BAB 3

PELAKSANAAN KERJA MAGANG

3.1 Kedudukan dan Koordinasi

Dalam pelaksanaan kerja magang di PT Sumber Inovasi Informatika yang menjabat sebagai Front-End Developer intern pada team easySR dengan *team leader* yaitu Bapak Hans Permana selaku COO PT Sumber Inovasi Informatika *dan product owner* yaitu Eriko Anggara. Eriko berperan dalam mengatur tahapan-tahapan dalam membangun *software* dengan menentukan pekerjaan pada setiap *sprint* yang akan dijalankan. Jalur koordinasi untuk rapat menggunakan Google Meet dan Discord. Untuk implementasi fitur-fitur yang telah dibuat menggunakan Gitlab untuk kolaborasi pengerjaan project. Bapak Hans Permana sebagai pembimbing dalam penggunaan React dan juga *reviewer code* saat *branch* tertentu akan di *merge* ke *staging*. Kedudukan pada proses kerja magang di PT Sumber Inovasi Informatika masuk pada tim Project easyDRKPL. Selain itu pada kesempatan kerja magang kali ini juga melakukan konsultasi dengan Bapak Hans dalam pembuatan tiap komponen pada *software*.

3.2 Tugas yang Dilakukan

Selama berlangsungnya kerja magang, berikut tanggung jawab yang diberikan yaitu:

1. Ikut serta dalam melakukan *sprint planning* dan *showcase* hasil *development* kepada *client* yang dilakukan 2 minggu sekali.
2. Implementasi desain dari *mockup* menjadi Front-End easyDRKPL,

3. Melakukan riset pada Blueprintjs untuk Front-End.
4. Melakukan riset pembuatan *file* excel menggunakan XlsxWriter.

3.3 Uraian Pelaksanaan Kerja Magang

3.3.1 Aktivitas Kerja Magang

Kerja magang dilakukan selama 16 minggu dengan timeline kerja sebagai berikut.

Table 3.1 Deskripsi Kerja Magang Per Minggu

Nama Kegiatan	Minggu ke-															
	1	2	3	4	5	6	7	8	9	10	11	12	13	14	15	16
Persiapan environment dan ReactJS																
Riset Blueprintjs dan XlsxWriter																
Implementasi Front-End																

Pada minggu pertama pekerjaan dimulai dengan melakukan persiapan *tools* dan *environment* yang dibutuhkan untuk membangun software dan berkolaborasi dengan anggota tim. Persiapan dilakukan seperti instalasi *tools* dan mempersiapkan akun dari beberapa aplikasi yang dibutuhkan seperti Slack, Trello, Gitlab, OwnCloud, dan WebStorm. Selanjutnya proses mempelajari ReactJS sebagai *library* utama dalam membuat Front-End website tersebut. Dimulai dari membaca dokumentasi dari website ReactJS dan berdiskusi bersama Pak Hans mengenai kesulitan ataupun kebingungan dalam menggunakan ReactJS.

Pada minggu kedua, Setelah proses persiapan selesai, maka dilanjutkan dengan mempelajari struktur *project* yang sudah ada. Pada kasus ini, website sebelumnya yang merupakan easySR memiliki penambahan fitur untuk membuat laporan DRKPL yang menjadi tugas utama agar dapat merealisasikan fitur tersebut. Setelah struktur *project* sudah dimengerti, maka dilanjutkan dengan tahap *development*. Pada tahap ini, setiap ingin merancang rencana kerja akan dimulai mengikuti *sprint planning* untuk menentukan pekerjaan yang akan dilakukan selama 2 minggu hingga *sprint planning* selanjutnya. Selain itu, setiap hari juga mengikuti *stand up* yang dilaksanakan kurang lebih 15 menit untuk mendiskusikan pekerjaan yang mau dikerjakan hari ini dan melaporkan pekerjaan yang sudah dikerjakan dihari sebelumnya.

Pada minggu ketiga dan keempat, proses melakukan riset untuk Blueprintjs dan XlsxWriter. Blueprintjs merupakan *library* yang memiliki komponen-komponen untuk membangun Front-End pada website. Pada kasus ini, dibutuhkan salah satu komponen dari Blueprinjs yaitu komponen table yang berguna untuk menampilkan banyak data angka dan memproses isi data table. Riset yang dilakukan bertujuan untuk memastikan bahwa fitur-fitur yang dibutuhkan telah didukung oleh *library* tersebut. Proses riset dilaksanakan dengan mencoba berbagai skenario seperti fitur mengedit nilai pada sel, *freezing* kolom dan baris, *styling* pada table, *sorting* nilai berdasarkan kolom dan sebagainya. Selanjutnya dilakukan riset terhadap *library* XlsxWriter dengan menggunakan bahasa pemrograman Python. *Library* ini akan digunakan untuk mengubah data dari *database* menjadi file excel secara otomatis. Riset dilakukan dengan membaca dokumentasi asli dari

XlsxWriter dan mencoba menghasilkan file excel dengan format dan gaya sesuai dengan laporan DRKPL.

Pada minggu keempat dan seterusnya, proses kerja magang dilanjutkan dengan dengan melakukan tahap pengembangan *software* mulai dari tampilan dashboard hingga menampilkan halaman laporan yang dapat diisi oleh pengguna. Pada proses pengembangan, menggunakan IDE WebStorm dan untuk *platform* kolaborasi antar *developer* menggunakan gitlab. Setiap pekerjaan yang akan dikerjakan, harus membuat *branch* pada gitlab dengan format nama tertentu. Setelah pekerjaan diselesaikan atau lagi proses pengerjaan akan *commit* dan *push* ke *project* gitlab easyDRKPL.

A. Framework dan Alur Kerja yang Digunakan

Dalam proses pengembangan easyDRKPL, React digunakan sebagai *library* utama pada Front-End aplikasi website. Bahasa pemrograman yang digunakan adalah typescript. Selain itu, dalam meningkatkan kualitas desain pada Front-End, maka digunakan sebuah *framework* yaitu SemanticUI.

Alur kerja yang diterpakan adalah metode agile yang terdiri dari *sprint planning*, *retrospective*, *standup* dan *showcase*. Dalam pelaksanaan *sprint planning*, kelompok *developer* beserta *product owner* akan bertemu dengan *client* dalam proses *requirement gathering* dan menentukan prioritas dari fitur yang akan dikerjakan. *Retrospective* merupakan kegiatan dimana kita mengevaluasi *sprint* yang sudah berlangsung. Proses ini dilakukan oleh seluruh anggota tim dan setiap anggota akan menuliskan beberapa poin yang mereka rasa perlu didiskusikan bersama. Selanjutnya, hasil dari *retrospective* adalah *action* yang akan menjadi

pengingat untuk *sprint* selanjutnya agar melakukan pekerjaan lebih baik dan mengurangi kesalahan yang sama. Agar proses pengembangan tepat waktu dan terukur, maka diadakan *standup* yang dilaksanakan kurang lebih 15 menit untuk membahas pekerjaan yang sudah ataupun mau dikerjakan. Jika ada masalah dalam pengembangan, maka akan dilakukan *pairing* dengan anggota lainnya untuk menyelesaikan masalah tersebut. Diakhir tiap *sprint*, tim akan melakukan *showcase* yang menampilkan hasil kerja pada *sprint* tersebut kepada *client* yang bersangkutan.

B. Perancangan Sistem dan Pengumpulan Kebutuhan

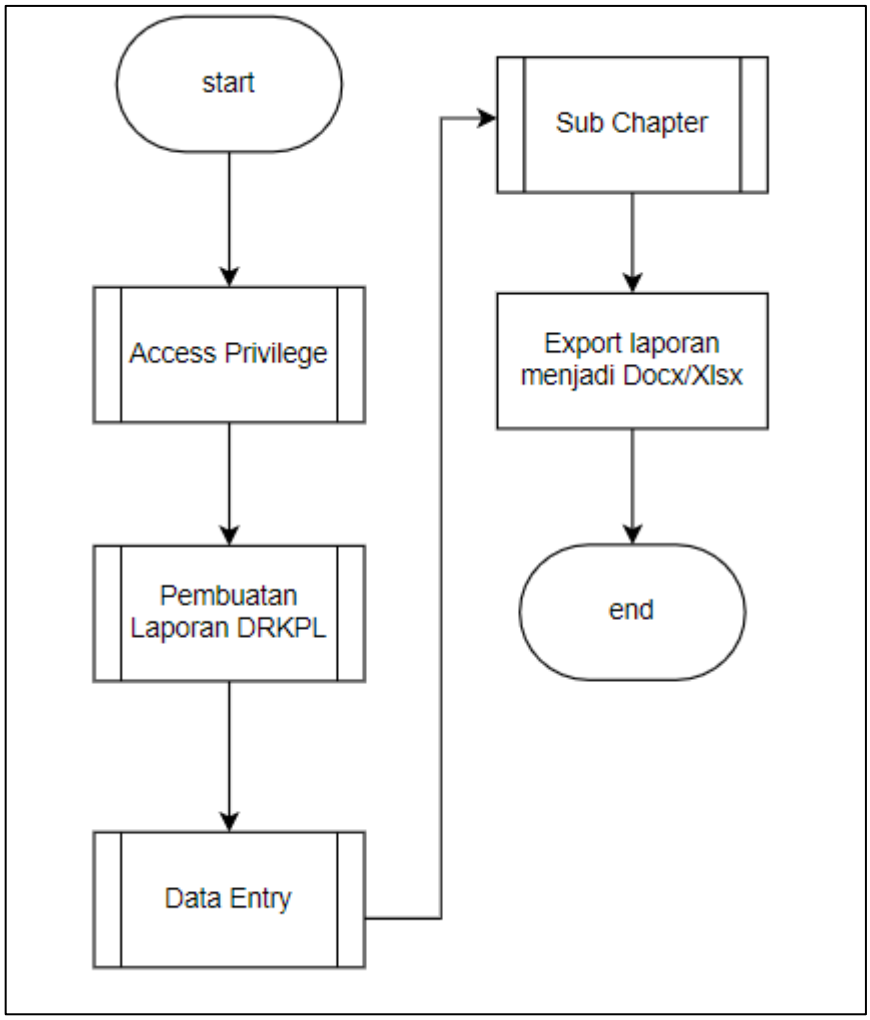
Perancangan sistem diawali dengan mengumpulkan kebutuhan dari *user*. Kebutuhan-kebutuhan tersebut diambil dari proses diskusi bersama *client*. Hasil dari proses diskusi tersebut adalah fitur-fitur yang akan diimplementasi pada easyDRKPL. Fitur-fitur tersebut adalah membuat laporan berdasarkan tahun tertentu, melakukan pengisiaan data pada table dan dapat mengisi penjelasan pada setiap sub chapter pada laporan tersebut. Setelah laporan telah diisi, maka data laporan yang ada di easyDRKPL akan dibuat dalam bentuk file excel dan word secara otomatis.

Perancangan sistem digambarkan pada beberapa diagram seperti flowchart, entity relationship dan mockup. Entity relationship merupakan diagram untuk menggambarkan struktur dan hubungan antar data yang digunakan pada sistem. Flowchart bertujuan untuk menjelaskan alur program bekerja dan mockup memiliki fungsi untuk menggambarkan perencanaan tampilan yang nantinya akan diimplementasi ke menjadi program.

Dapat dilihat pada diagram tersebut ada entitas dengan nama DRKPL Sub Chapter dapat memiliki banyak DRKPL Sub Chapter Table item. Pada satu DRKPL Sub Chapter akan memiliki banyak sekali *item-item* pada table yang berhubungan dengan sub chapter tersebut.

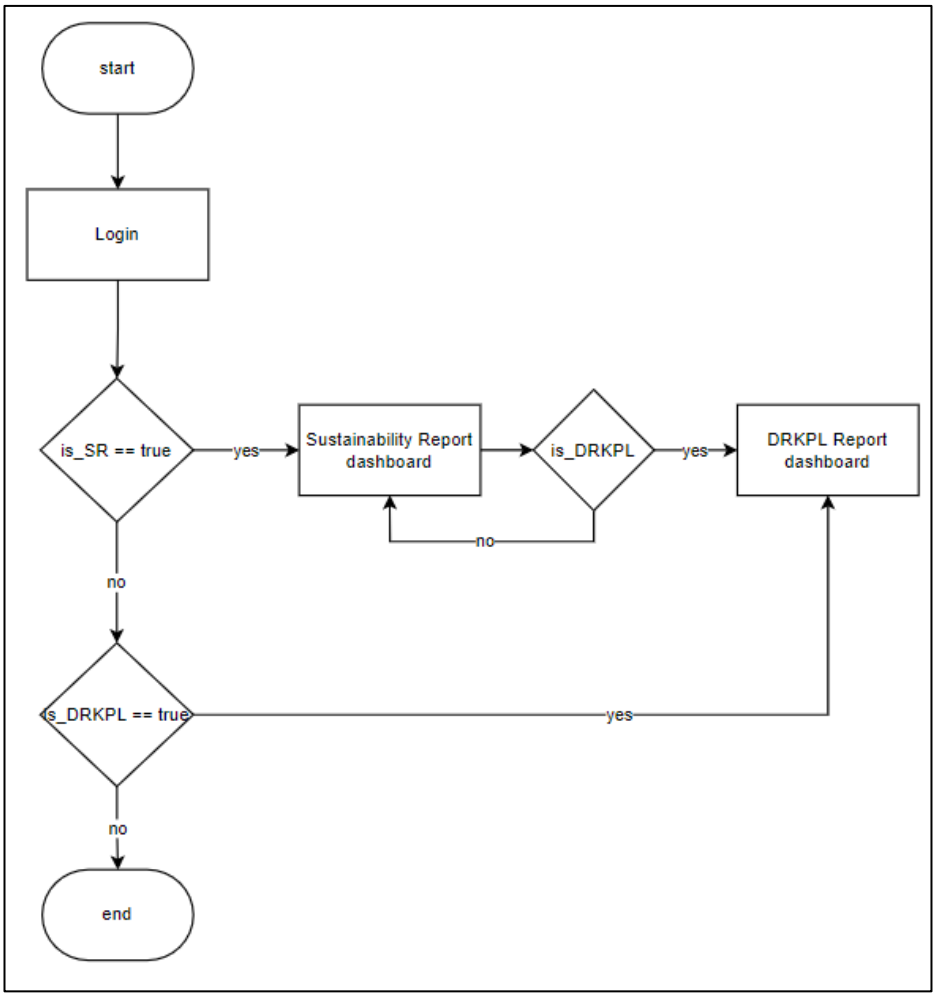
B.1 Flowchart

Pada bagian ini, flowchart akan dibagi dalam beberapa gambar yang dipecah berdasarkan fungsi. Flowchart yang ada merupakan fitur pada program yang secara aktif dikembangkan. Secara garis besar, fitur-fitur pada program yang dibuat adalah *access privilege* untuk DRKPL dan SR (Sustainibility Report), membuat laporan DRKPL, mengisi laporan DRKPL beserta data entry dan modul perhitungan formula dari rumus perhitungan pada data entry. Pada gambar 3.1 merupakan flowchart utama dari berlangsungnya proses pengunaan sistem mulai dari awal hingga akhir.



Gambar 3.1 Flowchart Utama

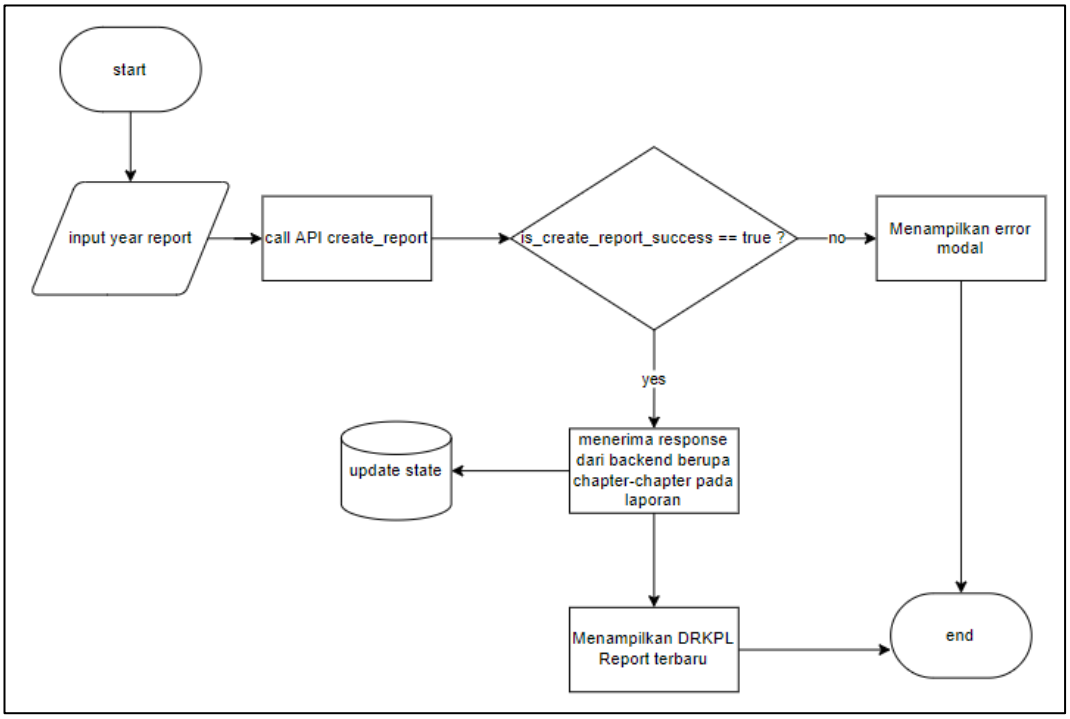
Gambar 3.1 menampilkan proses-proses pada sistem yang diawali dengan melakukan login. Setelah login sukses maka akan dilanjutkan ke pembuatan laporan agar *user* dapat mengisi data pada halaman data entry dan halaman sub chapter. Setelah seluruh data telah diisi, hasil akhir dari pengisiaan data tersebut berupa file Docx atau Excel yang pada kesempatan magang ini belum diimplementasikan dan masih tahap pengembangan. Gambar 3.2 menunjukan flowchart izin akses pada dashboard DRKPL ataupun SR.



Gambar 3.2 Flowchart *Access Privilege*

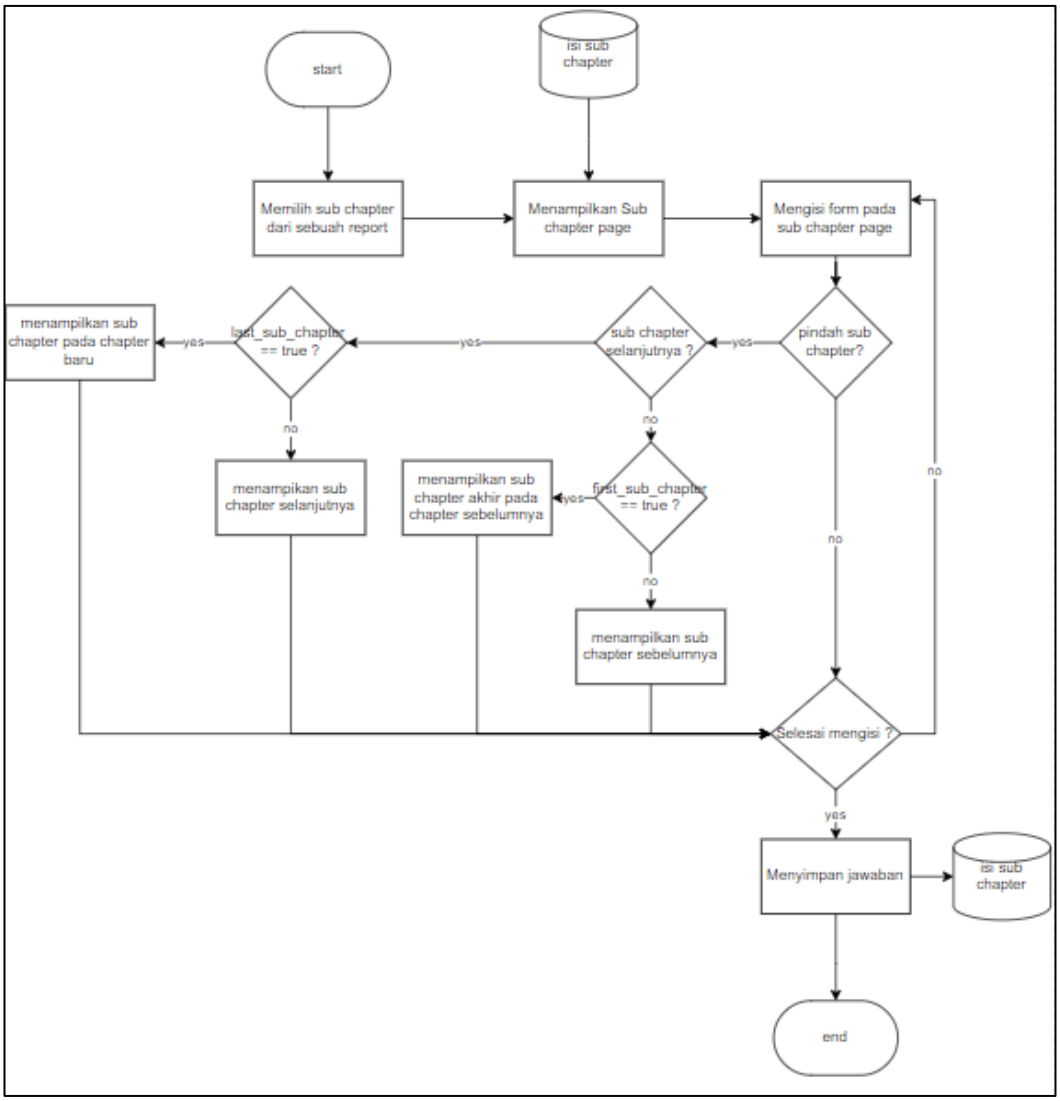
Setelah user melakukan login, aplikasi akan melakukan pemeriksaan apakah user tersebut memiliki hak akses ke laporan terentu. Jika dia memiliki akses SR dan DRKPL dengan variable is_SR dan is_DRKPL maka dia bisa mengakses dashboard SR dan DRKPL, sedangkan kalau dia hanya bisa mengakses salah satu laporan, maka aplikasi akan langsung mengarahkan ke dashboard yang berisi laporan-laporan tersebut. Setelah user dapat mengakses dashboard pada DRKPL. Dashboard akan menampilkan beberapa laporan yang pernah dibuat dan jika belum ada laporan yang dibuat, maka tampilan akan menunjukan bahwa tidak tersedia

laporan yang dapat diisi. Selanjutnya, fitur untuk pembuatan laporan DRKPL yang digambarkan alurnya pada gambar 3.3



Gambar 3.3 Pembuatan Laporan DRKPL

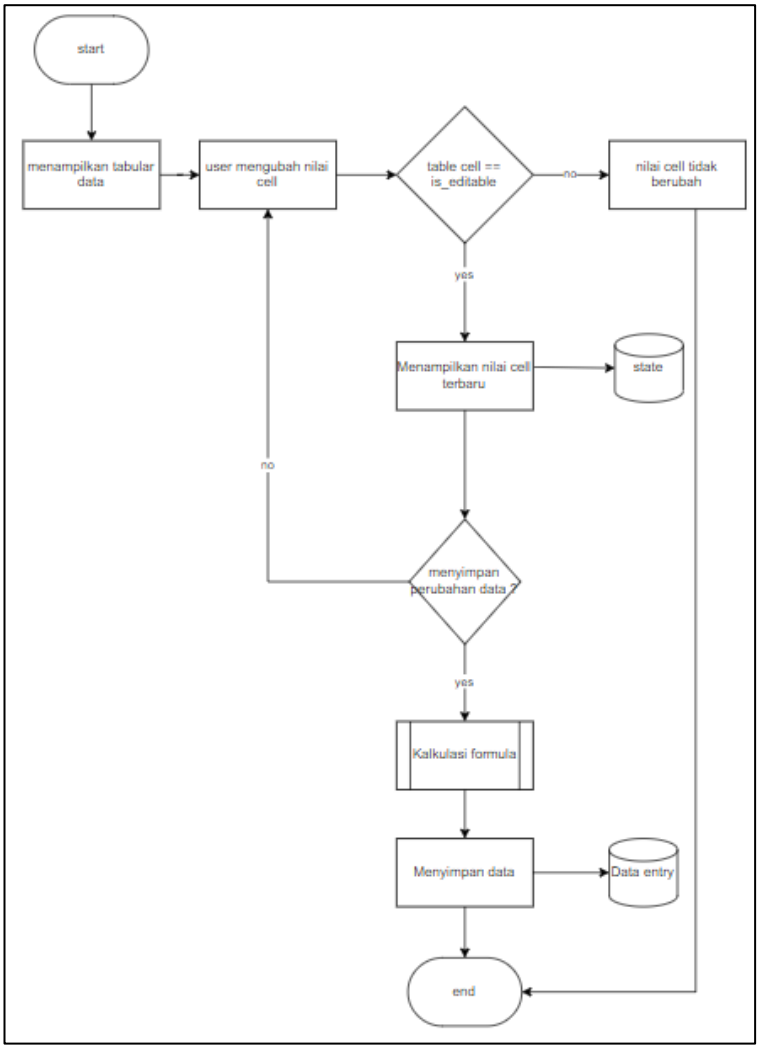
User akan memasukan nilai tahun yang mau dibuat laporannya. Setelah itu aplikasi akan memproses tahun dan nama perusahaan dari user melalui API. Respon API yang sukses akan menambah data pada database berupa nama laporan beserta isi laporan seperti chapter-chapter pada laporan yang harus diisi, lalu *state* pada aplikasi juga berubah dengan adanya penambahan laporan tersebut. Jika pembuatan laporan error maka akan menampilkan pesan bahwa gagal membuat laporan dan diminta untuk menghubungi admin. Setelah laporan berhasil dibuat, maka user dapat memilih salah satu chapter yang mau user isi kebutuhan data. User mengisi data tersebut pada halaman sub chapter. Alur kerja pengisian beserta navigasi antar sub chapter pada laporan terdapat pada gambar 3.4.



Gambar 3.4 Halaman Sub Chapter

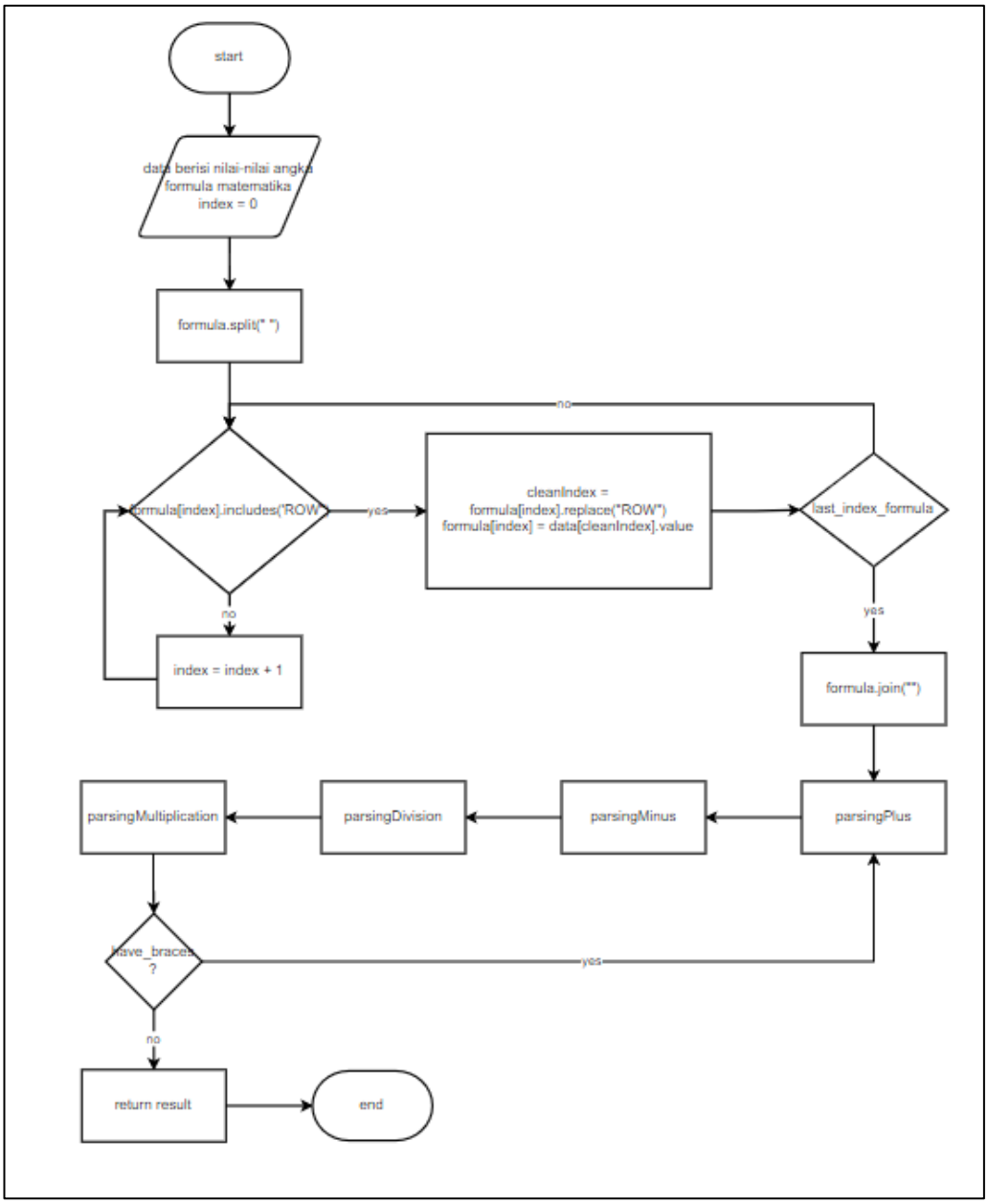
Setelah memilih sub chapter, user dapat mengisi data pada form yang sudah ditampilkan pada halaman tersebut. Jika user ingin berpindah ke sub chapter lainnya, dapat memilih antara sub chapter selanjutnya atau sebelumnya. Jika user ingin mengisi sub chapter selanjutnya, maka aplikasi akan mengecek apakah ada sub chapter, jika tidak ada sub chapter maka aplikasi akan mengarahkan ke chapter selanjutnya dengan sub chapter pertama pada chapter tersebut. Sama cara kerjanya seperti berpindah ke sub chapter sebelumnya, aplikasi akan mengecek ketersediaan

sub chapter pada chapter tersebut dan jika tidak memiliki sub chapter lagi, maka akan berpindah ke chapter sebelumnya dengan sub chapter paling akhir pada chapter tersebut. Kebutuhan untuk mengisi laporan bukan hanya dalam bentuk paragraf melainkan pada laporan DRKPL perlu mencantumkan data berupa table penggunaan energi dan dampak pada lingkungan. Agar dapat melakukan hal tersebut pada aplikasi ini, terdapat halaman data entry yang berupa table untuk mengisi nilai-nilai sesuai aktivitas yang ada. Alur kerja halaman tersebut terdapat pada gambar 3.5.



Gambar 3.5 Halaman Data Entry

User akan melihat table yang berisi aktivitas dan tahun-tahun yang tersedia. User dapat mengubah nilai sel pada setiap table yang merupakan sel yang dapat diedit. Sel yang dapat diedit adalah sel yang bukan hasil penjumlahan dari sel-sel lainnya atau disebut sebagai nilai utama yang berguna sebagai data untuk mengkalkulasi total dari aktivitas-aktivitas. Setelah mengubah nilai pada setiap sel, user dapat menyimpan hasil perubahan dengan menggunakan fitur save. Proses perhitungan rumus pada data entry dapat dilihat pada gambar 3.6.

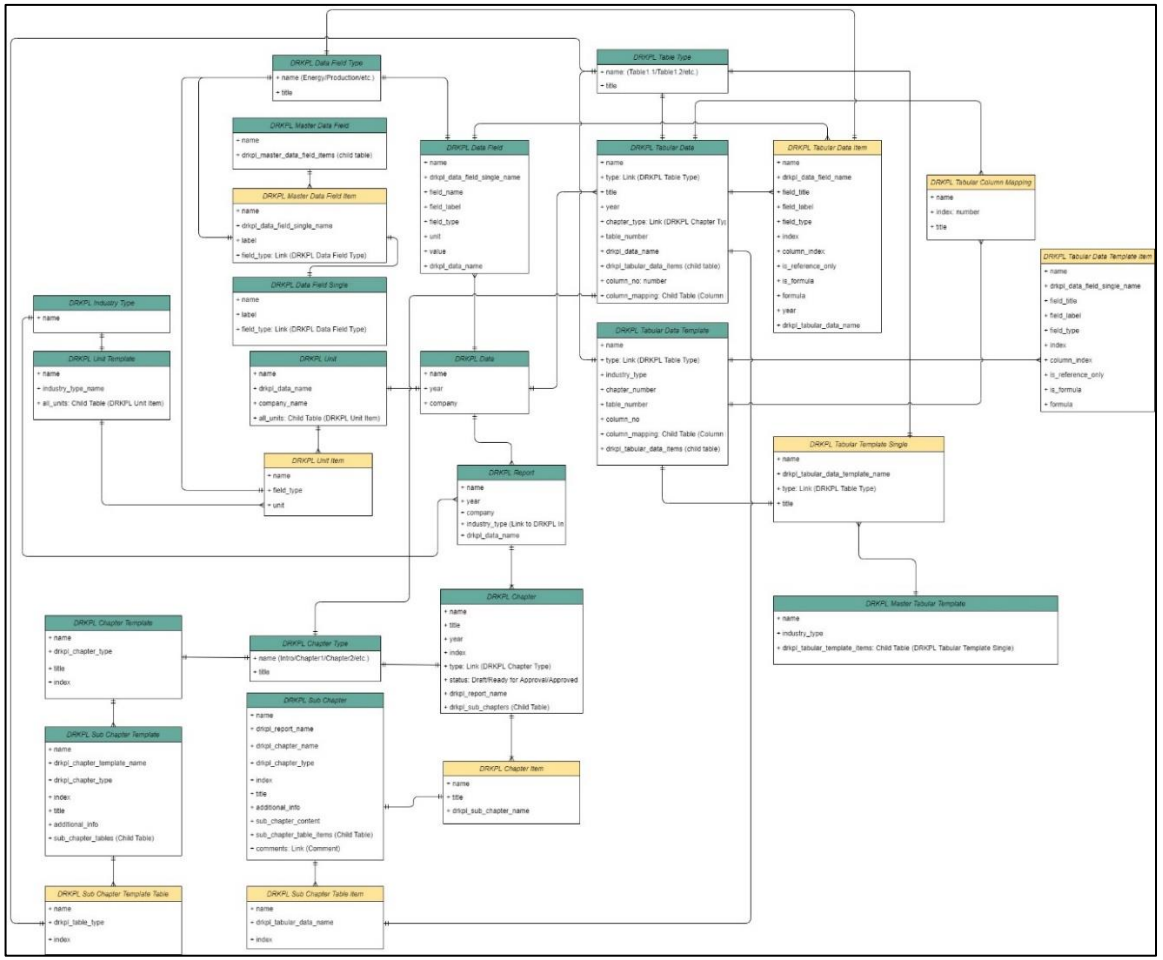


Gambar 3.6 Kalkulasi Rumus Data Entry

Pada modul ini diperlukana formula seperti “ROW0 + ROW1” yang dimana formula ini akan diparsing menjadi nilai pada kolom tabel tersebut. Dengan menggunakan proses parsing yang dimulai dari penjumlahan hingga perkalian berdasarkan tingkat prioritas perhitungan matematis. Hasil akhir dari modul ini adalah perhitungan sesuai dengan formula yang diberikan.

B.2 Diagram Entity Relationship

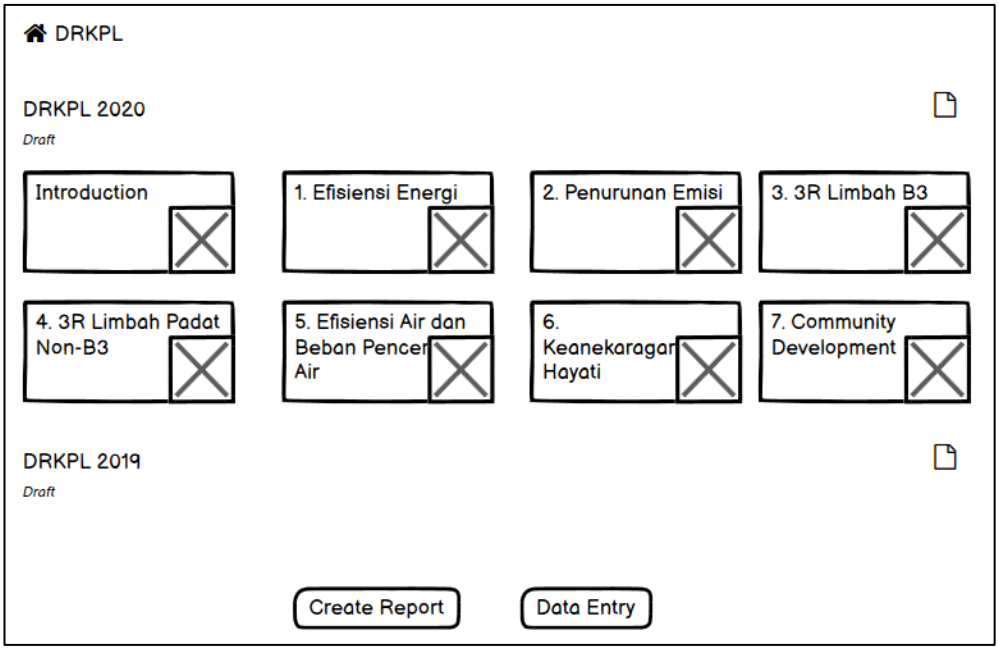
Diagram entity relationship merupakan diagram yang menjadi landasan dalam pembuatan aplikasi. Isi dari diagram ini merupakan struktur database dan atribut pada setiap entitas yang digunakan oleh backend dan data-data tersebut akan dikirim ke frontend. Diagram akan selalu diupdate jika ada perubahan karena kurangnya atribut pada entitas ataupun ada perubahan kebutuhan dengan menggunakan draw.io. Berikut adalah diagram dari aplikasi easyDRKPL pada gambar 3.7.



Gambar 3.7 Diagram Entity Relationship easyDRKPL

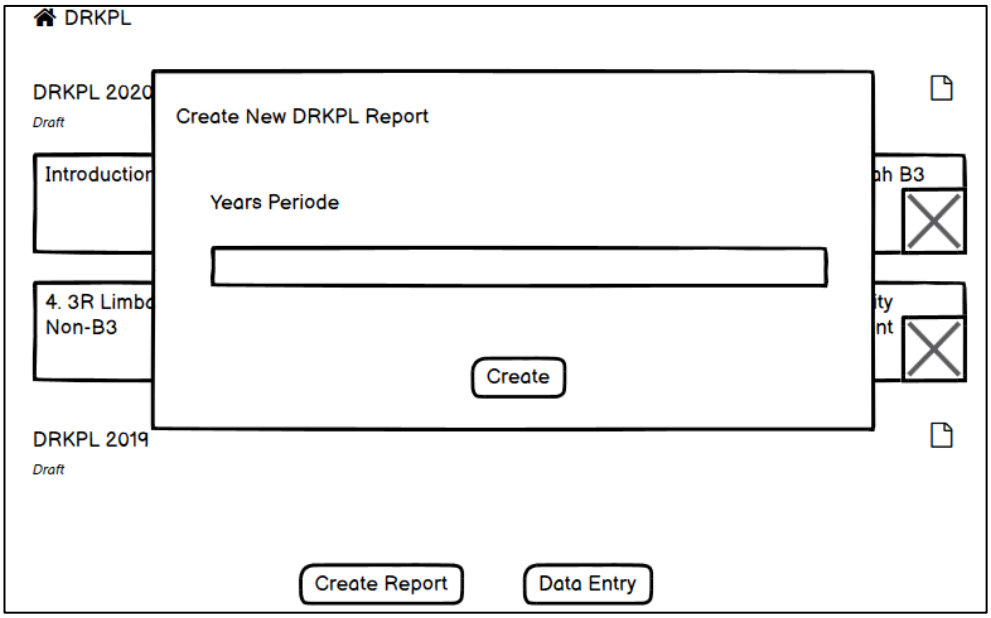
B.3 Mockup

Mockup dibuat oleh salah satu tim yang memegang posisi sebagai UI/UX designer dengan menggunakan adobe XD. Mockup meliputi tampilan dari aplikasi web hingga *flow* berjalannya aplikasi tersebut. Mockup yang ditampilkan dimulai dari *dashboard* yang berisi laporan-laporan DRKPL. Pada halaman *dashboard* ini, *user* dapat melihat banyaknya laporan secara garis besar dengan menampilkan *progress* pengisian data dari tiap *chapter* pada laporan.



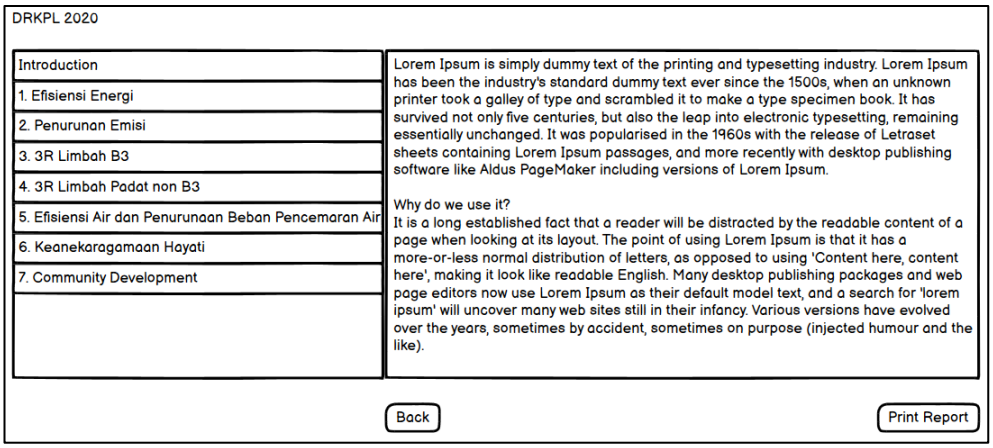
Gambar 3.8 Mockup Dashboard DRKPL

Pada halaman ini, *user* juga dapat membuat laporan DRKPL dengan memasukan tahun laporan yang ingin dibuat. Agar dapat mengakses fitur tersebut terdapat tombol pada bagian bawah halaman *dashboard*.



Gambar 3.9 Modal Membuat Laporan

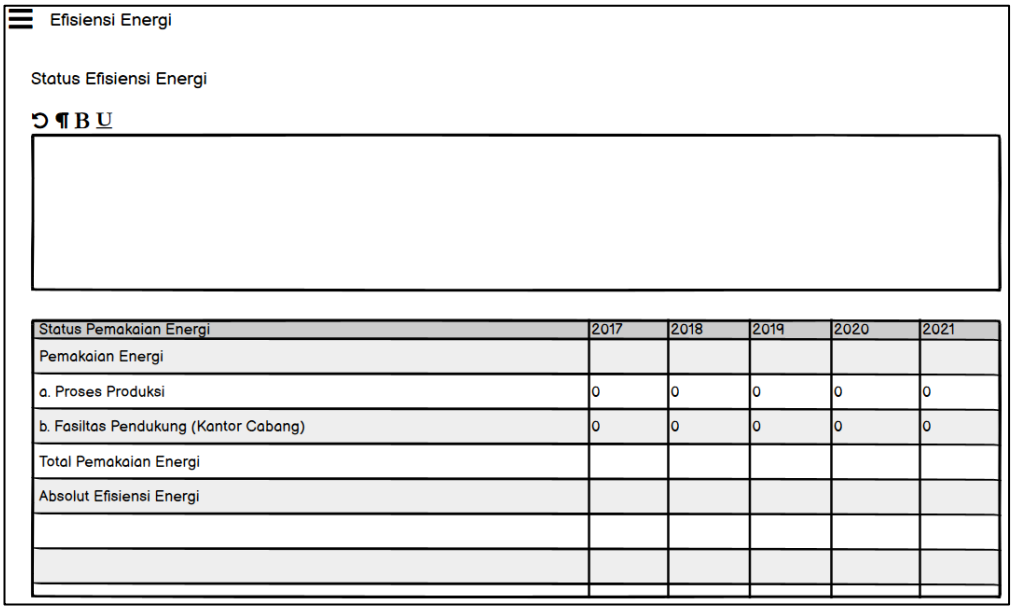
Jika ingin melakukan *preview* untuk isi dari laporan. Terdapat *icon* pada setiap judul laporan yang berada pada paling kanan dari laporan tersebut. Dengan menekan *icon* tersebut, *user* dapat melihat isi dari seluruh jawaban pada laporan yang sudah berhasil disimpan.



Gambar 3.10 Preview Laporan DRKPL

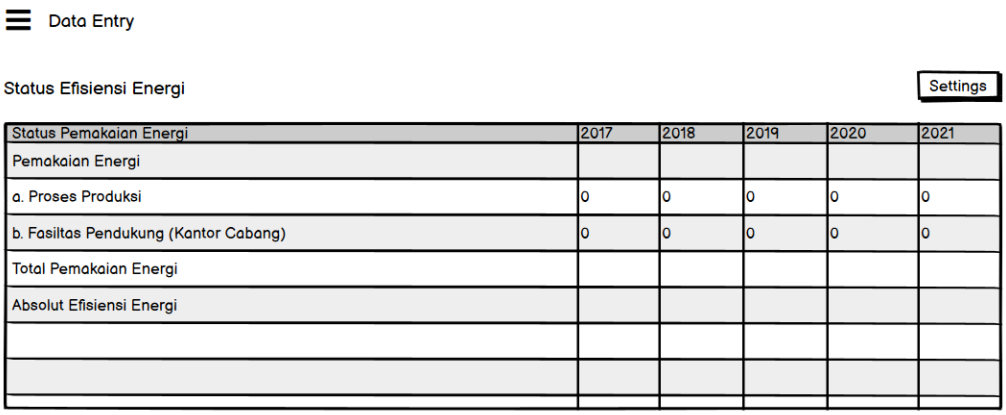
User dapat mengakses tiap *chapter* pada laporan agar dapat mengisi setiap bagian pada laporan. Halaman yang menampilkan isi dari setiap laporan secara

detail adalah halaman *sub chapter* yang dimana setiap *chapter* berisi beberapa *sub chapter*. Halaman tersebut berisi *textarea* dan data table yang berasal dari data entry. *Textarea* akan diisi oleh *user* dengan menjawab pertanyaan dari setiap *sub chapter* berupa paragraf ataupun poin-poin.



Gambar 3.11 Halaman Sub Chapter

Halaman yang memiliki fungsi untuk mengisi nilai pada table adalah halaman data entry. Pada halaman ini, *user* dapat mengisi data-data berupa numerikal pada setiap table pada *sub chapter* pada laporan. Tabel berisi aktifitas-aktifitas dan lima tahun sebelumnya mengenai penggunaan energi ataupun pengeluaran dana dalam proses produksi pada suatu perusahaan.



Gambar 3.12 Halaman Data Entry

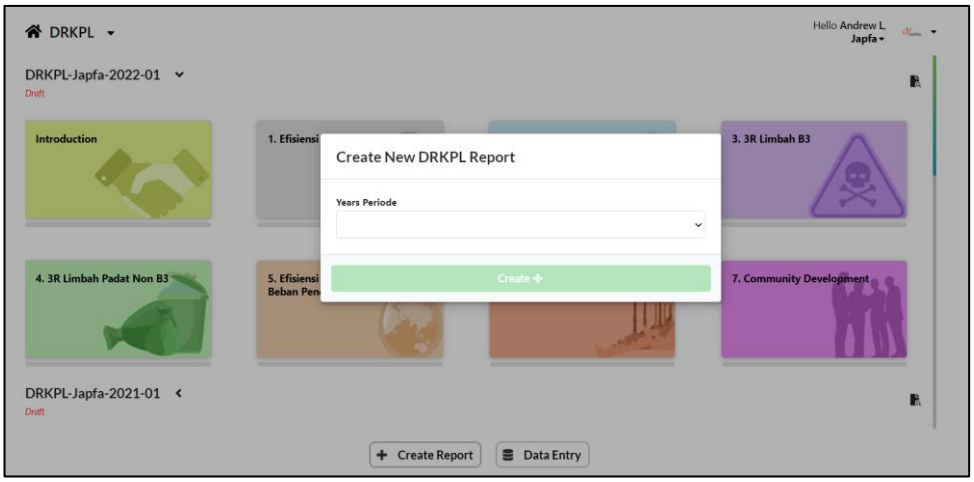
C. Implementasi

Pada tahap ini, mockup dan diagram yang telah dirancang sekaligus disetujui oleh *client* dan perusahaan akan diimplementasi menjadi aplikasi web sesungguhnya. Proses implementasi tidak sepenuhnya mirip dengan mockup yang dibuat, karena terjadi berbagai perubahan saat proses *development* yang disesuaikan dari hasil diskusi antar kedua belah pihak dan juga para *developer*. Sehingga gambar yang disajikan pada bagian ini merupakan hasil implementasi yang merupakan pengembangan dari evaluasi-evaluasi dari desain yang ada dimockup.

Untuk bagian pada halaman *dashboard* DRKPL, *user* yang telah melakukan login dan memiliki akses pada halaman ini dapat melihat seluruh laporan DRKPL pada perusahaanya yang telah sukses dibuat. *User* dapat melakukan fungsi seperti membuat laporan baru, mengakses setiap *chapter* dan melakukan *preview* pada laporan tersebut. Selain itu, *user* juga dapat berpindah ke *dashboard* laporan SR dengan menekan *dropdown* pada kiri atas yang bertuliskan DRKPL. Selain itu, halaman *dashboard* juga mengalami perubahaan dari sisi tampilan yaitu gaya tulisan dan tombol bagian bawah yang lebih tebal dan *dropdown* yang dibuat lebih padat dan ringkas.

Gambar 3.13 Implementasi Dashboard DRKPL

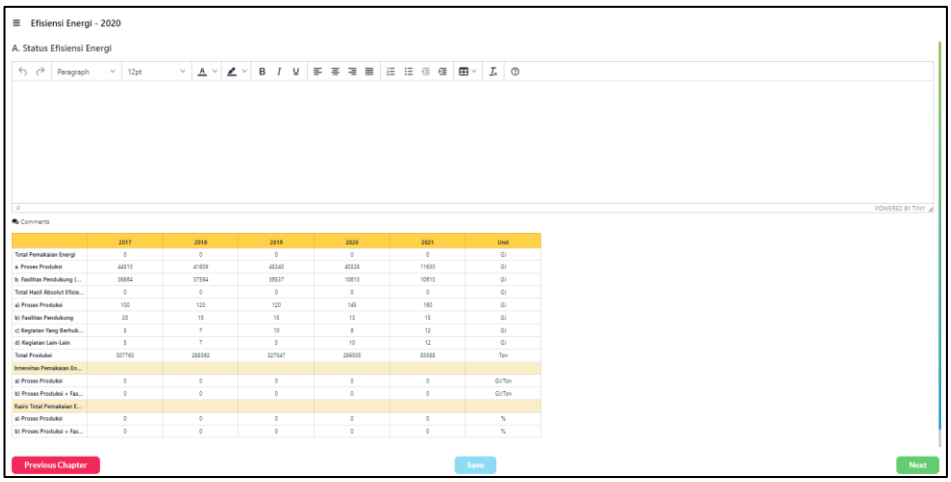
Implementasi pada fitur pembuatan laporan dengan menekan tombol “create report” juga mengalami perubahaan pada judul modal dan juga warna dari komponen tersebut. Selain itu, jika tombol tersebut ditekan, maka halaman *dashboard* akan meredup agar mendapatkan fokus pada komponen modal. *User* akan memasukkan tahun tertentu yang ingin dibuat laporan DRKPL. Setelah itu proses akan dilanjutkan dengan menekan tombol “create” yang memanggil API dengan *parameter* tahun beserta nama perusahaan. API yang sukses diproses akan mengembalikan status oke dan laporan terbaru akan disimpan pada *store* dan dilakukan proses *sorting* sesuai judul laporan.



Gambar 3.14 Modal Membuat Laporan DRKPL

Setelah pembuatan laporan berhasil, *user* dapat mengakses setiap *chapter* pada laporan dan halaman yang akan ditampilkan merupakan halaman *sub chapter*. Halaman ini berisi berbagai komponen seperti *sidebar* yang berisi *chapter* beserta *sub chapter*, *textarea* yang menggunakan tinyMCE dan tabel dari data entry. *User* akan mengisi *textarea* tersebut untuk menjawab setiap *sub chapter* yang terpilih. Jawaban tersebut akan disimpan pada *state* berupa data *string* yang memiliki format seperti HTML. *User* yang telah mengisi jawaban dapat menyimpan jawaban tersebut dengan menekan tombol save yang berada dibawah. Proses sebelum menyimpan adalah dengan mengecek apakah *state* jawaban menampung jawaban tersebut dan jika menampung maka *state* tersebut akan dikirimkan ke API menyimpan data dan setelah proses menyimpan data selesai, maka *state* tersebut akan dibersihkan. Selain itu tombol save hanya dapat digunakan jika ada perubahan nilai pada *state* jawaban pada *sub chapter*. Jika telah melakukan proses menyimpan data, maka *state* yang mengatur tombol save dapat ditekan akan dinonaktifkan. *User* dapat berpindah dari satu *sub chapter* ke *sub chapter* lainnya dengan menggunakan tombol navigasi *burger button* yang berada di atas kiri atau

menggunakan tombol yang berada dibawah yaitu previous dan next. Setiap *sub chapter* terakhir dan diawal, tombol akan berubah menjadi previous chapter atau next chapter. Cara kerja dari tombol navigasi next adalah dengan melakukan pengecekan pada index *sub chapter* selanjutnya, jika pada index selanjutnya tidak terdapat data maka tombol akan berubah menjadi next chapter. Tombol next chapter yang ditekan akan mengarahkan halaman ke *chapter* selanjutnya dengan index paling awal pada *chapter* tersebut. Sama seperti tombol previous, namun yang berbeda adalah saat perpindahan *chapter* maka *sub chapter* yang terpilih merupakan *chapter* paling akhir dari *chapter* tersebut.

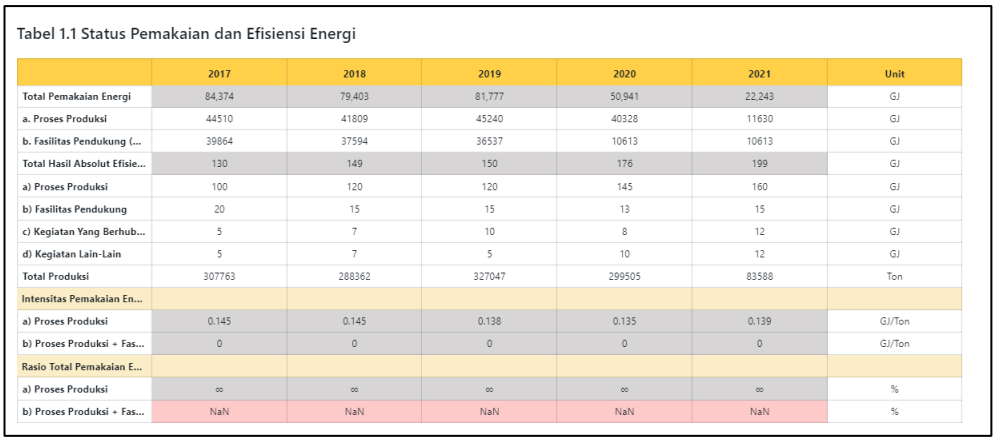


Gambar 3.15 Halaman Sub Chapter

Halaman data entry merupakan halaman yang berfungsi untuk mengisi data tabel dari setiap tahun pada perusahaan tersebut. Data tersebut berguna untuk membantu membuat laporan DRKPL yang berada di *sub chapter*. Pada halaman ini, komponen tabel menggunakan komponen table pada Blueprintjs dengan bantuan komponen segment pada SemanticUI untuk beberapa kasus tabel yang memiliki bentuk yang khusus. Pada halaman ini, setiap sel pada tabel terbagi menjadi beberapa jenis yaitu informational, reference, formula dan editable.

Informational merupakan sel yang hanya menampilkan informasi dari setiap aktifitas-aktifitas yang bersangkutan. Informational sel memiliki fungsi menjadi pemisah antar aktifitas-aktifitas yang berhubungan. Sel tersebut dapat dilihat dari warna kuning pada sel tersebut. Sel reference yaitu sel yang tidak dapat diganti nilainya karena nilai pada sel tersebut didapat dari tabel lain yang bukan berada di tabel tersebut dan diharuskan mengedit nilai dari table asal. Sel formula merupakan sel yang nilainya didapat dari perhitungan matematis dari baris pada kolom tersebut. Warna latar pada sel reference dan formula adalah abu-abu agar *user* dapat mudah mengetahui bahwa sel tersebut tidak dapat diganti nilainya. Formula perhitungan menggunakan fungsi formulaCalculation yang dibuat dengan konsep melakukan parsing dari ekspresi matematika menjadi hasil perhitungan. Contoh dari ekspresi matematika adalah "ROW0 + ROW1". Row pada eksrepsi tersebut menunjukan baris dan nilai setelah kata ROW merupakan index yang terdapat pada kolom tersebut. Proses diawali dengan melakukan pemisahan antara operator matematika dan ROW. Selanjutnya, dilakukan pengecekan pada setiap elemen pada array yang berisi hasil pemisahaan ekspresi tersebut. Jika pada elemen tersebut mengandung kata ROW, maka dilanjutkan dengan proses penghapusan kata ROW dan nilai index pada ROW tersebut akan digantikan dengan nilai pada data table, misalnya pada data table kolom 1 baris 0 memiliki nilai 50, maka nilai ROW0 akan digantikan menjadi 50 dan proses dilanjutkan hingga ekspresi matematika tersebut menjadi kumpulan angka. Contoh Hasil akhir dari proses tersebut adalah "50 + 75", selanjutnya ekspresi tersebut akan dihitung menggunakan empat fungsi utama yang diproses secara berurutan mulai dari parsing penjumlahan, pengurangan, pembagian dan perkalian. Urutan pada proses ini diliat dari tingkat prioritas

perhitungan matematika, seperti perkalian dan pembagian harus didahulukan baru menghitung penjumlahan dan pembagian. Proses-proses ini dilakukan secara rekursif hingga seluruh perhitungan berhasil. Perubahan dari setiap sel ditampung pada *state* dan juga mengubah nilai pada *store* aplikasi web. Setelah seluruh proses pengisian data dilakukan, maka *user* dapat melakukan penyimpanaan dengan menekan tombol save jika ada perubahan pada sel karena tombol save hanya akan aktif jika *state* memiliki nilai *true* pada perubahaan. Seluruh sel yang berubah akan ditampung pada suatu *state* yang berisi nama unik dari sel dan perubahan nilai pada sel tersebut. Selanjutnya proses penyimpanaan akan melakukan pengecekan pada *state* yang menampung jawaban dan dikirimkan melalui API. Jika berhasil menyimpan, maka data pada backend akan berubah.



Tabel 1.1 Status Pemakaian dan Efisiensi Energi

	2017	2018	2019	2020	2021	Unit
Total Pemakaian Energi	84,374	79,403	81,777	50,941	22,243	GJ
a. Proses Produksi	44510	41809	45240	40328	11630	GJ
b. Fasilitas Pendukung (...	39864	37594	36537	10613	10613	GJ
Total Hasil Absolut Efisie...	130	149	150	176	199	GJ
a) Proses Produksi	100	120	120	145	160	GJ
b) Fasilitas Pendukung	20	15	15	13	15	GJ
c) Kegiatan Yang Berhub...	5	7	10	8	12	GJ
d) Kegiatan Lain-Lain	5	7	5	10	12	GJ
Total Produksi	307763	288362	327047	299505	83588	Ton
Intensitas Pemakaian En...						
a) Proses Produksi	0.145	0.145	0.138	0.135	0.139	GJ/Ton
b) Proses Produksi + Fas...	0	0	0	0	0	GJ/Ton
Rasio Total Pemakaian E...						
a) Proses Produksi	∞	∞	∞	∞	∞	%
b) Proses Produksi + Fas...	NaN	NaN	NaN	NaN	NaN	%

Gambar 3.16 Halaman Data Entry

3.3.2 Kendala yang Ditemukan

Dalam melakukan pelaksanaan kerja magang, terdapat berbagai kendala yang ditemukan yaitu:

1. Belum mempunyai pengalaman menggunakan React Typescript.
2. Struktur *project* yang sudah ada cukup kompleks.
3. Mengalami berbagai error pada *staging* yang sebelumnya tidak pernah ditemukan pada proses *development* di lokal.
4. Pembuatan modul perhitungan matematika dari data *string* menjadi data numerikal yang dihitung secara otomatis.

3.3.3 Solusi Atas Kendala yang Ditemukan

Kendala yang dihadapi saat pelaksanaan kerja magang harus mempunyai solusi agar pada proses pengembangan aplikasi web dapat berjalan dengan baik dengan hasil yang maksimal, berikut adalah solusi dari kendala yang dihadapi.

1. Melakukan pelatihan bersama *supervisor* dengan membuat *roadmap* pembelajaran React Typescript. Selain itu juga aktif dalam melakukan diskusi pada kebingunggan atau kendala pada React.
2. Mempelajari struktur *project* dengan meneliti tiap *file* yang ada dan bertanya untuk memastikan konsep dengan *supervisor* agar tidak terjadi perbedaan pandangan.
3. Lebih teliti dalam membuat sebuah modul dan melakukan *testing* di beberapa perangkat dan meminta tim lainnya untuk melakukan *testing* agar tidak terjadi *error* pada *staging*.

4. Mencari referensi untuk pembuatan modul matematika dari berbagai sumber dan melakukan modifikasi terhadap referensi tersebut. Selanjutnya melakukan diskusi bersama *supervisor* dalam membentuk modul perhitungan matematika.