

Pengembangan Aplikasi Absensi dan Cuti Pegawai Baharkam Polri dengan Stack MERN dan Metode Agile

Roeslan Djutalov^{1*}, Agus Pangondian Silalahi², Dzikri Fauzi Ramdhani³, Rizqi Murtadho⁴

¹⁻⁴Fakultas Ilmu Komputer, Teknik Informatika, Universitas Pamulang, Jl. Raya Puspiptek No. 46, Kel. Buaran, Kec. Serpong, Kota Tangerang Selatan. Banten 15310, Indonesia

Email: ^{1*}djutalovroeslan@gmail.com, ²agus pangondian04@gmail.com,
³dzikrifauziramdhani16@gmail.com, ⁴Rizqimurtadho518@gmail.com

(* : coressponding author)

Abstrak - Pengelolaan absensi dan cuti pegawai yang efisien sangat penting bagi organisasi besar seperti Baharkam Polri. Penelitian ini bertujuan mengembangkan aplikasi web menggunakan teknologi MERN (MongoDB, Express.js, React.js, Node.js) untuk menggantikan sistem manual yang lamban dan rentan kesalahan. Metode Agile diterapkan untuk memastikan pengembangan yang adaptif dan sesuai kebutuhan. Aplikasi ini meningkatkan efisiensi pencatatan kehadiran dan pengajuan cuti, mengurangi kesalahan, serta menyediakan data pegawai real-time yang valid. Hasilnya mendukung pengambilan keputusan berbasis data secara cepat dan akurat. Studi ini berkontribusi pada penerapan teknologi modern dalam tata kelola sumber daya manusia di institusi kepolisian.

Kata Kunci: Absensi, Cuti Pegawai, MERN Stack, Agile, Teknologi Informasi

***Abstract** - Efficient management of employee attendance and leave was essential for large organizations such as Baharkam Polri. This study aimed to develop a web-based application using the MERN technology stack (MongoDB, Express.js, React.js, Node.js) to replace manual systems that were slow and prone to errors. The Agile methodology was applied to ensure adaptive development that met user needs. The application improved the efficiency of attendance recording and leave submissions, minimized errors, and provided real-time, accurate employee data. The results supported faster and more accurate data-driven decision-making. This study contributed to the adoption of modern information technology in enhancing human resource management within law enforcement institutions.*

Keywords: Attendance, Employee Leave, MERN Stack, Agile, Information Technology.

1. PENDAHULUAN

Badan Pemelihara Keamanan Polri (Baharkam Polri) adalah satuan kerja Kepolisian yang menyelenggarakan fungsi pemeliharaan dan peningkatan kondisi keamanan serta ketertiban masyarakat. Dalam menjalankan tugas sehari-hari, pengelolaan absensi, izin, dan cuti pegawai menjadi aspek yang sangat penting untuk mendukung operasional organisasi secara efisien dan tepat waktu. Namun, hingga saat ini, proses pengelolaan absensi dan cuti di Baharkam Polri masih menggunakan sistem manual. Sistem ini melibatkan pencatatan fisik kehadiran pegawai melalui apel pagi dan dikompilasi secara manual oleh staf administrasi. Proses tersebut sering kali memakan waktu yang lama, rentan terhadap kesalahan pencatatan, dan tidak memberikan data yang real-time kepada pimpinan untuk pengambilan keputusan strategis. Selain itu, prosedur pengajuan izin dan cuti yang berbasis dokumen fisik menambah kompleksitas dan inefisiensi dalam pengelolaan sumber daya manusia di Baharkam Polri

Sebuah organisasi menjadikan Sumber Daya Manusia (SDM) sebagai bagian yang penting untuk mencapai perkembangan yang signifikan. Perkembangan dan pertumbuhan sebuah organisasi tidak lepas dari kinerja para pegawai yang berkualitas (Pertiwi, Luchia, & Sinta, 2023). Presensi atau kehadiran merupakan salah satu hal penting dari penilaian didalam sebuah kelompok organisasi.

Perkembangan teknologi informasi telah berkembang dalam beberapa dekade terakhir, mengubah cara berkomunikasi, bekerja dan mengelola informasi, Sistem informasi modern tidak hanya menawarkan kecepatan dan ketepatan, tetapi juga fleksibilitas dalam penyesuaian terhadap kebutuhan organisasi yang terus berubah. (Maulidah, 2022). Oleh karena itu, pengembangan aplikasi pengelolaan absensi dan cuti pegawai berbasis web menggunakan stack MERN pada instansi Baharkam Polri dengan

metode pengembangan agile sebagai solusi atas permasalahan yang terjadi di Instansi Baharkam Polri. Penelitian ini bertujuan untuk mengembangkan sebuah aplikasi yang berguna untuk mengatur pengelolaan absensi, cuti dan izin para pegawai berbasis web menggunakan stack MERN dengan metode pengembangan Agile, dengan tujuan untuk meningkatkan efisiensi dan responsivitas pengelolaan absensi dan cuti pegawai (Rezki Triandy, Nurudin Santoso, 2020). Melalui pemanfaatan metode pengembangan Agile, diharapkan aplikasi dapat dikembangkan dengan fokus pada prioritas tugas yang jelas, penyelesaian yang cepat terhadap perubahan kebutuhan, dan optimalisasi aliran kerja pengembangan web (Nova, Widodo, & Warsito, 2022). Dengan adanya aplikasi ini, diharapkan staf admin pimpinan dapat mengelola data absensi, cuti dan izin para pegawai dengan lebih efisien, mengurangi kesalahan pencatatan, dan meningkatkan responsivitas.

2. METODE PENELITIAN

2.1 Teknik Pengumpulan Data

Teknik pengumpulan data dan mengolah Informasi yang diperlukan menggunakan beberapa metode sebagai berikut:

a. Observasi

Melakukan pengamatan secara langsung proses absensi di Baharkam Polri selama beberapa hari kerja. Observasi akan mencakup pengumpulan absensi, rekapitulasi manual, dan pencatatan izin atau cuti pegawai. Data yang diobservasi akan dianalisis untuk melihat potensi perbaikan dengan sistem berbasis aplikasi.

Tabel 1. Rekapitulasi absensi staf Rorenmin Baharkam Polri

HARI: SENIN TANGGAL: 2 SEPTEMBER 2024										
SUBSATKER	Jumlah			KETERANGAN						
	JUMLAH	KURANG	HADIR	SAKIT	IZIN	CUTI	DIK	DINAS	LD	TK
RORENMIN	56	12	44		1		3	6	2	-
DAFTAR PERSONEL YANG TIDAK APEL:										
1. SAKIT										
NO	NAMA		PANGKAT	SUBSATKER	KET					
1.	-		-	-	-					
2. IZIN										
NO	NAMA		PANGKAT	SUBSATKER	KET					
1.	RITA UTAMI		PENDA	BAG REN	-					
3. CUTI										
NO	NAMA		PANGKAT	SUBSATKER	KET					
1.	-		-	-	-					
4. DIK										
NO	NAMA		PANGKAT	SUBSATKER	KET					
1.	BESTIANA		KOMPOL	BAG BINFUNG	-					
2.	YANTO		BRIPKA	BAGSUMDA	DIK SIP					
3.	WIBOWO		BRIGADIR	BAGSUMDA	-					
5. DINAS										
NO	NAMA		PANGKAT	SUBSATKER	KET					
1.	DEWA		KOMBESPOL	BAGBINFUNG	-					
2.	SULAIMAN		KOMPOL	BAG BINFUNG	-					
3.	PRADANA		BRIPDA	URTU ROENMIN	-					
4.	CHAIRI		BRIPDA	BAG SUMDA	-					
5.	RUDY		BHARATU	URTU ROENMIN	-					
6.	MELISTYO		PENGATUR	BAG BINFUNG	BKO BARESKRIM					
7.	-		-	-	-					
6. LEPAS DINAS										
NO	NAMA		PANGKAT	SUBSATKER	KET					
1.	RICKO		BRIPDA	BAG SUMDA	-					
2.	RIZKY		BRIPDA	BAG REN	-					
7. TANPA KETERANGAN										
NO	NAMA		PANGKAT	SUBSATKER	KET					
1.	-		-	-	-					

b. Wawancara

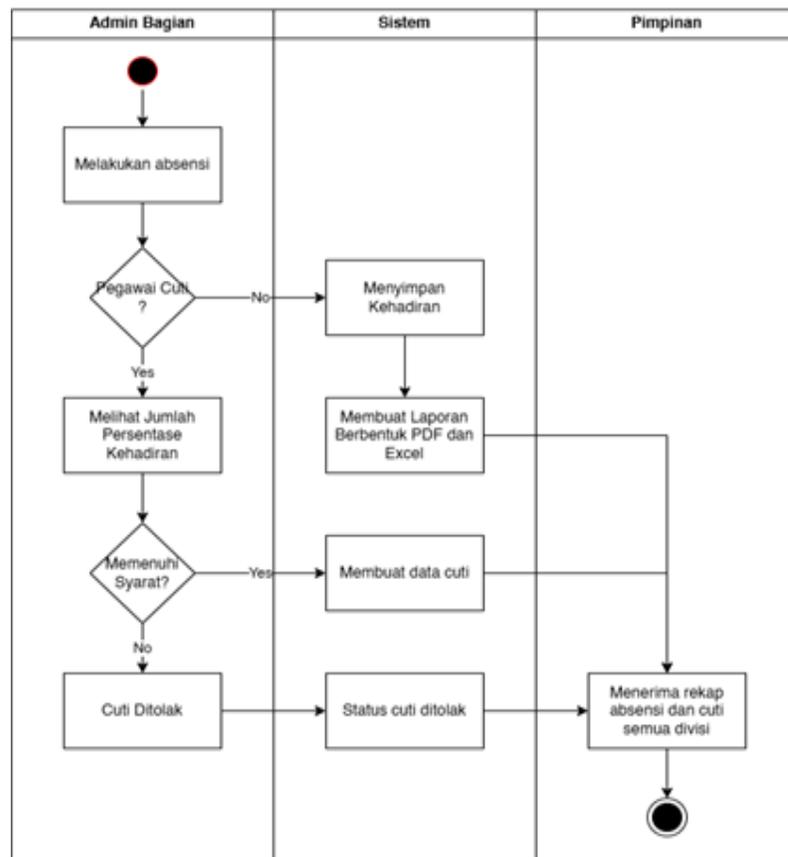
Wawancara dilakukan untuk mengetahui masalah yang timbul atau dialami langsung oleh Pegawai di Instansi BAHARKAM POLRI. Dalam kegiatan ini diajukan pertanyaan lisan dalam usaha untuk melengkapi data-data yang akan diperoleh

c. Studi Pustaka

Melakukan studi pustaka dari berbagai buku dan artikel yang menjadi referensi dalam penulisan, berbagai artikel jurnal penelitian tentang absensi dan referensi-referensi lainnya yang berkaitan dengan penyusunan laporan dan sebagai landasan teori untuk menyelesaikan masalah yang dihadapi.

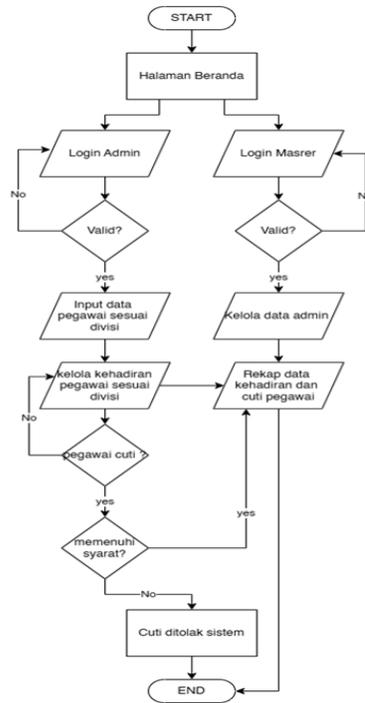
2.2 Perancangan Sistem

Dalam pengembangan suatu aplikasi, dibutuhkannya sebuah perancangan sistem yang berguna untuk memberikan gambaran yang jelas mengenai alur kerja, struktur dan komponen yang dibutuhkan. Perancangan sistem memastikan untuk meminimalisir kesalahan dan memudahkan dalam pengembangan. Dalam perancangan sistem terdapat elemen penting yakni aktivitas sistem, alur sistem dan juga ERD.



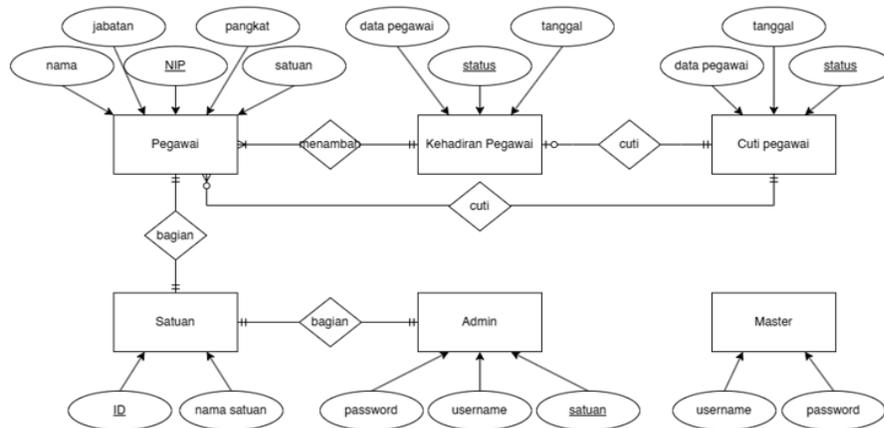
Gambar 1. Aktivitas Sistem

Aktivitas sistem menjelaskan sebuah dalam sistem untuk melakukan proses absensi dan cuti yang dilakukan oleh Admin bagian atau divisi yang nantinya akan direkap oleh sistem dan akan ditampilkan ke Pimpinan.



Gambar 2. Flowchart

Alur sistem atau *flowchart* menjelaskan proses terjadinya dan penggunaan sistem secara keseluruhan. Terdapat dua *role* penting dalam kegiatan. Admin akan melakukan pengelolaan data pegawai dan juga kehadiran pegawai sesuai dengan divisinya masing-masing. Selanjutnya semua data akan direkap oleh sistem yang dimana mengumpulkan semua data-data dari semua divisi yang nantinya dalam berbentuk laporan kehadiran dan cuti pegawai.



Gambar 3. Entity Relationship Diagram (ERD)

ERD menggambarkan kebutuhan dan juga hubungan antar *collection* atau tabel pada suatu database. Pada sistem yang dibutuhkan terdapat *collection* satuan, pegawai, admin, master kehadiran, dan cuti. Hubungan antar *collection* diwakilkan dengan satu garis yang menghubungkan antara kedua *collection*.

3. ANALISA DAN PEMBAHASAN

3.1 Implementasi pada Server dan Database Menggunakan ExpressJS dan MongoDB

Dalam pengimplementasian pada server, menggunakan ExpressJS dalam pembuatan *model view controller*(MVC). Dalam penggunaan database sendiri menggunakan MongoDB dengan ORM menggunakan Mongoose dalam konfigurasi koneksi dan pengelolannya. Server dibuat secara dinamik menggunakan *dotenv*() sebagai konfigurasi *environment variabelnya*. Hal ini untuk keamanan data yang lebih ketat dalam pengembangan.

```
const connectDb = async () => {
  try {
    const conn = await mongoose.connect(process.env.MONGO_URI, {
      useNewUrlParser: true,
      useUnifiedTopology: true,
    });
    console.log(`MongoDB Connected: ${conn.connection.host}`);
  } catch (error) {
    console.error("Error db config: ", error);
    process.exit(1);
  }
};
```

Gambar 4. Konfigurasi Database Pada Server

Pada server, sangat membutuhkan *routing API* untuk melakukan *fetching* data yang kedepannya akan dibutuhkan. *Routing API* yang buat dalam sistem ini mencakup untuk semua fitur yang nantinya akan dibutuhkan.

```
const express = require('express');
const app = express();
app.use(express.json());
app.use('/api/leave', require('./routes/leaveRoute.js'));
app.use('/api/employee', require('./routes/employeeRoute.js'));
app.use('/api/division', require('./routes/divisionRoute.js'));
app.use('/api/master', require('./routes/masterRoute.js'));
app.use('/api/admin', require('./routes/adminRoute.js'));
app.use('/api/attendance', require('./routes/attendanceRoute.js'));
app.use('/auth/admin', authRoutes);
```

Gambar 5. Routing API

Sebelum melakukan *controlling* pada API, dilakukan terlebih dahulu pembuatan model database dengan Mongoose, hal ini dilakukan agar tipe data yang akan dikirimkan ke database dan juga *middleware* pada server dapat diimplementasikan secara baik juga mengurangi kesalahan internal untuk kedepannya. Terdapat 6 model database yang digunakan, diantaranya : 1) adminModel; 2) attendanceModel; 3) divisionModel; 4) employeeModel; 5) leaveModel; dan 6) masterModel;. Dalam pembuatan model terdapat skema yang nantinya akan digunakan pada saat *controlling* data.

```
const AttendanceSchema = new mongoose.Schema({
  employee_id: {
    type: mongoose.Schema.Types.ObjectId,
    ref: 'Employee',
    required: true,
  },
  status: {
    type: String,
    enum: ['Present', 'Sick', 'Leave', 'Alpha'],
    required: true,
  },
  date: {
    type: Date,
    required: true,
  },
});
```

Gambar 6. Skema attendace Model

Dalam *controlling* data, terdapat beberapa *method* HTTP yang dipakai diantaranya adalah 1) GET; 2) POST; 3) PUT; 4) DELETE. Setiap *method* memiliki penanganan yang berbeda beda.

```
const getAllEmployees = async (req, res) => {
  try {
    const employees = await Employee.find().populate('employee_division');
    res.status(200).json(employees);
  } catch (error) {
    res.status(500).json({ message: error.message });
  }
};
```

Gambar 7. Method GET pada employee API

```
const createEmployee = async (req, res) => {
  try {
    const {
      employee_nrp,
      employee_name,
      employee_rank,
      employee_position,
      employee_division
    } = req.body;

    const newEmployee = new Employee({
      employee_nrp,
      employee_name,
      employee_rank,
      employee_position,
      employee_division
    });

    const employee = await Employee.create(newEmployee);

    res.status(201).json({ message: "Employee created successfully", data: employee });
  } catch (error) {
    res.status(400).json({ message: error.message });
  }
};
```

Gambar 8. Method POST pada employee API

```
const updateEmployee = async (req, res) => {
  try {
    const employee = await Employee.findByIdAndUpdate(req.params.id, req.body, {
      new: true,
      runValidators: true,
    });
  } catch (error) {
    return res.status(404).json({ message: "Employee not found" });
  }
  if (!employee) {
    return res.status(404).json({ message: "Employee not found" });
  }
  res.status(200).json(employee);
} catch (error) {
  res.status(500).json({ message: error.message });
};
```

Gambar 9. Method PUT pada employee API

```
const deleteEmployee = async (req, res) => {
  try {
    const employee = await Employee.findByIdAndDelete(req.params.id);
    if (!employee) {
      return res.status(404).json({ message: "Employee not found" });
    }
    res.status(200).json({ message: "Employee deleted successfully" });
  } catch (error) {
    res.status(500).json({ message: error.message });
  }
};
```

Gambar 10. Method DELETE pada employee API

3.2 Implementasi pada Aplikasi Menggunakan ReactJS

Pada implementasi aplikasi berbasis web, menggunakan ReactJS dalam mengelola tampilan. React menggunakan struktur komponen yang membuat pengembang melakukan *slicing* pada tampilan. Dalam melakukan *consume API* pada web, digunakan *library* axios untuk melakukan *fetching* data dan dikelola menggunakan *Hooks* pada ReactJS untuk melakukan *lifecycle* pada komponen.

```
useEffect(() => {
  const fetchEmployees = async () => {
    try {
      const response = await axios.get(
        `${import.meta.env.VITE_BACKEND_PORT}/api/employee`
      );
      const filteredEmployees = response.data.filter(
        (employee) =>
          employee.employee_division?.division_name === divisionName
      );

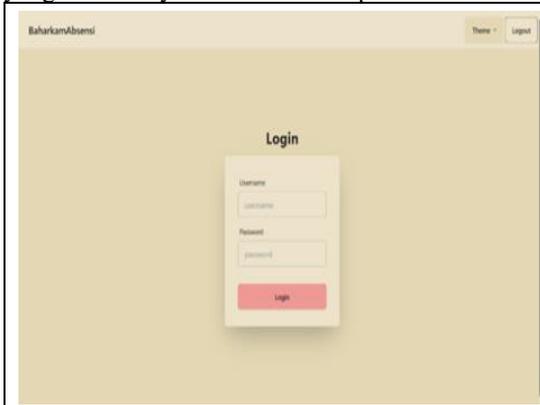
      setEmployees(filteredEmployees);
    } catch (err) {
      setError(err.message);
    }
  };
  fetchEmployees();
}, [divisionName]);
```

Gambar 11. Implementasi Lifecycle dengan React Hooks dan Axios

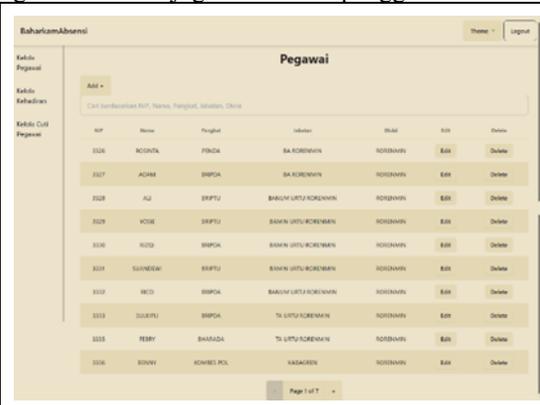
```
const fetchAttendanceData = async (employeeId, year, month) => {
  try {
    const response = await axios.get(
      `${
        import.meta.env.VITE_BACKEND_PORT
      }/api/attendance/${employeeId}/${year}/${month}`
    );
    const attendanceMap = {};
    response.data.forEach((attendance) => {
      const day = new Date(attendance.date).getDate();
      attendanceMap[day] = attendance.status;
    });
    setAttendanceData((prevState) => ({
      ...prevState,
      [employeeId]: attendanceMap,
    }));
  } catch (err) {
    setError(err.message);
  }
};
```

Gambar 12. Mengambil Data Kehadiran

Dalam pengelolaan tampilan, menggunakan *styling component* dengan DaisyUI untuk mempercepat pengembangan aplikasi. Penggunaan DaisyUI membuat pengguna dapat memilih tema yang lebih banyak macam dan dapat disesuaikan dengan selera dan juga kebutuhan pengguna.



Gambar 13. Tampilan Halaman *Login*



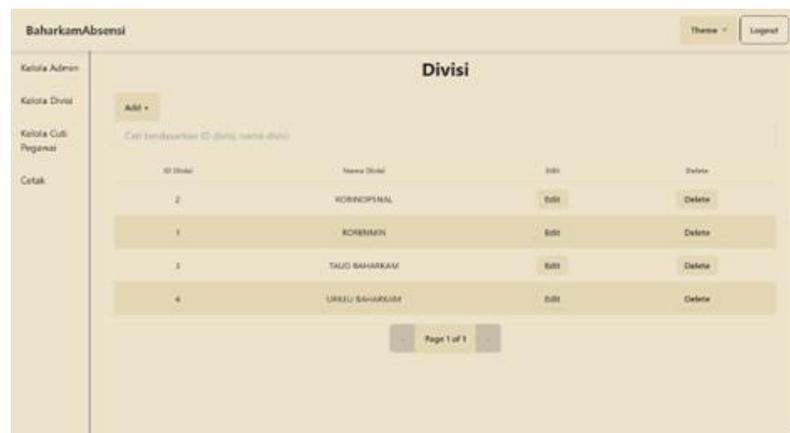
Gambar 14. Halaman Kelola Pegawai



Gambar 15. Halaman Kelola Kehadiran



Gambar 16. Halaman Data Cuti



Gambar 17. Halaman Kelola Divisi

Dalam melakukan pencetakan dan perekapan data dilakukan dengan menggunakan *library* dari JavaScript yakni *jspdf* untuk mengubah tabel menjadi pdf dan *xlsx* untuk mengubahnya ke dalam microsoft excel.

Laporan Kehadiran

Bulan: November 2024

Tanggal Cetak: 28 November 2024

Nama	Jumlah Hadir	Jumlah Sakit	Jumlah Izin	Jumlah Alpha	Persentase Kehadiran
ROSINTA	7	0	0	23	23%
ADAM	7	0	0	23	23%
ALI	0	0	0	30	0%
YOSIE	0	0	0	30	0%
RIZQI	0	0	0	30	0%
SUANDEWI	0	0	0	30	0%
RICO	0	0	0	30	0%
SULKIPLI	0	0	0	30	0%
FEBRY	0	0	0	30	0%
BENNY	0	0	0	30	0%
SYUHADA	0	0	0	30	0%
WIDYANA	0	0	0	30	0%
RITA	0	0	0	30	0%
DODY	0	0	0	30	0%

Gambar 18. Cetak Kehadiran Pegawai

3.3 Pengujian dan Evaluasi

Pengujian aplikasi dibagi menjadi dua jenis yakni uji *API* dengan POSTMAN dan juga fungsional aplikasi dengan menggunakan metode *Blackbox* dan *Whitebox*. Pengujian *API* melakukan semua *method HTTP* yang akan dikirimkan ke server. Dan pengujian fungsional secara keseluruhan aplikasi.

Tabel 2. Pengujian API

Route	Method	Deskripsi	Response	Status
/api/admin	GET, POST, PUT, DELETE	Kelola data Admin	200	OK
/api/master	GET	Mendapatkan data Master	200	OK
/api/employee	GET, POST, PUT, DELETE	Kelola data Pegawai	200	OK
/api/attendance	GET, POST, PUT	Kelola data kehadiran	200	OK
/api/leave	GET, POST, PUT, DELETE	Kelola cuti	200	OK
/api/division	GET, POST, PUT, DELETE	Kelola Satuan/Divisi	200	OK

Pengujian metode Blackbox pada Kelola Kehadiran Pegawai:

Tabel 3. Pengujian Metode *Blackbox* pada Absensi

No	Type	Test Case	Input	Expected result	Status
1	Positif	Button Add attendance	Klik Button Add attendance	Berhasil menggunakan fungsi button Add attendance	PASS
				Manampilkan halaman Baharkam Absensi	PASS

2	Positif	Memilih keterangan di Checkbox Kehadiran	Klik simbol panah bawah di button Checkbox	Berhasil menggunakan fungsi simbol	PASS
			Pilih item : “Hadir”	Tampil keterangan Hadir	PASS
			Pilih item :: “Cuti”	Tampil keterangan Cuti	PASS
			Pilih item :: “Sakit”	Tampil keterangan Sakit	PASS
			Pilih item :: “Alpha”	Tampil keterangan Alpha	PASS
			Pilih item :: “Libur”	Tampil keterangan Libur	PASS
3	Positif	Button Perbarui	Pilih ite Hadir di Checkbox disamping nama Budi. Klik button “Perbarui”	Berhasil memperbarui data absensi pegawai Budi	PASS
				Menampilkan pesan pop up “Data berhasil diperbaharui”	PASS
4	Positif	Button unduh pdf	Klik button “Unduh Pdf”	Berhasil Mengunduh data yang diperbarui dengan ekstensi “.pdf”	PASS
				Berhasil mengambil data dari database yang diperbarui	PASS
				Menampilkan data yang dimbil di dalam file pdf	PASS
5	Positif	Button unduh Excel	Klik button “Unduh Excel”	Berhasil Mengunduh data yang diperbarui dengan ekstensi “.xls”	PASS
				Berhasil mengambil data dari database yang diperbarui	PASS
				Menampilkan data yang dimbil di dalam file excel	PASS

Tabel 4. Pengujian Metode *Whitebox* pada Absensi

Input	Expected Output	Status
Memanggil API absensi saat pertama kali memuat halaman absensi	Menampilkan semua data absensi yang berjalan	FAIL
Klik tombol Update untuk melakukan update pada tabel maupun di database	Menampilkan data terbaru dan mengirim ke database	PASS
Membuka halaman absensi untuk ID karyawan 123, tahun 2024, dan bulan 11	Data absensi bulan November 2024 untuk karyawan dengan ID 123 berhasil ditampilkan	PASS

LocalStorage tidak memiliki item divisi, dan user mencoba membuka halaman admin	Halaman dialihkan ke halaman login	PASS
Mengubah status absensi untuk karyawan 123, hari 5, menjadi Present dan menyimpan perubahan	Data absensi hari 5 untuk karyawan 123 berhasil diperbarui menjadi Present di database	PASS
Mencoba menyimpan perubahan data absensi tanpa koneksi internet	Alert "Terjadi kesalahan saat menyimpan data" dan perubahan tidak tersimpan di database	PASS
/api/attendance/:employeeId/:year/:month dipanggil dengan parameter employeeId=123, year=2024, dan month=11	Server merespon dengan status 200 berbentuk JSON absensi karyawan 123 untuk bulan November 2024	PASS
/api/attendance/bulk-update dipanggil dengan data perubahan absensi yang valid	Server merespon status 200 dengan pesan "Absensi berhasil diperbarui"	PASS
/api/attendance/:year/:month dipanggil dengan year=2024 dan month=11, tetapi tidak ada data absensi dalam rentang waktu tersebut	Server merespon status 200 dengan array kosong	PASS
/api/attendance/:employeeId/:year/:month dipanggil dengan parameter employeeId=999432424, year=2024, dan month=11 (karyawan tidak ditemukan)	Server merespon status 200 dengan array kosong	PASS
/api/attendance/bulk-update dipanggil dengan data perubahan absensi yang tidak valid	Server merespon status 500 dengan pesan error: "Terjadi kesalahan saat memperbarui absensi"	PASS
/api/attendance/:year/:month dipanggil dengan parameter year=2024 dan month=13	Server merespon status 500 dengan pesan error: "Server error"	PASS

Dalam pengujian API, semua route sudah berhasil diimplementasikan dan dapat digunakan untuk melakukan pengelolaan data dan lainnya. Pada pengujian fungsional aplikasi, terdapat fungsi yang gagal yaitu pada saat membuka halaman pertama kali, gagal memanggil API, dan untuk mengatasinya harus menggunakan tombol "Perbarui" untuk memanggil ulang API nya.

4. KESIMPULAN

Pengelolaan absensi dan cuti pegawai memiliki peranan yang penting dalam mengawasi kinerja para pegawai. Pelaksanaan absensi dapat berupa macam dalam pelaksanaan tergantung kebutuhan dan kebijakan pada instansi atau organisasi tersebut. Dalam perkembangan teknologi seperti saat ini, bidang informasi dan teknologi dapat digunakan dalam pengelolaan absensi dan cuti pegawai untuk meningkatkan keefisienan. Dalam implementasi aplikasi terutama pada berbasis web, banyak sekali teknologi yang dapat digunakan salah satunya adalah menggunakan MERN STACK, yang dimana Stack teknologi ini memungkinkan dalam pengembangan aplikasi secara terintegrasi dan juga memiliki struktur aplikasi yang terbilang kompleks. Menggunakan teknologi ini dapat membuat server yang digunakan untuk melakukan pemrosesan data yang dikelola sebelum dihubungkan ke database. Dengan adanya sistem seperti ini, pengelolaan data absensi menjadi lebih aman karena bisa mendapatkan data secara *real-time*, dapat diakses dimana saja dan juga memiliki keamanan yang lebih baik dalam pengelolaan data. Adanya aplikasi ini, membuat proses merekap data absensi dan cuti pegawai dari semua divisi yang ada pada BAHARKAM POLRI menjadi lebih cepat karena sistem otomatis merekap data pada saat Admin melakukan inputan data yang dia lakukan dan Pimpinan bisa langsung mengakses data tersebut.

UCAPAN TERIMA KASIH

Terimakasih disampaikan kepada pihak yang telah membantu proses penelitian dari awal hingga terselesaikan. Semoga dengan adanya penelitian ini dapat memberikan kontribusi di bidang informatika.

REFERENCES

- Alifia, R. A., Irchami, D. M., & Nugroho, R. H. (2024). PERKEMBANGAN TEORI PENGAMBILAN KEPUTUSAN DALAM ERA DIGITAL DAN. MUSYTARI, 9.
- Chairiansyah, S., & Suharto, A. (2023). SISTEM INFORMASI PRESENSI SMARTCARD GURU BERBASIS RFID DENGAN METODE RAD (STUDI KASUS PADA SMK YPK JAYA). E-Bisnis, *Sistem Informasi, Teknologi Informasi*, 18(Vol. 18 No. 2 (2023): Jurnal ESIT(E-Bisnis, Sistem Informasi, Teknologi Informasi)), 60-67.
- Nova, S. H., Widodo, A. P., & Warsito, B. (2022). Analisis Metode Agile pada Pengembangan Sistem Informasi Berbasis Website: *Systematic Literature Review. Techno.Com*, 21(Vol. 21 No. 1 (2022): Februari 2022), 139-148.
- Pertiwi, T. A., Luchia, N. T., & Sinta, P. (2023). Perancangan Dan Implementasi Sistem Informasi Absensi Berbasis Web Menggunakan Metode Agile Software Development. *Jurnal Testing dan Implementasi Sistem Informasi*, 1(Vol. 1 No. 1 (2023): *Jurnal Testing dan Implementasi Sistem Informasi*), 53-66.
- Triandy, R., & Santoso, N. (2020). Pengembangan Aplikasi Web Reservasi Paket Wisata menggunakan MERN Stack (Studi Kasus: Zona Tamasya Tour Organizer). *Jurnal Pengembangan Teknologi Informasi dan Ilmu Komputer*, 4(Vol 4 No 6 (2020): Juni 2020), 1616–1624.