

BAB 3 PELAKSANAAN KERJA MAGANG

3.1 Kedudukan dan Organisasi

Pelaksanaan kerja magang dilakukan pada bagian *Software Engineer* di bawah bimbingan Bapak Fauzi Efriyadi sebagai *Senior Software Engineer* di CITIS KOMPAS Gramedia. Selama melakukan kerja magang ini, penulis tergabung dalam departemen ERP bagian tim *technology* dan difokuskan untuk pengembangan *software* ERP menggunakan SAP dan Odoo.

Setiap pekerjaan yang dilakukan untuk mengembangkan *software* ERP dilakukan secara langsung di kantor (WFO) dan juga di-remote dari rumah (WFH). Selama mengembangkan *software* ERP, penulis koordinasi langsung dengan mentor atau *senior developer* lainnya, serta koordinasi juga dengan analis terkait konsep atau alur kerja *task* yang diambil dan *review* atau *feedback* dari apa yang sudah dikerjakan. Setiap harinya di departemen ERP akan ada *daily meeting* di Microsoft Teams, hal ini bertujuan untuk melaporkan apa yang sudah kami kerjakan kemarin dan yang akan kami kerjakan hari ini. Dalam dua minggu sekali akan ada *Sprint Meeting* untuk koordinasi antar tim, diskusi mempertimbangkan permintaan klien, dan perencanaan kesibukkan dalam waktu dekat dan jauh.

UMMN
UNIVERSITAS
MULTIMEDIA
NUSANTARA

3.2 Tugas yang Dilakukan

Dalam pelaksanaan kerja magang di KOMPAS Gramedia, hal yang pertama kali penulis lakukan adalah menyesuaikan laptop dengan kebutuhan *software-software* yang digunakan, contohnya penulis menggunakan Microsoft Teams dan Anydesk untuk berkomunikasi dan koordinasi dengan *developer* lain. Sementara untuk kebutuhan membuat code menggunakan PyCharm dengan *framework* Odoo dan pengaturan *task* dan *pull request* menggunakan Azure DevOps. Setelah penyesuaian, penulis mendapat *task* pertama untuk mempelajari SAP, *framework* Odoo, dan mengulang pembelajaran bahasa pemrograman Python. Pembelajaran ini dilakukan menggunakan OpenVPN untuk mengakses materi-materi yang sudah di-*upload* oleh *Senior Developer*, pembelajaran dilakukan selama enam minggu di rumah serta dibimbing oleh Fauzi Efriyadi secara berkala setiap dua minggu sekali.

Setelah pembelajaran, penulis mendapatkan *task* di bidang SAP untuk mengembangkan program *Material Document List* menggunakan bahasa pemrograman ABAP. Program *Material Document List* ini adalah program yang akan menampilkan laporan terkait material yang sudah dibeli dari vendor atau dikirimkan oleh KOMPAS Gramedia dari satu gudang ke gudang lainnya. Laporan ini secara garis besar berisi beberapa *field* yang menampilkan nilai satuan harga material, tujuan pengiriman material, dan lain-lain. Tugas penulis dalam pengembangan program ini adalah menambahkan beberapa *field* yang dibutuhkan klien untuk memastikan keamanan, ketepatan, detil material dari laporan material yang dihasilkan oleh program ini.

Setelah pengerjaan di bidang SAP, penulis diarahkan ke pengembangan *software* ERP untuk KGXpress. Pengembangan *software* ERP untuk KGXpress ini dibagi menjadi beberapa *task* yang nantinya para *developer* dapat mengambil *task* tersebut. *Software* ERP yang dikembangkan akan berupa *website* yang bisa diakses dari *web* atau *mobile*. *Website* ini nantinya akan mempunyai beberapa modul yang dikerjakan oleh seluruh *developer* CITIS, tetapi penulis difokuskan untuk mengembangkan modul *Transportation*.

3.3 Uraian Pelaksanaan Magang

Pelaksanaan kerja magang diuraikan seperti pada Tabel 3.1.

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

Minggu Ke -	Pekerjaan yang dilakukan
1 - 4	Perkenalan dengan CITIS KOMPAS Gramedia, Briefing untuk karyawan magang, melakukan instalasi <i>software</i> yang dibutuhkan untuk penerapan SAP, dan mempelajari SAP dan bahasa pemrograman ABAP
5	Menambahkan beberapa <i>field</i> untuk program <i>Material Document List</i> SAP dengan t-code ZMB51
6	Memperbarui <i>code</i> untuk penambahan field sesuai <i>feedback</i> dari analis, instalasi <i>framework</i> Odoo, Pycharm, dan beberapa modul Python, dan mempelajari <i>e-book</i> Odoo
7	Menambahkan <i>field</i> untuk program yang membuat laporan <i>Purchase Requisition Status</i> dan memperbaiki <i>error</i> saat menginstalasi Odoo serta melanjutkan belajar <i>framework</i> Odoo
8	Mengatur pengaturan instalasi <i>addons</i> untuk <i>framework</i> Odoo serta instalasi Dbeaver untuk koneksi ke <i>database</i> lokal dan mempelajari modifikasi <i>addons</i> Odoo
9	Mengambil <i>database</i> yang digunakan dari <i>server</i> untuk disimpan di penyimpanan lokal dan mempelajari tahapan proses bisnis untuk modul <i>Transportation</i>
10	Melanjutkan pembelajaran modifikasi <i>addons</i> khususnya untuk <i>field</i> dan <i>model</i> , tampilan xml, dan jenis-jenis <i>view</i> yang digunakan di <i>framework</i> Odoo
11	Membuat program yang memperlihatkan daftar <i>business partner</i> dan <i>customer</i> yang ada di <i>master data</i> , menyesuaikan program tersebut dengan <i>feedback</i> analis, dan pembuatan t-code dan transport program tersebut ke <i>server master</i> dan memodifikasi <i>form order express</i> agar beberapa <i>field</i> yang terdapat di <i>form</i> tersebut tidak bisa ditambah dan diubah

12	Menambahkan beberapa <i>field</i> baru untuk program <i>Material Document List</i> di SAP, membuat <i>master data</i> serta <i>model</i> dan <i>field</i> untuk kode faktur pajak
13	Menyesuaikan penambahan <i>field</i> baru di program <i>Material Document List</i> di SAP sesuai <i>feedback</i> analis dan membuat fitur <i>archive</i> untuk data di <i>zone</i> dan <i>area</i> di bagian <i>transportation</i>
14	Diskusi untuk ketepatan dan kejelasan untuk beberapa <i>field</i> di program <i>Material Document List</i> di SAP dengan analis dan <i>developer</i> lain, memperbaiki <i>bug</i> untuk fitur <i>archive</i> data <i>zone</i> dan <i>area</i> di <i>Transportation</i>
15 - 16	Memodifikasi UI di bagian <i>bagging</i> serta penambahan fitur <i>unbagging</i> , penambahan <i>field state</i> di bagian <i>bagging</i> , dan memperbaiki penambahan <i>field</i> di program <i>Material Document List</i> di SAP
17	Memodifikasi <i>sub menu history order</i> di <i>partnership</i> dan tampilan surat jalan di <i>Express Order</i> , membuat fitur untuk <i>import master data</i> di bagian <i>zona routing</i> , dan mempelajari <i>object relation mapping</i> dan <i>API decorators</i> di Odoo
18	Memodifikasi tampilan <i>form bagging</i> dengan menghilangkan <i>Add</i> dan <i>Create an item button</i> serta merubah <i>field id location</i> menjadi harus diisi di <i>form bagging</i> di <i>web</i> dan <i>mobile mode</i>
19 - 20	Membuat fitur untuk menampilkan <i>report</i> untuk <i>waybills</i> barang di gudang dan menampilkan <i>salesperson</i> di <i>form invoice generator</i>

Tabel 3.1 merupakan gambaran singkat dari pelaksanaan kerja magang yang penulis lakukan selama dua puluh minggu di KOMPAS Gramedia. Pada dasarnya, kerja magang dibagi-bagi menjadi beberapa bagian pekerjaan atau *task*, total *task* yang sudah diambil oleh penulis ada 18 buah. Setiap *task* mempunyai tingkat kesulitan dan kompleksitas yang berbeda-beda.

Pada minggu pertama hingga minggu keempat, penulis mendapat arahan *briefing* terkait informasi CITIS KOMPAS Gramedia tentang struktur perusahaan, metode pengembangan yang dipakai adalah metode *agile* serta cara melaporkan apa yang sudah dikerjakan di *daily meeting* di Microsoft Teams, *software* yang digunakan dalam pengembangan *software* ERP adalah OpenVPN, PyCharm, SAP Logon, Dbeaver, PostgreSQL, Azure DevOps, dan Microsoft Teams, lalu penulis

diminta untuk melakukan instalasi OpenVPN dan SAP Logon serta mempelajari bahasa pemrograman ABAP, struktur *internal table*, memanipulasi data di program SAP sesuai arahan Fauzi Efriyadi.

Pada minggu kelima, menambahkan beberapa *field* yaitu *rate*, *cost*, *organization description*, *withdrawn value*, *recipient*, dan *moving price* ke program SAP yang menampilkan daftar dokumen material untuk dikirim dari gudang ke gudang lain KOMPAS Gramedia atau daftar material yang dibeli dari vendor ke KOMPAS Gramedia. Pada minggu keenam, memperbaiki beberapa *field* yang sebelumnya sudah dimasukkan ke program SAP daftar dokumen material dan melanjutkan instalasi Odoo, modul Python yang dibutuhkan, PyCharm, PostgreSQL, dan Dbeaver sesuai dengan arahan *senior developer*. Selama instalasi dilakukan, penulis mempelajari *framework* Odoo dari *e-book* yang sudah diberikan akses dari Fauzi Efriyadi. Pada minggu ketujuh, menambahkan beberapa *field* baru untuk program SAP yang menampilkan status dari suatu *purchase requisition* (*Purchase Requisition* adalah dokumen internal yang digunakan oleh karyawan perusahaan untuk memberi informasi tentang daftar barang yang dibutuhkan oleh perusahaan) yaitu *to be delivered* dan *open quantity*. Pada minggu kedelapan, menyesuaikan pengaturan yang dibutuhkan untuk instalasi *addons* yang akan digunakan dalam *framework* Odoo serta instalasi Dbeaver yang nantinya akan digunakan memonitor data yang ada di database dengan membuat koneksi antara Dbeaver dengan database yang ada di lokal. Dalam minggu ini, penulis juga mulai mempelajari modifikasi *addons* dari video yang disediakan oleh *developer* yang senior.

Pada minggu kesembilan, melakukan *setup database* dari *server* untuk disimpan di penyimpanan lokal dengan menggunakan Dbeaver dan pgAdmin, jenis *database* yang digunakan adalah *database* PostgreSQL dan mempelajari tahapan proses bisnis *Express* atau *Transportation* di KOMPAS Gramedia, mulai dari pemesanan barang dari vendor yang nanti akan dikirim ke gudang (*receiving*), pencatatan *waybill*, *invoice*, *purchase requisition*, *purchase order* dan jalur pergerakan barang dari gudang ke gudang (*issuing*). Pada minggu kesepuluh, melanjutkan pembelajaran modifikasi *addons* Odoo, khususnya penggunaan model dan *field* di *framework* Odoo, untuk modifikasi model menggunakan Python, sementara tampilan yang bertujuan untuk menampilkan *field-field* yang terdapat di model tersebut akan menggunakan xml. Tampilan-tampilan seperti *form view*, *kanban view*, dan *tree view* juga dipelajari oleh penulis dalam minggu kesepuluh ini. Pada minggu kesebelas, membuat program SAP yang bertujuan untuk

memperlihatkan semua daftar *business partner* dan *customer* yang berelasi dengan KOMPAS Gramedia. Data-data yang akan ditampilkan seperti, *id customer*, nama *customer*, nomor telepon, alamat, NPWP, dan lain-lain. Pengambilan data dilakukan dengan menggunakan bahasa ABAP yang dimana pengambilan data di ABAP dinilai mirip seperti *query* SQL yang menggunakan metode SELECT, dan menempatkan data tersebut di tabel yang sudah dirancang dengan *code* yang membuat tampilan alv. Setelah proses *code* selesai, program tersebut akan dipindahkan atau *transport* ke server SAP agar bisa digunakan oleh klien. Di minggu ini, penulis juga mempelajari pembuatan *t-code* untuk program SAP baru. Memodifikasi *form order express* dengan memodifikasi *field customer*, *bill to*, *area*, *comodity*, *surcharge*, *origin waybill*, *special handling*, dan *document handling* agar *user* tidak bisa menambahkan atau merubah *record* untuk *field-field* tersebut. Pada minggu keduabelas, melakukan penambahan pada program SAP yang sudah ada yaitu *Material Document List*. *Field-field* yang ditambahkan antara lain adalah *reference document*, *storage location name*, *amount LC*, dan *vendor name*, penambahan *field* ini dibutuhkan oleh klien untuk mengetahui informasi yang jelas saat terjadinya pengembalian barang yang sudah dipesan dari vendor dikarenakan barang yang tidak sesuai atau dalam kondisi yang baik. Untuk minggu ini di bagian *framework* Odoo, membuat *master data* serta model dan tampilan xml untuk menampilkan *field-field* yang berkaitan dengan kode faktur pajak. Contoh kode faktur pajak yang dimaksud adalah 01 - Selain Pemungut PPN, 02 - Bendaharawan Negara, dan seterusnya. Kode-kode tersebut akan dimasukkan *user* terkait tipe pajak dan kode pajak yang digunakan oleh *customer*.

Pada minggu ketigabelas, memperbaiki *code* untuk penambahan *field-field* di program SAP yaitu *Material Document List* sesuai *feedback* dari analis. *Field-field* yang diperbaiki adalah harga total ditambah pajak, *vendor name*, *storage location name*, dan *GL account*. Hal ini diperlakukan karena ada kesalahan penulis dalam pengambilan data sehingga data yang ditampilkan di program tersebut tidak tepat. Lalu di bagian *framework* Odoo, membuat fitur untuk *archive* dan *unarchive* setiap *record* yang ada di *transportation* bagian *zone* dan *area*. Pada minggu keempatbelas, melakukan diskusi dengan analis terkait program SAP *Material Document List* tentang validasi *value Amount LC* dimana saat terjadinya proses *receiving*, *Amount LC* yang ditampilkan bernilai positif dan saat proses *issuing* menampilkan *Amount LC* yang bernilai negatif, serta memperbaiki pengambilan nilai *GL Account*. Penulis juga memperbaiki *bug* untuk fitur *archive* dan *unarchive*, *bug* yang dimaksud adalah *record-record* yang sudah di *archive* tidak bisa dicari

dengan *search bar*. Hal ini dapat diatasi dengan penambahan filter untuk mencari *record-record* yang sudah di *archive*. Pada minggu kelimabelas sampai keenambelas, memodifikasi UI di bagian *bagging* pada *Transportation*, modifikasi yang dilakukan adalah menambahkan fitur untuk *unbagging* serta tampilan *tree view* dan *form view*, modifikasi *scanner barcode* untuk barang yang sudah dilepas kemasannya (*unbagging*) dapat bisa di-*scan*, dan penambahan fitur untuk bisa menambahkan dan menghapus *record* barang yang ada di *tab bagging* dan *tab unbagging*. Modifikasi ini dilakukan untuk memudahkan *user* bisa mengetahui informasi terkait barang yang sudah dikemas untuk dikirim dan barang yang sudah sampai di tujuan dan dilepas kemasannya.

Pada minggu ketujuhbelas, memodifikasi *sub menu history order* di bagian *partnership* pada *Transportation* untuk menampilkan jumlah nilai asuransi yang dimiliki oleh barang di *history order* tersebut. Lalu, mengubah tampilan surat jalan saat dihasilkan dari menu *partnership*, serta membuat fitur untuk memasukkan data dengan tipe *file Excel* di menu bagian *Import Master Routing Zone*. Pada minggu kedelapanbelas, memodifikasi menu *Bagging* di bagian *form* saat membuat atau menambahkan *record* barang yang ingin dikemas dan dikirim. Modifikasi yang dilakukan adalah menghilangkan fitur penambahan barang yang ada di *list* tabel barang *bagging*, hal ini dilakukan karena penambahan ini sudah dapat dilakukan dengan *button* dan *scanner* yang ada diatas tabel tersebut. Setelah itu, memodifikasi *field location id* menjadi harus diisi saat menambahkan *record* barang *Bagging*. Semua modifikasi ini dilakukan pada tampilan *web* dan *mobile*. Pada minggu kesembilanbelas sampai keduapuluh, membuat fitur pada *Transportation* untuk menghasilkan laporan terkait pencatatan *waybill*, dalam laporan ini akan memuat beberapa informasi seperti *customer*, tanggal pembuatan *waybill*, *warehouse*, grup perusahaan, dan lain-lain. Selama pembuatan fitur *report waybills*, penulis memodifikasi *form invoice generator* dengan menampilkan informasi terkait *salesperson* di *form* tersebut.

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

3.3.1 Proses Pelaksanaan

Setiap pengerjaan *task* dalam pengembangan *software* ERP untuk manajemen Transportasi, diperlukan *software* dan *hardware* yang tepat. *Hardware* dan *Software* yang digunakan dalam pengembangan *software* ERP ini adalah sebagai berikut:

1. *Hardware* : ACER Aspire E 14
 - (a) Sistem Operasi : Windows 10 Education 64bit
 - (b) *Processor* : Intel Core i5-6200U 2.3 GHz
 - (c) RAM : 8 GB DDR4
 - (d) *Graphic Card* : NVIDIA GeForce 940 MX
2. *Software*
 - (a) PyCharm *Community Edition* versi 2021.1.3
 - (b) PostgreSQL 12 dan pgAdmin 4
 - (c) Dbeaver *Community Edition* versi 21.3.1
 - (d) Git versi 2.30.1
 - (e) Azure DevOps
 - (f) SAP Logon
 - (g) Stimulsoft
 - (h) Microsoft Teams
 - (i) Bahasa pemrograman Python dengan *framework* Odoo

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

3.3.2 Feature Requirement

Berdasarkan dari permintaan dan *task* yang dibuat oleh analis CITIS KOMPAS Gramedia, berikut adalah *requirement* yang perlu dipenuhi dari fitur penghasil *report waybills* ini :

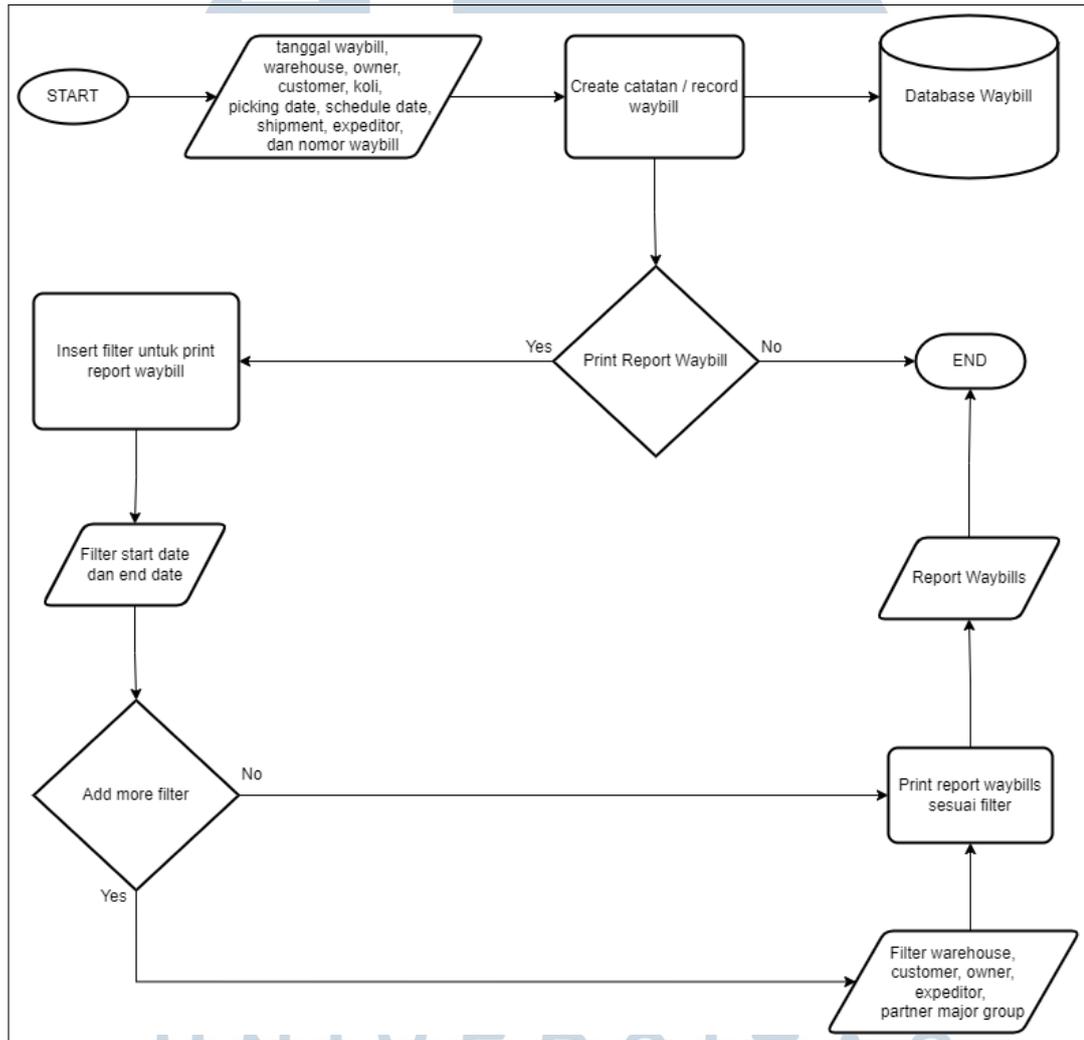
1. Dapat menampilkan *report waybills* yang berisikan daftar *record waybills* yang sudah dibuat
2. Dapat menampilkan informasi-informasi yang dimiliki oleh *record waybill* tersebut, seperti tanggal pembuatan *waybill*, *expeditor*, *service type*, nomor *waybill*, jumlah koli, *legacy number*, nama *customer*, *owner customer*, grup perusahaan, *picking number*, *source document*, *warehouse*, nomor *shipment*, *schedule date*, dan *picking date*
3. Dapat menampilkan informasi-informasi yang dimiliki *record waybill* sesuai dengan data yang disimpan di *database server*
4. Dapat melakukan filter untuk *record waybills*, mulai dari filter tanggal, filter *warehouse*, filter *customer*, filter *expeditor*, filter grup perusahaan, dan filter *owner customer*
5. Dapat membatasi *record waybills* yang akan ditampilkan sesuai dengan akses yang dimiliki oleh *user* terhadap perusahaan dan *warehouse* tertentu.
6. Dapat menampilkan tanggal beserta jam yang sesuai dengan *timezone user* tersebut

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

3.3.3 Perancangan

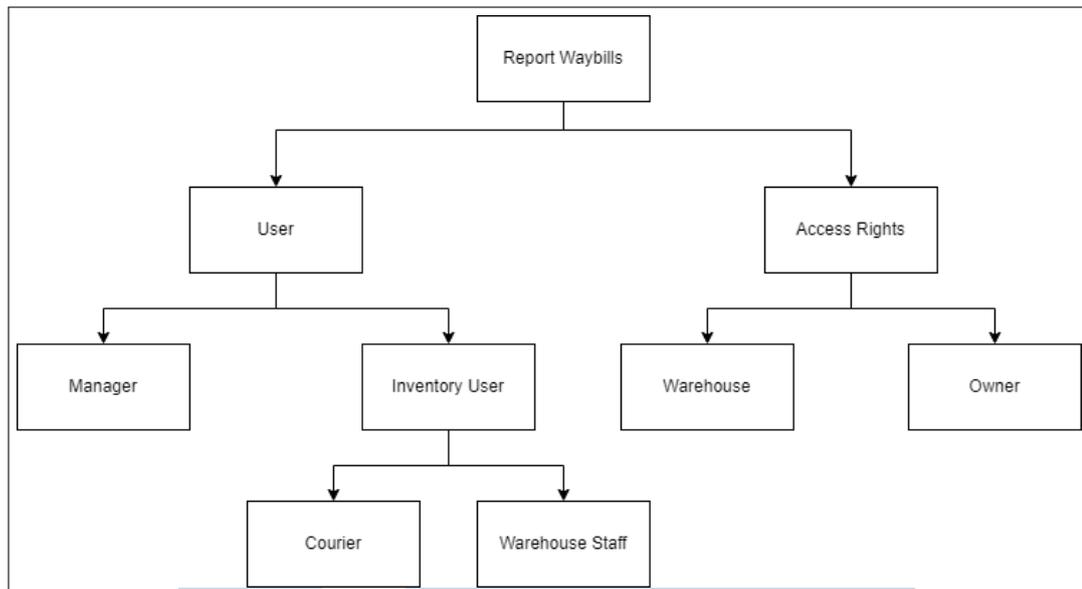
A Flowchart

Alur kerja untuk pencatatan *waybills* dalam bentuk *report waybills* digambarkan dalam flowchart di bawah ini.



Gambar 3.1. Flowchart Pembuatan Report Waybills

UNIVERSITAS
MULTIMEDIA
NUSANTARA

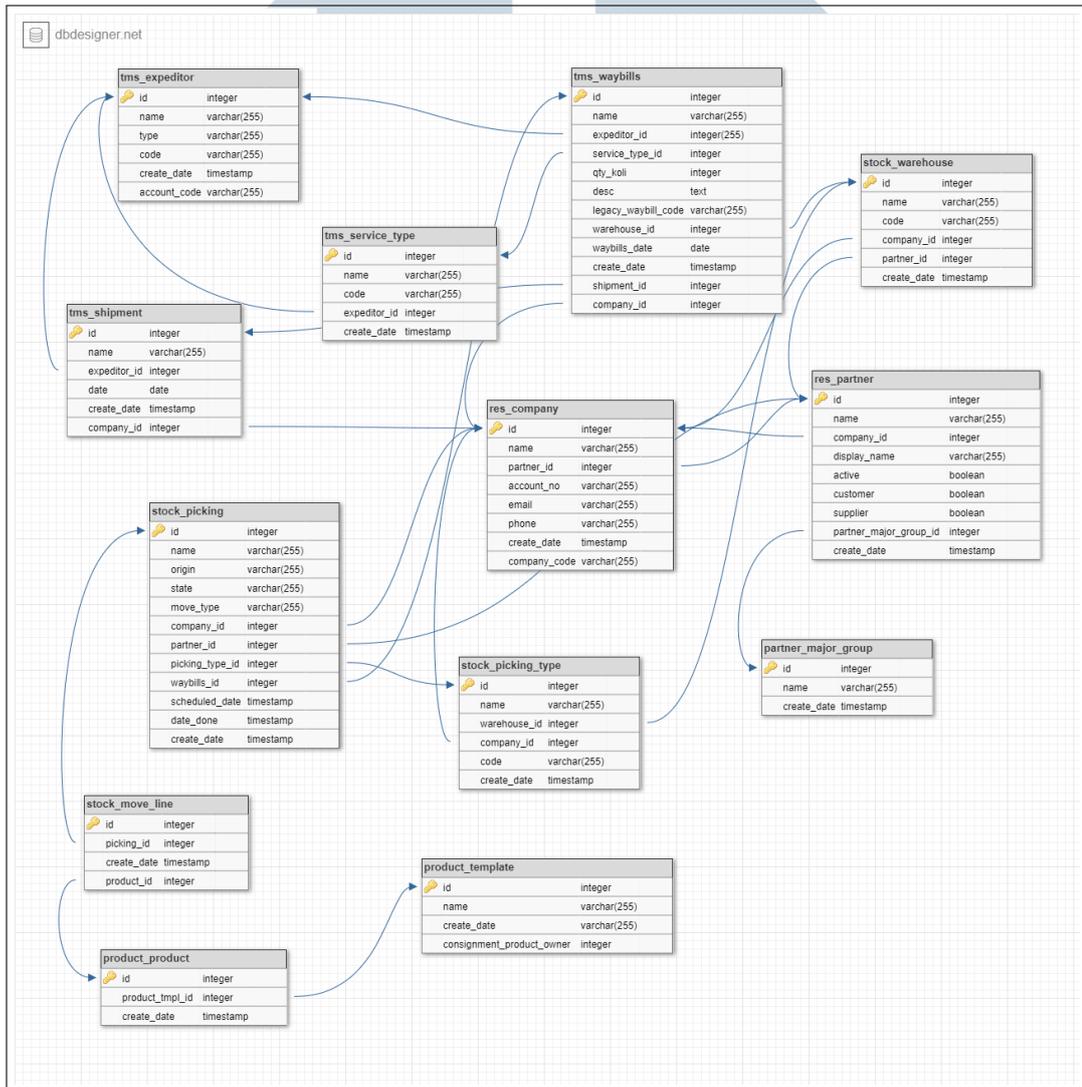


Gambar 3.2. Sitemap Report Waybills

Pada gambar 3.1 menggambarkan alur pembuatan *report waybills*. Saat adanya barang baru yang masuk ke gudang, akan dibuat catatan *waybill* terkait barang tersebut. Mulai dari keterangan nama barang, tanggal diterima dan dikirim, nama penerima dan pengirim, nama gudang, dan lain-lain. Sebelumnya, staf yang ada di gudang sudah dapat melihat rangkuman terkait catatan *waybill* yang ada. Akan tetapi, rangkuman tersebut hanya dapat diakses oleh staf gudang karena berupa *list* yang ada di *web*, sementara rangkuman catatan *waybill* ini dibutuhkan oleh kurir yaitu KGXpress untuk diperiksa ketepatan barang yang nantinya akan dikirim ke gudang lain. Untuk itu, setelah catatan *waybill* sudah dibuat, staf gudang akan membuat *report waybills* yang berisikan rangkuman catatan *waybill* dengan pilihan menggunakan filter atau tidak. Setelah *report waybills* sudah terbentuk di *software* akan di-*export* menjadi *file excel* untuk bisa dicetak dan menjadi bukti fisik. Sementara gambar 3.2 menggambarkan tipe *user* yang menggunakan fitur ini. *User manager* memiliki semua akses untuk *warehouse* dan *owner*, tetapi untuk *user inventory* memiliki akses untuk beberapa *warehouse* dan *owner*.

B Skema Database

Berikut adalah skema *database* yang digunakan dalam pembuatan fitur penghasil *report waybills* pada gambar di bawah ini.



Gambar 3.3. Skema Basis Data yang Digunakan dalam Pembuatan Fitur Penghasil *Report Waybills*

Dari skema basis data di atas, beberapa *field* yang diambil untuk dijadikan sebagai informasi terkait *record waybill* yang ditampilkan di *report waybills* tersebut. Selain *field* yang digunakan untuk ditampilkan di *report*, ada beberapa *field* yang menjadi penghubung antar tabel dan menjadi acuan saat melakukan *filtering record waybills*.

3.3.4 Implementasi

A Implementasi Pembuatan Model dan View di Odoo

Pembuatan *model* dan *view* di *framework* Odoo adalah hal yang paling umum dan dasar. Semua fitur penghasil *report* di Odoo menggunakan *model* yang dibuat di *file* Python dan *view* akan dibuat di *file* XML. Dalam *model* akan berisikan *field-field* yang nantinya ingin ditampilkan di *view*, seperti gambar di bawah ini.

```
start_date = fields.Datetime(help="Choose a date to get the trip start from that date",
                             default=_default_start_date)
end_date = fields.Datetime(help="Choose a date to get the trip up to that date", default=_default_end_date)
company_id = fields.Many2one(comodel_name='res.company', string='Company', required=True,
                             default=lambda self: self.env.user.company_id.id)
warehouse_id = fields.Many2one('stock.warehouse', string='Warehouse')
partner_id = fields.Many2one('res.partner', string='Customer')
partner_major_group_id = fields.Many2one('partner.major.group', string='Partner Major Group')
owner_ids = fields.Many2many('res.partner', string='Owner',
                              default=lambda self: self.env.user.product_owner_ids.ids)
expeditor_id = fields.Many2one('tms.expeditor', string='Expeditor',
                               help="Choose a expeditor to get specified report with expeditor")
```

Gambar 3.4. Model Fitur Penghasil *Report Waybills*

```
<field name="working_date" position="after">
  <field name="start_date" required="1"/>
  <field name="end_date" required="1"/>
  <field name="company_id" invisible="1"/>
  <field name="warehouse_id" options="{ 'no_create': True, 'no_open': True}"
        domain="[('company_id', '=', company_id)]"/>
  <field name="expeditor_id" options="{ 'no_create': True, 'no_open': True}"/>
  <field name="partner_id" options="{ 'no_create': True, 'no_open': True}"/>
  <field name="partner_major_group_id" options="{ 'no_create': True, 'no_open': True}"/>
  <field name="owner_ids" options="{ 'no_create': True, 'no_open': True}" force_save="1"
        context="{ 'tree_view_ref': 'kg_stock.kg_stock_view_partner_tree', 'show_internal_code': True}"
        widget="many2many_tags" domain="[('supplier', '=', True)]"/>
</field>
```

Gambar 3.5. View Fitur Penghasil *Report Waybills*

UNIVERSITAS
MULTIMEDIA
NUSANTARA

Field-field yang ditampilkan pada Gambar 3.3 dan Gambar 3.4 adalah *field* yang akan dijadikan sebagai filter untuk menyaring *record-record waybills* yang sesuai dengan keinginan *user*. Setiap *field* yang dideklarasikan di *model* memiliki tipe-tipe yang berbeda, contohnya untuk *field start_date* dan *end_date* menggunakan tipe *field Datetime* untuk menampilkan tanggal dan waktu dalam jam, menit, dan detik. Beberapa *field* seperti *company_id*, *warehouse_id*, *partner_id*, *partner_major_group_id*, dan *expeditor_id* menggunakan tipe *field* yang berelasi dengan *model* lain dengan relasi *Many to One*, artinya akan ada lebih dari satu *record* yang memiliki *company_id*, *warehouse_id*, *partner_id*, *partner_major_group_id*, dan *expeditor_id* yang sama dan dihubungkan ke *model* relasinya yang mempunyai *id* yang sifatnya unik. Jika diperlukan perubahan seperti perubahan teks *field* yang ditampilkan atau adanya penambahan fungsi di *field* tersebut, dapat menambahkan atribut di sebelah tipe *field*.

Field-field yang ada di *view XML* dapat dimodifikasi dengan menambahkan atribut pada *field* tersebut. Contohnya *field* ini harus diisi dapat menggunakan *required*, *field* tidak dapat menambahkan/mengubah *record field* yang sudah ada dapat menggunakan *options*, atau menyeleksi *record-record field* dengan kondisi tertentu dapat menggunakan *domain*.



B Implementasi Pembuatan Filter di Odoo

Pembuatan filter untuk fitur penghasil *report waybills* akan menerima *input* oleh *user*, seperti yang ada di gambar 3.5 dan disimpan dalam bentuk *config*, seperti gambar 3.6.

```
if not self.partner_id:
    conf_partner = 'ALL'
else:
    conf_partner = self.partner_id.display_name

if not self.warehouse_id:
    conf_warehouse = 'ALL'
else:
    conf_warehouse = self.warehouse_id.name

if not self.expeditor_id:
    conf_expeditor = 'ALL'
else:
    conf_expeditor = self.expeditor_id.name

if not self.partner_major_group_id:
    conf_partner_major_group = 'ALL'
else:
    conf_partner_major_group = self.partner_major_group_id.name

if not self.owner_ids:
    conf_owner = 'ALL'
else:
    conf_owner = ', '.join(self.owner_ids.mapped("name"))
```

Gambar 3.6. Kondisi Config Filter Fitur Penghasil *Report Waybills*

MULTIMEDIA
NUSANTARA

```
report_data = {
  "config": {
    "warehouse": conf_warehouse,
    "company_name": self.env.user.company_id.name,
    "start_date": f"{self.start_date}",
    "end_date": f"{self.end_date}",
    "expeditor": conf_expeditor,
    "partner_major_group_id": conf_partner_major_group,
    "owner": conf_owner,
    "customer": conf_partner,
    "printed_by": self.env.user.name,
    "printed_date": fields.Datetime.now()
  },
  "data": get_waybills_report_data
}
```

Gambar 3.7. Config Filter Fitur Penghasil *Report Waybills*

Dari *view* di XML dan konfigurasi untuk filter fitur penghasil *report waybills*, akan terbentuk filter dengan tampilan seperti gambar dibawah ini.

Report Filters:

Start Date	07/01/2021 00:00:00
End Date	12/15/2021 23:59:59
Warehouse	DC Cakung1
Expeditor	KGXpress
Customer	PT. ALFA OMEGA PRATAMA
Partner Major Group	K-Shop
Owner	ASUSTeK ✕

Buttons: View Report, Cancel

Gambar 3.8. View Filter Fitur Penghasil *Report Waybills*

Fungsi dari filter-filter tersebut akan berpengaruh terhadap kondisi yang harus dipenuhi saat pengambilan data dari *database*. Konfigurasi yang sudah menyimpan *input user* di filter akan dimasukkan dalam *query database* pada bagian *where*, contohnya adalah gambar di bawah ini.

```
where_customer_clause = ""
if self.partner_id:
    where_customer_clause = f"and sp.partner_id = {self.partner_id.id}"

where_expeditor_clause = ""
if self.expeditor_id:
    where_expeditor_clause = f"and tw.expeditor_id = {self.expeditor_id.id}"

where_partner_major_group = ""
if self.partner_major_group_id:
    where_partner_major_group = f"and pmg.id = {self.partner_major_group_id.id}"
```

Gambar 3.9. Pengaruh Filter Terhadap *Query Database* pada Fitur Penghasil *Report Waybills*



C Implementasi Query Database untuk Mengambil Record Waybill dari Database

Saat pembuatan *report waybills* dibutuhkan *record waybill* beserta data detail terkait *record waybill* tersebut untuk ditampilkan kepada *user*. Setiap *record waybill* ini bisa diambil dari *database* dengan penggunaan *query* dalam model yang sudah dibuat sebelumnya. *Query database* yang digunakan untuk mengambil setiap *record waybill* terletak pada gambar di bawah ini.

```
query = sql.SQL(
    """SELECT (tw.create_date at time zone 'UTC')::varchar as create_date, te.name as expeditor_name,
    tst.name as service_type_name, tw.name, tw.legacy_waybill_code, tw.qty_koli, rp.display_name as cust,
    rp2.display_name as owner, sp.name as picking_no, sp.origin as source_no,
    (sp.scheduled_date at time zone 'UTC')::varchar as scheduled_date,
    (sp.date_done at time zone 'UTC')::varchar as picking_date,
    sw.name as warehouse_name, ts.name as shipment_name
    FROM tms_waybills tw
    left join stock_picking sp on sp.waybills_id = tw.id
    left join tms_expeditor te on te.id = tw.expeditor_id
    left join tms_service_type tst on tst.id = tw.service_type_id
    left join tms_shipment ts on ts.id = tw.shipment_id
    left join stock_move_line sml on sml.picking_id = sp.id
    left join product_product pp on pp.id = sml.product_id
    left join product_template pt on pt.id = pp.product_tmpl_id
    left join res_partner rp on rp.id = sp.partner_id
    left join res_partner rp2 on rp2.id = pt.consignment_product_owner
    left join partner_major_group pmg on pmg.id = rp2.partner_major_group_id
    left join stock_warehouse sw on sw.id = tw.warehouse_id
    left join stock_picking_type spt on spt.id = sp.picking_type_id
    left join res_company rc on rc.id = rp2.company_id
    WHERE tw.create_date between {start_time} and {end_time}
    and tw.company_id = {company_id}
    and spt.code in ('outgoing')
    """ + where_expeditor_clause + """
    """ + where_warehouse_clause + """
    """ + where_owner_clause + """
    """ + where_partner_major_group + """
    """ + where_customer_clause + """
    """)
```

Gambar 3.10. *Query Database* untuk Pengambilan Data pada Fitur Penghasil *Report Waybills*

Setelah *record-record waybill* beserta data detail terkait *record* tersebut sudah diambil, nanti akan ditampilkan dalam bentuk tabel sebagai berikut.

07/09/2021 17:09:54	KGXpress	Reguler	KGXFalse000174	1	Toko Maju 88, Bpk. Ton	PT GRAMEDIA DIGITAL NUSANTARA	BSD/OUT/00128	S000003226	2021-09-07 18:49:21+07	2021-09-07 19:51:59+07	DC BSD	SHP/21/09/000155
08/09/2021 11:41:23	KGXpress	Reguler	KGXFalse000181	2	Toko Maju 88, Bpk. Ton	PT GRAMEDIA DIGITAL NUSANTARA	BSD/OUT/00128	S000003217	2021-09-03 14:50:41+07	2021-09-06 11:40:37+07	DC BSD	SHP/21/09/000154
08/09/2021 14:47:59	KGXpress	Reguler	KGXFalse000196	2	Toko Maju 88, Bpk. Ton	PT GRAMEDIA DIGITAL NUSANTARA	BSD/OUT/00127	S000003218	2021-09-06 14:47:13+07	2021-09-06 14:47:33+07	DC BSD	
13/09/2021 10:50:39	KGXpress	Reguler	KGXFalse000193	1	ASUSTEK CUSTOMER	ASUSTeK	CKG/OUT/00130	S000003226	2021-09-13 10:44:33+07	2021-09-13 10:47:33+07	DC Cakung1	SHP/21/09/000157
13/09/2021 10:50:39	KGXpress	Reguler	KGXFalse000193	1	ASUSTEK CUSTOMER	ASUSTeK	CKG/OUT/00130	S000003226	2021-09-13 10:44:33+07	2021-09-13 10:47:33+07	DC Cakung1	SHP/21/09/000157
21/09/2021 11:51:57	KGXpress	Reguler	KGXFalse000224	2	Toko Maju 88, Bpk. Ton	PT GRAMEDIA DIGITAL NUSANTARA	BSD/OUT/00134	S000003232	2021-09-21 11:50:50+07	2021-09-21 11:51:06+07	DC BSD	SHP/21/09/000158
18/10/2021 10:43:19	KGXpress	Reguler	KGXFalse0001157	1	PT. Dystar Colours Indonesia	PT GRAMEDIA DIGITAL NUSANTARA	CKG/OUT/00139	S000003385	2021-10-18 10:42:39+07	2021-10-18 10:42:51+07	DC Cakung1	SHP/21/10/000162
18/10/2021 09:22:40	KGXpress	Reguler	KGXFalse0001148	1	PT. Dystar Colours Indonesia	PT GRAMEDIA DIGITAL NUSANTARA	CKG/OUT/00137	S000003381	2021-10-18 09:14:04+07	2021-10-18 09:21:22+07	DC Cakung1	

Printed Date : 16/12/2021 05:17:50
Printed By : There

Gambar 3.11. View *Record-record* Waybill Beserta Data pada Fitur Penghasil *Report Waybills*

Jika tidak ada kondisi yang terbentuk dari *input user* di filter, maka akan menampilkan semua *record waybill* sesuai interval waktu tertentu. Sementara, bila ada filter apapun yang diisi, *record waybill* yang ditampilkan akan sesuai dengan kondisi dari filter tersebut. Saat *report waybills* sudah keluar, akan ditampilkan keterangan bahwa semua *record waybill* atau *record waybill* yang sudah disaring dalam interval tanggal tersebut yang ditampilkan di *report* tersebut, seperti gambar yang ada di bawah ini.

Report Waybills			
Start Date	:08.31.2021	Warehouse	: ALL
End Date	:12.15.2021	Expedito	: ALL
		Customer	: ALL
		Partner Major Group	: ALL
		Owner	: ALL

Gambar 3.12. Keterangan tanpa Pengaruh Filter untuk Pengambilan Data pada Fitur Penghasil *Report Waybills*

Report Waybills			
Start Date	:08.31.2021	Warehouse	: DC Cakung1
End Date	:12.15.2021	Expedito	: KGXpress
		Customer	: ASUSTEK CUSTOMER
		Partner Major Group	: K-Shop
		Owner	: ASUSTeK

Gambar 3.13. Keterangan dengan Pengaruh Filter untuk Pengambilan Data pada Fitur Penghasil *Report Waybills*

D Implementasi Penyesuaian Timezone untuk Field yang Bertipe Datetime

Saat ada barang yang masuk ke gudang, akan dilakukan pencatatan *waybill* beserta jam berapa barang tersebut masuk ke gudang. Hal ini dilakukan karena adanya jam operasional pada gudang tersebut. Karena ada perbedaan *timezone* dari sistem *framework* Odoo dengan *timezone* daerah gudang, sehingga perlu dilakukan penyesuaian. Penyesuaian yang dimaksud adalah dengan mengubah apapun *default timezone* yang diberikan dari Odoo pada *user* akan diubah menjadi GMT+7, seperti gambar di bawah ini.

```
user_tz = self.env.user.tz or "UTC"
user_tz = pytz.timezone(user_tz)
user_tz_datetime = user_tz.localize(fields.Datetime.from_string(fields.Date.today()).replace(hour=0))
utc_tz_datetime = user_tz_datetime.astimezone(pytz.utc)
utc_datetime = fields.Datetime.to_string(utc_tz_datetime)
return utc_datetime
```

Gambar 3.14. Penyesuaian *Timezone* pada *Field Datetime* pada Fitur Penghasil *Report Waybills*

Maka dari itu, tampilan *field* dengan tipe *Datetime*, akan ada perubahan jam di *view* dan *database* seperti gambar di bawah ini.

Schedule Date	Picking Date
2020-01-17 11:15:58+07	2020-01-17 11:28:00+07
2021-01-07 10:45:21+07	2021-01-07 10:46:48+07
2021-07-15 10:13:39+07	2021-07-15 10:14:04+07

Gambar 3.15. *View Field Datetime* sesuai *Timezone*

E Implementasi Limit yang Dimiliki User terhadap Warehouse dan Owner Tertentu

Setiap *user* memiliki *company* yang berbeda-beda dan sudah ditetapkan di setiap *user*. Karena adanya *company* yang sudah ditetapkan pada setiap *user*, maka *user* tersebut hanya bisa mengakses *customer* dan *owner* yang berelasi dengan *company* tersebut. Selain limit *company*, ada beberapa limit yang mempengaruhi *field-field* yang ditampilkan di *report waybill*. Pengaturan limit ini dapat diubah-ubah di pengaturan *user* dan limit yang bisa diganti hanya limit *owner*, limit *warehouse*, dan limit *company*. Jika limit *warehouse* dan limit *owner* ditetapkan pada pengaturan *user* tersebut, maka pilihan yang ada di filter *warehouse* dan filter *owner* akan sesuai dengan limit yang sudah ditetapkan di pengaturan *user*. Data-data yang ditampilkan pada *report waybills* juga akan sesuai dengan limit yang sudah ditetapkan pada pengaturan *user*. Contohnya, *user* ini menggunakan limit *warehouse* Dark Store dan DC Cakung seperti di gambar 3.15 dan limit *owner* ASUSTek dan PT. Anugrah Dina Fortuna (A) pada *company* egrosir.id, seperti gambar setelahnya.



Gambar 3.16. Limit Warehouse pada Fitur Penghasil Report Waybills



Gambar 3.17. Limit Owner dan Company pada Fitur Penghasil Report Waybills

Dari kedua gambar diatas dengan limit yang sudah ditetapkan, maka tampilan pilihan di filter hanya muncul yang sudah dilimit, seperti gambar di bawah ini.

Gambar 3.18. Tampilan Filter dengan Limit pada Fitur Penghasil *Report Waybills*

Jadi, dengan menggunakan atau tanpa menggunakan filter, *report waybills* yang dihasilkan akan sesuai dengan limit yang ditetapkan seperti gambar dibawah ini.

Report Waybills												
Start Date	: 06.30.2021					Warehouse	: ALL					
End Date	: 12.16.2021					Expeditor	: ALL					
						Customer	: ALL					
						Partner Major Group	: ALL					
						Owner	: ALL					
Date	Expeditor	Service Type	Waybills No	Logis No	Kol Customer	Owner	Packing No	Source Document	Schedule Date	Picking Date	Warehouse	Shipment
06/01/2021	KDHome	RCD	KDHPAAM000024	KD0270000000	1	ASUSTek	0500700005	00000117	02/01/2021	02/01/2021	DARK STORE	SHF210000048
06/02/2021	KDHome	RCD	KDHPAAM000024	KD0270000000	1	ASUSTek	0500700005	00000117	02/02/2021	02/02/2021	DARK STORE	SHF210000048
06/15/2021	KDHome	RCD	KDHPAAM000000		1	ASUSTek	0500700002	07C7019	02/15/2021	02/15/2021	DARK STORE	
06/16/2021	KDHome	RCD	KDHPAAM000000		1	ASUSTek	0500700002	07C7019	02/16/2021	02/16/2021	DARK STORE	
06/16/2021	KDHome	RCD	KDHPAAM000000		1	ASUSTek	0500700002	07C7019	02/16/2021	02/16/2021	DARK STORE	

Gambar 3.19. Tampilan Report dengan Limit pada Fitur Penghasil *Report Waybills*



3.4 Kendala dan Solusi yang Ditemukan

3.4.1 Kendala yang Ditemukan

Kendala-kendala yang ditemukan saat melakukan praktik kerja magang di KOMPAS Gramedia

- a. Kurangnya pengetahuan penulis akan penerapan *object relation mapping* pada *framework* Odoo saat pembuatan model.
- b. Kesulitan dalam melakukan implementasi salah satu *requirements* yang diberikan. Implementasi yang dimaksud adalah implementasi limit akses *user* untuk filter dan *record* di *report waybills*.

3.4.2 Solusi yang Ditemukan

Dari kendala-kendala tersebut, solusi yang dilakukan untuk mengatasi kendala tersebut dalam melakukan praktik kerja magang di KOMPAS Gramedia

- a. Melakukan pembelajaran mandiri di luar jam kerja, untuk mempelajari sistem *object relation mapping* pada *framework* Odoo dan mencari referensi dari fitur yang mirip dan sudah dibuat sebelumnya.
- b. Bertanya langsung ke analis mengenai alur dari *requirements* yang akan di implementasikan dan menanyakan *senior developer* solusi dari permasalahan itu.

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