# Jaringan dan Pengolahan Data Paralel

Tugas mpi4py

Mohammad Rizka Fadhli
Ikang
20921004@mahasiswa.itb.ac.id

24 November 2021

## SOAL

Buat program serial dan paralel (menggunakan *point to point communication*) untuk integrasi fungsi $f(x) = x^2$ menggunakan metode *trapezoid*, dari interval $x = 0$ sampai $x = 1$ dengan nilai $n$ bervariasi mulai dari $10^0$ sampai dengan $\$10^9$ dengan menggunakan 4 *processor core*.

Buatlah *plot* grafik antara nilai $n$ terhadap waktu untuk masing-masing nilai $n$ tersebut untuk membandingkan kecepatan perhitungan antara program serial dan program paralel yang menggunakan komunikasi *point to point*.

## JAWAB

Untuk membandingkannya, saya mencoba *generate* beberapa nilai *random* untuk $n$.

## Serial

Berikut adalah program yang dibuat:

```
import time
import numpy as np

def y(x):
    return x**2

def trapezoidal(a,b,n):
    h=(b-a)/n
    s=(y(a)+y(b))
    i=1
    for i in range (1,n):
          s+=2*y(a+i*h)
          i+=1
    return((h/2)*s)

x0 = 0
xn = 1
n = [1,7,16,28,36,43,52,62,69,77,93,322,359,
3620,4374,4513,5575,6680,7455,9334,9533,227373,235875,
359596,441394,611944,651434,707928,756462,
914272,947220,1911445,3542919,4448887,33173155,33744249,
53990692,60063717,60461719,68180038,72434461,
74711604,76351391,80842245,89589529,90095249,90814922,
92609483,95720398,96263304,97769854,217967736,314781656,
356988524,719449697,862513064,1000000000]

n_selang = []
```

```
integral = []
waktu = []

for i in range(0,len(n)):
  start=time.time()
  t_1 = trapezoidal(x0,xn,n[i])
  end = time.time()
  t_2 = end - start
  n_selang.append(n[i])
  integral.append(t_1)
  waktu.append(t_2)
```

Berikut adalah plot hasil *runtime*:

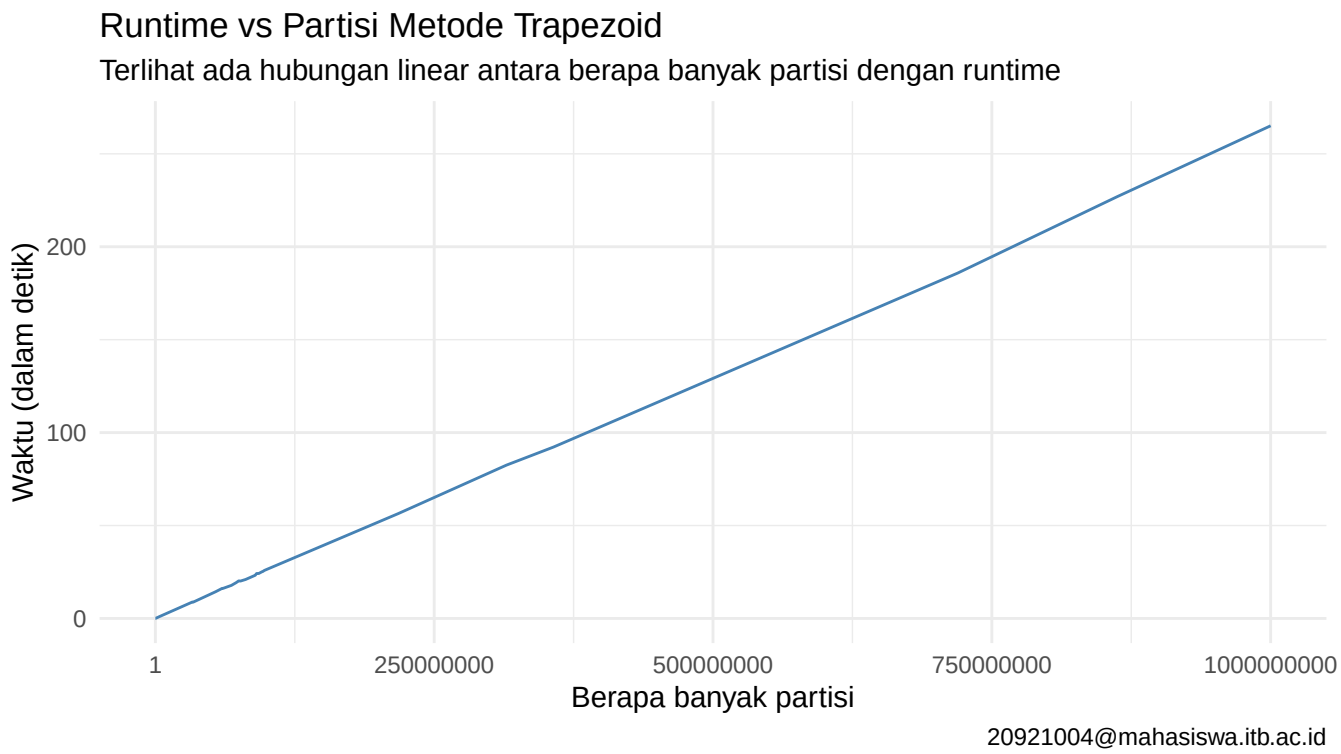


Figure 1: Pengolahan Data Serial

## Paralel

Berikut adalah program yang dibuat:

```
import timeit
import numpy as np
from mpi4py import MPI
from mpi4py.MPI import ANY_SOURCE

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

def y(x):
    return x**2

def trapezoidal(a,b,n):
    h=(b-a)/n
    s=(y(a)+y(b))
    i=1
    while i<n:
        s+=2*y(a+i*h)
        i+=1
    return((h/2)*s)

a=0
b=1
n = [1,7,16,28,36,43,52,62,69,77,93,322,359,3620,4374,
4513,5575,6680,7455,9334,9533,227373,235875,359596,441394,
611944,651434,707928,756462,914272,947220,1911445,3542919,
4448887,33173155,33744249,53990692,60063717,60461719,68180038,
```

```
72434461,74711604,76351391,80842245,89589529,90095249,90814922,
92609483,95720398,96263304,97769854,217967736,314781656,356988524,
719449697,862513064,1000000000]

integral = []
waktu = []

for i in range(0,len(n)):
  dest=0
  h=(b-a)/n[i]
  local_n=n[i]/size
  local_a=a+rank*local_n*h
  local_b=local_a+local_n*h
  t1 = MPI.Wtime()
  s=trapezoidal(local_a,local_b,local_n)
  if rank==0:
      total=s
      for _ in range(size-1):
          total2 = comm.recv(source = MPI.ANY_SOURCE)
          total += total2
      integral.append(total)
      waktu.append(MPI.Wtime()-t1)
  else:
      comm.send(s,dest=0)
```

Berikut adalah plot hasil *runtime*:

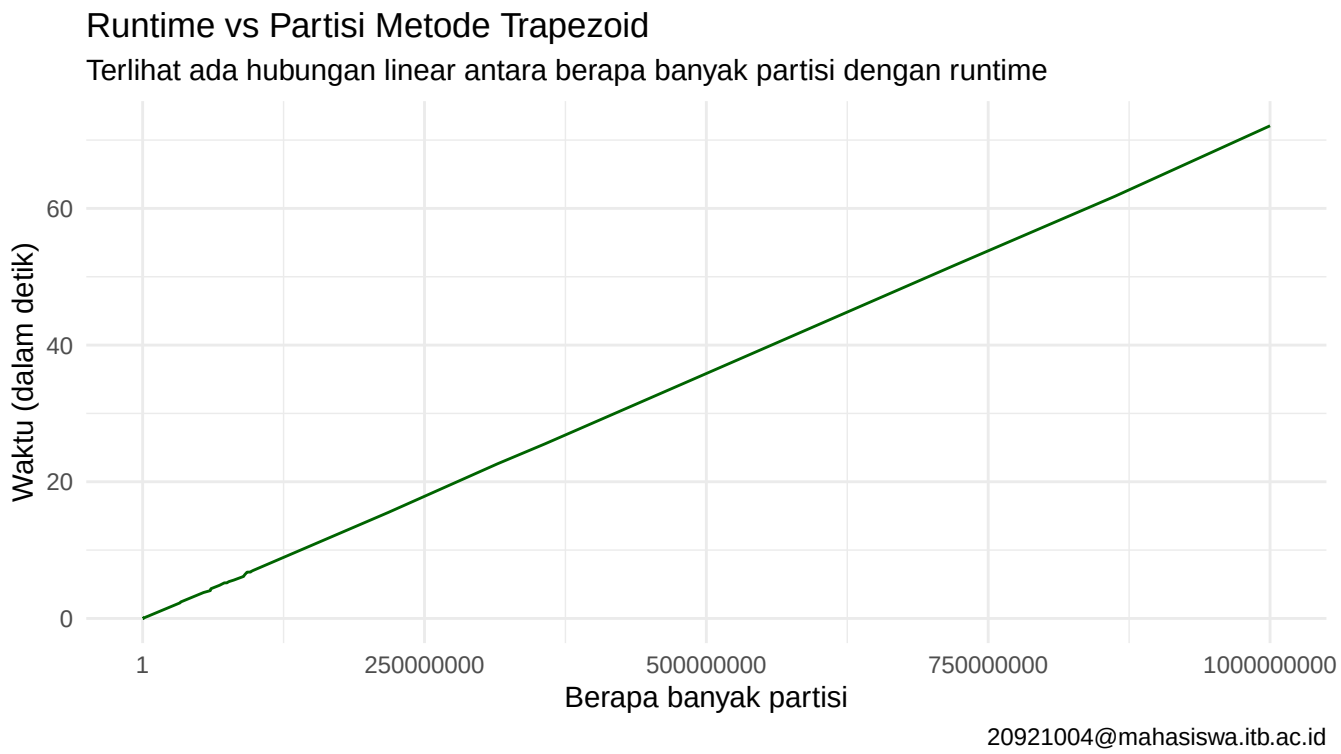


Figure 2: Pengolahan Data Paralel

# KESIMPULAN

*Paralel processing* menghemat *runtime* dengan signifikan.