Rancang Bangun Sistem Pengamanan *E-Mail* Berbasis Android Menggunakan Pendekatan *Hybrid Cryptosystem*

Andri[1*], **Rudy Prasetya**[2], **Sepniyanti**[3]
[1,2,3]Program Studi Teknik Informatika, Universitas Indraprasta PGRI
andriecitra@gmail.com*

e-ISSN: 2987-811X
MARAS: Jurnal Penelitian Multidisplin
https://ejournal.lumbungpare.org/index.php/maras
Vol. 4 No. 1 Maret 2026
Page: 159-173

Article History:
Received: 03-03-2026
Accepted: 07-03-2026

Abstrak *: Transformasi di bidang teknologi informasi telah menggeser secara signifikan pola masyarakat dalam mengakses berbagai layanan internet. Jika pada masa lalu penggunaan layanan seperti surat elektronik, penelusuran web, dan percakapan daring masih bergantung pada komputer dengan jaringan kabel, saat ini aktivitas tersebut dapat dilakukan secara fleksibel melalui perangkat bergerak seperti notebook, smartphone, maupun tablet. Pada sistem operasi Android, salah satu layanan yang paling sering dimanfaatkan adalah e-mail, terutama karena dukungan fitur push e-mail yang memungkinkan pesan diterima secara langsung dan real-time. E-mail telah menjadi media komunikasi utama dalam pertukaran informasi dan kerja sama, baik pada tingkat individu, komunitas, lembaga pendidikan, maupun perusahaan. Dalam praktiknya, pesan yang dikirimkan melalui e-mail sering kali memuat data yang bersifat pribadi atau rahasia, sehingga membutuhkan mekanisme perlindungan yang memadai agar tidak dapat diakses oleh pihak yang tidak berhak. Penelitian ini menawarkan suatu pendekatan pengamanan komunikasi e-mail pada perangkat bergerak berbasis Android melalui penerapan hybrid cryptosystem. Sistem kriptografi hibrida tersebut menggabungkan algoritma kriptografi simetris dan asimetris, fungsi hash, serta mekanisme pembangkitan kunci acak dalam satu kerangka keamanan terpadu. Dengan pendekatan ini, diharapkan seluruh prinsip keamanan informasi meliputi confidentiality, integrity, authentication, dan nonrepudiation dapat terpenuhi secara optimal, sehingga proses pertukaran pesan elektronik menjadi lebih aman dan terpercaya.*

Kata Kunci *: Email; Android; Linux; Internet; Hybrid Cryptosystem*

PENDAHULUAN

Seiring meningkatnya tingkat mobilitas masyarakat, fungsi perangkat bergerak sebagai sarana komunikasi dan pertukaran informasi mengalami perkembangan yang sangat signifikan. Dalam beberapa tahun terakhir, Android menjadi salah satu

platform sistem operasi *mobile* yang paling banyak digunakan oleh masyarakat (Suwarno et al., 2023). Android merupakan sistem operasi berbasis Linux yang bersifat *open source* dan dirancang untuk perangkat bergerak, serta memberikan fleksibilitas dan kebebasan bagi pengembang dalam menciptakan dan mengembangkan aplikasi mobile (Pradana et al., 2022). Salah satu layanan internet yang banyak dimanfaatkan pada perangkat Android adalah surat elektronik (*e-mail*), khususnya karena dukungan fitur *push e-mail*. Melalui perangkat *mobile* yang terhubung ke jaringan internet, layanan ini dapat diakses oleh berbagai lapisan pengguna untuk bertukar informasi serta menjalin kerja sama, baik dalam lingkup pribadi, komunitas, lembaga, maupun perusahaan (Zahay & Roberts, 2023). Dalam praktiknya, pemanfaatan *e-mail* untuk keperluan komunikasi dan kerja sama tidak terbatas pada penyampaian informasi biasa, melainkan juga mencakup data yang bersifat rahasia. Informasi dengan tingkat sensitivitas tinggi tersebut memerlukan perlindungan khusus, karena apabila disalahgunakan oleh pihak yang tidak berhak, dapat menimbulkan dampak merugikan bagi individu maupun organisasi terkait.

Dalam proses pengiriman informasi rahasia melalui *e-mail*, kemudahan akses dan penggunaan yang ditawarkan harus diiringi dengan penerapan langkah-langkah pengamanan yang memadai guna meminimalkan risiko penyalahgunaan (Kaylor et al., 2023). Kondisi tersebut terjadi karena internet pada dasarnya merupakan jaringan terbuka yang memiliki tingkat kerentanan terhadap penyalahgunaan serta pemanfaatan data oleh pihak yang tidak memiliki otoritas (Andrushia et al., 2024). Selain itu, sistem *e-mail* pada dasarnya tidak memberikan perlindungan terhadap integritas pesan, sehingga isi *e-mail* dapat dimodifikasi baik saat pengiriman maupun ketika tersimpan di server tanpa adanya deteksi (Remch et al., 2024). Di samping itu, sistem *e-mail* pada umumnya belum menyediakan mekanisme verifikasi identitas pengirim yang memadai. Kondisi ini menyebabkan tidak adanya kepastian bahwa pesan benar-benar dikirim oleh pihak yang tercantum pada alamat pengirim, serta membuka peluang bagi pengirim untuk melakukan penyangkalan terhadap pesan yang telah dikirimkan (Guo et al., 2024). Dalam upaya menjamin perlindungan data rahasia yang dikirimkan melalui *e-mail*, berbagai strategi dan prosedur pengamanan perlu diimplementasikan secara sistematis (Diqi et al., 2023). Dalam sistem keamanan informasi, terdapat empat pilar utama yang harus dipenuhi, yaitu menjaga kerahasiaan, memastikan integritas data, memverifikasi keaslian identitas, dan mencegah terjadinya penyangkalan atas proses komunikasi yang telah berlangsung.

METODE PENELITIAN

Studi ini berfokus pada perancangan mekanisme keamanan untuk komunikasi *e-mail* di lingkungan Android dengan memanfaatkan pendekatan *hybrid cryptosystem*, sehingga aspek-aspek utama keamanan informasi dapat diimplementasikan secara menyeluruh dan optima (Narasimharao et al., 2025). Guna mencapai sasaran yang telah ditetapkan, studi ini mengadopsi pendekatan penelitian dan pengembangan (*research and development*) sebagai metode utama (Handayani et al., 2022). Metode tersebut dimanfaatkan untuk menghasilkan luaran berupa produk tertentu serta menguji sejauh mana efektivitas implementasinya (Ordonez, 2006). Hasil dari metode ini tidak terbatas pada pengembangan produk berwujud seperti buku ajar, modul, dan perangkat laboratorium, melainkan juga mencakup pengembangan perangkat lunak,

antara lain aplikasi pengolahan data, sistem pembelajaran terintegrasi, perpustakaan digital, laboratorium virtual, serta model-model yang digunakan dalam konteks pendidikan, penilaian, dan pengelolaan organisasi.

HASIL DAN PEMBAHASAN

Surat Elektronik (*Electronic Mail*)

1. Tinjauan Konseptual tentang *E-Mail*

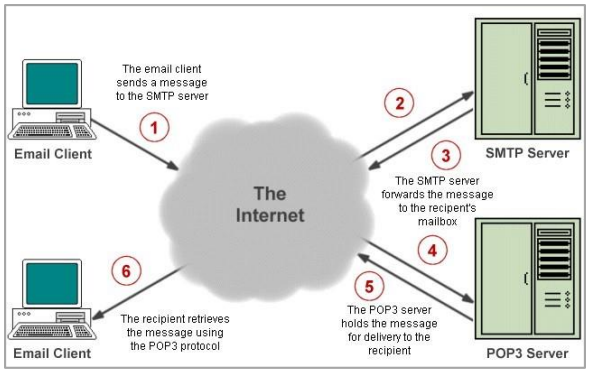
   Surat elektronik atau *e-mail* merupakan salah satu layanan internet yang paling populer dan banyak dimanfaatkan, baik untuk keperluan individu, kelompok, institusi, maupun perusahaan (Yeu et al., 2021). *E-mail* memungkinkan pertukaran informasi atau pengiriman pesan antara orang-orang yang terpisah secara geografis melalui perangkat komunikasi. Cara kerja *e-mail* memiliki kemiripan dengan pengiriman surat secara konvensional melalui layanan pos (Eberhart & Shi, 2001). Pengguna dapat menulis pesan layaknya menulis surat di atas kertas, kemudian memasukkannya ke dalam "amplop" digital (Ryan, 2020). Dengan mencantumkan nama dan alamat tujuan yang benar, pesan tersebut dapat dikirim dan diharapkan sampai ke penerima yang tepat. Apabila alamat pengirim juga dicantumkan, maka penerima akan mudah membalas pesan dengan mengarahkan surat balasan ke alamat tersebut. Sebagaimana mekanisme pengiriman surat konvensional melalui jasa pos, *e-mail* memiliki fungsi utama untuk menyampaikan pesan. Perbedaannya terletak pada media yang digunakan, yaitu bukan lagi kertas, melainkan aplikasi berbasis komputer yang memanfaatkan jaringan komputer atau internet sebagai media transmisinya. Melalui *e-mail*, seseorang dapat menulis pesan dalam bentuk teks maupun melampirkan *file* (*attachment*), lalu menentukan penerima dengan mencantumkan alamat *e-mail* tujuan sebelum mengirimkannya (Rifai et al., n.d.). Proses pengiriman *e-mail* ini serupa dengan mengirim surat melalui pos, di mana pesan yang telah dikirim akan masuk ke kotak masuk (*inbox*) penerima, dan penerima dapat membaca pesan tersebut setelah memeriksanya.

2. Skema Distribusi Pesan pada Sistem *E-Mail*

   Pertukaran pesan melalui *e-mail* dijalankan berdasarkan arsitektur protokol TCP/IP, sementara format penyusunan pesan mengikuti ketentuan IMF (*Internet Message Format*) yang mengatur komponen *header* sebagai pembungkus konten utama (Buhalis & Law, 2008). Pada sistem *e-mail*, mekanisme pengiriman pesan mengandalkan SMTP (*Simple Mail Transfer Protocol*). Pesan yang dibuat menggunakan *Mail User Agent* (MUA) selanjutnya dikirim ke *Mail Transfer Agent* (MTA) sebagai server pengelola yang mengatur distribusi pesan ke server penerima (Punniyakotti et al., 2023). MUA merupakan bagian dari sistem *e-mail* yang berinteraksi langsung dengan pengguna, berfungsi untuk mengelola pesan masuk dan keluar (Martel et al., 2024). Adapun MTA berperan dalam mengelola proses pemindahan pesan elektronik antar sistem atau server sampai pesan tersebut diterima oleh sistem tujuan.

   Sesudah *e-mail* diproses oleh server asal, pesan akan dialihkan menuju server tujuan penerima, yang biasanya mengimplementasikan POP3 (*Post Office Protocol* versi 3). Dengan dukungan protokol tersebut, klien *e-mail* pada sisi pengguna akan mengakses dan mengunduh pesan dari kotak surat yang tersimpan di server. Penting untuk dicatat bahwa penerimaan pesan hanya dapat dilakukan apabila

akun pengguna telah terdaftar secara *valid* pada server *e-mail* atau server POP3 yang bersangkutan. Secara konseptual, alur pengiriman dan penerimaan *e-mail* dapat direpresentasikan melalui diagram sebagaimana ditampilkan pada bagian berikut.



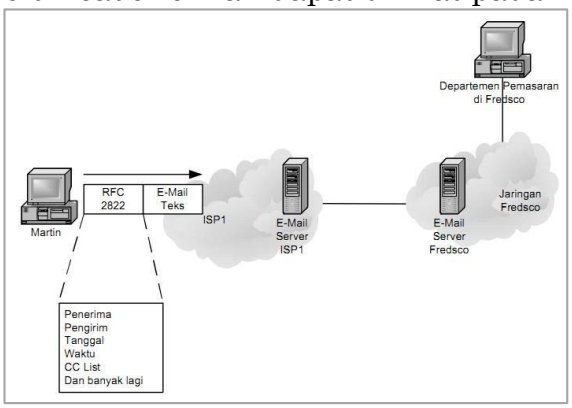
Gambar 1. Mekanisme Pengiriman dan Penerimaan *E-Mail*

3. Protokol dan Standar *E-mail*

E-mail menggunakan beragam protokol dan standar yang berperan penting dalam proses pengiriman serta penerimaan pesan, sehingga pesan dapat diterima oleh penerima yang dituju secara efektif. Berikut ini adalah beberapa protokol dan standar utama yang umumnya digunakan pada layanan *e-mail* (Carrera-Rivera et al., 2022). Kehadiran protokol-protokol tersebut tidak hanya menjamin pengantaran pesan, tetapi juga memungkinkan berbagai sistem dan aplikasi *e-mail* yang berbeda bisa saling berkomunikasi dengan baik.

a. Internet *Message* Format Berdasarkan Ketentuan RFC 2822

Ketika seseorang mengirimkan sistem *e-mail* dan infrastruktur server pendukung meneruskan pesan tersebut, yang dikirimkan melalui jaringan bukan hanya isi pesan dalam bentuk teks, tetapi juga bagian *header e-mail*. *Header* ini memuat berbagai informasi penting, seperti alamat *e-mail* pengirim dan penerima, tanggal, waktu pengiriman, daftar *Carbon Copy* (CC), serta informasi tambahan lainnya (Sachar & Kumar, 2021). Informasi pada *header* inilah yang membantu sistem *e-mail* dalam proses pengiriman, distribusi, dan pencatatan pesan. Susunan detail dari *field header e-mail* dapat dilihat pada ilustrasi berikut.



Gambar 2. Struktur *Header E-Mail* Mengacu pada RFC 2822

b. RFC 821 sebagai Spesifikasi *Simple Mail Transfer Protocol* (SMTP)

SMTP merupakan salah satu protokol yang paling umum digunakan dalam sistem *e-mail*. Dalam mekanisme pengiriman pesan, baik aplikasi klien maupun

server *e-mail* memanfaatkan protokol ini melalui port 25 untuk mengatur proses distribusi dan penerimaan pesan. Ketika sebuah *e-mail* hendak dikirim, klien SMTP pada aplikasi akan membangun koneksi komunikasi dua arah dengan server SMTP. Server tersebut dapat berperan sebagai tujuan akhir pengiriman, sebagai penghubung antara sistem pengirim dan penerima, maupun sebagai gerbang yang menjembatani komunikasi SMTP dengan protokol server lainnya (Ribeiro et al., 2024). Dapat ditegaskan bahwa SMTP memegang peran penting dalam proses pengiriman surat elektronik. Protokol ini mengelola pertukaran pesan antara klien dan server melalui koneksi dua arah yang umumnya menggunakan port 25. Server SMTP tidak hanya berfungsi sebagai tujuan akhir pesan, tetapi juga dapat berperan sebagai perantara serta penghubung dengan protokol lainnya. Oleh karena itu, SMTP menjadi elemen kunci dalam memastikan proses distribusi e-mail berlangsung secara efektif dan terstruktur.

Proses komunikasi antara SMTP *client* dan server dimulai dengan tahap inisiasi koneksi. Pada tahap ini, server SMTP memberikan respons mengenai ketersediaan layanannya. Apabila layanan tidak dapat diakses, maka hubungan komunikasi akan langsung dihentikan. Sebaliknya, jika server siap melayani, klien SMTP akan melanjutkan proses dengan mengirimkan perintah-perintah yang diperlukan, seperti identitas pengirim, alamat tujuan, serta konten pesan.

Setelah seluruh data pesan berhasil diteruskan ke server SMTP, klien memiliki opsi untuk menutup sesi komunikasi atau melanjutkannya guna mengirimkan pesan lainnya. Perlu dipahami bahwa tanggung jawab SMTP *client* terbatas pada proses penyerahan pesan kepada server SMTP beserta konfirmasi bahwa transmisi telah selesai. Hal tersebut tidak serta-merta menjamin bahwa pesan sudah benar-benar diterima oleh pihak penerima akhir (Tian et al., 2020).

c. RFC 1939 sebagai Spesifikasi *Post Office Protocol versi* 3 (POP3)

POP3 adalah protokol yang digunakan untuk mengambil *e-mail* dari *mailbox* di server dan menyimpannya ke komputer lokal pengguna. Protokol ini berjalan menggunakan port 110 pada jaringan TCP/IP (Cao et al., 2024). Untuk menggunakan layanan POP3, aplikasi klien terlebih dahulu melakukan koneksi ke server POP3. Setelah hubungan komunikasi terbentuk, server akan memberikan respons awal sebagai tanda kesiapan layanan. Selanjutnya, dilakukan tahap autentikasi, di mana klien wajib mengirimkan identitas pengguna berupa *user id* dan kata sandi guna melakukan verifikasi ke server. Jika proses otentikasi berhasil dan akun pengguna terdaftar pada server POP3, maka pengguna dapat mengakses dan mengambil *e-mail* yang dibutuhkan (Beyramysoltan et al., 2022). Tahap berikutnya adalah fase transaksi, yaitu saat pengguna dapat mengeksekusi berbagai perintah untuk berinteraksi dengan server POP3, misalnya menampilkan daftar pesan yang tersimpan dalam *mailbox*. Seluruh pertukaran data antara klien dan server POP3 dilakukan dalam bentuk kode ASCII, sedangkan struktur pesan *e-mail* yang digunakan mengacu pada ketentuan standar RFC 822.

Sehingga POP3 merupakan protokol yang berfungsi untuk mengakses dan mengunduh *e-mail* dari server ke perangkat lokal pengguna melalui jaringan TCP/IP dengan memanfaatkan port 110. Proses penggunaannya diawali dengan pembentukan koneksi antara klien dan server, dilanjutkan dengan tahap

autentikasi untuk memverifikasi identitas pengguna. Setelah berhasil terverifikasi, pengguna dapat memasuki fase transaksi guna mengelola dan mengambil pesan yang tersimpan dalam *mailbox*. Seluruh komunikasi berlangsung dalam format ASCII dan mengikuti standar struktur pesan yang ditetapkan dalam RFC 822. Dengan demikian, POP3 berperan penting dalam mekanisme penerimaan dan pengelolaan e-mail pada sisi pengguna.

Hybrid Cryptosytem

Protokol kriptografi merupakan suatu kerangka prosedural yang dirancang dengan mengintegrasikan berbagai algoritma kriptografi untuk mencapai tujuan keamanan tertentu. Protokol ini tidak hanya mengatur proses enkripsi dan dekripsi, tetapi juga mendefinisikan tahapan komunikasi yang harus dilakukan oleh pihak-pihak yang terlibat. Dalam konteks komunikasi rahasia, protokol kriptografi berfungsi untuk melindungi pesan dari ancaman penyadapan, manipulasi, maupun penyalahgunaan oleh pihak yang tidak memiliki hak akses (Paul et al., 2023) Pengembangan protokol kriptografi dapat disesuaikan dengan kebutuhan sistem, seperti distribusi kunci rahasia, pertukaran data sensitif, pembangkitan bilangan acak kriptografis, hingga proses autentikasi identitas pengguna. Salah satu pendekatan yang banyak digunakan adalah *hybrid cryptosystem*, yaitu metode yang mengombinasikan algoritma kriptografi simetris dan asimetris dalam satu mekanisme terpadu. Dalam skema ini, algoritma asimetris umumnya digunakan untuk proses pertukaran kunci secara aman, sedangkan algoritma simetris dimanfaatkan untuk mengenkripsi data karena memiliki efisiensi komputasi yang lebih tinggi. Dengan menggabungkan kedua pendekatan tersebut, *hybrid cryptosystem* mampu memberikan keseimbangan antara tingkat keamanan dan performa sistem, sehingga lebih efektif diterapkan pada lingkungan komunikasi digital seperti *e-mail* berbasis Android.

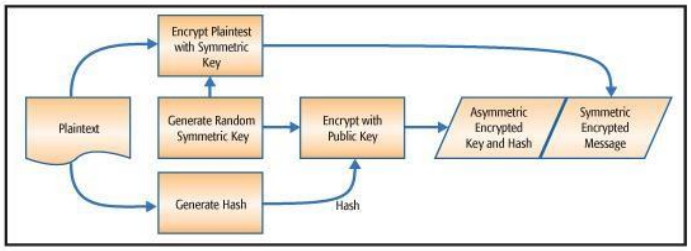
Dalam praktik modern, protokol kriptografi umumnya memadukan algoritma simetris dan asimetris sehingga membentuk suatu mekanisme yang dikenal sebagai *hybrid cryptosystem* (Lhamo et al., 2023). Kombinasi ini dirancang untuk mengoptimalkan aspek keamanan sekaligus efisiensi komputasi dalam proses pertukaran data. Secara konseptual, algoritma asimetris seperti RSA digunakan pada tahap awal komunikasi untuk melakukan pertukaran kunci secara aman (*secure key exchange*). Pada tahap ini, pengirim mengenkripsi kunci simetris menggunakan kunci publik milik penerima. Selanjutnya, penerima akan mendekripsi kunci simetris tersebut menggunakan kunci privatnya. Setelah proses *key establishment* berhasil, komunikasi data utama dilakukan menggunakan algoritma simetris seperti AES, yang memiliki keunggulan dalam kecepatan enkripsi dan dekripsi dibandingkan algoritma asimetris.

Dengan mekanisme tersebut, *hybrid cryptosystem* mampu mengatasi permasalahan distribusi kunci yang menjadi kelemahan sistem simetris murni, sekaligus mengurangi beban komputasi yang biasanya muncul pada sistem asimetris murni. Pendekatan ini menjadikan sistem lebih aman dalam menjaga kerahasiaan pesan serta lebih efisien untuk diterapkan pada komunikasi digital, termasuk layanan *e-mail* berbasis Android.

Konsep *hybrid cryptosystem* dikembangkan sebagai respons terhadap keunggulan dan keterbatasan yang dimiliki oleh algoritma kriptografi simetris maupun asimetris. Dari aspek performa, algoritma simetris dikenal memiliki kecepatan

pemrosesan yang sangat tinggi, bahkan dapat mencapai ratusan hingga ribuan kali lebih cepat dibandingkan algoritma asimetris dalam proses enkripsi dan dekripsi (Arif & Nurokhman, 2023). Meskipun demikian, pendekatan simetris memiliki kelemahan mendasar pada tahap distribusi kunci, karena kunci yang sama harus dibagikan kepada seluruh pihak yang terlibat dalam komunikasi. Tahapan pertukaran kunci ini berpotensi menjadi titik lemah yang dapat dimanfaatkan oleh pihak yang tidak berwenang untuk memperoleh akses terhadap informasi rahasia (UNWTO, 2018). Sebaliknya, algoritma asimetris dirancang untuk mengatasi persoalan tersebut dengan menyediakan mekanisme pertukaran kunci yang lebih aman melalui penggunaan pasangan kunci publik dan privat. Dengan memadukan kedua pendekatan tersebut, *hybrid cryptosystem* mampu mengoptimalkan kecepatan pemrosesan sekaligus menjaga keamanan distribusi kunci. Integrasi ini menghasilkan sistem yang lebih seimbang dalam hal efisiensi dan perlindungan data, sehingga komunikasi digital dapat berlangsung dengan tingkat keamanan yang lebih tinggi.

Selain tantangan dalam distribusi kunci, sistem komunikasi juga menghadapi persoalan lain, seperti verifikasi keaslian pesan, validasi identitas entitas yang terlibat, serta pencegahan penyangkalan atas tindakan komunikasi yang telah dilakukan (Informasi et al., 2019). Oleh sebab itu, berbagai protokol kriptografi modern umumnya menggabungkan beberapa teknik sekaligus, termasuk algoritma simetris, algoritma asimetris, fungsi hash, serta mekanisme pembangkitan kunci acak. Secara konseptual, integrasi seluruh komponen tersebut dalam skema *hybrid cryptosystem* dapat direpresentasikan melalui ilustrasi arsitektur sistem pada bagian berikut.



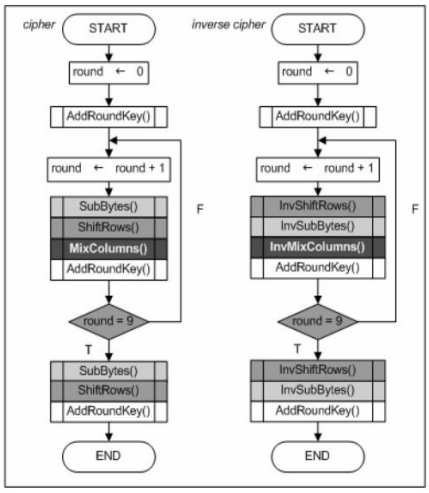
Gambar 3. Arsitektur dan Mekanisme *Hybrid Cryptosystem*

Advanced Encryption Standard (AES) sebagai Algoritma Kriptografi Simetris

Algoritma Rijndael secara resmi ditetapkan oleh *National Institute of Standards and Technology* (NIST) sebagai standar kriptografi nasional melalui dokumen *Federal Information Processing Standards* (FIPS) 197 pada tanggal 26 November 2001. Penetapan tersebut merupakan hasil dari proses seleksi dan evaluasi yang berlangsung selama kurang lebih lima tahun. Dari total 15 kandidat algoritma enkripsi yang diajukan dalam kompetisi terbuka, Rijndael dinilai paling memenuhi kriteria keamanan, efisiensi, serta fleksibilitas implementasi, sehingga dipilih sebagai dasar pembentukan *Advanced Encryption Standard* (AES). Standar AES kemudian digunakan secara luas untuk melindungi sistem komputer dan informasi penting, khususnya pada lingkungan pemerintahan federal Amerika Serikat yang memerlukan tingkat keamanan kriptografi yang tinggi.

Dalam operasionalnya, AES mengenkripsi data melalui empat transformasi utama yang saling terintegrasi. Tahap pertama adalah substitusi nonlinier yang memanfaatkan tabel substitusi khusus untuk menghasilkan perubahan nilai *byte* secara kompleks. Tahap kedua berupa pergeseran atau permutasi posisi *byte* dalam

blok data guna meningkatkan penyebaran informasi. (Sasmal et al., 2024). Selanjutnya, proses difusi dilakukan dengan mengombinasikan *byte-byte* dalam satu kolom menggunakan operasi matematis linier sehingga perubahan kecil pada *input* akan memengaruhi keseluruhan blok. Tahap terakhir adalah penambahan kunci (*key addition*), yaitu penggabungan data dengan kunci enkripsi menggunakan operasi XOR. Keempat tahapan tersebut masing-masing memiliki istilah teknis tersendiri dalam struktur algoritma AES dan membentuk rangkaian proses enkripsi yang sistematis.



Gambar 4. Proses C*ipher* dan *Inverse Cipher* AES

AES adalah jenis algoritma kriptografi simetris yang menggunakan pendekatan *block cipher*, sehingga proses enkripsi dan dekripsi sama-sama mengandalkan kunci rahasia tunggal. Varian AES-128 mengacu pada implementasi algoritma Rijndael yang memproses blok data dan kunci sepanjang 128 bit. Meskipun setiap blok data berukuran 128 bit, panjang kunci yang digunakan dapat bervariasi sesuai tingkat keamanan yang diinginkan, misalnya AES-128, AES-192, maupun AES-256.

Enkripsi AES (*Cipher*) Proses enkripsi menggunakan algoritma AES dimulai dengan menerima data asli atau *plaintext* beserta kunci rahasia yang akan dipakai. Kunci utama kemudian diperluas menjadi serangkaian kunci ronde yang berbeda untuk setiap tahap enkripsi. Tahap awal melibatkan penggabungan *plaintext* dengan kunci ronde pertama menggunakan operasi XOR, yang bertujuan menyamarkan data dari awal. Selanjutnya, data melewati beberapa ronde utama, yang masing-masing terdiri dari empat langkah transformasi: penggantian *byte* dengan *S-box* untuk memperkenalkan non-linearitas, pergeseran baris matriks untuk mendistribusikan informasi secara horizontal, penggabungan kolom melalui operasi matematis untuk menyebarkan data lebih merata, dan penambahan kunci ronde melalui XOR. Pada ronde terakhir, tahap penggabungan kolom dihilangkan, sehingga keluaran akhir berupa *ciphertext* yang siap untuk disimpan atau dikirim dengan aman.

Dekripsi AES (*Inverse Cipher*) proses dekripsi membalikkan urutan transformasi enkripsi untuk mendapatkan kembali *plaintext* asli. Pertama, *ciphertext* diterima bersama kunci rahasia, kemudian kunci utama diperluas menjadi kunci ronde yang sama seperti pada enkripsi. Tahap awal dekripsi dimulai dengan menambahkan kunci ronde terakhir ke *ciphertext* menggunakan XOR. Selanjutnya, data melewati beberapa ronde utama dengan urutan transformasi terbalik: baris matriks dikembalikan ke posisi semula melalui *InvShiftRows*, byte diubah menggunakan

inverse S-box, ditambahkan kunci ronde dengan XOR, dan efek penggabungan kolom dibalik melalui *InvMixColumns*. Pada ronde terakhir, tahap penggabungan kolom dibuang. Setelah semua ronde selesai, hasil akhirnya adalah *plaintext* asli yang sama dengan data sebelum dienkripsi.

Bentuk Penyajian Data

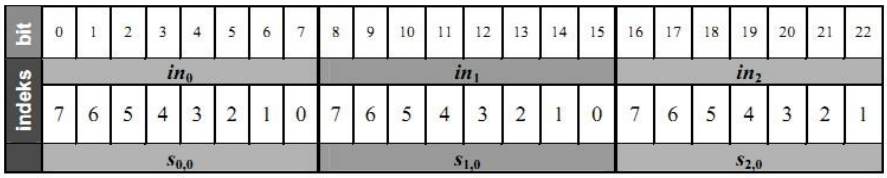
Susunan data dalam algoritma AES dibentuk dari blok *input* berukuran 128 bit, yang menjadi unit dasar pemrosesan. Setiap bit dalam blok ini diberikan penomoran mulai dari 0 hingga 127 ($0 \leq i \leq 127$), sehingga posisi masing-masing bit dapat dikenali secara spesifik. Dalam struktur ini, setiap kelompok 8 bit, atau yang dikenal sebagai satu *byte*, diperlakukan sebagai satu kesatuan logis. *Byte-byte* ini tidak hanya sekadar unit penyimpanan, tetapi juga dipandang sebagai elemen dalam *finite field*, yang memungkinkan operasi matematis seperti penjumlahan dan perkalian dilakukan secara terstruktur. Representasi ini sering ditulis dalam bentuk polinomial, yang memudahkan perhitungan algebraik pada setiap operasi transformasi AES, seperti pada tahap *SubBytes*, *MixColumns*, dan *AddRoundKey*. Dengan cara ini, pengolahan data menjadi lebih sistematis dan konsisten, serta mendukung keamanan kriptografi melalui operasi matematis yang kompleks dan sulit diterka tanpa mengetahui kunci rahasia.

$$b(x) = b_7x^7 + b_6x^6 + b_5x^5 + b_4x^4 + b_3x^3 + b_2x^2 + b_1x + b_0$$

Sebagai contoh untuk memahami representasi data dalam AES, sebuah *byte* dengan nilai biner {01100011} dapat dikonversi ke dalam bentuk persamaan polinomial, yaitu x^6+x^5+x+1. Konversi ini menunjukkan bahwa setiap bit dalam *byte* memiliki kontribusi tertentu terhadap eksponen dalam polinomial, di mana bit bernilai 1 menunjukkan hadirnya suku polinomial pada posisi eksponennya. Selain itu, *byte* tersebut juga dapat direpresentasikan dalam bentuk heksadesimal, yang dalam contoh ini menjadi {63}.

Penting untuk dicatat bahwa penomoran bit dalam sebuah *byte* mengikuti urutan tertentu dari bit paling signifikan (MSB) hingga bit paling tidak signifikan (LSB), dan susunan *byte* dalam satu blok 128-bit juga memiliki urutan yang konsisten. Hal ini memungkinkan setiap byte dan bit dapat diidentifikasi secara tepat ketika data diproses melalui operasi kriptografi, seperti substitusi, pergeseran baris, dan penggabungan kolom pada algoritma AES. Penomoran yang terstruktur ini sangat penting untuk memastikan bahwa transformasi matematis dapat diterapkan dengan benar dan konsisten pada setiap elemen blok.

Tabel 1 memperlihatkan susunan bit dalam satu *byte* serta urutan *byte* dalam blok 128-bit, sehingga memberikan panduan visual yang jelas mengenai bagaimana data diorganisasikan sebelum dan selama proses enkripsi. Representasi polinomial dan sistem penomoran ini merupakan fondasi penting bagi operasi di *finite field* (GF (2^8)), yang digunakan untuk melakukan perhitungan algebraic yang mendasari keamanan algoritma AES. Dengan demikian, pemahaman tentang struktur *byte*, urutan bit, dan representasi polinomial menjadi krusial untuk analisis dan implementasi kriptografi modern secara tepat.



Gambar 5. Pemberian Indeks pada Aliran Data Masuk

Proses enkripsi AES dilakukan dengan menggunakan sebuah array dua dimensi yang dikenal sebagai *state*, yang tersusun dari *byte-byte* data. Data dari blok *input* ditempatkan ke dalam *state* dengan format empat baris dan Nb kolom, di mana Nb merupakan jumlah kolom yang setara dengan panjang blok dibagi 32 bit, sehingga untuk AES-128 Nb bernilai 4. Setiap *byte* dalam *state* memiliki dua indeks sebagai penanda posisi, yaitu s[r,c], di mana rrr menunjukkan nomor baris ($0 \leq r < 4$) dan ccc menunjukkan nomor kolom ($0 \leq c < Nb$). Data yang tersimpan dalam *state* merepresentasikan hasil sementara dari blok input pada setiap tahap transformasi enkripsi.

Pada awal algoritma, baik pada *cipher* maupun *inverse cipher*, seluruh data input dimasukkan ke dalam *state array*. Selanjutnya, *state* diperbarui secara bertahap setelah setiap transformasi diterapkan, seperti ditunjukkan pada Gambar 1. Setelah transformasi terakhir selesai, isi state dikembalikan ke *output* dengan penomoran yang sama seperti yang tercantum pada Gambar 5.

Selain dipandang sebagai *array* dua dimensi, *state* juga dapat direpresentasikan sebagai kumpulan *word*, masing-masing terdiri dari empat *byte*. Indeks baris r pada s[r,c] menunjukkan posisi *byte* dalam setiap word, sedangkan indeks kolom ccc digunakan untuk mengidentifikasi *word* itu sendiri. Dengan demikian, *state* secara konseptual setara dengan *array* berisi empat *word*, yang diindeks berdasarkan kolom, sehingga struktur dan pengurutan *byte* dapat dimengerti secara konsisten selama proses transformasi AES.

$w0 = s0,0\,s1,0\,s2,0\,s3,0$
$w1 = s0,1\,s1,1\,s2,1\,s3,1$
$w2 = s0,2\,s1,2\,s2,2\,s3,2$
$w3 = s0,3\,s1,3\,s2,3\,s3,3$

Operasi Dasar

Meskipun setiap tahap transformasi dalam proses enkripsi AES memproses *state* sebagai satu blok data, unit operasi terkecil dalam algoritma ini tetaplah *byte*. Setiap *byte*, yang merupakan elemen dalam *finite field*, dapat menjalani operasi matematis seperti penjumlahan maupun perkalian sesuai kebutuhan langkah-langkah enkripsi.

1. Penjumlahan

   Operasi penjumlahan antara dua elemen pada *finite field* dilakukan dengan menerapkan XOR pada masing-masing bit dari elemen tersebut. Karena sifat algebraik dari *finite field*, operasi pengurangan juga setara dengan XOR, sehingga kedua operasi ini dapat diimplementasikan dengan cara yang sama. Dalam praktiknya, setiap *byte* dapat direpresentasikan dalam berbagai bentuk, seperti polinomial, biner, atau heksadesimal, yang semuanya memberikan cara alternatif untuk mengekspresikan nilai *byte* tersebut dalam konteks perhitungan kriptografi AES. Operasi penjumlahan dalam finite field dapat direpresentasikan sebagai berikut:

$(x^6 + x^4 + x^2 + x + 1) + (x^7 + x + 1) = x^7 + x^6 + x^4 + x^2$

Dalam bentuk biner, ini setara dengan:

{01010111} XOR {10000011} = {11010100}

Sedangkan dalam heksadesimal, hasilnya adalah:

57⊕83=D4

Artinya, penjumlahan di GF(2^8) dapat diimplementasikan dengan operasi XOR antar byte.

{57} XOR {83} = {d4}

2. Perkalian

   Perkalian antar elemen dalam Galois Field (2^8) (dilambangkan dengan •) pada bentuk polinomial dilakukan dengan operasi perkalian yang kemudian dimodulokan dengan operasi dilakukan polinomial modulus $m(x) = x^8 + x^4 + x^3 + x + 1$ yang menjadi dasar perhitungan di *finite field* GF(2^8).

3. $GF(2^8).xb(x) = b_8x^8 + b_7x^7 + b_6x^6 + b_5x^5 + b_4x^4 + b_3x^3 + b_2x^2 + b_1x^1 + b_0x$

   Perkalian antara x dan b(x) dapat diimplementasikan dengan melakukan pergeseran ke kiri (*left shift*) pada b(x), diikuti dengan operasi XOR kondisional dengan {1b}. Jika b_8 bernilai 1, maka dilakukan XOR, sedangkan jika $b_8 = 0$, XOR tidak dilakukan. XOR kondisional ini pada dasarnya merupakan proses modulo terhadap m(x). Rangkaian operasi left shift dan XOR tersebut digunakan untuk melakukan perkalian antar elemen pada *finite field*.

Derivasi kunci dari kunci utama

Derivasi kunci dari kunci utama adalah prosedur yang digunakan untuk menghasilkan kunci ronde adalah kunci khusus yang diterapkan pada pada setiap tahap atau putaran enkripsi, *cipher key* yang terdiri dari empat *word* (K) akan dikembangkan menjadi total 44 *word* hasil ekspansi, yang diberi notasi w_i dengan indeks $0 \leq i < 44$. Empat *word* pertama dari hasil ekspansi ini tetap merupakan kunci asli *(cipher key)*. Setiap *word* berikutnya, w_i , diperoleh dengan melakukan operasi XOR antara *word* sebelumnya w_{i-1} dan *word* yang berjarak N_k posisi lebih awal $w_i - N_k$. Proses ini memastikan bahwa setiap *word* turunan memiliki hubungan matematis dengan kunci utama, sehingga menghasilkan serangkaian kunci ronde yang unik untuk masing-masing putaran enkripsi. Untuk AES-128, nilai N_k adalah 4. Jika indeks i merupakan kelipatan N_k, maka sebelum dilakukan operasi XOR, w_i-N_k akan mengalami beberapa transformasi, diikuti dengan operasi XOR menggunakan konstanta *round*, Rcon.

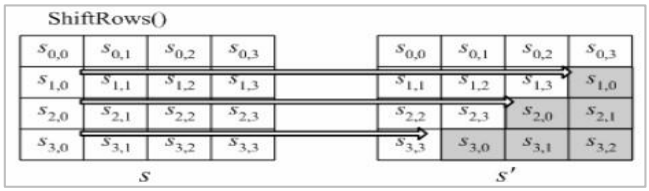
Transformasi pertama adalah SubWord(), yaitu pemetaan word w[i-4] ke nilai S-Box-nya. Hasil dari SubWord() kemudian diputar secara siklik melalui fungsi RotWord(), sehingga word [a0, a1, a2, a3] akan berubah menjadi [a1, a2, a3, a0]. Sementara itu, konstanta Rcon yang digunakan merupakan word dengan format [{02}i-1, {00}, {00}, {00}].

Proses penyandian (Enkripsi)

Proses enkripsi AES, sebagaimana ditunjukkan pada Gambar 2, melibatkan empat fungsi dasar (primitif), yaitu SubBytes(), ShiftRows(), MixColumns(), dan AddRoundKey(). Keempat fungsi ini dijalankan secara berurutan dan diulang sepanjang setiap putaran enkripsi, sehingga setiap blok data mengalami transformasi berlapis yang meningkatkan keamanan dan kompleksitas kriptografi. Sebanyak Nr - 1 kali sebagai loop utama (untuk AES-128, Nr = 10). Setiap pengulangan dari rangkaian

fungsi dasar ini disebut sebagai satu round. Sebelum memasuki loop utama, fungsi AddRoundKey() dijalankan terlebih dahulu sebagai round inisialisasi. Setelah sembilan ronde utama selesai, keempat fungsi SubBytes(), ShiftRows(), MixColumns(), dan AddRoundKey() dieksekusi secara berurutan sebagai bagian dari *final round*, menandai tahap terakhir dari proses enkripsi AES.

1. AddRoundKey()
   Pada tahap ini, *state* digabungkan dengan *round key* yang telah dihasilkan melalui proses perluasan kunci menggunakan operasi penjumlahan. Mekanisme penjumlahan ini dapat dijelaskan melalui persamaan berikut :
   [s'0,c, s'1,c, s'2,c, s'3,c] = [s0,c, s1,c, s2,c, s3,c] ⊕ [wround*4+c] dengan $0 \leq c < 4$ (penjumlahan per blok).
2. SubBytes()
   Tabel substitusi *byte* yang diterapkan pada tahap nonlinier dikenal sebagai S-Box. Transformasi ini dilakukan dengan cara mengambil invers multiplikatif dari setiap byte dalam GF(2^8), kemudian menerapkan transformasi *affine* untuk menghasilkan nilai *byte* yang baru.
3. ShiftRows()
   ShiftRows() merupakan operasi permutasi pada *state* di mana tiga baris terakhir digeser secara siklik, sedangkan baris pertama (r=0) tetap berada di posisi awal. Dalam proses ini, baris kedua digeser satu posisi ke kanan, baris ketiga digeser dua posisi, dan baris keempat digeser tiga posisi secara siklik, sebagaimana ditunjukkan pada Gambar 5.



Gambar 5. Operasi pergeseran baris

4. MixColumns()
   Proses difusi dicapai melalui transformasi MixColumns(), di mana setiap kolom pada state diproses satu per satu. Berikut adalah transformasi yang diterapkan pada setiap kolom state dalam langkah MixColumns():
   $s'_{0,c} = (\{02\} \bullet s_{0,c}) \oplus (\{03\} \bullet s_{1,c}) \oplus s_{2,c} \oplus s_{3,c}$
   $s'_{1,c} = s_{0,c} \oplus (\{02\} \bullet s_{1,c}) \oplus (\{03\} \bullet s_{2,c}) \oplus s_{3,c}$
   $s'_{2,c} = s_{0,c} \oplus s_{1,c} \oplus (\{02\} \bullet s_{2,c}) \oplus (\{03\} \bullet s_{3,c})$
   $s'_{3,c} = (\{03\} \bullet s_{0,c}) \oplus s_{1,c} \oplus s_{2,c} \oplus (\{02\} \bullet s_{3,c})$
   Setiap *byte* baru dalam satu kolom *state* dihasilkan dari kombinasi linier keempat *byte* pada kolom tersebut, dengan perkalian tetap {02} atau {03} di GF(2^8) dan digabungkan menggunakan operasi XOR. Transformasi ini meningkatkan difusi, sehingga perubahan satu byte input memengaruhi seluruh kolom *output*.

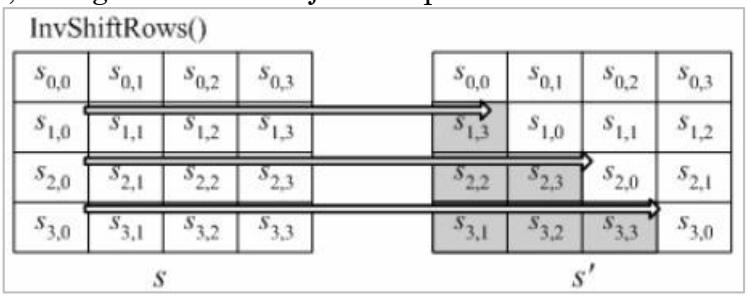
Proses Pembacaan Data Terenkripsi

Setiap fungsi yang digunakan dalam proses enkripsi AES memiliki operasi kebalikan masing-masing, sehingga dekripsi dilakukan dengan memanfaatkan *inverse cipher*, sebagaimana diperlihatkan pada Gambar 4. Proses dekripsi dimulai dengan penerapan fungsi AddRoundKey() sebagai *initial round*, kemudian dilanjutkan dengan sembilan ronde utama yang setiap putarannya mencakup InvShiftRows(),

InvSubBytes(), InvMixColumns(), dan AddRoundKey() secara berurutan. Pada ronde terakhir (*round* ke-10), tahap InvMixColumns() tidak diterapkan, sama seperti pada *final round* dalam proses enkripsi, untuk memastikan keluaran akhir sesuai dengan data asli sebelum enkripsi.

1. InvShiftRows()

   Inverse ShiftRows() merupakan operasi kebalikan dari ShiftRows() yang dilakukan dengan menggeser baris secara siklik ke arah berlawanan. Dalam proses ini, baris kedua digeser satu posisi ke kiri, baris ketiga dua posisi, dan baris keempat tiga posisi ke kiri, sebagaimana ditunjukkan pada Gambar 6.



Gambar 6. Operasi Pergeseran Baris Terbalik

2. InvSubBytes()

   Tabel substitusi *byte* versi pertama berfungsi sebagai kebalikan dari S-Box yang digunakan dalam langkah SubBytes.

3. InvMixColumns()

   Kebalikan dari operasi MixColumns() pada setiap kolom *state* dinyatakan melalui persamaan berikut:

   $s'_{0,c} = (\{0e\} \bullet s_{0,c}) \oplus (\{0b\} \bullet s_{1,c}) \oplus (\{0d\} \bullet s_{2,c}) \oplus (\{09\} \bullet s_{3,c})$
   $s'_{1,c} = (\{09\} \bullet s_{0,c}) \oplus (\{0e\} \bullet s_{1,c}) \oplus (\{0b\} \bullet s_{2,c}) \oplus (\{0d\} \bullet s_{3,c})$
   $s'_{2,c} = (\{0d\} \bullet s_{0,c}) \oplus (\{09\} \bullet s_{1,c}) \oplus (\{0e\} \bullet s_{2,c}) \oplus (\{0b\} \bullet s_{3,c})$
   $s'_{3,c} = (\{0b\} \bullet s_{0,c}) \oplus (\{0d\} \bullet s_{1,c}) \oplus (\{09\} \bullet s_{2,c}) \oplus (\{0e\} \bullet s_{3,c})$

   Persamaan ini digunakan untuk membalikkan efek transformasi MixColumns() selama proses dekripsi.

KESIMPULAN DAN SARAN

Salah satu layanan internet yang paling banyak digunakan pada perangkat *mobile* berbasis Android adalah *e-mail*. Dengan perangkat *mobile* yang terhubung ke internet, layanan *e-mail* dapat dimanfaatkan oleh berbagai kalangan untuk bertukar informasi dan melakukan kolaborasi. Tanpa disadari, penggunaan *e-mail* tidak hanya terbatas pada pertukaran informasi biasa, tetapi juga melibatkan data yang bersifat *sensitive* yakni informasi yang memiliki tingkat privasi informasi tinggi dan dapat menimbulkan kerugian jika diperoleh oleh pihak yang tidak berwenang.

Dengan demikian, semakin meluasnya penggunaan *e-mail* di perangkat *mobile* menuntut perhatian lebih terhadap aspek keamanan informasi yang dikirimkan. Perlindungan terhadap data sensitif menjadi sangat penting agar potensi risiko penyalahgunaan informasi dapat diminimalisir. Upaya pengamanan seperti penerapan teknologi kriptografi dan kebijakan perlindungan data perlu diterapkan untuk memastikan bahwa pertukaran informasi melalui *e-mail* tetap terjaga kerahasiaannya serta sulit diakses oleh pihak yang tidak memiliki izin.

DAFTAR PUSTAKA

[1] Andrushia, A. D., Neebha, T. M., Patricia, A. T., Sagayam, K. M., & Pramanik, S. (2024). Capsule network-based disease classification for *Vitis vinifera* leaves. *Neural Computing and Applications, 36*(2), 757–772. https://doi.org/10.1007/s00521-023-09058-y

[2] Arif, Z., & Nurokhman, A. (2023). Analisis perbandingan algoritma kriptografi simetris dan asimetris dalam meningkatkan keamanan sistem informasi. *Jurnal Teknologi Sistem Informasi, 4*(2), 394–405. https://doi.org/10.35957/jtsi.v4i2.6077

[3] Beyramysoltan, S., Chambers, M. I., Osborne, A. M., et al. (2022). Introducing "DoPP": A graphical user-friendly application for the rapid species identification of psychoactive plant materials and quantification of psychoactive small molecules. *Analytical Chemistry*. https://doi.org/10.1021/acs.analchem.2c01614

[4] Buhalis, D., & Law, R. (2008). Progress in information technology and tourism management: 20 years on and 10 years after the internet—The state of eTourism research. *Tourism Management, 29*(4), 609–623. https://doi.org/10.1016/j.tourman.2008.01.005

[5] Cao, J., Han, P., Zhang, W., Zhao, G., & Chen, K. (2024). Deep neural network-based plant protection strategy in rural garden landscape construction. *Crop Protection, 182*, 106716. https://doi.org/10.1016/j.cropro.2024.106716

[6] Carrera-Rivera, A., Larrinaga, F., & Lasa, G. (2022). Context-awareness for the design of smart-product service systems: Literature review. *Computers in Industry, 142*, 103730. https://doi.org/10.1016/j.compind.2022.103730

[7] Diqi, M., Sahal, A., & Aini, F. N. (2023). Multi-step vector output prediction of time series using EMA LSTM. *Jurnal Online Informatika, 8*(1), 107–114. https://doi.org/10.15575/join.v8i1.1037

[8] Eberhart, R. C., & Shi, Y. (2001). Particle swarm optimization: Developments, applications and resources. In *Proceedings of the 2001 Congress on Evolutionary Computation* (Vol. 1, pp. 81–86). IEEE. https://doi.org/10.1109/CEC.2001.934374

[9] Guo, Y., Wang, N., Wei, X., Zhou, M., Wang, H., & Bai, Y. (2024). Desert oasis vegetation information extraction by PLANET and unmanned aerial vehicle image fusion. *Ecological Indicators, 166*, 112516. https://doi.org/10.1016/j.ecolind.2024.112516

[10] Handayani, F., Nurhayati, N., & Kamila, A. (2022). Artificial intelligence as an educational media to improve adolescent reproductive health: Research and development studies. *Jurnal Keperawatan Padjadjaran, 10*(3), 170–176. https://doi.org/10.24198/jkp.v10i3.2104

[11] Kaylor, S. D., Snell Taylor, S. J., & Herrick, J. D. (2023). Estimates of biomass reductions of ozone sensitive herbaceous plants in California. *Science of the Total Environment, 878*, 163134. https://doi.org/10.1016/j.scitotenv.2023.163134

[12] Lhamo, S., Thinley, U., & Dorji, U. (2023). Spatio-temporal projection of invasion using machine learning algorithm MaxEnt. *Journal of Forest and Environmental Science, 39*(2), 105–117. https://doi.org/10.7747/JFES.2023.39.2.105

[13] Martel, C., Cifuentes, L., Cuesta, F., Stevenson, P. C., et al. (2024). Phoretic interaction between *Antherophagus* (Coleoptera) and *Bombus funebris* (Hymenoptera) using *Chuquiraga jussieui* (Asteraceae) as transfer stations. *Apidologie*. https://doi.org/10.1007/s13592-024-01075-7

[14] Ordonez, C. (2006). Association rule discovery with the train and test approach for heart disease prediction. *IEEE Transactions on Information Technology in Biomedicine, 10*(2), 334–343. https://doi.org/10.1109/TITB.2005.856864
[15] Paul, A., Ambuj, Nagar, H., & MacHavaram, R. (2023). Utilizing fine-tuned YOLOv8 deep learning model for greenhouse capsicum detection and growth stage determination. In *Proceedings of the 7th International Conference on I-SMAC (IoT in Social, Mobile, Analytics and Cloud)* (pp. 649–656). IEEE. https://doi.org/10.1109/I-SMAC58438.2023.10290335
[16] Pradana, Z. H., Nafi'ah, H., & Rochmanto, R. A. (2022). Chatbot-based information service using RASA open-source framework in Prambanan Temple tourism object. *Jurnal RESTI, 6*(4), 656–662. https://doi.org/10.29207/resti.v6i4.3913
[17] Remch, Z., Khoulji, S., & Kerkeb, M. L. (2024). Object detection techniques for strawberry disease detection: A comprehensive review. *Journal of Theoretical and Applied Information Technology, 102*(19), 7178–7201.
[18] Ribeiro, J. B., da Silva, R. R., Dias, J. D., Escarpinati, M. C., & Backes, A. R. (2024). Automated detection of sugarcane crop lines from UAV images using deep learning. *Information Processing in Agriculture, 11*(3), 385–396. https://doi.org/10.1016/j.inpa.2023.04.001
[19] Ryan, D. (2020). *Understanding digital marketing: Marketing strategies for engaging the digital generation* (5th ed.). Kogan Page.
[20] Sachar, S., & Kumar, A. (2021). Automatic plant identification using transfer learning. *IOP Conference Series: Materials Science and Engineering, 1022*(1), 012086. https://doi.org/10.1088/1757-899X/1022/1/012086
[21] Sasmal, B., Goswami, R. S., & Dutta, M. P. (2024). Identification of rice leaf disease using Gaussian mixture model: A machine learning approach using image classification techniques. In *Deep Learning Applications in Operations Research* (pp. 174–191). CRC Press. https://doi.org/10.1201/9781032725444-15
[22] Suwarno, Tan, T., & Jonathan. (2023). MobileNetV3-based handwritten Chinese recognition towards the effectiveness of learning Hanzi. *Jurnal RESTI, 7*(6), 1394–1402. https://doi.org/10.29207/resti.v7i6.5505
[23] Tian, H., Wang, T., Liu, Y., Qiao, X., & Li, Y. (2020). Computer vision technology in agricultural automation—A review. *Information Processing in Agriculture, 7*(1), 1–19. https://doi.org/10.1016/j.inpa.2019.09.006
[24] UNWTO. (2018). *Tourism for development: Volume I – Key areas for action*. World Tourism Organization.
[25] Yeu, I. W., Han, G., Ye, K. H., Hwang, C. S., & Choi, J. H. (2021). InterPhon: Ab initio interface phonon calculations within a 3D electronic structure framework. *Computer Physics Communications, 268*, 108089. https://doi.org/10.1016/j.cpc.2021.108089
[26] Zahay, D., & Roberts, M. (2023). *Internet marketing: Integrating online and offline strategies* (5th ed.). Cengage Learning.