

DOKUMENTASI PEMROGRAMAN JARINGAN

“Chat Client-Server”



OLEH

ANDRE SETIAWAN

5113100013

Institut Teknologi Sepuluh Nopember (ITS)

Surabaya

2015

PROTOKOL PENGGUNAAN PROGRAM CLIENT SERVER CHAT

Terdapat empat terminal yang akan diaktifkan. Terminal pertama berperan sebagai server. Sementara terminal kedua, ketiga dan keempat akan berperan masing-masing sebagai client A, client B, dan client C.

Server telah melakukan deklarasi socket, dan menetapkan HOST dan PORT yang akan digunakan sebagai alamat socket.

Dalam hal ini, HOST dan PORT server telah ditetapkan secara langsung, dimana program akan berjalan pada:

HOST = localhost, PORT = 10000.

Kemudian, server akan melakukan binding dan melakukan listen terhadap koneksi yang masuk.

```
socket.bind('localhost',10000).
```

```
socket.listen(20).
```

Pada client, socket client akan mencoba menghubungkan diri ke socket server dengan HOST dan PORT yang telah ditetapkan server.

```
sock.connect('localhost',10000).
```

REGISTER DAN LOGIN

Client A harus memasukkan username dan password untuk menggunakan program chat client server. Jika client belum melakukan login, maka client tidak dapat mengirim pesan kepada siapapun dan server akan mengirim pesan kepada client yang bersangkutan bahwa client harus melakukan login terlebih dahulu.

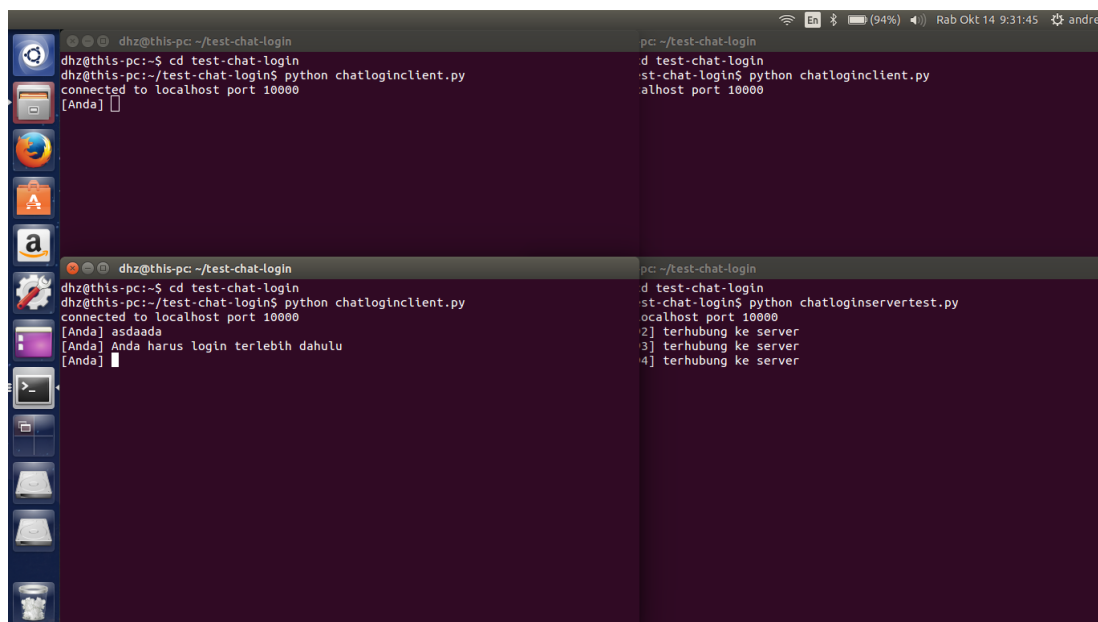
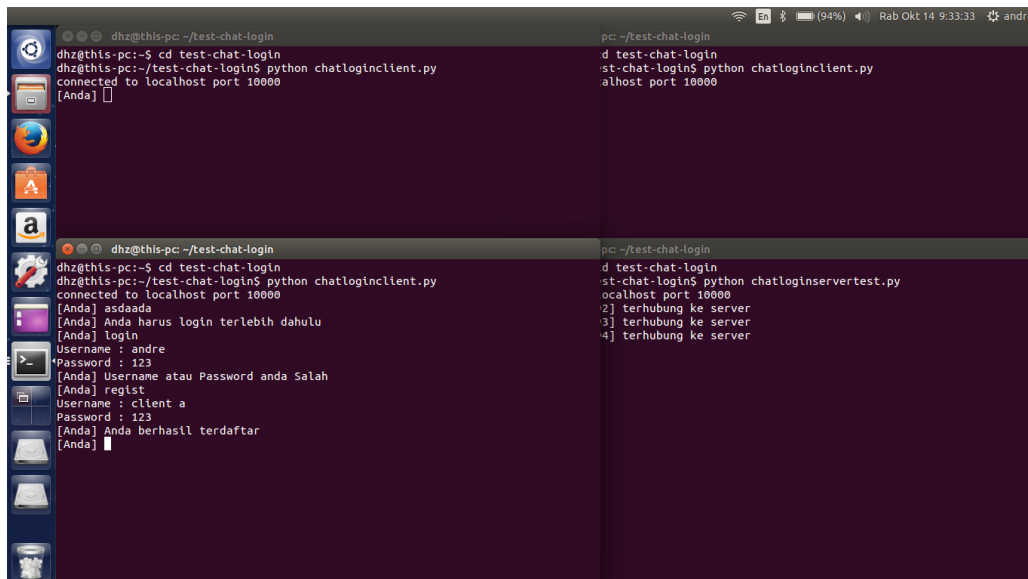


Figure 1 : Jika client belum login, maka client tidak dapat mengirim pesan

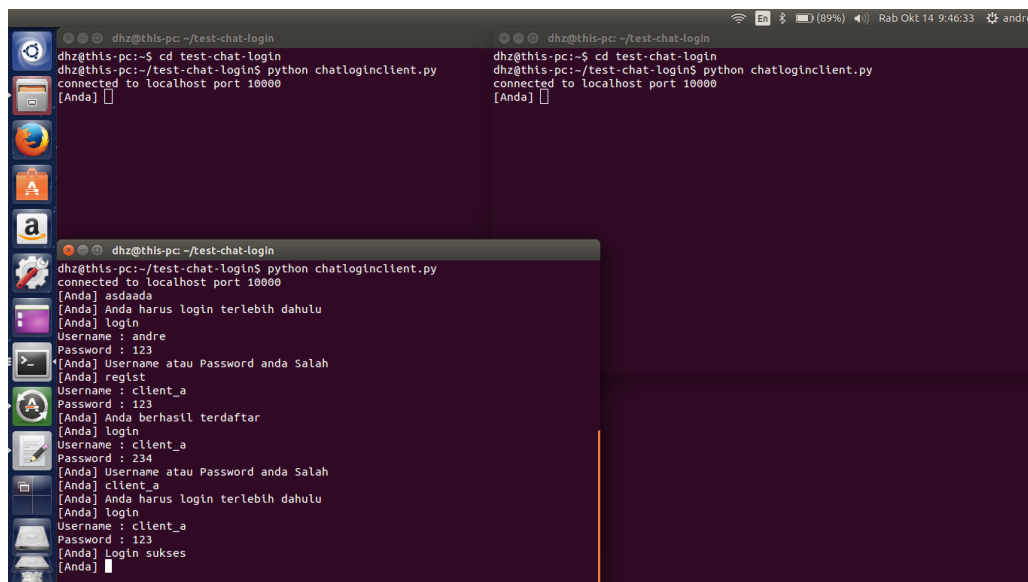
Sebelum login, client A harus melakukan registrasi terlebih dahulu dengan menggunakan command “regist”. Setelah itu, client A dapat memasukkan username dan password yang akan didaftarkan.

Setelah register, client A dapat melakukan login dengan username dan password yang telah didaftarkan.



```
dhz@this-pc: ~/test-chat-login
dhz@this-pc:~/test-chat-login$ cd test-chat-login
dhz@this-pc:~/test-chat-login$ python chatloginclient.py
connected to localhost port 10000
[Anda] 
[Anda] login
Username : andre
Password : 123
[Anda] Username atau Password anda Salah
[Anda] regist
Username : client a
Password : 123
[Anda] Anda berhasil terdaftar
[Anda]
```

Figure 2 : client melakukan registrasi



```
dhz@this-pc:~/test-chat-login
dhz@this-pc:~/test-chat-login$ cd test-chat-login
dhz@this-pc:~/test-chat-login$ python chatloginclient.py
connected to localhost port 10000
[Anda] 
[Anda] login
Username : client_a
Password : 123
[Anda] Anda berhasil terdaftar
[Anda] login
Username : client_a
Password : 123
[Anda] Login sukses
[Anda]
```

Figure 3 : client berhasil melakukan login

Apabila username dan password yang dimasukkan sama, maka server akan mengirimkan pesan kepada client A bahwa login telah sukses. Tetapi, jika username dan password yang dimasukkan tidak cocok atau username dan password yang dimasukkan belum terdaftar di server, maka server akan mengirim pesan ke client A bahwa username atau password yang dimasukkan salah.

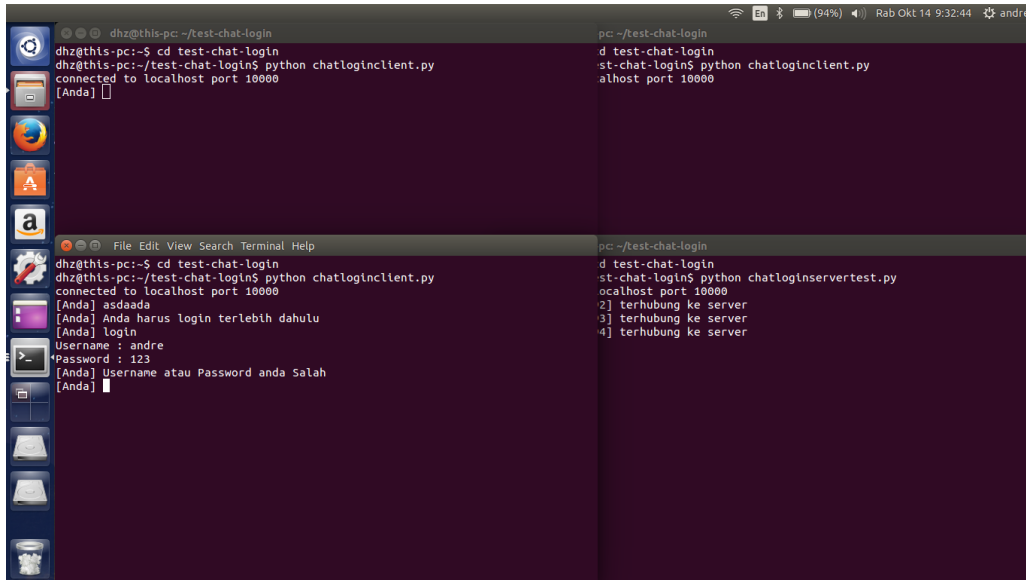


Figure 4 : ketika melakukan login tanpa register, maka server akan mengirim pesan bahwa username atau password salah, karena username belum terdaftar.

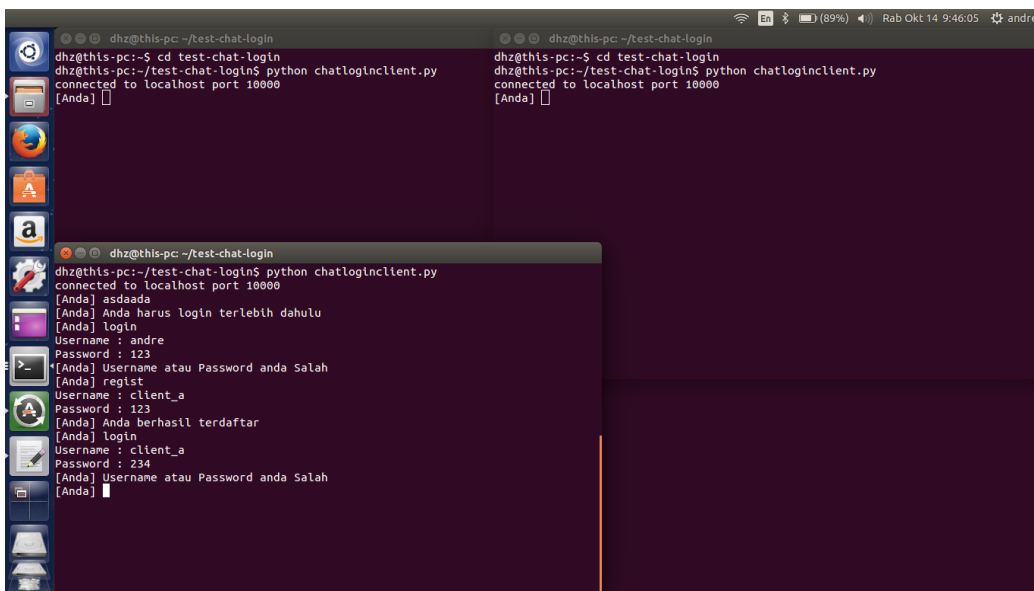


Figure 5 : ketika client melakukan login dan memasukkan username/password yang tidak sesuai, maka server akan mengirim pesan bahwa username/password yang dimasukkan salah.

Jika client A yang sudah login dan berusaha untuk melakukan login lagi, maka server akan mengirimkan pesan kepada client A bahwa yang bersangkutan telah melakukan login sebelumnya.

```
dhz@this-pc: ~/test-chat-login
dhz@this-pc:~$ cd test-chat-login
dhz@this-pc:~/test-chat-login$ python chatloginclient.py
connected to localhost port 10000
[Anda] 
[Anda] 
[Anda] Anda harus login terlebih dahulu
[Anda] login
Username : andre
Password : 123
[Anda] Username atau Password anda Salah
[Anda] regist
Username : client_a
Password : 123
[Anda] Anda berhasil terdaftar
[Anda] login
Username : client_a
Password : 234
[Anda] Username atau Password anda Salah
[Anda] client_a
[Anda] Anda harus login terlebih dahulu
[Anda] login
Username : client_a
Password : 123
[Anda] Login sukses
[Anda] login
Username : client_a
Password : 123
[Anda] Anda telah melakukan login sebelumnya.
[Anda]
```

Figure 6 : ketika client yang telah login melakukan login lagi, maka server akan mengirimkan pesan bahwa anda telah login sebelumnya.

Hal yang sama juga dilakukan oleh client B dan C.

Client yang lain harus melakukan registrasi akun terlebih dahulu sama dengan yang dilakukan oleh client A.

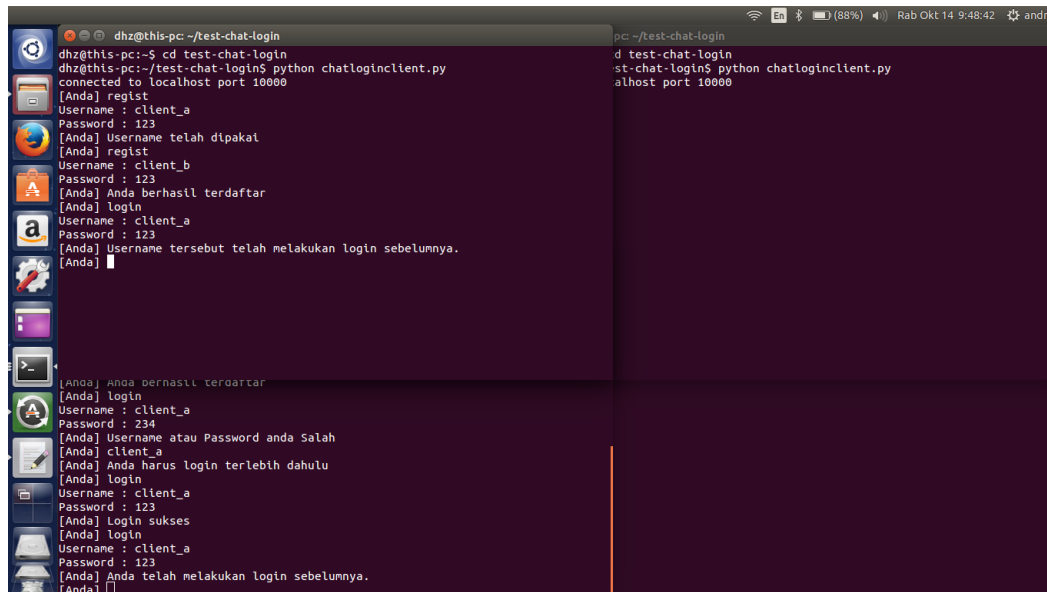
Tetapi, username yang didaftarkan oleh setiap client harus berbeda satu dengan yang lain. Jika terdapat ada client yang mendaftarkan diri menggunakan username yang sama dengan client lain yang telah terdaftar sebelumnya, maka server akan mengirim pesan kepada client yang bersangkutan bahwa username tidak dapat digunakan.

```
dhz@this-pc: ~/test-chat-login
dhz@this-pc:~$ cd test-chat-login
dhz@this-pc:~/test-chat-login$ python chatloginclient.py
connected to localhost port 10000
[Anda] regist
Username : client_a
Password : 123
[Anda] Username telah dipakai
[Anda] 
[Anda] Anda berhasil terdaftar
[Anda] login
Username : client_a
Password : 234
[Anda] Username atau Password anda Salah
[Anda] client_a
[Anda] Anda harus login terlebih dahulu
[Anda] login
Username : client_a
Password : 123
[Anda] Login sukses
[Anda] login
Username : client_a
Password : 123
[Anda] Anda telah melakukan login sebelumnya.
[Anda]
```

Figure 7 : jika client lain melakukan register menggunakan username yang sudah terdaftar sebelumnya, maka server akan mengirim pesan bahwa username telah terpakai

Setelah mendaftar, client B dan C dapat melakukan login sesuai dengan username dan password yang telah mereka buat.

Jika mereka mencoba untuk melakukan login dengan username dan password milik client lain yang sebelumnya telah melakukan login, maka server akan mengirim pesan kepada client yang bersangkutan bahwa username tersebut telah melakukan login sebelumnya.



```
dhz@this-pc: ~/test-chat-login
dhz@this-pc:~$ cd test-chat-login
dhz@this-pc:~/test-chat-login$ python chatloginclient.py
connected to localhost port 10000
[Anda] regist
Username : client_a
Password : 123
[Anda] Username telah dipakai
[Anda] regist
Username : client_b
Password : 123
[Anda] Anda berhasil terdaftar
[Anda] login
Username : client_a
Password : 123
[Anda] Username tersebut telah melakukan login sebelumnya.
[Anda]

[Anda] Anda berhasil terdaftar
[Anda] login
Username : client_a
Password : 234
[Anda] Username atau Password anda Salah
[Anda] client_a
[Anda] Anda harus login terlebih dahulu
[Anda] login
Username : client_a
Password : 123
[Anda] Login sukses
[Anda] login
Username : client_a
Password : 123
[Anda] Anda telah melakukan login sebelumnya.
[Anda]
```

```
pc: ~/test-chat-login
d test-chat-login
st-chat-login$ python chatloginclient.py
alhost port 10000
```

Figure 8 : Ketika client lain melakukan login menggunakan username yang telah login sebelumnya, maka server akan mengirim pesan bahwa username tersebut telah melakukan login sebelumnya.

Setelah client berhasil melakukan login, maka client dapat menggunakan fasilitas chat client server. Jika client belum melakukan login, maka client tidak dapat mengirim pesan apapun ke client lain.

Semua socket dan username dari client yang terhubung akan dimasukkan oleh server kedalam array `user_list[]`. Sementara, username dan password client yang telah terdaftar akan dimasukkan kedalam array `akun[]`.

PROSEDUR PENGGUNAAN PERINTAH CHAT CLIENT-SERVER

Semua client dapat melakukan pengiriman pesan kepada seluruh client yang terhubung, dimana setiap pesan akan dikirim oleh client kepada socket server dan server akan mengirimkan pesan kepada setiap client yang terhubung.

Tidak hanya itu, client juga dapat mengirim pesan hanya kepada salah satu client yang terhubung / private message dengan menggunakan perintah `send-to`.

Selain kedua fitur diatas, suatu client juga dapat melihat daftar client yang aktif dengan mengetikkan perintah `list-user`.

Supaya server dapat memahami apakah client sedang berusaha untuk mengirim pesan kepada seluruh client yang terhubung, mengirim pesan kepada salah satu client, atau sedang melihat daftar client yang terhubung, maka server perlu mendeteksi indeks pertama dari pesan yang dikirim oleh client.

Pesan tersebut akan dikirim ke socket server dan server akan memecah pesan tersebut menjadi array, dimana isi dari array tersebut adalah isi pesan yang dipisahkan dengan spasi.

Contoh:

Client A mengirim pesan berisi : Aku sayang kamu

Isi pesan tersebut akan dikirimkan ke server, selanjutnya server akan memecah pesan tersebut dan memasukkannya ke dalam array, dimana setiap indeks array diisi dengan setiap kata yang dipisahkan dengan spasi, dengan menggunakan fungsi `split`.

...

try:

```
data = socks.recv(4096)
if data:
    command = data.split()
```

...

Output dari syntax diatas adalah

```
command      = ['Aku', 'sayang', 'kamu']
command[0]    = 'Aku'
command[1]    = 'sayang'
command[2]    = 'kamu'
```

Dengan itu, server cukup melakukan pengecekan terhadap indeks ke 0 dari array. Pada indeks ke 0, apabila client memasukkan perintah seperti `send-to` dan `list-user`, maka server dapat mengetahui bahwa client sedang menggunakan perintah.

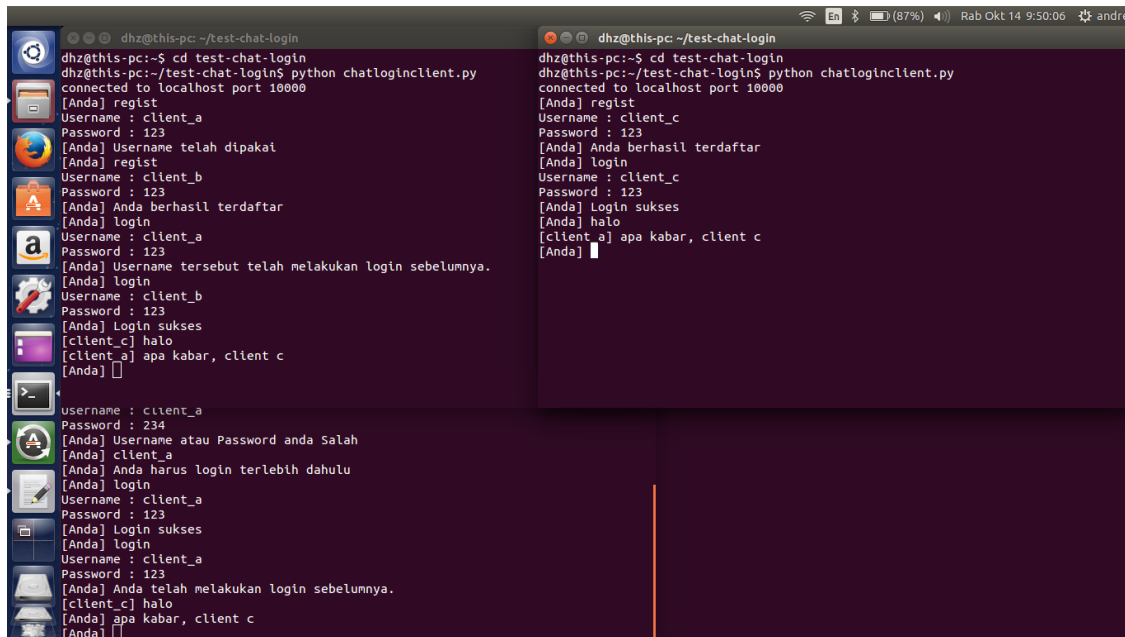
Prinsip yang sama juga dilakukan server untuk mengetahui bahwa client sedang berusaha untuk melakukan **login** atau **register**.

FITUR DAN PERINTAH YANG TERSEDIA

1. MENGIRIM PESAN KE SELURUH CLIENT YANG TERHUBUNG

Apabila suatu client ingin melakukan pengiriman pesan kepada seluruh client, maka client dapat langsung menuliskan pesan apapun, tetapi pada kata pertama tidak boleh terdapat perintah seperti `send-to` dan `list-user`, karena `send-to` dan `list-user` merupakan jenis perintah.

Pesan yang ditulis client akan dikirim ke socket server untuk selanjutnya dikirimkan ke seluruh client yang terhubung.



The image shows two terminal windows side-by-side. The left window shows a user logging in as 'client_a' and then 'client_b'. The right window shows a user logging in as 'client_c'. Both windows show the server's responses, including 'Anda berhasil terdaftar' (You successfully registered) and 'Anda berhasil melakukan login sebelumnya.' (You successfully logged in previously). The right window also shows a message from 'client_a' to 'client_c'.

```
dhz@this-pc: ~/test-chat-login
dhz@this-pc:~/test-chat-login$ python chatloginclient.py
connected to localhost port 10000
[Anda] regist
Username : client_a
Password : 123
[Anda] Username telah dipakat
[Anda] regist
Username : client_b
Password : 123
[Anda] Anda berhasil terdaftar
[Anda] login
Username : client_a
Password : 123
[Anda] Username tersebut telah melakukan login sebelumnya.
[Anda] login
Username : client_b
Password : 123
[Anda] Login sukses
[client_c] halo
[client_a] apa kabar, client c
[Anda]

Username : client_a
Password : 234
[Anda] Username atau Password anda Salah
[Anda] client_a
[Anda] Anda harus login terlebih dahulu
[Anda] login
Username : client_a
Password : 123
[Anda] Login sukses
[Anda] login
Username : client_a
Password : 123
[Anda] Anda telah melakukan login sebelumnya.
[client_c] halo
[Anda] apa kabar, client c
[Anda]
```

Figure 9 : client dapat mengirim pesan ke seluruh client dengan mengetikkan pesan secara langsung.

2. MENGIRIM PESAN KE SEORANG CLIENT

Untuk mengirim pesan kepada client tertentu, client dapat mengetikkan perintah berikut:

send-to<spasi>nama_client_yang_dituju<spasi>isi pesan

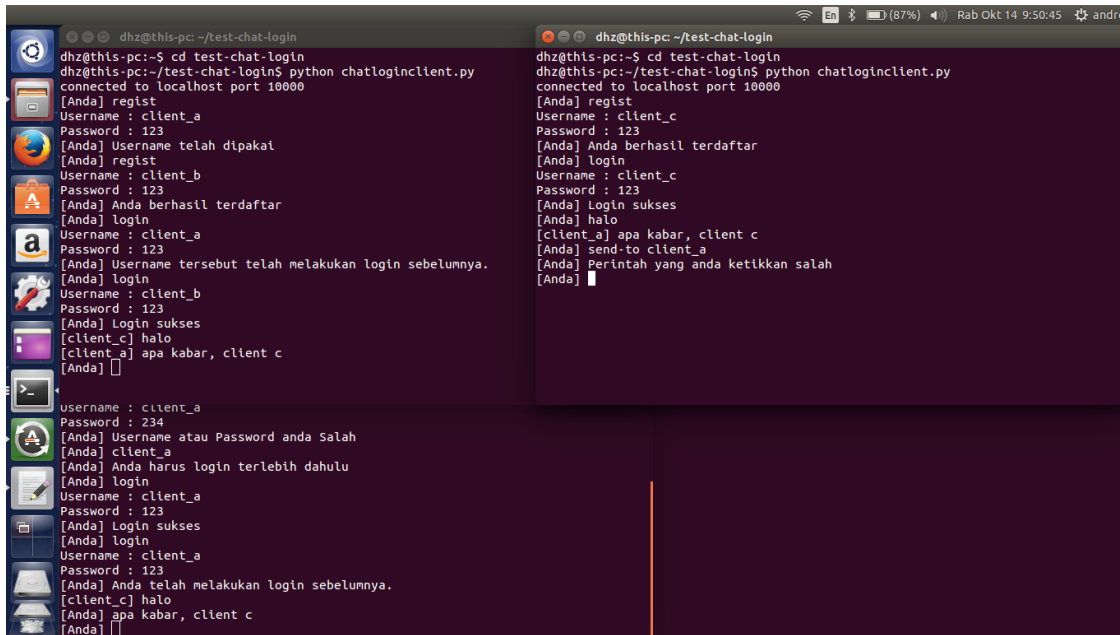
Server akan melakukan pemisahan setiap kata yang diterima oleh socket server dari client dan dimasukkan kedalam sebuah array menggunakan fungsi split seperti yang telah dijelaskan sebelumnya dan melakukan pengecekan terhadap indeks ke-0 dari pesan yang dikirim.

Jika server mendapati indeks ke 0 adalah string "send-to", maka selanjutnya user akan melakukan pengecekan apakah perintah send-to yang dikirimkan ke server valid.

Perintah send-to yang valid terdiri dari:

command[0]	= 'send-to'
command[1]	= 'nama_client_yang_dituju'
command[2] dan selanjutnya	= 'isi pesan'

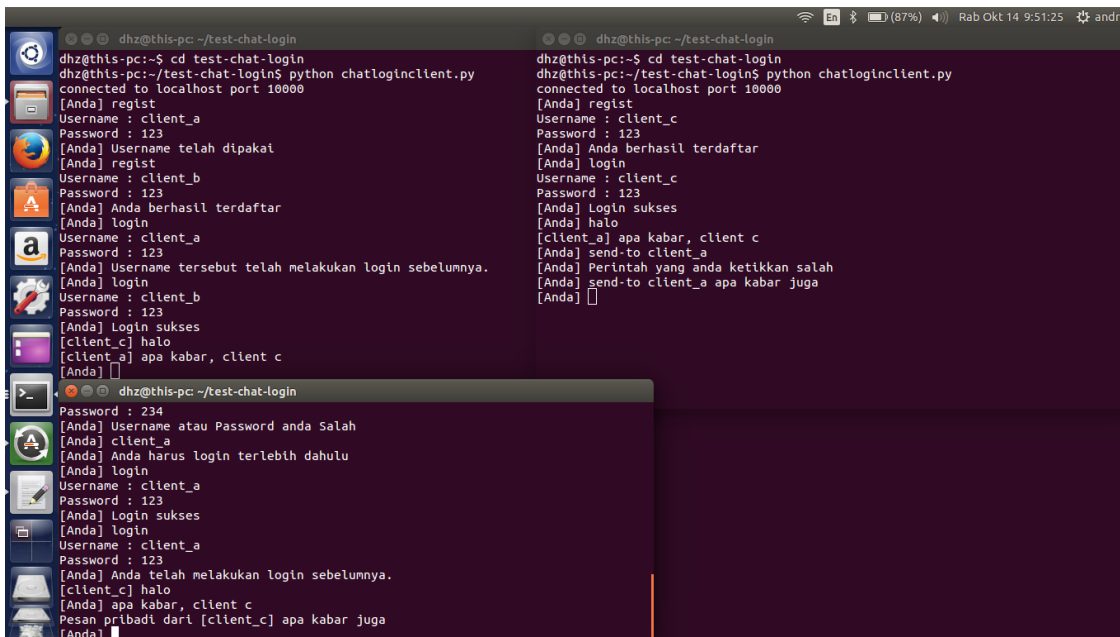
Jika panjang array yang dimasukkan lebih kecil dari 3, maka server akan mengirimkan pesan ke client bahwa inputan perintah yang dimasukkan salah.



```
dhz@this-pc: ~/test-chat-login
dhz@this-pc:~/test-chat-login$ python chatloginclient.py
connected to localhost port 10000
[Anda] regist
Username : client_a
Password : 123
[Anda] Username telah dipakat
[Anda] regist
Username : client_b
Password : 123
[Anda] Anda berhasil terdaftar
[Anda] login
Username : client_a
Password : 123
[Anda] Username tersebut telah melakukan login sebelumnya.
[Anda] login
Username : client_b
Password : 123
[Anda] Login sukses
[client_c] halo
[client_a] apa kabar, client c
[Anda]
Username : client_a
Password : 234
[Anda] Username atau Password anda Salah
[Anda] client_a
[Anda] Anda harus login terlebih dahulu
[Anda] login
Username : client_a
Password : 123
[Anda] Login sukses
[Anda] login
Username : client_a
Password : 123
[Anda] Anda telah melakukan login sebelumnya.
[client_c] halo
[Anda] apa kabar, client c
[Anda]
[Anda] send-to client_a
[Anda] Perintah yang anda ketikkan salah
[Anda]
```

Figure 10 : Jika tidak sesuai dengan syarat pengiriman perintah, maka server akan mengirim pesan bahwa perintah yang dimasukkan salah.

Jika inputan yang dimasukkan benar, maka server akan mengirimkan pesan kepada nama client yang dituju dengan cara mengambil socket dari client yang bersangkutan dan mengirim pesan ke client tersebut.



```
dhz@this-pc: ~/test-chat-login
dhz@this-pc:~/test-chat-login$ python chatloginclient.py
connected to localhost port 10000
[Anda] regist
Username : client_a
Password : 123
[Anda] Username telah dipakat
[Anda] regist
Username : client_b
Password : 123
[Anda] Anda berhasil terdaftar
[Anda] login
Username : client_a
Password : 123
[Anda] Username tersebut telah melakukan login sebelumnya.
[Anda] login
Username : client_b
Password : 123
[Anda] Login sukses
[client_c] halo
[client_a] apa kabar, client c
[Anda]
[Anda] send-to client_a
[Anda] Perintah yang anda ketikkan salah
[Anda] send-to client_a apa kabar juga
[Anda]
```

Figure 11 : perintah send-to berhasil dijalankan

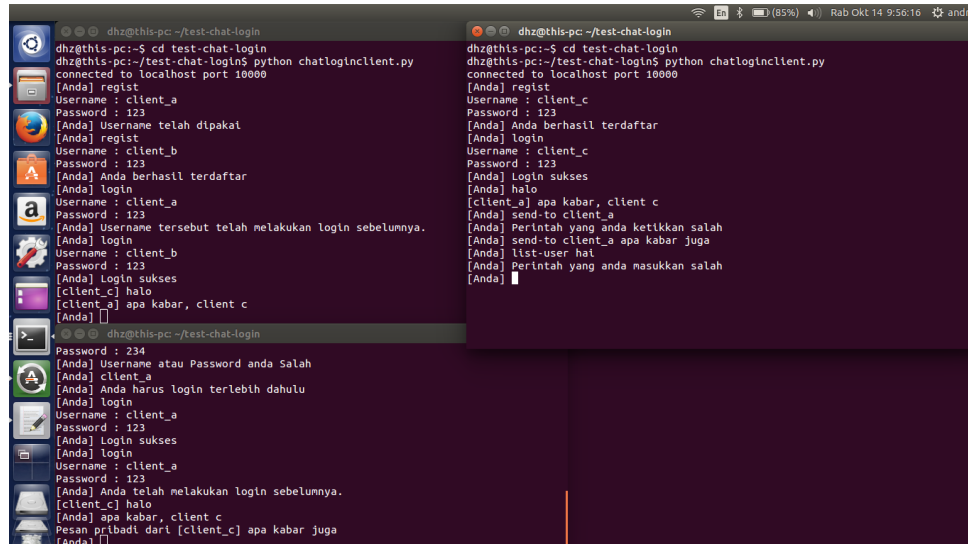
3. MELIHAT DAFTAR CLIENT YANG SEDANG TERSAMBUNG

Suatu client dapat melihat siapa saja client yang terhubung cukup dengan mengetikkan perintah `list-user`.

Perintah `list-user` yang valid:

`command[0]` = `list-user`

Jika panjang array yang dimasukkan lebih dari 1, maka server akan mengirim pesan ke client bahwa inputan perintah yang dimasukkan salah.



```
dhz@this-pc:~/test-chat-login
dhz@this-pc:~/test-chat-login$ python chatloginclient.py
connected to localhost port 10000
[Anda] regist
Username : client_a
Password : 123
[Anda] Username telah dipakai
[Anda] regist
Username : client_b
Password : 123
[Anda] Anda berhasil terdaftar
[Anda] login
Username : client_a
Password : 123
[Anda] Username tersebut telah melakukan login sebelumnya.
[Anda] login
Username : client_b
Password : 123
[Anda] Login sukses
[client_c] halo
[client_a] apa kabar, client c
[Anda] []

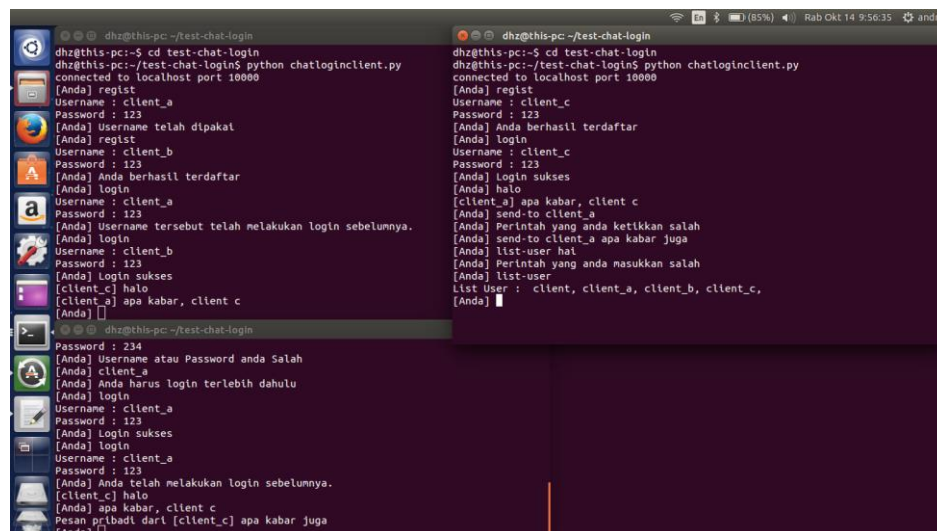
dhz@this-pc:~/test-chat-login
Password : 234
[Anda] Username atau Password anda Salah
[Anda] client_a
[Anda] Anda harus login terlebih dahulu
[Anda] login
Username : client_a
Password : 123
[Anda] Login sukses
[Anda] login
Username : client_a
Password : 123
[Anda] Anda telah melakukan login sebelumnya.
[client_c] halo
[Anda] apa kabar, client c
Pesan pribadi dari [client_c] apa kabar juga
[Anda] []
```

```
dhz@this-pc:~/test-chat-login
dhz@this-pc:~/test-chat-login$ python chatloginclient.py
connected to localhost port 10000
[Anda] regist
Username : client_c
Password : 123
[Anda] Anda berhasil terdaftar
[Anda] login
Username : client_c
Password : 123
[Anda] Login sukses
[Anda] halo
[client_a] apa kabar, client c
[Anda] send-to client_a
[Anda] Perintah yang anda ketikkan salah
[Anda] send-to client_a apa kabar juga
[Anda] list-user hal
[Anda] Perintah yang anda masukkan salah
[Anda] list-user
list user : client, client_a, client_b, client_c.
[Anda] []
```

Figure 12 : pengiriman perintah `list-user` yang tidak sesuai.

Pesan yang berisi perintah `list-user` akan dikirimkan kepada server untuk selanjutnya dilakukan `split`.

Jika server mendapati bahwa indeks ke 0 adalah `list-user`, maka server akan mengirimkan data kepada client yang bersangkutan berupa daftar client yang terhubung ke server.



```
dhz@this-pc:~/test-chat-login
dhz@this-pc:~/test-chat-login$ python chatloginclient.py
connected to localhost port 10000
[Anda] regist
Username : client_a
Password : 123
[Anda] username telah dipakai
[Anda] regist
Username : client_b
Password : 123
[Anda] Anda berhasil terdaftar
[Anda] login
Username : client_a
Password : 123
[Anda] Username tersebut telah melakukan login sebelumnya.
[Anda] login
Username : client_b
Password : 123
[Anda] Login sukses
[client_c] halo
[client_a] apa kabar, client c
[Anda] []

dhz@this-pc:~/test-chat-login
Password : 234
[Anda] Username atau Password anda Salah
[Anda] client_a
[Anda] Anda harus login terlebih dahulu
[Anda] login
Username : client_a
Password : 123
[Anda] Login sukses
[Anda] login
Username : client_a
Password : 123
[Anda] Anda telah melakukan login sebelumnya.
[client_c] halo
[Anda] apa kabar, client c
Pesan pribadi dari [client_c] apa kabar juga
[Anda] []
```

```
dhz@this-pc:~/test-chat-login
dhz@this-pc:~/test-chat-login$ python chatloginclient.py
connected to localhost port 10000
[Anda] regist
Username : client_c
Password : 123
[Anda] Anda berhasil terdaftar
[Anda] login
Username : client_c
Password : 123
[Anda] Login sukses
[Anda] halo
[client_a] apa kabar, client c
[Anda] send-to client_a
[Anda] Perintah yang anda ketikkan salah
[Anda] send-to client_a apa kabar juga
[Anda] list-user hal
[Anda] Perintah yang anda masukkan salah
[Anda] list-user
list user : client, client_a, client_b, client_c.
[Anda] []
```