Buletin Sistem Informasi dan Teknologi Islam
Vol 2, No 4, November 2021, pp. 255-264
ISSN: 2721-0901

Penerapan *Rivest Shamir Adleman* untuk Verifikasi Tanda Tangan *Digital* pada Lembar Pengesahan Skripsi

Muhammad Akbar[a,1,*], Poetri Lestari L.B[a,2], Farniwati Fattah[a,3]

[a]*Program Studi Teknik Informatikia, Fakultas Ilmu Komputer, Universitas Muslim Indonesia, Jl. Urip Sumohardjo KM.05, Makassar dan 90231, Indonesia*
[1]*akbarjie96@gmail.com;* [2]*poetrilestari@umi.ac.id;* [3]*farniwati.fattah@umi.ac.id*
**corresponding author*

INFORMASI ARTIKEL

Diterima : 08 – 09 – 2021
Direvisi : 28 – 10 – 2021
Diterbitkan : 30 – 11 – 2021

Kata Kunci:
Tanda tangan *digital*,
Rivest Shamir Adleman

ABSTRAK

Adanya kemudahan menandatangani lembar pengesahan skripsi secara digital menimbulkan masalah yaitu tanda tangan *digital* dapat diduplikasi sehingga mudah sekali digunakan oleh pihak yang tidak bertanggung jawab. Oleh karenanya dibutuhkan sistem verifikasi untuk mengenali keabsahan lembar pengesahan skripsi. Tujuan penelitian ini adalah menerapkan metode RSA untuk verifikasi tanda tangan digital pada lembar pengesahan skripsi dengan mengecek keabsahan lembar pengesahan skripsi mahasiswa, sehingga dapat mencegah tindakan curang mengesahkan lembar pengesahan skripsi oleh pihak yang tidak bertanggung jawabtanpa sepengetahuan dosen yang bersangkutan. Hasil dari penelitian yang telah dilakukan yaitu merancang aplikasi tanda tangan *digital* yang dapat memverifikasi tanda tangan dosen yang telah diekstrak menggunakan metode RSA pada lembar pengesahan skripsi mahasiswa berekstensi PDF, dan proses verifikasi sistem dengan cara membandingkan nilai hash antara tanda tangan digital yang tersimpan dalam *database* dengan tanda tangan *digital* yang termuat pada *file* lembar pengesahan skripsi mahasiswa.

I. Pendahuluan

Pengesahan dokumen tugas akhir mahasiswa Fakultas Ilmu Komputer Universitas Muslim Indonesia telah dapat dilakukan secara *digital*, mahasiswa yang skripsinya telah disetujui oleh kedua pembimbingnya, cukup mengirimkan lembar pengesahan skripsi kepada dosen pembimbing dan ketua program studi melalui media *chat online* seperti Whatsapp maupun Telegram yang sudah familiar digunakan oleh masyarakat pada umumnya.

Adanya kemudahan menandatangani lembar pengesahan skripsi secara *digital* menimbulkan masalah yaitu tanda tangan *digital* dapat diduplikasi sehingga mudah sekali digunakan oleh pihak yang tidak bertanggung jawab. Oleh karenanya dibutuhkan sistem verifikasi untuk mengenali keabsahan lembar pengesahan skripsi. *File* yang digunakana dalam melengkapi tanda tangan *digital* lembar pengesahan skripsi berektensi PDF.

Kecurangan selama ini terjadi adalah mahasiswa dapat mengajukan berkas pengurusan ijazah hanya dengan menambahkan tanda tangan dosen yang bersangkutan dari hasil menjiplak, meskipun revisi skripsi belum disetujui oleh dosen yang bersangkutan setelah mehasiswa melakukan ujian skripsi, sehingga dibutuhkan proses verifikasi terhadap lembar pengesahan skripsi mahasiswa,

Tanda tangan *digital* adalah salah satu teknologi yang dapat diterapkan untuk mengatasi permasalahan kegiatan-kegiatan yang memerlukan pengesahan dokumen dengan tanda tangan, baik urusan akademik maupun *non*-akademik [1]. Proses untuk menciptakan tanda tangan *digital* menggunakan enkripsi asimetris dimana *private key* digunakan untuk mengenkripsi pesan, maka *public key* digunakan untuk mendekripsi pesan tersebut [2]. Proses verifikasi yaitu membandingkan tanda tangan *digital* yang telah didekripsi menggunakan kunci *public* dengan nilai hasil lembar pengesahan skripsi. Jika tanda tangan *digital* sama dengan nilai hasil lembar pengesahan skripsi, maka lembar pengesahan skripsi berhasil diverifikasi, jika nilainya tidak sama maka lembar pengesahan skripsi gagal dikonfirmasi sehingga skripsi tidak dapat diajukan sebagai salah satu berkas pengurusan ijazah.

Metode yang digunakan untuk membuat tanda tangan *digital* adalah metode *Rivest Shamir Adleman* (RSA) karena merupakan algoritma kriptografi yang dapat digunakan untuk menerapkan tanda tangan digital [3],

dimana algoritma ini termasuk kelompok kriptografi kunci asimetris, artinya mempunyai kunci berbeda untuk enkripsi dan dekripsi. RSA memiliki 2 kunci yaitu kunci *public* yang boleh diketahui oleh siapa saja dan kunci *private* yang bersifat rahasia dan hanya diketahui oleh pihak-pihak tertentu saja [4]. Dengan penggunaan kunci yang berbeda untuk enkripsi dan dekripsi keamanan dan keaslian data lebih terjamin karena hanya orang yang mempunyai kunci saja yang bisa membuat dan merubah data tanda tangan *digital* [5].

Dengan menerapkan metode RSA untuk verifikasi tanda tangan *digital* pada lembar pengesahan skripsi, dapat menjadi salah satu solusi untuk mengecek keabsahan lembar pengesahan skripsi mahasiswa, sehingga dapat mencegah tindakan curang mengesahkan lembar pengesahan skripsi oleh pihak yang tidak bertanggung jawab tanpa sepengetahuan dosen yang bersangkutan.

II. Metode

A. Tahapan Pengumpukan Data

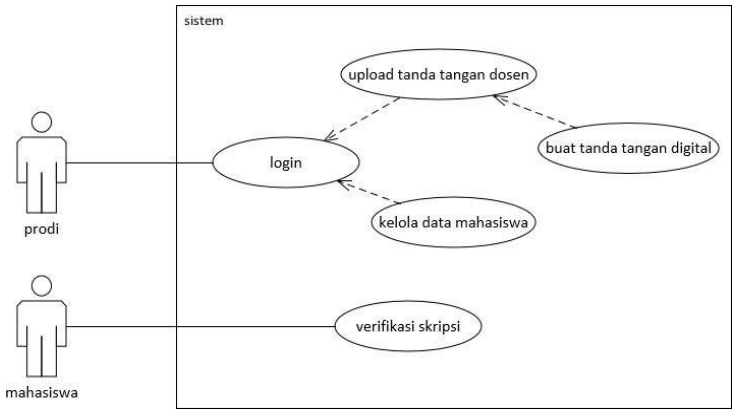
Dalam penelitian ini, metode penelitian yang digunakan adalah penelitian tindakan (*action research*), menurut [6] penelitian tindakan merupakan "penelitian yang berfokus langsung pada tindakan sosial. Penelitian tindakan ini merupakan metode yang didasarkan pada tindakan masyarakat yang seringkali diselenggarakan pada suatu latar yang luas, seperti di rumah sakit, pabrik, sekolah, dan lain sebagainya". Metode penelitian tindakan diduga kuat sesuai dengan permasalahan yang diteliti penulis karena didasarkan pada permasalah penggunaan tanda tangan elektronik yang tidak dapat menjamin keabsahan lembar pengesahan skripsi mahasiswa Fakultas Ilmu Komputer Universitas Muslim Indonesia, danuntuk pembuatan tanda tangan *digital* menggunakan metode RSA

B. Teknik Analisis Data

Metode yang digunakan untuk membuat tanda tangan *digital* adalah metode *Rivest Shamir Adleman* (RSA) karena merupakan algoritma kriptografi yang dapat digunakan untuk menerapkan tanda tangan digital, dimana algoritma ini termasuk kelompok kriptografi kunci asimetris, artinya mempunyai kunci berbeda untuk enkripsi dan dekripsi. RSA memiliki 2 kunci yaitu kunci *public* yang boleh diketahui oleh siapa saja dan kunci *private* yang bersifat rahasia danhanya diketahui oleh pihak-pihak tertentu saja. Dengan penggunaan kunci yang berbeda untuk enkripsi dan dekripsi keamanan dan keaslian data lebih terjamin karena hanya orang yang mempunyai kunci saja yang bisa membuat dan merubah data tanda tangan *digital* [7].

C. Analisis Sistem Usulan

Use Case Diagram



Gambar 1. Use case Diagram

Pada Gambar 1, berikut ini adalah penjelasnya:

a) Agar prodi dapat masuk kefitur utama aplikasi, terlebih dahulu harus *login*

b) Prodi *upload* tanda tangan dosen

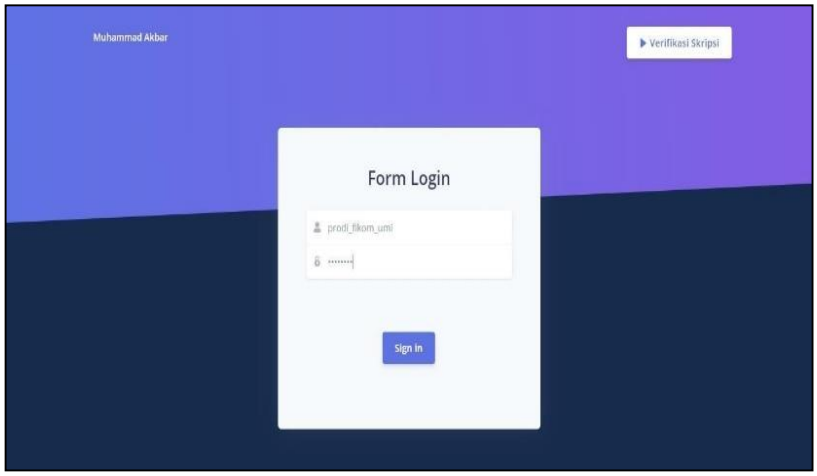
c) Prodi dapat mengelola data mahasiswa yaitu dosen yang bersangkutandengan tugas akhir mahasiswa

d) Prodi buat tanda tangan *digital* dosen

e) Mahasiswa dapat verifikasi dokumen skripsi

III.Hasil dan Pembahasan

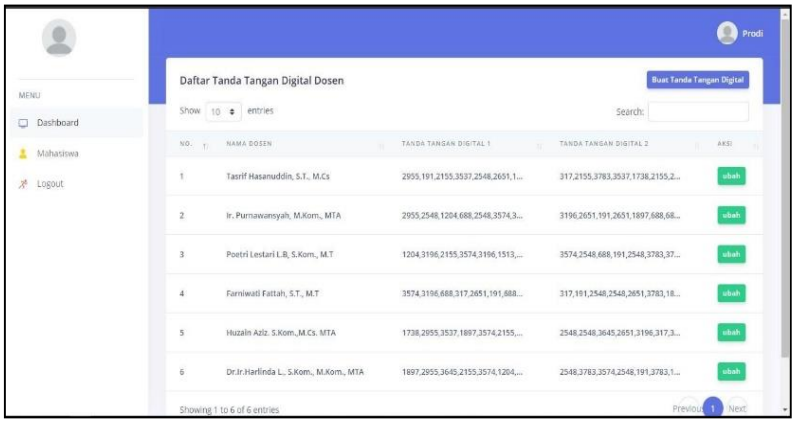
A. Halaman Login



Gambar 2. Halaman login

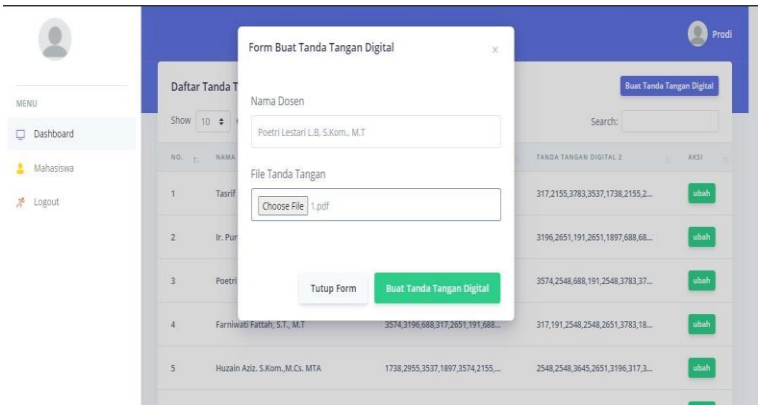
Merupakan halaman *login*, prodi harus memasukkan *username* dan *password* yang benar agar dapat masuk kehalaman utama aplikasi yaitu halaman *Dashboard*, jika prodi telah memasukkan *username* dan *password* yang benar maka akan dialihkan kehalaman *Dashboard*

B. Halaman Dashboard



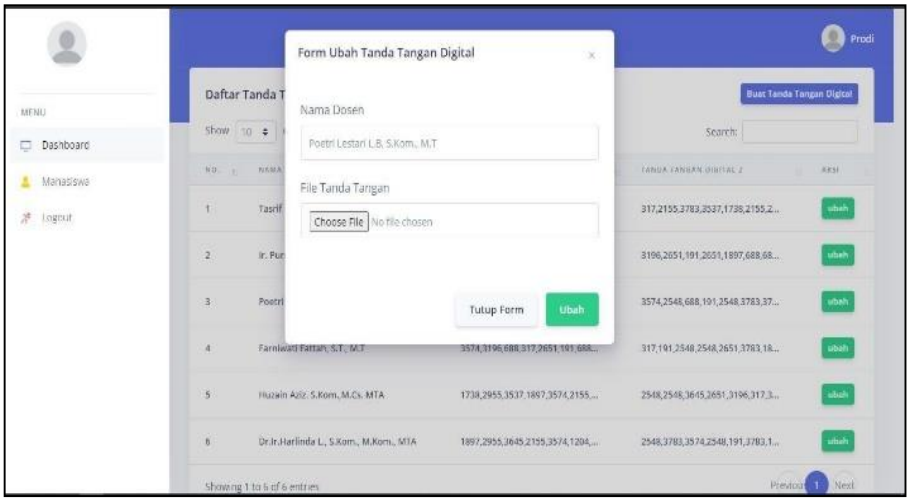
Gambar 3. Halaman *dashboard*

Merupakan halaman *dashboard*, dimana prodi dapat melihat daftar nama-nama dosen yang telah memiliki tanda tangan *digital*. Prodi dapat menambahkan tanda tangan *digital* dosen dengan cara memasukkan nama dosen dan dua gambar tanda tangan dosen yang disatukan dalam satu *file* PDF.



Gambar 4. Form input tanda tangan digital

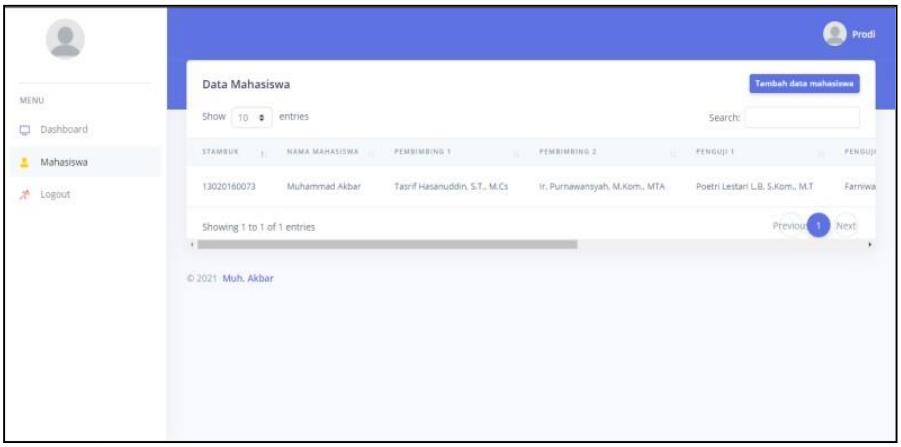
Prodi juga dapat mengubah nama atau tanda tangan *digital* dosen dengan cara memasukkan nama dosen yang baru atau mengunggah file tanda tangan dosenyang baru dalam bentuk *file* PDF.



Gambar 5. Form ubah tanda tangan digital

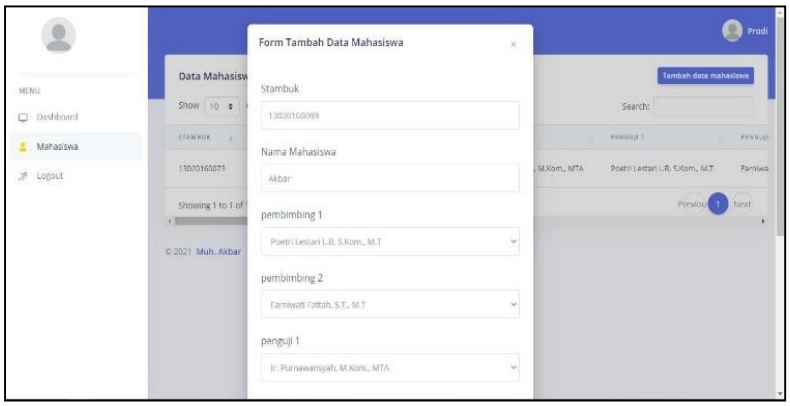
C. Halaman Data Mahasiswa

Menampilkan halaman data mahasiswa dimana prodi dapat melihat stambuk mahasiswa, nama mahasiwa, pembimbing, penguji, ketua sidang, ketua prodi, dan dekan dari mahasiswa.

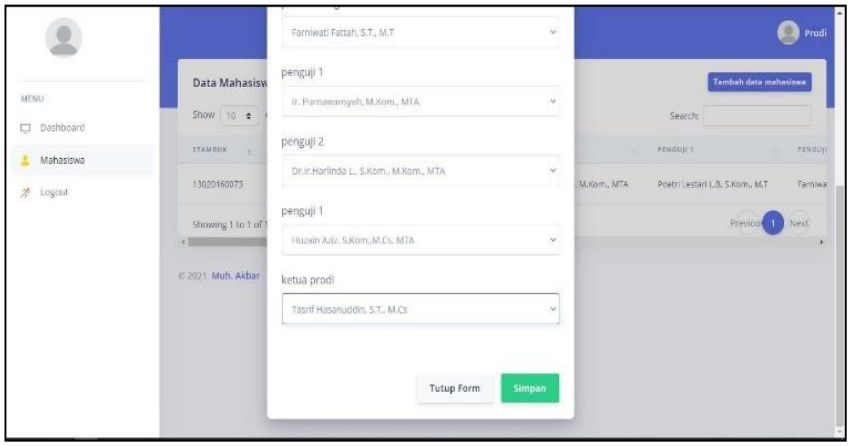


Gambar 6. Halaman data mahasiswa

Prodi juga dapat menambahkan mahasiswa baru yang dapat memverifikasi skripsi dengan cara memasukkan stambuk, nama mahasiswa, nama pembimbing, nama penguji, dan ketua prodi sesuai jenis prodi mahasiswa.

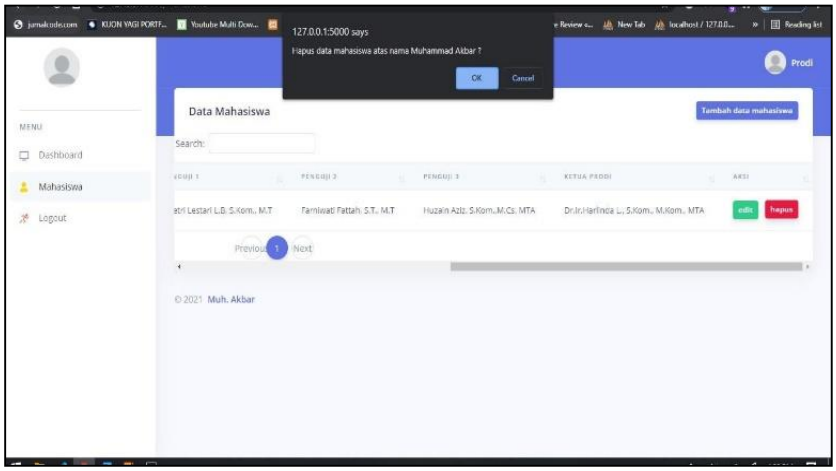


Gambar 7. Form tambah data mahasiswa



Gambar 8. Form tambah data mahasiswa setelah diisi

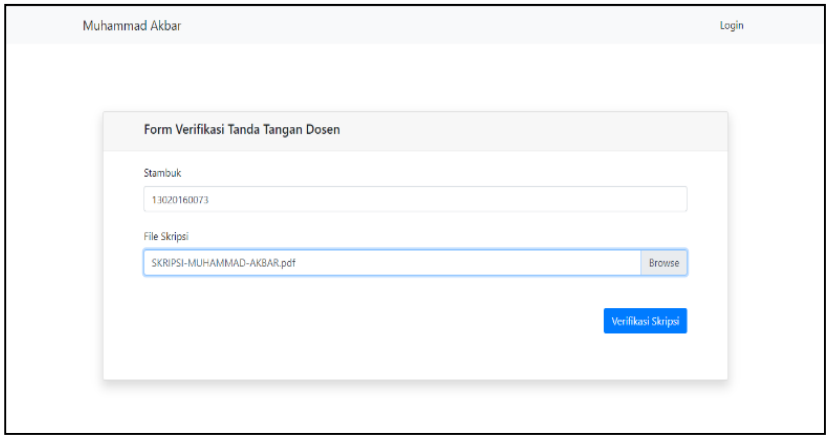
Prodi juga dapat menghapus data mahasiswa dengan menekan tombah hapus, jika tombol ditekan, akan menampilkan pemberitahun untuk konfirmasi ulang hapus data mahasiswa.



Gambar 9. Tampilan peringatan hapus data mahasiswa

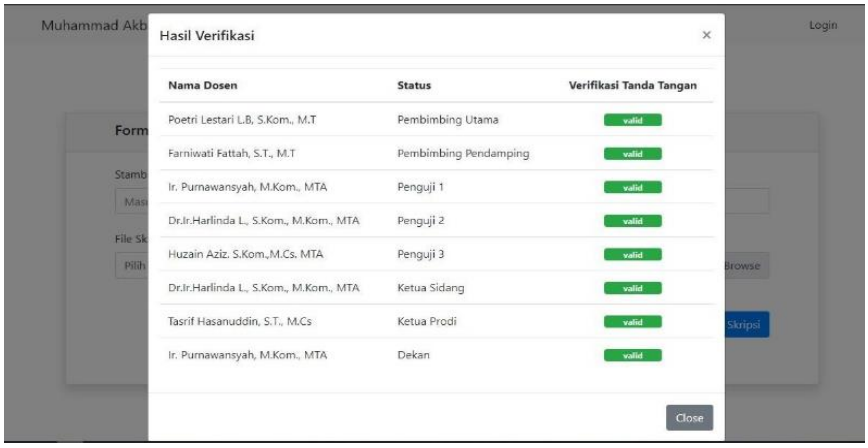
D. Halaman Verifikasi

Merupakan halaman verifikasi skripsi mahasiswa, dimana mahasiswa harus memasukkan stambuk dan *file* skripsi dalam bentuk *file* PDF, kemudian menekan tombol verifkasi untuk melakukan verifikasi skripsi.



Gambar 10. Halaman verifikasi

Hasil verifkasi dapat dilihat pada gambar 4.11 dimana akan muncul nama- nama dosen dari mahasiswa yang bersangkutan dan status verifikasi tanda tangan dosen, skripsi berhasil diverifikasi jika semua tanda tangan dosen *valid*.



Gambar 11. Hasil verifikasi

E. Implementasi Metode RS

```
1   p = 59
2   q = 67
3
4   n = p * q
5   tautient_n = (p-1) * (q-1)
```

Gambar 12. Implementasi metode RS

Menentukan bilangan prima p dan q, nilai n dan $\varphi(n)$ Pada gambar 4.11 baris ke-1 dan baris ke-2 merupakan *source code* untuk menentukan dua bilangan prima, dimana variabel *p* diberi nilai yaitu 59, dan variabel *q* diberi nilai 67. Nilai *p* dan *q* yang dipilih secara acak harus bilangan prima, batas nilai minimum yang digunakan yaitu angka 2 dan tidak ada batas maksimum karena semakin besar nilai *p* dan *q* keamanan data enkripsi RSA semakin bagus, namun pada penelitian ini penulis membatasi nilai yang digunakan pada angka puluhan karena ketidakmampuan perangkat yang digunakan penulis dalam menghitung nilai pangkat pada saat proses enkripsi yang hasilnya ratusan digit angka yang akan menyebabkan hasilnya tidak terdefinisi. Pada baris ke-4 gambar 4.1 variabel *n* berisi nilai hasil perkalianantara variabel *p* dan *q*. Baris ke-5 variabel *tautient_n* yang akan menampung nilai hasil dari:

$$(variabel\ p - 1) \times (variabel\ q - 1) \quad (1)$$

```
37  def generate_e(tautient_n):
38      for e in range(3, tautient_n):
39          if relatif_prima(tautient_n, e):
40              return e
```

Gambar 13. fungsi *generate*_e

Generate_e merupakan fungsi yang digunakan untuk mencari nilai *e*. Parameter yang digunakan yaitu *tautient_n*, untuk proses kerja fungsi yaitu dengan cara melakukan iterasi mulai dari angka 3 dan iterasi akan berhenti jika interasi ke-*n* sama dengan nilai *tautient_n*, jika nilai *e* relatif prima denga *tautient_n* proses iterasi dihentikan dan fungsi *generate_e* akan mengembalikan nilai *e*.

```
42  def generate_d(tautient_n, e):
43      k_value = 1
44      hasil_akhir = 0
45      while True:
46          hasil = (1 + (k_value * tautient_n)) / e
47          if cek_bilangan_bulat(hasil):
48              hasil_akhir = hasil
49              break
50          k_value +=1
51      return int(hasil_akhir)
```

Gambar 14. Fungsi *generate*_d

Merupakan fungsi untuk mencari nilai *d* yang menggunakan parameter *tautient_n*, dan *e*. untuk mendapatkan nilai *d* dilakukan iterasi hingga hasil perhitungan yang didapatkan adalah bilangan bulat.

```
54  def enkripsi(plaintext):
55      # private key (d, n)
56      global tautient_n, n
57      e = generate_e(tautient_n)
58      d = generate_d(tautient_n, e)
59
60      # convert plaintext ke ASCII
61      nilai_ascii = []
62      for karakter in plaintext:
63          nilai_ascii.append(ord(karakter))
64
65      # enkripsi
66      chipertext = []
67      for nilai in nilai_ascii:
68          enkripsi = (nilai**d) % n
69          chipertext.append(enkripsi)
70      return tuple(chipertext)
```

Gambar 15. Fungsi Enkripsi

Fungsi enkripsi menerima parameter *plaintext* yang akan dienkripsi, kemudian didalam fungsi fungsi enkripsi setiap karakter *plaintext* diubah menjadi nilai ASCII, untuk selanjutnya dilakukan enkripsi menggunakan kunci *private* (*d*, *n*) yang akan menghasilkan *chipertext* (nilai hasil enkripsi). Fungsi dekripsi akan mengunakan parameter *chipertext*, selanjutnya setiap nilai *chipertext* didekripsi menggunakan kunci *public* (*e*, *n*). Hasil dekripsi yang masih berupa nilai ASCII diubah menjadi *plaintext*.

F. Pembuatan Tanda Tangan Digital Dosen

```
# ekstrak gambar dari file pdf
nama_gambar = ekstrakGambar(path)
# hapus file
os.remove(path)
# get nilai hash semua gambar
daftar_hash = get_nilai_hash(nama_gambar)
data = []
for nilai_hash in daftar_hash:
    chipertext = enkripsi(nilai_hash)
    data.append(chipertext_to_str(chipertext))

# # insert data
Dosen.insert(nama_dosen, data)
```

Gambar 16. *Source code* enkripsi *file*

Pembuatan tanda tangan *digital*. Proses pembuatan tanda tangan *digital* dimulai dengan mengekstrak gambar dari *file* PDF, kemudian menghapus *file* yang telah unggah karena sudah tidak digunakan, selanjutnya membuat nilai hash dari gambar tanda tangan dosen yang telah diekstrak. Semua nilai hash gambar tanda tangan dosen dienkripsi untuk menghasilkan tanda tangan *digital* dosen yang selanjutnya disimpan kedalam *database*.

G. Proses Verifiiasi Tanda Tangan Dosen

Proses verifikasi tanda tangan dosen berdasarkan gambar 4.10, setelah mahasiswa memasukkan stambuk dan unggah *file* skripsi, kemudian menekan tombol verifikasi untuk melakukan verifikasi skripsi, proses verifikasinya adalah sebagai berikut.

```
# # ekstrak gambar dari file pdf
nama_gambar = ekstrakGambar(path)
daftar_hash = get_nilai_hash(nama_gambar)
# hapus file
os.remove(path)
```

Gambar 17. Proses verifikasi

Tanda tanga dosen diekstrak dari *file* skripsi yang telah diunggah oleh mahasiswa. Semua gambar yang telah diekstrak dibuatkan nilai *hash* kemudian hapus *file* skripsi yang telah diunggah karena sudah tidak digunakan lagi.

```
    # ekstrak gambar dari file pdf
    nama_gambar = ekstrakGambar(path)
    # hapus file
    os.remove(path)
    # get nilai hash semua gambar
    daftar_hash = get_nilai_hash(nama_gambar)
    data = []
    for nilai_hash in daftar_hash:
        chipertext = enkripsi(nilai_hash)
        data.append(chipertext_to_str(chipertext))

# # insert data
Dosen.insert(nama_dosen, data)
```

Gambar 18. Proses hash

Setelah proses ekstrak gambar dan pembuatan nilai *hash*, pada gambar 4.8 yaitu proses mengambil tanda tangan *digital* dosen dari *database* berdasarkan stambuk yang telah dimasukkan oleh mahasiswa, selanjutnya semua tanda tangan *digital* dosen didekripsi untuk mendapakan nilai *hash*.

```
hasil_verifikasi = list()
jabatan_dosen = ['Pembimbing Utama', 'Pembimbing Pendamping', 'Penguji 1',
                 'Penguji 2', 'Penguji 3', 'Ketua Sidang', 'Ketua Prodi', 'Dekan']
for dosen, jabatan in zip(hash_dosen, jabatan_dosen):
    for mhs in daftar_hash:
        if dosen['hash_1'] == mhs or dosen['hash_2'] == mhs:
            hasil_verifikasi.append({
                'nama_dosen':dosen['nama_dosen'],
                'jabatan': jabatan,
                'status': 'valid'
            })
            break
    else:
        hasil_verifikasi.append({
            'nama_dosen':dosen['nama_dosen'],
            'jabatan': jabatan,
            'status': 'tidak_valid'
        })
```

Gambar 19. Proses *matching*

Setelah proses dekripsi tanda tangan *digital* dosen, pada gambar 19 proses verifikasi dengan mencocokkan setiap nilai *hash* gambar tanda tangan yang telah diekstrak dari skripsi dengan nilai *hash* tanda tangan *digital* dosen yang diambil dari *database*. jika salah satu nilai *hash* gambar tanda tangan pada skripsi memiliki kecocokan dengan nilai hash tanda tangan *digital* dosen, maka tanda tangan pada skrispi dianggap *valid*, selain itu maka tanda tangan dianggap tidak *valid.*

IV. Kesimpulan dan saran

Berdasarkan hasil penelitian yang telah dilakukan, dapat ditarik kesimpulan bahwa: Telah merancang aplikasi tanda tangan *digital* yang dapat memverifikasi tanda tangan dosen yang telah diekstrak menggunakan metode RSA, pada lembar pengesahan skripsi mahasiswa berekstensi PDF. Proses verifikasi sistem dengan cara membandingkan nilai hash antara tanda tangan *digital* yang tersimpan dalam *database* dengan tanda tangan *digital* yang termuat pada *file* lembar pengesahan skripsi mahasiswa. Adapun saran untuk pengembangan penelitian kedepannya adalah sebagai berikut: Menggunakan pendekatan yang berbeda saat menerapkan tanda tangan *digital* seperti menggunakan teknik stenografi untuk menyisipkan pesan rahasia yang

mewakili tanda tangan dosen kedalam *file* skripsi mahasiswa. Menambahkan fitur tanda tangan secara langsung didalam aplikasi tanpa harus memasukkan gambar kedalam *file* PDF.

Daftar Pustaka

[1] T. Yuniati and M. F. Sidiq, "Literature Review: Legalisasi Dokumen ElektronikMenggunakan Tanda Tangan Digital sebagai Alternatif Pengesahan Dokumen di Masa Pandemi," *RESTI,* vol. IV, no. 6, p. 1058 –1069, 2020.

[2] M. U. Noor, "Tanda Tangan Digital: Otoritas pada Arsip Elektronik," *JIPI*, vol. 6, no. 1, pp. 17-26, 2021.

[3] O. K. Sulaiman, M. Ihwani and S. F. Rizki, "Model Keamanan Informasi Berbasis Tanda Tangan Digital Dengan Data Encryption Standard (Des) Algorithm," *InfoTekJar (Jurnal Nasional Informatika dan Teknologi Jaringan)*, vol. 1, no. 14-19, p. 1, 2016.

[4] Y. Anshori, A. E. Dodu, D. M. P and W. , "Implementasi Algoritma Kriptografi Rivest Shamir Adleman (RSA) pada Tanda Tangan Digital," *Techno.Com*, vol. 18, no. 2, pp. 110-121, 2019.

[5] F. Nuraeni, Y. H. Agustin and I. M. Muharam, "Implementasi Tanda Tangan Digital Menggunakan RSA dan SHA-512 Pada Proses Legalisasi Ijazah," *Konferensi Nasional Sistem Informasi (KNSI)*, pp. 864-869, 2018.

[6] Z. A. Hasibuan, "Metodologi Penelitian Pada Bidang Ilmu Komputer Dan Teknologi Informasi," *Metodologi Penelitian Pada Bidang Ilmu Komputer Dan Teknologi Informasi*, vol. 4, no. 1, p. 126–130, 2007.

[7] F. Nuraeni, Y. H. Agustin, D. Kurniadi and I. D. Ariyanti, "Implementasi Skema QR-Code dan Digital Signature menggunakan Kombinasi Algoritma RSA dan AES untuk Pengamanan Data Sertifikat Elektronik," *Seminar Nasional Teknologi Informasi, Komunikasi dan Industri (SNTIKI)*, vol. 12, pp. 43-52, 2020.