

**PENGEMBANGAN HALAMAN DETAIL PERFORMA DAN  
PROJECT SCOREBOARD (STUDI KASUS: *DASHBOARD  
TIMLOAD*)**



Disusun Oleh:

N a m a : Romzi Qutbi

NIM : 18523229

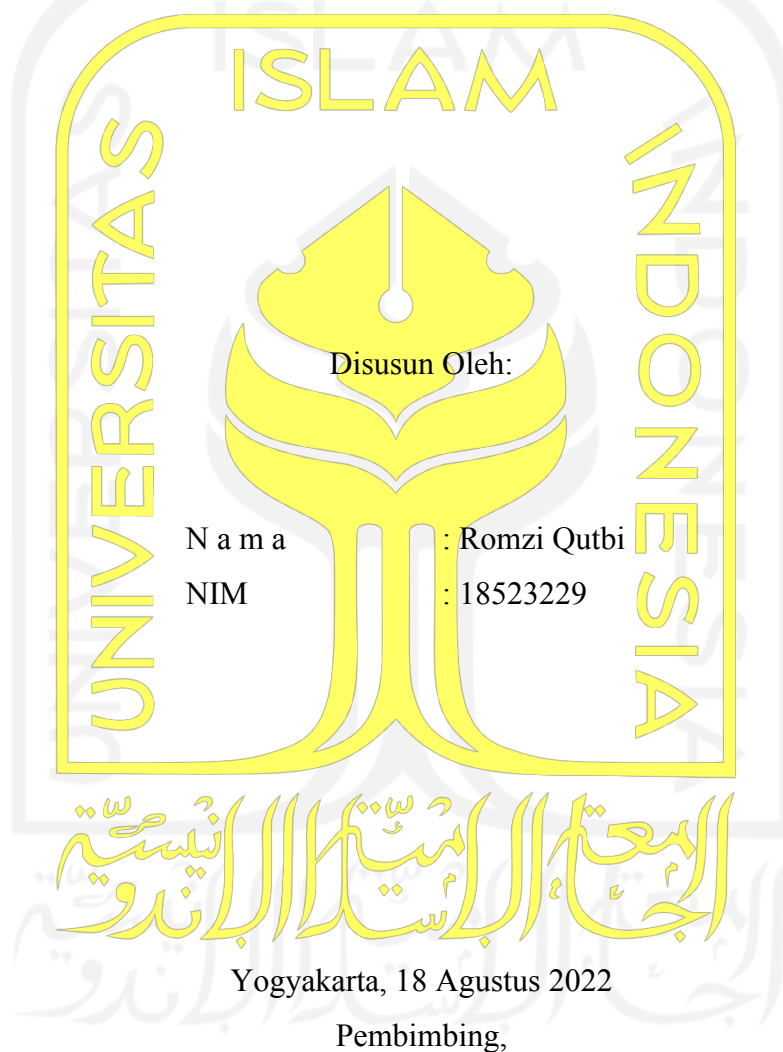
**PROGRAM STUDI INFORMATIKA – PROGRAM SARJANA  
FAKULTAS TEKNOLOGI INDUSTRI  
UNIVERSITAS ISLAM INDONESIA**

**2022**

HALAMAN PENGESAHAN DOSEN PEMBIMBING

PENGEMBANGAN HALAMAN DETAIL PERFORMA DAN  
PROJECT SCOREBOARD (STUDI KASUS: *DASHBOARD*  
TIMLOAD)

TUGAS AKHIR JALUR MAGANG



  
( Septia Rani, S.T., M.Cs. )

HALAMAN PENGESAHAN DOSEN PENGUJI

**PENGEMBANGAN HALAMAN DETAIL PERFORMA DAN  
PROJECT SCOREBOARD (STUDI KASUS: *DASHBOARD*  
TIMLOAD)**

**TUGAS AKHIR JALUR MAGANG**

Telah dipertahankan di depan sidang penguji sebagai salah satu syarat untuk memperoleh gelar Sarjana Komputer dari Program Studi Informatika – Program Sarjana di Fakultas Teknologi Industri Universitas Islam Indonesia

Yogyakarta, 18 Agustus 2022

Tim Penguji

Septia Rani, S.T., M.Cs.

**Anggota 1**

Sheila Nurul Huda, S.Kom., M.Cs.

**Anggota 2**

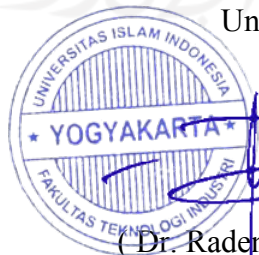
Chanifah Indah Ratnasari, S.Kom.,  
M.Kom.

Mengetahui,

Ketua Program Studi Informatika – Program Sarjana

Fakultas Teknologi Industri

Universitas Islam Indonesia



(Dr. Raden Teduh Dirgahayu, S.T., M.Sc.)

**HALAMAN PERNYATAAN KEASLIAN TUGAS AKHIR**

Yang bertanda tangan di bawah ini:

Nama : Romzi Qutbi

NIM : 18523229

Tugas akhir dengan judul:

**PENGEMBANGAN HALAMAN DETAIL PERFORMA DAN  
PROJECT SCOREBOARD (STUDI KASUS: *DASHBOARD  
TIMLOAD*)**

Menyatakan bahwa seluruh komponen dan isi dalam tugas akhir ini adalah hasil karya saya sendiri. Apabila di kemudian hari terbukti ada beberapa bagian dari karya ini adalah bukan hasil karya sendiri, tugas akhir yang diajukan sebagai hasil karya sendiri ini siap ditarik kembali dan siap menanggung risiko dan konsekuensi apapun.

Demikian surat pernyataan ini dibuat, semoga dapat dipergunakan sebagaimana mestinya.

Yogyakarta, 18 Agustus 2022



## HALAMAN PERSEMBAHAN

Bismillahirrahmanirrahim, Assalamu'alaikum Wr.Wb.

Puji syukur kehadirat Allah SWT yang maha pengasih lagi maha penyayang serta tidak lupa selawat dan salam senantiasa tercurahkan kepada junjungan kita Nabi Muhammad SAW. Laporan tugas akhir ini saya persembahkan kepada kedua orang tua dan kakak saya yang selalu mendo'akan, serta mendukung dalam hal apapun. Saya juga mempersembahkan laporan tugas akhir ini kepada sahabat dan teman-teman saya yang selalu memberi semangat dalam menyelesaikan laporan tugas akhir ini.

Wassalamu'alaikum Wr. Wb.



## HALAMAN MOTO

*“As long as you live, keep learning how to live.”*  
-Lucius Annaeus Seneca-



## KATA PENGANTAR

Assalamu'alaikum Wr. Wb.

Puji syukur atas kehadiran Allah SWT yang telah memberikan rahmat dan petunjuk, sehingga dapat menyelesaikan laporan tugas akhir ini yang berjudul “PENGEMBANGAN HALAMAN DETAIL PERFORMA DAN PROJECT SCOREBOARD (STUDI KASUS: *DASHBOARD* TIMLOAD)”. Laporan ini tidak akan selesai tanpa dukungan dan do'a dari orang tua dan bimbingan dari berbagai pihak. Dan tidak lupa penulis mengucapkan banyak terima kasih kepada.

1. Bapak Dr. Raden Teduh Dirgahayu, S.T., M.Sc, selaku Ketua Program Studi Informatika Universitas Islam Indonesia
2. Ibu Septia Rani, S.T., M.Cs., selaku dosen pembimbing selama menjalani program magang.
3. Lukas Wiku Kuswidiyanto dan Bayu Hendra Winata, selaku *supervisor* penulis dan *Leader* di *Internal Growth Team*.
4. Bapak Wisnu Manupraba, selaku CEO PT Javan Cipta Solusi.
5. Para dosen Program Studi Informatika Universitas Islam Indonesia yang memberikan ilmu kepada penulis di perkuliahan.
6. Rekan magang penulis, yang sama-sama berjuang bersama penulis dan saling membantu selama menjalani magang.
7. Teman-teman, di manapun berada baik di luar maupun di dalam lingkungan Program Studi Informatika Universitas Islam Indonesia.

Akhir kata penulis ucapkan terima kasih pada semua pihak yang telah berusaha membantu dalam penyusunan laporan tugas akhir ini, dan penulis berharap semoga laporan ini membantu menambah pengetahuan dan pengalaman bagi para pembaca.

Wassalamu'alaikum Wr. Wb.

Yogyakarta, 18 Agustus 2022



( Romzi Qutbi )

## SARI

PT Javan Cipta Solusi adalah perusahaan yang bergerak di bidang teknologi informasi. Dengan banyaknya data dari proyek yang dikerjakan, maka perlu ada sebuah *dashboard* yang dapat memberikan *insight* untuk membuat keputusan yang baik ke depannya pada perusahaan. Oleh karena itu, PT Javan Cipta Solusi menerapkan sebuah metode bernama *Balanced Scorecard* (Alat Ukur) untuk mengukur dan menilai kinerja dalam perusahaan. Timload adalah salah satu *tool* yang dikembangkan oleh tim internal di PT Javan Cipta Solusi untuk mencapai tujuan tersebut. Timload adalah aplikasi berbasis *web* yang menggunakan *framework* Django dengan mengambil data dari *database* Taiga dan ActiveCollab untuk diproses dan kemudian disajikan dalam bentuk *dashboard* yang akan mempermudah para *stakeholder* di perusahaan untuk membuat keputusan. Tujuan dari laporan ini adalah memaparkan bagaimana *dashboard* ini dikembangkan, khususnya terkait pengembangan dua fitur yaitu halaman Detail Performa dan Project Scoreboard.

Kata kunci: *Balanced Scorecard*, Timload.



## GLOSARIUM

<i>Balanced Scorecard</i>	Suatu metode pengukuran dan penilaian kinerja suatu perusahaan atau organisasi.
<i>Model-View-Template</i>	salah satu <i>design pattern</i> dalam pengembangan aplikasi <i>web</i> , khususnya digunakan pada <i>web framework</i> Django.
<i>Agile</i>	Metode pengembangan aplikasi secara berulang dan berfokus pada pengguna.
<i>Full-stack</i>	Pengembangan aplikasi pada sisi <i>back-end</i> sekaligus <i>front-end</i> .



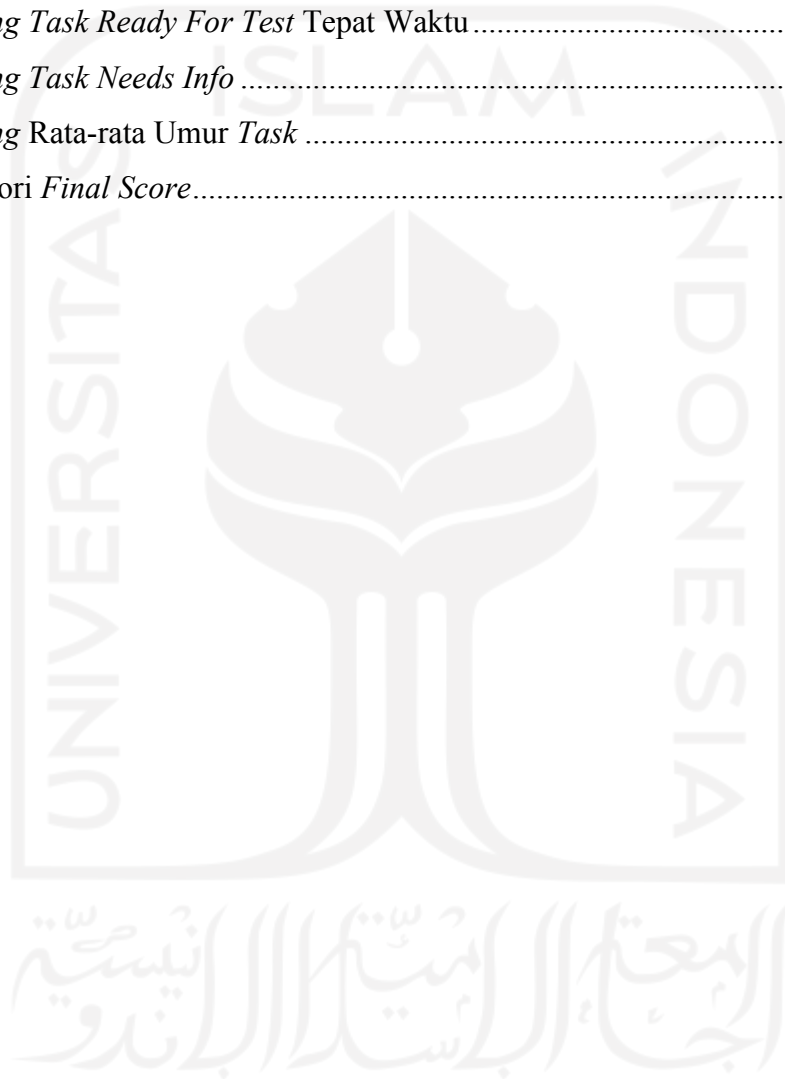
## DAFTAR ISI

HALAMAN JUDUL .....	i
HALAMAN PENGESAHAN DOSEN PEMBIMBING .....	ii
HALAMAN PENGESAHAN DOSEN PENGUJI .....	iii
HALAMAN PERNYATAAN KEASLIAN TUGAS AKHIR .....	iv
HALAMAN PERSEMBAHAN .....	v
HALAMAN MOTO .....	vi
KATA PENGANTAR .....	vii
SARI .....	viii
GLOSARIUM .....	ix
DAFTAR ISI .....	x
DAFTAR TABEL .....	xii
DAFTAR GAMBAR .....	xiii
<b>BAB I PENDAHULUAN</b> .....	<b>1</b>
1.1 Latar Belakang .....	1
1.2 Ruang Lingkup .....	2
1.3 Tujuan .....	2
1.4 Manfaat .....	2
1.5 Sistematika Penulisan .....	3
<b>BAB II LANDASAN TEORI</b> .....	<b>4</b>
2.1 <i>Balanced Scorecard</i> .....	4
2.2 <i>Mode-View-Template</i> .....	5
2.3 <i>Support Vector Machine</i> .....	6
<b>BAB III PELAKSANAAN MAGANG</b> .....	<b>8</b>
3.1 Manajemen Proyek .....	8
3.1.1 Tahap <i>Analysis</i> .....	9
3.1.2 Tahap <i>Design</i> .....	9
3.1.3 Tahap <i>Implementation</i> .....	9
3.1.4 Tahap <i>Evaluation</i> .....	9
3.1.5 Tahap <i>Deployment</i> .....	9
3.2 Aktivitas Magang .....	9
3.2.1 Masa Orientasi .....	10
3.2.2 Pengembangan <i>Dashboard Profiling Engineer</i> .....	13
3.2.3 Pengembangan Halaman Detail Performa .....	20
3.2.4 Pengembangan Project Scoreboard .....	25
3.3 Hasil dan Pembahasan Pelaksanaan Magang .....	34
3.3.1 <i>Dashboard Profiling Engineer</i> .....	34
3.3.2 Halaman Detail Performa .....	37
3.3.3 Project Scoreboard .....	41
<b>BAB IV REFLEKSI PELAKSANAAN MAGANG</b> .....	<b>45</b>
4.1 Relevansi Akademik .....	45
4.2 Pembelajaran Magang .....	45
4.2.1 Manfaat Magang .....	45
4.2.2 Kendala, Hambatan, dan Tantangan .....	46
<b>BAB V PENUTUP</b> .....	<b>48</b>
5.1 Kesimpulan .....	48
5.2 Saran .....	48
DAFTAR PUSTAKA .....	49



## DAFTAR TABEL

Tabel 3.1 Aktivitas Magang.....	10
Tabel 3.2 <i>Requirements</i> Aplikasi Rekrutmen <i>e-Learning</i> .....	12
Tabel 3.3 Daftar Label <i>Task</i> .....	16
Tabel 3.4 Sampel Data <i>Labelled</i> .....	19
Tabel 3.5 <i>Scoring Task</i> Tepat Waktu.....	28
Tabel 3.6 <i>Scoring Task Ready For Test</i> Tepat Waktu .....	28
Tabel 3.7 <i>Scoring Task Needs Info</i> .....	28
Tabel 3.8 <i>Scoring Rata-rata Umur Task</i> .....	29
Tabel 3.9 Kategori <i>Final Score</i> .....	29



## DAFTAR GAMBAR

Gambar 2.1 <i>Model-View-Template</i> .....	6
Gambar 3.1 Metode Pengembangan <i>Agile</i> .....	8
Gambar 3.2 <i>e-Learning</i> Javan .....	10
Gambar 3.3 Coderbyte .....	13
Gambar 3.4 <i>User Story Profiling Engineer</i> Berdasarkan <i>Assignment Task</i> di Taiga .....	14
Gambar 3.5 <i>Data Source</i> DataGrip .....	15
Gambar 3.6 Konfigurasi <i>Database</i> DataGrip .....	15
Gambar 3.7 <i>Query Task</i> Taiga .....	16
Gambar 3.8 Desain Grafik <i>Feedback</i> .....	18
Gambar 3.9 <i>User Story</i> Membuat Halaman Detail Performa .....	20
Gambar 3.10 <i>Query Task</i> Perorang .....	21
Gambar 3.11 Indikator Warna .....	22
Gambar 3.12 <i>Hover Subject Task</i> .....	22
Gambar 3.13 Tombol <i>Back</i> .....	22
Gambar 3.14 <i>Hover</i> Nama Tim .....	22
Gambar 3.15 <i>Filter Role</i> .....	23
Gambar 3.16 <i>Filter</i> Bulan .....	23
Gambar 3.17 <i>Filter</i> Tahun .....	24
Gambar 3.18 <i>JQuery Auto-Refresh</i> .....	24
Gambar 3.19 Keterangan Indikator Warna <i>Active Task</i> .....	25
Gambar 3.20 Keterangan Indikator Warna <i>Performance Analysis</i> .....	25
Gambar 3.21 <i>User Story</i> Membuat Project Scoreboard .....	26
Gambar 3.22 <i>Query</i> Project Scoreboard Untuk <i>List Project</i> .....	27
Gambar 3.23 Card Project .....	29
Gambar 3.24 <i>Filter</i> Bulan dan Tahun Project Scoreboard .....	30
Gambar 3.25 <i>Gauge Chart</i> Project Scoreboard .....	30
Gambar 3.26 <i>Query</i> Detail Performance Project Scoreboard .....	31
Gambar 3.27 <i>Line Chart</i> Detail Peforma .....	32
Gambar 3.28 <i>Gauge Chart</i> Detail Performa .....	32
Gambar 3.29 Tombol <i>Back</i> Detail Performa Project Scoreboard .....	33
Gambar 3.30 <i>Hover Line Chart</i> .....	33
Gambar 3.31 <i>Summray</i> Detail Performa Project Scoreboard .....	33

Gambar 3.32 Tampilan <i>Trend Project Scoreboard</i> .....	34
Gambar 3.33 Halaman <i>Task</i> .....	35
Gambar 3.34 Halaman <i>Engineer</i> .....	36
Gambar 3.35 Halaman <i>Detail Engineer</i> .....	37
Gambar 3.36 <i>Task Progress</i> .....	39
Gambar 3.37 <i>Active Task</i> .....	40
Gambar 3.38 <i>Performance Analysis</i> .....	40
Gambar 3.39 <i>Finished Task</i> .....	41
Gambar 3.40 Halaman <i>Index Project Scoreboard</i> .....	44
Gambar 3.41 Halaman <i>Detail Performa Project Scoreboard</i> .....	44



## BAB I PENDAHULUAN

### 1.1 Latar Belakang

Dewasa ini perkembangan teknologi informasi dan komunikasi sangatlah pesat. Dengan hal ini banyak sekali kehidupan manusia yang dipermudah. Banyak sekali perusahaan yang berusaha untuk bertahan di industri ini. Perusahaan-perusahaan dituntut untuk selalu berinovasi sehingga tujuan-tujuannya tercapai. Banyak sekali aspek yang harus diperhatikan mulai dari aspek internal (seperti visi misi perusahaan dan sumber daya manusia) hingga aspek eksternal (seperti perilaku konsumen dan *trend* pasar).

Perusahaan memerlukan sebuah sistem untuk mengelola setiap strateginya sebaik mungkin yang mana aspek-aspek yang penting dapat saling terintegrasi. Dari kebutuhan tersebut, perusahaan menerapkan sebuah metode pengukur kinerja bernama *Balanced Scorecard*. *Balanced Scorecard* adalah sistem manajemen yang bertujuan untuk menerjemahkan tujuan strategis perusahaan ke dalam serangkaian tujuan kinerja yang gilirannya, diukur, dipantau, dan diubah jika perlu untuk memastikan bahwa tujuan strategis perusahaan tercapai.

*Dashboard* Timload adalah bentuk dari penerapan *Balanced Scorecard* di PT Javan Cipta Solusi. *Dashboard* ini dibuat karena ditemukan beberapa masalah di proyek seperti ada tim yang kadang tidak mendapat *task*, ada tim yang mendapatkan terlalu banyak *task*, dan kedua tim tersebut diberi *reward* yang sama tanpa memperdulikan performa masing-masing, dan juga ini menyebabkan *deliver* produk ke *client* menjadi terlambat, dan jika ini terjadi maka pengembangan produk menjadi lama dan terjadi pembengkakan biaya yang tidak baik untuk perusahaan. Dengan adanya *dashboard* ini maka setiap tim akan termonitor dengan baik sehingga nantinya tim dengan performa yang bagus akan mendapatkan *reward* berupa bonus. Dan produk di-*deliver* dengan tepat waktu.

PT Javan Cipta Solusi adalah perusahaan teknologi informasi yang berdiri sejak 2008 di Bandung (PT Javan Cipta Solusi, 2021). Perusahaan ini telah mengembangkan produk dari berbagai klien dengan berbagai instansi seperti swasta, pemerintah kabupaten maupun kota, kementerian dan lembaga pusat hingga Perguruan Tinggi Negeri.

PT Javan Cipta Solusi mengembangkan sistem *Balancing Scorecard*-nya sendiri yang bernama Timload. Timload adalah *dashboard* alat ukur untuk untuk memantau performa tim hingga performa dari proyek yang sedang dikerjakan oleh PT Javan Cipta Solusi. Timload

dibangun menggunakan *framework* django. Django adalah *framework full-stack* untuk membangun *web* dengan menggunakan bahasa pemrograman python. Dengan kata lain django meliputi sisi *front-end* dan juga *back-end*. *Front-end* untuk sisi depan yang akan dilihat oleh pengguna, sedangkan *back-end* untuk sisi belakang yang berhubungan dengan basis data dan logika bisnis (Muhardian, 2016).

Pengembangan Timload di bawah *Internal Growth Team*, khususnya pada *Productivity Support Team*. Seperti namanya *Productivity Support Team* bertanggung jawab untuk membantu produktivitas di PT Javan Cipta Solusi dengan cara mengembangkan *tool* dan juga menjawab permasalahan teknis tim proyek.

Secara garis besar penulis terlibat dalam pengembangan dua fitur di Timload yaitu halaman detail performa dan Project Scoreboard. Halaman detail performa dikerjakan selama 1 bulan sedangkan Project Scoreboard dikerjakan selama 2.5 bulan.

## 1.2 Ruang Lingkup

Dalam menjalani proses magang, penulis masuk ke dalam Divisi *Internal Growth* dan sebagai *Full-stack Engineer* di *Productivity Support Team*. *Internal Growth* bertugas untuk memastikan fasilitas, *tools*, *knowledge* untuk tim terpenuhi sehingga mampu bekerja dengan efektif dan efisien. Proyek-proyek yang dikerjakan penulis selama menjalani magang yaitu:

- a. *Profiling engineer* berdasarkan *task* di Taiga.
- b. Mengerjakan halaman detail performa pada Timload.
- c. Mengerjakan fitur project scoreboard pada Timload.

## 1.3 Tujuan

Berdasarkan penjelasan latar belakang di atas, maka tujuan dari laporan ini adalah:

- a. Mendokumentasikan bagaimana halaman Detail Performa di Timload dikembangkan.
- b. Mendokumentasikan bagaimana Project Scoreboard di Timload dikembangkan.
- c. Bentuk pertanggungjawaban dari program magang di PT Javan Cipta Solusi sebagai bagian dari proses perkuliahan dan penjaluran di program studi informatika, Fakultas Teknologi Industri, Universitas Islam Indonesia.

## 1.4 Manfaat

Berdasarkan penjelasan tujuan di atas, laporan ini diharapkan dapat digunakan sebagai:

- a. Dokumen untuk bagaimana penerapan *balanced scorecard* pada *dashboard* Timload.



- b. Dokumen untuk bagaimana pengembangan halaman detail performa dan project scoreboard dikembangkan pada *dashboard* timload.

### 1.5 Sistematika Penulisan

Penulisan laporan ini terbagi dalam 5 BAB, yaitu:

- a. BAB I Pendahuluan

Pada BAB ini berisi tentang latar belakang dari kenapa laporan tugas akhir ini ditulis masalah yang dihadapi, solusi yang coba diterapkan serta penjelasan mengenai ruang lingkup magang, tujuan, manfaat, sekaligus sistematika penulisan.

- b. BAB II Landasan Teori

BAB ini berisi tentang konsep, teori, dan definisi yang digunakan sebagai landasan laporan ini dan pada saat melaksanakan proses magang.

- c. BAB III Pelaksanaan Magang

BAB ini berisi berbagai aktivitas yang dilakukan selama menjalani proses magang dan penjelasan bagaimana pengembangan berbagai proyek yang dikerjakan.

- d. BAB IV Refleksi Pelaksanaan Magang

BAB ini berisi tentang pelajaran yang didapatkan selama menjalani proses magang. dimulai dari relevansi akademik yang membahas tentang bagaimana perbandingan antara teori yang didapatkan waktu perkuliahan dengan apa yang terjadi di lapangan selama menjalani proses magang. Selain itu juga pada BAB ini membahas tentang pembelajaran yang dapat diambil selama menjalani proses magang.

- e. BAB V Penutup

BAB ini berisi kesimpulan dari keseluruhan laporan yang ditulis dan saran untuk proyek yang dikerjakan setelah menajalani magang.

## BAB II LANDASAN TEORI

### 2.1 *Balanced Scorecard*

*Balanced Scorecard* adalah suatu metode pengukuran dan penilaian kinerja suatu perusahaan atau organisasi dengan mengukur empat perspektif yaitu: perspektif keuangan, perspektif pelanggan, perspektif proses bisnis internal, serta perspektif pembelajaran dan pertumbuhan. *Balanced Scorecard* dapat dianalogikan sebagai tombol dan indikator di kokpit pesawat. Untuk tugas kompleks menavigasi dan menerbangkan pesawat, pilot memerlukan informasi rinci tentang banyak aspek penerbangan. Mereka membutuhkan informasi tentang bahan bakar, kecepatan udara, ketinggian, arah, tujuan, dan indikator lain yang merangkum lingkungan saat ini dan yang diperkirakan. Ketergantungan pada satu instrumen bisa berakibat fatal. Demikian pula, kompleksitas pengelolaan organisasi saat ini menuntut manajer untuk dapat melihat kinerja di beberapa bidang secara bersamaan (Kaplan & Norton, 1992).

Selain *Balanced Scorecard*, terdapat juga beberapa metode yang lain untuk pengukuran kinerja, diantaranya:

- a. *Performance Pyramid System (PPS)*. PPS adalah sebuah sistem yang saling terkait dari variabel kinerja yang berbeda, yang dikontrol pada tingkat organisasi yang berbeda.
- b. *The Tableau De Bord (TdB)*. Tujuan dari metode ini adalah memberikan manajer uraian dan parameter kunci untuk mendukung pengambilan keputusan yang memiliki dua implikasi penting. pertama, TdB tidak dapat menjadi dokumen tunggal yang berlaku sama baik untuk seluruh perusahaan karena setiap sub-unit memiliki tanggung jawab dan objektif yang berbeda. Ini menyebabkan keharusan adanya TdB untuk setiap sub-unit. kedua, berbagai TdB yang digunakan dalam perusahaan tidak boleh terbatas pada indikator-indikator keuangan.
- c. *Productivity Measurement and Enhancement System (ProMES)*. ProMES didasarkan pada teori perilaku kerja. dalam teori ini, motivasi dipandang sebagai suatu proses alokasi sumber daya ke seluruh tindakan dan tugas, di mana sumber daya tersebut adalah waktu dan tenaga seseorang.
- d. *Activity-Based Costing (ABC)*. ABC adalah untuk menganalisis biaya tidak langsung dalam perusahaan dan untuk menemukan kegiatan yang menyebabkan biaya-biaya tersebut.

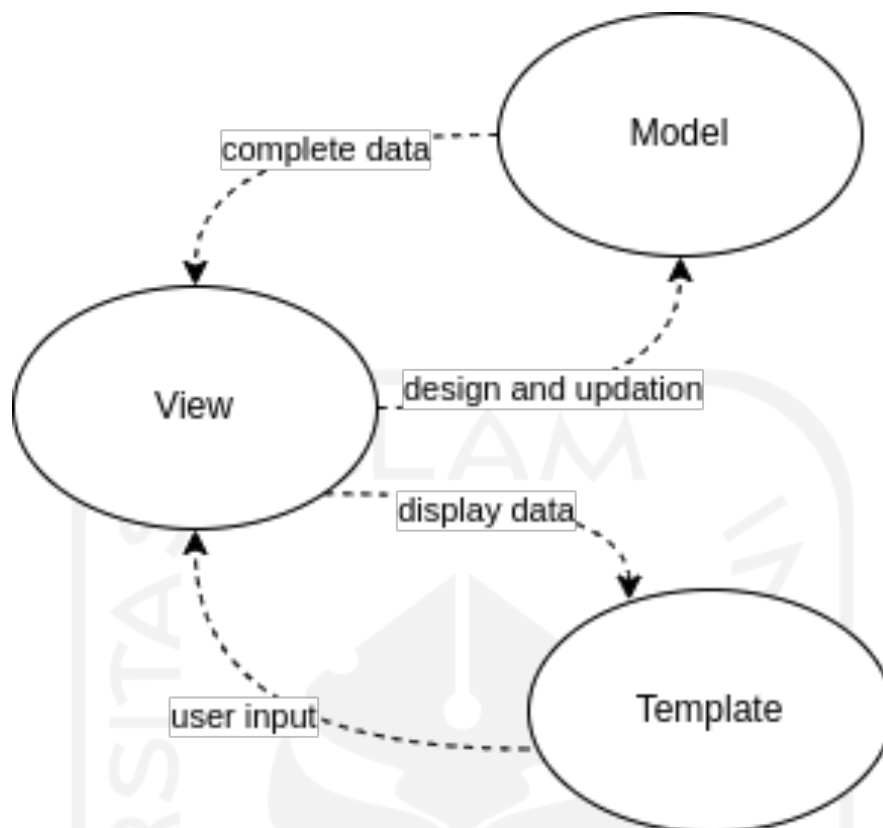
- e. *Sink and Tuttle*. metode pengukuran kinerja *Sink and Tuttle* adalah sebuah pendekatan klasik yang menyatakan bahwa kinerja suatu organisasi memiliki keterkaitan yang rumit antar tujuh kriteria kinerja. ketujuh kriteria kerja tersebut antara lain adalah Efektivitas, Efisiensi, Kualitas, Produktivitas, Kualitas Kehidupan Kerja, Inovasi, Profitabilitas/*Budgetability*.

*Balanced Scorecard* memiliki empat perspektif yang terdiri atas perspektif keuangan, perspektif proses bisnis internal, perspektif pelanggan, serta perspektif pembelajaran dan pertumbuhan. Masing-masing perspektif dapat dijelaskan sebagai berikut:

- a. Perspektif keuangan. Perusahaan harus mampu mengelola keuangan dengan baik agar keuangannya terus stabil. Misalnya, biaya operasional, biaya produksi, biaya bahan baku, biaya tenaga kerja, termasuk keuntungan dari aktivitas penjualan.
- b. Perspektif proses bisnis internal. Kemampuan dan keahlian yang dimiliki karyawan di dalam perusahaan akan menghasilkan proses bisnis internal yang bagus. Proses bisnis internal yang bagus akan mendatangkan pelanggan dan meraih keuntungan secara keuangan.
- c. Perspektif pelanggan. Membahas tentang tujuan kinerja terkait calon pelanggan dan pasar. Pelayanan yang bagus akan membuat konsumen loyal dan bisa saja mendatangkan konsumen baru.
- d. Perspektif pembelajaran dan pertumbuhan. Karyawan adalah elemen yang penting dalam perusahaan. Maka dari itu perusahaan dituntut untuk membangun sistem agar ada pengembangan pada Sumber Daya Manusia.

## **2.2 Mode-View-Template**

*Model-View-Template* adalah salah satu *design pattern* dalam pengembangan *web*, khususnya digunakan pada *web framework* Django. Pada Gambar 2.1 ditunjukkan bagaimana keterkaitan antara ketiganya.



Gambar 2.1 Model-View-Template

1. Model, berperan sebagai *interface* kepada *database*. Kelas sebagai tabel dalam *database* dan atribut kelas sebagai kolom dalam *database*.
2. View, berperan untuk mengeksekusi logika dan mengambil, membaca, atau menulis data dari *database* melalui model kemudian menampilkannya ke dalam *template*.
3. Template, berperan sebagai antarmuka yang di-*render* dan dilihat langsung oleh *user* yang mengunjungi *website* (Sachin, 2021).

### 2.3 Support Vector Machine

*Support Vector Machine* (SVM) adalah salah satu algoritma *supervised machine learning* yang mana data yang digunakan untuk *training* model adalah data yang sudah memiliki label sebelumnya. Sehingga dalam penentuan proses keputusan, mesin akan mengkategorikan data ke dalam label yang sesuai dengan karakteristik yang dimilikinya. Dalam melakukan klasifikasi, konsep SVM lebih matang dan lebih jelas secara matematis dibandingkan metode klasifikasi lainnya (samsudiney, 2019).

Dalam melakukan klasifikasi, SVM akan mencari *hyperline* terbaik dalam memaksimalkan jarak antar kelas berdasarkan N-dimensi tergantung banyak *features* sehingga

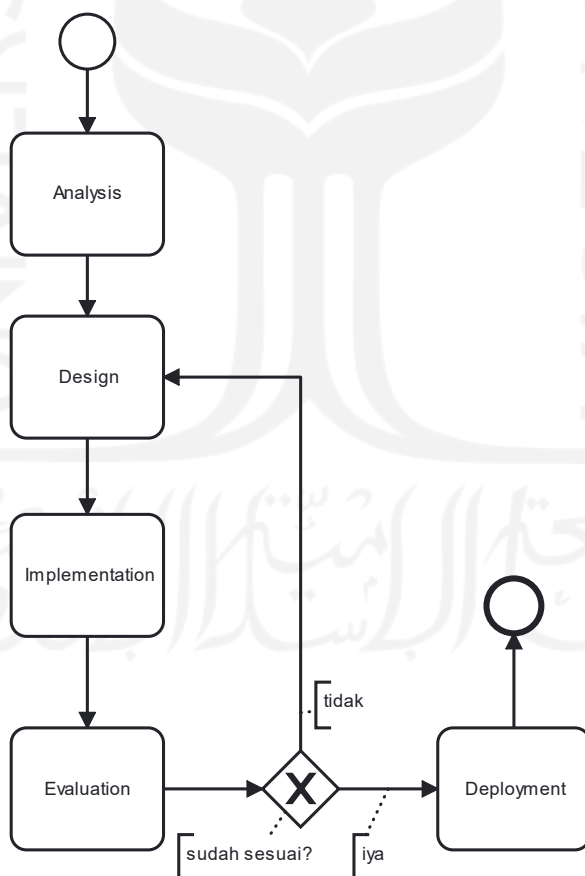
nantinya akan mudah menentukan kelas atau kategori data yang baru. Dalam hal ini penulis menggunakan SVM untuk mengklasifikan data teks ke dalam kelas-kelas tertentu (Delyani, 2021).



## BAB III PELAKSANAAN MAGANG

### 3.1 Manajemen Proyek

Dalam pengerjaan proyek *dashboard Timload* menggunakan metode *Agile Development*. Metode *Agile* dipilih karena dapat membantu *developer* mengembangkan perangkat lunak lebih efisien dan sesuai kebutuhan konsumen. Dilakukan secara bertahap dan berulang-ulang (Haekal, 2022). Alur tahap pengembangan perangkat lunak khususnya pada proyek ini dengan metode *agile* dapat dilihat pada Gambar 3.1. Terdapat 5 tahap, mulai dari tahap *analysis*, *design*, *implementation*, *evaluation*, dan *deployment*. Pada saat tahap pengembangan sudah mencapai tahap *evaluation*, maka nantinya terdapat dua pilihan yaitu jika *implementation* sudah sesuai maka akan berlanjut ke tahap *deployment*, dan jika tidak maka akan kembali ke tahap *design* lagi.



Gambar 3.1 Metode Pengembangan *Agile*

### 3.1.1 Tahap *Analysis*

Yang pertama yaitu tahap analisis. Pada tahap ini tim akan mengidentifikasi kebutuhan apa saja yang diperlukan dalam mencapai tujuan untuk menyelesaikan masalah yang ingin dipecahkan. Tahap analisis dapat berupa riset atau *meeting* dengan tim dari divisi lain di PT Javan Cipta Solusi dan mencatat apa masalahnya dan apa tujuannya.

### 3.1.2 Tahap *Design*

Yang kedua yaitu tahap desain. Pada tahap ini tim akan melakukan perancangan aplikasi mulai dari desain arsitektur, desain *User Interface*, hingga menentukan teknologi apa saja yang akan digunakan. *Output* dari tahap ini adalah berupa *Software Design Document* (SDD). SDD adalah rencana yang sudah didetailkan untuk mengembangkan perangkat lunak saja yang diperlukan (Lucid Content Team, 2022).

### 3.1.3 Tahap *Implementation*

Selanjutnya yaitu tahap implementasi. Di tahap ini tim *engineer* mulai mengimplementasikan semuanya sesuai dengan dokumen SDD. Pengerjaan meliputi *back-end* dan *front-end* dari perangkat lunak.

### 3.1.4 Tahap *Evaluation*

Tahap selanjutnya adalah tahap evaluasi, yang mana pada tahap ini fitur yang sudah dibuat oleh *engineer* akan diuji dan diberikan penilaian. Pengujian ini dilakukan pada *server staging*. *Staging server* adalah replika lingkungan produksi atau *production environment* untuk menguji kode baru dan pembaruan sebelum mempublikasikannya (Priharto, 2021). Jika lolos tahap ini maka akan berlanjut ke tahap *deployment*. Namun apabila tidak lolos maka akan kembali ke tahap *design* sesuai dengan *feedback* yang diberikan.

### 3.1.5 Tahap *Deployment*

Tahap yang terakhir adalah tahap *deployment*. Tahap *deployment* adalah tahap untuk melakukan publikasi aplikasi atau fitur yang sudah dikerjakan dan sudah melalui tahap *testing* sehingga dapat diakses dan digunakan oleh *user*. Tahap ini biasanya dikerjakan oleh *DevOps Engineer*.

## 3.2 Aktivitas Magang

Kegiatan magang dilakukan selama enam bulan, mulai dari 1 September 2021 sampai bulan 28 Februari 2022 sebagai *Full-Stack Engineer* di PT Javan Cipta Solusi. Secara garis besar terdapat 4 aktivitas yang terbagi dalam durasi waktu tertentu. Untuk lebih jelasnya aktivitas-aktivitas tersebut dapat dilihat pada Tabel 3.1. Adapun penjelasan secara rinci terkait dengan aktivitas-aktivitas tersebut pada subbab berikutnya.

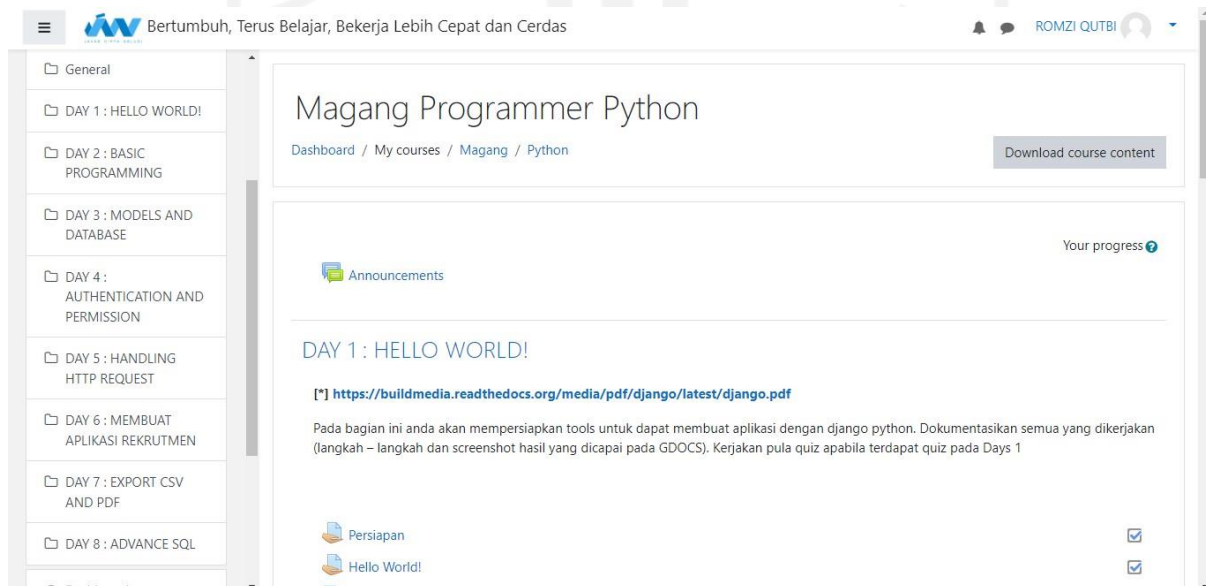
Tabel 3.1 Aktivitas Magang

No.	Aktivitas	Durasi
1.	Masa Orientasi	1 bulan
2.	Pengembangan <i>Dashboard</i> Profiling Engineer	1.5 bulan
3.	Pengembangan Halaman Detail Performa	1 bulan
4.	Pengembangan Project Scoreboard	2.5 bulan

### 3.2.1 Masa Orientasi

Masa orientasi dilalui selama maksimal satu bulan. Pada masa orientasi semua peserta magang diberikan tugas-tugas untuk dikerjakan sesuai dengan posisi setiap peserta magang. Diharapkan dengan adanya masa orientasi ini para peserta magang lebih siap waktu masuk ke proyek masing-masing. Karena penulis mengambil posisi *Python Developer*, maka tugas-tugas yang diberikan berhubungan dengan apa saja yang dilakukan *Python Developer* pada saat masuk ke proyek.

Pada masa orientasi setiap peserta magang akan di-*invite* ke *e-Learning* Javan. Setiap petunjuk pengerjaan, *tools* yang digunakan, hingga submit tugas yang diberikan selama masa orientasi sudah tertera pada *e-Learning* Javan. Pada Gambar 3.2 menunjukkan tampilan *e-Learning* Javan.

Gambar 3.2 *e-Learning* Javan

Tugas-tugas yang diberikan dalam *e-Learning* yaitu:



- a. Meng-*install tools* yang dipakai seperti Pycharm Professional, *setup database* (PostgreSQL), Django, Sonarlint, DataGrip, Gitkraken, dan membuat akun Gitlab Javan. Pada tugas ini penulis mengerjakan proyek yang menampilkan “Hello World” dengan cara membuat *branch* baru di Gitlab dan mengerjakannya di sana, setelah selesai mengerjakan proyeknya setelah itu membuat *merge request*, kemudian di-*merge*.
- b. *Basic Programming*. Pada bagian *Basic Programming* terdapat 3 tugas yaitu cetak ganjil genap, cetak huruf vokal, dan kalkulator sederhana. Pada tugas cetak ganjil genap instruksinya adalah program menerima dua input berupa angka kemudian mencetak angka dari kedua input tersebut sekaligus apakah dia ganjil atau genap. Tugas kedua adalah cetak huruf vokal, pada tugas ini memiliki instruksi berupa program menerima *input* berupa *string* kemudian program mencetak huruf vokal yang unik pada String tersebut. Kemudian tugas yang terakhir yaitu kalkulator sederhana, instuksinya adalah program menerima *input string* berupa operasi arimatika sederhana seperti “2+1”, “3x2”, dan lain-lain, kemudian dari *string* tersebut dihitung hasilnya oleh program.
- c. *Models and Database*. Pada bagian ini tugasnya berupa membuat *database*, kemudian membuat tabel-tabelnya sekaligus. Tabel-tabelnya sudah ditentukan pada *e-Learning* jadi tinggal bagaimana men-*setup*-nya, membuat modelnya di *database*, sekaligus membuat relasinya antara tabel yang satu dengan yang lain.
- d. *Authentication and Permission*. Pada bagian ini tugasnya terbagi atas 3 yaitu *form and validation*, *authentication*, dan *permission*. Pada tugas *form and validation* instruksinya berupa membuat sebuah *form register* yang memiliki *field* nama, *e-mail*, nomor telepon, alamat, *password*, dan pilihan pemilik atau peminjam buku, sekaligus dengan validasinya baik pada *front-end* maupun *back-end*, kemudian data *register* tadi dimasukkan ke dalam basis data dengan tabel *user*. Pada tugas *Authentication* terdapat instruksi untuk membuat sebuah *form login* yang menggunakan *e-mail* dan *password* yang sudah di-*register* tadi pada halaman *register*, setelah *login* maka akan masuk ke halaman ”Hello World”. Tugas yang ketiga dari bagian ini yaitu *Permission*, instruksinya adalah mengubah halaman yang ditampilkan tadi sewaktu *login* dari “Hello World” menjadi halaman yang lain sesuai dengan *role*-nya, pemilik buku hanya dapat masuk halaman yang memiliki *menu* “Home”, “Buku Saya”, “Peminjaman Saya”, sedangkan untuk *role* peminjam hanya memiliki *menu* “Home” dan “Pinjaman Saya”.
- e. *Handling HTTP Request*. Pada bagian ini terdapat 4 tugas yaitu fitur *Create*, fitur *Read*, fitur *Update*, fitur *Delete*. Pada tugas fitur *Create*, terdapat instruksi untuk membuat

halaman di mana pada halaman tersebut *role* pemilik dapat menambahkan buku ke dalam basis data. Tugas kedua yaitu fitur *Read*, setelah berhasil mengerjakan tugas yang pertama dan data yang dimasukkan sudah tersimpan ke *database* maka selanjutnya adalah membuat halaman yang berfungsi untuk mengambil data dari basis data dan menampilkannya pada halaman tersebut. Tugas yang ketiga adalah membuat fitur *update* di mana data yang sudah tersimpan tadi di basis data dapat di-*update* atau di-*edit*. Tugas yang terakhir yaitu membuat fitur *delete*, instruksinya adalah bagaimana caranya agar data di basis data dapat dihapus oleh pengguna sesuai keinginannya.

- f. Membuat aplikasi rekrutmen. Dari semua tugas yang diberikan bagian yang paling sulit yaitu membuat sebuah aplikasi *web* dengan menggunakan *framework* Django yang memiliki *requirements* pada Tabel 3.2.

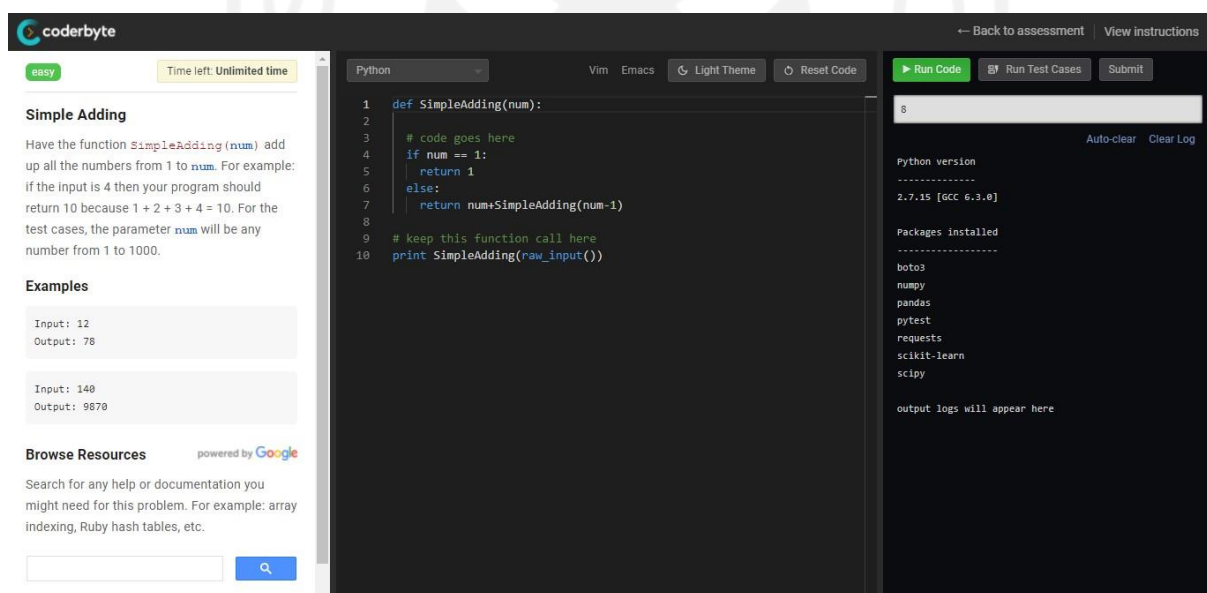
Tabel 3.2 *Requirements* Aplikasi Rekrutmen *e-Learning*

<b>Role</b>	<b>Requirement</b>	<b>Deskripsi</b>
HCD	Mengelola lowongan pekerjaan pada sistem rekrutmen	<ul style="list-style-type: none"> <li>• Tambah lowongan</li> <li>• <i>Edit</i> lowongan</li> <li>• Hapus lowongan</li> </ul>
	Review lamaran	<i>Screening</i> isi form data pelamar, CV, dan Portofolio (jika ada)
	Menentukan tes untuk pelamar	<ul style="list-style-type: none"> <li>• Tes psikologi (MBTI, temu bakat, psikologi)</li> <li>• Tes teknis</li> <li>• Coderbyte</li> </ul>
	Memberikan notifikasi terkait lamaran	<ul style="list-style-type: none"> <li>• Notifikasi lulus</li> <li>• Notifikasi tidak lulus</li> <li>• Notifikasi dapat dikirim melalui <i>e-mail</i> atau WhatsApp</li> </ul>
Pelamar	Membuat lamaran	<ul style="list-style-type: none"> <li>• Mengisi <i>form</i> lamaran</li> <li>• Mengunggah <i>file</i> yang dibutuhkan untuk lamaran</li> </ul>
	Mendapatkan notifikasi terkait lamaran	<ol style="list-style-type: none"> <li>a. Notifikasi lulus</li> <li>b. Notifikasi tidak lulus</li> </ol>
	Mengerjakan tes	<ul style="list-style-type: none"> <li>• Apabila mendapat notifikasi lulus maka pelamar dapat mengerjakan soal tes yang diberikan</li> </ul>
Korektor	Memeriksa hasil tes pelamar	Memeriksa hasil tes yang diunggah pelamar

	Memberikan rekomendasi (opsional) hasil tes	Berdasarkan hasil tes pelamar
--	---	-------------------------------

- g. *Export CSV and PDF*. Pada bagian ini terdapat 2 tugas yaitu membuat fitur *export csv* pada halaman pelamar dengan data-data siapa saja pelamar yang melamar, yang kedua yaitu membuat fitur *export PDF* pada halaman data pelamar atau hasil tes pelamar.
- h. *Advance SQL*. pada bagian terdapat instruksi untuk membuat query ke basis data agar didapatkan data-data yang diinginkan sesuai instruksi dengan menggunakan fungsi SQL seperti, JOIN, ORDER BY, WHERE, GROUP BY, DISTINCT, COUNT, dan lain-lain.

Selain mengerjakan *e-Learning* para peserta magang juga diberikan tugas lain berupa mengerjakan Coderbyte. Setiap posisi (*Engineer, Frontend, Analis, dan Quality Assurance*) mengerjakan level 1 sampai level 3, semakin tinggi levelnya maka kesulitannya juga semakin tinggi juga. Pada Gambar 3.3 menunjukkan salah satu soal Coderbyte yang dikerjakan oleh penulis pada masa orientasi.



Gambar 3.3 Coderbyte

### 3.2.2 Pengembangan *Dashboard Profiling Engineer*

Setelah masa orientasi, proyek pertama yang dikerjakan adalah Pengembangan *Dashboard Profiling Engineer* berdasarkan *task* di taiga. Dibuatnya proyek ini karena banyak data di *database* taiga Javan yang perlu diolah agar menghasilkan *insight* untuk membuat

keputusan kedepannya di perusahaan pada Gambar 3.4 adalah *user story* di Taiga yang dikerjakan oleh penulis.

The screenshot shows a Taiga user story interface. The title is "#180 Profiling Engineer Berdasar Assignment Task di Taiga". The story is in progress and was created by Bayu Hendra Winata on 15 Sep 2021 at 10:39. The description is "Membuat profil kerja karyawan berdasarkan task di Taiga". The requirements are:

1. Pengelompokan task berdasarkan data subject task
2. Performa sebuah task ("excellent", "good", "okay", "not good") berdasarkan waktu pengerjaan
3. Performa karyawan berdasarkan performa task (Berapa jumlah task yang masuk kategori "excellent", "good", "okay" atau "not good")

The flowchart illustrates the process: "data task di Taiga" and "Lama waktu pengerjaan task" are inputs to "Task subject". "Task subject" is then processed through "Klasifikasi task" to determine "Jenis task" and "Lama waktu pengerjaan task". The "Jenis task" is further classified into "Excellent", "Good", "Okay", and "Not Good" based on "Klasifikasi performa". The "Lama waktu pengerjaan task" is also used for "Klasifikasi performa".

The tasks list includes:

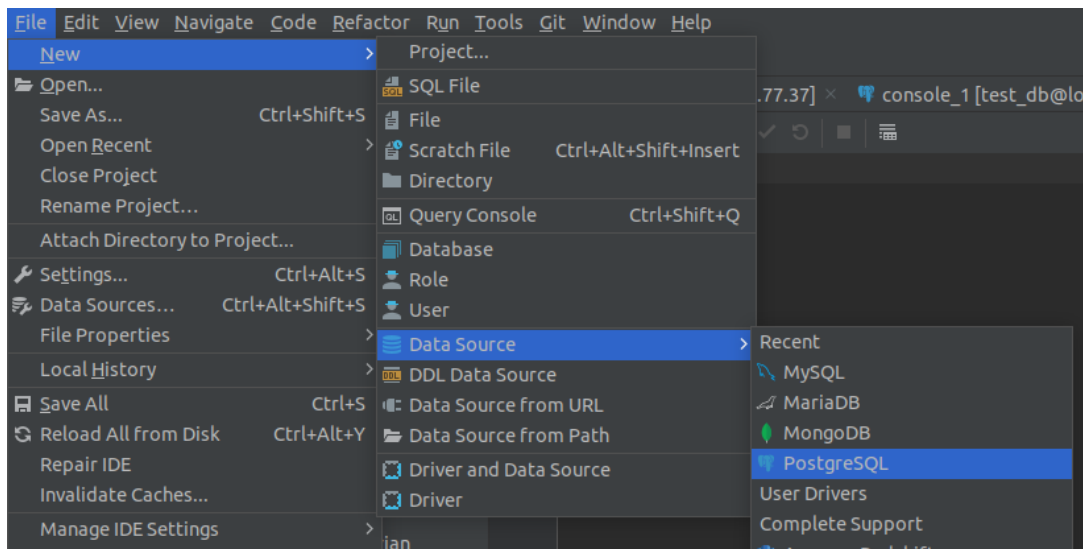
- #181 Riset data task di Taiga (Ready for test)
- #182 Query data task dari Taiga (Ready for test)
- #183 Labeling kategori task berdasarkan subject task (Ready for test)
- #185 Training dan uji akurasi model (Ready for test)
- #184 Riset metode klasifikasi (Ready for test)
- #187 Penyajian hasil (In progress)
- #213 Deploy aplikasi (New)

Gambar 3.4 *User Story Profiling Engineer Berdasarkan Assignment Task* di Taiga

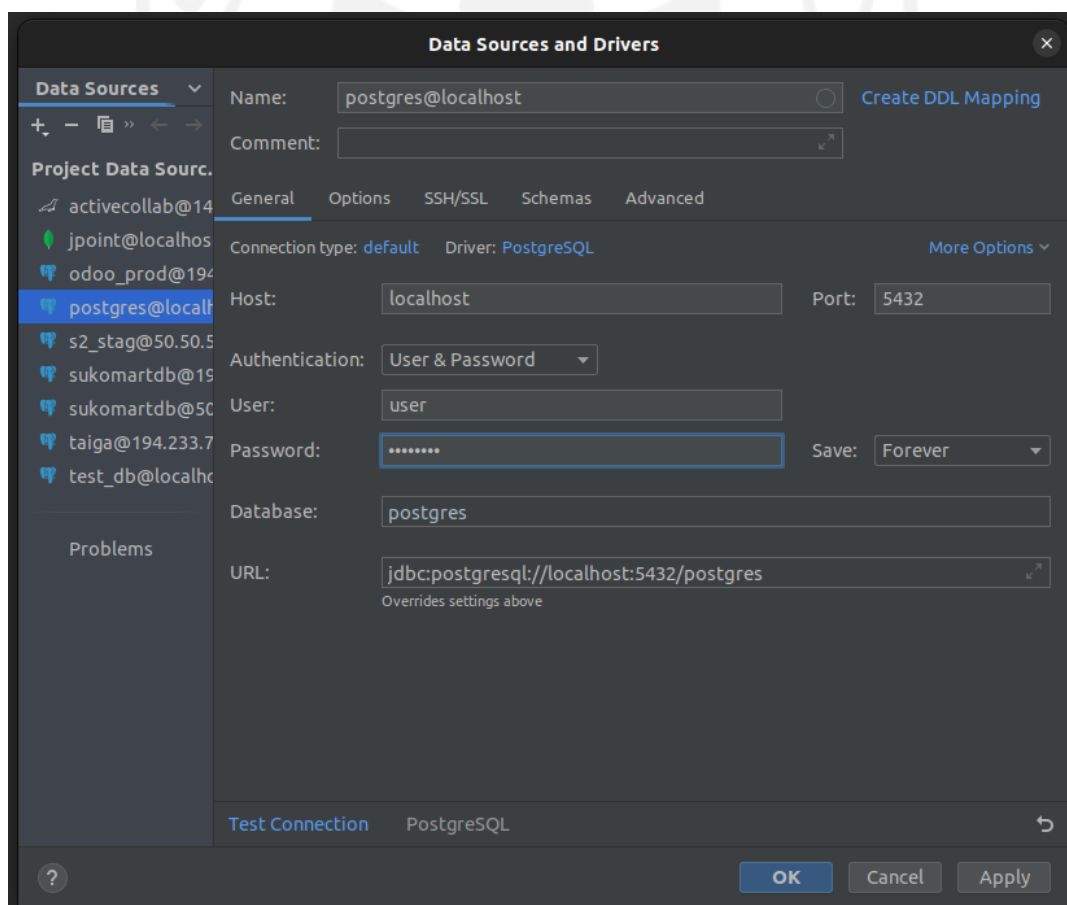
Dari *user story* tersebut terdapat 7 task yaitu:

- Riset Data *Task* Taiga, yaitu mengeksplorasi data *task* di basis data Taiga dan mencari tahu data apa saja yang dibutuhkan nantinya. Dalam melakukan riset ini, penulis menggunakan *tool* yang bernama DataGrip. DataGrip sendiri merupakan *Integrated Development Kit* (IDE) untuk *database* khususnya untuk *SQL Developer*. Untuk membuat data source baru

pilih ke File > New > Data Source > PostgreSQL seperti pada Gambar 3.5. Kemudian isi konfigurasi sesuai dengan *database* yang diinginkan seperti pada Gambar 3.6.



Gambar 3.5 *Data Source* DataGrip



Gambar 3.6 Konfigurasi *Database* DataGrip

- b. Query data *task* taiga, yaitu mengambil data *task* di taiga berdasarkan hasil riset. Keluaran dari *task* ini adalah diperolehnya data-data yang dibutuhkan dan disimpan dalam format file csv. Pada Gambar 3.7 menunjukkan query untuk mengambil data yang dibutuhkan.

```

select
  tasks_task.subject,
  tasks_task.created_date,
  tasks_task.due_date,
  tasks_task.finished_date
from tasks_task
join userstories_userstory uu on tasks_task.user_story_id = uu.id
order by tasks_task.created_date desc
limit 3000

```

Gambar 3.7 Query *Task* Taiga

- c. *Labelling* kategori *task* berdasarkan *subject task*, yaitu melabeli data yang yang didapatkan pada waktu *query* ke dalam kategori tertentu. Dalam melabeli data dibagi ke dalam dua macam yaitu berdasarkan *subject task* dan berdasarkan lama waktu pengerjaan. Berdasarkan lama waktu pengerjaan dapat dilakukan menggunakan pemrograman sedangkan berdasarkan *subject task* dilakukan secara manual. Pada Tabel 3.3 menunjukkan daftar label yang ditentukan penulis

Tabel 3.3 Daftar Label *Task*

Berdasarkan	Label	Deskripsi
<i>Subject task</i>	<i>Backend</i>	<i>Task</i> yang bersifat bagaimana data diproses pada server serta pembuatan API
	<i>Frontend</i>	<i>Task</i> yang bersifat merancang situs aplikasi dan pengintegrasian API
	<i>Devops</i>	<i>Task</i> yang bersifat untuk peluncuran produk atau aplikasi.
	Nonteknis	<i>Task</i> yang bersifat administratif
	Riset	<i>Task</i> yang bersifat untuk mencari tahu atau meriset
	<i>Query</i>	<i>Task</i> yang bersifat untuk mengambil data dari basis data.
	<i>Testing</i>	<i>Task</i> yang bersifat untuk <i>testing code</i> yang sudah direview.
	<i>Bugfixing</i>	<i>Task</i> yang bersifat <i>fixing bug</i> jika ditemukan.

	Fitur	<i>Task</i> yang bersifat untuk menambahkan fitur.
	<i>Adjustment</i>	<i>Task</i> yang bersifat untuk mengubah beberapa hal dari suatu fitur atau produk.
	Lainnya	<i>Task</i> yang tidak teridentifikasi
Durasi pengerjaan	<i>Excellent</i>	<i>Task closed</i> pada saat lebih dari satu hari sebelum <i>due date</i> .
	<i>Good</i>	<i>Task closed</i> pada saat satu hari sebelum <i>due date</i> atau tepat <i>due date</i> .
	<i>Okay</i>	<i>Task closed</i> pada saat <i>due date</i> sudah lewat sebelum atau tepat satu hari.
	<i>Not good</i>	<i>Task closed</i> pada sudah saat <i>due date</i> -nya lewat lebih dari satu hari

- d. Riset metode klasifikasi. Mencari tahu metode klasifikasi yang cocok untuk *dataset* yang sudah diberi label. Penulis memutuskan menggunakan metode klasifikasi *Support Vector Machine*. *Support Vector Machine* (SVM) adalah salah satu algoritma *supervised machine learning* yang mana data yang digunakan untuk *training* model adalah data yang sudah memiliki label sebelumnya. Sehingga dalam penentuan proses keputusan, mesin akan mengkategorikan data ke dalam label yang sesuai dengan karakteristik yang dimilikinya.
- e. *Training* dan uji akurasi model. Dari data *subject task* yang sudah dilabeli kemudian dilakukan pelatihan dan uji akurasi model. Dalam pengerjaan *task* ini, penulis menggunakan *tool* yaitu google colab. Sebelum melakukan *training*, data teks harus melalui proses *cleaning* dulu seperti menghilangkan *stopwords*, mengubah semua kata dalam bentuk huruf kecil, menghilangkan *link*, menghilangkan *tags* html, dan mengganti *symbol* dengan spasi. Kemudian dari data yang ada dibagi ke dalam dua kelompok yaitu 80% untuk *training* dan 20% untuk *testing*.

Akurasi yang didapatkan pada pertama kali mencoba kurang bagus yaitu sekitar 57%. Kemudian penulis mencoba meningkatkan akurasi modelnya dengan cara menggabungkan beberapa label *task* yang mirip ke dalam satu label. Seperti label kategori *frontend*, *backend*, dan fitur digabung menjadi *development*, label riset dan nonteknis digabung menjadi nonteknis, *bugfixing*, *testing*, dan *adjustment* digabung menjadi *bugfixing*. Setelah itu, modelnya dilatih lagi dan penulis mendapatkan akurasi sekitar 77%. Memang tidak

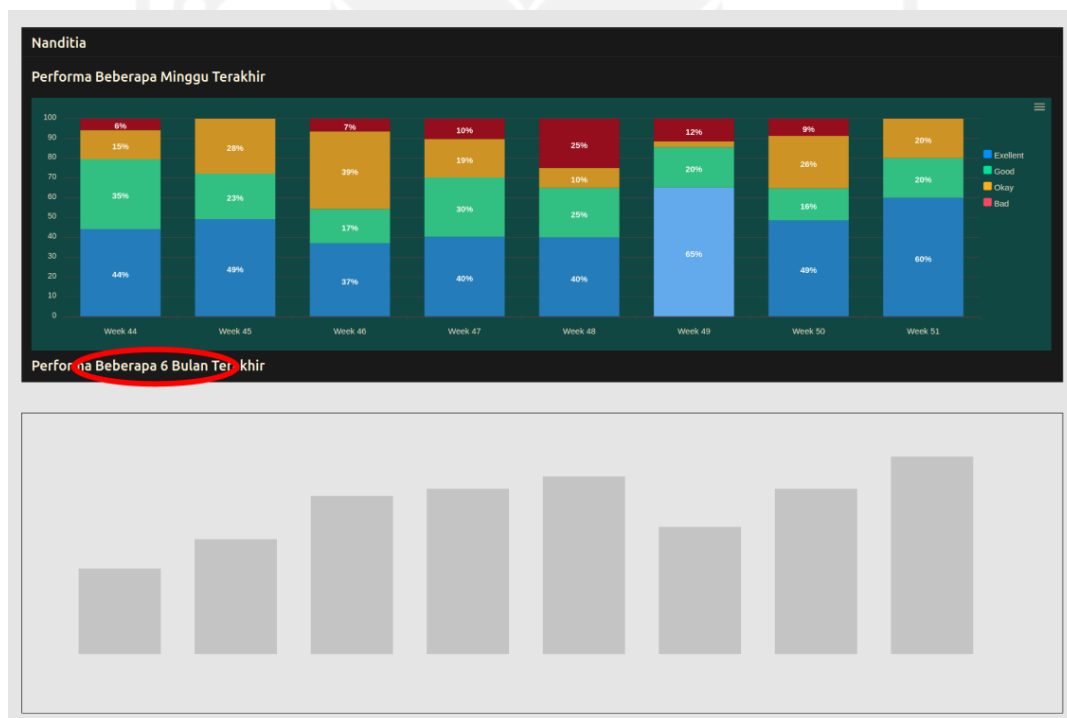


terlalu bagus, akan tetapi lebih baik dari model yang dilatih sebelumnya. Dari model yang sudah jadi kemudian di-*export* menggunakan *library* joblib.

- f. Penyajian hasil. Dari data model yang sudah dilatih dan data dari taiga, kemudian disajikan dalam bentuk aplikasi *web*. Untuk pembuatan aplikasi *web*, penulis menggunakan *framework python* bernama Django.

Setelah aplikasi *web*-nya sudah jadi, selanjutnya akan di-*review* oleh *supervisor*. Kemudian, dari *supervisor* memberikan *feedback* yaitu berupa *task* tambahan. Terdapat 7 task tambahan yaitu:

- Integrasi aplikasi dengan Timload Javan. Javan mempunyai beberapa *tools* internal untuk meningkatkan produktifitas para karyawannya. Aplikasi yang penulis buat akan diintegrasikan ke dalam aplikasi tersebut yaitu Timload tapi hanya satu fitur saja.
- Improvement* grafik performa beberapa minggu terakhir. pada Gambar 3.8 merupakan gambar desain grafik yang akan diubah. Bagian atas adalah grafik pada aplikasi yang penulis buat sedangkan pada bagian bawah adalah desain grafik yang diinginkan *supervisor*.



Gambar 3.8 Desain Grafik *Feedback*

- Integrasikan *custom field* dengan nama asli. pada aplikasi yang penulis buat masih menggunakan nama *custom field* (sejenis *nickname* jika di dalam proyek), nama di situ sebaiknya diubah kedalam bentuk nama lengkap masing-masing karyawan Javan.



- d. *Labelling* data kategori *development*. Karena pada model sebelumnya ada beberapa label yang penulis gabung untuk mendapatkan akurasi yang lebih bagus, akan tetapi dengan cara itu menyebabkan beberapa kategori sebenarnya dapat dipecah lagi menjadi beberapa kategori baru. Dalam mengerjakan *task* ini penulis memutuskan untuk mengambil data *subject task* dari *database* Taiga Javan melabeli ulang data tersebut dengan lebih hati-hati lagi. Sampel data yang sudah dilabeli seperti pada Tabel 3.4.

Tabel 3.4 Sampel Data *Labelled*

Subject Task	Kategori
Membuat Tampilan Home Geek Academy	frontend
Data tidak update ke bulk novation dev	bugfixing
[WBS] Perbaiki tab komunikasi dan diskusi disable di role spv telaah	adjustment
Analisa Indexing Tabel untuk Tuning Performance	nonteknis
[KOPI] Perbaiki redirect after delete Data Aduan Aroma	bugfixing
web - dashboard - refactor query	querying
Restore db `sppti_kpk` di server sepatu	devops
Restore db `sppti_djp2` dari drive	devops
Create database `sppti_kpk` di server sepatu	devops
Menambahkan How to Setup project FE di Readme	frontend
Persiapan Dokumen Nego	nonteknis

- e. *Training* dan uji akurasi untuk kategori *development*. Karena penulis melabeli ulang *dataset*-nya, maka lebih baik membuat melatih ulang modelnya secara keseluruhan. Dalam melakukan *training*, penulis memakai cara seperti sebelumnya pada waktu pertama kali membuat model, perbedaannya hanya pada *dataset*-nya yang dilabeli ulang. Dan pada saat melatih model dengan *dataset* yang sudah dilabeli ulang ini terdapat 1055 data tersebar ke beberapa label, yaitu nonteknis sebanyak 305, *frontend* sebanyak 217, *backend* sebanyak 156, *bugfixing* sebanyak 113, *adjustment* sebanyak 107, lainnya sebanyak 77, *querying* sebanyak 35, *devops* sebanyak 26, dan integrasi sebanyak 19. Kemudian dari 1055 data tersebut dibagi ke dalam 80% (844) data *training* dan 20% (211) data *testing*. penulis mendapat akurasi sebesar 83%.
- f. Tampilkan data menggunakan Highcharts.js. Karena dalam mengerjakan dalam proyek ini penulis menggunakan *library* ApexChart.js dan nantinya proyek ini akan diintegrasikan

dengan Timload (*tools internal Javan*) maka sebaiknya disamakan saja dengan *library* yang digunakan pada Timload sekaligus sesuaikan tema warnanya juga.

### 3.2.3 Pengembangan Halaman Detail Performa

Proyek kedua yang penulis kerjakan yaitu Halaman Halaman Detail Performa, yaitu berupa fitur untuk *tools* internal di PT Javan Cipta Solusi. Fitur ini berupa sebuah halaman yang menunjukkan detail performa setiap tim di Javan. Dibuatnya halaman ini adalah kelanjutan pengembangan *dashboard* Timload. Halaman ini sebenarnya sudah mulai dikerjakan dan ada beberapa hal sudah jadi akan tetapi masih ada beberapa kekurangan dan bug. Pada Gambar 3.9 merupakan *user story* beserta *task-task* yang dikerjakan.

The screenshot shows a Jira user story page for '#217 Membuat halaman Detail (per orang) Performa Tim Javan'. The page is titled 'USER STORY' and includes a description: 'Membuat halaman untuk menampilkan performa task per orang'. Below the description is a list of tasks, each with a status and an assignee. The tasks are:

Task ID	Description	Status	Assignee
#218	Membuat design halaman detail untuk performa per orang	Ready for test	Rizal Ilham
#219	Melakukan query task per orang	Ready for test	Rizal Ilham
#253	Design halaman detail untuk performa per orang (2)	Ready for test	Lukas Wiku
#220	Menampilkan data di halaman web	Ready for test	Rizal Ilham
#254	Menampilkan data di halaman web (2)	Ready for test	Rizal Ilham
#363	Perbaiki query task per orang	Ready for test	Romzi Q...
#364	Perbaiki routing halaman	Ready for test	Romzi Q...
#380	Optimasi tampilan halaman detail task per orang	Ready for test	Romzi Q...
#381	Menambah fitur untuk bisa filter setiap role	Ready for test	Romzi Q...
#384	Menambah fitur untuk bisa filter berdasarkan bulan	In progress	Romzi Q...
#385	Menambah fitur untuk bisa filter berdasarkan tahun	New	Romzi Q...
#389	Membuat halaman setiap beberapa detik (10 detik) update data	New	Not assign...
#390	Membuat keterangan setiap indikator warna	New	Not assign...

The page also shows a sidebar with navigation icons, a top navigation bar with 'Projects', and a right sidebar with 'POINTS', 'ASSIGNED', and 'WATCHERS' sections. The 'ASSIGNED' section lists 'Romzi Qutbi' and 'Lukas Wiku'. The 'WATCHERS' section has a '+ Add watchers' button and a 'Watch' button.

Gambar 3.9 *User Story* Membuat Halaman Detail Performa

Pada pengembangan halaman ini penulis mendapatkan 8 *task*, berikut penjelasan setiap *task* tersebut.

- a. Memperbaiki query *task* perorang. data pada halaman *Performance* di Timload tidak sinkron dengan data pada halaman detail performa Tim Javan. Pada Gambar 3.10 merupakan query yang sudah diperbaiki oleh penulis.

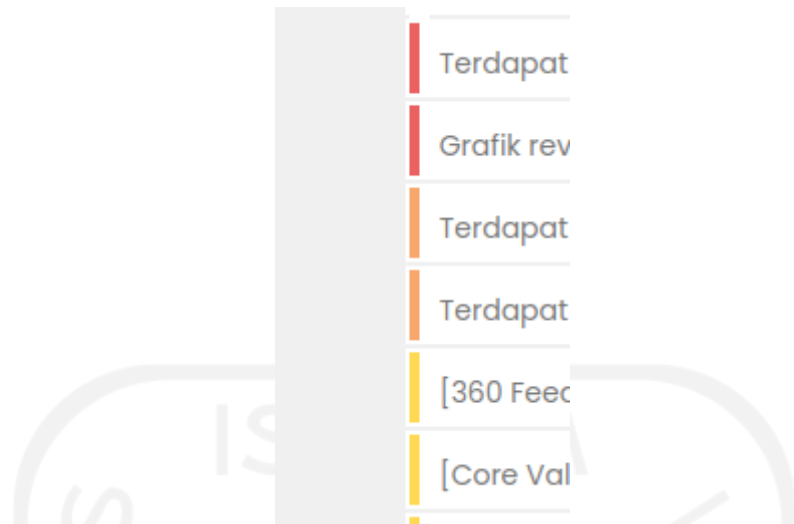
```

SELECT mea.nama_lengkap as nama,
       tasks_task.subject, projects_taskstatus.name, projects_project.name,
       projects_project.slug, tasks_task.ref, tasks_task.due_date, att.name
FROM tasks_task
LEFT JOIN
  projects_project ON tasks_task.project_id = projects_project.id
LEFT JOIN
  projects_taskstatus ON projects_taskstatus.id = tasks_task.status_id
LEFT JOIN
  custom_attributes_taskcustomattributesvalues as att_val ON
tasks_task.id = att_val.task_id
LEFT JOIN
  custom_attributes_taskcustomattribute as att
  ON att.id IN (
    SELECT
      jsonb_object_keys(att_val.attributes_values) :: INTEGER
  )
INNER JOIN
  master_employee_taiga ON master_employee_taiga.custom_field =
att_val.attributes_values ->> att.id :: VARCHAR
INNER JOIN
  master_employee_all mea ON master_employee_taiga.nip = mea.nip
WHERE
  tasks_task.created_date >= '{time[0]}'
  AND tasks_task.created_date <= '{time[1]} 24:00'
  AND mea.nama_lengkap = $$ {nama_staff} $$
  AND mea.status = 'Aktif'
ORDER BY tasks_task.due_date ASC

```

Gambar 3.10 Query *Task* Perorang

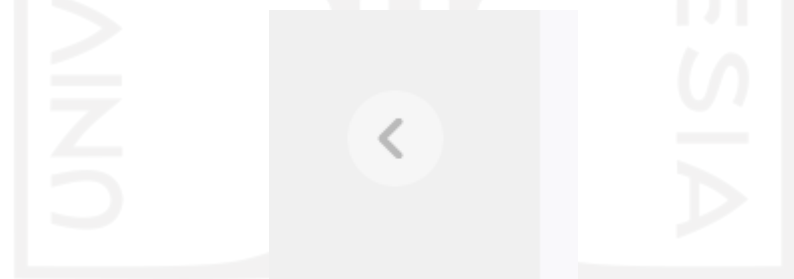
- b. Perbaiki *routing* halaman. Kadang-kadang dalam berganti dari satu halaman ke halaman yang lain terjadi *error*.
- c. Optimasi tampilan halaman detail perorang. Pada *task* ini penulis lumayan mendapatkan banyak hal yang perlu dikerjakan. Hal-hal yang perlu dikerjakan yaitu memberikan warna indikator pada masing-masing *task* yang masih aktif dengan setiap warna memiliki makna tertentu seperti Gambar 3.11, menambahkan link pada *subject task* sehingga ketika diklik membuka *tab* baru yang langsung mengarah halaman *task* yang diklik di Taiga Javan seperti pada Gambar 3.12, menambahkan tombol kembali ke halaman *performance* seperti pada Gambar 3.13, dan saat *hover* nama Tim Javan di halaman *performance* muncul garis bawah sehingga pengguna tahu kalau itu dapat diklik seperti pada Gambar 3.14.



Gambar 3.11 Indikator Warna

Terdapat perubahan mekanisme penyimpanan data pada cron reviewed	Testing
Terdapat perubahan mekanisme penyimpanan data pada vocabulary_user pada add vocabulary	Ready f
[360 Feedback] Terdapat Summary response dan Menhapus total employee pada tampilan stat...	Ready f

Gambar 3.12 Hover Subject Task

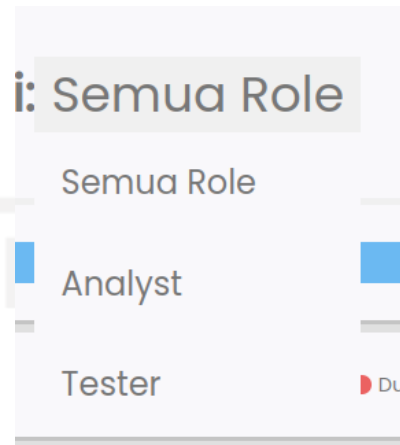


Gambar 3.13 Tombol Back

Muhammad Wahid Has...	Blue bar
<u>Maulvi Inayat Ali</u>	Blue, Orange, and Teal bars
Ahmad Rizqi Abdullah N...	Blue and Orange bars

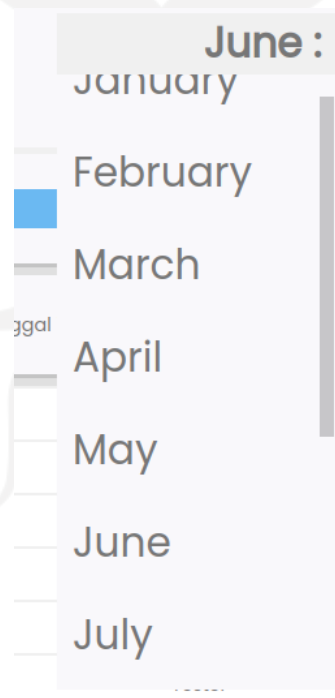
Gambar 3.14 Hover Nama Tim

- d. Menambah fitur untuk bisa *filter* setiap *role*. Di halaman detail performa tim javan menambahkan *filter* sebagai *role* apa saja Tim Javan pada berbagai *task* seperti pada Gambar 3.15.



Gambar 3.15 *Filter Role*

- e. Menambah fitur untuk bisa *filter* berdasarkan bulan. menambahkan *filter* untuk menampilkan data *task* dengan bulan yang diinginkan oleh pengguna seperti pada Gambar 3.16.



Gambar 3.16 *Filter Bulan*

- f. Menambah fitur untuk bisa *filter* berdasarkan tahun. menambahkan *filter* untuk menampilkan data *task* dengan tahun yang diinginkan oleh pengguna seperti pada Gambar 3.17.



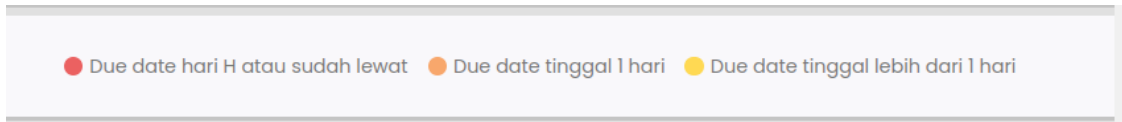
Gambar 3.17 Filter Tahun

- g. Membuat halaman setiap beberapa detik (10 detik) *update* data. Halaman detail performa Tim Javan dapat meng-*update* data-data yang ditampilkan agar setiap beberapa detik yang sudah ditentukan agar pengguna tidak perlu melakukan *refresh* manual. Untuk mengeksekusi *task* ini, penulis menggunakan *library* javascript bernama JQuery. Konsepnya seperti *refresh* manual tapi hanya untuk *element* tertentu saja pada HTML-nya. Kode program JQuery seperti pada Gambar 3.18.

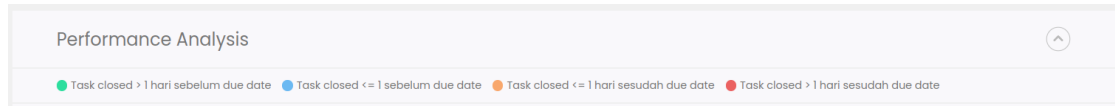
```
$(document).ready(function () {
  setInterval(function () {
    $.get(window.location.href, function(response){
      var $page = $(response)
      $('.progress-bar').replaceWith($page.find('.progress-bar'));
      $('.task-recap-sum').replaceWith($page.find('.task-recap-sum'));
      $('#ta-count').replaceWith($page.find('#ta-count'));
      $('#task-active').replaceWith($page.find('#task-active'));
      $('#tf-count').replaceWith($page.find('#tf-count'));
      $('#task-finished').replaceWith($page.find('#task-finished'));
    })
  }, 15000);
});
```

Gambar 3.18 JQuery *Auto-Refresh*

- h. Membuat keterangan setiap indikator warna. Memberikan keterangan setiap indikator warna pada detail performa agar pengguna tidak kebingungan dan lebih mudah paham sehingga mendapatkan *insight* dari halaman ini. Hasilnya seperti pada Gambar 3.19 pada fitur *Active Task* dan Gambar 3.20 pada fitur *Performance Analysis*.



Gambar 3.19 Keterangan Indikator Warna *Active Task*



Gambar 3.20 Keterangan Indikator Warna *Performance Analysis*

### 3.2.4 Pengembangan Project Scoreboard

Proyek ketiga yang dikerjakan oleh penulis adalah Project Scoreboard. Project Scoreboard adalah fitur tambahan pada *dashboard* Timload untuk mengukur performa semua *project* yang dikerjakan oleh PT Javan Cipta Solusi berdasarkan empat parameter yaitu *task* yang *closed* tepat waktu, *ready for test* tepat waktu, *task* yang *needs info*, dan rata-rata umur *task*. Pada Gambar 3.21 merupakan *user story* beserta *task-task* yang dikerjakan oleh penulis. terdapat 17 *task*.

The screenshot displays a Jira project page for the user story '#393 Membuat Project Scoreboard'. The main content area lists 17 tasks, each with a status (e.g., 'Closed', 'Ready for Test') and an assignee. The right-hand panel shows project statistics, including 'OPEN' items in progress, points, and assignees like Romzi Qulbi and Nydia Leony Charali. The bottom section shows '0 Comments' and '4 Activities'.

Gambar 3.21 User Story Membuat Project Scoreboard

- Membuat *mockup*. Sebelum masuk ke tahap pengkodean tampilannya harus didesain dulu, pada *task* ini penulis menggunakan Figma dalam mendesain.
- Query data dari *database* taiga Javan. Membuat query untuk mendapatkan data yang dibutuhkan khususnya data empat parameter perbulannya. Untuk hasil query-nya seperti pada Gambar 3.22. Query ini mengambil data *list project* dengan atribut nama *project*, bulan, tahun, total *task*, banyak *task* yang tepat waktu, rata-rata umur *task*, *project slug*, banyak *task* yang *ready for test* tepat waktu, dan banyak *task* yang *needs info*. Kemudian di-*grouping* berdasarkan *project*, bulan, dan tahun.



```

with tbl_p as (
    select
        p.name project_name, date_part('month', tt.due_date) bulan, date_part('year',
tt.due_date) tahun, count(*) total_task, count(case when extract(epoch from
to_timestamp(concat(to_char(tt.due_date , 'YYYY-MM-DD') , ' 11:59:59 PM'), 'YYYY-MM-DD
HH:MI:SS PM'))::integer >= extract(epoch from tt.finished_date) then 1 end )
on_time_task, avg(date_part('days',tt.finished_date-tt.created_date)) avg_task_age,
p.slug project_slug
    from
        projects_project p
    join tasks_task tt on p.id = tt.project_id
    join userstories_userstory us on us.id = tt.user_story_id
    where
        tt.due_date >= date_trunc('month', '{tahun}-{bulan}-15'::date) - interval '1
month' and tt.due_date < date_trunc('month', '{tahun}-{bulan}-15'::date) + interval '1
month' and tt.due_date is not null and tt.finished_date is not null {tags_q}
    group by
        bulan, tahun, p.name, slug
    order by
        p.name)
select
    tbl_p.project_name, tbl_p.bulan, tbl_p.tahun, tbl_p.total_task,
tbl_p.on_time_task, tbl_p.avg_task_age, tbl_p.project_slug,
    (
        select case when sum(rft.is_on_time)::integer is not null then
sum(rft.is_on_time)::integer else 0 end
        from(
            select
                distinct on (tt2.id)
                count(case when hh2.created_at <
to_timestamp(concat(to_char(tt2.due_date , 'YYYY-MM-DD') , ' 08:00:00'), 'YYYY-MM-DD
HH:MI:SS') then 1 end) is_on_time
            from
                projects_project p2
            join tasks_task tt2 on p2.id = tt2.project_id
            join userstories_userstory us2 on us2.id = tt2.user_story_id
            join history_historyentry hh2 on hh2.key = concat('tasks.task:', tt2.id)
            where
                date_part('year', tt2.due_date) = tbl_p.tahun and date_part('month',
tt2.due_date) = tbl_p.bulan and tt2.due_date is not null and tt2.finished_date is not
null and hh2.values_diff_cache::jsonb -> 'status' ->> -1 = 'Ready for test' and p2.name =
tbl_p.project_name
            group by
                tt2.id, hh2.created_at
            order by
                tt2.id, hh2.created_at desc
        ) as rft
    ) on_time_ready_for_test,
    (
        select
            count(distinct tt3.id)
        from
            projects_project p3
            join tasks_task tt3 on p3.id = tt3.project_id
            join userstories_userstory us3 on us3.id = tt3.user_story_id
            join history_historyentry hh3 on hh3.key = concat('tasks.task:', tt3.id)
            where
                date_part('year', tt3.due_date) = tbl_p.tahun and date_part('month',
tt3.due_date) = tbl_p.bulan and tt3.due_date is not null and tt3.finished_date is not
null and hh3.values_diff_cache::jsonb -> 'status' ->> -1 = 'Needs Info' and p3.name =
tbl_p.project_name
        ) needs_info
    from
        tbl_p

```

Gambar 3.22 Query Project Scoreboard Untuk *List Project*

- c. Membuat *scoring project*. Membuat *scoring* atau pembobotan terhadap masing-masing parameter. Penulis memberi masing-masing parameter nilai skor seperti pada Tabel 3.5, Tabel 3.6, Tabel 3.7, dan Tabel 3.8. Dari *score* parameter-parameter yang sudah dihitung akan didapatkan skor final dengan cara mengakumulasikan setiap *score* parameter. Kemudian dikategorikan ke dalam empat kategori seperti pada Tabel 3.9.

Tabel 3.5 *Scoring Task* Tepat Waktu

Persentase Task Tepat Waktu	Score (1-10)
$\geq 95\%$	10
$<95\%$ and $\geq 85\%$	9
$<85\%$ and $\geq 75\%$	8
$<75\%$ and $\geq 65\%$	7
$<65\%$ and $\geq 55\%$	6
$<55\%$ and $\geq 45\%$	5
$<45\%$ and $\geq 35\%$	4
$<35\%$ and $\geq 25\%$	3
$<25\%$ and $\geq 15\%$	2
$<15\%$ and $\geq 5\%$	1
$< 5\%$	0

Tabel 3.6 *Scoring Task Ready For Test* Tepat Waktu

Persentase Ready For Test Tepat Waktu	Score (0-5)
$\geq 95\%$	5
$<95\%$ and $\geq 85\%$	4
$<85\%$ and $\geq 75\%$	3
$<75\%$ and $\geq 65\%$	2
$<65\%$ and $\geq 55\%$	1
$<55\%$	0

Tabel 3.7 *Scoring Task Needs Info*

Persentase SLA Needs Info	Score (0-5)
$<5\%$	5
$\geq 5\%$ and $< 10\%$	4
$\geq 10\%$ and $< 15\%$	3
$\geq 15\%$ and $< 20\%$	2
$\geq 20\%$ and $< 25\%$	1

$\geq 25$	0
-----------	---

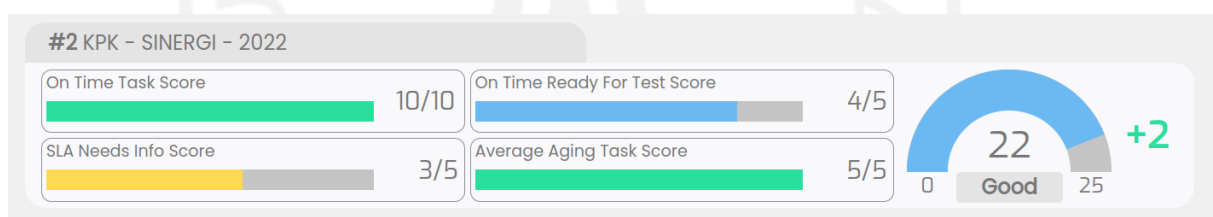
Tabel 3.8 *Scoring Rata-rata Umur Task*

Rata-Rata Umur Task (Hari)	Score (0-5)
$\leq 3$	5
$>3$ and $\leq 4$	4
$>4$ and $\leq 5$	3
$>5$ and $\leq 6$	2
$>6$ and $\leq 7$	1
$>7$	0

Tabel 3.9 Kategori *Final Score*

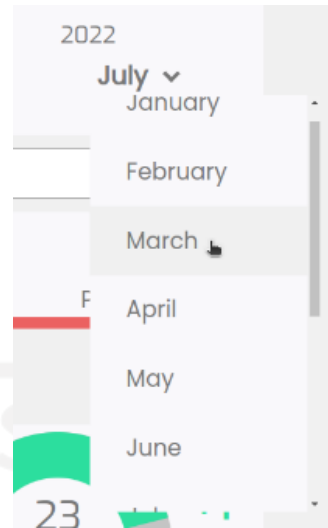
Mode	Score
Excellent	$\geq 23$
Good	$<23$ and $\geq 21$
Fair	$<21$ and $\geq 15$
Poor	$<15$

- d. Membuat *card project*. Membuat tampilan *card project* untuk setiap *project* seperti pada Gambar 3.23.



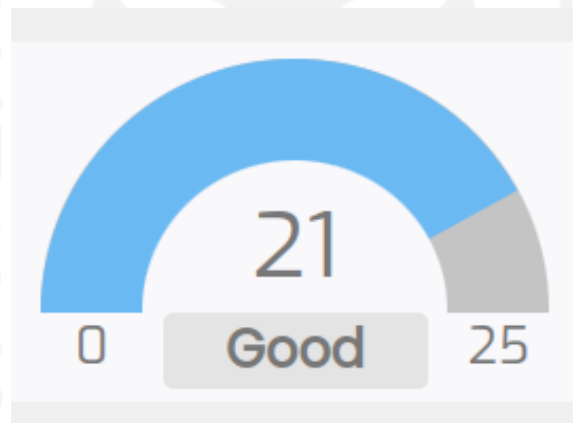
Gambar 3.23 Card Project

- e. Membuat *filter* berdasarkan bulan dan tahun pada halaman scoreboard. Untuk hasil akhirnya seperti pada Gambar 3.24.



Gambar 3.24 Filter Bulan dan Tahun Project Scoreboard

- f. Membuat *gauge chart* untuk *score project*. *Gauge chart* dibuat untuk memvisualisasikan nilai *score* yang sudah diraih suatu *project* dari *score* maksimal seperti pada Gambar 3.25.



Gambar 3.25 Gauge Chart Project Scoreboard

- g. Membuat desain detail performance. Untuk membuat desain halaman detail performance dari *project*, penulis menggunakan tool bernama figma juga.
- h. Query data dari *database* taiga. Query pada *task* ini maksudnya query untuk halaman detail performance untuk project scoreboard. Query-nya seperti pada Gambar 3.26. Query ini mengambil data dengan *atribut* nama *project*, tahun, minggu, total *task*, banyak *task* yang tepat waktu, banyak *task* yang *ready for test* tepat waktu, rata-rata umur *task*, dan banyak *task* yang *needs info*. Kemudian di-*grouping* berdasarkan nama *project*, tahun, dan minggu.

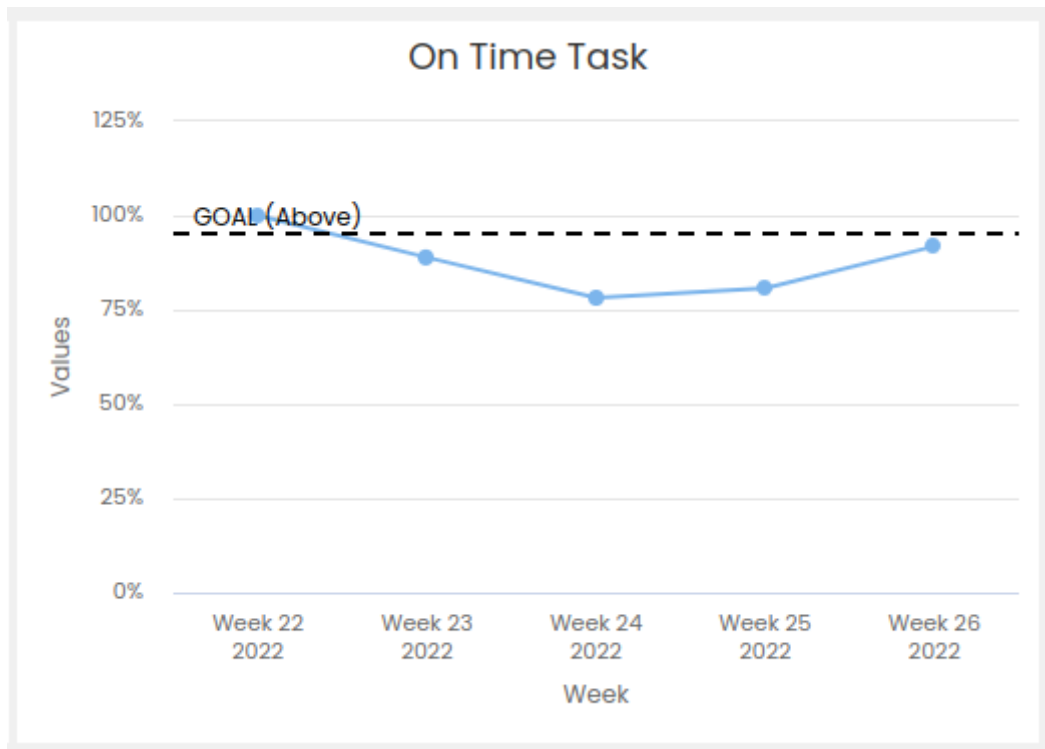
```

with tabel_baru as(
  select
    p.name nama_project, date_part('year', date_trunc('week', tt.due_date)) as
    year, date_part('week', tt.due_date) as week, count(*) total_task, count(case
    when extract(epoch from to_timestamp(concat(to_char(tt.due_date , 'YYYY-MM-
    DD') , ' 11:59:59 PM')), 'YYYY-MM-DD HH:MI:SS PM'))::integer >= extract(epoch
    from tt.finished_date) then 1 end ) on_time_task,
    sum(date_part('days',tt.finished_date-tt.created_date)) task_age
  from projects_project p
  join tasks_task tt on p.id = tt.project_id
  join userstories_userstory us on us.id = tt.user_story_id
  where
    date_part('year', tt.due_date) = {tahun} and date_part('month',
    tt.due_date) = {bulan} and tt.due_date is not null and tt.finished_date is not
    null and p.slug = '{slug}' and p.name not in {self.exlcuded_Project}
    group by year, week, nama_project
    order by year, week )
  Select
    tabel_baru.nama_project, tabel_baru.year, tabel_baru.week,
    tabel_baru.total_task, tabel_baru.on_time_task, tabel_baru.task_age, (select
    case when sum(rft.is_on_time)::integer is not null then
    sum(rft.is_on_time)::integer else 0 end
  from(
    select
      distinct on (tt2.id) count(case when hh2.created_at <
      to_timestamp(concat(to_char(tt2.due_date , 'YYYY-MM-DD') , ' 08:00:00'),
      'YYYY-MM-DD HH:MI:SS') then 1 end) is_on_time
    from projects_project p2
    join tasks_task tt2 on p2.id = tt2.project_id
    join userstories_userstory us2 on us2.id = tt2.user_story_id
    join history_historyentry hh2 on hh2.key = concat('tasks.task:', tt2.id)
    where
      date_part('year', tt2.due_date) = {tahun} and date_part('month',
      tt2.due_date) = {bulan} and tt2.due_date is not null and tt2.finished_date
      is not null and hh2.values_diff_cache::jsonb -> 'status' ->> -1 = 'Ready
      for test' and p2.slug = '{slug}' and date_part('week', tt2.due_date) =
      tabel_baru.week
      group by tt2.id, hh2.created_at
      order by tt2.id, hh2.created_at desc ) as rft
    ) on_time_ready_for_test,
    ( select
      count(distinct tt3.id)
    from projects_project p3
    join tasks_task tt3 on p3.id = tt3.project_id
    join userstories_userstory us3 on us3.id = tt3.user_story_id
    join history_historyentry hh3 on hh3.key = concat('tasks.task:', tt3.id)
    where
      date_part('year', tt3.due_date) = {tahun} and date_part('month',
      tt3.due_date) = {bulan} and tt3.due_date is not null and tt3.finished_date
      is not null and hh3.values_diff_cache::jsonb -> 'status' ->> -1 = 'Needs
      Info' and p3.slug = '{slug}' and date_part('week', tt3.due_date) =
      tabel_baru.week
    ) needs_info
  from tabel_baru

```

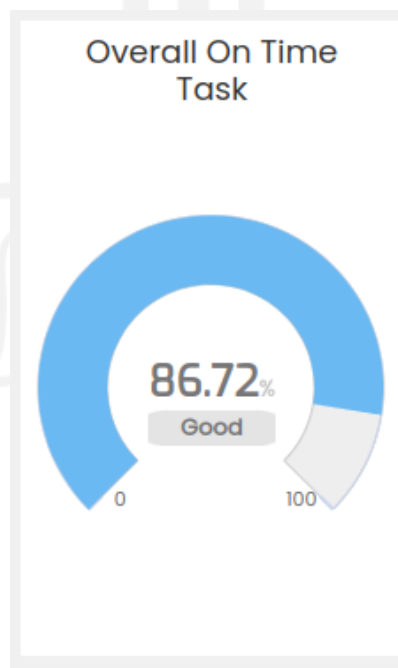
Gambar 3.26 Query Detail Performance Project Scoreboard

- i. Membuat *line chart* project performance visualisasi masing-masing variabel. *Line chart* ini dibuat pada halaman detail. Tampilannya seperti pada Gambar 3.27 untuk *task* yang tepat waktu.



Gambar 3.27 *Line Chart* Detail Peforma

- j. Membuat *gauge chart* project performance visualisasi masing-masing variabel. *Gauge chart* untuk halaman detail dari masing parameter atau variabel yang diukur. Pada Gambar 3.28 merupakan tampilan *gauge chart* untuk parameter task tepat waktu.

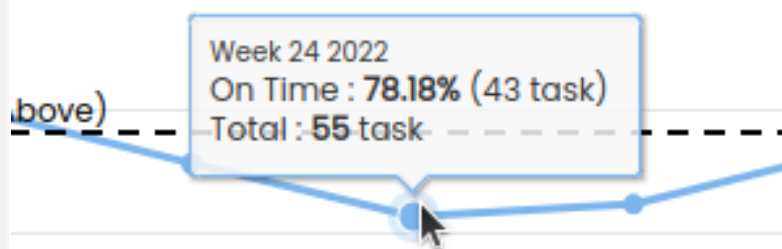


Gambar 3.28 *Gauge Chart* Detail Peforma

- k. Membuat *filter* bulan dan tahun. *Filter* bulan dan tahun untuk halaman detail. Tampilannya sama seperti pada halaman project scoreboard.
- l. Membuat tombol *back* ke halaman *index* project scoreboard. Tampilannya seperti pada Gambar 3.29.

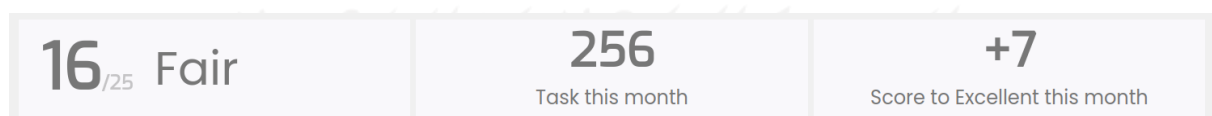
Gambar 3.29 Tombol *Back* Detail Performa Project Scoreboard

- m. Menambah keterangan jumlah *task* waktu *hover line chart*. Hasilnya seperti pada Gambar 3.30.



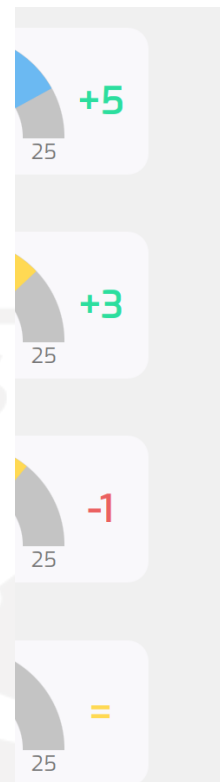
Gambar 3.30 *Hover Line Chart*

- n. Menambah *summary* pada halaman detail performa. Hasil akhirnya seperti pada Gambar 3.31.



Gambar 3.31 *Summary* Detail Performa Project Scoreboard

- o. Menambah *trend* pada halaman *index* project scoreboard. Menambah *trend* maksudnya adalah membandingkan posisi urutan *project* berdasarkan bulan sebelumnya jika naik maka dilambangkan dengan tanda “+” terus diikuti nilai naik berapa tingkat, jika turun maka dilambangkan dengan tanda “-” kemudian diikuti nilai turun berapa tingkat, jika data *project* pada bulan sebelumnya tidak ditemukan maka dilambangkan dengan tanda “?”. Untuk tampilannya seperti pada Gambar 3.32.



Gambar 3.32 Tampilan *Trend Project Scoreboard*

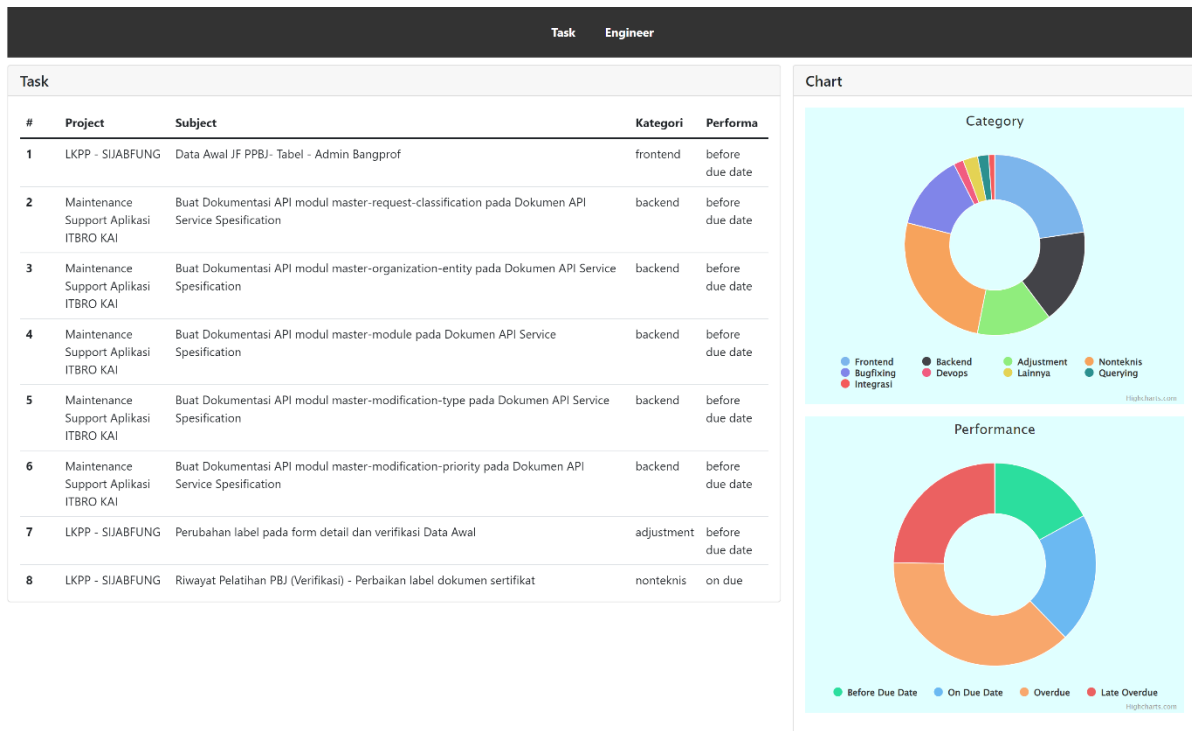
### 3.3 Hasil dan Pembahasan Pelaksanaan Magang

Selama menajalani proses magang selama 6 bulan dimaulai dari bulan 1 September 2021 sampai dengan 28 Februari 2022. Penulis memberikan kontribusi ke PT Javan Cipta Solusi berupa hasil dari proyek yang dikerjakan. Secara garis besar terdapat 3 *project* yaitu *Dashboard Profiling Engineer*, Halaman Detail Performa, dan Project Scoreboard.

#### 3.3.1 *Dashboard Profiling Engineer*

Produk pertama yang penulis selesaikan pada saat melaksanakan magang adalah *Dashboard Profiling Engineer*. *Dashboard* ini dibuat bertujuan untuk mengetahui performa setiap *engineer* di PT Javan Cipta Solusi sebagai bahan evaluasi ke depannya. *Dashboard* ini dibuat karena banyaknya data *task* Taiga yang perlu diolah dan sangat disayangkan jika dibiarkan begitu saja. *Dashboard* memiliki 3 fitur utama yaitu *Task*, *Engineer*, dan *Detail Engineer*. Pada Gambar 3.33 merupakan halaman awal dari *Dashboard* ini sekaligus fitur *Task*.



Gambar 3.33 Halaman *Task*

Pada fitur *task* terdapat detail *list task* yang ter-generate sejak 6 bulan sebelumnya detail *task*-nya berupa nama-nama proyek dari *task*, *subject task*, kategori *task*, dan performa *task*. Selain itu juga, terdapat 2 *donut chart* yang masing-masing memvisualisasikan kategori dan performa *task*.

Pada *chart* kategori terdapat 9 label dan setiap *task* memiliki labelnya masing-masing satu. Setiap *task* yang memiliki label tertentu diakumulasikan dan ditotalkan kemudian disajikan dalam bentuk *donut chart* agar lebih mudah dipahami. Dalam menentukan label dari masing-masing *task* ini penulis menggunakan teknologi *Machine Learning* yang modelnya sudah dilatih dengan 1000 sampel data *task* Taiga.

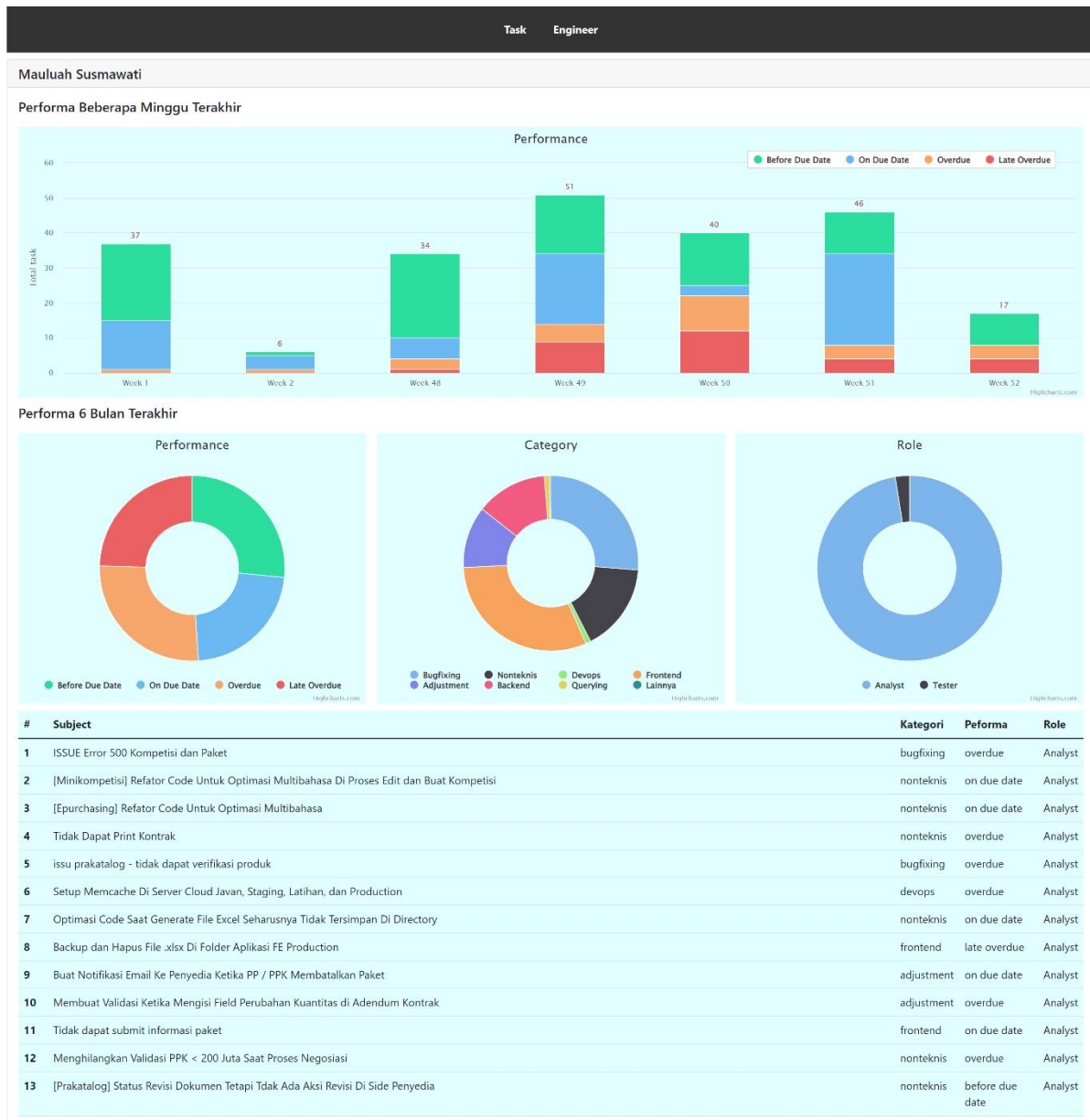
Pada *chart* Performa terdapat 4 label yang masing-masing mempunyai arti tertentu. Label *before due date* (hijau) menandakan *task* dinyatakan closed pada saat lebih dari satu hari sebelum *due date*, *on due date* (biru) menandakan *task* closed pada saat satu hari sebelum *due date* atau tepat *due date*, *overdue* menandakan *task* closed pada saat *due date* sudah lewat sebelum atau tepat satu hari, *late overdue* menandakan *task* closed pada sudah saat *due date*-nya lewat lebih dari satu hari.

Pada Gambar 3.34 merupakan fitur kedua yaitu *Engineer* yang berupa rangkuman daftar semua performa *engineer* pada saat mengerjakan *task*. Daftar ini diurutkan berdasarkan banyak *task* yang dikerjakan oleh masing-masing *engineer*.



Gambar 3.34 Halaman *Engineer*

Fitur yang terakhir yaitu detail *engineer* merupakan detail dari semua *task* yang dikerjakan oleh setiap *engineer* berupa performa, kategori, dan *role*-nya. Pada fitur ini terdapat 4 *chart*. *Chart* yang pertama yaitu *column chart* yang merangkum performa setiap minggunya dari *engineer* sebanyak 8 minggu ke belakang. *Chart* yang kedua yaitu *donut chart* yang merangkum performa *engineer* selama 6 bulan ke belakang. *Chart* yang ketiga yaitu *donut chart* yang merangkum kategori *task* yang dikerjakan selama 6 bulan ke belakang. Yang terakhir yaitu *donut chart* yang merangkum *role* dari *engineer* dalam mengerjakan *task* dalam 6 bulan ke belakang. Selain itu juga terdapat tabel *task* yang berisi *subject task*, kategori *task*, performa *task*, dan *role engineer* dari *task* tersebut dalam 6 bulan ke belakang. Pada Gambar 3.35 merupakan tampilan dari fitur detail *engineer* dari *Dashboard Profiling Engineer*.

Gambar 3.35 Halaman Detail *Engineer*

### 3.3.2 Halaman Detail Performa Analisis

Pada tahap analisis, penulis akan menganalisis masalah yang ingin dipecahkan untuk mencari solusinya. Analisis dapat dilakukan dengan berbagai cara seperti riset, diskusi dengan *user* atau pengguna yang akan menggunakan fitur ini. Halaman detail performa dikembangkan untuk memecahkan masalah di mana setiap tim di Javan dapat melihat performa masing-masing tim yang mengerjakan *task* agar nantinya digunakan sebagai bahan evaluasi internal. Hasil dari tahap analisis untuk halaman detail performa ini yaitu:

- a. Skema data-data yang dibutuhkan yaitu nama, *subject task*, status *task*, *project task*, *deadline task*, dan *role* perbulannya, untuk lebih jelasnya *query* dapat dilihat pada anak subbab 3.2.3 di bagian a.
- b. Selain itu juga terdapat data-data yang dibutuhkan untuk menganalisis performa dengan skema data yang sama akan tetapi dalam rentang waktu yang lebih lama yaitu 6 bulan.

### Desain

Setelah melakukan analisis selanjutnya adalah tahap desain. Pada tahap ini penulis menerjemahkan hasil dari analisis ke dalam bentuk bagaimana halaman ini akan dikembangkan. Pada tahap ini beberapa hal ditentukan yaitu:

- a. Desain *mockup* halaman ditentukan, untuk desain dapat dilihat pada Gambar 3.36, Gambar 3.37, Gambar 3.38, dan Gambar 3.39. Dari pemilihan warna menyesuaikan dengan tema warna yang sudah digunakan pada Timload dan keputusan dari *supervisor*.
- b. URL dari halaman yaitu `{{host}}/hcd/performance/recaps/{{nama_tim}}`.
- c. Skenario bagaimana halaman dibuka yaitu:
  1. Akses halaman *home*.
  2. Pilih nama tim yang ingin dilihat performanya.
- d. *Output* setelah melakukan menjalankan sesuai dengan skenario yaitu halaman detail performa akan tampil.

### Implementasi

Pada tahap ini penulis sebagai developer akan melakukan pengkodean sesuai dengan desain yang sudah ditentukan. Dalam melakukan pengkodean, penulis menggunakan beberapa teknologi yaitu Visual Studio Code sebagai *code editor*, Django sebagai *framework*, gitlab sebagai *repository*. Untuk lebih lengkap bagaimana tahap implementasi dikerjakan oleh penulis dapat dilihat pada anak subbab 3.2.3.

### Evaluasi

Pada tahap ini fitur yang sudah dibangun akan diuji. Kode yang sudah ditulis *developer* pada tahap implementasi akan di-*push* ke gitlab *repository* pada *branch develop* agar masuk ke dalam *server staging*. Pada *server staging* fitur akan diuji oleh *supervisor* apakah sesuai dengan skenario pada tahap desain. Jika tidak sesuai skenario, maka akan kembali ke tahap desain untuk direvisi. Jika sudah sesuai skenario maka akan langsung di-*merge* ke *branch master* agar masuk ke tahap *deployment* dan langsung digunakan oleh pengguna.

### Penjelasan

Produk kedua yang penulis bantu kembangkan adalah berupa fitur untuk *tools* internal di PT Javan Cipta Solusi. Fitur ini berupa *webpage* dari performa masing-masing Tim Javan berdasarkan *task* yang dikerjakan di Taiga. Fitur ini juga tidak terlepas dari *dashboard* yang penulis kembangkan sebelumnya karena dalam fitur ini mengambil fitur dari *Dashboard Profiling Engineer* pada bagian detail engineer khususnya pada bagian *chart-chart*. Pada halaman ini terdapat beberapa fitur utama yaitu Task Progress, Active Task, Performance Analysis, Finished Task.

Fitur yang pertama yaitu Task Progress, fitur menunjukkan akumulasi banyak *task* yang dikerjakan oleh tim dan mengkategorikannya berdasarkan status *task* tersebut seperti pada Gambar 3.36. Warna biru menandakan *task* yang sudah *closed* atau *finish* sedangkan warna yang lain masih dinyatakan masih aktif. Beberapa status *task* yang belum *closed* adalah *new* yaitu *task* yang baru dibuat akan tetapi belum siap dikerjakan oleh *programmer* dan harus dianalisis oleh analis, *ready* yaitu *task* yang sudah siap dikerjakan oleh *programmer*, *code review* yaitu *task* yang sudah siap untuk di-*review* oleh *code reviewer*, *ready for test* yaitu *task* yang sudah siap untuk dilakukan *testing* oleh *tester*, *feedback* yaitu jika *task* memiliki kekurangan dan ada masukan, *needs info* yaitu *task* yang masih memerlukan informasi lebih atau *task*-nya kurang jelas maksudnya.



Gambar 3.36 Task Progress

Fitur yang kedua yaitu fitur active task yaitu daftar *task-task* yang belum *closed*. Pada active task terdapat detail *task* yang dikerjakan oleh Tim Javan yaitu *subject task*, *status task*, asal proyek *task*, role tim di *task*, dan due date. pada bagian paling kiri setiap baris *task* terdapat indikator warna yang masing-masing warna menandakan hal yang berbeda. Warna merah menandakan *task* sudah mencapai hari *due date* atau sudah melampaui hari *due date*, warna *orange* menandakan satu hari lagi *task* sudah *due date*, warna kuning menandakan dua hari atau lebih *task* akan mencapai *due date*. Pada Gambar 3.37 merupakan tampilan fitur active task.

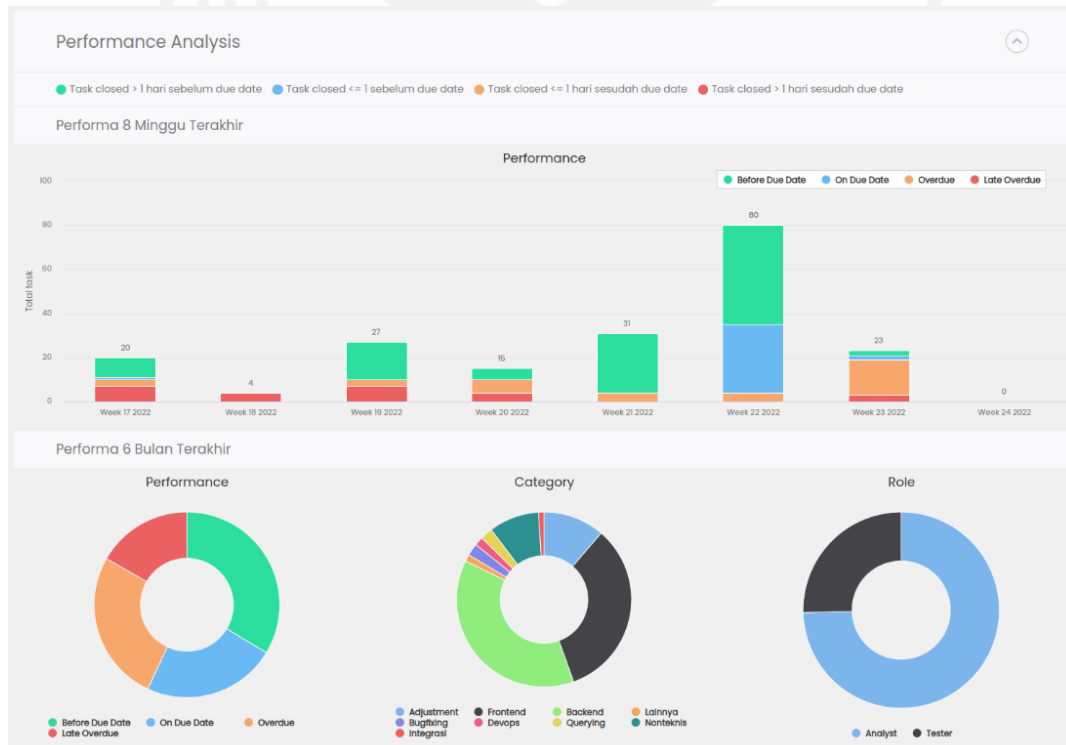
Active task : 30

● Due date hari H atau sudah lewat ● Due date tinggal 1 hari ● Due date tinggal lebih dari 1 hari

Task	Status	Project	Role	Due Date
[FE] Terdapat Halaman List LOA value config pada menu financial review	Testing	XL-LMS (FR)-2022	Analyst	June 15, 2022
[FE] Terdapat Halaman Update LOA value config pada menu financial review	Ready for test	XL-LMS (FR)-2022	Analyst	June 15, 2022
[INTEGRASI] pada Halaman Update LOA Value Config dapat melakukan update data yang telah di...	Ready for test	XL-LMS (FR)-2022	Analyst	June 16, 2022
[INTEGRASI] pada Halaman List LOA Value Config menampilkan list data , Bisa search , mengubah ...	Testing	XL-LMS (FR)-2022	Analyst	June 16, 2022
[INTEGRASI] pada Halaman Add New LOA Value Config dapat melakukan submit data yang telah ...	Testing	XL-LMS (FR)-2022	Analyst	June 16, 2022
Terdapat Halaman List Create FR pada menu financial review	In progress	XL-LMS (FR)-2022	Analyst	July 8, 2022
Terdapat API view detail pada menu modify FR Macro	Ready	XL-LMS (FR)-2022	Analyst	None
Terdapat API upload supporting document pada modify FR Macro	Ready	XL-LMS (FR)-2022	Analyst	None

Gambar 3.37 Active Task

Fitur yang ketiga yaitu Performance Analysis. Fitur ini kurang lebih sama dengan fitur detail *engineer* pada *Dashboard Profiling Engineer* tetapi tanpa daftar detail *task* yang dikerjakan *engineer* tersebut. Untuk fitur ini sudah dijelaskan pada subbab 3.3.1. Pada Gambar 3.38 merupakan tampilan dari fitur ini. Pada *chart* kategori terdapat 9 label dan setiap *task* memiliki labelnya masing-masing satu. Setiap *task* yang memiliki label tertentu diakumulasikan dan ditotalkan kemudian disajikan dalam bentuk *donut chart* agar lebih mudah dipahami. Dalam menentukan label dari masing-masing *task* ini penulis menggunakan teknologi *Machine Learning* dengan model *Support Vector Machine*.



Gambar 3.38 Performance Analysis

Fitur yang keempat yaitu daftar finished task. Fitur ini bertujuan untuk merangkum semua *task* yang sudah ditutup pada bulan ini, kurang lebih seperti fitur Active Task akan tetapi

dengan indikator warna yang berbeda yaitu warna biru yang menandakan *task* sudah selesai. Pada Gambar 3.39 merupakan tampilan dari fitur ini.

Finished task : 86

Task	Status	Project	Role	Due Date
Terdapat Api view detail pada menu master P2M Purpose	Closed	XL-LMS (FR)-2022	Analyst	June 7, 2022
Terdapat API Delete pada menu master P2M Purpose	Closed	XL-LMS (FR)-2022	Analyst	June 7, 2022
simpel Production terdapat LOG debugging dan sentry	Closed	Ombudsman - SIMPEL - 2022	Tester	June 7, 2022
Terdapat API Submit pada menu master P2M Purpose	Closed	XL-LMS (FR)-2022	Analyst	June 7, 2022
Terdapat API Update pada menu master P2M Purpose	Closed	XL-LMS (FR)-2022	Analyst	June 7, 2022
terdapat table master p2m purpose pada cloud	Closed	XL-LMS (FR)-2022	Analyst	June 7, 2022
simpel Production terdapat LOG debugging dan sentry	Closed	Ombudsman - SIMPEL - 2022	Analyst	June 7, 2022
Terdapat API List pada menu master P2M Purpose	Closed	XL-LMS (FR)-2022	Analyst	June 7, 2022
Terdapat API P2M Purpose Status pada menu master P2M Purpose	Closed	XL-LMS (FR)-2022	Analyst	June 7, 2022

Gambar 3.39 Finished Task

### 3.3.3 Project Scoreboard

#### Analisis

Pada tahap analisis, penulis akan menganalisis masalah yang ingin dipecahkan untuk mencari solusinya. Analisis dapat dilakukan dengan berbagai cara seperti riset, diskusi dengan *user* atau pengguna yang akan menggunakan fitur ini. Project Scoreboard dibuat untuk memonitor performa *project* yang sedang dikembangkan oleh PT Javan Cipta Solusi berdasarkan 4 parameter yaitu:

- Persentase *task* yang *closed* tepat waktu (jika *task closed* sebelum atau tepat pada tanggal *deadline task*, maka *task* dikatakan tepat waktu). Semakin tinggi persentase *task closed* tepat waktu maka semakin baik.
- Persentase *task ready for test* tepat waktu (*ready for test* dikatakan tepat waktu jika sebelum atau tepat jam 8 pagi *deadline task*). Semakin tinggi persentase *task ready for test* tepat waktu maka semakin baik.
- Persentase *task needs info*. Jika *programmer* belum paham mengenai *task* yang dibuat oleh analis maka *programmer* akan mengubah status *task* ke *needs info* dan dilempar kembali ke analis untuk dianalisis kembali. Jika ini terjadi maka akan memakan waktu lebih dari seharusnya lama *task* dikerjakan. Semakin rendah persentase *needs info* maka semakin baik.
- Rata-rata umur *task*. Idealnya umur *task* adalah tiga hari, semakin pendek umur *task* maka semakin baik.

Dari 4 parameter tersebut kemudian dilakukan pembobotan atau *scoring* dengan porsi *task closed* tepat waktu sebesar 40%, dan 3 parameter yang lainnya masing-masing 20%. Untuk



aturan pembobotan dapat dicek pada anak subbab 3.2.4 di bagian c. Porsi dan aturan pembobotan merupakan hasil diskusi antara penulis dengan *Project Manager Officer* (PMO) dan *supervisor*. Untuk *query* data yang dibutuhkan dapat dicek pada anak subbab 3.2.4 di bagian b dan h.

### Desain

Setelah melakukan analisis selanjutnya adalah tahap desain. Pada tahap ini penulis menerjemahkan hasil dari analisis ke dalam bentuk bagaimana halaman ini akan dikembangkan. Pada tahap ini beberapa hal ditentukan yaitu:

- e. Desain *mockup*. Terdapat 2 desain yaitu halaman index seperti pada Gambar 3.40 dan halaman detail seperti pada Gambar 3.41. Dari pemilihan warna menyesuaikan dengan tema warna yang sudah digunakan pada Timload dan diskusi dengan *supervisor*.
- f. URL untuk halaman index yaitu {{host}}/pm/project-scoreboard dan untuk halaman detail yaitu {{host}}/pm/project-scoreboard/{{project-slug}}.
- g. Skenario

Halaman index:

1. Akses halaman *home*.
2. Pilih fitur Project Scoreboard.

Halaman detail:

1. Akses halaman *home*.
2. Pilih fitur Project Scoreboard.
3. Pilih salah satu dan klik nama *project* yang ingin dilihat detailnya.

- h. *Output* Pada halaman index menampilkan *list project* dan *score* setiap parameter yaitu *On Time Task Score*, *SLA Ready For Test Score*, *SLA Needs Info Score*, dan *Average Aging Task Score*. Dan sekaligus akumulasi dari skor tersebut yang akan menentukan skor akhir dari *project*.

*Output* Pada halaman detail menampilkan chart dari masing masing variabel yang menentukan Performa project yaitu:

1. *Line chart* banyak *task ready for test* tepat waktu berdasarkan minggu dan *gauge chart* berdasarkan banyak *task ready for test* tepat waktu keseluruhan.
2. *Line chart* rata-rata umur *task* perminggunya dan *gauge chart* berdasarkan banyak rata-rata umur *task* keseluruhan.
3. *Line chart* banyak *task needs info* perminggunya dan *gauge chart* berdasarkan *task needs info* keseluruhan.



4. *Line chart* banyak *task closed* tepat waktu perminggunya dan *gauge chart* berdasarkan banyak *task closed* tepat waktu keseluruhan.

### **Implementasi**

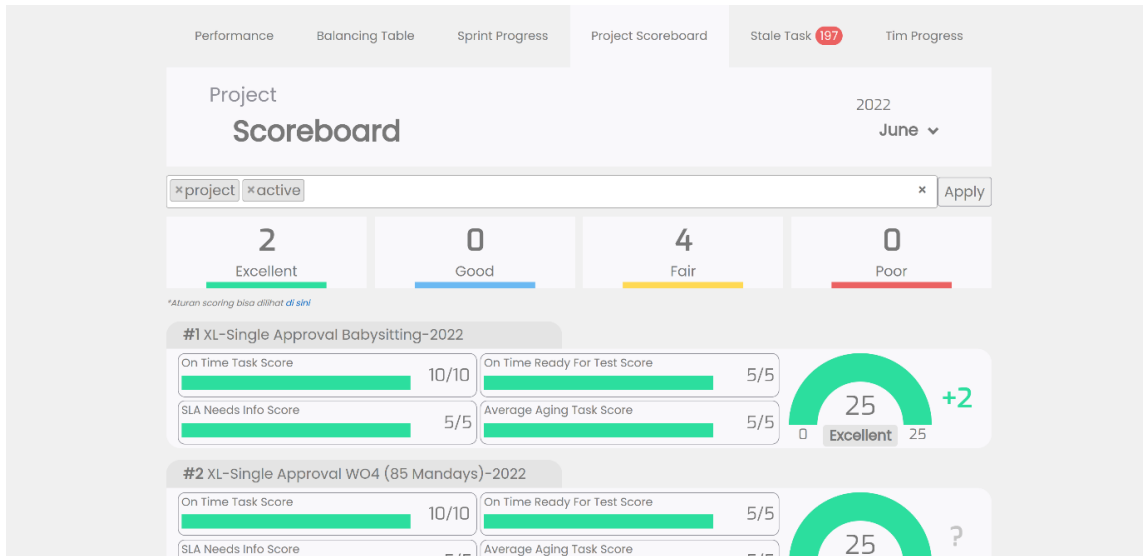
Pada tahap ini penulis sebagai developer akan melakukan pengkodean sesuai dengan desain yang sudah ditentukan. Dalam melakukan pengkodean, penulis menggunakan beberapa teknologi yaitu Visual Studio Code sebagai *code editor*, Django sebagai *framework*, gitlab sebagai *repository*. Untuk lebih lengkap bagaimana tahap implementasi dikerjakan oleh penulis dapat dilihat pada anak subbab 3.2.4.

### **Evaluasi**

Pada tahap ini fitur yang sudah dibangun akan diuji. Kode yang sudah ditulis *developer* pada tahap implementasi akan di-*push* ke gitlab *repository* pada *branch develop* agar masuk ke dalam *server staging*. Pada *server staging* fitur akan diuji oleh *supervisor* apakah sesuai dengan skenario pada tahap desain. Jika tidak sesuai skenario, maka akan kembali ke tahap desain untuk direvisi. Jika sudah sesuai skenario maka akan langsung di-*merge* ke *branch master* agar masuk ke tahap *deployment* dan langsung digunakan oleh pengguna.

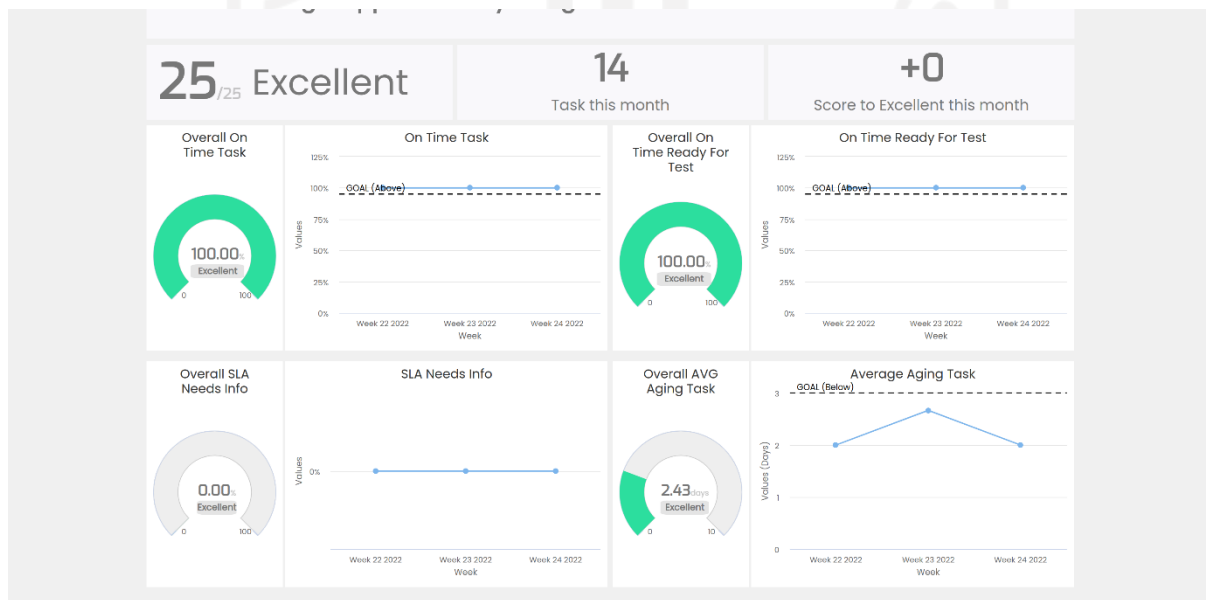
### **Penjelasan**

Project Scoreboard adalah produk ketiga yang penulis kembangkan. Project scoreboard memiliki dua halaman yaitu halaman *index* dan halaman detail. Pada halaman *index* menampilkan semua *project* yang pengembangannya masih berjalan di PT Javan Cipta Solusi. Semua *project* diurutkan berdasarkan skor yang paling besar hingga skor yang paling kecil berdasarkan performa perbulannya. Pada halaman ini juga menampilkan apakah *project* itu statusnya *excellent*, *good*, *fair*, ataupun *poor*. *Goal*-nya adalah semua *project* masuk ke dalam kategori *excellent*. Pada Gambar 3.40 merupakan tampilan halaman *index*.



Gambar 3.40 Halaman *Index* Project Scoreboard

Halaman kedua adalah halaman detail. Seperti namanya, halaman ini menampilkan detail dari *project* yang ada di halaman *index*. Halaman detail menampilkan *summary* dari *project* berupa status dari *project*, banyak *task* pada bulan ini, dan berapa skor lagi untuk mencapai *excellent*. Selain itu juga, halaman ini menampilkan *trend* dari setiap parameter yang diukur perminggunya pada bulan tertentu. Pada Gambar 3.41 merupakan tampilan dari halaman detail.



Gambar 3.41 Halaman Detail Performa Project Scoreboard

## BAB IV

### REFLEKSI PELAKSANAAN MAGANG

#### 4.1 Relevansi Akademik

Dalam pengimplementasian *balanced scorecard* pada *dashboard* timload, tidak mencakup semua perspektif yang ada pada *balanced scorecard*. Pada *dashboard* timload hanya mencakup tiga perspektif yaitu perspektif keuangan, perspektif proses bisnis internal, dan perspektif pembelajaran dan pertumbuhan. Di masa perkuliahan penulis tidak pernah mempelajari tentang *balanced scorecard* ini, karena materi ini lebih berkaitan dengan manajemen dan bisnis. Akan tetapi ini adalah suatu hal yang menarik bagi penulis karena mempelajari suatu hal yang baru di luar bidang keilmuan informatika.

Dalam mengerjakan *project* pada saat magang penulis menggunakan *design pattern Model-View-Template* karena *design pattern* ini adalah bawaan *default* dari *web framework* *django*, dan *design pattern* ini adalah hal yang baru juga bagi penulis karena pada masa perkuliahan biasanya mempelajari tentang *design pattern Model-View-Controller*. Menurut penulis dari dua *design pattern* ini tidak terlalu memiliki perbedaan yang signifikan.

#### 4.2 Pembelajaran Magang

Setelah menjalani proses magang selama enam bulan di PT Javan Cipta Solusi, banyak sekali pelajaran yang dapat diambil dari *project-project* yang dikerjakan. Pembelajaran berupa manfaat, kendala, hambatan dan tantangan.

##### 4.2.1 Manfaat Magang

Sejak memulai magang pada bulan September 2021 sampai bulan Februari 2022 banyak sekali manfaat yang dirasakan penulis. Penulis dapat tau implementasi apa yang dipelajari pada masa perkuliahan ke dunia kerja untuk mengatasi permasalahan yang nyata. Menjadi pengalaman yang berharga untuk mengenal dunia kerja atau memilih karir yang tepat nantinya. Magang juga sangat bermanfaat bagi penulis sekaligus juga untuk perusahaan tempat magang, dimana dari sisi penulis magang menjadi sarana untuk belajar mengembangkan *skill*, sedangkan bagi perusahaan sebagai sarana untuk mencari kandidat calon karyawan yang potensial.

Magang sangat bermanfaat untuk meningkatkan *softskill* seperti hidup dalam perbedaan. Pada saat menjalani magang pasti akan bertemu dengan orang yang berbeda-beda, latar belakang yang berbeda-beda, hingga sudut pandang yang berbeda-beda. Dari sini penulis dapat

belajar menerima dan bertoleransi mengenai perbedaan tersebut supaya nantinya terbiasa dengan diskusi dengan sudut pandang yang berbeda dan terbuka dengan ide-ide baru.

Pada saat menjalani magang pasti akan bertemu dengan orang-orang baru, mau tidak mau harus berkenalan dan bersosialisasi. Bagi yang agak canggung bertemu orang yang baru seperti penulis, cukup susah pada awalnya akan tetapi asalkan ingin membuka diri dan memulai pembicaraan maka akan terbiasa nantinya, karena pasti nanti pada saat lulus kuliah dan mulai masuk ke kehidupan nyata kita pasti tidak mungkin hidup sendiri karena manusia saling membutuhkan satu sama lain maka dalam menjalani magang *skill* ini bisa dapat diasah dengan lebih baik lagi.

Magang bermanfaat untuk membangun pengalaman kerja, portofolio, atau resume. Mayoritas orang diterima kerja berdasarkan pengalaman kerjanya, dan pengalaman kerja didapatkan melalui magang di perusahaan tertentu. Hal ini sangat berharga karena menjadi faktor penting atau bahan acuan bagi para pewawancara dalam mencari kandidat untuk direkut, apalagi kinerja pada saat magang sangat bagus maka peluang diterima bekerja akan semakin besar.

Dari segi keilmuan penulis mendapatkan pembelajaran seperti bagaimana penerapan *balanced scorecard* pada perusahaan magang dan bagaimana *balanced scorecard* sangat penting bagi perusahaan untuk mencapai strategisnya. Dan bagaimana sebuah teori *balanced scorecard* diimplementasikan dalam bentuk *dashboard* seperti proyek di mana penulis berkontribusi dalam menjalani proses magang.

Dan juga dari segi keilmuan penulis mempelajari tentang bagaimana *design pattern Model-View-Template* diterapkan dalam proyek. Karena pada saat masa kuliah penulis lebih sering menemui design pattern *Model-View-Controller*. Dari kedua *design pattern* tersebut ternyata tidak memiliki perbedaan yang terlalu signifikan, di mana pada MVT, *view* berperan untuk *logic*, sedangkan pada MVC, yang berperan sebagai *logic* adalah *Controller*.

#### **4.2.2 Kendala, Hambatan, dan Tantangan**

Ketika menjalani proses magang penulis mempunyai beberapa tantangan. Ada beberapa teknologi yang agak asing digunakan, tidak seperti pada waktu kuliah, jadi harus belajar lagi dalam menggunakan teknologi tersebut dengan cepat. Dan juga karena tidak masuk *project* dari awal *project* itu dibuat, penulis harus memahami seluk beluk proyek itu, mulai dari memahami kenapa proyek ini dibuat, memahami setiap fiturnya, memahami seluruh kode yang ditulis oleh *developer* sebelumnya, hingga memahami bagaimana cara pemeliharaan proyek tersebut.

Tantangan yang lain yang dialami oleh penulis cukup klasik yaitu manajemen waktu. Penulis harus menentukan kapan dan apa yang harus dikerjakan pada waktu tertentu, kapan mengerjakan *task* magang, kapan mengerjakan tugas kuliah, dan kapan mengerjakan hal lainnya. Penulis harus membuat skala prioritas mana yang lebih *urgent* agar lebih dahulu dikerjakan. Tantangan ini juga memunculkan tantangan yang baru yaitu kedisiplinan. Karena penulis sudah mengatur jadwal kapan dan apa yang dikerjakan maka supaya tidak mengganggu agenda yang lainnya, pekerjaan yang sudah dijadwalkan harus dikerjakan pada waktu itu juga.

Pada saat menjalani proses magang juga penulis mendapatkan tantangan khususnya pada saat membuat model klasifikasi dengan metode *Support Vector Machine*. Tantangannya adalah pada saat melabeli datanya, data yang harus dilabeli sebanyak seribu lebih. Maka dari itu perlu kesabaran untuk melakukannya agar data yang dihasilkan adalah data yang bagus. Tantangan yang lain yang dihadapi penulis adalah pada saat mengerjakan proyek Project Scoreboard. Karena yang mengerjakan langsung proyek ini adalah penulis, mulai dari tahap analisis, desain, implementasi, evaluasi, hingga *deploy*.

## **BAB V**

### **PENUTUP**

#### **5.1 Kesimpulan**

Setelah menjalani magang selama enam bulan dan berkontribusi ke dalam beberapa proyek dapat disimpulkan bahwa.

- a. Pengembangan halaman detail performa pada *dashboard* timload dapat selesai dengan baik sehingga tujuan dan manfaat laporan ini terpenuhi.
- b. Pengembangan project scoreboard pada timload dapat selesai dengan baik sehingga tujuan dan manfaat laporan ini terpenuhi.
- c. Penggunaan model klasifikasi SVM (*Support Vector Machine*) berhasil dengan didapatkannya akurasi model sebesar 83%.

#### **5.2 Saran**

Adapun juga saran dari penulis setelah menjalani proses magang dan mengerjakan halaman detail performa dan project scoreboard. Untuk mendapatkan hasil yang lebih baik, diperlukan pemeliharaan dan pengembangan selanjutnya. Berikut saran dari penulis.

- a. Dalam penulisan kode perlu dilakukan *refactoring* agar performa dari halman detail performa dan project scoreboard dapat lebih meningkat lagi.
- b. Perlu dilakukan pengoptimasian query khususnya pada project scoreboard, karena pada project scoreboard query-nya lumayan panjang, agar pengaksesan data dapat lebih cepat ke *database*.
- c. Peran SVM pada halaman detail performa masih kurang dimaksimalkan karena hanya menampilkan satu *donut chart* dari akumulasi semua *task*, sehingga pengguna tidak mengetahui *task* apa saja yang masuk dalam kategori yang diklasifikasikan oleh SVM agar dapat dilakukan evaluasi lebih lanjut.

## DAFTAR PUSTAKA

- Delyani, G. (2021, April 4). *Machine Learning Python : Kenali Tentang Algoritma Support Vector Machine, Yuk!* (DQLab) Dipetik Juni 15, 2022, dari <https://www.dqlab.id/kenali-tentang-algoritma-support-vector-machine>
- Haekal, M. M. (2022, Juli 30). *Apa Itu Agile? Pengertian, Prinsip, Metode, dan Kelebihan [Terlengkap]*. Dipetik Juni 18, 2022, dari <https://www.niagahoster.co.id/blog/agile-adalah/>
- Kaplan, R. S., & Norton, D. P. (1992, Februari). *The Balanced Scorecard—Measures that Drive Performance*. (Harvard Business Review) Dipetik juni 15, 2022, dari <https://hbr.org/1992/01/the-balanced-scorecard-measures-that-drive-performance-2>
- Lucid Content Team. (2022). *How to create software design documents*. (Lucidchart) Dipetik Juni 15, 2022, dari <https://www.lucidchart.com/blog/how-to-create-software-design-documents>
- Muhardian, A. (2016, Agustus 24). *Belajar Django #1: Pengenalan Dasar Django untuk Pemula*. Dipetik Juni 15, 2022, dari <https://www.petanikode.com/django-untuk-pemula/>
- Priharto, S. (2021, April 5). *Staging Environment: Pengertian, Manfaat dan Batasannya*. (Aksaragama) Dipetik Juni 15, 2022, dari <https://aksaragama.com/teknologi/staging-environment/>
- PT Javan Cipta Solusi. (2021). *Javan's Hanbook*.
- Sachin. (2021, Juni 7). *Django Tutorial - MVT Pattern on Django (Model View Template)*. (TutorTech) Dipetik Juni 15, 2022, dari <https://rrtutors.com/tutorials/django-mvt-design-pattern>
- samsudiney. (2019, Juli 25). *Penjelasan Sederhana tentang Apa Itu SVM?* (medium.com) Dipetik Juni 18, 2022, dari <https://medium.com/@samsudiney/penjelasan-sederhana-tentang-apa-itu-svm-149fec72bd02>

LAMPIRAN

