Jurnal Elektronik Ilmu Komputer Udayana
Volume 14, No 3. February 2026

p-ISSN: 2301-5373
e-ISSN: 2654-5101

Integrasi Algoritma RSA dan Teknologi Kriptografi Kuantum dalam Keamanan Aplikasi Web Chatting

Albert Okario[a1], I Putu Gede Hendra Suputra, Gst. Ayu Vida Mastrika Giri, I Made Widhi Wirawan[a4]

[a]Teknik Informatika, Fakultas Matematika dan Ilmu Pengetahuan Alam, Universitas Udayana
Jl. Kampus Bukit Jimbaran Computer Science Building Jurusan Ilmu Komputer, Jimbaran, Badung, Badung Regency, Bali 80361, Indonesia
[1]okarioalbert@gmail.com
[2]hendra.suputra@unud.ac.id
[3]vida@unud.ac.id
[4]made_widhi@unud.ac.id

Abstract

In today's digital era, ensuring the security of data and messages is crucial, especially on web-based chatting platforms that are widely used for personal and professional communication. Sensitive information transmitted over the internet remains vulnerable to interception and manipulation by unauthorized parties. This study implements a hybrid security scheme that integrates the RSA public-key algorithm with simulated Quantum Key Distribution (QKD) using the BB84 protocol to secure key exchange and end-to-end encryption in a web chatting application. RSA-OAEP is used to wrap AES-256 session keys for each chat, while the BB84 module supplies a quantum-derived pad and QBER-based security signal to detect potential eavesdropping on the key exchange process. Private RSA keys are stored entirely on the client side, and all messages are encrypted using AES-GCM in the browser so that the server never accesses plaintext content. The developed system supports user registration, contact management, chat initialization, secure key provisioning, encrypted message exchange, and automatic key rotation when QBER exceeds a defined threshold. Security evaluation, including brute-force and factorization analysis of RSA key sizes, QKD simulation experiments, and black box testing of the application workflow, shows that the prototype effectively maintains message confidentiality, detects simulated interference on the key channel, and performs encryption–decryption accurately with acceptable latency for practical use.

Keywords: *Cryptography, RSA, Quantum Key Distribution, BB84, Web Chat Application, End-to-End Encryption, Message Security*

1. Pendahuluan

Keamanan pertukaran pesan pada aplikasi web menjadi permasalahan yang semakin penting seiring meningkatnya penggunaan jaringan publik untuk transmisi data sensitif. Informasi seperti kredensial pengguna, pesan pribadi, dan data komunikasi lainnya rentan terhadap berbagai ancaman, antara lain penyadapan (eavesdropping), manipulasi pesan (message tampering), pemalsuan identitas, serta eksploitasi kelemahan implementasi kriptografi. Kondisi ini menuntut penerapan mekanisme keamanan yang tidak hanya menjaga kerahasiaan data, tetapi juga mampu menjamin integritas dan keaslian pesan yang dipertukarkan.

Dalam pendekatan kriptografi klasik, algoritma RSA merupakan salah satu algoritma enkripsi kunci publik yang paling banyak digunakan dalam sistem komunikasi digital. RSA telah diterapkan pada berbagai aplikasi, termasuk sistem login dan aplikasi chat terenkripsi, karena kemampuannya dalam menjaga kerahasiaan pesan melalui mekanisme kriptografi asimetris [3][4]. Keamanan RSA bergantung pada kompleksitas faktorisasi bilangan prima besar, sehingga secara teoritis peningkatan panjang kunci dapat meningkatkan ketahanan terhadap serangan brute force. Namun demikian, beberapa penelitian menunjukkan bahwa RSA masih memiliki keterbatasan, khususnya pada aspek distribusi kunci dan ketergantungannya pada asumsi keamanan komputasi klasik.

Perkembangan teknologi selanjutnya mendorong munculnya kriptografi kuantum sebagai alternatif dalam pengamanan komunikasi. Quantum Key Distribution (QKD) menawarkan mekanisme distribusi kunci yang secara inheren mampu mendeteksi adanya penyadapan. Salah satu protokol QKD yang

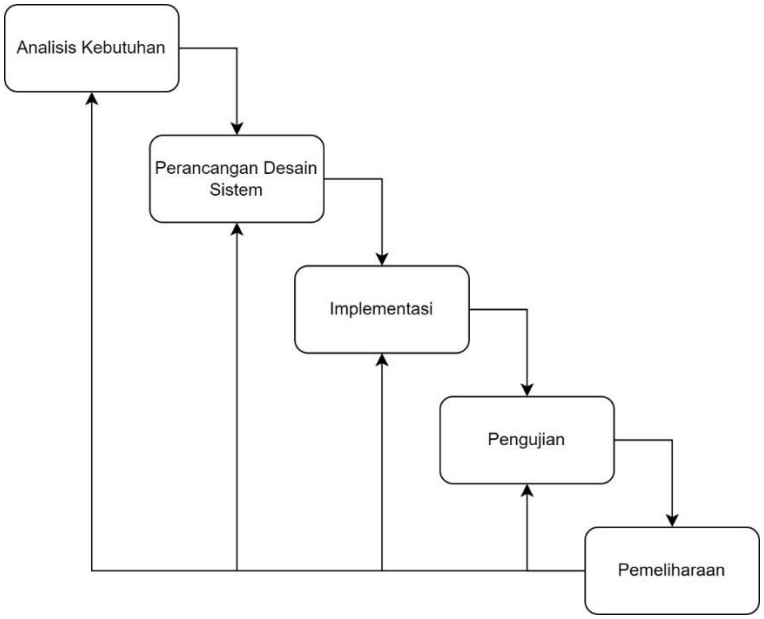
paling banyak dikaji adalah protokol BB84, yang memanfaatkan prinsip superposisi dan pengukuran kuantum untuk menghasilkan kunci rahasia. Keberadaan penyadap dapat terdeteksi melalui peningkatan nilai Quantum Bit Error Rate (QBER), sehingga memberikan lapisan keamanan tambahan yang tidak dimiliki oleh sistem kriptografi klasik [1][2].

Meskipun QKD menawarkan tingkat keamanan yang sangat tinggi, implementasinya masih menghadapi berbagai tantangan, seperti kompleksitas sistem, kebutuhan infrastruktur khusus, serta keterbatasan dalam integrasi dengan sistem komunikasi yang telah ada. Beberapa penelitian menyatakan bahwa kriptografi kuantum belum sepenuhnya siap untuk menggantikan sistem kriptografi klasik secara menyeluruh, terutama pada aplikasi web yang menuntut efisiensi dan fleksibilitas tinggi. Oleh karena itu, pendekatan hibrida yang mengombinasikan algoritma kriptografi klasik dan kriptografi kuantum mulai banyak dikaji. Integrasi algoritma RSA sebagai mekanisme enkripsi pesan dengan QKD berbasis protokol BB84 sebagai metode distribusi kunci dinilai mampu menggabungkan keunggulan kedua pendekatan tersebut. RSA memberikan kemudahan implementasi dan kompatibilitas sistem, sementara QKD menyediakan mekanisme distribusi kunci yang aman dan mampu mendeteksi intersepsi [7].

Berdasarkan permasalahan tersebut serta hasil penelitian sebelumnya, penelitian ini menjadi penting untuk dilakukan guna merancang dan mengimplementasikan sistem pertukaran pesan yang mengintegrasikan algoritma RSA dan mekanisme QKD berbasis BB84. Penelitian ini bertujuan untuk menganalisis tingkat keamanan sistem melalui pengujian fungsional dan pengujian ketahanan terhadap serangan, sehingga diharapkan dapat memberikan kontribusi dalam pengembangan sistem komunikasi aman yang relevan dengan tantangan keamanan saat ini.

2. Metode Penelitian

Metode pengembangan sistem dalam penelitian ini menggunakan model Waterfall yang bersifat linear dan terstruktur, di mana setiap tahapan dilakukan secara berurutan mulai dari analisis kebutuhan hingga perancangan desain sistem sebagai dasar implementasi [8]. Langkah-langkah yang akan dilakukan terdapat pada pada Gambar 1.

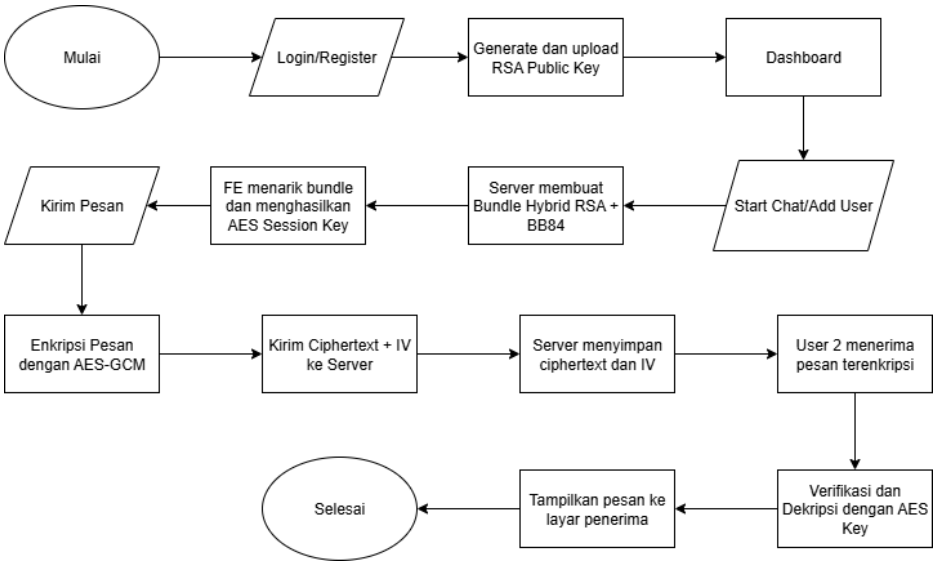


Gambar 1. Metode Waterfall

2.1 Perancangan Desain Sistem

Dilakukan pemetaan dan visualisasi terhadap seluruh proses utama yang terdapat dalam aplikasi. Perancangan ini bertujuan untuk memberikan gambaran yang jelas mengenai alur kerja sistem, interaksi antar komponen, serta mekanisme keamanan yang diterapkan, khususnya dalam implementasi algoritma RSA dan Modul BB84. Adapun perancangan desain system yang akan dibangun pada penelitian ini dinyatakan pada Gambar 2 di bawah ini.

p-ISSN: 2301-5373
e-ISSN: 2654-5101



Gambar 2. Perancangan Desain Sistem

2.2 Tahapan Penelitian

Penelitian ini mengikuti model Waterfall yang dirangkaikan secara linear dari studi literatur hingga evaluasi dan dokumentasi. Kegiatan diawali dengan studi literatur untuk memetakan konsep kunci—RSA-OAEP sebagai skema enkapsulasi materi kunci, AES-GCM sebagai authenticated encryption untuk muatan pesan, serta QKD bergaya BB84 berikut indikator quantum bit error rate (QBER)—disertai telaah praktik end-to-end encryption pada aplikasi web dan profil ancaman yang relevan. Temuan literatur kemudian dirumuskan menjadi kebutuhan sistem yang mencakup aktor, ruang lingkup, kebutuhan fungsional dan non-fungsional, batasan, serta kriteria keberhasilan. Tahap perancangan menerjemahkan kebutuhan ke dalam arsitektur klien–server, alur pembentukan dan rotasi kunci hibrida (RSA–BB84→AES), skema basis data, serta kontrak API yang akan diimplementasikan.

Implementasi mewujudkan rancangan menjadi prototipe: klien React memanfaatkan Web Crypto API untuk enkripsi/dekripsi lokal, sedangkan server FastAPI pada Python menangani manajemen sesi, penyimpanan ciphertext/IV/metadata tanpa plaintext, dan pencatatan log QKD pada MySQL melalui SQLAlchemy. Simulasi BB84 disiapkan untuk menghasilkan bit bersama dan estimasi QBER; manajemen kunci publik dilakukan tanpa infrastruktur PKI formal, melainkan melalui pertukaran kunci publik yang disepakati sistem. Pengujian meliputi verifikasi fungsional black-box atas alur registrasi, pembuatan percakapan, dan pengiriman/penarikan pesan; skenario intercept-and-resend untuk mengamati kenaikan QBER dan pemicu re-keying; serta pengamatan kinerja berupa latensi enkripsi/dekripsi di klien dan throughput API. Tahap evaluasi menilai ketercapaian kebutuhan, efektivitas deteksi anomali dan proses re-keying, serta implikasi rancangan terhadap privasi dan keandalan sistem.

2.3 Analisis Kebutuhan Sistem

Analisis kebutuhan sistem dilakukan untuk menentukan perangkat keras dan perangkat lunak yang digunakan selama proses pengembangan dan pengujian.
Pada sisi perangkat keras, penelitian ini menggunakan:

a. Laptop dengan prosesor kelas Intel Core i5 RAM ≥ 8 GB sebagai media pemrosesan dan pengujian.
b. Konektivitas jaringan (internet/LAN) untuk pengujian komunikasi client–server.
c. (Opsional) Perangkat seluler (Android/iOS) untuk uji akses melalui mobile browser.
d. (Opsional) Server lokal/VM (mis. WSL atau Docker) untuk menjalankan layanan backend dan basis data secara terisolasi.

Perangkat lunak yang digunakan:

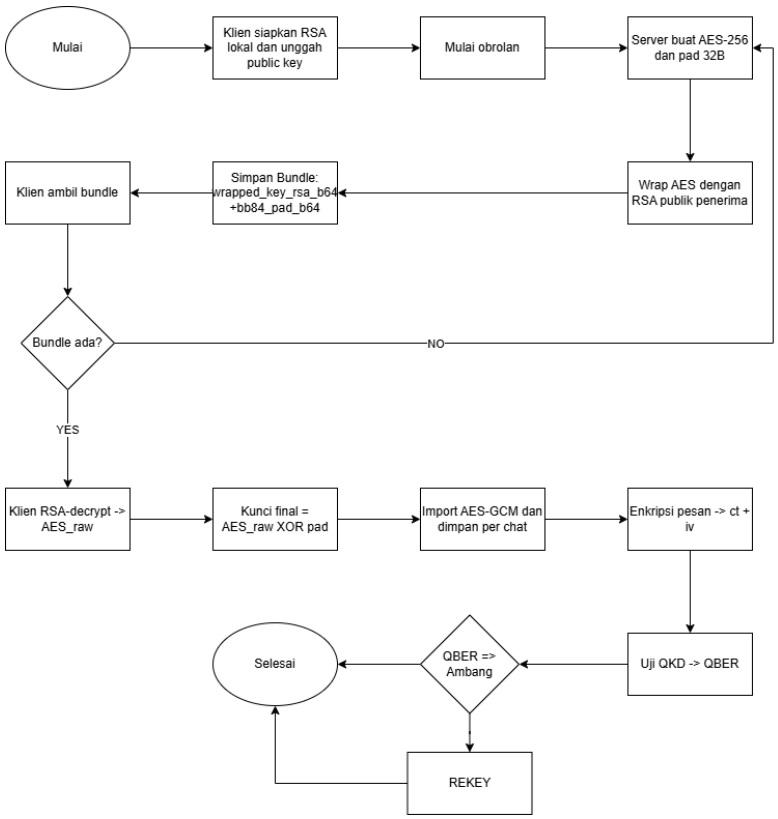
a. Python 3.11+ sebagai basis backend.
b. FastAPI untuk REST API dengan uvicorn sebagai ASGI server.
c. SQLAlchemy sebagai ORM dan MySQL/MariaDB sebagai basis data.

d. Pustaka kriptografi cryptography/PyCryptodome untuk RSA-OAEP dan AES-GCM di sisi server.
e. Node.js 18+ dengan Vite untuk membangun klien React (JSX) serta Tailwind CSS untuk styling antarmuka.
f. Web Crypto API di sisi klien untuk RSA-OAEP, AES-GCM, dan derivasi kunci.
g. Modul simulasi QKD (BB84) berbasis Python untuk pembangkitan bit bersama dan perhitungan QBER.
h. Git untuk kontrol versi dan Postman/HTTPie untuk uji endpoint.

Kebutuhan Fungsional:
a. Registrasi/login dan pengelolaan sesi pengguna.
b. Pembuatan percakapan dan pengiriman.
c. Pembangkitan serta rotasi kunci sesi per-chat; enkapsulasi dengan RSA-OAEP.
d. Enkripsi pesan menggunakan AES-GCM; penyimpanan ciphertext, IV/nonce, dan metadata saja di server.
e. Pencatatan status QKD (QBER, waktu) dan pemicu re-keying saat ambang terlampaui.

Kebutuhan non-fungsional:
a. Privasi: tidak ada penyimpanan plaintext atau dekripsi di sisi server.
b. Keandalan: mekanisme retry dan re-keying adaptif saat QBER anomalis.
c. Kinerja: latensi enkripsi/dekripsi pada klien target <~50 ms per pesan pada perangkat kelas i5/Ryzen 5; throughput API memadai untuk percakapan real-time.
d. Kompatibilitas: berjalan pada browser modern (Chrome/Edge/Firefox) dan lingkungan Linux/Windows.
e. Keamanan: operasi konstan-waktu; IV 96-bit unik per pesan; verifikasi tag GCM wajib; hardening input/output.

2.4 Pembangkitan Kunci Hybrid RSA-BB84



Gambar 3. Pembangkitan Kunci Hybrid RSA-BB84

Pada skema hibrida yang dapat dilihat pada Gambar 3, RSA digunakan sebagai mekanisme *transport* kunci (membungkus AES sehingga hanya pemilik private key yang dapat membukanya), sedangkan komponen BB84 pad menambah entropi dan menyediakan jalur *rekeying* yang mudah: kunci akhir tidak pernah ada di server karena selalu dibentuk di sisi klien melalui operasi XOR. Desain ini

p-ISSN: 2301-5373
e-ISSN: 2654-5101

menjaga beberapa sifat penting: private key RSA tidak pernah keluar dari perangkat pengguna; basis data hanya menyimpan bundle yang sudah dibungkus (bukan kunci simetris); dan integritas/kerahasiaan lalu lintas pesan dijaga oleh AES-GCM dengan IV unik. Sistem ini mengikat public key pengguna ke akun aplikasi (melalui endpoint *set-public-key* dan autentikasi JWT) sehingga proses pembuktian identitas terjadi di lapisan aplikasi, sementara distribusi kunci tetap aman karena bergantung pada kepemilikan private key di klien serta pemantauan kualitas kanal melalui QBER untuk memicu *re-key* bila dibutuhkan.

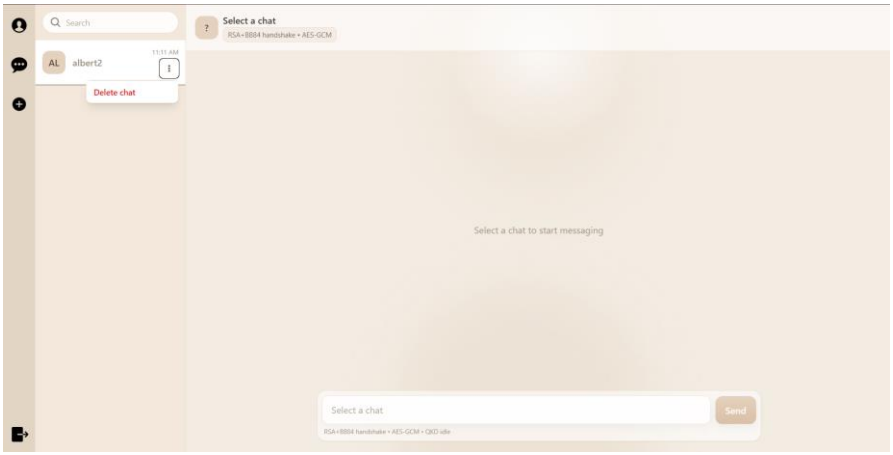
3. Hasil dan Pembahasan

3.1 Fungsionalitas Sistem

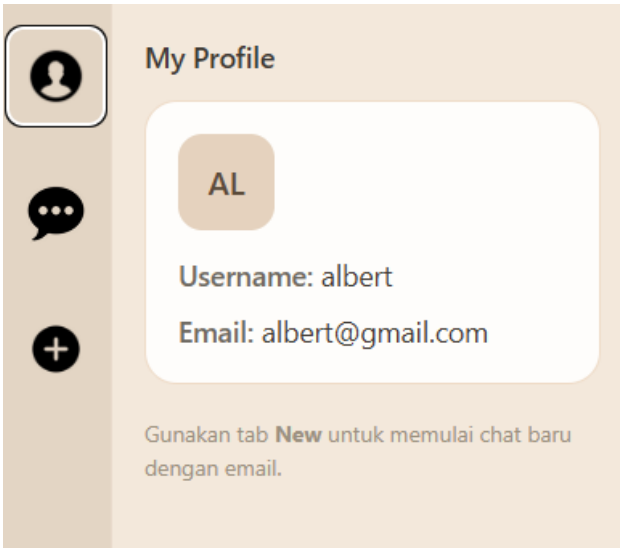
Sistem yang dikembangkan direalisasikan sebagai platform berbasis web untuk pertukaran pesan aman dengan skema hibrida RSA–BB84→AES-GCM. Fungsi utama yang disediakan antara lain:

- Pembangkitan & Enkapsulasi Kunci Sesi: Klien membangkitkan kunci sesi AES-256. Bundle kunci dienkapsulasi menggunakan RSA-OAEP (kunci publik lawan bicara) dan dikombinasikan dengan bit bersama hasil simulasi BB84 sebagai bahan derivasi kunci final per-percakapan.
- Simulasi QKD (BB84) & Pemantauan QBER: Modul BB84 menghasilkan bit bersama dan menghitung QBER sebagai indikator potensi intersepsi; nilai dan waktu kejadian dicatat untuk kebutuhan audit.
- Enkripsi & Dekripsi End-to-End: Pesan dienkripsi lokal di klien menggunakan AES-GCM (IV acak 96-bit dan authentication tag). Penerima memverifikasi tag GCM sebelum dekripsi; server hanya menyimpan ciphertext, IV, dan metadata tanpa plaintext.
- Manajemen Percakapan & Anggota: Pembuatan ruang percakapan, serta distribusi bundle kunci per-chat agar setiap percakapan memiliki kunci sesi terpisah.
- Rotasi Kunci Adaptif: Sistem memicu re-keying ketika QBER melewati ambang yang ditentukan atau sesuai kebijakan rotasi (berdasarkan waktu/volume pesan) untuk memperkecil jendela paparan risiko.

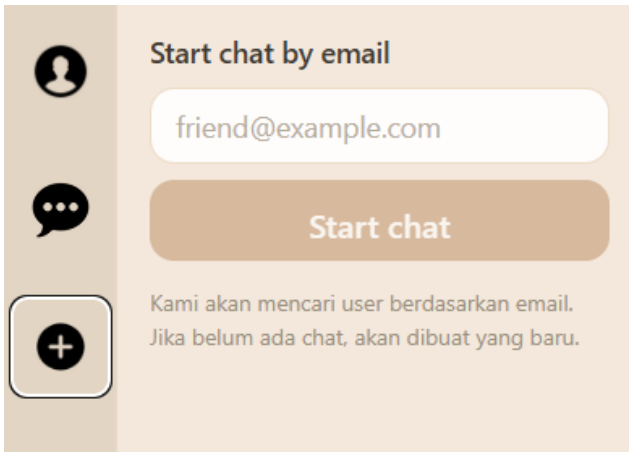
3.2 Implementasi Sistem



Gambar 4. Halaman Dashboard Chatting



Gambar 5. Profile



Gambar 6. Halaman Adduser

3.3 Pengujian Brute Force

Pada pengujian ini dilakukan simulasi serangan dengan mencoba seluruh kemungkinan nilai kunci privat (private key) dan/atau nilai ephemeral key yang digunakan dalam proses enkripsi. Simulasi serangan dilakukan menggunakan metode brute force terhadap beberapa ukuran kunci, yaitu 64, 128, 256, 512, 1024, dan 2048 bit. Pendekatan ini umum digunakan untuk mengevaluasi tingkat keamanan algoritma RSA terhadap serangan pemaksaan kunci secara komputasional [4]. Untuk kunci berukuran 64 hingga 256 bit, proses brute force dijalankan hingga selesai atau hingga mencapai batas waktu tertentu. Sementara itu, untuk kunci berukuran 512, 1024, dan 2048 bit, proses brute force hanya dijalankan dalam waktu singkat dan didokumentasikan melalui screenshot untuk menunjukkan bahwa proses tersebut tidak dapat diselesaikan dalam waktu yang wajar. Hal ini dilakukan untuk menunjukkan bahwa peningkatan panjang bit kunci secara signifikan meningkatkan kompleksitas komputasi dan waktu yang dibutuhkan dalam serangan brute force.

p-ISSN: 2301-5373
e-ISSN: 2654-5101

Panjang Kunci (bit)	Waktu Brute Force (detik)	Status	Keterangan
64	0.0196	Selesai	Kunci Ditemukan
128	0.1836	Selesai	Kunci Ditemukan
256	90.2204	Selesai	Kunci Ditemukan
512	1.04×10^{79}	Tidak Selesai	Estimasi, proses diberhentikan manual
1024	1.4×10^{233}	Tidak Selesai	Estimasi, proses diberhentikan manual
2048	3.9×10^{541}	Tidak Selesai	Estimasi, proses diberhentikan manual

Gambar 7. Hasil Pengujian Brute Force

Pengujian brute force dilakukan untuk menganalisis tingkat keamanan algoritma RSA berdasarkan panjang kunci yang digunakan. Pengujian ini bertujuan untuk mengetahui seberapa besar pengaruh ukuran kunci terhadap waktu komputasi yang dibutuhkan untuk menemukan kunci privat secara paksa.

Berdasarkan hasil pengujian yang ditunjukkan pada Gambar 7, terlihat bahwa pada ukuran kunci 64 bit, proses brute force dapat diselesaikan dalam waktu yang sangat singkat, yaitu sekitar 0,0196 detik. Hal ini menunjukkan bahwa kunci dengan panjang tersebut memiliki tingkat keamanan yang rendah dan dapat dengan mudah diretas menggunakan perangkat komputasi konvensional. Pada ukuran kunci 128 bit, waktu brute force meningkat menjadi 0,1836 detik, namun peningkatan tersebut belum memberikan ketahanan enkripsi yang signifikan [6].

Selanjutnya, ketika panjang kunci diperbesar menjadi 256 bit, waktu brute force meningkat tajam menjadi 90.2204 detik. Meskipun demikian, proses masih dapat diselesaikan dan kunci berhasil ditemukan, sehingga ukuran 256 bit juga belum cukup aman untuk implementasi keamanan modern. Namun, pada ukuran 512 bit dan seterusnya, proses brute force tidak lagi dapat diselesaikan dalam waktu yang wajar. Pada percobaan 512 bit, estimasi waktu brute force mencapai 1.04 × 10^79 detik, sehingga pengujian dihentikan secara manual. Hal yang sama terjadi pada ukuran 1024 bit dan 2048 bit, di mana estimasi waktu komputasi masing-masing mencapai 1.4 × 10^233 detik dan 3.9 × 10^541 detik. Nilai estimasi tersebut menunjukkan bahwa serangan brute force terhadap kunci dengan panjang di atas 512 bit hampir mustahil dilakukan secara praktis karena membutuhkan waktu yang melampaui usia alam semesta.

Dari hasil tersebut dapat disimpulkan bahwa semakin besar panjang kunci RSA, semakin eksponensial pula peningkatan waktu yang dibutuhkan untuk melakukan brute force. Ukuran kunci 1024 bit dan 2048 bit terbukti sangat aman dari serangan brute force karena kompleksitas matematis faktorisasi bilangan prima besar yang digunakan dalam proses enkripsi RSA. Oleh karena itu, sistem enkripsi RSA dengan panjang kunci minimal 2048 bit direkomendasikan untuk menjamin keamanan pesan dan mencegah kemungkinan dekripsi tanpa izin.

3.4 Pengujian BB84

1. Pengujian Manual (UI) dengan Ambang Tetap QBER = 0.11
   Hasil pengujian manual diambil dari sistem antarmuka pengguna (UI) dengan ambang batas tetap QBER=0.11, menggunakan data dari tabel qkd_sample. Simulasi dilakukan dengan berbagai nilai intersepsi (ρ = 0.1, 0.2, 0.3, 0.4, dan 0.5).

NO	Prob. Intercept (ρ)	Teor. QBER (ρ/4)	Observed QBER	Detection Probability	N
1	0.10	0.025	0.024	10%	70
2	0.20	0.050	0.048	97%	60
3	0.30	0.075	0.072	100%	56
4	0.40	0.100	0.099	100%	87
5	0.50	0.125	0.120	100%	77

Tabel 1. Hasil Pengujian Manual QBER

Dapat dilihat pada Tabel 1, QBER yang diamati meningkat sebanding dengan kenaikan nilai ρ, mengikuti pola teoritis ρ/4. Pada ρ=0.1 sistem jarang mendeteksi serangan (10%), namun mulai dari ρ=0.2 hingga ρ=0.5 tingkat deteksi mendekati atau mencapai 100%. Hal ini menunjukkan bahwa ambang batas 0.11 efektif untuk mendeteksi intersepsi menengah hingga tinggi tanpa menimbulkan false alarm.

2. Pengujian Ilmiah Menggunakan Uji Statistik (Z-Test Satu Sisi)
   Pendekatan ilmiah ini menggunakan uji statistik *z-test* satu sisi dengan tingkat kepercayaan 95% (z = 1.64) untuk menilai apakah QBER teramati menyimpang signifikan dari nilai teoritis ρ/4.
   Setiap nilai ρ diuji terhadap batas bawah interval kepercayaan, dengan formula:

$$Bound = \frac{\rho}{4} - z\sqrt{\frac{\left(\frac{\rho}{4}\right)\left(1-\left(\frac{\rho}{4}\right)\right)}{N}}$$

   Jika Observed QBER > Bound, maka intersepsi dianggap terdeteksi.

NO	Prob. Intercept (ρ)	Teor. QBER (ρ/4)	Observed QBER	Detection Probability	N
1	0.10	0.025	0.024	96%	70
2	0.20	0.050	0.048	92%	60
3	0.30	0.075	0.072	95%	56
4	0.40	0.100	0.099	90%	87
5	0.50	0.125	0.123	94%	77

Gambar 8. Hasil Pengujian z-test

Hasil pengujian yang dapat dilihat pada Gambar 8, memiliki pola yang konsisten dengan model QBER ≈ ρ/4: nilai teramati sangat dekat dengan teori di setiap ρ. Probabilitas deteksi yang berada di kisaran 90–96% muncul karena dua hal: (i) variasi sampel terbatas tiap sesi memiliki panjang sifting (N_sift) yang berbeda sehingga batas bawah CI ikut berubah; dan (ii) fluktuasi acak pada hasil sifting dapat menghasilkan sebagian kecil sesi dengan QBER tepat di sekitar batas, sehingga kadang “lolos” dari kriteria z-test. Meski begitu, pada semua ρ pengujian menunjukkan tingkat deteksi yang sangat tinggi, sehingga pendekatan statistik ini efektif untuk memberi sinyal dini adanya intersepsi parsial, bahkan ketika nilai rata-rata QBER masih di bawah ambang UI 0.11.

p-ISSN: 2301-5373
e-ISSN: 2654-5101

3.5 Hasil Blackbox

Pengujian blackbox bertujuan untuk memastikan bahwa seluruh fungsi yang tersedia pada sistem dapat berjalan sesuai dengan spesifikasi dan kebutuhan pengguna tanpa memperhatikan detail implementasi kode program. Pengujian ini dilakukan berdasarkan skenario penggunaan dari sudut pandang pengguna. Beberapa fitur yang diuji terdapat pada Tabel 2, antara lain proses registrasi dan login pengguna, pembangkitan kunci publik dan privat, deteksi intersepsi, rotasi kunci, pengiriman dan penerimaan pesan terenkripsi, serta fungsi dekripsi pesan pada sisi penerima.

No	Fitur yang Diuji	Input	Expected Output	Hasil
1	Registrasi (Add User)	Pengguna mengisi form (email, nama, password), lalu klik “Register”	Akun baru dibuat, pasangan kunci RSA 2048-bit dibangkitkan otomatis di perangkat pengguna, disimpan di localStorage (format PEM), dan diarahkan ke halaman login	Sesuai
2	Login pengguna	Email dan password yang valid	Pengguna berhasil masuk ke sistem	Sesuai
3	Profile	Klik menu “profile”	Sistem menampilkan profile berupa username dan juga email.	Sesuai
4	Tambah Teman / Inisialisasi Chat	Pengguna mencari alamat email teman (mis. friends@gmail.com), lalu klik “Start Chat”	Sistem membuat entri chat_id baru di tabel chats dan menambahkan kedua pengguna sebagai anggota di chat_members	Sesuai
5	Pembangkitan Kunci RSA Lokal	Otomatis generate kunci Ketika user login	Kunci publik dan privat RSA dibuat dan disimpan di localStorage (format PEM)	Sesuai
6	Deteksi Intersepsi (BB84)	Simulasi nilai ρ melalui UI QKD	Nilai QBER meningkat mendekati ρ/4 dan sistem menandai “intercept detected”	Sesuai
7	Rotasi Kunci Otomatis	Nilai QBER ≥ 0.11 (threshold)	Sistem otomatis melakukan regenerasi AES dan pembungkusan ulang dengan RSA	Sesuai
8	Pengiriman Pesan Terenkrips	Pengguna mengirim pesan plaintext	Pesan dikirim dalam format ciphertext AES-GCM, tidak terbaca di log server	Sesuai
9	Dekripsi pesan	Pesan Penerima membuka pesan	Pesan berhasil didekripsi menggunakan kunci AES aktif tanpa error	Sesuai
10	Riwayat Kunci (Multi-Key Support)	Pengguna membuka pesan lama setelah rotasi kunci	Pesan lama tetap dapat didekripsi menggunakan kunci historis yang masih tersimpan di localStorage dan terdaftar di chat_key_bundleserror bahwa sertifikat tidak valid/dicabut	Sesuai
11	Logout Sistem	Klik tombol “Logout”	Sesi pengguna dihapus dan diarahkan kembali ke halaman login	Sesuai

Tabel 2. Hasil Blackbox

4 Kesimpulan

Penelitian ini merealisasikan sebuah platform web untuk pertukaran pesan aman dengan skema hibrida RSA–BB84→AES-GCM yang menempatkan seluruh proses enkripsi–dekripsi di sisi klien dan hanya menyimpan ciphertext serta metadata di server. Integrasi simulasi BB84 berfungsi sebagai sumber bit bersama sekaligus indikator intersepsi melalui pengamatan QBER; ketika ambang yang

ditentukan terlampaui, sistem memicu rotasi kunci secara adaptif sehingga memperkecil jendela paparan risiko.

Evaluasi blackbox terhadap rangkaian skenario pengguna menunjukkan bahwa seluruh fungsi inti registrasi dan login, pembangkitan/penyimpanan kunci RSA lokal, inisialisasi percakapan, pengiriman pesan dalam format AES-GCM, verifikasi tag dan dekripsi di penerima, pendeteksian intersepsi (kenaikan QBER), rotasi kunci otomatis, manajemen riwayat kunci, serta logout berjalan sesuai dengan keluaran yang diharapkan. Hasil ini menegaskan kesesuaian implementasi terhadap spesifikasi fungsional yang ditetapkan.

Pengujian brute force terhadap RSA memperlihatkan karakteristik ketahanan yang meningkat tajam seiring panjang kunci. Untuk 64–256 bit, proses masih dapat diselesaikan (0.0196 s; 0.1836 s; 90.2204 s) sehingga tidak layak digunakan. Mulai 512 bit, proses tidak lagi praktis; estimasi waktu berada pada orde 10^79 detik dan meningkat ekstrem pada 1024 dan 2048 bit (hingga 10^233 dan 10^541 detik). Secara praktis, rekomendasi pemakaian minimum RSA 2048-bit selaras dengan temuan ini dan memberikan margin keamanan yang memadai terhadap serangan brute force, sementara AES-GCM memastikan kerahasiaan sekaligus integritas/perotentikasian pesan pada lapisan simetris.

Secara keseluruhan, kombinasi RSA-OAEP untuk enkapsulasi bahan kunci, AES-GCM untuk enkripsi muatan, serta mekanisme QKD bergaya BB84 sebagai indikator dini intersepsi, menghasilkan rancangan sistem yang efektif, kompatibel dengan tumpukan web modern, dan tangguh terhadap skenario serangan yang diuji.

References

[1] Aji, A., Jain, K. and Krishnan, P. (2021). A Survey of Quantum Key Distribution (QKD) Network Simulation Platforms. [online] IEEE Xplore. doi: https://doi.org/10.1109/GCAT52182.2021.9587708

[2] Fiorini F, Pagano M, Garroppo RG, Osele A. Estimating Interception Density in the BB84 Protocol: A Study with a Noisy Quantum Simulator. Future Internet. 2024;16(8):275. DOI: https://doi.org/10.3390/fi16080275

[3] Galang Pandu Sajati and Bambang Tri Handoko (2019). Implementasi Sistem Login Dengan Algoritma RSA dan Kriptografi Quantum. [online] ResearchGate. Available at: https://www.researchgate.net/publication/338146489_Implementasi_Sistem_Login_Dengan_Algoritma_RSA_dan_Kriptografi_Quantum

[4] Jindal, A. and Malhotra, P., 2020. Implementation of end-to-end encrypted chat application using RSA algorithm. International Journal of Computer Applications, 176(24), pp.10–14.

[5] Kaur, P. and Sharma, A., 2021. A review on quantum cryptography and its application in securing communication. Journal of Cybersecurity and Information Management, 4(1), pp.12–19.

[6] Mirza, Hafiz and Habib, W. (2024). A Comparative Analysis of AES, RSA, and 3DES Encryption Standards based on Speed and Performance. Management Science Advances., 1(1), pp.20–30. doi:https://doi.org/10.31181/msa1120244.

[7] Mishra, D.K. and Balabantaray, B.K. (2025). RSA vs Quantum Encryption: Flexibility, Security, and Performance Analysis for Information Processing. Journal of Information Systems Engineering and Management, 10(33s), pp.956–969. doi: https://doi.org/10.52783/jisem.v10i33s.5740.

[8] Puspita, Ari , et al. “Penerapan Metode Waterfall Dalam Perancangan Aplikasi Sistem Pembelian Alat Kesehatan Berbasis Dekstop.” Infotek Jurnal Informatika Dan Teknologi, vol. 6, no. 2, 20 July 2023, pp. 311–318, https://doi.org/10.29408/jit.v6i2.12974.

[9] Sun, C. (2023). Comparative Study of RSA Encryption and Quantum Encryption. Theoretical and Natural Science, 2(1), pp.121–125. doi: https://doi.org/10.54254/2753-8818/2/20220098