Perancangan Sistem Kode QR Akses Masuk Kantor Menggunakan Tanda Tangan Digital dengan Algoritma Kriptografi Kurva Eliptik dan Fungsi *Hash* SHA-3 Keccak

Fikri Muhammad Fahreza
Sekolah Teknik Elektro dan Informatika
Institut Teknologi Bandung
Bandung, Indonesia
aafahreza76@gmail.com

***Abstrak*—Dalam era transformasi digital yang berkembang pesat, keamanan informasi dan identitas memegang peran sentral dalam menjaga integritas dan akses terhadap berbagai sumber daya terbatas. Sistem tradisional pengamanan akses pintu masuk telah menghadapi risiko kerusakan, kehilangan, dan pencurian kartu akses, sehingga diperlukan inovasi dalam keamanan akses. Teknologi *fingerprint* dan *face recognition* telah digunakan untuk mengamankan akses terbatas dengan tingkat keamanan yang tinggi, namun memiliki algoritma yang kompleks dan proses komputasi yang tinggi. Di sisi lain, kode QR telah banyak digunakan di berbagai aplikasi, akan tetapi proses verifikasi pada kode QR seringkali membutuhkan koneksi internet untuk pencocokan pesan yang terkandung dalam kode QR dengan infomasi yang terdapat pada *database*. Oleh karena itu, tugas akhir ini mengusulkan perancangan sistem kode QR akses masuk kantor dengan menggunakan tanda tangan digital. Metodologi yang digunakan dalam penyusunan tugas akhir ini adalah dengan studi pustaka, menganalisis permasalahan terkait metode akses yang ada, menganalisis kebutuhan, merancang solusi, melakukan implementasi, dan melakukan evaluasi dengan melakukan pengujian fungsional dan pengujian kepada calon pengguna. Berdasarkan hasil pengujian fungsional, didapatkan bahwa fungsionalitas sistem yang dibuat dapat berjalan dengan baik tanpa adanya kendala. Hasil pengujian UAT pun menunjukkan bahwa sistem yang telah dibuat dapat memenuhi kepuasan calon pengguna. Dengan memanfaatkan pendekatan *edge computing* untuk verifikasi tanda tangan digital pada kode QR, proses verifikasi dapat dilakukan langsung di perangkat pemindai tanpa perlu selalu terhubung dengan internet. Hal ini menjadikannya lebih responsif dan adaptif di berbagai situasi sehingga membuatnya menjadi sistem manajemen akses yang lebih andal.**

Keywords— Area terbatas, Kode QR, Kriptografi

I. PENDAHULUAN

Di era ancaman keamanan semakin kompleks dan beragam, program Keamanan Fisik organisasi menjadi lapisan pertama perlindungan terhadap niat jahat yang menargetkan orang, aset, dan properti fisik. Deloitte mengungkapkan bahwa program dan teknologi Keamanan Fisik yang digunakan oleh banyak organisasi sering kali diabaikan dan semakin tidak efektif dalam mendeteksi serta merespons ancaman. Oleh karena itu, persiapan yang matang sangat penting untuk mengoptimalkan kerangka kerja Keamanan Fisik agar dapat secara efektif mengidentifikasi dan merespons ancaman siber, pelaku jahat, pelanggaran fisik, serta risiko internal dan eksternal (Deloitte, 2022).

Lebih lanjut, data yang dikutip dari Pro-Vigil menunjukkan bahwa persentase bisnis yang mengalami peningkatan insiden keamanan fisik pada tahun 2023 meningkat sebesar 25%. Hal ini menunjukkan bahwa ancaman terhadap keamanan fisik bukan hanya sebuah teori, melainkan realitas yang semakin mendesak. Peningkatan insiden ini mencerminkan kebutuhan mendesak bagi organisasi untuk memperbarui dan memperkuat program Keamanan Fisik mereka. Tanpa tindakan yang tepat, risiko terhadap keselamatan orang, keamanan aset, dan integritas properti fisik akan terus meningkat, mengancam stabilitas dan operasional bisnis secara keseluruhan (Pro-Vigil, 2023).

Di organisasi besar, tantangan keamanan fisik ini diperparah oleh kompleksitas pengelolaan kontrol akses. Banyaknya karyawan dengan beragam peran menuntut sistem yang dapat memberikan izin akses yang sesuai dengan kebutuhan masing-masing. Kebutuhan untuk sering memperbarui hak akses, terutama ketika karyawan berganti peran atau meninggalkan organisasi, menambah beban pengelolaan. Desentralisasi manajemen kontrol akses, di mana kantor atau departemen regional memiliki wewenang untuk membuat dan memodifikasi akun pengguna, seringkali menyebabkan inkonsistensi dan kesenjangan dalam pengendalian akses. Akibatnya, sulit untuk mempertahankan sistem manajemen akses yang terpadu dan aman. Selain itu, kontrol akses yang efektif memerlukan audit rutin untuk memastikan bahwa semua akun memiliki tingkat akses yang sesuai dan akun yang tidak aktif segera dinonaktifkan. Proses ini penting untuk mencegah akses yang tidak sah dan mempertahankan prinsip hak istimewa yang paling rendah. Namun, melakukan audit ini pada organisasi besar memerlukan banyak sumber daya dan tantangan logistik (NIST, 2023).

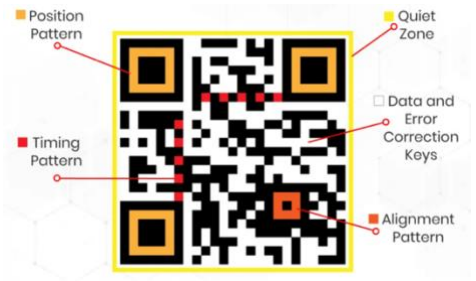
II. STUDI LITERATUR

A. Kode QR

Kode QR (*Quick Response*) adalah jenis kode matriks dua dimensi yang pertama kali dikembangkan oleh perusahaan Jepang, Denso Wave, pada tahun 1994. Kode QR memiliki kemampuan untuk menyimpan informasi dalam bentuk teks, URL, kontak, dan bahkan data biner dalam sebuah pola kotak bersegi banyak. Kode QR telah menjadi populer karena kemampuannya yang cepat dibaca, dan dapat dengan mudah dihasilkan, serta memiliki berbagai aplikasi yang luas dalam berbagai industri (Denso Wave Inc., 2021).

Gambar 2.1 Ilustrasi Kode QR (Sumber: BDC)

Komponen utama kode QR (Quick Response) terdiri dari; *Quite Zone*, *Position Pattern*, *Timing Pattern*, *Data and Error Correction Keys*, *Alignment Pattern* (Michael Chalberg, 2021).



Gambar 2.2 Komponen Kode QR (Sumber: Koronapos)

a) *Quite Zone* adalah area putih di sekeliling Kode QR yang memungkinkan pembaca untuk membedakan antara Kode QR dengan latar belakang lainnya. *Quite Zone* diperlukan agar pembaca Kode QR dapat mengenali batas kode dengan jelas dan memulai proses pembacaan dengan benar.

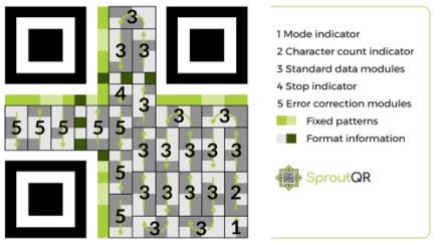
b) *Position Pattern* terdiri dari tiga pola persegi dengan sudut yang ditempatkan pada tiga sisi Kode QR. Pola ini digunakan untuk menentukan orientasi Kode QR dan memberikan petunjuk kepada pembaca tentang struktur kode.

c) *Timing Pattern* adalah serangkaian modul hitam dan putih yang membentuk pola vertikal dan horizontal di sepanjang Kode QR. Pola ini digunakan untuk mengatur waktu pembacaan dan membantu pembaca dalam menguraikan informasi yang tersimpan.

d) *Data and Error Correction Keys* adalah komponen utama dari Kode QR. Komponen ini terdiri dari pola-pola modul hitam dan putih yang menyimpan data yang sebenarnya. Selain itu, bagian ini juga memiliki mekanisme koreksi kesalahan yang memungkinkan pemulihan data jika ada kerusakan atau gangguan dalam pembacaan.

e) *Alignment Pattern* adalah pola-pola tambahan yang disisipkan ke dalam Kode QR untuk memastikan keakuratan pembacaan, terutama pada Kode QR yang lebih besar. Pola ini membantu pembaca dalam mengoreksi rotasi dan distorsi yang mungkin terjadi.

Pembacaan kode QR dimulai dari kanan bawah kode QR. Pembacaan bergerak naik dua modul data pada satu waktu sampai mencapai *position pattern* pertama. Pembacaan kemudian memindahkan dua modul data ke kiri dan bergerak turun. Pembacaan terus diulangi dengan memproses setiap modul data zig-zag kanan-kiri, naik-turun ini sampai setiap modul data terbaca. (Schulfer, 2021)



Gambar 2.3 Cara Kerja Pembacaan Kode QR (Sumber: SproutQR)

Berikut merupakan langkah lebih detail mengenai cara kerja pembacaan kode QR.

1. Pemindai akan mengenali tiga penanda posisi dalam kode QR sehingga pemindai akan menyadari di mana tepi kanan bawah kode QR berada.
2. Pemindai dimulai di kanan bawah, yakni dimulai dengan pembacaan mode *indicator*. Mode *indicator* ini terdiri dari empat modul data. Mode *indicator* menunjukkan tipe data (numerik, alfanumerik, byte, atau kanji) yang di-*encode* dalam kode QR.
3. Selanjutnya, pemindai melakukan pembacaan *character count indicator*. Mode ini terdiri dari delapan modul data. *Character count indicator* menunjukkan berapa banyak total karakter data yang dikodekan dalam kode QR.
4. Setelah mengetahui tipe data dan panjang karakter, pemindaian kemudian dilanjutkan di sepanjang modul data sampai mencapai *stop indicator*.
5. Setelah membaca *stop indicator*, pemindaian dilanjutkan sepanjang jalurnya ke *error correction modules* untuk menentukan *levels error code correction* dari kode QR.

Terdapat dua metode pasca-proses pembacaan kode QR, yaitu pembacaan kode QR secara *online* dan *offline*. Pembacaan kode QR *online* melibatkan verifikasi data ke *server* utama yang memerlukan koneksi internet, sementara pembacaan kode QR offline tidak memerlukan koneksi internet dan dapat berfungsi secara mandiri. (Nove, 2023)

Pasca-proses pembacaan kode QR secara *online* melibatkan aplikasi atau perangkat membaca kode QR terhubung ke internet untuk memverifikasi data yang terkandung dalam kode QR. Metode ini memiliki beberapa keunggulan, terutama dalam akses ke area terbatas dan penggunaan yang memerlukan otorisasi tinggi. Dengan pembacaan kode QR *online*, pemindai kode QR harus terhubung ke *server* utama untuk mendapatkan akses atau informasi yang diperlukan.

Terdapat beberapa komponen utama dalam metode pembacaan kode QR *online*, yakni *QR Code Scanner*, *main server*, dan juga koneksi internet.

1. *QR Code Scanner*: Aplikasi atau perangkat keras yang dapat memindai dan membaca kode QR. Aplikasi ini harus terhubung ke koneksi internet.
2. *Main server*: *Server* yang memproses permintaan dari aplikasi pembaca kode QR online. *Server* ini berfungsi untuk memverifikasi kode QR, memberikan otorisasi, dan memberikan data atau layanan yang diperlukan.
3. Koneksi Internet: Koneksi internet dibutuhkan untuk mengirimkan data dari pembaca kode QR ke *server* utama dan menerima respon dari *server*.

Pembacaan kode QR *online* umumnya digunakan dalam berbagai aplikasi, seperti akses ke area terbatas, tiket elektronik, dan pembayaran secara digital.

Pasca-proses pembacaan kode QR secara *offline* melibatkan aplikasi atau perangkat membaca kode QR yang tidak memerlukan koneksi internet untuk memverifikasi data yang terkandung dalam kode QR. Metode ini umumnya digunakan untuk tugas-tugas yang memerlukan kemampuan pembacaan mandiri tanpa ketergantungan pada jaringan internet.

Terdapat beberapa komponen utama dalam metode pembacaan kode QR *offline*, yakni *QR Code Scanner* dan data lokal.

1. *QR Code Scanner*: Aplikasi atau perangkat keras yang dapat memindai dan membaca kode QR tanpa koneksi internet.
2. Data Lokal: Data diperlukan untuk memverifikasi kode QR disimpan secara lokal di perangkat. Hal ini memungkinkan perangkat untuk melakukan verifikasi tanpa koneksi internet.

Metode pembacaan kode QR *offline* sering digunakan dalam aplikasi yang memerlukan kemudahan dan kecepatan dalam membaca kode QR tanpa ketergantungan pada koneksi internet, seperti memindai kode QR untuk mengakses informasi produk.

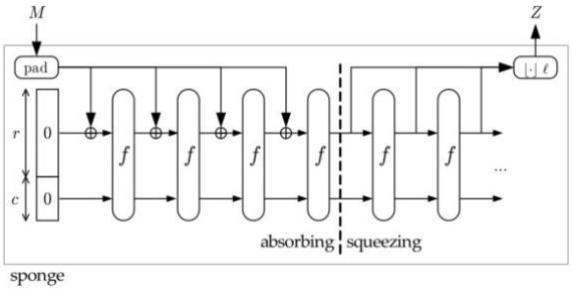
Pembangkitan kode QR atau QR *code encoding*, adalah proses konversi data menjadi gambar dua dimensi yang dapat dengan mudah dibaca oleh perangkat pemindai atau aplikasi QR. Kode QR menggunakan sejumlah besar kotak kecil dan hitam putih yang membentuk pola tertentu untuk mewakili informasi yang terkandung di dalamnya. Salah satu aspek penting dalam pembangkitan kode QR adalah penentuan struktur dan ukuran kode QR yang akan dihasilkan. Ini melibatkan keputusan mengenai kapasitas penyimpanan kode QR, tingkat kekeruhan, dan mode penyandian data. Pemilihan mode penyandian mempengaruhi cara data seperti teks, URL, atau data biner direpresentasikan dalam kode QR. Selain itu, pembangkitan kode QR juga mencakup penambahan elemen koreksi kesalahan untuk meningkatkan ketahanan terhadap gangguan atau kerusakan pada kode QR, memastikan keberhasilan pembacaan informasi meskipun terjadi gangguan kecil pada kode.

Pada tingkat implementasi, pembangkitan kode QR melibatkan pilihan algoritma pengkodean yang sesuai dengan jenis data yang akan disandikan. Beberapa algoritma umum melibatkan kompresi data, pemilihan mode penyandian yang optimal, dan pengaturan parameter seperti ukuran kode QR dan tingkat kekeruhan. Proses ini melibatkan transformasi data ke dalam bentuk bit yang diatur sesuai dengan struktur kode QR, dengan setiap bagian data yang diatur sedemikian rupa untuk memastikan keakuratan dan kestabilan pembacaan. Oleh karena itu, pembangkitan kode QR merupakan langkah kritis dalam memastikan bahwa informasi dapat disandikan dengan benar dan dapat diakses dengan cepat dan andal melalui pemindai atau perangkat lunak yang sesuai.

B. Fungsi Hash SHA-3 (Keccak)

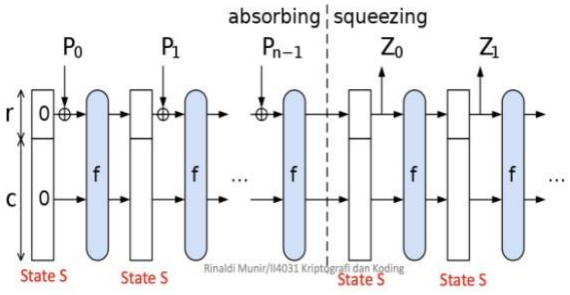
Fungsi *hash* SHA-3 (Secure Hash Algorithm 3), juga dikenal sebagai Keccak, adalah fungsi *hash* kriptografik yang digunakan untuk menghasilkan nilai *hash* yang unik dari suatu data atau pesan. Fungsi *hash* SHA-3 didasarkan pada permutasi Keccak, yang merupakan hasil dari kompetisi fungsi *hash* yang diadakan oleh Institut Teknologi NIST (National Institute of Standards and Technology) pada tahun 2007.

Keccak memiliki perbedaan dalam metode konstruksi dibandingkan dengan finalis SHA-3 lainnya, yaitu menggunakan konstruksi "spons" (*sponge construction*). Sementara desain SHA-3 lainnya bergantung pada "fungsi kompresi", Keccak menggunakan fungsi yang tidak memampatkan (non-kompresi) untuk proses "penyerapan" (*absorbing*) dan kemudian "pemerasan" (*squeezing*) nilai *hash*. Berikut merupakan langkah fungsi *hash* Keccak dalam menghasilkan *message digest*.



Gambar 2.4 Konstruksi Spons pada Fungsi *Hash* Keccak (Sumber: Slide Kuliah II4031 Kriptografi dan Koding - Rinaldi Munir)

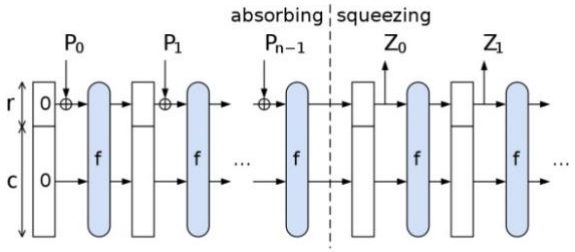
Pada tahap ini, pesan *M* diubah menjadi sebuah *string* P dengan menambahkan bit-bit pengganjal (*padding*), sehingga panjang *string* P dapat dibagi habis oleh nilai r atau n = panjang(P)/r. Setelah itu, *string* P dibagi menjadi blok-blok P*i* dengan panjang r-bit. Setelah itu, sejumlah b-bit dari peubah status (*state*) *S* diinsialisasi dengan nol.



Gambar 2.5 Fase Penyerapan pada Fungsi *Hash* Keccak (Sumber: Slide Kuliah II4031 Kriptografi dan Koding - Rinaldi Munir)

Pada fase ini, untuk setiap blok masukan P*i* dengan ukuran r-bit, dilakukan operasi XOR dengan r-bit pertama

dari *state S*. Hasilnya kemudian dimasukkan ke dalam fungsi permutasi *f*, yang menghasilkan *state S* baru. Setelah semua blok masukan diproses, konstruksi spons beralih ke fase berikutnya yakni fase pemerasan (*squeezing*).



Gambar 2.6 Fase Pemerasan pada Fungsi *Hash* Keccak (Sumber: Slide Kuliah II4031 Kriptografi dan Koding - Rinaldi Munir)

Pada fase ini, nilai *hash* pesan akan disimpan dalam variabel *Z*. Langkah awal adalah menginisialisasi *Z* dengan sebuah *string* kosong. Selanjutnya, selama panjang *Z* masih belum mencapai nilai yang ditentukan, r-bit pertama dari *state S* akan digabungkan (*append*) ke *Z*. Jika panjang *Z* masih belum mencapai nilai yang ditentukan, blok masukan akan dimasukkan ke dalam fungsi permutasi *f* untuk menghasilkan *state S* baru.

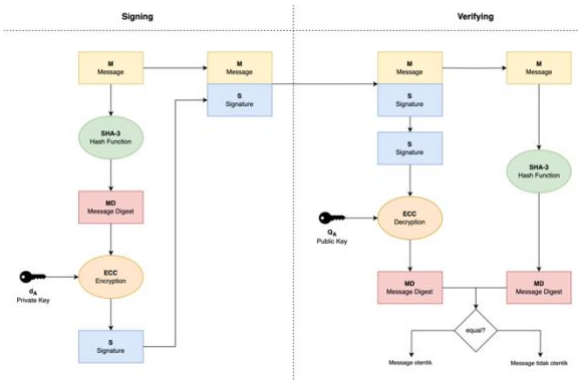
C. Tanda Tangan Digital

Tanda tangan digital pada makalah ini dilakukan dengan menggunakan kombinasi kriptografi kunci publik dan fungsi *hash* merupakan salah satu metode yang umum digunakan untuk memastikan keaslian dan integritas pesan digital. Kriptografi kunci publik digunakan untuk menghasilkan pasangan kunci, yaitu kunci publik dan kunci privat. Kunci publik dapat dibagikan kepada semua pihak yang ingin memverifikasi tanda tangan, sementara kunci privat harus dijaga dengan baik oleh pemiliknya. Kunci publik digunakan untuk memverifikasi keaslian tanda tangan, sedangkan kunci privat digunakan untuk menghasilkan tanda tangan yang unik.

Fungsi *hash* digunakan untuk menghasilkan nilai *hash* dari pesan yang akan ditandatangani. Fungsi *hash* mengubah pesan menjadi serangkaian angka unik dengan panjang tetap. Nilai *hash* ini merepresentasikan pesan asli dan akan digunakan sebagai "ringkasan" pesan yang akan ditandatangani.

Proses penandatanganan dimulai dengan menghasilkan nilai hash dari pesan menggunakan fungsi *hash* yang dipilih. Selanjutnya, nilai *hash* tersebut dienkripsi menggunakan kunci privat yang dimiliki oleh penandatangan. Hasil enkripsi ini merupakan tanda tangan digital yang akan dikirimkan dengan pesan tersebut.

Untuk memverifikasi tanda tangan, penerima pesan akan memisahkan pesan dan tanda tangan digital. Pesan akan di *hash* kembali dengan jenis fungsi *hash* yang sama, sedangkan tanda tangan digital akan didekripsi dengan menggunakan kunci publik pengirim. Penerima kemudian akan membandingkan *output* dari kedua proses tersebut. Jika sama, maka pesan tersebut dinyatakan otentik. Berikut merupakan diagram proses pembubuhan tanda dan verifikasi tangan digital.



Gambar 2.7 Gambaran Proses Tanda Tangan Digital

D. Algoritma Kriptografi Kurva Eliptik

Algoritma *Elliptic Curve Cryptography* (ECC) merupakan jenis algoritma kriptografi kunci publik yang juga sering digunakan. Algoritma ini didasarkan pada operasi matematika yang melibatkan titik-titik pada kurva eliptik. Keamanan ECC didasarkan *Elliptic Curve Discrete Logarithm Problem*.

a) Langkah Pembangkitan Kunci

1. Pilih sebuah kurva eliptik yang sesuai, misalnya kurva eliptik dengan persamaan

$$y^2 = x^3 + ax + b \ (mod\ p)$$

Dimana p adalah bilangan prima yang lebih besar dari nilai maksimum x dan y

2. Pilih titik basis (*base point*), $B(x_B,y_B)$ yang berada pada kurva eliptik yang dipilih
3. Pilih bilangan bulat d_A yang berada pada selang $[1, p-1]$
4. Hitung hasil kali antara bilangan bulat k dengan titik basis B

$$Q_A = d_A \times B$$

Hasil dari tahap ini adalah **kunci *private* d_A** dan **kunci publik Q_A**.

b) Langkah Enkripsi / *Signing*

1. Hitung *hash* pesan M menggunakan fungsi *hash*

$$h = hash(M)$$

2. Pilih nilai k yang berada pada selang $[1, n-1]$. n adalah urutan bilangan bulat dari B, yang berarti $n \times G = O$, dengan O adalah elemen identitas
3. Hitung titik kurva

$$k \times B = (x_1, y_1)$$

4. Hitung nilai r yang merupakan bagian awal dari tanda tangan (*first half signature*)

$$r = x_1 \ mod\ n$$

Jika $r = 0$, kembali ke langkah 2

5. Hitung nilai s yang merupakan bagian akhir dari tanda tangan (*the other half signature*)

$$s = k^{-1}(h + r \times d_A)\ mod\ n$$

Jika s = 0, kembali ke langkah 2

Hasil dari tahap ini adalah **pasangan nilai awal dan nilai akhir *signature* (r, s).**

c) Langkah Dekripsi / *Verification*

1. Hitung *hash* pesan M menggunakan fungsi *hash* yang sama pada saat proses *signing*.

$$h = hash(M)$$

2. Hitung nilai $s_{inverse}$

$$s_{inverse} = s^{-1}\ mod\ n$$

3. Hitung nilai u_1 dan u_2

$$u_1 = h \times s^{-1}\ mod\ n$$

$$u_2 = r \times s^{-1}\ mod\ n$$

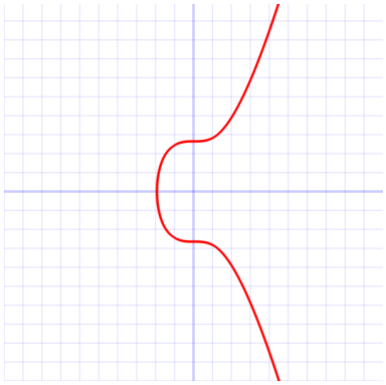
4. Hitung nilai titik verifikasi S

$$S = u_1 \times B + u_2 \times Q_A$$

Tanda tangan dikatakan **valid jika $r \equiv x_S$.**

d) Kurva Eliptik Secp256k1

Kurva eliptik Secp256k1 diperkenalkan oleh Certicom pada tahun 1999. Kurva ini dibangun di atas badan medan hingga (*prime field*) dan mengikuti standar dari Standards for Efficient Cryptography (SEC), yaitu "SEC 2: Recommended Elliptic Curve Domain Parameters". Kurva eliptik Secp256k1 menawarkan efisiensi komputasi yang baik dan tingkat keamanan yang sangat tinggi berkat pemilihan parameter yang tepat. (Certicom Research, 2000)



Gambar 2.8 Bentuk Kurva Eliptik Secp256k1 (Sumber: https://en.bitcoin.it/wiki/Secp256k1)

Kurva eliptik Secp256k1 memiliki persamaan matematis:

$$y^2 = x^3 + 7\ (mod\ p)$$

Dengan p adalah bilangan prima yang menggambarkan badan medan yang digunakan. Kurva ini memiliki orde n, yang menunjukkan jumlah titik-titik yang ada di atasnya. Berikut merupakan parameter-parameter kurva eliptik Secp256k1.

Tabel 2.1 Parameter Kurva Eliptik Secp256k1

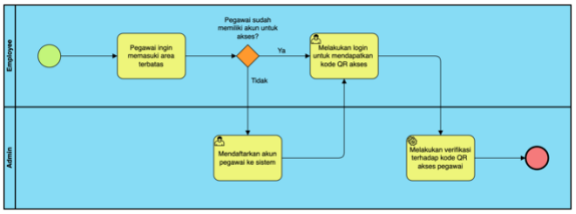
Parameter	Nilai
p	0xfffffffffffffffffffffffffffffffffffffffffffffffffffffffefffffc2f
a	0x0000000000000000000000000000000000000000000000000000000000000000
b	0x0000000000000000000000000000000000000000000000000000000000000007
G	(0x79be667ef9dcbbac55a06295ce870b07029bfcdb2dce28d959f2815b16f81798,0x483ada7726a3c4655da4fbfc0e1108a8fd17b448a68554199c47d08ffb10d4b8)
n	0xfffffffffffffffffffffffebaaedce6af48a03bbfd25e8cd0364141
h	0x1

Kurva eliptik Secp256k1 memiliki peran penting dalam dunia kriptografi kunci publik dan telah menjadi komponen kunci dalam berbagai aplikasi keamanan digital. Salah satu penggunaan utama kurva ini terletak dalam *cryptocurrency*, terutama dalam jaringan Bitcoin, di mana kurva Secp256k1 digunakan untuk pembuatan dan verifikasi tanda tangan digital. Dalam protokol Bitcoin, kunci privat adalah elemen kunci dalam mengamankan transaksi dan mengotorisasi penggunaan koin digital. Kurva ini juga digunakan dalam berbagai aplikasi lain, termasuk pertukaran kunci rahasia dalam protokol Diffie-Hellman, sehingga memastikan pertukaran informasi aman di seluruh jaringan komunikasi. Selain itu, kurva Secp256k1 digunakan dalam berbagai aplikasi keamanan lainnya, termasuk perlindungan data dalam komunikasi aman. Penggunaan luas kurva ini dalam berbagai kasus pengamanan digital menjadi bukti ketangguhan dan keandalannya dalam menghadapi berbagai tantangan kriptografi. (Antonopoulos & M., 2014)

III. RANCANGAN SISTEM

A. *Proses Bisnis*

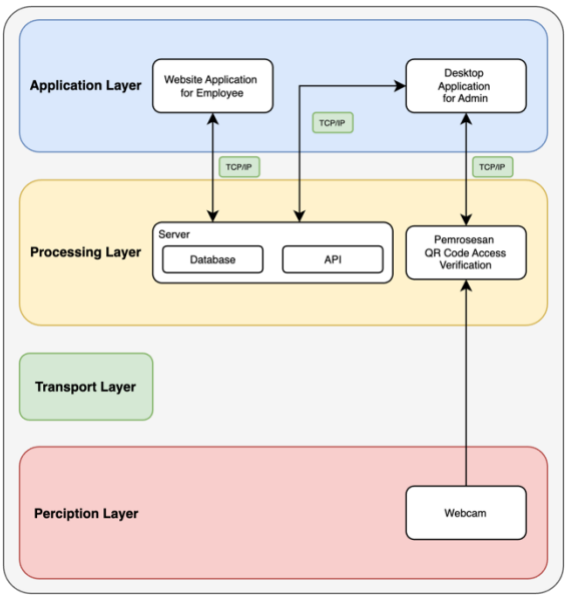
Penggambaran proses bisnis direpresentasikan dengan diagram BPMN (Business Process Modeling Notation). Diagram ini biasa digunakan untuk menggambarkan urutan proses dan pesan yang mengalir antara pengguna dalam kegiatan yang berbeda. Berikut merupakan gambaran proses bisnis sistem.



Gambar 3.1 Proses Bisnis

B. *Arsitektur Sistem*

Arsitektur sistem adalah suatu kerangka kerja yang mendefinisikan bagaimana komponen-komponen perangkat lunak diatur dan berinteraksi satu sama lain. Hal ini mencakup beberapa lapisan, seperti lapisan aplikasi, lapisan pemrosesan, lapisan transportasi, dan lapisan persepsi. Masing-masing lapisan memiliki fungsi spesifik. Berikut merupakan diagram arsitektur dari sistem yang akan dikembangkan.



Gambar 3.2 Diagram Arsitektur Sistem

C. *Kebutuhan Fungsional dan Non Fungsional*

Dalam menentukan fitur-fitur sistem yang akan dibangun, perlu memetakan kebutuhan fungsional dan non fungsional dari sistem itu sendiri. Kebutuhan fungsional sendiri adalah kebutuhan yang berisi proses-proses apa saja yang nantinya dapat dilakukan oleh sistem, sedangkan kebutuhan non fungsional adalah kebutuhan yang menitikberatkan pada properti perilaku yang dimiliki oleh sistem.

Tabel 3.1 Kebutuhan Fungsional

ID	Kebutuhan Fungsional
FC-01	Sistem dapat memungkinkan pengguna untuk masuk ke sistem dengan memasukkan informasi *login*, seperti *username* dan *password*.
FC-02	Sistem dapat memungkinkan pengguna untuk keluar dari sistem.
FC-03	Sistem dapat memungkinkan admin untuk mendaftarkan pegawai baru.
FC-04	Sistem dapat memungkinkan admin untuk melihat semua pegawai yang memiliki akses ke sistem.
FC-05	Sistem dapat memungkinkan admin untuk menghapus akses bagi akun pegawai non-aktif.
FC-06	Sistem dapat memverifikasi dan mengotentikasi keabsahan kode QR akses yang ditampilkan oleh aplikasi pegawai baik dalam keadaan *online* maupun *offline*.
FC-07	Sistem dapat memungkinkan admin untuk melihat daftar aktivitas akses masuk pegawai.
FC-08	Sistem dapat memungkinkan pegawai untuk mendapatkan kode QR akses yang valid.
FC-09	Sistem dapat membangkitkan kode QR akses pegawai secara dinamis.

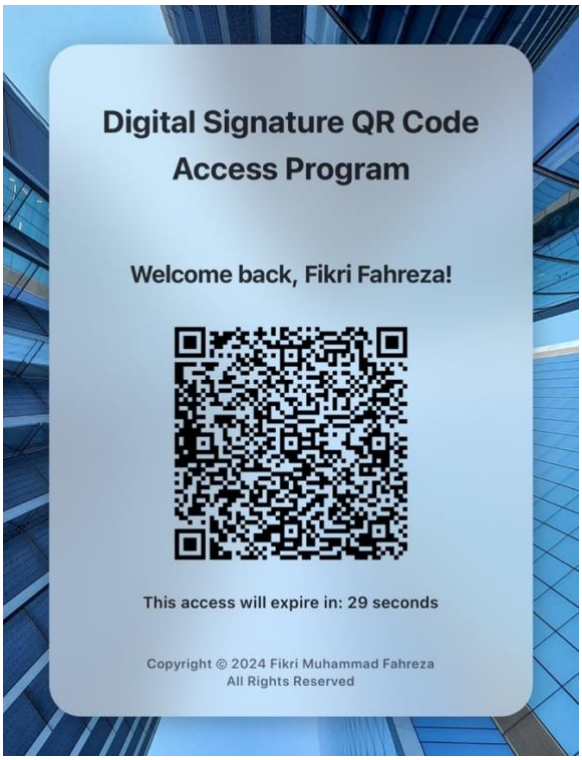
Tabel 3.2 Kebutuhan Non Fungsional

ID	Aspek	Kebutuhan Non Fungsional
NF-01	*Portability*	Sistem dapat diakses melalui perangkat pengguna.
NF-02	*Usability*	Sistem harus dapat dengan mudah digunakan dan dipahami pengguna.
NF-03	*Availability*	Sistem harus tetap tersedia sepanjang waktu operasional.

IV. IMPLEMENTASI SISTEM

A. *Implementasi Kode QR dengan Tanda Tangan Digital*

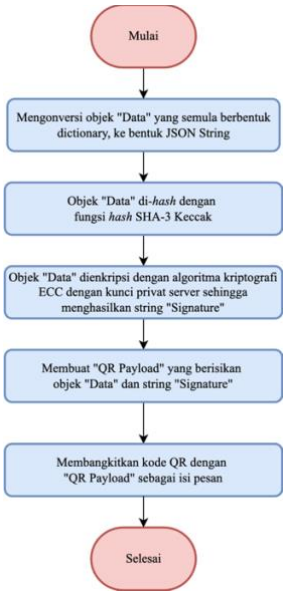
Berikut merupakan contoh hasil implementasi kode QR akses seorang pegawai.



Gambar 4.1 Contoh Hasil Implementasi Kode QR Akses

Gambar 4.1 menunjukkan contoh hasil implementasi pembangkitan kode QR akses yang akan diterima oleh pegawai sebagai metode autentikasi. Kode QR akses ini berlaku selama tiga puluh detik, setelah itu akan kadaluarsa dan server secara otomatis akan membangkitkan kode QR akses yang baru. Untuk membangkitkan kode QR dari sebuah pesan, penulis menggunakan *library* Python yang bernama qrcode. *Library* ini khusus dibuat untuk membangkitkan kode QR.

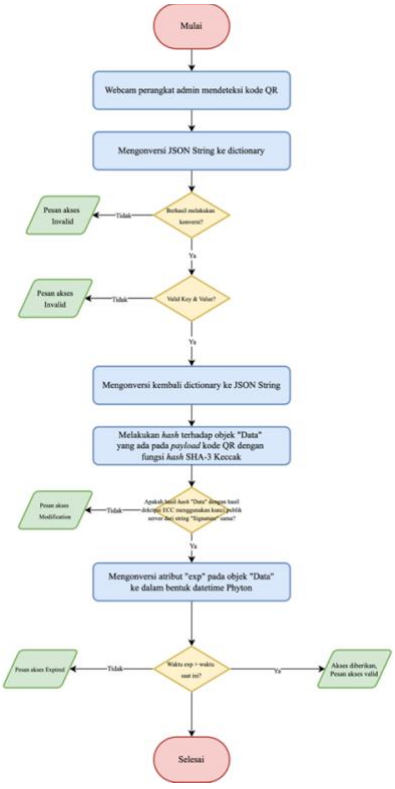
Kode QR yang dihasilkan menggunakan **level koreksi L**, yang memiliki kemampuan memulihkan data sekitar 7% jika kode QR rusak. Versi yang digunakan pada kode QR akses ini adalah **versi 10**, berukuran 57x57 piksel, dengan kapasitas menyimpan hingga 395 karakter alfanumerik. Berikut ini merupakan algoritma pembangkitan kode QR akses bagi pegawai yang digambarkan dengan diagram alir atau *flowchart*.



Gambar 4.2 Algoritma Pembangkitan Kode QR Akses

Kode QR akses yang berisi data-data pegawai dan tanda tangan digital akan diverifikasi oleh admin menggunakan modul pemrosesan kode QR yang terdapat pada aplikasi admin. Modul ini mengandung skrip program yang dapat melakukan verifikasi kode QR secara *offline* karena berbasis pada verifikasi tanda tangan digital yang dilakukan melalui *edge computing*. Namun, perlu dicatat bahwa **dalam mode *offline*, log akses tidak akan tercatat pada log aktivitas** karena penulisan log akses ke log aktivitas memerlukan koneksi internet untuk dapat menulis log akses ke basis data.

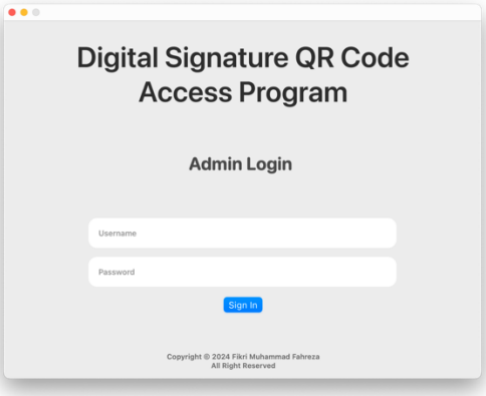
Berikut ini adalah algoritma verifikasi kode QR akses pegawai yang digambarkan dengan diagram alir atau *flowchart.*



Gambar 4.3 Algoritma Verifikasi Kode QR Akses

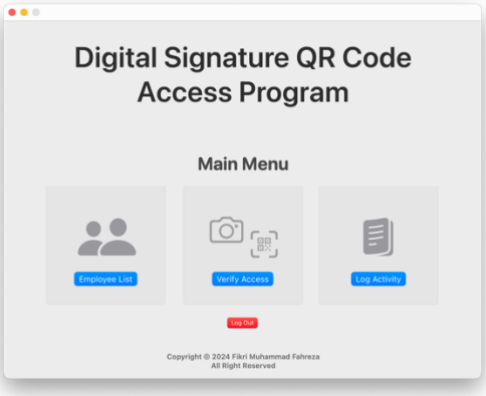
B. Implementasi Antarmuka Aplikasi Admin

Implementasi antarmuka aplikasi bagi pengguna dengan *role* admin dibuat menggunakan *library* PyQt6. *Library* ini dipilih karena beberapa alasan utama yang menjadikannya lebih unggul dan mudah digunakan dibandingkan *framework* GUI berbahasa Python lainnya. Berikut merupakan tangkapan layar dari antarmuka aplikasi Admin.



Gambar 4.4 Implementasi Antarmuka Halaman *Login* pada Aplikasi Admin

Gambar 4.4 merupakan halaman *login* yang merupakan tampilan awal dari aplikasi admin. Pada halaman ini terdapat dua buah *field* masukkan teks, yakni masukkan teks untuk *username* dan masukkan teks untuk *password.* Di bawah *field* masukkan teks terdapat *push button* yang berfungsi sebagai eksekutor untuk admin dapat melakukan *login*.



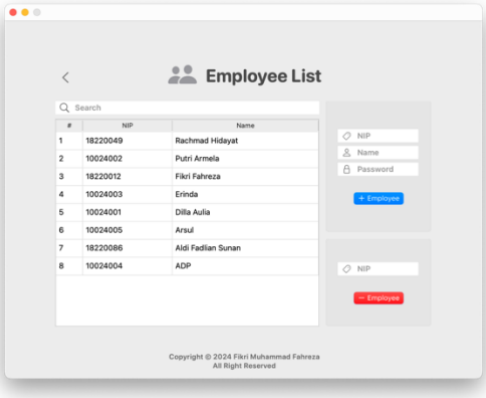
Gambar 4.5 Implementasi Antarmuka Halaman Menu pada Aplikasi Admin

Gambar 4.5 merupakan tampilan antarmuka halaman menu pada aplikasi admin. Pada halaman ini terdapat empat buah *push button* dengan fungsi yang berbeda-beda. Berikut merupakan penjelasan lebih lanjut untuk setiap *push button* yang terdapat pada halaman ini.

1. Employee List, *push button* yang berfungsi sebagai eksekutor untuk navigasi ke halaman Employee List sebagai manajemen akun pegawai.
2. Verify Access, *push button* yang berfungsi sebagai eksekutor untuk

navigasi ke halaman Verify Access sebagai halaman untuk melakukan verifikasi terhadap kode QR akses yang dimiliki pegawai.

3. Log Activity, *push button* yang berfungsi sebagai eksekutor untuk navigasi ke halaman Log Activity sebagai antarmuka dari daftar *logs* yang tersimpan di basis data.
4. Sign Out, *push button* yang berfungsi sebagai eksekutor untuk navigasi ke halaman Log In. Berfungsi untuk menghapus sesi pengguna admin.

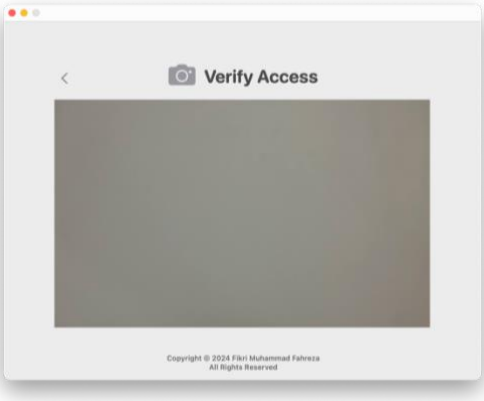


Gambar 4.6 Implementasi Antarmuka Halaman Employee List pada Aplikasi Admin

Gambar 4.6 merupakan tampilan antarmuka halaman Employee List pada aplikasi admin. Antarmuka ini adalah perbaikan dari rancangan antarmuka Show All Employee dan Register New Employee yang dijelaskan pada Bab III. Singkatnya, halaman Employee List merupakan halaman yang berisi kedua fungsionalitas tersebut. Hal ini membuat aplikasi lebih simpel dan mudah digunakan.

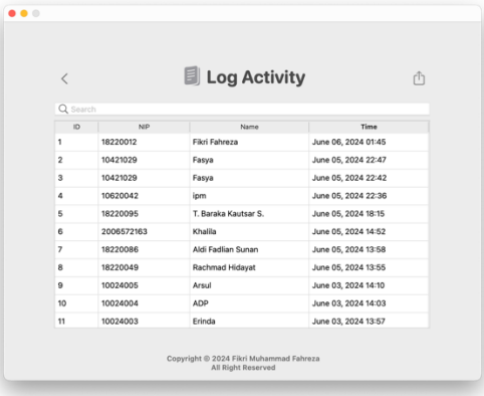
Halaman ini berisi daftar akun pegawai terdaftar yang mempunyai akses ke area terbatas. Halaman ini terdiri atas sebuah *table widget* untuk menampilkan daftar pegawai terdaftar sebagai antarmuka dari entitas *users*, sebuah *field* masukkan teks untuk pencarian *record* pada *table widget*, dan dua buah *group box*. Berikut penjelasan lebih lanjut untuk setiap *group box*.

1. Add Employee, berfungsi untuk menambahkan akun pegawai ke sistem. Terdiri dari tiga *field* masukkan teks, yakni masukkan teks untuk NIP pegawai, masukkan teks untuk nama pegawai dan masukkan teks untuk kata sandi pegawai. Di bagian bawah *field* masukkan teks terdapat satu buah *push button* yang yang berfungsi sebagai eksekutor untuk menambahkan akun pegawai ke sistem.
2. Delete Employee, berfungsi untuk menghapus akun pegawai tertentu dari sistem. Terdiri dari sebuah *field* masukkan teks, yakni masukkan teks untuk NIP pegawai. Di bagian bawah *field* masukkan teks terdapat satu buah *push button* yang yang berfungsi sebagai eksekutor untuk menghapus akun pegawai dari sistem.



Gambar 4.7 Implementasi Tampilan Antarmuka Halaman *Verify Access* pada Aplikasi Admin

Gambar 4.7 merupakan tampilan antarmuka halaman Verify Access pada aplikasi admin. Halaman ini terdiri atas sebuah *camWidget*, yakni *widget* yang menampilkan tangkapan kamera dari perangkat yang digunakan admin. Halaman ini berfungsi sebagai validator dari kode QR akses yang dimiliki seorang pegawai.



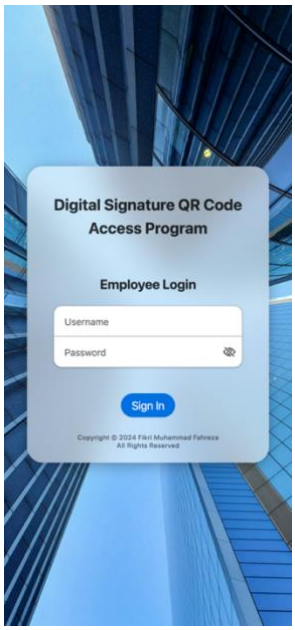
Gambar 4.8 Implementasi Tampilan Antarmuka Halaman Log Aktivitas pada Aplikasi Admin

Gambar 4.8 merupakan tampilan antarmuka halaman Log Activity pada aplikasi admin. Pada halaman ini terdapat sebuah *table widget* yang berfungsi untuk menampilkan daftar log aktivitas akses pegawai sebagai antarmuka dari entitas *logs*. Terdapat pula sebuah *field* masukkan teks untuk pencarian *record* pada *table widget*, dan sebuah *push button* "Export" untuk melakukan ekspor data log aktivitas apabila diperlukan untuk kebutuhan tertentu ataupun pemrosesan lebih lanjut.

C. Implementasi Antarmuka Aplikasi Pegawai

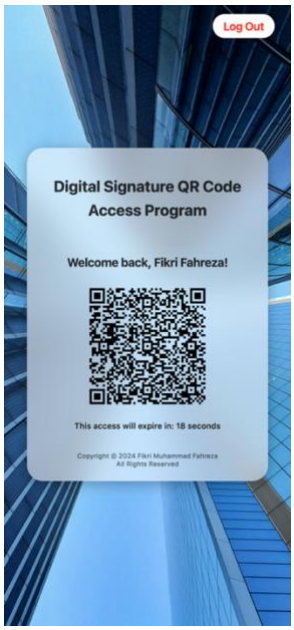
Implementasi antarmuka aplikasi bagi pengguna dengan *role* pegawai dibuat dengan bahasa HTML, *framework* Bootstrap, dan CSS. Kombinasi ketiga teknologi ini dipilih karena menawarkan fleksibilitas, kemudahan penggunaan, dan kemampuan untuk menghasilkan tampilan yang responsif dan menarik secara visual. HTML digunakan sebagai fondasi untuk menyusun struktur dasar halaman

website, sementara *framework* Bootstrap digunakan untuk mendapatkan komponen-komponen UI yang siap pakai, seperti tombol, formulir, navigasi, dan tata letak *grid*.



Gambar 4.9 Implementasi Tampilan Antarmuka Halaman *Login* pada Aplikasi Pegawai

Gambar 4.9 merupakan tampilan antarmuka halaman *login* yang merupakan tampilan awal dari aplikasi pegawai. Gambar latar belakang digunakan untuk memberi kesan personal pada perusahaan. Pada halaman ini terdapat dua buah *field* masukkan teks, yakni masukkan teks untuk *username* dan masukkan teks untuk *password.* Selain itu, terdapat sebuah *push button* dalam bentuk ikon berbentuk mata untuk me-*reveal* hasil input pengguna pada kolom *password*. Di bawah *field* masukkan teks terdapat *push button* yang berfungsi sebagai eksekutor untuk admin dapat melakukan *login*. Hanya akun pegawai terdaftar yang dapat masuk ke sistem untuk mendapatkan kode QR akses.



Gambar 4.10 Implementasi Tampilan Antarmuka Halaman *Landing* pada Aplikasi Pegawai

Gambar 4.10 merupakan tampilan antarmuka halaman utama atau halaman *landing* pada aplikasi pegawai. Pada halaman ini terdapat tampilan kode QR yang dihasilkan sistem sebagai alat akses untuk memasukki area terbatas (kantor). Pada satu waktu, kode QR akses yang ditampilkan mempunyai waktu 30 detik hingga akhirnya kadaluarsa. Pada bagian bawah kode QR terdapat hitung mundur (*countdown*) dari masa berlaku kode QR sebagai akses yang valid. Setelah lebih dari 30 detik, halaman akan otomatis tersegarkan dan membangkitkan kode QR akses yang baru. Pada bagian kanan atas halaman ini terdapat *push button* "Log Out" yang berfungsi sebagai eksekutor untuk pegawai melakukan *log out.*

V. PENGUJIAN

A. Functional Testing

Pengujian fungsional bertujuan untuk memastikan bahwa perangkat lunak berfungsi sesuai dengan spesifikasi yang telah ditetapkan. Melalui pengujian ini, setiap fitur dan alur kerja aplikasi diuji untuk memastikan mereka bekerja dengan benar dan memberikan hasil yang diharapkan. Pengujian fungsional membantu mendeteksi bug atau kesalahan dalam suatu fitur spesifik, sehingga perangkat lunak dapat beroperasi dengan baik dalam berbagai kondisi yang mungkin dihadapi oleh pengguna. Oleh karena itu, pengujian fungsional berkontribusi signifikan terhadap kualitas akhir produk dan kepuasan pengguna.

Pengujian fungsional diawali dengan membuat skenario testing. Skenario *testing* yang dihasilkan pada fase pengujian ini adalah sebanyak 47 skenario *testing*, yang terdiri atas 33 testing skenario untuk *role* pengguna admin, dan 14 *testing* skenario untuk *role* pengguna pegawai. Pengujian fungsional dilakukan dengan metode *black box testing*. Metode ini adalah metode pengujian tanpa melihat kode sumber program. Berikut merupakan rincian hasil status skenario uji yang diujikan pada pengujian fungsional.

Tabel 5.1 Rincian Hasil Status Skenario Uji pada Pengujian Fungsional

Status	Keterangan	Jumlah
PASS	Skenario uji yang berhasil dijalankan tanpa adanya *error* atau *bug* tertentu.	47
FAIL	Skenario uji gagal dijalankan karena adanya *error* atau *bug* tertentu.	0
UNEXECUTED	Skenario uji tidak dijalankan karena adanya ketergantungan, masalah lingkungan, atau alasan lainnya.	0
TOTAL *EXECUTED*		47

Berdasarkan keseluruhan hasil pengujian fungsional, dapat disimpulkan bahwa aplikasi telah berfungsi **normal**, **fungsional**, dan **sesuai** dengan ekspektasi skenario testing dan spesifikasi yang telah didefinisikan.

B. User Acceptance Testing

Tujuan dari pengujian Pengujian Penerimaan Pengguna (*User Acceptance Testing*/UAT) adalah untuk mengidentifikasi masalah atau kekurangan yang mungkin tidak terdeteksi pada tahap pengujian sebelumnya, dengan perspektif dari pengguna akhir. Selain itu, UAT juga membantu memastikan bahwa perangkat lunak memberikan pengalaman pengguna yang optimal dan memenuhi tujuan bisnis yang telah ditentukan. Dengan demikian, UAT berperan penting dalam memastikan kepuasan pengguna dan kelayakan perangkat lunak untuk diterapkan di lingkungan produksi.

Pada pengujian penerimaan pengguna, *tester* atau penguji UAT yang terlibat adalah seseorang dengan kriteria sebagai berikut.

a. Seseorang dengan status bekerja dan pernah menggunakan satu atau lebih jenis metode akses ke area terbatas, atau
b. Seseorang dengan status mahasiswa yang mempunyai pemahaman terkait desain interaksi sebuah aplikasi/sistem/perangkat lunak.

Pengujian diawali dengan mendistribusikan skenario UAT kepada para penguji bentuk *spreadsheet*. Skenario ini berisi berbagai kasus uji yang mencakup fitur dan fungsi aplikasi yang perlu diuji. Setiap skenario akan mencakup langkah-langkah yang harus diikuti oleh penguji, deskripsi, *pre-requiste*, ekspektasi, dan hasil pengujian. *Spreadsheet* ini disusun sedemikian rupa untuk memastikan pengujian dilakukan secara sistematis dan menyeluruh.

Setelah menerima skenario UAT, para penguji mulai menguji aplikasi sesuai dengan skenario yang diberikan. Mereka mengikuti langkah-langkah yang telah ditentukan dalam *spreadsheets*, memasukkan data yang diperlukan, dan mencatat hasil pengujian mereka. Proses ini bertujuan untuk memastikan bahwa setiap fungsi dan fitur aplikasi beroperasi sesuai dengan spesifikasi yang telah ditetapkan. Penguji harus mencermati setiap detail untuk mendeteksi bug atau masalah yang mungkin tidak terlihat selama tahap pengembangan.

Selama pengujian, penguji rutin melakukan *update* pada kolom hasil pada *spreadsheets* untuk mencatat hasil dari setiap kasus uji. Mereka menandai apakah suatu kasus uji lulus atau gagal, dan memberikan catatan tambahan jika diperlukan. Jika ditemukan bug atau masalah, penguji mencatat detail masalah tersebut dan menghubungkannya dengan kasus uji yang relevan. *Spreadsheets* yang diupdate secara berkala ini akan menjadi dasar untuk evaluasi dan analisis hasil pengujian.

Langkah terakhir dalam proses UAT adalah mengisi Google Forms yang telah disediakan. Formulir ini berfungsi sebagai sarana penguji sebagai calon pengguna akhir untuk melakukan penilaian terhadap aplikasi secara keseluruhan dan mengumpulkan umpan balik secara lebih terstruktur.

Berdasarkan proses UAT yang telah dilakukan, didapatkan 10 orang penguji UAT, dengan rincian sebagai berikut.

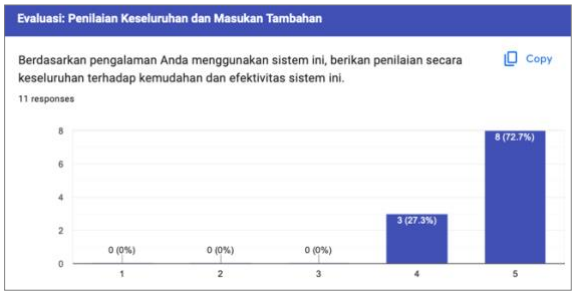
a. 5 orang dengan status bekerja, dan
b. 6 orang dengan status mahasiswa.

Berikut merupakan rincian hasil status skenario uji yang diujikan pada pengujian UAT.

Tabel 5.2 Rincian Hasil Status Skenario Uji pada Pengujian UAT

Status	Keterangan	Jumlah
PASS	Skenario uji yang berhasil dijalankan tanpa adanya *error* atau *bug* tertentu.	18
FAIL	Skenario uji gagal dijalankan karena adanya *error* atau *bug* tertentu.	0
UNEXECUTED	Skenario uji tidak dijalankan karena adanya ketergantungan, masalah lingkungan, atau alasan lainnya.	0
TOTAL *EXECUTED*		18

Berikut adalah hasil penilaian dari *tester* yang berperan sebagai pengguna akhir dengan *role* pegawai yang diberikan melalui platform Google Forms.



Gambar 5.1 Hasil Penilaian *Tester* pada Pengujian UAT

Penilaian secara keseluruhan dilakukan dengan skala linear, mulai dari nilai 1 sampai dengan 5. Nilai 1 berarti sistem dinilai sangat buruk, sedangkan nilai 5 berarti sistem dinilai sangat baik. Berdasarkan penilaian pada gambar 4.15, dari ke-11 *tester*, 3 diantaranya memberikan nilai 4 yang berarti baik, dan 8 diantaranya memberikan nilai 5 yang berarti sangat baik.

VI. KESIMPULAN DAN SARAN

Berdasarkan hasil penelitian dan pengujian, didapatkan berbagai kesimpulan bahwa penerapan tanda tangan digital dalam kode QR sebagai metode alternatif untuk akses masuk kantor dapat memberikan berbagai kemudahan dan keuntungan dibandingkan sistem akses yang ada saat ini. Sistem ini memanfaatkan pendekatan *edge computing* untuk verifikasi tanda tangan digital pada kode QR, sehingga proses verifikasi dapat dilakukan langsung di perangkat pemindai tanpa perlu selalu terhubung dengan internet. Hal ini menjadikannya lebih responsif dan adaptif terhadap berbagai situasi, serta mengurangi ketergantungan pada konektivitas internet, sehingga membuat sistem lebih andal.

Berikut merupakan beberapa saran yang dapat disampaikan oleh penulis terkait penelitian ini kepada

pihak yang membaca tugas akhir ini untuk dapat melakukan pengembangan lebih lanjut dan juga dapat memperbaiki kekurangan yang ada.

1. Mengembangkan aplikasi pegawai berbasis *mobile*. Hal ini dapat mempermudah pegawai dalam mendapatkan kode QR akses dan meningkatkan keandalan sistem untuk membangkitkan kode QR akses lokal tanpa harus melakukan *request* ke server.
2. Menggunakan komponen-komponen perangkat keras IoT, seperti sensor, perangkat pemindai QR khusus, dan mikrokontroler, hal ini dapat meningkatkan efisiensi dan keandalan sistem karena perangkat ini dirancang khusus untuk tugas-tugas yang spesifik dan dapat beroperasi secara mandiri dengan konsumsi daya yang rendah dan konektivitas yang optimal.
3. Melakukan integrasi dengan sistem presensi pegawai. Hal ini dapat meningkatkan efisiensi dan akurasi pencatatan kehadiran pegawai karena data akses masuk melalui kode QR otomatis tercatat dalam sistem presensi. Sistem kolaborasi ini dapat menghemat waktu dan biaya administratif karena data dikumpulkan dan diproses secara otomatis sehingga data presensi yang terintegrasi dapat memudahkan analisis dan menghasilkan laporan yang lebih komprehensif.

REFERENCES

[1] G. Eason, B. Noble, and I. N. Sneddon, “On certain integrals of Lipschitz-Hankel type involving products of Bessel functions,” Phil. Trans. Roy. Soc. London, vol. A247, pp. 529–551, April 1955. *(references)*

[2] Amara, M., & Siad, A. (2011, May). Elliptic curve cryptography and its applications. In International workshop on systems, signal processing and their applications, WOSSPA (pp. 247-250). IEEE.

[3] Alshaikhli, I. F., Alahmad, M. A., & Munthir, K. (2012, November). Comparison and analysis study of SHA-3 finalists. In 2012 International Conference on Advanced Computer Science Applications and Technologies (ACSAT) (pp. 366-371). IEEE

[4] Chao, Chia-Wei, Daniel Winden Hwang, Hung-Wen Tsai, Shih-Hsuan Lin, Wei-Li Chen, Chun-Rong Huang, and Pau-Choo Chung. 2023. "Multi-Magnification Attention Convolutional Neural Networks [AI-eXplained]." *IEEE Computational Intelligence Magazine* (IEEE) 18 (3): 54-55.

[5] Duke, Kellen. 2019. *Buildings website.* Diakses pada 13 November 2023. https://www.buildings.com/building-systems-om/automation-controls/article/10185529/access-control-technology-is-eating-key-cards.

[6] Dworkin, M. 2015. *SHA-3 Standard: Permutation-Based Hash and Extendable-Output Functions*. Federal Inf. Process. Stds. (NIST FIPS), National-Institute of Standards and Technology, Gaithersburg, MD.https://doi.org/10.6028/NIST.FIPS.202

[7] Hankerson, D., Vanstone, S., Menezes, A. 2004. *Guide to Elliptic Curve Cryptography.* Springer.

[8] IBM Security. 2023. *Cost of a Data Breach Report*. Diakses pada 13 Oktober 2023

[9] Jamil, Amirah. 2017. *Current Issues and Challenges of Fingerprint Recognition.* International Conferences on Information Technology and Business (ICITB). (pp. 147-152). ICITB

[10] George, A. Shaji. "The Dawn of Passkeys: Evaluating a Passwordless Future." *Partners Universal Innovative Research Publication* 2, no. 1 (2024): 202-220. https://doi.org/10.5281/zenodo.10697886.

[11] Kuacharoen, P., & Warasart, M. (2012). Paper-based document authentication using digital signature and qr code. In International Conference on Computer Engineering and Technology (pp. 1-5).

[12] Larson, Richard C. 2023. *Model Thinking for Everyday Life.* I N F O R M S: Institute for Operations Research & the Management Sciences.

[13] Maltoni, D., Maio, D., Jain, A. K., Prabhakar, S. (2009). *Handbook of fingerprint recognition*. Springer.

[14] Munir, Rinaldi. 2023. Slide Kuliah II4031 Kriptografi dan Koding: Algoritma RSA

[15] Munir, Rinaldi. 2023. Slide Kuliah II4031 Kriptografi dan Koding: Elliptic Curve Cryptography (ECC)

[16] Munir, Rinaldi. 2023. Slide Kuliah II4031 Kriptografi dan Koding: Tanda Tangan Digital

[17] NIST. 2023. *An Introduction to Computer Security: The NIST Handbook.*

[18] Jamil, Amirah. 2017. *Current Issues and Challenges of Fingerprint Recognition.* International Conferences on Information Technology and Business (ICITB). (pp. 147-152). ICITB

[19] Pippal, R.S., C.D., J., Tapaswi, S. (2012). Security Vulnerabilities of User Authentication Scheme Using Smart Card. In: Cuppens-Boulahia, N., Cuppens, F., Garcia-Alfaro, J. (eds) Data and Applications Security and Privacy XXVI. DBSec 2012. Lecture Notes in Computer Science, vol 7371. Springer, Berlin, Heidelberg. https://doi.org/10.1007/978-3-642-31540-4_8

[20] Ratha, N. K., Connell, J. H., Bolle, R. M. (2001). *Enhancing Security and Privacy in Biometrics-Based Authentication Systems*. IBM Systems Journal, vol. 40, no. 3, pp. 614-634, 2001, doi: 10.1147/sj.403.0614

[21] Sadikin, M. A., & Sunaringtyas, S. U. (2016 Implementing digital signature for the secure electronic prescription using QR-code based on android smartphone. In 2016 International Seminar on Application for Technology of Information and Communication (ISemantic) (pp. 306-311). IEEE.

[22] Segara, Patrick. 2023. *Framework Implementasi Green IT pada Perguruan Tinggi (Studi Kasus: Institut Teknologi Bandung)*. Tugas Akhir, Sekolah Teknik Elektro dan Informatika, Institut Teknologi Bandung, Bandung: Institut Teknologi Bandung.

[23] Schwartz, Jeffrey. "Google Expands Passkey Support With Passwordless Authentication." Dark Reading, May 5, 2023. Accessed June 20, 2024. https://www.darkreading.com/application-security/google-expands-passkey-support-with-passwordless-authentication.

[24] Shinde, Krishna & Kayte, C. 2022. *Fingerprint Recognition Based on Deep Learning Pre-Train with Our Best CNN Model for Person Identification.* The Electrochemical Society. https://doi.org/10.1149/10701.2209ecst

[25] t.t. "University of NewcastleUniversity of Newcastle Library Guides." *Chicago B: Author-Date Style: In-text citations.* Diakses pada 22 Oktober 2023. https://libguides.newcastle.edu.au/chicago-b/in-text.

[26] t.t. *Cara Mengutip Sumber Acuan Dalam Format Chicago Manual of Style.* Diakses pada 1 Oktober 2023. https://id.wikihow.com/Mengutip-Sumber-Acuan-Dalam-Format-Chicago-Manual-of-Style.

[27] Larson, Richard C. 2023. *Model Thinking for Everyday Life.* I N F O R M S: Institute for Operations Research & the Management Sciences.

[28] Varela, M., Pomares, J. (2019). QR Code-Based Access Control System with Off-Line Verifiable Codes. IEEE Transactions on Industrial Informatics

[29] Vanstone, S. A. (1997). Elliptic curve cryptosystem—the answer to strong, fast public-key cryptography for securing constrained environments. Information security technical report, 2(2), 78-87.

[30] Y. Taigman, M. Yang, M. Ranzato and L. Wolf, "DeepFace: Closing the Gap to Human-Level Performance in Face Verification," 2014 IEEE Conference on Computer Vision and Pattern Recognition, Columbus, OH, USA, 2014, pp. 1701-1708, doi: 10.1109/CVPR.2014.220