

BAB 3

PELAKSANAAN KERJA MAGANG

3.1 Kedudukan dan Organisasi

Pelaksanaan kerja magang di PT Maleo Edukasi Teknologi sebagai *Front-End Developer*, pada laporan kali ini akan menjelaskan pengembangan apa saja yang dibuat selama periode magang, adapun beberapa proyek lainnya yang akan dijelaskan diakhir laporan.

3.2 Tugas yang Dilakukan

Tugas yang dilakukan selama pelaksanaan kerja magang yaitu membuat LMS berbasis *website application* dengan perincian sebagai berikut :

1. Integrasi CRUD (*Create, Read, Update, Delete*) fitur-fitur yang ada di LMS tersebut.
2. *Handling* komponen-komponen sesuai dengan ketentuan tertentu.
3. Mengatur tampilan antarmuka LMS sesuai dengan permintaan.
4. Melakukan *testing* dan *fixing bug* sebelum masuk ke tahap *deploy* dan sesudah *deploy*.
5. Melakukan presentasi kepada *user* atau klien terkait *flow* dan fitur-fitur yang ada didalam LMS.

3.3 Uraian Pelaksanaan Magang

Minggu pertama pekerjaan yang dilakukan adalah membuat *website company profile* untuk PT Maleo Teknologi, dan memperbaiki *bug* yang dilaporkan di LMS Educourse. Pada minggu kedua dan minggu ketiga mempersiapkan untuk acara Gerakan Sejuta Siswa Digital Indonesia (GESSIT) di Semarang dan pelaksanaannya.

Pada minggu keempat sampai minggu kesembilan pekerjaan yang dilakukan adalah pengembangan fitur di LMS Educourse. Pada minggu kesepuluh melakukan *testing* ulang keseluruhan fitur termasuk yang sudah dikembangkan. Pada minggu

kesebelas sampai minggu keempatbelas mengerjakan proyek baru yaitu integrasi *dashboard* ASN Harati.

Pada minggu kelimabelas sampai ketujubelas menambahkan fitur di LMS Educourse dan dilanjutkan di minggu kedelapanbelas dengan proyek baru yaitu integrasi *website* Muslim Games 2022. Dan diakhiri di minggu kesembilanbelas dan duapuluh dengan acara Kiddo Fest dan GESSIT Tangerang. Seluruh pelaksanaan kerja magang dan proses yang dikerjakan dalam LMS ini dijabarkan dalam Tabel 3.1

Tabel 3.1. Deskripsi pekerjaan setiap minggu

Minggu Ke -	Pekerjaan yang dilakukan
1	<i>Fix</i> bug yang ada di LMS. Presentasi ke salah satu sekolah untuk sosialisasi LMS. Integrasi <i>company profile website</i> untuk PT Maleo Teknologi.
2	Ubah CSS <i>sidebar</i> LMS. Integrasi fitur baru yaitu <i>gallery</i> di LMS. Persiapan <i>Event</i> GESSIT Semarang.
3	<i>Event</i> GESSIT Semarang. Integrasi fitur baru <i>business partner</i> di LMS.
4	Integrasi fitur baru <i>event</i> di LMS. Memperbaiki beberapa UI di LMS. Integrasi fitur <i>subclass</i> .
5	Integrasi fitur baru testimoni dan <i>meeting</i> . <i>Testing</i> fitur dan memperbaiki beberapa <i>bug</i> .
6	Integrasi fitur baru <i>tracking</i> materi dan kumpulan materi. <i>Testing</i> fitur di <i>live</i> . Integrasi tabel <i>feedback</i> di dalam tab <i>course</i> .
7	Integrasi fitur baru rapor di <i>dashboard</i> admin. Integrasi fitur baru lihat <i>course</i> dan lihat kehadiran siswa. Integrasi fitur baru bikin kategori kelas di <i>dashboard</i> sekolah.
8	Integrasi <i>dashboard</i> baru yaitu uploader. Integrasi fitur baru lihat rapor di <i>dashboard</i> siswa. <i>Testing dashboard</i> admin dan sekolah.
Lanjut pada halaman berikutnya	

Tabel 3.1 – Deskripsi pekerjaan setiap minggu (lanjutan)

Minggu Ke -	Pekerjaan yang dilakukan
9	<i>Testing dashboard</i> siswa dan uploader. Presentasi ke sekolah untuk sosialisasi LMS. <i>Handling null</i> data di semua tabel LMS.
10	Mengerjakan catatan bug yang sudah dicatat saat <i>testing</i> di LMS.
11 - 14	Integrasi proyek baru yaitu ASN Harati Kalimantan Tengah.
15	Memperbaiki bug dari laporan <i>client</i> di LMS.
16	Menambahkan fitur baru di <i>dashboard</i> admin yaitu integrasi sertifikat. Menambahkan <i>breadcrumb</i> di LMS.
17	Integrasi fitur baru <i>assignment</i> di LMS. Integrasi fitur tagihan di <i>dashboard</i> sekolah.
18	<i>Testing</i> fitur baru. Persiapan <i>event</i> kiddo. Merubah <i>library</i> tabel dan beberapa UI.
19	Menjadi MC di <i>event</i> kiddo. Integrasi di proyek baru yaitu Muslim Games 2022.
20	Persiapan dan pelaksanaan <i>event</i> GESSIT Tangerang. Integrasi proyek baru Digital City.

Pekerjaan dapat dilihat pada Gambar 3.1 selama pelaksanaan kerja magang membuat LMS, selain itu ada juga beberapa proyek lainnya diantaranya terdapat ASN Harati, Muslim Games 2022, Digital City, dan beberapa *event* lainnya.

LMS Educourse sudah berhasil dibuat sejak bulan Juli 2022 namun ada beberapa fitur yang perlu ditambahkan untuk mendukung keperluan sekolah-sekolah dan guru-guru di Educourse. Dalam proyek ini pengerjaan yang dilakukan adalah integrasi *front-end* dengan *back-end* pada fitur-fitur yang akan dibuat.

Proyek ASN Harati adalah sebuah proyek membuat LMS untuk pemerintah Kalimantan Tengah yang bertujuan untuk memajukan sumber daya manusia di daerahnya dengan memberikan pelatihan-pelatihan dalam bentuk modul pembelajaran yang dimasukkan kedalam LMS. Dalam Proyek ini pengerjaan yang dilakukan sama dengan LMS Educourse yaitu melakukan integrasi *front-end* dengan *back-end* di dalam *dashboard* LMS. Proyek ini telah berhasil dibuat dan sudah berjalan.

Proyek Muslim Games 2022 adalah sebuah proyek dengan tujuan untuk mempererat persaudaraan antar Masjid di daerah Tangerang Selatan dengan mengadakan beberapa perlombaan. Dalam proyek ini pengerjaan yang dilakukan adalah membuat tampilan halaman untuk registrasi lomba dan beberapa informasi terkait dengan masjid yang menyelenggarakan perlombaan tersebut. Pekerjaan lainnya adalah melakukan integrasi *front-end* dengan *back-end* agar website bisa berjalan. Proyek ini sudah selesai dibuat tetapi masih ada beberapa *request* dari *client* untuk beberapa fitur dan tampilan yang diubah dan ditambahkan.

Proyek Digital City adalah sebuah proyek yang bertujuan untuk mendigitalisasi beberapa hal di lingkungan Bumi Serpong Damai khususnya di perumahan-perumahan. Dalam proyek ini pekerjaan yang dilakukan spesifik pada bagian *dashboard smart payment*. Pekerjaannya yaitu integrasi *front-end* dengan *back-end* di beberapa fitur. Proyek ini masih dalam tahap pengerjaan.

Event yang diadakan oleh Educourse ada dua yaitu Gerakan Sejuta Siswa Digital Tangerang, Semarang dan Kiddo Fest. pada *event* gessit bertugas sebagai perlengkapan dan pada *event* Kiddo Fest sebagai *Master of Ceremony Workshop*.

Framework yang digunakan dibagian *front-end* dalam proses pembuatan LMS ini adalah *Next.js* sedangkan untuk *framework* yang digunakan di bagian *Back-End* adalah *Django*. Sedangkan untuk proyek lainnya *framework* yang digunakan adalah *React.js*.

3.3.1 Integrasi Fitur LMS

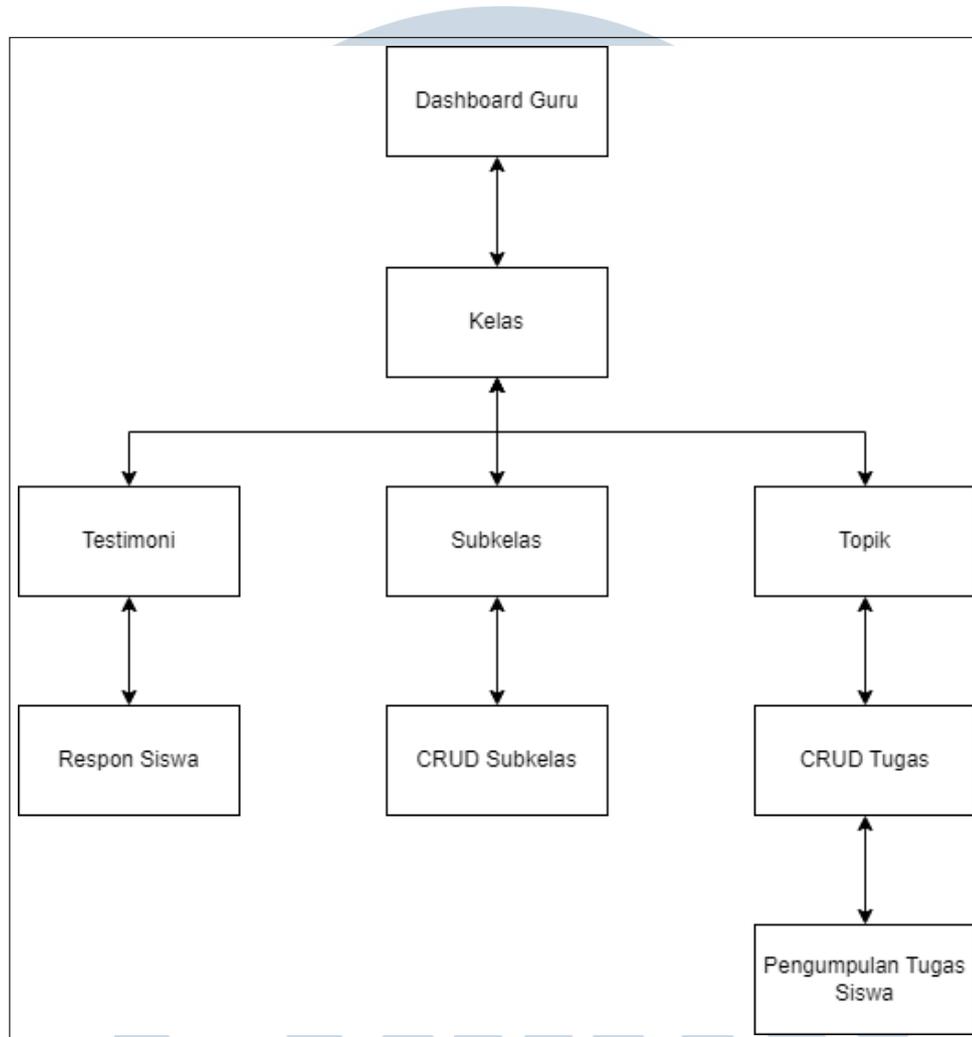
Integrasi pengembangan LMS ini dibagi menjadi beberapa bagian diantaranya *dashboard admin*, guru, siswa, sekolah dan *dashboard* untuk *role* baru yaitu *uploader*.

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



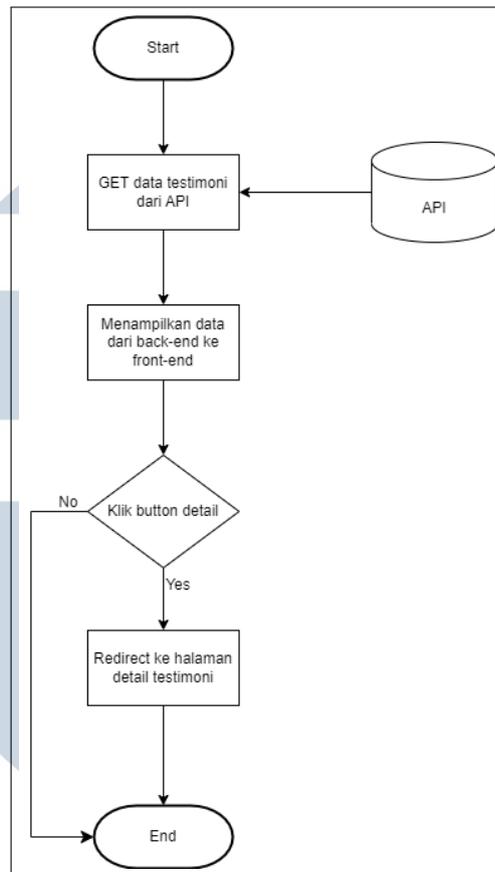
N U S A N T A R A
 Gambar 3.1. Usecase fitur-fitur yang dikembangkan

A. Dashboard Guru



Gambar 3.2. Sitemap pengembangan dashboard guru

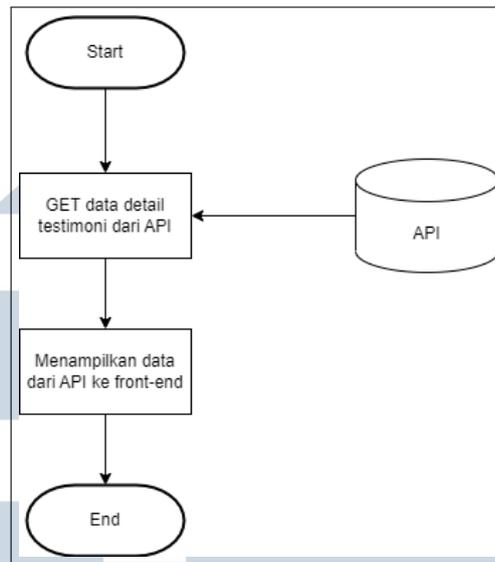
Gambar 3.2 adalah *sitemap* dari fitur yang diintegrasikan diantaranya ada tiga fitur yaitu testimoni, tagihan, dan rapor. Fitur testimoni dapat membantu guru dalam meningkatkan performa mengajarnya. Dalam fitur ini guru bisa melihat jumlah nilai yang diterima dari siswa-siswi yang diajarnya. proses penampilan data dilakukan dengan menggunakan *axios* get ke API yang sudah disediakan kemudian diintegrasikan ke dalam tampilan yang sudah disiapkan sebelumnya. *Flowchart* dari penampilan *list* testimoni yang didapat oleh guru dapat dilihat pada Gambar 3.3



Gambar 3.3. *Flowchart list testimoni*

Guru juga bisa melihat detail dari testimoninya yang berisi setiap siswa-siswi memberikan nilai berapa terhadap guru tersebut agar bisa melihat jika ada kesalahan atau cara mengajar yang kurang terhadap siswa-siswi tersebut. Proses pembuatan fitur ini sama seperti pada halaman testimoni diatas hanya saja tidak terdapat bagian aksi. *Flow* fitur detail testimoni bisa dilihat pada *flowchart* di Gambar 3.4.

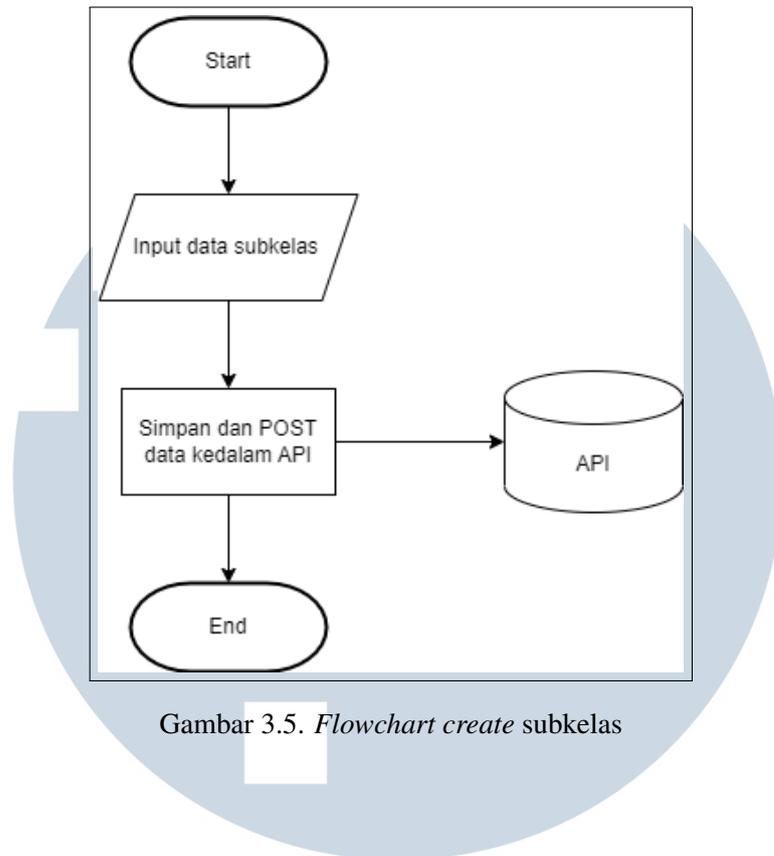
UNIVERSITAS
MULTIMEDIA
NUSANTARA



Gambar 3.4. *Flowchart list detail testimoni*

Setelah fitur testimoni, fitur lainnya yang ditambahkan adalah fitur subkelas dimana fitur ini berfungsi untuk membagi kelas yang dibuat menjadi beberapa kelas kecil, sebagai contohnya jika ada kelas yang dibuat untuk kelas 6 SD didalamnya akan ada beberapa kelas kecil seperti kelas 6A, 6B, dan seterusnya. Fitur ini dibuat untuk mempermudah guru melihat siswa-siswi secara spesifik sesuai kelasnya di sekolah tersebut. Guru bisa membuat subkelas secara mandiri dengan mengisi data yang diperlukan untuk membuat subkelas dalam form yang sudah disediakan, kemudian data akan di *POST* ke API yang sudah disediakan dari *back-end* menggunakan library *axios*. *Flow* proses *create* bisa dilihat pada *flowchart* di Gambar 3.5

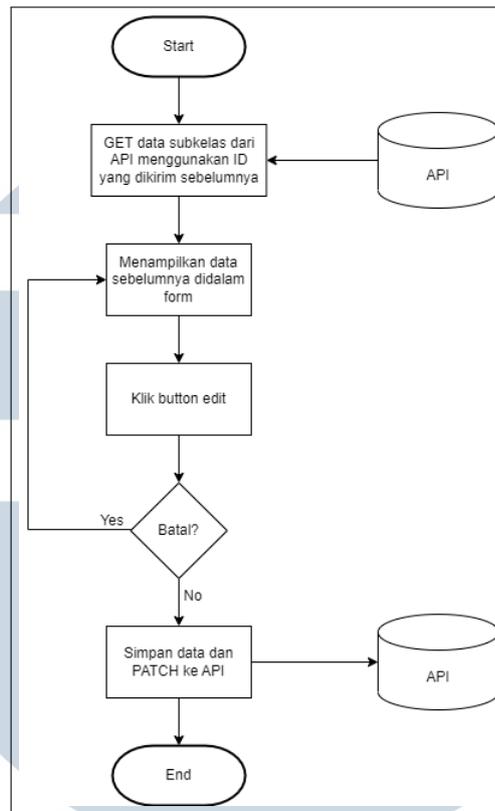
U N I V E R S I T A S
M U L T I M E D I A
N U S A N T A R A



Gambar 3.5. *Flowchart create subkelas*

Guru bisa melakukan *edit* pada data dalam subkelas yang sudah dibuat sebelumnya. Proses *edit* dilakukan dengan mengirim id subkelas pada *button* edit di tabel kemudian id dikirim ke halaman *edit* yang akan digunakan untuk melakukan *update* pada subkelas yang akan diubah setelah guru mengisi data yang akan dirubah menggunakan *axios patch*. *Flow edit* bisa dilihat pada *flowchart* di Gambar 3.6

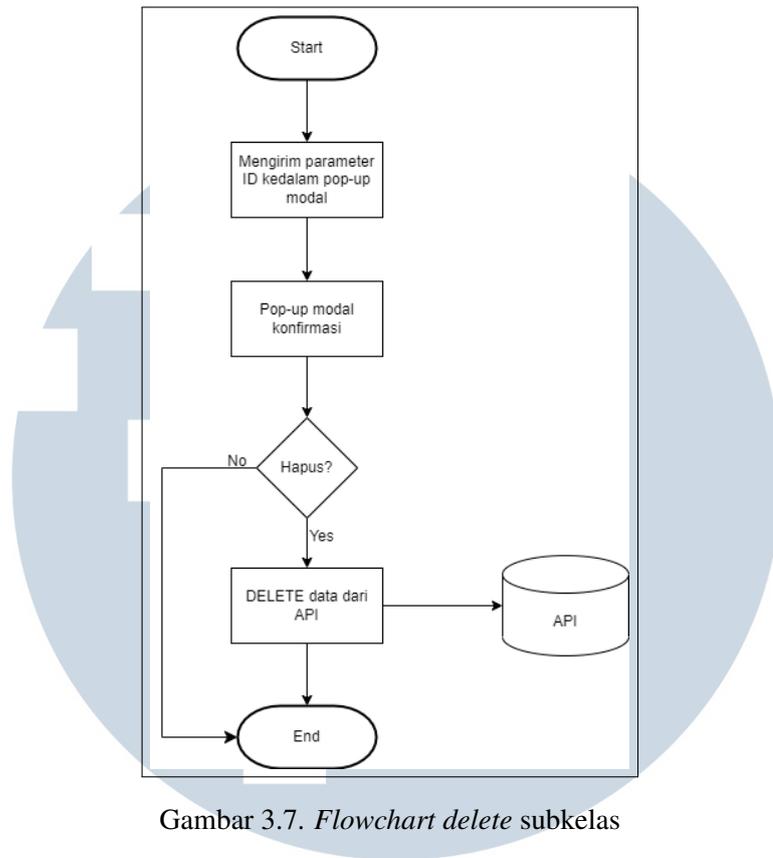
U M M N
 U N I V E R S I T A S
 M U L T I M E D I A
 N U S A N T A R A



Gambar 3.6. Flowchart edit subkelas

Guru juga bisa melakukan *delete* pada subkelas yang sudah dibuat jika diperlukan. Proses *delete* subkelas dilakukan dengan cara yang hampir sama dengan *edit* sama-sama mengirim parameter ID kemudian ID tersebut akan digunakan untuk menghapus subkelas yang diinginkan menggunakan *axios delete*. Flow *delete* bisa dilihat pada *flowchart* di Gambar 3.7

UNIVERSITAS
MULTIMEDIA
NUSANTARA

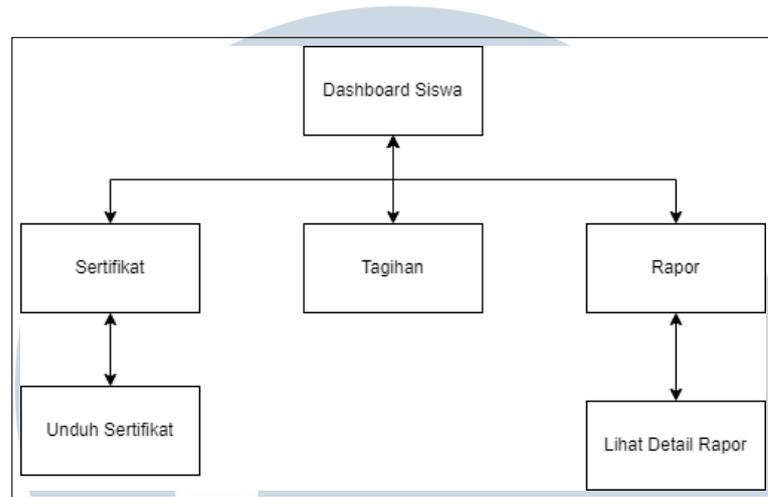


Gambar 3.7. Flowchart delete subkelas

Pengembangan LMS pada *dashboard* guru juga ada penambahan didalam fitur topik dimana didalamnya terdapat fitur tugas untuk guru dapat memberikan tugas-tugas dalam bentuk soal pdf dan siswa-siswi bisa mengirimkan jawabannya dalam bentuk file juga. Proses CRUD tugas sama seperti CRUD fitur lainnya yang membedakan hanya pada data yang dikirim dan *url link* API yang digunakan saat proses CRUD.

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

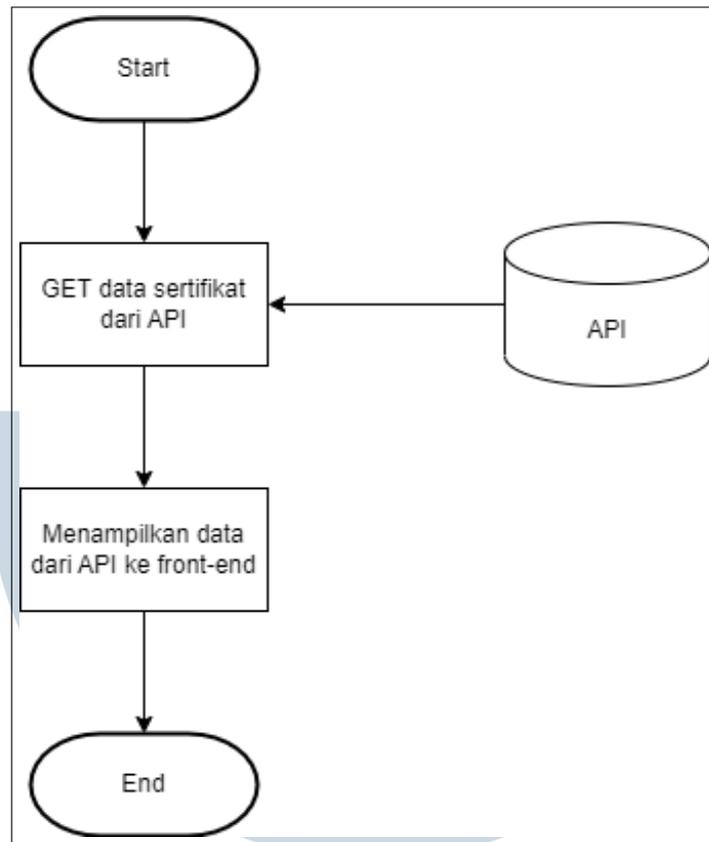
B. *Dashboard Siswa*



Gambar 3.8. *Sitemap pengembangan dashboard siswa*

Gambar 3.8 adalah *sitemap* dari fitur yang dikembangkan di *dashboard* siswa diantaranya terdapat lihat sertifikat, tagihan dari sekolah, dan lihat rapor. untuk fitur sertifikat dibuat agar siswa-siswi bisa melihat sertifikat yang didapat setelah selesai menyelesaikan kelas yang diikuti jika ada. untuk proses penampilan data masih sama dengan sebelumnya menggunakan *axios get* dari API yang sudah disiapkan. *Flow* penampilan data sertifikat dapat dilihat di Gambar 3.9

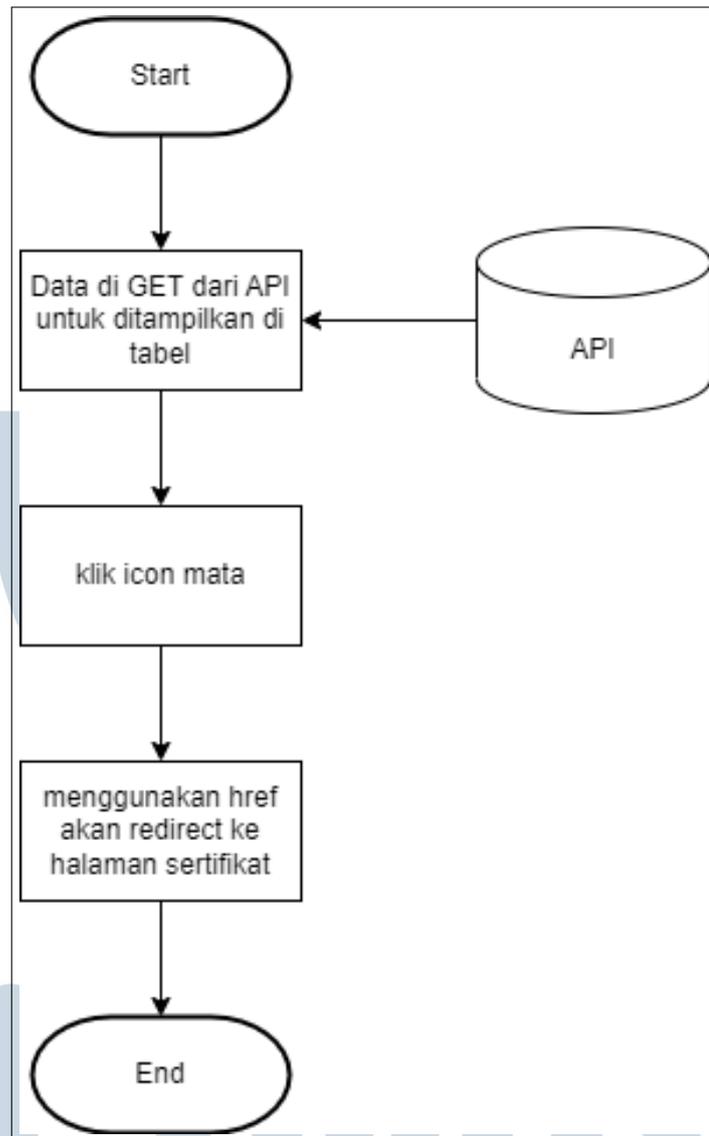
UMMN
UNIVERSITAS
MULTIMEDIA
NUSANTARA



Gambar 3.9. Flowchart penampilan data sertifikat

Siswa-siswi juga bisa melihat bagaimana bentuk sertifikat yang didapat dengan menekan *button* dengan logo mata yang nantinya akan di redirect ke halaman baru yang menampilkan sertifikatnya, namun sertifikatnya belum bisa diunduh. *Flow* menampilkan sertifikat ke halaman baru adalah dengan *redirect* ke *link* sertifikat yang sudah di generate dari API. untuk lebih jelasnya bisa dilihat di Gambar 3.10

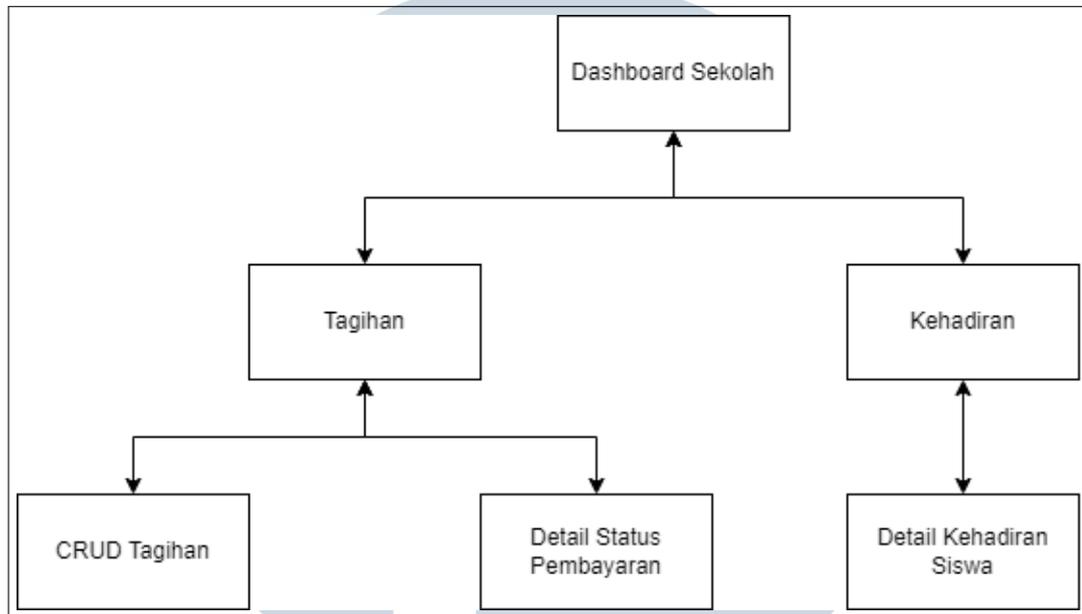
UNIVERSITAS
MULTIMEDIA
NUSANTARA



Gambar 3.10. Flowchart sertifikat

UNIVERSITAS
MULTIMEDIA
NUSANTARA

C. Dashboard Sekolah

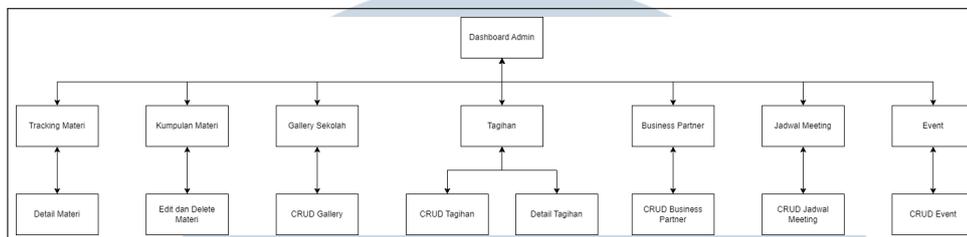


Gambar 3.11. Sitemap pengembangan dashboard sekolah

Gambar 3.11 adalah *sitemap* dari pengembangan yang dilakukan pada halaman *dashboard* sekolah diantaranya fitur tagihan dan fitur kehadiran. Fitur tagihan dibuat untuk mempermudah sekolah dalam membuat segala bentuk tagihan yang ditujukan kepada siswa-siswi sekolahnya bisa berupa tagihan uang sekolah dan lain-lain. Proses CRUD yang ada di fitur tagihan sama seperti CRUD yang ada di fitur sebelumnya yang membedakan hanya saja data yang diinput kedalam form dan API penyimpanan data tagihan.

Fitur kehadiran dibuat untuk mempermudah sekolah dalam melihat dan memantau kehadiran siswa-siswi di sekolahnya pada fitur ini hanya ada tabel untuk menampilkan nama kegiatan dan detail kehadiran untuk melihat siapa saja siswa yang menghadiri dan yang tidak dalam kegiatan tersebut. Untuk proses penampilan data sama seperti fitur testimoni.

D. Dashboard Admin



Gambar 3.12. Sitemap pengembangan dashboard admin

Gambar 3.12 adalah *sitemap* dari pengembangan fitur di *dashboard admin* diantaranya terdapat fitur *tracking* materi, kumpulan materi, *gallery* sekolah, tagihan, *business partner*, jadwal *meeting*, dan *event*. Untuk fitur *tracking* materi dan kumpulan materi dibuat agar *admin* bisa melihat dan memantau materi apa saja yang dibuat dan apakah sudah sesuai dengan kebutuhan siswa sesuai dengan tingkatnya masing-masing. pada fitur ini hanya menampilkan data saja dan proses pengambilan datanya sama seperti fitur-fitur sebelumnya yaitu mengambil data dari API dan diintegrasikan kedalam tabel.

Fitur *gallery* sekolah dibuat untuk melakukan dokumentasi kegiatan yang dilakukan di beberapa sekolah. Proses CRUD yang dibuat sama seperti fitur lainnya yang membedakan hanya data yang dikirim ke API dan *link* API untuk melakukan CRUD di fitur ini.

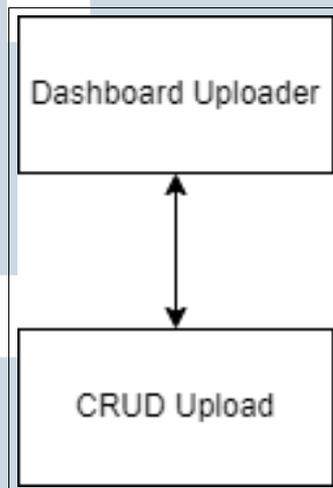
Fitur Tagihan dibuat untuk memantau dan membuat tagihan, sama seperti fitur yang ada di sekolah hanya saja fitur tagihan di *dashboard admin* dibuat khusus untuk *user-user* dari Educourse itu sendiri. Untuk proses pembuatan CRUD fitur ini sama seperti yang ada di *dashboard* sekolah yang membedakan hanya API nya saja.

Fitur *business partner* adalah fitur yang dibuat untuk para reseller ataupun pihak lain yang ingin melakukan kerjasama dengan Educourse namun fitur ini belum aktif digunakan. proses CRUD yang dilakukan sama dengan CRUD fitur lainnya yang membedakan hanya data dan API nya saja.

Fitur jadwal *metting* adalah sebuah fitur yang dibuat untuk menjadwalkan sebuah pertemuan yang bisa digunakan antara guru dan siswa ataupun untuk rapat internal guru-guru. fitur ini juga belum aktif digunakan. proses CRUD yang dilakukan sama seperti fitur lainnya.

Fitur *event* adalah sebuah fitur yang bertujuan untuk mempermudah Educourse jika ingin membuat sebuah kegiatan seperti seminar atau perlombaan di fitur ini juga mempermudah proses registrasi *user* yang ingin ikut berpartisipasi.

3.3.2 Dashboard Uploader



Gambar 3.13. Sitemap dashboard uploader

Pada Gambar 3.13 adalah *sitemap* dari *dashboard* uploader didalamnya hanya terdapat satu fitur yaitu untuk melakukan *upload* bukti pengajaran dari guru-guru di Educourse. Fitur ini berfungsi juga untuk memantau apakah guru-guru di Educourse benar-benar melakukan pengajaran. Proses CRUD di fitur ini sama seperti fitur-fitur yang melakukan proses CRUD.

3.3.3 Pekerjaan Umum yang Dilakukan didalam LMS

Pada proyek pengembangan LMS ini pekerjaan yang dilakukan diantaranya integrasi dari *front-end* dengan *back-end*, mengubah dan menambahkan beberapa komponen didalam LMS. potongan *coding* akan dilampirkan dibawah agar lebih jelas apa yang dikerjakan.

A. Mengambil Data dari API dan Ditampilkan di *Front-end*

Tabel yang digunakan pada LMS ini pertama-tama masih menggunakan *template* awal dan setelah berdiskusi ternyata belum efisien. Tabel dirubah menjadi

jenis tabel baru menggunakan *library* mdbreact. Untuk potongan *coding* bisa dilihat di Gambar 3.14, Gambar 3.15 dan Gambar 3.16.

```
7 import { MDBDataTable } from "mdbreact";
```

Gambar 3.14. *Import library* mdbreact

Pada Gambar 3.14 proses pertama yang dilakukan adalah melakukan *import library* yang sudah di *install* sebelumnya. proses ini berfungsi agar ada parameter yang dipanggil dari *library* untuk digunakan membuat tabelnya.

```
const data = {
  columns: [
    {
      label: t("dashboard_teacher.course_listing.no"),
      field: "index",
      sort: "asc",
      width: 20,
    },
    {
      label: t("curriculum.created_at"),
      field: "date",
      sort: "asc",
      width: 20,
    },
    {
      label: t("curriculum.name"),
      field: "title",
      sort: "asc",
      width: 20,
    },
    {
      label: t("curriculum.class_name"),
      field: "material",
      sort: "asc",
      width: 20,
    },
  ],
  rows: attendance.map((e, index) => ({
    index: `${index + 1}`,
    date: e.created_at,
    title: e.student_name !== null ? e.student_name : "-",
    material: e.class_name !== null ? e.class_name : "-",
  })),
};
```

Gambar 3.15. Data Tabel

Pada Gambar 3.15 proses selanjutnya yang dilakukan yaitu deklarasi kolom dan baris pada tabel untuk mengisi judul setiap baris dan *looping* dari database yang sudah diambil juga dilakukan disini pada bagian *rows* yang berisi data dari tabel tersebut.

```
124      [loading && <Spinner color='success' />]
125      <MDBDataTable
126          noBottomColumns={true}
127          theadTextWhite={false}
128          responsive
129          bordered
130          hover
131          data={data}
132      />
```

Gambar 3.16. Implementasi Tabel

Pada Gambar 3.16 langkah terakhir yang dilakukan yaitu memanggil komponen tabel yang sudah diambil dari *library*. Kemudian dimasukan data tabel yang sudah dideklarasikan sebelumnya. Ditambahkan juga komponen *loading* agar *user* tidak bingung saat data sedang di proses dari server jadi tidak menampilkan isi tabel yang kosong tetapi ada logo *loading* yang tertampil.

B. Menambahkan Breadcrumb di Setiap Halaman

Pengembangan LMS Educourse selain merubah tabel ada juga penambahan *Breadcrumb* pada halaman-halaman di LMS dengan tujuan untuk mempermudah *user* dalam berpindah ke halaman sebelumnya. Potongan *coding* dapat dilihat di Gambar 3.17

```
<nav aria-label="breadcrumb">
  <ol className="breadcrumb">
    <li className="breadcrumb-item"><a href="/">Home</a></li>
    <li className="breadcrumb-item"><a href="/dashboard/school/attendance/class"{t("curriculum.attendance")}></a></li>
    <li className="breadcrumb-item active" aria-current="page">{t("curriculum.attendance_detail")}</li>
  </ol>
</nav>
```

Gambar 3.17. Breadcrumb

Pada Gambar 3.17 proses yang dilakukan adalah menambahkan *url path* yang jika ditekan akan *redirect* ke halaman yang dituju sesuai dengan nama dari halaman yang tertera di *breadcrumb*.

C. Menambahkan Fitur Unduh Data Tabel ke Excel

Pada beberapa fitur di LMS terdapat beberapa kondisi yang memerlukan data dalam tabel bisa untuk diunduh dalam bentuk excel agar bisa di *print* fisik. Untuk potongan *coding* bisa dilihat pada Gambar 3.18, Gambar 3.19 dan Gambar 3.20

```
import { ExportToExcel } from "../../../../../components/ExportExcel/ExportToExcel";
```

Gambar 3.18. Potongan coding *import library* *exporttoexcel*

Pada Gambar 3.18 adalah proses pertama yang dilakukan yaitu *import library* yang sudah di *install* terlebih dahulu. Proses ini bertujuan untuk deklarasi komponen yang nanti akan dipanggil.

```
const fileName = "Tagihan Siswa"; // here enter filename for your excel file
const scores = [];
for (let i = 0; i < billing.length; i++) {
  scores.push({
    No: i + 1,
    Nama: billing[i].name_user,
    Amount: billing[i].amount_billing,
    Status: billing[i].status_billing,
  });
}
```

Gambar 3.19. Potongan coding proses deklarasi data

Pada Gambar 3.19 adalah proses selanjutnya yang dilakukan yaitu deklarasi data yang akan di *download* dalam bentuk excel. Data yang di *download* di ambil dari *database* yang sudah dipanggil dari *back-end* dan sudah dimasukkan kedalam *array* kemudian di *looping* datanya.

```
<div className="col-lg-12 col-md-12">
  <ul className="more_options_tt">
    <li>
      <ExportToExcel apiData={scores} fileName={fileName} />
    </li>
  </ul>
</div>
```

Gambar 3.20. Potongan coding implementasi *exporttoexcel*

Pada Gambar 3.20 adalah proses terakhir yang dilakukan yaitu memanggil komponen yang sudah *import* sebelumnya kemudian dimasukkan parameter yang sudah dideklarasikan sebelumnya yang berisi data yang akan di *download*.

D. Menambahkan Handling Data Null pada Setiap Tabel

Tabel yang sudah ada sekarang terkadang membuat website tidak bisa berjalan dengan baik, dikarenakan ada beberapa data baru yang di-input yang tidak lengkap isinya, hal ini mengakibatkan data yang diambil dari API bersifat *null* atau tidak ada isinya. *Handling* diperlukan di setiap tabel agar tidak terjadi hal tersebut. Potongan *coding* dapat dilihat pada Gambar 3.21

```
date: e.name_user !== null ? e.name_user : "-",
email: e.email_user !== null ? e.email_user : "-",
title: e.amount_billing !== null ? e.amount_billing : "-",
```

Gambar 3.21. *Handling null data* di Tabel

Potongan *coding* diatas adalah bentuk *if else* di Next.js dimana jika data tidak sama dengan *null* data diambil, jika data *null* maka akan muncul "-".

E. Testing dan Fixing Bug

LMS Educourse selama kerja magang sering mendapat laporan jika ada cukup banyak bug yang terjadi di LMS tersebut, akhirnya tim *developer* memutuskan untuk melakukan *testing* semua fitur di LMS dan dicatat kemudian diselesaikan satu per satu *bug* yang ditemukan.

Page	Bagian	Bagian yang diuji	Hasil	Keterangan	Status Selesai	Diperbaiki oleh	Keterangan setelah diperbaiki
Beranda	BreadCrumb	Klik BreadCrumb	Berfungsi dengan baik	Berfungsi dengan baik yaitu menuju halaman yang di klik	Selesai		
	Card Information	Klik Informasi Card	Berfungsi dengan baik	Berfungsi dengan baik yaitu menuju halaman yang di klik	Selesai		
	Search	Search	Berfungsi dengan baik	Berfungsi dengan baik	Selesai		
Nilai	BreadCrumb	Klik BreadCrumb	Berfungsi dengan baik	Berfungsi dengan baik yaitu menuju halaman yang di klik	Selesai		
	Search	Search	Berfungsi dengan baik	Berfungsi dengan baik	Selesai		
Sertifikat	BreadCrumb	Klik BreadCrumb	Berfungsi dengan baik	Berfungsi dengan baik yaitu menuju halaman yang di klik	Selesai		
	Aksi	Button "Lihat Sertifikat"	Berfungsi dengan baik	Berfungsi dengan baik yaitu menuju halaman sertifikat	Selesai		
Pembayaran	Search	Search	Berfungsi dengan baik	Berfungsi dengan baik	Selesai		
	BreadCrumb	Klik BreadCrumb	Berfungsi dengan baik	Berfungsi dengan baik yaitu menuju halaman yang di klik	Selesai		
	Aksi	Button "Bayar Sekarang"	Berfungsi dengan baik	Berfungsi dengan baik yaitu menuju halaman vendit	Selesai		
Kelas	BreadCrumb	Klik BreadCrumb	Berfungsi dengan baik	Berfungsi dengan baik yaitu menuju halaman yang di klik	Selesai		
	Kelas Baru	Button Kelas Baru	Berfungsi dengan baik	Berfungsi dengan baik yaitu menuju halaman buat kelas baru	Selesai		
	Tab Kelas	Search	Berfungsi dengan baik	Berfungsi dengan baik	Selesai		

Gambar 3.22. *Testing dan fixing bug*

3.3.4 Pekerjaan diluar Proyek LMS

Selain proyek LMS ada juga beberapa proyek lainnya yang dikerjakan, diantaranya :

1. *Dashboard* ASN Harati, di proyek ini membuat *dashboard* LMS seperti Educourse hanya saja untuk wilayah Kalimantan Tengah.
2. *Website Company* Maleo Teknologi, di proyek ini membuat *website company profile* dengan menggunakan Wordpress.
3. Integrasi *front-end* di *website* Muslim Games 2022, dalam proyek ini *website* berfungsi sebagai *platform* pendaftaran lomba-lomba antar masjid.
4. Panitia acara Gerakan Sejuta Siswa Digital di Semarang dan Tangerang. Kegiatan ini bertujuan untuk melakukan sosialisasi tentang dunia *digital* dalam dunia pendidikan kepada kepala sekolah di daerah tersebut.
5. Menjadi *master of ceremony* di *event* Kiddo Fest. *Event* ini dilaksanakan di Mall Pluit Village yang bertujuan memperkenalkan Educourse ke pengunjung mall.
6. *Smart Payment Digital City*, dalam proyek ini pekerjaan yang dilakukan adalah integrasi *front-end* dengan *back-end* pada *dashboard website* tersebut.

3.4 Kendala dan Solusi yang Ditemukan

3.4.1 Kendala

Kendala yang dihadapi selama pelaksanaan kerja magang diantaranya :

1. Sulit mencari *bug* yang dilaporkan dari beberapa *user* dikarenakan beberapa *bug* yang dilaporkan saat diuji tidak mengalami kendala seperti yang dilaporkan
2. Sulit membagi waktu antara proyek satu dengan yang lainnya karena dikerjakan secara paralel.
3. Sulit mengaplikasikan fitur baru yang diminta karena kurangnya dokumentasi yang disediakan oleh *library* tersebut.

3.4.2 Solusi

Solusi dari kendala yang ditemukan tentunya sudah ditemukan dan diselesaikan diantaranya :

1. Meminta bantuan dengan rekan kerja lain untuk rapat mencari bersama-sama solusi dari *bug* yang sulit ditemukan solusinya.
2. Meminta bantuan dari rekan kerja yang ada dan mulai melihat prioritas pekerjaan mana yang lebih penting, lebih susah, dan lebih dekat waktu *deadline*-nya.
3. Meminta bantuan dari supervisor atau teman kerja lainnya untuk memahami dokumentasi yang ada.

