Jurnal Multidisiplin Teknik, Sains, Pendidikan dan Teknologi

E-ISSN: 3064-6103
Volume 2, **Number** 1, Maret 2025
https://doi.org/10.62671/suliwa.v2i1.56

PERANCANGAN PROGRAM SEDERHANA DIGITAL SIGNATURE MENGGUNAKAN DSA DALAM BAHASA PYTHON

Muhammad Irby Syi'bu Al Huda[1], Reza Dwi Putri[2*], Teuku Muhammad Arinal[3] , Nur Padila Pohan[4], Ahmad Umraithi[5], Sayid Muhammad Jundullah[6], Ryan Rizky Ananda[7], M. Fatiha Assyfa[8]
[1,2,3,4,5,6,7,8]Universitas Malikussaleh, Indonesia
[1]muhammad.210170233@mhs.unimal.ac.id, [2]reza.210170061@mhs.unimal.ac.id, [3]teuku.210170085@mhs.unimal.ac.id, [4]nur.220170062@mhs.unimal.ac.id, [5]ahmad.220170109@mhs.unimal.ac.id, [6]sayid.220170045@mhs.unimal.ac.id, [7]ryan.220170061@mhs.unimal.ac.id, [8]muhammad.210170129@mhs.unimal.ac.id

ABSTRACT

Keamanan data digital menjadi hal yang sangat penting di era teknologi informasi saat ini. Salah satu metode yang digunakan untuk menjaga integritas dan keaslian data adalah tanda tangan digital. Penelitian ini membahas implementasi algoritma Digital Signature Algorithm (DSA) dalam bentuk program sederhana menggunakan bahasa pemrograman Python. Tujuan utama dari penelitian ini adalah memberikan pemahaman dasar mengenai cara kerja DSA melalui proses pembuatan kunci, penandatanganan pesan, dan verifikasi tanda tangan. Dengan menggunakan pustaka cryptography pada Python, program ini mampu menghasilkan tanda tangan digital dan memverifikasinya secara efisien. Hasil pengujian menunjukkan bahwa tanda tangan digital yang dihasilkan hanya valid jika pesan dan tanda tangan tidak mengalami perubahan. Implementasi ini diharapkan dapat menjadi langkah awal bagi pengembangan sistem keamanan digital yang lebih lanjut dan sebagai media pembelajaran kriptografi secara praktis.

Kata Kunci: *Digital Signature Algorithm (DSA), Keamanan Data, Kriptografi, Tanda Tangan Digital*

PENDAHULUAN

Dalam era digital saat ini, keamanan data menjadi aspek penting dalam berbagai aktivitas yang dilakukan secara daring. Salah satu bentuk perlindungan data yang digunakan adalah tanda tangan digital atau digital signature. Tanda tangan digital merupakan teknologi yang memungkinkan pembuktian secara matematis bahwa suatu data tetap utuh dan tidak mengalami perubahan yang tidak sah, sehingga dapat dijadikan solusi untuk memverifikasi integritas dokumen yang valid (Finandhita dan Afrianto, 2018) (Nugraha, 2017). Teknologi ini memungkinkan verifikasi keaslian suatu dokumen atau pesan digital tanpa perlu mengandalkan metode konvensional. Digital Signature Algorithm (DSA) merupakan salah satu algoritma yang digunakan untuk menghasilkan tanda tangan digital yang aman dan efisien.

DSA merupakan algoritma kriptografi yang dikembangkan oleh National Institute of Standards and Technology (NIST) sebagai bagian dari standar Digital Signature Standard (DSS) (Pratama et al.). DSA merupakan salah satu algoritma kriptografi seperti AES, Blowfish, DES, IDEA, RC4, dan lain-lain (Jasman, Arisandi and Sukri, 2017). Fungsi utama dari DSA adalah untuk menjamin bahwa pesan yang dikirim berasal dari pihak yang sah dan tidak mengalami perubahan(Nurhasanah, 2013).. Dalam penerapannya, DSA menggunakan prinsip kriptografi kunci publik, di mana satu kunci digunakan untuk menandatangani pesan (kunci privat) dan satu lagi digunakan untuk memverifikasi tanda tangan (kunci publik) (Yassein et al., 2017).

Penerapan algoritma DSA tidak hanya terbatas pada sistem keamanan berskala besar, tetapi juga dapat dipelajari dan diimplementasikan dalam bentuk program sederhana. Hal ini penting bagi para pemula atau mahasiswa yang ingin memahami konsep dasar tanda tangan digital melalui pendekatan praktis. Salah satu bahasa pemrograman yang cocok digunakan untuk eksperimen ini adalah Python, karena memiliki banyak pustaka (library) kriptografi yang mendukung DSA.

Dengan memanfaatkan pustaka seperti cryptography dalam Python, pembuatan kunci DSA, penandatanganan pesan, dan proses verifikasi dapat dilakukan dengan lebih mudah dan efisien. Program sederhana ini dapat menjadi langkah awal dalam memahami bagaimana algoritma DSA bekerja dan bagaimana algoritma tersebut dapat diimplementasikan dalam dunia nyata. Selain itu, pendekatan ini juga membuka peluang untuk mengembangkan sistem keamanan yang lebih kompleks di masa depan.

Tujuan dari pembuatan jurnal ini adalah untuk menjelaskan konsep dasar DSA serta mendemonstrasikan implementasinya dalam bentuk program Python sederhana. Fokus utama ditujukan pada bagaimana cara membuat kunci, menandatangani pesan, dan memverifikasi tanda tangan dengan menggunakan pustaka bawaan Python maupun pustaka tambahan. Hal ini bertujuan untuk memberikan pemahaman dasar yang praktis bagi pembaca.

Jurnal Multidisiplin Teknik, Sains, Pendidikan dan Teknologi

E-ISSN: 3064-6103
Volume 2, **Number** 1, Maret 2025
https://doi.org/10.62671/suliwa.v2i1.56

Melalui jurnal ini, diharapkan pembaca dapat memahami bagaimana tanda tangan digital bekerja secara teknis dan bagaimana DSA sebagai salah satu algoritma yang digunakan dalam pengamanan pesan dapat diimplementasikan secara langsung melalui bahasa pemrograman Python. Pendekatan praktis ini diharapkan menjadi jembatan antara teori kriptografi dan aplikasi nyata dalam pengembangan sistem keamanan digital.

KAJIAN LITERATUR/TINJAUAN PUSTAKA

Penelitian ini menggunakan pendekatan eksperimen praktis dengan fokus pada implementasi algoritma Digital Signature Algorithm (DSA) menggunakan bahasa pemrograman Python. Tujuannya adalah untuk memahami konsep dasar tanda tangan digital serta membuktikan proses penandatanganan dan verifikasi secara langsung melalui program sederhana. Algoritma DSA yang digunakan dalam proses penandatanganan pesan juga melibatkan fungsi SHA untuk menghasilkan message digest dari pesan tersebut (Pajcin and Ivanis, 2011).

METODE PENELITIAN

Studi Literatur

Peneliti melakukan studi literatur untuk memahami konsep dasar dari kriptografi kunci publik, tanda tangan digital, serta algoritma DSA. Referensi yang digunakan berasal dari dokumentasi resmi Python, modul cryptography, dan sumber ilmiah terkait keamanan data digital. Tujuannya adalah memperoleh pemahaman mengenai cara kerja algoritma DSA, termasuk proses pembangkitan kunci, penandatanganan, dan verifikasi.

Perancangan Program

Program dirancang untuk dapat melakukan dua fungsi utama, yaitu: (1) menandatangani sebuah teks menggunakan kunci privat, dan (2) memverifikasi tanda tangan digital menggunakan kunci publik. Program juga dirancang dengan antarmuka terminal interaktif berbasis menu untuk memudahkan pengguna dalam memilih opsi operasi. Kunci DSA dibuat secara dinamis pada saat program dijalankan dengan ukuran kunci 2048 bit.

Implementasi

Implementasi dilakukan menggunakan bahasa Python dan pustaka cryptography.hazmat.primitives. Modul dsa digunakan untuk pembangkitan kunci asimetris dan hashes untuk algoritma hash (SHA-256). Penandatanganan dilakukan dengan memanggil metode .sign() pada kunci privat, sedangkan verifikasi menggunakan .verify() pada kunci publik. Tanda tangan dikonversi ke format heksadesimal untuk memudahkan tampilan dan input pengguna.

```
# 1. Membuat Kunci DSA (Asimetris)
private_key = dsa.generate_private_key(key_size=2048)  # Kunci Privat
public_key = private_key.public_key()  # Kunci Publik
```

Gambar 1. Pembuatan Kunci DSA

Gambar tersebut menunjukkan kode program untuk menghasilkan kunci DSA dengan ukuran 2048 bit menggunakan pustaka *cryptography*. Kunci privat digunakan untuk menandatangani teks, sedangkan kunci publik digunakan untuk memverifikasi keaslian tanda tangan digital.

Setelah kunci berhasil dibuat, pengguna dapat memasukkan teks yang ingin ditandatangani. Fungsi sign_message() akan menghasilkan nilai *hash* dari teks, kemudian menandatanganinya dengan kunci privat. Hasilnya dikembalikan dalam bentuk heksadesimal. Ilustrasi proses ini ditunjukkan pada gambar berikut ini.

```
8   def sign_message(teks):
9       """Menandatangani teks dengan kunci privat."""
10      signature = private_key.sign(
11          teks.encode(),
12          hashes.SHA256()
13      )
14      return signature.hex()  # Langsung dikonversi ke heksadesimal
```

Gambar 2. Fungsi Penandatanganan Teks

Pengujian

Pengujian dilakukan dengan menjalankan program secara langsung dan memasukkan teks untuk ditandatangani. Hasil tanda tangan kemudian digunakan kembali untuk proses verifikasi. Untuk menguji integritas dan keaslian, dilakukan juga pengujian negatif dengan mengubah isi teks atau tanda tangan agar proses verifikasi gagal, guna membuktikan bahwa sistem dapat mendeteksi ketidaksesuaian.

Jurnal Multidisiplin Teknik, Sains, Pendidikan dan Teknologi

E-ISSN: 3064-6103
Volume 2, **Number** 1, Maret 2025
https://doi.org/10.62671/suliwa.v2i1.56

Fungsi verify_signature() digunakan untuk memverifikasi tanda tangan digital dengan menggunakan kunci publik. Jika tanda tangan valid, maka teks tersebut dinyatakan asli dan tidak mengalami perubahan. Jika tidak valid, maka ada kemungkinan tanda tangan tidak berasal dari pengirim yang sah atau terjadi modifikasi data. Gambar 3 menunjukkan implementasi proses verifikasi.

```
def verify_signature(teks, signature_hex):
    """Memverifikasi tanda tangan dengan kunci publik."""
    try:
        signature = bytes.fromhex(signature_hex)  # Konversi dari heksadesimal ke bytes
        public_key.verify(
            signature,
            teks.encode(),
            hashes.SHA256()
        )
        return True  # Jika valid
    except:
        return False  # Jika tidak valid
```

Gambar 3. Fungsi Verifikasi Tanda Tangan

Analisis

Hasil dari proses penandatanganan dan verifikasi diamati secara langsung melalui *output* terminal. Keberhasilan verifikasi menunjukkan bahwa tanda tangan valid dan teks tidak mengalami perubahan. Sebaliknya, jika verifikasi gagal, maka sistem berhasil mengenali adanya perubahan atau ketidaksesuaian. Hal ini membuktikan bahwa konsep dasar DSA telah berhasil diimplementasikan dengan baik dalam bentuk program sederhana.

Program yang dibuat bersifat interaktif, di mana pengguna diberikan pilihan untuk menandatangani, memverifikasi, atau keluar dari program. Tampilan interaksi ini ditunjukkan pada Gambar 4 berikut:

```
while True:
    print("\nPilih operasi:")
    print("1. Tanda tangani teks")
    print("2. Verifikasi tanda tangan")
    print("3. Keluar")

    pilihan = input("Masukkan pilihan (1, 2, atau 3): ")

    if pilihan == "1":
        teks = input("\nMasukkan teks yang ingin ditandatangani: ")
        tanda_tangan = sign_message(teks)
        print("\n✅ Teks telah ditandatangani!")
        print("Tanda tangan digital:", tanda_tangan)

    elif pilihan == "2":
        teks = input("\nMasukkan teks yang ingin diverifikasi: ")
        tanda_tangan_hex = input("Masukkan tanda tangan digital (heksadesimal): ")

        if verify_signature(teks, tanda_tangan_hex):
            print("\n✅ Verifikasi sukses! Tanda tangan valid.")
        else:
            print("\n❌ Verifikasi gagal! Tanda tangan tidak valid.")

    elif pilihan == "3":
        print("\nTerima kasih! Program selesai.")
        break  # Keluar dari loop

    else:
        print("\n❌ Pilihan tidak valid, coba lagi.")
```

Gambar 4. Interaksi Menu Program

HASIL DAN PEMBAHASAN

Program yang telah dibuat bertujuan untuk menerapkan algoritma Digital Signature Algorithm (DSA) menggunakan bahasa pemrograman Python. Program ini terdiri dari dua fungsi utama, yaitu: Penandatanganan teks (Sign) menggunakan kunci privat, Verifikasi tanda tangan digital (Verify) menggunakan kunci public.

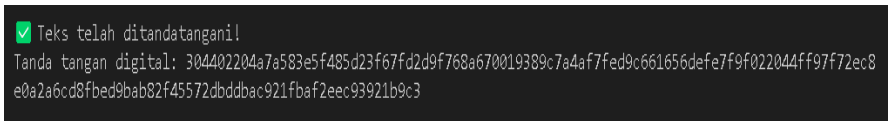
Proses Penandatanganan

Ketika pengguna memilih opsi 1 dan memasukkan teks (misalnya "bagus"), maka teks tersebut akan di encode menjadi byte dan ditandatangani menggunakan metode .sign() dari kunci privat DSA. Hasil dari penandatanganan ini

Jurnal Multidisiplin Teknik, Sains, Pendidikan dan Teknologi

E-ISSN: 3064-6103
Volume 2, **Number** 1, Maret 2025
https://doi.org/10.62671/suliwa.v2i1.56

adalah data dalam bentuk bytes yang kemudian dikonversi ke bentuk heksadesimal agar lebih mudah ditampilkan dan disimpan. Contoh hasil tanda tangan digital:



Gambar 5. Contoh Hasil Tanda Tangan Tanda tangan ini unik untuk teks "bagus" dan kunci privat tertentu.

Proses Verifikasi

Pengguna memilih opsi 2, program akan meminta dua input:

- Teks yang ingin diverifikasi
- Tanda tangan digital dalam bentuk heksadesimal

Program kemudian melakukan konversi tanda tangan ke bytes dan memverifikasi kecocokan antara teks dan tanda tangan tersebut menggunakan kunci publik DSA. Jika cocok, akan muncul pesan:

Gambar 6. Contoh Hasil Tanda Tangan

Jika isi teks atau tanda tangan diubah, maka tanda tangan dianggap tidak valid, dan muncul pesan:

Gambar 7. Contoh Hasil Tanda Tangan

KESIMPULAN

Berdasarkan hasil implementasi yang telah dilakukan, dapat disimpulkan bahwa algoritma Digital Signature Algorithm (DSA) berhasil diterapkan untuk membuat dan memverifikasi tanda tangan digital menggunakan bahasa pemrograman Python. Program yang dibuat mampu menjalankan fungsi utama DSA, yaitu menjamin integritas dan keaslian pesan melalui proses penandatanganan dengan kunci privat dan verifikasi menggunakan kunci publik. Dari pengujian yang dilakukan, sistem mampu mendeteksi apakah suatu pesan telah dimodifikasi atau tidak dengan memastikan bahwa tanda tangan digital tetap valid hanya jika teks dan tanda tangan tidak berubah. Hal ini menunjukkan bahwa DSA merupakan metode yang efektif dalam menjaga keamanan data, terutama dalam autentikasi dan verifikasi pesan digital. Implementasi sederhana ini juga membuktikan bahwa konsep kriptografi dapat dipahami dan diterapkan secara praktis oleh pemula, serta dapat menjadi dasar dalam pengembangan sistem keamanan data yang lebih kompleks di masa mendatang.

REFERENSI

Jasman, J., Arisandi, D. and Sukri, S. (2017) 'Rancang Bangun Aplikasi Enkripsi Coding Berbasis Php Program Menggunakan Algoritma Aes', in 2th Celscitech-UMRI 2017 Vol. Pekanbaru: LP2M-UMRI, pp. 49–61

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

Nurhasanah, F. (2013) 'Pembuatan Tanda Tangan Digital Menggunakan Digital Signature Algorithm', MATHunesa, 2(2).

Pajcin, B. R. and Ivanis, P. N. (2011) 'Analysis of Software Realized DSA Algorithm for Digital Signature', Guest Editorial W. Citeseer, p. 73

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), doi:10.34010/komputa.v6i2.2481.

Pratama Jr, M. D. Y. A., Arnaldy, D., TP, S., & Si, M. Analisis Integritas Dokumen Digital Pada Aplikasi Digisign UTD PNJ Menggunakan Tanda Tangan Digital.

Yassein, M. B. et al. (2017) 'Comprehensive study of symmetric key and asymmetric key encryption algorithms', in 2017 international conference on engineering and technology (ICET).IEEE, pp. 1–7.