GENERALISASI ALGORITMA KRIPTOGRAFI ELGAMAL ATAS GRUP PERGANDAAN MODULO POLINOMIAL *IRREDUCIBLE* DALAM PENGAMANAN PESAN RAHASIA

SKRIPSI

Untuk memenuhi sebagian persyaratan

Mencapai derajat Sarjana S-1

Diajukan oleh:

NajibMubarok

08610028

Kepada

PROGRAM STUDI MATEMATIKA

FAKULTAS SAINS DAN TEKNOLOGI

UIN SUNAN KALIJAGA

YOGYAKARTA

2013

SURAT PERSETUJUAN SKRIPSI/TUGAS AKHIR

Hal : Persetujuan Skripsi
Lamp : -

Kepada
Yth. Dekan Fakultas Sains dan Teknologi
UIN Sunan Kalijaga Yogyakarta
di Yogyakarta

Assalamu'alaikum wr. wb.

Setelah membaca, meneliti, memberikan petunjuk dan mengoreksi serta mengadakan perbaikan seperlunya, maka kami selaku pembimbing berpendapat bahwa skripsi Saudara:

Nama	: Najib Mubarok
NIM	: 08610028
Judul Skripsi	: Generalisasi Algoritma Kriptografi ElGamal atas Grup Pergandaan Modulo Polinomial *Irreducible* dalam Pengamanan Pesan Rahasia

sudah dapat diajukan kembali kepada Program Studi Matematika Fakultas Sains dan **Teknologi** UIN Sunan Kalijaga Yogyakarta sebagai salah satu syarat untuk memperoleh gelar Sarjana Strata Satu dalam bidang Matematika.

Dengan ini saya mengharap agar skripsi/tugas akhir Saudara tersebut di atas dapat segera dimunaqsyahkan. Atas perhatiannya saya ucapkan terima kasih.

Wassalamu'alaikum wr. Wb

Yogyakarta, 01 Februari 2013

Pembimbing I	Pembimbing II
Muhammad Zaki Riyanto, S.Si., M.Sc.	Malahayati, S.Si., M.Sc.
NIDN. 0513018402	NIP. 19840412 201101 2 010

Universitas Islam Negeri Sunan Kalijaga FM-UINSK-BM-05-07/R0

PENGESAHAN SKRIPSI/TUGAS AKHIR

Nomor : UIN.02/D.ST/PP.01.1/729/2013

Skripsi/Tugas Akhir dengan judul : Generalisasi Algoritma Kriptografi ElGamal atas Grup Pergandaan Modulo Polinomial *Irreducible* dalam Pengamanan Pesan Rahasia

Yang dipersiapkan dan disusun oleh :
Nama : Najib Mubarok
NIM : 08610028
Telah dimunaqasyahkan pada : 21 Februari 2013
Nilai Munaqasyah : A
Dan dinyatakan telah diterima oleh Fakultas Sains dan Teknologi UIN Sunan Kalijaga

TIM MUNAQASYAH :

Ketua Sidang

M. Zaki Riyanto, M.Sc
NIP. 0513018402

Penguji I

Dra.Khurul Wardati, M.Si.
NIP.19660731 200003 2 001

Penguji II

Malahayati, M.Sc
NIP.19840412 201101 2 010

Yogyakarta, 06 Maret 2013
UIN Sunan Kalijaga
Fakultas Sains dan Teknologi
Dekan

KEMENTERIAN AGAMA FAKULTAS SAINS DAN TEKNOLOGI UIN SUNAN KALIJAGA YOGYAKARTA

Prof. Drs. H. Akh. Minhaji, M.A, Ph.D
NIP. 19580919 198603 1 002

SURAT PERNYATAAN KEASLIAN

Yang bertanda tangan di bawah ini :

Nama : Najib Mubarok

NIM : 08610028

Program Studi : Matematika

Fakultas : Sains dan Teknologi

Menyatakan dengan sesungguhnya bahwa skripsi ini merupakan hasil pekerjaan penulis sendiri dan sepanjang pengetahuan penulis tidak berisi materi yang dipublikasikan atau ditulis orang lain, dan atau telah digunakan sebagai persyaratan penyelesaian Tugas Akhir di Perguruan Tinggi lain, kecuali bagian tertentu yang penulis ambil sebagai bahan acuan. Apabila terbukti pernyataan ini tidak benar, sepenuhnya menjadi tanggung jawab penulis.

Yogyakarta, 01 Februari 2013

Yang menyatakan

Najib Mubarok

NIM. 08610028

HALAMAN PERSEMBAHAN

Alhamdulillaahi robbil 'alamiin..

Karya kecil ini penulis persembahkan kepada Bapak dan Ibu tercinta, yang menyayangi penulis sejak kecil,selalu memprioritaskan pendidikan, mendoakan penulis, serta memberikan dukungan atas segala yang penulis lakukan.

Juga kepada seluruh kerabat, kakak-kakak penulis, "adik" penulis, semua dosen, ustadz, dan teman-teman yang selalu memberikan dukungan kepada penulis.

Serta kepada almamater tercinta program studi Matematika fakultas Sains dan Teknologi UIN Sunan Kalijaga Yogyakarta.

HALAMAN MOTTO

"Tak semua yang dihitung dapat diperhitungkan, dan tak semua yang diperhitungkan dapat dihitung."

(Albert Einstein)

"Hidup bukan tentang apa yang bisa kita dapatkan, melainkan apa yang bisa kita berikan."

(Najib Mubarok)

KATA PENGANTAR

Assalamu'alaikum Wr. Wb.

Segala puji bagi Allah SWT yang telah memberikan rahmat, taufik, dan hidayah-Nya, serta nafas kepada penulis sampai detik ini, sehingga penulis mampu meyelesaikan penulisan skripsi berjudul *"Generalisasi Algoritma Kriptografi ElGamal atas Grup Pergandaan Modulo Polinomial Irreducible"*dengan semaksimal mungkin. Sholawat dan salam semoga senantiasa terlimpahkan kepada Nabi Muhammad SAW yang telah membawa umat manusia menuju zaman yang terang benderang dengan kemajuan ilmu pengetahuan dan teknologi.

Penulis menyadari bahwa proses penulisan skripsi ini tidak terlepas dari dukungan, kerjasama dan bimbingan dari berbagai pihak. Oleh karena itu, iringan doa dan terimakasih penulis sampaikan dengan tulus kepada:

1) Prof. Drs. H. Akh. Minhaji., Ph.D selaku Dekan Fakultas Sains dan Teknologi Universitas Islam Negeri Sunan Kalijaga Yogyakarta.
2) Muchammad Abrori, S.Si., M.Kom. selaku ketua program studi Matematika.
3) M. Zaki Riyanto, S.Si., M.Sc. selaku pembimbing pertama, yang telah memberikan ilmu, arahan serta dukungan sehingga penulisan skripsi ini dapat terselesaikan.

4) Malahayati, S.Si., M.Sc. selaku pembimbing kedua, yang telah memberikan ilmu, arahan, serta dukungan dalam penulisan skripsi ini.
5) Ayah dan ibu tercinta yang selalu memberikan dukungan, doa dan kasih sayang, serta selalu memprioritaskan pendidikan penulis.
6) Semua guru, dosen dan ustadz atas arahan dan ilmu yang telah diberikan, serta bimbingan kepada penulis untuk menjadi manusia yang lebih baik.
7) Kakak-kakak penulis, yaitu Adib Aufal Marom, Achmad Nur Afnan, dan Luluk Ifadah yang selalu memberikan dukungan, serta kasih sayangnya kepada penulis.
8) Keluarga besar Bani Washito, Muhtarom, Muhtadi, Husain, Ainul, Muna, Putri, iqbal, Tata, Emil, dan semua kerabat dekat penulis yang tidak bisa disebut satu per satu.
9) Tidak lupa kepada "adik" spesial penulis, Nabeela Fanny Aditya yang selalu memberikan motivasi untuk menjadi lebih baik dan memberikan dorongan semangat tiada henti.
10) Teman-teman Matematika 2008, Santosa, Bayu, Ranto, Imron, Tatar, Adib, Ibul, Aris, Okta, Ial, Bowo, Riyanto, Nana, Ria, Reni, Aesa, Naifi, Zeni, Tuti, Ifti, Septa, serta teman-teman yang tidak bisa penulis sebutkan satu per satu, yang senantiasa menjadi teman belajar, serta menjadi keluarga penulis di kampus.
11) Teman teman PP Wahid Hasyim yang telah menjadi keluarga penulis di Yogyakarta.

12) Rekan-rekan PSPB (Pusat Studi dan Pengembangan Bahasa) dan WhEs Club (Wahid Hasyim *English Speaking Club*) yang telah menjadi rekan kerja dan teman belajar bahasa asing.
13) Teman-teman Ma'had Aly semester III Pondok Pesantren Wahid Hasyim yang telah menjadi teman belajar dan mengkaji ilmu agama.

Wassalaamu'alaikum Wr. Wb.

Yogyakarta, 27 Januari 2013

Penulis

DAFTAR ISI

DAFTAR GAMBAR

DAFTAR TABEL

DAFTAR LAMPIRAN

ARTI LAMBANG

$(G,+)$	: grup G atas operasi penjumlahan "$+$".
$b=-a$	: b adalah invers penjumlahan dari a.
$b=a^{-1}.$	: b adalah invers perkalian dari a.
$\mathbb{Z}$	: himpunan bilangan bulat.
$\mathbb{Q}$	: himpunan bilangan rasional.
$\|G\|$	: banyaknya anggota himpunan G.
$\mathbb{Z}_p$	: himpunan bilangan bulat modulo prima.
$\mathbb{Z}_P^*$	: grup pergandaan modulo prima.
$gcd(m,p)$	: pembagi persekutuan terbesar dari m dan p.
$a \pmod n$	: sisa pembagian dari a oleh p.
$\sum_{i=0}^{n} a_i x^i$	: polinomial dengan bentuk $a_n x^n + ... + a_2 x^2 + a_1 x + a_0$.
$\deg g(x)$	: derajat polinomial $g(x)$.
$F[x]$	: himpunan polinomial atas lapangan F.
$x \leftarrow a$	: nilai a dimasukkan ke x.
$f(x) \mid g(x)$	: $f(x)$ membagi $g(x)$.
$g(x) \operatorname{div} h(x).$	: hasil pembagian *(quotient)* dari $g(x)$ oleh $h(x)$.
$g(x) \bmod h(x)$	: sisa pembagian *(remainder)* dari $g(x)$ oleh $h(x)$.
$gcd[\,a(x), b(x)\,]$	: pembagi persekutuan terbesar polinomial $a(x)$ dan $b(x)$.

$F[x]/(f(x))$: himpunan sisa pembagian polinomial-polinomial atas lapangan *F* oleh polinomial $f(x)$ di $F[x]$.

$\mathbb{Z}_p[x]/(f(x))$: himpunan sisa pembagian polinomial-polinomial di $\mathbb{Z}_p[x]$ oleh polinomial $f(x)$ di $\mathbb{Z}_p[x]$.

$\mathbb{F}_q^*$: grup unit atas lapangan Lapangan berhingga $\mathbb{F}_q$.

$\mathbb{F}_{p^m}$: lapangan berhingga dengan order p^m yang dibentuk dari himpunan $\mathbb{Z}_p[x]/(f(x))$.

$\mathbb{F}_{p^m}^*$: grup pergandaan atas operasi modulo polinomial *irreducible* dengan order $p^m - 1$, $\mathbb{F}_{p^m}^* = \mathbb{Z}_p[x]/(f(x))^*$.

$\varphi(n)$: banyaknya bilangan bulat positif yang relatif prima dengan *n.*

$GF(2^m)$_ : *Galois field* dengan order 2^m.

$\mathbb{F}_{2^m}^*$: grup pergandaan yang dibentuk dari lapangan berhingga dengan karakteristik dua atas operasi modulo polinomial *irreducible* $f(x) \in \mathbb{Z}_2[x]$ dengan derajat *m.*

$^{\alpha}log\beta$: logaritma diskrit dari β dengan basis α.

$(f(x), \alpha, \beta)$: kunci publik algoritma kriptografi elgamal atas grup pergandaan $\mathbb{F}_{2^m}^*$.

X_i : blok-blok plainteks $X_1, X_2, X_3 \ \ldots\ , X_n$ dengan *n* adalah panjang pesan.

(γ_i, δ_i) : Blok-blok cipherteks.

ABSTRAK

GENERALISASI ALOGORITMA KRIPTOGRAFI ELGAMAL ATAS GRUP PERGANDAAN MODULO POLINOMIAL *IRREDUCIBLE* DALAM PENGAMANAN PESAN RAHASIA

Algoritma Kriptografi ElGamal merupakan salah satu algoritma kunci publik yang dikenalkan pertama kali oleh ilmuan Mesir bernama Taher ElGamal pada tahun 1985 M. Tingkat keamanan algoritma ini didasarkan atas masalah logaritma diskrit *(discrete logarithm problem)* terhadap suatu grup siklik tertentu. Grup siklik yang digunakan harus dipilih dengan hati-hati agar memenuhi dua syarat yaitu efisiensi *(efficiency)* dan keamanan *(security),* sehingga grup siklik mudah diaplikasikan dan masalah logaritma diskrit sulit dihitung.

Pengamanan pesan rahasia menggunakan algoritma kriptografi ElGamal terdiri dari tiga proses. Pertama adalah pembangkitan kunci *(key generation)*, yaitu membuat kunci publik *(public key)* dan kunci rahasia *(private key)*. Kedua adalah enkripsi yang merupakan pemetaan *(mapping)* dari pesan asli *(plaintext)* menjadi kode yang tidak bisa dibaca *(chipertext).* Proses kedua ini dilakukan dengan menggunakan kunci publik. Proses ketiga adalah dekripsi yang merupakan proses merubah *ciphertext* menjadi pesan asli dengan menggunakan kunci rahasia.

Algoritma kriptografi ElGamal secara khas *(classical)* bekerja atas grup pergandaan bilangan bulat modulo prima $\mathbb{Z}_P^*$. Namun sebenarnya, algoritma ini dapat digeneralisasi untuk bekerja pada sebarang grup siklik berhingga. Hal ini dikarenakan masalah logaritma diskrit yang dapat digeneralisasi pada sebarang grup siklik berhingga. Grup siklik yang dipilih dalam pengamanan pesan rahasia pada skripsi ini adalah grup pergandaan modulo polinomial *irreducible* yang dinotasikan dengan $\mathbb{F}_{2^m}^*$ yang merupakan grup siklik berhingga yang dibentuk dari lapangan berhingga $\mathbb{F}_{2^m}$ dengan karakteristik dua. Grup pergandaan ini mendapatkan perhatian khusus dalam kriptografi, sebab operasi aritmatika dalam grup pergandaan ini dapat dilakukan secara efisien baik dalam *hardware* maupun *software*.

Kata kunci : kriptografi, ElGamal, pesan rahasia, polinomial *irreducible*, kunci publik, lapangan berhingga, masalah logaritma diskrit.

BAB I

PENDAHULUAN

1.1.Latar Belakang

Masalah keamanan dan kerahasiaan merupakan salah satu aspek yang sangat penting dari suatu data atau informasi. Perkembangan teknologi informasi dalam bidang komunikasi elektronik yang sangat pesat memberikan kemudahan kepada manusia untuk saling berkomunikasi dengan mengirim dan menerima suatu data melalui jalur komunikasi elektronik, baik data biasa maupun data yang bersifat rahasia. Di sisi lain, jalur komunikasi elektronik seperti jaringan internet dan telepon merupakan jalur yang tidak aman *(insecure channel)*, sehingga kemudahan ini juga dapat dimanfaatkan oleh pihak-pihak yang tidak berwenang *(unauthorized)* untuk menyadap data tersebut. Penyadapan data ini akan berdampak sangat berbahaya jika data yang disadap merupakan data rahasia, misalnya data-data rahasia kemiliteran, perbankan, pemerintahan ataupun data-data rahasia lainnya. Di sinilah peran kriptografi sebagai sebuah studi matematika yang berhubungan dengan aspek-aspek keamanan data atau informasi untuk menawarkan sebuah solusi dalam pengamanan data rahasia yang dikirim pada suatu jalur yang tidak aman seperti jaringan internet ataupun telepon (Menezes, Orschot and Vanstone, 1996: 4).

Berdasarkan kunci yang digunakan, algoritma kriptografi dapat dibedakan menjadi dua, yaitu kriptografi kunci simetri *(simmetric-key cryptography)* dan

kriptografi kunci asimetri *(asimmetric-key cryptography)* yang sering disebut dengan algoritma kunci publik *(public key cryptography)* (Buchmann, 2000 :71).

Contoh dari kriptografi kunci simetri adalah AES *(Advance Encryption Standard)* dan DES *(Data Encryption Standard)*, sedangkan contoh dari kriptografi kunci publik adalah algoritma kriptografi RSA, Mc-Alice, dan ElGamal. Dalam skripsi ini akan dibahas tentang salah satu dari algoritma kunci publik yaitu Algoritma Kriptografi ElGamal.

Keamanan dari Algoritma Kriptografi ElGamal terletak pada sulitnya permasalahan mencari nilai logaritma diskrit *(intractability of Discrete Logaritm Problem)* atas suatu grup siklik (Menezes, Orschot and Vanstone, 1996: 294). Pada tahun 2007, M. Zaki Riyanto menulis sebuah skripsi yang berjudul "Pengamanan Pesan Rahasia Menggunakan Algoritma Kriptografi Elgamal atas Grup Pergandaan $\mathbb{Z}_P^*$", skripsi ini megkaji mengenai Algoritma Kriptografi ElGamal *(Classical ElGamal Algorithm)* atas grup pergandaan bilangan prima $\mathbb{Z}_P^*$. Algoritma kriptografi Elgamal secara khas *(typical)* bekerja atas grup pergandaan bilangan bulat modulo prima $\mathbb{Z}_P^*$, namun sebenarnya, algoritma ini dapat digeneralisasi untuk bekerja pada sebarang grup siklik berhingga (Menezes, Orschot and Vanstone, 1996).

Skripsi ini akan membahas mengenai generalisasi dari algoritma kriptografi ElGamal *(Generalized ElGamal Algorithm)* yang bekerja atas grup pergandaan modulo polinomial *irreducible*, grup ini dinotasikan dengan $\mathbb{F}_{2^m}^*$ yang merupakan grup siklik berhingga yang dibentuk dari lapangan berhingga dengan karakteristik dua. Grup pergandaan ini mendapatkan perhatian khusus dalam

kriptografi, salah satu sebabnya adalah operasi aritmatik dalam grup pergandaan ini yang bisa dilakukan secara efisien baik dalam *hardware* maupun *software* (Menezes, Orschot and Vanstone, 1996: 154).

1.2. Batasan Masalah

Pembahasan dalam skripsi ini hanya akan difokuskan pada konsep matematis yang melandasi algoritma kriptografi ElGamal, proses penyandian pesan atas grup pergandaan modulo polinomial *irreducible* yang dibentuk dari lapangan berhingga dengan karakteristik dua, dan implementasinya dengan menggunakan program komputer sederhana untuk memudahkan proses perhitungan.

1.3. Rumusan Masalah

Berdasarkan latar belakang masalah yang telah dipaparkan, maka dirumuskan permasalahan-permasalahan sebagai berikut :

1) Bagaimana konsep-konsep matematis yang melandasi algoritma kriptografi ElGamal atas grup pergandaan modulo polinomial *irreducible* yang dibentuk dari lapangan berhingga dengan karakteristik dua?
2) Bagaimana proses pengamanan pesan rahasia menggunakan Algoritma kriptografi ElGamal atas grup pergandaan modulo polinomial *irreducible* yang dibentuk dari lapangan berhingga dengan karakteristik dua?

3) Bagaimana implementasi algoritma kriptografi ElGamal atas grup pergandaan modulo polinomial *irreducible* yang dibentuk dari lapangan berhingga dengan karakteristik dua menggunakan program komputer MATLAB versi 7.1?

1.4. Tujuan Penelitian

Berdasarkan rumusan masalah yang telah disebutkan, tujuan dari penulisan skripsi ini adalah sebagai berikut :

1) Mengkaji konsep matematis yang melandasi algoritma kriptografi ElGamal atas grup pergandaan modulo polinomial *irreducible* yang dibentuk dari lapangan berhingga dengan karakteristik dua.
2) Mengkaji proses pengamanan pesan rahasia dengan algoritma kriptografi ElGamal atas grup pergandaan modulo polinomial *irreducible* yang dibentuk dari lapangan berhingga dengan karakteristik dua.
3) Mengimplementasikan algoritma kriptografi ElGamal grup pergandaan modulo polinomial *irreducible* yang dibentuk dari lapangan berhingga dengan karakteristik dua dalam sebuah program MATLAB 7.1 yang ditujukan untuk mempermudah perhitungan.

1.5. Manfaat Penelitian

Hasil dari penulisan skripsi ini diharapkan dapat memberikan manfaat-manfaat sebagai berikut :

1) Memberikan kontribusi dalam kajian aljabar dan kriptografi tentang landasan matematis yang melandasi algoritma kriptografi ElGamal.
2) Memberikan kontribusi dalam kajian kriptografi mengenai salah satu algoritma kriptografi kunci publik yaitu algoritma kriptografi ElGamal
3) Memberikan kontribusi dalam kajian kriptografi tentang algoritma kriptografi ElGamal yang digeneralisasi pada grup pergandaan modulo polinomial *irreducible* yang dibentuk dari lapangan berhingga dengan karakteristik dua.
4) Sebagai dasar untuk penelitian selanjutnya dalam dunia aljabar dan kriptografi.

1.6. Tinjauan Pustaka

Tinjauan pustaka dari skripsi ini adalah skripsi yang ditulis oleh M. Zaki Riyanto pada tahun 2007 yang berjudul "Pengamanan Pesan Rahasia Mengguanakan Algoritma Kriptografi Elgamal atas grup pergandaan $\mathbb{Z}_P^*$". Dalam skripsi tersebut, dijelaskan mengenai algoritma kriptografi ElGamal klasik *(Classical ElGamal Algorithm)* yang meliputi konsep matematis yang melandasi algoritma kriptografi ElGamal atas grup pergandaan bilangan bulat $\mathbb{Z}_P^*$, proses

penyandian, dan juga implementasi dengan menggunakan program komputer sederhana dengan bahasa Pascal.

Selain itu, digunakan juga beberapa buku sebagai referensi utama diantaranya *Handbook Of Applied Cryptography*, karangan A. Menezes, P. van Oorschot and S. Vanstone yang diterbitkan pada tahun 1996, di dalam buku ini dijelaskan bahwa algoritma kriptografi ElGamal dapat digeneralisasi pada sebarang grup siklik berhingga, selain itu juga dijelaskan beberapa definisi dan teorema tentang dasar struktur aljabar yang melendasi beberapa algoritma kriptografi. Buku-buku lain yang digunakan sebagai referensi antara lain *Introduction to Cryptography* karangan Johannes A. Buchman tahun 2000, *Understanding Cryptography* karangan Christof Paar dan Jan Pelzl tahun 2009, *Cryptography and Network Security Principles and Practices* karangan William Stallings tahun 2003 dan juga buku-buku lain yang diperlukan, baik buku tentang aljabar maupun kriptografi. Selain referensi dari beberapa buku, digunakan juga referensi-referensi lain yang bersumber dari internet, artikel bebas, dan jurnal.

Skripsi ini akan membahas mengenai generalisasi dari algoritma kriptografi ElGamal *(Generalized ElGamal Algorithm)* yang akan menjelaskan bahwa algoritma kriptografi ElGamal dapat digeneralisasi untuk bekerja pada sebarang grup siklik berhingga. Grup yang digunakan dalam skripsi ini adalah grup pergandaan modulo polinomial *irreducible* yang dinotasikan dengan $\mathbb{F}_{2^m}^*$. Grup ini merupakan grup pergandaan yang dibentuk dari lapangan berhingga $\mathbb{F}_{2^m}$ dengan karakteristik dua yang merupakan lapangan perluasan *(Extension Field)* dari lapangan bilangan bulat modulo dua $\mathbb{Z}_2$ (Menezes, Orschot and Vanstone,

1996: 297). Selain itu, dalam skripsi ini akan ditambahkan sebuah program komputer sederhana dengan menggunakan MATLAB versi 7.1 untuk memudahkan proses perhitungan dalam penyandian pesan.

1.7. Sistematika Penulisan

Penulisan skripsi ini terbagi menjadi lima bab yang disusun secara sistematis dengan rincian masing-masing bab sebagai berikut : Bab I berisi tentang pendahuluan yang meliputi latar belakang masalah, batasan masalah, rumusan masalah, tujuan penelitian, manfaat penelitian, tinjauan pustaka, sistematika penulisan dan metode penelitian. Bab II menguraikan tentang teori-teori tentang kriptografi secara umum serta menguraikan tentang teori-teori matematis yang mendasari terbentuknya grup pergandaan modulo polinomial *irreducible* yang dibentuk dari lapangan berhingga dengan karakteristik dua. Teori matematis ini meliputi beberapa struktur aljabar dan juga beberapa hal yang berhubungan dengan polinomial. Bab III membahas masalah logaritma diskrit *(Discrete Logarithm Problem)* yang merupakan sebuah konsep yang menjadi tumpuan dari keamanan algoritma kriptografi ElGamal. Selain itu dibahas pula proses penyandian dengan menggunakan algoritma kriptografi ElGamal atas grup pergandaan modulo polinomial *irreducible.* Bab IV Memberikan gambaran proses penyandian dengan menggunakan program komputer MATLAB versi 7.1. Proses penyandian dilakukan dengan melakukan operasi aritmatika polinomial dengan menggunakan parameter bilangan yang cukup besar, sehingga akan sangat

menyusahkan jika dilakukan penghitungan secara manual. Bab V menyampaikan kesimpulan umum yang merupakan jawaban dari rumusan masalah yang terdapat di Bab I dan juga saran dari penulis tentang penelitian yang dilakukan.

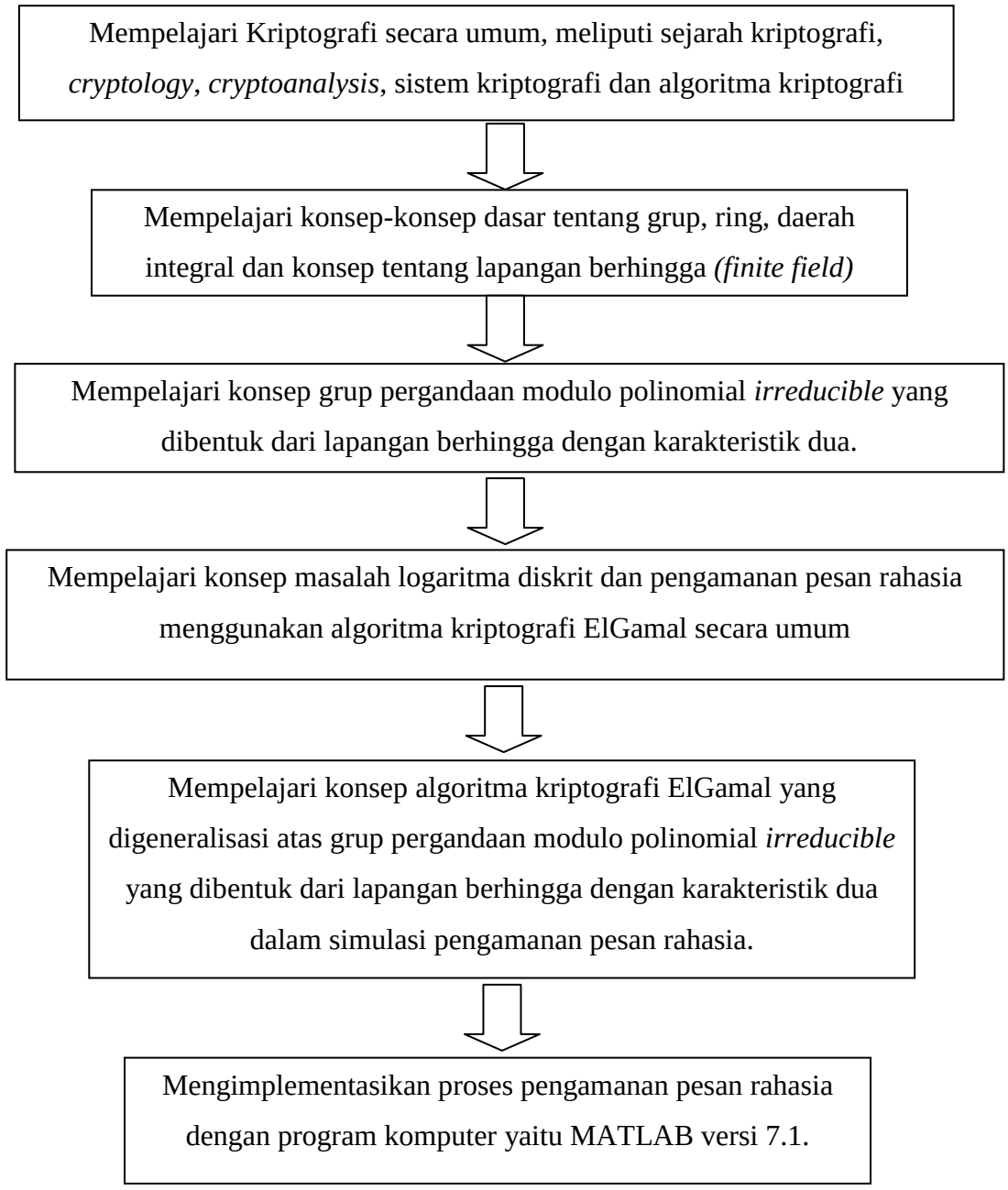
1.8. Metode Penelitian

Metode yang digunakan dalam penulisan skripsi ini adalah studi literatur. Penelitian ini dilakukan dengan cara membahas dan menjabarkan materi-materi yang sudah ada dalam literatur dalam bentuk buku-buku, dokumen-dokumen atau jurnal-jurnal untuk kemudian dipilh topik bahasan yang sesuai dengan masalah yang diangkat.

Langkah awal yang dilakukan dalam penelitian ini adalah mengkaji beberapa hal yang berhubungan dengan kriptografi. Kemudian mengkaji konsep-konsep dasar tentang grup, ring, daerah integral, dan lapangan beserta sifat-sifatnya yang dinyatakan dalam beberapa definisi dan teorema. Konsep-konsep ini nantinya digunakan sebagai dasar dalam memahami grup pergandaan modulo polinomial *irreducible* yang merupakan grup yang dibentuk dari lapangan berhingga *(finite field)* dengan karakteristik dua. Setelah itu, membuktikan bahwa grup pergandaan tersebut siklik.

Selanjutnya, dipelajari tentang aspek-aspek yang berhubungan dengan kriptografi, masalah logaritma diskrit, dan pengamanan pesan rahasia dengan algoritma kriptografi ElGamal yang digeneralisasi atas grup pergandaan modulo

polinomial *irreducible*. Grup ini dinotasikan dengan $\mathbb{F}_{2^m}^*$ yang dibentuk dari lapangan berhingga dengan karakteristik dua. Setelah itu, konsep algoritma kriptografi ElGamal yang sudah dibahas diimplementasikan dalam sebuah program komputer yaitu MATLAB versi 7.1. Alur penelitian secara utuh diberikan dalam bagan alir penelitian berikut ini :



BAB V

PENUTUP

5.1. Kesimpulan

Algoritma kriptografi ElGamal dapat digeneralisasi untuk bekerja atas sebarang grup siklik berhingga, salah satunya adalah grup pergandaan atas operasi pergandaan modulo polinomial *irreducible* $f(x)$ berderajat *m* yang dinotasikan dengan $\mathbb{F}_{2^m}^*$ atau $\mathbb{Z}_p[x]/(f(x))^*$. Grup ini dibentuk dari lapangan berhingga dengan karakteristik dua yang dinotasikan dengan $\mathbb{F}_{2^m}$. Konsep matematis yang meliputi beberapa aspek yang berhubungan dengan struktur aljabar, polinomial, dan order elemen dapat digunakan untuk membuktikan bahwa grup $\mathbb{F}_{2^m}^*$ merupakan grup siklik. Diberikan sebuah polinomial *irreducible* $f(x)$ berderajat *m*, maka grup pergandaan $\mathbb{F}_{2^m}^*$ adalah grup siklik dan mempunyai sebanyak $\varphi(2^m - 1)$ pembangun. Dengan demikian, grup pergandaan $\mathbb{F}_{2^m}^*$ dapat digunakan untuk pengamanan pesan rahasia dengan menggunakan algoritma kriptografi ElGamal.

Keamanan algoritma kriptografi ElGamal atas grup pergandaan $\mathbb{F}_{2^m}^*$ bertumpu pada masalah logaritma diskrit yang tergeneralisasi. Algoritma ini terdiri tiga proses yaitu pembangkitan kunci, enkripsi, dan dekripsi. Pembangkitan kunci dilakukan dengan menentukan suatu polinomial *irreducible* $f(x)$ berderajat *m*, menentukan nilai α yang merupakan pembangun grup $\mathbb{F}_{2^m}^*$, dan

memilih kunci rahasia $a \in \{0,1, \ldots, 2^m - 2\}$. Setelah itu, dihitung nilai $\beta \equiv \alpha^a \bmod f(x)$ sehingga dihasilkan kunci publik ($f(x), \alpha, \beta$) dan kunci rahasia a. Enkripsi adalah pemetaan plainteks *X* menjadi cipherteks (γ, δ) dengan menghitung $\gamma = \alpha^k \bmod f(x)$ dan $\delta = \beta^k . X \bmod f(x)$ untuk $k \in \{0,1, \ldots, 2^m - 2\}$. Dekripsi adalah pemetaan cipherteks (γ, δ) menjadi plainteks *X* dengan menghitung $X = \delta . \gamma^{2^m - 1 - a} \bmod f(x)$.

Implementasi pengamanan pesan rahasia atas grup pergandaan $\mathbb{F}_{2^m}^*$ dengan menggunakan program MATLAB versi 7.1. dibuat dengan menggunakan perulangan *(looping)* dalam penyandian pesan. Hal ini membuat program yang telah dibuat akan berjalan lambat jika menggunakan parameter bilangan yang besar. Fungsi-fungsi *built-in* yang ada pada MATLAB versi 7.1. memudahkan proses pembuatan program dan mejadikan program yang dibuat menjadi lebih ringkas. Program yang telah dibuat dalam penelitian ini adalah program utama yang terdiri dari 11 *m-file* yang masing masing *m-file* terhubung satu sama lain.

5.2. Saran

Setelah menyelesaikan penelitian ini, saran-saran yang dapat disampaikan adalah sebagai berikut :

1) Pembahasan landasan matematis dalam penelitian ini seperti, lapangan perluasan *(extension field)*, lapangan berhingga *(finite field)*, dan

irreducible polinomial diharapkan dapat menjadi dasar penelitian selanjutnya dalam penelitian kajian aljabar.

2) Penelitian ini hanya membahas pengamanan pesan rahasia dengan menggunakan algoritma kriptografi ElGamal atas grup pergandaan $\mathbb{F}_{2^m}^*$. Oleh karena itu, perlu diteliti lagi penerapannya pada grup pergandaan lain.
3) Penelitian ini menggunakan plainteks berupa pesan teks, sehingga perlu dilakukan penelitian lebih lanjut dengan menggunakan plainteks lain seperti *file* gambar, suara, ataupun video.
4) Perlu dilakukan penelitian lebih lanjut tentang implementasi program komputer dengan bahasa pemrograman lain yang mempu melakukan penyandian pesan dengan parameter bilangan yang lebih besar.

DAFTAR PUSTAKA

Buchmann, Johannes A., 2000, *Introduction to Cryptography*, Springer-Verlag Inc., USA.

Fraleigh, John B., 2000, *A First Course in Abstract Algebra*, Sixth Edition, Addison-Wesley Publishing Company, Inc., USA

Irving, Ronald S, 2000, *Integers, Polinomials, and Rings*, Springer-Verlag Inc. USA.

Lidl, Rudolf. Niederreiter, Harald., 1986, *Introduction to Finite Fields and Their Application*, Cambridge University Press, USA.

Menezes, Oorcshot, and Vanstone, 1996, *Handbook of Applied Cryptography*, CRC Press, Inc. USA.

Paar, Christof. Pelzl, Jan, 2009, *Understanding Cryptography*, Springer-Verlag Inc. USA.

Schneier, Bruce, 1996, *Applied Cryptography,SecondEdition:Protocol,Algorithms and Source Code in C*, John Wiley and Sons, Inc.

Stallings, William, 2003, *Cryptography and Network Security Principles and Practices*, Pearson Education, Inc. New Jersey.

Wikipedia, 2006, *Elgamal Encryption*,

http://en.wikipedia.org/wiki/Elgamal Encryption, 12 Januari 2013, 12:18.

Wikipedia, 2006, *Matlab*,

http://en.wikipedia.org/wiki/Matlab, 12 Januari 2013, 12:23.

Lamptran 1 : Kode *M-file*

Source Code elgamal.m

```
clc
h1=('_____________________________________________________________________');
h2=('#####################################################################');
h3=('#                                                                   #');
h4=('#                  Algoritma Kriptografi ElGamal                    #');
h5=('#                   Atas Grup Pergandaan F2^m*                      #');
h6=('#          oleh : Najib Mubarok Matematika 2008 UIN SuKa            #');
h7=('#                                                                   #');
h8=('#####################################################################');


header=strvcat(h1,h2,h3,h4,h5,h6,h7,h8);
menu=strvcat('1. Pembangkitaan Kunci','2. Enkripsi','3. Dekripsi');
disp(header)
disp('  ')
disp('   <<<Algoritma ElGamal>>>')
disp('  ')
disp(menu)
pilihan=input('Masukkan Pilihan = ');
    switch pilihan
        case 1
            pembangkitankunci
        case 2
            encipher
        case 3
            decipher
        otherwise
            disp('masukan hanya 1,2 atau 3')

    end
```

Source Code pembangkitankunci.m

```
clc

disp(header)
disp('  ')
disp('   <<<Pembangkitan Kunci>>>   ')
disp('  ')
menu1=('1. Polinomial Irreducible');
menu2=('2. Test elemen primitif');
menu3=('3. Buat Kunci');
menu=strvcat(menu1,menu2,menu3);
disp(menu)
tttt=input('Masukkan pilihan = ');
switch tttt
    case 1
        clc
        disp(header)
        disp('  ')
        disp('   <<<Polinomial Irreducible>>>   ')
        disp('  ')

        m=input('masukkan derajat irreducible polinomial = ');
        disp(strvcat('1. tampilkan satu','2. tampilkan semua'))
```

```
        ttttt=input('Masukkan pilihan (1/2)= ');
            switch ttttt
                  case 1
                     primpoly(m)
   ttt=input('tekan 1 untuk melanjutkan atau sebarang angka untuk
keluar');
                        switch ttt
                        case 1
                            elgamal
                            otherwise
                        clc
                        end
                    case 2
                        primpoly(m,'all')
   ttt=input('tekan 1 untuk melanjutkan atau sebarang angka untuk
keluar');
                        switch ttt
                        case 1
                            elgamal
                            otherwise
                        clc
                        end
                otherwise
                    disp('masukkan harus 1 atau 2')
            end
    case 2
    primitifcek

    case 3

clc
        disp(header)
        disp('   ')
        disp('    <<<Buat Kunci>>>    ')
        disp('   ')
    otherwise
        disp('masukan hanya 1,2 atau 3')
end

inkunci3=input('masukkan irreducible polinomial f(x) berderajat m = ');
deg=length(inkunci3)-1;
inkunci4=input('Masukkan elemen pembangun = ');
inkunci2=input([('Pilih kunci rahasia dari 0 - ') num2str((2^deg)-2) (' =
')]);

bentukbiner=inkunci3;

converttodecimal;

fx=dec2bin(bentukdesimal);

bentukbiner=inkunci4;

converttodecimal;

desinkunci4=bentukdesimal;

in1=desinkunci4;
in2=desinkunci4;
power=inkunci2;
pfx=inkunci3;

perpangkatan;
```

```
disp('    ')
disp('    <<<Kunci publiknya adalah>>>')
disp('    ')
disp(['f(x) = ' (fx)])
gen=dec2bin(desinkunci4,deg);
disp(['Alpha = ' (gen)])
disp(['betha = ' (betha)])
disp('    ')
disp('    <<<kunci rahasianya adalah>>>')
aaaa=num2str(inkunci2);
disp(['nilai a yang dipilih = ' (aaaa)])
disp('    ')
ttt=input('tekan 1 untuk melanjutkan atau sebarang angka untuk keluar');
switch ttt
    case 1
        elgamal
    otherwise
        clc
end
```

Source Code `encipher.m`

```
clc

disp(header)
disp('   ')
disp('    <<<Enkripsi>>>')
disp('   ')
disp('Inputkan semua kunci publik')
alfa=input('Masukkan generator = ');
beta=input('Masukkan beta = ');
irre=input('Masukkan irreducible polinomial derajat m = ');
derajat=length(irre)-1;
pesen=input('Masukkan pesan rahasia (maksimal 25 karakter = ');
lengthpesen=length(pesen);
disp(['Masukkan ' num2str(lengthpesen) ' bilangan acak k dari 0 - '
num2str((2^derajat)-2)])

for ooo=1:lengthpesen
    switch ooo
        case 1
            me1=double(pesen(1,1));
            k1=input('k1 = ');
        case 2
            me2=double(pesen(1,2));
            k2=input('k2 = ');
        case 3
            me3=double(pesen(1,3));
            k3=input('k3 = ');
        case 4
            me4=double(pesen(1,4));
            k4=input('k4 = ');
        case 5
            me5=double(pesen(1,5));
            k5=input('k5 = ');
        case 6
            me6=double(pesen(1,6));
            k6=input('k6 = ');
        case 7
            me7=double(pesen(1,7));
            k7=input('k7 = ');
        case 8
```

```
            me8=double(pesen(1,8));
            k8=input('k8 = ');
        case 9
            me9=double(pesen(1,9));
            k9=input('k9 = ');
        case 10
            me10=double(pesen(1,10));
            k10=input('k10 = ');
        case 11
            me11=double(pesen(1,11));
            k11=input('k11 = ');
        case 12
            me12=double(pesen(1,12));
            k12=input('k12 = ');
        case 13
            me13=double(pesen(1,13));
            k13=input('k13 = ');
        case 14
            me14=double(pesen(1,14));
            k14=input('k14 = ');
        case 15
            me15=double(pesen(1,15));
            k15=input('k15 = ');
        case 16
            me16=double(pesen(1,16));
            k16=input('k16 = ');
        case 17
            me17=double(pesen(1,17));
            k17=input('k17 = ');
        case 18
            me18=double(pesen(1,18));
            k18=input('k18 = ');
        case 19
            me19=double(pesen(1,19));
            k19=input('k19 = ');
        case 20
            me20=double(pesen(1,20));
            k20=input('k20 = ');
        case 21
            me21=double(pesen(1,21));
            k21=input('k21 = ');
        case 22
            me22=double(pesen(1,22));
            k22=input('k22 = ');
        case 23
            me23=double(pesen(1,23));
            k23=input('k23 = ');
        case 24
            me24=double(pesen(1,24));
            k24=input('k24 = ');
        case 25
            me25=double(pesen(1,25));
            k25=input('k25 = ');
    end
end


for z =1:(length(pesen))
    switch z
    case 1
        plainteks=me1;
        random=k1;
        subencipher
```

```
        gama1=keluaranA;
        delta1=keluaranB;
        disp(['plainteks  = char1,ASCII1   = ' (pesen(1,1)) ','
dec2bin(me1,8) ])
        disp(['cipherteks = gama1,delta1   = ' (gama1) ',' (delta1)])
        case 2
        plainteks=me2;
        random=k2;
        subencipher
        gama2=keluaranA;
        delta2=keluaranB ;
        disp(['plainteks  = char2,ASCII2   = ' (pesen(1,2)) ','
dec2bin(me2,8) ])
        disp(['cipherteks = gama2,delta2   = ' (gama2) ',' (delta2)])
        case 3
        plainteks=me3;
        random=k3;
        subencipher
        gama3=keluaranA;
        delta3=keluaranB;
        disp(['plainteks  = char3,ASCII3   = ' (pesen(1,3)) ','
dec2bin(me3,8) ])
        disp(['cipherteks = gama3,delta3   = ' (gama3) ',' (delta3)])
        case 4
        plainteks=me4;
        random=k4;
        subencipher
        gama4=keluaranA;
        delta4=keluaranB;
        disp(['plainteks  = char4,ASCII4   = ' (pesen(1,4)) ','
dec2bin(me4,8) ])
        disp(['cipherteks = gama4,delta4   = ' (gama4) ',' (delta4)])
        case 5
        plainteks=me5;
        random=k5;
        subencipher
        gama5=keluaranA;
        delta5=keluaranB;
        disp(['plainteks  = char5,ASCII5   = ' (pesen(1,5)) ','
dec2bin(me5,8) ])
        disp(['cipherteks = gama5,delta5   = ' (gama5) ',' (delta5)])
        case 6
        plainteks=me6;
        random=k6;
        subencipher
        gama6=keluaranA;
        delta6=keluaranB;
        disp(['plainteks  = char6,ASCII6   = ' (pesen(1,6)) ','
dec2bin(me6,8) ])
        disp(['cipherteks = gama6,delta6   = ' (gama6) ',' (delta6)])
        case 7
        plainteks=me7;
        random=k7;
        subencipher
        gama7=keluaranA;
        delta7=keluaranB;
        disp(['plainteks  = char7,ASCII7   = ' (pesen(1,7)) ','
dec2bin(me7,8) ])
        disp(['cipherteks = gama7,delta7   = ' (gama7) ',' (delta7)])
        case 8
        plainteks=me8;
        random=k8;
        subencipher
```

```
        gama8=keluaranA;
        delta8=keluaranB;
        disp(['plainteks  = char8,ASCII8   = ' (pesen(1,8)) ','
dec2bin(me8,8) ])
        disp(['cipherteks = gama8,delta8   = ' (gama8) ',' (delta8)])
        case 9
        plainteks=me9;
        random=k9;
        subencipher
        gama9=keluaranA;
        delta9=keluaranB;
        disp(['plainteks  = char9,ASCII9   = ' (pesen(1,9)) ','
dec2bin(me9,8) ])
        disp(['cipherteks = gama9,delta9   = ' (gama9) ',' (delta9)])
        case 10
        plainteks=me10;
        random=k10;
        subencipher
        gama10=keluaranA;
        delta10=keluaranB;
        disp(['plainteks  = char10,ASCII10 = ' (pesen(1,10)) ','
dec2bin(me10,8) ])
        disp(['cipherteks = gama10,delta10 = ' (gama10) ',' (delta10)])
        case 11
        plainteks=me11;
        random=k11;
        subencipher
        gama11=keluaranA;
        delta11=keluaranB;
        disp(['plainteks  = char11,ASCII11 = ' (pesen(1,11)) ','
dec2bin(me11,8) ])
        disp(['cipherteks = gama11,delta11 = ' (gama11) ',' (delta11)])
        case 12
        plainteks=me12;
        random=k12;
        subencipher
        gama12=keluaranA;
        delta12=keluaranB;
        disp(['plainteks  = char12,ASCII12 = ' (pesen(1,12)) ','
dec2bin(me12,8) ])
        disp(['cipherteks = gama12,delta12 = ' (gama12) ',' (delta12)])
        case 13
        plainteks=me13;
        random=k13;
        subencipher
        gama13=keluaranA;
        delta13=keluaranB;
        disp(['plainteks  = char13,ASCII13 = ' (pesen(1,13)) ','
dec2bin(me13,8) ])
        disp(['cipherteks = gama13,delta13 = ' (gama13) ',' (delta13)])
        case 14
        plainteks=me14;
        random=k14;
        subencipher
        gama14=keluaranA;
        delta14=keluaranB;
        disp(['plainteks  = char14,ASCII14 = ' (pesen(1,14)) ','
dec2bin(me14,8) ])
        disp(['cipherteks = gama14,delta14 = ' (gama14) ',' (delta14)])
        case 15
        plainteks=me15;
        random=k15;
        subencipher
```

```
        gama15=keluaranA;
        delta15=keluaranB;
        disp(['plainteks  = char15,ASCII15 = ' (pesen(1,15)) ','
dec2bin(me15,8) ])
        disp(['cipherteks = gama15,delta15 = ' (gama15) ',' (delta15)])
        case 16
        plainteks=me16;
        random=k16;
        subencipher
        gama16=keluaranA;
        delta16=keluaranB;
        disp(['plainteks  = char16,ASCII16 = ' (pesen(1,16)) ','
dec2bin(me16,8) ])
        disp(['cipherteks = gama16,delta16 = ' (gama16) ',' (delta16)])
        case 17
        plainteks=me17;
        random=k17;
        subencipher
        gama17=keluaranA;
        delta17=keluaranB;
        disp(['plainteks  = char17,ASCII17 = ' (pesen(1,17)) ','
dec2bin(me17,8) ])
        disp(['cipherteks = gama17,delta17 = ' (gama17) ',' (delta17)])
        case 18
        plainteks=me18;
        random=k18;
        subencipher
        gama18=keluaranA;
        delta18=keluaranB;
        disp(['plainteks  = char18,ASCII18 = ' (pesen(1,18)) ','
dec2bin(me18,8) ])
        disp(['cipherteks = gama18,delta18 = ' (gama18) ',' (delta18)])
        case 19
        plainteks=me19;
        random=k19;
        subencipher
        gama19=keluaranA;
        delta19=keluaranB;
        disp(['plainteks  = char19,ASCII19 = ' (pesen(1,19)) ','
dec2bin(me19,8) ])
        disp(['cipherteks = gama19,delta19 = ' (gama19) ',' (delta19)])
        case 20
        plainteks=me20;
        random=k20;
        subencipher
        gama20=keluaranA;
        delta20=keluaranB;
        disp(['plainteks  = char20,ASCII20 = ' (pesen(1,20)) ','
dec2bin(me20,8) ])
        disp(['cipherteks = gama20,delta20 = ' (gama20) ',' (delta20)])
        case 21
        plainteks=me21;
        random=k21;
        subencipher
        gama21=keluaranA;
        delta21=keluaranB;
        disp(['plainteks  = char21,ASCII21 = ' (pesen(1,21)) ','
dec2bin(me21,8) ])
        disp(['cipherteks = gama11,delta11 = ' (gama21) ',' (delta21)])
        case 22
        plainteks=me22;
        random=k22;
        subencipher
```

```
        gama22=keluaranA;
        delta22=keluaranB;
        disp(['plainteks  = char22,ASCII22 = ' (pesen(1,22)) ','
dec2bin(me22,8) ])
        disp(['cipherteks = gama22,delta22 = ' (gama22) ',' (delta22)])
        case 23
        plainteks=me23;
        random=k23;
        subencipher
        gama23=keluaranA;
        delta23=keluaranB;
        disp(['plainteks  = char23,ASCII23 = ' (pesen(1,23)) ','
dec2bin(me23,8) ])
        disp(['cipherteks = gama23,delta23 = ' (gama23) ',' (delta23)])
        case 24
        plainteks=me24;
        random=k24;
        subencipher
        gama24=keluaranA;
        delta24=keluaranB;
        disp(['plainteks  = char24,ASCII24 = ' (pesen(1,24)) ','
dec2bin(me24,8) ])
        disp(['cipherteks = gama24,delta24 = ' (gama24) ',' (delta24)])
        case 25
        plainteks=me25;
        random=k25;
        subencipher
        gama25=keluaranA;
        delta25=keluaranB;
        disp(['plainteks  = char25,ASCII25 = ' (pesen(1,25)) ','
dec2bin(me25,8)])
        disp(['cipherteks = gama25,delta25 = ' (gama25) ',' (delta25)])

    end
end
ttt=input('Tekan 1 untuk melanjukan atau sebarang angka untuk keluar');
switch ttt
    case 1
        elgamal
    otherwise
        clc
end
```

Source Code `decipher.m`

```
clc

disp(header)
disp('  ')
disp('   <<<Dekripsi>>>')
disp('  ')

private=input('masukkan kunci rahasia = ');
disp('  ')
disp('tekan 1 untuk menggunakan cipherteks dari proses enkripsi
sebelumnya')
disp('atau sebarang angka untuk memasukkan cipherteks baru')
bbb=input(' ');
switch bbb
    case 1
        zv=lengthpesen;
        disp(['digunakan ' num2str(lengthpesen) ' chiperteks dari proses
sebelumnya'])
```

```
        deg=derajat;
        pfx=irre;

    otherwise

        irreduce=input('Masukkan irreducible polinomial derajat m = ');
        deg=length(irreduce)-1;
        pfx=irreduce;
        lengthcipher=input('masukkan jumlah cipherteks = ');
        zv=lengthcipher;
        disp(['masukkan ' num2str(lengthcipher) 'cipherteks (dalam bentuk
string)'])
        for qqq=1:lengthcipher
            switch qqq
                case 1
                    gama1=input('masukkan gama1  = ');
                    delta1=input('masukkan delta1 = ');
                case 2
                    gama2=input('masukkan gama2  = ');
                    delta2=input('masukkan delta2 = ');
                case 3
                    gama3=input('masukkan gama3  = ');
                    delta3=input('masukkan delta3 = ');
                case 4
                    gama4=input('masukkan gama4  = ');
                    delta4=input('masukkan delta4 = ');
                case 5
                    gama5=input('masukkan gama5  = ');
                    delta5=input('masukkan delta5 = ');
                case 6
                    gama6=input('masukkan gama6  = ');
                    delta6=input('masukkan delta6 = ');
                case 7
                    gama7=input('masukkan gama7  = ');
                    delta7=input('masukkan delta7 = ');
                case 8
                    gama8=input('masukkan gama8  = ');
                    delta8=input('masukkan delta8 = ');
                case 9
                    gama9=input('masukkan gama9  = ');
                    delta9=input('masukkan delta9 = ');
                case 10
                    gama10=input('masukkan gama10  = ');
                    delta10=input('masukkan delta10 = ');
                case 11
                    gama11=input('masukkan gama11  = ');
                    delta11=input('masukkan delta11 = ');
                case 12
                    gama12=input('masukkan gama12  = ');
                    delta12=input('masukkan delta12 = ');
                case 13
                    gama13=input('masukkan gama13  = ');
                    delta13=input('masukkan delta13 = ');
                case 14
                    gama14=input('masukkan gama14  = ');
                    delta14=input('masukkan delta14 = ');
                case 15
                    gama15=input('masukkan gama15  = ');
                    delta15=input('masukkan delta15 = ');
                case 16
                    gama16=input('masukkan gama16  = ');
                    delta16=input('masukkan delta16 = ');
                case 17
```

```
                    gama17=input('masukkan gama17  = ');
                    delta17=input('masukkan delta17 = ');
                case 18
                    gama18=input('masukkan gama18  = ');
                    delta18=input('masukkan delta18 = ');
                case 19
                    gama19=input('masukkan gama19  = ');
                    delta19=input('masukkan delta19 = ');
                case 20
                    gama20=input('masukkan gama20  = ');
                    delta20=input('masukkan delta20 = ');
                case 21
                    gama21=input('masukkan gama21  = ');
                    delta21=input('masukkan delta21 = ');
                case 22
                    gama22=input('masukkan gama22  = ');
                    delta22=input('masukkan delta22 = ');
                case 23
                    gama23=input('masukkan gama23  = ');
                    delta23=input('masukkan delta23 = ');
                case 24
                    gama24=input('masukkan gama24  = ');
                    delta24=input('masukkan delta24 = ');
                case 25
                    gama25=input('masukkan gama25  = ');
                    delta25=input('masukkan delta25 = ');
            end
        end
end

que=input('Tekan 1 untuk mulai dekripsi atau sebarang angka untuk
kembali');
        switch que
            case 1
        for zvv=1:zv
        switch zvv
            case 1
                inputA=gama1;
                inputB=delta1;
                disp(['cipherteks = gama1,delta1     = ' '       '
(gama1) ') (' (delta1)])
                subdecipher;
                gamapangkata1=outp;
                origin1=output;
                disp(['pesan semula                  = ' (origin1)])
            case 2
                inputA=gama2;
                inputB=delta2;
                disp(['cipherteks = gama2,delta2     = ' '       '
(gama2) ',' (delta2)])
                subdecipher;
                gamapangkata2=outp;
                origin2=output;
                disp(['pesan semula                  = ' (origin2)])
            case 3
                inputA=gama3;
                inputB=delta3;
                disp(['cipherteks = gama3,delta3     = ' '       '
(gama3) ',' (delta3)])
                subdecipher;
                gamapangkata3=outp;
                origin3=output;
                disp(['pesan semula                  = ' (origin3)])
```

```
            case 4
                inputA=gama4;
                inputB=delta4;
                disp(['cipherteks = gama4,delta4     = ' '       '
(gama4) ',' (delta4)])
                subdecipher;
                gamapangkata4=outp;
                origin4=output;
                disp(['pesan semula                  = ' (origin4)])
            case 5
                inputA=gama5;
                inputB=delta5;
                disp(['cipherteks = gama5,delta5     = ' '       '
(gama5) ',' (delta5)])
                subdecipher;
                gamapangkata5=outp;
                origin5=output;
                disp(['pesan semula                  = ' (origin5)])
            case 6
                inputA=gama6;
                inputB=delta6;
                disp(['cipherteks = gama6,delta6     = ' '       '
(gama6) ',' (delta6)])
                subdecipher;
                gamapangkata6=outp;
                origin6=output;
                disp(['pesan semula                  = ' (origin6)])
            case 7
                inputA=gama7;
                inputB=delta7;
                disp(['cipherteks = gama7,delta7     = ' '       '
(gama7) ',' (delta7)])
                subdecipher;
                gamapangkata7=outp;
                origin7=output;
                disp(['pesan semula                  = ' (origin7)])
            case 8
                inputA=gama8;
                inputB=delta8;
                disp(['cipherteks = gama8,delta8     = ' '       '
(gama8) ',' (delta8)])
                subdecipher;
                gamapangkata8=outp;
                origin8=output;
                disp(['pesan semula                  = ' (origin8)])
            case 9
                inputA=gama9;
                inputB=delta9;
                disp(['cipherteks = gama9,delta9     = ' '       '
(gama9) ',' (delta9)])
                subdecipher;
                gamapangkata9=outp;
                origin9=output;
                disp(['pesan semula                  = ' (origin9)])
            case 10
                inputA=gama10;
                inputB=delta10;
                disp(['cipherteks = gama10,delta10   = ' '       '
(gama10) ',' (delta10)])
                subdecipher;
                gamapangkata10=outp;
                origin10=output;
                disp(['pesan semula                  = ' (origin10)])
```

```
            case 11
                inputA=gama11;
                inputB=delta11;
                disp(['cipherteks = gama11,delta11   = ' '       '
(gama11) ',' (delta11)])
                subdecipher;
                gamapangkata11=outp;
                origin11=output;
                disp(['pesan semula                  = ' (origin11)])
            case 12
                inputA=gama12;
                inputB=delta12;
                disp(['cipherteks = gama12,delta12   = ' '       '
(gama12) ',' (delta12)])
                subdecipher;
                gamapangkata12=outp;
                origin12=output;
                disp(['pesan semula                  = ' (origin12)])
            case 13
                inputA=gama13;
                inputB=delta13;
                disp(['cipherteks = gama13,delta13   = ' '       '
(gama13) ',' (delta13)])
                subdecipher;
                gamapangkata13=outp;
                origin13=output;
                disp(['pesan semula                  = ' (origin13)])
            case 14
                inputA=gama14;
                inputB=delta14;
                disp(['cipherteks = gama14,delta14   = ' '       '
(gama14) ',' (delta14)])
                subdecipher;
                gamapangkata14=outp;
                origin14=output;
                disp(['pesan semula                  = ' (origin14)])
            case 15
                inputA=gama15;
                inputB=delta15;
                disp(['cipherteks = gama15,delta15   = ' '       '
(gama15) ',' (delta15)])
                subdecipher;
                gamapangkata15=outp;
                origin15=output;
                disp(['pesan semula                  = ' (origin15)])
            case 16
                inputA=gama16;
                inputB=delta16;
                disp(['cipherteks = gama16,delta16   = ' '       '
(gama16) ',' (delta16)])
                subdecipher;
                gamapangkata16=outp;
                origin16=output;
                disp(['pesan semula                  = ' (origin16)])
            case 17
                inputA=gama17;
                inputB=delta17;
                disp(['cipherteks = gama17,delta17   = ' '       '
(gama17) ',' (delta17)])
                subdecipher;
                gamapangkata17=outp;
                origin17=output;
                disp(['pesan semula                  = ' (origin17)])
```

```
            case 18
                inputA=gama18;
                inputB=delta18;
                disp(['cipherteks = gama18,delta18   = ' '       '
(gama18) ',' (delta18)])
                subdecipher;
                gamapangkata18=outp;
                origin18=output;
                disp(['pesan semula                  = ' (origin18)])
            case 19
                inputA=gama19;
                inputB=delta19;
                disp(['cipherteks = gama19,delta19   = ' '       '
(gama19) ',' (delta19)])
                subdecipher;
                gamapangkata19=outp;
                origin19=output;
                disp(['pesan semula                  = ' (origin19)])
            case 20
                inputA=gama20;
                inputB=delta20;
                disp(['cipherteks = gama20,delta20   = ' '       '
(gama20) ',' (delta20)])
                subdecipher;
                gamapangkata20=outp;
                origin20=output;
                disp(['pesan semula                  = ' (origin20)])
            case 21
                inputA=gama21;
                inputB=delta21;
                disp(['cipherteks = gama21,delta21   = ' '       '
(gama21) ',' (delta21)])
                subdecipher;
                gamapangkata21=outp;
                origin21=output;
                disp(['pesan semula                  = ' (origin21)])
            case 22
                inputA=gama22;
                inputB=delta22;
                disp(['cipherteks = gama22,delta22   = ' '       '
(gama22) ',' (delta22)])
                subdecipher;
                gamapangkata22=outp;
                origin22=output;
                disp(['pesan semula                  = ' (origin22)])
            case 23
                inputA=gama23;
                inputB=delta23;
                disp(['cipherteks = gama23,delta23   = ' '       '
(gama23) ',' (delta23)])
                subdecipher;
                gamapangkata23=outp;
                origin23=output;
                disp(['pesan semula                  = ' (origin23)])
            case 24
                inputA=gama24;
                inputB=delta24;
                disp(['cipherteks = gama24,delta24   = ' '       '
(gama24) ',' (delta24)])
                subdecipher;
                gamapangkata24=outp;
                origin24=output;
                disp(['pesan semula                  = ' (origin24)])
```

```
            case 25
                inputA=gama25;
                inputB=delta25;
                disp(['cipherteks = gama25,delta25   = ' '       '
(gama25) ',' (delta25)])
                subdecipher;
                gamapangkata25=outp;
                origin25=output;
                disp(['pesan semula                  = ' (origin25)])

        end
        end
```

Source Code subencipher.m

```
bentukbiner=alfa;
converttodecimal;
desimalalfa=bentukdesimal;

deg=derajat;
in1=desimalalfa;
in2=desimalalfa;
if random<2
    random=random+((2^deg)-1);
    uu=random;
end
power=random;
pfx=irre;
perpangkatan;
keluaranA=betha;

bentukbiner=beta;
converttodecimal;
desimalbeta=bentukdesimal;

in1=desimalbeta;
in2=desimalbeta;
perpangkatan;
betapangkatk=bin2dec(betha);

in1=plainteks;
in2=betapangkatk;
in1xin2;
keluaranB=betha;
```

Source Code subdecipher.m

```
desimalgama=bin2dec(inputA);
desimaldelta=bin2dec(inputB);

in1=desimalgama;
in2=desimalgama;
hhh=(2^deg)-1-private;

if hhh<2
    hhh=hhh+((2^deg)-1);
```

```
    uu=random;
end

power=hhh;
perpangkatan;
outp=betha;

in1=desimaldelta;
in2=bin2dec(outp);
in1xin2;
output=char(bin2dec(betha));
```

Source Code converttodecimal.m

```
panjangbentukbiner=length(bentukbiner);
for z =1:panjangbentukbiner
    switch z
    case 1
        a1=mod(bentukbiner(:,1),2); a2=num2str(a1);
    case 2
        b1=mod(bentukbiner(:,2),2); b2=num2str(b1);
    case 3
        c1=mod(bentukbiner(:,3),2); c2=num2str(c1);
    case 4
        d1=mod(bentukbiner(:,4),2); d2=num2str(d1);
    case 5
        e1=mod(bentukbiner(:,5),2); e2=num2str(e1);
    case 6
        f1=mod(bentukbiner(:,6),2); f2=num2str(f1);
    case 7
        g1=mod(bentukbiner(:,7),2); g2=num2str(g1);
    case 8
        h1=mod(bentukbiner(:,8),2); h2=num2str(h1);
    case 9
        i1=mod(bentukbiner(:,9),2); i2=num2str(i1);
    case 10
        j1=mod(bentukbiner(:,10),2); j2=num2str(j1);
    case 11
        k1=mod(bentukbiner(:,11),2); k2=num2str(k1);
    case 12
        l1=mod(bentukbiner(:,12),2); l2=num2str(l1);
    case 13
        m1=mod(bentukbiner(:,13),2); m2=num2str(m1);
    case 14
        n1=mod(bentukbiner(:,14),2); n2=num2str(n1);
    case 15
        o1=mod(bentukbiner(:,15),2); o2=num2str(o1);
    case 16
        p1=mod(bentukbiner(:,16),2); p2=num2str(p1);
    end
end
switch panjangbentukbiner
    case 1
        bentukstring=strcat(a2);
    case 2
        bentukstring=strcat(a2,b2);
    case 3
        bentukstring=strcat(a2,b2,c2);
    case 4
        bentukstring=strcat(a2,b2,c2,d2);
    case 5
```

```
        bentukstring=strcat(a2,b2,c2,d2,e2);
    case 6
        bentukstring=strcat(a2,b2,c2,d2,e2,f2);
    case 7
        bentukstring=strcat(a2,b2,c2,d2,e2,f2,g2);
    case 8
        bentukstring=strcat(a2,b2,c2,d2,e2,f2,g2,h2);
    case 9
        bentukstring=strcat(a2,b2,c2,d2,e2,f2,g2,h2,i2);
    case 10
        bentukstring=strcat(a2,b2,c2,d2,e2,f2,g2,h2,i2,j2);
    case 11
        bentukstring=strcat(a2,b2,c2,d2,e2,f2,g2,h2,i2,j2,k2);
    case 12
        bentukstring=strcat(a2,b2,c2,d2,e2,f2,g2,h2,i2,j2,k2,l2);
    case 13
        bentukstring=strcat(a2,b2,c2,d2,e2,f2,g2,h2,i2,j2,k2,l2,m2);
    case 14
        bentukstring=strcat(a2,b2,c2,d2,e2,f2,g2,h2,i2,j2,k2,l2,m2,n2);
    case 15

bentukstring=strcat(a2,b2,c2,d2,e2,f2,g2,h2,i2,j2,k2,l2,m2,n2,o2);
    case 16

bentukstring=strcat(a2,b2,c2,d2,e2,f2,g2,h2,i2,j2,k2,l2,m2,n2,o2,p2);
end
bentukdesimal=bin2dec(bentukstring);
```

Source Code `perpangkatan.m`

```
for i=1:power-1
dessisa= in1;

binery1=dec2bin(in1,deg);
binery2=dec2bin(in2,deg);

ddd=length(binery1);
for tt=1:ddd
    switch tt
        case 1
            m1=str2num(binery1(1,1));
            n1=str2num(binery2(1,1));
        case 2
            m2=str2num(binery1(1,2));
            n2=str2num(binery2(1,2));
        case 3
            m3=str2num(binery1(1,3));
            n3=str2num(binery2(1,3));
        case 4
            m4=str2num(binery1(1,4));
            n4=str2num(binery2(1,4));
        case 5
            m5=str2num(binery1(1,5));
            n5=str2num(binery2(1,5));
        case 6
            m6=str2num(binery1(1,6));
            n6=str2num(binery2(1,6));
        case 7
            m7=str2num(binery1(1,7));
            n7=str2num(binery2(1,7));
        case 8
            m8=str2num(binery1(1,8));
```

```
            n8=str2num(binery2(1,8));
        case 9
            m9=str2num(binery1(1,9));
            n9=str2num(binery2(1,9));
        case 10
            m10=str2num(binery1(1,10));
            n10=str2num(binery2(1,10));
        case 11
            m11=str2num(binery1(1,11));
            n11=str2num(binery2(1,11));
        case 12
            m12=str2num(binery1(1,12));
            n12=str2num(binery2(1,12));
        case 13
            m13=str2num(binery1(1,13));
            n13=str2num(binery2(1,13));
        case 14
            m14=str2num(binery1(1,14));
            n14=str2num(binery2(1,14));
        case 15
            m15=str2num(binery1(1,15));
            n15=str2num(binery2(1,15));
        case 16
            m16=str2num(binery1(1,16));
            n16=str2num(binery2(1,16));
    end
end

switch ddd
    case 1
        message1= [m1];
        message2= [n1];
result=conv(message1,message2);
hasil=[mod(result(:,1),2)];

    case 2
        message1= [m1 m2];
        message2= [n1 n2];
result=conv(message1,message2);
hasil=[mod(result(:,1),2) mod(result(:,2),2) mod(result(:,3),2)];

    case 3
        message1= [m1 m2 m3];
        message2= [n1 n2 n3];
result=conv(message1,message2);
hasil=[mod(result(:,1),2) mod(result(:,2),2) mod(result(:,3),2)
mod(result(:,4),2) mod(result(:,5),2)];

    case 4
        message1= [m1 m2 m3 m4];
        message2= [n1 n2 n3 n4];
result=conv(message1,message2);
hasil=[mod(result(:,1),2) mod(result(:,2),2) mod(result(:,3),2)
mod(result(:,4),2) mod(result(:,5),2) mod(result(:,6),2)
mod(result(:,7),2)];

    case 5
        message1= [m1 m2 m3 m4 m5];
        message2= [n1 n2 n3 n4 n5];
result=conv(message1,message2);
hasil=[mod(result(:,1),2) mod(result(:,2),2) mod(result(:,3),2)
mod(result(:,4),2) mod(result(:,5),2) mod(result(:,6),2)
mod(result(:,7),2) mod(result(:,8),2) mod(result(:,9),2)];
```

```
    case 6
        message1= [m1 m2 m3 m4 m5 m6];
        message2= [n1 n2 n3 n4 n5 n6];
result=conv(message1,message2);
hasil=[mod(result(:,1),2) mod(result(:,2),2) mod(result(:,3),2)
mod(result(:,4),2) mod(result(:,5),2) mod(result(:,6),2)
mod(result(:,7),2) mod(result(:,8),2) mod(result(:,9),2)
mod(result(:,10),2) mod(result(:,11),2)];

    case 7
        message1= [m1 m2 m3 m4 m5 m6 m7];
        message2= [n1 n2 n3 n4 n5 n6 n7];
result=conv(message1,message2);
hasil=[mod(result(:,1),2) mod(result(:,2),2) mod(result(:,3),2)
mod(result(:,4),2) mod(result(:,5),2) mod(result(:,6),2)
mod(result(:,7),2) mod(result(:,8),2) mod(result(:,9),2)
mod(result(:,10),2) mod(result(:,11),2) mod(result(:,12),2)
mod(result(:,13),2)];

    case 8
        message1= [m1 m2 m3 m4 m5 m6 m7 m8];
        message2= [n1 n2 n3 n4 n5 n6 n7 n8];
result=conv(message1,message2);
hasil=[mod(result(:,1),2) mod(result(:,2),2) mod(result(:,3),2)
mod(result(:,4),2) mod(result(:,5),2) mod(result(:,6),2)
mod(result(:,7),2) mod(result(:,8),2) mod(result(:,9),2)
mod(result(:,10),2) mod(result(:,11),2) mod(result(:,12),2)
mod(result(:,13),2) mod(result(:,14),2) mod(result(:,15),2)];
    case 9
        message1= [m1 m2 m3 m4 m5 m6 m7 m8 m9 ];
        message2= [n1 n2 n3 n4 n5 n6 n7 n8 n9 ];
result=conv(message1,message2);
hasil=[mod(result(:,1),2) mod(result(:,2),2) mod(result(:,3),2)
mod(result(:,4),2) mod(result(:,5),2) mod(result(:,6),2)
mod(result(:,7),2) mod(result(:,8),2) mod(result(:,9),2)
mod(result(:,10),2) mod(result(:,11),2) mod(result(:,12),2)
mod(result(:,13),2) mod(result(:,14),2) mod(result(:,15),2)
mod(result(:,16),2) mod(result(:,17),2)];
    case 10
        message1= [m1 m2 m3 m4 m5 m6 m7 m8 m9 m10];
        message2= [n1 n2 n3 n4 n5 n6 n7 n8 n9 n10];
result=conv(message1,message2);
hasil=[mod(result(:,1),2) mod(result(:,2),2) mod(result(:,3),2)
mod(result(:,4),2) mod(result(:,5),2) mod(result(:,6),2)
mod(result(:,7),2) mod(result(:,8),2) mod(result(:,9),2)
mod(result(:,10),2) mod(result(:,11),2) mod(result(:,12),2)
mod(result(:,13),2) mod(result(:,14),2) mod(result(:,15),2)
mod(result(:,16),2) mod(result(:,17),2) mod(result(:,18),2)
mod(result(:,19),2)];
    case 11
        message1= [m1 m2 m3 m4 m5 m6 m7 m8 m9 m10 m11];
        message2= [n1 n2 n3 n4 n5 n6 n7 n8 n9 n10 n11];
result=conv(message1,message2);
hasil=[mod(result(:,1),2) mod(result(:,2),2) mod(result(:,3),2)
mod(result(:,4),2) mod(result(:,5),2) mod(result(:,6),2)
mod(result(:,7),2) mod(result(:,8),2) mod(result(:,9),2)
mod(result(:,10),2) mod(result(:,11),2) mod(result(:,12),2)
mod(result(:,13),2) mod(result(:,14),2) mod(result(:,15),2)
mod(result(:,16),2) mod(result(:,17),2) mod(result(:,18),2)
mod(result(:,19),2) mod(result(:,20),2) mod(result(:,21),2)];
    case 12
        message1= [m1 m2 m3 m4 m5 m6 m7 m8 m9 m10 m11 m12];
```

```
        message2= [n1 n2 n3 n4 n5 n6 n7 n8 n9 n10 n11 n12];
result=conv(message1,message2);
hasil=[mod(result(:,1),2) mod(result(:,2),2) mod(result(:,3),2)
mod(result(:,4),2) mod(result(:,5),2) mod(result(:,6),2)
mod(result(:,7),2) mod(result(:,8),2) mod(result(:,9),2)
mod(result(:,10),2) mod(result(:,11),2) mod(result(:,12),2)
mod(result(:,13),2) mod(result(:,14),2) mod(result(:,15),2)
mod(result(:,16),2) mod(result(:,17),2) mod(result(:,18),2)
mod(result(:,19),2) mod(result(:,20),2) mod(result(:,21),2)
mod(result(:,22),2) mod(result(:,23),2)];
    case 13
        message1= [m1 m2 m3 m4 m5 m6 m7 m8 m9 m10 m11 m12 m13];
        message2= [n1 n2 n3 n4 n5 n6 n7 n8 n9 n10 n11 n12 n13];
result=conv(message1,message2);
hasil=[mod(result(:,1),2) mod(result(:,2),2) mod(result(:,3),2)
mod(result(:,4),2) mod(result(:,5),2) mod(result(:,6),2)
mod(result(:,7),2) mod(result(:,8),2) mod(result(:,9),2)
mod(result(:,10),2) mod(result(:,11),2) mod(result(:,12),2)
mod(result(:,13),2) mod(result(:,14),2) mod(result(:,15),2)
mod(result(:,16),2) mod(result(:,17),2) mod(result(:,18),2)
mod(result(:,19),2) mod(result(:,20),2) mod(result(:,21),2)
mod(result(:,22),2) mod(result(:,23),2) mod(result(:,24),2)
mod(result(:,25),2)];
    case 14
        message1= [m1 m2 m3 m4 m5 m6 m7 m8 m9 m10 m11 m12 m13 m14];
        message2= [n1 n2 n3 n4 n5 n6 n7 n8 n9 n10 n11 n12 n13 n14];
result=conv(message1,message2);
hasil=[mod(result(:,1),2) mod(result(:,2),2) mod(result(:,3),2)
mod(result(:,4),2) mod(result(:,5),2) mod(result(:,6),2)
mod(result(:,7),2) mod(result(:,8),2) mod(result(:,9),2)
mod(result(:,10),2) mod(result(:,11),2) mod(result(:,12),2)
mod(result(:,13),2) mod(result(:,14),2) mod(result(:,15),2)
mod(result(:,16),2) mod(result(:,17),2) mod(result(:,18),2)
mod(result(:,19),2) mod(result(:,20),2) mod(result(:,21),2)
mod(result(:,22),2) mod(result(:,23),2) mod(result(:,24),2)
mod(result(:,25),2) mod(result(:,26),2) mod(result(:,27),2)];
    case 15
        message1= [m1 m2 m3 m4 m5 m6 m7 m8 m9 m10 m11 m12 m13 m14 m15];
        message2= [n1 n2 n3 n4 n5 n6 n7 n8 n9 n10 n11 n12 n13 n14 n15];
result=conv(message1,message2);
hasil=[mod(result(:,1),2) mod(result(:,2),2) mod(result(:,3),2)
mod(result(:,4),2) mod(result(:,5),2) mod(result(:,6),2)
mod(result(:,7),2) mod(result(:,8),2) mod(result(:,9),2)
mod(result(:,10),2) mod(result(:,11),2) mod(result(:,12),2)
mod(result(:,13),2) mod(result(:,14),2) mod(result(:,15),2)
mod(result(:,16),2) mod(result(:,17),2) mod(result(:,18),2)
mod(result(:,19),2) mod(result(:,20),2) mod(result(:,21),2)
mod(result(:,22),2) mod(result(:,23),2) mod(result(:,24),2)
mod(result(:,25),2) mod(result(:,26),2) mod(result(:,27),2)
mod(result(:,28),2) mod(result(:,29),2)];
    case 16
        message1= [m1 m2 m3 m4 m5 m6 m7 m8 m9 m10 m11 m12 m13 m14 m15
m16];
        message2= [n1 n2 n3 n4 n5 n6 n7 n8 n9 n10 n11 n12 n13 n14 n15
n16];
result=conv(message1,message2);
hasil=[mod(result(:,1),2) mod(result(:,2),2) mod(result(:,3),2)
mod(result(:,4),2) mod(result(:,5),2) mod(result(:,6),2)
mod(result(:,7),2) mod(result(:,8),2) mod(result(:,9),2)
mod(result(:,10),2) mod(result(:,11),2) mod(result(:,12),2)
mod(result(:,13),2) mod(result(:,14),2) mod(result(:,15),2)
mod(result(:,16),2) mod(result(:,17),2) mod(result(:,18),2)
mod(result(:,19),2) mod(result(:,20),2) mod(result(:,21),2)
```

```
mod(result(:,22),2) mod(result(:,23),2) mod(result(:,24),2)
mod(result(:,25),2) mod(result(:,26),2) mod(result(:,27),2)
mod(result(:,28),2) mod(result(:,29),2) mod(result(:,30),2)
mod(result(:,31),2)];

end
while hasil(:,1)==0
    hasil(:,1)=[];
end
while pfx(:,1)==0
    pfx(:,1)=[];
end
[hasilbagi,sisa]=deconv(hasil,pfx);
while sisa(:,1)==0
    sisa(:,1)=[];
end

nn=length(sisa);
for zx =1:nn
    switch zx
    case 1
        sa1=mod(sisa(:,1),2); sa2=num2str(sa1);
    case 2
        sb1=mod(sisa(:,2),2); sb2=num2str(sb1);
    case 3
        sc1=mod(sisa(:,3),2); sc2=num2str(sc1);
    case 4
        sd1=mod(sisa(:,4),2); sd2=num2str(sd1);
    case 5
        se1=mod(sisa(:,5),2); se2=num2str(se1);
    case 6
        sf1=mod(sisa(:,6),2); sf2=num2str(sf1);
    case 7
        sg1=mod(sisa(:,7),2); sg2=num2str(sg1);
    case 8
        sh1=mod(sisa(:,8),2); sh2=num2str(sh1);
    case 9
        si1=mod(sisa(:,9),2); si2=num2str(si1);
    case 10
        sj1=mod(sisa(:,10),2); sj2=num2str(sj1);
    case 11
        sk1=mod(sisa(:,11),2); sk2=num2str(sk1);
    case 12
        sl1=mod(sisa(:,12),2); sl2=num2str(sl1);
    case 13
        sm1=mod(sisa(:,13),2); sm2=num2str(sm1);
    case 14
        sn1=mod(sisa(:,14),2); sn2=num2str(sn1);
    case 15
        so1=mod(sisa(:,15),2); so2=num2str(so1);
    case 16
        sp1=mod(sisa(:,16),2); sp2=num2str(sp1);
    end
end
switch nn
    case 1
        binsisa=strcat(sa2);
    case 2
        binsisa=strcat(sa2,sb2);
    case 3
        binsisa=strcat(sa2,sb2,sc2);
    case 4
        binsisa=strcat(sa2,sb2,sc2,sd2);
```

```
    case 5
        binsisa=strcat(sa2,sb2,sc2,sd2,se2);
    case 6
        binsisa=strcat(sa2,sb2,sc2,sd2,se2,sf2);
    case 7
        binsisa=strcat(sa2,sb2,sc2,sd2,se2,sf2,sg2);
    case 8
        binsisa=strcat(sa2,sb2,sc2,sd2,se2,sf2,sg2,sh2);
    case 9
        binsisa=strcat(sa2,sb2,sc2,sd2,se2,sf2,sg2,sh2,si2);
    case 10
        binsisa=strcat(sa2,sb2,sc2,sd2,se2,sf2,sg2,sh2,si2,sj2);
    case 11
        binsisa=strcat(sa2,sb2,sc2,sd2,se2,sf2,sg2,sh2,si2,sj2,sk2);
    case 12
        binsisa=strcat(sa2,sb2,sc2,sd2,se2,sf2,sg2,sh2,si2,sj2,sk2,sl2);
    case 13

binsisa=strcat(sa2,sb2,sc2,sd2,se2,sf2,sg2,sh2,si2,sj2,sk2,sl2,sm2);
    case 14

binsisa=strcat(sa2,sb2,sc2,sd2,se2,sf2,sg2,sh2,si2,sj2,sk2,sl2,sm2,sn2);
    case 15

binsisa=strcat(sa2,sb2,sc2,sd2,se2,sf2,sg2,sh2,si2,sj2,sk2,sl2,sm2,sn2,so
2);
    case 16

binsisa=strcat(sa2,sb2,sc2,sd2,se2,sf2,sg2,sh2,si2,sj2,sk2,sl2,sm2,sn2,so
2,sp2);
end


dessisa=bin2dec(binsisa);
betha=dec2bin(dessisa,deg);

in1=dessisa;

end
```

Source Code in1xin2.m

```
Source Code m-file ini sama dengan perpangkatan.m
```

Source Code primitifcek.m

```
clc

disp(header)
disp('  ')
disp('   <<<Tes Elemen Primitif>>>')
disp('  ')

pfx=input('masukkan irreducible polinomial = ');
deg=length(pfx)-1;
power=(2^deg)-1;
g=input('polinomial yang ingin ditest = ');

bentukbiner=g;
converttodecimal;
desimalg=bentukdesimal;
```

```
in1=desimalg;
in2=in1;
disp('sedang diproses...')

orderalfa

disp(' ')
ttt=input('tekan 1 untuk melanjutkan atau sebarang angka untuk keluar');
switch ttt
    case 1
        elgamal
    otherwise
        clc
end
```

Source Code `orderalfa.m`

```
for i=1:power-1
dessisa= in1;
ii=i+1;
binery1=dec2bin(in1,deg);
binery2=dec2bin(in2,deg);

perpangkatan

if dessisa==1
            disp(['order dari elemen ini adalah = ' num2str(ii)])

            if ii==power
            disp('<<<elemen ini adalah generator>>>')
            else
            disp('<<<elemen ini bukan generator>>>')
            end
            break
        end
```

Lampiran 2 : Kode ASCII (desimal-biner-kode)

Kode ASCII 0-127

0	00000000	NUL	32	00100000		64	01000000	@	96	01100000	`
1	00000001	SOH	33	00100001	!	65	01000001	A	97	01100001	a
2	00000010	STX	34	00100010	"	66	01000010	B	98	01100010	b
3	00000011	ETX	35	00100011	#	67	01000011	C	99	01100011	c
4	00000100	EOT	36	00100100	$	68	01000100	D	100	01100100	d
5	00000101	ENQ	37	00100101	%	69	01000101	E	101	01100101	e
6	00000110	ACK	38	00100110	&	70	01000110	F	102	01100110	f
7	00000111	BEL	39	00100111	'	71	01000111	G	103	01100111	g
8	00001000	BS	40	00101000	(	72	01001000	H	104	01101000	h
9	00001001	HT	41	00101001	)	73	01001001	I	105	01101001	i
10	00001010	LF	42	00101010	*	74	01001010	J	106	01101010	j
11	00001011	VT	43	00101011	+	75	01001011	K	107	01101011	k
12	00001100	FF	44	00101100	,	76	01001100	L	108	01101100	l
13	00001101	CR	45	00101101	-	77	01001101	M	109	01101101	m
14	00001110	SO	46	00101110	.	78	01001110	N	110	01101110	n
15	00001111	SI	47	00101111	/	79	01001111	O	111	01101111	o
16	00010000	DLE	48	00110000	0	80	01010000	P	112	01110000	p
17	00010001	DC1	49	00110001	1	81	01010001	Q	113	01110001	q
18	00010010	DC2	50	00110010	2	82	01010010	R	114	01110010	r
19	00010011	DC3	51	00110011	3	83	01010011	S	115	01110011	s
20	00010100	DC4	52	00110100	4	84	01010100	T	116	01110100	t
21	00010101	NAK	53	00110101	5	85	01010101	U	117	01110101	u
22	00010110	SYN	54	00110110	6	86	01010110	V	118	01110110	v
23	00010111	ETB	55	00110111	7	87	01010111	W	119	01110111	w
24	00011000	CAN	56	00111000	8	88	01011000	X	120	01111000	x
25	00011001	EM	57	00111001	9	89	01011001	Y	121	01111001	y
26	00011010	SUB	58	00111010	:	90	01011010	Z	122	01111010	z
27	00011011	ESC	59	00111011	;	91	01011011	[	123	01111011	{
28	00011100	FS	60	00111100	<	92	01011100	\	124	01111100	|
29	00011101	GS	61	00111101	=	93	01011101	]	125	01111101	}
30	00011110	RS	62	00111110	>	94	01011110	^	126	01111110	~
31	00011111	US	63	00111111	?	95	01011111	_	127	01111111	

Kode ASCII *Extended* (128-255)

128	10000000	€	160	10100000		192	11000000	À	224	11100000	à
129	10000001		161	10100001	¡	193	11000001	Á	225	11100001	á
130	10000010	‚	162	10100010	¢	194	11000010	Â	226	11100010	â
131	10000011	ƒ	163	10100011	£	195	11000011	Ã	227	11100011	ã
132	10000100	„	164	10100100	¤	196	11000100	Ä	228	11100100	ä
133	10000101	…	165	10100101	¥	197	11000101	Å	229	11100101	å
134	10000110	†	166	10100110	¦	198	11000110	Æ	230	11100110	æ
135	10000111	‡	167	10100111	§	199	11000111	Ç	231	11100111	ç
136	10001000	ˆ	168	10101000	¨	200	11001000	È	232	11101000	è
137	10001001	‰	169	10101001	©	201	11001001	É	233	11101001	é
138	10001010	Š	170	10101010	ª	202	11001010	Ê	234	11101010	ê
139	10001011	‹	171	10101011	«	203	11001011	Ë	235	11101011	ë
140	10001100	Œ	172	10101100	¬	204	11001100	Ì	236	11101100	ì
141	10001101		173	10101101		205	11001101	Í	237	11101101	í
142	10001110	Ž	174	10101110	®	206	11001110	Î	238	11101110	î
143	10001111		175	10101111	¯	207	11001111	Ï	239	11101111	ï
144	10010000		176	10110000	°	208	11010000	Ð	240	11110000	ð
145	10010001	‘	177	10110001	±	209	11010001	Ñ	241	11110001	ñ
146	10010010	’	178	10110010	²	210	11010010	Ò	242	11110010	ò
147	10010011	“	179	10110011	³	211	11010011	Ó	243	11110011	ó
148	10010100	”	180	10110100	´	212	11010100	Ô	244	11110100	ô
149	10010101	•	181	10110101	µ	213	11010101	Õ	245	11110101	õ
150	10010110	–	182	10110110	¶	214	11010110	Ö	246	11110110	ö
151	10010111	—	183	10110111	·	215	11010111	×	247	11110111	÷
152	10011000	˜	184	10111000	¸	216	11011000	Ø	248	11111000	ø
153	10011001	™	185	10111001	¹	217	11011001	Ù	249	11111001	ù
154	10011010	š	186	10111010	º	218	11011010	Ú	250	11111010	ú
155	10011011	›	187	10111011	»	219	11011011	Û	251	11111011	û
156	10011100	œ	188	10111100	¼	220	11011100	Ü	252	11111100	ü
157	10011101		189	10111101	½	221	11011101	Ý	253	11111101	ý
158	10011110	ž	190	10111110	¾	222	11011110	Þ	254	11111110	þ
159	10011111	Ÿ	191	10111111	¿	223	11011111	ß	255	11111111	ÿ

Lampiran 3 : Curiculum Vitae

Curriculum Vitae

Nama	: Najib Mubarok
Fak/prodi	: Sains dan Teknologi/ Matematika 2008
TTL	: Temanggung, 03 Januari 1990
Golongan darah	: O
No. HP	: 085729085720
Alamat asal	: Talun RT 01/01, Gunungsari, Bansari, Temanggung, Jawa Tengah.
Alamat Jogja	: Jl. Wahid Hasyim n0.3
Nama orang tua	: Muh. Dahlan / Hidayah
Email	: sleepygoblin@gmail.com
Motto hidup	: "*Hidup bukan tentang apa yang bisa kita dapatkan, melainkan apa yang bisa kita berikan"*

Riwayat Pendidikan

Nama Sekolah	Tahun
SD Negeri Gunungsari	1996 – 2002
MTs N Model Parakan	2002 – 2005
SMK N 1 Magelang	2005 – 2008
UIN Sunan Kalijaga	2008 – 2012

Pengalaman Organisasi

Nama Organisasi	Tahun	Jabatan
PSPB (Pusat Studi dan Pengembangan Bahasa) PP Wahid Hasyim	2009-2010 2010-2011 2011-2012	Divisi Pendidikan Sekretaris Ketua

Riwayat Pekerjaan

Nama Pekerjaan	Tahun
Guru / Pengajar di Madrasah Aliyah PP Wahid Hasyim	2011-2013