PERANCANGAN APLIKASI DIGISIGN SEBAGAI ALAT PENGESAHAN DOKUMEN PADA UTD PNJ

LAPORAN SKRIPSI

MUHAMMAD DIMAS YUDHA ADHI PRATAMA JR.

1807422025

PROGRAM STUDI TEKNIK MULTIMEDIA DAN JARINGAN

JURUSAN TEKNIK INFORMATIKA DAN KOMPUTER

POLITEKNIK NEGERI JAKARTA

2022

PERANCANGAN APLIKASI DIGISIGN SEBAGAI ALAT PENGESAHAN DOKUMEN PADA UTD PNJ

ANALISIS INTEGRITAS DOKUMEN DIGITAL PADA APLIKASI DIGISIGN UTD PNJ MENGGUNAKAN TANDA TANGAN DIGITAL

LAPORAN SKRIPSI

Dibuat untuk Melengkapi Syarat-Syarat yang Diperlukan untuk Memperoleh Diploma Empat Politeknik

MUHAMMAD DIMAS YUDHA ADHI PRATAMA JR.

1807422025

PROGRAM STUDI TEKNIK MULTIMEDIA DAN JARINGAN

JURUSAN TEKNIK INFORMATIKA DAN KOMPUTER

POLITEKNIK NEGERI JAKARTA

2022

HALAMAN PERNYATAAN BEBAS PLAGIARISME

Saya yang bertanda tangan di bawah ini:

Nama	:	Muhammad Dimas Yudha Adhi Pratama Jr.
NIM	:	1807422025
Jurusan / Program Studi	:	Teknik Informatika dan Komputer / Teknik Multimedia dan Jaringan
Judul Skripsi	:	Perancangan Aplikasi DigiSign Sebagai Alat Pengesahan Dokumen Pada UTD PNJ
Sub Judul Skripsi	:	Analisis Integritas Dokumen Digital Pada Aplikasi Digisign UTD PNJ Menggunakan Tanda Tangan Digital

Menyatakan dengan sebenarnya bahwa skripsi ini benar-benar merupakan hasil karya saya sendiri, bebas dari peniruan terhadap karya dari orang lain. Kutipan pendapat dan tulisan orang lain ditunjuk sesuai dengan cara-cara penulisan karya ilmiah yang berlaku.

Apabila di kemudian hari terbukti atau dapat dibuktikan bahwa dalam skripsi ini terkandung ciri-ciri plagiat dan bentuk-bentuk peniruan lain yang dianggap melanggar peraturan, maka saya bersedia menerima sanksi atas perbuatan tersebut.

Depok, 15 Juli 2022

Yang membuat pernyataan

(Muhammad Dimas Yudha Adhi Pratama Jr.)

NIM. 1807422025

HALAMAN PENGESAHAN

Skripsi diajukan oleh:

Nama : Muhammad Dimas Yudha Adhi Pratama Jr.

NIM : 1807422025

Program Studi : Teknik Multimedia dan Jaringan – Sistem Keamanan Informasi

Judul Skripsi : Perancangan Aplikasi DigiSign Sebagai Alat Pengesahan Dokumen Pada UTD PNJ

Sub Judul Skripsi : Analisis Integritas Dokumen Digital Pada Aplikasi Digisign UTD PNJ Menggunakan Tanda Tangan Digital

Telah diuji oleh tim penguji dalam Sidang Skripsi pada hari Selasa, Tanggal 26, Bulan Juli, Tahun 2022 dan dinyatakan **LULUS**.

Disahkan oleh

Pembimbing I : Defiana Arnaldy, S.Tp., M. Si .....

Penguji I : Dr. Prihatin Oktivasari, S.Si., M.Si. .....

Penguji II : Indra Hermawan, S.Kom., M.Kom .....

Penguji III : Ariawan Andi Suhandana, S.Kom., M.TI. .....

Mengetahui:
Jurusan Teknik Informatika dan Komputer
Ketua

Mauldy Laya, S.Kom.,M.Kom.
NIP.197802112009121003

SURAT PERNYATAAN PERSETUJUAN PUBLIKASI SKRIPSI UNTUK KEPENTINGAN AKADEMIS

Sebagai sivitas akademik Politeknik Negeri Jakarta, saya bertanda tangan di bawah ini :

Nama	:	Muhammad Dimas Yudha Adhi Pratama Jr.
NIM	:	1807422025
Program Studi	:	Teknik Multimedia dan Jaringan

Demi pengembangan ilmu pengetahuan , menyetujui untuk memberikan kepada Politeknik Negeri Jakarta Hak Bebas Royalti Non-Eksklusif atas karya ilmiah saya yang berjudul :

Analisis Integritas Dokumen Digital Pada Aplikasi Digisign UTD PNJ Menggunakan Tanda Tangan Digital

Beserta perangkat yang ada (jika diperlukan). Dengan Hak Bebas Royalti Non-Eksklusif ini Politeknik Negeri Jakarta Berhak menyimpan, mengalihmediakan/formatkan, mengelola dalam bentuk pangkalan data (database), merawat, dan mempublikasikan skripsi saya tanpa meminta izin dari saya selama tetap mencantumkan nama saya sebagai penulis/pencipta dan sebagai pemilik Hak Cipta.

Demikian pernyataan ini saya buat dengan sebenarnya.

Depok, 15 Juli 2022

Yang membuat pernyataan

(Muhammad Dimas Yudha Adhi Pratama Jr.)

NIM. 1807422025

Analisis Integritas Dokumen Digital Pada Aplikasi Digisign UTD PNJ Menggunakan Tanda Tangan Digital

ABSTRAK

Dewasa ini, penggunaan dokumen dalam bentuk digital atau dokumen elektronik semakin sering digunakan. Dikarenakan dokumen tersebut bersifat digital, akan sangat mudah untuk dilakukan modifikasi oleh segala pihak yang memiliki dokumen tersebut. Pada saat ini Politeknik Negeri Jakarta sedang membangun aplikasi Digisign yang digunakan untuk membuat dan memverifikasi dokumen digital dengan menerapkan teknologi tandatangan digital. Dalam *Digital Signature Standard* yang diterbitkan NIST, terdapat beberapa algoritma penandatanganan yang memenuhi standar tersebut, seperti DSA, RSA, dan ECDSA. Setiap algoritma memiliki kelebihan dan kekurangan masing-masing, maka dari itu perlu dikaji lebih lanjut dari ketiga algoritma yang merupakan algoritma terbaik untuk aplikasi Digisign dengan melakukan pengujian performa setiap algoritma pada aplikasi Digisign. Parameter pada pengujian ini meliputi penggunaan waktu penggunaan CPU, dan penggunaan memori pada tiga proses yang dilakukan pada aplikasi Digisign, yaitu pembuatan pasangan kunci, penandatanganan dokumen, dan verifikasi dokumen. Dalam pengujiannya digunakan 3 buah sampel dalam bentuk skema antara lain dokumen yang hanya berisi teks, dokumen campuran teks dan gambar, dan dokumen yang telah memiliki tandatangan. Hasil dari penelitian ini menunjukkan algoritma ECDSA lebih unggul dalam penggunaan waktu yang lebih cepat dibanding dua algoritma lainnya, sedangkan DSA lebih unggul dalam penggunaan CPU yang lebih sedikit, serta RSA yang lebih unggul dalam penggunaan memori.

Kata Kunci: Algoritma tandatangan digital, DSA, RSA, ECDSA, Performa algoritma

Kata Pengantar

Puji serta syukur penulis ucapkan kepada Allah SWT, karena atas rahmat dan karunia-Nya penulis dapat menyelesaikan laporan skripsi ini dengan baik. Penulisan laporan skripsi ini dilakukan dalam rangka memenuhi salah satu syarat untuk mencapai gelar Sarjana Terapan Politeknik. Dalam proses pembuatan laporan ini penulis menyadari tanpa bimbingan dari berbagai pihak, sehingga penulis dapat menyelesaikan laporan ini tepat waktu. Oleh karena itu penulis mengucapkan terima kasih kepada:

1. Bapak Defiana Arnaldy, S.Tp., M. Si selaku dosen pembimbing yang telah menyediakan waktu dan tenaga untuk mengarahkan penulis dalam penyusunan laporan skripsi ini;
2. Hilmi Abdurrahman Fakhrudin selaku teman sekelompok yang telah bersama-sama membuat aplikasi Digisign dan membantu dalam penulisan laporan skripsi ini.
3. Ayah dan Ibu yang selalu memberikan doa, kasih sayang, dan dukungan moral dalam menyelesaikan skripsi ini.
4. Serta teman-teman yang telah banyak membantu dalam menyelesaikan skripsi ini.

Penulis tahu bahwa laporan ini masih jauh dari kata sempurna, namun penulis berharap laporan ini dapat berguna dalam inovasi-inovasi teknologi khususnya dibidang teknologi informasi.

Depok, 13 Juli 2022

Penulis

DAFTAR ISI

DAFTAR GAMBAR

DAFTAR TABEL

BAB I

PENDAHULUAN

1.1 Latar Belakang

Dewasa ini, penggunaan dokumen dalam bentuk digital atau dokumen elektronik semakin sering digunakan yang berbanding lurus dengan upaya transformasi digital, terutama dokumen yang dikeluarkan dari pihak institusi resmi, khususnya institusi pendidikan. Dikarenakan dokumen tersebut bersifat digital, akan sangat mudah untuk dilakukan modifikasi oleh segala pihak yang memiliki dokumen tersebut, sehingga perlu dipastikan integritas data dari dokumen digital yang beredar dengan cara melakukan verifikasi dokumen digital tersebut dan penandatanganannya (Arisandi, Sukri dan Yusuf, 2020).

Unit Transformasi Digital Politeknik Negeri Jakarta atau UTD PNJ adalah unit yang bertanggung jawab dalam melakukan transformasi dari penggunaan dokumen fisik berupa kertas ke dokumen digital, dengan memperhatikan aspek keamanan suatu dokumen yang dikeluarkan. Namun, menurut ketua UTD PNJ “Hingga saat ini masih belum ada suatu alat untuk memverifikasi dokumen tersebut apakah benar-benar dikeluarkan oleh pihak PNJ”. Hal ini menjadi krusial karena dapat disalahgunakan untuk pemalsuan dan bahkan tindakan kriminal.

Tanda tangan digital adalah salah satu teknologi yang dapat digunakan untuk melakukan pembuktian secara matematis bahwa data tidak mengalami modifikasi secara ilegal, sehingga bisa digunakan sebagai salah satu solusi untuk melakukan pemeriksaan integritas data dokumen yang sah (Nugraha, 2017; Finandhita and Afrianto, 2018). Namun, kekuatan dan ketahanan dari tanda tangan digital sangat bergantung dengan metode kriptografi dan panjang kunci yang digunakan (Afrianto *et al.*, 2020).

Pada saat ini sedang dirancang aplikasi Digisign berbasis web yang memungkinkan pengguna melakukan pengesahan dokumen digital, penandatanganan dokumen digital, dan verifikasi dokumen digital pada ruang lingkup kampus PNJ. Aplikasi ini menggunakan 2 algoritma utama, yaitu algoritma *hash* yang sesuai dengan

Secure Hash Standard dan algoritma penandatanganan yang sesuai dengan *Digital Signature Standard*.

Hal terpenting dari aplikasi Digisign ini adalah memberikan keamanan pada dokumen yang telah disahkan dan memiliki performa yang efektif dan efisien. Dalam *Digital Signature Standard* yang diterbitkan NIST, terdapat beberapa algoritma penandatanganan yang memenuhi standar tersebut, seperti DSA, RSA, dan ECDSA. Setiap algoritma memiliki kelebihan dan kekurangannya masing-masing. DSA dinilai relatif lama dalam proses penandatanganannya (Muhtadin, 2017), tetapi dinilai lebih cepat dibandingkan 2 algoritma lain dan sudah cukup untuk penandatanganan digital untuk surat dinas (Somad, 2013). Sedangkan algoritma RSA memiliki ketahanan yang lebih kuat, tetapi untuk penandatanganan digital dinilai lambat dan membebani komputer dalam pemrosesannya (Prabowo dan Afrianto, 2017; Saepulrohman dan Ismangil, 2021). Untuk algoritma ECDSA memiliki proses yang lebih cepat dibandingkan dengan RSA dengan kekuatan kunci yang sama, namun terbilang sulit dalam proses implementasinya (Toradmalle *et al.*, 2018; Kavin dan Ganapathy, 2021).

Berdasarkan dari hasil penelitian-penelitian sebelumnya, penulis perlu mengkaji lebih lanjut dan mendalam algoritma seperti apa yang memenuhi kriteria yang sesuai untuk aplikasi Digisign UTD PNJ agar dapat berjalan secara optimal dan menjamin integritas dokumen yang telah disahkan oleh aplikasi Digisign tersebut.

1.2 Perumusan Masalah

Dilihat dari latar belakang tersebut adapun perumusan masalahnya sebagai berikut:

a. Bagaimana performa algoritma tanda tangan digital DSA, RSA, dan ECDSA pada aplikasi Digisign UTD PNJ ?
b. Apa saja parameter yang dapat menjadi tolak ukur dalam menentukan algoritma tanda tangan digital yang paling sesuai untuk aplikasi Digisign dalam menjamin integritas dokumen digital ?

1.3 Batasan Masalah

Adapun batasan masalah yang disebutkan bertujuan agar pembahasan dapat lebih terarah. Pembatasan masalah tersebut antara lain :

a. Algoritma tanda tangan digital yang dibandingkan adalah DSA, RSA dan ECDSA.

b. Bentuk dokumen digital yang diujikan hanya dokumen dengan ekstensi PDF.

1.4 Tujuan dan Manfaat

1.4.1 Tujuan

Adapun tujuan dari penelitian ini adalah menemukan algoritma tanda tangan digital yang sesuai untuk aplikasi Digisign UTD PNJ dengan performa terbaik saat mengeluarkan dokumen digital yang telah disahkan.

1.4.2 Manfaat

Adapun manfaat dari analisis integritas dokumen digital pada aplikasi Digisign UTD PNJ yaitu untuk mengidentifikasi, mengukur performa algoritma tanda tangan digital dalam memproses dokumen digital, dan memberikan umpan balik untuk pengembangan aplikasi Digisign UTD PNJ.

1.5 Sistematika Penulisan

Sistematika penulisan penelitian ini dibagi menjadi lima bab, yakni:

BAB I PENDAHULUAN

Bab ini berisi pendahuluan yang menjelaskan latar belakang, rumusan masalah, batasan masalah, tujuan penelitian, manfaat, relevansi dan sistematika penulisan.

BAB II TINJAUAN PUSTAKA

Definisi dan penjelasan pustaka yang dijadikan referensi dalam penelitian ini akan dijelaskan pada bab dua. Teori yang dipaparkan di antaranya mengenai keamanan informasi, integritas data, kriptografi, dan tanda tangan digital.

BAB III METODE PENELITIAN DAN PERANCANGAN

Bab ini menggambarkan uraian dan urutan pekerjaan yang akan dilakukan dalam penyusunan penelitian ini dan menjelaskan perancangan studi kasus yang diangkat, objek penelitian, perangkat yang dilakukan oleh penulis untuk mengumpulkan data kondisi kekinian, serta metode pengolahan data.

BAB IV HASIL DAN PEMBAHASAN

Bab ini berisi tentang bentuk hasil sistem yang dibangun dan analisa performa algoritma tanda tangan digital dalam proses mengeluarkan dokumen digital yang telah disahkan atau telah ditandatangani melalui aplikasi Digisign UTD PNJ.

BAB V KESIMPULAN DAN SARAN

Bab ini berisi tentang simpulan dari keseluruhan penelitian dan saran maupun rekomendasi terhadap penelitian ini untuk perbaikan ataupun penelitian lanjutan yang memiliki kesamaan dengan topik yang diangkat.

BAB V

PENUTUP

5.1 Kesimpulan

Berdasarkan hasil dari penelitian yang telah dilakukan, dapat ditarik kesimpulan bahwa:

1. Performa algoritma tandatangan digital pada aplikasi Digisign dilakukan dengan 3 skema yang berbeda dan dalam 3 proses yang berbeda. Algoritma ECDSA unggul dalam penggunaan waktu pada proses pembuatan pasangan kunci, proses penandatanganan dokumen, dan proses verifikasi dokumen. Sementara itu algoritma DSA unggul dalam parameter penggunaan CPU dari ketiga proses. Disisi lain algoritma RSA lebih unggul dalam penggunaan memori pada ketiga proses.
2. Parameter penggunaan waktu, penggunaan CPU, dan penggunaan memori, serta dilakukan evaluasi dengan pengujian *load testing* dapat menjadi parameter tolak ukur untuk menentukan algoritma tanda tangan digital yang paling sesuai untuk aplikasi Digisign.
3. Dari hasil penelitian ini, algoritma ECDSA menjadi algoritma tandatangan digital yang paling ideal dan memiliki performa paling stabil dibanding algoritma RSA dan DSA.

5.2 Saran

Berdasarkan hasil dari penelitian yang telah dilakukan, didapatkan beberapa hal yang perlu ditingkatkan, meliputi:

1. Pada penelitian ini, khususnya pada pengujian *load testing* diketahui bahwa spesifikasi perangkat keras yang digunakan hanya mampu menangani kurang dari 1000 pengguna, sehingga perlu dilakukan pengujian lebih lanjut dengan spesifikasi perangkat keras yang lebih tinggi.

2. Menambahkan subjek algoritma tanda tangan digital lain dengan panjang kunci yang lebih beragam dari penelitian yang telah dilakukan.
3. Melalukan teknik pengujian lain pada algoritma tanda tangan digital dalam aplikasi Digisign.

DAFTAR PUSTAKA

Afrianto, I. *et al.* (2020) 'Prototype of E-Document Application Based on Digital Signatures to Support Digital Document Authentication', *IOP Conference Series: Materials Science and Engineering*, 879(1). doi:10.1088/1757-899X/879/1/012042.

Arisandi, D., Sukri and Yusuf, M.B. (2020) 'Pemeriksaan Integritas Dokumen Dengan Digital Signature Algorithm', 4(1), pp. 1–6.

Cote, C. (2021) *What is Data Integrity and Why Does it Matter?*, *Harvard Business School Online*. Available at: https://online.hbs.edu/blog/post/what-is-data-integrity.

Dr. Abdel-Rahman Ismail (2014) 'Research Methodologies in Information Technology Research: A Comparative Study', *Implementation Science*, 39(1), pp. 1–15. Available at: http://dx.doi.org/10.1016/j.biochi.2015.03.025%0Ahttp://dx.doi.org/10.1038/nature10402%0Ahttp://dx.doi.org/10.1038/nature21059%0Ahttp://journal.stainkudus.ac.id/index.php/equilibrium/article/view/1268/1127%0Ahttp://dx.doi.org/10.1038/nrmicro2577%0Ahttp://.

Finandhita, A. and Afrianto, I. (2018) 'Development of E-Diploma System Model with Digital Signature Authentication', *IOP Conference Series: Materials Science and Engineering*, 407(1). doi:10.1088/1757-899X/407/1/012109.

Fox, P. (2022) *Central Processing Unit (CPU), Khan Academy*. Available at: https://www.khanacademy.org/computing/computers-and-internet/xcae6f4a7ff015e7d:computers/xcae6f4a7ff015e7d:computer-components/a/central-processing-unit-cpu (Accessed: 14 July 2022).

IBM (2021) 'Why is data security important?' Available at: https://www.ibm.com/topics/data-security.

IONOS (2020) *High CPU usage*. Available at:

https://www.ionos.com/digitalguide/server/know-how/cpu-usage/ (Accessed: 14 July 2022).

Kaspersky (2021) *What is Data Encryption ?*, *Kaspersky*. Available at: https://www.kaspersky.com/resource-center/definitions/encryption.

Kavin, B.P. and Ganapathy, S. (2021) 'A new digital signature algorithm for ensuring the data integrity in cloud using elliptic curves', *International Arab Journal of Information Technology*, 18(2), pp. 180–190. doi:10.34028/IAJIT/18/2/6.

Muhtadin, P. (2017) 'Implementasi Digital Signature Dalam Validasi Online Local E-Government Menggunakan Algoritme Rsa Dan Md5 Prasta Muhtadin'.

NIST (2015) 'Digital Signature Standard', *Safeguarding Critical E-Documents*, (July), pp. 221–221. doi:10.1002/9781119204909.app1.

Nugraha, P. (2017) 'Implementasi Digital Signature Pada File Text Dengan Menggunakan Algoritma Schnorr Berbasis Android'.

Oyinola, J.M. (2020) *Authentication In A Body Area Network (Ban) using Openssl*. Fredericton.

Panda Security (2020) *What is RAM*. Available at: https://www.pandasecurity.com/en/mediacenter/tips/how-to-free-up-ram/ (Accessed: 14 July 2022).

Prabowo, E.C. and Afrianto, I. (2017) 'Penerapan Digital Signature Dan Kriptografi Pada Otentikasi Sertifikat Tanah Digital', *Komputa : Jurnal Ilmiah Komputer dan Informatika*, 6(2), pp. 83–90. doi:10.34010/komputa.v6i2.2481.

Pradeep, S. and Sharma, Y.K. (2019) 'A Pragmatic Evaluation of Stress and Performance Testing Technologies for Web Based Applications', *Proceedings - 2019 Amity International Conference on Artificial Intelligence, AICAI 2019*, pp. 399–403. doi:10.1109/AICAI.2019.8701327.

Pramono, P.P. (2019) 'Pendeteksian Dini Tingkat Kemanan Informasi Berbasis Iso 27001 : 2013 Menggunakan Metode Ahp (Analytical Hierarchy Process)', *Cyber Security dan Forensik Digital*, 2(2), pp. 57–64. doi:10.14421/csecurity.2019.2.2.1480.

Ramesh, A. and Suruliandi, A. (2013) 'Performance analysis of encryption algorithms for information security', *Proceedings of IEEE International Conference on Circuit, Power and Computing Technologies, ICCPCT 2013*, pp. 840–844. doi:10.1109/ICCPCT.2013.6528957.

Saepulrohman, A. and Ismangil, A. (2021) 'Data integrity and security of digital signatures on electronic systems using the digital signature algorithm (DSA) Agus Ismangil', *International Journal of Electronics and Communications System*, 1(1), pp. 11–15. Available at: http://ejournal.radenintan.ac.id/index.php/IJECS/index://creativecommons.org/licenses/by-sa/4.0/.

Somad, W.A. (2013) 'Sistem tanda tangan digital Online Untuk Naskah Dinas Menggunakan Algoritma Dsa (Digital Signature Algorithm)'.

Sukumaran, J. (2020) *Syrupy System Resource Usage Profiler*. Available at: https://github.com/jeetsukumaran/Syrupy.

Svetlin Nakov, P. (2018) 'Practical Cryptography for Developers', in *SoftUni (Software University)*. Sofia, Bulagria: MIT License. Available at: https://cryptobook.nakov.com/digital-signatures (Accessed: 29 March 2022).

The openssl project (2021) *OpenSSL*. Available at: https://www.openssl.org/ (Accessed: 11 May 2022).

Toradmalle, D. *et al.* (2018) 'Prominence Of ECDSA Over RSA Digital Signature Algorithm', *2018 2nd International Conference on I-SMAC (IoT in Social, Mobile, Analytics and Cloud) (I-SMAC)I-SMAC (IoT in Social, Mobile, Analytics and Cloud) (I-SMAC), 2018 2nd International Conference on*, pp.

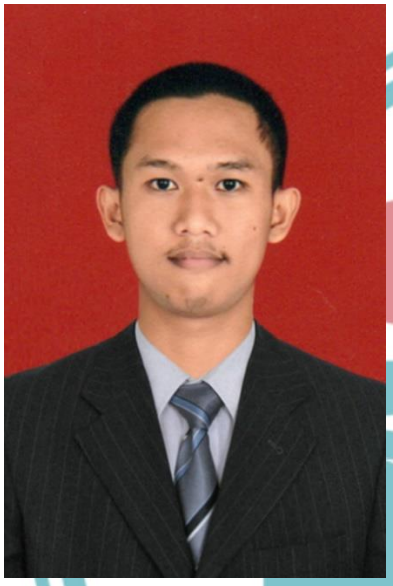
253–257. doi:10.1109/I-SMAC.2018.8653689.

Valvekens, M. (2022) *PyHanko*. Available at: https://github.com/MatthiasValvekens/pyHanko (Accessed: 11 May 2022).

LAMPIRAN

Lampiran 1 – Daftar Riwayat Hidup

Daftar Riwayat Hidup Penulis



Muhammad Dimas Yudha Adhi Pratama Jr.

Lahir di Argamakmur, 31 Agustus 2000. Lulus dari SDN 16 Seluma tahun, SMPN 20 Bengkulu, SMAN 1 Bangka Selatan. Saat ini sedang menempuh pendidikan Sarjana Terapan Program Studi Teknik Multimedia dan Jaringan – Sistem Keamanan Informasi Jurusan Teknik Informatika dan Komputer di Politeknik Negeri Jakarta

Lampiran 2 - *User Requirement*

KEMENTERIAN PENDIDIKAN, KEBUDAYAAN, RISET DAN TEKNOLOGI
POLITEKNIK NEGERI JAKARTA
Jl. Prof. DR. G.A. Siwabessy, Kampus UI, Depok 16425
Telp: (021)91274097, Fax: (021) 7863531, (021) 7270036 Hunting
Laman: https://www.pnj.ac.id, email: tik.pnj@gmail.com

SURAT PERNYATAAN

Yang bertandatangan di bawah ini perwakilan pihak UTD sebagai pihak pertama

Nama	: Mera Kartika Delimayanti , S.Si., M.T., Ph.D
NIP	: 197904282005012002
Jabatan	: Kepala Unit Transformasi Digital, Lektor Kepala

Memberikan *user requirement* untuk aplikasi DigiSign kepada pihak kedua:

Nama	: Muhammad Dimas Yudha Adhi Pratama Jr.
NIM	: 1807422025
Nama	: Hilmi Abdurrahman Fakhrudin
NIM	: 1807422008

Dengan isi *user requirement*, sebagai berikut:

1. Aplikasi berbasis web berbasis CodeIgniter/Laravel
2. Database yang digunakan adalah PostgreSQL
3. Dokumen yang dilakukan tandatangan digital adalah dokumen dengan bentuk PDF
4. Tandatangan yang diimplementasikan ke sebuah dokumen harus bisa lebih dari satu tandatangan

Kedua pihak diatas menyatakan setuju terhadap user requirement yang ditentukan.

Depok, 28 Juli 2022

Pihak 1	Pihak 2
Mera Kartika Delimayanti , S.Si., M.T., Ph.D	Muhammad Dimas Yudha Adhi Pratama Jr.
NIP. 197904282005012002	NIM. 1807422025

Lampiran 3 – Serah Terima Barang

KEMENTERIAN PENDIDIKAN, KEBUDAYAAN, RISET DAN TEKNOLOGI
POLITEKNIK NEGERI JAKARTA
Jl. Prof. DR. G.A. Siwabessy, Kampus UI, Depok 16425
Telp: (021)91274007, Fax: (021) 7863531, (021) 7270036 Hunting
Laman: https://www.pnj.ac.id, email: tik.pnj@gmail.com

BERITA ACARA SERAH TERIMA BARANG

Pada hari ini Kamis tanggal 28 Juli 2022 bertempat di Unit Transformasi Digital Politeknik Negeri Jakarta (UTD PNJ) telah terjadi penyerahan/penerimaan aplikasi DigiSign dari **PIHAK PERTAMA**:

Nama	: Muhammad Dimas Yudha Adhi Pratama Jr.
NIM	: 1807422025
Nama	: Hilmi Abdurrahman Fakhrudin
NIM	: 1807422008

Memberikan aplikasi DigiSign kepada **PIHAK KEDUA**, sebagai perwakilan pihak UTD PNJ:

Nama	: Muhammad Farhan Hanif, S. Tr. Kom
NIP	: 199912312021113312
Jabatan	: Staf Unit Transformasi Digital

Telah melakukan serah terima aplikasi DigiSign. Aplikasi ini diberikan oleh pihak pertama ke pihak kedua untuk digunakan untuk kebermanfaatan UTD dan lingkungan PNJ.
Demikian beirta acara ini dibuat untuk dapat dipergunakan sebagaimana mestinya.

Depok, 28 Juli 2022

Pihak 2	Pihak 1
Muhammad Farhan Hanif, S. Tr. Kom	Muhammad Dimas Yudha Adhi Pratama Jr.
NIP. 199912312021113312	NIM. 1807422025

Lampiran 4 – Data Hasil Pengujian

Skema: **none**

pada proses: pembuatan kunci

	DSA (2048)			RSA(2048)			ECDSA(256)		
Percobaan ke-n	**kecepatan (s)**	**CPU(%)**	**RAM(%)**	**kecepatan (s)**	**CPU(%)**	**RAM(%)**	**kecepatan (s)**	**CPU(%)**	**RAM(%)**
1	0,720177889	1	1,6	0,39349699	1,0	1,6	0,038043976	1	1,6
2	0,39527607	1	1,6	0,176780939	0,0	1,6	0,037375212	1	1,6
3	0,249484062	1	1,6	0,245392084	1,0	1,6	0,038659811	1	1,6
4	1,258862019	1,5	1,6	0,46775198	1,0	1,6	0,039572001	1	1,6
5	1,283180237	0	1,6	0,190605879	1,0	1,6	0,041780949	1	1,6
6	0,448888063	1	1,6	0,136781931	1,0	1,6	0,054794073	2	1,6
7	0,380090952	1	1,6	0,165223122	0,0	1,6	0,052760124	2	1,6
8	0,293491125	0	1,6	0,408558846	1,0	1,6	0,049395084	1	1,6
9	0,634973049	1	1,6	0,253695011	2,0	1,6	0,036956072	1	1,6
10	0,365586996	2	1,6	0,236176014	1,0	1,6	0,037636042	1	1,6
AVG	0,603001046	0,95	1,6	0,26744628	0,9	1,6	0,042697334	1,2	1,6

(Lanjutan)

Skema: *Hanya Text (1)*
pada proses: Pembuatan ttd

	DSA (2048)			RSA(2048)			ECDSA(256)		
Percobaan ke-n	**kecepatan (s)**	**CPU(%)**	**RAM(%)**	**kecepatan (s)**	**CPU(%)**	**RAM(%)**	**kecepatan (s)**	**CPU(%)**	**RAM(%)**
7	0,752851009	0	1,6	0,942784071	1	1,6	0,758874178	2	1,6
8	0,769820929	0	1,6	0,956191063	1	1,6	0,759085894	1	1,6
4	0,797912121	2	1,6	0,934975147	0	1,6	0,75927186	0	1,6
6	0,773308992	2	1,6	0,935301781	1	1,6	0,770231009	1	1,6
5	0,799028158	1	1,6	0,93727994	1	1,6	0,773448944	1	1,6
10	0,763890982	1	1,6	1,307873964	1,5	3,2	0,774819136	0	1,6
3	0,766093016	1	1,6	0,94119215	1	1,6	0,775182962	1	1,6
9	0,770813942	1	1,6	1,339250803	3	3,2	0,788018942	1	1,6
2	0,763462067	1	1,6	0,960141897	1	1,6	0,798892021	2	1,3
1	0,789427042	1	1,6	0,94410181	1	1,6	0,863104105	2	1,6
AVG	0,774660826	1	1,6	1,019909263	1,15	1,92	0,782092905	1,1	1,57

(Lanjutan)

Skema: *Hanya Text (1)*
pada proses: Verifikasi

	DSA (2048)			RSA(2048)			ECDSA(256)		
Percobaan ke-n	**kecepatan (s)**	**CPU(%)**	**RAM(%)**	**kecepatan (s)**	**CPU(%)**	**RAM(%)**	**kecepatan (s)**	**CPU(%)**	**RAM(%)**
1	0,567673922	1	1,6	0,775493145	1	1,6	0,579154015	1	1,6
2	0,539285183	1	1,6	0,562984943	1	1,6	0,558665037	1	1,6
3	0,527480841	0	1,6	0,552788973	1	1,6	0,531002045	1	1,6
4	0,528064013	1	1,6	0,519622087	1	1,6	0,544126987	1	1,6
5	0,539366007	1	1,6	0,544433117	1	1,6	0,543833017	1	1,6
6	0,544037104	0	1,6	0,562918901	2	1,6	0,549071074	1	1,6
7	0,536472797	1	1,6	0,552724838	1	1,6	0,552273989	1	1,6
8	0,543040037	1	1,6	0,544177055	1	1,6	0,599714994	1	1,6
9	0,731327057	1,5	1,6	0,550225019	2	1,6	0,538659811	0	1,6
10	0,74947691	1	1,6	0,555149078	0	1,6	0,544983864	2	1,6
AVG	0,580622387	0,85	1,6	0,572051716	1,1	1,6	0,554148483	1	1,6

(Lanjutan)

Skema: *Campuran*
pada proses: *Gambar (2)* Pembuatan ttd

	DSA (2048)			RSA(2048)			ECDSA(256)		
Percobaan ke-n	**kecepatan (s)**	**CPU(%)**	**RAM(%)**	**kecepatan (s)**	**CPU(%)**	**RAM(%)**	**kecepatan (s)**	**CPU(%)**	**RAM(%)**
1	0,904968977	2	1,5	0,704382181	2	1,6	0,91229105	1	1,6
2	0,92882514	1	1,6	1,095671177	1,5	3,2	0,919603109	1	1,6
3	0,936150074	1	1,6	1,118997097	1,5	3,2	0,944227934	2	1,6
4	0,940124989	1	1,6	1,11946702	1,5	3,2	0,971886158	1	1,6
5	0,944844961	1	1,6	1,11977911	1,5	3,2	0,978307009	1	1,6
6	0,948312044	0	1,6	1,122169018	1,5	3,2	0,998042107	1	1,6
7	0,995463133	1	1,6	1,130771875	1,5	3,2	1,003211975	1	1,6
8	1,05212307	1,5	3,2	1,132941008	1	3,2	1,009623051	2	1,6
9	1,232100964	2	3,2	1,136716127	1,5	3,2	1,02393508	1	3
10	1,519422054	2	3,2	1,272650957	1,5	3,2	1,104804039	1,5	3,2
AVG	1,040233541	1,25	2,07	1,095354557	1,5	3,04	0,986593151	1,25	1,9

(Lanjutan)

Skema: *Campuran Gambar (2)*
pada proses: Verifikasi

	DSA (2048)			RSA(2048)			ECDSA(256)		
Percobaan ke-n	**kecepatan (s)**	**CPU(%)**	**RAM(%)**	**kecepatan (s)**	**CPU(%)**	**RAM(%)**	**kecepatan (s)**	**CPU(%)**	**RAM(%)**
1	0,943161011	1	1,6	0,723173857	1	1,5	0,767415047	2	1,6
2	0,80548501	2	1,6	0,686320066	1	1,6	0,76493907	2	1,3
3	0,679335117	1	1,6	0,738094091	1	1,6	0,756788015	1	1,6
4	0,737313986	1	1,5	0,818843842	1	1,6	0,761165857	1	1,6
5	0,70309186	1	1,6	0,782119036	0	1,6	0,710264921	1	1,6
6	0,725275993	1	1,6	0,79163003	2	1,5	0,711076975	1	1,6
7	0,726840973	1	1,6	0,793395042	0	1,6	0,698557138	1	1,6
8	1,004837036	0	1,6	0,750545979	1	1,6	0,748820066	0	1,6
9	0,740381002	1	1,6	0,731799841	3	1,1	0,714202881	0	1,6
10	0,651048899	1	1,6	0,790036917	2	1,6	0,623108864	1	1,6
AVG	0,771677089	1	1,59	0,76059587	1,2	1,53	0,725633883	1	1,57

(Lanjutan)

Skema: *incremental update (3)*
pada proses: Pembuatan ttd

	DSA (2048)			RSA(2048)			ECDSA(256)		
Percobaan ke-n	**kecepatan (s)**	**CPU(%)**	**RAM(%)**	**kecepatan (s)**	**CPU(%)**	**RAM(%)**	**kecepatan (s)**	**CPU(%)**	**RAM(%)**
1	0,769917011	0	1,6	0,931828022	1	1,6	0,764050961	2	1,5
2	0,773511171	1	1,6	0,93759799	0	1,6	0,76510191	1	1,6
3	0,773874044	1	1,6	0,942924976	1	1,6	0,767498016	1	1,6
4	0,78488493	1	1,6	0,944394827	1	1,6	0,777648926	1	1,6
5	0,788284063	1	1,6	0,950521946	1	1,6	0,7978971	0	1,6
6	0,7883811	1	1,6	0,955535889	1	1,6	0,851912022	1	1,6
7	0,788472891	1	1,6	0,962792873	1	1,6	0,996753931	1	1,6
8	0,794603825	0	1,6	0,975279093	1	1,6	1,011538982	2	1,6
9	0,813542843	1	1,6	0,988014936	1	1,6	1,049519062	1,5	3,2
10	0,917294025	1	1,6	0,990113974	1	1,6	1,369845867	3	3,2
AVG	0,79927659	0,8	1,6	0,957900453	0,9	1,6	0,915176678	1,35	1,91

(Lanjutan)

Skema: *incremental update (3)*
pada proses: Verifikasi

	DSA (2048)			RSA(2048)			ECDSA(256)		
Percobaan ke-n	**kecepatan (s)**	**CPU(%)**	**RAM(%)**	**kecepatan (s)**	**CPU(%)**	**RAM(%)**	**kecepatan (s)**	**CPU(%)**	**RAM(%)**
1	0,629390955	2	1,6	0,618134022	1	1,6	0,579081059	1	1,6
2	0,642303944	1	1,6	0,621454	2	1,6	0,594385862	1	1,6
3	0,623746872	1	1,6	0,6370399	1	1,6	0,591111898	1	1,5
4	0,627647877	1	1,6	0,635203123	1	1,6	0,701158047	2	1,6
5	0,595148087	1	1,6	0,639147997	1	1,6	0,604074955	2	1,6
6	0,615674973	1	1,6	0,632057905	1	1,6	1,001462936	1	1,6
7	0,821267128	1	1,6	0,626697063	0	1,6	0,607722998	1	1,6
8	0,626378775	2	1,6	0,626843929	1	1,6	0,618963003	1	1,6
9	0,630823135	0	1,6	0,631109953	1	1,6	0,596469164	1	1,6
10	0,635562897	1	1,6	0,622711897	2	1,5	0,606172085	1	1,6
AVG	0,644794464	1,1	1,6	0,629039979	1,1	1,59	0,650060201	1,2	1,59

Lampiran – Skrip Penandatanganan Dokumen

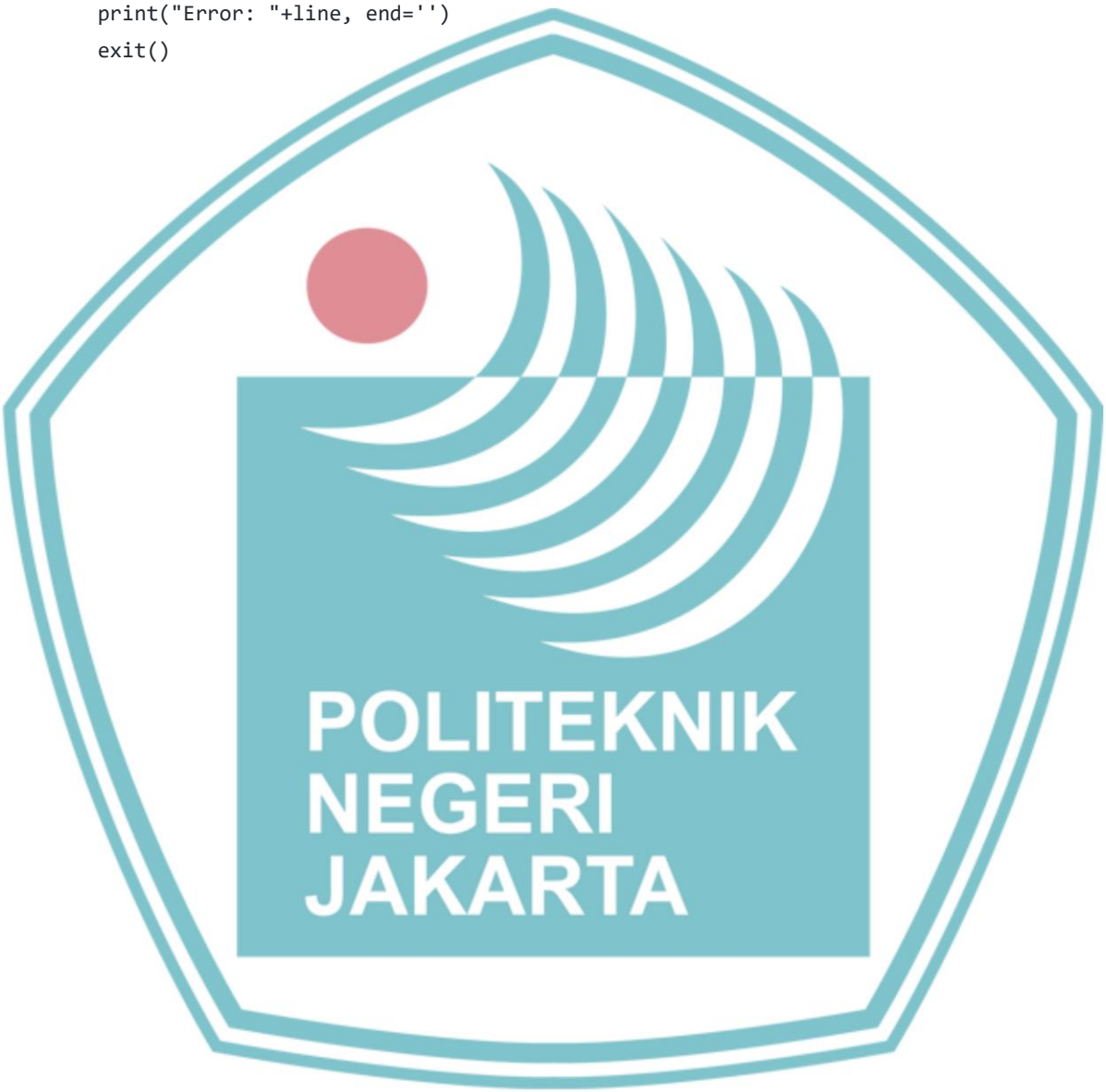
```
Import os
import traceback
import datetime
from pyhanko import stamp
from pyhanko.pdf_utils import text, layout, images
from pyhanko.pdf_utils.font import opentype
from pyhanko.pdf_utils.incremental_writer import
IncrementalPdfFileWriter
from pyhanko.sign import signers
from pyhanko.sign import fields
from urllib.request import urlopen
import json
import argparse

parser = argparse.ArgumentParser()
parser.add_argument('-t', '--token', help="Signature Token")

args = parser.parse_args()

if not args.token:
    print("Error: Token Needed", end='')
    exit()

token = args.token
url = "http://localhost/ApiAccess/getSignatureData/"+token

try:
    response = urlopen(url)
    data_json = json.loads(response.read())
except Exception as e:
    print("Error: Failed to Contact API Server", end='')
    print(e, end='')
    exit()


if data_json is None:
    print("Error: Incorrect Token", end='')
    exit()


# Signer Data
```

```
try:
    signerPrivKey = data_json['userData']['priv_path']
    signerPubKey = data_json['userData']['pub_path']

    # DocumentData
    documentName = data_json['docData']['file_name']
    signatureName = data_json['signData']['signature_field']
    qrCodeName = data_json['signData']['signature_token']+'.png'
    # page signature
    documentPage = int(data_json['signData']['signature_page_pos'])
    # coordinate signature
    bottom = int(data_json['signData']['signature_pos_yb'])
    left = int(data_json['signData']['signature_pos_xb'])
    top = int(data_json['signData']['signature_pos_yt'])
    right = int(data_json['signData']['signature_pos_xt'])
except Exception as e:
    print('Error: Failed to retrieve data! ({})'.format(e), end='')
    exit()

# Define Project Path
mainPath = os.path.dirname(os.path.dirname(__file__))

# Define CA Path
caKeyPath = os.path.realpath(mainPath+'/signCore/cert/ca/cacert.pem')

# Define Signer Certificate Path
privateKeyPath = os.path.realpath(
    mainPath+'/signCore/cert/usrpriv/'+signerPrivKey)
publicKeyPath = os.path.realpath(
    mainPath+'/signCore/cert/usrpub/'+signerPubKey)

# Define Document Path
documentPath = os.path.realpath(
    mainPath+'/writable/uploads/signedDocs/'+documentName)

# Define QR Code Path
qrCodePath =
os.path.realpath(mainPath+'/writable/uploads/qrcodes/'+qrCodeName)

# Check if all file Exist
if not os.path.exists(privateKeyPath):
    print("Error: Private Key Not Found!", end='')
    exit()
```

```
if not os.path.exists(publicKeyPath):
    print("Error: Public Key Not Found!", end='')
    exit()

if not os.path.exists(caKeyPath):
    print("Error: CA Not Found!", end='')
    exit()

if not os.path.exists(documentPath):
    print('Error: Document Not Found!', end='')
    exit()

if not os.path.exists(qrCodePath):
    print('Error: QR Code Not Found!', end='')
    exit()


# Load Certificate File
try:
    signerPerson = signers.SimpleSigner.load(
        key_file=privateKeyPath,
        cert_file=publicKeyPath,
        ca_chain_files=(caKeyPath,),
    )
except Exception as e:
    print("Error: Failed to load signature data! ({})".format(e),
end='')

try:
    # Load File
    with open(documentPath, 'rb+') as inf:

        # Add incremental update
        w = IncrementalPdfFileWriter(inf, strict=False)
        # Create Signature Field
        fields.append_signature_field(
            w, sig_field_spec=fields.SigFieldSpec(
                signatureName, on_page=documentPage, box=(
                    left, bottom, right, top),
                     doc_mdp_update_value=fields.MDPPerm.ANNOTATE
            )
        )
```

```
        pdf_signer = signers.pdf_signer.PdfSigner(

        # Select Signature Fielad
            signers.PdfSignatureMetadata(field_name=signatureName),

            # Load Certificate Data
            signer=signerPerson,

            # Define Signature Style
            stamp_style=stamp.TextStampStyle(

                # Set Signature Text, set empty for no text
                stamp_text='',

                # Set Background Image (QR Code)
                background=images.PdfImage(qrCodePath),

                # Set Backround (QR Code) Opacity to Max
                background_opacity=1,

                # Remove Border
                border_width=0,

                # Set Backgorund position to center and fit the image
to field
                background_layout=layout.SimpleBoxLayoutRule(
                    x_align=layout.AxisAlignment.ALIGN_MID,
                    y_align=layout.AxisAlignment.ALIGN_MID,
                    margins=layout.Margins.uniform(0)
                )
            ),
        )
        # Sign the PDF with formatted style to the same file that
loaded before
        pdf_signer.sign_pdf(w, in_place=True)
        print("success", end='')
        currTime = datetime.datetime.now()
        line = "[{}] {} | {}".format(str(currTime), 'success', token)
        file1 =  open(os.path.dirname(__file__)+'/log.log', 'a')
        file1.write(line+"\n")
        file1.close()

except Exception as e:
```

```
currTime = datetime.datetime.now()
line = "[{}] {} | {}".format(str(currTime), str(e), token)
file1 =  open(os.path.dirname(__file__)+'/log.log', 'a')
file1.write(line+"\n")
file1.close()
print("Error: "+line, end='')
exit()
```



Lampiran – Skrip Verifikasi Dokumen

```
#!/usr/bin/python
from cryptography import x509
import sys, json, base64, os
from pyhanko.sign.general import load_cert_from_pemder
from pyhanko_certvalidator import ValidationContext
from pyhanko.pdf_utils.reader import PdfFileReader
from pyhanko.sign.validation import validate_pdf_signature
from asn1crypto import x509

# Define Project Path
mainPath=os.path.dirname(os.path.dirname(__file__))

#get file uploaded and define doc path
documentName = sys.argv[1]
# documentName = "SignDouble.pdf"
# documentName = "1655868991_7e463252188e186a9370pdf.pdf"
# documentName = "logbook-signed (1).pdf" #untrusted
# documentName = "1654443615_eaf89dd260734ef2353a.pdf" #undetected

# documentPath=os.path.realpath(mainPath+'/signCore/testDocs/'+documentName)
documentPath=os.path.realpath(mainPath+'/writable/uploads/verifyDocs/'+documentName)

# Define CA Path
caKeyPath=os.path.realpath(mainPath+'/signCore/cert/ca/cacert.pem')

#load CA
root_cert = load_cert_from_pemder(caKeyPath)
vc = ValidationContext(trust_roots=[root_cert])

def str2Dict(str):
  output =  dict((x.strip(), y.strip())
              for x, y in (element.split(':')
                  for element in str.split(', ')))
  return output

last_coverage = None

final_judgement = "VALID"

signData = {}
```

```
signData['signature'] = {}

with open(documentPath, 'rb') as doc:
      r = PdfFileReader(doc,strict=False)
      i = 0
      thisdict = None
      while i < len(r.embedded_signatures):
        sig = r.embedded_signatures[i] #list index out of range
        fieldName = sig.field_name

        status = validate_pdf_signature(sig, vc, skip_diff=True)
        id = "Unparsable"
        try:
          data: x509.Certificate = status.signing_cert
          id = data.subject.human_friendly
          userData = dict(x.split(": ") for x in id.split(", "))
        except Exception:
          pass


        last_coverage = status.coverage._name_
        trusted = status.trusted
        valid = status.valid


        currData = {
          'fieldName'     : sig.field_name,
          'signTime'      : sig.self_reported_timestamp.strftime("%m/%d/%Y %H:%M:%S %z"),
          'signerID'      : userData,
          'signCoverage'  : status.coverage._name_,
          'signerTrusted' : status.trusted,
          'signatureValid': status.valid,
          'digestAlgo'    : sig.md_algorithm
        }

        signData['signature'][i] = currData

        if final_judgement == "VALID":
          if trusted == False or valid == False:
            final_judgement = "NOT VALID"

        i+=1
```

```
if final_judgement == "VALID" and last_coverage != "ENTIRE_FILE":
  final_judgement = "NOT VALID"

signData['final_judgement'] = final_judgement
```