PENERAPAN KRIPTOGRAFI DAN GRAF DALAM APLIKASI KONFIRMASI JARKOM

Mario Orlando Teng

(13510057)

Program Studi Teknik Informatika
Sekolah Teknik Elektro dan Informatika
Institut Teknologi Bandung,
Jl. Ganesha 10 Bandung 40132, Indonesia

mario.orlando@students.itb.ac.id

***Abstrak* – Makalah ini membahas tentang penggunaan kriptografi dan graf yang telah diajarkan dalam mata kuliah IF2091 Struktur Diskrit dalam membangun sebuah aplikasi untuk konfirmasi jarkom. Kriptografi digunakan untuk keamanan dalam penggunaan aplikasi ini, prinsipnya akan sama dengan penerapan kriptografi dalam email. Sedangkan pemahaman graf diperlukan baik dalam membangun sistem database jarkom maupun dalam menangani *deadlock*.**

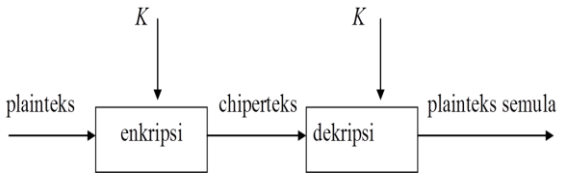
Kata kunci – jarkom, enkripsi, dekripsi, deadlock

I. PENDAHULUAN

I.I Kriptografi

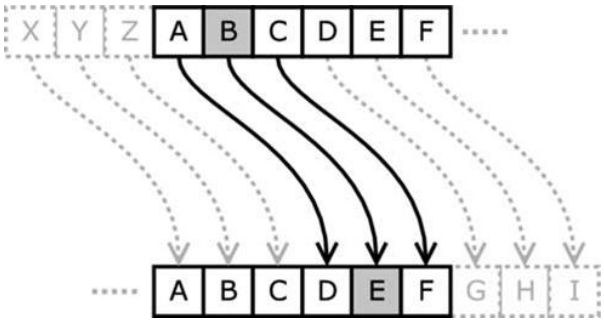
Kriptografi adalah ilmu yang digunakan untuk menyamarkan suatu pesan yang bersifat rahasia dari orang-orang yang tidak berhak untuk membaca pesan tersebut. Pesan yang sesungguhnya disamarkan menjadi suatu bentuk pesan yang tidak memiliki makna dan cara penyamarannya memiliki suatu pola tertentu (kunci penyandian) untuk setiap elemen pesan.

Pesan yang dirahasiakan dinamakan **plainteks**, sedangkan pesan hasil penyandian disebut **cipherteks**. Proses menyandikan plainteks menjadi cipherteks disebut **enkripsi**, sedangkan proses penyandian cipherteks menjadi plainteks kembali disebut **dekripsi**.



(*Proses enkripsi - dekripsi*)

Contoh pemakaian kriptografi yang cukup sederhana adalah Caesar Cipher. Prinsip Caesar Cipher adalah penggeseran kata sebanyak 3 huruf ke kanan.



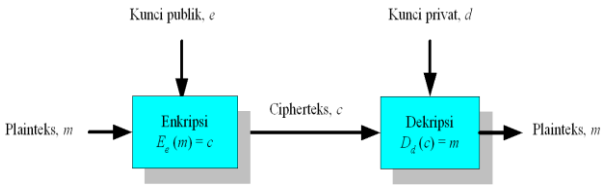
(*Caesar Cipher*)

Contoh penggunaan kriptografi yaitu dalam bentuk Caesar Cipher. Caesar Cipher adalah jika plainteks "AWASI ASTERIX" dienkripsi akan menjadi "DZDVL DVWHULA".

Pada kasus umum di atas, dapat dilihat bahwa kunci yang diperlukan untuk melakukan enkripsi dan kunci yang diperlukan untuk dekripsi adalah kunci yang sama sehingga keamanan penyandian tidak terjamin.

Untuk mengatasi hal tersebut maka diciptakan algoritma RSA yang memiliki pasangan kunci untuk setiap pengguna yaitu :

1. Kunci publik, e untuk enkripsi
2. Kunci privat, p untuk dekripsi (rahasia)



(*Proses enkripsi – dekripsi dalam algoritma RSA*)

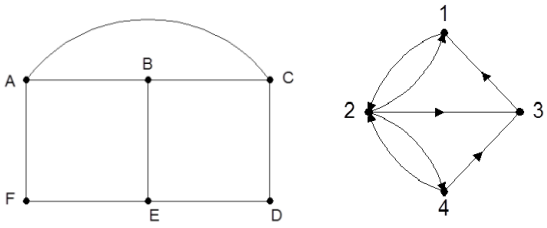
Algoritma pembangkitan pasangan kunci

1. Pilih dua bilangan prima, p dan q (rahasia)
2. Hitung $n = pq$. Besaran n tidak perlu dirahasiakan.
3. Hitung $m = (p - 1)(q - 1)$.
4. Pilih sebuah bilangan bulat untuk kunci publik, e, relatif prima terhadap m.
5. Hitung kunci dekripsi, d, melalui kekongruenan $ed \equiv 1$ (**mod** m).

I.II Graf

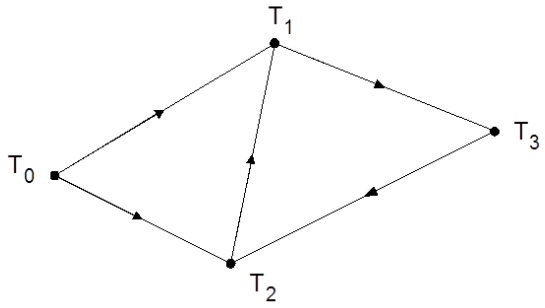
Graf G didefinisikan sebagai pasangan himpunan (V,E), ditulis dengan notasi G = (V,E) dimana V adalah himpunan tidak kosong dari simpul-simpul dan E adalah himpunan sisi yang menghubungkan 2 simpul.

Suatu graf dapat ditampilkan dengan representasi graf berarah dan tidak berarah. Graf berarah adalah graf yang setiap sisinya diberikan orientasi arah. Sedangkan graf tidak berarah adalah graf yang sisinya tidak mempunyai orientasi arah sehingga urutan pasangan simpul tidak diperhatikan, contohnya (u,v) = (v,u).



(*Contoh graf tidak berarah dan berarah*)

Pada pembahasan graf ini akan dikonsentrasikan mengenai deadlock. Deadlock adalah kondisi yang terjadi karena ada beberapa transaksi yang saling menunggu transaksi lainnya sehingga sistem akan berhenti bekerja. Kondisi deadlock dapat digambarkan berupa graf.



(*Deadlock*)

Pada contoh diatas dapat dilihat keadaan yang mencerminkan deadlock.

- Transaksi T_0 menunggu transaksi T_1 dan T_2.
- Transaksi T_2 menunggu transaksi T_1.
- Transaksi T_1 menunggu transaksi T_3.
- Transaksi T_3 menunggu transaksi T_2.

II. LATAR BELAKANG MASALAH

Pada zaman sekarang, perkembangan teknologi dan penggunaan telepon genggam atau handphone sangat memasyaraka. Pada suatu organisasi atau perkumpulan biasanya membutuhkan suatu sistem untuk memberitahukan kepada anggotanya tentang berbagai hal. Untuk kepentingan tersebut biasanya diberlakukan sistem jaringan komunikasi (jarkom).

Jarkom ini digunakan untuk menyampaikan informasi-informasi yang berkaitan dengan suatu acara. Pada beberapa kasus, ada jarkom yang membutuhkan konfirmasi dari orang-orang yang mendapatkan jarkom tersebut seperti menanyakan kepastian ikut acara tersebut, menanyakan waktu yang tepat untuk keberlangsungan suatu acara untuk keperluan survey, dan lain-lain.

Pada umumnya, mereka yang menerima jarkom yang membutuhkan konfirmasi ini tidak membalas untuk melakukan konfirmasi.

Dari pengamatan penulis, alasan mereka tidak membalas jarkom tersebut karena malas atau lupa untuk membalas setelah meninggalkan *handphone* untuk melakukan sesuatu selama beberapa saat .

Untuk mengatasi masalah-masalah inilah, penulis merancang suatu aplikasi konfirmasi jarkom menggunakan dasar pengetahuan yang telah didapat dari mata kuliah IF2092 – Struktur Diskrit yaitu kriptografi dan graf.

III. PENERAPAN GRAF DALAM APLIKASI

Prinsip dasar aplikasi ini adalah ketika seseorang mengirimkan jarkom pada orang lain, maka sistem handphone orang yang dijarkom tersebut akan mengalami deadlock hingga orang tersebut membalas jarkom dengan konfirmasi yang tepat.

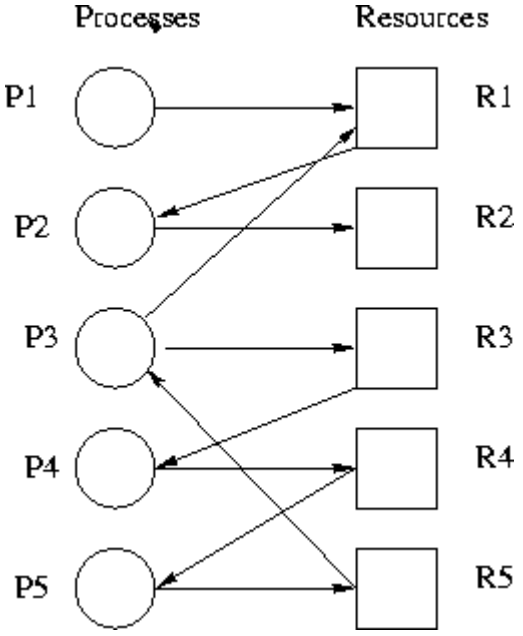
Ada 4 kondisi yang harus terpenuhi agar deadlock tersebut terjadi, yaitu :

1. **Mutual Exclusion**
   Hanya ada satu proses yang boleh memakai sumber daya, dan proses lain yang ingin memakai sumber daya tersebut harus menunggu hingga sumber daya tadi dilepaskan atau tidak ada proses yang memakai sumber daya tersebut.
2. **Hold and Wait**
   Proses yang sedang memakai sumber daya boleh meminta sumber daya lagi maksudnya menunggu hingga benar-benar sumber daya yang diminta tidak dipakai oleh proses lain. Hal ini dapat menyebabkan kelaparan sumber daya sebab dapat saja sebuah proses tidak mendapat sumber daya dalam waktu yang lama.
3. **No Preemption**
   Sumber daya yang ada pada sebuah proses tidak boleh diambil begitu saja oleh proses lainnya. Untuk mendapatkan sumber daya tersebut, maka harus dilepaskan terlebih dahulu oleh proses yang memegangnya, selain itu seluruh proses menunggu dan mempersilahkan hanya proses yang dimiliki sumber daya yang boleh berjalan.

4. **Circular Wait**
   Kondisi yang dapat digambarkan seperti rantai keterkaitan, yaitu sebuah proses membutuhkan sumber daya yang dipegang oleh proses yang akan berlangsung berikutnya.

Untuk mengatasi deadlock memang belum dibahas karena merupakan bagian dari kuliah Sistem Operasi dan Sistem Basis Data. Tetapi untuk memperjelas cara kerja aplikasi maka penulis akan menjelaskan cara menangani deadlock dengan penjelasan yang simpel.

Untuk menangani deadlock diperlukan cara untuk melakukan deteksi terlebih dahulu tentang bagian graf yang merupakan tempat terjadinya deadlock.



(*Contoh kasus deadlock yang lebih rumit*)

Pada contoh di atas penelusuran sirkuit deadlock dapat dilakukan dengan menggunakan variabel penampung L dan dimulai dari *node* P1.

- Tambahkan P1 ke L
- Telusuri sisi P1, didapatkan R1. Tambahkan R1 ke L.
- Telusuri sisi R1, didapatkan P2. Tambahkan P2 ke L.
- Telusuri sisi P2, didapatkan R2. Tambahkan R2 ke L.
- R2 tidak memiliki sisi, kembali ke P2.
- P2 tidak memiliki sisi lain, kembali ke R1.
- R1 tidak memiliki sisi lain, kembali ke P1.
- P1 adalah *starting point* sehingga kita menyimpulkan tidak terjadi deadlock dalam lintasan ini.

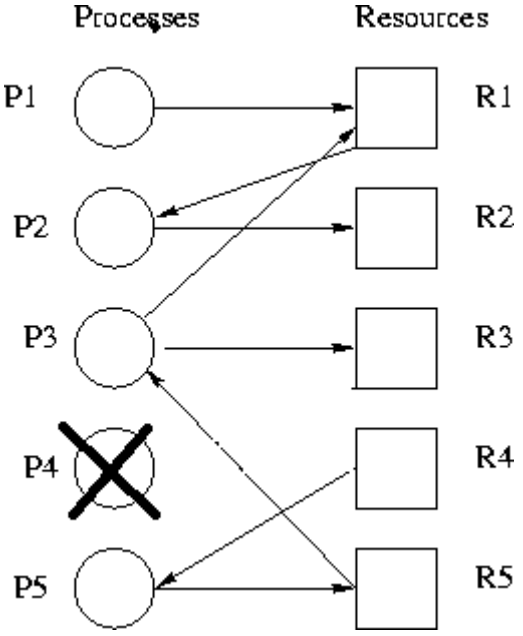
Setelah itu lakukan hal yang sama untuk *node* baru yaitu P3. Dari penelusuran sisi-sisinya maka akan didapatkan :

- P3 – R1 – P2 – R2 (tidak terjadi deadlock)
- P3 – R3 – P4 – R4 – P5 – R5 – P3 (deadlock)

Setelah penelusuran deadlock dilakukan maka dapat dilakukan penanganan deadlock. Ada 3 cara untuk menangani deadlock :

1. **Preemption**
   Prinsip yang digunakan cara ini adalah kembali ke titik awal dimana sumber daya yang telah dipakai dilepas dan diberikan kepada sumber lain.
   Kerugian yang didapat adalah misalnya ketika deadlock terjadi saat kondisi print maka sistem aan kembali ke kondisi semula dan print hanya berjalan sebagian.
2. **Rollback**
   Pada Rollback, sistem akan mencatat state dari setiap proses yang telah dilakukan dan ketika deadlock terjadi, sistem akan mengembalikan ke keadaan dimana deadlock belum terjadi (checkpoint).
   Kerugian cara ini adalah kita kehilangan setiap pekerjaan yang telah kita lakukan dimulai dari state akhir hingga checkpoint.
3. **Kill Process**
   Pada cara ini kita akan menghentikan sebuah proses demi keberlangsungan keseluruhan proses agar tidak terjadi deadlock.
   Kerugian yang didapat dari cara ini adalah adanya proses yang tidak dilakukan karena proses tersebut dihentikan saat terjadi deadlock.

Cara yang akan dipakai untuk menangani deadlcok dalam kasus ini adalah cara ketiga yaitu Kill Process dimana proses yang menyebabkan terjadinya deadlock tersebut adalah aplikasi konfirmasi jarkom sendiri.



(*Penanganan Deadlock dengan Kill Process P4*)

Saat terjadi deadlock, kondisi handphone akan dibuat sedemikian rupa sehingga tidak akan dibuat hang sepenuhnya. Deadlock akan menyebabkan sistem hanya berputar di bagian “pesan” yang mengharuskan pengguna untuk membalas konfirmasi jarkom dengan format tertentu (misal Y / N) agar penanganan deadlock dengan Kill Process terjadi.

Penjelasan mengenai pembuatan deadlock agar kondisi *handphone* tidak hang sepenuhnya masih belum bisa

dijelaskan karena penulis belum menemukan bahan yang tepat untuk itu. Selain itu, penulis masih dalam tahap belajar dan belum pernah melakukan oprek secara langsung terhadap deadlock *handphone*.

IV. PENERAPAN KRIPTOGRAFI DALAM APLIKASI

Aplikasi konfirmasi jarkom ini tentu akan berbahaya jika disalahgunakan orang lain yang iseng sehingga mengharuskan orang yang dikirim pesan harus membalas agar handphonenya terlepas dari kondisi deadlock.

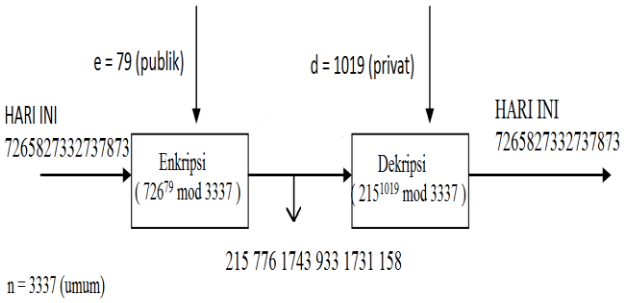
Untuk keamanan apikasi ini maka akan diterapkan kriptografi yaitu algoritma RSA. Prinsip kerjanya akan sama dengan prinsip kerja email dimana setiap orang akan mempunyai kunci publik berupa ID yang dapat disebar ke orang lain, tetapi untuk mengakses sistem di dalamnya diperlukan kunci privat berupa password yang bersifat rahasia.

Contoh penggunaan algoritma RSA dalam kasus yang lebih simpel yaitu dalam proses enkripsi dan dekripsi suatu plainteks menjadi cipherteks dapat dilihat dari contoh di bawah ini.

- Plainteks : "HARI INI"
- Jadikan dalam ASCII : 7265827332737873
- Pecah menjadi blok-blok kecil (3 digit)
  $p_1 = 726$ $p_4 = 273$
  $p_2 = 582$ $p_5 = 787$
  $p_3 = 733$ $p_6 = 003$
- Pilih a = 47, b = 71 (rahasia)
- n = ab = 3337 (umum)
- m = (a-1)(b-1) = 3220 (rahasia)
- Pilih kunci publik e = 79 (umum)
- Kunci dekripsi d = 1019 (rahasia)
  Dari $ed \equiv 1$ (**mod** m)

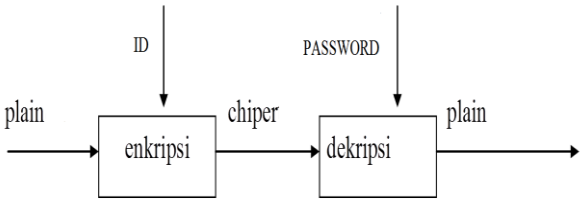
- Untuk proses enkripsi
  $c_1 = 726^{79} \bmod 3337 = 215$
  $c_2 = 582^{79} \bmod 3337 = 776$
  dst untuk sisa blok lainnya
  Diperoleh cipherteks C = 215 776 1743 933 1731 158.

- Untuk proses dekripsi
  $p_1 = 215^{1019} \bmod 3337 = 726$
  $p_2 = 776^{1019} \bmod 3337 = 582$
  dst untuk sisi blok lainnya
  Diperoleh plainteks = 7265827332737873 / HARI INI



(*Dekripsi dan enkripsi teks simpel*)

Pada proses pembentukan ID dan password juga dapat dilakukan hal serupa dengan pengenkripsian dan pendekripsian yang lebih kompleks. Nilai e akan berguna sebagai ID pengguna dan nilai d akan berguna sebagai password.



(*Dekripsi dan enkripsi menggunakan ID dan password*)

Pada proses dekripsi dan enkripsi menggunakan ID dan password, yang menjadi bahan enkripsi dan dekripsi bukanlah teks lagi melainkan sebuah sistem yang nantinya akan digunakan secara bersama. Dalam hal ini sistem yang akan digunakan secara bersama tersebut adalah aplikasi konfirmasi jarkom.

Sehingga dapat disimpulkan bahwa untuk menggunakan aplikasi ini, orang yang mengirimkan jarkom dan orang yang menerima jarkom harus sepakat untuk memakai aplikasi ini. Selain itu mereka harus bertukar ID aplikasi karena aplikasi ini menerapkan sistem keamanan dengan kriptografi RSA.

V. KESIMPULAN

Saat ini kemajuan teknologi telah berkembang dengan pesat dan seiring kemajuan tersebut, muncul berbagai hal baru yang tidak terduga sebelumnya.

Hal yang muncul tersebut kebanyakan berupa masalah-masalah yang dapat dipecahkan dengan suatu bentuk pemikiran yang kreatif dan inovatif.

Contohnya adalah kasus jarkom ini dimana banyak sekali orang tidak melakukan konfirmasi saat diminta.

Untuk mengatasi hal ini penulis mencoba merancang aplikasi konfirmasi jarkom yang menggunakan materi kuliah Struktur Diskrit.

Materi yang digunakan adalah graf khususnya di bidang deadlock untuk memaksa pengguna melakukan konfirmasi dari jarkoom dan kriptografi untuk menjaga keamanan aplikasi agar tidak disalahgunakan oleh orang lain yang ingin berbuat jahat.

Dengan pemanfaatan graf dan kriptografi maka kita dapat mempermudah konfirmasi dari suatu jaringan komunikasi.

VI. DAFTAR PUSTAKA

[1] Munir, Rinaldi, Matematika Diskrit. Ed.3, Bandung : Informatika Bandung, 2007

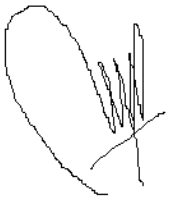
[2] http://wahyuadam.web.ugm.ac.id/?p=57
Waktu akses : 10 Desember 2011 Pukul 00.15

[3] http://www.cs.rpi.edu/academics/courses/fall04/os/c10/index.html
Waktu akses : 10 Desember 2011 Pukul 00.26

PERNYATAAN

Dengan ini saya menyatakan bahwa makalah yang saya tulis ini adalah tulisan saya sendiri, bukan saduran, atau terjemahan dari makalah orang lain, dan bukan plagiasi.

Bandung, 10 Desember 2011



Mario Orlando Teng
13510057