**Tugas IPC**

1. **Tugas 1: Koneksi TCP**

Terdapat 2 buah komputer yang bertindak sebagai client dan server. Client dan server tersebut berkomunikasi menggunakan socket TCP. Client akan mengirimkan sebuah pesan ke server. Pesan tersebut akan dikirimkan kembali ke client oleh server. Client akan menerima replay pesan dari server dan menampilkannya.

1. Download skeleton source yang diberikan!
2. Lengkapi source code sehingga program bisa berjalan!
3. Jalankan program client dan server pada 2 laptop yang berbeda!
4. Kumpulkan source code yang dibuat serta screenshoot hasil program yang berjalan!
5. **Tugas 2: Koneksi UDP**

Terdapat 2 buah komputer yang bertindak sebagai client dan server. Client dan server tersebut berkomunikasi menggunakan socket UDP. Client akan mengirimkan pesan ke server sebanyak 10 kali. Pesan tersebut akan diterima server dan ditampilkan pada server.

1. Download skeleton source yang diberikan!
2. Lengkapi source code sehingga program bisa berjalan!
3. Jalankan program client dan server pada 2 laptop yang berbeda!
4. Kumpulkan source code yang dibuat serta screenshoot hasil program yang berjalan!
5. **Tugas 3: upload file**

Terdapat 2 buah komputer yang bertindak sebagai client dan server. Client akan mengirimkan/upload file bernama “file\_diupload.txt” ke server. Server menerima file tersebut dan menyimpan file tersebut dengan nama “hasil\_upload.txt”.

1. Download skeleton source yang diberikan!
2. Lengkapi source code sehingga program bisa berjalan!
3. Jalankan program client dan server pada 2 laptop yang berbeda!
4. Kumpulkan source code yang dibuat serta screenshoot hasil program yang berjalan!
5. **Tugas 3: download file**

Terdapat 2 buah komputer yang bertindak sebagai client dan server. Client akan mendowload file bernama “file\_didownload.txt” yang berada di server. Client kemudian menyimpan file tersebut dengan nama “hasil\_download.txt”

1. Download skeleton source yang diberikan!
2. Lengkapi source code sehingga program bisa berjalan!
3. Jalankan program client dan server pada 2 laptop yang berbeda!
4. Kumpulkan source code yang dibuat serta screenshoot hasil program yang berjalan!
5. Buat layanan echo menggunakan protocol UDP. Layanan echo adalah client mengirimkan pesan kemudian server mengirimkan kembali pesan tersebut dan client akan menampilkan pesan tersebut.

**Happy coding!**