

TUGAS 5
DOKUMENTASI
LAYANAN *CHAT MULTITHREADED SERVER*



PEMROGRAMAN JARINGAN B

05111740007005 Ihdiannaja

Dosen:

Royyana Muslim Ijtihadie, S.Kom., Ph.D

Departemen Teknik Informatika

Fakultas Teknologi Elektro dan Informatika Cerdas

Institut Teknologi Sepuluh Nopember Surabaya

NAMA LAYANAN

Layanan Chat dengan Multithreaded Server

DESKRIPSI

Layanan Chat Multithreaded Server merupakan layanan untuk bertukar pesan antar client dengan menggunakan socket dengan multithreaded server. Cara kerja layanan ini dimulai dengan server membuka socket dan menunggu adanya koneksi dari client. Setelah itu, client akan mengirimkan request ke server dan akan direspon kembali oleh server menuju client sesuai dengan request yang telah dikirimkan.

Penggunaan multithreaded server dalam layanan ini memungkinkan satu server dapat menangani lebih dari satu client secara bersamaan dalam satu jaringan yang sama.

FORMAT DATA

Format data dari tiap fitur dalam layanan ini adalah sebagai berikut:

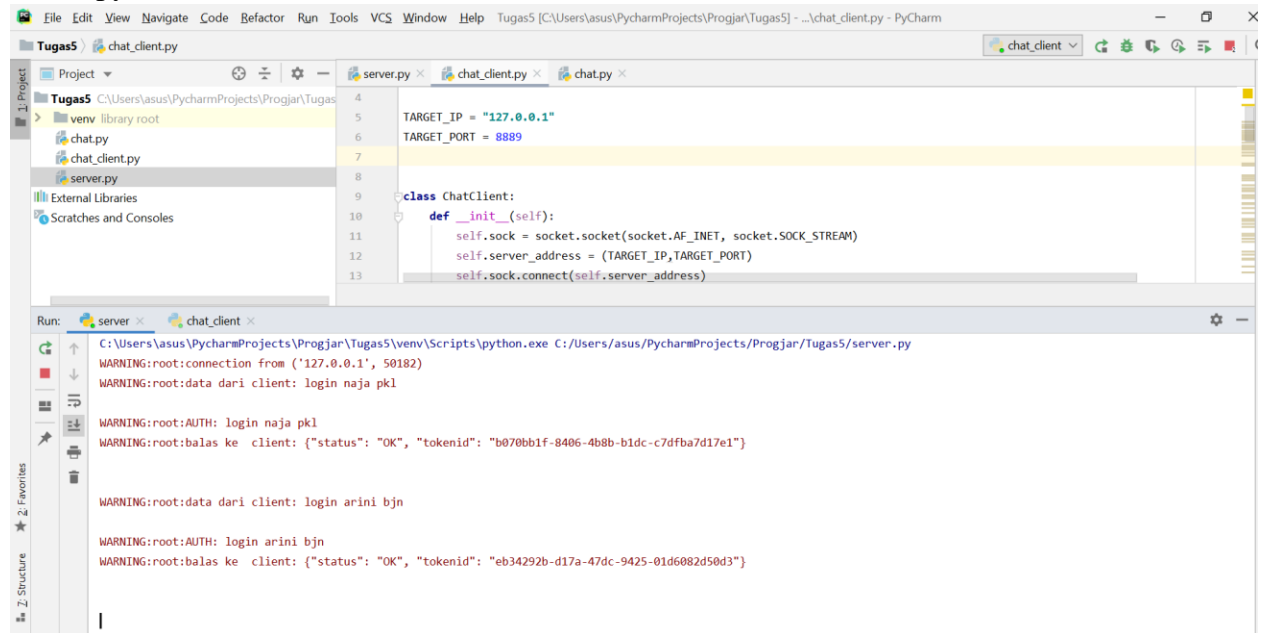
1. **Fitur Login**
Input command: **login** (*username*) (*password*)
2. **Fitur chat user lain**
Input command: **chat** (*user_tujuan*) (*pesan*)
3. **Fitur kotak masuk**
Input command: **kotakmasuk**
4. **Fitur melihat user aktif**
Input command: **user_on**
5. **Fitur Logout**
Input command: **logout**

DETAIL FITUR

1. **LOGIN**
FUNGSI : Melakukan autentikasi user agar dapat mengakses fitur-fitur yang lain.
PARAMETER: (*username*) (*password*)
SYARAT : Username dan password yang dimasukkan sudah dalam daftar users atau diinputkan secara manual pada program.
OUTPUT : a. Jika respon berhasil (OK), outputnya adalah username [user] loggen in, token: [tokenid]
b. Jika respon gagal (ERROR), outputnya adalah message [message_error]

DOKUMENTASI LOGIN:

server.py



The screenshot shows the PyCharm IDE with the `server.py` file open. The code defines a `ChatClient` class with an `__init__` method that sets `TARGET_IP` to `"127.0.0.1"` and `TARGET_PORT` to `8889`. The class connects to the target IP and port. The Run window shows the output of the server script, which includes warnings for connection, data received from the client, authentication, and response.

```
4 TARGET_IP = "127.0.0.1"
5 TARGET_PORT = 8889
6
7
8
9 class ChatClient:
10     def __init__(self):
11         self.sock = socket.socket(socket.AF_INET, socket.SOCK_STREAM)
12         self.server_address = (TARGET_IP, TARGET_PORT)
13         self.sock.connect(self.server_address)
```

Run: server.py

```
C:\Users\asus\PycharmProjects\Progjar\Tugas5\Scripts\python.exe C:/Users/asus/PycharmProjects/Progjar/Tugas5/server.py
WARNING:root:connection from ('127.0.0.1', 50182)
WARNING:root:data dari client: login naja pkl

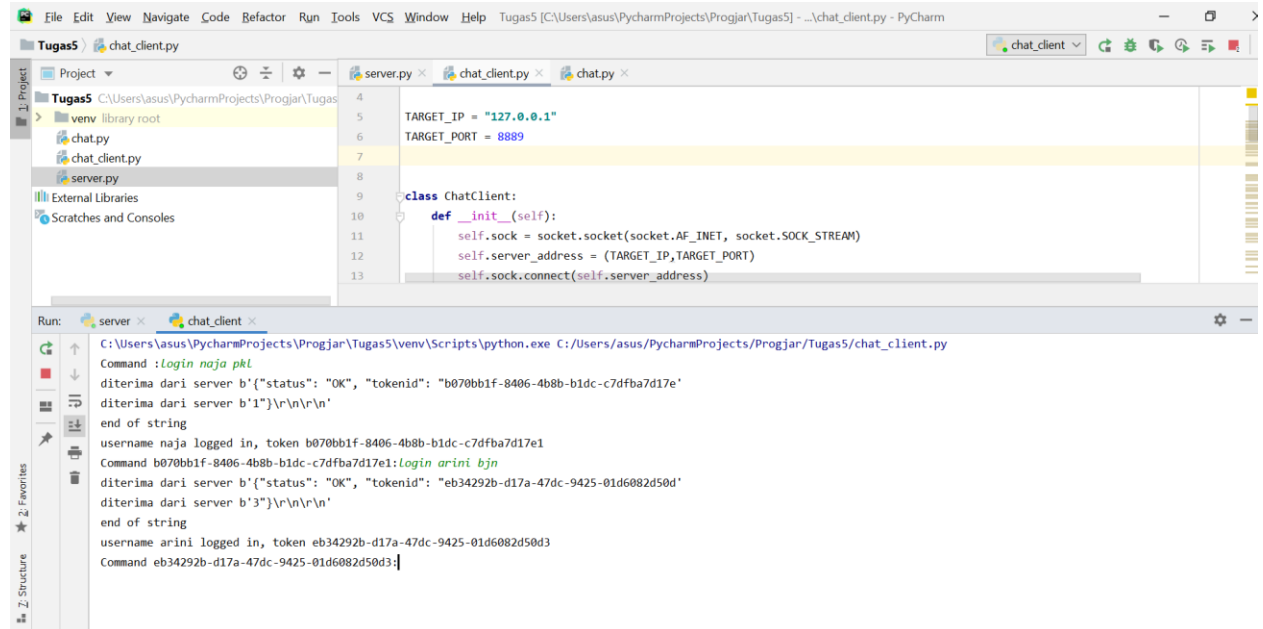
WARNING:root:AUTH: login naja pkl
WARNING:root:balas ke client: {"status": "OK", "tokenId": "b070bb1f-8406-4b8b-b1dc-c7dfba7d17e1"}

WARNING:root:data dari client: login arini bjn

WARNING:root:AUTH: login arini bjn
WARNING:root:balas ke client: {"status": "OK", "tokenId": "eb34292b-d17a-47dc-9425-01d6082d50d3"}

|
```

chat_client.py



The screenshot shows the PyCharm IDE with the `chat_client.py` file open. The code defines a `ChatClient` class with an `__init__` method that sets `TARGET_IP` to `"127.0.0.1"` and `TARGET_PORT` to `8889`. The class connects to the target IP and port. The Run window shows the output of the chat client script, which includes commands like `login naja pkl` and `login arini bjn`, and the server's response.

```
4 TARGET_IP = "127.0.0.1"
5 TARGET_PORT = 8889
6
7
8
9 class ChatClient:
10     def __init__(self):
11         self.sock = socket.socket(socket.AF_INET, socket.SOCK_STREAM)
12         self.server_address = (TARGET_IP, TARGET_PORT)
13         self.sock.connect(self.server_address)
```

Run: chat_client.py

```
C:\Users\asus\PycharmProjects\Progjar\Tugas5\Scripts\python.exe C:/Users/asus/PycharmProjects/Progjar/Tugas5/chat_client.py
Command: login naja pkl
diterima dari server b'{"status": "OK", "tokenId": "b070bb1f-8406-4b8b-b1dc-c7dfba7d17e1"}'
end of string
username naja logged in, token b070bb1f-8406-4b8b-b1dc-c7dfba7d17e1
Command b070bb1f-8406-4b8b-b1dc-c7dfba7d17e1: login arini bjn
diterima dari server b'{"status": "OK", "tokenId": "eb34292b-d17a-47dc-9425-01d6082d50d3"}'
end of string
username arini logged in, token eb34292b-d17a-47dc-9425-01d6082d50d3
Command eb34292b-d17a-47dc-9425-01d6082d50d3: |
```

2. CHAT

FUNGSI : Mengirim pesan antar client.

PARAMETER: (*user_tujuan*) (*pesan*)

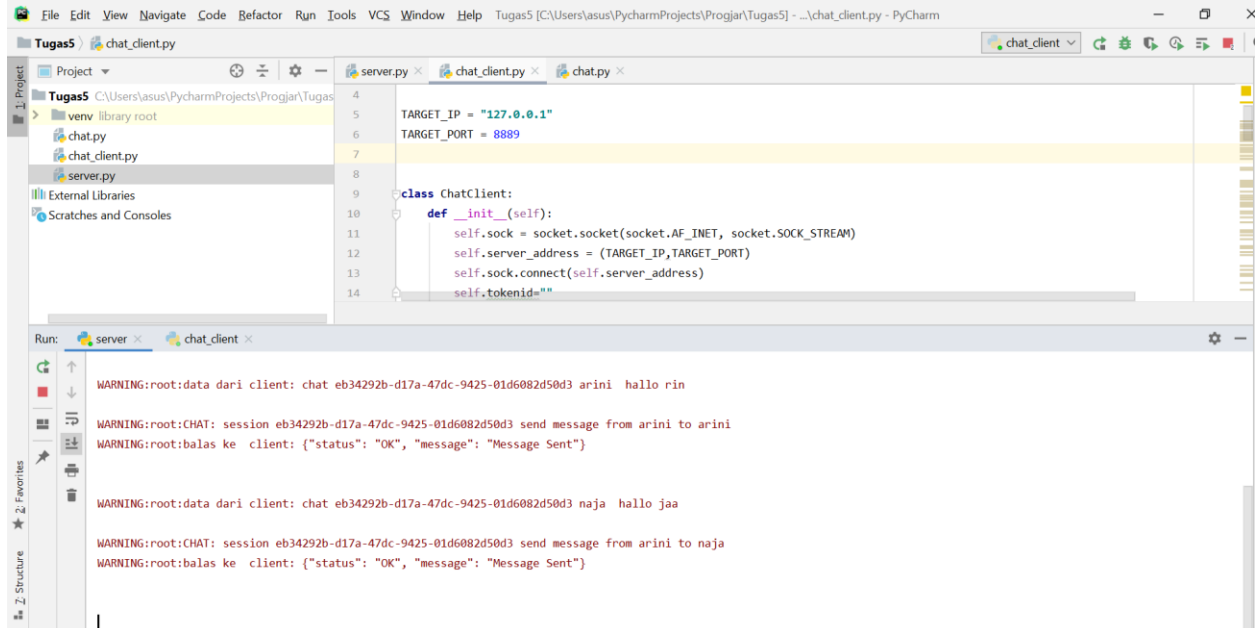
SYARAT : Sender harus melakukan login terlebih dahulu

OUTPUT : a. Jika respon berhasil (OK), outputnya adalah send to [username to]

b. Jika respon gagal (ERROR), outputnya adalah message [message_error]

DOKUMENTASI CHAT

server.py



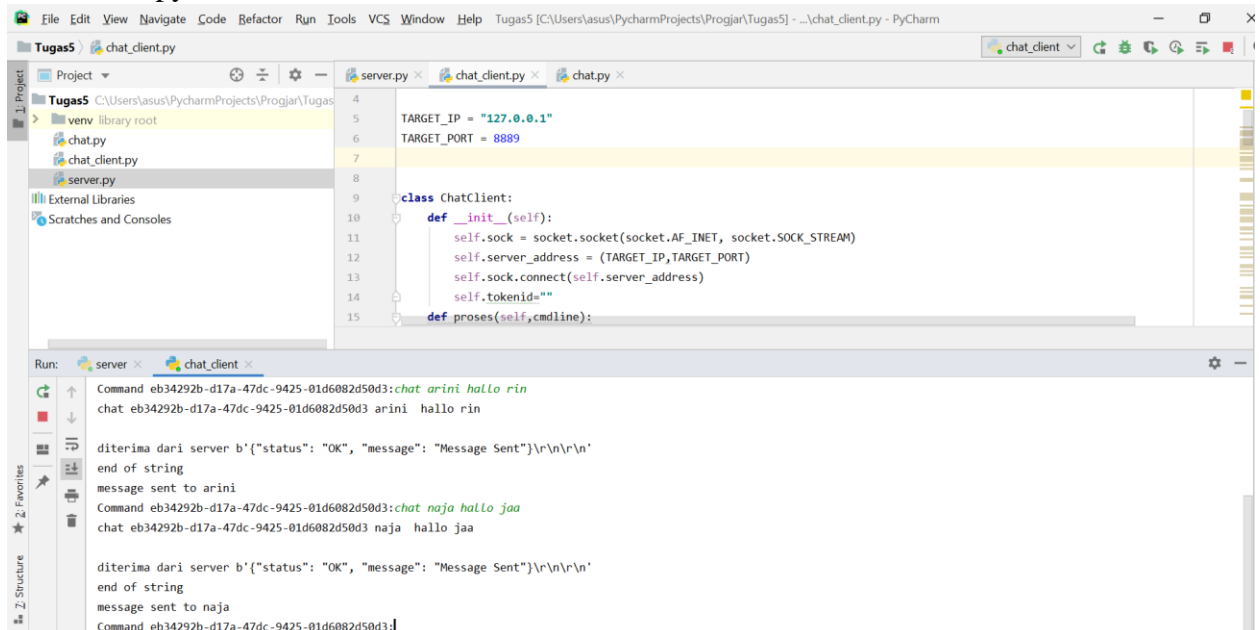
The screenshot shows the PyCharm IDE with the `server.py` file open. The code defines a `ChatClient` class with an `__init__` method that sets `TARGET_IP` to `"127.0.0.1"` and `TARGET_PORT` to `8889`. The `__init__` method also creates a socket, connects it to the target IP and port, and sets `self.tokenid` to an empty string. The `Run` tab shows the output of the server, which includes warnings about data from the client and messages sent from the client to the server.

```
4 TARGET_IP = "127.0.0.1"
5 TARGET_PORT = 8889
6
7
8
9 class ChatClient:
10     def __init__(self):
11         self.sock = socket.socket(socket.AF_INET, socket.SOCK_STREAM)
12         self.server_address = (TARGET_IP, TARGET_PORT)
13         self.sock.connect(self.server_address)
14         self.tokenid = ""
```

Run: server × chat_client ×

```
WARNING:root:data dari client: chat eb34292b-d17a-47dc-9425-01d6082d50d3 arini halo rin
WARNING:root:CHAT: session eb34292b-d17a-47dc-9425-01d6082d50d3 send message from arini to arini
WARNING:root:balas ke client: {"status": "OK", "message": "Message Sent"}
WARNING:root:data dari client: chat eb34292b-d17a-47dc-9425-01d6082d50d3 naja halo jaa
WARNING:root:CHAT: session eb34292b-d17a-47dc-9425-01d6082d50d3 send message from arini to naja
WARNING:root:balas ke client: {"status": "OK", "message": "Message Sent"}
```

chat_client.py



The screenshot shows the PyCharm IDE with the `chat_client.py` file open. The code defines a `ChatClient` class with an `__init__` method that sets `TARGET_IP` to `"127.0.0.1"` and `TARGET_PORT` to `8889`. The `__init__` method also creates a socket, connects it to the target IP and port, and sets `self.tokenid` to an empty string. The `proses` method is also defined. The `Run` tab shows the output of the client, which includes commands sent to the server and the server's response.

```
4 TARGET_IP = "127.0.0.1"
5 TARGET_PORT = 8889
6
7
8
9 class ChatClient:
10     def __init__(self):
11         self.sock = socket.socket(socket.AF_INET, socket.SOCK_STREAM)
12         self.server_address = (TARGET_IP, TARGET_PORT)
13         self.sock.connect(self.server_address)
14         self.tokenid = ""
15     def proses(self, cmdline):
```

Run: server × chat_client ×

```
Command eb34292b-d17a-47dc-9425-01d6082d50d3: chat arini halo rin
chat eb34292b-d17a-47dc-9425-01d6082d50d3 arini halo rin
diterima dari server b'{"status": "OK", "message": "Message Sent"}\n\n\n'
end of string
message sent to arini
Command eb34292b-d17a-47dc-9425-01d6082d50d3: chat naja halo jaa
chat eb34292b-d17a-47dc-9425-01d6082d50d3 naja halo jaa
diterima dari server b'{"status": "OK", "message": "Message Sent"}\n\n\n'
end of string
message sent to naja
Command eb34292b-d17a-47dc-9425-01d6082d50d3:
```

3. KOTAKMASUK

FUNGSI : Melihat pesan masuk yang dikirimkan antar user.

PARAMETER: -

SYARAT : User harus melakukan login terlebih dahulu

OUTPUT : a. Jika respon berhasil (OK), outputnya adalah message

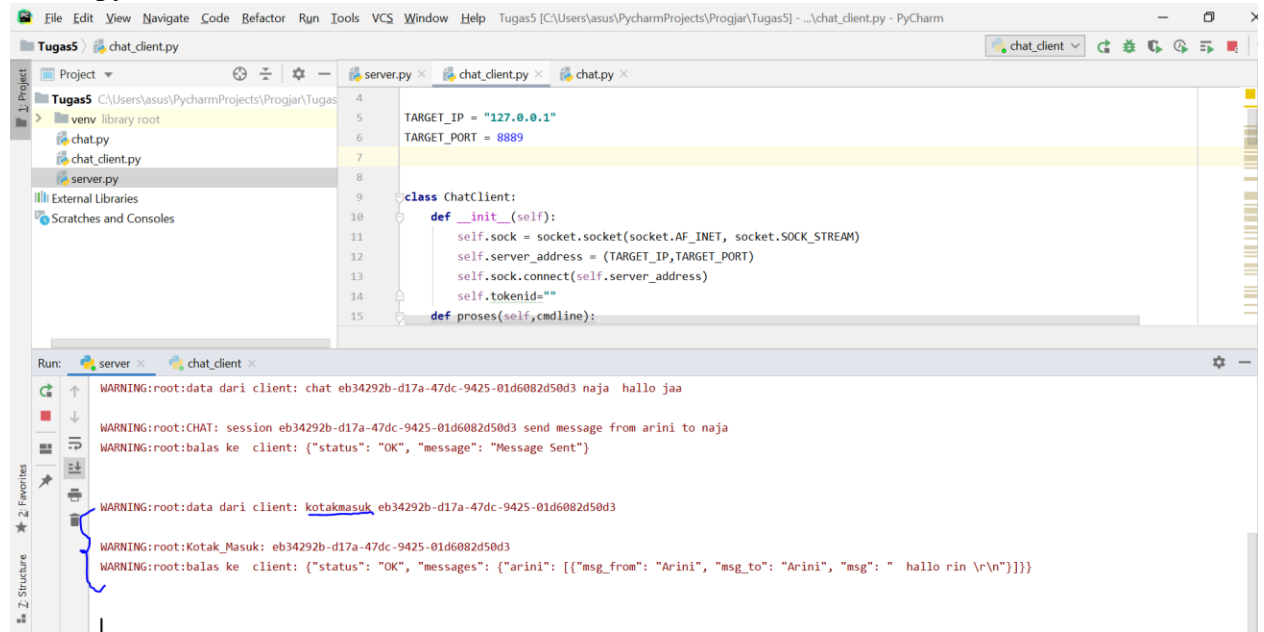
[usernamefrom] [username to] [message]

b. Jika respon gagal (ERROR), outputnya adalah message

[message_error]

DOKUMENTASI KOTAKMASUK

server.py

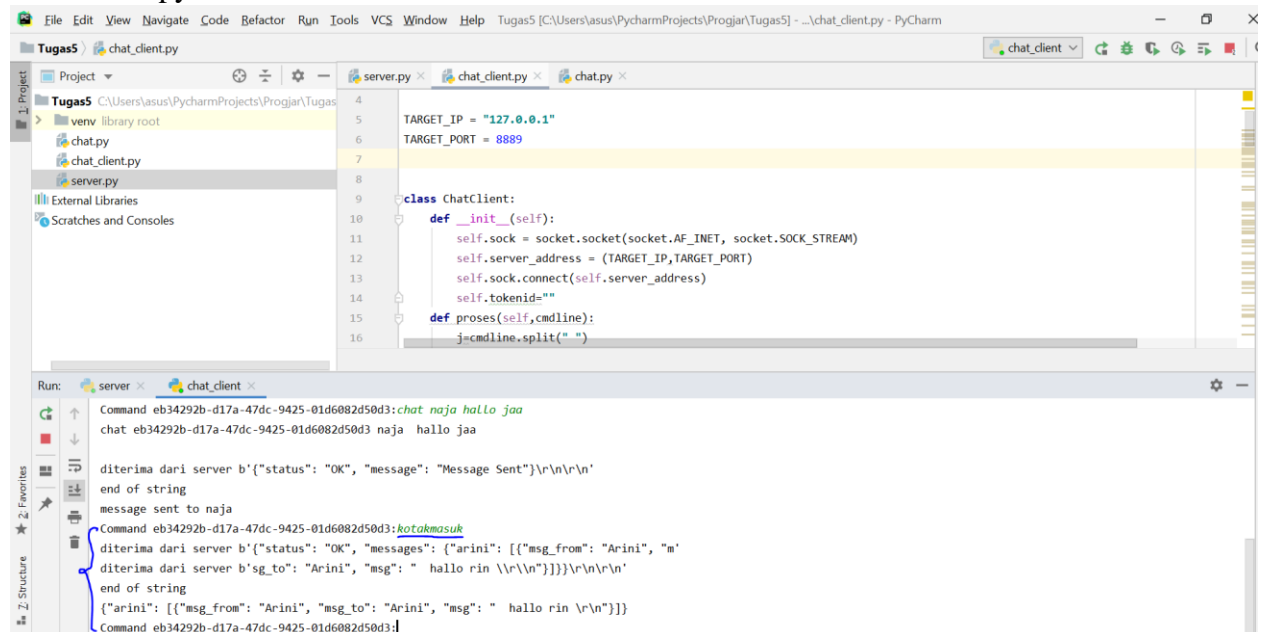


The screenshot shows the PyCharm IDE with the `server.py` file open. The code defines a `ChatClient` class with an `__init__` method that sets `self.sock`, `self.server_address`, and `self.tokenid`. It also has a `proses` method. The console output shows the following messages:

```
WARNING:root:data dari client: chat eb34292b-d17a-47dc-9425-01d6082d50d3 naja hallo jaa
WARNING:root:CHAT: session eb34292b-d17a-47dc-9425-01d6082d50d3 send message from arini to naja
WARNING:root:balas ke client: {"status": "OK", "message": "Message Sent"}

WARNING:root:data dari client: kotakmasuk eb34292b-d17a-47dc-9425-01d6082d50d3
WARNING:root:Kotak_Masuk: eb34292b-d17a-47dc-9425-01d6082d50d3
WARNING:root:balas ke client: {"status": "OK", "messages": {"arini": [{"msg_from": "Arini", "msg_to": "Arini", "msg": "hallo rin \r\n"}]}}
```

chat_client.py



The screenshot shows the PyCharm IDE with the `chat_client.py` file open. The code defines a `ChatClient` class with an `__init__` method that sets `self.sock`, `self.server_address`, and `self.tokenid`. It also has a `proses` method that splits the command line. The console output shows the following messages:

```
Command eb34292b-d17a-47dc-9425-01d6082d50d3: chat naja hallo jaa
chat eb34292b-d17a-47dc-9425-01d6082d50d3 naja hallo jaa

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

Command eb34292b-d17a-47dc-9425-01d6082d50d3: kotakmasuk
diterima dari server b'{"status": "OK", "messages": {"arini": [{"msg_from": "Arini", "m
diterima dari server b'sg_to": "Arini", "msg": "hallo rin \r\n"}]}\r\n\r\n'
end of string
{"arini": [{"msg_from": "Arini", "msg_to": "Arini", "msg": "hallo rin \r\n"}]}
```

4. USER_ON

FUNGSI : Melihat user yang sedang aktif/login

PARAMETER: -

SYARAT : User harus melakukan login terlebih dahulu

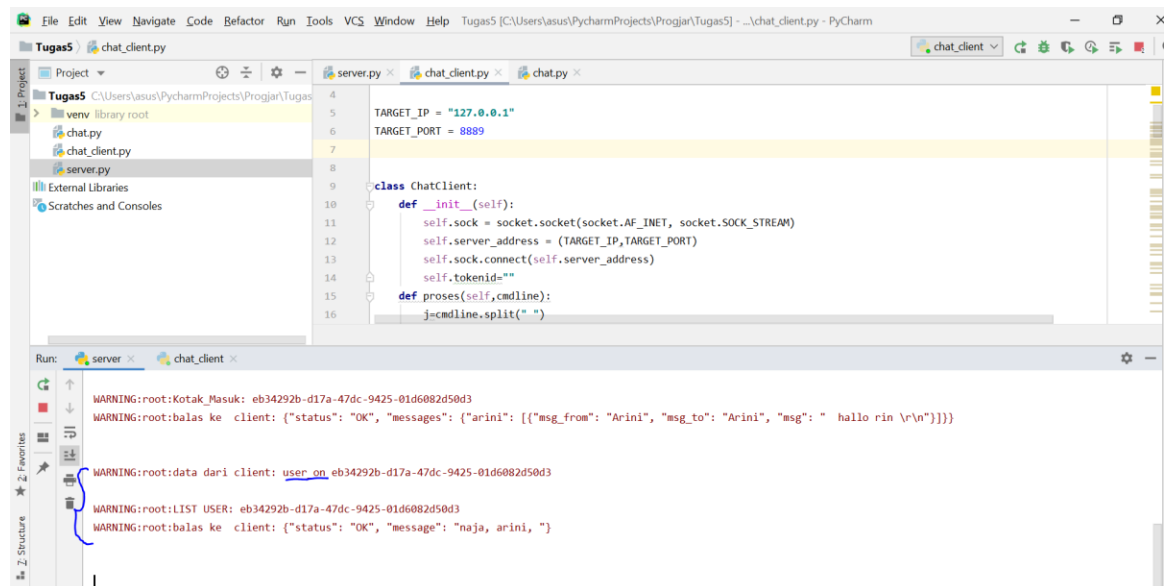
OUTPUT : a. Jika respon berhasil (OK), outputnya adalah message user aktif:

[username_user_aktif]

b. Jika respon gagal (ERROR), outputnya adalah message [message_error]

DOKUMENTASI USER_ON

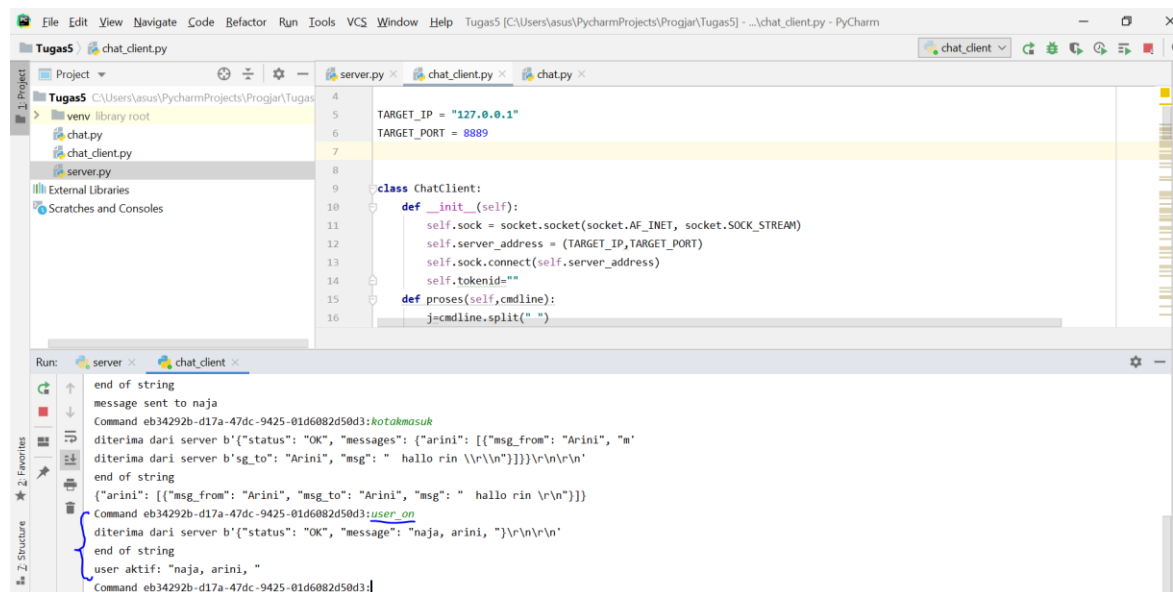
server.py



The screenshot shows the PyCharm IDE with the `server.py` file open. The code defines a `ChatClient` class with methods `__init__` and `proses`. The `__init__` method sets up a socket connection to `127.0.0.1` on port `8889`. The `proses` method splits the input command. The Run console shows the following output:

```
WARNING:root:Kotak_Masuk: eb34292b-d17a-47dc-9425-01d6082d50d3
WARNING:root:balas ke client: {"status": "OK", "messages": [{"msg_from": "Arini", "msg_to": "Arini", "msg": "hallo rin \r\n\r\n"}]}
WARNING:root:data dari client: user_on eb34292b-d17a-47dc-9425-01d6082d50d3
WARNING:root:LIST USER: eb34292b-d17a-47dc-9425-01d6082d50d3
WARNING:root:balas ke client: {"status": "OK", "message": "naja, arini, "}
```

chat_client.py



The screenshot shows the PyCharm IDE with the `chat_client.py` file open. The code defines a `ChatClient` class with methods `__init__` and `proses`. The `__init__` method sets up a socket connection to `127.0.0.1` on port `8889`. The `proses` method splits the input command. The Run console shows the following output:

```
end of string
message sent to naja
Command eb34292b-d17a-47dc-9425-01d6082d50d3:kotakmasuk
diterima dari server b'{"status": "OK", "messages": [{"msg_from": "Arini", "m
diterima dari server b'sg_to": "Arini", "msg": "hallo rin \r\n\r\n"}]\r\n\r\n'
end of string
{"arini": [{"msg_from": "Arini", "msg_to": "Arini", "msg": "hallo rin \r\n\r\n"}]}
Command eb34292b-d17a-47dc-9425-01d6082d50d3:user_on
diterima dari server b'{"status": "OK", "message": "naja, arini, "}\r\n\r\n'
end of string
user aktif: "naja, arini, "
Command eb34292b-d17a-47dc-9425-01d6082d50d3:
```

5. LOGOUT

FUNGSI : Mengakhiri session pada user

PARAMETER: -

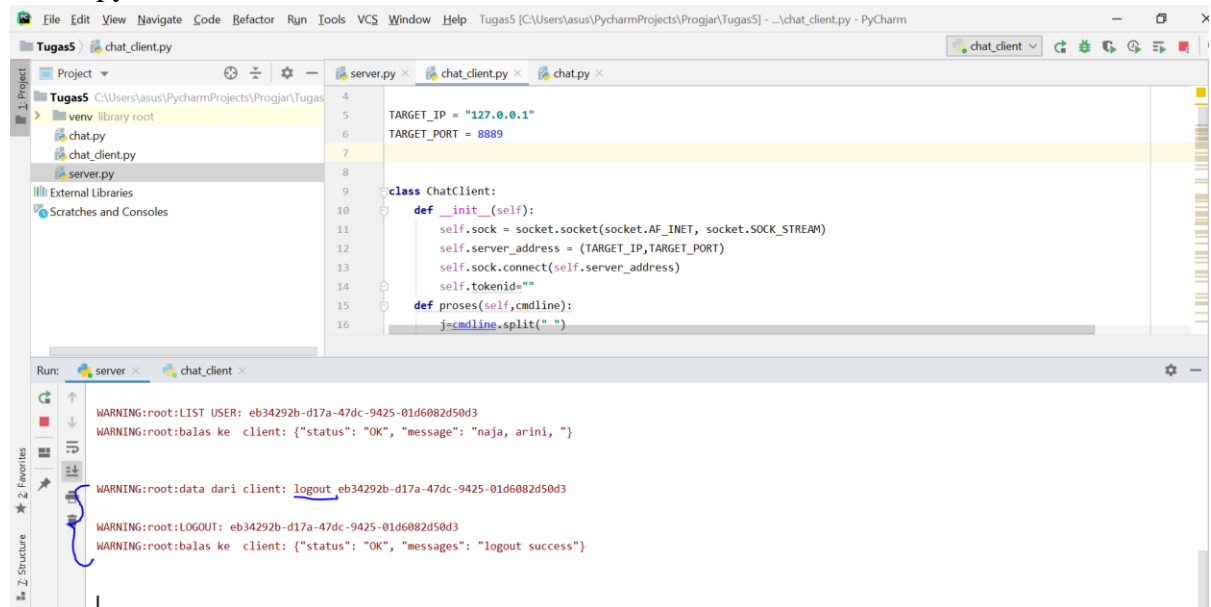
SYARAT : User harus melakukan login terlebih dahulu

OUTPUT : a. Jika respon berhasil (OK), outputnya adalah message logout success

b. Jika respon gagal (ERROR), outputnya adalah message [message_error]

DOKUMENTASI LOGOUT

server.py



```
File Edit View Navigate Code Refactor Run Tools VCS Window Help Tugas5 [C:\Users\asus\PycharmProjects\Progar\Tugas5] - ...chat_client.py - PyCharm
Tugas5 chat_client.py
Project
Tugas5 C:\Users\asus\PycharmProjects\Progar\Tugas5
venv library root
chat.py
chat_client.py
server.py
External Libraries
Scratches and Consoles
server.py
4
5
6
7
8
9
10
11
12
13
14
15
16
TARGET_IP = "127.0.0.1"
TARGET_PORT = 8889

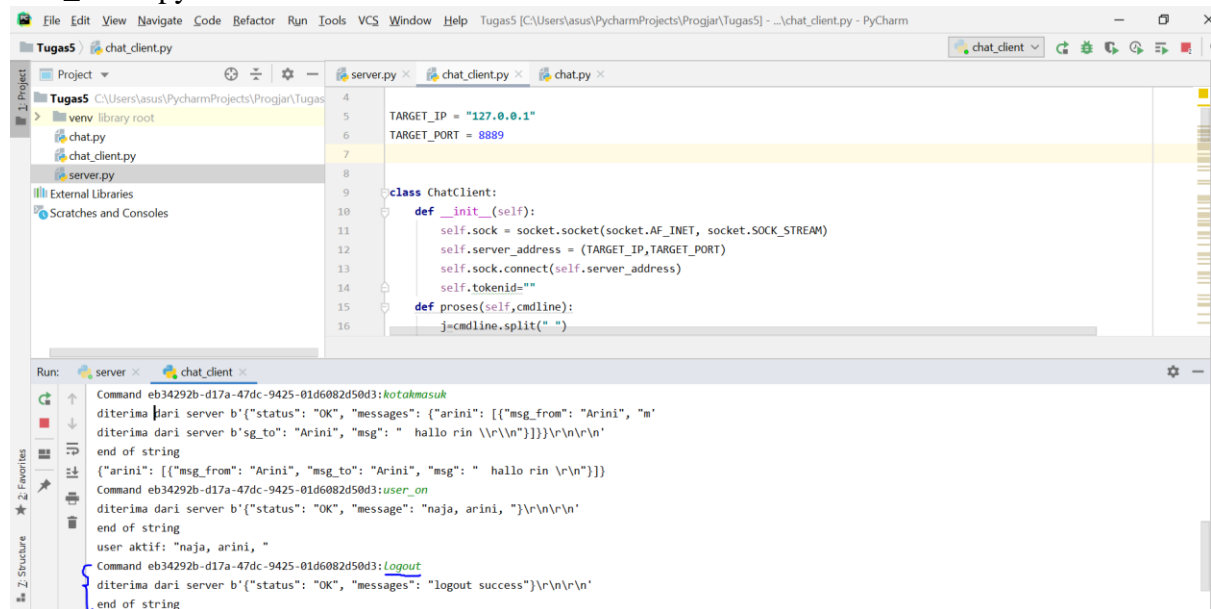
class ChatClient:
    def __init__(self):
        self.sock = socket.socket(socket.AF_INET, socket.SOCK_STREAM)
        self.server_address = (TARGET_IP, TARGET_PORT)
        self.sock.connect(self.server_address)
        self.tokenid=""
    def proses(self,cmdline):
        j=cmdline.split(" ")

Run: server chat_client
WARNING:root:LIST USER: eb34292b-d17a-47dc-9425-01d6082d50d3
WARNING:root:balas ke client: {"status": "OK", "message": "naja, arini, "}

WARNING:root:data dari client: logout eb34292b-d17a-47dc-9425-01d6082d50d3

WARNING:root:LOGOUT: eb34292b-d17a-47dc-9425-01d6082d50d3
WARNING:root:balas ke client: {"status": "OK", "messages": "logout success"}
```

chat_client.py



```
File Edit View Navigate Code Refactor Run Tools VCS Window Help Tugas5 [C:\Users\asus\PycharmProjects\Progar\Tugas5] - ...chat_client.py - PyCharm
Tugas5 chat_client.py
Project
Tugas5 C:\Users\asus\PycharmProjects\Progar\Tugas5
venv library root
chat.py
chat_client.py
server.py
External Libraries
Scratches and Consoles
chat_client.py
4
5
6
7
8
9
10
11
12
13
14
15
16
TARGET_IP = "127.0.0.1"
TARGET_PORT = 8889

class ChatClient:
    def __init__(self):
        self.sock = socket.socket(socket.AF_INET, socket.SOCK_STREAM)
        self.server_address = (TARGET_IP, TARGET_PORT)
        self.sock.connect(self.server_address)
        self.tokenid=""
    def proses(self,cmdline):
        j=cmdline.split(" ")

Run: server chat_client
Command eb34292b-d17a-47dc-9425-01d6082d50d3: kotakmasuk
diterima dari server b'{"status": "OK", "messages": {"arini": [{"msg_from": "Arini", "m'
diterima dari server b'sg_to": "Arini", "msg": "hallo rin \r\n\r\n\r\n\r\n\r\n'
end of string
{"arini": [{"msg_from": "Arini", "msg_to": "Arini", "msg": "hallo rin \r\n\r\n\r\n\r\n\r\n'
Command eb34292b-d17a-47dc-9425-01d6082d50d3: user_on
diterima dari server b'{"status": "OK", "message": "naja, arini, "}\r\n\r\n\r\n\r\n'
end of string
user aktif: "naja, arini, "
Command eb34292b-d17a-47dc-9425-01d6082d50d3: logout
diterima dari server b'{"status": "OK", "messages": "logout success"}\r\n\r\n\r\n\r\n'
end of string
```

