ChatBox

Pemrograman Jaringan C - 05111740000029 - Ananta Dwi Prasetya Purna Yuda

# Deskripsi

ChatBox merupakan service chatting yang mana pesan dikirim ke tujuan user lain dan user tersebut dapat melihat isi pesan tersebut. Service ini menerapkan arsitektur client-server dimana user/client akan dapat berkomunikasi dengan user/client lain dengan bantuan server dan semua data disimpan dalam server.

# Format Data

Komunikasi dikirimkan ke socket dengan mengirimkan teks alfanumerik dengan format

COMMAND spasi PARAMETER1 spasi PARAMETER2 spasi PARAMETER3. Pada dokumentasi ini format data untuk setiap fitur dapat dilihat pada parameter input bagian API Request.

# Daftar Fitur

## Autentikasi Login

### Deskripsi dan Tujuan

Fitur yang digunakan untuk autentikasi dengan cara mendaftarkan sesi user ke server

### Parameter input

#### Input CLI

auth <username> <password>

username (string) : username user

password (string) : password user

#### API Request

auth <username> <password>

username (string) : username user

password (string) : password user

### Hasil keluaran

#### Sukses

{'status': 'OK', 'tokenid': tokenid}

tokenid (string) : id token / id sesi yang digunakan sebagai autentikasi

#### Error

User tidak ditemukan

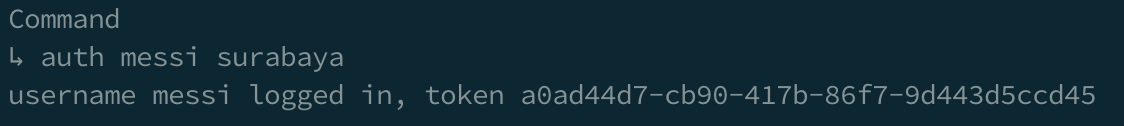
{'status': 'ERROR', 'message': 'User Tidak Ada'}

Password salah

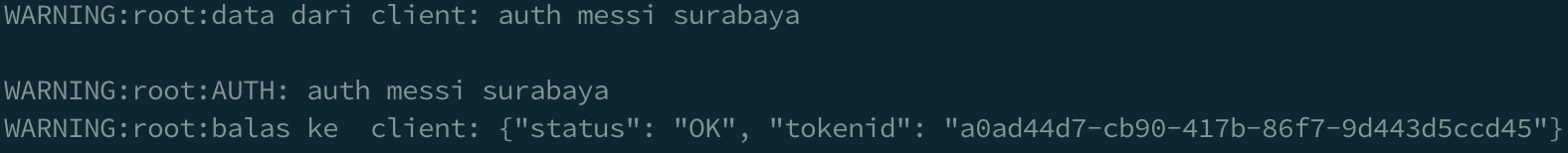
{'status': 'ERROR', 'message': 'Password Salah'}

### Contoh message input dan keluaran

Client



Server



## Logout

### Deskripsi dan Tujuan

Fitur yang digunakan untuk menghapus sesi user dari server. Ketika program client diterminate (Ctrl+C) maka secara otomatis program tersebut akan mengirim perintah logout ke server.

### Parameter input

#### Input CLI

logout

#### API Request

logout <sessionid>

sessionid (string) : id token/id sesi milik user

### Hasil keluaran

#### Sukses

{'status': 'OK'}

#### Error

Sesi tidak ditemukan

{'status': 'ERROR', 'message': 'Session Tidak Ditemukan'}

### Contoh message input dan keluaran

Client

A picture containing drawing

Description automatically generated

Server

A black sign with white text

Description automatically generated

## Mengirim Pesan

### Deskripsi dan Tujuan

Fitur yang digunakan untuk mengirim pesan text ke user lain

### Parameter input

#### Input CLI

send <username\_dest> <message>

username\_dest (string) : username tujuan

message (string) : konten pesan yang akan dikirim (dapat berisi spasi)

#### API Request

send <sessionid> <username\_dest> <message>

sessionid (string) : id token/id sesi milik user

username\_dest (string) : username tujuan

message (string) : konten pesan yang akan dikirim (dapat berisi spasi)

### Hasil keluaran

#### Sukses

{'status': 'OK', 'message': 'Message Sent'}

#### Error

Sesi tidak ditemukan

{'status': 'ERROR', 'message': 'Session Tidak Ditemukan'}

User tujuan tidak ditemukan

{'status': 'ERROR', 'message': 'User Tidak Ditemukan'}

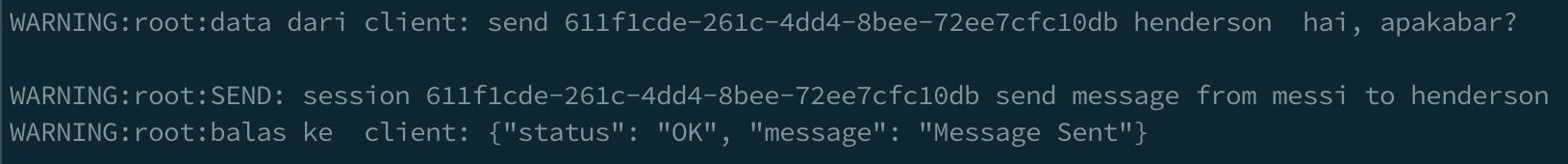
### Contoh message input dan keluaran

Client

A picture containing drawing

Description automatically generated

Server



## Melihat Pesan Masuk

### Deskripsi dan Tujuan

Fitur yang digunakan untuk melihat pesan masuk yang dikirim user lain

### Parameter input

#### Input CLI

inbox

#### API Request

inbox <sessionid>

sessionid (string) : id token/id sesi milik user

### Hasil keluaran

#### Sukses

{'status': 'OK', 'messages': messages}

messages (JSON Object) : isi pesan masuk

Contoh format JSON messages pada user henderson

{"messi": [{"msg\_from": "Lionel Messi", "msg\_to": "Jordan Henderson", "msg": " hai, apakabar? \r\n"}]}

msg\_from (string) : Nama user pengirim

msg\_to(string) : Nama user penerima

msg (string) : Isi pesan

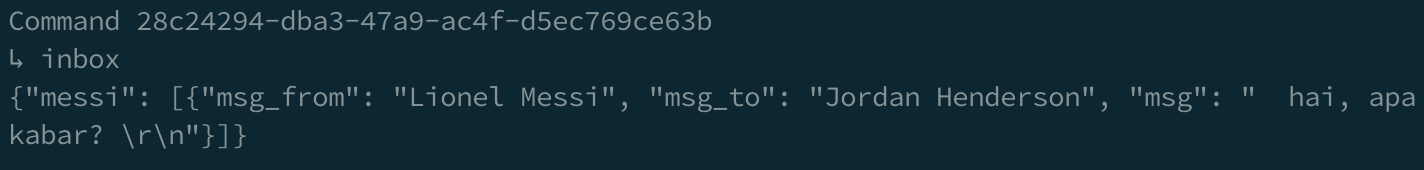
#### Error

Sesi tidak ditemukan

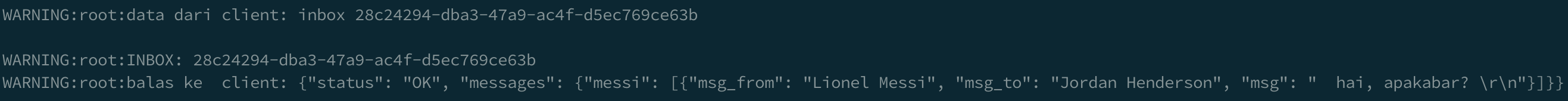
{'status': 'ERROR', 'message': 'Session Tidak Ditemukan'}

### Contoh message input dan keluaran

Client



Server



## Melihat User Online

### Deskripsi dan Tujuan

Fitur yang digunakan untuk melihat user yang sedang online

### Parameter input

#### Input CLI

online

#### API Request

online <sessionid>

sessionid (string) : id token/id sesi milik user

### Hasil keluaran

#### Sukses

{'status': 'OK', 'users': users}

users (JSON Array) : daftar user yang online

Contoh format JSON users pada user henderson

[{"username": "messi", "userdetail": {"nama": "Lionel Messi", "negara": "Argentina"}}]

username (string) : Username user

userdetail (JSON Object) : Obyek detail user

nama (string) : Nama user

negara (string) : Asal negara user

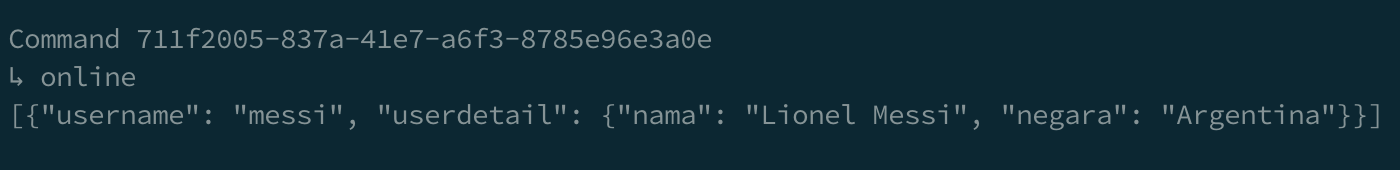
#### Error

Sesi tidak ditemukan

{'status': 'ERROR', 'message': 'Session Tidak Ditemukan'}

### Contoh message input dan keluaran

Client



Server

