# Persiapan In Class Project

## Deskripsi Umum

Pada bagian ini akan dicontohkan mengenai interaksi antara dua jenis client websocket yaitu yang berbasis web dan yang berbasis console, dengan server websocket. Terdapat 3 file yaitu :

1. simple\_websocket\_server.py

Implementasi websocket di sisi server

1. browser\_websocket\_client.html

Implementasi websocket client di browser

1. python\_websocket\_client.py

Implementasi websocket client di Python

## Cara Kerja

Jalankan server websocket. Ketika server tersebut dijalankan, maka dia akan menunggu apakah ada client yang akan melakukan permintaan koneksi.

Jalankan client browser dengan membukanya di browser. Client browser akan membuat koneksi ke server websocket.

Ketika sebuah koneksi berhasil dibuat, maka server akan menambahkan koneksi tersebut sebuah **variabel global berupa list** dengan nama **clients**.

|  |
| --- |
| # List untuk menampung koneksi  **clients = []**  # Definisikan class untuk protocol yg akan kita buat  class EchoServer(protocol.Protocol) :  # Callback func ketika koneksi berhasil dibuat  def connectionMade(self) :  print "Koneksi baru berhasil dibuat"  # Jika terkoneksi, tambahkan client ke list koneksi  **clients.append(self)** |

Koneksi tersebut akan tetap di-maintain sampai client melakukan pemutusan koneksi yang ditandai dengan penghapusan koneksi dari list **clients**.

|  |
| --- |
| # Definisikan class untuk protocol yg akan kita buat  class EchoServer(protocol.Protocol) :  def connectionLost(self, reason) :  print "Koneksi putus"  # Jika koneksi putus, hapus client dari list koneksi  **clients.remove(self)** |

Jalankan client websocket Python. Prosedur penambahan koneksi yang sama akan dilakukan di sisi server.

Ketika server menerima message dari sebuah client, misalkan dari client websocket Python, maka dia akan melakukan iterasi terhadap semua koneksi yang ada di list clients dan mengirimkan message balasan ke semua client.

|  |
| --- |
| class Echo(protocol.Protocol):  """This is just about the simplest possible protocol"""    def dataReceived(self, data):  print "Receive : "+data  # Kirim data untuk setiap koneksi yang ada  **for cl in clients :**  **cl.transport.write("Broadcast "+data)** |

## Running

1. Install library txWS

sudo pip install txWS

1. Install library websocket client

sudo pip install websocket-client

1. Jalankan **simple\_websocket\_server.py**. Port yang dipakai adalah port 8877
2. Jalankan **browser\_websocket\_client.html** di browser. Pastikan client terkoneksi.
3. Jalankan **python\_websocket\_client.py** dan pastikan server dan client browser menerima message.