

KRIPTOGRAFI KRIPTOPUBLIK

Abdul Malik

18.83.0276

<https://github.com/alinol0>

Sri Suraningsih

18.83.0290

<https://github.com/SriSurani>
[ngsih](#)

Nur Dian Yustikarini

18.83.0298

<https://github.com/>
[NurDian-298](#)

Diah Pingkan Sari

18.83.0326

<https://github.com/DiahPingkanSari>

Mohamad Dzulfiquar S

18.83.0331

<https://github.com/zulfiqqar02>

Tools:

Dev C++

QT Creator 4.12.3

Tema:

Cyber Security

Abstraksi:

Kami membuat aplikasi enkripsi dan deskripsi dengan menerapkan kriptografi kunci publik. KriptoPublik adalah sistem enkripsi (penyandian) yang menggunakan dua kunci, yaitu kunci publik dan kunci privat. Pengirim akan melakukan proses enkripsi dan penerima akan mendeskripsikan pesan tersebut. Nantinya pengirim dan penerima saling bertukar kunci publiknya untuk menghitung kunci enkripsi dan deskripsinya.

Konten:

Versi CLI

```
#include <conio.h>
#include <iostream>
using namespace std;

int main() {
    int m, n, x, y, pil;
    long int total = 1;
    long int enkrip = 1;

    char karakter;
    long int pemodulo, pengirim, penerima, key, chiper, deskrip;
```

Pada program ini kami memakai 2 library yaitu conio.h dan iostream. Kemudian memakai 3 nilai yaitu integer, long integer dan char untuk mendeklarasikan nilai yang kami pakai seperti pada gambar diatas.

```
cout << "===== " << endl;
cout << "      Kriptografi dengan Algoritma KriptoPublik      " << endl;
cout << "===== " << endl << endl;
cout << "1. ENKRIPSI" << endl;
cout << "2. DESKRIPSI" << endl;
cout << "-----" << endl;

cout << "Masukkan Pilihan : ";
cin >> pil;
```

Kemudian untuk output memunculkan bahwa ini adalah program kriptografi dengan algoritma kriptopublik, dan membuat 2 pilihan. Pilihan pertama untuk enkripsi dan pilihan 2 untuk deskripsi, masukan pilihan yang ingin dipilih.

```
switch(pil){
    case 1 :
        cout << "Masukkan Pesan yang Akan Dienkripsi : ";
        cin >> karakter;
        cout << "Hasil konversi ke dalam bentuk ASCII : " << int(karakter) << endl;
        cout << "-----" << endl << endl;
        cout << "Menghitung Kunci Publik (Pengirim)" << endl << endl;
        cout << "Masukkan nilai g          : ";
        cin >> m;
        cout << endl << "Masukkan kunci privat (Pengirim) : ";
        cin >> n;
        for (int i = 1; i <= n; i++) {
            total = total * m;
        }
```

Kemudian membuat switch agar bisa memilih. Untuk case pertama diminta untuk menginputkan pesan yang akan dienkripsi, hanya untuk 1 karakter saja. Kemudian hasil inputan tersebut diubah kebentuk ascii. Kemudian langkah selanjutnya menghitung kunci public pengirim dengan cara menginputkan nilai g, kunci privat pengirim, dan dilakukan perulangan untuk menghitung hasilnya.

```
cout << "\nBilangan pemodulo (p)          : ";
cin >> pemodulo;

pengirim = total % pemodulo;

cout<<"\n-----" << endl;
cout << endl << "Kunci publik (Pengirim) = g^x mod (p)"<<endl;
cout << "          = (" << m << " )^" << n << " mod " << pemodulo <<endl;
cout << "          = " << pengirim << endl;

endl (cout);
```

Kemudian menginputkan bilangan pemodulo dan dihitung bahwa $\text{pengirim} = \text{total} \% \text{pemodulo}$. Itu artinya sama dengan rumus kunci public pengirim $= g^x \text{ mod } p$ dan dihitung.

```
cout<<"-----" << endl;
cout<<"Pengirim dan Penerima melakukan pertukaran kunci publik"<<endl;
cout<<"Menghitung Kunci Enkripsi-Deskripsi"<<endl;
cout<<"Masukkan Kunci Publik Penerima : ";
cin>>penerima;
cout<<"Masukkan kunci privat kembali : ";
cin>>x;

for (int j = 1; j<= x; j++) {
    enkrip = enkrip * penerima;
}

key = enkrip % pemodulo;
```

Kemudian pada langkah ini menghitung kunci enkripsi dan deskripsinya dengan melakukan pertukaran kunci public. Menginputkan kunci public penerima dan diminta memasukan kunci privat pengirim kembali dan dihitung dengan perulangan dan untuk $\text{enkrip} = \text{enkrip} * \text{penerima}$. Dan menghitung $\text{key} = \text{enkrip} \% \text{pemodulo}$, maka nilai key nya didapatkan.

```

cout<<"\nKunci Enkripsi-Deskripsi = Kunci Publik (Penerima) ^ Kunci Privat (Pengirim) mod (p)"<<endl;
cout<<"          = (" << penerima << ")^" << n << " mod " << pemodulo <<endl;
cout<<"          = " << key <<endl;

endl (cout);

cout<<"-----" << endl;
cout<<"\nProses Enkripsi = Pesan XOR Kunci Enkripsi Deskripsi"<<endl;
chiper = int(karakter) ^ key;
cout<<"      = "<<chiper<<endl;
cout<<"Hasil Enkripsi dari Pesan "<<karakter<<" adalah : "<<char(chiper)<<endl;

```

Dan melakukan perhitungan dengan rumus $\text{kunci public penerima}^{\text{kunci privat pengirim}} \bmod p$. setelah di hitung maka akan di XOR nilainya dan hasil enkripsi dari pesan akan dirubah menjadi karakter. Proses enkripsipun selesai.

```

break;
case 2 :
    cout<<"Masukkan Pesan yang akan dideskripsi : ";
    cin>>karakter;
    cout<<"Hasil konversi ke dalam bentuk ASCII : "<<int(karakter)<<endl;
    cout<<"-----" <<endl<<endl;
    cout << "Menghitung Kunci Publik (Penerima)" << endl << endl;
    cout << "Masukkan nilai g          : ";
    cin >> m;
    cout << endl << "Masukkan kunci privat (Penerima) : ";
    cin >> n;
    for (int i = 1; i <= n; i++) {
        total = total * m;
    }

```

Kemudian untuk case 2 memproses dengan memasukan nilai pesan yang akan dideskripsi. Ingat hanya bisa mendeskripsi 1 karakter saja, kemudian dirubah ke bentuk ascii dan menghitung pula kunci public penerima dengan memasukan nilai g, kunci privat penerima dan dihitung dengan melakukan perulangan.

```

cout << "\nBilangan pemodulo (p)          : ";
cin >> pemodulo;

pengirim = total % pemodulo;

cout<<"\n-----" << endl;
cout << endl << "Kunci publik (Penerima) = g^x mod (p)"<<endl;
cout <<"          = (" << m << ")^" << n << " mod " << pemodulo <<endl;
cout <<"          = " << pengirim << endl;

endl (cout);

```

Kemudian menginputkan bilangan pemodulo dan dihitung bahwa $\text{pengirim} = \text{total} \% \text{pemodulo}$. Itu artinya sama dengan rumus kunci public penerima = $g^x \bmod p$ dan dihitung.

```
cout<<"-----" << endl;
cout<<"Pengirim dan Penerima melakukan pertukaran kunci publik"<<endl;
cout<<"Menghitung Kunci Enkripsi-Deskripsi"<<endl;
cout<<"Masukkan Kunci Publik Pengirim : ";
cin>>penerima;
cout<<"Masukkan Kunci privat kembali : ";
cin>>y;

for (int j = 1; j<= y; j++) {
    enkrip = enkrip * penerima;
}

key = enkrip % pemodulo;
```

Kemudian pada langkah ini menghitung kunci enkripsi dan deskripsinya dengan melakukan pertukaran kunci public. Menginputkan kunci public pengirim dan diminta memasukan kunci privat penerima kembali dan dihitung dengan perulangan dan untuk $\text{enkrip} = \text{enkrip} * \text{penerima}$. Dan dihitung $\text{key} = \text{enkrip} \% \text{pemodulo}$, maka nilai key nya didapatkan.

```
cout<<"\nKunci Enkripsi-Deskripsi = Kunci Publik (Pengirim) ^ Kunci Privat (Penerima) mod (p)"<<endl;
cout<<"      = (" << penerima << ")^" << n << " mod " << pemodulo <<endl;
cout<<"      = " << key <<endl;

endl (cout);

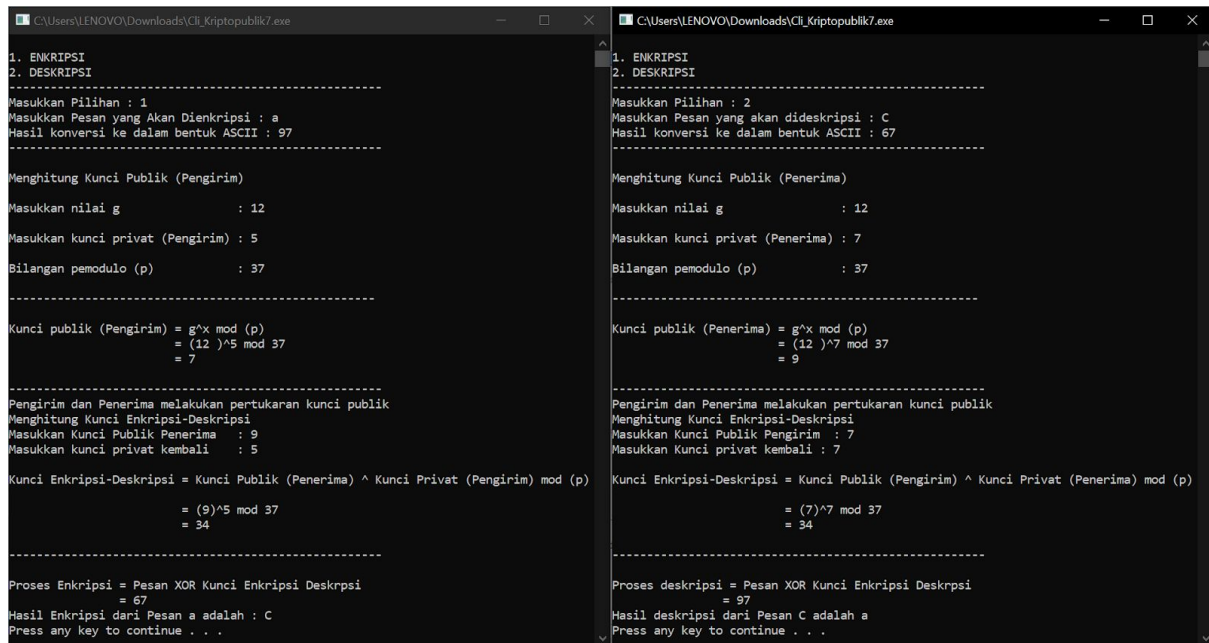
cout<<"-----" << endl;
cout<<"\nProses deskripsi = Pesan XOR Kunci Enkripsi Deskripsi"<<endl;
chiper = int(karakter) ^ key;
cout<<"      = " << chiper <<endl;
cout<<"Hasil deskripsi dari Pesan " << karakter << " adalah " << char(chiper) <<endl;

}

system("pause");
getch();
}
```

Dan melakukan perhitungan dengan rumus $\text{kunci public pengirim}^{\text{kunci privat penerima}} \bmod p$. setelah di hitung maka akan di XOR nilainya dan hasil deskripsinya dari pesan akan dirubah menjadi karakter. Proses deskripsinya selesai.

Output CLI :



```
1. ENKRIPSI
2. DESKRIPSI
-----
Masukkan Pilihan : 1
Masukkan Pesan yang Akan Dienkripsi : a
Hasil konversi ke dalam bentuk ASCII : 97
-----

Menghitung Kunci Publik (Pengirim)

Masukkan nilai g          : 12
Masukkan kunci privat (Pengirim) : 5
Bilangan pemodulo (p)      : 37
-----

Kunci publik (Pengirim) =  $g^x \text{ mod } (p)$ 
                        =  $(12)^5 \text{ mod } 37$ 
                        = 7
-----

Pengirim dan Penerima melakukan pertukaran kunci publik
Menghitung Kunci Enkripsi-Deskripsi
Masukkan Kunci Publik Penerima : 9
Masukkan kunci privat kembali : 5
-----

Kunci Enkripsi-Deskripsi = Kunci Publik (Penerima) ^ Kunci Privat (Pengirim) mod (p)
                        =  $(9)^5 \text{ mod } 37$ 
                        = 34
-----

Proses Enkripsi = Pesan XOR Kunci Enkripsi Deskripsi
                  = 67
Hasil Enkripsi dari Pesan a adalah : C
Press any key to continue . . .
```

```
1. ENKRIPSI
2. DESKRIPSI
-----
Masukkan Pilihan : 2
Masukkan Pesan yang akan didekripsi : C
Hasil konversi ke dalam bentuk ASCII : 67
-----

Menghitung Kunci Publik (Penerima)

Masukkan nilai g          : 12
Masukkan kunci privat (Penerima) : 7
Bilangan pemodulo (p)      : 37
-----

Kunci publik (Penerima) =  $g^x \text{ mod } (p)$ 
                        =  $(12)^7 \text{ mod } 37$ 
                        = 9
-----

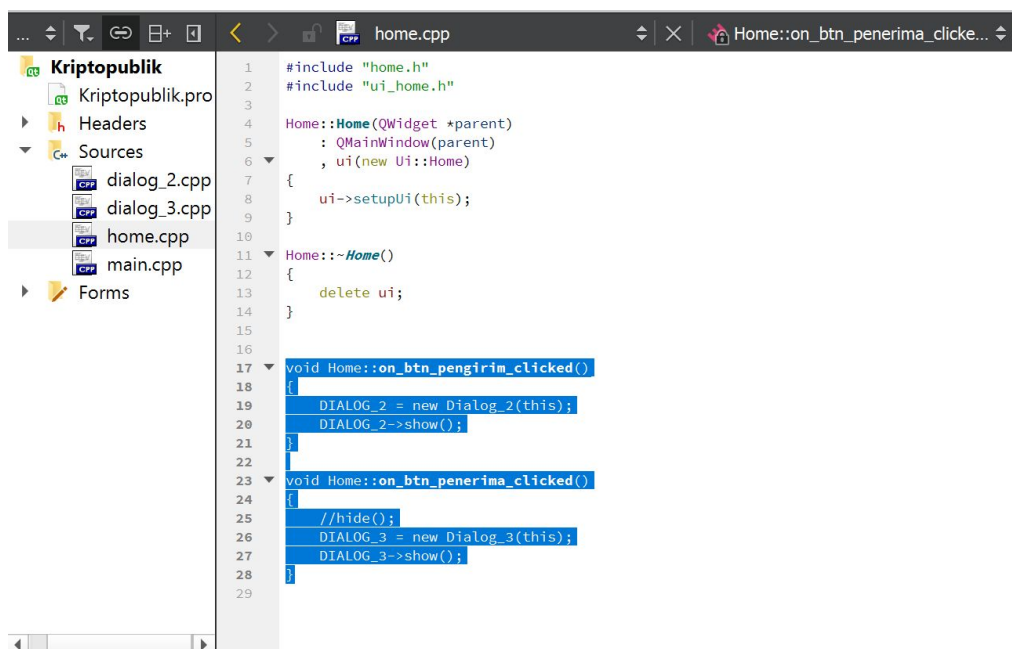
Pengirim dan Penerima melakukan pertukaran kunci publik
Menghitung Kunci Enkripsi-Deskripsi
Masukkan Kunci Publik Pengirim : 7
Masukkan Kunci privat kembali : 7
-----

Kunci Enkripsi-Deskripsi = Kunci Publik (Pengirim) ^ Kunci Privat (Penerima) mod (p)
                        =  $(7)^7 \text{ mod } 37$ 
                        = 34
-----

Proses deskripsi = Pesan XOR Kunci Enkripsi Deskripsi
                  = 97
Hasil deskripsi dari Pesan C adalah a
Press any key to continue . . .
```

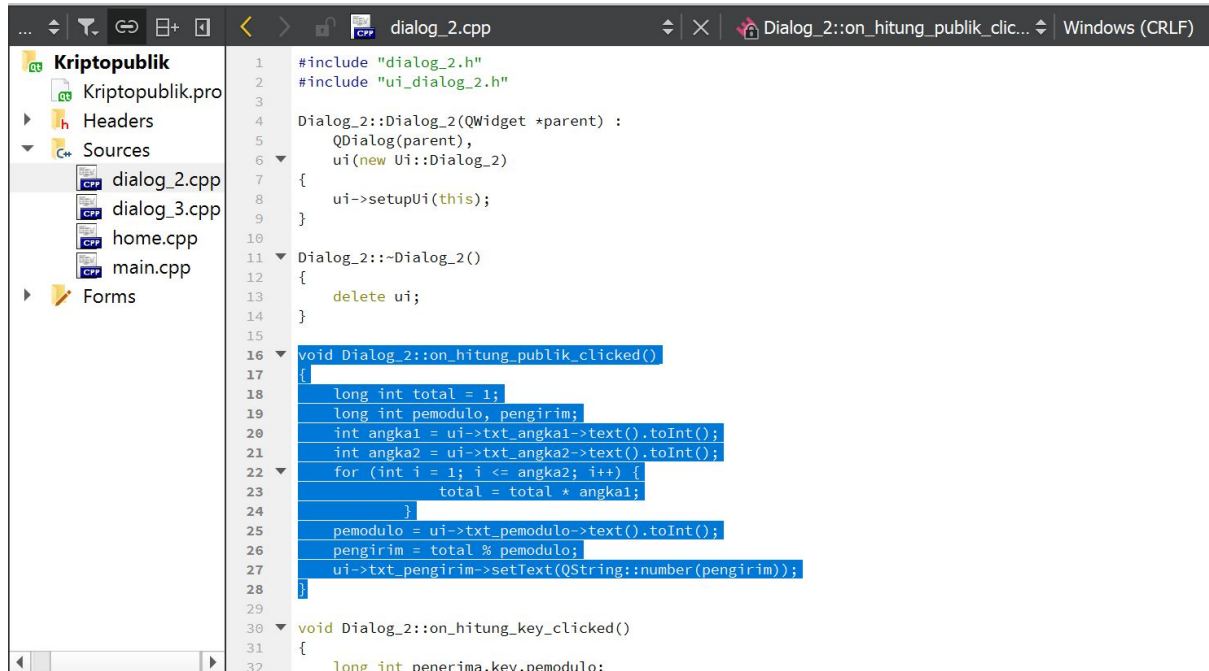
Versi GUI

Pada pembuatan gui ini, kami membuat 3 jendela, yaitu yang pertama jendela home, yang kedua jendela dialog_2, dan yang ke tiga adalah jendela dialog_3. yang dimana jendela home adalah tampilan awal saat program di eksekusi. untuk jendela dialog_2 berisi tentang mengenkripsi pesan(pengirim), dan untuk jendela dialog_3 berisi tentang dekripsi pesan (penerima).



```
home.cpp
1  #include "home.h"
2  #include "ui_home.h"
3
4  Home::Home(QWidget *parent)
5      : QMainWindow(parent)
6      , ui(new Ui::Home)
7  {
8      ui->setupUi(this);
9  }
10
11 ~Home()
12 {
13     delete ui;
14 }
15
16
17 void Home::on_btn_pengirim_clicked()
18 {
19     Dialog_2 * dialog_2 = new Dialog_2(this);
20     dialog_2->show();
21 }
22
23 void Home::on_btn_penerima_clicked()
24 {
25     //hide();
26     Dialog_3 * dialog_3 = new Dialog_3(this);
27     dialog_3->show();
28 }
29
```

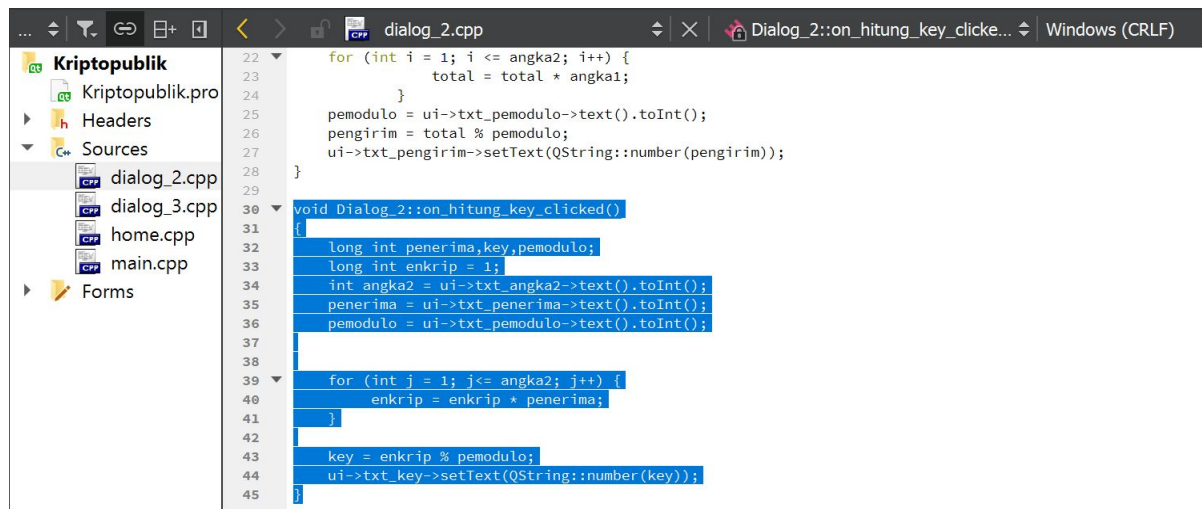
Perhitungan kunci publik



```
1  #include "dialog_2.h"
2  #include "ui_dialog_2.h"
3
4  Dialog_2::Dialog_2(QWidget *parent) :
5      QDialog(parent),
6      ui(new Ui::Dialog_2)
7  {
8      ui->setupUi(this);
9  }
10
11 Dialog_2::~Dialog_2()
12 {
13     delete ui;
14 }
15
16 void Dialog_2::on_hitung_publik_clicked()
17 {
18     long int total = 1;
19     long int pemodulo, pengirim;
20     int angka1 = ui->txt_angka1->text().toInt();
21     int angka2 = ui->txt_angka2->text().toInt();
22     for (int i = 1; i <= angka2; i++) {
23         total = total * angka1;
24     }
25     pemodulo = ui->txt_pemodulo->text().toInt();
26     pengirim = total % pemodulo;
27     ui->txt_pengirim->setText(QString::number(pengirim));
28 }
29
30 void Dialog_2::on_hitung_key_clicked()
31 {
32     long int penerima, key, pemodulo;
```

Pada program ini, kami menggunakan fungsi tanpa nilai balik (void). Source code diatas digunakan untuk melakukan perhitungan kunci publik pengirim. Di source code ini, kami menggunakan tipe data long int dan int. Sedangkan variabel yang digunakan adalah total (untuk menghitung perpangkatan), angka1 (untuk memasukkan nilai G), angka2 (untuk memasukkan kunci privat pengirim), pemodulo (untuk bilangan yang akan digunakan untuk melakukan modulo terhadap hasil perpangkatan), dan pengirim (untuk hasil akhir perhitungan kunci publik). Disini user dapat melakukan input pada kotak dialog txt_angka1 dan txt_angka2, serta dapat menampilkan hasil akhirnya pada kotak dialog txt_pengirim.

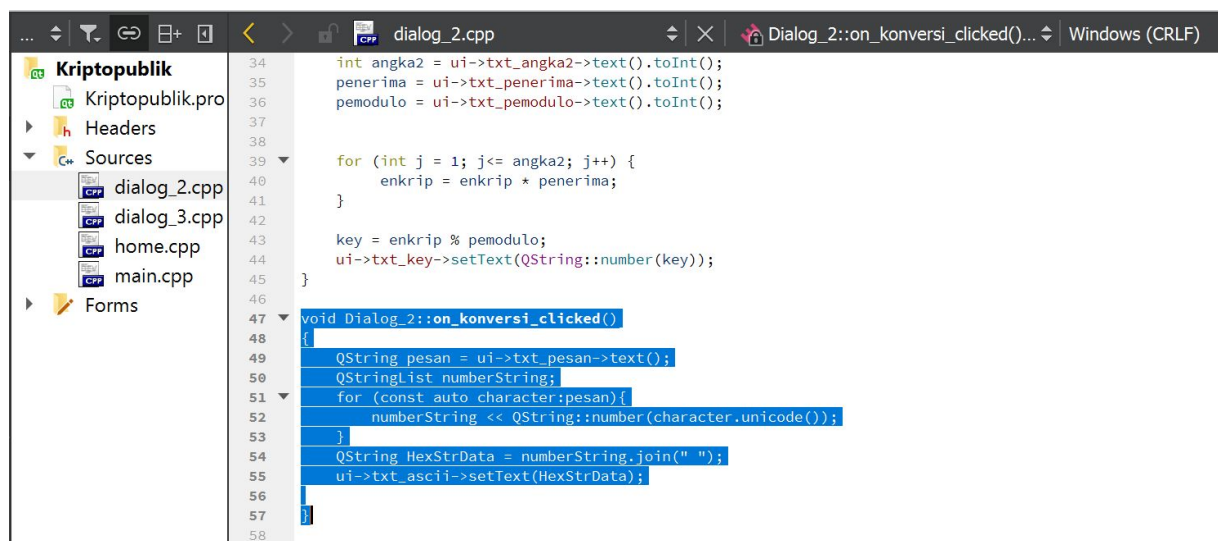
Perhitungan kunci enkripsi-deskripsi



```
22 for (int i = 1; i <= angka2; i++) {
23     total = total * angka1;
24 }
25 pemodulo = ui->txt_pemodulo->text().toInt();
26 pengirim = total % pemodulo;
27 ui->txt_pengirim->setText(QString::number(pengirim));
28 }
29
30 void Dialog_2::on_hitung_key_clicked()
31 {
32     long int penerima, key, pemodulo;
33     long int enkrip = 1;
34     int angka2 = ui->txt_angka2->text().toInt();
35     penerima = ui->txt_penerima->text().toInt();
36     pemodulo = ui->txt_pemodulo->text().toInt();
37
38     for (int j = 1; j <= angka2; j++) {
39         enkrip = enkrip * penerima;
40     }
41
42     key = enkrip % pemodulo;
43     ui->txt_key->setText(QString::number(key));
44 }
45
46
```

Pada program ini, kami menggunakan fungsi tanpa nilai balik (void). Source code diatas digunakan untuk melakukan perhitungan kunci enkripsi dan deskripsi. Pada fungsi ini, kami menggunakan tipe data int dan long int. Untuk menghitung kunci enkripsi dan deskripsi, dibutuhkan kunci privat pengirim, kunci publik penerima, dan bilangan pemodulo. Langkah pertama, pangkatkan kunci publik penerima dengan kunci privat pengirim. Selanjutnya, lakukan modulo pada hasil perpangkatan tersebut. Kemudian hasil perhitungan tadi akan ditampilkan ke dalam dialog txt_key.

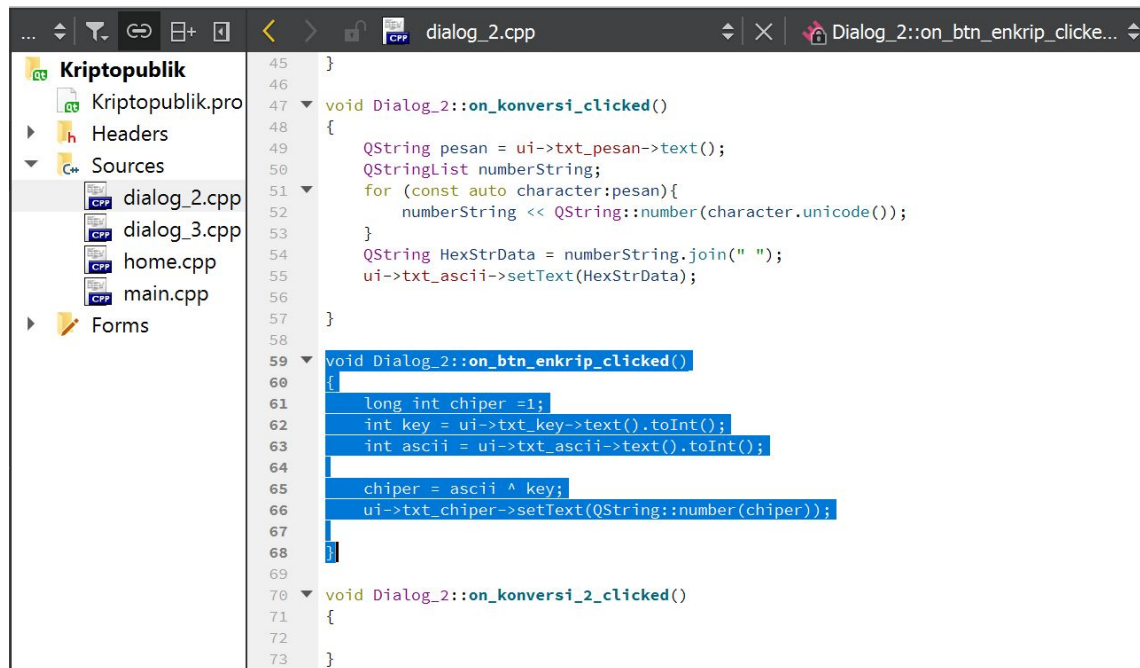
Konversi dari karakter ke ascii



```
34 int angka2 = ui->txt_angka2->text().toInt();
35 penerima = ui->txt_penerima->text().toInt();
36 pemodulo = ui->txt_pemodulo->text().toInt();
37
38 for (int j = 1; j <= angka2; j++) {
39     enkrip = enkrip * penerima;
40 }
41
42 key = enkrip % pemodulo;
43 ui->txt_key->setText(QString::number(key));
44 }
45
46 void Dialog_2::on_konversi_clicked()
47 {
48     QString pesan = ui->txt_pesan->text();
49     QStringList numberString;
50     for (const auto character:pesan){
51         numberString << QString::number(character.unicode());
52     }
53     QString HexStrData = numberString.join(" ");
54     ui->txt_ascii->setText(HexStrData);
55 }
56
57
58
```

Pada source code diatas menjelaskan tentang konversi pesan dari char ke ascii. Source code ini memiliki *qstringlist* jika di cli ini adalah arraynya, dengan variabel *numberString*.

Terdapat perulangan dengan pesan bertipe data character. Kemudian variabel *numberString* yang bernilai string diubah ke number. Selanjutnya mendeklarasikan jika QString dengan variabel *HexStrData* sama dengan hasil dari variabel *numberString* (konversi pesan dari char ke ascii) ditambahkan spasi.



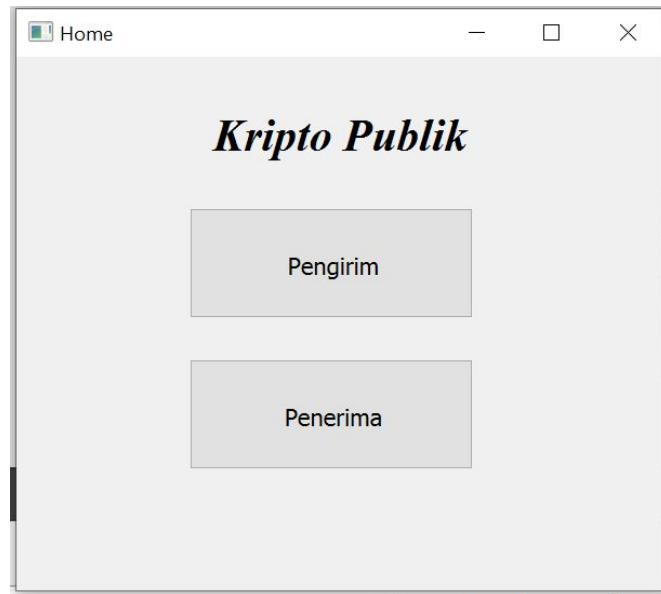
```
45 }
46
47 void Dialog_2::on_konversi_clicked()
48 {
49     QString pesan = ui->txt_pesan->text();
50     QStringList numberString;
51     for (const auto character:pesan){
52         numberString << QString::number(character.unicode());
53     }
54     QString HexStrData = numberString.join(" ");
55     ui->txt_ascii->setText(HexStrData);
56 }
57
58
59 void Dialog_2::on_btn_enkrip_clicked()
60 {
61     long int chiper =1;
62     int key = ui->txt_key->text().toInt();
63     int ascii = ui->txt_ascii->text().toInt();
64
65     chiper = ascii ^ key;
66     ui->txt_chiper->setText(QString::number(chiper));
67 }
68
69
70 void Dialog_2::on_konversi_2_clicked()
71 {
72
73 }
```

Di source code ini menghitung enkripsinya. pertama mendeklarasikan terlebih dahulu tipe data. Selanjutnya masukkan rumus enkripsinya yaitu pesan yang telah dikonversikan ke ascii di xor dengan kunci enkripsi, maka hasilnya akan keluar dalam bentuk number.

Output Program GUI :

Jendela 1 :

User dapat memilih untuk menggunakan proses enkripsi (pengirim) atau deskripsi (penerima)



Jendela 2 dan jendela 3 :

Pada jendela ini, user dapat melakukan perhitungan kunci publik, kunci enkripsi-deskripsi, serta melakukan proses enkripsi-deskripsi.

Kripto Publik (A)

Masukkan G: 12
Private: 5
Pemodulo: 37
Hitung

Kunci Publik A: 7
Kunci Publik B: 9
Hitung Key

Key: 34
Pesan: a
Konversi

ASCII: 97
Enkrip

Hasil: 67
Reset Pesan

Kripto Publik B

Masukkan G: 12
Private: 7
Pemodulo: 37
Hitung

Kunci Publik B: 9
Kunci Publik A: 7
Hitung key

Key: 34
Chiper: C
Konversi

ASCII: 67
Deskrip

Pesan: 97
Reset Pesan

Tugas Anggota Kelompok :

<i>Nama Anggota</i>	<i>NIM</i>	<i>Kontribusi pengerjaan</i>
<i>Abdul Malik</i>	<i>18.83.0276</i>	Membuat gui, membuat tampilan untuk aplikasi
<i>Sri Suraningsih</i>	<i>18.83.0290</i>	Membuat program switch case di cli
<i>Nur Dian Yustikarini</i>	<i>18.83.0298</i>	Membuat gui
<i>Diah Pingkan Sari</i>	<i>18.83.0326</i>	Membuat program perhitungan kunci di cli, membuat gui
<i>Mohamad Dzulfiquqar S.</i>	<i>18.83.0331</i>	Membuat laporan

Referensi:

<https://stackoverflow.com/questions/45772951/converting-qstring-to-ascii-value-vice-versa-in-qt>

<https://youtu.be/N3R9qmO4d-E>

<https://youtu.be/VigUMAfE2q4>

https://www.youtube.com/watch?v=Is8ma3K-_zo

Link Project:

<https://github.com/DiahPingkanSari/Pemrograman-C-Kriptopublik->