

## **BAB III**

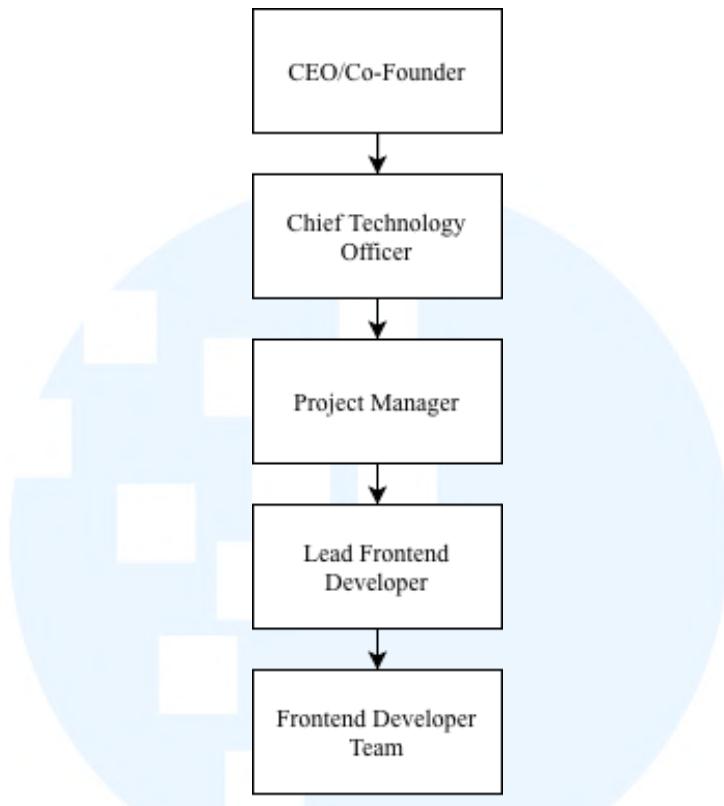
### **PELAKSANAAN KERJA**

#### **3.1 Kedudukan dan Koordinasi**

##### **3.1.1 Kedudukan**

CEO adalah *stakeholder* utama di MLS sekaligus pengawas tim operasional perusahaan. CTO berperan sebagai ahli dari teknologi yang dipakai perusahaan, menjadi pengembang dan pengelola MLS, serta memberikan panduan kepada tim operasional. PM bertanggung jawab atas pengelolaan proyek, pengawasan dan pembagian tugas kepada tim *UI/UX Designer*, *Frontend Developer*, dan *Backend Developer*, serta peninjauan hasil kerja tim operasional. *Lead Frontend Developer* bertugas koordinasi dan meninjau hasil kerja tim *Frontend Developer* dan mengembangkan *frontend*. Tim *Frontend Developer*, yang terdiri dari anggota senior dan *intern*, bertugas mengembangkan *Frontend* berdasarkan desain yang dibuat tim *UI/UX Designer*, dan mengintegrasikannya ke *Backend*. Hierarki kedudukan terlihat di Gambar 3.1. Penulis menempati posisi *Frontend Web Developer Intern* di PT. Minilemon Nusantara dan dibimbing oleh Muhammad Ghazariz, selaku CEO perusahaan.

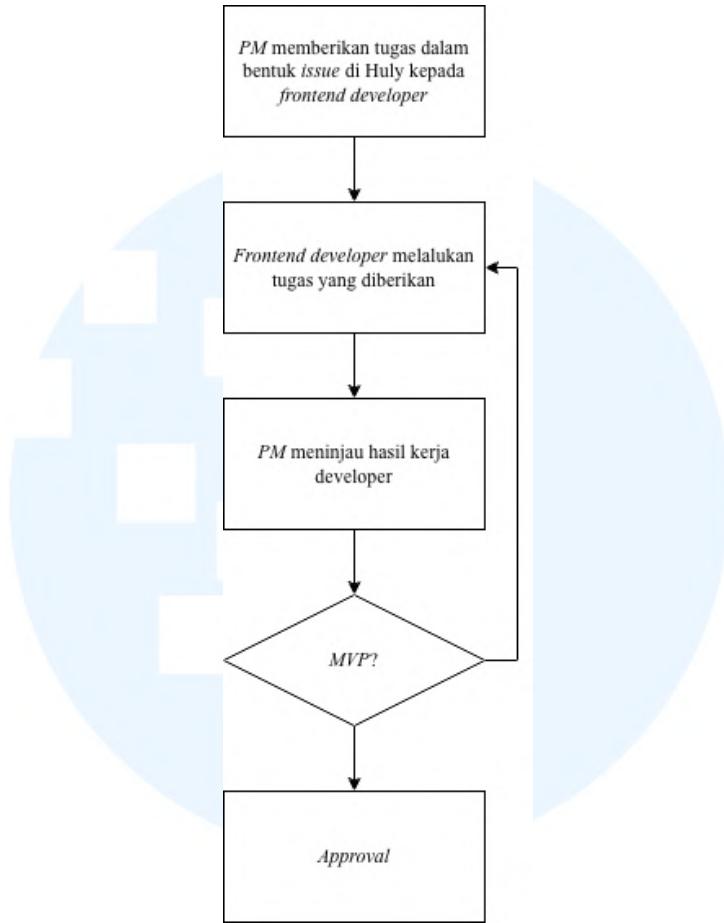




Gambar 3.1 Kedudukan penulis

### 3.1.2 Koordinasi

Gambar 3.2 mencerminkan alur kerja pengembangan MLS. Proses dimulai saat PM membagikan tugas dalam bentuk *issue* di Huly, platform manajemen proyek, ke tim *Frontend Developer*. Setiap tugas yang diberikan melingkup satu fitur di salah satu beranda di MLS (Admin, Mentor, atau Siswa). Setelah menerima tugas, *Frontend Developer* akan melakukan *slicing* antarmuka dan integrasi ke *backend* menggunakan API. Selama mengerjakan tugas, jika *Frontend Developer* memiliki kendala seperti *endpoint* API yang belum dibuat atau inkonsistensi dalam desain, *Developer* bisa membuat *issue* di Huly untuk mengkomunikasikan kendala ke tim yang bersangkutan. Setelah selesai mengerjakan tugas, *Developer* membuat *pull request* di GitHub dan PM dan *Lead* meninjau hasilnya dan *Developer* akan terus melakukan perbaikan selama hasilnya belum memenuhi kriteria MVP. Jika sudah memenuhi kriteria MVP, hasil kerjanya *merge* ke *branch* utama proyek, dan *Developer* melanjutkan penggerjaan tugas berikutnya. Alur ini diulang sampai seluruh fitur di MLS sudah memenuhi kriteria MVP.



Gambar 3.2 Alur koordinasi

### 3.2 Tugas yang Dilakukan

Tabel 3.1 Pekerjaan yang dilakukan

Minggu ke-	Keterangan
1	<ul style="list-style-type: none"> <li>Mempelajari <i>tech stack</i> (React, Tailwind CSS, Next.js) dan <i>tools</i> (GitHub, Huly) yang digunakan perusahaan.</li> </ul>
2	<ul style="list-style-type: none"> <li>Memahami desain di Figma dan struktur proyek.</li> </ul>
3-4	<ul style="list-style-type: none"> <li><i>Slicing</i> fitur Tugas Saya untuk Mentor.</li> </ul>
5-7	<ul style="list-style-type: none"> <li>Mempelajari Convex sebagai <i>backend</i>.</li> </ul>
8	<ul style="list-style-type: none"> <li><i>Request API</i> Tugas Saya untuk Mentor.</li> </ul>
9	<ul style="list-style-type: none"> <li><i>Slicing</i> fitur Video Tutorial untuk Mentor.</li> </ul>

10-11	<ul style="list-style-type: none"> <li>Mempelajari backend menggunakan Next.js dan Prisma.</li> <li>Menjadi <i>lead</i> untuk tim <i>frontend</i>.</li> <li><i>Slicing</i> fitur Modul Belajar untuk Mentor.</li> </ul>
12-13	<ul style="list-style-type: none"> <li><i>Slicing</i> fitur Modul Belajar untuk Siswa.</li> <li>Memperbarui tampilan <i>header</i>.</li> <li><i>Slicing</i> fitur <i>Ubah Program Studi</i> untuk Mentor.</li> </ul>
14	<ul style="list-style-type: none"> <li><i>Slicing</i> fitur Modul Belajar untuk Admin.</li> </ul>
15	<ul style="list-style-type: none"> <li><i>Slicing</i> fitur Tugas Saya untuk Siswa.</li> </ul>
16-17	<ul style="list-style-type: none"> <li>Integrasi API untuk fitur Modul Belajar Mentor.</li> </ul>

### 3.3 Uraian Pelaksanaan Kerja

#### 3.3.1 Proses Pelaksanaan

- **Ekplorasi proyek**

*Tech stack* yang digunakan untuk pengembangan MLS adalah React, Tailwind CSS, dan Next.js. Karena penulis sudah memiliki pengalaman menggunakan *tech stack* tersebut, penulis dapat beradaptasi dengan cepat di proyek ini.

*Tools* yang dipakai untuk komunikasi antar tim seputar projek MLS dilakukan di GitHub dan Huly. GitHub digunakan untuk mempermudah kolaborasi dengan *developer* lain dalam satu *repository*. *Repository* MLS terdiri dari beberapa jenis *branch*:

1. *main*, yang berisi kode *production-ready*,
2. *develop*, yang digunakan untuk integrasi fitur atau perbaikan *bug* baru,
3. *feature/feature-name*, yang berisi kode fitur baru,
4. *fix/bug-description*, yang berisi kode perbaikan *bug*,  
*hotfix/critical-fix*, yang berisi kode perbaikan *critical bug*.

*Developer* mengerjakan tugas di *branch feature* atau *fix*. Setelah itu, *developer* melakukan *pull request* (PR), menunggu *approval* dari PM, dan *branch* itu dimerge ke *branch develop*.

Pembagian tugas juga dilakukan di GitHub. PM menuliskan *issue* yang berisi daftar tugas untuk setiap *frontend developer*. Selain PM, *frontend developer* juga dapat menulis *issue* yang ditujukan ke tim *backend* untuk keperluan API.

Sementara itu, Huly digunakan untuk menuliskan *issue* yang ditujukan ke seluruh tim, terutama tim UI/UX *Designer*. Penulis dan anggota *developer* lain melakukan rapat dengan PM dan CTO untuk mempelajari cara menggunakan Huly untuk menulis *issue*. Gambar 3.3 menunjukkan tampilan halaman utama di aplikasi Huly yang berisi daftar *issue* yang sudah dibuat.

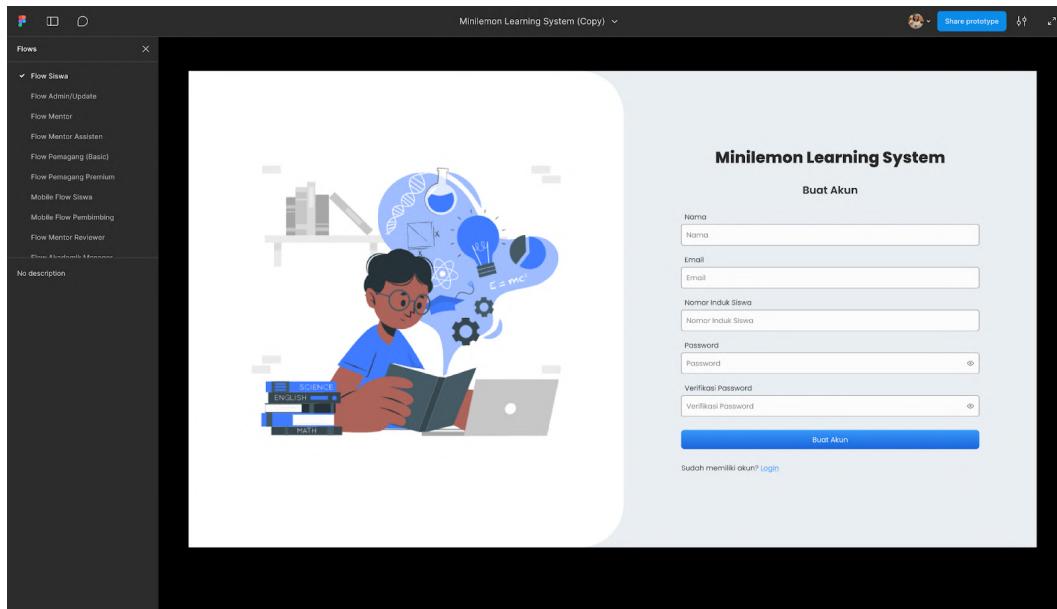
The screenshot shows the Huly application's main interface. On the left is a sidebar with navigation links: Reception, Tracker, My issues, All issues, All projects, Labels, STARRED, MLS (with Issues, Components, Milestones, Templates, Pull requests), YOUR PROJECTS (ML BUZZER, ML UNIVERSE), and Help & Support. The main area is titled 'Issues' and shows a list of 18 items under the heading 'Review in progress'. Each item has a small icon, a title, a status, and a due date. For example, the first item is 'feat: admin manajemen akademik update' with a due date of Dec 24. The interface has a dark theme with light-colored text and icons.

Issue ID	Title	Status	Due Date
MLS_F-1385	feat: admin manajemen akademik update	mis-frontend	0h Dec 24
MLS_F-1371	Feat/tanya jawab siswa, mentor dan admin	mis-frontend	0h Dec 24
MLS_F-1383	feat: modul belajar mentor (CREATE, READ, DELETE), perbaikan UI modul belajar admin	mis-frontend	0h Dec 22
MLS_F-1372	feat: student assignments	mis-frontend	0h Dec 22
MLS_F-1377	Mentor : Classes / Ruang kelas (Revisi)	mis-frontend	0h Dec 20
MLS_F-1380	feat: added video tutorial api	mis-frontend	0h Dec 17
MLS_F-1373	chore: fix typo and assignments tasks format	mis-frontend	0h Dec 16
MLS_F-1379	feat: add mentor active study program management endpoints	mis-frontend	0h Dec 16
MLS_F-1340	fix/feat-Video-Tutorial-Student	mis-frontend	0h Dec 5
MLS_F-1284	Implementasi Flow Mentor Assistant	mis-frontend	0h Nov 20
MLS_F-1314	Implementasi Mentor Dashboard	mis-frontend	0h Nov 20
MLS_F-1316	Task mentor dashboard(alur-kerja) dan admin dashboard(beranda dan sidebar)	mis-frontend	0h Nov 17
MLS_F-1286	feat: update ui student	mis-frontend	0h Nov 17
MLS_F-1285	Mentor-Reviewer	mis-frontend	0h Nov 17
MLS_F-1114	feat: add list progression tugas	mis-frontend	0h Nov 17
MLS_F-1073	fix: resolve API connection	mis-frontend	0h Nov 17
MLS_F-1311	Komponen Chat & Profile Ruang mentor	mis-frontend	0h Nov 17

Gambar 3.3 Tampilan aplikasi Huly

Selain kedua *tools* tersebut, Discord juga dipakai sebagai sarana komunikasi yang lebih cepat. Segala isu yang bisa diselesaikan dengan cepat akan dilakukan di Discord.

Penulis juga diminta untuk pelajari desain MLS di Figma. Tim UI/UX *Designer* membuat prototipe interaktif dari desain Figma untuk setiap tipe pengguna (Admin, Mentor, Siswa). Penulis menggunakan prototipe ini sebagai referensi untuk memahami alur setiap fitur untuk setiap tipe pengguna. Gambar 3.4 menunjukkan satu halaman di prototipe serta daftar alur untuk setiap tipe pengguna.



Gambar 3.4 Tampilan prototipe desain MLS di Figma

Setelah eksplorasi alur dari setiap tipe pengguna, penulis mengetahui bahwa:

1. Akademik memiliki hierarki tertinggi di MLS dan terdiri dari beberapa Program Studi,
2. Setiap Program Studi memiliki berbagai fitur untuk memberi materi kepada siswa seperti Tugas, Video Tutorial, Modul Belajar, dll,
3. Admin memiliki kontrol CRUD penuh atas segala Akademik, Program Studi, fitur-fitur di dalamnya, serta manajemen pengguna,
4. Mentor memiliki kontrol CRUD atas fitur-fitur di dalam Program Studi yang ditugaskan kepada mereka,
5. dan Siswa hanya dapat akses *read* segala fitur-fitur di dalam Program Studi yang mereka ikuti, kecuali dalam pengumpulan Tugas.

Selain mempelajari desain, penulis juga mengeksplorasi kode proyek MLS untuk memahami struktur proyek. Setelah eksplorasi, penulis mempelajari bahwa:

1. MLS menggunakan *page-based routing* yang terbagi menurut tipe user (Admin, Mentor, dan Siswa),
2. Sebagian besar komponen UI menggunakan *library* komponen dari Radix UI yang dimodifikasi dengan *styling* proyek MLS,
3. dan ikon yang digunakan di UI berasal dari Lucide.

- **Slicing UI**

Sebagai *frontend web developer*, tugas utama penulis itu melakukan *slicing*, yaitu mengkonversi desain *UI* dari Figma menjadi kode di aplikasi web. Penulis ditugaskan fitur-fitur seperti Tugas Saya, Video Tutorial, Modul Belajar, *header*, dan Ubah Program Studi.

- **Fitur Tugas Saya**

Untuk fitur Tugas Saya, penulis melakukan *slicing* untuk tipe pengguna Mentor dan Siswa. Fitur ini tempat Mentor mengunggah tugas yang akan dikerjakan oleh Siswa. Masing-masing tipe pengguna memiliki akses CRUD yang berbeda di fitur ini, dan ini tercermin di Tabel 3.2. Mentor memiliki akses lengkap untuk mengelola tugas, sedangkan Siswa hanya dapat melakukan *read*.

Tabel 3.2 Tabel akses CRUD di fitur Tugas Saya

Tipe pengguna	Akses <b>CRUD</b> di fitur Tugas Saya			
	Create	Read	Update	Delete
Mentor				
Siswa				

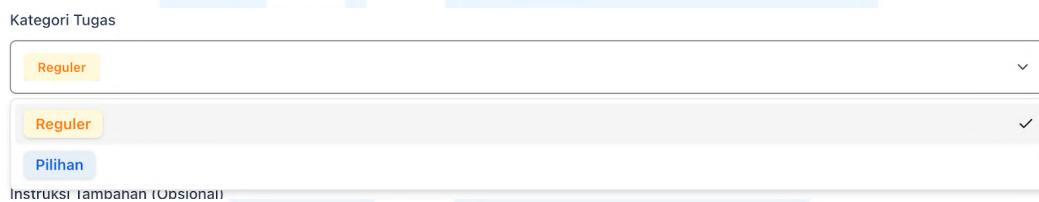
Penulis diberikan tugas untuk memperbarui tampilan dua halaman di fitur Tugas Saya di tipe user Mentor. Halaman pertama yang diubah adalah halaman formulir tambah tugas. Ada tiga *field* yang diubah pada formulir ini, yaitu: Tambah Assignee, Kategori Tugas, dan Instruksi Tambahan.

**Tambah Assignee**

The screenshot shows a user interface for adding assignees. At the top, there is a search bar with a magnifying glass icon and two names: "Amalia" and "Budi". Below the search bar is a button labeled "Semua Siswa". Underneath this, there is a list of student names with checkboxes next to them. The names listed are Amalia, Budi, Chandra, and Dini. The checkboxes for Amalia and Budi are checked, while those for Chandra and Dini are unchecked. A vertical scrollbar is visible on the right side of the list.

Gambar 3.5 *Field* Tambah Assignee

*Field* Tambah Assignee (Gambar 3.5) merupakan fitur *multiselect* dan sekaligus pencarian. Saat fitur ini diklik, *dropdown* berisi daftar nama Siswa muncul. Pengguna dapat mengetik nama Siswa di *search bar* yang tersedia untuk memfilter daftar nama. Nama yang dipilih akan diberi centang biru dan *badge* berwarna biru di dalam *search bar*. Jika ingin menghapus pilihan, pengguna dapat klik *checkbox* yang sama atau klik *badge* biru yang berisi nama yang ingin dihapus. Selain itu, jika semua nama Siswa dipilih, *badge* Semua Siswa akan muncul di *search bar*. Pengguna juga dapat langsung klik *badge* Semua Siswa di dalam *dropdown* untuk melakukan hal yang sama dengan lebih cepat.



Kategori Tugas

- Reguler
- Pilihan

Instruksi Tambahan (Opsional)

Gambar 3.6 *Field* Kategori Tugas

Penulis juga membuat sedikit perubahan ke *field* Kategori Tugas (Gambar 3.6). Sebelumnya, terdapat kategori Khusus di *dropdown*. Desain UI yang baru menghapus kategori ini, sehingga hanya terdapat kategori Reguler dan Pilihan. Penulis menghapus opsi kategori Khusus dari *dropdown*.



Progression  
0%

Instruksi Tambahan (Opsional)

Add a main task	ADD SUB TASK
Add a sub task	REMOVE TASK
Add a main task	ADD SUB TASK
Add a sub task	REMOVE TASK
Add a main task	ADD SUB TASK

Cancel Add

Gambar 3.7 *Field* Instruksi Tambahan

Penulis juga menambahkan *field* Instruksi Tambahan pada formulir Tambah Tugas (Gambar 3.7). *Field* ini terdiri dari *input field* untuk Main Task dan Subtask

dan tombol untuk menambahkan atau mengurangi masing-masing Task. Penulis melewati beberapa revisi sesuai arahan supervisor untuk memastikan bahwa teks di dalam setiap *input field* dapat terlihat dan tidak tertutup oleh tombol.

The screenshot shows a web-based application interface for adding assignments. On the left is a sidebar with navigation links like Beranda, Ruang Kelas, Tugas Saya, Tanya Jawab, Bimbingan Khusus, Video Tutorial, Modul Belajar, Template Gratis, and Ruang Mentor. The main area has a header 'Beranda' and a sub-header 'Tambah Tugas'. It contains fields for 'Nama tugas', 'Materi', 'Deskripsi Tugas', 'Tipe Tugas' (radio buttons for Individu or Kelompok), 'Tambah Assignee' (a search bar), 'Kategori Tugas' (a dropdown menu showing 'Regular'), and a 'Progression' section with a checked checkbox and a progress bar at 0%. Below these are sections for 'Instruksi Tambahan (Opsiional)' with 'Add a main task' and 'Add a sub task' buttons, and a 'Cancel' or 'Add' button at the bottom. A calendar for December 2025 is visible on the right.

Gambar 3.8 Tampilan formulir Tambah Tugas untuk Mentor

Hasil akhir dari tampilan formulir tambah tugas Mentor setelah perubahan dan perbaikan dapat dilihat di Gambar 3.8.

Setelah itu, penulis juga ditugaskan untuk menambahkan dan memperbaiki beberapa bagian di halaman Detail Tugas Saya untuk Mentor, yaitu: bagian *header* dan List Progression Tugas.

The screenshot shows a 'Halaman' (Page) header with a back arrow and the name 'Hanny'. Below it is a card for an assignment titled 'Etika Visual dan Hak Cipta dalam Dunia Kreatif Etika Visual dan...'. The card displays 'On Progress', 'Mata pelajaran lorem ipsum dolor sit amet lorem ips..', 'Start Date: 02-02-2025', 'Due Date: 14-03-2025', and two small icons. At the bottom of the card are five status indicators: 'Days left 12', 'Coming Soon', 'Total Submission 27', and 'Reviewed 12'.

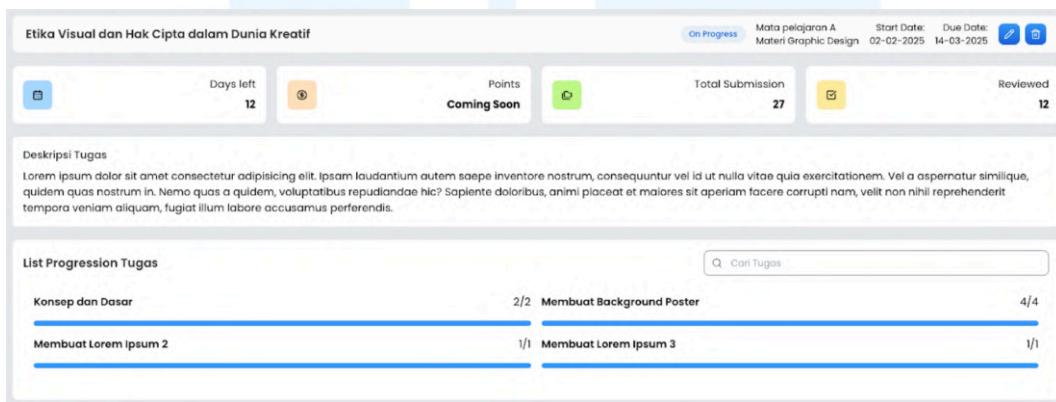
Gambar 3.9 Header Detail Tugas Saya untuk Mentor

Penulis melakukan sedikit perubahan kepada Judul Tugas dan Mata Pelajaran di *header* (Gambar 3.9). Saat teks pada kedua bagian ini terlalu panjang, penulis melakukan *truncate* untuk menambahkan ellipsis di akhir teks untuk memastikan teks tidak akan mengganggu tata letak header.



Gambar 3.10 List Progression Tugas

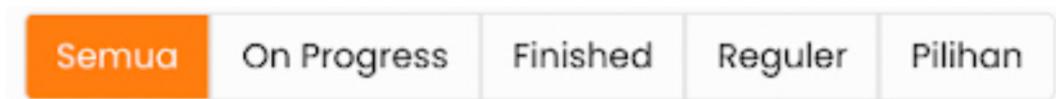
Penulis juga ditugaskan untuk menambahkan List Progression Tugas yang berisi *progress bar* untuk setiap Instruksi Tambahan (Gambar 3.10). Bagian ini mendaftarkan setiap Main Task dan jumlah Subtask ditulis di sebelahnya. Terdapat juga *search bar* untuk memfilter Main Task.



Gambar 3.11 Halaman Detail Tugas Saya untuk Mentor

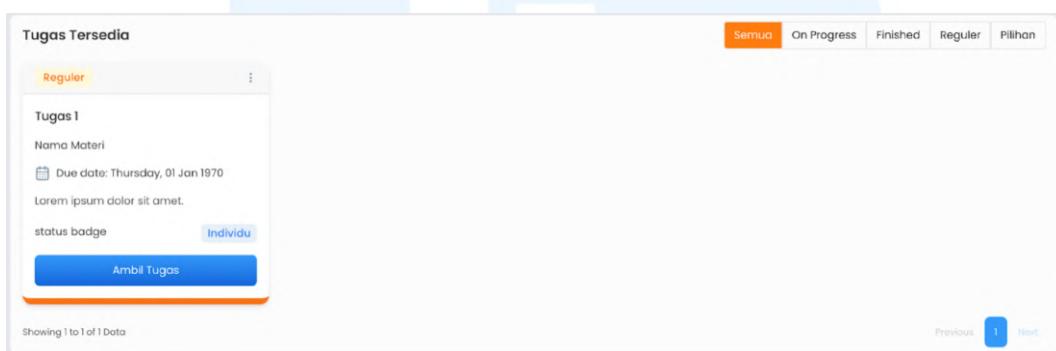
Gambar 3.11 menunjukkan hasil akhir perubahan yang dibuat penulis pada halaman detail Tugas Saya Mentor. Pada saat ini, belum tersedia *endpoint API* untuk menghubungkan *frontend* dengan *backend*. Oleh karena itu, penulis hanya melakukan *slicing* menggunakan data *dummy*.

Sedangkan untuk tipe pengguna Siswa, penulis menambahkan filter di bagian Tugas Tersedia untuk menyaring Tugas dari Status atau Kategori Tugas. Karena komponen UI yang dibutuhkan penulis belum ada di folder proyek MLS, penulis menambahkan komponen *toggle-select* dari *library Radix UI* untuk digunakan sebagai filter.



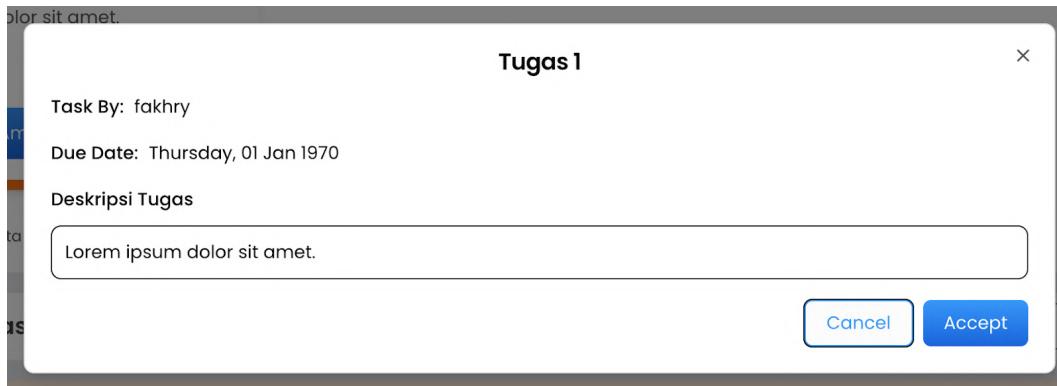
Gambar 3.12 Filter Tugas Tersedia untuk Siswa

Penulis juga membuat perbaikan minor untuk membuat *card* Tugas untuk Siswa lebih akurat dengan desain di Figma dan lebih konsisten dengan *card* Tugas untuk Mentor (Gambar 3.13). Pada tahap ini, skema untuk Tugas belum diberikan Status. Oleh karena itu, penulis belum bisa menambahkan *badge* untuk menunjukkan Status Tugas, dan filter Tugas belum fungsional.



Gambar 3.13 Riwayat Tugas menggunakan data *dummy*

Penulis juga melakukan slicing di halaman Detail Tugas Saya Siswa dengan dengan sebagian yang sudah terintegrasi API (Gambar 3.14). Bagian Detail Tugas sudah terintegrasi API, sedangkan bagian Aktivitas Siswa masih menggunakan data *dummy*. Penulis menemukan bahwa struktur Main Task dan Subtask (bagian *checklist*) di skema Tugas di *backend* belum sesuai dengan struktur yang ada di *frontend*.



Gambar 3.14 Halaman utama Video Tutorial Mentor

The screenshot shows the "Tugas Saya" (My Tasks) page. On the left, there is a sidebar with navigation links: Beranda, Ruang Kelas, Tugas Saya, Tanya Jawab, Bimbingan Khusus, Video Tutorial, Modul Belajar, and Template Gratis. The main area displays the following:

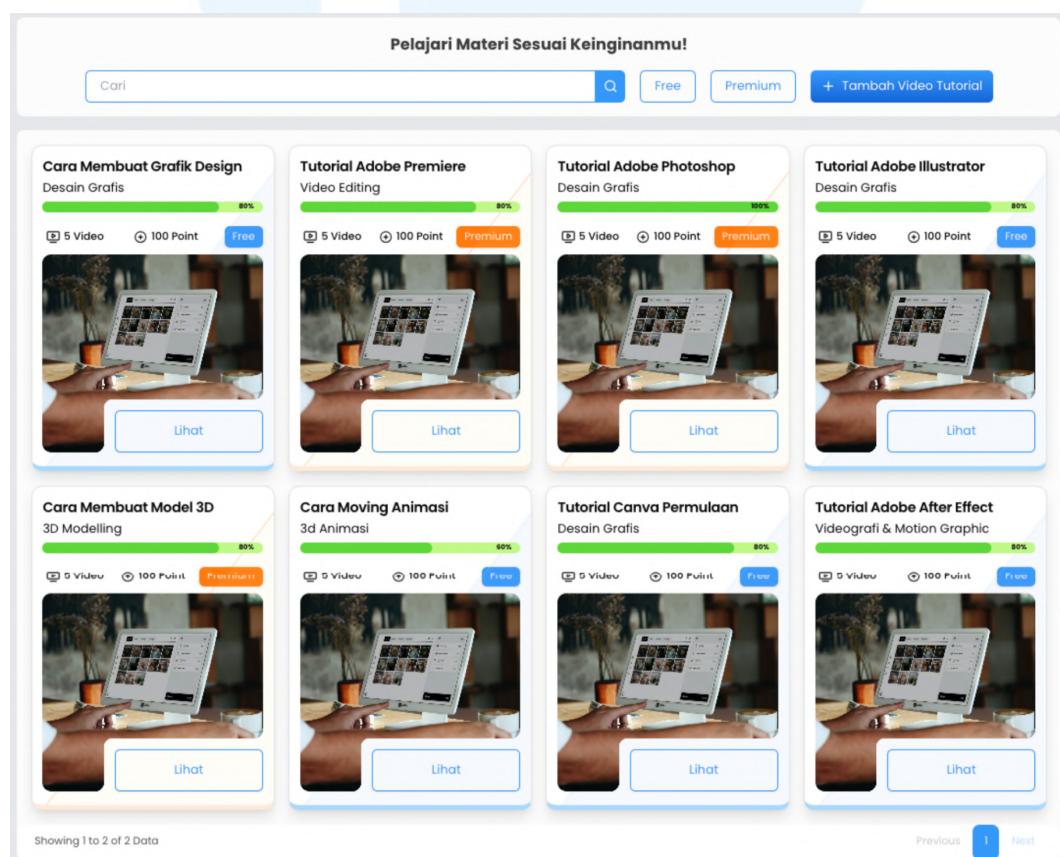
- Tugas 1** [On Progress]
- Detail** (Nama Materi: Thursday, 01 January 1970; Author: fakhry)
- Deskripsi** (Placeholder: "Lorem ipsum dolor sit amet.")
- Aktivitas Siswa** (Progress bars for tasks: "Konsep dan dasar" (2/2), "Membuat Background Poster" (1/4), "Membuat Lorem Ipsum" (0/1), "Membuat Background Poster" (1/4), "Membuat Lorem Ipsum" (0/1)).
- Konsep dan dasar** (Completed):
  - Tentukan Tema Poster
  - Pilih Warna Utama
- Membuat Background Poster** (Progress 1/4):
  - Buat layout Background
  - Tambahkan Tekstur/Shape
  - Gunakan Foto atau Ilustrasi Pendukung
  - Cek Kontras & Keterbacaan
- Membuat Lorem Ipsum** (Progress 0/1)
- Membuat Lorem Ipsum** (Progress 0/1)

Gambar 3.15 Halaman formulir tambah/edit Video Tutorial untuk Mentor

Gambar 3.15 menunjukkan struktur Main Task dan Subtask yang diperlukan di *frontend*. Struktur Subtask terdiri dari *string* teks dan deskripsi, serta *boolean* untuk menentukan Task tersebut sudah dicentang atau belum. Struktur Main Task terdiri dari *string* teks, *boolean* untuk menentukan jika Task tersebut sudah selesai, dan *array* berisi Subtask. Sedangkan skema di *backend* mencantumkan Task hanya sebagai *string*. Karena perbedaan ini, penulis membuat *issue* agar tim *backend* memperbaiki struktur skema supaya integrasi bisa dilakukan di halaman ini.

- **Fitur Video Tutorial**

Video Tutorial merupakan fitur dimana pengguna Mentor dapat mempublikasikan materi dalam bentuk video untuk ditonton oleh Siswa. Penulis melakukan *slicing* untuk memperbarui UI di beberapa halaman fitur Video Tutorial untuk Mentor.



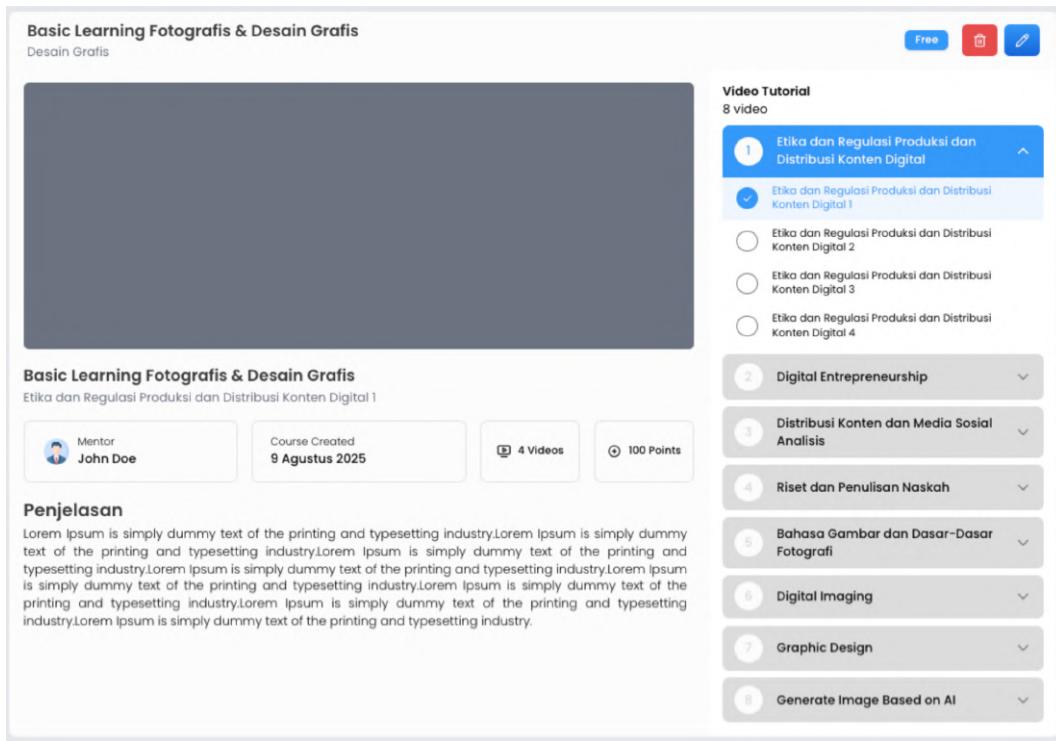
Gambar 3.16 Halaman utama Video Tutorial untuk Mentor

Penulis melakukan perbaikan minor kepada *styling* di halaman utama Video Tutorial Mentor (Gambar 3.16). Penulis mengubah *header* yang tadinya menggunakan *div* biasa dengan *class Tailwind* menjadi komponen *SectionHeader* yang sudah dibuat khusus untuk *header*. Selain itu, penulis juga mengubah tombol menjadi komponen *Button* dari *library* komponen.

The screenshot shows a form for adding or editing a video tutorial. At the top, there are fields for 'Judul Tutorial' (Title) and 'Subjudul Tutorial' (Subtitle). Below these is a 'Upload Cover Image' section with a dashed box for file upload, a 'Support File' button for PNG, JPG, WEBP formats, and a note about landscape orientation (854px X 440px (16:9)). The main content area contains two sections: 'Materi 1' and 'Materi 2'. Each section has a '+' icon to add more, a 'Tambah submateri +' button, and edit/delete icons. A 'Batalkan unggah' (Cancel upload) button is at the bottom left, and a 'Unggah tutorial' (Upload tutorial) button is at the bottom right.

Gambar 3.17 Halaman formulir tambah/edit Video Tutorial untuk Mentor

Pada halaman formulir tambah/edit Video Tutorial Mentor (Gambar 3.17), penulis juga melakukan perbaikan minor kepada *styling*. Penulis mengubah radius border *div* menjadi *rounded-lg* dan mengubah varian di *Button* Tambah Materi dan Submateri menjadi lebih sesuai dengan desain UI/UX di Figma.



Gambar 3.18 Halaman Detail Video Tutorial untuk Mentor

Sedangkan di halaman detail Video Tutorial Mentor (Gambar 3.18), penulis memperbarui tampilan beberapa bagian. Penulis menambahkan Judul dan Subjudul di bawah *div* untuk Video dan Detail Video (Nama Mentor, Tanggal Pembuatan, Jumlah Video, dan Jumlah Poin). Selain itu, penulis juga mengubah bagian *dropdown* yang berisi daftar Materi dan Submateri. Perubahan dilakukan kepada ukuran *font* dan warna. Penulis juga menambahkan nomor indeks Materi di dalam lingkaran setiap *dropdown*.

#### ○ Fitur Modul Belajar

Modul Belajar merupakan fitur yang berisi materi berbentuk *rich text* untuk Siswa. Penulis mengerjakan fitur ini untuk ketiga tipe pengguna MLS. Tabel 3.3 mendeskripsikan perbedaan akses CRUD yang dimiliki setiap tipe pengguna di fitur Modul Belajar. Pengguna Admin dan Mentor memiliki akses CRUD lengkap untuk mengelola Modul Belajar, sedangkan Siswa hanya bisa melihat isi materi. Selain itu, Modul Belajar yang dibuat oleh Mentor memerlukan persetujuan oleh Admin sebelum bisa dipublikasikan dan dilihat oleh Siswa.

Tabel 3.3 Tabel akses CRUD di fitur Modul Belajar

Tipe pengguna	Akses <b>CRUD</b> di fitur Modul Belajar			
	Create	Read	Update	Delete
Admin				
Mentor				
Siswa				

Untuk pengguna Mentor, penulis mengerjakan tiga halaman untuk fitur ini, yaitu: halaman utama, formulir tambah/edit dan halaman detail.

The screenshot shows a web-based application for managing learning modules. At the top, there is a header with the text "Pelajari Bagaimana Modul Belajar Hanya Disini!" and a search bar labeled "Cari". Below the header is a button labeled "+ Tambah Modul Belajar". The main content area displays a grid of eight cards, each representing a different developer role:

- Web Development** by Alya Rahma
- Frontend Dev** by Rafi Nugraha
- Backend Dev** by Zahra Lestari
- Game Dev** by Iqbal Maulana
- 3D Animation** by Putri Ayu
- UI/UX Design** by Dimas Prasetyo
- AI Research** by Salma Nuraini
- Full Stack Dev** by Galih Rizky

Each card features a small thumbnail image of a person working at a desk with a laptop, the developer's name, their role, and a "Lihat" (View) button. At the bottom of the page, there is a footer with the text "Showing 1 to 8 of 12 Data" and a navigation bar with buttons for "Previous", "1", "2", and "Next".

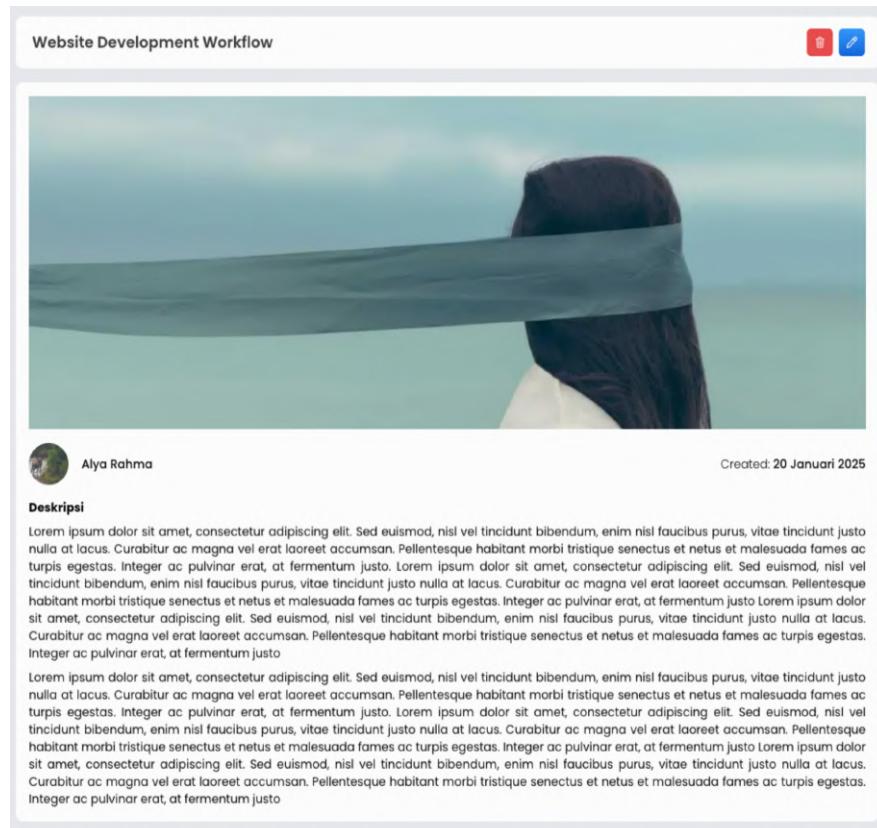
Gambar 3.19 Halaman formulir tambah/edit Modul Belajar untuk Mentor

Pada halaman utama, penulis melakukan sedikit perbaikan styling seperti ukuran font dan ukuran radius border, sehingga tampilannya seperti di Gambar 3.19.

The screenshot shows a Figma prototype of a 'Belajar Modul' (Learning Module) form. At the top left, there are input fields for 'Judul Modul Belajar' (Module Title) and 'Mata Pelajaran' (Subject). Below these is a 'Upload Image Thumbnail' section with a dashed box for file upload, indicating support for PNG, JPG, and WEBP formats in landscape orientation (854px x 440px, 16:9 ratio). To the right, there's a 'Dipublikasikan Pada' (Published On) section with a radio button for 'Otomatis' (Automatic) and another for 'Setel Tanggal' (Set Date). A calendar for December 2025 is displayed, showing the 31st as the selected date. Below the calendar is a note 'Tanggal Dipilih:' (Selected Date:) followed by 'Belum dipilih' (Not yet selected). At the bottom are two buttons: a red 'Batalkan Unggahan' (Cancel Upload) and a blue 'Unggah Modul Belajar' (Upload Learning Module) button.

Gambar 3.20 Halaman Detail Modul Belajar untuk Mentor

Pada halaman formulir tambah/edit Modul Belajar Mentor, penulis melakukan perbaikan kepada *padding* untuk membuat tampilan halaman lebih akurat dengan desain prototipe Figma (Gambar 3.20). Selain itu, penulis juga menambahkan dua *field* baru di formulir, yakni: Text Editor dan Tanggal Publikasi. Text Editor dibuat penulis menggunakan *library* Quill karena *editor*-nya perlu kemampuan untuk edit *rich text*. Sedangkan untuk Tanggal Publikasi, penulis menggunakan *radio select* untuk kedua opsi tanggal publikasi, yaitu: “Otomatis” dan “Setel Tanggal”. Jika pengguna memilih “Otomatis”, maka Modul Belajar akan dipublikasikan di hari yang sama. Sedangkan jika pengguna memilih “Setel Tanggal”, pengguna dapat memilih tanggal di kalender dan Modul Belajar akan dipublikasikan pada tanggal yang dipilih.



Gambar 3.21 Halaman utama Modul Belajar untuk Siswa

Pada halaman Detail Modul Belajar Mentor, penulis melakukan perbaikan minor kepada ukuran *font* dan *padding* supaya lebih akurat dengan tampilan desain di Figma (Gambar 3.21).

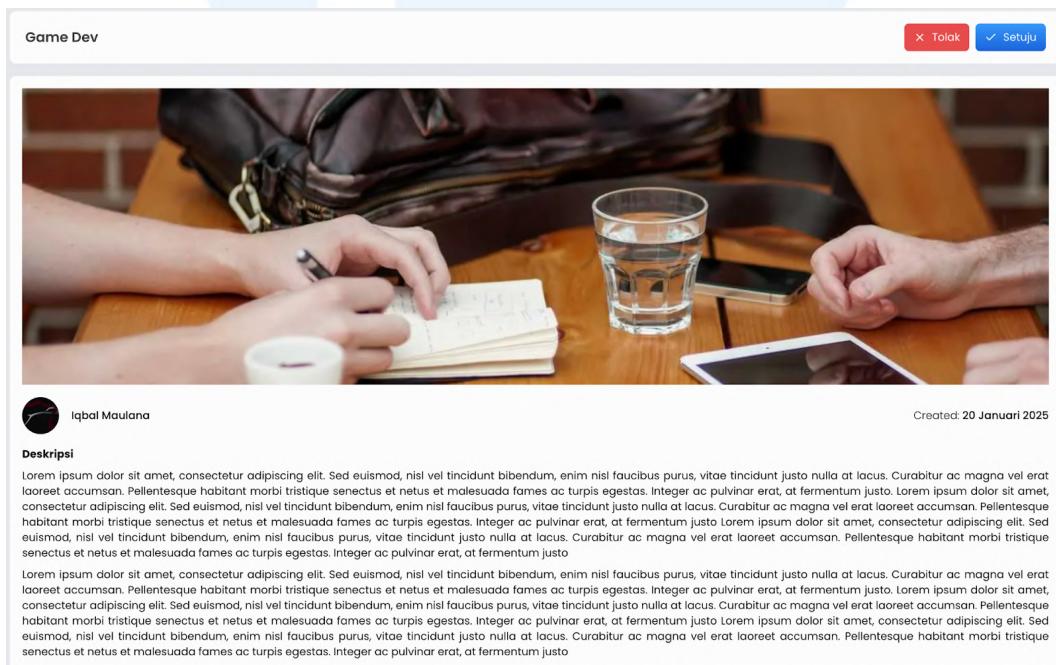
Sedangkan bagi pengguna Admin, akses di Modul Belajar ini sama, dengan tambahan fitur tabel Permintaan Persetujuan. Saat Mentor menunggu Modul Belajar, materinya perlu persetujuan dari Admin sebelum bisa dipublikasikan untuk Siswa. Penulis melakukan *slicing* untuk memperbarui tabel tersebut sehingga tampilannya sesuai dengan desain di Figma, dan ini tercermin di Gambar 3.22.

Permintaan Persetujuan				
ID	Judul	Name Mentor	Tanggal Permintaan	Action
1		Game Dev	Iqbal Maulana	20-01-2025 <button>Detail</button>
2		3D Animator	Putri Ayu	20-01-2025 <button>Detail</button>
3		Mobile App Design	Dimas Prasetyo	20-01-2025 <button>Detail</button>
4		AI Research	Salma Nuraini	20-01-2025 <button>Detail</button>
5		Full Stack Dev	Galih Rizky	20-01-2025 <button>Detail</button>

Showing 1 to 5 of 9 Data Previous 1 2 Next

Gambar 3.22 Tabel Permintaan Persetujuan Modul Belajar untuk Admin

Selain itu, pada halaman Detail Modul Belajar untuk Admin, tombol edit dan hapus di *header* diubah dengan tombol Tolak dan Setuju untuk Admin mengkonfirmasi persetujuan atau penolakannya (Gambar 3.23).



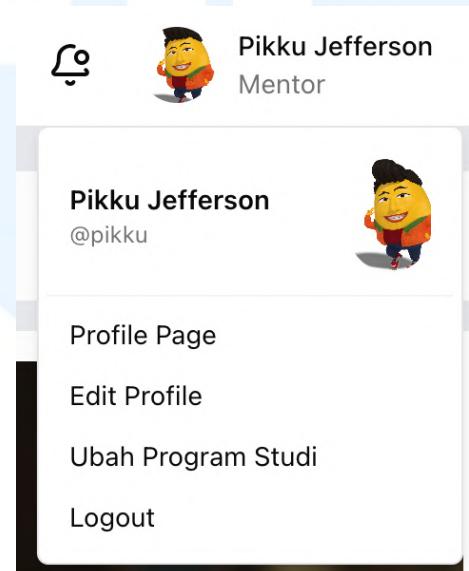
Gambar 3.23 Halaman Detail Modul Belajar untuk Admin

### o *Header*

*Header* adalah komponen yang muncul di setiap halaman beranda untuk ketiga tipe pengguna. Penulis ditugaskan untuk memperbarui tata letak *header* dan menambahkan fitur notifikasi.

Gambar 3.24 *Header* MLS

Penulis mengubah tata letak informasi pengguna supaya sesuai dengan pembaruan desain dari tim UI/UX *Designer* (Gambar 3.24). Letak foto diubah menjadi di sebelah kiri nama pengguna dan tipe pengguna. Selain itu, penulis juga menambahkan ikon lonceng untuk fitur notifikasi di *header*.



Gambar 3.25 *Dropdown* profil pengguna

Informasi pengguna di *header* juga berperan sebagai *dropdown* untuk ke berbagai halaman profil pengguna. Penulis menambahkan opsi untuk ke halaman Ubah Program Studi di *dropdown* (Gambar 3.25).

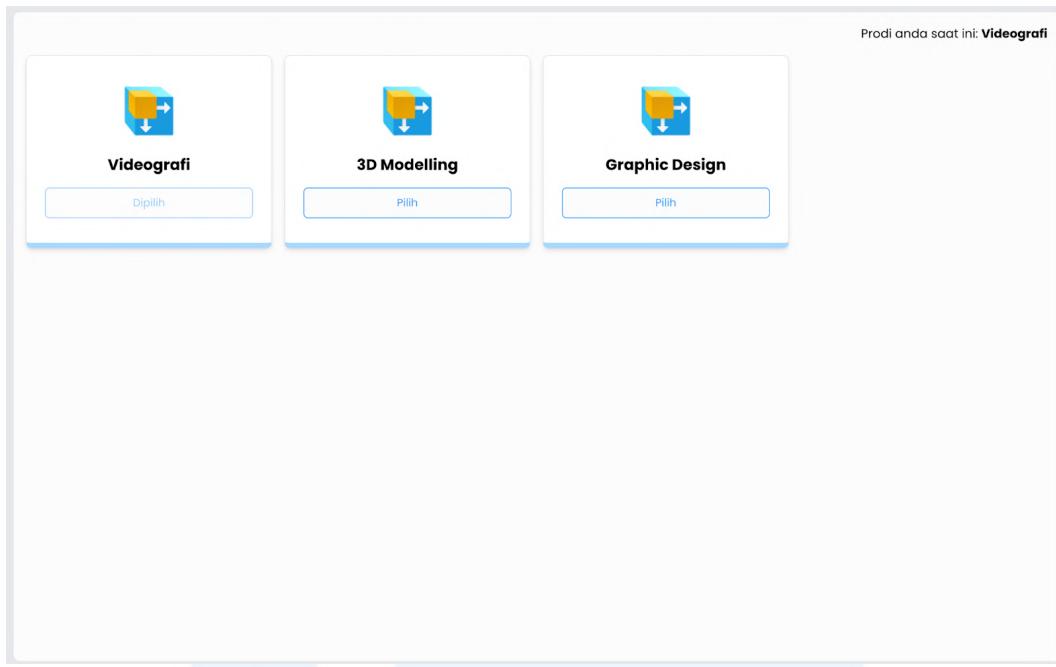


Gambar 3.26 *Dropdown* notifikasi pengguna

Penulis juga menambahkan *dropdown* untuk fitur notifikasi yang berisi data *dummy* (Gambar 3.26).

- **Fitur Ubah Program Studi**

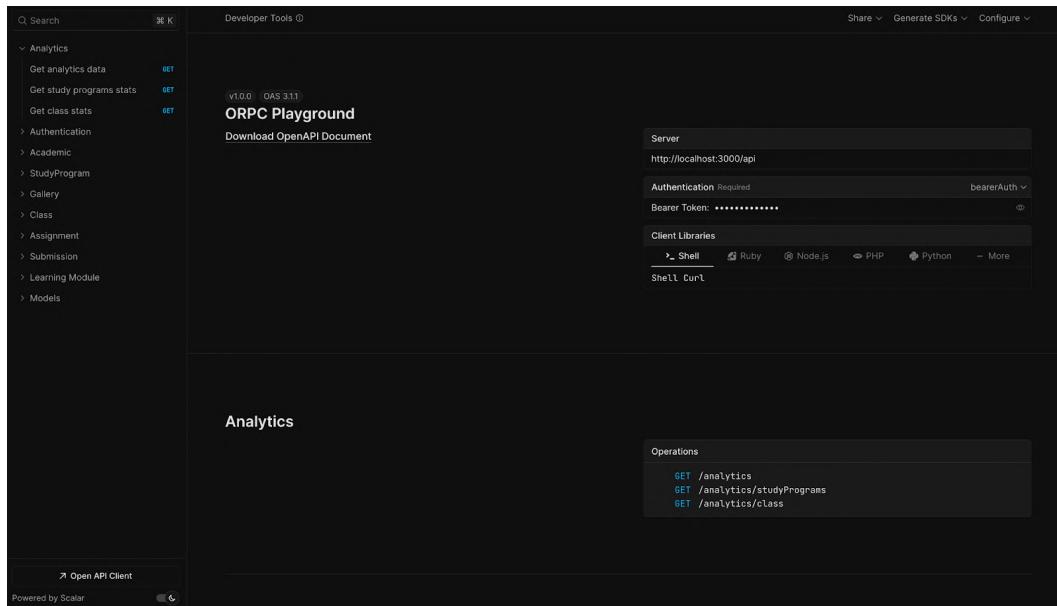
Seorang pengguna Mentor dapat ditugaskan di beberapa Program Studi oleh Admin. Setiap Program Studi berisi materinya tersendiri di setiap fitur. Oleh karena itu, Mentor perlu fitur untuk memilih Program Studi yang ingin diakses. Mentor bisa melakukan fitur tersebut di halaman yang diakses melalui klik opsi Ubah Program Studi pada *dropdown* profil di *header* (Gambar 3.27).



Gambar 3.27 Halaman Ubah Program Studi untuk Mentor

- **Integrasi API**

Selain melakukan *slicing UI*, penulis juga bertugas untuk mengintegrasikan *frontend* dan *backend* via API. Selama proses kerja magang, ada beberapa perubahan dari tim *backend*. Di awal, proyek MLS menggunakan *backend* Next.js, dan beralih ke Convex. Namun, di pertengahan masa kerja magang penulis, tim *backend* kembali menggunakan Next.js sebagai *backend* untuk MLS. Karena ini hal baru untuk penulis, penulis mempelajari cara mengintegrasikan API menggunakan *tech stack* ini. Penulis menggunakan referensi API yang dibuat oleh tim *backend* untuk mempelajari *endpoint-endpoint* yang tersedia. Gambar 3.28 menunjukkan halaman utama referensi API yang mendaftarkan *endpoint* menurut entitas di *database*.



Gambar 3.27 Referensi API

Setelah membaca referensi dengan cermat, penulis seringkali bertemu dengan *endpoint* yang tidak sesuai dengan desain fitur atau *endpoint* yang sama sekali belum dibuat untuk fitur tertentu. Ketika penulis bertemu dengan hambatan-hambatan tersebut, penulis membuat *issue* di GitHub yang berisi detail soal *endpoint* yang perlu diperbaiki atau ditambah.

Tabel 3.4 Daftar *issue* yang dibuat oleh penulis ke tim *backend*



<b>Deskripsi issue</b>	<b>Fitur yang bersangkutan</b>	<b>Tipe pengguna yang bersangkutan</b>
Status tugas ("On Progress", "Revision") belum terdaftar di skema Tugas, sehingga <i>badge</i> Status belum bisa ditampilkan di <i>card</i> Tugas.	Tugas Saya	Mentor, Siswa
Belum ada <i>endpoint</i> untuk fetch daftar Siswa menurut ID Program Studi, sehingga formulir untuk tambah tugas belum bisa menambahkan Assignee dan formulir belum bisa digunakan untuk Tambah Tugas.	Tugas Saya (formulir Tambah Tugas)	Mentor
Skema Tugas di <i>backend</i> masih terdapat <i>field</i> Program Studi, sedangkan sudah dihapus di desain UI/UX, sehingga formulir belum bisa digunakan untuk menambahkan Tugas.	Tugas Saya (formulir Tambah Tugas)	Mentor
Belum ada <i>endpoint API</i> untuk fitur Video Tutorial, sehingga <i>slicing</i> dilakukan menggunakan data <i>dummy</i> .	Video Tutorial (semua halaman)	Mentor
Belum ada <i>endpoint API</i> untuk fitur Modul Belajar, sehingga <i>slicing</i> dilakukan menggunakan data <i>dummy</i>	Modul Belajar (semua halaman)	Mentor

<ul style="list-style-type: none"> <li>Belum ada <i>endpoint</i> untuk menentukan Program Studi yang sedang aktif.</li> <li>Belum didesain sistem untuk menentukan Program Studi yang sedang aktif, sehingga studyProgram diisi dengan <i>null</i> (menampilkan Tugas dari semua Program Studi)</li> </ul>	Ubah Program Studi, Tugas Saya	Mentor, Siswa
Struktur Main Task dan Subtask di skema Tugas di backend belum sesuai dengan struktur yang ada di <i>frontend</i> ( <i>backend</i> hanya menyimpan string, <i>frontend</i> membutuhkan struktur kompleks dengan teks, deskripsi, dan <i>boolean</i> )	Tugas Saya (Detail Tugas - bagian checklist)	Siswa
Setelah <i>endpoint</i> Modul Belajar dibuat, ada kesalahan di struktur endpoint update (formulir tidak memiliki field untuk mengisi nama Mentor, tetapi endpoint update membutuhkan nilai <i>createdBy</i> di <i>request body</i> ).	Modul Belajar (formulir Edit)	Mentor

Lalu, Tabel 3.5 mendaftarkan integrasi API yang berhasil dilakukan oleh penulis.

Tabel 3.5 Daftar integrasi API yang berhasil dilakukan penulis

Fitur	Tipe Pengguna	Jenis API <i>Request</i>	Keterangan
Ubah Program Studi	Mentor	GET (useQuery)	<i>Fetch</i> daftar Program Studi yang ditugaskan kepada pengguna Mentor.
Tugas Saya	Mentor, Siswa	GET (useQuery)	<i>Fetch</i> daftar Tugas untuk Siswa dengan parameter: <i>page</i> , <i>limit</i> , <i>deleted</i> , <i>studyProgram</i> .
Modul Belajar (Halaman Utama)	Mentor	GET (useQuery)	<i>Fetch</i> daftar Modul Belajar milik Mentor.
Modul Belajar (Formulir Tambah)	Mentor	POST (useMutation)	<i>Create</i> Modul Belajar baru.
Modul Belajar (Formulir Edit)	Mentor	PUT/PATCH (useMutation)	<i>Update</i> Modul Belajar (masih ada error di <i>endpoint</i> ).

Penulis menggunakan fungsi `useQuery` dari library TanStack Query untuk fetch data, dan menggunakan fungsi `useMutation` untuk menangani operasi yang mengubah data di server.

- **Menjadi *lead* untuk tim *frontend developer***

Di suatu rapat tim proyek MLS dengan PM dan anggota senior perusahaan, ada pembahasan mengenai kinerja tim. Perkembangan proyek MLS dinilai kurang

efektif dan oleh karena itu perlu ada perombakan tim. PM proyek MLS ditukar dengan PM yang lebih senior di perusahaan. Selain itu, *lead* dari setiap tim juga diganti dengan anggota lain. Penulis mengajukan diri menjadi *lead* untuk tim frontend developer.

Karena ada perubahan ini, alur penggeraan tugas berubah sedikit. Sebagai lead frontend developer, penulis mendapatkan tugas tambahan untuk meninjau hasil pekerjaan setiap *frontend developer* di MLS. Oleh karena itu, setiap PR perlu persetujuan dari *lead* dan PM sebelum bisa digabungkan dengan *branch* utama di *repository*.

### **3.3.2 Kendala yang Ditemukan**

- Kemajuan proyek kurang efektif karena komunikasi kurang baik antar tim.
- Penulis tidak bisa lanjut integrasi API akibat *endpoint* yang belum tersedia.
- Dokumentasi proyek yang minim dari setiap tim, baik *frontend developer*, *backend developer*, dan *UI/UX designer*, sehingga penulis memakan banyak waktu untuk memahami struktur proyek.

### **3.3.3 Solusi atas Kendala yang Ditemukan**

- Penulis menyampaikan kekurangan serta saran untuk solusi kepada seluruh anggota tim saat berada di rapat.
- Penulis membuat *issue* di platform komunikasi tim dengan deskripsi yang rinci serta solusi yang bisa diterapkan ke tim yang bersangkutan.
- Penulis mengeksplorasi desain di Figma dan *repository* di GitHub sendiri, serta berkomunikasi kepada anggota senior untuk mendapatkan penjelasan.