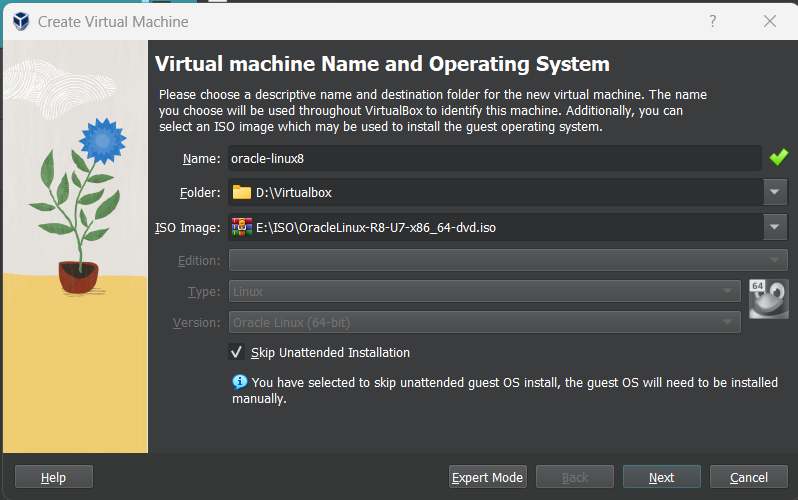
1. Download dan install software virtualbox. Download file iso Oracle Linux 8 pada link berikut:

<https://yum.oracle.com/oracle-linux-isos.html>

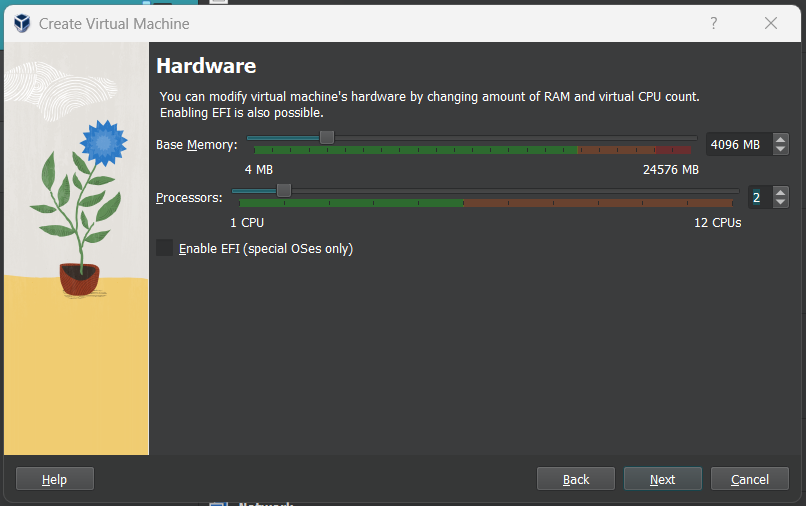
1. Buka virtualbox. Pada menu bagian atas pilih **New**.



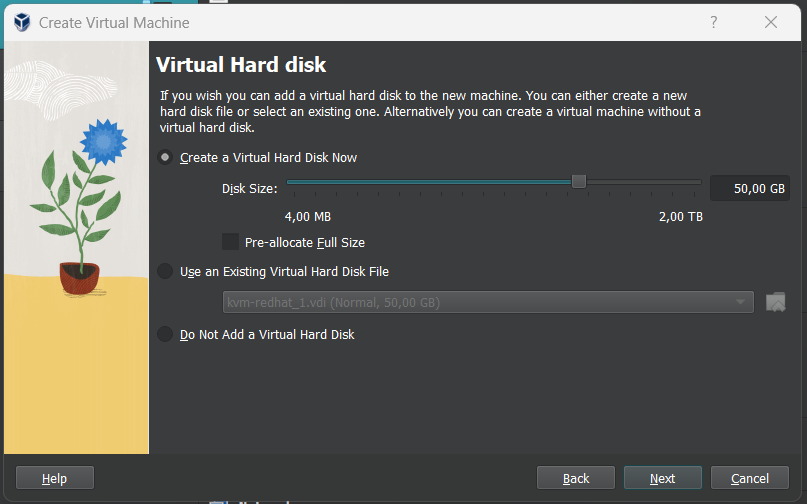
1. Berikan nama untuk vm yang akan dibuat. Masukkan juga image iso oracle linux 8 yang sudah didownload.



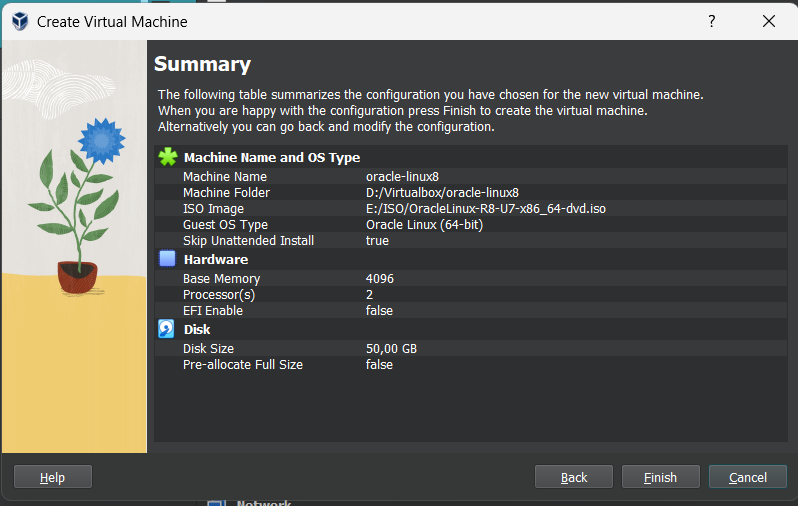
1. Setting memory ke 4 GB dan processor 2 core.



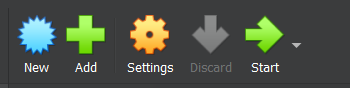
1. Setting disk menjadi 50GB. Agar kita bisa memiliki space yang lebih untuk oracle database nantinya.



1. Jika sudah terakhir tinggal klik finish.



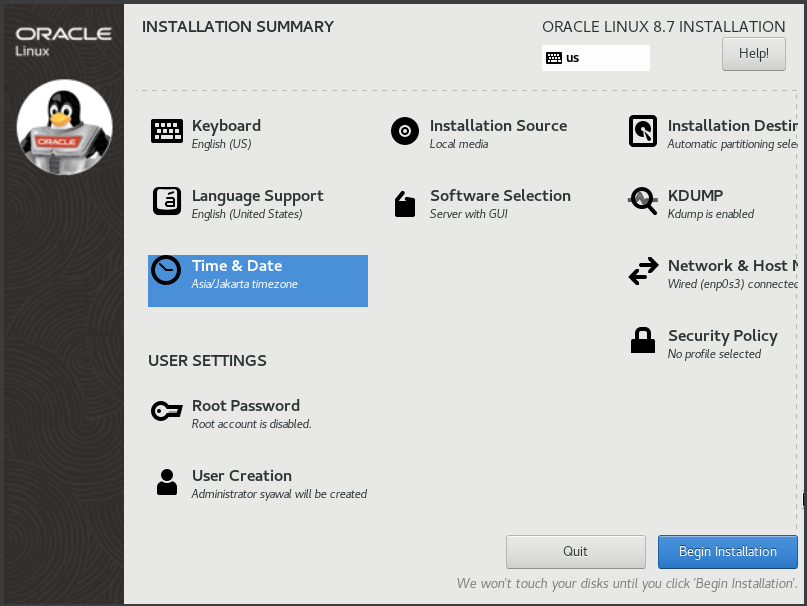
1. Pilih vm yang telah dibuat. Kemudian klik tombol start pada menu bagian atas.



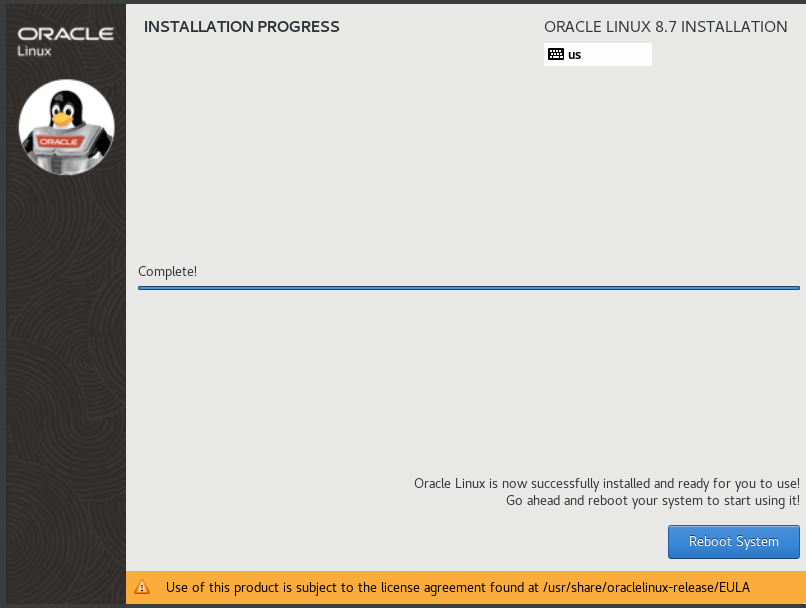
1. Pada bagian pemilihan bahasa biarkan default.



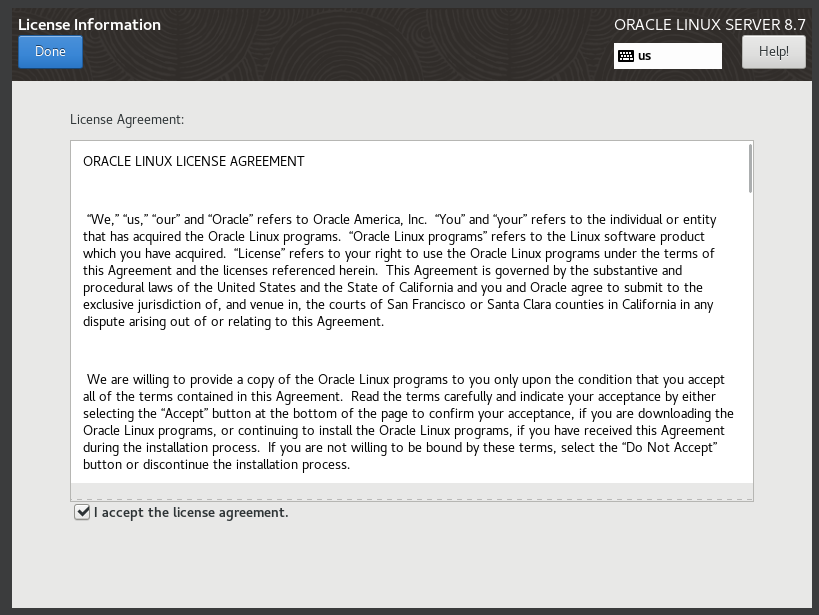
1. Disini jangan lupa setting time & date, user creation dan network & host. Pada bagian user creation jangan lupa centang dibagian **make this user administrator**.



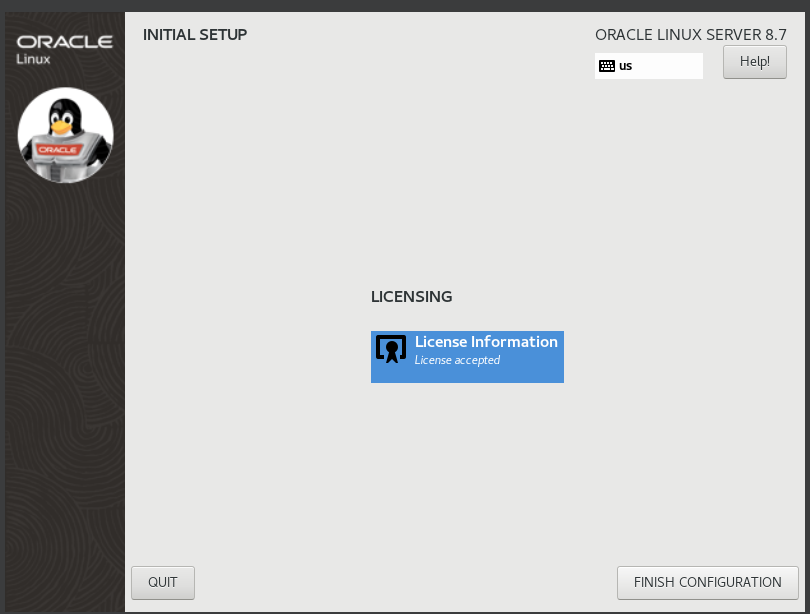
1. Tunggu proses instalasinya, disini biasanya butuh waktu yang cukup lama. Jika sudah selesai klik tombol **reboot system**.



1. Jika ada diminta untuk license agreement, cukup centang dan klik done.

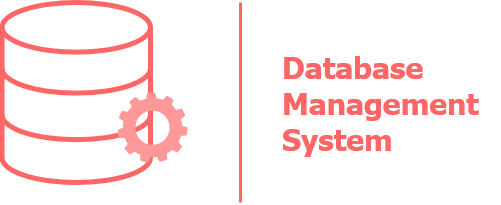


1. Jika sudah klik finish configuration. Instalasi sudah selesai, oracle linux 8 sudah bisa digunakan.



# INTRODUCTION

## Introduction DBMS



Sebelum kita berkenalan dengan Database Management System (DBMS), kita harus memahami dulu nih arti Database sendiri itu apa. Database atau Basis Data merupakan kumpulan data yang terorganisir, yang umumnya disimpan dan diakses secara elektronik dari suatu sistem komputer [wikipedia]. Database ini umumnya berada pada suatu sistem dan dikelola secara otomatis oleh DBMS. Jadi dapat kita ketahui nih, bahwa DBMS merupakan suatu sistem atau perangkat lunak yang dirancang untuk mengelola suatu basis data (database) [wikipedia]. Database sendiri terbagi menjadi 2 jenis, yaitu Relational Database dan Non Relational Database.

Relational Database merupakan tipe basis data yang menggunakan struktur tabel untuk mengelola datanya. Struktur dari relational database terdiri dari baris dan kolom. Sedangkan, Non Relational Database merupakan tipe basis data yang dirancang untuk menyimpan dan mengelola data dalam format yang lebih fleksibel daripada model tabel tradisional yang digunakan dalam relational database. Simpelnya Relational Database menggunakan format SQL, sedangkan non Relational Database menggunakan format JSON atau BSON.

## Introduction Oracle Database

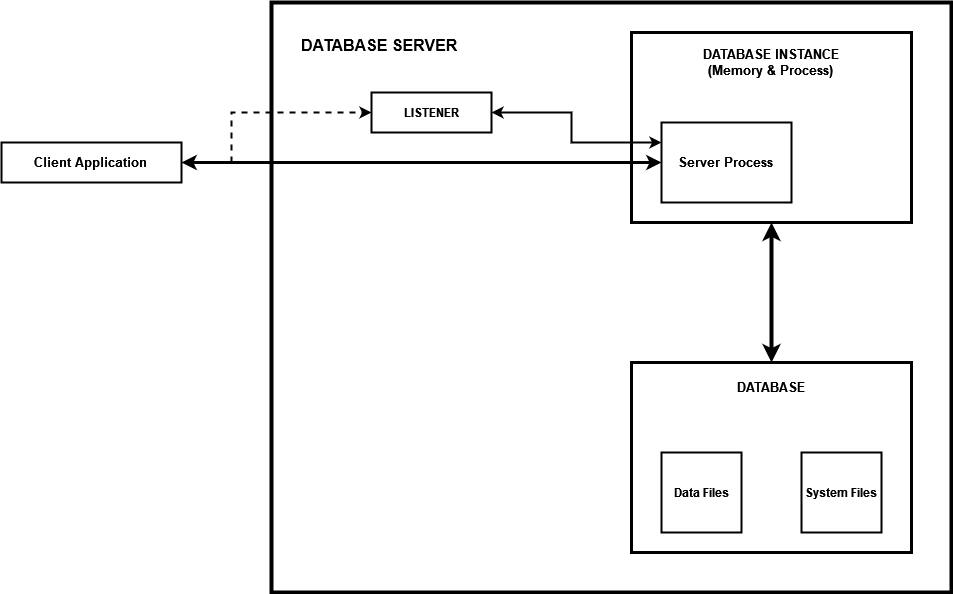


Oracle adalah salah satu dari RDBMS atau sering disebut dengan Relation database management system buatan dari oracle corporation. Oracle database ini sering digunakan untuk membangun sebuah system yang berskala besar, seperti yang ada pada sistem perbankan.  Seperti system RDBMS, oracle ini juga menggunakan SQL atau structure query language sebagai bahasa saat mengoperasikan.

**Tipe Release Oracle Database**

* Long Term Support Release
  + Dapat premier suport 5 tahun dan 3 tahun extended support
  + Recommended untuk production
* Innovation Release
  + Dapat premier support 2 tahun dan tidak ada extended support
  + Recommended untuk belajar fitur terbaru pada lingkungan development

## Oracle Database Architecture

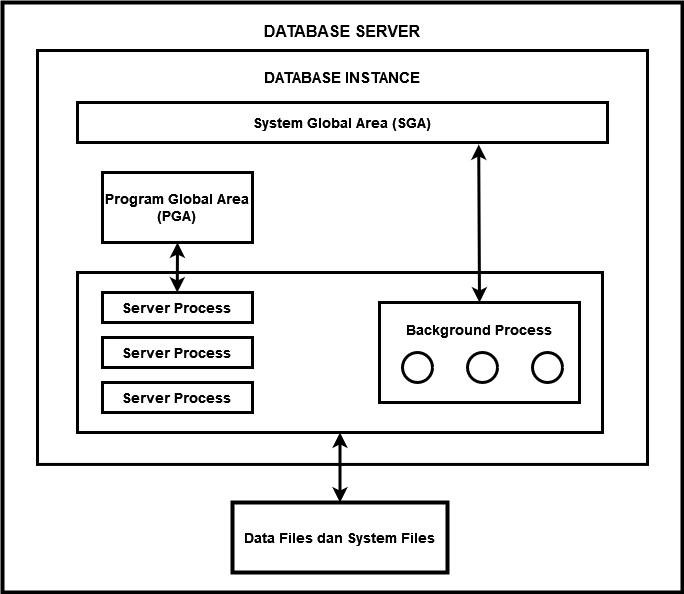


Pada database server oracle terdapat 2 group komponen utama, yaitu

* Database Instance
  + Berada/berjalan pada memory yang dimana memiliki group memory area dan group processes.
    - Memory Area digunakan untuk meyimpan spesific data
    - Processes bertanggung jawab terhadap suatu fungsi (Server processes dan Client Processes)
* Database
  + Berada/berjalan pada penyimpanan atau storage

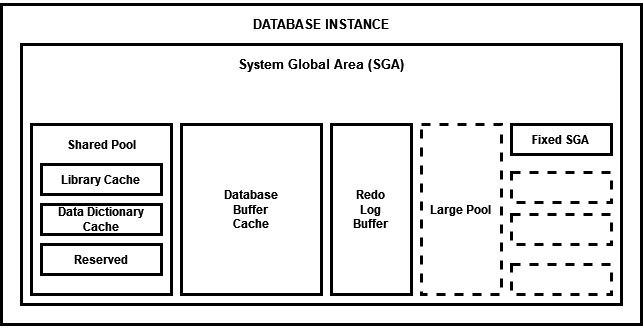
Jika diperhatikan dengan baik, terdapat sebuah komponen bernama listener. Listener ini berfungsi untuk menghandle traffic client yang masuk menuju database. Jadi ketika ada request dari aplikasi client menuju server, process tidak langsung menuju database. tetapi melewati listener terlebih dahulu. Listener bukan merupakan bagian dari process database, namun akan terbuat dengan sendirinya. Setelah dari listener, database instance akan membuat suatu process baru yang disebut server process. Dapat dibilang, semakin banyak request client, semakin banyak pula server process yang akan dibentuk.

### Database Instance Architecture



Pada materi sebelumnya kita sudah mengetahui nih pada database instance terdapat memory area dan processes. Memory sendiri terdiri dari 2 bagian, yaitu SGA dan PGA. PGA sendiri menggunakan memory untuk menjalankan server process, yang dimana setiap process memiliki private areanya. Karena itulah semakin banyak process client, semakin banyak pula memory PGA yang perlu di alokasikan ke client. Jika PGA digunakan untuk client, maka SGA sendiri digunakan untuk bagian database.

### System Global Area Components



* **Shared Pool :** Tempat penyimpanan SQL dan PL/SQL Statements.
* **Database buffer cache:** Menyimpan salinan data blocks read dari data files untuk optimasi phsycal I/O.
* **Large Pool**: (optional) tempat alokasi memory yang besar untuk RMAN (Recovery Manager),buffers for deferred inserts, message buffers used parallel executions, and others.
* **Redo Log buffer**: untuk menyimpan redo log.

# BASIC ADMINISTRATION

## Instalasi Oracle Database 19c

1. Jalankan VM Server Oracle Linux 8, login sebagai user root. Jalankan perintah dibawah untuk menginstall package preinstall Oracle Database 19c.

# dnf install oracle-database-preinstall-19c -y

1. Download file RPM Oracle Database 19c pada link dibawah.

<https://www.oracle.com/cis/database/technologies/oracle19c-linux-downloads.html>

1. Setting password untuk user oracle.

# passwd oracle

1. Setting SELINUX menjadi permissive.

# vim /etc/selinux/config

---edit baris berikut---

SELINUX=permissive

---end---

//jalankan perintah dibawah

# setenforce Permissive

# systemctl stop firewalld

# systemctl disable firewalld

1. Setting environment variable untuk user oracle.

# vim /home/oracle/.bash\_profile

---tambahkan baris perintah dibawah---

export ORACLE\_HOME=/opt/oracle/product/19c/dbhome\_1

PATH=/opt/oracle/product/19c/dbhome\_1/bin:$PATH

---end---

1. Selanjutnya kita lakukan instalasi menggunakan file RPM yang sudah didownload.

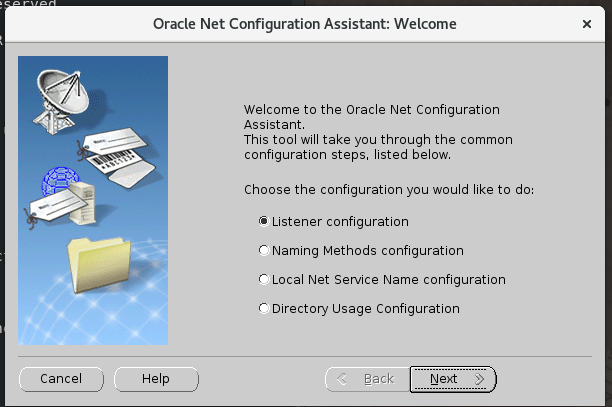
# export CV\_ASSUME\_DISTID=OEL7.6

# dnf localinstall oracle-database-ee-19c-1.0-1.x86\_64.rpm -y

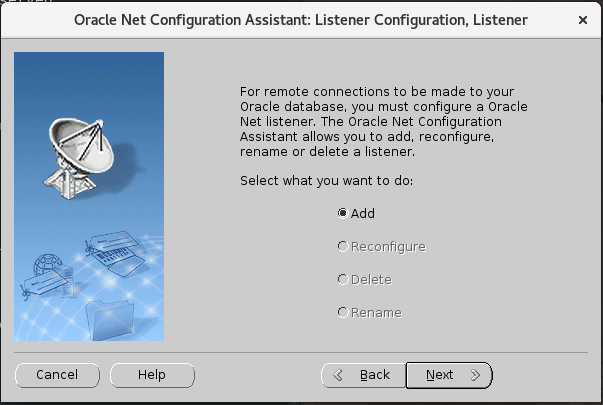
## Create Listener

Pada perkenalan architecture oracle kita sudah berkenalan nih dengan komponen Listener. Listener ini digunakan sebagai jembatan antara request client dengan Database Instance. Sekarang kita akan mencoba untuk membuat listener untuk oracle database kita.

1. Jalankan perintah “netca” pada terminal linux. Nantinya akan muncul window seperti gambar dibawah. Karena kita akan membuat listener, pilih **Listener configuration**.



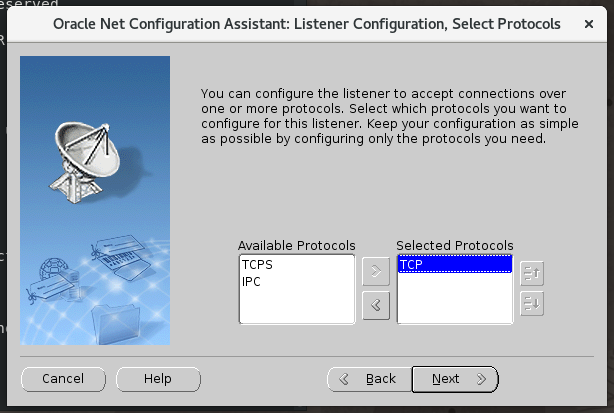
1. Pilih options **Add**.



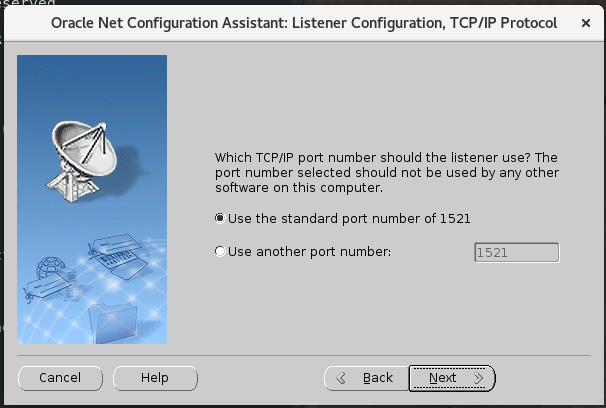
1. Berikan nama untuk Listener yang akan dibuat.



1. Disini kita akan memilih protocol **TCP**.



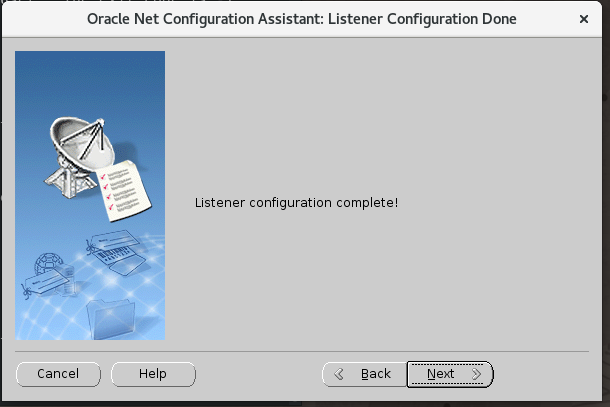
1. Untuk port listener nya kita biarkan default. Bagi kalian yang mau dibikin custom juga bisa, dengan memilih options **Use antoher port number**.



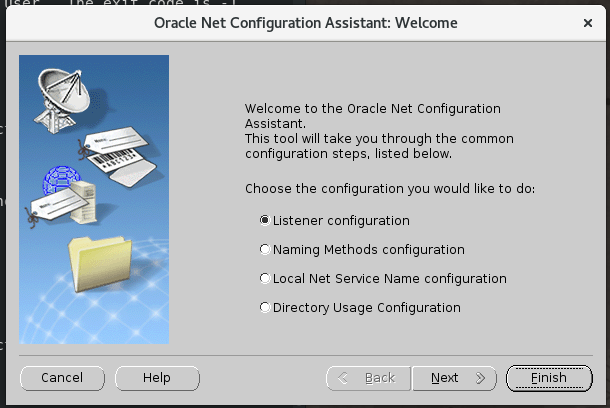
1. Karena kita tidak akan melakukan konfigurasi lain untuk listener, pilih aja **No**.



1. Jika sudah complete klik Next.



1. Disini kita finish saja, karena kita sudah tidak akan membuat listener lagi.

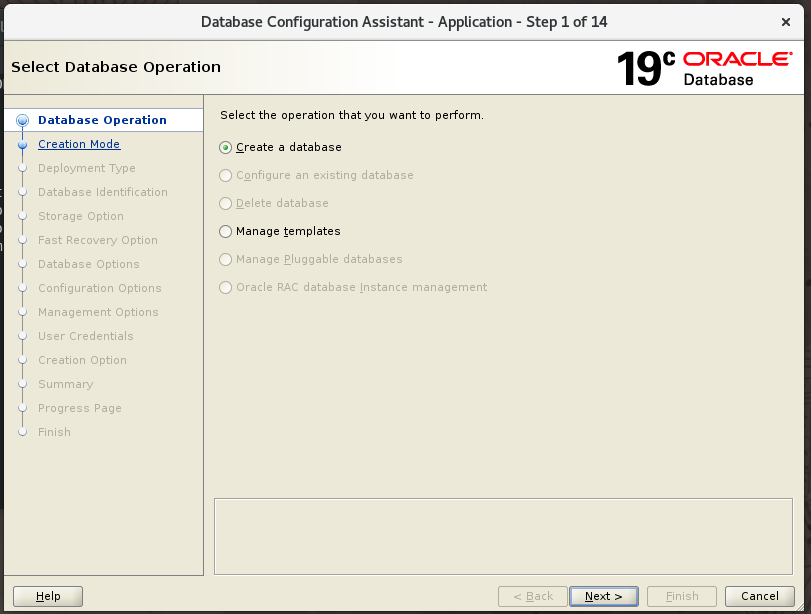


## Create Database

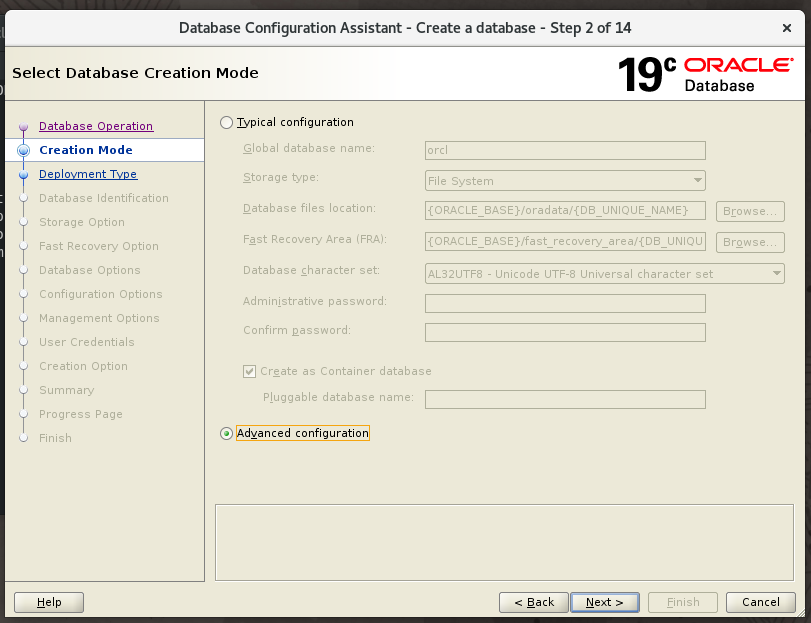
1. Jalankan perintah dibawah.

$ dbca

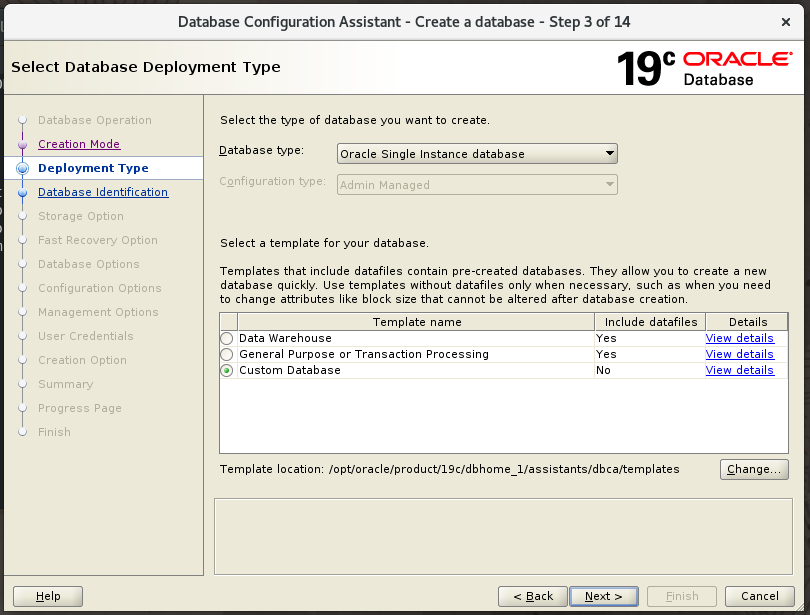
1. Nantinya akan muncul windows seperti gambar dibawah. Pada bagian ini pilih yang **Create a database**.



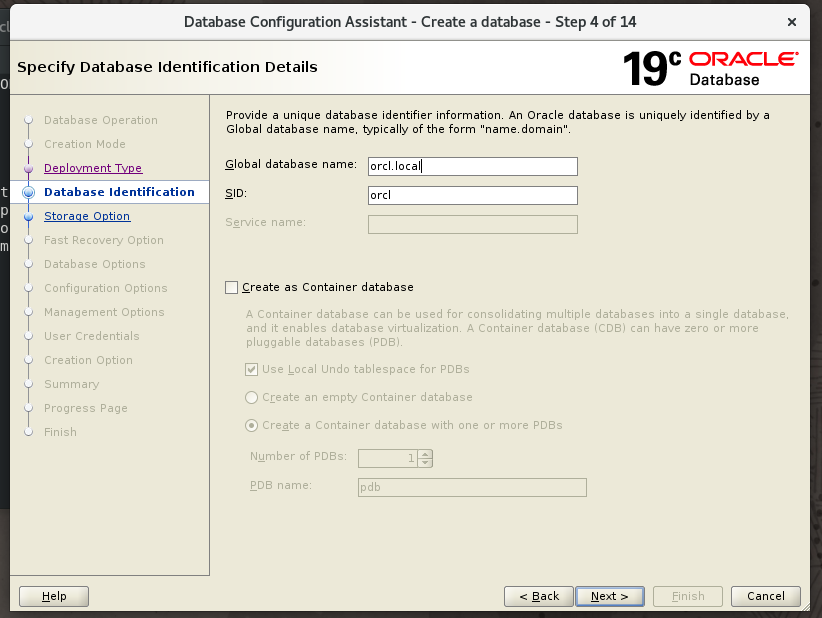
1. Pada tab **Creation Mode** pilih yang **Advanced configuration**.



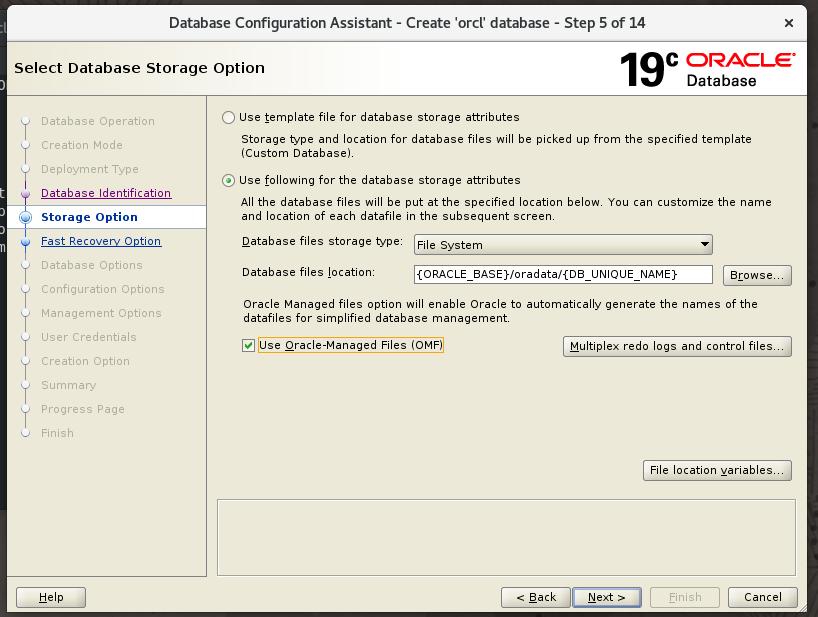
1. Pada bagian **Deployment Type** untuk **Database Type** biarkan default ke **Oracle Single Instance database**. Dibagian **Template name** kalian bisa tentukan sendiri, disini saya memilih yang **Custom Database**.



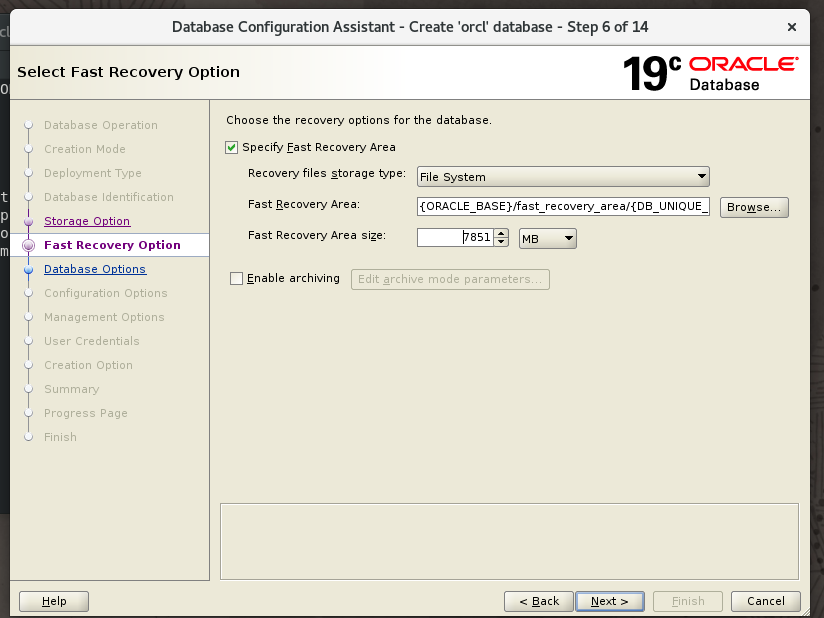
1. Dibagian **Database Identification**, isikan **Nama Database** dan juga **SID** nya. Ini digunakan sebagai identifier untuk koneksi Database nantinya. Uncheck pada bagian **Create as Container database**. Disini kita tidak akan membuat container database dulu.



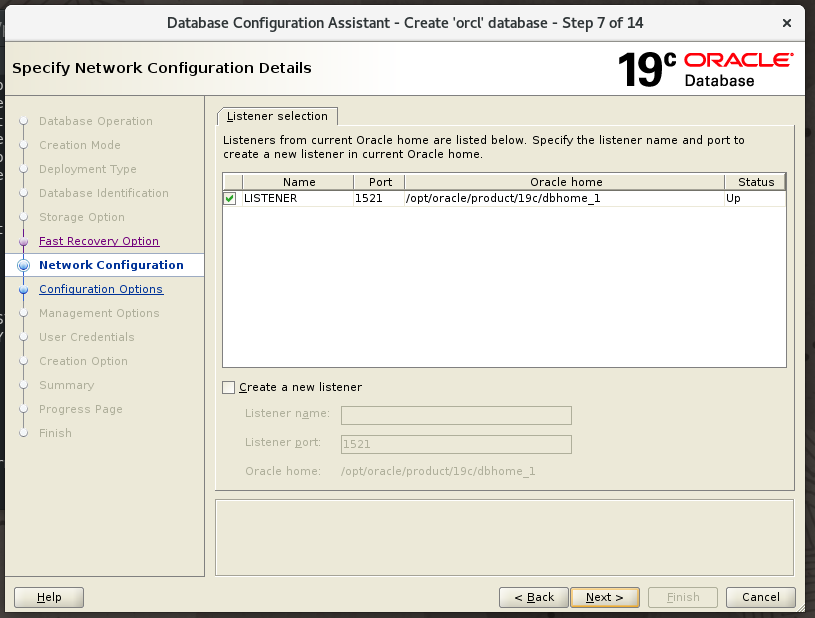
1. Pada bagian **Storage Option**, pilih option **Use following for the database storage attributes**. Jangan lupa untuk centang bagian **User Oracle-Managed File (OMF)**.



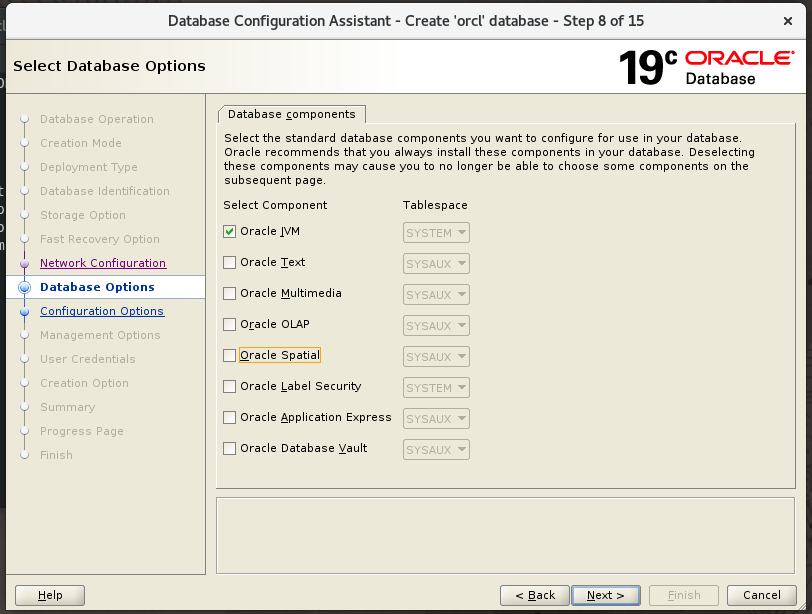
1. Pada bagian **Fast Recovery Option**, centang di **Specifiy Fast Recovery Area**. Untuk bagian **Fast Recovery Area size** bisa disesuaikan.



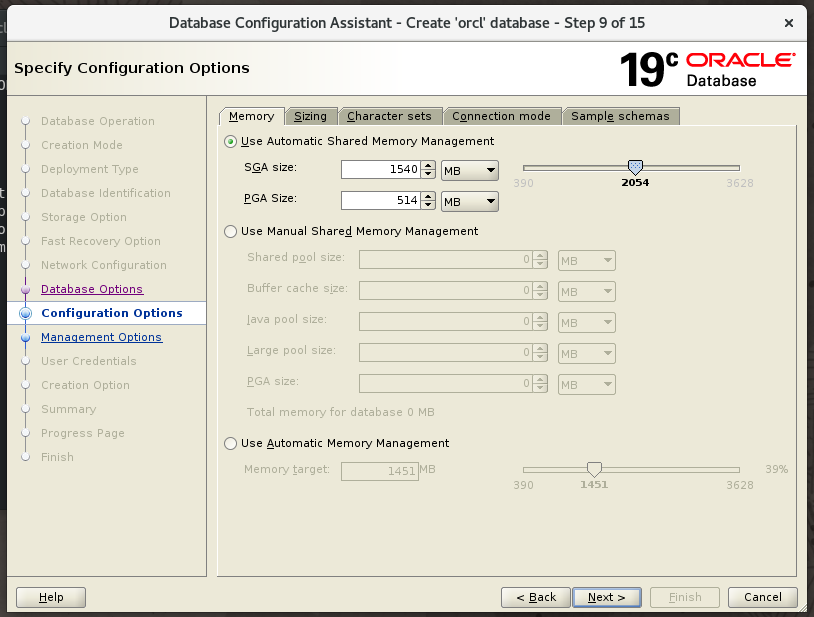
1. Pada bagian **Network Configuration**, disini kita akan menggunakan Listener yang telah kita buat sebelumnya. Cukup dicentang pada bagian Listenernya. Selain itu, disini kita juga dapat membuat listener juga.



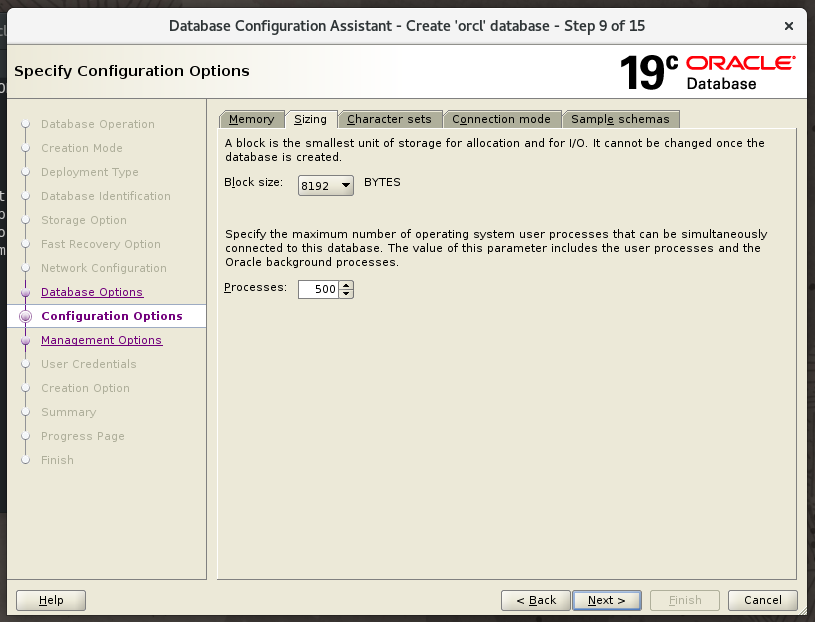
1. Pada bagian **Database Options** cukup pilih yang **Oracle lVM**.



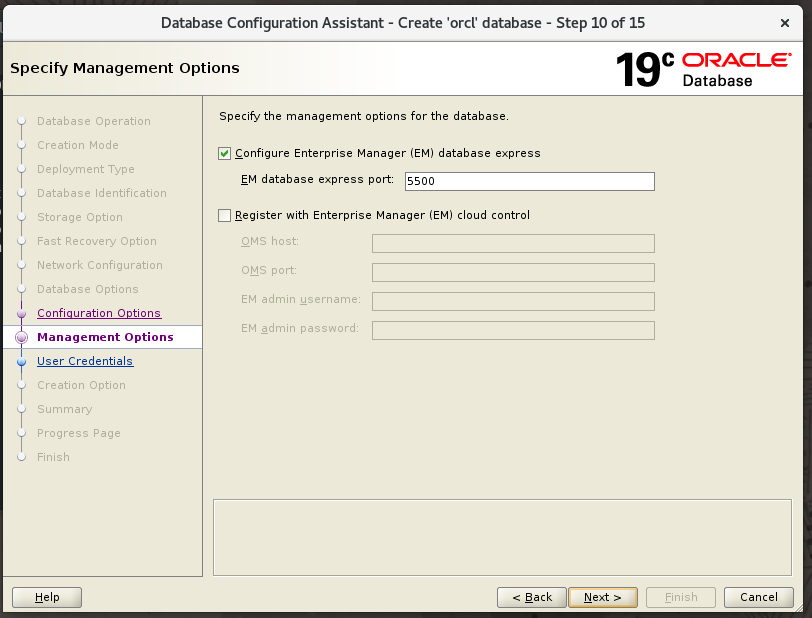
1. Pada bagian **Configuration Options**, di tab memory pilih yang **Use Automatic Shared Memory Management**.



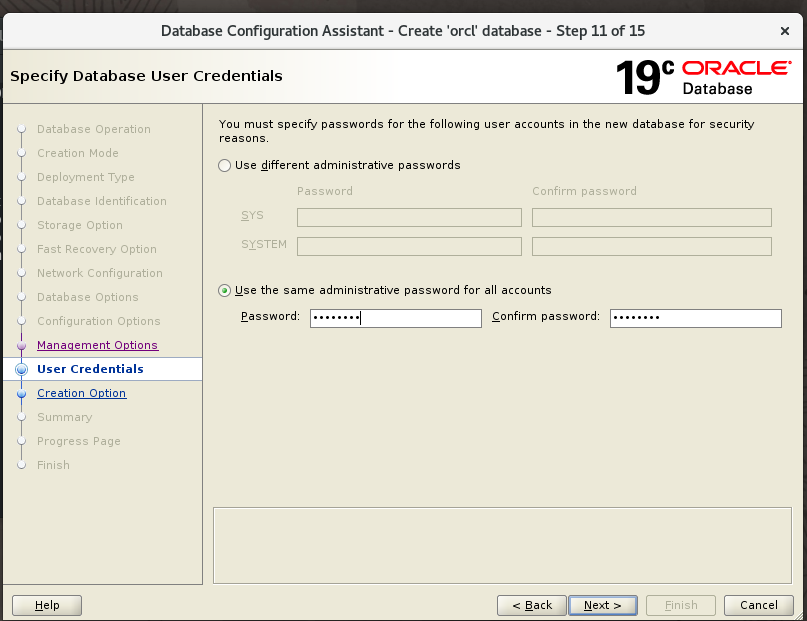
1. Pergi ke menut tab **Sizing** dan ubah **processes** menjadi 500. Bisa juga disesuaikan sesuai kebutuhan untuk berapa banyak processes yang di inginkan.



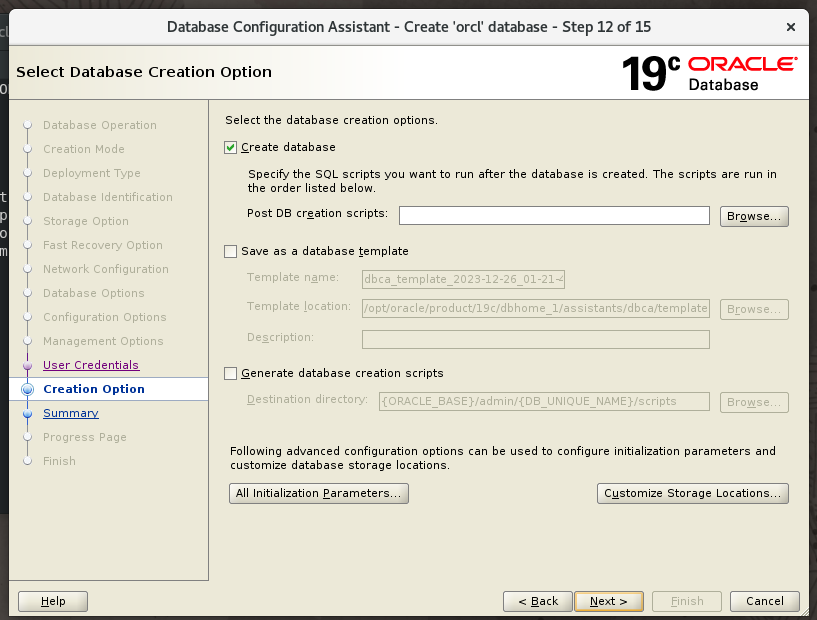
1. Pada bagian **Management Options**, centang pada **Configure Enterprise Manager (EM) database express**. Biarkan port nya default.



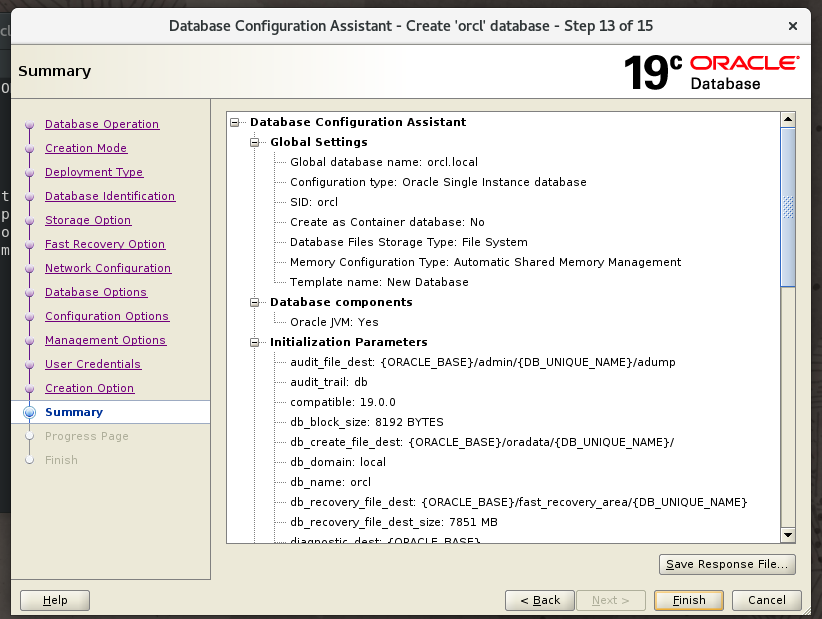
1. Pada bagian **User Credential** bisa disesuaikan. Jika ingin user SYS dan SYSTEM memiliki password yang berbeda pilih **Use different administrative passwords**. Sedangkan, jika ingin memiliki password yang sama bisa pilih **Use the same administrative password for all accounts**.



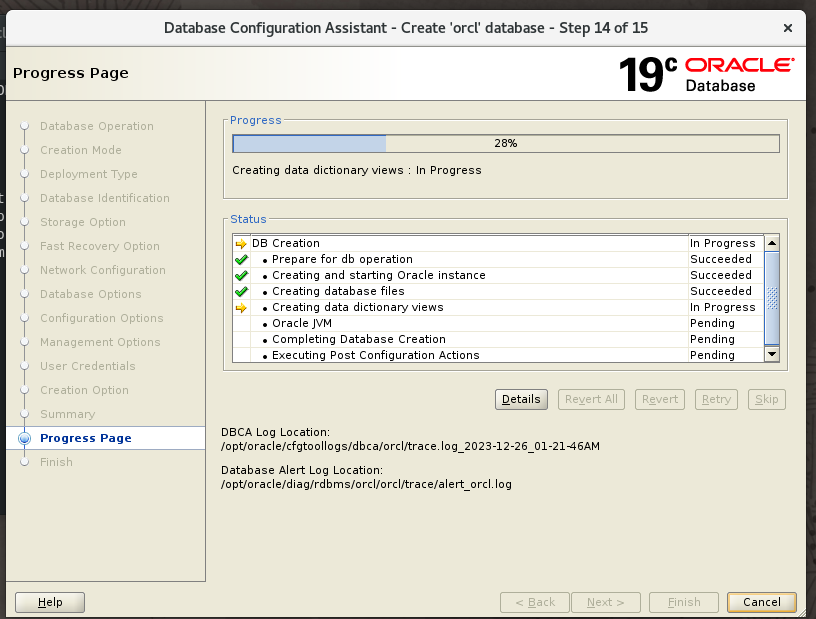
1. Pada bagian **Creation Option** kita pilih **Create database**, karena kita ingin membuat database nya langsung.



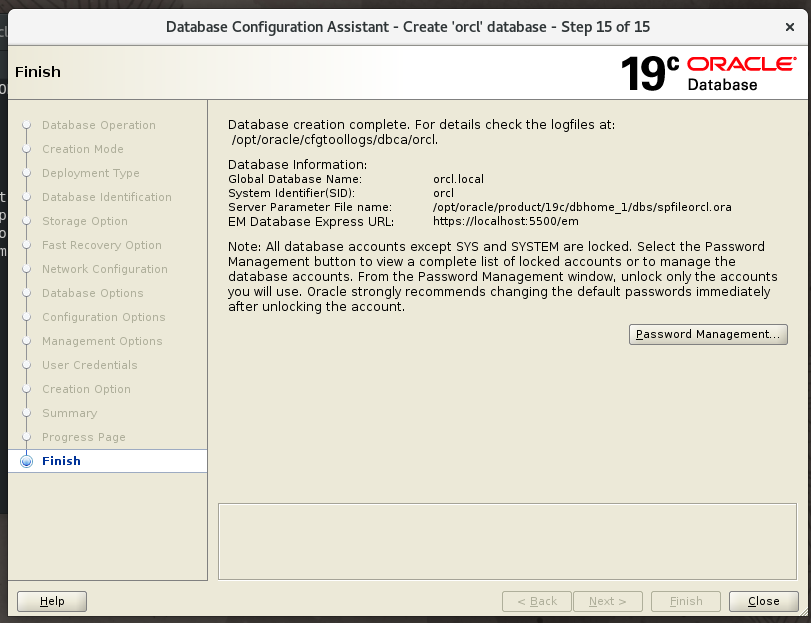
1. Pada bagian S**ummary** kita lihat lagi konfigurasi kita, jika dirasa sudah benar langsung klik **Finish**.



1. Tunggu proses createnya selesai.



1. Jika sudah selesai bisa langsung klik **Close**.



## Connect to Database

### Using SQLPLUS

1. Masuk ke database server. Kemudian jalankan perintah sqlplus.

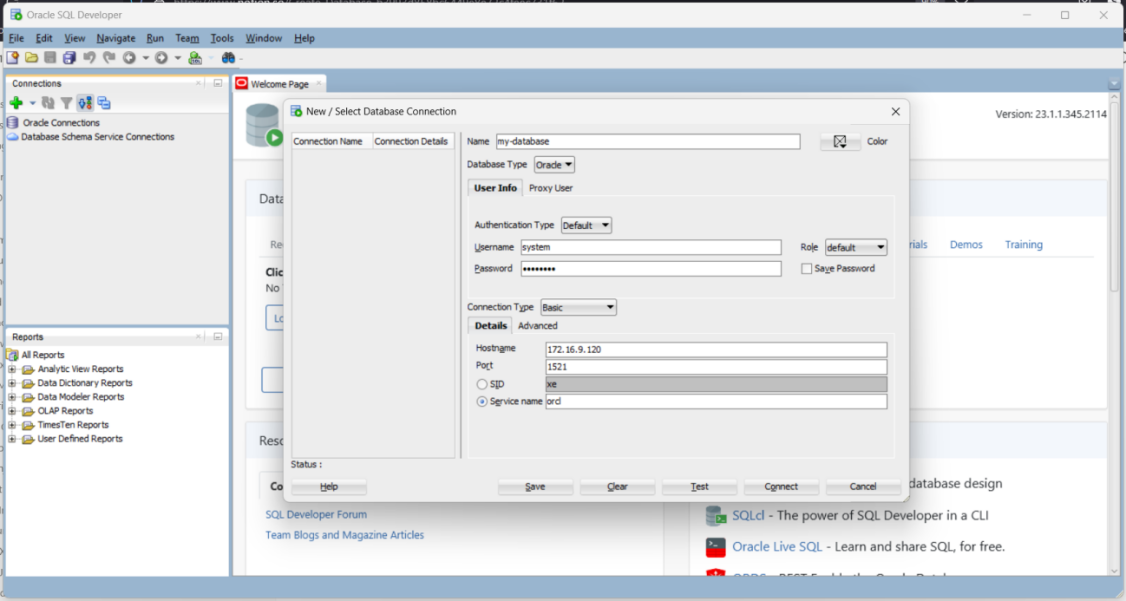
$ sqlplus / as sysdba

### Using SQL Developer

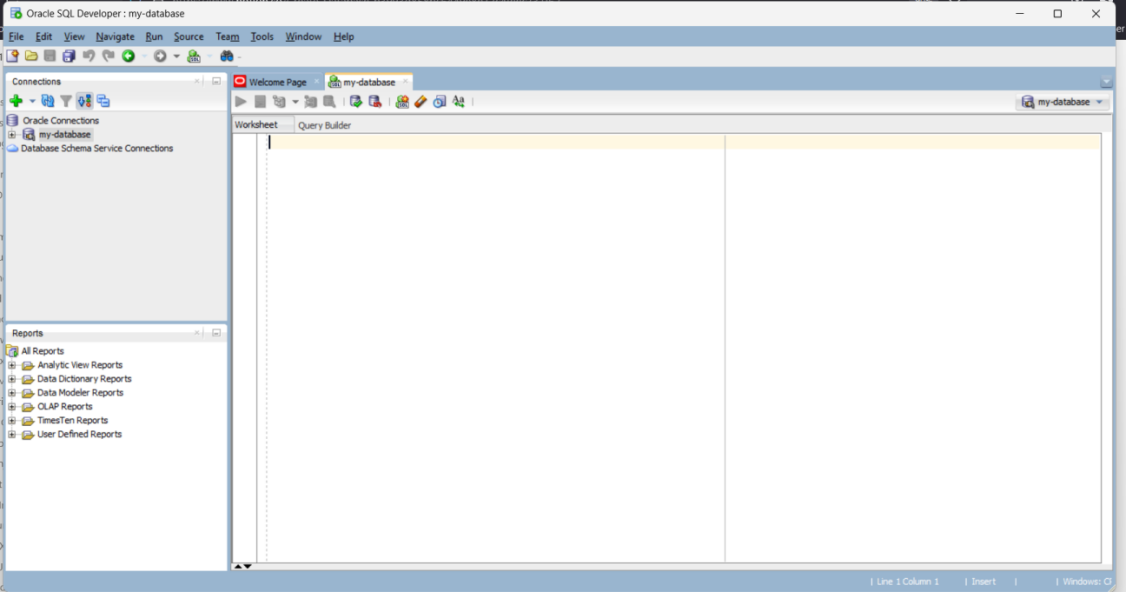
1. Download SQL developer pada link berikut.

<https://www.oracle.com/database/sqldeveloper/technologies/download/>

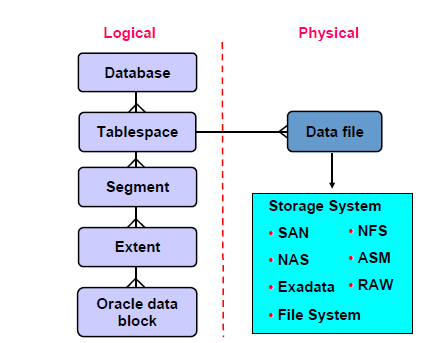
1. Extract file SQL Developer. Masuk ke folder hasil extract dan klik 2 kali pada file sqldeveloper.exe.
2. Buat koneksi baru ke database oracle. Beri nama untuk koneksi databasenya. Masukkan username dan juga password database. Pada bagian hostname isikan dengan IP database. Kemudian pada bagian Service name sesuaikan juga dengan nama service databasenya.



1. Jika berhasil maka hasilnya akan seperti ini. Kita bisa menjalankan query apapun untuk databasenya melalui SQL Developer ini.



## Tablespace



Tablespace adalah unit penyimpanan logis dalam database yang digunakan untuk menyimpan objek database. Tablespace biasanya terdiri dari beberapa segment. Segment terdiri dari beberapa extent. Extent adalah kumpulan data block. Segment digunakan untuk menyimpan objek database. Yang termasuk segment adalah table, index, cluster, rollback (undo), lobsegment, table partition, index partition, lob partition, temporary segment, dll.

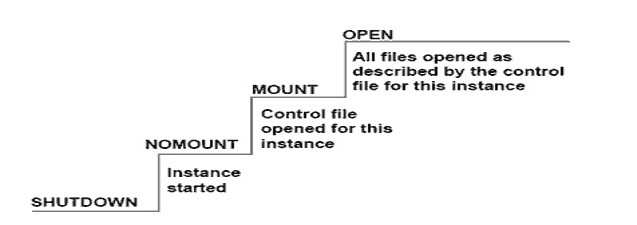
Tablespace terdiri dari beberapa tipe, yaitu:

* Temporary
  + Tablespace yang digunakan untuk menyimpan data sementara selama proses query atau transaksi.
* Permanent
  + Tablespace yang digunakan untuk menyimpan objek-objek database yang permanen, seperti tabel, indeks, dan sequence.
* Undo
  + Tablespace yang digunakan untuk menyimpan data undo. Data undo adalah data yang digunakan untuk mengembalikan database ke keadaan sebelumnya jika terjadi kesalahan.

## Start and Shutdown Instance

Pada materi architecture oracle kita telah mempelajari salah satu komponen utama pada oracle database, yaitu instance. Sekarang kita akan belajar administrasi dasar untuk Instance, yaitu **startup**, **shutdown** dan **alter**. Secara umum proses startup/menghidupkan instance terbagi menjadi berikut:

* Shutdown: Instance dan Background process mati. Memori belum dialokasikan
* Nomount: Instance dan Background process running. Memory dialokasikan
* Mount: Instance membaca control file, namun belum membaca datafile. Control file berisi konfigurasi database.
* Open: Instance sudah membaca data file dan Database siap diakses.



**Perintah untuk startup Instance**:

**>** startup

**>** startup open

**>** startup nomount

**>** startup mount

Perintah “startup” by default sama seperti perintah “startup open”. Jadi ketika sudah menjalankan perintah “startup” kita tidak perlu lagi menjalankan perintah “startup open” untuk membaca datafile.

**Perintah untuk shutdown Instance**:

**>** shutdown normal

**>** shutdown transactional

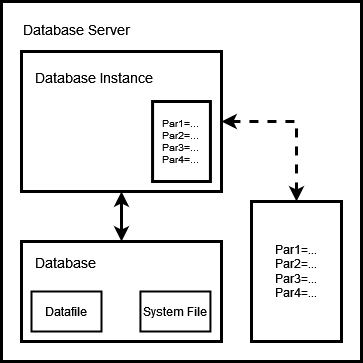
**>** shutdown immediate

**>** shutdown abort

Berikut perbandingan proses shutdown normal (N), transactional (T), immediate (I) dan abort (A):

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| Proses | Shutdown | | | |
| A | I | T | N |
| Memperbolehkan koneksi baru | No | No | No | No |
| Menunggu sampai semua session disconnect | No | No | No | Yes |
| Menunggu sampai semua transkasi selesai | No | No | Yes | Yes |
| Melakukan checkpoint dan closefile | No | Yes | Yes | Yes |

## Database Initialization Parameters



Pada oracle terdapat istilah initialization parameters. Initialization parameter files pada Oracle adalah file konfigurasi yang digunakan untuk menentukan pengaturan awal (initialization parameters) untuk instance database Oracle. File ini menyimpan informasi tentang cara database harus dikonfigurasi saat dijalankan.

Terdapat 2 initialization parameter files, yaitu:

* Pfile
  + File teks sederhana yang berisi daftar parameter dan nilainya
  + Parameter ini dapat diedit secara manual menggunakan editor teks.
  + Biasanya dinamai init(SID).ora, dimana "SID" adalah System Identifier untuk instance database tertentu
* SPfile
  + Versi biner dari Parameter File yang tidak dapat diedit secara langsung menggunakan editor teks.
  + Biasanya dinamai spfile(SID).ora

**Create Pfile from SPfile**

Pertama kita cek terlebih dahulu directory default SPfile.

**>** show parameter spfile

NAME TYPE VALUE

------------------------------------ ----------- ------------------------------

spfile string /opt/oracle/product/19c/dbhome

\_1/dbs/spfile(SID).ora

Membuat pfile menggunakan spfile.

**>** create pfile='/home/oracle/pfile\_test.conf' from spfile;

Kalian bisa melihat isi dari pfile yang telah dibuat menggunakan perintah dibawah.

**>** less /home/oracle/pfile\_test.conf

Kemudian kita shutdown instancenya.

**>** shutdown immediate

Database closed.

Database dismounted.

ORACLE instance shut down.

Selanjutnya kita rename file default spfile

$ cd $ORACLE\_HOME/dbs

$ mv spfile(SID).ora ori\_spfile(SID).ora

Sekarang kita coba jalankan kembali instance, harusnya instance akan mengalami error karena tidak dapat menemukan initialization parameter filenya.

**>** startup

Nah, sekarang kita akan mencoba untuk memindahkan pfile yang telah kita buat ke directory initialization parameter filenya.

$ cp /home/oracle/pfile\_test.conf $ORACLE\_HOME/dbs/init(SID).ora

Kemudian kita jalankan kembali instance kita, harusnya nanti sudah bisa jalan.

$ sqlplus / as sysdba

**>** startup

Ketika kita show kembali initialization parameternya, maka dibagian value akan menjadi kosong.

**>** show parameter spfile

**Recreate SPfile from Pfile**

Sebelumnya kita telah membuat Pfile menggunakan SPfile, sekarang kita akan mencoba recreate SPfile nya menggunakan Pfile.

$ sqlplus / as sysdba

**>** create spfile='/opt/oracle/product/19c/dbhome\_1/dbs/spfile(SID).ora' FROM PFILE='/opt/oracle/product/19c/dbhome\_1/dbs/init(SID).ora';

Kita lihat semua initialization parameter filenya

$ls $ORACLE\_HOME/dbs/\*ora

/opt/oracle/product/19c/dbhome\_1/dbs/init(SID).ora //pfile we create

/opt/oracle/product/19c/dbhome\_1/dbs/init.ora //example pfile

/opt/oracle/product/19c/dbhome\_1/dbs/ori\_spfile(SID).ora //original spfile

/opt/oracle/product/19c/dbhome\_1/dbs/spfile(SID).ora //new spfile from pfile

Selanjutnya kita hidup dan matikan instancenya

**>** shutdown immediate

Database closed.

Database dismounted.

ORACLE instance shut down.

**>** startup

ORACLE instance started.

Selanjutnya kita jalankan kembali show parameter spfile untuk melihat apakah instance sudah jalan kembali menggunakan SPfile atau belum.

**>** show parameter spfile

NAME TYPE VALUE

------------------------------------ ----------- ------------------------------

spfile string /opt/oracle/product/19c/dbhome

\_1/dbs/spfile(SID).ora

## Data Dictionary

Pada umumnya saat kita ingin melihat informasi yang ada pada database, kita akan menjalankan query tertentu. Di oracle sendiri terdapat 2 tipe Query yang sering digunakan untuk melihat informasi pada database, yaitu Dynamic Performance Views dan Data Dictionary Views.

### Dynamic Performance Views

Dynamic Performance Views adalah tampilan sistem yang menyediakan informasi dinamis tentang kinerja dan status internal dari sebuah instance Oracle Database. Tampilan ini memberikan visibilitas langsung ke dalam kegiatan operasional dan performa database. Beberapa query-nya mungkin tersedia bahkan ketika database dalam status nomount dan mount. Dynamic Performance Views biasanya ditandai dengan query “V$”.

Contoh Query Dynamic Performace Views:

**>** SELECT \* FROM V$MEMORY\_DYNAMIC\_COMPONENTS ;

**>** SELECT \* FROM V$SESSION;

**>** SELECT LOG\_MODE,OPEN\_MODE,DATABASE\_ROLE FROM V$DATABASE;

### Data Dictionary Views

Data Dictionary Views adalah kumpulan tabel dan tampilan sistem yang menyimpan metadata atau informasi mengenai struktur dan objek database dalam Oracle Database. Data Dictionary menyediakan informasi penting tentang objek-objek database, hak akses, pengguna, serta konfigurasi dan statistik database.

Tipe - Tipe Data Dictionary Views:

* **DBA\_\***
  + Menyediakan informasi penuh tentang semua objek yang ada didalam database.
  + Hanya bisa diakses oleh DBA.
  + Contoh: DBA\_TABLES, DBA\_TAB\_COLUMNS, DBA\_USERS
* **ALL\_\***
  + Menyediakan informasi tentang objek-objek yang dapat diakses oleh pengguna saat ini. Termasuk objek yang dimiliki oleh pengguna dan objek yang diakses melalui privileges dan roles.
  + Contoh: ALL\_TABLES, ALL\_TAB\_COLUMNS
* **USER\_\***
  + Menyediakan informasi tentang objek-objek (tabel, kolom, view) yang dimiliki oleh pengguna.
  + Contoh: USER\_TABLES, USER\_VIEWS

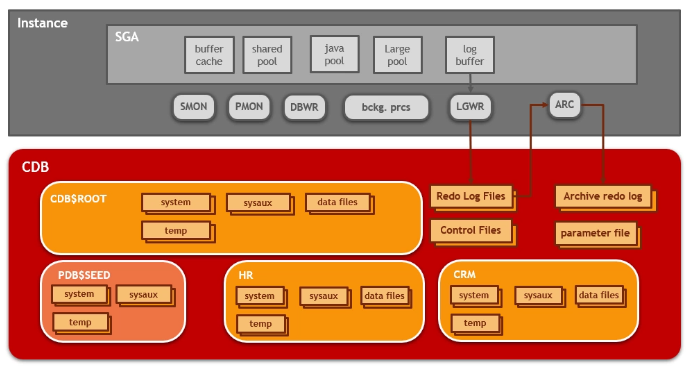
Contoh Query Data Dictionary Views:

**>** SELECT USERNAME, ACCOUNT\_STATUS FROM DBA\_USERS;

**>** SELECT TABLE\_NAME, TABLESPACE\_NAME FROM USER\_TABLE;

# FUNDAMENTAL ORACLE DATABASE MULTITENANCY

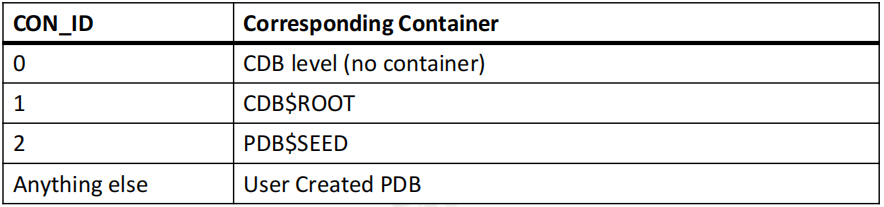
## CDB & PDB



CDB (Container Database) merupakan container yang dimana bisa menampung banyak database. Database yang ada di CDB di kenal dengan nama PDB. CDB memiliki 1 CDB$ROOT container (hanya 1). CDB$ROOT merupakan bagian utama dari CDB yang berisi informasi mengenai container dalam CDB dan menyediakan datafile dan metadata (data dictionary, system users).

PDB (Pluggable Database), pada dasarnya PDB merupakan database biasa. Bedanya setiap PDB terisolasi satu sama yang lain, jadi bisa dibilang setiap PDB tidak dapat mengakses satu sama lainnya. Pada PDB terdapat istilah PDB$SEED. PDB$SEED ini digunakan sebagai template untuk membuat sebuah PDB baru.

## Container Identification

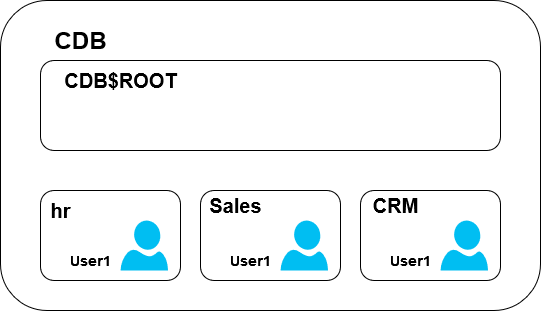


Pada konteks Oracle Multitenant (CDB), terdapat identifier yang digunakan untuk identifikasi CDB dan PDB secara unik. Identifier ini disebut CON\_ID dan GUID. CON\_ID atau Container ID merupakan angka unik yang digunakan untuk mengidentifikasi CDB, CDB$ROOT dan setiap PDB yang ada didalam CDB. CON\_ID bisa dilihat menggunakan perintah “select con\_id, name from v$containers;”.

GUID merupakan 16 string karakter unik secara global yang digunakan untuk identifikasi CDB dan PDB. Untuk melihat GUID dapat menggunakan perintah “select name,from v$containers;”. GUID juga dapat terlihat sebagai identifier untuk menyimpan datafile setiap CDB dan PDB.

## Local Users and Common Users

### Local Users



Local users terletak pada setiap PDB (pluggable database) dan hanya dapat diakses melalui PDB tersebut. Memiliki schema dan objek yang hanya bisa diakses melalui PDB.

### Common Users

### common users.drawio

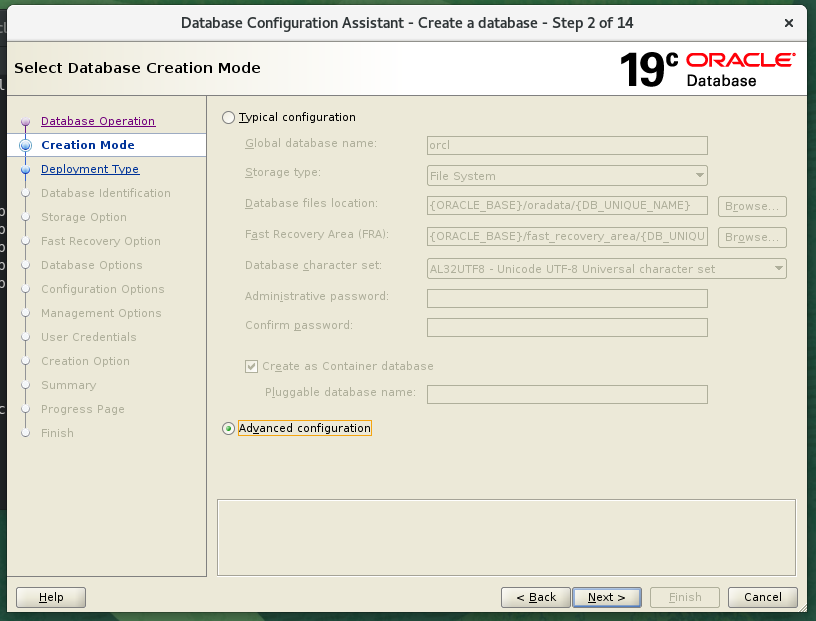
Common users terletak ditingkat CDB dan dapar terhubung ke semua PDB yang ada didalam CDB tersebut. Memiliki schema dan objek yang dapat diakses bersama antara PDB dan CDB, contohnya user System.

## Create CDB dan PDB Using DBCA

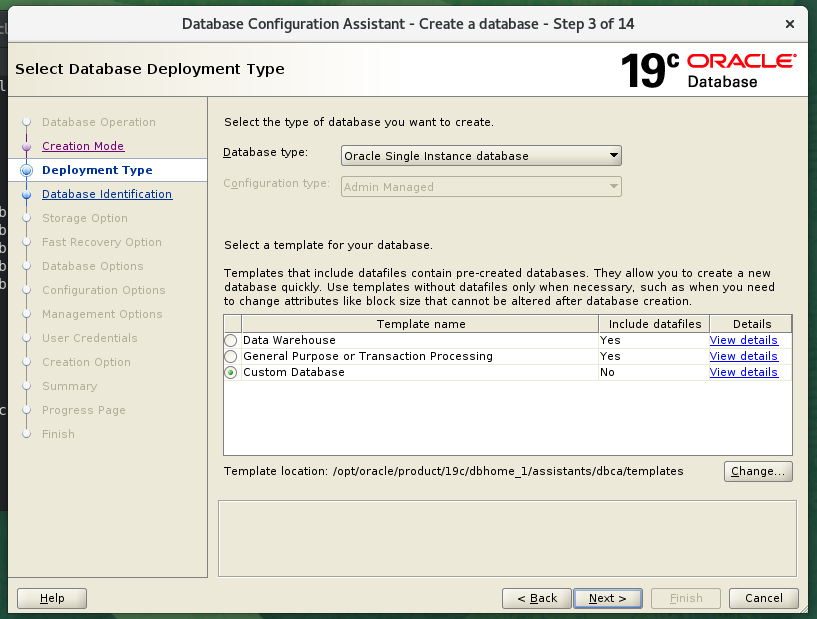
1. Pada materi sebelumnya kita telah belajar membuat Database yang non-pluggable menggunakan DBCA. Sekarang kita akan mencoba untuk membuat container database dan pluggable database menggunakan DBCA. Jalankan perintah “dbca” pada terminal linux. Ketika sudah muncul window seperti gambar dibawah, pilih optiona **create a database**.



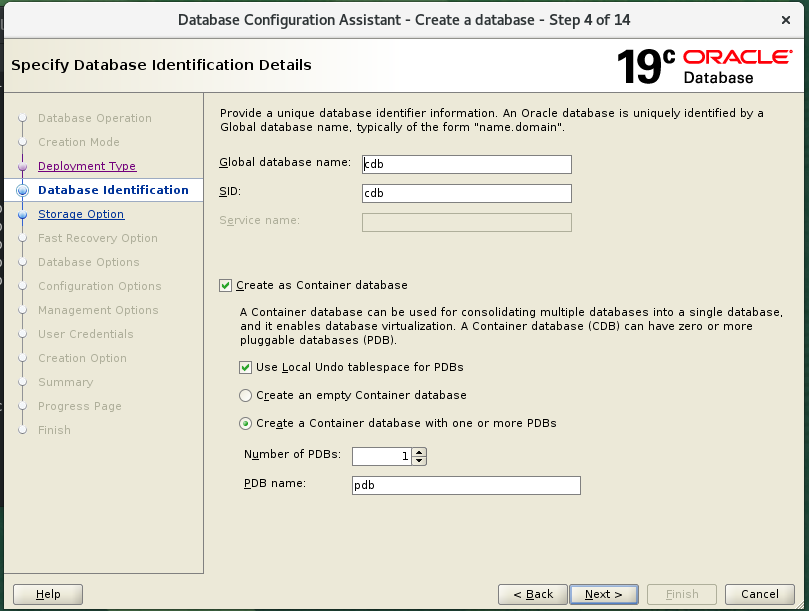
1. Pada bagian **Creation Mode** pilih yang **Advanced configuration**.



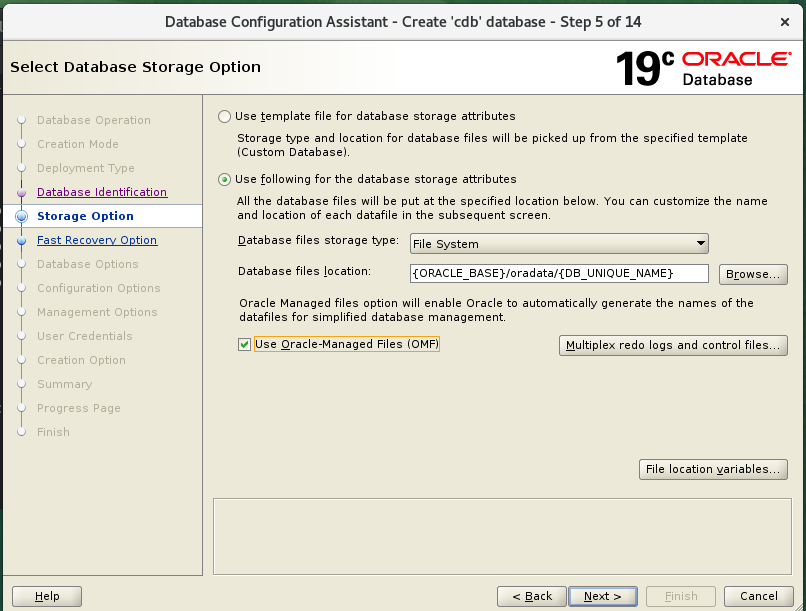
1. Pada bagian **Deployment Type**, dibagian **Database type** biarkan **default** dan bagian **template** kita pilih yang **Custom Database**.



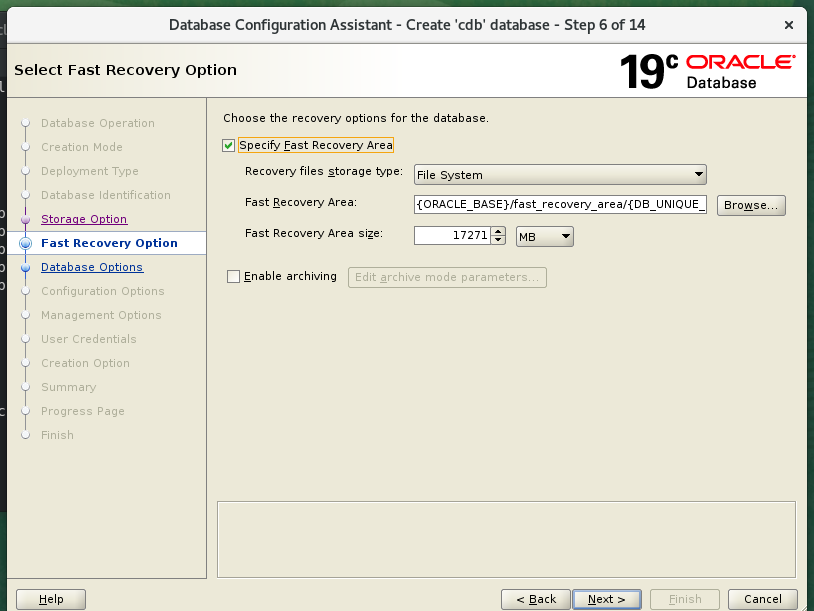
1. Pada bagian **Database Identification** kita set **Global database name** dan **SID** nya menjadi cdb, untuk membedakan database biasa yang telah kita buat sebelumnya. Kemudian centang pada Creaet as Container database, karena kita akan membuat databasenya sebagai CDB dan PDB.



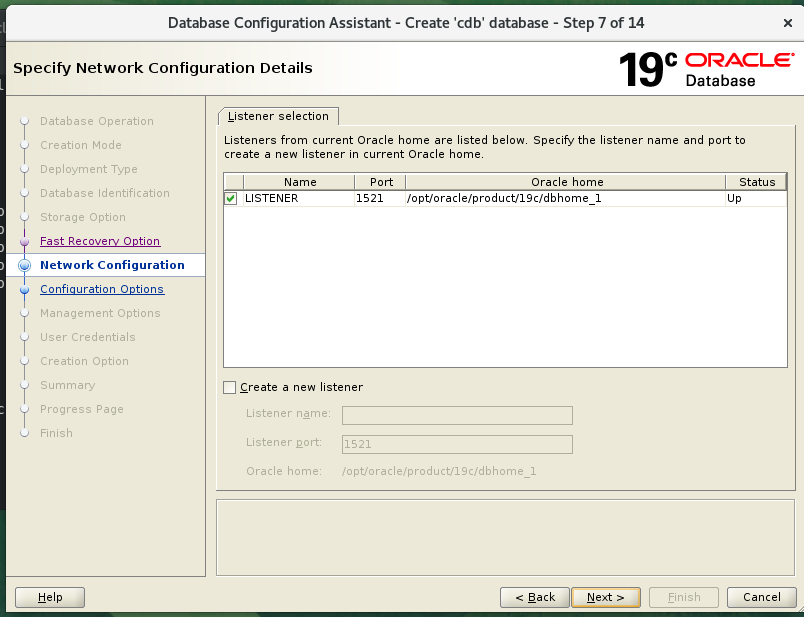
1. Pada bagian **Storage Option** pilih **Use following for the database storage attributes**. Centang juga bagian **Use Oracle-Managed Files (OMF)**.



1. Pada bagian **Fast Recovery Option**, pilih yang **Specify Fast Recovery Area** dan tentukan **Fast Recovery Area Size**, disini saya biarkan default.



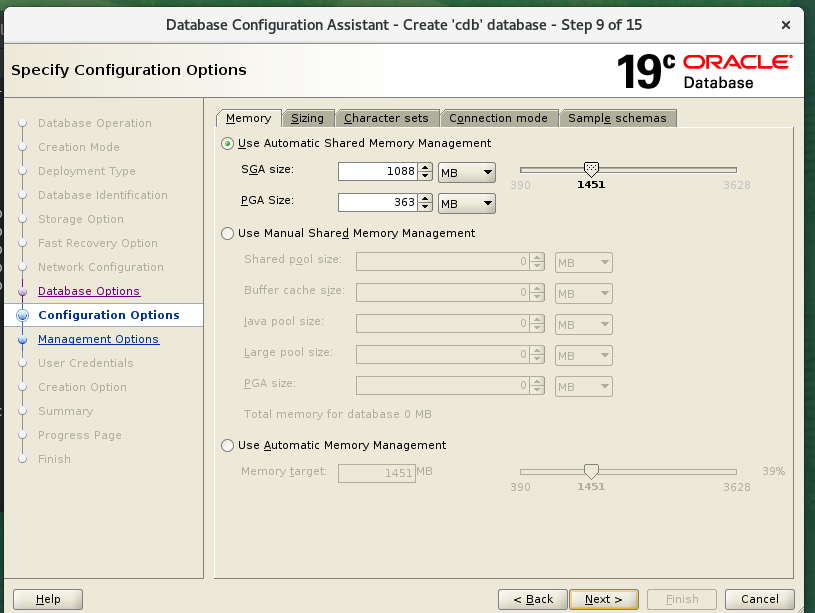
1. Pada bagian **Network Configuration** pilih Listener yang akan digunakan.



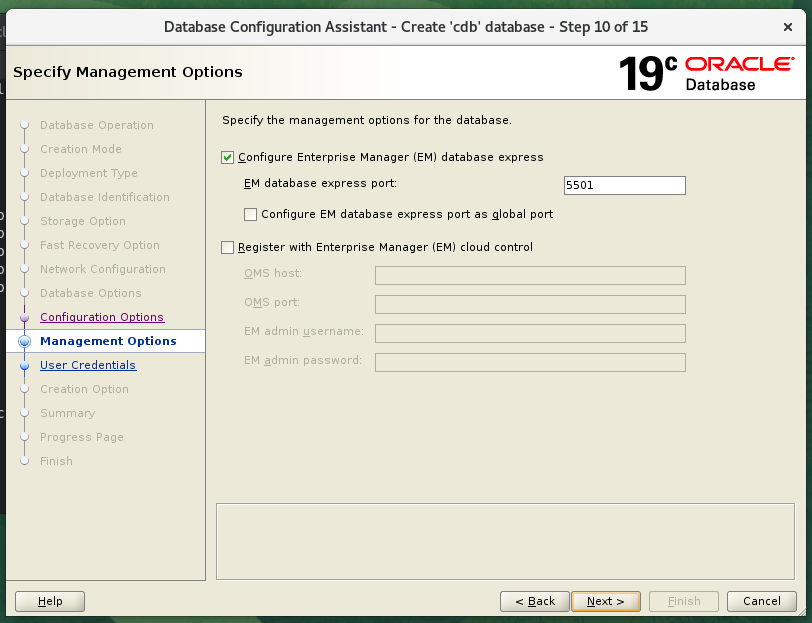
1. Pada bagian **Database Options** kita biarkan default saja.



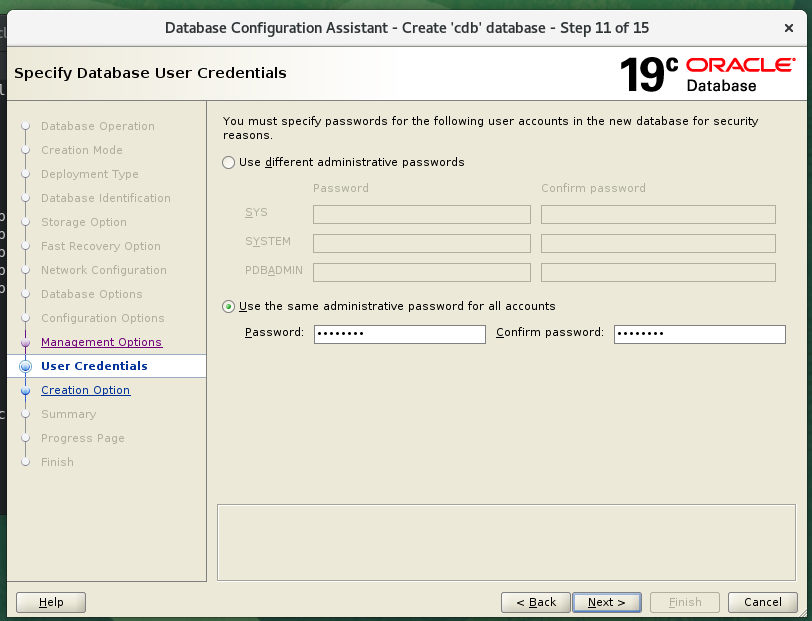
1. Pada bagian **Configuration Options** pilih yang **Use Automatic Shared Memory Management**.



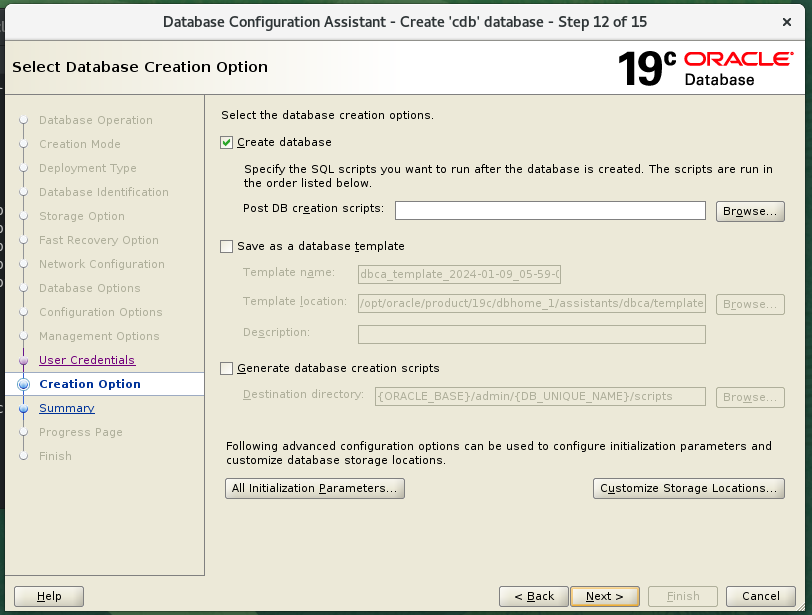
1. Pada bagian **Management Options** biarkan saja default.



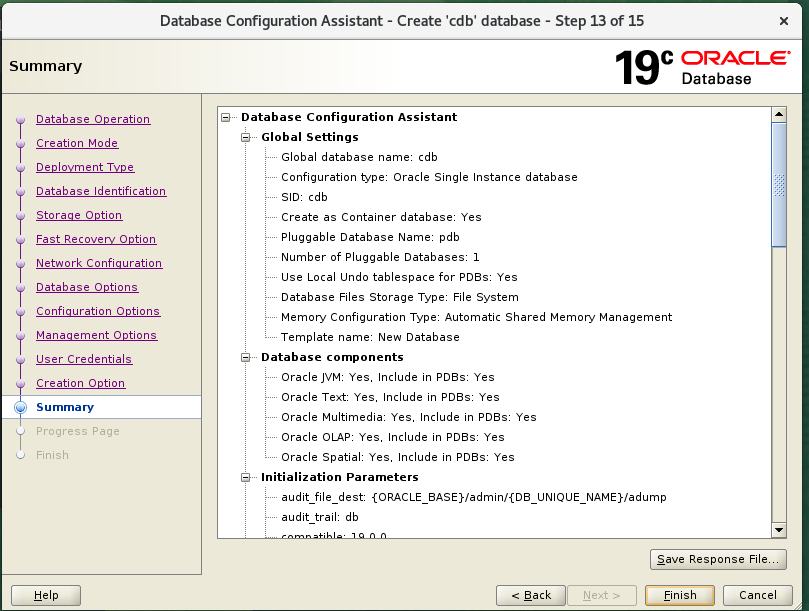
1. Pada bagian User Credential bisa disesuaikan. Karena disini saya mau untuk setiap user admin memiliki password yang sama, saya pilih opsi **Use the same administrative password for all accounts**.



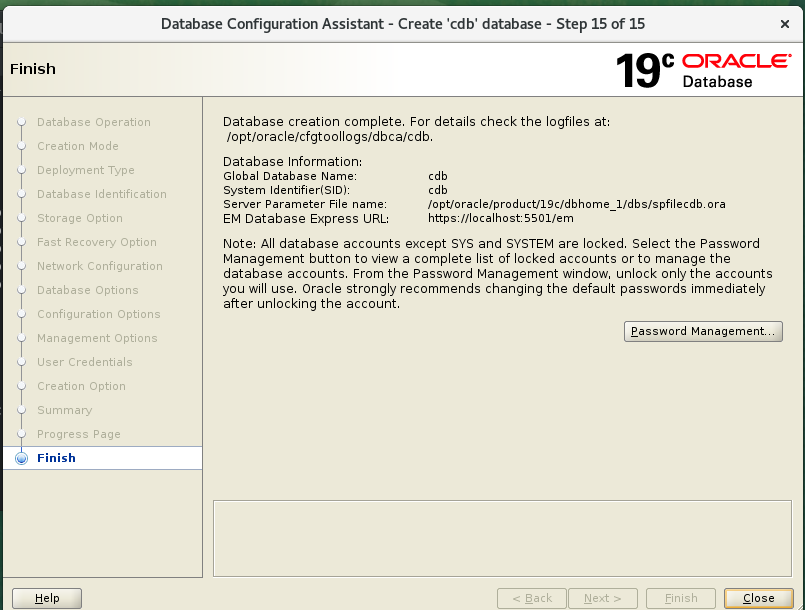
1. Pada bagian **Creation Option** biarkan default.



1. Selanjutnya pada bagian ini klik **Finish**. Jika sudah tinggal menunggu proses pembuatan databasenya.



1. Jika proses create database selesai bisa langsung diclose,



## Create PDB Using PDB SEED

1. Untuk membuat PDB menggunakan PDB$SEED kita perlu menemukan terlebih dahulu directory PDB$SEED-nya. Masuk ke SQLPLUS dan jalankan perintah dibawah.

**>** SELECT name from V$CONTAINERS;

NAME

---------------

CDB$ROOT

PDB$SEED

PDB

**>** ALTER SESSION SET CONTAINER=PDB$SEED;

**>** SELECT name FROM V$DATAFILE;

NAME

--------------------------------------------------------------------------------

/opt/oracle/oradata/CDB/xxx/datafile/o1\_mf\_system\_lspt3n3s\_.dbf

/opt/oracle/oradata/CDB/xxx/datafile/o1\_mf\_sysaux\_lspt3pow\_.dbf

/opt/oracle/oradata/CDB/xxx/datafile/o1\_mf\_undotbs1\_lspt3qgm\_.dbf

1. Nah, kita sudah tau nih dimana letak directory PDB$SEED-nya. Selanjutnya kita create PDB yang baru menggunakan perintah dibawah.

CREATE PLUGGABLE DATABASE pdb2 ADMIN USER adminpdb2 IDENTIFIED BY Itnsa123 ROLES=(dba) FILE\_NAME\_CONVERT=('/opt/oracle/oradata/CDB/xxx/datafile/','/opt/oracle/oradata/CDB/pdb2/pdb2');

Perintah FILE\_NAME\_CONVERT perlu didefine menggunakan 2 argument. Yang pertama source PDB$SEED dan kedua destination untuk PDB yang baru.

## Clone PDB

1. Sekarang kita akan coba cara lain lagi nih untuk membuat PDB nya. Dengan cara mengclone PDB yang lain. Seperti biasa masuk ke SQLPLUS.

$ sqlplus / as sysdba

SQL> show pdbs

CON\_ID CON\_NAME OPEN MODE RESTRICTED

----------- ------------------------------ ------------------ ----------

2 PDB$SEED READ ONLY NO

3 PDB READ WRITE NO

4 PDB2 READ WRITE NO

Nah, disini saya memiliki 2 PDB, yaitu PDB dan PDB2. Kita akan mencoba untuk mengclone PDB menjadi PDB3.

1. Selanjutnya kita ubah mode dari PDB menjadi Read Only.

SQL> alter pluggable database PDB close;

Pluggable database altered.

SQL> alter pluggable database PDB open READ ONLY;

Pluggable database altered.

SQL> show pdbs;

CON\_ID CON\_NAME OPEN MODE RESTRICTED

----------- ------------------------------ ------------------- ----------

2 PDB$SEED READ ONLY NO

3 PDB READ ONLY NO

4 PDB2 READ WRITE NO

1. Nah, setelah di ubah ke Read Only baru kita bisa clone PDB untuk membuat PDB lain.

SQL> CREATE PLUGGABLE DATABASE PDB3 FROM PDB CREATE\_FILE\_DEST='/opt/oracle/oradata';

1. Sekarang kita check nih, apakah Clone PDB nya sudah selesai atau belum.

SQL> show pdbs;

CON\_ID CON\_NAME OPEN MODE RESTRICTED

----------- ------------------------------ ------------------- ----------

2 PDB$SEED READ ONLY NO

3 PDB READ ONLY NO

4 PDB2 READ WRITE NO

5 PDB3 MOUNTED NO

1. Jangan lupa untuk mengubah OPEN MODE pada PDB. By default ketika kita selesai membuat PDB di SQLPLUS, PDB nya masih dalam keadaan MOUNTED. Jadi kita juga akan mengubah OPEN MODE PDB3.

SQL> alter pluggable database pdb close;

Pluggable database altered.

SQL> alter pluggable database pdb open;

Pluggable database altered.

SQL> alter pluggable database PDB3 open;

Pluggable database altered.

1. Sekarang kita cek kembali status PDB nya.

SQL> show pdbs;

CON\_ID CON\_NAME OPEN MODE RESTRICTED

----------- ------------------------------ ------------------- ----------

2 PDB$SEED READ ONLY NO

3 PDB READ ONLY NO

4 PDB2 READ WRITE NO

1. PDB3 READ WRITE NO

SQL> select name from v$datafile where con\_id=6;

NAME

--------------------------------------------------------------------------------

/opt/oracle/oradata/CDB/xxx/datafile/o1\_mf\_system\_lss8t92x\_.dbf\

/opt/oracle/oradata/CDB/xxx/datafile/o1\_mf\_sysaux\_lss8t92y\_.dbf

/opt/oracle/oradata/CDB/xxx/datafile/o1\_mf\_undotbs1\_lss8t92y\_.dbf

/opt/oracle/oradata/CDB/xxx/datafile/o1\_mf\_users\_lss8t92y\_.dbf

1. P

# MANAGING USERS & SECURITY

## Database Users

Database users dapat berupa users, application, atau group suatu object yang dimana digunakan untuk user dalam mengakses database tertentu. Pada konteks CDB, application users biasanya berada didalam PDB. Setiap database users biasanya memiliki tablespace, authentication method, unique name, temporary tablespace, user profile, initial consumer group dan account status.

Contoh syntax untuk membuat database users:

CREATE USER username

IDENTIFIED BY password

[DEFAULT TABLESPACE tablespace-name]

[QUOTA {size | UNLIMITED} ON tablespace-name]

[PROFILE {profile-name | DEFAULT}]

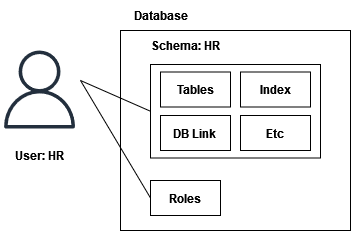
[TEMPORARY TABLESPACE temporary-tablespace-name]

[ACCOUNT {LOCK|UNLOCK}];

Penjelasan tentang syntax yang digunakan:

* Unique username untuk user yang dibuat
* Authentication method
  + password yang digunakan untuk user login ke database.
* Default tablespace
  + Tempat untuk user dalam membuat object database
* Temporary tablespace
  + Tempat menyimpan temporary object/object sementara, seperti temporary tables.
* Quota
  + Maksimum space untuk tablespace yang bisa digunakan oleh user. Gunakan syntax UNLIMITED jika tidak ingin dibatasi.
* User profile
  + Untuk limit resource database dan memberikan batasan pada password user. Jika kita skip syntax ini, biasanya akan menggunakan profile default.
* Account status
  + Untuk mengunlock dan juga lock user accounts.

## Schema



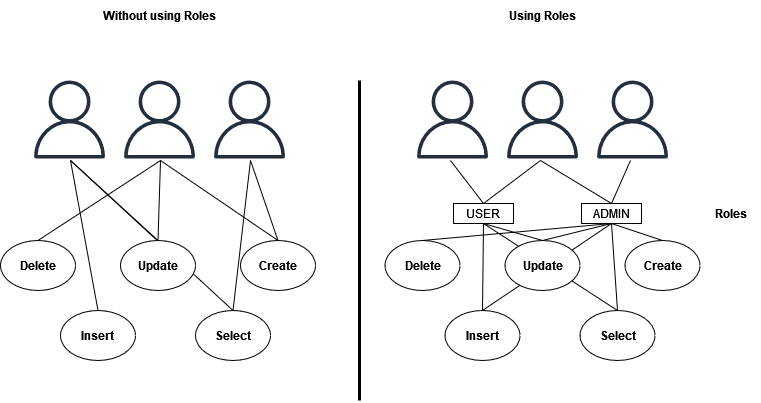
Pada oracle database, USER merupakan account name dan SCHEMA merupakan kumpulan database object yang dimiliki oleh database user. Schema biasanya memiliki nama yang sama dengan user accountnya. Secara umum, schema object mencakup semua yang dibuat oleh user pada database. Contohnya tables, indexes, synonyms, procedures, database links dan lain-lain.

## User Privileges

Oracle database juga memiliki fitur user privileges layaknya database yang lain. Pada oracle database user privileges dibagi menjadi 3, yaitu system privileges, object privileges dan administrative privileges. Berikut ini penjelasannya:

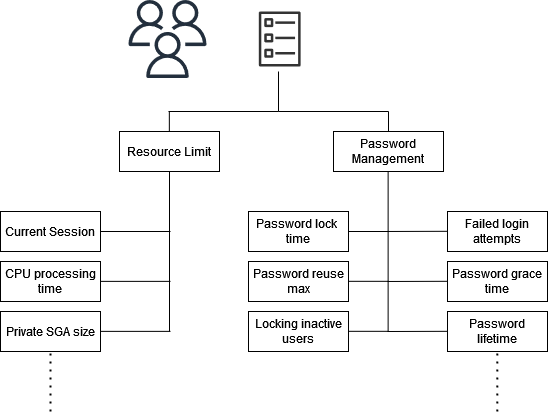
* **System privileges**
  + System privileges merupakan hak akses yang memberikan akses tingkat sistem dan operasi di dalam database
  + Contohnya: create tablespace, create table, connect to database dll.
* **Object privileges**
  + Object privileges merupakan hak akses yang diberikan ke users untuk melakukan operasi tertentu pada object database.
  + Contohnya: insert, delete, update, select dll
* **Administrative privileges**
  + Administrative privileges merupakan hak akses yang diberikan ke user untuk melakukan tugas tugas administrative.
  + Contohnya: backup and recovery, dataguard, dll

## User Roles



Sebelumnya kita telah belajar tentang privileges yang dimana memberikan hak akses kepada suatu user. Sedangkan roles merupakan kumpulan hak akses atau privileges yang dapat diberikan kepada user ataupun roles lain. Bisa dibilang roles ini gabungan dari beberapa hak akses. Dengan roles dapat mempermudah kita dalam memberikan hak akses ke user. Cukup kita assign aja roles ke usernya, tidak perlu lagi beri privileges satu-satu lagi ke setiap user. Beberapa roles bisa kita berikan kepada user yang sama.

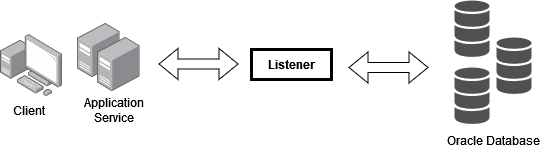
## User Profiles



User profiles merupakan kumpulan parameter dan aturan yang mengontrol hak akses user ke dalam database. User profiles mengatur terkait aspek-aspek tertentu, seperit resource limit dan manajemen password.

# MANAGING DATABASE CONNECTIVITY

## Oracle Net Listener



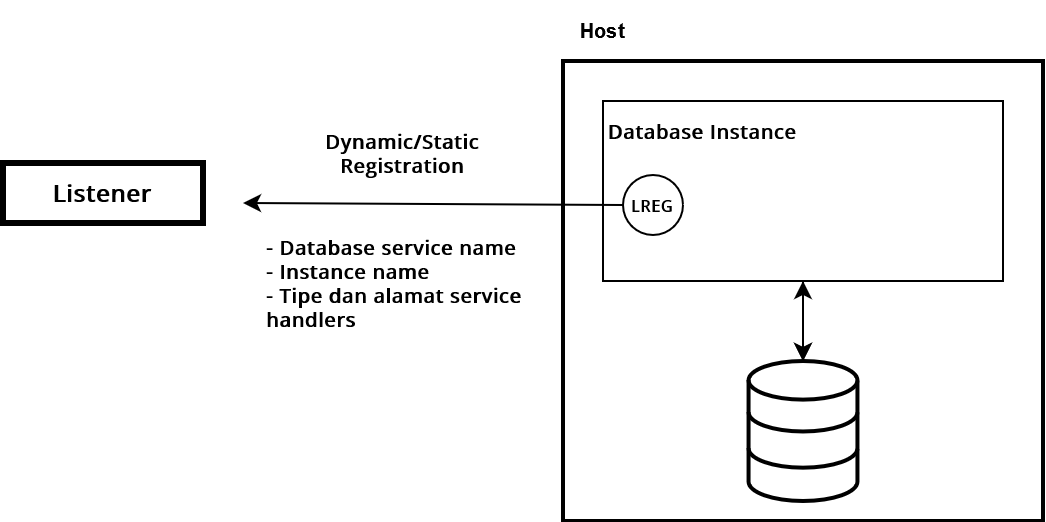
Pada oracle database kita ketahui bahwa ketika client melakukan request/transaksi, request/transaksi client tidak akan masuk ke database secara langsung, melainkan dihandle/melalui component oracle database yang disebut listener. Jika kita stop atau start listener, hal itu tidak mempengaruhi fungsionalitas database dengan cara apa pun. Bahkan, secara teknis, kita dapat membiarkan listener berjalan di server yang terpisah, tetapi kita biasanya menjalankannya di server database yang sama. Listener bekerja untuk membangun koneksi antara klien dan database dengan berkoordinasi dengan database.

## Oracle Net Service



Oracle net service adalah lapisan jaringan kecil yang berada setelah aplikasi di sisi klien dan sebelum database di sisi database. Oracle net bertindak sebagai kurir data antara aplikasi klien dan mesin database. Oracle net bertanggung jawab untuk membangun dan memelihara koneksi antara aplikasi klien dan server database, serta bertukar pesan di antara keduanya. Di sisi database, Oracle Net secara otomatis terinstal sebagai bagian dari instalasi software database. Di sisi klien, biasanya menginstal Oracle Net sebagai bagian dari menginstal software oracle client database. Namun hal ini tidak diperlukan, terkadang client application memiliki Oracle net component yang dibangun di dalam aplikasi dan oleh karena itu dapat berkomunikasi dengan database Oracle tanpa perlu menginstal software oracle client database.

## Service Registration



Proses menginformasikan listener tentang layanan database yang sedang berjalan disebut service registration. Tanpa service registration, listener tidak tahu tentang layanan database yang sedang berjalan dan dalam hal ini tidak dapat membantu permintaan koneksi dari klien. Informasi yang diperoleh listener sebagai bagian dari database registration termasuk database service name, instance name, dan ketika server bersama dikonfigurasi, ia perlu mengetahui tipe dan alamat service handlers. Terdapat 2 tipe service registration, yaitu dynamic registration dan static registration.

Dynamic registration berarti database Oracle secara otomatis melakukan registrasi di listener. Di Oracle 19c, proses yang bertanggung jawab atas dynamic registration ke dalam listener disebut LREG (Listener Registration process).

Static registration biasanya dilakukan oleh DBA. Biasanya menggunakan utilitas yang disebut Net manager atau menggunakan text editor untuk menyediakan informasi yang diperlukan tentang database service ke listener. Pada rilis oracle database yang paling awal, static registration merupakan satu-satunya pilihan yang tersedia untuk mendaftarkan database di listener.

## Naming Methods

Naming methods merujuk pada cara untuk mengidentifikasi database yang dituju dalam suatu jaringan. Naming methods dapat berarti juga cara untuk berkoneksi ke database melalui pengenalan (naming). Berikut ini beberapa jenis naming methods:

* Easy Connect
  + Metode yang memungkinkan klien menentukan alamat database langsung dalam connection string tanpa bergantung pada file konfigurasi tnsnames.ora.
  + Contoh:

sqlplus user/pass@ip\_or\_hostname\_db:1521/mydb

* Local Naming
  + Metode yang menggunakan konvensi alamat file tnsnames.ora di mesin klien untuk menentukan alamat database.
  + Contoh file tnsname.ora:

DB1 =

(DESCRIPTION =

(ADDRESS = (PROTOCOL = TCP)(HOST = localhost)(PORT = 1521))

(CONNECT\_DATA =

(SERVER = DEDICATED)

(SERVICE\_NAME = db1)

)

)

* + Contoh koneksi:

sqlplus user/pass@DB1

* Directory Naming
  + Memanfaatkan direktori atau layanan direktori, seperti ldap.
  + Contoh:

sqlplus user/pass@ldap\_db

* External Naming
  + Menggunakan layanan eksternal seperti NIS (Network Information Service) atau NIS+ untuk menentukan alamat database.
  + Contoh:

sqlplus user/pass@nis\_db

## Database Links

Database Link adalah objek database yang memungkinkan koneksi dan akses data antara dua database yang berbeda. Dengan database link memungkinkan kita untuk menjalankan query dan perintah SQL disebuah database dan dapat merujuk ke objek database yang lain. Singkatnya kita bisa melakukan remote ke server database lain melalui database local kita.

Database link dibagi menjadi 3 tipe, yaitu:

* Public
  + Tipe database link yang dapat diakses semua user database.

CREATE PUBLIC DATABASE LINK <nama db-link>

* Private
  + Tipe database link yang hanya dapat diakses oleh ownernya.

CREATE DATABASE LINK <nama db-link>

* Global
  + Tipe database link yang didefine pada directory service. Ini terjadi ketika oracle network menggunakan directory server.

## Database Link Users

Ketika kita ingin membuat database link, kita perlu menentukan user yang akan mempunyai akses remote kedatabase yang lain. Disini ada beberapa tipe database link user yang bisa kita gunakan.

* Connected User
  + Local user yang mengakses database link tanpat fixed username dan password.
  + Jika SYSTEM mengakses public database link, maka connected usernya adalah SYSTEM.
  + Connected user tidak harus user yang membuat database link, tapi semua user yang terhubung ke database link.
  + Perintah yang digunakan untuk membuat connected user:

CREATE PUBLIC DATABASE LINK <nama db-link> USING '<connection name>';

* Fixed User
  + User dan password yang didefine pada database link.
  + Perintah yang digunakan untuk membuat fixed user:

CREATE [PUBLIC] DATABASE LINK <nama db-link>

CONNECT TO <user name> IDENTIFIED BY <password>

USING '<connection string>';

* Current User
  + User yang terhubung pada global user.
  + Local user yang terhubung sebagai global user dalam context store procedure, tanpa harus menyimpan password global user.
  + Contoh, ketika terhubung ke database sebagai user “adi”, “adi” dapat mengakses prosedur yang user lain buat seperti prosedur user “budi” dan mengakses account dan schema user “budi”.

## Naming Database Links

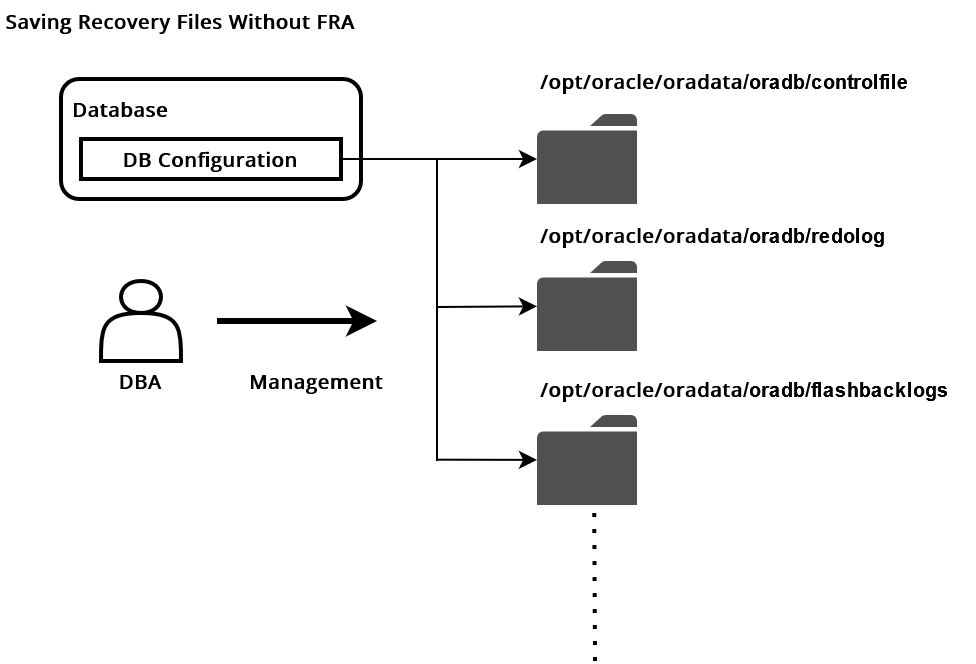
Pada oracle database, terdapat parameter yang disebut global names. Jika parameter GLOBAL\_NAMES di set TRUE , maka nama database link harus sama dengan Global database name dari remote database. Jika parameter GLOBAL\_NAMES di set FALSE, maka nama database link tidak harus sama dengan Global database name dari remote database, bisa diberi nama apa saja. Sangat direkomendasikan jika parameter GLOBAL\_NAMES di set TRUE.

## Lab

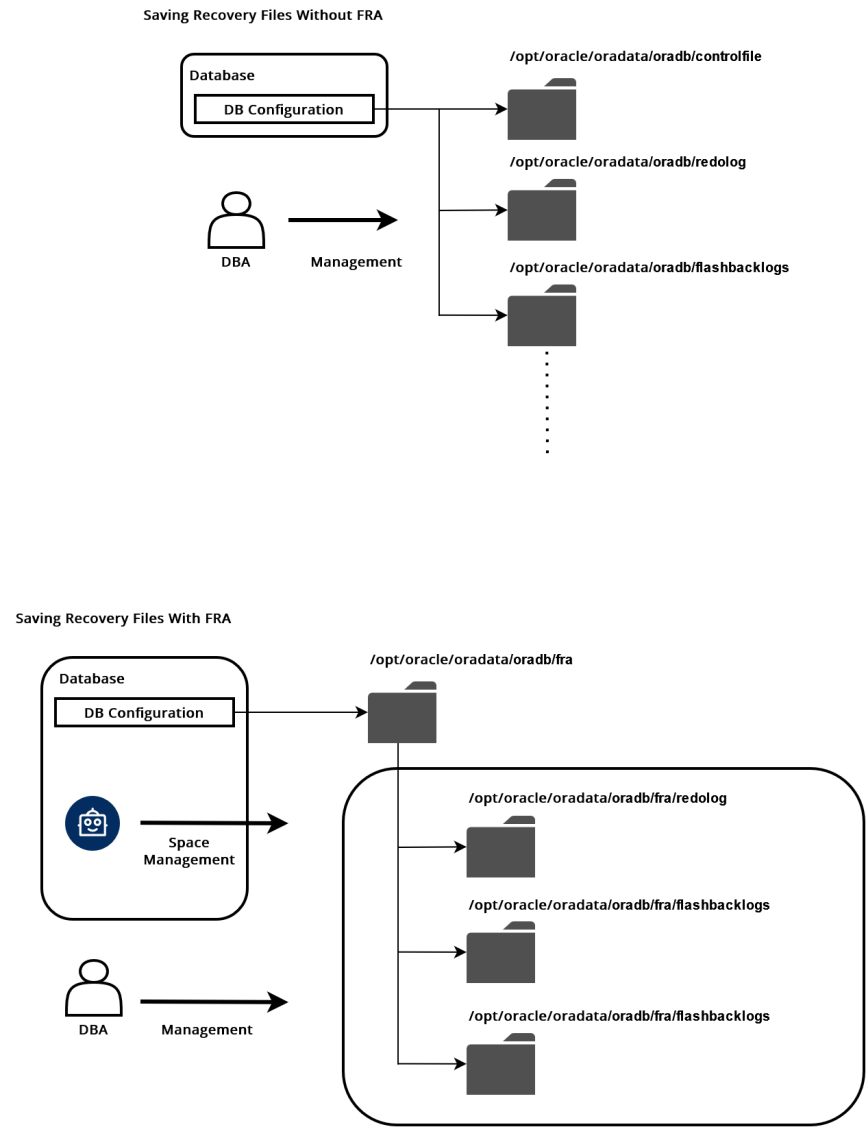
# UNDO & FLASHBACK

## Fast Recovery Area

Fast Recovery Area atau FRA merupakan disk area/lokasi penyimpanan yang dibuat untuk memfasilitasi dan menyederhanakan proses recovery dan pengelolaan backup dalam sistem database oracle. FRA menyediakan ruang penyimpanan yang terpusat secara otomatis dikelola oleh oracle database, memungkinkan administrasi dan proses recovery lebih efisien. Perlu diingat FRA juga perlu disetting oleh DBA dalam menentukan seberapa besar size yang dimiliki FRA.



Normalnya jika kita tidak menggunakan FRA untuk menyimpan beberapa database file tertentu harus dikelola oleh DBA. Dan hal ini cukup merepotkan, karena ketika file tersebut penuh DBA juga harus melakukan pembersihan secara manual.



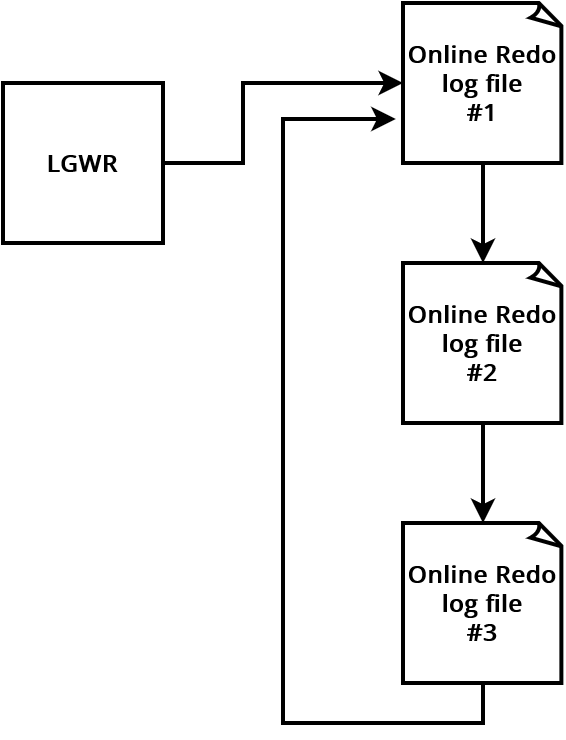
Dengan menggunakan FRA penyimpanan database file akan lebih terpusat dibawah directory FRA. Contohnya seperti subdirectory controlfiles, redolog dll. Dengan ini dapat membuat kita mudah dalam mengelolanya karena tersimpan dalam satu sub directory. Selain management otomatis, kita juga bisa mengkontrol FRA untuk menghapus beberapa fle tertentu yang tidak digunakan. Kita juga mengecualikan beberapa file untuk disimpan directory yang berbeda.

|  |  |
| --- | --- |
| **File Types** | **Permanent or Transient (\*)** |
| Control Files | Permanent |
| Online Redo Log Files | Permanent |
| Archived Redo Log Files | Transient |
| Flashback Logs | Transient |
| RMAN Backups | Transient |

\* Permanent: active files used by running database instance. Transient: other file type no required by the database operation but required for recoverability.

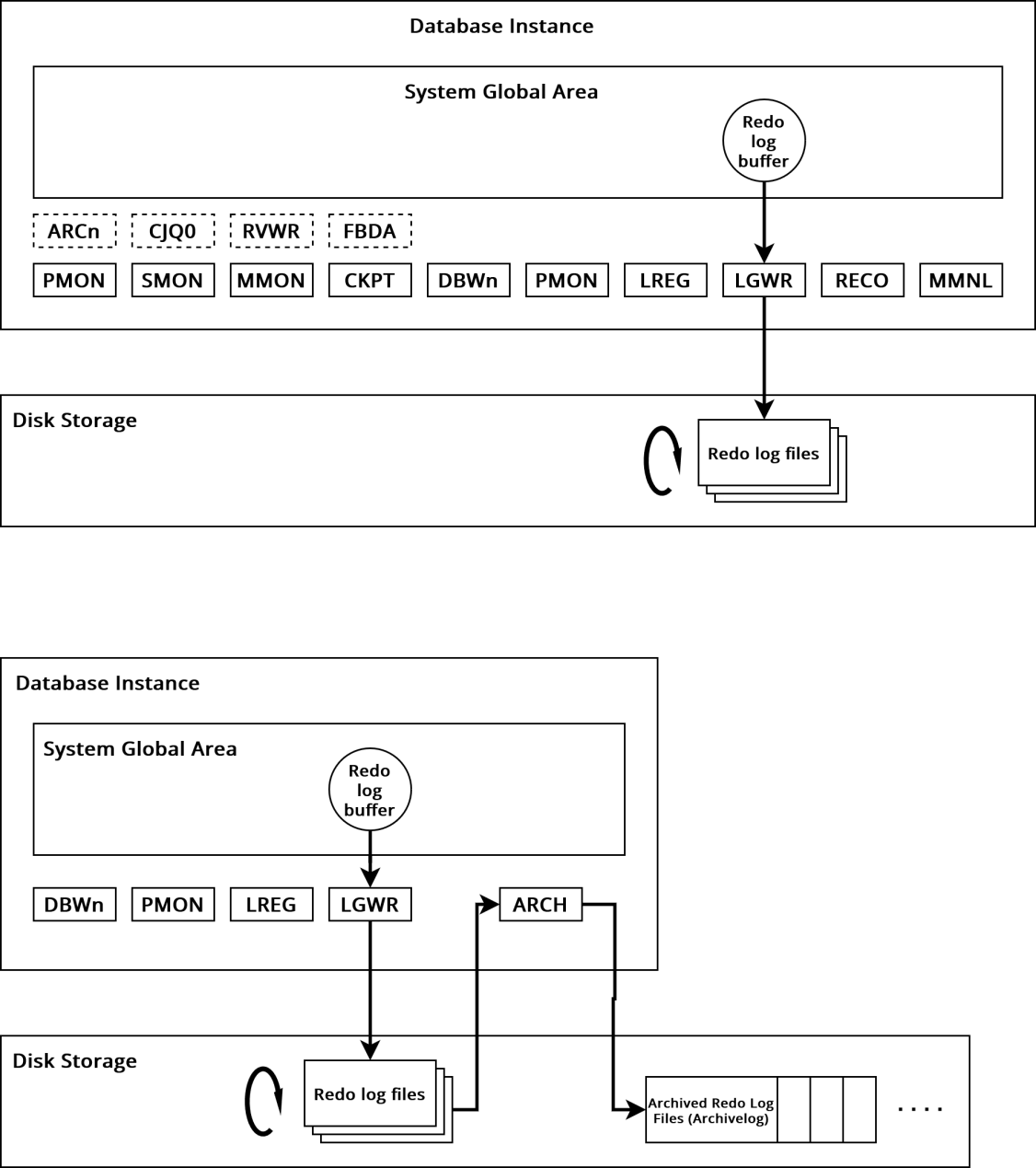
## Redo Log

Redo log merupakan komponen dalam struktur atau architecture oracle database yang bertanggung jawab untuk merekam perubahan data pada tingkat transaksi. Informasi ini kemudian digunakan untuk melakukan recovery pada kondisi sistem yang abnormal. Catatan perubahan yang terjadi pada database kita kenal dengan sebutan redo entries. Redo log terdiri dari dua atau lebih redo log files. Database memerlukan minimal dua redo log file untuk menjamin bahwa satu file selalu tersedia untuk ditulis, sementara file yang lain diarsipkan (jika database dalam mode archive log).



LGWR (Log Writer) menyimpan redo entries atau menulis redo log files secara melingkar. Ketika redo log files saat ini terisi, LGWR akan menulis redo log file berikutnya yang tersedia. Ketika redo log files terakhir juga terisi, maka LGWR akan kembali ke redo log files yang pertama dan begitu pun seterusnya, jadi berulang-ulang seperti lingkaran. Redo log sendiri terbagi menjadi 2 mode, yaitu:

* NOARCHIVE LOG

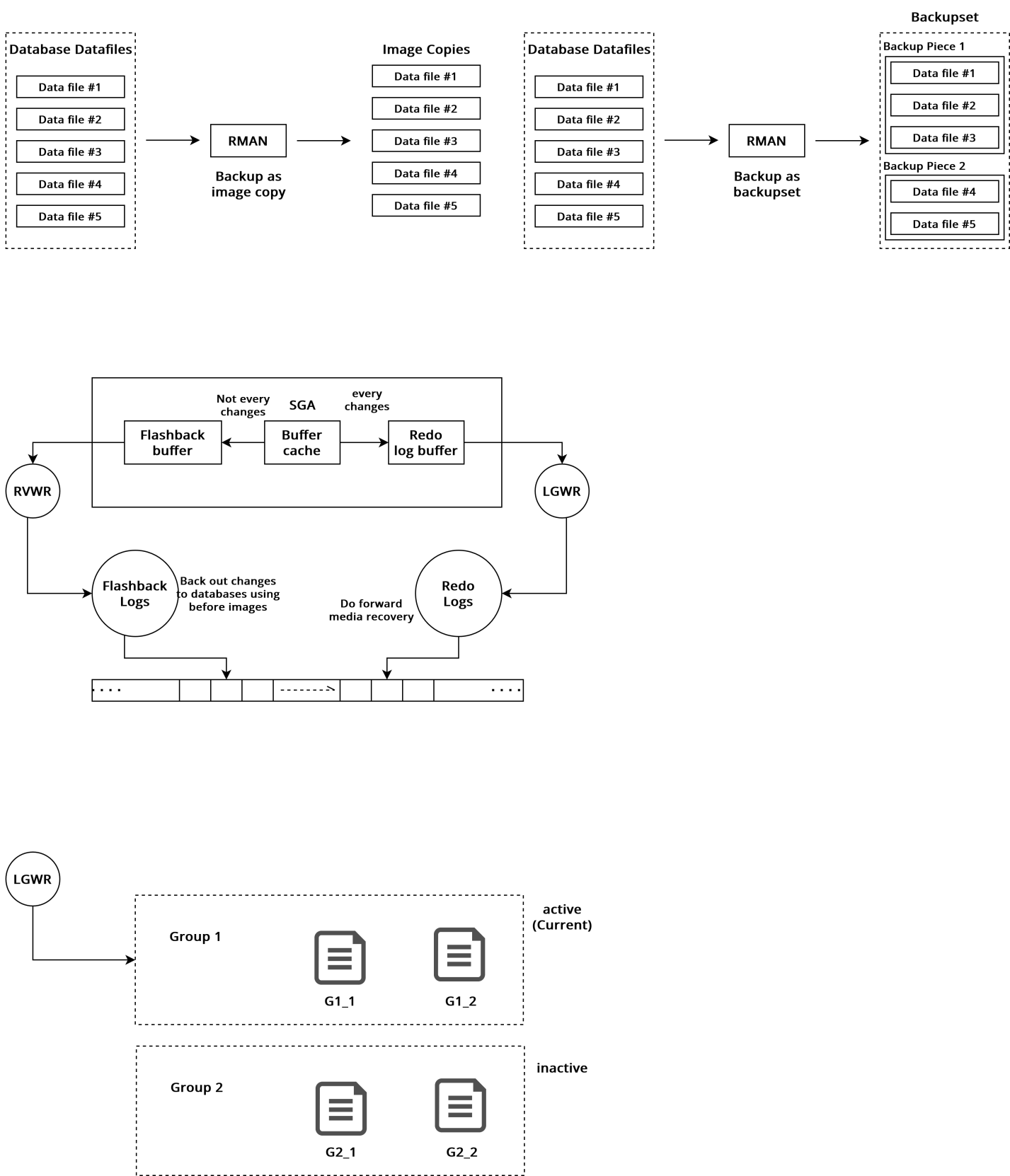


Noarchive log berarti redo log files yang sudah terisi akan tersedia untuk digunakan LGWR apabila perubahan yang terjadi sudah direkam ke datafiles.

* ARCHIVE LOG



Sedangkan pada archive log, redo log files yang sudah terisi dapat digunakan kembali oleh LGWR apabila perubahan yang terjadi sudah direkam ke datafiles dan file tersebut sudah diarsipkan.



Oracle database hanya akan menggunakan satu log file dalam satu waktu untuk menyimpan redo entries yang ditulis dari redo log buffer. Redo log files yang sedang ditulis oleh LGWR disebut dengan current redo log file. Redo log files yang diperlukan untuk recovery disebut dengan active redo log file. Sedangkan redo log files yang tidak lagi diperlukan disebu dengan inative redo log files. Pada saat proses penulisan oleh LGWR ke redo log data files dicommit. LGWR juga akan menentukan system change number (SCN) sebagai identitas redo entry untuk setiap transaksi yang dilakukan. **SCN** itu merupakan nomor unik yang dibuat LGWR sebagai identitas pada saat kita melakukan perubahan pada database.

Secara struktur redo log file terdiri dari satu atau lebih grup. Setiap grup memiliki beberapa redo log member (salinan fisik dari redo log). Setiap redo log member merupakan file biner yang disimpan di disk fisik.

## Archive Log

Pada materi sebelumnya kita telah belajar tentang mode-mode yang ada pada redo log, yaitu noarchive log dan archive log. Dimateri ini kita bakal memahami lebih lanjut tentang archive log dan cara mengelolanya.

Perbandingan noarchive log dan archive log:

|  |  |
| --- | --- |
| **NoArchiveLog** | **ArchiveLog** |
| Melindungi dari kegagalan instance tapi tidak dengan kegagalan media | Melindungi dari kegagalan media |
| Tidak bisa online backup | Bisa online backup |
| Membutuhkan lebih sedikit biaya administrasi | Membutuhkan lebih banyak biaya administrasi: menetapkan kebijakan penghapusan untuk archive log |
| Membutuhkan lebih sedikit penyimpanan | Membutuhkan penyimpanan yang lebih tinggi |
| Menimbulkan lebih sedikit kinerja | Menimbulkan sedikit lebih banyak kinerja |
| Digunakan untuk POC or testing environments | Digunakan untuk production |

**Enabling archive log mode:**

* Menentukan letak archive log file.
  + Letak archive log file dapat berupa direktori yang ditentukan DBA atau bisa juga dimanage by FRA.
  + Bisa dikonfigurasi multiple destination filenya.
  + Perintah yang digunakan untuk destination arhive logs file:

LOG\_ARCHIVE\_DEST\_n

* + Contoh:

ALTER SYSTEM SET LOG\_ARCHIVE\_DEST\_1 = 'LOCATION=/opt/oracle/oradata/arc';

//dengan menggunakan FRA

ALTER SYSTEM SET LOG\_ARCHIVE\_DEST\_2=

'LOCATION=USE\_DB\_RECOVERY\_FILE\_DEST';

* Menentukan archive log file name format (optional).
* Enable archive log mode
  + Setting destination archive log file dan format name.
  + Shutdown database (cleanly).

SHUTDOWN IMMEDIATE

* + Mount the database.

STARTUP MOUNT

* + Setting archive log mode.

ALTER DATABASE ARCHIVELOG;

* + Open the database.

ALTER DATABASE OPEN ;

## Undo

Pada oracle database terdapat fitur untuk membalikkan suatu kondisi atau keadaan data sebelum diubah oleh suatu transaksi. Fitur ini disebut dengan undo. Layaknya fitur undo pada database lainnya, di oracle database juga dapat membalikkan keadaan suatu data yang belum dicommit oleh suatu transaksi. Untuk menggunakan fitur undo cukup menjalankan perintah:

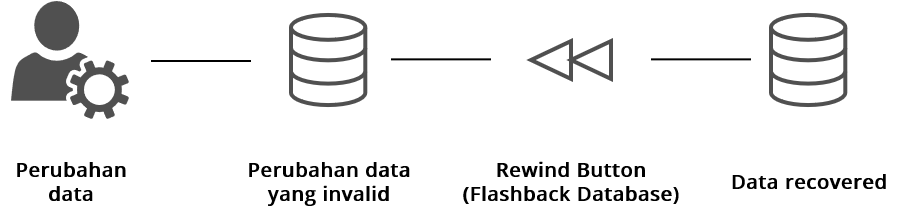
ROLLBACK;

## Flashback

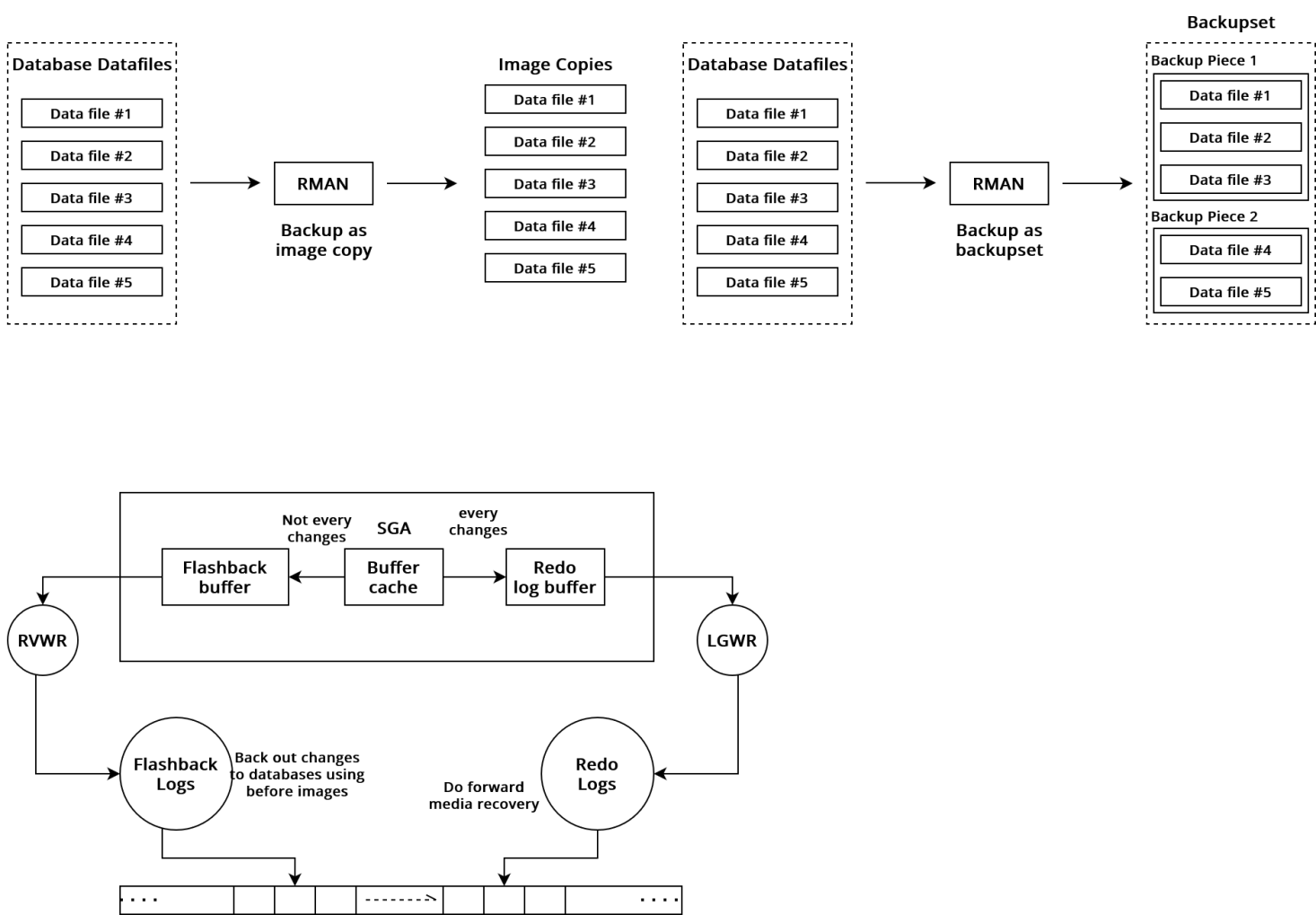
Flashback layaknya tombol rewind pada database. Yang mana bisa digunakan untuk recover logical data yang disebabkan oleh user.

Flashback bisa digunakan untuk skenario dibawah:

* Memperbaiki kegagalan data logis seperti data yang tidak valid yang dimasukkan oleh pengguna.
* Mengembalikan perubahan yang dibuat pada database (seperti upgrade patch aplikasi).
* Testing environments.



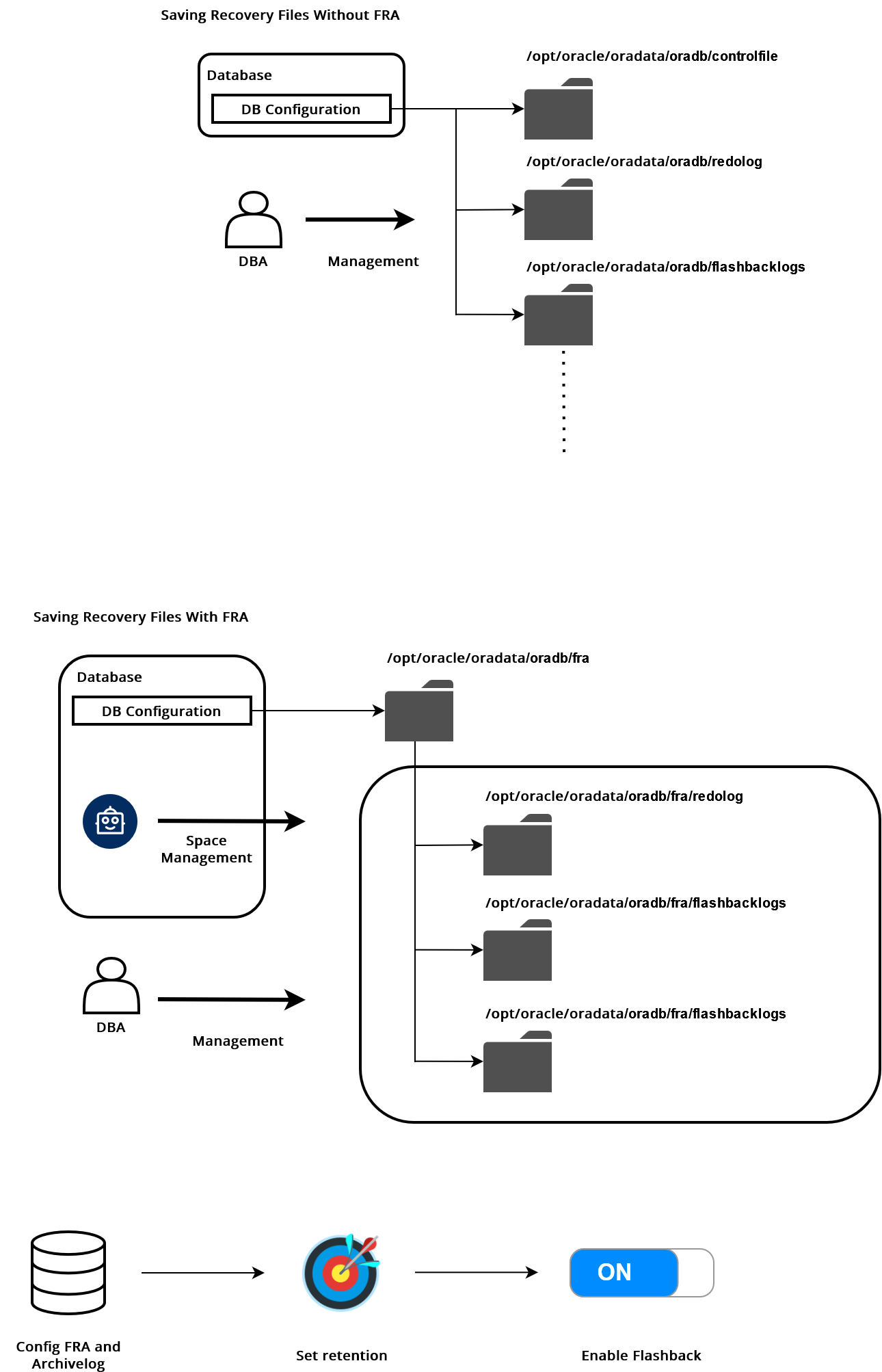
### Flashback Database Architecture



Untuk menggunakan flashback kita perlu mengenabled kan-nya terlebih dahulu. Ketika flashback enabled oracle database akan membuat spesial area memory yang disebut Flashback buffer. Flashback buffer dimanaged oleh RVWR (Flashback Writer). Ketika flashback start, system akan juga membuat sebuah file bernama Flashback logs pada storage. Oracle database menggunakan informasi yang ada pada Flashback logs untuk melakukan UNDO perubahan/changes pada database.

Sistem secara otomatis membuat, menghapus, dan menyesuaikan ukuran log ini di fast recovery area (FRA). Flashback database hanyalah salah satu options yang ada pada flashback options. Flashback optoins sendiri ada berbagai macam seperti flashback query, flashback table dll.

### Configuring Flashback Database



Pada saat melakukan konfigurasi flashback database kita perlu mengecek terlebih dahulu apakah kita punya cukup space pada FRA, jadi pastikan FRA memiliki space yang cukup untuk menampung file flashback. Kemudian kita konfigurasi FRA dan setup database dalam ARCHIVELOG MODE.

Selanjutnya, kita setting retention target, retention target digunakan untuk seberapa jauh bisa melakukan flashback database. Rentang retentionnya menggunakan satuan menit.

ALTER SYSTEM SET DB\_FLASHBACK\_RETENTION\_TARGET=2800 SCOPE=BOTH;

Perlu diingat bahwa ini adalah target bukan jaminan. Jumlah waktu aktual yang dapat digunakan untuk mem-flashback database tergantung pada berapa banyak data flashback yang tersimpan di dalam FRA.

Terakhir baru enabled flashback database.

ALTER DATABASE FLASHBACK ON;

SELECT FLASHBACK\_ON FORM V$DATABASE;

## Restore Point

Restore point merupakan salah satu fitur yang terdapat pada oracle database. Restore point berguna sebagai penanda/penamaan suatu titik waktu tertentu atau SCN pada database, yang dapat digunakan untuk melakukan flashback dan recovery. Dengan restore point ini dapat mempermudah kita dalam melakukan flashback pada database tertentu. Restore point terbagi menjadi 2 jenis, yaitu:

### Normal Restore Point

Normal restore point merupakan jenis restore point yang dibuat secara sederhana dan hanya bertahan selama database tidak bermasalah. Misalkan ketika database mengalami restart maka normal restore point akan terhapus secara otomatis.

Cara membuat normal restore point:

* Membuat restore point berdasarkan SCN saat ini.

CREATE RESTORE POINT name\_restore\_point;

* Membuat restore point berdasarkan spesifik SCN dan timestamp.

CREATE RESTORE POINT name\_restore\_point AS OF TIMESTAMP ‘…’;

CREATE RESTORE POINT name\_restore\_point AS OF SCN ‘…’;

* Membuat restore point berdasarkan spesifik PDB.

CREATE RESTORE POINT name\_restore\_point FOR PLUGGABLE DATABASE name\_pdb;

### Guaranteed Restore Point

Tipe yang kedua adalah Guaranteed Restore Point. Berbeda dengan normal restore point yang dimana dapat terhapus suatu waktu ketika database mengalami masalah, sedangkan guaranteed restore point tidak akan terhapus walau database mengalam restart atau semacamnya. Guaranteed restore point berguna ketika ingin membuat suatu titik restore yang tetap, namun perlu diingat bahwa guaranteed restore point perlu dihapus secara manual suatu waktu.

Cara membuat Guaranteed Restore Point:

CREATE RESTORE POINT name\_restore\_point GUARANTEE FLASHBACK DATABASE;

Cara menghapus Restore Point:

* Menghapus restore point pada container saat ini.

DROP RESTORE POINT name\_restore\_point;

* Menghapus restore point berdasarkan spesifik PDB.

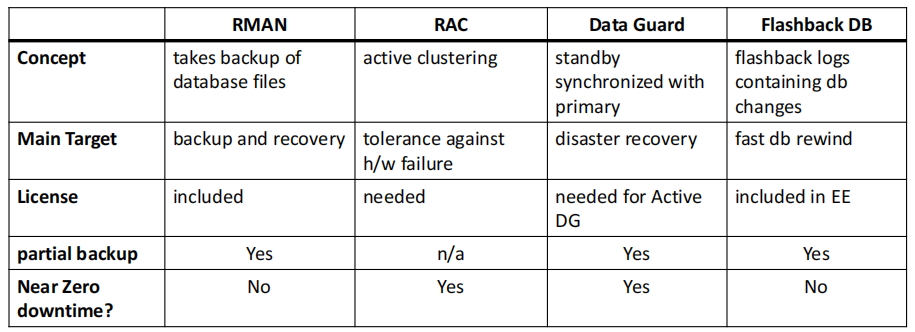
DROP RESTORE POINT name\_restore\_point FOR PLUGGABLE DATABASE name\_pdb;

# BACKUP & RECOVERY

## Database Backup and Recovery

Backup merupakan proses untuk mencadangkan database. Sedangkan recovery berarti proses mengembalikan database ke kondisi konsisten setelah terjadi kegagalan atau hilangnya data. Pada saat mengembangkan planning untuk backup dan recovery, sangat penting bagi seorang DBA untuk memahami tentang jenis kegagalan yang mungkin terjadi pada database. Seperti yang diketahui metode yang digunakan mungkin berbeda tergantung dari kategori kegagalannya. Berikut ini beberapa contoh kategori kegagalan (categories of failure) yang bisa terjadi pada database:

* Media failure
  + Jenis kegagalan yang terjadi ketika penyimpanan/disk mengalami kerusakan. Kerusakan pada disk paling baik ditangani dengan menggunakan oracle recovery manager.
* Human errors
  + Biasanya terjadi karena kesalahan manusia. Contohnya drop table, update incorrect data, delete data dll
* Network failure
  + Koneksi ke database terputus. Cara terbaik adalah menyediakan jalur koneksi redundant. Contoh network failure adalah listener fails dan NIC fails.
* Machine Failure
  + Ketika instance mati secara tak terduga. Tidak ada data yang hilang namun user tidak bisa mengakses database. Dalam hal ini DBA harus mencoba menjalankan kan ulang instance dengan perintah STARTUP. Memulihkan dari kegagalan instance bersifat otomatis, termasuk rolling forward redo log dan kemudian rolling back semua transaksi yang belum dicommit.

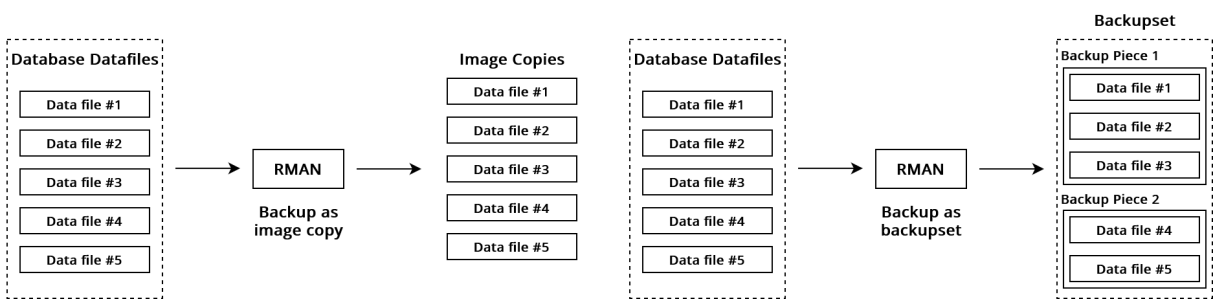


## RMAN

Recovery Manager atau yang dikenal dengan RMAN merupakan teknologi backup dari oracle yang ditugaskan untuk menangani backup, restore dan recovery database. RMAN menyediakan antarmuka via command line. RMAN support beberapa tipe file untuk dibackup, seperti data files, control files, server parameter file, archived redo logs. Untuk tipe file yang tidak dapat dibackup oleh RMAN adalah online redo log files dan auxiliary files (network configuration files, password files, Keystore (wallet)).

RMAN Support 2 tipe backup, yaitu:

* **Full Backups**: full backup berarti membuat salinan semua database. RMAN membuat salinan dari setiap data block yang berisi data.
* **Incremental Backups**: menyalin data block yang telah berubah/changes sejak pencadangan sebelumnya. Incremental backup terbagi menjadi 2, yaitu:
  + **Level 0**: level 0 incremental backup itu sebenarnya sama dengan full backup.
  + **Level 1**: level 1 incremental backup dibagi menjadi 2 bagian, yaitu:
    - **Level 1 Differential**: tipe default dari incremental backup. Yang dimana melakukan backup setelah perubahan terbaru dari incremental backup terbaru.
    - **Level 1 Cumulative**: pada level 1 akan melakukan backup setelah terjadi perubahan level 0 terbaru.



Pada RMAN terdapat dua cara dalam menyimpan database yang dibackup, yaitu **image copies** dan **backup sets**. **Image copies** adalah salinan dari bit-bit file. Image copies layaknya kita mengcopy semua atau seluruh file database. Kita dapat melakuakn image copies menggunakan RMAN ketika database dalam keadaan OPEN. Sedangkan **backup sets** merupakan kumpulan backup pieces yang telah dibackup oleh RMAN. Backup pieces sendiri berisi backup data files, control files atau archive redo log yang berasal dari database. Bisa dibilang setiap pieces atau potongan dari backup set berisi data files, control files, archived redo log dan sebagainya. Backup set merupakan logical concepts dan backup pieces merupakan physical conceptnya.

## P

## p

## P

# DATA GUARD

## Data Guard

Data guard merupakan solusi High Availability (HA) dari oracle database. Data guard menyediakan fungsionalitas replikasi pada database. Sehingga setiap terjadinya perubahan pada primary maka akan ada update ke standby server database. Selain itu data guard memiliki fungsi sebagai switchover/failover database. Sehingga jika terjadinya kegagalan atau kerusakan pada primary database, secondary/standby dapat menggantikan fungsi primary database.

## Prepare Primary Server Database

Pertama kita enable mode archivelog pada primary database.

SELECT log\_mode from v$database ;

ALTER SYSTEM SET log\_archive\_dest\_1 = 'LOCATION=USE\_DB\_RECOVERY\_FILE\_DEST' scope=both;

ALTER SYSTEM SET LOG\_ARCHIVE\_DEST\_STATE\_2=ENABLE;

ALTER SYSTEM SET LOG\_ARCHIVE\_FORMAT='%t\_%s\_%r.arc' SCOPE=SPFILE;

ALTER SYSTEM SET REMOTE\_LOGIN\_PASSWORDFILE=EXCLUSIVE SCOPE=SPFILE;

shutdown immediate

startup mount

ALTER DATABASE ARCHIVELOG;

ALTER DATABASE OPEN;

ALTER SYSTEM SWITCH logfile;

SELECT name from v$archived\_log;

SELECT log\_mode from v$database ;

Kemudian, enable force logging.

SELECT force\_logging from v$database;

ALTER database force logging;

SELECT force\_logging from v$database;

Sekarang kita check redo log size yang dimiliki primary database.

SELECT group#, thread#, bytes/1024/1024 mb from v$log;

SELECT thread#, instance from v$thread ;

Setelah itu, kita akan membuat standby redo logs, disini pastikan thread dan size nya sama dengan redo log size yang tadi kita check.

ALTER database add standby logfile thread 1 size 200m;

ALTER database add standby logfile thread 1 size 200m;

ALTER database add standby logfile thread 1 size 200m;

Kemudian kita check nih standby redo log sizenya.

SELECT group#, thread#, sequence#, bytes/1024/1024 mb, archived, status from v$standby\_log;

SELECT member from v$logfile where type = 'STANDBY';

Check db\_unique\_name yang dimiliki primary database. Nantinya db\_unique\_name ini kita perlukan untuk membedakan standby dan primary.

show parameter db\_unique\_name

Sekarang kita setting log\_archive\_config. Pastikan pada dg\_config berisi db\_unique\_name untuk primary dan standby.

ALTER SYSTEM set log\_archive\_config='DG\_CONFIG=(ORCLDB,STBYDB)' scope=both;

Setting log\_archive\_dest\_2 untuk menentukan lokasi archive log secondarynya.

ALTER SYSTEM set log\_archive\_dest\_2='SERVICE=STBYDB ASYNC VALID\_FOR=(ONLINE\_LOGFILES,PRIMARY\_ROLE) DB\_UNIQUE\_NAME=STBYDB' scope=both;

Kemudian kita set fal\_server agar nantinya data guard dapat menentukan archive log secondarynya.

show parameter fal\_server ;

ALTER SYSTEM set fal\_server ='STBYDB' scope=both;

Selanjutnya kita setting standby\_file\_management menjadi auto.

show parameter standby\_file\_management

ALTER SYSTEM set standby\_file\_management='AUTO' scope=both;

check list directory parameter yang dimiliki primary database kita.

SELECT name, value from v$parameter where upper(value) like upper('%/orcldb/%');

Selanjutnya kita enable atau jalankan fitur flashback.

ALTER database flashback on;

Check letak dari password file primary database. Nantinya password file ini akan digunakan untuk auth dari standby ke primary database.

ls /opt/oracle/product/19c/dbhome\_1/dbs/orapw\*

Edit file tnsname.ora pada primary database dan tambahkan baris command untuk koneksi ke standby database.

# tnsnames.ora Network Configuration File: /opt/oracle/product/19c/dbhome\_1/network/admin/tnsnames.ora

# Generated by Oracle configuration tools.

LISTENER\_ORCLDB =

(ADDRESS = (PROTOCOL = TCP)(HOST = oracle1)(PORT = 1521))

ORCLDB =

(DESCRIPTION =

(ADDRESS = (PROTOCOL = TCP)(HOST = oracle1)(PORT = 1521))

(CONNECT\_DATA =

(SERVER = DEDICATED)

(SERVICE\_NAME = orcldb)

)

)

STBYDB =

(DESCRIPTION =

(ADDRESS = (PROTOCOL = TCP)(HOST = oracle2)(PORT = 1521))

(CONNECT\_DATA =

(SERVER = DEDICATED)

(SERVICE\_NAME = stbydb)

)

)

Kemudian kita lanjut untuk edit file listener.ora

# listener.ora Network Configuration File: /opt/oracle/product/19c/dbhome\_1/network/admin/listener.ora

# Generated by Oracle configuration tools.

LISTENER =

(DESCRIPTION\_LIST =

(DESCRIPTION =

(ADDRESS = (PROTOCOL = TCP)(HOST = oracle1)(PORT = 1521))

(ADDRESS = (PROTOCOL = IPC)(KEY = EXTPROC1521))

)

)

SID\_LIST\_LISTENER =

(SID\_LIST=

(SID\_DESC=

(GLOBAL\_DBNAME=orcldb)

(SID\_NAME=orcldb)

(ORACLE\_HOME=/opt/oracle/product/19c/dbhome\_1)

)

)

Setelah semua konfigurasi dilakukan kita reload listenernya.

lsnrctl reload

## Prepare Standby Server Database

Sekarang kita akan lanjut prepare pada standby server database. Pertama kita akan membuat init file yang hanya berisi db\_name untuk standby database.

echo 'DB\_NAME=ORCLDB' > $ORACLE\_HOME/dbs/initstbydb.ora

Selanjutnya kita copy password file dan file tnsname.ora dari primary server.

scp oracle@oracle1:$ORACLE\_HOME/dbs/orapw\* $ORACLE\_HOME/dbs/orapwstbydb

scp oracle@oracle1:$ORACLE\_HOME/network/admin/tnsnames.ora $ORACLE\_HOME/network/admin/tnsnames.ora

Buatlah file listener baru.

# listener.ora Network Configuration File: /opt/oracle/product/19c/dbhome\_1/network/admin/listener.ora

# Generated by Oracle configuration tools.

LISTENER =

(DESCRIPTION\_LIST =

(DESCRIPTION =

(ADDRESS = (PROTOCOL = TCP)(HOST = oracle2)(PORT = 1521))

(ADDRESS = (PROTOCOL = IPC)(KEY = EXTPROC1521))

)

)

SID\_LIST\_LISTENER=

(SID\_LIST=

(SID\_DESC=

(GLOBAL\_DBNAME=stbydb)

(SID\_NAME=stbydb)

(ORACLE\_HOME=/opt/oracle/product/19c/dbhome\_1)

)

)

Jalankan listener dan reload listener.

lsnrctl start

lsnrctl reload

Start database standby dengan mode nomount

. oraenv

ORACLE\_SID = [orcldb] ? stbydb

ORACLE\_HOME = [/home/oracle] ? /opt/oracle/product/19c/dbhome\_1

sqlplus / as sysdba

startup nomount

exit

Sekarang kita mulai replikasi databasenya. Connect ke RMAN pada primary dan standby diwaktu yang sama.

rman target sys/Itnsa123@ORCLDB auxiliary sys/Itnsa123@STBYDB

Jalankan perintah dibawah untuk memulai proses replikasi database.

DUPLICATE TARGET DATABASE FOR STANDBY FROM ACTIVE DATABASE DORECOVER

SPFILE

SET DB\_UNIQUE\_NAME 'STBYDB' COMMENT 'Is standby'

SET LOG\_ARCHIVE\_DEST\_2 'SERVICE=ORCLDB ASYNC VALID\_FOR=(ONLINE\_LOGFILES,PRIMARY\_ROLE) DB\_UNIQUE\_NAME=ORCLDB'

SET AUDIT\_FILE\_DEST '/opt/oracle/admin/stbydb/adump'

SET CONTROL\_FILES '/opt/oracle/oradata/STBYDB/controlfile/control01.ctl'

SET FAL\_SERVER 'ORCLDB'

NOFILENAMECHECK;

## Testing

Check paramater db\_name dan db\_unique\_name pada kedua server. pastikan db\_name pada kedua server sama dan db\_unique\_name nya berbeda.

sqlplus / as sysdba

SELECT database\_role from v$database;

show parameter db\_name

show parameter db\_unique\_name

Jalankan redo apply pada process pada standby server.

ALTER database recover managed standby database disconnect from session;

Check pada standby database untuk menerapkan archive.

SELECT role, thread#, sequence#, action from v$dataguard\_process;

Lakukan hal yang sama pada primary database dan set switch logfile untuk membuat archiver.

SELECT sequence#, first\_time, next\_time from v$archived\_log order by sequence#;

ALTER SYSTEM switch logfile;

Sekarang kembali lagi standby database untuk melihat apakah archiver baru sudah ke apply atau belum.

SELECT sequence#, first\_time, next\_time, applied from v$archived\_log order by sequence#;

Stop recovery process ada standby server.

ALTER database recover managed standby database cancel;

Jalankan atau enable flashback pada standby server.

ALTER database flashback on;

Kemudian start lagi recovery process pada standby server.

ALTER database recover managed standby database disconnect from session;

## Switchover

Masuk ke primary database untuk verify switchover. Pastikan untuk tidak mendapatkan error.

sqlplus / as sysdba

ALTER database switchover to STBYDB verify;

Check gap status pada primary database.

SELECT status, gap\_status from v$archive\_dest\_status where dest\_id = 2;

Switchover ke standby database server. Jalankan perintah dibawah pada primary database.

ALTER database switchover to STBYDB;

Alter database menjadi open pada standby server (sekarang menjadi primary).

alter database open;

Startup mount pada primary server (sekarang menjadi standby).

startup mount

alter database recover managed standby database disconnect;

Check status pada kedua server.

select database\_role from v$database;

select role, thread#, sequence#, action from v$dataguard\_process;

## p