

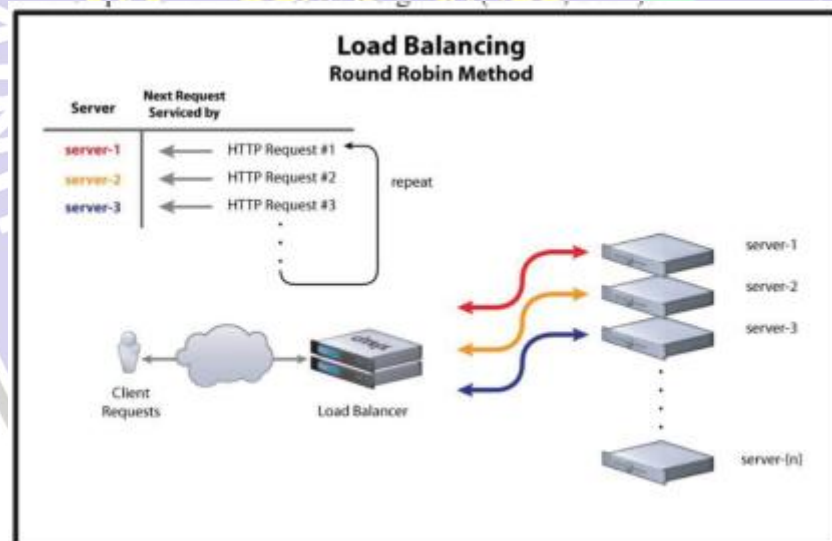
BAB II

TINJAUAN PUSTAKA

2.1 Algoritma Round Robin

Konsep dasar dari algoritma ini adalah dengan menggunakan time-sharing. Pada dasarnya algoritma ini sama dengan FCFS (*First-Come First-Serve*), hanya saja bersifat *preemptive*. Setiap proses mendapatkan waktu mesin yang disebut dengan waktu *quantum* (*quantum time*) untuk membatasi waktu proses, biasanya 1-100 milidetik. Setelah waktu habis, proses ditunda dan ditambahkan pada *ready queue*.

Algoritma *Round Robin* merupakan algoritma yang mudah dan banyak digunakan untuk perangkat *load balance*. Algoritma *Round Robin* bekerja membagi beban kerja komputer secara berurutan dan bergiliran dengan memanfaatkan *time sharing* atau paling sederhananya adalah memproses antrian secara bergiliran (Nasser and Witono, 2016)



Gambar 2.1 Proses Round Robin

2.2 Server

Komputer server adalah perangkat yang digunakan untuk mengelola segala aktivitas yang terjadi di dalam jaringan tersebut. Oleh karena itu, komputer ini memiliki berbagai fungsi, termasuk:

-) Menyediakan *database* atau *file* yang dapat digunakan bersama-sama oleh komputer *client*;
-) Melayani permintaan komputer *client* untuk menggunakan *database* atau *file* tersebut;
-) Mengatur lalu lintas transfer data atau file yang diminta komputer *client*;
-) Menyimpan data atau *file* yang dikirim oleh komputer *client*;
-) Mengatur hak akses data atau *file* dalam sebuah jaringan;
-) Melindungi komputer *client* dari *malware* dengan anti *malware* atau *firewall*.

2.3 Cluster Computer

Merujuk pada Greg Pfister “*cluster* adalah sebuah sistem paralel yang terdiri dari sebanyak komputer yang saling terhubung dan digunakan sebagai sumber daya komputasi tunggal (G. F. Pfister, 1996). *cluster* adalah konfigurasi sekumpulan *server independen*, muncul di jaringan sebagai satu mesin. Sekumpulan *server* ini dapat dikelola sebagai sistem tunggal, berbagi *namespaces* yang sama, dan dirancang khusus untuk menoleransi kegagalan komponen dan mendukung penambahan atau pengurangan komponen dengan cara yang transparan bagi pengguna.

Menurut (Vaugh, 2014) komponen-komponen pembentuk cluster computer terdiri dari:

-) *Shared Storage*
Adalah media tempat penyimpanan data yang bisa diakses bersama oleh anggota *cluster*
-) *Diffrent Network (publik dan private netowrok)*
Private network berguna dalam komunikasi antar anggota *cluster* dan publik *network* digunakan sebagai jalur data antara *cluster* dengan pengguna.
-) *Multipathing*
Pengenal LUN (*Logical Unit Number*) pada *SAN storage* untuk anggota *cluster* melalui path yang berbeda dan berfungsi untuk redundansi

) *Bonded Network Devices*

Teknologi yang memungkinkan lebih dari satu *Network Interface Card* (NIC) hanya dibaca sebagai satu NIC saja dengan keperluan untuk menghindari kegagalan dari satu NIC bisa menggunakan NIC saja, sehingga penerima/pengiriman paket masih bisa berfungsi.

2.3.1 *Tipe Cluster*

Menurut (Ault and Tumma, 2004), *cluster computer* bisa digolongkan menjadi lima kategori, yaitu:

-) *Fail Over Cluster*, beberapa komputer bekerja sama untuk ketersediaan *resource* apabila terjadi kerusakan dari salah satu komputer sehingga meminimalisir terjadinya *downtime*.
-) *High Performance Cluster*, biasanya di level *middle ware* seperti database paralel. Keuntungan dari *cluster* ini adalah database bisa di buat untuk kebutuhan *load balance* dan *fail over*
-) *Application Cluster*, beberapa komputer bekerja sama dalam mengerjakan di level aplikasi *server* yang membutuhkan banyak *resource*

2.4 *Domain Name System (DNS)*

DNS (*Domain name system*) adalah fitur yang disediakan oleh sistem operasi untuk mengubah alamat/*domain* website ke alamat IP. Saat membuka *website*, yang ditulis merupakan alamat dari *website* tersebut yang sebetulnya adalah alamat IP merupakan alamat IP. Sebagai contoh yaitu ketika sedang mengakses www.google.com maka yang sebenarnya dituju adalah alamat IP dari google, yaitu 173.194.38.146. Alamat IP ini bisa diketahui dengan cara mengetikkan perintah *ping google.com* di aplikasi CMD, maka ketika komputer sudah terkoneksi dengan internet terlihat alamat IP dari *google.com*. ((Komputer, 2013).

2.5 RDBMS Oracle

Database adalah sekumpulan data yang yang disimpan dalam komputer secara sistematis sehingga dapat diakses menggunakan suatu program komputer untuk memperoleh informasi dari data tersebut (Kusrini, 2007).

RDBMS Oracle dapat mengelola data dalam ukuran besar dan dapat dikelola oleh banyak *user* sehingga memungkinkan beberapa *user* dapat mengakses data yang sama secara bersamaan. Pada saat yang bersamaan, *Oracle* dapat mencegah akses dari *user* yang tidak berhak dan menyediakan solusi yang efisien untuk *recovery* kegagalan penyimpanan data.

User dapat terhubung dengan *server Oracle* dengan 3 cara :

1. *User log on* ke dalam sistem operasi yang menjalankan *instance Oracle* dan menjalankan aplikasi atau *tool* yang mengakses database pada sistem tersebut.
2. *User* menjalankan aplikasi atau *tool* pada komputer lokal dan terhubung dengan komputer lain yang mempunyai database *Oracle* melalui jaringan/*network*. Konfigurasi ini disebut *client/server*.

Arsitektur sistem database *client/server* terdiri dari 2 bagian, yaitu : *front end (client)* dan *back end (server)* yang terhubung dengan jaringan.

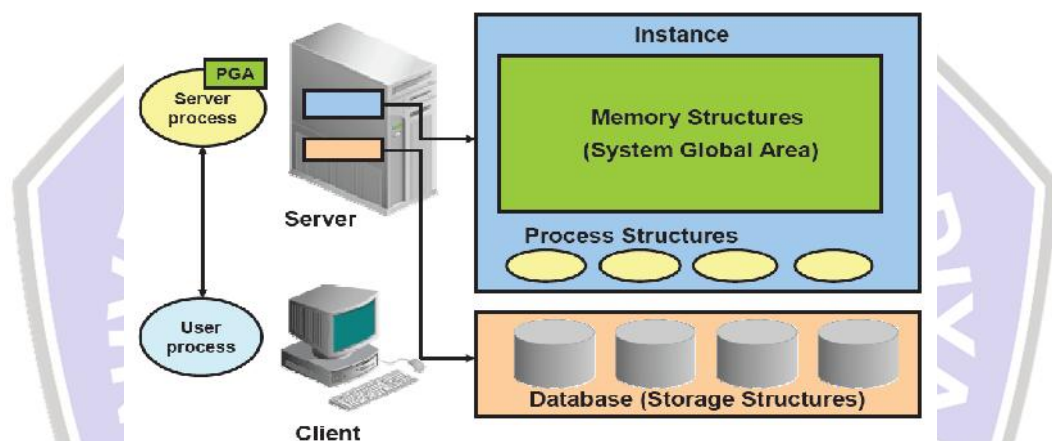
- *Client* adalah aplikasi database yang menginisiasi *request* sebuah operasi untuk dijalankan pada database. Beberapa *client* dapat berjalan secara simultan dan mengakses satu *server* yang sama.
- *Server* menjalankan *software* database *Oracle* dan menangani fungsi-fungsi yang membutuhkan akses data secara bersamaan. *Server* menerima dan memproses permintaan yang berasal dari *client*.

User mengakses aplikasi pada *server* melalui *tool* (misal : web browser) pada komputer lokal (*client*). *Server* aplikasi kemudian berinteraksi dengan *server back-end* database dengan bertindak sebagai *client*.

Terdapat 3 struktur penting pada arsitektur database server *Oracle*, yaitu : *memory structures*, *process structures*, dan *storage structures* (database). Sedangkan sistem database *Oracle* terdiri dari database *Oracle* dan *instance*

database. Database terdiri dari struktur *logic* dan fisik yang terpisah sehingga media penyimpanan fisik dapat dikelola tanpa mempengaruhi akses ke struktur *logic*.

Instance database terdiri dari *memory structures* dan *process structures*. Tiap kali *instance* dijalankan, *shared memory* yang disebut *System Global Area (SGA)* dialokasikan dan *background* proses berjalan. Setelah menjalankan *instance* database, Oracle mengasosiasikan *instance* tersebut dengan database tertentu. Hal ini disebut *mounting* database. Setelah itu database siap untuk dibuka, yang memungkinkan database tersebut dapat diakses oleh *user* yang telah diberi otorisasi.



Gambar 2.2 Arsitektur RDBMS Oracle

2.6 Oracle RAC

2.6.1 Arsitektur Oracle RAC

Pada arsitektur tradisional *symmetric multiprocessing* (SMP), setiap *node* memiliki memori sistem serta sistem operasinya sendiri, database *instance*, dan perangkat lunak aplikasi. Oracle RAC, sebaliknya, menggunakan teknik yang disebut *Cache Fusion* untuk membuat *cache virtual* tunggal di semua *node cluster* (Aldila C, Yuwono and Asmara, 2011)(Oracle and Paper, 2001).

Node berbagi akses ke subsistem penyimpanan dan sumber daya yang mengelola data, tetapi tidak secara fisik berbagi memori utama. Meminimalkan I/O disk karena permintaan basis data apa pun dapat dilayani oleh node mana pun. Ini juga menyederhanakan tugas konfigurasi, karena

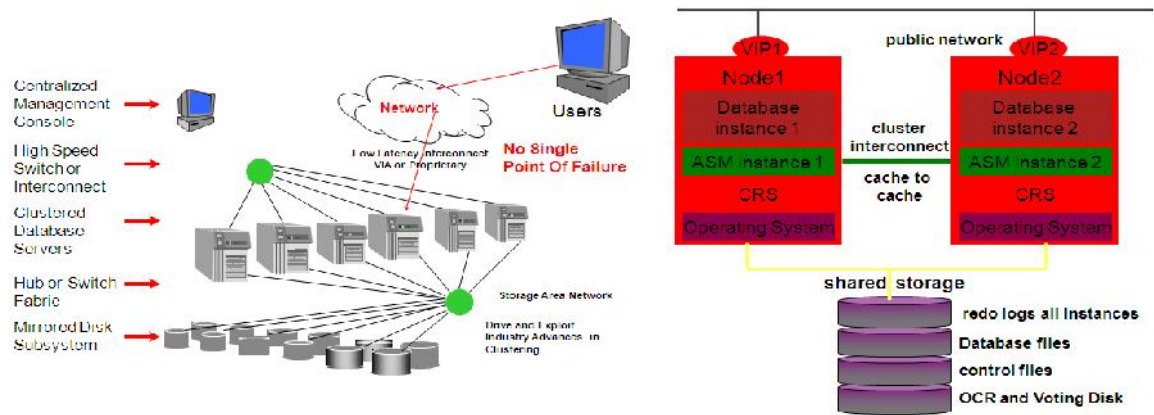
aplikasi tidak harus ditulis ulang, didesain ulang, atau diprogram ulang untuk berjalan secara bersamaan di beberapa server. Oracle9i mengenali aplikasi dan secara transparan menyeimbangkan beban di semua node.

Karena fleksibilitas ini, Oracle RAC sangat ideal untuk aplikasi yang menuntut yang membutuhkan skalabilitas di luar sistem multiproses simetris *high-end* tunggal. Teknologi ini memungkinkan aplikasi database tumbuh dan menyusut secara transparan saat kapasitas pemrosesan dan penyimpanan ditambahkan atau ditarik, tanpa mengorbankan ketersediaan. Ini juga memungkinkan semua jenis aplikasi untuk diskalakan tanpa harus disesuaikan untuk lingkungan pengelompokan.

Arsitektur Cache Fusion menggunakan *cache* yang saling berhubungan dari semua *node* dalam *cluster* untuk memenuhi permintaan database. Permintaan kueri dapat dipenuhi oleh *cache* lokal atau *cache* lainnya. Operasi pembaruan tidak memerlukan operasi tulis dan baca disk yang berurutan untuk sinkronisasi, karena node lokal dapat memperoleh blok yang diperlukan secara langsung dari *cache* database mana pun.

Cache Fusion memanfaatkan protokol interkoneksi *cluster* latensi rendah untuk mengirim blok data dari *cache* node jarak jauh ke *cache* lokal. Ini menghapus operasi disk lambat dari jalur kritis sinkronisasi ruas. Akses disk dilakukan hanya ketika transaksi pembaruan dilakukan, membutuhkan jaminan penulisan disk.

Karena implementasi *Cache Fusion* lengkap di Oracle menghilangkan latensi yang terkait dengan koordinasi *cache* berbasis disk, aplikasi dapat diskalakan secara efektif tanpa harus "mewaspada *cluster*". Dalam istilah praktis, ini memungkinkan perusahaan untuk menskalakan tingkat basis data secara horizontal sebagai respons terhadap peningkatan penggunaan dan permintaan.



Gambar 2.3 Arsitektur Real Application Cluster

RAC terdiri dari beberapa *node (server)*, terhubung satu sama lain oleh interkoneksi pribadi. File database disimpan pada subsistem penyimpanan bersama, di mana mereka dapat diakses oleh semua *node*. Dan setiap *node* memiliki koneksi jaringan *publik*. Dalam hal perangkat lunak dan konfigurasi, RAC memiliki tiga komponen dasar: perangkat lunak cluster, perangkat lunak database, dan metode pengelolaan subsistem penyimpanan bersama seperti yang ditunjukkan pada gambar 2.1.

1. Perangkat lunak cluster dapat disediakan oleh vendor atau disediakan oleh Oracle, tergantung pada *platform*. Layanan Siap *Cluster*, atau CRS, adalah fitur baru dalam 10g. Dimana vendor *clusterware* digunakan, CRS berinteraksi dengan vendor *clusterware* untuk mengkoordinasikan informasi keanggotaan *cluster*; tanpa *vendor clusterware*, CRS, yang juga dikenal sebagai Oracle OSD *Clusterware*, menyediakan manajemen *cluster* yang lengkap.
2. Perangkat lunak database adalah Oracle 11R2 dengan opsi RAC.
3. Terakhir, subsistem penyimpanan bersama dapat dikelola dengan salah satu opsi berikut: perangkat mentah; Manajemen Penyimpanan Otomatis (ASM); Sistem file *cluster* (CFS) yang disediakan vendor, Sistem File *Cluster Oracle* (OCFS), atau manajer volume *logis* (LVM) yang disediakan *vendor*; atau *Networked File System* (NFS) pada perangkat *Network Attached Storage* (NAS) bersertifikat.

2.6.2 Storage (Penyimpanan)

Sebuah database *single-instance* biasa memiliki tiga tipe dasar file: *software* database dan *file dump*; file data, *spfile*, file kontrol dan *file log*, sering disebut sebagai "file database"; dan mungkin memiliki file pemulihan, jika menggunakan RMAN. Database RAC memiliki jenis file tambahan yang disebut sebagai "file CRS". Ini terdiri dari *Oracle Cluster Registry* (OCR) dan *disk voting*.

Tabel 2.1 Opsi Storage untuk Kebutuhan *Sharing Storage*

<i>Abbrev</i>	<i>Storage Option</i>
Raw	<i>Raw Devices, no file system</i>
ASM	<i>Automatic Storage Management</i>
CFS	<i>Cluster File System</i>
OCFS	<i>Oracle Cluster File Syatem</i>
LVM	<i>Logical Volume Manager</i>
NFS	<i>Network File System (SAN or NAS)</i>

Tidak semua file ini harus berada di subsistem penyimpanan bersama. File database dan file CRS harus dapat diakses oleh semua instance, jadi harus berada di subsistem penyimpanan bersama. Perangkat lunak basis data dapat berada di subsistem bersama dan dibagi di antara *node*; atau setiap *node* dapat memiliki *ORACLE_HOME* sendiri. Area pemulihan *flash* harus dibagikan oleh semua instans, jika digunakan.

Beberapa opsi penyimpanan tidak dapat menangani semua jenis file ini. Untuk mengambil contoh yang jelas, perangkat lunak database dan *file dump* tidak dapat disimpan pada perangkat mentah. Ini tidak penting untuk *file dump*, tetapi itu berarti bahwa memilih perangkat mentah menghalangi memiliki *ORACLE_HOME* bersama di perangkat penyimpanan bersama.

2.6.3 Automatic Storage Management (ASM)

ASM sedikit mirip dengan manajer *volume logis* dan memberikan banyak manfaat dari LVM. Tetapi ini juga memberikan manfaat yang tidak

dimiliki LVM: *striping/mirroring* tingkat file, dan kemudahan pengelolaan. Alih-alih menjalankan perangkat lunak LVM, Anda menjalankan *instance* ASM, jenis "*instance*" baru yang sebagian besar terdiri dari proses dan memori dan menyimpan informasinya dalam disk ASM yang dikelolanya.

