1. Diffie-Hellman : Tentukan Nilai K, jika nilai n = (2 angka Terakhir NIM Anda, klu bukan bilangan prima tambahkan dengan nilai berapa pun agar jadi bilangan prima), nilai g = 5, ingat g < n.

   Jawab :

   Rumus : $K = g^{ab} \% n = K = g^{16} \% 61$

   Diketahui
   - **n** adalah bilangan prima publik (diberikan sebagai 61).
   - **g** adalah basis (diberikan sebagai 5, dan sudah memenuhi syarat g<ng < ng<n).
   - **1** adalah bilangan acak rahasia.
   - **6** adalah bilangan acak rahasia.

   Langkah Penyelesaian

   1.
   2. Hitung pangkat 5^6
      5^6 = 15625
   3. Hitung 15625 % 61 : Lakukan pembagian 15625 ÷ 61, dan ambil sisanya

      15625 ÷ 61 = 256 (Sisa : 9) = > 15625 % 61 = 9

      Jadi, Nilai K = 9

```python
n = 61
g = 5
a = 1
b = 6

K = (g ** (a * b)) % n
print("Nilai K =", K)

```

```
[Running] python -u "e:\CODINGAN\Tugas Kripto\Quis\Diffie-Hellman.py"
Nilai K = 9

[Done] exited with code=0 in 0.15 seconds
```

2. ElGamal : Enkripsi 4 angka NIM anda sebagai Plainteksnya. Tentukan :

   a. Pasangan Kunci Publik dan Kunci Private nya dengan p = 2 angka terahir NIM anda, jika bukan bilangan prima, maka tambahkan dengan angka berapapun agar menjadi bilangan prima, g = 3, dan x = 2 angka pertama NIM anda, jika bukan bilangan prima tambakan dengan angka berapapun agar menjadi bilangan prima. Ingat g < p.
   b. Hasil Ekripsi
   c. Hasil Dekripsi

   Jawab :

   1. Pasangan Kunci Publik dan Kunci Privat
      Diketahui :
      p = 61 (Bilangan Prima),
      g = 3 (Primitive Root),
      x = 23 (Kunci Private).

      Langkah pertama adalah menghitung kunci publik y dengan rumus

      $$y = g^x \% p$$

      Mari kita hitung

      $$y = g^{23} \% 61$$

      Langkah-langkah perhitungannya adalah sebagai berikut :

      - $3^1 \% 61 = 3$
      - $3^2 \% 61 = 9$
      - $3^4 \% 61 = (3^2)^2 \% 61 = 9^2 \% 61 = 81 \% 61 = 20$
      - $3^8 \% 61 = (3^4)^2 \% 61 = 20^2 \% 61 = 400 \% 61 = 34$
      - $3^{16} \% 61 = (3^8)^2 \% 61 = 34^2 \% 61 = 1156 \% 61 = 51$
      - $3^{23} \% 61 = 3^{16} \, x \, 3^4 \, x \, 3^2 \, x \, 3^1 \, x \% 61$

      Menghitung

      $$3^{16} \% 61 = 51, 3^4 \% 61 = 20, 3^2 \% 61 = 9, 3^1 \% 61 = 3$$

      Hitung Bertahap

      $$3^{23} \% 61 = (51 \, . \, 20 \% 61) \, . \, 9 \% 61 \, . \, 3 \% 61$$

      51 . 20 % 61 = 1020 % 61 = 4

      44. 9 % 61 = 396 % 61 = 30

      30 . 3 % 61 = 90 % 61 = 27

      y = 27

      Kunci Publik (P = 61 | g = 3 | y = 27

      Kunci Private 23

```python
p = 61
g = 3
x = 23

y = pow(g, x, p)

print(f"Kunci Publik: (p = {p}, g = {g}, y = {y})")
print(f"Kunci Privat: x = {x}")

```

```
[Running] python -u "e:\CODINGAN\Tugas Kripto\Quis\ElGamal Encryption ( Pasangan Kunci Publik dan Kunci Privat ).py"
Kunci Publik: (p = 61, g = 3, y = 27)
Kunci Privat: x = 23

[Done] exited with code=0 in 0.109 seconds
```

2. Enkripsi
   Diketahui
   - Modulus (p) = 61
   - Generator (g) = 3
   - Private Key (x) = 23
   - Publik Key (y = g^x % p)

     $$y = 3^{23} \% 61 = 27$$
   - Plaintext (M) : 2061
   - Random Integer (k) : 15

     Ditanyakan C1 & C2

     Langka Pertama c2

     $$c_1 = g^{15} \% 61$$

     Gunakan pemangkatan modular dengan metode pemangkatan biner :
     - Representasi biner dari 151515 : 1111_2.
     - Langkah pemangkatan modular :
       $3^1 = \% 61 = 3.$
       $3^2 = \% 61 = 9.$
       $3^4 = \% 61 = (9^2) \% 61 = 20$
       $3^8 = \% 61 = (20^2) \% 61 = 34$

     Gabungkan

     $$3^{15} = 3^4 \, x \, 3^4 \, x \, 3^2 \, x \, 3 \% 61 = (34 \, x \, 20 \, x \, 9 \, x \, 3) \% 61$$

     Langkah Perhitungan

     37 x 25 = 925 % 61 = 10,

     10 x 16 = 160 % 61 = 28,

     38 x 27 = 1026 % 61 = 22.

Hasilnya = $y^k \% p = 22$

Langkah Kedua => Hitung c2

Subsitusi :

$$c_2 = \left(M.y^k\right) \% p = (2061.22) \% 61$$

- Hitung 2061 % 61
  $2061 \div 61 = 33$ (Sisa 48)
  Jadi, 2061 % 61 = 48

- Hitung
  $c_2 = \left(M.y^k\right) \% p = (2061.22) \% 61$
  1056 ÷ 61 = 17 (Sisa 13)

- Hasil
  $c_2 = 13$

Ciphertext

$(C_1, C_2) = (60, 30)$

```python
def mod_exp(base, exp, mod):
    result = 1
    base = base % mod
    while exp > 0:
        if exp % 2 == 1:
            result = (result * base) % mod
        base = (base * base) % mod
        exp = exp // 2
    return result

p = 61
g = 3
x = 23
y = mod_exp(g, x, p)
M = 2061
k = 15

c1 = mod_exp(g, k, p)
yk = mod_exp(y, k, p)
c2 = (M * yk) % p

print(f"Plaintext (M) = {M}")
print(f"Kunci Publik (p = {p}, g = {g}, y = {y})")
print(f"Bilangan Acak (k) = {k}")
print(f"Ciphertext (c1, c2) = ({c1}, {c2})")

```

```
[Running] python -u "e:\CODINGAN\Tugas Kripto\Quis\tempCodeRunnerFile.py"
Plaintext (M) = 2061
Kunci Publik (p = 61, g = 3, y = 27)
Bilangan Acak (k) = 15
Ciphertext (c1, c2) = (60, 13)

[Done] exited with code=0 in 0.117 seconds
```

3. Deskripsi
   Diketahui

   - Modulus (p) = 61
   - Generator (g) = 3
   - Private Key (x) = 23
   - Plaintext (M) : 2061
   - Random Integer (k) : 15
   - Ciphertext (60,13)

   Ditanyakan Mendekripsi ciphertext untuk mendapatkan plaintext M

   Langkah 1: Hitung $M_{\%} = \left(C_2 \,.\, C_1^{p-1-x}\right) \% \, p$

   Rumus Deskripsi $M_{\%} = \left(C_2 \,.\, C_1^{p-1-x} \% \, p \right)$

   Diketahui

   $C_1 = 60, C_2 = 13, x = 23$

   p – 1 – x = 61 – 1 – 23 = 27

   Hitung $C_1^{p-1-x} \% \, p$ dengan Pemangkatan Modular

   - Representasi biner dari 37 : 100101_2
   - Pemangkatan modular

     $60^1 \% \, 61 = 60$
     $60^2 \% \, 61 = (60.60) \% \, 61 = 3600 \% \, 61 = 1$
     $60^4 \% \, 61 = (1.1) \% \, 61 = 1$
     $60^8 \% \, 61 = (1.1) \% \, 61 = 1$
     $60^{16} \% \, 61 = (1.1) \% \, 61 = 1$
     $60^{32} \% \, 61 = (1.1) \% \, 61 = 1$

     Gabungkan
     $60^{37} \% \, 61 = 60 \,.\, 1 \,.\, 1 = 60$

     Substitusi ke Rumus $M_{\%}$:

     $M_{\%} = \left(C_2 \,.\, C_1^{p-1-x} \right) \% \, p$
     $M_{\%} = (13 \,.\, 60) \% \, 61$
     $M_{\%} = 780 \% \, 61 = 48$

     Hasil Sementara 48

Langkah Kedua Restorasi Plaintext Asli :
Plaintext asli M direstorasi menggunakan :

M = $M_{\%} + kp$

Dengan K adalah bilangan bulat sehingga M sesuai konteks masalah
Diketahui M = 2061

Hitung :

M = 48 + 33 . 61

M = 48 + 2013 = 2061

```python
def modular_exponentiation(base, exponent, modulus):
    result = 1
    base = base % modulus
    while exponent > 0:
        if exponent % 2 == 1:
            result = (result * base) % modulus
        base = (base * base) % modulus
        exponent //= 2
    return result
p = 61
g = 3
x = 23
C1 = 60
C2 = 13
p_minus_1_minus_x = p - 1 - x
C1_exp = modular_exponentiation(C1, p_minus_1_minus_x, p)
M_mod = (C2 * C1_exp) % p
k = 33
M = M_mod + k * p
print("Hasil dekripsi:")
print(f"M_mod: {M_mod}")
print(f"Plaintext asli (M): {M}")

```

```
[Running] python -u "e:\CODINGAN\Tugas Kripto\Quis\ElGamal Encryption Hasil Deskripsi.py"
Hasil dekripsi:
M_mod: 48
Plaintext asli (M): 2061

[Done] exited with code=0 in 0.114 seconds
```

3. ElGamal : Enkripsi 4 angka NIM anda sebagai Plainteksnya. Tentukan :

- Tentukan Pasangan Kunci RSA

Langkah 1 : Hitung Modulus N = >

n = p x q

n = 7 x 17 = 119

Langka 2 : Hitung Totient ø (n) :

Ø (n) = (p – 1) x (q – 1)

Ø (n) = (7 -1) x (17 – 1) = 6 x 16 = 96

Langkah 3 : Pilih Eksponen Publik e

Pilih e sehinga 1 < e < Ø (n) dan GCD (e, Ø (n)) = 1. Misalkan e = 5 (karena GCD (5,96) = 1)

Langkah 4 : Hitung Eksponen Privat d

d adalah kebalikan modular dari e % Ø (n) , yaitu :

d x e ≡ 1 (% Ø (n))

dengan e = 5 dan Ø (n) = 96, kita cari d:

d = 77 (Karena 5 > 77 = 385 dan 385 % 96 = 1

Pasangan Kunci :

- Kunci Publik (e = 5, n = 119)
- Kunci Private (d = 77, n = 119)

```python
from math import gcd


def modular_inverse(e, phi):
    for d in range(1, phi):
        if (e * d) % phi == 1:
            return d
    return None
p = 7
q = 17
n = p * q
phi = (p - 1) * (q - 1)
e = 5
if gcd(e, phi) != 1:
    raise ValueError("e dan ϕ(n) harus coprime!")
d = modular_inverse(e, phi)
if d is None:
    raise ValueError("Tidak ditemukan invers modular untuk e dan ϕ(n)!")
print("Pasangan Kunci RSA:")
print(f"Kunci Publik: (e={e}, n={n})")
print(f"Kunci Privat: (d={d}, n={n})")

```

- Enskirpsi RSA
  - Diketahui
  - p = 7, q = 17
  - n = p x q = 7 x 17 = 119
  - Ø (n) = (p – 1) x (q – 1) = 6 x 16 = 96
  - Eksponen publik e = 5
  - Plaintext : “ Sitti “
  - Representasi ASCII :

    S = 83 | i = 105 | t = 116 | t = 116 | i = 105

Langkah 1 = > Ubah Plaintext ke Representasi Numerik ASCII

[S,i,t,t,i]=[83,105,116,116,105]

Langkah 2 = > Enkripsi dengan Rumus RSA

C = $M^c \% n$

e = 5, n = 119

Lakukan perhitungan modular untuk setiap M

Untuk M = 83 :

$C = 83^5 \% 119$

Hitung 83^5 menggunakan pengurangan modular setiap langkah

- $83^2 \% 119 = (83 \; x \; 83) \% 119 = 6889 \% 119 = 102$
- $83^4 \% 119 = (102 \; x \; 102) \% 119 = 10404 \% 119 = 105$
- $83^5 \% 119 = (105 \; x \; 83) \% 119 = 8715 \% 119 = 104$

Ciphertext = 104

Untuk M = 104

$C = 105^5 \% 119$

Hitung 105^5 menggunakan pengurangan modular

- $105^2 \% 119 = (105 \; x \; 105) \% 119 = 11025 \% 119 = 102$
- $105^4 \% 119 = (102 \; x \; 102) \% 119 = 10404 \% 119 = 105$
- $105^5 \% 119 = (105 \; x \; 105) \% 119 = 11025 \% 119 = 56$

Ciphertext = 56

Untuk M = 116

$C = 116^5 \% 119$

Hitung 116^5 menggunakan pengurangan modular

- $116^2 \% 119 = (116 \; x \; 116) \% 119 = 13456 \% 119 = 25$
- $116^4 \% 119 = (25 \; x \; 25) \% 119 = 625 \% 119 = 31$
- $116^5 \% 119 = (31 \; x \; 116) \% 119 = 3596 \% 119 = 114$

Langkah 3 = > Ciphertext Akhir

Hasil enkripsi untuk plaintext "Sitti" adalah = > C=[104,56,114,114,56]

```python
def mod_exp(base, exp, mod):
    result = 1
    while exp > 0:
        if exp % 2 == 1:
            result = (result * base) % mod
        base = (base * base) % mod
        exp //= 2
    return result
def rsa_encrypt(plaintext, e, n):
    return [mod_exp(ord(char), e, n) for char in plaintext]
e = 5
n = 119
plaintext = "Sitti"
ciphertext = rsa_encrypt(plaintext, e, n)
print(f"Plaintext (ASCII): {[ord(char) for char in plaintext]}")
print(f"Ciphertext: {ciphertext}")

```

```
[Running] python -u "e:\CODINGAN\Tugas Kripto\Quis\tempCodeRunnerFile.py"
Plaintext (ASCII): [83, 105, 116, 116, 105]
Ciphertext: [104, 56, 114, 114, 56]

[Done] exited with code=0 in 0.115 seconds
```

- Deskripsi RSA

  Kunci Publik e = 5, n = 119
  Kunci Private d = 77, n = 119

  Ciphertext yang Diberikan [71,63,36,36,63]

  Rumus Dekripsi RSA $M = C^d \% n$

  DI mana
  - C Adalah ciphertext
  - d Adalah kunci private
  - n Adalah modulus
  - M adalah plaintext yang di deskripsi

  Langkah 1 = > Dekripsi Setiap Karakter

Dekripsi untuk C = 71

$M = 71^{77} \% 119 = 71^{77} \% 119 = 83$ (S)

Dekripsi untuk C = 63

$M = 63^{77} \% 119 = 63^{77} \% 119 = 105$ (i)

Dekripsi untuk C = 36

$M = 36^{77} \% 119 = 36^{77} \% 119 = 116$ (t)

Dekripsi untuk C = 36

$M = 36^{77} \% 119 = 36^{77} \% 119 = 116$ (t)

Dekripsi untuk C = 63

$M = 63^{77} \% 119 = 63^{77} \% 119 = 105$ (i)

Langkah 2 = > Mengubah Hasil Dekripsi ke dalam Bentuk String

M = [83,105,116,116,105]

Plaintext = "Sitti"

```python
def mod_exp(base, exp, mod):
    result = 1
    while exp > 0:
        if exp % 2 == 1:
            result = (result * base) % mod
        base = (base * base) % mod
        exp //= 2
    return result
def rsa_decrypt(ciphertext, d, n):
    return ''.join(chr(mod_exp(char, d, n)) for char in ciphertext)
e = 5
d = 77
n = 119
ciphertext = [mod_exp(ord(char), e, n) for char in "Sitti"]
plaintext = rsa_decrypt(ciphertext, d, n)
print(f"Ciphertext: {ciphertext}")
print(f"Plaintext (ASCII): {[ord(char) for char in plaintext]}")
print(f"Plaintext (String): {plaintext}")

```

```
[Running] python -u "e:\CODINGAN\Tugas Kripto\Quis\RSA - Deskripsi.py"
Ciphertext: [104, 56, 114, 114, 56]
Plaintext (ASCII): [83, 105, 116, 116, 105]
Plaintext (String): Sitti

[Done] exited with code=0 in 0.105 seconds
```