Dokumentasi Protokol

# Login

Login dilakukan dengan menggunakan mekanisme pengamanan dengan pertukaran kunci ***Diffie-Hellman Key Exchange Algorithm*** terlebih dahulu lalu *username* dan *password* dikirimkan terenkripsi dengan algoritma ***Advanced Encryption Standard* (AES)** dengan kunci hasil pertukaran yang dilakukan sebelumnya.

Urutan:

1. Client mengirimkan pesan berisi kode MSG\_LOGIN (0x00).

|  |
| --- |
| MSG\_LOGIN (byte) |

1. Server membalas dengan pesan berisi nilai ***p*** dan ***g*** dalam format UTF.

|  |  |
| --- | --- |
| p (UTF) | g (UTF) |

1. Client lalu membuat pasangan kunci publik dan private. Panjang kunci publik dalam satuan byte (CLI\_PUB\_LENGTH) dan isi dari kunci publik (CLIPUB) ini lalu dikirimkan ke server.

|  |  |
| --- | --- |
| CLI\_PUB\_LENGTH (int) | CLI\_PUB (bytes) |

1. Server lalu membalas dengan panjang kunci publik dalam satuan byte (SERV\_PUB\_LENGTH) dan isi dari kunci publik milik server (SERV\_PUB) ke Client.

|  |  |
| --- | --- |
| SERV\_PUB\_LENGTH (int) | SERV\_PUB (bytes) |

1. Client kemudian akan menghasilkan *secret code* bersama. Lalu username dan password akan dienkripsi dengan AES menjadi AES\_UNAME dan AES\_PASSWD. Client akan mengirimkan secara berturut-turut panjang dari AES\_NAME (AES\_UNAME\_LENGHTH), AES\_UNAME, panjang dari AES\_PASSWD (AES\_PASSWD\_LENGTH), dan AES\_PASSWD ke server.

|  |  |  |  |
| --- | --- | --- | --- |
| AES\_UNAME\_LENGHTH (int) | AES\_UNAME (bytes) | AES\_PASSWD\_LENGTH (int) | AES\_PASSWD (bytes) |

1. Server akan mendekripsi *username* dan *password* yang dikirimkan tadi dengan terlebih dahulu menghasilkan *sercet code* bersama.
   1. Jika data login benar, maka akan dikirimkan pesan berisi kode MSG\_SUCCESS (0x7f).

|  |
| --- |
| MSG\_SUCCESS (byte) |

* 1. Jika data login salah, maka akan dikirimkan pesan berisi kode MSG\_FAILED (0xff).

|  |
| --- |
| MSG\_FAILED (byte) |

# 

# Update

Update hanya bisa dilakukan setelah client berhasil melakukan login. Pada awalnya, client mengupdate status pada tugas yang dimilikinya. Lalu membuat catatan ke log. Jika sudah ada log dengan ID tugas yang sama, maka log dengan ID tersebut dihapus. Lalu akan dilakukan update ke server dengan mengirimkan message ke server.

|  |  |  |  |
| --- | --- | --- | --- |
| MSG\_UPDATE  (byte) | SESSION\_ID  (long) | LOG\_COUNT  (int) | LOG\_UPDATE\_LIST  (...) |

MSG\_UPDATE berupa kode 0x10.



# List

Client yang telah login dapat mendapatkan daftar tugas. Pada awalnya, client mengirimkan MSG\_LIST berupa kode 0x20 lalu client mengirimkan pula list tugas yang dimilikinya saat ini. Server menerima list tugas client lalu membuat query yang sesuai dengan username dari client. Bila saat dibandingkan timestamp tugas pada list user lebih kecil daripada timestamp tugas pada list query maka dikirimkan kode LIST\_STATUS\_UPD berupa kode 0x03 diikuti dengan tugasnya. Bila saat dibandingkan timestamp tugas pada list user sama dengan timestamp tugas pada list query maka dikirimkan kode LIST\_STATUS\_EQL berupa kode 0x01 diikuti dengan id tugas. Bila saat dibandingkan tugas pada list user tidak ada pada list query maka dikirimkan kode LIST\_STATUS\_DEL berupa kode 0x02 dan juga id tugas yang sudah di delete tersebut. Bila masih ada sisa pada list query maka dikirimkan kode LIST\_STATUS\_NEW berupa kode 0x00 diikuti dengan tugasnya. Setelah semua message query selesai dibandingkan dikirimkan kode LIST\_STATUS\_END berupa kode -1. Kemudian client menerima pesan-pesan dari server. Bila client menerima pesan LIST\_STATUS\_NEW dan tugasnya, maka client akan menambahkan tugas itu ke list client. Bila client menerima pesan LIST\_STATUS\_EQL dan id tugasnya, maka client tidak melakukan apapun. Bila client menerima pesan LIST\_STATUS\_ DEL dan id tugasnya, maka client menghapus tugas dengan id tersebut dari list tugas. Bila client menerima pesan LIST\_STATUS\_ UPD dan tugasnya, maka client melakukan update tugas dengan id tersebut di list tugas. Bila client menerima pesan LIST\_STATUS\_ END ,maka client melakukan list ke layar dari list tugas client.



# Logout

Client yang telah login dapat melakukan logout (0x01) dengan mengirimkan message berikut.

|  |  |
| --- | --- |
| MSG\_LOGOUT  (byte) | SESSION\_ID  (long) |

Lalu server akan membalas dengan MSG\_SUCCESS.

|  |
| --- |
| MSG\_SUCCESS (byte) |

