

Nama : Dika Pramudyatama  
NIM : L200190224

## Praktikum Alogaritma dan Pemrograman Modul 10

### Kegiatan 1. Data diri dari server

```
import socket

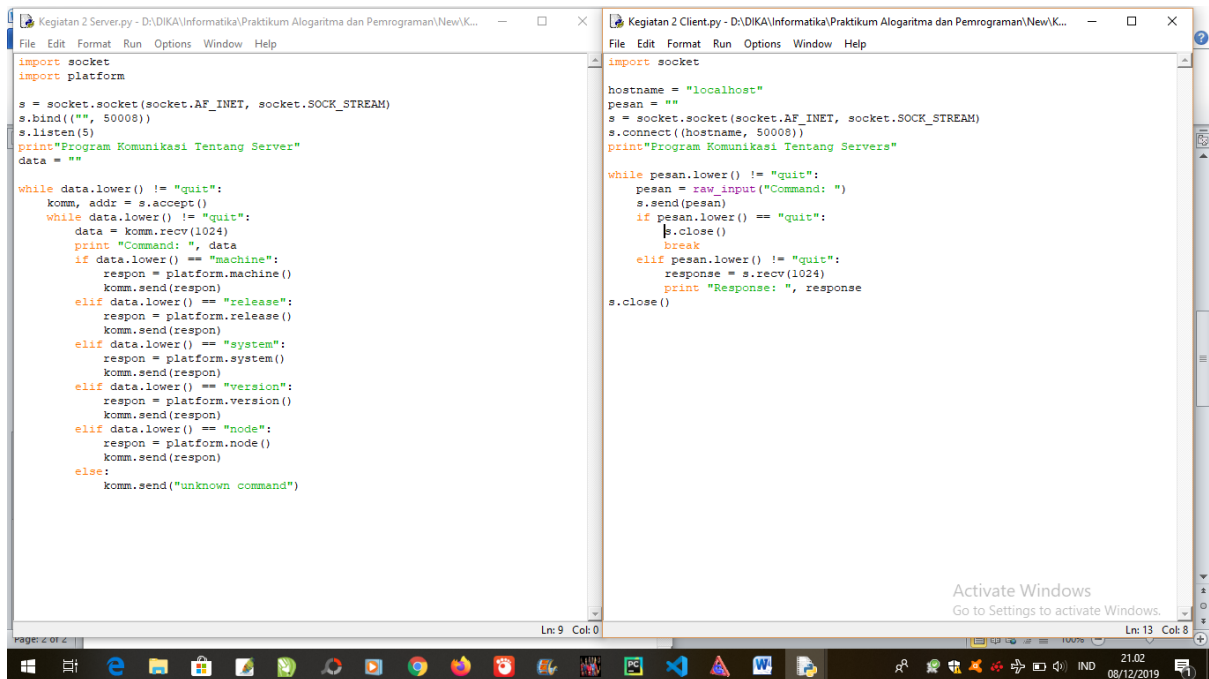
s = socket.socket(socket.AF_INET, socket.SOCK_STREAM)
s.bind(("", 50008))
s.listen(5)
print("Program Komunikasi Tentang Data Diri")
data = ""
kamus = {"nama": "Dika Pramudyatama",
          "NIM": "L200190224",
          "angkatan": "2019",
          "keluar": "Siap!"}
while data.lower() != "keluar":
    komm, addr = s.accept()
    while data.lower() != "keluar":
        data = komm.recv(1024)
        print("Perintah: ", data)
        if kamus.has_key(data):
            komm.send(kamus[data])
        else:
            komm.send("Maaf, Perintah Tidak Dimengerti")

import socket

hostname = "localhost"
pesan = ""
s = socket.socket(socket.AF_INET, socket.SOCK_STREAM)
s.connect((hostname, 50008))
print("Program Komunikasi Tentang Data Diri")

while pesan.lower() != "q":
    pesan = raw_input("Perintah :")
    s.send(pesan)
    if pesan.lower() == "keluar":
        response = s.recv(1024)
        print "Jawab: ", response
        s.close()
        break
    elif pesan.lower() != "keluar":
        response = s.recv(1024)
        print "Jawab: ", response
    s.close()
```

## Kegiatan 2. Informasi tentang server



The screenshot shows two side-by-side Python IDE windows. The left window, titled 'Kegiatan 2 Server.py', contains a server script that listens on port 50008 and responds to various commands like 'machine', 'release', 'system', 'version', and 'node'. The right window, titled 'Kegiatan 2 Client.py', contains a client script that connects to the server on port 50008 and sends/receives data. Both scripts use the 'socket' and 'platform' modules. The IDEs are running on a Windows 10 desktop, with the taskbar and system clock visible at the bottom.

```
import socket
import platform

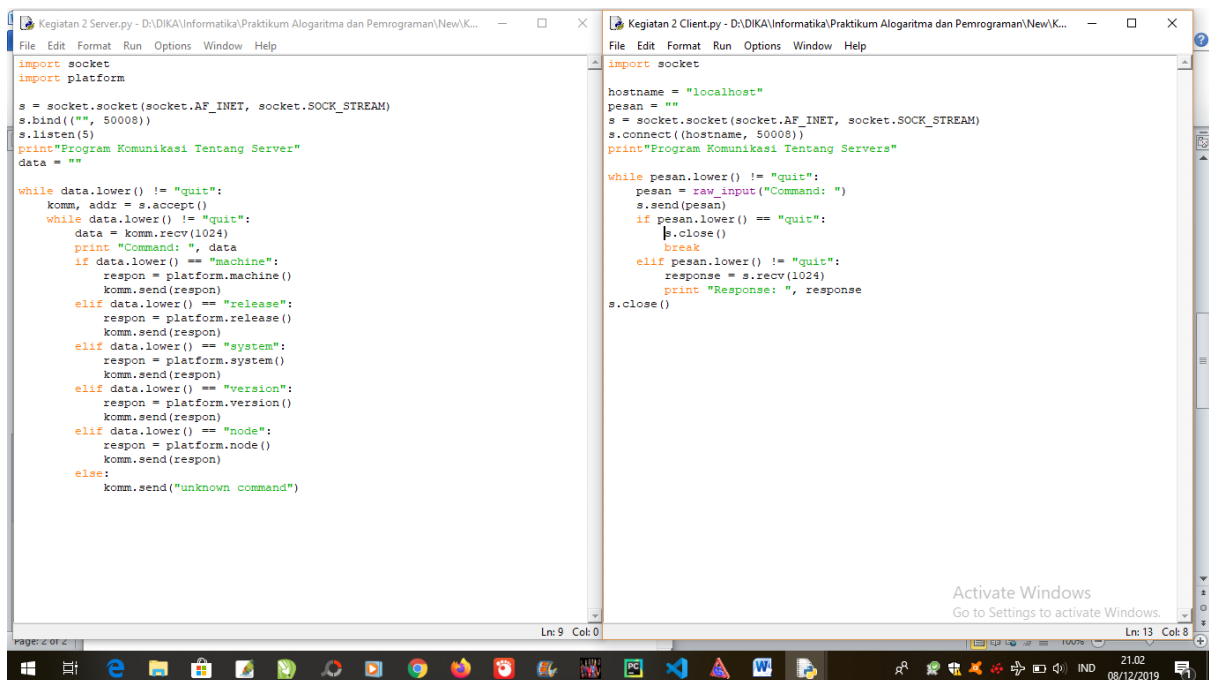
s = socket.socket(socket.AF_INET, socket.SOCK_STREAM)
s.bind(("", 50008))
s.listen(5)
print("Program Komunikasi Tentang Server")
data = ""

while data.lower() != "quit":
    komm, addr = s.accept()
    while data.lower() != "quit":
        data = komm.recv(1024)
        print "Command: ", data
        if data.lower() == "machine":
            respon = platform.machine()
            komm.send(respon)
        elif data.lower() == "release":
            respon = platform.release()
            komm.send(respon)
        elif data.lower() == "system":
            respon = platform.system()
            komm.send(respon)
        elif data.lower() == "version":
            respon = platform.version()
            komm.send(respon)
        elif data.lower() == "node":
            respon = platform.node()
            komm.send(respon)
        else:
            komm.send("unknown command")
```

```
import socket

hostname = "localhost"
pesan = ""
s = socket.socket(socket.AF_INET, socket.SOCK_STREAM)
s.connect((hostname, 50008))
print("Program Komunikasi Tentang Servers")

while pesan.lower() != "quit":
    pesan = raw_input("Command: ")
    s.send(pesan)
    if pesan.lower() == "quit":
        s.close()
        break
    elif pesan.lower() != "quit":
        response = s.recv(1024)
        print "Response: ", response
s.close()
```



This screenshot is identical to the one above, showing the same Python IDE windows with the server and client code. The code is the same as in the first image, and the IDEs are running on the same Windows 10 desktop environment.

```
import socket
import platform

s = socket.socket(socket.AF_INET, socket.SOCK_STREAM)
s.bind(("", 50008))
s.listen(5)
print("Program Komunikasi Tentang Server")
data = ""

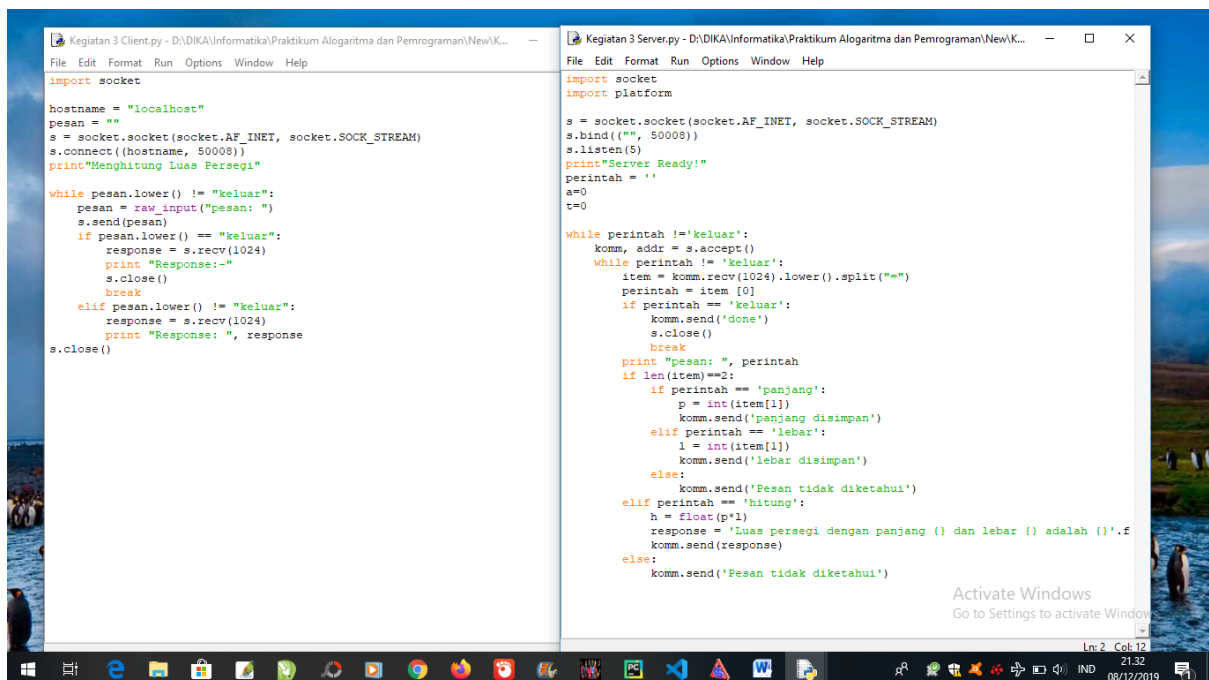
while data.lower() != "quit":
    komm, addr = s.accept()
    while data.lower() != "quit":
        data = komm.recv(1024)
        print "Command: ", data
        if data.lower() == "machine":
            respon = platform.machine()
            komm.send(respon)
        elif data.lower() == "release":
            respon = platform.release()
            komm.send(respon)
        elif data.lower() == "system":
            respon = platform.system()
            komm.send(respon)
        elif data.lower() == "version":
            respon = platform.version()
            komm.send(respon)
        elif data.lower() == "node":
            respon = platform.node()
            komm.send(respon)
        else:
            komm.send("unknown command")
```

```
import socket

hostname = "localhost"
pesan = ""
s = socket.socket(socket.AF_INET, socket.SOCK_STREAM)
s.connect((hostname, 50008))
print("Program Komunikasi Tentang Servers")

while pesan.lower() != "quit":
    pesan = raw_input("Command: ")
    s.send(pesan)
    if pesan.lower() == "quit":
        s.close()
        break
    elif pesan.lower() != "quit":
        response = s.recv(1024)
        print "Response: ", response
s.close()
```

### Kegiatan 3. Menghitung luas bangun geometri



The screenshot shows two Python IDE windows side-by-side. The left window is titled 'Kegiatan 3 Client.py' and the right window is titled 'Kegiatan 3 Server.py'. Both windows show Python code for a socket-based client-server application. The client code (left) connects to localhost:50008 and sends messages to the server. The server code (right) listens on localhost:50008 and responds to the client's requests. The server code includes logic to calculate the area of a square based on the input length and width.

```
import socket

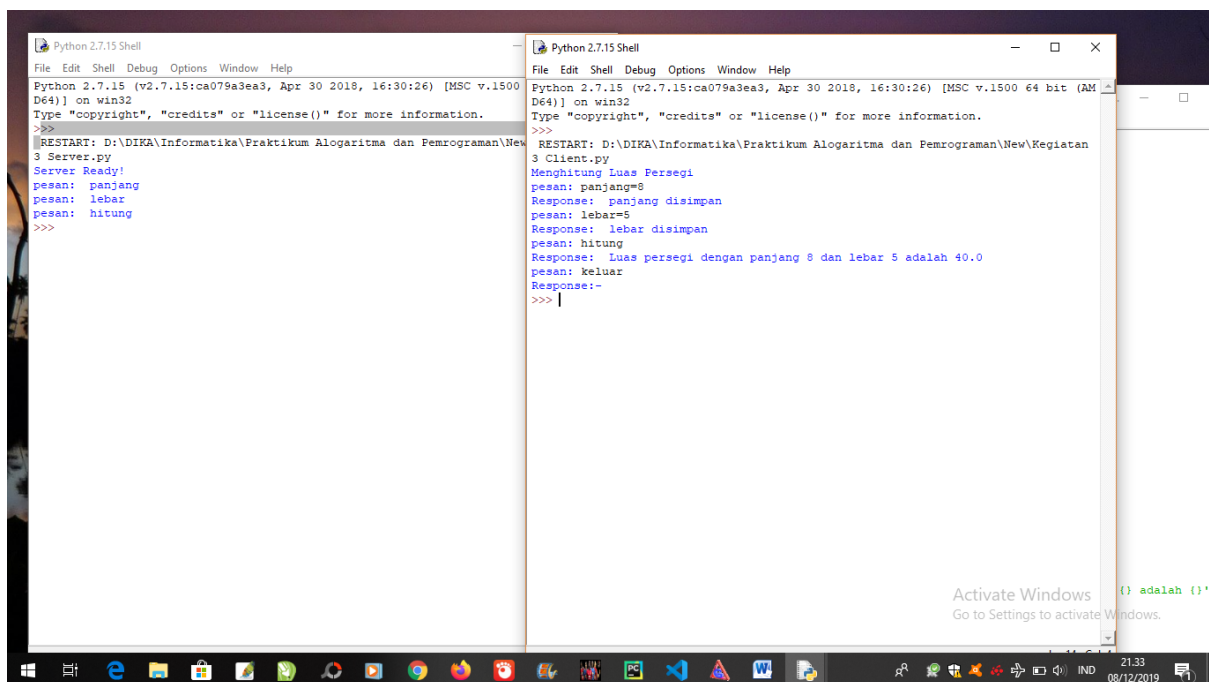
hostname = "localhost"
pesan = ""
s = socket.socket(socket.AF_INET, socket.SOCK_STREAM)
s.connect((hostname, 50008))
print("Menghitung Luas Persegi")

while pesan.lower() != "keluar":
    pesan = raw_input("pesan: ")
    s.send(pesan)
    if pesan.lower() == "keluar":
        response = s.recv(1024)
        print "Response: -"
        s.close()
        break
    elif pesan.lower() != "keluar":
        response = s.recv(1024)
        print "Response: ", response
s.close()
```

```
import socket
import platform

s = socket.socket(socket.AF_INET, socket.SOCK_STREAM)
s.bind(("", 50008))
s.listen(5)
print "Server Ready!"
perintah = ''
a=0
t=0

while perintah != 'keluar':
    komm, addr = s.accept()
    while perintah != 'keluar':
        item = komm.recv(1024).lower().split("=")
        perintah = item[0]
        if perintah == 'keluar':
            komm.send('done')
            s.close()
            break
        print "pesan: ", perintah
        if len(item)==2:
            if perintah == 'panjang':
                p = int(item[1])
                komm.send('panjang disimpan')
            elif perintah == 'lebar':
                l = int(item[1])
                komm.send('lebar disimpan')
            else:
                komm.send('Pesan tidak diketahui')
        elif perintah == 'hitung':
            h = float(p*l)
            response = 'Luas persegi dengan panjang {} dan lebar {} adalah {}'.f
            komm.send(response)
        else:
            komm.send('Pesan tidak diketahui')
```



The screenshot shows two Python 2.7.15 Shell windows side-by-side. The left window shows the execution of the client program, and the right window shows the execution of the server program. The client program sends messages to the server, and the server program responds with the calculated area of the square.

```
Python 2.7.15 Shell
File Edit Shell Debug Options Window Help
Python 2.7.15 (v2.7.15:ca079a3ea3, Apr 30 2018, 16:30:26) [MSC v.1500
D64] on win32
Type "copyright", "credits" or "license()" for more information.
>>>
RESTART: D:\DIKA\Informatika\Praktikum Alogaritma dan Pemrograman\New\
3 Server.py
Server Ready!
pesan: panjang
pesan: lebar
pesan: hitung
>>>
```

```
Python 2.7.15 Shell
File Edit Shell Debug Options Window Help
Python 2.7.15 (v2.7.15:ca079a3ea3, Apr 30 2018, 16:30:26) [MSC v.1500 64 bit (AM
D64)] on win32
Type "copyright", "credits" or "license()" for more information.
>>>
RESTART: D:\DIKA\Informatika\Praktikum Alogaritma dan Pemrograman\New\Regiatan
3 Client.py
Menghitung Luas Persegi
pesan: panjang=8
Response: panjang disimpan
pesan: lebar=5
Response: lebar disimpan
pesan: hitung
Response: Luas persegi dengan panjang 8 dan lebar 5 adalah 40.0
pesan: keluar
Response: -
>>>
```