# Keamanan *Email Client* Menggunakan *Hash-Based Message Authentication Code* dan *Pretty Good Privacy*

**Ayu Nur Oktaviani[1], Asep Id Hadiana[2], Melina[3]**
Program Studi Informatika
Universitas Jenderal Achmad Yani
Jl Terusan Jend. Sudirman, Cibeber, Cimahi, Jawa Barat 405314, Indonesia
e-mail: [1]ayunuroktaviani2002@gmail.com, [2]asep.hadiana@lecture.unjani.ac.id,
[3]melina@lecture.unjani.ac.id

Correspondence : ayunuroktaviani2002@gmail.com

Diajukan: 15 Agustus 2024; Direvisi: 23 Agustus 2024; Diterima: 24 Agustus 2024

***Abstrak***

*Dalam era teknologi informasi yang berkembang pesat, keamanan data dan privasi informasi telah menjadi hal yang sangat penting. Risiko penyalahgunaan dan serangan siber, pencurian data melalui jaringan komputer juga semakin meningkat. Upaya melindungi data pribadi menjadi sangat penting, termasuk pengamanan email untuk melindungi informasi dari akses yang tidak sah. Penelitian ini bertujuan untuk mengembangkan aplikasi email client dengan fitur keamanan tingkat lanjut yang menggabungkan dua metode keamanan yaitu Hash-Based Message Authentication Code (HMAC) dan Pretty Good Privacy (PGP). HMAC digunakan sebagai autentikasi pesan email dengan menghasilkan hash berdasarkan pesan yang ditulis, kemudian PGP digunakan untuk enkripsi pesan dan tanda tangan digital. Pengujian sistem dilakukan dengan menggunakan black box testing untuk verifikasi fungsi dengan spesifikasi yang ditetapkan. Selain itu juga, dilakukan pengujian entropy untuk menilai keacakan kunci PGP dengan hasil nilai tertinggi adalah 6.01139 bits dengan panjang kunci 4096 bit. Hasil penelitian ini menunjukkan bahwa sistem dalam penelitian ini berhasil menggabungkan HMAC dan PGP dalam proses pengiriman email. Penelitian ini berkontribusi dalam pengembangan keamanan komunikasi di era teknologi yang penuh dengan tantangan serangan siber.*

***Kata kunci:*** *Enkripsi, hash, HMAC, Pesan Email, PGP.*

***Abstract***

*In the era of rapidly developing information technology, data security and information privacy have become very important. The risk of misuse and cyber-attacks, data theft through computer networks is also increasing. Efforts to protect personal data have become very important, including email security to protect information from unauthorized access. This research aims to develop an email client application with advanced security features that combines two security methods, namely Hash-Based Message Authentication Code (HMAC) and Pretty Good Privacy (PGP). HMAC is used to authenticate email messages by generating hashes based on the messages written, then PGP is used for message encryption and digital signatures. An entropy test was conducted to assess the randomness of the key with the highest value result being 6.01139 bits with a key length of 4096 bits. The results of this study show that the system in this study successfully combines HMAC and PGP in the process of sending emails effectively. This research contributes to the development of communication security in a technological era full of cyberattack challenges.*

***Keywords****: Encryption, hash, HMAC, Email Message, PGP.*

## 1. Pendahuluan

Dalam era teknologi informasi yang berkembang pesat, keamanan data dan privasi informasi telah menjadi isu krusial. Risiko penyalahgunaan dan serangan siber yaitu tindakan ilegal yang dilakukan untuk mengakses, mengubah, merusak, atau mencuri data melalui jaringan komputer semakin meningkat [1]. Upaya melindungi data pribadi menjadi hal yang sangat penting untuk diterapkan, termasuk pengamanan email yang merupakan sarana pertukaran pesan bisnis, pribadi, dan sensitif [2]. Email yang dikirim tanpa

adanya perlindungan keamanan sangat rentan dicuri serta dibaca oleh pihak-pihak yang tidak bertanggung jawab [3].

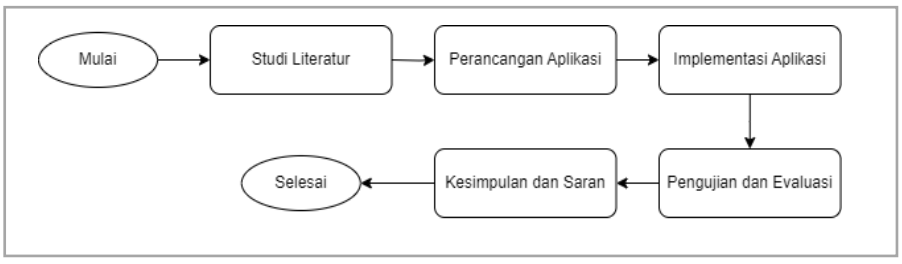
Pada tahun 2020, pencurian informasi melalui email terjadi beberapa kali di dibeberapa lembaga yang cukup besar, seperti 25.000 kredensial email lembaga besar *World Health Organization* (WHO), *National Institutes of Health* (NIH) dan *Gates Foundation* dicuri oleh kelompok Neonazi dan digunakan untuk menyebarkan berita tentang kampanye dan berbagi teori konspirasi tentang COVID-19 [4]. Apabila terjadi pencurian data dan seseorang berhasil memperoleh akses ke *server* email menggunakan kata sandi, orang tersebut dapat membaca email yang terkirim atau dapat mengirim pesan email palsu atas nama pemilik email [5]. Salah satu cara mengamankan informasi yang dikirim melalui email adalah dengan cara mengenkripsi [6]. Penerapan tunggal dari suatu teknik kriptografi memiliki risiko kebocoran data yang lebih tinggi dibandingkan dengan keamanan yang diperoleh melalui penggunaan lebih dari satu teknik kriptografi. Oleh karena itu, penting untuk menerapkan dua teknik kriptografi agar pesan email dapat diamankan secara optimal sebelum proses pengiriman [7].

Beberapa penelitian terdahulu yang mengkaji tentang keamanan email yaitu penelitian [8] tentang keamanan email menggunakan metode *Pretty Good Privacy* dan Algoritma *Rivest Shamir Aldeman* (RSA). Penelitian ini berhasil mengamankan email yang cukup baik dalam penggunaan metode *Pretty Good Privacy*. Penelitian [9] menggabungkan fungsi hash (SHA256) dengan kunci rahasia. Penelitian ini menunjukan bahwa keluaran dari algoritma SHA256 memiliki tingkat pengacakan yang bagus. Penelitian [10] mengungkapkan bahwa kombinasi Pretty Good Privacy (PGP) untuk menghasilkan kunci enkripsi dan *Hash-based Message Authentication Code* (HMAC) untuk otentikasi kunci adalah gabungan yang cocok untuk pengamanan email. Protokol yang diusulkan dalam penelitian ini menggunakan PGP untuk menghasilkan kunci publik dan kunci privat yang digunakan dalam enkripsi dan dekripsi pesan, sementara HMAC digunakan untuk memastikan keaslian kunci yang dikirim pengguna.

Berdasarkan penelitian-penelitian yang telah diuraikan, penelitian ini mengeksplorasi penggunaan dua teknologi keamanan yang kritis : *Pretty Good Privacy* (PGP) untuk menghasilkan kunci publik dan privat serta mengenkripsi dan *Hash-based Message Authentication Code* (HMAC) untuk authentikasi pesan email dengan tujuan untuk membantu dalam pemahaman, mengimplementasikan, dan menguji teknologi keamanan ini sehingga berkontribusi positif dalam pengembangan praktik keamanan email yang lebih baik dan membantu menjaga keamanan komunikasi di era digital yang penuh dengan tantangan serangan siber.

## 2. Metode Penelitian

Penelitian ini mengimplementasikan perancangan aplikasi email client dengan keamanan menggunakan HMAC dan PGP. Terdapat beberapa tahapan dalam pembangunan sistem pengamanan email client ini, pertama adalah lieteratur review yaitu membaca dan mengkaji beberapa literatur mengenai metode yang digunakan dengan cara mencari jurnal, buku, dan artikel terkait dengan HMAC dan PGP. Tahap kedua adalah pengembangan model yang dirancang ke dalam bentuk prototipe. Ini melibatkan pembuatan diagram dan spesifikasi teknis yang lebih detail. Tahap ketiga membuat perancangan aplikasi kemudian tahap keempat mengimplementasikannya menjadi perangkat lunak yang berjalan dengan baik. Tahap kelima dilakukan pengujian dan evaluasi terhadap aplikasi dengan menguji kekuatan kunci yang dihasilkan dan percobaan pembobolan proses pengiriman untuk memastikan sistem dengan keamanan siap digunakan. Metode penelitian ditunjukkan pada Gambar 1.




Gambar 1. Metode Penelitian

### A. *Hash-based Message Authentication Code (HMAC)*

*Hash-based Message Authentication Code* (HMAC) merupakan variasi dari *Message Authentication Code* (MAC) yang melibatkan penggunaan fungsi hash pada pesan. HMAC adalah metode autentikasi kriptografi yang memanfaatkan fungsi hash bersama dengan kunci rahasia. HMAC digabungkan dengan fungsi hash kriptografi apa pun, misalnya, md5, sha1, sha256. Fungsi hash digabungkan ke suatu kelas

sebagai salah satu parameter templat di HMAC dan kelas pembungkus hanya memiliki fungsi statis yang melibatkan fungsi hash [11]. Algoritma HMAC secara umum dijelaskan pada persamaan berikut.

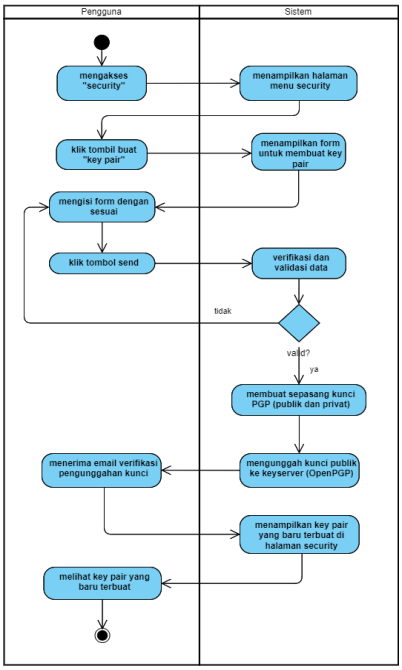
$$HMAC_K(m) = h\left((K \oplus opad)||h((K \oplus ipad)||m\right) \quad (1)$$

*B. Pretty Good Privacy* (PGP)

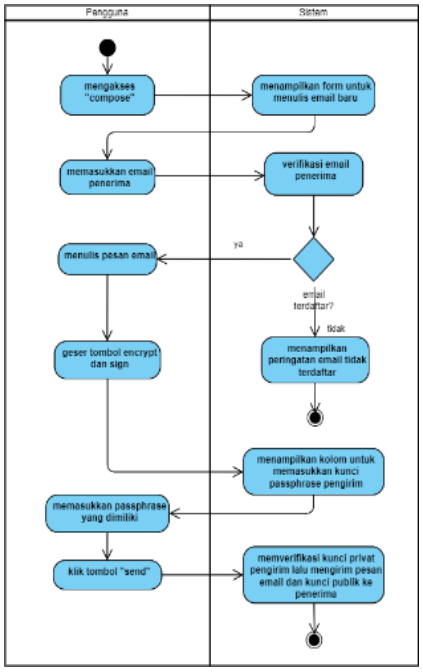
*Pretty Good Privacy* (PGP) adalah program komputer yang memungkinkan untuk mengirim pesan, email, atau file dengan menambahkan fitur kerahasiaan dan tambahan otentikasi pengguna yaitu dengan tanda tangan digital untuk memastikan isi pesan yang dikirim sesuai dengan yang diterima. PGP memiliki kunci sesi yang digunakan untuk mengenkripsi data dan pasangan kunci milik pengirim dan penerima suatu file [12]. Dasar-dasar PGP didasarkan pada konsep kriptografi kunci privat sebagai dasar otorisasi. Kunci ini digunakan untuk mengenkripsi komunikasi antara dua mesin. Untuk menjaga kerahasiaan data, kriptografi mengubah pesan *plaintext* menjadi *ciphertext* yang tidak dapat dikenali. *Ciphertext* ini kemudian dikirim oleh pengirim kepada penerima. PGP menghasilkan dua kunci saat pembuatan kunci: kunci privat (*private key*) dan kunci publik (*public key*), dimana kunci publik diumumkan secara luas. Seseorang yang ingin mengirim pesan kepada orang lain harus mencari kunci publik orang tersebut di suatu situs. Kunci publik ini digunakan untuk mengenkripsi pesan, karena hanya penerima yang memiliki kunci privat untuk mendekripsi pesan tersebut, maka pesan *ciphertext* yang melewati jaringan tetap aman dari penyusup. Salah satu metode yang digunakan untuk menciptakan pasangan kunci publik dan kunci privat adalah metode *Rivest, Shamir, Adleman* (RSA) [10][13].

### 2.1. Perancangan Aplikasi

Perancangan aplikasi dibuat menggunakan *activity* diagram untuk proses membuat pasangan kunci publik dan privat dan proses mengirim serta menerima pesan email.




Gambar 2. *Activity* Diagram *Security*




Gambar 3. *Activity* Diagram Pengiriman Email

### 2.2. Pengujian

Dilakukan pengujian untuk mengecek keacakan pasangan kunci publik dan privat yang dihasilkan menggunakan *shannon entropy*. Metode ini memiliki ukuran ruang kunci K dengan nilai entropi terbaik adalah 8. Semakin besar nilai entropi, semakin sulit untuk memecahkan *ciphertext*. Perhitungan entropi dilakukan menggunakan persamaan berikut [14].

$$H(X) = -\sum_{i=0}^{n} a_i \, 2\log(p(S_i)) \quad (2)$$

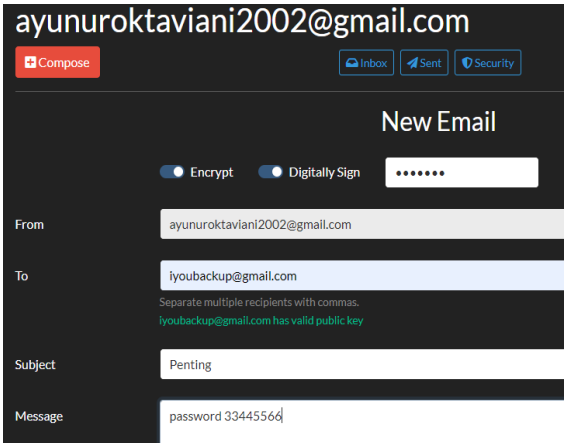
Dimana X merupakan pesan (kunci) yang akan dihitung keacakannya, $S_i$ sebagai simbol pesan, $p(S_i)$ untuk peluang terjadinya $S_i$, dan $a_i$ merupakan banyaknya kemunculan $S_i$.

## 3. Hasil dan Pembahasan

### 3.1. Hasil Implementasi Aplikasi

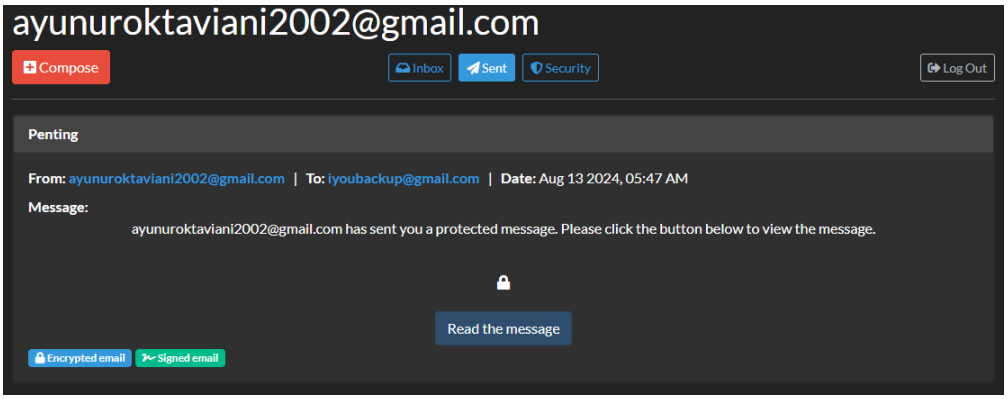
Aplikasi dikembangkan menggunakan *framework* django yang menghasilkan aplikasi *email client* dengan fitur pembuatan kunci publik dan privat, mengirim dan menerima email dengan keamanan enkripsi dan dekripsi menggunakan HMAC dan PGP.

A. Fitur Mengirim Pesan Email

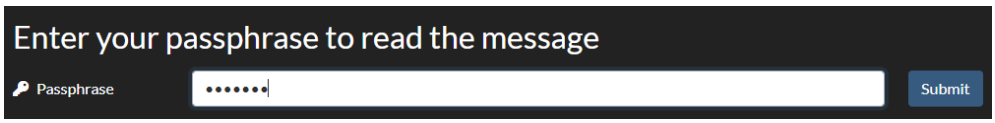



Gambar 4. Menulis Pesan Email

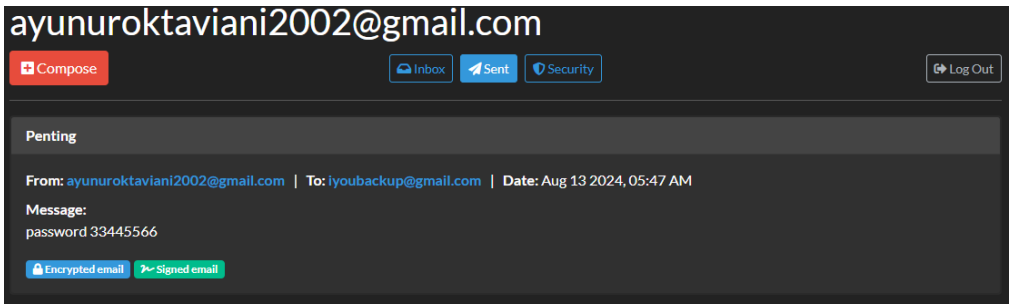
B. Fitur Membuka Email dengan Keamanan




Gambar 5. Membuka Pesan Email




Gambar 2. Memasukkan kunci *Passphrase* yang telah dibuat pada *Generate Key*




Gambar 6. Pesan Email dengan Kemanan dapat Terbaca

### 3.2. Hasil Pengujian

#### 3.2.1. *Black Box Testing*

Pengujian *black box* dilakukan untuk menguji proses membuat kunci, mengirim email, dan menerima email. Hasil pengujian *black box* ditunjukkan pada Tabel 1.

Tabel 1. Hasil Pengujian Black Box.

| Skenario | Hasil Diharapkan | Hasil |
|---|---|---|
| Pengguna mengisi form membuat kunci kemudian menekan tombol *generate key pair* | Menghasilkan kunci publik dan kunci privat | Diterima |
| Pengirim menulis email kemudian mengaktifkan fitur *encrypt* dan sign lalu menekan tombol *send* | Menghasilkan pesan terenkripsi dan mengirimnya ke pengguna yang dituju | Diterima |
| Penerima menerima email masuk pada fitur inbox dan membukanya menngunakan kunci yang telah dibuat sebelumnya | Pesan dengan keamanan dapat terlihat | Diterima |

Berdasarkan asil pengujian terhadap perangkat lunak Sistem *Email Client* menggunakan *black box testing* dapat disimpulkan bahwa sistem berjalan sesuai dengan spesifikasi yang telah ditetapkan.

#### 3.2.2. Pengujian Keamanan

Pengujian dilakukan menggunakan *tools* berupa *Online Calculator* dari planetcalc.com. Pengujian dilakukan pada kunci privat dan publik yang dihasilkan PGP. *Tools* akan menghitung entropy kunci dengan hasil berupa bits. Perhitungan entropy dilakukan pada 5 pasang kunci yang dihasilkan PGP pada sistem email client ini. Hasil perhitungan *entropy* menyimpulkan bahwa nilai tertinggi adalah 6.01139 dengan ukuran kunci terpanjang yaitu 4096. Ini menandakan semakin panjang ukuran kunci maka semakin tinggi nilai *entropy*.

## 4. Kesimpulan

Penggabungan dua metode keamanan *Hash-Based Message Authentication Code* (HMAC) dan *Pretty Good Privacy* (PGP), berhasil diaplikasikan pada sistem email client. HMAC digunakan untuk menghasilkan hash berdasarkan pesan email yang ditulis dan PGP digunakan untuk mengenkripsi pesan dan tanda tangan digital dengan cara menghasilkan sepasang kunci (publik dan privat) yang mana kunci publik digunakan untuk mengirim pesan email dan kunci privat digunakan untuk membuka pesan email yang diterima dengan memasukkan *passphrase*. Penggabungan keduanya dapat mengamankan pesan email secara maksimal.

Berjalan dengan baiknya sistem dibuktikan dengan pengujian *black box* dengan hasil 100% sesuai dengan spesifikasi yang telah ditetapkan. Pengujian keamanan dilakukan dengan menghitung nilai entropy dengan hasil nilai tertinggi adalah 6.01139 bits dengan panjang kunci 4096 bit. Penelitian selanjutnya dapat meningkatkan ukuran kunci PGP yang dihasilkan agar nilai entropy mencapai 8 bits sehingga kunci sangat acak supaya tidak mudah diretas.

## Daftar Pustaka

[1] C. Umam, L. B. Handoko, C. A. Sari, E. H. Rachmawanto, and L. A. R. Hakim, “Kombinasi

Vigenere dan Autokey Cipher dalam Proses Proteksi SMS Berbasis Android," *Pros. Sains Nas. dan Teknol.*, vol. 12, no. 1, p. 492, 2022, doi: 10.36499/psnst.v12i1.7108.

[2] S. Kurniawan, "Peranan Pemerintah Melalui Undang-Undang Perlindungan Data Pribadidalam Menanggulangi Phisingdi Indonesia," *Innov. J. Soc. Sci. Res.*, vol. 3, 2023, doi: 10.5040/9781635577068-0537.

[3] D. Abdullah and Surnihayati, "Pengamanan Email Menggunakan Metode Vigenere Chiper," *JISAMAR (Journal Inf. Syst. Applied, Manag. Account. Res.*, vol. 1, no. November, pp. 1–9, 2017.

[4] R. Wijaya, "Implementasi Algoritma Aes Dan Rc4 Untuk Pengamanan Pesan Email," vol. 11, no. 2, pp. 64–71, 2020.

[5] J. David, "15 Reasons to Use Encrypted Emails for Company Communication," 2021. https://fleep.io/blog/encrypted-emails-for-company-communication/ (accessed Apr. 04, 2024).

[6] D. Nurani, "Perancangan Aplikasi Email Menggunakan Algoritma Caesar CIPHER dan Base64," *JISKA (Jurnal Inform. Sunan Kalijaga)*, vol. 2, no. 3, p. 175, 2018, doi: 10.14421/jiska.2018.23-07.

[7] M. I. Zulfikar, G. Abdillah, and A. Komarudin, "Kriptografi untuk Keamanan Pengiriman Email Menggunakan Blowfish dan Rivest Shamir Adleman (RSA)," *Semin. Nas. Apl. Teknol. Inf.*, pp. 19–26, 2019.

[8] R. I. Ananda, Fauziah, and N. Hayati, "Keamanan Email Menggunakan Metode Pretty Good Privacy Dengan Algoritma Rsa," *J. Ilm. Inform. Komput.*, vol. 25, no. 3, pp. 213–224, 2020, doi: 10.35760/ik.2020.v25i3.3118.

[9] M. Ichwan, M. Gustian, and N. R. Nurjaman, "Implementasi Keyed-Hash Message Authentatication Code Pada Sistem Keamanan Rumah," *MIND J.*, vol. 1, no. 1, p. 9, 2018, doi: 10.26760/mindjournal.v1i1.9.

[10] T. Maier, "Automated Key Management for End-To-End Encrypted Email Communication," no. section III.

[11] D. Ayu, M. Kirana, and P. Dewi, "Analisis Penggunaan HMAC - SHA256 pada Keamanan Aplikasi Chatting," no. 18220084, 2023.

[12] E. Barker and A. Roginsky, "Transitioning the Use of Cryptographic Algorithms and Key Lengths," *NIST Spec. Publ. 800-131A Revis. 2*, no. March, pp. 17–18, 2019, [Online]. Available: https://doi.org/10.6028/NIST.SP.800-131Ar2

[13] M. Melina, F. Sukono, H. Napitupulu, and V. A. Kusumaningtyas, "Verifikasi Tanda Tangan Elektronik dengan Teknik Otentikasi Berbasis Kriptografi Kunci Publik Sistem Menggunakan Algoritma Kriptografi Rivest-Shamir-Adleman," *J. Mat. Integr.*, vol. 18, no. 1, p. 27, 2022, doi: 10.24198/jmi.v18.n1.38343.27-39.

[14] R. Ravida and H. A. Santoso, "Advanced Encryption Standard (AES) 128 Bit untuk Keamanan Data Internet of Things (IoT) Tanaman Hidroponik," *J. RESTI (Rekayasa Sist. dan Teknol. Informasi)*, vol. 4, no. 6, pp. 1157–1164, 2020, doi: 10.29207/resti.v4i6.2478.