PENGODEAN TEKS BERDASARKAN ALGORITMA *CIPHER BLOCK CHAINING* (CBC) DAN PEWARNAAN TITIK PADA GRAF

Skripsi

Oleh

Asiil Rahma Sahrani

NIM 181810101065

JURUSAN MATEMATIKA

FAKULTAS MATEMATIKA DAN ILMU PENGETAHUAN ALAM

UNIVERSITAS JEMBER

2022

PENGODEAN TEKS BERDASARKAN ALGORITMA *CIPHER BLOCK CHAINING* (CBC) DAN PEWARNAAN TITIK PADA GRAF

SKRIPSI

Disusun guna melengkapi tugas akhir dan memenuhi salah satu syarat
untuk menyelesaikan Program Studi Matematika (S1)
dan mencapai gelar Sarjana Sains

Oleh
Asiil Rahma Sahrani
181810101065

JURUSAN MATEMATIKA
FAKULTAS MATEMATIKA DAN ILMU PENGETAHUAN ALAM
UNIVERSITAS JEMBER
2022

PERSEMBAHAN

Skripsi ini saya persembahkan untuk :

1. Diri saya sendiri yang mampu bertahan hingga sejauh ini;
2. Ayah, Ibu, dan keluarga yang senantiasa mendukung dan mendoakan yang terbaik;
3. Dr. Kiswara Agung Santoso, S.Si., M.Kom. selaku Dosen Pembimbing Utama dan Ikhsanul Halikin, S.Pd, M.Si. selaku Dosen Pembimbing Anggota yang telah membimbing dengan sepenuh hati;
4. Guru-guru dan dosen sejak TK hingga perguruan tinggi yang telah memberi ilmu dan pengalaman;
5. Muhammad Rafi yang selalu memberi motivasi dan semangat selama pengerjaan skripsi;
6. Teman-teman ARITHMETIC'18 yang telah memberi doa dan semangat;
7. Almamater Jurusan Matematika Fakultas Matematika dan Ilmu Pengetahuan Alam Universitas Jember.

MOTO

"Boleh jadi kamu tidak menyenangi sesuatu, padahal itu baik bagimu, dan boleh jadi kamu menyukai sesuatu, padahal itu tidak baik bagimu. Allah mengetahui, sedang kamu tidak mengetahui."

(QS Al-Baqarah: 216)[1]

[1] Departemen Agama Republik Indonesia. 2010. *Mushaf Aisyah Al-Qur'an dan Terjemahan untuk Wanita*. Bandung: Jabal.

PERNYATAAN

Saya yang bertanda tangan di bawah ini

Nama : Asiil Rahma Sahrani

NIM : 181810101065

menyatakan dengan sesungguhnya bahwa karya ilmiah yang berjudul "Pengodean Teks Berdasarkan Algoritma *Cipher Block Chaining* (CBC) dan Pewarnaan Titik pada Graf" adalah benar-benar hasil karya ilmiah sendiri, kecuali jika dalam pengutipan substansi disebutkan sumbernya dan belum pernah diajukan pada institusi manapun, serta bukan karya jiplakan. Saya bertanggung jawab atas keabsahan dan kebenaran isinya sesuai dengan sikap ilmiah yang harus dijunjung tinggi.

Demikian pernyataan ini saya buat dengan sebenarnya, tanpa ada tekanan dan paksaan dari pihak manapun serta bersedia mendapat sanksi akademik jika ternyata di kemudian hari pernyataan ini tidak benar.

Jember, 19 Desember 2022

Yang menyatakan,

Asiil Rahma Sahrani

NIM 181810101065

SKRIPSI

PENGODEAN TEKS BERDASARKAN ALGORITMA *CIPHER BLOCK CHAINING* (CBC) DAN PEWARNAAN TITIK PADA GRAF

Oleh

Asiil Rahma Sahrani

181810101065

Pembimbing

Dosen Pembimbing Utama : Dr. Kiswara Agung Santoso, S.Si., M.Kom.

Dosen Pembimbing Anggota : Ikhsanul Halikin, S.Pd, M.Si.

PENGESAHAN

Skripsi berjudul "Pengodean Teks Berdasarkan Algoritma *Cipher Block Chaining* (CBC) dan Pewarnaan Titik pada Graf", telah diuji dan disahkan pada:

hari, tanggal :

tempat : Fakultas Matematika dan Ilmu Pengetahuan Alam
Universitas Jember

Tim Penguji:

Ketua,

Dr. Kiswara Agung Santoso, S.Si., M.Kom.
NIP 197209071998031003

Anggota I,

Ikhsanul Halikin, S.Pd., M.Si.
NIP 198610142014041001

Anggota II,

Dr. Kristiana Wijaya, S.Si., M.Si.
NIP 197408132000032004

Anggota III,

Abduh Riski, S.Si., M.Si.
NIP 199004062015041001

Mengesahkan
Dekan Fakultas Matematika dan Ilmu Pengetahuan Alam
Universitas Jember

Drs. Achmad Sjaifullah, M.Sc., Ph.D.
NIP 195910091986021001

RINGKASAN

Pengodean Teks Berdasarkan Algoritma Cipher Block Chaining (CBC) dan Pewarnaan Titik pada Graf; Asiil Rahma Sahrani; 181810101065; 60 halaman; Jurusan Matematika Fakultas Matematika dan Ilmu Pengetahuan Alam Universitas Jember.

Kemajuan teknologi dan informasi saat ini semakin canggih, seperti pada pengiriman pesan yang dapat dilakukan dimana saja dan kapan saja. Masalah keamanan data merupakan salah satu aspek terpenting pada sistem teknologi dan informasi. Keamanan data pengiriman pesan yang seharusnya ditujukan kepada seseorang atau suatu golongan tertentu dan bersifat rahasia bisa saja jatuh pada pihak yang tidak berwenang sehingga pihak tersebut dapat melakukan penyalahgunaan data yang akan menimbulkan kerugian bagi seseorang maupun golongan tertentu.

Kriptografi merupakan solusi untuk pengamanan suatu data. Fungsi utama dari kriptografi adalah menjaga kerahasiaan data maupun kunci sehingga tidak terjadi tindak kejahatan seperti kebocoran data sehingga mengakibatkan penyalahgunaan data oleh pihak yang tidak berwenang. Untuk menjaga kerahasiaan data, proses transformasi data dapat dilakukan yaitu, dengan mengubah data asli atau data yang mudah dibaca (plainteks*)* menjadi data yang sulit dikenali (cipherteks)*,* setelah data diubah menjadi cipherteks selanjutnya data akan ditransformasikan menjadi plainteks sehingga data dapat dibaca dengan jelas oleh penerima, proses ini merupakan alur pengiriman data dari pengirim pada penerima. Kriptografi diklasifikasi menjadi dua macam, yaitu kriptografi klasik dan kriptografi modern. Kriptografi modern memiliki berbagai jenis, salah satunya adalah *Cipher Block Chaining*.

Kunci pada kriptografi memiliki peran penting dalam keamanan suatu data. Pada penelitian ini kunci yang akan digunakan berupa graf. Pewarnaan graf menggunakan algoritma *Welch Powell* untuk menghitung bilangan kromatik pada graf yang digunakan. Teks yang akan dikodekan di-XOR-kan dengan kunci graf

yang telah ditentukan sehingga akan menghasilkan sebuah pesan yang sulit dibaca (cipherteks). Penelitian ini bertujuan untuk mengodekan teks berdasarkan algoritma CBC dan pewarnaan titik pada graf. Hasil dari penelitian ini adalah pengodean teks berdasarkan algoritma CBC dan Pewarnaan titik pada graf. Program pengodean teks menggunakan *software* Matlab 2009a dengan memasukkan teks yang akan dikodekan (plainteks) dan kunci yang telah ditentukan kedalam program dan menghasilkan suatu pesan yang telah dikodekan (cipherteks). Sehingga didapat hasil dari pengodean teks adalah pesan yang telah dikodekan berdasarkan algoritma CBC dan pewarnaan titik pada graf.

PRAKATA

Puji syukur kepada Allah SWT atas segala rahmat dan karunia-Nya sehingga penulis dapat menyelesaikan skripsi dengan judul "Pengodean Teks Berdasarkan Algoritma CBC dan Pewarnaan Titik pada Graf". Skripsi ini disusun untuk memenuhi salah satu syarat dalam menyelesaikan pendidikan strata satu (S1) pada Jurusan Matematika Fakultas Matematika dan Ilmu Pengetahuan Alam Universitas Jember.

Penyusunan skripsi ini tidak lepas dari bantuan berbagai pihak. Oleh karena itu, penulis menyampaikan terima kasih kepada:

1. Dr. Kiswara Agung Santoso, S.Si., M.Kom. selaku Dosen Pembimbing Utama dan Ikhsanul Halikin, S.Pd., M.Si.selaku Dosen Pembimbing Anggota yang telah membimbing dan mengarahkan penulis dalam menyelesaikan penulisan skripsi ini;
2. Dr. Kristiana Wijaya, S.Si., M.Si. selaku Dosen Penguji I dan Abduh Riski, S.Si., M.Si. selaku Dosen Penguji II yang telah memberikan kritik, saran, dan masukan yang membangun demi kesempurnaan skripsi ini;
3. Ahmad Kamsyakawuni, S.Si., M.Si. selaku Dosen Pembimbing Akademik yang telah membimbing saya dari awal masa perkuliahan.
4. Teman-teman ARITHMETIC'18 yang telah memberi doa dan semangat kepada penulis;
5. Semua pihak yang telah memberikan semangat, dukungan dan bantuan baik secara moral, spiritual, maupun materi.

Penulis juga menerima segala kritik dan saran dari semua pihak demi kesempurnaan skripsi ini. Penulis berharap, semoga skripsi ini dapat bermanfaat.

Jember, 19 Desember 2022

Penulis

DAFTAR ISI

DAFTAR TABEL

DAFTAR GAMBAR

DAFTAR LAMPIRAN

BAB 1. PENDAHULUAN

1.1 Latar Belakang

Kemajuan teknologi dan informasi saat ini sudah semakin canggih, seperti pada pengiriman pesan yang dapat dilakukan kapan saja dengan praktis. Masalah keamanan data merupakan salah satu aspek terpenting pada sistem teknologi dan informasi. Keamanan data pengiriman pesan yang seharusnya ditujukan kepada seseorang atau suatu golongan tertentu dan bersifat rahasia bisa saja jatuh pada pihak yang tidak berwenang sehingga pihak tersebut dapat melakukan penyalahgunaan data yang akan menimbulkan kerugian bagi seseorang maupun golongan tertentu.

Kriptografi merupakan solusi untuk pengamanan suatu data. Fungsi utama dari kriptografi adalah menjaga kerahasiaan data maupun kunci sehingga tidak terjadi tindak kejahatan seperti kebocoran data sehingga mengakibatkan penyalahgunaan data oleh pihak yang tidak berwenang. Untuk menjaga kerahasiaan data, proses transformasi data dapat dilakukan yaitu, dengan mengubah data asli atau data yang mudah dibaca (plainteks*)* menjadi data yang sulit dikenali (cipherteks)*,* setelah data diubah menjadi cipherteks selanjutnya data akan ditransformasikan menjadi plainteks sehingga data dapat dibaca dengan jelas oleh penerima, proses ini merupakan alur pengiriman data dari pengirim pada penerima. Kriptografi diklasifikasi menjadi dua macam, yaitu kriptografi klasik dan kriptografi modern. Kriptografi modern memiliki berbagai jenis, salah satunya adalah *Cipher Block Chaining*.

Beberapa penelitian yang berkaitan dengan algoritma CBC yaitu dari Andriani (2017) yang berjudul Perancangan Penyandian Teks menggunakan Algoritma *Cipher Block Chaining*. Pada penelitian tersebut, meneliti tentang proses perancangan suatu aplikasi tujuan mengimplementasikan enkripsi dan dekripsi teks berdasarkan algoritma CBC dengan tujuan dapat memperlihatkan prosedur dan hasil yang didapatkan untuk mengamankan pesan teks khususnya prosedur yang dilakukan pada kegiatan penyandian dan pengembalian teks ke bentuk semula. Algoritma CBC dapat dikombinasikan dengan graf yang dicari bilangan

kromatiknya dari pewarnaan titik suatu graf. Graf digunakan sebagai kunci untuk proses enkripsi dan dekripsi. Penelitian yang berkaitan dengan pewarnaan titik pada graf yaitu penelitian oleh Santoso (2020) yang berjudul *The Modification of Caesar Cryptosystem Based on Binary Vertices Colouring*, meneliti tentang proses enkripsi dan dekripsi pada algoritma *Caesar Cipher* yang dimodifikasi dengan kunci yang didapat dari pewarnaan titik pada graf, dan penelitian lain yang berkaitan dengan hal tersebut yaitu dari Santoso (2019) berjudul *Vertex Colouring Using The Adjacency Matrix.* Hasil dari penelitian tersebut adalah pewarnaan titik pada suatu graf untuk menentukan bilangan kromatik suatu graf yang diubah menjadi matriks *adjacency*.

Berdasarkan beberapa penelitian diatas, penulis tertarik untuk melakukan penelitian mengenai pengodean teks berdasarkan algoritma CBC dan pewarnaan titik pada graf karena belum ada penelitian yang meneliti tentang CBC dan pewarnaan titik pada graf. Algoritma pewarnaan titik yang akan digunakan yaitu menggunakan *Welch Powell*.

1.2 Rumusan Masalah

Rumusan masalah dalam penelitian ini adalah bagaimana mengodekan teks berdasarkan algoritma CBC dan pewarnaan titik pada graf

1.3 Batasan Masalah

Pada penelitian ini graf yang akan digunakan sebagai kunci untuk pengodean teks menggunakan algoritma CBC adalah graf sederhana.

1.4 Tujuan Penelitian

Tujuan dari penelitian ini adalah mengodekan teks berdasarkan algoritma CBC dan pewarnaan titik pada graf.

1.5 Manfaat Penelitian

Manfaat dari penelitian ini adalah menambah wawasan terkait dengan pengodean algoritma CBC dan pewarnaan titik pada graf.

BAB 2. LANDASAN TEORI

2.1 Definisi dan Terminologi Graf

Graf adalah struktur diskrit yang terdiri dari sejumlah himpunan berhingga objek yang disebut simpul atau titik (*vertex*) dan himpunan sisi (*edge*) yang menghubungkan titik-titik tersebut. Graf digunakan untuk merepresentasikan objek-objek diskrit dan hubungan antara suatu objek-objek tersebut. Notasi untuk sebuah graf adalah $G = (V, E)$, dimana V merupakan himpunan tak kosong dari titik-titik (*vertices*), E merupakan himpunan dari sisi-sisi (*edges*).

Graf dikategorikan sebagai graf berarah dan graf tidak berarah. Graf berarah (*directed graph*) merupakan graf yang mempunyai sisi yang berarah, dimana satu titik yang dihubungkan oleh sisi tersebut merupakan titik awal (*initial vertex*) dan titik yang lain sebagai titik akhir (*terminal vertex*). Sedangkan graf tidak berarah merupakan graf yang sisinya tidak mempunyai orientasi arah (Adiwijaya, 2016).

Adiwijaya (2016) menyebutkan terdapat beberapa terminologi graf dasar yang perlu diketahui. Berikut ini adalah beberapa terminologi yang penting diantaranya yaitu :

1. Bertetangga

   Dua buah titik v_j dan v_k pada suatu graf G dikatakan bertetangga jika kedua titik terhubung langsung oleh suatu sisi e. Contoh dari titik yang bertetangga atau *adjacency* dapat dilihat pada Gambar 2.1, titik v_1 dan v_2 bertetangga, tetapi pada titik v_1 dan v_3 tidak bertetangga.

2. Bersisian

   Suatu sisi e dikatakan bersisian atau *incidency* dengan titik v_j dan v_k jika e menghubungkan kedua titik tersebut, dengan kata lain $e = (v_j, v_k)$. Contoh dari sisi yang bersisian dapat dilihat pada Gambar 2.2, pada sisi e_1 bersisian dengan titik v_1 dan v_2, tetapi pada sisi tersebut tidak terhubung dengan titik v_6. Sehingga untuk titik v_6 dan v_2 bersisian dengan sisi e_5.

3. Titik terpencil

Titik terpencil merupakan titik yang tidak mempunyai sisi yang bersisian dengan titik tersebut. Contoh dari simpul terpencil dapat dilihat pada Gambar 2.2, pada titik v_5 merupakan simpul terpencil atau *isolated vertex* karena tidak mempunyai sisi yang bersisian pada titik tersebut.

4. Derajat

Derajat suatu titik v merupakan banyaknya sisi e yang bersisian pada titik tersebut dan jika bersisian pada sisi gelang (*loop*) maka dihitung dua kali. Notasi untuk derajat atau *degree* suatu titik adalah $d(v)$. Seperti contoh pada Gambar 2.2 derajat untuk setiap titiknya yaitu :

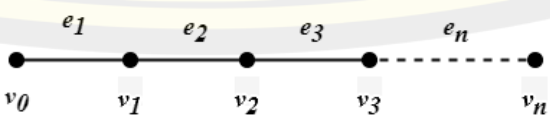
$$d(v_1) = d(v_3) = 3, \text{ sedangkan } d(v_2) = d(v_6) = 5.$$

5. Lintasan

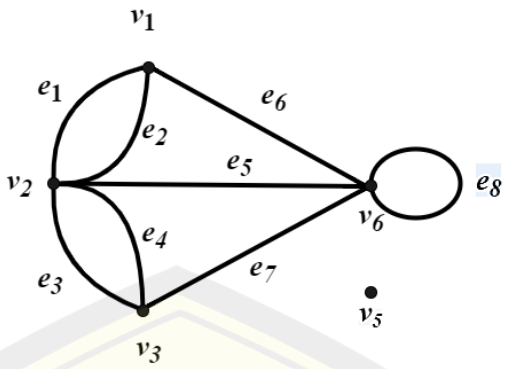
Lintasan dengan panjang *n* dari titik v_0 ke titik v_n pada suatu graf G adalah barisan berselang seling titik dan sisiyang diawali dan diakhiri dengan titik yang terbentuk $v_0, e_1, v_1, e_2, v_2, \dots, v_{n-1}, e_n, v_n$ sedemikian sehingga $e_i = (v_{i-1}, v_i)$ untuk $i = 1,2, \dots$, Lintasan ini dapat dituliskan dengan :

$$v_0, v_1, v_2, \dots, v_n$$

Contoh lintasan dari v_0 ke v_n dapat dilihat pada Gambar 2.1. Lintasan ini memuat *n* buah sisi yang dilewati oleh suatu titik awal v_0 ke titik akhir v_n dalam suatu graf G. Lintasan yang berawal dan berakhir pada titik yang sama dinamakan siklus atau sirkuit**.** Contoh lintasan terdapat pada Gambar 2.2, lintasan v_1, v_6, v_2 memiliki panjang 2, sedangkan pada lintasan v_1, v_6, v_3, v_2 memiliki panjang 3. Pada lintasan v_1, v_6, v_3, v_2, v_1 merupakan siklus atau sirkuit. Untuk titik v_6 dan v_5 tidak dapat ditemukan lintasan.



Gambar 2.1 Contoh lintasan pada graf



Gambar 2.2 Contoh terminologi graf

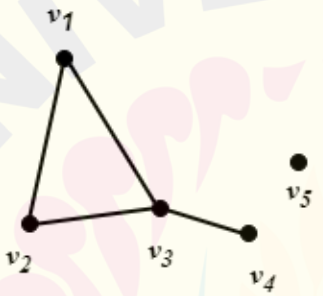
2.2 Representasi Graf dalam Matriks

Menurut Lipson (1992) matriks dapat digunakan untuk merepresentasikan suatu graf. Perhitungan bilangan kromatik pada graf akan lebih sederhana jika dinyatakan kedalam bentuk matriks dan memudahkan proses dengan pemrograman pada komputer. Representasi graf dalam matriks terdapat dua macam yaitu matriks bersisian dan matriks ketetanggan. Matriks bersisian merepresentasikan hubungan antar sisi dan titik dari graf sedangkan matrik ketetanggan merepresentasikan hubungan antar titik dari graf.

Matriks ketetanggan atau matriks *adjacency* merupakan representasi graf ke dalam bentuk matriks suatu graf sederhana G yang mempunyai n titik dengan $n \geq 1$ membentuk matriks dengan a_{ij} adalah jumlah sisi yang menghubungkan titik v_i dengan titik v_j dengan $i, j = 1,2, \dots, n$. Karena jumlah sisi yang menghubungkan titik v_i dengan v_j selalu sama dengan jumlah garis yang menghubungkan titik v_j dengan titik v_i, maka jelas bahwa matriks ketetanggaan selalu merupakan matriks yang simetris (Siang, 2002). Unsur dari matriks *adjacency* hanya terdiri dari dua bilangan yaitu 0 dan 1, jika titik v_i dengan v_j bertetangga maka elemen matriks akan berisi 1, namun jika tidak bertangga maka berisi 0. Baris dan kolom pada matriks ini masing-masing merupakan representasi dari setiap titik pada graf tersebut. Jika matriks dinamakan $A = [a_{ij}]$, dapat dilihat pada rumus 2.1.

$$[\,a_{ij}] = \begin{cases} 1, & \text{titik } i \text{ dan } j \text{ bertetangga} \\ 0, & \text{lainnya} \end{cases} \qquad (2.1)$$

Munir (2010) menyatakan karena matriks ketetanggan hanya berisikan 0 dan 1, maka matriks tersebut dinamakan juga dengan matriks nol-satu (*zero-one*). Elemen matriks dapat dinyatakan juga dengan nilai *false* (menyatakan 0) dan nilai *true* (menyatakan 1). Untuk graf berarah, matriks ketetanggaan belum tentu simetri sehingga akan menjadi simetri jika itu adalah grafik berarah lengkap. Berikut merupakan contoh representasi graf dalam bentuk matriks pada Gambar 2.3.



(a)

$$A = \begin{array}{c} \\ v_1 \\ v_2 \\ v_3 \\ v_4 \\ v_5 \end{array} \begin{array}{c} \begin{array}{ccccc} v_1 & v_2 & v_3 & v_4 & v_5 \end{array} \\ \begin{bmatrix} 0 & 1 & 1 & 0 & 0 \\ 1 & 0 & 1 & 0 & 0 \\ 1 & 1 & 0 & 1 & 0 \\ 0 & 0 & 1 & 0 & 0 \\ 0 & 0 & 0 & 0 & 0 \end{bmatrix} \end{array}$$

(b)

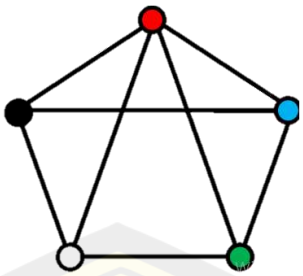
Gambar 2.3 (a) Graf (b) Representasi graf dalam matriks ketetanggan

2.3 Pewarnaan Graf

Pewarnaan pada graf dibedakan menjadi tiga, yaitu pewarnaan titik, pewwarnaan sisi, dan pewarnaan wilayah.

1. Pewarnaan titik (*vertex coloring*)

   Pewarnaan titik merupakan pemberian warna pada setiap titik yang ada di dalam suatu graf sedemikian sehingga setiap dua titik yang bertetanggaan memiliki warna yang berbeda (Siregar, 2018). Dalam pewarnaan suatu graf terdapat bilangan kromatik sebagai jumlah warna minimum yang dapat digunakan untuk mewarnai titik pada graf (Siregar, 2018). Bilangan kromatik pada graf G disimbolkan dengan $\chi(G)$, jika suatu graf G yang mempunyai bilangan kromatik k dilambangkan dengan $\chi(G) = k$ (Siregar, 2018). Contoh pewarnaan titik dapat dilihat pada Gambar 2.4.



Gambar 2. 4 Graf pewarnaan titik

2. Pewarnaan sisi (*edge coloring*).

   Pewarnaan sisi merupakan memberikan warna pada sisi-sisi dalam suatu graf sedemikian sehingga pada sisi-sisi yang berdekatan memiliki warna yang berbeda.

3. Pewarnaan wilayah (*face coloring*)

   Pewarnaan wilayah tidak jauh berbeda dengan pewarnaan sisi, dimana wilayah yang saling berdekatan dibuat sedemikian sehingga tidak memiliki warna yang sama. Pemanfaatan pewarnaan wilayah untuk mengatur agar warna setiap wilayah yang berdekatan pada peta tidak memiliki warna yang sama. Aplikasi pewarnaan wilayah di implementasikan pada pemberian warna peta (Jeado, 2020).

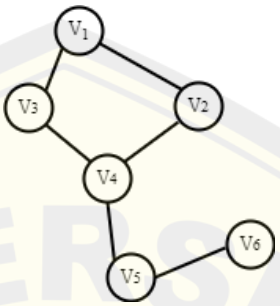
2.4 Algoritma Welch Powell

Algoritma Welch Powell digunakan untuk mewarnai titik suatu graf berdasarkan derajat tertinggi dari titik-titiknya. Menurut Danial dan Taneo (2019) Langkah-langkah dalam algoritma *Welch Powell* adalah sebagai berikut.

1. Urutkan titik-titik dari graf G dalam urutan derajat terbesar ke terkecil.
2. Gunakan satu warna tertentu untuk mewarnai titik derajat tertinggi. Secara berurut, setiap titik dalam daftar yang tidak berelasi dengan titik sebelumnya diwarnai dengan warna ini.

3. Ulangi langkah 2 di atas untuk titik dengan urutan derajat tertinggi yang belum diwarnai.
4. Ulangi langkah 3 di atas sampai semua titik dalam daftar terwarnai.

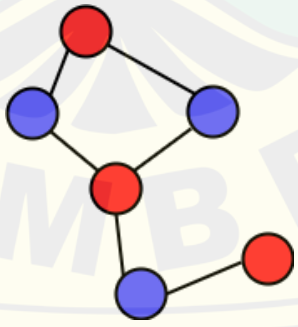
Berikut contoh mencari bilangan kromatik menggunakan algoritma *Welch powell*;



Gambar 2.5 Graf *G*

Titik	v_4	v_1	v_2	v_3	v_5	v_6
Derajat	3	2	2	2	2	1
Warna	*a*	*a*	*b*	*b*	*b*	*a*

Jadi, hasil yang didapat dari graf *G* diatas adalah 2 jenis warna yaitu *a* dan *b*, dengan dimisalkan bahwa warna *a* adalah warna merah dan *b* adalah warna biru. Sehingga bilangan kromatiknya adalah $\chi(G) = 2$. Berikut graf *G* yang telah diwarnai dapat dilihat pada Gambar 2.6.



Gambar 2.6 Graf *G* setelah pewarnaan titik

2.5 Kriptografi

Kriptografi merupakan sistem pengodean untuk mengamankan suatu hal yang bersifat rahasia dan berfungsi sebagai sistem pengamanan suatu data. Kriptografi berasal dari bahasa Yunani, *crypto* dan *graphia*. *Crypto* artinya *secret* (rahasia) dan *graphia* artinya *writing* (tulisan). Menurut terminologinya, kriptografi adalah Ilmu dan seni untuk menjaga keamanan suatu pesan ketika pesan dikirim dari suatu tempat ke tempat lain (Ariyus, 2008). Perkembangannya mengikuti arah menuju kedewasaan sesuai dengan masalah yang dihadapi setiap pengamanan data sehingga muncu beberapa istilah yang dipergunakan untuk menandai aktivitas-aktivitas rahasia untuk mengirim pesan (Mukhtar, 2018). Terdapat beberapa istilah pada komponen proses pengacakan pesan disebut enkripsi dan saat merapikan pesan teracak disebut dekripsi. Pesan awal yang belum dikodekan disebut dengan plainteks, sedangkan pesan yang telah dikodekan disebut cipherteks. *Key* merupakan kunci yang digunakan dalam proses kriptografi. Algoritma adalah metode yang akan digunakan untuk melakukan proses kriptografi. Gambar 2.7 merupakan ilustrasi proses enkripsi dan dekripsi pada kriptografi.

Gambar 2.7 Ilustrasi proses enkripsi dan dekripsi

Berdasarkan perkembangannya, kriptografi terbagi menjadi 2 macam, yaitu kriptografi klasik dan modern. Kriptografi klasik sudah digunakan sejak era komputerisasi dan kebanyakan teknik ini menggunakan kunci simetris. Metode yang digunakan yaitu subtitusi dan transposisi. Teknik subtitusi merupakan teknik yang menggantikan karakter pada plainteks menjadi karakter lain yang hasilnya adalah cipherteks dengan cara permutasi karakter. Kombinasi keduanya secara kompleks merupakan latar belakang terbentuknya berbagai macam algoritma kriptografi modern. Pada kriptografi modern memiliki tingkat kesulitan yang kompleks dan kekuatan kriptografinya ada pada *key* atau kuncinya. Algoritma ini menggunakan pengolahan simbol biner karena berjalan mengikuti operasi komputer digital.

Sehingga membutuhkan dasar berupa pengetahuan terhadap matematika untuk menguasainya.

Kunci pada kriptografi dibedakan menjadi 2 jenis, yaitu simetris dan asimetris. Algoritma simetris menggunakan kunci yang sama untuk proses enkripsi dan dekripsi nya sehingga disebut juga dengan algoritma kunci tunggal. Kunci pada algoritma ini juga bersifat *private key* dimana kunci yang digunakan bersifat rahasia dan tidak dapat diketahui orang lain. Sedangkan pada algoritma asimetris menggunakan kunci yang berbeda pada proses enkripsi dan deskripsinya. Kunci yang digunakan pada proses enkripsinya adalah *public key* yang dapat diketahui orang lain sedangkan kunci untuk deskripsinya adalah *private key* yang bersifat rahasia.

2.6 ASCII (*American Standart Code for Information Interchange*)

American Standart Code for Information Interchange atau biasa disingkat ASCII adalah suatu standar internasional kode dan simbol yang bersifat universal. Kode ASCII digunakan komputer untuk menunjukkan teks (Mahendra, 2016). Terdapat 256 kode pada ASCII yang dikelompokkan kedalam beberapa bagian. Bagian yang sering digunakan disebut ASCII Printable Character (32-127) merupakan karakter yang terdapat pada keyboard komputer yang berupa huruf, angka, dan simbol (Rahmawati, 2017).

2.7 Operasi XOR

Operasi XOR merupakan operasi *Exclusive OR* yang dilambangkan dengan tanda "$\oplus$". Operasi XOR merupakan operasi logika bitwise yang bekerja dengan membandingkan dua buah nilai (p dan q) yang apabila pada salah satu nilai nya bernilai benar, maka hasil akhir operasi XOR tersebut adalah benar. Namun, apabila kedua nilai yang akan dibandingkan salah atau keduanya bernilai benar maka hasil akhir operasi XOR tersebut adalah salah.

Tabel 2.1 Tabel Kebenaran Operasi XOR

P	Q	$p \oplus q$
Benar	Benar	Salah
Benar	Salah	Benar
Salah	Benar	Benar
Salah	Salah	Salah

Elemen Operasi XOR dalam representasi bit terdiri dari 1 dan 0. Bit 0 adalah pernyataan bernilai salah dan bit 1 adalah pernyataan bernilai benar. Pada tabel 2.2 merupakan tabel kebenaran operasi XOR dalam representasi bit.

Tabel 2.2 Operasi XOR dalam Representasi Bit

P	Q	$p \oplus q$
1	1	0
1	0	1
0	1	1
0	0	0

2.8 *Cipher Block Chaining*

Kriptografi dibedakan menjadi dua, yaitu kriptografi klasik dan kriptografi modern. Kriptografi modern menggunakan gagasan dasar yang sama seperti kriptografi klasik (permutasi dan transposisi) tetapi penekanannya berbeda (Andriani, 2017). Algoritma kriptografi modern umumnya beroperasi dalam mode bit ketimbang mode karakter. Operasi dalam mode bit berarti semua data dan informasi (baik kunci, plainteks, maupun cipherteks) dinyatakan dalam rangkaian (*string*) bit biner, 0 dan 1. Algoritma enkripsi dan dekripsi memproses semua data dan informasi dalam bentuk rangkaian bit. Rangkaian bit yang menyatakan plainteks di enkripsi menjadi cipherteks dalam bentuk rangkaian bit, demikian sebaliknya. Perkembangan algoritma kriptografi modern berbasis bit didorong oleh penggunaan komputer digital yang mempresentasikan data dalam bentuk biner (Munir, 2006).

Cipher Block adalah salah satu teknik kriptografi modern. Pada *cipher block* rangkaian bit-bit plainteks dibagi menjadi blok-blok bit dengan panjang yang sama.

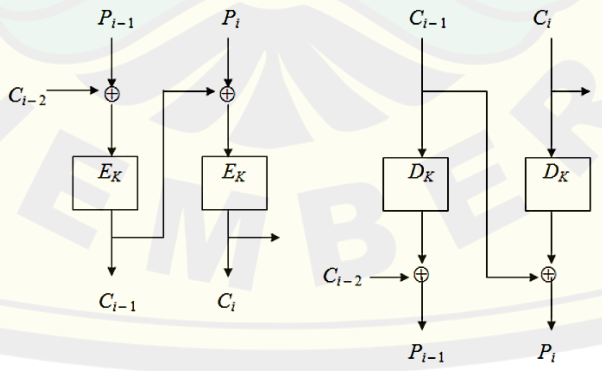
Enkripsi (E_k) dilakukan terhadap blok bit menggunakan bit-bit kunci (k) yang ukurannya sama dengan ukuran blok plainteks. Algoritma enkripsi menghasilkan blok cipherteks (C) yang berukuran sama dengan blok plainteks (P). Dekripsi (D_k) dilakukan dengan cara serupa seperti pada enkripsi. Dimana enkripsi dengan kunci dapat dinyatakan dengan Persamaan 2.2.

$$E_k(P) = C \tag{2.2}$$

Dan deskripsi dapat dinyatakan dengan Persamaan 2.3.

$$D_k(C) = P \tag{2.3}$$

Algoritma CBC merupakan penerapan mekanisme umpan-balik (*feedback*) pada sebuah blok bit dimana hasil enkripsi blok sebelumnya diumpan balikkan ke dalam proses enkripsi blok selanjutnya (Andriani, 2017). Proses enkripsi pada algoritma CBC dimulai pada blok plainteks yang di-XOR-kan terlebih dahulu dengan kunci sehingga menghasilkan blok cipherteks yang digunakan sebagai kunci blok enkripsi berikutnya. Dengan algoritma CBC ini, setiap blok cipherteks tidak hanya bergantung pada blok plainteksnya, tetapi juga pada seluruh hasil blok cipherteks sebelumnya. Dekripsi dilakukan dengan memasukkan blok cipherteks ke fungsi dekripsi, kemudian meng-XOR-kan hasilnya dengan blok cipherteks sebelumnya. Blok cipherteks sebelumnya berfungsi sebagai umpan maju (*feed forward*) pada akhir proses dekripsi (Rosmala, 2012). Pada Gambar 2.8 memperlihatkan skema proses enkripsi dan dekripsi pada *Cipher Block Chaining*.



(a) (b)

Gambar 2.8 (a) Enkripsi algoritma CBC (b) Dekripsi algoritma CBC

Dari gambar diatas, untuk mendapatkan blok cipherteks ke i (C_i). Perlu proses enkripsi dengan kunci (k) yang telah ditetapkan dengan cara meng-XOR-kan blok plainteks ke i (P_i) dengan blok cipherteks sebelumnya (C_{i-1}). Secara matematis, proses enkripsi dengan algoritma CBC dinyatakan pada persamaan 2.4.

$$C_i = E_k(P_i \oplus C_{i-1}) \tag{2.4}$$

Dan proses dekripsi dapat dinyatakan dengan persamaan 2.5.

$$P_i = D_k(C_i) \oplus C_{i-1} \tag{2.5}$$

Berikut adalah contoh dari proses enkripsi dengan plainteks berupa teks "KRIPTO" dan kunci yang digunakan adalah 2.

1. Ubah tiap blok plainteks dan kunci menjadi bentuk biner berdasarkan tabel ASCII

   Plainteks : K = 01001011
   R = 01010010
   I = 01001001
   P = 01010000
   T = 01010100
   O = 01001111

   Kunci : 2 = 00110010

2. Blok plainteks pertama di-XOR-kan dengan kunci, sehingga hasil XOR akan digunakan sebagai kunci di blok plainteks dan akan berkelanjutan sampai blok plainteks terakhir.

$$C_1 = P_1 \oplus \text{k}$$
$$= 01001011 \oplus 00110010 = 01111001$$
$$C_2 = P_2 \oplus C_1$$
$$= 01010010 \oplus 01111001 = 00101011$$
$$C_3 = P_3 \oplus C_2$$
$$= 01001001 \oplus 01111001 = 01100010$$
$$C_4 = P_4 \oplus C_3$$

$$= 01010000 \oplus 01100010 = 00110010$$

$$C_5 = P_5 \oplus C_4$$

$$= 01010100 \oplus 00110010 = 01100110$$

$$C_6 = P_6 \oplus C_5$$

$$= 01001111 \oplus 01100110 = 00101001$$

3. Hasil tiap blok cipherteks yang berupa bentuk biner dikonversikanmenjadi karakter.

$$C_1 = 01111001 = \text{y}$$
$$C_2 = 00101011 = +$$
$$C_3 = 01100010 = \text{b}$$
$$C_4 = 00110010 = 2$$
$$C_5 = 01100110 = \text{f}$$
$$C_6 = 00101001 =)$$

Sehingga hasil proses enkripsi dari plainteks “KRIPTO” menghasilkan cipherteks “y+b2f)”.

Setelah proses enkripsi dilakukan, selanjutnya adalah proses dekripsi untuk mengembalikan cipherteks ke bentuk plainteks agar dapat dibaca secara jelas oleh penerima. Berikut contoh proses dekripsi dari hasil cipherteks yang telah didapat sebelumnya, yaitu “y+b2f)” dan kunci yang telah digunakan pada proses enkripsi dan dekripsi sama, yaitu 2.

1. Ubah blok cipherteks dan kunci ke bentuk biner

Cipherteks : $y = 01111001$

$$+ = 00101011$$
$$b = 01100010$$
$$2 = 00110010$$
$$f = 01100110$$
$$) = 00101001$$

Kunci : 2 = 00110010

2. Blok cipherteks terakhir di-XOR-kan dengan blok cipherteks sebelumnya sehingga hasil dari XOR tersebut adalah blok plainteks terakhir, tahapan ini berkelanjutan hingga blok cipherteks paling akhir di-XOR-kan dengan kunci.

$$P_6 = C_6 \oplus C_5$$
$$= 00101001 \oplus 01100110 = 01001111$$
$$P_5 = C_5 \oplus C_4$$
$$= 01100110 \oplus 00110010 = 01010100$$
$$P_4 = C_4 \oplus C_3$$
$$= 00110010 \oplus 01100010 = 01010000$$
$$P_3 = C_3 \oplus C_2$$
$$= 01100010 \oplus 00101011 = 01010010$$
$$P_2 = C_2 \oplus C_1$$
$$= 00101011 \oplus 01111001 = 01010010$$
$$P_1 = C_1 \oplus k$$
$$= 01111001 \oplus 00110010 = 01001011$$

3. Hasil tiap blok plainteks yang didapat dikonversikan menjadi karakter.

$$P_1 = 01001011 = \text{K}$$
$$P_2 = 01010010 = \text{R}$$
$$P_3 = 01010010 = \text{I}$$
$$P_4 = 01010000 = \text{P}$$
$$P_5 = 01010100 = \text{T}$$
$$P_6 = 01001111 = \text{O}$$

Sehingga hasil proses dekripsi yang didapat dari cipherteks “y+b2f)” adalah plainteks “KRIPTO”.

BAB 3. METODE PENELITIAN

3.1 Data Penelitian

Data yang akan digunakan pada penelitian ini berupa teks dan graf. Pesan teks tersebut berupa alfabet, angka, maupun simbol. Pesan teks akan diubah ke dalam kode ASCII sebelum dienkripsi. Graf didapat dari graf sederhana digunakan sebagai kunci untuk mengkodekan teks, algoritma yang digunakan untuk pewarnaan titik pada graf adalah algoritma *Welch Powell* untuk mencari bilangan kromatiknya.

3.2 Langkah-Langkah Penelitian

Langkah-langkah yang dilakukan dalam penelitian ini, diuraikan sebagai berikut :

Gambar 3.1 Alur Penelitian

a. Studi Literatur

Pada tahap ini, studi literatur dilakukan dengan mempelajari teori-teori yang berkaitan dengan penelitian ini. Teori-teori yang digunakan yaitu, algoritma *Cipher Block Chaining*, *Welch Powell*, dan ASCII.

b. Mengkodekan Teks Berdasarkan Algoritma CBC dan Pewarnaan Titik pada Graf.

Tahap penelitian selanjutnya akan dilakukan terkait penelitian tentang pengodean berdasarkan algoritma CBC dan pewarnaan titik pada graf adalah sebagai berikut :

1) Proses Enkripsi

Proses enkripsi dilakukan dengan membagi tiap karakter plainteks menjadi tiap blok biner dimana tiap blok bergantung satu sama lain.

a) Masukkan plainteks yang akan dikodekan
b) Mengkonversi plainteks menjadi bit biner berdasarkan kode ASCII
c) Masukkan graf sebagai kunci
d) Menghitung bilangan kromatik dengan langkah-langkah sebagai berikut :

1. Menentukan graf yang akan diteliti;

   Menentukan graf yang akan digunakan sebagai kunci dengan n titik. Graf yang digunakan adalah graf sederhana yang akan dicari bilangan kromatiknya.
2. Mengurutkan titik-titik dari derajat terbesar ke terkecil;

   Mengurutkan titik-titik dari derajat terbesar sampai derajat terkecil dengan cara membentuk matriks

adjacent yang menyatakan keterhubungan antar titik. Untuk menentukan derajat tiap titik dengan menghitung jumlah dari elemen tiap baris.

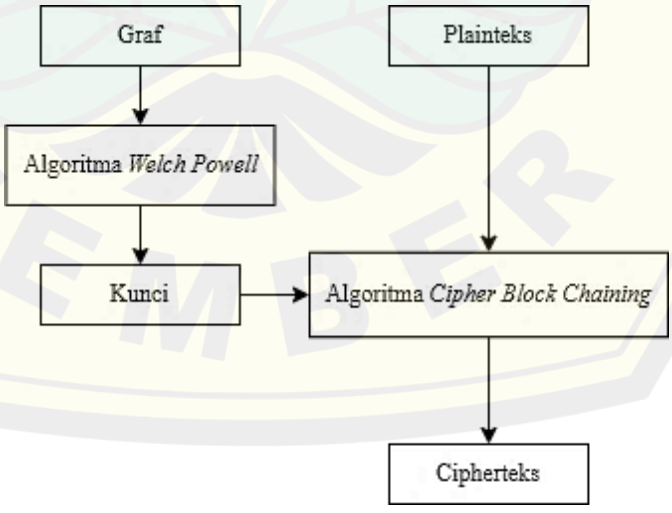
3. Mewarnai titik pada graf;

   Pewarnaan titik dengan memilih titik berderajat tertinggi. Titik derajat tertinggi dan diberi warna terlebih dahulu. Periksa semua titik yang tidak bertetangga dan diberi yang warna sama. Kemudian mengulangi dimulai dari titik berderajat tertinggi berikutnya yang belum diberi warna sampai semua titik terwarnai.

4. Menentukan bilangan kromatik;

   Menentukan bilangan kromatik dengan mengetahui minimal warna yang didapat.

e) Blok pertama plainteks di-XOR-kan dengan kunci
f) Didapat cipherteks blok pertama
g) Blok plainteks selanjutnya di-XOR-kan dengan hasil cipherteks sebelumnya
h) Ulangi proses diatas sampai blok plainteks selesai



Gambar 3. 2 Alur enkripsi penelitian

2) Proses Dekripsi

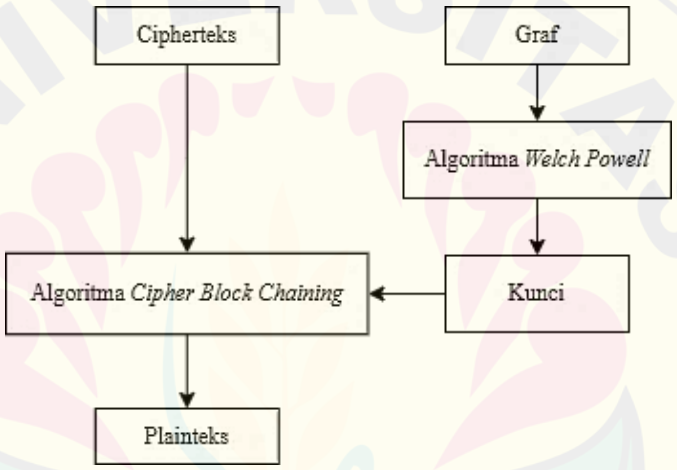
Data input dekripsi adalah cipherteks, output dari proses enkripsi. Agar mendapatkan plainteks yang sesuai, harus menggunakan kunci yang sama dengan proses enkripsi.

a) Masukkan cipherteks yang akan dikodekan
b) Mengkonversi cipherteks menjadi bit biner berdasarkan kode ASCII
c) Masukkan graf sebagai kunci
d) Menghitung bilangan kromatk dengan langkah-langkah sebagai berikut :

1. Menentukan graf yang akan diteliti;
   Menentukan graf yang akan digunakan sebagai kunci dengan n titik. Graf yang digunakan adalah graf sederhana yang akan dicari bilangan kromatiknya.
2. Mengurutkan titik-titik dalam derajat menurun;
   Mengurutkan titik-titik dari derajat terbesar sampai derajat terkecil dengan cara membentuk matriks *adjacent* yang menyatakan keterhubungan antar titik. Untuk menentukan derajat tiap titik dengan menghitung jumlah dari elemen tiap baris.
3. Mewarnai titik pada graf;
   Pewarnaan titik dengan memilih titik berderajat tertinggi. Titik derajat tertinggi dan diberi warna terlebih dahulu. Periksa semua titik yang tidak bertetangga dan diberi yang warna sama. Kemudian mengulangi dimulai dari titik berderajat tertinggi berikutnya yang belum diberi warna sampai semua titik terwarnai.
4. Menentukan bilangan kromatik;

Menentukan bilangan kromatik dengan mengetahui minimal warna yang didapat.

e) Blok cipherteks terakhir dan blok cipherteks sebelumnya saling di-XOR-kan
f) Didapat blok plainteks
g) Ulangi proses (e) hingga blok cipherteks terakhir
h) Pada blok cipherteks terakhir, di-XOR-kan dengan kunci yang berupa graf
i) Didapat blok plainteks



Gambar 3. 3 Alur dekripsi penelitian

c. Pembuatan Program

Pembuatan program dilakukan berdasarkan enkripsi dan dekripsi yang disusun menggunakan MATLAB 2009a.

d. Analisa Hasil

Menguji program yang telah dibuat berjalan sesuai metode yang digunakan.

e. Kesimpulan

Mengambil kesimpulan dari penelitian yang dilakukan yaitu menganalisa hasil dari pengodean teks berdasarkan algoritma CBC dan pewarnaan titik pada graf.

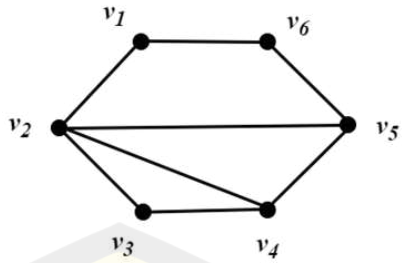
BAB 4. HASIL DAN PEMBAHASAN

Pada bab ini akan dibahas mengenai proses pengodean teks berdasarkan algoritma CBC dan pewarnaan titik pada graf. Implementasi pengodean teks pada program dengan menggunakan *software* Matlab 2009a dan pembahasan mengenai pengodean secara manual berdasarkan algoritma CBC dan perwarnaan titik pada graf.

4.1 Hasil

Hasil dari penelitian ini adalah program untuk mengodekan suatu teks berdasarkan algoritma CBC dan pewarnaan titik pada graf. Pembuatan program pada penelitian ini didesain dan dibuat dengan *software* Matlab 2009a. Program ini digunakan untuk pengodean suatu teks dengan memasukkan teks dan kunci sehingga akan memunculkan hasil berupa teks yang telah dikodekan. Tampilan awal program dapat dilihat pada Gambar 4.1. Teks yang akan dienkripsikan adalah "kode" dengan kunci sebuah graf pada Gambar 4.2. Teks dimasukkan pada kolom "Masukkan Teks" dan kunci dimasukkan pada kolom "Kunci" dengan mendefinisikan jumlah titik dari suatu graf dan keterhubungan antar tiap titik dari graf.

Gambar 4.1 Tampilan awal program



Gambar 4.2 Kunci pengodean

$$\begin{array}{c} \\ v_1 \\ v_2 \\ v_3 \\ v_4 \\ v_5 \\ v_6 \end{array} \begin{array}{c} \begin{array}{cccccc} v_1 & v_2 & v_3 & v_4 & v_5 & v_6 \end{array} \\ \begin{bmatrix} 0 & 1 & 0 & 0 & 0 & 1 \\ 1 & 0 & 1 & 1 & 1 & 0 \\ 0 & 1 & 0 & 1 & 0 & 0 \\ 0 & 1 & 1 & 0 & 1 & 0 \\ 0 & 1 & 0 & 1 & 0 & 1 \\ 1 & 0 & 0 & 0 & 1 & 0 \end{bmatrix} \end{array}$$

Gambar 4.3 Matriks *adjacency* dari graf

Untuk menjalankan proses pengodean perlu menekan tombol "Enkripsi" atau "Dekripsi". Proses enkripsi dapat berjalan dengan menekan tombol "Enkripsi" dan akan mendapatkan hasil enkripsi berupa cipherteks pada kolom "Hasil Teks". Sedangkan untuk menjalankan proses dekripsi perlu menekan tombol "Dekripsi" untuk menjalankan program dan mendapatkan hasil yang merupakan plainteks pada kolom "Hasil Teks".

Langkah - langkah penggunaan program adalah sebagai berikut :

1. Masukkan teks yang akan dikodekan pada kolom input "Masukkan Teks"
2. Masukkan jumlah titik graf yang digunakan sebagai kunci pada kolom input jumlah titik
3. Masukkan hubungan antar titik graf dengan bilangan 0 atau 1 pada kolom input hubungan titik 1 dan titik 2, dan seterusnya.
4. Pilih tombol "Enkripsi" untuk proses enkripsi sehingga akan muncul hasil cipherteks pada kolom "Hasil Teks"

5. Untuk proses dekripsi, masukkan cipherteks pada kolom “Masukkan Teks” lalu pilih tombol “Dekripsi” sehingga akan muncul hasil berupa plainteks pada kolom “Hasil Teks”.

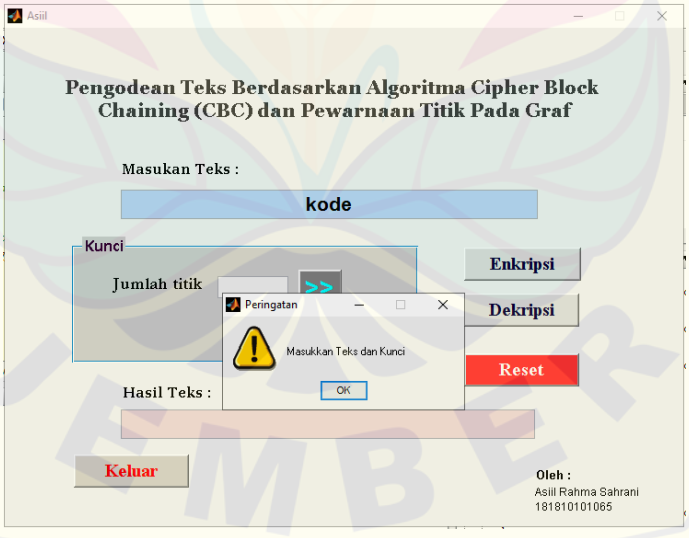
Kolom kunci pada tampilan awal program perlu mendekripsikan bentuk graf pada gambar 4.2 dengan jumlah titik 6 dan matriks *adjacency* pada gambar 4.3 menunjukkan keterhubungan antar titik. Setelah memasukkan hubungan antar titik, Untuk mengetahui keberhasilan dari program yang telah dibuat, berikut adalah tampilan dari proses enkripsi dengan menggunakan plainteks dan kunci dari graf yang telah ditentukan. Sesuai dengan langkah-langkah yang telah dijelaskan sebelumnya, langkah pertama adalah memasukkan teks yang akan dikodekan dan kunci yang akan digunakan. Berikut tampilan input program dapat dilihat pada Gambar 4.4.

Gambar 4.4 Tampilan *input* program

Setelah kunci dimasukkan, kolom input hubungan antar titik akan otomatis hilang untuk memperkuat kerahasiaan kunci program. Untuk menjalankan proses enkripsi, pilih tombol “Enkripsi” dan hasil dari proses enkripsi akan muncul pada kolom “Hasil Teks” seperti pada Gambar 4.5 berikut.

Gambar 4.5 Tampilan hasil enkripsi program

Program dapat berjalan baik jika kolom input teks dan kunci terisi. Jika kolom input kosong akan muncul peringatan "Masukkan Teks dan Kunci" sehingga program dapat kembali berjalan sesuai metode yang telah dibuat. Berikut tampilan jika kolom input kunci kosong, tampilan peringatan akan muncul seperti pada Gambar 4.6 di bawah.



Gambar 4. 6 Tampilan peringatan jika kolom *input* kosong

Untuk proses dekripsi dapat dilakukan dengan memasukkan teks dan kunci dan memilih tombol “Dekripsi”. Hasil dari proses dekripsi adalah plainteks yang akan muncul pada kolom “Hasil Teks”. Pada simulasi program ini teks yang dimasukkan adalah cipherteks pada proses enkripsi sebelumnya, dan kunci yang digunakan sama seperti pada proses enkripsi. Sehingga hasil dari proses dekripsi dapat dilihat pada Gambar 4.7.

Gambar 4. 7 Tampilan hasil dekripsi program

4.2 Pembahasan

4.2.1 Pengodean Teks dengan Algoritma CBC dan Pewarnaan Titik pada Graf

Data yang digunakan adalah teks berupa alfabet, angka, maupun simbol dan kunci yang digunakan berupa graf. Plainteks dikodekan menjadi cipherteks yang memiliki panjang yang sama. Cipherteks juga dapat didekripsi menjadi plainteks dan kembali menjadi pesan yang semula seperti dimasukkan. Plainteks dikodekan berdasarkan algoritma CBC dan graf diproses berdasarkan algoritma *Welch Powell* untuk menentukan bilangan kromatiknya. Langkah awal untuk mengkodekan teks menggunakan algoritma CBC adalah menentukan karakter dan kunci yang akan dikodekan, berikut contoh karakter plainteks yang akan dikodekan yaitu “kode” dan

kunci berupa graf pada Gambar 4.2. Berikut pembahasan dari proses enkripsi dan dekripsi pada pengodean teks dengan algoritma CBC dan pewarnaan titik pada graf.

1. Proses Enkripsi

Pada proses ini, plainteks "kode" akan diubah menjadi cipherteks dengan kunci dari graf Gambar 4.2. Berikut langkah-langkah untuk proses enkripsi :

1. Konversikan tiap blok plainteks menjadi bentuk biner
   Plainteks yang digunakan adalah "kode" dan dibagi tiap bloknya menjadi $P_1 = \text{k}$, $P_2 = \text{o}$, $P_3 = \text{d}$, dan $P_4 = \text{e}$. Tiap blok plainteks berupa karakter dikonversikan menjadi bentuk biner 8 digit.
   - Bentuk biner dari k $= 01101011$
   - Bentuk biner dari o $= 01101111$
   - Bentuk biner dari d $= 01100100$
   - Bentuk biner dari e $= 01100101$

2. Masukkan graf sebagai kunci dan menghitung bilangan kromatik.

   Langkah-langkah untuk menentukan bilangan kromatik menggunakan algoritma *Welch Powell* adalah sebagai berikut:

   1. Membuat matrik *adjacency* berdasarkan graf yang akan digunakan, matriks *adjacency* dari graf pada Gambar 4.2 dapat dilihat pada Gambar 4.3.
   2. Mengurutkan titik-titik pada graf berdasarkan urutan dari derajat terbesar ke terkecil, berikut tabel urutan derajat dari tiap titik dari graf Gambar 4.2 dapat dilihat pada Tabel 4.1 di bawah.

      Tabel 4.1 Urutan Derajat Terbesar ke Terkecil dari Graf

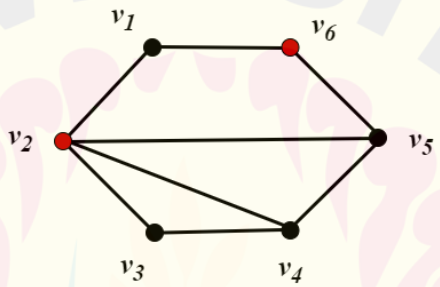
Titik	Derajat
v_2	4
v_4	3
v_5	3
v_1	2
v_3	2
v_6	2

   3. Mewarnai titik pada derajat yang paling tinggi yaitu titik v_2 dan titik yang tidak berelasi dengan titik v_2 yaitu v_6 dengan warna a. Berikut

tabel dari titik yang terwarnai dengan warna *a,* dapat dilihat pada Tabel 4.2 dan warna titik graf dari Gambar 4.2 yang terwarnai dapat dilihat pada Gambar 4.8.

Tabel 4. 2 Pewarnaan Titik dengan Warna *a* pada Titik Graf Sampel

Titik	Derajat	Warna
v_2	4	a
v_4	3	
v_5	3	
v_1	2	
v_3	2	
v_6	2	a

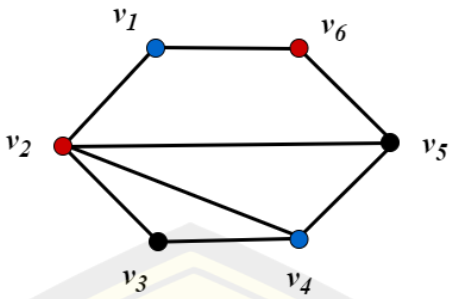


Gambar 4.8 Pewarnaan titik v_2 dan v_6 dengan warna a

4. Mewarnai titik yang belum diwarnai yaitu titik v_4 dan titik yang tidak bertetangga dengan titik v_4 yaitu v_1 dengan warna *b.* Berikut tabel dari titik yang terwarnai dengan warna *b,* dapat dilihat pada Tabel 4.3 dan warna titik graf dari Gambar 4.2 yang terwarnai dapat dilihat pada Gambar 4.9.

Tabel 4. 3 Pewarnaan Titik dengan Warna b pada Titik Graf Sampel

Titik	Derajat	Warna
v_2	4	a
v_4	3	b
v_5	3	
v_1	2	b
v_3	2	
v_6	2	a

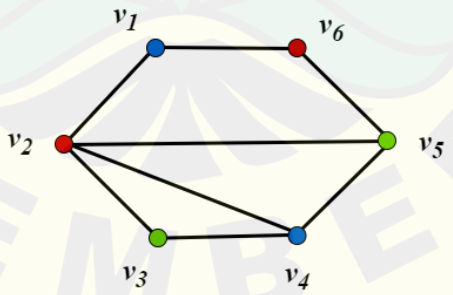


Gambar 4.9 Pewarnaan titik v_1 dan v_4 dengan warna b

5. Mewarnai titik yang belum diwarnai yaitu titik v_5 dan titik yang tidak berelasi dengan titik v_5 yaitu v_3 dengan warna *c*. Berikut tabel dari titik yang terwarnai dengan warna *c,* dapat dilihat pada Tabel 4.4 dan warna titik graf dari Gambar 4.2 yang terwarnai dapat dilihat pada Gambar 4.10.

Tabel 4. 4 Pewarnaan Titik dengan Warna *c* pada Titik Graf Sampel

Titik	Derajat	Warna
v_2	4	a
v_4	3	b
v_5	3	c
v_1	2	b
v_3	2	c
v_6	2	a



Gambar 4. 10 Pewarnaan titik v_3 dan v_5 dengan warna c

6. Setelah semua titik terwarnai, didapat 3 jenis warna yang digunakan untuk mewarnai graf, yaitu warna *a,b,* dan *c*. Sehingga bilangan kromatik $\chi(G)$ yang didapat yaitu 3.

3. Nilai desimal dari plainteks P_1 merupakan 107 kemudian dikurangi 32 sehingga menjadi 75. Bentuk biner dari 75 yaitu 01001011 di-XOR-kan dengan bentuk biner dari $\chi(G) = 3$ yaitu 00000011.

$$\begin{array}{rll} \text{Biner } 75 & : & 01001011 \\ \text{Biner } \chi(G) & : & 00000011 \\ \hline C_1 & : & 01001000 \end{array} \oplus$$

Hasil XOR diatas dikonversikan menjadi desimal yaitu 72, kemudian dimodulo kan dengan 95.

$$72 \; mod \; 95 = 72$$

Hasil setelah dimodulo ditambah 32 sehingga menjadi 104, dan dikonversikan menjadi bentuk karakter sebagai blok cipherteks pertama (C_1). Sehingga didapat C_1 yaitu h.

4. Nilai desimal dari blok plainteks kedua (P_2) adalah 111 dikurangi 32 menjadi 79. Bentuk biner dari 79 yaitu 01001111 di-XOR-kan dengan C_1.

$$\begin{array}{rll} \text{Biner } 79 & : & 01001111 \\ \text{Biner } C_1 & : & 01001000 \\ \hline C_2 & : & 00000111 \end{array} \oplus$$

Hasil XOR diatas dikonversikan menjadi desimal yaitu 7, kemudian dimodulo kan dengan 95.

$$7 \; mod \; 95 = 7$$

Hasil setelah dimodulo ditambah 32 sehingga menjadi 39, dan dikonversikan menjadi bentuk karakter sebagai blok cipherteks kedua (C_2). Sehingga didapat C_2 yaitu ‘.

5. Nilai desimal dari blok plainteks ketiga (P_3) adalah 100 dikurangi 32 menjadi 68. Bentuk biner dari 68 yaitu 01000100 di-XOR-kan dengan bentuk biner dari C_2 yaitu 00000111.

$$\begin{array}{rll} \text{Biner } 68 & : & 01000100 \\ \text{Biner } C_2 & : & 00000111 \\ \hline C_3 & : & 01000011 \end{array} \oplus$$

Hasil XOR diatas dikonversikan menjadi desimal yaitu 67, kemudian dimodulo kan dengan 95.

$$67\ mod\ 95 = 67$$

Hasil setelah dimodulo ditambah 32 sehingga menjadi 99, dan dikonversikan menjadi bentuk karakter sebagai blok cipherteks kedua (C_3). Sehingga didapat C_3 yaitu c.

6. Nilai desimal dari blok plainteks terakhir (P_4) adalah 101 dikurangi 32 menjadi 69. Bentuk biner dari 69 yaitu 01000101 di-XOR-kan dengan bentuk biner dari C_3 yaitu 01000011.

$$\begin{array}{rl} \text{Biner } P_4 : & 01000101 \\ \text{Biner } C_3 & 01000011 \oplus \\ \hline C_4 : & 00000110 \end{array}$$

Hasil XOR diatas dikonversikan menjadi desimal yaitu 6, kemudian dimodulo kan dengan 95.

$$6\ mod\ 95 = 6$$

Hasil setelah dimodulo ditambah 32 sehingga menjadi 38, dan dikonversikan menjadi bentuk karakter sebagai blok cipherteks kedua (C_4). Sehingga didapat C_4 yaitu &.

7. Setelah semua blok plainteks di-XOR-kan, didapat blok cipherteks
   - Bentuk karakter dari $C_1 = h$
   - Bentuk karakter dari $C_2 = '$
   - Bentuk karakter dari $C_3 = c$
   - Bentuk karakter dari $C_4 = \&$

2. Proses Dekripsi

Pada proses ini, cipherteks “h’c&” akan diubah menjadi plainteks berdasarkan algoritma CBC dengan kunci yang telah ditentukan sebelumnya. Berikut langkah-langkah untuk proses dekripsi :

1. Masukkan cipherteks dan bagi menjadi tiap blok cipherteks.
   Cipherteks yang digunakan adalah "h'c&" dan dibagi tiap bloknya menjadi $C_1 =$ h, $C_2 =$ ', $C_3 =$ c, dan $C_4 =$ d.

2. Masukkan graf sebagai kunci dan menghitung bilangan kromatiknya.
   Kunci yang akan digunakan adalah graf pada Gambar 4.2. Langkah-langkah untuk mencari bilangan kromatik dapat dilihat pada langkah (2) proses enkripsi yang telah dijelaskan sebelumnya. Hasil dari pewarnaan titik adalah 3 jenis warna yang digunakan untuk mewarnai graf, yaitu warna a,b, dan c. Sehingga bilangan kromatik χ(G) yang didapat yaitu 3. Hasil pewarnaan titik dapat dilihat pada Gambar 4.10.

3. Blok cipherteks terakhir (C_4) dan C_3 dikonversikan menjadi bentuk desimal sehingga nilai desimal dari C_4 adalah 38 dan C_3 adalah 99. Setelah didapat nilai desimalnya kemudian masing-masing dikurangi 32.

   $$C_4 - 32 = 38 - 32 \qquad C_3 - 32 = 99 - 32$$
   $$= 6 \qquad = 67$$

   Setelah didapat hasil penguran dengan 32, kemudian C_4 dan C_3 dikonversikan menjadi bentuk biner sehingga bentuk biner dari C_4 adalah 00000110 dan C_3 adalah 01000011 . Bentuk biner dari C_4 dan C_3 di-XOR-kan. Hasil dari operasi XOR tersebut adalah P_4.

   $$\begin{array}{lll} \text{Biner } C_4 : & 00000110 & \\ \text{Biner } C_3 : & 01000011 & \oplus \\ \hline & P_4: \ 01000101 & \end{array}$$

   Hasil XOR diatas dikonversikan menjadi desimal yaitu 69, kemudian dimodulo kan dengan 95.

   $$69 \ mod \ 95 = 69$$

   Hasil setelah dimodulo ditambah 32 sehingga menjadi 101, dan dikonversikan menjadi bentuk karakter sebagai P_4. Sehingga didapat P_4 yaitu e.

4. Blok cipherteks C_3 dan C_2 dikonversikan menjadi bentuk desimal sehingga nilai desimal dari C_2 adalah 96. Setelah didapat nilai desimalnya kemudian dikurangi 32 sehingga menghasilkan 64. Nilai desimal dan hasil pengurangan pada blok cipherteks C_3 telah dijelaskan pada tahapan sebelumnya. Setelah didapat hasil penguran dengan 32, kemudian C_3 dan C_2 dikonversikan menjadi bentuk biner sehingga bentuk biner dari C_3 adalah 01000011dan C_2 adalah 00000111. Bentuk biner dari C_3 dan C_2 di-XOR-kan. Hasil dari operasi XOR tersebut adalah P_3.

$$\begin{array}{rl} \text{Biner } C_3 : & 01000011 \\ \text{Biner } C_2 & 00000111 \oplus \\ \hline P_3: & 01000100 \end{array}$$

Hasil XOR diatas dikonversikan menjadi desimal yaitu 68, kemudian dimodulo kan dengan 95.

$$68 \; mod \; 95 = 68$$

Hasil setelah dimodulo ditambah 32 sehingga menjadi 100, dan dikonversikan menjadi bentuk karakter sebagai P_3. Sehingga didapat P_3 yaitu d.

5. Blok cipherteks C_2 dan C_1 dikonversikan menjadi bentuk desimal sehingga nilai desimal dari C_1 adalah 104. Setelah didapat nilai desimalnya kemudian dikurangi 32 sehingga menghasilkan 72. Nilai desimal dan hasil pengurangan pada blok cipherteks C_2 telah dijelaskan pada tahapan sebelumnya. Setelah didapat hasil penguran dengan 32, kemudian C_2 dan C_1 dikonversikan menjadi bentuk biner sehingga bentuk biner dari C_2 adalah 01000011dan C_1 adalah 00000111. Bentuk biner dari C_2 dan C_1 di-XOR-kan. Hasil dari operasi XOR tersebut adalah P_2.

$$\begin{array}{rl} \text{Biner } C_2 : & 00000111 \\ \text{Biner } C_1 & 01001000 \oplus \\ \hline P_2: & 01001111 \end{array}$$

Hasil XOR diatas dikonversikan menjadi desimal yaitu 79, kemudian dimodulo kan dengan 95.

$$79 \; mod \; 95 = 79$$

Hasil setelah dimodulo ditambah 32 sehingga menjadi 101, dan dikonversikan menjadi bentuk karakter sebagai P_2. Sehingga didapat P_2 yaitu o.

6. Blok cipherteks C_1 dikonversikan menjadi bentuk desimal kemudian dikurangi 32, sehingga hasil dari pengurangan tersebut adalah 72. Bentuk biner dari 72 adalah 01001000 kemudian di-XOR-kan dengan bentuk biner dari $\chi(G) = 3$ yaitu 00000011. Hasil dari operasi XOR tersebut adalah P_1.

$$\begin{array}{rl} \text{Biner } C_1 & : \quad 01001000 \\ \text{Biner } \chi(G) & \quad\;\; 00000011 \\ \hline P_1: & \quad 01001011 \end{array} \oplus$$

Hasil XOR diatas dikonversikan menjadi desimal yaitu 75, kemudian dimodulo kan dengan 95.

$$75 \; mod \; 95 = 75$$

Hasil setelah dimodulo ditambah 32 sehingga menjadi 107, dan dikonversikan menjadi bentuk karakter sebagai P_1. Sehingga didapat P_1 yaitu k.

7. Setelah didapat plainteks dalam bentuk biner, kemudian dikonversikan menjadi karakter.
   - Bentuk karakter dari $P_1 = k$
   - Bentuk karakter dari $P_2 = o$
   - Bentuk karakter dari $P_3 = d$
   - Bentuk karakter dari $P_4 = e$

4.2.2 Analisis Hasil

Secara umum, pengodean CBC adalah algoritma pengodean pesan yang memiliki tujuan mengodekan suatu teks dengan kunci tertentu yang bersifat rahasia. Penulis membuat penelitian ini dengan kunci yang berupa graf. Kunci yang telah ditentukan perlu dicari bilangan kromatiknya dengan algoritma *Welch Powell.* Sehingga pihak yang tidak memiliki wewenang tidak dapat mengetahui kunci yang sebenarnya.

Algoritma CBC pada rangkaian bit-bit data teks yang dikodekan perlu dibagi menjadi blok-blok bit dengan panjang yang sama. Proses enkripsi dilakukan dengan blok bit plainteks yang di-XOR-kan dengan bit kunci yang telah ditentukan sehingga menghasilkan blok bit cipherteks. Proses dekripsi dilakukan dengan cara serupa seperti pada proses enkripsi. Pada penelitian kode ASCII yang digunakan berjumlah 95 dari karakter 32 sampai karakter ke 127. Sehingga pada sebelum proses XOR perlu di kurangi 32 begitupun setelah proses XOR dimodulo kan dengan 95 dan hasilnya ditambahkan 32 kembali. Pada proses enkripsi dilakukan dengan blok bit plainteks dikurangi 32 kemudian di-XOR-kan dengan kunci berupa graf yang perlu dicari bilangan kromatiknya melalui algoritma *Welch Powell* terlebih dahulu. Hasil enkripsi berupa cipherteks dimodulo kan dengan 95, kemudian ditambah 32 sehingga menghasilkan blok bit cipherteks. Hasil blok bit cipherteks digunakan sebagai kunci untuk mendapatkan blok bit cipherteks selanjutnya, hal ini akan terus berulang hingga blok bit cipherteks dan plainteks habis. Blok-blok bit cipherteks yang dihasilkan kemudian dikonversikan dari bentuk biner menjadi bentuk karakter. Proses dekripsi sama seperti proses enkripsi hanya saja membutuhkan blok bit cipherteks dan blok bit kunci yang akan menghasilkan blok bit plainteks. Blok bit cipherteks paling akhir dan sebelumnya dikurangi 32 dan kemudian di-XOR-kan sehingga menghasilkan blok bit plainteks paling akhir yang perlu dimodulo kan dengan 95 dan ditambah 32 kembali sehingga menghasilkan blok plainteks terakhir. Perulangan proses ini dilakukan hingga blok cipherteks tidak memiliki pasangan untuk di-XOR-kan sehingga perlu di-XOR-kan dengan kunci sehingga menghasilkan blok plainteks pertama.

Program dapat dapat dikatakan sesuai jika menghasilkan hasil yang sama dengan pengodean secara manual. Pada perhitungan manual, proses enkripsi dengan plainteks yang digunakan adalah “kode” dengan kunci sebuah graf pada Gambar 4.2 menghasilkan cipherteks berupa “h’c&”. Sesuai dengan simulasi program yang telah dibahas sebelumnya dengan plainteks dan kunci yang sama seperti perhitungan manual menghasilkan hasil teks yang sama seperti perhitungan manual. Begitupun pada proses dekripsi, program menjalankan proses pengodean

dengan sesuai dengan perhitungan manual. Sehingga pembuatan program ini sudah sesuai dengan metode dan algoritma yang digunakan.

BAB 5. PENUTUP

5.1 Kesimpulan

Berdasarkan hasil dan pembahasan mengenai pengodean teks berdasarkan algoritma CBC dan pewarnaan titik pada graf. Teks yang akan dikodekan perlu dikonversikan menjadi bentuk biner yang selanjutnya di-XOR-kan dengan bentuk biner dari kunci yang ditentukan. Kunci yang digunakan adalah suatu graf sederhana. Graf yang digunakan sebagai kunci dicari bilangan kromatiknya melalui algoritma *Welch Powell*. Proses Enkripsi menggunakan pengodean teks berdasarkan CBC dan pewarnaan titik pada graf menghasilkan pesan teks yang telah dikodekan. Program pengodean teks dengan memasukkan teks yang akan dikodekan (plainteks) dan kunci yang telah ditentukan kedalam program dan menghasilkan suatu pesan yang telah dikodekan (cipherteks). Sehingga didapat hasil dari pengodean teks adalah pesan yang telah dikodekan berdasarkan algoritma CBC dan pewarnaan titik pada graf.

5.2 Saran

Penelitian mengenai kriptografi masih terbuka bagi peneliti lain karena pada kriptografi terdapat bermacam-macam algoritma yang dapat digunakan selain algoritma CBC, seperti algoritma Hill Cipher atau algoritma kriptografi lainnya. Untuk lebih menyulitkan pengodean maka peneliti lain dapat menggunakan algoritma selain *Welch Powell* untuk membentuk kunci yang lebih sulit pada algoritma CBC. Algoritma Kruskal atau Prim dapat menjadi solusi untuk membentuk kunci dari graf berbobot dengan menentukan bobot minimum dari graf.

DAFTAR PUSTAKA

Adiwijaya. 2016. *Matematika Diskrit dan Aplikasinya*. Bandung: Alfabeta.

Ahmad, F., P. N. Basuki, dan R. Christ. 2016. Perancangan Kriptografi Block Cipher berbasis CBC (*Cipher Block Chaining*) Termodifikasi dalam Pengamanan Data Lokasi pada *Database Server* Aplikasi *MeetApss*. *Artikel Ilmiah*. Salatiga : Universitas Kristen Satya Wacana.

Ariyus, D. 2008. *Pengantar Ilmu Kriptografi Teori, Analisis, dan Implementasi*. Yogyakarta : ANDI.

Daniel, F. dan P. N. L Taneo, 2019, *Teori Graf*, Yogyakarta: Deepublish

Joedo, J. C.. 2020. *Analisis Bilangan Kromatik Packing Pada Graf Hasil Operasi Edge Corona dan Relevansinya dengan Berpikir Kreatif*. *Skripsi*. Jember : Universitas Jember.

Kustandi, Cecep dan Darmawan, Daddy, 2020, Pengembangan Media Pembelajaran: Konsep dan Aplikasi Pengembangan Media Pembelajaran Bagi Pendidik di Sekolah dan Masyarakat, Jakarta: Kencana.

Lipson, M. L. 1992. *2000 Solved Problem in Discrete Mathematics.* New York: Mc Graw Hill.

Mahendra, A. Z. 2016. Implementasi Kriptografi Affine Cipher Pada Citra Digital Hasil Steganografi Metode Paruty Coding Dengan Pseudo Random Number Generator (PRNG). *Skripsi*. Jember : Universitas Jember.

Mukhtar, H. 2018. *Kriptografi Untuk Keamanan Data*. Sleman : DEEPUBLISH.

Munir, R. 2006. *Kriptografi*, Bandung : Informatika,

Munir, R. 2010. *Matematika Diskrit*. Edisi ketiga. Bandung: Informatika Bandung.

Rahmawati, R. 2017. Penggabungan Vigenere cipher dengan hill cipher pada pengkodean plaintext dengan kunci bertahap. *Skripsi*. Jember: Universitas Jember.

Rosmala, D. 2012. *Implementasi Mode Cipher Block Chaining (CBC) pada pengamanan Data*, Vol.3.

Santoso, K A, Dafik, I. H. Agustin, R. M. Prihandini, dan R. Alfarisi. 2019. *Vertex Colouring Using The Adjacency Matrix. Journal of Physics : Conf. Series. IOP Publishing:* 1211(1). 16.

Santoso, K A, I. H. Agustin, dan R. M. Prihandini. 2019. *The Modification of Caesar Cryptosystem Based on Vertices Colouring*. *Journal of Physics: Conference Series. IOP Publishing:* 012006.

Siang, J.J. 2002. *Matematika Diskrit dan Aplikasinya pada Ilmu Komputer.* Andi, Yogyakarta.

Siregar, M. K. 2018. *Matematika Diskrit*. Lampung: Perahu Litera.

LAMPIRAN

Lampiran A. Tabel ASCII

Desimal	Octa	Hexa	Biner	Karakter
0	000	00	00000000	NULL
1	001	01	00000001	SOH
2	002	02	00000010	STX
3	003	03	00000011	ETX
4	004	04	00000100	EOT
5	005	05	00000101	ENQ
6	006	06	00000110	ACK
7	007	07	00000111	BEL
8	010	08	00001000	BS
9	011	09	00001001	HT
10	012	0A	00001010	LF
11	013	0B	00001011	VT
12	014	0C	00001100	FF
13	015	0D	00001101	CR
14	016	0E	00001110	SO
15	017	0F	00001111	SI
16	020	10	00010000	DLE
17	021	11	00010001	DC1
18	022	12	00010010	DC2
19	023	13	00010011	DC3
20	024	14	00010100	DC4
21	025	15	00010101	NAK
22	026	16	00010110	SYN
23	027	17	00010111	ETB
24	030	18	00011000	CAN
25	031	19	00011001	EM
26	032	1A	00011010	SUB
27	033	1B	00011011	ESC
28	034	1C	00011100	FS
29	035	1D	00011101	GS
30	036	1E	00011110	RS
31	037	1F	00011111	US
32	040	20	00100000	(Spasi)
33	041	21	00100001	!
34	042	22	00100010	"
35	043	23	00100011	#

Desimal	Octa	Hexa	Biner	Karakter
37	045	25	00100101	%
38	046	26	00100110	&
39	047	27	00100111	'
40	050	28	00101000	(
41	051	29	00101001	)
42	052	2A	00101010	*
43	053	2B	00101011	+
44	054	2C	00101100	,
45	055	2D	00101101	-
46	056	2E	00101110	.
47	057	2F	00101111	/
48	060	30	00110000	0
49	061	31	00110001	1
50	062	32	00110010	2
51	063	33	00110011	3
52	064	34	00110100	4
53	065	35	00110101	5
54	066	36	00110110	6
55	067	37	00110111	7
56	070	38	00111000	8
57	071	39	00111001	9
58	072	3A	00111010	:
59	073	3B	00111011	;
60	074	3C	00111100	<
61	075	3D	00111101	=
62	076	3E	00111110	>
63	077	3F	00111111	?
64	100	40	01000000	@
65	101	41	01000001	A
66	102	42	01000010	B
67	103	43	01000011	C
68	104	44	01000100	D
69	105	45	01000101	E
70	106	46	01000110	F
71	107	47	01000111	G
72	110	48	01001000	H
73	111	49	01001001	I
74	112	4A	01001010	J
75	113	4B	01001011	K
76	114	4C	01001100	L
77	115	4D	01001101	M

Desimal	Octa	Hexa	Biner	Karakter
79	117	4F	01001111	O
80	120	50	01010000	P
81	121	51	01010001	Q
82	122	52	01010010	R
83	123	53	01010011	S
84	124	54	01010100	T
85	125	55	01010101	U
86	126	56	01010110	V
87	127	57	01010111	W
88	130	58	01011000	X
89	131	59	01011001	Y
90	132	5A	01011010	Z
91	133	5B	01011011	[
92	134	5C	01011100	\
93	135	5D	01011101	]
94	136	5E	01011110	^
95	137	5F	01011111	_
96	140	60	01100000	`
97	141	61	01100001	a
98	142	62	01100010	b
99	143	63	01100011	c
100	144	64	01100100	d
101	145	65	01100101	e
102	146	66	01100110	f
103	147	67	01100111	g
104	150	68	01101000	h
105	151	69	01101001	i
106	152	6A	01101010	j
107	153	6B	01101011	k
108	154	6C	01101100	l
109	155	6D	01101101	m
110	156	6E	01101110	n
111	157	6F	01101111	o
112	160	70	01110000	p
113	161	71	01110001	q
114	162	72	01110010	r
115	163	73	01110011	s
116	164	74	01110100	t
117	165	75	01110101	u
118	166	76	01110110	v
119	167	77	01110111	w

Desimal	Octa	Hexa	Biner	Karakter
121	171	79	01111001	y
122	172	7A	01111010	z
123	173	7B	01111011	{
124	174	7C	01111100	|
125	175	7D	01111101	}
126	176	7E	01111110	~
127	177	7F	01111111	•
128	200	80	10000000	€
129	201	81	10000001	•
130	202	82	10000010	,
131	203	83	10000011	*ƒ*
132	204	84	10000100	„
133	205	85	10000101	…
134	206	86	10000110	†
135	207	87	10000111	‡
136	210	88	10001000	ˆ
137	211	89	10001001	‰
138	212	8A	10001010	Š
139	213	8B	10001011	‹
140	214	8C	10001100	Œ
141	215	8D	10001101	•
142	216	8E	10001110	Ž
143	217	8F	10001111	•
144	220	90	10010000	•
145	221	91	10010001	‘
146	222	92	10010010	’
147	223	93	10010011	“
148	224	94	10010100	”
149	225	95	10010101	•
150	226	96	10010110	–
151	227	97	10010111	—
152	230	98	10011000	˜
153	231	99	10011001	™
154	232	9A	10011010	š
155	233	9B	10011011	›
156	234	9C	10011100	œ
157	235	9D	10011101	•
158	236	9E	10011110	ž
159	237	9F	10011111	Ÿ
160	240	A0	10100000	
161	241	A1	10100001	¡

Desimal	Octa	Hexa	Biner	Karakter
163	243	A3	10100011	£
164	244	A4	10100100	¤
165	245	A5	10100101	¥
166	246	A6	10100110	¦
167	247	A7	10100111	§
168	250	A8	10101000	¨
169	251	A9	10101001	©
170	252	AA	10101010	ª
171	253	AB	10101011	«
172	254	AC	10101100	¬
173	255	AD	10101101	
174	256	AE	10101110	®
175	257	AF	10101111	¯
176	260	B0	10110000	°
177	261	B1	10110001	±
178	262	B2	10110010	²
179	263	B3	10110011	³
180	264	B4	10110100	´
181	265	B5	10110101	µ
182	266	B6	10110110	¶
183	267	B7	10110111	·
184	270	B8	10111000	¸
185	271	B9	10111001	¹
186	272	BA	10111010	º
187	273	BB	10111011	»
188	274	BC	10111100	¼
189	275	BD	10111101	½
190	276	BE	10111110	¾
191	277	BF	10111111	¿
192	300	C0	11000000	À
193	301	C1	11000001	Á
194	302	C2	11000010	Â
195	303	C3	11000011	Ã
196	304	C4	11000100	Ä
197	305	C5	11000101	Å
198	306	C6	11000110	Æ
199	307	C7	11000111	Ç
200	310	C8	11001000	È
201	311	C9	11001001	É
202	312	CA	11001010	Ê
203	313	CB	11001011	Ë

Desimal	Octa	Hexa	Biner	Karakter
205	315	CD	11001101	Í
206	316	CE	11001110	Î
207	317	CF	11001111	Ï
208	320	D0	11010000	Đ
209	321	D1	11010001	Ñ
210	322	D2	11010010	Ò
211	323	D3	11010011	Ó
212	324	D4	11010100	Ô
213	325	D5	11010101	Õ
214	326	D6	11010110	Ö
215	327	D7	11010111	×
216	330	D8	11011000	Ø
217	331	D9	11011001	Ù
218	332	DA	11011010	Ú
219	333	DB	11011011	Û
220	334	DC	11011100	Ü
221	335	DD	11011101	Ý
222	336	DE	11011110	Þ
223	337	DF	11011111	ß
224	340	E0	11100000	à
225	341	E1	11100001	á
226	342	E2	11100010	â
227	343	E3	11100011	ã
228	344	E4	11100100	ä
229	345	E5	11100101	å
230	346	E6	11100110	æ
231	347	E7	11100111	ç
232	350	E8	11101000	è
233	351	E9	11101001	é
234	352	EA	11101010	ê
235	353	EB	11101011	ë
236	354	EC	11101100	ì
237	355	ED	11101101	í
238	356	EE	11101110	î
239	357	EF	11101111	ï
240	360	F0	11110000	ð
241	361	F1	11110001	ñ
242	362	F2	11110010	ò
243	363	F3	11110011	ó
244	364	F4	11110100	ô
245	365	F5	11110101	õ

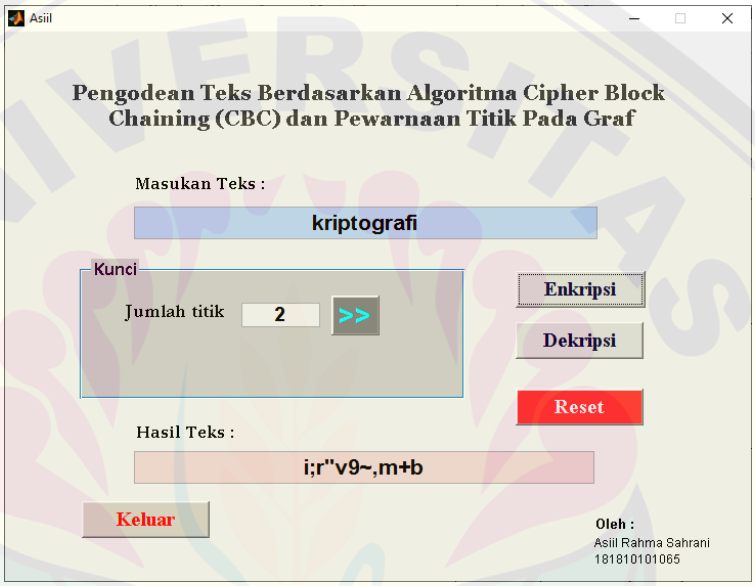
Desimal	Octa	Hexa	Biner	Karakter
247	367	F7	11110111	÷
248	370	F8	11111000	ø
249	371	F9	11111001	ù
250	372	FA	11111010	ú
251	373	FB	11111011	û
252	374	FC	11111100	ü
253	375	FD	11111101	Ý
254	376	FE	11111110	Þ
255	377	FF	11111111	Ÿ

Lampiran B. Hasil Pengodean Teks pada Program

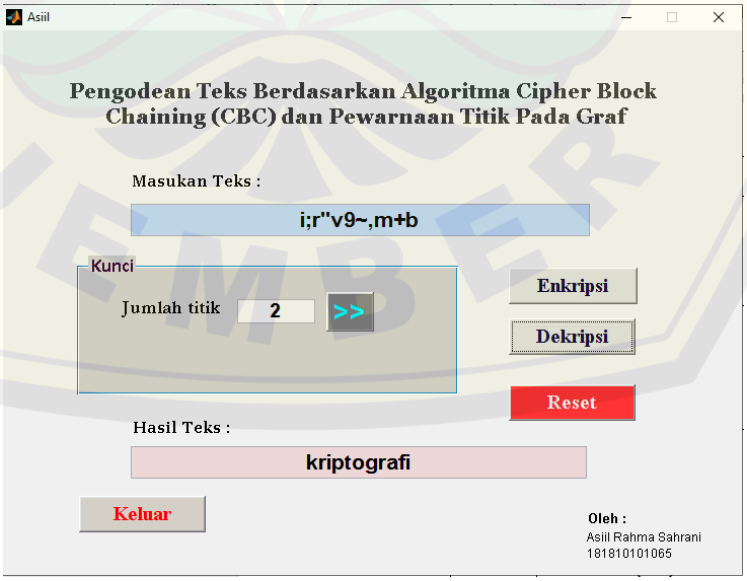
Berikut hasil dari pengodean teks pada program :

1. Plainteks : kriptografi
   Kunci :

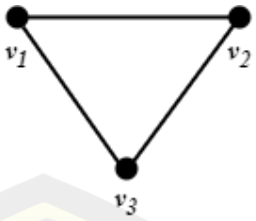
Hasil enkripsi program :



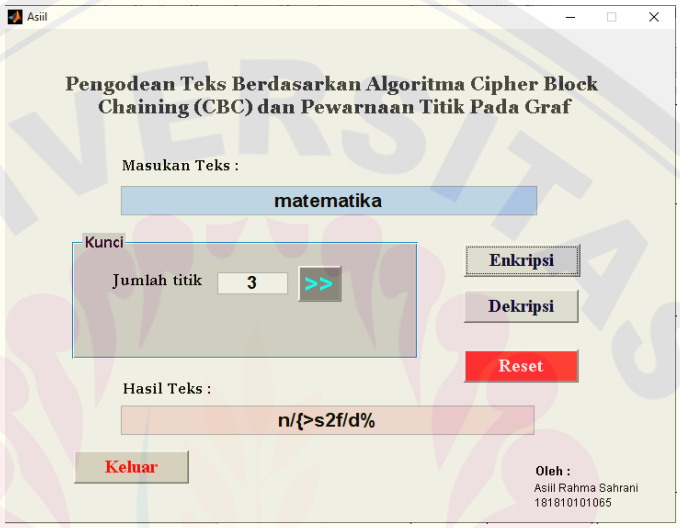
Hasil dekripsi program :



2. Plainteks : matematika
   Kunci :

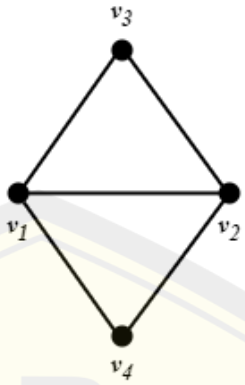


Hasil enkripsi program :



Hasol dekripsi program :

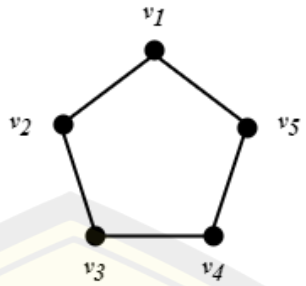
3. Plainteks : program
   Kunci :



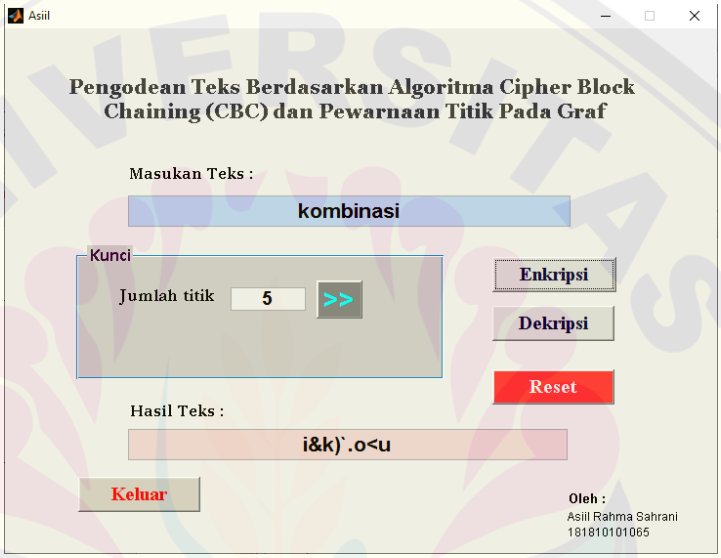
Hasil enkripsi program :

Hasil dekripsi program :

4. Plainteks : kombinasi
   Kunci : Graf 5 titik

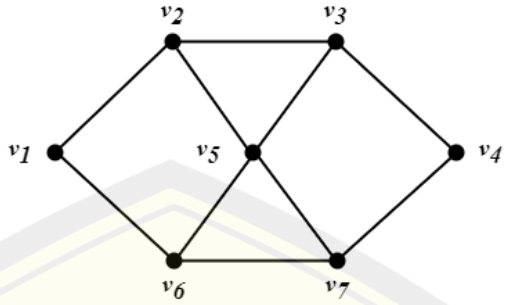


Hasil enkripsi program :

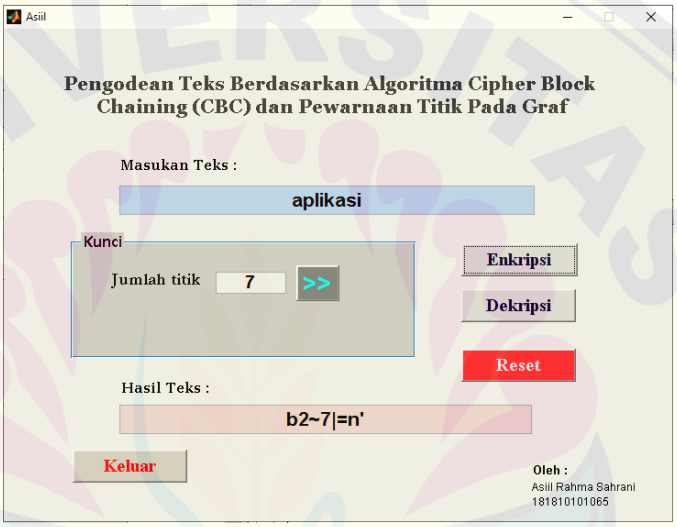


Hasil dekripsi program :

5. Plainteks : aplikasi
   Kunci : Graf 7 titik

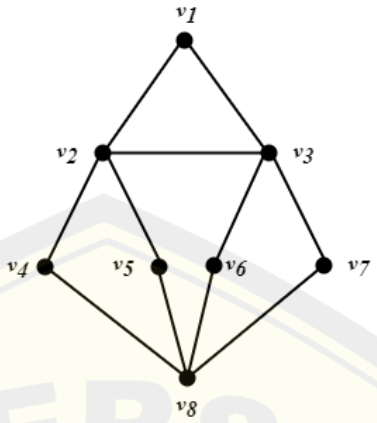


Hasil enkripsi program :



Hasil dekripsi program :

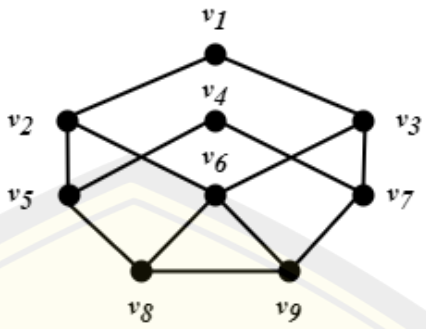
6. Plainteks : universitas
   Kunci : Graf 8 titik



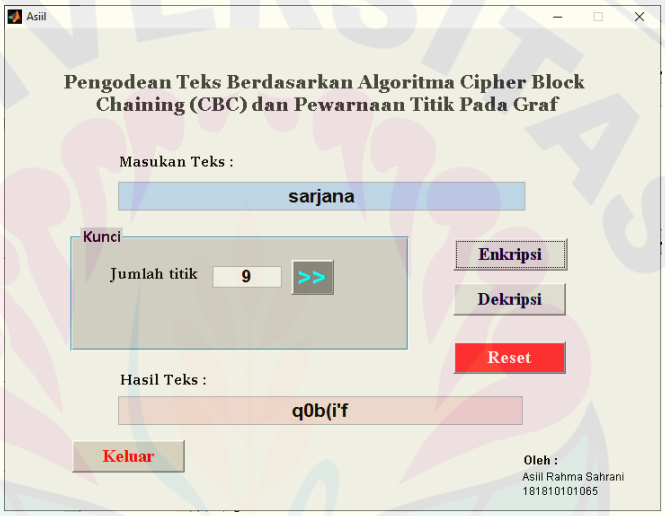
Hasil enkripsi program :

Hasil dekripsi program :

7. Plainteks : sarjana
   Kunci : Graf 9 titik

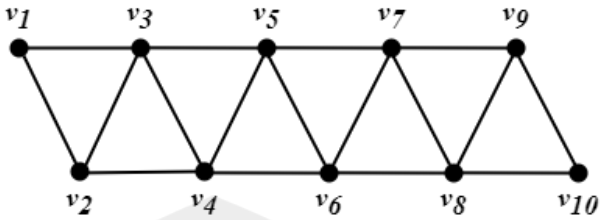


Hasil enkripsi program :



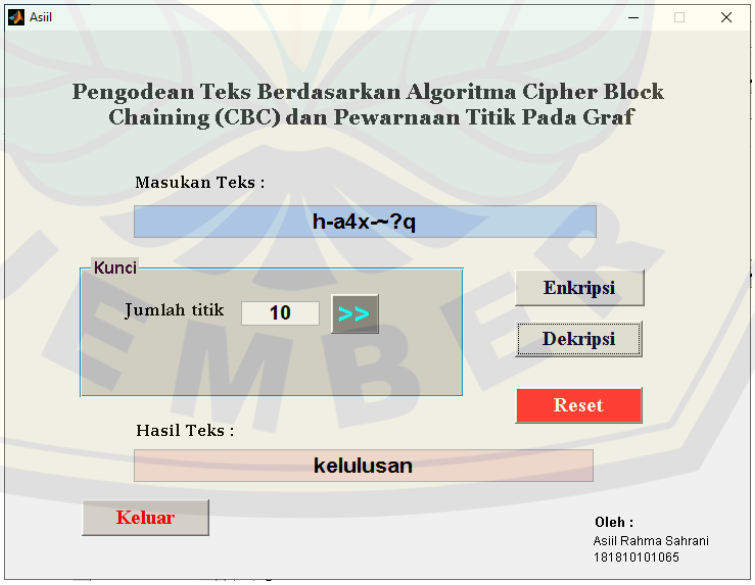
Hasil dekripsi program :

8. Plainteks : kelulusan
   Kunci : Graf 10 titik



Hasil enkripsi program :

Hasil dekripsi program :



Lampiran C. *Source Code* Program

```
function varargout = Asiil(varargin)
% ASIIL MATLAB code for Asiil.fig
%%
gui_Singleton = 1;
gui_State = struct('gui_Name',       mfilename, ...
                   'gui_Singleton',  gui_Singleton, ...
                   'gui_OpeningFcn', @Asiil_OpeningFcn, ...
                   'gui_OutputFcn',  @Asiil_OutputFcn, ...
                   'gui_LayoutFcn',  [] , ...
                   'gui_Callback',   []);
if nargin && ischar(varargin{1})
    gui_State.gui_Callback = str2func(varargin{1});
end

if nargout
    [varargout{1:nargout}] = gui_mainfcn(gui_State,
varargin{:});
else
    gui_mainfcn(gui_State, varargin{:});
end

function Asiil_OpeningFcn(hObject, eventdata, handles,
varargin)
handles.output = hObject;
guidata(hObject, handles);

function varargout = Asiil_OutputFcn(hObject, eventdata,
handles)

varargout{1} = handles.output;

function plain_Callback(hObject, eventdata, handles)

function plain_CreateFcn(hObject, eventdata, handles)

if ispc && isequal(get(hObject,'BackgroundColor'),
get(0,'defaultUicontrolBackgroundColor'))
    set(hObject,'BackgroundColor','white');
end

%Tombol pada jumlah titik
```

```
function tt_Callback(hObject, eventdata, handles)
set(handles.th,'visible','on');
set(handles.h,'visible','on');
set(handles.I,'visible','on');
set(handles.J,'visible','on');
set(handles.text6,'visible','on');set(handles.titik,'enable'
,'off');

function V_Callback(hObject, eventdata, handles)

function V_CreateFcn(hObject, eventdata, handles)

if ispc && isequal(get(hObject,'BackgroundColor'),
get(0,'defaultUicontrolBackgroundColor'))
    set(hObject,'BackgroundColor','white');
end

function c1_Callback(hObject, eventdata, handles)

function c2_Callback(hObject, eventdata, handles)

function cipher_Callback(hObject, eventdata, handles)

function cipher_CreateFcn(hObject, eventdata, handles)

if ispc && isequal(get(hObject,'BackgroundColor'),
get(0,'defaultUicontrolBackgroundColor'))
    set(hObject,'BackgroundColor','white');
end

function h_Callback(hObject, eventdata, handles)

function h_CreateFcn(hObject, eventdata, handles)

if ispc && isequal(get(hObject,'BackgroundColor'),
get(0,'defaultUicontrolBackgroundColor'))
    set(hObject,'BackgroundColor','white');
end

%Mendefinisikan matriks adjacency pada input hubungan antar
titik
function th_Callback(hObject, eventdata, handles)
global v;
V= str2num(get(handles.titik,'string'));
```

```
i= str2num(get(handles.I,'string')); j=
str2num(get(handles.J,'string'));
v(i,j)= str2num(get(handles.h,'string'));
if v(i,j)==1
    v(j,i)=1;
elseif v(i,j)==0
    v(j,i)=0;
else
    msgbox('Masukkan nilai 1 atau 0','Peringatan','warn');
    return
end;
if j==V && i==j-1
    set(handles.th,'visible','off');
set(handles.h,'visible','off');
    set(handles.I,'visible','off');
set(handles.J,'visible','off');

set(handles.text6,'visible','off');set(handles.titik,'enable
','on');
    set(handles.tt,'enable','on'); end;
if j<V
    j=j+1;
elseif i<V
    i=i+1; j=i+1; end; set(handles.h,'string','');
set(handles.I,'string',num2str(i));
set(handles.J,'string',num2str(j));

%=======================================================
%             Tahap Dekripsi
%=======================================================
function text2_Callback(hObject, eventdata, handles)
global v;
ttk=str2num(get(handles.titik,'string'));
cipher=get(handles.plain,'string');
c(ttk)=0; warna=0;
if isempty(ttk) == 1 || isempty(cipher) == 1
 warndlg('Masukkan Teks dan Kunci','Peringatan');
else
    for i=1 : ttk
    row(i)=0;
    for j= 1 : ttk
        if v(i,j)==1
            row(i)= row(i)+1; end;
    end; end;
for z=1:ttk
```

```
    [x,y]=max(row); row(y)=0;
    if c(y)==0
        warna=warna+1; c(y)=warna;
        for k=1:ttk
            if v(y,k)==0 && c(k)==0
                c(k)=warna; end; end; end; end;
Plain='';
for i=1:length(cipher)-1
    c(i)=mod(bitxor(cipher(i)-32,cipher(i+1)-32),95);
    Plain=[Plain,char(c(i)+32)]; end;
Plain=[char(mod(bitxor(cipher(1)-32,warna),95)+32),Plain];
 set(handles.cipher,'string',Plain);
end

%=====================================================
%             Tahap Enkripsi
%=====================================================
function text5_Callback(hObject, eventdata, handles)
global v;
ttk=str2num(get(handles.titik,'string'));
plain=get(handles.plain,'string');
c(ttk)=0; warna=0;
if isempty(ttk) == 1 || isempty(plain) == 1
 warndlg('Masukkan Teks dan Kunci','Peringatan');
else
    for i=1 : ttk
    row(i)=0;
    for j= 1 : ttk
        if v(i,j)==1
            row(i)= row(i)+1; end;
    end; end;
for z=1:ttk
    [x,y]=max(row); row(y)=0;
    if c(y)==0
        warna=warna+1; c(y)=warna;
        for k=1:ttk
            if v(y,k)==0 && c(k)==0
                c(k)=warna; end; end; end; end;
c(1)=mod(bitxor(plain(1)-32,warna),95);
cipher=char(c(1)+32);
for i=2:length(plain)
    c(i)=mod(bitxor(plain(i)-32,c(i-1)),95);
    cipher=[cipher,char(c(i)+32)]; end;
set(handles.cipher,'string',cipher);
end
```

```
%Tombol untuk reset
function pushbutton3_Callback(hObject, eventdata, handles)
set(handles.plain,'string','');
set(handles.cipher,'string','');
set(handles.titik,'string','');
set(handles.h,'string','');
set(handles.I,'string','1');
set(handles.J,'string','2')

%Tombol untuk keluar
function pushbutton4_Callback(hObject, eventdata, handles)
close;
```