**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 *multhreading* 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 *user*name 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 *user*name dan password lagi.

Tujuan:

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

**auth [*user*name] [password]**

Command = auth

Param1 = *user*name 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*

***user*name [*user*name] logged in, token [tokenID]**

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

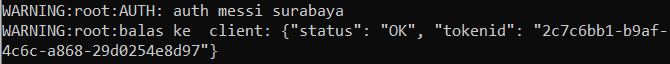
**ERROR, [message]**

* Contoh

Client



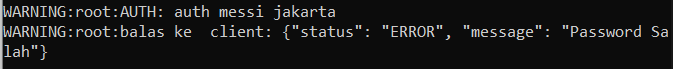
Server



Client



Server



1. 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*

***user*name [*user*name] logged out**

Jika gagal,

**ERROR, [message]**

* Contoh

Client



Server



Client (Error karena tidak mempunya sessionID atau belum login)



1. 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 [*user*name] [message]**

Command = send

Param1 = *user*name *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

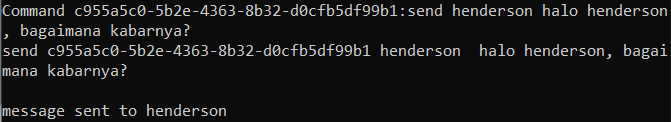
**message sent to [*user*name]**

Jika gagal,

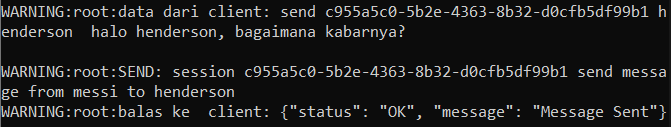
**ERROR, [message]**

* Contoh

Client (messi)



Server



1. 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

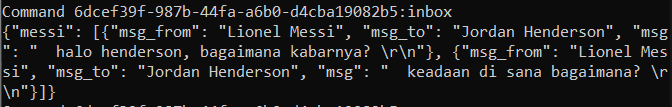
**{[*user*name]: [messagedetail]}**

Jika gagal,

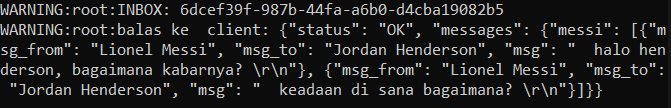
**ERROR, [message]**

* Contoh

Client (henderson)



Server



1. 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

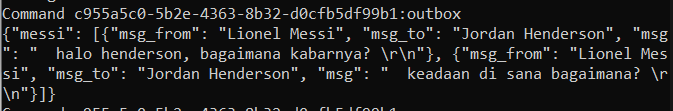
**{[*user*name]: [messagedetail]}**

Jika gagal,

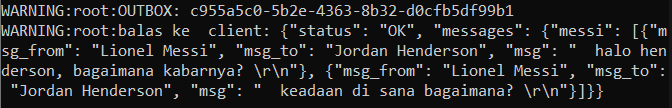
**ERROR, [message]**

* Contoh

Client (messi)



Server



1. 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 (onlline). 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

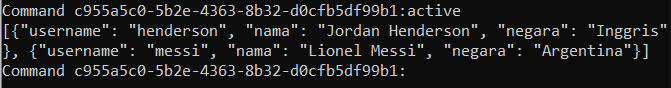
**[{detail*user*}, {detail*user*}]**

Jika gagal,

**ERROR, [message]**

* Contoh

Client



Server

