

BAB 3

PELAKSANAAN KERJA MAGANG

3.1 Kedudukan dan Organisasi

Pelaksanaan kerja magang di PT Maleo Edukasi Teknologi sebagai *Full-stack Developer*, namun pada proyek yang dijelaskan di laporan ini pekerjaan yang dilakukan lebih mengarah pada *Front-end Web Developer* dikarenakan lebih menguasai bidang tersebut.

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*.
5. Melakukan presentasi kepada *user* atau klien terkait *flow* dan fitur-fitur yang ada didalam LMS.

3.3 Uraian Pelaksanaan Magang

Dua minggu pertama, kedua dan ketiga dari pelaksanaan kerja magang digunakan untuk mempelajari bahasa yang digunakan oleh perusahaan yaitu *ReactJs*, diberikan materi pembelajaran melalui *Udemy* yang telah disediakan oleh perusahaan agar saat terjun kedalam proyek tidak membutuhkan waktu lama untuk dapat memahami dan mengerjakannya, selain hanya belajar di *Udemy* diberikan juga tugas-tugas kecil untuk menguji pemahaman selama belajar yang akan dipresentasikan setiap tugas tersebut telah selesai dibuat.

Selanjutnya minggu keempat mempelajari *flow* coding yang sudah ada sehingga tidak kesulitan dalam melanjutkan proyek yang sudah berjalan, mulai dari

halaman *login* sampai masuk kedalam UI (*User Interface*) dari *dashboard* yang sudah dibuat oleh tim UI/UX.

Pada minggu kelima sampai minggu keduabelas mulai mengerjakan proyek LMS ini, tugas yang diberikan adalah melakukan integrasi CRUD pada setiap *dashboard* sesuai dengan *role* yang sudah ada, selain melakukan integrasi CRUD mengerjakan *handling* beberapa komponen untuk di tampilkan dan disembunyikan sesuai dengan ketentuan yang sudah ditetapkan, serta melakukan beberapa revisi tampilan antarmuka yang dirasa masih kurang tepat oleh supervisor.

Pada minggu ketigabelas dan keempatbelas melakukan *testing* dan memperbaiki *error* yang ditemukan serta mempresentasikan *flow* kerja LMS ini kepada klien. 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 | Mempelajari bahasa pemrograman yang digunakan dari materi <i>Udemy</i> yang diberikan. |
| 2 | Melanjutkan materi pembelajaran dari <i>Udemy</i> serta mengerjakan tugas-tugas dasar dari supervisor. Mempresentasikan tugas-tugas dasar yang sudah dikerjakan. |
| 3 | Menyelesaikan pembelajaran yang tersisah dari <i>Udemy</i> dan mengerjakan tugas-tugas dasar lainnya yang diberikan supervisor. Mempresentasikan tugas-tugas dasar terakhir dari supervisor. |
| 4 | Mempelajari <i>flow</i> codingan yang sudah dibuat sebelumnya. |
| 5 | Membuat <i>website</i> dimulai dari bagian <i>dashboard</i> guru dengan melakukan integrasi CRUD pada pembuatan kelar oleh akun guru. |
| 6 | Melanjutkan integrasi CRUD pada <i>dashboard</i> guru bagian isi dari kelas yang sudah dibuat seperti topik dan materi. |
| 7 | Melanjutkan integrasi CRUD pada <i>dashboard</i> guru bagian kuis beserta halaman pembuatan dan penilaian soal kuis. |
| 8 | Mengerjakan integrasi <i>read</i> dan <i>edit</i> pada <i>dashboard</i> sekolah bagian verifikasi siswa dan guru. |
| Lanjut ke halaman selanjutnya | |

Tabel 3.1 – Lanjutan dari halaman sebelumnya

| Minggu Ke - | Pekerjaan yang dilakukan |
|--------------------|--|
| 9 | Mengganti beberapa data yang masih di hardcode dengan data dari API. Memperbaiki beberapa kesalahan minor seperti kurang <i>spinner loading</i> |
| 10 | Mengatur materi dan video agar tidak dapat di unduh oleh pengguna. Membuat <i>pagination</i> di setiap tabel yang ada. |
| 11 | Menyelesaikan revisi yang diterima dari supervisor di berbagai halaman dan menambahkan beberapa tombol-tombol agar mempermudah pengguna dalam mengakses <i>website</i> . |
| 12 | Menambahkan dua bahasa di <i>website</i> dalam bahasa inggris dan juga bahasa indonesia. |
| 13 | Melakukan <i>testing</i> keseluruhan <i>website</i> dan memperbaiki <i>bug</i> yang ditemukan. |
| 14 | Melakukan presentasi kepada klien yang akan menggunakan LMS. |

Berdasarkan uraian Tabel 3.1 pekerjaan yang dilakukan secara garis besar selama pelaksanaan kerja magang membuat LMS, selain itu ada juga beberapa aktivitas yang dilakukan seperti *weekly meeting*, *team tech discussion*, *meeting investor* dan menjadi panitia di kegiatan offline kantor.

Weekly meeting adalah sebuah kegiatan mingguan di kantor yang bertujuan untuk memberikan perkembangan masing-masing divisi kepada CEO (*Chief Executive Officer*) dan CTO (*Chief Technology Officer*) agar bisa mendapatkan *feedback* untuk perbaikan pekerjaan di minggu tersebut, selain itu ada juga sesi *book review* dimana sebelum rapat mingguan dimulai salah satu orang akan menceritakan buku apa yang sudah dibaca kepada rekan-rekan kerja lainnya, hal ini bertujuan untuk melestarikan budaya membaca dan menambah ilmu baru setiap minggunya. di rapat ini kami juga melakukan *update OKR* (*Objective and Key Result*) agar setiap minggunya memiliki target yang harus dicapai.

Team tech discussion adalah sebuah kegiatan yang dilakukan setiap senin setelah *weekly meeting* dimana semua tim *tech* berdiskusi untuk membahas apa saja yang perlu diselesaikan minggu tersebut dan pembagian tugas, selain itu jika memiliki masalah dibagian *coding* akan didiskusikan solusi yang bisa dicoba atau

diterapkan.

investor meeting dalam waktu tertentu tim *tech* dibutuhkan di *investor meeting* untuk mempresentasikan produk yang sudah dibuat seperti LMS dan beberapa kali mendapatkan kesempatan untuk mempresentasikan di depan *investor* terkait LMS yang sudah dibuat, ataupun proyek lainnya.

event luring juga sudah mulai dilakukan karena pandemi yang sudah mulai membaik, dikarenakan jumlah rekan kantor yang tidak terlalu banyak membuat semua orang harus ikut membantu merencanakan dan melaksanakan *event* tersebut, karena di perusahaan belum ada divisi kreatif membuat pekerjaan terkadang merangkap sebagai ketua *event* atau penyusun acara, salah satunya adalah *event* sosialisasi program GESIT (Gerakan Sejuta Siswa Digital) yang dihadiri seluruh kepala sekolah SD (Sekolah Dasar) di Tangerang Selatan.

Framework yang digunakan dalam proses pembuatan STEAM LMS ini adalah *Next.js* sedangkan untuk *framework* yang digunakan di bagian *Back-End* adalah *Django*.

3.3.1 Analisis Kebutuhan

STEAM LMS ini dibuat dengan tujuan untuk memperluas produk ke arah B2B (*Business to Business*), untuk mengetahui apa saja fitur-fitur yang diperlukan oleh LMS. *Senior software engineer* bersama CTO berdiskusi, setelah itu *senior software engineer* menyampaikan hasil diskusi tersebut dengan hasil sebagai berikut:

1. Dibutuhkan empat *role* yaitu murid, guru, sekolah, dan *admin*.
2. *Page* murid terdapat beberapa fitur diantaranya, bisa melakukan proses belajar dalam bentuk video dan modul, bisa melakukan ujian dalam bentuk pilihan ganda, dan melihat nilai serta sertifikat yang didapat. Murid juga bisa melakukan pembayaran di LMS tersebut.
3. *Page* guru terdapat beberapa fitur diantaranya, bisa melakukan CRUD untuk *course* yang dibuat beserta isi dari *course* tersebut, dan guru juga bisa mengikuti *course* yang disediakan oleh LMS Educourse. guru juga bisa melakukan pembayaran *course* yang ingin diikuti di LMS Educourse
4. *Page* sekolah terdapat fitur untuk melakukan verifikasi siswa dan guru untuk sistem keamanan agar tidak ada pihak asing yang tidak memiliki afiliasi

dengan sekolah tersebut.

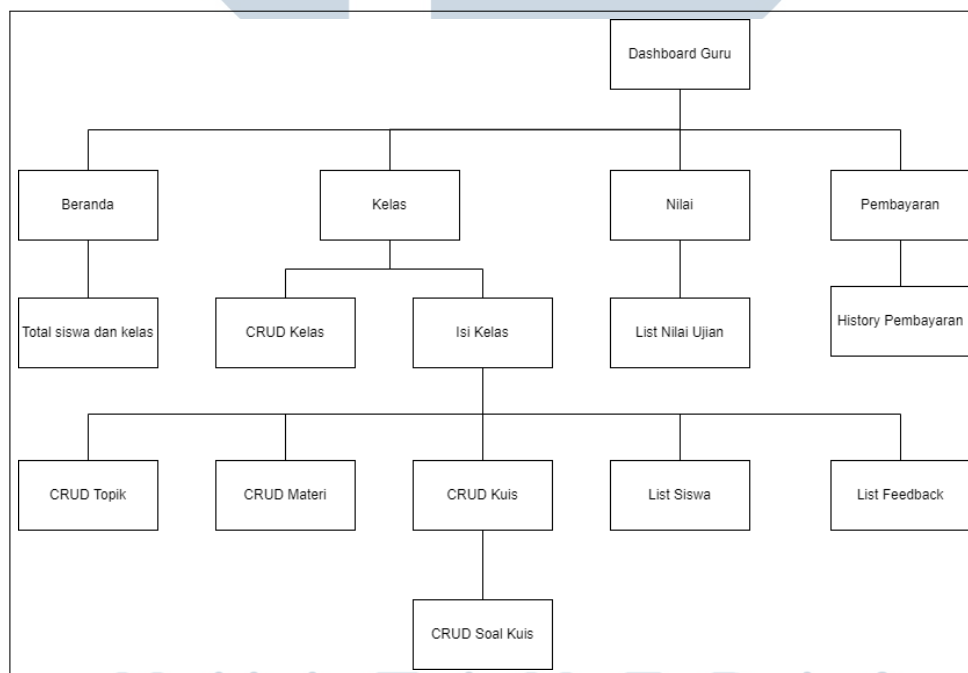
5. *Page admin* terdapat fitur untuk melakukan verifikasi sekolah yang sudah melakukan kerjasama atau membeli layanan LMS Educourse.

Ketentuan diatas adalah ketentuan pertama kali saat ingin membuat LMS ini, untuk fitur-fitur tambahan menyesuaikan dengan permintaan CTO atau ide-ide baru yang muncul selama proses pembuatan LMS tersebut.

3.3.2 Pembuatan LMS

Pembuatan LMS ini dibagi menjadi beberapa bagian, diantaranya integrasi CRUD pada bagian *dashboard* murid, guru, sekolah dan admin.

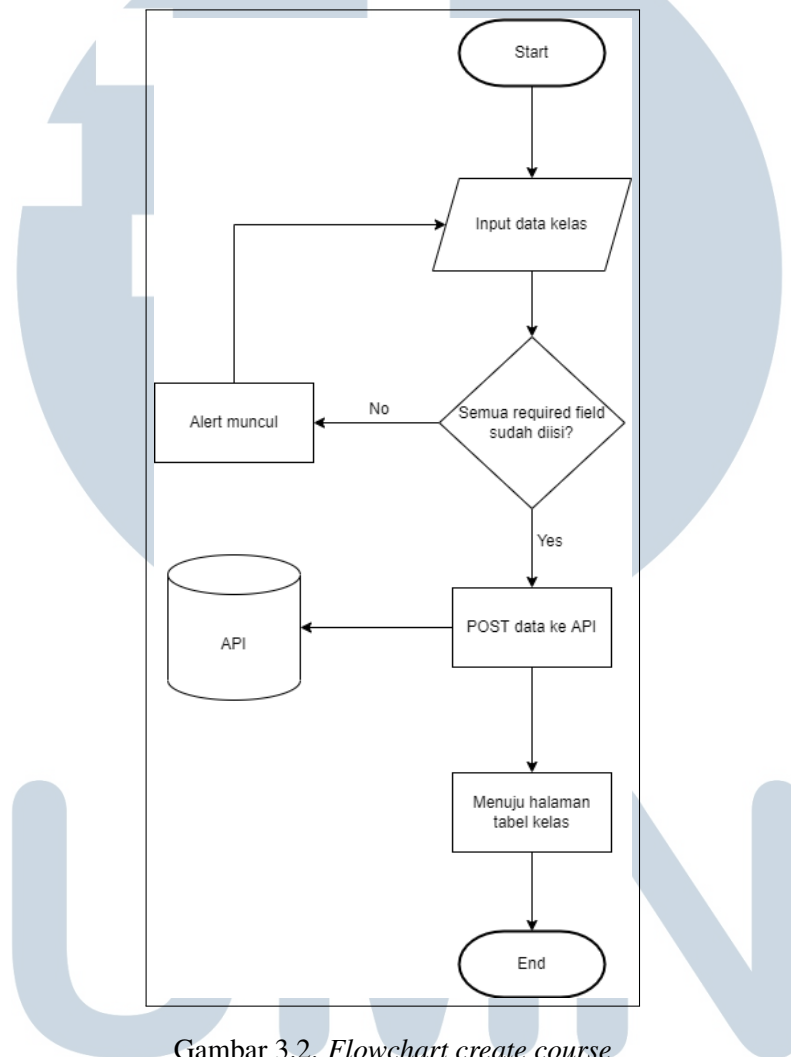
A. *Dashboard Guru*



Gambar 3.1. Sitemap dashboard guru

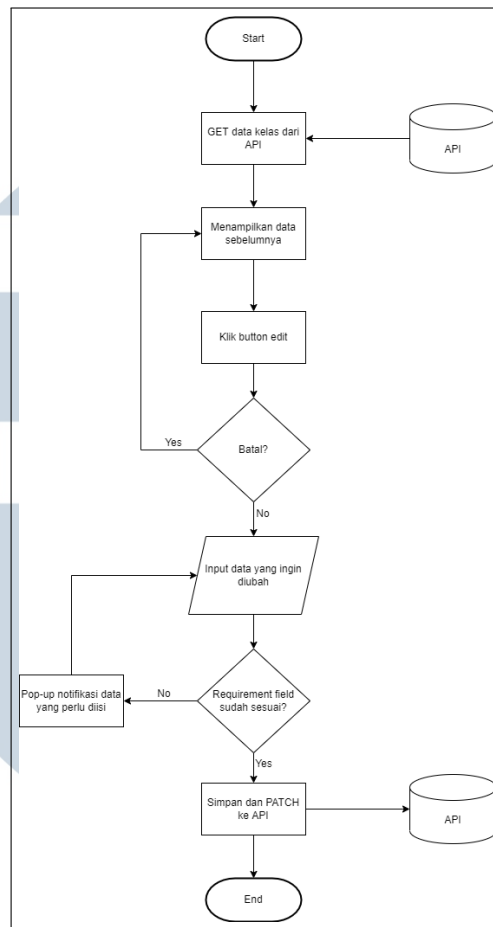
Gambar 3.1 adalah *sitemap* dari halaman *dashboard* guru sebagai gambaran besar bagaimana *flow* dari yang bisa dilakukan pada halaman ini. *Dashboard* guru hanya dapat diakses oleh akun yang mempunyai *role* guru saja, di *dashboard* ini

guru memiliki berbagai fitur diantaranya dapat membuat *course* dan menampilkan *course* yang sudah dibuat oleh guru tersebut. *Flowchart* dari proses pembuatan *course* oleh guru bisa dilihat pada Gambar 3.2



Gambar 3.2. *Flowchart create course*

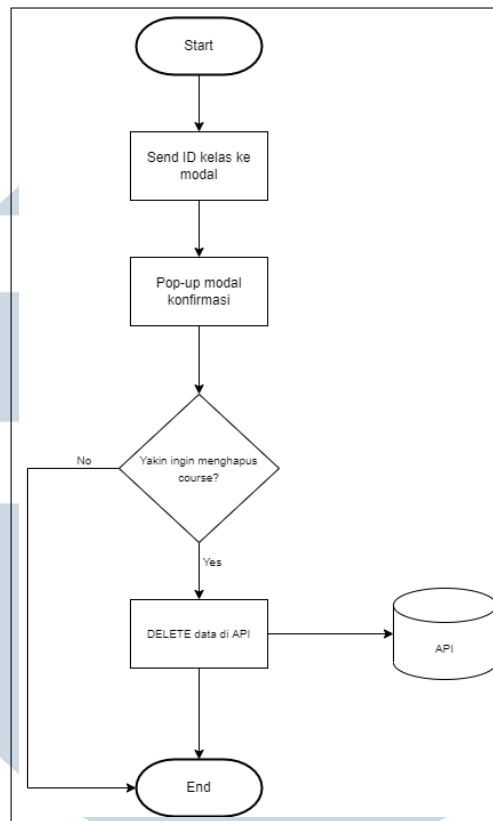
Guru juga bisa melakukan *edit* pada *course* yang sudah dibuat. Proses fitur *edit* dimulai dari menerima parameter id yang sudah dikirim dari halaman sebelumnya untuk menampilkan data dalam *field* yang tersedia tetapi dalam bentuk yang tidak bisa diubah, jika ingin mengubah maka harus menekan tombol *edit* terlebih dahulu baru bisa diubah kolom yang ada, setelah selesai mengisi data yang ingin dirubah baru data di *patch* ke API. *Flow* fitur *edit* bisa dilihat pada *flowchart* di Gambar 3.3.



Gambar 3.3. Flowchart edit course

Guru bisa menghapus *course* yang sudah tidak digunakan lagi. Fitur menghapus data dari API diperlukan mengirim id *course* terlebih dahulu kemudian akan muncul modal untuk konfirmasi apakah yakin untuk dihapus atau tidak baru data dihapus dari API. Proses *flow* fitur *delete* bisa dilihat pada *flowchart* di Gambar 3.4

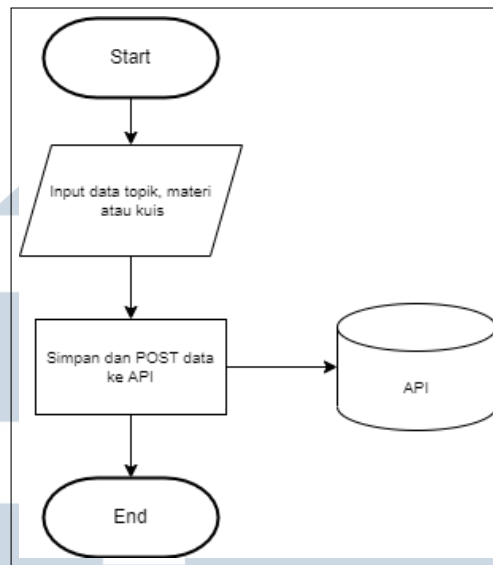
UNIVERSITAS
MULTIMEDIA
NUSANTARA



Gambar 3.4. Flowchart delete course

Setelah proses CRUD pada bagian *course* guru bisa menambahkan isi konten dari *course* tersebut dengan menekan logo tambah yang terdapat pada kolom aksi di tabel *list course*, yang didalamnya terdapat topik, materi, dan kuis. Konten di dalam *course* tersebut juga bisa dilakukan proses CRUD, pembuatan ketiga fitur ini memiliki *flow* yang sama hanya saja data yang dimasukan berbeda oleh karena itu *flowchart* yang akan ditampilkan dibawah ini dijadikan satu. *Flow* proses *create* bisa dilihat pada *flowchart* di Gambar 3.5

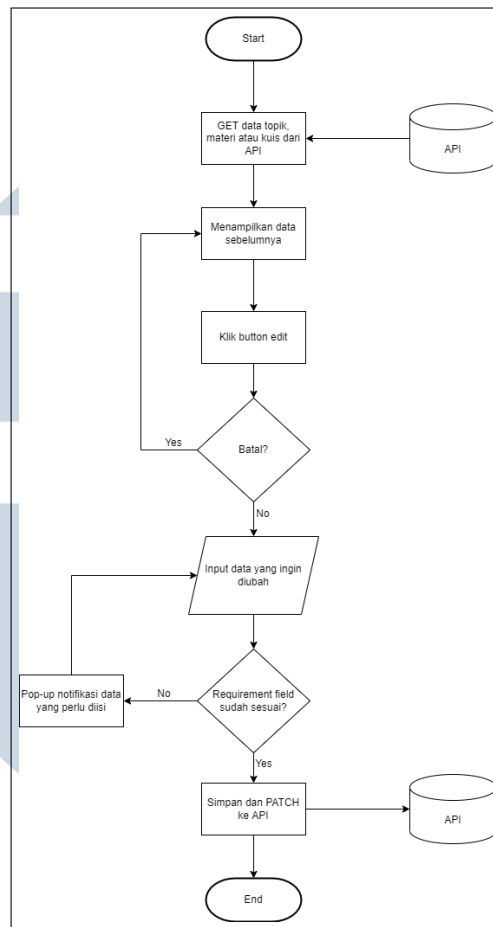
UNIVERSITAS
MULTIMEDIA
NUSANTARA



Gambar 3.5. *Flowchart create topik, materi, dan kuis*

Guru bisa melakukan *edit* pada data dalam ketiga fitur tersebut. Proses berjalannya fitur ini dimulai dengan mengambil parameter id yang dikirim dari halaman sebelumnya untuk dapat melakukan *get* data untuk ditampilkan terlebih dahulu kemudian pengguna bisa menekan tombol ubah agar dapat menulis data yang ingin diganti, setelah itu menekan tombol simpan agar data bisa di *post* kedalam API dengan menggunakan parameter id yang ada supaya data yang dikirim masuk sesuai dengan topik, materi, atau kuis yang ingin diubah. *Flow edit* bisa dilihat pada *flowchart* di Gambar 3.6

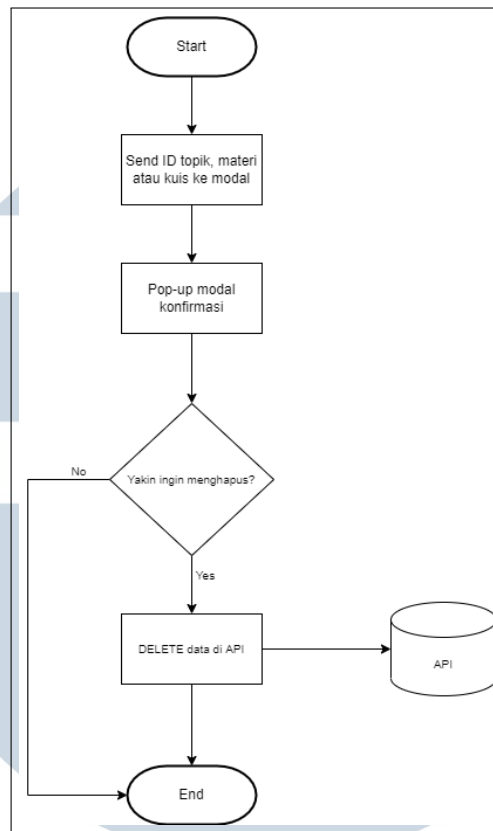
U M I N
 U N I V E R S I T A S
 M U L T I M E D I A
 N U S A N T A R A



Gambar 3.6. Flowchart edit topik, materi, dan kuis

Guru juga bisa melakukan *delete* pada data ketiga fitur yang sudah tidak digunakan kembali. Pada fitur ini proses yang berjalan tidak berbeda jauh dengan fitur *edit* hanya saja yang dilakukan adalah mengirim parameter id untuk melakukan *delete* pada data yang terdapat di API. *Flow delete* bisa dilihat pada *flowchart* di Gambar 3.7

UNIVERSITAS
MULTIMEDIA
NUSANTARA



Gambar 3.7. Flowchart delete topik, materi, dan kuis

Selain isi dari *course* yang bisa dibuat oleh guru, terdapat juga dua fitur tambahan untuk membantu guru yaitu, fitur *list* siswa untuk menampilkan nama-nama siswa yang terdaftar pada *course* tersebut dan fitur *feedback* untuk menampilkan perasaan siswa selama proses belajar dari konten yang ada didalam *course* tersebut.

Fitur nilai di dalam *dashboard* guru dibuat dengan mengambil data nilai dari API yang ada kemudian ditampilkan secara dinamis di dalam tabel yang sudah dibuat HTMLnya, berikut adalah potongan codingan untuk tabel yang sudah dirubah menjadi dinamis

UNIVERSITAS
MULTIMEDIA
NUSANTARA

```
75 <tbody>
76   {currentItems === null ? <Spinner color="success"/> : currentItems !== null ? (
77     currentItems
78     .filter((item) => {
79       if (search.length < 1) {
80         return item;
81       } else if (
82         item.answer_detail.quiz_detail.topic_detail.course_detail.title
83           .toLowerCase()
84           .includes(search.toLowerCase()) ||
85         item.answer_detail.quiz_detail.name
86           .toLowerCase()
87           .includes(search.toLowerCase())
88       ) {
89         return item;
90       }
91     })
92     .map((item, z) => (
93       <tr key={item.id}>
94         <td className="text-center">{z + 1}</td>
95         <td className="text-center">
96           {item.created_at}
97         </td>
98         <td>
99           {
100             item.answer_detail.quiz_detail
101               .topic_detail.course_detail.title
102           }
103         </td>
104         <td className="text-center">
105           {item.answer_detail.quiz_detail.name}
106         </td>
107         <td className="text-center">
108           <span
109             style={
110               item.score >= 70
111                 ? { color: "#0fb800" }
112                 : { color: "#ff1100" }
113             }
114           >
115             <b>
116               {item.score !== null ? item.score : "0"}
117             </b>
118           </span>
119         </td>
120       </tr>
121     ))

```

Gambar 3.8. Potongan coding nilai

Pengambilan data menggunakan *library axios* untuk melakukan GET API yang kemudian disimpan kedalam sebuah parameter *useState*, berikut adalah potongan codingan untuk proses pengambilan data ke API.

```

157     useEffect(() => {
158         const token = localStorage.getItem(LOCALSTORAGE_TOKEN);
159         const options = {
160             headers: {
161                 "Content-Type": "application/json",
162                 Authorization: `Token ${token}`,
163             },
164         };
165         axios
166             .get(`${URL_SCOREQUIZ_BYUSER}`, options)
167             .then((res) => {
168                 const endOffset = itemOffset + page;
169                 setCurrentItems(res.data.slice(itemOffset, endOffset));
170                 setPageCount(Math.ceil(res.data.length / page));
171                 setScore(res.data);
172                 setLoading(true);
173             })
174             .catch((err) => {
175                 console.log(err);
176             });
177     }, [itemOffset, page]);

```

Gambar 3.9. Potongan coding pengambilan data menggunakan *axios*

Fitur riwayat pembayaran dibuat sama seperti tabel nilai. Mekanisme pengambilan datanya juga sama, yang berbeda adalah terdapat *handling button* di status pembayaran yang memiliki fungsi berbeda, seperti saat sudah lunas akan diarahkan ke halaman pembelajaran dan jika belum lunas akan diarahkan ke halaman pembayaran yang menggunakan *third party xendit*. Berikut adalah potongan codingan untuk *handling* status pembayaran.

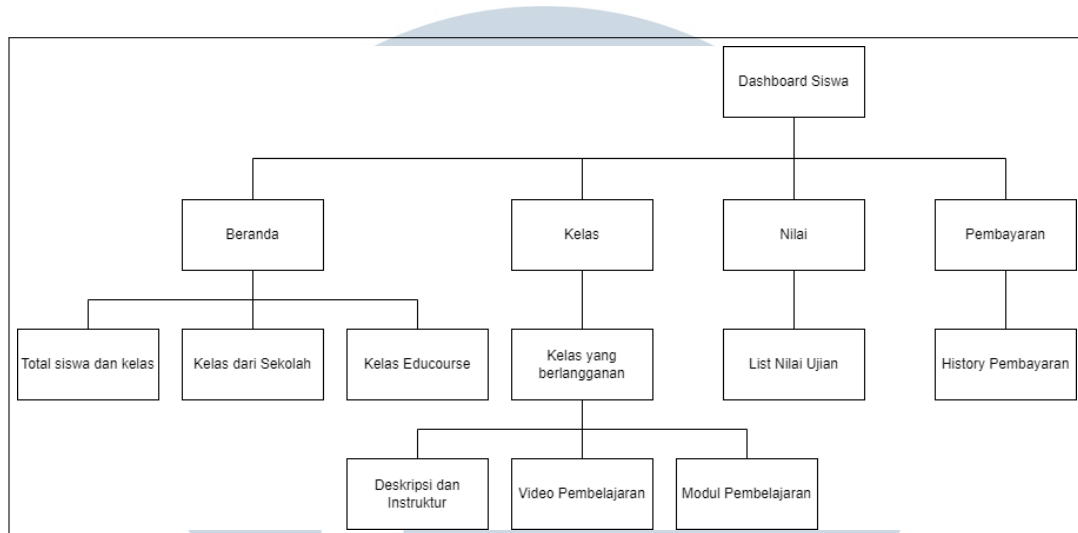
UNIVERSITAS
MULTIMEDIA
NUSANTARA

```
<td className="text-center">
  {item.status === "Lunas" &&
    (myUser.role === "Student" ? [
      <div>
        <Button className="default-mulaibelajar ">
          <Link href="/dashboard/student/mycourse">
            {t('dashboard_teacher.purchased.start')}
          </Link>
        </Button>
      </div>
      : (
        <div>
          <Button className="default-mulaibelajar ">
            <Link href="/dashboard/teacher/courses/listing">
              {t('dashboard_teacher.purchased.start')}
            </Link>
          </Button>
        </div>
      )
    )}
  {item.status === "EXPIRED" && (
    <div>
      <Button className="default-expired">
        <Link href="/courses/listing/list-all-category">
          {t('dashboard_teacher.purchased.find')}
        </Link>
      </Button>
    </div>
  )}
  {item.status === "Belum Lunas" && (
    <div>
      <Button className="default-belumlunas">
        <Link href={item.payment_url}>
          {t('dashboard_teacher.purchased.pay')}
        </Link>
      </Button>
    </div>
  )}
</td>
```

Gambar 3.10. Potongan coding *handling* status pembayaran

UIN
UNIVERSITAS
MULTIMEDIA
NUSANTARA

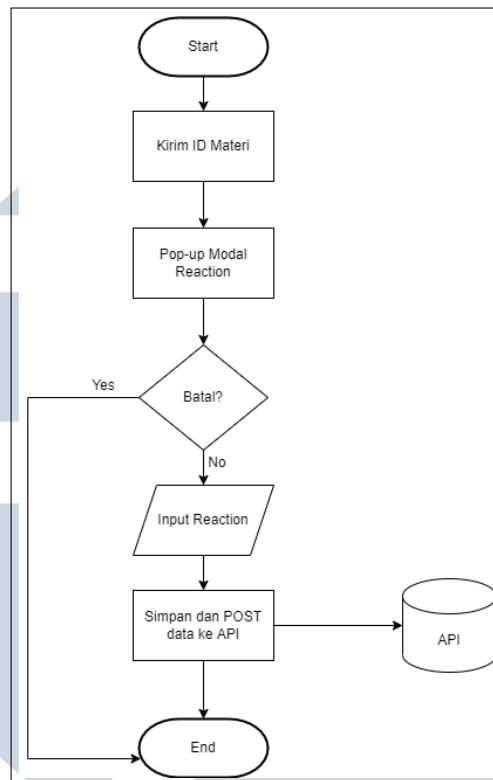
B. Dashboard Siswa



Gambar 3.11. Sitemap dashboard siswa

Gambar 3.8 adalah *sitemap* dari *dashboard* siswa, pada halaman ini siswa melakukan proses belajar menggunakan video dan modul yang sudah disediakan oleh guru, dibagian ini pekerjaan yang dilakukan adalah integrasi pengambilan data kelas, video dan modul kedalam halaman tersebut. Data kelas yang sudah berlangganan oleh siswa ditampilkan di halaman *my course*, data sudah di sortir di *back-end* sehingga hanya perlu melakukan *get* dari API yang sudah disediakan, kemudian setelah kelas ditekan akan masuk kedalam halaman pembelajaran disini pekerjaan yang dilakukan adalah *get* dari API konten *course* yang sudah disediakan dan melakukan *post feedback* dari siswa ke API sebagai data respon siswa terhadap materi yang dipelajari agar guru dapat meningkatkan kualitas dari materi yang sudah dibuat, untuk *flow feedback* dapat dilihat di Gambar 3.9

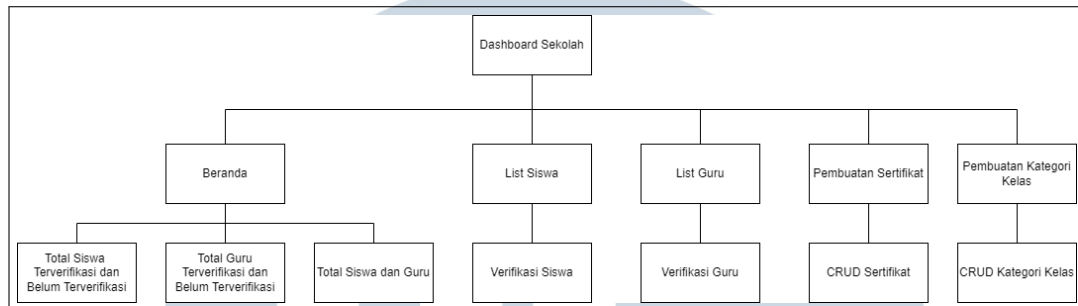
UNIVERSITAS
MULTIMEDIA
NUSANTARA



Gambar 3.12. *Flowchart feedback*

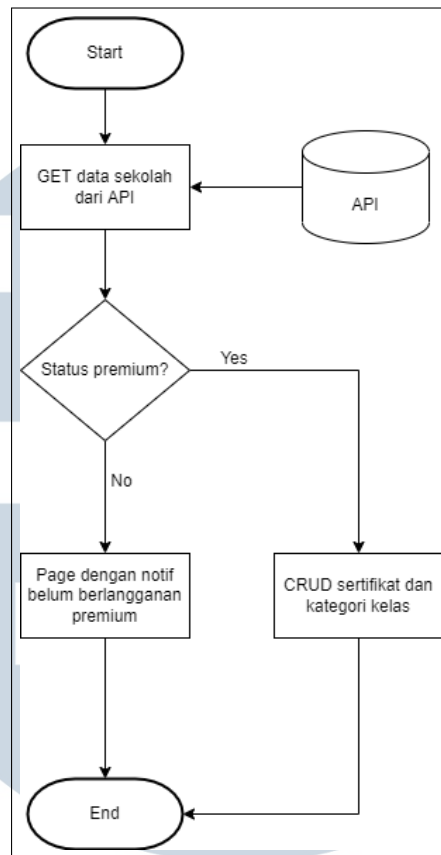
Siswa juga bisa melihat nilai dari kuis yang sudah diikuti atau dikerjakan pada halaman ini integrasi yang dilakukan hanya melakukan *get* dari API, jika nilai siswa tidak mencapai standar nilai yang sudah ditentukan siswa bisa mengambil kuis lagi sampai mendapatkan nilai diatas standar. Pada *dashboard* ini siswa bisa melihat sertifikat yang didapat setelah menyelesaikan *course* yang diambil, pada halaman ini integrasi yang dilakukan hanya melakukan *get* dari API, siswa juga dapat mengunduh sertifikatnya ke perangkat masing-masing, selain itu siswa bisa melihat riwayat pembelian *course* yang dibeli, pada halaman ini siswa juga bisa menunda pembayaran sampai waktu yang sudah ditentukan. Integrasi yang dilakukan hanya melakukan *get* dari API.

C. Dashboard Sekolah



Gambar 3.13. Sitemap dashboard sekolah

Gambar 3.10 adalah *sitemap* dari halaman *dashboard* sekolah, halaman ini akan dipegang oleh *admin* sekolah, halaman ini dibuat sebagai salah satu bentuk keamanan untuk pihak sekolah dimana guru dan siswa yang mendaftar perlu diverifikasi terlebih dahulu oleh pihak sekolah agar tidak ada oknum dari luar sekolah yang bisa masuk kedalam LMS tersebut. Adapun integrasi yang dilakukan untuk fitur verifikasi hanya melakukan *get* data dari API siswa dan guru yang mendaftar dan pihak *admin* sekolah bisa melakukan *edit* di setiap siswa dan guru untuk merubah status verifikasi menjadi ter-verifikasi atau verifikasi ditolak, selain fitur verifikasi sekolah juga bisa membuat sertifikat khusus sekolah sendiri dan kategori kelas khusus dari sekolah masing-masing, tetapi hanya bisa digunakan jika sekolah mengambil LMS dengan paket *premium*, disini *handling* dilakukan sesuai dengan status berlangganan LMS dari setiap sekolah, untuk *flow handling* bisa dilihat pada *flowchart* di Gambar 3.11



Gambar 3.14. Flowchart handling halaman sekolah

terdapat juga potongan codingan untuk *handling* halaman sekolah berdasarkan status sekolah tersebut untuk dapat mengakses beberapa fitur di LMS tersebut. Cara melakukan *handling* adalah dengan mengambil data status sekolah dari API yang kemudian disimpan dalam sebuah parameter kemudian parameter tersebut yang digunakan sebagai acuan halaman memberikan akses fitur tersebut atau tidak. berikut adalah potongan codingannya.

```

const localdata = JSON.parse(localStorage.getItem(LOCALSTORAGE_USERDETAIL));
setMyUser(localdata);
setAllowCreateVoucher(localdata.school_detail.package);
  
```

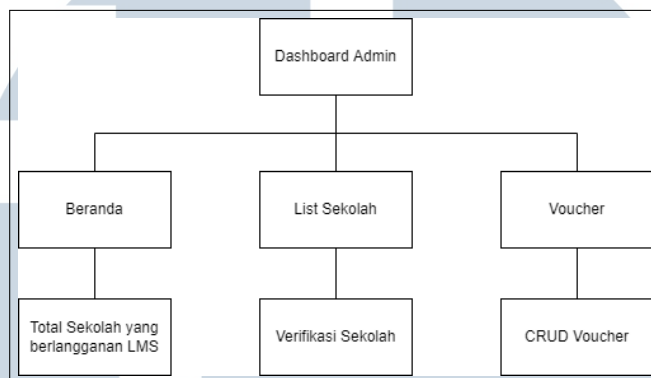
Gambar 3.15. Pengambilan data status

```

301 {`${allowCreateVoucher}`.includes("Premium") ? (
  
```

Gambar 3.16. Potongan coding *handling* status sekolah

D. *Dashboard Admin*



Gambar 3.17. Sitemap dashboard admin

Gambar 3.12 adalah *sitemap* dari *dashboard admin*, halaman ini hanya bisa diakses oleh *admin* Educourse bertujuan untuk melakukan verifikasi sekolah yang mendaftar agar tidak ada sekolah-sekolah lain yang belum melakukan tanda tangan MOU (*Memorandum of Understanding* mengakses LMS ini. *Admin* bisa melakukan *edit* untuk merubah status sekolah dari belum ter-verifikasi menjadi sudah ter-verifikasi.

Integrasi yang dilakukan di halaman verifikasi sekolah diantaranya melakukan *get* data dari API untuk menunjukkan *list* sekolah beserta status verifikasi sekolah pada tabel yang ada, kemudian membuat *edit* page untuk *admin* bisa merubah status verifikasi sekolah, untuk *flow edit* sama seperti di halaman verifikasi siswa dan guru di *dashboard* sekolah, selain itu *admin* Educourse juga bisa melakukan CRUD pada *voucher* untuk potongan harga yang akan dibuat untuk *course* Educourse. *Flow* CRUD *voucher* sama seperti *flow* CRUD di bagian *course* di *dashboard* guru.

3.3.3 Pekerjaan Umum yang Dilakukan didalam LMS

Semua tabel yang ada di LMS ini pertama diberikan dalam bentuk statis atau di *hard code* kemudian diubah menjadi dinamis dengan cara menyimpan data kedalam parameter berbentuk *array* kemudian di *mapping* dalam tabel. berikut adalah salah satu contoh potongan codingan tabel yang dirubah menjadi dinamis.

A. Mengubah Tabel Statis Menjadi Dinamis

Pada LMS ini sudah dibuat HTML dan CSS dari UI/UX *designer* yang dilakukan adalah mengubah form, tabel, dan tampilan lainnya dari yang statis atau di *hard code* menjadi dinamis, mengubah tampilan yang dirasa kurang cocok didalam LMS, menampilkan data dari *back-end* ke halaman antarmuka dan menambahkan beberapa fitur pembantu untuk LMS seperti *pagination*, *search* dan *translate* semua page.

```
useEffect(() => {
  const token = localStorage.getItem(LOCALSTORAGE_TOKEN);
  const options = {
    headers: {
      "Content-Type": "application/json",
      Authorization: `Token ${token}`,
    },
  };
  axios
    .get(URL_INSTRUCTOR_BYUSER, options)
    .then((res) => {
      const endOffset = itemOffset + page;
      setCurrentItems(res.data.slice(itemOffset, endOffset));
      setPageCount(Math.ceil(res.data.length / page));
      setCourseListTeacher(res.data);
    })
    .catch((err) => {
      console.log(err);
    });
}, [itemOffset, page]);
```

Gambar 3.18. Pengambilan data

```
255 const [courseListTeacher, setCourseListTeacher] = useState([]);
```

Gambar 3.19. Penyimpanan data

```

.map((item, z) => (
  <tr key={item.course_detail.id}>
    <td>{z + 1}</td>
    <td>{item.course_detail.course_code}</td>
    <td>
      {" "}
      <Link
        href='/dashboard/teacher/courses/preview/[id]'
        as={` /dashboard/teacher/courses/preview/${item.course_detail.id}`}>
      <a title={t("dashboard_teacher.course_listing.click")}>
        <b className='course_active'>
          {item.course_detail.title}
        </b>
      </a>
    </Link>
    </td>
    <td>{item.course_detail.category_detail.name}</td>
    <td>
      {item.course_detail.valid_start}~
      {item.course_detail.valid_end}
    </td>
    <td className="text-center">
      {`${item.course_detail.isPublish}`.includes("true")
        ? "√"
        : "X"}
    </td>
    <td className="text-center">
      {item.course_detail.price === 0 ? (
        "Gratis"
      ) : (
        <div>
          <NumberFormat
            // className="font-weight-black"
            value={item.course_detail.price}
            displayType={"text"}
            thousandSeparator={true}
            prefix={"Rp"}
          />
        </div>
      )}
    </td>
  </tr>
)
)

```

Gambar 3.20. Mapping data

B. Mengubah *Form* dari Statis Menjadi Dinamis

Form yang dibuat dalam bentuk statis dikirim, kemudian mengubahnya menjadi bentuk dinamis dan bisa mengirim data ke API yang ada menggunakan *library axios* dengan metode POST. Cara melakukan pengiriman data ke API diawali dengan memasukkan data kedalam sebuah parameter *array* yang kemudian parameter tersebut di masukan kedalam *axios* untuk di POST ke API. berikut adalah salah satu contoh potongan codingannya.

```

const handleSubmit = async (e) => {
  e.preventDefault();
  const data = new FormData();
  data.append("course_code", e.target.course_code.value);
  data.append("valid_start", e.target.valid_start.value);
  data.append("valid_end", e.target.valid_end.value);
  data.append("title", e.target.title.value);
  data.append("overview", content);
  if (e.target.gradeSchoolId) {
    data.append("gradeSchoolId", e.target.gradeSchoolId.value);
  }
  data.append("price", e.target.price.value);
  data.append("categories_img", e.target.categories_img.files[0]);
  data.append("courses_img", e.target.courses_img.files[0]);
  data.append("courses_small_img", e.target.courses_small_img.files[0]);
  data.append("course_preview_video", e.target.course_preview_video.files[0]);
  data.append("duration", e.target.duration.value);
  data.append("lesson_hours", e.target.lesson_hours.value);
  data.append("isPublish", checked);
  // data.append("access", e.target.access.value);
  data.append("category", e.target.category.value);
  data.append("subcategory", e.target.subcategory.value);
  data.append("userId", myUser.id);
}

```

Gambar 3.21. Penyimpanan data kedalam parameter

```

136      axios
137      .post(URL_COURSES_LIST, data, options)
138      .then((response) => {
139        setLoading(false);
140        setLoading(true);
141        // create table enrolled teacher
142        if (myUser.role == "Teacher") {
143          const dataEnrolled = new FormData();
144          dataEnrolled.append("courseId", response.data.id);
145          dataEnrolled.append("userId", myUser.id);
146          dataEnrolled.append("position", myUser.role);
147          console.log(data);
148          axios
149            .post(URL_INSTRUCTOR, dataEnrolled, options)
150            .then((res) => {
151              setLoading(false);
152            })
153            .catch((err) => {
154              console.log("err", err);
155            });
156        }

```

Gambar 3.22. Pengiriman data dengan metode *post*

```

244     <div className='col-lg-12 col-md-12'>
245       <div className='ui search focus mt-30 lbel25'>
246         <label>{t("dashboard_teacher.create_course.title")}*</label>
247         <div className='ui left icon input swdh19'>
248           <input
249             className='prompt srch_explore'
250             style={{ textTransform: "capitalize" }}
251             type='text'
252             placeholder={t("dashboard_teacher.create_course.title")}
253             name='title'
254             data-purpose='edit-course-title'
255             maxLength='60'
256             id='main[title]'
257             required
258           />
259           <div className='badge_num'>60</div>
260         </div>
261         <div className='help-block'>
262           |({t("dashboard_teacher.create_course.max")})
263         </div>
264       </div>
265     </div>

```

Gambar 3.23. Potongan coding *field form*

C. Membuat fitur *Search* dalam Tabel

Pada setiap tabel dibuat fitur *search* untuk mempermudah pengguna untuk mencari data didalam tabel. Cara membuat *search* dengan melakukan pencocokan variabel dalam kolom *search* dengan data yang ditampilkan didalam tabel, berikut adalah salah satu contoh potongan codingan *search*.

```

45     <div className='ui left icon input swdh11 swdh15'>
46       <input
47         onChange={(e) => setSearch(e.target.value)}
48         className='prompt srch_explore'
49         type='text'
50         placeholder={t(
51           "dashboard_teacher.course_listing.find_here"
52         )}
53       />
54       <i className='uil uil-search-alt icon icon8'></i>
55     </div>

```

Gambar 3.24. Potongan coding kolom *search*

```

90     {currentItems === null ? (
91       <Spinner
92         color='success'
93         style={{ textAlign: "center", verticalAlign: "middle" }}
94       />
95     ) : currentItems !== null ? (
96       currentItems
97         .filter((item) => {
98           if (search.length < 1) {
99             return item;
100          } else if (
101            item.course_detail.title
102              .toLowerCase()
103                .includes(search.toLowerCase()) ||
104            item.course_detail.course_code
105              .toLowerCase()
106                .includes(search.toLowerCase()) ||
107            item.course_detail.category_detail.name
108              .toLowerCase()
109                .includes(search.toLowerCase())
110          ) {
111            return item;
112          }
113        })

```

Gambar 3.25. Potongan coding proses *filter*

D. Membuat Fitur *Pagination* dalam Tabel

Pada setiap tabel dibuat fitur *pagination* agar data yang ditampilkan jika terlalu banyak tidak akan memakan tempat di satu halaman sehingga terlihat lebih nyaman oleh pengguna. Untuk fitur ini menggunakan *library React Paginate*, berikut adalah salah satu potongan codingan *pagination* di tabel.

```

21   import ReactPaginate from "react-paginate";

```

Gambar 3.26. *Import library pagination*


```

256 // We start with an empty list of items.
257 const [currentItems, setCurrentItems] = useState(null);
258 const [pageCount, setPageCount] = useState(0);
259 // Here we use item offsets; we could also use page offsets
260 // following the API or data you're working with.
261 const [itemOffset, setItemOffset] = useState(0);
262 const [page, setPage] = useState(10);
263

```

Gambar 3.27. Deklarasi parameter yang digunakan

```

272 axios
273   .get(URL_INSTRUCTOR_BYUSER, options)
274   .then((res) => {
275     const endOffset = itemOffset + page;
276     setCurrentItems(res.data.slice(itemOffset, endOffset));
277     setPageCount(Math.ceil(res.data.length / page));
278     setCourseListTeacher(res.data);
279   })
280   .catch((err) => {
281     console.log(err);
282   });
283 }, [itemOffset, page]);

```

Gambar 3.28. Proses penghitungan *page*

UMMN
UNIVERSITAS
MULTIMEDIA
NUSANTARA

```
296 <DataCoursesTeacher currentItems={currentItems} /> <br />
297 <Row>
298   <Col></Col>
299   <Col md='auto'>
300     <ReactPaginate
301       breakLabel='...'
302       nextLabel='>'
303       onPageChange={handlePageClick}
304       pageRangeDisplayed={5}
305       pageCount={pageCount}
306       previousLabel='<'
307       renderOnZeroPageCount={null}
308       containerClassName={"paginationBttns"}
309       previousLinkClassName={"previousBttns"}
310       nextLinkClassName={"nextBttn"}
311       disabledClassName={"paginationDisabled"}
312       activeClassName={"paginationActive"}
313     />
314   </Col>
315   {courseListTeacher.length !== 0 ? (
316     <Col xs lg='2'>
317       <select
318         name='page'
319         onChange={(e) => setPage(e.target.value)}
320         style={{ height: "33px" }}>
321         <option value='10' style={{ textAlign: "center" }}>
322           10
323         </option>
324         <option value='20' style={{ textAlign: "center" }}>
325           20
326         </option>
327         <option value='30' style={{ textAlign: "center" }}>
328           30
329         </option>
330         <option value='40' style={{ textAlign: "center" }}>
331           40
332         </option>
333         <option
334           value={courseListTeacher.length}
335           style={{ textAlign: "center" }}>
336           {t("dashboard_teacher.course_listing.all")}
337         </option>
338       </select>
339     </Col>
```

Gambar 3.29. Menampilkan *page* tabel

E. Membuat *Translate* Halaman LMS

LMS Educourse dibuat fitur bahasa sehingga pengguna bisa membaca LMS dalam bahasa Inggris atau Indonesia. Pembuatan fitur ini menggunakan *library React-i18next* dengan membuat dua file bahasa yang akan digunakan kemudian mengganti kata yang di *hard code* dengan kata yang sudah di *translate* di file baru tersebut. Berikut adalah salah satu contoh potongan coding fitur *translate*.

```
22 import { useTranslation } from "react-i18next";
```

Gambar 3.30. Import library *translate*

```
25 const { t } = useTranslation("common");
```

Gambar 3.31. Deklarasi parameter data *translate*

```
1 {
2   "landing_page": {
3     "popular_courses": {
4       "best_course": "Kelas Terbaik Educourse",
5       "talent": "Siap jadi talenta digital terbaik!",
6       "pilih_puluhan": "Pilih puluhan kelas video online menggunakan Asynchronous Learning dengan standar 13P : 10 menit. Temukan kelas barumu disini!",
7       "segera_daftarkan": "Segera daftarkan Sekolah Anda untuk mendapatkan benefit terbaik.",
8       "daftar": "Daftar disini",
9       "gratis": "Gratis",
10      "materi": "Materi",
11      "peserta": "Peserta",
12      "belum_ada_kelas": "Belum ada kelas yang bisa Anda ikuti",
13      "tunggu_kelas": "Tunggu kelasnya agar bisa belajar di kelas ini!",
14      "kelas_lain": "Lihat Kelas Lainnya",
15      "yuk": "Yuk, cari kelasnya dan belajar sekarang juga!"
16    },
17    "navber_dashboard": {
18      "halo": "Halo,",
19      "logout": "Keluar",
20      "login": "Login"
21    },
22    "navber_page": {
23      "kelas": "Kelas",
24      "login_regis": "Masuk/Daftar",
25      "semua_kelas": "Semua Kelas",
26      "halo": "Halo,",
27      "dashboard": "Beranda",
28      "profil": "Profil",
29      "logout": "Keluar"
30    },
31    "footer": {
32      "company_profile": "Educourse adalah perusahaan edutech di Indonesia yang menyediakan dan memberikan solusi untuk mempercepat kualitas pembelajaran di Indonesia dengan menggunakan Pendidikan
33      "semua_kelas": "Semua Kelas",
34      "alamat": "Alamat",
35      "kelas": "Kelas"
36    },
37    "feedback": {
38      "testi": "Testimonial",
39      "edubuddy": "Apa kata Edubuddy?",
40      "say": "Kata orang tua dan siswa setelah belajar sambil bermain bersama di Educourse!"
41    },
42    "instance": {
43      "teacher": "Kompetensi dan Pedagogik Guru",
44      "teacher_1": "Ayo Tingkatkan kompetensi Guru!",
45      "teacher_2": "Guru harus memiliki kompetensi yang akan menunjang tugas profesionalnya. Mari tingkatkan kompetensi dirimu sebagai guru untuk dapat mencerdaskan anak-anak agar lebih terarah.",
46      "regis": "Daftar Disini"
47    }
48  }
49 }
```

Gambar 3.32. Data bahasa Indonesia

```

1 {
2   "landing_page": {
3     "popular_courses": {
4       "best_course": "Our Top Courses",
5       "talent": "READY TO BE BEST DIGITAL TALENT?",
6       "pilih_pilihan": "Choose from dozens of online video classes using Asynchronous Learning with the standard lesson hours for 10 minutes. Find your new class here!",
7       "segera_daftarkan": "Immediately register your school to get the best benefits.",
8       "daftar": "Register here",
9       "gratis": "Free",
10      "materi": "Learning Materials",
11      "peserta": "Participant",
12      "belum_ada_kelas": "There are no classes you can take yet",
13      "tunggu_kelas": "Wait for the class so you can study in this class!",
14      "kelas_lain": "See More Classes",
15      "yuk": "Come find the class and learn right now!"
16    },
17    "navbar_page": {
18      "kelas": "Class",
19      "login_regis": "Login/Register",
20      "semua_kelas": "All Classes",
21      "halo": "Hello",
22      "dashboard": "Dashboard",
23      "profile": "Profile",
24      "logout": "Logout"
25    },
26    "footer": {
27      "company_profile": "Educourse is an edutech company in Indonesia that provides solutions to accelerate the quality of learning in Indonesia by using STEAM-based education supported by AI",
28      "semua_kelas": "All Classes",
29      "alamat": "Address",
30      "kelas": "Class"
31    },
32    "feedback": {
33      "testis": "Testimonials",
34      "edubuddy": "What did Edubuddy say?",
35      "say": "Parents and students say after learning while playing together on Educourse!"
36    },
37    "instance": {
38      "teacher": "Teacher Competencies and Pedagogics",
39      "teacher_1": "Let's Improve Teacher's Competence",
40      "teacher_2": "Teachers must have competencies that will support his professional duties. Let's increase your competence as a teacher to be able to educate children to be more focused.",
41      "regis": "Register Here"
42    },
43    "main_banner": {
44      "title": "Education Platform with STEAM method for School",
45      "text": "Let's make children and teachers with digital talents to face the challenges of the future."
46    }
47  }

```

Gambar 3.33. Data bahasa Inggris

```

<h6>
  {{t('dashboard_teacher.course_listing.cant_create_class')}}
</h6>

```

Gambar 3.34. Potongan coding menggunakan *translate*

3.3.4 Pekerjaan Setelah Pembuatan LMS

Setelah LMS selesai dibuat kegiatan yang dilakukan adalah *testing* pada keseluruhan fitur yang ada didalam LMS, lalu mencatat semua *bug* dan kesalahan penulisan yang dibuat, kemudian melakukan pembagian tugas untuk melakukan perbaikan *bug* yang ditemukan, setelah perbaikan *bug* selesai dilakukan tim *tech* mempresentasikan cara kerja LMS kepada tim Educourse dan juga beberapa sekolah yang tertarik untuk menggunakan LMS tersebut.

3.3.5 Pekerjaan diluar Proyek LMS

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

1. NUMED AI, sebuah website dan aplikasi medis yang memiliki *Artificial Intelligence* yang dapat mendeteksi *corona virus disease 2019* dari hasil rontgen paru-paru pasien, di dalam proyek ini pekerjaan yang dilakukan adalah membantu integrasi *get data* dari API, dikarenakan tenggat waktu yang semakin sedikit dan diperlukan bantuan tenaga dalam proyek ini.
2. *Mobile Version LMS*, dalam proyek ini adapun pekerjaan tambahan yaitu membantu melakukan integrasi CRUD pada LMS berbasis *mobile* menggunakan bahasa *flutter*.

3.4 Kendala dan Solusi yang Ditemukan

3.4.1 Kendala

Kendala yang dihadapi selama pelaksanaan kerja magang diantaranya :

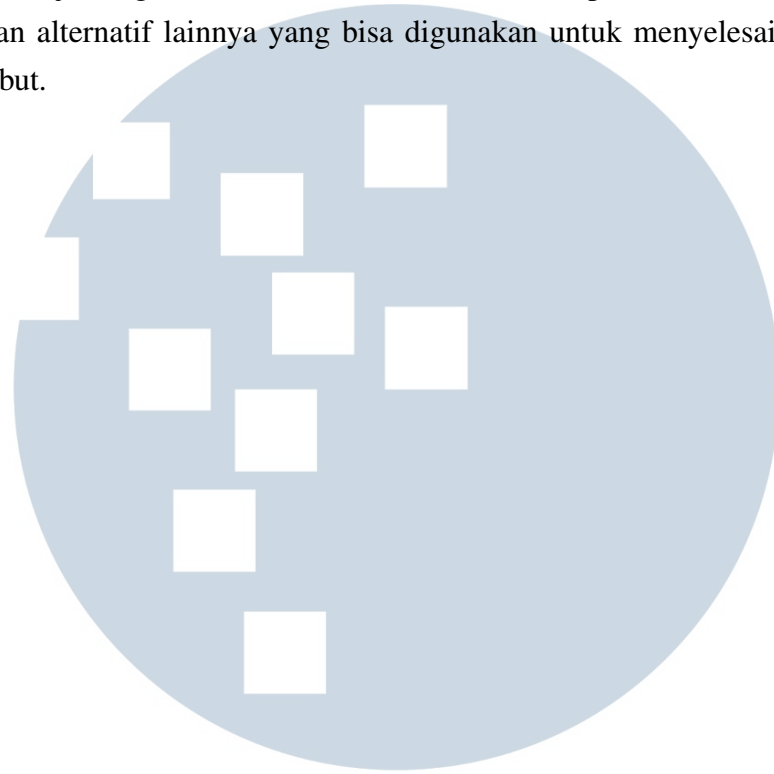
1. Sulit memahami *flow* codingan yang sudah ada, dikarenakan telatnya masuk kedalam project beberapa hari.
2. Sulit menemukan *library* pendukung yang digunakan untuk beberapa fitur seperti *search* dan *pagination* karena masih menggunakan *NextJs* sedangkan kebanyakan *library* bisa digunakan di *ReactJs*.
3. Sulit mengaplikasikan beberapa *library* kedalam codingan karena sedikitnya *documentation* yang disediakan oleh platform tersebut.

3.4.2 Solusi

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

1. Meminta bantuan pembimbing serta rekan-rekan kerja lainnya untuk *flow* yang sudah dibuat.
2. Melakukan diskusi terkait *library* pengganti yang bisa digunakan oleh *NextJs* ataupun mencari solusi lain dari beberapa *platform* seperti *GeeksforGeeks*, *StackOverflow* dan sebagainya.

3. Kesulitan dalam mengaplikasikan beberapa *library* dibantu oleh beberapa rekan kerja dengan malukan diskusi bersama CTO perusahaan sehingga ditemukan alternatif lainnya yang bisa digunakan untuk menyelesaikan bagian tersebut.



UMMN
UNIVERSITAS
MULTIMEDIA
NUSANTARA