

LAPORAN TUGAS 5

PEMROGRAMAN JARINGAN

**“Implementasi Protokol Chat
dengan Multithread Server”**



Oleh

Nama : Isnaini Nurul KurniaSari

NRP : 05111740000010

Kelas : Pemrograman Jaringan C

Tahun Pelajaran 2019/2020

Departemen Teknik Informatika

Institut Teknologi Sepuluh Nopember

SOAL TUGAS 5

1. Menjalankan program server File chat.py, File chatcli.py serta File server_thread_chat.py. Kemudian, dengan mencoba File chat-cli.py untuk berkomunikasi dengan server.
2. Menambahkan fitur untuk :
 - a. Untuk memperlihatkan pengguna-pengguna yang lain
 - b. Melogout user yang sedang aktif

PENJELASAN

NAMA SERVICE : Chat

DESKRIPSI :

Service ini merupakan sebuah program sebagai sarana untuk melakukan pertukaran pesan atau informasi yang secara mudah dan praktis. Fungsi dari chat adalah untuk mengirim dan menerima pesan yang berupa pesan yang berisi informasi penting.

FORMAT DATA :

Komunikasi dikirimkan ke socket dengan mengirimkan teks alfanumerik dengan format **COMMAND** spasi **PARAMETER1** spasi **PARAMETER2**, dst. Berikut ini adalah format data yang digunakan untuk service **chat** :

1. Auth (untuk login)
 - ⇒ Command yang digunakan ketika client menggunakan fitur ini adalah "**auth**" pada terminal.
 - ⇒

auth [spasi] username [spasi] password
2. Send (Mengirim Pesan)
 - ⇒ Command yang digunakan ketika client menggunakan fitur ini adalah "**send**" pada terminal.

send [spasi] usernametujuan [spasi] message

3. List (melihat list user yang sedang aktif)

⇒ Command yang digunakan ketika client menggunakan fitur ini adalah "**list**" pada terminal.

⇒

list

4. Inbox (melihat pesan masuk)

⇒ Command yang digunakan ketika client menggunakan fitur ini adalah "**inbox**" pada terminal.

⇒

inbox

5. Logout (keluar)

⇒ Command yang digunakan ketika client menggunakan fitur ini adalah "**logout**" pada terminal.

⇒

logout

DAFTAR FITUR

1. Nama Fitur : Auth

a. Deskripsi dan tujuan

Merupakan fitur untuk melakukan login dimana akan dilakukan authentication atau autentikasi dengan melakukan konfirmasi data user yang sebelumnya sudah tersimpan.

Tujuan authentication adalah untuk mengamankan data dengan melakukan konfirmasi data user yang sebelumnya sudah tersimpan di database. Cara kerja dari fitur auth adalah sebagai berikut :

- 1) user yang akan mengakses sistem diminta untuk memasukkan username dan password (asli)
- 2) kemudian sistem akan mencocokkan username dan password dengan yang tersimpan dalam database berdasarkan username yang terdaftar.
- 3) Jika account tersebut ada dalam database, maka user tadi bisa masuk ke dalam sistem. Jika tidak sama, maka user tadi tidak berhak masuk ke dalam sistem.

b. Parameter Input

Parameter input yang digunakan adalah [nama command] : **[username]** <spasi> **[password]**. Pastikan user mengisi username dan password dengan benar.

c. Hasil keluaran, penjelasan status keluaran

- ✓ **Jika user berhasil login, muncul status keluaran berupa :**

```
username {nama user} logged in, token {id token}
```

```
username lineker logged in, token f5d2faba-7e6e-40d7-9678-a1472ef590e0
```

Keterangan :

{nama user} merupakan nama username yang telah berhasil login dan {id token} merupakan id token untuk session yang dibuat saat user berhasil untuk login. Kemudian program akan menyimpan token id yang akan digunakan sebagai session (artinya user telah berhasil login) untuk mengakses program dan menjalankan fitur-fitur pada program.

Penjelasan :

1) Username

Argumen : id pengguna server tertentu (wajib)

Batasan : hanya dapat diberikan otorisasi setelah user berhasil login

Respon yang mungkin terjadi :

- Jika user *salah* memasukkan **username** maka akan muncul :

```
Error, User Tidak Tersedia, Silahkan Coba Lagi
```

2) Password

Argumen : password pengguna server tertentu (wajib)

Batasan : hanya dapat diberikan otorisasi setelah user berhasil login

Respon yang mungkin terjadi :

- Jika user *salah* memasukkan *password*, maka akan muncul :

```
Error, Password yang Anda Masukkan Salah, Silahkan Coba Lagi
```

d. **Contoh message input dan keluaran**

input ➔

```
auth [spasi] username [spasi] password
```

Command username dan password akan dikirim ke server sebagai request.

Kemudian server akan memberikan status dan token id session untuk client.

contoh input : *auth lineker surabaya*

```
C:\Users\asus\AppData\Local\Programs\Python\Python38-64\Scripts\python.exe
Command :auth lineker surabaya
```

output -> *username {namauser} logged in, token {id token}*

contoh output :

```
username lineker logged in, token f5d2faba-7e6e-40d7-9678-a1472ef590e0
```

2. **Nama Fitur : Send**

a. **Deskripsi dan tujuan**

Merupakan fitur untuk mengirimkan suatu pesan berupa informasi penting kepada user yang dituju. Tujuannya untuk menyampaikan atau mengirim pesan.

b. **Parameter Input**

Parameter input yang digunakan adalah [nama command] : **[username]**
<spasi> **[message]**.

c. **Hasil keluaran, penjelasan status keluaran**

- Jika user berhasil mengirim pesan, muncul status keluaran berupa :

```
message send to {usernametujuan}
```

```
diterima dari server b'{"status": "OK", "message": "Message Sent"}\r\n\r\n'
end of string
message sent to messi
```

Keterangan :

Akan muncul status keluaran tersebut, jika user telah berhasil untuk mengirim pesan (message) dimana sebagai contoh diatas *messi* {usernametujuan} merupakan nama username yang menerima pesan tersebut.

Penjelasan :

✓ Untuk Fitur Send

Argumen : pengguna (user) wajib login dan mengisi *usernametujuan* (wajib)

Batasan : hanya dapat mengirim pesan kepada *usernametujuan* jika user berhasil login.

➤ Respon yang mungkin terjadi :

a) Jika user *belum login* maka akan muncul status keluaran :

```
Error, not authorized
```

b) Jika user *salah* memasukkan nama *usernametujuan*, maka akan muncul status keluaran :

```
Error, User Tidak Ditemukan
```

d. Contoh message input dan keluaran

Input -> `send [spasi] usernametujuan [spasi] message`

contoh input : send messi Selamat Pagi Messi

send messi Belajar Bareng Yuk :)

```
Command 38b0d974-d19f-483e-9177-1d3230a5eaae:send messi Selamat Pagi Messi
```

```
Command 455ec1d3-2c93-4086-8f61-70b36afb9288:send messi Belajar Bareng Yuk :)
```

3. Nama Fitur : List

a. Deskripsi dan tujuan

Merupakan fitur untuk melihat daftar atau list user yang sudah mempunyai akun (terdaftar). Tujuannya untuk menampilkan list user yang telah terdaftar yang berfungsi sebagai referensi untuk mengirim pesan di Chat.

b. Parameter Input

Parameter input yang digunakan adalah [nama command] <spasi>
[parameter request token id session]

c. Hasil keluaran, penjelasan status keluaran

- Jika sistem berhasil menampilkan list user, muncul status keluaran berupa :

```
Command 150247ea-da64-4316-98e1-3f4ce36dfc92: list
diterima dari server b'{"status": "OK", "messages": "messi,lineker,henderson,"}\r\n\r\n'
end of string
User yang sedang aktif adalah :
"messi,lineker,henderson,"
```

Keterangan :

Muncul status keluaran tersebut, jika sistem telah berhasil menampilkan list user yang telah terdaftar.

Penjelasan :

✓ Untuk Fitur List

Argumen : pengguna atau server (wajib) untuk login sehingga bisa melihat list user yang telah terdaftar

Batasan : hanya dapat melihat list user yang telah terdaftar

- Respon yang mungkin terjadi :

- a) Jika user *belum login* maka akan muncul status keluaran :

```
Error, not authorized
```

d. Contoh message input dan keluaran

- ✓ Input ->

```
Command 150247ea-da64-4316-98e1-3f4ce36dfc92: list
```

- ✓ Output :

```
Command 150247ea-da64-4316-98e1-3f4ce36dfc92: list
diterima dari server b'{"status": "OK", "messages": "messi,lineker,henderson,"}\r\n\r\n'
end of string
User yang sedang aktif adalah :
"messi,lineker,henderson,"
```

Keterangan :

Sebagai contoh diatas, output atau keluaran untuk menampilkan list user yang sedang aktif chat adalah (*messi*, *lineker*, *henderson*).

4. Nama Fitur : Inbox

a. Deskripsi dan tujuan

Merupakan fitur untuk menerima pesan yang berisi informasi penting dari user pengirim. Tujuannya untuk melihat semua chat yang telah masuk ke penerima.

b. Parameter Input

Parameter input yang digunakan adalah [nama command] <spasi> [parameter request username penerima pesan]

c. Hasil keluaran, penjelasan status keluaran

- Jika user berhasil menerima pesan, muncul status keluaran berupa :

Output json berupa pesan yang telah diterima sebagai berikut :

```
diterima dari server b'{"status": "OK", "messages": [{"henderson": [{"msg_from": "Jordan"
diterima dari server b' Henderson", "msg_to": "Lionel Messi", "msg": " Belajar Bareng "
diterima dari server b'Yuk :) \r\n"}]}, {"lineker": [{"msg_from": "Gary Lineker", "msg_to":
diterima dari server b'": "Lionel Messi", "msg": " Selamat Pagi Messi \r\n"}]}}\r\n\r\n'
end of string
{"henderson": [{"msg_from": "Jordan Henderson", "msg_to": "Lionel Messi", "msg": " Belajar Bareng Yuk :) \r\n"}]}, {"lineker": [{"msg_from": "Gary Lineker", "msg_to":
"Lionel Messi", "msg": " Selamat Pagi Messi \r\n"}]}}
{"henderson": [{"msg_from": "Jordan Henderson", "msg_to": "Lionel Messi", "msg": " Belajar Bareng Yuk :) \r\n"}]}, {"lineker": [{"msg_from": "Gary Lineker", "msg_to":
"Lionel Messi", "msg": " Selamat Pagi Messi \r\n"}]}}
```

Keterangan :

Akan muncul status keluaran tersebut, jika user telah berhasil menerima pesan (message) dimana sebagai contoh diatas *henderson* dan *lineker* {usernamepengirim} merupakan nama username yang telah mengirim pesan kepada *messi*.

Penjelasan :

✓ Untuk Fitur Inbox

Argumen : pengguna atau user (wajib) untuk login sehingga bisa melihat isi pesan

Batasan : hanya dapat menerima pesan dari *usernamepengirim* jika user (sebagai penerima pesan) berhasil login.

➤ Respon yang mungkin terjadi :

a) Jika user *belum login* maka akan muncul status keluaran :

```
Error, not authorized
```

d. Contoh message input dan keluaran

✓ Input ->

```
Command 7d5df05b-14ba-474b-b88d-2638e7029ca2: inbox
```

Keterangan :

Sebagai contoh diatas, *messi* (sebagai penerima pesan) login dan ingin melihat semua pesan yang masuk di chat.

✓ Output :

```
diterima dari server b'{"status": "OK", "messages": [{"henderson": [{"msg_from": "Jordan",  
diterima dari server b' Henderson", "msg_to": "Lionel Messi", "msg": " Belajar Bareng '  
diterima dari server b'Yuk :) \r\n\r\n"}], "lineker": [{"msg_from": "Gary Lineker", "msg_to":  
diterima dari server b'": "Lionel Messi", "msg": " Selamat Pagi Messi \r\n\r\n"}]}\r\n\r\n\r\n'  
end of string  
[{"henderson": [{"msg_from": "Jordan Henderson", "msg_to": "Lionel Messi", "msg": " Belajar Bareng Yuk :) \r\n\r\n"}], "lineker": [{"msg_from": "Gary Lineker", "msg_to":  
"Lionel Messi", "msg": " Selamat Pagi Messi \r\n\r\n"}]}
```

Keterangan :

Sebagai contoh diatas, *messi* memiliki inbox (pesan masuk) yang berisi handerson (yang berisi pesan “Belajar Bareng Yuk :)”) dan lineker (yang berisi pesan “Selamat pagi Messi).

5. Nama Fitur : Logout

a. Deskripsi dan tujuan

Merupakan fitur proses keluar untuk memutus akses ke sistem komputer maupun layanan tertentu. Tujuannya sebagai upaya untuk memutuskan atau mematikan aktifitas yang sedang berlangsung yang sangat berhubungan dengan keamanan yang membuat akun kita bisa terjaga dengan baik.

b. Parameter Input

Parameter input yang digunakan adalah [nama command] <spasi> [**token id client yang sekarang**]

c. Hasil keluaran, penjelasan status keluaran

- Jika user berhasil melakukan logout, muncul status keluaran berupa :

```
Command c5aead50-6729-4abd-abc1-25f05b87f772:logout|
diterima dari server b'{"status": "OK", "messages": "logout telah berhasil, Terimakasih'
diterima dari server b' telah menggunakan Chat :)}"}\r\n\r\n'
end of string
"logout telah berhasil, Terimakasih telah menggunakan Chat :)"
```

Keterangan :

Akan muncul status keluaran tersebut, jika user telah berhasil logout dan akun telah keluar dari sistem.

Penjelasan :

✓ Untuk Fitur Logout

Argumen : -

Batasan : hanya dapat menerima pesan dari *usernamepengirim* jika user (sebagai penerima pesan) berhasil login.

- Respon yang mungkin terjadi :

b) Jika user *belum login* maka akan muncul status keluaran :

```
WARNING! ERROR silahkan login terlebih dahulu
```

d. Contoh message input dan keluaran

✓ Input ->

```
Command 08d68e91-21ab-4c99-bd91-8a140a4dde61:logout
```

✓ Output :

```
Command c5aead50-6729-4abd-abc1-25f05b87f772:logout|
diterima dari server b'{"status": "OK", "messages": "logout telah berhasil, Terimakasih'
diterima dari server b' telah menggunakan Chat :)}"}\r\n\r\n'
end of string
"logout telah berhasil, Terimakasih telah menggunakan Chat :)"
```

Keterangan :

Sebagai contoh diatas, *messi* berhasil untuk *logout* dan akun *messi* berhasil keluar.