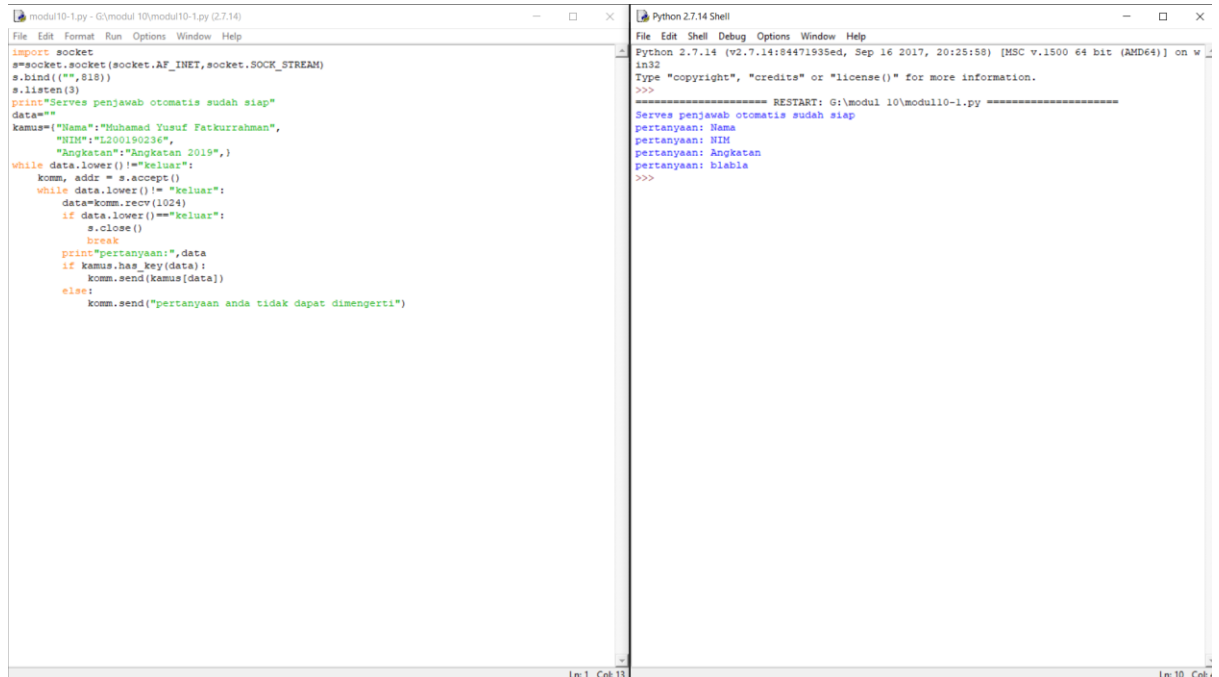


Nama : Muhamad Yusuf Fatkurrahman

NIM : L200190236

Kegiatan 1

a. Server



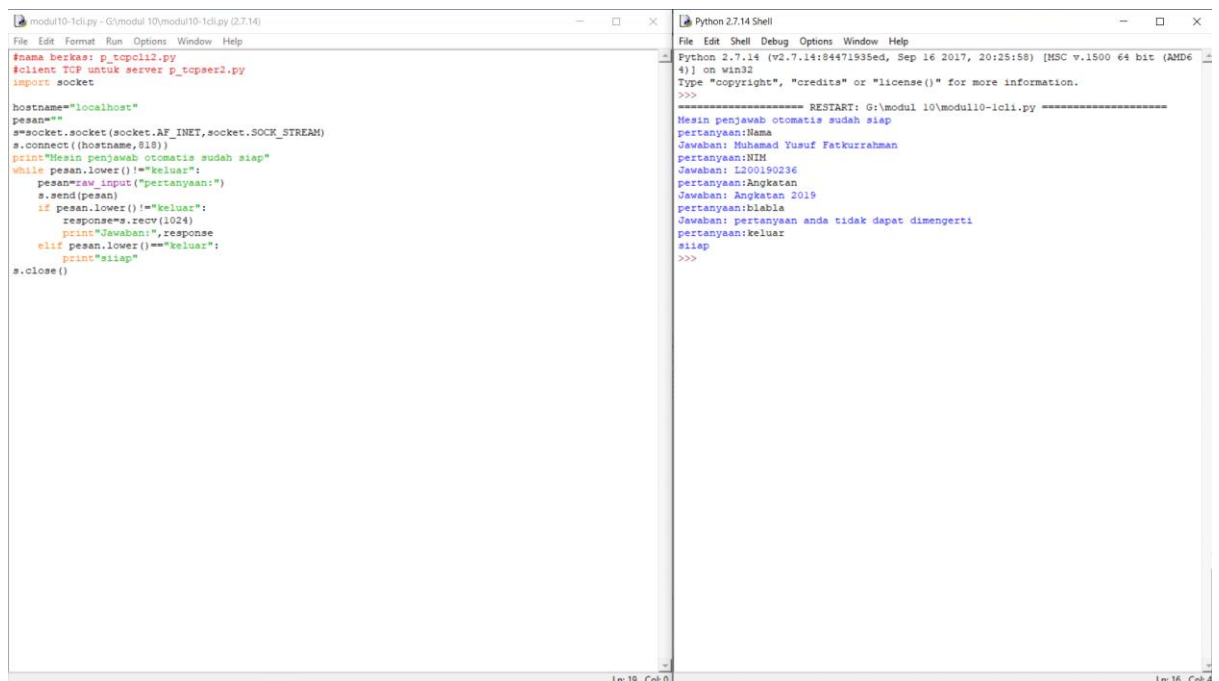
```
modul10-1.py - G:\modul 10\modul10-1.py (2.7.14)
File Edit Format Run Options Window Help

import socket
s=socket.socket(socket.AF_INET,socket.SOCK_STREAM)
s.bind(('',818))
s.listen(3)
print "Server penjawab otomatis sudah siap"
data=""
kamus={"Nama":"Muhamad Yusuf Fatkurrahman",
       "NIM":"L200190236",
       "Angkatan":"Angkatan 2019",}
while data.lower()!="keluar":
    komm, addr = s.accept()
    while data.lower()!="keluar":
        data=komm.recv(1024)
        if data.lower()!="keluar":
            s.close()
            break
        print "pertanyaan:",data
        if kamus.has_key(data):
            komm.send(kamus[data])
        else:
            komm.send("pertanyaan anda tidak dapat dimengerti")

Python 2.7.14 Shell
File Edit Shell Debug Options Window Help

Python 2.7.14 (v2.7.14:84471935ed, Sep 16 2017, 20:25:58) [MSC v.1500 64 bit (AMD64)] on win32
Type "copyright", "credits" or "license()" for more information.
>>>
===== RESTART: G:\modul 10\modul10-1.py =====
Server penjawab otomatis sudah siap
pertanyaan: Nama
pertanyaan: NIM
pertanyaan: Angkatan
pertanyaan: blabla
>>>
```

b. Client



```
modul10-1cli.py - G:\modul 10\modul10-1cli.py (2.7.14)
File Edit Format Run Options Window Help

#nama berkas: p_topcli2.py
#client TCP untuk server p_topser2.py
import socket

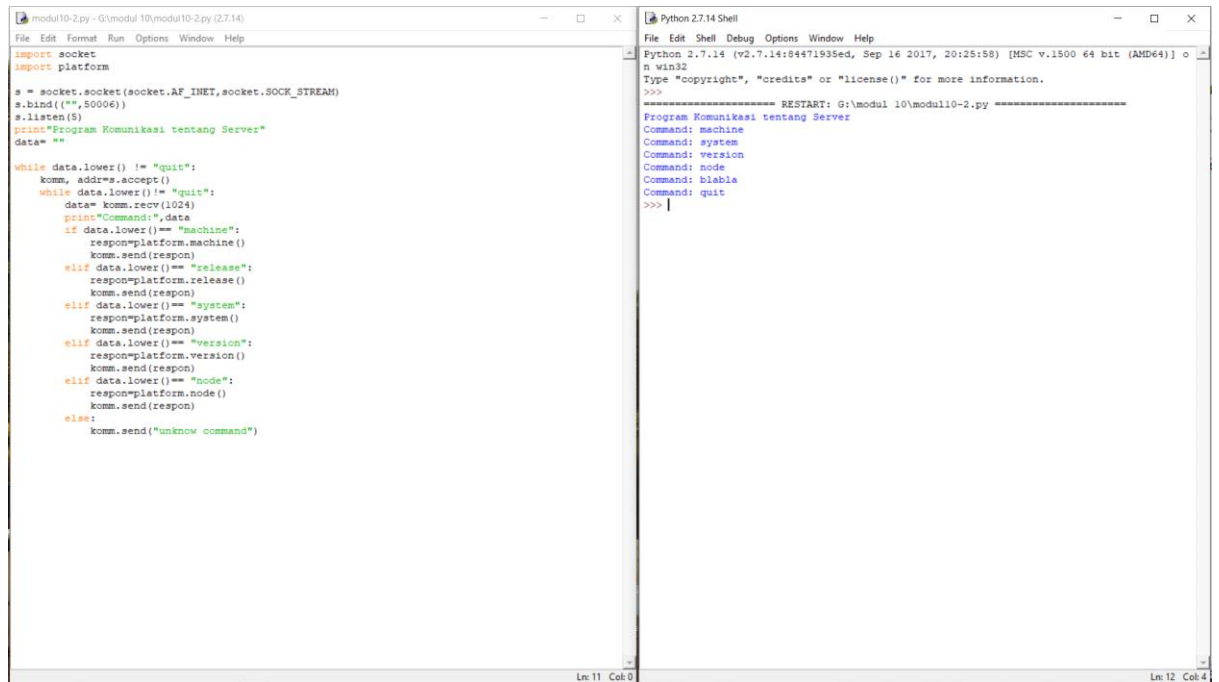
hostname="localhost"
pesan=""
s=socket.socket(socket.AF_INET,socket.SOCK_STREAM)
s.connect((hostname,818))
print "Mesin penjawab otomatis sudah siap"
while pesan.lower()!="keluar":
    pesan=raw_input("pertanyaan:")
    s.send(pesan)
    if pesan.lower()!="keluar":
        response=s.recv(1024)
        print "Jawabani", response
    elif pesan.lower()!="keluar":
        print "siap"
s.close()

Python 2.7.14 Shell
File Edit Shell Debug Options Window Help

Python 2.7.14 (v2.7.14:84471935ed, Sep 16 2017, 20:25:58) [MSC v.1500 64 bit (AMD64)] on win32
Type "copyright", "credits" or "license()" for more information.
>>>
===== RESTART: G:\modul 10\modul10-1cli.py =====
Mesin penjawab otomatis sudah siap
pertanyaan: Nama
Jawaban: Muhamad Yusuf Fatkurrahman
pertanyaan: NIM
Jawaban: L200190236
pertanyaan: Angkatan
Jawaban: Angkatan 2019
pertanyaan: blabla
Jawaban: pertanyaan anda tidak dapat dimengerti
pertanyaan: keluar
siap
>>>
```

Kegiatan 2

a. Server



The screenshot displays a Python 2.7.14 IDE with two windows. The left window, titled 'modul10-2.py - G:\modul 10\modul10-2.py (2.7.14)', contains a server script. The script imports 'socket' and 'platform', binds to port 50006, and listens for connections. It handles commands like 'machine', 'release', 'system', 'version', 'node', and 'node' by sending platform information. The right window, titled 'Python 2.7.14 Shell', shows the output of running the script. It displays the program's title, a restart message, and the results of various commands: 'machine' returns 'AMD64', 'system' returns 'Windows', 'version' returns '10.0.18362', 'node' returns 'DESKTOP-GLK9VH6', 'blabla' returns 'unknown command', and 'quit' returns an empty string.

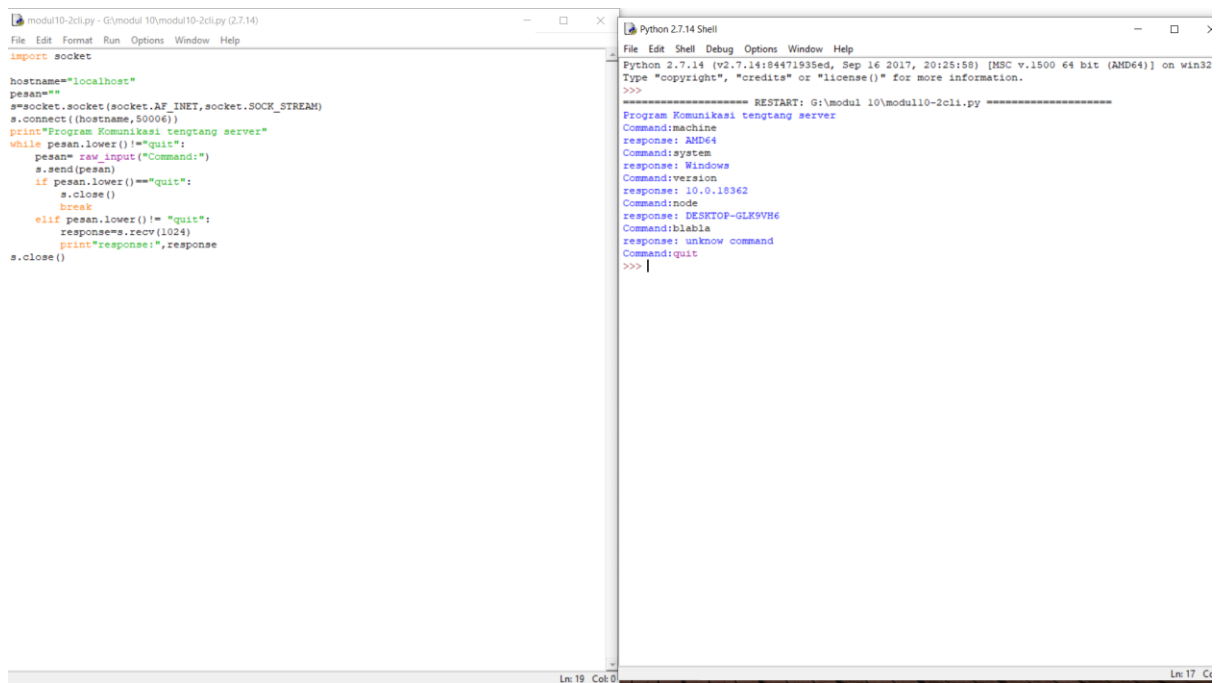
```
import socket
import platform

s = socket.socket(socket.AF_INET, socket.SOCK_STREAM)
s.bind(('', 50006))
s.listen(5)
print "Program Komunikasi tentang Server"
data = ""

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

```
Python 2.7.14 (v2.7.14:84471935ed, Sep 16 2017, 20:25:58) [MSC v.1500 64 bit (AMD64)] o
n win32
Type "copyright", "credits" or "license()" for more information.
>>>
===== RESTART: G:\modul 10\modul10-2.py =====
Program Komunikasi tentang Server
Command: machine
Command: system
Command: version
Command: node
Command: blabla
Command: quit
>>> |
```

b. Client



The screenshot displays a Python 2.7.14 IDE with two windows. The left window, titled 'modul10-2cli.py - G:\modul 10\modul10-2cli.py (2.7.14)', contains a client script. The script connects to 'localhost' on port 50006, sends a message, and then enters a loop where it prompts the user for a command and sends it to the server. The right window, titled 'Python 2.7.14 Shell', shows the output of running the script. It displays the program's title, a restart message, and the results of various commands: 'machine' returns 'AMD64', 'system' returns 'Windows', 'version' returns '10.0.18362', 'node' returns 'DESKTOP-GLK9VH6', 'blabla' returns 'unknown command', and 'quit' returns an empty string.

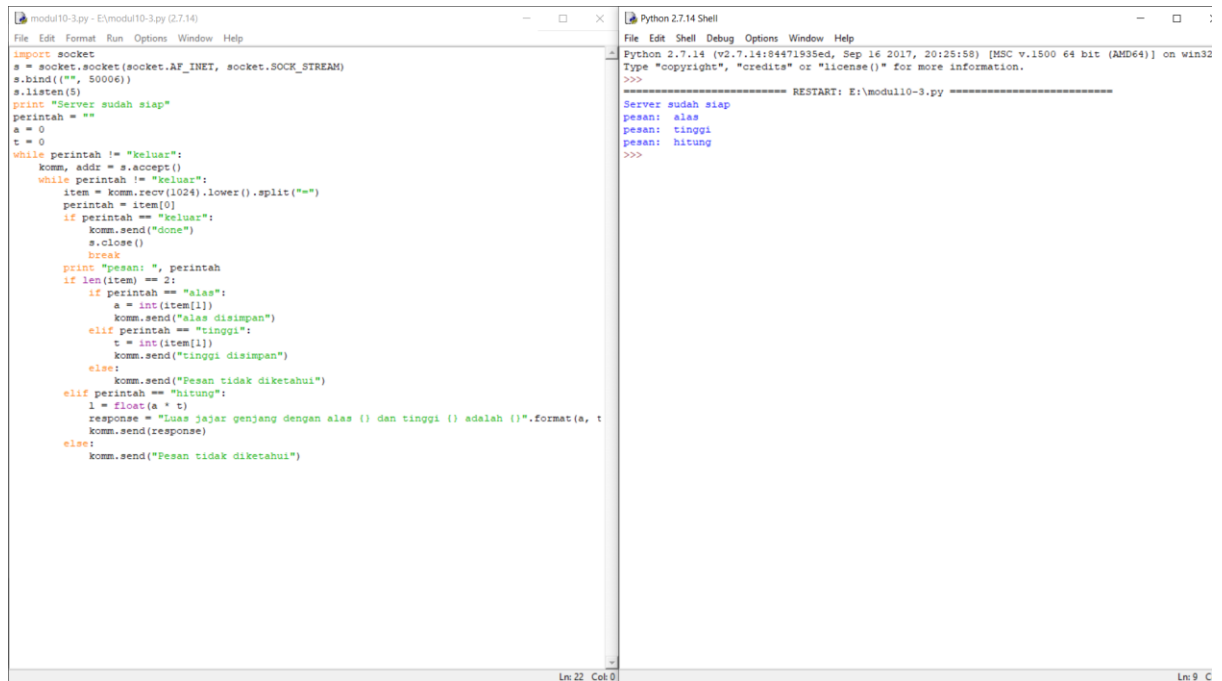
```
import socket

hostname = "localhost"
pesan = ""
s = socket.socket(socket.AF_INET, socket.SOCK_STREAM)
s.connect((hostname, 50006))
print "Program Komunikasi tentang server"
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()
```

```
Python 2.7.14 (v2.7.14:84471935ed, Sep 16 2017, 20:25:58) [MSC v.1500 64 bit (AMD64)] on win32
Type "copyright", "credits" or "license()" for more information.
>>>
===== RESTART: G:\modul 10\modul10-2cli.py =====
Program Komunikasi tentang server
Command: machine
response: AMD64
Command: system
response: Windows
Command: version
response: 10.0.18362
Command: node
response: DESKTOP-GLK9VH6
Command: blabla
response: unknown command
Command: quit
>>> |
```

Kegiatan 3

a. Server

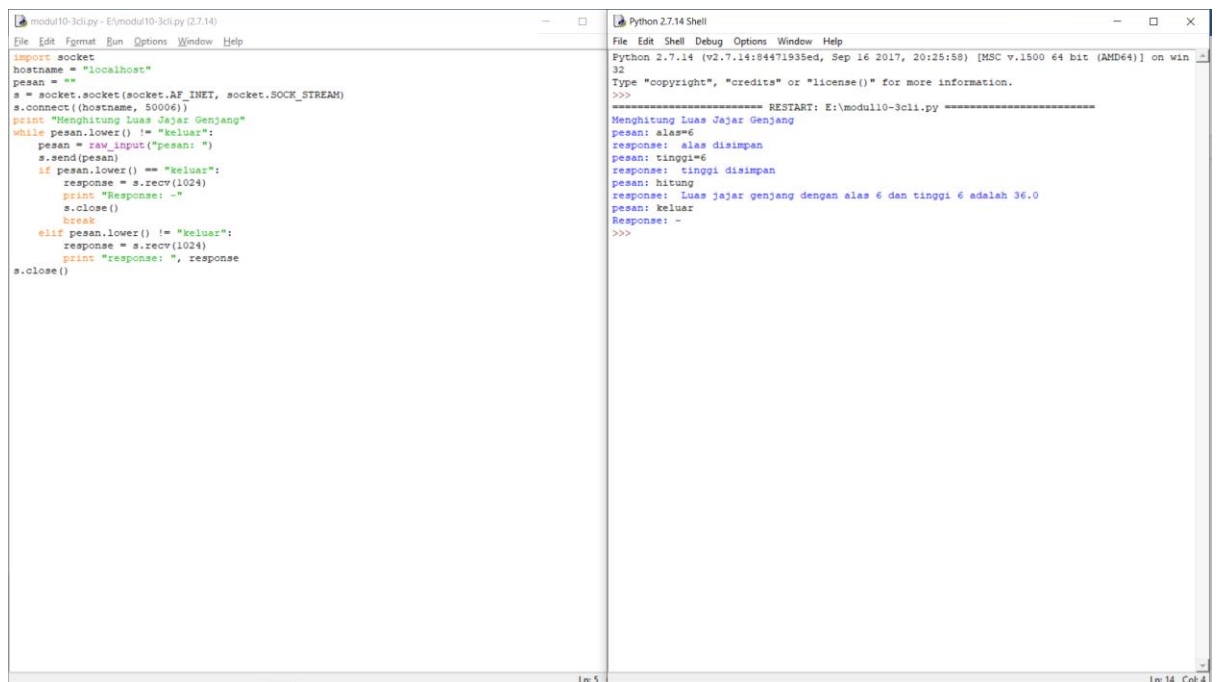


The screenshot shows a Python 2.7.14 IDE with two windows. The left window displays the code for a server script named `modul10-3.py`. The code uses the `socket` module to create a server that listens on port 50006. It handles incoming connections and processes commands: `alas` (sends area), `tinggi` (sends height), and `hitung` (calculates the area of a parallelogram based on the previous inputs). The right window shows the output of the script, indicating it has started and is ready to receive connections. The output also shows the restart command: `RESTART: E:\modul10-3.py`.

```
import socket
s = socket.socket(socket.AF_INET, socket.SOCK_STREAM)
s.bind(('', 50006))
s.listen(5)
print "Server sudah siap"
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 == "alas":
                a = int(item[1])
                komm.send("alas disimpan")
            elif perintah == "tinggi":
                t = int(item[1])
                komm.send("tinggi disimpan")
            else:
                komm.send("Pesan tidak diketahui")
        elif perintah == "hitung":
            l = float(a * t)
            response = "Luas jajar genjang dengan alas () dan tinggi () adalah {}".format(a, t)
            komm.send(response)
        else:
            komm.send("Pesan tidak diketahui")
```

```
Python 2.7.14 Shell
Python 2.7.14 (v2.7.14:84471935ed, Sep 16 2017, 20:25:58) [MSC v.1500 64 bit (AMD64)] on win32
Type "copyright", "credits" or "license()" for more information.
>>>
===== RESTART: E:\modul10-3.py =====
Server sudah siap
pesan: alas
pesan: tinggi
pesan: hitung
>>>
```

b. Client



The screenshot shows a Python 2.7.14 IDE with two windows. The left window displays the code for a client script named `modul10-3cli.py`. The code connects to the server at `localhost` on port 50006. It prompts the user for input and sends commands: `alas`, `tinggi`, and `hitung`. The right window shows the output of the script, indicating it has connected to the server and is ready to receive responses. The output also shows the restart command: `RESTART: E:\modul10-3cli.py`.

```
import socket
hostname = "localhost"
pesan = ""
s = socket.socket(socket.AF_INET, socket.SOCK_STREAM)
s.connect((hostname, 50006))
print "Menghitung Luas Jajar Genjang"
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()
```

```
Python 2.7.14 Shell
Python 2.7.14 (v2.7.14:84471935ed, Sep 16 2017, 20:25:58) [MSC v.1500 64 bit (AMD64)] on win32
Type "copyright", "credits" or "license()" for more information.
>>>
===== RESTART: E:\modul10-3cli.py =====
Menghitung Luas Jajar Genjang
pesan: alas=6
response: alas disimpan
pesan: tinggi=6
response: tinggi disimpan
pesan: hitung
response: Luas jajar genjang dengan alas 6 dan tinggi 6 adalah 36.0
pesan: keluar
Response: -
>>>
```

