**PEMROSESAN PARALLEL**

**LAPORAN PROGRES PROYEK UAS**

****

Nama : Dimas Aditya Kristianto

NIM : 09011281823132

Jurusan : Sistem Komputer

**FAKULTAS ILMU KOMPUTER**

**UNIVERSITAS SRIWIJAYA**

**2020/2021**

1. Kasus

Host mengirim beberapa nilai yang akan dikirim kepada node yang sedang aktif untuk dihitung menggunakan rumus luas dan keliling bangun datar segitiga, lingkaran, dan persegi panjang. Kemudian hasil dari perhitungan tersebut akan dikirim Kembali ke host

1. Metode

Pada sisi host dan setiap node, memiliki program masing masing.

* 1. Host

Program yang ada pada host adalah program input nilai. Dimana progam akan menampung nilai apa saja yang akan dikirim ke masing -masing node.

Pada saat program di sisi host dijalankan, program akan memeriksa apakah semua node yang ada berstatus “ONLINE” atau “OFFLINE”. Kemudian memasukkan input nilai.

Pada saat input nilai, input nilai pertama akan menampung pemilihan operasi bangun datar yang akan dilakukan oleh setiap node. Tergantung pada input pertama, total input berikutnya akan berbeda. Total input sengaja dibuat berbeda karena setiap operasi bangun datar memerlukan total input yang berbeda

Jika input pertama adalah 1, maka operasi yang dipilih adalah segitiga. Total input untuk operasi bangun datar segitiga adalah 2, yaitu untuk input alas dan tinggi. Jika input pertama adalah 2, maka operasi yang dipilih adalah lingkaran. Total input untuk operasi bangun datar lingkaran adalah 1, yaitu untuk input jari-jari. Jika input pertama adalah 3, maka operasi yang dipilih adalah persegi Panjang. Total input untuk operasi bangun datar persegi Panjang adalah 2, yaitu untuk input Panjang dan lebar.

Ketika host sudah selesai memasukkan semua input yang diperlukan, program akan memasukkan nya pada sebuah file bernama ‘file\_send.txt’ yang berisi setiap input yang dipisahkan menggunakan tanda koma(,). Pada node akan menerima file yang sama dengan nama ‘input.txt’. File tersebut akan dikirim ke setiap node dan hasil dari operasi tersebut akan ditampilkan pada konsol host

* 1. Node

Program yang ada pada node adalah pemrosesan input nilai. Dimana program akan mulai memproses file yang diterima dari sisi Host.

Pada saat program dijalankan, program akan membaca file yang bernama ‘input.txt’ yang berisi input dari sisi host. Isi dari file tersebut memiliki tipe data string. Isi file tersebut dipisahkan menggunakan tanda koma(,) sehingga didapatkan sebuah array yang menampung nilai string. Setiap string yang terdapat dalam array, akan diubah menjadi tipe data numerik supaya program dapat menghitung input nilai yang diberikan oleh host.

Array index pertama adalah pilihan operasi yang akan dilakukan. Array berikutnya adalah input nilai yang akan dihitung menggunakan rumus masing masing bangun datar.

Jika array index pertama adalah 1, maka operasi yang dipilih adalah segitiga. Rumus segitiga yang digunakan :

Jika array index pertama adalah 2, maka operasi yang dipilih adalah lingkaran. Rumus lingkaran yang digunakan :

Jika array index pertama adalah 3, maka operasi yang dipilih adalah persegi Panjang. Rumus persegi Panjang yang digunakan :

Hasil dari operasi akan ditampilkan pada konsol Host

1. Mode jaringan

Pada setiap node, mode jaringan yang digunakan adalah mode Bridge Adapter. Mode Bridge Adapter dipilih karena Bridge Adapter adalah satu satunya mode yang dapat menghubungkan antara host dan setiap node

1. Program

Bahasa pemrograman yang digunakan adalah python versi 3.7.4 .

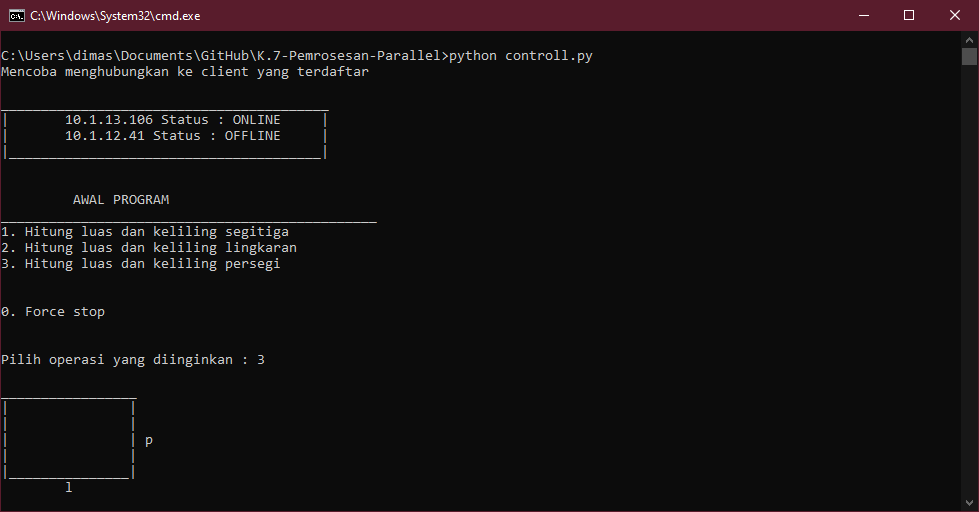
* 1. Host

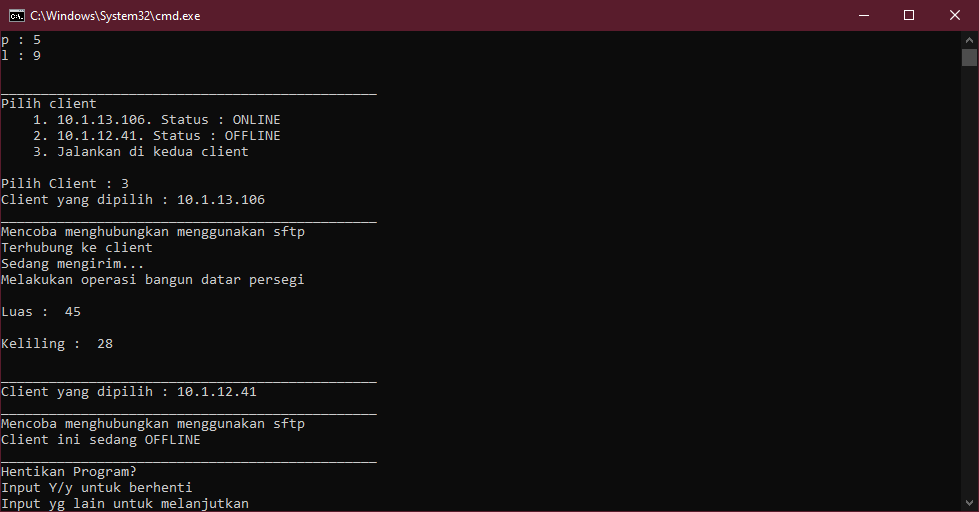
Program pada host menggunakan package paramiko untuk menghubungkan setiap node menggunakan SSH. Program memerlukan SSHClient dan AutoAddPolicy yang ada pada paramiko.client untuk menghubungkan host dan setiap node. Untuk mencegah berhentinya program saat error muncul, diperlukan package socket untuk menangkap error nya. Error yang sejauh praktik ini dilakukan adalah :

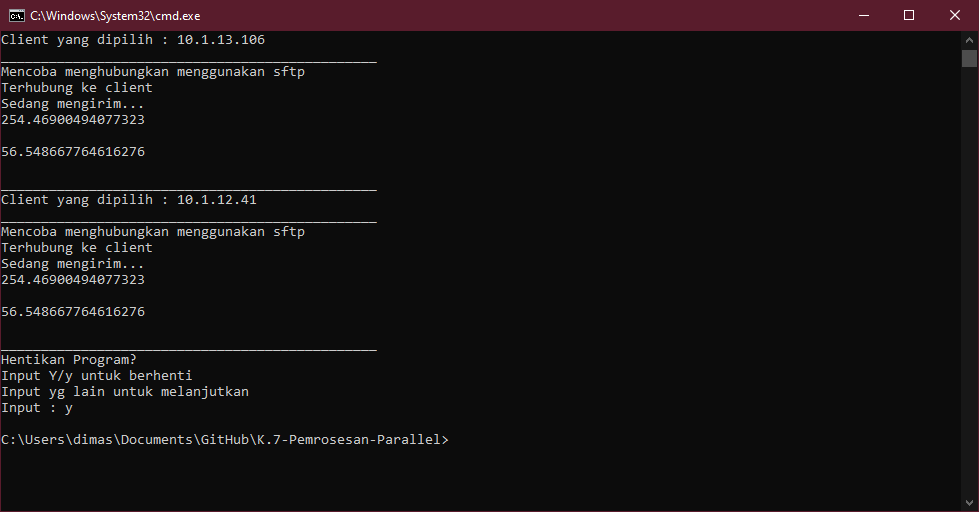
* + - * socket.gaierror = terjadi saat ip address tidak dapat dikenali
      * socket.timeout = terjadi saat waktu untuk menghubungkan host dan node, terlalu lama
  1. Node

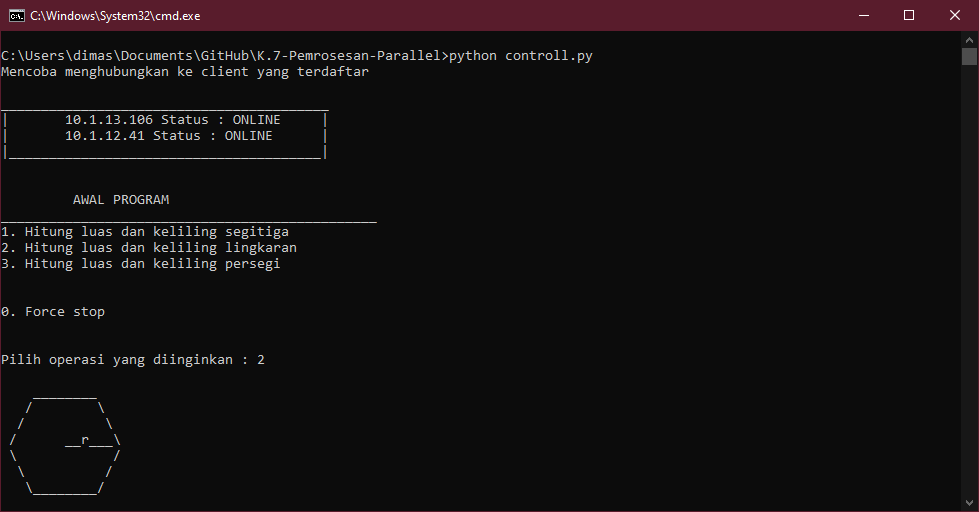
Program pada node menggunakan package math untuk mendapatkan nilai pi supaya perhitungan menjadi lebih akurat

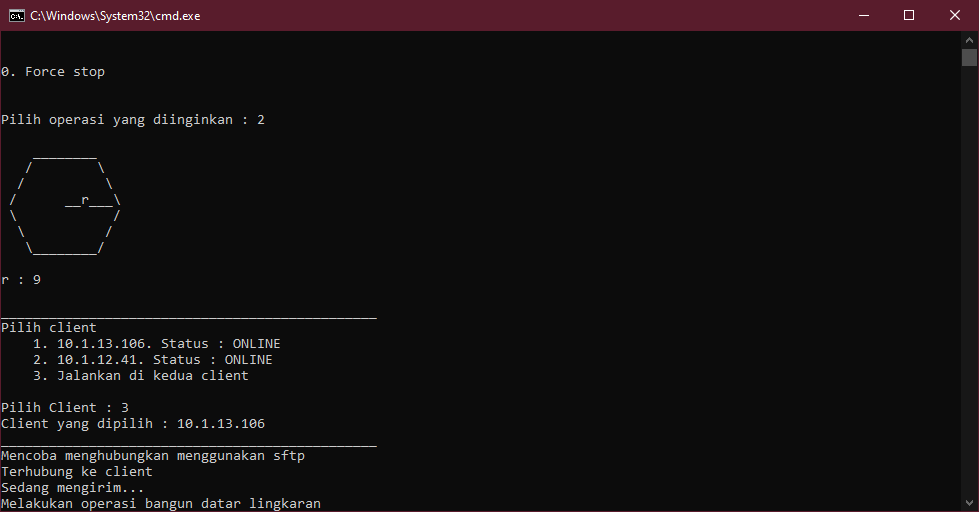
1. Lampiran

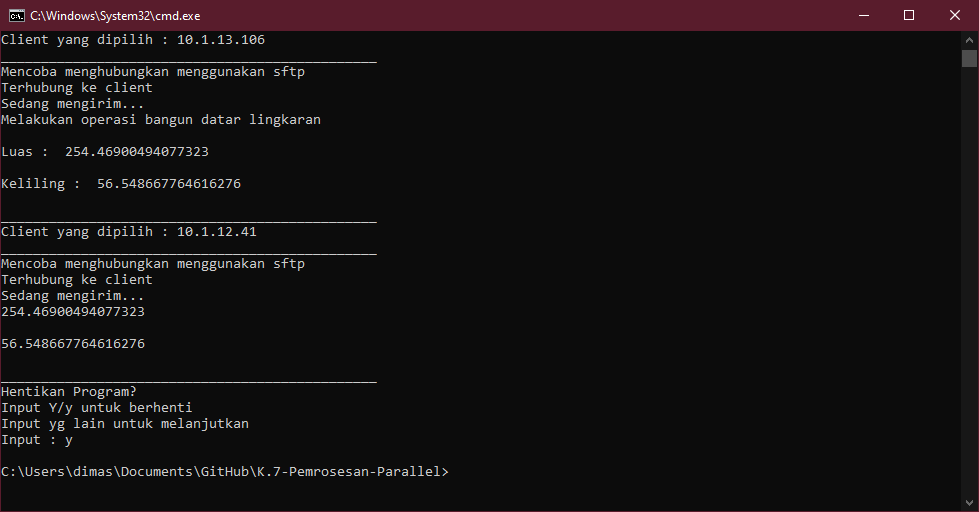












1. Source code

from paramiko.client import SSHClient, AutoAddPolicy

import sys

import socket

ssh = '10.1.13.106'

username = 'ldimas'

passhehe = 'root'

ssh2 = '10.1.12.41'

username2 = 'ldimas'

passhehe2 = 'root'

client = []

status = []

client.append(SSHClient())

client.append(SSHClient())

print("Mencoba menghubungkan ke client yang terdaftar")

client[0].set\_missing\_host\_key\_policy(AutoAddPolicy())

try:

    client[0].connect(ssh, 22, username, passhehe, timeout=5)

    status.append('ONLINE')

except (socket.gaierror, socket.timeout):

    status.append('OFFLINE')

client[1].set\_missing\_host\_key\_policy(AutoAddPolicy())

try:

    client[1].connect(ssh2, 22, username2, passhehe2, timeout=5)

    status.append('ONLINE')

except (socket.gaierror, socket.timeout):

    status.append('OFFLINE')

print('''

\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_

|\t{} Status : {}\t|

|\t{} Status : {}\t|

|\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_|

    '''.format(ssh,  status[0],

               ssh2, status[1])

      )

def pilihan():

    print('''

\t AWAL PROGRAM

\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_

1. Hitung luas dan keliling segitiga

2. Hitung luas dan keliling lingkaran

3. Hitung luas dan keliling persegi

0. Force stop

    ''')

    choice = input('Pilih operasi yang diinginkan : ')

    return choice

def inp\_sisi(masukan):

    if masukan == '1':

        print('''

    |\

    | \

  a |  \ c

    |   \

    |\_\_\_\_\

      b

        ''')

        a = input('a : ')

        b = input('b : ')

        c = input('c : ')

        return [a, b, c]

    elif masukan == '2':

        print('''

    \_\_\_\_\_\_\_\_

   /        \

  /          \

 /      \_\_r\_\_\_\

 \            /

  \          /

   \\_\_\_\_\_\_\_\_/

        ''')

        r = input('r : ')

        return [r]

    elif masukan == '3':

        print('''

\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_

|               |

|               |

|               | p

|               |

|\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_|

        l

        ''')

        p = input('p : ')

        l = input('l : ')

        return [p, l]

def write\_it(a):

    file1 = open("file\_send.txt", "w")

    joined = ','.join(map(str, a))

    file1.write(joined)

    file1.close()

def sendandcount\_it(cc):

    print('\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_')

    print('Mencoba menghubungkan menggunakan sftp')

    try:

        sftp = cc.open\_sftp()

        print('Terhubung ke client')

        print('Sedang mengirim...')

        sftp.put('./file\_send.txt', './input.txt')

        stdin, stdout, stderr = cc.exec\_command('python3 hitung.py')

        baris = stdout.readlines()

        baris\_err = stderr.readlines()

        for i in baris\_err:

            print(i)

        for i in baris:

            print(i)

    except:

        print("Client ini sedang OFFLINE")

    print('\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_')

def CloseAllClient(cli):

    for i in cli:

        i.close()

menu = True

while(menu):

    pilih\_menu = True

    menu\_rumus = []

    while(pilih\_menu):

        menu\_rumus.clear()

        menu\_rumus.append(pilihan())

        if menu\_rumus[0] == '1' or menu\_rumus[0] == '2' or menu\_rumus[0] == '3':

            menu\_rumus = menu\_rumus + inp\_sisi(menu\_rumus[0])

            pilih\_menu = False

        elif menu\_rumus[0] == '0':

            pilih\_menu = False

            menu = False

            CloseAllClient(client)

            sys.exit()

        else:

            print('''

\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_

    MASUKKAN PILIHAN YANG BENAR!!!

\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_

''')

    print('''

\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_

Pilih client

    1. {}. Status : {}

    2. {}. Status : {}

    3. Jalankan di kedua client

    '''.format(ssh,  status[0],

               ssh2, status[1])

          )

    input\_menu = input('Pilih Client : ')

    if (input\_menu == '1'):

        print('Client yang dipilih : '+ssh)

        write\_it(menu\_rumus)

        sendandcount\_it(client[0])

    elif (input\_menu == '2'):

        print('Client yang dipilih : '+ssh2)

        write\_it(menu\_rumus)

        sendandcount\_it(client[1])

    elif (input\_menu == '3'):

        write\_it(menu\_rumus)

        print('Client yang dipilih : '+ssh)

        sendandcount\_it(client[0])

        print('Client yang dipilih : '+ssh2)

        sendandcount\_it(client[1])

    keluar = input(

        "Hentikan Program?\nInput Y/y untuk berhenti\nInput yg lain untuk melanjutkan\nInput : ")

    if (keluar == 'Y' or keluar == 'y'):

        CloseAllClient(client)

        menu = False