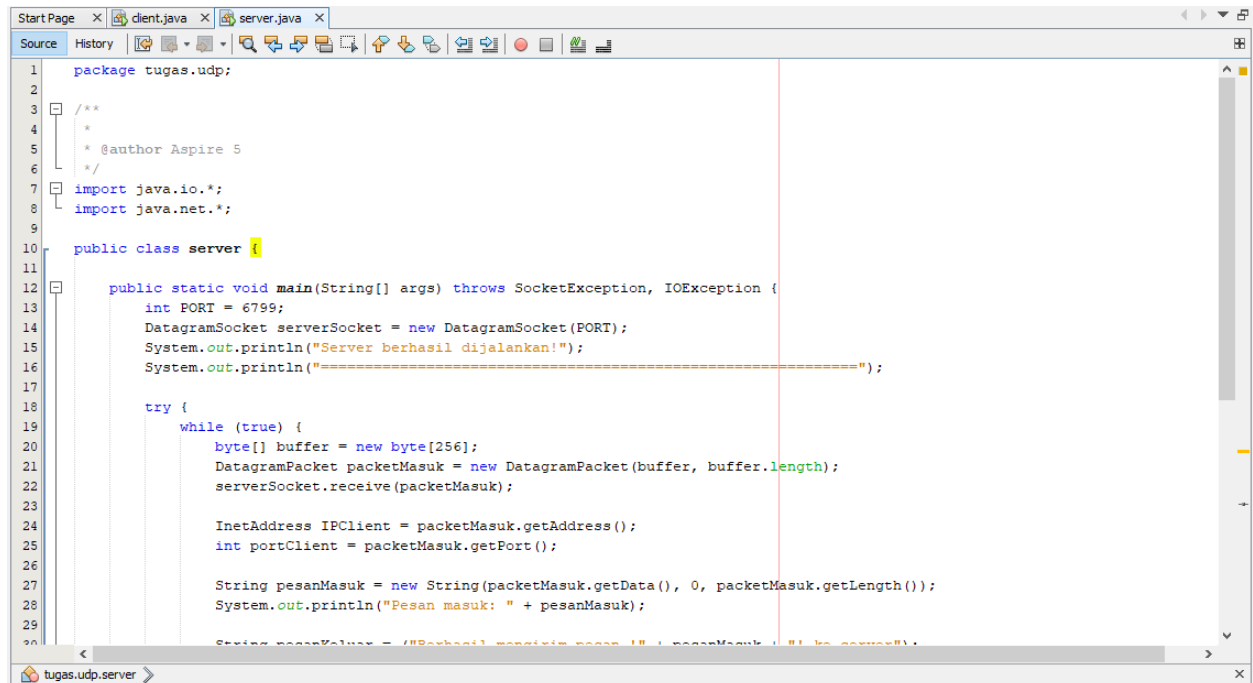


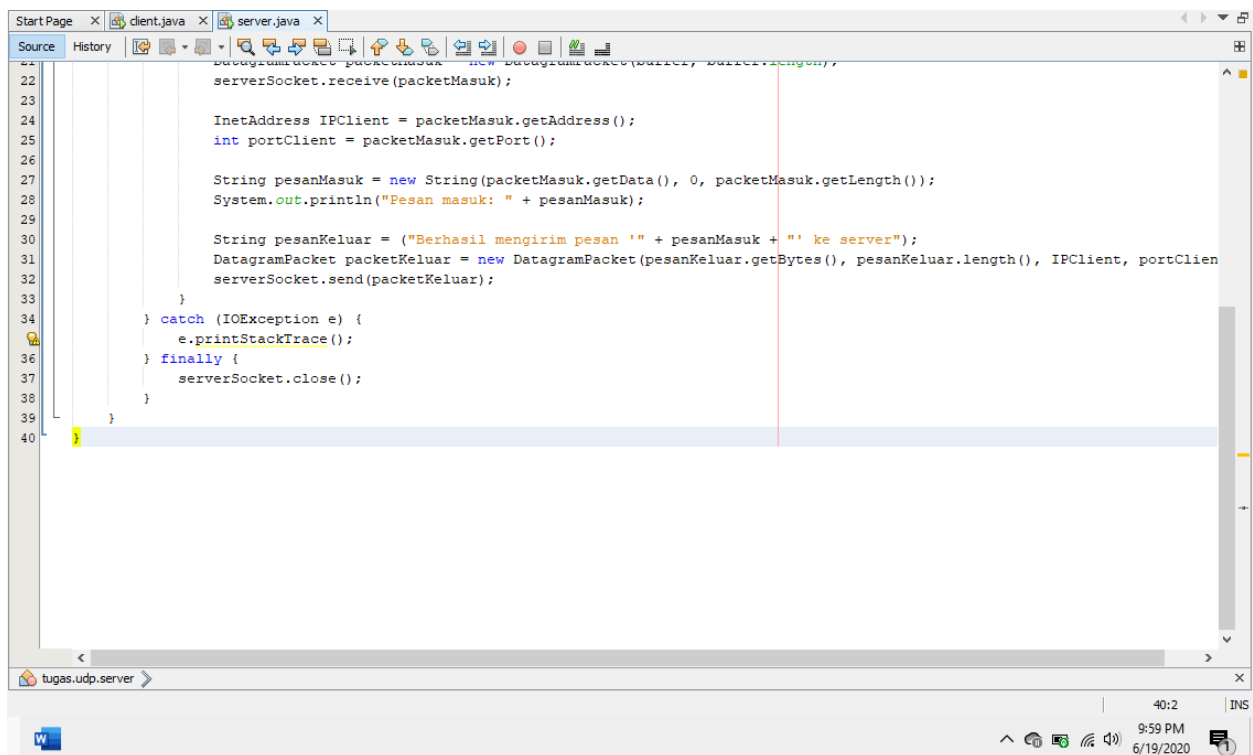
Nama : Dimas Widyatama

Npm : 06.2017.1.06887

Source Code Server



```
1 package tugas.udp;
2
3 /**
4  *
5  * @author Aspire 5
6  */
7 import java.io.*;
8 import java.net.*;
9
10 public class server {
11
12     public static void main(String[] args) throws SocketException, IOException {
13         int PORT = 6799;
14         DatagramSocket serverSocket = new DatagramSocket(PORT);
15         System.out.println("Server berhasil dijalankan!");
16         System.out.println("=====");
17
18         try {
19             while (true) {
20                 byte[] buffer = new byte[256];
21                 DatagramPacket packetMasuk = new DatagramPacket(buffer, buffer.length);
22                 serverSocket.receive(packetMasuk);
23
24                 InetAddress IPClient = packetMasuk.getAddress();
25                 int portClient = packetMasuk.getPort();
26
27                 String pesanMasuk = new String(packetMasuk.getData(), 0, packetMasuk.getLength());
28                 System.out.println("Pesan masuk: " + pesanMasuk);
29
30                 String pesanKeluar = ("Berhasil mengirim pesan " + pesanMasuk + " ke server");
```



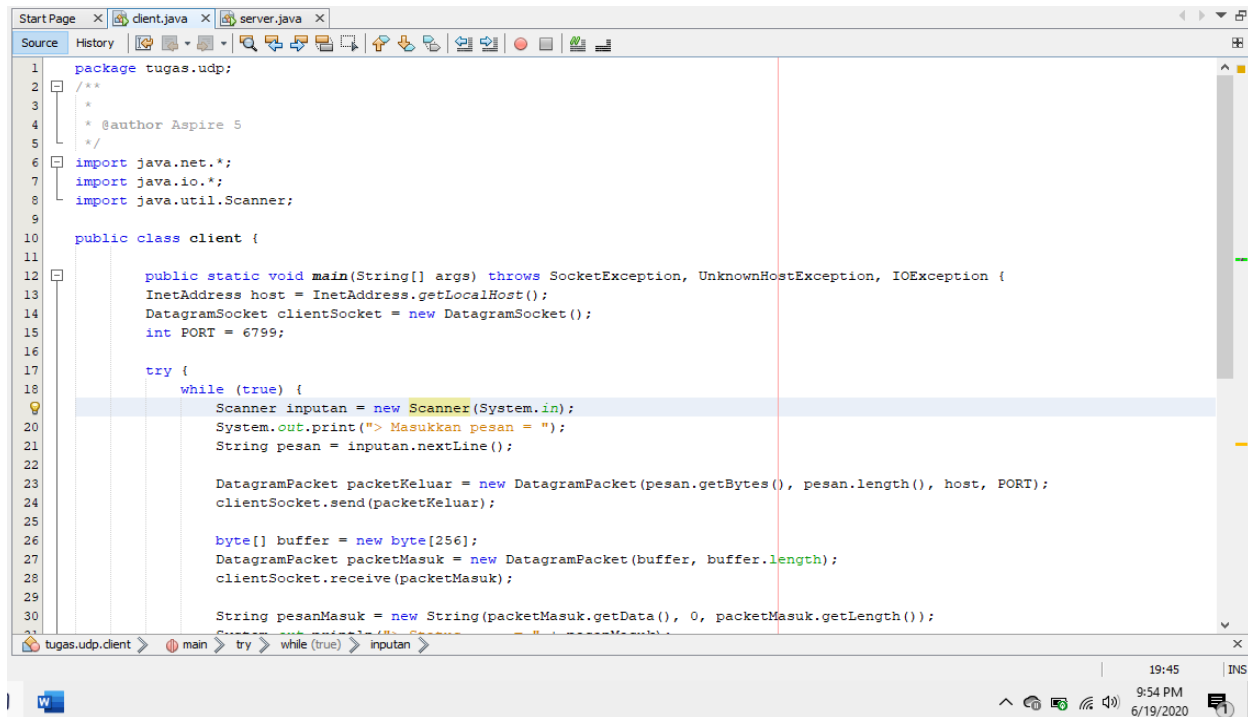
```
31                 DatagramPacket packetKeluar = new DatagramPacket(pesanKeluar.getBytes(), pesanKeluar.length(), IPClient, portClient);
32                 serverSocket.send(packetKeluar);
33             }
34         } catch (IOException e) {
35             e.printStackTrace();
36         } finally {
37             serverSocket.close();
38         }
39     }
40 }
```

Nama : Dimas Widyatama

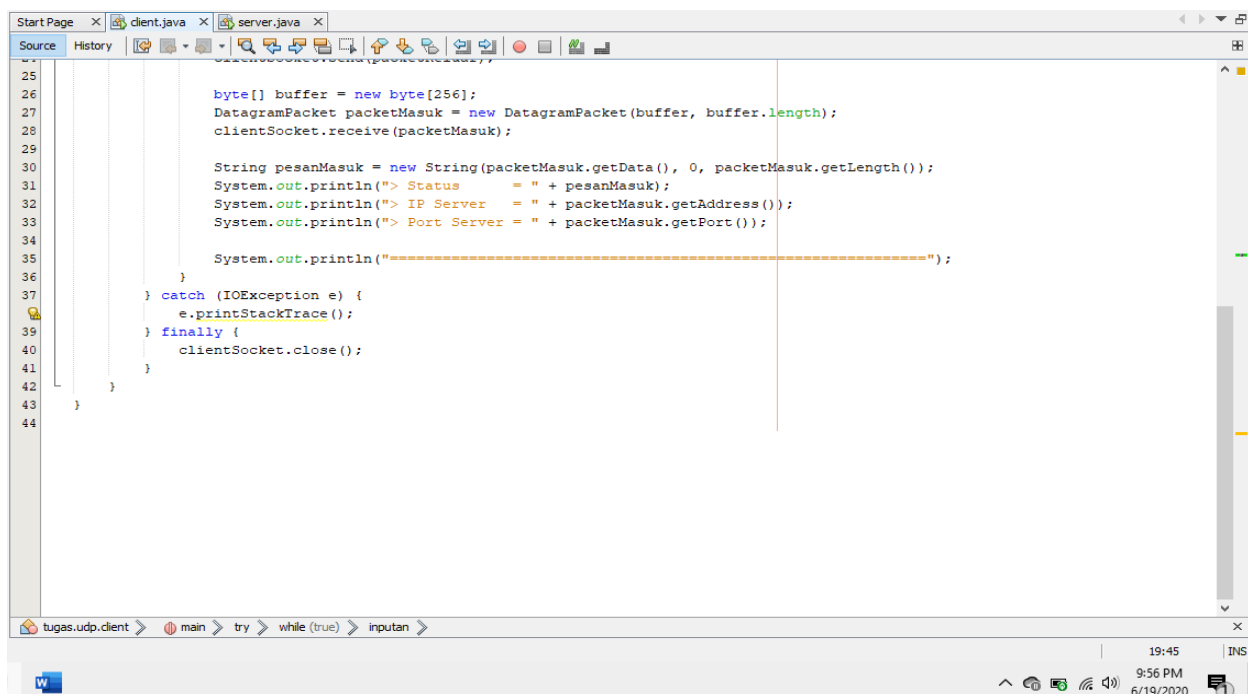
Npm : 06.2017.1.06887

Penjelasan Server : Di bagian ini terdapat deklarasi port yang nantinya akan digunakan dan akan menjalankan server dengan port yang sudah di deklarasikan. Setelahnya server akan mengambil port dan alamat ip client setelah mendapatkannya server akan menerima pesan yang masuk melalui method `getData()` dan `getLength` untuk mengambil panjang data dan setelahnya data akan ditampilkan

Source Code Client



```
1 package tugas.udp;
2
3 /**
4  * @author Aspire 5
5  */
6 import java.net.*;
7 import java.io.*;
8 import java.util.Scanner;
9
10 public class client {
11
12     public static void main(String[] args) throws SocketException, UnknownHostException, IOException {
13         InetAddress host = InetAddress.getLocalHost();
14         DatagramSocket clientSocket = new DatagramSocket();
15         int PORT = 6799;
16
17         try {
18             while (true) {
19                 Scanner inputan = new Scanner(System.in);
20                 System.out.print("> Masukkan pesan = ");
21                 String pesan = inputan.nextLine();
22
23                 DatagramPacket packetKeluar = new DatagramPacket(pesan.getBytes(), pesan.length(), host, PORT);
24                 clientSocket.send(packetKeluar);
25
26                 byte[] buffer = new byte[256];
27                 DatagramPacket packetMasuk = new DatagramPacket(buffer, buffer.length);
28                 clientSocket.receive(packetMasuk);
29
30                 String pesanMasuk = new String(packetMasuk.getData(), 0, packetMasuk.getLength());
```



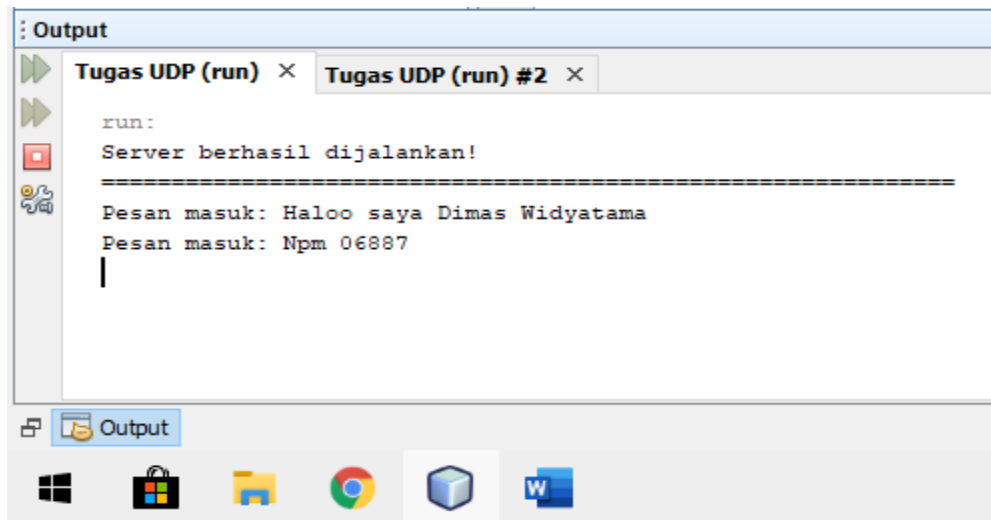
```
31         System.out.println("> Status      = " + pesanMasuk);
32         System.out.println("> IP Server   = " + packetMasuk.getAddress());
33         System.out.println("> Port Server = " + packetMasuk.getPort());
34
35         System.out.println("=====");
36     }
37     catch (IOException e) {
38         e.printStackTrace();
39     } finally {
40         clientSocket.close();
41     }
42 }
43
44 }
```

Nama : Dimas Widyatama

Npm : 06.2017.1.06887

Penjelasan Client : Di bagian ini client akan mengambil data host lokal dan setelahnya akan mendeklarasikan port yang digunakan dan setelah itu masukan inputan yang akan dimasukan. Setelahnya akan menyimpan data yang dikirim ke server berdasarkan host dan port, setelah menerima data yang masuk akan menampilkan pesan yang dikirim oleh server dan akan menampilkan alamat ip dan port yang digunakan

Hasil Output Server



The screenshot shows the VS Code Output window with two tabs: 'Tugas UDP (run)' and 'Tugas UDP (run) #2'. The 'Tugas UDP (run)' tab is active and displays the following text:

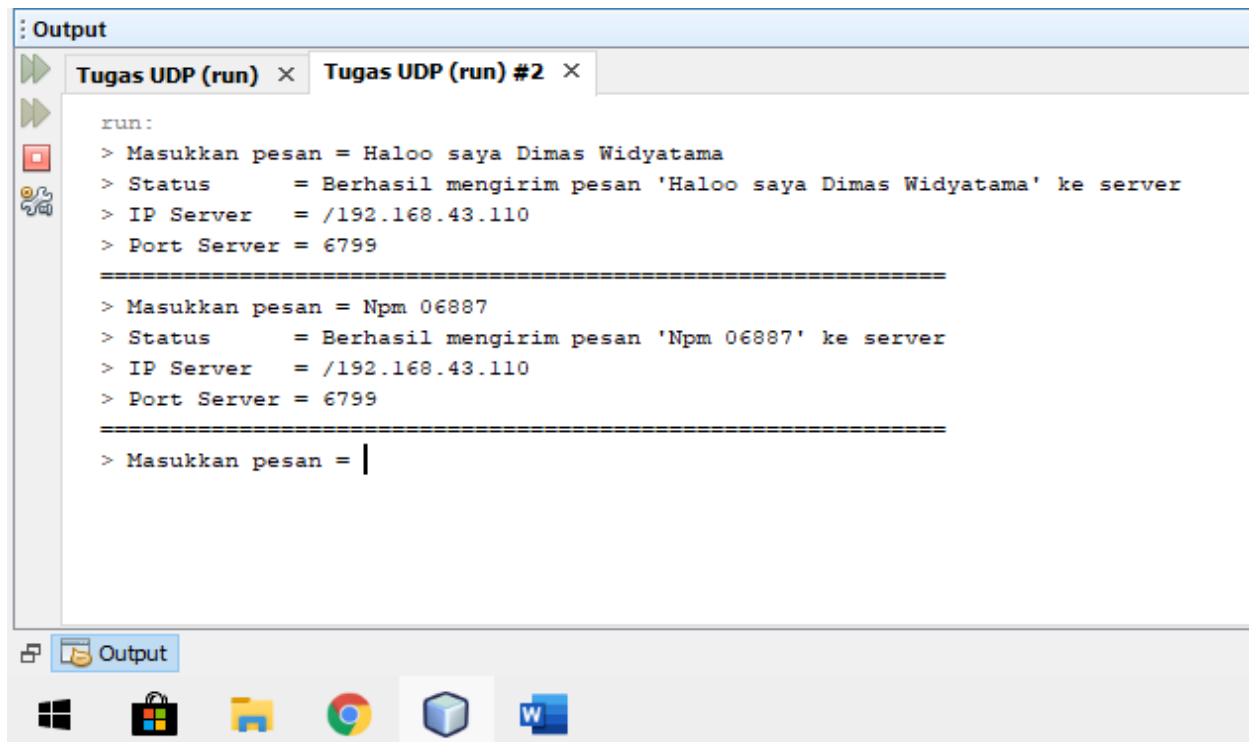
```
run:
Server berhasil dijalankan!

=====

Pesan masuk: Haloo saya Dimas Widyatama
Pesan masuk: Npm 06887
|
```

The taskbar at the bottom shows icons for Windows, VS Code, File Explorer, Google Chrome, Docker, and Microsoft Word.

Hasil Output Client



The screenshot shows the VS Code Output window with two tabs: 'Tugas UDP (run)' and 'Tugas UDP (run) #2'. The 'Tugas UDP (run)' tab is active and displays the following text:

```
run:
> Masukkan pesan = Haloo saya Dimas Widyatama
> Status      = Berhasil mengirim pesan 'Haloo saya Dimas Widyatama' ke server
> IP Server   = /192.168.43.110
> Port Server = 6799

=====

> Masukkan pesan = Npm 06887
> Status      = Berhasil mengirim pesan 'Npm 06887' ke server
> IP Server   = /192.168.43.110
> Port Server = 6799

=====

> Masukkan pesan = |
```

The taskbar at the bottom shows icons for Windows, VS Code, File Explorer, Google Chrome, Docker, and Microsoft Word.