



Tersedia Online : <http://e-journals.unmul.ac.id/>

ADOPSI TEKNOLOGI DAN SISTEM INFORMASI (ATASI)

Alamat Jurnal : <http://e-journals2.unmul.ac.id/index.php/atasi/index>



Implementasi *Cron Job* pada Aplikasi Web untuk Otomasi Ekstraksi dan Distribusi Data Presensi Mesin *Fingerprint* Menggunakan REST API

Muhammad Erfan ^{1)*}, Rommi Kaestria ²⁾, Elok Faiqotul Himmah ³⁾

^{1,2)} Program Studi Sistem Informasi STMIK Palangkaraya

³⁾ Program Studi Teknik Informatika STMIK Palangkaraya

E-Mail : muhammad651@gmail.com ¹⁾; rokafordev@gmail.com ²⁾; el.faiqotul@gmail.com ³⁾;

ARTICLE INFO

Article history:

Received : 17/03/2026

Revised : 25/04/2026

Accepted : 04/05/2026

Available online : 31/08/2026

Keywords:

Cron jobs
Automation
Extraction
Fingerprint
REST API

Kata Kunci :

Cron job
Otomasi
Ekstraksi
Sidik jari
REST API

ABSTRACT

This research at STMIK Palangkaraya aims to develop an application that automates data extraction from the Solution X105 fingerprint machine and provides attendance data in JSON format through an API, enabling other applications that require attendance data to access it seamlessly. The study employs Design Science Research (DSR) combined with the Extreme Programming (XP) system development methodology. The application was built using the CodeIgniter version 4 web framework and a MySQL database. This web application automates the extraction of attendance data from the fingerprint machine via a cron job running twice every 24 hours with a 0% error rate, eliminating the need for the HR subdivision to perform manual data retrieval. The attendance data distribution workflow becomes significantly faster, as data can be accessed directly through an authenticated API. Seven JSON format options are available, allowing other units to easily integrate attendance data according to their specific needs.

ABSTRAK

Penelitian di STMIK Palangkaraya ini bertujuan untuk mengembangkan aplikasi otomasi penarikan (ekstraksi) data dari mesin *fingerprint* Solution X105 dan menyediakan data presensi dalam bentuk data JSON melalui API agar dapat digunakan oleh aplikasi lain yang membutuhkan data presensi. Penelitian ini merupakan *Design Science Research* (DSR) dipadukan dengan metode pengembangan sistem *Extreme Programming* (XP). Pengembangan aplikasi ini menggunakan *web framework* CodeIgniter versi 4 dan basis data MySQL. Aplikasi web ini mengotomatiskan ekstraksi data presensi dari mesin *fingerprint* menggunakan *cron job* dua kali dalam 24 jam dengan tingkat *error* 0%, subbagian SDM tidak perlu melakukan penarikan data manual. Alur distribusi data presensi menjadi lebih cepat karena data dapat diakses langsung melalui API terotentikasi. Tersedia tujuh pilihan format JSON, sehingga unit lain dapat dengan mudah mengintegrasikan data presensi sesuai kebutuhan.

2026 Adopsi Teknologi dan Sistem Informasi (ATASI) with CC BY SA license.

1. PENDAHULUAN

Sistem presensi memanfaatkan biometrik berupa sidik jari sudah banyak diadopsi oleh institusi pendidikan dan organisasi lainnya untuk meningkatkan keamanan, akurasi, dan efisiensi pencatatan kehadiran dibandingkan cara manual tanda tangan di atas kertas (Khan dkk., 2020; Memane dkk., 2022). Berbagai penelitian menunjukkan bahwa *fingerprint* mampu mengurangi kecurangan presensi, menekan kesalahan manusia, dan mempercepat proses pencatatan secara signifikan (Akinola dkk., 2021; Gaikar dkk., 2025; Roy dkk., 2025). STMIK Palangkaraya menggunakan mesin *fingerprint* Solution X105 sebagai pintu masuk data kehadiran dosen maupun tenaga kependidikan. Data yang terekam pada perangkat ini direkap oleh subbagian Sumber Daya Manusia (SDM) menggunakan aplikasi *Attendance Management System* (AMS) bawaan vendor secara berkala, baik setiap akhir bulan maupun saat laporan presensi dibutuhkan.

Meskipun sistem telah mendukung otomasi pencatatan kehadiran di level perangkat, proses rekapitulasi masih memiliki sejumlah keterbatasan. Masalah utama adalah pada proses penarikan (ekstraksi) data presensi dari mesin dan format rekap bawaan yang baku sehingga sulit disesuaikan dengan kebutuhan institusi. Penelitian sebelumnya

*) Corresponding Author

<https://doi.org/10.30872/atasi.v5i2.4513>

2026 Adopsi Teknologi dan Sistem Informasi (ATASI) with CC BY SA license.

pada sistem presensi menunjukkan bahwa fleksibilitas pengolahan dan integrasi data menjadi faktor kunci untuk mendukung analitik lanjutan dan pengambilan keputusan berbasis data (Akinola dkk., 2021; Gaikar dkk., 2025; Roy dkk., 2025). Karena itu dibutuhkan sebuah mekanisme penarikan dan ekstraksi data presensi dari perangkat *fingerprint* yang lebih adaptif dan dapat diintegrasikan dengan berbagai aplikasi internal.

Pemanfaatan *cron job* sebagai penjadwal tugas di sisi server telah terbukti efektif untuk mengotomasi proses ekstraksi, transformasi, dan pemindahan data secara berkala tanpa intervensi manual (Baron dkk., 2022; Wibowo dkk., 2025). Perkembangan teknologi *web service* dan REST API juga memungkinkan pertukaran data terstruktur dalam format *JavaScript Object Notation* (JSON) yang ringan dan mudah diolah oleh aplikasi web maupun sistem pihak ketiga. Di sisi lain, integrasi *cron job* dengan REST API dapat membangun alur kerja otomatis dengan cara data presensi dari mesin ditarik dan diekstraksi sesuai interval tertentu, kemudian didistribusikan dalam bentuk JSON melalui API. Pendekatan ini yang menjadi tujuan penelitian dan diharapkan dapat mengurangi beban kerja, meningkatkan konsistensi dan akurasi data presensi dengan tingkat *error* $\leq 0,5\%$, sekaligus menyediakan format data yang fleksibel untuk berbagai kebutuhan di lingkungan STMIK Palangkaraya.

2. TINJAUAN PUSAKA

A. Cron Job

Cron job digambarkan sebagai mekanisme penjadwalan tugas (*task scheduling*) pada sistem berbasis Unix yang dapat menjalankan perintah atau skrip secara otomatis pada interval waktu tertentu tanpa intervensi manual dari pengguna (Wibowo dkk., 2025). *Cron job* memungkinkan server web untuk melakukan tugas berulang dan spesifik pada waktu tertentu (Chigwada dkk., 2022). Karena itu *cron job* disebut sebagai cara yang sederhana namun kuat (*straightforward yet powerful*) untuk mengeksekusi tugas terjadwal sehingga efektif untuk tugas berulang seperti ekstraksi, transformasi, dan pemuatan data (Wibowo dkk., 2025).

B. REST API

REST (*Representational State Transfer*) adalah gaya arsitektur untuk merancang layanan web, sementara API (*Application Programming Interface*) mengikuti prinsip REST, sehingga memungkinkan sistem yang berbeda dapat saling berkomunikasi melalui protokol HTTP (Gowda & Gowda, 2024). REST API menggunakan JSON (*JavaScript Object Notation*) sebagai format standar yang baku untuk komunikasi data (Afriansyah dkk., 2023).

C. Penelitian Terdahulu

Penelitian tentang *cron job* yang berjudul “Cron Job Implementation for Automated Data Processing and Transfer in Cloud Infrastructure” menunjukkan bahwa penerapan *cron job* terbukti efektif dalam mengotomatisasi pemrosesan dan transfer data, sistem menjadi lebih efisien dan berjalan secara konsisten tanpa intervensi manual (Wibowo dkk., 2025). Adapun penelitian dengan judul “Perancangan Aplikasi Pemrograman Antarmuka Berbasis Web Menggunakan Gaya Arsitektur Representasi untuk Sistem Presensi Sekolah” menerapkan REST API dan format JSON sehingga memudahkan dalam komunikasi data (Afriansyah dkk., 2023).

Sementara penelitian lain berjudul “Implementasi Teknologi Application Programming Interface pada Perancangan Aplikasi Absensi Pegawai” menghasilkan sebuah aplikasi web untuk absensi berbasis teknologi API agar data absensi dapat terintegrasi dengan aplikasi lain (Hermansyah & Maryam, 2023). Penelitian lain berjudul “Desain dan Implementasi Aplikasi Absensi Kepegawaian Berbasis Web Menggunakan Framework Laravel di Diskominfo Provinsi Jawa Barat” menghasilkan sistem informasi optimalisasi pengolahan data absensi menggunakan REST API untuk integrasi data (Tanjung & Liptia, 2023).

3. METODE PENELITIAN

Metode yang digunakan adalah *Design Science Research* (DSR), yaitu sebuah paradigma pemecahan masalah yang berupaya meningkatkan pengetahuan manusia melalui penciptaan artefak inovatif. Artefak yang dihasilkan dari DSR ini sendiri dapat berupa konstruksi, model, metode, atau instansiasi (vom Brocke dkk., 2020). Penerapan metode ini dikolaborasikan dengan metode pengembangan sistem *Extreme Programming* (XP) yang merupakan salah satu pendekatan yang paling populer dalam pengembangan perangkat lunak secara cepat, karena menyediakan tahapan yang berlangsung dalam waktu singkat dan berorientasi pada tujuan pengembangan (Borman dkk., 2020). Tabel 1 menunjukkan *mapping* proses antara DSR dan XP yang dilakukan oleh peneliti.

Tabel 1. *Mapping* antara proses DSR dan tahapan XP.

No.	Proses DSR	Tahapan XP	Keterangan	Aktivitas
1	<i>Problem identification</i>	<i>Planning</i>	Pemahaman kebutuhan objek terhadap masalah yang akan diselesaikan	Mengidentifikasi kebutuhan, mendapatkan jumlah dan data presensi dari mesin, spesifikasi REST API
2	<i>Objectives</i>		Definisi tujuan fungsional dan kriteria kesesuaian artefak	
3	<i>Design and development</i>	<i>Design, Coding</i>	Pengembangan artefak secara iteratif dan deteksi dini kesalahan fungsional artefak	Merancang, membuat basis data, membuat diagram proses, antarmuka, membangun REST API dengan parameter penarikan data dan parameter permintaan data
4	<i>Demonstration</i>			
5	<i>Evaluation</i>	<i>Testing</i>	Uji fungsionalitas dan kesesuaian artefak	Menguji fungsi penarikan dan permintaan, perbandingan data mesin dan hasil penarikan
6	<i>Communication</i>		Refleksi dan perencanaan iterasi selanjutnya	

Sumber : Diolah oleh peneliti.

4. HASIL DAN PEMBAHASAN

A. Problem Identification

Praktik pengelolaan data presensi di STMIK Palangkaraya setidaknya memiliki dua permasalahan yang utama, yaitu:

- 1) Penarikan data presensi dari mesin setiap bulan sebelum penggajian menggunakan aplikasi AMS bawaan oleh subbagian SDM.
- 2) Tidak tersedia penyajian data presensi secara terbuka untuk beragam kebutuhan, hanya untuk data penggajian yang diunduh oleh sub bagian SDM melalui aplikasi AMS dalam format *spreadsheet* (.xls) dengan struktur dan data baku bawaan aplikasi.

B. Objectives

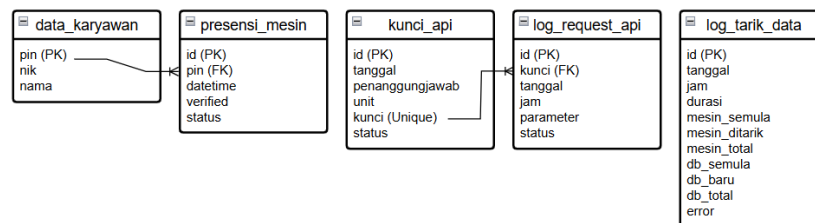
Tujuan penelitian ini untuk mengembangkan aplikasi yang dapat mengotomatisasi penarikan (ekstraksi) data dari mesin *fingerprint* Solution X105 dengan akurasi tingkat *error* $\leq 0,5\%$ dan menyediakan data presensi dalam bentuk data JSON melalui API dalam beberapa format data yang berbeda agar dapat digunakan oleh aplikasi lain yang membutuhkan data presensi.

C. Design and Development

Pengembangan aplikasi ini menggunakan bahasa PHP berbasis *web framework* CodeIgniter versi 4 dan basis data MySQL dengan antarmuka aplikasi untuk admin menggunakan Tailwind CSS. Arsitektur ini dipilih untuk memudahkan dalam proses pengembangan aplikasi.

1) Desain basis data

Tabel *presensi_mesin* digunakan untuk menyimpan hasil penarikan dan ekstraksi data presensi yang berasal dari mesin *fingerprint* melalui proses terjadwal menggunakan *cron job*. Setiap proses penarikan data dan *request* data presensi melalui *endpoint* dicatat pada tabel *log_penarikan_data* dan tabel *log_tarik_data*. Setiap request bisa diproses jika memiliki *key* yang valid dan tersedia di tabel *kunci_api*.



Gambar 1. Desain dan relasi tabel.

2) Desain parameter dan endpoint

Format *request* data mengikuti parameter sebagai berikut:

Tabel 2. Parameter *endpoint* untuk *request* data.

Tipe	Parameter	Keterangan/Value
Statis	nik	Untuk definisi NIK
	md	Mulai Dari, mendefinisikan tanggal awal
	sd	Sampai Dengan, mendefinisikan tanggal akhir
	dj	Durasi Jam, mendefinisikan permintaan data durasi
Dinamis	{kunci}	Kode kunci khusus untuk melakukan permintaan data (wajib ada dan wajib valid)
	{nik}	Berisi NIK
	{yyyy-mm-dd}	Format tanggal yang valid untuk parameter md dan sd
	{1}	1 = aktif, untuk permintaan data durasi

Format permintaan data mengikuti tujuh *endpoint* berikut:

- Mendapatkan daftar semua karyawan aktif
`api/{kunci}`
- Mendapatkan semua data presensi dari karyawan tertentu
`api/{kunci}?nik={nik}`
- Mendapatkan data presensi periodik dari karyawan tertentu
`api/{kunci}?nik={nik}&md={yyyy-mm-dd}&sd={yyyy-mm-dd}`
- Mendapatkan data presensi periodik dari semua karyawan
`api/{kunci}?md={yyyy-mm-dd}&sd={yyyy-mm-dd}`
- Mendapatkan semua durasi presensi per tanggal dari karyawan tertentu
`api/{kunci}?nik={nik}&dj={1}`
- Mendapatkan durasi presensi periodik per tanggal dari karyawan tertentu
`api/{kunci}?nik={nik}&md={yyyy-mm-dd}&sd={yyyy-mm-dd}&dj={1}`

- g) Mendapatkan semua durasi presensi periodik dari semua karyawan
`api/{kunci}?md={yyyy-mm-dd}&sd={yyyy-mm-dd}&dj={1}`

3) Pengaturan *cron job*

Cron job diatur melakukan penarikan data otomatis dua kali dalam 24 jam yaitu pada pukul 11.00 WIB dan pukul 23.00 WIB. *Cron job* mengeksekusi *command* TarikMesin untuk menjalankan *controller* TarikCliCron yang menarik data dengan sistem *batch* per 1000 data sampai dengan data presensi terbaca semua dari mesin. Catatan (*log*) proses penarikan data disimpan dalam *file* log-tarik-11-23.txt. Gambar 2 berikut menunjukkan baris pengaturan *cron job* penarikan data presensi dari mesin.

Minute	Hour	Day	Month	Weekday	Command
0	11,23	*	*	*	<pre>/usr/local/bin/php /root/.ssh//easy-api-v1/spark tarik:mesin >> /root/.ssh//log-tarik-11-23.txt 2>&1</pre>

Gambar 2. Pengaturan *cron job* untuk penarikan data presensi dari mesin.

Controller TarikCliCron membaca dan mengekstraksi data dari mesin *fingerprint* yang berformat XML kemudian menyimpan ke *database* dengan sistem batch. Gambar 3 menunjukkan sebagian kode penarikan data dan ekstraksi, sedangkan Gambar 4 menunjukkan kode penyimpanan data hasil ekstraksi.

```
$soap_request = "<?xml version='1.0' encoding='UTF-8'>>\r\n"
    . "<GetAttLog>\r\n"
    . "<ArgComKey xsi:type='xsd:integer'>{$Key}</ArgComKey>\r\n"
    . "<Arg><PIN xsi:type='xsd:string'>All</PIN></Arg>\r\n"
    . "</GetAttLog>";

$ch = curl_init($host);
curl_setopt_array($ch, [
    CURLOPT_POST => true,
    CURLOPT_RETURNTRANSFER => true,
    CURLOPT_POSTFIELDS => $soap_request,
    CURLOPT_HTTPHEADER => ["Content-Type: text/xml"],
    CURLOPT_SSL_VERIFYPEER => false,
    CURLOPT_SSL_VERIFYHOST => false,
    CURLOPT_TIMEOUT => 1800,
]);
$response = curl_exec($ch);

if (curl_errno($ch)) {
    echo "Curl Error: " . curl_error($ch);
    return;
}
curl_close($ch);

$dataXML = $this->parse_data($response, "<GetAttLogResponse>", "</GetAttLogResponse>");
$rows = explode("\r\n", $dataXML);
```

Gambar 3. Skrip pembaca dan ekstraksi data presensi format XML.

```
foreach ($rows as $row) {  
    $rowData = $this->parse_data($row, "<Row>", "</Row>");  
    if (!$rowData) continue;  
    $PIN = $this->parse_data($rowData, "<PIN>", "</PIN>");  
    $DateTime = $this->parse_data($rowData, "<DateTime>", "</DateTime>");  
    $Verified = $this->parse_data($rowData, "<Verified>", "</Verified>");  
    $Status = $this->parse_data($rowData, "<Status>", "</Status>");  
  
    $batch[] = [  
        'pin' => $PIN,  
        'datetime' => $DateTime,  
        'verified' => $Verified,  
        'status' => $Status  
    ];  
  
    if (count($batch) >= $batchSize) {  
        $db->reconnect();  
        $this->dml->dataInsertBatchIgnore($tabel, $batch);  
        $affected = $db->affectedRows();  
        $db_baru += $affected;  
        $batch = [];  
    }  
}
```

Gambar 4. Skrip penyimpanan data presensi ke *database*.

D. Demonstration

1) Log penarikan data

Setiap *cron job* berjalan, akan dibuat catatan (*log*) pada *file* log-tarik-11-23.txt berisi tanggal dan waktu penarikan disertai jumlah data terbaru yang berhasil ditarik dari mesin dan jumlah data terbaru yang berhasil disimpan ke *database*. Ditampilkan juga persentase *error* dari perbandingan data ditarik dan data disimpan. Gambar 5 memperlihatkan tangkapan isi *file log* tersebut.

```
CodeIgniter v4.6.3 Command Line Tool - Server Time: 2026-03-12 11:03:02 UTC+07:00  
✅ Penarikan data presensi selesai! # Data terbaru dari mesin: 18 # Disimpan ke DB: 18 # Persentase error: 0,00000 %  
  
CodeIgniter v4.6.3 Command Line Tool - Server Time: 2026-03-12 23:03:02 UTC+07:00  
✅ Penarikan data presensi selesai! # Data terbaru dari mesin: 0 # Disimpan ke DB: 0 # Persentase error: 0 %  
  
CodeIgniter v4.6.3 Command Line Tool - Server Time: 2026-03-13 11:03:02 UTC+07:00  
✅ Penarikan data presensi selesai! # Data terbaru dari mesin: 3 # Disimpan ke DB: 3 # Persentase error: 0,00000 %  
  
CodeIgniter v4.6.3 Command Line Tool - Server Time: 2026-03-13 23:03:02 UTC+07:00  
✅ Penarikan data presensi selesai! # Data terbaru dari mesin: 46 # Disimpan ke DB: 46 # Persentase error: 0,00000 %  
  
CodeIgniter v4.6.3 Command Line Tool - Server Time: 2026-03-14 11:03:02 UTC+07:00  
✅ Penarikan data presensi selesai! # Data terbaru dari mesin: 36 # Disimpan ke DB: 36 # Persentase error: 0,00000 %  
  
CodeIgniter v4.6.3 Command Line Tool - Server Time: 2026-03-14 23:03:02 UTC+07:00  
✅ Penarikan data presensi selesai! # Data terbaru dari mesin: 0 # Disimpan ke DB: 0 # Persentase error: 0 %  
  
CodeIgniter v4.6.3 Command Line Tool - Server Time: 2026-03-15 11:03:02 UTC+07:00  
✅ Penarikan data presensi selesai! # Data terbaru dari mesin: 0 # Disimpan ke DB: 0 # Persentase error: 0 %  
  
CodeIgniter v4.6.3 Command Line Tool - Server Time: 2026-03-15 23:03:02 UTC+07:00  
✅ Penarikan data presensi selesai! # Data terbaru dari mesin: 32 # Disimpan ke DB: 32 # Persentase error: 0,00000 %  
  
CodeIgniter v4.6.3 Command Line Tool - Server Time: 2026-03-16 11:03:02 UTC+07:00  
✅ Penarikan data presensi selesai! # Data terbaru dari mesin: 40 # Disimpan ke DB: 40 # Persentase error: 0,00000 %  
  
CodeIgniter v4.6.3 Command Line Tool - Server Time: 2026-03-16 23:03:01 UTC+07:00  
✅ Penarikan data presensi selesai! # Data terbaru dari mesin: 22 # Disimpan ke DB: 22 # Persentase error: 0,00000 %
```

Gambar 5. Log penarikan dan penyimpanan data presensi ke *database*.

2) Perbandingan jumlah data presensi

Halaman *log* tarik data presensi pada akun administrator menampilkan lebih detail penarikan data dan penyimpanannya serta tingkat *error* dalam proses ekstraksi data presensi, sebagaimana tampak pada Gambar 6.

Log Tarik Data Presensi Menampilkan 50 data terbaru

TANGGAL	JAM	DURASI PENARIKAN	DATA MESIN			DATA SISTEM			ERROR (%)
			DILEWATI	DITARIK	TOTAL	SEMULA	DISIMPAN	TOTAL	
17 Maret 2026	11:00:03	0 menit 42 detik	46.722	41	46.763	46.722	41	46.763	0.00000
16 Maret 2026	23:03:01	0 menit 42 detik	46.700	22	46.722	46.700	22	46.722	0.00000
16 Maret 2026	11:03:02	0 menit 43 detik	46.660	40	46.700	46.660	40	46.700	0.00000
15 Maret 2026	23:03:02	0 menit 42 detik	46.628	32	46.660	46.628	32	46.660	0.00000
15 Maret 2026	11:03:02	0 menit 03 detik	46.628	0	46.628	46.628	0	46.628	0.00000
14 Maret 2026	23:03:02	0 menit 03 detik	46.628	0	46.628	46.628	0	46.628	0.00000
14 Maret 2026	11:03:02	0 menit 41 detik	46.592	36	46.628	46.592	36	46.628	0.00000
13 Maret 2026	23:03:02	0 menit 43 detik	46.546	46	46.592	46.546	46	46.952	0.00000
13 Maret 2026	11:03:02	0 menit 43 detik	46.543	3	46.546	46.543	3	46.546	0.00000
12 Maret 2026	23:03:02	0 menit 03 detik	46.543	0	46.543	46.543	0	46.543	0.00000
12 Maret 2026	11:03:02	0 menit 41 detik	46.525	18	46.543	46.525	18	46.543	0.00000
11 Maret 2026	23:03:02	0 menit 41 detik	46.480	45	46.525	46.480	45	46.525	0.00000
11 Maret 2026	11:03:01	0 menit 42 detik	46.451	29	46.480	46.451	29	46.480	0.00000
10 Maret 2026	23:03:02	0 menit 41 detik	46.404	47	46.451	46.404	47	46.451	0.00000

Gambar 6. Halaman log tarik data presensi.

3) JSON hasil request melalui endpoint API

Berikut adalah contoh hasil request data presensi melalui endpoint API yang dipresentasikan pada browser menggunakan extension Talend API Tester:

a) Mendapatkan data presensi periodik dari karyawan tertentu

api/{kunci}?nik=198111272010107&md=2026-02-16&sd=2026-03-15

```
{
  status: "success",
  message: "Data presensi periode tertentu untuk karyawan tersebut berhasil diambil!",
  data: {
    nik: "198111272010107",
    nama: "Muhammad Erfan",
    presensi: [
      {
        id: "44729",
        nik: "198111272010107",
        nama: "Muhammad Erfan",
        tanggal: "2026-02-18",
        jam: "07:04:16",
        status: "M"
      },
      {
        id: "44770",
        nik: "198111272010107",
        nama: "Muhammad Erfan",
        tanggal: "2026-02-18",
        jam: "14:44:27",
        status: "P"
      },
      {id: "44835", nik: "198111272010107", nama: "Muhammad Erfan", tanggal: "2026-02-19", ...},
      {id: "44871", nik: "198111272010107", nama: "Muhammad Erfan", tanggal: "2026-02-19", ...}
    ]
  }
}
```

Gambar 7. Data JSON presensi periodik dari salah satu karyawan.

b) Mendapatkan semua durasi presensi periodik dari semua karyawan

api/{kunci}?md=2026-02-16&sd=2026-03-15&dj=1

```
{
  status: "success",
  message: "Data durasi presensi periode tertentu untuk semua karyawan berhasil diambil!",
  data: {
    durasi: [
      {
        nik: "199202062010001",
        nama: "A'an Fransisca",
        total_hari: "22",
        total_jam: "129.79",
        total_jam_diakui: "129.79"
      },
      {
        nik: "199006252014107",
        nama: "Abdi Dwi Nugraha",
        total_hari: "22",
        total_jam: "120.92",
        total_jam_diakui: "119.18"
      },
      {nik: "198505102021101", nama: "Abdul Hadi", total_hari: "19", total_jam: "86.61",...},
      {nik: "199010082011101", nama: "Ahmad Baidawi", total_hari: "26", total_jam: "211.69",...},
      {nik: "197206262014108", nama: "Ahmad Mujib", total_hari: "22", total_jam: "133.17",...},
      {nik: "199304092016006", nama: "Amaya Andri Damaini", total_hari: "15", total_jam: "55.45",...},
      {nik: "198510082010002", nama: "Arliyana", total_hari: "5", total_jam: "21.80",...}
    ]
  }
}
```

Gambar 8. Data JSON durasi presensi periodik dari semua karyawan.

E. Evaluation

Penarikan dan ekstraksi data presensi dari mesin menunjukkan keberhasilan dengan tingkat *error* 0%. Kapasitas penyimpanan mesin *fingerprint* Solution X105 adalah maksimal 200.000 data presensi. Kapasitas terisi berdasarkan hasil demonstrasi sudah memenuhi hampir seperempat dari kapasitas maksimal. Semakin besar jumlah data tersimpan dalam mesin akan berpengaruh pada kecepatan penarikan dan ekstraksi data karena mesin menyimpan data presensi dalam format XML yang tidak dapat difilter dan harus dibaca dari awal sampai akhir.

F. Communication

Proses penghapusan data presensi segera setelah selesai ditarik sesuai jadwalnya adalah solusi yang akan mempercepat dan mempersingkat durasi penarikan data. Namun langkah ini harus mendapatkan persetujuan dari unsur pimpinan karena berhubungan dengan data presensi asli sejak awal digunakan.

5. KESIMPULAN

Aplikasi web ini berhasil direalisasikan dengan memanfaatkan *cron job* untuk otomasi ekstraksi data presensi *fingerprint* dari mesin Solution X105. Pihak subbagian SDM tidak perlu melakukan penarikan data manual setiap bulan melalui aplikasi AMS. Catatan (*log*) menunjukkan keberhasilan penarikan data presensi dan ekstraksi dari mesin dengan tingkat *error* 0% yang berarti seluruh data presensi baru dapat disimpan dengan jumlah sama ke dalam *database*. Distribusi data presensi tersedia dan dapat langsung dilakukan secara *real-time* tanpa menunggu penyediaan data dari subbagian SDM. Aplikasi menyediakan tujuh format data presensi dalam bentuk JSON yang dapat diperoleh melalui API menggunakan kunci otentikasi. Unit lain dapat langsung memanfaatkan data presensi melalui API ini.

6. DAFTAR PUSTAKA

- Afrianyah, R., Sholeh, M., & Andayati, D. (2023). Perancangan Aplikasi Pemrograman Antarmuka Berbasis Web Menggunakan Gaya Arsitektur Representasi untuk Sistem Presensi Sekolah. *Jurnal SCRIPT*, 11(1), 33–42. <https://doi.org/10.34151/script.v11i1.4767>
- Akinola, O. A., Olopade, S. O., & Afolabi, A. S. (2021). Development of mobile and desktop applications for a fingerprint-based attendance management system. *Indonesian Journal of Electrical Engineering and Computer Science*, 24(1), 570–580. <https://doi.org/10.11591/ijeecs.v24.i1.pp570-580>
- Baron, D. D. P., Renaldi, F., & Santikarama, I. (2022, Juni 11). *Improving Webservice REST API Performances Using Design Patterns: A case in Healthcare Business Intelligence Systems*. 7th North American International Conference on Industrial Engineering and Operations Management. <https://doi.org/10.46254/NA07.20220458>
- Borman, R. I., Priandika, A. T., & Edison, A. R. (2020). Implementasi Metode Pengembangan Sistem Extreme Programming (XP) pada Aplikasi Investasi Peternakan. *JUSTIN (Jurnal Sistem dan Teknologi Informasi)*, 8(3), 272–277. <https://doi.org/10.26418/justin.v8i3.40273>
- Chigwada, J., Mazunga, F., Nyamhere, C., Mazheke, V., & Taruvunga, N. (2022). Remote Poultry Management System for Small to Medium Scale Producers using IoT. *Scientific African*, 18, e01398. <https://doi.org/10.1016/j.sciaf.2022.e01398>
- Gaikar, Mst. P., Patil, Mst. S., Bhoir, Mst. Y., & Kapde, Prof. K. (2025). Fingerprint Based Attendance System. *International Journal of Advanced Research in Science, Communication and Technology*, 696–703. <https://doi.org/10.48175/IJARSCT-24494>
- Gowda, P., & Gowda, A. N. (2024). Best Practices in REST API Design for Enhanced Scalability and Security. *Journal of Artificial Intelligence, Machine Learning and Data Science*, 2(1), 827–830. <https://doi.org/10.51219/JAIMLD/priyanka-gowda/202>

- Hermansyah, H. Y., & Maryam, M. (2023). Implementasi Teknologi Application Programming Interface pada Perancangan Aplikasi Absensi Pegawai. *JUPI (Jurnal Ilmiah Penelitian dan Pembelajaran Informatika)*, 8(3), 744–754. <https://doi.org/10.29100/jupi.v8i3.3890>
- Khan, N. I., Mustafina, S. N., Jhumu, F. F., Zobyer, A. H. M., Mahin, M. H., Tarek, Md. A. I., Rahman, R., & Islam, M. N. (2020). Towards Developing an Automated Attendance Management System using Fingerprint Sensor. *2020 Emerging Technology in Computing, Communication and Electronics (ETCCE)*, 1–6. <https://doi.org/10.1109/ETCCE51779.2020.9350882>
- Memane, R., Jadhav, P., Patil, J., Mathapati, S., & Pawar, A. (2022). Attendance Monitoring System Using Fingerprint Authentication. *2022 6th International Conference On Computing, Communication, Control And Automation (ICCUBEA)*, 1–6. <https://doi.org/10.1109/ICCUBEA54992.2022.10010791>
- Roy, D. G., Bhattacharjee, A., Das, R., & Das, P. (2025). Trigger Based Smart Attendance Framework with Machine Learning for Predictive Student Performance Analysis. *International Journal of Research and Review*, 12(5), 321–330. <https://doi.org/10.52403/ijrr.20250535>
- Tanjung, F. R., & Liptia, V. (2023). Desain dan Implementasi Aplikasi Absensi Kepegawaian Berbasis Web Menggunakan Framework Laravel di Diskominfo Provinsi Jawa Barat. *JURNAL MEDIA INFOTAMA*, 19(1), 111–118. <https://doi.org/10.37676/jmi.v19i1.3485>
- vom Brocke, J., Hevner, A., & Maedche, A. (2020). Introduction to Design Science Research. Dalam J. vom Brocke, A. Hevner, & A. Maedche (Ed.), *Design Science Research. Cases* (hlm. 1–13). Springer International Publishing. https://doi.org/10.1007/978-3-030-46781-4_1
- Wibowo, A., Fathimah, A., Waskito, D., & Aprilianto, R. A. (2025). Cron Job Implementation for Automated Data Processing and Transfer in Cloud Infrastructure. *Saintekno : Jurnal Sains Dan Teknologi*, 23(1), 11–20. <https://doi.org/10.15294/saintekno.v23i1.22649>