

Menu:

Menggunakan switch untuk mengganti antara menggunakan fungsi Enkripsi(1), Dekripsi(2), atau Cari Key(3). Respon selain dari 3 angka tersebut akan memunculkan "Respon tidak sesuai"

Enkripsi:

```
string Encrypt(int key[][2], string text){
    // ascii dec range 65 ... 90 (ascii for A to Z)
    // ascii dec range 97 ... 122 (ascii for a to z)
    string encrypted;
    int con[2];
    int con2 = 0;
    int z = 0;
    if(text.length()%2 == 0){
        z = text.length()/2;
    }else{
        z = (text.length()/2) + 1;
    }
    for(int i = 0; i < z; i++){
        con[0] = 23; con[1] = 23;
        for(int j = 0; j < 2; j++){ // alphabet -> ASCII number
            if((int)text[j + (i*2)] >= 97 && (int)text[j + (i*2)] <=
122) { // alfabet kecil
                con[j] = (int)text[j + (i*2)] - 97;
            }else if((int)text[j + (i*2)] >= 65 && (int)text[j + (i*2)]
<= 90) { // alfabet kapital
                con[j] = (int)text[j + (i*2)] - 65;
            }else if(isblank(text[j + (i*2)])){
                con[j] = 23;
            }
        }
        for(int j = 0; j < 2; j++){// Calculation
            con2 = ((key[j][0]*con[0] + key[j][1]*con[1])%26) + 65;
            encrypted += (char)con2;
        }
    }
    return encrypted;}
```

Setelah deklarasi variable yang dibutuhkan, sistem menyesuaikan jumlah loop yang akan digunakan dengan cara dibagi dengan ukuran matriks. Setelah itu, text diubah menjadi angka 0 – 25 lalu dikalikan dengan matriks key. Setelah loop selesai maka akan ditambahkan ke variable hasil.

elCheck:

```

void elCheck(int key[][2], string text){
    int det = 0;
    det += (key[0][0]*key[1][1]);
    det -= (key[0][1]*key[1][0]);
    int result = min(det%26, 26);

    while (result > 0) {
        if (det % result == 0 && 26 % result == 0) {
            break;
        }
        result--;
    }
    cout << "\nGCD 26 = " << result << endl;
    if(result != 1){
        cout << "Matrix is not inversible mod 26\n";
    }else{
        cout << "Can be Decrypted\n";
        cout << "Hasil Enkripsi : " << endl << Decrypt(key, text) <<
endl;
    }
}

```

Elcheck adalah fungsi yang digunakan untuk memastikan matriks key mempunyai inverse modulo 26 dengan cara mencari determinan yang di mod 26 lalu dicari factor perkalian tertingginya. Jika 1 maka memenuhi sehingga dapat melakukan fungsi Decrypt, tetapi jika tidak maka akan muncul "Matrix is not inversible mod 26".

Decrypt:

```

string Decrypt(int key[][2], string text){
    // ascii dec range 65 ... 90 (ascii for A to Z)
    // ascii dec range 97 ... 122 (ascii for a to z)
    string decrypted;
    int con[2];
    int con2 = 0;
    int** invKey = inverse(key);

    int z = 0;
    if(text.length()%2 == 0){
        z = text.length()/2;
    }else{
        z = (text.length()/2) + 1;
    }
    for(int i = 0; i < z; i++){
        con[0] = 23; con[1] = 23;
        for(int j = 0; j < 2; j++){ // alphabet -> ASCII number
            if((int)text[j + (i*2)] >= 97 && (int)text[j + (i*2)] <=
122) { // alfabet kecil
                con[j] = (int)text[j + (i*2)] - 97;
            }
        }
    }
}

```

```
        }else if((int)text[j + (i*2)] >= 65 && (int)text[j + (i*2)]
<= 90) { // alfabet kapital
            con[j] = (int)text[j + (i*2)] - 65;
        }else if(isblank(text[j + (i*2)])){
            con[j] = 23;
        }
    }
    for(int j = 0; j < 2; j++){// Calculation
        con2 = ((invKey[j][0]*con[0] + invKey[j][1]*con[1])%26) +
65;
        decrypted += (char)con2;
    }

}
return decrypted;}
```

Fungsi ini mirip dengan enkripsi, tetapi sebelumnya ditambahkan program untuk mencari invers modulo 26 dan invers matriks pada matriks key sehingga membentuk matriks key baru (InvKey) untuk melakukan proses perkalian matriks seperti fungsi Encrypt.