BAB III
METODOLOGI PENELITIAN

Bagian ini membahas terkait rencana penelitian yang dilakukan, mulai dari identifikasi masalah, pengembangan model dasar, konstruksi program aplikasi, hingga rancangan tampilan program aplikasi.

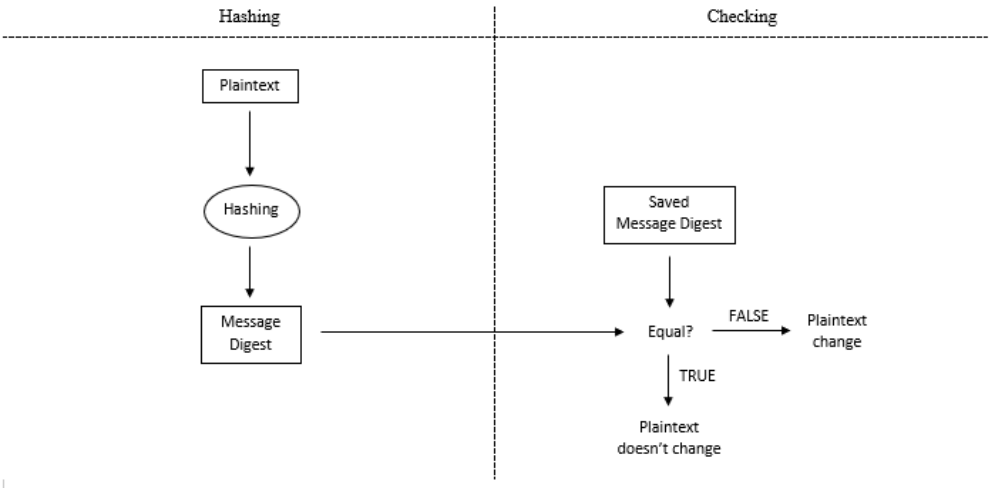
3.1 Identifikasi Masalah

Di era yang serba teknologi saat ini, banyak terjadi kasus pemalsuan sebuah dokumen digital. Dahulu, berbagai institusi merekap dokumen dengan tulis tangan sehingga cukup sulit dipalsukan, tetapi sebagian besar institusi kini sudah menerapkan sistem file PDF dengan tanda tangan digital pihak terkait yang berfungsi untuk menjaga keaslian sebuah dokumen. Dengan adanya sistem digital ini, tidak dapat menjamin bahwa dokumen tersebut aman dan belum dirusak oleh oknum tak berkepentingan.

Oleh karena itu, penelitian ini menggunakan fungsi hash SHA-256 dan algoritma kunci publik RSA yang ditingkatkan untuk mengecek keaslian dari sebuah dokumen digital dengan berbasis QR Code.

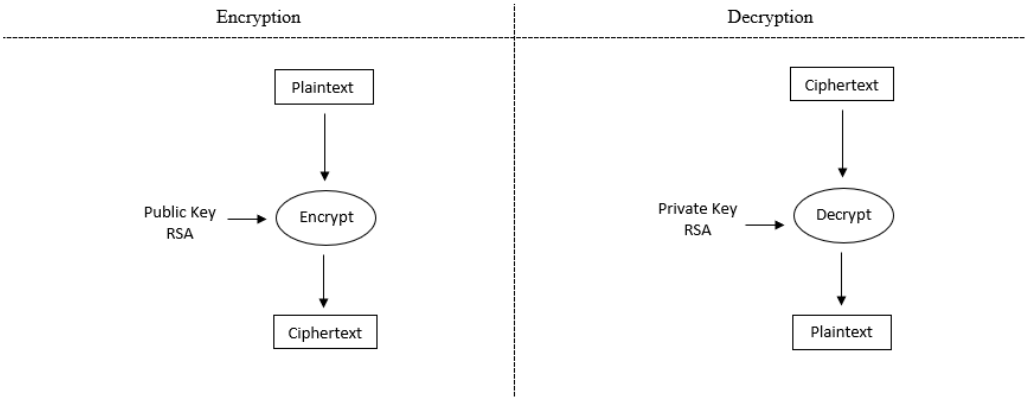
3.2 Model Dasar

Terdapat dua model dasar yang digunakan dalam penelitian ini, yaitu fungsi hash dan kriptografi asimetri.



Gambar 3.1 Skema Fungsi Hash

Skema pada Gambar 3.1 menjelaskan bahwa fungsi hash (h) dilakukan kepada plainteks (M) sehingga menghasilkan nilai hash atau *message digest* (h(M)), lalu dilakukan pencocokan terhadap nilai hash yang dimiliki. Hasil diterima ketika kedua *message digest* tersebut memiliki kecocokan, sedangkan hasil ditolak ketika kedua *message digest* tersebut tidak memiliki kecocokan dikarenakan terjadinya perubahan pada plainteks yang dimasukkan.

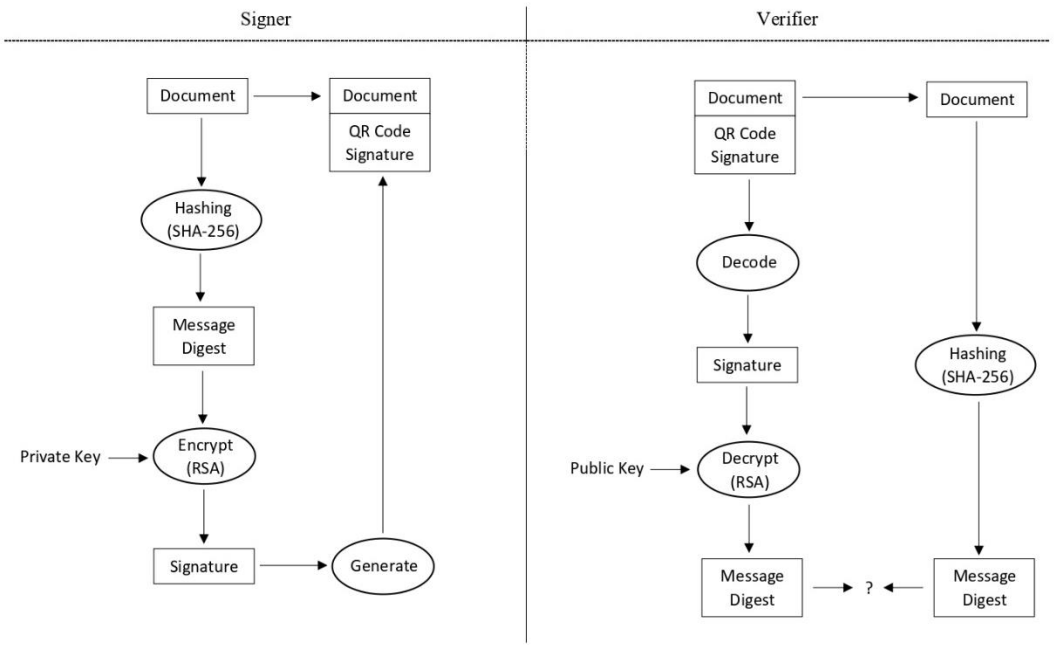


Gambar 3.2 Skema Algoritma RSA

Skema pada Gambar 3.2 menjelaskan bahwa proses enkripsi dan dekripsi pada RSA menggunakan dua kunci yang berbeda. Dilakukan enkripsi plainteks oleh pengirim menggunakan kunci publik penerima, lalu dekripsi pada cipherteks akan dilakukan dengan menggunakan kunci privat penerima. Skema RSA yang ditingkatkan memiliki kesamaan dengan skema RSA standar.

3.3 Pengembangan Model

Dalam mengembangkan model dasar untuk penelitian ini menggabungkan kedua algoritma yang telah diuraikan sebelumnya, di mana fungsi hash yang digunakan adalah SHA-256 dan kriptografi asimetri yang digunakan adalah RSA yang ditingkatkan. Selain berfungsi sebagai enkripsi dan dekripsi pesan, algoritma RSA yang ditingkatkan juga dapat digunakan dalam pembuatan tanda tangan digital. Selain menggabungkan algoritma fungsi hash dan kriptografi asimetri, hasil tanda tangan digital dari proses enkripsi RSA yang ditingkatkan akan dibuat ke dalam QR Code sebelum diberikan kepada penerima.



Gambar 3.3 Skema Pengembangan Model

Skema pada Gambar 3.3 menjelaskan bahwa pengirim akan melakukan *hashing* terhadap dokumen dengan SHA-256 dan menghasilkan *message digest*, lalu dilakukan enkripsi RSA yang ditingkatkan dengan kunci privat pengirim sehingga menghasilkan tanda tangan digital berupa QR Code yang kemudian dikirim kepada penerima. Penerima dokumen dan QR Code harus melakukan *hashing* terhadap dokumen dengan SHA-256 sehingga menghasilkan *message digest*, serta lakukan pula dekripsi RSA yang ditingkatkan terhadap tanda tangan digital dengan kunci publik sehingga menghasilkan *message digest*. Untuk memeriksa keaslian dokumen, penerima harus menyamakan kedua *message digest* yang telah diperoleh. Jika cocok maka dokumen dengan tanda tangan digital dijamin keasliannya, dan jika tidak cocok maka dokumen dengan tanda tangan digital diragukan keasliannya.

3.4 Konstruksi Program Aplikasi

Pembuatan program aplikasi dilakukan dengan bahasa pemrograman Python. Program aplikasi tersebut terdiri dari beberapa menu utama, seperti *key generator*, *signing*, dan *verifying*.

3.4.1 *Input* dan *Output*

Terdapat tiga menu utama, yaitu pembangkitan kunci, pembuatan tanda tangan digital, dan autentikasi dokumen digital. Pada menu pembangkitan kunci, *input* berupa tiga bilangan prima dan satu bilangan bulat, sedangkan *output* berupa sepasang kunci publik dan sepasang kunci privat. Pada menu pembuatan tanda tangan digital, *input* berupa sepasang kunci privat dan dokumen PDF, sedangkan *output* berupa QR Code. Pada menu autentikasi dokumen digital, *input* berupa sepasang kunci publik, dokumen PDF, dan QR Code, sedangkan *output* berupa penjelasan terkait keautentikan dokumen dan tanda tangan digital.

3.4.2 Algoritma Deskriptif

Program aplikasi yang akan dibuat menggunakan tiga algoritma utama dalam melakukan autentikasi dokumen digital dengan SHA-256 dan RSA yang ditingkatkan, yaitu:

A. Pembangkitan Kunci

Algoritma pertama dilakukan oleh pengirim dengan membangkitkan sepasang kunci publik yang berguna untuk autentikasi dokumen dan sepasang kunci privat yang berguna untuk pembuatan tanda tangan.

1. Pilih tiga bilangan prima berbeda yang besar secara acak, p, q, dan r.
2. Hitung $n = pqr$.
3. Hitung $\varphi(n) = (p - 1)(q - 1)(r - 1)$.
4. Pilih sebuah bilangan bulat e sebagai kunci publik dan harus relatif prima terhadap $\varphi(n)$, di mana $2 < e < \varphi(n)$.
5. Hitung kunci privat, d, dengan persamaan:
$$ed \equiv 1 \pmod{\varphi(n)} \text{ atau } d \equiv e^{-1} \pmod{\varphi(n)}$$
6. *Output* yang diperoleh, yaitu:
   a. Kunci publik adalah pasangan (e, n) dan tidak rahasia.
   b. Kunci privat adalah pasangan (d, n) dan harus dirahasiakan.

B. Pembuatan Tanda Tangan Digital (*Signer*)

Algoritma kedua dilakukan oleh pengirim untuk membuat tanda tangan digital. Pada algoritma ini, dimanfaatkan fungsi hash dan proses enkripsi RSA dengan kunci privat milik pengirim.

1. Pilih sebuah dokumen digital yang akan diberi tanda tangan.
2. Lakukan *hashing* atau proses meringkas dokumen menjadi pesan singkat berukuran 256-bit.
3. Enkripsi *message digest* dengan kunci privat yang sudah dibangkitkan.
4. *Generate* QR Code yang berisi tanda tangan digital hasil enkripsi.
5. Kirim sepasang dokumen digital dan QR Code kepada penerima.

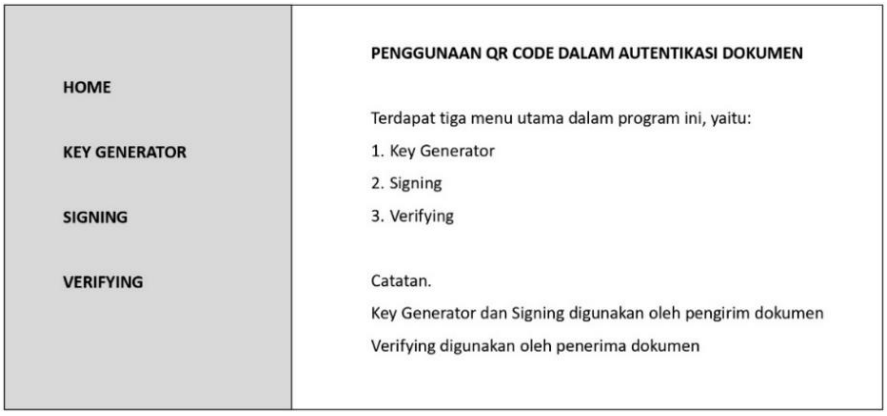
C. Autentikasi Dokumen Digital (*Verifier*)

Algoritma ketiga oleh penerima untuk mengecek keaslian sebuah dokumen digital. Pada algoritma ini, dimanfaatkan fungsi hash dan proses dekripsi RSA dengan kunci publik milik pengirim.

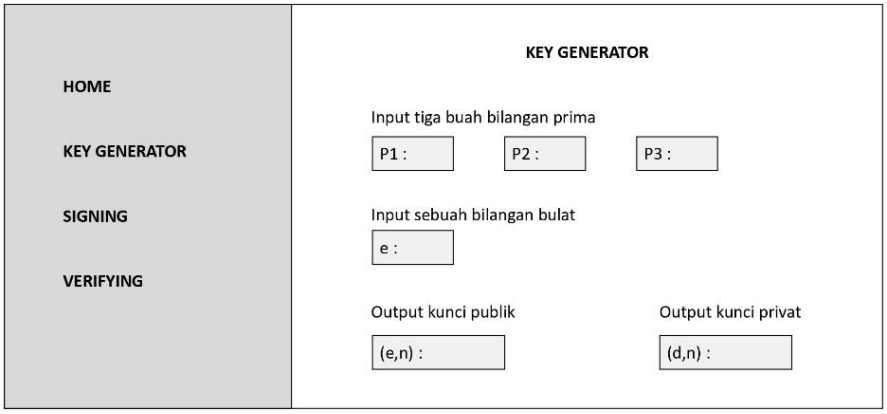
1. Lakukan *hashing* terhadap dokumen digital yang diterima sehingga diperoleh *message digest* dokumen (MD1).
2. Lakukan *decoding* terhadap QR Code yang diterima sehingga diperoleh tanda tangan digital.
3. Dekripsi tanda tangan digital dengan kunci publik yang sudah dibangkitkan sehingga diperoleh *message digest* tanda tangan (MD2).
4. Lakukan pencocokan terhadap MD1 dan MD2 dengan hasil:
   a. Jika cocok, maka dokumen digital dan QR Code autentik atau keduanya masih asli.
   b. Jika tidak cocok, maka dokumen digital dan QR Code tidak autentik atau terjadi perubahan pada salah satunya.

3.4.3 Desain Tampilan Program

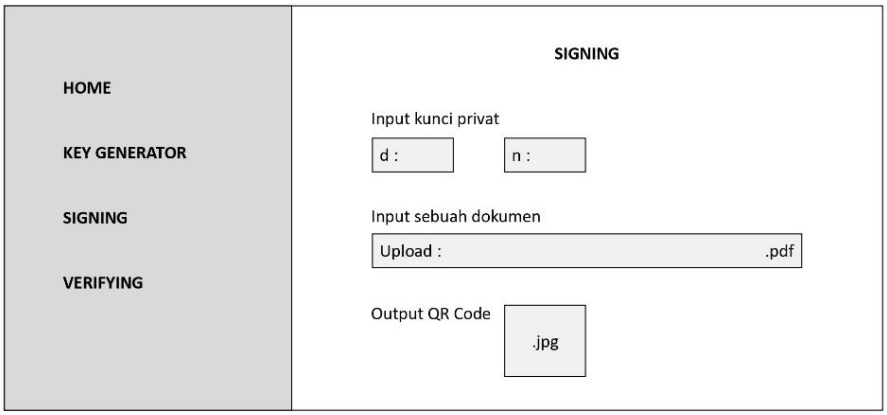
Berikut rancangan awal tampilan program aplikasi yang akan dibuat.



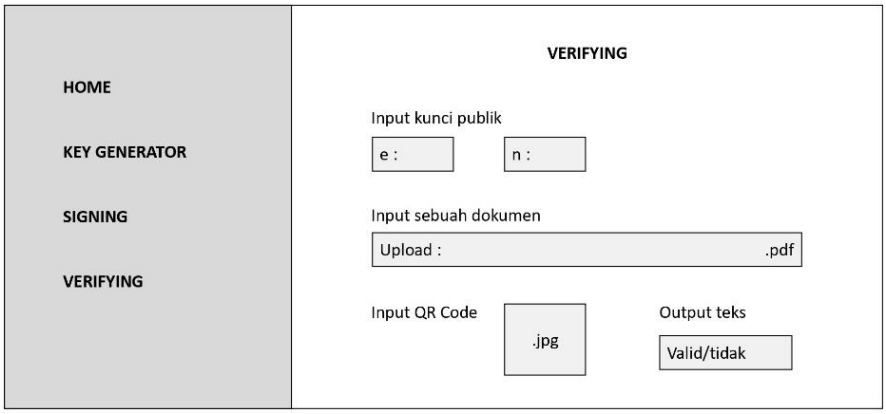
Gambar 3.4 Rancangan Tampilan Menu Utama Program



Gambar 3.5 Rancangan Tampilan Menu *Key Generator* Program



Gambar 3.6 Rancangan Tampilan Menu *Signing* Program



Gambar 3.7 Rancangan Tampilan Menu *Verifying* Program

3.4.4 *Library* Python

Pada pembuatan program autentikasi dokumen digital dengan SHA-256 dan RSA yang ditingkatkan, konstruksi akan dilakukan menggunakan aplikasi Python GUI (*Graphical User Interface*) dengan bantuan *software* Visual Studio Code.

Berikut beberapa penggunaan *library* dalam program aplikasi yang akan dirancang dalam penelitian ini (Van Rossum & Drake, 1995).

1. `tkinter`

   Modul `tkinter` merupakan *interface* Python standar yang digunakan untuk mengakses *toolkit* Tk GUI Python. Terdapat dua cara untuk memanggil modul ini, yaitu `import tkinter` atau `from tkinter import *`.

2. `messagebox`

   Modul `messagebox` termasuk modul yang termuat dalam Tk dan memberikan akses untuk memunculkan kotak dialog standar.

3. `filedialog`

   Modul `filedialog` adalah dialog umum yang memungkinkan pengguna menentukan file yang akan dibuka atau disimpan.

4. `hashlib`

   Modul `hashlib` mengimplementasikan *interface* umum ke dalam berbagai *secure hash* dan algoritma *message digest* yang berbeda, seperti SHA1, SHA224, SHA256, SHA384, dan SHA512, serta algoritma MD5 RSA.

5. `from PIL import Image, ImageTk`

   Modul `PIL` (*Python Imaging Library*) dirancang untuk akses cepat ke data yang disimpan dalam beberapa format pixel dasar sehingga memiliki kemampuan menambahkan pemrosesan gambar yang cukup kuat ke dalam suatu program.

   Class terpenting dalam PIL adalah `Image` yang dapat digunakan untuk memuat gambar dari file, memproses gambar lain, ataupun membuat gambar dari awal. *Class* `ImageTk` memuat dukungan untuk menciptakan dan mengubah Tkinter BitmapImage dan PhotoImage objects ke dalam PIL (Clark, 2015).

6. `cv2`

   Modul `OpenCV` mencakup serangkaian algoritma canggih yang komprehensif sehingga dapat digunakan dalam mendeteksi dan mengenali wajah, serta menemukan gambar serupa dari *database* gambar. *Frame* dapat dibaca atau ditulis ke dalam file dengan menggunakan modul `cv2` (Ismael & Irina, 2020).

7. `qrcode`

   Modul `qrcode` berguna dalam menghasilkan gambar QR Code (GNE Guerfi Sahra, 2022).

3.5 Proses Validasi

Proses validasi dibagi ke dalam tiga kasus, yaitu:

1. Jika *input* dokumen asli dan QR Code asli, maka *output* menyatakan bahwa dokumen digital dan tanda tangan elektronik autentik atau tidak ada perubahan pada keduanya.
2. Jika *input* dokumen palsu dan QR Code asli, maka *output* menyatakan bahwa dokumen digital dan tanda tangan elektronik tidak autentik atau sudah terjadi perubahan pada dokumen digital.
3. Jika *input* dokumen asli dan QR Code palsu, maka *output* menyatakan bahwa dokumen digital dan tanda tangan elektronik tidak autentik atau sudah terjadi perubahan pada tanda tangan elektronik.

3.6 Penarikan Kesimpulan

Penarikan kesimpulan akan diambil setelah melakukan percobaan terhadap program aplikasi yang dihasilkan dan melihat hasil validasinya. Setelah tervalidasi, program aplikasi dapat disimpulkan berjalan dengan baik dan diimplementasikan dalam pembuatan tanda tangan digital serta autentikasi keaslian dokumen.