

# Rancang dan Bangun Sistem Informasi Presensi QR Code Berbasis Web Menggunakan Database Firebase

Wildana Alfi Anzaqi<sup>1</sup>, Novian Adi Prasetyo<sup>2</sup>, Fatimah Az Zahra<sup>3\*</sup>

<sup>1,2,3</sup> *Teknik Informatika, Fakultas Informatika, Institut Teknologi Telkom Purwokerto  
Jl. DI Panjaitan No 128, Purwokerto Selatan, Banyumas, Jawa Tengah, Indonesia 53147*

<sup>1</sup> 16102214@ittelkom-pwt.ac.id

<sup>2</sup> novian@ittelkom-pwt.ac.id

<sup>3</sup> 2211102160@ittelkom-pwt.ac.id

Received on 30-10-2023, revised on 07-11-2023, accepted on 15-11-2023

## Abstract

Kegiatan presensi memiliki peranan penting sebagai pendataan kehadiran di berbagai institusi, contohnya di sekolah, universitas, perkantoran, maupun rumah sakit. Presensi penting karena berfungsi sebagai bukti kehadiran seseorang dan pendukung proses belajar mengajar di bagian administrasi. Berdasarkan dari permasalahan yang didapat oleh penulis melalui wawancara maka dengan penelitian yang dilakukan penulis diharapkan dapat membantu menyelesaikan permasalahan tersebut dengan cara dibuatnya sistem informasi presensi QR Code berbasis web menggunakan database Firebase dengan menggunakan metode waterfall. Pengujian aplikasi menggunakan *black box testing* dengan menguji aplikasi sesuai rancangan *test-case* yang telah dibuat penulis. Hasil pengujian yang didapat dari rancangan *test-case* A001, A002, A003, A004, A005, A006 dan B001 mendapatkan kesimpulan “berhasil/valid” pada tiap pengujian. Hal ini dapat disimpulkan bahwa aplikasi telah berfungsi sesuai fitur fungsionalitasnya.

**Keywords:** black box testing, cloud, database, server, web.

*This is an open access article under the [CC BY-SA](#) license.*



---

### Corresponding Author:

Fatimah Az Zahra  
Institut Teknologi Telkom Purwokerto (ITTP)  
Jl. DI Panjaitan No 128, Purwokerto Selatan, Banyumas, Jawa Tengah, Indonesia 53147  
Email: 2211102160@ittelkom-pwt.ac.id

---

## I. PENDAHULUAN

PRESENSI merupakan kegiatan yang memiliki peranan penting sebagai pendataan kehadiran di berbagai institusi, seperti di sekolah, universitas, perkantoran maupun rumah sakit [1]. Kehadiran memiliki peranan penting karena berperan sebagai tanda bahwa seseorang hadir dan membantu dalam pengelolaan administrasi dalam proses belajar mengajar. Di dalam lingkungan sekolah, presensi juga berfungsi sebagai alat informasi yang menunjukkan sejauh mana kedisiplinan guru dan siswa. Sistem presensi telah mengalami perkembangan dari cara manual dengan pencatatan tangan, dan sekarang semakin banyak yang mengadopsi sistem presensi digital [2]. Saat ini, presensi telah ditingkatkan dengan dukungan sistem informasi yang bertujuan untuk menghasilkan ringkasan data presensi selama periode tertentu.

Ringkasan ini akan berperan dalam menilai tingkat kedisiplinan, dan di lingkungan universitas, menjadi salah satu faktor penentu apakah seorang mahasiswa memiliki kelayakan untuk mengikuti ujian atau tidak[3].

Pada tanggal 2 Maret 2020, Presiden Republik Indonesia, Joko Widodo, mengumumkan bahwa telah ada warga Negara Indonesia yang telah dinyatakan positif terinfeksi virus Covid-19[4]. Diketahui, virus Covid-19 menyebar dengan cepat di berbagai Negara termasuk Indonesia. Berdasarkan data situs indonesia.go.id tercatat pada Senin (2/3/2020), jumlah kasus Covid-19 telah mencapai angka 88.382[5]. Penyakit ini dapat menular melalui partikel kecil yang dikeluarkan dari hidung atau mulut saat seseorang batuk atau bersin. Kemudian, jika seseorang yang tidak sadar menyentuh permukaan yang sudah terkontaminasi oleh partikel-partikel tersebut, yang sering disebut sebagai droplet, dan kemudian menyentuh mata, hidung, atau mulut, maka orang tersebut berisiko terpapar virus Covid-19[6]. Karena virus Covid-19, banyak lembaga, perusahaan, pemerintahan, dan sekolah telah beralih ke sistem presensi yang mengurangi kontak fisik dengan objek, seperti menggunakan teknologi seperti pengenalan wajah, kode QR, dan verifikasi lokasi koordinat[7].

Dari informasi dan penjelasan di atas, muncul peluang untuk menjadikan topik penelitian. Setelah peneliti melakukan pengamatan di Sekolah Dasar Negeri 2 Pajerukan dan melakukan wawancara dengan seorang narasumber, peneliti memperoleh pemahaman mendalam tentang penggunaan sistem presensi yang melibatkan verifikasi wajah atau face reader. Namun, terdapat sejumlah masalah yang timbul dalam penggunaan sistem ini, seperti kesulitan mendeteksi wajah saat melakukan pemindaian, potensi perubahan data presensi oleh pihak yang tidak berwenang, kerusakan peralatan pembaca yang disebabkan oleh petir, dan ketiadaan server untuk menyimpan data, sehingga data absensi tidak terhubung dengan komputer sekolah.

Berdasarkan permasalahan yang telah dijelaskan, penelitian yang dilakukan penulis diharapkan dapat memberikan solusi melalui pengembangan sistem informasi dan presensi menggunakan teknologi QR Code yang berbasis Firebase. Pemilihan platform berbasis web untuk aplikasi ini dilakukan karena memiliki keunggulan yang memungkinkan penggunaannya di berbagai perangkat, termasuk komputer, tablet, dan smartphone. Aplikasi berbasis web tidak memerlukan instalasi karena semua proses telah dijalankan di server, sehingga pengguna dapat segera memanfaatkan fitur yang tersedia tanpa perlu menginstal apa pun[8]. Aplikasi ini memiliki dua fitur utama, yaitu login dan fitur presensi khusus untuk guru. Dengan melakukan penelitian ini, harapannya adalah bahwa aplikasi yang dibuat dapat mengatasi kendala yang ada dalam sistem presensi guru yang saat ini masih menggunakan metode manual.

## II. METODE PENELITIAN

### A. Tahapan Penelitian

Metode penelitian ini diawali dengan mengidentifikasi permasalahan hingga menentukan menentukan objek & subjek penelitian, melakukan wawancara, menentukan rumusan masalah, pengembangan aplikasi, hingga dihasilkannya produk sistem informasi presensi QR Code berbasis web. Untuk lebih jelasnya dapat dilihat pada gambar 1 berikut.



Gambar 1. Tahapan Penelitian

1. Identifikasi Masalah dan Studi Literatur  
 Dari konteks yang telah diperkenalkan dalam bagian Pendahuluan, dapat disimpulkan bahwa kegiatan presensi memiliki peran sentral dalam proses pembelajaran, sebagai indikator penting untuk mengukur tingkat disiplin siswa di SDN 2 Pajerukan. Permasalahan yang menjadi fokus penelitian adalah berkaitan dengan ketidakberfungsian sistem presensi berbasis *face reader* yang berdampak pada penggunaan metode manual kembali oleh para guru di institusi tersebut.
2. Menentukan Subjek dan Objek Penelitian  
 Dalam penelitian ini, subjek yang menjadi fokus adalah para guru yang mengajar di SDN 2 Pajerukan, sementara objek penelitiannya adalah sistem presensi konvensional yang digunakan di SDN 2 Pajerukan.
3. Melakukan Wawancara  
 Dalam rangka penelitian ini, penulis melakukan sesi wawancara dengan seorang guru yang bertugas di SDN 2 Pajerukan, yang dilakukan dengan menyajikan sejumlah pertanyaan.
4. Hasil Wawancara  
 Menurut hasil wawancara yang dilakukan oleh penulis, SDN 2 Pajerukan saat ini masih menggunakan sistem presensi konvensional. Guru-guru telah mencoba menggunakan presensi berbasis *face reader*, namun menghadapi sejumlah kendala dalam pelaksanaannya. Alasan utama penggunaan sistem konvensional adalah karena sebagian besar siswa di sekolah tersebut belum memiliki *smartphone*. Beberapa masalah yang dihadapi dalam implementasi presensi digital melalui *face reader* atau *fingerpint* antara lain, alat pembaca seringkali tidak mendeteksi dengan baik, data presensi dapat diedit oleh siapa saja, dan pernah terjadi kerusakan alat akibat tersambar petir. Proses rekapitulasi presensi masih dilakukan secara manual.  
 Kepala sekolah SDN 2 Pajerukan berpendapat bahwa sistem presensi yang ideal adalah yang mudah digunakan, tidak dapat dimanipulasi oleh pihak yang tidak berwenang, serta mampu menghasilkan rekapitulasi secara otomatis. Selain itu, sekolah juga menginginkan aplikasi yang mencakup profil sekolah dan data siswa, agar dapat diakses dengan mudah melalui *smartphone* saat diperlukan.
5. Pengembangan Aplikasi  
 Pengembangan aplikasi ini mengadopsi pendekatan metode air terjun atau *Waterfall*. Alasan pemilihan metode ini didasari oleh kejelasan kebutuhan aplikasi yang sudah didefinisikan secara komprehensif pada awalnya dan karena kemudahan penerapannya. Metode *Waterfall* terkenal dengan pendekatan yang sistematis dan terstruktur, melalui tahap-tahap yang berurutan, dimulai dari perencanaan hingga implementasi [9]. Menurut dengan F. Ikhsan [10], metode air terjun (*Waterfall*) sering juga disebut sebagai model sekuensial linier atau alur hidup klasik.

### III. HASIL DAN PEMBAHASAN

Dalam pengembangan sistem informasi presensi berbasis QR Code ini, prosesnya terbagi menjadi beberapa tahap. Tahap pertama adalah tahap pengembangan aplikasi, yang terdiri dari dua sub-tahap, yaitu pengembangan *backend* (termasuk pengumpulan data, pembuatan database, konfigurasi web server, inialisasi FastAPI, serta penjelasan tentang setiap Endpoint) dan pengembangan *frontend* yang berkaitan dengan tampilan antarmuka aplikasi. Tahap kedua adalah tahap pengujian aplikasi.

#### A. Pengkodean *Backend*

1. Proses Pengumpulan data dan pembuatan database

Dalam proses pengumpulan data, penulis membuat sebuah basis data dengan menggunakan Google Spreadsheet. Data yang digunakan diambil dari informasi peserta didik tahun ajaran 2022/2023 di SDN 2 Pajerukan, yang dapat diakses melalui tautan berikut: <https://bit.ly/SKRIPSI-DATA-PESDIK>. Setelah data berhasil terkumpul dalam *spreadsheet*, langkah selanjutnya adalah membuat sebuah basis data menggunakan Firebase sebagai tempat penyimpanan cloud. Untuk mengintegrasikan API Firebase, penulis mengakses kredensial dan firestore. Sebelumnya, penulis juga melakukan beberapa langkah persiapan, seperti menggunakan perintah "from dotenv import load\_dotenv, dotenv\_values, find\_dotenv" yang digunakan untuk membaca data dari file *.env*. Hal ini penting agar data kredensial dapat diakses dari setiap perangkat yang terhubung ke Web Server ini dengan aman.

Kredensial yang telah diambil dari .env masing-masing perangkat digunakan untuk login menggunakan API Firebase tersebut dan firestore sebagai database yang ada pada Firebase. Pada line `cred=credential.Certificate` akan mengambil data autentifikasi atau token yang tersimpan sebagai file “geofencing-94da1-firebase-adminsdk-wym11-b451bf4e8c.json”. Pada `if cred:` jika value yang didapat adalah “True” maka autentifikasi sudah berhasil. Setelah autentifikasi berhasil maka pada line `firebaseApp = firebase_admin.initialize_app(credential=cred)` akan diiniasi `firebaseApp` dengan kredensial yang sudah ada tadi. Namun API Firebase ini belum dapat digunakan, karena diperlukan konfigurasi untuk autentifikasi sehingga pengguna dapat menggunakan aplikasi ini sesuai role masing – masing.

Pembuatan struktur Web Server yang berfungsi sebagai program untuk menjalankan API dan library – library pendukung lainnya.

Namun API Firebase ini belum dapat digunakan, karena diperlukan konfigurasi untuk autentifikasi sehingga pengguna dapat menggunakan aplikasi ini sesuai role masing – masing. Baris kode di bawah ini menjelaskan proses pengguna login pada aplikasi dan kemudian akan tercatat pada list database. Pada Class `Auth` dibuat Singleton karena *class/object* autentifikasi hanya diperlukan satu kali objek dalam runtime/eksekusi aplikasi ini. Hal ini dibutuhkan untuk menjaga agar aplikasi berjalan secara efisien dan tidak membuang – buang alokasi memori karena . Pada baris kode `class AuthDataStore(AuthSingleton)` dijelaskan proses pembuatan data dan juga memiliki fungsi untuk membuat list data pada Class `AuthSingleton`. Dapat dilihat pada line `def __init__(self)` dan seterusnya menjelaskan mengenai konstruktor yang dimana fungsi tersebut akan dipanggil pertama kali saat sebuah objek dibuat. Pada baris kode `def add_data(self, new_data)` dijelaskan fungsi `add_data` yang memiliki parameter `self` untuk menambahkan data baru pada variable `datas`. Pada baris kode selanjutnya `def get_datas(self) -> list` berfungsi.

## 2. Proses pembuatan web server

Dalam tahap ini, Web Server dikembangkan menggunakan pustaka `FastAPI` dan pustaka `uvicorn`. Proses ini melibatkan pengimporan pustaka `FastAPI` dan `uvicorn` ke dalam file `main.py`. Penggunaan `FastAPI` bertujuan untuk menghilangkan loop yang tidak diperlukan, sehingga memungkinkan aplikasi berjalan dengan lebih efisien. Selain itu, baris kode `"from fastapi.middleware.cors import CORSMiddleware"` digunakan sebagai alat bagi pengembang untuk membuat dan mengatur API dalam aplikasi ini. Penggunaan `"import uvicorn"` membantu menyederhanakan koneksi antara Web Server dan memungkinkan `FastAPI` untuk mengelola permintaan secara *asynchronous* atau independen, tanpa perlu menjalankannya bersamaan. Ini meningkatkan fleksibilitas aplikasi dan mempercepat proses yang dijalankan oleh Web Server.

## 3. Proses Inisiasi FastAPI

Pada tahapan ini menjelaskan proses ketika Web Server yang penulis buat akan menggunakan `FastAPI` dan dapat dilihat pada line `app = FastAPI()`. Pada line selanjutnya `app.add_middleware()` berfungsi menambahkan *library middleware* pada aplikasi ini.

## 4. Penjelasan Masing-Masing Endpoint

### a) Authentication

#### 1) {{HOST}}/auth/login [METHOD : POST]

Sesuai dengan namanya, *endpoint* ini adalah titik akhir publik yang digunakan untuk mengotentikasi pengguna dalam aplikasi. Untuk melakukan otentikasi, pengguna harus menyertakan *username* dan *password* dalam permintaan yang dikirimkan melalui *request body*. Agar proses login berhasil, data yang dimasukkan oleh pengguna harus sesuai dengan informasi yang terdapat dalam basis data Firebase aplikasi manajemen presensi. Data yang dikirimkan dalam *request body* memiliki tipe data JSON.

### b) Master Data

#### 1) {{HOST}}/people [METHOD : GET]

merupakan titik akhir yang akan memberikan daftar semua nama individu yang terdaftar dalam sistem. Untuk mengakses titik akhir ini, Anda harus menggunakan `access_token`. Tidak ada parameter, kueri, atau data permintaan yang diperlukan untuk mengakses titik akhir ini.

#### 2) {{HOST}}/people [METHOD : POST]

Meskipun memiliki nama yang sama dengan *endpoint* sebelumnya, pengaksesan ke *endpoint* ini memerlukan metode permintaan HTTP yang berbeda, yaitu POST. Untuk mengakses titik akhir ini, diperlukan `access_token`, dan juga harus mengirimkan data dalam format JSON melalui *request body*. Data tersebut diperlukan oleh server backend untuk membuat entri nama individu dalam basis data Firebase.

#### 3) {{HOST}}/people/:person\_id [METHOD : GET]

Merupakan endpoint yang akan memberikan rincian informasi mengenai individu yang terdaftar dalam sistem. Untuk mengakses titik akhir ini, Anda harus memanfaatkan `access_token`, dan juga harus menyertakan parameter berupa id orang (`person_id`). Berikut adalah contoh tanggapan yang diberikan oleh server kepada klien saat permintaan ke titik akhir ini berhasil:

4) `{{HOST}}/people/:person_id [METHOD : PATCH]`

*Endpoint* ini dirancang untuk melakukan pembaruan pada data individu yang terdaftar dalam sistem. Untuk mengakses titik akhir ini, diperlukan `access_token`. Selain itu, perlu menyediakan parameter berupa id orang (`person_id`) dan juga mengirimkan data dalam format JSON melalui request body agar dapat memanfaatkan titik akhir ini.

c) *Employee Management*

1) `{{HOST}}/employee [METHOD : GET]`

Merupakan *endpoint* yang akan memberikan daftar semua nama pegawai yang tercatat dalam sistem. Struktur data pegawai merupakan turunan dari data individu (`person`) dan peran yang bersifat statis dalam basis data Firebase. Data pegawai mencakup informasi individu dan perannya. Untuk mengakses titik akhir ini, diperlukan `access_token`. Tidak ada parameter, kueri, atau data permintaan yang diperlukan untuk mengakses titik akhir ini.

2) `{{HOST}}/employee [METHOD : POST]`

*Endpoint* yang digunakan untuk menyertakan informasi pegawai baru dengan menggunakan metode permintaan HTTP POST. Untuk mengakses titik akhir ini, Anda harus memanfaatkan `access_token`. Selain itu, perlu mengirimkan data dalam format JSON melalui request body yang berisi informasi yang diperlukan oleh server *backend* untuk membuat entri pegawai dalam basis data Firebase. Karena pegawai merupakan turunan data individu dan peran, maka data yang harus disertakan dalam request body adalah `person_id` dan `role_id`.

3) `{{HOST}}/employee/:employee_id [METHOD : GET]`

*Endpoint* yang memberikan informasi rinci mengenai pegawai yang terdaftar dalam sistem. Untuk dapat mengakses titik akhir ini, dibutuhkan `access_token`. Selain itu, Anda perlu memberikan parameter berupa id pegawai (`employee_id`) agar dapat menggunakan titik akhir ini.

4) `{{HOST}}/employee/:employee_id [METHOD : PATCH]`

Endpoint ini digunakan untuk memberikan detail lengkap mengenai pegawai yang tercatat dalam sistem. Akses ke titik akhir ini memerlukan keberadaan `access_token`. Selain itu, Anda harus mengirimkan parameter berupa id pegawai (`employee_id`) untuk menggunakan titik akhir ini.

d) *Role Management*

1) `{{HOST}}/role [METHOD : GET]`

Merupakan *endpoint* yang akan menyajikan daftar lengkap nama semua pegawai yang tercatat dalam sistem. Data pegawai merupakan turunan dari data individu (`person`) dan peran yang bersifat statis dalam basis data Firebase. Data pegawai mencakup informasi individu dan peran mereka. Untuk dapat mengakses titik akhir ini, Anda perlu menggunakan `access_token`. Tidak ada parameter, kueri, atau data permintaan yang dibutuhkan untuk mengakses titik akhir ini.

2) `{{HOST}}/role/:role_id [METHOD : GET]`

*Endpoint* yang digunakan untuk memberikan rincian informasi mengenai peran yang tercatat dalam sistem dan digunakan untuk mengidentifikasi antarmuka pengguna dalam sistem. Akses ke titik akhir ini memerlukan keberadaan `access_token`. Selain itu, Anda harus memberikan parameter berupa id peran (`role_id`) untuk menggunakan titik akhir ini.

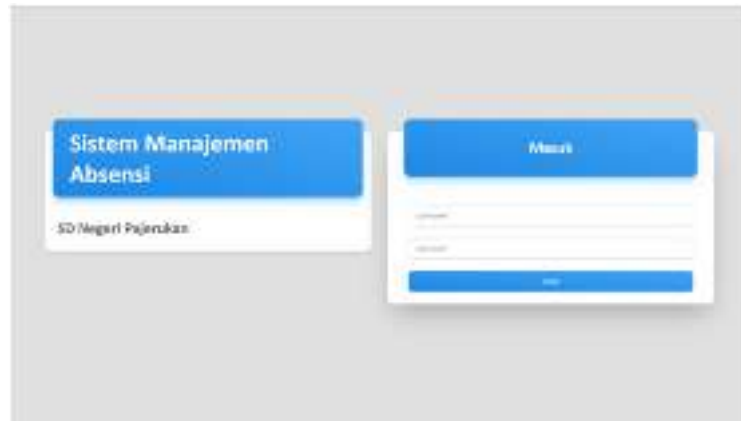
e) *Attendance*

1) `{{HOST}}:5000/attendance [METHOD : POST]`

*Endpoint* yang digunakan untuk mencatat kehadiran dalam basis data. Proses ini melibatkan pengiriman `access_token` dan informasi koordinat yang terkait dengan perangkat pengguna. `Access_token` digunakan sebagai bagian dari permintaan body untuk menghubungkan data antara individu (orang) dan id presensi dalam database. Hal ini bertujuan untuk mempermudah akses data kehadiran, baik dari perspektif pengguna maupun dalam pengembangan aplikasi. Selain itu, detail koordinat juga diperlukan dalam pencatatan ini, dan digunakan untuk melakukan validasi data di sisi klien guna memeriksa apakah pengguna

berada di lokasi yang sesuai dengan yang telah ditentukan oleh server *backend*. Berikut ini adalah rincian yang diperlukan dalam request body oleh server *backend*.

#### B. Implementasi Antarmuka



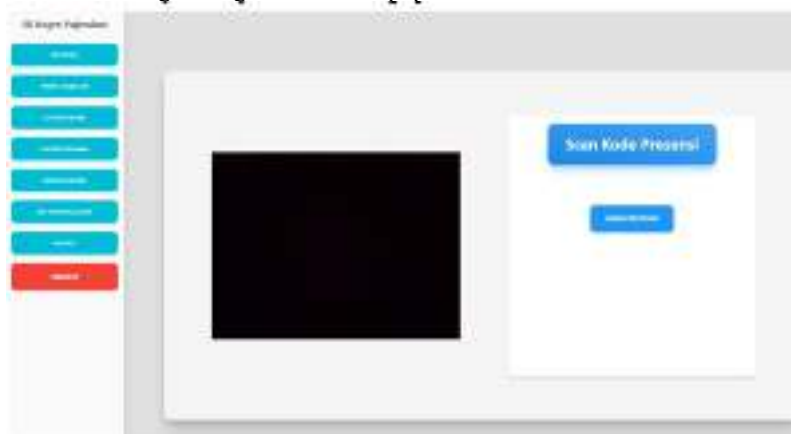
Gambar 2. Tampilan Halaman Login

Gambar 1 menampilkan tampilan antarmuka untuk halaman login yang telah diimplementasikan. Antarmuka ini memiliki dua formulir, yaitu untuk memasukkan *username* dan *password*. Saat pengguna mengisi kolom *password*, nilai yang dimasukkan tidak ditampilkan untuk menjaga kerahasiaan data pengguna. Halaman Login ini juga dilengkapi dengan tombol "Submit" yang bertujuan untuk mengirim data yang telah dimasukkan oleh pengguna ke dalam database.



Gambar 3. Halaman Profil Sekolah

Gambar 2 menunjukkan implementasi halaman Profil Sekolah yang berisi informasi tentang visi dan misi SD N 2 Pajerukan. Halaman ini dirancang hanya untuk menampilkan informasi mengenai profil sekolah, dan tidak memiliki fungsi-fungsi interaktif apapun.



Gambar 4. Tampilan Halaman presensi Fitur Scan Presensi

Gambar 3 menampilkan implementasi halaman Presensi dalam fitur Scan Presensi. Di halaman ini, pengguna dapat melakukan proses presensi. Setelah pengguna berhasil masuk, mereka dapat melakukan presensi dengan mengklik tombol "Scan Presensi". Pengguna akan melakukan pemindaian menggunakan QR Code yang telah dibuat oleh admin. Setelah berhasil melakukan pemindaian, data presensi akan disimpan dalam database, dan sistem akan memberikan pesan bahwa pemindaian presensi berhasil dilakukan.

#### C. Fase Pengujian Aplikasi

Pada tahap ini, metode pengujian yang diterapkan adalah *black box testing*. *Black box testing* menggunakan metode pengujian tipe tertentu yang disebut *equivalence partitioning testing*. Metode ini berfokus pada pemeriksaan jenis dan kondisi input yang sesuai dengan *Software Requirements Specification* (SRS). Dalam konteks ini, input dibagi menjadi beberapa kelas ekuivalensi, yaitu kelas yang valid dan kelas yang tidak valid.

Rencana pengujian akan mengacu pada tabel *test case* dari aplikasi yang dibuat. Kemudian, pengujian akan dimulai dengan menginisialisasi hasilnya, dan hasil dari pengujian tersebut akan didokumentasikan. Tabel *test case* digunakan sebagai skenario pengujian untuk menentukan apakah pengujian berhasil atau tidak. Tabel 1 berisi rancangan *test case* untuk pengujian fitur Login, dengan pengujian yang difokuskan pada kolom *username*. Skenario pengujian dalam hal ini adalah mencoba mengosongkan kolom *username* dan kemudian menekan tombol Login. Hasil yang diharapkan dari *test case* ini adalah bahwa sistem akan menolak perintah inputan tersebut, dan proses Login tidak akan dilanjutkan.

Tabel 1 Rancangan *test case* pengujian *black box*

Id	SKENARIO PENGUJIAN	HASIL YANG DIHARAPKAN
A001	Mengosongkan kolom <i>username</i> dan <i>password</i> kemudian menekan tombol <i>Login</i>	Sistem akan menolak <i>inputan</i> , dan akan memberi peringatan, dan proses tidak akan berlanjut
A002	Memasukkan "Admin" pada kolom <i>username</i> , dan mengosongkan kolom <i>password</i> , kemudian menekan tombol <i>Login</i>	Sistem akan menolak <i>inputan</i> , dan akan memberi peringatan, dan proses tidak akan berlanjut
A003	Mengosongkan kolom <i>username</i> , memasukkan "12345612" pada kolom <i>password</i> , kemudian menekan tombol <i>Login</i>	Sistem akan menolak <i>inputan</i> , dan akan memberi peringatan, dan proses tidak akan berlanjut
A004	Memasukkan "Dewi" pada kolom <i>username</i> , memasukkan "12345612" pada kolom <i>password</i> , kemudian menekan tombol <i>Login</i>	Sistem akan menolak <i>inputan</i> , dan akan memberi peringatan, dan proses tidak akan berlanjut
A005	Memasukkan "admin_widana" pada kolom <i>username</i> , dan memasukkan "12345612" pada kolom <i>password</i> , kemudian menekan tombol <i>Login</i>	Sistem akan menerima <i>inputan</i> , dan akan berlanjut ke proses berikutnya
A006	Memasukkan "dewi_ratnawati" pada kolom <i>username</i> , dan memasukkan "12345612" pada kolom <i>password</i> , kemudian menekan tombol <i>Login</i>	Sistem akan menerima <i>inputan</i> , dan akan berlanjut ke proses berikutnya
B001	Melakukan <i>Scan QR Code</i> dengan <i>Smartphone</i>	Sistem akan memproses, dan akan memvalidasi QR Code tersebut, kemudian jika berhasil data presensi akan masuk pada database.

#### D. Hasil Pengujian Aplikasi

Setelah melewati serangkaian pengujian aplikasi dengan berbagai skenario yang tercatat dalam rancangan *test case*, hasil pengujian akan dicatat dalam formulir pengujian. Setelah formulir pengujian terisi dan hasilnya dianalisis, kesimpulan akan diambil dengan membandingkan hasil yang diperoleh dengan hasil yang diharapkan. Jika hasil pengujian sesuai dengan apa yang telah diantisipasi dalam rancangan *test case*, maka kesimpulan yang diberikan adalah berhasil atau valid. Namun, jika hasil pengujian tidak sesuai dengan harapan yang ada dalam rancangan *test case*, maka kesimpulan yang diberikan adalah tidak berhasil atau tidak valid.

Tabel di bawah ini berisi kesimpulan dari hasil pengujian yang telah saya lakukan berdasarkan fase pengujian yang telah dijalankan sesuai dengan rancangan test case.

Tabel 2 Hasil Pengujian

<b>Id</b>	<b>Skenario Pengujian</b>	<b>Test Case</b>	<b>Hasil Yang Diharapkan</b>	<b>Hasil Pengujian</b>	<b>Kesimpulan</b>
A001	Mengosongkan kolom <i>username</i> dan <i>password</i> kemudian menekan tombol <i>Login</i>	<i>Username:</i> - <i>Password:</i> -	Sistem akan menolak <i>inputan</i> , dan akan memberi peringatan, dan proses tidak akan berlanjut	Sesuai Harapan	Berhasil/Valid
A002	Memasukkan "Admin" pada kolom <i>username</i> , dan mengosongkan kolom <i>password</i> , kemudian menekan tombol <i>Login</i>	<i>Username:-</i> <i>Password:-</i>	Sistem akan menolak <i>inputan</i> , dan akan memberi peringatan, dan proses tidak akan berlanjut	Sesuai Harapan	Berhasil/Valid
A003	Mengosongkan kolom <i>username</i> , memasukkan "12345612" pada kolom <i>password</i> , kemudian menekan tombol <i>Login</i>	<i>Username:-</i> <i>Password:-</i>	Sistem akan menolak <i>inputan</i> , dan akan memberi peringatan, dan proses tidak akan berlanjut	Sesuai Harapan	Berhasil/Valid
A004	Memasukkan "Dewi" pada kolom <i>username</i> , memasukkan "12345612" pada kolom <i>password</i> , kemudian menekan tombol <i>Login</i>	<i>Username:-</i> <i>Password:-</i>	Sistem akan menolak <i>inputan</i> , dan akan memberi peringatan, dan proses tidak akan berlanjut	Sesuai Harapan	Berhasil/Valid
A005	Memasukkan "admin_widana" pada kolom <i>username</i> , dan memasukkan "12345612" pada kolom <i>password</i> , kemudian menekan tombol <i>Login</i>	<i>Username:</i> - <i>Password:-</i>	Sistem akan menerima <i>inputan</i> , dan akan berlanjut ke proses berikutnya	Sesuai Harapan	Berhasil/Valid
A006	Memasukkan "dewi_ratnawati" pada kolom <i>username</i> , dan memasukkan "12345612" pada kolom <i>password</i> , kemudian menekan tombol <i>Login</i>	<i>Username:</i> - <i>Password:</i> -	Sistem akan menerima <i>inputan</i> , dan akan berlanjut ke proses berikutnya	Sesuai Harapan	Berhasil/Valid
B001	Melakukan <i>Scan QR Code</i> dengan <i>Smartphone</i>		Sistem akan memproses, dan akan memvalidasi QR Code tersebut, kemudian jika berhasil data presensi akan masuk pada database.	Sesuai Harapan	Berhasil/Valid.

Berdasarkan hasil pengujian yang dilakukan penulis menggunakan *black box testing*. Dengan dilakukannya pengujian menggunakan metode *Equivalence Partitioning* didapatkan hasil pengujian tersebut yang kemudian didokumentasikan sebagai bahan evaluasi dan pengukuran pada nilai tingkat efektivitas. Hal ini dapat disimpulkan bahwa aplikasi telah berfungsi sesuai fungsionalitasnya. Kesimpulan hasil keseluruhan pengujian aplikasi ini dapat dilihat pada tabel 3.



Tabel 3 Kesimpulan hasil Pengujian

No	ID	Kesimpulan
1	A001	Berhasil
2	A002	Berhasil
3	A003	Berhasil
4	A004	Berhasil
5	A005	Berhasil
6	A006	Berhasil
7	B001	Berhasil

#### IV. KESIMPULAN

Sesuai dengan harapan hasil pengujian yang telah dirancang dalam rancangan test case, skenario yang telah digunakan untuk mengevaluasi interaksi pengguna dengan sistem menghasilkan hasil yang sesuai dengan yang diharapkan. Dengan pembuatan aplikasi ini dan melihat sejauh mana kesesuaiannya dengan tujuan penelitian yang telah ditetapkan, aplikasi ini telah berhasil mencapai tujuan penelitian, yaitu menciptakan sistem informasi presensi yang dapat mengatasi masalah para guru di SDN 2 Pajerukan. Meskipun aplikasi ini belum mencapai tahap implementasi penuh.

Berdasarkan hasil penelitian ini, penulis memberikan saran untuk meningkatkan kualitas penelitian dan pengembangan sistem. Aplikasi ini saat ini belum mencapai tahap implementasi sepenuhnya. Oleh karena itu, diharapkan bahwa peneliti di masa depan dapat mengembangkan aplikasi serupa dengan tingkat kesempurnaan yang lebih tinggi dan mencapai tahap implementasi. Selain itu, disarankan untuk melakukan penyempurnaan pada tampilan antarmuka pengguna (UI/UX) agar sesuai dengan prinsip-prinsip desain UI/UX, sehingga pengguna dapat menggunakan aplikasi dengan lebih nyaman dan interaksi antara pengguna dan sistem menjadi lebih intuitif.

#### REFERENSI

- [1] Y. Supendi, I. Supriadi, and A. A. W. Isto, "Pemanfaatan Teknologi QR-Code Pada Sistem Presensi Mahasiswa Berbasis Mobile," in *SEMINAR NASIONAL APTIKOM (SEMNASTIK) 2019*, 2019, pp. 550–558.
- [2] D. P. M. Putri and H. Supriyono, "Rancang Bangun Sistem Presensi Berbasis QR Code Menggunakan Framework Codeigniter (Studi Kasus Kehadiran Asisten Praktikum)," *Jurnal insypro*, vol. 4, no. 1, pp. 1–9, 2019.
- [3] H. D. Rjeib, N. S. Ali, A. Al Farawn, B. Al-Sadawi, and H. Alsharqi, "Attendance and information system using RFID and web-based application for academic sector," *International Journal of Advanced Computer Science and Applications*, vol. 9, no. 1, 2018.
- [4] R. N. Velarosdela, "Kilas Balik Kronologi Munculnya Kasus Pertama Covid-19 di Indonesia," *Retrieved from Kompas.com: <https://megapolitan.kompas.com/read/2021/03/02/05300081/kilas-balik-kronologi-munculnya-kasus-pertama-covid-19-di-indonesia>*, 2021 (accessed Aug.09 2022)
- [5] P. I. Indonesia, "Kasus Covid-19 Pertama, Masyarakat Jangan Panik," *Indonesia.go.id*, vol. 2019, pp. 20–25, 2020. <https://indonesia.go.id/narasi/indonesia-dalamangka/ekonomi/kasus-covid-19-pertama-masyarakat-jangan-panik%20>(accessed Aug.09, 2022).

- 
- [6] R. I. Kemenkes, “Infeksi Emerging Kementerian Kesehatan RI,” [infeksiemerging.kemkes.go.id](https://infeksiemerging.kemkes.go.id), 2020. <https://infeksiemerging.kemkes.go.id/situasi-infeksi-emerging/tanya-jawab-coronavirus-disease-covid-19-qna-update-6-maret-2020> (accessed Aug. 09, 2022).
- [7] D. Yusuf and F. N. Afandi, “Aplikasi absensi berbasis android menggunakan validasi kordinat lokasi dan nomor handphone guna menghindari penularan virus covid 19,” *EXPERT: Jurnal Manajemen Sistem Informasi Dan Teknologi*, vol. 10, no. 1, pp. 16–22, 2020.
- [8] H. Sulistiani, “Rancang Bangun Aplikasi Presensi SMS Gateway Berbasis Web Dengan Framework Codeigniter Pada SMKN 1 Trimurjo,” *Jurnal Informatika Dan Rekayasa Perangkat Lunak*, vol. 1, no. 1, pp. 43–50, 2020.
- [9] A. A. Wahid, “Analisis metode waterfall untuk pengembangan sistem informasi,” *J. Ilmu-ilmu Inform. dan Manaj. STMIK*, no. November, pp. 1–5, 2020.
- [10] I. Kurniawan and F. Rozi, “REST API Menggunakan NodeJS pada Aplikasi Transaksi Jasa Elektronik Berbasis Android,” *JITSI: Jurnal Ilmiah Teknologi Sistem Informasi*, vol. 1, no. 4, pp. 127–132, 2020.