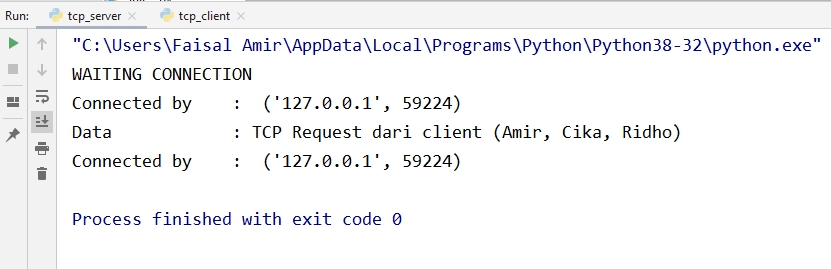
Analisis TCP Server

|  |  |
| --- | --- |
| CODE | FUNGSI (PENJELASAN) |
| **import** socket **from** com.frogobox.base.config **import** \* | Import function untuk memanggil fungsi socket |
| *# definisikan alamat IP binding yang akan digunakan* IP\_ADDR = BASE\_CONFIG\_IP\_ADDR  *# definisikan port number binding yang akan digunakan* PORT = BASE\_CONFIG\_PORT  *# definisikan ukuran buffer untuk mengirimkan pesan* BUFFER = BASE\_CONFIG\_BUFFER ADDR = (IP\_ADDR, PORT) | Deklarasi IP Address, Port, Buffer dan Address. |
| print(BASE\_MESSAGE\_CONNECTING) | Deklarasi Connection |
| *# buat socket bertipe TCP* **with** socket.socket(  socket.AF\_INET,  socket.SOCK\_STREAM ) **as** tcpServer:  *# lakukan bind* tcpServer.bind(ADDR)   *# server akan listen menunggu hingga ada koneksi dari client* tcpServer.listen()   *# menerima koneksi* connection, address = tcpServer.accept()   **with** connection:   *# lakukan loop forever* **while True**:   *# menampilkan koneksi berupa IP dan port client yang terhubung menggunakan print* print(**'Connected by \t: '**, address)   *# menerima data berdasarkan ukuran buffer* data = connection.recv(BUFFER)   **if not** data:  **break** *# menampilkan pesan yang diterima oleh server menggunakan print,* print(**'Data \t\t\t:'**, str(data, **'utf-8'**))   *# mengirim kembali data yang diterima dari client kepada client* connection.send(data) |  |
| *# tutup koneksi* tcpServer.close() |  |

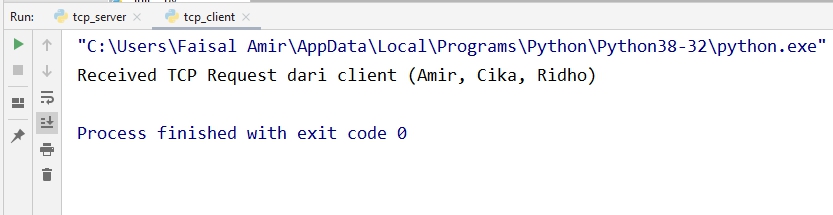
Screenshoot



Analisis TCP Client

|  |  |
| --- | --- |
| CODE | FUNGSI (PENJELASAN) |
| *# import library socket karena akan menggunakan IPC socket* **import** socket **from** com.frogobox.base.config **import** \* | Import function untuk memanggil fungsi socket |
| *# definisikan tujuan IP server* SERVER\_ADDR = BASE\_CONFIG\_IP\_ADDR  *# definisikan port dari server yang akan terhubung* SERVICE\_PORT = BASE\_CONFIG\_PORT  *# definisikan ukuran buffer untuk mengirimkan pesan* BUFFER = BASE\_CONFIG\_BUFFER  *# definisikan pesan yang akan disampaikan* MESSAGE = **'TCP '** + MESSAGE\_REQUEST |  |
| *# buat socket TCP* tcpClient = socket.socket(socket.AF\_INET, socket.SOCK\_STREAM) |  |
| *# lakukan koneksi ke server dengan parameter IP dan Port yang telah didefinisikan* tcpClient.connect((SERVER\_ADDR, SERVICE\_PORT)) |  |
| *# kirim pesan ke server, pesan bebas, dan ditambahkan nama anggota kelompok* tcpClient.send(MESSAGE.encode()) |  |
| *# terima pesan dari server* data = tcpClient.recv(BUFFER) |  |
| *# tampilkan pesan/reply dari server* print(**'Received'**, data.decode()) |  |
| *# tutup koneksi* tcpClient.close() |  |

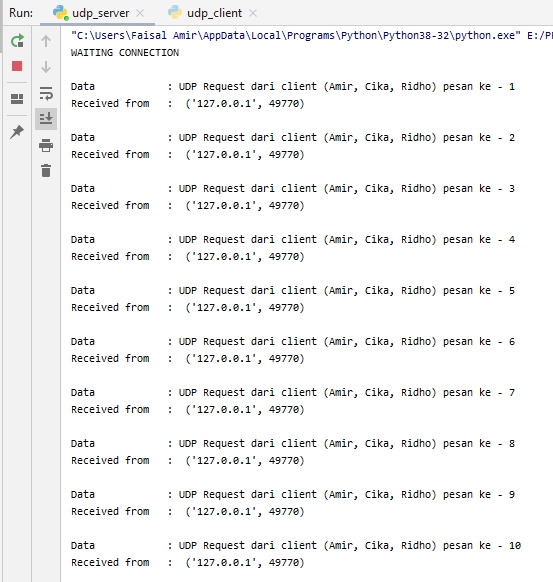
Screenshoot



Analisis UDP Server

|  |  |
| --- | --- |
| CODE | FUNGSI (PENJELASAN) |
| *# import library socket karena akan menggunakan IPC socket* **import** socket **from** com.frogobox.base.config **import** \* | Import function untuk memanggil fungsi socket |
| *# Berikan tampilan (print bahwa socket berhasil dibuat)* print(BASE\_MESSAGE\_CONNECTING) print() |  |
| *# definisikan alamat IP bind dari server* IP\_ADDR = BASE\_CONFIG\_IP\_ADDR  *# definisikan port number untuk bind dari server* PORT = BASE\_CONFIG\_PORT BUFFER = BASE\_CONFIG\_BUFFER ADDR = (IP\_ADDR, PORT) |  |
| *# buat socket bertipe UDP* **with** socket.socket(  socket.AF\_INET,  socket.SOCK\_DGRAM ) **as** udpServer:  *# lakukan bind* udpServer.bind(ADDR)   *# Berikan tampilan (print) bahwa socket bind ke alamat IP dengan port yang telah didapatkan   # loop forever* **while True**:  *# terima pesan dari client* data, address = udpServer.recvfrom(BUFFER)   **if not** data:  **break** *# menampilkan hasil pesan dari client* print(**'Data \t\t\t:'**, data.decode())   *# print (addr)* print(**'Received from \t: '**, address)  print() |  |
| udpServer.close() |  |

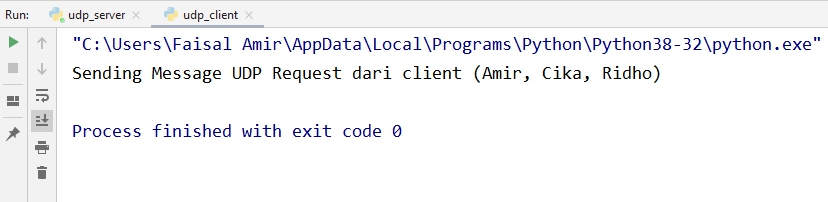
Screenshoot



Analisis UDP Client

|  |  |
| --- | --- |
| CODE | FUNGSI (PENJELASAN) |
| *# import library socket karena akan menggunakan IPC socket* **import** socket **from** com.frogobox.base.config **import** \* | Import function untuk memanggil fungsi socket |
| *# definisikan target IP server yang akan dituju* UDP\_IP = BASE\_CONFIG\_IP\_ADDR  *# definisikan target port number server yang akan dituju* UDP\_PORT = BASE\_CONFIG\_PORT TARGET = (UDP\_IP, UDP\_PORT) MESSAGE = **'UDP '** + MESSAGE\_REQUEST |  |
| print(**'Sending Message '** + MESSAGE) |  |
| *# buat socket bertipe UDP* **with** socket.socket(  socket.AF\_INET,  socket.SOCK\_DGRAM ) **as** udpClient:  udpClient.connect(TARGET)  **for** x **in** range(10):  udpClient.sendto(str(MESSAGE + **' pesan ke - '** + str(x + 1)).encode(), TARGET) |  |
| udpClient.close() |  |

Screenshoot



Telkom University / IFX - 43 – GAB

Friskadini Ismayanti (1301198496)

Muhammad Faisal Amir (1301198497)

Ridho Maulana Cahyudi (1301198515)