

Implementasi Steganografi Unicode pada Emoji dalam Media Komunikasi Modern

Rafiki Prawhira Harianto - 13522065

Program Studi Teknik Informatika

Sekolah Teknik Elektro dan Informatika

Institut Teknologi Bandung, Jl. Ganesha 10 Bandung 40132, Indonesia

E-mail: 13522065@std.stei.itb.ac.id, rafikiprawhira@gmail.com

Abstraksi—Perkembangan komunikasi digital modern telah menjadikan teks dan emoji sebagai media utama pertukaran informasi. Penelitian ini mengkaji implementasi steganografi berbasis Unicode dengan memanfaatkan *variation selector* pada emoji sebagai kanal penyisipan data tersembunyi. Teknik ini memungkinkan penyembunyian informasi tanpa mengubah tampilan visual pesan, sehingga sulit dideteksi oleh pihak ketiga. Implementasi dilakukan menggunakan bahasa Python dengan pemetaan setiap byte data ke karakter *variation selector* yang valid. Pengujian dilakukan pada empat media komunikasi populer—LINE, WhatsApp, Telegram, dan Discord—untuk mengevaluasi keberhasilan penyisipan dan ekstraksi data serta transparansi visual. Hasil menunjukkan bahwa metode ini efektif untuk payload kecil hingga menengah, namun memiliki keterbatasan pada ukuran besar akibat kebijakan pemrosesan teks masing-masing media. Penelitian ini menyoroti potensi steganografi Unicode dalam komunikasi modern serta memberikan rekomendasi untuk pengembangan lebih lanjut, termasuk integrasi kriptografi dan mekanisme deteksi.

Keywords— Unicode, Emoji, Steganografi Teks, Variation Selector, Keamanan Komunikasi

I. PENDAHULUAN

Perkembangan sistem komunikasi digital modern telah menjadikan teks sebagai media utama pertukaran informasi, baik melalui aplikasi pesan instan, media sosial, maupun media kolaborasi daring. Di dalam komunikasi berbasis teks tersebut, penggunaan emoji semakin meluas sebagai elemen ekspresif yang terstandarisasi dalam Unicode.

Steganografi bertujuan untuk menyembunyikan keberadaan pesan itu sendiri di dalam suatu media pembawa, sehingga pihak ketiga tidak menyadari bahwa suatu komunikasi tersembunyi sedang berlangsung. Steganografi secara tradisional banyak diterapkan pada media gambar, audio, atau video, tetapi perkembangan standar teks digital membuka peluang baru bagi steganografi berbasis teks.

Salah satu fitur dalam standar Unicode yang jarang diketahui adalah *variation selector*, yaitu karakter khusus yang digunakan untuk memodifikasi tampilan suatu simbol tanpa mengubah makna dasarnya. *Variation selector* pada umumnya tidak memiliki representasi visual yang dapat

dibedakan oleh pengguna, tetapi tetap tersimpan secara eksplisit dalam representasi teks. Karakteristik ini memungkinkan *variation selector* dimanfaatkan sebagai kanal tersembunyi untuk menyisipkan data tambahan tanpa mengubah tampilan teks secara kasat mata. Dengan mengombinasikan *variation selector* dan emoji sebagai pembawa, dimungkinkan untuk menyelipkan data ke dalam sebuah pesan teks yang secara kasat mata tampak normal.

Penelitian dan eksplorasi terkini menunjukkan bahwa teknik steganografi berbasis Unicode memiliki implikasi signifikan terhadap keamanan media komunikasi modern. Informasi tersembunyi dapat melewati media moderasi konten maupun proses pemrosesan teks otomatis yang tidak melakukan normalisasi Unicode secara menyeluruh. Oleh karena itu, diperlukan kajian lebih lanjut mengenai bagaimana teknik steganografi Unicode pada emoji dapat diimplementasikan, diuji, dan dievaluasi dalam lingkungan komunikasi modern.

Makalah ini bertujuan untuk mengkaji implementasi steganografi Unicode pada emoji dengan memanfaatkan *variation selector* sebagai media penyisipan data. Fokus pembahasan meliputi landasan teoritis Unicode dan *variation selector*, mekanisme implementasi penyisipan dan ekstraksi data, serta evaluasi efektivitasnya dalam media komunikasi modern.

II. LANDASAN TEORI

A. Unicode Emoji

Unicode merupakan standar internasional yang dirancang untuk merepresentasikan karakter teks dari berbagai bahasa, simbol, dan sistem penulisan dalam satu skema pengkodean universal [1]. Setiap karakter dalam Unicode direpresentasikan oleh sebuah poin kode yang bersifat unik, sehingga memungkinkan interoperabilitas teks lintas media dan sistem operasi.

Dalam praktiknya, Unicode diimplementasikan melalui beberapa skema pengkodean, seperti UTF-8, UTF-16, dan UTF-32 [2]. UTF-8 merupakan skema yang paling umum digunakan dalam media komunikasi modern karena kompatibel dengan ASCII dan efisien dalam penyimpanan [3]. Unicode juga mencakup karakter yang tidak ditampilkan secara langsung kepada pengguna untuk

berbagai keperluan, seperti *variation selector*.

Emoji merupakan bagian dari standar Unicode yang biasanya direpresentasikan oleh satu atau lebih poin kode. Emoji dapat dikombinasikan dengan karakter tambahan untuk menghasilkan variasi tampilan tertentu, seperti variasi warna kulit atau gaya tampilan [4]. Contohnya, emoji *woman* diikuti emoji *modifier medium-dark-skin-tone* menghasilkan emoji wanita dengan warna kulit coklat (Gambar 4.). Kompleksitas ini memungkinkan Unicode dimanfaatkan sebagai media penyembunyian informasi dalam konteks steganografi teks.



Gambar 1. Kombinasi Emoji *woman* dengan emoji *modifier medium-dark skin tone* [4].

B. Variation Selector

Unicode *Variation Selector* adalah karakter khusus dalam standar Unicode yang berfungsi untuk menentukan variasi representasi grafis dari karakter sebelumnya [5]. *Variation selector* tidak memiliki tampilan visual sendiri dan hanya memengaruhi cara karakter pendahulunya dirender oleh sistem.

Terdapat dua kelompok utama *variation selector*, yaitu *Variation Selectors* (VS1–VS16) dan *Variation Selectors Supplement* (VS17–VS256) [5], [6]. Ketika *variation selector* disisipkan setelah sebuah karakter yang tidak memilikinya, khususnya emoji, sistem tetap menampilkan karakter yang serupa secara visual, sementara informasi variasi tetap tersimpan dalam representasi teksnya [4].

Karakteristik ini menjadikan *variation selector* sangat sesuai digunakan sebagai media steganografi, karena data dapat disembunyikan tanpa menimbulkan perubahan visual yang mencolok. Selain itu, banyak sistem pemrosesan teks tidak melakukan validasi atau normalisasi terhadap *variation selector*, sehingga karakter tersebut dapat bertahan selama proses penyimpanan, transmisi, maupun penyalinan teks.

C. Steganografi Teks Unicode

Steganografi merupakan ilmu dan seni menyembunyikan pesan rahasia dengan suatu cara sedemikian sehingga tidak seorang pun yang mencurigai keberadaan pesan tersebut [7]. Dengan teks Unicode, Steganografi dilakukan dengan menyembunyikan informasi dengan memanfaatkan karakter atau fitur dalam standar Unicode yang tidak memengaruhi tampilan visual teks secara langsung. Pendekatan ini berbeda dengan steganografi media lain, seperti citra atau audio [7], karena tidak bergantung pada redundansi sinyal, melainkan pada struktur representasi karakter dalam teks digital.

Salah satu implementasi steganografi teks Unicode yang relevan adalah teknik penyisipan data menggunakan Unicode *Variation Selector* yang dikaitkan dengan karakter emoji. Teknik ini dipopulerkan dan

didemonstrasikan secara praktis oleh Butler (2025), yang menunjukkan bahwa *variation selector* dapat digunakan untuk menyelundupkan data arbitrer ke dalam satu karakter emoji tanpa mengubah tampilan visualnya [8]. Dalam pendekatan tersebut, setiap *variation selector* dipetakan ke nilai byte tertentu, sehingga rangkaian *variation selector* yang ditempatkan setelah emoji dapat merepresentasikan data biner tersembunyi (Gambar 2.). Secara teoritis, keberadaan hingga 256 *variation selector* memungkinkan pemetaan langsung terhadap nilai satu byte data [8].

This prints:

```
"👩🏭\u{e0158}\u{e0155}\u{e015c}\u{e015c}\u{e015f}"
```

This reveals the characters that were “hidden” in the original output.

Gambar 2. Biner tersembunyi pada hasil *encoded* emoji [8].

Keunggulan utama dari teknik ini adalah transparansi visual yang tinggi, karena emoji tetap dirender identik di sisi penerima. Selain itu, karena *variation selector* merupakan bagian sah dari standar Unicode, karakter tersebut umumnya tidak difilter atau dihapus oleh sistem komunikasi modern, seperti aplikasi pesan instan, media sosial, maupun pipeline pemrosesan teks otomatis. Hal ini memungkinkan data tersembunyi bertahan selama proses transmisi, penyimpanan, dan penyalinan teks. Namun, dari sudut pandang implementasi, kapasitas data yang dapat disisipkan terbatas oleh jumlah *variation selector* yang dapat disisipkan secara praktis tanpa menimbulkan anomali pada representasi teks.

III. IMPLEMENTASI

Bab ini membahas implementasi steganografi teks Unicode berbasis emoji dengan memanfaatkan Unicode *Variation Selector* sebagai media penyisipan data. Implementasi dilakukan menggunakan bahasa pemrograman Python dan mengacu pada pendekatan yang diperkenalkan oleh Butler (2025) [8], yaitu dengan memetakan setiap byte data ke satu karakter *variation selector* yang disisipkan setelah sebuah karakter dasar unicode yang umumnya emoji.

Unicode menyediakan dua rentang *variation selector* yang valid, yaitu rentang U+FE00 hingga U+FE0F dan rentang U+E0100 hingga U+E01EF. Kedua rentang tersebut menghasilkan total 256 *variation selector*, sehingga memungkinkan pemetaan langsung terhadap seluruh nilai byte dari 0 hingga 255. Pendekatan ini memungkinkan penyisipan data secara deterministik dan reversibel.

Proses pemetaan byte ke *variation selector* diimplementasikan melalui fungsi *byte_to_variation_selector*. Fungsi ini menerima sebuah nilai byte dan mengonversinya ke karakter Unicode *variation selector* yang sesuai berdasarkan rentang nilai byte tersebut.

```
def byte_to_variation_selector(byte):
    if not 0 <= byte <= 255:
        raise ValueError(f"Byte must be between 0
and 255, got {byte}")

    if byte < 16:
        return chr(0xFE00 + byte)
    else:
        return chr(0xE0100 + (byte - 16))
```

Sebaliknya, fungsi *variation_selector_to_byte* digunakan untuk melakukan proses invers, yaitu mengonversi *variation selector* kembali menjadi nilai byte semula.

```
def variation_selector_to_byte(char):
    codepoint = ord(char)

    if 0xFE00 <= codepoint <= 0xFE0F:
        return codepoint - 0xFE00
    elif 0xE0100 <= codepoint <= 0xE01EF:
        return codepoint - 0xE0100 + 16
    else:
        return None
```

Proses *encoding* dilakukan dengan menyisipkan rangkaian *variation selector* setelah satu karakter dasar yang umumnya berupa emoji. Data yang akan disisipkan dapat berupa string atau byte array. Apabila data berupa string, data terlebih dahulu dikonversi ke representasi UTF-8 agar dapat diproses dalam bentuk byte.

Fungsi *encode* mengimplementasikan proses tersebut dengan cara mengiterasi setiap byte data dan menambahkan *variation selector* hasil pemetaan ke dalam string hasil encoding. Hasil akhirnya berupa sebuah string Unicode yang secara kasat mata hanya menampilkan emoji, tetapi mengandung data tersembunyi.

```
def encode(base_char, data):
    if isinstance(data, str):
        data = data.encode('utf-8')

    result = base_char
    for byte in data:
        result +=
byte_to_variation_selector(byte)

    return result
```

Sebaliknya, proses *decoding* bertujuan untuk mengekstraksi kembali data tersembunyi dari teks steganografi. Fungsi *decode* melakukan iterasi terhadap setiap karakter dalam teks dan memeriksa apakah karakter tersebut merupakan *variation selector* yang valid. Ketika *variation selector* ditemukan, nilainya dikonversi kembali menjadi byte dan disimpan. Proses *decoding* dihentikan ketika karakter non-*variation selector* pertama ditemukan setelah rangkaian data dimulai.

```
def decode(text):
    result = []
    started = False

    for char in text:
        byte = variation_selector_to_byte(char)
        if byte is not None:
            result.append(byte)
            started = True
        elif started:
            # stop at first non-variation-
selector
            break

    return bytes(result)
```

Untuk mempermudah penggunaan, disediakan fungsi pembungkus yang menerima dan menghasilkan teks. Fungsi *encode_text* digunakan untuk menyisipkan teks ke dalam emoji, sedangkan fungsi *decode_text* digunakan untuk mengekstraksi kembali teks tersembunyi. Pada proses decoding, hasil ekstraksi dicoba untuk dikonversi kembali ke string UTF-8. Apabila konversi gagal, data dikembalikan dalam bentuk representasi heksadesimal.

```
def encode_text(base_char, text):
    return encode(base_char, text)

def decode_text(stego_text):
    data = decode(stego_text)
    try:
        return data.decode('utf-8')
    except UnicodeDecodeError:
        return data.hex()
```

Setelah fungsi pembungkus untuk proses encoding dan decoding, implementasi ini dilengkapi dengan fungsi utama yang berperan sebagai antarmuka *command-line interface* (CLI) sederhana untuk *encode* dan *decode*, dilengkapi mode *sample test* untuk menjalankan pengujian Bab IV secara otomatis dan menyimpan hasil *encoding* sebagai berkas text (Gambar 3.).

```

PS D:\Kuliah-ITB\Semester 7\Kriptografi\
"d:/Kuliah-ITB/Semester 7/Kriptografi/en
=== Emoji Steganography ===

1. Encode a message
2. Decode a message
3. Exit
4. Sample Test

Choose an option (1-4): 1
Enter base emoji/character: 🤪
Enter message to encode: Halo, Dunia!
Encoded result: 🤪
Length: 13 characters

1. Encode a message
2. Decode a message
3. Exit
4. Sample Test

Choose an option (1-4): 2
Enter encoded text to decode: 🤪
Decoded message: Halo, Dunia!

```

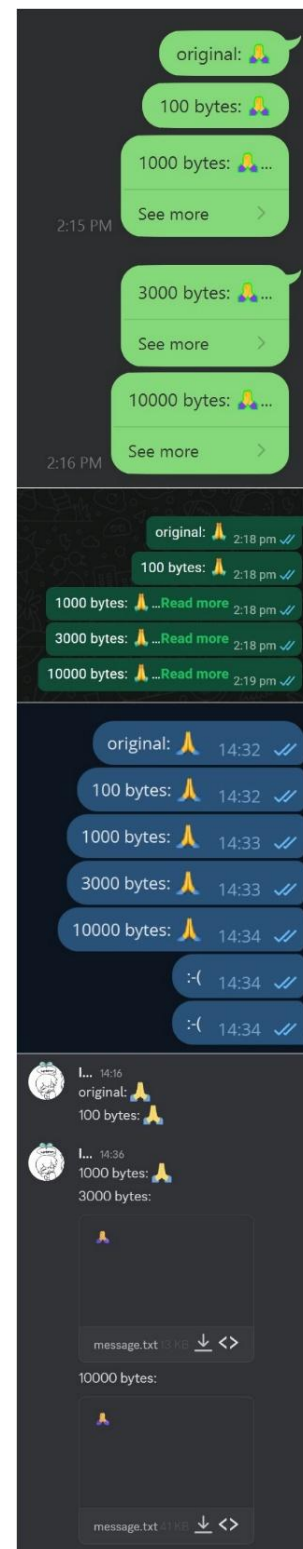
Gambar 3. Tampilan program dan contoh penggunaan *encode* dan *decode*.

IV. PENGUJIAN DAN EVALUASI

Bab ini membahas pengujian dan evaluasi terhadap implementasi steganografi Unicode berbasis emoji yang telah dibangun pada Bab III. Pengujian dilakukan untuk mengevaluasi apakah data tersembunyi yang disisipkan menggunakan Unicode Variation Selector dapat bertahan selama proses pengiriman pesan pada media komunikasi modern, serta untuk menilai transparansi visual emoji dan keberhasilan proses ekstraksi data setelah transmisi.

Pengujian dirancang untuk merepresentasikan penggunaan nyata pada aplikasi pesan instan yang umum digunakan. Empat media komunikasi modern dipilih, yaitu LINE, WhatsApp, Telegram, dan Discord, karena masing-masing memiliki karakteristik pemrosesan teks yang berbeda. Seluruh pengujian dilakukan menggunakan perangkat desktop, dengan aplikasi resmi dari masing-masing media, serta lingkungan Python yang digunakan untuk melakukan proses encoding dan decoding pesan steganografi.

Data uji yang digunakan berupa teks *lorem ipsum* yang disisipkan ke dalam satu karakter emoji menggunakan mekanisme steganografi Unicode. Untuk mengamati batas kemampuan masing-masing media, pengujian dilakukan pada empat ukuran payload yang berbeda, yaitu 100 byte, 1000 byte, 3000 byte, dan 10000 byte. Setiap payload di-*encode* menggunakan implementasi pada Bab III, kemudian dikirimkan melalui pesan teks standar dari akun pribadi ke akun pribadi tanpa modifikasi tambahan (Gambar 4.).



Gambar 4. Pengujian pengiriman beragam ukuran payload pada media Line, WhatsApp, Telegram, dan Discord.

Berdasarkan hasil pengujian, LINE menunjukkan perubahan perilaku antarmuka mulai terlihat pada payload berukuran 1000 byte, di mana pesan ditampilkan dalam mode *see more*. Pada ukuran payload hingga 3000 byte, pesan masih dapat ditampilkan secara visual tanpa anomali yang signifikan dan proses *encoding* serta *decoding*

berhasil dilakukan. Namun, pada payload berukuran 10000 byte, terjadi kegagalan visualisasi yang ditandai dengan munculnya simbol pengganti karakter, serta data tersembunyi tidak lagi dapat diekstraksi secara utuh.

Pada WhatsApp, perubahan antarmuka mulai muncul pada payload 1000 byte dalam bentuk *read more*. Proses encoding dan decoding masih berhasil dilakukan hingga payload 3000 byte. Untuk payload 10000 byte, emoji masih ditampilkan secara visual dengan baik tanpa indikasi kerusakan tampilan, namun data tersembunyi tidak dapat didekode kembali. Hal ini menunjukkan bahwa meskipun tampilan visual tetap terjaga, sebagian *variation selector* kemungkinan tidak dipertahankan selama proses transmisi atau penanganan pesan.

Hasil yang berbeda ditunjukkan oleh Telegram. Media ini mampu mempertahankan proses *encoding* dan *decoding* hingga payload berukuran 3000 byte. Namun, pada payload 10000 byte, emoji tidak lagi ditampilkan sebagaimana mestinya dengan sebagian *variant selector* digantikan dengan representasi teks berupa simbol “:- (“). Pada kondisi tersebut, data tersembunyi tidak dapat diekstraksi kembali, yang menunjukkan bahwa perubahan representasi visual berkorelasi dengan hilangnya *variation selector*.

Discord menunjukkan perilaku yang paling mengizinkan terhadap payload berukuran besar. Pada payload 3000 byte, pesan mulai ditampilkan sebagai text file oleh antarmuka aplikasi. Meskipun demikian, hingga payload 10000 byte, proses encoding dan decoding masih dapat dilakukan dengan baik, dan data tersembunyi dapat diekstraksi secara utuh. Hal ini mengindikasikan bahwa Discord cenderung mempertahankan representasi teks Unicode secara lebih lengkap dibandingkan media lain, meskipun mengubah cara pesan ditampilkan kepada pengguna.

V. KESIMPULAN DAN SARAN

Berdasarkan hasil implementasi dan pengujian yang telah dilakukan, dapat disimpulkan bahwa steganografi teks berbasis Unicode dengan memanfaatkan *variation selector* pada emoji dapat direalisasikan secara efektif tanpa mengubah tampilan visual secara kasat mata pada kondisi tertentu. Implementasi yang dibangun pada penelitian ini mampu menyisipkan dan mengekstraksi data tersembunyi secara deterministik dengan memetakan setiap byte data ke satu karakter Unicode *variation selector*.

Hasil pengujian pada empat media komunikasi modern, yaitu LINE, WhatsApp, Telegram, dan Discord, menunjukkan bahwa keberhasilan steganografi Unicode sangat bergantung pada kebijakan pemrosesan teks dan mekanisme penanganan Unicode yang diterapkan oleh masing-masing media. Pada ukuran payload kecil hingga menengah, teknik ini umumnya dapat bertahan dan data tersembunyi dapat diekstraksi kembali dengan baik. Namun, pada payload berukuran besar, beberapa media menunjukkan keterbatasan berupa perubahan tampilan visual, pembatasan antarmuka, atau kegagalan dalam mempertahankan *variation selector* selama proses

transmisi pesan.

Dari sisi transparansi visual, penggunaan emoji sebagai karakter pembawa efektif dalam menyamarkan keberadaan data tersembunyi pada sebagian besar skenario pengujian. Namun demikian, perubahan perilaku antarmuka aplikasi, seperti mode *read more*, tampilan simbol pengganti, atau konversi pesan menjadi berkas teks, dapat menjadi indikator tidak langsung terhadap keberadaan payload steganografi berukuran besar.

Berdasarkan hasil tersebut, dapat disimpulkan bahwa steganografi Unicode berbasis emoji memiliki kapasitas praktis yang terbatas oleh karakteristik media komunikasi yang digunakan, meskipun secara teoritis mampu menyimpan data arbitrer. Oleh karena itu, teknik ini lebih sesuai digunakan untuk penyisipan pesan berukuran kecil dibandingkan untuk transmisi data berskala besar.

Sebagai saran untuk penelitian selanjutnya, evaluasi dapat diperluas dengan menguji teknik steganografi Unicode ini pada versi aplikasi dan sistem operasi yang berbeda, terutama lingkungan *mobile*. Selain itu, integrasi mekanisme kriptografi pada payload sebelum proses penyisipan dapat dipertimbangkan untuk meningkatkan aspek kerahasiaan data. Penelitian lanjutan juga dapat difokuskan pada pengembangan metode deteksi atau mitigasi steganografi Unicode sebagai bagian dari sistem keamanan dan moderasi konten pada media komunikasi modern.

VI. PRANALA KODE PROGRAM

Kode program implementasi dapat diakses pada pranala berikut: <https://github.com/Intermaze/emoji-stegano>

VII. UCAPAN TERIMA KASIH

Penulis mengucapkan syukur kepada Tuhan Yang Maha Esa atas berkat dan bimbingan-Nya dalam menyelesaikan makalah ini. Penulis juga mengucapkan terima kasih kepada Bapak Rinaldi Munir sebagai dosen mata kuliah IF4020 Kriptografi yang memberikan ilmu pengetahuan mengenai topik yang di bahas dalam makalah ini. Terakhir, Penulis mengucapkan terima kasih kepada Paul Butler atas artikel *Smuggling Arbitrary Data Through an Emoji* yang menjadi inspirasi utama dalam pengembangan ide dan implementasi steganografi Unicode pada makalah ini.

REFERENCES

- [1] “What is Unicode?” Diakses: 26 Desember 2025. [Daring]. Tersedia pada: <https://www.unicode.org/standard/WhatIsUnicode.html>
- [2] “UTR#17: Unicode Character Encoding Model.” Diakses: 26 Desember 2025. [Daring]. Tersedia pada: <https://www.unicode.org/reports/tr17/>
- [3] “Unicode - UTF-8, UTF-16 and UTF32,” GeeksforGeeks. Diakses: 26 Desember 2025. [Daring]. Tersedia pada: <https://www.geeksforgeeks.org/computer-organization-architecture/what-is-unicode/>
- [4] “UTS #51: Unicode Emoji.” Diakses: 26 Desember 2025. [Daring]. Tersedia pada: <https://www.unicode.org/reports/tr51/>
- [5] “Variation Selectors, Range: FE00–FE0F,” *Unicode Stand.*, vol. Version 17.0, [Daring]. Tersedia pada: <https://www.unicode.org/charts/PDF/UFE00.pdf>

- [6] “Variation Selectors Supplement, Range: E0100–E01EF,” *Unicode Stand.*, vol. Version 17.0, [Daring]. Tersedia pada: <https://www.unicode.org/charts/PDF/UE0100.pdf>
- [7] R. Munir, “Steganografi (Bagian 1)”, [Daring]. Tersedia pada: <https://informatika.stei.itb.ac.id/~rinaldi.munir/Kriptografi/2025-2026/08-Steganografi-Bagian1-2025.pdf>
- [8] P. Butler, “Smuggling arbitrary data through an emoji.” Diakses: 26 Desember 2025. [Daring]. Tersedia pada: <https://paulbutler.org/2025/smuggling-arbitrary-data-through-an-emoji/>

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, 26 Desember 2025



Rafiki Prawhira Harianto
13522065