Dokumentasi Protokol

1 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)

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

p (UTF)	g (UTF)

3. 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)

4. 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)

5. 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.

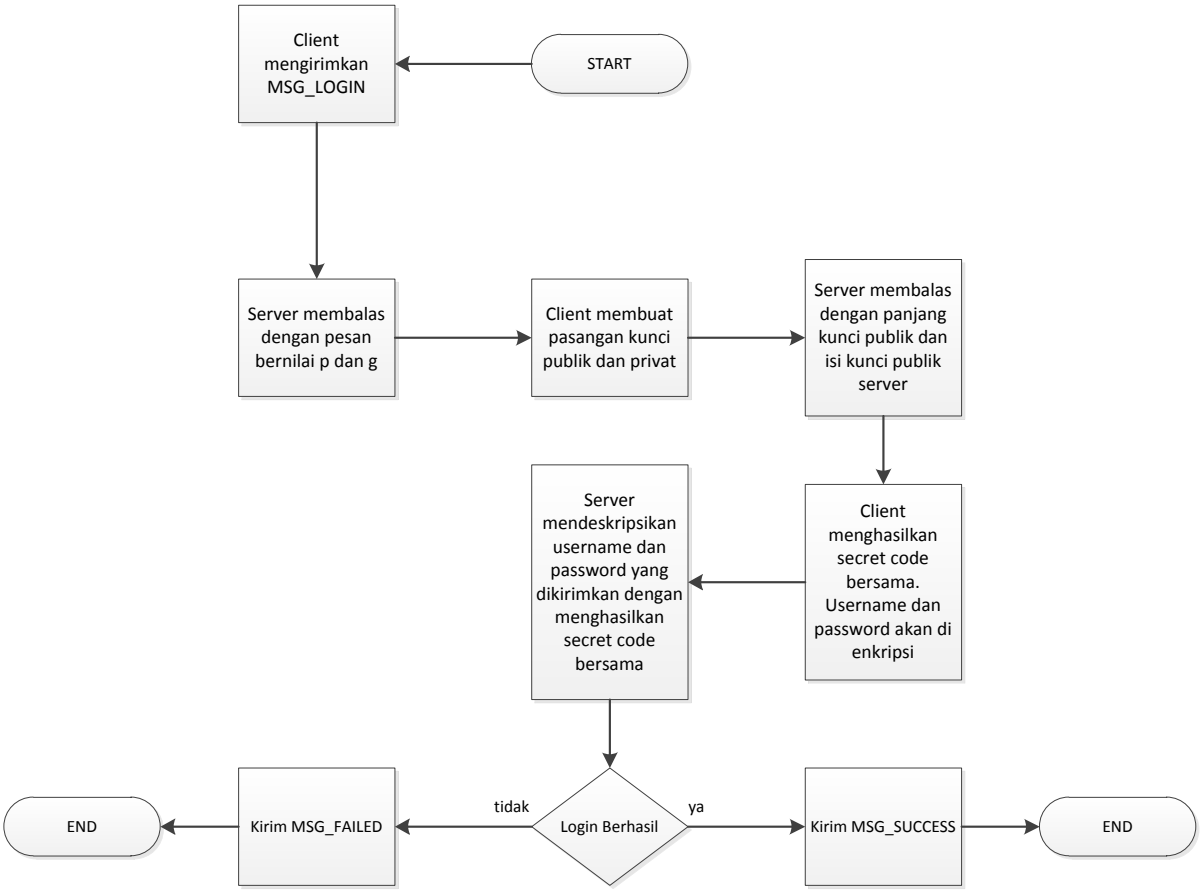
AES_UNAME_LENGHTH (int)	AES_UNAME (bytes)	AES_PASSWD_LENGTH (int)	AES_PASSWD (bytes)

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

MSG_SUCCESS (byte)

    b. Jika data login salah, maka akan dikirimkan pesan berisi kode MSG_FAILED (0xff).

MSG_FAILED (byte)

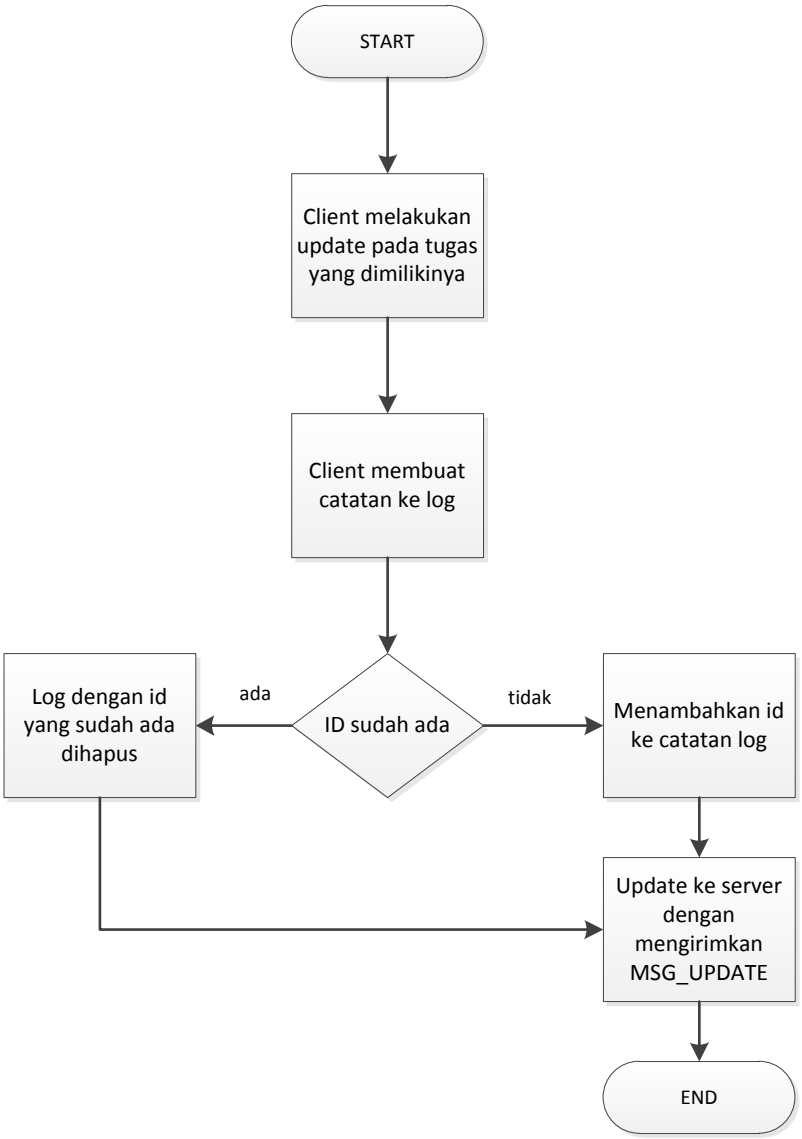

START
Client mengirimkan MSG_LOGIN
Server membalas dengan pesan bernilai p dan g
Client membuat pasangan kunci publik dan privat
Server membalas dengan panjang kunci publik dan isi kunci publik server
Client menghasilkan secret code bersama. Username dan password akan di enkripsi
Server mendeskripsikan username dan password yang dikirimkan dengan menghasilkan secret code bersama
Login Berhasil
tidak
ya
Kirim MSG_FAILED
Kirim MSG_SUCCESS
END
END

2 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 (...)

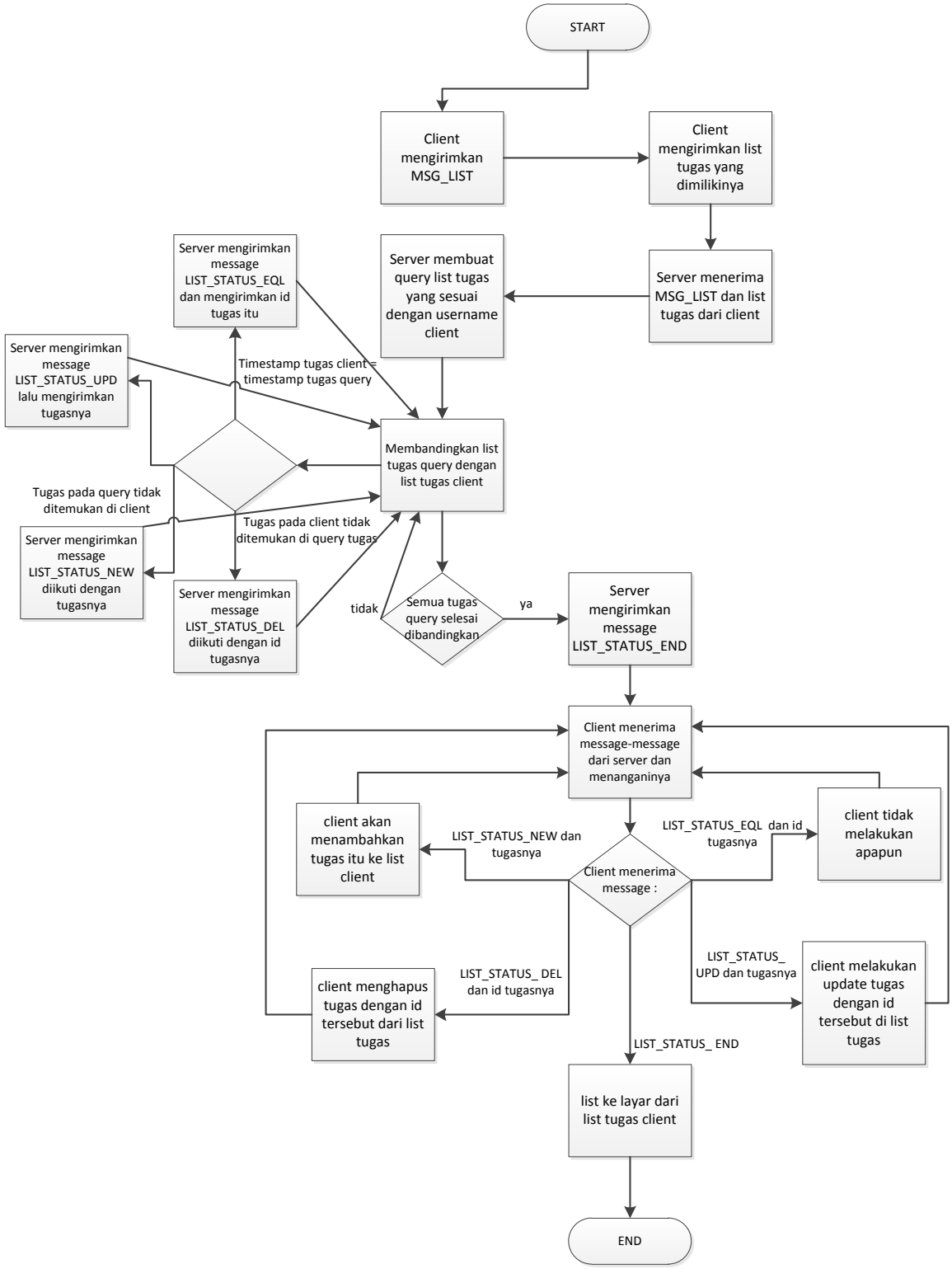
MSG_UPDATE berupa kode 0x10.



3 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.


START
Client mengirimkan MSG_LIST
Client mengirimkan list tugas yang dimilikinya
Server menerima MSG_LIST dan list tugas dari client
Server membuat query list tugas yang sesuai dengan username client
Server mengirimkan message LIST_STATUS_EQL dan mengirimkan id tugas itu
Server mengirimkan message LIST_STATUS_UPD lalu mengirimkan tugasnya
Timestamp tugas client = timestamp tugas query
Membandingkan list tugas query dengan list tugas client
Tugas pada query tidak ditemukan di client
Server mengirimkan message LIST_STATUS_NEW diikuti dengan tugasnya
Tugas pada client tidak ditemukan di query tugas
Server mengirimkan message LIST_STATUS_DEL diikuti dengan id tugasnya
tidak
Semua tugas query selesai dibandingkan
ya
Server mengirimkan message LIST_STATUS_END
Client menerima message-message dari server dan menanganinya
client akan menambahkan tugas itu ke list client
LIST_STATUS_NEW dan tugasnya
Client menerima message :
LIST_STATUS_EQL dan id tugasnya
client tidak melakukan apapun
LIST_STATUS_ UPD dan tugasnya
client melakukan update tugas dengan id tersebut di list tugas
LIST_STATUS_ DEL dan id tugasnya
client menghapus tugas dengan id tersebut dari list tugas
LIST_STATUS_ END
list ke layar dari list tugas client
END

4 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)