

BAB 3 PELAKSANAAN KERJA MAGANG

3.1 Kedudukan dan Organisasi

Kerja magang dilakukan pada perusahaan PT SMART Tbk, dan ditempatkan pada divisi IT dalam tim *digital innovation*. Divisi IT bertugas untuk mengembangkan aplikasi website ataupun *dashboard* apabila dibutuhkan oleh tim lain. Tim ini juga terdiri dari 4 bagian yaitu, *support*, *product analyst*, *website developer*, dan *RPA development*. Selama kerja magang dijalankan, bimbingan diberikan Bapak Cristianto Rian Tarra dan Bapak Faris Salman Hakim, sebagai rekan *developer* dari tim *digital innovation*. Selain itu, Bapak Arif Khairiansyah selaku ketua dari tim *digital innovation* yang memberikan proyek kepada tim *digital innovation*.

3.2 Tugas yang Dilakukan

Dalam periode magang ini, tugas yang dilakukan adalah memperbaiki *back-end* dan membuat *stored procedure* untuk *website* CBS WebTool yang dimiliki sinarmas Agribusiness and Food. *Website* CBS WebTool merupakan platform internal yang dimiliki oleh SMART Tbk, yang digunakan oleh divisi lain untuk membuat *tasklist* besertakan *deadline*-nya. *Stored procedure* merupakan bagian didalam database sql yang digunakan untuk mengumpulkan data yang akan dikeluarkan ke *website* CBS WebTool. Pembuatan form pada aplikasi basys yang ada di *website* CBS Apps, dalam form tersebut akan terdapat *validation* data yang harus dibuat dan terintegrasi pada database.

3.3 Uraian Pelaksanaan Magang

Berikut merupakan tabel pekerjaan yang dilakukan setiap minggu selama pelaksanaan kerja magang pada Tabel 3.1.

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

Minggu Ke -	Pekerjaan yang dilakukan
1	Perkenalan lingkungan kerja dan mempelajari React
2	Membuat <i>small project</i> dalam React dan NodeJS
3	Mempelajari <i>package - package</i> Handsontable dan MUI
4-5	Melakukan instalasi serta mempelajari .net
6	Mempelajari <i>back-end</i> CBS WebTool (<i>TaskScheduler</i>)
7-8	Memperbarui <i>back-end TaskScheduler</i> generate KTU daily, weekly & monthly
9	Mempelajari database CBS WebTool (<i>TaskScheduler</i>)
10	Memperbarui <i>stored procedure taskscheduler</i> KTU daily & weekly untuk menampilkan <i>duedate</i>
11	Mengerjakan laporan magang
12-14	Membuat <i>frontend</i> dan <i>validation</i> untuk <i>website</i> CBS Apps halaman form bc261 aplikasi basys
15	Membuat <i>frontend</i> dan <i>validation</i> untuk <i>website</i> CBS Apps halaman form bc262 dan bc3.0 aplikasi basys

3.3.1 Pengenalan lingkungan kerja dan mempelajari React

Pada minggu pertama kerja magang yang dilakukan adalah mengenal lingkungan kerja magang dan mempelajari React. Pengenalan dimulai dari orang - orang yang akan menjadi anggota tim, pengenalan terhadap divisi - divisi lain yang ada dalam PT SMART Tbk, pengenalan terhadap pekerjaan dan proyek yang akan di kerjakan pada tim *digital innovation*, dan juga kebiasaan pada tim *digital innovation*.

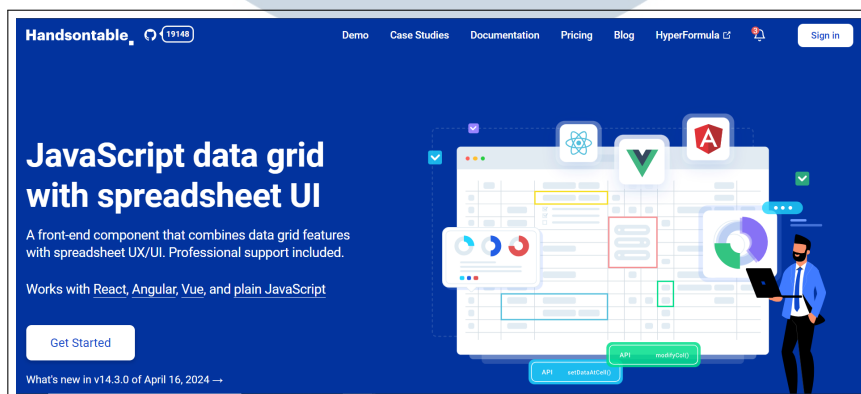
Proyek - proyek yang dikerjakan oleh tim *digital innovation* dibuat dengan menggunakan *framework* React. Oleh karena itu, diperlukan pembelajaran mengenai ReactJS dan JS Vanilla. Ketika mengalami kesulitan atau kendala, dapat menanyakan kepada rekan *developer* sehingga bisa mendapatkan bimbingan dan jawaban terkait javascript ataupun ReactJS.

3.3.2 Membuat *small project* React dan NodeJS

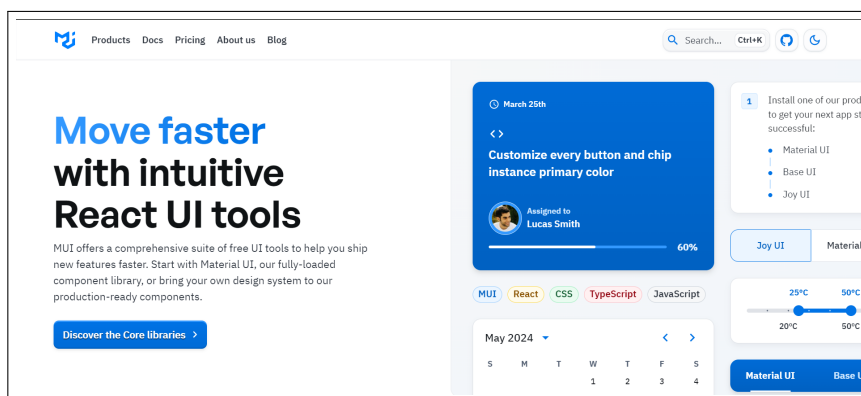
Tugas diberikan semasa magang dengan membuat *small project* seperti membuat kalkulator, *form*, dan *table page* untuk men-data *list* karyawan. Projek - projek kecil ini memiliki manfaat untuk mengenalkan kepada fitur-fitur yang akan ada pada projek - projek yang akan dikerjakan pada saat magang berlangsung.

3.3.3 Mempelajari *package - package* Handsontable dan MUI

Selama magang berlangsung dilakukan pembelajaran mengenai *package - package* yang akan digunakan pada projek pengembangan *website* CBS Apps pada aplikasi basys. Handsontable merupakan *package* yang berfungsi untuk menampilkan *table* pada sebuah *website* yang memiliki fungsi mendekati microsoft excel, seperti pada fungsi untuk meng-*copy* dan *paste* banyak *cell* dari excel ke dalam *table* tersebut. *Main page* pada handsontable dapat dilihat pada Gambar 3.1. Pada Gambar 3.2 merupakan MUI atau *Material UI*, berfungsi untuk *template* dari tampilan pada *website* CBS Apps.



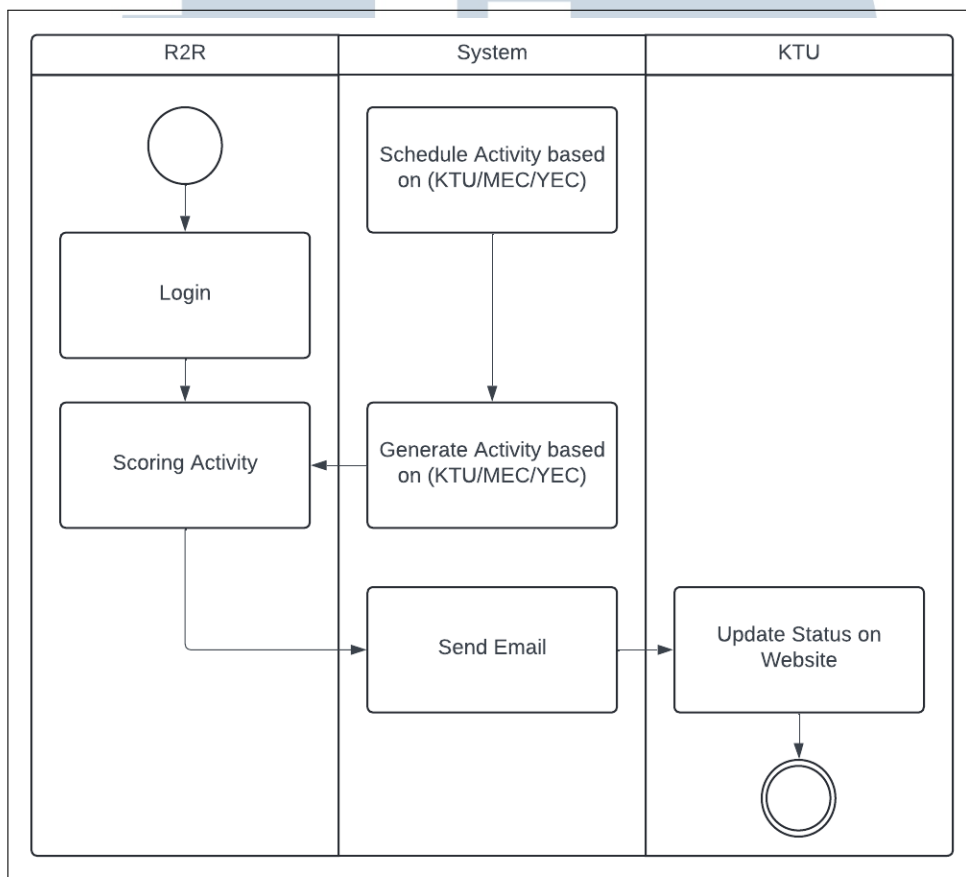
Gambar 3.1. Main Page Handsontable



Gambar 3.2. Main Page Material UI

3.3.4 CBS WebTool (*TaskScheduler*)

Projek CBS WebTool *taskscheduler* merupakan *website* yang menggunakan .NET untuk menampilkan *tasklist* yang ada pada *website* CBS WebTool untuk keperluan divisi lain pada PT SMART Tbk. Projek ini, bertujuan untuk menampilkan *tasklist* yang akan dikerjakan melalui *website*. Berikut merupakan *activity diagram* yang menggambarkan bagaimana *activity* pada CBS WebTool berjalan:



Gambar 3.3. *Activity diagram* pada CBS WebTool

Seperti pada Gambar 3.3, CBS WebTool akan digunakan oleh R2R yang merupakan *user* untuk melakukan *scoring activity*. Tujuan dari *scoring activity* ini adalah untuk memastikan apakah semua *schedule* yang telah dibuat, sudah dikerjakan dan diselesaikan oleh yang bertanggung jawab. Setelah melakukan *scoring*, *system* akan mengirimkan email melalui *website* dan akan dilakukan *update* status sesuai dengan *scoring* yang diberikan.

PCA Code	Activity Name	Transaction Type	Checklist Type	TCode	Score	Schedule Date	Due Date	Updated By	Updated Date
SMAO	Pengecekan WB error/ multikomplek	TRANSACTION	Both	ZRET - WB01		05/06/2024	05/07/2024		01/01/1900
SMAO	Pengecekan Laporan Posting BKM	REPORT	Both	Z2_RPBKM		05/06/2024	05/07/2024		01/01/1900
SMAO	Cek Smart Form yang Outstanding ataupun yg baru	TRANSACTION		SO04		05/06/2024	05/07/2024		01/01/1900
PHLM	Pengecekan WB error/ multikomplek	TRANSACTION	Both	ZRET - WB01		05/06/2024	05/07/2024		01/01/1900
LPVM	Pengecekan WB error/ multikomplek	TRANSACTION	Both	ZRET - WB01		05/06/2024	05/07/2024		01/01/1900
LPVM	Pengecekan Laporan Posting BKM	REPORT	Both	Z2_RPBKM		05/06/2024	05/07/2024		01/01/1900
LPVM	Cek Smart Form yang Outstanding ataupun yg baru	TRANSACTION		SO04		05/06/2024	05/07/2024		01/01/1900
BSWM	Pengecekan Laporan Posting BKM	REPORT	Both	Z2_RPBKM		05/06/2024	05/07/2024		01/01/1900
BSWM	Cek Smart Form yang Outstanding ataupun yg baru	TRANSACTION		SO04		05/06/2024	05/07/2024		01/01/1900
MKNM	Pengecekan WB error/ multikomplek	TRANSACTION	Both	ZRET - WB01		05/06/2024	05/07/2024		01/01/1900
MKNM	Pengecekan Laporan Posting BKM	REPORT	Both	Z2_RPBKM		05/06/2024	05/07/2024		01/01/1900

Gambar 3.4. Contoh tampilan *duedate* KTU

PCA Code	Tab	Activity Type	Activity	Report / Transaction	TCode	Related To	Due Date Related Party	Due Date	Ais / New Activity	Responsible PIC	PIC To Monitor	Due date
SSC RZR	Pre Month-End process		Edit assignment dan attach bukti posting di GL Uang Muka PPh	BOTH			0	30				1/5/2024
SSC RZR	Pre Month-End process		Jalankan YRID_GL_AUTOTEXT untuk mengisi field text yang belum diisi	TRANSACTION	YRID_GL_AUTOTEXT		0	30				1/5/2024
SSC RZR	Pre-Month End Closing		Perapahan assignment GL CTR	TRANSACTION	FAGL03 / FBL3N		0	32				1/5/2024
SSC RZR	Pre-Month End Closing		Check dan rekas GL 21001000 Piutang/Hutang Interdivision Document Curr IDR menjadi USD	TRANSACTION	FAGL03_F-02/F-05/ZEFO03		0	33				1/5/2024

Gambar 3.5. Contoh tampilan *duedate* MEC

Pada Gambar 3.4 dan Gambar 3.5 merupakan tampilan yang akan terlihat oleh *user* ketika telah terdapat data pada database terkait *task - task* yang ada besertakan *duedate*. Tugas yang diberikan pada kerja magang adalah memperbaiki hasil dari *duedate* yang tampil pada *website*, dikarenakan sebelum perbaikan *duedate* yang dihasilkan tidak sesuai dengan yang diinginkan oleh *user*. Sehingga dilakukan perubahan *code* pada *sql* dan *backend* seperti *code* yang dibawah ini. *Code* tersebut telah memperbaiki *duedate* sesuai dengan yang dibutuhkan.

```

1 CREATE PROCEDURE dbo.spInsertKTU
2 as
3 INSERT INTO TrxKTU
4     SELECT KTUActivityID ,
5           PCANo,
6           ' ' Score ,
7           NULL Remark ,

```

```

8      DueDateR2R + DATEDIFF(day , CONVERT( date , DATEADD( day ,
DueDateR2R , CONVERT( date , GETDATE( ) ) ) ) , [ dbo ].[
fnGetNextWorkingDay ](CONVERT( date , DATEADD( day , dueDateR2R ,
GETDATE( ) ) ) ) ) DueDateNumber ,
9      EmployeeNo ,
10     GETDATE() CreatedDate ,
11     ' ' UpdatedBy ,
12     ' ' UpdatedDate
13 FROM GetKTUSchedule() WHERE CONVERT( date , Nextoccurence ) =
CONVERT( date , GETDATE( ) )
14 AND RecurringType = 'Daily '
15
16 EXEC spInsertKTUWeekly

```

Kode 3.1: Code sql untuk *duedate* KTU Daily

Code 3.1 merupakan *code stored procedure* yang dibuat untuk menentukan *duedate* dari *jobdesk* yang harus diselesaikan setiap harinya. Pada bagian `[dbo].[fnGetNextWorkingDay]` merupakan function yang akan melakukan pengecekan bila *duedate* berada pada hari sabtu, minggu, dan hari libur, maka *duedate* akan bertambah 1. Apabila sebuah *task* dibuat di hari libur, maka *task* tersebut tidak akan ter-generate. Pada bagian akhir *code* terdapat *EXEC spInsertKTUWeekly* yang digunakan untuk memanggil *store procedure generate task weekly*.

```

1 CREATE PROCEDURE dbo.spInsertKTUWeekly
2 AS
3 INSERT INTO TrxKTU
4     SELECT KTUActivityID ,
5         PCANo ,
6         ' ' Score ,
7         NULL Remark ,
8         CASE
9             WHEN MONTH(GETDATE()) IN ( 4 , 7 , 10 )
10            THEN DueDateR2R
11            ELSE DueDateR2R + DATEDIFF( day , CONVERT( date , DATEADD( day ,
DueDateR2R , CONVERT( date , GETDATE( ) ) ) ) , [ dbo ].[
fnGetNextWorkingDay ](CONVERT( date , DATEADD( day , dueDateR2R ,
GETDATE( ) ) ) ) )
12            END DueDateNumber ,
13            EmployeeNo ,
14            GETDATE() CreatedDate ,
15            ' ' UpdatedBy ,
16            ' ' UpdatedDate

```

```

17 FROM GetKTUSchedule() WHERE CONVERT(date , Nextoccurence) =
    CONVERT(date , GETDATE())
18 and RecurringType='Weekly'

```

Kode 3.2: Code sql untuk *duedate* KTU Weekly

Code 3.2 merupakan *code* yang digunakan untuk mengkalkulasikan *task* yang akan berlangsung selama seminggu. Pada *code* ini terdapat sebuah *case*, apabila bulan sebuah *task* di-generate adalah bulan april, juli, dan oktober, maka *duedate* akan tetap seminggu tanpa memperdulikan hari libur. Namun, untuk bulan lainnya akan terdapat pengecekan kembali, jika *duedate* jatuh pada hari libur, maka *duedate* tersebut akan menjadi hari berikutnya.

```

1 public void GenerateMECTaskSchedule(MdKTUActPCA pcaType , MdPCA pca
  , MdKTUPCA mapping , string actType){
2   MdKTUAct task = ...;
3   if (task != null)
4     {
5       var rule =...;
6       if (rule.PCAType.Contains(";"))
7         {
8           ... ,
9         }
10      if (task != null && rule != null)
11        {
12          var lstRecurring = ... ,
13          foreach (var rec in lstRecurring)
14            {
15              ... ,
16              if (this.CurrentMonth + 1 == 13)
17                {
18                  ... ,
19                }
20
21              if (IsValidDate (...))
22                {
23                  switch (rec.RecurringType)
24                    {
25                      case "Weekly":
26                        InsertMECTask (...);
27                        break;
28                      case "Monthly":
29                        if (task.DueDateR2R.Value >= 5)
30                          {

```

```

31         ... ,
32     }
33     if (task.DueDateR2R.Value == 0)
34         // ini buat di tanggal yang sama
dengan timelinestart
35     {
36         if (task.FixedDate > 0)
37         {
38             if (task.FixedDate.Value !=
new DateTime(this.CurrentYear, this.CurrentMonth + 1, 1).
AddDays(-1).Day)
39                 {
40                     task.FixedDate = ...;
41                 }
42                 ... ,
43             }
44             else
45             {
46                 ... ,
47             }
48         }
49         else
50         {
51             DateTime createdDate = this.
CurrentDate;
52             DateTime tmpDueDate = dateEOM.
AddDays(dueDate);
53             dueDate = tmpDueDate.Subtract(
createdDate).Days;
54             tmpDueDate);
55             // smt add here
56             // QUarter
57             if ((this.CurrentMonth == 4) || (
this.CurrentMonth == 7) || (this.CurrentMonth == 10))
58                 {
59                     dueDate = dueDate + task.
DueDateR2R.Value;
60                 }
61             // Normal
62             else
63             {
64

```



```

65         dueDate = dueDate +
GetDueDateWorkingDays (dateEOM, task.DueDateR2R.Value);
66     }
67     // until here
68     }
69     break;
70     case "Yearly":
71
72     break;
73     default:
74     break;
75     }
76     if (pca != null)
77     {
78     InsertMECTask (...);
79     }
80     }
81     }
82     Save ();
83     }
84     }
85 }

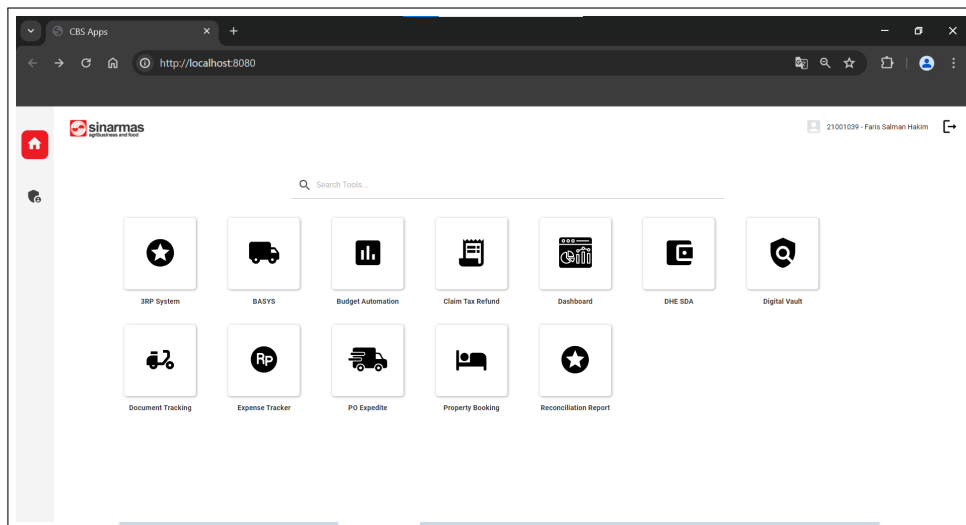
```

Kode 3.3: Code untuk mengatur *duedate* MEC

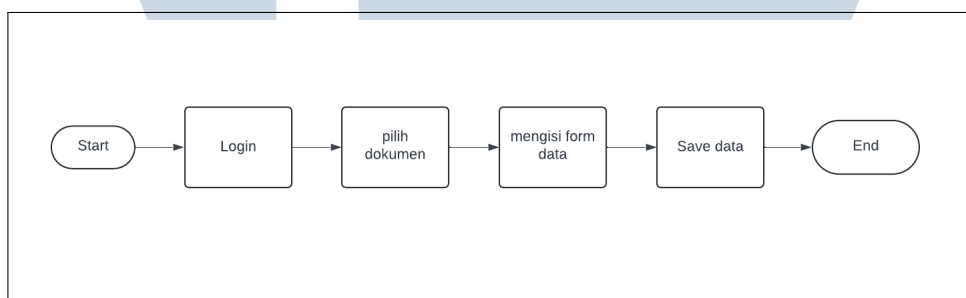
Code 3.3 ini merupakan *code* untuk mengatur *duedate* pada MEC (*Month End Closing*). *Code* ini dibuat agar seluruh *task* yang menjadi bagian *closing* dapat terlaksanakan dengan baik dan tidak terkendala.

3.3.5 CBS Apps Aplikasi Basys

CBS Apps merupakan *platform* yang berisikan aplikasi - aplikasi internal SMART yang digunakan untuk kebutuhan bisnis, salah satu aplikasi tersebut adalah basys. Basys merupakan sistem yang memberikan layanan kepada pengguna jasa untuk membuat dokumen pabean [8]. Pabean adalah sebuah instansi pengurus dan pengawas bea masuk (impor) maupun bea keluar (ekspor), baik melalui darat, udara, maupun laut [9]. Proyek ini bertujuan untuk melakukan re-platform terhadap aplikasi basys sehingga dapat berjalan lebih baik. Pada Gambar 3.6 merupakan halaman utama dari *website* CBS Apps yang memiliki beberapa aplikasi-aplikasi termasuk basys.



Gambar 3.6. Main Page CBS Apps



Gambar 3.7. Flow form dokumen

Pada Gambar 3.7 merupakan alur bagaimana aplikasi basys berjalan, pembuatan aplikasi dibatasi dengan mengerjakan re-platform pada *frontend*. Untuk membuka setiap form dokumen dibutuhkan *code* seperti pada Kode 3.4. *Code* tersebut merupakan router yang akan menghubungkan *component* dokumen dengan semua page form dokumen. Pada Kode 3.5 merupakan *code* yang digunakan untuk inialisasi data dokumen yang diperlukan sebagai *requirement* dari form.

```

1 <Suspense fallback={<div>Loading... </div>}>
2   <Switch>
3     <Route path={` ${match.url} /dokumenpabean23 `} exact={true}
4     component={DokumenPabeanBc23} />
5     <Route path={` ${match.url} /dokumenpabean261 `} exact={true}
6     component={DokumenPabeanBc261} />
7     <Route path={` ${match.url} /dokumenpabean262 `} exact={true}
8     component={DokumenPabeanBc262} />
9   </Switch>
10 </Suspense>
  
```

Kode 3.4: Code router form Basys

```

1 import { Header261, Entitas261, Dokumen261, Pengangkut261,
    KemasanPetikemas261, Transaksi261, Barang261, Jaminan261,
    Pungutan261, Pernyataan261,
2 } from "../../components";
3
4 export const components = {
5   HEADER261: {
6     component: Header261,
7   },
8   ENTITAS261: {
9     component: Entitas261,
10  },
11  ... ,
12 };
13
14 export const tabList = [
15   {
16     value: "HEADER261",
17     label: "Header",
18   },
19   {
20     value: "ENTITAS261",
21     label: "Entitas",
22   },
23   {
24     value: "DOKUMEN261",
25     label: "Dokumen",
26   },
27   ... ,
28 ];
29
30 export const FORM261 = {
31   asalData: "S",
32   asuransi: 0,
33   ... ,
34   entitas: [
35     {
36       alamatEntitas: "",
37       kodeEntitas: "3", // Pengusaha
38       ... ,
39     },
40     {
41       alamatEntitas: "",

```

```

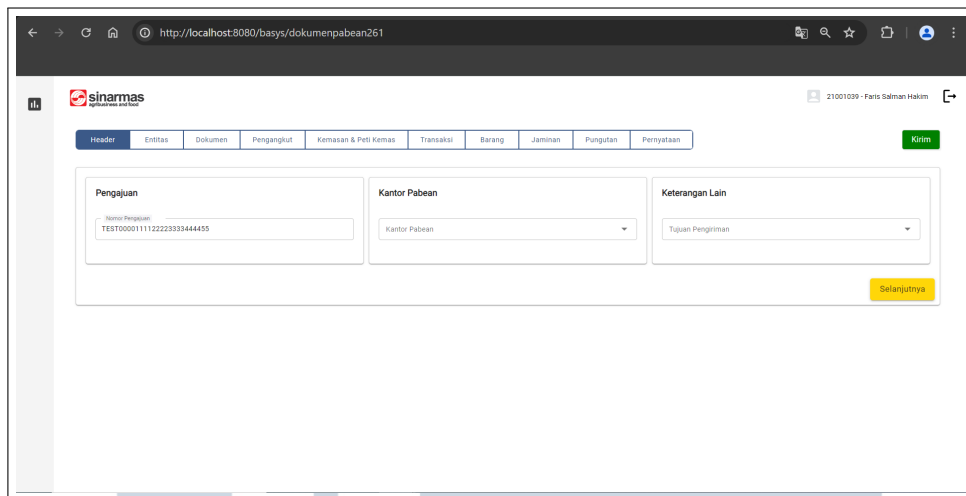
42     kodeEntitas: "7", // Pemilik Barang
43     ... ,
44 },
45 {
46     alamatEntitas: "",
47     kodeEntitas: "8", // Penerima Barang
48     ... ,
49 }
50 ],
51 dokumen: [],
52 pengangkut: [
53     {
54         kodeCaraAngkut: "",
55     }
56 ],
57 kemasan: [],
58 kontainer: [],
59 jaminan: [],
60 pungutan: [],
61 barang: [
62     {
63         ... ,
64         bahanBaku: [
65             {
66                 ... ,
67             }
68         ]
69     }
70 ],
71 };

```

Kode 3.5: *Code* inialisasi data dokumen

Pada bagian di bawah ini merupakan *code* tampilan - tampilan beserta *content* yang akan digunakan dalam membuat form untuk dokumen pada aplikasi basys. Terdapat tab Header, Entitas, Dokumen, Pengangkut, Kemasan & Peti Kemas, Transaksi, Barang, Jaminan, Pungutan, dan Pernyataan sebagai list form yang akan digunakan pada dokumenpabean261.

Beberapa contoh tampilan tab yang terdapat pada dokumenpabean261 besertakan dengan *code*-nya, sebagai berikut:



Gambar 3.8. Tampilan Header pada dokumenpabean261

```

1 import React from "react";
2 import Grid from "@mui/material/Grid";
3 import { Card, CardParent } from "../../index";
4 import {
5   ContentPengajuan ,
6   ContentKantorPabean ,
7   ContentKeteranganLain ,
8 } from "../molecules/headerContent";
9
10 function header () {
11   return (
12     <div style={{ marginTop: "2%" }}>
13       <CardParent>
14         <Grid container spacing={1}>
15           <Grid item xs={12} md={4}>
16             <Card
17               title={"Pengajuan"}
18               content={()=>ContentPengajuan ()}
19               width="100%"
20             />
21           </Grid>
22           <Grid item xs={12} md={4}>
23             <Card
24               title={"Kantor Pabean"}
25               content={()=>ContentKantorPabean ()}
26               width="100%"
27             />
28           </Grid>
29           <Grid item xs={12} md={4}>

```

```

30         <Card
31             title={"Keterangan Lain"}
32             content={()=>ContentKeteranganLain()}
33             width="100%"
34         />
35     </Grid>
36 </Grid>
37 </CardParent>
38 </div>
39 );
40 }
41 export default header;

```

Kode 3.6: Code tampilan Header

```

1 import React from "react";
2 import Grid from "@mui/material/Grid";
3 import { TextField, Autocomplete } from "../../index";
4
5 export function ContentPengajuan() {
6     return (
7         <Grid container>
8             <TextField
9                 label="Nomor Pengajuan"
10                name="nomorAju"
11                maxLength={26}
12            />
13        </Grid>
14    );
15 }
16
17 export function ContentKantorPabean() {
18     return (
19         <Grid container>
20             <Grid item xs={12}>
21                 <Autocomplete
22                     label="Kantor Pabean"
23                     name="kodeKantor"
24                 />
25             </Grid>
26        </Grid>
27    );
28 }
29

```

```

30 export function ContentKeteranganLain() {
31   return (
32     <Grid>
33       <Grid item xs={12}>
34         <Autocomplete
35           label="Tujuan Pengiriman"
36           name="kodeTujuanPengiriman"
37         />
38       </Grid>
39     </Grid>
40   );
41 }

```

Kode 3.7: Code content Header

Gambar 3.9. Tampilan Entitas pada dokumenpabean261

Gambar 3.9 merupakan tampilan untuk Entitas pada dokumenpabean261, besertakan dengan *code* yang digunakan untuk tampilan.

```

1 import React from "react";
2 import Grid from "@mui/material/Grid";
3 import { useFieldArray, useFormContext } from "react-hook-form";
4 import { Card, CardParent } from "../../index";
5 import {
6   ContentImportir,
7   ContentPemasok,
8   ContentPemilik,
9 } from "../molecules/entitasContent";
10
11 function entitas() {
12   const { control } = useFormContext();
13   const { fields } = useFieldArray({

```

```

14     control , name: "entitas"
15   });
16
17   return (
18     <div style={{ marginTop: "2%" }}>
19       <CardParent>
20         <Grid container spacing={2}>
21           <Grid item xs={12} md={4}>
22             <Card
23               key={fields [0][ "id" ]}
24               title={ "Pengusaha TPB/Pengusaha Kena Pajak" }
25               content={()=>ContentImportir () }
26               width="100%"
27             />
28           </Grid>
29           <Grid item xs={12} md={4} flexDirection="column">
30             <Card
31               key={fields [1][ "id" ]}
32               title={ "Penerima Barang/Pembeli Barang Kena Pajak /
33                 Penerima Jasa Kena Pajak" }
34               content={()=>ContentPemasok () }
35               width="100%"
36             />
37           </Grid>
38           <Grid item xs={12} md={4}>
39             <Card
40               key={fields [2][ "id" ]}
41               title={ "Pemilik Barang" }
42               content={()=>ContentPemilik () }
43               width="100%"
44             />
45           </Grid>
46         </CardParent>
47       </div>
48     );
49   }
50   export default entitas ;

```

Kode 3.8: Code tampilan Entitas

```

1   import React from "react";
2   import Grid from "@mui/material/Grid";
3   import { TextField, Autocomplete, DatePicker } from "../../.../

```



```

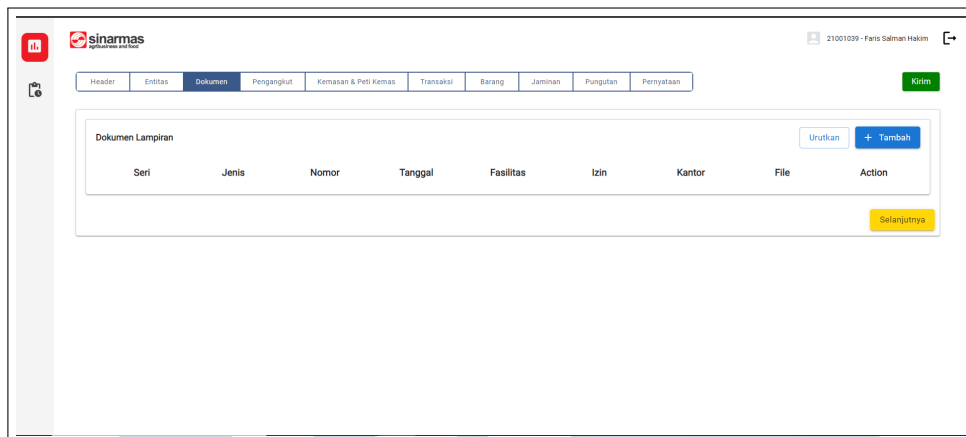
    index”];
4
5 export function ContentImportir() {
6   return (
7     <Grid container spacing={1}>
8       ... ,
9     </Grid>
10  );
11 }
12
13 export function ContentPemasok() {
14   return (
15     <Grid container>
16       ... ,
17     </Grid>
18  );
19 }
20
21 export function ContentPemilik() {
22   return (
23     <Grid container>
24       ... ,
25     </Grid>
26  );
27 }

```

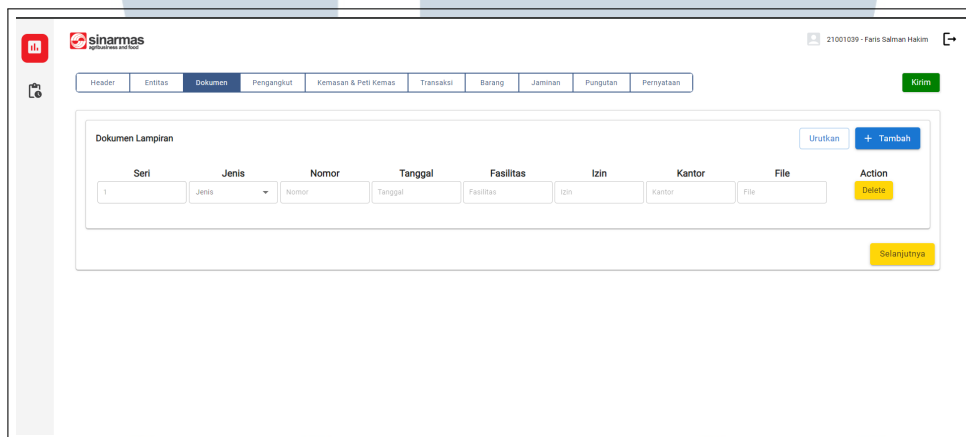
Kode 3.9: Code content Entitas

Pada bagian Dokumen, Kemasan & Peti Kemas dan Jaminan terdapat *button* tambah yang akan menampilkan 1 baris tabel kosong yang harus diisi dengan data sesuai dengan headernya (Gambar 3.10). Pada tabel tersebut terdapat *button delete* yang berfungsi untuk menghapus baris yang ditambahkan sebelumnya (Gambar 3.11). Berikut *page* sebelum dan sesudah terdapat tabel:

UNIVERSITAS
MULTIMEDIA
NUSANTARA



Gambar 3.10. Tampilan Dokumen pada dokumenpabean261



Gambar 3.11. Tampilan Dokumen pada dokumenpabean261 setelah tambah

```

1 import React, { useState } from "react";
2 import { useFieldArray, useFormContext } from "react-hook-form";
3 import Grid from "@mui/material/Grid";
4 import Button from "@mui/material/Button";
5 import AddIcon from "@mui/icons-material/Add";
6
7 import { Card, CardParent, ContentDokumen261 } from "../../../../../
  index";
8
9 function dokumen() {
10   const [tempIndex, setTempIndex] = useState(0);
11   const { control, watch } = useFormContext();
12   const methods = useFieldArray({
13     control,
14     name: "dokumen",
15   });

```

```

16  const addDocument = () => {
17      methods.append({})
18  };
19  const onRequestDelete = (idx) => {
20      methods.remove(idx);
21  };
22
23  return (
24      <div style={{ marginTop: "2%" }}>
25          <CardParent>
26              <Grid container>
27                  <Grid item xs={12}>
28                      <Card
29                          title={"Dokumen Lampiran"}
30                          content={() => ContentDokumen261({ tableData: methods
31                          .fields , onRequestDelete })}
32                          width="100%"
33                          action={
34                              <Button
35                                  variant="outlined"
36                                  style={{ marginRight: "10px" }}
37                                  sx={{ textTransform : "none" }}
38                                  >Urutkan </Button>
39                                  <Button
40                                      variant="contained"
41                                      style={{ marginRight: "8px" }}
42                                      sx={{ textTransform : "none" }}
43                                      startIcon={<AddIcon />}
44                                      onClick={addDocument}
45                                  >Tambah </Button>
46                              </>
47                          }
48                      />
49                  </Grid>
50              </Grid>
51          </CardParent>
52      </div>
53  );
54  }
55  export default dokumen;

```

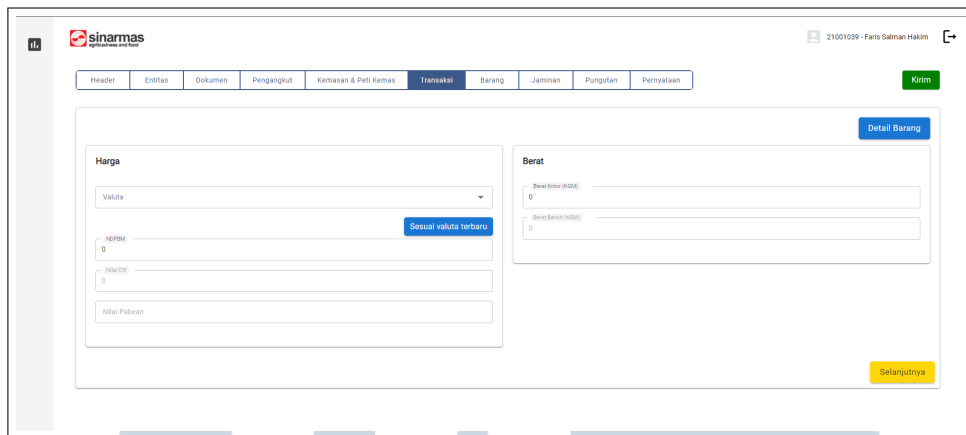
Kode 3.10: Code tampilan Dokumen

```

1 import React from "react";
2 import Table from "../../../../../common/Table";
3
4 const headers = [
5   "Seri", "Jenis", "Nomor", "Tanggal", "Fasilitas", "Izin", "
6   Kantor", "File", "Action",
7 ];
8 export function ContentDokumen261({ tableData, onRequestDelete })
9   {
10    const fields = [
11      {
12        name: "seriDokumen",
13        placeholder: "Seri",
14        type: "TextField",
15      },
16      {
17        name: "jenis",
18        placeholder: "Jenis",
19        type: "Autocomplete",
20      },
21      ... ,
22      {
23        placeholder: "Delete",
24        type: "Button",
25        functionSubmit: onRequestDelete ,
26      },
27 ];
28 return (
29   <div>
30     <Table
31       headers={headers}
32       tableData={tableData}
33       fields={fields}
34       objectKey="dokumen"
35     />
36   </div>
37 );

```

Kode 3.11: *Code content* Dokumen



Gambar 3.12. Tampilan Transaksi pada dokumenpabean261

Gambar 3.12 merupakan tampilan untuk Transaksi pada dokumenpabean261, besertakan dengan *code* yang digunakan untuk tampilan.

```

1 import React from "react";
2 import Grid from "@mui/material/Grid";
3 import Button from "@mui/material/Button";
4 import { Card, CardParent } from "../../index";
5 import {
6   ContentHarga,
7   ContentBerat,
8 } from "../molecules/transaksiContent";
9
10 function transaksi() {
11   return (
12     <div style={{ marginTop: "2%" }}>
13       <CardParent>
14         <div style={{ display: "flex", justifyContent: "end" }}>
15           <Button
16             variant="contained"
17             style={{ marginBottom: "10px" }}
18             sx={{ textTransform: "none" }}
19             >Detail Barang</Button>
20         </div>
21         <Grid container spacing={2}>
22           <Grid item xs={12} md={6}>
23             <Card
24               title="Harga"
25               content={()=>ContentHarga()}
26               width="100%"
27             />
28           </Grid>

```

```

29     <Grid container item xs={12} md={6} flexDirection="
column" rowGap={1}>
30       <Card
31         title={"Berat"}
32         content={()=>ContentBerat()}
33         width="100%"
34       />
35     </Grid>
36   </Grid>
37 </CardParent>
38 </div>
39 );
40 }
41
42 export default transaksi;

```

Kode 3.12: Code tampilan Transaksi

```

1 import React from "react";
2
3 import Grid from "@mui/material/Grid";
4 import { TextField, Autocomplete } from "../../index";
5 import { Button } from "@mui/material";
6
7 export function ContentHarga() {
8   return (
9     <Grid container columnGap={0.7}>
10      <Grid item xs={12}>
11        <Autocomplete
12          label="Valuta"
13          name="kodeValuta"
14        />
15      </Grid>
16      ... ,
17    </Grid>
18  );
19 }
20
21 export function ContentBerat() {
22   return (
23     <Grid container>
24       <Grid item xs={12}>
25         <TextField
26           label="Berat Kotor (KGM)"

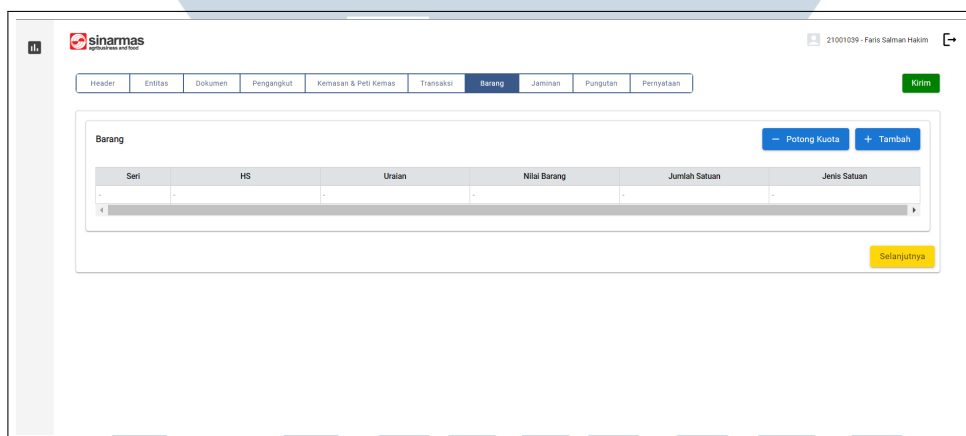
```

```

27     name="bruto"
28     type="number"
29   />
30 </Grid>
31 <Grid item xs={12}>
32   <TextField
33     label="Berat Bersih (KGM)"
34     name="netto"
35     type="number"
36     disabled={true}
37   />
38 </Grid>
39 </Grid>
40 );
41 }

```

Kode 3.13: Code content Transaksi



Gambar 3.13. Tampilan Barang pada dokumenpabean261

Pada bagian Barang terdapat *button* tambah yang akan mengarahkan kepada sebuah drawer yang berisi detail permintaan data yang diperlukan untuk ditampilkan pada tabel (Gambar 3.14). Pada *content* "Bahan Baku Impor" & "Bahan Baku Lokal" terdapat *button* yang akan mengarahkan kepada sebuah modal yang berisi keperluan yang harus diisi untuk ditampilkan pada tabel (Gambar 3.15).

```

1 import React, { useRef, useState } from 'react'
2 import Grid from "@mui/material/Grid";
3 import Button from "@mui/material/Button";
4 import AddIcon from "@mui/icons-material/Add";
5 import RemoveIcon from "@mui/icons-material/Remove";
6 import { Card, CardParent, DrawerBarang261,
  ModalDokumenFasilitas261, TableBarang261 } from "../././././././

```

```

    index";
7
8 function barang() {
9   const [isDrawerOpen, setIsDrawerOpen] = useState(false);
10  const [isModalOpen, setIsModalOpen] = useState(false);
11  const [tableData, setTableData] = useState([
12    {
13      seriBarang: "-",
14      posTarif: "-",
15      uraian: "-",
16      nilaiBarang: "-",
17      jumlahSatuan: "-",
18      kodeSatuanBarang: "-",
19    },
20  ])
21
22  const barangRef = useRef(null);
23
24  const toggleDrawer = (open) => (event) => {
25    ... ,
26    setIsDrawerOpen(open);
27  };
28
29  const submitBarang = () => {
30    alert("Submit Barang");
31  };
32
33  const submitBahanBaku = () => {
34    alert("Submit Barang");
35  };
36
37  const toggleDrawer = (open) => (event) => {
38    if (
39      event.type === "keydown" &&
40      (event.key === "Tab" || event.key === "Shift")
41    ) {
42      return;
43    }
44    setIsDrawerOpen(open);
45  };
46
47  return (
48    <div style={{ marginTop: "2%" }}>

```



```

49     <ModalDokumenFasilitas261
50         isModalDokumenOpen={isModalOpen}
51         setIsModalDokumenOpen={setIsModalOpen}
52         submit={submitBahanBaku}
53     />
54     <DrawerBarang261
55         isDrawerOpen={isDrawerOpen}
56         toggleDrawer={toggleDrawer}
57         submit={() => submitBarang()}
58     />
59     <CardParent>
60         <Grid container>
61             <Grid item xs={12}>
62                 <Card
63                     title={"Barang"}
64                     content={() => TableBarang261({ barangRef, tableData
65                 })}
66                     width="100%"
67                     action={
68                         <
69                             <Button
70                                 ... ,
71                                 >Potong Kuota</Button>
72                                 <Button
73                                     onClick={toggleDrawer(true)}
74                                     ... ,
75                                     >Tambah</Button>
76                             </>
77                         }
78                     />
79                 </Grid>
80             </Grid>
81         </CardParent>
82     </div>
83 );
84 }
85 export default barang

```

Kode 3.14: Code tampilan Barang

```

1 import React from "react";
2 import { HotTable } from "@handsontable/react";
3 import Button from "@mui/material/Button";
4 import Grid from "@mui/material/Grid";

```

```

5
6 import { Drawer, BarangDrawer261 } from "../index";
7 import { barang } from "../config/tableConfig261";
8 import { CustomizedDialogs } from "../common/
  material";
9
10 export function TableBarang261({ barangRef, tableData }) {
11   return (
12     <Grid container>
13       <Grid item xs={12}>
14         <HotTable
15           ref={barangRef}
16           settings={{
17             data: tableData,
18             ... barang.defaultConfig,
19           }}
20         />
21       </Grid>
22     </Grid>
23   );
24 }
25
26 export function DrawerBarang261({
27   isDrawerOpen,
28   toggleDrawer,
29   setIsModalOpen,
30 }) {
31   return (
32     <Drawer isDrawerOpen={isDrawerOpen} toggleDrawer={toggleDrawer}
33     >
34       <BarangDrawer261/>
35     </Drawer>
36   );
37 }
38 export function ModalDokumenFasilitas261({
39   isModalDokumenOpen,
40   setIsModalDokumenOpen,
41   submit
42 }) {
43   return (
44     <CustomizedDialogs
45     className=""

```

```

46     open={isModalDokumenOpen}
47     handleClose={() => setIsModalDokumenOpen(false)}
48     dialogtitle="Confirmation"
49     dialogcontent={
50       <div className="flex flex-col" style={{ gap: "10px" }}>
51         <label>Are you sure you want redeem the item?</label>
52       </div>
53     }
54     fullscreen={false}
55     width="sm"
56     dialogactions={
57       <div
58         className="flex"
59         style={{
60           gap: "10px",
61           justifyContent: "flex-end",
62         }}
63       >
64         ... ,
65       </div>
66     }
67   />
68 );
69 }

```

Kode 3.15: Code content Barang

Gambar 3.14. Tampilan drawer Barang pada dokumenpabean261

Gambar 3.14 merupakan tampilan drawer setelah dibuka melalui *button* tambah yang ada pada bagian barang. Tampilan ini digunakan untuk menginput data lebih detail terkait barang. Berikut merupakan *code* tampilan drawer:

```

1 import React, {useRef, useState} from "react";
2 import { useFormContext, useForm } from "react-hook-form";
3 import Button from "@mui/material/Button";
4 import Grid from "@mui/material/Grid";
5 import { joiResolver } from "@hookform/resolvers/joi";
6 import HotTable from "@handsontable/react";
7 import { Card, TextField, Autocomplete, ModalBahanImpor261,
8   ModalBahanLokal261 } from "../../../../../index";
9 import { schemaBarangBc261 } from "../../../../../common/
10  validators/bc261";
11 import { barangBahanBaku } from "../../../../../config/
12  tableConfig261";
13 import { formBarang261 } from "../../../../../common/constant/
14  bc261";
15
16 export function BarangDrawer261() {
17   const methods = useFormContext();
18   const barangRef = useRef(null);
19   const [isModalLokalOpen, setIsModalLokalOpen] = useState(false);
20   const [isModalImporOpen, setIsModalImporOpen] = useState(false);
21
22   const methodsBarang = useForm({
23     mode: "onBlur",
24     resolver: joiResolver(schemaBarangBc261),
25     defaultValues: formBarang261,
26   });
27
28   const onSubmit = (data) => {
29     methods.setValue("Barang", data);
30     console.log(data, "datacuy");
31   };
32
33   const handleError = (errors) => {
34     console.log(errors, "masuk error barang");
35   };
36
37   return (
38     <>
39     <form key={2}>
40       <div
41         style={{
42           display: "flex",
43           flexDirection: "column",

```

```

40     gap: "10px",
41     padding: "1%",
42   }}
43   >
44   <Card
45     title="Detil Barang"
46     width="100%"
47     content={() => (
48       <
49         <div style={{ position: "fixed" }}>
50           <button
51             onClick={methodsBarang.handleSubmit(onSubmit,
handleError)}
52           >
53             Simpan Test
54           </button>
55         </div>
56         <ModalBahanLokal261
57           isModalLokalOpen={isModalLokalOpen}
58           setIsModalLokalOpen={setIsModalLokalOpen}
59         />
60         <ModalBahanImpor261
61           isModalImporOpen={isModalImporOpen}
62           setIsModalImporOpen={setIsModalImporOpen}
63         />
64         <Grid container spacing={1}>
65           <Grid item xs={12} md={4}>
66             <Card
67               title={"Jenis"}
68               width="100%"
69               content={() => (
70                 <Grid container spacing={1}>
71                   <Grid item xs={12}>
72                     <TextField
73                       disabled={true}
74                       label="Seri"
75                       otherMethods={methodsBarang}
76                       name="seriBarang"
77                       type="integer"
78                     />
79                   </Grid>
80                   ... ,
81                 </Grid>

```

```

82     })
83     />
84   </Grid>
85   <Grid item xs={12} md={4}>
86     <Card
87       title={"Jumlah & Berat"}
88       width="100%"
89       content={() => (
90         ... ,
91       )}
92     />
93   </Grid>
94   <Grid item xs={12} md={4}>
95     <Card
96       title={"Harga"}
97       width="100%"
98       content={() => (
99         ... ,
100      )}
101     />
102   </Grid>
103   <Grid item xs={12} md={6}>
104     <Card
105       title={"Bahan Baku Impor"}
106       width="100%"
107       margin="10px 0 0 0"
108       content={() => (
109         ... ,
110       )}
111       action={
112         <Button
113           variant="contained"
114           size="small"
115           style={{marginRight: "8px"}}
116           sx={{textTransform : "none"}}
117           onClick={() => setIsModalImporOpen(true)}
118         />
119       }
120     />
121   </Grid>
122   <Grid item xs={12} md={6}>
123     <Card

```

```

124     title={"Bahan Baku Lokal"}
125     width="100%"
126     margin="10px 0 0 0"
127     content={() => (
128         ... ,
129     )}
130     action={
131         ... ,
132     }
133     />
134 </Grid>
135 </Grid>
136 </>
137 )}
138 ></Card>
139 </div>
140 </form>
141 </>
142 )
143 }

```

Kode 3.16: Code drawer Barang

Gambar 3.15. Tampilan modal Bahanbaku Barang pada dokumenpabean261

Gambar 3.15 merupakan tampilan yang muncul pada saat melakukan interaksi pada *button* aksi dalam tabel disesuaikan dengan keperluan bahan baku impor atau lokal. Berikut merupakan *code* tampilan bahanbaku barang:

```

1 import React from "react";
2 import Grid from "@mui/material/Grid";
3 import { useFormContext, useForm } from "react-hook-form";
4 import { joiResolver } from "@hookform/resolvers/joi";

```

```

5 import { Autocomplete, Card, DatePicker, TextField, Button } from
  ".../.../.../ index";
6 import { schemaBarangBc261 } from ".../.../.../.../ common/
  validators/bc261";
7 import { formBarang261 } from ".../.../.../.../ common/ constant/
  bc261";
8 import { CustomizedDialogs } from ".../.../.../.../.../ common/
  material";
9 import styles from ".../.../.../.../ common/basys.module.css"
10
11 export function ModalBahanImpor261({ isModalImporOpen,
  setIsModalImporOpen }) {
12   const methods = useFormContext();
13   const methodsBahanBaku = useForm({
14     mode: "onBlur",
15     resolver: joiResolver(schemaBarangBc261),
16     defaultValues: formBarang261.bahanBaku,
17   });
18
19   const onSubmit = (data) => {
20     methods.setValue("bahanBaku", data);
21     console.log(data, "datacuy");
22   };
23
24   const handleError = (errors) => {
25     console.log(errors, "masuk ke error bahanbaku");
26   };
27
28   return (
29     <CustomizedDialogs
30       className=""
31       open={isModalImporOpen}
32       handleClose={() => setIsModalImporOpen(false)}
33       dialogtitle="Bahan Baku Impor"
34       dialogcontent={
35         <>
36           <Grid container spacing={1}>
37             <Grid item xs={12} md={4}>
38               <Card
39                 title={"Dokumen Asal"}
40                 content={()=>(
41                   ... ,
42                 )}

```



```

43         width="100%"
44     />
45 </Grid>
46 <Grid item xs={12} md={4}>
47     <Card
48         title={"Jenis"}
49         content={()=>(
50             ... ,
51         )}
52         width="100%"
53     />
54 </Grid>
55 <Grid item xs={12} md={4}>
56     <Card
57         title={"Jumlah & Satuan"}
58         content={()=>(
59             ... ,
60         )}
61         width="100%"
62     />
63     <Card
64         title={"Harga"}
65         margin="10px 0 0 0"
66         content={()=>(
67             ... ,
68         )}
69         width="100%"
70     />
71 </Grid>
72 </Grid>
73 </>
74 }
75 fullscreen={false}
76 width="lg"
77 dialogactions={
78     ... ,
79 }
80 />
81 )
82 }
83
84 export function ModalBahanLokal261({isModalLokalOpen ,
    setIsModalLokalOpen }) {

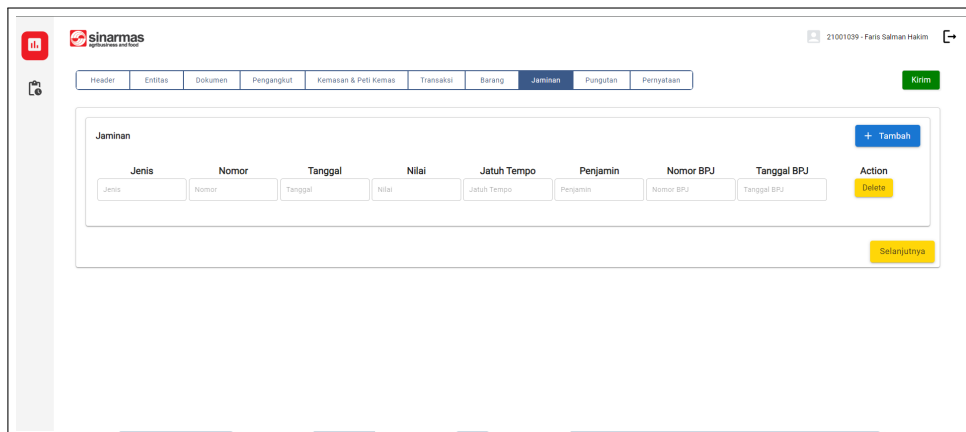
```

```

85     ... ,
86
87     return (
88         <CustomizedDialogs
89             className=""
90             open={isModalLokalOpen}
91             handleClose={() => setIsModalLokalOpen( false )}
92             dialogtitle="Bahan Baku Lokal"
93             dialogcontent={
94                 <>
95                     ... , ( sama dengan impor hanya beda permintaan data)
96                 </>
97             }
98             fullscreen={ false }
99             width="lg"
100            dialogactions={
101                <div
102                    className=" flex "
103                    style={{
104                        gap: "10px" ,
105                        justifyContent: "flex -end" ,
106                    }}
107                >
108                    <Button
109                        title="No"
110                        className={ styles . buttonRed }
111                        functionSubmit={()=>setIsModalLokalOpen( false )}
112                    />
113                    <Button
114                        title="Yes"
115                        functionSubmit={methodsBahanBaku . handleSubmit ( onSubmit
116                        , handleError )}
117                        className={ styles . buttonBlue }
118                    />
119                </div>
120            }
121        />
122    )
}

```

Kode 3.17: Code modal Bahanbaku Barang



Gambar 3.16. Tampilan Jaminan pada dokumenpabean261

Gambar 3.16 merupakan tampilan Jaminan yang ada pada dokumenpabean261, beserta *code* yang digunakan untuk tampilan:

```

1 import React, { useRef, useState } from 'react'
2 import Grid from "@mui/material/Grid";
3 import Button from "@mui/material/Button";
4 import AddIcon from "@mui/icons-material/Add";
5 import { useFieldArray, useFormContext } from 'react-hook-form';
6 import { Card, CardParent, ContentJaminan261 } from "../../../../../index";
7
8 function jaminan() {
9   const [tempIndex, setTempIndex] = useState(0);
10  const { control, watch } = useFormContext();
11  const methods = useFieldArray({
12    control, name: "jaminan", });
13  const addDocument = () => {methods.append({})};
14  const onRequestDelete = (idx) => {methods.remove(idx)};
15
16  return (
17    <div style={{ marginTop: "2%" }}>
18      <CardParent>
19        <Grid container>
20          <Grid item xs={12}>
21            <Card
22              title={"Jaminan"}
23              content={() => ContentJaminan261({ tableData: methods
24                .fields, onRequestDelete })}
25              width="100%"
26              action={
27                <Button

```

```

27         ... ,
28         onClick={addDocument}
29       >Tambah</Button>
30     }
31   />
32 </Grid>
33 </Grid>
34 </CardParent>
35 </div>
36 )
37 }
38 export default jaminan

```

Kode 3.18: *Code tampilan Jaminan*

```

1 import React from "react";
2 import { Table } from "../../../../../index";
3
4 const headers = [
5   "Jenis", "Nomor", "Tanggal", "Nilai", "Jatuh Tempo", "Penjamin",
6   "Nomor BPJ", "Tanggal BPJ", "Action",
7 ];
8 export function ContentJaminan261({ tableData, onRequestDelete }) {
9   const fields = [
10    {
11      name: "kodeJenisJaminan",
12      placeholder: "Jenis",
13      type: "TextField",
14    },
15    ... ,
16  ];
17   return (
18     <div>
19       <Table
20         headers={headers}
21         tableData={tableData}
22         fields={fields}
23         objectKey="jaminan"
24       />
25     </div>
26   );
27 }

```

Kode 3.19: *Code content Jaminan*

Gambar 3.17. Tampilan Validasi data pada dokumenpabean261

Gambar 3.17 merupakan tampilan yang digunakan untuk memvalidasi sebuah form yang dikirim, apakah tidak terdapat data yang diinput oleh *user*. Validasi digunakan untuk memastikan semua kolom form terisi dengan baik dan sesuai dengan ketentuan yang telah dibuat. Pada Kode 3.20 dibawah ini merupakan *code* untuk memvalidasi data - data yang tidak terisi ataupun terdapat ketentuan - ketentuan tersendiri. Pada *code* ini digunakan *import Joi* yang merupakan sebuah konsep untuk memvalidasi skema objek JavaScript secara otomatis sesuai dengan skema yang didefinisikan [10].

```

1 import Joi from "joi";
2
3 export const schemaFormBc261 = Joi.object({
4   nomorAju: Joi.string().pattern(/^[A-Za-z0-9]{26}$/).messages({
5     "string.pattern.base": `Nomor Pengajuan harus 26 digit`,
6     "string.empty": `Nomor Pengajuan harus diisi`,
7   }),
8   kodeKantor: Joi.string().required().messages({
9     "string.pattern.base": `Kode Kantor harus diisi`,
10  }),
11  ... ,
12  entitas: Joi.array()
13    .ordered(
14      Joi.object({
15        ... ,
16      }),
17      ... ,
18    ),
19  pengangkut: Joi.array()
20    .items(

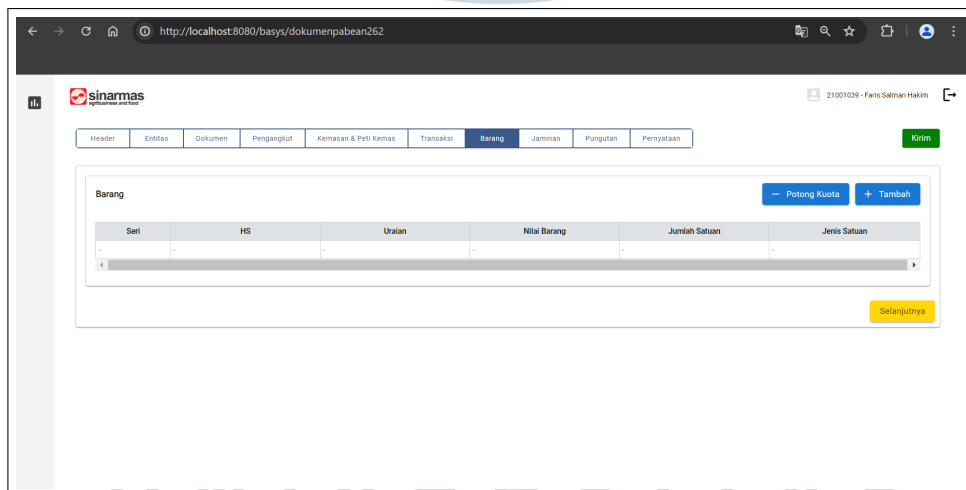
```

```

21     Joi.object({
22         kodeCaraAngkut: Joi.string().required().messages({
23             "string.empty": `Cara Pengangkutan harus diisi`,
24         }),
25     }),
26 ),
27 barang: Joi.array()
28     .items(
29         Joi.object({
30             ... ,
31             bahanBaku: Joi.array()
32                 .items(
33                     Joi.object({
34                         ... ,
35                     }),
36                 )
37             })
38         ),
39     }).options({ allowUnknown: true });

```

Kode 3.20: Code sebagai validasi data



Gambar 3.18. Tampilan dokumenpabea262

Gambar 3.18 merupakan tampilan dokumenpabea262 pada halaman basys. Pembuatan seluruh tampilan tersebut dibentuk dengan *code* yang serupa dengan dokumenpabea261, dengan beberapa *index* yang berbeda.

3.4 Kendala dan Solusi yang Ditemukan

3.4.1 Kendala yang ditemukan

1. CBS WebTool (TaskScheduler) memerlukan beberapa perbaikan pada tanggal yang digunakan untuk *tracking deadline* yang disebabkan oleh kesalahan pada penjadwalan ataupun berubahnya kebijakan berdasarkan internal memo.
2. Sistem Basys pada CBS Apps pada saat ini merupakan sistem yang tidak mengalami pembaharuan, sehingga menyebabkan penggunaan yang tidak efektif.

3.4.2 Solusi yang ditemukan

1. Melakukan perubahan pada TaskScheduler untuk membuat *deadline* menjadi fleksibel, sehingga *deadline* dari *activity* yang dibuat dapat mengikuti *calendar day*, *working date*, maupun hari libur nasional.
2. Melakukan re-platforming untuk perubahan pada UI/UX menjadi lebih modern dan sederhana.

