Nama : Baginda

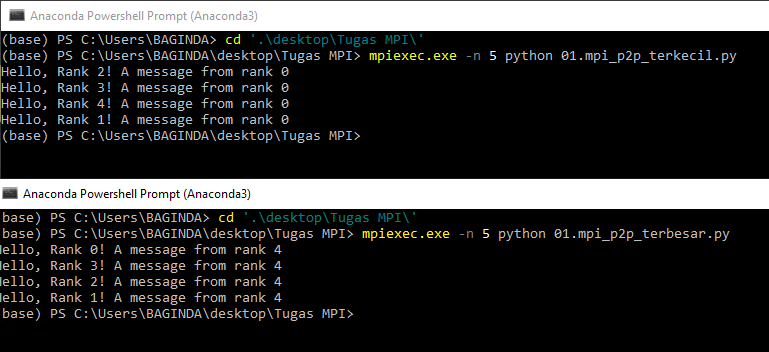
Kelas : IF-41

NIM : 130-

**CSH3J3 – SISTEM PARALEL DAN TERDISTRIBUSI**

**TUGAS MPI**

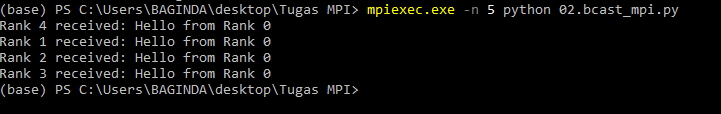
* **Tugas 1: Point-to-Point Communication**

****

Pada kedua *file* di atas, terdapat lima *rank* proses, yaitu *rank* 0 s.d. 4. Pada *file* **01.mpi\_p2p\_terkecil.py** lima program pada kode tersebut dijalankan sekaligus secara paralel. Hasilnya, *rank* 0 mengirimkan pesan kepada semua *rank* lain yang tersedia (yaitu *rank* 1-4), dan *rank* 1-4 menampilkan pesan yang mereka terima dari *rank* 0.

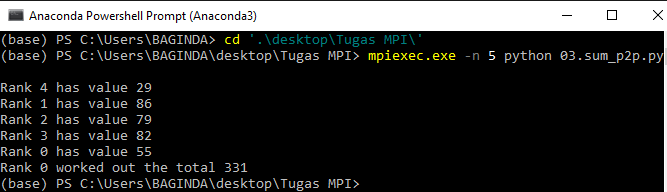
Pada *file* **01.mpi\_p2p\_terbesar.py** kelima program pada kode ini juga dijalankan sekaligus secara paralel. Hanya saja, di kode ini *rank* yang terbesar yang akan mengirim pesan ke *rank-rank* yang lebih kecil. Hasilnya, *rank* 4 mengirimkan pesan kepada *rank* 0-3, dan *rank* 0-3 menampilkan pesan yang mereka terima dari *rank* 4.

* **Tugas 2: Broadcast Communication**

****

Pada *file* **02.bcast\_mpi.py** *rank* 0 melakukan pengiriman pesan secara *broadcast* (ke seluruh *rank*yang ada). Hasilnya, *rank* 0 mengirimkan pesan “Hello from Rank 0”, dan *rank* lainnya (1-4) akan menampilkan pesan yang diterima dari *rank* 0. Program ini hampir sama dengan program **01.mpi\_p2p\_terkecil.py**. Bedanya adalah, program **terkecil** menggunakan perulangan untuk mengirimkan pesan yang sama sebanyak ***n*** *rank*. Sedangkan, pada program **broadcast** hanya menggunakan fungsi bcast().

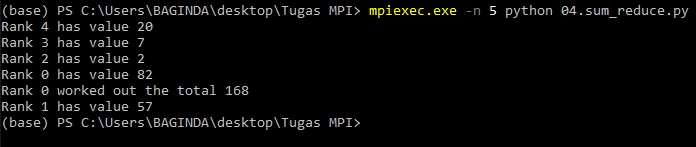
* **Tugas 3: Penjumlahan MPI Teknik P2P**

****

Pada *file* **03.sum\_p2p.py** dilakukan penjumlahan *value* dari *rank-rank­* yang ada menggunakan teknik P2P. Dalam hal ini, setiap proses/*rank* diberikan nilai secara acak, hasilnya *rank* 0, 1, 2, 3, dan 4 masing-masing mendapatkan *value* yaitu 55, 86, 79, 82, dan 29.

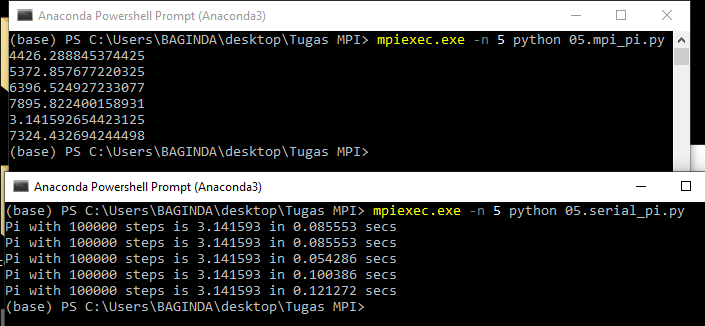
Pada teknik ini, *rank* 0 yang bertugas untuk mengumpulkan/menerima *value* dari *rank* yang lain. Setelah terkumpul, *rank* 0 akan menjumlahkan *value-value* tersebut, kemudian ditampilkan. Hasilnya, *rank* 0 menjumlahkan total *value* sebesar 331.

* **Tugas 4: Penjumlahan MPI Teknik Reduce**

****

Pada *file* **04.sum\_reduce.py** hal yang dilakukan adalah sama seperti *file* **03.sum\_p2p.py** yaitu menjumlahkan *value* acak dari *rank* yang ada. Bedanya, program ini menggunakan teknik *reduce*. Teknik ini adalah teknik *native* dari MPI, yang dapat dipanggil dengan fungsi reduce(). Teknik ini lebih cepat jika dibandingkan dengan teknik P2P yang berjalan secara sekuensial.

* **Tugas 5: Menghitung Nilai *pi***

****

Pada *file* **05.serial\_pi.py** dilakukan perhitungan nilai *pi* secara serial menggunakan pendekatan 4/(1+x^2). Pada teknik ini, akan dihitung luas area di bawah kurva 4/(1+x^2). Menghitung luas kurva dapat didekati secara diskit dengan menentukan banyaknya *step* yang digunakan. Semakin banyak *step* yang digunakan maka hasilnya akan lebih akurat tetapi proses akan sangat lama dan membutuhkan banyak *resource*. Agar mempercepat proses, digunakan MPI untuk melakukan perhitungan aproksimasi nilai *pi* dengan solusi *“decompose then collect”*.

Dengan teknik MPI, nilai *pi* dihitung secara paralel, yaitu setiap proses akan mendapatkan bagian untuk menghitung hasil bagiannya masing-masing. Pada file **05.mpi\_pi.py**, proses dibagi menjadi lima, sehingga akan menghasilkan pembagian sebesar 0.2 / 20% perhitungan pada masing-masing prosesnya.