

FRONTEND DEVELOPMENT WEB INTRADIKTI MENGGUNAKAN NEXTJS DI SEKRETARIAT DIREKTORAT JENDERAL PENDIDIKAN TINGGI, RISET, DAN TEKNOLOGI KEMENTERIAN PENDIDIKAN, KEBUDAYAAN, RISET, DAN TEKNOLOGI

Rifansyah Aditya Mahendra
Teknologi Rekayasa Multimedia
Telkom University
Bandung, Indonesia
rifansyaham@student.telkomuniversity.
ac.id

Rio Korio Utoro, S.Kom., M.T.
Teknologi Rekayasa Multimedia
Telkom University
Bandung, Indonesia
korioutoro@telkomuniversity.ac.id

Tafta Zani, M.T.
Teknologi Rekayasa Multimedia
Telkom University
Bandung, Indonesia
taftazani@telkomuniversity.ac.id

Projek ini bertujuan mengembangkan sebuah dashboard web yang ditujukan untuk pegawai Sekretariat Jenderal Pendidikan Tinggi Riset dan Teknologi. Dashboard ini akan mencakup beberapa fitur, seperti biodata pegawai, riwayat status pegawai, pengajuan cuti, gaji pegawai, Employee of The Month, naskah dinas. Projek ini mengimplementasikan metode agile dan waterfall dalam pengembangannya dan akan menggunakan teknologi frontend seperti framework NextJS, Tailwind CSS, serta mengintegrasikan fitur-fitur backend seperti fetch data dan API. Tujuan projek ini adalah memberikan memberikan solusi yang efektif dan efisien bagi pegawai Sekretariat Jenderal Pendidikan Tinggi Riset dan Teknologi serta memberikan pengalaman praktis dalam pengembangan aplikasi web bagi peserta magang.

Kata kunci - Web, NextJS, Intradikti, Tailwind, API

I. PENDAHULUAN

Dalam era digital yang terus berkembang, teknologi informasi memainkan peran yang sangat penting dalam meningkatkan efisiensi dan efektivitas berbagai sektor, termasuk sektor pendidikan tinggi, riset, dan teknologi. Sekretariat Jenderal Pendidikan Tinggi Riset dan Teknologi (Dikti) memahami bahwa untuk mencapai misi dan visinya, adopsi teknologi informasi adalah suatu keharusan. Oleh karena itu, Dikti mengambil langkah progresif dengan mengembangkan sistem dan aplikasi yang inovatif, salah satunya adalah proyek Intradikti.

Seiring berjalannya waktu, kebutuhan pegawai Dikti di bagian sumber daya semakin kompleks. Kebutuhan ini mencakup berbagai aspek mulai dari absensi kehadiran, pembaruan biodata pegawai, manajemen keuangan, hingga pengelolaan dinas. Setiap kebutuhan tersebut memiliki

requirement tersendiri yang sulit dilakukan secara manual. Proses manual yang melibatkan banyak data dan dokumen tidak hanya memakan waktu tetapi juga rentan terhadap kesalahan. Untuk mengatasi tantangan ini, Dikti memerlukan sebuah solusi yang terintegrasi dan efisien, sehingga lahirlah superapp Intradikti.

Proyek Intradikti bertujuan untuk menciptakan sebuah dashboard yang komprehensif dan user-friendly bagi para pegawai Dikti. Dashboard ini dirancang untuk memenuhi berbagai kebutuhan administrasi dan manajemen pegawai. Automasi melalui superapp ini diharapkan dapat mengurangi beban administrasi, meningkatkan akurasi data, dan mempermudah pengambilan keputusan berbasis data.

A. Rumusan Masalah

1. Bagaimana mengembangkan dashboard untuk pegawai Dirjen Dikti yang mencakup informasi pegawai.
2. Bagaimana mengintegrasikan fitur-fitur khusus untuk admin.
3. Bagaimana memastikan sistem dapat diakses dan digunakan dengan mudah oleh pegawai dan admin, serta bagaimana mengatasi kendala teknis yang muncul selama pengembangan dan implementasi sistem.

B. Tujuan

- Tujuan dari pelaksanaan program ini antara lain :
1. Mengembangkan dashboard untuk pegawai Dirjen Dikti.
 2. Menyediakan fitur-fitur khusus untuk admin.
 3. Memastikan sistem mudah diakses dan digunakan oleh pegawai dan admin, serta mampu mengatasi kendala teknis yang muncul selama pengembangan dan implementasi.

C. Batasan Masalah

Batasan masalah dari program ini adalah:

1. Pengembangan aplikasi hanya terbatas pada fitur-fitur yang telah ditentukan.
2. Fokus pengembangan adalah pada antarmuka pengguna dan integrasi fitur, tanpa mencakup pengembangan backend atau server-side yang tidak terkait langsung dengan tugas frontend developer.
3. Proses pengembangan mengikuti metodologi yang telah ditentukan oleh organisasi, yaitu dari sistem agile ke sistem waterfall, dan penyesuaian dilakukan sesuai kebutuhan proyek.
4. Aplikasi diutamakan untuk digunakan oleh pegawai dan admin Dirjen Dikti, dan tidak mencakup fitur untuk publik atau pihak eksternal lainnya.

D. Metode Pengerjaan

Pada kegiatan magang ini terdapat 2 metode pengerjaan yang diterapkan dalam periode yang berbeda yang pertama adalah metode agile yang di terapkan selama periode bulan Februari sampai bulan Mei

Alur pengerjaannya dilakukan seperti berikut :

1. Informasi Pengerjaan Proyek
 2. Penyusunan Business Requirement Document (BRD)
 3. Feasibility meeting
 4. Penyusunan Product Requirement Document (PRD)
 5. Kick off meeting
 6. Proses development
 7. User Acceptance Testing (UAT) dan System Integration Testing (SIT)
 8. Regression Test
 9. Post Implementation Review (PIR)
- kedua metode yang diterapkan adalah waterfall, metode ini diterapkan dimulai bulan Mei sampai akhir kegiatan magang

Alur pengerjaannya dilakukan seperti Berikut :

Proyek Life Cycle :

1. Informasi Pengerjaan Proyek
2. Penyusunan BRD
3. Feasibility Meeting
4. Modul Proyek

Modul Life Cycle :

1. Penyusunan PRD
2. Kickoff Meeting
3. Proses Development
4. UAT, SIT dan Regression Test
5. Production Trial Run

II. KAJIAN TEORI

A. Website

Menurut Rochmawati I. website adalah kumpulan informasi yang terdiri dari halaman web yang saling terhubung satu sama lain yang disediakan secara perorangan, kelompok, atau pun organisasi. Situs web yang baik menampilkan visual yang menarik dan berfungsi sesuai dengan kebutuhan pengguna[1].

B. Frontend

Frontend adalah bagian dari web yang bertanggung jawab untuk menampilkan user interface yang dilihat oleh pengguna

untuk berinteraksi dengan halaman web[2]. Dalam konteks intradikti frontend adalah halaman dashboard yang digunakan untuk melihat berbagai informasi kepegawaian yang meliputi data-data pegawai.

C. Typescript

TypeScript adalah sebuah bahasa pemrograman yang berbasis pada JavaScript, tetapi menambahkan fitur-fitur seperti pengetikan statis, kelas, antarmuka, dan modul. TypeScript bertujuan untuk membuat pengembangan Website lebih mudah dan lebih aman[3].

D. Tailwind CSS

Tailwind CSS adalah CSS framework yang menawarkan konsep baru dengan konsep utility first di mana tidak perlu lagi memikirkan panjang nama kelas untuk komponen HTML[4]. Pada penerapannya, penggunaan Tailwind CSS cukup sederhana dikarenakan Tailwind CSS berbasis utilities cukup berbeda dengan CSS biasa ataupun framework bootstrap yang biasa ditemukan.

E. NextJS

Next.js adalah sebuah kerangka kerja (framework) atau platform pengembangan web yang open-source untuk mengembangkan aplikasi web modern berbasis React yang dibuat oleh Zeit [5].

F. Framework

Framework adalah kumpulan intruksi-intruksi yang dikumpulkan dalam class dan function-function dengan fungsi masing-masing untuk memudahkan developer dalam memanggilmnya tanpa harus menuliskan syntax program yang sama berulang-ulang serta dapat menghemat waktu [6].

G. API

API adalah suatu penghubung yang memungkinkan suatu aplikasi untuk berinteraksi dengan aplikasi lainnya dan berbagi data [7]. API berfungsi untuk memudahkan penggunaan teknologi tertentu ketika membangun perangkat lunak atau aplikasi bagi pengembang. API memiliki beberapa arsitektur, ada REST (Representational State Transfer), GraphQL, dll [8].

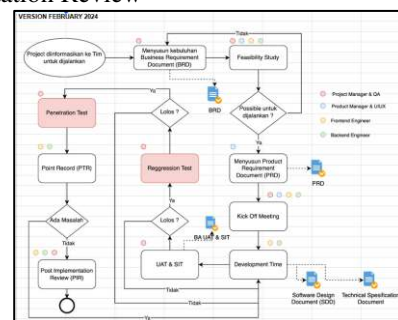
H. Microsoft Visual Studio

Microsoft Visual Studio Code adalah sebuah teks editor ringan dan handal yang dibuat oleh Microsoft untuk sistem operasi multiplatform, artinya tersedia juga untuk versi Linux, Mac, dan Windows [9].

III. METODE

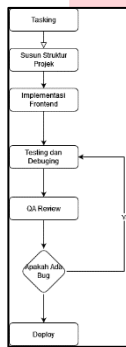
A. Metode Agile

Alur kerja dengan metode ini diterapkan pada tim selama periode bulan Februari sampai bulan Mei. Gambar 6 merupakan alur kerja secara umum dimulai dari pemberitahuan pengerjaan proyek sampai Post Implementation Review



Gambar 1 Alur Kerja Metode Agile

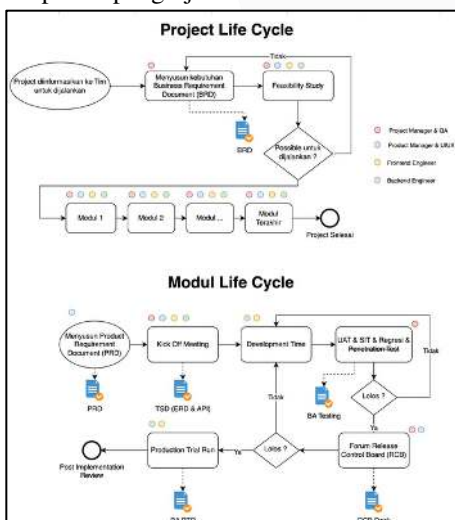
Secara spesifik flow pengerjaan selama proses development terlampir pada gambar 7. Dimulai dari tasking kepada developer, di tahap ini mockup User Interface (UI) sudah dikerjakan oleh desainer sehingga penulis sebagai developer bisa langsung melakukan proses penyusunan struktur proyek dan implementasi desain menjadi frontend web. Setelah implementasi desain selesai frontend developer akan menunggu pengerjaan bagian backend selesai baru setelah itu dapat dilanjutkan untuk mengintegrasikan API dan menambahkan fungsional pada fitur yang dibuat. Setelah pengembangan frontend selesai pengembang akan melakukan testing dan debugging untuk memastikan fitur berjalan dengan baik lalu fitur akan dideploy ke server pengembangan, selanjutnya akan dilakukan review oleh Quality Assurance (QA) jika terdapat bug pada fitur yang dikerjakan maka pengerjaan akan dikembalikan ke pengembang untuk dilakukan refactor code, jika tidak fitur akan dideploy ke lingkungan produksi untuk digunakan oleh pengguna akhir.



Gambar 2 Flowchart Agile

B. Metode WaterFall

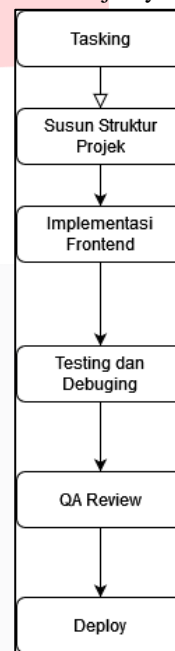
Alur kerja fase ini diterapkan pada tim selama periode bulan Mei sampai masa magang berakhir di bulan Juni. Gambar 7 merupakan alur kerja secara umum dimulai dari informasi pengerjaan proyek sampai ke pengerjaan modul.



Gambar 3 Alur Kerja Metode Waterfall

Secara spesifik flow pengerjaan selama proses development terlampir pada gambar 9. Dimulai dari tasking kepada developer, di tahap ini mockup UI

sudah dikerjakan oleh desainer sehingga penulis sebagai *developer* bisa langsung melakukan proses penyusunan struktur proyek dan implementasi desain menjadi frontend web. Setelah implementasi desain selesai *frontend developer* akan menunggu pengerjaan bagian *backend* selesai baru setelah itu dapat dilanjutkan untuk mengintegrasikan API dan menambahkan fungsional pada fitur yang dibuat. Setelah pengembangan frontend selesai pengembang akan melakukan *testing* dan *debugging* untuk memastikan fitur berjalan dengan baik lalu fitur akan dideploy ke server pengembangan, selanjutnya akan dilakukan review oleh QA tetapi fitur akan tetap dideploy ke lingkungan produksi terlepas dari adanya *bug* atau tidak pada fitur, hasil review akan menjadi catatan dan diberikan ke tim feedback untuk selanjutnya diperbaiki.



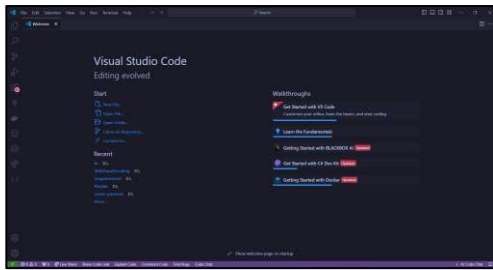
Gambar 4 Flowhart Waterfall

IV. HASIL DAN PEMBAHASAN

a. Kebutuhan Implementasi

- 1) Microsoft Visual Studio Code (VS Code)

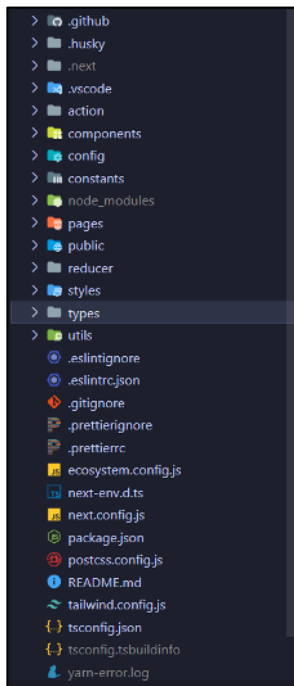
Dalam pengerjaan proyek ini penulis menggunakan VS Code sebagai text editor. VS Code dipilih karena familiaritas penulis dengan aplikasi ini, selain itu ketersediaan extensi yang banyak juga menjadi salah satu pertimbangan dalam pemilihan text editor. Gambar 27 merupakan tampilan saat VS Code dibuka.



Gambar 5 Visual Studio Code

2) Framework NextJS

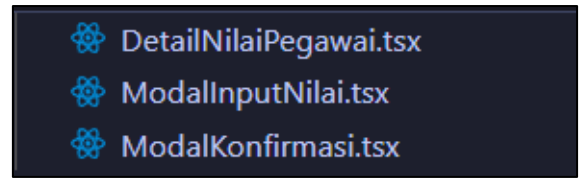
Gambar 28 merupakan gambaran dari struktur proyek yang dibuat, pengerjaan proyek ini menggunakan framework NextJS. Next.js menyediakan fitur server-side rendering dan static site generation yang membantu dalam optimalisasi performa.



Gambar 6 Struktur Proyek NextJS

3) Typescript

Untuk bahasa pemrograman yang digunakan dalam proyek ini adalah typescript. Gambar 29 merupakan penamaan file menggunakan ekstensi typescript.



Gambar 7 Penamaan File Typescript

4) Tailwind CSS

Gambar 30 memperlihatkan penggunaan kelas-kelas Tailwind CSS dalam sebuah komponen UI. Dengan Tailwind CSS, penulis dapat dengan mudah membuat desain yang konsisten dan responsif.



Gambar 8 Penggunaan Tailwind CSS

b. Proses Implementasi

Proyek ini terikat oleh perjanjian Non-Disclosure Agreement (NDA). Oleh karena itu, semua contoh yang ditampilkan dalam dokumen ini berasal dari lingkungan pengembangan (development environment) dan bukan dari lingkungan produksi dan variabel yang digunakan adalah dummy. Hal ini dilakukan untuk menjaga kerahasiaan dan keamanan informasi sensitif yang terkait dengan proyek ini.

i. Integrasi API

1) Pengaturan URL

Ini merupakan proses pengaturan URL yang endpoint yang telah di buat oleh backend developer.



Gambar 9 Pengaturan URL Endpoint

2) Pengaturan Interface API

Setelah pengaturan endpoint selesai selanjutnya penulis melakukan pengaturan interface untuk menentukan struktur dan type data sesuai yang telah disepakati bersama backend developer, hal ini dilakukan agar tidak terjadi miskomunikasi ataupun error pada kode karena ketidaksuaian data dengan bagian backend.


```

)
  if (!isEmpty) {
    console.log('Data Pegawai: ', pegawai);
  }
  return (
    <div>
      <h3>Daftar Pegawai</h3>
      <table border="1">
        <thead>
          <tr>
            <th>No</th>
            <th>Nama</th>
            <th>Jabatan</th>
            <th>Gaji</th>
          </tr>
        </thead>
        <tbody>
          <tr>
            <td>1</td>
            <td>John Doe</td>
            <td>Software Engineer</td>
            <td>10000000</td>
          </tr>
          <tr>
            <td>2</td>
            <td>Jane Smith</td>
            <td>Product Manager</td>
            <td>12000000</td>
          </tr>
          <tr>
            <td>3</td>
            <td>Mike Johnson</td>
            <td>Marketing Specialist</td>
            <td>8000000</td>
          </tr>
          <tr>
            <td>4</td>
            <td>Emily White</td>
            <td>HR Manager</td>
            <td>9000000</td>
          </tr>
          <tr>
            <td>5</td>
            <td>David Brown</td>
            <td>Sales Representative</td>
            <td>7000000</td>
          </tr>
        </tbody>
      </table>
    </div>
  );
}
}

```

Gambar 10 Pengaturan Interface

3) Pemanggilan API

Setelah pengaturan URL dan interface selesai selanjutnya API bisa digunakan.

```

const [filter, setFilter] = React.useState<DetailReq>({
  halaman: 1,
  pagination: 5,
  bulan: properties.selectedMonth ? Number(properties.selectedMonth) : 3,
  tahun: properties.selectedYear ? Number(properties.selectedYear) : 2024,
  direktorat: direktorat,
});

const { data: Detaillist, isValidating } = useCommonApi<DetailReq, DetailRes>({
  EOTM_API_GET_DETAIL,
  filter,
  { method: 'GET' }
});

```

Gambar 11 Pemanggilan API

ii. Mapping data tabel

Setelah konfigurasi API selesai selanjutnya penulis melakukan mapping data tabel untuk melihat apakah API bisa digunakan dan menampilkan data dengan benar jika data berhasil dimunculkan selanjutnya penulis tinggal melakukan penyesuaian UI dan memasukan fitur-fitur lainnya yang dibutuhkan dalam halaman tersebut.

```

({DetailNilaiList?.list || []}).map((each, index) => (
  <tr key={index} className="bg-white hover:bg-gray-100">
    <td className="px-6 py-4 text-xs font-medium text-gray-900">
      {filterState.per_page * (filterState.page - 1) + (index + 1)}
    </td>
    <td className="whitespace-nowrap px-6 py-4 text-xs font-medium text-gray-900">
      {each.nama_pegawai}
    </td>
    <td className="whitespace-nowrap px-6 py-4 text-xs font-medium text-gray-900">
      {each.persentase_presensi}
    </td>
    <td className="whitespace-nowrap px-6 py-4 text-xs font-medium text-gray-900">
      {each.skor_nilai_log_harian}
    </td>
    <td className="whitespace-nowrap px-6 py-4 text-xs font-medium text-gray-900">
      {each.jumlah_log_harian}
    </td>
    <td className="whitespace-nowrap px-6 py-4 text-xs font-medium text-gray-900">
      {each.nilai_pimpinan}
    </td>
    <td className="whitespace-nowrap px-6 py-4 text-xs font-medium text-gray-900">
      {each.total_nilai_akhir}
    </td>
    <td>
      {console.log(each.is_eotm)}
      {filterState.page === 1 && index < 5 && !isFourteenth && (

```

Gambar 12 Mapping Data

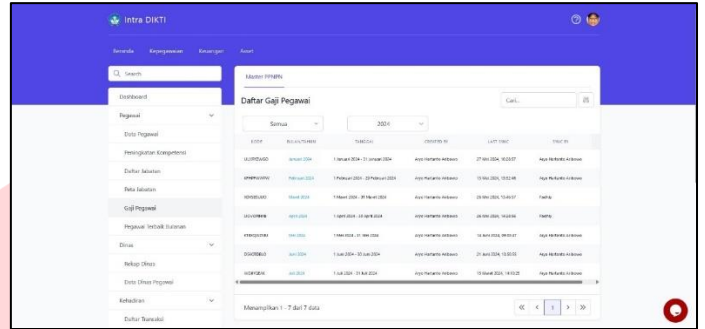
c. Hasil Implementasi

i. Gaji Pegawai

Merupakan halaman untuk admin memantau nilai gaji setiap pegawai dalam pengaplikasiannya halaman ini hanya

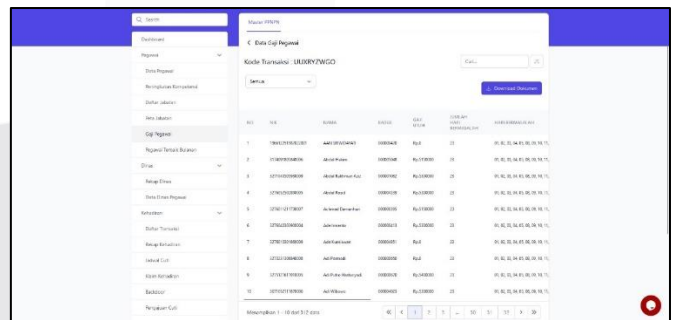
berisikan gaji untuk pegawai berstatus pnpn saja, dan detailnya dibagi berdasarkan bulan.

Pada gambar 35 menunjukkan hasil implementasi dari halaman utama gaji pegawai, halaman ini merupakan halaman yang akan dilihat pengguna saat membuka url gaji pegawai, dalam halaman ini pengguna akan diperlihatkan daftar bulan yang memiliki data gaji pegawai didalamnya.



Gambar 13 Halaman Utama Gaji Pegawai

Pada gambar 36 menunjukkan hasil implementasi dari halaman detail gaji pegawai, halaman ini menampilkan informasi yang berhubungan dengan gaji pegawai, seperti gaji pokok, potongan gaji dan gaji akhir yang diterima pegawai. Pada halaman ini pengguna dapat mencari nama pegawai yang diinginkan secara spesifik dan juga membatasi data berdasarkan unit kerja yang diinginkan. Pengguna juga dapat mengunduh data yang ada pada halaman ini menggunakan tombol *download* dokumen.

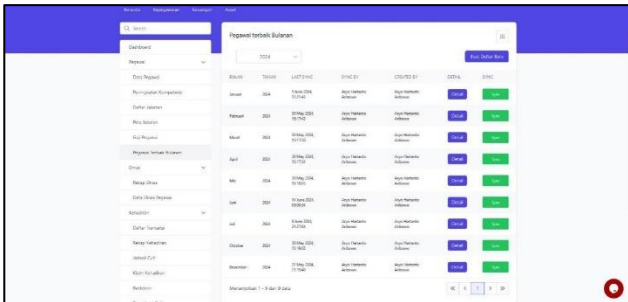


Gambar 14 Detail Gaji Pegawai

ii. Employee of The Month (EOTM)

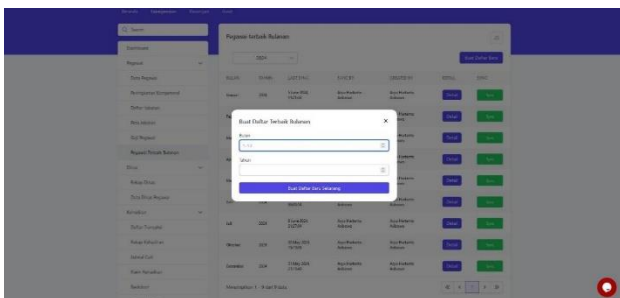
Fitur untuk pemilihan *Employee of The Month* (EOTM) dimana pemilihan EOTM akan dilakukan berdasarkan 5 pegawai terbaik dan pemilihan dilakukan setiap tanggal 14 di setiap bulannya oleh Kepala Tata Usaha (KaTU) ada 2 cara pemilihan EOTM yang pertama KaTU memberikan nilai untuk masing-masing pegawai dan sistem akan memilih pegawai teratas untuk menjadi EOTM di bulan tersebut, yang

kedua KaTU bisa memilih langsung 1 nama dari 5 anama teratas untuk dijadikan EOTM di bulan tersebut. Pada gambar 37 menunjukkan hasil implementasi dari halaman utama EOTM, halaman ini merupakan halaman yang akan dilihat pengguna saat membuka url EOTM, dalam halaman ini pengguna akan diperlihatkan daftar bulan yang memiliki data detail nilai pegawai didalamnya.



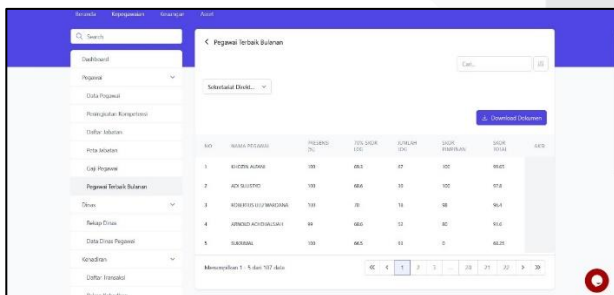
Gambar 15 Halaman Utama EOTM

Gambar 38 merupakan form untuk menambahkan bulan pada halaman utama EOTM, disini pengguna harus memasukan bulan dan tahun yang ingin dibuat.



Gambar 16 Form Tambah Daftar EOTM

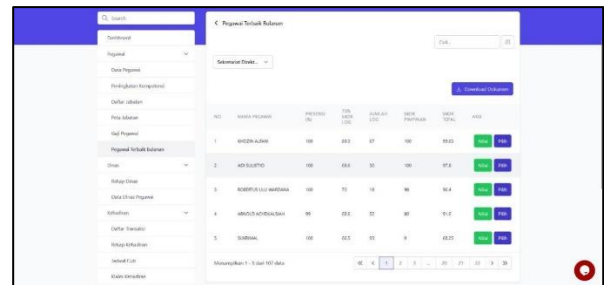
Gambar 39 merupakan halaman detail EOTM halaman ini berisikan detail nilai pegawai dan diurutkan berdasarkan rata-rata nilai tertinggi, dan 5 pegawai teratas adalah pegawai yang menjadi nominasi untuk menjadi EOTM.



Gambar 17 Detail EOTM

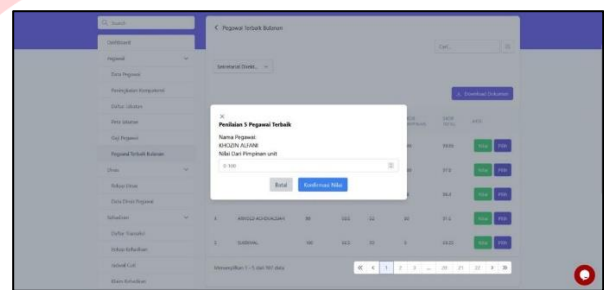
Gambar 40 merupakan tampilan detail nilai EOTM khusus untuk tanggal 14, ditanggal ini akan muncul 2 tombol yang bisa digunakan untuk memilih pegawai terbaik, yang pertama

ada tombol nilai agar pengguna dapat memberi penilaian bagi pegawai yang diinginkan, yang kedua adalah tombol pilih yang akan mengubah status pegawai yang dipilih menjadi EOTM secara langsung tanpa dipengaruhi oleh nilai pegawai.



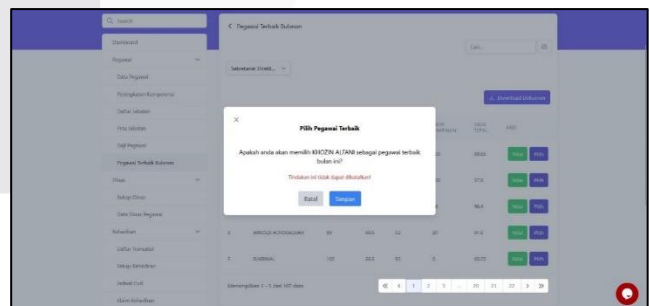
Gambar 18 Detail EOTM Dengan Tombol Aksi

Gambar 41 merupakan form untuk pengisian nilai pegawai, form ini dapat dibuka melalui tombol nilai yang ada di halaman detail EOTM.



Gambar 19 Form Pemberian Nilai Pegawai

Gambar 42 merupakan pop up untuk memvalidasi pemilihan pegawai terbaik, pop up ini akan muncul jika tombol pilih di halaman detail EOTM diklik. Jika tombol simpan diklik maka pegawai yang dipilih akan otomatis menjadi pegawai terbaik di bulan itu.



Gambar 20 Form Pemilihan EOTM

V. KESIMPULAN

Proyek Intradikti berhasil mengembangkan sebuah dashboard komprehensif untuk pegawai Dirjen Dikti yang mencakup berbagai fitur penting seperti gaji pegawai, EOTM. Penerapan metodologi pengembangan yang

fleksibel, dari sistem agile hingga waterfall, telah memungkinkan penyesuaian yang diperlukan selama proses pengembangan. Meskipun terdapat tantangan teknis dan kendala lainnya, proyek ini mampu memberikan solusi yang efektif dan meningkatkan efisiensi serta efektivitas operasional di lingkungan Dirjen Dikti.

REFERENSI

- [1] I. Rochmawati, "ANALISIS USER INTERFACE SITUS WEB IWEARUP.COM," 2019. [Daring]. Tersedia pada: www.iwearup.com
- [2] Terttiaavini, I Made Agus Oka Gunawan, Kraugusteeliana, E. Winarno, dan Rony Sandra Yofa Zebua, "Perancangan dan Implementasi Frontend Web untuk Sistem Pengaduan Masyarakat," *Jurnal Informasi dan Teknologi*, hlm. 112–126, Apr 2023, doi: 10.37034/jidt.v5i1.290.
- [3] D. Setyo Nugroho dkk., "Inovasi Pembaruan Desain Website Penyederhana Link Unesa Menggunakan Typescript Dan Node.js," *Jurnal Ilmu Teknik*, vol. 1, no. 2, hlm. 35–40, 2023.
- [4] R. J. Romadhondaru dan A. Basuki, "Visualisasi Topologi Jaringan berdasarkan Data Routing Border Gateway Protocol," 2022. [Daring]. Tersedia pada: <http://j-ptiik.ub.ac.id>
- [5] Maulana Akbar Ramadhan, "Pengembangan Frontend Sistem Informasi Layanan Tekos Berbasis Web Menggunakan Framework NextJS," Universitas Telkom, S1 Rekayasa Perangkat Lunak, hlm. 4–6, 2023.
- [6] A. F. Sallaby dan I. Kanedi, "Perancangan Sistem Informasi Jadwal Dokter Menggunakan Framework Codeigniter," 2020.
- [7] I. Kurniawan dan F. Rozi, "REST API Menggunakan NodeJS pada Aplikasi Transaksi Jasa Elektronik Berbasis Android," 2020. [Daring]. Tersedia pada: <http://jurnal-itsi.org>
- [8] N. K. Akmal dan M. N. Dasaprawira, "RANCANG BANGUN APPLICATION PROGRAMMING INTERFACE (API) MENGGUNAKAN GAYA ARSITEKTUR GRAPHQL UNTUK PEMBUATAN SISTEM INFORMASI PENDATAAN ANGGOTA UNIT KEGIATAN MAHASISWA (UKM) STUDI KASUS UKM STARLABS," 2022. [Daring]. Tersedia pada: <http://www.jurnal.umk.ac.id/sitech>
- [9] J. Sains dkk., "APLIKASI BUKU TAMU MENGGUNAKAN FITUR KAMERA DAN AJAX BERBASIS WEBSITE PADA KANTOR DISPORA KOTA MEDAN," 2022.