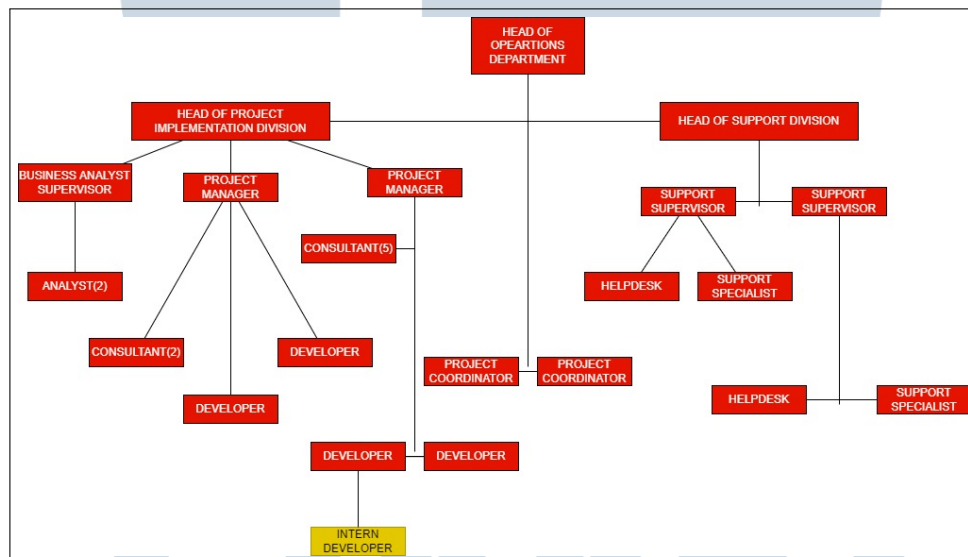


BAB 3 PELAKSANAAN KERJA MAGANG

3.1 Kedudukan dan Organisasi

Pada divisi *Software Programmer Intern* akan di bagi menjadi beberapa kelompok yaitu tim *Product* dan juga tim proyek yang dikepalai oleh seorang *Senior Software Programmer ERP*, dimana pada awalnya akan ditinjau pemahaman mengenai bahasa pemrograman Python, XML, CSS, Odoo, PostgreSQL beserta JavaScript.



Gambar 3.1. Struktur Departemen Agile pada perusahaan PT Hashmicro Solusi Indonesia

Sumber: [2]

Koordinasi pelaksanaan tugas diawasi oleh Bapak Arif Dharmawan selaku ERP Developer di PT Hashmicro Solusi Indonesia. Tugas dan koordinasi dilakukan melalui media *Skype* atau *Google Sheet* atau bisa secara langsung menggunakan media komunikasi *Gather* yang disediakan oleh *Project Manager* dan juga *Software Implementation Consultant*, lalu tugas yang sudah dikumpulkan akan dibagikan oleh *Software Programmer ERP* selaku mentor dan juga *Software Implementation Consultant* kepada setiap pemegang.

Kemudian untuk media komunikasi di dalam PT. Hashmicro Solusi Indonesia untuk pertukaran informasi, peraturan dan pengumuman sehubungan dengan

pelaksanaan magang berlangsung melalui 2 media seluler yaitu *Telegram* dan juga *WhatsApp*.

3.2 Tugas yang Dilakukan

Tugas yang dilakukan selama waktu kerja magang di PT. Hashmicro Solusi Indonesia adalah mengembangkan aplikasi *Enterprise Resource Planning* pada Modul untuk salah satu perusahaan client yang bergerak di bidang perminyakan. *Enterprise Resource Planning* yang digunakan oleh Hashmicro berbasis Odoo dengan menggunakan bahasa pemrograman Python.

Selama menjalankan tugas sebagai *Software Programmer Intern*, peserta magang diminta untuk melakukan pemantauan dan memperbaiki *bug* serta bekerja sama dengan *Software Implementation Consultant* untuk mengembangkan fitur yang dibutuhkan oleh pelanggan. Lalu terdapat juga pembuatan tampilan laporan yang dibuat menggunakan *Extensible Markup Language (XML)* yang telah di *inherited* oleh *template* aslinya seperti *Invoice* yang akan di *print*.

3.3 Uraian Pelaksanaan Magang

Untuk pelaksanaan kerja magang di PT Hashmicro Solusi Indonesia Berlangsung selama 4 bulan, dimulai dari 16 Agustus 2022 hingga 31 Desember 2022, di mana pada 3 minggu pertama akan diberikan pembelajaran mengenai sistem ERP dan modul modifikasi yang dimiliki oleh Hashmicro yang menggunakan Odoo versi 15. Lalu untuk waktu yang tersisa akan digunakan untuk mengerjakan tugas pada proyek yang sudah berjalan. Pelaksanaan kerja magang diuraikan pada Tabel 3.1.

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

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

Minggu Ke -	Pekerjaan yang dilakukan
1	Dikarenakan batch sebelumnya sudah mengikuti magang di Hash-micro, maka dari itu langsung di masuk kan kembali ke dalam project yang batch sebelumnya sedang dikerjakan dan melanjutkan task yang sudah ada. Dikarenakan masih task atau permintaan dari project belum ada sehingga dipersilahkan untuk mengikuti training sambil menunggu.
2	Tugas 1: Melakukan penambahan field sesuai kebutuhan consultan pada Purchase Order dan juga RFQ, Tugas 2: Melakukan modifikasi untuk <i>Printout</i> pada <i>Purchase Order</i> dan juga <i>Request For Quotation</i> , karena adanya masukan yang diterima oleh consultant dari client, Tugas 3: Dikarenakan tree pada <i>Manufacturing Order</i> , <i>Request For Quotation</i> , <i>Purchase Request</i> dan <i>Purchase Order</i> menunjukkan banyak data yang ada, sehingga perlu adanya tambahan <i>optional show</i> agar <i>client</i> dapat melakukan filtering bagian mana saja yang ingin ditampilkan. Tugas 4: Melakukan modifikasi print-out BSTB dan melakukan penambahan field yang diperlukan.
3	Tugas 5: Mengubah field yang berada di sub-task, agar mengambil data departemen. Tugas 6: Menambahkan Field pada Time Sheet yang mengambil nama nama user yang sudah tersedia. Tugas 7: Melakukan penambahan field pada vendor yaitu NIK dan juga ATTACHMENT. Tugas 8: Melakukan modifikasi untuk laporan SPB, SR dan BSTB.
4	Tugas 8: Melakukan pengubahan nama pada menu di INVENTORY dan juga DashBoard. Tugas 9: Melakukan <i>printout workload</i> , melakukan pengumpulan data dari berbagai model agar mempermudah proses <i>looping</i> , Melakukan penyesuaian pada EXCEL <i>WorkLoad</i> , dan membuat <i>PDF WorkLoad</i> untuk ditunjukan sebagai adanya progress pengerjaan.
5	Tugas 10: Membuat <i>TYPE OF MR</i> muncul di <i>tree</i> pada menu <i>Material Request</i> . Tugas 11: Mengubah <i>Purchase Tender</i> menjadi <i>Template RFQ</i> pada setiap menu yang memiliki <i>purchase tender</i> . Tugas 12: Membuat <i>Approval Matrix</i> pada setiap menu yang menggunakan model stock picking. Tugas 13: Mengambil data <i>Manhours</i> untuk <i>Workorder</i> yang di ambil dari menu SPPR.

Tabel 3.2. Pekerjaan yang dilakukan tiap minggu selama pelaksanaan kerja magang (Lanjutan)

	<p>Tugas 13: Melakukan modifikasi pada excel <i>Workload</i> dan pembuatan <i>Progress Weekly</i></p>
6	<p>Tugas 14: Membuat REPORT BSTBAD untuk menu <i>Internal Transfer</i>. Tugas 15: Menyesuaikan approval matrix pada stock picking. Tugas 16: Menambahkan field preference vendor, Menambahkan field activa number pada tree form master asset. Tugas 17: Mengerjakan 3 task untuk accounting yaitu: 1. Penambahan field referensi, agar dapat langsung mengambil <i>Invoice</i> saja perlu menentukan kondisi/filter hanya pada <i>Invoice</i>. 2. Melakukan Validasi ketika <i>Attachment</i> pada menu <i>Petty Cash</i> kosong. 3. Menambahkan <i>Account Analytic</i> pada master <i>Customer</i>.</p>
7	<p>Tugas 18: Membuat setiap field pada <i>Purchase Order Readonly</i> pada saat state di luar dari draft dan juga menghilangkan field yang <i>double</i> (Keluar beberapa kali). Tugas 19: Membuat button untuk <i>report BSTBAD</i> langsung muncul di dalam <i>button action print</i>, karena <i>client</i> merasa kesulitan jika adanya <i>state</i> tertentu untuk melakukan akses pada button BSTBAD. Tugas 20: Menambahkan 2 field baru pada <i>Warehouse</i> yaitu <i>Internal Analytic Account</i> dan <i>Internal Analytic Category</i>, lalu melakukan pengecekan pekerjaan pada salah satu rekan kerja. Tugas 21: Melakukan pembuatan 2 report baru untuk Work order dan juga Production record, perlunya waktu dan masukan dari consultant ataupun client mengenai kedua report tersebut.</p>
8	<p>Tugas 22: Melakukan fixing error terhadap Material Consumption (Production Record). Tugas 23: Membuat Overhead Time dan juga Overhead Material pada Work Center. Tugas 24: Melakukan perubahan pada <i>Report Job Cost Sheet</i> karena tidak dapat dilakukan <i>Print</i>. Tugas 25: Dikarenakan masih adanya button yang menuju ke view Purchase Order yang lama sehingga perlu adanya pengarahannya yang baru ke view yang baru. Tugas 26: Membuat <i>report</i> baru untuk BAPPAT. Tugas 27: Pembuatan report baru yaitu SPIK. Tugas 28: Melakukan modifikasi untuk report BSTB. Tugas 29: Melakukan perubahan pada printout Work Order dan mengubahnya menjadi RKH/LKH.</p>

Tabel 3.3. Pekerjaan yang dilakukan tiap minggu selama pelaksanaan kerja magang (Lanjutan)

9	<p>Tugas 30: Memunculkan Serial Number pada tab Notebook di inventory dan juga memunculkan data tersebut pada Report BAP-PAT. Tugas 31: Membuat menu Landed Cost untuk keperluan accounting. Tugas 32: Melakukan penghapusan agar untuk field atau view yang tidak diperlukan pada Landed Cost, Membuat menu Work Order Lines Pivot view</p>
10	<p>Tugas 33: Membuat menu gain or loss. Tugas 34: Melakukan modifikasi pada report BSTB.</p>
11	<p>Tugas 35: Membuat <i>template email</i>, agar pada saat ingin melakukan pengiriman <i>email</i> adanya <i>template</i> yang dapat digunakan langsung. Tugas 36: Membuat 2 <i>field</i> baru yaitu <i>field delivery</i> yang merupakan <i>field</i> pilihan dan juga <i>field delivery date</i> pada menu <i>sale order</i>. Tugas 37: Melakukan penambahan <i>field</i> pada <i>product usage</i>, dan juga perhitungan yang sesuai berdasarkan kebutuhan <i>client</i>. Tugas 38: Melakukan perubahan atau modifikasi pada report BPB atau <i>product usage</i>. Tugas 39: Melakukan penambahan <i>field</i> pada menu <i>inquiry</i> sesuai arahan dari <i>consultant</i>, berdasarkan <i>form</i> yang disediakan oleh <i>client</i>. Tugas 40: Membuat komputasi pada <i>field</i> di <i>Job estimate</i> pada <i>tab</i> Material</p>
12	<p>Tugas 41: Menambahkan <i>field</i> baru untuk kebutuhan sppr pada menu <i>Inquiry</i>. Tugas 42: Melakukan modifikasi pada <i>form Sale order</i> dan juga melakukan komputasi agar dapat mengambil dari <i>inquiry</i>, tapi masih dilakukan perubahan. Tugas 43: Melakukan komputasi pada sppr agar dapat mengambil data dari <i>Sale Order</i> dan masih dapat dilakukan perubahan</p>
13	<p>Tugas 44: Mengubah <i>field plan date</i> agar tidak wajib diisi karena adanya kesulitan pada proyek yang sudah berjalan. Tugas 45: Melakukan simplifikasi kode berdasarkan arahan dari <i>lead programmer</i>. Tugas 46: Melakukan perubahan pada <i>report</i> BPB pada <i>Product Usage</i>, melakukan penyesuaian <i>report</i> untuk ukuran kertas pada BPB. Tugas 47: Melakukan perubahan pada sppr. Tugas 48: Melakukan penambahan fungsi untuk mendapatkan judul dari <i>project scope</i> yang digunakan pada <i>job cost sheet</i>.</p>

Tabel 3.4. Pekerjaan yang dilakukan tiap minggu selama pelaksanaan kerja magang (Lanjutan)

	<p>Tugas 49: Melakukan perubahan pada menu <i>inquiry</i> yang dituntun dan dicek langsung oleh <i>project manager</i>. Tugas 50: Melakukan pengecekan pada total <i>job cost sheet</i> kenapa tidak menambahkan total pada <i>notebook Logistic</i>. Tugas 51: Melakukan pengecekan kenapa <i>report</i> spik dan BPB tidak ditemukan.</p>
14	<p>Tugas 52: Melakukan modifikasi tambahan, untuk <i>report</i> sprr pada menu <i>manufacturing</i>. Tugas 53: Melakukan <i>setup</i> ulang untuk proyek baru yang diberikan oleh <i>consultant</i>. Tugas 54: Melakukan pengerjaan pada proyek kedua dalam menyesuaikan <i>form</i> dengan <i>field</i> yang dibutuhkan. Tugas 55: Modifikasi report untuk <i>RFQ</i> dan <i>Purchase Order</i>. Tugas 56: Melakukan penyesuaian <i>form</i> dengan <i>field</i> yang dibutuhkan untuk <i>Invoice/Vendor bill</i> pada menu <i>accounting</i>.</p>
15	<p>Tugas 57: Melakukan penambahan <i>field</i> pada <i>SO</i>, <i>MRP PLAN</i> dan juga <i>Lead</i>, dikarenakan adanya kebutuhan dari <i>client</i> dimana data yang dari <i>LEAD</i> diminta dikirimkan kepada <i>SO</i> dan juga data pada <i>SO</i> akan digunakan pada <i>MRP PLAN</i>, sehingga perlu adanya pembuatan <i>Field Compute</i> agar <i>field</i> yang digunakan dapat otomatis mengisi dan juga masih dapat dilakukan pergantian input. Tugas 58: Melakukan penambahan <i>field</i> dan juga penyesuaian bentuk laporan SPPR sesuai dengan keinginan <i>client</i> dan juga <i>Consultant</i>. Tugas 59: Membuatkan <i>field Finish Good</i> pada <i>Job Estimate</i> yang mengambil dari <i>Project Scope</i>.</p>
16	<p>Tugas 60: Mendapatkan masukan dari <i>Client</i> dan juga <i>Consultant</i> perlu adanya penambahan <i>field</i> dan juga penyesuaian bentuk laporan SPPR. Tugas 61: Melakukan <i>tracing</i> dan juga <i>fixing</i> mengenai mengapa tidak bisa melakukan <i>Import Data Excel</i> pada <i>Product</i>. Tugas 62: Sebelumnya sempat membuatkan modul HR Sementara untuk keperluan data karyawan dan juga departemennya, Dikarenakan <i>Client</i> ternyata menggunakan modul HR yang disediakan oleh Hashmicro sehingga perlu adanya pembaruan mengenai data departemen yang ada di <i>PO</i> menggunakan <i>Module HR</i> dari Hashmicro.</p>

Tabel 3.5. Pekerjaan yang dilakukan tiap minggu selama pelaksanaan kerja magang (Lanjutan)

17	Tugas 63: Melakukan pengerjaan pada Laporan atau <i>Report Purchase Order</i> dan juga <i>Request For Quotation</i> . Tugas 64: Melakukan Finishing dan menunggu masukan dari <i>Client</i> dan juga <i>Consultant</i> mengenai pengerjaan pada Laporan atau <i>Report Purchase Order</i> dan juga <i>Request For Quotation</i> .
18	Tugas 65: Melakukan <i>fixing</i> untuk <i>Printing PO</i> dan <i>RFQ</i> terjadi <i>error</i> ketika <i>timestamp</i> pada <i>Approving Matrix</i> kosong. Tugas 66: Menambahkan <i>Field</i> untuk keperluan Portal pada <i>Purchase Order</i> .
19	Tugas 67: Adanya permintaan untuk datang ke <i>Client</i> dan menjalankan ERP secara Local dan menggunakan jaringan dari <i>Client</i> , dikarenakan jika local dijalankan secara remote, pada saat permintaan untuk mendapatkan <i>Token Authentication</i> , akan memberikan <i>error</i> karena tidak masuk ke dalam jaringan <i>Client</i> , dan juga pada saat melakukan permintaan untuk mendapatkan <i>Token Authentication</i> menggunakan server yang disediakan oleh Hashmicro dan menggunakan jaringan dari <i>Client</i> , tetap tidak diberikan token tersebut karena dianggap <i>third party</i> oleh jaringan <i>Client</i> . Sehingga perlunya Datang langsung ke <i>Client</i> untuk melakukan test tersebut.
20	Tugas 68: Melakukan pembuatan dari awal Laporan Good / Material <i>Reciept Note</i> yang disesuaikan dengan kebutuhan <i>Client</i>

Salah satu client dari perusahaan Hashmicro yang bergerak dalam bidang perminyakan dan perbankan. Dimana Bapak Samuel Soeharli

3.3.1 Pengerjaan Tugas Kerja Magang

Proyek yang diberikan selama kerja magang merupakan salah satu *client* dari perusahaan Hashmicro yang bergerak dalam bidang perminyakan dan perbankan. Dimana Bapak Samuel Soeharli menjadi *Project Manager* untuk kedua proyek tersebut yang dibantu oleh Bapak Andhika sebagai *Software Implementation Consultant Intern* dan juga beberapa *Software Implementation Consultant intern*.

Pengembangan yang dilakukan untuk kedua proyek tersebut dikerjakan berdasarkan tugas yang diberikan oleh Pak Samuel, Pak Andhika dan juga *Software*

Implementation Consultant Intern. Modul ERP yang dilakukan pengembangan ada *Accounting, Purchase, Assets, Manufacturing, Construction*. Berikut akan diuraikan penjelasan mengenai salah satu tugas yang dikerjakan pada masa kerja magang:

A. Module Construction

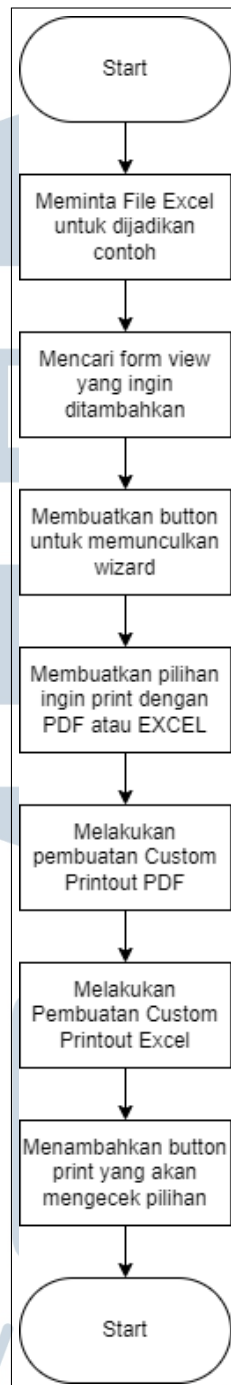
Modul ini digunakan untuk mengelola penawaran dari Vendor, mengelola berbagai jenis kontrak berbasis proyek, mengoptimasi pada pembuatan jadwal dan kolaborasi pada proyek konstruksi, melakukan analisis keuangan di beberapa lokasi konstruksi.

A.1 Pembuatan Report Workload dalam bentuk PDF dan EXCEL

Pada *task* ini diminta untuk membuat laporan WorkLoad yang disesuaikan dengan contoh *printout* yang sudah disediakan. Pada awalnya *client* hanya meminta untuk pembuatan *printout* dalam bentuk EXCEL saja, namun meminta tambahan lagi untuk membuatnya dalam bentuk PDF jika data yang ingin ditunjukkan hanya berdasarkan satu bulan saja dari suatu proyek. sehingga yang dilakukan untuk memenuhi *task* ini adalah sebagai berikut:

- Meminta contoh Excel yang akan diperlukan untuk pembuatan desain.
- Melakukan pengecekan apakah *field* atau data yang tersedia sudah memenuhi kebutuhan *printout*.

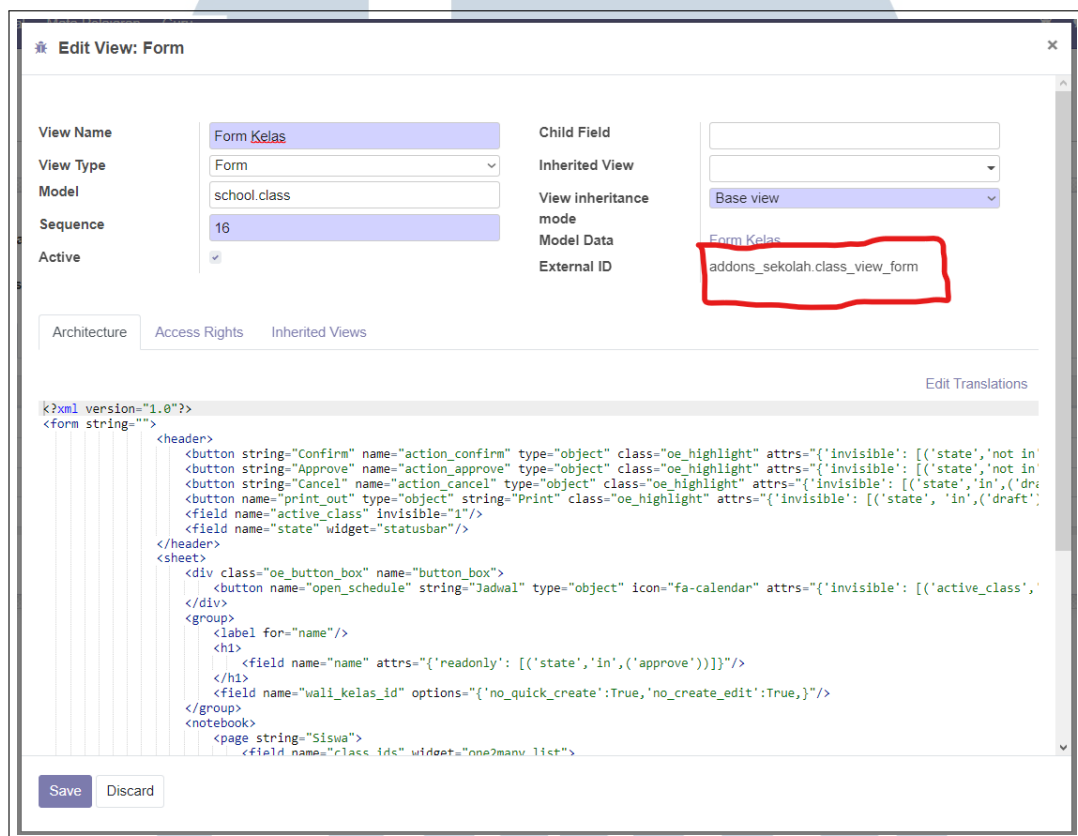
U N I V E R S I T A S
M U L T I M E D I A
N U S A N T A R A



Gambar 3.2. Arus pengerjaan pembuatan Report Workload dalam bentuk PDF dan EXCEL

Dikarenakan adanya kontrak konfidensial sehingga untuk pembuatan kode dan bentuk *printout* akan dibuat berbeda dengan aslinya sehingga dibuatkannya *custom module* sendiri untuk memberikan contoh hasil kerja. Jadi tahap yang pertama adalah menggunakan Module Custom yang telah dibuat, melakukan pengecekan

terhadap *view form* yang digunakan untuk menambahkan *button custom* yang akan memunculkan *wizard*. Jadi untuk melihat *External id* dari suatu *form* awalnya perlu mengaktifkan *Developer Mode* yang dapat diaktifkan di *setting* jika sudah buka kembali *form view* yang ingin di lakukan pengecekan lalu pilih debug selanjutnya membuka edit:View Form, dari situlah akan didapatkan *External ID* yang dapat dilihat pada gambar3.3



Gambar 3.3. Pop up yang muncul ketika membuka *Edit: View Form*

Bisa dilihat untuk *External ID* adalah *addons_sekolah.class_view_form*. Dimana *addons_sekolah* merupakan folder atau *module view* tersebut berada, dan *class_view_form* merupakan *ID* dari *View Form* nya. Setelah didapatkan *External ID* selanjutnya penambahan *button* dan juga fungsi untuk memunculkan *wizard* tersebut. Sebelum memunculkan *wizard* tersebut perlu dibuatnya *button* terlebih dahulu untuk juga *view* yang dapat dilihat pada gambar3.4.

```

<record id="class_view_form" model="ir.ui.view">
  <field name="name">Form Kelas</field>
  <field name="model">school.class</field>
  <field name="arch" type="xml">
    <form string="">
      <header>
        <button string="Confirm" name="action_confirm" type="object" class="oe_highlight" attrs="{i
        <button string="Approve" name="action_approve" type="object" class="oe_highlight" attrs="{i
        <button string="Cancel" name="action_cancel" type="object" class="oe_highlight" attrs="{inv
      </header>
      <button
        name="print_out"
        type="object"
        string="Print"
        class="oe_highlight"
        attrs="{invisible': [('state', 'in', ('draft'))]}"/>
    </form>
  </field>
</record>

```

Gambar 3.4. Penambahan *button* pada *Form View*

Dibuatnya *button* dengan *name* *print_out* dimana memiliki *string* *print* dengan *class* nya *oe_highlight* ini digunakan agar warna yang digunakan untuk *button* lebih mencolok, dan dengan atributnya *button* akan menghilang ketika *state* nya berada di dalam *draft*. Agar tidak terjadinya *error* pada saat *upgrade* modul untuk menerapkan *button* yang baru dibuat, sehingga perlu nya diberikan fungsi sederhana agar *button* tersebut dapat terlihat, fungsinya seperti pada gambar3.5

```

def print_out(self):
    pass

```

Gambar 3.5. Diberikan *Pass* agar *button* dapat muncul terlebih dahulu

Selanjutnya membuat model untuk *wizard* yang akan dibuat, dan perlu adanya 2 field yaitu *print_option* yang merupakan *field selection* dan juga *field school_class_id* yang merupakan *field many2one*. *Field Many2one* adalah *field* tipe relasi pada *Odoo* yang digunakan untuk menghubungkan *record* dari model yang satu dengan model yang lain[4]. *Field many2one* dapat diumpamakan dengan sebuah daftar kontak di sebuah telepon genggam. Setiap kontak dalam daftar tersebut merupakan *record* dari model suatu model, yang mewakili seorang individu atau

perusahaan. Setiap *record* juga memiliki *ID* unik yang digunakan untuk mengidentifikasi *record* tersebut. Hasil model yang dibuat akan ditunjukkan pada gambar3.6

```
addons_sekolah > wizard > school_class_report_wizard.py > ...
1  from odoo import api, fields, models, _
2
3
4  class SchoolClassReportWizard(models.TransientModel):
5      _name = 'school.class.report.wizard'
6      _description = 'School Class Print Options'
7
8      print_option = fields.Selection(
9          string='Type',
10         selection=[('excel', 'Excel'), ('pdf', 'PDF')],
11         default = 'excel')
12
13     school_class_id = fields.Many2one('school.class', string='School Class')
```

Gambar 3.6. Model baru untuk *School Class Wizard*

Untuk model nya dibuat dengan tipe *TransientModel*. Disebut *Transient Model* karena *records data* yang berada pada Wizards akan secara otomatis terhapus oleh *database* dalam kurun waktu tertentu[4]. Wizards dapat memberikan sesi interaktif (Kotak dialog) yang memperlihatkan formulir yang dinamis [5]. Selanjutnya adalah membuat *view form* untuk bentuk tampilan pada saat *wizard* terbuka yang dapat dilihat pada gambar3.7.

```
addons_sekolah > wizard > school_class_report_wizard_view.xml > odoo
1  <?xml version='1.0' encoding='utf-8'?>
2  <odoo>
3      <record id="wizard_view_school_class_report" model="ir.ui.view">
4          <field name="name">swizard_view_school_class_report</field>
5          <field name="model">school.class.report.wizard</field>
6          <field name="arch" type="xml">
7              <form string="">
8                  <sheet>
9                      <!-- <separator string="Print Options" colspan="4" /> -->
10                     <group>
11                         <group>
12                             <field name="print_option" widget="radio"/>
13                         </group>
14                     </group>
15                     <separator/>
16                     <footer>
17                         <button name="print_school_class" string="Print" type="object" class="oe_highlight" />
18                         <button string="Cancel" special="cancel" class="btn-secondary" />
19                     </footer>
20                 </sheet>
21             </form>
22         </field>
23     </record>
24 </odoo>
```

Gambar 3.7. View baru untuk *School Class Wizard*

Sama seperti sebelumnya untuk button print perlu dilakukannya pass ter-

lebih dahulu agar *view* dapat diproses, seperti pada gambar3.8.

```
def print_school_class(self):  
    pass
```

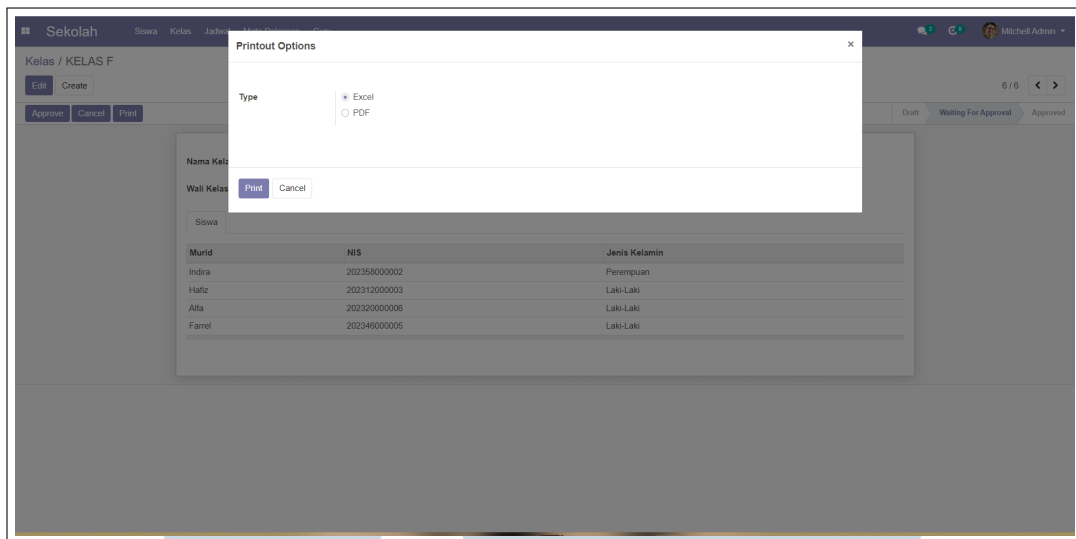
Gambar 3.8. Fungsi Pass agar button dapat digunakan *School Class Wizard*

Karena sudah dibuatnya model *School Class Wizard* dan juga sudah membuatnya *view* untuk *wizard*. Sekarang dapat digunakan untuk fungsi *print_out* yang sebelumnya dibuat menjadi *pass* akan diubah untuk membuka *wizard form view* yang akan diberi nama *Printout Options* dan dapat dilihat pada gambar3.9.

```
def print_out(self):  
    return{  
        'type' : 'ir.actions.act_window',  
        'name' : 'Printout Options',  
        'view_mode' : 'form',  
        'target' : 'new',  
        'res_model' : 'school.class.report.wizard',  
        'context' : {  
            'default_school_class_id' : self.id,  
        }  
    }
```

Gambar 3.9. Fungsi baru untuk *print_out* untuk membuka *wizard form view*

Pada *context* terdapat *default_school_class_id* yang berisikan *self.id* id disini merupakan id dari record yang sedang aktif, atau yang sekarang ini sedang dibuka, dan membuat *school_class_id* pada model *shool.class.report.wizard* secara *default* berisikan id dari model *school.class*.



Gambar 3.10. Fungsi baru untuk print_out untuk membuka wizard form view

Setelah melakukan penambahan *button* pada *form view* kelas, membuat fungsi yang akan membuka *Wizard Form View* setelah *button* tersebut ditekan, lalu pembuatan model untuk *School Class Wizard*, dan juga membuat *Wizard Form View* untuk *Wizard* yang telah dibuat, maka akan memberikan hasil seperti pada gambar 3.10. Selanjutnya membuat desain *Printout PDF* dan juga *Excel*.

UMMN
UNIVERSITAS
MULTIMEDIA
NUSANTARA

```

addons_sekolah > report > school_class_report.xml > odoo > report
4 <template id="school_class_report" name="school_class_report">
5 <t t-foreach="docs" t-as="o">
6 <t t-call="web.html_container">
7 <t t-call="addons_sekolah.school_class_external_layout">
8 <div class="page d-block" style="margin-top: 50px; display:block;">
9 <div class="col page" style="margin-top: 16px">
10 <div class="row" >
11 <table width="100%" style="font-size:15px">
12 <tr>
13 <td style="border:1px solid black" align="middle">
14 <strong>Murid</strong>
15 </td>
16 <td style="border:1px solid black" align="middle">
17 <strong>NIS</strong>
18 </td>
19 <td style="border:1px solid black" align="middle">
20 <strong>Jenis Kelamin</strong>
21 </td>
22 </tr>
23 <t t-if="o.school_class_id.class_ids">
24 <t t-foreach="o.school_class_id.class_ids" t-as="line">
25 <tr style="border:1px solid black;border-top:0px solid white">
26 <td style="border-right:1px solid black;padding: 0px 10px;" width="40%">
27 <span t-field="line.student_id"/>
28 </td>
29 <td style="border-right:1px solid black;padding: 0px 10px;" width="25%">
30 <span t-field="line.nis_murid"/>
31 </td>
32 <td style="border-right:1px solid black;padding: 0px 10px;" width="25%">
33 <span t-field="line.gender"/>
34 </td>
35 </tr>
36 </t>
37 </t>
38 <tr height="300" style="border:1px solid black;border-top:0px solid white">
39 <td style="font-size:2px;" colspan="3" >&#160;</td>
40 </tr>
41 <tr style="border:1px solid black;border-top:0px solid white" height="100">
42 <td colspan="3">
43 <table style="width:100%; font-size: 12px;page-break-inside: avoid;">
44 <tr style="page-break-inside: avoid;" >

```

Gambar 3.11. Pembuatan *Report* dengan tipe PDF



```

45 |         <td style="text-align: center; width: 100%; ">Prepared by, &#160;</td>
46 |     </tr>
47 |     <tr style="page-break-inside: avoid;" >
48 |         <td style="width: 100%; text-align: center;" >
49 |             <table width="100%" >
50 |                 <tr align="middle" width="100%" >
51 |                     <td >
52 |                         1.
53 |                     </td>
54 |                     <td >
55 |                         &#160;<span t-field="o.school_class_id.create_uid" />
56 |                     </td>
57 |                     <td >
58 |                         &#160;<span t-field="o.school_class_id.create_date"/>
59 |                     </td>
60 |                 </tr>
61 |             </table>
62 |         </td>
63 |     </tr>
64 | </table>
65 | </td>
66 | </tr>
67 | </table>
68 | </div>
69 | </div>
70 | </div>
71 | </t>
72 | </t>
73 | </template>
74 |
75 | <template id="school_class_external_layout">
76 |     <t t-if="not o" t-set="o" t-value="doc"/>
77 |     <t t-call="addons_sekolah.external_layout_school_class">
78 |         <t t-raw="0"/>
79 |     </t>
80 | </template>
81 |
82 | <template id="external_layout_school_class">
83 |     <div style="display: block;" class="header d-block">
84 |
85 |

```

Gambar 3.12. Pembuatan Report dengan tipe PDF (Lanjutan)

Pada gambar 3.11 dan gambar 3.12 merupakan badan dari *report* yang akan dibuat. Dimana untuk pengulangan data dilakukan dengan variabel *o*. Lalu untuk melakukan pengulangan pada *record* untuk model *school.class* yang memiliki field dengan tipe data one2many perlu diarahkan kepada field tersebut, pada kali ini digunakan *o.school_class_id.class_ids* dimana pengulangan tersebut akan diberikan variabel *line*. Sehingga untuk mengakses tiap *line* pada *class_ids* hanya perlu menggunakan *line.field.yang ingin diperlihatkan*, tanpa perlu menggunakan *o.school_class_id.class_ids* dari awal.

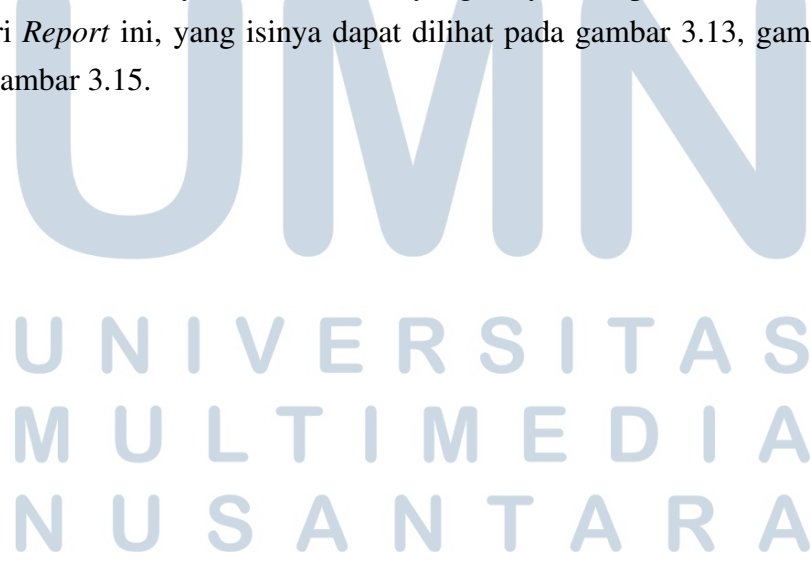

```

addons_sekolah > report > school_class_report.html > odoo > report
86 <table border="0" cellpadding="0" cellspacing="0" width="100%">
87 <tbody>
88 <tr>
89 <td >
90 <center>
91 <table border="1" cellpadding="0" cellspacing="0" style="padding:0px 20px 0px 20px; margin:0; background:#fff;" width="100%">
92 <tr>
93 <td align="left" colspan="2" t-attf-style="color:#{text_color or '#ffffff'}; padding:0 0px;" valign="top">
94 <table border="1" cellpadding="0" cellspacing="0" style="line-height: 45px;" width="100%">
95 <tr>
96 <td align="middle" width="100%"
97 style="font-weight:bold;color:black; padding:10px 0px; width:10%;border:1px solid black;font-size:23px"
98 valign="top"
99 colspan="9">
100 Data Kelas
101 </td>
102 </tr>
103 <tr height="23">
104 <td style="font-size:2px;" colspan="9" >#160;</td>
105 </tr>
106 <tr>
107 <td style="" colspan="4" width="50%">#160;
108 </td>
109 <td style="" width="20%" colspan="4">#160;</td>
110 <td style="" width="30%">
111 <table width="100%" style="color: black">
112 <tr width="100%" style="">
113 <td style="border:1px solid black;line-height:25px;" align="middle">
114 <strong>Date</strong><br/>
115 <t t-if="o.school_class_id.create_date">
116 <span t-esc="o.school_class_id.create_date.strftime('%m/%d/%Y')"/>
117 </t>
118 </td>
119 <td style="border:1px solid black;line-height:25px;" align="middle">
120 <strong>Page</strong><br/><span class="page"/>
121 </td>
122 </tr>
123 </table>
124 <tr width="100%">
125 <td colspan="2" style="border:1px solid black;line-height:25px;" align="middle">
126 <span t-field="o.school_class_id.wali_kelas_id" /> - <span t-field="o.school_class_id.name" />

```

Gambar 3.13. Pembuatan *Report* dengan tipe PDF (Lanjutan)

Pada gambar 3.11 bisa dilihat adanya pemanggilan `addons_sekolah.school_class_external_layout`, ini digunakan sebagai *container* atau menyimpan keseluruhan *report* nya. Dikarenakan pada `school_class_external_layout` dipanggilnya juga `addons_sekolah.external_layout_school_class` yang isinya merupakan *Header* dan juga *Footer* dari *Report* ini, yang isinya dapat dilihat pada gambar 3.13, gambar 3.14, dan juga gambar 3.15.



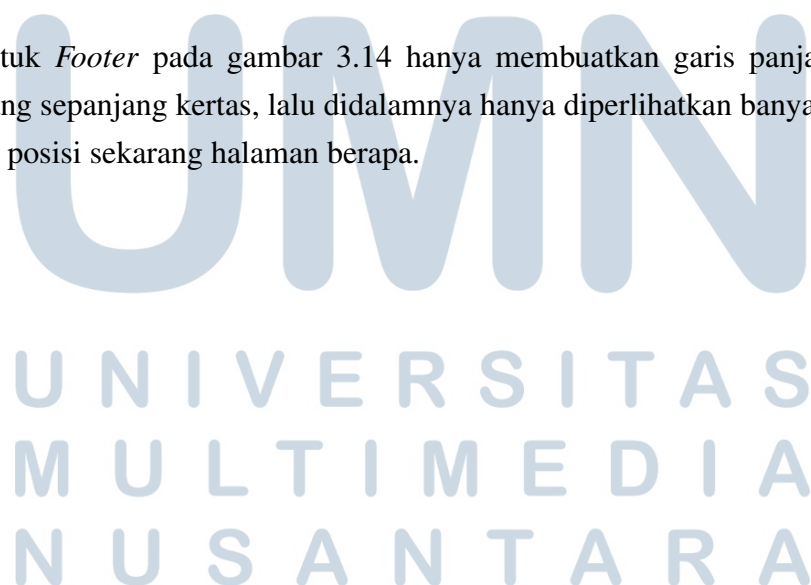
```

addons_sekolah > report > school_class_report.xml > odoo > report
127
128
129
130
131
132
133
134
135
136
137
138
139
140
141
142
143
144
145
146
147
148
149
150
151
152
153
154
155
156
157
158
159
160
161
162
163
164
165
166
167
</tr>
</table>
</td>
</tr>
</table>
</td>
</tr>
</tr>
</table>
</center>
</td>
</tr>
</tbody>
</table>
</div>
<div style="display:block;" t-att-style="display: block;"
t-attf-class="article o_report_layout_standard"
t-att-data-oe-model="o and o.school_class_id.name"
t-att-data-oe-id="o and o.school_class_id.id">
<div class="mt-3 d-block" style="display:block;">
</div>
<t t-raw="0"/>
</div>
<div t-attf-class="footer o_standard_footer">
<div class="text-center" style="border-top: 1px solid black;text-align: center;">
<table width="100%">
<tr>
<td style="text-align: left;font-size:14px" colspan="3">
</td>
</tr>
<tr>
<td style="text-align: center;" colspan="6">
<div t-if="report_type == 'pdf'" class="text-muted">
Page: <span class="page"/> / <span class="topage"/>
</div>
</td>
</tr>
</table>
</div>
</div>
</div>

```

Gambar 3.14. Pembuatan Report dengan tipe PDF (Lanjutan)

Untuk Footer pada gambar 3.14 hanya membuat garis panjang yang membentang sepanjang kertas, lalu didalamnya hanya diperlihatkan banyaknya halaman dan posisi sekarang halaman berapa.



```

addons_sekolah > report > school_class_report.xml > odoo > report
168     </template>
169
170     <record id="school_class_paperformat" model="report.paperformat">
171         <field name="name">Data Kelas</field>
172         <field name="default" eval="True" />
173         <field name="format">A4</field>
174         <field name="page_height">0</field>
175         <field name="page_width">0</field>
176         <field name="orientation">Portrait</field>
177         <field name="margin_top">55</field>
178         <field name="margin_bottom">20</field>
179         <field name="margin_left">5</field>
180         <field name="margin_right">5</field>
181         <field name="header_line" eval="False" />
182         <field name="header_spacing">50</field>
183         <field name="dpi">90</field>
184     </record>
185     <report
186         id="action_report_school_class"
187         model="school.class.report.wizard"
188         report_type="qweb-pdf"
189         string="Data Kelas"
190         name="addons_sekolah.school_class_report"
191         file="addons_sekolah.school_class_report"
192         menu="False"
193         paperformat="addons_sekolah.school_class_paperformat"
194         print_report_name="'Data Kelas - %s' % (object.school_class_id.name)"/>
195 </odoo>

```

Gambar 3.15. Pembuatan *Report* dengan tipe PDF (Lanjutan)

Pada gambar 3.15 adanya `school_class_paperformat`, ini digunakan untuk menentukan format *report* seperti pada kali ini menggunakan format kertas A4. Untuk `action_report_school_class` merupakan *action* dimana akan digunakannya *action* ini pada fungsi `print_school_class`. Untuk tipe *report* yang akan digunakan adalah `qweb-pdf` lalu untuk data yang akan saat *action* berjalan adalah `addons_sekolah.school_class_report`, lalu menggunakan *paperformat* yang sebelumnya sudah dibuat, dan membuat `print_report_name` atau nama *default* pada saat *file pdf* tersebut di *print*.

```

addons_sekolah > controller > school_class_excel_report.py > SchoolClassController > get_job_estimate_report
11 class SchoolClassController(http.Controller):
12     @http.route('/addons_sekolah/school_class_excel_report/model("school_class_report_wizard"):data', type='http', auth='user', csrf=False)
13     def get_job_estimate_report(self, data=None, **args):
14         response = request.make_response(
15             None,
16             headers=[
17                 ('Content-Type', 'application/vnd.ms-excel'),
18                 ('Content-Disposition', content_disposition(self.get_filename(data)))
19             ]
20         )
21         # Get Line in for Student Data in School Class
22         line_ids = data.school_class_id.class_ids
23         # Get Data
24         wali_kelas = data.school_class_id.wali_kelas_id.name
25         nama_kelas = data.school_class_id.name
26         user_create_name = data.school_class_id.create_uid.name
27         user_create_date = data.school_class_id.create_date
28         # Join String
29         name = wali_kelas + ' - ' + nama_kelas
30         # create workbook
31         output = io.BytesIO()
32         workbook = xlwt.Workbook(output, ('in_memory': True))
33         # Style Configuration
34         title_text_style = workbook.add_format({'font_name': 'Calibri', 'font_size': 12, 'bold': True, 'align': 'center', 'valign': 'vcenter', 'bg_color': '#FFFFFF', 'left': 1, 'bottom': 1})
35         date = workbook.add_format({'font_name': 'Calibri', 'font_size': 11, 'bold': True, 'align': 'center', 'bg_color': '#FFFFFF', 'left': 1, 'top': 1})
36         input_date = workbook.add_format({'font_name': 'Calibri', 'font_size': 11, 'align': 'center', 'bg_color': '#FFFFFF', 'left': 1, 'bottom': 1})
37         sheet_title = workbook.add_format({'font_name': 'Calibri', 'font_size': 11, 'bold': True, 'align': 'center', 'bg_color': '#FFFFFF', 'right': 1, 'top': 1})
38         input_sheet = workbook.add_format({'font_name': 'Calibri', 'font_size': 11, 'align': 'center', 'bg_color': '#FFFFFF', 'right': 1, 'bottom': 1})
39         name_style = workbook.add_format({'font_name': 'Calibri', 'font_size': 11, 'align': 'center', 'bg_color': '#FFFFFF', 'left': 1, 'right': 1, 'bottom': 1})
40         header = workbook.add_format({'font_name': 'Calibri', 'font_size': 11, 'bold': True, 'align': 'center', 'bg_color': '#FFFFFF', 'left': 1, 'bottom': 1, 'right': 1, 'top': 1})
41         lines = workbook.add_format({'font_name': 'Calibri', 'font_size': 11, 'align': 'left', 'bg_color': '#FFFFFF', 'left': 1, 'bottom': 1, 'right': 1, 'top': 1})
42         lines_nis = workbook.add_format({'font_name': 'Calibri', 'font_size': 11, 'align': 'center', 'bg_color': '#FFFFFF', 'left': 1, 'bottom': 1, 'right': 1, 'top': 1})
43         prepare_style = workbook.add_format({'font_name': 'Calibri', 'font_size': 11, 'align': 'center', 'bg_color': '#FFFFFF', 'left': 1, 'right': 1, 'top': 1})
44         number = workbook.add_format({'font_name': 'Calibri', 'font_size': 11, 'align': 'left', 'bg_color': '#FFFFFF', 'left': 1})
45         user_create_style_excel = workbook.add_format({'font_name': 'Calibri', 'font_size': 11, 'align': 'left', 'bg_color': '#FFFFFF'})
46         user_create_date_excel = workbook.add_format({'font_name': 'Calibri', 'font_size': 11, 'align': 'center', 'bg_color': '#FFFFFF', 'right': 1})
47         blank_space = workbook.add_format({'font_name': 'Calibri', 'font_size': 11, 'align': 'center', 'bg_color': '#FFFFFF', 'left': 1, 'right': 1})
48         sheet = workbook.add_worksheet(self.get_title(data))
49         sheet.set_landscape()
50         sheet.set_paper(9)
51         # Title

```

Gambar 3.16. Pembuatan *Report* dengan tipe Excel

Pada gambar 3.16 adanya *Response*. Respons tersebut akan menggunakan header yang mengindikasikan bahwa isi dari respons tersebut adalah file excel dengan tipe "application/vnd.ms-excel" dan juga akan menambahkan header "Content-Disposition" yang mengindikasikan nama file yang akan disimpan saat mengunduh respons. Nama file tersebut akan dihasilkan dari fungsi yang dipanggil dengan nama `self.get_filename(data)`.



```
addons_sekolah > controller > school_class_excel_report.py > SchoolClassController > get_filename
52 sheet.merge_range('B2:I3', self.get_title(data), title_text_style)
53
54 # Info
55 sheet.merge_range('G5:H5', "DATE", date)
56 sheet.write('I5', "SHEET", sheet_title)
57 sheet.merge_range('G6:H6', user_create_date.strftime('%m/%d/%Y'), input_date)
58 sheet.write('I6', self.get_title(data), input_sheet)
59 sheet.merge_range('G7:I7', name, name_style)
60
61 # Header
62 sheet.merge_range('B9:E9', "Murid", header)
63 sheet.merge_range('F9:G9', "NIS", header)
64 sheet.merge_range('H9:I9', "Jenis Kelamin", header)
65
66 # Line Murid
67 r1 = 9
68 c1 = 1
69 r2 = 9
70 c2 = 4
71 for line in line_ids:
72     sheet.merge_range(r1,c1,r2,c2,line.student_id.name,lines)
73     r1 += 1
74     r2 += 1
75
76 # Prepared By
77 r1 += 3
78 sheet.merge_range(r1,c1,r1,8, "Prepared By", prepare_style)
79 r1 += 1
80 sheet.write(r1,c1,"1.", number)
81 c1 += 1
82 sheet.merge_range(r1,c1,r1,5, user_create_name, user_create_style_excel)
83 c1 = 6
84 sheet.merge_range(r1,c1,r1,8, user_create_date.strftime('%m/%d/%Y'), user_create_date_excel)
85 # Black Space
86 r1 +=1
87 c1 -= 5
88 r2 += 8
89 c2 += 4
90
91 sheet.merge_range(r1,c1,r2,c2,' ', blank_space)
92
```

Gambar 3.17. Pembuatan *Report* dengan tipe Excel (Lanjutan)

Pada gambar 3.17 adanya method `merge_range` dan `write`. Kedua *method* tersebut merupakan *method* dari object *Worksheet* dari *library* *xlsxwriter*. Dimana `merge_range` digunakan untuk menggabungkan beberapa sel dalam sebuah *work-sheet* menjadi satu sel lebih besar dan `write` digunakan untuk menuliskan data ke dalam sel spesifik dalam sebuah *worksheet*.

```
addons_sekolah > controller > school_class_excel_report.py > SchoolClassController > get_title
93     # Line NIS
94     r1 = 9
95     c1 = 5
96     r2 = 9
97     c2 = 6
98     for line in line_ids:
99         sheet.merge_range(r1,c1,r2,c2,line.nis_murid,lines_nis)
100        r1 += 1
101        r2 += 1
102
103     # Line Jenis Kelamin
104     r1 = 9
105     c1 = 7
106     r2 = 9
107     c2 = 8
108     for line in line_ids:
109         if line.gender == "perempuan":
110             sheet.merge_range(r1,c1,r2,c2,line.gender.capitalize(),lines)
111         else:
112             sheet.merge_range(r1,c1,r2,c2,"Laki - Laki",lines)
113         r1 += 1
114         r2 += 1
115
116     workbook.close()
117     output.seek(0)
118     response.stream.write(output.read())
119     output.close()
120     return response
121
122     def get_title(self, data):
123         return "Data Kelas"
124
125     def get_filename(self,data):
126         current_datetime = datetime.now()
127         current_datetime_string = current_datetime.strftime("%d-%m-%Y")
128         filename = self.get_title(data) + current_datetime_string + '.xlsx'
129         return filename
```

Gambar 3.18. Pembuatan *Report* dengan tipe Excel(Lanjutan)

Pada gambar 3.18 adanya fungsi `get.title` dan juga `get.filename`. Untuk `get.title` merupakan sebuah fungsi yang menerima satu argument yaitu `data` dan mengembalikan string "Data Kelas" sebagai judul file excel yang akan dibuat. Untuk `get.filename` merupakan sebuah fungsi yang menerima satu argument yaitu `data` dan mengembalikan nama file yang akan digunakan untuk menyimpan file excel yang akan dibuat. Nama file tersebut terdiri dari hasil pemanggilan fungsi `self.get.title(data)` yang mengembalikan string "Data Kelas" diikuti dengan tanggal dan waktu saat ini dalam format "dd-mm-yyyy" diikuti dengan ekstensi file ".xlsx".

```

def print_school_class(self):
    if self.print_option == 'excel':
        return {
            'type': 'ir.actions.act_url',
            'url' : '/addons_sekolah/school_class_excel_report/%s' % (self.id),
            'target': 'new',
        }
    else:
        return self.env.ref('addons_sekolah.action_report_school_class').report_action(self)

```

Gambar 3.19. Fungsi baru untuk `print_school_class` untuk melakukan pemilihan mengenai *Output* PDF atau Excel

Selanjutnya fungsi `print_school_class` diperbarui seperti pada gambar 3.19. Fungsi tersebut memeriksa nilai dari variabel `self.print_option` dan jika nilainya sama dengan string `excel`, maka fungsi tersebut akan mengembalikan sebuah dictionary yang berisi informasi action yang akan dijalankan. Action tersebut akan mengarah ke sebuah URL yang ditentukan dan membuka URL tersebut pada tab baru di browser. Dimana akan ditangkap oleh controller yang sudah dibuat untuk menghasilkan printout Excel. Jika nilai dari `self.print_option` bukan `excel`, maka fungsi tersebut akan mengembalikan hasil dari pemanggilan method `report_action` pada sebuah object report yang merupakan referensi dari `action_report_school_class`. Sehingga Action tersebut akan mengarahkan kepada laporan yang telah ditentukan yaitu *pdf*.

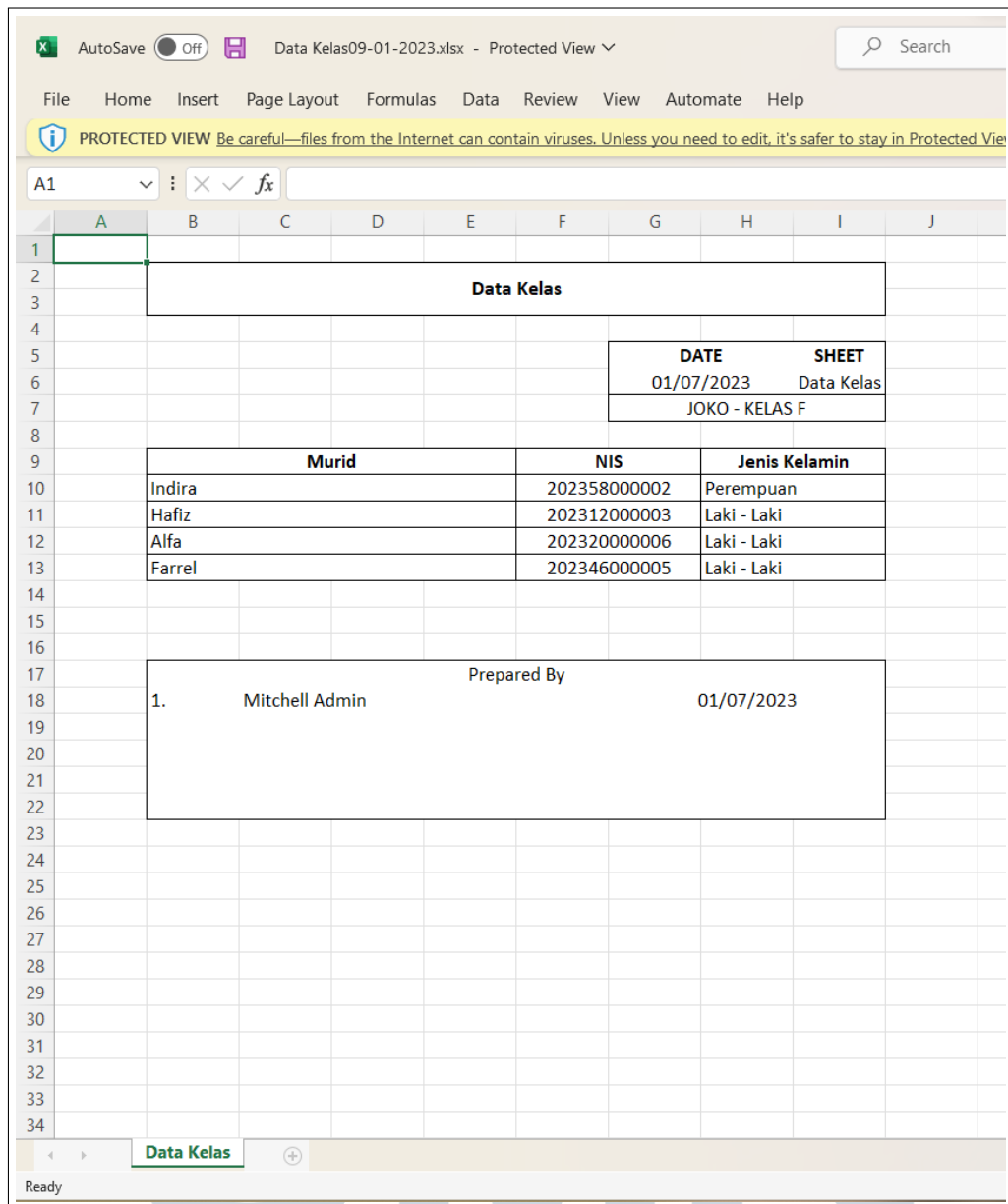
UMMN
UNIVERSITAS
MULTIMEDIA
NUSANTARA

Data Kelas								
		<table border="1" style="float: right; border-collapse: collapse;"> <tr> <td style="text-align: center;">Date</td> <td style="text-align: center;">Page</td> </tr> <tr> <td style="text-align: center;">01/07/2023</td> <td style="text-align: center;">1</td> </tr> <tr> <td colspan="2" style="text-align: center;">JOKO - KELAS F</td> </tr> </table>	Date	Page	01/07/2023	1	JOKO - KELAS F	
Date	Page							
01/07/2023	1							
JOKO - KELAS F								
Murid	NIS	Jenis Kelamin						
Indira	202358000002	Perempuan						
Hafiz	202312000003	Laki-Laki						
Alfa	202320000006	Laki-Laki						
Farrel	202346000005	Laki-Laki						
1.	Mitchell Admin	Prepared by, 01/07/2023 19:04:34						

Page: 1 / 1

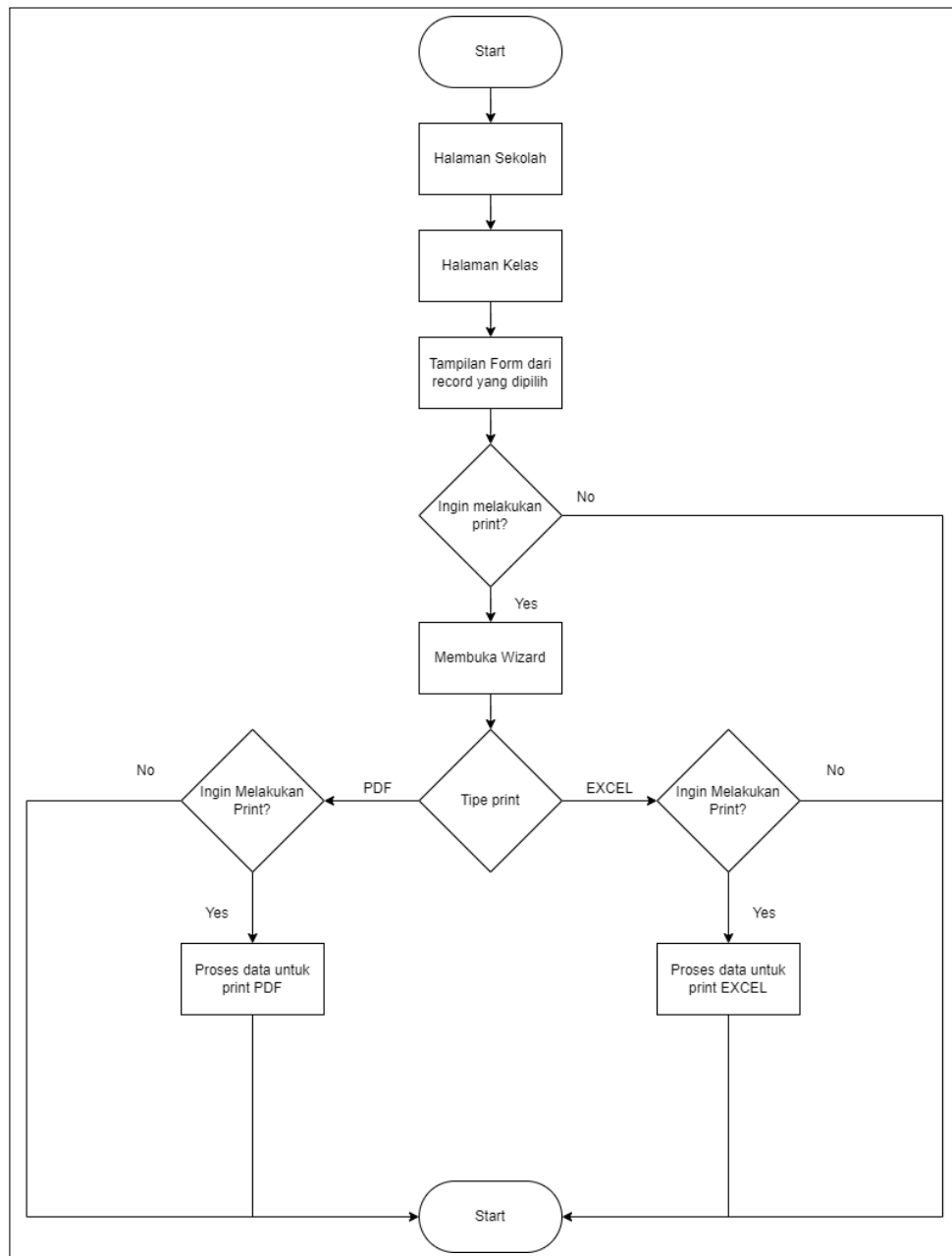
Gambar 3.20. Hasil PDF menggunakan desain yang telah dibuat

Jika opsi yang dipilih adalah selain *excel*, akan muncul seperti di gambar 3.20 dengan header nya adalah Data Kelas, lalu memperlihatkan tanggal pembuatan, nama wali kelas dan kelas, dan juga sekarang ada di halaman berapa, selanjutnya baru dilanjutkan dengan isinya yaitu nama murid yang terdaftar pada kelas tersebut.



Gambar 3.21. Hasil Excel menggunakan desain yang telah dibuat

Jika opsi yang dipilih adalah *excel*, akan muncul seperti di gambar 3.20 dengan desain yang sama seperti *PDF* dengan header nya adalah Data Kelas, lalu memperlihatkan tanggal pembuatan, nama wali kelas dan kelas, dan yang membedakan hanya jika sebelumnya menggunakan *page*, disini menggunakan *sheet*, selanjutnya baru dilanjutkan dengan isinya yaitu nama murid yang terdaftar pada kelas tersebut.



Gambar 3.22. Arus yang terjadi ketika sudah dibuatnya *report* dalam bentuk PDF dan Excel menggunakan *wizard* sebagai pengguna

Terakhir, setelah diterapkannya tahapan tersebut dalam membuat *wizard* yang akan memberikan pilihan ingin melakukan *print* dengan *excel* atau dengan *pdf* akan diperlihatkan arusnya yang terjadi ketika pengguna ingin melakukan print. Yaitu, Masuk dulu kepada halaman Sekolah, Masuk ke dalam Halaman Kelas, memilih data yang ingin dibuka *form* nya, jika tidak ingin melakukan print maka

akan langsung selesai, jika ingin melakukan print, maka akan dimunculkannya *wizard*, jika opsi *print* yang dipilih adalah pdf, selanjutnya kembali lagi ke pengguna, jika pengguna tidak ingin melakukan *print* maka akan langsung selesai, jika ingin melakukan *print* maka akan diproses terlebih dahulu datanya untuk keperluan *print* dan jika sudah berhasil melakukan download tandanya sudah selesai. Langkah yang sama juga berlaku jika opsi yang dipilih adalah *excel*.

3.4 Kendala

Berikut ini adalah kendala yang dialami saat menjalani kerja magang di PT Hashmicro Solusi Indonesia:

- Terdapat *module python* yang tidak dapat digunakan kecuali jika menggunakan *Operating System Linux*.
- Adanya Screenshot monitor yang dapat membuat saya selalu merasa untuk menggerakkan *mouse* setiap 5 menit

3.5 Solusi

Berikut beberapa solusi yang ditemukan selama menjalani kerja magang di PT. Hashmicro Solusi Indonesia:

- Melakukan *Dual Booting* jika diperlukannya menggunakan OS Linux karena jika menggunakan seperti *Virtual Machine* akan memakan banyak *Resource*.
- Bisa dengan melakukan pengecekan dalam satu minggu sudah berapa kali melakukan *commit* atau *push* pada *Repository GitLab*

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