**Laporan Praktikum 3**

**IF3230 Sistem Paralel dan Terdistribusi**

***“Penggunaan OpenMPI – Quicksort”***



Disusun oleh :

**Atika Firdaus 13514009**

**Adam Rotal Y. 13514091**

**PROGRAM STUDI TEKNIK INFORMATIKA**

**SEKOLAH TEKNIK ELEKTRO DAN INFORMATIKA**

**INSTITUT TEKNOLOGI BANDUNG**

**2016/2017**

1. **Deskripsi Solusi Paralel**

Pada praktikum 3 ini, kami diminta membuat implementasi permasalahan *quicksort* paralel menggunakan OpenMPI. Pada OpenMPI, terdapat beberapa metode di antaranya adalah *Send and Receive, Broadcast, Scatter and Gather, Reduce,* dan *Communicator and Group.* Kami menggunakan metode *Scatter and Gather.*

1. **Analisis Solusi**
2. **Pengukuran Performansi**

Pengujian kami lakukan dengan mengeksekusi program menggunakan kedua metode, yaitu *quicksort* serial dan paralel, menggunakan empat kasus yaitu *array of integer* berukuran 50.000, 100.000, 200.000, dan 400.000. Hasil eksekusi program dapat dilihat dalam gambar berikut.

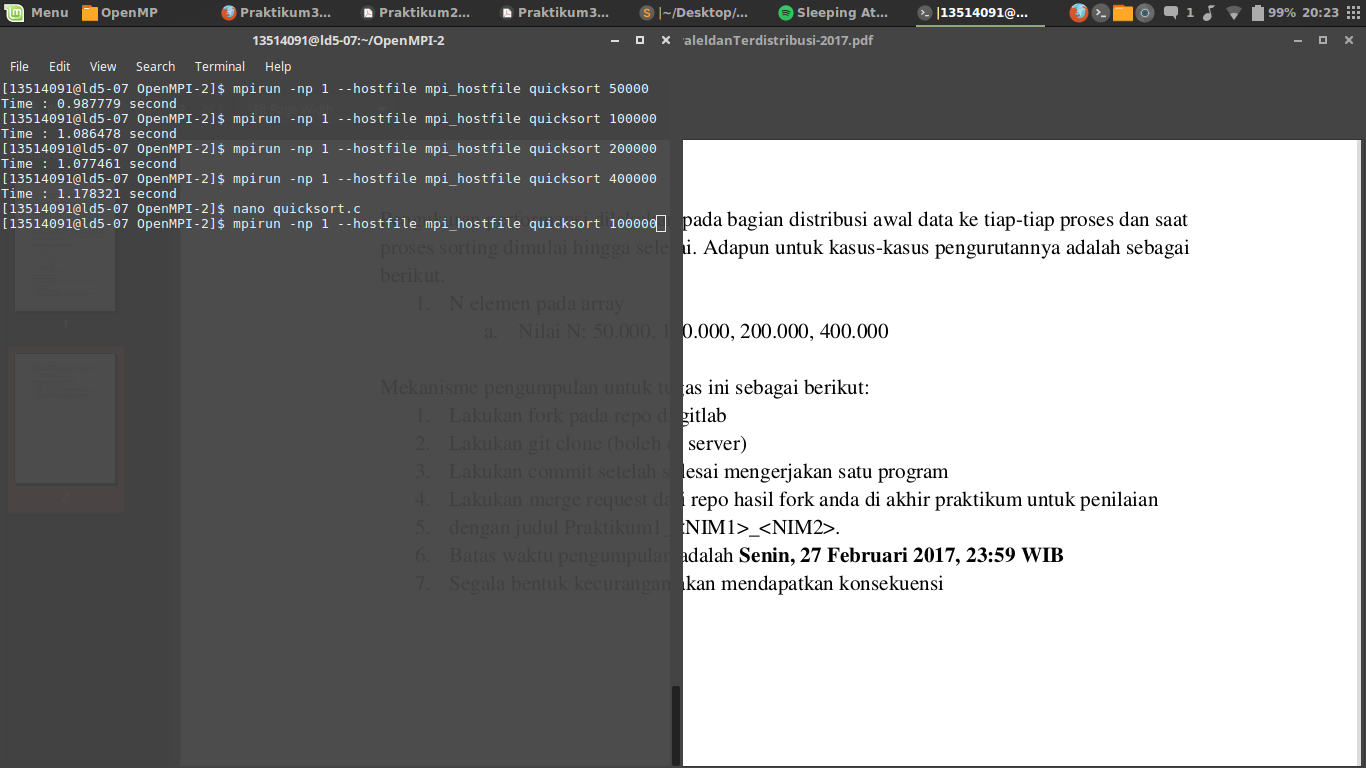


Figure – Hasil eksekusi program serial

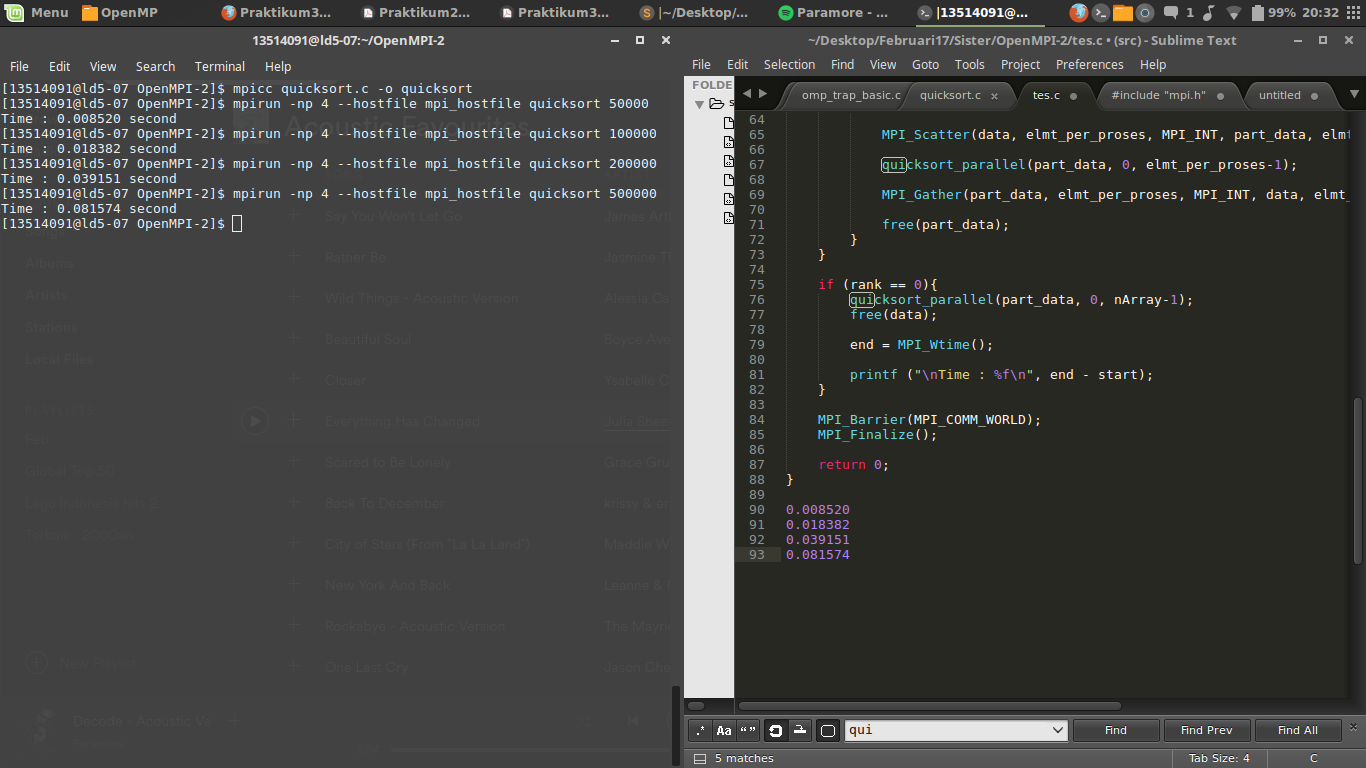


Figure – Hasil eksekusi program paralel

1. **Analisis Perbandingan Performansi Serial dan Paralel**

Berdasarkan pengujian yang kami lakukan, program paralel pada semua kasus ukuran *array of integer* membutuhkan waktu untuk mengeksekusi proses lebih cepat dibanding proses pada program serial. Pada praktikum ini, kami menggunakan metode *Scatter and Gather* pada OpenMPI. Metode ini memecah proses yang utuh ke dalam proses-proses kecil kemudian mengumpulkan hasil dari proses-proses kecil yang telah dilaksanakan. Proses *scatter and gather* dilakukan secara berulang sesuai hasil pembagian dari jumlah elemen pada *array of integer* dengan jumlah *thread* yang digunakan.

Proses dari progam paralel membutuhkan waktu eksekusi yang lebih singkat daripada program serial karena data yang diolah setiap satu proses *sorting* lebih sedikit. Oleh karena itu, proses pembandingan elemen *array* serta proses *swap* juga lebih singkat. Selain itu, setiap proses *sorting* setelah *scatter* berjalan secara paralel, sehingga mereka tidak perlu menunggu proses lain hingga selesai untuk melakukan prosesnya.