



LABORATORIUM PEMBELAJARAN ILMU KOMPUTER
FAKULTAS ILMU KOMPUTER
UNIVERSITAS BRAWIJAYA

BAB : KRIPTOGRAFI
NAMA : RAMA SAKTI GAGAH PRAWIRATAMA
NIM : 215150200111011
TANGGAL : 29/05/2023
ASISTEN : DIMAS TRI MUSTAKIM

A. ENKRIPSI / DEKRIPSI

1. Install library pyCryptodome

```
pip install PyCryptodome  
# pengguna Endeavor/ArchLinux dapat pula menggunakan pacman  
sudo pacman -S python-pycryptodome
```

2. Jalankan kode berikut:

```
from Crypto.Cipher import AES  
from Crypto.Util.Padding import pad  
  
key = b'KunciRahasiaSaya'  
cipher = AES.new(key, AES.MODE_ECB)  
  
message = b'Pesanrahasia123'  
padded_message = pad(message, AES.block_size)  
ciphertext = cipher.encrypt(padded_message)  
ciphertext
```

3. Jalankan kode berikut:

```
from Crypto.Cipher import AES  
  
key = b'KunciRahasiaSaya'  
cipher2 = AES.new(key, AES.MODE_ECB)  
cipher2.decrypt(ciphertext)
```

Penjelasan output

```
kali kali ~ python
Python 3.11.2 (main, Feb 12 2023, 00:48:52) [GCC 12.2.0] on linux
Type "help", "copyright", "credits" or "license" for more information.
>>> from Crypto.Cipher import AES
>>> from Crypto.Util.Padding import pad
>>> key = b'KunciRahasiaSaya'
>>> cipher = AES.new(key, AES.MODE_ECB)
>>> message = b'Pesanrahasia123'
>>> padded_message = pad(message, AES.block_size)
>>> ciphertext = cipher.encrypt(padded_message)
>>> ciphertext
b'\x11\xc5\x9d\xdc\xf4b\xf1;\xd0\xc7\xf2\xb10H\xe6'
>>> from Crypto.Cipher import AES
>>> key = b'KunciRahasiaSaya'
>>> cipher2 = AES.new(key, AES.MODE_ECB)
>>> cipher2.decrypt(ciphertext)
b'Pesanrahasia123\x01'
>>> █
```

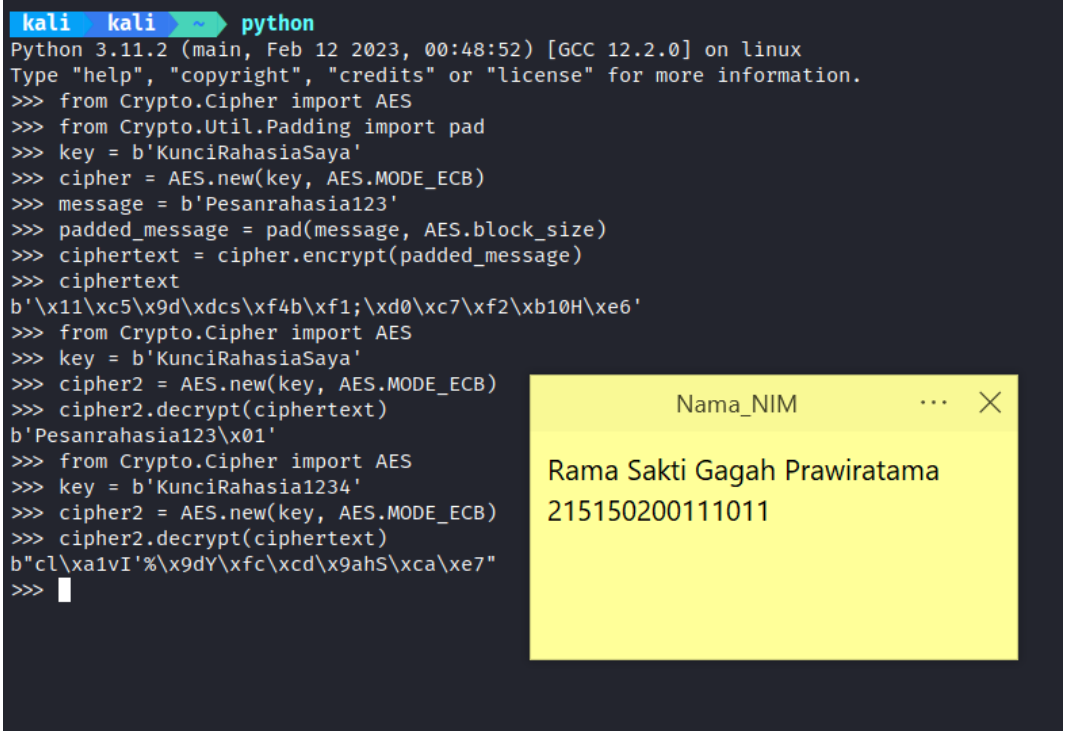
Nama_NIM

Rama Sakti Gagah Prawiratama
215150200111011

Ketika decrypt sebuah pesan enkripsi dengan key yang sama maka akan muncul sebuah pesan tersembunyi yaitu Pesanrahasia123.

4. Pada langkah 3, gantilah nilai key = 'KunciRahasia1234'. Apakah error yang Anda dapatkan?

Penjelasan output



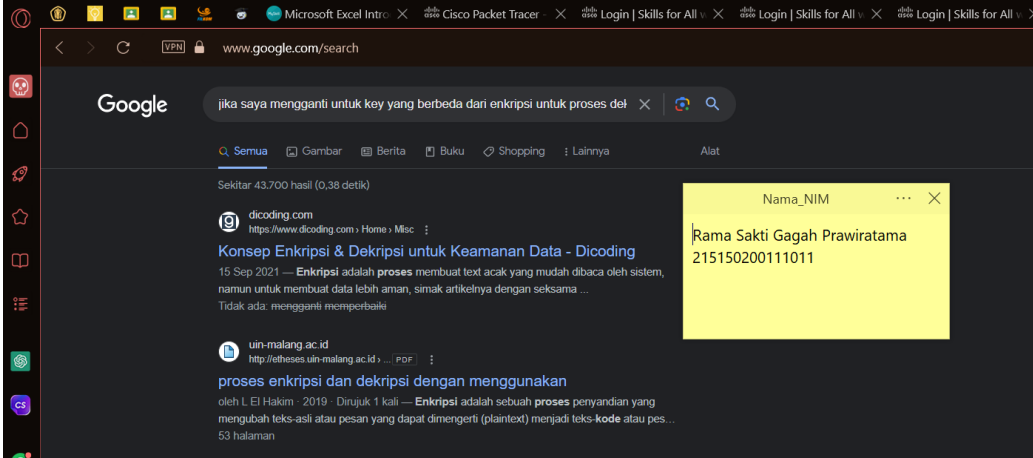
```

kali kali python
Python 3.11.2 (main, Feb 12 2023, 00:48:52) [GCC 12.2.0] on linux
Type "help", "copyright", "credits" or "license" for more information.
>>> from Crypto.Cipher import AES
>>> from Crypto.Util.Padding import pad
>>> key = b'KunciRahasiaSaya'
>>> cipher = AES.new(key, AES.MODE_ECB)
>>> message = b'Pesanrahasia123'
>>> padded_message = pad(message, AES.block_size)
>>> ciphertext = cipher.encrypt(padded_message)
>>> ciphertext
b'\x11\xc5\x9d\xdcs\xf4b\xf1;\xd0\xc7\xf2\xb10H\xe6'
>>> from Crypto.Cipher import AES
>>> key = b'KunciRahasiaSaya'
>>> cipher2 = AES.new(key, AES.MODE_ECB)
>>> cipher2.decrypt(ciphertext)
b'Pesanrahasia123\x01'
>>> from Crypto.Cipher import AES
>>> key = b'KunciRahasia1234'
>>> cipher2 = AES.new(key, AES.MODE_ECB)
>>> cipher2.decrypt(ciphertext)
b"cl\xalvI'%\x9dY\xfc\xcd\x9ahS\xca\xe7"
>>>

```

Ketika key untuk decrypt tidak sama dengan key di encrypt pesan tersembunyi nya tidak akan muncul atau salah.

5. Koreksilah hasil error pada nomor 4 dengan menggunakan rekomendasi hasil search engine yang biasa Anda gunakan. Screen capture kode hasil koreksi dan hasilnya.



Google search results for "jika saya mengganti untuk key yang berbeda dari enkripsi untuk proses det". The results show a link to "Konsep Enkripsi & Dekripsi untuk Keamanan Data - Dicoding" and another link to "proses enkripsi dan dekripsi dengan menggunakan". A yellow sticky note is placed over the search results, containing the text "Nama_NIM" and "Rama Sakti Gagah Prawiratama 215150200111011".

6. Koreksilah hasil error pada nomor 4 dengan menggunakan hasil rekomendasi ChatGPT. Screen capture kode hasil koreksi dan hasilnya.

Here's an example of how you can modify the decryption source code to obtain the original plaintext message:

```
python Copy code  
  
from Crypto.Cipher import AES  
from Crypto.Util.Padding import unpad  
  
key = b'KunciRahasiaSaya'  
cipher2 = AES.new(key, AES.MODE_ECB)  
decrypted_message = unpad(cipher2.decrypt(ciphertext), AES.block_size)  
print(decrypted_message)
```

In this modified version, we use the `unpad` function from `Crypto.Util.Padding` to remove the padding from the decrypted ciphertext. Make sure to import the `unpad` function to use it in the code. The resulting `decrypted_message` should be the original plaintext message.

Double-checking the mentioned aspects should help ensure that the decryption produces the correct output. Regenerate response

Nama_NIM

Rama Sakti Gagah Prawiratama
215150200111011

```
>>> from Crypto.Cipher import AES  
>>> from Crypto.Util.Padding import unpad  
>>>  
>>> key = b'KunciRahasiaSaya'  
>>> cipher2 = AES.new(key, AES.MODE_ECB)  
>>> decrypted_message = unpad(cipher2.decrypt(ciphertext), AES.block_size)  
>>> print(decrypted_message)  
b'Pesannrahasia123'  
>>> █
```

Nama_NIM

Rama Sakti Gagah Prawiratama
215150200111011

7. Amati hasil dari output langkah 5 dan 6, apakah yang dapat disimpulkan?

Jika mencari solusi error pada search engine browser, tidak muncul sebuah solusi untuk menyelesaikannya. Namun jika menggunakan ChatGPT solusi akan diberikan lengkap dengan penjelasan nya.

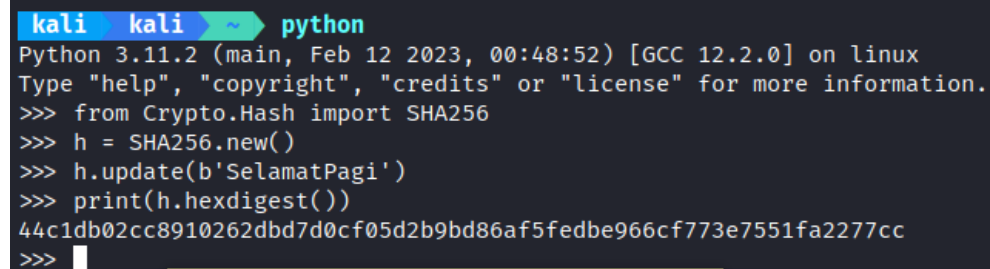
B. HASHING

1. Jalankan kode berikut

```
from Crypto.Hash import SHA256
h = SHA256.new()
h.update(b'SelamatPagi')

print(h.hexdigest())
```

Penjelasan output



```
kali kali ~ python
Python 3.11.2 (main, Feb 12 2023, 00:48:52) [GCC 12.2.0] on linux
Type "help", "copyright", "credits" or "license" for more information.
>>> from Crypto.Hash import SHA256
>>> h = SHA256.new()
>>> h.update(b'SelamatPagi')
>>> print(h.hexdigest())
44c1db02cc8910262dbd7d0cf05d2b9bd86af5fedbe966cf773e7551fa2277cc
>>> █
```

Output nya merupakan sebuah nilai hash dengan menggunakan metode hexdigest, metode yang mengembalikan representasi hexadesimal dari nilai hash. Output yang dikeluarkan sebanyak 64 karakter

2. Jalankan kode berikut

```
from Crypto.Hash import SHA512
h = SHA512.new()
h.update(b'SelamatPagi')
print(h.hexdigest())
```

Penjelasan output

```
kali kali python
Python 3.11.2 (main, Feb 12 2023, 00:48:52) [GCC 12.2.0] on linux
Type "help", "copyright", "credits" or "license" for more information.
>>> from Crypto.Hash import SHA256
>>> h = SHA256.new()
>>> h.update(b'SelamatPagi')
>>> print(h.hexdigest())
44c1db02cc8910262dbd7d0cf05d2b9bd86af5fedbe966cf773e7551fa2277cc
>>> from Crypto.Hash import SHA512
>>> h = SHA512.new()
>>> h.update(b'SelamatPagi')
>>> print(h.hexdigest())
516731c65849d27931955e702d5cf9bedddc85faf598a1ecaef8b862c462d1d86b4d40a56c04a4f40bea51eceebe954377d522d18ed8f4a5614a96e2dcbe8bd3
>>>
```

Nama_NIM ... X
Rama Sakti Gagah Prawiratama
215150200111011

Output nya merupakan sebuah nilai hash dengan menggunakan metode hexdigest, metode yang mengembalikan representasi hexadesimal dari nilai hash. Output yang dikeluarkan sebanyak 128 karakter

3. Jalankan kode berikut

```
h = MD5.new()
h.update(b'SelamatPagi')
print(h.hexdigest())
```

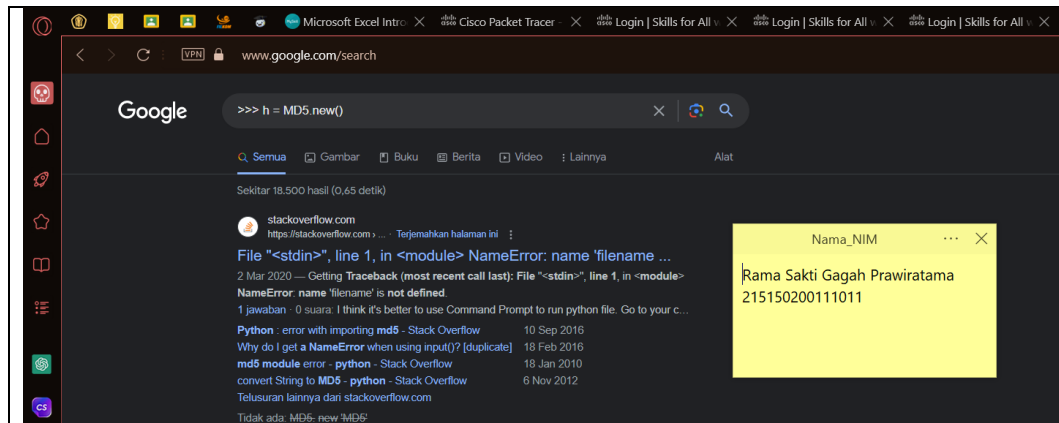
Penjelasan output

```
>>> h = MD5.new()
Traceback (most recent call last):
  File "<stdin>", line 1, in <module>
NameError: name 'MD5' is not defined
>>> h.update(b'SelamatPagi')
>>> print(h.hexdigest())
d823d98a5501f96210bd2a5818bd2a3136371f8b911d9779106298c8ff07233af18b58712b09dd44f53206bdf38cc3faeb0e7c2d5dda670c9626810ab29e4105
>>>
```

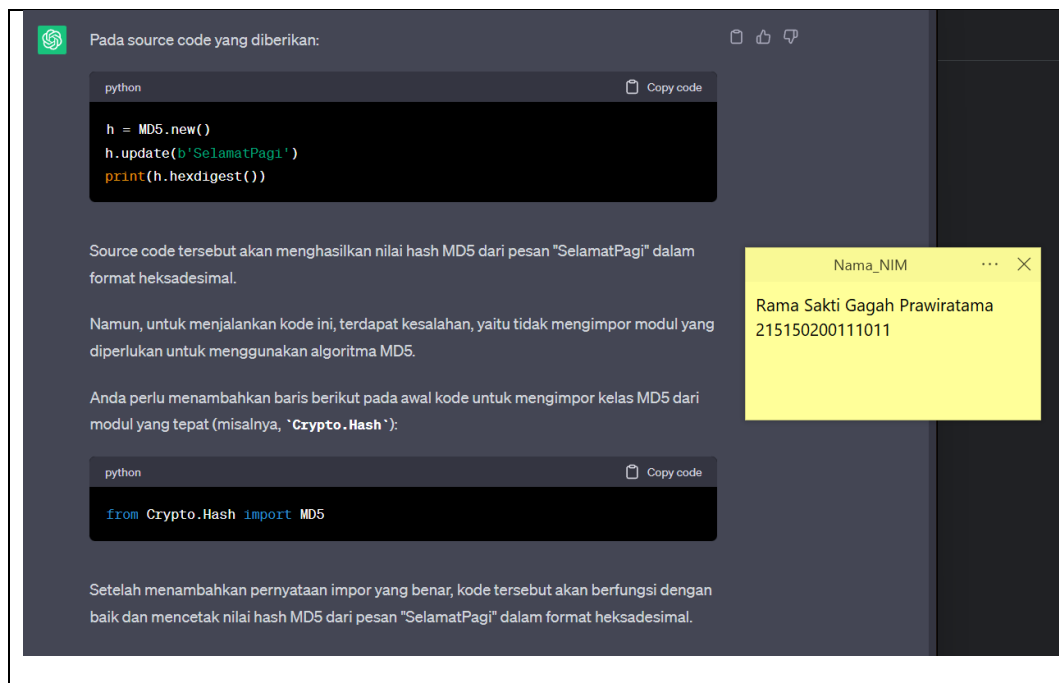
Nama_NIM ... X
Rama Sakti Gagah Prawiratama
215150200111011

Terjadi kesalahan karena tidak mengimpor modul yang diperlukan untuk menggunakan algoritma MD5

4. Koreksilah hasil error pada nomor 3 dengan menggunakan rekomendasi hasil search engine yang biasa Anda gunakan. Screen capture kode hasil koreksi dan hasilnya.



5. Koreksilah hasil error pada nomor 3 dengan menggunakan hasil rekomendasi ChatGPT. Screen capture kode hasil koreksi dan hasilnya



```
>>> from Crypto.Hash import MD5
>>>
>>> h = MD5.new()
>>> h.update(b'SelamatPagi')
>>> print(h.hexdigest())
51bdb070d5065c5eb07497b1197d1af0
>>>
```

Nama_NIM ... X
Rama Sakti Gagah Prawiratama
215150200111011

6. Amati hasil dari output langkah 4 dan 5, apakah yang dapat disimpulkan?

Jika mencari solusi error pada search engine browser, tidak muncul sebuah solusi untuk menyelesaikannya. Namun jika menggunakan ChatGPT solusi akan diberikan lengkap dengan penjelasan nya.

C. DIGITAL SIGNATURE

1. Jalankan kode berikut

```
from Crypto.PublicKey import RSA

kunci = RSA.generate(bits=1024)

print(f"Kunci Publik: (e={hex(kunci.e)})")
print(f"Kunci Privat: (e={hex(kunci.d)})")
```

Penjelasan output

```
kali kali ~ python
Python 3.11.2 (main, Feb 12 2023, 00:48:52) [GCC 12.2.0] on linux
Type "help", "copyright", "credits" or "license()" for more information.
>>> from Crypto.PublicKey import RSA
>>> kunci = RSA.generate(bits=1024)
>>> print(f"Kunci Publik: (e={hex(kunci.e)})")
Kunci Publik: (e=a=10001)
>>> print(f"Kunci Privat: (e={hex(kunci.d)})")
Kunci Privat: (e=0x2fa0302a54072ff0e12e3f0a2f7121bc82a780ee0bcb63120fe294c2ee678a5e48deeeb5286feabf8c40536572631046966d812bcb11aa2d27ade4f89259380c96f84c20e07105411c310097c77ed0e6acc71802b3d62e02c9bc78d17e9f0a0b769720c545b311b58478966731329278abce30c40b705009933aic5)
>>>
```

Nama_NIM ... X
Rama Sakti Gagah Prawiratama
215150200111011

Menampilakn 2 buah kunci, kunci publik dan kunci privat, yang dimana kunci publik untuk enkripsi dan kunci private untuk decrypt

2. Jalankan kode berikut

```
pesan = 234
sign = pow(pesan, kunci.d, kunci.n)
print("Digital Signature:", hex(sign))
```

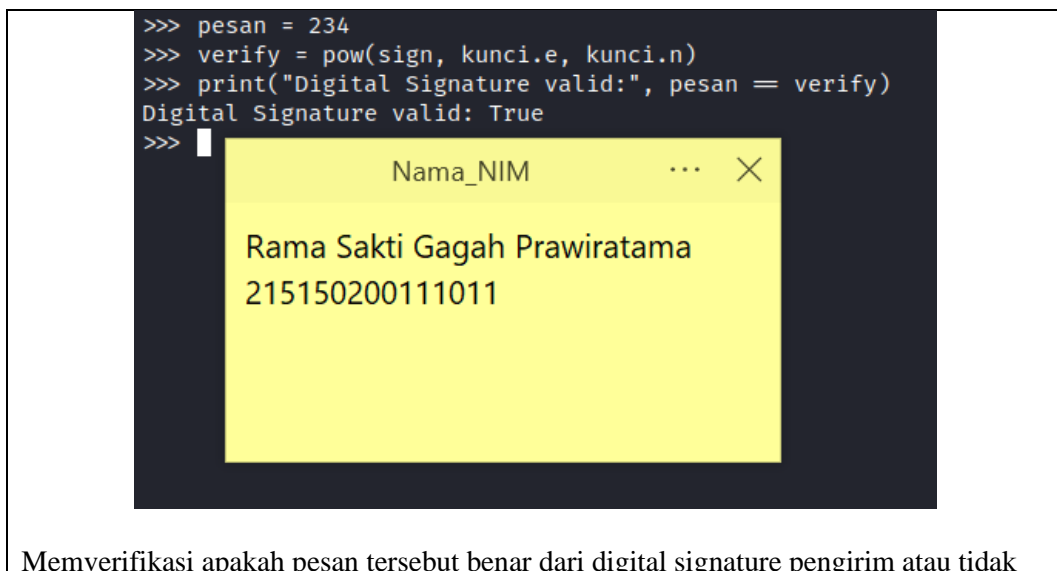
Penjelasan output



3. Jalankan kode berikut

```
pesan = 234
verify = pow(sign, kunci.e, kunci.n)
print("Digital Signature valid:", pesan == verify)
```

Penjelasan output



4. Jalankan kode berikut

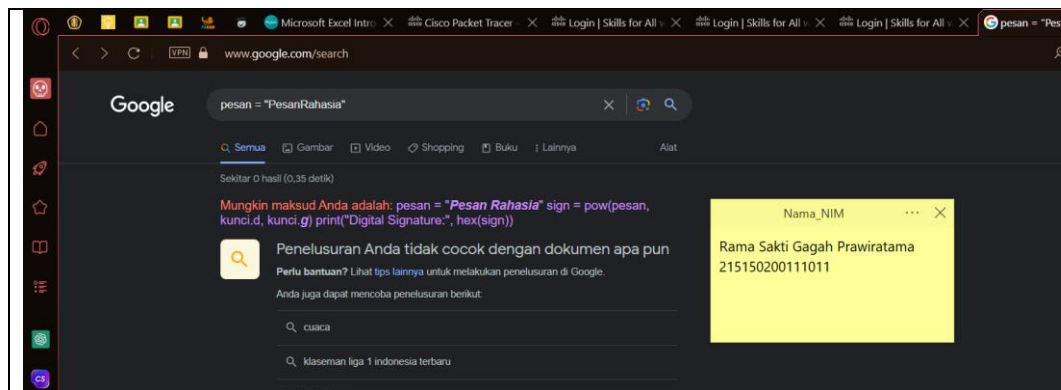
```
pesan = "PesanRahasia"
sign = pow(pesan, kunci.d, kunci.n)
print("Digital Signature:", hex(sign))
```

Penjelasan output

```
>>> pesan = "PesanRahasia"
>>> sign = pow(pesan, kunci.d, kunci.n)
Traceback (most recent call last):
  File "<stdin>", line 1, in <module>
TypeError: unsupported operand type(s) for ** or pow(): 'str', 'int', 'int'
>>> print("Digital Signature:", hex(sign))
Digital Signature: 0x904ec863b01c971db1be0b046fa1888fffe66b3e5738c567e0278e47ca7c5c59d2bfc924d334c21e46c6dca8edd3485ef07f1f86e5715f3a23b58ba35783aaf9c9e21cda795f3c2edc193df6790c977beca5ec685d139bedf92629
>>>
```

Akan mengalami error karena terdapat operasi yang tidak valid. Tanda tangan digital dalam kriptografi RSA harus diterapkan pada nilai numerik, bukan pada string teks biasa.

5. Koreksilah hasil error pada nomor 4 dengan menggunakan rekomendasi hasil search engine yang biasa Anda gunakan. Screen capture kode hasil koreksi dan hasilnya



6. Koreksilah hasil error pada nomor 4 dengan menggunakan hasil rekomendasi ChatGPT. Screen capture kode hasil koreksi dan hasilnya

Untuk memperbaiki kesalahan tersebut, Anda perlu mengonversi pesan menjadi representasi numerik yang sesuai. Salah satu pendekatan adalah mengonversi pesan ke dalam bentuk bilangan bulat menggunakan encoding UTF-8.

Berikut adalah contoh perbaikan kode dengan menggunakan bilangan bulat menggunakan encoding UTF-8:

python

Copy code

```
pesan = "PesanRahasia"
pesan_encoded = pesan.encode('utf-8')
pesan_int = int.from_bytes(pesan_encoded, byteorder='big')

sign = pow(pesan_int, kunci.d, kunci.n)
print("Digital Signature:", hex(sign))
```

Pada kode di atas, kami menggunakan metode `encode()` untuk mengonversi pesan ke dalam bentuk byte menggunakan encoding UTF-8. Kemudian, kami menggunakan metode

```
>>> pesan = "PesanRahasia"
>>> pesan_encoded = pesan.encode('utf-8')
>>> pesan_int = int.from_bytes(pesan_encoded, byteorder='big')
>>> sign = pow(pesan_int, kunci.d, kunci.n)
>>> print("Digital Signature:", hex(sign))
Digital Signature: 0x2ecae372d26cd03b7794d986c087cde8eeb83a237eb17bc123df2eaf930877b198d4585267aeb3e3c2987c1f75e868e2c98849c88c564c957577f4938d488c2dc1c9d38ea3abd35108cb11ccff25c75c4f88fce91c523585798c5d8
>>>
```

Nama_NIM

Rama Sakti Gagah Prawiratama
215150200111011

KESIMPULAN

Kriptografi adalah bidang ilmu yang berkaitan dengan teknik-teknik untuk mengamankan dan melindungi informasi. Berikut adalah kesimpulan singkat mengenai konsep-konsep kriptografi yang dibahas:

1. Enkripsi dan Dekripsi:

- Enkripsi adalah proses mengubah data menjadi bentuk yang tidak terbaca atau tidak dapat dimengerti oleh pihak yang tidak berwenang.
- Dekripsi adalah proses mengembalikan data yang telah dienkripsi menjadi bentuk aslinya menggunakan kunci yang sesuai.

2. Hashing:

- Hashing adalah proses mengubah data menjadi nilai hash yang unik dan tetap dengan panjang tetap.
- Nilai hash digunakan untuk memverifikasi integritas data dan tidak dapat diubah kembali menjadi data asli.

3. Tanda Tangan Digital:

- Tanda tangan digital digunakan untuk memastikan keaslian dan integritas pesan atau dokumen.
- Tanda tangan digital dibuat dengan menggunakan kunci privat untuk menghasilkan tanda tangan, dan dapat diverifikasi menggunakan kunci publik yang sesuai.

Dalam kriptografi, enkripsi dan dekripsi digunakan untuk melindungi kerahasiaan data, hashing digunakan untuk memverifikasi integritas data, dan tanda tangan digital

digunakan untuk memastikan keaslian pesan. Kriptografi berperan penting dalam keamanan komunikasi, perlindungan data, dan memastikan bahwa informasi tetap rahasia dan tidak berubah.

EVALUASI

1. Jelaskan perbedaan dari proses enkripsi dan hashing

1. Tujuan:

- Enkripsi: Tujuan utama dari enkripsi adalah untuk melindungi kerahasiaan data dengan mengubahnya menjadi bentuk yang tidak terbaca atau tidak dapat dimengerti oleh pihak yang tidak berwenang. Enkripsi menggunakan algoritma kunci simetris atau asimetris untuk mengubah data menjadi bentuk yang dienkripsi.
- Hashing: Tujuan utama dari hashing adalah untuk menghasilkan nilai hash yang unik untuk setiap input. Hashing menggunakan algoritma hash untuk mengonversi data atau pesan menjadi nilai hash tetap dengan panjang tetap. Tujuan utama hashing adalah untuk mengintegritaskan data dan memverifikasi apakah data telah berubah atau tidak.

2. Operasi:

- Enkripsi: Proses enkripsi melibatkan penggunaan kunci rahasia untuk mengubah data menjadi bentuk yang tidak terbaca. Ini melibatkan operasi matematika yang kompleks seperti substitusi, permutasi, atau kombinasi dari keduanya.
- Hashing: Proses hashing melibatkan penggunaan algoritma hash yang mengambil input data dan menghasilkan nilai hash unik dengan panjang tetap. Operasi hashing bersifat satu arah, artinya nilai hash tidak dapat diubah kembali menjadi data asli tanpa mengorbankan keamanan.

3. Keluaran:

- Enkripsi: Keluaran dari proses enkripsi adalah data yang diubah menjadi bentuk yang dienkripsi. Untuk mengembalikan data ke bentuk aslinya, proses dekripsi dengan menggunakan kunci yang sama diperlukan.
- Hashing: Keluaran dari proses hashing adalah nilai hash, yaitu representasi numerik tetap dengan panjang tetap yang unik untuk setiap input. Nilai hash tidak dapat diubah kembali menjadi data asli, dan tugas utama hashing adalah membandingkan nilai hash yang dihasilkan untuk memverifikasi integritas data.

4. Penggunaan:

- Enkripsi: Enkripsi digunakan untuk melindungi kerahasiaan data saat disimpan atau ditransmisikan. Ini digunakan dalam komunikasi aman, penyimpanan data yang terenkripsi, dan proteksi data pribadi.
- Hashing: Hashing digunakan untuk memverifikasi integritas data dan memastikan bahwa data tidak berubah saat transit atau penyimpanan. Ini digunakan dalam verifikasi kata sandi, penentuan kesesuaian data, dan pengecekan integritas file.

2. Jalankan algoritma hashing lain yaitu SHA384, dan amati hasil outputnya dan simpulkan

- SHA-384 adalah algoritma hash yang termasuk dalam keluarga Secure Hash Algorithm (SHA).
- SHA-384 menghasilkan nilai hash dengan panjang 384 bit atau 48 byte.
- Hasil output berupa nilai hash dalam bentuk heksadesimal.
- SHA-384 menghasilkan nilai hash yang unik untuk setiap input yang berbeda.
- Perubahan kecil pada input akan menghasilkan perubahan drastis pada nilai hash.
- SHA-384 digunakan untuk verifikasi integritas data, keamanan penyimpanan, dan validasi file.
- SHA-384 dianggap lebih kuat dan lebih aman daripada algoritma hash yang lebih pendek seperti SHA-256 atau MD5.