Implementasi dan Analisis Penggunaan Self-Signed Digital Signature dengan Kakas Sumber Terbuka

Akbar Maulana Ridho - 13521093
Program Studi Sistem dan Teknologi Informasi
Sekolah Teknik Elektro dan Informatika
Institut Teknologi Bandung, Jalan Ganesha 10 Bandung
E-mail: 13521093@std.stei.itb.ac.id

***Abstract*—This electronic document is a "live" template and already defines the components of your paper [title, text, heads, etc.] in its style sheet. **CRITICAL: Do Not Use Symbols, Special Characters, or Math in Paper Title or Abstract.* (*Abstract*)**

Keywords—component; formatting; style; styling; insert (key words)

I. PENDAHULUAN

Di tengah pesatnya digitalisasi dokumen, terdapat kesalahpahaman mendasar mengenai keamanan tanda tangan. Masih ada individu dan organisasi yang menganggap bahwa menempelkan gambar pindaian tanda tangan basah ke dalam dokumen digital (*digitized signature*) sudah cukup aman. Padahal, metode ini rentan terhadap pemalsuan karena pada hakikatnya tanda tangan tersebut hanyalah sebuah gambar yang dapat disalin dan ditempel oleh siapa saja.

Tanda tangan digital (*digital signature*) menawarkan keamanan yang jauh lebih baik. Pendekatan ini tidak hanya menempelkan gambar, tetapi juga menggunakan prinsip kriptografi untuk mengikat identitas pemilik ke dalam dokumen secara matematis. Pendekatan ini menjamin keaslian (*authenticity*), integritas, dan tidak dapat disangkal (*non-refudiation*).

Untuk memperoleh tanda tangan digital yang diakui secara luas, diperlukan sertifikat dari otoritas sertificat (Certificate Authority/ CA) komersial, yang seringkali membutuhkan biaya dna proses verifikasi. Meskipun begitu, terdapat alternatif yang lebih fleksibel dan ekonomis, yaitu *self-signed digital signature*. Pendekatan ini dapat digunakan untuk kebutuhan internal atau lingkungan terkontrol. Sertifikat ini dapat dibuat dan divalidasi oleh entitas yang berkaitan dengan menggunakan kakas seperti OpenSSL. Pendekatan ini memungkinkan penerapan tanda tangan digital tanpa bergantung pada pihak ketika yang berbayar.

Kebutuhan akan representasi visual tanda tangan (*digitized signature*) masih tinggi, sementara pemahaman akan keamanan tanda tangan masih perlu ditingkatkan. Oleh karena itu, penelitian ini akan mengeksplorasi sebuah alur kerja untuk menggabungkan metode *digitized signature* dan tanda tangan digital. Alur yang ingin dibuat adalah penyisipan *digitized signature* ke dalam dokumen, yang kemudian akan diamankan lebih lanjut dengan menggunakan *self-signed digital signature*. Dengan demikian, kebutuhan pengguna akan aspek visual tetap terpenuhi, sekaligus memperkenalkan lapisan keamanan kriptografis yang esensial untuk menjamin keaslian dan integritas dokumen di lingkungan digital.

II. DASAR TEORI

Dasar teori ini membahas konsep-konsep fundamental yang mendasari implementasi tanda tangan digital, mulai dari layanan keamanan dalam kriptografi, evolusi dari tanda tangan konvensional, hingga metode dan algoritma spesifik yang digunakan.

A. Kriptografi sebagai Jaminan Keamanan

Kriptografi menyediakan serangkaian layanan keamanan untuk melindungi data dan komunikasi. Berikut adalah jaminan yang disediakan dalam kriptografi:

1. Kerahasiaan (*confidentiality*). Menjaga agar isi pesan tidak dapat diketahui oleh pihak yang tidak berwenang melalui proses enkripsi dan deskripsi.
2. Integritas data (*data integrity*). Memastikan bahwa pesan masih asli dan belum diubah selama pengiriman. Jaminan ini dicapai dengan menggunakan fungsi hash dan *message authentication code* (MAC).
3. Otentikasi (*authentication*). Memverifikasi identitas pengirim pesan serta keaslian pesan itu sendiri. Jaminan ini didukung oleh penggunaan MAC atau tanda tangan digital.
4. Anti-penyangkalan (*non-refudiation*). Mencegah pengirim menyangkal bahwa mereka telah mengirim sebuah pesan. Tanda tangan digital merupakan mekanisme utama untuk menyediakan layanan ini.

Konteks makalah ini berfokus pada jaminan otentikasi dan anti-penyangkalan yang dijamin oleh tanda tangan digital.

B. Digitalisasi Tanda Tangan

Secara sejarah, tanda tangan basah pada dokumen cetak telah lama digunakan untuk tujuan otentikasi. Tanda tangan konvensional ini memiliki karakteristik penting:

1. Merupakan bukti yang otentik.
2. Tidak dapat dilupakan oleh pemiliknya.

3. Tidak dapat dipindahkan untuk digunakan pada dokumen lain.
4. Dokumen yang sudah ditanda tangani tidak dapat diubah isinya (*integrity*).
5. Pemilik tanda tangan tidak dapat menyangkalnya.

Karakteristik ini kemudian diadopsi ke dalam ranah digital untuk mengotentikasi data elektronik seperti pesan atau dokumen, Mekanisme ini disebut sebagai tanda tangan digital (*digital signature*).

Meskipun begitu, penting untuk ditekankan bahwa tanda tangan digital dalam konteks kriptografi bukanlah gambar tanda tangan yang dipindai atau difoto (*digitized signature*).

C. Konsep Tanda Tangan Digital

Tanda tangan digital merupakan sebuah nilai kriptografi yang untuk, yang nilainya bergantung pada dua buah elemen: isi pesan yang ditanda tangani dan sebuah kunci rahasia milik pengirim. Berbeda dengan tanda tangan konvensional yang selalu sama, tanda tangan digital akan selalu berbeda untuk setiap pesan yang berbeda atau kunci yang berbeda.

Sebuah tanda tangan harus memenuhi syarat berikut:

1. Berupa rangkaian bit yang bergantung pada pesan yang ditanda tangani.
2. Harus menggunakan kunci privat pengirim untuk mencegah pemalsuan dan penyangkalan.
3. Proses pembangkitan tanda tangan harus relatif mudah bagi pengirim.
4. Proses untuk mengenali dan memverifikasi tanda tangan harus relatif mudah bagi penerima.
5. Secara komputasi, harus sangat sulit (hampir tidak mungkin) untuk memalsukan tanda tangan, baik dengan membuat pesan baru yang cocok dengan tanda tangan yang ada, maupun membuat tanda tangan palsu untuk sebuah pesan.
6. Penyimpanan salinan tanda tangan digital harus praktis dan mudah dilakukan.

Sebagaimana dibahas sebelumnya, terdapat dua proses utama dalam tanda tangan digital:

1. Penandatanganan (*signing*): Proses pengirim menggunakan kunci privatnya untuk membuat tanda tangan pada sebuah pesan.
2. Verifikasi: Proses penerima menggunakan kunci publik pengirim untuk memeriksa keabsahan tanda tangan yang diterima.

D. Implementasi Tanda Tangan Digital

Kombinasi fungsi hash dan kriptografi kunci publik merupakan metode yang paling umum digunakan, terutama ketika kerahasiaan pesan tidak diperlukan, tetapi otentikasi dan integritas tetap menjadi prioritas utama. Proses ini efisien karena tidak mengenkripsi keseluruhan pesan, tetapi hanya nilai hashnya saja.

1) Proses penanda tanganan (dari sisi pengirim)

- Sebuah fungsi hash diaplikasikan pada keseluruhan pesan (M) untuk menghasilkan sebuah nilai hash atau message digest (h) dengan panjang tetap. $\mathrm{h} = \mathrm{H(M)}$.
- Nilai hash (h) inilah yang kemudian dienkripsi menggunakan kunci privat (SK) pengirim. Hasil enkripsi ini adalah tanda tangan digital (S). $\mathrm{S} = E_{SK}\ (\mathrm{h})$.
- Pengirim kemudian mengirimkan pesan asli (M) bersama dengan tanda tangan digital (S) kepada penerima.

2) Proses verifikasi (dari sisi penerima)

- Penerima memisahkan pesan (M) dari tanda tangan (S).
- Penerima mendekripsi tanda tangan (S) menggunakan kunci publik (PK) milik pengirim untuk mendapatkan kembali nilai hash asli, sebut saja h'. $h' = D_{pk}(S)$.
- Penerima kemudian menghitung sendiri nilai hash dari pesan (M) yang ia terima menggunakan fungsi hash yang sama, sebut saja h. $\mathrm{h} = \mathrm{H(M)}$.
- Terakhir, penerima membandingkan kedua nilai hash tersebut. Jika kedua nilai hash sama, maka tanda tangan dianggap valid, yang membuktikan bahwa pesan tersebut otentik, tidak berubah, dan benar-benar berasal dari pengirim tersebut. Jika tidak sama, maka tanda tangan tidak otentik.

E. Algoritma Tanda Tangan Digital

Beberapa algoritma kriptografi dapat digunakan untuk tanda tangan digital. Dua hal paling populer adalah RSA dan ElGamal.

- RSA. Pada RSA, proses enkripsi dan dekripsi secara matematis identik, sehingga algoritma yang sama digunakan untuk enkripsi/dekripsi pesan maupun untuk penandatanganan/verifikasi.
- ElGamal. Berbeda dengan RSA, algoritma ElGamal untuk enkripsi tidak sama dengan algoritma untuk tanda tangan.
- Digital Signature Algorithm (DSA). Algoritma ini dirancang khusus untuk tujuan penanda tanganan yang menjadi bagian dari Digital Signature Standard (DSS).

III. IMPLEMENTASI

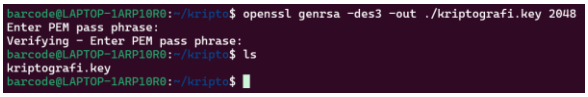
Terdapat berbagai macam solusi atau kakas sumber terbuka (*open source*) yang mendukung *digitized signature* dan tanda tangan digital. Meskipun begitu, alur kombinasi antara *digitized signature* dengan tanda tangan digital adalah sebagai berikut: pembangkitan sertifikat (*self-signed certificate*) termasuk kunci publik dan privat, penandatanganan dokumen dengan *digitized signature*, penandatanganan dokumen dengan tanda tangan

digital, lalu verifikasi dokumen yang sudah diberi tanda tangan digital.

A. Pembangkitan *Self-Signed Certificate*

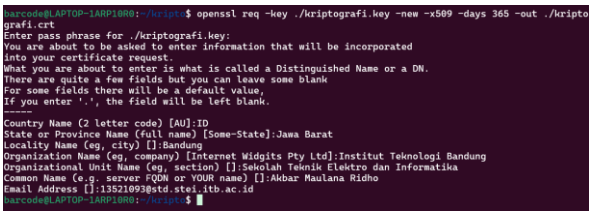
Salah satu cara untuk membangkitkan *self-signed certificate* serta kunci privat dan publik adalah dengan menggunakan kakas bernama openssl. Proses pembangkitan ini dapat disesuaikan dengen preferensi pengguna, tetapi salah satu cara pembangkitan adalah sebagai berikut:

1. Jalankan perintah
   openssl genrsa -des3 -out ./kriptografi.key 2048



   Perintah ini menginstruksikan OpenSSL untuk menghasilkan kunci dengan algoritma RSA yang memiliki panjang 2048 bit. Selain itu, opsi des3 memastikan bahwa kunci privat yang dihasilkan dienkripsi dan dilindungi oleh sebuah *passphrase*.

2. Jalankan perintah
   openssl req -key ./kriptografi.key -new -x509 -days 365 -out ./kriptografi.crt

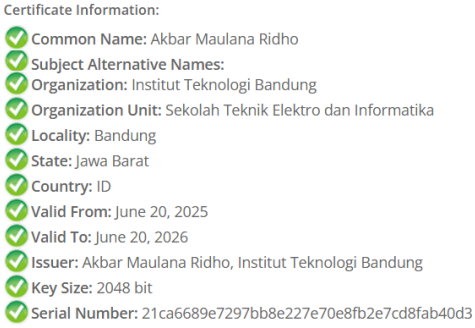


   Perintah ini menginstruksikan OpenSSL untuk membuat identitas publik sebagai dasar. Perintah ini menghasilkan sebuah sertifikat digital X.509. Opsi -x509 membuat sertifikat ini *self-signed*. Luaran dari perintah ini adalah kunci publik dan informasi indentitas yang terverifikasi. Saat perintah ini dijalankan, pengguna akan diminta untuk memasukkan berbagai informasi, seperti nama organisasi, nama pemilik, surel, dan lain-lain.

3. Jalankan perintah
   openssl pkcs12 -inkey ./kriptografi.key -in ./kriptografi.crt -export -out ./kriptografi.pfx

   Perintah ini berfungsi untuk mengemas kunci privat dan sertifikat menjadi satu berkas agar mudah digunakan. Perintah ini mengambil kunci privat dan sertifikat publiknya, lalu menggabungkannya menjadi satu berkas arsip portabel yang aman dengan format PKCS#12. Berkas ini dilindungi dengan kata sandi ekspor, sehingga data tetap aman.

Apabila sertifikat tersebut dibuka, akan muncul informasi yang berkaitan dengan pembuat sertifikat.



Sertifikat ini juga sudah berisi kunci publik yang dapat digunakan untuk memverifikasi tanda tangan digital.

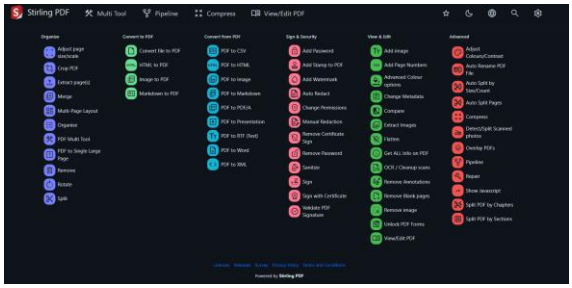
Sertifikat publik ini perlu didistribusikan kepada pihak-pihak yang akan menjadi penerima dokumen yang telah ditanda-tangani secara digital dengan kunci privat ini. Hal ini diperlukan karena sertifikat yang dibuat sendiri (*self-signed certificate*) ini *by default* tidak dipercaya oleh banyak sistem. Oleh karena itu, pengguna harus menambahkan sendiri sertifikatnya agar tanda tangan dapat diverifikasi oleh penerima.

B. Penandatanganan dengan *Digitized Signature*

Tahap ini merupakan tahap yang sudah cukup familiar dan umum dilakukan oleh kebanyakan orang. Selain itu, sudah terdapat banyak kakas yang mendukung operasi ini, mulai dari Microsoft Edge, Adobe Acrobat Reader, Foxit PDF Reader, ILovePDF, dan lain-lain. Contoh ini akan mendemonstraksikan penggunaan StilingPDF untuk menambahkan *digitized signature* pada sebuah dokumen.

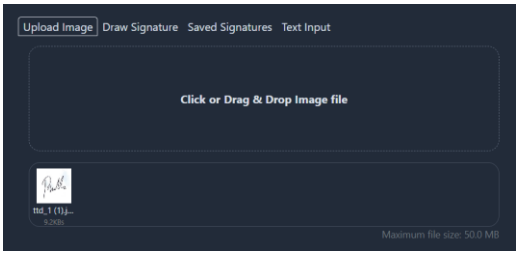
StilingPDF merupakan sebuah kakas sumber terbuka yang dapat di-*hosting* secara mandiri (*self-host*). Situs proyeknya dapat diakses pada laman GitHub berikut. Meksipun begitu, contoh ini akan menunjukkan penandatanganan dengan StilingPDF versi web agar lebih mudah diakses.

1. Buka laman StilingPDF. Cari kakas **sign** lalu klik tombol tersebut untuk melakukan navigasi.

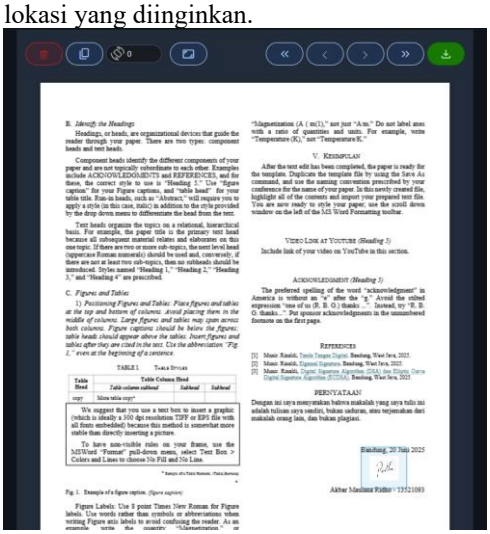


2. Tambahkan berkas yang ingin ditandatangani.

3. Tambahkan berkas *digitized signature* atau gambar tanda tangan bila diperlukan. Pemilihan *digitized signature* diserahkan pada preferensi pengguna.



4. Navigasikan tampilan peletakan *digitized signature* dan atur sehingga letak tanda tangan sesuai dengan lokasi yang diinginkan.



5. Selanjutnya, tekan tombol unduh untuk mengunduh dokumen yang sudah ditandatangani.

Perlu diingat bahwa proses juga dapat dilakukan dengan kakas lain yang sudah umum atau familiar digunakan oleh pengguna. Proses selanjutnya adalah penandatanganan dokumen dengan tanda tangan digital. Pada makalah ini, terdapat dua kakas yang akan dibahas/ digunakan sebagai contoh.

Contoh pertama adalah penggunaan StirlingPDF untuk proses penandatanganan tanda tangan digital. Kakas ini digunakan sebagai contoh solusi sumber terbuka berbasiskan web/ kakas dengan GUI.

Contoh kedua adalah penggunaan ypHanko untuk proses penandatanganan tanda tangan digital. Kakas ini digunakan sebagai contoh solusi sumber terbuka berbasiskan CLI.

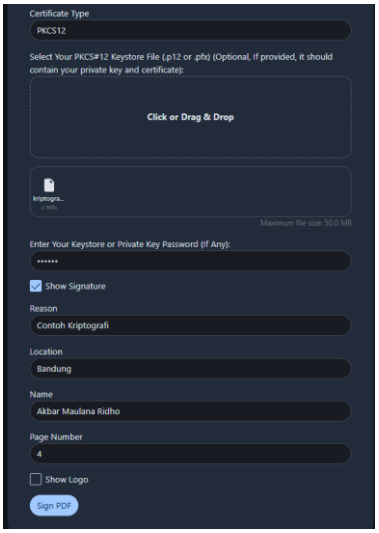
C. Tanda Tangan Digital dengan StirlingPDF

Berikut adalah tahapan yang diperlukan untuk melakukan proses tanda tangan digital dengan StirlingPDF.

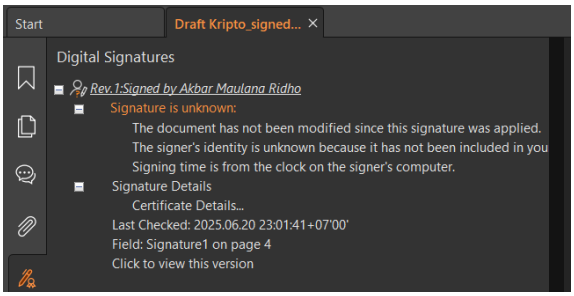
1. Buka laman StilingPDF. Cari kakas **Sign with Certificate** lalu klik tombol tersebut untuk melakukan navigasi.

2. Tambahkan berkas yang ingin ditandatangani.

3. Pada menu tipe kunci, pilih PKCS#12, lalu sertakan berkas dengan ekstensi .pfx yang sebelumnya sudah dibuat. Masukkan password dan informasi tambahan apabila diperlukan sebagaimana ditunjukkan pada contoh berikut.



4. Tekan tombol **Sign PDF** untuk menandatangani dokumen. Selanjutnya, berkas akan otomatis diunduh.
5. Saat dokumen dibuka, akan muncul peringatan bahwa sertifikat yang disertakan tidak dapat diverifikasi. Hal ini merupakan kelemahan dari *self-signed certificate*. Agar aplikasi dapat memverifikasi dokumen, penerima harus menyertakan sertifikat pengirim ke dalam *trust store*.



D. Tanda Tangan Digital dengan pyHanko

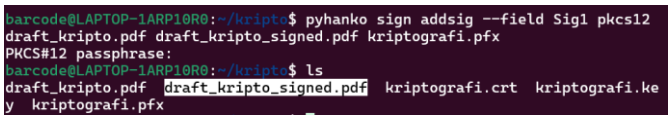
Sebelum menggunakan pyHanko, jalankan perintah berikut untuk menginstall kakas tersebut ke dalam perangkat pengguna.

```
pip install 'pyHanko[pkcs11,image-support,opentype,qr]'
pyhanko-cli
```

Selanjutnya, berikut adalah perintah untuk menandatangani dokumen secara digital dengan pyHanko:

```
pyhanko sign addsig --field Sig1 pkcs12 draft_kripto.pdf
draft_kripto_signed.pdf kriptografi.pfx
```

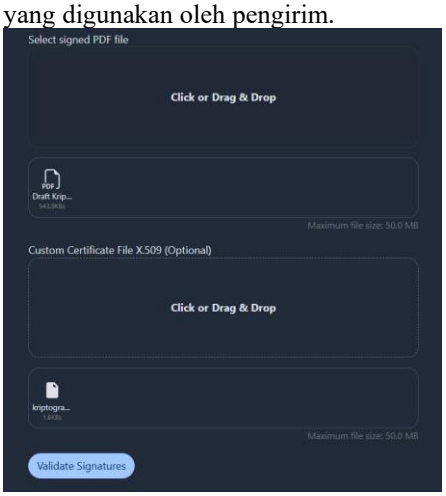
Perintah tersebut menerima nama masuk dan luaran dokumen dan juga berkas kunci yang sudah dibuat sebelumnya. Saat proses penandatanganan, pengguna akan diminta untuk memasukkan kata sandi berkas agar berkas dapat dibuka. Setelah itu, dokumen akan berhasil ditandatangani.



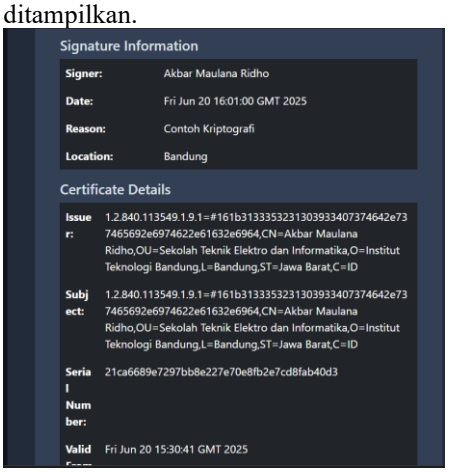
E. Verifikasi Dokumen dengan StirlingPDF

Proses verifikasi dapat dilakukan dengan kakas pembaca PDF mana saja seperti Adobe Acrobat dan Foxit PDF Reader dengan catatan penerima sudah memasukkan sertifikat pengirim ke dalam *trust store* penerima. Contoh ini akan mendemonstrasikan proses verifikasi dengan kakas StirlingPDF.

1. Buka halaman StirlingPDF, lalu cari kakas **Sign with Certificate**. Sebagai alternatif, buka pranala berikut.
2. Masukkan dokumen yang sudah ditandatangani. Selain itu, penerima harus memasukkan sertifikat yang digunakan oleh pengirim.



3. Tekan tombol validasi untuk memverifikasi dokumen. Apabila berhasil, informasi hasil verifikasi akan ditampilkan.



F. Verifikasi Dokumen dengan pyHanko

Jalankan perintah berikut untuk melakukan verifikasi dokumen.

```
pyhanko sign validate --cert kriptografi.crt --pretty-print
draft_kripto_signed.pdf
```

Perintah ini menerima alamat sertifikat yang dipercaya (dalam hal ini sertifikat pengirim) serta alamat dokumen yang ingin diperiksa validitasnya.

IV. KESIMPULAN

Berdasarkan eksplorasi di atas, penggunaan *self-signed certificate* bersamaan dengan *digitized signature* merupakan hal yang memungkinkan untuk dilakukan. Terlebih lagi, dengan banyaknya kakas sumber terbuka proses ini dapat dilakukan

dengan lebih mudah. Meskipun begitu, kelemahan utama dari penggunaan *self-signed certificate* terletak pada sertifikat yang digunakan tidak secara otomatis dipercaya oleh banyak perangkat. Agar sertifikat dapat dipercaya, diperlukan biaya dan proses verifikasi yang panjang dari Certificate Authority. Di sisi lain, layanan tanda tangan digital menawarkan penandatanganan dokumen dengan sertifikat atau turunan sertifikat yang sudah dipercaya. Penggunaan sertifikat tersebut dapat mempermudah proses verifikasi dari sisi penerima. Pada akhirnya, penggunaan *self-signed certificate* dikembalikan kepada kebutuhan individu dan organisasi. Meskipun begitu, makalah ini mengeksplorasi bahwa hal penandatanganan dokumen dengan *self-signed certificate* merupakan hal yang mungkin dilakukan.

We suggest that you use a text box to insert a graphic (which is ideally a 300 dpi resolution TIFF or EPS file with all fonts embedded) because this method is somewhat more stable than directly inserting a picture.

To have non-visible rules on your frame, use the MSWord "Format" pull-down menu, select Text Box > Colors and Lines to choose No Fill and No Line.

VIDEO LINK AT YOUTUBE *(Heading 5)*

https://youtu.be/LOOnIEF-qTY

REFERENCES

[1] Munir. Rinaldi, Tanda Tangan Digital. Bandung, West Java, 2025.

[2] Munir. Rinaldi, Elgamal Signature. Bandung, West Java, 2025.

[3] Munir. Rinaldi, Digital Signature Algorithm (DSA) dan Elliptic Curve Digital Signature Algorithm (ECDSA). Bandung, West Java, 2025.

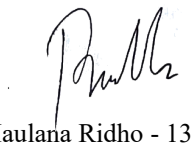
[4] Stirling-Tools, *Stirling-PDF*. GitHub repository. [Online]. Available: https://github.com/Stirling-Tools/Stirling-PDF.

[5] M. Valvekens, *pyHanko*. GitHub repository. [Online]. Available: https://github.com/MatthiasValvekens/pyHanko.

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, 20 Juni 2025



Akbar Maulana Ridho - 13521093