

## Source Program Lengkap

src.py

```
def gcd(a,b):
    while(b!=0):
        a,b = b,a%b
    return a

def lcm(a,b):
    return (a*b)//gcd(a,b)

def prime(a):
    for i in range(2,a):
        if (a%i ==0):
            return False
    return True

def inverse_mod(base,m):
    for i in range(1,base):
        if ((i*m)%base)==1:
            return i

def makeblocks(message,length):
    messagenum = ""
    separator = 0
    for i in range(len(message)):
        if (i > 0 and len(messagenum)%length==separator):
            messagenum += ","
            separator += 1
        if (ord(message[i])%97<10):
            messagenum += "0"
        messagenum += str(ord(message[i])%97)
        if (separator==length):
            separator = 0
    blocks = messagenum.split(",")
    return blocks

def blockstotext(blocks):
    decryptednum, decrypted = "", ""
    for numbers in blocks:
        decryptednum += str(numbers)
    current = ""
    for i in range(len(decryptednum)):
        current += str(decryptednum[i])
        if (i % 2 != 0):
            decrypted += chr(int(current)+97)
            current = ""
    return decrypted

def ersal(p,q):
    if(not prime(p)):
        print("p not prime")
        return (0,0)
    if(not prime(q)):
        print("q not prime")
        return (0,0)
    n = p*q
    toitent = (p-1)*(q-1)
    return (n,toitent)
```

```

def ersa2(n,e,message):
    if (gcd(n,e)!=1):
        print("e is not coprime with n")
        return []
    valid = False
    length = len(str(n))
    while (not valid):
        currentlength = length
        blocks = makeblocks(message,length)
        for i in range(len(blocks)):
            if (int(blocks[i])>=n-1):
                length -= 1
        if (length==currentlength):
            valid = True
    for i in range(len(blocks)):
        blocks[i] = (int(blocks[i])**e) % n
    encrypted = ""
    for i in range(len(blocks)):
        if (i==len(blocks)-1):
            encrypted += str(blocks[i])
            break
        encrypted += str(blocks[i]) + " "
    return encrypted

def elgamalkey(p,g,x):
    if (g<p and 1<x<=p-2):
        return (g**x) % p
    else:
        print("value not valid")

def eelgamal(y,p,g,k,message):
    if (1<=k<=p-2):
        valid = False
        length = len(str(p))
        while (not valid):
            currentlength = length
            blocks = makeblocks(message,length)
            for i in range(len(blocks)):
                if (int(blocks[i])>=p-1):
                    length -= 1
            if (length==currentlength):
                valid = True
        enc1 = []
        enc2 = []
        for i in range(len(blocks)):
            enc1.append((g**k) % p)
            enc2.append(((y**k) * int(blocks[i])) % p))
        return(enc1,enc2)
    else:
        print("k tidak valid")
        return([],[])

def paillierkey(p,q,g):
    if(not prime(p)):
        print("p not prime")
        return (0,0)
    if(not prime(q)):
        print("q not prime")
        return (0,0)
    if(gcd(p,q)!=1):
        print("not co prime")

```

```

        return (0,0)

    n = p*q
    yss = lcm(p-1,q-1)
    myu = inverse_mod(n, ((g**yss)%(n**2))-1)/n
    return n,yss,myu

def epaillier(p,g,n,r,message):
    if(r<0 or r>n or gcd(r,n)!=1):
        print("r not valid")
        return (0,0)
    blocks = makeblocks(message,2)
    enc = []
    for i in range(len(blocks)):
        enc.append(((g**int(blocks[i]))*(r**n))%(n**2))
    return enc

def dpaillier(p,n,yss,myu,enc):
    blocks = []
    for i in range(len(enc)):
        plainnumber = (((((enc[i]**yss)%(n**2))-1)/n)*myu)%n
        blockselem = str(int(plainnumber))
        if (i==len(enc)-1):
            if (len(blockselem)<2):
                blockselem = "0" + blockselem
        while (len(blockselem)<2):
            blockselem = "0" + blockselem

        blocks.append(blockselem)
    return blockstotext(blocks)

def delgamal(x,p,enc1,enc2):
    blocks = []
    for i in range(len(enc1)):
        plainnumber = (enc2[i]*(enc1[i]**(p-1-x)) % p) % p
        blockselem = str(plainnumber)
        valid = False
        if (i==len(enc1)-1):
            if (len(blockselem) % 2 != 0):
                blockselem = "0" + blockselem
            valid = True
        while (not valid):
            if(len(blockselem)<len(str(p))):
                blockselem = "0" + blockselem
            else:
                valid = True
        blocks.append(blockselem)
    return blockstotext(blocks)

def drsa(n,toitent,e,encrypted):
    d = inverse_mod(toitent,e)
    blocks = encrypted.split(" ")
    for i in range(len(blocks)):
        blocks[i] = (int(blocks[i])**d) % n
        if (len(str(blocks[i]))==len(str(n))-1):
            blocks[i] = "0" + str(blocks[i])
    return blockstotext(blocks)

```

```

from flask import Flask, render_template, request
import src as algo

app = Flask(__name__)

@app.route('/', methods=["GET", "POST"])
def home():
    return render_template('index.html')

@app.route('/encrypt', methods=["GET", "POST"])
def encrypt():
    if (request.method == "POST"):
        cypher = request.form['methodInput']
        message = request.form['messageinput'].lower().replace(" ", "")
        if (cypher=="RSA"):
            p = int(request.form['pInput'])
            q = int(request.form['qInput'])
            e = int(request.form['eInput'])
            (n,toitent) = algo.ersal(p,q)
            encrypt = algo.ersa2(n,e,message)
            print(encrypt)
            return render_template("index.html", answer = encrypt, mode = "encrypted")
        elif (cypher=="ElGamal"):
            p = int(request.form['pInput'])
            g = int(request.form['gInput'])
            x = int(request.form['xInput'])
            k = int(request.form['kInput'])
            y = algo.elgamalkey(p,g,x)
            enc1, enc2 = algo.eelgamal(y,p,g,k,message)
            enc1.append(enc2)
            return render_template("index.html", answer = enc1, mode = "encrypted")
        elif (cypher=="Paillier"):
            p = int(request.form['pInput'])
            q = int(request.form['qInput'])
            g = int(request.form['gInput'])
            r = int(request.form['rInput'])
            n, yss, myu = algo.paillierkey(p,q,g)
            enc = algo.epaillier(p,g,n,r,message)
            return render_template("index.html", answer = enc, mode = "encrypted")
        else:
            return render_template("index.html")

@app.route('/decrypt', methods=["GET", "POST"])
def decrypt():
    if (request.method == "POST"):
        cypher = request.form['methodInput']
        encrypted = request.form['cypher1Input']
        if (cypher=="RSA"):
            print("rsa")
            p = int(request.form['pInput'])
            q = int(request.form['qInput'])
            (n,toitent) = algo.ersal(p,q)
            e = int(request.form['eInput'])
            decrypt = algo.drsa(n,toitent,e,encrypted)
            return render_template("index.html", answer1 = decrypt, mode= "decrypted")
        elif (cypher=="ElGamal"):
            p = int(request.form['pInput'])
            x = int(request.form['xInput'])
            encrypted2 = request.form['cypher2Input']
            enc1 = encrypted.split(", ")
            enc2 = encrypted2.split(", ")

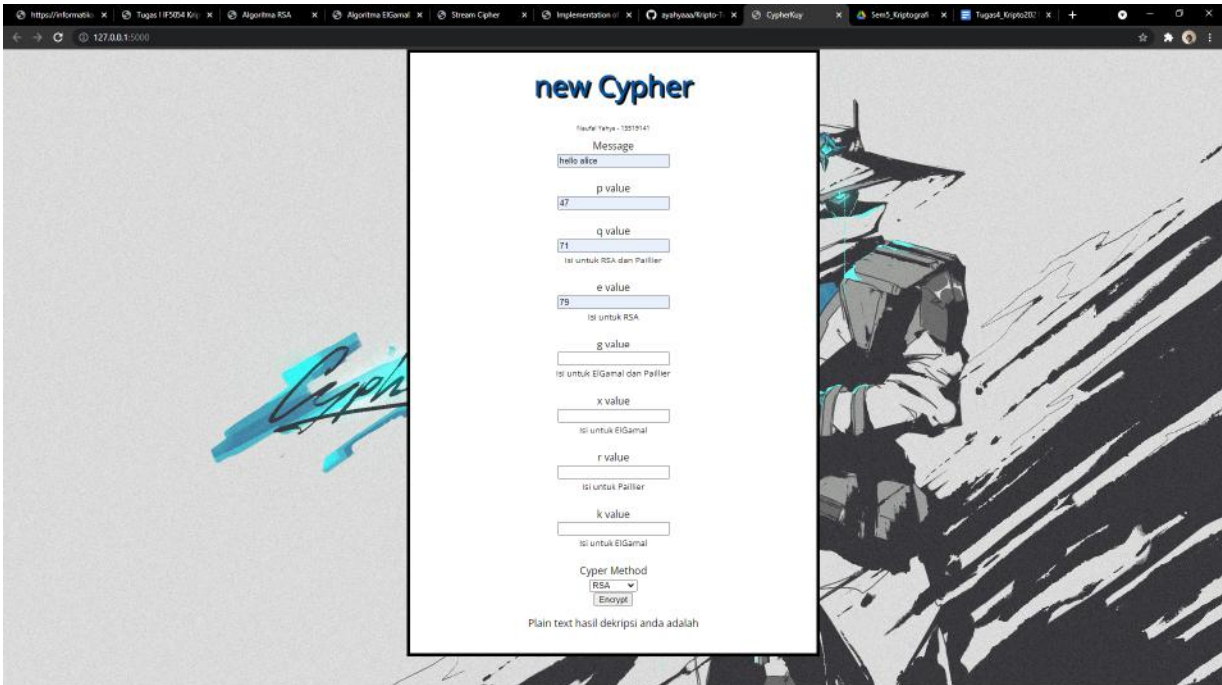
```

```
for i in range(len(enc1)):
    enc1[i] = int(enc1[i])
for i in range(len(enc2)):
    enc2[i] = int(enc2[i])
decrypted = algo.delgamal(x,p,enc1,enc2)
return render_template("index.html", answer1 = decrypted, mode= "decrypted")
elif (cypher=="Paillier"):
    p = int(request.form['pInput'])
    q = int(request.form['qInput'])
    g = int(request.form['gInput'])
    n, yss, myu = algo.paillierkey(p,q,g)
    encrypt = encrypted.split(", ")
    for i in range(len(encrypt)):
        encrypt[i] = int(encrypt[i])
    decrypted = algo.dpaillier(p,n,yss,myu,encrypt)
    return render_template("index.html", answer1 = decrypted, mode= "decrypted")
else:
    return render_template("index.html")

if __name__ == '__main__':
    app.run(debug=True)
```

## Tampilan Antarmuka dan Contoh Penggunaan

### Algoritma RSA

The screenshot shows a web browser window with the title 'new Cypher'. The URL bar shows 'https://informasi...' and the address '127.0.0.1:5000'. The page has a dark background with a stylized illustration of a person in a futuristic suit on the right. The main content area is a white box with the title 'new Cypher' and a subtitle 'Naufal Yahya - 13519141'. It contains several input fields: 'Message' (with 'hello alice' entered), 'p value' (with '47' entered), 'q value' (with '71' entered), 'e value' (with '79' entered), 'g value', 'x value', 'r value', and 'k value'. Below these are instructions: 'isi untuk RSA dan Paillier', 'isi untuk RSA', 'isi untuk ElGamal dan Paillier', 'isi untuk ElGamal', and 'isi untuk Paillier'. At the bottom, there is a 'Cypher Method' dropdown menu with 'RSA' selected, and an 'Encrypt' button. Below the button, it says 'Plain text hasil dekripsi anda adalah'.

Sebelum Enkripsi

new Cypher

Naufal Yahya - 13519141

cypher text anda adalah 328 301 2653 2986 1164

Cypher text

Cypher text

isi untuk ElGamal dengan array yang berada di dalam array

p value

q value

isi untuk RSA dan Paillier

e value

isi untuk RSA

g value

isi untuk ElGamal dan Paillier

x value

isi untuk ElGamal

Cypher Method

RSA

isi sama dengan Metode Cypher Enkripsi

Decrypt

Hasil Enkripsi

new Cypher

Naufal Yahya - 13519141

cypher text anda adalah 328 301 2653 2986 1164

Cypher text

328 301 2653 2986 1164

Cypher text

isi untuk ElGamal dengan array yang berada di dalam array

p value

47

q value

71

isi untuk RSA dan Paillier

e value

79

isi untuk RSA

g value

isi untuk ElGamal dan Paillier

x value

isi untuk ElGamal

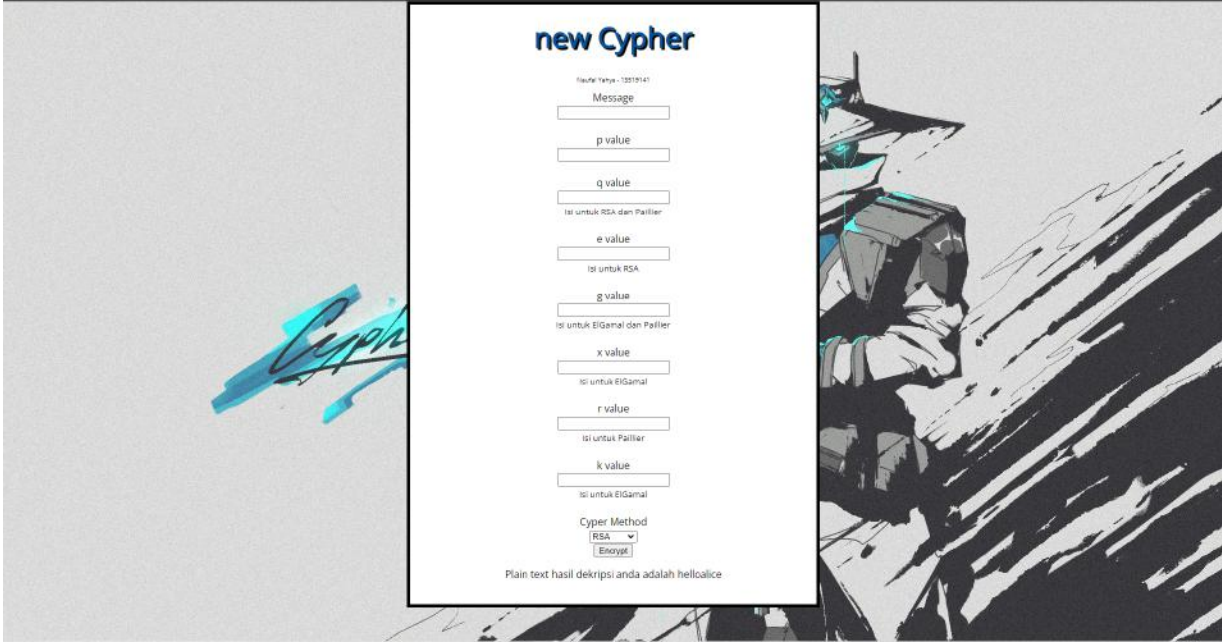
Cypher Method

RSA

isi sama dengan Metode Cypher Enkripsi

Decrypt

Sebelum Dekripsi



new Cypher

Naufal Yahya - 13519141

Message

p value

q value

Isi untuk RSA dan Paillier

e value

Isi untuk RSA

g value

Isi untuk ElGamal dan Paillier

x value

Isi untuk ElGamal

r value

Isi untuk Paillier

k value

Isi untuk ElGamal

Cypher Method

RSA

ElGamal

Plain text hasil dekripsi anda adalah helloalice

Hasil Dekripsi

Pesan: Hello Alice

CypherTeks: 328 301 2653 2986 1164

p: 47

q: 71

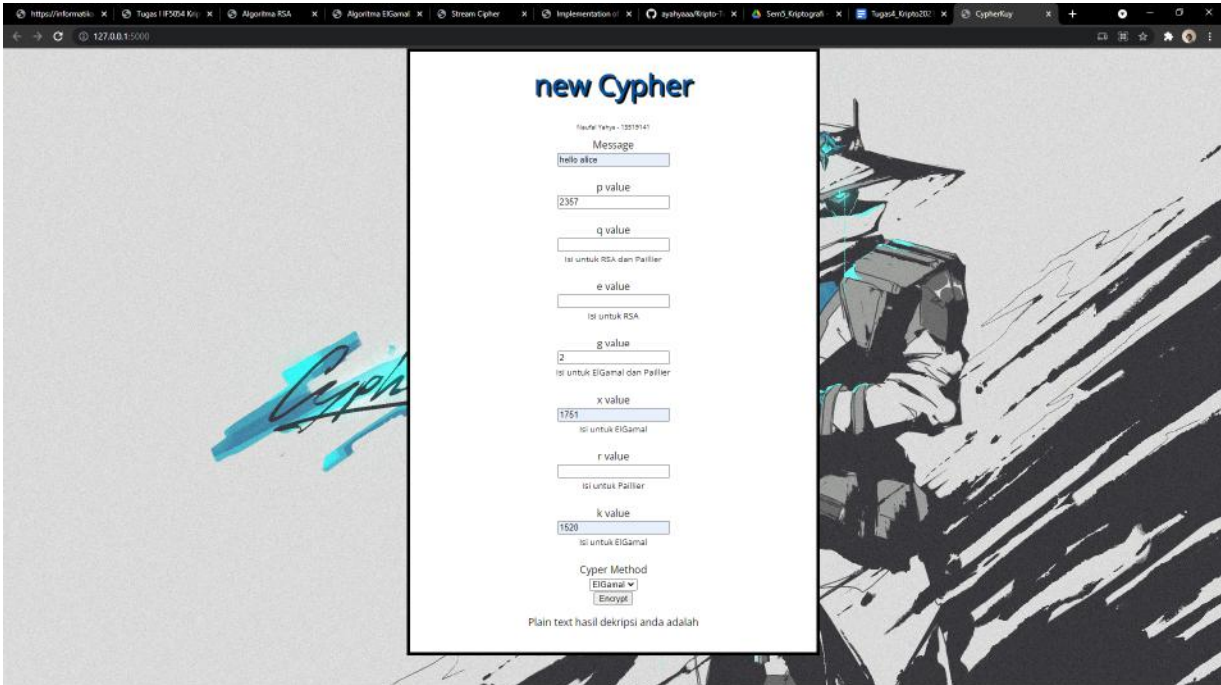
e: 79



Naufal Yahya Kurnianto 13519141

Tugas 4 IF4020 Kriptografi

Algoritma ElGamal

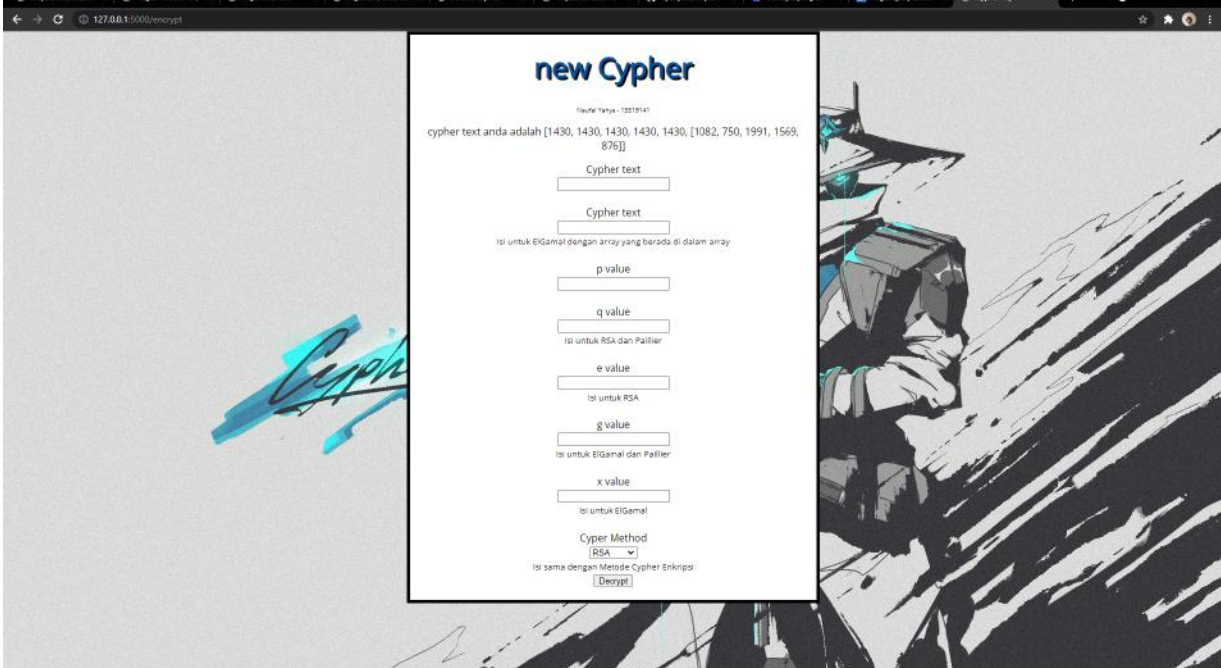


The screenshot shows a web browser with multiple tabs. The active tab is 'CypherKey'. The page title is 'new Cypher'. The user is 'Naufal Yahya - 13519141'. The message to be encrypted is 'hello alic'. The parameters are set as follows:

- p value: 2357
- q value: (empty)
- e value: (empty)
- g value: 2
- x value: 1751
- r value: (empty)
- k value: 1520

The 'Cypher Method' is set to 'ElGamal'. The 'Encrypt' button is visible. Below the form, it says 'Plain text hasil dekripsi anda adalah'.

Sebelum Enkripsi



The screenshot shows the same web browser. The page title is 'new Cypher'. The user is 'Naufal Yahya - 13519141'. The cypher text is displayed as an array: [1430, 1430, 1430, 1430, 1430, 1082, 750, 1991, 1569, 876]. The 'Cypher Method' is set to 'RSA'. The 'Decrypt' button is visible. Below the form, it says 'Isi sama dengan Metode Cypher Enkripsi'.

Hasil Enkripsi



new Cypher

Naufal Yahya - 13519141

cypher text anda adalah [1430, 1430, 1430, 1430, 1430, [1082, 750, 1991, 1569, 876]]

Cypher text  
[1430, 1430, 1430, 1430, 1430]

Cypher text  
[1082, 750, 1991, 1569, 876]

isi untuk ElGamal dengan array yang berada di dalam array

p value  
[2357]

q value  
[ ]  
isi untuk RSA dan Paillier

e value  
[ ]  
isi untuk RSA

g value  
[2]  
isi untuk ElGamal dan Paillier

x value  
[1751]  
isi untuk ElGamal

Cypher Method  
[ElGamal]  
isi sama dengan Metode Cypher Enkripsi

[Decrypt]

Sebelum Dekripsi

new Cypher

Naufal Yahya - 13519141

Message  
[ ]

p value  
[ ]

q value  
[ ]  
isi untuk RSA dan Paillier

e value  
[ ]  
isi untuk RSA

g value  
[ ]  
isi untuk ElGamal dan Paillier

x value  
[ ]  
isi untuk ElGamal

r value  
[ ]  
isi untuk Paillier

k value  
[ ]  
isi untuk ElGamal

Cypher Method  
[RSA]  
[Encrypt]

Plain text hasil dekripsi anda adalah helloalice

Hasil Dekripsi

Pesan: Hello Alice

CypherTeks1: [1430, 1430, 1430, 1430, 1430]

CypherTeks2: [1082, 750, 1991, 1569, 876]

Naufal Yahya Kurnianto 13519141

Tugas 4 IF4020 Kriptografi

p: 2357

g: 2

x: 1751

y =  $g^x = 1185$

k: 1520

Naufal Yahya Kurnianto 13519141

Tugas 4 IF4020 Kriptografi

## Algoritma Paillier

**new Cypher**

Naufal Yahya - 13519141

Message

p value

q value

Isi untuk RSA dan Paillier

e value

Isi untuk RSA

g value

Isi untuk ElGamal dan Paillier

x value

Isi untuk ElGamal

r value

Isi untuk Paillier

k value

Isi untuk ElGamal

Cypher Method

Plain text hasil dekripsi anda adalah

Sebelum Enkripsi

**new Cypher**

Naufal Yahya - 13519141

cypher text anda adalah [2028, 5791, 1945, 1945, 3382, 606, 1945, 1499, 2556, 5791]

Cypher text

Cypher text

Isi untuk ElGamal dengan array yang berada di dalam array

p value

q value

Isi untuk RSA dan Paillier

e value

Isi untuk RSA

g value

Isi untuk ElGamal dan Paillier

x value

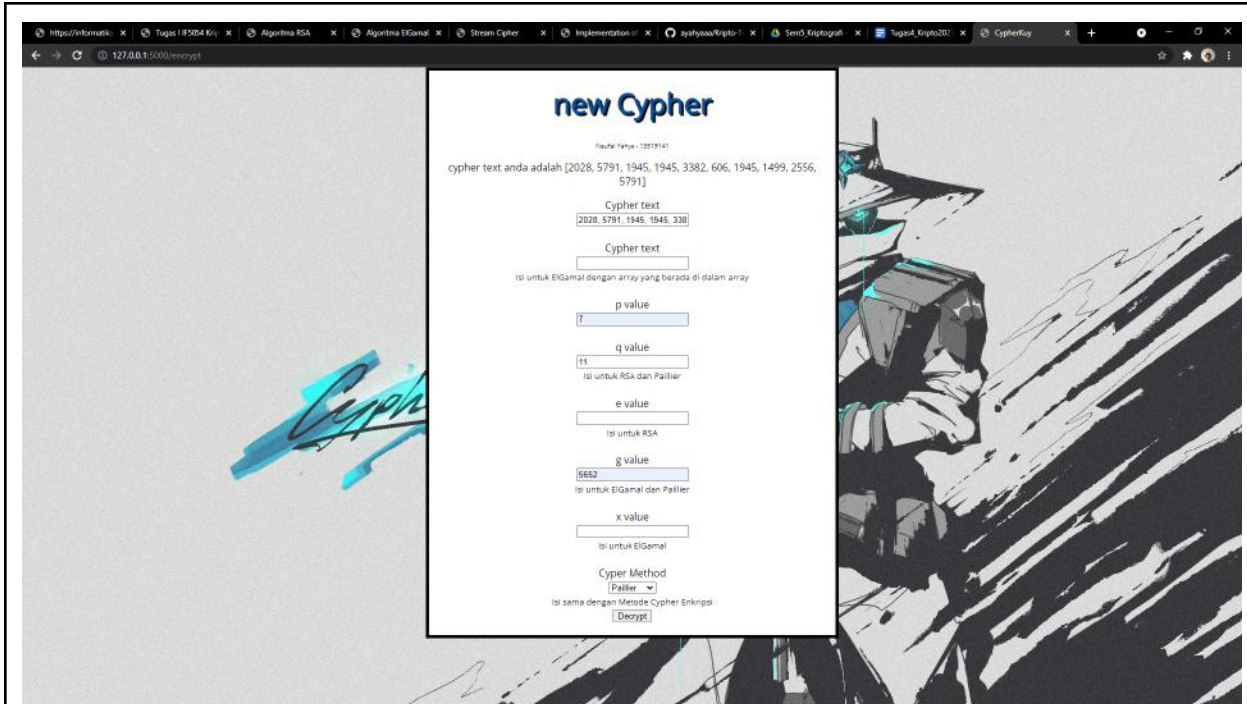
Isi untuk ElGamal

Cypher Method

Isi sama dengan Metode Cypher Enkripsi

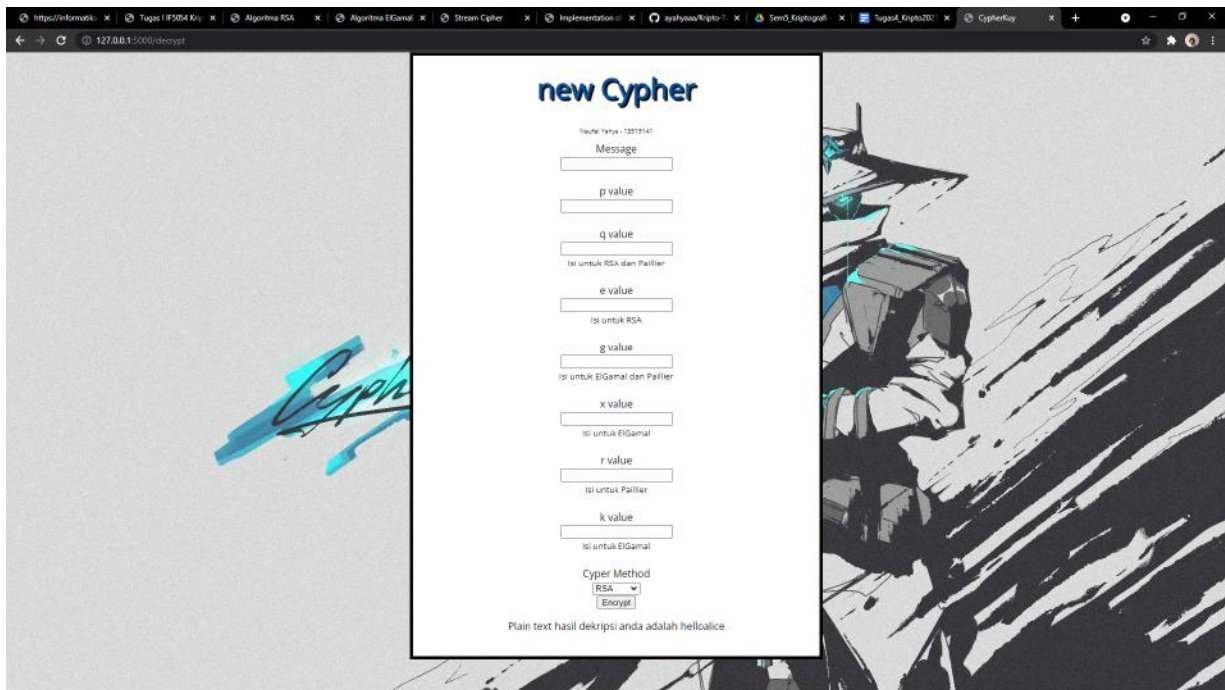
Hasil Enkripsi





The screenshot shows a web browser window with the URL `https://informasi.ky.../127.0.0.1:5000/encrypt`. The page title is "new Cypher". The user is identified as "Naufal Yahya - 13519141". The cypher text is displayed as `[2028, 5791, 1945, 1945, 3382, 606, 1945, 1499, 2556, 5791]`. Below this, there are input fields for "Cypher text" (containing the same array), "p value" (7), "q value" (11), "e value" (empty), "g value" (5652), and "x value" (empty). The "Cypher Method" is set to "Paillier". The "Decrypt" button is visible.

Sebelum Dekripsi



The screenshot shows the same web browser window, but the URL is `https://informasi.ky.../127.0.0.1:5000/decrypt`. The page title is "new Cypher". The user is identified as "Naufal Yahya - 13519141". The "Message" field is empty. The "p value" is 7, "q value" is 11, "e value" is empty, "g value" is empty, "x value" is empty, "r value" is empty, and "k value" is empty. The "Cypher Method" is set to "RSA". The "Encrypt" button is visible. The "Decrypt" button is also visible.

Hasil Dekripsi

Pesan: Hello Alice

CypherTeks: [2028, 5791, 1945, 1945, 3382, 606, 1945, 1499, 2556, 5791]

p: 7

q: 11

Naufal Yahya Kurnianto 13519141

Tugas 4 IF4020 Kriptografi

```
n = p*q = 77  
g: 5652  
r: 23  
(lambda, myu) = (30, 74)
```

Repo dapat diakses di <https://github.com/ayahyaaa/Kripto-Tucil4>