SISTEM APLIKASI TANDA TANGAN DIGITAL PADA WEBSITE ALGORITMA RSA

TUGAS AKHIR

Diajukan Sebagai Salah Satu Syarat untuk Memperoleh Gelar Sarjana Teknik
pada Prodi Teknik Elektro Fakultas Sains dan Teknologi

UIN SUSKA RIAU

oleh:

SAID RIKZAN
11750514686

PROGRAM STUDI TEKNIK ELEKTRO
FAKULTAS SAINS DAN TEKNOLOGI
UNIVERSITAS ISLAM NEGERI SULTAN SYARIF KASIM RIAU
PEKANBARU

2024

LEMBAR PERSETUJUAN

SISTEM APLIKASI TANDA TANGAN DIGITAL PADA WEBSITE ALGORITMA RSA

TUGAS AKHIR

Oleh :

SAID RIKZAN

11750514686

Telah diperiksa dan disetujui sebagai laporan Tugas Akhir Program Studi Teknik Elektro

Di Pekanbaru, pada tanggal 27 Juni 2024

Ketua Program Studi Teknik Elektro	Pembimbing
Dr. Zulfatri Aini, S.T., M.T.	Ir. Oktaf B. Kharisma, ST., MT,. PM., APEC Eng
NIP. 19721021 200604 2 001	NIP. 19841012 201503 1 003

LEMBAR PENGESAHAN

SISTEM APLIKASI TANDA TANGAN DIGITAL PADA WEBSITE ALGORITMA RSA

TUGAS AKHIR

Oleh :

SAID RIKZAN
11750514686

Telah dipertahankan di depan Sidang Dewan Penguji
sebagai salah satu syarat untuk memperoleh gelar Sarjana Teknik
Fakultas Sains dan Teknologi Universitas Islam Negeri Sultan Syarif Kasim Riau
di Pekanbaru, pada tanggal 27 Juni 2024

Pekanbaru, 27 Juni 2024

Mengesahkan,

Dekan	Ketua Program Studi Teknik Elektro
Dr. Hartono, M.Pd	Dr. Zulfatri Aini, S.T., M.T.
NIP. 19640301 199203 1 003	NIP. 19721021 200604 2 001

DEWAN PENGUJI :

Ketua	:	Abdillah, S.Si, MIT
Sekretaris	:	Ir. Oktaf B. Kharisma, ST., MT,. IPM., APEC_Eng
Anggota I	:	Dr. Harris Simaremare, ST, MT
Anggota II	:	Ewi Ismaredah, S.Kom., M.Kom.

LEMBAR HAK ATAS KEKAYAAN INTELEKTUAL

Tugas Akhir yang tidak diterbitkan ini terdaftar dan tersedia di Perpustakaan Universitas Islam Negeri Sultan Syarif Kasim Riau dan terbuka untuk umum dengan ketentuan bahwa hak cipta ada pada penulis. Referensi kepustakaan diperkenankan dicatat, tetapi pengutipan atau ringkasan hanya dapat dilakukandengan mengikuti kaidah pengutipan yang berlaku.

Penggandaan atau penerbitan sebagian atau seluruh Tugas Akhir ini harus memperoleh izin dari Dekan Fakultas Sains dan Teknologi Universitas Islam Negeri Sultan Syarif Kasim Riau. Perpustakaan yang meminjamkan Tugas Akhir ini untuk anggotanya diharapkan untuk mengisi nama, tanda peminjaman dan tanggal pinjam.

LEMBAR PERNYATAAN

Dengan ini saya menyatakan bahwa di dalam Tugas Akhir ini tidak terdapat karya yang pernah diajukan oleh saya maupun orang lain untuk untuk keperluan lain, dan sepanjang pengetahuan saya juga tidak memuat karya atau pendapat yang pernah ditulis atau diterbitkan oleh orang lain kecuali disebutkan dalam referensi dan didalam daftar pustaka.

Saya bersedia menerima sanksi jika pernyataan ini tidak sesuai dengan yang sebenarnya.

Pekanbaru, 27 Juni 2024

Yang membuat pernyataan

SAID RIKZAN

NIM.11750514686

SURAT PERNYATAAN

Saya yang bertandatangan di bawah ini:

Nama	:	Said Rikzan
NIM	:	11750514686
Tempat, Tgl. Lahir	:	Kasikan, 7 Juli 1998
Fakultas	:	Sains dan Teknologi
Prodi	:	Teknik Elektro
Judul Jurnal	:	**Sistem Aplikasi Tanda Tangan Digital Pada Website Algoritma Rsa**

Menyatakan dengan sebenar-benarnya bahwa:

1. Penulisan skripsi dengan judul sebagaimana tersebut di atas adalah hasil pemikiran dan penelitian saya sendiri.
2. Semua kutipan pada karya tulis saya ini sudah disebutkan sumbernya.
3. Oleh karena itu skripsi saya ini, saya nyatakan bebas dari plagiat.
4. Apabila di kemudian hari terbukti terdapat plagiat dalam penulisan skripsi saya tersebut, maka saya bersedia menerima sanksi sesuai peraturan perundang-undangan.

Demikianlah Surat Pernyataan ini saya buat dengan penuh kesadaran dan tanpa paksaan dari pihak manapun juga.

Pekanbaru, 11 Juli 2024
Yang membuat pernyataan

Said Rikzan
NIM : 11750514686

LEMBAR PERSEMBAHAN

بسم الله الرحمن الرحيم

Pada lembar ini aku ingin menyampaikan sedikit ucapan terimah kasih kepada orang orang yang hadir dalam perjalanan hidupku yang mungkin tidak bisa ditulis satu persatu karena keterbatasan ingatanku. yang terlintas pertama kali dipikiranku yg sangat layak kusampaikan terimahkasih kepada keluarga yang sudah mau merawat membesarkan mendidik menyekolahkan dan keluarga sebagai madrasah pertama yang aku tempuh, dengan didikannya dan dedikasihnya aku bisa sampai ke hari ini dengan segala dinamika hidup bisa sampai mengantarkanku ke tahapku saat ini.

Terimah kasih aba

Terimah Kasih Ibu

Terima kasih keluargaku

Tak mungkin kulupakan jasa para guruku yang sudah memberikan pengetahuannya kepadaku, Maulana Jalaluddin Rumi berkata "*Guru adalah lilin yang membakar dirinya sendiri untuk menerangi jalan orang lain"* bermakna guru lah yang membimbing dan sebagai peggerak ilmu pengetahuan, yang rela mengabdikan pengetahuannya untuk mencerdaskan dan membimbing muridnya, bagai lilin yang menerangi kegelapan.

Terimah kasih Para Guruku

Terimah kasih kepada rekan, teman, sahabat yang sudah menemani hariku mewarnai jalan hidupku melukiskan sedikit banyak pada kanvas hidupku.dan terakhir tapi tidak kala pentingnya kepada diriku sendiri yang sudah hidup dan menjadi seperti sekarang akan banyak lembaran baru yang akan ditulis, dilukis, diwarnai, dicoret dan tetaplah barkarya.

Sebagai pengingat Maulana Jalaluddin Rumi :

"Jangan melihat ke luar,lihatlah kedalam diri sendiri dan carilah itu"

SISTEM APLIKASI TANDA TANGAN DIGITAL PADA WEBSITE ALGORITMA RSA

SAID RIKZAN

NIM : 11750514686

Tanggal Sidang : 27 Juni 2024

Program Studi Teknik Elektro

Fakultas Sains dan Teknologi

Universitas Islam Negeri Sultan Syarif Kasim Riau

Jl. Soebrantas No. 155 Pekanbaru

ABSTRAK

Perkembangan teknologi informasi dan komunikasi mempengaruhi keamanan dan validitas informasi. Tanda tangan digital menggunakan kriptografi menjadi solusi penting untuk memastikan integritas, kerahasiaan, dan otentikasi data. Tujuan Penelitian ini mengimplementasikan sistem tanda tangan digital pada website menggunakan algoritma Rivest Shamir Adleman (RSA), yang merupakan algoritma asimetris dengan kunci publik dan kunci privat untuk enkripsi dan dekripsi.Studi literatur dilakukan untuk memahami konsep kriptografi, tanda tangan digital, dan algoritma RSA. Sistem dirancang menggunakan Unified Modelling Language (UML) dan diuji dengan metode black box untuk memastikan fungsionalitasnya. Aplikasi yang dikembangkan berbasis web, dengan ekstensi dokumen yang digunakan adalah PDF.Hasil penelitian menunjukkan bahwa sistem aplikasi tanda tangan digital berbasis website dengan algoritma RSA berfungsi sesuai harapan, memungkinkan pengguna mengunggah dokumen, menambahkan tanda tangan digital, dan mengunduh dokumen yang disahkan. Sistem ini diharapkan dapat meningkatkan penggunaan dan keamanan transaksi elektronik di berbagai sektor di Indonesia.

Kata Kunci : Tanda tangan digital, Kriptografi, Algoritma RSA, Keamanan informasi, Website.

Digital Signature Application System on Website Using RSA Algorithm

SAID RIKZAN

NIM : 11750514686

Date of Final Exam : 27 June 2024

Department of Electrical Engineering

Faculty on Science and Technology

State Islamic of Sultan Syarif kasim Riau

Soebrantas St. No 155 Pekanbaru – Indonesia

ABSTRACT

The advancement of information and communication technology influences the security and validity of information. Digital signatures using cryptography have become an essential solution to ensure data integrity, confidentiality, and authentication. This research aims to implement a digital signature system on a website using the Rivest Shamir Adleman (RSA) algorithm, an asymmetric algorithm utilizing public and private keys for encryption and decryption. A literature review was conducted to understand the concepts of cryptography, digital signatures, and the RSA algorithm. The system was designed using Unified Modeling Language (UML) and tested with the black box method to ensure its functionality. The developed application is web-based, with the document extension used being PDF. The research results show that the website-based digital signature system using the RSA algorithm functions as expected, allowing users to upload documents, add digital signatures, and download authenticated documents. This system is expected to improve the efficiency and security of electronic transactions in various sectors in Indonesia.

Keywords: Digital Signature, Cryptography, RSA Algorithm, Information Security, Website.

KATA PENGANTAR

Assalamu'alaikum Wr.Wb

Penuh rasa syukur dan kebahagiaan, penulis ingin memanjatkan puji dan syukur kehadirat Allah SWT atas limpahan rahmat dan hidayah-Nya yang tiada terkira, sehingga penulis dapat menyelesaikan Tugas Akhir ini dengan lancar. Shalawat serta salam tak lupa penulis sampaikan kepada junjungan kita Nabi Muhammad SAW, sebagai sosok pemimpin dan panutan bagi seluruh umat manusia yang patut kita contoh dan ikuti. Dengan izin dan ridho Allah SWT, penulis telah menyelesaikan Tugas Akhir ini dengan judul "SISTEM APLIKASI TANDA TANGAN DIGITAL PADA WEBSITE ALGORITMA RSA".

Pencapaian ini tidak lepas dari bimbingan dan arahan berharga dari para pembimbing yang senantiasa memberikan dukungan dan motivasi selama proses penulisan. Doa dan restu dari keluarga tercinta serta orang-orang di sekitar penulis juga menjadi kekuatan yang mendorong penulis untuk menyelesaikan Tugas Akhir ini dengan penuh kesabaran dan keteguhan.

Sebagai wujud rasa terima kasih yang mendalam, penulis ingin menyampaikan penghargaan setinggi-tingginya kepada: kepada :

1. Aba,omak, kakak dan seluruh keluarga besar tercinta yang telah memberikan semangat, dukungan moril maupun materil dan doa kepada penulis yang selalu mendoakan penulis.

2. Bapak Prof. Dr. Khairunnas, M.Ag selaku Rektor UIN SUSKA Riau beserta kepada seluruh staf dan jajarannya.

3. Bapak Dr. Hartono, M.Pd selaku Dekan Fakultas Sains dan Teknologi UIN SUSKA Riau beserta kepada seluruh Pembantu Dekan, Staf dan jajarannya.

4. Ibu Dr. Zulfatri Aini, S.T., M.T selaku ketua Program Studi Teknik Elektro Fakultas Sains dan Teknologi UIN SUSKA Riau.

5. Bapak Sutoyo, S.T., M.T selaku sekretaris Program Studi Teknik Elektro Fakultas Sains dan Teknologi UIN SUSKA Riau.

6. Ibu Marhama Jelita, S.Pd., S.Mc selaku dosen Pembimbing Akademik selama perkuliahan penulis.

7. Bapak Ir. Oktaf B. Kharisma, ST., MT,. IPM., APEC_Eng. selaku dosen pembimbing yang sudah meluangkan waktu serta pemikirannya dalam memberikan penjelasan dan masukan yang sangat berguna sehingga penulis menjadi lebih mengerti dalam menyelesaikan Tugas Akhir ini.

8. Bapak Abdillah, S.Si, MIT selaku Ketua Sidang, bapak Dr. Harris Simaremare, S.T.,M.T selaku Dosen Penguji I dan Ibu Ewi Ismaredah, S.Kom, M.Kom selaku dosen penguji II yang yang telah banyak memberi masukan berupa kritik dan saran laporan tugas akhir ini

9. Bapak / Ibu dosen, staff dan keluarga besar Program Studi Teknik Elektro Fakultas Sains dan Teknologi Universitas Islam Negeri Sultan Syarif Kasim Riau yang tidak dapat penulis sebutkan satu-persatu yang telah memberikan ilmu sehingga penulis dapat meyelesaikan pelaksanaan tugas akhir ini..

10. Teman teman Teknik Elektro Angkatan 17 pada umumnya dan terkhusus kepada teman teman konsentrasi komputer yang sudah memnerikan semangat penulis dalam menyeleseikan tugas akhir ini.

11. Teman teman *Thrones Of Alfajar* yang telah memberikan dukungan dan motivasi kepada penulis dalam menyelesaikan Tugas Akhir ini.

12. *Last but not least*,ucapan terimah kasih pada diri sendiri yang sudah mendorong dirinya untuk maju menyelesaikan apa yang sudah dia mulai dan memulai langkah baru dalam chapter hidupnya yang berikutnya.

Saya ingin mengucapkan terima kasih yang sebesar-besarnya kepada semua pihak yang telah membantu saya dalam menyelesaikan Tugas Akhir ini, baik secara moral maupun materi. Saya berharap bantuan yang telah diberikan akan mendapatkan balasan pahala dari Allah SWT. Saya juga berharap Tugas Akhir ini dapat bermanfaat bagi saya dan para pembaca pada umumnya.

Pekanbaru, 27 Juni 2024

Penulis,

Said Rikzan
NIM. 11750514686

DAFTAR ISI

DAFTAR GAMBAR

Gambar **HAL**

DAFTAR RUMUS

DAFTAR LAMBANG

$\sum$ = Sigma

$\prod$ = panjang

ϕ = Totient

σ = sigma kecil

$\oplus$ = Xor

$\wedge$ = And

$\neg$ = Not

ROTR = Rotate right

SHR = Shift right

DAFTAR SINGKATAN

ASCII = *American Standard Code for Information Interchange.*

RSA = Rivest Shamir Adleman

DES = *Data Encryption Standar*

SHA = *Secure Hash Algorithm*

DSA = Digital Signature Algorithm

BAB I
PENDAHULUAN

1.1 Latar belakang

Meningkatnya perkembangan ilmu pengetahuan terbaru dan teknologi akan terus mempengaruhi isu-isu yang berkaitan dengan aktivitas manusia,seiring pesatnya perkembangan ilmu pengetahuan dan teknologi kehidupan manusia terpengaruhi olehnya, Ilmu sebagai hasil kegiatan manusia yang mengkaji berbagai bidang. Sedangkan teknologi dijadikan alat atau instrumen bagi manusia memenuhi kebutuhan-kebutuhannya[1]. Sektor teknologi informasi dan komunikasi merupakan salah satu sektor yang mengalami pertumbuhan yang signifikan.Pada tahun 2019, terdapat sebanyak 4,10 miliar orang yang menggunakan internet (sebesar 54,00 persen dari populasi dunia). Jumlah pengguna internet telah meningkat menjadi 4,90 miliar pada tahun 2021 sebesar 63 persen dari populasi [2].

Perkembangan teknologi informasi juga dirasakan diberbagai bidang kehidupan dan telah mengubah cara pertukaran dan pengiriman informasi dengan cara yang luar biasa Hal ini telah memungkinkan komunikasi yang lebih cepat dan efisien, dengan kemampuan untuk mengirim pesan, gambar, video dan file dokumen penting dalam hitungan detik. Perkembangan teknologi yg sangat mudah dilakukan dan sangatt cepat. Namun dengan kemudahan tersebut menimbulkan tantangan baru terkait kebenaran dan validitas informasi,tingkat keamanannya perlu diwaspadai karena rentan terhadap pihak ketiga yang ingin mengubah isi pesan tersebut[3]untuk mengamankan informasi pesan tersebut dari pihak Ketiga seperti menjaga integritas data ,kerahasiaan pesan, serta otentikasi dibutuhkan pengguanaan kriptografi.

Kriptografi merupakan ilmu yang mempelajari teknik matematika yang berkaitan dengan aspek keamanan data dan informasi seperti integritas data, otentikasi data dan validasi data. Di dalam kriptografi ada proses konversi seperti sistem enkripsi.Sistem enkripsi sendiri adalah suatu fungsi yang dapat digunakan untuk mengubah pesan yang jelas (plaintext) menjadi pesan yang disandikan (ciphertext).kemudian ada proses yang menkonversi ciphertext ke plaintext disebut dekripsi. Satu atau lebih kunci enkripsi digunakan dalam proses enkripsi dan dekripsi[4] dibidang kriptografi sendiri memiliki sebuah cara untuk melakukan otentikasi untuk memberikan stempel sebuah sandi yang bertindak sebagai pengganti tanda tangan.

Seiring dengan kemajuan ini, timbul berbagai tantangan terutama terkait keamanan informasi. Adanya risiko perubahan dan manipulasi data oleh pihak ketiga Tanda tangan sendiri sudah dijadikan manusia sebagai pembuktian otentikasi dokumen kertas sejak berabad abad lalu tanda tangan digunakan untuk validasi sebuah dokumen yg menerangkan sebuah dokumen yang diketahui dan disetujui untuk ditanda tangani. tanda tangan digital secara garis besar ialah skema matematis mengidentifikasi secara unik seorang pengirim, dan juga sebagai pembuktian keaslian dari si pengirim sebuah pesan atau dokumen digital,sehingga sebuah tanda tangan digital itu asli,dan membuat si penerima percaya pesan atau dukumen yg dikirim dikirim dari sumber yg sudah dipercaya.permasalahan yg sering dihadapi ialah terkait perubahan data dari sipengirim sampai ke penerima[5]

Semakin berkembangnya tanda tangan digital semakin banyak pula penelitian yang dilakukan dalam tanda tangan digital ini seperti penelitian yang dilakukan untuk membuktikan secara matematis bahwa data pada tanda tangan digital tidak dapat mengalami perubahan secara ilegal,sehingga menjadi salah satu solusi untuk memverifikasi suatu data, peneliti menggunakan metode algoritma tanda tangan DSA (Digital Signature Algorithm) dengan SHA-1.hasil dari penelitiannya dipatkankan hasilnya tanda tangan digital DSA Dengan SHA-1 dapat menjadi salah satu pilihan dalam pengamanan data, keamanannya terletak pada kesulitan mencari kunci privat dan kesulitan proses SHA-1[5].

Penelitian berikutnya berjudul Tanda tangan Digital Menggunakan algoritma keccak dan RSA dilatarbelakangi rentannya sebuah dokumen yang dtransmisikan diinternet terkena serangan pihak ketiga yang ingin memodifikasi isi sebuah dukumen atau pesan yg dikirim,dan si pengirim diketahui siapa pengirimnya dan tidak dapat menyangkalnya,penelitian munggunakan metode algoritma keccak dan RSA.hasil dari penelitian yang dilakukan oleh Rezania Agramanisti Azdy ini menggunakan bahasa java , dalam memudahkan penulis mengimpletasikan algoritma kriptografi pada java dibangunlah program provider bouncy castle,dengan algoritma keccak belum ditemukan dua data yang berbeda menghsailkan messege digest yg sama,penggunaan algoritma RSA dalam hal autentikasi dan non-repuadiation dapat tejamin tingkat keamannnya[7].

Berikutnya penelitian terkait tanda tangan digital Dhea Pungky Precilia dan Ahmad Izzuddin ,menngunakan metode algoritma message digest 5, penelitiannya dilatarbelakangi oleh pengiriman e-mail (elektronic mail) yang biasanya dalam pesan dilampirkan surat pemberitahuan

atau surat penagihan yg dikirim berupa file digital, pengiriman yg dilakukan melalui internet dari segi keamanannya tidak begitu kuat, sehinga dibutuhkan sebuah sistem untuk menjaga keamannya menggunakan sistem tanda tangan digital dengan algoritma messege digest 5. Dari penelitian tersebut didapatkan algoritma messege digest 5 dapat menjaga intregitas data nirpenyangkalan (non-repudition) dan otentikasi sebuah dokumen digital[8].

Algoritma yg sering digunakan dalam tanda tangan digital adalah algoritma rivest shamir adleman (RSA) [5].Algoritma RSA dikembangkan oleh Ron Rivest, Adi Shamir dan Len Adleman dari Masschusetts Institute of Technology, Algoritma Rsa merupakan algoritma asimetris yg dimana memiliki dua kunci berbeda publik dan privat untuk enskripsi dan deskripsi, kunci algoritma rsa memiliki panjang yang bervariasi 40 sampai 2048 bit,hingga saat ini keamanan algoritma rsa terjamin keaamanannya karena sulitnya untuk memfaktorkan bilangannya yg besar ke faktor prima[6].

Proses bisnis di berbagai sektor di Indonesia masih konvensional. Ketika akan menyetujui atau menandatangani sebuah dokumen, masih harus bertemu langsung. Misalnya, ketika seseorang membutuhkan tanda tangan, orang yang dimintai tanda tangan tidak ada di tempat dan tidak tahu kapan akan bisa bertemu atau menandatangani dokumen tersebut[11]. Dokumen tanda tangan biasanya juga bukan hanya ditandatangani oleh satu atau dua orang, tetapi bisa juga ada 3 sampai 5 orang yang membubuhkan tanda tangan tersebut. Sehingga, dibutuhkan sebuah cara untuk mengakomodir kebutuhan tersebut.

Penggunaan tangan digital di indonesia sendiri masih belum dimanfaatkan secara maksimal di berbagai sektor,di indonesia penggunaannya lebih banyak disektor perbankan. Untuk sektor bisnis dan pelayanan publik masih belum digunakan secara maksimal,masih banyak pengesahan dokumen secara konvensional padahal penggunaan tanda tangan digital sendiri sudah di atur pemerintah dengan kekuatan hukumnya menurut pasal 11 ayat (1) Udang-Undang Nomor 11 Tahun 2008 jo Undang-Undang Nomor 19 Tahun 2016 tentang Informasi dan Transaksi elektronik

Dengan permasalahan diatas penulis akan menyusun dan mengambil judul penelitian **"Sistem Aplikasi Tanda Tangan Digital Pada Website Algoritma RSA "**

1.2 Rumusan Masalah

Dengan latar belakang yang penulis paparkan sebelumnya,maka dapat diperoleh rumusan masalah yaitu bagaimana mengimplementasikan sistem tanda tangan digital dengan algoritma RSA pada sebuah website

1.3 Tujuan Penelitian

Tujuan dari penelitian ini mengimplementasikan tanda tangan digital dengan algoritma RSA padas sebuah website untuk proses pelayanan pengesahan dokumen

1.4 Batasan Masalah

Supaya pemabahasan tidak terlalu luas dan lebih terarah maka penulis membatasi masalah sebagai berikut:

1. Dokumen berekstensi.pdf yg digunakan pada file dokumen surat penelitian ini

2. Penelitian ini hanya membahas tentang penggunaan tanda tangan digital dengan algoritma RSA

3. Aplikasi yg dibuat berbasis web

1.5 Manfaat Penelitian

1. Diharapkan sebagai bahan referensi dalam mengimplementasikan penggunaan tanda tangan digital

2. Memberikan pemahaman yang mendalam tentang konsep algoritma RSA dan implementasi tanda tangan digital pada website.

BAB II

TINJAUAN PUSTAKA

2.1 Penelitian terkait

. Pengembangan penelitian Di dalam penelitian tugas akhir,peneliti melakukan literatur *riview* yg bersumber dari paper, jurnal, buku dan sumber terkait lainnya, dengan literatur *review* ini penulis diharapkan dapat referensi dan teori teori yg terkait dengan permasalahan yang ingin diselesaikan.adapun penelitiannya seperti penelitian yang dilakukan untuk membuktikan secara matematis bahwa data pada tanda tangan digital tidak dapat mengalami perubahan secara ilegal,sehingga menjadi salah satu solusi untuk memverifikasi suatu data, peneliti menggunakan metode algoritma tanda tangan DSA (Digital Signature Algorithm) dengan SHA-1.hasil dari penelitiannya dipatkankan hasilnya tanda tangan digital DSA Dengan SHA-1 dapat menjadi salah satu pilihan dalam pengamanan data, keamanannya terletak pada kesulitan mencari kunci privat dan kesulitan proses SHA-1[5].

Berikutnya penelitian tentang tanda tangan Digital Menggunakan Algoritme Keccak dan RSA dilatarbelakangi rentannya sebuah dokumen yg dtransmisikan diinternet terkena serangan pihak ketiga yg ingin memodifikasi isi sebuah dukumen atau pesan yang dikirim,dan sipengirim diketahui siapa pengirimnya dan tidak dapat menyangkalnya,penelitian munggunakan metode algoritma keccak dan RSA.hasil dari penelitian yg dilakukan oleh Rezania Agramanisti Azdy ini menggunakan bahasa java , dalam memudahkan penulis mengimpletasikan algoritma kriptografi pada java dibangunlah program provider bouncy castle,dengan algoritme keccak belum ditemukan dua data yg berbeda menghsailkan messege digest yg sama,penggunaan algoritma RSA dalam hal autentikasi dan non-repuadiation dapat tejamin tingkat keamannnya[7].

Penelitian selanjutnya menggunakan metode algoritma message digest 5, penelitiannya dilatarbelakangi oleh pengiriman e-mail (elektronic mail) yg biasanya dalam pesan dilampirkan surat pemberitahuan atau surat penagihan yg dikirim berufa file digital, pengiriman yang dilakukan melalui internet dari segi keamanannya tidak begitu kuat, sehinnga dibutuhkan sebuah sistem untuk menjaga keamannya menggunakan sistem tanda tangan digital dengan algoritma messege

digest 5 yg. Dari penelitian tersebut didaptkan algoritma messege digest 5 dapat menjaga intregitas data nirpenyangkalan (non-repudition) dan otentikasi sebuah dokumen digital[8].

Selanjutnya penelitian berjudul model digital signature pada dokumen formal akademik penelitian ini bertujuan mengembangkan model digital signature dengan validasi keaslian signature yang baik, dan mempercepat layanan penerbitan dokumen di universitas raharja, dengan metode agile yang dimodifikasi.model pengembangan dilakukan melalui beberapa tahap seperti desain,mengkontruksi, tes, pengiriman dan juga evaluasiModel tandatangan digital melalui penelitian ini membuktika dapat mempercepat suatu proses penerbitan dokumen format universitas, dan dapat juga menjaga kevalidasian sebuah file dokumen yang diluncurkan. Penggunaan metode berhasil dalam menyederhanakan alur proses bisnis karena dapat melakukan validasi dan pemeriksaan terhadap penerbitan file sebuah dokumen. Di lain sisi, pengembangan metode mempunyai keterbatasan apabila petugas yang mengesahkan sebuah file dokumen berganti[9].

Selanjutnya penelitian yang dilaksanakan pada Sistem Informasi Universitas Atma Jaya Yogyakarta, dilatar belakangi oleh Kebutuhan akan kecepatan dalam proses bsinis menjadikan perubahan terhadap aspek aspek bisnis tersebut, seperti penandatanganan dokumen dan validasi dokumen yg konvensional harus diubah untuk mempercepat proses bisnis tersebut,dan maraknya terjadi pemalsuan dukumen untuk menghindari kerugian, penelitian mnggunakan metode Quick Response Code, dibangun menggunakan ASP Net Core dan Microsoft SQL Server, yang dimana hasilnya. Proses penandatanganan dan verifikasi sebuah file dokumen di dalam lingkungan kampus Universitas Atma Jaya melibatkan pembuatan dokumen dalam yang berformat PDF yang dilengkapi dengan sebuah tanda tangan digital berupa kode QR[10].

2.2 Kriptografi

Kriptografi merupakan ilmu yang berhubungan dengan aspek keamanan informasi seperti integritas data, kerahasiaan dan otentikasi yg menggunakan teknik teknik matematika dalam penerapannya. Kata kriptografi berasal dari bahasa yunani *"cryptos"* berarti rahasia, dan *"graphein*" berarti tulisan jadi *cryptography* memeilikin arti tulisan rahasia,dahulu ilmu dan seni menjaga keamanan pesan disebut kriptografi.

Sejarah kriptografi sudah ada sejak 4000 tahun yang lalu oleh bangsa mesir berupa *hieroglyph* yg ada dipiramid dalam sejarah kriptografi kalisik menggunakan kertas dan pensil atau alat bantu sederhana dalam metode enkripsi,sejarah kriptografi pernah mencatat berhasil menyelematkan ratu elizabeth 1 dari terget pembunuhan setelah surat rahasia dari ratu skotlandia queen mary yg ditulisnya didalam penjara berhasil dipecahkan kodenya.dalam perang dunia II juga dari jerman membuat sebuat alat enkripsi bernama enigma, alat ini menggunakan enkripsi yang rumit, tetapi pihak sekutu berhasil memecahkan kodenya sehingga dapat mempercepat berakhirnya perang dunia ke II,orang yang memehcahkan sebuah kode kriptografi disebut kriptanalisis

Gambar 2. 1 Mesin Enigma

(Sumber: https://www.cryptomuseum.com)

Berkembangnya teknologi komputer digital memicu berkembangnya kriptografi modern yang lebih kompleks kriptografi modern beroperasi pada string biner seperti penemuan algoritma RSA yang paling banyak diknenal dalam kriptografi modern, sedangkan kriptografi klasik hanya mengenskripsi karakter per karakter,kriptografi modern juga melahirkan konsep baru seperti sertifikat digital dan tanda tangan digital[12].

Kriptografi sendiri memiliki 3 algoritma dibagi berdasarkan kunci yang dipakainya seperti :

a. Algoritma simetri

Algoritma simetri sering disebut algoritma klasik karena untuk enkripsi dan deskripsinya menggunakan kunci yg sama,penerima pesan harus mengetahui kunci dari sipengirim pesannya agar sipenerima bisa mengenkripkan pesannya tersebut.algoritma ini sudah ada sejak lebih dari 4000 tahun yg lalu.algoritma kunci simetri antara lain seperti data encryption standar (DES),international data encryption algorithm (IDEA), one time pad (OTP), Advanced Encryption standard (AES), rivest chiper 2 sampai 6 dan sebagainya

b. Algoritma asimetri

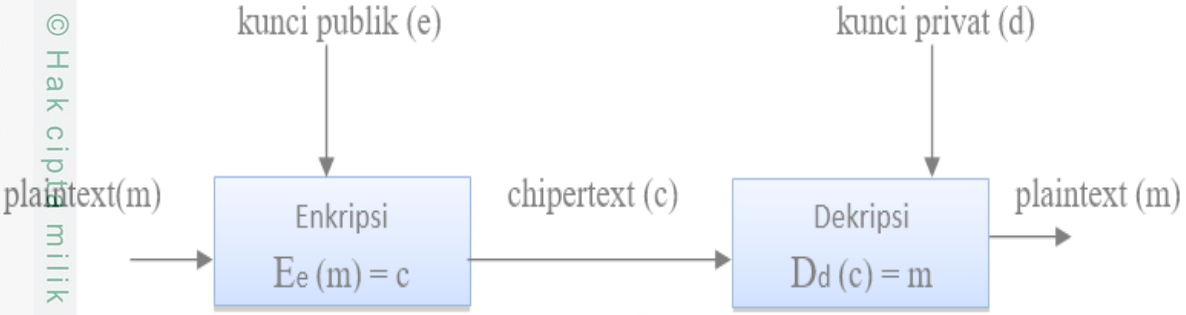
Algoritma asimetri sering disebut algoritma kunci publik,karena untuk enkripsi dan dekripsinya menggunakan kunci yang berbeda.kunci pada algoritma asimetri menggunakan 2 kunci yaitu *public key* dan *private key*, kunci yg dapat diketahui oleh semua orang disebut kunci umum *(public key)*,dan kunci yang dirahasiakan disebut kunci rahasia *(privat key)*,orang yang memeiliki kunci rahasia yang dapat mendekripsikan pesan tersebut. Algoritma kunci asimetri antara lain seperti Rivest Shamir Adleman (RSA),digital signature algorithm (DSA),eliptic curve cryptography (ECG), diffie-hellman (DH), kriptografi quantum dan lain sebagainya.

c. Fungsi hash

Fungsi hash mengambil masukan dari panjang variabel yang sembarang untuk menkonversikannya ke panjang biner yang tetap dengan mengggunakan fungsi metematika. Sering disebut juga sebagai one-way function,fingerprint,messege authentication code (MAC),messege digest. Algoritma fungsi hash antara lain seperti sha-0,sha-1,sha-256,sha-512,md2,md4,md5,ripemd,whirlpool[13].

2.3 Kriptografi Kunci Publik

Sebelum tahun 1975 hanya ada kunci simetri yang digunakan untuk enkripsi dan deskripsi untuk kunci yg sama,kemudian diffie dan hellman mengusulkan adanya kriptografi nirsimetri atau bisa disebut juga kriptografi kunci publik.karena pada kunci publik kunci untuk enskripsi sudah diumumkan pada publik,sedangkan kunci pada dekripsi hanya diketahui oleh si penerima pesan.keuntungan yg didapat kriptografi kunci publik dibandingkan dengan kriptografi kunci simetri tidak diperlukan kunci privat didistribusikan, kunci publik bisa dikirimkan melalui saluran yg sama untuk mengirimkan pesan,keuntungan berikutnya kunci publik tidak perlu banyak melakukan komunikasi rahasia dengan orang tidak diperlukan kunci rahasia sebanyak jumlah orang tersebut.



Gambar 2. 2 Kriptografi publik

Fungsi enskripsi disimbolkan E, fungsi dekripsi disimbolkan D,(e,d) adalah ialah pasangan kunci enskripsi dan dekripsi sehigga dirumuskan sebagai berikut :

$$Ee(m) = c \text{ dan } Dd(c) = m$$

Jika si A ingin mengirimkan pesan kepada B, maka B memiliki 2 kunci yaitu kunci publik dan kunci privat e dan d. si B mengirimkan kunci enskripsi (e) kepada A di sembarang saluran komunikasi, tapi si B tidak memberikan kunci dekripsinya d,selanjutnya si A ingin mengirimkan pesan m ke B, si a mengenkripsikan pesan m dengan kunci publik (e) untyk memperoleh c = $E_e(m)$,kemudian A mendeskrisikan chipertext c memakai kunci privanya (d) untuk mendapatkan m = $D_d(c)$.

Algoritma Pada kriptografi kunci publik seperti algoritma Rsa,algoritma Elgamal, algoritma diffie-hellman,algoritma knapsack dan algoritma perpangkatan-modulo

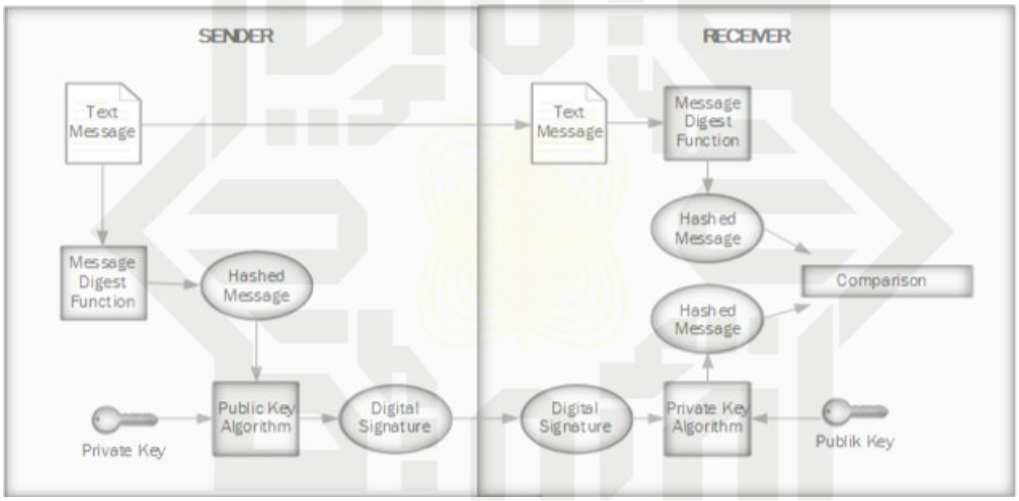
2.4 Tanda Tangan Digital

Tanda tangan sudah sangat lama digunakan sebagai otentikasi dokumen kertas seperti ijazah,buku,surat berharga,karya seni dan sebagainya, tanda tangan sendiri memiliki karakteristik sendiri sebagai berikut :

1. Tanda tangan tidak dapat dilupakan
2. Tanda tangan sebagai bukti yang otektik
3. Dokumen tidak dapat diubah setelah ditandatangani
4. Tanda tangan tidak dapat dipindah untuk pemakaian berulang ulang

5. Tanda tangan tidak bisa disangkal (*repudiation*)

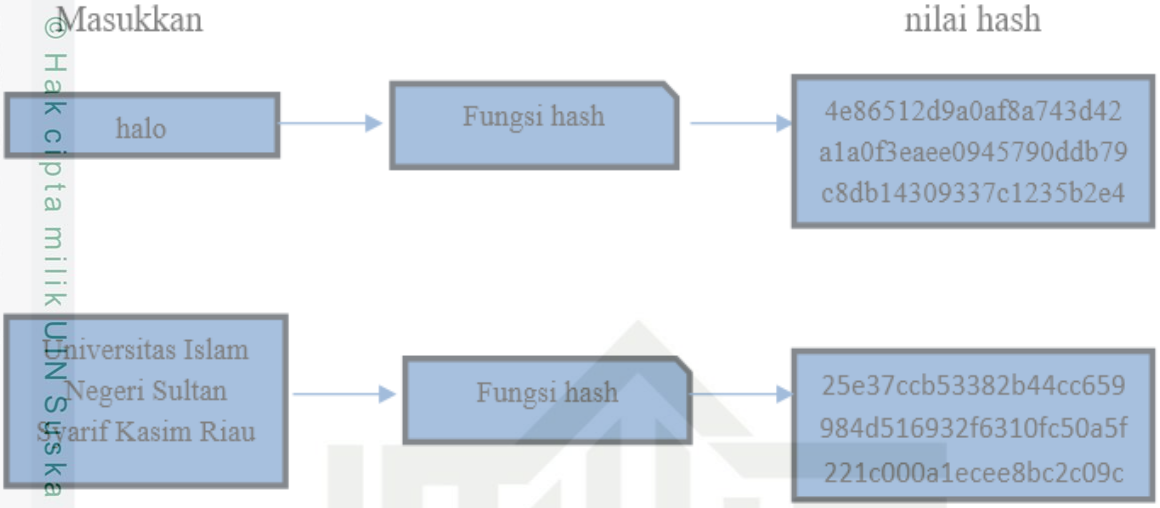
Sedangkan tanda tangan digital bukanlah sebuah tanda tangan yang mengunakan alat scanner atau memfoto tanda tangan kemudian dimasukkan di didokumen supaya di digitalisasi, tetapi tangan tangan digital adalah mekanisme pemilik pesan untuh menabuhkan sebuah sandi yang dijadikan sebagai tanda tangannya,tanda tangan digital menggunakan nilai kriptografis dalam mengotentikasi dokumen atau tanda tangannya dalam mengirim pesannya[3].jika dalam tanda tangan dokumen kertas biasa tanda tangan nya selalu sama pada semua dokumen berbeda dengan tanda tangan digital setiap dokumen memiliki tanda tangan yang bernilai kriptografis yang selalu berbeda disetiap dokumennya.



Gambar 2. 3 Tanda Tangan Digital

2.5 Fungsi Hash Sha-256

Fungsi hash adalah fungsi matematika yang mengubah suatu pesan masukan string (M) yang memiliki panjang sembarang menjadi string (h) yang keluarannya berukuran tetap (fixed).hasil dari fungsi hash disebut digest.



Gambar 2. 4 Fungsi Hash Algoritma Sha-256

Karakteristik dari fungsi hash

a. collision Resistance : sangat sulit menemukan dua input a dan b sedemikian hingga H(a) = H (b) dengan kata lain sangat sulit dan hampir tidak dapat menemukan dua set pesan yg berbeda yang memiliki nilai digest yg sama.
b. preimage resistance : untuk sembarang output y , sukar menemukan input a sedemikian hingga H (a) = y. artinya tidak dapat menemukan pesan yg sudah diberikan ilai digest
c. second preimage resistanceuntuk input a dan output y = H (a), sangat sulit menemukan input kedua b sedemikian hingga H (b) = y, dengan kata lain tidak dapat memperoleh pesan yg mempunyai nilai digest yg sama dengan digest pesan lain.

Sha sendiri adalah singkatan dari secure hash function yg merupakan fungsi hash satu arah dipublikasikan oleh sebuah lembaga bernama national intitute of standars and technology (NIST) dan dibuat oleh national secutity agency (NSA),sebelumnya adanya sha ini fungsi hash menggunakan algoritma md5 tapi karena algoritma md5 ini sudah ditemukan kolusinya, maka sha melanjutkannya.ada berbagai variasi dari algoritma sha ini,yaitu sha-0, sha-1, sha-224, sha-256, sha-384 dan sha-512.Pada prinsipnya sha-256 untuk messege-digestnya dengan panjang maksimum 2^ 64 bit.algotitma sha-256 sampai saat ini belum ada kolusinya atau yg dapat memecahkan algoritmanya.

Tahap memulai messege digest SHA-256 ada dua tahap yaitu

1. preprocesing

a. messege padding

pada tahap awal ini input pesan diubah ke bentuk biner,kemudian dibagi menjadi blok-blok, berikutnya tambahkan atau penambalan bit 1 dan 0 agar pesan memiliki panjang 512 bit. Untuk mencari panjang 0 sejumlah k menggunakan rumus :

$$l+1+k \equiv 448 \bmod 512, \qquad (2.1)$$

kemudian pada blok terakhir panjang l ditambahkan diblok terakhir

b. partisi messege

kemudian hasil pading pesannnya dibagi 16 blok setiap blok berjumalah 32 word bit,dinotasikan menjadi M_1 hingga M_{15}.

c. Berikutnya inisialisasi hash awal sha-256 dan kontanta

Nilainya sudah memiliki ketentuan yg sudah ditetapkan, nilainya sebagai berikut :

$a = H_0^{(i-1)}$ = 6a09e667

$b = H_1^{(i-1)}$ = bb67ae85

$c = H_2^{(i-1)}$ = 3c6ef372

$d = H_3^{(i-1)}$ = a54ff53a

$e = H_4^{(i-1)}$ = 510e527f

$f = H_5^{(i-1)}$ = 9b05688c

$g = H_6^{(i-1)}$ = 1f83d9ab

$h = H_7^{(i-1)}$ = 5be0cd19

berikutnya nilai kontanta sha-256

kontanta K SHA-256 sebagai berikut : K0{256}, k1{256},….k63{256}

428a2f98	71374491	b5c0fbcf	e9b5dba5	3956c25b	59f111f1	923f82a4	ab1c5ed5
d807aa98	12835b01	243185be	550c7dc3	72be5d74	80deb1fe	9bdc06a7	c19bf174
e49b69c1	efbe4786	0fc19dc6	240ca1cc	2de92c6f	4a7484aa	5cb0a9dc	76f988da
983e5152	a831c66d	b00327c8	bf597fc7	c6e00bf3	d5a79147	06ca6351	14292967
27b70a85	2e1b2138	4d2c6dfc	53380d13	650a7354	766a0abb	81c2c92e	92722c85
a2bfe8a1	a81a664b	c24b8b70	c76c51a3	d192e819	d6990624	f40e3585	106aa070
19a4c116	1e376c08	2748774c	34b0bcb5	391c0cb3	4ed8aa4a	5b9cca4f	682e6ff3
748f82ee	78a5636f	84c87814	8cc70208	90befffa	a4506ceb	bef9a3f7	c67178f2

2. Hash computation sha-256

Langkah berikutnya perhitungan menggunakan operasi perhitungan +,and,or,xor,shr,rot

a. Prepare the messege schedule {wt}

Ditahap ini siapkan penjadwalan pesan

For *i*=1 to *N*:

$$W_t = \begin{cases} M_t^{(i)} & 0 \le t \le 15 \\ \sigma_1^{\{256\}}(W_{t-2}) + W_{t-7} + \sigma_0^{\{256\}}(W_{t-15}) + W_{t-16} & 16 \le t \le 63 \end{cases} \quad (2.2)$$

b. initialize 8 variabel kerja

Variabel nya a,b,c,d,e,f,g, dan h, dengan nilai hash ke $(i\text{-}1)_{st}$

$$a = H_0^{(i-1)}$$
$$b = H_1^{(i-1)}$$
$$c = H_2^{(i-1)}$$
$$d = H_3^{(i-1)}$$
$$e = H_4^{(i-1)}$$
$$f = H_5^{(i-1)}$$
$$g = H_6^{(i-1)}$$
$$h = H_7^{(i-1)}$$

c. Berikutnya lakukan iterasi

For t=0 to 63:

{

$$T1 = h + \sum_1^{(256)}(e) + Ch(e,f,g) + K_{o\,(256)} + wt \quad (2.3)$$

$$T2 = \sum_0^{(256)}(a) + Maj\,(a,b,c) \quad (2.4)$$

$$h = g \quad (2.5)$$

g = f (2.6)

h = e (2.7)

e = d + T1 (2.8)

d = c (2.9)

c = b (2.10)

b = a (2.11)

a = T1 + T2 (2.12)

d. Compute the i_{th} intermediate hash value $H_{(i)}$:

Berikutnya lakukan penjumlahan Menjumlahkan hasil akhir a, b, c, d, e, f, g, h + initial hash value

$H0^{(i)} = a + H0^{(i-1)}$ (2.13)

$H1^{(i)} = b + H1^{(i-1)}$ (2.14)

$H2^{(i)} = c + H2^{(i-1)}$ (2.15)

$H3^{(i)} = d + H3^{(i-1)}$ (2.16)

$H4^{(i)} = e + H4^{(i-1)}$ (2.17)

$H5^{(i)} = f + H5^{(i-1)}$ (2.18)

$H6^{(i)} = g + H6^{(i-1)}$ (2.19)

$H7^{(i)} = h + H7^{(i-1)}$ (2.20)

Dan tahap terakhir setelah melakukan komputasi dari 4 tahap tadi maka didapatkan 256 bit messege digest dari pesan M

$$H_0^{(N)} \| H_1^{(N)} \| H_2^{(N)} \| H_3^{(N)} \| H_4^{(N)} \| H_5^{(N)} \| H_6^{(N)} \| H_7^{(N)}$$

2.6 Algoritma Rsa

Algoritma Rsa pertamakali dipublikasikan pada tahun 1978 oleh tiga orang yg bernama Ron Rivest, Len Adleman, dan Adi Shamir dari Masschusetts Institute of Technology , nama singkatan RSA sendiri diambil dari nama penciptanya, algoritma RSA menjadi sistem kriptografi *publik key*

yg populer,dan sangat baik digunakan untuk tanda tangan digital, penggunaan RSA bisa dikatakan bisa digukanan disemua standar protokol kriptografi termasuk SSH (*secure shell*) dan SSL/TLS (pengamanan http)[14].

Besaran dalam algoritma RSA adalah :

1. P dan q bilangan prima
2. $\phi(n) = (p-1)(q-1)$ (2.21)
3. d (kunci dekripsi)
4. m (plainteks)

diatas adalah yg harus dirahasikan, hal hal yg tidak rahasia seperti :

1. n = p.q (2.22)
2. e (kunci enskripsi)
3. c (chiperteks)

Pembangkitan kunci

1. pilih 2 bilangan prima untuk p dan q
2. hitung n = p.q
3. hitung $\phi(n) = (p-1)(q-1)$.
4. Pilih kunci publik, *e*, yang relatif prima terhadap $\phi(n)$.
5. Bangkitkan kunci privat dengan menggunakan persamaan $d = \frac{1+k\phi(n)}{e}$ (2.23)

Enkripsi pada algoritma RSA mengunakan plaintext menjadi chipertext dengan rumus sebagai berikut

$$Ci = Mi^{e}\ Mod\ n \quad (2.24)$$

Dekripsi untuk chipertext menjadi plaintext dengan rumus sebagai berikut

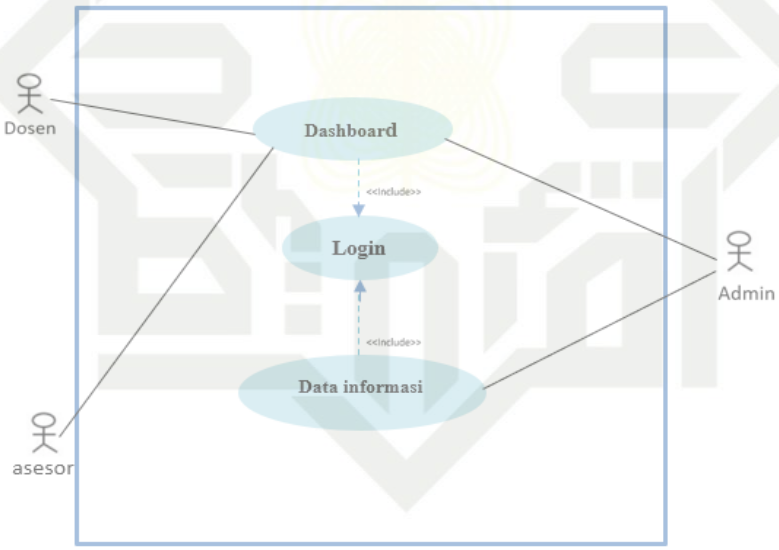
$$Mi = Ci^{d}\ Mod\ n \quad (2.25)$$

2.7 UML (Unified Modelling Language)

Uml merupakan model atau alat digunakan dalam mengembangkan perangkat lunak yg berbasis pemograman berbasis object, UML digunakan untuk membuat analisis, dan desain, mendefinisikan *requirement,* dan juga sebagai memvisualisasikan arsitektur dalam *Object Oriented Programming.*uml sendiri menjadi bahasa standar yg banyak digunakan dalam industri, Oktober 1994 penegmbangan bahasa UML seacara resmi dimulai. Pembagian dari UML seperti *use case diagram, activity diagram ,sequens diagram*

2.7.1 Use Case Diagram

Use case diagram bukan menekankan "bagaimana" sebuah sistem dibuat tapi lebih menekankan "apa" yg diperbuat sistem.use case diagram sendiri menggambarkan fungsional dari sebuah sistem yg dibuat,teknik untuk merekam persyaratan fungsional dari sebuah sistem.interaksi antara aktor dengan sistem itu sebagai representase dari use case itu sendiri.

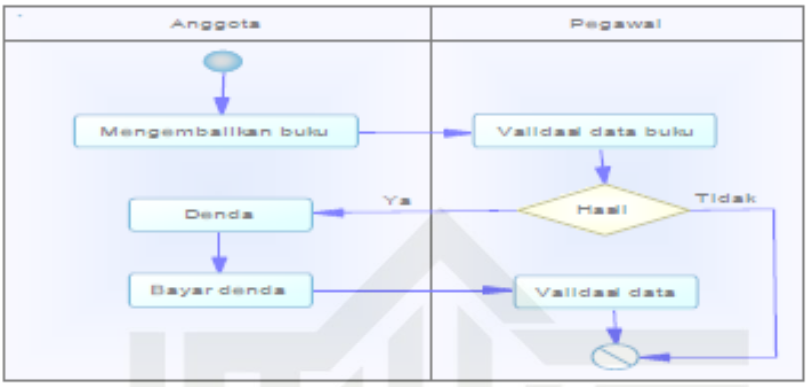


Gambar 2. 5 *Use Case Diagram*

2.7.2 Activity Diagram

Setiap use case yg dibuat,setidaknya terdapat satu activity diagram didalamnya. Activity diagram dibuat untuk menggambarkan aktivitas atau langkah langkah pada suatu sistem. activity diagram dirancang berdasarkan use case diagram yg dibuat, untuk menggambarkan sebuah sistem yg akan dirancang, daimulai dari awal sistem dijalankan,berbagai kemungkinan kemugkinaan yg akan terjadi, serta bagaimana berakhirnya dan mengagambarkan hal hal yg mungkin terjadi pada

beberapa eksekusi.activity diagram lebih menggambarkan jalur aktivitas dan proses proses dari level atas secara umum bukan menggambarkan *behaviour* internal dari sebuah sistem.



Gambar 2. 6 Activity Diagram

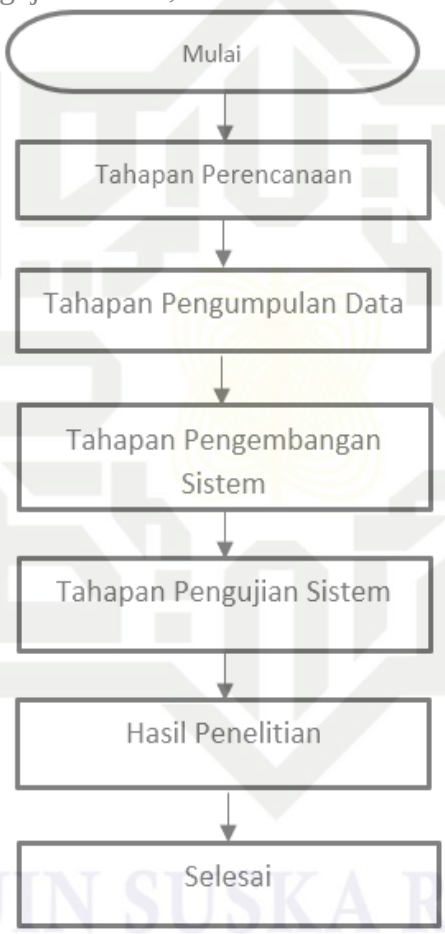
2.7.3 Sequense Diagram

Sequense diagram adalah pemodelan uml yg digunakan untuk menampilkan dan menjelaskan intersksi antar objek dari sebuah sistem dengan berurut terperinci secara jelas,sequens diagram juga menampilkan perintah dan pesan yg dirim terhadap waktu kapan dieksekusinya.

BAB III
METODE PENELITIAN

3.1 Tahapan Penelitian

Ada beberapa tahapan yg dilakukan peneliti dalam tugas akhir ini,tahapannya dimulai dari perencanaan ,kemudian teknik pengumpulan data,selanjutnya tahap pengembangan sistem,setalahnya dilakukan pengujian sistem,dan selesai



Gambar 3. 1 Flowchart Prosedur Penelitian

3.2 Tahapan Perencanaan

Peneliti melakukan alur perencanaan dimulai dengan meingidentifikasi masalah yg diteliti,menentukan judul dari penelitian, dan tujuan apa yg akan dicapai dari penelitian ini.alur kegiatan yg dilakukan dalam tahapan perencanaan ini sebagai berikut :

1. Identifikasi Masalah

Identifikasi masalah dalam penelitian ini meliputi masalah keamanan terkait dengan penggunaan tanda tangan digital. Meskipun tanda tangan digital merupakan metode yang efektif untuk mengamankan transaksi elektronik, terdapat risiko perubahan dan manipulasi data oleh pihak ketiga. Selain itu, proses penandatanganan yang masih konvensional terdapat kelemahan ketika proses persetujuan dokumen orang yang berkepentingan tidak ada ditempat mengakibatkan proses bisnis tersebut terhambat.

Selain itu, kurangnya pemanfaatan tanda tangan digital di berbagai sektor di Indonesia, terutama di sektor bisnis dan pelayanan publik, meskipun penggunaannya telah diatur oleh pemerintah. Hal ini menunjukkan potensi untuk mengembangkan sistem aplikasi tanda tangan digital yang dapat meningkatkan keamanan transaksi elektronik di berbagai sektor.

2. Judul Penelitian

Peneliti berikutnya menentukan judul yg digunakan dalam penelitian ini berdasarkan latar belakang masalah yg sudah diindentifikasi bersamaan data yg sudah dianalisa,maka penulis mendapatkan judul penelitian sebagai berikut “SISTEM APLIKASI TANDA TANGAN DIGITAL PADA WEBSITE ALGORITMA RSA”

3. Tujuan dari Penelitian

Tujuan dari penelitian tugas akhir ini adalah mengimplementasikan tanda tangan digital dengan algoritma RSA pada sebuah website dalam penggunaan tanda tangan digital. Hal ini bertujuan untuk meningkatkan proses penggunaan tanda tangan digital bagi masyarakat secara umum.

3.3 Tahapan Pengumpulan Data

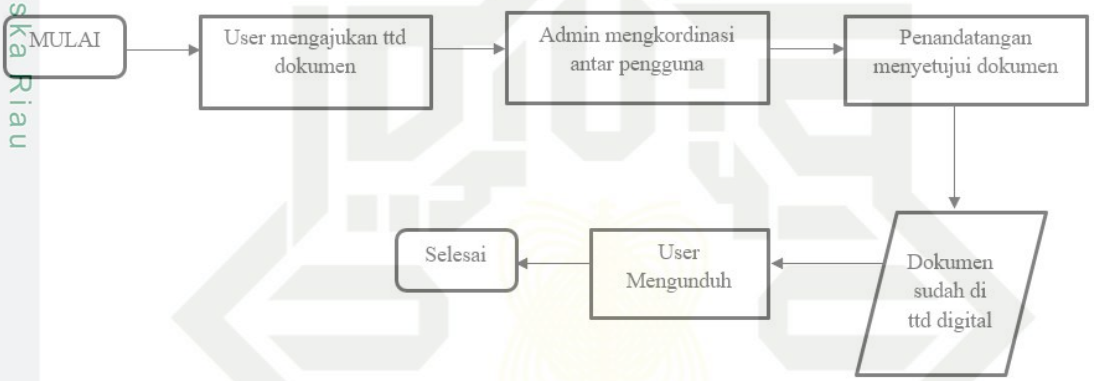
Pada tahap ini, penulis mengumpulkan data melalui studi literatur mengenai pemodelan matematis dari tanda tangan digital, mencari dan mempelajari penelitian terkait dari berbagai referensi seperti buku, jurnal, paper, atau sumber lainnya. Hal-hal yang perlu dipelajari mencakup pemahaman tentang kriptografi, jenis-jenis algoritma kriptografi yang digunakan dalam tanda tangan digital seperti DSA, RSA, dan algoritma message digest 5, karakteristik algoritma RSA, pengalaman penelitian terdahulu tentang penggunaan tanda tangan digital dengan berbagai algoritma, serta kebijakan hukum terkait penggunaan tanda tangan digital di Indonesia..

3.4 Tahapan Pengembangan Sistem

Dalam pengembagan sistem pada tugas akhir ini ada beberapa hal yg dilakukan seperti analisa sistem, untuk merancang sistem menggunakan model UML software visual paradigma for uml, dan komputasi tanda tangan digital

3.4.1 Perancangan Sistem

Proses aplikasi tanda tangan digital pada website sebagai berikut :



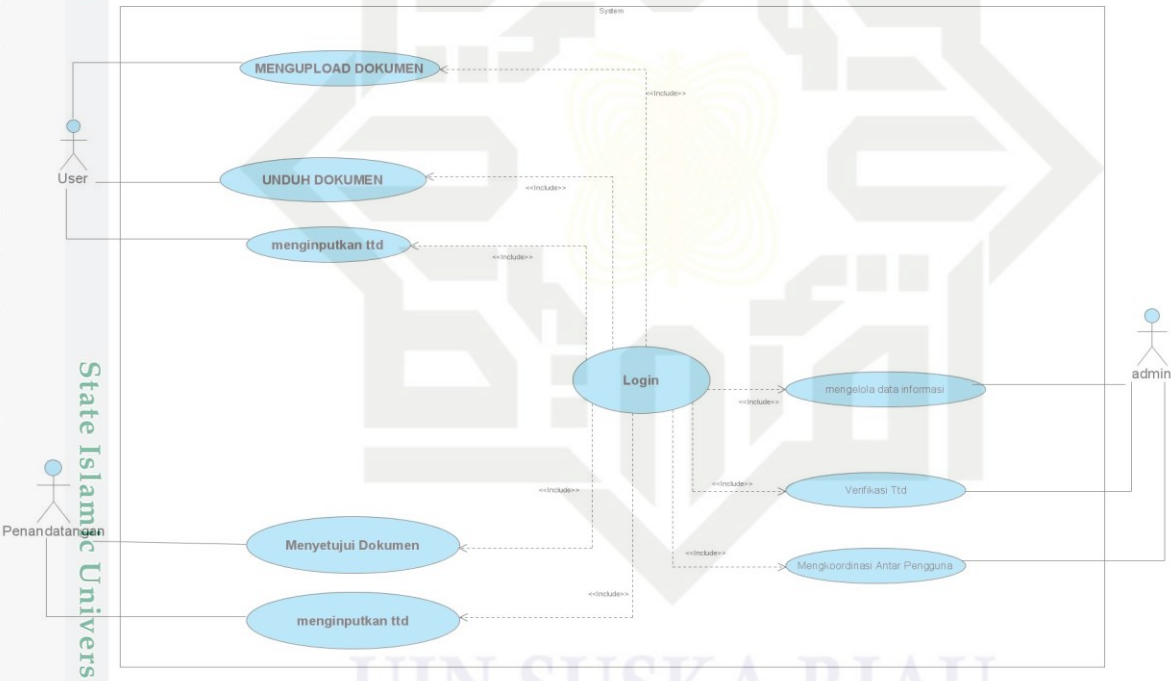
Gambar 3. 2 Alur Proses Aplikasi Tanda Tangan Digital Pada Website tanda tangan digital

Diagram alur yang ditampilkan menggambarkan proses kerja dari sistem tanda tangan digital dengan melibatkan tiga aktor utama: User, Admin, dan Penandatangan. Langkah-langkah Proses di mulai ketika pengguna (User) mengajukan dokumen untuk ditandatangani secara digital. Berikutnya user Mengajukan tanda tangan dokumen yaitu Pengguna mengunggah dokumen yang memerlukan tanda tangan digital ke dalam sistem. Proses berikutnya Admin Mengkoordinasi Antar Pengguna, setelah dokumen diajukan admin mengkoordinasikan proses antara pengguna yang mengajukan dokumen dan pihak yang akan menandatangani dokumen. admin memastikan bahwa dokumen tersebut diteruskan ke penandatangan yang tepat dan memantau alur kerja untuk memastikan tidak ada hambatan. Tahap selanjutnya penandatangan Menyetujui dokumen, penandatangan meninjau dokumen dan jika semuanya sesuai, menandatangani dokumen secara digital.Setelah penandatangan menyetujui dokumen, dokumen tersebut dianggap sudah ditandatangani secara digital. Dokumen yang telah ditandatangani secara digital tersedia dalam sistem.Status dokumen berubah menjadi sudah ditandatangani digital, yang menunjukkan bahwa

dokumen siap untuk diunduh oleh pengguna. terakhir pengguna dapat mengunduh dokumen yang sudah ditandatangani dari sistem. Proses berakhir setelah pengguna berhasil mengunduh dokumen yang sudah ditandatangani, dokumen sekarang bisa digunakan oleh pengguna untuk keperluan lebih lanjut.

3.4.2 *Use Case Diagram*

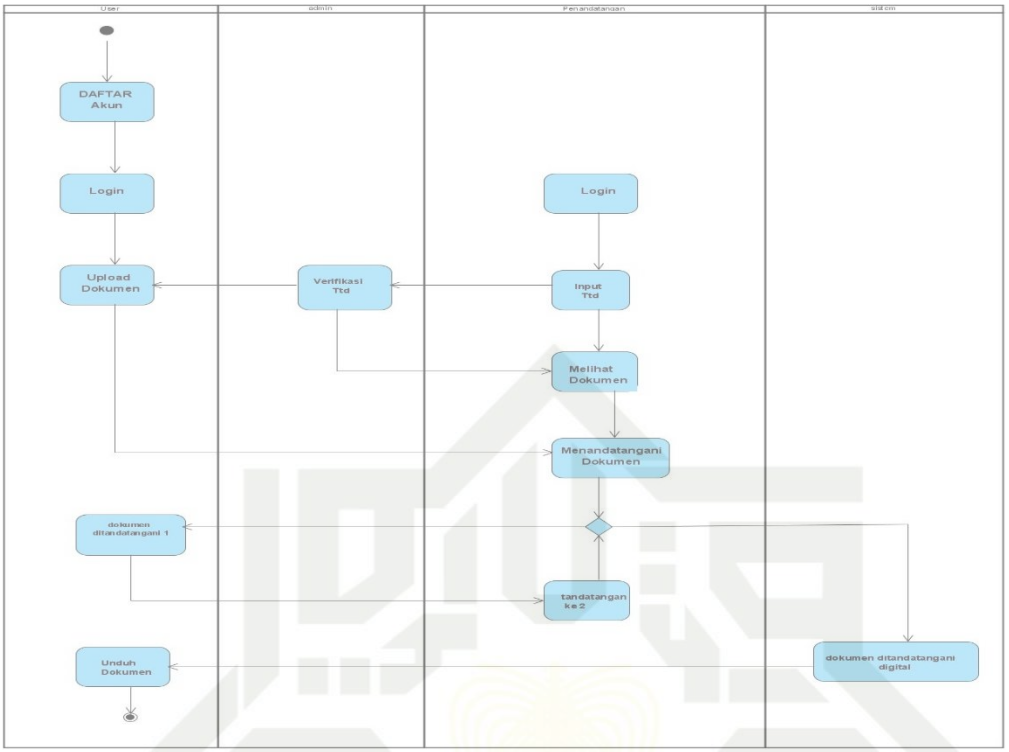
berdasarkan gambar 3.2 *Use case* dibawah, terdapat 3 aktor didalam sistem yaitu user,penandatangan, dan admin. User mengupload dokumen,unduh dokumen dan menginputkan ttd,penandatangan memiliki tugas untuk menandatangani dokumen setelah memeriksa dokumennya,dan juga untuk menginputkan tandatangannya.sedangkan admin memiliki tugas untuk mengelola data informasi,memverifikasi ttd, dan mengkoordinasikan antar pengguna.



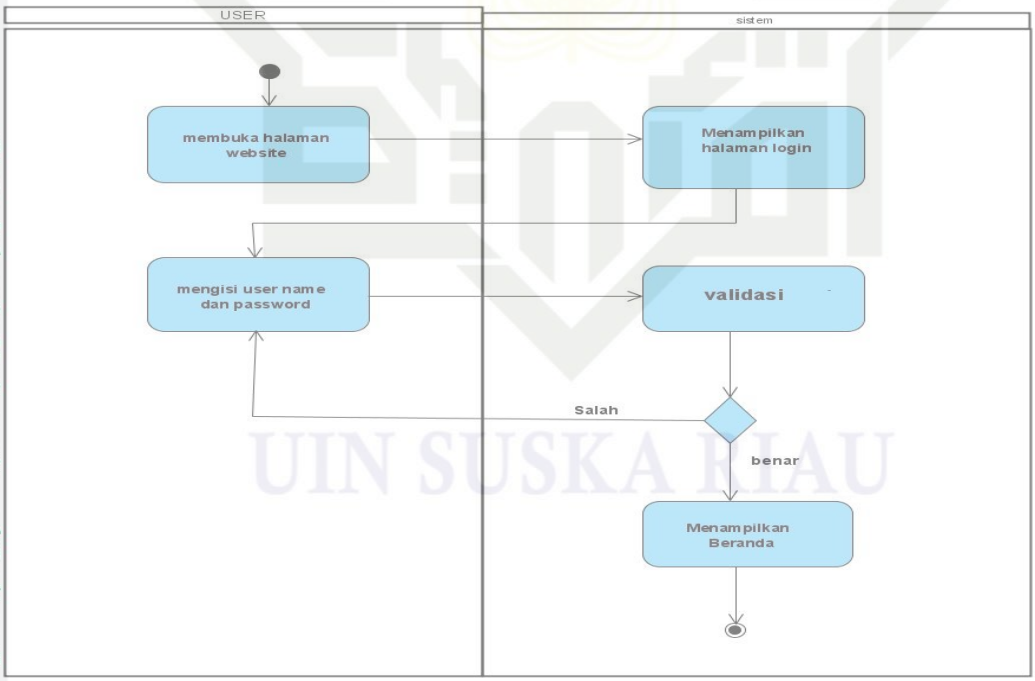
Gambar 3. 3 Use Case website tanda tangan digital

3.4.3 *Activity Diagram*

Pada gambar 3.4 user, memasukkan tanda tangan mereka. Setelah itu, seorang penandatangan juga memasukkan tanda tangan mereka sebagai langkah verifikasi. Yg bertugas untuk memvalidasi nya yaitu admin. Selanjutnya user mengunplod dokumen yang akan diajukan kepada verifikator untuk di tandatangani,penandatngan akan meninjau terlebih dahulu dokumen yg diajukan sebelum ditandatangani.setelah semua selesai user menunduh dokumen yg sudah ditandatangani tersebut.



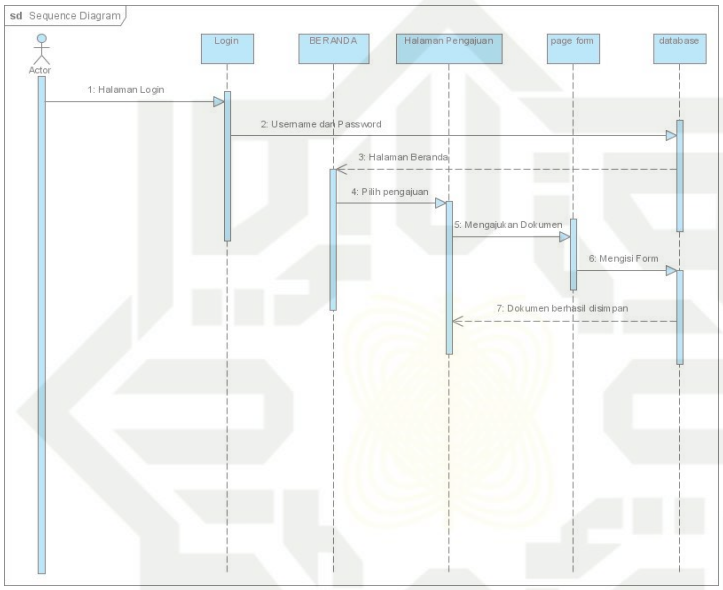
Gambar 3. 4 *Aktivity Diagram*



Gambar 3. 5 *Activity Diagram Login*

3.4.4 *Sequens Diagram*

Pada sequens diagram digambarkan langkah demi langkah proses didalam aplikasi seperti langkah awal user yaitu memasukkan username dan password, berikutnya memilih halam pengajuan. kemudian mengajukan dokumen yg akan di tanda tangani,langkah berikutnya user mengisi form yang isinya untuk menentukan dokumen tersebut nama file dokumen tersebut,berikutnya mengunggah dokumen yg akan ditandatangani oleh pihak penandatangan.



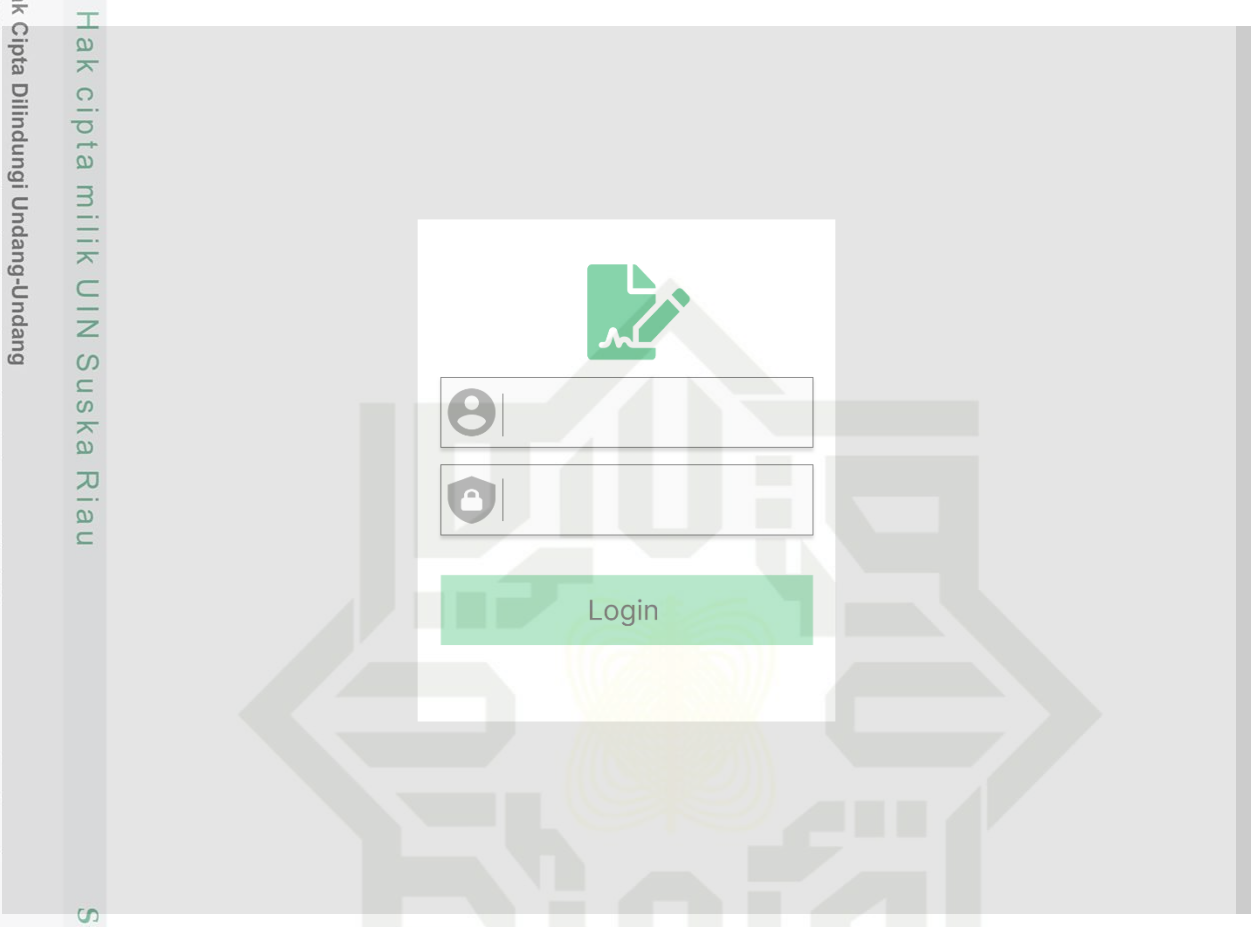
Gambar 3. 6 *Sequens Diagram* mengajukan dokumen

3.4.5 Perancangan Penandatanganan Digital pada Website

Alur awal penandatanganan dengan cara mengubah dokumen menjadi messege digest,kemudian mengenkripsi messege digest tersebut. Hasil enksripsi tersebut lalu disematkan pada file dokumen

Gambar 3. 7 Alur Proses Tanda Tangan Digital Sebuah Dokumen diwebsite tanda tangan digital

3.4.6 Perancangan Antarmuka Aplikasi Tanda Tangan Digital



Gambar 3. 8 Halaman Login

Pada saat mengakses halaman web ,akan ditampilkan halaman untuk login terlebih dahulu,halaman login berisi inputan untuk Id penguna dan juga kata sandi

Gambar 3. 9 Halaman Beranda

Pada halaman depan ditampilkan beranda , dan pada side bar akan diberikan beberapa pilihan menu yaitu Beranda , Dokumen Saya, pengajuan, profil dan logout.

Gambar 3. 10 pengajuan dokumen

Pada gambar 3. Menampilkan Halaman pengajuan jika user ingin mengajuakan dokumen yang ingin diminta persetujuam dan ditandatangani.

Gambar 3. 11 pengesahan dokumen

Pada halaman pengesahan dokumen. Pihak yg memberi persetujuan tanda tangan dapat melihat isi dokumen apa yg yg akan disetujui dan menambahkan penandatangan

Gambar 3. 12 dokumen user

Berikutnya file dokumen yg sudah ditanda tangani dapat di unduh oleh user

3.4.7 Perancangan fungsi hash sha

Langkah pertama ketika melakukan penandatanganan digital pada sebuah dokumen adalah mengubah e-dokumen tersebut menjadi message digest dengan menggunakan fungsi hash SHA-256.ada beberapa tahap dalam algoritma sha-256 sebagai berikut:

1. Tahap praprocesing

Praprocesing atau prapemrosesan SHA-256 ini terdiri dari messege padding, partisi messege, dan inisialisasi hash awal sha-256 dan kontanta.

a. Messege Padding

Pada tahap awal ini mengubah pesan inputan dikonversi kebiner dengan menggunakan tabel ASCII, Misalkan memasukkan input teks “bkd”.berikutnya mengubah inputan teks “bkd” dikonversi ka dalam bentuk biner

M = bkd

M :	b	k	d
	01100010	01101011	01100100
	8 bit	8 bit	8 bit

Memiliki panjang messege l = 24 bit

Selanjutnya messge padding tambahkan bit “1” dan sisanya bit “0” sejumlah k hingga panjang pesannya menjadi 512 bit

l+1+k ≡ 448 mod 512

k = l + 1 ≡ 448 mod 512 k = 24 + 1 ≡ 448 mod 512

k = 25 ≡ 448 mod 512 k = 448 – 25 = 423

maka nilai k = 423. Maka banyak bit “0” ditamhkan sebanyak 423 bit

berikutnya tambahkan jumlah pesan pada akhir pesan yg dipading

l = 24 = 00011000 maka hasil yg dipadding menjadi [$M^{(0)}$] :

01100010 01101011 01100100 10000000 00000000 00000000 00000000 00000000

00000000 00000000 00000000 00000000 00000000 00000000 00000000 00000000

00000000 00000000 00000000 00000000 00000000 00000000 00000000 00000000
00000000 00000000 00000000 00000000 00000000 00000000 00000000 00000000
00000000 00000000 00000000 00000000 00000000 00000000 00000000 00000000
00000000 00000000 00000000 00000000 00000000 00000000 00000000 00000000
00000000 00000000 00000000 00000000 00000000 00000000 00000000 00011000

b. Partisi Message

Berikutnya membagi setiap 512-bit menjadi 16 word 32-bit

$M_0^{(i)}$	=	0110 0010	0110 1011	0110 0100	1000 0000
$M_1^{(i)}$	=	0000 0000	0000 0000	0000 0000	0000 0000
$M_2^{(i)}$	=	0000 0000	0000 0000	0000 0000	0000 0000
$M_3^{(i)}$	=	0000 0000	0000 0000	0000 0000	0000 0000
$M_4^{(i)}$	=	0000 0000	0000 0000	0000 0000	0000 0000
$M_5^{(i)}$	=	0000 0000	0000 0000	0000 0000	0000 0000
$M_6^{(i)}$	=	0000 0000	0000 0000	0000 0000	0000 0000
$M_7^{(i)}$	=	0000 0000	0000 0000	0000 0000	0000 0000
$M_8^{(i)}$	=	0000 0000	0000 0000	0000 0000	0000 0000
$M_9^{(i)}$	=	0000 0000	0000 0000	0000 0000	0000 0000
$M_{10}^{(i)}$	=	0000 0000	0000 0000	0000 0000	0000 0000
$M_{11}^{(i)}$	=	0000 0000	0000 0000	0000 0000	0000 0000
$M_{12}^{(i)}$	=	0000 0000	0000 0000	0000 0000	0000 0000
$M_{13}^{(i)}$	=	0000 0000	0000 0000	0000 0000	0000 0000
$M_{14}^{(i)}$	=	0000 0000	0000 0000	0000 0000	0000 0000
$M_{15}^{(i)}$	=	0000 0000	0000 0000	0000 0000	0001 1000

c. Inisialisasi variabel dan konstanta

Tahap ini kta menginisialisasi variabel awal SHA-256 sebagai berikut

a = $H_0^{(i-1)}$ = 6a09e667

b = $H_1^{(i-1)}$ = bb67ae85

c = $H_2^{(i-1)}$ = 3c6ef372

d = $H_3^{(i-1)}$ = a54ff53a

e = $H_4^{(i-1)}$ = 510e527f

f = $H_5^{(i-1)}$ = 9b05688c

g = $H_6^{(i-1)}$ = 1f83d9ab

h = $H_7^{(i-1)}$ = 5be0cd19

kontanta K SHA-256 sebagai berikut : $K_0^{\{256\}}$, $k_1^{\{256\}}$,….$k_{63}^{\{256\}}$

428a2f98	71374491	b5c0fbcf	e9b5dba5	3956c25b	59f111f1	923f82a4	ab1c5ed5
d807aa98	12835b01	243185be	550c7dc3	72be5d74	80deb1fe	9bdc06a7	c19bf174
e49b69c1	efbe4786	0fc19dc6	240ca1cc	2de92c6f	4a7484aa	5cb0a9dc	76f988da
983e5152	a831c66d	b00327c8	bf597fc7	c6e00bf3	d5a79147	06ca6351	14292967
27b70a85	2e1b2138	4d2c6dfc	53380d13	650a7354	766a0abb	81c2c92e	92722c85
a2bfe8a1	a81a664b	c24b8b70	c76c51a3	d192e819	d6990624	f40e3585	106aa070
19a4c116	1e376c08	2748774c	34b0bcb5	391c0cb3	4ed8aa4a	5b9cca4f	682e6ff3
748f82ee	78a5636f	84c87814	8cc70208	90befffa	a4506ceb	bef9a3f7	c67178f2

2. Hash computation sha-256

a. Prepare the messege schedule {wt}

Pada tahap ini kita melakukan messege schedule,pertama kita ubah $M_0^{(i)}$, $M_0^{(i)}$, ..., $M_0^{(i)}$ menjadi W0,W1, ..., W_{63}

W_0 =	01100010	01101011	01100100	10000000
W_1 =	00000000	00000000	00000000	00000000
W_2 =	00000000	00000000	00000000	00000000
W_3 =	00000000	00000000	00000000	00000000

W_4 = 00000000 00000000 00000000 00000000

W_5 = 00000000 00000000 00000000 00000000

W_6 = 00000000 00000000 00000000 00000000

W_7 = 00000000 00000000 00000000 00000000

W_8 = 00000000 00000000 00000000 00000000

W_9 = 00000000 00000000 00000000 00000000

W_{10} = 00000000 00000000 00000000 00000000

W_{11} = 00000000 00000000 00000000 00000000

W_{12} = 00000000 00000000 00000000 00000000

W_{13} = 00000000 00000000 00000000 00000000

W_{14} = 00000000 00000000 00000000 00000000

W_{15} = 00000000 00000000 00000000 00011000

Untuk mendapatkan W_{16} sampai W_{63} menggunakan rumus yg ada pada bab 2 2.2

for *i*=1 to *N*:

$$W_t = \begin{cases} M_t^{(i)} & 0 \le t \le 15 \\ \sigma_1^{\{256\}}(W_{t-2}) + W_{t-7} + \sigma_0^{\{256\}}(W_{t-15}) + W_{t-16} & 16 \le t \le 63 \end{cases}$$

Langkah pertama mencari nilai W_{16} sehingga :

$W_t = \sigma_1^{\{256\}}(W_{t-2}) + W_{t-7} + \sigma_0^{\{256\}}(W_{t-15}) + W_{t-16}$

$w_{16} = \sigma_1^{\{256\}}(W_{16-2}) + W_{16-7} + \sigma_0^{\{256\}}(W_{16-15}) + W_{16-16}$

$w_{16} = \sigma_1^{\{256\}}(W_{14}) + W_9 + \sigma_0^{\{256\}}(W_1) + W_0$

σ_1 = ROTR^{7}(X) $\oplus$ RORT^{18}(X) $\oplus$ SHR^{3}(X)

σ_1 = ROTR^{17}(X) $\oplus$ RORT^{19}(X) $\oplus$ SHR^{10}(X)

$\sigma_0^{\{256\}}$ = ROTR^{7}(X) $\oplus$ RORT^{18}(X) $\oplus$ SHR^{3}(X)

$= \text{ROTR}^{7}(w14) \oplus \text{RORT}^{18}(w14) \oplus SHR^{3}(w14)$

$= \text{ROTR}^{7}00000000000000000000000000000000 \oplus$

$\text{RORT}^{18}00000000000000000000000000000000 \oplus$

$SHR^{3}00000000000000000000000000000000$

$= 00000000000000000000000000000000 \oplus 00000000000000000000000000000000 \oplus$

$00000000000000000000000000000000$

$\sigma_0^{\{256\}} = 00000000000000000000000000000000$

$\sigma_1^{\{256\}} \text{ROTR}^{17}(X) \oplus \text{RORT}^{19}(X) \oplus SHR^{10}(X)$

$= \text{ROTR}^{17}(w14) \oplus \text{RORT}^{19}(w14) \oplus SHR^{10}(w14)$

$= \text{ROTR}^{17}00000000000000000000000000000000 \oplus$

$\text{RORT}^{19}00000000000000000000000000000000 \oplus$

$SHR^{10}00000000000000000000000000000000$

$= 00000000000000000000000000000000 \oplus 00000000000000000000000000000000 \oplus$

$00000000000000000000000000000000$

$\sigma_1^{\{256\}} = 00000000000000000000000000000000$

$W_{16} = \sigma_1^{\{256\}}(w_{14}) + w_9 + \sigma_0^{\{256\}}(w_1) + w_0$

$= 00000000000000000000000000000000 + 00000000000000000000000000000000 +$

$00000000000000000000000000000000 + 01100001011000100110001110000000$

$W_{16} = 01100010011010110110010010000000$

Ulangi lagi untuk mendapatkan nilai W_{17} hingga W_{63} maka diperoleh :

W_0 =	01100010	01101011	01100100	10000000
W_1 =	00000000	00000000	00000000	00000000
W_2 =	00000000	00000000	00000000	00000000
W_3 =	00000000	00000000	00000000	00000000

W_4 =	00000000	00000000	00000000	00000000
W_5 =	00000000	00000000	00000000	00000000
W_6 =	00000000	00000000	00000000	00000000
W_7 =	00000000	00000000	00000000	00000000
W_8 =	00000000	00000000	00000000	00000000
W_9 =	00000000	00000000	00000000	00000000
W_{10} =	00000000	00000000	00000000	00000000
W_{11} =	00000000	00000000	00000000	00000000
W_{12} =	00000000	00000000	00000000	00000000
W_{13} =	00000000	00000000	00000000	00000000
W_{14} =	00000000	00000000	00000000	00000000
W_{15} =	00000000	00000000	00000000	00011000
W_{16} =	01100010	01101011	01100100	10000000
W_{17} =	00000000	00001111	00000000	00000000
W_{18} =	11011110	11001000	10100111	10100001
W_{19} =	01100000	00000000	00000011	11000110
W_{20} =	01000111	00010011	01100110	10010100
W_{21} =	00000001	10000011	11111100	00000000
W_{22} =	11011111	10001001	01101111	11001010
W_{23} =	11100011	11101011	11000100	10001110
W_{24} =	10011010	00111010	11010110	01101110
W_{25} =	01111001	10110110	11111111	00011010
W_{26} =	10010001	11011100	00010100	10110101
W_{27} =	11100111	10000011	10000100	11100110
W_{28} =	10001010	01101100	00001001	11010000
W_{29} =	10010010	01011111	10111111	10011010
W_{30} =	10011001	11100010	10010100	00001010

W_{31} =	10010111	11111111	00000101	11010100
W_{32} =	10110100	11000110	11001001	00000101
W_{33} =	01100101	11101101	00101101	01111111
W_{34} =	00000100	00001101	00100010	01001111
W_{35} =	00010111	00111110	10001110	01110001
W_{36} =	00001110	00010110	00000000	00010101
W_{37} =	00001000	00010110	01111111	10101001
W_{38} =	00101000	00101101	00110000	11001001
W_{39} =	00000100	10111110	10111000	00011000
W_{40} =	11000010	10100010	11101010	10111100
W_{41} =	10000101	00000111	10101101	10010111
W_{42} =	00000001	01011010	11000011	01011010
W_{43} =	11001011	00011111	01110100	01110101
W_{44} =	10010100	11010010	10001011	11001010
W_{45} =	10110010	00100000	10111101	01100101
W_{46} =	00101101	11111111	10110011	11000001
W_{47} =	01010011	10000101	00010001	10001110
W_{48} =	00100010	10001011	11101001	11010110
W_{49} =	01101000	01000100	01110010	11011101
W_{50} =	10011100	01011011	11110111	11001111
W_{51} =	00001111	00011011	11010001	11111000
W_{52} =	00010010	01101000	10001001	11100100
W_{53} =	10100100	00111110	10101010	01110111
W_{54} =	01110000	00010100	11111111	00001110
W_{55} =	10000011	00100100	11110011	01001010
W_{56} =	11100000	00110001	01101100	01111000
W_{57} =	00001110	01001111	11101011	11101011

W_{58} = 11011011 11101010 10011000 00111000

W_{59} = 00001011 01010100 11000100 11000111

W_{60} = 01001010 10110111 11111100 11001101

W_{61} = 10001000 01000011 10110000 11000100

W_{62} = 00000101 11000110 10000101 11110000

W_{63} = 00110100 01110001 00100110 00010110

b. initialize 8 variabel kerja

inisialisasi variabel a, b,c ,d ,e ,f ,g dan h. untuk $M^{(i)}$ dengan nilai hash awal

a = $H_0^{(i-1)} = H_0^{(0)}$ = 6a09e667

b = $H_1^{(i-1)} = H_1^{(0)}$ = bb67ae85

c = $H_2^{(i-1)} = H_2^{(0)}$ = 3c6ef372

d = $H_3^{(i-1)} = H_3^{(0)}$ = a54ff53a

e = $H_4^{(i-1)} = H_4^{(0)}$ = 510e527f

f = $H_5^{(i-1)} = H_5^{(0)}$ = 9b05688c

g = $H_6^{(i-1)} = H_6^{(0)}$ = 1f83d9ab

h = $H_7^{(i-1)} = H_7^{(0)}$ = 5be0cd19

c. Berikutnya lakukan iterasi

selanjutnya perhitungan untuk memperoleh nilai a, b, c, d, e, f, g, h

for t = 0 to 63 ;

{

T1 = h + $\sum_1^{(256)}(e)$ + Ch(e,f,g) + $K_0^{(256)}$ + wt

T2 = $\sum_0^{(256)}(a)$ + Maj (a,b,c)

h = g

g = f

f = e

e = d + T1

a = T1 + T2

$T1 = h + \sum_1^{(256)}(e) + Ch(e,f,g) + K_0^{(256)} + W_t$

Langkah pertama mencari nilai untuk $\sum_1^{(256)}(e)$ $\sum_0^{(256)}(a)$ pada T1 dan T2dan dan ubah nilai hexadesimal ke biner

$\sum_1^{(256)}(e)$ = RORT^6(e)⊕RORT^{11}(e)⊕RORT^{25}(e)

= RORT^6 01010001000011100101001001111111 ⊕

RORT^{11} 01010001000011100101001001111111 ⊕

RORT^{25} 01010001000011100101001001111111

=11111101010001000011100101001001⊕01001111111101010001000011100101 0⊕

10000111001010010011111110101000

$\sum_1^{(256)}(e)$ = 00110101100001110010011100101011

$\sum_0^{(256)}(a)$ = RORT^2(a)⊕RORT^{13}(a)⊕RORT^{22}(a)

= RORT^2 01101010000010011110011001100111 ⊕

RORT^{13} 01101010000010011110011001100111 ⊕

RORT^{22} 01101010000010011110011001100111 ⊕

=11011010100000100111100110011001⊕ 00110011001110110101000001001111⊕

00100111100110011001110110101000

$\sum_0^{(256)}(a)$ = 11001110001000001011010001111110

Ch(e,f,g) = (e ∧ f) xor (not e ∧ g)

= (01010001000011100101001001111111 ∧ 10011011000001010110100010001100) ⊕

(not 01010001000011100101001001111111 ∧ 00011111100000111101100110101011)

= (01010001000011100101001001111111 ∧ 10011011000001010110100010001100) ⊕

10101110111110001101011011000000 ∧ 00011111100000111101100110101011)

00010001000001000100000000001100 ⊕ 00001110100000011000100110000000

Ch(e,f,g) = 00011111100001011100100110001100

Maj (a,b,c) = (a ∧ b) ⊕ (a ∧ c) ⊕ (b ∧ c)

(01101010000010011110011001100111 and 10111011011001111010111010000101) ⊕

(01101010000010011110011001100111 and 00111100011011101111001101110010) ⊕

(10111011011001111010111010000101 and 00111100011011101111001101110010)

00101010000000001101001100000010 5 ⊕ 00101000000010001110001001100010 ⊕

00111000011001101010001000000000

Maj (a,b,c) = 00111010011011111110011001100111

Setelah masing masing fungsi sudah dapat nilainya,berikutnya hitung nilai T1 dan T2

$T1 = h + \sum_1^{(256)}(e) + Ch(e,f,g) + K_0^{(256)} + W_t$

$T1 = h + \sum_1^{(256)}(e) + Ch(e,f,g) + K_0^{(256)} + W_0$

T1 = 01011011111000001100110100011001 + 00110101100001110010011100101011 +

00011111110000101110010011000110 0 + 10000101000101000101111110011000 +

01100010011010110110010010000000

T1 = 01010101111000110101000111101000

$T2 = \sum_0^{(256)}(a)$ + Maj (a,b,c)

= 11001110001000001011010001111110 + 00111010011011111110011001100111

T2 = 00001000100100001001101011100101

Berikutnya beri nilai a,b,c,d,e,f,g,h dengan ketentuan rumus diatas tadi

h = g

g = f

f = e

e = d + T1

d = c

c = b

b = a

a = T1 + T2

a = T1 + T2

a = 01001010111100011010100011110100 0 + 00001000100100001001101011100101

a = 01011110011100111111011001100 1101

a = a

b = 01101010000010011110011001100111

b = b

c = 10111011011001111010111010000101

c = c

d = 00111100011011101111001101110010

d = d + T1

e = 00001010010101001111111110101001110 10 + 01010101111000110101000111101000

e = 11111011001100110100011100100010

f = e

f = 01010001000011100101001001111111

g = f

g = 10011011000001010110100010001100

h = g

h = 00011111110000011110110011010101 1

selanjutnya masing masing variabel diubah ke hexadesimal t = 0

menjadi

a = 5e73eccd b = 6a09e667 c = bb67ae85 d = 3c6ef372

e = fb334722 f = 510e527f g = 9b05688c h = 1f83d9ab

ulangi langkahnya hingga t = 63

maka hasilnya nanti dibuat tabel seperti dibawah ini

Tabel 3.1 Intermedinate Hash Value

	a	b	c	d	e	f	g	h
init:	6a09e667	bb67ae85	3c6ef372	a54ff53a	510e527f	9b05688c	1f83d9ab	5be0cd19
t= 0	5e73eccd	6a09e667	bb67ae85	3c6ef372	fb334722	510e527f	9b05688c	1f83d9ab
t= 1	507aef8d	5e73eccd	6a09e667	bb67ae85	1440b665	fb334722	510e527f	9b05688c
t= 2	38c35f1c	507aef8d	5e73eccd	6a09e667	d5e5115f	1440b665	fb334722	510e527f
t= 3	735d5c61	38c35f1c	507aef8d	5e73eccd	8945f1fd	d5e5115f	1440b665	fb334722
t= 4	fb334722	735d5c61	38c35f1c	507aef8d	93afcf64	8945f1fd	d5e5115f	1440b665
t= 5	ce259d60	704489e3	735d5c61	38c35f1c	2d2e01ef	93afcf64	8945f1fd	d5e5115f
t= 6	c126a275	ce259d60	704489e3	735d5c61	38a9cee4	2d2e01ef	93afcf64	8945f1fd
t= 7	b1a612e1	c126a275	ce259d60	704489e3	6b706f35	38a9cee4	2d2e01ef	93afcf64
t= 8	47731f9d	b1a612e1	c126a275	ce259d60	936187d1	6b706f35	38a9cee4	2d2e01ef
t= 9	9ac44b62	47731f9d	b1a612e1	c126a275	467b4c6b	936187d1	6b706f35	38a9cee4
t=10	2681fe64	9ac44b62	47731f9d	b1a612e1	673a3667	467b4c6b	936187d1	6b706f35
t=11	679de987	2681fe64	9ac44b62	47731f9d	1509c1f8	673a3667	467b4c6b	936187d1
t=12	b250b253	679de987	2681fe64	9ac44b62	f0c38b7a	1509c1f8	673a3667	467b4c6b
t=13	273f4a29	b250b253	679de987	2681fe64	5eb0aa6c	f0c38b7a	1509c1f8	673a3667
t=14	6106588	273f4a29	b250b253	679de987	1fc629fd	5eb0aa6c	f0c38b7a	1509c1f8
t=15	9051e8e1	6106588	273f4a29	b250b253	658d6666	1fc629fd	5eb0aa6c	f0c38b7a
t=16	5635a0cd	9051e8e1	6106588	273f4a29	9db96b81	658d6666	1fc629fd	5eb0aa6c
t=17	9cb4b44a	5635a0cd	9051e8e1	6106588	287d30e4	9db96b81	658d6666	1fc629fd
t=18	3d527172	9cb4b44a	5635a0cd	9051e8e1	14d6c4df	287d30e4	9db96b81	658d6666
t=19	38a8ff3e	3d527172	9cb4b44a	5635a0cd	903ccede	14d6c4df	287d30e4	9db96b81
t=20	17cd8844	38a8ff3e	3d527172	9cb4b44a	61373039	903ccede	14d6c4df	287d30e4
t=21	af3b1388	17cd8844	38a8ff3e	3d527172	9f50114c	61373039	903ccede	14d6c4df
t=22	a1a303b9	af3b1388	17cd8844	38a8ff3e	80d8ac0c	9f50114c	61373039	903ccede
t=23	4db4db60	a1a303b9	af3b1388	17cd8844	e507aba6	80d8ac0c	9f50114c	61373039
t=24	63546b5e	4db4db60	a1a303b9	af3b1388	b26c05b2	e507aba6	80d8ac0c	9f50114c
t=25	a1aa6ac1	63546b5e	4db4db60	a1a303b9	5ba5b8d8	b26c05b2	e507aba6	80d8ac0c
t=26	1b18775d	a1aa6ac1	63546b5e	4db4db60	c33a7e71	5ba5b8d8	b26c05b2	e507aba6
t=27	5460bd51	1b18775d	a1aa6ac1	63546b5e	e1097bdc	c33a7e71	5ba5b8d8	b26c05b2
t=28	ba9b48c6	ba9b48c6	1b18775d	a1aa6ac1	cf5f6bd3	e1097bdc	c33a7e71	5ba5b8d8
t=29	bf90a2a4	ba9b48c6	ba9b48c6	1b18775d	c172740f	cf5f6bd3	e1097bdc	c33a7e71
t=30	83e6b1b3	bf90a2a4	ba9b48c6	ba9b48c6	6823397a	c172740f	cf5f6bd3	e1097bdc
t=31	df577154	83e6b1b3	bf90a2a4	ba9b48c6	808210a9	6823397a	c172740f	cf5f6bd3
t=32	60172792	df577154	83e6b1b3	bf90a2a4	9a254911	808210a9	6823397a	c172740f
t=33	67dbfee0	60172792	df577154	83e6b1b3	6c0cef15	9a254911	808210a9	6823397a
t=34	dbb314dd	67dbfee0	60172792	df577154	77340dc8	6c0cef15	9a254911	808210a9
t=35	59b7725c	dbb314dd	67dbfee0	60172792	b92a3d1f	77340dc8	6c0cef15	9a254911
t=36	7f716517	59b7725c	dbb314dd	67dbfee0	2a8eb583	b92a3d1f	77340dc8	6c0cef15
t=37	f204d4e9	7f716517	59b7725c	dbb314dd	cb396039	2a8eb583	b92a3d1f	7340dc8
t=38	1ba65a40	f204d4e9	7f716517	59b7725c	b386f7ec	cb396039	2a8eb583	b92a3d1f
t=39	302bbf4c	1ba65a40	f204d4e9	7f716517	c13e919b	b386f7ec	cb396039	2a8eb583
t=40	68eac792	302bbf4c	1ba65a40	f204d4e9	5d9ef013	c13e919b	b386f7ec	cb396039
t=41	a2426c1b	68eac792	302bbf4c	1ba65a40	4de4a2b3	5d9ef013	c13e919b	b386f7ec
t=42	5044463f	a2426c1b	68eac792	302bbf4c	c981c9c9	4de4a2b3	5d9ef013	c13e919b

t=43	e59a9147	5044463f	a2426c1b	68eac792	c0909b8c	c981c9c9	4de4a2b3	5d9ef013
t=44	961f7f00	e59a9147	5044463f	a2426c1b	5ab6b60	c0909b8c	c981c9c9	4de4a2b3
t=45	4d605ef1	961f7f00	e59a9147	5044463f	7b0506a2	5ab6b60	c0909b8c	c981c9c9
t=46	37058c8e	4d605ef1	961f7f00	e59a9147	9c85ba01	7b0506a2	5ab6b60	c0909b8c
t=47	58d459ce	37058c8e	4d605ef1	961f7f00	27e6a8c2	9c85ba01	7b0506a2	5ab6b60
t=48	e8561251	58d459ce	37058c8e	4d605ef1	18104ad1	27e6a8c2	9c85ba01	7b0506a2
t=49	c5946448	e8561251	58d459ce	37058c8e	e9cd28e6	18104ad1	27e6a8c2	9c85ba01
t=50	dbc1973d	c5946448	e8561251	58d459ce	740bcdc	e9cd28e6	18104ad1	27e6a8c2
t=51	63dd8a20	dbc1973d	c5946448	e8561251	299b80e9	740bcdc	e9cd28e6	18104ad1
t=52	9ccf3830	63dd8a20	dbc1973d	c5946448	8172c66e	299b80e9	740bcdc	e9cd28e6
t=53	68d38cd6	9ccf3830	63dd8a20	dbc1973d	78426ce8	8172c66e	299b80e9	740bcdc
t=54	e3b449e	68d38cd6	9ccf3830	63dd8a20	ce6857a1	78426ce8	8172c66e	299b80e9
t=55	fcb54b22	e3b449e	68d38cd6	9ccf3830	94aefe7	ce6857a1	78426ce8	8172c66e
t=56	b3f97809	fcb54b22	e3b449e	68d38cd6	b0ff2713	94aefe7	ce6857a1	78426ce8
t=57	28b5cd95	b3f97809	fcb54b22	e3b449e	897c2b5b	b0ff2713	94aefe7	ce6857a1
t=58	15436dfa	28b5cd95	b3f97809	fcb54b22	7917072e	897c2b5b	b0ff2713	94aefe7
t=59	1c34ca5a	15436dfa	28b5cd95	b3f97809	ffc10eb3	7917072e	897c2b5b	b0ff2713
t=60	21e57a42	1c34ca5a	15436dfa	28b5cd95	b2b46931	ffc10eb3	7917072e	897c2b5b
t=61	51423acf	21e57a42	1c34ca5a	15436dfa	951144ce	b2b46931	ffc10eb3	7917072e
t=62	7ff1afe8	51423acf	21e57a42	1c34ca5a	790f69b1	951144ce	b2b46931	ffc10eb3
t=63	1ab62384	7ff1afe8	51423acf	21e57a42	1f097d8c	790f69b1	951144ce	b2b46931

4. Menghitung hasil akhir dengan menjumlah semua variabel dan nilai awal *hash* H_i
   Berikutnya lakukan penjumlahan computer intermediate hash value + initial hash value

$H_0^{(i)} = a + H_0^{(i-1)} = 1ab62384 + 6a09e667 = 84c009eb$

$H_1^{(i)} = b + H_1^{(i-1)} = 7ff1afe8 + bb67ae85 = 3b595e6d$

$H_2^{(i)} = c + H_2^{(i-1)} = 51423acf + 3c6ef372 = 8db12e41$

$H_3^{(i)} = d + H_3^{(i-1)} = 21e57a42 + a54ff53a = c7356f7c$

$H_4^{(i)} = e + H_4^{(i-1)} = 1f097d8c + 510e527f = 7017d00b$

$H_5^{(i)} = f + H_5^{(i-1)} = 790f69b1 + 9b05688c = 1414d23d$

$H_6^{(i)} = g + H_6^{(i-1)} = 951144ce + 1f83d9ab = b4951e79$

$H_7^{(i)} = h + H_7^{(i-1)} = b2b46931 + 5be0cd19 = 0e95364a$

3. hasil mesege digest

$H_0^{(n)} \| H_1^{(n)} \| H_2^{(n)} \| H_3^{(n)} \| H_4^{(n)} \| H_5^{(n)} \| H_6^{(n)} \| H_7^{(n)} \|$

84c009eb 3b595e6d 8db12e41 c7356f7c 7017d00b 1414d23d b4951e79 0e95364a

3.4.8 Perancangan Algoritma RSA

Langkah pertama bangkitkan kunci pada algoritma Rsa dengan langkah sebagai berikut :

1. Bangkitkan 2 bilangan prima p dan q
   p = 11 dan q = 17

2. Hitung n = p x q = 11 x 17 = 187
3. Hitung $\phi(n) = (p-1)(q-1) = 10 \times 16 = 160$
4. Pilih kunci publik e, yang relatif prima terhadap ϕ (n). e = 7
5. Bangkitkan kunci privat dengan menggunakan persamaan $d = \frac{1+k\phi(n)}{e}$

$d = \frac{1+(k\ X\ 160)}{7}$ = 23. nilai k didapatkan nilai d yg bulat

Maka diperoleh n = 187, e = 7 dan d 23

Proses enkripsi

Misal pesan yg dienkripsi meiliki nilai hash yg sudah kita lakukan diatas tadi yaitu

"84C009EB3B595E6D8DB12E41C7356F7C7017D00B1414D23DB4951E790E95364A"

Kemudian ubah setiap karakter menggunakan tabel ASCII menjadi desimal

56 52 67 48 48 57 69 66 51 66 53 57 53 69 54 68

56 68 66 49 50 69 52 49 67 55 51 53 54 70 55 67

55 48 49 55 68 48 48 66 49 52 49 52 68 50 51 68

66 52 57 53 49 69 55 57 48 69 57 53 51 54 52 65"

Selanjunya ubah ke chipertext dengan rumus

$Ci = Mi^e\ Mod\ n$

56^7 mod 187 =78	56^7 mod 187 =78	55^7 mod 187 =132	66^7 mod 187 = 110
52^7 mod 187 =35	68^7 mod 187 = 51	48^7 mod 187 = 159	52^7 mod 187 = 35
67^7 mod 187 =67	66^7 mod 187 = 110	49^7 mod 187 = 25	57^7 mod 187 = 150
48^7 mod 187 =159	49^7 mod 187 = 25	55^7 mod 187 = 132	53^7 mod 187 = 26
48^7 mod 187 =159	50^7 mod 187 = 118	68^7 mod 187 = 51	49^7 mod 187 = 25
57^7 mod 187 =150	69^7 mod 187 = 86	48^7 mod 187 = 159	69^7 mod 187 = 86
69^7 mod 187 =86	52^7 mod 187 = 35	48^7 mod 187 = 159	55^7 mod 187 = 132
66^7 mod 187 =110	49^7 mod 187 = 25	66^7 mod 187 = 110	57^7 mod 187 = 150
51^7 mod 187 =17	67^7 mod 187 = 67	49^7 mod 187 = 25	48^7 mod 187 = 159
66^7 mod 187 =110	55^7 mod 187 =132	52^7 mod 187 = 35	69^7 mod 187 = 86

$53^7 \bmod 187 = 26$ $51^7 \bmod 187 = 17$ $49^7 \bmod 187 = 25$ $57^7 \bmod 187 = 150$

$57^7 \bmod 187 = 150$ $53^7 \bmod 187 = 26$ $52^7 \bmod 187 = 35$ $53^7 \bmod 187 = 26$

$53^7 \bmod 187 = 26$ $54^7 \bmod 187 = 164$ $68^7 \bmod 187 = 51$ $51^7 \bmod 187 = 17$

$69^7 \bmod 187 = 86$ $70^7 \bmod 187 = 60$ $50^7 \bmod 187 = 118$ $54^7 \bmod 187 = 164$

$54^7 \bmod 187 = 164$ $55^7 \bmod 187 = 132$ $51^7 \bmod 187 = 17$ $52^7 \bmod 187 = 35$

$68^7 \bmod 187 = 51$ $67^7 \bmod 187 = 67$ $68^7 \bmod 187 = 51$ $65^7 \bmod 187 = 142$

Sehingga didapatkan chipertext :

"78356715915915086110861101711026150268616451785111025118863525671321726164601326713215925132511591591102535253551118175111035150262586132150159861502617164351 42"

3.5 Pengujian Sistem

Setelah menyelesaikan pengembangan aplikasi, peneliti melakukan pengujian awal untuk mengevaluasi respons aplikasi terhadap berbagai input dan skenario. Pengujian ini dilakukan dengan mengikuti tahapan pengujian black box yang diuraikan dalam tabel 3.7.

Tabel 3.2 Pengujian sistem

No	Skenario Pengujian	Hasil Yang diharapkan	Hasil Pengujian	Kesimpulan
1.	Login menggunakan username dan pasword yang benar	Menampilkan beranda		
2.	Login menggunakan username dan pasword yang salah	Meanampilkan pop up terjadi kesalahan		
3.	Mengklik tombol Logout	Kembali ke menu login		
4.	Mengklik tombol ajukan dokumen bagi pengguna	Manampilkan pop up dokumen yg akan diunpload		
5.	Pengguna mengunduh file yg sudah disetujui	File barhasil diunduh		
6.	Memilih tombol file dokumen pada menu pengesahan dokumen	Penandatangan dapat melhat dokumen yg akan ditandatangani		
7.	Penandatangan mengklik tombol buka editor pada menu dokumen saya	Penandatangan dapat menambahkan tanda tangan		

BAB V
KESIMPULAN DAN SARAN

5.1 Kesimpulan

Pembuatan sistem aplikasi tanda tangan digital yang dibangun dengan menggunakan algoritma RSA telah berhasil diterapkan pada website.Implementasi sistem ini juga berhasil melakukan proses pengesahan dokumen, memungkinkan pengguna untuk mengunggah, penandatangan untuk menandatangani dokumen, dan user untuk mengunduh dokumen yang telah disetujui.

5.2 Saran

Berdasarkan penelitian ini, peneliti menyarankan agar aplikasi ini dikembangkan lebih lanjut menjadi aplikasi Android. Pengembangan ini akan memberikan fleksibilitas dan aksesibilitas yang lebih besar, memungkinkan pengguna untuk menandatangani dokumen dari mana saja dan kapan saja menggunakan perangkat *mobile*.

DAFTAR PUSTAKA

[1] S. Habibah, "Implikasi Filsafat Ilmu terhadap Perkembangan Ilmu Pengetahuan dan Teknologi", Dar el-Ilmi : jurnal studi keagamaan, pendidikan dan humaniora, vol. 4, no. 1, hal. 166-180, Apr. 2017

[2] Badan Pusat Statistik 2021, *indeks Pembangunan Teknologi Informasi dan Komunikasi 2021*, katalog no. 8305012, jakarta,BPS

[3] F. Nurhasanah and R. Sulaiman, "Pembuatan tanda tangan digital menggunakan digital signature algorithm," MATHunesa J. Ilm. Mat., vol. 2, no. 2, hal. 1–7, May 2013

[4] A. R. Tulloh, Y. Permanasari, and E. Harahap, "Kriptografi Advanced Encryption Standard (AES) Untuk Penyandian File Dokumen," Jurnal Matematika UNISBA, vol. 15, no. 1, 2016.

[5] S. . and P. Atika, "DIGITAL SIGNATURE DENGAN ALGORITMA SHA-1 DAN RSA SEBAGAI AUTENTIKASI", Jurnal Cendikia, vol. 16, no. 2 Oktober, hal. 74-83, Oct. 2018.

[6] D. Puspitasari and Y. Permanasari, "Kriptografi Rivest Shamir Adleman (RSA) pada Tanda Tangan Digital," in Prosiding Matematika, vol. 6, no. 1, hal. 14–20, 2020.

[7] R. A. Azdy, "Tanda tangan Digital Menggunakan Algoritme Keccak dan RSA," J. Nas. Tek. Elektro dan Teknol. Inf., vol. 5, no. 3, hal. 184–191, 2016.

[8] D. Precilia and A. Izzuddin, "Aplikasi Tanda Tangan Digital (Digital Signature) Menggunakan Algoritma Message Digest 5 (MD5)", energy, vol. 5, no. 1, hal. 14 -19, May 2015.

[9] Hendri, D. Rositawati, and P. Romansyah, "Model digital signature pada dokumen formal akademik", Cyberpreneurship innovative and creative axact and social science, vol. 6, no. 1, hal. 22 -32, 2020.

[10] A. B. Jati. T. A. P. Sidhi, and J. E. Samodra, "Pembangunan Sistem Tanda Tangan Digital pada Sistem Informasi Universitas Atma Jaya Yogyakarta", Jurnal Informatika Atma Jogja, Vol. 2, no. 2, Nov. 2021.

[11] Arisandi, D., Yusuf, M., & Sukri, S. (2020)."Pemeriksaan Integritas Dokumen Dengan Digital Signature Algorithm. Joisie",*Journal Of Information Systems And Informatics Engineering*, Vol 4, no 1, hal.1-6, Juni. 2020.

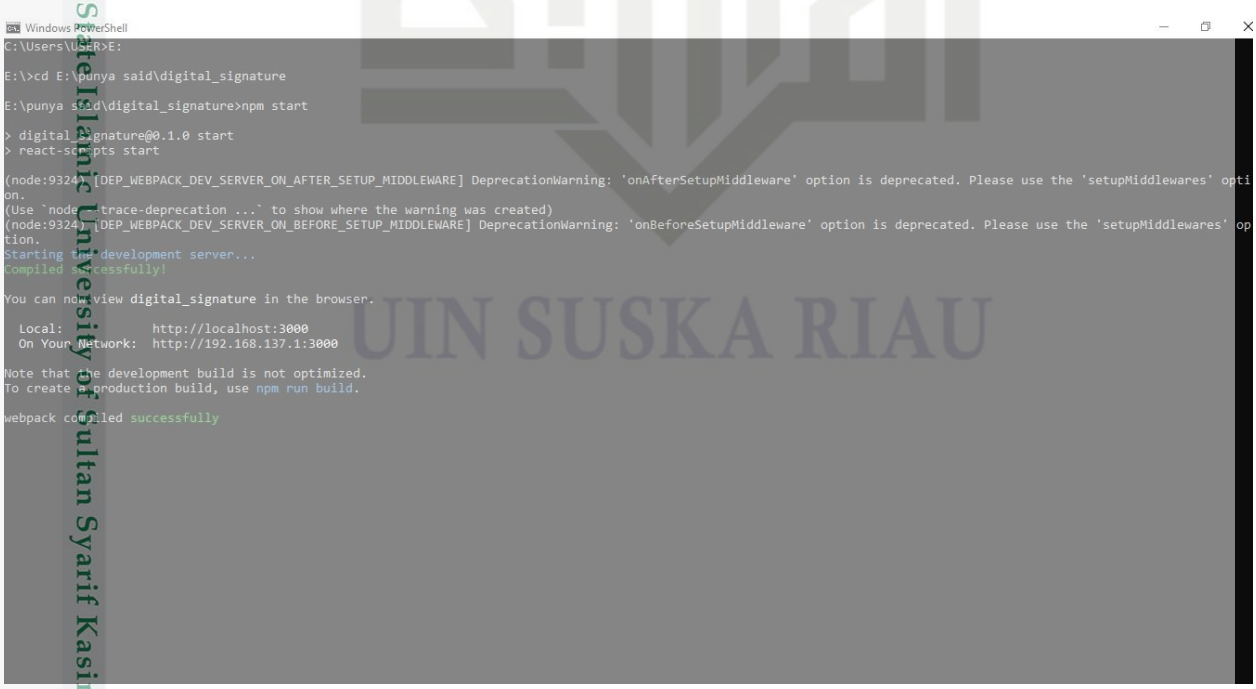
[12] R. Munir, Kriptografi, 2rd ed. Bandung : Penerbit Informatika, 2019.

[13] D. Ariyus, Pengantar ilmu kriptografi : Teori, analisis dan implementasi, Yogyakarta : Penerbit Andi, 2008.

[14] S. Kromodimoeljo, Teori dan Aplikasi Kriptografi, jakarta : Spk It Consulting, 2009.

Lampiran

menjalankan server node.js



Source code :

utilities.js

```
const forge = require('node-forge');
const fs = require('fs');
const crypto = require('crypto');
const { join } = require('path')


const create_document = async (model, body = { content: "default" }) => {
    const Create_document = await model.insertMany([body])
    return Create_document
}


const read_document = async (model, where = { field: "default" }, exclude = {}, limit = 1) => {
    if (where.field === "default") {
        const Read_document = await model.find({}, exclude).limit(limit)
        return Read_document
    } else {
        const Read_document = await model.find(where, exclude).limit(limit)
        return Read_document
    }
}


const update_document = async (model, where = { field: "dafault" }, body = { content: "default" }) => {
    const Update_document = await model.updateOne(where, { $set: body })
```

```
    return Update_document

}

const delete_document = async (model, where = { field: "default" }) => {
    const Delete_document = await model.remove(where)
    return Delete_document

}

const count_document = async (model, where = {}) => {
    const Count_document = await model.countDocuments(where)
    return Count_document

}

const generateKeyPair = () => {
    const { privateKey, publicKey } = forge.pki.rsa.generateKeyPair(2048);
    fs.writeFileSync(join(__dirname, '/keys/privateKey.pem'),
forge.pki.privateKeyToPem(privateKey));
    fs.writeFileSync(join(__dirname, '/keys/publicKey.pem'),
forge.pki.publicKeyToPem(publicKey));
    console.log('RSA key pair generated and saved to keys/ directory.');
}

const encryptPDF = (inputFilePath, outputFilePath, publicKeyPath) => {
    try {
        const publicKeyPem = fs.readFileSync(publicKeyPath, 'utf8');
```

```
    const publicKey = forge.pki.publicKeyFromPem(publicKeyPem);
    const pdfBuffer = fs.readFileSync(inputFilePath);
    const encryptedBuffers = [];
    const chunkSize = 190; // 190 bytes to leave space for padding
    for (let i = 0; i < pdfBuffer.length; i += chunkSize) {
      const chunk = pdfBuffer.slice(i, i + chunkSize);
      const encryptedChunk = publicKey.encrypt(chunk.toString('binary'), 'RSA-OAEP');
      encryptedBuffers.push(Buffer.from(encryptedChunk, 'binary'));
    }
    const encryptedPDF = Buffer.concat(encryptedBuffers);
    fs.writeFileSync(outputFilePath, encryptedPDF);
    console.log('PDF file encrypted and saved.');
    return true
  } catch (error) {
    console.log(`error when encryption process => ${error}`)
    return false
  }
}

const decryptPDF = (inputFilePath, outputFilePath, privateKeyPath) => {
  try {
    const privateKeyPem = fs.readFileSync(privateKeyPath, 'utf8');
    const privateKey = forge.pki.privateKeyFromPem(privateKeyPem);
```

```
    const encryptedPDFBuffer = fs.readFileSync(inputFilePath);
    const decryptedBuffers = [];
    const chunkSize = 256; // 256 bytes for 2048-bit RSA
    for (let i = 0; i < encryptedPDFBuffer.length; i += chunkSize) {
      const chunk = encryptedPDFBuffer.slice(i, i + chunkSize);
      const decryptedChunk = privateKey.decrypt(chunk.toString('binary'),
'RSA-OAEP');
      decryptedBuffers.push(Buffer.from(decryptedChunk, 'binary'));
    }
    const decryptedPDF = Buffer.concat(decryptedBuffers);
    fs.writeFileSync(outputFilePath, decryptedPDF);
    console.log('PDF file decrypted and saved.');
    return true
  } catch (error) {
    console.log(`error when decryption process => ${error}`)
    return false
  }


module.exports = {
  create_document,
  read_document,
  update_document,
  delete_document,
  count_document,
```

DAFTAR RIWAYAT HIDUP

Said Rikzan,lahir di Desa Kasikan Tapung Hulu pada tanggal 07 juli 1998.Merupakan anak kedua dari dua bersaudara dari pasangan Said Riduan dan Karmila. Adapun pengalaman pendidikan yang ditempuh dimulai di SDN 001 kasikan, kemudian dilanjutkan di MTSS LKMD KASIKAN, dan dilanjutkan di SMAN 1 Tapung Hulu.setelah lulus SMA penulis melanjutkan pendidikan di Universitas Islam Negeri Sultan Syarif Kasim Riau, Fakultas Sains dan Teknologi, program studi Teknik Elektro dengan mengambil konsentrasi Komputer dan Multimedia. Penulis menyelesaikan studi dengan menyelesaikan tugas akhir yang berjudul SISTEM APLIKASI TANDA TANGAN DIGITAL PADA WEBSITE ALGORITMA RSA.

Email : saidrikzan1@gmail.com