Tahap 2 – Mode Kluster Server

Tugas Besar Sistem Paralel dan Terdistribusi – IF3230

**DOKUMENTASI**



**Oleh**

Kelompok K2-3 – Moe Dikra Prasetya

Riandy Rahman Nugraha (13511014)

Tito D Kesumo Siregar (13511018)

M Dikra Prasetya (13511030)

Sekolah Teknik Elektro dan Informatika - Institut Teknologi Bandung

Jl. Ganesha 10, Bandung 40132

1. Eksekusi File

File yang telah dibuat untuk pengumpulan tahap 2 adalah:

a. Client.java, berisi implementasi klien sederhana untuk terkoneksi ke server.

b. Server.java, berisi kode server.

c. ServerThread.java, berisi kode thread yang dijalankan oleh server.

Untuk menjalankan Client, user diharapkan untuk memasukkan dua buah argument yaitu <nama-host> dan <port>. Kedua hal ini dibutuhkan untuk deklarasi Socket ke Server. Sedangkan untuk Server hanya membutuhkan satu buah argument yaitu <port>. Koneksi dari Client dan Server menggunakan kelas Socket yang disediakan oleh java.net.\*.

Percobaan telah dilaksanakan di sebuah komputer desktop Windows 7 dan sebuah komputer laptop Windows 8, dengan klien dijalankan di komputer desktop dan server dijalankan di kedua komputer.

2. Pendistribusian Data

Untuk mendistribusikan data, dilakukan perhitungan hash key dari key dengan menggunakan nilai hashCode() dari Java. Kemudian, dihitung token server yang harus mendistribusikan data, yaitu hash modulo jumlah server. Setelah itu, berdasarkan nilai tersebut data dikirim ke server yang seharusnya menyimpan data.

Untuk mengambil data, server mengontak semua server teman dan mengumpulkan data-data yang ada, kemudian diproses (di-sort) dan diambil data dengan timestamp terakhir, baru dikirimkan kembali ke client.

Dengan kata lain, server memiliki token berupa urutan kehadirannya, dengan toke berupa angka 0..N-1, di mana N adalah banyaknya server. Kemudian, server 0 akan mengurus data 0, N, 2N, 3N, ..., server 1 akan mengurus data 1, N+1, 2N+1, 3N+1, ..., begitu seterusnya hingga server N-1 akan mengurus data N-1, 2N-1, 3N-1, .... Adapun nilai maksimum hash, karena menggunakan implementasi hashCode() Java, terbatas di antara 0..231-1 (maksimum nilai signed integer).

Keuntungan dari metode pendistribusian data ini adalah:

* User dapat mengirimkan data cukup ke salah satu server. Server akan mengontak server teman yang seharusnya menyimpan data.
* User dapat cukup mengontak salah satu server untuk mendapatkan semua server.

Kelemahan dari metode pendistribusian data ini adalah:

* Dapat terjadi unbalanced load jika hasil perhitungan hash membuat satu server menyimpan data terus-menerus. Diasumsikan bahwa hashCode() yang digunakan dapat mendistribusikan data dengan baik.
* Dapat terjadi unbalanced load jika data dengan key yang sama disimpan terus-menerus.

3. Penyimpanan Data

Implementasi penyimpanan data pada program ini dilakukan di dalam file teks. Implementasi struktur yang digunakan untuk menyimpan tabel adalah berupa HashMap. Dimana nama tabel adalah key dari HashMap dan arraylist yang berisi data-data tabel adalah value dari HashMap tersebut. Berikut ini adalah struktur dari HashMap untuk menyimpan tabel tersebut:

*HashMap<String, ArrayList<String>>*

Insert elemen dari arraylist(yang merupakan representasi data dari sebuah tabel) dilakukan secara LIFO. Hal ini ditujukan agar elemen dengan timestamp terbaru selalu berada di posisi teratas sehingga pencarian data dengan timestamp terbaru dapat dilakukan dengan lebih efisien.

Saat dilakukan perintah display maka server akan menelusuri seluruh elemen yang terdapat pada arraylist dan menyimpan key elemen tersebut dalam sebuah string yang berisi kumpulan key-key mana saja yang sudah diakses. Jika key elemen yang dilewati oleh server sudah pernah diakses maka elemen tersebut tidak akan dimasukkan sebagai reply untuk client, namun jika key dari elemen tersebut berlum terdaftar di string yang berisi kumpulan key maka elemen tersebut akan dijadikan reply untuk client. Berikut ini adalah struktur dari elemen arraylist (berupa string):

*<Key, Value, Timestamp>*

Berikut adalah contoh isi file teks yang ditulis oleh server:

*senjata*

*3*

*<mp44, 7.92, 2014-05-10 15:44:47.655>*

*<m16, 5.56, 2014-05-10 15:44:35.351>*

*<m4, 7.62, 2014-05-10 15:44:29.23>*

*kucing*

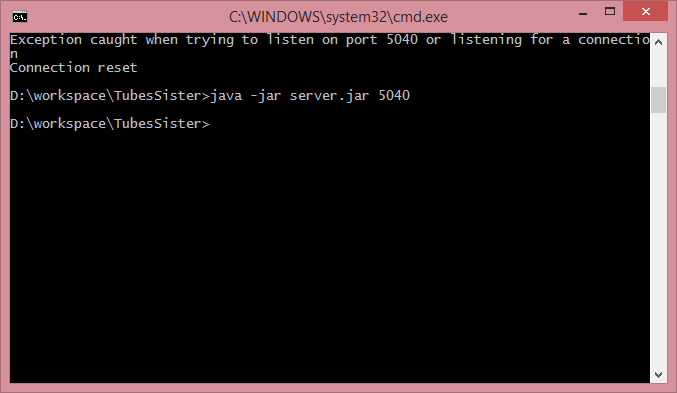
*2*

*<persia, prince, 2014-05-10 15:43:46.327>*

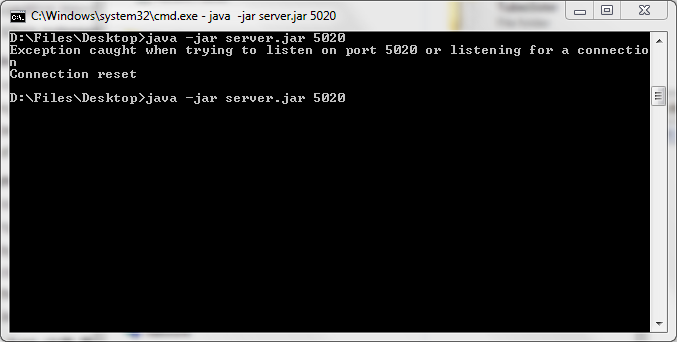
*<angora, miaw, 2014-05-10 15:43:26.063>*

Karena penambahan elemen dilakukan secara LIFO maka sudah pasti elemen yang ditemukan pertama kali adalah elemen dengan timestamp terbaru.

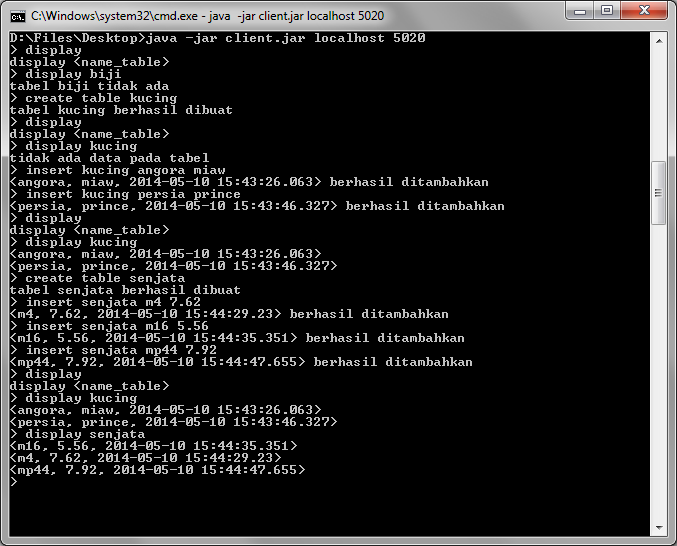
4. Screenshot



Tampilan server pertama



Tampilan server kedua



Tampilan klien