

## BAB III

### PELAKSANAAN KERJA MAGANG

#### 3.1. Kedudukan dan Koordinasi

Praktik kerja magang di PT Siloam International Hospitals Tbk berlangsung selama kurang lebih tiga bulan, dimulai pada 10 Juni 2024 dan berakhir pada 9 September 2024. Selama periode magang ini, posisi yang diemban adalah sebagai *Demand Planning and Inventory Management (DPIM)*, yang memegang peranan penting dalam mengelola permintaan dan persediaan di rumah sakit. Selama praktik kerja, supervisor beserta tim selalu memberikan bimbingan secara langsung serta masukan berharga terkait perkembangan kinerja. Masukan yang diberikan tidak hanya membantu meningkatkan kualitas pekerjaan, tetapi juga berfungsi sebagai sarana pengembangan diri, terutama dalam kemampuan analisis data. Dalam proses tersebut, penggunaan alat *business intelligence* seperti Microsoft Power BI juga diperkenalkan, yang sangat membantu dalam visualisasi data dan pengambilan keputusan.

Selama menjalankan program magang di PT Siloam International Hospitals Tbk, peran yang dipegang berada langsung di bawah unit *Head Of Business Support and Service*. Dalam posisi ini, berbagai tantangan terkait supply chain harus dihadapi, yang menuntut penyelesaian melalui pendekatan *data-driven* serta teknologi terkini. Beberapa proyek yang diberikan berfokus pada penyelesaian masalah rantai pasokan, termasuk optimasi stok dan distribusi barang. Algoritma dan *dashboard* Power BI digunakan secara efektif untuk meningkatkan efisiensi dalam proses bisnis dan meminimalisir kesalahan dalam pengelolaan stok. Dengan adanya pendekatan ini, proses yang sebelumnya dilakukan secara manual kini dapat diotomatisasi, sehingga menghemat waktu dan meningkatkan akurasi.

Hasil akhir dari proyek-proyek tersebut adalah pembuatan *website* sederhana yang diharapkan dapat menjadi solusi jangka panjang bagi tim IT PT Siloam International Hospitals Tbk. *Website* ini dirancang agar dapat digunakan oleh seluruh unit rumah sakit Siloam di berbagai daerah di Indonesia, dengan tujuan

mempermudah monitoring dan pengelolaan stok produk. Selain itu, *output* dari *dashboard* Power BI yang dihasilkan sangat bermanfaat bagi divisi *Demand Planning and Inventory Management*, khususnya dalam memantau kinerja setiap unit rumah sakit secara *real-time*. Dengan adanya *dashboard* ini, setiap unit dapat mengakses data terkait performa mereka, sehingga pengambilan keputusan dapat dilakukan dengan lebih cepat dan akurat, yang pada akhirnya meningkatkan efisiensi operasional di seluruh jaringan rumah sakit Siloam.

### 3.2. Tugas dan Uraian Kerja Magang

Tabel 3.1 berikut merupakan jabaran setiap pekerjaan yang dilakukan selama praktik program magang yang berlangsung di PT Siloam International Hospitals Tbk:

Tabel 3.1 *Timeline* Program Kerja Magang

No.	Aktivitas	Waktu Mulai	Waktu Selesai	Minggu Ke-
1.	Pengenalan <i>Project</i> dan lingkungan kerja	10 Juni 2024	10 Juni 2024	24
2.	<i>Create Prediction Website</i>			
2.1	<i>Scratches to create a website</i>	12 Juni 2024	21 Juni 2024	24 - 25
2.2	<i>Implementing CRISP-DM methodology and Deployment</i>	18 Juni 2024	03 Juli 2024	25 - 26
3.	<i>Create and Import Data to Power BI Dashboard</i>			
3.1	<i>Create FAT Calculation by Store Daily</i>	27 Juni 2024	1 Agustus 2024	26 – 31
3.2	<i>Create DOI Daily Summary</i>	12 Juli 2024	16 Agustus 2024	28 – 33
3.3	<i>Create Change Request</i>	12 Juli 2024	09 September 2024	28 - 37
3.4	<i>Create InterHU Tracker</i>	29 Juli 2024	09 September 2024	31 - 37

Tugas yang dilaksanakan selama mengikuti praktik program magang di PT Siloam International Hospitals Tbk sebagai *Demand Planning and Inventory Management* (DPIM) terdapat sebanyak 3 *project* besar yang dilakukan berdasarkan setiap minggunya. Berikut adalah tugas kerja magang dari ke-3 *project*, sebagai berikut:

### **3.2.1. Pengenalan Project dan lingkungan kerja**

Pengenalan dilakukan pada hari pertama masuk pada tanggal 10 Juni 2024. Pertama kali dipertemukan oleh *Human Resources* PT Siloam International Hospitals Tbk di ruangan lantai 28. Setelah itu dilakukan perbincangan mengenai *benefit* dan juga lingkungan kerja selama praktik program magang berjalan selama 3 bulan. Setelah itu diantar kepada supervisor di lantai 31, dan kemudian dilakukan pemaparan materi mengenai divisi dan juga diperkenalkan dengan rekan satu tim selama bekerja sebagai divisi *Demand Planning and Inventory Management* (DPIM). Selanjutnya dijelaskan juga mengenai beberapa *project* yang akan dilakukan selama menjalankan praktik kerja magang di PT Siloam International Hospitals Tbk termasuk diberikan akses untuk mempelajari setiap data yang kiranya akan digunakan selama praktik kerja magang berjalan.

Maka untuk uraian dari kerja magang pada pengenalan *project* dan lingkungan kerja adalah seperti berikut. Dimulai pada tanggal 10 Juni 2024, dilaksanakan praktik program magang pertama di PT Siloam International Hospitals. Sebelumnya pada tanggal 7 Juni 2024 diinformasikan oleh *Human Resources* PT Siloam International Hospitals Tbk untuk melakukan pertemuan di *Head Office* Siloam Hospital yang berada di Lippo Village, Tangerang. Pertemuan dijanjikan berlangsung dari pukul 8:00.



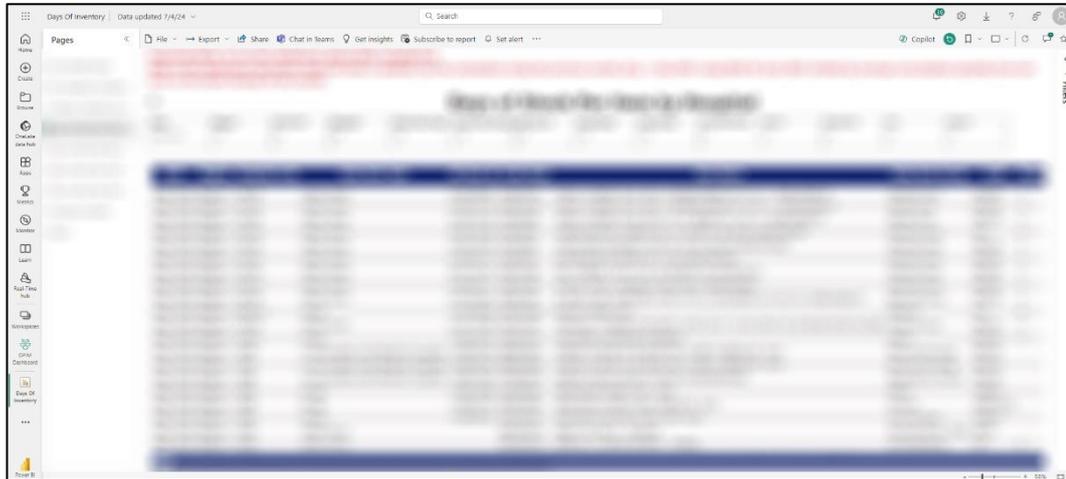
Gambar 3.1 Ruang tunggu hari pertama

Gambar 3.1 merupakan gambar ruang tunggu yang mempertemukan dengan *human resources* pada hari pertama melakukan praktik kerja magang di PT Siloam International Hospitals. Setelah menunggu beberapa menit, selanjutnya dilakukan beberapa paparan terkait peraturan perusahaan, *benefit* yang didapatkan selama menjalankan praktik kerja magang dan juga beberapa informasi mengenai praktik kerja magang di PT Siloam International Hospitals seperti harus menjaga nama baik hingga menjaga data untuk tidak disebarluaskan ke pihak manapun. Setelah mendapatkan pemaparan materi secara individu dengan *human resources*, maka selanjutnya diantar ke ruangan bekerja di lantai 31..

Setelah berada di lantai 31, kemudian diperkenalkan dengan anggota tim. Saat dipertemukan dengan tim, dilakukannya pemaparan materi mengenai divisi *Demand Planning and Inventory Management* di PT Siloam International Hospitals Tbk. Setelah dilakukan pemaparan materi



Juni tersebut) benar-benar dilakukan pemahaman terkait data yang akan digunakan selama praktik kerja magang.



Gambar 3.3 Dashboard Power BI Monthly Hospitals

Gambar 3.3 merupakan Power BI *dashboard* yang digunakan untuk melakukan pembelajaran, dimana terdapat banyak *page dashboard* yang dibuat oleh divisi khusus yang mengurus *Power BI* untuk *user-user* di PT Siloam International Hospitals Tbk. Namun untuk pembelajaran tanggal 11 Juni 2024, hanya menggunakan Power BI *dashboard Days of Inventory* khususnya *page monthly stock per hospital*.

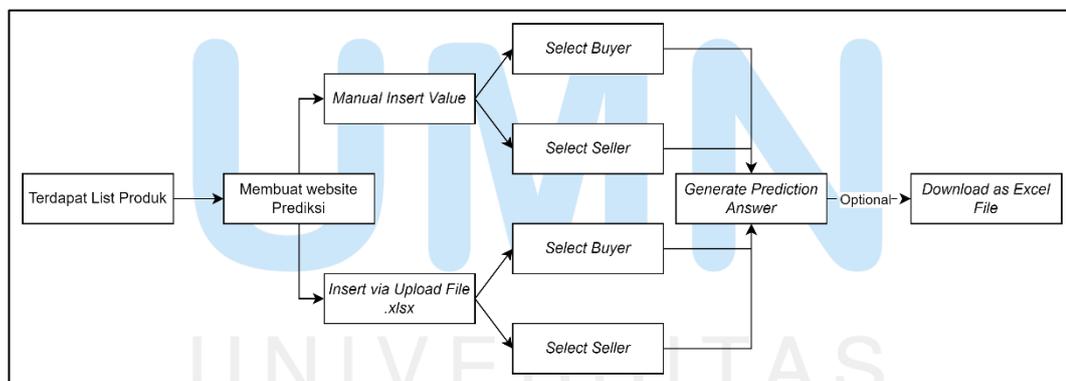
Selanjutnya pada tanggal 12 Juni 2024, dilakukan *recap* data dari *dashboard* Power BI menuju microsoft excel. Tujuan dari dilakukannya *recap* data adalah untuk melihat produk apa saja pada rumah sakit dan *store* apa yang memiliki nilai *infinity* dan NaN pada produknya atau dapat dikatakan tidak ada konsumsi selama 3 bulan terakhir terhadap produk tersebut. Saat melakukan *recap* data, tidak sedikit produk yang dijual oleh banyak rumah sakit. Akan tetapi produk tersebut tidak berjalan, dimana produk tersebut akhirnya *overload* di *inventory* dan memiliki kemungkinan untuk *expired*.

### 3.2.2. Create Prediction Website

Pada *project* pertama ini, dilakukannya pemahaman data yang sebelumnya diberikan. Data *Raw* yang diberikan memiliki banyak sekali

nilai yang tidak sempurna. Dari situ diterapkannya pembelajaran selama perkuliahan yaitu menggunakan metode CRISP-DM. Sebab hasil akhir dari *project* ini adalah untuk dapat menampilkan prediksi rumah sakit yang kiranya dapat membeli atau menjual obat tertentu berdasarkan frekuensi. Tujuannya adalah untuk menyelesaikan permasalahan yang sedang dihadapi divisi dan juga meningkatkan efisiensi waktu dalam mendapatkan hasil.

Untuk uraian dari pelaksanaan kerja magang pada *project create prediction website* adalah seperti berikut, pada 12 Juni 2024, bertepatan dengan melanjutkan pekerjaan mengenai *Days of Inventory Reduction* pada microsoft excel untuk melakukan *recap* data. Pada hari yang sama, dilakukannya juga pembuatan *automation* untuk prediksi rumah sakit yang dapat membeli dan menjual obat yang dimiliki oleh rumah sakit lain. Tujuannya adalah untuk mempermudah divisi *Demand Planning and Inventory Management* dalam memberikan saran kepada unit rumah sakit Siloam lain dalam menjual obat dengan frekuensi penjualan dibawah 3.



Gambar 3.4 Flow website Prediksi Otomatis

Gambar 3.4 merupakan alur proses dari penggunaan *website*. Pada tahap awal diawali dengan adanya data (*list product item*) yang dimiliki oleh pihak *inventory* rumah sakit untuk dicari *buyer / seller* yang kiranya ingin membeli produk mereka. Penulisan *product* tidak dapat sembarangan, karena sudah terdapat *template* khusus yang dibuat supaya sistem *website*

dapat membacanya. Pada *template* tersebut *user* hanya diminta untuk memasukkan 3 variabel utama yaitu *region*, *hospital code* dan *item name*.

### 3.2.2.1. *Scratches to create a website*

Region	Hospital Code	Item Type	Code Hope	AX Item Id	Item Name	Frequency	DOS	QOH	Priority
--------	---------------	-----------	-----------	------------	-----------	-----------	-----	-----	----------

Gambar 3.5 Desain Website Prediksi

Gambar 3.5 merupakan tampilan awal *website* menggunakan *coding* *HyperText Markup Language (html)* dan *Coding Style Sheets (css)* yang dilakukan. Kemudian pada tahap ini *user* akan *list* semua produk, maka selanjutnya *user* akan membuka *website* seperti pada gambar 3.5. *Website* kali ini belum dapat *go live* dikarenakan beberapa kendala. Namun untuk penggunaan sendiri sudah dapat digunakan, dimana *website* berbentuk web sederhana yang hanya meminta *input* berupa *dropdown*.

Hal yang wajib dilakukan oleh *user* saat pertama kali mengakses *website* adalah, memilih metode *input* yaitu Item Name, AX Item Id, dan Hope Code. Kedua, *user* akan memilih region yang menentukan kemana produk dapat dijual. Sebab semua rumah sakit sudah dilakukan *grouping* terhadap lokasinya. Ketiga, *user* akan melakukan *input* nama rumah sakit dengan tujuan supaya prediksi tidak mengeluarkan nama rumah sakit pada hasil prediksi. Selanjutnya *user* akan melakukan *input* nilai pada *textbox*, nilai yang dimasukkan adalah nama produk yang ingin dicarikan referensi. Setelah itu *user* akan memilih *button*, dengan tujuan untuk memastikan hasil

prediksi yang akan diberikan. Setelah memilih, maka prediksi akan langsung keluar di tabel bawah. Namun saat *output* sudah keluar, *user* dapat melakukan *filter* untuk menentukan prioritas. Semakin tinggi prioritas maka akan lebih diutamakan oleh *unit* rumah sakit untuk menjual produk atau membeli produk pada rumah sakit tersebut.

Gambar 3.6 Tampilan *Website* revisi

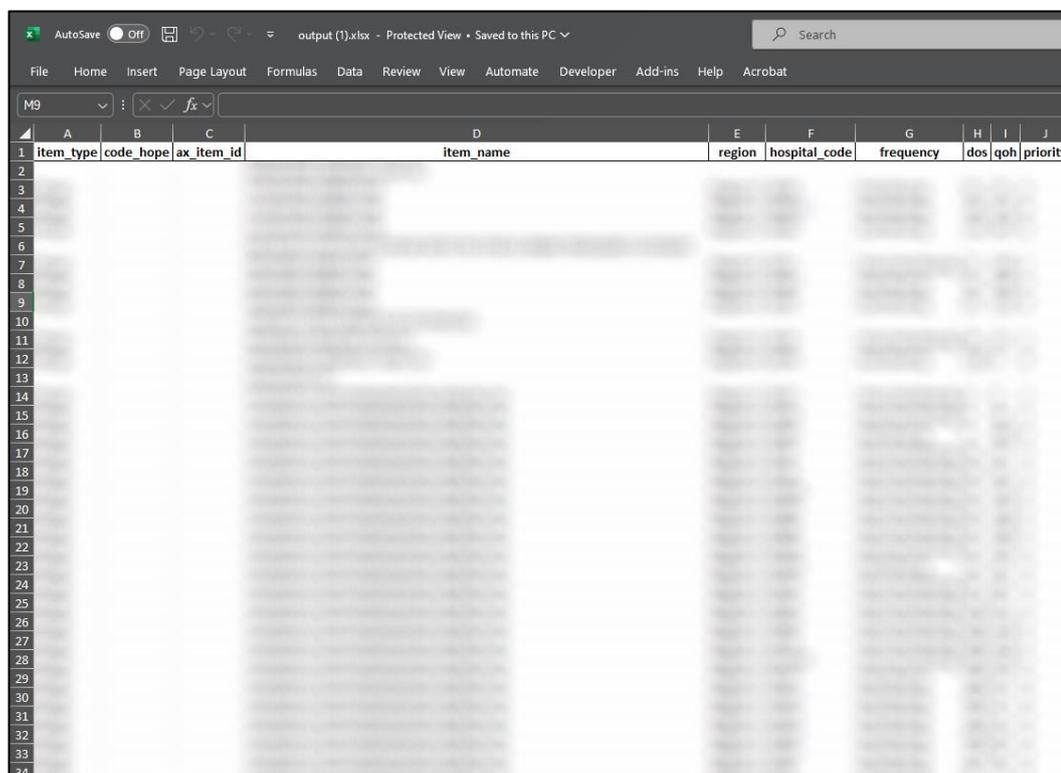
Gambar 3.6 merupakan gambar tampilan *website* yang sudah diperbaharui, beserta dengan *coding* JavaScript yang telah diperbaharui supaya dapat menerima nilai nama produk tanpa harus menambahkan koma ‘,’ melainkan hanya tinggal dibuat secara kebawah (*enter*). Selain itu ditambahkan juga fitur *upload* yang mempermudah *user* dalam menggunakannya, karena saat *user* sudah melakukan *upload* maka semua *dropdown* akan terisi (metode, dan region). Selain itu pada revisi ini ditambahkan fitur *search* yang berguna supaya *user* dapat melakukan pencarian nama produk, frekuensi, item type dan hospital code secara otomatis.

Region	Hospital Code	AX Code	Hope Code	Item Name

Gambar 3.7 *Template* Excel untuk *Upload*

Gambar 3.7 merupakan gambar *template* yang dapat diterima oleh *website* saat *user* ingin langsung mencari data-data produk yang ingin dijual atau dibeli dari rumah sakit lain. Seperti yang telah dijelaskan sebelumnya, bahwa *template* yang di *upload* oleh *user* akan secara otomatis mengisi nilai *region*, dan juga *item name*. Namun untuk saat *website* dibuat, baru dapat menerima *upload* berupa file Excel. Namun untuk kedepannya akan dipelajari lebih dalam supaya *website* dapat menerima file lain seperti csv maupun pdf.

Selain itu pada gambar 3.8 dibawah merupakan gambar dari *output* yang telah di*download* setelah *user* memilih untuk mencari *potential buyers* atau *potential sellers*. Hasil *download* akan secara otomatis masuk ke folder *download* pada sistem operasi *windows* maupun *mac os*. File yang sudah di *download* juga akan berbentuk excel yang dimana akan memudahkan *user* dalam melihat hasil-hasil berdasarkan prioritas tertinggi dari setiap produk hingga terendah.



	A	B	C	D	E	F	G	H	I	J		
1	item_type	code	hope	ax	item_id	item_name	region	hospital_code	frequency	dos	qoh	priority
2												
3												
4												
5												
6												
7												
8												
9												
10												
11												
12												
13												
14												
15												
16												
17												
18												
19												
20												
21												
22												
23												
24												
25												
26												
27												
28												
29												
30												
31												
32												
33												
34												

Gambar 3.8 Output Download

Namun kalau dilihat dengan teliti, hasil *download* akan memuat informasi yang kiranya sangat dibutuhkan oleh *user* dalam menentukan strategi dalam mengurus *inventory* pada setiap unit rumah sakit. Selain itu kalau dilihat pada gambar, terdapat beberapa *row* yang hanya menampilkan *item name* dan terdapat tulisan *region* pada kolom sebelah kiri maupun frekuensi, selain itu tidak terdapat juga nilai *Days of Stock* (DOS) maupun *Quantity on Hand* (QOH).

Hal tersebut dapat terjadi karena, saat algoritma dijalankan dengan 200 ribu *rows* data, tidak semua nilai akurasi dapat mencapai 100%. Dari hal tersebutlah yang menyebabkan terkadang terdapat hasil prediksi yang salah. Maka dari itu tidak jarang divisi *Demand Planning and Inventory Management* (DPIM) melakukan pengecekan ulang terhadap data pada Power BI untuk memastikan bahwa memang betul tidak ada rumah sakit yang tidak mau membeli atau ada rumah sakit yang ingin membeli namun tidak terdeteksi oleh sistem.

Kemudian pada tanggal 03 July 2024, dilakukan pembaharuan data. Hal tersebut dikarenakan, divisi *Demand Planning and Inventory Management* (DPIM) ingin melihat pergerakan *stock inventory product* dari setiap rumah sakit secara rutin. Dari sini dilakukan perubahan data dari data bulanan menuju data *daily* yang terus dilakukan *update* setiap harinya. Namun saat ingin dilakukan perbaikan, ditemukan banyak sekali *error*. *Error* yang paling sering terjadi yaitu nama variabel yang berbeda antara data *monthly* dan juga data *daily*. Namun hal tersebut dapat ditindaklanjuti dengan meng-copy data *daily* ke *template* data *monthly*.

Setelah menyelesaikan *error* terhadap data, terdapat *error* lain yang ditemukan. *Error* kedua yang ditemukan adalah akurasi algoritma yang digunakan yang semakin kecil. Namun setelah dicoba beberapa penerapan algoritma, maka ditetapkan algoritma Decision Tree dan Random Forest sebagai algoritma dengan nilai akurasi pada nilai *y* yaitu frekuensi yang paling baik.

Pemilihan algoritma untuk membandingkan Decision Tree dan Random Forest disebabkan karena keinginan untuk melihat apakah Random Forest akan menghasilkan nilai akurasi yang lebih baik dibandingkan Decision Tree. Sebab perbedaan dari Random Forest dengan Decision Tree adalah dua algoritma machine learning yang digunakan untuk melakukan klasifikasi dan regresi. Namun untuk prinsip kerja Decision Tree adalah membangun sebuah pohon keputusan berdasarkan aturan-aturan yang dipelajari dari data. Sedangkan untuk prinsip kerja dari Random Forest adalah menggunakan metode *ensemble* dari banyak Decision Tree yang dibangun secara independen. Kemudian setelah menemukan algoritma terbaik, selanjutnya dilakukan penyimpanan model menjadi *file pickle* (pkl) yang nantinya akan digunakan sebagai model yang digunakan untuk *deployment website* yang telah dibuat.

Masuk ke hari selanjutnya yaitu 08 Juli 2024, dilakukan beberapa perubahan dan penyesuaian terhadap data. Perubahan yang dilakukan adalah menyelesaikan *error* yang terjadi, dimana *error* tersebut menyebabkan Ax Item Id dan Code Hope tidak mau keluar. Namun saat dilakukan beberapa perubahan pada coding, hal hasil Ax Item Id dan Code Hope dapat keluar.

Selanjutnya untuk gambar 3.9 dibawah merupakan gambar dari hasil perubahan terhadap *code*, dimana saat *user* menekan salah satu *button* (*find potential seller / find potential buyer*) maka *output* pada *website* akan mengeluarkan seluruhnya. Selain itu terdapat beberapa penyesuaian yang dilakukan pada *website* untuk memudahkan *user* dalam memahami setiap fitur yang terdapat pada *website*.

**Find Potential Buyers / Sellers**

Select Input Method:  Select Your Region:

Enter Your Hospital Code:  Select Priority:

Enter Values:  Find:

Upload XLSX File:  No file chosen  Search Keywords:

Item Type	Code Hope	AX Item Id	Item Name	Region	Hospital Code	Frequency	DOS	QOH	Priority
-----------	-----------	------------	-----------	--------	---------------	-----------	-----	-----	----------

Gambar 3.9 Tampilan *website* prediksi final

Selain melakukan perubahan pada tampilan *website* dilakukan juga perubahan pada *code* dimana sebelumnya terdapat *error* pada saat *user* ingin melakukan *download* terhadap hasil *output* prediksi. Error ini berlangsung saat *user* membuka *file* *xlsx*. Pada saat itu juga *file* langsung tidak mau mengeluarkan hasil, melainkan mengeluarkan *warning* yang menyampaikan bahwa *file* excel mengalami *corrupt*. Dari hal tersebut dilakukannya pembetulan pada *packages* yang di *install* pada anaconda.

### 3.2.2.2. Implementing CRISP-DM methodology and Deployment

```

C:\Users\HP VICTUS>rem proj-data is installed because its license was copied over
(base) C:\Users\HP VICTUS>conda activate data
(data) C:\Users\HP VICTUS>conda install xlsxwriter
Retrieving notices: ...working... done
Collecting package metadata (current_repodata.json): done
Solving environment: \
The environment is inconsistent, please check the package plan carefully
The following packages are causing the inconsistency:
- defaults/win-64::bokeh==3.4.0=py310h9999e9c_0
- anaconda/noarch::ipywidgets==7.6.5=pyhd3eb1b0_1
- defaults/win-64::jupyter==1.0.0=py310haa95532_0
- defaults/win-64::jupyterlab==3.5.3=py310haa95532_0
- defaults/noarch::jupyterlab_server==2.10.3=pyhd3eb1b0_1
- defaults/win-64::jupyter_server==1.23.4=py310haa95532_0
- defaults/win-64::nbclassic==0.5.4=py310haa95532_0
- anaconda/win-64::nbconvert==6.4.0=py310haa95532_0
- anaconda/win-64::notebook==6.4.11=py310haa95532_0
- defaults/win-64::notebook-shim==0.2.2=py310haa95532_0
- defaults/win-64::tensorboard==2.10.0=py310haa95532_0
- defaults/win-64::tensorflow==2.10.0=ml_py310hd99672f_0
- defaults/win-64::tensorflow-base==2.10.0=ml_py310ha7f48e_0
- anaconda/win-64::widgetsnbextension==3.5.2=py310haa95532_0

```

Gambar 3.10 Instalasi *Package*

Pada gambar 3.10 merupakan gambar dari *instalasi* pada anaconda yang disebabkan oleh tidak ditemukannya *package* yang digunakan untuk

melakukan *deployment*. *Package* yang digunakan adalah selain *package* pandas, matplotlib, scikit-learn adalah xlswriter. *Package* xlswriter adalah *package* yang digunakan untuk membuat *file* excel dari awal, dimana bisa untuk membuat *file* excel baru atau memodifikasi *file* yang sudah ada. Secara tidak langsung penggunaan dari *package* tersebut adalah untuk fitur *download* yang disediakan untuk meng-*generate* hasil prediksi.

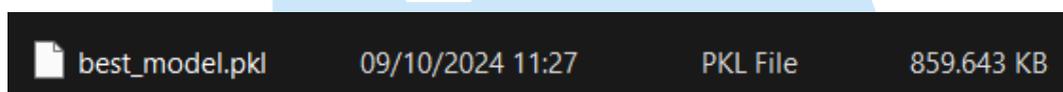
```
import pandas as pd
import numpy as np
from sklearn.preprocessing import LabelEncoder, StandardScaler
from sklearn.ensemble import RandomForestClassifier
from sklearn.model_selection import train_test_split
from sklearn.metrics import accuracy_score, classification_report
from sklearn.tree import DecisionTreeClassifier, plot_tree
import matplotlib.pyplot as plt
import pickle
```

Gambar 3.11 *Libraries* dalam Python untuk Data Mining

Pada gambar 3.11 merupakan gambar *libraries* lain yang digunakan dalam pembuatan *website* prediksi. Pertama yang digunakan adalah *libraries* Pandas, *libraries* tersebut berguna sebagai pustaka analisis dan untuk melakukan manipulasi data dengan memungkinkan pemuatan, pembersihan, transformasi, dan analisis data secara efisien. Kedua adalah *libraries* NumPy yang berguna untuk komputasi numerik serta menyediakan fungsi aljabar *linear*, pembuatan angka acak, dan untuk operasi matematika yang efisien pada jumlah data yang besar.

Selanjutnya untuk *libraries* Scikit-learn atau *Sklearn* digunakan cukup banyak diantaranya adalah *label encoder* dan *standar scaler*. *Label Encoder* digunakan untuk mengubah label kategorikal menjadi nilai numerik, dan *standar scaler* untuk menstandarisasikan fitur dengan menghilangkan rata-rata dan menskalakan ke unit *varians*. Kedua *Accuracy score* dan *classification report*. *Accuracy score* digunakan untuk menghitung akurasi dari model klasifikasi sedangkan untuk *classification report* digunakan untuk menyediakan metrik evaluasi detail untuk model

yang didalamnya termasuk presisi, *recall*, *F1-score*. Ketiga adalah *decision tree classifier* dan *plot tree*. *Decision tree classifier* digunakan untuk mengimplementasikan algoritma *decision tree* yang berguna untuk membuat prediksi berdasarkan struktur seperti pohon keputusan didasarkan pada nilai fitur. Sedangkan untuk *plot tree* digunakan untuk memvisualisasikan struktur pohon keputusan, serta membantu untuk memahami logika model dan bagaimana model sampai pada prediksi.



Gambar 3.12 Penyimpanan *File Best Model* dalam *Pickle*

Gambar 3.12 merupakan gambar setelah selesai melakukan evaluasi, model dengan nilai akurasi yang paling baik akan disimpan dalam bentuk *pickle*. *Pickle*, merupakan *libraries* yang berguna untuk melakukan penyimpanan model terlatih atau objek Python ke sebuah *disk* atau zip untuk penggunaan *deployment*. Jadi pada saat melakukan *coding* untuk *deployment* menggunakan bahasa Python, *file* dalam bentuk *pkl* akan dipanggil untuk digunakannya model sebagai alat prediksi pada *deployment*.

```
app.py X
1 from flask import Flask, render_template, request, send_from_directory, jsonify, send_file
2 import pandas as pd
3 import numpy as np
4 from sklearn.preprocessing import LabelEncoder, StandardScaler
5 from sklearn.model_selection import train_test_split
6 from io import BytesIO
7 import pickle
8 import openpyxl
9
10 app = Flask(__name__, template_folder='C:/templates/templates')
11
12 # Variable to store uploaded data
13 uploaded_data = None
14
15 # Load the best model, encoders, and data
16 with open('best_model.pkl', 'rb') as f:
17     model_data = pickle.load(f)
18     best_model = model_data['model']
19     le_type = model_data['le_type']
20     le_item = model_data['le_item']
21     le_hospital = model_data['le_hospital']
22     le_region = model_data['le_region']
23     le_code_hope = model_data.get('le_code_hope', None) # Get Code Hope encoder if available
24     le_ax_item_id = model_data.get('le_ax_item_id', None) # Get AX Item Id encoder if available
25     data = model_data['data']
```

Gambar 3.13 *Load Pickle File*

Gambar 3.13 merupakan gambar dari tahapan untuk membuat *deployment* dari file *app.py*. File yang telah dibuat akan menggunakan *pickle* yang telah tersimpan untuk digunakan dan dipanggil, supaya dapat membuat prediksi sesuai dengan *best model* yang digunakan. Selain melakukan *load* pada *file pickle*, dilakukan juga *load file encode* dan data. Tujuannya adalah tetap menggunakan data yang telah dilakukan *pre-processing*.

Setelah semua itu selesai, hingga sampai akhir program praktik magang, *database* dalam bentuk excel terus di *update* dengan data terbaru dari pusat. Selain itu *website* juga terus digunakan untuk mencari *buyer* maupun *seller*, walaupun hanya dapat digunakan di laptop pribadi. Namun untuk kedepannya akan diserahkan kepada pihak IT PT Siloam International Hospitals untuk dikembangkan dan di *go-live* untuk karyawan dan *unit* di seluruh Siloam International Hospitals.

### **3.2.3. Create and Import Data to Power BI Dashboard**

Pada *project* ini selain memahami setiap data yang digunakan untuk membuat *dashboard* dan juga rumus pada Power BI. Pada *project* ini dilakukannya juga *update* data dari Power BI pusat ke data excel pribadi (menggunakan *onedrive* perusahaan). Hal ini dilakukan untuk terus melakukan *update* pada *dashboard* Power BI yang telah dibuat, supaya *unit* dari setiap rumah sakit mengetahui setiap pertumbuhan dari *unit-unit* yang ada di rumah sakit tersebut. Selain itu dilakukan juga pembuatan *dashboard* Power BI untuk membantu tim dalam memberikan masukan kepada unit lain di rumah sakit Siloam di Indonesia. Dengan adanya *dashboard* ini, divisi dapat melihat *store* atau *hospitals* apa yang memiliki stok *item* yang memiliki kelebihan di rumah sakit Siloam lainnya. Tujuannya supaya mempermudah dalam pencarian untuk diberikan masukan untuk menjual produk, supaya nilai kelebihan pada unit tersebut bisa berkurang.



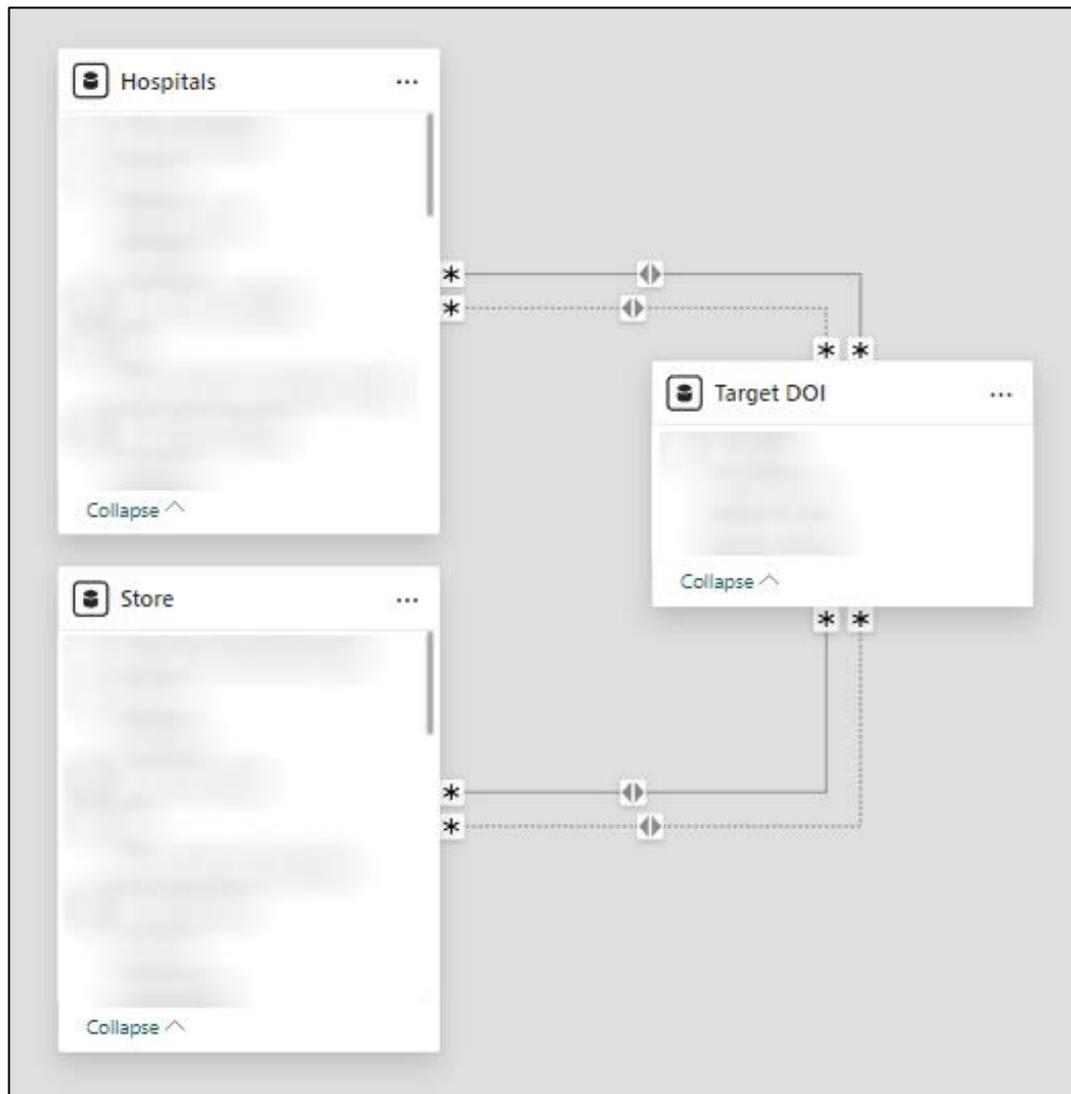
Date	Region	HospitalCd	SalesItemAXItemId	CodeHope	ItemName	SalesItemf	UoM	Pereto	Frequency	VEN	Volume	Category	f	Days of Stc	Quantityor	QOH_Value	Avg Consumption	TotalConsumptionQuantity	TotalConsumpti	

Gambar 3.15 Data Power BI (*Hospitals*)

Gambar 3.15 merupakan gambar dari data *Hospitals* yang digunakan untuk melakukan analisis di Power BI. Hal yang membedakan data *Store* dengan data *Hospitals* adalah perbedaan pada 1 kolom yaitu, pada data *Store* terdapat nama-nama *store* atau *unit* di rumah sakit tersebut. Sedangkan pada data *Hospitals* tidak ada nama *store* melainkan hanya nama rumah sakit. Kemudian hal yang membedakan juga terdapat pada tanggal, dimana data *Hospitals* dapat memuat hingga 4 hari. Namun berbeda dengan data *store* yang hanya dapat memuat hingga 3 hari. Hal tersebut dapat terjadi karena keterbatasan pada *rows* yang dapat ditampung oleh excel, dimana batas akhir *rows* yang dapat ditampung oleh excel adalah sebanyak 1 juta *rows*. Maka dari itu terdapat perbedaan hari untuk data *store* dan *Hospitals*.

UIN  
UNIVERSITAS  
MULTIMEDIA  
NUSANTARA



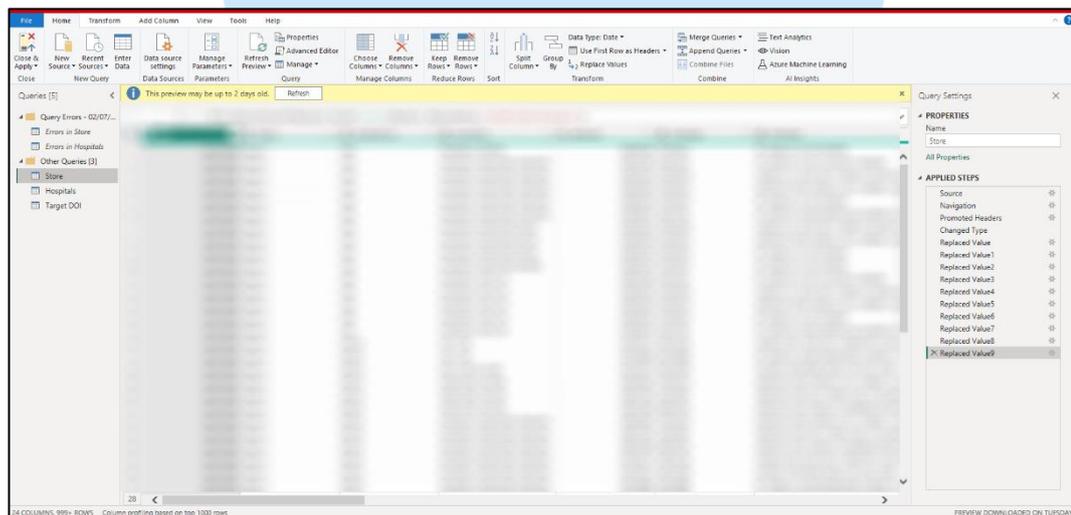


Gambar 3.17 Data Model

Gambar 3.17 merupakan gambar dari penghubungan antar tabel. Dapat terlihat pada gambar bahwa terdapat beberapa variabel yang dilakukan penghubungan *many-to-many* dengan data pada tabel lain. Namun dalam pemahaman *database* terdapat beberapa relasi yang dapat dibangun antar data seperti *One-to-One*, *One-to-Many*, dan *Many-to-Many*. *One-to-one Relationship* biasa terjadi saat setiap entitas memiliki hanya satu keterkaitan (bukan hubungan ganda). Sedangkan untuk *One-to-many Relationship* biasa terjadi saat suatu entitas memiliki banyak hubungan dengan entitas yang lain. Terakhir adalah *Many-to-many Relationship* yang

biasanya terjadi saat setiap entitas memiliki banyak hubungan dengan entitas lainnya [11].

Setelah membahas mengenai relasi antar tabel *hospitals*, *store*, dan *Target DOI*. Selanjutnya maka akan dibuatkan desain dari tampilan *dashboard*. Namun sebelum masuk ke *dashboard*, hal pertama yang harus dilakukan adalah membuat *raw* data menjadi lebih baik (*preprocessing data*) yang dapat dilakukan pada *transform data*. *Transform data*, merupakan tempat dimana pengguna dapat melakukan pengeditan data, melakukan manipulasi pada data, hingga mengubah tipe data yang akan digunakannya.



Gambar 3.18 *Transform Data*

Gambar 3.18 merupakan gambar dari *flow* dalam melakukan data *preprocessing*. Pada gambar sebelah kanan dapat terlihat bahwa dilakukannya beberapa *replace value* guna membuat data yang *blank* atau *null* terisi dengan nilai 0. Sedangkan terdapat juga *Change Type* yang dilakukan untuk mengubah tipe data dari suatu variabel dalam data. Setelah menyelesaikan *transform data* maka selanjutnya adalah melakukan *save* dan keluar dari tahapan *transform data* untuk melanjutkan pembuatan *dashboard*.

Store	Product	Fat Calculation
Store 1	Product A	Value 1
Store 1	Product B	Value 2
Store 2	Product A	Value 3
Store 2	Product B	Value 4
Store 3	Product A	Value 5
Store 3	Product B	Value 6

Gambar 3.19 *Dashboard Fat Calculation (Store)*

Gambar 3.19 merupakan gambar dari pembuatan *dashboard Fat Calculation* untuk Store. Pembuatan *dashboard* dilandasi dari kesulitan dalam melakukan pemantauan kinerja dari setiap store pada rumah sakit tertentu. Selain kesulitan pemantauan kinerja, *dashboard* juga dibuat demi untuk mengetahui kerugian dari produk-produk yang tidak terjual atau mengalami penumpukan yang diakibatkan oleh pembelian yang terlalu banyak.

Hospital Unit	Product	Fat Calculation
Unit 1	Product A	Value 1
Unit 1	Product B	Value 2
Unit 2	Product A	Value 3
Unit 2	Product B	Value 4
Unit 3	Product A	Value 5
Unit 3	Product B	Value 6

Gambar 3.20 *Dashboard Fat Calculation (Hospitals)*

Gambar 3.20 merupakan gambar yang lebih sederhana dibandingkan dengan gambar 3.19. Gambar 3.20 merupakan gambar dari *dashboard Fat Calculation* yang dibuat untuk setiap Hospital Unit. Secara tidak langsung *dashboard* ini akan merangkum setiap obat yang dimiliki oleh store pada setiap rumah sakit. Selain itu pada gambar 3.17 dan gambar

3.18 keduanya memiliki *filter* yang sama untuk memudahkan *user* dalam melakukan pencarian. Selain terdapat *filter* terdapat beberapa hal yang harus dilakukan pada *dashboard* Store dan juga Hospitals. Hal yang harus dilakukan adalah membuat rumus atau *calculation field* pada Power Bi menggunakan DAX.

Rumus yang harus dibuat diantaranya adalah *Consumption Quantity Hospitals*, *Consumption Value per day Hospitals*, *Current DOI Hospitals*, *DOI Fat Hospitals*, *DOI Target Hospitals*, *FAT Quantity Hospitals*, *FAT Value Hospitals*, *Inventory Target Quantity Hospitals*, *Inventory Target Value Hospitals*, *Last Day in Month Hospitals*, dan *QOH\_Ratio Hospitals*. Sedangkan untuk rumus yang digunakan untuk *dashboard* Store kurang lebih sama dengan Hospitals yaitu *Consumption Quantity*, *Consumption Quantity Store*, *Consumption Value per Day Store*, *Current DOI Store*, *DOI FAT Store*, *DOI Target Store*, *FAT Quantity Store*, *FAT Value Store*, *Inventory Target Quantity Store*, *Inventory Target Value Store*, *Last Day in Month Store*, dan *QOH Ration Store*. Secara keseluruhan untuk *dashboard* Store dibuat 12 *calculation field*, sedangkan untuk *dashboard* Hospitals dibuat sebanyak 11 *calculation field*.

Setelah menyelesaikan pembuatan *dashboard*, maka selanjutnya akan masuk ke tahap pengecekan oleh *user*. Tujuan dari pengecekan ini adalah untuk mencocokkan penghitungan secara manual dengan rumus / *calculation* yang ada di Power BI. Pada minggu pertama pengecekan masih terdapat penghitungan yang salah dan masih diminta untuk melakukan penambahan kolom untuk memperjelas perhitungan. Kemudian 1 minggu kemudian dilakukan pengecekan kembali terhadap *dashboard* dan didapatkan masih kurang tepatnya perhitungan. Kemudian dari masukan yang diberikan oleh *user* maka dalam 3 minggu *dashboard* telah disetujui oleh *user* pertama. Namun masih terdapat masukan dari *user* kedua yang meminta untuk melakukan perbaikan pada visualisasi tabel. Masukan tersebut akhirnya dapat diselesaikan dalam 1 bulan pengerjaan.

Setelah semua masukan telah diberikan dan dilakukan perbaikan. Maka *dashboard* dapat digunakan oleh *user* dalam pemantauan kinerja Hospitals dan juga Store pada rumah sakit. Namun karena adanya keterbatasan akses dari perusahaan, maka *dashboard* hanya dapat di *publish* pada *share folder* tim.

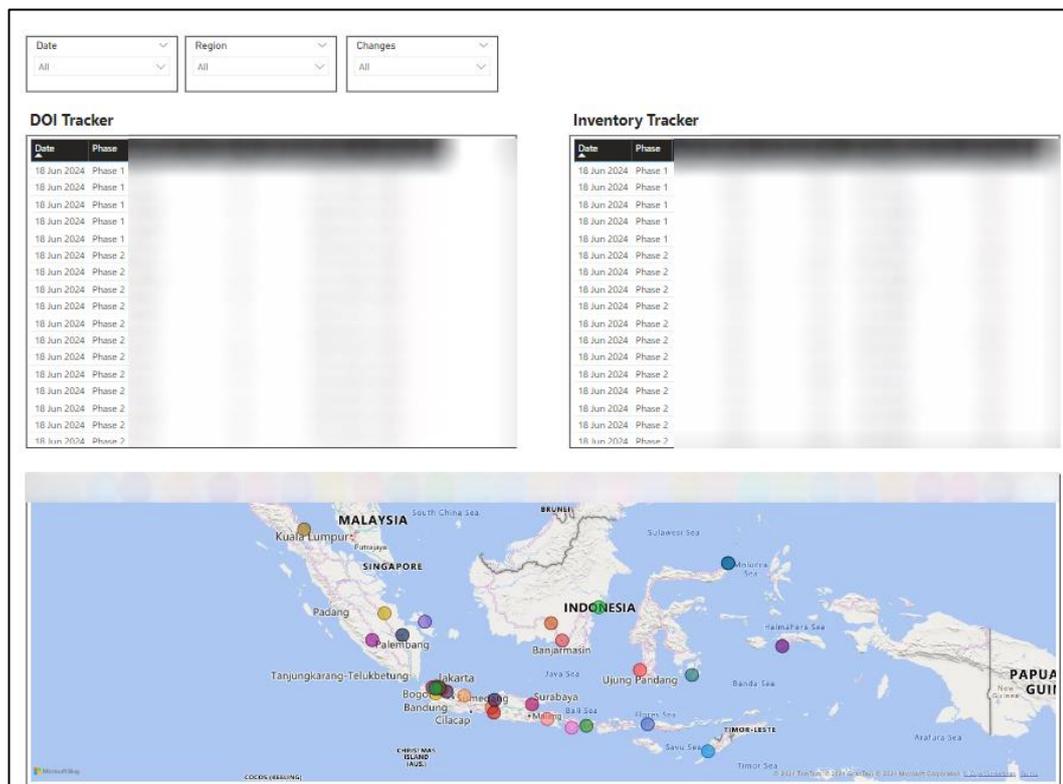
### 3.2.3.2. Create DOI Daily Summary

Kemudian masuk bulan Juli, tepatnya 12 Juli 2024 diberikan lagi *project* Power BI baru oleh *user* untuk membuat *Dashboard* mengenai *Days of Inventory* dari beberapa *phase* rumah sakit di 5 wilayah. Pembuatan *dashboard* bertujuan untuk memudahkan *user* dalam pemantauan kinerja dari setiap rumah sakit di setiap wilayah. Selain melakukan pemantauan kinerja, dalam *dashboard* juga terdapat visualisasi pergerakan *days of inventory* dan juga nilai *inventory* dari tanggal-tanggal sebelumnya (*time series data*).

Date	Target DOI	DOI Target (Lowest)	Inventory Value Target	Inventory Value (Lowest)	Hospital Code	DOI	Inv Value	D DOI	D Inv Value
18-Jun-24									
18-Jun-24									
18-Jun-24									
18-Jun-24									
18-Jun-24									
18-Jun-24									
18-Jun-24									
18-Jun-24									
18-Jun-24									
18-Jun-24									
18-Jun-24									
18-Jun-24									
18-Jun-24									
18-Jun-24									
18-Jun-24									
18-Jun-24									

Gambar 3.21 Dataset *Timeseries DOI Monitoring*

Gambar 3.21 merupakan dataset yang digunakan untuk membuat *dashboard* pemantauan *Days of Inventory*. Dataset bermula dari 18 Juni hingga saat ini. Namun saat menyentuh 1 juta baris maka 18 Juni akan tergantikan dengan tanggal diatasnya (19 juni) supaya pemantauan sampai saat ini dapat terus berjalan. Pembuatan dataset juga dapat terbilang masih manual karena nilai atau *value* pada setiap variabel masih diambil dari *dashboard* lain. Hal tersebut yang membuat penginputan nilai masih manual, karena kurangnya akses kedalam database perusahaan.

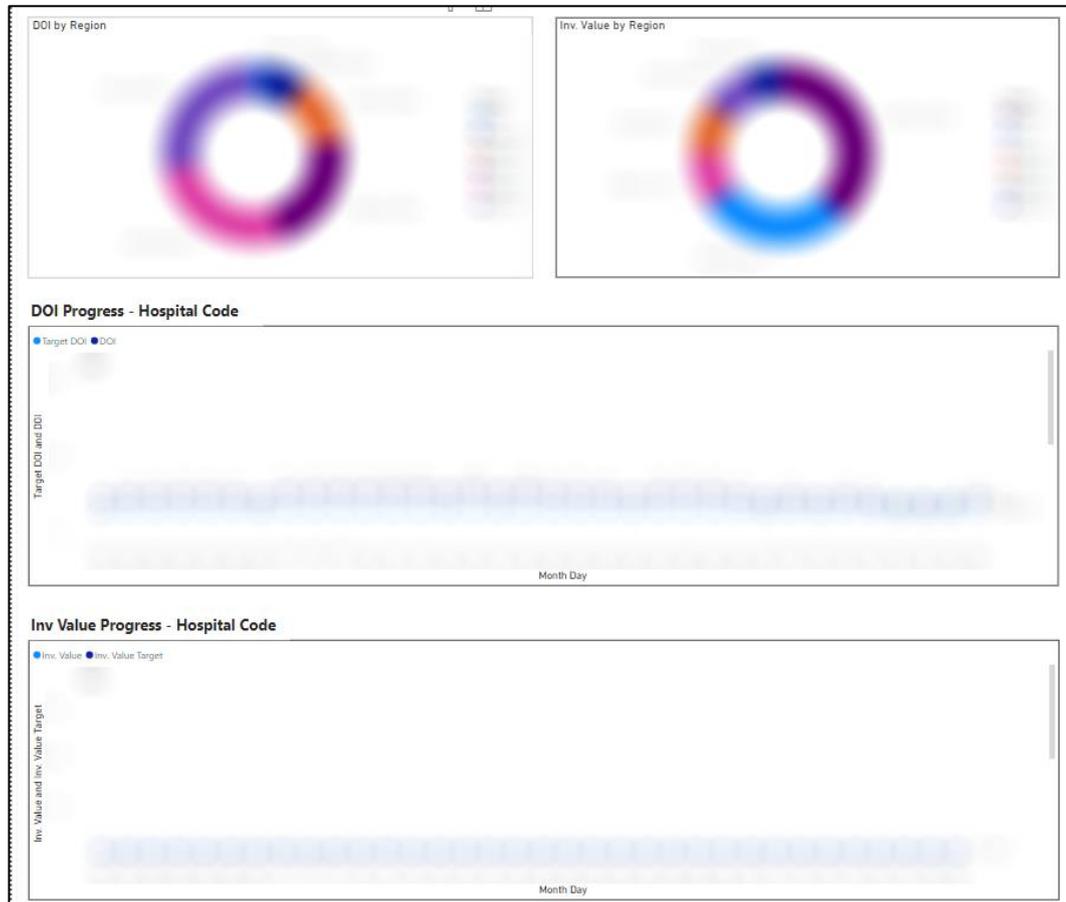


Gambar 3.22 Dashboard DOI Monitoring (1)

Gambar 3.22 merupakan gambar *section 1* dan *2* dari *dashboard Days of Inventory*. Tampilan awal menampilkan 2 tabel yang memberikan informasi mengenai perubahan terhadap nilai *days of inventory* dan juga nilai *inventory* dari setiap rumah sakit yang dapat terlihat pada peta. Selain menampilkan nilai-nilai, terdapat juga kolom khusus yang menampilkan perubahan, apabila nilai dari hari sebelumnya meningkat maka akan menampilkan panah keatas dan sebaliknya kalau turun maka panah akan turun.

Rumus yang digunakan dalam pembuatan kondisi tersebut adalah seperti berikut:  $\text{Changes} = \text{IF} (\text{Table\_name} [\text{variabel}] < 0, \text{“↓”}, \text{IF} (\text{Table\_name}[\text{variabel}] > 0 \text{“↑”}, \text{“”}))$ . Pada rumus ini dapat terlihat bahwa apabila variabel memiliki nilai kurang dari 0 maka langsung menunjukkan arah panah kebawah, sedangkan apabila variabel memiliki nilai lebih dari 0 maka arah panah akan keatas. Namun apabila tidak ada perubahan maka pada kolom Changes akan kosong. Akan tetapi kondisi Changes kosong

tidak ada anak panah sangatlah jarang, sebab setiap hari pasti akan terjadi peningkatan atau penurunan *consumption* pada setiap rumah sakit.

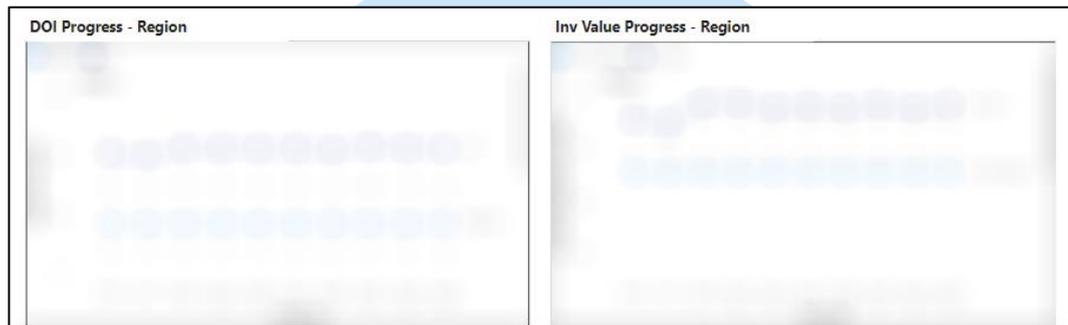


Gambar 3.23 Dashboard DOI Monitoring (2)

Gambar 3.23 merupakan gambar untuk *section* 3, 4 dan 5. Pada *section* 3 dijelaskan menggunakan *pie chart* mengenai total DOI pada setiap *region*. Sedangkan di sebelah kanan merupakan *pie chart* mengenai total *inventory value* dari total pada setiap *region*. Dari visualisasi ini diharapkan tim dapat mengetahui keseluruhan *region* yang masih harus dilakukan perbaikan untuk menurunkan tingkat *days of inventory* (DOI).

Kemudian berlanjut pada *section* 4 yang menampilkan data *time series* mengenai target DOI dan *actual* DOI dari *hospital* yang dipilih pada filter. Tujuan dari pembuatan visualisasi adalah untuk mengetahui apakah terdapat rumah sakit yang mengalami penurunan atau peningkatan tinggi

dari target yang telah dibuat. Sedangkan di bawahnya adalah *section 5* yang mirip dengan *section 4*, dimana pembedanya adalah untuk *section 4* merupakan DOI. Sedangkan untuk *section 5* lebih membahas mengenai nilai *inventory* dari rumah sakit yang di filter.



Gambar 3.24 *Dashboard DOI Monitoring (3)*

Gambar 3.24 merupakan *section* terakhir yang ada pada *dashboard Days Of Inventory*. Pada *section* ini dibuatkan 2 visualisasi yang mirip seperti *section 4* dan 5. Namun pembeda dari visualisasi 4 dan 5 adalah, visualisasi pada *section* terakhir lebih ditunjukkan untuk mengetahui pengelompokan dari setiap rumah sakit. Tujuannya adalah untuk mengetahui nilai DOI dan juga nilai *inventory value* dari total rumah sakit di wilayah-wilayah yang sudah dikelompokkan oleh rumah sakit.

### **3.2.3.3. Create Change Request**

Memasuki bulan Agustus, terjadi penambahan pekerjaan yang diberikan ke beberapa anggota tim. Penambahan pekerjaan kebetulan masih berhubungan dengan Power BI. Pekerjaan yang dimaksud adalah pembuatan *Change Request (CR)* yang nantinya akan diberikan kepada tim *business intelligence*. Jadi prosesnya adalah, disaat *user* ingin membuat sebuah *dashboard* yang dapat *go live* maka dibutuhkan CR sebagai bentuk validasi terhadap apa yang diminta kepada tim *business intelligence*.

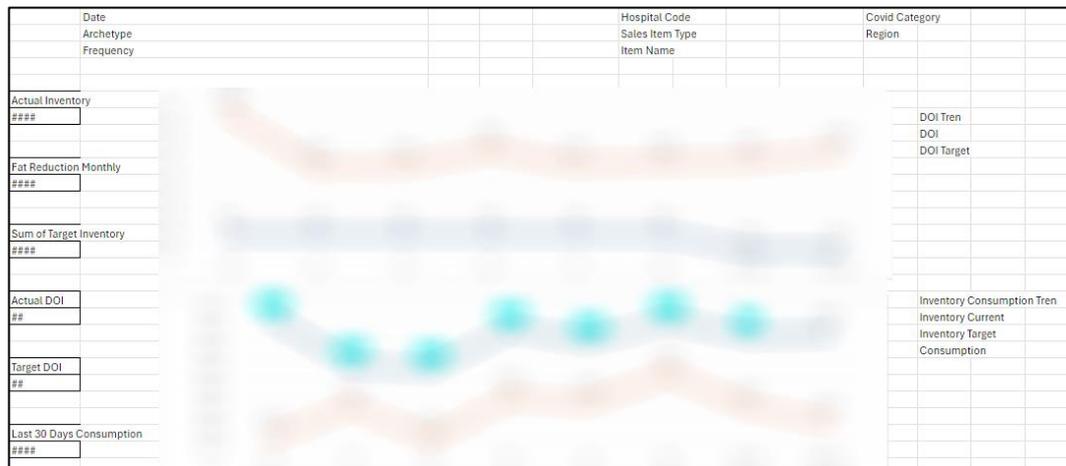
The image shows two identical 'CHANGE REQUEST FORM' templates. Each form is divided into several sections:

- Header:** 'CHANGE REQUEST FORM' and 'CR NO.' field.
- Filed by Requester:** A table with fields for DATE (Monday, August 12, 2024), REQUESTER NAME (Eryanto Chandra), DEPARTMENT (Supply Chain), and HOSPITAL UNIT (Head Office).
- PROJECT NAME:** DAI Daily Summarization.
- CASE FOR CHANGE:** A large central area with three sub-sections: PROPOSED CHANGE, REASON FOR CHANGE, and INTENDED OUTCOME.
- REQUEST APPROVAL:** A table with fields for REQUEST BY (Eryanto Chandra), HO DIV, and DIRECTOR, each with a corresponding SIGNATURE field.
- Footer:** 'Filed by BI Team' and 'POTENTIAL IMPACT' checkboxes (ANALYTICS, BI, DATA, REPORTS, TOOLS, OTHER).

Gambar 3.25 Contoh CR Form

Gambar 3.25 merupakan gambar dari pembuatan CR untuk melakukan pembuatan *dashboard*. Pada CR terdapat beberapa ketentuan yang wajib diletakkan seperti *proposed change* yang didalamnya menjelaskan apa saja yang ingin diubah atau ditambahkan pada *dashboard*. Selain itu terdapat juga *intended outcome* yang didalamnya wajib dijelaskan *output* apa yang ingin dikeluarkan, seperti visualisasi seperti apa hingga *value* apa yang ingin dikeluarkan.

Setelah CR selesai dibuat, maka selanjutnya adalah membuat *requirement* yang didalamnya akan dijelaskan berbagai rumus atau visualisasi apa saja yang ingin digunakan seperti *bar chart*, *pie chart*, atau hanya menggunakan tabel. Tujuan akhir dari adanya *requirement* adalah untuk membantu tim *business intelligence* supaya mendapatkan gambaran yang diharapkan oleh *user*. Namun dalam pembuatan *requirement* perlu dilakukan beberapa konsultasi dengan *user*, sebab tidak sedikit akan terdapat masukan yang harus ditambahkan pada *requirement*.



Gambar 3.26 Contoh Requirement CR

Gambar 3.26 merupakan gambar dari contoh *requirement* yang wajib dibuat dalam CR. Pada gambar dapat terlihat sebuah tampilan yang mirip dengan *dashboard* Power BI. Pada sebelah kiri akan dimuat beberapa data. Kemudian pada bagian atas akan dibuat sebuah filter yang berguna untuk *user* dalam mencari data, serta memperkecil lingkup data. Selain itu pada bagian tengah terdapat visualisasi berupa *line chart* yang berguna sebagai *insight* bagi *user* dalam melakukan pemantauan. Maka secara keseluruhan *requirement* CR berguna untuk membantu tim *business intelligence* dalam membuat tampilan yang semirip mungkin dengan menggunakan data yang sudah ada pada perusahaan.

Setelah CR dan *requirement* selesai dibuat maka akan ditinjau kembali oleh tim dan juga *general manager*. Apabila sudah tidak ada perubahan maka *general manager* akan memberikan tanda tangan. Setelah itu tidak lupa untuk meminta tanda tangan kepada direktur. Setelah seluruh tanda tangan telah ada, maka *form* akan diberikan kepada tim *project executive* untuk dibuatkan dalam bentuk FD. Setelah selesai dibuat dalam bentuk FD maka akan dilakukan pengecekan kembali terkait rumus dan juga *requirement* yang diminta pada CR. Apabila sudah sesuai maka FD akan diteruskan langsung kepada tim *business intelligence* untuk dibuatkan *dashboard* Power BI.

Namun setelah itu masih terdapat tugas lain, yaitu melakukan pelaporan *progress* dari setiap CR yang sudah diberikan kepada tim *business intelligence*. Pelaporan *progress* biasanya disampaikan pada *meeting internal* dan juga diletakkan pada excel yang telah dibuat. Tujuan pelaporan adalah untuk mengetahui proses sudah sampai tahap apa dan selalu mengingatkan tim *business intelligence* bahwa target yang ditetapkan sudah dekat. Namun tidak dipungkiri terkadang tim *business intelligence* mengalami kendala yang mengakibatkan tertundanya target yang telah mereka tetapkan dan dari kejadian tersebut dibutuhkan konfirmasi lagi terhadap target selanjutnya.

Pembuatan CR dan proses pemantauan berlangsung setiap hari dari hari senin hingga jumat, dan pada setiap hari selasa akan dilakukan pelaporan terhadap perkembangan. Namun selain membuat CR untuk tim *business intelligence* perusahaan. Terdapat juga pembuatan CR dengan *template* berbeda untuk pihak ketiga perusahaan dalam pembuatan *dashboard*. Penggunaan pihak ketiga bertujuan untuk mempercepat proses pembuatan *dashboard* di dalam tim, sebab setiap *dashboard* memiliki prioritasnya berbeda-beda. Maka dari itu dibutuhkan pihak ketiga untuk mempercepat pembuatan *dashboard* supaya bisa digunakan segera oleh *user*.

Sejalan dengan penambahan pekerjaan, dilakukannya juga pembuatan *dashboard* Power BI untuk melakukan *tracking* terhadap barang-barang di rumah sakit yang ingin dibeli dan dijual oleh unit kepada unit rumah sakit lain. Secara tidak langsung *dashboard* ini adalah lanjutan dari hasil pembuatan *website* sederhana terjadi prediksi *potential buyer* dan *potential seller*. Selain sebagai *tracker*, *dashboard* juga menjadi acuan untuk melihat aliran dana dari setiap order terhadap barang dan hasil penjualan yang dapat dibuat sebagai alat ukur untuk melihat berapa banyak barang yang sudah *expired*.

### 3.2.3.4. Create InterHU Tracker

Selanjutnya, terdapat *project* pembuatan *dashboard* dengan judul *InterHu Tracker*. *Project* ini dirancang untuk memantau pergerakan barang di seluruh rumah sakit, memberikan visibilitas yang komprehensif terhadap stok dan distribusi. Dengan *dashboard*, *user* dapat melacak barang-barang yang bergerak dari satu lokasi ke lokasi lainnya, serta memantau aliran dana yang terkait dengan barang-barang tersebut.

Tujuan akhir dari *project* adalah untuk mengidentifikasi barang-barang yang akan dijual dan memahami aliran dana yang diperoleh melalui berbagai tahapan proses. *Project* ini terdiri dari 6 tahapan, yang masing-masing dirancang untuk memastikan pemantauan dan pelaporan yang akurat serta efisien. Setiap tahapan memainkan peran penting dalam memberikan gambaran yang jelas tentang status dan pergerakan barang, sehingga memfasilitasi pengambilan keputusan yang lebih jelas.

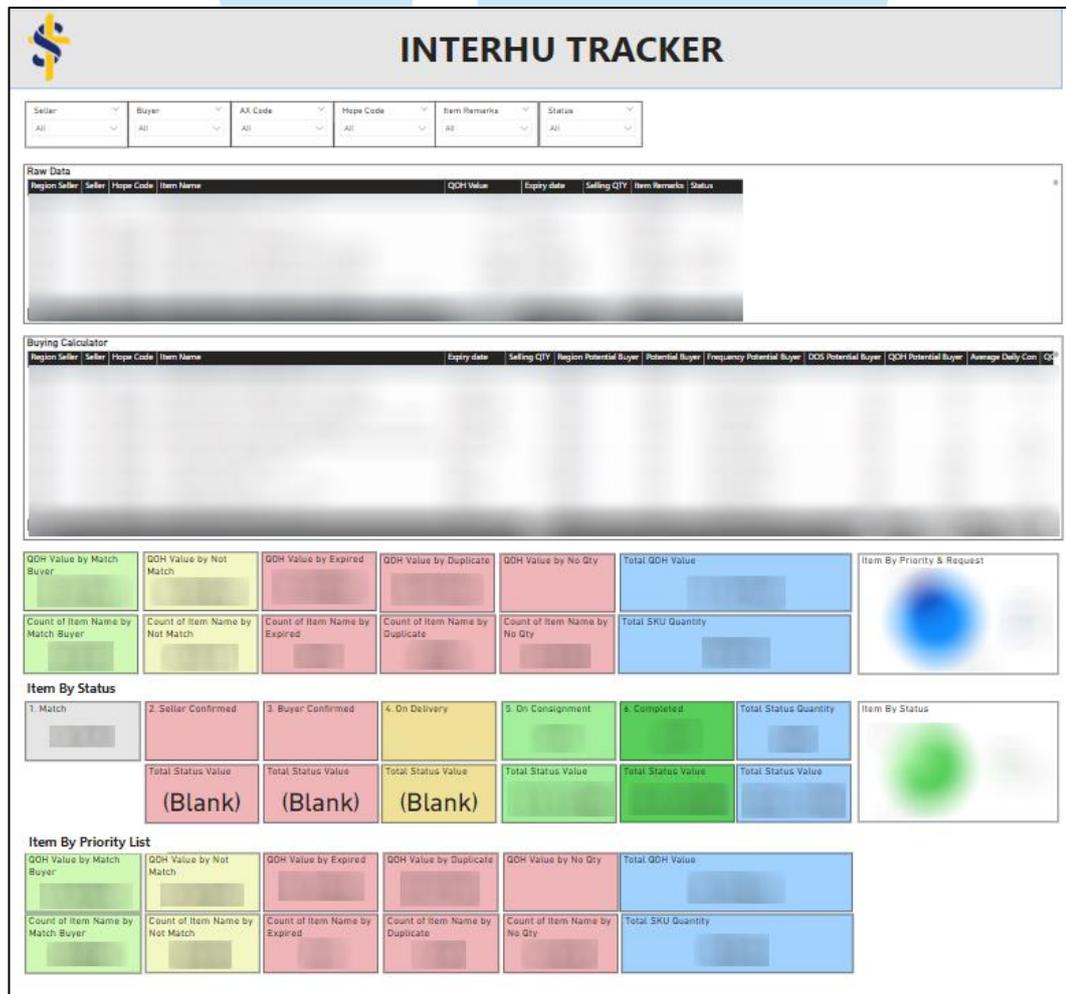
Item By Status						
1. Match	2. Seller Confirmed	3. Buyer Confirmed	4. On Delivery	5. On Consignment	6. Completed	Total Status Quantity
	Total Status Value (Blank)	Total Status Value (Blank)	Total Status Value (Blank)	Total Status Value	Total Status Value	Total Status Value

Gambar 3.27 Tahapan *InterHu Tracker*

Gambar 3.27 merupakan contoh dari ke-6 tahapan dalam *project InterHu Tracker*. Penjelasan dari setiap tahapan adalah sebagai berikut:

- *Match*, merupakan jumlah produk (nama obat) yang terindikasi memiliki frekuensi pergerakan *slow moving*, *not used*, dan *obsolete*.
- *Seller Confirmed*, merupakan tahapan dimana tim *Demand Planning and Inventory Management* menghubungi atau dihubungi oleh *inventory planner* dari setiap rumah sakit yang ingin menjual obat mereka.
- *Buyer Confirmed*, merupakan tahapan dimana tim *Demand Planning and Inventory Management* mencarikan *potential buyers* dari setiap obat yang dimiliki oleh rumah *sellers* untuk dijual kepada *potential buyers*.

- *On Delivery*, merupakan tahapan dimana *sellers* akan mengirimkan obat-obat yang telah ditemukan *buyers*, ke rumah sakit *buyers*.
- *On Consignment*, merupakan tahapan dimana sudah sampai ditempat tujuan (*buyers*) dan selanjutnya akan dilakukan pengecekan terlebih dahulu terhadap barang. Pada saat ini, tim *Demand Planning and Inventory Management* akan menunggu konfirmasi dari rumah sakit (*buyer*) apakah barang sudah sesuai atau belum.
- *Completed*, merupakan tahapan dimana *buyer* telah melakukan konfirmasi bahwa obat yang diterima oleh *seller* telah sesuai dengan yang ada.

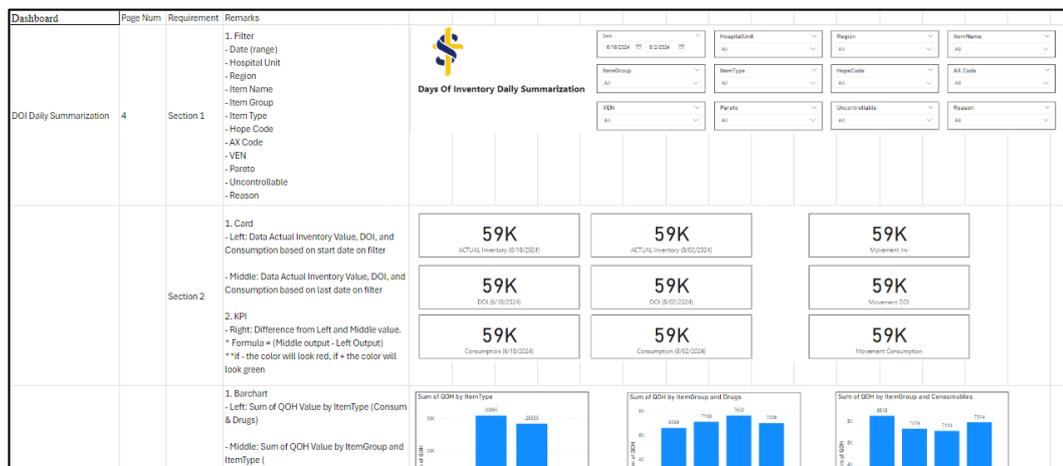


Gambar 3.28 Dashboard InterHu Tracker

Gambar 3.28 merupakan gambar dari keseluruhan *dashboard InterHu Tracker* yang semuanya memiliki fungsi untuk pemantauan dari sisi *buyer* hingga sisi *seller*. Selain dari sisi *buyer* dan *seller*, tujuan dari *dashboard* ini adalah untuk melihat aliran dana dari setiap obat yang sedang bergerak (terjual). Selain itu dalam *dashboard InterHu Tracker* disematkan fitur yang berfungsi untuk memudahkan dalam *update* data. Hal tersebut dapat terjadi sebab digunakannya *import* data menggunakan *web link*. Jadi saat terdapat perubahan yang dilakukan oleh tim *Demand Planning and Inventory Management* maka *user* hanya tinggal menekan *button refresh*. Secara otomatis maka visualisasi akan langsung ter-*update* mengikuti data yang ada.

Setelah penyelesaian *dashboard InterHu Tracker*, proyek baru diberikan untuk pembuatan *dashboard Increment QOH*. Tujuan dari pembuatan *dashboard* adalah untuk melacak jenis obat dan memantau pergerakan konsumsi obat di berbagai wilayah (*Region 1-5*) rumah sakit. *Dashboard* diharapkan dapat memberikan *insight* yang lebih mendalam mengenai stok dan distribusi obat, serta mendukung efisiensi operasional dan pengambilan keputusan yang lebih baik. Namun, proyek pada akhirnya akan dialihkan kepada tim *Business Intelligence* (BI) perusahaan, dan untuk itu diperlukan pembuatan *Change Request* (CR) yang mencakup modifikasi serta penyesuaian seperti berikut:

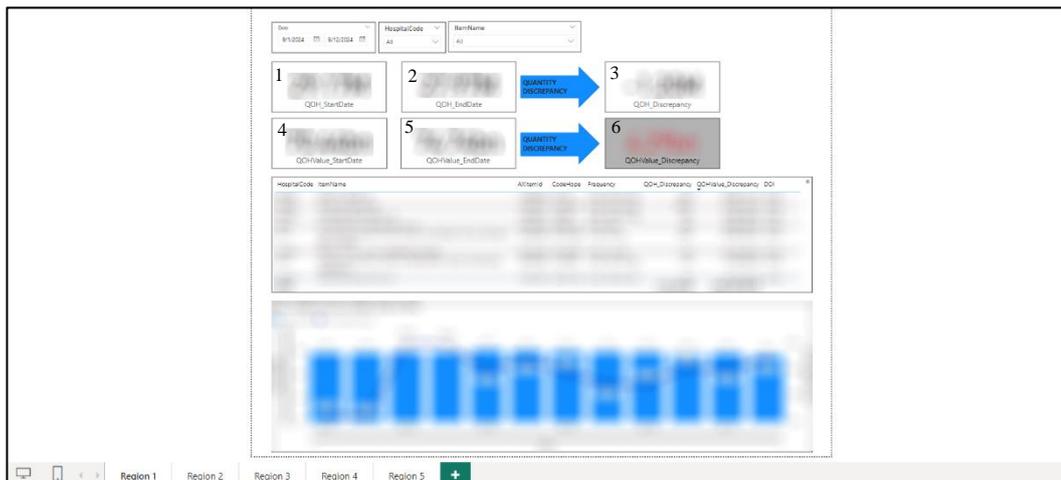
UNIVERSITAS  
MULTIMEDIA  
NUSANTARA



Gambar 3.29 CR Project Increment QOH

Gambar 3.29 menampilkan *Change Request (CR)* yang terkait dengan pembuatan *dashboard Increment QOH*. Data yang digunakan merupakan ilustrasi, bukan angka asli, dengan tujuan untuk memberikan panduan kepada tim *Business Intelligence (BI)* dalam pengembangan *dashboard* di Power BI. *Dashboard* dirancang untuk memantau konsumsi dan stok obat di berbagai wilayah (*region 1-5*) rumah sakit. Visualisasi diatas diharapkan dapat mendukung proses pelaporan yang lebih efisien. Oleh karena itu, *dashboard Increment QOH* sangat penting untuk kebutuhan operasional *Demand Planning and Inventory Management* sehari-hari.

Namun, *user Demand Planning and Inventory Management* membutuhkan *dashboard* segera untuk pelaporan kepada setiap *inventory planner (IP)*. Mengingat urgensi tersebut, pengerjaan proyek dipercepat sebelum dialihkan ke tim *Business Intelligence (BI)*. Tim ini biasanya memerlukan waktu lebih lama karena proses *ticketing* dan antrian prioritas dari *dashboard* yang lainnya. Untuk memenuhi kebutuhan *user*, maka *dashboard Increment QOH* dapat diselesaikan lebih awal. Dengan demikian, pelaporan operasional dapat berjalan tanpa hambatan.



Gambar 3.30 UAT *Dashboard Increment QOH*

Gambar 3.30 menampilkan *User Acceptance Testing* (UAT) yang telah dilakukan untuk *dashboard Increment QOH*. Dashboard ini dilengkapi dengan tiga fitur utama, yaitu *date*, *hospital code*, dan *item name*. Filter-filter tersebut membantu memudahkan *user* dalam mempersonalisasi data yang ingin di cari konsumsinya. Selain itu, dibawahnya terdapat 6 *card* (3 *card* untuk *Quantity on Hand*, 3 untuk *Quantity on Hand values*) yang menyediakan informasi penting bagi pengguna. Terlihat juga bahwa *card* 1 menampilkan nilai *Quantity on Hand (start date)*, yang menunjukkan jumlah awal barang di hari awal terpilih. Sebagai contoh kalau *user* memilih tanggal 1 september maka *card* akan menampilkan nilai *Quantity on Hand* untuk 1 September saja. Untuk rumus yang digunakan untuk Card 1 adalah sebagai berikut:

$$QOH\ StartDate = Calculate( Sum ('tablename'[Quantity On Hand]), 'tablename'[Date] = MIN('tablename'[Date]))$$

*Card* 2 menampilkan *Quantity on Hand (end date)*, yang menunjukkan jumlah akhir barang di hari terakhir pada filter *date*. Dari kedua nilai yang telah didapatkan, selanjutnya akan dihitung *discrepancy* atau selisih untuk mengetahui perubahan *Quantity on Hand*. Jadi tujuan dari adanya *card* 3 adalah untuk melengkapi informasi dengan *detail* tambahan

yang relevan untuk analisis. Dengan adanya fitur ini, *dashboard Increment QOH* mampu memberikan gambaran lengkap mengenai jumlah *Quantity on Hand* di setiap unit rumah sakit. Untuk rumus yang digunakan untuk Card 2 dan 3 adalah sebagai berikut:

$$QOH\ EndDate = Calculate( Sum ('Tablename'[Quantity\ On\ Hand]), 'Tablename'[Date] = MAX('Tablename'[Date]))$$
$$QOH\ Discrepancy = 'Tablename'[QOH\ Startdate] - 'Tablename'[QOH\ Enddate]$$

Selanjutnya terdapat juga *Card* di bawah nomor 1, 2, dan 3. Perbedaan *card* 1, 2, dan 3 dengan *card* 4, 5, dan 6 adalah pada *variabel* yang digunakan. Pada *card* 4, 5, dan 6 menggunakan variabel *QOH\_value*. Untuk rumus yang digunakan pada *card* 4, 5, dan 6 kurang lebih sama dengan *card* 1, 2, dan 3. Hal yang membedakan hanya pada variabel pada rumus *card* 1, 2, dan 3 yang digunakan adalah *variabel Quantity On Hand*. Sedangkan untuk *card* 4, 5, dan 6 menggunakan variabel *QOH\_Value*.

Selain itu untuk *card* 6 yaitu *discrepancy* diterapkan *conditional formatting* dengan rumus seperti berikut:

If value  $\geq$  Min and  $\leq$  Max then green

If value  $\geq$  0.1 and  $\leq$  Max then red

Rumus diatas menunjukkan *conditional formatting* yang diterapkan di Power BI untuk mengidentifikasi perbedaan atau *discrepancy* antara *Quantity on Hand* (QOH) pada *start date* dan *end date*. Pada *conditional formatting* diterapkan untuk menandai nilai jika perbedaan antara QOH berada dalam rentang tertentu. Jika nilai *QOH\_Value* lebih besar atau sama dengan “Min” dan kurang atau sama dengan “Max”, maka tulisan akan diberikan format berwarna hijau. Ini menandakan bahwa tidak ada masalah atau stok barang tetap terjaga dengan baik, sesuai dengan rentang nilai yang diinginkan.

Pada *conditional formatting* kedua, menandai nilai yang berada dibawah ambang batas tertentu. Jika nilai *QOH\_Value* lebih besar atau sama dengan 0.1 dan kurang atau sama dengan “Max”, maka akan diberikan warna merah. Hal tersebut mengindikasikan bahwa adanya *discrepancy* yang signifikan, menunjukkan bahwa stok mengalami penurunan di bawah batas yang diinginkan. Dengan format visual tersebut, *user* dapat dengan mudah mengidentifikasi apakah ada peningkatan, penurunan, atau kondisi kritis dalam jumlah *QOH\_Value* pada waktu tertentu.

Selanjutnya, terdapat visualisasi berupa tabel yang terletak dibawah *card* 4, 5, dan 6. Tabel ini juga menyajikan berbagai informasi yang penting seperti *hospital code*, *item name*, *ax item ide*, *code hope*, *frequency*, *QOH\_discrepancy*, *QOHValue\_discrepancy* dan *doi*. Informasi dalam tabel juga diorganisir sedemikian rupa untuk memudahkan *user* dalam melakukan analisis dan pemantauan dari setiap *hospital unit*.

Untuk meningkatkan fokus, tabel yang dibuat juga dilengkapi dengan filter yang memastikan hanya menampilkan data dari hari terakhir yang dipilih pada filter *date*. Selain itu, akan diterapkan filter tambahan untuk menampilkan 10 barang yang memiliki nilai *discrepancy* tertinggi. Tujuannya adalah untuk mempermudah *user* dalam mengidentifikasi *item-item* yang memerlukan perhatian lebih lanjut dan tindakan korektif berdasarkan frekuensi *item* juga.

Dibawah visualisasi tabel , terdapat juga visualisasi yang menggabungkan *bar chart* dengan *line chart*. Visualisasi ini dirancang untuk memberikan gambaran yang jelas tentang pergerakan *Quantity on hand* (QOH) dan *QOH\_Value* dari hari pertama hingga hari terakhir yang dipilih pada filter tanggal. Kombinasi *bar chart* dan *line chart* memungkinkan *user* untuk membandingkan dua set data secara bersamaan, dan memberikan *insight* yang lebih mendalam mengenai tren dan perubahan dari hari ke hari untuk *Quantity on Hand* dan *QOH\_Value*.

Setelah menyelesaikan pengaturan visualisasi di halaman pertama untuk *region 1*, langkah berikutnya adalah melakukan *duplicate* halaman untuk mencakup hingga *region 5*. Proses duplikasi bertujuan untuk memastikan bahwa format dan analisis yang sama telah diterapkan di seluruh *region*. Selain itu juga tujuannya adalah untuk menjaga konsistensi data dan penempatan visualisasi.

Setelah seluruh perbaikan untuk *region 5* selesai, *dashboard* kemudian akan diserahkan kepada *user*. Tujuannya adalah untuk mendapatkan *feedback* dari *user*. *Feedback* akan digunakan untuk membantu dalam melakukan penyesuaian lebih lanjut untuk meningkatkan fungsionalitas *dashboard* dalam memberikan *insight* mengenai pergerakan *Quantity on Hand* dan juga pergerakan *QOH\_Value* dari hari ke hari.

Selanjutnya, *project* baru telah dimulai berdasarkan permintaan dari tim operasi. Tim operasi menginginkan pembuatan *dashboard* Power BI khusus untuk pelacakan *sub-store* di setiap rumah sakit. Tujuan utama dari *dashboard* adalah untuk memantau dan mengelola *sub-store* dengan cara yang efisien, sehingga dapat mengurangi nilai *days of inventory* (DOI) di setiap *sub-store* yang ada. Pengurangan *sub-store* diharapkan dapat menurunkan nilai DOI secara signifikan, karena semakin banyak *sub-store* yang ada dalam sebuah rumah sakit, semakin tinggi dampaknya terhadap nilai DOI keseluruhan.

*Dashboard* yang dirancang untuk memberikan visibilitas yang jelas mengenai *sub-store* yang beroperasi dan menganalisis efektivitas masing-masing dalam konteks pengelolaan stok obat di rumah sakit. Dengan informasi ini, tim dapat mengidentifikasi *sub-store* mana yang mungkin tidak efisien dan dapat mempertimbangkan untuk dimatikan/menonaktifkan *sub-store*. Dengan langkah yang diambil untuk diharapkan dapat membantu untuk mengoptimalkan pengelolaan stok obat yang memiliki frekuensi pergerakan yang rendah dan mengurangi nilai DOI secara keseluruhan,

yang pada akhirnya dapat meningkatkan efisiensi operasional pada rumah sakit tersebut.



Gambar 3.31 *Dashboard Substore Reduction*

Gambar 3.31 menggambarkan *dashboard* untuk *project substore reduction*. *Dashboard* dirancang untuk memberikan wawasan yang mendalam mengenai *substore* dengan berbagai fitur yang memudahkan *user* dalam mencari *insight* yang relevan. Seperti umumnya pada *dashboard* serupa, terdapat beberapa filter yang memungkinkan *user* untuk menyaring data sesuai kebutuhan, sehingga mempermudah proses analisis dan pengambilan keputusan.

Pada bagian pertama atau *section* 1, terdapat *card* yang menampilkan total *substore* dari seluruh *region*. Total ini dihitung berdasarkan fungsi *Counta*, dimana *substore* dengan nama yang sama akan dihitung sebagai satu entitas. Selanjutnya, data dikelompokkan berdasarkan

*substore* yang bersifat *mandatory* dan *non-mandatory*. Pengelompokan ini membantu dalam pemahaman lebih baik mengenai distribusi dan status masing-masing *sub-store*.

*Sub-store* yang *non-mandatory* kemudian dikelompokkan lebih mendalam menjadi 2 kategori, yaitu *Non-Pharmacy* dan *Pharmacy*. Untuk kategori *Non-Mandatory Non-Pharmacy*, *dashboard* akan menampilkan nilai keseluruhan *sub-store* dengan mengeluarkan *item type Drugs* dan *Consumables* dari perhitungan. Sebaliknya, untuk kategori *Non-Mandatory Pharmacy*, nilai yang ditampilkan akan memberikan acuan bagi *user* dalam menentukan *sub-store* mana yang perlu dinonaktifkan.

Di bawah *section 1*, terdapat *section 2* yang menampilkan *barchart* dengan nilai dalam persentase. *Barchart* ini digunakan untuk mengidentifikasi *sub-store* mana yang harus ditutup, dengan indikator warna merah menunjukkan *sub-store* yang perlu ditutup. Di sebelah kanan *barchart*, terdapat tabel yang menampilkan *itemname* yang berkontribusi pada keputusan tersebut, sehingga memberikan informasi tambahan mengenai dampak item terhadap status *sub-store*.

*Section 3*, yang terletak di bawah *section 2*, memiliki fitur yang mirip dengan *section 2* tetapi dengan fokus yang berbeda. Sementara *section 2* berfokus pada *Pharmacy*, *section 3* lebih menekankan pada kategori *Non-Pharmacy*. Perbedaan ini memastikan bahwa analisis dan rekomendasi yang diberikan sesuai dengan kategori masing-masing *sub-store*.

Dengan pembagian yang jelas antara kategori dan fitur analisis, *dashboard* dapat memberikan panduan komprehensif dalam pengelolaan *sub-store*. Selain itu memungkinkan *user* untuk membuat keputusan yang lebih terinformasi dan strategis mengenai *sub-store* mana yang sebaiknya dipertahankan atau ditutup untuk mengoptimalkan pengelolaan *days of inventory* pada setiap rumah sakit.

### 3.3. Kendala yang Ditemukan

1. Dalam menggunakan Power BI, sering kali terjadi kesalahan pada *dashboard* akibat kurangnya pemahaman dalam penulisan dan penggunaan rumus, terutama DAX (*Data Analysis Expressions*). Hal ini mengakibatkan data yang ditampilkan menjadi tidak akurat atau visualisasi tidak berjalan sesuai dengan harapan *user*.
2. Kurangnya pemahaman mengenai prosedur pembuatan *Change Request* (CR) dan pembacaan *Functional Specification Document* (FSD) sering kali menghambat proses perubahan dan pengembangan *dashboard* Power BI, serta tidak jarang menimbulkan miskomunikasi dengan tim.
3. Salah satu tantangan dalam membuat *dashboard* Power BI adalah memilih kombinasi warna yang sesuai. Pemilihan warna yang kurang tepat sering kali membuat visualisasi sulit dipahami oleh *user*.
4. Seringkali terjadi kesalahan dalam visualisasi atau laporan di Power BI karena pemahaman yang kurang mengenai hubungan antar tabel (*relationship*) dalam database, seperti *many-to-many* atau *one-to-many*. Hal ini dapat menyebabkan data yang ditampilkan menjadi tidak sesuai dengan yang diharapkan.

### 3.4. Solusi atas Kendala yang Ditemukan

1. Untuk mengatasi masalah ini, penting untuk memperdalam pemahaman mengenai rumus-rumus yang digunakan oleh tim *Demand Planning and Inventory Management* dalam menghitung setiap variabel. Selain itu, solusi lain adalah melakukan tes pada dataset yang lebih sederhana sebelum menerapkannya ke dataset yang lebih kompleks. Seperti mengambil hanya 1 rumah sakit dan 1 sub-store sebagai data untuk melakukan validasi.
2. Solusinya adalah dengan bertanya kepada rekan tim untuk melakukan pelatihan singkat terkait proses pembuatan *Change Request* dan pemahaman mengenai FSD.
3. Untuk mengatasi kendala ini, penting untuk terus melakukan konsultasi dengan *user* dan juga memahami setiap arti warna yang sudah ditetapkan oleh tim *Demand Planning and Inventory Management* dalam membuat

*dashboard* Power BI. Selain itu selalu melakukan uji coba dan meminta *feedback* dari *user* sebelum dilakukan *publish* pada *dashboard* Power BI kedalam *workspace* tim.

4. Solusinya adalah dengan cara memperdalam pemahaman mengenai konsep database *relationship*, terutama dengan cara menonton video di youtube atau membaca blog yang berkaitan dengan database seperti medium.



UMMN

UNIVERSITAS  
MULTIMEDIA  
NUSANTARA