Map Reduce

Big Data
STF1724

Rini Nuraini Sukmana, M.T
0020087901 - 08882024236
rini.nuraini@usbypkp.ac.id

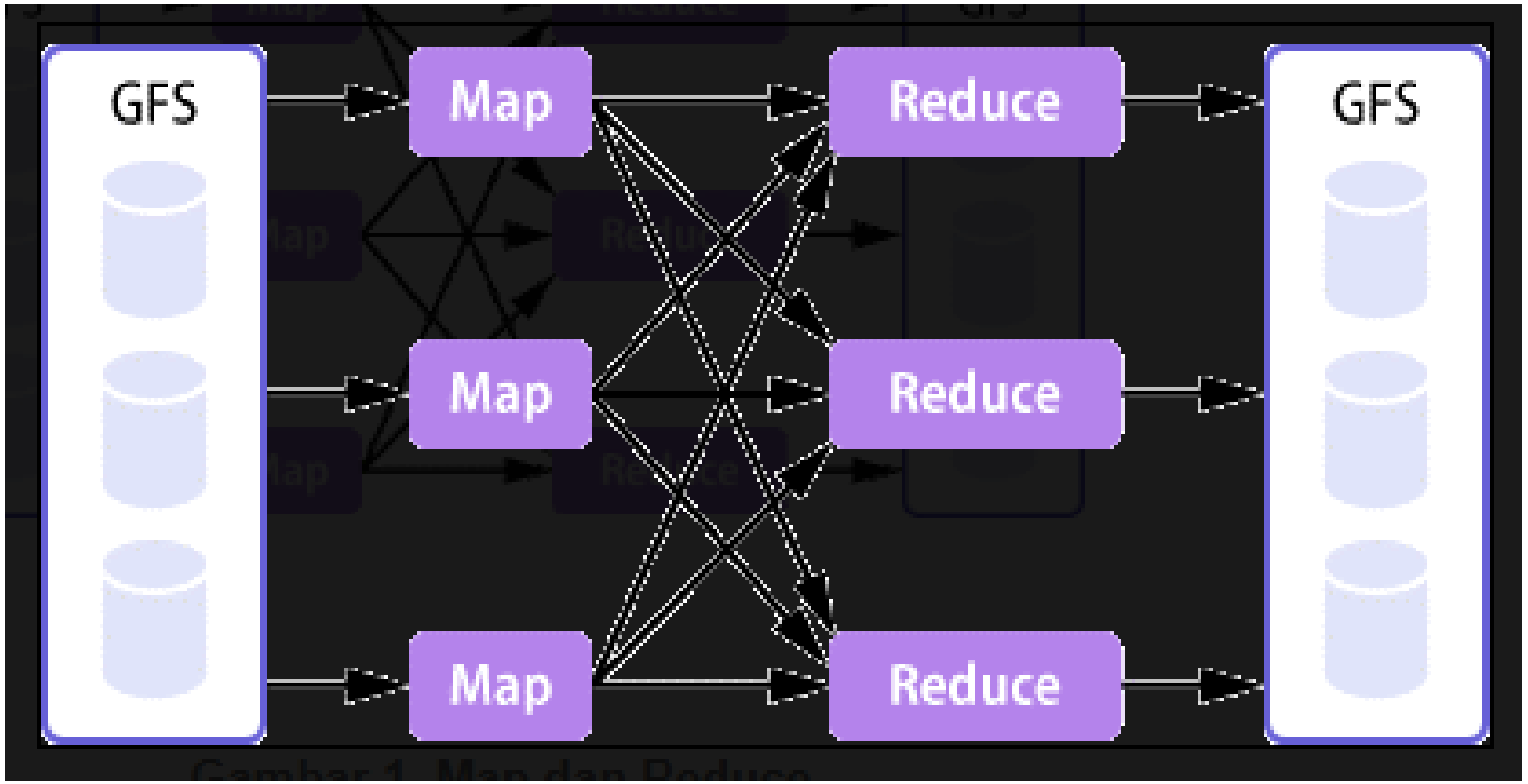
Definisi Map Reduce

- MapReduce adalah model pemrograman rilisan Google yang ditujukan untuk memproses data berukuran raksasa secara terdistribusi dan paralel dalam cluster yang terdiri atas ribuan komputer.
- Dalam memproses data, secara garis besar MapReduce dapat dibagi dalam dua proses yaitu proses Map dan proses Reduce.
- Kedua jenis proses ini didistribusikan atau dibagi-bagikan ke setiap komputer dalam suatu cluster (kelompok komputer yang saling terhubung) dan berjalan secara paralel tanpa saling bergantung satu dengan yang lainnya.
- Proses Map bertugas untuk mengumpulkan informasi dari potongan-potongan data yang terdistribusi dalam tiap komputer dalam cluster.
- Hasilnya diserahkan kepada proses Reduce untuk diproses lebih lanjut.
- Hasil proses Reduce merupakan hasil akhir yang dikirim ke pengguna.

Desain dan Struktur MapReduce

- Untuk memproses sebuah data raksasa, data itu harus dipotong-potong kemudian dibagi-bagikan ke tiap komputer dalam suatu cluster.
- Lalu proses Map dan proses Reduce pun harus dibagi-bagikan ke tiap komputer dan dijalankan secara paralel.
- Hasil akhirnya juga disimpan secara terdistribusi.
- MapReduce telah didesain sangat sederhana, untuk menggunakan MapReduce,
  - seorang programer cukup membuat dua program yaitu program yang memuat kalkulasi atau prosedur yang akan dilakukan oleh proses Map dan Reduce. Jadi tidak perlu pusing memikirkan bagaimana memotong-motong data untuk dibagi-bagikan kepada tiap komputer,
  - dan memprosesnya secara paralel kemudian mengumpulkannya kembali.
  - Semua proses ini akan dikerjakan secara otomatis oleh MapReduce yang dijalankan diatas Google File System (GFS).

Desain dan Struktur MapReduce



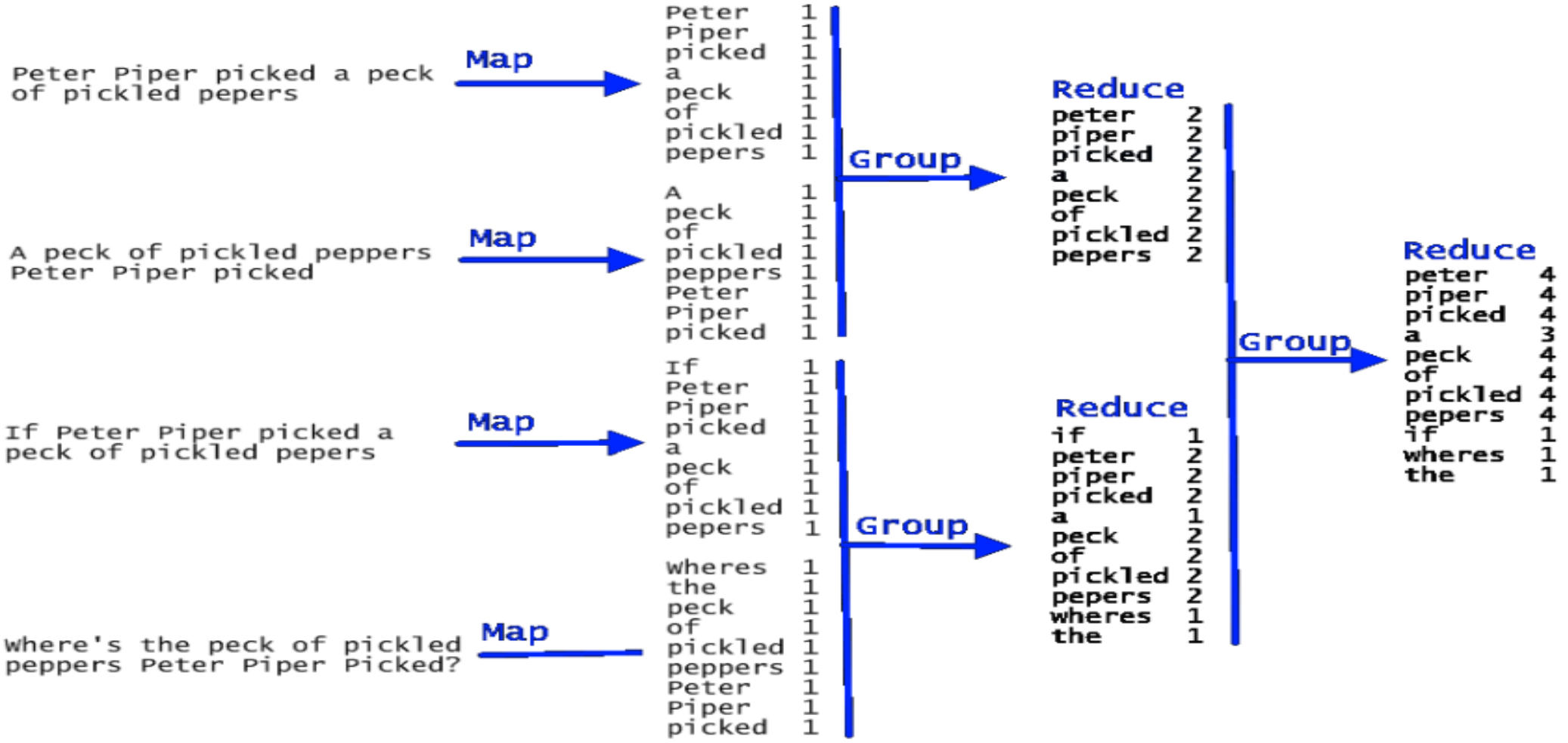
Desain dan Struktur MapReduce

- Program yang memuat kalkulasi yang akan dilakukan dalam proses Map disebut Fungsi Map, dan yang memuat kalkulasi yang akan dikerjakan oleh proses Reduce disebut Fungsi Reduce. Jadi, seorang programmer yang akan menjalankan MapReduce harus membuat program Fungsi Map dan Fungsi Reduce.
- Fungsi Map bertugas untuk membaca input dalam bentuk pasangan Key/Value, lalu menghasilkan output berupa pasangan Key/Value juga.
- Pasangan Key/Value hasil fungsi Map ini disebut pasangan Key/Value intermediate. Kemudian, fungsi Reduce akan membaca pasangan Key/Value intermediate hasil fungsi Map, dan menggabungkan atau mengelompokkannya berdasarkan Key tersebut. Lain katanya, tiap Value yang memiliki Key yang sama akan digabungkan dalam satu kelompok. Fungsi Reduce juga menghasilkan output berupa pasangan Key/Value.

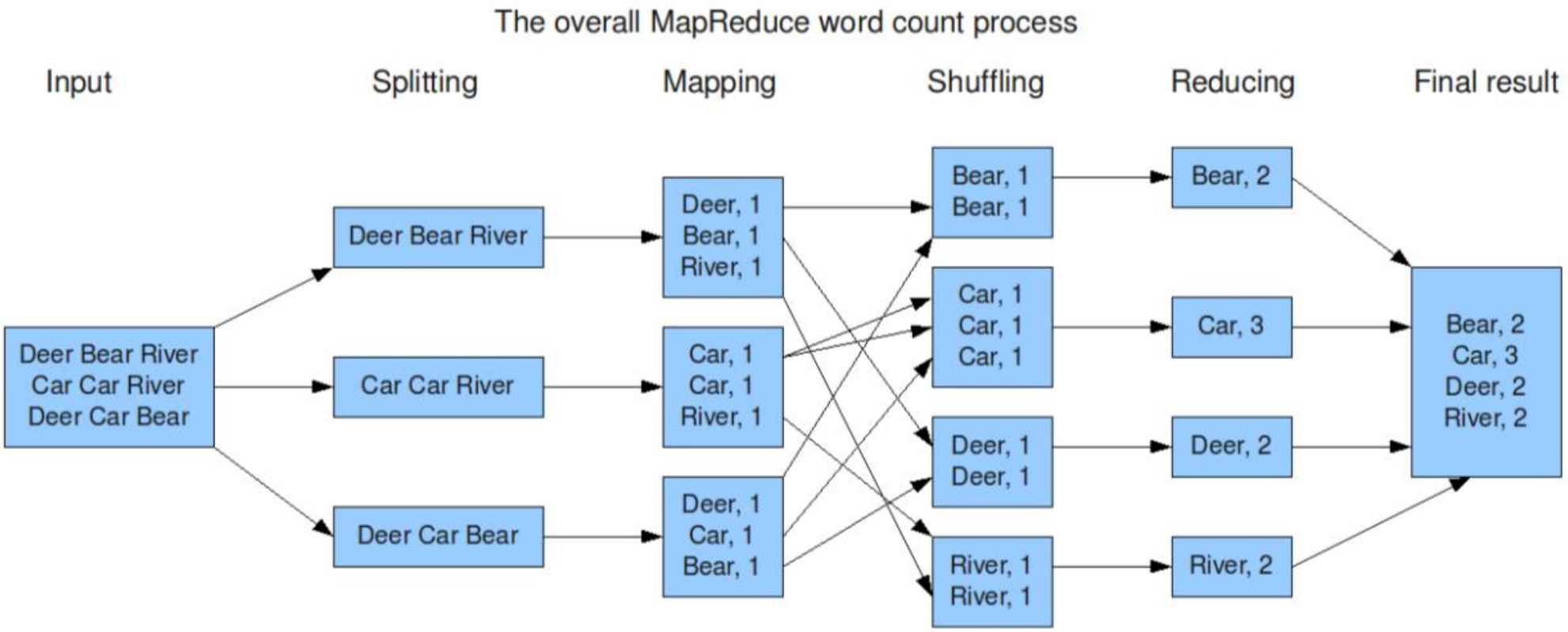
Contoh program MapReduce untuk *menghitung jumlah tiap kata* dalam beberapa file teks yang berukuran besar :

```
map(String key, String value):
        //key : nama file teks.
        //value: isi file teks tersebut.
        for each word W in value:
            emitIntermediate(W,"1");
    reduce(String key, Iterator values):
        //key : sebuah kata.
        //values : daftar yang berisi hasil hitungan.
        int result = 0;
        for each v in values:
            result+=ParseInt(v);
        emit(AsString(result));
```

Hasil akhir dari program ini adalah jumlah dari tiap kata yang terdapat dalam file teks yang dimasukkan sebagai input program ini.



Langkah-langkah Map reduce (studi kasus: Word Count)



Langkah-langkah Map reduce (studi kasus: Word Count)

1. **Pemecahan data masukan (*Splitting)*.** Pada proses ini data masukan yang diberikan oleh pengguna MapReduce (klien) akan dipecah menjadi bagian-bagian yang lebih kecil.
2. ***Mapping*.** Mapping adalah salah satu tahap terpenting dari MapReduce. Pada fase Mapping, bongkahan data yang telah dipecah akan di proses untuk menghasilkan *intermediary key-value pairs*. Pada contoh wordcount (Gambar 1) diatas, data yang mengandung "Dear Bear River" akan diproses sehingga menghasilkan pasangan key-value Dear:1, Bear:1, dan River:1.
3. **Shuffling.** Sebelum fase *reduce*, fase *shuffling* bertugas untuk mengumpulkan satu atau lebih *key* yang berbeda di sebuah mesin tertentu agar aggregasi dapat dilakukan dengan mudah. Pada contoh di atas, seluruh kata ***Bear*** yang dihasilkan fase *mapping* akan berada dalam sebuah mesin yang sama. Begitu juga dengan kata-kata lain.
4. **Reducing.** Fase *reducing* bertugas untuk melakukan aggregasi terhadap seluruh pasangan intermediary key-value dengan key yang sama. Pada gambar diatas, pasangan *key-value* Bear:1 dan Bear:1 akan diaggregasi oleh reducer sehingga pada akhirnya reducer akan menghasilkan keluaran Bear:2.