

Dokumentasi Tugas 5

Nama: Adnan Erlangga Raharja

NRP: 05111740000100

Kelas: Pemrograman Jaringan – B

Nama Service: Chat

Deskripsi:

Service ini merupakan sebuah aplikasi sederhana berbasis python yang berfungsi layaknya aplikasi chatting. Service ini terdiri dari sebuah hubungan client dengan server. Client dapat mengirimkan request atau parameter ke server. Server dapat menerima request dari client, memprosesnya, kemudian mengirimkan kembali balasan atas request yang telah diterima. Service ini memiliki beberapa fitur layaknya aplikasi chatting pada umumnya seperti login, logout, kirim pesan, baca pesan, dan melihat user yang online.

Format Data:

Komunikasi dari client dikirimkan ke socket pada server dengan mengirimkan teks alfanumerik dengan format seperti dibawah

```
[COMMAND] [PARAMETER1] [PARAMETER2]
```

Secara umum service ini dapat dioperasikan dengan menginputkan perintah dengan format seperti diatas. Untuk lebih jelasnya dapat dilihat parameter input tiap fitur pada penjelasan dibawah.

Daftar Fitur:

1. Login

a. Deskripsi dan tujuan

Setiap pengguna yang ingin menggunakan service ini diharuskan login terlebih dahulu. Pengguna mengirimkan command untuk login beserta dengan username dan password yang valid ke server. Kemudian setelah di cek oleh server bahwa data yang dikirimkan valid, server men-generate sebuah token yang unik dan membuat session bagi pengguna tersebut. Kemudian server mengirimkan status “OK” beserta token yang telah di generate sebelumnya sebagai respon dan tanda bahwa pengguna tersebut telah terautentikasi. Pengguna yang telah terautentikasi dapat menggunakan fitur-fitur lain yang tersedia pada service chat ini seperti logout, kirim pesan, baca pesan, dan melihat

user yang online. Apabila pengguna memasukkan username dan password yang tidak valid maka server akan merespon dan pada client akan di tampilkan pesan error.

Tujuan dari fitur ini adalah untuk mengautentikasi user agar dapat menggunakan service ini secara penuh serta mengakses semua fitur yang terdapat pada service ini.

b. Parameter input

Parameter input yang digunakan untuk login adalah seperti gambar dibawah.

```
auth [USERNAME] [PASSWORD]
```

c. Hasil keluaran

Ada 4 skenario yang mungkin terjadi:

1. Pengguna memasukkan command untuk login dengan benar beserta dengan username dan password yang valid. Maka client akan menampilkan pesan dengan format seperti gambar dibawah.

```
"username {} logged in, token {}".format(username, tokenid)
```

Variabel *tokenid* adalah token.

2. Pengguna memasukkan command untuk login dengan benar beserta dengan username yang valid tetapi password tidak valid. Maka client akan menampilkan pesan dengan format seperti gambar dibawah.

```
"Error, Password Salah"
```

3. Pengguna memasukkan command untuk login dengan benar beserta dengan username dan password yang tidak valid atau dengan kata lain mencoba login tetapi pengguna belum terdaftar. Maka client akan menampilkan pesan dengan format seperti gambar dibawah.

```
"Error, User Tidak Ada"
```

4. Pengguna hanya memasukkan command untuk login dengan benar tanpa parameter lain seperti username dan password. Maka client akan menampilkan pesan dengan format seperti gambar dibawah.

```
"-Maaf, command tidak benar"
```

d. Contoh message input dan keluaran

- Contoh seperti skenario 1:

Input:

```
Command :auth messi surabaya
```

Output:

```
diterima dari server b'{"status": "OK", "tokenid": "c5624216-519d-483c-87a5-3fbb9b0293a6"}\r\n\r\n'
diterima dari server b'6"\r\n\r\n'
end of string
username messi logged in, token c5624216-519d-483c-87a5-3fbb9b0293a6
```

- Contoh seperti skenario 2:

Input:

```
Command :auth messi bandung
```

Output:

```
diterima dari server b'{"status": "ERROR", "message": "Password Salah"}\r\n\r\n'
end of string
Error, Password Salah
```

- Contoh seperti skenario 3:

Input:

```
Command :auth neymar bandung
```

Output:

```
diterima dari server b'{"status": "ERROR", "message": "User Tidak Ada"}\r\n\r\n'
end of string
Error, User Tidak Ada
```

- Contoh seperti skenario 4:

Input:

```
Command :auth
```

Output:

```
-Maaf, command tidak benar
```

2. Mengirim pesan

a. Deskripsi dan tujuan

Setiap pengguna yang telah memiliki token atau sudah terautentikasi dapat mengirimkan pesan kepada pengguna lain melalui service ini. Pengguna yang dituju harus pengguna yang terdaftar pada service ini. Pesan tetap dapat terkirim walaupun pengguna yang dituju tidak sedang aktif atau sedang tidak online. Pesan yang telah dikirimkan tetap akan tersimpan selama server terus berjalan.

Tujuannya adalah agar pengguna yang terautentikasi dapat berkomunikasi dengan pengguna lain dengan cara berkirim pesan melalui service ini.

b. Parameter input

Parameter input yang digunakan untuk mengirimkan pesan kepada pengguna lain adalah seperti gambar dibawah.

```
send [USERNAME TUJUAN] [PESAN]
```

c. Hasil keluaran

Ada 4 skenario yang mungkin terjadi:

1. Pengguna memasukkan command untuk mengirim pesan dengan benar beserta dengan username tujuan yang valid dan sebuah pesan. Maka client akan menampilkan pesan dengan format seperti gambar dibawah.

```
"message sent to {}" . format(username)
```

Variabel *username* adalah username tujuan.

2. Pengguna memasukkan command untuk mengirim pesan dengan benar beserta dengan username tujuan yang tidak valid dan sebuah pesan. Maka client akan menampilkan pesan dengan format seperti gambar dibawah.

```
"Error, User Tidak Ditemukan"
```

3. Pengguna hanya memasukkan command untuk mengirim pesan dengan benar tanpa parameter lain seperti username tujuan dan pesan. Maka client akan menampilkan pesan dengan format seperti gambar dibawah.

```
"-Maaf, command tidak benar"
```

4. Pengguna yang belum terautentikasi mencoba mengirimkan pesan ke pengguna lain tanpa login terlebih dahulu. Maka client akan menampilkan pesan dengan format seperti gambar dibawah.

```
"Error, not authorized"
```

d. Contoh message input dan keluaran

- Contoh seperti skenario 1:

Input:

```
Command 2b8aed86-232e-4764-9a87-3deb6e1170e0:send henderson hey
```

Output:

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

- Contoh seperti skenario 2:

Input:

```
Command 2b8aed86-232e-4764-9a87-3deb6e1170e0:send neymar hey
```

Output:

```
send 2b8aed86-232e-4764-9a87-3deb6e1170e0 neymar hey  
  
diterima dari server b'{"status": "ERROR", "message": "User Tidak Ditemukan"}\r\n\r\n'  
end of string  
Error, User Tidak Ditemukan
```

- Contoh seperti skenario 3:

Input:

```
Command 2b8aed86-232e-4764-9a87-3deb6e1170e0:send
```

Output:

```
-Maaf, command tidak benar
```

- Contoh seperti skenario 4:

Input:

```
Command :send messi hey
```

Output:

```
Error, not authorized
```

3. Melihat pesan

a. Deskripsi dan tujuan

Setiap pengguna yang telah memiliki token atau sudah terautentikasi dapat melihat pesan yang telah dikirimkan oleh pengguna lain kepada pengguna tersebut melalui service ini. Pesan yang diterima terdiri dari detail pengirim seperti username dan nama pengirim, lalu nama penerima, dan juga isi pesannya. Pesan tetap dapat diterima walaupun pengguna tidak sedang aktif atau sedang tidak online. Pesan yang telah diterima oleh pengguna tetap akan tersimpan selama server terus berjalan.

Tujuannya adalah agar pengguna yang terautentikasi dapat berkomunikasi dengan pengguna lain dengan cara membaca pesan yang dikirimkan oleh pengguna lain melalui service ini.

b. Parameter input

Parameter input yang digunakan untuk melihat pesan yang telah diterima adalah seperti gambar dibawah.

inbox

c. Hasil keluaran

Ada 2 skenario yang mungkin terjadi:

1. Pengguna menginputkan command untuk melihat pesan yang telah diterima dengan benar. Maka client akan menampilkan pesan dengan format seperti gambar dibawah.

```
{{}}: [{ 'msg_from': {}, 'msg_to': {}, 'msg': {} }]]. format(username_from, s_fr['nama'], s_to['nama'], message)
```

Keterangan:

- username_from = username pengirim pesan.
 - s_fr['nama'] = nama pengirim pesan.
 - s_to['nama'] = nama penerima pesan.
 - message = isi pesan.
2. Pengguna yang belum terautentikasi mencoba melihat pesan tanpa login terlebih dahulu. Maka client akan menampilkan pesan dengan format seperti gambar dibawah.

"Error, not authorized"

d. Contoh message input dan keluaran

- Contoh seperti skenario 1:

Input:

Command ef562d18-3620-4f39-a05f-9399f5a73041:inbox

Output:

```
diterima dari server b'{"status": "OK", "messages": {"messi": [{"msg_from": "Lionel Mes'
diterima dari server b'si", "msg_to": "Jordan Henderson", "msg": " hey \\r\\n"}]}\r\n\r\n'
end of string
{"messi": [{"msg_from": "Lionel Messi", "msg_to": "Jordan Henderson", "msg": " hey \r\n"}]}
```

- Contoh seperti skenario 2:

Input:

Command :inbox

Output:

Error, not authorized

4. Melihat pengguna

a. Deskripsi dan tujuan

Setiap pengguna yang telah memiliki token atau sudah terautentikasi dapat melihat daftar pengguna yang sedang aktif melalui service ini. Daftar yang dapat dilihat adalah pengguna yang lain yang telah terdaftar dan telah terautentikasi/sudah login. Jadi apabila tidak ada pengguna lain yang aktif maka ketika menggunakan fitur ini hanya pengguna saat ini saja yang ada di dalam daftar pengguna.

Tujuannya adalah agar pengguna yang terautentikasi dapat mengetahui siapa saja pengguna lain yang sedang aktif saat ini melalui service ini.

b. Parameter input

Parameter input yang digunakan untuk melihat daftar pengguna aktif adalah seperti gambar dibawah.

```
users
```

c. Hasil keluaran

Ada 2 skenario yang mungkin terjadi:

1. Pengguna menginputkan command untuk melihat daftar pengguna aktif saat ini dengan benar. Maka client akan menampilkan pesan dengan format seperti gambar dibawah.

```
"Active Users:\n[\"{} [{}]]\" . format(username, name)
```

2. Pengguna yang belum terautentikasi mencoba melihat daftar pengguna aktif saat ini tanpa login terlebih dahulu. Maka client akan menampilkan pesan dengan format seperti gambar dibawah.

```
"Error, not authorized"
```

d. Contoh message input dan keluaran

- Contoh seperti skenario 1:

Input:

```
Command 65b56154-700e-48be-96d3-b9de2ffa2496:users
```

Output:

```
diterima dari server b'{"status": "OK", "messages": ["messi [Lionel Messi]"]}\r\n\r\n'
end of string
Active Users:
["messi [Lionel Messi]"]
```

- Contoh seperti skenario 2:

Input:

```
Command :users
```

Output:

```
Error, not authorized
```

5. Logout

a. Deskripsi dan tujuan

Setiap pengguna yang telah memiliki token atau sudah terautentikasi dapat logout/keluar dari service ini. Saat seorang pengguna logout maka token dan session untuk user tersebut di hapus. Pengguna yang sudah logout tidak dapat menggunakan fitur-fitur yang ada pada service ini kecuali melakukan login kembali.

Tujuannya adalah agar pengguna yang terautentikasi dapat keluar dari service ini.

b. Parameter input

Parameter input yang digunakan untuk logout dari service ini adalah seperti gambar dibawah.

```
logout
```

c. Hasil keluaran

Ada 2 skenario yang mungkin terjadi:

1. Pengguna memasukkan command untuk logout dengan benar. Maka client akan menampilkan pesan dengan format seperti gambar dibawah.

```
"LOGGING OUT: {} [{}]\nSuccessfully logged out!" . format(username, name)
```

2. Pengguna yang belum terautentikasi mencoba untuk logout tanpa login terlebih dahulu. Maka client akan menampilkan pesan dengan format seperti gambar dibawah.


```
"Error, not authorized"
```

d. Contoh message input dan keluaran

- Contoh seperti skenario 1:

Input:

```
Command 65b56154-700e-48be-96d3-b9de2ffa2496:logout
```

Output:

```
diterima dari server b'{"status": "OK", "messages": "LOGGING OUT: messi [Lionel Messi]"'  
diterima dari server b', "tokenid": ""}\r\n\r\n'  
end of string  
"LOGGING OUT: messi [Lionel Messi]"  
Successfully logged out!
```

- Contoh seperti skenario 2:

Input:

```
Command :logout
```

Output:

```
Error, not authorized
```