TUGAS AKHIR – IF184802

PERANCANGAN DAN IMPLEMENTASI TANDA TANGAN DIGITAL DI LINGKUNGAN ITS MENGGUNAKAN DATA TERENKRIPSI

CLAUS PETER CHRISTO HUTABARAT
NRP 05111740000090

Dosen Pembimbing I
MUCHAMMAD HUSNI, M.Kom

Dosen Pembimbing II
WAHYU SUADI, S.Kom, M.Kom.

DEPARTEMEN TEKNIK INFORMATIKA
Fakultas Teknologi Elektro dan Informatika Cerdas
Institut Teknologi Sepuluh Nopember
Surabaya 2021

TUGAS AKHIR – IF184802

PERANCANGAN DAN IMPLEMENTASI TANDA TANGAN DIGITAL DI LINGKUNGAN ITS MENGGUNAKAN DATA TERENKRIPSI

CLAUS PETER CHRISTO HUTABARAT
NRP 05111740000090

Dosen Pembimbing I
MUCHAMMAD HUSNI, M.Kom

Dosen Pembimbing II
WAHYU SUADI, S.Kom, M.Kom.

DEPARTEMEN TEKNIK INFORMATIKA
Fakultas Teknologi Elektro dan Informatika Cerdas
Institut Teknologi Sepuluh Nopember
Surabaya 2021

(Halaman ini sengaja dikosongkan)

UNDERGRADUATE THESIS – IF184802

DESIGN AND IMPLEMENATION OF DIGITAL SIGNATURE IN ITS ENVIRONMENT USING ENCRYPTED DATA

CLAUS PETER CHRISTO HUTABARAT
NRP 05111740000090

Main Supervisor
MUCHAMMAD HUSNI, M.Kom

Co-Supervisor
WAHYU SUADI, S.Kom, M.Kom.

DEPARTEMEN TEKNIK INFORMATIKA
Fakultas Teknologi Elektro dan Informatika Cerdas
Institut Teknologi Sepuluh Nopember
Surabaya 2021

(Halaman ini sengaja dikosongkan)

LEMBAR PENGESAHAN

PERANCANGAN DAN IMPLEMENTASI TANDA TANGAN DIGITAL DI LINGKUNGAN ITS MENGGUNAKAN DATA TERENKRIPSI

TUGAS AKHIR

Diajukan untuk Memenuhi Salah Satu Syarat
Memperoleh Gelar Sarjana Komputer
pada
Bidang Studi Komputasi Berbasis Jaringan
Program Studi S-1 Departemen Teknik Informatika
Fakultas Teknologi Elektro Dan Informatika Cerdas
Institut Teknologi Sepuluh Nopember

Oleh:
CLAUS PETER CHRISTO HUTABARAT
NRP 05111740000090

Disetujui oleh Pembimbing tugas akhir:

1. Muchammad Husni, M.Kom ……
   (NIP. 196002211984031001) (Pembimbing1)

2. Wahyu Suadi, S.Kom, M.Kom. ……
   (NIP. 197110302002121001) (Pembimbing2)

SURABAYA
JUNI, 2021

(Halaman ini sengaja dikosongkan)

PERANCANGAN DAN IMPLEMENTASI TANDA TANGAN DIGITAL DI LINGKUNGAN ITS MENGGUNAKAN DATA TERENKRIPSI

Nama Mahasiswa	**:**	**Claus Peter Christo Hutabarat**
NRP	**:**	**05111740000090**
Departemen	**:**	**Teknik Informatika FTEIC ITS**
Dosen Pembimbing 1	**:**	**Ir. MUCHAMMAD HUSNI, M.Kom.**
Dosen Pembimbing 2	**:**	**WAHYU SUADI, S.Kom., M.M., M.Kom.**

Abstrak

Tanda tangan digital atau *Digital Signature* adalah bentuk tanda tangan yang dibuat oleh sesorang secara digital dan yang membedakannya dengan tanda tangan biasa adalah keterlibatan sebuah algoritma untuk menandatangani dan memvalidasi keaslian tanda tangan tersebut. Tanda tangan digital telah berkembang dan diimplementasikan menggunakan algoritma yang berbeda beda.

Pada sistem Tanda Tangan Digital yang akan di buat, diharapkan sistem ini dapat membantu proses tanda tangan yang tradisional menjadi online tanpa perlu berinteraksi dengan yang bersangkutan dan tetap terjamin keasliannya.

Dalam proses perancangan sistem tersebut diharapkan dapat menjaga keutuhan sebuah tanda tangan digital dan dapat diuji kebenaranannya.

Hasil dari implementasi tanda tangan digital ini dapat memudahkan proses tanda tangan sesorang dengan menghemat unsur waktu dan jarak, sehingga proses tatap muka tidak perlu dilakukan.

Kata kunci: ***Database, Digital Signature, Public Key, Security***

DESIGN AND IMPLEMENATION OF DIGITAL SIGNATURE IN *ITS* ENVIRONMENT USING ENCRYPTED DATA

Student's Name : Claus Pieter Christo Hutabarat
Student's ID : 05111740000090
Department : Teknik Informatika FTEIC-ITS
First Advisor : Ir. MUCHAMMAD HUSNI, M.Kom.
Second Advisor : WAHYU SUADI, S.Kom., M.M., M.Kom.

Abstract

A Digital Signature is a form of signature that is digitally made. What distinguishes it from a regular signature is the involvement of an algorithm to sign and validate the authenticity of the signature. Digital signatures have been developed and implemented using different algorithms.

In the Digital Signature system that will be created, it is hoped that this system can help the traditional signature process go online without the need to interact with the person concerned and still guarantee its authenticity.

In the process of designing the system, it is hoped that it can maintain the integrity of a digital signature and can be verified.

The results of the implementation of this digital signature can facilitate the process of someone's signature by saving the elements of time and distance, so that the face-to-face process does not need to be done.

Keywords: ***Database, Internet of things, Resource, Round Time Trip, Throughput.***

KATA PENGANTAR

Puji syukur kepada Tuhan Yesus Kristus atas segala kasih dan anugrahNya sehingga penulis dapat menyelesaikan tugas akhir yang berjudul

PERANCANGAN DAN IMPLEMENTASI TANDA TANGAN DIGITAL DI LINGKUNGAN ITS MENGGUNAKAN DATA TERENKRIPSI Harapan dari penulis, semoga apa yang tertulis di dalam buku tugas akhir ini dapat bermanfaat bagi pengembangan ilmu pengetahuan saat ini dan ke depannya, serta dapat memberikan kontribusi yang nyata.

Dalam pelaksanaan dan pembuatan tugas akhir ini tentunya sangat banyak bantuan yang penulis terima dari berbagai pihak, tanpa mengurangi rasa hormat penulis ingin mengucapkan terima kasih sebesar-besarnya kepada:

1. Tuhan Yesus, yang telah membimbing penulis selama hidup.
2. Keluarga penulis (Papa, Mama, dan keluarga penulis yang lain) yang selalu memberikan dukungan baik berupa doa, moral, dan material yang tak terhingga kepada penulis, sehingga penulis dapat menyelesaikan tugas akhir ini.
3. Bapak Muchammad Husni, M.Kom dan bapak Wahyu SuadI, S.Kom, M.Kom. selaku Dosen Pembimbing penulis yang telah membimbing, memberikan nasihat, dan memotivasi penulis sehingga penulis dapat menyelesaikan tugas akhir ini.
4. Ibu Dr.Eng. Chastine Fatichah, S.Kom., M.Kom. selaku kepala Departemen Teknik Informatika ITS.
5. Bapak dan Ibu Dosen yang telah memberikan ilmunya selama penulis berkuliah di Informatika ITS.
6. Sahabat baik penulis yaitu Nandha, Grup Budak Mekdi, Grup Produktif dan Kampus Kota, yang selalu support dan mendukung penulis selama bertahun-tahun.

7. Annas dan Siraj yan telah memberikan semangat pada penulis pada saat penulisan buku.
8. Teman-teman angkatan 2017 yang sudah menjadi saksi hidup perjalanan karir penulis selama berkuliah di Teknik Informatika ITS.
9. Support system penulis yaitu Emy yang turut membantu dan memberikan moral support.
10. Untuk orang-orang yang tidak dapat disebutkan satu persatu oleh penulis dan pembaca buku tugas akhir ini.

Penulis telah berusaha sebaik-baiknya dalam menyusun tugas akhir ini. Namun, penulis memohon maaf apabila terdapat kekurangan, kesalahan maupun kelalaian yang telah penulis lakukan. Kritik dan saran yang membangun dapat disampaikan sebagai bahan perbaikan selanjutnya.

Surabaya, 28 Juni 2021
Claus Pieter Christo Hutabarat

(Halaman ini sengaja dikosongkan)

DAFTAR ISI

LEMBAR PENGESAHAN 5
Abstrak 7
Abstract 8
KATA PENGANTAR 9
BAB I PENDAHULUAN 23
1.1. Latar Belakang 23
1.2. Rumusan Permasalahan 24
1.3 Batasan Masalah 24
1.4 Tujuan 25
1.5 Manfaat 25
1.6 Tahapan Tugas Akhir 25
1.6.1 Penyusunan Proposal Tugas akhir 25
1.6.2 Studi Literatur 26
1.6.3 Perancangan Sistem 26
1.6.4 Implementasi Sistem 26
1.6.5 Pengujian dan evaluasi 26
1.6.6 Penyusunan buku Tugas Akhir 27
1.7 Sistematika Penulisan Laporan Tugas Akhir 27
BAB II TINJAUAN PUSTAKA 30
2.1 *Encryption* 30
2.2 *EdDSA* 31
2.3 ChaCha20-Poly1305 31
2.4 *Digital Signature* 32

2.5 *Public Key Infrastructure* 33
2.6 *Certificate Authority* 35
2.7 *Advanced Encryption Standard* 36
2.8 Laravel 37
2.9 MySQL 37
2.10 PHP 38
2.11 QR Code 38
2.12 Google Map Javascript API 38
2.13 SSL, TLS, HTTPS 38
BAB III PERANCANGAN SISTEM 40
3.1 Deskripsi Umum 40
3.2 Daftar Istilah 41
3.3 Arsitektur Sistem 42
3.3.1 Perancangan Sistem 42
3.3.2 Perancangan Proses Registrasi 45
3.3.3 Perancangan Proses Enkripsi 46
3.3.4 Perancangan Proses Dekripsi 48
3.3.5 Perancangan Proses Verifikasi 49
3.3.5 Perancangan Basis Data 50
3.4 *Use Case Diagram* 50
BAB IV IMPLEMENTASI 52
4.1 Lingkungan Implementasi 52
4.1.1 Lingkungan Implementasi Perangkat Keras . 52
4.1.2 Lingkungan Implementasi Perangkat Lunak 53
4.2 Implementasi Web Sistem Tanda Tangan Online . 53

4.2.1 Implementasi Halaman Login/Register..... 54
4.2.2 Implementasi Halaman Utama..... 55
4.2.3 Implementasi Halaman Pembuatan Tanda Tangan 56
4.2.3 Implementasi Halaman Dekripsi..... 58
4.2.5 Implementasi Halaman Verifikasi..... 58
4.2.6 Implementasi Halaman Informasi Tanda Tangan 59
4.2.7 Implementasi Halaman Pencarian Tanda Tangan 61
4.3 Implementasi *Code* Program Web Sistem Tanda Tangan Online..... 61
4.3.1 Implementasi Proses Registrasi..... 62
4.3.2 Implementasi Proses Enkripsi..... 62
4.3.3 Implementasi Proses Dekripsi..... 64
4.3.4 Implementasi Proses Verifikasi 64
4.3.5.1 Implementasi Penggunaan Google Map Javascript API 65
4.3.5.2 Implementasi Penggunaan DOMPPDF..... 66
4.3.5.3 Implementasi Penggunaan Minisign..... 66
4.3.5.4 Implementasi Penggunaan Simple QR-Code 67
BAB V UJI COBA DAN EVALUASI 68
5.1 Lingkungan Pengujian 68
5.2 Data Pengujian..... 69
5.3 Skenario Uji Coba Fungsionalitas..... 69

5.3.1 Skenario Uji Coba (UJ-F01) – Mendapatkan Tanda Tangan Dalam Bentuk Enkripsi 69

5.3.2 Skenario Uji Coba (UJ-F02) – Mendapatkan Tanda Tangan Dalam Bentuk Dekripsi 73

5.3.3 Skenario Uji Coba (UJ-F03) – Memverifikasi Tanda Tangan User 74

5.3.4 Skenario Uji Coba (UJ-F04) – Scan *QrCode* Tanda Tangan..... 76

5.4 Evaluasi Hasil Uji Coba 78

BAB VI KESIMPULAN DAN SARAN 79

6.1 Kesimpulan 79

6.2 Saran 80

DAFTAR PUSTAKA 81

BIODATA PENULIS 83

(Halaman ini sengaja dikosongkan)

DAFTAR GAMBAR

Gambar 2. 1 Algo ChaCha 32
Gambar 2. 2 Alur PKI (Public key Infrastructure)[14] 34
Gambar 2. 3 Alur Certificate Authority (CA)[13] 35
Gambar 2. 4 Alur Advance Encryption Standar (AES)[15] . 36

Gambar 3. 1 Flow system secara keseluruhan 44
Gambar 3. 2 Perancangan alur registrasi 45
Gambar 3. 3 Perancangan alur enkripsi 47
Gambar 3. 4 Perancangan sistem dekripsi 48
Gambar 3. 5 Perancangan sistem verifikasi 49
Gambar 3. 6 Tahap perancangan basis data 50
Gambar 3. 7 Use case diagram..... 51

Gambar 4. 1 Tampilan halaman login/register..... 54
Gambar 4. 2 Tampilan halaman login/register..... 55
Gambar 4. 3 Tampilan halaman utama 55
Gambar 4. 4 Tampilan halaman pembuatan tanda tangan 57
Gambar 4. 5 Tampilan halaman dekripsi 58
Gambar 4. 6 Tampilan halaman verifikasi..... 59
Gambar 4. 7 Tampilan halaman informasi tanda tangan 60

Gambar 5. 1 Tanda tangan yang ditampilkan 71
Gambar 5. 2 Data png dalam bentuk base64 encoding..... 71
Gambar 5. 3 Monitoring https pada wireshark 72
Gambar 5. 4 Database sistem..... 72
Gambar 5. 5 Hasil tanda tangan yang diterima user 73
Gambar 5. 6 Hasil tanda tangan..... 74
Gambar 5. 7 Tanda tangan terverifikasi..... 76
Gambar 5. 8 Tanda tangan tidak terverifikasi..... 76

(Halaman ini sengaja dikosongkan)

DAFTAR TABEL

Tabel 3. 1 Daftar istilah 41
Tabel 3. 2 Beberapa subproses dalam perancangan sistem... 42

Tabel 4. 1 Detail perangkat keras yang digunakan 52
Tabel 4. 2 Perangkat lunak yang dibutuhkan..... 53

Tabel 5. 1 Spesifikasi perangkat keras dan perangkat lunak 68
Tabel 5. 2 Skenario uji coba 69
Tabel 5. 3 Skenario uji coba mendapatkan tanda tangan dalam bentuk terdekripsi..... 73
Tabel 5. 4 Skenario uji coba memverifikasi tanda tangan user 75
Tabel 5. 5 Skenario uji coba scan QR code tanda tangan 76
Tabel 5. 6 Hasil uji coba 78

(Halaman ini sengaja dikosongkan)

DAFTAR PSEUDOCODE

Pseudocode 4. 1 Halaman Pencarian Tanda Tangan 61
Pseudocode 4. 2 Proses registrasi 62
Pseudocode 4. 3 Proses enkripsi 63
Pseudocode 4. 4 Proses dekripsi 64
Pseudocode 4. 5 Proses verifikasi 65

Code 4. 1 Penggunaaan Google Map Javascript API 66
Code 4. 2 Penggunaan DOMPPDF..... 66
Code 4. 3 Penggunaan simple QR code..... 67

(Halaman ini sengaja dikosongkan)

BAB I
PENDAHULUAN

Pada bab ini dibahas mengenai latar belakang, rumusan masalah, batasan masalah, tujuan, manfaat, metodologi, dan sistematika laporan tugas akhir. Diharapkan dari penjelasan dalam bab ini gambaran tugas akhir secara umum dapat dipahami.

1.1. Latar Belakang

Pada era globalisasi saat ini kemajuan di bidang teknologi informasi sangatlah pesat.. Teknologi informasi sudah banyak mengubah bagaimana masyarakat berperilaku dalam kehidupan sehari hari.

Tanda tangan (dalam Bahasa Inggris: *signature*) atau paraf adalah bentuk dari identifikasi yang dituliskan pada dokumen sebagai sebuah bukti dari identitas dan kemauan. Bisa juga dibilang sebagai segel dari bukti.

Tanda tangan digital adalah bentuk tanda tangan yang diubah menjadi digital atau elektronik tetapi memiliki fungsi yang sama seperti tanda tangan yang seharusnya. Dengan adanya tanda tangan digital maka pembuatan sebuah tanda tangan akan dipermudah dan lebih fleksibel. Tanda Tangan Digital memiliki beberapa unsur yang harus dipernuhi untuk memenuhi kriteria yang sah[10].

Pada Tahun 2019, Indonesia dilanda musibah pandemi COVID-19 yang menyebabkan aktivitas perkuliahan di ITS harus dilakukan secara *WFH* (*Work From Home*). Dengan keadaan seperti ini maka banyak aktivitas yang sangat krusial harus di "*online*" kan dan salah satunya adalah kegiatan yang memerlukan sebuah tanda tangan.

Sebagai jawaban dari permasalahan tersebut maka dibuatlah Tanda Tangan Digital di Lingkungan ITS sebagai media untuk pemangku kepentingan di ITS memberikan tanda tangan yang berbentuk digital. Sistem pada Tugas Akhir ini akan diterapkan di lingkungan ITS dan legalitas tanda tangan hanya tercakup pada lingkup lingkungan ITS.

Diharapkan dengan adanya Implementasi Tanda Tangan Digital maka Kegiatan Menandatangan di lingkungan ITS akan semakin mudah dan menghemat biaya serta mempercepat alurnya tanpa harus meluangkan waktu bertatap muka (*offline*).

1.2. Rumusan Permasalahan

Rumusan masalah yang diangkat dalam Tugas Akhir ini adalah sebagai berikut :

1. Bagaimana cara membuat tanda tangan untuk lingkungan ITS yang praktis dan dapat diakses dengan mudah ?
2. Bagaiman cara mengetahui keabsahan dari tanda tangan yang dibuat secara digital?
3. Bagaimana cara menjaga keutuhan tanda tangan yang dikirim sercara digital?

1.3 Batasan Masalah

Permasalahan yang dibahas dalam Tugas Akhir ini memiliki beberapa batasan, di antaranya sebagai berikut.

1. Tidak menggunakan *third party Certificate Authority (CA).*
2. Menggunakan *database* MySql sebagai *database* penyimpan tanda tangan.
3. Menggunakan kerangka kerja berbasis php yaitu Laravel sebagai kerangka kerja sistem tanda tangan berbasis web.

1.4 Tujuan

Tujuan dari pembuatan tugas akhir ini adalah sebagai berikut :

1. Menghasilkan tanda tangan *online* yang memiliki aspek sekuritas yaitu *Identification and Authentication, Confidentiality, Data Integritty and Non-Repudiation.*
2. Menghasilkan tanda tangan online yang digambar dengan tangan.

1.5 Manfaat

Hasil dari tugas akhir ini memiliki manfaat untuk menghasilkan sebuah sistem tanda tangan online yang dibuat dengan tangan dan terbukti serta aman. Tanda tangan yang dikumpulkan di *database* dapat tersusun dengan rapih dan tercatat, sehingga pembuktian dan pencarian tanda tangan dapat dilakukan dengan mudah.

1.6 Tahapan Tugas Akhir

Tahapan-tahapan yang dilakukan dalam pengerjaan Tugas Akhir ini adalah sebagai berikut.

1.6.1 Penyusunan Proposal Tugas akhir

Proposal tugas akhir ini terdiri dari 8 bab. Bab 1 berisi identitas saya sebagai pengusul proposal Bab 2 yang berisi judul tugas akhir yang di usulkan. Bab 3 yaitu pendahuluan yang terdiri dari beberapa subbab antara lain latar belakang tugas akhir, rumusan masalah, batasan masalah, tujuan pengerjaan tugas akhir, dan manfaat dari hasil pengerjaan tugas akhir. Bab 4 menjabarkan tinjauan pustaka yaitu data yang dipakai sebagai refensi pendukung pengerjaan tugas akhir. Bab 5 berisi ringkasan tugas akhir yang berisi detail aturan dan runut pengerjaannya. Bab 6 berisi metodologi yang digunakan mulai dari tahap penyusunan proposal hingga penyusunan buku tugas akhir. Bab 7 berisi *timeline* pengerjaan yang

menunjukkan jadwal pengerjaan tugas akhir. Dan Bab 8 berisi daftar pustaka yang dipakai dalam penyusunan proposal ini.

1.6.2 Studi Literatur

Studi literatur ini akan dipeljari sejumlah referensi yang diperlukan dalam pembuatan sistem yaitu terkait penggunaan Kriptografi, flow PKI (Public Key Infrastructure), dan bahasa pemograman laravel untuk sistem website.

1.6.3 Perancangan Sistem

Tahap ini meliputi perancangan sistem berdasarkan studi literatur dan pembelajaran konsep teknologi dari sistem yang ada. Tahap ini mendefinisikan alur dari implementasi. Langkah-langkah yang dikerjakan juga didefinisikan pada tahap ini. Pada tahapan ini dibuat *prototype* sistem, yang merupakan rancangan dasar dari sistem yang akan dibuat. Kemudian dilakukan desain suatu sistem dan desain proses-proses yang ada.

1.6.4 Implementasi Sistem

Implementasi merupakan tahap membangun rancangan program yang telah dibuat. Pada tahapan ini, rancangan system yang terdapat pada tahapan sebelumnya dibangun menjadi sebuah perangkat lunak yang sesuai dengan apa yang telah direncanakan.

1.6.5 Pengujian dan evaluasi

Pada tahap ini dilakukan uji coba terhadap perangkat lunak yang telah dibuat sesuai dengan desain sistem untuk mengetahui kemampuan aplikasi, mengamati kinerja sistem, serta mengidentifikasi kendala yang mungkin timbul pada aplikasi yang dibuat.

1.6.6 Penyusunan buku Tugas Akhir

Pada tahapan ini disusun buku yang memuat dokumentasi mengenai pembuatan serta hasil dari implementasi perangkat lunak yang telah dibuat.

1.7 Sistematika Penulisan Laporan Tugas Akhir

Pada tahap ini dilakukan penyusunan laporan yang menjelaskan dasar teori dan metode yang digunakan dalam tugas akhir ini. Buku tugas akhir juga berisi hasil implementasi dan pengujian aplikasi perangkat lunak. Sistematika penulisan buku tugas akhir secara garis besar meliputi :

Bab I Pendahuluan

Bab yang berisi mengenai latar belakang, tujuan, dan manfaat dari pembuatan Tugas Akhir. Selain itu permasalahan, batasan masalah, metodologi yang digunakan, dan sistematika penulisan juga merupakan bagian dari bab ini.

Bab II Tinjauan Pustaka

Bab ini berisi penjelasan secara detail mengenai dasar dasar penunjang dan teori-teori yang digunakan untuk mendukung pembuatan Tugas Akhir ini.

Bab III Perancangan Sistem

Bab ini berisi tentang desain sistem yang disajikan dalam bentuk deskripsi rinci dari tahapan-tahapan proses yang ada dalam sistem.

Bab IV Implementasi

Bab ini membahas implementasi dari desain yang telah dibuat pada bab sebelumnya. Penjelasan berupa pseudocode yang digunakan untuk proses implementasi.

Bab V Uji Coba Dan Evaluasi

Bab ini menjelaskan kemampuan sistem dengan melakukan pengujian kebenaran dan pengujian kinerja dari sistem yang telah dibuat.

Bab VI Kesimpulan Dan Saran

Bab ini merupakan bab terakhir yang menyampaikan kesimpulan dari hasil uji coba yang dilakukan dan saran untuk pengembangan sistem ke depannya.

(Halaman ini sengaja dikosongkan)

BAB II
TINJAUAN PUSTAKA

2. Tinjauan Pustaka

Bab ini berisi pembahasan mengenai teori-teori dasar atau penjelasan dari metode dan alat yang digunakan dalam tugas akhir. Bab ini bertujuan untuk memberikan gambaran secara umum terhadap program yang dibuat dan berguna sebagai penunjang dalam pengembangan riset yang berkaitan.

2.1 *Encryption*

Enkripsi adalah proses mengubah informasi biasa (*plaintext*) menjadi teks yang tidak dapat dipahami (*ciphertext*). Dekripsi adalah kebalikannya, atau dengan kata lain, berpindah dari *ciphertext* yang tidak dapat dipahami kembali ke teks biasa. *Cryptosystem* adalah daftar elemen yang diurutkan teks biasa yang mungkin terbatas, teks sandi yang mungkin terbatas, hingga kemungkinan kunci, dan enkripsi dan dekripsi algoritma yang sesuai dengan setiap kunci[11]..

Proses Enkripsi dan dekripsi memerlukan sebuah kunci. Kunci ini dibutuhkan agar data yang di enkripsi bisa dibuka dan sebaliknya saat data ingin dienkripsi.

Enkripsi dapat dibagi menjadi dua[1]..

1. *Symmetric Encryption*
   Symmetric Encryption adalah bentuk enkripsi dimana pengirim dan penerima menggunakan kunci yang sama. Algoritma yang menggunakan *Symmetric Encryption* disebut dengan *Symmetric Algorithm*
2. *Asymmetric Encryption*
   Asymmetric Encryption adalah bentuk enkripsi yang menggunakan dua kunci yang berbeda. Kunci ini sering

dinamakan dengan public key dan private key. Proses ini sering juga disebut dengan *Public Key Encryption*. Beberapa Algoritma *Asymmetric* yaitu : DSA, RSA , ECDSA, EdDSA.

2.2 *EdDSA*

EdDSA (Edwards-curve Digital Signature Algorithm) adalah algoritma tanda tangan digital modern dan aman berdasarkan kurva eliptik yang dioptimalkan pada kinerja, seperti kurva 255-bit *Curve25519* dan kurva 448-bit *Curve448-Goldilocks*. Tanda tangan *EdDSA* menggunakan bentuk kurva elips *Edwards* (untuk alasan kinerja), masing-masing *edwards25519* dan *edwards448*. Algoritma *EdDSA* didasarkan pada algoritma tanda tangan *Schnorr* dan bergantung pada kesulitan masalah *ECDLP*.

Algoritma tanda tangan *EdDSA* dan variannya Ed25519 dan Ed448 secara teknis dijelaskan dalam RFC 8032.

Ed25519 dan Ed448 menggunakan *Secret Key/ Private Ke*y kecil (masing-masing 32 atau 57 byte), *Public Key* kecil (32 atau 57 byte) dan *Signature* kecil (64 atau 114 byte) dengan tingkat keamanan tinggi pada saat yang sama (masing-masing 128-bit atau 224-bit) [5].

2.3 ChaCha20-Poly1305

Chacha20-Poly1305 adalah gabungan 2 jenis enkripsi simetrik yang tergolong *AEAD (Authenticated Encryption with Additional Data)*. ChaCha beroperasi pada 32-bit word, mengambil sebagai input kunci 256-bit K = (k0, k1, k2, k3, k4, k5, k6, k7) dan counter 32-bit C = (c0), dan menghasilkan urutan blok keystream 512-bit. Fungsi ini bekerja pada matriks 4 × 4.[3]

Algorithm 1 ChaCha

Input: Key K, Counter C, and Nonce N

Output: Keystream Z

Generate initial matrix X using K, C, and N

$y \leftarrow X$

for $i \leftarrow 0$ **to** 9 **do**

/* Column Round */

$(x_0, x_4, x_8, x_{12}) \leftarrow quarterround(x_0, x_4, x_8, x_{12})$

$(x_5, x_9, x_{13}, x_1) \leftarrow quarterround(x_5, x_9, x_{13}, x_1)$

$(x_{10}, x_{14}, x_2, x_6) \leftarrow quarterround(x_{10}, x_{14}, x_2, x_6)$

$(x_{15}, x_3, x_7, x_{11}) \leftarrow quarterround(x_{15}, x_3, x_7, x_{11})$

/* Diagonal Round */

$(x_0, x_5, x_{10}, x_{15}) \leftarrow quarterround(x_0, x_5, x_{10}, x_{15})$

$(x_1, x_6, x_{11}, x_{12}) \leftarrow quarterround(x_1, x_6, x_{11}, x_{12})$

$(x_2, x_7, x_8, x_{13}) \leftarrow quarterround(x_2, x_7, x_8, x_{13})$

$(x_3, x_4, x_9, x_{14}) \leftarrow quarterround(x_3, x_4, x_9, x_{14})$

end for

$Z \leftarrow X + y$

return Z

Gambar 2. 1 Algo ChaCha

Poly1305 adalah kode otentikasi pesan kriptografi (MAC). Input ke Poly1305 adalah kunci satu kali 256-bit dan pesan panjang sesuka hati. Outputnya adalah tag 128-bit.

Maka dapat disimpulkan bahwa fungsi penggabungan algo ChaCha dan Poly adalah agar pengenkripsian data yang dilakukan oleh algoritma chacha dapat memiliki autentikasi tersendiri yang dioperasikan oleh Poly.

2.4 *Digital Signature*

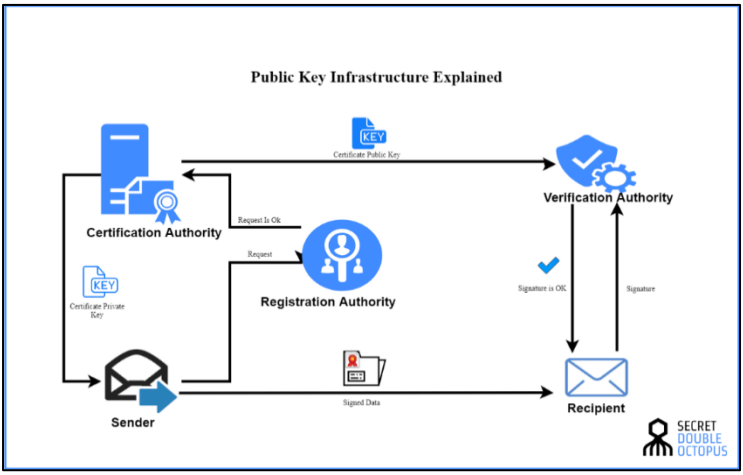
Digital Signature atau dalam bahasa Indonesia disebut dengan tanda tangan digital adalah skema matematis yang menjamin privasi percakapan, integritas data, keaslian pesan / pengirim digital dan non-repudiasi pengirim. *Digital Signature* menjamin keaslian sebuah data yang terbuat secara online. Berbeda dengan *E-signature* (dalam bahasa indonesia : tanda tangan elektronik), *Digital Signature* lebih condong kepada teknologi *cryptography* dan sekuritas sedangkan *E-signature* hanya sebuah tanda tangan yang terbuat secara online.

Digital Signature memiliki 4 faktor utama yang harus diperhatikan yaitu *Privacy, Authenticatiion, Integrity dan Non-Repudiation*[1].

1. *Privacy*
   Privacy menjamin keamanan data agar tidak bisa diubah dan diketahui oleh pihak manapun selain pengirim dan penerima.
2. *Authentication*
   Authentication menjadi sebuah bukti bahwa data tersebut bersifat autentik. Banyak unsur yang dapat dijadikan autentikasi sebuah data sebagai contoh tempat dibuatnya data, nama pengirim, lokasi pembuatan data, dan masih banyak lagi tergantung dengan srandar lembaga yang dijadikan patokan otorisasi. Lembaga yang mengatur *Digital Signature* disebut juga dengan *CA (Certificate Authority)*.
3. *Integrity*
   adalah keutuhan sebuah data yang terlepas dari adanya perubahan dan penghilangan informasi dari data tersebut. Integritas sebuah data harus jelas tanpa adanya campur tangan dari pihak lain.
4. *Non-Repudiation*
   Non-Repudiation mencegah terjadinya penyangkalan dari pihak yang memberikan sumber data / melakukan transaksi. Hal ini diperlukan agar tidak terjadinya penipuan karena penyangkalan.

2.5 *Public Key Infrastructure*

Public Key Infrastructure (PKI) adalah fondasi di mana aplikasi lain, sistem, dan komponen keamanan jaringan dibangun. *PKI* merupakan komponen penting dari keseluruhan strategi keamanan yang harus bekerja sama dengan keamanan lainnya mekanisme, praktik bisnis, dan upaya manajemen risiko[2]..



Gambar 2. 2 Alur PKI (Public key Infrastructure)[14]

Pada gambar di atas dijelaskan cara kerja framework PKI secara keseluruhan.

Pengirim akan mengirimkan data kepada penerima. Sebelum pengiriman data, pengirim harus memiliki *public key* dari penerima , sedangkan penerima harus memiliki *private key* miliknya sendiri. Semua key ini diisukan oleh *Certificate Authority* berdasarkan *Digital Certificate* dari tiap individu.

PKI mempunyai 4 fungsi proses utama.

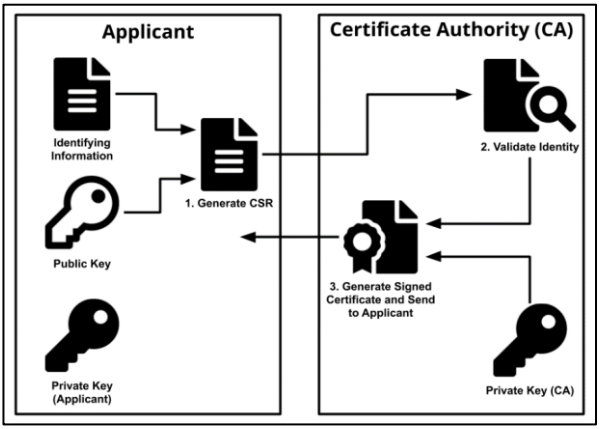
1. *Public Key Cryptography*
   Meliputi pembuatan, distribusi, administrasi, dan pengaturan dari kunci enkripsi (*Cryptographic Keys*).
2. *Certificate Issuance*
   Mengikat public-key dengan suatu individu, organisasi, atau entity, atau sebuah data.
3. *Certificate Validation*
   Melakukan verifikasi terhadap adanya sebuah

hubungan dan masih berlakunya sebuah *certificate* untuk sebuah operasi.

4. *Certificate Revocation*
   Membatalkan sertifikat yang diterbitkan sebelumnya dan menerbitkannya pembatalan ke daftar pencabutan sertifikat atau mengaktifkan proses status protokol sertifikat *online*.

2.6 *Certificate Authority*

Certificate Authority (CA) adalah perusahaan atau organisasi yang bertindak untuk memvalidasi identitas entitas (seperti situs web, alamat email, perusahaan, atau individu) dan mengikatnya ke kunci kriptografi melalui penerbitan dokumen elektronik yang dikenal sebagai digital certificate[13].



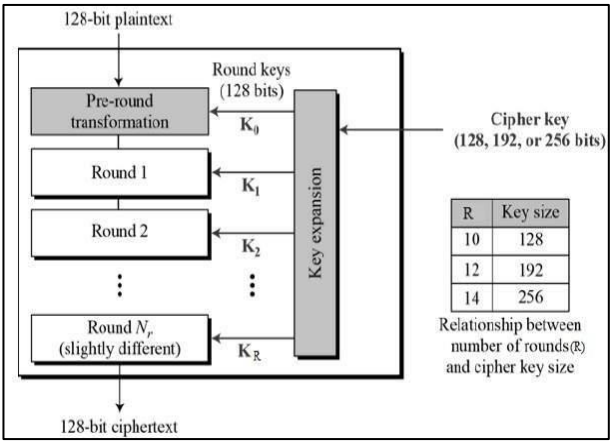
Gambar 2. 3 Alur Certificate Authority (CA)[13]

Fungsi utama CA adalah untuk validasi informasi agar bisa memproduksi kunci yang diperlukan, maka dari itu dibutuhkan suatu organisasi khusus yang dapat memvalidasi data tersebut dan diakui oleh umum/publik. Dalam hal ini, *Certificate Authority* juga bisa dibuat untuk keperluan *local*/

tidak untuk umum dengan membuatnya menggunakan *opensource CA* yang ada seperti contohnya *OpenSSL, Let's Encrypt*, dan masih banyak lagi.

2.7 *Advanced Encryption Standard*

Advance Encryption Standard (AES) adalah algoritma kriptografi yang dapat digunakan untuk mengamankan data. Algoritma *AES* adalah chiper blok simetris yang dapat mengenkripsi dan mendekripsi informasi. *AES* dipilih karena algoritma ini kuat untuk serangan diferensial, serangan diferensial terpotong, linier serangan, serangan interpolasi, dan serangan persegi[4].



Gambar 2. 4 Alur Advance Encryption Standar (AES)[15]

Jenis *AES* terbagi menjadi tiga.

1. *AES*-128

2. *AES*-192

3. *AES*-256

Yang membedakan tiap jenis *AES* adalah jumlah kunci di belakangnya sesuai pada judulnya. *AES*-128 menggunakan 10 *round*, *AES*-192 menggunakan 12 *round*, dan *AES*-256 menggunakan 14 *round*.

Algoritma *AES* terdiri dari 4 *stage* yaitu dari awal *Substitute Bytes Transformation*, *Shift Rows Transformation*, *Mix Columns Transformation*, dan *stage* terakhir adalah *AddRoundKey Transformation*. Pada *stage* pertama dibutuhkan *S-box* sebagai referensi untuk mengubah bentuk *byte* dengan *substitute*[12].

2.8 Laravel

Laravel adalah *framework* yang digunakan untuk membuat aplikasi berbasis web dengan menggunakan bahasa PHP. Laravel memiliki banyak ekstensi dan library yang bisa digunakan hanya dengan *install* menggunakan *composer*. Untuk IDE akan menggunakan visual studio code. Untuk lebih lanjut bisa membaca journal ini [6].

2.9 MySQL

MySQL adalah *Relational Database Management System* (RDBMS) *open source* yang menggunakan *Structured Query Language* (SQL). SQL adalah bahasa paling populer untuk menambahkan, mengakses, dan mengelola konten dalam database. Ini paling terkenal karena pemrosesannya yang cepat, keandalan yang teruji, kemudahan dan fleksibilitas penggunaan. Dalam pengoperasian di Tugas Akhir ini, MySQL akan dijalankan dengan GUI phpMyAdmin [7].

2.10 PHP

PHP *Hypertext Preprocessor* (PHP) adalah bahasa pemrograman yang memungkinkan pengembang web membuat konten dinamis yang berinteraksi dengan database. PHP pada dasarnya digunakan untuk mengembangkan aplikasi perangkat lunak berbasis web. Tutorial ini membantu Anda membangun basis dengan PHP[8].

2.11 QR Code

QR code(*Quick Response Code)* adalah *barcode* yang memberikan beragam jenis informasi hanya dalam bentuk *barcode* tersebut. Cara menggunakannya adalah dengan meng *scan* barcode tersebut dengan sebuah alat. *QR code* mampu menyimpan 2089 digit atau 4289 karakter, termasuk tanda baca dan karakter spesial. *QR code* lebih praktis dibanding *barcode* karena mampu menyimpan lebih banyak data. Hal tersebut membuatnya mampu di-*scan* oleh *smartphone* dan menampilkan data atau informasi yang dimuatnya.

2.12 Google Map Javascript API

Maps JavaScript API memungkinkan Anda menyesuaikan peta dengan konten dan citra Anda sendiri untuk ditampilkan di halaman web dan perangkat seluler. Maps JavaScript API menampilkan empat tipe peta dasar (peta jalan, satelit, hibrid, dan medan) yang dapat Anda modifikasi menggunakan lapisan dan gaya, kontrol dan peristiwa, serta berbagai layanan dan pustaka[9].

2.13 SSL, TLS, HTTPS

SSL (*Secure Sockets Layer*) adalah bentuk teknologi keamanan untuk menjaga hubungan *online* aman dan

melindungi data sensitif yang dikirim dua sistem atau lebih, mencegah pihak ketiga membaca dan mengubah informasi apa pun yang ditransfer, termasuk detail pribadi yang potensial. Kedua sistem tersebut dapat berupa server dan klien atau server ke server. TLS (*Transport Layer Security*) hanyalah versi SSL yang diperbarui dan lebih aman. HTTPS (*Hyper Text Transfer Protocol Secure*) muncul di URL saat situs web diamankan dengan sertifikat SSL.

BAB III
PERANCANGAN SISTEM

Perancangan sistem merupakan bagian penting dalam pembuatan perangkat lunak dan perangkat keras yang berupa perencanaan secara teknis dari sistem yang dibuat. Pada bab ini akan dibahas mengenai beberapa hal secara umum yaitu deskripsi umum sistem, arsitektur umum sistem, diagram kasus penggunaan, perancangan basis data, diagram alur dan desain antarmuka perangkat lunak.

3.1 Deskripsi Umum

Tugas akhir membangun aplikasi Tanda Tangan Online atau *E-Signature* menggunakan Enkripsi dan pengembangan aplikasi berbasis web.

Enkripsi data yang digunakan pada sistem ini ada 2 jenis yaitu *Symmetric Enryption* dan *Asymmetric Encryption*. Kedua jenis Enkripsi ini memiliki tujuannya masing – masing dan akan dijelaskan pada tugas akhir ini. Tanda tangan yang dibuat pada aplikasi ini akan diproses dengan 2 enkripsi ini sehingga sistem dapat memenuhi aspek sekuritas yaitu *Identification and Authentication, Confidentiality, Data Integritty and Non-Repudiation.*

Selain enkripsi pada data, enkripsi juga diterapkan pada konektivitas. Web dibungkus dengan menggunakan SSL serta pengiriman email melalui sistem menggunakan TLS. Maka dari itu dengan menggunakan SSL maka web diakses dalam bentuk HTTPS.

Khusus untuk penerapan sistem *CA (Certificate Authorit)* pada sistem ini, *CA* yang digunakan adalah OpenSSL *Lets Encrypt* untuk penyediaan kunci HTTPS. Untuk penguncian dan pembuatan *digital signature* pada tanda tangan menerapkan pembuatan proses secara sendiri menggunakan

minisign. OpenSSL *Lets Encrypt* serta Minisgin adalah sistem dan *library* yang dapat digunakan oleh setiap pengembang secara gratis (*Open Source*).

3.2 Daftar Istilah

Daftar istilah yang sering digunakan pada buku tugas akhir ini dapat diliat pada Tabel 3.1.

Tabel 3. 1 Daftar istilah

No.	Istilah	Penjelasan
1	*Digital Signature*	Dalam bahasa Indonesia disebut juga Tanda tangan digital, adalah sebuah tanda tangan yang dibentuk dengan perangkat elektronik dan memiliki bukti dalam bentuk digital yang sah dari pembuat tanda tangan.
2	*Database*	Kumpulan informasi yang disimpan di dalam komputer secara sistematik sehingga dapat diperiksa menggunakan suatu program komputer untuk memperoleh informasi dari basis data tersebut
3	*Secret Key*	Kunci yang terbentuk dari randomisasi integer dan string yang memiliki ukuran byte yang berbeda beda sesuai dengan algoritma yang dipakai. Kunci ini digunakan untuk mengunci / mengenkripsi data. *Secret Key* yang memiliki *Public Key* biasanya disebut juga dengan *Private Key*.
4.	*Public Key*	Kunci yang terbentuk untuk membuka data yang terenkripsi dengan *Secret Key*. *Public Key* yang terbentuk sangat bergantung pada *Secret Key* dan tidak

		bisa membuka enkripsi dengan *Secret Key* yang bukan pasangannya
5.	*QR Code*	*QR Code* adalah kode matriks atau barcode dua dimensi yang berasal dari kata "*Quick Response*", dimana isi kode dapat diuraikan dengan cepat dan tepat.
6.	*IP*	*IP* atau *Internet Protocol* adalah seperangkat aturan yang mengatur aktivitas internet dan bisa memfasilitasi untuk penyelesaian berbagai tindakan di World Wide Web.

3.3 Arsitektur Sistem

Bab ini menjelaskan arsitektur umum sistem yang akan dibangun. Sistem ini terdiri dari beberapa fitur yang memiliki proses masing – masing yang berbeda dan akan dijelaskan secara terpisah.

3.3.1 Perancangan Sistem

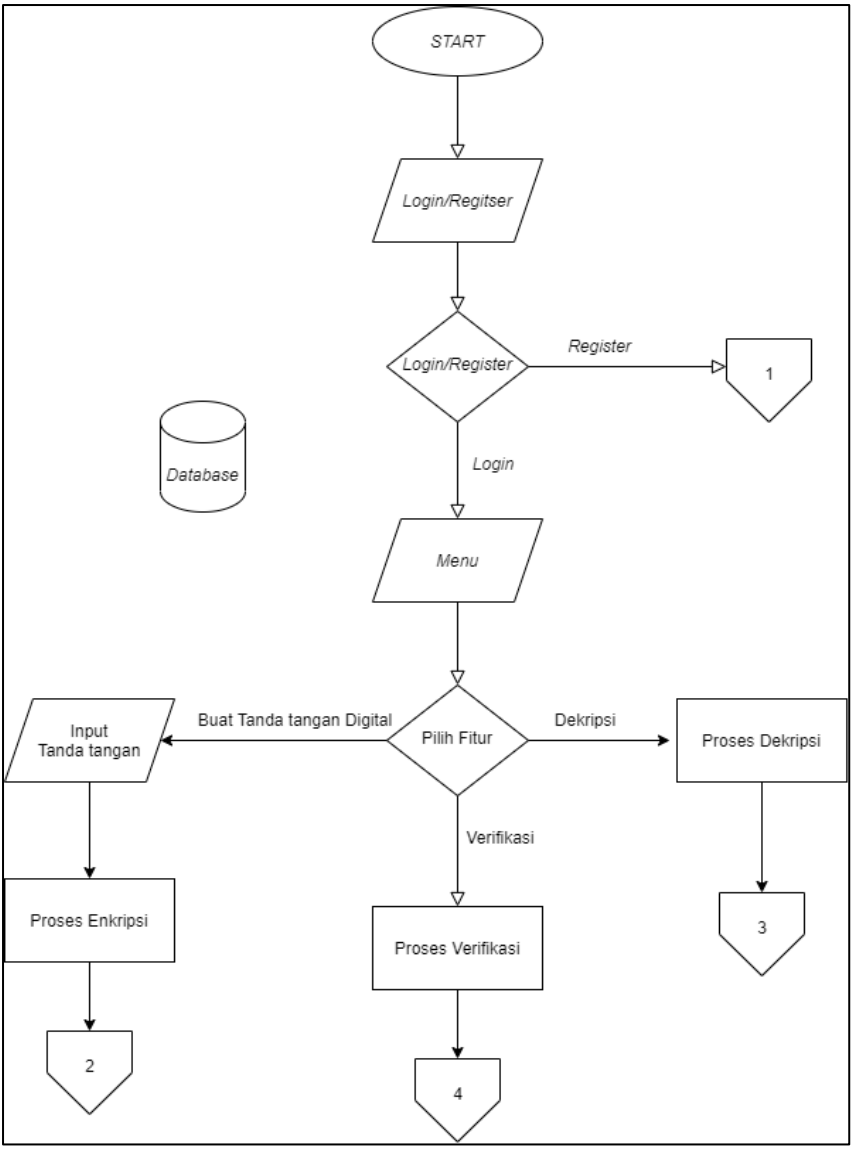
Perancangan Sistem dari Sistem Tanda Tangan Online memiliki beberapa subproses yang akan ditampilkan pada Tabel 3.2.

Setiap subproses memiliiki fungsinya masing – masing sehingga penjelasan sistem akan dibagi sesuai kegunaan subproses dan detail subproses. Flow sistem secara keseluruhan dapat dilihat pada Flowchart 3.1

Tabel 3. 2 Beberapa subproses dalam perancangan sistem

No.	SubProses	Deskripsi
1	Proses	Proses ini memiliki fungsi untuk

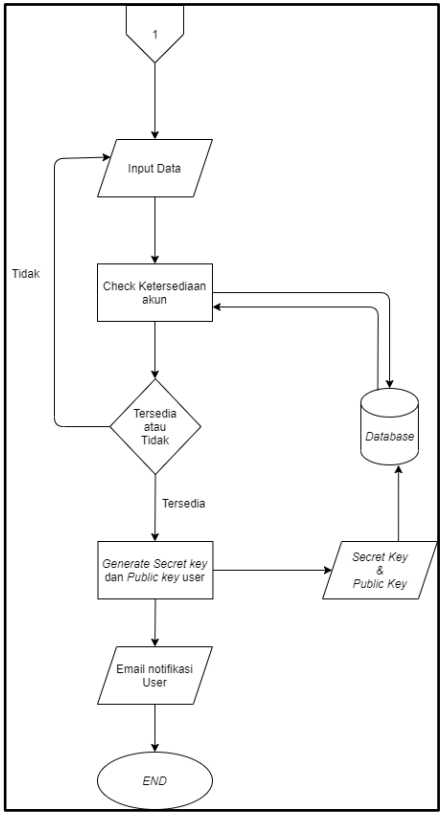
	Registrasi	menyimpan data *User* baru dan sekaligus membuat *Secret key* dan *Private Key unique* tiap *user* yang akan digunakan pada proses yang berbeda.
2	Proses Enkripsi	Proses pembuatan tanda tangan online akan melalui proses ini yang bertujuan untuk mengamankan tanda tangan sehingga terkunci secara fisik dan dapat dibuktikan/ diverifikasi oleh sistem pada proses yang lainnya.
3	Proses Dekripsi	Proses pengubahan tanda tangan yang sudah di enkripsi agar dapat dilihat dan digunakn oleh *user*.
4.	Proses Verifikasi	Proses ini bertujuan agar siapapun *user* yang mendapatkan tanda tangan dari aplikasi dapat menguji keutuhan data tanda tangan tersebut.



Gambar 3. 1 Flow system secara keseluruhan

3.3.2 Perancangan Proses Registrasi

Pada proses ini, *User* yang belum terdata dalam aplikasi akan menginput data yang diminta oleh aplikasi. Setelah menginput data, maka sistem akan membuat *private* dan *public key user* yang berbeda dengan tiap user lainnya. Setelah itu *user* akan mendapatkan notifikasi lewat email bahwa mereka sudah terdaftar.

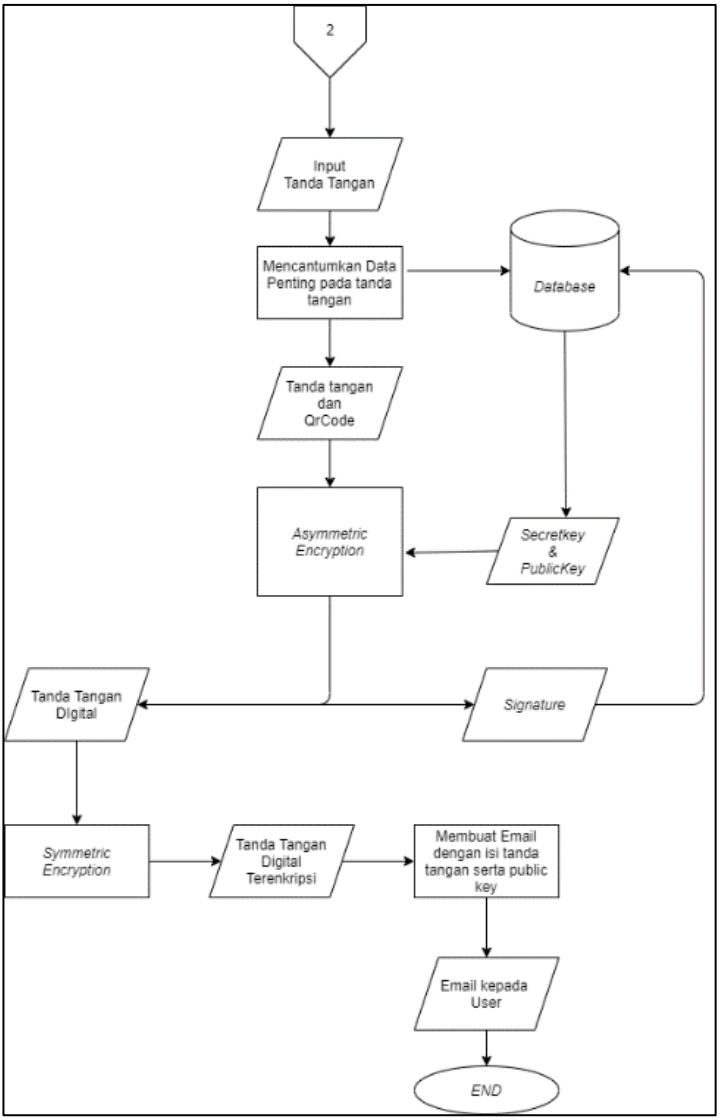


Gambar 3. 2 Perancangan alur registrasi

3.3.3 Perancangan Proses Enkripsi

Pada proses ini, user akan memasukan input berupa *password* dan tanda tangan. Aplikasi tidak hanya menerima *password* dan tanda tangan, tetapi hal lainnya seperti perihal pembuatan tanda tangan, pilihan data yang akan di *display* pada tanda tangan, dan data lokasi *user* yang di dapatkan menggunakan Google Map JavaScript API. Pada saat awal pembukaan halaman, *user* akan mendapatkan *prompt* untuk menyalakan *GPS* pada *device* yang akan mendeteksi lokasi *user*. Tanda tangan dapat berupa gambar yang sudah jadi ataupun tanda tangan secara manual melewati *browser*. Lalu setelah *input*, tanda tangan akan dienkripsi secara *Asymmetric* dan *Symmetric Encryption* dengan menggunakan algoritmanya masing masing.

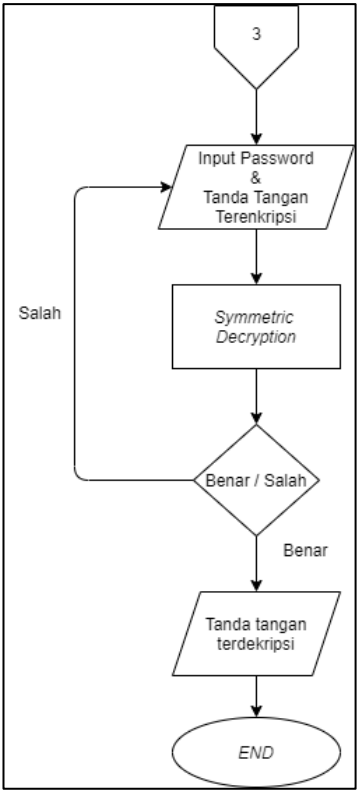
Asymmetric Encryption pada bagian ini digunakan untuk menandatangani/ *signing*, bukan untuk mengenkripsi data agar tidak terbaca. Lalu setelah itu barulah digunakannya *Symmetric Encryption* untuk mengunci data secara fisik agar pada saat data di *sniffing*, pencuri data akan kesulitan mengubah bentuk tanda tangan menjadi sebuah gambar yang dimengerti manusia. *Database* akan menerima *Digital Signature* yang terbentuk dari *Asymmetric Encryption.* Setelah tanda tangan terenkripsi maka dikirimkan kepada *user* dalam bentuk terenkripsi melalui email. Tanda tangan yang terbentuk juga memiliki *QR Code* agar dapat ditelusuri detailnya.



Gambar 3. 3 Perancangan alur enkripsi

3.3.4 Perancangan Proses Dekripsi

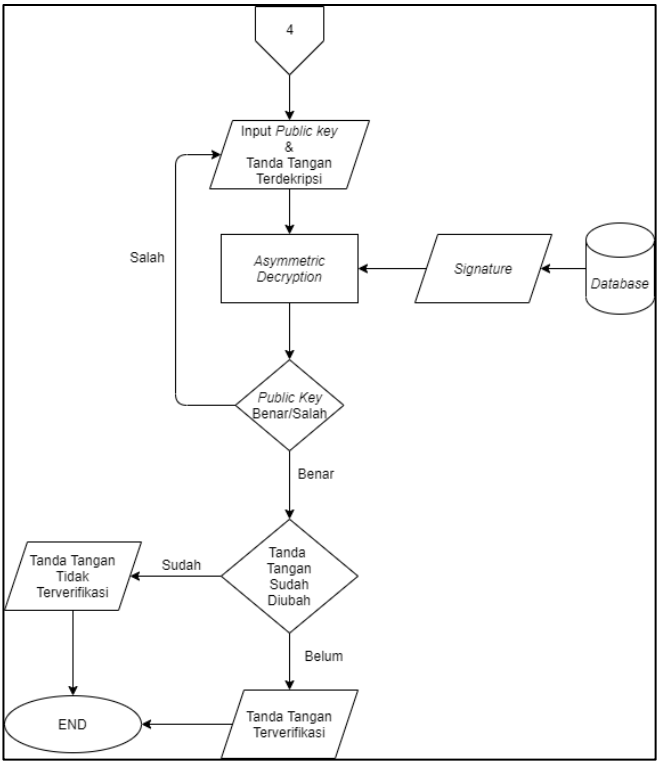
Pada proses ini, tanda tangan yang sudah terenkripsi akan dimasukan ke sistem oleh *user*, disertai dengan *password* yang akan diberitahu oleh pemilik tanda tangan. Jika *password* benar maka tanda tangan akan dikirimkan kepada *user* dalam bentuk sudah terdekripsi. Jika *password* salah maka tanda tangan tidak akan terdekripsi.



Gambar 3. 4 Perancangan sistem dekripsi

3.3.5 Perancangan Proses Verifikasi

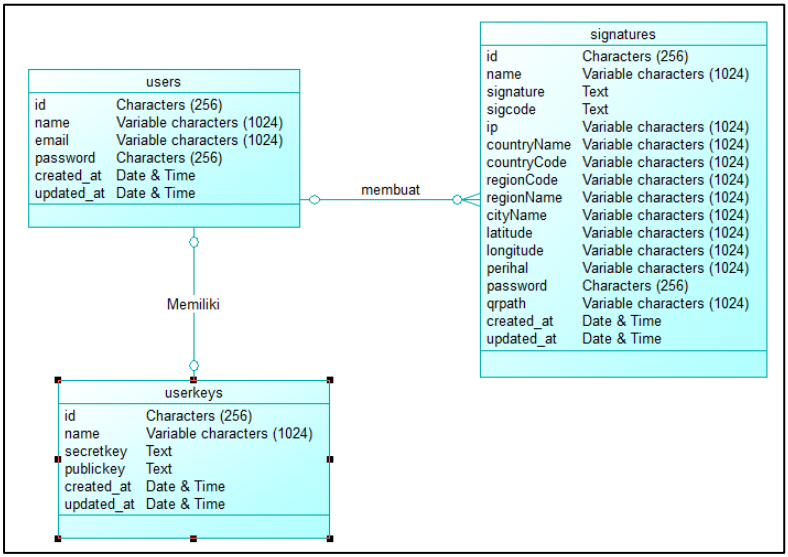
Pada proses ini, setelah *User* mendapatkan tanda tangan , maka tanda tangan tersebut dapat diketahui keutuhannya dengan memverifikasi menggunakan kode yang diberikan kepada *user*. Kode tersebut adalah kunci *public key* dari pemiliki tanda tangan dan akan disamakan dengan *Digital Signature* yang ada pada *database*. Semua aspek yang ada harus sesuai agar tanda tangan dapat terverifikasi.



Gambar 3. 5 Perancangan sistem verifikasi

3.3.5 Perancangan Basis Data

Perancangan basis data merupakan satu tahap dalam pembuatan aplikasi atau *website* yang berfungsi untuk menggambarkan struktur dan kerangka *Database* yang digunakan dalam pembuatan *website* Tanda Tangan Online. gambar diatas merupakan *Database* yang akan diimplementasikan. Sistem manajemen basis data yang digunakan adalah Mysql dengan GUI phpmyadmin. Untuk memudahkan pemahaman berikut ini adalah contoh data dalam bentuk diagram.

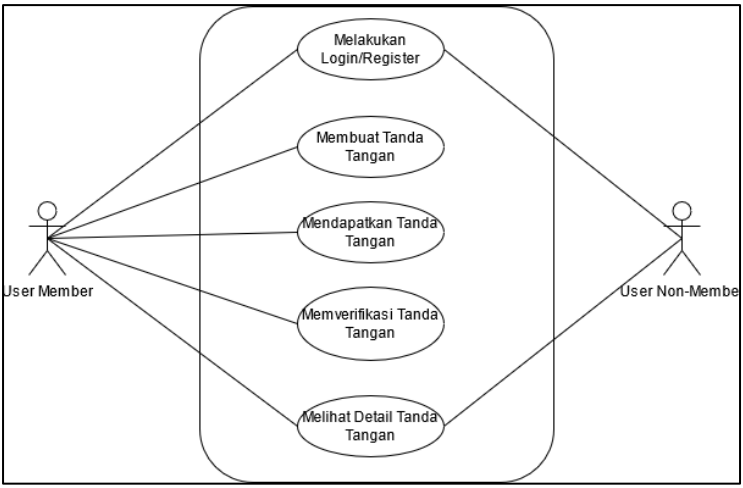


Gambar 3. 6 Tahap perancangan basis data

3.4 *Use Case Diagram*

Pada bagian ini, terdapat *use case diagram* yang menggambarkan akses penggunaan seperti pada gambar diatas. Terlihat bahwa *user* terbagi menjadi 2 yaitu *user member* dan *user non member*. *User member* adalah *user* yang terdaftar di

dalam sistem sedangkan *user non-member* adalah *user* yang tidak terdaftar di dalam sistem. Kedua *user* dapat mengakses fitur *login* dan *register* tetapi *user non member* mempunyai batasan dan hanya dapat melihat detail tanda tangan dengan cara meng *scan* QR code yang terdapat pada tanda tangan *user member*.



Gambar 3. 7 Use case diagram

BAB IV
IMPLEMENTASI

Bab ini membahas mengenai implementasi sistem yang dilakukan berdasarkan rancangan yang telah dijabarkan pada bab sebelumnya. Implementasi berupa kode sumber untuk membangun program,spesifikasi *hardware*. Implementasi yang dijelaskan dibagi menjadi lingkungan Implementasi perangkat keras dan lingkungan implementasi perangkat lunak pembangunan sistem.

4.1 Lingkungan Implementasi

Lingkungan implementasi merupakan lingkungan dimana sistem akan dibangun. Lingkungan implementasi dibagi menjadi lingkungan implementasi perangkat keras dan lingkungan implementasi perangkat lunak.

4.1.1 Lingkungan Implementasi Perangkat Keras

Pada bagian ini dijelaskan perangkat keras yang digunakan untuk menampung sistem.

Tabel 4. 1 Detail perangkat keras yang digunakan

Perangkat	**Detail**
Komputer Server	**Memory:** • **1 GB** **Disk:** • **20 GB** **OS:** • **UBUNTU 20.04 64 BIT** **CPU Cores:** • **1** **CPU Speed:** • **2400 Mhz**

4.1.2 Lingkungan Implementasi Perangkat Lunak

Pada bagian ini akan dibahas mengenai perangkat lunak apa saja yang dibutuhkan untuk membangun sistem. Lingkungan implementasi perangkat lunak dari sistem yang akan dibangun secara lebih lengkap di jelaskan pada **Error! Reference source not found.**.

Tabel 4. 2 Perangkat lunak yang dibutuhkan

Perangkat Lunak	**Detail**
Google Map Javascript API	API yang menghubungkan service pencarian lokasi yang disediakan oleh Google dalam bentuk javascript. Lokasi didapat dari informasi *latitude* dan *longitude* dari user.
Visual Studio Code	*Visual studio code* merupakan *text editor* yang digunkan untuk melakukan pemograman dan *support* terhadap IDE PlatformIO
minisign-php Library	*Library* yang digunakan untuk membuat secret key dan public key dengan menggunakan algoritma EdDSA dengan cepat.
Simple QR-Code Library	*Library* yang digunkan untuk menmbuat *QR Code* secara otomatis.
DOMPPDF Library	*Library* yang digunakan untuk merubah file html menjadi ekstensi pdf.

4.2 Implementasi Web Sistem Tanda Tangan Online

Pada bagian ini dijelaskan implementasi dari Sistem yang secara keseluruhannya menggunakan website yang di deploy pada suatu server.

4.2.1 Implementasi Halaman Login/Register

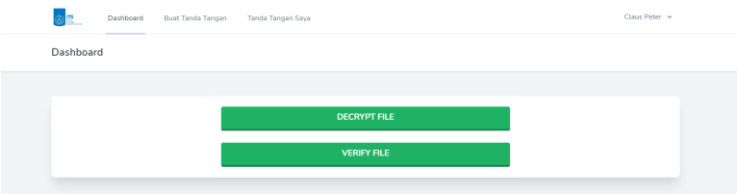
Pada Halaman ini, *User* diberi pilihan untuk login ataupun register aplikasi. *User* yang belum pernah mendaftarkan diri akan dianjurkan untuk membuat akun baru, supaya sistem dapat membuat *public key* dan *private key* untuk *user* yang baru mendaftar. Setiap *user* akan memiliki user akun yang berbeda yang *unique* antar *user*, begitu juga dengan *password* yang masuk. Dianjurkan *password* yang dibuat lebih dari 8 character ASCII.

Gambar 4. 1 Tampilan halaman login/register

Gambar 4. 2 Tampilan halaman login/register

4.2.2 Implementasi Halaman Utama

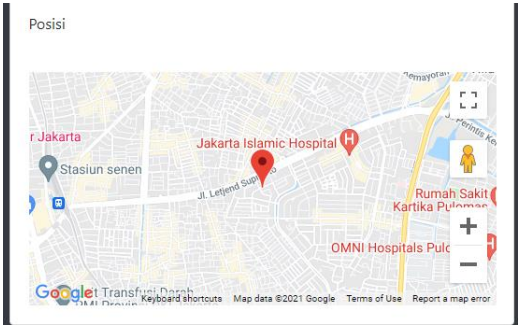
Halaman Utama Sistem Tanda Tangan *online* sangat *simple*. Tujuan dari halaman utama ini adalah agar *user* dapat memilihi fitur unutk membuat tanda tangan *online*, mendekripsi tanda tangan yang sudah dibuat di aplikasi, dan memverifikasi tanda tangan yang berasal dari aplikasi ini.

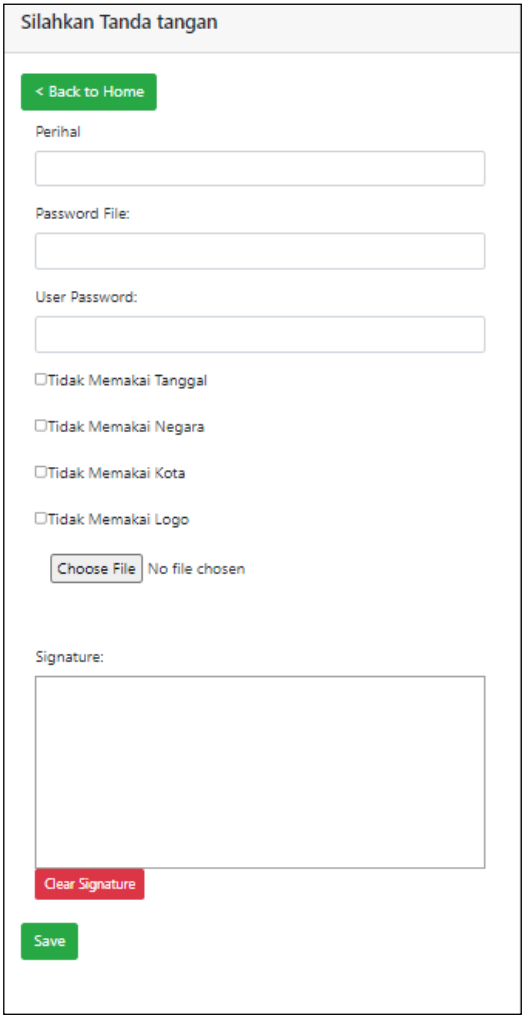


Gambar 4. 3 Tampilan halaman utama

4.2.3 Implementasi Halaman Pembuatan Tanda Tangan

Pada Halaman Ini, *User* akan membuat tanda tangan *online* dengan beberapa pilihan yang harus diisi, maupun tidak diisi. Perihal dan *Password file* harus diisi dikarenakan perihal berguna untuk pendataan tanda tangan yang akan terdaftar pada sistem. *Password File* adalah *password* yang akan digunakan untuk mengenkripsi *file* secara symmetric dan user tidak boleh lupa dengan password yang dibuatnya. *User password* adalah *password user* yang harus dimasukan, berguna untuk proses verifikasi apakah benar yang membuat tanda tangan adalah *user* sendiri. Lalu dibagian bawah ada 4 *checkbox* yang memiliki nilainya masing – masing yang akan berdampak pada hasil tanda tangan yang akan dibuat. *User* diberikan 2 pilihan, apakah mau membuat tanda tangan pada *browser* atau meng-*upload* tanda tangan dalam bentuk ekstensi gambar (png,jpg,svg). Tombol *save* akan memproses tanda tangan anda dan jika berhasil maka tanda tangan akan dikirimkan ke *user*. Pada bagian bawah pembuatan tanda tangan terdapat gambar peta dengan *pinpoint* merah. Letak *pinpoint* menandakan letak lokasi *user* saat ini. Untuk melakukan tanda tangan, *user* diminta untuk menyalakan *GPS* pada *device* terlebih dahulu. Jika *GPS* tidak dinyalakan maka sistem tidak akan berjalan dan tanda tangan online tidak dapat dilakukan.

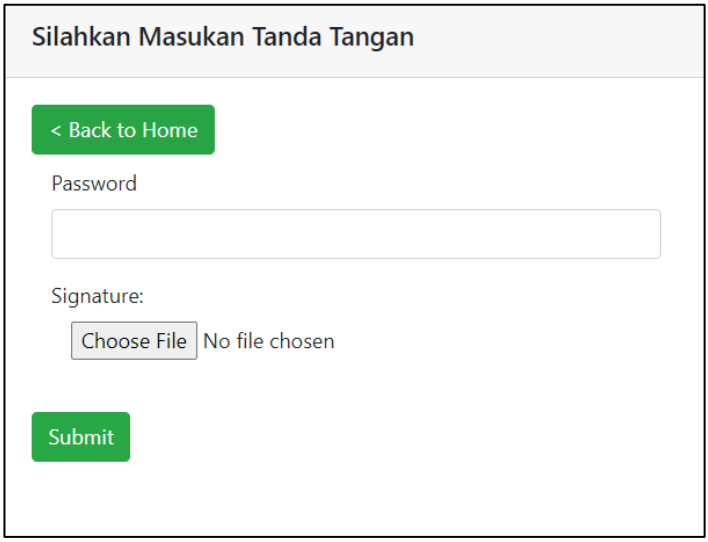




Gambar 4. 4 Tampilan halaman pembuatan tanda tangan

4.2.3 Implementasi Halaman Dekripsi

Halaman ini berguna untuk mendekripsi tanda tangan yang telah dibuat. *Password* adalah *password* tanda tangan yang dibuat oleh pemilik tanda tangan untuk mengunci file dan jika *input password* salah maka tanda tangan tidak akan dikirmkan kepada *user*.

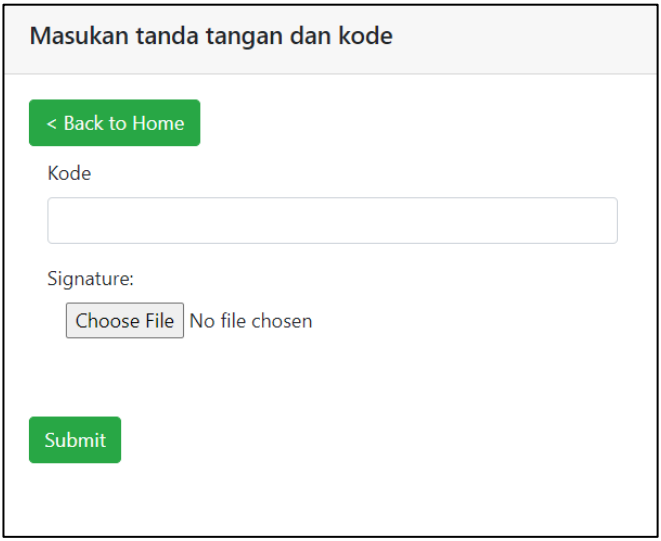


Gambar 4. 5 Tampilan halaman dekripsi

4.2.5 Implementasi Halaman Verifikasi

Halaman ini berguna untuk memverifikasi tanda tangan yang diterima oleh siapapun *user* yang memiliki *public key* penanda tangan. Jika *public key* benar dan tanda tangan belum dimanipulasi/dirubah bentuknya maka verifikasi berhasil serta data penandatangan akan muncul. Dan jika salah satu dari kedua syarat (kode *public key*/tanda tangan) maka

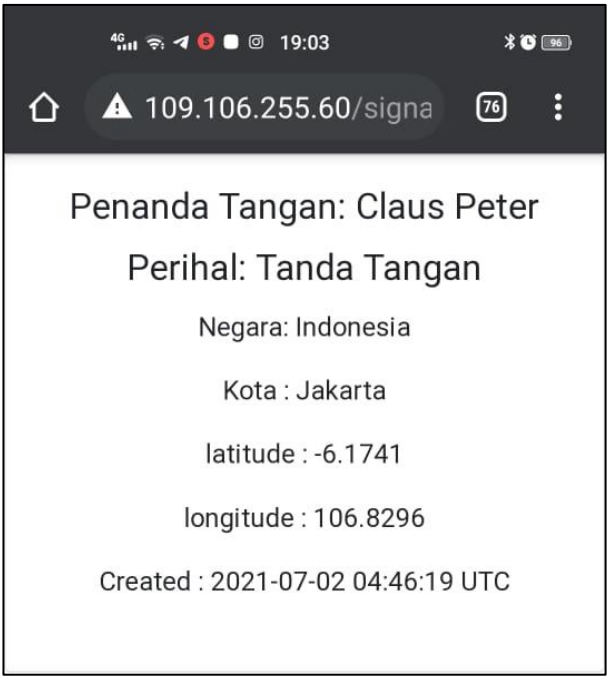
verifkasi akan memunculkan peringatan dan tanda tangan tidak terverifikasi.



Gambar 4. 6 Tampilan halaman verifikasi

4.2.6 Implementasi Halaman Informasi Tanda Tangan

Halaman ini akan muncul jika *QR Code* dari tanda tangan digital di *scan*. *QR Code* akan mengarahkan *link* menuju *page* yang menampilkan data detail dari penandatangan yang memiliki tanda tangan tersebut.



Gambar 4. 7 Tampilan halaman informasi tanda tangan

4.2.7 Implementasi Halaman Pencarian Tanda Tangan

```
nama  ←  user pengguna
password  ← password pengguna

passwordkey      ←  password kunci simetrik

secretkey ← kunci private asimetrik
publickey ← kunci public asimetrik

signature ← tanda tangan online

location ← mendapatkan lokasi

email ← email user

generate ← generate asymmetric keypair

d_signature ← digital signature

save ← save to database

location ← get location details

asymmetric_signature ← asymmetric encrypted signature
symmetric_signature ← symmetric encrypted signature
```

Pseudocode 4. 1 Halaman Pencarian Tanda Tangan

4.3 Implementasi *Code* Program Web Sistem Tanda Tangan Online

Pada bab ini akan di implementasikan *code* program Web Sistem Tanda Tangan Online. Dalam implementasi *code* program akan dibagi beberapa bagian dan selanjutnya akan dijelaskan pada subbab-subbab tersendiri.

4.3.1 Implementasi Proses Registrasi

Subbab ini membahas *code* program halaman registrasi. Untuk membuat *Secret Key* dan *Public Key*, maka digunakan *library* Minisign yang menjalankan algoritma EdDSA untuk pembuatan kuncinya.

```
SecretKey::generate()
        secretkey = serialize(password)
        publickey = secretkey->getPublickey()
        publickey->serialize()
        secretkey->save()
        publickey->save()
```

Pseudocode 4. 2 Proses registrasi

4.3.2 Implementasi Proses Enkripsi

Subab ini membahas *code* program halaman pembuatan Tanda tangan online yang harus melalui 2 tahap enkripsi. Enkripsi pertama bertujuan untuk membuat tanda tangan digital yang disimpan di *database* sedangkan yang ke 2 adalah *Symmetric Encryption* yang bertujuan untuk menghalangi *user* yang tidak diiginkan membuka file tanda tangan.

```
location = location::get();
qrcode = makeqr();
signature.loadlocation();
asymmetric()
        secretkeyGet();
        trustedcomment = signaturedetails;
        pwash()
                secretkey;
                prehash;
                trustedcomment;
                untrustedcomment;
        Sign(secretkey,signature)
                ->d_signature;
                ->asymmetric_signature;
        d_signature->save();
symmetric()
        key = pwash()
                passwordkey;
                salt;
                opslimit;
                memlimit;
                alg;
        file = xchacha20poly1305(esignature,key);
        symmetric_signature = file;
        file -> save();
email(symmetric_signature).
```

Pseudocode 4. 3 Proses enkripsi

4.3.3 Implementasi Proses Dekripsi

Subbab ini membahas *code* program halaman dekripsi tanda tangan online yang sudah didapat oleh *user*. Diperlukan *password* dalam pembuatan tanda tangan agar file dapat di dekripsi dengan benar.

```
Symmetric decryption()
        key = pwash()
                passwordkey;
                salt;
                opslimit;
                memlimit;
                alg;
        file = _xchacha20poly1305(symmetric signature,
                key);
        If(key == false){
                file = symmetric_signature;
        }
        else
                file = asymmetric_signature;
email(asymmetric_signature);
```

Pseudocode 4. 4 Proses dekripsi

4.3.4 Implementasi Proses Verifikasi

Subab ini membahas code program halaman verifikasi tanda tangan online. Dengan cara menyamakan tanda tangan dari *user* dan *digital signature* yang sudah disimpan di *database*, proses verifikasi menggunakan *public key* yang merupakan pasangan dari *secret key user* yang menandatangani file tersebut.

```
Asymmetric decryption()
    publickey;
    asymmetric signature;
    d_signature;
    verify(asymmetric_signature,d_signature,publickey)
        if(verify == false){
                "not authentic"
        }
        else
                "authentic","user_details"
```

Pseudocode 4. 5 Proses verifikasi

4.3.5 Implementasi Penggunaan Library dan API

Pada bagian ini akan dijelaskan Implementasi khusus dalam penggunaan *library* yang akan dipakai dalam pembuatan sistem ini. Bagian ini dibuat untuk memperjelas kegunaan secara detail dan penggunaan langsung *library* dalam sistem yang akan dibuat.

4.3.5.1 Implementasi Penggunaan Google Map Javascript API

Penggunaan API ini akan dilakukan pada *tag* <script> yang biasanya ditulis di dalam *page* HTML dan pada sistem ini terdapat pada bagian *view* di Laravel. Terdapat 2 fungsi yang akan kita pakai untuk mendapatkan posisi *user* pada sistem. Yang pertama adalah getCurrentPosition() yang berguna untuk mendapatkan nilai *latitude* dan *longitude user* yang akan berguna untuk pencarian lokasi. Lalu kita mendeklarasi sebuah *variable* baru untuk memanggil fungsi yang kedua yaitu Geocoder(). Pencarian lokasi dengan metode seperti ini sering

juga disebut dengan *Reverse Geocoding* yang dapat dilihat lebih detail pada laman Google Map JavaScript API.

```
navigator.geolocation.getCurrentPosition(showPosition, showError);
geocoder = new google.maps.Geocoder();
geocoder.geocode({'latLng': latlon}, function(results, status)
```

Code 4. 1 Penggunaaan Google Map Javascript API

4.3.5.2 Implementasi Penggunaan DOMPPDF

Library ini bertujuan untuk membuat sebuah file PDF yang dibentuk dari perubahan halaman HTML. Cara menginstallnya cukup menggunakan *composer install* untuk *library* Laravel lalu mendeklarasikannya dengan perintah berikut yang bertujuan untuk memasukan isi variable data ke dalam *view* "signaturepdf" lalu menentukan ukuran pdf dengan funsgi setPaper().

```
$pdf = PDF::loadView('signaturepdf', $data)-
>setPaper('a7', 'landscape');
```

Code 4. 2 Penggunaan DOMPPDF

4.3.5.3 Implementasi Penggunaan Minisign

Minisign adalah suatu *library* yang digunakan untuk membuat dan menghasilkan kunci serta menjalankan algoritma Ed25519 yaitu skema tanda tangan EdDSA menggunakan SHA-512 dan Curve 25519.

Penggunaan Minisign pada sistem ini terpanggil di 3 controller berbeda-beda, yaitu controller Login, controller enkripsi dan controller verifikasi.

4.3.5.4 Implementasi Penggunaan Simple QR-Code

Untuk menampilkan *QR-Code* pada hasil tanda tangan, maka kita perlu membuatnya terlebih dahulu dengan sebuah *library* untuk Laravel yaitu *Simple QR-CODE*. Cara penggunaan nya sangat mudah, hanya perlu menambahkan 1 *line* seperti ini pada controller yang akan menghasilkan *QR-Code*.

```
$qrcode = QrCode::format('svg')->generate('https://clauspeter.id/signature/' . $id);
```

Code 4. 3 Penggunaan simple QR code

Fungsi generate() diatas menampung data alamat web yang akan dituju.

BAB V
UJI COBA DAN EVALUASI

Pada bab ini berisi penjelasan mengenai skenario uji coba yang dilakukan dan evaluasi terhadap sistem Tanda Tangan Online terhadap keamanan data tanda tangan dan fungsionalitas web. Mekanisme uji coba dilakukan dengan mengikuti serangkaian skenario uji coba yang dibuat. Bagian akhir dari bab ini akan membahas mengenai evaluasi dari serangkaian uji coba yang telah dilakukan

5.1 Lingkungan Pengujian

Lingkungan yang digunakan untuk pengujian memiliki spesifikasi perangkat keras dan perangkat lunak yang ditunjukkan pada Table 5.1.

Tabel 5. 1 Spesifikasi perangkat keras dan perangkat lunak

Perangkat	**Jenis Perangkat**	**Spesifikasi**
Perangkat Keras (PC)	Prosesor	AMD Ryzen 7 Mobile 3700U
	Memori	8 GB
Perangkat Keras(Server)	Prosesor	-
	Memori	1 GB
Perangkat Lunak	Sistem Operasi	Windows 10
	Perangkat Pengembang	IDE Visual Studio Code
	Bahasa Pemrograman	Laravel, PHP

5.2 Data Pengujian

Pada tugas akhir ini, yang menjadi data untuk pengujian adalah tanda tangan yang sudah di enkripsi oleh aplikasi. Tanda tangan yang sudah ter-enkripsi harus dapat memenuhi aspek securitas yang sudah disebutkan dalam Bab I.

Pengujian akan meliputi tanda tangan yang dilakukan secara manual, ataupun ditempelkan dalam bentuk foto, dengan size yang berbeda.

5.3 Skenario Uji Coba Fungsionalitas

Uji coba Fungsionalitas merupakan sebuah pengujian yang dilakukan untuk menguji fungsi Web Sistem Tanda Tangan Online yang telah dibuat.

5.3.1 Skenario Uji Coba (UJ-F01) – Mendapatkan Tanda Tangan Dalam Bentuk Enkripsi

Uji coba ini dijalankan dengan tujuan untuk menguji fungsionalitas dari *website* pada halaman pembuatan tanda tangan. Pada halaman ini, terdapat page HTML dengan bentuk *form* yang harus diisi dan pilihan untuk membuat tanda tangan atau meng-*upload* tanda tangan. *Outputnya* adalah pembuatan tanda tangan *user* yang didapatkan oleh *user* dalam bentuk ternkripsi dan dikirimkan lewat email. Skenario akan dijelaskan lebih detail pada berikut.

Tabel 5. 2 Skenario uji coba

ID	**UJ-F01**
Nama	Uji Coba Mendapatkan Tanda Tangan Dalam Bentuk Enkripsi
Tujuan Uji Coba	Menguji Fungsionalitas *website* untuk mengirimkan tanda tangan terenkripsi kepada email *user*
Kondisi Awal	Membuka fitur Buat Tanda Tangan

Skenario	1. Membuka halaman web 2. *Login* sebagai pengguna 3. Membukan halaman Buat Tanda Tangan 4. Mendapatkan Email berisi tanda tangan terenkripsi
Masukan	1. Memasukan *input* dari form 2. Menggambar tanda tangan/meng-*upload* tanda tangan.
Keluaran	Tanda tangan terenkripsi yang dikrimkan melalui email

Untuk pengujian keamanan data saat pengiriman melalui email, maka saat pengujian pada skenario ini terdapat upaya untuk menggunakan jalur http sehingga traffic dapat disaring dengan *Wireshark*. Hasil yang didapat dengan wireshark terdapat pada gambar dibawah ini yang jika kita ubah dari bentuk *base64* maka gambar tanda tangan akan didapatkan.

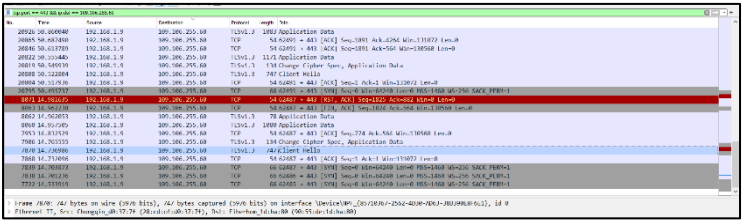
```
-----------------------------4990544533943152662508852262
Content-Disposition: form-data; name="signed"


3de5xN5f4H8IXcJSKJyK0MxqChi0giHelCJSG3hjAIyaVDoURCKEI1Uy7VSXIoJERilFw
KnThExEEuoVAon98f3996PXvPzey9n7Wevdb+vF+veZ3T7DXr+c5j5vnMs/
aznmWBiIiIImaZLoCIiMgPGKhEREQaMFCJiIg0YKASERFpwEAlIiLSgIFKRESkAQOViIh
IAwYqERGRBgxUIiIiDRioREREGjBQiYiINGCgEoXg1Clg48bMP5YtM10dEZnEQCXKoeRk
wLKy/
yhdGkhJMV0pEZnAQCXKgWefVaFZuTKmQmBj8kZAAFCyojqlRA5g61XTVROQmBirRJQSG6c
yZ2R+bmgrEx6vj69VpXHxFFDwYqURbCDVNbWhp
QvDhDlShWMFCJMjF4cGRhamOoEsUOBipROt266QlTW1oaUKSInG/
8+MjPR0TRiYFKFGD2bBWmycn6zjtqlJzznnv0nZOIogsDlej/rVmjwrR7dz3nTk4GLE+
vt7zElH0YKASAdi5UzZlsCygd2/R9YKAgUsw7exaoW1cCr2VL59qwL
CB/
fmfOT0TmMVAp5rVsKWFXt64z5EnNdTn1Ps1b5HjzrXBhGZw0ClmNa1q4TcNdfIZV8nVa8ubW3d6
mw7RGQGA5Vi1owZkQZOVn9H8xdSpLYsBGPBQi6kLjwQtpbhGF6l+PTTkhb9mdOgmOAvLGr2kvbffdq
c9InIXA5Vi0tNPS7g1a+Zem/buS6NHHu9cmEbmHgUox59ZqtvsuXl5SMDOf/8avca/
eRR6TNO+5wr00icg8DlWLO/
```

Gambar 5. 2 Data png dalam bentuk base64 encoding

Gambar 5. 1 Tanda tangan yang ditampilkan

Lalu saat menggunakan jalur https, *wireshark* tidak dapat mendeteksi method *post* dan *get* pada aplikasi sehingga upaya tetap dilakukan dan tetap tidak terdapat jejak di dalam *traffic*.



Gambar 5. 3 Monitoring https pada wireshark

Tanda tangan yang sudah dibuat dan terenkripsi datanya akan masuk dalam *database* sistem. Dalam database terdapat detail posisi serta lokasi *user* dan detail – detail lainnya yang dapat dilihat pada gambar di bawah ini

countryName	countryCode	regionCode	regionName	cityName	latitude	longitude	perihal
		Jakarta	Jakarta				
eyJpdiI6ImhlYXgxOGpqVWFNV3doNTVBVEgwa2c9PSIsInZhbH...	ID	Daerah Khusus Ibukota Jakarta	Daerah Khusus Ibukota Jakarta	Kota Jakarta Utara	-6.138884407001893	106.91921084549878	Tanda tangan
eyJpdiI6Ik9yQy92MFp2anh5SDJaeHFzc1RwM1E9PSIsInZhbH...	none	none	none	none	0	0	Tanda tangan
eyJpdiI6Inh3YzNTYWFqQWthUGVyRDdRWitSb2c9PSIsInZhbH...	ID	Daerah Khusus Ibukota Jakarta	Daerah Khusus Ibukota Jakarta	Kota Jakarta Utara	-6.1388760815267815	106.91921629500115	Tanda tangan
eyJpdiI6IngyREVLVEM5WDZ2RlhINDc2VjRreGc9PSIsInZhbH... Original length 188	ID	Daerah Khusus Ibukota Jakarta	Daerah Khusus Ibukota Jakarta	Kota Jakarta Utara	-6.1388760815267815	106.91921629500115	Tanda tangan
eyJpdiI6IlJ1UUoza1IDUFE0N1E4TmhBODY2bGc9PSIsInZhbH...	ID	Jawa Timur	Jawa Timur	Kota Surabaya	-7.3143107	112.7986227	Persetujuan TA
eyJpdiI6InZvYkxocUNnV2U2Z3NKMFdZVGdnY3c9PSIsInZhbH...	ID	Daerah Khusus Ibukota Jakarta	Daerah Khusus Ibukota Jakarta	Kota Jakarta Utara	-6.138867933854077	106.91921320157711	Tanda tangan
eyJpdiI6IlRvUjFGTFBLZUttY1BCa0xZUE5CeUE9PSIsInZhbH...	ID	Jawa Timur	Jawa Timur	Kota Surabaya	-7.3143107	112.7986227	Persetujuan TA
eyJpdiI6Ik95YmNxSlBReTNSUUwvajVSa05rMGc9PSIsInZhbH...	ID	Daerah Khusus Ibukota Jakarta	Daerah Khusus Ibukota Jakarta	Kota Jakarta Utara	-6.138867933854077	106.91921320157711	Tanda tangan
eyJpdiI6IktrUjJqaVlldzhoZzFtVkFzL3hqN3c9PSIsInZhbH...	ID	Daerah Khusus Ibukota Jakarta	Daerah Khusus Ibukota Jakarta	Kota Jakarta Utara	-6.138867933854077	106.91921320157711	Tanda tangan

Gambar 5. 4 Database sistem

Dan yang terakhir adalah hasil tanda tangan yang diterima oleh *user* adalah seperti ini

Gambar 5. 5 Hasil tanda tangan yang diterima user

5.3.2 Skenario Uji Coba (UJ-F02) – Mendapatkan Tanda Tangan Dalam Bentuk Dekripsi

Uji coba ini dijalankan dengan tujuan untuk menguji fungsionalitas dari website pada halaman *Decrypt File*. Pada halaman ini, terdapat page HTML dengan bentuk form yang harus diisi dan outputnya adalah pembuatan tanda tangan user yang didapatkan oleh *user* dalam bentuk terdekripsi dan dikirimkan lewat email. Skenario akan dijelaskan lebih detail pada berikut

Tabel 5. 3 Skenario uji coba mendapatkan tanda tangan dalam bentuk terdekripsi

ID	**UJ-F02**
Nama	Uji Coba Mendapatkan Tanda Tangan Dalam Bentuk terdekripsi
Tujuan Uji Coba	Menguji Fungsionalitas website untuk mengirimkan tanda tangan terdekripsi kepada email *user*
Kondisi Awal	Membuka fitur *Decrypt File*

Skenario	1. Membuka halaman web 2. *Login* sebagai pengguna 3. Membuka Halaman *Decrypt File* 4. Mendapatkan Email berisi tanda tangan terdekripsi
Masukan	1. Memasukan *input* dari form 2. Memasukan tanda tangan terenkripsi
Keluaran	Tanda tangan terdekripsi yang dikrimkan melalui email

Bentuk tanda tangan yang sudah terdekripsi memiliki banyak unsur yang tercatat secara fisik seperti *longitude*, *latitude*, waktu pembuatan, serta QR-code yang dapat mengarahkan ke detail tanda tangan.

Gambar 5. 6 Hasil tanda tangan

5.3.3 Skenario Uji Coba (UJ-F03) – Memverifikasi Tanda Tangan User

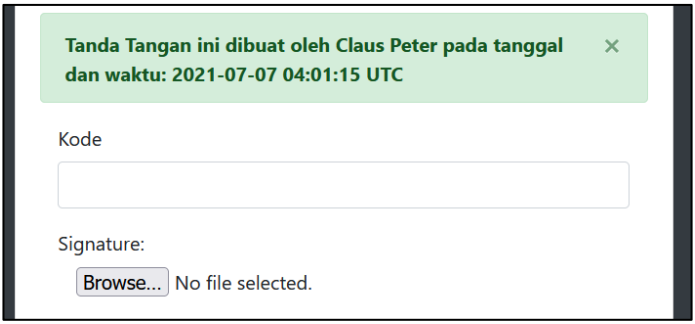
Uji coba ini dijalankan dengan tujuan untuk menguji fungsionalitas dari *website* pada halaman *Verify File*. Pada halaman ini, terdapat page HTML dengan bentuk form yang harus diisi dan outputnya adalah notifikasi detail tanda tangan

serta status terverifikasi. Skenario akan dijelaskan lebih detail pada berikut

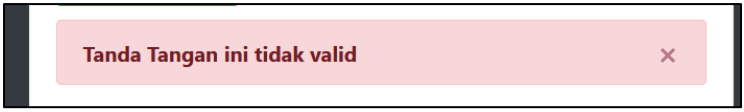
Tabel 5. 4 Skenario uji coba memverifikasi tanda tangan user

ID	**UJ-F03**
Nama	Uji Coba Memverifikasi Tanda Tangan *User*
Tujuan Uji Coba	Menguji Fungsionalitas website untuk mengirimkan notifikasi tanda tangan yang terverifikasi
Kondisi Awal	Membuka fitur *Verify File*
Skenario	1. Membuka halaman web 2. *Login* sebagai pengguna 3. Membuka Halaman *Verify File* 4. Mendapatkan notifikasi tanda tangan terverifikasi
Masukan	1. Memasukan *input* dari form 2. Memasukan tanda tangan terdekripsi
Keluaran	Notifikasi status tanda tangan

Hasil dari notifikasi akan memunculkan 2 kondisi yaitu apabila tanda tangan terverifikasi dan satu lagi ketika tanda tangan tidak terverifikasi.



Gambar 5. 7 Tanda tangan terverifikasi



Gambar 5. 8 Tanda tangan tidak terverifikasi

5.3.4 Skenario Uji Coba (UJ-F04) – Scan *QrCode* Tanda Tangan

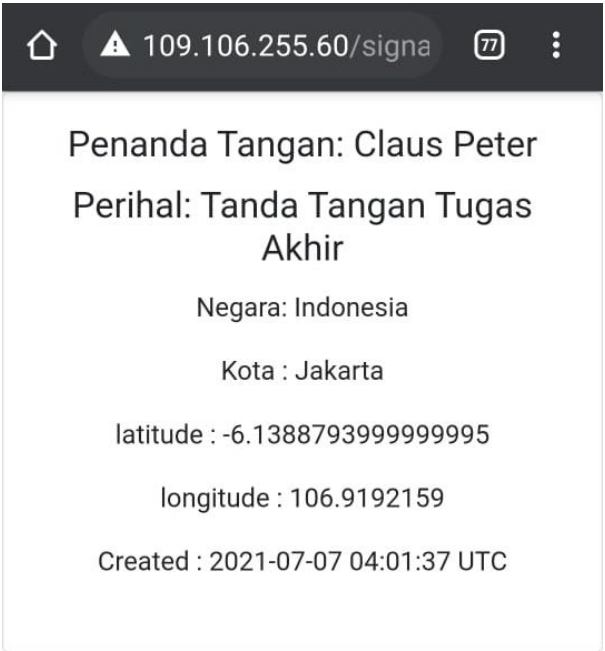
Uji coba ini dijalankan dengan tujuan untuk menguji fungsionalitas dari website dalam menerima *route* dari *QrCode*. Pada halaman ini, terdapat *page* HTML dengan detail tanda tangan. Skenario akan dijelaskan lebih detail pada berikut

Tabel 5. 5 Skenario uji coba scan QR code tanda tangan

ID	**UJ-F04**
Nama	Uji Coba *Scan QrCode* Tanda Tangan
Tujuan Uji Coba	Menguji Fungsionalitas *website* untuk membaca *QRcode* tanda tangan
Kondisi Awal	*Scan QR Code*
Skenario	*1. Scan QR Code*

	2. Mendapatkan halaman detail tanda tangan
Masukan	*1. Scan Qr Code*
Keluaran	Halaman detail tanda tangan

Setelah *user* melakukan *scan* pada *QR Code* yang tertera pada tanda tangan maka *device* akan mengarahkan kepada alamat web yang menampung detail tanda tangan *user*.



Gambar 5.9 Hasil Scan QRCode Tanda Tangan

5.4 Evaluasi Hasil Uji Coba

Pada bagian sebelumnya, telah dilakukan uji coba terhadap sistem yang dibuat. Uji coba yang telah dilakukan berkenaan dengan uji coba fungsionalitas aplikasi Tanda Tangan Digital. Berikut ini merupakan hasil uji coba dari Tugas Akhir ini.

Tabel 5. 6 Hasil uji coba

No	Kode Uji Coba	Evaluasi
1	UJ-F01	Dapat Mengirimkan Tanda Tangan dalam Bentuk Terenkripsi
2	UJ-F02	Dapat Mengirimkan Tanda Tangan dalam bentuk Terdekripsi
3	UJ-F03	Dapat Memverifikasi Tanda Tangan yang Didapat
4	UJ-F04	Dapat Menampilkan Detail Tanda Tangan Melalui *QR-Code*

BAB VI
KESIMPULAN DAN SARAN

Bagian terakhir yang dibahas dalam buku ini adalah kesimpulan dan saran. Kesimpulan yang dijabarkan di bawah ini di dapatkan berdasarkan dari pengamatan uji coba sitem. Selain kesimpulan, dijabarkan pula saran yang di harapkan dapat berguna untuk pengembangan sistem dan penelitian selanjutnya.

6.1 Kesimpulan

Kesimpulan yang di dapatkan berdasarkan Hasil Uji coba yang telah didapat adalah sebagai berikut :

1. *Website* dengan status HTTPS dapat mengenkripsi dan menyembunyikan *traffic data* dengan baik sehingga sulit untuk diretas oleh pihak pihak lain.
2. Dibutuhkan upaya yang sangat sulit untuk mendapatkan tanda tangan *user* oleh pihak ketiga yang tidak sebanding dengan resikonya.
3. Pengamanan tanda tangan digital tidak cukup hanya menggunakan 1 metode enkripsi dikarenakan adanya celah-celah dan kelebihan atau kelemahan tiap jenis algoritma
4. Pengiriman Email menggunakan enkripsi TLS masih bisa dipercaya dikarenakan memiliki performa mirip dengan HTTPS;

6.2 Saran

Saran yang diberikan dari hasil uji coba dan evaluasi pada tugas akhir ini adalah sebagai berikut:

1. Sistem dapat menggunakan ekstensi berbayar untuk mencegah *user* dalam menggunkan vpn untuk mengakses lokasi.
2. Menggunakan server dengan performa yang lebih baik agar sistem dapat berjalan lebih cepat.
3. Menggunakan *library* lain yang lebih akurat khususnya di Indonesia dibandingkan dengan Google Maps API Javascript dikarenakan dengan fiturnya yang berbayar.
4. Menggambar tanda tangan lebih bagus digunakan dengan *drawing pen digital.*
5. Sistem menerbitkan tanda tangan dibawah satu instansi yang diakui oleh dunia untuk keabsahannya, seperti menggunakan CA (*Certificate Authority*).

DAFTAR PUSTAKA

[1] R. Kaur and A. Kaur, "Digital signature," *Proc. Turing 100 - Int. Conf. Comput. Sci. ICCS 2012*, pp. 295–301, 2012, doi: 10.1109/ICCS.2012.25.

[2] R. Farrow, "Public Key Infrastructure Basics," *Edpacs*, vol. 26, no. 11, pp. 17–17, 1999, doi: 10.1201/1079/43239.26.11.19990501/30209.6.

[3] L. Review, "AES (Advanced Encryption Standard) and RSA (Rivest – Shamir – Adleman) Encryption on Digital Signature Document : A Literature Review," vol. 2, no. 1, pp. 1–3, 2020.

[4] M. Husni *et al.*, "Security audit in cloud-based server by using encrypted data AES -256 and SHA-256," *IOP Conf. Ser. Mater. Sci. Eng.*, vol. 830, no. 3, 2020, doi: 10.1088/1757-899X/830/3/032015.

[5] Imamah, A. Djunaidy, and M. Husni, "Penerapan AES Untuk Otentikasi Akses Cloud Computing," *Ilm. SimanteC*, vol. 4, no. 1, pp. 3–5, 2014, [Online]. Available: http://www.ncbi.nlm.nih.gov/pubmed/22352137.

[6] M. Stauffer, *Laravel: Up & Running*, vol. 53, no. 9. 2017.

[7] The phpMyAdmin devel team, "phpMyAdmin Documentation," 2014.

[8] Tutorialspoint, "PhP Hypertext Processor," *Tutorials Point Pvt. Ltd.*, pp. 1–13, 2019.

[9] B. Tulu, D. Lafky, B. N. Hilton, T. A. Horan, H. Li, and S. Chatterjee, "Design and Implementation of a Digital Signature Solution for a Healthcare Enterprise," *10th Am. Conf. Inf. Syst.*, no. August, pp. 295–303, 2004.

[10] I. W. Ariadi, "Bentuk-Bentuk Digital Signature Yang Sah Dalam Transaksi Elektronik Di Indonesia," *J. Magister Huk. Udayana (Udayana Master Law Journal)*, vol. 5, no. 1, p. 175, 2016, doi: 10.24843/jmhu.2016.v05.i01.p16.

[11] E. Publishing, S. N. Kumar, and M. T. Science,

"Review on Network Security and Cryptography," vol. 3, no. 1, pp. 1–11, 2015, doi: 10.12691/iteces-3-1-1.

[12] A. M. Abdullah, "Advanced Encryption Standard (AES) Algorithm to Encrypt and Decrypt Data," 2017.

[13] "What Is a Certificate Authority (CA)? - SSL.com." https://www.ssl.com/faqs/what-is-a-certificate-authority/ (accessed Jan. 30, 2021).

[14] "What is Public Key Infrastructure | How Does PKI Work? | Security Wiki." https://doubleoctopus.com/security-wiki/digital-certificates/public-key-infrastructure/ (accessed Jul. 23, 2021).

[15] "Advanced Encryption Standard - Tutorialspoint." https://www.tutorialspoint.com/cryptography/advanced_encryption_standard.htm (accessed Jul. 23, 2021).

BIODATA PENULIS

Penulis dilahirkan di Padang, 16 Oktober 1998 dengan nama lengkap Claus Peter Christo Hutabarat. Dalam pendidikan formalnya, penulis pernah mengenyam di SD Don Bosco 2, SMPK Penabur 4, SMAK 5 Penabur, dan SMA TUNAS GADING. Setelah itu melanjutkan sebagai mahasiswa Departemen Informatika ITS pada tahun 2017. Selama masa perkuliahan, penulis aktif dalam Bermusik dan Berorganisasi, tergabung dalam band kampus bernama Kampus Kota, menjadi asisten dosen, staff organisasi mahasiswa jurusan. Selain itu penulis juga pernah melakukan magang di beberapa perusahaan Indonesia seperti Ternaknesia, Indosat, dan Pertamina. Penulis sangat terbuka bagi para pembaca yang ingin berdiskusi maupun memberikan saran atau kritik mengenai Tugas Akhir ini melalui email claus.peter54@gmail.com atau nomor 082219171116.