Implementasi dan Evaluasi Sistem Tanda Tangan Digital Dokumen PDF Berbasis Ed25519 untuk Verifikasi Integritas Dokumen

Alfaza Naufal Zakiy – 18222126
Program Studi Sistem dan Teknologi Informasi
Sekolah Teknik Elektro dan Informatika
Institut Teknologi Bandung, Jalan Ganesha 10 Bandung
E-mail: alfazazakiy21@gmail.com , 18222126@std.stei.itb.ac.id

***Abstract*—Dokumen PDF banyak digunakan karena mampu mempertahankan tampilan dokumen pada berbagai perangkat, tetapi tetap memiliki risiko dimodifikasi atau dipalsukan tanpa perubahan yang mudah terlihat. Makalah ini membahas implementasi sistem tanda tangan digital dokumen PDF berbasis Ed25519 untuk memverifikasi integritas dokumen. Sistem dibangun menggunakan Python dan mencakup pembangkitan pasangan kunci, penandatanganan PDF, verifikasi *signature*, serta pengujian terhadap dokumen yang dimodifikasi. Pengujian dilakukan pada tiga dokumen PDF berukuran 0,79 MB, 5,69 MB, dan 24,22 MB. Hasil pengujian menunjukkan bahwa dokumen asli menghasilkan status VALID, sedangkan dokumen yang mengalami perubahan metadata, penambahan halaman, penghapusan halaman, dan perubahan satu byte menghasilkan status INVALID. *Signature* Ed25519 yang dihasilkan selalu berukuran 64 byte, sementara waktu signing dan verification meningkat seiring bertambahnya ukuran dokumen, tetapi tetap kurang dari 0,1 detik untuk dokumen terbesar. Hasil ini menunjukkan bahwa Ed25519 dapat digunakan sebagai mekanisme tanda tangan digital yang sederhana dan efisien untuk memverifikasi integritas dokumen PDF.**

Kata kunci—Ed25519, tanda tangan digital, dokumen PDF, verifikasi integritas, kriptografi kunci publik.

I. PENDAHULUAN

Perkembangan teknologi informasi telah mendorong penggunaan dokumen digital dalam berbagai aktivitas sehari-hari. Saat ini berbagai dokumen seperti laporan, kontrak, sertifikat, surat resmi, hingga dokumen akademik umumnya disimpan dan didistribusikan dalam bentuk digital. Salah satu format dokumen yang paling banyak digunakan adalah *Portable Document Format* (PDF) karena mampu mempertahankan tampilan dokumen secara konsisten pada berbagai sistem operasi maupun perangkat yang berbeda.

Meningkatnya penggunaan dokumen digital juga diikuti dengan meningkatnya risiko terhadap keamanan dokumen tersebut. Dokumen PDF dapat dengan mudah disalin, dimodifikasi, ataupun dipalsukan tanpa meninggalkan jejak yang terlihat secara kasat mata. Perubahan sekecil apa pun pada isi dokumen dapat mengubah informasi yang terkandung di dalamnya sehingga menimbulkan kerugian bagi pihak yang berkepentingan. Oleh karena itu, diperlukan suatu mekanisme yang mampu menjamin bahwa dokumen yang diterima benar-benar berasal dari pihak yang berwenang serta tidak mengalami perubahan sejak pertama kali ditandatangani [1].

Salah satu solusi untuk menjamin keaslian dan integritas dokumen digital adalah dengan menggunakan tanda tangan digital (*digital signature*). Berbeda dengan tanda tangan hasil pemindaian (*scanned signature*), tanda tangan digital dibentuk menggunakan teknik kriptografi kunci publik sehingga memiliki keterkaitan langsung dengan isi dokumen. Tanda tangan digital mampu memberikan layanan keamanan berupa autentikasi identitas penandatangan, integritas data, serta *non-repudiation* (nirpenyangkalan). Apabila isi dokumen mengalami perubahan setelah proses penandatanganan, maka proses verifikasi tanda tangan digital akan gagal sehingga perubahan tersebut dapat segera dideteksi [1].

Berbagai algoritma tanda tangan digital telah dikembangkan, seperti RSA, DSA, ECDSA, dan EdDSA. Salah satu implementasi EdDSA yang banyak digunakan saat ini adalah Ed25519. Algoritma ini dibangun di atas *Twisted Edwards Curve Curve25519* dan dirancang untuk menghasilkan proses pembangkitan kunci, penandatanganan, serta verifikasi yang cepat tanpa mengurangi tingkat keamanan. Selain memiliki ukuran kunci yang relatif kecil, Ed25519 juga telah digunakan pada berbagai aplikasi keamanan modern karena efisiensi dan ketahanannya terhadap berbagai serangan kriptografi [2], [3].

Makalah ini membahas implementasi sistem tanda tangan digital dokumen PDF menggunakan algoritma Ed25519. Implementasi dilakukan menggunakan bahasa pemrograman Python dengan memanfaatkan pustaka kriptografi yang mendukung algoritma Ed25519. Sistem yang dibangun mampu melakukan pembangkitan pasangan kunci, penandatanganan dokumen PDF, serta verifikasi tanda tangan digital untuk memastikan integritas dokumen.

Pengujian dilakukan terhadap beberapa skenario, yaitu verifikasi dokumen asli, perubahan isi dokumen, perubahan metadata, serta penambahan maupun penghapusan halaman pada dokumen PDF. Hasil pengujian diharapkan dapat menunjukkan bahwa algoritma Ed25519 mampu mendeteksi

setiap perubahan yang terjadi pada dokumen sehingga dapat digunakan sebagai mekanisme verifikasi integritas dokumen digital secara efektif.

II. Dasar Teori

A. Dokumen PDF (*Portable Document Format*)

Portable Document Format (PDF) merupakan format dokumen digital yang dikembangkan oleh Adobe Systems untuk mempertahankan tampilan dokumen agar tetap konsisten pada berbagai perangkat dan sistem operasi. Saat ini PDF telah menjadi standar internasional ISO 32000 dan digunakan secara luas dalam berbagai bidang, seperti administrasi, pendidikan, pemerintahan, maupun bisnis. Salah satu keunggulan format PDF adalah kemampuannya menyimpan teks, gambar, grafik, serta elemen lainnya dalam satu berkas tanpa mengubah tata letak dokumen ketika dibuka pada perangkat yang berbeda [4].

Meskipun memiliki berbagai keunggulan, dokumen PDF tetap memiliki risiko keamanan. Dokumen dapat dimodifikasi, disalin, maupun dipalsukan tanpa mengubah tampilan secara signifikan. Oleh karena itu, diperlukan mekanisme tambahan untuk memastikan bahwa dokumen yang diterima masih sama dengan dokumen yang dikirimkan. Salah satu mekanisme yang umum digunakan adalah tanda tangan digital (*digital signature*).

B. Tanda Tangan Digital (*Digital Signature*)

Tanda tangan digital merupakan mekanisme kriptografi yang digunakan untuk menjamin autentikasi, integritas data, dan *non-repudiation* (nirpenyangkalan). Berbeda dengan tanda tangan hasil pemindaian, tanda tangan digital dibentuk menggunakan algoritma kriptografi kunci publik sehingga bergantung langsung pada isi dokumen yang ditandatangani [1].

Secara umum proses tanda tangan digital terdiri atas dua tahap, yaitu proses penandatanganan (*signing*) dan proses verifikasi (*verification*). Pada proses penandatanganan, dokumen terlebih dahulu diproses menggunakan fungsi hash untuk menghasilkan *message digest*. Selanjutnya *message digest* tersebut ditandatangani menggunakan kunci privat sehingga menghasilkan tanda tangan digital. Pada proses verifikasi, penerima menggunakan kunci publik untuk memverifikasi tanda tangan dan membandingkan nilai hash dokumen. Apabila kedua nilai hash sama, maka dokumen dinyatakan autentik dan tidak mengalami perubahan [1].

C. Algoritma EdDSA dan Ed25519

Edwards-curve Digital *Signature* Algorithm (EdDSA) merupakan algoritma tanda tangan digital berbasis kurva eliptik yang dirancang untuk memberikan keamanan tinggi dengan performa yang lebih baik dibandingkan beberapa algoritma sebelumnya, seperti DSA maupun ECDSA. EdDSA memanfaatkan kurva Edwards sehingga proses operasi aritmetika pada kurva menjadi lebih efisien dan memiliki ketahanan yang baik terhadap berbagai serangan implementasi [2].

Salah satu implementasi EdDSA yang paling banyak digunakan adalah Ed25519. Algoritma ini menggunakan kurva *Twisted Edwards Curve25519* dengan tingkat keamanan sekitar 128-bit. Ed25519 menghasilkan ukuran kunci publik sebesar 32 byte, kunci privat sebesar 32 byte, dan tanda tangan digital berukuran 64 byte. Selain memiliki ukuran kunci yang relatif kecil, Ed25519 juga menawarkan proses pembangkitan kunci, penandatanganan, dan verifikasi yang cepat sehingga banyak diterapkan pada berbagai aplikasi keamanan modern, seperti OpenSSH, Git, Signal, dan WireGuard [2], [3].

D. Fungsi Hash SHA-512

Fungsi hash merupakan fungsi satu arah (*one-way function*) yang mengubah data dengan ukuran sembarang menjadi nilai hash (*message digest*) dengan ukuran tetap. Salah satu karakteristik utama fungsi hash adalah perubahan kecil pada data masukan akan menghasilkan nilai hash yang sangat berbeda. Selain itu, fungsi hash juga memiliki sifat sulit dibalik (*pre-image resistant*) serta tahan terhadap pencarian dua masukan berbeda yang menghasilkan hash yang sama (*collision resistant*) [5].

Pada algoritma Ed25519, fungsi hash SHA-512 digunakan dalam beberapa tahapan pembentukan pasangan kunci maupun proses pembangkitan tanda tangan digital. Penggunaan SHA-512 bertujuan menghasilkan *message digest* yang memiliki tingkat keamanan tinggi sehingga mendukung keandalan algoritma Ed25519 [2].

E. Verifikasi Integritas Dokumen

Integritas dokumen merupakan jaminan bahwa isi dokumen tidak mengalami perubahan sejak pertama kali dibuat atau ditandatangani. Pada sistem tanda tangan digital, integritas dokumen diverifikasi dengan menghitung kembali nilai hash dari dokumen yang diterima, kemudian membandingkannya dengan nilai hash yang diperoleh melalui proses verifikasi tanda tangan digital menggunakan kunci publik [1].

Apabila isi dokumen mengalami perubahan, meskipun hanya satu karakter atau perubahan metadata tertentu yang memengaruhi isi berkas, maka nilai hash yang dihasilkan akan berbeda sehingga proses verifikasi tanda tangan digital dinyatakan gagal. Dengan demikian, tanda tangan digital tidak hanya berfungsi sebagai bukti identitas penandatangan, tetapi juga sebagai mekanisme untuk memastikan integritas dokumen digital.

III. Implementasi

Implementasi sistem tanda tangan digital dokumen PDF pada penelitian ini dilakukan dengan menggunakan bahasa pemrograman Python. Sistem yang dibangun berfokus pada proses kriptografi berbasis Ed25519, yaitu pembangkitan pasangan kunci, penandatanganan dokumen PDF, serta verifikasi tanda tangan digital untuk memastikan integritas dokumen. *Signature* disimpan sebagai berkas terpisah dengan ekstensi .sig, sedangkan kunci privat dan kunci publik disimpan dalam format PEM.

Implementasi ini dirancang sebagai aplikasi berbasis command line agar mudah dijalankan melalui terminal tanpa memerlukan antarmuka grafis. Dokumen PDF diproses sebagai data biner sehingga tanda tangan digital yang dihasilkan bergantung pada keseluruhan isi byte dokumen. Oleh karena itu setiap perubahan pada dokumen, baik perubahan isi, metadata, struktur halaman, maupun perubahan byte secara langsung, akan menyebabkan proses verifikasi gagal.

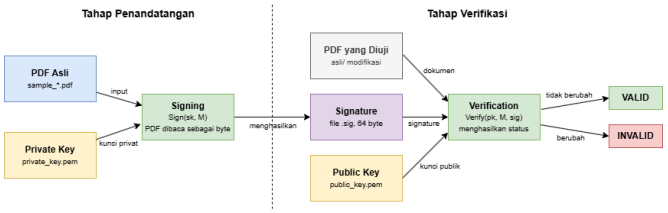
A. Perancangan Sistem

Sistem yang dirancang terdiri atas tiga proses utama, yaitu pembangkitan pasangan kunci, penandatanganan dokumen PDF, dan verifikasi tanda tangan digital. Pada proses pembangkitan kunci, sistem menghasilkan pasangan kunci Ed25519 yang terdiri atas kunci privat dan kunci publik. Kunci privat digunakan oleh penandatangan untuk menghasilkan tanda tangan digital, sedangkan kunci publik digunakan oleh pihak penerima untuk melakukan verifikasi.

Alur umum sistem yang diimplementasikan ditunjukkan pada Fig. 1. Pada tahap penandatanganan, dokumen PDF asli dan kunci privat digunakan sebagai masukan untuk proses signing. Proses tersebut menghasilkan berkas *signature* dengan ekstensi .sig. Pada tahap verifikasi, sistem menggunakan dokumen PDF yang diuji, *signature*, dan kunci publik untuk menentukan apakah dokumen masih valid atau telah mengalami perubahan.

Pada proses penandatanganan, dokumen PDF dibaca dalam bentuk byte. Seluruh byte dokumen tersebut kemudian diproses menggunakan algoritma Ed25519 dengan kunci privat untuk menghasilkan tanda tangan digital. *Signature* yang dihasilkan tidak disisipkan ke dalam dokumen PDF, melainkan disimpan sebagai berkas terpisah dengan ekstensi .sig. Pemisahan ini dilakukan agar dokumen asli tidak mengalami perubahan setelah proses penandatanganan.

Pada proses verifikasi, sistem membaca kembali dokumen PDF, file *signature*, dan kunci publik. Ketiga komponen tersebut digunakan untuk memeriksa apakah tanda tangan digital masih sesuai dengan isi dokumen. Jika dokumen tidak mengalami perubahan sejak ditandatangani, maka sistem menghasilkan status VALID. Sebaliknya, apabila terdapat perubahan pada dokumen atau *signature* tidak sesuai, maka sistem menghasilkan status INVALID.



Gambar. 1. Alur sistem tanda tangan digital dokumen PDF berbasis Ed25519.

B. Pembangkitan Pasangan Kunci

Pembangkitan pasangan kunci dilakukan menggunakan algoritma Ed25519 yang tersedia pada pustaka cryptography. Proses ini menghasilkan dua berkas utama, yaitu private_key.pem dan public_key.pem. Kunci privat disimpan pada folder keys/ dan digunakan dalam proses penandatanganan dokumen. Kunci publik juga disimpan pada folder yang sama dan digunakan dalam proses verifikasi tanda tangan digital.

Kunci privat disimpan dalam format PEM tanpa enkripsi agar proses pengujian dapat dilakukan secara sederhana melalui terminal. Meskipun demikian, pada penerapan nyata, kunci privat sebaiknya disimpan secara aman, misalnya dengan enkripsi atau mekanisme manajemen kunci khusus. Pada implementasi ini, pembangkitan pasangan kunci dijalankan melalui perintah python main.py keygen.

Berdasarkan hasil pengujian, proses pembangkitan pasangan kunci berhasil menghasilkan kunci privat dan kunci publik. Kunci privat tersimpan pada keys/private_key.pem, sedangkan kunci publik tersimpan pada keys/public_key.pem. Waktu pembangkitan pasangan kunci yang tercatat pada pengujian adalah 0,008907 detik.

C. Proses Penandatanganan Dokumen PDF

Proses penandatanganan dilakukan terhadap dokumen PDF yang telah disiapkan pada folder sample_pdfs/. Sistem membaca dokumen PDF sebagai data biner, kemudian menandatangani data tersebut menggunakan kunci privat Ed25519. Hasil dari proses ini adalah tanda tangan digital berukuran 64 byte yang disimpan pada folder signatures/.

Secara konseptual, proses penandatanganan dapat dituliskan sebagai berikut:

M = byte dokumen PDF

$\sigma = \text{Sign}(sk, M)$

Pada persamaan tersebut, M merupakan byte dokumen PDF, sk merupakan kunci privat Ed25519, dan σ merupakan tanda tangan digital yang dihasilkan. Dalam implementasi, proses ini dijalankan melalui perintah python main.py sign sample_pdfs/sample_small.pdf. *Signature* yang dihasilkan diberi nama berdasarkan nama file PDF yang ditandatangani, misalnya sample_small.pdf.sig.

Pada pengujian awal terhadap sample_small.pdf, proses penandatanganan berhasil menghasilkan file signatures/sample_small.pdf.sig. Waktu signing yang tercatat adalah 0,003821 detik dengan ukuran *signature* sebesar 64 byte.

D. Proses Verifikasi Integritas Dokumen

Proses verifikasi dilakukan untuk memastikan bahwa dokumen PDF yang diterima masih sama dengan dokumen pada saat proses penandatanganan dilakukan. Pada tahap ini, sistem membaca dokumen PDF, *signature*, dan kunci publik. Selanjutnya, algoritma Ed25519 digunakan untuk memverifikasi apakah *signature* tersebut sesuai dengan isi dokumen.

Secara konseptual, proses verifikasi dapat dituliskan sebagai berikut:

$\text{Verify}(pk, M, \sigma) = \text{VALID atau INVALID}$

Pada persamaan tersebut, pk merupakan kunci publik Ed25519, M merupakan byte dokumen PDF yang diverifikasi, dan σ merupakan *signature* yang sebelumnya dihasilkan dari proses penandatanganan. Apabila dokumen tidak berubah, maka hasil verifikasi adalah VALID. Namun, apabila dokumen mengalami perubahan sekecil apa pun, maka hasil verifikasi menjadi INVALID.

Dalam implementasi ini, proses verifikasi dijalankan melalui perintah python main.py verify sample_pdfs/sample_small.pdf signatures/sample_small.pdf.sig. Berdasarkan pengujian awal, verifikasi terhadap dokumen PDF asli menghasilkan status VALID dengan waktu verifikasi 0,004486 detik. Hal ini menunjukkan bahwa *signature* yang dihasilkan pada proses signing sesuai dengan isi dokumen PDF asli.

Verifikasi juga dilakukan terhadap file PDF yang telah dimodifikasi untuk menguji kemampuan sistem dalam mendeteksi perubahan dokumen. Modifikasi yang diuji meliputi perubahan metadata, penambahan halaman, penghapusan halaman, dan perubahan satu byte pada file PDF. Jika salah satu perubahan tersebut dilakukan setelah proses penandatanganan, maka *signature* tidak lagi sesuai dengan isi dokumen sehingga sistem menghasilkan status INVALID.

IV. Hasil dan Pengujian

Bab ini membahas hasil pengujian dari sistem tanda tangan digital dokumen PDF berbasis Ed25519 yang telah diimplementasikan. Pengujian dilakukan untuk mengetahui keberhasilan proses pembangkitan kunci, penandatanganan dokumen, verifikasi tanda tangan digital, serta kemampuan sistem dalam mendeteksi perubahan pada dokumen PDF. Selain itu, dilakukan pula pengukuran performa dasar berupa waktu penandatanganan, waktu verifikasi, dan ukuran *signature* yang dihasilkan.

A. Lingkungan Pengujian

Pengujian dilakukan pada lingkungan berbasis command line menggunakan bahasa pemrograman Python. Library utama yang digunakan adalah cryptography untuk implementasi algoritma Ed25519, pypdf untuk memodifikasi dokumen PDF pada skenario pengujian integritas, pandas untuk menyimpan hasil pengujian ke dalam file CSV, serta matplotlib untuk menghasilkan grafik performa.

Dokumen uji yang digunakan terdiri atas tiga file PDF dengan ukuran berbeda. Ketiga dokumen tersebut diletakkan pada folder sample_pdfs/ dan digunakan untuk mengukur pengaruh ukuran dokumen terhadap waktu penandatanganan dan verifikasi. Rincian dokumen uji ditunjukkan pada Tabel I.

TABEL I. Dokumen PDF yang Digunakan Dalam Pengujian

Nama File	Ukuran Dokumen
sample_small.pdf	0,79 MB
sample_medium.pdf	5,69 MB
sample_large.pdf	24,22 MB

B. Hasil Penandatanganan

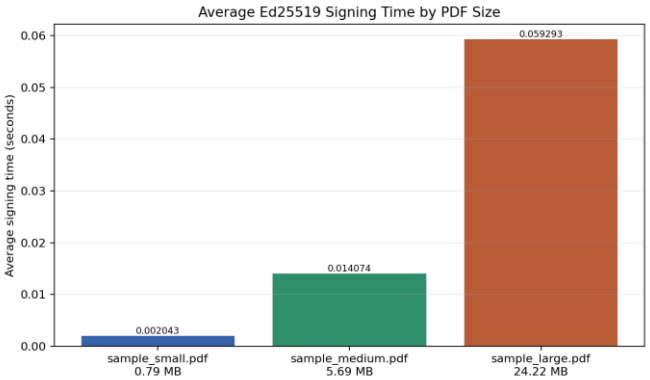
Pengujian penandatanganan dilakukan dengan menandatangani masing-masing dokumen PDF menggunakan kunci privat Ed25519. Setiap pengujian dilakukan sebanyak lima kali, kemudian dihitung nilai rata-rata waktu penandatanganan dan verifikasi. *Signature* yang dihasilkan disimpan sebagai file terpisah dengan ekstensi .sig.

Hasil pengujian menunjukkan bahwa proses penandatanganan berhasil dilakukan pada seluruh dokumen uji. Ukuran *signature* yang dihasilkan selalu sebesar 64 byte untuk setiap dokumen. Hal ini sesuai dengan karakteristik algoritma Ed25519 yang menghasilkan tanda tangan digital dengan ukuran tetap, terlepas dari ukuran dokumen yang ditandatangani.

TABEL II. Hasil Pengujian Performa Signing dan Verification

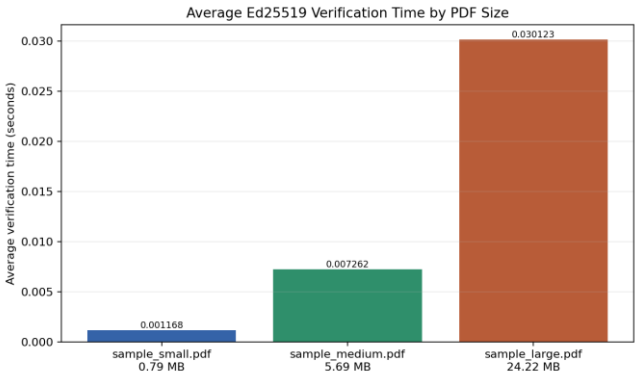
Nama File	Ukuran Dokumen	Rata-rata Signing	Rata-rata Verification	Ukuran Signature
sample_small.pdf	0,79 MB	0,002043 s	0,001168 s	64 byte
sample_medium.pdf	5,69 MB	0,014074 s	0,007262 s	64 byte
sample_large.pdf	24,22 MB	0,059293 s	0,030123 s	64 byte

Gambar 2 menunjukkan grafik rata-rata waktu penandatanganan untuk setiap dokumen PDF. Berdasarkan grafik tersebut, terlihat bahwa waktu penandatanganan meningkat seiring bertambahnya ukuran dokumen.



Gambar. 2. Rata-rata waktu penandatanganan dokumen PDF.

Gambar 3 menunjukkan grafik rata-rata waktu verifikasi untuk setiap dokumen PDF. Pola yang sama juga terlihat pada proses verifikasi, yaitu dokumen dengan ukuran lebih besar membutuhkan waktu verifikasi yang lebih lama.



Gambar. 3. Rata-rata waktu verifikasi dokumen PDF.

C. Pengujian Modifikasi Dokumen

Pengujian modifikasi dokumen dilakukan untuk mengetahui kemampuan sistem dalam mendeteksi perubahan pada dokumen PDF setelah proses penandatanganan. Pada pengujian ini, dokumen asli terlebih dahulu ditandatangani sehingga menghasilkan *signature*. Selanjutnya, *signature* tersebut digunakan untuk memverifikasi dokumen asli dan beberapa dokumen hasil modifikasi.

Dokumen yang digunakan pada pengujian integritas adalah sample_medium.pdf. Dokumen ini dipilih karena memiliki lebih dari satu halaman sehingga skenario penghapusan halaman dapat dilakukan. Skenario pengujian yang dilakukan meliputi verifikasi dokumen asli, perubahan metadata, penambahan halaman, penghapusan halaman, dan perubahan satu byte pada file PDF.

TABEL III. HASIL PENGUJIAN INTEGRITAS DOKUMEN PDF

Skenario	File yang Diuji	Hasil yang Diharapkan	Hasil Aktual	Status
Original PDF	sample_medium.pdf	VALID	VALID	PASS
Metadata modified	sample_medium_metadata_modified.pdf	INVALID	INVALID	PASS
Page added	sample_medium_page_added.pdf	INVALID	INVALID	PASS
Page removed	sample_medium_page_removed.pdf	INVALID	INVALID	PASS
Byte modified	sample_medium_byte_modified.pdf	INVALID	INVALID	PASS

Berdasarkan hasil pada Tabel III, dokumen PDF asli berhasil diverifikasi dengan status VALID. Sementara itu, seluruh dokumen yang telah dimodifikasi menghasilkan status INVALID. Hal ini menunjukkan bahwa sistem mampu mendeteksi perubahan pada dokumen PDF, baik perubahan pada metadata, struktur halaman, maupun perubahan kecil pada tingkat byte.

D. Analisis Hasil

Hasil pengujian menunjukkan bahwa algoritma Ed25519 dapat digunakan untuk melakukan tanda tangan digital pada dokumen PDF dan memverifikasi integritas dokumen tersebut. Dokumen PDF asli menghasilkan status VALID karena isi dokumen masih sama dengan kondisi saat proses penandatanganan dilakukan. Sebaliknya, dokumen yang telah mengalami perubahan menghasilkan status INVALID karena *signature* tidak lagi sesuai dengan isi dokumen.

Dari sisi performa, waktu penandatanganan dan verifikasi meningkat seiring bertambahnya ukuran dokumen PDF. Hal ini terjadi karena sistem membaca seluruh byte dokumen sebagai masukan pada proses signing dan verification. Oleh karena itu, semakin besar ukuran dokumen, semakin banyak data yang harus diproses oleh sistem. Meskipun demikian, waktu yang dibutuhkan masih relatif kecil, yaitu kurang dari 0,1 detik untuk dokumen berukuran 24,22 MB pada lingkungan pengujian yang digunakan.

Selain itu, ukuran *signature* yang dihasilkan selalu tetap sebesar 64 byte pada seluruh dokumen uji. Hal ini menunjukkan bahwa ukuran tanda tangan digital Ed25519 tidak bergantung pada ukuran dokumen yang ditandatangani. Dengan demikian, Ed25519 dapat memberikan efisiensi dari sisi ukuran *signature* sekaligus tetap mampu mendeteksi perubahan pada dokumen PDF secara akurat.

Berdasarkan keseluruhan pengujian, sistem yang diimplementasikan telah memenuhi tujuan utama, yaitu melakukan pembangkitan pasangan kunci, penandatanganan dokumen PDF, verifikasi tanda tangan digital, serta pendeteksian perubahan dokumen. Hasil pengujian integritas juga menunjukkan bahwa perubahan sekecil apa pun pada file PDF dapat menyebabkan proses verifikasi gagal, sehingga sistem dapat digunakan sebagai mekanisme untuk menjaga integritas dokumen digital.

V. KESIMPULAN

Berdasarkan implementasi dan pengujian yang telah dilakukan, sistem tanda tangan digital dokumen PDF berbasis Ed25519 berhasil dibangun menggunakan bahasa pemrograman Python. Sistem mampu melakukan pembangkitan pasangan kunci, penandatanganan dokumen PDF, serta verifikasi tanda tangan digital menggunakan pasangan kunci Ed25519. *Signature* yang dihasilkan disimpan sebagai berkas terpisah dengan ekstensi .sig, sehingga dokumen PDF asli tidak mengalami perubahan setelah proses penandatanganan.

Hasil pengujian menunjukkan bahwa dokumen PDF asli dapat diverifikasi dengan status VALID, sedangkan dokumen yang telah dimodifikasi menghasilkan status INVALID. Modifikasi yang diuji meliputi perubahan metadata, penambahan halaman, penghapusan halaman, dan perubahan satu byte pada file PDF. Seluruh skenario modifikasi tersebut berhasil terdeteksi oleh sistem, sehingga dapat disimpulkan bahwa tanda tangan digital berbasis Ed25519 mampu menjaga dan memverifikasi integritas dokumen PDF.

Dari sisi performa, waktu penandatanganan dan verifikasi meningkat seiring bertambahnya ukuran dokumen PDF. Hal ini disebabkan karena sistem memproses seluruh byte dokumen sebagai masukan dalam proses signing dan verification. Meskipun demikian, waktu proses yang diperoleh masih relatif kecil, bahkan untuk dokumen berukuran 24,22 MB. Selain itu, ukuran *signature* yang dihasilkan selalu tetap sebesar 64 byte pada seluruh dokumen uji, sesuai dengan karakteristik algoritma Ed25519.

Secara keseluruhan, implementasi ini menunjukkan bahwa Ed25519 dapat digunakan sebagai mekanisme tanda tangan digital yang sederhana, efisien, dan efektif untuk memverifikasi integritas dokumen PDF. Sistem yang dibangun dapat mendeteksi perubahan dokumen secara akurat serta mudah dijalankan melalui terminal, sehingga sesuai digunakan sebagai studi implementasi tanda tangan digital pada dokumen digital.

SOURCE CODE

https://github.com/AlfazaNaufalZakiy/ed25519-pdf-signature

UCAPAN TERIMA KASIH

Penulis mengucapkan puji syukur kepada Tuhan Yang Maha Esa atas rahmat dan karunia-Nya sehingga makalah ini dapat diselesaikan dengan baik. Penulis juga menyampaikan terima kasih yang sebesar-besarnya kepada Bapak Rinaldi Munir selaku dosen pengampu mata kuliah Kriptografi atas ilmu, bimbingan, dan materi pembelajaran yang telah diberikan selama perkuliahan. Pengetahuan dan wawasan yang diberikan menjadi dasar penting bagi penulis dalam memahami konsep kriptografi serta menyusun makalah ini.

Penulis juga mengucapkan terima kasih kepada keluarga dan teman-teman yang telah memberikan dukungan, semangat, serta masukan selama proses penyusunan makalah. Semoga makalah ini dapat memberikan manfaat bagi pembaca dan menjadi kontribusi kecil dalam pemahaman penerapan tanda tangan digital untuk menjaga integritas dokumen digital.

REFERENSI

[1] R. Munir, "Tanda Tangan Digital," Materi Kuliah II4021 Kriptografi, STEI ITB, 2026.

[2] S. Josefsson and I. Liusvaara, "Edwards-Curve Digital Signature Algorithm (EdDSA)," RFC 8032, Internet Engineering Task Force, 2017.

[3] D. J. Bernstein, N. Duif, T. Lange, P. Schwabe, and B.-Y. Yang, "High-speed high-security signatures," Journal of Cryptographic Engineering, vol. 2, no. 2, pp. 77-89, 2012.

[4] ISO, "ISO 32000-2: Document management - Portable document format - Part 2: PDF 2.0," International Organization for Standardization, 2020.

[5] NIST, "FIPS PUB 180-4: Secure Hash Standard (SHS)," National Institute of Standards and Technology, 2015.

PERNYATAAN

Dengan ini saya menyatakan bahwa makalah yang saya tulis ini adalah tulisan saya sendiri, bukan saduran, atau terjemahan dari makalah orang lain, dan bukan plagiasi.

Bandung, 19 Juni 2026

Alfaza Naufal Zakiy dan 18222126