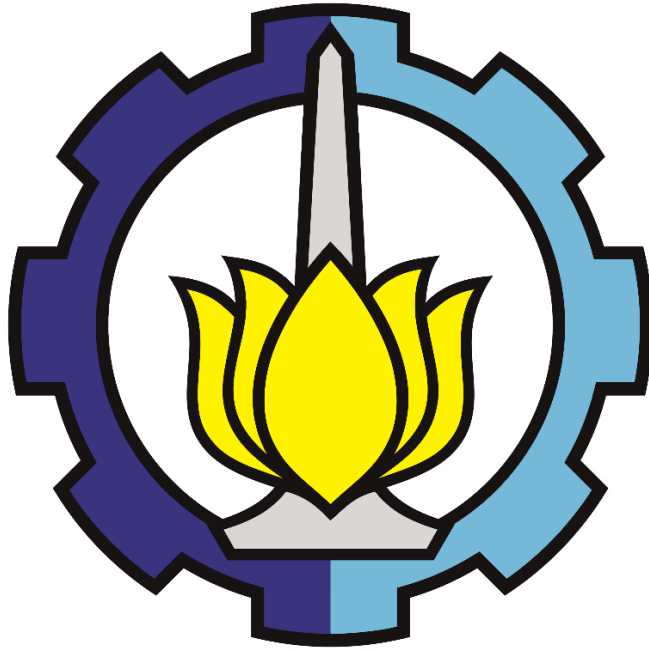


# DOKUMENTASI PROTOKOL

## *Chat Service*



Kelas :

Pemrograman Jaringan – C

Ditulis oleh:

Ersad Ahmad Ishlahuddin

05111740000016

**DEPARTEMEN TEKNIK INFORMATIKA**

**FAKULTAS TEKNOLOGI ELEKTRO DAN INFORMATIKA CERDAS**

**INSTITUT TEKNOLOGI SEPULUH NOPEMBER**

**2020**

## Chat Service

*Service* ini merupakan *service* yang memungkinkan pengguna yang berada pada satu jaringan untuk berkomunikasi lewat *chat*. *Service* ini menggunakan arsitektur *server-client* dan menggunakan *multithreading* dalam penerapannya, sehingga banyak *client* dapat terhubung ke 1 *server*.

Langkah pertama yang dilakukan adalah menghidupkan *server* terlebih dahulu. Setelah *client* terhubung dengan *server*, *client* melalui *command line interface* dapat menggunakan fitur yang disediakan oleh sistem. Secara umum *client* dapat mengirimkan *command* lewat *CLI* tersebut diikuti parameter (jika ada) sesuai protokol. Keamanan pada sistem ini terletak pada *session* pada masing-masing *user* dan enkripsi pada saat pengiriman data, dimana setiap kali *user* menggunakan fitur (selain *login*) harus disertai dengan *sessionID* yang didapatkan ketika *login*.

*Chat service* menyediakan fitur-fitur pada umumnya, antara lain:

- *Login*, untuk masuk ke dalam sistem
- *Logout*, untuk keluar dari sistem
- *Send Message*, untuk mengirim pesan pada *user* yang lain
- *List Inbox*, untuk melihat daftar pesan yang *user* terima
- *List Outbox*, untuk melihat daftar pesan yang dikirimkan oleh *user*
- *Display Active User*, untuk melihat *user* active (sedang login) di dalam sistem

## Format Data

Secara garis besar format data pada *client* adalah

COMMAND (spasi) PARAMETER1 (spasi) PARAMETER2 (dst jika ada)

Komunikasi pada *Chat Service* ini menggunakan format data alfanumerik pada *client*, dan *client* akan menerima respon JSON dari *command* yang dikirim. Setiap fitur memiliki format request dan response tersendiri yang akan dijelaskan pada section di bawah ini.

## Daftar Fitur

### 1. Login (Autentikasi User)

- Deskripsi dan Tujuan

Fitur ini memungkinkan *user* untuk masuk ke dalam *service* dan menggunakan fitur yang disediakan oleh *service*. Autentikasi pada *service* menggunakan pencocokan antara *username* dan *password* yang diinputkan oleh *client* dengan data yang ada pada *server*, kemudian setiap *user* akan menerima *sessionID* masing-masing. Jika sesuai maka *user* dapat masuk ke dalam *service*, jika tidak sesuai *user* bisa mencoba memasukkan *username* dan *password* lagi.

Tujuan:

- Memverifikasi *user* yang akan masuk ke dalam *service*, autentikasi menggunakan pencocokan *username* dan *password*.
- Meningkatkan keamanan pada *service*, hanya *user* terautentikasi yang bisa menggunakan fitur yang disediakan.

- Parameter input

**auth [username] [password]**

Command = auth

Param1 = *username* yang dimiliki oleh *user*

Param2 = *password* yang dimiliki oleh *user*

Setiap *command* dan parameter dipisahkan dengan spasi

- Hasil Keluaran  
Hasil keluaran dari fitur merupakan response berupa JSON yang dikirimkan oleh server dan diterima oleh client, terdiri dari 2 jenis response:

Jika sukses, *user* berhasil masuk ke dalam *service*

**username [username] logged in, token [tokenID]**

Jika gagal, *user* tidak ditemukan atau password *user* tidak sesuai

**ERROR, [message]**

- Contoh

Client

```
Command :auth messi surabaya
username messi logged in, token 2c7c6bb1-b9af-4c6c-a868-29d0254e8d97
```

Server

```
WARNING:root:AUTH: auth messi surabaya
WARNING:root:balas ke client: {"status": "OK", "tokenid": "2c7c6bb1-b9af-4c6c-a868-29d0254e8d97"}
```

Client

```
Command :auth messi jakarta
Error, Password Salah
```

Server

```
WARNING:root:AUTH: auth messi jakarta
WARNING:root:balas ke client: {"status": "ERROR", "message": "Password Salah"}
```

## 2. Logout

- Deskripsi dan Tujuan  
Fitur ini memungkinkan *user* untuk keluar dari *service* dan tidak dapat menggunakan fitur yang disediakan oleh *service*. Ketika *user* logout maka sessionID yang didapat ketika login akan dihapus dari *service*. Pada saat logout hanya memerlukan sessionID, dimana tidak bisa logout jika tidak login.

Tujuan:

- Meningkatkan keamanan pada akun, mencegah agar akun tidak digunakan oleh orang lain

- Parameter input

**logout**

Command = logout

- Hasil Keluaran

Hasil keluaran dari fitur merupakan response berupa JSON yang dikirimkan oleh server dan diterima oleh client, terdiri dari 2 jenis response:

Jika sukses, *user* berhasil keluar dari *service*

**username [username] logged out**

Jika gagal,

**ERROR, [message]**

- Contoh

Client

```
Command 171e07d0-b669-43ef-8caa-587fe8fd73ae:logout
username messi logged out
```

Server

```
WARNING:root:LOGOUT: logout 171e07d0-b669-43ef-8caa-587fe8fd73ae
WARNING:root:balas ke client: {"status": "OK", "username": "messi"}
```

Client (Error karena tidak mempunyai sessionID atau belum login)

```
Command :logout
Error, not authorized
```

### 3. Send Message

- Deskripsi dan Tujuan  
Fitur ini memungkinkan *user* untuk mengirim pesan ke *user* lain yang telah terdaftar di dalam server. Pesan yang dikirimkan akan disimpan di outbox *user* dan pesan yang diterima *user* lain akan disimpan di inbox masing-masing.

Tujuan:

- Berkomunikasi dengan *user* lain yang telah terdaftar dalam server.

- Parameter input

**send [username] [message]**

Command = send

Param1 = *username user* (receiver) yang akan menerima pesan

Param2 = pesan yang akan dikirimkan oleh *user* (sender)

Setiap command dan parameter dipisahkan dengan spasi, dan message bisa lebih dari 1 kata

- Hasil Keluaran

Hasil keluaran dari fitur merupakan response berupa JSON yang dikirimkan oleh server dan diterima oleh client, terdiri dari 2 jenis response:

Jika sukses, *user* berhasil mengirim pesan ke receiver

**message sent to [username]**

Jika gagal,

**ERROR, [message]**

- Contoh

Client (messi)

```
Command c955a5c0-5b2e-4363-8b32-d0cfb5df99b1:send henderson halo henderson
, bagaimana kabarnya?
send c955a5c0-5b2e-4363-8b32-d0cfb5df99b1 henderson halo henderson, bagaima
na kabarnya?

message sent to henderson
```

Server

```
WARNING:root:data dari client: send c955a5c0-5b2e-4363-8b32-d0cfb5df99b1 h
enderson halo henderson, bagaimana kabarnya?

WARNING:root:SEND: session c955a5c0-5b2e-4363-8b32-d0cfb5df99b1 send messa
ge from messi to henderson
WARNING:root:balas ke client: {"status": "OK", "message": "Message Sent"}
```

### 4. Inbox

- Deskripsi dan Tujuan  
Fitur ini memungkinkan *user* untuk melihat pesan yang diterimanya. Response yang diterima berupa JSON list pengirim dan detail pesan-pesannya.

Tujuan:

- Melihat pesan yang pernah diterima, sebagai history
- Parameter input  
**inbox**  
Command = inbox
- Hasil Keluaran  
Hasil keluaran dari fitur merupakan response berupa JSON yang dikirimkan oleh server dan diterima oleh client, terdiri dari 2 jenis response:  
Jika sukses, mendapatkan list sender dan detail pesannya  
**{[username]: [messagedetail]}**  
Jika gagal,  
**ERROR, [message]**

- Contoh  
Client (henderson)

```
Command 6dcef39f-987b-44fa-a6b0-d4cba19082b5:inbox
{"messi": [{"msg_from": "Lionel Messi", "msg_to": "Jordan Henderson", "msg": " halo henderson, bagaimana kabarnya? \r\n"}, {"msg_from": "Lionel Messi", "msg_to": "Jordan Henderson", "msg": " keadaan di sana bagaimana? \r\n"}]}
```

Server

```
WARNING:root:INBOX: 6dcef39f-987b-44fa-a6b0-d4cba19082b5
WARNING:root:balas ke client: {"status": "OK", "messages": {"messi": [{"msg_from": "Lionel Messi", "msg_to": "Jordan Henderson", "msg": " halo henderson, bagaimana kabarnya? \r\n"}, {"msg_from": "Lionel Messi", "msg_to": "Jordan Henderson", "msg": " keadaan di sana bagaimana? \r\n"}]}}
```

## 5. Outbox

- Deskripsi dan Tujuan  
Fitur ini memungkinkan *user* untuk melihat pesan yang dikirimnya. Response yang diterima berupa JSON list penerima dan detail pesan-pesannya.

Tujuan:

- Melihat pesan yang pernah dikirim, sebagai history
- Parameter input  
**outbox**  
Command = outbox
- Hasil Keluaran  
Hasil keluaran dari fitur merupakan response berupa JSON yang dikirimkan oleh server dan diterima oleh client, terdiri dari 2 jenis response:  
Jika sukses, mendapatkan list receiver dan detail pesannya  
**{[username]: [messagedetail]}**  
Jika gagal,  
**ERROR, [message]**

- Contoh  
Client (messi)

```
Command c955a5c0-5b2e-4363-8b32-d0cfb5df99b1:outbox
{"messi": [{"msg_from": "Lionel Messi", "msg_to": "Jordan Henderson", "msg": " halo henderson, bagaimana kabarnya? \r\n"}, {"msg_from": "Lionel Messi", "msg_to": "Jordan Henderson", "msg": " keadaan di sana bagaimana? \r\n"}]}
```

Server

```
WARNING:root:OUTBOX: c955a5c0-5b2e-4363-8b32-d0cfb5df99b1
WARNING:root:balas ke client: {"status": "OK", "messages": {"messi": [{"msg_from": "Lionel Messi", "msg_to": "Jordan Henderson", "msg": " halo henderson, bagaimana kabarnya? \r\n"}, {"msg_from": "Lionel Messi", "msg_to": "Jordan Henderson", "msg": " keadaan di sana bagaimana? \r\n"}]}}
```

## 6. List Active User

- Deskripsi dan Tujuan  
Fitur ini memungkinkan *user* untuk melihat *user* lain yang telah terdaftar dalam sistem dan dalam keadaan sedang login ke sistem (online). Response yang diterima berupa JSON detail *user*nya.

Tujuan:

- Mempermudah *user* dengan bisa melihat *user* lain yang sedang online.

- Parameter input

**active**

Command = active

- Hasil Keluaran

Hasil keluaran dari fitur merupakan response berupa JSON yang dikirimkan oleh server dan diterima oleh client, terdiri dari 2 jenis response:

Jika sukses, mendapatkan list *user* yang online

**[{detailuser}, {detailuser}]**

Jika gagal,

**ERROR, [message]**

- Contoh

Client

```
Command c955a5c0-5b2e-4363-8b32-d0cfb5df99b1:active
[{"username": "henderson", "nama": "Jordan Henderson", "negara": "Inggris"}, {"username": "messi", "nama": "Lionel Messi", "negara": "Argentina"}]
Command c955a5c0-5b2e-4363-8b32-d0cfb5df99b1:
```

Server

```
WARNING:root:ACTIVE USER REQ by: c955a5c0-5b2e-4363-8b32-d0cfb5df99b1
WARNING:root:balas ke client: {"status": "OK", "messages": [{"username": "henderson", "nama": "Jordan Henderson", "negara": "Inggris"}, {"username": "messi", "nama": "Lionel Messi", "negara": "Argentina"}]}}
```