

BAB 3

PELAKSANAAN KERJA MAGANG

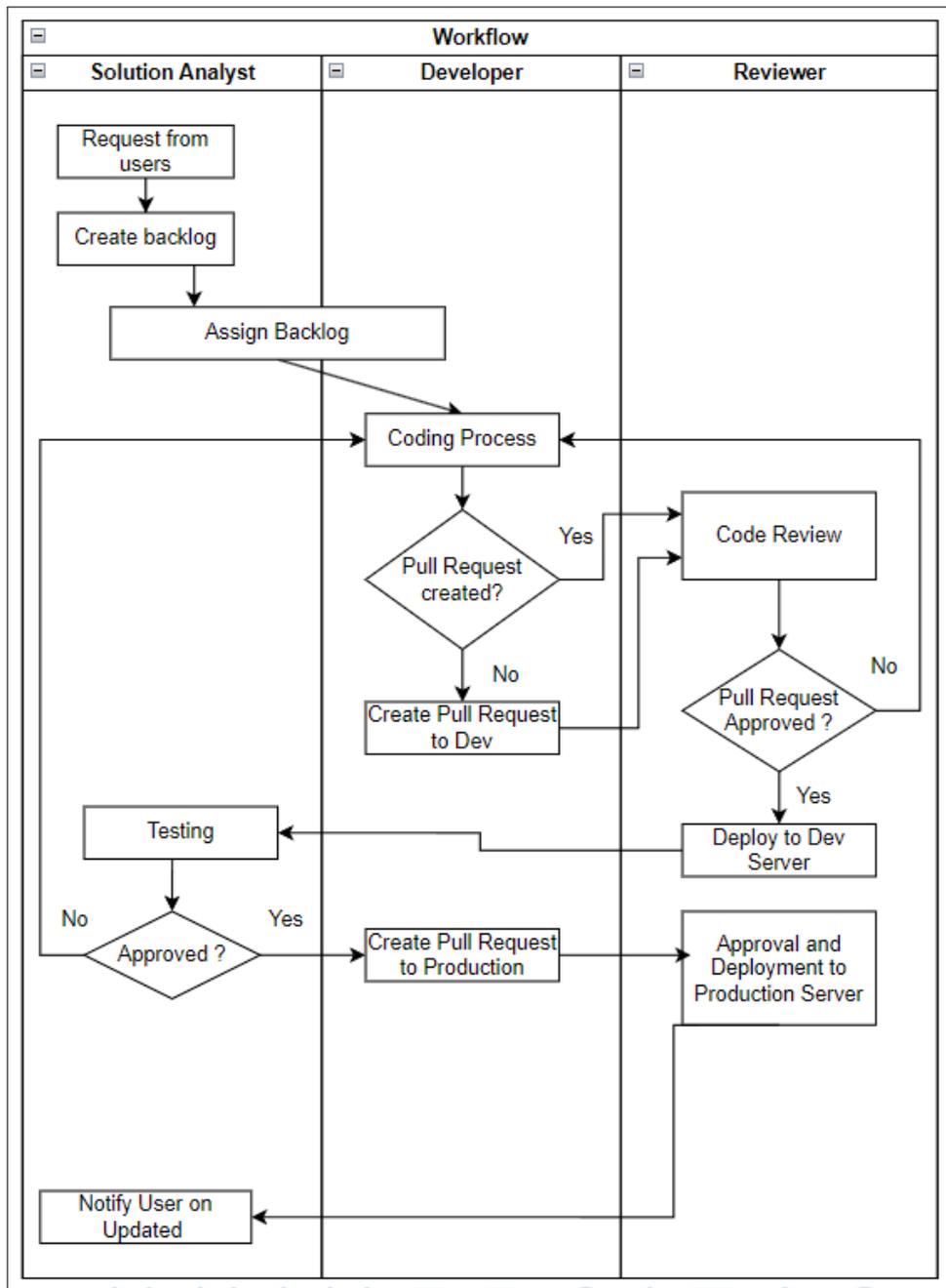
3.1 Kedudukan Dalam Perusahaan

Selama periode pemagangan mahasiswa magang ditugaskan di Departemen *Enterprise Technology* CITIS (Corporate IT & IS) di bawah kepemimpinan Arnold Rompas selaku *Tech Manager*. Selama magang, mahasiswa magang bertugas sebagai *Intern Junior Software Engineer* dan bertanggung jawab kepada *Senior Software Engineer* bernama Ritz Moondrian. Ritz Moondrian sekaligus menjadi mentor dan supervisi bagi mahasiswa magang. Mahasiswa magang mempertanggungjawabkan pekerjaannya kepada supervisi, yang selanjutnya supervisi bertanggung jawab kepada manajer. Demikian pula, tugas-tugas diberikan oleh manajer departemen melalui supervisi hingga sampai kepada mahasiswa magang.

Proses penugasan dan alokasi proyek di CITIS dimulai dengan penyampaian semua proyek yang akan dilakukan kepada direktur CITIS. Kemudian, direktur berkomunikasi dengan *general manager* divisi terkait untuk meneruskan informasi proyek. Selanjutnya, *general manager* menyampaikan proyek kepada *manager* dari departemen yang bertanggung jawab atas pengembangan proyek tersebut. Setelah menerima informasi proyek, *Department Manager* mendistribusikan tugas-tugas melalui rapat departemen bersama anggota departemen.

3.2 Pekerjaan yang Dilakukan

Selama melakukan kerja magang di CITIS Kompas Gramedia sebagai *software engineer*, pekerjaan yang dilakukan adalah melakukan kustomisasi fitur-fitur yang telah ada, fitur-fitur tersebut ditulis dalam bahasa pemrograman Python dengan *framework* Odoo. Selama pemagangan pekerjaan dilakukan dengan alur sebagai berikut:



Gambar 3.1. Alur kerja dalam *Corporate IT & IS*

UNIVERSITAS
MULTIMEDIA
NUSANTARA

Gambar 3.1 menggambarkan alur dalam pelaksanaan kerja magang, dan di bawah ini adalah penjelasan mengenai ilustrasi tersebut:

1. Input Backlog

- (a) Analis yang bertugas untuk menghubungkan antara *developer* dengan *user* akan melakukan *input backlog-backlog* ke dalam Azure DevOps - Boards berdasarkan kebutuhan yang telah didiskusikan bersama *user*.
- (b) Penugasan *developer* untuk mengerjakan *backlog* yang ada dapat dilakukan dengan dua cara yaitu:
 - Ditunjuk langsung oleh analis berdasarkan spesialisasi tiap *developer* dan pembahasan yang telah dilakukan sebelumnya
 - Melakukan *self-assign*. dengan mengambil *backlog* langsung di Azure Devops

2. Pekerjaan Coding

- (a) *Developer* mengerjakan *backlog* yang telah dibuat oleh analis sesuai dengan deskripsi dan ketentuan yang diuraikan oleh analis.
- (b) apabila terdapat hal yang belum jelas dan perlu diklarifikasi *developer* dapat bertanya kepada analis atau tim teknis yang terdiri dari *developer* lain.
- (c) Setelah *developer* selesai melakukan *coding* maka *textitdeveloper* akan melakukan *commit* dan *pull request* ke *branch dev* untuk dilakukan *review*

3. Code Review

- (a) *Code* yang telah di *pull request* akan di *review* oleh para *reviewer* untuk di cek kesesuaian dan keefektifan dari *code* tersebut.
- (b) Apabila terdapat catatan dari *reviewer* maka *developer* wajib untuk memperbaiki *code* sesuai dengan permintaan dari *reviewer*
- (c) Apabila pekerjaan sudah disetujui, maka *code* tersebut akan dilakukan *deployment* menuju *development server* untuk dilakukan pengujian oleh pihak-pihak terkait.

3.3 Uraian Pelaksanaan Magang

Pelaksanaan kerja magang diuraikan pada Tabel 3.1:

Tabel 3.1. Uraian pekerjaan yang dilakukan per minggu

| Minggu ke - | Pekerjaan yang dilakukan |
|-------------|---|
| 1 - 2 | Mempelajari python dan odoo |
| 3 | Mempelajari odoo dan mempelajari mengenai <i>sprint</i> rutin dari <i>Enterprise Technology</i> serta mengikuti <i>sprint meeting</i> pertama |
| 4 | Mempelajari proses bisnis di Kompas Gramedia dan mengikuti <i>sprint</i> |
| 5 | <ul style="list-style-type: none">• Mengerjakan <i>backlog</i> menghapus <i>open & edit access</i> di SPT masa• Mengerjakan <i>backlog Enable multi-company selection in generate report</i> pada daftar terminasi cadangan untuk memungkinkan pemilihan banyak <i>company</i> sekaligus pada laporan daftar terminasi cadangan |
| 6 | <ul style="list-style-type: none">• Mengerjakan <i>backlog</i> untuk mengubah perhitungan <i>months of service</i> dari karyawan kompas.• Mengerjakan <i>backlog</i> menambahkan <i>field spouse name</i> di bawah <i>field spouse_identification_id</i> pada daftar karyawan .• Mempelajari alur kerja pada <i>Corporate IT & IS</i> dan sistematisasi <i>backlog</i> yang ditampilkan pada azure devops |
| 7 | Melakukan eksplorasi mandiri pada <i>source code</i> yang terdapat pada modul <i>kg_payroll</i> |

Tabel 3.1 Uraian pekerjaan yang dilakukan per minggu (lanjutan)

| Minggu ke - | Pekerjaan Yang Dilakukan |
|-------------|--|
| 8 | <ul style="list-style-type: none"> • Melakukan eksplorasi mandiri pada <i>source code</i> yang terdapat pada modul kg_payroll • Melakukan konsultasi bersama <i>solution analyst</i> mengenai alur <i>backlog</i> dan <i>backlog</i> yang bisa diambil berkaitan dengan fitur <i>payroll</i> |
| 9 | <ul style="list-style-type: none"> • Melakukan eksplorasi mandiri pada codingan yang terdapat pada modul kg_payroll • Mengerjakan <i>backlog</i> untuk menambahkan pengaturan baru mengenai penggunaan <i>sub area</i> dan menambahkan <i>field spouse name</i> pada <i>company additional setting</i> |
| 10 | <ul style="list-style-type: none"> • Melanjutkan <i>backlog</i> untuk menambahkan pengaturan baru mengenai penggunaan <i>sub area</i> dan <i>field spouse name</i> pada <i>company additional setting</i> • Mengikuti <i>sprint meeting</i> dari <i>enterprise technology</i> |
| 11 | <ul style="list-style-type: none"> • Mengerjakan <i>backlog</i> mengganti <i>visible condition</i> dari <i>field spouse name</i> supaya <i>field spouse name</i> tidak bisa diisi pada kondisi tertentu • Mengerjakan <i>backlog</i> membalik posisi <i>field is all company</i> dan <i>field multiselection company</i> untuk meningkatkan pengalaman pengguna. |

Tabel 3.1 Uraian pekerjaan yang dilakukan per minggu (lanjutan)

| Minggu ke - | Pekerjaan Yang Dilakukan |
|-------------|---|
| 12 | <ul style="list-style-type: none"> • Melakukan pembelajaran mandiri dengan mengerjakan projek-projek dari udey dan mengambil sertifikat penguasaan odoo dari udey. • Melakukan setup odoo dikarenakan penambahan <i>jobdesk</i> untuk membantu <i>engineer</i> pada KG-ERP. • Mengerjakan backlog untuk menambahkan pengaturan lokasi di <i>company additional setting</i> untuk <i>purchase organization group</i> |
| 13 | <ul style="list-style-type: none"> • Melanjutkan <i>backlog</i> untuk menambahkan pengaturan lokasi di <i>company additional setting</i> untuk <i>purchase organization group</i> • Memperbaiki munculnya <i>error</i> ketika membuat <i>purchase request</i> |
| 14. | <ul style="list-style-type: none"> • Melakukan penyesuaian agar pengguna dapat mengedit permintaan <i>maintenance request</i> selama permintaan tersebut belum di <i>approve</i>. • Melakukan <i>adjustment</i> pada <i>product list</i> yang muncul di master data agar menampilkan produk dengan kategori <i>equipment</i> dan <i>food & beverage</i>. |
| 15. | <ul style="list-style-type: none"> • Mengerjakan <i>backlog</i> dengan tugas untuk menghapus <i>user restriction access</i> pada <i>form utilities usage line</i> sehingga semua <i>user</i> bisa mengakses <i>form</i> tersebut. • Kustomisasi akses tombol <i>action posting</i> pada <i>Receipt debit order</i> agar <i>user</i> yang memiliki akses <i>Head Office System</i> dapat langsung melakukan <i>posting debit order</i> |

Berdasarkan tabel 3.1 pekerjaan-pekerjaan yang telah disebutkan dapat dikelompokkan sebagai berikut:

3.3.1 Pembelajaran

Selama masa pembelajaran yang dimulai dari minggu pertama hingga minggu keempat peserta magang mempelajari hal-hal berikut:

A. Pengenalan Framework Odoo ERP

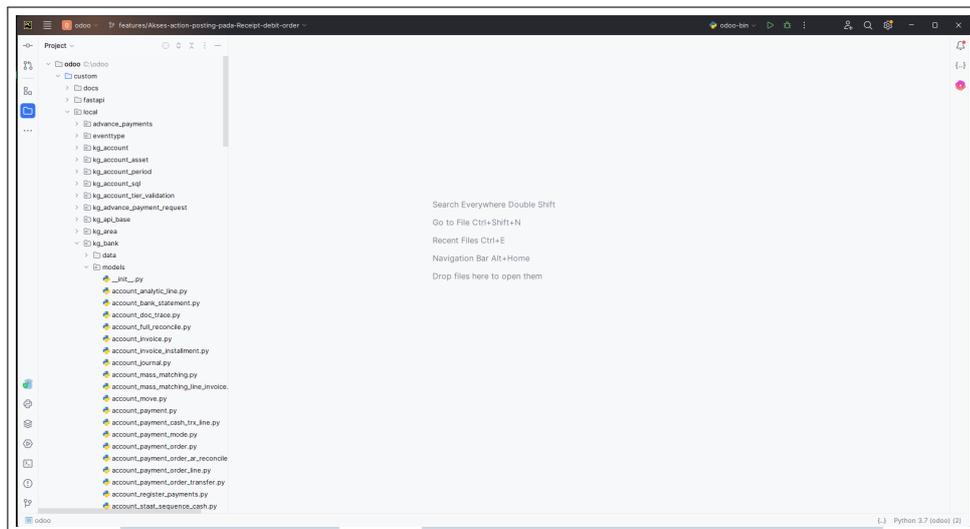
Kompas Gramedia memberikan waktu selama satu bulan untuk melakukan pembelajaran mandiri mengenai *framework* odoo, terdapat panduan-panduan yang diberikan oleh *supervisor* agar pembelajaran menjadi lebih terarah berupa *learning path* yang berisi materi-materi pembelajaran mengenai Odoo yang harus dipelajari setiap minggunya.

B. Alur Kerja di dalam CITIS

Setelah melakukan pembelajaran mengenai konsep-konsep dasar *framework* Odoo, pembelajaran selanjutnya adalah mempelajari alur kerja di dalam CITIS yang menerapkan metodologi *sprint agile* dalam pengembangan *project*.

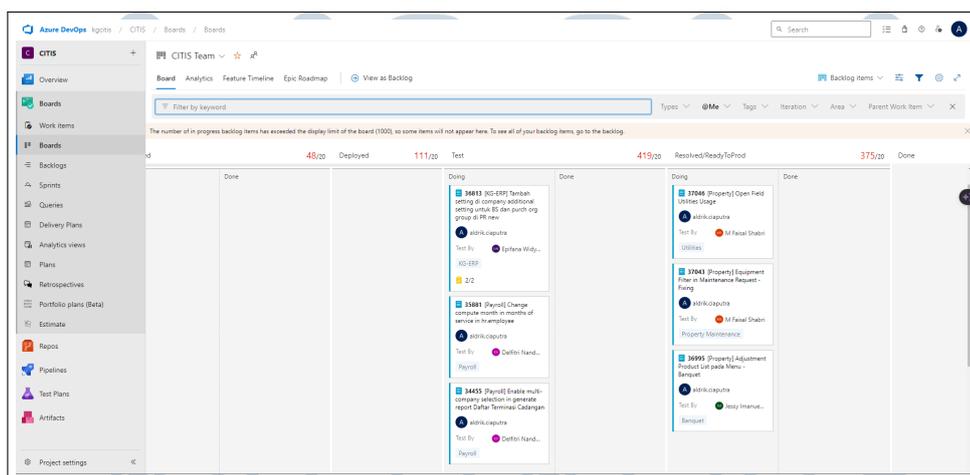
C. Tools yang digunakan

Pengenalan juga dilakukan terhadap berbagai *tools* yang digunakan, salah satunya adalah PyCharm sebagai *code editor*. Dalam pengenalan tersebut diajarkan mengenai fitur-fitur yang dimiliki oleh PyCharm dan bagaimana fitur-fitur tersebut dapat mempermudah *engineer* untuk melakukan *coding* pada *framework* Odoo. Tampilan UI (*User Interface*) PyCharm dapat dilihat pada Gambar 3.2.



Gambar 3.2. Tampilan UI PyCharm

Selain itu, juga dikenalkan pada Azure DevOps sebagai media kolaborasi utama, Azure DevOps memiliki fitur bernama Azure Boards yang mana fitur ini digunakan oleh para analis untuk melakukan *posting backlog-backlog* apa saja yang bisa dikerjakan oleh para developer sekaligus juga untuk memantau *progress* dari tiap *backlog* yang di *posting*. Para developer juga menggunakan Azure DevOps untuk melakukan *pull request* terhadap kode yang telah dibuat. Tampilan UI Azure DevOps dapat dilihat pada Gambar 3.3.



Gambar 3.3. Tampilan UI Azure DevOps

3.3.2 Pengerjaan Backlog

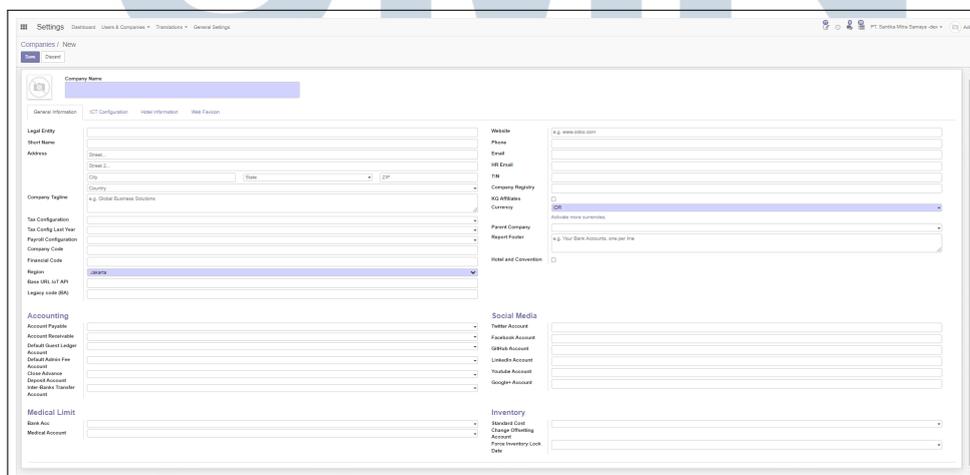
Selama periode magang *backlog-backlog* yang dikerjakan terfokus pada modul *payroll* dan *property*, untuk lebih jelasnya berikut uraian beberapa *backlog-backlog* yang pernah dikerjakan.

A. Backlog Yang Dikerjakan Pada Module Payroll

Module payroll merupakan *module* yang dirancang untuk melakukan pembayaran gaji karyawan, *backlog-backlog* yang dikerjakan pada *module payroll* adalah sebagai berikut:

- Menghapus Akses *Open* dan *Create Company* di Formulir SPT Masa

Backlog ini merupakan tugas pertama yang peserta magang terima, tujuan dari *backlog* ini yaitu menghapus akses *open* dan *edit field company* pada *form* SPT Masa, SPT Masa (Surat Pemberitahuan Masa) merupakan dokumen yang digunakan Wajib Pajak (WP) untuk melaporkan kegiatan perpajakan secara bulanan, dokumen SPT Masa ini wajib diisi oleh pihak perusahaan sesuai dengan ketentuan dari undang-undang PPh Pasal 21 yang berbunyi pajak penghasilan dipotong dari gaji karyawan oleh perusahaan, dan perusahaan wajib melaporkan dan membayarnya. Pada *form* pengisian SPT Masa terdapat opsi untuk membuka dan membuat daftar *company* baru yang mana hal tersebut seharusnya tidak boleh dilakukan sebab *form* pembuatan *company* baru seharusnya hanya bisa dibuat melalui menu *setting* kemudian akses *form* pembuatan *company* baru seperti yang dapat dilihat pada Gambar 3.4.

The image shows a screenshot of a web-based application interface for creating a new company. The page is titled 'Companies / New' and 'Company Menu'. It features a sidebar with navigation options like 'General Information', 'Tax Configuration', 'Social Media', and 'Inventory'. The main content area is divided into several sections: 'Legal Entity' with fields for Short Name, Address, and Company Register; 'Tax Configuration' with fields for Tax Code, Company Code, and Region; 'Social Media' with fields for Facebook, GitHub, LinkedIn, and Google+ accounts; and 'Inventory' with fields for Stock Code, Opening Inventory, and Inventory Lock Date. The interface is clean and professional, typical of a business management system.

Gambar 3.4. Formulir pembuatan *company* baru

Untuk mengatasi permasalahan tersebut maka opsi buka dan edit pada *field company* yang dapat dilihat pada Gambar 3.5 harus dihapus. *field* tersebut hendak dimodifikasi agar pengguna tidak dapat mengedit atau membuat daftar *company* baru melalui formulir SPT Masa, agar pengguna hanya dapat memilih untuk melaporkan SPT Masa dari *company* yang telah didaftarkan sebelumnya.

The screenshot shows the Odoo SPT Masa form for PT. Santika Mitra Samaya -dev. The form includes a header with navigation tabs like 'Payroll', 'Payslips Batches', and 'Employee Payslips'. Below the header, there are buttons for 'Save', 'Discard', and 'Confirm'. The main form area contains several fields: 'Period' (January 2024), 'Salary Structure' (Percentage (%)), 'Company' (PT. Santika Mitra Samaya -dev), and 'Tax Object Code' (21-100-01). The 'Company' field is highlighted with a red box, indicating it is the focus of the modification.

Gambar 3.5. *Field company* pada formulir SPT Masa

penyelesaian *backlog* ini dilakukan dengan menambahkan *options no_create_edit: true* dan *no_open: true* pada kodingan yang sudah ada sebelumnya seperti yang tampak pada kode 3.1.

```
< field name = " company_id " attrs = " { 'readonly ' : [ ( ' state ' , '=' , ' confirmed ' ) ] } " options = " { 'no_create_edit ' : True , 'no_open ' : True } " />
```

Kode 3.1: Kode untuk hapus akses buka dan edit

- Mengganti Perhitungan *Months of Service* Karyawan

Sebelumnya perhitungan *months of service* karyawan dihitung mulai dari tanggal karyawan tersebut masuk bekerja, namun hal ini menimbulkan kerumitan bagi *human resource* dalam melakukan kalkulasi *months of service* karyawan sehingga perhitungan *months of service* karyawan perlu diubah dari yang sebelumnya dihitung berdasarkan tanggal karyawan tersebut masuk diubah menjadi dihitung berdasarkan bulan dimulainya karyawan bekerja hal ini dilakukan untuk membuat perhitungan *months of service* karyawan menjadi lebih *simple*. Penyelesaian *backlog* ini dilakukan dengan mengganti nilai variabel yang akan di *input* ke *function calculate months_of_service* dengan memodifikasi nilai tanggal dari variabel *from_dt* ke tanggal satu seperti yang dapat dilihat pada kode 3.2, sehingga format penanggalan yang dimasukkan ke dalam variabel yaitu YYYY-MM-01.

```

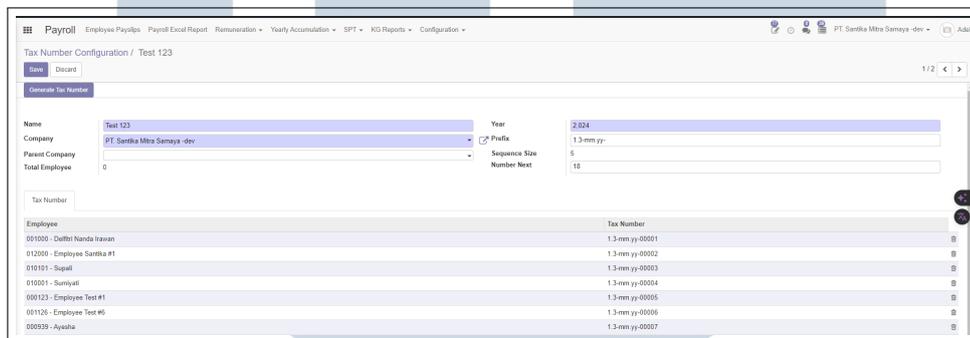
1 from_dt = fields . Date.from_string (employee .
2 initial_employment_date [:8] + " 01 "

```

Kode 3.2: Kode untuk mengubah tanggal menjadi tanggal satu

- Mengganti *Employee Tax Number Identifier*

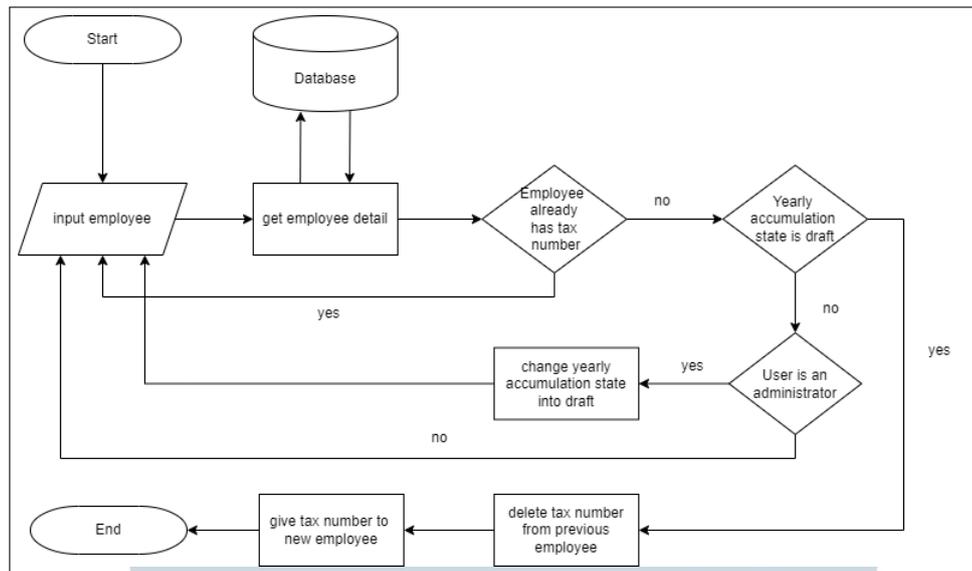
Pada sistem perpajakan yang terdapat di modul *payroll* Kompas Gramedia setiap karyawan Kompas memiliki *tax number identifier* yang unik, *tax number identifier* tersebut dapat dikonfigurasi dengan mengakses menu *tax number configuration* seperti yang terlihat pada Gambar 3.6.



Gambar 3.6. Menu *tax number configuration*

Terdapat masalah pada sistem *tax number configuration* saat yaitu konfigurasi tidak bekerja dengan semestinya, ketika *employee* yang diberikan *tax number* diganti, seharusnya *tax number identifier* milik *employee* tersebut juga diganti, namun hal itu tidak terjadi, hal ini membuat *tax number employee* tidak bisa diganti dari menu *tax number configuration*. Maka dari itu penugasan dari *backlog* ini adalah untuk membuat konfigurasi tersebut bekerja sehingga ketika *employee* diganti maka *tax number* yang diberikan pada *employee* tersebut juga ikut berubah. Alur kerja dari *tax number configuration* dapat dilihat seperti Gambar 3.7.

UNIVERSITAS
MULTIMEDIA
NUSANTARA



Gambar 3.7. Menu alur *tax number configuration*

Ketika pengguna mengganti *employee* dari menu *tax number configuration* maka sistem akan meminta data tentang *employee* baru tersebut, kemudian dilakukan pengecekan apakah *employee* tersebut sudah memiliki *tax number* atau belum, jika *user* sudah memiliki *tax number identifier* maka pengguna tersebut tidak dapat di *input*, jika belum maka akan dilakukan pengecekan terhadap status *yearly accumulation* dari *employee* jika status *yearly accumulation* adalah *draft* maka hapus *tax number identifier* dari *employee* lama dan berikan *tax number identifier* dari *employee* lama itu ke *employee* baru jika status *yearly accumulation* bukan *draft* maka pengecekan selanjutnya dilakukan untuk mengecek apakah pengguna yang masuk ke sistem adalah seorang administrator atau bukan, jika iya maka pengguna dapat mengganti status *yearly accumulation* dari *employee* ke *state draft* agar bisa diganti dan melakukan *input* ulang di *tax number configuration*, jika pengguna bukan administrator maka pengguna harus memasukkan *input employee* lain karena pengguna tidak bisa mengubah status *yearly accumulation*.

backlog ini diselesaikan dengan melakukan *coding* yang dapat dilihat pada kode 3.3.

```
1 @api.multi
2     def write(self, vals):
3         for record in self:
4             if record.employee_id.id != vals.get('employee_id',
5             False):
6                 yearly_accumulation_old_employee = self.env['hr.
7                 payslip.yearly.accumulation'].search([
8                     ('employee_id', '=', record.employee_id.id),
9                     ('company_id', '=', record.company_id.id),
10                    ('year', '=', record.year), ], limit=1
11                )
12                yearly_accumulation_new_employee = self.env['hr.
13                payslip.yearly.accumulation'].search([
14                    ('employee_id', '=', vals.get('employee_id',
15                    False)),
16                    ('company_id', '=', record.company_id.id),
17                    ('year', '=', record.year)], limit=1
18                )
19                if yearly_accumulation_old_employee.state == '
20                draft' and yearly_accumulation_new_employee.state == 'draft':
21                    yearly_accumulation_old_employee.write({'
22                    tax_yearly_number': ''})
23                    yearly_accumulation_new_employee.write({'
24                    tax_yearly_number': record.tax_number})
25                else:
26                    raise ValidationError('Cannot update employee
27                    for closed or release state')
28                return super(KGPayrollTaxNumberLineConfiguration, self).
29                write(vals)
```

Kode 3.3: Konfigurasi *tax number*

Coding tersebut bekerja didasarkan pada alur yang digambarkan pada Gambar 3.7, *coding* tersebut berfungsi untuk mendapatkan data *employee* lama dan *employee* baru yang selanjutnya melakukan pengecekan terhadap kondisi *state* dari *yearly_accumulation* jika kondisi terpenuhi maka *tax number* pada *employee* lama akan dikosongkan kemudian diberikan kepada *employee* baru.

B. Backlog Yang Dikerjakan Pada Module Property

- Menambahkan Pengaturan Lokasi *Bank Statement* di *Company Additional Setting*

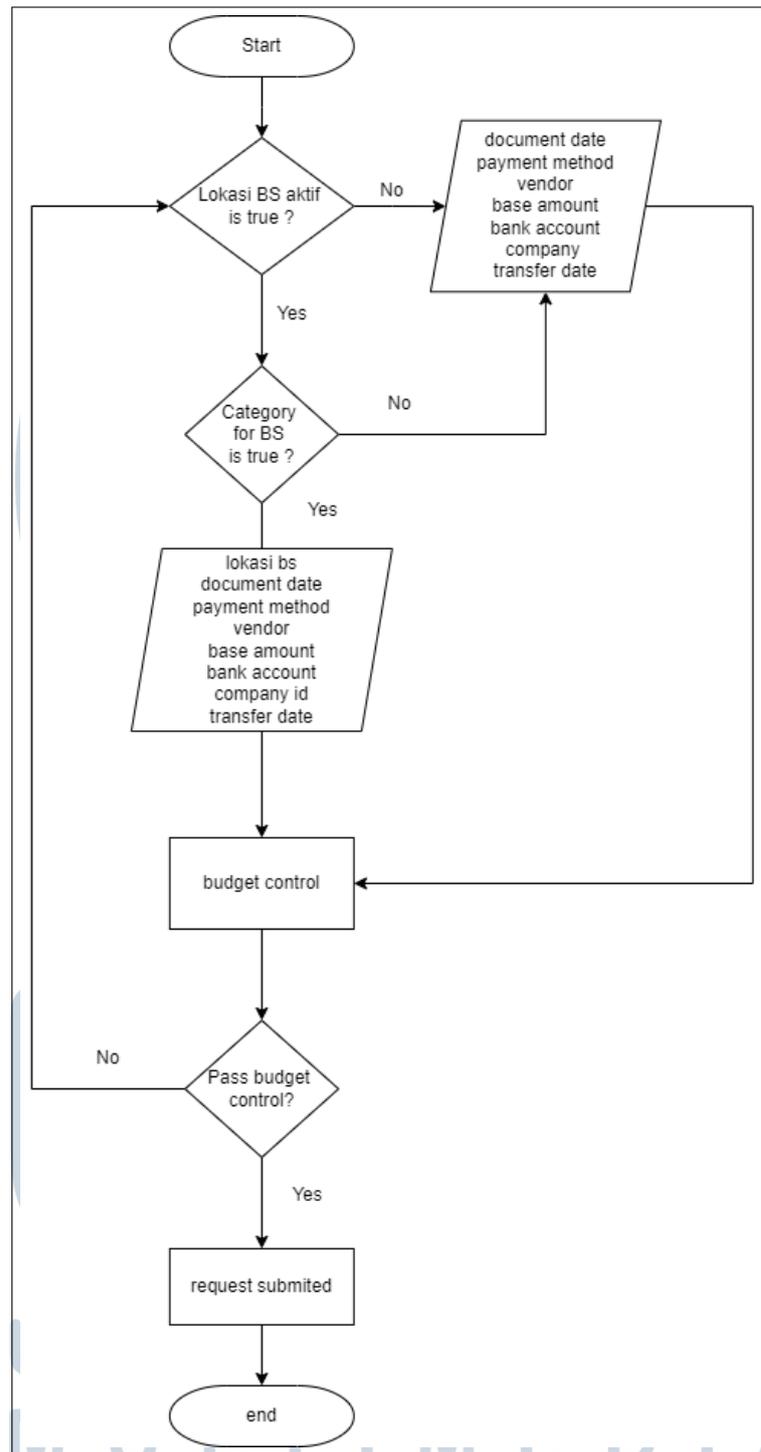
Pada formulir pembuatan *advance payment request* perusahaan dapat membuat permintaan untuk melakukan pembayaran baik itu pembayaran dari *vendor* atau pembayaran ke sebuah *vendor*. pada sistem *advance payment request* terdapat sebuah permasalahan yaitu saat melakukan pembayaran pihak perusahaan tidak mengetahui lokasi cabang bank yang mengurus aktivitas transaksi, mengetahui lokasi *bank statement* merupakan hal yang dibutuhkan untuk mengetahui bank mana yang bertanggung jawab untuk melakukan aktivitas transaksi.

Menu *advance payment request* dapat di akses dengan cara, pertama pengguna masuk ke menu *account payable* kemudian klik menu *advance payment request* yang terdapat di *navigation bar*, selanjutnya pengguna dapat memilih untuk membuat atau mengedit *advance payment request* yang ada. Penulis ditugaskan untuk menambahkan *field* baru ke dalam *form* tersebut.

| | | | |
|------------------|------------------------------|---------------------|-----|
| Document Date | 06/04/2024 | Company | KGX |
| Category | Uang Muka Kerja BS | Project | |
| Advance Payment | 1400000 Uang Muka Kerja (BS) | Purchase Order | |
| Account | | Budget | 0 |
| Payment Method | Bank Transfer - EBanking | Faktur Pajak Date | |
| Vendor | | Nomor SKB | |
| Base Amount | Rp0 | Value Date | |
| Payment Amount | Rp 0 | Posting Date | |
| PIC BS | | Source | |
| Lokasi BS | | Doc No | |
| Bank Account | | Payment Order | |
| Partner | | Request | |
| Account Holder | | Payment Transaction | |
| Name | | | |
| Description | | | |
| Transfer Request | | | |
| Date | | | |
| Estimation Date | | | |
| Remaining Amount | Rp 0 | | |
| Payment Type | Send Money | | |

Gambar 3.8. Field lokasi bank statement

Alur yang perlu dilakukan oleh pengguna untuk mengisi formulir digambarkan dalam *flowchart* berikut:



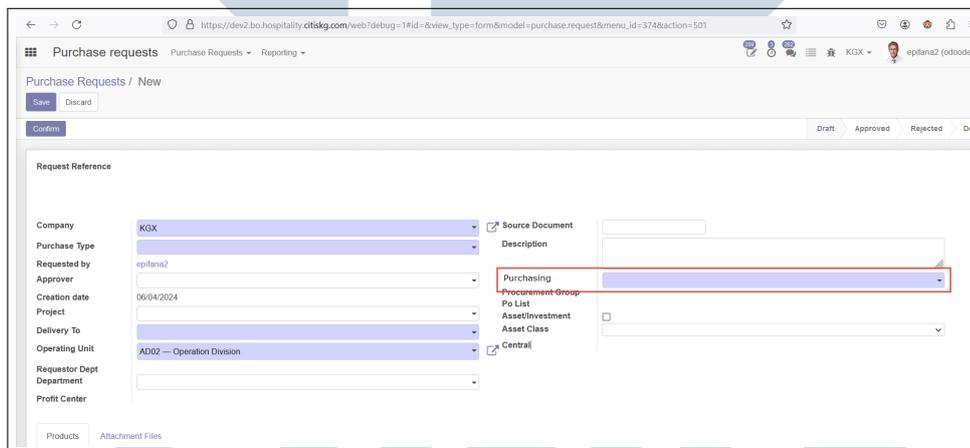
Gambar 3.9. Flowchart advance payment request

Proses dimulai dengan pengguna mengakses formulir *advance payment request*, saat pengguna masuk ke formulir sistem akan mengecek apakah pengaturan lokasi BS di *company additional setting* bernilai *true* atau *false*, jika *false*, maka pengguna diwajibkan untuk mengisi informasi seperti:

document date, payment method, vendor, jumlah dasar, rekening bank, bank account, company transfer date. Sedangkan jika pengaturan lokasi BS bernilai *true* maka ada satu tambahan *field* yang harus diisi pengguna yaitu *field* lokasi bs. Setelah formulir *advance payment request* disimpan maka proses selanjutnya adalah *budget control* jika permintaan lulus *budget control* maka selanjutnya *advance payment* akan dikirimkan kepada bagian *finance*.

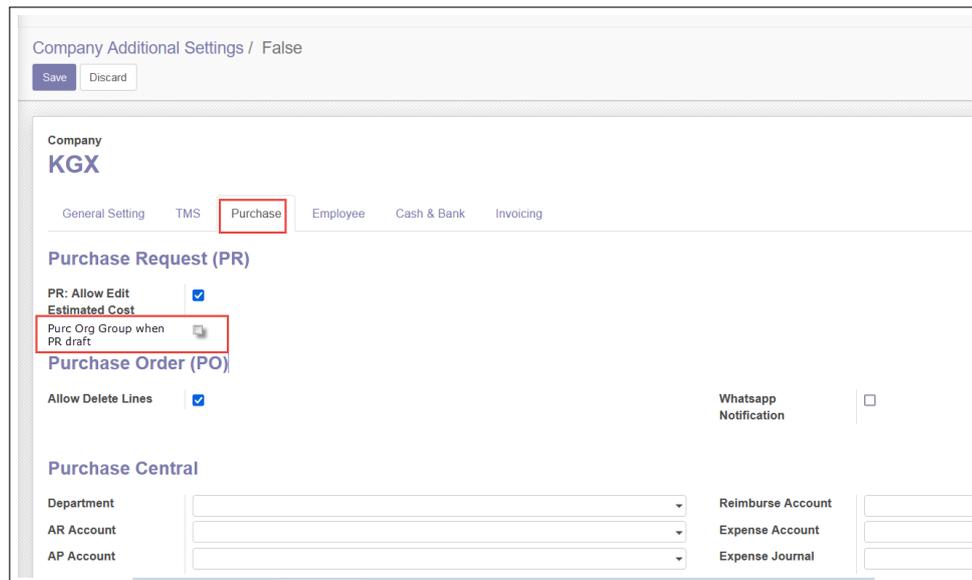
- Mengaktifkan *Group Purchasing*

Pada menu *advance payment request* terdapat *field purchasing org* yang tidak bisa diisi karena selalu disembunyikan dari tampilan pengguna, *field* tersebut selalu disembunyikan karena terdapat *error* pada sistem pembelian secara grup, namun saat backlog ini diterima sistem pembelian *organization group* tersebut sudah diperbaiki sehingga sudah dapat digunakan oleh pengguna, *field* tersebut ingin ditampilkan pada formulir ditempat yang ditunjukkan oleh kotak merah seperti yang dapat dilihat pada Gambar 3.10.



Gambar 3.10. *Field purchase organization*

field tersebut ingin dimodifikasi agar tampil di *user interface* pengguna dan dapat diisi oleh pengguna dengan aturan bahwa pada *field Purc Org Group When PR Draft* yang terdapat pada *company additional setting* bernilai *true* seperti pada Gambar 3.11.



Gambar 3.11. *Setting organization group*

Untuk menyelesaikan *backlog* ini, pertama-tama yang harus dilakukan adalah menambahkan *field boolean Purc Org Group When PR Draft* di *company additional setting* dengan memasukkan kode 3.4.

```
1 purchase_organization_group = fields.Boolean(string="Purc Org
  Group when PR draft")
```

Kode 3.4: Kode untuk menambahkan *field Purc Org Group* ketika PR dalam *state draft*

Selanjutnya melakukan pengecekan dengan kode 3.5 apakah nilai dari *field purchase_organization_group* bernilai *true* atau *false*, jika bernilai *true* maka *field purchasing* pada UI dan tidak lagi disembunyikan dari pengguna.

```
1 is_purchase_organization_group = fields.Boolean(compute="
  _get_purchase_organization_group")
2 @api.depends('company_id')
3 def _get_purchase_organization_group(self):
4     for record in self:
5         record.is_purchase_organization_group = record.
  company_id.get_setting('purchase_organization_group', False)
```

Kode 3.5: Kode untuk mengecek *value* dari *purchase organization group*

MULTIMEDIA
NUSANTARA

3.4 Kendala dan Solusi yang Ditemukan

Selama periode magang dilakukan ditemui beberapa permasalahan antara lain:

A. Kendala Yang Dihadapi

1. Belum pernah melakukan pemrograman menggunakan *framework* Odoo sebab *framework* Odoo sendiri tidak dipelajari selama periode perkuliahan. Sehingga *framework* Odoo harus dipelajari saat pemagangan di Kompas Gramedia
2. Terdapat berbagai kesulitan saat menginstalasi *framework* Odoo dikarenakan perbedaan perbedaan versi python yang pernah di *install* dan versi python yang digunakan oleh Kompas Gramedia.
3. Sering mengalami kesulitan memahami *backlog* dikarenakan kurangnya pengalaman terhadap fungsionalitas Odoo milik Kompas Gramedia, sehingga memahami *backlog* dengan deskripsi yang singkat menjadi kesulitan tersendiri

Untuk mengatasi permasalahan tersebut solusi yang diterapkan adalah sebagai berikut:

B. Solusi yang Ditemukan

1. Untuk mengatasi kendala belum pernah melakukan pemrograman dengan *framework* Odoo, maka dilakukan pembelajaran mandiri dengan menonton video pembelajaran dari youtube serta membeli modul belajar dari platform Udemy.
2. Untuk mengatasi permasalahan sulitnya melakukan instalasi Odoo maka bantuan dari *developer* lain sangat diperlukan untuk memberikan arahan instalasi python versi lama dan setup *environment local* agar bisa melakukan *coding* di laptop.
3. Ketika mendapatkan *backlog* dengan deskripsi singkat dan kurang jelas maka *backlog* tersebut harus didiskusikan dengan analis untuk meminta uraian lebih lengkap mengenai pekerjaan yang harus dilakukan.