**PARALLELISASI ALGORITMA K-NEAREST NEIGHBOR MENGGUNAKAN MPI**

Laporan

diajukan untuk memenuhi tugas mata kuliah Sistem Terdistribusi

oleh :

Dwinov Satrio Wibowo (1301160787)

Rere Gilang Nuri Auladi (1301160801)

Brenda Irena (1301164641)

Husnul Khotimah Farid (1301162742)



**IFIK – 40 - 03**

**PROGRAM STUDI S1 INFORMATIKA**

**FAKULTAS INFORMATIKA**

**UNIVERSITAS TELKOM**

**BANDUNG**

**2019**

**ANALISIS MASALAH**

*K-Nearest Neighbor* adalah salah satu teknik klasifikasi yang sering digunakan dalam pengklasifikasian data karena metode ini memiliki konsistensi yang kuat, dengan cara mencari kasus dengan menghitung kedekatan antara kasus baru dengan kasus lama berdasarkan pencocokkan bobot.

Namun, semakin banyak data yang diolah, maka semakin lama juga waktu yang diperlukan untuk mengeksekusi data karena proses dikerjakan secara serial dimana proses selanjutnya akan dikerjakan setelah proses sebelumnya selesai. Waktu eksekusi dari metode pengklasifikasian *K-Nearest Neighbor* dapat dipercepat dengan menggunakan pemograman parallel dengan menggunakan MPI.

*Message Passing Interface (MPI)* merupakan *message passing* yang telah di standarisasi dan portable yang didesain oleh suatu grup peneliti yang memiliki fungsi pada berbagai arsitektur atau penggunaan *parallel computing.* MPI menggunakan syntax dan inti library yang berguna user yang menggunakan *message passing* program pada platform Bahasa C, C++, Fortran, dan juga Python.

**METODE PENYELESAIAN MASALAH**

Pertama, kami lakukan analisis terhadap algoritma program serial terlebih dahulu untuk memastikan apakah program tersebut dapat di parallelkan atau tidak. Setelah itu, kita tentukan metode apa yang akan digunakan, pada kasus ini kita menggunakan metode *MPI All Reduce* dimana disini kita harus menghitung pembagian porsi pengerjaan *looping* pada bagian *looping* KNN yang telah ditentukan untuk diparallelkan.

Bagian yang kami parallelkan pada program KNN ini adalah perulangan pengerjaan fungsi *K-Nearest Neighbor* sebanyak data test pada fungsi *cross validation* untuk mencari nilai k terbaik.

Sebelum di parallelkan :

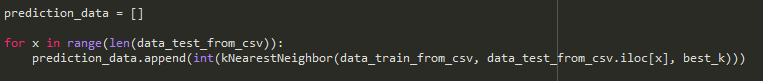


Setelah di parallelkan :

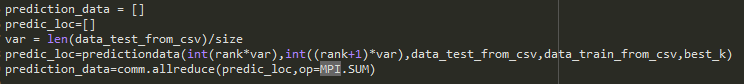


Selain itu juga pada bagian main untuk menentukan kelas dari setiap data pada file datatest dimana bagian yang diparallelkan ditunjukan pada algoritma yang dilingkari dibawah ini.

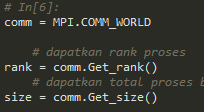
Sebelum di parallelkan :



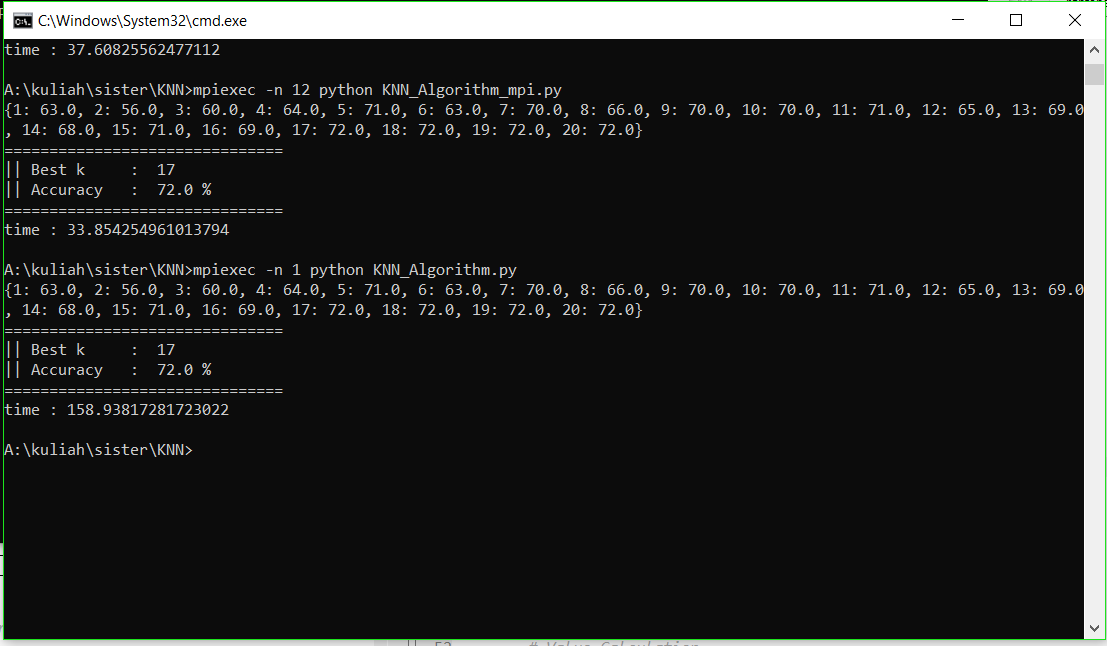
Setelah di parallelkan :



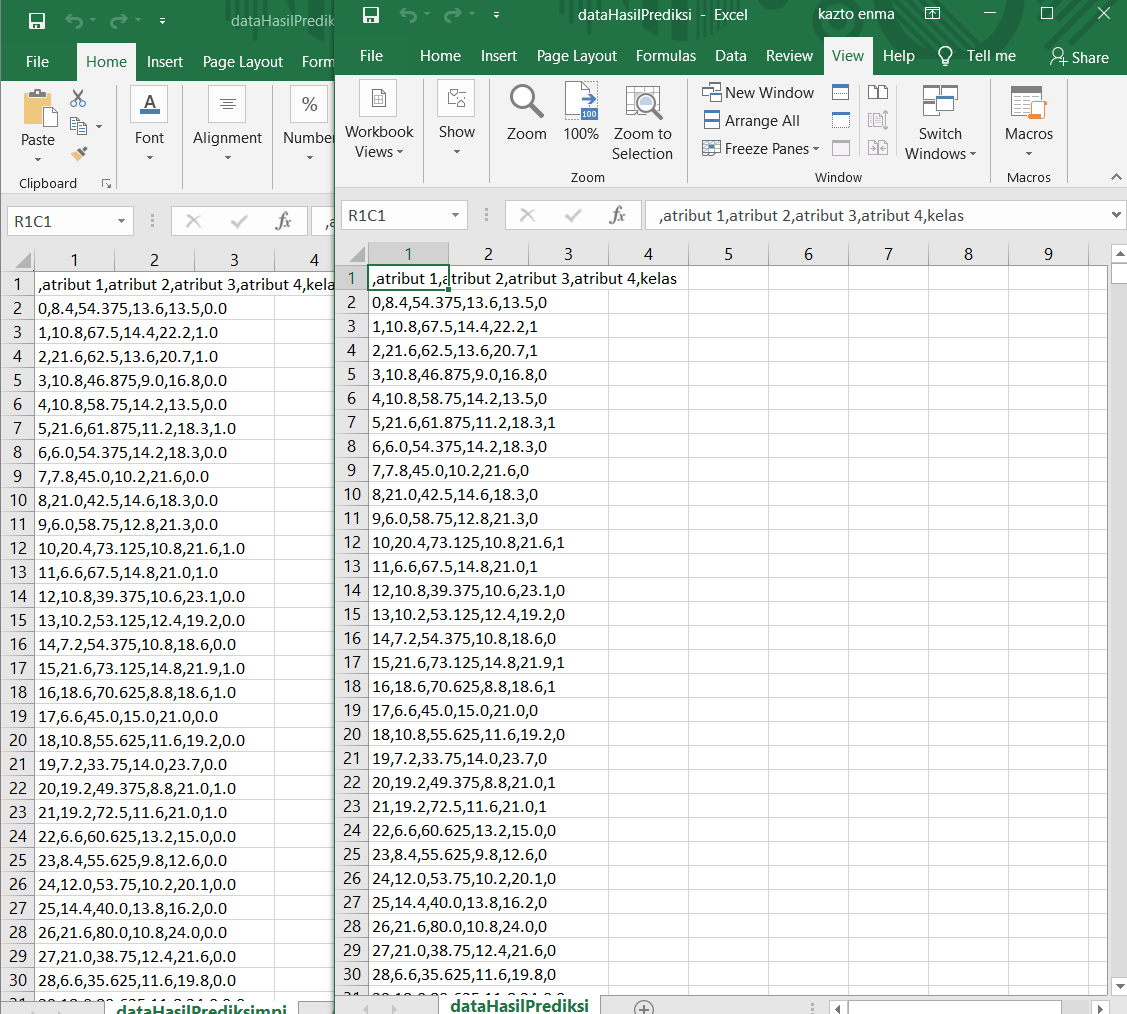
Sebelum kita memparallelkan, jangan lupa untuk mengimport library MPI untuk Python yaitu “**from mpi4py import MPI”** dan juga mengambil nilai range proses dan total proses yang bejalan dengan cara :



**HASIL UJI COBA**



Gambar diatas adalah hasil run program. Bisa kita lihat bahwa pada algoritma *K-Nearest Neighbor* dengan MPI memiliki waktu yang lebih cepat yaitu 33.85 detik. Sedangkan algoritma *K-Nearest Neighbor* secara serial yaitu 158.94 detik.



Gambar diatas merupakan data hasil prediksi dari algoritma KNN dengan MPI (sebelah kiri) dan secara serial (sebelah kanan). Data yang dihasilkan tidak ada perubahan yang berarti akurasi serial dan MPI sama.

KESIMPULAN

Berdasarkan hasil uji coba diatas, dapat disimpulkan bahwa algoritma KNN yang diparallelkan menggunakan MPI memiliki waktu eksekusi yang lebih baik jika dibandingkan dengan algoritma KNN secara serial yang bisa memakan waktu hingga 158.94 detik dimana algoritma KNN secara parallel dapat bekerja 4.7 kali lebih cepat dibandingkan waktu eksekusi secara serial.