

BAB 3 PELAKSANAAN KERJA MAGANG

3.1 Kedudukan dan Organisasi

Pada PT. Sumber Inovasi Informatika memiliki kedudukan sebagai *Software Developer Intern (Backend)* pada *project* milik klien PT Sumber Inovasi Informatika, yaitu UDN dengan *jobdesc* untuk melakukan pengembangan untuk sistem *Enterprise Resource Planning* dan memperbaiki *bug* yang muncul pada sistem berdasarkan pada permintaan klien PT Sumber Inovasi Informatika. Dalam melakukan magang di PT. Sumber Inovasi Informatika, program magang disupervisi oleh Bapak Hans Permana yang memiliki kedudukan sebagai *Chief Operating Officer*.

3.2 Tugas yang Dilakukan

Selama menjalani magang pada PT. Sumber Inovasi Informatika, terdapat beberapa tugas yang dilakukan. Diantaranya adalah sebagai berikut.

1. Mempelajari mengenai *framework* ERP yaitu Frappe untuk digunakan selama melakukan magang di PT Sumber Inovasi Informatika
2. Mempelajari dan menggunakan metode Agile dalam mengerjakan tugas-tugas yang diberikan
3. Mengembangkan sistem *Enterprise Resource Planning* untuk klien dari PT Sumber Inovasi Informatika
4. Memperbaiki *Bug* atau menambahkan *feature* agar tidak menghambat proses jual beli yang dilakukan oleh klien

3.3 Uraian Pelaksanaan Magang

Dalam melakukan magang di PT Sumber Inovasi Informatika, pelaksanaan magang dilakukan dengan cara mengerjakan tugas berdasarkan pada *ticket* yang diberikan oleh *Product Owner*. *Ticket* merupakan sebuah sistem yang berisikan mengenai sebuah pelaporan permasalahan [5], *bug*, atau permintaan penambahan *feature* dari klien PT Sumber Inovasi Informatika. Pada *framework* Frappe

yang digunakan untuk mengembangkan ERP dan *middleware*, apabila melakukan perubahan karena menambahkan *feature* baru maka akan dilakukan pada sebuah *doctype* yang terkait. *Doctype* atau *document type* dapat digambarkan sebagai sebuah *database* yang berisikan *field*, *property* dan hal-hal terkait dengan sebuah *field* dalam sebuah dokumen. Kemudian, pelaksanaan magang dapat diuraikan seperti pada tabel 3.1.

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

Minggu Ke -	Pekerjaan yang dilakukan
1	Mempelajari mengenai <i>Enterprise Resource Planning</i> dan mempelajari <i>framework</i> yang digunakan, yaitu Frappe. Selain itu, melakukan <i>employee orientation</i> di PT. Sumber Inovasi Informatika
2	Melakukan pengenalan terhadap <i>project</i> dan melakukan instalasi <i>docker</i> , <i>ubuntu</i> dan <i>local environment</i> yang akan digunakan selama mengerjakan <i>project</i> milik klien
3	Mengerjakan <i>ticket</i> untuk menambahkan fitur dan mengubah <i>property</i> pada beberapa <i>field warehouse</i> pada ERP, <i>middleware gudang</i> dan <i>middleware penjualan</i>
4	Mengerjakan <i>ticket</i> untuk mengubah <i>property</i> pada beberapa <i>field</i> menjadi <i>not hidden</i> di ERP dan <i>middleware penjualan</i>
5	Mengerjakan <i>ticket</i> untuk menghilangkan desimal dari hasil <i>translate</i> dan membuat <i>field</i> baru pada <i>middleware gudang</i>
6	Mengerjakan <i>ticket</i> untuk memodifikasi <i>doctype customer</i> dan <i>doctype e-cash</i>
7	Melakukan <i>investigate barcode</i> pada <i>doctype Item</i> serta mengerjakan <i>feedback</i> untuk beberapa tiket pada <i>doctype Customer</i>
8	Melakukan <i>investigate barcode</i> pada <i>doctype Item</i> serta melakukan modifikasi untuk <i>doctype Journal Entry</i> dan <i>E-Cash</i>
9	Mengerjakan <i>edit workflow</i> dan mengerjakan filter divisi berdasarkan pada <i>division head</i>
10	Mengerjakan <i>feedback</i> dari tiket filter divisi serta melakukan <i>investigate</i> dan <i>fix Sales Invoice ID</i> dan <i>Customer ID</i> pada <i>E-Cash</i>

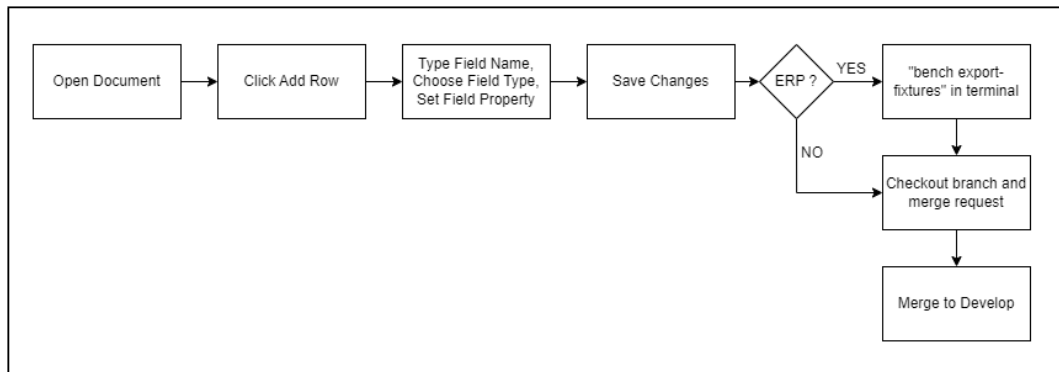
Minggu Ke -	Pekerjaan yang dilakukan
11	Melakukan discuss mengenai <i>investigate</i> dan <i>fix</i> Sales Invoice ID dan Customer ID pada E-Cash. Kemudian menambahkan Delivery Note ID dan mengerjakan <i>Event Streaming</i>
12	Melakukan modifikasi API Get Detailed Customer dan menambahkan <i>field</i> baru pada <i>doctype</i> Customer
13	Menambahkan <i>field</i> pada Sales Order dan memodifikasi <i>job auto create</i> Sales Order. Kemudian, melakukan handover bersama dengan tim
14	Melakukan <i>fix</i> Stock Entry <i>target warehouse</i> . Kemudian, menambahkan <i>field</i> baru dan mengerjakan beberapa <i>tiket</i> yang mendapatkan <i>feedback</i>
15	Menambahkan opsi dan <i>field</i> baru pada <i>doctype</i> Customer yang dilanjutkan dengan memodifikasi API. Kemudian, melakukan discuss dan review bersama dengan tim.
16	Menambahkan validasi saat Sales Invoice minus atau 0. Kemudian, mengerjakan <i>job customer registration</i>
17	Melakukan <i>investigate and fix write off</i> pada Sales Invoice
18	Menambahkan <i>field</i> Overdays beserta dengan <i>function</i> untuk <i>field</i> tersebut

3.4 Hasil dan Implementasi

Pada *project* yang dikerjakan selama melakukan magang, terdapat tiga *sites* yang dikerjakan, yaitu ERPNext, *Middleware* penjualan dan *Middleware* gudang. Pada ERPNext, kustomisasi terhadap sebuah *field* dilakukan dengan cara *Customized Field*. Kemudian, pada *Middleware* penjualan dan gudang, kustomisasi terhadap sebuah *field* dilakukan dengan cara *Edit Doctype*.

3.4.1 Custom Field

Pada *project* yang dikerjakan selama melakukan magang, apabila menambahkan sebuah *field* baru pada *document* pada ERP maupun pada *middleware*, maka penambahan tersebut akan ditambahkan pada bagian *Custom Field*. Flow dari pembuatan *field* baru atau *Custom Field* tersebut dapat dilihat pada gambar 3.1



Gambar 3.1. *Flow Custom Field*

Pada *flow* tersebut dimulai dari memilih *doctype* yang akan ditambahkan sebuah *field* baru. Kemudian, klik *add row* yang terletak pada paling bawah *doctype* tersebut yang dilanjutkan dengan mengisi nama, tipe dan properti dari *field* baru tersebut.

Jika telah sesuai dengan *ticket* yang diberikan, selanjutnya adalah *save* untuk perubahan tersebut. Apabila perubahan tersebut dilakukan pada ERP, maka selanjutnya adalah mengetikkan "*bench export-fixtures*" pada terminal yang mengarah ke ERP yang dilanjutkan dengan melakukan *checkout branch* ke *sites* ERP untuk dilakukan *merge request* kepada Senior Developer. Sedangkan jika perubahan tersebut dilakukan pada *middleware*, selanjutnya adalah melakukan *checkout branch* ke *sites middleware* yang dituju yaitu penjualan atau gudang untuk dilakukan *merge request* kepada Senior Developer.

Pada pembuatan *field* baru tersebut, tipe yang dapat dipilih seperti *Select*, *Check*, *Time*, *Link*, *Text* dengan *default field type* adalah *Data*. Pada laporan magang ini, akan difokuskan pada *field* baru dengan tipe *Checkbox* atau *Add New Tagging*, dan tipe *Select* atau *Add New Options*.

A. *Add New Tagging*

Pada *project* yang dikerjakan selama melakukan magang, saat membuat sebuah *tag* baru atau *checkbox* baru, Hal yang dilakukan adalah membuat *field* baru dengan tipe *check*. Hal tersebut dapat dilihat pada gambar 3.2.

Gambar 3.2. Tampilan pada bagian *edit doctype Item Group*

Setelah *field* tersebut dibuat dan kemudian disimpan, maka hal yang dilakukan adalah menambahkan fixtures pada file *hooks.py* seperti pada gambar 3.3.

```
"Item Group-custom is bonus item",
```

Gambar 3.3. Tampilan pada file *hooks.py*

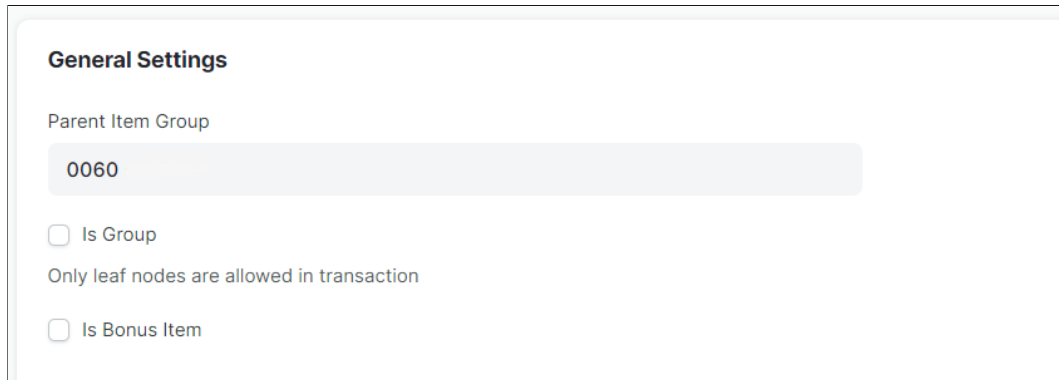
Setelah menambahkan fixtures tersebut, selanjutnya adalah melakukan "*bench export-fixtures*". Setelah dilakukan *bench export-fixtures*, maka pada *file custom field* akan otomatis berubah seperti pada gambar 3.4.

```
{
  "allow_in_quick_entry": 1,
  "allow_on_submit": 0,
  "bold": 0,
  "collapsible": 0,
  "collapsible_depends_on": null,
  "columns": 0,
  "default": null,
  "depends_on": null,
  "description": null,
  "docstatus": 0,
  "doctype": "Custom Field",
  "dt": "Item Group",
  "fetch_from": null,
  "fetch_if_empty": 0,
  "fieldname": "is_bonus_item",
  "fieldtype": "Check",
```

Gambar 3.4. Tampilan pada file *property setter*

Sehingga, pada akhirnya akan menghasilkan sebuah *checkbox* baru. Dalam

hal ini, contoh *doctype* yang digunakan adalah *doctype* Item Group.



General Settings

Parent Item Group

0060

Is Group

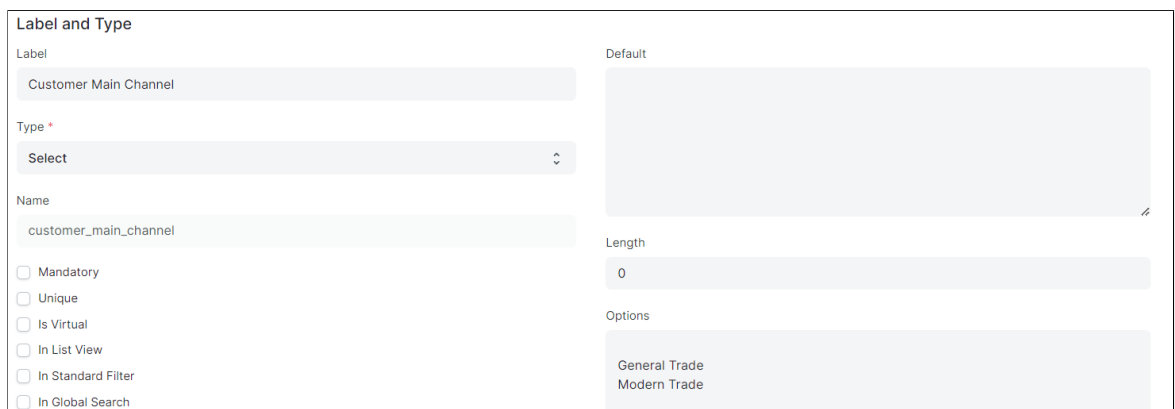
Only leaf nodes are allowed in transaction

Is Bonus Item

Gambar 3.5. Tampilan *checkbox* pada *doctype* Item Group

B. Add Option Field

Pada *project* yang dikerjakan selama melakukan magang, saat membuat sebuah *field* yang memiliki opsi, perubahan yang dilakukan adalah menambahkan *value* pada *field options* dan dengan menggunakan tipe data Select. Hal tersebut dapat dilihat pada gambar 3.6



Label and Type

Label

Customer Main Channel

Type *

Select

Name

customer_main_channel

Mandatory

Unique

Is Virtual

In List View

In Standard Filter

In Global Search

Default

Length

0

Options

General Trade

Modern Trade

Gambar 3.6. Tampilan pada bagian *edit doctype*

Kemudian, setelah perubahan tersebut disimpan, maka hal yang dilakukan adalah menambahkan *fixtures* pada file hooks seperti pada gambar 3.7

```
"Customer-custom_customer_main_channel",
```

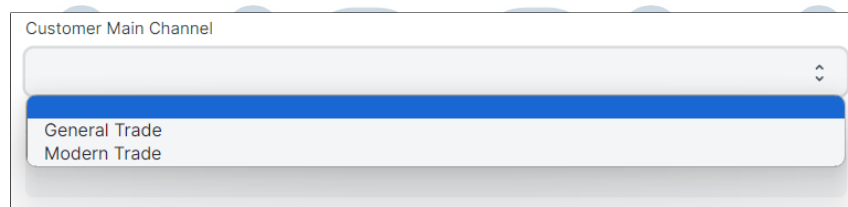
Gambar 3.7. Tampilan pada bagian file hooks

Setelah menambahkan *code fixtures* tersebut, selanjutnya melakukan *”bench export-fixtures”* pada terminal yang telah terhubung dengan docker. Sehingga, pada *custom field* akan otomatis menambahkan *code* seperti yang ditunjukkan pada gambar 3.8

```
"insert_after": "customer_type",
"is_system_generated": 0,
"is_virtual": 0,
"label": "Customer Main Channel",
"length": 0,
"mandatory_depends_on": null,
"modified": "2024-04-30 08:57:16.223478",
"module": null,
"name": "Customer-custom_customer_main_channel",
"no_copy": 0,
"non_negative": 0,
"options": "\nGeneral Trade\nModern Trade",
"permlevel": 0,
"precision": "",
```

Gambar 3.8. Tampilan pada bagian file custom field

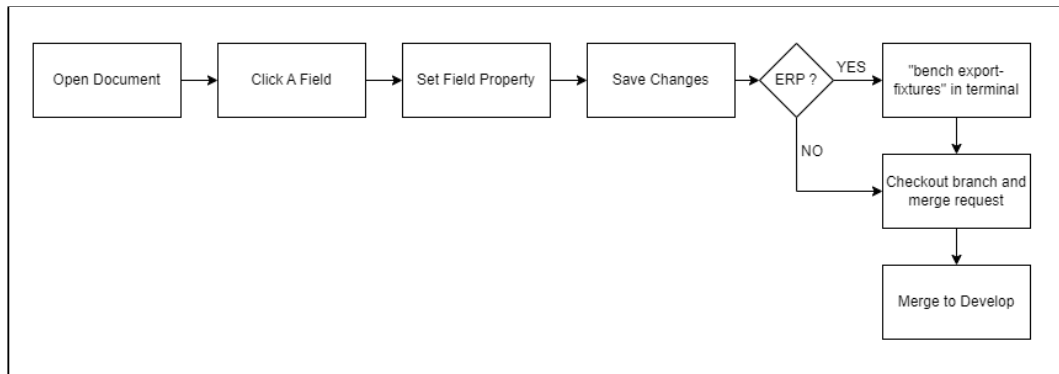
Sehingga pada doctype yang telah ditambahkan *field* dengan tipe data *options* akan muncul dengan opsi yang telah ditambahkan pada bagian edit doctype seperti pada gambar 3.9



Gambar 3.9. Tampilan pada bagian *field Customer Main Channel*

3.4.2 *Property Setter*

Pada *project* yang dikerjakan selama melakukan magang, apabila menambahkan sebuah *property* pada *field* yang ada pada ERP maupun pada *middleware*, maka penambahan tersebut akan dimasukkan pada bagian *Property Setter*. Flow dari penambahan *property* tersebut dapat dilihat pada gambar 3.10



Gambar 3.10. *Flow Property Setter*

Pada *flow* tersebut dimulai dari memilih *doctype* yang akan diubah. Kemudian, klik *field* pada *doctype* tersebut yang akan ditambahkan sebuah *property*. Selanjutnya adalah memilih *property* berdasarkan pada *ticket* yang telah diberikan dan dilanjutkan dengan *save* perubahan tersebut apabila sudah sesuai dengan yang ada pada *ticket*.

Jika telah sesuai dengan *ticket* yang diberikan, selanjutnya adalah *save* untuk perubahan tersebut. Apabila perubahan tersebut dilakukan pada ERP, maka selanjutnya adalah mengetikkan "*bench export-fixtures*" pada terminal yang mengarah ke ERP yang dilanjutkan dengan melakukan *checkout branch* ke *sites* ERP untuk dilakukan *merge request* kepada Senior Developer. Sedangkan jika perubahan tersebut dilakukan pada *middleware*, selanjutnya adalah melakukan *checkout branch* ke *sites middleware* yang dituju yaitu penjualan atau gudang untuk dilakukan *merge request* kepada Senior Developer.

Pada penambahan *property* pada sebuah *field*, *property* yang dapat dipilih umumnya adalah *mandatory*, *hidden*, *read only*, *allow on submit*, *ignore user permission*, *in list view*, *in standard filter*. Pada laporan magang ini akan difokuskan pada *property mandatory*, *hidden*, dan *ignore user permission*.

A. *Set Field to Mandatory*

Pada *project* yang dikerjakan selama melakukan magang, saat membuat sebuah *field* menjadi *mandatory*, perubahan yang dilakukan adalah mencentang pada bagian *checkbox* pada *box Mandatory*. Hal tersebut dapat dilihat pada gambar 3.11.

Editing Row #15

Label: Warehouse

Type: Data

Name: warehouse

Length: 0

Mandatory

Virtual

Index

Options: [Empty field]

For Links, enter the DocType as range. For Select, enter list of Options, each on a new line.

Gambar 3.11. Tampilan pada bagian edit doctype field warehouse

Setelah *checkbox* tersebut dicentang dan kemudian disimpan, maka pada *file* json akan otomatis menambahkan "reqd": 1. Dalam hal ini, contoh *file* yang otomatis berubah adalah quotation.py sehingga seperti pada gambar 3.12.

```
{
  "fetch_from": "branch.standard_warehouse",
  "fieldname": "warehouse",
  "fieldtype": "Data",
  "label": "warehouse",
  "reqd": 1
},
```

Gambar 3.12. Tampilan pada quotation.py

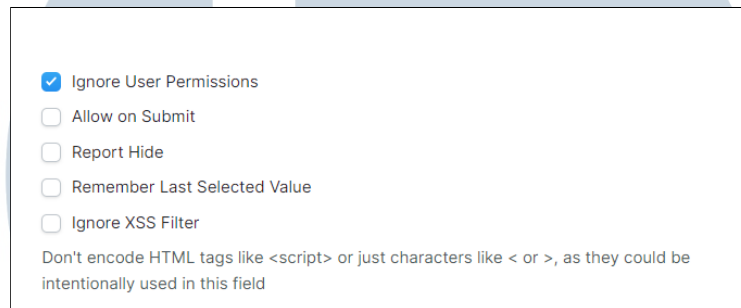
Sehingga, pada akhirnya akan menghasilkan sebuah *field* menjadi *mandatory*. Dalam hal ini, contoh *field* yang digunakan adalah *field* warehouse. Pada *field* warehouse akan memiliki tanda bintang berwarna merah atau dapat dikatakan sebagai *field* mandatory seperti pada gambar 3.13.

Warehouse *

Gambar 3.13. Hasil edit pada field Warehouse

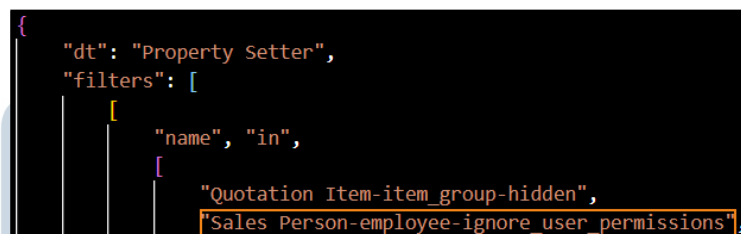
B. Ignore User Permission

Pada *project* yang dikerjakan selama melakukan magang, saat membuat sebuah *field* menjadi bersifat *ignore user permission*, perubahan yang dilakukan adalah mencentang pada bagian *checkbox* pada *box Ignore User Permissions*. Hal tersebut dapat dilihat pada gambar 3.14.



Gambar 3.14. Tampilan pada bagian *edit doctype field employee*

Setelah *checkbox* tersebut dicentang dan kemudian disimpan, maka hal yang dilakukan adalah menambahkan fixtures pada file *hooks.py* seperti pada gambar 3.15.



Gambar 3.15. Tampilan pada file *hooks.py*

Setelah menambahkan fixtures tersebut, selanjutnya adalah melakukan "*bench export-fixtures*". Setelah dilakukan *bench export-fixtures*, maka pada *file property setter* akan otomatis berubah seperti pada gambar 3.16.

MULTIMEDIA
NUSANTARA

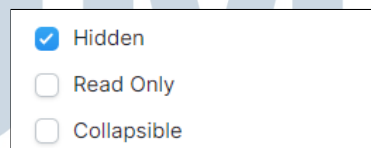
```
{
  "default_value": null,
  "doc_type": "Sales Person",
  "docstatus": 0,
  "doctype": "Property Setter",
  "doctype_or_field": "DocField",
  "field_name": "employee",
  "is_system_generated": 0,
  "modified": "2024-02-02 11:01:45.124714",
  "module": null,
  "name": "Sales Person-employee-ignore_user_permissions",
  "property": "ignore_user_permissions",
  "property_type": "Check",
  "row_name": null,
  "value": "1"
},
```

Gambar 3.16. Tampilan pada file property setter

Sehingga, pada akhirnya akan menghasilkan sebuah *field* menjadi *ignore user permissions* yang memiliki arti semua *user* dapat melakukan *edit*. Dalam hal ini, contoh *field* yang digunakan adalah *field employee*.

C. Set Field Hidden

Pada *project* yang dikerjakan selama melakukan magang, saat membuat sebuah *field* menjadi bersifat *hidden*, perubahan yang dilakukan adalah *klik check* pada bagian *checkbox* pada *box Hidden*. Hal tersebut dapat dilihat pada gambar 3.17.



Gambar 3.17. Tampilan pada bagian *edit doctype Quotation*

Setelah *checkbox* tersebut dilakukan *check* dan kemudian disimpan, maka hal yang dilakukan adalah menambahkan fixtures pada file *hooks.py* seperti pada gambar 3.18.

```
"Quotation-company_address_section-hidden",
```

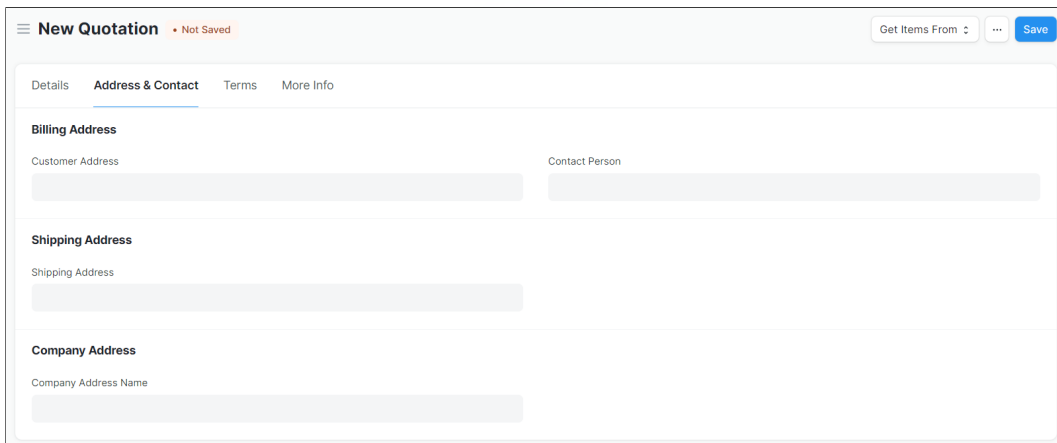
Gambar 3.18. Tampilan pada file *hooks.py*

Setelah menambahkan fixtures tersebut, selanjutnya adalah melakukan *"bench export-fixtures"*. Setelah dilakukan *bench export-fixtures*, maka pada *file property setter* akan otomatis berubah seperti pada gambar 3.19.

```
{
  "default_value": null,
  "doc_type": "Quotation",
  "docstatus": 0,
  "doctype": "Property Setter",
  "doctype_or_field": "DocField",
  "field_name": "company_address_section",
  "is_system_generated": 0,
  "modified": "2024-02-21 19:34:37.136738",
  "module": null,
  "name": "Quotation-company_address_section-hidden",
  "property": "hidden",
  "property_type": "Check",
  "row_name": null,
  "value": "1"
}
```

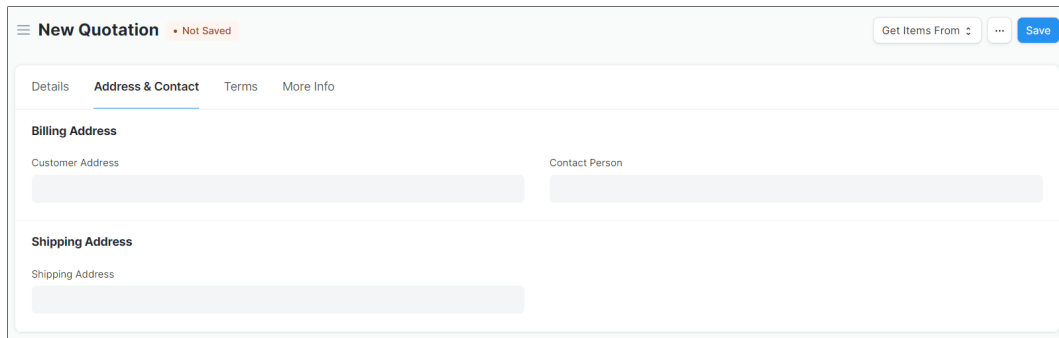
Gambar 3.19. Tampilan pada file property setter

Sehingga, pada mulanya bagian *company address* yang memiliki *field Company Address Name* akan terlihat oleh *user* seperti pada gambar 3.20



Gambar 3.20. Sebelum dilakukan *edit* pada *doctype Quotation*

Kemudian, setelah dilakukan *edit* pada *field company address* yaitu *check* pada *checkbox field hidden*, maka yang semulanya bagian *company address* terlihat, selanjutnya bagian *company address* akan tidak terlihat atau bersifat *hidden* seperti pada gambar 3.21



Gambar 3.21. Hasil edit pada *doctype Quotation*

3.4.3 Pembuatan *Function*

Pada *project* yang dikerjakan selama melakukan magang, selain *custom field* dan menambahkan *property*, pekerjaan yang dilakukan selama melakukan magang adalah membuat sebuah *function* untuk *field* dalam sebuah *doctype*.

Pembuatan *function* dapat dilakukan jika *field* yang dituju sudah tersedia. Pada laporan ini akan difokuskan pada pembuatan *function* untuk *remove decimal in translated words*, *modify job per branch*, *filter field based on master settings*, *investigate and fix issue*, *validation with condition*, *calculate overdays* dan *modify API*.

A. *Remove Decimal in Translated Words*

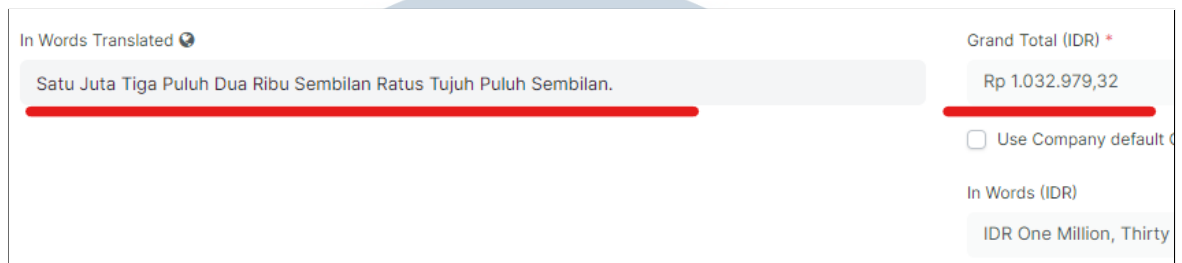
Pada *project* yang dikerjakan selama melakukan magang, saat membuat sebuah *function*, perubahan yang dilakukan adalah mengubah file python yang terhubung dengan *doctype* yang dituju. Dalam hal ini, *function* akan digunakan untuk menghilangkan desimal pada hasil *translate* dari *grand total*. Hal tersebut dapat dilihat pada gambar 3.22.

```
def auto_fill_in_words_translated_on_si(si, method):
    translated_words = num2words(si.grand_total, lang="id").title()
    whole_number, decimal_part = translated_words.split(' Koma ')
    si.in_words_translated = whole_number + "."
```

Gambar 3.22. Tampilan pada file *account_utils.py*

Setelah perubahan tersebut dilakukan dan kemudian disimpan, kegiatan selanjutnya adalah melakukan *run sites* yang dituju. Dalam hal ini, *sites* yang

dilakukan perubahan adalah ERPNext. Sehingga menghasilkan *output* seperti pada gambar 3.23.



Gambar 3.23. Hasil pada *doctype* Sales Invoice

B. *Modify Job per Branch*

Pada *project* yang dikerjakan selama melakukan magang, saat membuat sebuah *schedule job* dapat dijalankan berdasarkan pada *branch*, pekerjaan yang dilakukan adalah menambahkan *function* baru seperti pada gambar 3.24

```
def write_off_invoice():
    master_settings_doc = frappe.get_single("Master Settings")

    for branch in frappe.get_all("Branch"):
        enqueue(
            job_name=f"Writing off from branch {branch.name}",
            is_async=True,
            queue=master_settings_doc.worker_for_branch_background_jobs
            if master_settings_doc.worker_for_branch_background_jobs
            else "long",
            method=sync_write_off,
            branch=branch.name,
            master_settings=master_settings_doc
        )
```

Gambar 3.24. Tampilan pada *file tasks.py*

Pada gambar tersebut, pada mulanya akan mengambil data dari Master Settings dan Branch. Kemudian, data branch yang dimiliki oleh *doctype* Branch akan dilakukan *looping* dan dilakukan *enqueue*. *Enqueue* digunakan untuk menambahkan data pada *queue* dari sebuah data. Pada bagian *enqueue*, *queue* menggunakan *worker* yang ada pada *field worker for branch* dan dengan *method sync write off* yang merupakan sebuah *function*. Sehingga, *job write off* dapat berjalan berdasarkan pada *branch* seperti yang ditunjukkan pada gambar 3.25

Background Jobs			Scheduler: Active
Branch_queue	Writing off from branch [REDACTED]		Queued
Branch_queue	Writing off from branch [REDACTED]		Queued
Branch_queue	Writing off from branch SOLO		Queued
Branch_queue	Writing off from branch [REDACTED]		Queued

Gambar 3.25. Tampilan pada *background jobs*

C. Filter Field based on Master Settings

Pada *project* yang dikerjakan selama melakukan magang, saat membuat sebuah *function* untuk membuat sebuah filter berdasarkan pada *field* pada Master Settings, hal yang dilakukan adalah membuat sebuah *field* yang akan menampung sebuah data. Dalam membuat *field* tersebut, pekerjaan yang dilakukan ditunjukkan pada gambar 3.26

Label	Options
Salesman Department to be Excluded in Quotation	
Type *	
Data	
Name	
Length	
0	
<input type="checkbox"/> Mandatory	
<input type="checkbox"/> Virtual	
<input type="checkbox"/> Index	

For Links, enter the DocType as range. For Select, enter list of Options, each on a new line.

Gambar 3.26. Tampilan pada bagian *edit* Master Settings

Setelah *field* tersebut dibuat pada *edit doctype* Master Settings, maka pada bagian *master_settings.json* akan otomatis membuat sebuah field seperti pada gambar 3.27 dan 3.28

`"salesman_department_to_be_excluded_in_quotation"`

Gambar 3.27. Tampilan pada file *hooks.py*


```
{
  "fieldname": "salesman_department_to_be_excluded_in_quotation",
  "fieldtype": "Data",
  "label": "Salesman Department to be Excluded in Quotation"
},
```

Gambar 3.28. Tampilan pada file hooks.py

Dalam hal ini, *field* tersebut akan digunakan untuk melakukan filter berdasarkan pada *department*. Hal tersebut dapat dilihat pada gambar 3.29

Gambar 3.29. Tampilan pada Master Settings

Setelah *field* tersebut telah berhasil dibentuk, selanjutnya adalah membuat *function* agar *department* yang berada pada *field* tersebut akan digunakan sebagai filter. Hal tersebut dapat dilihat pada gambar 3.30

```
@frappe.whitelist()
@frappe.validate_and_sanitize_search_inputs
def filter_branch_in_quotation(doctype, txt, searchfield, start, page_len, filters):
    if filters and len(filters) and filters[0] and len(filters[0]) == 4:
        branch = filters[0][3]
        if branch:
            excluded_departments = frappe.db.get_value("Master Settings", None,
                "salesman_department_to_be_excluded_in_quotation")
            excluded_departments = excluded_departments.split(',')

            branch_filters = frappe.db.sql(
                """SELECT `tabSales Person`.document_name, `tabSales Person`.sales_person_name
                FROM `tabSales Person`
                WHERE `tabSales Person`.branch = %s AND `tabSales Person`.department NOT IN %s """, (
                    branch,
                    tuple(excluded_departments) if excluded_departments else ()
                ))
            return branch_filters

    return []
```

Gambar 3.30. Tampilan pada file utils.py

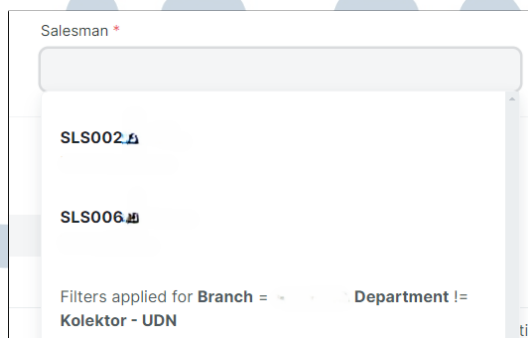
Pada gambar diatas, file utils digunakan untuk membuat *function* agar apabila terdapat sebuah *department* yang di-input pada *field* "Salesman Department to be Excluded in Quotation", maka pada *doctype* *Quotation* khususnya pada *field* *Salesman*, *department* tersebut akan mengecualikan *department* tersebut ketika

sedang menambahkan salesman. Kemudian, terdapat *form scripts* yang digunakan pada *field Salesman*, yaitu *onload*. *Onload* dapat membuat sebuah *form* menjadi memiliki *value onload* atau ketika klik *field Salesman*, maka *filter* tersebut akan bekerja. *Function* tersebut dapat dilihat pada gambar 3.31

```
frappe.ui.form.on("Quotation", "onload", function (frm) {
  frappe.db.get_single_value('Master Settings', 'salesman_department_to_be_excluded_in_quotation')
    .then(salesman => {
      frm.set_query("salesman", function (doc, cdt, cdn) {
        return {
          query: "_sales_app.utils.filter_branch_in_quotation",
          filters: [
            ["Sales Person", "branch", "=", doc.branch],
            ["Sales Person", "department", "!=", salesman.split(',').map(function (s) {
              return s.trim();
            })]
          ],
        };
      });
    });
});
```

Gambar 3.31. Tampilan pada file quotation.js

Setelah *function* tersebut berhasil dijalankan, maka pada *field Salesman* akan seperti pada gambar 3.32. Pada gambar tersebut, *Salesman* akan ditampilkan berdasarkan pada *Salesman* yang terdaftar dan dengan mengecualikan *department* yang telah dimasukkan pada *doctype* *Master Settings*.



Gambar 3.32. Tampilan pada *field Salesman*

D. Investigate and Fix Issue

Pada *project* yang dikerjakan selama melakukan magang, ketika melakukan investigasi untuk sebuah *bug* atau *issue* yang ditemukan, hal yang dilakukan adalah melakukan *restore* data dari *erpNext* atau *middleware*, baik penjualan atau gudang ke *localhost*. *Restore* data tersebut bergantung pada *sites* yang memiliki *issue*

tersebut. Dalam hal ini, *issue* yang ditemukan adalah hilangnya *Sales Invoice ID* dan *Customer ID* setelah sebuah *Sales Invoice* terbentuk. Saat kegiatan *restore* telah berhasil dilakukan, selanjutnya adalah menginput sebuah data dengan *flow* yang berkaitan pada *issue* tersebut. Ketika penyebab dari *issue* atau *bug* tersebut telah ditemukan, *file* yang berkaitan harus dilakukan perubahan seperti pada gambar 3.33

```
e_cash_doc.customer_id = sales_invoice.customer
```

Gambar 3.33. Tampilan pada file selling utils

Setelah *file* tersebut dilakukan perubahan dan disimpan, selanjutnya adalah menjalankan *code* tersebut sehingga dapat menghasilkan *output* seperti pada gambar 3.34

Sales Invoice ID	Custom...	Invoice Outstanding A...
	CS07500...	
	CS07500...	
	CS07500...	

Gambar 3.34. Tampilan pada file selling utils

E. *Modify API*

Pada *project* yang dikerjakan selama melakukan magang, saat ingin mendapatkan sebuah *value* berdasarkan pada API, kegiatan yang dilakukan adalah mengubah file *api.py* dengan menambahkan *fieldname* yang *value*-nya ingin didapatkan. Hal tersebut dapat dilihat pada gambar 3.35 dan 3.36

```
detailed_data = {}
detailed_data.update({
    "address": address,
    "customer_main_channel": customer_doc.customer_main_channel,
    "allow_order_without_pjp": customer_doc.allow_order_without_pjp,
    "customer_division": [],
    "customer_finance": []
})
```

Gambar 3.35. Tampilan pada file api

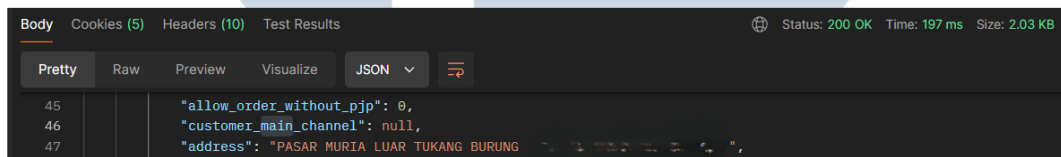
```

for finance_row in customer_doc.customer_finance:
    detailed_data["customer_finance"].append({
        "total_unpaid": finance_row.total_unpaid,
        "credit_limit": finance_row.credit_limit,
        "annual_billing": finance_row.billing_this_year
    })
customer_row.update(detailed_data)

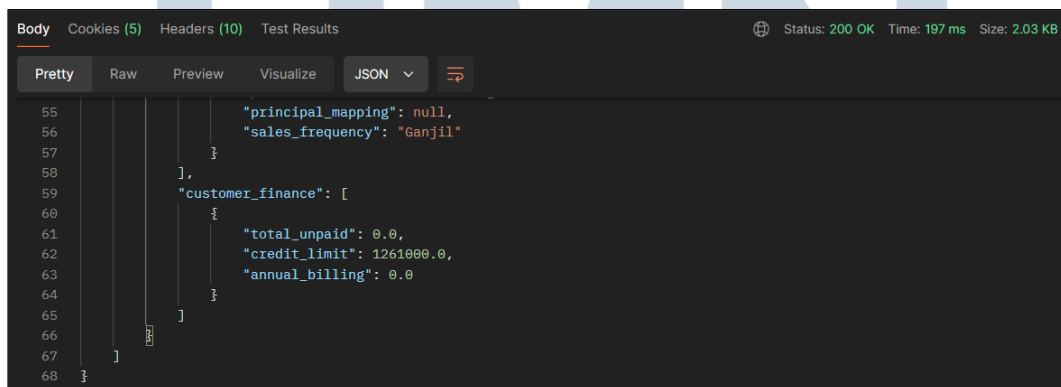
```

Gambar 3.36. Tampilan pada file api

Pada *code* diatas, *value* yang ingin didapatkan berdasarkan dengan API adalah *main channel* dari customer, *allow order without PJP* yang memiliki *value* 0 atau 1. Kemudian pada bagian *customer finance*, *field* yang ingin didapatkan adalah *total unpaid*, *credit limit* serta *annual billing*. Dalam hal ini, *software* yang digunakan dalam mengambil *value API* adalah Postman. Hal tersebut dapat dilihat pada gambar 3.37 dan 3.38



Gambar 3.37. Tampilan Postman pada *field customer main channel* dan *allow order without PJP*



Gambar 3.38. Tampilan Postman pada *customer finance*

F. Add Validation with Conditions

Pada *project* yang dikerjakan selama melakukan magang, saat membuat sebuah validasi berdasarkan pada sebuah kondisi, pekerjaan yang dilakukan adalah

mengubah *function* validasi apabila sebuah *sales invoice* memiliki *invoice*. Hal tersebut dapat dilihat pada gambar 3.39

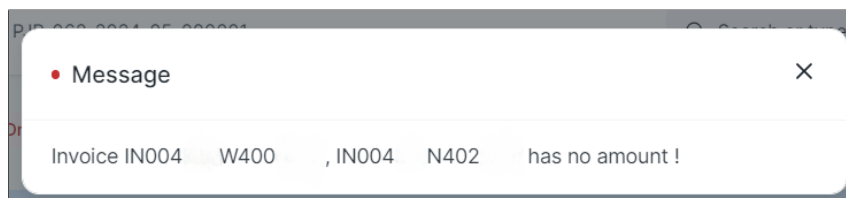
```
def validate_invoice_outstanding(self):
    if self.type == "collection":
        sales_invoice_list = []

        for item_row in self.customer_stop:
            if item_row.invoice_amount is None or item_row.invoice_amount <= 0:
                sales_invoice_list.append(item_row.sales_invoice_id)

        if len(sales_invoice_list) > 0:
            frappe.throw(
                _(f"Invoice {', '.join(sales_invoice_list)} has no amount !")
            )
```

Gambar 3.39. Tampilan pada *file permanent journey plan*

Pada gambar tersebut, *function* akan bekerja apabila tipe dari *permanent journey plan* adalah *Collection*. Sehingga, alur dari *code* tersebut pada awalnya akan melakukan inisialisasi list kosong. Kemudian, setiap baris pada *customer stop* akan dilakukan *looping* dengan kondisi jika *invoice amount* tidak ada atau *invoice amount* kurang dari sama dengan 0, maka *sales invoice ID* tersebut akan ditambahkan pada list yang telah dilakukan inisialisasi. Sehingga, apabila list tersebut panjangnya lebih besar dari 0, maka akan mengeluarkan *pop-up message* "Invoice has no amount !" seperti pada gambar 3.40



Gambar 3.40. Tampilan *pop-up message*

G. Calculate Overdays

Pada *project* yang dikerjakan saat magang, saat membuat sebuah *function* yang digunakan untuk menghitung *overdays* untuk sebuah *sales invoice*, pekerjaan yang dilakukan adalah menghitung perbedaan antara tanggal ketika *sales invoice* tersebut dibentuk dengan *due date* dari *sales invoice* yang dibuat. Hal tersebut dapat dilihat pada gambar 3.41

```
function overdays_calculate(frm) {
  if (frm.doc.status == "Overdue") {
    frm.set_value("overdays", frappe.datetime.get_day_diff(
      frappe.datetime.nowdate(), frm.doc.due_date))
  }
}
```

Gambar 3.41. Tampilan pada file sales invoice

Pada gambar tersebut, apabila status dari *sales invoice* adalah *Overdue*, maka *field overdays* akan muncul dibawah *due date*. *Value* dari *field overdays* didapatkan dengan mengurangi tanggal terkini dengan tanggal *due date* dari *sales invoice* tersebut. Kemudian, *field* tersebut akan muncul ketika dilakukan *reload sales invoice* dengan menggunakan *form script "onload post render"* seperti pada gambar 3.42

```
onload_post_render: function (frm) {
  overdays_calculate(frm)
}
```

Gambar 3.42. Form script untuk field overdays

Sehingga, pada *sales invoice* dengan status *Overdue* akan menjadi seperti pada gambar

The screenshot shows a web interface for a sales invoice. At the top, there is a red status indicator 'Overdue'. Below it are tabs for 'Details', 'Payments', 'Contact & Address', 'Terms', 'More Info', and 'Connections'. The main form contains several fields:

Customer	Date *	23-04-2024
Customer Name	Posting Time	22:31:45
Is Synchronizing	Payment Due Date *	23-04-2024
Linking Sales Invoice to Tax Invoice Number	Overdays	35

Gambar 3.43. Tampilan pada sales invoice

3.5 Hasil Implementasi

Setelah *project* dikerjakan dengan baik dan sesuai dengan kebutuhan dan keinginan klien, maka *project* ini dapat digunakan oleh klien dari PT Sumber Inovasi Informatika. Hal tersebut dapat dilihat pada lampiran 7 yaitu Surat Pernyataan pengerjaan Project yang telah disetujui serta telah ditandatangani oleh *supervisor* pada pelaksanaan kerja magang pada PT Sumber Inovasi Informatika.

3.6 Kendala dan Solusi yang Ditemukan

3.6.1 Kendala

Kendala yang dihadapi selama melakukan magang di PT Sumber Inovasi Informatika adalah sebagai berikut.

1. Kurang memahami mengenai *framework* yang digunakan selama magang di PT Sumber Inovasi Informatika.
2. Terdapat banyak *file* yang harus dipahami dalam melakukan pengembangan ERP dan *middleware*.
3. Sulitnya menggabungkan antara ERP dengan *middleware* penjualan dan *middleware* gudang.

3.6.2 Solusi

Berdasarkan pada kendala yang dihadapi selama melakukan magang di PT Sumber Inovasi Informatika adalah sebagai berikut.

1. Melakukan *pairing* atau *discuss* bersama dengan *Senior Developer* dan *Product Owner* serta membaca dokumentasi melalui *website* resmi Frappe mengenai permasalahan yang ditemukan
2. Membaca *coding* milik *Senior Developer* atau melalui *google* yang serupa dengan permasalahan yang ditemukan selama bekerja