**Laporan Projek**

**Keamanan Komputasi**

**Chatroom with Encryption**

****

**Devi Maya**

**Ruben Ng**

**2015**

**Tangerang - Karawaci**

Program *messenger* yang kami buat adalah *multi-threaded,* berbasis *socket,* dan menggunakan enkripsi AES untuk pengirimnan pesan. Algoritma enkripsi AES diambil dari library java “javax.crypto.”

Program kami terdiri dari 4 class :

* ChatServer  
  Kelas ini berfungsi sebagai server. Pemanggilan kelas akan membentuk sebuah chatroom. Server tidak dapat melihat pesan-pesan yang dikirim oleh client – ia hanya berfungsi sebagai pengolah dan pengirim pesan, dan juga menerima dan mengatur client yang masuk ke chatroom.
* ServerThread  
  Thread ini akan dibuat setiap client terkoneksi ke server. Thread ini berfungsi untuk menerima input dari client dan mengirimkannya ke server.
* ChatClient  
  Kelas ini adalah client. Client dapat terkoneksi ke sebuah chatroom dengan memasukkan address dan port dari chatroom yang ada. Pesan yang dikirimkan dari client akan dienkripsi sebelum dikirim.
* ClientThread

Thread ini berada di class ChatClient. Thread ini berfungsi untuk menerima pesan yang dikirim oleh Server (misalnya pesan yang dikirim client lain).

* EDCrypt

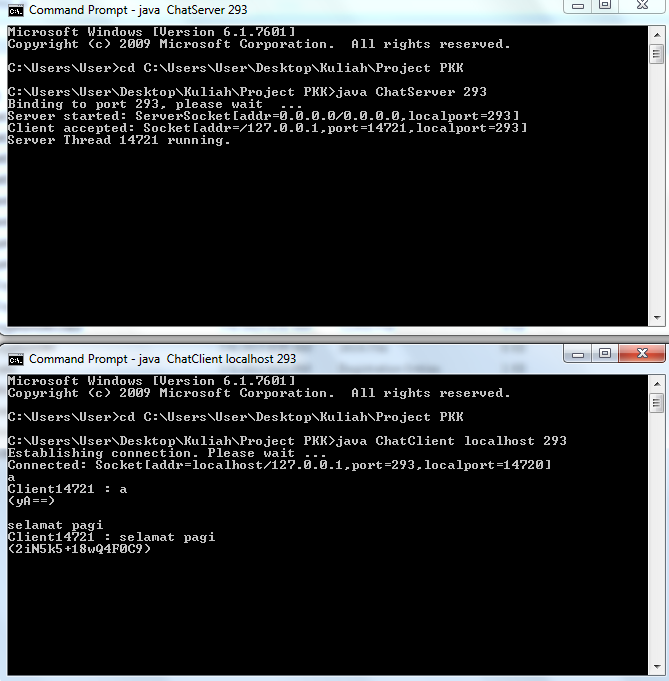
Kelas ini berisikan fungsi-fungsi untuk melakukan enkripsi dan dekripsi sebuah String. Key untuk melakukan enkripsi disimpan di kelas ini.

Notes :

* Server tidak dapat berkomunikasi langsung dengan chatroom
* Ada beberapa perintah yang dapat dipakai client:  
  “.bye” untuk keluar dari chatroom.  
  “.list” akan menampilkan list client di dalam chatroom.
* Program ini merupakan lanjutan dari projek yang sudah saya buat sebelumnya, jadi mohon maklum jika ada kesamaan dengan kelompok lain (Poppy, Merdiana, Aurellia).

Cara kerja program :

1. Execute kelas ChatServer dengan argument berupa integer untuk port yang akan digunakan.
2. Execute satu (atau lebih) ChatClient dengan argument berupa string untuk address dari chatroom, dan integer untuk port chatroom yang dituju.



Contoh *execute* program

**Pseudocode proses dalam mengolah pesan**

1. ChatClient mengetikkan pesan ke console.
2. Dalam loop *run* di kelas ChatClient, pesan akan dienkripsi, kemudian dikirimkan ke ServerThread.
3. ServerThread akan memanggil fungsi *handle* milik ChatServer.
4. Di dalam fungsi *handle,* Server akan mendekripsi pesan untuk mengecek apakah pesan tersebut merupakan perintah. Jika iya (contoh .bye), server akan menjalankan fungsi yang sesuai. Jika tidak, server akan mengirimkan pesan dan ID pengirim ke seluruh ClientThread yang terkoneksi dengannya.
5. Setiap ClientThread akan memanggil fungsi *handle* milik client dengan argumen berupa pesan (yang masih dalam bentuk ciphertext) dan ID pengirim.
6. Di fungsi *handle,* client akan mendekripsi pesan dan mengeluarkannya ke console.