

*kelas jarkoma*

KELAS JARKOMA

# TUGAS SOCKET PROGRAMMING TCP/UDP

TUGAS JARINGAN KOMPUTER

TUGAS KELOMPOK



*DIBUAT OLEH*

*BENHARD SIMANULLANG*

*24060120140136*

*JULIUS ADRIAN*

*24060120130127*

# PROTOKOL TCP

- TCP/IP adalah standar komunikasi data yang digunakan oleh komunitas internet dalam proses tukar-menukar data dari satu komputer ke komputer lain di dalam jaringan Internet. Protokol ini tidaklah dapat berdiri sendiri, karena memang protokol ini berupa kumpulan protokol (protocol suite). Protokol ini juga merupakan protokol yang paling banyak digunakan saat ini. Data tersebut diimplementasikan dalam bentuk perangkat lunak (software) di sistem operasi. Istilah yang diberikan kepada perangkat lunak ini adalah TCP/IP stack.

# THREE WAY HANDSHAKE



Three-way-handshake merupakan mekanisme yang diterapkan dalam TCP/IP untuk memastikan paket data dapat terkirim dengan baik. Sebagaimana kita ketahui bahwa protokol TCP bersifat connection-oriented sehingga paket data tidak akan dikirimkan kecuali setelah pengirim dan penerima sudah sepakat dan siap untuk proses transfer data.



# CODE TCP

```
> python client.py > ...
1 from http import client, server
2 from socket import *
3 serverName = "localhost"
4 serverPort = 12001
5 clientSocket = socket(AF_INET, SOCK_STREAM)
6 clientSocket.connect((serverName, serverPort))
7 sentence = input("Nama Anggota Kelompok : ")
8 clientSocket.send(sentence.encode())
9 modifiedSentence = clientSocket.recv(1024)
0 print('From Server : ', modifiedSentence.decode())
1 clientSocket.close()
```



```
> python server.py > ...
from multiprocessing import connection
from socket import *
from sqlite3 import connect
serverPort = 12001
serverSocket = socket(AF_INET, SOCK_STREAM)
serverSocket.bind(('',serverPort))
serverSocket.listen(1)
print(" The server is ready to receive ")
while True:
    connectionSocket, addr = serverSocket.accept()
    sentence = connectionSocket.recv(1024).decode()
    capitalizedSentence = sentence.upper()
    connectionSocket.send(capitalizedSentence.encode())
    connectionSocket.close()
```

client

server

code akan berjalan pada port 12001, client server tersebut memiliki fungsionalitas dimana, pada saat mengirim huruf kecil pada client, server akan mengembalikan string dalam huruf kapital, menggunakan huruf kapital

# CONTOH APLIKASI PROTOKOL TCP

```
PS D:\Document\tugas-socket\tcp> python server.py
Nama Anggota Kelompok : benhard_simanullang_24060120140136 julius_adrian_
24060120130127
From Server : BENHARD_SIMANULLANG_24060120140136 JULIUS_ADRIAN_240601201
30127
PS D:\Document\tugas-socket\tcp>
```

input client : benhard\_simanullang\_24060120140136 Julius\_adrian\_24060120130127  
( dalam huruf kecil )

output server : BENHARD\_SIMANULLANG\_24060120140136 BENHARD\_SIMANULLANG\_24060120130127  
( dalam huruf kapital)

# PROTOKOL UDP

- UDP adalah kepanjangan dari User Datagram Protocol. Salah satu protocol yang menjadi lapisan transport TCP/IP sehingga mendukung atau memungkinkan terjadinya komunikasi yang tidak andal (unreliable), tanpa adanya koneksi antar host jaringan yang menggunakan TCP/IP..
- Melalui UDP, aplikasi maupun software dapat mengirimkan pesan kepada komputer lain di dalam jaringan lain tanpa perlu melakukan komunikasi awal. UDP menerapkan komunikasi secara sederhana dengan mekanisme yang minimalis. Terdapat proses checksum guna memproteksi integritas data.

# CODE UDP

```
udp > ⚒ client.py > ...
1  from socket import *
2  serverName = 'localhost'
3  serverPort = 12000
4  clientSocket = socket(AF_INET, SOCK_DGRAM)
5  clientSocket.connect((serverName, serverPort))
6  message=input("Nama Anggota Kelompok : ")
7  clientSocket.sendto(message.encode(),(serverName,serverPort))
8  modifiedMessage, serverAddress = clientSocket.recvfrom(2048)
9  print('From Server : ', modifiedMessage.decode())
10 clientSocket.close()
```

client

code akan berjalan pada port 12000, client server tersebut memiliki fungsionalitas dimana, pada saat mengirim huruf kecil pada client, server akan mengembalikan string dalam huruf kapital, menggunakan protokol udp

```
udp > ⚒ server.py > ...
1  from email import message
2  from socket import *
3  serverPort = 12000
4  serverSocket = socket(AF_INET, SOCK_DGRAM)
5  serverSocket.bind(("" ,serverPort))
6  print("The server is ready to receive")
7  while True :
8      message, clientAddress = serverSocket.recvfrom(2048)
9      modifiedMessage=message.decode().upper()
10     serverSocket.sendto(modifiedMessage.encode(),clientAddress)
```

server

# CONTOH APLIKASI PROTOKOL UDP

```
PS C:\Users\Julius'\Downloads\socket-python> cd udp
PS C:\Users\Julius'\Downloads\socket-python\udp> python -u .\client.py
Nama Anggota Kelompok : benhard_simanullang_24060120140136 julius_adrian_24060120130127
From Server : BENHARD_SIMANULLANG_24060120140136 JULIUS_ADRIAN_24060120130127
PS C:\Users\Julius'\Downloads\socket-python\udp> 
```

input : benhard\_simanullang\_24060140136 julius\_adrian\_24060120130127 ( dalam huruf kecil )  
output : BENHARD\_SIMANULLANG\_24060120140136 JULIUS\_ADRIAN\_24060120130127 (dalam huruf kapital )

*LINK VIDEO : HTTPS://YOUTUBE/NBLHBVA-0JG*



TERIMA KASIH