BAB III

KEGIATAN SELAMA KERJA PRAKTEK

3.1 Jadwal kerja praktek

Dalam penyusunan laporan kerja praktek ini penyusun mengadakan penelitian pada Perjan Rumah Sakit Hasan Sadikin yang beralamat di Jalan Pasteur No. 33 – Bandung. selama 1 bulan mulai tanggal 03 November 2003 sampai dengan 16 Desember 2003

UNIT PELAKSANA TEKNIS PERPUSTAKAAN UNIKOM

Jadwal kerja praktek adalah sebagai berikut :

- Dimulai dari pukul 07.30 hingga pukul 14.00

3.2 Cara/teknik kerja praktek

1. Memasukkan data pasien pada bagian rawat jalan
2. Meneliti sistem informasi yang digunakan pada Perjan Hasan Sadikin unit Rawat Jalan

3.3 Data kerja praktek

Penyusun akan mencoba untuk menjelaskan mengenai penggunaan dari Sistem Informasi Terpadu RS Hasan Sadikin yang telah terintegrasi dengan Oracle 8.0. Sistem ini mempermudah dalam memasukkan data dikarenakan kelengkapan dan user interface yang friendly sehingga pengguna awam sekalipun akan dapat menggunakannya dengan mudah.

3.3.1 Arsitektur Oracle

Sebelum memasuki penjelasan dari program itu sendiri, penyusun akan menjelaskan arsitektur dari Oracle itu sendiri. Arsitektur Oracle adalah kumpulan-kumpulan dari proses-proses, bagian-bagian memori, dan file-file yang membentuk sebuah Oracle database dan layanannya

3.3.1.1 Proses

Kumpulan proses-proses yang membentuk arsitektur Oracle terbagi atas dua bagian, yaitu *background processes* dan *foreground processes.* Background processes adalah proses yang berjalan di latar belakang dan tidak memiliki Graphical User Interface (GUI). Sedangkan Foreground processes adalah proses yang berjalan di latar depan, biasamya memiliki GUI.

Background processes yang *mandatory* di Oracle DBMS terdiri atas:

- Background processes pembentuk instance
- Listener
- Server Processes

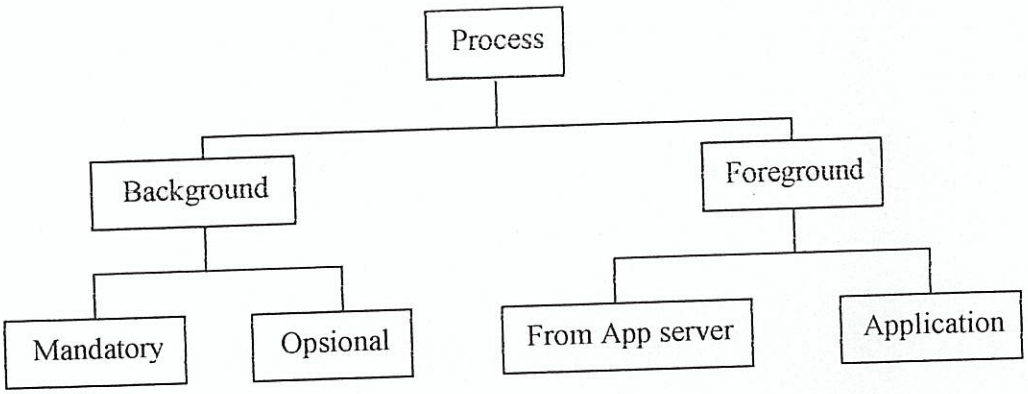
Background processes yang opsional terdiri dari:

- Dispatcher (opsional)
- Management server
- NM (network Management) agent
- Data Gatherer

Foreground processes di Oracle adalah client process, yang terbagi atas:

- Client procesess dari aplikasi server

- Client process yang merupakan aplikasi



Gambar 3-1 Klasifikasi proses di Oracle

Selain sekumpulan proses, ada bagian-bagian memori yang membentuk arsitektur Oracle, yang juga bisa dibagi atas bagian yang mandatory dan bagian yang opsional.

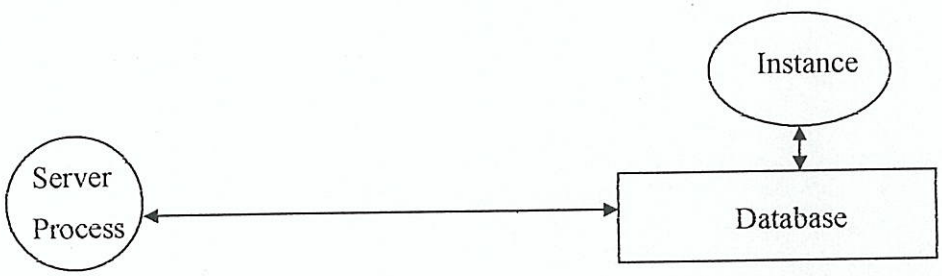
Bagian memori yang harus ada terdiri atas:

- Repository cache
- Data cache
- Log cache
- Server-process cache
- Sort cache

Bagian memori yang terdiri atas *Miscellaneous cache* (Large pool) dan *Java cache*.

3.3.1.2 Instance sebagai Operasi (Process)

Jika digabung, arsitektur Oracle instance (dari sudut pandang proses dan bagian-bagian memori pembentuknya) dapat digambarkan sebagai berikut:

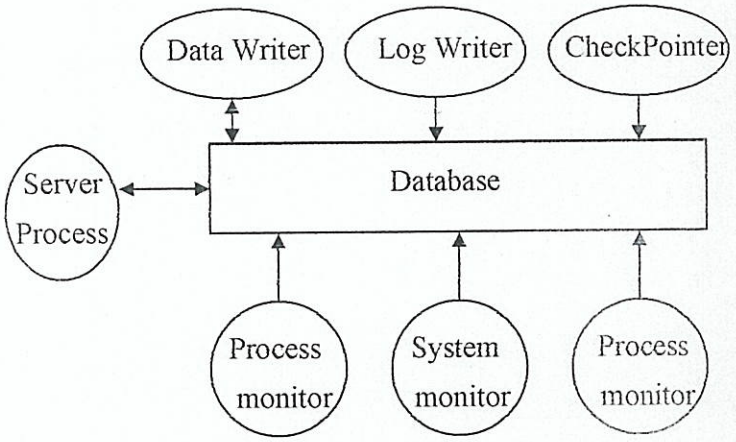


Gambar 3-2 Server process, Instance, dan Database

Pada Gambar 3-2, instance dan server process digambarkan sebagai operasi (notasi elips untuk diagram operasi), sementara database digambarkan sebagai tipe (notasi bujursangkar untuk diagram tipe).

Sebuah instance merupakan sebuah kumpulan background process. Nama-nama background process pembentuknya yang harus terdiri atas:

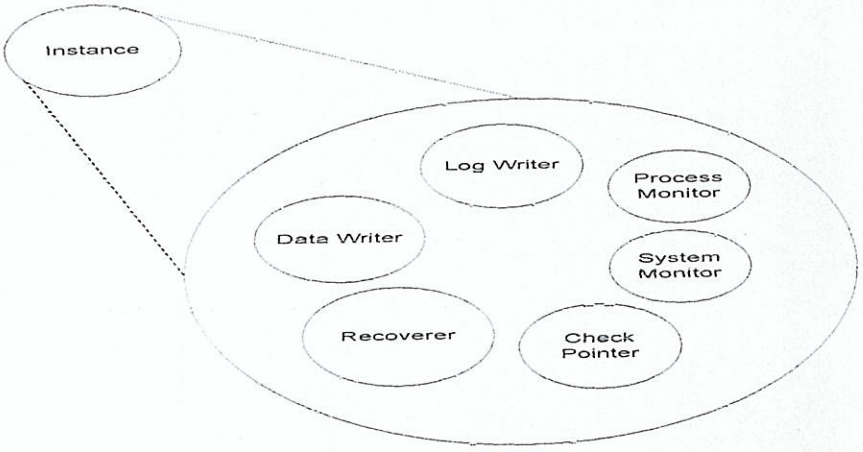
- **Checkpointer,** disingkat menjadi **ckpt.**
- **Data-writer** (walau banyak literatur Oracle lainnya menyebutnya database writer) yang disingkat **dbw0.**
- **Log-writer,** disingkat menjadi **prnon.**
- **System-monitor,** disingkat menjadi **smon.**
- **Recoverer**, disingkat menjadi **reco.**



Gambar 3-3 Enam proses/operasi utama pembentuk instance

Background process pembentuknya yang harus ada. Kata-kata harus ada perlu diberikan karena terdapat beberapa background process yang opsional untuk instance. Contoh background process yang opsional tersebut adalah archiver (biasa disingkat arc0). Archiver hanya ada dalam suatu instance jika instance tersebut mengoperasikan suatu database dalam mode archived log.

Untuk selanjutnya akan sangat berguna jika kita memandang instance sebagai suatu operasi. Gambar 3-4 dapat disebut sebagai **operation-hierarchy diagram** (bagian hierarki operasi). Bagan hierarki operasi mirip dengan DFD (Data Flow Diagram), namun bagan hierarki operasi menekankan hierarki antaroperasi, bukan aliran data.

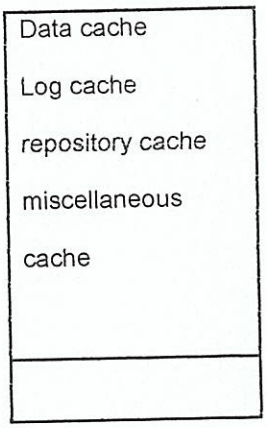


Gambar 3-4 Bagan grafis instance sebagai kumpulan operasi/proses

Namun instance bisa dipandang tak hanya sebagai sebuah operasi, melainkan juga sebagi sebuah tipe. Kita bisa menggambarkan instance sebagai berikut.

3.3.1.3 Instance Sebagai Suatu Tipe

Pada diagram tipe, kotak pertama berisi nama tipe, kotak kedua bagian-bagian tipe, kotak ketiga dari operasi-operasi eksternal yang bisa dilakukan ke objek dari tipe tersebut. Kita bisa memodelkan instance sebagai suatu tipe dengan bagian-bagian data cache, log cache, dan metadata cache.

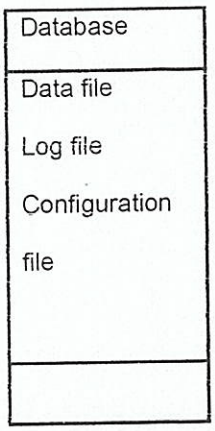


Gambar 3-5 Instance sebagai sebuah tipe

Pentingnya sisi pandang ini akan jelas tatkala kita mengkaji bagian-bagian dari database.

3.3.1.4 Database

Database pun dapat dimodelkan sebagai suatu tipe. Pada Subbab **3.3.1.2** instance dimodelkan sebagai suatu tipe dengan bagian-bagiannya di kotak kedua. Kita akan melakukan hal yang sama untuk database, dan melakukan bagian-bagiannya di kotak kedua. Hasilnya sebagai berikut,

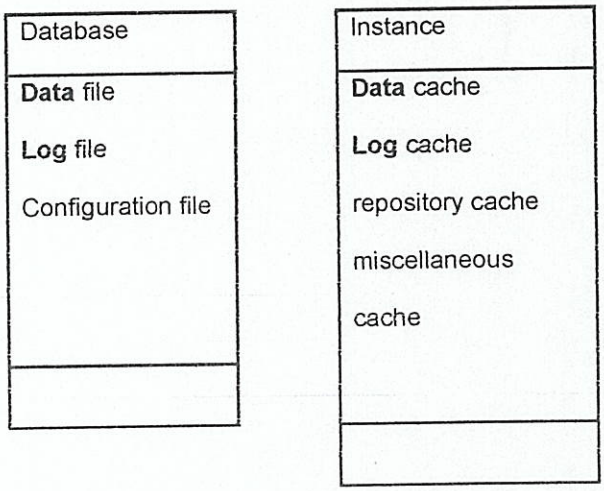


Gambar 3-6 database sebagai suatu tipe.

UNIT PELAKSANA TEKNIS PERPUSTAKAAN UNIKOM

3.3.1.5 Kaitan Database dengan Instance (1)

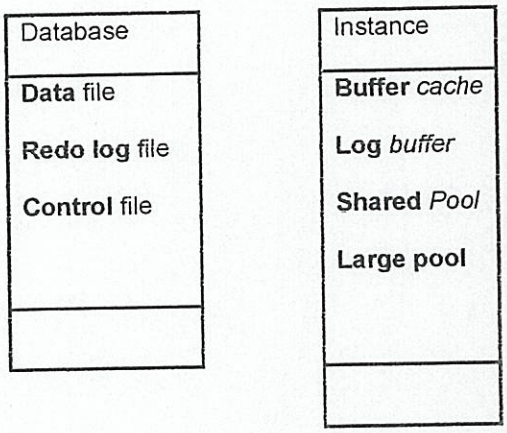
Gambar 3-7 menunjukkan keterkaitan erat antara database dengan instance.



Gambar 3-7 Kaitan database dengan instance

Data file adalah data yang disimpan di disk, sedangkan data cache adalah file data yang ada di memori. Log file merupakan (transaction) log yang berada didisk, sedangkan log cache berisi transaction log yang berada di memori. Configuration file pun pada dasarnya berisi metadata di disk, sedangkan metadata cache berisi metadata di memori.

Keterkaitannya yang (seharusnya) sangat mudah dilihat ini sayangnya dipersulit oleh Oracle Corp sendiri, karena pemakaian istilah-istilah yang 'agak' sembarangan. Berikut adalah istilah-istilah resmi dari Oracle (bandingkan dengan istilah-istilah yang penulis pilih).



Gambar 3-8 Istilah-istilah resmi Oracle

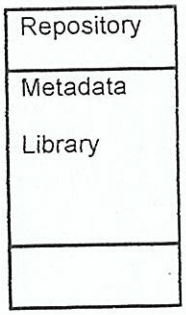
Untuk instance saja, lihat bahwa ada akhiran *cache*, ada akhiran *buffer*, dan ada akhiran *pool*. Ini tidak memperlihatkan kesamaan antara ketiganya, yakni bahwa sama-sama cache. Demikian pula, awalan instance berbeda dengan awalan di database. Awalan di database adalah data (untuk data file), sedangkan awalan di instance adalah buffer (untuk buffer cache).

Dalam buku ini, konsep-konsep akan dipaparkan dengan istilah-istilah yang 'tidak resmi' ini, namun implementasi akan memakai istilah-istilah resmi dari Oracle. Ini akan memampukan pengguna untuk membedakan konsep dengan implementasi. konsep-konsep yang dipaparkan di sini berlaku juga untuk DBMS lain (misal konsep tentang data cache juga bisa diterapkan di SQL Server), namun implementasi tak berlaku (SQL Server mungkin tak memakai istilah buffer cache).

3.3.1.6 Data, Log, dan Metadata

Tiga bagian ini merupakan bagian yang secara standar ada di hampir semua database, kita akan melihat esensi dari ketiganya untuk membantu melihat lebih jauh keterkaitan database dengan instance.

Secara ringkas, metadata + (programming) library = repository, suatu pengertian yang sangat umum dalam pemograman. Dengan diagram tipe, kita bisa menggambarkannya sebagai berikut.



Gambar 3-9 Repository

Kita abaikan bagian repository yang berupa library untuk membahas data, log, dan metadata: untuk mengkaji apa makna data, log, dan metadata.

Katakanlah kita memiliki suatu tabel T dengan satu kolom C, yang pada awalnya tidak berisi satu record pun. Andaikan pula bahwa tabel ini sudah dimasukkan ke memori. Pada saat itu. memori instance akan berisi:

Data cache *T*	**Log cache**	**Metadata cache** *DBA_tables*	
A		Table_name	Num_rows
		T	O

Gambar 3-10 Data, log, dan metadata sebelum pemasukkan record

Jiks kita masukkan satu record ke dalam T, dengan statement **INSERT INTO t VALUES (1).**

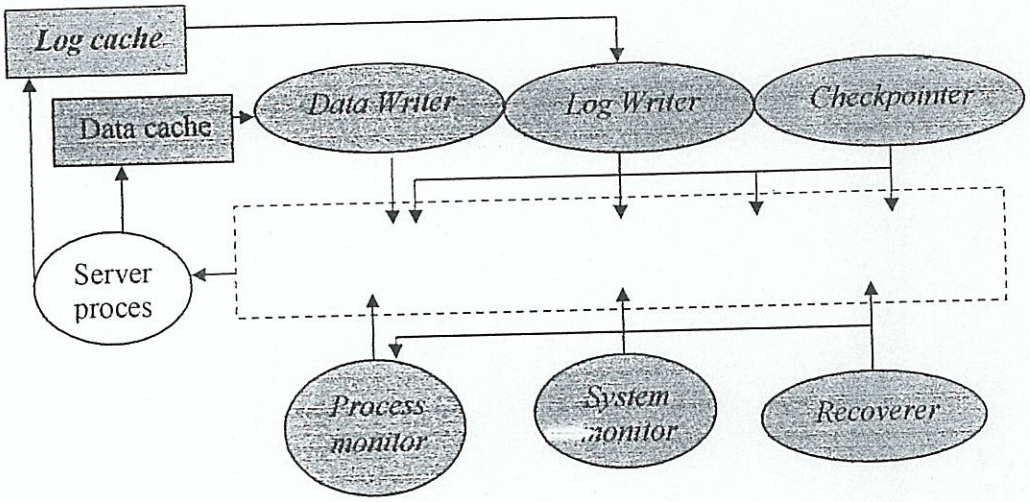
Data cache *T*	Log cache	Metadata cache *DBA_tables*	
A	INSERT INTO t VALUES (3) ;	Table_name	Num_rows
1		T	1

Gambar 3-11 Data, log, dan metadata usai pemasukkan record

Jelas bahwa data cache berisi data, log berisi transaction log, statement pemasukkan data (bukanlah datanya sendiri), dan metadata berisi data tentang data. Perubahan di data cache akan masuk ke data file, perubahan di log cache akan masuk ke log files.

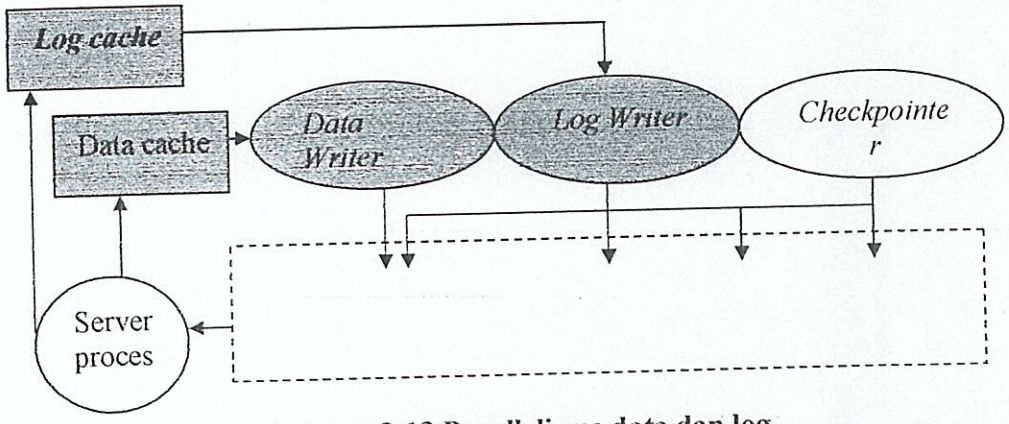
3.3.1.7 Kaitan database dengan instance (2)

Berbekal pembahasan di subbab-subbab sebelumnya, kita bisa membuat gambaran yang lebih baik tentang interaksi database dengan instance. Perlu diingat bahwa ini gambaran umum dan di sederhanakan.Dikarenakan lebih rumitnya mengenai interaksi instance dengan database sehingga dibuat gambaran sederhanaagar lebih mudah untuk dimengerti oleh pengguna baik dari tingkat pemula hingga tingkat menengah



Gambar 3-12 Kaitan database(dilingkupi garis putus-putus) dengan instance (diberi raster)

Hal menarik dari gambar di atas bisa kita peroleh dengan membuang beberapa bagian sehingga hasilnya adalah sebagai berikut:



Gambar 3-13 Parallelisme data dan log

Kita dengan mudah melihat parallelisme berikut:

Data	Log
Data cache	Log cache
Data file	Log file
Data writer	Log writer

Perhatikan juga bahwa baik data writer maupun log writer bekerja searah, membaca data hanya dari cache, dan menulis data hanya ke file. Pembacaan data file dan log file ke data cache dan log cache dilakukan oleh server proses, bukan instance.

3.3.2Mengenai Program

Program ini digunakan berdasar atas data yang diambil dari pasien yang akan masuk ke Rumah Sakit Hasan Sadikin.Hal ini akan lebih memudahkan di karenakan banyaknya pasien yang masuk.

Sebagai susunan dari cara memasukkan data ke dalam program sebagai berikut:

1. Masukkan nomor medical record (medrec), jika pasien sudah pernah mendaftar maka akan digunakan nomor medrec yang lama dengan cara mengklik tombol ambil kartu.

2. Kemudian masukkan biodata pasien berdasarkan kartu tanda pengenal (KTP/SIM/KTM).
3. Masukkan jenis penyakit pasien dari penjelasan pasien sendiri.
4. Kemudian menentukan cara bayar dari pasien apakah menggunkan askes atau bukan.
5. Setelah itu akan ditentukan jumlah yang harus dibayar oleh pasien.
6. Jika semua data telah dilengkapi, klik tombol ambil karcis untuk mencetak karcis yang kan digunakan sebagai nomor urut.
7. Jika pasien lama menginginkan untuk merubah data yang ada atau kehilangan kartu berobat, maka akan di data ulang kemudian klik tombol Daftar_ulang untuk mengganti data yang sudah ada.

Program ini menggunakan Oracle sebagai databse server di karenakan tingkat keamanannya yang lebih tinggi dari aplikasi database yang lainnya

COMPUTER CONTROLLED
NIKON