

## LAMPIRAN A

### TERM OF REFERENCE PROYEK NOMOR 1

#### TERM OF REFERENCE Project PTSP Kemenag Kota Bekasi

##### 1. Pendahuluan

Pelayanan Terpadu Satu Pintu (PTSP) dilatar belakangi hal berikut:

- Pentingnya peningkatan *service level* Kementerian Agama Kota Bekasi dalam memberikan pelayanan prima kepada masyarakat.
- Pentingnya meminimalisir kontak langsung masyarakat dengan petugas dimasa pandemi, layanan Kemenag Kota Bekasi dapat diakses kapan saja dan dimana saja.
- Pentingnya penyajian data bagi pimpinan dan stakeholder terkait yang akurat dan riil time untuk menjadi rujukan pengambilan keputusan.
- Pentingnya penyajian data bagi pimpinan dan stakeholder terkait yang akurat dan riil time untuk menjadi rujukan pengambilan keputusan.

##### 2. Ruang Lingkup

Sistem PTSP KEMENAG KOTA BEKASI yang akan dikembangkan melingkupi hal sebagai berikut:

- a) Website *frontend* untuk masyarakat atau pemohon
- b) Dashboard untuk petugas *frontdesk* and seksi terkait
- c) *Apps* antrian *offline* yang terintegrasi dengan PTSP

##### 3. Waktu dan Biaya Pengerjaan

Proyek ini akan dikerjakan dalam 5 (lima) Sprint, dimana dalam satu Sprint memakan waktu 14 hari kalender, sehingga total waktu yang dibutuhkan adalah 70 hari kalender. Gambar berikutnya menunjukkan *project milestone* yang merupakan bagian dari manajemen waktu proyek, yaitu proses merencanakan, menyusun, dan mengendalikan jadwal kegiatan proyek, dimana dalam perencanaan dan penjadwalannya telah disediakan pedoman yang spesifik untuk menyelesaikan aktivitas proyek dengan lebih cepat dan efisien. *Backlog* atau spesifikasi atau fitur-fitur atau aktivitas proyek yang harus dikerjakan tetap mengacu ke pada bagian ruang lingkup (*project scope*).



Terdapat 3 (tiga) tim pengembang (*developer*) yang masing-masing dari komposisi tim sewaktu-waktu dapat berubah atau perpindahan anggota tim mengikuti permasalahan yang muncul pada *life cycle* pengembangan. Tabel 1 adalah *rate* gaji dalam perbulan dan Tabel 2 berisikan kebutuhan *manpower* dari masing-masing tim ketika proyek dimulai.

Tabel 1. *Rate gaji perbulan*

Type	Rate Monthly (IDR)	Average Man-Day (22 Working Days)
UI/UX Designer	7.500.000	341.000
DevOps	8.000.000	364.000
Junior Frontend Developer	8.500.000	386.000
Junior Backend Developer	9.000.000	409.000
Business & System Analyst	10.000.000	455.000
Senior Frontend Developer	12.000.000	545.000
Senior Backend Developer	13.000.000	591.000
Project Manager / Scrum Master	17.000.000	773.000

Tabel 2. Estimasi kebutuhan *manpower*

Man-Power	Man-Day (IDR)	Working-Day	Subtotal (IDR)	Remark	
Team 1	Business & System Analyst	455.000	56	25.480.000	Sprint 1, Sprint 2, Sprint 4, and Sprint 5
	UI/UX Designer	341.000	33	11.253.000	Sprint 1, Sprint 3, and extra 5 days
	DevOps	364.000	56	20.384.000	Sprint 2, Sprint 3, Sprint 4, and Sprint 5
Team 2	Senior Backend Developer	591.000	70	41.370.000	All Sprint
	Junior Frontend Developer	386.000	70	27.020.000	All Sprint
Team 3	Junior Backend Developer	409.000	70	28.630.000	All Sprint
	Senior Frontend Developer	545.000	70	38.150.000	All Sprint
Project Manager / Scrum Master	773.000	70	54.110.000		
<b>Grand Total</b>			<b>Rp 246.397.000</b>		

Untuk proyek yang mengimplementasikan kerangka kerja Scrum tidak mengharuskan anggotanya untuk mengikuti aturan dari panduan Scrum secara keseluruhan. Setiap penyelesaian masalah mengacu kepada **Sprint** yang memiliki durasi yang konsisten sepanjang *life cycle* pengembangan produk, karena Sprint merupakan inti dari proses kerangka kerja **Scrum**.

#### 4. System Platform

- Persyaratan perangkat lunak untuk komputer Server
- Sistem Operasi Linux Ubuntu Server versi 20.04 edisi 64-bit
  - Nginx Web Server

- PHP versi 7.4
- Database MySQL versi 5.7 or higher
- IP Public pointing ke sub atau nama domain
- Sertifikat SSL (Secure Socket Layer)
- SMTP Mail Server untuk email notifikasi
- Using HA (*high availability*) Database Server

Persyaratan minimum untuk para pengguna

- Sistem Operasi Windows atau Mac IOS atau Linux dengan browser Google Chrome atau Firefox dengan javascript yang diaktifkan

##### 5. Hardware Platform

Server yang digunakan Dell PowerEdge R240 seharga **Rp 21.600.000**, dengan spesifikasi sebagai berikut:

- ✓ Processor Intel Xeon E-2224 (3.4GHz 8M cache 4C/4T turbo 71W)
- ✓ Chassis 4 x 3.5 Hotplug Hard Drives
- ✓ RAM 16GB (2 x 8GB) 3200MT/s DDR4 ECC UDIMM
- ✓ 1TB 7.2K RPM SATA 6Gbps 3.5in Hotplug Hard Drive
- ✓ PERC H330 RAID Controller, DVD+/-RW SATA, Single Cabled Power
- ✓ Supply 250W, Dual Port 1GB, iDRAC9 Basic
- ✓ No Operating System

Server Environment dibagi menjadi menjadi 2 (dua) bagian, dengan Proxmox *Virtual Environment* (VE):

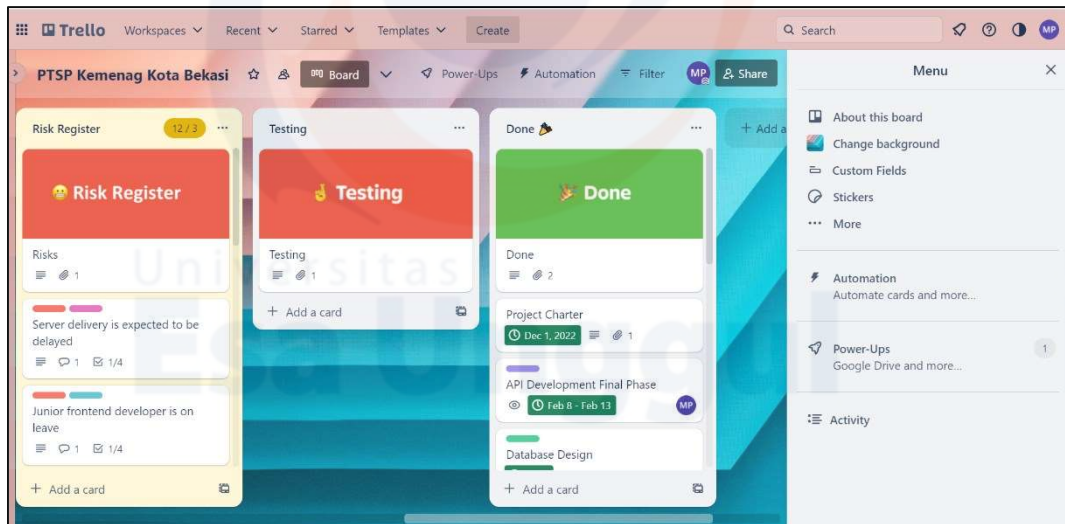
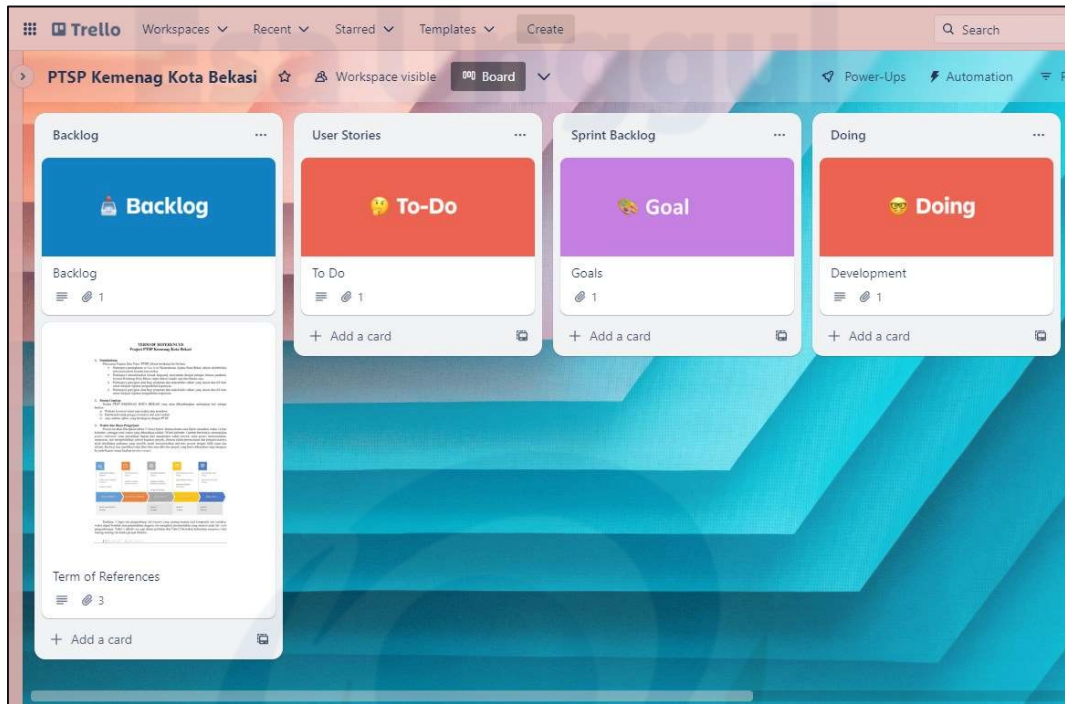
- Lingkungan Server Pengembangan (*Development*), dengan spesifikasi sebagai berikut

Specification	
Processor	1 core 64 bit
Memory	4 GB
Storage	200 GB for System & Data

- Lingkungan Server *Production* (Live), dengan spesifikasi sebagai berikut

Specification	
Processor	3 core 64 bit
Memory	12 GB
Storage	800 GB for System & Data

## LAMPIRAN B PEMODELAN TRELLO DI PROYEK 1



## LAMPIRAN C

### KLASIFIKASI RISIKO DENGAN *MACHINE LEARNING*

```
from google.colab import drive
drive.mount('/content/drive')
```

Mounted at /content/drive

```
import pandas as pd
import os
import json
import numpy as np
```

#### Connect to Kaggle

```
from google.colab import files

uploaded = files.upload()

for fn in uploaded.keys():
    print('User uploaded file "{name}" with length {length} bytes'.format(
        name=fn, length=len(uploaded[fn])))

# Then move kaggle.json into the folder where the API expects to find it.
!mkdir -p ~/.kaggle/ && mv kaggle.json ~/.kaggle/ && chmod 600 ~/.kaggle/kaggle.json
```

```
Choose Files kaggle.json
• kaggle.json(application/json) - 71 bytes, last modified: 6/14/2022 - 100% done
Saving kaggle.json to kaggle.json
User uploaded file "kaggle.json" with length 71 bytes
```

```
dir = f'drive/MyDrive/Colab Notebooks'
files = os.listdir(dir)
```

```
!kaggle kernels output marzukipiliang/803k-issues-keras-risk-matrix -p /content/dir
```

```
Output file downloaded to /content/dir/categories_priority.png
Output file downloaded to /content/dir/df_keras_risk_matrix.pickle
Output file downloaded to /content/dir/keras_cluster.png
Output file downloaded to /content/dir/priority_chart.png
Output file downloaded to /content/dir/risk_matrix.png
Kernel log downloaded to /content/dir/803k-issues-keras-risk-matrix.log
```

```
!kaggle kernels output marzukipiliang/803k-issues-keras-classify-risk -p /content/dir
```

```
Output file downloaded to /content/dir/lstm_classification.h5
Output file downloaded to /content/dir/lstm_matrix.png
Output file downloaded to /content/dir/model_history.png
Output file downloaded to /content/dir/naive_matrix.png
Output file downloaded to /content/dir/priority_chart.png
Output file downloaded to /content/dir/svm_matrix.png
Output file downloaded to /content/dir/train_label.png
Output file downloaded to /content/dir/validation_label.png
Kernel log downloaded to /content/dir/803k-issues-keras-classify-risk.log
```

```
import tensorflow as tf
import tensorflow.python.keras
import seaborn as sns
import pickle
from keras.preprocessing.text import Tokenizer
from keras.utils import pad_sequences
from datetime import datetime
from sklearn.metrics import classification_report, confusion_matrix, accuracy_score
```

#### Load LSTM Model

```
MODEL_PATH = f'/content/dir/lstm_classification.h5'
```

```
lstm_model = tf.keras.models.load_model(MODEL_PATH)
```

```
# Risk Priority
#####
# 0 -> Extreme Risk
# 1 -> High Risk
# 2 -> Medium Risk
# 3 -> Small (Low) Risk
#####
CLASSES = ['Extreme', 'High', 'Medium', 'Small']
```

```
MAX_LEN = 185
```

```
with open(f'/content/dir/df_keras_risk_matrix.pickle', 'rb') as fp:
    df = pickle.load(fp)
```

```
tokenizer = Tokenizer(num_words=50000, oov_token='oov')
```

```
tokenizer.fit_on_texts(df['cleaned'].tolist())
```

```
def lstm_priority(corpus):
    tkn = tokenizer.texts_to_sequences(corpus)
    tkn_padded = pad_sequences(tkn, padding='post', truncating='post', maxlen=MAX_LEN)
    prd = lstm_model.predict(tkn_padded)
    return CLASSES[int(np.argmax(prd, axis=1))]
```

```
# test model
print(lstm_priority(['good performance']))

1/1 [=====] - 1s 1s/step
Small
```

## ↳ Parsing Trello Data

```
f = open(os.path.join(dir, 'IsoGsgt8.json'),)
data = json.load(f)
```

```
lists = {l['id']: l['name'] for l in data['lists']}
users = {u['id']: u['fullName'] for u in data['members']}
labels = {l['id']: l['name'] for l in data['labels']}
```

```
parsed_cards = [
    {
        "name": c['name'],
        "list": lists[c['idList']],
        "description": c['desc'],
        "members": [u for k, u in users.items() if k in c['idMembers']],
        "labels": [l for k, l in labels.items() if k in c['idLabels']]
    } for c in data['cards']]
```

```
output = {
    "board_data": {
        "name": data['name'],
        "url": data['shortUrl']
    },
    "cards": parsed_cards
}
```

```
output['cards']
```

```
{
  'description': 'The results of the trial and penetration testing by the central Ministry of Religion have the potential to finish the project not on time, if the results obtained are not up to standard',
  'members': [],
  'labels': ['Risk Avoidance'],
  'name': 'Testing',
  'list': 'Testing',
  'description': 'Cards come here after being code reviewed, and we're in a place for QA and fixing things QA finds. Design review also happens at this stage.\n\nIf we have too many cards here, we should throw a flag and figure out whether we're building things that take too many passes of QA, or if we're bottlenecked on QA time.',
  'members': [],
  'labels': [],
  'name': 'Done',
  'list': 'Done',
  'description': 'We shipped it to production!',
  'members': [],
  'labels': [],
  'name': 'Project Charter',
  'list': 'Done',
  'description': 'Project charter here is represented by term of references',
  'members': [],
  'labels': [],
  'name': 'API Development Final Phase',
  'list': 'Done',
  'description': '',
  'members': ['Marzuki Pilliang'],
  'labels': ['Backend Developer'],
  'name': 'Database Design',
  'list': 'Done',
  'description': 'Database Design for PTSP',
  'members': [],
  'labels': ['BA & System Analyst'],
  'name': 'The specification of Server',
  'list': 'Done',
  'description': 'The server used by the Dell PowerEdge R240 costs IDR 21,600,000, with the following specifications:\n\n Processor Intel Xeon E-2224 (3.4GHz 8M cache 4C/4T turbo 71W)\n- Chassis 4 x 3.5 Hotplug Hard Drives\n- RAM 16GB (2 x 8GB) 3200MT/s DDR4 ECC UDIMM\n- 1TB 7.2K RPM SATA 6Gbps 3.5in Hotplug Hard Drive\n- PERC H330 RAID Controller, DVD+-RW SATA, Single Cabled Power\n- Supply 250w, Dual port 1GB, IDRAC9 Basic\n- No Operating System\n\n\nu200c',
  'members': [],
  'labels': ['Scrum Master'],
  'name': 'Flowchart of offline queue apps',
  'list': 'Done',
  'description': 'Todo Business Analyst',
  'members': [],
  'labels': ['BA & System Analyst'],
  'name': 'Flowchart of PTSP Website Frontend',
  'list': 'Done',
  'description': 'Flowchart of PTSP Kementerian Agama Kota Bekasi',
  'members': [],
  'labels': ['BA & System Analyst'],
  'name': 'UI/UX Design for Website Frontend',
  'list': 'Done',
  'description': 'User Interface and User Experience (UI/UX) Design for Website Frontend',
  'members': [],
  'labels': ['UI/UX Designer'],
  'name': 'API Development Phase 1',
  'list': 'Done',
  'description': 'API Development Phase 1 for Website Frontend and Backoffice user.\n\n\nTodo by Backend Developer',
  'members': [],
  'labels': ['Backend Developer']
}
```

```
df = pd.DataFrame(output['cards'])
```

```
list(df.columns)
['name', 'list', 'description', 'members', 'labels']
```

```
risks = df.loc[(df['description'] != '')
 & (df['list'] == 'Risk Register')
 & (df['name'] != 'Risks')]['description'].tolist()
```

```
df_risk = pd.DataFrame(risks, columns=['risk'])
df_risk
```

	risk
0	Server delivery from vendors is expected to be...
1	The developer is on holiday for Christmas and ...
2	Requests for changes to flow from the Administ...
3	In this first sprint there is a possibility th...
4	The Product Owner was confirmed that the domai...
5	The main link internet was not installed yet, ...
6	The request for the Email Blast service was re...
7	The Product Owner requested a change to the FL...
8	SSL security cannot be activated, because the ...
9	There is no captcha validation yet on the new ...
10	The results of the trial and penetration testi...

### ▼ Risk Priority Classification using Model LSTM

```
df_risk['prediction'] = df_risk['risk'].apply(lambda x: lstm_priority([x]))
```

```
1/1 [=====] - 0s 251ms/step
1/1 [=====] - 0s 40ms/step
1/1 [=====] - 0s 42ms/step
1/1 [=====] - 0s 39ms/step
1/1 [=====] - 0s 42ms/step
1/1 [=====] - 0s 45ms/step
1/1 [=====] - 0s 39ms/step
1/1 [=====] - 0s 54ms/step
1/1 [=====] - 0s 40ms/step
1/1 [=====] - 0s 46ms/step
1/1 [=====] - 0s 45ms/step
```

```
df_risk
```

1 to 11 of 11 entries

index	risk	prediction
0	Server delivery from vendors is expected to be delayed because they are waiting for 8 GB of RAM which will be used for upgrading specs	Small
1	The developer is on holiday for Christmas and new year	Medium
2	Requests for changes to flow from the Administration section, the flow of outgoing mail and incoming mail must go through the front-desk and the knowledge of the heads of sub-sections	Small
3	In this first sprint there is a possibility that the number of DoD user stories will be small, some user stories will be included in the next Sprint	Medium
4	The Product Owner was confirmed that the domain will be ready in two weeks	Small
5	The main link internet was not installed yet, the client still looking for best price	Small
6	The request for the Email Blast service was rejected by the provider because the email DKIM requirements have not been fulfilled	Small
7	The Product Owner requested a change to the Flowchart for the Incoming and Outgoing Mail flows in Sprint #3	Medium
8	SSL security cannot be activated, because the client has not purchased a paid SSL	Small

✓ 0s completed at 9:27PM

**LAMPIRAN D**  
**DAFTAR PERTANYAAN DALAM WAWANCARA**

Aspek	Pertanyaan Wawancara	Responden
<b>PBI</b>	1) Berapa banyak <i>Product Backlog Item</i> (PBI) yang direncanakan dalam proyek ini? 2) Apakah ada perubahan jumlah PBI selama proyek berjalan? 3) Berapa banyak PBI yang masuk ke dalam DoD ( <i>Definition of Done</i> ) pada tiap-tiap Sprint?	PO dan PM
<b>Sprint</b>	1) Berapa banyak Sprint yang direncanakan dalam proyek ini? 2) Berapa hari durasi dalam 1 (satu) Sprint? 3) Apakah ada perubahan jumlah Sprint selama proyek berjalan?	PM
<b>Biaya</b>	1) Berapa besar (rupiah) yang dianggarkan dalam proyek ini? 2) Berapa besar (rupiah) anggaran yang diserap dalam tiap-tiap Sprint?	PO dan PM
<b>Kerangka Kerja Manajemen Risiko</b>	1) Sudah berapa tahun metodologi Scrum diterapkan dalam proyek pengembangan perangkat lunak? 2) Apakah sebelumnya pernah menerapkan manajemen risiko dalam metodologi Scrum? 3) Sudah berapa tahun menggunakan Trello® sebagai alat manajemen proyek? 4) Pertanyaan berikut ini berkaitan dengan kerangka kerja manajemen risiko yang diusung. Jawab dengan (SETUJU / TIDAK SETUJU) <ul style="list-style-type: none"> <li>a. Alur kerangka kerja ini (Gambar 32 dan Gambar 33) mudah untuk diikuti?</li> <li>b. Kerangka kerja ini (Gambar 32 dan Gambar 33) mudah untuk diterapkan?</li> <li>c. Pemodelan <i>Board</i> Trello® (LAMPIRAN B) dengan model klasifikasi prioritas risiko pada kerangka kerja ini (Gambar 32 dan Gambar 33), membuat risiko-risiko mudah untuk dipantau dan dimitigasi secara terjadwal?</li> <li>d. Kerangka kerja manajemen risiko ini (Gambar 32 dan Gambar 33) merupakan faktor pendukung keberhasilan proyek?</li> <li>e. Akan menggunakan kerangka kerja ini (Gambar 32 dan Gambar 33) pada proyek selanjutnya?</li> </ul>	PO, PM dan SD



LAMPIRAN E  
**TERM OF REFERENCE PROYEK NOMOR 2**

**TERM OF REFERENCE  
Project Retail Management System**

**1. Pendahuluan**

*Retail Management System* (RMS) adalah sistem penting yang harus digunakan pada perusahaan yang ingin lebih sukses. RMS adalah cara untuk mengendalikan dan mengawasi persediaan, mengendalikan jumlah produk yang tersedia untuk dibeli oleh pelanggan. Persediaan terdiri dari kombinasi barang, bahan mentah dan produk jadi, dan manajemen yang baik dari barang-barang ini, mengarah pada kelancaran operasi bisnis tertentu. Bisnis skala menengah dan besar menggunakan berbagai jenis perangkat lunak *retail management system* yang memiliki berbagai fitur seperti mengontrol persediaan, manajemen pesanan, label barcode, dan pemindaian, dan lainnya.

PT. Japfa Comfeed membutuhkan sebuah *Retail Management System* untuk membantu mengoptimalkan operasinya melalui manajemen yang efektif dan pelaksanaan proses yang mulus seperti pembelian, distribusi, inventaris, dan keuangan. Sebuah sistem yang mencakup ujung ke ujung, dengan fungsi manajemen persediaan yang dinilai sebagai salah faktor keberhasilan suatu sistem.

**1.1. Tujuan**

Pembangunan RMS ini bertujuan untuk membantu dalam memaksimalkan keuntungan organisasi dan untuk menjaga likuiditas perusahaan pada tingkat standar. Oleh karena itu dalam arus bisnis yang kompetitif ini, akurasi yang tinggi dalam peramalan persediaan adalah salah satu faktor utama, yang ingin dicapai oleh perusahaan, untuk meningkatkan tingkat keuntungan mereka sekaligus mengurangi biaya persediaan dan administrasi.

**1.2. Manfaat**

Adapun manfaat pembangunan RMS adalah sebagai berikut:

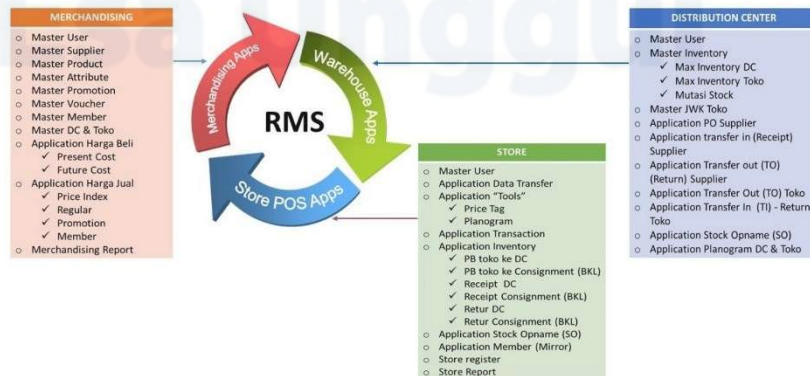
- Menjaga kelancaran aktivitas penjualan dan pembelian barang
- Mengindari hilangnya kesempatan menjual dan mendapatkan profit dari suatu produk
- Menghasilkan penghematan moneter, daya saing yang lebih besar, hubungan dengan supplier yang meningkat, dan kepuasan pelanggan
- Lebih akurat dalam *forecasting* penjualan eceran

**2. Ruang Lingkup**

Guna memudahkan pengembang sistem, proyek RMS dipisah-pisah menjadi 3 bagian antara lain *Merchandising Apps*, *Warehouse (Distribution Center) Apps* dan *Store Apps*. Berikut ini detail fitur dari masing-masing bagian:

- a) **Merchandising Apps** (Web base) → Master User; Master Supplier; Master Product; Master Attribute; Master Promotion; Master Voucher; Master Member; Master DC dan Toko; Modul Harga Beli; Modul Harga Jual; Dashboard Report
- b) **Warehouse Apps** (Web base) → Master User; Master Inventory; Master JWK (Jadwal Waktu Kirim Toko); Modul PO Supplier; Module Transfer In (Good Receipt) dari Supplier; Modul Transfer Out (Return) ke Supplier; Modul Transfer Out (Outbound) ke Toko; Modul Transfer In (Return) dari Toko; Modul Stock Opname; Modul Planogram DC dan Toko
- c) **Store Apps** (Android Native) → Master User; Modul Data Transfer; Module Price Tag dan Planogram; Modul Penjualan; Modul Inventory; Modul Persediaan; Modul Stock Opname; Store Register dan Dashboard Laporan

Tiap-tiap bagian tersebut merupakan satu kesatuan yang tidak dapat dipisahkan (berdiri sendiri). Hubungan antar bagian tersebut dapat dilihat pada Gambar 1 dan Gambar 2.

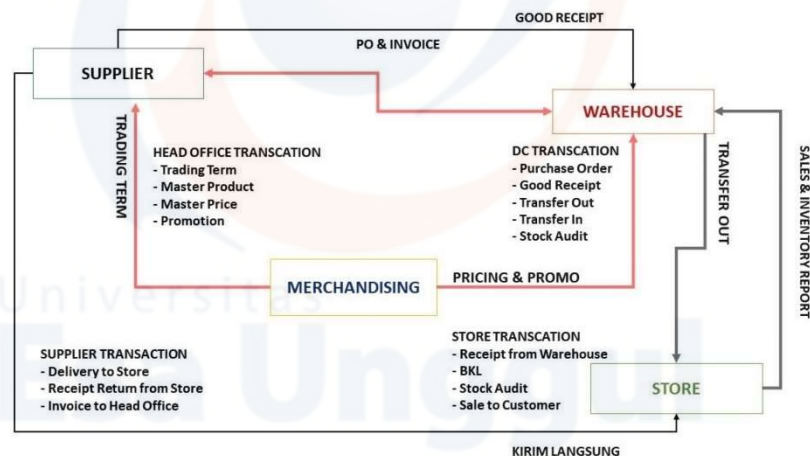


Gambar 1. Fitur-fitur RMS

### 3. Desain Sistem

Tahapan desain menerjemahkan kebutuhan perangkat lunak ke dalam model. Desain sistem merupakan salah satu tahapan pengembangan perangkat lunak yang menjadi jembatan dari analisis dan implementasi program. Beberapa kegiatan dari tahap ini antara lain: Mendefinisikan tujuan desain; Mendefinisikan subsistem; Pemetaan subsistem ke dalam platform yang digunakan; Pengelolaan *persistent* data; Mendefinisikan kendali akses; Mendefinisikan kendali aliran (*control flow*); Mendefinisikan *boundary condition*.

Dari desain sistem ini akan dihasilkan detail teknis sehingga memudahkan *developer* dalam mengimplementasikan atau membuat perangkat lunak. Desain sistem dari RMS dapat dilihat pada Gambar 2.



Gambar 2. Desain utama sistem

### 3.1. Aliran Persediaan

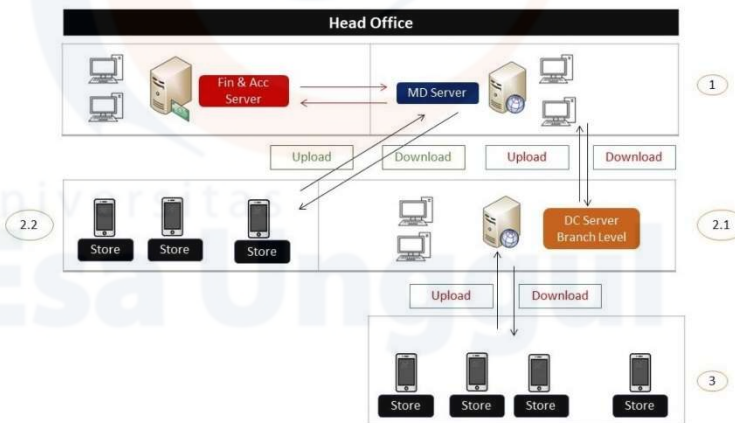
Ketika barang benar-benar dijual oleh penjual kepada pelanggannya atau dapat dikatakan waktu aktual kapan barang dijual akan dianggap sebagai aliran fisik persediaan. Pada Gambar 3 merupakan ilustrasi aliran persediaan dalam RMS.



Gambar 3. Aliran persediaan

### 3.2. Topologi

Topologi merupakan layout atau arsitektur atau diagram jaringan komputer. Topologi merupakan aturan bagaimana menghubungkan komputer, server, workstation, router dan switch dan lainnya secara fisik. Topologi juga mengatur bagaimana tiap-tiap perangkat tersebut saling berkomunikasi melalui media transmisi data. Berikut ini Topologi yang akan digunakan dalam RMS terlihat pada Gambar 4.



Gambar 4. Topologi dalam RMS

#### 4. Metodologi

Proyek ini akan menggunakan metode Agile dengan kerangka Scrum, dimana Scrum melibatkan kelompok orang secara kolektif yang memiliki semua keterampilan dan keahlian untuk melakukan pekerjaan dan berbagi atau memperoleh keterampilan yang dibutuhkan. Proyek ini akan terdiri dari seorang Product Owner perwakilan dari PT. Japfa Comfeed, seorang Scrum Master dan 3 tim pengembang (*developer*).

Total *manpower* yang dibutuhkan adalah 8 orang dengan rincian sebagai berikut:

- 1 orang Scrum Master
- 1 orang System Analyst
- 1 orang UI/UX Designer
- 1 orang DevOps
- 2 orang Junior Developer (Frontend dan Backend)
- 2 Senior Developer (Frontend dan Backend).

#### 5. Waktu dan Biaya Pengerjaan

Proyek ini akan dikerjakan dalam 5 Sprint, dimana dalam satu Sprint memakan waktu 4 minggu atau 28 hari kalender, sehingga total waktu yang dibutuhkan adalah 140 hari kalender. Gambar 5 menunjukkan *project milestone* yang merupakan bagian dari manajemen waktu proyek, yaitu proses merencanakan, menyusun, dan mengendalikan jadwal kegiatan proyek, dimana dalam perencanaan dan penjadwalannya telah disediakan pedoman yang spesifik untuk menyelesaikan aktivitas proyek dengan lebih cepat dan efisien. Backlog atau spesifikasi atau fitur-fitur atau aktivitas proyek yang harus dikerjakan tetap mengacu ke pada bagian ruang lingkup (*project scope*).



Gambar 5. Project milestone

Terdapat 3 tim pengembang (*developer*) yang masing-masing dari komposisi tim sewaktu-waktu dapat berubah atau perpindahan anggota tim mengikuti permasalahan yang muncul pada *life cycle* pengembangan. Tabel 1 adalah *rate* gaji dalam perbulan dan Tabel 2 berisikan kebutuhan *manpower* dari masing-masing tim ketika proyek dimulai.

Tabel 1. *Rate* gaji perbulan

Type	Rate Monthly (IDR)	Average Man-Day (22 Working Days)
UI/UX Designer	7.500.000	341.000
DevOps	8.000.000	364.000
Junior Frontend Developer	8.500.000	386.000

Type	Rate Monthly (IDR)	Average Man-Day (22 Working Days)
Junior Frontend Developer	8.500.000	386.000
Junior Backend Developer	9.000.000	409.000
System Analyst	10.000.000	455.000
Senior Frontend Developer	12.000.000	545.000
Senior Backend Developer	13.000.000	591.000
Scrum Master	17.000.000	773.000

Tabel 2. Estimasi kebutuhan *manpower*

Man-Power	Man-Day (IDR)	Working-Day	Subtotal (IDR)	Remark	
Team 1	System Analyst	455.000	112	50.960.000	Sprint 1, Sprint 2, Sprint 4 and Sprint 5
	UI/UX Designer	341.000	66	22.506.000	Sprint 1, Sprint 2, and extra 10 days
	DevOps	364.000	112	40.768.000	Sprint 2, Sprint 3, Sprint 4 and Sprint 5
Team 2	Senior Backend Developer	591.000	140	82.740.000	
	Junior Frontend Developer	386.000	140	54.040.000	
Team 3	Junior Backend Developer	409.000	140	57.260.000	
	Senior Frontend Developer	545.000	140	76.300.000	
Scrum Master	773.000	140	108.220.000		
<b>Grand Total</b>				<b>Rp 492.794.000</b>	

#### 6. System Platform

Persyaratan perangkat lunak untuk komputer Server

- Sistem Operasi Linux Ubuntu Server versi 20.04 edisi 64-bit
- Nginx Web Server
- PHP versi 7.4
- Database MySQL versi 5.7 or higher
- IP Public pointing ke sub atau nama domain
- Sertifikat SSL (Secure Socket Layer)
- SMTP Mail Server untuk email notifikasi
- Using HA (*high availability*) Database Server (rekomendasi)

Persyaratan minimum untuk para pengguna

- Sistem Operasi yang mempunyai browser Chrome atau Firefox dengan javascript yang diaktifkan
- Sistem Operasi Android 6.0 Mashmallow

### 7. Hardware Platform

Server Environment dibagi menjadi menjadi 2 bagian:

- Lingkungan Server Pengembangan (*Development*), dengan spesifikasi sebagai berikut

Recommendation	
Processor	3 core 64 bit
Memory	8 GB
Storage	200 GB for System & Data

- Lingkungan Server *Production* (Live), dengan spesifikasi sebagai berikut

	Recommendation	Minimum
Processor	12 core 64 bit	6 core 64 bit
Memory	32 GB	16 GB
Storage	2000 GB for System & Data	1000 GB for System & Data