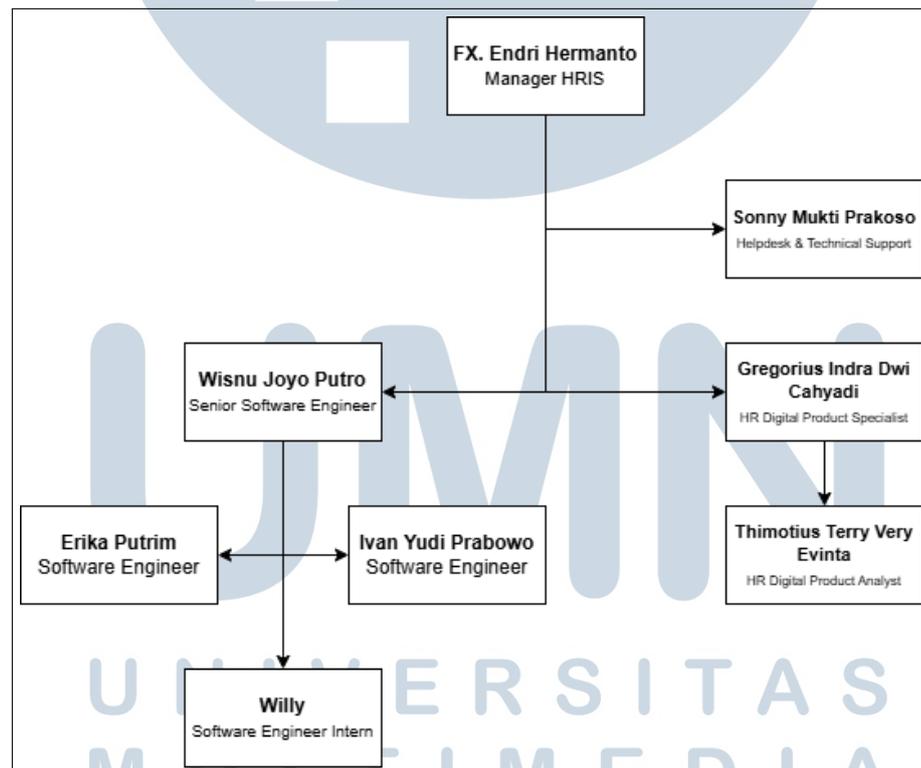


BAB 3 PELAKSANAAN KERJA MAGANG

3.1 Kedudukan dan Koordinasi

Kegiatan kerja magang dilaksanakan di Functional Unit (FU) Corporate Human Resources (CHR), tepatnya pada Departemen Human Resource Information System (HRIS) dengan posisi sebagai Software Engineer Intern. Departemen HRIS berada di bawah kepemimpinan Bapak FX. Endri Harmanto selaku manajer. Secara struktural, departemen ini terbagi ke dalam tiga seksi utama, yaitu Technical Support, Software Engineer, dan HR Digital Product, yang seluruhnya berada di bawah koordinasi langsung beliau. Gambaran Struktur Organisasi dan Kedudukan dalam Departemen HRIS ditunjukkan pada Gambar 3.1.



Gambar 3.1. Struktur Organisasi dan Kedudukan dalam Departemen HRIS

Divisi HR Digital Product memiliki tanggung jawab utama dalam menganalisis kebutuhan sistem serta merancang proses bisnis yang mendukung operasional sumber daya manusia. Divisi ini dipimpin oleh Bapak Gregorius Indra Dwi Cahyadi selaku HR Digital Product Specialist, dengan dukungan dari Bapak

Thimotius Terry Very Ervinta yang menjabat sebagai HR Digital Product Analyst. Sementara itu, seksi Software Engineer bertugas dalam proses pengembangan, implementasi, serta pemeliharaan sistem. Seksi ini dikoordinasikan oleh Bapak Wisnu Joyo Putro sebagai Senior Software Engineer dan didukung oleh dua Software Engineer, yaitu Bapak Ivan Yudi Prabowo dan Ibu Erika Putrim. Sebagai bagian dari upaya menjaga layanan teknis tetap optimal, Departemen HRIS juga didukung oleh Bapak Sonny Mukti Prakoso yang bertugas sebagai Helpdesk Technical Support.

3.2 Tugas yang Dilakukan

Selama menjalani program magang, bertanggung jawab dalam rancang bangun sistem pengolahan data mentah mesin absensi berbasis web dan sql server yang bertujuan untuk menggantikan metode pengolahan data mentah dari mesin absensi yang sebelumnya digunakan. Sistem ini dirancang agar mampu mengolah data mentah berupa angka numerik dari mesin absensi menjadi format yang lebih terstruktur, mencakup informasi seperti NIK, tanggal, jam, status kehadiran, dan terminal. Selain itu, sistem ini juga memiliki mekanisme penggabungan data check-in dan check-out yang dilakukan melalui stored procedure yang dijalankan secara otomatis berdasarkan jadwal menggunakan SQL Server Agent. Data absensi yang telah diproses kemudian dikirimkan ke sistem ERP Odoo dengan bantuan Pentaho, sehingga proses pengolahan dan pengiriman data menjadi lebih cepat, efisien, dan minim kesalahan dibandingkan dengan metode sebelumnya. Setelah data dikirim ke Odoo, tim Helpdesk Technical Support akan memantau status dan keberhasilan proses pengiriman tersebut.

Dalam membangun, sistem ini dilengkapi dengan berbagai fitur untuk mendukung kelancaran proses kerja. Di antaranya adalah fitur unggah data absensi secara manual yang digunakan sebagai alternatif ketika terjadi ketidaksesuaian format file, serta fitur log error yang berfungsi untuk mencatat data yang gagal diproses agar dapat ditinjau dan diperbaiki di kemudian hari. Sistem ini juga memungkinkan pengguna untuk mengunduh data mentah berdasarkan tanggal tertentu dan menyediakan fitur filter berdasarkan tanggal dan NIK untuk mempermudah pencarian dan pengecekan data. Selain itu, tersedia pula fitur pengiriman otomatis yang memungkinkan sistem memproses dan mengirim data secara berkala tanpa perlu intervensi manual. Setiap entri data dilengkapi dengan status pengiriman yang menunjukkan apakah data berhasil dikirim atau mengalami

kegagalan. Untuk mendukung transparansi proses, ditambahkan pula progress bar dan estimasi waktu sebagai indikator visual selama proses pengiriman berlangsung. Lokasi penyimpanan data yang telah maupun yang belum diproses juga dapat diubah sesuai kebutuhan melalui pengaturan yang disimpan dalam tabel konfigurasi di SQL Server. Proses unggah data dapat dilakukan secara manual maupun otomatis, di mana metode manual berguna ketika data perlu dikoreksi terlebih dahulu, sedangkan metode otomatis memungkinkan sistem memantau folder secara real-time dan memproses file secara langsung.

Setelah data berhasil masuk ke dalam database, stored procedure akan secara otomatis menggabungkan data check-in dan check-out menjadi satu entri lengkap, termasuk menyesuaikan shift malam berdasarkan lokasi kerja atau PersArea. Proses ini juga mencakup pengambilan ID karyawan, ID terminal check-in dan check-out, serta pembentukan data hasil penggabungan (concat) berdasarkan parameter tersebut. Penggabungan ini dilakukan sesuai jadwal yang telah ditentukan melalui SQL Server Agent, sehingga menghasilkan data yang sudah terstruktur dan siap dikirim ke Odoo sebagai satu entri utuh. Sistem ini dibangun menggunakan bahasa pemrograman Python dengan framework Flask sebagai backend utama, serta HTML dan JavaScript untuk membangun antarmuka pengguna. Basis data yang digunakan adalah SQL Server, sedangkan proses transformasi data dibantu oleh library Pandas, dan koneksi database dikelola menggunakan library pyodbc. Seluruh proses pengembangan dilakukan dalam lingkungan Visual Studio Code. Proyek ini merupakan hasil kerja individu tanpa menggunakan platform kolaborasi atau version control seperti GitHub, karena pengembangannya bersifat personal.

3.3 Uraian Pelaksanaan Magang

Pelaksanaan kerja magang dapat diuraikan dalam beberapa tahap, yaitu proses pelaksanaan, perancangan sistem, dan tampilan web.

3.3.1 Proses Pelaksanaan

Pada minggu-minggu awal, kegiatan difokuskan pada pengenalan lingkungan kerja, pemahaman alur kerja tim, serta teknologi yang digunakan dalam pengembangan proyek. Mulai mempelajari framework Flask sebagai backend dengan bahasa pemrograman Python, serta JavaScript untuk membangun antarmuka pengguna (UI) yang interaktif dan responsif. Selain itu, juga

diperkenalkan pada penggunaan SQL Server sebagai sistem manajemen basis data utama untuk menyimpan dan mengelola data aplikasi. Karena ini merupakan pengalaman pertama menggunakan SQL Server, jadi membutuhkan waktu untuk mempelajari cara konfigurasi database, pembuatan query, serta cara mengintegrasikan database tersebut dengan aplikasi Flask. Setelah memahami dasar-dasarnya, pengembangan proyek dimulai secara bertahap, mulai dari pembangunan backend menggunakan Flask hingga pengembangan UI dengan JavaScript yang menampilkan data absensi secara dinamis dan menyediakan fitur unggah data.

Sejak awal, sistem absensi yang di rancang dan di bangun untuk mengolah data mentah dari mesin absensi menjadi data yang lebih terstruktur, mencakup informasi seperti NIK, tanggal, jam, status kehadiran, dan terminal. Data tersebut kemudian disimpan ke dalam database dan awalnya direncanakan untuk langsung dikirim ke sistem ERP Odoo. Namun, seiring berjalannya waktu dan adanya perubahan kebijakan internal perusahaan, proses pengiriman ke Odoo tidak lagi dilakukan langsung oleh sistem yang telah dibuat. Sebagai gantinya, sistem difokuskan untuk mempersiapkan dan menyimpan data absensi dalam format yang telah ditentukan, yang kemudian diproses dan dikirim ke Odoo menggunakan alat integrasi Pentaho yang dikelola oleh tim internal perusahaan.

Sebagai bagian dari penyesuaian tersebut, sistem yang telah dibuat dimodifikasi agar tidak hanya menyimpan data absensi awal dalam bentuk terstruktur, tetapi juga melakukan pemrosesan lanjutan menggunakan stored procedure pada SQL Server. Prosedur ini bertugas menggabungkan data check-in dan check-out menjadi satu entri lengkap, menyesuaikan kondisi shift malam berdasarkan lokasi kerja (PersArea), mengambil ID karyawan, ID terminal check-in dan check-out, serta membentuk data gabungan (concat) untuk keperluan integrasi selanjutnya. Perubahan ini membuat sistem menjadi lebih kompleks dibandingkan versi awal. Selama proses pengembangan tersebut, Pemahaman mengenai SQL Server terus diperdalam guna mendukung pengelolaan serta optimalisasi performa penyimpanan data. Dengan demikian, seluruh komponen dalam sistem—baik dari sisi backend, frontend, maupun pengelolaan basis data—dapat terintegrasi secara optimal dan sesuai dengan kebutuhan di lingkungan kerja. Pelaksanaan kerja magang diuraikan seperti pada Tabel 3.1.

Tabel 3.1. Pekerjaan yang dilakukan tiap minggu selama pelaksanaan kerja magang

Minggu Ke -	Pekerjaan yang dilakukan
1, 2	Fokus pada cleansing dan input data karyawan ke Odoo, pembuatan salary rule serta struktur payroll, dan pembaruan data user di SAP.
3, 4	Melakukan mapping data dari SAP ke SQL Server dan Odoo, pengujian integrasi, revisi relasi tabel, serta mulai perancangan API dan penanganan error pada payslip.
5, 6	Pengembangan program pengolahan data absensi menggunakan Python dan antarmuka UI/UX. Data terhubung ke SQL Server dan mendukung upload otomatis dari folder lokal.
7, 8	Integrasi sistem absensi ke Odoo via API, penambahan fitur log gagal, filter data, dan download per NIK/tanggal. Proses upload dan pengiriman data dioptimalkan.
9, 10	Peningkatan kecepatan pengiriman data, perbaikan format waktu, dan penyesuaian tampilan UI seperti progress bar dan indikator pemrosesan.
11, 12	Implementasi retry otomatis, validasi NIK sebelum input data, pembaruan struktur status data (flag), serta pembuatan tabel validasi data karyawan.
13, 14	Pemisahan logika retry dari backend utama, perbaikan pelacakan progres berdasarkan response Odoo, serta penanganan deadlock dan urutan pengiriman data check-in/out.
15, 16	Pembuatan sistem penggabungan otomatis check-in dan check-out melalui trigger dan stored procedure, penambahan kolom ID terminal dan mapping id.employee.
17, 18	Peralihan dari trigger ke stored procedure untuk stabilitas pemrosesan data besar, penambahan validasi agar data tidak tertimpa, serta pembentukan format data siap kirim.
19, 20	Penanganan data dengan format salah melalui tabel khusus, penyesuaian tampilan dan parsing, serta penghapusan fungsi Odoo karena proses dialihkan ke Pentaho.

Minggu Ke -	Pekerjaan yang dilakukan
21	Menyesuaikan UI, validasi file upload, memperbaiki error log, dan menambahkan fitur auto-refresh serta log retry agar sistem lebih informatif.

Selain tugas yang dikerjakan tiap minggu nya yang dijabarkan pada tabel 3.1 terdapat juga Progress Report Meeting setiap 1 kali seminggu yang dilakukan pada hari jumat pukul 14:00 sampai dengan selesai.

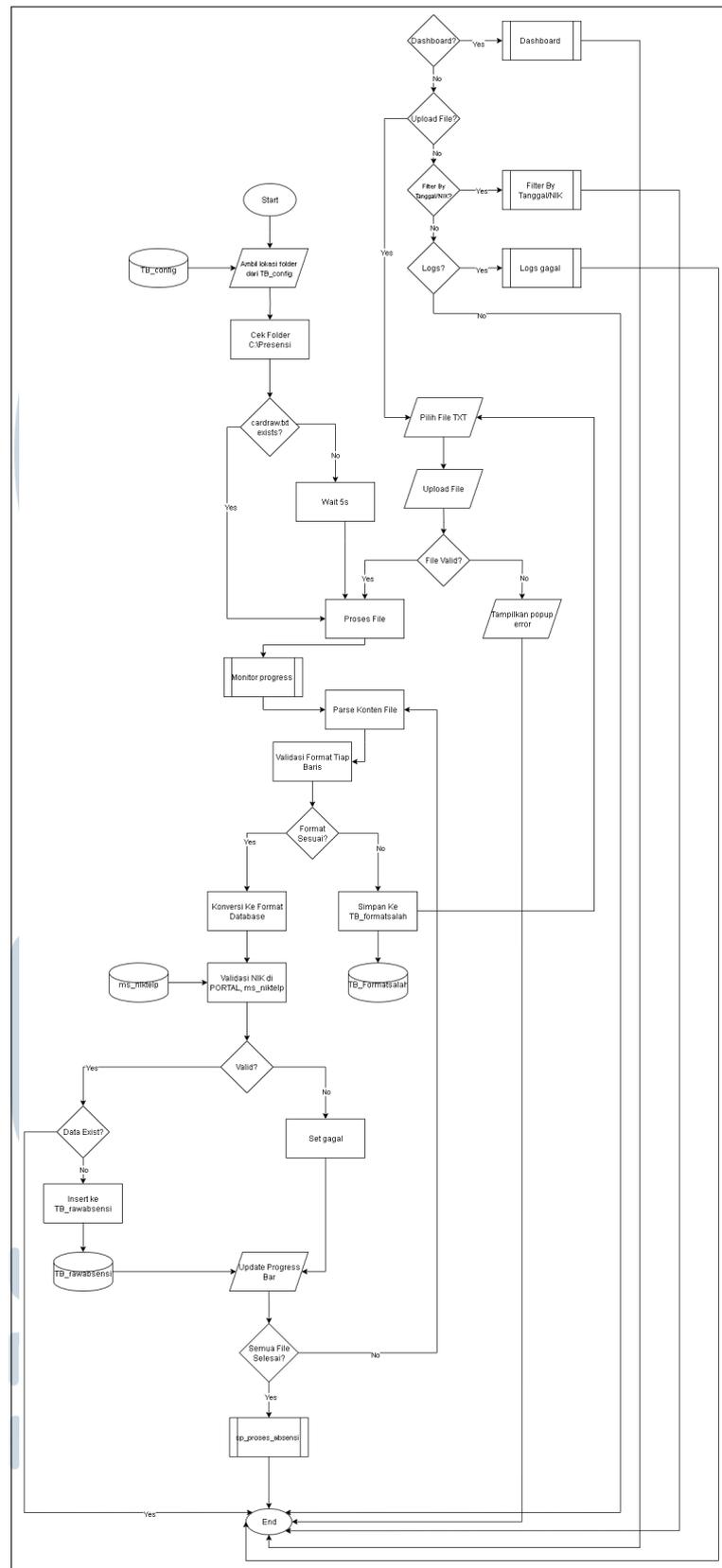
3.3.2 Perancangan Sistem

Perancangan sistem pengolahan data absensi ditampilkan dalam bentuk flowchart yang menggambarkan alur kerja proses sistem tersebut, serta dilengkapi dengan struktur tabel yang digunakan dalam pengolahan data absensi.

A Flowchart

Flowchart adalah sebuah diagram yang digunakan untuk menggambarkan alur logika atau langkah-langkah dalam suatu proses secara sistematis dan terstruktur. Flowchart memanfaatkan simbol-simbol grafis untuk menunjukkan urutan aktivitas, pengambilan keputusan, input, output, dan proses lainnya dalam sistem. Dalam konteks ini, flowchart berfungsi untuk memvisualisasikan proses pengolahan data mentah dari mesin absensi hingga menjadi data yang sudah diproses dan siap untuk dikirim ke sistem Odoo. Dengan adanya flowchart, pemahaman terhadap logika dan jalannya sistem menjadi lebih mudah dan jelas, sehingga dapat memudahkan proses pengembangan, pemeliharaan, serta perbaikan sistem ke depannya.

U N I V E R S I T A S
M U L T I M E D I A
N U S A N T A R A



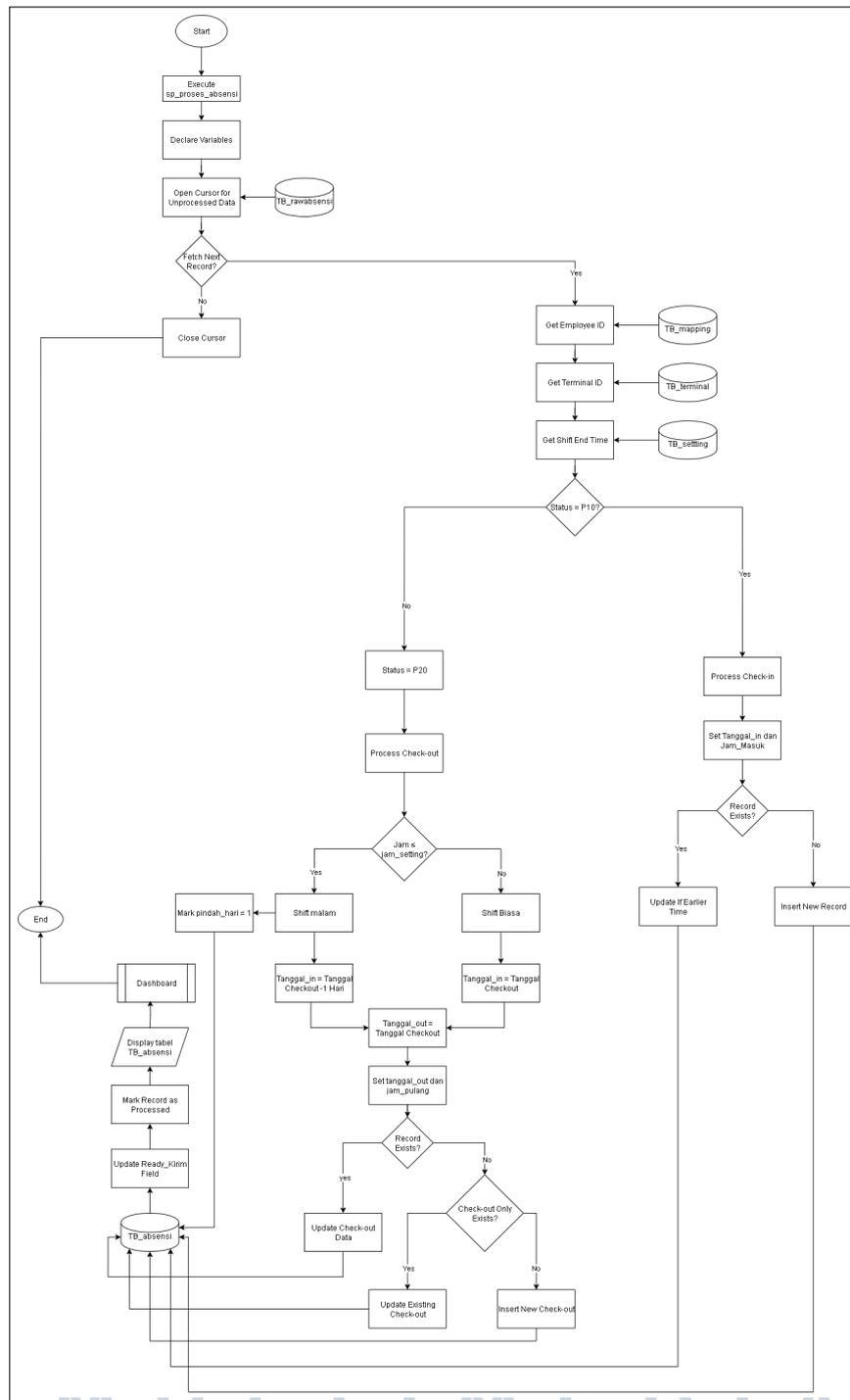
Gambar 3.2. Flowchart Sistem Absensi

Flowchart yang digambarkan pada Gambar 3.2 merupakan integrasi antara sistem *auto upload* dan fitur *upload manual* pada sistem pengolahan data absensi. Proses dimulai dari pengambilan lokasi folder sumber dari tabel konfigurasi TB_config. Sistem kemudian melakukan pengecekan terhadap folder C:\Presensi untuk mendeteksi apakah terdapat file bernama cardraw.txt. Jika file tidak ditemukan, sistem akan menunggu selama 5 detik sebelum melakukan pengecekan ulang. Jika file tersedia, sistem akan melanjutkan proses file tersebut.

Setelah file ditemukan, sistem akan mengaktifkan fitur *monitor progress* untuk mencatat status pemrosesan. File kemudian di-*parse* dan tiap baris akan divalidasi formatnya. Jika format tidak sesuai, maka data tersebut disimpan ke dalam tabel TB_format_salah. Jika format sesuai, data dikonversi ke format database dan dilakukan validasi NIK terhadap data master pada database PORTAL, tabel ms_nikelp. Jika NIK valid dan data belum ada sebelumnya, maka data akan dimasukkan ke dalam tabel TB_rawabsensi. Jika tidak valid, maka status data ditandai sebagai gagal. Setelah tiap file diproses, sistem memperbarui progress bar, dan jika semua file selesai, maka prosedur tersimpan tp_proses_absensi dijalankan untuk memproses data ke tahap berikutnya.

Di sisi lain, pengguna juga dapat mengunggah file secara manual melalui dashboard aplikasi. User akan memilih file .TXT terlebih dahulu, lalu sistem akan melakukan validasi apakah file tersebut valid. Jika file tidak valid, maka sistem menampilkan *popup error*. Jika valid, sistem akan memproses file dan melalui langkah-langkah parsing, validasi format, serta validasi NIK seperti pada proses otomatis. Proses ini juga akan mengupdate progress bar dan memicu prosedur tp_proses_absensi jika semua file telah selesai diproses.

Selain dua fitur utama tersebut, flowchart ini juga menunjukkan navigasi utama ke beberapa bagian sistem seperti dashboard, halaman unggah file, halaman log gagal, dan fitur unduh data (download.txt). Hal ini memastikan pengguna memiliki kontrol penuh atas seluruh proses mulai dari upload data mentah hingga monitoring kesalahan dan ekspor hasil akhir.



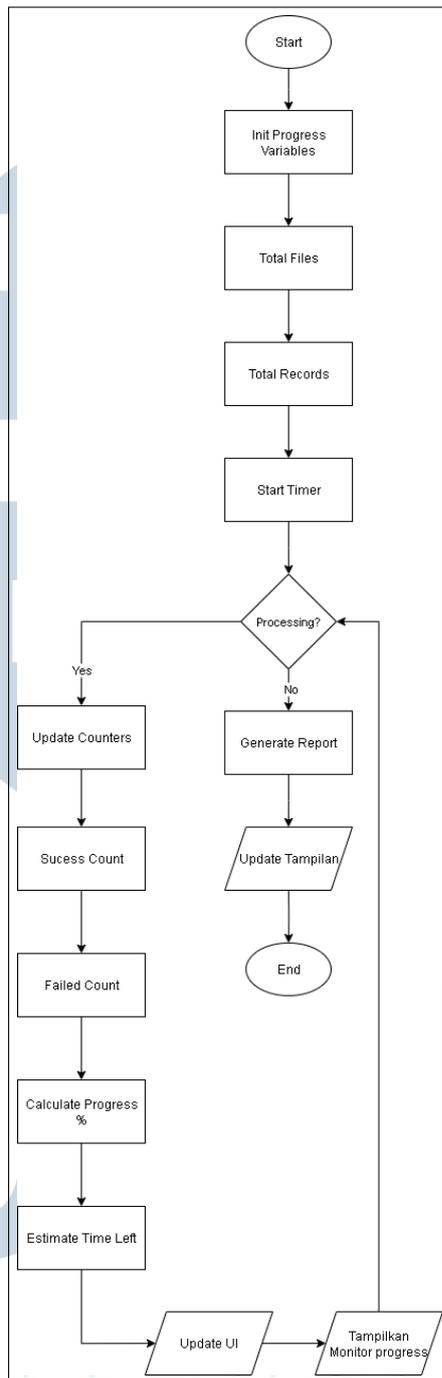
Gambar 3.3. Flowchart Stored Procedure

Flowchart yang digambarkan pada Gambar 3.3 merupakan proses utama dalam sistem pengolahan data absensi menggunakan prosedur tersimpan (stored procedure) bernama `sp_proses_absensi`. Proses dimulai dari pemanggilan prosedur melalui `tp_proses_absensi`, kemudian sistem mendeklarasikan variabel yang

diperlukan dan membuka *cursor* untuk membaca data mentah yang belum diproses dari tabel TB_rawabsensi. Selanjutnya, sistem melakukan iterasi terhadap setiap baris data. Untuk setiap record, sistem mengambil informasi terkait seperti ID karyawan dari TB_mapping, ID terminal dari TB_terminal, dan waktu akhir shift dari TB_setting. Setelah data referensi diperoleh, sistem mengecek status absensi apakah berupa *check-in* (status P10) atau *check-out* (status P20).

Jika status adalah P10, maka sistem memproses data masuk dengan menyimpan Tanggal_in dan Jam_masuk. Jika data sudah ada, maka waktu masuk akan diperbarui hanya jika jam baru lebih awal dari yang sebelumnya. Untuk status P20, sistem akan memproses jam keluar. Jika jam keluar lebih kecil dari batas shift malam yang ditentukan pada jam_setting, maka dianggap sebagai shift malam dan Tanggal_in akan diset satu hari sebelum Tanggal_out. Setelah itu, sistem memeriksa apakah record check-in untuk NIK tersebut sudah ada. Jika sudah, data check-out akan diperbarui. Jika belum ada, maka akan dibuat record baru sebagai check-out. Setelah semua data berhasil diproses, sistem akan memperbarui kolom status (ready_kirim) dan menandai record tersebut sebagai sudah diproses. Hasil akhirnya dimasukkan ke dalam tabel TB_absensi dan akan ditampilkan di dashboard sistem.



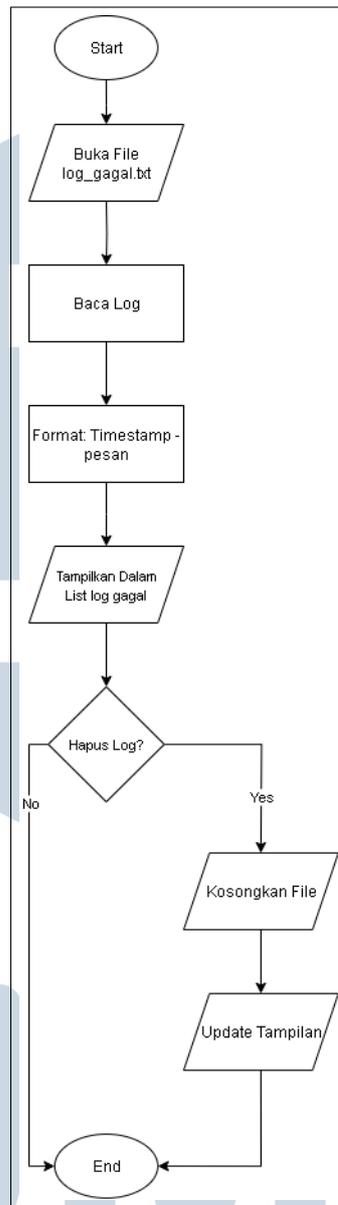


Gambar 3.4. Flowchart Monitoring Tampilan Data

Flowchart yang digambarkan pada Gambar 3.4 merupakan fitur *Monitor Progress Tracking* berfungsi untuk memantau secara real-time proses pemrosesan file absensi yang telah diunggah ke dalam sistem. Proses dimulai dengan inialisasi modul *Monitor Progress*, kemudian sistem menginisialisasi variabel-variabel progress seperti jumlah total file dan jumlah total record yang akan

diproses. Setelah itu, sistem memulai timer untuk menghitung estimasi waktu. Selama proses pemrosesan berlangsung, sistem akan terus memperbarui jumlah data yang berhasil diproses (success count) dan data yang gagal diproses (failed count). Selanjutnya, sistem menghitung persentase progress dan estimasi waktu yang tersisa berdasarkan data yang sudah diproses. Semua informasi ini kemudian diperbarui secara berkala ke tampilan UI untuk memberikan informasi transparan kepada pengguna. Jika proses sudah selesai, sistem akan menghasilkan laporan akhir dan menampilkan hasilnya pada antarmuka pengguna. Fitur ini memberikan visibilitas penuh terhadap proses backend secara interaktif dan informatif.

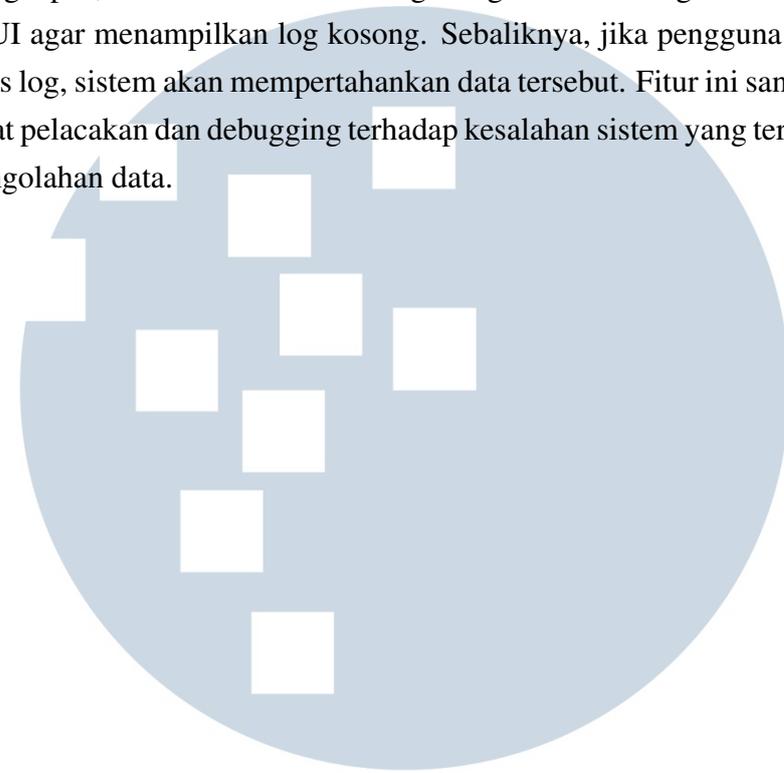




Gambar 3.5. Flowchart Proses Log Data Gagal Upload

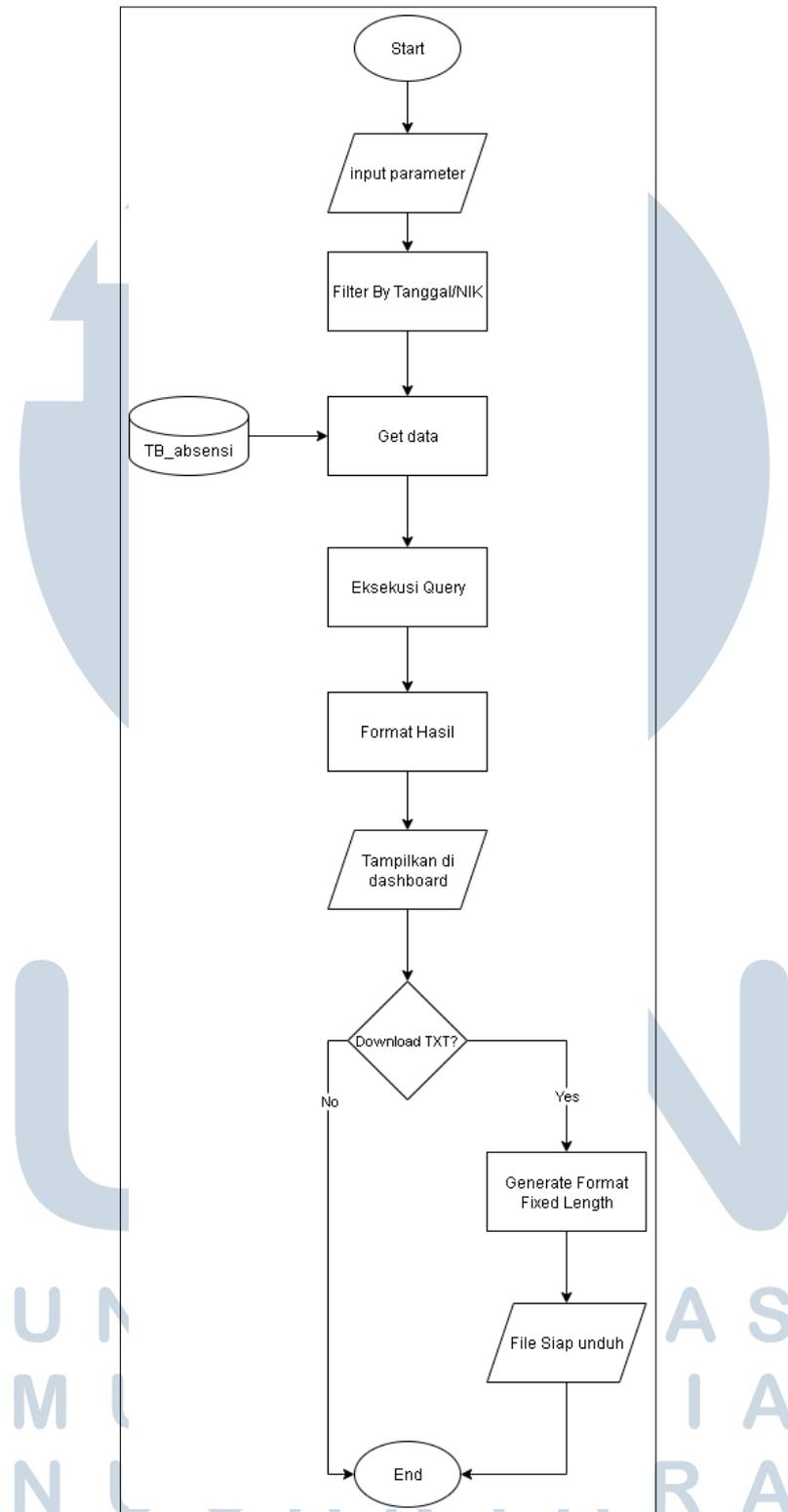
Flowchart yang digambarkan pada Gambar 3.5 merupakan fitur yang berfungsi untuk menampilkan daftar data absensi yang gagal diproses secara otomatis maupun manual. Proses dimulai dengan membuka file log bernama log_gagal.txt yang menyimpan catatan kesalahan selama proses upload atau konversi data. Sistem membaca isi log tersebut, lalu memformat setiap baris log menjadi format yang mudah dibaca, yaitu dalam bentuk timestamp diikuti oleh pesan kesalahan. Data log yang telah diformat ini kemudian ditampilkan dalam daftar log gagal pada antarmuka pengguna. Pengguna diberikan opsi

untuk menghapus seluruh isi log apabila diperlukan. Jika pengguna memilih untuk menghapus, maka sistem akan mengosongkan isi file log dan memperbarui tampilan UI agar menampilkan log kosong. Sebaliknya, jika pengguna tidak ingin menghapus log, sistem akan mempertahankan data tersebut. Fitur ini sangat penting sebagai alat pelacakan dan debugging terhadap kesalahan sistem yang terjadi selama proses pengolahan data.



UMMN

UNIVERSITAS
MULTIMEDIA
NUSANTARA



Gambar 3.6. Flowchart Proses Filter By Tanggal/NIK

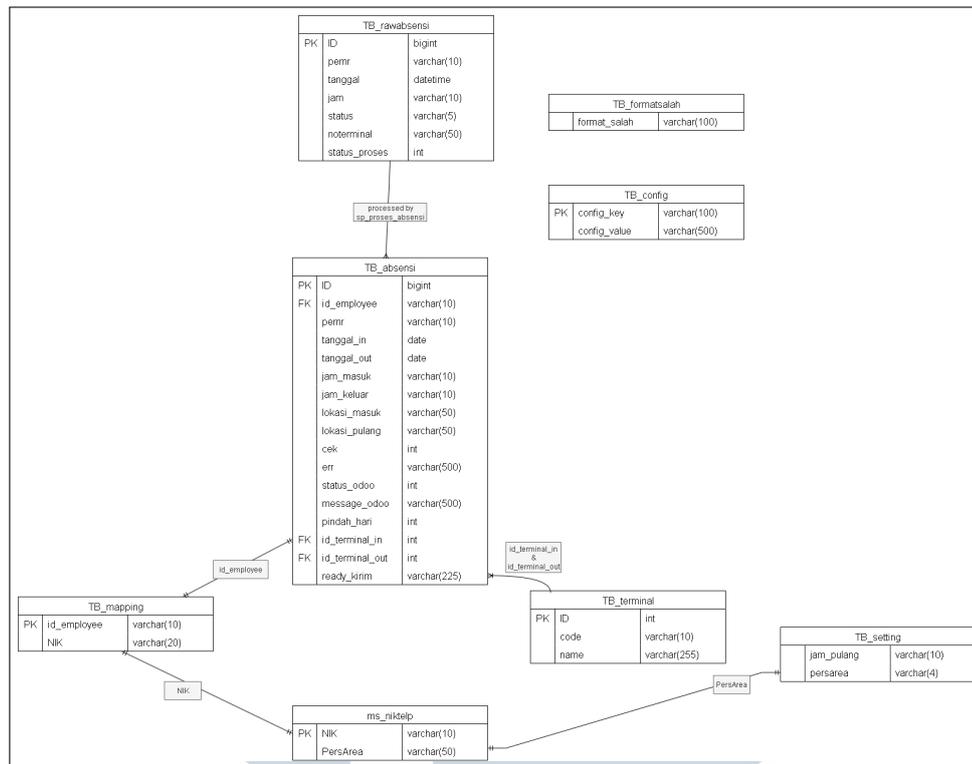
Flowchart yang digambarkan pada Gambar 3.6 merupakan fitur *Filter dan*

Download yang memungkinkan pengguna untuk melakukan penyaringan (filtering) data absensi berdasarkan parameter tertentu, seperti tanggal dan NIK, kemudian mengunduh hasil filter tersebut dalam bentuk file .TXT dengan format *fixed length*. Proses diawali dengan input parameter oleh pengguna melalui antarmuka sistem, kemudian sistem akan melakukan penyaringan terhadap data yang tersimpan di tabel TB_absensi. Setelah data yang relevan diperoleh, sistem mengeksekusi query dan memformat hasilnya agar siap ditampilkan pada dashboard. Selanjutnya, sistem memberikan opsi kepada pengguna untuk mengunduh data tersebut. Jika pengguna memilih untuk mengunduh, maka sistem akan membentuk file dalam format tetap (*fixed length*) dan menyediakannya untuk diunduh. Fitur ini dirancang untuk mendukung kebutuhan export data absensi ke sistem lain, seperti sistem payroll atau laporan internal HR, dengan format yang konsisten dan mudah diproses lebih lanjut.

B. Schema Database

Bagian ini menjelaskan struktur basis data yang digunakan dalam sistem pengolahan data absensi. Skema database dirancang untuk mendukung alur pemrosesan data absensi mulai dari data mentah yang diambil dari mesin hingga menjadi data yang siap dikirim ke sistem eksternal seperti Odoo. Setiap tabel memiliki peran masing-masing dan saling terhubung melalui relasi yang telah dirancang secara sistematis untuk memastikan integritas dan efisiensi proses. Relasi antar tabel dalam sistem ini ditunjukkan pada gambar 3.7.

U I M N
U N I V E R S I T A S
M U L T I M E D I A
N U S A N T A R A



Gambar 3.7. Schema Database Sistem Pengolahan Data Absensi

- **Raw Absensi**

Merupakan entitas yang menyimpan data mentah hasil pencatatan absensi dari mesin, seperti nomor pegawai, tanggal, jam, status (masuk atau keluar), serta terminal yang digunakan. Data ini belum diproses dan digunakan sebagai input awal untuk menghasilkan data absensi yang valid. Entitas ini tidak memiliki relasi langsung dengan entitas lain.

- **Absensi**

Menyimpan data absensi yang telah diproses dari entitas *Raw Absensi*. Berisi informasi seperti tanggal masuk dan keluar, jam masuk dan pulang, lokasi absensi, serta terminal yang digunakan. Entitas ini memiliki relasi many-to-one dengan *Mapping* dan *Terminal*.

- **Mapping**

Berfungsi untuk menghubungkan *id_employee* internal dengan *NIK* karyawan. Entitas ini memiliki relasi one-to-one dengan entitas *ms_niktelp*.

- **ms_niktelp**

Menyimpan data master karyawan seperti *NIK* dan area kerja (*PersArea*). Relasi yang terbentuk bersifat one-to-one dengan entitas *Mapping* berdasarkan

NIK, dan one-to-one dengan entitas *Setting* berdasarkan *PersArea*.

- **Terminal**

Merupakan entitas yang menyimpan informasi perangkat absensi seperti kode dan nama terminal. Relasi yang terbentuk bersifat one-to-many terhadap entitas *Absensi*, baik sebagai terminal masuk maupun keluar.

- **Setting**

Digunakan untuk menyimpan pengaturan jam kerja berdasarkan area kerja (*PersArea*), seperti jam pulang standar. Entitas ini memiliki relasi one-to-one dengan *ms_niktelp*.

- **Format Salah**

Menyimpan format data absensi yang tidak sesuai atau tidak dapat diproses oleh sistem. Entitas ini berdiri sendiri dan tidak memiliki relasi langsung dengan entitas lain.

- **Config**

Menyimpan konfigurasi global sistem dalam bentuk pasangan *key-value*. Entitas ini digunakan sebagai parameter sistem dan tidak memiliki relasi dengan entitas lain.

C. Potongan Code

Potongan-potongan kode berikut merupakan bagian dari implementasi sistem absensi yang telah dibangun selama kegiatan magang. Kode ini mencakup proses parsing dan validasi data absensi mentah dari file *cardraw.txt*, mekanisme pemantauan otomatis terhadap file oleh sistem, tampilan antarmuka pengguna untuk monitoring progres pemrosesan, serta prosedur tersimpan (*stored procedure*) yang digunakan untuk mengelola dan menyimpan data ke dalam struktur tabel absensi yang telah ditentukan di dalam basis data.

Potongan kode 3.1 merupakan fungsi yang digunakan untuk membaca dan mem-parsing isi file *cardraw.txt*, lalu menyimpan data yang valid ke dalam list data.

```
1 # Proses file
2 with open(file_path, "r", encoding="utf-8") as file:
3     for line in file:
4         line = line.strip()
5         if len(line) < 23:
6             insert_incorrect_format_data(line)
7         continue
```

```

8
9     try:
10         year, month, day = line[0:4], line[4:6], line[6:8]
11         hour, minute = line[8:10], line[10:12]
12         pernr = line[12:18]
13         status = int(line[18:19])
14         terminal = line[19:22]
15
16         # Validasi komponen tanggal
17         datetime.strptime(f"{year}-{month}-{day}", "%Y-%m-%d")
18         datetime.strptime(f"{hour}:{minute}", "%H:%M")
19
20         # Validasi status
21         if status not in (0, 1):
22             raise ValueError("Invalid status")
23
24         # Validasi terminal
25         if not terminal.isdigit():
26             raise ValueError("Invalid terminal")
27
28         tanggal = datetime.strptime(
29             f"{year}-{month}-{day} {hour}:{minute}:00",
30             "%Y-%m-%d %H:%M:%S"
31         )
32
33         data.append((tanggal, pernr, status, terminal))
34
35         with processing_lock:
36             file_processing_stats[filename]['processed'] = len
37             (data)
38
39     except Exception as e:
40         insert_incorrect_format_data(line)
41         continue

```

Kode 3.1: Potongan kode program app.py

Potongan kode 3.2 merupakan fungsi yang digunakan untuk memantau file cardraw.txt secara otomatis setiap 5 detik, dan memprosesnya jika ditemukan.

```

1 def auto_upload_worker():
2     while True:
3         try:
4             # Proses file cardraw.txt setiap 5 detik

```

```

5         file_path = os.path.join(get_source_folder(), "carddraw
.txt")
6         if os.path.exists(file_path):
7             print(f"        File {file_path} ditemukan, mulai
diproses...")
8             process_carddraw_file(file_path)
9         else:
10            print("        Tidak ada file carddraw.txt ditemukan
, menunggu...")
11
12            time.sleep(5)
13
14        except Exception as e:
15            log_error(f"Error in auto_upload_worker: {str(e)}")
16            time.sleep(5)

```

Kode 3.2: Potongan kode program file_watcher.py

Potongan kode 3.3 merupakan kode yang digunakan untuk menampilkan animasi loading, progress bar, status keseluruhan, dan detail tiap file yang sedang diproses.

```

1 <!-- Loading Modal -->
2 <div class="modal fade" id="loadingModal" tabindex="-1" aria-
hidden="true" data-bs-backdrop="static" data-bs-keyboard="false
">
3     <div class="modal-dialog modal-dialog-centered modal-lg">
4         <div class="modal-content">
5             <div class="modal-body text-center">
6                 <div class="d-flex justify-content-center mb-3">
7                     <div class="spinner-border text-primary" role=
"status" style="width: 3rem; height: 3rem;">
8                         <span class="visually-hidden">Loading...</
span>
9
10                    </div>
11                </div>
12                <h4 class="mt-3">Memproses file...</h4>
13
14            <!-- Progress Summary -->
15            <div id="progressSummary" class="mt-4">
16                <div class="d-flex justify-content-between
align-items-center mb-2">
17                    <h5 class="mb-0">Progress Keseluruhan</h5>
18                    <span id="overallProgressText" class="

```

```

18     badge bg-primary">0/0 (0%)</span>
19         </div>
20         <div class="progress" style="height: 25px;">
21             <div id="overallProgressBar" class="
22 progress-bar progress-bar-striped progress-bar-animated"
23             role="progressbar" style="width: 0%">
24 </div>
25         </div>
26         <div class="d-flex justify-content-between mt
27 -2">
28             <small class="text-muted">Status: <span id
29 ="overallStatus">Memproses...</span></small>
30             <small class="text-muted">Berhasil: <span
31 id="successCount">0</span> | Gagal: <span id="failedCount">0</
32 span></small>
33             <small class="text-muted">Estimasi waktu
34 tersisa: <span id="remainingTime">-</span></small>
35         </div>
36     </div>
37
38     <!-- Individual File Progress -->
39     <div class="mt-4">
40         <h5>Detail File</h5>
41         <div id="fileProgressDetails" class="mt-3"
42 style="max-height: 300px; overflow-y: auto;">
43             <!-- Will be populated by JavaScript -->
44         </div>
45     </div>
46 </div>
47 </div>
48 </div>
49 </div>
50 </div>

```

Kode 3.3: Potongan kode index.html

Potongan kode 3.4 merupakan potongan kode stored procedure yang digunakan untuk memproses data absensi mentah dari tabel TB_rawabsensi yang belum diproses (status_proses = 0). Proses dimulai dengan membaca data satu per satu menggunakan CURSOR, lalu mencocokkan NIK (perrn) dengan TB_mapping untuk mendapatkan id_employee, serta mencocokkan noterminal dengan TB_terminal untuk mendapatkan id_terminal. Setelah itu, sistem mengambil pengaturan jam pulang (jam_pulang) dari tabel TB_setting, berdasarkan PersArea yang diambil dari ms_niktelp. Jam ini digunakan untuk mendeteksi shift malam jika

jam presensi lebih kecil atau sama dengan jam pulang, maka dianggap sebagai shift malam (`pindah_hari = 1`). Prosedur ini adalah langkah awal untuk mengolah data menjadi format yang lebih bersih dan siap disimpan di `TB_absensi`, dengan mempertimbangkan logika shift dan mapping karyawan.

```
1 ALTER PROCEDURE sp_proses_absensi
2 AS
3 BEGIN
4     SET NOCOUNT ON;
5
6     DECLARE
7         @id INT,
8         @pernr VARCHAR(20),
9         @tanggal DATE,
10        @jam TIME,
11        @status VARCHAR(5),
12        @noterminal VARCHAR(50),
13
14        -- Dari mapping dan setting
15        @id_employee INT,
16        @id_terminal INT,
17        @jam_setting TIME,
18        @pindah_hari BIT,
19
20        -- Data absensi
21        @tanggal_in DATE,
22        @tanggal_out DATE,
23        @jam_masuk VARCHAR(8),
24        @jam_pulang VARCHAR(8),
25        @jam_masuk_lama VARCHAR(8);
26
27     DECLARE cur CURSOR FOR
28     SELECT id, pernr, tanggal, CAST(jam AS TIME), status,
29     noterminal
30     FROM TB_rawabsensi
31     WHERE status_proses = 0
32     ORDER BY pernr, tanggal, jam;
33
34     OPEN cur;
35     FETCH NEXT FROM cur INTO @id, @pernr, @tanggal, @jam, @status,
36     @noterminal;
37
38     WHILE @@FETCH_STATUS = 0
```

```

37 BEGIN
38     -- 1. Reset dan ambil id_employee
39     SET @id_employee = NULL;
40
41     SELECT TOP 1 @id_employee = m.id_employee
42     FROM TB_mapping m
43     WHERE RTRIM(m.NIK) = RTRIM(@pernr);
44
45     -- 2. Ambil id_terminal
46     SELECT @id_terminal = t.id
47     FROM TB_terminal t WHERE t.code = @noterminal;
48
49     -- 3. Ambil jam_pulang shift berdasarkan PersArea
50     SELECT @jam_setting = s.jam_pulang
51     FROM PORTAL.dbo.ms_niktelp n
52     JOIN TB_setting s ON RTRIM(n.PersArea) = RTRIM(s.PersArea)
53     WHERE n.NIK = @pernr;
54
55     -- 4. Hitung pindah_hari (shift malam)
56     SET @pindah_hari = CASE
57         WHEN @jam <= @jam_setting THEN 1
58         ELSE 0
59     END;

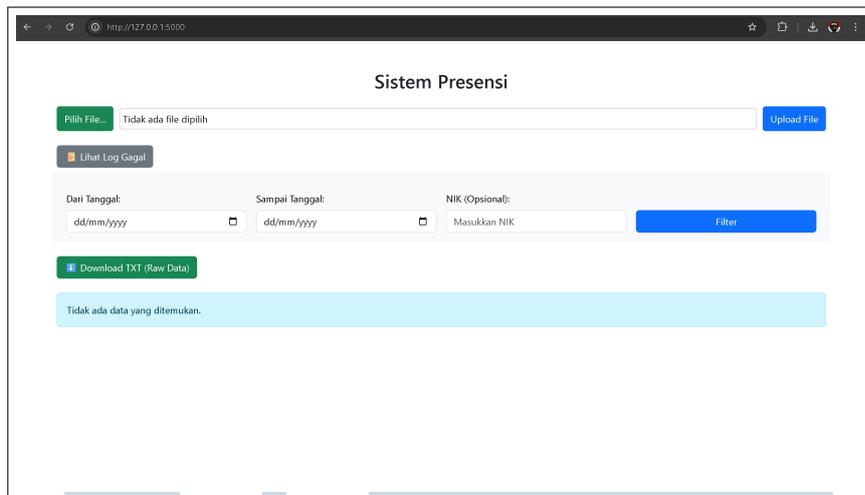
```

Kode 3.4: Potongan kode sp_proses_absensi

3.3.3 Tampilan Antarmuka Website

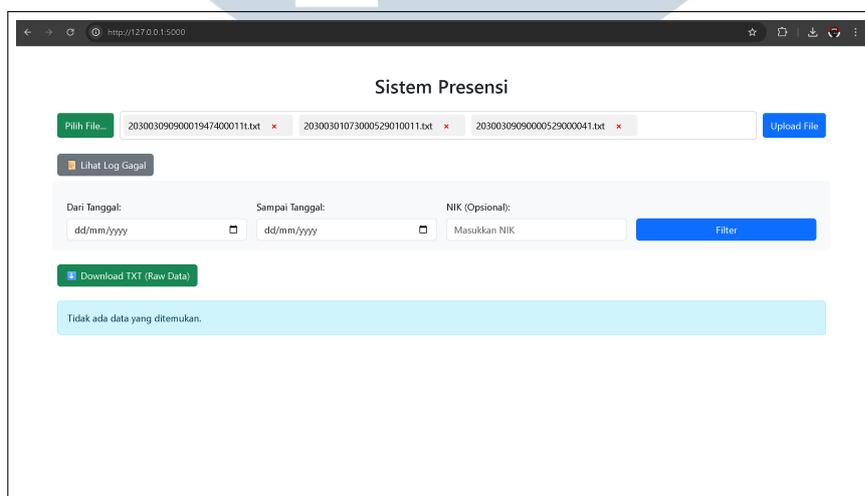
Tampilan antarmuka sistem absensi untuk mengolah data mentah mesin absensi mempunyai beberapa menu yang akan dijabarkan dalam gambar-gambar screenshoot berikut:

UNIVERSITAS
MULTIMEDIA
NUSANTARA



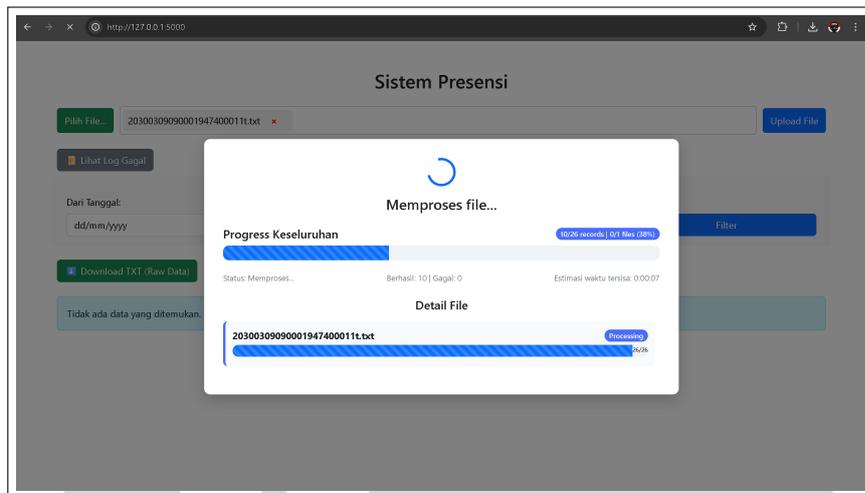
Gambar 3.8. Tampilan Awal Sistem.

Gambar 3.8 merupakan tampilan dashboard, pilih file, Upload File, Lihat Log Gagal, Filter berdasarkan tanggal dan NIK (opsional) serta Download TXT (Raw Data)



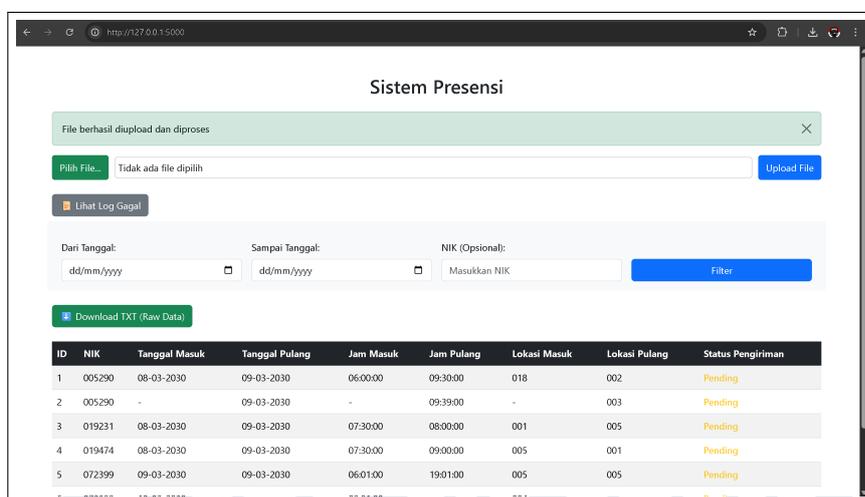
Gambar 3.9. Tampilan Sesudah Memilih File.

Gambar 3.9 merupakan tampilan kolom file yang sudah dipilih dan terdapat ikon silang untuk menghapus file yang tidak sengaja terpilih.



Gambar 3.10. Tampilan Progress Monitoring Proses Upload.

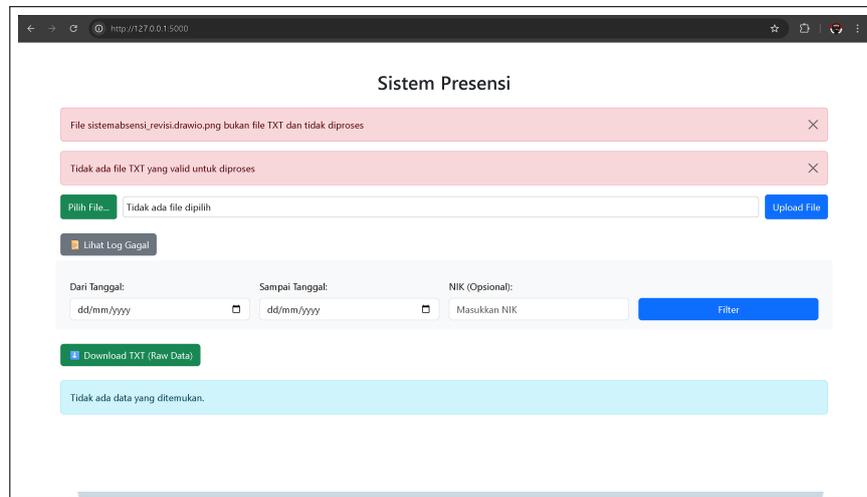
Gambar 3.10 merupakan tampilan proses pengiriman data, terdapat bar progress keseluruhan, status, jumlah data berhasil dan gagal serta jumlah record data yang diupload, jumlah file yang di proses, estimasi waktu dan detail file merupakan tampilan file yang lagi diproses dan terdapat bar, status, record untuk tiap file.



Gambar 3.11. Tampilan Setelah File Selesai Diproses.

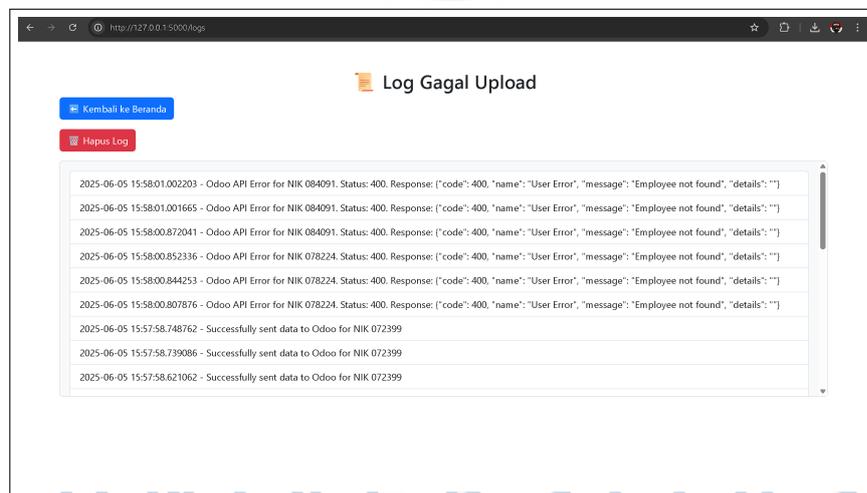
Gambar 3.11 merupakan tampilan data check-in dan check-out yang sudah diproses oleh sistem absensi dan stored procedure. Terdapat kolom ID, NIK, Tanggal Masuk, Tanggal Pulang, Jam Masuk, Jam Pulang, Lokasi Masuk, Lokasi Pulang dan Status Pengiriman ini merupakan beberapa bagian yang ditampilkan dari tabel TB_absensi dan juga terdapat pop up “File berhasil diupload dan

diproses”.



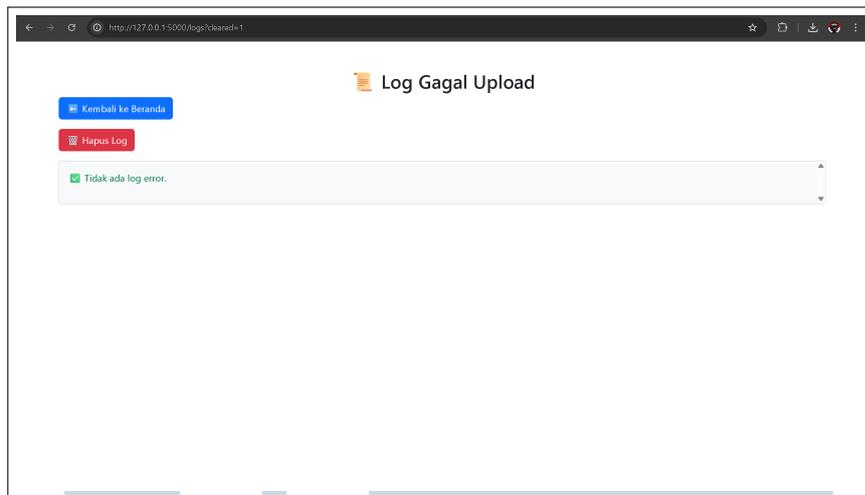
Gambar 3.12. Tampilan Pop-Up Data Gagal Upload.

Gambar 3.12 merupakan tampilan pesan error ketika sistem gagal memproses file karena format tidak sesuai (bukan TXT), dengan opsi untuk memilih file baru dan filter log error.



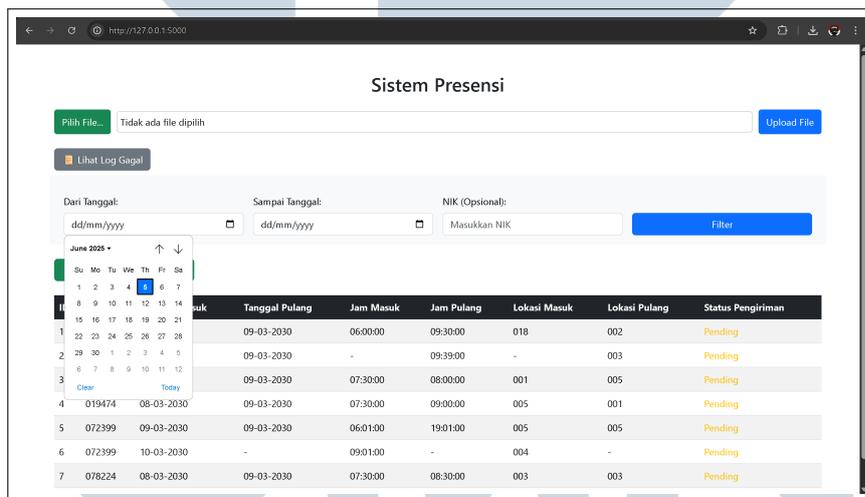
Gambar 3.13. Tampilan Log Error Data Gagal Upload.

Gambar 3.13 merupakan tampilan route Log Gagal Upload terdapat tombol untuk kembali ke Beranda, Hapus Log dan Log nya merupakan response dari odoo.



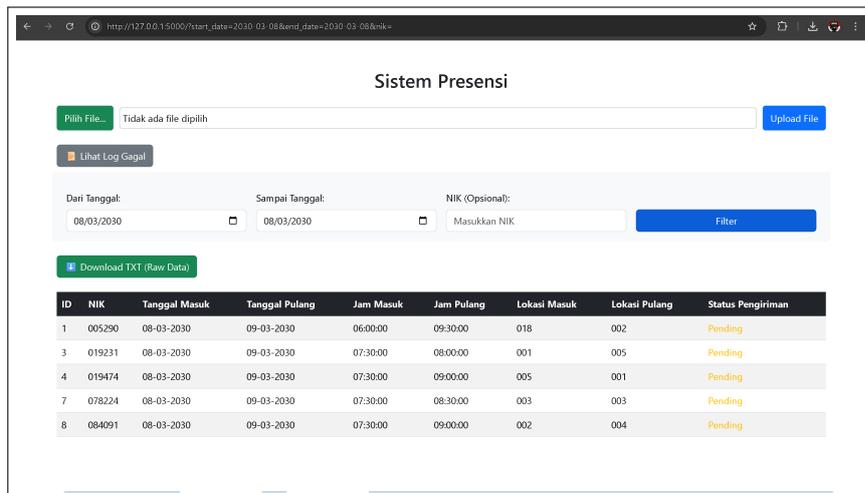
Gambar 3.14. Tampilan Setelah Log Dihapus.

Gambar 3.14 merupakan tampilan Log Gagal Upload yang sudah di hapus menggunakan tombol Hapus Log.



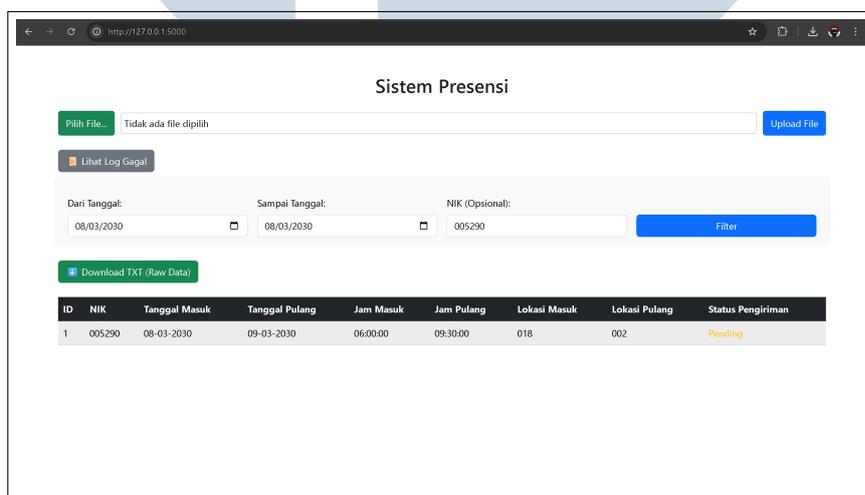
Gambar 3.15. Tampilan Filter Berdasarkan Tanggal.

Gambar 3.15 merupakan tampilan untuk memilih tanggal, dari tanggal berapa sampai tanggal berapa yang ingin di filter.



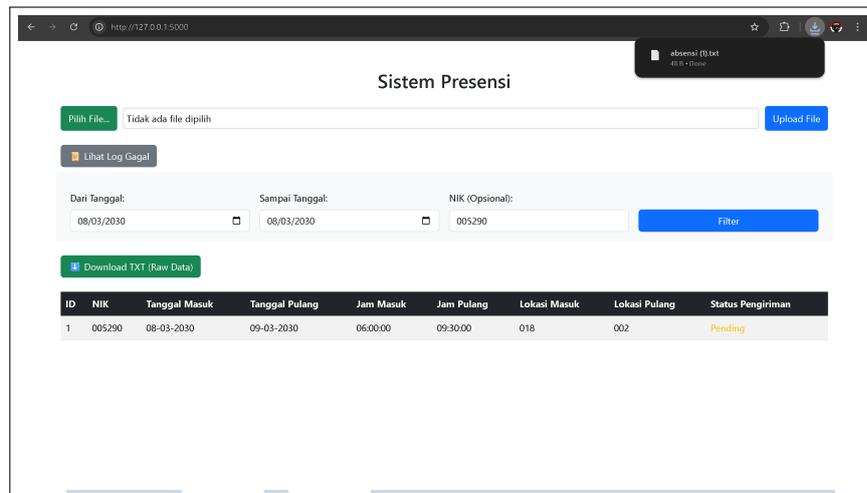
Gambar 3.16. Tampilan Sesudah Filter Tanggal.

Gambar 3.16 merupakan tampilan sesudah melakukan Filter Tanggal. yang ditampilkan hanya yang sesuai periode tanggal yang dipilih.



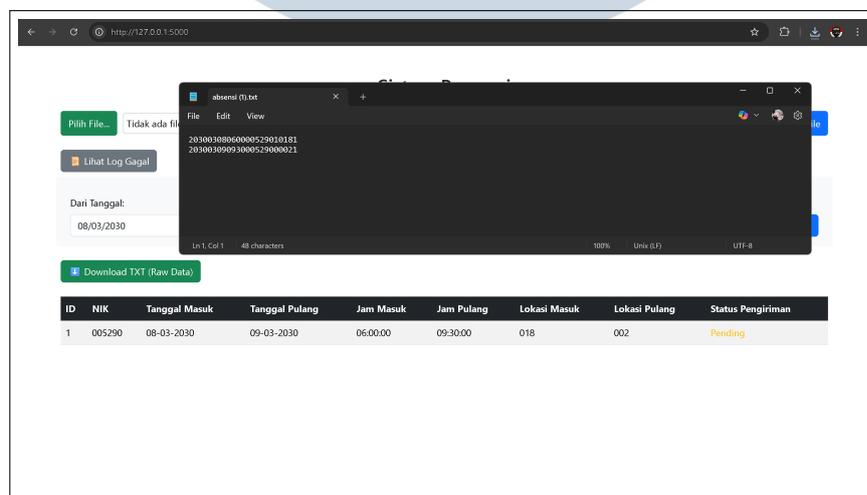
Gambar 3.17. Tampilan Sesudah filter Tanggal dan NIK.

Gambar 3.17 merupakan tampilan jika menggunakan Filter Tanggal dan NIK serta yang ditampilkan hanya periode Tanggal dan NIK yang di pilih.



Gambar 3.18. Tampilan Setelah Menekan Tombol Download.

Gambar 3.18 merupakan tampilan sesudah Download TXT (RawData). Karena sudah memilih periode Tanggal dan NIK maka yang terdownload hanya yang masuk periode Tanggal dan NIK, kalo tidak memilih periode Tanggal atau NIK maka semua data yang diupload akan terdownload semua.



Gambar 3.19. Tampilan Isi File Hasil Download.

Gambar 3.19 merupakan tampilan data Raw yang sudah didownload. Jadi, ketika ada data yang salah bisa didownload dan diperbaiki secara manual.

3.4 Kendala

Selama menjalani kerja magang, terdapat beberapa kendala yang muncul dalam proses pengembangan sistem pengolahan data mentah mesin absensi, antara

lain:

1. Sering terjadi perubahan dan penyesuaian terhadap alur atau fungsi sistem yang telah dibangun sebelumnya. Hal ini menyebabkan proses pengerjaan menjadi lebih lama karena perlu dilakukan revisi atau penyesuaian ulang terhadap bagian yang sudah selesai.
2. Pengalaman pertama dalam menggunakan SQL Server, baik dari segi antarmuka, pengelolaan database, maupun konfigurasi koneksi dengan Visual Studio Code, sehingga membutuhkan waktu tambahan untuk mempelajarinya.

3.5 Solusi

Untuk mengatasi kendala-kendala tersebut, solusi yang diterapkan selama pelaksanaan proyek magang antara lain:

1. Melakukan *meeting progress* secara rutin bersama pembimbing magang untuk membahas setiap perubahan sistem dan memastikan arah pengembangan tetap sesuai kebutuhan. Diskusi ini membantu memperjelas langkah-langkah yang harus diambil dan mempercepat proses penyesuaian.
2. Melaksanakan pembelajaran mandiri secara bertahap melalui dokumentasi resmi, video tutorial, dan forum diskusi online, sehingga pemahaman mengenai SQL Server dan integrasinya dengan Visual Studio Code meningkat secara bertahap.

U M N
U N I V E R S I T A S
M U L T I M E D I A
N U S A N T A R A