MODUL 3
Model Komputasi Paralel

1.1. Tujuan

Mahasiswa mampu memahami berbagai arsitektur komputasi sehingga dapat memilih model yang paling tepat guna mengoptimalkan kinerja pemrosesan data dan algoritma sesuai kebutuhan.

1.2. Alat & Bahan

1. Perangkat Lunak:
   - VirtualBox (open source)
   - ISO Linux (Ubuntu Server/Desktop)
   - Library (numpy, multiprocessing, time
2. Perangkat Keras:
   - Laptop/PC dengan RAM minimal 8 GB
   - Koneski internet

1.3. Materi

A. SISD (Single Instruction Single Data)

SISD adalah model komputasi di mana satu instruksi dieksekusi pada satu data pada satu waktu. Model ini identik dengan komputer klasik (CPU single-core) yang bekerja secara sekuensial. Penerapan model ini adalah eksekusi loop for di Python yaitu proses nya menjalankan perintah satu per satu cocok digunakan untuk proses logika dan alur kontrol yang tidak bisa diparalelkan.

B. SIMD (Single Instruction Multiple Data)

SIMD adalah model komputasi di mana satu instruksi dieksekusi bersamaan pada banyak data. Cocok untuk operasi matematis yang sama pada array atau matriks besar. Penerapannya GPU untuk training AI (semua neuron dalam layer diproses bersamaan).NumPy dan TensorFlow yang menggunakan vectorization. Image processing (filter diterapkan pada semua pixel).

C. MISD (Multiple Instruction Single Data)

MISD adalah model komputasi di mana beberapa instruksi berbeda dijalankan pada data yang sama. Model ini jarang digunakan untuk percepatan, tapi dipakai untuk fault tolerance dan redundansi. Penerapannya ensemble learning yaitu satu data dianalisis dengan banyak model berbeda.

D. MIMD (Multiple Instruction Multiple Data)

MIMD adalah model komputasi di mana instruksi berbeda dijalankan pada data berbeda secara paralel. Model ini adalah dasar dari komputer modern multi-core dan cluster komputasi. Penerapan nya distributed training model AI di beberapa node. Big Data processing (Spark, Hadoop). Multi-core CPU server.

1.4.Langkah Praktikum

A. SISD

Kode pada gambar 1 menerapkan Single Instruction Single Data (SISD) dimana hanya ada satu instruksi (x * 2) yang dieksekusi untuk satu data pada setiap iterasi. Prosesnya berjalan secara sekuensial dari awal sampai akhir satu per satu. Semua elemen dikalikan dua, tapi dikerjakan bertahap satu per satu. Pada gambar 1 dapat dilihat tidak ada paralelisme sehingga semua elemen diproses secara linear. Contoh code SISD sederhana cocok diimplementasikan untuk data kecil namun akan lambat bila data sangat besar karena setiap elemen harus diproses bergantian.

```
# SISD: Single Instruction Single Data
data = [1, 2, 3, 4, 5]
result = []

# proses sequential
for x in data:
    result.append(x * 2)

print("SISD Result:", result)
```

Gambar 1. Contoh SISD

B. SIMD

Kode pada gambar 2 menerapkan Single Instruction Multiple Data dimana Satu instruksi (*2) diterapkan ke banyak data sekaligus ([1,2,3,4,5]). Berbeda dengan SISD yang memproses data satu per satu dalam loop. SIMD memproses semua elemen array yang dihitung secara bersamaan tanpa harus ditulis dalam loop eksplisit. Pada gambar 1, contoh yang ditampilkan adalah contoh untuk instruksi yang sama. SIMD hanya cocok jika semua data menjalani instruksi yang sama sehingga tidak fleksibel untuk data dengan instruksi berbeda-beda.

```
# SIMD: Single Instruction Multiple Data
import numpy as np

data = np.array([1, 2, 3, 4, 5])

# operasi vectorized (semua elemen dikali 2 secara paralel)
result = data * 2

print("SIMD Result:", result)
```

Gambar 2. Contoh SIMD

C. MISD

Kode pada gambar 4 menerapkan Multiple Instruction Single Data artinya satu data diproses oleh beberapa instruksi berbeda secara bersamaan. Dalam contoh code dapat dilihat terdapat data x=5 dengan instruksi kuadrat, pangkat tiga, dan akar kuadrat. Model ini tidak efisien untuk komputasi besar karena satu data saja diproses oleh banyak instruksi, bukan banyak data. MISD di Python lebih ke simulasi konsep. Dalam dunia nyata penerapan MISD lebih relevan di sistem real-time dan fault-tolerant computing.

```
# MISD: Multiple Instruction Single Data
x = 5

# instruksi berbeda pada data yang sama
square = x ** 2
cube   = x ** 3
root   = x ** 0.5

print("MISD Result:", {"square": square, "cube": cube, "root": root})
```

Gambar 3 contoh MISD

D. MIMD

Kode pada gambar 4 menerapkan Multiple Instruction Multiple Data artinya beberapa instruksi berbeda dieksekusi pada data yang berbeda secara paralel. Pada gambar 5 dapat dilihat ada instruksi pertama yaitu menghitung kuadrat untuk data1 dan instruksi kedua yaitu menghitung pangkat tiga untuk data2. Keduanya dijalankan secara paralel menggunakan multiprocessing

```python
# MIMD: Multiple Instruction Multiple Data
from multiprocessing import Process

def square(numbers):
    print("Square:", [n**2 for n in numbers])

def cube(numbers):
    print("Cube:", [n**3 for n in numbers])

if __name__ == "__main__":
    data1 = [1, 2, 3, 4]
    data2 = [5, 6, 7, 8]

    # proses paralel
    p1 = Process(target=square, args=(data1,))
    p2 = Process(target=cube, args=(data2,))

    p1.start()
    p2.start()
    p1.join()
    p2.join()
```

Gambar 4. Contoh MIMD