

# Pengembangan Aplikasi Sederhana untuk Verifikasi Tanda Tangan Digital pada Gambar yang Dibagikan di Sosial Media X

Azmi Alfatih Shalahuddin - 13520158

Program Studi Teknik Informatika  
Sekolah Teknik Elektro dan Informatika  
Institut Teknologi Bandung, Jalan Ganesha 10 Bandung  
E-mail: azmi.alfatihs@gmail.com

**Abstract**— Sosial Media X salah satu sosial media paling populer pada 2024 ini. Sosial Media ini merupakan media yang mengedepankan nilai *free-to-speech* sehingga semua pengguna di dalamnya diperbolehkan memberikan pandangannya dalam hal apapun. Terdapat dampak negatif dari konsep tersebut, yaitu sangat mudahnya berita tersebar tanpa ada kejelasan identitas pengirim dan integritas informasi. Penelitian ini dilakukan dengan tujuan untuk menyelesaikan permasalahan tersebut dengan mengembangkan aplikasi sederhana untuk verifikasi pengirim dan integritas pada informasi gambar yang dibagikan dengan menggunakan konsep tanda tangan digital atau *digital signature* (DS). Hasil dari penelitian ini berupa *endpoint* server yang berfungsi autentikasi pengguna, memberikan dan memverifikasi tanda tangan digital pada suatu gambar. Dibuat juga rancangan model sistem yang diharapkan dapat dikembangkan lebih lanjut.

**Kata kunci**— Tanda Tangan Digital, Verifikasi Gambar, Sosial Media, Integritas, Autentikasi.

## I. PENDAHULUAN

### A. Latar Belakang

Sosial Media X merupakan salah satu sosial media yang paling populer pada saat ini. Salah satu keunikannya, Sosial Media X mengedepankan *value* “*free-to-speech*” atau kebebasan berpendapat sehingga semua penggunanya diperbolehkan memberikan pandangannya dalam hal apapun. Karena itu, persebaran informasi pada sosial media X sangatlah cepat dan bebas.

Terdapat berbagai sisi positif dan negatif dari adanya hal tersebut. Contoh sisi positifnya yaitu penggunaannya dapat mengetahui berbagai berita, baik regional maupun global, secara cepat. Sisi negatifnya, berita tersebut tidak dapat dijamin integritasnya. Pengguna tidak dapat mengetahui dengan jelas siapa yang pertama kali menyebarkan suatu berita. Selain itu, pengguna juga tidak dapat mengetahui apakah berita tersebut sudah diubah dalam penyebarannya atau masih original.

Dari segi jenis media yang dibagikan, sosial media X berisi media dengan jenis yang sangat variative. Hal ini terjadi karena sosial media X memiliki visi untuk menjadi aplikasi super. Artinya, akan ada begitu banyak fitur dan variasi media yang

terus dikembangkan untuk dapat dibagikan di dalamnya, mulai dari teks, gambar, GIF, video panjang, video pendek, penyiaran langsung, dan lain sebagainya. Akan tetapi, media yang termasuk paling sering digunakan oleh pengguna ialah media berjenis teks dan gambar.

### B. Rumusan Masalah

Pada makalah ini, permasalahan yang difokuskan ialah mengenai kurangnya integritas dan keaslian (*authenticity*) dari media berjenis gambar pada sosial media X. Permasalahan tersebut muncul karena media gambar dapat diunduh dan diunggah secara bebas dari satu pengguna ke pengguna lainnya. Pada prosesnya, tidak dapat dipastikan jaminan keaslian gambar tersebut.

Oleh karena itu, diperlukan mekanisme agar seluruh pengguna sosial media X mampu melakukan verifikasi pemilik gambar tersebut dan jaminan bahwa gambar tersebut tidak dimodifikasi pada prosesnya.

### C. Tujuan

Tujuan dari adanya makalah ini ialah membuat suatu prototipe penjaminan *message authentication* (autentikasi pemilik pesan) dan *integrity* (keaslian) dari pesan. Prototipe tersebut memiliki fungsi sebagai berikut : autentikasi (pendaftaran akun dan *sign in*), penandatanganan gambar (*digital signing*), dan verifikasi pemilik gambar (*owner verification*).

### D. Manfaat

Diharapkan dengan adanya makalah ini, dapat dibentuk sebuah sistem yang membuat pengguna mendapatkan informasi terkhusus media gambar yang lebih terjamin pemilik dan integritasnya.

## II. DASAR TEORI

### A. Layanan Keamanan Tanda Tangan Digital

Secara umum, terdapat empat jenis layanan keamanan yang disediakan oleh kriptografi, yakni kerahasiaan pesan (*confidentiality*), keaslian pesan (*data integrity*), otentikasi (*authentication*), dan anti-penyangkalan (*non-repudiation*). Keaslian pesan dapat dicapai dengan mekanisme *message authentication code* (MAC), otentikasi dapat dicapai dengan menggunakan MAC dan tanda tangan digital, sedangkan anti-penyangkalan dapat dicapai hanya dengan menggunakan tanda tangan digital.

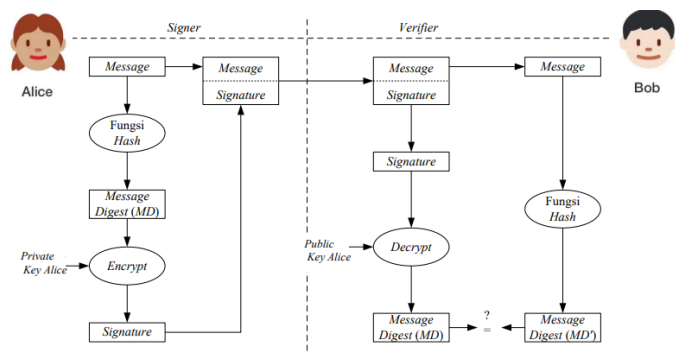
Tanda tangan mempunyai karakteristik sebagai berikut :

1. Berupa bukti otentik
2. Tidak dapat hilang
3. Tidak dapat dipindah untuk digunakan ulang
4. Dokumen yang ditandatangani tidak dapat diubah
5. Tidak dapat disangkal

Nilai kriptografis dari tanda tangan digital bergantung pada isi pesan dan kunci yang hanya dimiliki oleh pemilik tanda tangan. Tanda tangan digital selalu berbeda-beda antara satu pesan dengan pesan lain, dan juga antara satu kunci dengan kunci yang lain.

Proses-proses dalam tanda tangan digital dapat dibagi menjadi dua, yakni menandatangani pesan (*signing*) dan memverifikasi pesan (*verification*).

Untuk pesan yang tidak perlu dirahasiakan, cara menandatangani pesan ialah dengan mengombinasikan fungsi hash dan kriptografi kunci-publik. Diagram proses tanda tangan digital menggunakan kombinasi kriptografi kunci public dan fungsi hash dapat dilihat pada Gambar 2.1.



Gambar 2.1. Proses Tanda Tangan Digital untuk Pesan tidak Rahasia (sumber : “Tanda Tangan Digital” (Munir, R., 2024))

Contoh algoritma kriptografi yang dapat digunakan dalam tanda tangan digital ialah RSA. Langkah Langkah proses pemberian dan verifikasi tanda tangan digital dengan algoritma RSA ialah sebagai berikut:

1. Pemberian Tanda Tangan (*signing*):
  - a. Pengirim menghitung nilai hash dari pesan M.

$$h = H(M)$$

- b. Pengirim mengenkripsi  $h$  dengan kunci privatnya menggunakan persamaan enkripsi RSA dan menghasilkan signature S:

$$S = h^{SK} \bmod n, \quad n = pq$$

- c. Pengirim mentransmisikan  $M + S$  ke penerima

### 2. Verifikasi Tanda Tangan (*verifying*):

- a. Penerima menghitung nilai hash dari pesan M yang diterima:

$$h = H(M)$$

- b. Penerima melakukan deskripsi terhadap tanda-tangan S dengan kunci public si pengirim (PK) menggunakan persamaan deskripsi RSA:

$$h' = S^{PK} \bmod n$$

- c. Penerima membandingkan  $h$  dengan  $h'$ . Jika  $h=h'$  maka tanda tangan digital adalah otentik. Jika tidak sama, maka tanda tangan tidak otentik sehingga pesan atau pengirimnya dianggap tidak asli lagi.

### B. Sertifikat Digital

Seperti yang diketahui, sistem tanda tangan digital menggunakan kriptografi kunci-publik, yang mensyaratkan pengguna memiliki sepasang kunci : kunci privat dan kunci publik. Kunci privat bersifat rahasia, hanya diketahui oleh pemilik, tidak dibagi kepada pihak lain, tetapi kunci publik tersedia untuk umum.

Permasalahan yang muncul dari hal tersebut ialah kunci publik tidak mempunyai suatu kode yang mengidentifikasi pemiliknya. Selain itu, pihak lain dapat menyalahgunakan kunci public yang bukan miliknya untuk *impersonation attack*

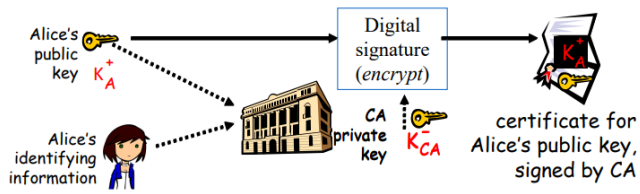
Sertifikat digital muncul untuk menjadi solusi dari permasalahan tersebut. sertifikat digital ialah dokumen yang mengikat kunci public dengan informasi pemiliknya. Sertifikat digital dikeluarkan oleh pemegang otoritas sertifikasi yang disebut *certification authority* (CA).

Informasi minimal dalam sertifikat digital ada empat, yaitu :

1. Identitas subjek
2. Kunci public subjek
3. Nama CA
4. Tanda tangan CA.

Selain itu, dapat ditambahkan informasi lain seperti waktu kadaluwarsa dan sebagainya.

Alur proses mendapatkan sertifikat digital dapat dilihat pada gambar 2.2.



Gambar 2.2. Proses Pengguna mendapatkan Sertifikat Digital  
Sumber Gambar : Group 11 Members (Rackenee Rhule et al, Digital Certificates)

### III. RANCANGAN SOLUSI

Akan dibangun sebuah sistem tanda tangan digital beserta CA untuk verifikasi media gambar pada sosial media X.

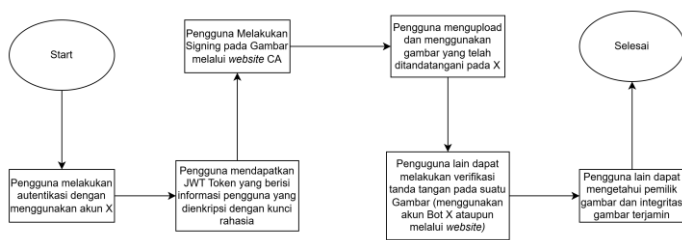
#### A. Model Sertifikat Digital

Dibutuhkan sebuah model CA dan mekanisme semacam sertifikat digital yang dapat digunakan oleh seluruh pengguna sosial media X untuk melakukan penandatanganan digital dan juga verifikasi autentikasi. cara kerja CA dan struktur sertifikat digital cukup berbeda dan disederhanakan dari standar yang ada untuk menyesuaikan dan menspesifikkan fungsi sesuai kebutuhan.

CA yang dibangun akan memiliki fungsionalitas sebagai berikut, yakni :

1. Autentikasi dan Pendaftaran Pengguna
2. Penandatanganan Gambar
3. Verifikasi Gambar

Diagram alir dari fungsi model sertifikat digital yang akan dibangun dapat dilihat pada Gambar 3.1.



Gambar 3.1. Diagram Alir Model Sertifikat Digital

#### 1) Autentikasi dan Pendaftaran Pengguna

Proses autentikasi dan pendaftaran pengguna bertujuan agar server dapat menyimpan informasi pengguna beserta kunci rahasia yang disimpan dan dapat diproses secara otomatis oleh server.

Pengguna akan melakukan registrasi dengan memasukkan *username* X. Akan diperlukan verifikasi untuk memastikan bahwa pendaftar benar-benar merupakan pemilik dari akun X

tersebut. Setelah registrasi berhasil, maka pada server akan terdaftar *username* beserta kunci rahasia yang tidak akan keluar dari server.

Proses pendaftaran pada dasarnya dapat dilewati apabila sistem ini benar-benar terhubung dengan sosial media X. Dengan begitu, pengguna dapat langsung melakukan *sign in* secara langsung dan akan mendapatkan JSON *web token* (JWT) Setelah berhasil melakukan *sign in*. Akan tetapi, JWT tersebut harus dienkripsi menggunakan *secret key* yang kuat, disimpan dengan aman pada server, dan terdapat mekanisme *refresh token* untuk meningkatkan keamanan.

JWT ini berisi informasi tentang siapa pengguna yang akan melakukan *digital signing*, seperti *username* X. Server akan mengirim token untuk digunakan pengguna dalam jangka waktu tertentu.

#### 2) Penandatanganan Gambar

Alur penandatanganan gambar oleh pengguna ialah sebagai berikut.

1. Pengguna mengupload gambar pada platform sistem sertifikat digital.
2. Pengguna menyertakan token JWT yang sebelumnya didapat dari proses login.
3. Server memverifikasi token JWT
4. Apabila JWT valid, server memuat kunci pribadi
5. Server memuat *private key* yang digunakan untuk menandatangani gambar.
6. Server membuat tanda tangan digital menggunakan algoritma RSA-SHA256 dan *private key*.
7. Server menyimpan tanda tangan digital dan identitas pengguna dari JWT dalam database yang berhubungan dengan ID file gambar.
8. Server mengirim response ke pengguna bahwa penandatanganan berhasil beserta tanda tangan digital.

#### 3) Verifikasi Tanda Tangan

Alur verifikasi tanda tangan digital dapat dilakukan dengan cara otomatis ataupun manual (sebagai alternatif). Verifikasi dapat dilakukan secara otomatis apabila sistem terhubung secara resmi dengan sosial media X sehingga terdapat fitur berupa tombol untuk mengecek tanda tangan gambar secara otomatis. Cara kedua, dapat dibuat sebuah akun bot yang dapat di-*summon* untuk memberikan *reply* secara otomatis mengenai siapa pemilik tanda tangan tersebut dan jaminan integritas dari gambar.

Apabila dilakukan secara manual, user dapat menggunakan platform sistem sertifikat digital, sebagaimana user apabila ingin melakukan penandatanganan digital. User akan mengirim gambar beserta *signature*. Prosesnya ialah sebagai berikut:

1. Pengguna mengunggah gambar beserta *signature* pada platform sistem sertifikat digital. (tanpa membutuhkan JWT)
2. Server memuat *public key* yang sesuai yang digunakan untuk verifikasi tanda tangan.
3. Server menggunakan algoritma RSA-SHA256 dan kunci public untuk verifikasi tanda tangan digital dengan gambar.
4. Jika verifikasi berhasil, server mencari informasi tentang siapa yang menandatangani gambar dari file *database*.
5. Server mengirim respons apakah verifikasi berhasil dan informasi tentang siapa yang menandatangani gambar.

### B. Rancangan Prototipe

Pada prototipe ini, akan dibentuk sebuah program yang hanya berupa program *back-end* yang berjalan pada server local. Server local ini dapat diuji fungsionalitasnya menggunakan aplikasi Postman.

Desain mengenai prototipe ini dipilih dengan mempertimbangkan *feasibility* dan kesulitan yang mungkin muncul apabila harus melakukan pembuatan akun bot X, tampilan front-end, ataupun melakukan *deployment* server secara nyata.

Meskipun hanya berupa server *local*, diharapkan *back-end* ini benar-benar berfungsi sebagaimana mestinya dan dapat diuji. Dengan begitu, *gap* antara prototipe yang akan dibuat dan rancangan sistem nyata hanyalah berupa *interface* kepada user, yang dapat berupa akun bot ataupun *front-end* platform yang menarik.

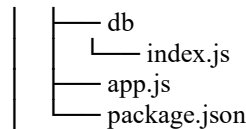
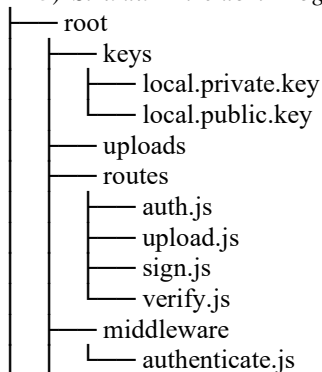
#### 1) Teknologi yang digunakan

Teknologi atau *framework* yang digunakan pada sistem ini ialah Express.js

#### 2) Fungsionalitas Server

- a. Autentikasi pengguna dengan JWT
- b. Manajemen unggahan gambar
- c. Penandatanganan gambar menggunakan *private key*
- d. Verifikasi tanda tangan gambar menggunakan *public key*

#### 3) Struktur Direktori Program



## IV. HASIL DAN PEMBAHASAN

### A. End-point Server

#### 1) Endpoint untuk autentikasi pengguna

- `/auth/register` (POST): Mendaftarkan pengguna baru dengan menyimpan username dan password yang telah di-hash ke dalam database.
- `/auth/login` (POST): Memverifikasi kredensial pengguna dan menghasilkan token JWT sebagai bukti autentikasi.

#### 2) Endpoint untuk mengelola unggahan gambar

- `/upload` (POST): Mengunggah gambar ke server dan menyimpannya di direktori uploads. Mengembalikan nama file yang telah diunggah.

#### 3) Endpoint untuk penandatanganan gambar

- `/sign` (POST): Memungkinkan pengguna untuk menandatangani gambar yang telah diunggah. Menghasilkan tanda tangan digital menggunakan *private key* statis dan menyimpannya bersama informasi pengguna di database.

#### 4) Endpoint untuk verifikasi tanda tangan gambar

- `/verify` (POST): Memverifikasi tanda tangan digital pada gambar yang telah ditandatangani. Menggunakan *public key* statis untuk memverifikasi tanda tangan dan mengembalikan informasi tentang pemilik gambar.

### B. Deskripsi Fungsionalitas

#### 1) Autentikasi Pengguna

Sistem berhasil mengautentikasi pengguna menggunakan JWT. Pengguna dapat mendaftar, login, dan mendapatkan token JWT yang valid.

#### 2) Upload Gambar

Pengguna dapat mengunggah gambar dengan endpoint `/upload`. Sistem menyimpan gambar dengan nama unik di direktori uploads.

#### 3) Penandatanganan Gambar

Pengguna dapat menandatangani gambar yang telah diunggah menggunakan endpoint `/sign`. Tanda tangan digital dihasilkan menggunakan *private key* statis dan disimpan di database bersama dengan informasi pengguna.

#### 4) Verifikasi Tanda Tangan

Pengguna dapat memverifikasi tanda tangan digital menggunakan endpoint `/verify`. Sistem berhasil memverifikasi integritas dan autentikasi gambar yang ditandatangani.

### C. Implementasi Program

Berikut merupakan beberapa sampel kode dari program yang diimplementasi.

#### 1) App.js

```
require("dotenv").config();

const express = require("express");
const routes = require("./routes");
const path = require("path");
const authRoutes = require("./routes/auth");
const app = express();
const fs = require("fs");
const PORT = process.env.PORT || 3000;

//to ensure the uploads directory exist
const uploadsDir = path.join(__dirname, "uploads");
if (!fs.existsSync(uploadsDir)) {
  fs.mkdirSync(uploadsDir);
}

const signaturesDir = path.join(__dirname, "signatures");
if (!fs.existsSync(signaturesDir)) {
  fs.mkdirSync(signaturesDir);
}

app.use(express.json());
app.use(express.urlencoded({ extended: true }));

const privateKeyPath = process.env.PRIVATE_KEY_PATH;
const publicKeyPath = process.env.PUBLIC_KEY_PATH;

const privateKey = fs.readFileSync(
  path.resolve(__dirname, privateKeyPath),
  "utf-8"
);
const publicKey = fs.readFileSync(
  path.resolve(__dirname, publicKeyPath),
  "utf-8"
);
console.log("Private Key Loaded:", privateKey ? "Yes" : "No");
console.log("Public Key Loaded:", publicKey ? "Yes" : "No");

app.locals.privateKey = privateKey;
app.locals.publicKey = publicKey;

app.use("/api", routes);
app.use("/auth", authRoutes);

app.get("/", (req, res) => {
  res.send("Welcome to Digital Signature App");
});

app.listen(PORT, () => {
  console.log(`Server is running on port ${PORT}`);
});
```

```
});
```

#### 2) routes/sign.js

```
const express = require('express');
const crypto = require('crypto');
const fs = require('fs');
const { signatures } = require('../db');
const router = express.Router();

router.post('/', authenticateToken, async (req, res) => {
  const { filename } = req.body;
  const privateKey = fs.readFileSync('local.private.key', 'utf8');
  const fileData = fs.readFileSync(`uploads/${filename}`);
  const signer = crypto.createSign('RSA-SHA256');

  signer.update(fileData);
  signer.end();

  const signature = signer.sign(privateKey, 'hex');

  await signatures.insert({ userId: req.user.userId, filename, signature });

  res.json({ message: 'File signed successfully', signature });
});
```

```
module.exports = router;
```

#### 3) Routes/verify.js

```
const express = require('express');
const crypto = require('crypto');
const fs = require('fs');
const { signatures } = require('../db');
const router = express.Router();

router.post('/', async (req, res) => {
  const { filename, signature } = req.body;
  const publicKey = fs.readFileSync('local.public.key', 'utf8');
  const fileData = fs.readFileSync(`uploads/${filename}`);
  const verifier = crypto.createVerify('RSA-SHA256');

  verifier.update(fileData);
  verifier.end();

  const isValid = verifier.verify(publicKey, signature, 'hex');

  if (isValid) {
    res.json({ message: 'Signature verified successfully' });
  } else {
    res.status(400).json({ message: 'Invalid signature' });
  }
});
```

```
});  
  
module.exports = router;
```

#### D. Keamanan

##### 1) Autentikasi

Penggunaan JWT memastikan bahwa hanya pengguna yang terautentikasi yang dapat mengakses fungsi kritis seperti penandatanganan dan verifikasi.

##### 2) Integritas Gambar

Tanda tangan digital memastikan bahwa gambar yang diunggah tidak dapat dimodifikasi tanpa terdeteksi.

##### 3) Nir Penyangkalan

Sistem menyimpan informasi tanda tangan dan pengguna di database, memastikan bahwa pengguna tidak dapat menyangkal telah menandatangani gambar.

#### E. Kinerja Sistem

Waktu respons untuk berbagai endpoint (registrasi, login, upload, sign, verify) diuji dan tercatat. Hasil menunjukkan bahwa sistem merespons dalam waktu yang wajar di bawah kondisi beban normal.

Berikut waktu respons dari tiap-tiap endpoint:

| Endpoint       | Waktu Response (ms) |
|----------------|---------------------|
| /api/sign      | 44                  |
| /api/upload    | 16                  |
| /api/verify    | 41                  |
| /auth/register | 279                 |
| /auth/login    | 83                  |

#### F. Potensi Perbaikan Sistem

Potensi yang dapat dilakukan untuk perbaikan selanjutnya ialah sebagai berikut :

1. Perbaikan proses autentikasi
2. Pembuatan *interface* ataupun akun Bot untuk otomatisasi tanda-tangan dan verifikasi.
3. Integrasi secara *real* terhadap X (cukup sulit dilakukan)
4. *Deployment* secara *real* dan penggunaan *cloud* untuk skalabilitas

### V. PENUTUP

#### A. Kesimpulan

Implementasi sistem penandatanganan dan verifikasi digital gambar berbasis Express.js dengan autentikasi JWT berhasil memenuhi tujuan utama yaitu memastikan integritas, autentikasi, dan nir penyangkalan data. Sistem menunjukkan performa yang baik di bawah kondisi beban normal.

Sistem ini memberikan manfaat bagi pengguna platform sosial media X dengan menyediakan cara yang andal untuk memverifikasi pemilik gambar dan memastikan integritas gambar yang diunggah. Hal ini membantu mencegah penyalahgunaan gambar dan meningkatkan kepercayaan di antara pengguna platform tersebut.

#### B. Saran

Untuk meningkatkan fungsionalitas dan penggunaan sistem, beberapa saran perbaikan dapat dipertimbangkan:

1) *Pembuatan Antarmuka Pengguna (Frontend Platform)*: Mengembangkan antarmuka pengguna yang ramah pengguna dan responsif menggunakan platform frontend seperti React.js atau Vue.js. Ini akan memudahkan pengguna dalam menggunakan fitur-fitur sistem.

2) *Akun Bot X yang Terhubung Langsung ke Server*: Memperkenalkan akun bot yang terhubung langsung ke server dapat menjadi shortcut penggunaan yang efisien bagi pengguna. Bot ini dapat melakukan tugas-tugas sederhana seperti penandatanganan dan verifikasi gambar secara otomatis.

3) *Penggunaan Cloud dan Deployment Server Secara Real*:

Memindahkan sistem ke lingkungan cloud seperti AWS, Azure, atau Google Cloud Platform dapat meningkatkan skalabilitas, keandalan, dan ketersediaan sistem secara keseluruhan. Ini juga memudahkan proses deployment dan manajemen infrastruktur.

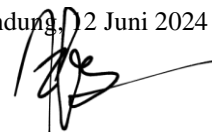
### REFERENSI

- [1] Munir, R. "Tanda Tangan Digital", Salindia Mata Kuliah IF4020 Kriptografi, Program Studi Teknik Informatika, Sekolah Teknik Elektro dan Informatika Institut Teknologi Bandung, 2024.
- [2] Munir, R. "Sertifikat Digital", Salindia Mata Kuliah IF4020 Kriptografi, Program Studi Teknik Informatika, Sekolah Teknik Elektro dan Informatika Institut Teknologi Bandung, 2024.
- [3] Munir, R. "Public Key Infrastructure", Salindia Mata Kuliah IF4020 Kriptografi, Program Studi Teknik Informatika, Sekolah Teknik Elektro dan Informatika Institut Teknologi Bandung, 2024.

### PERNYATAAN

Dengan ini saya menyatakan bahwa makalah yang saya tulis ini adalah tulisan saya sendiri, bukan saduran, atau terjemahan dari makalah orang lain, dan bukan plagiasi.

Bandung, 12 Juni 2024



Azmi Alfath Shalahuddin  
13520158

