VERIFIKASI TANDA TANGAN DIGITAL BERBASIS DSA DAN FUNGSI *HASH* BLAKE2S UNTUK REKAM MEDIS ELEKTRONIK

SKRIPSI

OLEH
ZUHRUFUL HIKMATUZ ZAHRO
NIM. 220601110062

PROGRAM STUDI MATEMATIKA
FAKULTAS SAINS DAN TEKNOLOGI
UNIVERSITAS ISLAM NEGERI MAULANA MALIK IBRAHIM MALANG
2026

VERIFIKASI TANDA TANGAN DIGITAL BERBASIS DSA DAN FUNGSI *HASH* BLAKE2S UNTUK REKAM MEDIS ELEKTRONIK

SKRIPSI

Diajukan Kepada
Fakultas Sains dan Teknologi
Universitas Islam Negeri Maulana Malik Ibrahim Malang
untuk Memenuhi Salah Satu Persyaratan dalam
Memperoleh Gelar Sarjana Matematika (S.Mat)

Oleh
Zuhruful Hikmatuz Zahro
NIM. 220601110062

PROGRAM STUDI MATEMATIKA
FAKULTAS SAINS DAN TEKNOLOGI
UNIVERSITAS ISLAM NEGERI MAULANA MALIK IBRAHIM MALANG
2026

VERIFIKASI TANDA TANGAN DIGITAL BERBASIS DSA DAN FUNGSI *HASH* BLAKE2S UNTUK REKAM MEDIS ELEKTRONIK

SKRIPSI

Oleh
Zuhruful Hikmatuz Zahro
NIM. 220601110062

Telah Disetujui Untuk Diuji

Malang, 20 Februari 2026.

Dosen Pembimbing I	Dosen Pembimbing II
Muhammad Khudzaifah, M.Si	Dr. Fachrur Rozi, M.Si
NIPPPK. 19900511 202321 1 029	NIP. 19800527 200801 1 012

Mengetahui,
Ketua Program Studi Matematika

Dr. Fachrur Rozi, M.Si
NIP. 19800527 200801 1 012

VERIFIKASI TANDA TANGAN DIGITAL BERBASIS DSA DAN FUNGSI *HASH* BLAKE2S UNTUK REKAM MEDIS ELEKTRONIK

SKRIPSI

Oleh
Zuhruful Hikmatuz Zahro
NIM. 220601110062

Telah Dipertahankan di Depan Dewan Penguji Skripsi
dan Dinyatakan Diterima Sebagai Salah Satu Persyaratan
untuk Memperoleh Gelar Sarjana Matematika (S.Mat)

Tanggal 07 April 2026

Ketua Penguji : Prof. Dr. H. Turmudi, M.Si., Ph.D. .....

Anggota Penguji 1 : Mohammad Nafie Jauhari, M.Si. .....

Anggota Penguji 2 : Muhammad Khudzaifah, M.Si .....

Anggota Penguji 3 : Dr. Fachrur Rozi, M.Si .....

Mengetahui,
Ketua Program Studi Matematika

Dr. Fachrur Rozi, M.Si
NIP. 19800527 200801 1 012

PERNYATAAN KEASLIAN TULISAN

Saya yang bertanda tangan di bawah ini:

Nama	:	Zuhruful Hikmatuz Zahro
NIM	:	220601110062
Program Studi	:	Matematika
Fakultas	:	Sains dan Teknologi
Judul Skripsi	:	Verifikasi Tanda Tangan Digital Berbasis DSA dan Fungsi *Hash* BLAKE2s untuk Rekam Medis Elektronik

Menyatakan dengan sebenarnya bahwa skripsi yang saya tulis ini benar-benar merupakan hasil karya sendiri, bukan merupakan pengambilan data, tulisan, atau pikiran orang lain yang saya akui sebagai hasil tulisan dan pikiran saya sendiri, kecuali dengan mencantumkan sumber cuplikan pada daftar pustaka. Apabila di kemudian hari terbukti atau dapat dibuktikan skripsi ini hasil jiplakan, maka saya bersedia menerima sanksi atas perbuatan tersebut

Malang, 07 April 2026

Yang membuat pernyataan,

Zuhruful Hikmatuz Zahro

NIM. 220601110062

MOTO

“kewajiban kita itu usaha bukan hasil, kewajiban kita itu rajin bukan pintar, maka urusan pintar dan paham, serahkan pada-Nya”

-Buya Mun’im-

“ air mata, lelah, jatuh bangun, dan pertolongan Allah yang selalu datang. Ini bukan akhir, ini awal perjalanan seumur hidup yang menjadi *lillah*”

-Zuhruf-

"pertandingan tinju, belum berakhir saat kamu terjatuh. Mereka memberimu sepuluh detik penuh dalam hitungan mundur untuk membiarkanmu bangkit kembali. Saat keadaan terlalu sulit bagimu, kamu bisa tetap berbaring, sama seperti saat terjatuh. Saat mendapatkan kembali napasmu, kamu bisa bangkit dan bertarung lagi. Aku yakin kehidupanku berada pada hitungan kelima atau keenam."

-Count-

PERSEMBAHAN

Dengan penuh rasa syukur ke hadirat Allah SWT atas segala rahmat, pertolongan dan kemudahan-Nya, sehingga skripsi ini dapat terselesaikan dengan baik, Karya ini penulis persembahkan kepada:

Orang tua penulis yang senantiasa memberikan doa, dukungan serta nasihat demi kesuksesan penulis. Sahabat serta teman-teman penulis yang banyak membantu dalam menyelesaikan skripsi. Terkhusus penulis persembahkan karya sederhana ini untuk diri penulis pribadi. Terima kasih untuk tidak menyerah, terima kasih untuk tetap memutuskan melangkah, terima kasih untuk setiap malam yang penuh sesak oleh kata, terima kasih untuk hati yang tetap kuat saat pikiran berkata lain. Perjalanan ini berat tapi nyatanya kamu kuat.

KATA PENGANTAR

Assalamu'alaikum Warahmatullahi Wabarakatuh

Segala puji bagi Allah *Subhanahu Wa Ta'ala* yang telah melimpahkan rahmat, taufik, dan hidayah-Nya, sehingga penulis dapat menyelesaikan penyusunan skripsi yang berjudul "Verifikasi Tanda Tangan Digital Berbasis DSA dan Fungsi *Hash* Blake2s untuk Rekam Medis Elektronik". Skripsi ini disusun sebagai salah satu syarat untuk memperoleh gelar sarjana pada Program Studi Matematika, Fakultas Sains dan Teknologi, Universitas Islam Negeri Maulana Malik Ibrahim Malang.

Penulis sadar, di balik selesainya naskah ini, ada banyak pihak yang jasanya tak akan pernah bisa saya balas. Terutama, saya ingin menyampaikan rasa terima kasih yang sebesar-besarnya dan setulus-tulusnya kepada:

1. Prof. Dr. Hj. Ilfi Nur Diana, M.Si., CAHRM., CRMP. selaku Rektor Universitas Islam Negeri Maulana Malik Ibrahim Malang.
2. Dr. Agus Mulyono, M.Kes. selaku Dekan Fakultas Sains dan Teknologi Universitas Islam Negeri Maulana Malik Ibrahim Malang.
3. Dr. Fachrur Rozi, M.Si. selaku Ketua Program Studi Matematika, Fakultas Sains dan Teknologi, Universitas Islam Negeri Maulana Malik Ibrahim Malang dan Dosen Pembimbing II yang telah memberikan banyak ilmu, arahan, masukan, serta nasihat selama penyusunan proposal skripsi.
4. Muhammad Khudzaifah, M.Si. selaku Dosen Pembimbing I yang telah memberikan banyak ilmu sekaligus telah sabar memberi arahan, masukan, dan nasihat kepada penulis.
5. Seluruh sivitas akademika Program Studi Matematika, Fakultas Sains dan Teknologi, Universitas Islam Negeri Maulana Malik Ibrahim Malang, terutama jajaran dosen yang telah memberikan ilmu dan wawasan selama masa perkuliahan.
6. Kedua orang tua penulis, Sri Wahyuni sebagai ibunda penulis, yang mendukung tidak hanya melalui doa terbaik, namun juga dukungan kasih sayang serta semangat yang tiada henti. Ayahanda, Wiyono, yang selalu bersabar membimbing penulis dalam setiap langkah yang penulis tempuh.

Terima kasih juga pada adik-adik, yang menjadi salah satu alasan dan motivasi bagi penulis untuk terus berusaha dan menyelesaikan karya ilmiah ini sebagai bentuk tanggung jawab serta proses memantaskan diri menjadi seorang kakak

7. Penghuni grup "Ranu Regulo" dan Nada, yang telah banyak mendukung, membersamai, memberikan semangat dan waktu yang berharga selama proses penelitian. Terima kasih untuk banyaknya penghiburan yang penulis dapat selama proses penyusunan skripsi ini.
8. Sahabat-sahabat penulis, Nasywa Nur Fildzah dan Aris Fadhilah, yang senantiasa menjadi tempat berbagi cerita, serta memberi doa tulus di setiap langkah. Meskipun tidak terlibat langsung dalam dunia perkuliahan, dukungan dari sahabat berarti besar bagi penulis.
9. Para "BigHit Boys" yang banyak menemani penulis selama proses penulisan skripsi ini. Penulis ucapkan banyak terima kasih atas banyaknya semangat dan harapan yang disalurkan melalui musik yang penulis dengar.
10. Teman-teman "Mathdeux" angkatan 2022 yang berbagi pengalaman serta dukungan selama proses penyusunan skripsi ini.

Semoga Allah SWT. memberikan balasan yang terbaik atas segala bantuan dan kebaikan yang telah diberikan kepada penulis. Penulis menyadari bahwa dalam penyusunan skripsi ini terdapat banyak kekurangan, baik dari segi isi maupun penulisan. Oleh karena itu, penulis mengharapkan kritik dan saran yang membangun demi perbaikan di masa mendatang. Akhir kata, semoga skripsi ini dapat memberikan manfaat bagi penulis sendiri maupun bagi pembaca yang memerlukannya, khususnya dalam bidang kriptografi.

Wassalamu'alaikum Warahmatullahi Wabarakatuh

Malang, 07 April 2026

Penulis

DAFTAR ISI

DAFTAR TABEL

DAFTAR GAMBAR

DAFTAR SIMBOL

p	: Bilangan prima besar dalam algoritma DSA
q	: Faktor prima dari $(p - 1)$ pada algoritma DSA
g	: Generator dalam algoritma DSA
x	: Kunci privat (*private key*)
y	: Kunci publik (*public key*), dihitung dari $y = g^x \ mod \ p$
k	: Bilangan acak sementara
r	: Komponen pertama tanda tangan digital
s	: Komponen kedua tanda tangan digital
$H(m)$	: Nilai *hash* dari pesan atau dokumen m
m	: Pesan atau dokumen yang ditandatangani
w	: Invers modulo dari s dalam proses verifikasi
u_1	: Nilai antara dalam proses verifikasi, dihitung dari $(H(m) \cdot w) \ mod \ q$
u_2	: Nilai antara dalam proses verifikasi, dihitung dari $(r \cdot w) \ mod \ q$
v	: Hasil akhir verifikasi tanda tangan digital
mod	: Operasi modulo (sisa pembagian)
$\equiv$	: Notasi kongruensi dalam aritmetika modular
h_{i}	: Nilai inisialisasi (*initialization vector*) pada BLAKE2s
$\oplus$	: Operasi XOR

DAFTAR LAMPIRAN

ABSTRAK

Zahro, Zuhruful Hikmatuz. 2026. **Verifikasi Tanda Tangan Digital Berbasis DSA dan Fungsi *Hash* BLAKE2s untuk Rekam Medis Elektronik.** Skripsi. Program Studi Matematika, Fakultas Sains dan Teknologi, UIN Maulana Malik Ibrahim Malang. Pembimbing: (I) Muhammad Khudzaifah, M.Si. (II) Dr. Fachrur Rozi, M.Si.

Kata Kunci: *Digital Signature Algorithm*; Rekam Medis Elektronik; Tanda Tangan Digital; Verifikasi.

Rekam medis elektronik termasuk kebutuhan penting dalam layanan kesehatan, namun menyimpan secara elektronik rentan terhadap pemalsuan dan manipulasi data. Penelitian ini bertujuan untuk membangun sistem verifikasi yang dapat memastikan keaslian dan integritas dokumen RME melalui penggabungan algoritma *Digital Signature Algorithm* dan fungsi *hash* BLAKE2s. 25 dokumen RME dengan format PDF diuji menggunakan enam perlakuan berbeda yang menggambarkan kemungkinan modifikasi di dunia nyata. Setiap dokumen melewati tahap membuat pasangan kunci, pembuatan tanda tangan digital, dan pemeriksaan validasi. Melalui penelitian, didapatkan bahwa sistem verifikasi dapat mendeteksi seluruh modifikasi dengan akurasi 100%. Fungsi *hash* BLAKE2s memiliki sensitifitas pada satu bit data dengan berubahnya nilai *hash* yang dihasilkan. Sementara algoritma DSA memastikan keaslian dokumen melalui verifikasi perbandingan elemen tanda tangan. Penelitian ini juga memenuhi keterkaitan dengan ajaran Islam. Sistem verifikasi ini mewujudkan tanggung jawab dalam melindungi data, mencegah kerugian, serta menjaga martabat dan privasi pasien sesuai dengan Q.S. Al-Hujurat ayat 12. Penelitian ini merekomendasikan penerapan DSA dan BLAKE2s dalam sistem RME, tidak hanya untuk menjamin keamanan data, tetapi juga bentuk ikhtiar yang selaras dengan nilai etika dan agama.

ABSTRACT

Zahro, Zuhruful Hikmatuz. 2026. **Digital Signature Verification Based on DSA and BLAKE2s Hash Function for Electronic Medical Records.** Thesis. Mathematics Study Program, Faculty of Science and Technology, Universitas Islam Negeri Maulana Malik Ibrahim Malang. Supervisor: (I) Muhammad Khudzaifah, M.Si. (II) Dr. Fachrur Rozi, M.Si.

Keywords: Digital Signature Algorithm; electronic medical records; Digital Signatures; Verification.

Electronic medical records are an essential necessity in healthcare, but storing them electronically is vulnerable to falsification and data manipulation. This study aims to build a verification system that can ensure the authenticity and integrity data of RME documents through the incorporation of the Digital Signature Algorithm and the BLAKE2s hash function. 25 RME documents in PDF format were tested using six different treatments that illustrate the possibility of modification in the real world. Each document goes through the stages of creating a key pair, creating a digital signature, and signature verification. Through research, it was found that the verification system can detect all modifications with 100% accuracy. The BLAKE2s hash function has a sensitivity to a single bit changes, resulting in a completely different hash value. While the DSA algorithm ensures the authenticity of documents through comparative verification of signature elements. This research also fulfills the connection with Islamic values. This verification system realizes the responsibility to protect data, prevent loss, and maintain the dignity and privacy of patients in accordance with Q.S. Al-Hujurat verse 12. This study recommends the implementation of DSA and BLAKE2s in the RME system, not only to ensure data security, but also a form of effort that is in harmony with ethical and religious values.

مستخلص البحث

الزهرة زخرف، الحكمة. ٢٠٢٦. **التحقق من التوقيع الرقمي القائم على *DSA* ووظيفة التجزئة *BLAKE2s* للسجلات الطبية الإلكترونية**. البحث العلمي.قسمالرياضيات، كلية العلوم والتكنولوجيا، جامعة مولانا مالك إبراهيممولانامالك أبراهيم الاسلامية الحكومية ،مالانج. المشرف: (الأولى) محمدخديفة، الماجستير. (الثاني) الدكتور فخرالرازي،الماجستير .

الكلمات الأساسية: خوارزمية التوقيع الرقمي؛ السجلات الطبية الإلكترونية؛ توقيعات رقمية ؛ التحقق.

تعد السجلات الطبية الإلكترونية ضرورة أساسية في الرعاية الصحية، لكن تخزينها إلكترونيا عرضة للتزوير والتلاعب بالبيانات. هدفت هذه الدراسة إلى بناء نظام تحقق يمكنه ضمان مصداقية بيانات سلامة مستندات *RME* من خلال دمج خوارزمية التوقيع الرقمي ودالة التجزئة *BLAKE2s*. تم اختبار 25 وثيقة RME بصيغة *PDF* باستخدام ست معالجات مختلفة توضح إمكانية التعديل في العالم الحقيقي. مر كل مستند بمراحل إنشاء زوج مفاتيح، وإنشاء توقيع رقمي، والتحقق من صحة الشيكات. من خلال البحث، تبين أن نظام التحقق يمكنه اكتشاف جميع التعديلات بدقة 100٪. دالة التجزئة *BLAKE2s* لها حساسية تجاه بت واحد من البيانات مع التغير في قيمة التجزئة الناتجة . بينما تضمن خوارزمية DSA أصالة المستندات من خلال التحقق المقارن من عناصر التوقيع. كما أن هذا البحث يفي بالارتباط بالتعاليم الإسلامية. يدرك هذا النظام مسؤولية حماية البيانات، ومنع الفقدان، والحفاظ على كرامة وخصوصية المرضى وفقا لآية القدس الحجارة الآية 12. توصي هذه الدراسة بتطبيق *DSA* و*BLAKE2s* في نظام *RME*، ليس فقط لضمان أمن البيانات، بل أيضا لشكل من أشكال الجهد المتناغم مع القيم الأخلاقية والدينية.

BAB I

PENDAHULUAN

1.1 Latar Belakang

Kemajuan teknologi informasi yang berkembang dengan cepat telah mengubah cara manusia menyimpan, mengirim, dan memeriksa data di dunia yang serba digital (Uslu dan Stausberg, 2021). Dengan meningkatnya penggunaan digitalisasi, maka timbul pula tuntutan dalam perlindungan data digital yang harus terpenuhi. Setiap data yang dikirim secara digital memiliki kemungkinan untuk bisa dipalsukan, diubah, maupun disalahgunakan. Sehingga untuk mengurangi kemungkinan tersebut diperlukan suatu sistem yang dapat memastikan keaslian dan keutuhan data (Mukti dan Setiawan, 2020). Menurut Nuraeni dkk. (2025) salah satu metode yang dibuat untuk tujuan memastikan keaslian dan keutuhan data adalah tanda tangan digital. Metode ini menggunakan cara kerja kriptografi untuk menjamin identitas pengirim dan keaslian pesan tetap terjaga.

Penerapan tanda tangan digital juga sangat relevan dalam sistem Rekam Medis Elektronik (RME). RME adalah bentuk digitalisasi data kesehatan pasien yang umumnya berisi informasi pribadi, riwayat medis, dan hasil pemeriksaan, di mana informasi ini sangat rentan terhadap masalah keamanan. Menurut Uslu dan Stausberg (2021), digitalisasi dalam bidang kesehatan memberi keuntungan yang besar dalam efisiensi layanan dan kemudahan akses informasi, namun tentunya juga memiliki kebutuhan besar dalam segi perlindungan data pasien.

Di Indonesia, penerapan RME diatur oleh Peraturan Menteri Kesehatan No. 17 Tahun 2023. Peraturan ini menegaskan bahwa RME bertujuan untuk memperbaiki

kualitas pelayanan kesehatan, meningkatkan keamanan, kerahasiaan, integritas, serta ketersediaan data pasien (Novianti dan Bakhtiar, 2024). Selain itu, Azza dkk. (2024) menyebutkan bahwa tanda tangan digital dalam RME dianggap sah selama memenuhi kriteria keamanan tertentu. Namun, berbagai penelitian terdahulu seperti yang dikatakan Novianti dan Bakhtiar (2024), menunjukkan bahwa terdapat banyak fasilitas kesehatan yang belum memiliki sistem keamanan siber yang cukup, sehingga kemungkinan kebocoran atau penyalahgunaan data tinggi. Maka dari itu, penggunaan algoritma tanda tangan digital dibutuhkan agar menjaga integritas data sekaligus meningkatkan kinerja sistem penyimpanan medis elektronik.

Dalam pembuatan tanda tangan digital, dibutuhkan dua komponen utama, yaitu algoritma tanda tangan digital dan fungsi *hash* kriptografi (Harahap dkk., 2023). Keduanya digunakan untuk menghasilkan tanda tangan digital yang unik, aman serta tidak dapat dipalsukan. Secara matematis, algoritma tanda tangan digital menggunakan konsep kriptografi kunci publik. Salah satu algoritma yang populer dan telah distandarisasi oleh *National Institute of Standards and Technology* (NIST) adalah *Digital Signature Algorithm* (DSA) (Stallings, 2014).

Algoritma DSA dibuat menggunakan logaritma diskrit dalam aritmatika modulo, yang secara sederhana akan sangat sulit untuk menebak tanda tangan digital yang benar apabila tanpa kunci privat yang sepasang dengan tanda tangan digital terkait (Stallings, 2014). Inilah yang membuat algoritma DSA menjadi pilihan tepat digunakan untuk menjaga keaslian dan keamanan data digital. DSA dibuat dengan melalui 3 tahapan utama, yaitu pembuatan sepasang kunci (kunci publik dan kunci privat), pembuatan tanda tangan, dan pengecekan (verifikasi) (Nuraeni dkk., 2025). Pada tahap penandatanganan, pesan yang akan

ditandatangani diubah menjadi nilai *hash*, lalu digabungkan dengan kunci privat untuk menghasilkan tanda tangan digital yang unik. Pada tahap verifikasi, tanda tangan akan diuji dengan kunci publik untuk memastikan tidak ada perubahan pada dokumen yang ditandatangani. Berdasarkan penelitian terbaru Nuraeni dkk. (2025), DSA unggul dalam keseimbangan antara kecepatan, efisiensi, dan ketahanan terhadap serangan kriptografi, ini menjadikan DSA lebih baik dibanding algoritma yang umum digunakan seperti RSA (*Rivest Shamir Adleman*) dalam hal efisiensi dan keamanan tanda tangan digital.

Menurut Stallings (2014) dalam penggunaan algoritma DSA, juga dibutuhkan fungsi *hash* kriptografi untuk memperkuat tanda tangan digital yang dihasilkan. Adapun kegunaan fungsi *hash* adalah untuk mengubah data yang berukuran besar menjadi rangkuman data berukuran tetap (deretan angka dan huruf yang panjangnya selalu sama), yang disebut *message digest* (Mufidah dan Nuha, 2024). Menurut Mufidah dan Nuha (2024) pula, dalam penelitiannya bahwa nilai *hash* ini memiliki sifat yang unik dan berbeda untuk setiap dokumen atau pesan, sehingga ketika terdapat perubahan sekalipun hanya pada satu huruf dalam dokumen, akan menghasilkan nilai *hash* yang berbeda pula. Karakteristik ini menjadikan fungsi *hash* sebagai salah satu komponen utama dalam pembuatan tanda tangan digital serta memastikan keutuhan data.

Dalam penelitian terbaru Mufidah dan Nuha (2024), memaparkan bahwasanya fungsi *hash* yang menunjukkan kinerja yang bagus adalah BLAKE2s. Algoritma ini adalah pengembangan dari model sebelumnya dengan nama BLAKE, dan dibuat untuk menawarkan kecepatan pemrosesan yang lebih baik serta efisiensi memori yang lebih tinggi dibandingkan dengan SHA-2 (*Secure Hash Algorithm* 2) dan

SHA-3. Berdasarkan penelitiannya, BLAKE2s menunjukkan waktu komputasi yang lebih cepat serta efisiensi tinggi tanpa menghilangkan aspek keamanan, sehingga bagus digunakan dalam sistem yang membutuhkan kecepatan dan keandalan tinggi seperti tanda tangan digital.

Oleh karena itu, penggabungan algoritma DSA dan fungsi *hash* BLAKE2s dapat membuat tanda tangan digital yang tidak hanya kuat secara matematis, tetapi juga efisien dalam proses komputasinya (Nuraeni dkk., 2025). Dalam hal itu, DSA berperan untuk memastikan keaslian dan keabsahan tanda tangan, sementara BLAKE2s menjamin keaslian pesan dan dokumen melalui nilai *hash* yang aman dan unik. Kombinasi antara keduanya diharapkan dapat menghasilkan tanda tangan digital yang kuat terhadap serangan dan efisien dalam penggunaan sumber daya.

Dalam penelitiannya, Uslu dan Stausberg (2021) memaparkan bahwa dari sisi sosial, pengembangan sistem tanda tangan digital memberi manfaat dalam membuat kepercayaan akan sistem digital (*digital trust*). Mengingat keamanan data bukan hanya aspek teknis, tetapi juga bentuk tanggung jawab sosial, untuk itu menjaga kerahasiaan dan keaslian informasi dalam dokumen tentunya diperlukan (Mukti dan Setiawan, 2020). Maka penggunaan algoritma yang efektif dan aman akan membuat kepercayaan dalam banyaknya penggunaan digital seperti transaksi keuangan digital hingga dokumen hukum menjadi asli, transparan dan terjaga dari manipulasi (Novianti dan Bakhtiar, 2024).

Dalam pandangan Islam, melindungi keaslian dan kerahasiaan informasi merupakan hal penting, sebagai perwujudan nilai *amanah* yang wajib dipenuhi semua orang. Allah SWT menyatakan dalam Surah An- Nisa’ ayat 58, bahwa:

اِنَّ اللّٰهَ يَأْمُرُكُمْ اَنْ تُؤَدُّوا الْاَمٰنٰتِ اِلٰٓى اَهْلِهَاۙ

"Sesungguhnya Allah menyuruhmu menyampaikan amanat kepada yang berhak menerimanya..." (QS. An-Nisa': 58).

Selain itu, Rasulullah SAW bersabda:

"Tunaikanlah amanah kepada orang yang mempercayaimu dan janganlah kamu mengkhianati orang yang mengkhianatimu" (HR. Abu Dawud No. 3535 dan At-Tirmidzi No. 1264)

Ayat dan hadis di atas menekankan bahwa melindungi data dan informasi menggunakan algoritma yang kuat sebagai bentuk *ikhtiar* akan nilai *amanah* yang menjadi kewajiban moral yang tidak bisa diabaikan. Oleh karena itu penerapan tanda tangan digital menggunakan DSA dan BLAKE2s tidak hanya memperkuat keamanan informasi secara teknis, tetapi juga mencerminkan nilai integritas, kejujuran, serta amanah dalam ajaran Islam (Harahap dkk., 2023).

Berdasarkan penjelasan di atas, penelitian ini bertujuan untuk menganalisis verifikasi tanda tangan digital pada rekam medis elektronik dengan menggabungkan algoritma DSA dan fungsi *hash* BLAKE2s. Oleh karena itu, penggabungan kedua algoritma kriptografi ini, diharapkan dapat menjadi solusi untuk perlindungan keamanan data pasien yang lebih baik.

1.2 Rumusan Masalah

Berdasarkan latar belakang yang telah diuraikan, rumusan masalah dalam penelitian ini adalah:

1. Bagaimana hasil implementasi verifikasi tanda tangan digital berbasis algoritma DSA dan fungsi *hash* BLAKE2s pada dokumen RME?
2. Bagaimana keakuratan verifikasi tanda tangan digital dalam mendeteksi perubahan pada dokumen RME?

1.3 Tujuan Penelitian

Sesuai dengan rumusan masalah yang telah diuraikan, tujuan yang ingin dicapai dalam penelitian ini adalah:

1. Menganalisis hasil penerapan verifikasi tanda tangan digital berbasis algoritma DSA dan fungsi *hash* BLAKE2s pada dokumen RME.
2. Menganalisis keakuratan verifikasi tanda tangan digital dalam mendeteksi perubahan pada dokumen RME.

1.4 Manfaat Penelitian

Berdasarkan latar belakang yang telah diuraikan, penelitian ini memiliki manfaat sebagai berikut:

1. Menambah pengetahuan terkait implementasi tanda tangan digital pada dokumen RME, serta pemahaman dan keterampilan penerapan metode DSA dan fungsi *hash* BLAKE2s.
2. Menambah literatur terkait performa algoritma tanda tangan digital dalam konteks rekam medis elektronik (RME).
3. Memberikan rekomendasi praktis bagi institusi kesehatan dalam penerapan sistem informasi yang lebih aman dan andal.
4. Memberikan dasar ilmiah dan kerangka acuan untuk studi-studi lanjutan mengenai implementasi tanda tangan digital.

1.5 Batasan Masalah

Agar penelitian ini tetap fokus dan terarah, maka ditetapkan batasan-batasan sebagai berikut:

1. Penelitian ini hanya menggunakan fungsi *hash* BLAKE2s dan algoritma tanda tangan digital DSA (*Digital Signature Algorithm*) untuk verifikasi tanda tangan digital pada dokumen rekam medis.
2. Jenis data yang digunakan terbatas pada data simulasi yang merepresentasikan dokumen Rekam Medis Elektronik. Format data yang digunakan dibatasi pada *Portable Document Format* (.PDF). Data penelitian menggunakan data set publik yang diperoleh dari *kaggle.*
3. Aspek yang dikaji terbatas pada keakuratan verifikasi tanda tangan digital dalam membedakan dokumen asli dengan dokumen yang telah diubah, sensitivitas sistem terhadap perubahan data yang diuji dengan berbagai perlakuan pada dokumen, serta keabsahan tanda tangan digital yang dihasilkan melalui kemampuan sistem untuk menjamin integritas dan keaslian dokumen.

1.6 Definisi Istilah

Untuk memastikan pemahaman yang konsisten dalam pembahasan, beberapa istilah teknis dan konsep utama yang digunakan dalam penelitian ini perlu dijelaskan terlebih dahulu. Definisi istilah berikut menjadi acuan dasar agar pembaca dapat mengikuti isi penelitian dengan lebih mudah dan tepat.

1. Rekam Medis Elektronik (RME): merupakan sistem penyimpanan data kesehatan pasien secara elektronik yang menggantikan rekam medis yang menggunakan kertas. Pelaksanaannya di Indonesia diatur oleh Peraturan Menteri Kesehatan No. 17 Tahun 2023. RME bertujuan untuk memperbaiki

kualitas pelayanan kesehatan meningkatkan keamanan, kerahasiaan, integrasi, serta ketersediaan data pasien (Novianti dan Bakhtiar, 2024).

2. Tanda Tangan Digital (*digital signature*): merupakan sistem kriptografi yang digunakan untuk memastikan keaslian, integritas dan keutuhan pesan atau dokumen digital. Sistem ini memungkinkan penerima untuk memastikan bahwa pesan itu benar-benar berasal dari pengirim yang sah dan tidak mengalami perubahan selama pengiriman. Dalam kriptografi, tanda tangan digital biasa diimplementasikan dengan menggunakan fungsi *hash* kriptografi, yang keamanannya bergantung pada karakteristik *hash* (Algazy dkk., 2024).
3. *Digital Signature Algorithm* (DSA): algoritma kriptografi yang bertujuan untuk memastikan keaslian data dan otentikasi identitas penandatangan. Algoritma ini menggunakan sepasang kunci (kunci publik dan kunci privat). Adanya kunci publik adalah untuk menyatakan keaslian tanda tangan tersebut serta dokumen yang dikirim tidak mengalami perubahan selama pengiriman, fungsi kunci privat sendiri adalah untuk membuat tanda tangan yang unik pada setiap dokumen yang berbeda (Nuraeni dkk., 2025).
4. Fungsi *Hash* Kriptografi: merupakan suatu algoritma matematis yang mengubah data berukuran tertentu menjadi nilai unik dengan ukuran tetap (*hash*). Fungsinya untuk memastikan integritas data dimana setiap perubahan kecil pada data asli akan menghasilkan *hash* yang sangat berbeda. Dalam tanda tangan digital hash dipakai untuk membuat ringkasan dokumen yang unik sebelum ditandatangani, menjamin keaslian dan integritas dokumen (Amaludin dan Rahmatulloh, 2024).

5. BLAKE2s: algoritma kriptografi yang merupakan bagian dari fungsi *hash*, dibuat untuk tingkat keamanan tinggi dan kecepatan pemrosesan, khususnya pada sistem 32-bit. Algoritma ini menghasilkan hingga 32 *byte* (256 bit) dan dari segi kinerja lebih unggul dibandingkan algoritma fungsi *hash* lain seperti SHA-256 bit dan SHA3-256. BLAKE2s direkomendasikan sebagai pilihan, khususnya dalam bidang yang membutuhkan keamanan dan keandalan tinggi seperti kesehatan dan industri (Mufidah dan Nuha, 2024).
6. *Non-Repudiation*: merupakan salah satu sifat yang harus terpenuhi untuk sebuah algoritma dikatakan aman untuk diimplementasikan pada tanda tangan digital. *Non-Repudiation* sendiri berarti bahwa menghindari pengirim pesan menyangkal telah mengirim atau menandatangani suatu dokumen, atau penerima menyangkal telah menerima pesan tersebut. Salah satu pihak tidak dapat menolak terkait keaslian suatu dokumen atau data (Harahap dkk., 2023).

BAB II

KAJIAN TEORI

2.1 *Digital Signature Algorithm* (DSA)

2.1.1 Kriptografi

Kriptografi adalah ilmu yang mempelajari cara melindungi informasi melalui penyandian untuk mengamankan komunikasi dan informasi dari pihak yang tidak berhak. Secara etimologis, kata kriptografi berasal dari bahasa Yunani yakni *kryptos* yang berarti tersembunyi dan *graphein* yang berarti menulis, atau dimaksudkan sebagai tulisan yang tersembunyi. Secara umum, kriptografi terbagi menjadi dua kategori utama yakni :

1. Kriptografi Simetris adalah metode yang menggunakan satu kunci yang sama untuk enkripsi dan deskripsi.
2. Kriptografi Asimetris adalah metode yang menggunakan pasangan kunci publik dan privat (dua kunci yang berbeda).

Selain dua kategori tersebut, kriptografi juga meliputi fungsi *hash* kriptografi. Dalam kriptografi, fungsi *hash* diterapkan untuk menjamin integritas data. Fungsi *hash* didefinisikan sebagai:

$$h = H(m) \tag{2. 1}$$

seperti ditunjukkan dalam Persamaan 2.1, bahwa m adalah pesan yang asli dan h adalah nilai *hash* dengan panjang n bit. Berdasarkan penuturan Stallings (2014), fungsi *hash* yang aman harus memenuhi tiga karakteristik utama, yaitu:

1. *Preimage Resistance,* sulit untuk menentukan pesan asli (m) dari nilai *hash* tertentu (h).

2. *Second Preimage Resistance*, diberikan pesan awal (m_1), sulit menemukan pesan lain (m_2) dengan $m_2 \neq m_1$, sehingga $H(m_2) = H(m_1)$.
3. *Collision Resistant*, sulit menemukan dua pesan yang berbeda m_1 dan m_2 (dengan $m_1 \neq m_2$) sehingga $H(m_1) = H(m_2)$.

2.1.2 Kongruensi

Kongruensi adalah suatu konsep dalam teori bilangan yang digunakan untuk mengetahui apakah suatu bilangan memiliki sisa ketika dibagi oleh bilangan tertentu. Irawan dkk., (2017) menjelaskan definisi terkait kongruensi yakni, jika sebuah bilangan bulat M yang tidak nol membagi selisih $a - b$, maka definisi kongruensi dapat ditulis dalam Persamaan 2.2.

$$a \equiv b \ (mod\ M) \Leftrightarrow M \mid (a - b) \qquad (2.\ 2)$$

Dengan a dan b adalah bilangan bulat dan M adalah modulus yang merupakan selisih $a - b$ habis dibagi M. Contohnya:

$$27 \equiv 2 \ (mod\ 5)$$

karena $27 - 2 = 25 = 5 \times 5$, juga bisa dilihat dalam bentuk $27 = 2 + 5 \times 5$, maka keduanya memiliki sisa pembagian yang sama terhadap 5.

2.1.3 Konsep *Digital Signature Algorithm* (DSA)

Digital Signature Algorithm (DSA) merupakan salah satu algoritma kriptografi kunci publik munir (2019). Dalam bukunya, Munir (2019) menjelaskan bahwa, berbeda dengan algoritma kunci publik lain yang dapat digunakan untuk enkripsi, DSA memiliki fungsi yang khusus, yaitu untuk pembuatan tanda tangan digital. Secara umum, DSA memiliki dua fungsi utama:

1. Pembuatan Tanda Tangan (*Signature Generation*)
2. Verifikasi Tanda Tangan (*Signature Verification*)

Seperti halnya sistem kriptografi kunci-publik pada umumnya, DSA menggunakan sepasang kunci yakni kunci privat dan kunci publik. Proses membuat tanda tangan dilakukan dengan menggunakan kunci privat yang dirahasiakan oleh pemilik. Sebaliknya, proses untuk memeriksa keabsahan tanda tangan tersebut dilakukan dengan menggunakan kunci publik yang disebarluaskan. Dalam prosesnya, fungsi *hash* yang digunakan dalam penelitian ini adalah fungsi *hash* BLAKE2s. Fungsi ini bertugas untuk mengonversi pesan asli menjadi *message digest*, yang kemudian akan ditandatangani. Mekanisme pengerjaan algoritma DSA melalui tahap berikut:

1. Parameter DSA

   p : bilangan prima

   q : bilangan prima pembagi habis dari $(p-1)$

   $g = h^{(p-1)/q} \ mod \ p, 1 < h < p-1$ (2. 3)

   di mana:

   x = bilangan bulat yang lebih kecil dari q (kunci privat)

   m = data yang akan diberi tanda tangan digital

   $y = g^x \ mod \ p$ (2. 4)

2. Pembangkitan kunci

   a. Pilih bilangan prima p dan q

   b. Hitung nilai g dengan rumus

   $$g = h^{(p-1)/q} \ mod \ p$$

   c. Tentukan pasangan kunci, yakni kunci privat (x) dan kunci publik (y) menggunakan perhitungan Persamaan 2.4 dengan syarat memenuhi Persamaan 2.5

$$1 < x < q - 1\ (kunci\ privat), \qquad (2.5)$$
$$y = g^x\ mod\ p\ (kunci\ publik)$$

Pilih angka x dan nilai y akan digunakan untuk memverifikasi tanda tangan

3. Pembuatan tanda tangan

   a. Ubah pesan m menjadi *message digest*, pesan m (data yang akan ditandatangani) diproses dengan fungsi *hash* untuk menghasilkan $h(m)$, yaitu sidik jari digital dalam pesan

   b. Pilih angka acak, tentukan angka bilangan acak k dengan syarat $1 < k < q - 1$, di mana pemakaian k hanya sekali dan harus berbeda di setiap menandatangani dokumen

   c. Hitung nilai tanda tangan (r, s) dengan perhitungan Persamaan 2.6

$$r = (g^k\ mod\ p)\ mod\ q \qquad (2.6)$$
$$s = k^{-1} \cdot (H(m) + x \cdot r)\ mod\ q$$

   dimana r dan s adalah hasil tanda tangan digital, k^{-1} adalah invers perkalian modulo q dari k, sementara $h(m)$ merupakan hasil *hash* pesan, serta x adalah kunci privat.

   d. Kirim pesan dan tanda tangan. Pesan m dikirim dengan nilai tanda tangan (r, s) dan penerima dapat memverifikasi keaslian pesan menggunakan kunci publik.

4. Verifikasi keabsahan tanda tangan

   Dalam langkah verifikasi setelah perhitungan dengan menggunakan Persamaan (2.7), jika hasil $v = r$, maka tanda tangan valid dimana pesan asli dan benar-benar dibuat pemilik kunci privat. Penggunaan Persamaan

2.7 adalah untuk membuktikan bahwa pesan berasal dari pengirim yang sah dan tidak ada perubahan (dimodifikasi).

Hitung:

$$w = s^{-1} mod\ q$$
$$u1 = (H(m) \cdot w)\ mod\ q$$
$$u2 = (r \cdot w)\ mod\ q$$
$$v = ((g^{u1} \cdot y^{u2})\ mod\ p)\ mod\ q \quad (2.7)$$

2.1.4 Tanda Tangan Digital

Tanda tangan digital adalah salah satu penerapan teknologi kriptografi yang bertujuan untuk memastikan keaslian (*authenticity*), integritas (*integrity*), dan *non-repudiation* (anti penyangkalan) dari dokumen elektronik atau pesan digital. Dalam sistem keamanan informasi, tanda tangan digital berperan penting sebagai cara yang menjamin pesan atau dokumen memang berasal dari pengirim yang valid dan tidak mengalami modifikasi selama proses pengiriman (Stallings, 2014). Proses pembuatan tanda tangan digital dilakukan dengan menghasilkan nilai *message digest* melalui fungsi *hash* kriptografi, yang selanjutnya dienkripsi menggunakan kunci privat dari pengirim untuk membuat tanda tangan digital. Pihak penerima kemudian dapat melakukan verifikasi dengan mencocokkan hasil *hash* dokumen yang diterima dengan tanda tangan digital menggunakan kunci publik dari pengirim. Mekanisme ini menjamin bahwa hanya pemilik kunci privat yang valid yang dapat menghasilkan tanda tangan yang sah (Amaludin dan Rahmatulloh, 2024).

Dalam bukunya, Stallings (2014) menjelaskan bahwa tanda tangan digital adalah hasil penerapan algoritma kriptografi dengan kunci publik, di mana setiap

individu memiliki sepasang kunci, kunci privat untuk melakukan tanda tangan dan kunci publik untuk memverifikasi tanda tangan tersebut. Oleh karena itu, tanda tangan digital dapat dianggap sebagai sistem otentikasi elektronik yang menggunakan kriptografi untuk menjamin keaslian identitas pengirim, melindungi integritas data dari perubahan yang tidak sah, serta memberikan jaminan hukum terhadap transaksi digital yang dilakukan secara elektronik. Teknologi ini merupakan komponen penting dalam sistem keamanan informasi kontemporer, khususnya dalam penggunaan dokumen elektronik dan sistem rekam medis digital.

2.2 Fungsi *Hash*

2.2.1 ASCII

Menurut Stallings (2014), ASCII (*American Standard Code for Information Interchange*) adalah standar pengkodean karakter yang dipakai untuk mempresentasikan teks dalam bentuk angka biner agar bisa diproses pada komputer. Tiap huruf, angka, maupun simbol memiliki nilai yang unik dan berbeda, keunikan ini merupakan dasar dalam pengolahan data digital. Secara formal, proses pengubahan karakter menjadi biner bisa dinyatakan sebagai:

$$ASCII(x) = Kode\ biner\ dari\ karakter\ x \tag{2. 8}$$

dengan x adalah karakter yang diubah ke dalam bentuk biner. Contohnya:

$$ASCII(A) = 65_{10} = 41_{16} = 01000001_2$$

yang menunjukkan bahwa karakter A memiliki nilai 65 dalam desimal, 41 dalam heksadesimal, dan 01000001 dalam biner. Contoh konversi karakter ke ASCII ditunjukkan dalam Tabel 2.1.

Penggunaan ASCII penting dalam pembuatan tanda tangan digital. Nuraeni dkk., (2025) menyebutkan bahwa data dokumen yang akan di *hash* dan ditandatangani harus diubah ke bentuk biner ASCII agar fungsi *hash* dan algoritma yang dipakai dapat memproses data secara konsisten.

Tabel 2.1 Contoh Hasil ASCII

Karakter	Desimal	Heksadesimal	Biner
A	65	41	01000001
B	66	42	01000010
C	67	43	01000011
a	97	61	01100001
b	98	62	01100010
c	99	63	01100011
0	48	30	00110000
1	49	31	00110001
@	64	40	01000000
!	33	21	00100001

2.2.2 Operasi XOR (*Exclusive* OR)

Operasi XOR adalah salah satu dasar penting dalam sistem digital yang mempunyai aturan operasi yang unik, yaitu menghasilkan nilai benar jika dua masukkan berbeda dan salah jika dua masukan sama. Secara matematis, operasi XOR dapat ditulis dalam Persamaan 2.9.

$$A \oplus B = (A + B) \bmod 2 \qquad (2.9)$$

dengan A dan B merupakan bit masukan (0 atau 1), simbol $\oplus$ menyatakan operasi XOR, dan mod 2 berarti hasil pembagian yang terbatas dalam 2 nilai yakni 0 atau 1. Cara kerjanya adalah operasi ini mendeteksi perbedaan. Operasi ini akan menghasilkan keluaran dengan nilai 1, ketika kedua masukan memiliki nilai yang berbeda, misalnya satu bernilai 0 dan yang lain bernilai 1. Sebaliknya

jika kedua masukan sama, baik keduanya 0 atau keduanya 1, maka keluaran akan selalu 0 (Heidari dkk., 2025). Tabel Kebenaran operasi XOR ditunjukkan dalam Tabel 2.2.

Tabel 2.2 Operasi XOR

A	B	A XOR B
0	0	0
0	1	1
1	0	1
1	1	0

Sifat–sifat penting dalam operasi XOR adalah sebagai berikut:

$$A \oplus 0 = A$$

$$A \oplus A = 0$$

$$A \oplus B = B \oplus A\ (komutatif)$$

$$(A \oplus B) \oplus C = A \oplus (B \oplus C)\ \ (asosiatif)$$

dengan A, B, C merupakan bilangan biner. Sifat sifat ini menunjukkan bahwa operasi XOR memiliki sifat komutatif dan asosiatif, serta memiliki elemen identitas 0. Heidari dkk. (2025) menjelaskan dalam penelitiannya, bahwa sifat unik ini menjadikan operasi XOR sangat berguna dalam berbagai sistem keamanan data, kriptografi dan deteksi kesalahan.

2.2.3 Konsep Fungsi *Hash* BLAKE2s

BLAKE2s adalah pengembangan dari algoritma BLAKE yang merupakan finalis dalam kompetisi SHA-3 yang diadakan oleh NIST (*National Institute of Standards and Technology*). BLAKE2s dibuat untuk tingkat keamanan yang sama dengan SHA-3 dan SHA-256, tapi dengan kecepatan *hash* yang lebih cepat dan penggunaan sumber daya yang lebih efektif. Mufidah dan Nuha (2024) mengatakan bahwa BLAKE2s memiliki metode yang menggunakan operasi

sederhana seperti penjumlahan modulo 2^{32}, operasi XOR, serta rotasi bit, sehingga BLAKE2s sangat efisien pada perangkat dengan sumber daya terbatas, seperti sistem IoT (*Internet of Things*) dan aplikasi berbasis web. Mekanisme pengerjaan BLAKE2s dapat dijelaskan melalui tahapan berikut:

1. *Input* dan *Padding*: Data masukan (dapat berupa pesan atau *key* opsional dengan panjang maksimum 32 *byte*) terlebih dahulu diproses menggunakan teknik *padding*, yaitu penambahan bit 0 hingga panjangnya sesuai dengan blok 16 kata. *Digest* yang dihasilkan dapat ditentukan antara 1–32 byte sesuai kebutuhan aplikasi.
2. *Initialization Vector* (IV): BLAKE2s memanfaatkan IV berupa konstanta 32-bit yang bersifat tetap. IV ini berfungsi sebagai titik awal dalam proses *hash* dan telah ditentukan oleh perancang algoritma.
3. *Chain Value* (h): Setelah inisialisasi IV, dibentuk *chain value* awal sepanjang 256 bit yang terus diperbarui selama proses pembuatan *hash*. Nilai awal *chain value* antara lain:

$$h0 = 6B08E647 \qquad h4 = 510E527F$$
$$h1 = BB67AC85 \qquad h5 = 9B05688C$$
$$h2 = 3C6EF372 \qquad h6 = 1F83D9AB$$
$$h3 = A45FF53A \qquad h7 = 5BE0CD19$$

4. Fungsi Kompresi (*Compression Function*): Pada tahap ini, IV, *permutation*, dan *chain value* dipadukan untuk menghasilkan nilai awal v. Nilai ini kemudian diproses dalam 10 ronde (*rounds*) dengan operasi aritmatika modular sederhana, XOR, dan rotasi bit.

5. *Round Function* dan G *Function*: Setiap ronde melibatkan delapan fungsi G0 sampai G7 yang mengoperasikan variabel (a, b, c, d). Operasi modular, XOR, dan rotasi bit yang digunakan bertujuan memperkuat difusi, sehingga setiap bit *input* mempengaruhi keseluruhan *output*.
6. Finalisasi: Pada tahap akhir, nilai *chain value* diperbarui dengan melakukan XOR terhadap nilai awal dan hasil kompresi. Proses ini menghasilkan *message digest* yang bersifat unik untuk setiap *input*.

2.3 Rekam Medis Elektronik (RME)

Rekam Medis Elektronik (RME) adalah sistem digital yang digunakan untuk mencatat, menyimpan, dan mengelola informasi kesehatan pasien secara elektronik. Sistem ini mengganti pencatatan yang awalnya tradisional menggunakan kertas menjadi menggunakan sistem digital yang memudahkan akses, pertukaran dokumen, serta analisis data pasien dengan cepat dan aman. Novianti dan Bakhtiar (2024) menjelaskan, bahwa RME tidak hanya berperan sebagai tempat penyimpanan data pasien, tetapi juga memiliki peran penting pada transformasi digital di bidang kesehatan. Dimana RME ini mendukung efisiensi layanan dan kebijakan kesehatan nasional berbasis data. Dalam konteks ini, RME meliputi rekam medis, hasil pengujian, resep obat, serta tindakan medis yang diberikan kepada pasien. Melalui penerapan sistem ini, tenaga medis dapat mengakses informasi pasien secara langsung agar proses pengambilan keputusan klinis menjadi lebih cepat dan tepat.

Menurut Torab dkk. (2025) RME merupakan kebutuhan mendesak dalam sistem kesehatan militer untuk mengatasi tantangan operasional. Dalam keadaan

darurat medis di lapangan, akses cepat dan langsung terhadap riwayat kesehatan, alergi, dan obat-obatan individu sangat dibutuhkan untuk penanganan yang efektif. RME memastikan perawatan kesehatan yang lancar bagi personel militer sekalipun sering mengalami perpindahan tugas atau transisi ke fasilitas kesehatan sipil, sehingga mencegah terputusnya riwayat kesehatan. Aspek keamanan data menjadi sangat penting mengingat sensitivitas informasi kesehatan personel militer, yang dilindungi melalui mekanisme enkripsi dan protokol akses yang ketat. Selain itu, RME mendukung pemantauan berkelanjutan terhadap kondisi kesehatan mental (seperti gangguan stres pasca trauma (PTSD)), meningkatkan efisiensi administrasi untuk penghematan biaya, serta menyediakan data yang berharga untuk tujuan penelitian dan pengembangan protokol kesehatan yang lebih baik di masa depan (Torab dkk., 2025).

Keunggulan utama penerapan RME terletak pada peningkatan layanan kesehatan, efisiensi dalam administrasi, dan kemampuan integrasi data pada fasilitas kesehatan. Uslu dan Stausberg, (2021) melalui penelitiannya, menemukan bahwa sebagian besar penelitian menunjukkan perbaikan kualitas perawatan pasien setelah implementasi sistem rekam medis elektronik. Ini terjadi karena tenaga kesehatan memiliki akses langsung ke informasi pasien yang lengkap dan terbaru saat melakukan keputusan klinis. Di samping itu, penerapan RME juga dapat meningkatkan efisiensi biaya operasional, terutama dapat mempercepat proses dokumentasi. Uslu dan Stausberg (2021) mencatat bahwa lebih dari 50% penelitian yang mereka teliti menunjukkan penurunan biaya operasional dan waktu administrasi setelah penerapan RME. Sekalipun menawarkan banyak keuntungan, RME juga memiliki beberapa tantangan teknis dan hukum. Novianti dan Bakhtiar,

(2024) menjelaskan bahwa masalah paling utama dalam penggunaan RME adalah perlindungan keamanan dan privasi data pasien. Informasi kesehatan termasuk sebagai data sensitif, sehingga kebocoran atau penyalahgunaan informasi dapat menimbulkan masalah hukum.

2.4 Kajian Integrasi Topik dengan Al-Qur'an dan Hadits

2.4.1 Prinsip Kejujuran dan *Maqasid Syari'ah*

Nilai kejujuran (*ash-shidqu*) dan menunaikan amanah merupakan pilar etika dalam Islam yang memiliki relevansi langsung dengan integritas data. Firman Allah dalam Q.S. Al-Baqarah [2]: 283, "...Dan janganlah kamu menyembunyikan kesaksian...", menegaskan imperatif untuk bersikap transparan dan jujur. Dalam konteks RME, setiap entri data mulai dari diagnosis, resep, hingga laporan operasi merupakan sebuah "kesaksian" medis yang harus dijaga keaslian dan kebenarannya. Dalam RME, setiap data yang dimasukkan seperti diagnosis, resep, dan laporan tindakan medis adalah bentuk kesaksian yang memiliki nilai hukum. Dengan demikian, keaslian dan kebenaran data itu harus dipertahankan dengan sangat bertanggung jawab. Dengan cara ini, sistem tanda tangan digital adalah bentuk implementasi nilai moral *ash-shidqu* dalam mempertahankan kejujuran dan kebenaran informasi medis.

Selain itu, menurut Muda dkk. (2023) menjelaskan bahwa prinsip *dar'u al-mafasid* lebih diutamakan dibandingkan dengan *jalb al-mashalih* (menghindari kerusakan lebih penting daripada mencapai kemaslahatan) memberikan dasar dalam implementasi keamanan RME. Peluang kerusakan seperti kebocoran data pasien, pemalsuan dokumen medis, atau gangguan sistem dapat berakibat serius

mulai dari kesalahan diagnosis, kesalahan pengobatan, sampai ancaman terhadap keselamatan jiwa pasien. Kerusakan ini pastinya lebih merugikan dibandingkan keuntungan yang didapat dari penghematan biaya dengan menggunakan sistem yang tidak aman.

Oleh karena itu, mengutamakan teknologi kriptografi dan tanda tangan digital yang kuat untuk RME tidak hanya merupakan kebutuhan, tetapi juga wujud nyata dari prinsip *dar'u al-mafasid*, yaitu usaha untuk mencegah kerugian yang lebih besar demi kepentingan bersama. Ini sesuai dengan tujuan utama syariat Islam (*maqashid syari'ah*) yang meliputi yaitu menjaga agama (*hifdz al-din*), menjaga jiwa (*hifdz al-nafs*), menjaga akal (*hifdz al-aql*), menjaga keturunan (*hifdz al-nasl)*, dan menjaga harta (*hifdz al-mal*) (Muda dkk., 2023).

2.4.2 Prinsip Menjaga Kehormatan dan Rahasia

Islam sangat menghargai privasi dan martabat setiap individu. Dalam Islam larangan untuk menghormati orang lain tertera dalam Q.S. Al-Hujurat [49]: 12, "...Dan janganlah kamu mencari-cari kesalahan orang lain...", mengajarkan agar tidak mengganggu urusan pribadi orang lain. Tafsir ayat ini menegaskan terkait menjaga kerahasiaan orang lain, dimana informasi kesehatan pasien merupakan salah satu rahasia yang paling pribadi, yang jika digunakan untuk hal yang tidak seharusnya, dapat merusak kehormatan dan martabat orang yang berkaitan. Karena itu, memastikan kerahasiaan data RME adalah bukan hanya tanggung jawab profesional, tetapi juga kewajiban. Sabda Nabi SAW, "Siapa yang menutupi kesalahan seorang muslim, maka Allah akan menutupi kesalahannya di dunia dan akhirat" (HR. Ibnu Majah no. 2544), memberikan dorongan spiritual yang mendalam bagi pekerja dalam bidang kesehatan dan pengembang sistem

untuk merancang mekanisme keamanan (seperti enkripsi dan kontrol akses yang ketat) yang mampu melindungi kerahasiaan (*confidentiality*) data pasien secara optimal. Ini merupakan wujud nyata dalam menutupi kehinaan dan menghormati kepercayaan yang diberikan oleh pasien.

2.5 Kajian Topik dengan Teori Pendukung

Keamanan Rekam Medis Elektronik (RME) adalah syarat penting dalam sistem informasi kesehatan, karena data pasien memiliki karakter yang sensitif dan peraturan hukum yang tinggi. Digitalisasi layanan kesehatan memerlukan adanya sistem yang dapat memastikan kerahasiaan, integritas, dan kemudahan akses data. Berdasarkan evaluasi kebutuhan dan perkembangan teknologi yang ada, terdapat kekurangan penelitian yang perlu segera diatasi. Pada penelitian Torab dkk., (2025) menyatakan bahwa Rekam Medis Elektronik (RME) sangat diperlukan dalam sistem kesehatan militer guna menjamin akses cepat, kelanjutan perawatan, serta perlindungan data kesehatan yang bersifat sensitif. Di sisi lain, tanda tangan digital yang menurut Stallings (2014) merupakan suatu metode kriptografi yang terbukti dapat menjamin keaslian identitas, integritas data, serta kekuatan hukum pada dokumen elektronik. Namun, penggunaan tanda tangan digital secara spesifik dalam konteks RME masih belum diteliti secara mendalam. Karena itu, studi yang mengevaluasi implementasi algoritma tanda tangan digital yang digabungkan dengan fungsi *hash* pada dokumen RME menjadi penting.

Menurut Uslu dan Stausberg, (2021) implementasi RME meningkatkan efisiensi serta layanan, tetapi juga menimbulkan risiko keamanan yang harus dicegah dengan metode kriptografi yang terpercaya. Salah satu metode kriptografi yang efisien

dalam melindungi keamanan data RME adalah tanda tangan elektronik. Amaludin dan Rahmatulloh (2024) menyatakan bahwa tanda tangan digital memanfaatkan kombinasi kunci publik dan privat untuk melakukan enkripsi dan verifikasi, sehingga hanya pihak berwenang yang dapat menandatangani dokumen.

Penelitian Nuraeni dkk. (2025) menunjukkan bahwa penggunaan *Digital Signature Algorithm* (DSA) dalam sistem tanda tangan digital dapat memperkuat keamanan dan efisiensi proses otentikasi data. Selain algoritma tanda tangan, fungsi *hash* kriptografi juga memiliki peranan penting dalam menjaga integritas data. Mufidah dan Nuha (2024) menunjukkan bahwa BLAKE2s sangat efisien sehingga menjadikannya pilihan yang sangat baik untuk fungsi *hash* seperti SHA-3. Fungsi *hash* ini menjamin bahwa setiap perubahan kecil pada data dapat teridentifikasi melalui nilai *message digest* yang khas. Kombinasi DSA dan BLAKE2s membuat keseimbangan antara keamanan dan efektivitas dalam pengelolaan data medis elektronik. Mukti dan Setiawan (2020) mengatakan dalam penelitiannya bahwa metode kriptografi ini penting untuk sistem RME karena dapat digunakan tanpa memerlukan beban komputasi yang besar. Oleh karena itu, integrasi tanda tangan digital yang menggunakan DSA dan fungsi *hash* BLAKE2s dalam sistem RME diharapkan dapat melindungi otentikasi, integritas, dan *non-repudiation* data medis secara efektif serta mendukung perubahan digital dalam sektor kesehatan.

BAB III

METODE PENELITIAN

3.1 Jenis Penelitian

Penelitian ini menggunakan metode kuantitatif eksperimental, metode ini dipilih dengan tujuan untuk mengimplementasikan dan menguji keberhasilan proses tanda tangan digital pada dokumen RME. Pada pendekatan ini, perlakuan tertentu diberikan kepada subjek penelitian untuk diamati dan diukur hasilnya.

3.2 Pra Penelitian

Sebelum penelitian dilaksanakan, dilakukan tahap pra-penelitian untuk memastikan kesiapan pelaksanaan penelitian. Tahap ini meliputi:

1. Studi Awal
    a. Kajian Literatur: Dilakukan pengkajian terkait algoritma DSA dan fungsi *hash* BLAKE2s, serta kelengkapan rekam medis.
    b. Ketersediaan Data: Data yang digunakan berasal dari data set publik *Kaggle* yang dapat diakses secara legal. Data set mencakup variasi parameter yakni nama, umur, jenis kelamin, golongan darah, kondisi medis, *provider* kesehatan (nama dokter dan rumah sakit), asuransi dan biaya pengobatan.
    c. Kemampuan Teknis: Peneliti telah mempelajari dasar-dasar kriptografi, bahasa pemrograman *Python,* dan penggunaan platform *Google Collab.*
2. Penyusunan Instrumen Penelitian

a. Penyusunan Kode Program: Kode untuk implementasi disusun dan diuji dengan data sederhana sebelum digunakan pada data simulasi RME.
b. Penyiapan Data set: Data *dummy* RME dalam format PDF disiapkan berdasarkan data set asli dari *Kaggle*.

3. Uji Coba Awal
   a. Dilakukan uji coba menggunakan fungsi *hash* BLAKE2s dan proses pembangkitan tanda tangan DSA dengan data percobaan untuk memastikan tidak ada kesalahan dalam pemrograman.
   b. Diverifikasi bahwa *output hash* konsisten (*deterministik*) dan sensitif terhadap perubahan data (*avalanche effect*) melalui perlakuan yang diterapkan pada setiap dokumen.

3.3 Tahapan Penelitian

Tahapan penelitian dilakukan dengan beberapa tahapan yang sistematis dan terstruktur untuk memastikan keaslian hasil yang diperoleh. Secara garis besar, tahapan penelitian ini dapat diuraikan sebagai berikut:

3.3.1 Pembangkitan Pasangan Kunci DSA

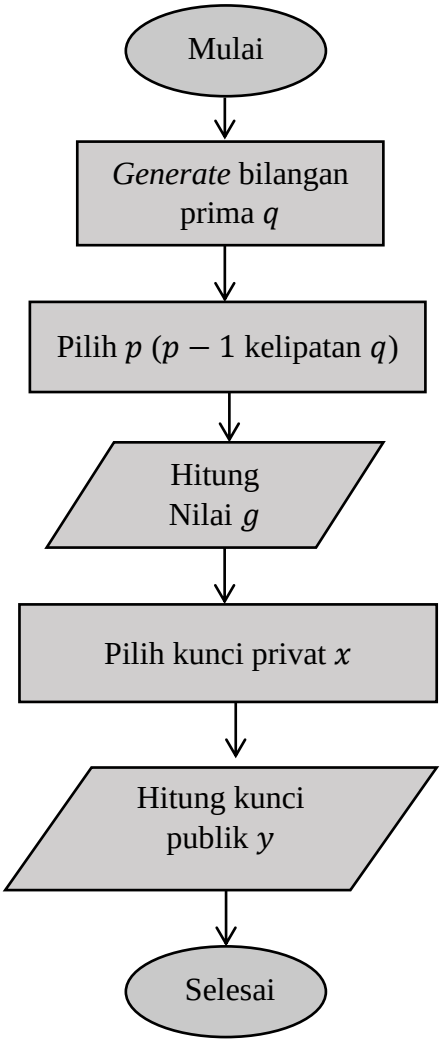
Proses pertama yang dilakukan adalah membuat pasangan kunci yang akan digunakan untuk menandatangani dan memverifikasi dokumen. Tahapan penelitian pada proses ini mengikuti Gambar 3.1 dan dijelaskan sebagai berikut:

1. Mulai: Tahap awal dalam proses membuat pasangan kunci
2. *Input* parameter DSA (p, q, g): Sistem memasukkan parameter-parameter yang telah ditentukan untuk algoritma DSA. Parameter ini terdiri dari:
   a. p : Bilangan prima (minimal 1024 bit mengikuti standar NIST).

b. q : Bilangan prima pembagi habis dari $(p-1)$, berukuran 256 bit.

c. g : Bilangan generator yang memenuhi $g = h^{(p-1)/q} \, mod \, p$, dengan $1 < h < p-1$.

3. Pilih kunci privat (x): Sistem memilih angka acak x yang memenuhi kondisi $1 < x < q-1$.
4. Hitung kunci publik (y): Sistem menghitung kunci publik y menggunakan rumus $y = g^x \, mod \, p$.
5. Simpan kunci: Parameter (p, q, g) dan pasangan kunci (x, y) yang telah dibuat disimpan..
6. Selesai: Proses pembuatan kunci selesai.



Gambar 3.1 *Flowchart* Pembangkitan Kunci

3.3.2 Pembuatan Tanda Tangan Digital

Setelah memperoleh pasangan kunci mengikuti Gambar 3.1, langkah berikutnya adalah membuat tanda tangan digital untuk dokumen RME. Tahapan penelitian pada proses ini mengikuti alur yang ditunjukkan dalam Gambar 3.2 dan dijelaskan sebagai berikut:

1. Mulai: Langkah awal dari proses tanda tangan pada dokumen.
2. Masukkan dokumen RME (m): Sistem menerima *input* dokumen RME asli dalam format PDF yang akan ditandatangani.

3. Hitung nilai *hash* menggunakan BLAKE2s: Dokumen RME (m) diproses dengan fungsi *hash* BLAKE2s.
4. Pilih bilangan acak (k): Sistem memilih bilangan acak k dengan syarat $1 < k < q - 1$.
5. Hitung tanda tangan (r, s): Sistem menghitung dua nilai r dan s menggunakan rumus:

a. $r = (g^k \ mod \ p) \ mod \ q$

b. $s = k^{-1} \cdot (H(m) + x \cdot r) \ mod \ q$

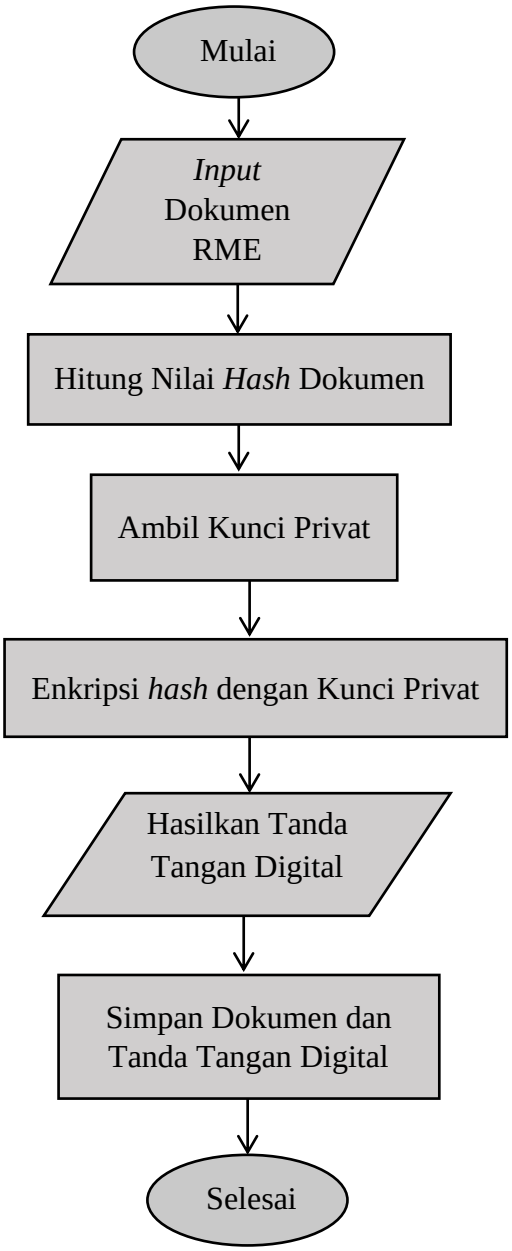
6. *Output* (dokumen dan tanda tangan): Proses ini menghasilkan dua *output* yakni dokumen asli RME (m) dan tanda tangan digital (r, s).
7. Selesai: Proses penandatanganan dokumen selesai.

3.3.3 Verifikasi Tanda Tangan Digital

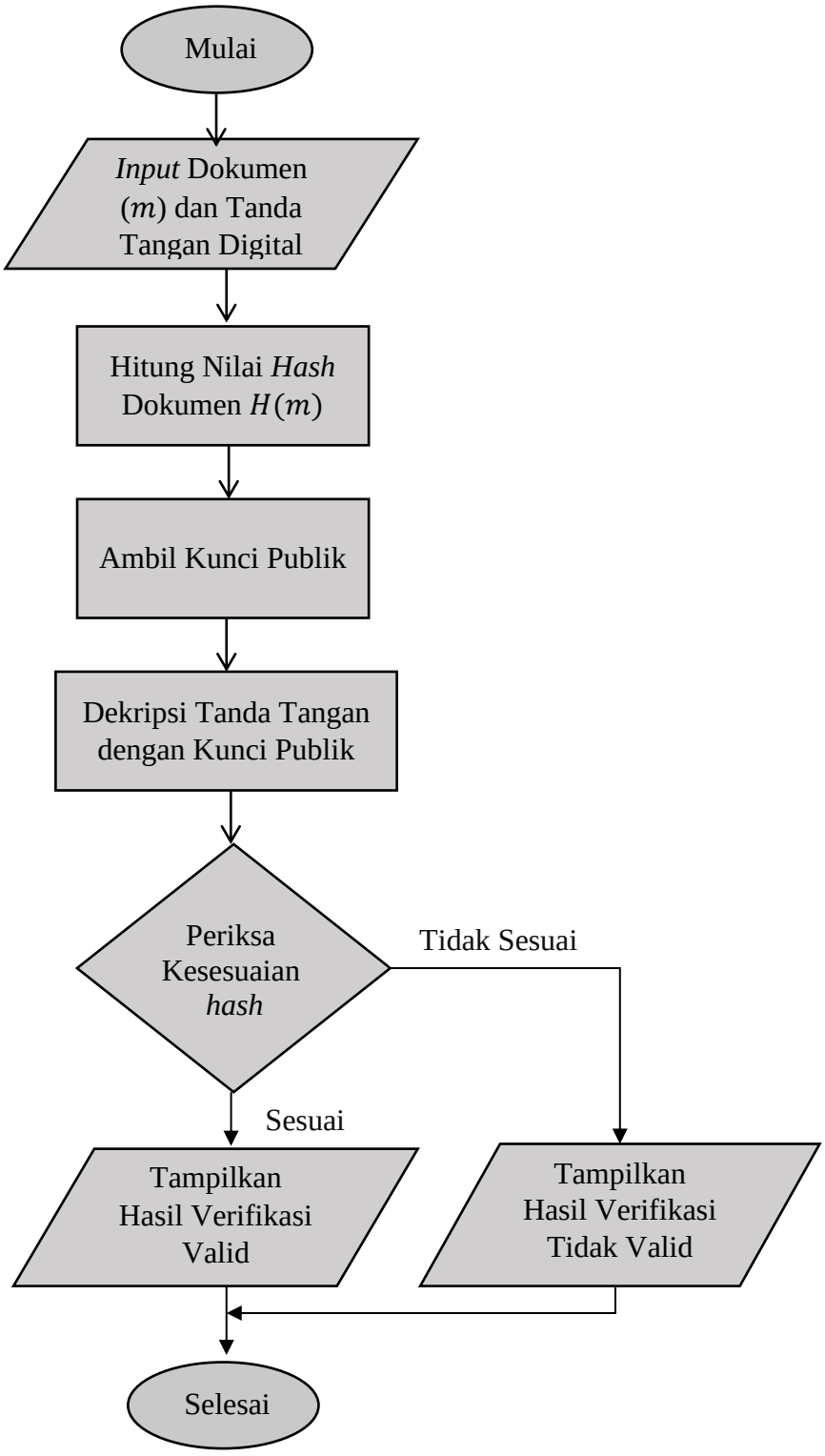
Pada pihak penerima, keaslian serta integritas dokumen yang sudah ditandatangani dapat diuji melalui proses verifikasi. Tahapan Penelitian pada proses ini sesuai dengan Gambar 3.3 dan dijelaskan sebagai berikut:

1. Mulai: Langkah awal proses pemeriksaan dokumen.
2. *Input* (dokumen, tanda tangan, kunci publik):
    a. Dokumen RME yang akan diperiksa (m').
    b. Tanda tangan digital (r, s) milik dokumen terkait.
    c. Kunci publik (y) milik pengirim yang berwenang.
3. Hitung nilai *hash* dokumen: Dokumen yang diterima dihitung nilai *hash* (m') untuk menghasilkan $H(m')$.
4. Hitung nilai verifikasi (v): Sistem melakukan rangkaian perhitungan matematis $(w, u1, u2, v)$ untuk mendapatkan nilai v.

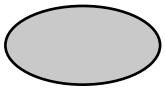
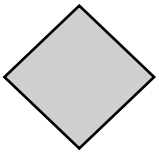
5. Bandingkan v dengan r: Sistem membandingkan hasil perhitungan v dengan komponen tanda tangan r.
6. *Output* (hasil verifikasi): Sistem memberikan hasil akhir, dinyatakan valid jika $v = r$, dan dinyatakan tidak valid ketika $v \neq r$.
7. Selesai: Proses verifikasi selesai.



Gambar 3.2 *Flowchart* Pembuatan Tanda Tangan Digital



Gambar 3.3 *Flowchart* Verifikasi Tanda Tangan Digital

Simbol	Keterangan
	Menunjukkan proses awal (mulai) dan akhir (selesai).
	*Input* dan *output* yang digunakan untuk proses memasukkan atau mengeluarkan data.
	Menunjukkan proses penelitian atau langkah pemrosesan data.
	Menunjukkan percabangan logika, digunakan untuk pengambilan keputusan.
	Menunjukkan arah proses dari satu langkah ke langkah berikutnya

BAB IV

HASIL DAN PEMBAHASAN

4.1 Pembangkitan Kunci DSA

Tahap awal dalam sistem tanda tangan digital adalah menghasilkan pasangan kunci yang akan dipakai untuk menandatangani dan memeriksa dokumen. Dalam penelitian ini, pembuatan kunci dilakukan dengan bahasa pemrograman *Python* mengikuti langkah-langkah yang terdapat pada Gambar 3.1. Kode program yang utuh untuk proses ini terdapat di Lampiran 8.

Parameter yang digunakan mengikuti standar *National Institute of Standards and Technology* (NIST) FIPS 186-4 dengan level keamanan 2048 bit. Pemilihan ukuran kunci ini berdasarkan saran NIST untuk sistem kriptografi sampai tahun 2030. Sesuai dengan yang ditunjukkan dalam Persamaan 2.3 yakni, parameter yang dipakai dalam algoritma DSA meliputi:

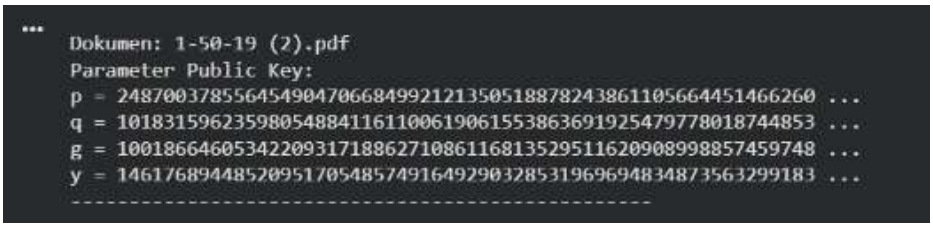
1. Bilangan prima q (256 bit) yang pertama kali dihasilkan
2. Bilangan prima p (2048 bit) dengan ketentuan $(p-1)$ adalah kelipatan q.
3. Generator g dengan urutan q, dihitung dari $g = h^{(p-1)/q} \ mod \ p$ untuk nilai h tertentu $(1 < h < p-1)$

Setelah parameter ditetapkan, langkah berikutnya adalah menghasilkan pasangan kunci melalui langkah berikut:

1. Kunci privat x ditentukan sebagai angka acak yang memenuhi kondisi $1 < x < q-1$. Kunci ini bersifat rahasia dan hanya diketahui oleh pihak yang menandatangani

2. Kunci publik y dihitung dengan persamaan $y = g^x \ mod \ p$, sesuai dengan Persamaan 2.4. Kunci ini diberikan kepada pihak yang akan melakukan verifikasi tanda tangan.

Dalam implementasi ini, pasangan kunci yang telah dihasilkan, disimpan secara terpisah dari dokumen asli untuk melindungi kerahasiaan kunci privat. Melalui sistem, 25 dokumen RME diunggah manual pada kode program, serta seluruhnya berhasil diproses. Untuk setiap dokumen, dihasilkan pasangan kunci DSA melalui sistem secara otomatis. Setiap dokumen RME dibangkitkan pasangan kunci yang berbeda, sehingga tanda tangan digital yang didapat bersifat khas. Contoh *output* kode program satu dokumen pada proses pembangkitan kunci ditunjukkan pada Gambar 4.1.



Gambar 4.1 Pasangan Kunci DSA

Didapatkan melalui penelitian bahwa, seluruh dokumen berhasil menghasilkan pasangan kunci yang unik dengan hasil pada Lampiran 13. Karena kunci privat bersifat rahasia, maka dalam hal ini hanya ditampilkan sebagian nilai kunci publik sebagai representasi hasil langkah pembangkitan pasangan kunci menggunakan algoritma DSA.

Selanjutnya, untuk ilustrasi perhitungan manual pada subbab 4.2 dan 4.3, digunakan parameter ukuran kecil untuk memudahkan. Parameter ini dihasilkan dengan mengikuti langkah-langkah pada Gambar 3.1.

1. Dipilih bilangan prima $p = 59$

2. Dipilih angka prima $q = 29$ (membagi (p $-$ 1))
3. Generator $g = 3$ dipilih yang memenuhi kondisi sebagai elemen pembangun subgrup berorde q
4. Dipilih kunci pribadi $x = 17$ (memenuhi $1 < 17 < 28$)
5. Dipilih kunci publik $y = g^x \, mod \, p = 3^{17} \, mod \, 59 = 19$

   Perhitungan menggunakan metode *square* dan *multipy* untuk memudahkan perhitungan yakni sebagai berikut:

   $3^1 = 3 \, mod \, 59 = 3$

   $3^8 = (3^4)^2 = 22^2 = 484 \, mod \, 59 = 12$

   $3^{16} = (3^8)^2 = 12^2 = 144 \, mod \, 59 = 26$

   $3^{17} = 3^{16} \cdot 3^1 = 26 \cdot 3 = 78$

   $= 78 \, mod \, 59 = 19$

Parameter kecil ini tidak diterapkan dalam implementasi nyata, melainkan untuk untuk merinci langkah matematis yang mendasari algoritma DSA. Dalam praktiknya, seluruh proses pembuatan kunci dilakukan melalui kode program dengan menggunakan bilangan prima besar (2048 bit) sesuai dengan standar keamanan. Pasangan kunci yang telah dihasilkan ini selanjutnya akan digunakan dalam pembuatan tanda tangan digital dan proses verifikasi.

4.2 Proses Pembuatan Tanda Tangan Digital

4.2.1 Rancangan Pengujian

Implementasi tanda tangan digital dalam penelitian ini, dimana menggunakan algoritma fungsi *hash* BLAKE2s dan DSA dilakukan pada dua puluh lima dokumen berformat .pdf dengan bantuan bahasa pemrograman *Python*

yakni *google colab*. Uji coba dilakukan untuk melihat kemampuan sistem tanda tangan digital dengan melihat perubahan pada dokumen rekam medis. Uji coba dilakukan pada dokumen asli dan dokumen yang telah diubah dengan beberapa jenis perlakuan yang berbeda. Perubahan yang diimplementasikan bertujuan untuk mengamati dampaknya pada nilai *hash* dan keaslian tanda tangan digital. Pengujian dilakukan dengan menghitung terlebih dahulu nilai *hash* dan tanda tangan digital dari dokumen yang asli. Lalu, hasil dokumen asli tersebut dibandingkan dengan dokumen modifikasi. Langkah perbandingan yang dilakukan adalah dengan menghitung *hash* dari setiap dokumen dan melakukan verifikasi tanda tangan digital untuk menentukan apakah dokumen tersebut masih utuh dan tidak ada perubahan setelah penandatanganan.

Terdapat enam jenis perlakuan yang berbeda yang akan dilakukan pada tahap verifikasi. Enam jenis perlakuan tersebut masuk dalam rancangan pengujian yang dalam penelitian ini ditampilkan pada Tabel 4.1.

Tabel 4.1 Perlakuan pada Dokumen Asli

No.	Dokumen	Perlakuan
1.	Dokumen asli	Dokumen tanpa modifikasi
2.	Dokumen Perlakuan 1 (P1)	Satu karakter diganti
3.	Dokumen Perlakuan 2 (P2)	Dua karakter ditukar
4.	Dokumen Perlakuan 3 (P3)	Perubahan besar-kecil huruf
5.	Dokumen Perlakuan 4 (P4)	Penghapusan dan penulisan ulang
6.	Dokumen Perlakuan 5 (P5)	Dikirim melalui platform komunikasi
7.	Dokumen Perlakuan 6 (P6)	Edit, hapus, dan simpan ulang

4.2.2 Perhitungan *Hash*

Langkah awal dalam pembuatan tanda tangan digital yakni menghasilkan nilai *hash*, dimana dalam penelitian ini adalah menggunakan algoritma fungsi

hash BLAKE2s. Pengujian dilakukan pada dokumen yang disertakan pada Gambar 4.2.


RUMAH SAKIT PROOF D-DAYS

Data Rekam Medis Pasien #9

Name:

Age: 82

Gender: Male

Blood Type:

Medi:

Diagnosed:

Doctor :

Hospital :

Insurance Provider:

Billing Amount :

Tanda Tangan Digital:

(Digital Signature)

7

Gambar 4.2 Contoh Dokumen Rekam Medis

Tabel 4.2 Contoh Hasil ASCII Desimal dan Heksadesimal

No	Karakter	ASCII Decimal	ASCII Hex
1.	R	82	0x52
2.	U	85	0x55
3.	M	77	0x4D
4.	A	65	0x41
5.	H	72	0x48
6.	[spasi]	32	0x20

Perhitungan nilai *hash* pada dokumen mencakup tahap-tahap berikut:

1. *Preprocessing*

Dalam langkah ini meliputi pengubahan teks menjadi representasi heksadesimal (*textstream*) dan melakukan *padding.* Tujuannya adalah untuk memastikan dokumen selalu dalam format yang sama. Pertama, dokumen rekam medis diubah menjadi kode ASCII untuk setiap karakter, lalu konversi

kode ASCII ke heksadesimal mengikuti Persamaan 2.8. Sehingga didapat hasil kode ASCII dari Gambar 4.1, sebagaimana tercantum pada Lampiran 2 dengan panjang total 422 *byte*. Contoh 6 byte pertama dalam Gambar 4.1 disertakan dalam Tabel 4.2.

Konversi kode ASCII ke heksadesimal dengan cara membagi bilangan dengan 16, lalu ambil *quotient* (hasil pembagian bulat) sebagai digit pertama heksadesimal dan *remander* (sisa pembagian) sebagai digit kedua dalam heksadesimal. Karena BLAKE2s memproses data setiap 64 *byte*, maka didapatkan hasil 64 *byte* pertama dalam heksadesimal yakni:

52 55 4D 41 48 20 53 41 4B 49 54 20 50 52 4F 4F

46 20 44 2D 44 41 59 53 0A 4A 6C 2E 20 59 6F 75

20 4E 65 76 65 72 20 57 61 6C 6B 20 41 6C 6F 6E

65 2C 20 42 75 6C 6C 65 74 70 72 6F 6F 66 0A 30

Dalam Gambar 4.1 didapatkan 422 *byte* yang nantinya akan menjadi 7 potongan 64 *byte*, diantaranya 6 *byte* awal adalah isi dokumen Gambar 4.1 dan 1 potongan terakhir berisi 38 *byte* ditambah *padding* serta *lenght* (jumlah *byte* dalam heksadesimal) sampai berjumlah 64 *byte*. Setelah *padding* didapatkan 7 blok data yakni 6 blok berisi 64 *byte* data asli dan 1 blok terakhir berisi 64 *byte* dengan tambahan *padding* dan *length*.

2. Proses *compression function*

Dalam langkah ini data diproses per blok dan dijalankan secara berulang. Contoh proses fungsi kompresi algoritma BLAKE2s dengan menggunakan blok pertama dari data rekam medis Gambar 4.2. Blok pertama terdiri dari 64 *byte* di atas, kemudian dibagi menjadi 16 word berukuran 32-bit dengan

format *little-endian* (simpan secara terbalik), sehingga didapat *message word* sebagai berikut:

$m[0] = 0x414D5552$ $m[8] = 0x76654E20,$

$m[1] = 0x41532048,$ $m[9] = 0x57207265$

$m[2] = 0x2054494B,$ $m[10] = 0x206B6C61$

$m[3] = 0x4F4F5250,$ $m[11] = 0x6E6F6C41$

$m[4] = 0x2D442046,$ $m[12] = 0x42202C65$

$m[5] = 0x53594144,$ $m[13] = 0x656C6C75$

$m[6] = 0x2E6C4A0A,$ $m[14] = 0x6F727074$

$m[7] = 0x756F5920,$ $m[15] = 0x300A666F$

State internal BLAKE2s terdiri dari 16 kata 32-bit. Delapan *word* pertama diinisialisasi dengan menggunakan konstanta *Initialization Vector* (IV), sementara delapan *word* berikutnya adalah salinan dari IV itu. IV adalah nilai awal standar yang digunakan sebelum memproses data pertama kali dalam *hash* dimana sudah ditetapkan dan dispesifikasi algoritma. Secara matematis, *state* awal ditulis sebagai berikut:

$v0 = 0x6A09E667,$ $v8 = v0$

$v1 = 0xBB67AE85,$ $v9 = v1$

$v2 = 0x3C6EF372,$ $v10 = v2$

$v3 = 0xA54FF53A,$ $v11 = v3$

$v4 = 0x510E527F,$ $v12 = v4$

$v5 = 0x9B05688C,$ $v13 = v5$

$v6 = 0x1F83D9AB,$ $v14 = v6$

$v7 = 0x5BE0CD19,$ $v15 = v7$

untuk setiap blok 64 *byte* dilakukan langkah kompresi fungsi G. Fungsi G merupakan operasi utama dalam BLAKE2s. Fungsi G terdapat 8 langkah, setiap ronde dalam BLAKE2s menjalankan 8 Fungsi G yang mengombinasikan 4 register dan 2 *message word*. Pemilihan pasangan register dan *message word* mengikuti tabel permutasi. Jadi urutan G di ronde 1 yakni:

a. $G(v0, v4, v8, v12, m[0], m[1])$
b. $G(v1, v5, v9, v13, m[2], m[3])$
c. $G(v2, v6, v10, v14, m[4], m[5])$
d. $G(v3, v7, v11, v15, m[6], m[7])$
e. $G(v0, v5, v10, v15, m[8], m[9])$
f. $G(v1, v6, v11, v12, m[10], m[11])$
g. $G(v2, v7, v8, v13, m[12], m[13])$
h. $G(v3, v4, v9, v14, m[14], m[15])$

Sebagai contoh langkah dalam proses BLAKE2s, ditunjukkan satu kali operasi Fungsi G pada satu ronde. Dalam contoh, digunakan parameter indeks $a = 0, b = 4, c = 8$ dan $d = 13$, dengan *input message word* $m[0]$ dan $m[1]$. Nilai awal yang digunakan adalah:

$$v0 = 0x6A09E667 = 1779033703$$

$$v4 = 0x510E527F = 1359893119$$

$$v8 = 0x6A09E667 = 1779033703$$

$$v12 = 0x510E527F = 1359893119$$

$$m0 = 0x414D5552 = 1094991186$$

$$m1 = 0x41532048 = 1095778376$$

Nilai-nilai ini yang akan dimasukkan dalam delapan Fungsi G berikut:

$a1 = (v0 + v4 + m0)\ mod\ 2^{32}$

$d2 = \text{ROTR}_{32}(v12 \oplus a1, 16)$

$c3 = (v8 + d2)\ mod\ 2^{32}$

$b4 = \text{ROTR}_{32}(v4 \oplus c3, 12)$

$a5 = (a1 + b4 + m1)\ mod\ 2^{32}$

$d6 = \text{ROTR}_{32}(d2 \oplus a5, 8)$

$c7 = (c3 + d6) mod\ 2^{32}$

$b8 = \text{ROTR}_{32}(b4 \oplus c7, 7)$

Dalam perhitungan Fungsi G, setiap langkah melibatkan Persamaan 2.2 dan Persamaan 2.9. Maka penerapan Fungsi G adalah sebagai berikut:

a. $a1 = (v0 + v4 + m0)\ mod\ 2^{32}$

$= (0x6A09E667 + 0x510E527F + 0x414D5552) mod\ 2^{32}$

$\equiv (0xBB1838E6 + 0x44D5552)\ mod\ 2^{32}$

$a1 \equiv 0xFC658E38$

b. $d2 = \text{ROTR}_{32}(v12 \oplus a1, 16)$

$= \text{ROTR}_{32}(0x510E527F \oplus 0xFC658E38, 16)$

$= \text{ROTR}_{16}(0xAD6BDC47)$

$d2 = 0xDC47AD6B$

c. $c3 = (v8 + d2)\ mod\ 2^{32}$

$= (0x6A9E667 + 0xDC47AD6B) mod\ 2^{32}$

$c3 \equiv 0x46593D2$

d. $b4 = \text{ROTR}_{32}(v4 \oplus c3, 12)$

$= \text{ROTR}_{32}(0x510E527F \oplus 0x465193D2, 12)$

$= \text{ROTR}_{12}(175FC1AD) = 0x1AD175FC$

e. $a5 = (a1 + b4 + m1)\ mod\ 2^{32}$

$= (0xFC658E38 + 0x1AD175FC + 0x41532048) mod\ 2^{32}$

$= (0x17370434 + 0x41532048)\ mod\ 2^{32}$

$a5 \equiv 0x588A247C$

f. $d6 = \text{ROTR}_{32}(d2 \oplus a5, 8)$

$= \text{ROTR}_{32}(0xDC47AD6B \oplus 0x588A247C, 8)$

$= \text{ROTR}_{8}(0x84CD8917)$

$d6 = 0x1784CD89$

g. $c7 = (c3 + d6)\ mod\ 2^{32}$

$= (0x1AD175FC + 0x5dd66158)\ mod\ 2^{32}$

$c7 \equiv 0x5DD6615B$

h. $b8 = \text{ROTR}_{32}(b4 \oplus c7, 7)$

$= \text{ROTR}_{32}(0x1AD175FC \oplus 0x5DD6615B, 7)$

$= \text{ROTR}_{7}(0x470714A7)$

$b8 = 0x4E8E0E29$

Setelah satu kali Fungsi G dilakukan, hanya empat elemen *state* v, yakni $v0, v4, v8$ dan $v12$ yang berubah, sedangkan elemen yang lain tetap. Dengan demikian didapatkan hasil dari Fungsi G pertama adalah:

$$v0 = 0x588A247C,$$

$$v4 = 0x4E8E0E29,$$

$$v8 = 0x5DD6615B,$$

$$v12 = 0x1784CD89.$$

Langkah ini menggambarkan penerapan penjumlahan modulo. Operasi XOR, dan rotasi bit yang berfungsi untuk mencampur data dalam algoritma *hash*. Perhitungan manual dalam langkah ini dibatasi dalam satu kali penerapan Fungsi G untuk menunjukkan proses dasar algoritma BLAKE2s. Tahap selanjutnya yakni mengulangi pengerjaan Fungsi G sebanyak delapan kali untuk membentuk satu ronde penuh pada setiap blok pesan. Karena langkah dilakukan secara berulang, maka perhitungan selanjutnya dilakukan menggunakan kode program untuk menjaga keakuratan hasil. Setelah dilakukan satu ronde pertama algoritma BLAKE2s dengan menggunakan kode program pada Lampiran 3, maka didapatkan *state* $v0$ sampai $v15$ adalah berikut:

$$v[0] = 0x1cacb14a \qquad v[8] = 0x2e96d536$$
$$v[1] = 0x14349fbc \qquad v[9] = 0xbfe7e03a$$
$$v[2] = 0xbb2ca74b \qquad v[10] = 0xa7e3e360$$
$$v[3] = 0xa96220eb \qquad v[11] = 0x9494400d$$
$$v[4] = 0x678ba116 \qquad v[12] = 0xeadc74ca$$
$$v[5] = 0x7e270f76 \qquad v[13] = 0x60eb7915$$
$$v[6] = 0x1d05b39d \qquad v[14] = 0x95dd73f5$$
$$v[7] = 0x15b98513 \qquad v[15] = 0xdf61a4f5$$

Untuk perhitungan selanjutnya yakni mengulangi proses yang sama yakni Fungsi G dengan sigma $[r]$ berbeda di setiap ronde. Urutan penggunaan sigma dalam BLAKE2s adalah meliputi:

$$\sigma[0] = [0, 1, 2, 3, 4, 5, 6, 7, 8, 9, 10, 11, 12, 13, 14, 15]$$
$$\sigma[1] = [14, 10, 4, 8, 9, 15, 13, 6, 1, 12, 0, 2, 11, 7, 5, 3]$$

$$\sigma[2] = [11, 8, 12, 0, 5, 2, 15, 13, 10, 14, 3, 6, 7, 1, 9, 4]$$

$$\sigma[3] = [7, 9, 3, 1, 13, 12, 11, 14, 2, 6, 5, 10, 4, 0, 15, 8]$$

$$\sigma[4] = [9, 0, 5, 7, 2, 4, 10, 15, 14, 1, 11, 12, 6, 8, 3, 13]$$

$$\sigma[5] = [2, 12, 6, 10, 0, 11, 8, 3, 4, 13, 7, 5, 15, 14, 1, 9]$$

$$\sigma[6] = [12, 5, 1, 15, 14, 13, 4, 10, 0, 7, 6, 3, 9, 2, 8, 11]$$

$$\sigma[7] = [13, 11, 7, 14, 12, 1, 3, 9, 5, 0, 15, 4, 8, 6, 2, 10]$$

$$\sigma[8] = [6, 15, 14, 9, 11, 3, 0, 8, 12, 2, 13, 7, 1, 4, 10, 5]$$

$$\sigma[9] = [10, 2, 8, 4, 7, 6, 1, 5, 15, 11, 9, 14, 3, 12, 13, 0]$$

Untuk proses perhitungan pada ronde selanjutnya, dilakukan menggunakan kode program untuk menjaga konsistensi dan efisiensi perhitungan. Setelah proses percampuran pada setiap blok selesai, yakni dengan melakukan 10 putaran fungsi kompresi. Setiap putaran terdiri dari delapan operasi Fungsi G yang mengolah *state* internal $v[0 \ldots 15]$ dan *message word* $m[0 \ldots .15]$. Nilai $v[0 \ldots 15]$ yang didapat, yang telah di campur total dengan nilai *Initialization Vector* (IV), nilai tersebut masih perlu dilakukan proses final agar dapat menghasilkan nilai *hash* yang valid.

3. Final

Setelah semua ronde fungsi kompresi pada algoritma BLAKE2s selesai, didapatkan nilai *state* internal $v[0 \ldots 15]$. *State* ini yang kemudian di proses dalam langkah final. Proses final dilakukan dengan menggabungkan *state* dari kompresi dengan nilai *hash* sebelumnya dengan menggunakan operasi XOR. Tujuan langkah final ini adalah untuk mengubah *state internal* terakhir menjadi hasil *hash* yang tetap. Delapan word 32 bit yang didapat kemudian

digabungkan secara *little-endian* untuk membentuk *hash* akhir dengan panjang 256 bit.

Seluruh proses final dilakukan dengan penerapan kode program untuk menjaga konsistensi dan menghindari kesalahan perhitungan. Nilai *hash* yang dihasilkan kemudian ditunjukkan dalam format heksadesimal. Maka didapatkan hasil *hash* akhir dokumen rekam medis menggunakan algoritma BLAKE2s adalah:

fc3a509f5bd51fda5e5279f6b1010394f70d6ae4f1df63e14ac417

a7661bdb8b

Penggunaan nilai *hash* yang diperoleh ini yang akan digunakan sebagai masukan pada proses pembuatan tanda tangan digital. Nilai *hash* hasil algoritma BLAKE2s digunakan sebagai representasi data rekam medis yang kemudian ditandatangani secara digital. Selanjutnya akan dibahas pembuatan tanda tangan digital dengan berdasarkan nilai *hash*.

4.2.3 Pembuatan Tanda Tangan Digital

Setelah memperoleh nilai *hash* dari dokumen pada subbab 4.2.2, langkah berikutnya adalah membuat tanda tangan digital menggunakan algoritma DSA. Proses ini mengikuti alur pada Gambar 3.2 dan menggunakan parameter serta kunci yang telah dibangkitkan pada subbab 4.1 ($p = 59, q = 29, g = 3, x = 17, y = 19$).

Langkah awal dalam pembuatan tanda tangan adalah memilih bilangan acak k dengan syarat $1 < k < q - 1$. Nilai acak ini bersifat rahasia dan harus berbeda untuk setiap tanda tangan dokumen. Penggunaan nilai k yang identik pada dokumen yang berbeda dapat mengancam keamanan karena mempermudah pihak

lawan untuk mendapatkan kunci privat. Dalam ilustrasi manual ini ditentukan nilai $k = 13$.

Selanjutnya dihitung komponen tanda tangan r dan s. Hitung nilai r dengan menggunakan Persamaan 2.6:

$$r = (g^k \bmod p) \bmod q$$
$$= (3^{13} \bmod 59) \bmod 29$$
$$= ((3^8 \cdot 3^4 \cdot 3^1) \bmod 59) \bmod 29$$
$$= ((12 \bmod 59)(22 \bmod 59)(3 \bmod 59)) \bmod 59) \bmod 29$$
$$= \big((12 \cdot 22 \cdot 3) \bmod 59\big) \bmod 29$$
$$= \Big(\big((264 \bmod 59) \cdot 3\big) \bmod 59\Big) \bmod 29$$
$$= \Big(\big((28) \cdot 3\big) \bmod 59\Big) \bmod 29$$
$$= (84 \bmod 59) \bmod 29$$
$$= 25 \bmod 29 = 25$$

Perhitungan $3^{13} \bmod 59$ dikerjakan dengan metode *square* dan *multiply*:

$$3^1 = 3 \bmod 59 = 3$$
$$3^8 = (3^4)^2 = 22^2 = 484 \bmod 59 = 12$$
$$3^{16} = (3^8)^2 = 12^2 = 144 \bmod 59 = 26$$
$$3^{17} = 3^{16} \cdot 3^1 = 26 \cdot 3 = 78$$
$$= 78 \bmod 59 = 19$$

Sehingga didapatkan $r = 25 \bmod 29 = 25$. Sebelum menghitung komponen s diperlukan invers dari k, maka dihitung:

$$k^{-1} \bmod q = 13^{-1} \bmod 29$$
$$= (13 \cdot 9)^{-1} \bmod 29$$

$$= 117 \; mod \; 29 = 117 - (29 \cdot 4) = 117 - 116 = 1$$

Maka didapatkan nilai $k^{-1} = 9$. Lalu hitung komponen tanda tangan kedua (s)

$$s = k^{-1} \cdot (H(m) + x \cdot r) \; mod \; q$$
$$= 9 \cdot (H(m) + 17 \cdot 25) \; mod \; q$$
$$= 9 \cdot (7 + 425) mod \; q$$
$$= 9 \cdot (432) \; mod \; 29$$
$$= 9 \cdot 432 \; mod \; 29$$
$$= 9 \cdot 26$$
$$= 234 \; mod \; 29 = 2$$

Dari perhitungan di atas diperoleh tanda tangan digital $(r, s) = (25,2)$ dengan kunci publik $y = 19$. Pasangan nilai ini adalah representasi elektronik dari tanda tangan untuk dokumen yang relevan. Hasil pembuatan tanda tangan digital 25 dokumen dalam penelitian, disajikan pada Lampiran 1. Dalam penerapan nyata dengan kode program, nilai r dan s yang diperoleh berukuran besar (hingga 256 bit) sesuai dengan parameter DSA 2048 bit yang telah dibuat pada subbab 4.1.

4.3 Proses Verifikasi Tanda Tangan Digital

4.3.1 Verifikasi Tanda Tangan Digital

Dalam proses verifikasi, terdapat beberapa perhitungan, termasuk diperlukannya s^{-1}. Dalam langkah ini melibatkan Persamaan 2.7 yakni dengan perhitungan berikut:

$$w = s^{-1} \; mod \; q$$
$$= 2^{-1} \; mod \; 29$$
$$= (2 \cdot 15) \; mod \; 29 = 30 \; \; mod \; 29 = 1$$

Maka didapatkan nilai $w = 15$. Lalu hitung $u1$

$$u1 = (H(m) \cdot w)\ mod\ q$$

$$= (7 \cdot 15)\ mod\ 29$$

$$= 105\ mod\ 29 = 18$$

Didapat nilai $u1 = 18$, lalu hitung pula nilai $u2$ dengan rumus:

$$u2 = (r \cdot w)\ mod\ q$$

$$= (25 \cdot 15) mod\ 29$$

$$= 375\ mod\ 29 = 27$$

Lalu hitung v dengan memasukkan nilai yang sudah didapat dalam perhitungan sebelumnya.

$$v = \big((g^{u1} \cdot y^{u2}) mod\ p\big) mod\ q$$

Untuk memudahkan mari hitung nilai satu persatu untuk nilai g dan y

$$A = g^{u1}\ mod\ p$$

$$= 3^{18}\ mod\ 59 = 57$$

$$B = y^{u2}\ mod\ p$$

$$= 19^{27}\ mod\ 59 = 17$$

$$C = g^{u1} \cdot y^{u2}\ mod\ p$$

$$= A \cdot B\ mod\ p$$

$$= 57 \cdot 17\ mod\ 59 = 25$$

Nilai v dihitung kembali dengan menggunakan nilai C yang baru yang didapat melalui perhitungan $(g^{u1} \cdot y^{u2})\ mod\ p$.

$$v = (C\ mod\ p) mod\ q$$

$$v = 25\ mod\ 29 = 25$$

Didapatkan nilai $v = 25$ sama dengan nilai r dari tanda tangan yakni $r = 25$, yang diverifikasi pula dengan kode program pada Lampiran 6. Karena $v = r$, maka tanda tangan valid.

Penting untuk dimengerti bahwa perhitungan manual dalam subbab ini memanfaatkan representasi teks dari dokumen RME (Gambar 4.1) agar memudahkan memahami cara kerja algoritma BLAKE2s. Perhitungan ini hanya menggunakan teks yang terlihat, tanpa memperhatikan meta data, informasi dokumen, maupun struktur dari dokumen. Sementara itu, penerapan dengan kode program berfungsi pada file RME dalam format biner lengkap, sesuai dengan standar dalam sistem kriptografi. Dalam implementasi nyata, tanda tangan digital perlu melindungi seluruh berkas, termasuk meta data maupun informasi dokumen untuk memastikan integritas secara keseluruhan. Perbedaan ini tidak mengurangi keabsahan penelitian, karena keduanya memiliki tujuan yang berbeda yakni, manual untuk menggambarkan algoritma, sedangkan kode program untuk menerapkan verifikasi RME yang sebenarnya.

4.3.2 Pengujian Keaslian Dokumen Berdasarkan Perlakuan

Untuk memastikan keaslian tanda tangan digital, terdapat pengujian yang dilakukan pada 25 dokumen rekam medis asli. Setiap dokumen menerima 6 perlakuan yang berbeda, yakni P1 sampai P6 dengan perincian perlakuan sesuai dengan Tabel 4.1. Dengan penerapan 6 perlakuan ini, total dokumen yang diuji menjadi 150 dokumen modifikasi. Dokumen-dokumen ini kemudian dibandingkan dengan dokumen aslinya. Tujuan dari penggunaan enam perlakuan ini adalah untuk mengamati dampak setiap variasi perlakuan terhadap keaslian

tanda tangan digital pada dokumen. Berikut penjelasan untuk setiap perlakuan dalam penelitian:

1. Perlakuan P1 (Perubahan Satu Karakter atau Nilai Data)

   Perlakuan P1 dilakukan dengan mengedit satu atau lebih karakter dalam data rekam medis, seperti modifikasi huruf, angka, simbol, atau nilai tertentu. Pada dokumen 25-1, nama pasien “JacksOn” diubah menjadi “JacksOh”. Perubahan ini hanya mengganti satu karakter terakhir yang awalnya ‘n’ menjadi ‘h’, namun secara visual tetap mirip dengan nama asli. Didapatkan hasil pengujian pada seluruh dokumen yang diberikan perlakuan P1, dinyatakan tidak valid. Hal ini terjadi karena setiap perubahan kecil pada dokumen akan menyebabkan perubahan pada nilai *hash*nya juga. Sehingga tanda tangan digital yang dihasilkan dari dokumen asli tidak akan cocok lagi. Hasil ini menunjukkan bahwa algoritma tanda tangan digital memiliki responsivitas yang sangat baik ketika terdapat perubahan isi pada dokumen, sehingga dapat melindungi integritas data rekam medis.

2. Perlakuan P2 (Kesalahan Penulisan)

   Perlakuan P2 berfokus pada kesalahan penulisan seperti pertukaran huruf, kesalahan urutan karakter atau angka. Berbeda dengan P1 yang mengganti karakter, P2 mempertahankan karakter yang sama tetapi mengubah letaknya. Pada dokumen 25-1, istilah "Cancer" (penyakit kanker) diganti menjadi "Cacner". Kesalahan ini adalah typo yang sering terjadi di mana huruf 'n' dan 'c' bertukar tempat. Dari hasil penelitian didapatkan bahwa seluruh dokumen yang mendapatkan perlakuan P2 dianggap tidak sah. Tidak ada satu dokumen pun yang berhasil melewati verifikasi meskipun

hanya ada perubahan urutan. Ini terjadi karena kesalahan penulisan dapat mengubah tatanan *byte* dalam dokumen asli, sehingga juga akan menghasilkan nilai *hash* yang berbeda dengan dokumen aslinya. Hasil ini membuktikan bahwa sistem verifikasi tidak hanya peka terhadap perubahan karakter tetapi juga terhadap urutannya.

3. Perlakuan P3 (Perubahan Kapitalisasi Huruf)

   Perlakuan P3 dilakukan dengan mengubah huruf kapital, seperti dari huruf besar menjadi huruf kecil atau sebaliknya, tanpa mengubah susunan karakter. Pengujian perlakuan ini penting karena secara visual teks tetap terlihat sama, tetapi secara teknis representasi bytenya berbeda. Dalam dokumen 25-1, nama "Miller" disesuaikan menjadi "miller" dengan merubah huruf 'M' besar menjadi huruf kecil 'm'. Hasil pengujian menunjukkan bahwa semua dokumen dengan perlakuan P3 dinyatakan tidak valid. Sistem verifikasi berhasil mengidentifikasi perubahan huruf kapital meskipun arti kata tidak mengalami perubahan. Dalam kode ASCII, karakter kapital dan karakter kecil memiliki nilai yang berbeda dimana 'M' besar adalah 0x4D secara heksadesimal (77 desimal) dan 'm' kecil nilai heksadesimalnya adalah 0x6D (109 desimal). Perubahan satu bit pada representasi biner ini dapat merubah seluruh nilai *hash*.

4. Perlakuan P4 (Penghapusan Sebagian Data)

   Perlakuan P4 dilakukan dengan menghapus sebagian isi data di dokumen, seperti nama, angka, atau keterangan tertentu dimana setelah penghapusan karakter, akan di simpan lalu ditulis ulang. Perlakuan ini memberi percobaan situasi di mana pengguna mengubah dokumen, menyimpan,

kemudian mengembalikannya ke keadaan awal. Pada dokumen 25-1, teks "Blue Cross" diubah menjadi "Blue Cros", file disimpan, kemudian pengguna menulis ulang menjadi "Blue Cross". Secara visual, dokumen akhir tampak sama dengan dokumen yang asli. Berdasarkan hasil perlakuan, semua dokumen yang didapatkan dari perlakuan P4 dinyatakan tidak valid. Adanya tanda tangan digital adalah untuk mengautentikasi file fisik yang disimpan, bukan isi visual yang muncul di layar. Saat file disimpan setelah sebagian dihapus, beberapa hal bisa berubah, diantaranya adalah file metadata (*modification time, size*). Sehingga, walaupun teks yang ditampilkan serupa, nilai *hash* BLAKE2s dari file itu tidak sama dengan aslinya. Ini menunjukkan bahwa sistem verifikasi menjaga integritas file secara keseluruhan, bukan hanya isi yang terlihat secara visual pada dokumen.

5. Perlakuan P5 (Dokumen Tanpa Perubahan Isi)

   Perlakuan P5 dilakukan dengan mengirimkan dokumen pada platform komunikasi dengan mempertahankan isi dokumen sepenuhnya. Perlakuan ini memberi percobaan dimana dilakukan distribusi dokumen yang normal dalam sistem catatan medis. Pada dokumen, file dikirim lewat email, diunduh oleh penerima, penyalinan ulang, mengubah nama dokumen, melihat isi dokumen serta mengirim ulang dokumen tanpa mengubah konten. dan dibuka tanpa adanya perubahan. Hasil perlakuan didapatkan bahwa semua dokumen dari perlakuan P5 dinyatakan valid. Hasil ini menunjukkan beberapa hal yang signifikan yakni bahwa konsistensi fungsi *hash* BLAKE2s (nilai *hash* untuk konten yang serupa selalu sama). Ini

menunjukkan bahwa sistem verifikasi bergantung hanya pada isi file, bukan pada atribut luarannya. Perlakuan ini menunjukkan bahwa sistem tanda tangan digital yang diterapkan dapat diandalkan dan konsisten dalam membedakan dokumen yang telah dimodifikasi dengan dokumen asli.

6. Perlakuan P6 (Edit, Hapus, dan Simpan Ulang)

   Perlakuan P6 dilaksanakan untuk menguji apakah proses editing yang tidak disertai penyimpanan berpengaruh terhadap keabsahan tanda tangan digital. Pengujian dilakukan pada 25 dokumen RME dengan skenario, dokumen diakses, dilakukan pengeditan (menyisipkan/menghapus karakter), kemudian perubahan disimpan sebelum file ditutup. Skenario ini mencerminkan situasi sebenarnya di mana pengguna memodifikasi isi dokumen dan menyimpan *update*-nya. Hasil uji menunjukkan bahwa semua 25 dokumen yang diberi perlakuan P6 dinyatakan tidak sah. Proses penyimpanan yang dilakukan mengakibatkan perubahan pada file fisik di disk, sehingga nilai hash BLAKE2s yang dihasilkan tidak sama dengan dokumen aslinya. Sebagai hasilnya, otentikasi tanda tangan digital gagal.

   Sebagai ilustrasi, di dokumen 25-9, setelah proses pengeditan dan penyimpanan, hash berganti dari “ecd3b2b7f8dc66d993990b54f589e945.. ” menjadi “37a3232ac8755069a8371f065af...” dan status verifikasi berubah menjadi tidak sah. Perlakuan ini menunjukkan bahwa verifikasi tanda tangan digital berfungsi pada file fisik yang tersimpan di media penyimpanan, bukan pada representasi grafis yang muncul di layar. Hal ini krusial dalam sistem rekam medis elektronik, di mana keaslian file yang disimpan memastikan integritas data. Setelah file disimpan setelah

dilakukan modifikasi, integritasnya tidak dapat terjaga dan tanda tangan digital menjadi tidak valid

Hasil pengujian seluruh perlakuan menunjukkan bahwa kevalidan tanda tangan digital sangat dipengaruhi oleh keaslian isi dokumen. Setiap perubahan sekecil apa pun pada berkas, termasuk kesalahan ketik, pengubahan huruf, atau penghapusan informasi, mengakibatkan dokumen dianggap tidak valid dengan persentase 100%. Sebaliknya, dokumen yang tidak mengalami modifikasi isi masih dapat diverifikasi dengan akurat. Maka dapat dikatakan bahwa algoritma DSA yang menggunakan fungsi *hash* BLAKE2s memiliki level sensitivitas dan keandalan tinggi dalam menjaga keutuhan dokumen rekam medis.

4.3.3 Perubahan Nilai *Hash* pada Dokumen Uji

Untuk memberikan gambaran yang lebih jelas mengenai pengaruh perubahan isi dokumen terhadap nilai *hash* dan keaslian tanda tangan digital, pada bagian ini ditampilkan contoh perbandingan nilai *hash* antara satu dokumen asli dengan dokumen perlakuan. Contoh hasil perbandingan *hash* pada Tabel 4.3, bertujuan sebagai ilustrasi hasil pengujian antara dokumen asli dengan dokumen perlakuan.

Berdasarkan Tabel 4.3 perbandingan nilai *hash*, dapat dilihat bahwa dokumen asli dengan nama “1-25-9” memberikan nilai *hash* tertentu saat mengimplementasikan fungsi *hash* BLAKE2s. Nilai *hash* ini kemudian yang digunakan dalam proses penandatanganan dan verifikasi dokumen dengan menggunakan algoritma *Digital Signature Algorithm* atau DSA.

Pada dokumen hasil perlakuan P1, P2, P3, P4, dan P6, nilai *hash* yang dihasilkan berbeda dari nilai *hash* dokumen asli. Perbedaan ini terjadi sekalipun perubahan yang dilakukan pada dokumen relatif kecil, seperti perubahan karakter,

kesalahan penulisan, perubahan kapitalisasi huruf, serta penghapusan sebagian data dimana jumlah karakter yang dimodifikasi tidak terbatas hanya pada satu karakter. Hal ini menunjukkan bahwa fungsi *hash* BLAKE2s memiliki tingkat sensitivitas yang tinggi terhadap perubahan sekecil apa pun pada isi dokumen. Perbedaan nilai *hash* tersebut membuat proses verifikasi tanda tangan digital menggunakan algoritma DSA tidak berjalan dengan baik. Akibatnya, dokumen yang sudah diproses dengan metode P1, P2, P3, P4, dan P6 dinyatakan tidak valid. Ini berarti bahwa integritas dokumen telah berubah setelah proses penandatanganan selesai.

Tabel 4.3 Perbandingan Hasil *Hash* pada Dokumen Uji

Dokumen Asli	**Hasil *Hash***	**Dokumen Perlakuan**	**Hasil *Hash***	**Status**
1-25-9	ecd3b2b7f8dc66d993990b54f589e9	P1	37a3232ac8755069a8371f065af	Tidak Valid
		P2	73ee1e4e759fd182e8029523744	Tidak Valid
		P3	e73a2c1a218d5756c09879d1a42	Tidak Valid
		P4	9fac3b876160756aa61210f3a62	Tidak Valid
		P5	ecd3b2b7f8dc66d993990b54f58	Valid
		P6	1663e980bf427f2b3ba5d359ce7d	Tidak Valid

Berbeda dengan dokumen perlakuan yang lain, pada dokumen hasil perlakuan P5, nilai *hash* yang dihasilkan identik dengan nilai *hash* dokumen asli. Hal ini dikarenakan, pada perlakuan P5 tidak terdapat perubahan isi dokumen. Dengan demikian, proses verifikasi tanda tangan digital dapat dilakukan dan

menghasilkan status valid. Hasil ini menunjukkan bahwa sistem tanda tangan digital yang diterapkan mampu membedakan secara jelas antara dokumen yang mengalami perubahan dan dokumen yang tetap mempertahankan keasliannya.

4.4 Integrasi Islam dalam Verifikasi Rekam Medis Elektronik

4.4.1 Prinsip Amanah dan Integritas Data

Dalam Islam terdapat konsep yang disebut amanah atau kepercayaan. Ini adalah suatu tanggung jawab yang penting dan harus dihormati. Seperti dalam Al-Qur'an surat An-Nisa' ayat 58, bahwa menjaga dan menghormati kepercayaan yang telah diberikan adalah sebuah keharusan. Dalam rekam medis, setiap data yang ada harus akurat dan tidak boleh diubah maupun dimanipulasi. Maka dari itu, bentuk kepercayaan atas data rekam medis ini sangat penting. Oleh karena itu, sistem RME menggunakan tanda tangan digital yang dihasilkan dari kombinasi DSA dan BLAKE2s untuk memastikan bahwa dokumen medis tidak dimanipulasi atau dipalsukan. Dengan demikian, nilai kepercayaan dalam bentuk teknologi dapat diwujudkan dalam praktik.

Dalam ajaran Islam juga menekankan bahwa setiap tindakan yang kita lakukan harus kita pertanggungjawabkan. Tanda tangan digital memiliki ciri khas yang disebut *non-repudiation*, artinya orang yang menandatangani dokumen tidak bisa mengingkari bahwa mereka telah menandatangani dokumen tersebut. Hal ini selaras dengan ajaran Rasulullah SAW yang mengajarkan untuk menunaikan amanah dan tidak berkhianat, sebagaimana yang diriwayatkan oleh Abu Dawud No. 3535. Dalam dunia medis, *non-repudiation* menjamin bahwa dokter atau tenaga medis bertanggung jawab atas diagnosis dan tindakan yang dilakukan.

Sistem verifikasi yang memanfaatkan kunci publik dapat menjamin keaslian tanda tangan, sehingga memperkuat akuntabilitas profesional dalam bidang kesehatan.

Hasil penelitian menunjukkan bahwa sistem dapat mengenali perubahan sekecil apa pun pada dokumen, baik itu penggantian karakter, pertukaran huruf, maupun modifikasi lainnya. Kemampuan ini memenuhi prinsip kejujuran untuk memenuhi amanah dalam Islam, di mana setiap pencatatan data medis harus akurat dan dapat dipertanggungjawabkan.

4.4.2 Prinsip *Dar'ul Mafasid*

Prinsip *dar'ul mafasid* atau mencegah kerusakan dalam *maqasid syari'ah* menekankan bahwa mencegah bahaya lebih diutamakan dibandingkan dengan mencari keuntungan atau kebaikan. Dalam lingkup RME, data kesehatan adalah informasi penting yang langsung memengaruhi keselamatan pasien. Jenis kerusakan dalam RME bisa berupa bocornya data, pemalsuan resep, atau kesalahan diagnosis akibat data yang tidak bisa dipastikan keasliannya. Kerusakan ini dapat menyebabkan kesalahan dalam pengambilan keputusan medis.

Maka dari itu, implementasi tanda tangan digital menggunakan DSA dan BLAKE2s berfungsi sebagai tindakan pencegahan untuk menghindari kerusakan tersebut. Penggunaan tanda tangan digital dengan algoritma DSA dan fungsi *hash* BLAKE2s berfungsi sebagai langkah pencegahan untuk mengurangi kemungkinan terjadinya kerusakan tersebut. Tanda tangan digital memastikan keaslian (*authenticity*) dan integritas (*integrity*) dokumen, sehingga setiap perubahan sekecil apapun pada dokumen medis dapat teridentifikasi dengan jelas.

Dengan sistem ini, tenaga medis dapat memastikan bahwa data yang digunakan untuk diagnosis berasal dari sumber yang valid dan belum dimanipulasi.

Hasil penelitian menunjukkan bahwa sistem dapat membedakan antara dokumen asli dan yang telah dimodifikasi, baik melalui modifikasi isi maupun perlakuan khusus pada dokumen. Kemampuan ini secara langsung mendukung prisip *dar'ul mafasid* karena menghindari kesalahan klinis yang dapat membahayakan pasien. Oleh karena itu, implementasi tanda tangan digital dalam RME tidak hanya memiliki nilai teknis, tetapi juga sesuai dengan tujuan syariat dalam melindungi keselamatan jiwa manusia (*hifzh an-nafs*).

4.4.3 Perlindungan Privasi dan Martabat Pasien

Islam sangat menghormati privasi dan martabat individu sebagai nilai yang hal yang sangat dihargai, seperti yang dinyatakan dalam Q.S. Al-Hujurat ayat 12 yang melarang perilaku untuk melihat dan mempermalukan orang lain. Informasi kesehatan adalah data yang sangat pribadi dan sensitif. Sehingga harus dikelola dengan aman serta bertanggung jawab. Pelanggaran privasi data kesehatan tidak hanya memiliki konsekuensi hukum dan sosial, tetapi juga dapat berdampak negatif pada pasien secara psikologis dan moral.

Pemakaian tanda tangan digital dalam sistem RME menjamin bahwa hanya pihak yang berwenang yang dalam hal ini tenaga medis yang dapat mengakses, menandatangani, memverifikasi dokumen medis. Implementasi sistem ini dapat menghindari penyalahgunaan data dan melindungi kerahasiaan (*confidentiality*) informasi pasien. Selain itu, sistem verifikasi yang diterapkan memastikan bahwa dokumen yang diterima tetap asli dan tidak mengalami perubahan selama proses distribusi. Hal ini penting untuk mempertahankan kepercayaan antara pasien dan

tenaga medis, serta melindungi martabat pasien dari potensi pemalsuan atau kebocoran data yang bisa mengakibatkan efek negatif lainnya. Efek negatif tersebut dapat berupa penyalahgunaan identitas, gangguan psikologi, kesalahan dalam pengambilan keputusan medis, serta munculnya permasalahan hukum bagi pasien maupun institusi layanan kesehatan. Dengan demikian, penggunaan tanda tangan digital dalam RME tidak hanya memperkuat aspek keamanan data, tetapi juga sejalan dengan *maqasid sya'riah* dalam melindungi kehormatan dan martabat manusia (*hifzh al-ird*)

BAB V

PENUTUP

5.1 Kesimpulan

Berdasarkan hasil implementasi verifikasi dan pengujian tanda tangan digital dengan algoritma Digital Signature Algorithm (DSA) dan fungsi *hash* BLAKE2s pada dokumen Rekam Medis Elektronik (RME), diperoleh kesimpulan sebagai berikut:

1. Sistem verifikasi yang dibuat dapat menghasilkan tanda tangan digital yang khas untuk setiap dokumen RME melalui gabungan DSA dan BLAKE2s. Fungsi *hash* BLAKE2s terbukti sensitif terhadap perubahan data, di mana perubahan sekecil satu karakter dalam dokumen menghasilkan nilai *hash* yang berbeda secara signifikan (*avalanche effect*).
2. Berdasarkan skenario pengujian yang dilakukan pada 25 dokumen asli dan 150 dokumen hasil modifikasi, proses verifikasi mencapai tingkat akurasi 100% dalam membedakan dokumen asli dan dokumen yang telah dimodifikasi. Uji coba terhadap 25 dokumen dengan enam jenis perlakuan (P1-P6) menunjukkan bahwa sistem dapat mengidentifikasi semua jenis modifikasi, termasuk perubahan satu karakter, kesalahan ketik, perubahan huruf kapital, penghapusan informasi, serta proses pengeditan dan penyimpanan ulang. Dengan demikian, prinsip integritas dan *non-repudiation* dalam perlindungan data rekam medis dapat terpenuhi.

Penggabungan DSA dan BLAKE2s memberikan keseimbangan yang baik antara keamanan dan efisiensi dalam sistem verifikasi RME. Secara teknis, sistem ini dapat memastikan keautentikan dokumen dan ketahanan terhadap perubahan.

Dalam hal nilai, pelaksanaannya selaras dengan prinsip Islam yang mengutamakan pemeliharaan amanah, pencegahan kerugian (*dar'ul mafasid*), serta perlindungan terhadap privasi dan harga diri pasien

5.2 Saran

Berdasarkan hasil dari penelitian, penulis menyampaikan rekomendasi sebagai berikut:

1. Melakukan perhitungan manual fungsi *hash* BLAKE2s pada tingkat *bitstream* (representasi biner murni) untuk mendapatkan ilustrasi yang lebih akurat sesuai dengan implementasi kriptografi sebenarnya.
2. Menggunakan dokumen yang lebih memiliki variansi ukuran atau jumlah *byte* yang berbeda, untuk analisis mendalam pada algoritma.
3. Menguji sistem dalam jaringan nyata untuk menilai kinerjanya saat menghadapi beban pengguna yang tinggi.
4. Merekomendasikan penerapan sistem tanda tangan digital yang menggunakan DSA dan BLAKE2s untuk memperkuat keamanan serta keaslian dokumen RME.
5. Memanfaatkan kumpulan data yang lebih bervariasi dan realistis untuk pengujian, termasuk dokumen dengan ukuran dan format yang beragam.

Dengan demikian, penelitian ini diharapkan menjadi acuan dan motivasi untuk pengembangan sistem keamanan informasi kesehatan yang lebih kuat dan sejalan dengan kemajuan teknologi digital

DAFTAR PUSTAKA

Algazy, K., Sakan, K., Khompysh, A., & Dyusenbayev, D. (2024). Development of a New Post-Quantum Digital Signature Algorithm: Syrga-1. *Computers*, *13*(1), 26. https://doi.org/10.3390/computers13010026

Amaludin, L., & Rahmatulloh, A. (2024). Penerapan ECDSA dan BLAKE2B Untuk Membentuk Tanda Tangan Digital Sebagai Autentikasi Dokumen. *Jurnal Informatika dan Multimedia*, *16*(2), 20–26. https://doi.org/10.33795/jtim.v16i2.6599

Azza, V. M., Hartana, H., & Tio Rae, G. nyoman. (2024). Application of Personal data protection on electronic signatures in Indonesia. *Jurnal Indonesia Sosial Teknologi*, *5*(5), 2430–2439. https://doi.org/10.59141/jist.v5i5.1102

Harahap, M. I., Suherman, S., & Sembiring, R. W. (2023). Three Pass Protocol for Key Security Using Affine Cipher Algortima and Exclusive-or (Xor) Combination. *Sinkron*, *8*(4), 2602–2614. https://doi.org/10.33395/sinkron.v8i4.13051

Heidari, F., Pashabadi, D., Fathi, M., & Razaghi, M. (2025). Optical XOR logic gate design in two dimensional photonic crystal using ANN and PSO. *Scientific Reports*, *15*(1), 26471. https://doi.org/10.1038/s41598-025-12146-9

Irawan, Wahyu Henky, Hijriyah, Nurul and Habibi, Azwar Riza. (2017). *Pengantar teori bilangan*. UIN-Maliki Press, Malang. https://repository.uin-malang.ac.id/1291/

Muda, Z., Omar, N., Pengiran Haji Hashim, P. H. N., Haji Ramlee, H. N., & Dato Seri Setia Haji Rajid, Z. Z. (2023). The Authoritative Role of Maqasid Shariah As A Basis for Determining Hukm When Dealing with Contemporary Issues. *International Journal of Academic Research in Business and Social Sciences*, *13*(4), Pages 711-720. https://doi.org/10.6007/IJARBSS/v13-i4/16685

Mufidah, N. F., & Nuha, H. H. (2024). Performance and Security Analysis of Lightweight Hash Functions in IoT. *Jurnal Informatika: Jurnal Pengembangan IT*, *9*(3), 264–270. https://doi.org/10.30591/jpit.v9i3.7633

Novianti & Handar Subhandi Bakhtiar. (2024). Implementation of Electronic Medical Record System in Indonesia Viewed from the Perspective of Legal Certainty. *International Journal of Engineering Business and Social Science*, *2*(04), 1114–1122. https://doi.org/10.58451/ijebss.v2i04.145

Nuraeni, F., Amrulloh, M. F., Mulyani, A., & Kurniadi, D. (2025). Implementasi Superenkripsi Dsa dan Aes 128 Bit Dalam Pengamanan File Surat Digital.

Jurnal Algoritma, *22*(1), 601–613. https://doi.org/10.33364/algoritma/v.22-1.1832

Stallings, W. (2014). *Cryptography and network security: Principles and practice* (Sixth edition). Pearson.

Torab-Miandoab, A., Basiri, M., Dabbagh-Moghaddam, A., & Gholamhosseini, L. (2025). Electronic health record in military healthcare systems: A systematic review. *PLOS ONE*, *20*(2), e0313641. https://doi.org/10.1371/journal.pone.0313641

Uslu, A., & Stausberg, J. (2021). Value of the Electronic Medical Record for Hospital Care: Update From the Literature. *Journal of Medical Internet Research*, *23*(12), e26323. https://doi.org/10.2196/26323

Wening Werdi Mukti, G., & Setiawan, H. (2020). Designing and Building Secure Electronic Medical Record Application by Applying AES-256 and RSA Digital Signature. *IOP Conference Series: Materials Science and Engineering*, *852*(1), 012148. https://doi.org/10.1088/1757-899X/852/1/012148

LAMPIRAN

Lampiran 1 Hasil Tanda Tangan Digital dan *Hash*

No.	Nama File	*Hash*	r	s
1	25-1 (1).pdf	4a83cc0f3d783ea3a7 d46a98b08bd5e2a597 1966d0783cee3b9e9b a075063b7a	44172297386286656653 63062066654532478624 17397359906285437309 17200600036922904	62135113646607925168 19388327926921675238 44473377721475523728 83396409702238509
2	25-2 (1).pdf	677674d4b3436e302 659bc9192b7e3637d 38e312cec3780dc998 17df104fca7a	19512932911539183337 26113546054993042686 58783829803866498298 99129590099463433	29982660084621832973 40589686404764216222 46154409213366158015 97746551045599341
3	25-3 (1).pdf	d9931c87d53d71843 b6a1fe5ca05cca2f776 e4a1d1f9653ce88172 a7505301c5	49729184407787506420 75423881815846398720 97254617202489412307 18339843607265194	32596103288112284381 71688770315367506453 47702242810874853566 42356826978487988
4	25-4 (1).pdf	406e5858c400cd5976 7543552b746c3e48be a969544f6cf2b454f8 02dc220e30	70018441951125101472 88748822299870240810 92711805656913858082 78697868302584441	26561656678416997521 06006114493928380909 51433549861131833562 45585577817529480
5	25-5 (1).pdf	9757d3afe183ceb24c bb30b45685e905dfa3 6c3e745ab1d55c90b7 1fefbdae80	61973554927005712921 16502327768999823865 97083675673027006801 04309450575202185	57887862192571563859 65331972917897806905 57799802868222891647 99712427465329783
6	25-6 (1).pdf	2db0d22ec8cc43693b 89ed61f90729c38ac0 e9dd12a4b521d0cff6 3e8629e92d	65377720628369556489 37659683467248266660 86580701207642282767 86915315337962059	41761965143135956346 16776143147822039107 32459994110484243352 17353669668539925
7	25-7 (1).pdf	24601fc558640111ab 1761a496ce32b7ab6b 7e7a28790e15488f56 3719fa5b71	78527812535355742235 00526267069391600562 51744615191455861150 68198354836823558	82953084084780908342 70893714350790026381 22319961997627106386 62538655454900184
8	25-8 (1).pdf	fba99163ba2e542d74 995f2162621739d6b5 44ec1d3b50a77e9887 5812fab4b4	11502524186040314723 18815573490524286669 67356556147808696713 84086396487849443	21591916772519646545 46341309457797226830 75373569377287766380 84518453172639407
9	25-9 (1).pdf	ecd3b2b7f8dc66d993 990b54f589e945eacd 027ef0c232048be66a c1dcc8d7d7	62981857494753585043 62946204452284129886 37217509313356982580 0265319511458051	64721212459749083129 09613623549378326940 86389525043816129550 12559121009648061
10	25-10 (1).pdf	cfc5ac968b4ea49385 22aeb938e14ff49748 27330360eb5f65a7c4 bfda4c847f	11908637048899636914 35541103031764149385 67026133541589364172 87574699217801512	95085748415040552636 84007209189616217615 39680084311984052663 4417414428803909
11	25-11 (1).pdf	bee7b789edfa11ba28 69e4b79869ded1093 def70c9a3ee027755f b37e4d1c378	11117263953637653370 40835329960211350682 62182754007096948021 90739618831443466	61635725263273033173 44086927557518565653 43096099547560403351 10631521929864443
12	25-12 (1).pdf	87b59d69eb15e96a4e 662ea4b61b02246f4e ade75fa794d371dbae d107655d07	13463344219731205819 84258460874006925104 44068288795646876573 6399734228965200	12125713023095612946 85545821128348419576 90335913152269312397 98435767577560091
13	25-13 (1).pdf	d1b441d358f0e39915 4a803292214963171	49358378201505813875 94392870708577614276	13159424069600179770 03005940008911489427

		ac8bcdfe97978d77c5 725a7472396	73077734571596234009 60085346526252084	60026858849920892682 9734197582162459
14	25-14 (1).pdf	f21b2685ace8cbf9ce1 7a68f3d6f4fbf0edc4c e198d75ee9302e0096 2c118c3d	60329038171667307095 04174457058747985959 80789576574704458869 87759979553793943	85837301372263146191 13098188206503712473 87059293430425842861 63407869980299780
15	25-15 (1).pdf	39149b0e9d4e488c55 aa5eefbe0c88f07438e 3bd8782820cb3626d b3b9db7b42	31025345557410345545 05833645057561444991 65849725357100878249 72882932880775434	21306058667439851468 51916568510433569045 44023432316554150826 1427933051032628
16	25-16 (1).pdf	0e65d0f670ccc3189d fa2c90739caa4e282b 16dba3f0dd64d7baad 8f9c95d752	85484900757480632051 43765515152366573334 49888227933831247292 82294284711477637	31862950458384139270 79813930116838191319 79888296542165042102 12519754734179497
17	25-17 (1).pdf	703ad6e7e8c5e7543e bcf12b1ae8eab06423 056148ebdd0b4a106 bf92f9c564f	69948819326789555351 48165278852414129341 82004803505978165331 93700555660072730	81767082287481154827 93798409476650434559 98729683330717109864 52866265828744712
18	25-18 (1).pdf	23f94678b79ccf263f 5dbb27f1d36dc90753 587b436c07107a2dc4 725e6e0221	22266346710848070202 47183874728926610482 32498532555031156209 26651928051755353	58073381622879432571 81774816546722867755 53337354385028915818 4350196097687169
19	25-19 (1).pdf	b1bde99bc353d7852f 0bdabef4dd6a2f16e2 cf9a3abeba0ae19c62a c9572f387	77793724863196280269 89150698061302656462 76176677528366726415 03136872935430442	10432987004485184167 19150557531698178368 68216138417166344279 4103581458031951 2
20	25-20 (1).pdf	d91c204d902b4e1b6f 98bd7fc295e7091cca d4a01bf3d72ad8822d 874855f95d	43030456621518766714 33345955707081895681 37479654072718972652 3731088070835740	26016204374021471902 39740446002077519053 63494199392625796553 87840481841855135
21	25-21 (1).pdf	bf5da6d5e771b3bc29 098f1d7815828742ff 36a5b8c6c78958732b 4dcd2eb486	62298052087595544212 48030495177151638663 57539586127329626886 59478060396781164	60382283031449899601 85666942448456272196 10743534411617227307 74393329187729981
22	25-22 (1).pdf	78acf2f5e0923ede9e3 09b6f0b468372df165 e8c2100953c41d2b63 63198d952	59788971855300226668 71998852333259681040 98949250051498626985 26136224957414726	76913854308677551801 71456045150102360032 10157716618170406364 30246206342429842
23	25-23 (1).pdf	26e3b3c4ac92906d10 2bae4d8a909824e0bc 5ddd10d421281a466 0ff9631d55c	61936492246116594683 19154795412998024272 77897486627207412995 52078079533692195	37844236660685557830 79532175953647588851 93294906166973868068 94324638884041089
24	25-24 (1).pdf	416e24b494acca86ac 1a8bc40fbfef994522e c55df141cbd02b84a4 086e59de7	55310204690329290209 39054871138706116377 93661673668892142173 89435666351045504	31506773033494309957 81062399461220088121 12114386136693712221 04717200102003037
25	25-25 (1).pdf	98a573d1b76bff86b7 b4832dcc2412b6588 76787fdc9f3320befaf e257aa53be	28252646658551473905 54506309723831245970 26741108682645461322 26653789800708958	65453903048654285561 64317332939207347253 96854809299206087180 88409665029466998

Lampiran 2 Hasil Tabel ASCII Data "1-25-9.pdf"

No	Karakter	ASCII Decimal	ASCII Hex	No.	Karakter	ASCII Decimal	ASCII Hex
1.	R	82	0x52	6.	[spasi]	32	0x20
2.	U	85	0x55	7.	S	83	0x53
3.	M	77	0x4D	8.	A	65	0x41
4.	A	65	0x41	9.	K	75	0x4B
5.	H	72	0x48	10.	I	73	0x49
11.	T	84	0x54	22.	A	65	0x41
12.	[spasi]	32	0x20	23.	Y	89	0x59
13.	P	80	0x50	24.	S	83	0x53
14.	R	82	0x52	25.	\n	10	0x0A
15.	O	79	0x4F	26.	D	68	0x44
16.	O	79	0x4F	27.	A	65	0x41
17.	F	70	0x46	28.	t	116	0x74
18.	[spasi]	32	0x20	29.	a	97	0x61
19.	D	68	0x44	30.	[spasi]	32	0x20
20.	-	45	0x2D	31.	R	82	0x52
21.	D	68	0x44	32.	e	101	0x65

Lampiran 3 Kode Program *Output* Ronde Pertama

```
import struct
# ===== ROTR =====
def rotr(x, n):
    return ((x >> n) | (x << (32 - n))) & 0xFFFFFFFF
# ===== FUNGSI G =====
def G(v, a, b, c, d, x, y):
    v[a] = (v[a] + v[b] + x) & 0xFFFFFFFF
    v[d] = rotr(v[d] ^ v[a], 16)
    v[c] = (v[c] + v[d]) & 0xFFFFFFFF
    v[b] = rotr(v[b] ^ v[c], 12)
    v[a] = (v[a] + v[b] + y) & 0xFFFFFFFF
    v[d] = rotr(v[d] ^ v[a], 8)
    v[c] = (v[c] + v[d]) & 0xFFFFFFFF
    v[b] = rotr(v[b] ^ v[c], 7)
# ===== IV BLAKE2s =====
IV = [
    0x6A09E667, 0xBB67AE85, 0x3C6EF372, 0xA54FF53A,
    0x510E527F, 0x9B05688C, 0x1F83D9AB, 0x5BE0CD19
]
# ===== STATE AWAL =====
h = IV.copy()
v = h + IV.copy()
# ===== PESAN SESUAI MANUAL =====
m = [0] * 16
m[0] = 0x414d5552
m[1] = 0x41532048
# ===== SATU RONDE =====
G(v, 0, 4, 8, 12, m[0], m[1])
# ===== OUTPUT 4 VARIABEL SAJA =====
print("Setelah G pertama:")
print("v0 =", hex(v[0]))
print("v4 =", hex(v[4]))
print("v8 =", hex(v[8]))
print("v12 =", hex(v[12]))
```

Lampiran 4 Kode Program *Output Hash* Final

```
import hashlib
from Crypto.PublicKey import DSA
from Crypto.Signature import DSS
from Crypto.Hash import SHA256 as PyCryptoSHA256 # Use SHA256 for DSS
compatibility
# Hash dokumen asli (using standard hashlib for displaying the hex
digest)
hash_asli_obj = hashlib.sha256(doc_asli) # Changed to SHA256
hash_asli = hash_asli_obj.hexdigest()
# Buat kunci
private_key = DSA.generate(2048)
public_key = private_key.publickey()
# Buat signature
# Create a PyCryptodome hash object for signing (using SHA256)
h = PyCryptoSHA256.new(data=doc_asli)
signer = DSS.new(private_key, 'fips-186-3')
signature = signer.sign(h) # Pass the PyCryptodome SHA256 hash object
print("Hash Asli:", hash_asli)
print("Signature dibuat.")
def verify_hash_mod():
    hash_hex                                                    =
"fc3a509f5bd51fda5e5279f6b1010394f70d6ae4f1df63e14ac417a7661bdb8b"
```

Lampiran 5 Hasil *Hash* Akhir ke H(m)

```
    q = 29
    print("Verifikasi Hash mod 29")
    print("=" * 50)
    # Metode 1: Konversi ke integer lalu mod
    hash_int = int(hash_hex, 16)
    result1 = hash_int % q
    print(f"Metode 1 (int lalu mod): {result1}")
    # Metode 2: Horner's method (langkah demi langkah)
    value = 0
    hex_digits = "0123456789abcdef"
    print("\nMetode 2 (Horner's method):")
    for i, char in enumerate(hash_hex.lower()):
        digit_value = hex_digits.index(char)
        old_value = value
        value = (value * 16 + digit_value) % q
        if i < 5 or i >= len(hash_hex)-5:  # Tampilkan awal dan akhir
saja
            print(f"Digit {i+1:2d}: '{char}' = {digit_value:2d}")
            print(f"  ({old_value} × 16 + {digit_value}) mod {q}")
            print(f"  = ({old_value*16} + {digit_value}) mod {q}")
            print(f"  = {old_value*16+digit_value} mod {q}")
            print(f"  = {value}")
    print(f"\nHasil akhir: {value}")
    # Verifikasi kedua metode sama
    if result1 == value:
        print(f"\n✓ VERIFIKASI BERHASIL: Hash mod 29 = {value}")
    else:
        print(f"\n✗ VERIFIKASI GAGAL: {result1} ≠ {value}")
    return value
# Jalankan verifikasi
result = verify_hash_mod()
```

Lampiran 6 Verifikasi Tanda Tangan

```
p = 59
q = 29
g = 3
y = 19
# Nilai hash pesan (hasil reduksi hash BLAKE2s mod q)
H_m = 7
# Tanda tangan digital (r, s)
r = 25
s = 2
def mod_inverse(a, m):
    """Mencari inverse modular a mod m"""
    for x in range(1, m):
        if (a * x) % m == 1:
            return x
    return None
def dsa_verify(p, q, g, y, H_m, r, s):
    # Cek validitas r dan s
    if not (0 < r < q and 0 < s < q):
        return False
    # Hitung inverse s
    w = mod_inverse(s, q)
    if w is None:
        return False
    # Hitung u1 dan u2
    u1 = (H_m * w) % q
    u2 = (r * w) % q
    # Hitung nilai v
    v = ((pow(g, u1, p) * pow(y, u2, p)) % p) % q
    # Verifikasi
    return v == r
# Jalankan verifikasi
is_valid = dsa_verify(p, q, g, y, H_m, r, s)
print("Hasil Verifikasi Tanda Tangan Digital:")
print("Valid" if is_valid else "Tidak Valid")
```

Lampiran 7 Hasil Perlakuan pada Dokumen Asli

```
import os
import hashlib
import time
import json
import base64
from datetime import datetime, timezone, timedelta
from google.colab import files
from cryptography.hazmat.primitives.asymmetric import dsa
from cryptography.hazmat.primitives import hashes
from      cryptography.hazmat.primitives.asymmetric.utils      import
decode_dss_signature
from cryptography.hazmat.backends import default_backend
from cryptography.hazmat.primitives import serialization
print("="*100)
print(" DIGITAL SIGNATURE VERIFICATION - COMPLETE VERSION")
print("="*100)
os.makedirs("complete_docs", exist_ok=True)
os.makedirs("complete_results", exist_ok=True)
print("\n UPLOAD FILE:")
print("1. Upload file ASLI (tandatangan dibuat dari file ini)")
print("2. Upload file MODIFIKASI (untuk perbandingan)")
print("3. Bisa upload multiple file untuk perbandingan")
print("\n Tekan 'Choose Files' dan pilih file Anda...")
uploaded = files.upload()
if not uploaded:
    print(" Tidak ada file yang diupload!")
    exit()
print("\n MENYIMPAN FILE...")
for filename, content in uploaded.items():
    filepath = os.path.join("complete_docs", filename)
    with open(filepath, "wb") as f:
        f.write(content)
    print(f"✓ {filename} ({len(content):,} bytes)")
file_list = sorted(os.listdir("complete_docs"))
print(f"\n TOTAL FILE DIUPLOAD: {len(file_list)}")

for i, filename in enumerate(file_list, 1):
    filepath = os.path.join("complete_docs", filename)
    size = os.path.getsize(filepath)
    print(f"{i:2d}. {filename:40} ({size:8,} bytes)")
# Pilih file asli (default: file pertama)
original_index = 0
original_file = file_list[original_index]
original_path = os.path.join("complete_docs", original_file)
modified_files  =  [f  for  i,  f  in  enumerate(file_list)  if  i  !=
original_index]
print(f"\n FILE ASLI DIPILIH: {original_file}")
print(f"   Key Type  : {type(private_key).__name__}")
print("\n" + "="*100)
print("  SIGNING FILE ASLI")
print("="*100)
print("  SIGNING FILE ASLI")
print("="*100)
signing_start_time = datetime.now(timezone.utc)
print(f" FILE UNTUK PERBANDINGAN: {len(modified_files)} file")
```

```
print("\n" + "="*100)
print(" ANALISIS FILE ASLI")
print("="*100)
with open(original_path, "rb") as f:
    original_data = f.read()
with open(original_path, "rb") as f:
    original_data = f.read()
# Hitung berbagai hash untuk file asli
print(f"\n FILE: {original_file}")
print(f" UKURAN: {len(original_data):,} bytes")
hash_blake2s = hashlib.blake2s(original_data).hexdigest()
hash_sha256 = hashlib.sha256(original_data).hexdigest()
hash_md5 = hashlib.md5(original_data).hexdigest()
print(f"\n HASH FILE ASLI:")
print(f"   BLAKE2s  : {hash_blake2s}")
print(f"   SHA-256  : {hash_sha256}")
print(f"   MD5      : {hash_md5}")
print("\n" + "="*100)
print("  GENERATE KUNCI DSA")
print("="*100)
keygen_start_time = datetime.now(timezone.utc)
keygen_clock_start = time.perf_counter()
private_key          =          dsa.generate_private_key(key_size=2048,
backend=default_backend())
public_key = private_key.public_key()
keygen_clock_end = time.perf_counter()
keygen_end_time = datetime.now(timezone.utc)
print(f" Waktu Generate Kunci:")
print(f"   Mulai    : {keygen_start_time.isoformat()}")
print(f"   Selesai  : {keygen_end_time.isoformat()}")
print(f"    Durasi    : {(keygen_clock_end  -  keygen_clock_start):.6f}
detik")
print(f"\n INFO KUNCI:")
print(f"   Algoritma : DSA")
print(f"   Key Size  : {private_key.key_size} bits") signing_clock_start
= time.perf_counter()
# Buat signature
ignature = private_key.sign(original_data, hashes.BLAKE2s(32))
signing_clock_end = time.perf_counter()
signing_end_time = datetime.now(timezone.utc)
# Decode signature ke (r, s)
r, s = decode_dss_signature(signature)
signature_b64 = base64.b64encode(signature).decode()
print(f" TIMESTAMP SIGNING:")
print(f"   Mulai    : {signing_start_time.isoformat()}")
print(f"   Selesai  : {signing_end_time.isoformat()}")
print(f"    Durasi    : {(signing_clock_end - signing_clock_start):.6f}
detik")
print(f"\n INFO SIGNATURE:")
print(f"   Panjang  : {len(signature)} bytes")
print(f"   Format   : Base64 encoded")
print(f"   Base64   : {signature_b64[:64]}...")
print(f"\n COMPONENTS (r, s):")
print(f"   r (decimal) : {r}")
print(f"   s (decimal) : {s}")
print(f"   r (hex)     : {hex(r)}")
```

```
print(f"   s (hex)     : {hex(s)}")
print("\n MENYIMPAN KUNCI DAN SIGNATURE...")
pub_key_pem = public_key.public_bytes(
    encoding=serialization.Encoding.PEM,
    format=serialization.PublicFormat.SubjectPublicKeyInfo
)
with open("complete_results/public_key.pem", "wb") as f:
    f.write(pub_key_pem)
with open("complete_results/signature.b64", "w") as f:
    f.write(signature_b64)
with open("complete_results/signature_components.json", "w") as f:
    json.dump({
        "r_decimal": str(r),
        "s_decimal": str(s),
        "r_hex": hex(r),
        "s_hex": hex(s),
        "signature_length": len(signature)
    }, f, indent=2)
print("✓ Kunci publik disimpan: complete_results/public_key.pem")
print("✓ Signature disimpan   : complete_results/signature.b64")
print("✓           Components           disimpan                        :
complete_results/signature_components.json")
def verify_document(filepath, signature, public_key, doc_label=""):
    """Verifikasi satu dokumen dengan timestamp lengkap"""
    verification_start = datetime.now(timezone.utc)
    verification_clock_start = time.perf_counter()
    with open(filepath, "rb") as f:
        data = f.read()
    filename = os.path.basename(filepath)
   hash_blake = hashlib.blake2s(data).hexdigest()
    hash_sha256 = hashlib.sha256(data).hexdigest()
    # Coba verifikasi
    try:
        public_key.verify
(signature, data, hashes.BLAKE2s(32))

        is_valid = True
        status = " VALID"
        error_msg = None
    except Exception as e:
        is_valid = False
        status = " TIDAK VALID"
        error_msg = f"{type(e).__name__}: {str(e)}"
    verification_clock_end = time.perf_counter()
    verification_end = datetime.now(timezone.utc)
    return {
        "filename": filename,
        "size_bytes": len(data),
        "hash_blake2s": hash_blake,
        "hash_sha256": hash_sha256,
        "is_valid": is_valid,
        "status": status,
        "error": error_msg,
        "verification_timestamps": {
            "start": verification_start.isoformat(),
```

```
            "end": verification_end.isoformat(),
            "duration_seconds":        verification_clock_end        -
verification_clock_start
        },
        "is_original": hash_blake == hash_blake2s
    }
print("\n" + "="*100)
print("VERIFIKASI SEMUA FILE")
print("="*100)
all_results = []
print(f"\n RINGKASAN SIGNATURE:")
print(f"   File asli      : {original_file}")
print(f"   Timestamp sign : {signing_start_time.isoformat()}")
print(f"   Signature size : {len(signature)} bytes")
print(f"   Components     : r={hex(r)[:20]}..., s={hex(s)[:20]}...")
print(f"\n VERIFIKASI FILE ASLI:")
print("-"*80)
# Verifikasi file asli
result_original = verify_document(original_path, signature, public_key,
"ORIGINAL")
all_results.append(result_original)
print(f"File    : {result_original['filename']}")
print(f" Ukuran  : {result_original['size_bytes']:,} bytes")
print(f" Hash BLAKE2s: {result_original['hash_blake2s']}")
print(f" Hash SHA256 : {result_original['hash_sha256'][:32]}...")
print(f" Status  : {result_original['status']}")
print(f" Waktu :
{result_original['verification_timestamps']['duration_seconds']:.6f}
detik")
print(f"                              Mulai                              :
{result_original['verification_timestamps']['start']}")
print(f"                                                  Selesai              :
{result_original['verification_timestamps']['end']}")
print(f"\n VERIFIKASI FILE MODIFIKASI ({len(modified_files)} file):")
print("-"*80)
# Verifikasi semua file modifikasi
for i, filename in enumerate(modified_files, 1):
    filepath = os.path.join("complete_docs", filename)
    print(f"\n{i}.  {filename}")
    result   =   verify_document(filepath,   signature,   public_key,
f"MODIFIED_{i}")
    all_results.append(result)
    print(f"   Ukuran  : {result['size_bytes']:,} bytes")
    print(f"    Hash BLAKE2s: {result['hash_blake2s'][:32]}...")
    print(f"    Hash SHA256 : {result['hash_sha256'][:32]}...")
    print(f"    Status  : {result['status']}")
    if result['error']:
        print(f"    Error   : {result['error'][:80]}...")
    print(f" Waktu
: {result['verification_timestamps']['duration_seconds']:.6f} detik")
    print(f"                                    Mulai                     :
{result['verification_timestamps']['start'][11:19]}")
    print(f"                                                Selesai        :
{result['verification_timestamps']['end'][11:19]}")
    # Bandingkan hash dengan asli
    if result['hash_blake2s'] == hash_blake2s:
        print(f"    Hash vs Asli:  SAMA PERSIS")
```

```
    else:
        print(f"    Hash vs Asli:  BERBEDA")
print("\n" + "="*100)
print(" STATISTIK HASIL VERIFIKASI")
print("="*100)
total_files = len(all_results)
valid_files = sum(1 for r in all_results if r['is_valid'])
same_hash_files = sum(1 for r in all_results if r['is_original'])
avg_verify_time = sum(r['verification_timestamps']['duration_seconds']
for r in all_results) / total_files
print(f"\n OVERVIEW:")
print(f"   Total file dianalisis  : {total_files}")
print(f"   File dengan signature VALID  : {valid_files}")
print(f"   File dengan hash SAMA dengan asli : {same_hash_files}")
print(f"\n  PERFORMANCE:")
print(f"       Key  generation  time         :  {(keygen_clock_end  -
keygen_clock_start):.6f} detik")
print(f"     Signing  time                   :  {(signing_clock_end  -
signing_clock_start):.6f} detik")
print(f"   Avg verification time  : {avg_verify_time:.6f} detik")
print(f"\n SIGNATURE DETAILS:")
print(f"   Algorithm              : DSA-2048 + BLAKE2s(32)")
print(f"   Signature length       : {len(signature)} bytes")
print(f"   Signature timestamp    : {signing_start_time.isoformat()}")
print(f"   r (first 20 chars)     : {hex(r)[:20]}...")
print(f"   s (first 20 chars)     : {hex(s)[:20]}...")
print("\n" + "="*100)
print(" TABEL RINGKASAN SEMUA FILE")
print("="*100)
print("\nNo.  File Name                                  Size        Hash
(BLAKE2s)                    Status      Time (s)")
print("-"*120)
for i, result in enumerate(all_results):
    filename     =     result['filename'][:30]     +     "..."     if
len(result['filename']) > 30 else result['filename']
    size_mb = f"{result['size_bytes']/1024:.1f} KB"
    hash_short = result['hash_blake2s'][:16] + "..."
    status = result['status'][:2]  # Hanya icon
    verify_time                                                    =
f"{result['verification_timestamps']['duration_seconds']:.4f}"
    print(f"{i+1:2d}.    {filename:35}  {size_mb:10}  {hash_short:35}
{status:10} {verify_time:>8}")
print("\n MENYIMPAN HASIL LENGKAP...")
complete_results = {
    "metadata": {
        "program": "Digital Signature Verification Complete",
        "timestamp": datetime.now(timezone.utc).isoformat(),
        "algorithm": "DSA-2048 + BLAKE2s(32)",
        "key_size_bits": private_key.key_size
    },
    "signing_info": {
        "original_file": original_file,
        "signing_timestamp": signing_start_time.isoformat(),
        "signing_duration_seconds": signing_clock_end
- signing_clock_start,
        "signature_length_bytes": len(signature),
```

```
        "signature_base64": signature_b64,
        "signature_components": {
            "r_decimal": str(r),
            "s_decimal": str(s),
            "r_hex": hex(r),
            "s_hex": hex(s)
        }
    },
    "key_generation_info": {
        "start_time": keygen_start_time.isoformat(),
        "end_time": keygen_end_time.isoformat(),
        "duration_seconds": keygen_clock_end - keygen_clock_start
    },
    "original_file_hashes": {
        "blake2s": hash_blake2s,
        "sha256": hash_sha256,
        "md5": hash_md5,
        "size_bytes": len(original_data)
    },
    "verification_results": all_results,
    "summary": {
        "total_files": total_files,
        "valid_signatures": valid_files,
        "files_with_same_hash_as_original": same_hash_files,
        "average_verification_time_seconds": avg_verify_time
    }
}
timestamp_str = datetime.now().strftime("%Y%m%d_%H%M%S")
json_file = f"complete_results/complete_analysis_{timestamp_str}.json"
with open(json_file, "w", encoding="utf-8") as f:
    json.dump(complete_results, f, indent=2, ensure_ascii=False)
# Simpan ke CSV
import pandas as pd
df_data = []
for result in all_results:
    df_data.append({
        "filename": result["filename"],
        "size_bytes": result["size_bytes"],
        "hash_blake2s": result["hash_blake2s"],
        "hash_sha256": result["hash_sha256"],
        "signature_valid": result["is_valid"],
        "verification_status": result["status"],
        "verification_start":
result["verification_timestamps"]["start"],
        "verification_end": result["verification_timestamps"]["end"],
        "verification_duration":
result["verification_timestamps"]["duration_seconds"],
        "is_original_file": result["is_original"]
    })
df = pd.DataFrame(df_data)
csv_file = f"complete_results/summary_{timestamp_str}.csv"
df.to_csv(csv_file, index=False, encoding='utf-8')
print(f"✓ Hasil lengkap disimpan  : {json_file}")
print(f"✓ Ringkasan tabel disimpan: {csv_file}")
print("\n" + "="*100)
print(" INFORMASI PENTING")
```

```
print("="*100)
print(f"\n FILE ASLI:")
print(f"   Nama     : {original_file}")
print(f"   Hash     : {hash_blake2s}")
print(f"   Ditandatangani pada: {signing_start_time.isoformat()}")
print(f"   Signature components:")
print(f"     r = {hex(r)[:40]}...")
print(f"     s = {hex(s)[:40]}...")
print(f"\n HASIL VERIFIKASI:")

for result in all_results:
    if result['is_original']:
        print(f"    {result['filename']}: SIGNATURE VALID (file asli)")
    else:
        if result['is_valid']:
            print(f"{result['filename']}:  SIGNATURE  VALID  (perhatian:
file modifikasi tapi signature valid!)")
        else:
            print(f"    ✓ {result['filename']}: Signature  TIDAK  VALID
(seharusnya, karena file diubah)")
print(f"\n FILE YANG BISA DIDOWNLOAD:")
print(f"   1. {json_file} - Hasil analisis lengkap (JSON)")
print(f"   2. {csv_file} - Tabel ringkasan (CSV)")
print(f"   3. complete_results/public_key.pem - Kunci publik")
print(f"   4. complete_results/signature.b64 - Signature file asli")
print("\n" + "="*100)
print("PROGRAM SELESAI")
print("="*100)
```

Lampiran 8 Kode Program Pembuatan Pasangan Kunci

```
from cryptography.hazmat.primitives.asymmetric.utils import
decode_dss_signature
dsa_numbers = private_key.private_numbers()
public_numbers = dsa_numbers.public_numbers
params = public_numbers.parameter_numbers
p = params.p
q = params.q
g = params.g
y = public_numbers.y
x = dsa_numbers.x
results = []
doc_no = 1
for file_name in uploaded_files.keys():
    if file_name.lower().endswith(".pdf"):
        document = read_document(file_name)
        # Hash dokumen
        doc_hash = hash_document(document)
        # Tanda tangan digital
        signature = private_key.sign(
            doc_hash,
            hashes.BLAKE2s(32)
        )
        # Ambil nilai (r, s)
        r, s = decode_dss_signature(signature)
        results.append({
            "Dokumen": doc_no,
            "Nama File": file_name,
            "Hash (BLAKE2s)": doc_hash.hex(),
            "r": r,
            "s": s
        })
        doc_no += 1
print("PARAMETER & KUNCI PUBLIK DSA")
print("p =", p)
print("q =", q)
print("g =", g)
print("y =", y)
print("=" * 80)
for data in results:
for data in results:
    print(f"Dokumen ke-{data['Dokumen']}")
    print(f"Nama File : {data['Nama File']}")
    print(f"Hash      : {data['Hash (BLAKE2s)']}")
print(f"r         : {data['r']}")
    print(f"s         : {data['s']}")
    print("-" * 80)
```

Lampiran 9 Kode Program 5 Perlakuan

```
# -*- coding: utf-8 -*-
"""25 DOKUMEN DAN P.ipynb
Automatically generated by Colab.
Original file is located at
!pip install pycryptodome
from Crypto.PublicKey import DSA
from Crypto.Signature import DSS
from Crypto.Hash import BLAKE2s
from google.colab import files
def hash_file(filename):
    h_blake = BLAKE2s.new(digest_bits=256)
    with open(filename, 'rb') as f:
        while chunk := f.read(4096):
            h_blake.update(chunk)
    # kompatibilitas DSA (standar FIPS)
    return SHA256.new(h_blake.digest())
private_key = DSA.generate(2048)
public_key = private_key.publickey()
signer = DSS.new(private_key, 'fips-186-3')
verifier = DSS.new(public_key, 'fips-186-3')
print(" Upload 25 DOKUMEN ASLI")
uploaded_asli = files.upload()
signature_ref = {}
for filename in uploaded_asli:
    if not filename.lower().endswith(".pdf"):
        continue
    # ambil angka TERAKHIR saja
    doc_id = filename.replace(".pdf", "").split("-")[-1].strip()
    doc_id = str(int(doc_id))  # normalisasi: hapus 0 depan
    signature_ref[doc_id] = signer.sign(hash_file(filename))
    print(f"Referensi: {filename} → ID {doc_id}")
print("\n ID referensi:", sorted(signature_ref.keys()))
print(" Upload DOKUMEN HASIL PERLAKUAN")
uploaded_modif = files.upload()
print("\n=== HASIL PENGUJIAN VALIDITAS ===")
for filename in uploaded_modif:
    if not filename.lower().endswith(".pdf"):
        continue
    fname = filename.upper().replace(".PDF", "")
    doc_id = fname.split("-P")[0].strip()
    doc_id
str(int(doc_id))  # normalisasi
    if doc_id not in signature_ref:
        print(f"{filename} → REFERENSI TIDAK DITEMUKAN (ID {doc_id})")
        continue
    try:
        verifier.verify(
            hash_file(filename),
            signature_ref[doc_id]
        )
        status = "VALID"
    except ValueError:
        status = "TIDAK VALID"
    print(f"{filename} → {status}")
```

Lampiran 10 Data Modifikasi Perlakuan pada Dokumen Asli

No. Dokumen	Perlakuan	ASLI	Modifikasi
25-1	P1	JacksOn	JacksOh
	P2	Cancer	Cacner
	P3	Miller	Miller
	P4	Blue Cross	Blue Cros
25-2	P1	Samantha	Samantaa
	P2	LesLie	LesLei
	P3	Medicare	Medicare
	P4	33643	33643
25-3	P1	Obesity	Obesiti
	P2	Aetna	Aenta
	P3	DaNnY	DannY
	P4	76	-
25-4	P1	Hernandez	Hernandes
	P2	37909	39709
	P3	Wells	Wells
	P4	Medicare	Medicar
25-5	P1	bEll	bEtl
	P2	14238	41238
	P3	Female	Female
	P4	bEll	-
25-6	P1	A+	A-
	P2	Asthma	Ashtma
	P3	Newton	Newton
	P4	Newton	-
25-7	P1	AB-	AB+
	P2	03	30
	P3	EDWaRDs	EdWaRDs
	P4	Olson	-
25-8	P1	20	25
	P2	2021	2012
	P3	MARtinez	mARtinez
	P4	Thomas	-
25-9	P1	07	09
	P2	Cigna	Cigan
	P3	AB+	Ab+
	P4	Rich and	Rich
25-10	P1	19784	19794
	P2	Walker	Wakler
	P3	UnitedHealthare	Unitedhealthare
	P4	Day	-
25-11	P1	O+	A+
	P2	daniELs	daniLEs
	P3	Duncan	Duncan
	P4	12576	12
25-12	P1	A-	B-
	P2	MARtiNeZ	AMRtiNeZ
	P3	Hypertension	hypertension
	P4	Douglas	-
25-13	P1	43282	4328B
	P2	75	57
	P3	Miller	miller
	P4	HANsEn	-

25-14	P1	AB+	AB-
	P2	bAuer	bAure
	P3	rObeRt	rObert
	P4	AB+	A
25-15	P1	brady	bradi
	P2	10	01
	P3	Arellano	arellano
	P4	bROOkE	bROOk
25-16	P1	MS. nAtalIE	MR. nAtalIE
	P2	Dougherty	Doughetry
	P3	AB-	Ab-
	P4	Female	male
25-17	P1	A+	O+
	P2	Spencer	Specner
	P3	Cervantes	cervantes
	P4	UnitedHealthcare	United
25-18	P1	Justin Kim	Justin Nim
	P2	Cigna	Cinga
	P3	cAMPBELl	cAMpBELl
	P4	17440	1740
25-19	P1	A-	A+
	P2	Moore Jr	Moore rJ
	P3	Houston PLC	Houston PlC
	P4	Hypertension	ypertension
25-20	P1	63	60
	P2	daNIEL	daNEIL
	P3	Galloway	galloway
	P4	Hammond Ltdÿ	Ham
25-21	P1	67	37
	P2	tIMOTHY	tIMOTYH
	P3	Jones LLC	jones LLC
	P4	Krista Smith	Krista Smit
25-22	P1	48	47
	P2	BRiGhT	RBiGhT
	P3	Aetna	aetna
	P4	Gregory Smith	Gregory
25-23	P1	Clark-Mayo	Clark-Kayo
	P2	O+	+O
	P3	Vanessa Newton	Vanessa newton
	P4	KatHRYn StewArt	KatHRYn
25-24	P1	EilEEn	EilEIn
	P2	Male	Mael
	P3	and Sons Smit	and sons Smit
	P4	Martinez MD	Martinez M
25-25	P1	Hypertension	Hipertension
	P2	2020	2002
	P3	AB+	aB+
	P4	Wilson Gro	Wilson

Lampiran 11 Data Hasil Akhir Tanda Tangan Digital

No.	Dokumen	Nilai r (Base64)	Nilai s (Base64)
1.	D1	PU0nCzZA0UE=	VjrTqmJ1jMQ=
2.	D2	GxRi6O/VCX0=	KZv7DJiES4E=
3.	D3	RQNa+WebWrI=	LTx2Uur8zRY=
4.	D4	YSuMMrZP4sM=	JNyYmhhK76g=
5.	D5	VgFt/FE7xpw=	UFXmZWd/0yE=
6.	D6	WrrVDXPemgA=	OfTUmnPi6vI=
7.	D7	bPqu8bvVqM8=	cx7bDA2lxlI=
8.	D8	D/aEevztXfA=	Hfb9Y/yVTu4=
9.	D9	V2emkebn9yg=	WdGX+4GXzZg=
10.	D10	EIbMOunfNKs=	g/U+cJQQv98=
11.	D11	D22lRDMVLFk=	VYlojoMYjJU=
12.	D12	Eq8kBgqWtkU=	ENPrLMr+g84=
13.	D13	RH+eWKMDiIs=	EkMqpZ+5cjk=
14.	D14	U7kuDtpSNdU=	dx+JFM0wg3s=
15.	D15	Kw5qvZJRmDo=	HZFuv1Ojc/o=
16.	D16	dqJWbprqV+U=	LDf+dmTJIGc=
17.	D17	YRLQDqsggg8=	cXmAz96Lfho=
18.	D18	HuaYb6aUI2w=	UJfPSj2+Hlk=
19.	D19	a/XiDzaHXvo=	DnqKo/LcURA=
20.	D20	O7d9OzDZID8=	JBrQAPvAWXY=
21.	D21	VnS21Jf90XU=	U8wYplmqtQg=
22.	D22	UvlPUAwZiwo=	ar1KTHV8wKw=
23.	D23	VfRDJ4hZ/sw=	NIT5RgeZCAc=
24.	D24	TMIhz94fRvw=	K7l0UHwHStM=
25.	D25	JzVbPqTCPX4=	WtXlzaXKJEw=

Lampiran 12 Tabel ASCII

Decimal	Binary	Octal	Hex	ASCII	Decimal	Binary	Octal	Hex	ASCII	Decimal	Binary	Octal	Hex	ASCII	Decimal	Binary	Octal	Hex	ASCII
0	00000000	000	00	NUL	32	00100000	040	20	SP	64	01000000	100	40	@	96	01100000	140	60	`
1	00000001	001	01	SOH	33	00100001	041	21	!	65	01000001	101	41	A	97	01100001	141	61	a
2	00000010	002	02	STX	34	00100010	042	22	"	66	01000010	102	42	B	98	01100010	142	62	b
3	00000011	003	03	ETX	35	00100011	043	23	#	67	01000011	103	43	C	99	01100011	143	63	c
4	00000100	004	04	EOT	36	00100100	044	24	$	68	01000100	104	44	D	100	01100100	144	64	d
5	00000101	005	05	ENQ	37	00100101	045	25	%	69	01000101	105	45	E	101	01100101	145	65	e
6	00000110	006	06	ACK	38	00100110	046	26	&	70	01000110	106	46	F	102	01100110	146	66	f
7	00000111	007	07	BEL	39	00100111	047	27	'	71	01000111	107	47	G	103	01100111	147	67	g
8	00001000	010	08	BS	40	00101000	050	28	(	72	01001000	110	48	H	104	01101000	150	68	h
9	00001001	011	09	HT	41	00101001	051	29	)	73	01001001	111	49	I	105	01101001	151	69	i
10	00001010	012	0A	LF	42	00101010	052	2A	*	74	01001010	112	4A	J	106	01101010	152	6A	j
11	00001011	013	0B	VT	43	00101011	053	2B	+	75	01001011	113	4B	K	107	01101011	153	6B	k
12	00001100	014	0C	FF	44	00101100	054	2C	,	76	01001100	114	4C	L	108	01101100	154	6C	l
13	00001101	015	0D	CR	45	00101101	055	2D	-	77	01001101	115	4D	M	109	01101101	155	6D	m
14	00001110	016	0E	SO	46	00101110	056	2E	.	78	01001110	116	4E	N	110	01101110	156	6E	n
15	00001111	017	0F	SI	47	00101111	057	2F	/	79	01001111	117	4F	O	111	01101111	157	6F	o
16	00010000	020	10	DLE	48	00110000	060	30	0	80	01010000	120	50	P	112	01110000	160	70	p
17	00010001	021	11	DC1	49	00110001	061	31	1	81	01010001	121	51	Q	113	01110001	161	71	q
18	00010010	022	12	DC2	50	00110010	062	32	2	82	01010010	122	52	R	114	01110010	162	72	r
19	00010011	023	13	DC3	51	00110011	063	33	3	83	01010011	123	53	S	115	01110011	163	73	s
20	00010100	024	14	DC4	52	00110100	064	34	4	84	01010100	124	54	T	116	01110100	164	74	t
21	00010101	025	15	NAK	53	00110101	065	35	5	85	01010101	125	55	U	117	01110101	165	75	u
22	00010110	026	16	SYN	54	00110110	066	36	6	86	01010110	126	56	V	118	01110110	166	76	v
23	00010111	027	17	ETB	55	00110111	067	37	7	87	01010111	127	57	W	119	01110111	167	77	w
24	00011000	030	18	CAN	56	00111000	070	38	8	88	01011000	130	58	X	120	01111000	170	78	x
25	00011001	031	19	EM	57	00111001	071	39	9	89	01011001	131	59	Y	121	01111001	171	79	y
26	00011010	032	1A	SUB	58	00111010	072	3A	:	90	01011010	132	5A	Z	122	01111010	172	7A	z
27	00011011	033	1B	ESC	59	00111011	073	3B	;	91	01011011	133	5B	[	123	01111011	173	7B	{
28	00011100	034	1C	FS	60	00111100	074	3C	<	92	01011100	134	5C	\	124	01111100	174	7C	|
29	00011101	035	1D	GS	61	00111101	075	3D	=	93	01011101	135	5D	]	125	01111101	175	7D	}
30	00011110	036	1E	RS	62	00111110	076	3E	>	94	01011110	136	5E	^	126	01111110	176	7E	~
31	00011111	037	1F	US	63	00111111	077	3F	?	95	01011111	137	5F	_	127	01111111	177	7F	DEL

Lampiran 13 Pasangan Kunci 25 Dokumen

No	Nama	*p* (2048-bit)	*q* (256-bit)	*g*	*y* (Public Key)
1	25-1 .pdf	2145468572335254 6379603957359863 3098598176583550 468421895325 ...	8308851652804067 8158389976399516 0732672124912468 441923113775 ...	1684394656361001 0745657853889661 5110261615346233 427963346368 ...	512933525396493295 171627100008849520 280647376787334371 379969 ...
2	25-2 .pdf	2562872713600016 4127326011493171 0160322022276826 777353193964 ...	1152769794854412 0799171104472185 5352172142636761 986829292683 ...	1585059554352163 6310695788537542 3391114005521247 335283121003 ...	239932794042137158 468759154921511724 761300875338285247 033542 ...
3	25-3 .pdf	1811051178568076 8703371523961917 8419827953728220 741784822635 ...	6609012222969331 0229926742849629 6196425298655054 337485911017 ...	1634305571498846 2575519058416293 6284954555397073 531153451639 ...	156527069214225480 717193040322897018 383815218933751506 090881 ...
4	25-4 .pdf	2162607864268480 7861644546359464 0257896065253711 446342555660 ...	8642051117529268 2601466233061001 6056345429683213 224351136878 ...	1984940098721457 8546329321777840 2603854965335752 216962285078 ...	152506197056018143 723492815370672517 887843938029581805 579740 ...
5	25-5 .pdf	3117735769280868 2949450304181343 7398966524501531 932357279616 ...	7448025939356776 8840255456162992 6652390389362672 076841408956 ...	1862855836900284 2877065067322004 5620389323934951 943499029497 ...	286877852095997408 958813184584381658 095808811820756864 201913 ...
6	25-6 .pdf	2316805549401631 8665555363022377 0368205022667556 350425929725 ...	7133890873783134 7114602191787162 5465133939331787 295427654941 ...	4777618238255724 3268899159940028 5830211325022587 340845827791 ...	487180085336058793 086551615227019238 535117859912407118 315885 ...
7	25-7 .pdf	1778254251752730 0338954989674342 4950079223351228 265588706016 ...	9721359461147938 2929486351144130 3139757904147970 769341031087 ...	2561634142025897 3458546983092987 4539366846967697 153181776869 ...	511244993728579644 628722800029577687 651430487916155701 789383 ...
8	25-8 .pdf	2830467164989780 9181235039388282 5404703276913770 147554387094 ...	9724754004310580 0203625167932527 4904104830578602 993531995411 ...	2705124967193091 5558696083434791 8868346957409354 969369415752 ...	240365667808015745 956354482862715558 202581198531106589 053650 ...
9	25-9 .pdf	2890403993216425 7198349952164612 0731280598463944 016821196567 ...	1070107588688423 6610137538602939 9516587046800000 493677157706 ...	2126038283057859 2359203935710475 4095479429073063 855005318966 ...	184244016067884612 528070218706916272 242182691136265259 563306 ...
10	25-10 .pdf	3137425433426032 6538488936495335 5294047152910414 922433624896 ...	8459678326584777 2271238224147609 2021643666624099 385175847498 ...	2498068798947312 5528876998190669 6465331635549646 038635066909 ...	631774995432305215 291328641087383967 986069037433382880 359755 ...
11	25-11 .pdf	3072144596196031 4305691702192371 3267344520170491 545401305671 ...	9583393142255991 5101442104724568 3588937181270278 243627748196 ...	7016320684314452 1048723107709446 1364269890324397 342909930749 ...	191128127403149453 328948258685669016 160735083865589060 200732 ...
12	25-12 .pdf	2400729511642772 1160853516385571 4359405946365651 323917099074 ...	8943811177338390 8058568667038629 8651305955488649 973786316858 ...	2529160463733498 4922980513673775 1093666616333735 302184763917 ...	160754695655159165 016494930900777126 104808265534498311 338465 ...
13	25-13 .pdf	1700966931957090 9686481947704016 5864226481247949 109194434535 ...	9831807692943555 4158704919924403 1170962991243880 499801529413 ...	1217359378794836 8268430624590407 3199665850309327 512788406496 ...	185008475730932933 179089738358342789 643540877206990460 299052 ...
14	25-14 .pdf	3015434926273145 6338128774263058 1165426744202259 958945989141 ...	6953293890398073 3365143934581844 6575810365951193 296499094359 ...	2211714986461869 5723789476088156 4469109952420462 460089783234 ...	233054517416213591 372473697197834115 199457330374776162 940696 ...
15	25-15 .pdf	2143903325057346 1005197038576357 0989663762158063 010500459617 ...	7894263953777583 9879126418556478 9523628979493270 031049679580 ...	1804733540650867 7159412789509901 9573649290880941 291337557325 ...	100297820339026364 116255141086198272 712307051484875483 566834 ...
16	25-16 .pdf	2232380131526030 8432270912283321 5704292937467026 078490222961 ...	9897809913296889 6510132029507684 3311117468346180 513220714766 ...	2215022214744812 4659273192569989 8118491194995217 106844453544 ...	101829596267513768 754332002664739127 758205317028086889 984009 ...
17	25-17 .pdf	2997392454706919 4380766841672751 6665911914208821 479480681201 ...	5796605695962021 6447899881581471 2369835891959335 826227406907 ...	1936129087426556 3974201429633806 2763337931114859 350684382069 ...	266862433029095378 163038690168325420 578548288322259716 606124 ...

18	25-18 .pdf	1872665841603587 0607398228788441 8623607178732207 552397563458 ...	7999364657804494 7050351270604508 0235725261261263 368350711415 ...	1059179715862507 2686279962068788 5638751928311539 602524792443 ...	685060755506999001 174393381519894842 850507515380818703 255373 ...
19	25-19 .pdf	2487003785564549 0470668499212135 0518878243861105 664451466260 ...	1018315962359805 4884116110061906 1553863691925479 778018744853 ...	1001866460534220 9317188627108611 6813529511620908 998857459748 ...	146176894485209517 054857491649290328 531969694834873563 299183 ...
20	25-20 .pdf	2027063206587869 4643092323735353 4675607601105759 728943665455 ...	6851731315600642 4614634141752480 0503194382608563 383814226852 ...	1321824573615997 0857931296525903 1283923853380357 292040371760 ...	886572648427245165 228966014213582616 414155235395238396 080399 ...
21	25-21 .pdf	1824689968781513 6175504779052181 4608228325395940 307608328356 ...	1017179354929927 1308262447301115 3515042610023874 241252827042 ...	1160940494744387 9346598703519348 6004219899983056 874930810776 ...	712501357153848522 284800517356311161 420152589521476414 552226 ...
22	25-22 .pdf	2125765093116481 0869716820401362 1118798636269437 484705796967 ...	8095285732374433 7228738244303922 5534556721131676 083157428684 ...	1697265198861770 5081742272601372 2231494087543250 456146157203 ...	103024647211471024 896855061037020541 135773815303203424 313343 ...
23	25-23 .pdf	1952226287255619 0342687997020797 6346395921515801 489786835139 ...	1047359358062893 8020468908089841 4093646589761827 556642409311 ...	1845007394163017 0821611042953180 3928034719554561 599181807341 ...	176486856117913030 324292570907980662 981196656892139251 678550 ...
24	25-24 .pdf	2104227975099236 7373401543538732 4475583995093756 857736327142 ...	7419971764477398 3490298067679042 6623910201162963 553697430187 ...	1502357907068376 0552354956187959 6785177126493548 606270233783 ...	254957627469707244 587114218493888823 834668580578952130 882009 ...
25	25-25 .pdf	2429318913420725 9802845945832944 8467060693940111 342152093095 ...	1049106657426134 0173058971236861 9671522814800131 752628276717 ...	9118007406395482 6855691430311413 9690298898904353 402213874564 ...	120501842864783856 020120858199674215 768506587617866017 617162 ...

RIWAYAT HIDUP

Penulis bernama Zuhruful Hikmatuz Zahro, lahir di Kota Batu pada tanggal 08 Oktober 2003. Penulis merupakan anak pertama dari tiga bersaudara dari pasangan Bapak Wiyono dan Ibu Sri Wahyuni.

Riwayat pendidikan penulis dimulai dari SD MI Tarbiyatul Ulum, kemudian melanjutkan ke SMP IT Asy-Syadzili dan SMK IT Asy-Syadzili. Selain menempuh pendidikan formal, penulis juga pernah belajar di Pondok Pesantren PPSQ Asy-Syadzili 2 pada tahun 2016 sampai tahun 2022. Selama di pondok, penulis banyak mendapatkan pengalaman berharga, terutama dalam hal kedisiplinan, tanggung jawab, dan pendalaman ilmu agama. Setelah lulus jenjang menengah akhir, penulis melanjutkan pendidikan di UIN Maulana Malik Ibrahim Malang, Program Studi Matematika, Fakultas Sains dan Teknologi pada tahun 2022.

Selama masa perkuliahan, penulis aktif dalam kegiatan akademik maupun organisasi. Selain itu, penulis juga melaksanakan Praktik Kerja Lapangan (PKL) dan menyusun skripsi dengan judul "Verifikasi Tanda Tangan Digital Berbasis Dsa dan Fungsi *Hash* Blake2s untuk Rekam Medis Elektronik". Penulis berharap karya ilmiah ini dapat memberikan kontribusi dalam pengembangan ilmu pengetahuan, khususnya di bidang kriptografi dan keamanan data.

KEMENTERIAN AGAMA RI
UNIVERSITAS ISLAM NEGERI
MAULANA MALIK IBRAHIM MALANG
FAKULTAS SAINS DAN TEKNOLOGI
Jl. Gajayana No.50 Dinoyo Malang Telp. / Fax. (0341)558933

BUKTI KONSULTASI SKRIPSI

Nama : Zuhruful Hikmatuz Zahro
NIM : 220601110062
Fakultas / Jurusan : Sains dan Teknologi / Matematika
Judul Skripsi : Verifikasi Tanda Tangan Digital Berbasis DSA dan Fungsi *Hash* BLAKE2s untuk Rekam Medis Elektronik
Pembimbing I : Muhammad Khudzaifah, M.Si
Pembimbing II : Dr. Fachrur Rozi, M.Si

No	Tanggal	Hal	Tanda Tangan
1.	10 September 2025	Konsultasi Bab I	1.
2.	17 September 2025	Konsultasi Bab I, dan II	2.
3.	24 September 2025	Konsultasi Bab I, II, dan III	3.
4.	01 Oktober 2025	Konsultasi Bab I, II, dan III	4.
5.	01 Oktober 2025	Konsultasi Kajian Agama, Bab I, dan II	5.
6.	09 Oktober 2025	Konsultasi Bab I, II, dan III	6.
7.	13 Oktober 2025	Konsultasi Kajian Agama, Bab I, II, dan III	7.
8.	16 Oktober 2025	Konsultasi Bab I, II, dan III	8.
9.	21 Oktober 2025	Konsultasi Kajian Agama, Bab I, II, dan III	9.
10.	22 Oktober 2025	Konsultasi Bab I, II, dan III	10.
11.	30 Oktober 2025	ACC Bab I, II, dan III	11.
12.	31 Oktober 2025	ACC Kajian Agama Bab I dan II	12.
13.	10 November 2025	ACC Seminar Proposal	13.
14.	26 November 2025	Konsultasi Revisi Seminar Proposal	14.
15.	10 Desember 2025	Konsultasi Bab IV dan V	15.
16.	17 Desember 2025	Konsultasi Bab IV dan V	16.

KEMENTERIAN AGAMA RI
UNIVERSITAS ISLAM NEGERI
MAULANA MALIK IBRAHIM MALANG
FAKULTAS SAINS DAN TEKNOLOGI
Jl. Gajayana No.50 Dinoyo Malang Telp. / Fax. (0341)558933

19.	26 Januari 2026	Konsultasi Kajian Agama Bab IV	19.
20.	04 Februari 2026	ACC Bab IV dan V	20.
21.	05 Februari 2026	ACC Kajian Agama Bab IV	21.
22.	13 Februari 2026	ACC Seminar Hasil	22.
23.	23 Februari 2026	Konsultasi Revisi Seminar Hasil	23.
24.	07 Maret 2026	ACC Sidang Skripsi	24.
25.	17 Maret 2026	ACC Keseluruhan	25.

Malang, 20 April 2026
Mengetahui,
Ketua Program Studi Matematika

Dr. Fachrur Rozi, M.Si.
NIP. 19800527 200801 1 012