LAPORAN PRAKTIKUM
MESSAGE PASSING INTERFACE (MPI)

Disusun Oleh:

Kelompok 7

M. Athallah Rafif Aldera	(09011282227066)
Tiara Putri Amanda	(09011182227015)
Raihan Putra Ersananda	(09011282227117)
Nanda Ausil Jannati	(09011282227096)

DOSEN PENGAMPU:

Adi Hermansyah, S.Kom., M.T.

PROGRAM STUDI SISTEM KOMPUTER

FAKULTAS ILMU KOMPUTER

UNIVERSITAS SRIWIJAYA

2023

Langkah 1 : Persiapan

Pertama kita siapkan hostname seperti pada praktikum sebelumnya, setelah hostname disiapkan kita akan menyiapkan MPI pada master, setelah semua persiapan selesai kita akan membuat dan menjalankan program numerik eliminasi gauss.

Langkah 2 : Pembuatan Program

Kita jalankan pada computer master command
$ sudo nano koli.py
Lalu tuliskan codingan seperti di bawah

```
    for i in range(baris):
        diag = A[i][i]
        for j in range(kolom):
            A[i][j] /= diag
        for k in range(i + 1, baris):
            diag1 = A[k][i]
            for j in range(0, kolom):
                A[k][j] = A[k][j] - diag1 * A[i][j]

    comm.barrier()

    if rank == 0:
        print("Eliminasi Gauss:")
        for i in range(baris):
            for j in range(kolom):
                print(f"{A[i][j]:.2f}", end=" ")
            print()

        x = np.zeros(ordo, dtype=float)
        for i in range(ordo - 1, -1, -1):
            x[i] = A[i][kolom - 1]
            for j in range(i + 1, ordo):
                x[i] -= A[i][j] * x[j]

        end_time = time.time()
        elapsed_time = end_time - start_time

        print("\nSolution:")
        for i in range(ordo):
            print(f"x{i + 1} = {x[i]:.2f}")

        print(f"Elapsed time: {elapsed_time:.4f} seconds")
```

```
from mpi4py import MPI
import numpy as np
import time

comm = MPI.COMM_WORLD
rank = comm.Get_rank()
size = comm.Get_size()

def main():
    ordo = None

    if rank == 0:
        ordo = int(input("Masukkan ordo matrix: "))

    ordo = comm.bcast(ordo, root=0)

    if ordo is None:
        return

    baris = ordo
    kolom = ordo + 1

    A = np.zeros((baris, kolom), dtype=float)

    if rank == 0:
        for i in range(baris):
            print(f"Masukkan persamaan ke-{i + 1}")
            for j in range(kolom):
                A[i][j] = float(input(f"Masukkan angka baris-{i + 1} kolom-{j + 1}: "))

    A = comm.bcast(A, root=0)

    start_time = time.time()
```

```
if __name__ == '__main__':
    main()
```

Langkah 3 : Menjalankan Program Numerik

Untuk menjalankan program numerik caranya sama seperti menjalankan program bubble sort pada praktikum sebelumnya yaitu :

$ mpiexec -n 2 -host aral-amd-lol,tiaraputri python3 koli.py

Jumlah host, Hostname, Nama file, dapat disesuaikan sesuai kondisi

Output dari program numerik yang dijalankan adalah sebagai berikut

```
^C^C^Cke17@aral-amd-lmpiexec -n 2 -host aral-amd-lol,tiaraputri python3 /home/ke17/ke17/koli.py
Masukkan ordo matrix: 2
Masukkan persamaan ke-1
Masukkan angka baris-1 kolom-1: 1
Masukkan angka baris-1 kolom-2: 1
Masukkan angka baris-1 kolom-3: 1
Masukkan persamaan ke-2
Masukkan angka baris-2 kolom-1: 1
Masukkan angka baris-2 kolom-2: 2
Masukkan angka baris-2 kolom-3: 2
Eliminasi Gauss:
1.00 1.00 1.00
0.00 1.00 1.00

Solution:
x1 = 0.00
x2 = 1.00
Elapsed time: 0.0676 seconds
ke17@aral-amd-lol:~$ _
```

KESIMPULAN :

Dalam laporan ini, dilakukan implementasi eliminasi Gauss menggunakan MPI (Message Passing Interface) dengan bahasa pemrograman Python menggunakan modul mpi4py. Berikut adalah kesimpulan utama yang dapat diambil dari laporan ini:

Laporan ini membahas langkah-langkah implementasi eliminasi Gauss secara terdistribusi dengan memanfaatkan MPI. Penggunaan MPI memungkinkan program untuk bekerja secara paralel di beberapa node, mempercepat penyelesaian sistem persamaan linear dengan ordo yang lebih besar.

Pentingnya interaktivitas pengguna ditekankan di mana program memungkinkan pengguna untuk memasukkan ordo matriks dan elemen matriks augmented. Hal ini memberikan fleksibilitas kepada pengguna untuk menyesuaikan program dengan kebutuhan mereka.

Konsep distribusi proses juga dijelaskan, di mana setiap proses bekerja pada bagian tertentu dari matriks augmented. Ini memungkinkan program untuk memproses data secara efisien dan paralel, meningkatkan kinerja keseluruhan.

Output dari program, termasuk matriks eselon tereduksi dan solusi dari sistem persamaan linear, memberikan informasi yang jelas dan berguna kepada pengguna. Waktu eksekusi program juga dicatat, memberikan pemahaman tentang kinerja program.