

BAB 3

PELAKSANAAN KERJA MAGANG

3.1 Kedudukan dan Organisasi

Pelaksanaan kerja magang di PT. Global Vita Nutritech dilaksanakan pada 14 Juni 2021 hingga 13 Desember 2021 dengan posisi sebagai *full stack engineer* pada divisi IT yang di bimbing langsung oleh bapak Ade Aminudin Nazib selaku Supervisi IT di PT. Global Vita Nutritech. Proses kerja magang di lakukan secara *hybrid*, *work from home* dan juga *work from office* yang dilaksanakan di *Innovation and Development Center* PT sanghiang perkasa yang terletak di Cakung Timur, Jakarta Timur.

Dalam kerja magang penulis bekerjasama langsung dengan Bapak Ade Aminudin Nazib dan juga departemen *PPIC and Procurement*. Kodirnasi antara departemen PPIC and Procurement dan juga IT dilakukan dengan cara meeting di kantor dan juga meeting secara daring melalui *zoom* untuk membahas *requirement* dan juga testing aplikasi, sedangkan komunikasi antara penulis dan pembimbing dilakukan secara langsung di kantor dan juga secara daring melalui *whatsapp* untuk mediskusikan hal yang harus dikerjakan dan juga mengenai kendala yang dihadapi.

3.2 Tugas yang Dilakukan

Selama Proses Kerja magang, tugas dan juga tanggung jawab yang dilakukan adalah sebagai berikut:

1. Mempelajari dan mengimplementasikan Standard IT yang ada di Kalbe Nutritional
2. Mempelajari sistem MRP yang sudah ada dan mengimplementasikannya kedalam aplikasi Web

3. Membuat *flow diagram* dan juga *mockup design* aplikasi web MRP
4. Membuat kodingan *user interface* halaman *dashboard*, *document view*, *approval list*, dan juga *material usage* untuk aplikasi web MRP
5. Membuat *buzzlogic* untuk mengambil *data formula*, *data on hand*, *data recipe*, dan *data master item* dari oracle, mengambil data *document* dari database berdasarkan status, *update document status*, dan *update purchase requisition data*.
6. Membuat *Javascript* untuk *datatable*, *update material usage*, *pop up modal*, dan *reject function*
7. Membuat Dokumentasi Project berdasarkan standard Kalbe Nutritionals
8. Melakukan testing bersama *user* dan melakukan *bug fixing*

3.3 Uraian Pelaksanaan Magang

Pelaksanaan kerja magang diuraikan seperti pada Tabel 3.1.



Tabel 3.1. Pekerjaan yang dilakukan tiap minggu selama pelaksanaan kerja magang

Minggu Ke -	Pekerjaan yang dilakukan
1	<ul style="list-style-type: none"> • Pengenalan terhadap Kalbe Group dan program magang yang ada • Melakukan standarisasi sesuai standard yang ada di Kalbe Nutritionals bersama dengan mentor
2	<ul style="list-style-type: none"> • Mengerjakan tugas yang diberikan oleh mentor untuk mempelajari ASP.NET • Melakukan review tugas bersama mentor
3	<ul style="list-style-type: none"> • Mengimplementasikan standard yang ada kedalam sebuah mini project yang sudah disediakan • Melakukan review tugas bersama mentor
4	<ul style="list-style-type: none"> • Menghadiri meeting pertama mengenai <i>project requirements</i> • Membuat <i>flowchart</i> untuk <i>project MRP</i> • Mempelajari Datatable
5	<ul style="list-style-type: none"> • Melanjutkan pembelajaran mengenai Datatable • Mempelajari cara bagaimana mengubah nilai data dari datatable ke database
6	<ul style="list-style-type: none"> • Membuat <i>mockup design</i> untuk proyek MRP

7	<ul style="list-style-type: none"> • Mempelajari <i>business process</i> dari MRP • Membaca dan mempelajari kodingan MRP Marco Excel
8	<ul style="list-style-type: none"> • Mengerjakan kodingan UI untuk MRP • Melakukan meeting bersama mentor dan juga user
9	<ul style="list-style-type: none"> • Membuat kodingan untuk membaca data dari excel kedalam database • Merancang table baru untuk database • Melakukan meeting mengenai perkembangan proyek
10	<ul style="list-style-type: none"> • Membuat kodingan untuk mengubah data di database langsung dari view yang ada menggunakan javascript,controller,buzzlogic • Membuat tampilan Document View,approval list dan juga Dashboard serta pengambilan data dari database untuk tampilan tersebut
11	<ul style="list-style-type: none"> • Membuat Buzzlogic untuk memanggil data di oracle • Memperbaiki UI agar lebih menarik

12	<ul style="list-style-type: none"> • Meeting bersama <i>user</i>, <i>IT product</i>, dan juga mentor mengenai MRP dan juga <i>business process</i> MRP • Membuat fungsi Approval
13	<ul style="list-style-type: none"> • Membuat Buzzlogic untuk mengubah status pada document • Membuat fungsi reject
14	<ul style="list-style-type: none"> • Membuat fungsi Logstatus pada sebuah document • Menambahkan Icon-icon kedalam UI • Memperbaiki UI agar lebih rapih
15	<ul style="list-style-type: none"> • Menambahkan fungsi Encrypt dan Decrypt • Memperbaiki UI agar lebih rapih
16	<ul style="list-style-type: none"> • Mempelajari kalkulasi untuk simulasi
17	<ul style="list-style-type: none"> • Meeting bersama user untuk membahas proyek • Menambahkan <i>fixed column</i>, <i>column color</i>, dan juga <i>export Excel</i>

18	<ul style="list-style-type: none"> • Membuat <i>template code</i> untuk <i>future project user</i> • Melakukan testing MRP bersama user
19	<ul style="list-style-type: none"> • Melakukan debugging untuk mencari error pada perhitungan
20	<ul style="list-style-type: none"> • Membuat <i>UI Client</i> untuk <i>future project user</i>
21	<ul style="list-style-type: none"> • Memahami sistem dokumentasi proyek pada Kalbe Nutritionals • Membuat dokumentasi Proyek MRP
22	<ul style="list-style-type: none"> • Membuat dokumentasi Proyek MRP
23	<ul style="list-style-type: none"> • Membuat dokumentasi Proyek MRP
24	<ul style="list-style-type: none"> • Melakukan UIT • bug Fixing

25	<ul style="list-style-type: none"> • Mempersiapkan presentasi KNIP • Melakukan UIT • Bug Fixing
26	<ul style="list-style-type: none"> • Presentasi KNIP • Meeting Rakor 2021

3.3.1 Proses Pelaksanaan

Proses pelaksanaan rancang bangun aplikasi web MRP membutuhkan bantuan perangkat lunak dan juga perangkat keras. Perangkat lunak yang digunakan dalam proses pelaksanaan rancang bangun aplikasi web MRP adalah sebagai berikut:

- Windows 10
- Visual Studio Code 2019
- Visual Studio Code 2017
- Microsoft SQL Server Management Studio 18
- Oracle SQL Developer
- Oracle Data Access Components
- Global Protect VPN
- Pulse Secure

Perangkat keras yang digunakan dalam proses pelaksanaan rancang bangun aplikasi web MRP adalah sebagai berikut:

- Intel Core I7-7700HQ @2.80 GHz
- RAM 16 GB
- NVIDIA GeForce GTX 1060 6 GB

Proses pelaksanaan rancang bangun aplikasi web MRP sendiri menggunakan ASP.NET sebagai *framework* dan menggunakan *gitlab* sebagai *repository*, hal tersebut merupakan standard perusahaan yang sudah ada. Pembuatan aplikasi web MRP sendiri terbagi menjadi 5 modul utama yaitu *Dashboard*, *MRP Simulation*, *Document View*, *Approval List* dan juga *Material Usage*. Pengerjaan proyek ini sendiri terbagi menjadi 4 fase yaitu:

1. Fase *requirements*
2. Fase *project design*
3. Fase *project development*
4. Fase *UIT/testing*

Berikut adalah *requirement*, *flowchart*, dan juga hasil implementasi dari proyek MRP.

A. Requirement

Requirement adalah dasar kebutuhan dari sebuah proyek. *Requirement* ini berisi hal apa saja yang dibutuhkan oleh suatu modul, hal apa yang akan dikerjakan, fungsi dari suatu modul, fitur apa saja yang harus dimasukkan kedalam suatu modul, dan juga kebutuhan lain-lainnya yang diperlukan agar aplikasi tersebut bisa berjalan sesuai yang diinginkan.

A.1 Dashboard

Modul ini merupakan halaman awal dari alur aplikasi MRP. Modul *dashboard* berfungsi untuk mengunggah *forecast RM* yang sudah ada dalam bentuk excel untuk dilakukannya perhitungan di modul berikutnya.

Pada modul ini pengguna harus mengunggah file sesuai dengan format yang ada, memasukan jumlah bulan yang akan di proses dan juga memasukan tanggal mulai perhitungan. Data yang sudah diunggah pada modul ini akan tersimpan kedalam suatu *viewmodel* yang sudah ada.

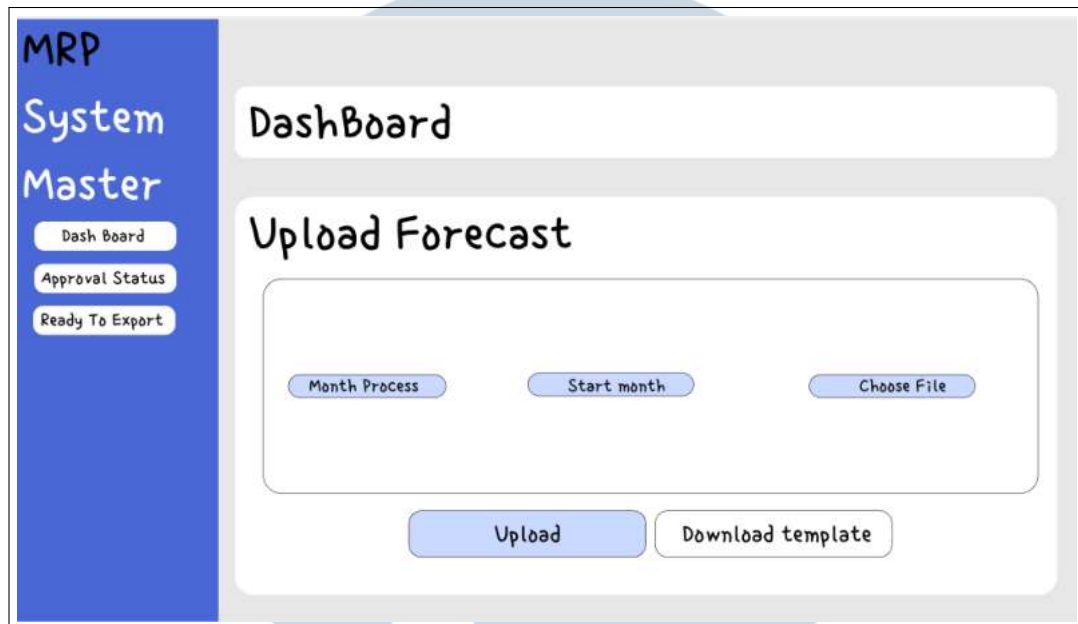
Modul ini dapat diakses oleh *user PPIC* dari menu *transaction* pada *navigation bar* yang sudah tersedia pada bagian kiri halaman. *Button* pada halaman ini memiliki fungsi dan kegunaan masing-masing yang tertera pada Tabel 3.2

Tabel 3.2. Dashboard Action Control

Button Name	Function	Action
Select File	Untuk memilih file pada folder laptop	Membuka folder pada Laptop
Upload	Untuk melakukan upload dan simulasi mrp	mengupload file dan melakukan MRP simulation
Download Template	Untuk mendownload template upload forecast RM	mendownload template forecast RM

UNIVERSITAS
MULTIMEDIA
NUSANTARA

Tampilan desain awal dari modul *dashboard* ini dapat dilihat pada Gambar 3.1



Gambar 3.1. Mockup Design Dashboard MRP

A.2 MRP Simulation

Modul ini merupakan modul kalkulasi yang tidak mempunyai tampilan. Pada modul ini data yang sudah diunggah pada modul sebelumnya akan diproses dengan mengkalkulasikan data yang ada dan data yang diambil dari *Oracle* secara *real time* dan akan menghasilkan data yang nantinya akan digunakan oleh *user* untuk membuat *purchase requisition*. Jika ada *error* pada proses kalkulasi, terdapat *log* yang menyediakan informasi terhadap *error* yang terjadi.

A.3 Document View

Modul *document view* merupakan modul untuk menampilkan semua dokumen MRP yang ada. Data yang sudah di kalkulasikan pada modul sebelumnya akan di kelompokkan dan di kumpulkan menjadi satu dokumen. Setiap dokumen akan memiliki status yang berbeda-beda. Sta-

tus *draft* dimana merupakan status bawaan dokumen setelah melewati proses kalkulasi, status *submit* adalah status dimana dokumen sudah di *submit* oleh PPIC dan menunggu *approval*, dan status *approved* dimana dokumen sudah disetujui oleh *role approval*.

Modul ini dapat diakses oleh *role* PPIC dan juga *role approval* dari menu *transaction* pada *navigation bar* yang tertera pada bagian kiri halaman. Role PPIC dapat melihat dokumen dengan semua status yang ada, tetapi untuk *role approval* hanya dapat melihat dokumen yang berstatus *submit* dan juga *approved*. *Button* pada halaman ini memiliki fungsi dan juga kegunaannya masing - masing yang tertera pada Tabel 3.3

Tabel 3.3. Document View Action Control

Button Name	Function	Action
Upload	Mendirect ke halaman upload dokumen	Membuka halaman dashboard
Details	Membuka Material Usage dari dokumen	Membuka material usage dari dokumen



Tampilan desain awal dari modul *document view* ini dapat dilihat pada Gambar 3.2



Gambar 3.2. Mockup Design Document View MRP

A.4 Approval List

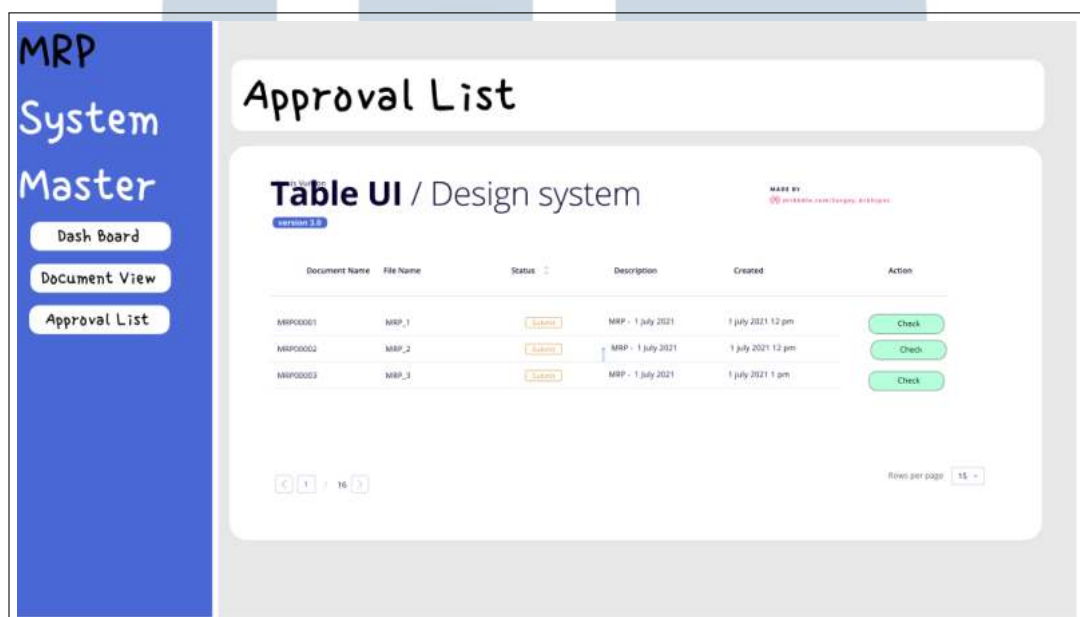
Modul *approval list* merupakan modul yang menampilkan dokumen yang sudah di submit pada modul sebelumnya dan menunggu persetujuan dokumen dari *user* dengan *role approval*. Pada modul ini, dokumen yang akan ditampilkan hanyalah dokumen yang mempunyai status *Submit*.

Modul ini hanya dapat diakses oleh *role approval* pada menu *transaction* pada *navigation bar* yang tertera pada bagian kiri halaman. *Button* pada halaman ini memiliki fungsi dan juga kegunaannya masing - masing yang tertera pada Tabel 3.4

Tabel 3.4. Approval List Action Control

Button Name	Function	Action
Details	Membuka material usage dari dokumen	membuka material usage dari dokumen

Tampilan desain awal dari modul *approval list* ini dapat dilihat pada Gambar 3.3



Gambar 3.3. Mockup Design Approval List MRP

A.5 Material Usage

Modul ini merupakan satu halaman yang menampilkan detail dari suatu dokumen. Pada halaman ini detail dari setiap stok barang yang ada akan ditampilkan, mulai dari jumlah stok yang ada digudang, stok yang akan datang, jumlah stok setelah diproses pada bulan ini, estimasi stok yang akan kedarluasa, dan masih banyak lagi data yang ditampilkan pada halaman ini. Data yang paling penting pada halaman ini adalah data jumlah stok yang harus dibeli pada beberapa bulan kedepan atau biasa disebut dengan *purchase requisition*.

Pada modul ini ada 3 jenis halaman yang berbeda yang memiliki fungsi tertentu tergantung dari status dokumen yang ada. Setiap halaman tersebut akan menampilkan *button* yang berbeda - beda. Tetapi ada beberapa *button* yang selalu ada dan berikut adalah fungsi dan kegunaannya tertera pada Tabel 3.5

Tabel 3.5. Material Usage Action Control

Button Name	Function	Action
Instant Export	Meng-export dokumen	meng-export table material Usage sesuai format pilihan
History	Melihat Log History perubahan pada suatu dokumen	Pop up modal berisi log history
Go back	kembali ke halaman sebelumnya	Kembali ke halaman sebelumnya

A.5.1 Material Usage Draft

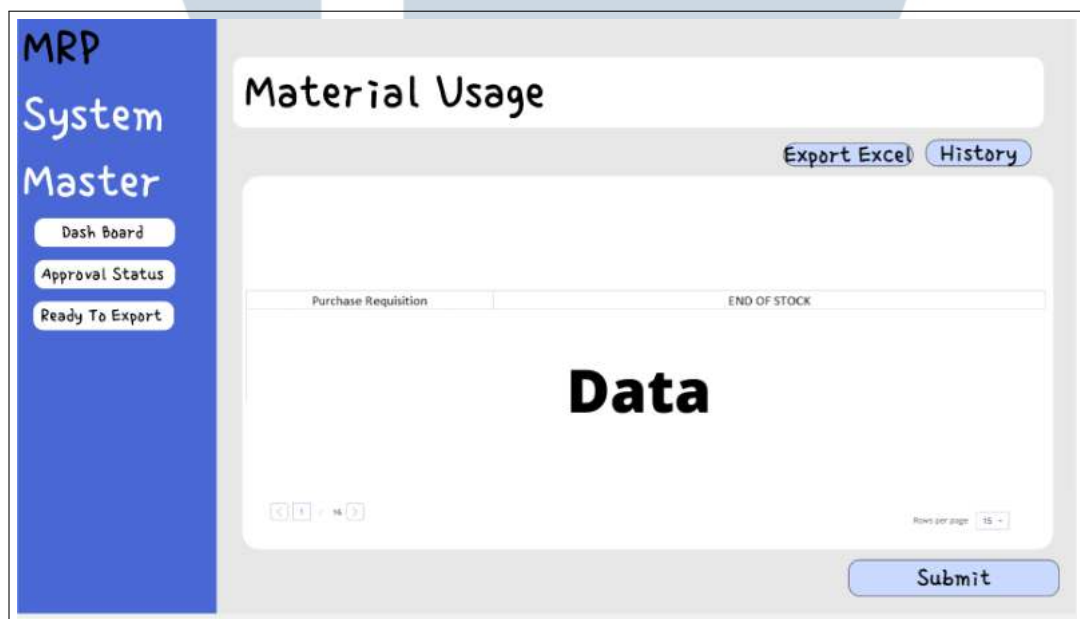
Halaman ini merupakan tampilan dokumen yang mempunyai status *draft*. Dokumen ini biasanya adalah dokumen yang baru saja selesai di kalkulasikan atau dokumen yang sudah disubmit lalu di *reject* oleh *role approval*.

Pada halaman ini pengguna dapat meng-edit *purchase requisition* yang sudah di kalkulasikan sebelumnya jika diperlukan, nantinya data yang baru akan otomatis tersimpan pada database. Berikut adalah fungsi dan kegunaan *button* yang ada pada halaman ini tertera pada Tabel 3.6

Tabel 3.6. Material Usage Draft Action Control

Button Name	Function	Action
Submit	Submit dokumen yang sudah di-review dan siap di submit	Status dokumen berubah menjadi "Submit"

Tampilan desain awal dari halaman *material usage draft* ini dapat dilihat pada Gambar 3.4. Adapun peraturan dari perusahaan untuk tidak mendokumentasikan data yang ada di halaman ini, maka dari itu data pada Gambar 3.4 akan dihilangkan.



Gambar 3.4. Mockup Design Material Usage Draft MRP

A.5.2 Material Usage Submit

Halaman ini merupakan tampilan dokumen yang mempunyai status *submit*. Dokumen dengan status ini merupakan dokumen yang sudah di review dan di submit oleh *role* PPIC. Halaman ini hanya dapat diakses oleh *user* dengan *role approval*.

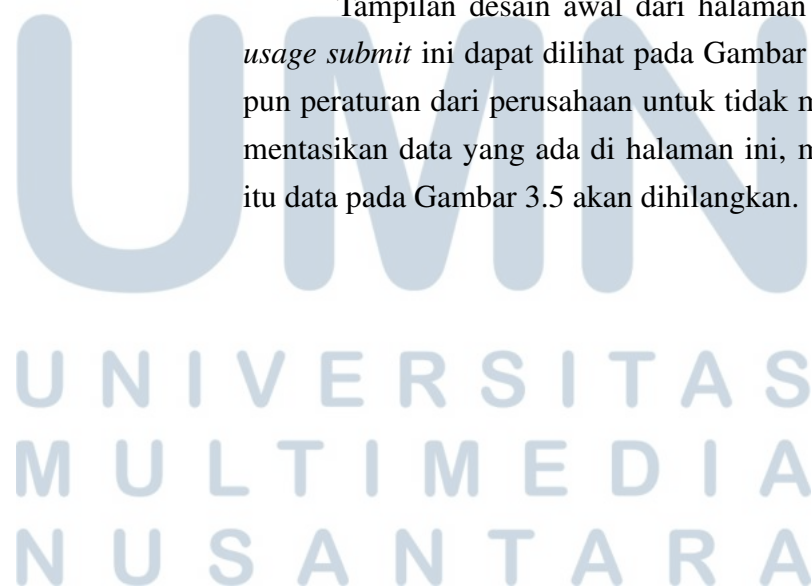
Pada halaman ini pengguna dapat meng-

approve dokumen ataupun mereject dokumen. Jika dokumen di approve, status dokumen akan berubah menjadi *approved* dan jika dokumen di reject, pengguna harus memasukan alasan reject pada modal yang muncul dan status dokumen akan berubah menjadi *draft*. Berikut adalah fungsi dan kegunaan *button* yang ada pada halaman ini tertera pada Tabel 3.7

Tabel 3.7. Material Usage Submit Action Control

Button Name	Function	Action
Approve	Approve dokumen yang sudah di submit	Status dokumen berubah menjadi "Approved"
Reject	Untuk mengeluarkan pop up modal yang nantinya di isi dengan alasan reject	Pop Up modal
Modal Reject	Reject Dokumen Yang sudah di submit dengan alasan	Mereject dokumen yang sudah disubmit

Tampilan desain awal dari halaman *material usage submit* ini dapat dilihat pada Gambar 3.5. Adapun peraturan dari perusahaan untuk tidak mendokumentasikan data yang ada di halaman ini, maka dari itu data pada Gambar 3.5 akan dihilangkan.



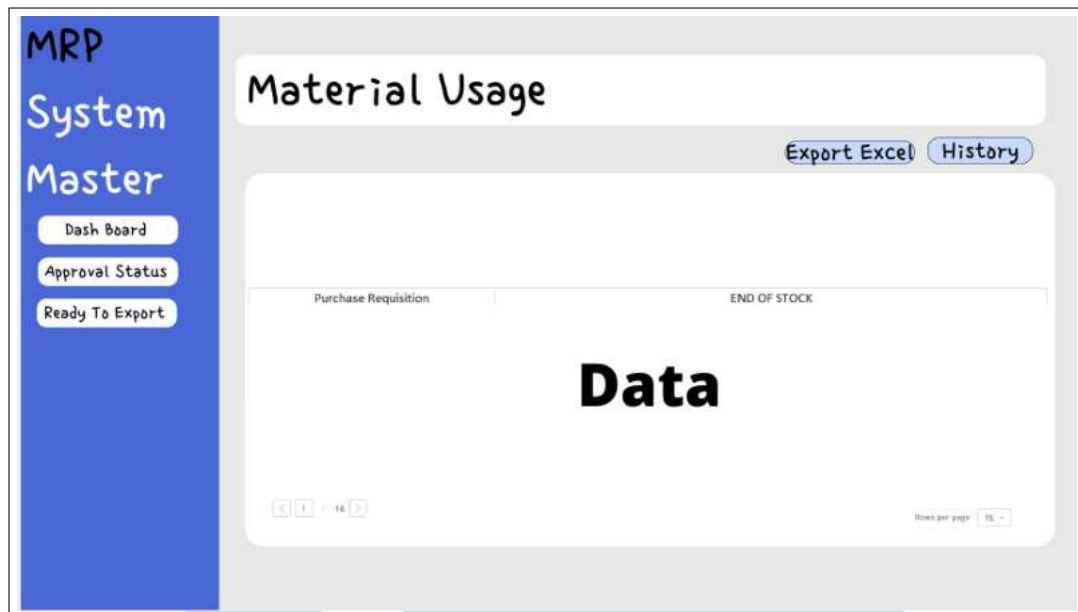


Gambar 3.5. Mockup Design Material Usage Submit MRP

A.5.3 Material Usage Approved

Halaman ini merupakan tampilan dokumen yang memiliki status *approved*. Dokumen yang memiliki status ini merupakan dokumen yang sudah di setujui oleh *role approval*. Halaman ini dapat diakses oleh semua *role yang ada*. Halaman ini bertujuan untuk melihat details dokumen yang sudah selesai di proses dan juga di setujui.

Tampilan desain awal dari halaman *material usage approved* ini dapat dilihat pada Gambar 3.6. Adapun peraturan dari perusahaan untuk tidak mendokumentasikan data yang ada di halaman ini, maka dari itu data pada Gambar 3.6 akan dihilangkan.

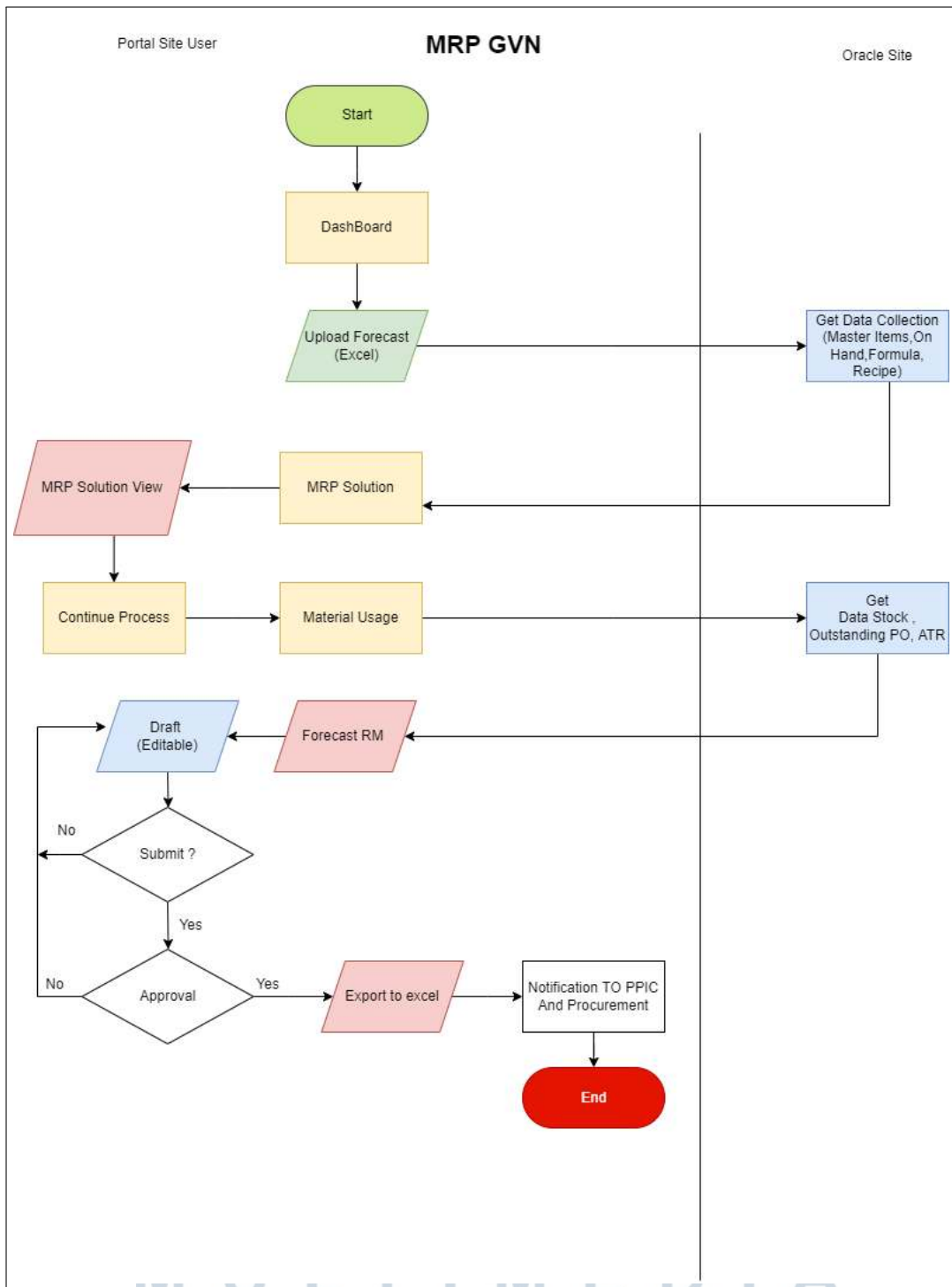


Gambar 3.6. Mockup Design Material Usage Approved MRP

B. Flowchart

B.1 MRP Flow System

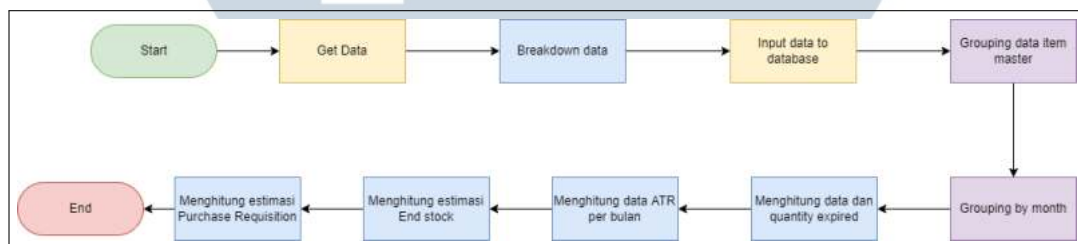
Alur sistem MRP ini berawal dari halaman dashboard dimana dihalaman tersebut pengguna harus mengunggah *file excel* serta berapa banyak bulan yang akan diproses dan juga tanggal awal mulai perhitungan. Setelah itu sistem akan menjalankan kalkulasi yang ada. Setelah proses kalkulasi selesai pengguna bisa melihat dan mengubah dokumen yang berada di dokumen view yang nantinya harus di-*submit* untuk melanjutkan proses berikutnya. Dokumen yang sudah di-*submit* akan di cek oleh *user approval* yang nantinya dokumen tersebut bisa di-*approve* ataupun di-*reject*. Dokumen yang sudah di *approve* akan bisa dilihat di halaman *document view* dan bisa di *export* sebagai *file excel* dan dokumen yang di-*reject* akan kembali menjadi dokumen dengan status *draft*. Berikut *flowchart* dari sistem MRP tertera pada Gambar 3.7.



Gambar 3.7. Flowchart MRP Flow System

B.2 MRP Calculation

Proses *MRP Calculation* merupakan proses perhitungan yang terjadi di dalam sistem ini. Data akan didapatkan dari data yang sudah diunggah pada halaman dashboard yang selanjutnya akan di *breakdown* sesuai ketentuan yang ada. Data yang sudah di *breakdown* akan disimpan kedalam database, selanjutnya data akan di-*grouping* berdasarkan *item master* dan juga bulan yang ada. Setelah data dikelompokkan, data yang ada akan digunakan untuk menghitung kadarluasa, ATR perbulan, estimasi *end stock*, dan juga *purchase requisition*. Berikut *flowchart* dari kalkulasi MRP tertera pada Gambar 3.8.



Gambar 3.8. Flowcart MRP Calculation

C. Hasil Implementasi

Berikut adalah hasil implementasi dari modul Dashboard, MRP Simulation, Document View, Approval List, dan juga Approval List setelah menjalani proses kerja magang.

C.1 Dashboard

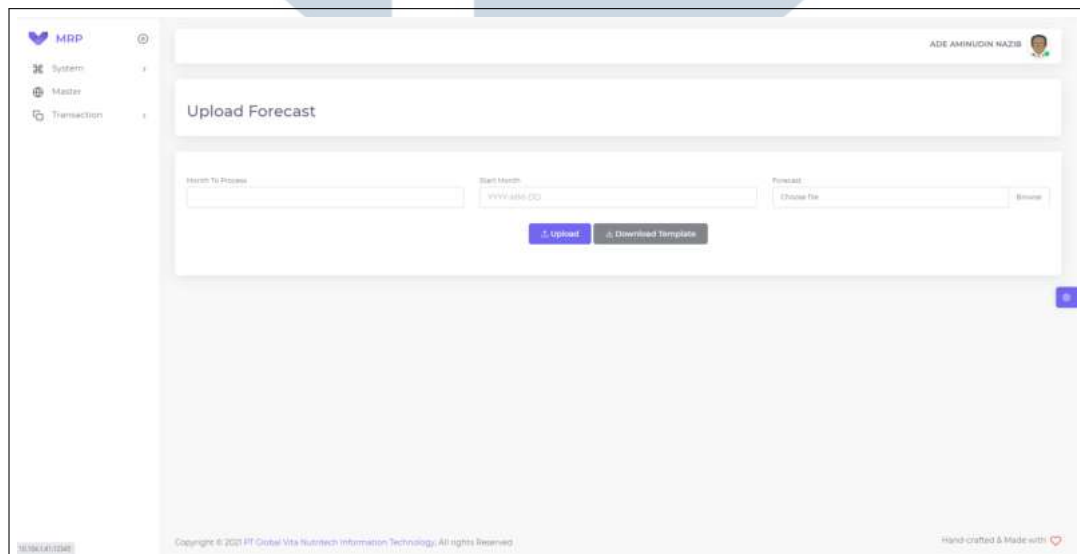
Modul ini merupakan tahap awal dari proses MRP. Modul ini dapat diakses dari menu *transaction* yang terletak pada *navigation bar* yang ada di sebelah kiri halaman. Modul ini berfungsi untuk menunggah *file excel forecast RM* yang nantinya akan digunakan untuk melakukan *MRP Simulation*.

Hasil dari modul Dashboard ini sudah sesuai dengan

rencana dan juga *requirements* awal yang sudah ditentukan sebelumnya. Berikut adalah hasil tampilan aplikasi dan juga potongan kode dari modul ini, Adapun peraturan dari perusahaan untuk tidak menampilkan beberapa data dan juga potongan kode.

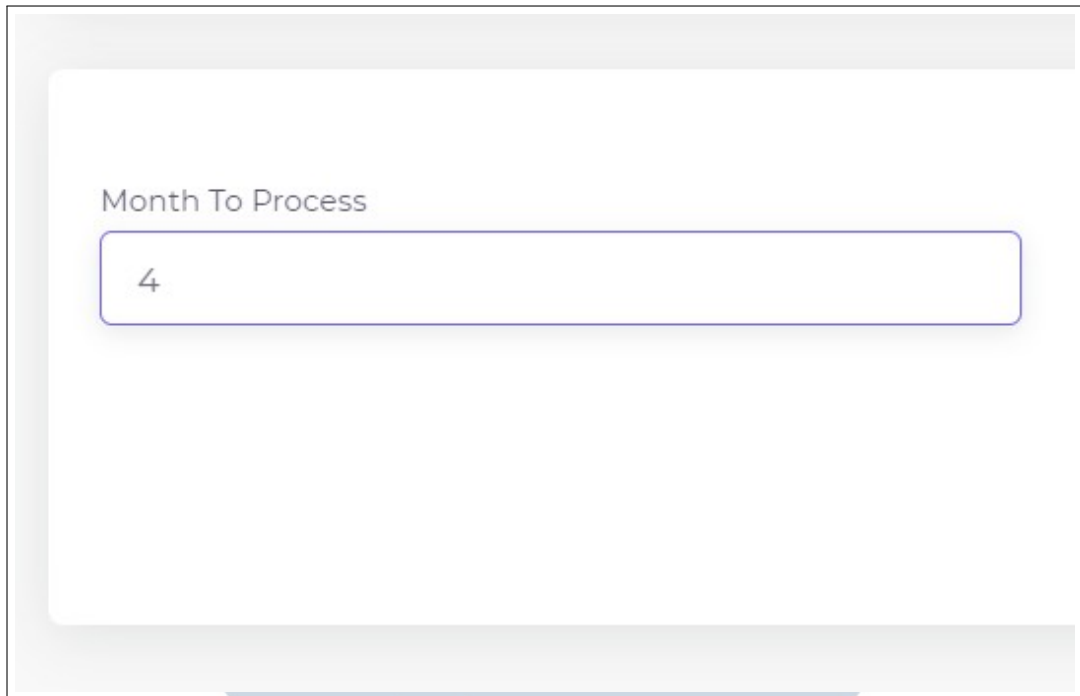
C.1.1 Tampilan Aplikasi

Berikut adalah tampilan aplikasi dari dashboard. Gambar 3.9 merupakan tampilan dashboard MRP, Gambar 3.10 merupakan tampilan month to process, Gambar 3.11 merupakan tampilan *choose starting date* pada dashboard MRP, dan Gambar 3.12 merupakan tampilan ketika pengguna mau melakukan *upload file* pada dashboard MRP.

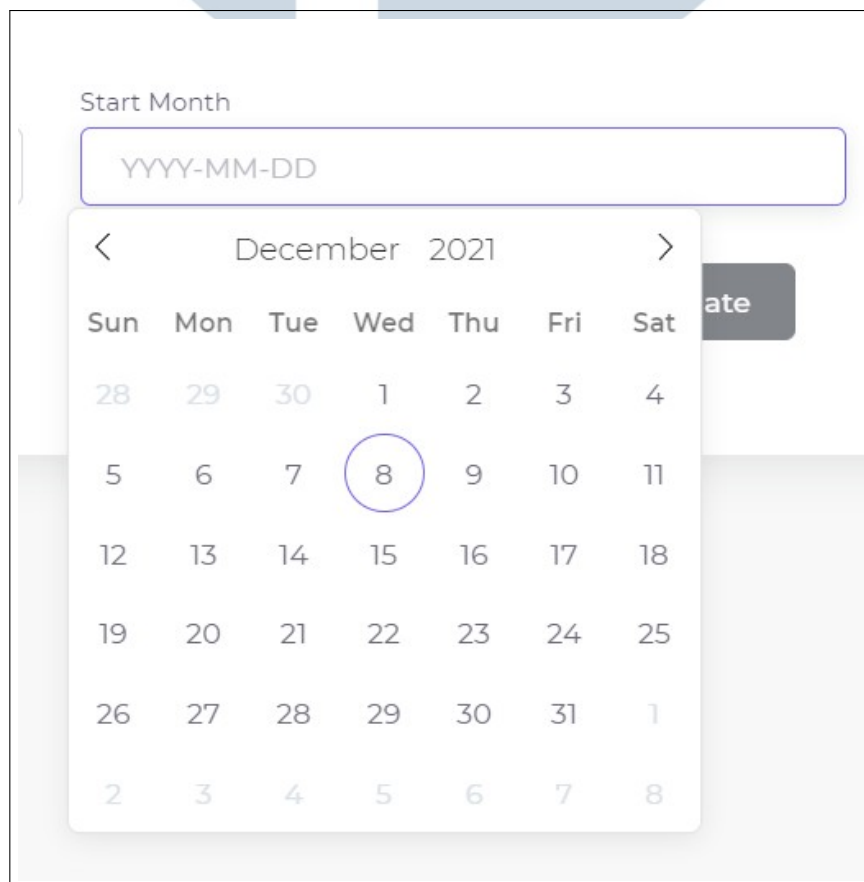


Gambar 3.9. Tampilan Dashboard MRP

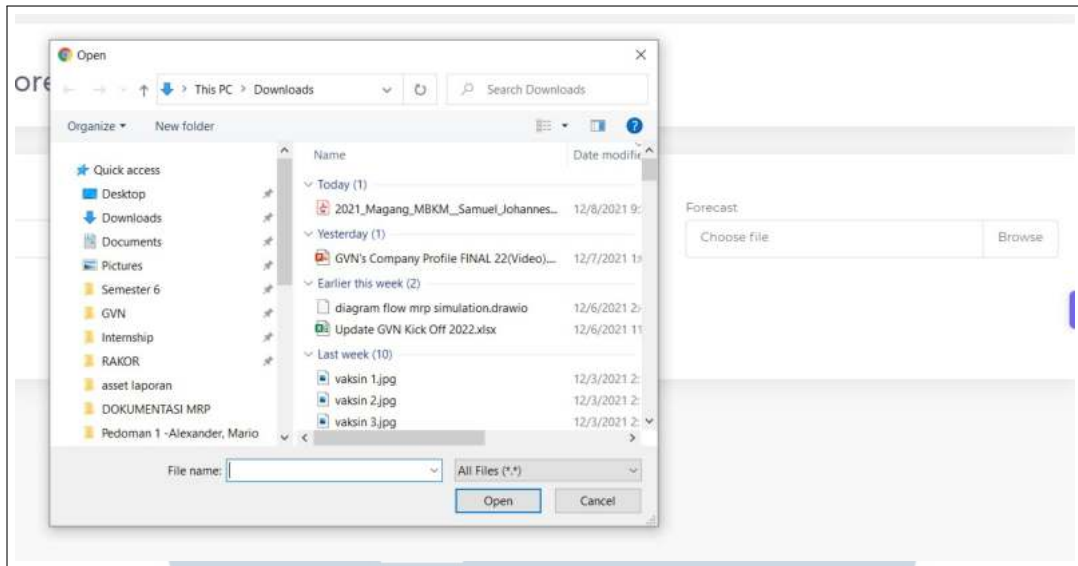
UNIVERSITAS
MULTIMEDIA
NUSANTARA



Gambar 3.10. Month to Process pada Dashboard



Gambar 3.11. Choose Starting Date pada Dashboard MRP



Gambar 3.12. Upload File pada Dashboard MRP

C.1.2 Potongan Kode

Berikut merupakan dokumentasi potongan kode dari modul dashboard MRP. Gambar 3.13 menampilkan html, Gambar 3.14 menampilkan potongan kode dari java script, dan Gambar 3.15 menampilkan potongan kode dari *controller* untuk *upload file excel* yang ada.

```

<div class="card-body">
  <div class="row">
    <div class="col-xl-4 col-md-6 col-12 mh-1">
      <div class="form-group">
        <label for="basicInput">Month To Process</label>
        <input type="number" class="form-control" id="intMonth">
      </div>
    </div>
    <div class="col-xl-4 col-md-6 col-12 mh-1">
      <div class="form-group">
        <label for="helpinputTop">Start Month</label>
        <input type="text" id="fp-default" class="form-control flatpickr-basic" placeholder="YYYY-MM-DD" />
      </div>
    </div>
    <div class="col-xl-4 col-md-6 col-12 mh-1">
      <div class="form-group">
        <label for="customFile">Forecast</label>
        <div class="custom-file">
          <input type="file" class="custom-file-input" id="InputFile">
          <label class="custom-file-label" for="customFile">Choose file</label>
        </div>
      </div>
    </div>
  </div>
  <center>
    <button type="button" class="btn btn-primary" onclick="return p_saveData();" <i data-feather="upload"></i> Upload</button>
    <a href="--/Data/example-template.xlsx" class="btn btn-secondary"><i data-feather="download"></i> Download Template</a>
  </center>

```

Gambar 3.13. Index.cshtml Dashboard MRP

```

@section scripts{
<script src="~/Scripts/customs/GlobalScriptVuexy.js"></script>
<script>
    var clsGlobal = new clsGlobalClass();
    //$('#frmUpload').on('submit', function () {
    //    p_saveData();
    //    return false;
    //})

    function p_saveData() {
        clsGlobal.showLoading();
        var data = new FormData();
        var files = $("#InputFile").get(0).files;
        if (files.length > 0) {
            data.append("InputFile", files[0]);
            data.append("intMonthToProcess", $("#intMonth").val());
            data.append("dtStartMonth", $("#fp-default").val());
        }
    }
}

```

Gambar 3.14. JavaScript pada Index Dashboard MRP

```

[FilterConfig.CheckSessionTimeout()]
[FilterConfig.CheckAuthorizationAttribute]
[AcceptVerbs(HttpVerbs.Post)]
[HttpPost]
public ActionResult UploadForecast()
{
    try
    {
        int intMonthToProcess = clsGlobal.ParseToInteger(System.Web.HttpContext.Current.Request.Form["intMonthToProcess"].ToString());
        DateTime dtStartMonth = clsGlobal.ParseToDate(System.Web.HttpContext.Current.Request.Form["dtStartMonth"].ToString());
        var dtEndMonth = dtStartMonth.AddMonths(intMonthToProcess - 1);
        List<DataUploadWB> retListDat = new List<DataUploadWB>();

        var txtMessage = string.Empty;
        string filename = "";
        if (System.Web.HttpContext.Current.Request.Files.AllKeys.Any())
        {
            //InputFile ambil dari form upload di cshtml id="InputFile"
            var file = System.Web.HttpContext.Current.Request.Files["InputFile"];
            if (file.ContentLength > 0)
            {
                string fileExtension = System.IO.Path.GetExtension(file.FileName);
                if (fileExtension == ".xls" || fileExtension == ".xlsx")
                {
                    //untuk simpan file upload ke web server folder "Data"
                    string fileLocation = System.IO.Path.Combine(Server.MapPath("~/Data/") + Request.Files["InputFile"].FileName);
                    filename = file.FileName;
                    if (System.IO.File.Exists(fileLocation))
                    {
                        filename = filename.Substring(0, filename.Length - fileExtension.Length);
                        fileLocation = System.IO.Path.Combine(Server.MapPath("~/Data/") + filename + "_" + DateTime.Now.ToString("yyMMdd-hhmmss") + fileExtension);
                        //System.IO.File.Delete(fileLocation);
                        filename = filename + "_" + DateTime.Now.ToString("yyMMdd-hhmmss") + fileExtension;
                    }
                }
                file.SaveAs(fileLocation);
            }
        }
    }
}

```

Gambar 3.15. Controller Upload Excel pada Dashboard MRP

C.2 MRP Simulation

Modul MRP *Simulaton* merupakan modul perhitungan atau kalkulasi yang ada di proyek ini. Modul ini bertujuan untuk menghitung perkiraan *purchase requisition* untuk setiap material yang ada berdasarkan dari *forecast RM* yang sudah di upload sebelumnya. Pada saat perhitungan, data akan diambil dari excel yang sudah di upload dan juga

mengambil data dari oracle secara *real time*.

Hasil dari pengerjaan modul ini sesuai dengan rencana dan *requirement* awal. Halaman ini tidak mempunyai tampilan, proses kalkulasi semua berajalan di *background*, adapun peraturan dari perusahaan untuk tidak menampilkan potongan kode untuk bagian kalkulasi ini.

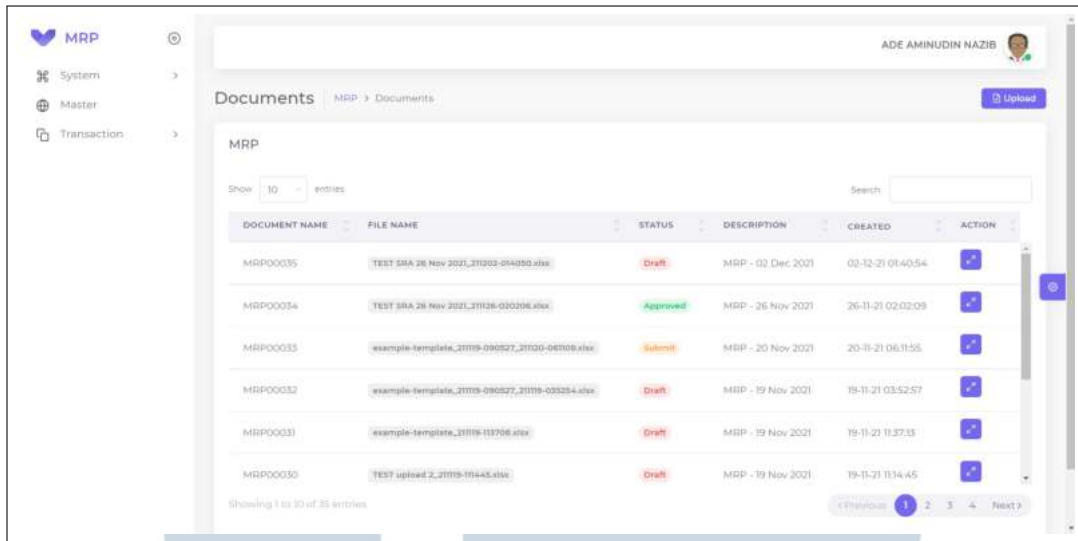
C.3 Document View

Modul ini merupakan modul untuk menampilkan dokumen-dokumen yang sudah di kalkulasikan. Modul ini berfungsi untuk menampilkan *list* dokumen-dokumen yang ada dan mempunyai *button* untuk menuju *detail* atau juga disebut *material usage* dari dokumen tersebut. Modul ini dapat diakses dari menu *transacation* yang terletak pada *navigation bar* yang ada di sebelah kiri halaman. Modul ini dapat diakses oleh *user PPIC* dan juga *approval*, tetapi untuk *user approval* hanya bisa melihat dokumen yang berstatus *submit* dan *approved*, sedangkan untuk *user PPIC* bisa melihat dari semua status yang ada.

Hasil implementasi dari modul *Document View* ini sudah sesuai dengan rencana dan juga *requirements* awal yang sudah ditentukan sebelumnya. Berikut adalah tampilan aplikasi dan juga potongan kode dari modul ini, Adapun peraturan dari perusahaan untuk tidak menampilkan beberapa data dan juga potongan kode.

C.3.1 Tampilan Aplikasi







Berikut merupakan tampilan aplikasi dari modul *document view*. Gambar 3.16 merupakan tampilan *document view* pada MRP dan Gambar 3.17 merupakan tampilan *button* menuju *material usage* suatu dokumen.



Gambar 3.16. Tampilan *Document View* MRP

UMN
UNIVERSITAS
MULTIMEDIA
NUSANTARA

Search:

DESCRIPTION	CREATED	ACTION
MRP - 02 Dec 2021	02-12-21 01:40:54	
d MRP - 26 Nov 2021	26-11-21 02:02:09	
MRP - 20 Nov 2021	20-11-21 06:11:55	
MRP - 19 Nov 2021	19-11-21 03:52:57	
MRP - 19 Nov 2021	19-11-21 11:37:13	
MRP - 19 Nov 2021	19-11-21 11:14:45	

< Previous 1 2 3 4 Next >

Gambar 3.17. Button menuju Material Usage Dokumen

C.3.2 Potongan Kode

Potongan kode untuk modul *Document View* tertera pada Gambar 3.18 untuk halaman index dan Gambar 3.19 untuk Controller.

```

67 <div class="card-body">
68
69 <div class="table-responsive">
70 <table id="tblDocument" class="table">
71 <thead>
72 <tr>
73 <th><%= Html.Label("Document Name") %>/th>
74 <th><%= Html.Label("File Name") %>/th>
75 <th><%= Html.Label("Status") %>/th>
76 <th><%= Html.Label("Description") %>/th>
77 <th><%= Html.Label("Created") %>/th>
78 <th><%= Html.Label("Action") %>/th>
79 </tr>
80 </thead>
81 <tbody>
82 <tr>
83 <td colspan="6"><%= Html.Label("No Document Found") %></td>
84 </tr>
85 <tr>
86 <td colspan="6"><%= Html.Label("No Document Found") %></td>
87 </tr>
88 <tr>
89 <td colspan="6"><%= Html.Label("No Document Found") %></td>
90 </tr>
91 <tr>
92 <td colspan="6"><%= Html.Label("No Document Found") %></td>
93 </tr>
94 <tr>
95 <td colspan="6"><%= Html.Label("No Document Found") %></td>
96 </tr>
97 <tr>
98 <td colspan="6"><%= Html.Label("No Document Found") %></td>
99 </tr>
100 <tr>
101 <td colspan="6"><%= Html.Label("No Document Found") %></td>
102 </tr>
103 <tr>
104 <td colspan="6"><%= Html.Label("No Document Found") %></td>
105 </tr>
106 <tr>
107 <td colspan="6"><%= Html.Label("No Document Found") %></td>
108 </tr>

```

Gambar 3.18. DocumentIndex.cshtml Document View MRP

```

namespace GV2021_MRP.MVC.Controllers.Master
{
    0 references
    public class DocumentViewController : Controller
    {
        // GET: DocumentView
        [FilterConfig.CheckSessionTimeOut()]
        [FilterConfig.CheckAuthorizationAttribute]
        0 references
        public ActionResult DocumentIndex()
        {
            List<trDocument> data = new List<trDocument>();

            try
            {
                if (GlobalClass.dLogin.roleDat.txtRoleName == "Approval")
                {
                    data = clsDocumentBL.GetDataApprovalDocument();
                    return View(data);
                }

                else
                {
                    data = clsDocumentBL.GetDataDocument();
                    return View(data);
                }
            }
            catch (Exception ex)
            {
                return RedirectToAction("DocumentIndex");
            }
        }
    }
}

```

Gambar 3.19. Controller Document View MRP

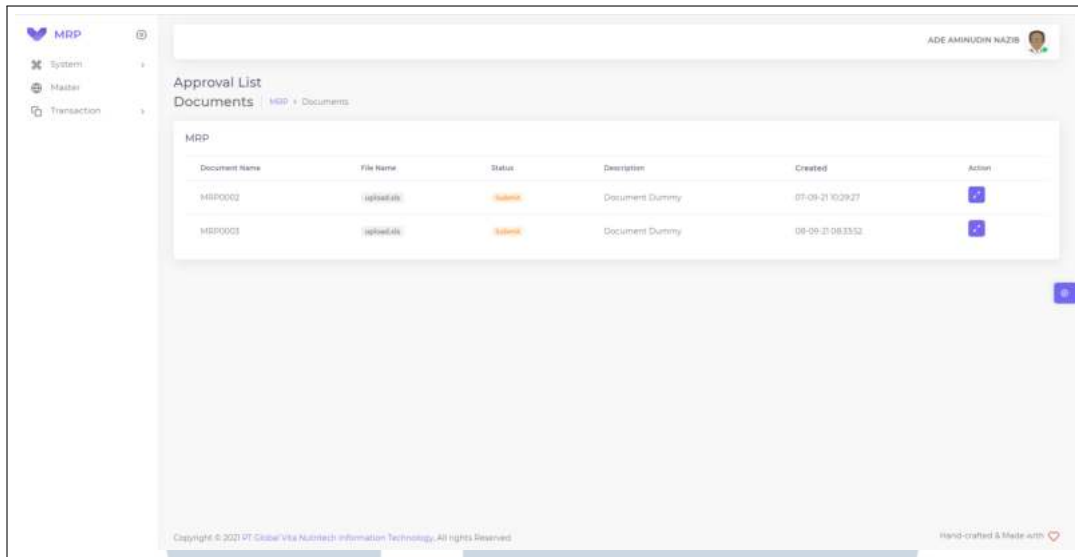
C.4 Approval List

Modul ini merupakan modul untuk menampilkan dokumen - dokumen yang sudah di *submit* sebelumnya. Dokumen yang masih berstatus *draft* ataupun dokumen yang sudah berstatus *approved* tidak akan muncul pada halaman ini, hanya yang berstatus dokumen yang berstatus *submit* yang akan muncul pada halaman ini. Modul ini berfungsi untuk menampilkan *list* dokumen-dokumen yang butuh persetujuan dari *role approval*. Modul ini mempunyai *button* untuk menuju *detail* atau juga disebut *material usage* dari dokumen tersebut. Modul ini dapat diakses dari menu *transacation* yang terletak pada *navigation bar* yang ada di sebelah kiri halaman. Modul ini hanya dapat diakses oleh *user Approval*.

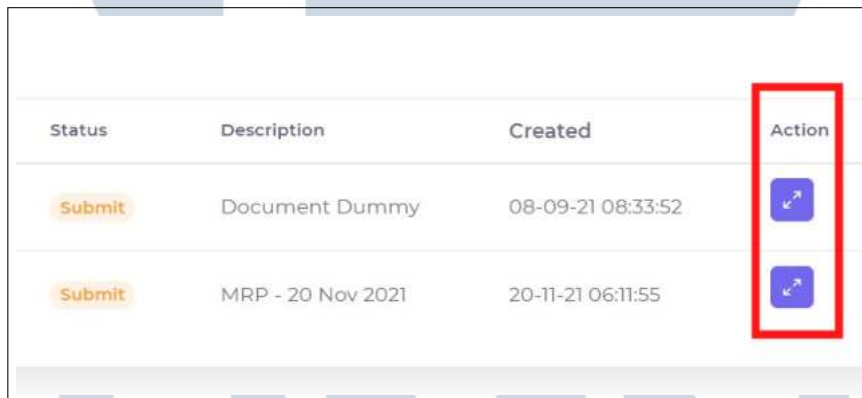
Hasil implementasi dari modul Approval List ini sudah sesuai dengan rencana dan juga *requirements* awal yang sudah ditentukan terlebih dahulu. Berikut adalah tampilan aplikasi dan juga potongan kode dari modul Approval List, Adapun peraturan dari perusahaan untuk tidak menampilkan beberapa data dan juga potongan kode.

C.4.1 Tampilan Aplikasi

Berikut merupakan tampilan aplikasi dari modul *Approval List* Gambar 3.20 merupakan tampilan *Approval List* dan Gambar 3.21 merupakan *button* menuju *material usage* suatu dokumen.



Gambar 3.20. Tampilan *Approval List* MRP



Gambar 3.21. *Button* menuju *Material Usage* Dokumen

C.4.2 Potongan Kode

Potongan kode untuk modul *Approval List* tertera pada Gambar 3.22 untuk halaman index dan Gambar 3.23 untuk Controller.

UNIVERSITAS
MULTIMEDIA
NUSANTARA

```

<div class="card-body">
  <div class="table-responsive">
    <table class="table">
      <thead>
        <tr>
          <th><%= Html.Label("Document Name") %>/th>
          <th><%= Html.Label("Title Name") %>/th>
          <th><%= Html.Label("Status") %>/th>
          <th><%= Html.Label("Description") %>/th>
          <th><%= Html.Label("Created") %>/th>
          <th><%= Html.Label("Action") %>/th>
        </tr>
      </thead>
      <tbody>
        <tr>
          <td>
            <%= item.txtDocumentName %>
          </td>
          <td>
            <span class="badge badge-light-secondary"><%= item.txtFileName %>/span>
          </td>
          <td>
            <%= item.txtStatus %>
          </td>
          <td>
            <span class="badge badge-pill badge-light-danger"><%= item.txtStatus %>/span>
          </td>
          <td>
            <%= item.txtCreated %>
          </td>
          <td>
            <%= item.txtAction %>
          </td>
        </tr>
      </tbody>
    </table>
  </div>
</div>

```

Gambar 3.22. *ApprovalIndex.cshtml Approval List MRP*

```

0 references
public class ApprovallistController : Controller
{
    // GET: Approvallist
    [FilterConfig.CheckSessionTimeout()]
    [FilterConfig.CheckAuthorizationAttribute]
    0 references
    public ActionResult ApprovalIndex()
    {
        List<trDocument> data = new List<trDocument>();

        try
        {
            data = clsDocumentBL.GetDataSubmitDocument();
            return View(data);
        }
        catch (Exception ex)
        {
            return RedirectToAction("ApprovalIndex");
        }
    }
}

```

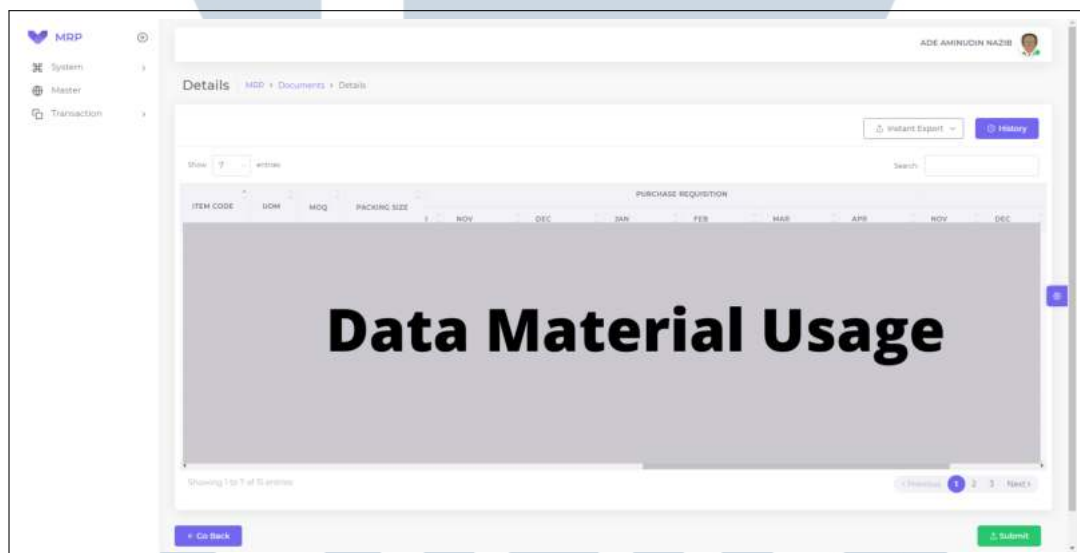
Gambar 3.23. *Controller Approval List MRP*

C.5 Material Usage Draft

Halaman ini adalah bagian dari modul *Material Usage* yang hanya menampilkan dokumen dengan status *draft*. Halaman ini menggunakan *plugin datatable* untuk menampilkan

data yang ada di database. Terdapat dua fitur dari halaman ini yaitu fitur *Update Purchase Requisition Data* dan juga fitur *submit* dokumen. Halaman ini dapat diakses melalui modul *document view* dengan menekan *button detail* di dokumen yang berstatus *draft*.

Berikut adalah tampilan aplikasi dan juga potongan kode dari halaman ini. Adapun peraturan dari perusahaan untuk tidak menampilkan beberapa data dan juga potongan kode, maka dari itu ada beberapa gambar dan potongan kode yang di sensor ataupun tidak dimasukan. Gambar 3.24 merupakan tampilan dari *material usage draft* dan Gambar 3.25 merupakan potongan kode dari halaman *material usage draft*.



Gambar 3.24. Tampilan Material Usage Draft

UNIVERSITAS
MULTIMEDIA
NUSANTARA

Berikut merupakan potongan kode dari *Update Purchase Requisition Data*. Gambar 3.27 merupakan form inputan pada halaman *material usage*, Gambar 3.28 merupakan JavaScript dari fungsi dari *Update Purchase Requisition Data*, Gambar 3.29 merupakan controller dari *Update Purchase Requisition Data*, dan yang terakhir Gambar 3.30 merupakan *buzzlogic* dari fungsi *Update Purchase Requisition Data*.

```

if (item.txtStatus == "Draft")
{

}

```

Gambar 3.27. Form Input pada Halaman HTML

```

function p_txtTableValue_Changed(objCaller, intIndex, decparam,intCalculationId) {
    debugger;
    // objCaller.value =
    var suggestvalue = $('#decSuggestValue_' + intIndex).val();

    clsGlobal.showLoading();
    $.ajax({
        type: "POST",
        url: "/Master/materialUsage/UpdateItem",
        data: { suggestvalue: suggestvalue, intCalculationId: intCalculationId},
        success: function (retDat) {
            console.log(retDat);
            if (retDat.bitSuccess == true) {
                clsGlobal.getInformationMessage(retDat.txtMessage);
            } else {
                clsGlobal.getAlert(retDat.txtMessage);
            }
            clsGlobal.hideLoading();
        },
        error: function (retDat) {
            clsGlobal.hideLoading();
        }
    });
}

```

Gambar 3.28. JavaScript untuk *Update Purchase Requisition data*

UNIVERSITAS
MULTIMEDIA
NUSANTARA

```

[HttpPost()]
[FilterConfig.CheckSessionTimeout()]
[FilterConfig.CheckAuthorizationAttribute]
0 references
public ActionResult UpdateItem(string suggestvalue, int intCalculationId)
{
    try
    {
        bool bitSuccess = false;
        trCalculation objDat = new trCalculation();
        objDat.decSuggestValue = clsGlobal.ParseToDecimal(suggestvalue);
        objDat.intCalculationId = intCalculationId;
        string txtStatus = string.Empty;
        if (intCalculationId != 0)
        {
            if (clsCalculationBL.IsExisttrCalculation(objDat.intCalculationId))
            {
                bitSuccess = clsCalculationBL.UpdateData(objDat, GlobalClass.dLogin.userDat.txtUserName);
            }
            else
            {
                return View();
            }
        }
        return Json(clsAPI.CreateResult(bitSuccess, objDat, txtStatus, string.Empty));
    }
}

```

Gambar 3.29. Controller untuk *Update Purchase Requisition Data*

```

query.txtUpdatedBy = txtUserName;
query.dtmUpdated = DateTime.Now;
query.decSuggestValue = data.decSuggestValue;

dObjContext.trCalculations.Add(query);
dObjContext.Entry(query).State = EntityState.Modified;
dObjContext.SaveChanges();

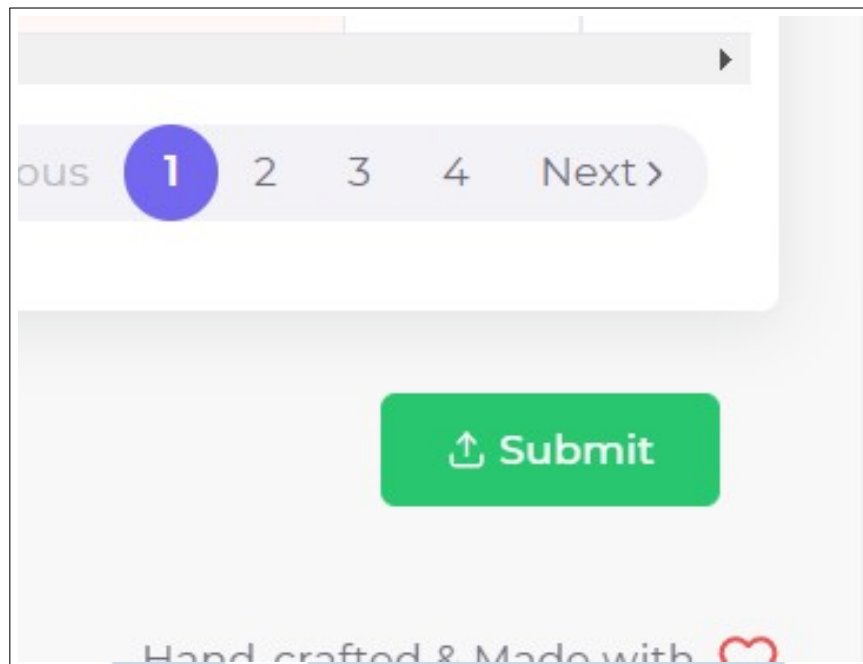
return true;

```

Gambar 3.30. Buzzlogic untuk *Update Purchase Requisition Data*

C.5.2 Submit

Pada halaman ini pengguna dapat mengumpulkan atau *submit* dokumen yang sudah ada. Dokumen yang sudah di kalkuklasikan harus dikumpulkan untuk melanjutkan proses *approval* dari *role approval*. Status dokumen nantinya akan berubah dari *draft* menjadi *submit* setelah dokumen dikumpulkan. Tampilan *Button submit* tertera pada Gambar 3.31



Gambar 3.31. *Button Submit* pada Material Usage Draft

Berikut merupakan potongan kode dari fungsi *Submit*. Gambar 3.32 merupakan Potongan Kode fungsi *Submit* pada Halaman HTML, Gambar 3.33 adalah controller untuk fungsi *Submit*, dan yang terakhir Gambar 3.34 adalah *buzzlogic* untuk fungsi *Submit* yang akan merubah status dokumen dari *draft* menjadi *submit*.

```
if (item.txtStatus == "Draft")  
<a href="/Transaction/materialUsage/UpdateDocumentIntoSubmit?id=HttpUtility.UrlEncode  
item.txtStatus" >Submit
```

Gambar 3.32. Potongan Kode fungsi *Submit* pada Halaman HTML

UNIVERSITAS
MULTIMEDIA
NUSANTARA

```

[FilterConfig.CheckSessionTimeout()]
[FilterConfig.CheckAuthorizationAttribute]
[HttpGet()]

0 references
public ActionResult UpdateDocumentIntoSubmit(String id)
{
    try
    {
        int intDocumentId = clsGlobal.ParseToInteger(clsRijndael.Decrypt(id));
        bool bitSuccess = false;

        if (intDocumentId != 0)
        {
            if (clsUpdateDocumentStatusBL.IsExisttrDocument(objDat.intDocumentId))
            {
                bitSuccess = clsUpdateDocumentStatusBL.UpdateStatusIntoSubmit(objDat, GlobalClas
            }
            else
            {
                return View();
            }
        }
        return RedirectToAction("DocumentIndex", "DocumentView");
    }
    catch (Exception ex)
    {
        return Json(clsMainBL.CreateError(ex, GlobalClass.dLogin.userDat.intUserID.ToString(),
    }
}

```

Gambar 3.33. Potongan Kode fungsi *Submit* pada Controller

```

// Set User login.
query.txtUpdatedBy = txtUserName;
query.dtmUpdated = DateTime.Now;
query.txtStatus = "Submit";

dObjectContext.trDocuments.Add(query);
dObjectContext.Entry(query).State = EntityState.Modified;
dObjectContext.SaveChanges();

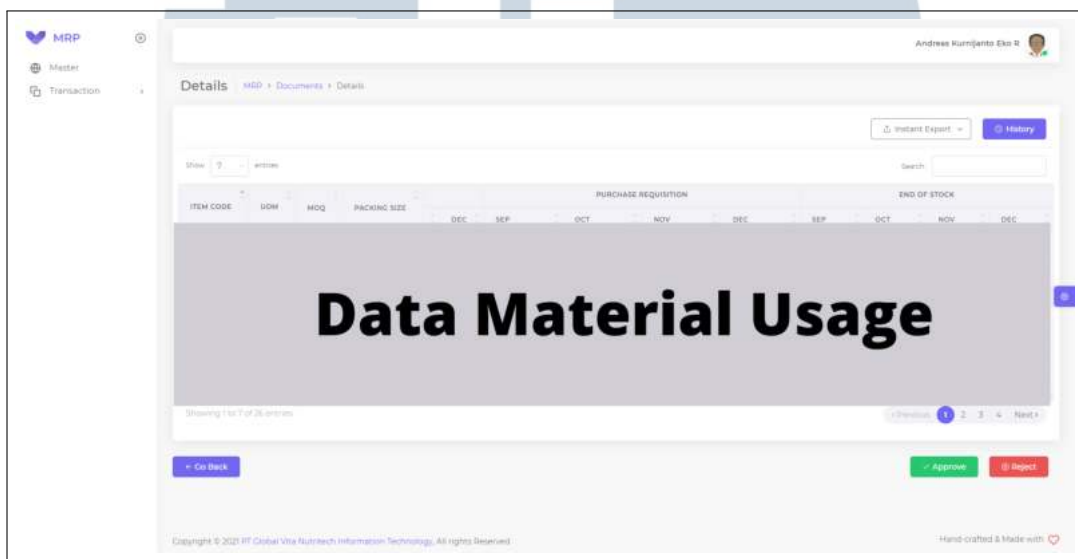
```

Gambar 3.34. Potongan Kode fungsi *Submit* pada Buzzlogic

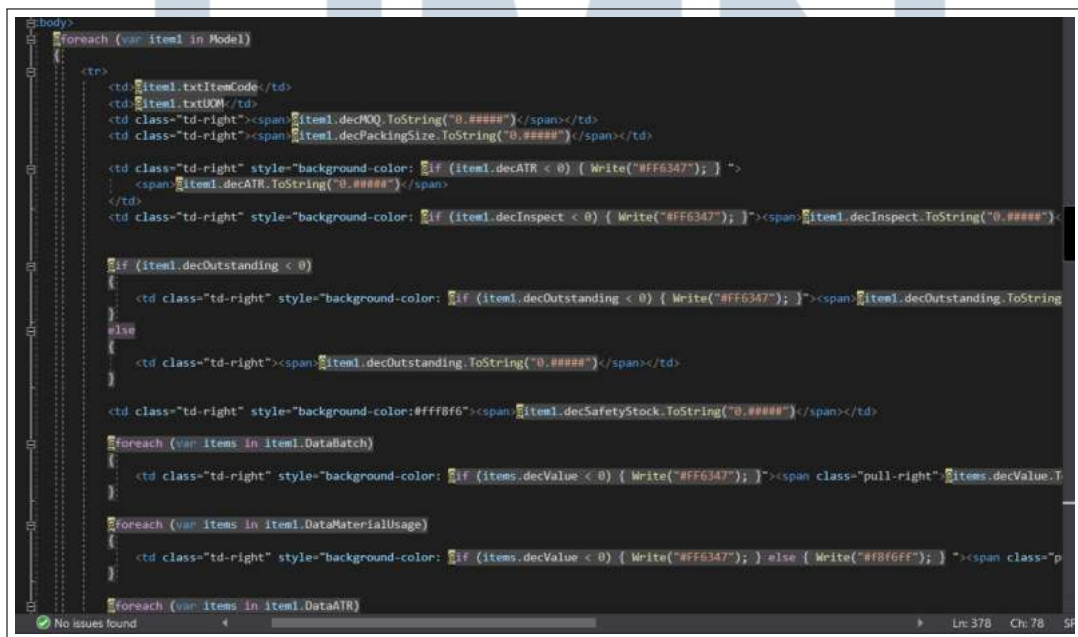
C.6 Material Usage Submit

Halaman ini adalah bagian dari modul *Material Usage* yang hanya menampilkan dokumen dengan status *submit*. Halaman ini menggunakan *plugin datatable* untuk menampilkan data yang ada di database. Terdapat dua fitur dalam halaman ini yaitu *approve document* dan juga *reject document*. Halaman ini hanya dapat diakses oleh *role approval* melalui modul *document view* dengan menekan *button detail* di dokumen yang berstatus *submit*.

Berikut adalah tampilan aplikasi dan juga potongan kode dari halaman ini. Adapun peraturan dari perusahaan untuk tidak menampilkan beberapa data dan juga potongan kode, maka dari itu ada beberapa gambar dan potongan kode yang di sensor ataupun tidak dimasukkan. Gambar 3.35 merupakan tampilan aplikasi dari *material usage submit* dan Gambar 3.36 merupakan potongan kode dari *material usage submit*.



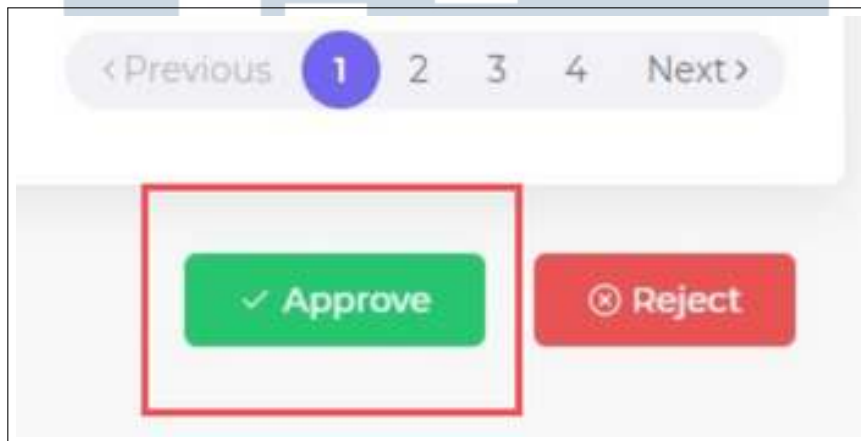
Gambar 3.35. Tampilan Material Usage Submit



Gambar 3.36. Potongan Kode Material Usage Submit

C.6.1 Approve

Pada halaman ini pengguna dapat menyetujui atau *approve* suatu dokumen. Hal ini hanya dapat dilakukan oleh *user approval*, jika pengguna bukan *user approval* *button approve* tidak akan muncul pada halaman. Status dokumen yang sudah diapprove nantinya akan berubah dari *submit* menjadi *approved*. Tampilan *button approve* tertera pada Gambar 3.37



Gambar 3.37. Tampilan *Button Approve* pada Material Usage Submit

Berikut merupakan potongan kode dari fungsi *approve*. Gambar 3.38 merupakan Potongam kode fungsi *approve* pada halaman HTML, Gambar 3.39 merupakan potongan kode fungsi *approve* yang terdapat pada controller, dan Gambar 3.40 merupakan potongan kode fungsi *approve* yang berada pada *buz-zlogic* yang akan merubah status dokumen dari *submit* menjadi *approved*.

```

<div id="item-btnStatus" class="btn btn-success">
  <button class="btn btn-danger float-right" style="margin-bottom: 5px; margin-right: 10px;" id="htmlRejectAllocation" data-feather="x-circle"> Reject </button>
  <a href="/transaction/materialusage/submitdocumentintoApproved?id=<id>&token=<token>" class="btn btn-success float-right" style="margin-bottom: 5px; margin-right: 10px;"> Approve </a>
</div>

```

Gambar 3.38. Potongan Kode fungsi *Approve* pada halaman HTML

```

[HttpPost()]
[FilterConfig.CheckSession(true)]
[FilterConfig.CheckAuthorization(true)]
public ActionResult UpdateDocumentInfoApproved(string id)
{
    try
    {
        int intDocumentId = Global.ParseToInt(id);
        bool bitSuccess = false;

        if (intDocumentId != 0)
        {
            if (!Global.UpdateDocumentStatus.ExistsDocument(objDet.IntDocumentId))
            {
                bitSuccess = Global.UpdateDocumentStatus.UpdateStatusInfoApproved(objDet, Global.Login.UserDet.txtUsername, Global.Login.UserDet.txtFullName);
            }
            else
            {
                return View();
            }
        }
        return RedirectToAction("DocumentIndex", "DocumentInfo");
    }
    catch (Exception ex)
    {
        return Json(Global.CreateError(ex, Global.Login.UserDet.IntUserID.ToString(), Global.Login.UserDet.txtAngID.ToString(), Request.Url.ToString(), Global.Login.UserDet, JsonRequestBehavior.AllowGet));
    }
}

```

Gambar 3.39. Potongan Kode fungsi *Approve* pada Controller

```

query.txtUpdatedBy = txtUserName;
query.dtmUpdated = DateTime.Now;
query.txtStatus = "Approved";

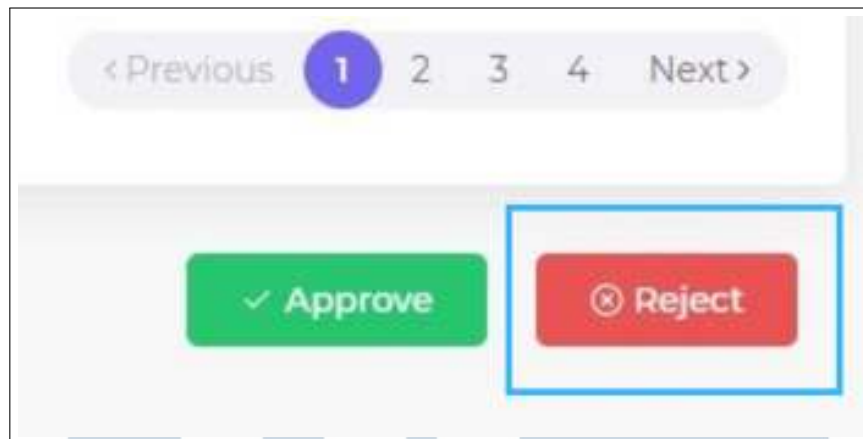
dobjContext.trDocuments.Add(query);
dobjContext.Entry(query).State = EntityState.Modified;
dobjContext.SaveChanges();

```

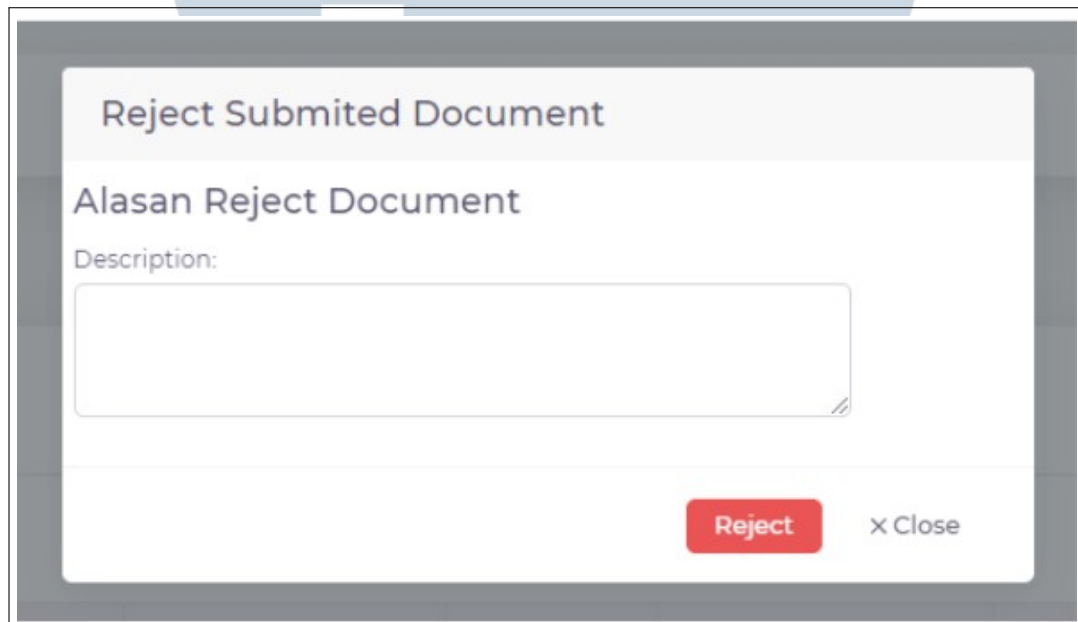
Gambar 3.40. Potongan Kode fungsi *Approve* pada Buzzlogic

C.6.2 Reject

Pada halaman ini pengguna dapat menolak atau *reject* suatu dokumen. Penolakan ini harus diikuti oleh alasan, nantinya akan ada *pop up* modal yang keluar untuk pengguna memasukkan alasan *reject* dokumen. Alasan akan tersimpan dalam database yang nantinya bisa ditampilkan pada *Log Status*. fungsi *reject* hanya dapat dilakukan oleh *user approval*, jika pengguna bukan *user approval* maka *button reject* tidak akan muncul pada halaman. Status dokumen yang sudah disumbit tetapi direject akan kembali lagi menjadi *draft*. Tampilan *button reject* tertera pada Gambar 3.41 dan tampilan modal reject tertera pada Gambar 3.42



Gambar 3.41. Tampilan *Button Reject* pada Material Usage Submit



Gambar 3.42. Tampilan modal *reject*

Berikut merupakan potongan kode dari fungsi reject. Gambar 3.43 merupakan potongan kode fungsi *reject* pada halaman HTML, Gambar 3.44 merupakan potongan kode modal yang akan muncul ketika pengguna mereject suatu dokumen. Gambar 3.45 merupakan potongan kode Javascript untuk memunculkan modal dan juga penyimpanan alasan fungsi *reject* pada database, Gambar 3.46 merupakan po-

ongan kode controller yang mengoper alasan *reject* dari Javascript menuju buzzlogic, Gambar 3.47 merupakan buzzlogic untuk mengubah status kembali menjadi *draft* dan Gambar 3.48 merupakan buzzlogic untuk menyimpan alasan *reject* di dalam *log status*.

```

<div class="modal-dialog">
  <div class="modal-content">
    <div class="modal-header">
      <h4>Reject Submitted Document</h4>
    </div>
    <div class="modal-body">
      <div class="form-group">
        <input type="text" id="reason" value="" required="" />
      </div>
      <div class="text-right">
        <button type="submit" class="btn btn-danger">Reject</button>
        <button type="button" class="btn btn-default" data-dismiss="modal">Close</button>
      </div>
    </div>
  </div>
</div>

```

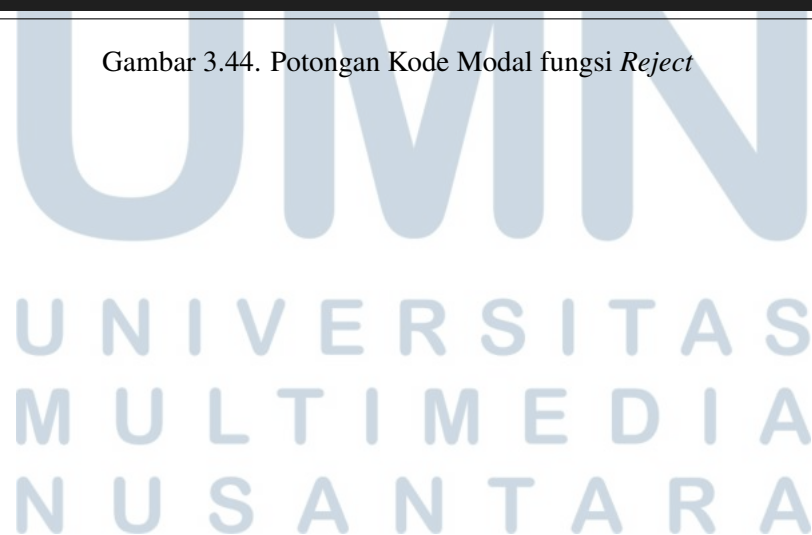
Gambar 3.43. Potongan Kode fungsi *Reject* pada Halaman HTML

```

<!-- Modal buat reject -->
<div class="modal fade" id="PopupReject" tabindex="-1" role="dialog" aria-hidden="true" style="display: none;">
  <form id="formreject">
    <input type="hidden" id="intDocumentId" value="{{item.intDocumentId}}"/>
    <div class="modal-dialog">
      <div class="modal-content">
        <div class="modal-header text-center">
          <h4 id="lov_title" class="modal-title">Reject Submitted Document</h4>
        </div>
        <div class="modal-body">
          <div class="row">
            <div class="block">
              <div class="block-content" style="width:450px">
                <h4>Alasan Reject Document</h4>
                <div class="form-group" style="width:max-content">
                  <label>Description</label>
                  <textarea id="reason" class="form-control" style="width:400px" required="" />
                </div>
              </div>
            </div>
          </div>
          <div class="modal-footer">
            <button type="submit" class="btn btn-sm btn-danger">Reject</button>
            <button id="close_lov_modal" type="button" class="btn btn-default btn-sm" data-dismiss="modal">Close</button>
          </div>
        </div>
      </div>
    </div>
  </form>
</div>

```

Gambar 3.44. Potongan Kode Modal fungsi *Reject*



```

<script>
    function RejectAllocation() {
        clsGlobal.showLoading();
        debugger;
        var intDocumentId = $('#intDocumentId').val();
        var reason = $('#reason').val();
        $.ajax({
            url: "/Transaction/materialUsage/UpdateDocumentIntoReject",
            data: { intDocumentId: intDocumentId, reason: reason },
            type: 'post',
            success: function (result) {
                debugger;
                if (result.bitSuccess == true) {
                    $('#btnRejectAllocation').hide();
                    window.location = "../../DocumentView/DocumentIndex";
                }
                else {
                    clsGlobal.hideLoading();
                    clsGlobal.getAlert(result.txtMessage);
                }
            },
            complete: function () {
            },
            error: function () {
                clsGlobal.hideLoading();
                clsGlobal.getAlert("Terjadi Kesalahan Pada Sistem!");
            }
        });
    }

    $('#formReject').submit(function () {
        RejectAllocation();
        return false;
    });

    $('#btnRejectAllocation').click(function () {
        $('#PopupReject').modal({ backdrop: 'static', keyboard: false, show: true });
    })
</script>

```

Gambar 3.45. Potongan Kode fungsi *Reject* pada Javascript

```

[HttpPost]
[FilterConfig.CheckSessionTimeout()]
[FilterConfig.CheckAuthorizationAttribute]
public ActionResult UpdateDocumentIntoReject(int intDocumentId, string reason)
{
    try
    {
        bool bitSuccess = false;

        if (intDocumentId != 0)
        {
            if (clsUpdateDocumentStatusRL.IsExisttrDocument(objDot.intDocumentId))
            {
                bitSuccess = clsUpdateDocumentStatusRL.UpdateStatusIntoReject(objDot, reason, GlobalClass.dLogin.userDot.txtUserName, GlobalClass.dLogin.userDot.txtFullName);
            }
            else
            {
                throw new Exception("gagal");
            }
        }
        return Json(clsAPI.CreateResult(bitSuccess, null, "Success Rejected!", string.Empty));
    }
    catch (Exception ex)
    {
        return Json(clsAPI.CreateError(ex, GlobalClass.dLogin.userDot.intUserID.ToString(), GlobalClass.dLogin.txtLangID.ToString(), Request.Url.ToString(), GlobalClass.dLogin.userDot), JsonRequestBehavior.AllowGet);
    }
}

```

Gambar 3.46. Potongan Kode fungsi *Reject* pada Controller

```
// Set User login.
query.txtUpdatedBy = txtUserName;
query.dtmUpdated = DateTime.Now;
query.txtStatus = "Draft";

dObjContext.trDocuments.Add(query);
dObjContext.Entry(query).State = EntityState.Modified;
dObjContext.SaveChanges();
```

Gambar 3.47. Potongan Kode fungsi *Reject* pada Buzzlogic untuk Mengubah Status

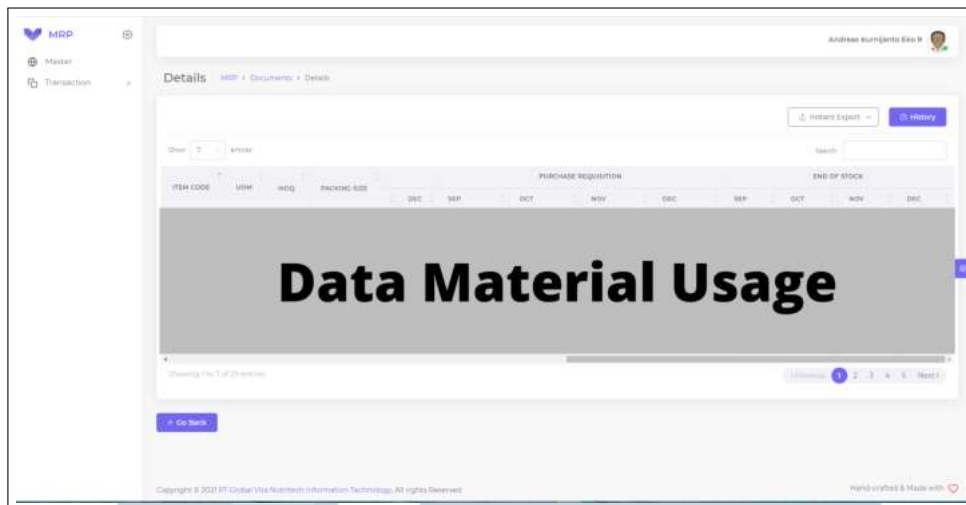
```
txtSourceName = "MRP",
intSourceId = data.intDocumentId,
txtStatus = "Reject",
txtDescription = "(" + reason + ")" + " " + "REJECTED BY " + txtFullName.ToUpper(),
bitApprove = false,
bitReject = true,
bitActive = true,
dtmCreatedDate = DateTime.Now,
txtCreatedBy = txtUserName
```

Gambar 3.48. Potongan Kode fungsi *Reject* pada Buzzlogic untuk menyimpan *Reason Reject*

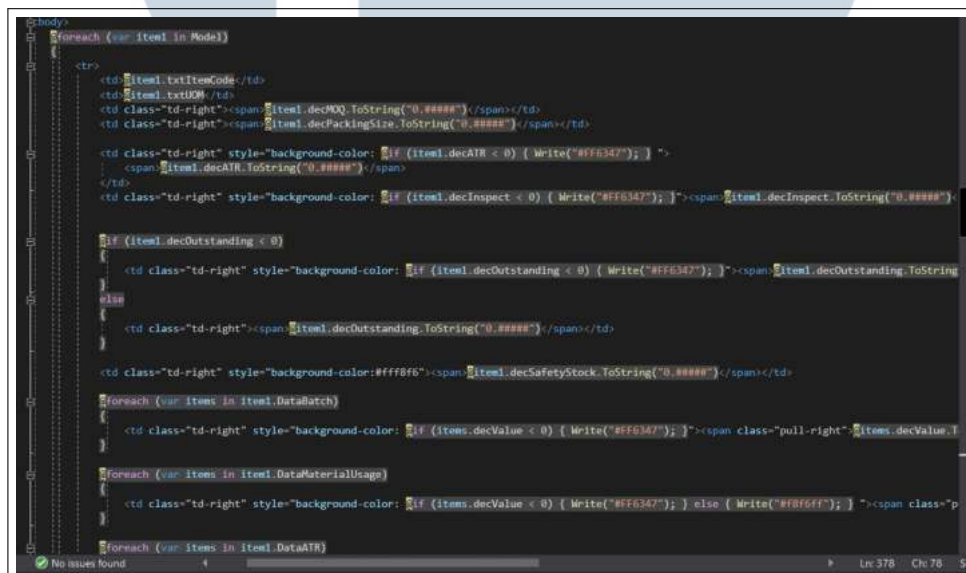
C.7 Material Usage Approved

Halaman ini merupakan bagian dari modul *Material Usage* yang menampilkan dokumen dengan status *approved*. Halaman ini menggunakan *plugin datatable* untuk menampilkan data yang ada di database. Halaman ini dapat diakses melalui modul *document view* dengan menekan *button detail* di dokumen yang berstatus *approved*.

Berikut adalah tampilan aplikasi dan juga potongan kode dari halaman ini. Adapun peraturan dari perusahaan untuk tidak menampilkan beberapa data dan juga potongan kode, maka dari itu ada beberapa gambar dan potongan kode yang di sensor ataupun tidak dimasukan. Gambar 3.49 merupakan tampilan aplikasi dari *material usage approved* dan Gambar 3.50 merupakan potongan kode dari *material usage approved*.



Gambar 3.49. Tampilan *Material Usage Approved*



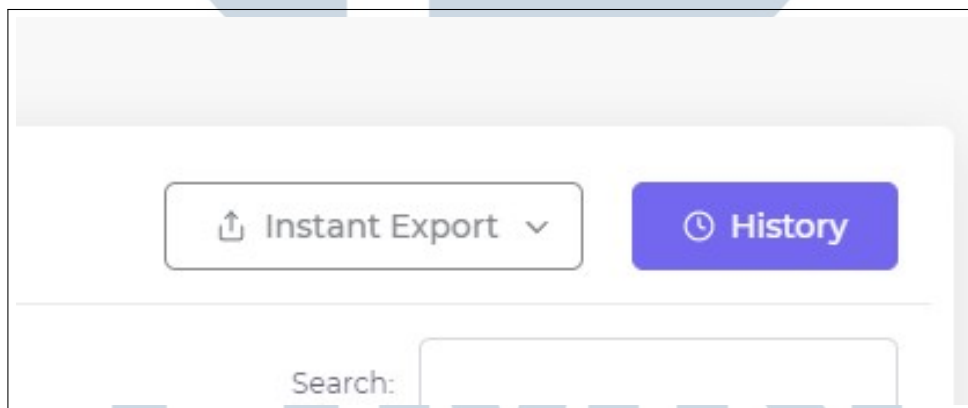
Gambar 3.50. Potongan Kode *Material Usage Approved*

C.8 Fitur Log Status

Fitur ini merupakan bagian dari modul *Material Usage*. Fitur ini bertujuan untuk menampilkan log *history* dari suatu dokumen. Fitur ini berfungsi untuk menampilkan perubahan status dan juga perubahan lainnya yang terjadi terhadap suatu dokumen. Jika ada perubahan status pada dokumen akan secara otomatis tersimpan di dalam *log status* yang

ada, termasuk pengguna yang melakukan perubahan tersebut dan juga deskripsinya.

Berikut merupakan tampilan aplikasi dan potongan kode dari fitur ini. Tampilan fitur ini akan berupa modal *pop up*. Adapun peraturan dari perusahaan untuk tidak menampilkan beberapa data dan juga potongan kode, maka dari itu ada beberapa gambar dan potongan kode yang di sensor ataupun tidak dimasukan. Gambar 3.51 merupakan tampilan *button Log Status*, Gambar 3.52 merupakan tampilan *Log Status*, Gambar 3.53 merupakan potongan Kode Modal untuk *Log Status*, Gambar 3.54 merupakan potongan kode Javascript *Log Status*, Gambar 3.55, Gambar 3.56, dan Gambar 3.57 merupakan potongan Kode untuk menambahkan *Log Status* ke dalam database.



Gambar 3.51. Tampilan *Button Log Status*



Gambar 3.52. Tampilan *Log Status*

```

<!-- Modal user details -->
<div class="modal-size-lg d-inline-block">
  <!-- Button trigger modal -->
  <!-- Modal -->
  <div class="modal fade text-left"
  id="large"
  tabindex="-1"
  role="dialog"
  aria-labelledby="myModalLabel17"
  aria-hidden="true">
    <div class="modal-dialog modal-dialog-centered modal-lg" role="document">
      <div class="modal-content">
        <div class="modal-header">
          <h4 class="modal-title" id="myModalLabel17">Details</h4>
          <button type="button" class="close" data-dismiss="modal" aria-label="Close">
            <span aria-hidden="true">&times;</span>
          </button>
        </div>
        <div class="modal-body">
          <h5>Log Status</h5>
          <table id="table_status" class="table table-striped">
            <thead>
              <tr>
                <th>Status</th>
                <th>Description</th>
                <th>CreatedBy</th>
                <th>Time</th>
              </tr>
            </thead>
            <tbody>
              <!--foreach (var items in item.DataStatus)
              <tr>
                <td>items.txtStatus</td>
                <td>items.txtDescription</td>
                <td>items.txtCreatedBy</td>
                <td>items.dtmCreateDate</td>
              </tr>
              </tbody>
            </table>
          </div>
          <div class="modal-footer">
            <button type="button" class="btn btn-primary" data-dismiss="modal"><i data-feather="x"></i> Close</button>
          </div>
        </div>
      </div>
    </div>
  </div>
</div>

```

Gambar 3.53. Potongan Kode Modal Log Status

```

{
  text: feather.icons.clock.toSvg({
    class: "mr-50 font-small-4"
  }) + "History",
  className: "btn btn-primary",
  attr: {
    "data-toggle": "modal",
    "data-target": "#large"
  },
  init: function(e, a, t) {
    $(a).removeClass("btn-secondary")
  }
}

```

Gambar 3.54. Potongan kode Javascript Log Status

```

trLogStatu dtStatus = new trLogStatu()
{
    txtSourceName = "MRP",
    intSourceId = data.intDocumentId,
    txtStatus = "Approved",
    txtDescription = "Approved BY " + txtFullName.ToUpper(),
    bitApprove = true,
    bitReject = false,
    bitActive = true,
    dtmCreatedDate = DateTime.Now,
    txtCreatedBy = txtUserName
};

```

Gambar 3.55. Potongan Kode untuk Menambahkan *Log Status*

```

txtSourceName = "MRP",
intSourceId = data.intDocumentId,
txtStatus = "Reject",
txtDescription = "(" + reason + ")" + " " + "REJECTED BY " + txtFullName.ToUpper(),
bitApprove = false,
bitReject = true,
bitActive = true,
dtmCreatedDate = DateTime.Now,
txtCreatedBy = txtUserName

```

Gambar 3.56. Potongan Kode untuk Menambahkan *Log Status*

```

{
    txtSourceName = "MRP",
    intSourceId = data.intDocumentId,
    txtStatus = "Submit",
    txtDescription = "SUBMITTED BY " + txtFullName.ToUpper(),
    bitApprove = false,
    bitReject = false,
    bitActive = true,
    dtmCreatedDate = DateTime.Now,
    txtCreatedBy = txtUserName
};

```

Gambar 3.57. Potongan Kode untuk Menambahkan *Log Status*

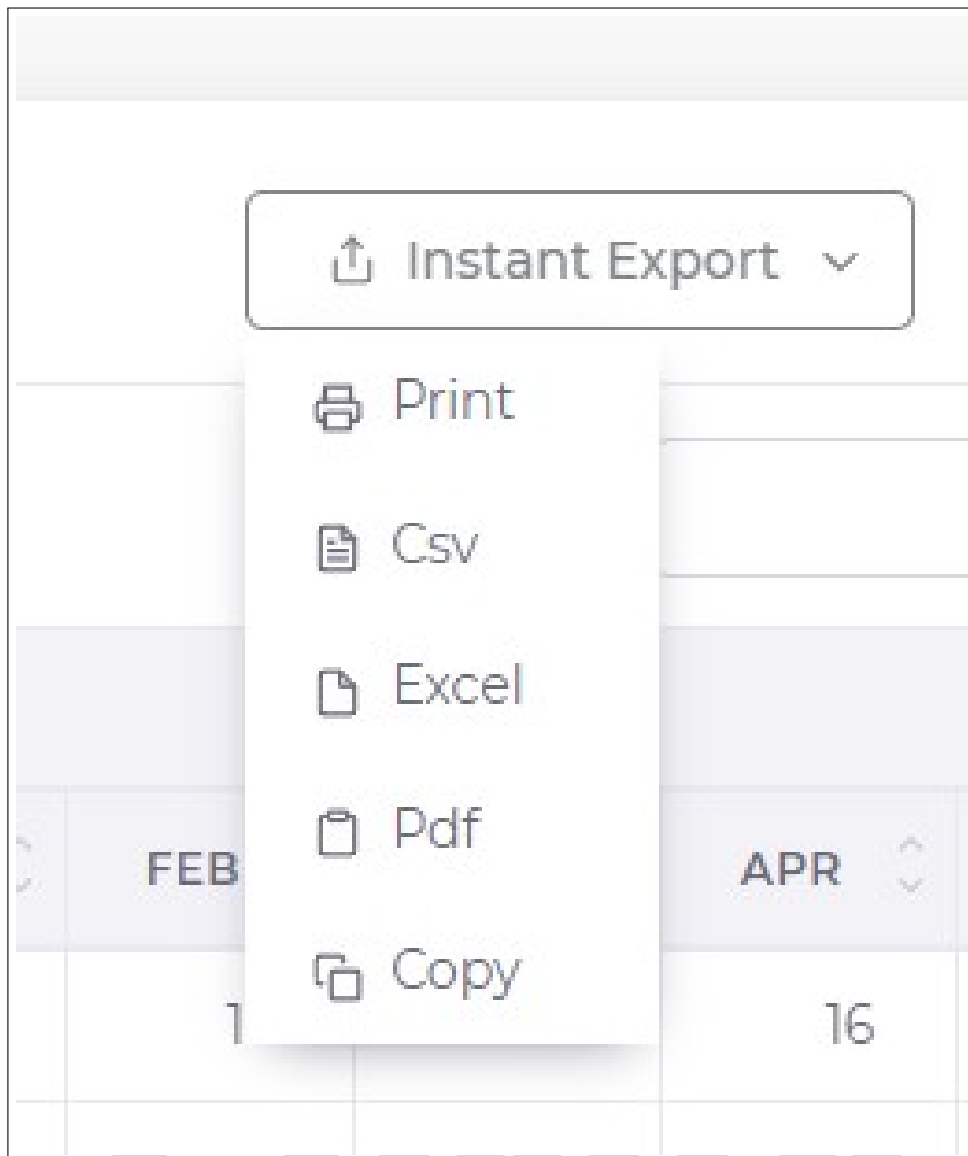
C.9 Fitur Export Excel

Fitur ini merupakan bagian dari modul *Material Usage*. *Export Excel* ini menggunakan fitur yang telah disediakan oleh *plugin datatable*. Fitur ini akan otomatis meng-export data yang ditampilkan di tabel. Pengguna juga dapat

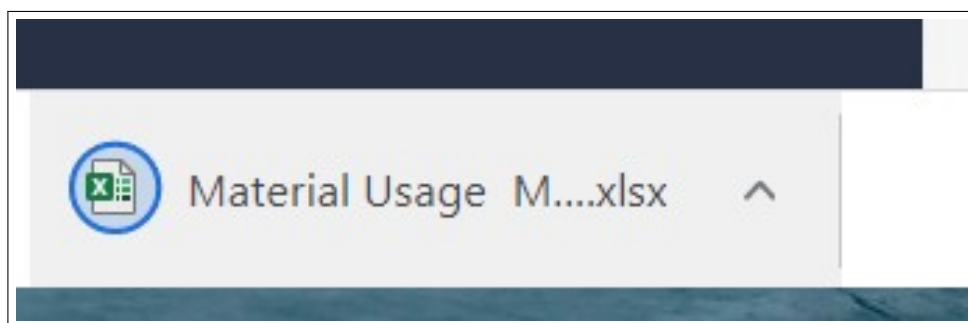
meng-*export file* dalam format lain seperti pdf, csv, dan juga dapat melakukan *print* langsung ataupun menyalin *datatable*.

Berikut merupakan tampilan aplikasi dan potongan kode dari fitur ini. Adapun peraturan dari perusahaan untuk tidak menampilkan beberapa data dan juga potongan kode, maka dari itu ada beberapa gambar dan potongan kode yang di sensor ataupun tidak dimasukan. Gambar 3.58 merupakan tampilan *button export* Excel, Gambar 3.59 merupakan tampilan *file excel* yang sudah di-*download*, Gambar 3.60 merupakan potongan kode Javascript deklarasi *datatable*, dan Gambar 3.61 merupakan potongan kode Javascript untuk fitur *export excel*.





Gambar 3.58. Tampilan *Button Export Excel*



Gambar 3.59. Tampilan *file Excel yang Sudah di-download*

```

$(document).ready(function () {
  $('#tbldata').DataTable({
    scrollX: true,
    scrollY: 400,
    scrollCollapse: true,
    paging: true,
    fixedHeader: true,
    fixedColumns: {
      leftColumns: 4,
    },
  },

```

Gambar 3.60. Potongan Kode Javascript Deklarasi Datatable

```

},
dom: '<"card-header border-bottom p-1"><"head-label"><"dt-action-buttons text-right"B><"d-flex justify-content-between align-
displayLength: 7,
lengthMenu: [7, 10, 25, 50, 75, 100],
buttons: [{
  extend: "collection",
  className: "btn btn-outline-secondary dropdown-toggle mr-2",
  text: feather.icons.share.toSvg({
    class: "font-small-4 mr-50"
  }) + "Instant Export",
  buttons: [{
    extend: "print",
    text: feather.icons.printer.toSvg({
      class: "font-small-4 mr-50"
    }) + "Print",
    className: "dropdown-item"
  }, {
    extend: "csv",
    text: feather.icons["file-text"].toSvg({
      class: "font-small-4 mr-50"
    }) + "Csv",
    className: "dropdown-item"
  }, {
    extend: "excel",
    text: feather.icons.file.toSvg({
      class: "font-small-4 mr-50"
    }) + "Excel",
    className: "dropdown-item",
    customize: function (xlsx) {
      addCustomNumberFormat(xlsx, '#,###.####');
      /*formatTargetColumn(xlsx, 'B');*/ // Excel column B
    }
  }, {
    extend: "pdf",
    text: feather.icons.clipboard.toSvg({
      class: "font-small-4 mr-50"
    }) + "Pdf",
    className: "dropdown-item"
  }, {
    extend: "copy",
    text: feather.icons.copy.toSvg({
      class: "font-small-4 mr-50"
    })
  }
]
}

```

No issues found

Gambar 3.61. Potongan Kode Javascript Fitur Export Excel

3.4 Kendala dan Solusi yang Ditemukan

Dalam proses kerja magang tentu saja terdapat berbagai kendala yang dihadapi, terlebih lagi kerja magang ini adalah pertama kali penulis terjun kedalam dunia kerja yang sebenarnya. Dari setiap kendala yang dihadapi tentu ada solusi yang harus ditemukan, berikut adalah kendala dan solusi yang di alami selama proses kerja magang.

- Saat proses kerja magang penulis menggunakan bahasa pemograman baru dan *framework* baru yaitu bahasa pemograman *C sharp* dan juga *framework* ASP.NET yang sebelumnya tidak pernah digunakan, hal tersebut menjadi kendala saat penulis mau melakukan koding. Untuk mengatasi kendala ini penulis melakukan standarisasi dan juga *training* bersama mentor selama kurang lebih 1 bulan agar penulis kedepanya bisa memahami bahasa pemograman yang baru dan juga standard yang ada di perusahaan.
- Tidak memahami *business process* yang ada. Untuk mengatasi kendala yang ada solusi yang penulis lakukan adalah mencari tahun dan juga mempelajari informasi yang bisa didapatkan dari kolega yang lebih paham terhadap *business process yang ada*.
- Menemukan kodingan yang error di dalam proses pengerjaan. Untuk mengatasi kendala tersebut, solusi yang dilakukan penulis ada mencari jawaban di internet ataupun menanyakan langsung kepada mentor.
- Diberikan tugas yaitu berupa modul ataupun fitur baru yang belum pernah dikerjakan dan dimengerti sebelumnya. Untuk mengatasi kendala tersebut, solusi yang dilakukan adalah mencari dari internet ataupun mencari dokumentasi dari proyek-proyek yang sudah ada sebelumnya.