Penerapan *Elliptic Curve Digital Signature Algorithm* pada Tanda Tangan Digital dengan Studi Kasus Dokumen Surat – Menyurat

Aulia Nadzifarin[1], Asmunin[2]
[1,2]Jurusan Teknik Informatika/Teknik Informatika, Universitas Negeri Surabaya
[1]aulianadzifarin.18005@mhs.unesa.ac.id
[2]asmunin@unesa.ac.id

***Abstrak*— Perubahan pola kehidupan manusia melalui media digital elektronik menjadi dampak dari perkembangan teknologi yang sangat pesat, salah satunya pada perubahan pola dari *paper* ke *paperless.* Bersamaan dengan masa pandemi covid-19 yang berpengaruh pada kebijakan pada sebagian besar bidang ketenagakerjaan dan pendidikan dilakukan via *online*, sehingga dokumen elektronik sering digunakan dalam setiap proses bidang tersebut. Seperti dalam hal administrasi surat – menyurat ataupun transaksi pertukaran dokumen yang lain. Dan keaslian dokumen sangat diperlukan dengan menggunakan mekanisme salah satunya tanda tangan digital. Tanda tangan digital dapat membantu dalam proses menjaga kerahasiaan, autentikasi, penyangkalan, intergritas dan keabsahan bahwa tanda tangan dalam suatu dokumen yang telah dikirim dan diterima. Dengan memanfaatkan algoritma tanda tangan digital *Elliptic Curve Digital Signature Algorithm* (ECDSA) untuk proses enkripsi dan dekripsi dalam meningkatkan keamanan dan autentikasi data. Representasi akhir dari tanda tangan digital diterapkan pada dokumen surat menyurat dengan tipe PDF, dengan menggunakan bahasa pemrograman phyton ECDSA sebagai proses jalannya sistem.**

***Kata Kunci*— Tanda tangan digital, ECDSA, Enkripsi, Dekripsi, Dokumen elektronik**

I. PENDAHULUAN

Perkembangan teknologi yang sangat signifikan, mempengaruhi pola kehidupan manusia melalui media digital elektronik, dengan hubungan perubahan pola dari *paper* ke *paperless*[1]. Bersamaan dengan masa pandemi covid-19 sejak bulan maret 2020 yang mengancam keselamatan jiwa banyak orang di Indonesia, pemerintah membuat kebijakan pada sebagian besar bidang ketenagakerjaan dan pendidikan dikakukan melalui via *online* untuk mencegah penyebaran virus covid-19. Dengan kebijakan tersebut, dokumen elektronik sering digunakan dalam proses ketenagakerjaan dan pendidikan, seperti dalam hal administrasi surat – menyurat ataupun transaksi pertukaran dokumen yang lain. Sehingga keaslian dokumen sangat diperlukan dengan menggunakan suatu mekanisme diantaranya menggunakan tanda tangan digital. Karena dengan tanda tangan digital dapat membantu suatu dokumen yang ditransmisikan melalui jaringan internet tidak terjadi modifikasi atau manipulasi dalam proses pengiriman, serta bagi orang yang awam akan teknologi akan sulit untuk membuktikan dan membedakan keaslian dokumen digital tersebut [2]. Dan dapat dengan mudahnya pengirim menyangkal terhadap isi suatu dokumen, bahwa dia yang sudah menulis ataupun mengirim dokumen tersebut.

Tanda tangan digital merupakan mekanisme kriptografi yang digunakan untuk memverifikasi keaslian dan integritas data digital. Mereka mengizinkan penandatangan yang telah membuat kunci verifikasi publik untuk menandatangani pesan sedemikian rupa sehingga pihak lain mana pun dapat memverifikasi bahwa pesan tersebut berasal dari penandatangan dan tidak dimodifikasi dengan cara apapun. Primitif ini sangat penting untuk membangun sistem yang aman dan digunakan di sebagian besar protokol keamanan dunia nyata. Namun, untuk sebagian besar skema tanda tangan, banyak perangkat komputasi dengan daya komputasi terbatas tidak dapat melakukan verifikasi tanda tangan yang memakan waktu dan daya.

Sebuah algoritma tanda tangan digital secara sistematis mengacu pada standar untuk tanda tangan digital, yang didasarkan pada sifat aljabar masalah logaritma diskrit dan eksponensial modular berdasarkan prinsip kriptosistem kunci publik. Tanda tangan digital bekerja berdasarkan prinsip dari dua kunci kriptografi yang saling mengautentikasi, yaitu, *public key* dan *privat key* [3], dengan sistem *public key* digunakan untuk proses enkripsi dan *privat key* digunakan untuk deskripsi[4]. akan tetapi dalam proses *digital signature*, *privat key* digunakan sebagai proses enkripsi dokumen dan *public key* digunakan sebagai dekripsi. Adapun salah satu teknik dari kriptografi kunci publik yang digunakan adalah kriptografi kurva eliptik yang berupa Elliptic Curve Digital Signature Algorithm (ECDSA), algoritma kurva eliptik dikenalkan secara tersendiri oleh Victor Miller pada tahun 1985 dan dijelaskan berdasarkan struktur aljabar kurva elliptik di atas bidang terbatas.

Dengan algoritma *Elliptic Curve Digital Signature Algoritthm* (ECDSA) yang merupakan suatu algoritma

kriptografi kunci publik yang digunakan untuk proses tanda tangan digital dengan menggunakan kurva eliptik sebagai dasar perhitungan. Adapun proses yang dilakukan pada algoritma ini adalah key generator, signing dan verifying[5]. karena ukuran kuncinya yang kecil, dengan asumsi kesulitan yang lebih kuat dari suatu masalah pada kurva eliptik, ECDSA memiliki algoritma penandatanganan cepat dan oleh karena itu digunakan di banyak perpustakaan kriptografi standar dan dalam mata uang kripto[6]. Adapun keunggulan dari proses enkripsi tanda tangan digital merupakan fitur keamanan yang digunakan untuk proses kerahasiaan, autentikasi, non-repudensi, integritas dan memastikan bahwa tanda tangan dalam suatu dokumen yang dikiriman sah. Berdasarkan permasalahan serta kajian dari beberapa penelitian tersebut, pada artikel ini akan mengusung judul Penerapan Pada Tanda Tangan Digital Dengan Studi Kasus Dokumen Surat - Menyurat sehingga keluaran yang dihasilkan berupa keaslian tanda tangan digital dan keamanan suatu dokumen dapat terjaga dengan baik.

II. METODE PENELITIAN

Peneliti menggunakan jenis metode penelitian kuantitatif, sesuai dengan tujuan yang ingin didapat untuk membuktikan serta menguji teori yang digunakan dalam penelitian terhadap implementasi tanda tangan digital dalam dokumen surat menyurat. Adapun tahapan penelitian adalah sebagai berikut:

A. Identifikasi Masalah

Pada tahapan identifikasi masalah berfungsi untuk mengidentifikasi masalah yang ada pada ruang lingkup penelitian. Dengan latar belakang masalah bahwasanya untuk menjaga kemamanan suatu dokumen dari proses modifikasi, manipulasi dan anti-penyangkalan, maka diperlukan teknologi untuk mendeteksi hal tersebut untuk memudahkan pengirim ataupun penerima dalam membuktikan keaslian dokumen.

B. Studi literatur

Tahap studi literatur merupakan tahapan yang digunakan untuk mencari dan menemukan suatu referensi baik dari sumber jurnal ilmiah, buku, internet, laporan praktikum dan sumber lainnya sebagai bahan pendukung penelitian.

1. Elliptic Curve Digital Signature Algorithm (ECDSA)

Untuk membangun sebuah sistem kriptografi kurva eliptik ada tiga hal yang harus diperhatikan[7], yaitu :

a. Pemilihan bidang terbatas, yaitu Fq dengan representasi elemen dari Fq, implementasi yang dipilih : Fp dengan representasi elemen berupa bilangan bulat yang sangat besar.

Dengan dapat dijelaskan bahwa bidang terbatas sering disebut juga dengan Galois Field (GF) yang merupakan bidang yang hanya dimiliki oleh elemen bilangan terbatas. Jika q merupakan pangkat prima, maka akan ada satu bidang terbatas dengan derajat q. Dengan simbol Fq atau GF(q) yang di representasikan seperti, jika $q=p^m$, p adalah bilangan prima dan m adalah bilangan integer positif.

Bidang terbatas yang biasa digunakan oleh kriptografi adalah q=p, dengan p merupakan bilangan prima (Fp) dan q=2m, dimana m integer lebih besar dari satu dengan simbol F2m.

b. Pemilihan kurva eliptik E pada Fq, karena tidak semua kurva eliptik aman digunakan, dengan syarat:
   1) Jumlah nilai titik kurva E, adalah E(Fq), harus bisa dibagi dengan sebuah bilangan prima n yang cukup besar
   2) Kurva sedemikian rupa E(Fq) $\neq$ q
   3) *n* tidak membagi q^k-1 untuk semua 1$\leqslant$ k $\leqslant$20
c. Penentuan kurva eliptik, implementasi yang dipilih oleh peneliti adalah protokol ECDSA.

Algoritma ECDSA merupakan turunan dari tanda tangan elgamal dengan perubahan dan pengoptimalan menangani masalah representasi elemen grup titik kurva eliptik seperti penggunakan ECDSA dengan Kurva secp521r1, proses tanda tangan dan verifikasi bekerja seperti pada bidang terbatas Fq dengan $q=pn$ adalah pangkat utama $p\neq2,3$ dan kurver eliptik E/Fq adalah kurver nonsingular yang memenuhi persamaan $y2=x3+ax+b$. Himpunan titik pada E terletak pada Fq ditambah titik tak terhingga berubah menjadi suatu kelompok, dilambangkan $E(Fq)$.

2. Fungsi *Hash* Blake2b

Hash merupakan hasil enkripsi dari sebuah informasi yang dianggap penting, yang berupa susunan huruf ataupun angka acak. Adapun salah satu diantara macam – macam fungsi hash adalah fungsi hash Blake2b.

Algoritma blake2b adalah salah satu algoritma fungsi hash kriptografi optimasi dari bagian algoritma Blake2. Algoritma blake2 terdiri dari dua jenis, yaitu Blake2b dan Blake2s[8]. Adapun perbedaan dari keduanya adalah pada penggunaan platform, blake2b

bekerja pada platform sistem 64 bit, sedangkan blake2s bekerja pada platform sistem 32 bit.

Algoritma blake2b dapat menghasilkan keluaran atau digest size dengan ukuran 160 bit, 256 bit, 384 bit, dan 512 bit. Untuk tahapan yang digunakan dalam algoritma Blake2b dapat dijelaskan dalam fungsi yang berbeda, pertama inisialisasi konstruktur Blake2b dengan masukan parameter buffer, state internal dari nilai hash, total bit, pointer dan ukuran digest. Fungsi aritmatis dan matematis yang digunakan pada algoritma Blake2b terdapat di fungsi F, yang digunakan untuk proses kompresi dan perulangan mutasi pesan dengan fungsi G. dengan memanfaatkan konstan SIGMA yang menghasilkan message digest dari proses komputasi algoritma Blake2b[9].

3. Analisis

Analisis adalah tahap upaya untuk menentukan pola yang berkaitan dengan pengujian penelitian secara sistematis secara keseluruhan. Adapaun tahap analisis yang dibutuhkan adalah sebagai berikut :

a. Analisis Kebutuhan

Penelitian ini menggunakan metode *Elliptic Curve Digital Signature Algorithm* (ECDSA) dalam proses enkripsi dan dekripsi sebuah dokumen, sehingga penelitian ini membutuhkan sistem pendukung yang digunakan sebagai berikut:

1) Perangkat keras (*hardware*)

Perangkat keras yang digunakan pada penelitian ini dengan spesifikasi :

a) *Processor* : Intel(R) Core(TM) i3-7020U @ 2.30GHz
b) RAM : 16384 MB
c) *System type* : 64 bit

2) Perangkat lunak (*software*)

Perangkat lunak yang digunakan yaitu:

a) Sistem operasi : Microsoft Windows 10 Pro
b) Bahasa pemrograman : phyton

b. Analisis Sistem

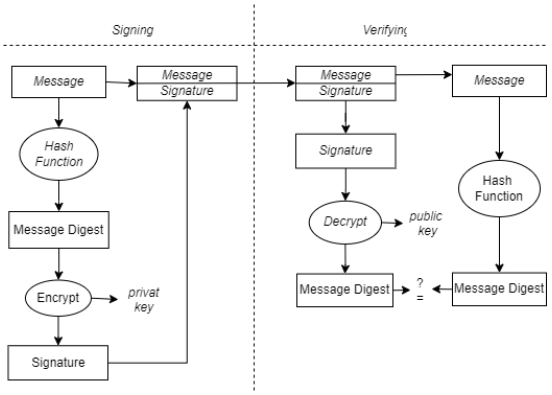
Peneliti mengimplementasikan algoritma ECDSA untuk proses enkripsi dan dekripsi dokumen melalui tanda tangan digital, dengan menggunakan fungsi hash blake2b untuk proses tranformasi *message digest* yang dihasilkan.

Penelitian ini dilakukan dengan proses enkripsi pada sebuah dokumen, dari *plaintext* kebentuk *chipertext* begitupun sebaliknya pada proses dekripsi.

c. Analisis kebutuhan proses

Pada tahap analisa kebutuhan proses akan dijelaskan mengenai proses enkripsi dan dekripsi dokumen dengan menggunakan algoritma tanda tangan digital *Elliptic Curve Digital Signature Algorithm.*

Adapun skema umum alur dari penggunaan fungsi *hash* dan algoritma kriptografi ECDSA yang digunakan adalah sebagai berikut :



Gbr. 1 Skema Umum Alur Tanda Tangan Digital

Pada Gbr. 1, Pada proses enskripsi atau *sign* dalam tanda tangan digital bekerja sebagaimana berikut:

1) Pengirim melakukan proses transformasi pesan yang akan dikirim menjadi *message digest*, menggunakan fungsi hash kriptografi Blake2b
2) Kemudian Message digest di enkripsi dengan menggunakan *privat key* pada algroritma ECDSA, hasil dari enkripsi inilah yang disebut dengan *signature (s)*.
3) Pengirim mengirimkan pesan M bersamaan dengan hasil signature, dengan kesimpulan bahwa pesan M sudah memiliki tanda tangan digital dari pengirim.

Sedangkan, pada algoritma untuk verifikasi ataupun proses dekripsi ECDSA berfungsi sebagai berikut:

1) Penerima memisahkan pesan M dengan tanda tangan yang dihasilkan.
2) Penerima melakukan proses dekripsi *signature* dengan menggunakan kunci publik dari pengirim yang menghasilkan *message digest* semula.

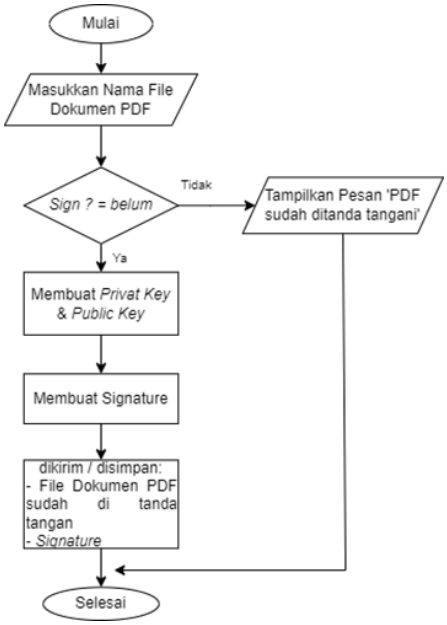
3) Menghitung *hash* pesan dengan fungsi hash kriptografi yang sama yang digunakan selama penandatanganan oleh pengirim.
4) Jika hasil dari dekripsi *signature* dan nilai *hash* pesan M menunjukkan nilai yang benar, apabila MD = MD' maka data yang dikirim valid.

4. Desain sistem

Pada tahap ini akan dilakukan perancangan desain sistem yang akan berjalan dengan tujuan untuk memudahkan implementasi dan proses penyederhaan pada masalah. Berikut Gbr. 2, yang merupakan desain sistem tampilan awal jalannya proses enkripsi dan dekripsi dokumen PDF.

Gbr. 2 Desain Sistem Tampilan Awal

Pada bagian alur *sign*, yang akan dilakukan yaitu membuat *privat key* dan *public key*, memilih dan mengunggah file dokumen yang di inginkan, sehingga program akan melakukan proses pembuatan tanda tangan digital yang nantinya dikirim secara bersamaan dengan file dokumen surat – menyurat tersebut. Akan tetapi file yang sudah ditanda tangani tidak dapat ditanda tangani lagi, dengan diberikan pemberitahuan bahwasanya file sudah di tanda tangani. Sebagaimana proses yang ditunjukkan oleh diagram alur pada Gbr. 3.



Gbr. 1 Diagram Alur Proses *Sign*.

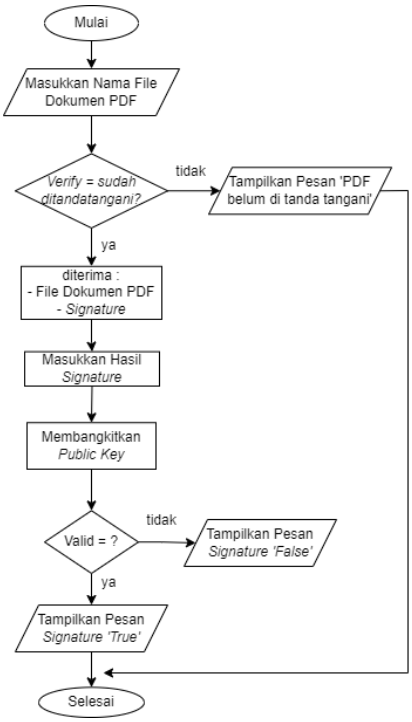
Adapun desain sistem tampilan proses *sign* atau enkripsi sebagai berikut :

Gbr. 4 Desain Sistem *Input* File PDF

Gbr. 5 Desain Sistem Tampilan Proses Enkripsi / *Sign*

Gbr. 6 Desain Sistem Tampilan Sudah di Tanda Tangan

Dan untuk alur *Verify* dalam proses verifikasi tanda tangan digital nantinya akan menghasilkan tanda tangan digital *valid* ataupun *invalid* dengan keluaran *true or false*. Dan ketika dilakukan proses *verify* pada dokumen PDF yang belum ditandatangani maka akan terdapat pemberitahuan bahwasanya file tersebut belum ditanda tangani. Seperti diagram alur yang tertera pada Gbr. 7.



Gbr. 7 Diagram Alur Proses *Verify.*

Yang ditunjukkan pada desain sistem tampilan berikut:

Gbr. 8 Desain Sistem Tampilan Proses Dekripsi / *Verify.*

Gbr. 9 Desain Sistem Tampilan Sudah di Tanda Tangan.

5. Implementasi

Pada tahap implementasi akan dilakukan penerapan algoritma ecdsa denga kurva NIST521p dengan pembangkit kunci *hash* Blake2b terhadap suatu sistem yang akan dijalankan dengan menggukan bahasa pemrograman python, yang berupa sistem aplikasi dekstop.

Adapun implementasi algoritma ECDSA dengan panjang modulus 521 bit dan *hash* Blake2b pada proses enkripsi dengan menggunakan program python, sebagaimana berikut Gbr. 10.

```
def encrypt(private_key, text):
    sk     =
ecdsa.SigningKey.from_string(bytes.fromhex(p
rivate_key), curve=ecdsa.NIST521p,
hashfunc=blake2b)
    enkrip = sk.sign(text.encode('utf-8'))
    return enkrip.hex()
```

Gbr. 10 Potongan Kode Program Proses Enkripsi.

Sedangkan untuk potongan kode program pada proses dekripsi tertera pada Gbr. 11.

```
def decrypt(public_key, signature, text):
    try:
        vk =
ecdsa.VerifyingKey.from_string(bytes.fromhex
(public_key), curve=ecdsa.NIST521p,
hashfunc=blake2b)
        return
vk.verify(bytes.fromhex(signature),
text.encode('utf-8'), blake2b)
    except:
        return False
```

Gbr. 11 Potongan Kode Program Proses Dekripsi.

6. Pengujian

Pada tahap ini akan dilakukan pengujian dari sistem yang telah dibuat. Yang meliputi cara pengujian penerapan algoritma ECDSA pada proses enkripsi dan dekripsi, pengujian pada proses dekripsi dengan andanya modifikasi dari suatu pesan dokumen, serta pengujian terhadap waktu proses enkripsi dan dekripsi dengan berbagai ukuran file masukan. Pengujian tersebut dilakukan untuk melihat pengaruh algoritma ECDSA dengan fungsi hash Blake2b yang digunakan pada suatu pesan dokumen, dengan tingkat tujuan kerahasiaan (*confendiality*), integritas data (*data integrity*), autentikasi (*authentication*), dan anti-penyangkalan (*non-repudiation*).

III. HASIL DAN PEMBAHASAN

Pada penelitian ini menghasilkan enkripsi dan dekripsi dengan menggunakan metode *Elleptic Curve Digital Signature Algorithm* (ECDSA), yaitu :

A. Penerapan Algoritma *Elleptic Curve Digital Signature Algorithm* (ECDSA) pada dokumen surat menyurat

Berdasarkan Gbr. 10, yang merupakan potongan program dari proses enkripsi atau proses *sign* dapat dijelaskan bahwasanya, untuk melakukan proses enkripsi atau penandatanganan dapat dijabarkan dalam sebuah proses algoritma sebagai berikut:

1. Hitung *hash* pesan, menggunakan fungsi *hash* kriptografi
2. Hasilkan angka acak *k* dalam rentang [1.. *n-1*], Dalam kasus deterministik-ECDSA, nilai *k* adalah HMAC-berasal dari *h* + *privKey*
3. Hitung titik acak $R = k * G$ dan ambil koordinat x: $r = R.x$
4. Hitung bukti tanda tangan: $s = k^{-1} * (h + r * privKey) \pmod{n}$. Kebalikan modular $k^{-1} \pmod{n}$ adalah bilangan bulat, sedemikian rupa sehingga $k * k^{-1} \equiv 1 \pmod{n}$
5. Kembalikan tanda tangan $\{r, s\}$.

Yang ditunjukkan oleh potongan program algoritma ECDSA pada Gbr. 12.

```
def sign(self, hash, random_k):
G = self.public_key.generator
n = G.order()
```

```
k = random_k % n

ks = k + n
kt = ks + n
if bit_length(ks) == bit_length(n):
    p1 = kt * G
else:
    p1 = ks * G
r = p1.x() % n
if r == 0:
    raise RSZeroError("amazingly unlucky
random number r")
s = (
    numbertheory.inverse_mod(k, n)
    * (hash + (self.secret_multiplier *
r) % n)
) % n
if s == 0:
    raise RSZeroError("amazingly unlucky
random number s")
return Signature(r, s)
```

Gbr. 12 Potongan Program Proses *Sign* ECDSA.

Sedangkan pada bagian proses dekripsi atau *verify* dari algoritma ECDSA dapat dijabarkan sebagai berikut:

1. Hitung *hash* pesan, dengan fungsi *hash* kriptografi yang sama yang digunakan selama penandatanganan: h = hash(*msg*)
2. Hitung kebalikan modular dari bukti tanda tangan: $s1 = s^{-1} \pmod n$
3. Pulihkan titik acak yang digunakan selama penandatanganan: $R' = (h * s1) * G + (r * s1) * pubKey$
4. Ambil dari R' koordinat x-nya: $r' = R'.x$
5. Hitung hasil validasi tanda tangan dengan membandingkan apakah $r' == r$

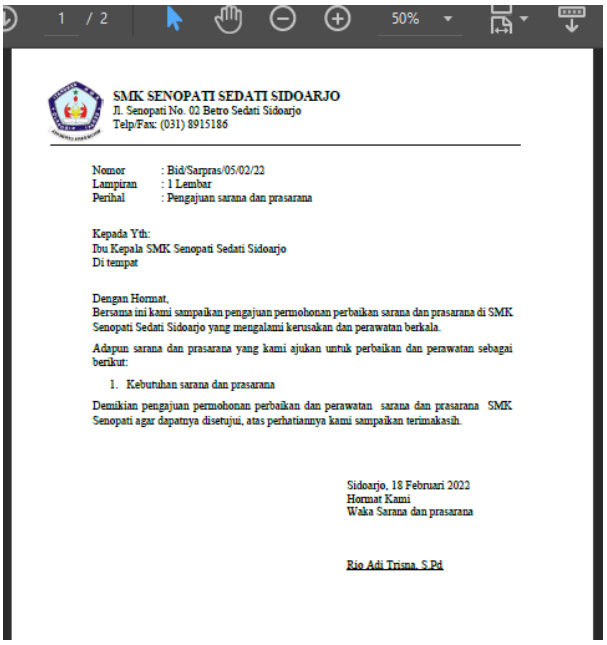
Yang ditunjukkan oleh potongan program pada Gbr. 13.

```
def verifies(self, hash, signature):
G = self.generator
n = G.order()
r = signature.r
s = signature.s
if r < 1 or r > n - 1:
    return False
if s < 1 or s > n - 1:
    return False
c = numbertheory.inverse_mod(s, n)
u1 = (hash * c) % n
u2 = (r * c) % n
if hasattr(G, "mul_add"):
    xy = G.mul_add(u1, self.point, u2)
else:
    xy = u1 * G + u2 * self.point
v = xy.x() % n
return v == r
```

Gbr. 13 Potongan Program Proses *Verify* ECDSA.

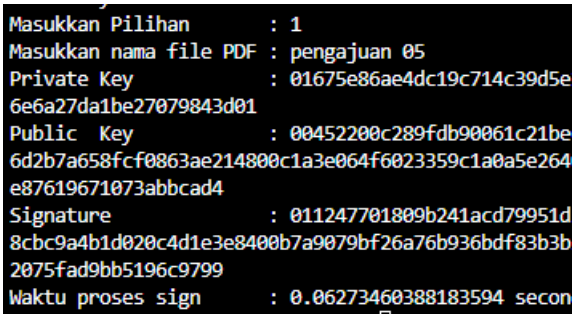
B. Pengujian keaslian dokumen dengan menggunakan tanda tangan digital

Untuk melakukan pengujian ini, dokumen PDF dapat digambarkan sebagai berikut:



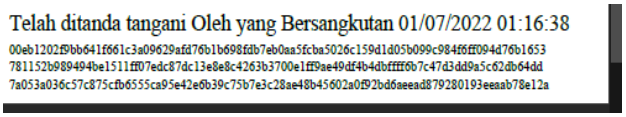
Gbr. 14 Contoh Dokumen PDF Surat – Menyurat Asli.

Pada Gbr. 14 dilakukan proses *sign*, dengan hasil sebagai berikut:



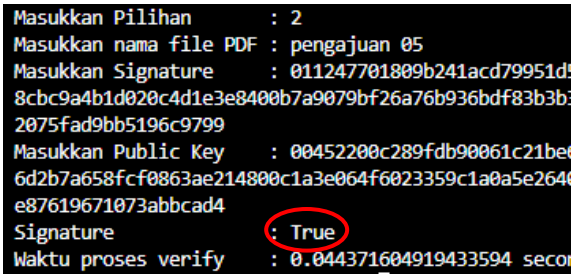
Gbr. 15 Hasil Proses *Sign* Dokumen PDF.

Dokumen file PDF yang sudah di tanda tangani akan memiliki tanda bahwasanya dokumen tersebut sudah ditanda tangani, yang terletak pada *footer* kanan dari dokumen yang berisi penandatanganan oleh yang bersangkutan dengan keterangan tanggal, waktu dan hasil *signature*,seperti yang ditunjukkan oleh Gbr. 16.



Gbr. 16 Tanda untuk Dokumen yang Sudah ditandatangan.

Dengan melakukan proses *verify* terhadap dokumen PDF yang telah ditanda tangani, adapun hasil *verify* sebagai berikut:



Gbr. 17 Hasil Proses *Verify* Dokumen PDF.

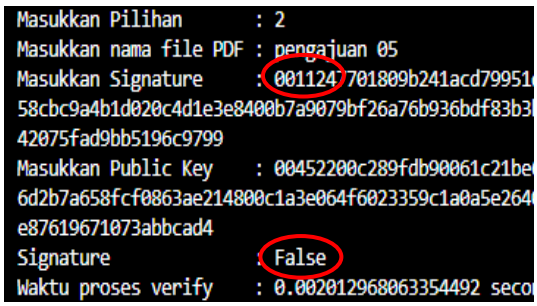
Pada proses pengujian keaslian dokumen ini akan diuji dengan melakukan beberapa percobaan, diantaranya:

1. Uji dengan memasukkan *signature* yang tidak sesuai

Berdasarkan Gbr. 15, file yang dilakukan proses *sign* menghasilkan *signature*:

Signature :
011247701809b241acd79951d5b9326f4e1de468cdc7c4b46a
a3f0384172b69c601ae55751b31ba35d589b616956a60b5d10
20e94ccdb58cbc9a4b1d020c4d1e3e8400b7a9079bf26a76b9
36bdf83b3b3c261979bc15e9799feb5d774a644d591693e8e0
77e9725b30a758559be161abe0fd1b691d40aeef20ad42075fa
d9bb5196c9799

Dengan uji coba menambahkan angka '0' pada hasil *signature*, maka hasil dari dokumen tersebut akan menunjukkan nilai '*false*'.



Gbr. 18 Hasil Proses *Verify* Dokumen PDF.

2. Uji dengan memasukkan *public key* yang tidak sesuai

Berdasarkan Gbr. 15, file yang dilakukan proses *sign* menghasilkan *public key*:

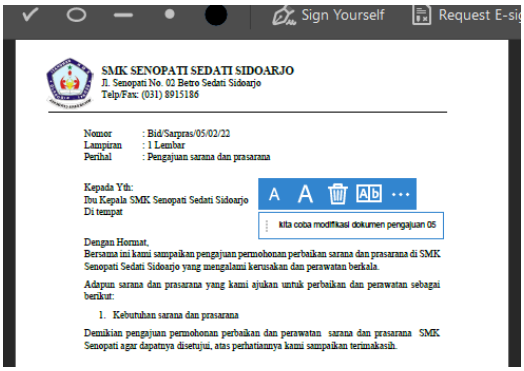
Public Key:
00452200c289fdb90061c21be62e906da557a67d802d51a52d
75a9eac95175db594f2189a327d22001e609ad8b7fa79ef4403
89764ec5d6d2b7a658fcf0863ae214800c1a3e064f6023359c1
a0a5e2640fbcca1a19a1200de8305ddb6d9709ead0d2f90c864
b6ca9d9d5c5ffb1945ac53f86588d587167fe2b48ce87619671
073abbcad43

Dengan uji coba menambahkan angka '10' pada hasil *public key*, maka hasil dari dokumen tersebut akan menunjukkan nilai '*false*'.

Gbr. 19 Hasil Proses *Verify* dengan *Public Key* yang Tidak Sesuai.

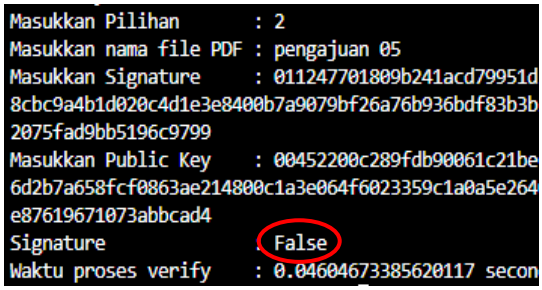
3. Uji dengan memodifikasi dokumen PDF yang sudah diberikan tanda tangan

Berdasarkan gambar 3.3, yang merupakan gambar asli dokumen PDF yang belum diedit, akan diuji dengan melakukan modifikasi berupa teks tambahan 'kita coba modifikasi dokumen pengajuan 05' yang berada dalam dokumen.



Gbr. 20 Contoh File Dokumen PDF yang Dimodifikasi.

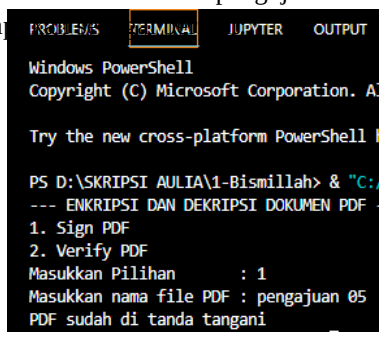
Adapun hasil proses *verify* dari file PDF yang sudah dimodifikasi adalah sebagai berikut:



Gbr. 21 Hasil Proses Verify dengan Dokumen PDF yang Sudah Dimodifikasi.

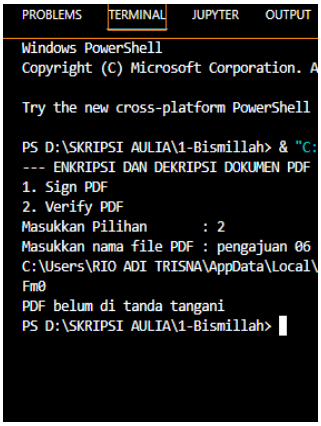
Pengujian tambahan sistem ini dilakukan ketika dokumen PDF sudah ditandatangani, maka

akan menampilkan pesan "PDF sudah di tanda tangani". Hasil dari pengujian tersebut dapat ditam



Gbr. 22 Hasil Proses *Sign* pada PDF yang Sudah Ditandatangan.

Sedangkan ketika melakukan proses verify terhadap dokumen PDF yang belum ditandatangani, maka akan menampilkan pesan bahwasanya "PDF belum di tanda tangani". Sebagaimana yang tertera pada gambar berikut ini:



Gbr. 23 Hasil Proses *Verify* pada PDF yang Belum Ditandatangan.

C. Tingkat performa proses *signing* dan *verifying* pada tanda tangan digital dengan Algoritma ECDSA

Pada proses ini akan dilakukan pengujian kecepatan tingkat *sign* dan *verify* pada dokumen PDF yang berbeda.

Adapun hasil uji dari proses *sign* terdapat pada Tabel. 1, dengan menggunakan 11 sampel file Pdf dengan ukuran yang berbeda-beda, sehingga menghasilkan perbedaan kinerja waktu yang didapat.

Tabel. 1 Kecepatan Waktu Proses *Sign*

No	Nama	Ukuran	Kecepatan
1	Undangan Pembinaan	12 kb	0,0197749137878417
2	Surat Permohonan	50 kb	0,1720037460327140
3	Undangan Peserta Bpc	65 kb	0,0249924659729003
4	Pengajuan 05	87 kb	0,1329996585845940
5	Surat Serah Terima	104 kb	0,2410242557525630
6	Undangan Senam Sehat Kkn	126 kb	0,5016126632690420
7	Pengajuan 04	174 kb	0,2089993953704830
8	Undangan Pengambilan Rapor	182 kb	0,0159845352172851
9	Undangan Cabang Bjn	226 kb	0,3470261096954340
10	Undangan Pengurus	288 kb	0,4420371055603020
11	Undangan Alumni	325 kb	0,2880232334136960

Sedangkan untuk performa kecepatan waktu proses *verify* terdapat pada Tabel. 2.

Tabel.2 Kecepatan Waktu Proses *Verify*

No	Nama	Ukuran	Kecepatan
1	Undangan Pembinaan	12 kb	0.017177343368530273
2	Surat Permohonan	50 kb	0.016010284423828125
3	Undangan Peserta Bpc	65 kb	0.016011476516723633
4	Pengajuan 05	87 kb	0.017386674880981445
5	Surat Serah Terima	104 kb	0.01601243019104004
6	Undangan Senam Sehat Kkn	126 kb	0.017180442810058594
7	Pengajuan 04	174 kb	0.036002159118652344
8	Undangan Pengambilan Rapor	182 kb	0.017000436782836914
9	Undangan Cabang Bjn	226 kb	0.01601266860961914
10	Undangan Pengurus	288 kb	0.016631603240966797
11	Undangan Alumni	325 kb	0.015998125076293945

Gbr. 23 Grafik Perbandingan Kecepatan Waktu *Sign* & *Verify*.

Berdasarkan tabel 3.1 dan 3.2 tersebut dirangkum dalam sebuah grafik yang menjelaskan bahwasanya proses *verify* itu lebih cepat dibandingkan

ketika proses *Sign.* Karena terdapat proses perhitungan dalam penentuan nilai algoritma yang bertujuan untuk menghasilkan nilai acak untuk *privat key*, *public key* dan *signature*.

IV. KESIMPULAN

Berdasarkan penelitian ini terdapat beberapa kesimpulan yaitu:

1. Penggunaan Algoritma ECDSA pada *digital signature* dalam suatu dokumen surat – menyurat dengan format file PDF pada proses *sign* dan *verify* dapat diterapkan dan dijalankan dokumen yang berjalan dengan baik oleh sistem.
2. Untuk mendapatkan dokumen yang valid atau bernilai *true* maka diperlukan masukan hasil nilai *signature*, *public key* yang sesuai dan file dokumen yang belum dimodifikasi setelah proses penandatanganan.
3. Berdasarkan hasil uji dari performa kecepatan waktu, bahwasanya tingkat ukuran dokumen tidak berpangaruh pada kecepatan waktu proses *sign.* Akan tetapi waktu proses *sign* menghasilkan proses yang lebih lama dibandingkan dengan proses *verify*.

UCAPAN TERIMAKASIH

Ucapan terimakasih dan syukur saya sampaikan kepada :

1. Allah SWT atas segala rahmat dan hidayah, sehingga penulis bisa menyelesaikan penelitian ini.
2. Orang tua dan keluarga penulis yang senantiasa menberikan dukungan untuk selalu melakukan apaun yang terbaik.
3. Bapak Asmunin, S.Kom., M.Kom. selaku Dosen Pembimbing yang dengan sabar membimbing penelitian ini dari awal hingga akhir.
4. Bapak I Made Suartana, S.Kom., M.Kom. dan bapak Ricky Eka Putra, S.kom.,M.Kom selaku dosen penguji yang selalu memberikan saran yang baik
5. Rio adi trisna yang selalu memberikan selalu memberikan dukungan terbaiknya.
6. Sahabat-sahabat dan teman-teman yang selalu memberikan semangat dan dukungan

REFERENSI

[1] Anjelina, S. (2020). Aspek Hukum Penggunaan Tanda Tangan Digital Dalam Menyelesaikan Penandatanganan Dokumen-Dokumen Bisnis Perusahaan-Perusahaan Yang Dikembangkan Oleh Privyid (. 2019, 1–42.

[2] Anshori, Y., Erwin Dodu, A. Y., & Wedananta, D. M. P. (2019). Implementasi Algoritma Kriptografi Rivest Shamir Adleman (RSA) pada Tanda Tangan Digital. Techno.Com, 18(2), 110–121.

[3] Aggarwal, S., & Kumar, N. (2021). Digital signatures☆. In *Advances in Computers* (Vol. 121, pp. 95–107). Academic Press Inc.

[4] Saputro, T. H., Hidayati, N. H., & Ujianto, E. I. H. (2020). Survei Tentang Algoritma Kriptografi Asimetris. Jurnal Informatika Polinema, 6(2), 67–72.

[5] Kusuma, V., Matematika, J., & Matematika, F. (2014). Elliptic Curve dan Implementasinya pada Algoritma Tanda Tangan Digital. Jurnal Sains Dan Seni ITS, 3(2), 3–6.

[6] Taleb, A. R., & Vergnaud, D. (2020). Speeding-up verification of digital signatures. Journal of Computer and System Sciences, 116, 22–39..

[7] Triwinarko, A. (2005). Elliptic Curve Digital Signature Algorithm (ECDSA). 1–6.

[8] Aumasson, J., Neves, S., Hearn, Z. W., & Winnerlein, C. (2013). BLAKE2 : Simpler , Smaller , Fast as MD5. 119–135.

[9] Gustiarum, P., Kusyanti, A., & Data, M. (2019). Implementasi Algoritme BLAKE2b untuk Pengecekan Integritas File. 3(4), 3702–3708.