

BAB 3 PELAKSANAAN KERJA MAGANG

3.1 Kedudukan dan Koordinasi

Selama melaksanakan kegiatan magang di PT Accelist Lentera Indonesia sebagai **Full Stack Developer Intern**. Dalam peran ini, pengembangan aplikasi internal dilakukan menggunakan teknologi utama seperti **ASP.NET** untuk *backend* dan web, **React Native** untuk pengembangan aplikasi mobile, serta **Azure SQL** sebagai basis data utama. Proyek yang terlibat, yaitu:

- **AIS Marketing**, sistem internal untuk manajemen data dan aktivitas pemasaran.
- **AIS Peminjaman Barang (AIS PAM)**, sistem pencatatan peminjaman barang dan aset perusahaan.

Setiap proyek dikembangkan oleh satu tim khusus yang terdiri dari:

- **1 Project Manager (PM)** yang memimpin proyek secara umum.
- **1 Lead Developer** yang mengawasi teknis implementasi.
- Beberapa **Senior Developer** yang memberikan arahan dan melakukan *code review*.
- **Business Analyst (BA) Intern** dan **Developer Intern** lainnya yang turut mendukung proses analisis dan pengembangan sistem.

Koordinasi dilakukan secara rutin dan terstruktur, meliputi:

- **Meeting mingguan setiap hari Senin dan Jumat** untuk menyampaikan progres, hambatan, dan rencana mingguan.
- **Laporan harian (EOD – End of Day)** yang disampaikan di grup komunikasi tim untuk melaporkan hasil kerja harian.
- Komunikasi aktif menggunakan **WhatsApp Group**, terutama untuk diskusi cepat, perintah mendesak, dan pengalihan tugas (*handover*) antar anggota tim.

- Mengikuti kegiatan *transfer knowledge* (TF) saat terjadi rotasi developer dalam proyek, khususnya untuk AIS Marketing.
- **Pengujian sistem dilakukan setiap dua minggu sekali** oleh Lead Developer, Project Manager, dan seluruh Business Analyst untuk memastikan bahwa sistem berjalan sesuai kebutuhan pengguna dan tidak terdapat regresi pada fitur yang telah dikembangkan.

Sebagai bagian dari dokumentasi dan transparansi kerja tim, setiap anggota diwajibkan mengisi dan memperbarui file Excel bersama yang berisi:

- **Report progress item**, yaitu status pengerjaan fitur atau modul tertentu dalam proyek.
- **Bug list**, daftar kendala atau error yang ditemukan selama proses pengembangan dan pengujian.

Dokumen Excel ini menjadi alat bantu penting dalam evaluasi harian, pelacakan isu teknis, serta referensi untuk kegiatan *handover* apabila terjadi rotasi personel di dalam tim. Hal ini menciptakan lingkungan kerja yang terdokumentasi dengan baik dan terkoordinasi secara efisien, bahkan di tengah dinamika pergantian proyek internal.

3.2 Tugas yang Dilakukan

Selama melaksanakan magang sebagai *Full Stack Developer Intern* di PT Accelist Lentera Indonesia, didapatkan berbagai tugas yang melibatkan pengembangan fitur baru, perbaikan bug, serta pengujian terhadap modul aplikasi internal. Tugas-tugas tersebut mencakup pekerjaan di sisi backend menggunakan **ASP.NET**, frontend mobile menggunakan **React Native**, serta pengelolaan data menggunakan **Azure SQL**. Seluruh progres pekerjaan dicatat dalam file Excel internal yang memuat daftar tugas dan bug list.

Selama menjalani program magang, didapatkan kepercayaan untuk terlibat langsung dalam proses pengembangan dua proyek utama internal perusahaan, yaitu **AIS PAM** dan **AIS Marketing**. Proyek **AIS PAM** dikembangkan berbasis **.NET Web**, sedangkan **AIS Marketing** dikembangkan menggunakan **React Native** untuk sisi mobile dan **ASP.NET** untuk backend-nya. Tugas yang dikerjakan mencakup

pengembangan frontend dan backend, sesuai dengan kebutuhan modul yang sedang dikembangkan.

Secara lebih spesifik, proyek AIS terdiri dari beberapa modul internal seperti Calender, Inbox, HC Admin, Asset Management, Task Management, dan Payment Request di sisi HR service. Sedangkan untuk versi mobile, terdiri dari modul Finance, Human Resource, General Affair, Business Pipeline, dan Management Board. Pada pengembangan website internal berbasis .NET, ditugaskan secara khusus mengerjakan modul **Asset Management** yang mencakup fitur *Borrow Request* barang. Sementara itu, pada aplikasi mobile berbasis React Native, ditugaskan untuk mengerjakan modul **Business**, terutama pada bagian *Pipeline Project*, mulai dari sisi frontend hingga backend-nya.

Fitur-fitur yang dikembangkan antara lain mencakup tampilan daftar proyek pipeline, form untuk input data proyek baru, serta tombol aksi seperti edit dan delete. Di sisi Asset Management, dikembangkan fitur untuk request peminjaman barang, form input permintaan, dan tampilan status pengembalian serta notifikasi pengembalian barang. Seluruh fitur ini dikembangkan secara modular dan sudah terintegrasi dengan backend service yang stabil menggunakan Web API.

Berikut merupakan tugas yang dikerjakan dan penambahan fitur pada website

dan aplikasi AIS (Accelist Information System):

Frontend (Tampilan Pengguna):

- AIS Marketing Mobile – List Screen
- AIS Marketing Mobile – Form AvtPipeline
- AIS Marketing Mobile – Project Run Status Detail
- AIS PAM Web – Modal Notifikasi
- AIS PAM Web – Form Input
- AIS PAM Web – Borrow List Screen

Backend (Layanan API):

- GET, POST, DELETE, dan PUT untuk data proyek PipelineAvt
- GET, POST, DELETE, dan PUT untuk Project Run Status Detail

- Pembuatan endpoint notifikasi pengingat (reminder notification)
- GET, POST, DELETE, dan PUT untuk AssetManagement
- Endpoint sorting untuk data AssetManagement

Sebagai catatan tambahan, sebelum tahap perancangan sisi backend dimulai, dilakukan terlebih dahulu proses sinkronisasi dengan repositori GitHub yang telah tersedia. Karena proyek sudah memiliki struktur branch untuk masing-masing modul seperti Business Pipeline dan Peminjaman Barang, dilakukan git pull pada branch terkait. Selanjutnya, konfigurasi koneksi dengan PostgreSQL disiapkan untuk membangun *model*, *controller*, dan *service*. Setelah sisi backend terselesaikan, pengembangan frontend dilanjutkan untuk .NET Web pada modul Asset Management dan React Native untuk Pipeline Project di modul Business. Seluruh proses dilaksanakan secara berurutan dan mengacu pada praktik pengembangan sistem berbasis layanan yang terstruktur serta didukung dokumentasi yang baik.

3.3 Uraian Pelaksanaan Magang

Bagian ini menjelaskan aktivitas kerja magang selama enam belas minggu yang dilakukan di PT Accelist Lentera Indonesia, dengan rincian kegiatan setiap minggunya sebagaimana ditampilkan pada Tabel 3.1, Tabel 3.2, Tabel 3.3, dan Tabel 3.4 berikut.



Tabel 3.1. Uraian Pelaksanaan Kerja Magang Minggu 1

Minggu	Uraian Kegiatan
1	<ul style="list-style-type: none"> • Pembuatan frontend tugas individual website ticket menggunakan Next.js • Pembuatan backend tugas individual website ticket menggunakan .NET • Bug fixing pada aplikasi Next.js • Database migration (<i>code first</i>) ke Azure SQL dari .NET Entity Framework Core



Tabel 3.2. Uraian Pelaksanaan Kerja Magang Minggu 2–5

Minggu	Uraian Kegiatan
2	<ul style="list-style-type: none"> • Pembagian kelompok dan perencanaan klon Tokopedia (web dan mobile) • Pembuatan backend API CRUD untuk Order dan Cart • CRUD API untuk input dan hapus item
3	<ul style="list-style-type: none"> • Pembuatan frontend dengan Next.js dan React Native • Pembuatan frontend Cart • Pembuatan form input barang terhubung API
4	<ul style="list-style-type: none"> • Mulai masuk ke proyek internal AIS Human Resource • Nonton video transfer knowledge dan pengenalan tugas • Pembuatan backend item Business Pipeline (service)
5	<ul style="list-style-type: none"> • Penyelesaian backend AIS Marketing • Pembuatan model untuk proses data • Pembuatan PipelineAvtService • Pembuatan PipelineAviationController

Tabel 3.3. Uraian Pelaksanaan Kerja Magang Minggu 6–10

Minggu	Uraian Kegiatan
6	<ul style="list-style-type: none"> • Mulai menangani AIS PAM • Video transfer knowledge AIS PAM • Pengembangan backend Asset Management
7	<ul style="list-style-type: none"> • Penyelesaian backend AIS PAM • Pembuatan model data • Pembuatan <code>AssetManagementService</code> dan <code>AssetManagementController</code>
8	<ul style="list-style-type: none"> • Pengembangan <code>BorrowReturnHistoryService</code> • Pembuatan modal notifikasi pengembalian barang
9	<ul style="list-style-type: none"> • Pembuatan frontend AIS Marketing (React Native) • Pembuatan modal Project Status
10	<ul style="list-style-type: none"> • Pembuatan screen <code>PipelineAvt</code> • Form <code>CreateEditPipelineAvtForm</code> terhubung ke API • Pembuatan constraint untuk penerimaan data

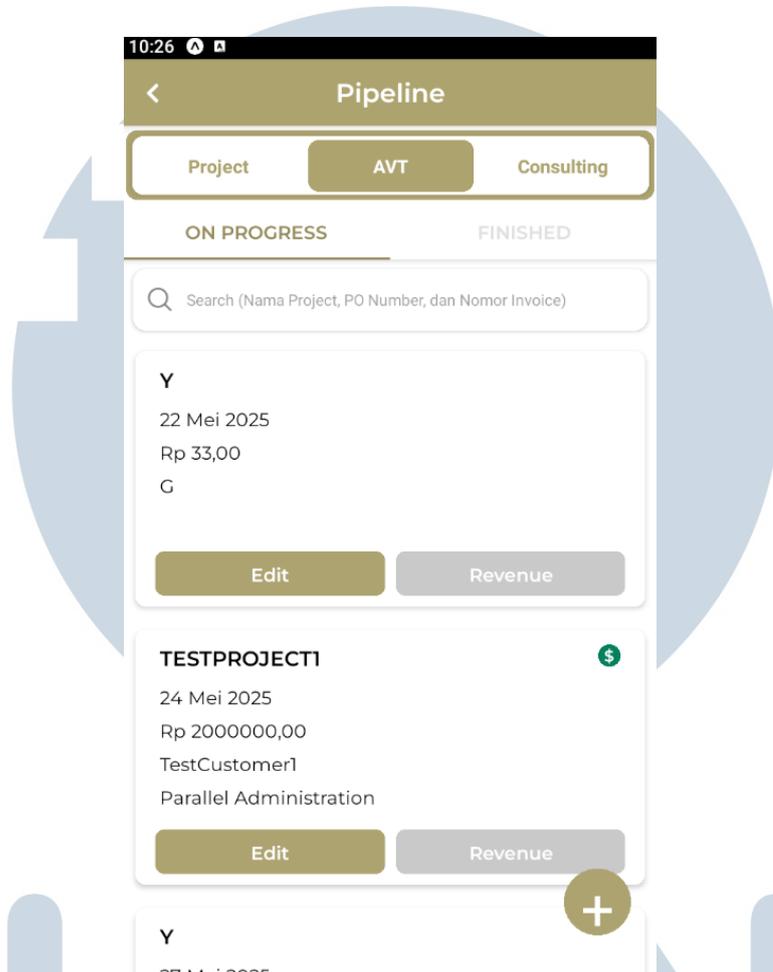
Tabel 3.4. Uraian Pelaksanaan Kerja Magang Minggu 11–16

Minggu	Uraian Kegiatan
11	<ul style="list-style-type: none"> • Bug fixing form CreateEditPipelineAvtForm • Penyesuaian tampilan dan validasi data input
12	<ul style="list-style-type: none"> • Pembuatan tabel Excel (React Native) untuk daftar proyek • Bug fixing impor Excel dan highlight data
13–14	<ul style="list-style-type: none"> • Implementasi modal notifikasi • Pembuatan list dan form BorrowAsset
15	<ul style="list-style-type: none"> • Bug fixing AIS Marketing dan AIS PAM
16	<ul style="list-style-type: none"> • Minor adjustment: sort/get PipelineAvt by date • Handover dan penilaian ke lead developer

3.3.1 AIS Marketing

Tugas-tugas yang dikerjakan dalam pengembangan modul **AIS Marketing** mencakup beberapa bagian *frontend* dan *backend*, antara lain:

A. AIS Marketing Mobile – List Screen



Gambar 3.1. Tampilan AIS Marketing Mobile – List Screen

Gambar 3.1 Gambaran *PipelineScreen* yang sudah berada di halaman AVT yang menampilkan list hasil pembuatan laporan keuangan tiap proyek.

UNIVERSITAS
MULTIMEDIA
NUSANTARA

B. AIS Marketing Mobile – Form Screen

9:41

< Create Revenue

Date
5 April 2023

Customer Name *

Select Item

Customer name is required

Project/Product Name *

Project name is required

PM Customer *

PM Customer is required

PM Accelist *

PM Accelist is required

Nilai Project *

Nilai Project is required

Purchase Order Date

Planned* Actual

PO Number

Include PPN

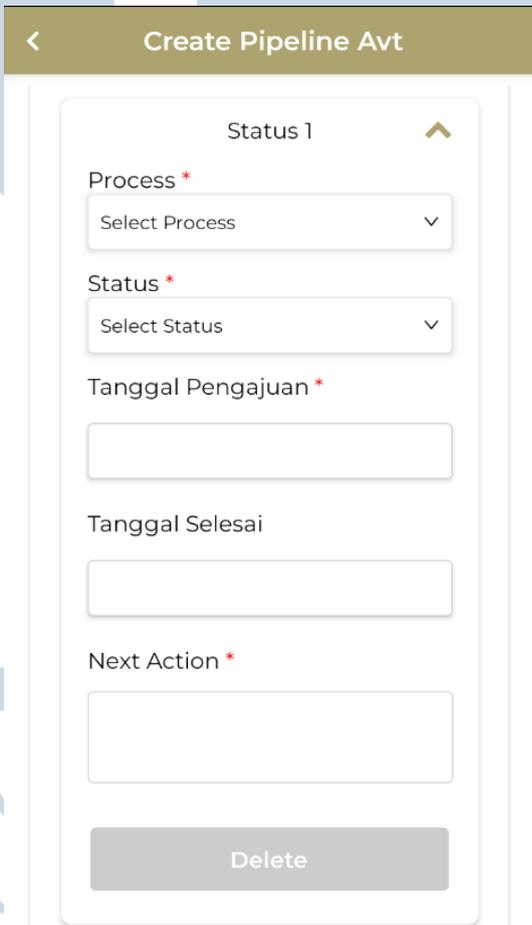
Include PPH

Gambar 3.2. Tampilan AIS Marketing Mobile – List Screen

Form ini digunakan untuk mencatat pemasukan laporan keuangan dari proyek yang terdaftar dalam pipeline AIS Marketing. Pengguna diminta mengisi berbagai informasi penting, seperti nama pelanggan yang dipilih melalui dropdown, nama

proyek atau produk, nama Project Manager dari pihak customer dan Accelist, serta nilai proyek yang dikerjakan. Selain itu, form ini juga menyediakan input untuk tanggal purchase order yang terbagi menjadi *Planned* dan *Actual*, serta kolom untuk mengisi nomor PO. Di bagian bawah, tersedia checkbox untuk menyertakan perhitungan pajak PPN dan PPH. *Form* ini telah dilengkapi dengan validasi wajib (*required*) yang memberikan peringatan jika ada field yang belum diisi. Seluruh data kemudian dikirim ke server melalui metode POST untuk dicatat sebagai revenue resmi proyek.

C. AIS Marketing Mobile – Project Run Status Detail



The screenshot displays a mobile application interface for creating a pipeline. The title bar at the top is olive green with a back arrow and the text "Create Pipeline Avt". Below the title bar, the form is titled "Status 1" with an upward arrow icon. The form contains five input fields, each with a red asterisk indicating a required field:

- Process ***: A dropdown menu with the placeholder text "Select Process" and a downward arrow.
- Status ***: A dropdown menu with the placeholder text "Select Status" and a downward arrow.
- Tanggal Pengajuan ***: A text input field.
- Tanggal Selesai**: A text input field.
- Next Action ***: A text input field.

At the bottom of the form, there is a grey button labeled "Delete". The background of the image features a large, faint watermark of the Universitas Multimedia Nusantara logo.

Gambar 3.3. Tampilan AIS Marketing Mobile – List Screen

Komponen ini merupakan bagian akhir dari proses input proyek yang berfungsi untuk mencatat dan memantau status perkembangan proyek secara

bertahap. Pengguna dapat menambahkan satu atau beberapa status perkembangan (*status detail*) yang mencerminkan tahapan pengerjaan, seperti *Quotation Stage*, *Negotiation Stage*, atau *Purchasing Stage*. Setiap status dilengkapi dengan informasi tanggal dan keterangan yang menjelaskan progres secara rinci.

D. Model CreateEditPipelineAvtRequestModel

Potongan kode `CreateEditPipelineAvtRequestModel` merupakan model data yang digunakan untuk merepresentasikan informasi proyek pipeline yang akan dibuat atau diedit dalam modul AIS Marketing. Model ini dikirim dari sisi klien (*frontend*) ke server sebagai bagian dari permintaan pembuatan atau pembaruan data pipeline.

Model ini memiliki beberapa properti penting, antara lain `PipelineId` bertipe `Guid?` yang bersifat opsional dan digunakan saat melakukan edit data. Properti `CustomerName` menyimpan nama pelanggan, sedangkan `EstValueProject` merepresentasikan estimasi nilai proyek dalam bentuk angka desimal. Tanggal mulai dan selesai proyek diwakili oleh `ProjectStartDate` dan `ProjectEndDate` bertipe `DateTime`. Model ini dapat diperluas dengan properti tambahan sesuai kebutuhan sistem.

```
1 public class CreateEditPipelineAvtRequestModel
2 {
3     public Guid? PipelineId { get; set; }
4     public int PipelineTypeId { get; set; }
5     [Required]
6     public Guid AccountManagerId { get; set; }
7     [Required]
8     public string CustomerName { get; set; }
9     [Required]
10    public string OptyProject { get; set; }
11    public string PmCustomer { get; set; }
12    public string PmAccelistEmployeeId { get; set; }
13    public decimal? EstValueProject { get; set; }
14    [Required]
15    public DateTime EstPoDate { get; set; }
16    [Required]
17    public DateTime ProjectStartDate { get; set; }
18    [Required]
19    public DateTime ProjectEndDate { get; set; }
```

```

20     public int PipelineResultId { get; set; }
21     public DateTime? ResultDate { get; set; }
22     public string Reason { get; set; }
23     [Required]
24     public List<CreateEditPipelineProcessModel> PipelineProcesses
25     { get; set; }

```

Kode 3.1: Model CreateEditPipelineAvtRequestModel

Bagian ini menjelaskan fungsi-fungsi utama dalam PipelineAvtService yang digunakan untuk mengelola data proyek pipeline pada sistem AIS Marketing.

E1. Mengambil Detail Proyek Berdasarkan ID

Fungsi ini digunakan untuk mengambil data lengkap sebuah proyek berdasarkan ID-nya. Data ini ditampilkan di form edit proyek agar pengguna dapat memperbarui data yang sudah ada.

```

1 [HttpGet]
2 public async Task<ActionResult<CreateEditPipelineAvtRequestModel>>
3     GetPipelineDetail([FromQuery] Guid pipelineId)
4 {
5     var pipeline = await _pipelineAvtService.GetPipelineDetail(
6     pipelineId);
7     return Ok(pipeline);
8 }

```

E2. Menampilkan Daftar Prioritas Proyek

Fungsi ini digunakan untuk mengambil daftar prioritas proyek seperti High, Medium, atau Low, yang digunakan untuk pengelompokan proyek. Data ini biasanya ditampilkan dalam bentuk dropdown di halaman form input.

```

1 [HttpGet("GetPipelinePriorities")]
2 public async Task<ActionResult<List<PipelinePriorityViewModel>>>
3     GetAllPipelinePriorityAsync()
4 {
5     var pipelinePriorities = await _pipelineAvtService.
6     GetPipelinePrioritiesAsync();
7     return Ok(pipelinePriorities);
8 }

```

E3. Menambahkan Proyek Baru ke Sistem

Fungsi ini digunakan untuk membuat proyek baru ke dalam sistem. Pengguna mengisi form proyek, dan data dikirim ke server melalui fungsi ini. Jika data valid, proyek akan ditambahkan dan ditampilkan di halaman list proyek.

```
1 [HttpPost("Create")]
2 public async Task<ActionResult> CreateAsync([FromBody]
   CreateEditPipelineAvtRequestModel createForm)
3 {
4     if (ModelState.IsValid == false)
5     {
6         return BadRequest(ModelState);
7     }
8
9     var result = await this._pipelineAvtService.CreateAsync(
   createForm);
10
11    if (!result)
12    {
13        return BadRequest("Failed to create Pipeline Project");
14    }
15
16    return Ok("Successfully created Pipeline Project");
17 }
```

E4. Memperbarui Data Proyek yang Sudah Ada

Fungsi ini digunakan untuk memperbarui atau mengedit proyek pipeline yang sudah tersimpan di sistem. Sama seperti saat membuat proyek, data dikirim melalui form, lalu diproses di backend melalui fungsi ini.

```
1 [HttpPost("Update")]
2 public async Task<ActionResult> UpdateAsync([FromBody]
   CreateEditPipelineAvtRequestModel editForm)
3 {
4     if (ModelState.IsValid == false)
5     {
6         return BadRequest(ModelState);
7     }
8
```

```

9     var result = await this._pipelineAvtService.EditAsync(editForm
10    );
11
12    if (!result)
13    {
14        return BadRequest("Failed to edit Pipeline Avt");
15    }
16
17    return Ok("Successfully edited Pipeline Avt");
18 }

```

F. Controller untuk Proyek Pipeline (CreateEditPipelineAvtController)

Berikut adalah potongan kode controller yang digunakan untuk menangani request HTTP terkait proyek pipeline. Controller ini berfungsi sebagai penghubung antara client (frontend) dan service layer.

F1. Mengambil Detail Proyek Pipeline

Fungsi ini digunakan untuk mengambil informasi proyek berdasarkan pipelineId yang dikirim melalui parameter query. Data yang diambil akan digunakan untuk menampilkan atau mengedit proyek tertentu.

```

1 [HttpGet]
2 public async Task<ActionResult<CreateEditPipelineAvtRequestModel>>
3     GetPipelineDetail([FromQuery] Guid pipelineId)
4 {
5     var pipeline = await _pipelineAvtService.GetPipelineDetail(
6     pipelineId);
7     return Ok(pipeline);
8 }

```

F2. Mengambil Daftar Prioritas Pipeline

Endpoint ini mengambil data prioritas proyek untuk ditampilkan dalam dropdown pada halaman input proyek. Data ini berupa daftar seperti High, Medium,

dan Low yang digunakan sebagai parameter klasifikasi proyek.

```
1 [HttpGet("GetPipelinePriorities")]
2 public async Task<ActionResult<List<PipelinePriorityViewModel>>>
   GetAllPipelinePriorityAsync()
3 {
4     var pipelinePriorities = await _pipelineAvtService.
   GetPipelinePrioritiesAsync();
5     return Ok(pipelinePriorities);
6 }
```

F3. Menambahkan Proyek Baru

Fungsi ini menangani proses pembuatan data proyek pipeline baru. Data dikirim melalui body dalam bentuk `CreateEditPipelineAvtRequestModel`. Validasi dilakukan sebelum diteruskan ke service layer untuk diproses.

```
1 [HttpPost("Create")]
2 public async Task<ActionResult> CreateAsync([FromBody]
   CreateEditPipelineAvtRequestModel createForm)
3 {
4     if (ModelState.IsValid == false)
5     {
6         return BadRequest(ModelState);
7     }
8
9     var result = await this._pipelineAvtService.CreateAsync(
   createForm);
10
11    if (!result)
12    {
13        return BadRequest("Failed to create Pipeline Project");
14    }
15
16    return Ok("Successfully created Pipeline Project");
17 }
```

F4. Memperbarui Data Proyek yang Sudah Ada

Fungsi ini digunakan untuk memperbarui data proyek pipeline yang telah tersimpan di sistem. Data yang dikirim akan divalidasi terlebih dahulu, lalu diteruskan ke `_pipelineAvtService` untuk diproses.

```
1 [HttpPost("Update")]
2 public async Task<ActionResult> UpdateAsync([FromBody]
   CreateEditPipelineAvtRequestModel editForm)
3 {
4     if (ModelState.IsValid == false)
5     {
6         return BadRequest(ModelState);
7     }
8
9     var result = await this._pipelineAvtService.EditAsync(editForm
   );
10
11    if (!result)
12    {
13        return BadRequest("Failed to edit Pipeline Avt");
14    }
15
16    return Ok("Successfully edited Pipeline Avt");
17 }
```

3.3.2 Pengembangan AIS PAM

Pada proyek AIS PAM, penulis bertanggung jawab dalam pengembangan fitur peminjaman dan pengelolaan aset, baik dari sisi tampilan pengguna maupun API:

A. AIS PAM Web

Modul ini merupakan bagian dari sistem AIS PAM yang menampilkan riwayat peminjaman dan pengembalian aset oleh karyawan. Fitur ini penting untuk melacak pergerakan aset secara transparan, termasuk kondisi dan status approval-nya.

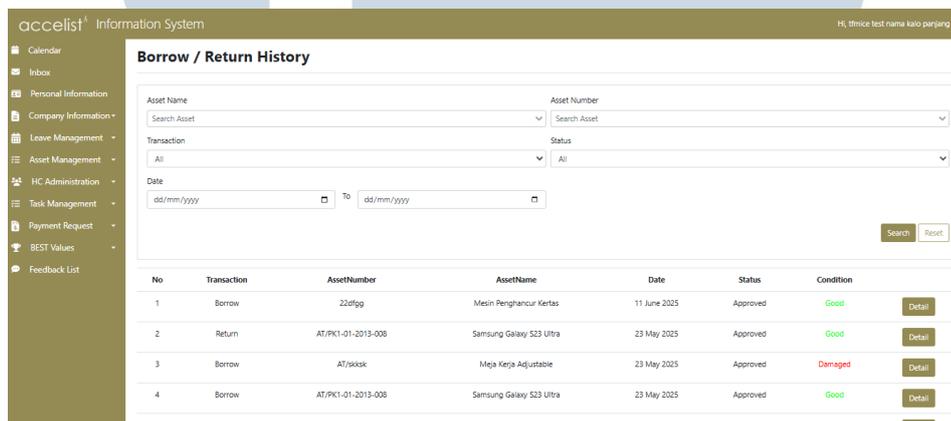
3.4 Pengembangan AIS PAM

Pada proyek **AIS PAM**, penulis bertanggung jawab dalam pengembangan fitur peminjaman dan pengelolaan aset, baik dari sisi tampilan pengguna maupun API:

A. AIS PAM Web

Modul ini merupakan bagian dari sistem AIS PAM yang menampilkan riwayat peminjaman dan pengembalian aset oleh karyawan. Fitur ini penting untuk melacak pergerakan aset secara transparan, termasuk kondisi dan status approval-nya.

A1. AIS PAM Web – Frontend



Gambar 3.4. Halaman Borrow / Return History pada AIS PAM Web

Gambar tersebut menunjukkan tampilan halaman *Borrow / Return History* pada sistem AIS PAM. Pengembangan antarmuka pengguna mencakup beberapa perbaikan dan penambahan fitur, antara lain:

- Perbaikan pada form filter agar data dapat dikirimkan dengan benar ke API backend sesuai parameter yang diminta.
- Penambahan fitur *sorting* berdasarkan tanggal peminjaman, menggantikan metode urutan default sebelumnya yang hanya berdasarkan nama atau ID.
- Implementasi highlight warna pada kolom *Condition* untuk menunjukkan kondisi aset secara visual, seperti *Good*, *Damaged*, dan lainnya.

Sebelumnya, tampilan tersebut belum dilengkapi dengan *form* khusus untuk melakukan penyortiran data, dan tabel masih disusun berdasarkan nama secara *default*. Oleh karena itu, fitur penyortiran berdasarkan tanggal peminjaman ditambahkan sesuai permintaan pengguna sistem, guna mempermudah pelacakan riwayat peminjaman secara kronologis. Ketidakhadiran opsi sortir manual sebelumnya dianggap menyulitkan dalam proses pencarian data berbasis waktu.

A2. AIS PAM Web – Backend

Untuk menangani pemrosesan data riwayat peminjaman dan pengembalian aset, dilakukan pengembangan pada sisi backend berupa implementasi metode `GetBorrowHistory`. Metode ini bertugas untuk mengambil data riwayat peminjaman dari database, dengan dukungan parameter penyaringan seperti ID transaksi, status aset, rentang tanggal, dan ID karyawan.

```
1 public async Task<PaginatedList<BorrowReturnHistoryModel>>
2   GetBorrowHistory(Guid? historyId,
3     string assetNumber,
4     string assetName,
5     int? transactionId,
6     int? assetStatusId,
7     int? currentPage,
8     DateTime? startDate,
9     DateTime? endDate,
10    string searchEmployeeId)
11 {
12   if (currentPage.HasValue == false || currentPage.Value < 1)
13   {
14     currentPage = 1;
15   }
16
17   var assetQuery = (
18     (from a in DB.Asset ...
19       select new BorrowReturnHistoryModel { ... })
20     .Union
21     (from a in DB.Asset ...
22       select new BorrowReturnHistoryModel { ... })
23   ).OrderByDescending(Q => Q.UpdatedAt);
```

```

24     var pageSize = 5;
25     var dataPaging = await PaginatedList<
    BorrowReturnHistoryModel>.CreateAsync(assetQuery, currentPage.
    Value, pageSize);
26     return dataPaging;
27 }

```

Kode 3.2: Metode GetBorrowHistory di Backend

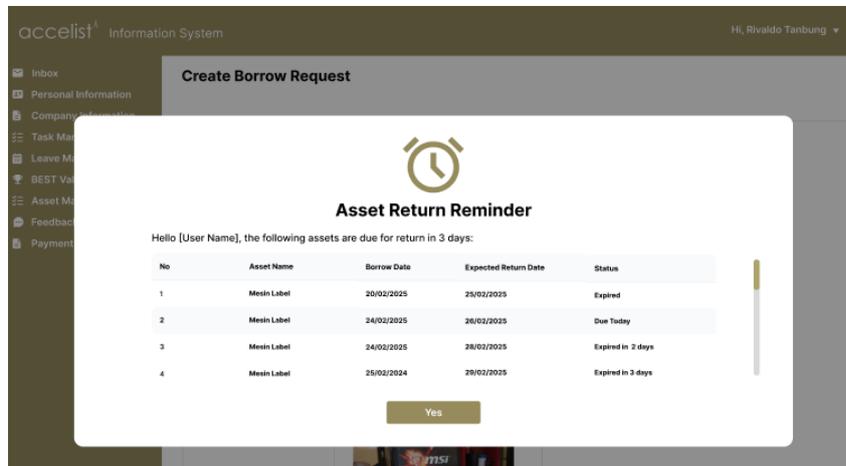
Sebelumnya, fitur penyortiran data belum tersedia di sisi backend, sehingga data riwayat hanya disusun berdasarkan nama aset secara default. Untuk meningkatkan fleksibilitas dan memenuhi kebutuhan pengguna, dilakukan pengembangan tambahan berupa logika penyortiran berdasarkan kolom waktu pembaruan (`UpdatedAt`) yang merepresentasikan tanggal peminjaman atau pengembalian aset.

Dengan adanya fitur ini, data yang dikirim ke frontend dapat ditampilkan secara kronologis, sehingga memudahkan pengguna dalam melacak transaksi terbaru maupun historis. Dukungan penyaringan berdasarkan tanggal melalui parameter `startDate` dan `endDate` juga menjadi bagian dari perbaikan ini untuk menunjang akurasi pencarian.

B. Pembuatan Fitur Notifikasi Pengembalian Aset

Karena sebelumnya sistem belum memiliki mekanisme pengingat pengembalian aset, maka dikembangkan fitur **Asset Return Reminder** agar pengguna maupun admin mendapatkan pemberitahuan aset apa saja yang akan jatuh tempo dalam waktu dekat (3 hari sebelum tanggal pengembalian). Fitur ini terdiri dari dua bagian: frontend dan backend.

B1. Frontend – Tampilan Notifikasi



Gambar: Tampilan Modal Notifikasi Pengembalian Aset

Pada sisi frontend, notifikasi ditampilkan dalam bentuk *popup modal* saat pengguna membuka halaman *Create Borrow Request*. Tampilan ini memuat daftar aset yang akan jatuh tempo dalam 3 hari ke depan. Informasi yang ditampilkan meliputi:

- Nama aset
- Tanggal peminjaman
- Tanggal pengembalian yang diharapkan
- Status jatuh tempo (misalnya: *Expired*, *Due Today*, *Due in 2 days*, dll.)

Desainnya sederhana dan informatif agar pengguna segera menyadari bahwa aset perlu segera dikembalikan.

B2. Backend – Logika Pengingat Pengembalian Aset

Untuk menghasilkan data notifikasi tersebut, dibuat metode `GetAdminReturnReminders()` yang berfungsi untuk mengecek data peminjaman aset yang masih aktif (belum dikembalikan) dan sudah disetujui. Metode ini akan memfilter transaksi yang memiliki tanggal pengembalian dalam rentang antara satu hari sebelum hingga tiga hari setelah tanggal hari ini. Jika memenuhi kriteria, data akan dimasukkan ke dalam daftar notifikasi, lengkap dengan informasi status

seperti keterlambatan (*Expired*) atau jumlah hari tersisa hingga jatuh tempo (*Due Today*, *Due in 2 days*, dan sebagainya).

```
1 public async Task<List<AdminBorrowReturnReminderModel>>
  GetAdminReturnReminders ()
2 {
3     var result = new List<AdminBorrowReturnReminderModel>();
4
5     var activeLoans = await DB.AssetLoanHistory
6         .Include(a => a.Asset)
7         .Join(
8             DB.Employee,
9             loan => loan.CreatedBy,
10            emp => emp.EmployeeId,
11            (loan, emp) => new { Loan = loan, Employee = emp }
12        )
13        .Where(item =>
14            item.Loan.ReturnedAt == null &&
15            item.Loan.ApprovedAt != null &&
16            item.Loan.ExpectedReturnDate != null)
17        .OrderBy(item => item.Loan.ExpectedReturnDate)
18        .ToListAsync();
19
20    var today = DateTime.Today;
21    int listNumber = 1;
22
23    for (int i = 0; i < activeLoans.Count; i++)
24    {
25        var item = activeLoans[i];
26        var loan = item.Loan;
27        var employee = item.Employee;
28        var expectedReturnDate = loan.ExpectedReturnDate?.Date;
29
30        if (expectedReturnDate.HasValue)
31        {
32            var daysRemaining = (expectedReturnDate.Value - today)
33                .Days;
34
35            if (daysRemaining <= 3 && daysRemaining >= -1)
36            {
37                string status;
38                if (daysRemaining < 0) status = "Expired";
39                else if (daysRemaining == 0) status = "Due Today";
40                else if (daysRemaining == 1) status = "Due
```

```

40     Tomorrow";
41         else if (daysRemaining == 2) status = "Due in 2
42     days";
43         else status = "Due in 3 days";
44
45     result.Add(new AdminBorrowReturnReminderModel
46     {
47         ListNumber = listNumber++,
48         RequesterName = employee.FullName,
49         AssetName = loan.Asset.Name ?? "Unknown Asset"
50     },
51         BorrowDate = loan.RequestedDate,
52         ExpectedReturnDate = loan.ExpectedReturnDate,
53         Status = status
54     });
55
56     return result;
57 }

```

Kode 3.3: Metode Backend GetAdminReturnReminders

3.5 Kendala dan Solusi yang Ditemukan

Selama proses pengerjaan proyek, dihadapi dengan berbagai kendala teknis maupun non-teknis yang memengaruhi efisiensi pengembangan dan kolaborasi tim. Beberapa kendala utama yang dihadapi antara lain:

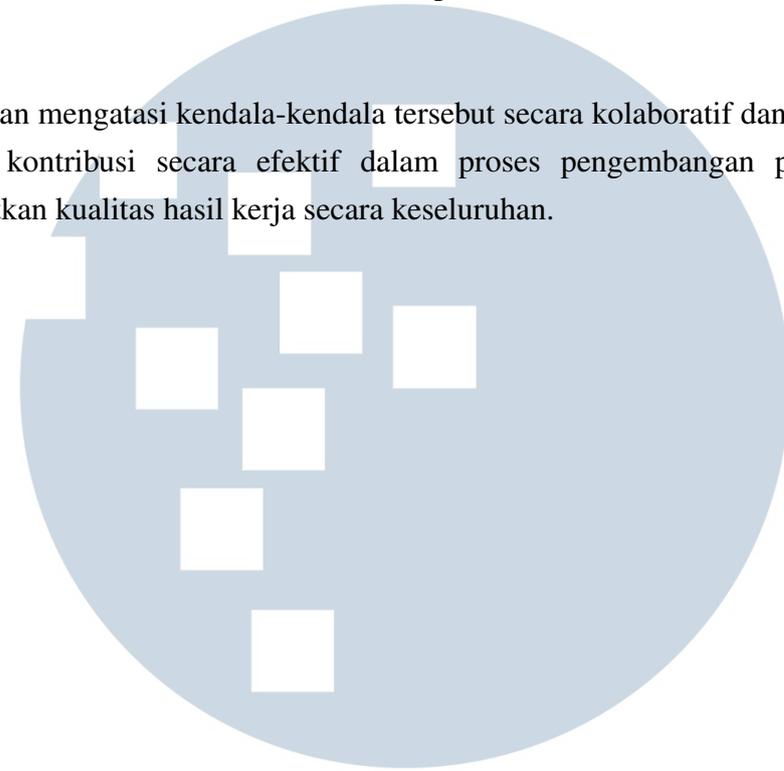
- **Perbedaan Versi Node.js dan Expo Go:** Pada proyek berbasis React Native, terjadi ketidaksesuaian versi Node.js dan Expo Go antara lingkungan pengembangan dengan dokumentasi proyek. Hal ini menyebabkan aplikasi tidak dapat dijalankan dengan semestinya. Solusinya adalah melakukan penyesuaian versi melalui pengaturan versi Node.js menggunakan `nvm` dan memperbarui Expo Go sesuai kebutuhan proyek.
- **Permasalahan Instalasi Dependencies:** Beberapa kendala muncul saat instalasi package, seperti konflik dependensi atau error saat `npm install`. Solusi yang diterapkan adalah menghapus `node_modules` dan

package-lock.json, kemudian melakukan instalasi ulang menggunakan versi yang sesuai.

- **Perbedaan Versi .NET:** Pada proyek berbasis .NET, terdapat perbedaan versi framework antara satu developer dengan developer lain, sehingga memunculkan error saat build atau menjalankan aplikasi. Solusinya adalah menyamakan versi SDK dan tools yang digunakan melalui dokumentasi internal proyek.
- **Komunikasi antar Business Analyst (BA) dan Developer:** Terkadang, BA belum melakukan pull terhadap kode terbaru dari developer, sehingga terjadi ketidaksesuaian saat pengujian. Hal ini diselesaikan dengan meningkatkan komunikasi dan koordinasi menggunakan platform seperti Discord dan grup WhatsApp serta melakukan *daily stand-up* secara rutin.
- **Komunikasi antar Developer dan Project Manager (PM):** Perbedaan persepsi terkait fitur atau logika bisnis terkadang memunculkan konflik. Solusi yang diambil adalah mengadakan diskusi klarifikasi bersama agar semua pihak memahami kebutuhan sistem dengan konsisten.
- **Kesulitan Menemukan File Tertentu:** Jumlah file yang sangat banyak dalam repositori menyulitkan pencarian file yang perlu diedit. Untuk mengatasi hal ini, digunakan fitur pencarian proyek di Visual Studio dan dibuat dokumentasi internal mengenai struktur folder serta file penting.
- **API dan Controller yang Belum Berfungsi:** Beberapa endpoint API masih belum berfungsi sebagaimana mestinya. Solusinya adalah melakukan debugging bersama tim backend dan memastikan controller telah terhubung dengan benar ke service layer dan basis data.
- **Koneksi Database yang Kurang Stabil:** Akses ke basis data yang dihosting secara remote kadang mengalami keterlambatan atau terputus, terutama saat proses migrasi atau pengujian. Solusi sementara yang diterapkan adalah menggunakan koneksi lokal terlebih dahulu untuk pengembangan, serta menjadwalkan akses database pada waktu dengan trafik minimal.
- **Koneksi VPN yang Sering Terputus:** Koneksi VPN yang tidak stabil menyebabkan gangguan dalam mengakses server internal dan menghambat integrasi sistem. Solusinya adalah mengatur ulang konfigurasi VPN dan

melakukan koneksi ulang secara berkala, serta memastikan bahwa koneksi internet utama berada dalam kondisi optimal.

Dengan mengatasi kendala-kendala tersebut secara kolaboratif dan sistematis, terjadilah kontribusi secara efektif dalam proses pengembangan proyek dan meningkatkan kualitas hasil kerja secara keseluruhan.



UMMN

UNIVERSITAS
MULTIMEDIA
NUSANTARA