Terakreditasi SINTA Peringkat 2
Surat Keputusan Dirjen Penguatan Riset dan Pengembangan Ristek Dikti No. 10/E/KPT/2019
masa berlaku mulai Vol. 1 No. 1 tahun 2017 s.d Vol. 5 No. 3 tahun 2021

Terbit online pada laman web jurnal: **http://jurnal.iaii.or.id**

JURNAL RESTI
(Rekayasa Sistem dan Teknologi Informasi)
Vol. 4 No. 6 (2020) 1085 – 1091 ISSN Media Elektronik: 2580-0760

Implementasi *JWT* pada Aplikasi Presensi dengan Validasi *Fingerprint*, *Geotagging* dan *Device Checker*

Arief Umarjati[1], Arief Wibowo[2]
[12]Teknik Informatika, Fakultas Teknologi Informasi, Universitas Budi Luhur
[1]ariefumarjati@gmail.com*, [2]arief.wibowo@budiluhur.ac.id

Abstract

During the Covid-19 pandemic the government implement the imposition of Large-Scale Social Restrictions (PSBB). This PSBB also has an impact on companies in Jabodetabek including PT Akses Digital Indonesia. In order to comply with regulations given by the government, PT Akses Digital Indonesia has implemented a Work From Home (WFH) policy for its employees. During the implementation of the WFH policy, had difficulty monitoring the performance of its employees. Attendance is one measure of the level of performance, especially employee discipline. Based on the identification of the problem, an employee presence web service application is needed. Of course, this application should be as effective as conventional fingerprint machines in offices. This application is accompanied by a validation feature using geotagging, fingerprint and device checkers to minimize fraud when employees make attendance. This study implements the RESTful API security feature on web services using JSON Web Token (JWT) based on the HMAC SHA-256 algorithm. All implementation stages are tested using the Black Box method and show that JWT can secure the authentication process and secure data. The validation feature is able to provide attendance data with an accuracy of 90,9%.

Keywords: Attendance, Web Service, Fingerprint, Geotagging, JSON Web Token

Abstrak

Pada masa pandemi *Covid-19* pemerintah membuat beberapa peraturan untuk mencegah penyebaran penyakit berbahaya tersebut. Salah satu kebijakan yang diterapkan adalah pemberlakuan Pembatasan Sosial Berskala Besar (PSBB). PSBB ini juga memberi dampak terhadap perusahaan-perusahaan di Jabodetabek termasuk PT Akses Digital Indonesia. Demi mematuhi peraturan yang diberikan pemerintah, PT Akses Digital Indonesia melakukan kebijakan *Work From Home* (WFH) bagi karyawannya. Selama diberlakukannya kebijakan WFH, PT Akses Digital Indonesia mengalami kesulitan untuk mengawasi kinerja dari para karyawan. Presensi adalah salah satu tolak ukur dari tingkat kinerja, terutama kedisplinan karyawan. Berdasarkan identifikasi masalah tersebut, diperlukan aplikasi *web service* presensi karyawan. Tentunya aplikasi ini diharuskan sama efektifnya dengan mesin *fingerprint* konvensional yang berada di kantor. Aplikasi ini disertai fitur validasi menggunakan *geotagging*, *fingerprint* dan *device checker* untuk meminimalisir adanya kecurangan saat karyawan melakukan presensi. Penelitian ini mengimplementasikan fitur keamanan RESTful API pada web services dengan menggunakan *JSON Web Token* (JWT) berbasis algoritma *HMAC SHA-256*. Seluruh tahap implementasi diuji menggunakan metode *Black Box* dan menunjukkan bahwa JWT dapat mengamankan proses autentikasi, melakukan proses *request* & *response*, dan pengamanan data. Selain itu, fitur validasi mampu memberikan data presensi dengan akurasi sebesar *90,9%*.

Kata kunci: Presensi, *Web Service*, *Fingerprint*, *Geotagging*, *JSON Web Token*

Diterima Redaksi : 19-11-2020 | Selesai Revisi : 13-12-2020 | Diterbitkan Online : 20-12-2020

1. Pendahuluan

Karyawan merupakan ujung tombak dari perusahaan. Memiliki karyawan dengan kedisplinan yang baik adalah aset yang paling berharga bagi perusahaan. Untuk itu pengawasan terhadap kedisplinan karyawan sangat perlu dilakukan. Tingkat kedisiplinan dapat dilihat salah satunya dengan nilai presensi pegawai. Semakin baik presensi pegawai maka semakin tinggi pula kedisiplinan pegawai tersebut [1].

Permasalahan muncul semenjak pandemi Covid-19 mulai mewabah di Indonesia. Presiden Indonesia Joko Widodo dalam pidatonya mengintruksikan untuk masyarakat Indonesia mengurangi kegiatan di luar rumah yang tidak penting. Termasuk menerapkan sistem kerja *Work From Home* (WFH) [2]. Demi mematuhi kebijakan tersebut, PT Akses Digital Indonesia juga memberlakukan WFH kepada seluruh karyawannya.

Presensi karyawan sangat penting untuk karyawan PT Akses Digital karena presensi masuk dalam salah satu parameter dalam penilaian KPI (*Key Performance Indicator*) yang dimiliki masing-masing karyawan setiap bulannya. Sebelumnya pencatatan presensi karyawan di kantor menggunakan mesin presensi konvensional. Tapi semenjak WFH pencatatan presensi karyawan menjadi tidak teratur dan kurang dapat dipertanggungjawabkan, dikarenakan perbedaan lokasi dan kurangnya pengawasan secara langsung oleh kantor. Sehingga tidak diketahui apakah benar karyawan sedang berada dirumah atau tidak. Hal tersebut berpengaruh pada kebijakan PT Akses Digital bahwa kegiatan WFH itu bukan hanya berkerja dari rumah tapi juga menjaga kesehatan karyawan itu sendiri agar terhindar dari Covid-19 yang dapat mengganggu kinerjanya saat melakukan pekerjaan kantor. Masalah utamanya yaitu bagaimana cara menerapkan pengawasan presensi karyawan dengan fasilitas yang sama seperti mesin fingerprint yang berada di kantor selama WFH dilakukan serta pengawasan lokasi karyawan juga diperlukan untuk memastikan bahwa karyawan benar berada dirumah saat jam kerja WFH.

Solusi yang diberikan dari penelitian ini adalah penggunaan Aplikasi Presensi Karyawan yang dapat memberikan data *check-in* dan *check-out* kehadiran karyawan tanpa terkendala perbedaan lokasi serta dapat mengawasi karyawan bahwa karyawan benar berada di lokasi *WFH* yaitu di rumahnya. Penggunaan perangkat aplikasi juga secara simultan berpengaruh positif dan signifikan terhadap disiplin pegawai [3]. Data tersebut langsung terkirim ke *server* dan dapat dilihat langsung oleh admin. Aplikasi ini dibangun dengan teknologi *web service* berbasis *Restful*, sehingga dapat melakukan pertukaran dengan dengan aplikasi *mobile* android.

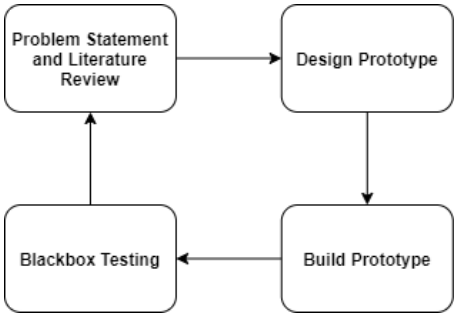
Web service adalah antarmuka yang menyediakan kumpulan operasi yang dapat diakses oleh jaringan internet [4]. *Web Service* digunakan untuk suatu fasilitas yang disediakan oleh suatu *website* untuk menyediakan layanan dalam bentuk informasi kepada sistem yang lainnya, sehingga sistem lain dapat berinteraksi dengan sistem tersebut yang melalui layanan-layanan (*service*). *Web Service* menyimpan data informasi dalam format standar seperti *HTTP*, *XML*, dan *JSON* [5].

Android adalah sistem operasi mobile yang saat ini paling populer. Tercatat per November 2020 pengguna Android di Indonesia sebesar 91.5% [6]. Dan pengguna Android di PT Akses Digital Indonesia sebesar 100%. Sehingga pengimplementasian aplikasi android ini sudah sangat tepat. Android juga menyediakan beragam fitur yang dapat mendukung dalam pengembangan aplikasi di penelitian ini, seperti fitur *GPS* dan *Fingerprint* [7].

Penelitian ini bertujuan untuk mengimplementasikan metode keamanan menggunakan *JSON Web Token* (JWT) pada *web service* di bagian autentikasi sistem. Studi ini juga mengimplementasikan fitur *geotagging*, *fingerprint*, dan *device checker* yang diharapkan dapat menjadi solusi untuk keamanan dan keefektifan data pada Aplikasi Presensi Karyawan di PT Akses Digital Indonesia dan semoga dapat menyempurnakan penelitian sebelumnya.

2. Metode Penelitian

Metode penelitian yang dilakukan pada studi ini menggunakan metode prototipe. Prototipe adalah versi awal dari sistem perangkat lunak yang digunakan untuk mendemonstrasikan konsep, mencoba opsi desain, dan mencari tahu lebih banyak tentang masalah dan kemungkinan solusinya. Pengembangan prototipe yang cepat dan berulang sangat penting agar biaya dapat dikendalikan dan pemangku kepentingan sistem dapat bereksperimen dengan prototipe di awal proses perangkat lunak [8]. Dengan 4 tahapan yaitu *Problem Statement and Literature Review* (Pernyataan Masalah dan Tinjauan Pustaka), *Design Prototype* (Desain Prototipe), *Build Prototype* (Membangun Prototipe), *Blackbox Testing* (Pengujian Blackbox).



Gambar 1. Tahap penelitian

2.1. *Problem Statement and Literature Review*

Tahap pertama, penulis menggali masalah dan solusi yang dibutuhkan untuk penelitian ini dengan cara mengumpulkan semua data karyawan dan data *resourse*

yang dimiliki oleh para karyawan di PT Akses Digital Indonesia.

Masalah yang ditemukan adalah PT Akses Digital Indonesia membutuhkan suatu alat bantu untuk mencatat presensi karyawan yang dapat digunakan selama *Work From Home* dengan fasilitas yang sama atau lebih dari mesin presensi konventional yang hanya bisa digunakan saat datang ke kantor. Penulis mendapatkan data bahwa 100% karyawan PT Akses Digital Indonesia menggunakan smartphone dengan basis android dan memiliki fitur sensor *fingerprint* dan GPS.

Dengan data dan informasi yang telah didapatkan sebelumnya. Maka solusi yang dapat diberikan adalah pembuatan Aplikasi Presensi Karyawan berbasis *web service* dengan *client* web dari sisi admin dan *client mobile* android dari sisi karyawan. Komunikasi data antar client akan menggunakan metode RESTful API dan *JSON Web Token* sebagai pengamannya.

Pada penelitian sebelumnya sudah banyak yang membangun *web service* aplikasi presensi dengan komunikasi RESTful API. Salah satunya adalah penelitian membahas algoritma *Point Clipping* pada aplikasi presensi karyawan [9]. Akan tetapi penelitian tersebut masih belum menerapkan keamanan dalam implementasi RESTful API. Sehingga kemungkinan adanya *request* yang dilakukan dari luar lingkungan *web service* dan tidak mempunyai ijin akan mudah masuk ke dalam *internal system* [10].

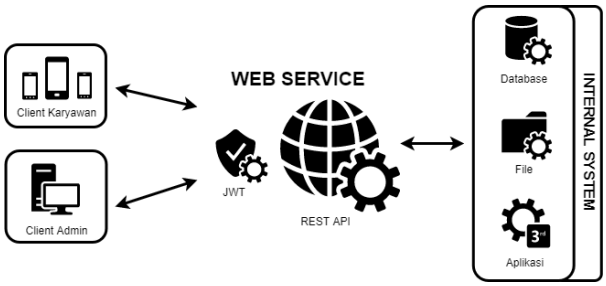
Di penelitian lainnya, penerapan *fingerprint* menggunakan android memang cukup efektif untuk memvalidasi kehadiran [11]. Akan tetapi setiap perangkat android dapat didaftarkan hingga 5 data *biometric fingerprint*. Sehingga kemungkinan *user* satu mengakses aplikasi *user* lainnya itu sangat besar.

Untuk pencegahan adanya kecurangan dalam melakukan presensi oleh karyawan, pada penilitian ini akan menerapkan 3 validasi, yaitu *fingerprint*, *geotagging*, dan *device checker*.

Validasi *fingerprint* digunakan untuk memastikan bahwa tidak sembarang *user* dapat melakukan *check in* dan *check out* presensi. Validasi *geotagging* digunakan untuk memastikan bahwa *user* berada dilingkungan *Work From Home*, yaitu dirumahnya. Validasi *device checker* digunakan untuk memastikan bahwa *user* hanya melakukan *check in* dan *check out* presensi melalui *device* miliki *user* itu sendiri.

2.2. *Design Prototype*

Tahap kedua, yaitu merancang Aplikasi Presensi dengan 2 client dan 1 internal sistem untuk bagian *backend*. Desain sistem akan terlihat seperti Gambar 2.



Gambar 2. Desain Aplikasi Karyawan

Client Karyawan akan dibangun dengan *platform* berbasis *Android* dengan menggunakan bahasa dart dan *framework* flutter. Di aplikasi ini karyawan akan melakukan *check in* dan *check out* presensi.

Client Admin akan dibangun dengan *platform* berbasis web dengan menggunakan bahasa *javascript* dan *framework* vue.js. Di aplikasi ini admin dapat mengawasi dan mengolah data karyawan dan presensi.

Bagian internal sistem akan dibangun dengan bahasa *javascript* dan *framework* Express.js. Di aplikasi ini akan mengimplementasikan *JSON Web Token*, berhubungan dengan database, dan menangani *request* dan *response* dari *client*. Komunikasinya akan menggunakan metode RESTful API.

2.3 *Build Prototype*

Tahap ketiga, dilakukan yaitu pengimplementasian RESTful *Web Service* dan *JSON Web Token* di Express JS, dimulai dengan memasang *jsonwebtoken package* dengan *Node Package Manager* (NPM) .

NPM Command
```
$ npm install jsonwebtoken
```

Proses selanjutnya adalah mendeklarasikan package di dalam file dimana request login diproses. Dalam penelitian ini *jsonwebtoken package* dideklarasikan di *authController.js* dengan menambahkan baris *code* seperti dibawah.

Declare JWT
```
const jwt = require("jsonwebtoken");
```

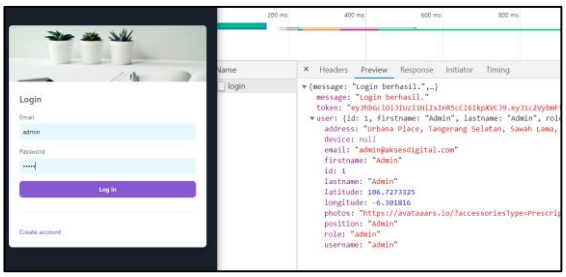
Menambahkan JWT KEY di file .env. JWT KEY ini akan digunakan saat membuat *token*.

JWT KEY
```
JWT_KEY=secret
```

Pemasangan *JSON Web Token* pada aplikasi telah selesai. Tahap terakhir adalah memberikan *token* kepada *user* setiap kali *user* melakukan *login*.

Declare JWT
```
const token = jwt.sign({username: user.username, role : user.role}, process.env.JWT_KEY, {algorithm: 'HS256});
```

Selanjutnya adalah pengujian *request post* dengan *route login* melalui client web admin. Dengan *input request username* dan *password* seperti pada Gambar 3.



Gambar 3. *Request dan Response Route Login*

Saat *request login* terkirim *server web service* akan melakukan pengecekan ke *database* dan memeriksa apakah *username* yang di-*input* ada. Jika tidak ditemukan maka *server web service* akan mengirimkan *response error* "User tidak ditemukan" dan *web service* tidak akan memberikan *token*. Tapi jika *username* ditemukan di *database*, selanjutnya *web service* akan mencocokan *password* yang dikirim dengan *username* miliki di database. Jika *password* cocok selanjutnya *server web service* akan mengirimkan *response* data *user* dan *token* seperti di gambar 3.

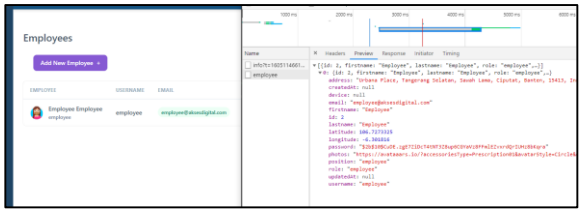
Response yang didapatkan pada gambar 3 sukses dengan *response code* 200, *response time* 136ms, dan ukuran 1.0kb. Selanjutnya token akan disimpan oleh client dan akan digunakan sebagai *claim* bahwa *user* tersebut dapat melakukan *request* lain. Setiap *user* ingin melakukan *request*, *user* tersebut juga harus mengirimkan token melalui *header request* dengan *key* "*Authorization*" dan value "*Bearer Token*".

Header Authorization yang dikirimkan oleh *user* setiap kali melakukan *request* selanjutnya akan diperiksa oleh *middleware* di *server web service*. *Web service* akan men-*verify token* yang dikirim dan akan dicocokan oleh data *user* di *database*. Jika data *user* tidak cocok maka *request* akan ditolak oleh *server web service* dan mendapatkan *response* "*Unauthorized Access API*".

```
Verify TOKEN
  jwt.verify(token, process.env.JWT_KEY);
```

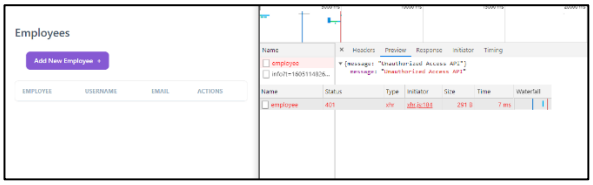
Pengujian selanjutnya yang akan dilakukan adalah melakukan *request* lain pada *route /employee* dengan menggunakan *token* dan tanpa menggunakan *token* untuk menguji keamanan dari *JSON WEB TOKEN*.

Pada Gambar 4 *request route /employee* mengunakan header *Authorization* dengan *Bearer Token* mendapatkan *response code 200*, *response message* "Sukses" dan data employee, *response time* 153ms, dan ukuran *response* sebesar 924b.



Gambar 4. *Request Get Employee* Dengan *Header Token*

Sedangkan pada Gambar 5 *request route employee* tanpa mengunakan header *Authorization* dengan *Bearer Token* mendapatkan *response code error* 401 *Unauthorized* dan *response message* "*Unanthorized Access API*".



Gambar 5. *Request Get Employee* Tanpa *Header Token*

Selanjutnya dilakukan Implementasi validasi pada aplikasi presensi saat melakukan *check in* dan *check out*. Dari sisi *internal system / backend* kita sudah memberikan keamanan dalam proses komunikasi data RESTful API dengan menggunakan *JSON Web Token*. Akan tetapi dari segi fungsionalnya kita perlu juga menjaga agar data yang diinput itu merupakan data yang jujur dan sah.

Aplikasi akan meminta sidik jari karyawan saat melakukan *check in* maupun *check out* presensi [12]. Saat sidik jari disentuhkan ke sensor *fingerprint* handphone, aplikasi akan mengambil data lokasi dan nama handphone karyawan, sehingga karyawan tidak dapat melakukan kecurangan saat berada di luar lokasi *WFH* atau menggunakan *device android* yang lain. Sehingga logikanya akan tergambar seperti dibawah.

```
CHECK IN AND CHECK OUT LOGIC
Input: location, deviceId
  Initialization user_deviceId, radius = 200

  if (fingerprint_success) {
      if (location <= radius) {
          if(deviceId == user_deviceId) {
                doCheckIn() || doCheckOut()
          } else {
                return print("gunakan device
                anda")
          }
      } else {
          return "anda harus berada dirumah"
      }
  } else {
    return print("presensi gagal")
  }
```

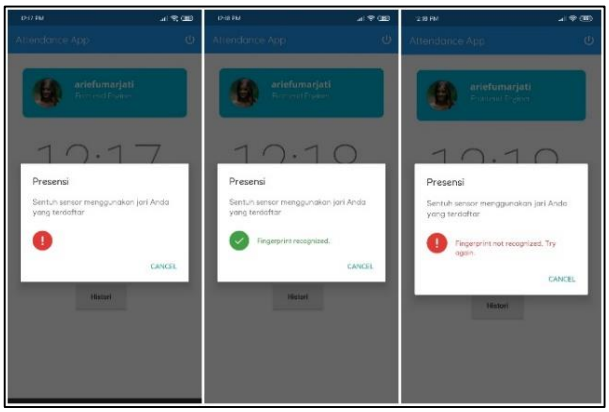
Aplikasi presensi sisi karyawan akan berbasis android dengan menggunakan bahasa pemrograman *dart* dan *framework Flutter*. Untuk dapat menggunakan 3 validasi yang disebutkan sebelumnya maka kita perlu memasang *package geolocator*, *fingerprint* dan *device info*.

```
ADD PACKAGE
  Dependencies:
  local_auth: ^0.6.3+2
  geolocator: ^6.0.0
  device_info: ^1.0.0
```

Sesuai *logic* yang tertulis sebelumnya, yang pertama perlu dilakukan adalah meminta user untuk melakukan

fingerprint. Jalankan *function* dari *package local_auth*. Setiap kali *user* menekan tombol *check in* atau *check out* akan muncul *pop up* untuk meminta *fingerprint* seperti Gambar 6.

```
GET POSITION
    Authenticated = await
    auth.authenticateWithBiometrics(
  localizedReason: 'Sentuh sensor menggunakan
  jari Anda yang terdaftar', useErrorDialogs:
  true, stickyAuth: false, androidAuthStrings:
  AndroidAuthMessages(signInTitle:
  Presensi"));
```



Gambar 6. *Pop Up Fingerprint* (kiri), *fingerprint* dikenali (tengah), *fingerprint* tidak dikenali (kanan)

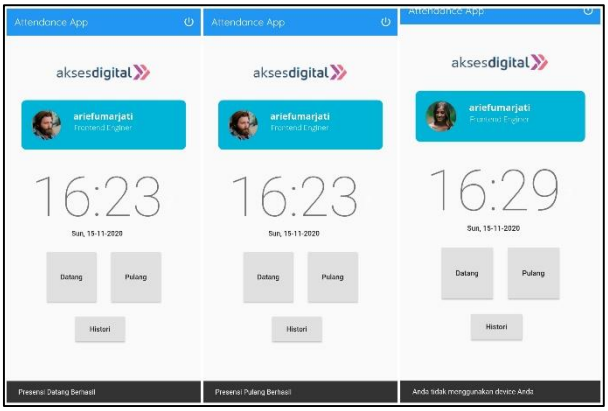
Selanjutnya jika *fingerprint* dikenali maka aplikasi akan mengakses lokasi *user* saat ini. Jalankan *function* dari *package geolocator* untuk mengambil data *latitude* dan *longitude*. Setelah itu kita harus mengukur jarak lokasi *user* saat ini dengan lokasi rumah *user*. Lokasi rumah *user* didapatkan dari data *user* di *database*. Jalankan *function* untuk mengukur jarak.

```
GET POSITION AND DISTANCES
  Position position = await
     getCurrentPosition(desiredAccuracy:
     LocationAccuracy.high);
  double distance =
     distanceBetween(home_latitude,
     home_longitude, position.latitude,
     position.longitude);
```

Jika lokasi sesuai dalam jangkauan radius rumah *user* maka aplikasi akan memindai data *device* dan mengambil data *AndroidId* untuk melakukan *request post check in* atau *check out* menggunakan *function package device_info*.

```
GET ANDROID ID
  DeviceInfoPlugin deviceInfo =
     DeviceInfoPlugin();
      AndroidDeviceInfo androidInfo = await
     deviceInfo.androidInfo;
```

Request yang terkirim ke *server web service* selanjutnya diolah dan akan dicek apakah menggunakan *device* yang sama dengan *device* yang terdaftar di data *user*. Jika data *device* yang dikirim sesuai maka presensi berhasil, sedangkan jika data *device* tidak sesuai maka presensi gagal dan akan memunculkan keterangan "Anda tidak menggunakan device Anda" seperti pada Gambar 7.



Gambar 7. Presensi Datang Berhasil (kiri), Presensi Pulang Berhasil (tengah), Presensi Gagal (kanan)

Selanjutnya *user* dapat melihat *history* presensi dengan menekan tombol "*History*" seperti pada Gambar 8.

Gambar 8. *History* Presensi

3. Hasil dan Pembahasan

Di tahap akhir ini adalah tahap dimana perlu dilakukannya pengujian dan evaluasi. Pengujian dan evaluasi ini perlu dilakukan untuk mengetahui bahwa

aspek fungsionalitas dari *web service* ini sudah memenuhi kebutuhan, keamanan, dan keefektifan data. Metode pengujian menggunakan metode *blackbox*. Pengujian melibatkan 11 karyawan di PT Akses Digital dengan lokasi rumah yang berbeda-beda. Selanjutnya hasil pengujian dijelaskan pada Tabel 1.

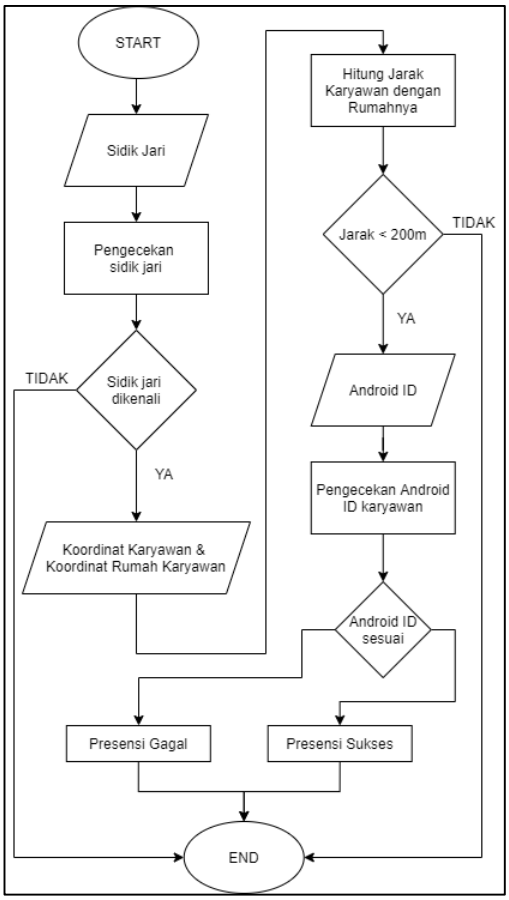
Tabel 1. Tabel proses uji *request dan response web service*

Services	*Route*	*Method*	*Platform*	*Time*
Register	/auth/register	POST	Web	300ms
Login	/auth/login	POST	Web	276ms
Edit Admin	/auth/admin/:id/edit	PUT	Web	167ms
Edit Employee	/employee/:id/edit	PUT	Web	72ms
List Employee	/employee	GET	Web	139ms
Delete Employee	/employee/delete/:id	DELETE	Web	80ms
Statistik	/configuration/dashboard	GET	Web	193ms
Configuration	/configuration	GET	Web	60ms
Edit Configuration	/configuration/edit	PUT	Web	79ms
List Attendance	/attendance	GET	Web	189ms
List employee Attendance	/attendance/:id	GET	Android	109ms
Presence	/attendance/:id/presence	POST	Android	157ms
			Rata-rata	**151.75ms**

Pada tabel 1 terdapat kolom dari nama *services*, *route*, *method*, *platform*, dan *time*. Dapat diketahui dari tabel tersebut bahwa seluruh layanan dari *webservice* berhasil diakses menggunakan *route* dan *method* yang tersedia. Serta dapat diakses dari *platform* yang berbeda. Dengan rata-rata kecepatan sebesar 151.75ms.

Selanjutnya adalah pengujian validasi. Pada gambar 9 adalah bentuk dari skenario pengujian validasi yang dilakukan. Validasi yang pertama dilakukan adalah validasi *fingerprint*, apabila sidik jari berhasil terbaca oleh sensor dan terdaftar maka sistem akan melakukan pengambilan data lokasi untuk memvalidasi lokasi, jika lokasi karyawan tidak lebih 200m selanjutnya sistem akan mengambil data Android ID milik karyawan. Lalu sistem akan membandingan Android ID milik karyawan dengan Android ID yang terdaftar pada database user. Jika Android ID sesuai maka karyawan berhasil melewati seluruh validasi dan presensinya sukses, sebaliknya jika tidak sesuai maka presensinya akan gagal.

Pada tabel 2 adalah hasil pengujian presensi dengan melewati validasi *fingerprint*, *location*, dan *device checker*. Keterangan untuk kolom U adalah *username*, FP adalah *fingerprint*, DL adalah *distance location*, dan Aid adalah androidId didapatkan dari *package device_info*. FP bernilai *true* apabila user berhasil melewati validasi *fingerprint* dengan cara menyentuhkan jari ke sensor *fingerprint*. DL didapatkan dengan membandingkan jarak antara lokasi *user* saat ini dengan lokasi rumah user yang sudah terdaftar di *database user*. DL terhitung sukses apabila lokasi karyawan berada pada radius toleransi yang diberikan PT Akses Digital Indonesia, yaitu sebesar 200m dari rumahnya. Aid akan bernilai *true* apabila *user* menggunakan *device* yang terdaftar di *database user*. Pada kolom Ket akan bernilai "sesuai" apabila kesalahan diakibatkan oleh kesalahan *user*, dan "tidak sesuai" apabila diakibatkan oleh kesalahan sistem aplikasi. Validasi FP, DL, dan Aid bersifat berurutan dan saling berkaitan. Jadi apabila salah satu validasi gagal maka validasi berikutnya juga ikut gagal.



Gambar 9. Skenario Pengujian Validasi

Tabel 2. Tabel Hasil Pengujian Validasi

U	FP	DL	Aid	Hasil	Ket
ariefumarj***	true	15m	true	sukses	Sesuai
allyapu***	true	20m	true	sukses	Sesuai
amandalaks***	true	116m	true	sukses	Sesuai
topazab**	true	150m	true	sukses	Sesuai
rivaldisu***	true	17m	true	sukses	Sesuai
fannyely***	false	-	-	gagal, tangan berminyak	Sesuai
fajara***	true	370m	-	gagal, lokasi	Tidak sesuai

				tidak akurat	
rezzaanug***	true	34m	true	sukses	Sesuai
ismunan***	true	55m	true	sukses	Sesuai
agungnue***	true	90m	true	sukses	Sesuai
dimasag***	true	9m	false	gagal, pakai device lain	Sesuai

Dari Tabel 2 dapat diketahui bahwa validasi menggunakan *fingerprint*, *geotagging*, dan *checking device* dapat menghasilkan data dengan akurasi sebesar 90,9%. Dengan 1 ketidaksesuaian data (dengan catatan kesalahan dari sistem) dan 10 data sesuai.

Tabel 3. Tabel Pengujian Fungsi Aplikasi

No	Nama Fungsi	Hasil
1	Aplikasi berhasil berjalan pada sistem web	Sukses
2	Aplikasi berhasil berjalan pada sistem android	Sukses
3	Aplikasi berhasil melakukan request ke web service	Sukses
4	Aplikasi berhasil mendapatkan response dari web service	Sukses
5	Aplikasi berhasil mengambil data *fingerprint*, lokasi, dan data *device*	Sukses

Pada Tabel 3 dijelaskan bahwa seluruh fungsi yang terdapat pada *web service* ini berjalan dengan baik dan 100% berhasil sukses.

4. Kesimpulan

Berdasarkan hasil dari pengujian dan evaluasi yang telah dilakukan, Aplikasi Presensi dengan menggunakan 3 validasi (*fingerprint, geotagging,* dan *device id*) ini sukses mengatasi masalah pengawasan kehadiran karyawan di PT Akses Digital Indonesia dengan tingkat kesuksesan sebesar 90.9%. Dengan fitur yang hampir sama atau lebih dibandingkan dengan mesin *fingerprint* konvensional seperti yang beda di kantor karena data presensi dapat langsung diawasi oleh admin secara *realtime* tanpa harus mengambil data secara manual dari mesin *fingerprint* konvensional. Implementasi *JSON Web Token* pada *web service* di aplikasi ini juga mampu memberikan *token* sebagai validasi bahwa *user* yang melakukan *login* dari aplikasi *client* web admin maupun aplikasi *client* android karyawan itu merupakan *user* yang sah dan dapat melakukan *request* data ke *server web service*.

Daftar Rujukan

[1] E. Ilham, "Membangun Sistem Pengelolaan Presensi Untuk Meningkatkan Kedisiplinan Pegawai (Studi Kasus : Pemda Sidoarjo)," *Seminar*, vol. 2007, no. Snati, pp. 1–5, 2007.

[2] A. Purwanto, "Studi eksplorasi Dampak WFH Terhadap Kinerja Guru," *J. Educ. Psychol. Couns.*, vol. 2, no. 1, pp. 92–100, 2020.

[3] R. Fadila and M. Septiana, "Pengaruh Penerapan Sistem Absensi Finger Print Terhadap Disiplin Pegawai Pada Markas Komando Direktorat Pengamanan Badan Pengusahaan Batam," *J. Appl. Bus. Adm.*, vol. 3, no. 1, pp. 53–63, 2019, doi: 10.30871/jaba.v3i1.1287.

[4] E. Edy, F. Ferdiansyah, W. Pramusinto, and S. Waluyo, "Pengamanan Restful API menggunakan JWT untuk Aplikasi Sales Order," *J. RESTI (Rekayasa Sist. dan Teknol. Informasi)*, vol. 3, no. 2, pp. 106–112, 2019, doi: 10.29207/resti.v3i2.860.

[5] K. Gottschalk, S. Graham, H. Kreger, and J. Snell, "Introduction to Web services architecture," *IBM Syst. J.*, vol. 41, no. 2, pp. 170–177, 2002, doi: 10.1147/sj.412.0170.

[6] "Mobile Operating System Market Share Indonesia." [Online]. Available: https://gs.statcounter.com/os-market-share/mobile/indonesia. [Accessed: 12-Nov-2020].

[7] H. N. Lengkong, A. A. E. Sinsuw, and A. S. . Lumenta, "Perancangan Penunjuk Rute Pada Kendaraan Pribadi Menggunakan Aplikasi Mobile GIS Berbasis Android Yang Terintegrasi Pada Google Maps," *E-journal Tek. Elektro dan Komput.*, vol. 2015, no. 2015, pp. 18–25, 2015.

[8] I. Sommerville, *Software Engineering*. 2013.

[9] A. Rahmatulloh, R. Rianto, and M. Q. Shihab, "Point Clipping Algorithm on Employee Presence Application for Geolocation of Employee Position," *Kinet. Game Technol. Inf. Syst. Comput. Network, Comput. Electron. Control*, vol. 4, no. 4, pp. 345–356, 2019, doi: 10.22219/kinetik.v4i4.796.

[10] R. C. Rajagukguk, "Penggunaan Kriptografi pada JWT (JSON Web Token) dalam Implementasi Keamanan API," 2018.

[11] A. Sumarudin, W. Permana, A. Suheryadi, K. Maulana, and N. Ibrahim, "Penerapan Sistem Absensi Sekolah Menggunakan Fingerprint Terintegrasi Dengan Smartphone Android," *J. Appl. Informatics Comput.*, vol. 3, no. 1, pp. 18–22, 2019, doi: 10.30871/jaic.v3i1.1051.

[12] N. A. Muhammad, "Pembuatan Aplikasi Presensi Perkuliahan Berbasis Fingerprint," *J. Tek. POMITS*, vol. 2, no. 3, pp. 465–469, 2013.