

PRAKTIKUM KRIPTOGRAFI

TUGAS 3



Nama : Novem Romadhofi Kika

NPM : 140810220083

Kelas : A

UNIVERSITAS PADJAJARAN

Fakultas Matematika Dan Ilmu Pengetahuan Alam

Program Studi S-1 Teknik Informatika

2024

2. Buatlah program untuk enkripsi, dekripsi, dan mencari kunci Hill Cipher (bahasa pemrograman bebas)

Source Code :

```
/*
Nama : Novem Romadhofi Kika
NPM : 140810220083
Kelas : A
Program : Hill Cipher
*/

#include <iostream>
#include <vector>
#include <iomanip> // Untuk tampilan format yang lebih baik

using namespace std;

// Fungsi untuk mencari invers matriks modulo 26
int modInverse(int a, int m) {
    a = a % m;
    for (int x = 1; x < m; x++) {
        if ((a * x) % m == 1)
            return x;
    }
    return -1;
}

// Fungsi untuk mencari determinan matriks modulo 26
int determinantModulo(vector<vector<int>> matrix, int size) {
    if (size == 2) {
        return (matrix[0][0] * matrix[1][1] - matrix[0][1] * matrix[1][0]) % 26;
    }
    return -1;
}

// Fungsi untuk memastikan nilai hasil mod tidak negatif
int mod26(int x) {
    return (x % 26 + 26) % 26;
}

// Fungsi untuk enkripsi teks menggunakan Hill Cipher
```

```

string encrypt(vector<vector<int>> keyMatrix, string message) {
    int n = keyMatrix.size();

    // Jika panjang pesan tidak kelipatan 2, tambahkan padding
    if (message.size() % n != 0) {
        message += 'X'; // Menambahkan karakter 'X' sebagai padding
    }

    string cipherText = "";

    for (int i = 0; i < message.size(); i += n) {
        vector<int> messageVector(n);
        for (int j = 0; j < n; j++) {
            messageVector[j] = message[i + j] - 'A';
        }

        vector<int> resultVector(n, 0);
        for (int row = 0; row < n; row++) {
            for (int col = 0; col < n; col++) {
                resultVector[row] += keyMatrix[row][col] * messageVector[col];
            }
            resultVector[row] = mod26(resultVector[row]); // Menggunakan mod26 untuk
menghindari negatif
        }

        for (int r = 0; r < n; r++) {
            cipherText += (resultVector[r] + 'A');
        }
    }
    return cipherText;
}

// Fungsi untuk dekripsi teks menggunakan Hill Cipher
string decrypt(vector<vector<int>> keyMatrix, string cipherText) {
    int n = keyMatrix.size();

    // Hitung determinan dan cari invers matriks
    int det = determinantModulo(keyMatrix, n);
    int detInverse = modInverse(det, 26);

    // Matrik invers harus dihitung di sini (untuk simplicity, hanya diterapkan untuk ukuran
2x2)

```

```

vector<vector<int>> inverseMatrix = {
    {keyMatrix[1][1], -keyMatrix[0][1]},
    {-keyMatrix[1][0], keyMatrix[0][0]}
};

// Apply mod 26 dan kalikan dengan invers determinan
for (int i = 0; i < n; i++) {
    for (int j = 0; j < n; j++) {
        inverseMatrix[i][j] = mod26(inverseMatrix[i][j] * detInverse);
    }
}

// Dekripsi
string plainText = "";
for (int i = 0; i < cipherText.size(); i += n) {
    vector<int> cipherVector(n);
    for (int j = 0; j < n; j++) {
        cipherVector[j] = cipherText[i + j] - 'A';
    }

    vector<int> resultVector(n, 0);
    for (int row = 0; row < n; row++) {
        for (int col = 0; col < n; col++) {
            resultVector[row] += inverseMatrix[row][col] * cipherVector[col];
        }
        resultVector[row] = mod26(resultVector[row]);
    }

    for (int r = 0; r < n; r++) {
        plainText += (resultVector[r] + 'A');
    }
}
return plainText;
}

// Fungsi untuk input matriks kunci dari pengguna
void inputKeyMatrix(vector<vector<int>>& keyMatrix) {
    cout << "Masukkan elemen matriks kunci 2x2 (4 angka): ";
    for (int i = 0; i < 2; i++) {
        for (int j = 0; j < 2; j++) {
            cin >> keyMatrix[i][j];
        }
    }
}

```

```

    }
}

// Fungsi untuk mencari key berdasarkan pesan terenkripsi dan pesan asli
void cariKey(string plaintext, string ciphertext) {
    // Proses Brute Force untuk menemukan kunci
    vector<vector<int>> keyMatrix(2, vector<int>(2));
    bool found = false;

    for (int a = 0; a < 26; a++) {
        for (int b = 0; b < 26; b++) {
            for (int c = 0; c < 26; c++) {
                for (int d = 0; d < 26; d++) {
                    keyMatrix[0][0] = a;
                    keyMatrix[0][1] = b;
                    keyMatrix[1][0] = c;
                    keyMatrix[1][1] = d;

                    // Enkripsi pesan dengan matriks kunci saat ini
                    string encryptedText = encrypt(keyMatrix, plaintext);

                    if (encryptedText == ciphertext) {
                        found = true;
                        break;
                    }
                }
            }
            if (found) break;
        }
        if (found) break;
    }
    if (found) break;
}

if (found) {
    cout << "Kunci ditemukan:\n";
    cout << "[ " << keyMatrix[0][0] << " " << keyMatrix[0][1] << " ]\n";
    cout << "[ " << keyMatrix[1][0] << " " << keyMatrix[1][1] << " ]\n";
} else {
    cout << "Kunci tidak ditemukan.\n";
}
}

```

```

// Fungsi untuk menampilkan menu
void tampilkanMenu() {
    cout << "\n===== " << endl;
    cout << "      HILL CIPHER MENU      " << endl;
    cout << "===== " << endl;
    cout << setw(15) << "1. " << "Enkripsi" << endl;
    cout << setw(15) << "2. " << "Dekripsi" << endl;
    cout << setw(15) << "3. " << "Cari Key" << endl;
    cout << setw(15) << "4. " << "Keluar" << endl;
    cout << "===== " << endl;
    cout << "Masukkan pilihan: ";
}

int main() {
    int pilihan;
    vector<vector<int>> keyMatrix(2, vector<int>(2));
    string message, result;

    do {
        tampilkanMenu();
        cin >> pilihan;

        switch (pilihan) {
            case 1: {
                inputKeyMatrix(keyMatrix);
                cout << "Masukkan pesan yang akan dienkrpsi (huruf kapital tanpa spasi): ";
                cin >> message;
                result = encrypt(keyMatrix, message);
                cout << "\nHasil Enkripsi: " << result << endl;
                break;
            }

            case 2: {
                inputKeyMatrix(keyMatrix);
                cout << "Masukkan pesan yang akan didekripsi (huruf kapital tanpa spasi): ";
                cin >> message;
                result = decrypt(keyMatrix, message);
                cout << "\nHasil Dekripsi: " << result << endl;
                break;
            }

            case 3: {

```

```

        cout << "Masukkan pesan asli: ";
        cin >> message;
        cout << "Masukkan pesan terenkripsi: ";
        string cipher;
        cin >> cipher;
        cariKey(message, cipher);
        break;
    }

    case 4:
        cout << "\nTerima kasih.\n";
        break;

    default:
        cout << "Pilihan tidak valid. Silakan coba lagi.\n";
        break;
    }
} while (pilihan != 4);

return 0;
}

```

Hasil Program :

Enkripsi :

```

PS C:\Semester 5\Praktikum Kriptografi\Hill Cipher> g++ HillCipher.cpp -o HillCipher
PS C:\Semester 5\Praktikum Kriptografi\Hill Cipher> ./HillCipher

=====
HILL CIPHER MENU
=====

    1. Enkripsi
    2. Dekripsi
    3. Cari Key
    4. Keluar

=====
Masukkan pilihan: 1
Masukkan elemen matriks kunci 2x2 (4 angka): 6 24 1 13
Masukkan pesan yang akan dienkripsi (huruf kapital tanpa spasi): ALBERTI

Hasil Enkripsi: ENYBMECV

```

Deskripsi :

```
=====
HILL CIPHER MENU
=====
1. Enkripsi
2. Dekripsi
3. Cari Key
4. Keluar
=====
Masukkan pilihan: 2
Masukkan elemen matriks kunci 2x2 (4 angka): 6 24 1 13
Masukkan pesan yang akan didekripsi (huruf kapital tanpa spasi): ENYBMECV

Hasil Dekripsi: AEYSSOKG
```

Key :

```
=====
HILL CIPHER MENU
=====
1. Enkripsi
2. Dekripsi
3. Cari Key
4. Keluar
=====
Masukkan pilihan: 3
Masukkan pesan asli: ALBERTI
Masukkan pesan terenkripsi: ENYBMECV
Kunci ditemukan:
[ 6 24 ]
[ 1 13 ]
```

Keluar :


```
=====
Masukkan pilihan: 3
Masukkan pesan asli: ALBERTI
Masukkan pesan terenkripsi: ENYBMECV
Kunci ditemukan:
[ 6 24 ]
[ 1 13 ]

=====
HILL CIPHER MENU
=====
1. Enkripsi
2. Dekripsi
3. Cari Key
4. Keluar
=====
Masukkan pilihan: 4
```

3. Push program tersebut ke repository **NPM-Kripto24** dan sertakan juga screenshot di dalamnya. <https://github.com/Novem03/83-Kripto2024.git>