WORKSHOP PEMROGRAMAN JARINGAN
MODUL 5
(APLIKASI CHATTING UDP + OPERASI FILE)

Mochammad Zen Samsono Hadi, ST. MSc. Ph.D

TOPIK PEMBAHASAN

- Operasi file: write, open file
- Membuat aplikasi berbasis UDP
- Membuat aplikasi chatting seperti whatsapp
  - Berbasis UDP
  - Operasi file

Operasi File

- File handling: open() function
- Syntax:

```
f = open (“namaFile.txt”,”modeFile”)
```

- Mode File:
  - "r" - Read - Opens a file for reading, error if the file does not exist
  - "a" - Append - Opens a file for appending, creates the file if it does not exist
  - "w" - Write - Opens a file for writing, creates the file if it does not exist
  - "x" - Create - Creates the specified file, returns an error if the file exists
- Referensi:

  https://www.w3schools.com/Python/python_file_handling.asp

Operasi File

corona.txt

Stay at home !
Lakukan social distancing

Open & read file

```
f = open("corona.txt", "r")
print(f.read())
```

Close file

```
f = open("corona.txt", "r")
print(f.read())
f.close()
```

- Terdapat beberapa cara untuk membaca file:
  - f.read(5) => hanya membaca 5 karakter awal
  - f.readline() => membaca 1 baris data dari sebuah file

Operasi File: write

- Menulis data ke file yang ada, maka menambahkan parameter di fungsi open():
  - "a" - Append - will append to the end of the file
  - "w" - Write - will overwrite any existing content
- Contoh program

```
f = open("corona.txt", "a")
f.write("Semoga wabah ini segera diangkat oleh Alloh")
f.close()

#open and read the file after the appending:
f = open("corona.txt", "r")
print(f.read())
```

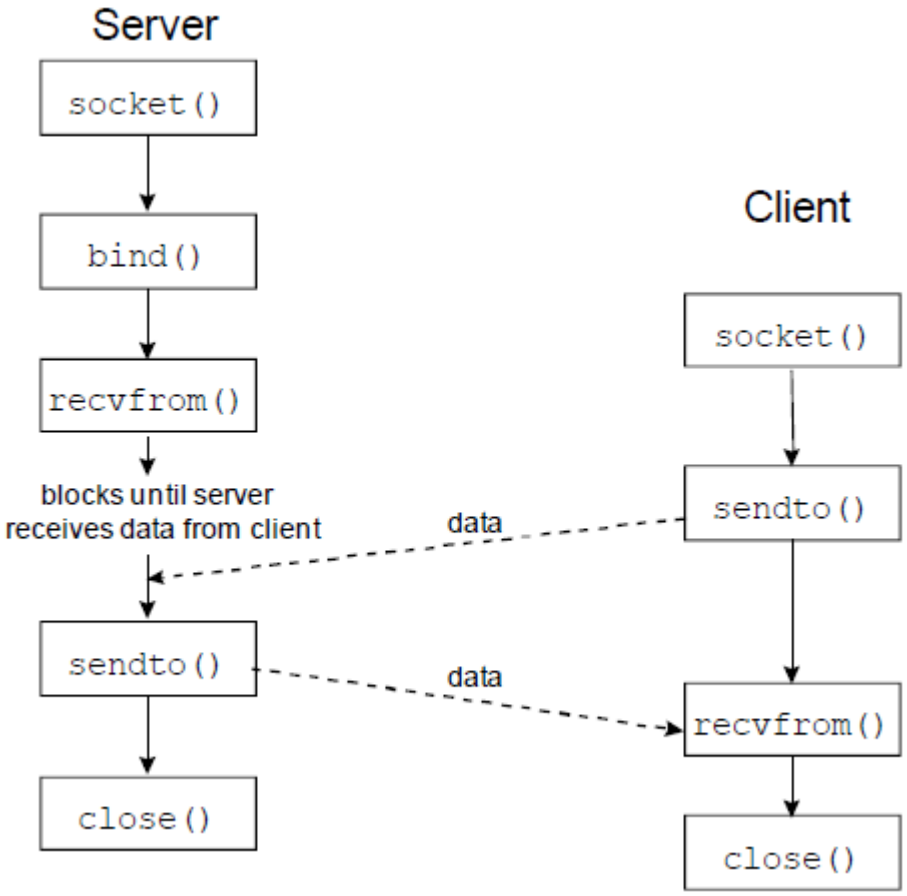
- Gantilah "a" dengan "w" dan lihat bedanya

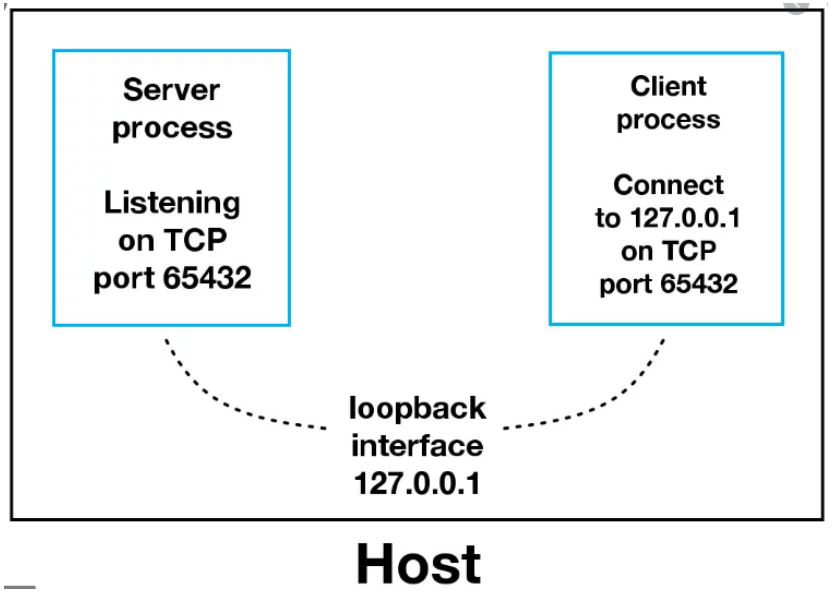
Membuat aplikasi berbasis UDP

- Koneksi client – server
  - Harus mengetahui IP dan port tujuan
  - Aplikasi tersebut berbasis TCP / UDP

Aplikasi UDP

- Dengan cara yang hampir sama, buat aplikasi berbasis UDP pada listing 1.14





Server (Listing 1-14)

```
import socket
import sys
import argparse

host = 'localhost'
data_payload = 2048

def echo_server(port):
    """ A simple echo server """
    # Create a UDP socket
    sock = socket.socket(socket.AF_INET,
                         socket.SOCK_DGRAM)

    # Bind the socket to the port
    server_address = (host, port)
    print ("Starting up echo server
            on %s port %s" % server_address)

    sock.bind(server_address)

    while True:
        print ("Waiting to receive message
                  from client")
        data, address = sock.
                        recvfrom(data_payload)
        print ("received %s bytes
                from %s" % (len(data), address))
        print ("Data: %s" %data)
        if data:
            sent = sock.sendto(data, address)
            print ("sent %s bytes back
                    to %s" % (sent, address))
```

```
if __name__ == '__main__':
    parser = argparse.ArgumentParser
             (description='Socket Server Example')
    parser.add_argument('--port', action="store", dest="port", type=int,
    required=True)
        given_args = parser.parse_args()
        port = given_args.port
        echo_server(port)
```

Client

```
host = 'localhost'
data_payload = 2048

def echo_client(port):
    """ A simple echo client """
    # Create a UDP socket
    sock = socket.socket(socket.AF_INET,
                         socket.SOCK_DGRAM)

    server_address = (host, port)
    print ("Connecting to %s port %s" % server_address)
    message = 'This is the message.  It will be
              repeated.'

    try:

        # Send data
        message = "Test message. This will be
                  echoed"
        print ("Sending %s" % message)
        sent = sock.sendto(message.encode
               ('utf-8'), server_address)

        # Receive response
        data, server = sock.recvfrom(data_payload)
        print ("received %s" % data)
    finally:
        print ("Closing connection to the server")
        sock.close()
```

Perhatikan sock.sendto, format data harus diubah ke universal code 8 bit (ascii)

received b'Test message. This will be echoed'

Agar karakter b’ hilang, tambahkan: data.decode()

Aplikasi UDP

Client

IP: localhost
Port: 43542

Server

IP: localhost
Port: 9900

```
$ python 1_14a_echo_server_udp.py --port=9900
Starting up echo server on localhost port 9900
Waiting to receive message from client
```

```
$ python 1_14b_echo_client_udp.py --port=9900
Connecting to localhost port 9900
Sending Test message. This will be echoed
received Test message. This will be echoed
Closing connection to the server
```

```
received 33 bytes from ('127.0.0.1', 43542)
Data: Test message. This will be echoed
sent 33 bytes back to ('127.0.0.1', 43542)
Waiting to receive message from client
```

TUGAS

- Buatlah aplikasi chatting berbasis UDP antara 2 komputer
- Simpan data dalam bentuk file

Client

IP: 172.10.3.4

Text file

Menu Pilihan:

1. Koneksi ke server
2. Melihat history

Server

IP: 200.1.3.4
Port: 9900

Menunggu koneksi

TUGAS

Jika pilihan 1:

Masukkan IP Server: 200.1.3.4
Masukkan port server: 9900
Koneksi berhasil

Client: Halo Server
Msg dari server: ini dari server
Client: Awas corona, STAY AT HOME
Msg dari server: siap
Client: quit
Data disimpan

Jika pilihan 2:

Masukkan nama file: 200.1.3.4.txt
Pesan:

Client: Halo Server
Server: ini dari server
Client: Awas corona, STAY AT HOME
Server: siap

Koneksi dari client: 172.10.3.4

Msg dari client: Halo Server
Server: ini dari server
Msg dari client: Awas corona, STAY AT HOME
Server: siap