Nama: Mifthahul Hoiri Bachrudin Basir

NIM: H071221072

```
import socket
hostname = socket.gethostname()
print("Ip address: %s" %socket.gethostbyname(hostname))
```

```
import socket
hostname = socket.gethostname()
print("Host name: %s" %hostname)
```

```
import socket
UDP_IP = "127.0.0.1"
UDP_PORT = 5102
print ("UDP target IP:", UDP_IP)
print ("UDP target port:", UDP_PORT)
# print ("message:", MESSAGE)
sock = socket.socket(socket.AF_INET, # Internet
socket.SOCK_DGRAM) # UDP
sock.connect((UDP_IP, UDP_PORT))
while 1:
   pesan = input('pesan anda : ')
   sock.send(pesan.encode())
```

```
import argparse, socket
from datetime import datetime
MAX_BYTES = 65535
def server(port):
   sock = socket.socket(socket.AF_INET, socket.SOCK_DGRAM)
   sock.bind(('127.0.0.1', port))
   print('Listening at {}'.format(sock.getsockname()))
```

```
    while True:
        data, address = sock.recvfrom(MAX_BYTES)
        text = data.decode('ascii')
        print('The client at {} says {!r}'.format(address, text))
        text = 'Your data was {} bytes long'.format(len(data))
        data = text.encode('ascii')
        sock.sendto(data, address)
def client(port):
    sock = socket.socket(socket.AF_INET, socket.SOCK_DGRAM)
    text = 'The time is {}'.format(datetime.now())
    data = text.encode('ascii')
    sock.sendto(data, ('127.0.0.1', port))
    print('The OS assigned me the address {}'.format(sock.getsockname()))
    data, address = sock.recvfrom(MAX_BYTES) # Danger! See Chapter 2
    text = data.decode('ascii')
    print('The server {} replied {!r}'.format(address, text))
if __name__ == '__main__':
    choices = {'client': client, 'server': server}
    parser = argparse.ArgumentParser(description='Send and receive UDP locally')
    parser.add_argument('role', choices=choices, help='which role to play')
    parser.add_argument('-p', metavar='PORT', type=int, default=1060, help='UDP, port (default
1060)')

    args = parser.parse_args()
    function = choices[args.role]
    function(args.p)


import argparse, random, socket, sys
MAX_BYTES = 65535
def server(interface, port):
    sock = socket.socket(socket.AF_INET, socket.SOCK_DGRAM)
```

```
    sock.bind((interface, port))
    print('Listening at', sock.getsockname())
    while True:
        data, address = sock.recvfrom(MAX_BYTES)
        if random.random() < 0.5:
            print('Pretending to drop packet from {}'.format(address))
            continue
        text = data.decode('ascii')
        print('The client at {} says {!r}'.format(address, text))
        message = 'Your data was {} bytes long'.format(len(data))
        sock.sendto(message.encode('ascii'), address)

def client(hostname, port):
    sock = socket.socket(socket.AF_INET, socket.SOCK_DGRAM)
    hostname = sys.argv[2]
    sock.connect((hostname, port))
    print('Client socket name is {}'.format(sock.getsockname()))
    delay = 0.1 # seconds
    text = 'This is another message'
    data = text.encode('ascii')
    while True:
        sock.send(data)
        print('Waiting up to {} seconds for a reply'.format(delay))
        sock.settimeout(delay)
        try:
            data = sock.recv(MAX_BYTES)
        except socket.timeout:
            delay *= 2 # wait even longer for the next request
            if delay > 2.0:
                raise RuntimeError('I think the server is down')
            else:
```

```
            break # we are done, and can stop looping
            print('The server says {!r}'.format(data.decode('ascii')))
if __name__ == '__main__':
    choices = {'client': client, 'server': server}
    parser = argparse.ArgumentParser(description='Send and receive UDP, pretending packets are often dropped')

    parser.add_argument('role', choices=choices, help='which role to take')
    parser.add_argument('host', help='interface the server listens at; host the client sends to')

    parser.add_argument('-p', metavar='PORT', type=int, default=1060, help='UDP port (default 1060)')

args = parser.parse_args()
function = choices[args.role]
function(args.host, args.p)
```

Ip address: 192.168.1.11

Host name: Vin

UDP target IP: 127.0.0.1

UDP target port: 5102

pesan anda : Aku

pesan anda : dan

pesan anda : Kamu