

```

int main(){
    string plainText;
    string cipherText;
    int menu;
    int key[2][2];

    cout << "\n\t---- Program Enkripsi Hill Cipher (key 2x2) ----\n";
    cout << "\nMenu\n1. Enkripsi dan Dekripsi pesan\n2. Mencari matriks kunci ";
    cout << "\n\nNomor menu : ";
    cin >> menu;
    switch(menu){
        case 1 :
            cout << "\nMasukkan plain text : ";
            cin >> plainText;

            cout << "Nilai Kunci K\n";
            for(int i = 0; i<2; i++){
                for(int j = 0; j<2; j++){
                    cout << "K [" << i << "][" << j << "] : ";
                    cin >> key[i][j];
                }
            }
            if (determinan(key) == 0) {
                cout << "\ndeterminan = 0!!\n";
                break;
            }
            enkripsi(plainText, key);
            break;

        case 2 :
            cout << "\nMasukkan plain text : ";
            cin >> plainText;

            cout << "\nMasukkan cipher text : ";
            cin >> cipherText;
            break;

        default : break;
    }
}

```

Switch case untuk 2 kasus

-mencari cipher teks dan di dekripsi lagi

Akan di cek dahulu jika determinan tidak sama dengan 0, jika sama dengan nol maka akan keluar program.

-mencari key dari plain teks dan cipher teks

```
int determinan(int key[2][2]){
    return key[0][0] * key[1][1] - key[0][1] * key[1][0];
}
```

Me return determinan matriks key 2x2

```
int invMod(int a, int b){
    for (int i = 1; i < b; i++){
        if (((a % b) * (i % b)) % b == 1) return i;
    }
}
```

Me return invers modulo dari 2 angka

```
void enkripsi(string plainText, int key[2][2]){
    string cipherText = "";
    int matriksPesan[2][1];
```

Matrikspesan untuk menampung nilai angka dari plainteks

Looping pertama

```
int matriksPesan[2][1];
for(int i = 0; i < plainText.length()/2; i++){
    for(int j = i*2; j < i*2+1; j++){
        matriksPesan[j%2][0] = (plainText[j] % 65;
    }
}
```

Looping pada matriks pesan sehingga terbagi menjadi 2 baris 1 kolom, looping pertama dengan iterasi i untuk jumlah matriks setelah plainteks terbagi.

```
        matriksPesan[j%2][0] = (plainText[j] % 65;
    }
    for(int k = 0; k < 2; k++){
        cipherText += (key[k][0] * matriksPesan[0][0] + key[k][1] * matriksPesan[1][0]) % 26 + 65;
    }
}
cout << "Hasil Enkripsi : " << cipherText;
dekripsi(cipherText, key);
```

Cipher teks akan langsung dihitung dan ditambahkan ke string cipherText untuk bisa dioutput nanti

```

void dekripsi(string cipherText, int key[2][2]){
    int det = determinan(key);
    int iMod = invMod(det, 26);
    int matriksPesan[2][1];
    int dKey[2][2];
    string resultText = "";

    dKey[0][0] = key[1][1] * iMod % 26;
    dKey[0][1] = ((key[0][1] * -iMod) % 26 + 26) % 26;
    dKey[1][0] = ((key[1][0] * -iMod) % 26 + 26) % 26;
    dKey[1][1] = key[0][0] * iMod % 26;

    for(int i = 0; i<cipherText.length()/2; i++){
        for(int j = i*2; j<=i*2+1; j++){
            matriksPesan[j%2][0] = (cipherText[j]) % 65;
        }
        for(int k = 0; k<2; k++){
            resultText += (dKey[k][0] * matriksPesan[0][0] + dKey[k][1] * matriksPesan[1][0]) % 26 + 65;
        }
    }
    cout << "\nHasil Dekripsi : " << resultText;
}

```

Dkey adalah key dekripsi

Looping pada fungsi dekripsi hamper sama dengan enkripsi hanya berbeda dari isi matikspesan yang berupa cipher teks dan key diubah menjadi dkey atau key dekripsi