

# SKRIPSI

## **PENGEMBANGAN APLIKASI PELAPORAN *PROGRESS-PLAN-PROBLEM* UNTUK MANAJEMEN TUGAS DAN PENENTUAN OKR DI KRAFTHAUS INDONESIA**



Disusun Oleh:

N a m a : Muhammad Pandu Widodo

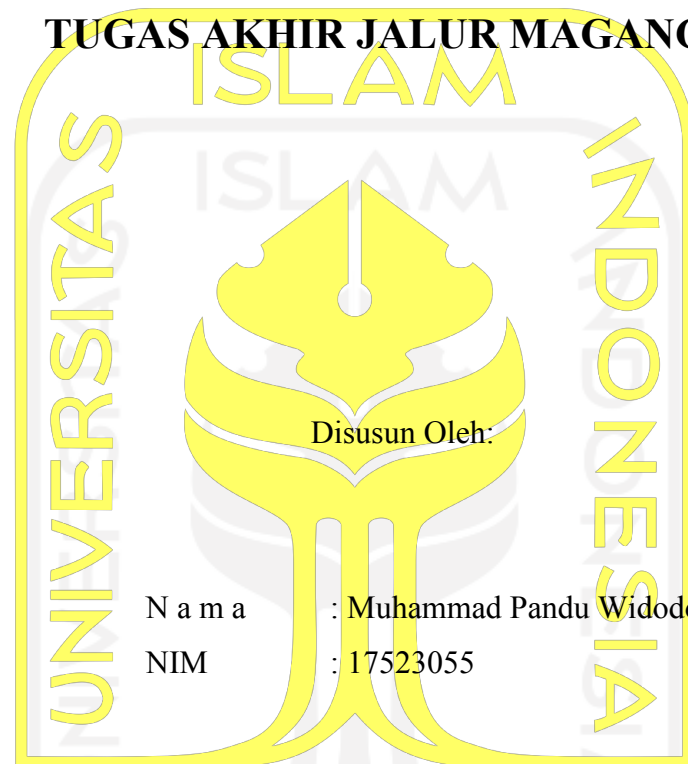
NIM : 17523055

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

**2020**

**PENGEMBANGAN APLIKASI PELAPORAN *PROGRESS-PLAN-PROBLEM* UNTUK MANAJEMEN TUGAS DAN PENENTUAN OKR DI KRAFTHAUS INDONESIA**

**TUGAS AKHIR JALUR MAGANG**



N a m a : Muhammad Pandu Widodo

NIM : 17523055

الجمعة الاستاذة الباندية

Yogyakarta, 27 Desember 2020

Pembimbing,

Andhik Budi Cahyono, S.T., M.T.

**PENGEMBANGAN APLIKASI PELAPORAN *PROGRESS-PLAN-PROBLEM* UNTUK MANAJEMEN TUGAS DAN PENENTUAN OKR DI KRAFTHAUS INDONESIA**

**TUGAS AKHIR JALUR MAGANG**

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

Yogyakarta, 27 Desember 2020

Tim Penguji

**Ketua Penguji**

Andhik Budi Cahyono, S.T., M.T.



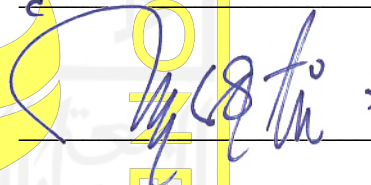
**Anggota 1**

Hendrik, S.T., M.Eng.



**Anggota 2**

Moh. Idris, S.Kom., M.Kom.

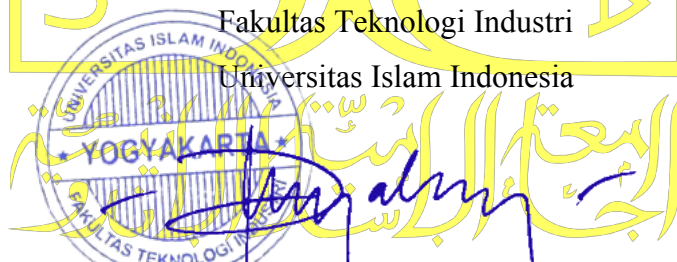


Mengetahui,

Ketua Program Studi Informatika – Program Sarjana

Fakultas Teknologi Industri

Universitas Islam Indonesia



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

Yang bertanda tangan di bawah ini:

Nama : Muhammad Pandu Widodo  
NIM : 17523055

Tugas akhir dengan judul:

**PENGEMBANGAN APLIKASI PELAPORAN *PROGRESS-PLAN-  
PROBLEM* UNTUK MANAJEMEN TUGAS DAN PENENTUAN  
OKR DI KRAFTHAUS INDONESIA**

Menyatakan bahwa seluruh komponen dan isi dalam tugas akhir ini adalah hasil karya saya sendiri. Apabila dikemudian 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 resiko dan konsekuensi apapun.

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

Yogyakarta, 27 Desember 2020



Muhammad Pandu Widodo

## HALAMAN PERSEMBAHAN

*Alhamdulillah Rabbil 'Alamin*, puji syukur kepada Allah atas segala rahmat yang dilimpahkan-Nya serta sholawat serta salam selalu terucap kepada Rasulullah Muhammad. *Alhamdulillah*, setelah menjalani seluruh proses perkuliahan, sampailah kepada laporan tugas akhir yang disusun sebagai penuntasan apa yang telah dimulai. Laporan tugas akhir ini saya persembahkan khususnya kepada keluarga saya, Ayah dan Ibu, yang selalu memberikan dukungan yang tak terhitung dalam perjalanan kuliah saya. Juga kepada seluruh pihak yang menjadi *support system* selama pengerjaan sebagai rekan narasumber pengumpulan data, rekan kerja, hingga rekan konsultasi. Kemudian laporan tugas akhir ini juga saya persembahkan untuk siapa saja yang nantinya akan membaca laporan ini sebagai referensi yang dibutuhkan.



## HALAMAN MOTO

"Di manapun engkau berada selalulah menjadi yg terbaik dan berikan yang terbaik dari yang bisa kita berikan."

Baharuddin Yusuf Habibie

"Jika kamu tidak tahan terhadap penatnya belajar, maka kamu akan menanggung bahayanya kebodohan"

Abu Abdullah Muhammad bin Idris asy-Syafi'i al-Muththalibi al-Qurasyi

*"I mean you just have to put in 80 to 100 hour weeks every week. [This] improves the odds of success. If other people are putting in 40 hour workweeks and you're putting in 100 hour workweeks, then even if you're doing the same thing, you know that you will achieve in four months what it takes them a year to achieve."*

Elon Reeve Musk

"It is better to be a warrior in a garden than to be a gardener in a war"

Bruce Lee

*"I think it is possible for ordinary people to choose to be extraordinary."*

Elon Reeve Musk

## KATA PENGANTAR

### *Assalamu'alaikum Wr. Wb.*

Segala puji dan syukur selalu dihaturkan kepada Allah Subhanahu Wata'ala atas berkat rahmat limpah-Nya yang senantiasa tidak dapat dihitung. Serta shalawat dan salam penulis ucapkan kepada Nabi Muhammad Sallallahu Alaihi Wasallam, yang telah membawa umat Islam dari zaman kebodohan menuju zaman yang terang benderang dengan Islam dan ilmu pengetahuan. Puji syukur yang sebesar-besarnya selalu dihaturkan kepada Allah terutama atas berkah dan izinNya, proses kuliah hingga magang dapat terlaksana dengan baik sebagai media pembelajaran terhadap ilmu dan ciptaan-Nya.

Adapun dari berlangsungnya kegiatan magang di PT Krafthaus Indonesia, kemudian laporan ini disusun untuk memenuhi persyaratan tugas jalur magang di Fakultas Teknologi Industri Jurusan Informatika Universitas Islam Indonesia. Dalam penyusunan laporan magang ini, tidak lepas dari arahan dan bimbingan berbagai pihak. Penulis mengucapkan rasa hormat dan terima kasih kepada semua pihak yang telah membantu. Pihak-pihak yang terkait diantaranya sebagai berikut:

1. Allah Subhanahu Wata'ala yang telah memberikan kehidupan, keselamatan, kesehatan jasmani dan rohani serta kesempatan untuk dapat melaksanakan magang dengan baik,
2. Kedua orang tua tercinta yang selalu mendukung dan memberikan doa, bantuan positif baik secara moral maupun material dalam kegiatan magang ini.
3. Dr. Raden Teduh Dirgahayu, S.T., M.Sc., selaku Ketua Program Studi Teknik Informatika Program Sarjana Fakultas Teknologi Industri Universitas Islam Indonesia
4. Andhik Budi Cahyono, S.T., M.T. selaku Dosen Pembimbing yang telah bersedia membimbing dan mengarahkan penyusunan laporan ini.
5. Bapak Muhammad Arief Widyananda, selaku Managing Director PT Krafthaus Indonesia yang telah menerima penulis dengan baik.
6. Bapak Muhammad Asyari Indraputra selaku ketua tim pengembang Paperlust dan segenap tim pengembang Paperlust di PT Krafthaus Indonesia
7. Bapak Indra Ramdhani selaku Solution Architect PT Krafthaus Indonesia yang membimbing secara teknis dalam tugas-tugas yang penulis kerjakan.
8. Segenap karyawan dan karyawan PT Krafthaus Indonesia yang telah dengan tulus menerima, memberi pengarahannya dan membantu penulis selama melaksanakan magang di perusahaan tersebut.

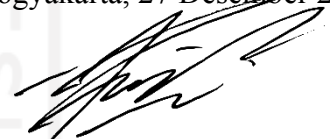
9. Seluruh keluarga besar Jurusan Informatika Universitas Islam Indonsia yang telah memberikan bantuan dan dukungan selama perkuliahan, magang hingga penulisan laporan tugas akhir ini.
10. Semua pihak yang telah membantu dalam penulisan laporan ini sebagai rekan konsultasi, hingga rekan cerita dan berkeluh kesah.

Pengalaman melaksanakan magang di PT Krafthaus Indonesia menjadi pengalaman yang membuka penulis terhadap berbagai konsep dan teknologi baru di bidang informatika khususnya dalam pengembangan aplikasi berbasis web. Pengalaman ini didapat dengan terjun langsung dalam lingkup industri teknologi informasi bersama dengan tim profesional yang senantiasa membimbing penulis untuk mempelajari konsep dan menyusun kode untuk menyelesaikan permasalahan yang ada.

Laporan ini telah dibuat dengan usaha terbaik dari penulis, tetapi masih jauh dari kata sempurna. Sehingga diperlukan saran dan kritik yang membangun dari pembaca untuk penyempurnaan laporan ini. Akhir kata, penulis berharap dari disusunnya laporan ini dapat memberikan manfaat bagi semua pihak.

***Wassalamu'alaikum Wr. Wb.***

Yogyakarta, 27 Desember 2020



Muhammad Pandu Widodo



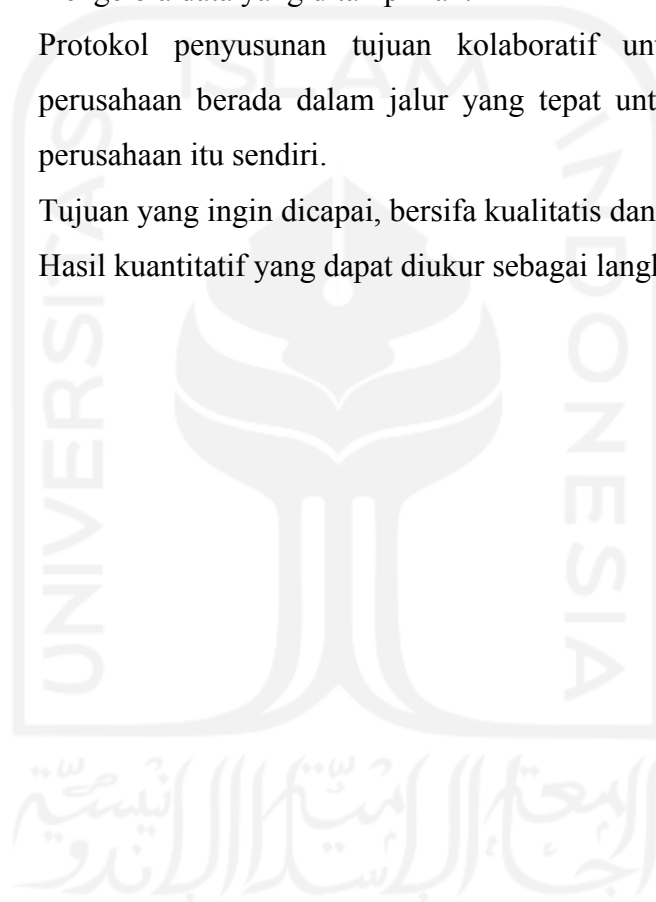
## SARI

Krafthaus Indonesia menerapkan *framework Objective-Key Result* (OKR) untuk membantu penentuan arah dan tujuan strategis perusahaan. Pemantauan keselarasan pencapaian tujuan strategis perusahaan dengan eksekusi pekerjaan teknis harian menjadi tantangan bagi pihak manajemen. Aplikasi pelaporan *Progress-Plan-Problem* (PPP) App dikembangkan sebagai aplikasi internal Krafthaus Indonesia untuk membantu penerapan PPP oleh seluruh karyawan. PPP merupakan teknik manajemen pelaporan status pekerjaan dalam batas waktu tertentu (harian, mingguan atau bulanan). Implementasinya adalah dengan mewajibkan setiap karyawan untuk melaporkan beberapa poin (rentang ideal antara 3 sampai 5 poin) pencapaian, target, dan hambatan dalam kurun waktu pelaporan tertentu. Pengembangan mengikuti alur siklus pengembangan perangkat lunak dan menggunakan MERN Stack sebagai teknologi yang digunakan. MERN Stack menggunakan MongoDB sebagai basis data berbasis JSON, Express.js untuk *back-end* aplikasi, kerangka kerja React.js untuk pengembangan *front-end* aplikasi, dan Node.js *runtime environment*. Setelah dilakukan pengembangan, pengujian dan integrasi dengan alur pelaporan rutin perusahaan, *Progress-Plan-Problem* App dinilai dapat membantu pemantauan tugas karyawan dengan relevansinya terhadap pencapaian tujuan strategis perusahaan atau OKR.

Kata kunci: *Progress-Plan-Problem*, MERN Stack, OKR, Perencanaan.

## GLOSARIUM

MERN-Stack	<i>Framework</i> yang disusun dari komponen-komponen sumber terbuka yang menyediakan kerangka kerja <i>end-to-end</i> untuk pengembangan aplikasi berbasis web yang komprehensif dan memudahkan peramban untuk terhubung dengan basis data.
CRUD	Empat fungsi dasar penyimpanan tetap. Dimiliki oleh hampir seluruh aplikasi pengolahan data untuk memberikan pengguna cara untuk mengelola data yang ditampilkan.
OKR	Protokol penyusunan tujuan kolaboratif untuk memastikan bahwa perusahaan berada dalam jalur yang tepat untuk mencapai tujuan dari perusahaan itu sendiri.
<i>Objective</i>	Tujuan yang ingin dicapai, bersifat kualitatif dan inspirasional.
<i>Key-Result</i>	Hasil kuantitatif yang dapat diukur sebagai langkah pemenuhan <i>objective</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	ix
GLOSARIUM	x
DAFTAR ISI	xi
DAFTAR TABEL	xii
DAFTAR GAMBAR	xiii
BAB I PENDAHULUAN	1
1.1 Latar Belakang	1
1.2 Ruang Lingkup Magang	3
1.3 Tujuan	3
1.4 Manfaat	4
1.5 Sistematika Penulisan	4
BAB II DASAR TEORI	5
2.1 Aplikasi Pelaporan	5
2.2 <i>Progress-Plan-Problem Reporting</i>	5
2.3 <i>Objective-Key Result</i>	6
2.4 MERN Stack	7
2.4.1 MongoDB	7
2.4.2 Express.js	8
2.4.3 React.js	8
2.4.4 Node.js	8
2.5 Trello	9
2.6 Slack	9
2.7 Gitlab	9
BAB III PELAKSANAAN MAGANG	11
3.1 Pengembangan <i>Progress-Plan-Problem App</i>	11
3.1.1 Pendefinisian Proyek	11
3.1.2 Inisialisasi Proyek	11
3.1.3 Perencanaan Proyek	12
3.1.4 Pelaksanaan Proyek	13
3.1.5 Pengendalian, Pengujian dan Evaluasi Proyek	24
3.1.6 Penutupan Proyek	26
BAB IV REFLEKSI PELAKSANAAN MAGANG	27
4.1 Aplikasi Pelaporan <i>Progress-Plan-Problem App</i> Krafthaus Indonesia	27
4.2 Manajemen Tugas dan Penentuan OKR	28
4.3 Manfaat Magang untuk Mahasiswa	29
BAB V KESIMPULAN DAN SARAN	31
5.1 Kesimpulan	31
5.2 Saran	32
DAFTAR PUSTAKA	33
LAMPIRAN	34

**DAFTAR TABEL**

Tabel 3. 1 Pembagian waktu aktivitas magang

12



**DAFTAR GAMBAR**

Gambar 2.1 Ilustrasi metode pelaporan <i>Progress, Plan, Problem</i>	5
Gambar 2.2 Ilustrasi <i>Objective-Key Result</i>	6
Gambar 2.3 Ilustrasi MERN Stack	7
Gambar 3.1 Tangkapan layar Miro Board sebagai <i>platform</i> curah ide kolaboratif	14
Gambar 3.2 Tangkapan layar salah satu kluster basis data di MongoDB Atlas	15
Gambar 3.3 Pengujian <i>end-point back-end</i> dengan Postman	16
Gambar 3.4 Tampilan halaman dasbor utama <i>Progress-Plan-Problem</i> App	17
Gambar 3.5 Tampilan modal pengisian & penyuntingan data <i>Progress-Plan-Problem</i>	18
Gambar 3.6 Halaman Dasbor dengan Redux DevTools di kanan layar	19
Gambar 3.7 Tampilan <i>pop-up</i> modal sunting <i>Progress-Plan-Problem</i>	20
Gambar 3.8 Tampilan halaman komparasi <i>Progress-Plan-Problem</i> antar pekan	21
Gambar 3.9 Halaman Pencarian <i>Progress-Plan-Problem</i>	22
Gambar 3.10 Halaman pelaporan <i>Progress-Plan-Problem</i>	23
Gambar 3.11 Daftar pelaporan <i>bug</i> pada papan Trello pengembangan aplikasi	24
Gambar 3.12 Pemodelan OKR pada tim Account di Paperlust	25
Gambar 3.13 <i>Progress-Plan-Problem</i> salah satu karyawan tim Account Paperlust	26

## **BAB I**

### **PENDAHULUAN**

#### **1.1 Latar Belakang**

Krafthaus Indonesia merupakan sebuah *Digital Agency* yang memberikan layanan *digital marketing* dan pengembangan layanan percetakan *wedding stationery* berbasis website. Layanan *digital marketing* ditangani oleh anak perusahaan bernama Online Business Development (OBD) yang berfokus pada menyediakan *Search Engine Marketing* (SEM) dan *Search Engine Optimization* (SEO) untuk membantu klien meningkatkan jangkauan layanannya. Layanan percetakan *wedding stationery* ditangani oleh Paperlust yang memberikan layanan utama sebagai jembatan antara desainer kreatif dengan pelanggan di seluruh dunia untuk menghasilkan dan menjual berbagai jenis kartu undangan dan *wedding stationery* lainnya.

Melihat dari jumlah pekerjaan yang ditangani oleh Krafthaus Indonesia dalam jangka waktu harian, mingguan, bulanan, dan seterusnya, penggunaan sebuah metodologi manajemen tugas dan pelaporan menjadi hal yang esensial. Tujuannya adalah untuk memudahkan dari sisi manajemen untuk mengalokasikan sumber daya manusia untuk mengerjakan tugas-tugas untuk memenuhi kebutuhan klien. Implementasi manajemen tugas sesuai riset telah memunculkan peningkatan produktivitas dengan mempromosikan standarisasi dan transparansi, memungkinkan untuk pelacakan pekerjaan lampau, peningkatan efektivitas mekanisme pengendalian dan pemantauan, serta memudahkan dalam sinkronisasi dan koordinasi dari berbagai aktivitas dalam lingkup perusahaan (Riss et al., 2005)

Metode pelaporan *Progress, Plan, Problem* (PPP) merupakan teknik manajemen pelaporan status pekerjaan dalam batas waktu tertentu (harian, mingguan atau bulanan). Implementasinya adalah dengan mewajibkan setiap karyawan untuk melaporkan beberapa poin (rentang ideal antara 3 sampai 5 poin) pencapaian, target, dan hambatan dalam kurun waktu pelaporan. Pencapaian yang dimaksud di sini adalah hal apa saja yang telah berhasil dikerjakan dalam kurun waktu pelaporan. Target adalah perencanaan dari hal apa saja yang akan dikerjakan di kurun waktu pelaporan selanjutnya. Hambatan adalah kendala yang ditemui dan membutuhkan bantuan dari rekan kerja lainnya.

Metode pelaporan tugas *Progress, Plan, Problem* (PPP) telah banyak diimplementasikan pada berbagai perusahaan internasional di antaranya: Apple, Facebook, Skype, dan perusahaan percepatan *startup* Seedcamp. Implementasi dari PPP sendiri juga menyesuaikan dengan faktor-faktor unik dari setiap perusahaan yang menggunakannya.

Pada awalnya, Krafthaus Indonesia menggunakan metode perencanaan dan pelaporan tugas dengan menggunakan *spreadsheet* untuk menuliskan perencanaan pekerjaan yang akan datang, kemajuan dari setiap pekerjaan serta masalah yang ditemui dalam pengerjaan. Di antaranya adalah dari perbedaan implementasi PPP dari setiap tim dalam Krafthaus Indonesia yang menjadikan pelaporan oleh tim manajemen sendiri mengalami kesulitan. Juga dari kendala di mana format pelaporan PPP pada *spreadsheet* yang digunakan cenderung sulit untuk dibaca dan dipahami. Dari kendala ini, Krafthaus Indonesia berinisiatif mengembangkan aplikasi *inhouse* untuk memudahkan pelaporan tugas *Progress, Plan, Problem* (PPP).

Berdasarkan uraian yang telah dikemukakan di atas, Krafthaus Indonesia memilih metode pelaporan tugas *Progress, Plan, Problem* (PPP) sebagai metode pelaporan status pekerjaan dalam batas waktu tertentu. Selain itu, dalam implementasi *Progress, Plan, Problem* (PPP) juga membantu dalam pengukuran *Objective Key Result* (OKR) yang merupakan protokol penyusunan tujuan kolaboratif untuk memastikan bahwa perusahaan berada dalam jalur yang tepat untuk mencapai tujuan dari perusahaan itu sendiri (Trinkenreich et al., 2019). Hal ini akan mendukung sisi manajemen sebagai acuan penilaian kinerja setiap individu dalam manajemen sumber daya manusia di Krafthaus Indonesia dan pemantauan pencapaian tujuan strategis organisasi dari Krafthaus Indonesia.

OKR menghubungkan tujuan strategis organisasi dengan langkah-langkah konkrit untuk mencapai tujuan tersebut. OKR memberikan arah pada organisasi, bahwa ada tujuan besar yang ingin dicapai oleh organisasi dalam *timeline* tertentu. Tujuan dan arah perusahaan yang jelas dan terukur, memberikan karyawan kemampuan untuk fokus, memiliki akuntabilitas dan keteraturan dalam organisasi untuk mewujudkan tujuan perusahaan. Secara teknis, hal ini akan mendorong karyawan untuk memberikan hasil pekerjaan yang lebih efektif dan performa lebih tinggi. Hal ini akan mendukung sisi manajemen perusahaan sebagai acuan penilaian kinerja setiap individu dalam manajemen sumber daya manusia di Krafthaus Indonesia dan menjadi jembatan antara penyusunan strategi perusahaan dan eksekusi teknis pekerjaan lapangan. Secara budaya perusahaan, akan mengubah paradigma perusahaan dari berfokus pada *output* pekerjaan seperti berapa artikel terpublikasi atau berapa *item* terjual menjadi fokus pada *outcome* yaitu hasil atau dampak yang terwujud oleh usaha yang dilakukan oleh perusahaan.

Laporan ini mencakup rangkuman aktivitas selama magang yang dilaksanakan di PT Krafthaus Indonesia, pun sebagai bentuk keluaran laporan tugas akhir untuk penjaluran magang periode pertama tahun ajaran 2020/2021.

## 1.2 Ruang Lingkup Magang

Pelaksanaan magang di PT. Krafthaus Indonesia selama kurun waktu 7 bulan dalam periode Maret 2020 hingga September 2020. PT. Krafthaus Indonesia merupakan perusahaan yang membawahi Paperlust, yaitu sebuah wadah pemesanan *wedding stationery* secara daring berbasis *website*, dan Online Business Development (OBD), yaitu layanan *digital marketing* untuk berbagai klien dari Krafthaus Indonesia juga merupakan cabang dari Krafthaus Pty Ltd yang berbasis di Melbourne, Australia. Adapun aktivitas yang dilakukan selama magang adalah sebagai berikut:

- a. Mempelajari MERN (MongoDB, Express.js, React.js, Reduc, Node.js) Stack dan Git, dengan mempelajari tutorial dari platform udemy dibawah bimbingan Solution Architect Krafthaus Indonesia
- b. Masuk ke proyek pengembangan PPP-App, yang merupakan aplikasi *inhouse* dari Krafthaus Indonesia untuk manajemen pelaporan *Progress, Plan, Problem*.
- c. Membuat basis data PPP-App dengan MongoDB sebagai penyimpanan data PPP yang dilaporkan.
- d. Membuat *back-end* PPP-App dengan Express.js sebagai infrastruktur komunikasi data antara *front-end* PPP-App dengan basis data yang telah dibuat.
- e. Membuat *front-end* PPP-App dengan React.js sebagai antarmuka yang digunakan seluruh karyawan Krafthaus Indonesia dalam pelaporan *Progress, Plan, Problem*.
- f. Mengembangkan CRUD PPP-App untuk memasukkan dan mengubahsesuaikan PPP dari setiap karyawan.
- g. Mengembangkan komparasi PPP per-pekan untuk dapat memantau kesesuaian dan perkembangan PPP yang sedang dikerjakan saat ini dengan PPP dari minggu sebelumnya.
- h. Mengembangkan pencarian PPP untuk dapat menemukan PPP dengan kriteria tertentu.
- i. Mengembangkan pelaporan PPP untuk memudahkan pemantauan kesesuaian pekerjaan dari berbagai tim karyawan dalam Krafthaus Indonesia.

## 1.3 Tujuan

Tujuan dari pengembangan aplikasi pelaporan *Progress-Plan-Problem* untuk manajemen tugas dan penentuan OKR di Krafthaus Indonesia adalah sebagai berikut:

- Membuat sebuah aplikasi yang memudahkan karyawan menyusun pelaporan *Progress-Plan-Problem*
- Membantu penerapan metode pelaporan *Progress-Plan-Problem* di Krafthaus Indonesia



- Mengetahui implementasi *Progress-Plan-Problem* di dunia industri.

#### 1.4 Manfaat

Manfaat dari pengembangan aplikasi pelaporan *Progress-Plan-Problem* untuk manajemen tugas dan penentuan OKR Di Krafthaus Indonesia adalah sebagai berikut:

- Aplikasi pelaporan *Progress-Plan-Problem* dapat dalam penggunaannya menyesuaikan dengan lingkungan kerja serta memberikan arahan pekerjaan dalam perusahaan.
- Memudahkan proses perencanaan, pemantauan dan pemecahan masalah dari pekerjaan setiap karyawan.
- Pengembangan tim dan pemantauan OKR oleh manajemen dapat dilakukan dengan lebih baik melalui pelaporan *Progress-Plan-Problem* secara mingguan.

#### 1.5 Sistematika Penulisan

Untuk memahami lebih jelas laporan ini, materi-materi yang tertera pada Laporan ini dikelompokkan menjadi beberapa sub bab dengan sistematika penyampaian sebagai berikut :

ABSTRAK

BAB I PENDAHULUAN

Bab ini berisi tentang latar belakang, ruang lingkup magang, tujuan dan manfaat penelitian, dan sistematika penulisan.

BAB II DASAR TEORI

Bab ini berisi teori yang berupa pengertian dan definisi yang diambil dari kutipan buku yang berkaitan dengan penyusunan laporan serta beberapa *literature review*.

BAB III PELAKSANAAN MAGANG

Bab ini berisi dokumentasi dari kegiatan dan tugas yang dilaksanakan selama menjalani proses magang

BAB IV REFLEKSI PELAKSANAAN MAGANG

Bab ini menjelaskan hasil dari penerapan aplikasi pelaporan *Progress-Plan-Problem* dalam manajemen tugas dan penentuan OKR di Krafthaus Indonesia

BAB V KESIMPULAN DAN SARAN

Bab ini berisi kesimpulan dan saran yang berkaitan berdasarkan yang telah diuraikan pada bab-bab sebelumnya.

DAFTAR PUSTAKA

LAMPIRAN

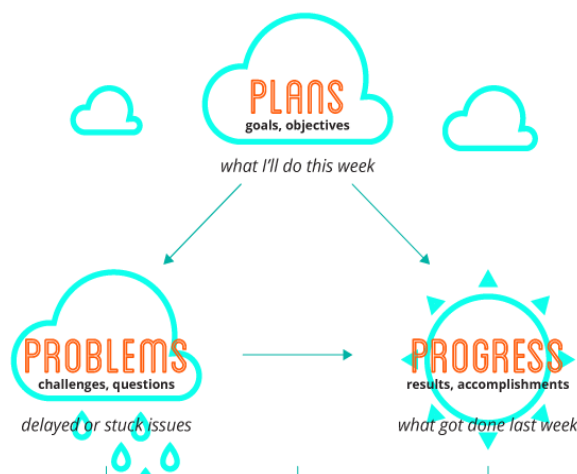
## BAB II DASAR TEORI

### 2.1 Aplikasi Pelaporan

Aplikasi pelaporan adalah aplikasi yang digunakan dalam suatu organisasi yang mempertemukan kebutuhan pengelolaan data harian untuk mendukung operasional organisasi, bersifat manajerial, terikat dengan strategi organisasi dan menyediakan laporan yang diperlukan. Aplikasi ini menjadi media untuk menyampaikan laporan terkait kemajuan kerja, perencanaan serta kendala yang dihadapi oleh pengguna. Kemudian dari setiap laporan yang tersimpan dalam sistem, dapat ditinjau kembali oleh pengguna, baik dari karyawan ataupun dari bagian manajerial untuk kemudian menyusun langkah-langkah yang dibutuhkan untuk menyesuaikan setiap kemajuan kerja dengan tujuan dan strategi organisasi.

### 2.2 *Progress-Plan-Problem Reporting*

Metode pelaporan *Progress, Plan, Problem* (PPP) merupakan teknik manajemen pelaporan status pekerjaan dalam batas waktu tertentu (harian, mingguan atau bulanan). Implementasinya adalah dengan mewajibkan setiap karyawan untuk melaporkan beberapa poin (rentang ideal antara 3 sampai 5 poin) pencapaian, target, dan hambatan yang diilustrasikan pada gambar 2.1 dalam kurun waktu pelaporan (Marovt, 2020). Pencapaian yang dimaksud di sini adalah hal apa saja yang telah berhasil dikerjakan dalam kurun waktu pelaporan. Target adalah perencanaan dari hal apa saja yang akan dikerjakan di kurun waktu pelaporan selanjutnya. Hambatan adalah kendala yang ditemui dan membutuhkan bantuan dari rekan kerja lainnya.

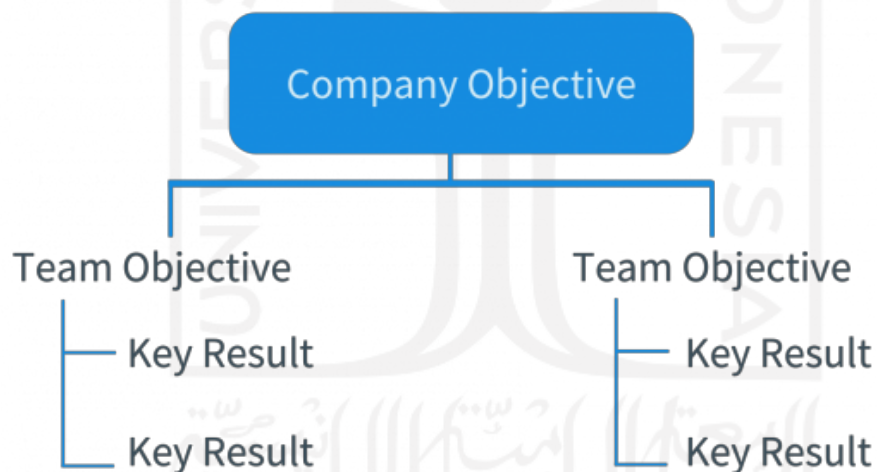


Gambar 2.1 Ilustrasi metode pelaporan Progress, Plan, Problem

Metode ini memberikan keuntungan dengan memudahkan pengelola pekerjaan untuk mengetahui setiap kemajuan yang terjadi, memberikan kesempatan untuk melakukan evaluasi dan perubahan jika dibutuhkan, membuka ruang diskusi terhadap kendala yang menghambat kemajuan pekerjaan dan secara tidak langsung memaksa setiap pihak yang terlibat untuk menyusun jadwal pekerjaan dengan lebih teratur (Anyaeche & Oluwole, 2012).

### 2.3 Objective-Key Result

*Objective-Key Result* (OKR) adalah protokol penentuan tujuan secara kolaboratif untuk membantu memastikan bahwa organisasi secara konsisten fokus dan memprioritaskan usaha untuk mencapai tujuan yang sama dalam organisasi (Doerr, 2018). Metode OKR membantu dalam mendefinisikan dan pemantauan dari tujuan serta hasil yang diharapkan, sehingga metode ini menjadi banyak digunakan di berbagai industri. OKR memiliki dua komponen, yang diilustrasikan dalam gambar 2.2.



Gambar 2.2 Ilustrasi Objective-Key Result

*Objective* yang bersifat kualitatif dan inspirasional, serta *key result* yang bersifat kuantitatif dan dapat diukur. Proses untuk dapat mendefinisikan OKR adalah dengan menentukan *objectives*, kemudian menentukan *key result* untuk setiap *objective*, dan melakukan tindakan untuk mencapai *objective* tersebut, dan diakhiri dengan memberikan masukan secara rutin berkala (Trinkenreich et al., 2019).

## 2.4 MERN Stack

Istilah MERN Stack disusun dari komponen-komponen sumber terbuka yang menyediakan kerangka kerja *end-to-end* untuk pengembangan aplikasi berbasis web yang komprehensif dan memudahkan peramban untuk terhubung dengan basis data (Bhardwaj & Kumar, 2018). Ilustrasi interaksi setiap komponen yang menyusun MERN Stack dapat dilihat pada gambar 2.3 sebagai berikut:



Gambar 2.3 Ilustrasi MERN Stack

### 2.4.1 MongoDB

MongoDB adalah basis data sumber terbuka berbasis dokumen yang juga dapat disebut sebagai basis data noSQL. MongoDB memanfaatkan format dokumen JavaScript Object Notation (JSON) dengan skema untuk manajemen data. Dengan mengingat semakin bertambahnya volume dan variasi data yang lebih mudah dikelola dengan basis data non-relasional. MongoDB mengalami peningkatan yang signifikan dalam penggunaannya di kalangan pengembang karena memenuhi kebutuhan tersebut dengan penyimpanan data berbasis dokumen. Setiap basis data dapat memiliki berbagai koleksi dan dalam setiap koleksi data disimpan dalam bentuk senarai dari objek JSON.

MongoDB memiliki manajemen *cluster* dalam basis datanya. Sebuah *cluster* dalam MongoDB adalah basis data MongoDB yang bekerja untuk satu keperluan dengan mendistribusikan data ke dalam beberapa *server*. Hal ini ditujukan untuk meningkatkan performa pengolahan data karena data diproses oleh beberapa *server* sekaligus atau sering disebut *horizontal scaling*. Sehingga peningkatan kekuatan server tidak perlu dilakukan dengan meningkatkan performa sebuah *server* dengan menggunakan komponen yang memiliki spesifikasi tinggi, namun cukup menambah jumlah *server* seperlunya untuk membagi tugas pengolahan data.

MongoDB menggunakan istilah *collection* dalam pengelolaan basis data. Sebuah *collection* dapat disetarakan dengan sebuah tabel dalam basis data relasional. *Collection* terbatas dalam

sebuah basis data, tidak memiliki skema format yang baku sehingga setiap dokumen yang tersimpan dalam *collection* dapat memiliki kolom yang berbeda.

### 2.4.2 Express.js

Express.js merupakan kerangka kerja aplikasi web berbasis Node.js yang ringkas, fleksibel dan menyediakan fitur fundamental penyusun aplikasi web. Express.js memudahkan pengembangan Application Programming Interfaces (APIs) dengan akses ke *middleware*. *Middleware* berfungsi untuk memberikan akses pada objek *request* dan *response* bersama dengan fungsi selanjutnya yang mana ketika dipanggil, akan melaksanakan *middleware* yang mengikutinya. Secara fungsionalitas, Express.js menyediakan fungsi seperti Spring yang umum digunakan pada aplikasi berbasis Java untuk memudahkan pengembangan kerangka kerja aplikasi web.

Express.js digunakan untuk menyusun *routing* yang merupakan mekanisme akses permintaan HTTP dari komputer klien ke komputer *server* untuk dapat mengarahkan *server* pada kode khusus yang menangani permintaan tersebut. Setiap *route* yang disusun akan memberitahukan *server* apa yang diminta oleh klien, kemudian *server* akan memberikan respon yang sesuai untuk selanjutnya dikirim kembali kepada klien.

### 2.4.3 React.js

React.js adalah *front-end JavaScript library* yang efisien dan fleksibel untuk pengembangan antarmuka pengguna. React.js dikelola oleh Facebook sebagai JavaScript library sumber terbuka yang dapat digunakan untuk membuat *views* yang dijalankan di atas HTML. React.js tidak seperti kerangka kerja utuh yang mengelola seluruh aspek dari pola seperti *Model-View-Controller* (MVC), tetapi hanya *JavaScript library* sehingga hanya menangani penyusunan dan eksekusi *view* dari halaman web,

### 2.4.4 Node.js

Node.js adalah platform untuk menjalankan JavaScript secara asinkron berbasis *event-driven* yang didesain untuk mengembangkan aplikasi jaringan dan APIs. Node.js dibangun menggunakan Chrome V8 Engine yang merupakan *engine Javascript* sumber terbuka yang dikembangkan di atas C++ dan mengimplementasikan *ECMA script*. Node.js menggunakan model *event-driven* asinkron dan *non-blocking I/O model* yang menjadikannya ringan, efisien dan memiliki performa tinggi (Tilkov & Vinoski, 2010). Ekosistem paket aplikasi dalam Node.js yang biasa disebut *node*

*package manager* (npm) memiliki perpustakaan modul sumber terbuka terbesar di dunia yang menyediakan berbagai modul dan *dependencies* untuk pengembangan aplikasi web.

## 2.5 Trello

Trello adalah *platform* kolaborasi yang digunakan oleh berbagai tim di seluruh dunia sebagai tempat koordinasi untuk menyelesaikan sebuah proyek bersama-sama. Satu papan trello akan terdiri dari berbagai daftar *cards* yang biasanya merepresentasikan alur proses pengerjaan atau pembagian tertentu dan *cards* yang biasanya merepresentasikan satu *item* khusus dari hal yang perlu ditinjau atau dilaporkan. Selama pengembangan *Progress-Plan-Problem App*, digunakan daftar *cards* seperti *Project Initialization*, *Idea/Enhancement*, *On-Going Development* serta *Bug Reports*.

Alur pemantauan dengan Trello adalah dengan fitur baru yang belum dikembangkan dituliskan detail dan alur fitur tersebut pada card di dalam daftar *Idea/Enhancement*, kemudian ketika *software developer* mulai mengerjakan fitur tersebut maka card tersebut disalin ke *On-Going Development*. Setelah pengembangan selesai, akan diberikan keterangan pada *cards* yang bersangkutan bahwa fitur tersebut siap untuk dicek. Kemudian dari tim *quality assurance* akan melakukan pengujian dan pengecekan pada fitur tersebut, dan menuliskan laporan permasalahan yang ditemukan dan masukan singkat di daftar *Bug Reports*. Dilanjutkan dengan *software developer* yang akan melakukan perbaikan *bug* tersebut kemudian melaporkannya kembali untuk dicek oleh *quality assurance*.

## 2.6 Slack

Slack adalah aplikasi perpesanan untuk mendukung komunikasi dan koordinasi antar tim. Berbagai perusahaan pengembangan, termasuk juga Krafthaus Indonesia menggunakan Slack salah satunya karena berbagai fitur yang tersedia diantaranya untuk mengelola proyek, berbagi berkas, dan dukungan *chatbot* yang dapat dikustomisasi sesuai kebutuhan tim. Selain itu juga untuk memfokuskan koordinasi pekerjaan melalui Slack yang dibatasi penggunaannya hanya pada lingkup satu perusahaan saja untuk mencegah distraksi dari kepentingan lain di luar pekerjaan.

## 2.7 Gitlab

Gitlab adalah *platform* DevOps yang mencakup seluruh *development lifecycles* dalam satu aplikasi. Sehingga dengan menggunakan Gitlab, dapat dilakukan pengelolaan versi dari kode aplikasi dengan *versioning* berbasis git, dan langsung dilanjutkan dengan *deployment* dan integrasi

ke *server* tanpa memerlukan kolaborasi berbagai aplikasi yang terpisah. Git sebagai *distributed version control system* sumber terbuka telah digunakan di berbagai proyek untuk mengelola *versioning* dari aplikasi yang dikembangkan. Pengembangan *Progress-Plan-Problem App* memiliki alur pengembangan dengan membuat *branch* baru untuk setiap fitur yang dikembangkan, kemudian ketika fitur tersebut telah siap, maka akan diintegrasikan dengan *branch* utama untuk kemudian dilakukan *deployment* pada server.

## 2.8 Postman

Postman adalah aplikasi yang digunakan untuk melakukan verifikasi dari *Application Programming Interface* (API) sebuah proyek. Postman menyediakan antarmuka grafis untuk menyusun *request* dan membaca *responses* yang disusun dalam pengembangan *back-end* proyek untuk memastikan setiap *endpoint* API bekerja sesuai yang diinginkan. Dalam Postman, *request* dapat disusun untuk memiliki setiap parameter dan komponen yang diinginkan untuk mensimulasikan penggunaan *endpoint* seolah-olah dalam lingkup aplikasi sesungguhnya.

## BAB III PELAKSANAAN MAGANG

### 3.1 Pengembangan *Progress-Plan-Problem* App

#### 3.1.1 Pendefinisian Proyek

*Progress-Plan-Problem* App merupakan aplikasi web yang akan digunakan untuk karyawan internal Krafthaus Indonesia untuk pelaporan *Progress-Plan-Problem* mingguan yang selama ini menggunakan *spreadsheet* atau Google Form yang dinilai kurang efektif dan *user-friendly*. *Progress-Plan-Problem* (PPP) sendiri adalah konsep teknik manajemen untuk pelaporan status pekerjaan dalam jangka waktu tertentu secara berulang. Secara sederhana, dapat digambarkan seperti dengan seorang karyawan melaporkan 3 sampai 5 poin pencapaian, tujuan, dan kendala yang dihadapi dalam satu periode pelaporan, seperti harian atau mingguan. Dengan harapan pelaporan ini dapat memberikan gambaran tentang apa yang sedang terjadi dan dilakukan oleh setiap anggota dalam tim (Marovt, 2020).

#### 3.1.2 Inisialisasi Proyek

Bersama dengan kuantitas dan skala pekerjaan yang semakin beragam, Krafthaus Indonesia melakukan perbaikan dan pengembangan infrastruktur pendukung untuk memastikan setiap karyawan dari Krafthaus Indonesia bekerja dengan baik sesuai dengan tujuan yang ingin dicapai oleh perusahaan. Proses pengembangan *Progress-Plan-Problem* App terbagi dalam sejumlah peran yaitu:

- *Project Manager*, bertanggung jawab untuk memantau, mengendalikan dan mengelola produk secara keseluruhan. Juga sebagai otoritas tertinggi untuk melakukan konsultasi apabila ada kendala yang ditemui selama proses pengembangan aplikasi.
- *Quality Assurance*, bertanggung jawab untuk memastikan implementasi solusi yang telah dikerjakan oleh *software developer* berjalan sesuai dengan kesepakatan yang telah dibuat sebelumnya termasuk seperti *flow* dari solusi yang dikembangkan, mengidentifikasi *error*, *bug*, serta masalah yang mungkin muncul dalam penggunaan aplikasi oleh *end-user*.
- *Software Developer*, bertanggung jawab atas eksekusi dan implementasi rancangan kebutuhan dari aplikasi dalam bentuk kode program. Menyusun algoritma dan



penggunaan teknologi yang sesuai untuk dapat mewujudkan solusi yang sebelumnya telah dirancang.

### 3.1.3 Perencanaan Proyek

*Progress-Plan-Problem* App sebagai aplikasi pelaporan pekerjaan yang digunakan oleh seluruh karyawan Krafthaus Indonesia secara setara, sehingga tidak ada perbedaan *user role* dalam penggunaan aplikasi ini. Setiap pengguna dengan aplikasi ini dapat mengisikan laporan *Progress-Plan-Problem* masing-masing, melihat laporan dari karyawan lainnya melalui fitur pencarian, membandingkan laporan saat ini dengan laporan pekan sebelumnya, dan membandingkan laporan beberapa orang sekaligus. Setiap pengguna diwajibkan *log-in* dengan menggunakan sistem autentikasi Panel Krafthaus yang telah dikembangkan sebelumnya, sehingga termasuk dalam rencana pengembangan adalah integrasi dengan sistem Panel Krafthaus yang sudah ada.

Selama magang, pembagian waktu selama pengembangan *Progress-Plan-Problem* App dapat dilihat pada tabel 3.1. Pembagian tersebut belum termasuk beberapa aktivitas yang sering saling bersinggungan dalam masa pengerjaannya. Ketika aktivitas yang telah dilakukan sebelumnya mendapat umpan balik dari tim *Quality Assurance*, maka perbaikan yang perlu dilakukan atas umpan balik tersebut akan menambah waktu pengerjaan.

Tabel 3. 2 Pembagian waktu aktivitas magang

No	Aktivitas	Durasi Waktu
1	Diskusi <i>requirement</i> dari <i>Progress-Plan-Problem</i> App	1 Pekan
2	Mengembangkan basis data <i>Progress-Plan-Problem</i> App dengan MongoDB	1 Pekan
3	Mengembangkan kerangka <i>back-end</i> <i>Progress-Plan-Problem</i> App dengan Express.js	2 Pekan
4	Mengembangkan kerangka <i>front-end</i> <i>Progress-Plan-Problem</i> App dengan React.js	1 Pekan
5	Membuat Redux Store untuk manajemen data pada <i>front-end</i> <i>Progress-Plan-Problem</i> App	1 Pekan
6	Mengembangkan fitur CRUD <i>Progress-Plan-Problem</i> App	1 Bulan
7	Mengembangkan fitur komparasi <i>Progress-Plan-Problem</i> per-pekan	1 Bulan

8	Mengembangkan fitur pencarian <i>Progress-Plan-Problem</i>	2 Pekan
9	Mengembangkan fitur pelaporan <i>Progress-Plan-Problem</i> App	2 Bulan

### 3.1.4 Pelaksanaan Proyek

#### Diskusi *requirement* dari *Progress-Plan-Problem* App

*Progress-Plan-Problem* App sebagai aplikasi internal melibatkan rekan-rekan pengembang dan *User Experience Analyst* dari berbagai tim yang ada dalam Krafthaus Indonesia. Diskusi mengenai problem statement, ideation, hingga perancangan solusi sempat dilakukan secara tatap muka 1 kali. Dalam diskusi pembahasan *requirement* ini, setiap dari karyawan Krafthaus Indonesia yang hadir terlibat aktif untuk saling mengemukakan ide dan pendapatnya. Beberapa karyawan menyampaikan kendala yang dihadapi dengan sistem pelaporan *Progress-Plan-Problem* yang digunakan sebelumnya, yang didominasi oleh tidak seragamnya format penulisan dari setiap *item* pelaporan pada berbagai tim yang ada dalam Krafthaus Indonesia.

Dengan adanya kebijakan untuk bekerja dari rumah selama pandemi Covid-19 maka diskusi dialihkan menggunakan platform daring. Menjadi sebuah tantangan yang cukup berat dengan berubahnya *flow* dan pola kerja yang sebelumnya bersama di studio pengembangan Krafthaus Indonesia yang kemudian harus berpindah ke rumah masing-masing karyawan dengan kolaborasi daring. Platform Miro yang ditunjukkan pada gambar 3.1, digunakan sebagai papan kolaborasi dan curah ide dalam waktu nyata dengan bantuan canvas tak terbatas, *widget*, *sticky notes*, dan *template* lainnya sehingga seolah-olah setiap anggota diskusi berada dalam satu ruangan yang sama dengan papan kolaborasi yang sama. Juga digunakan Google Meet *video conference* untuk komunikasi suara dalam sesi diskusi.



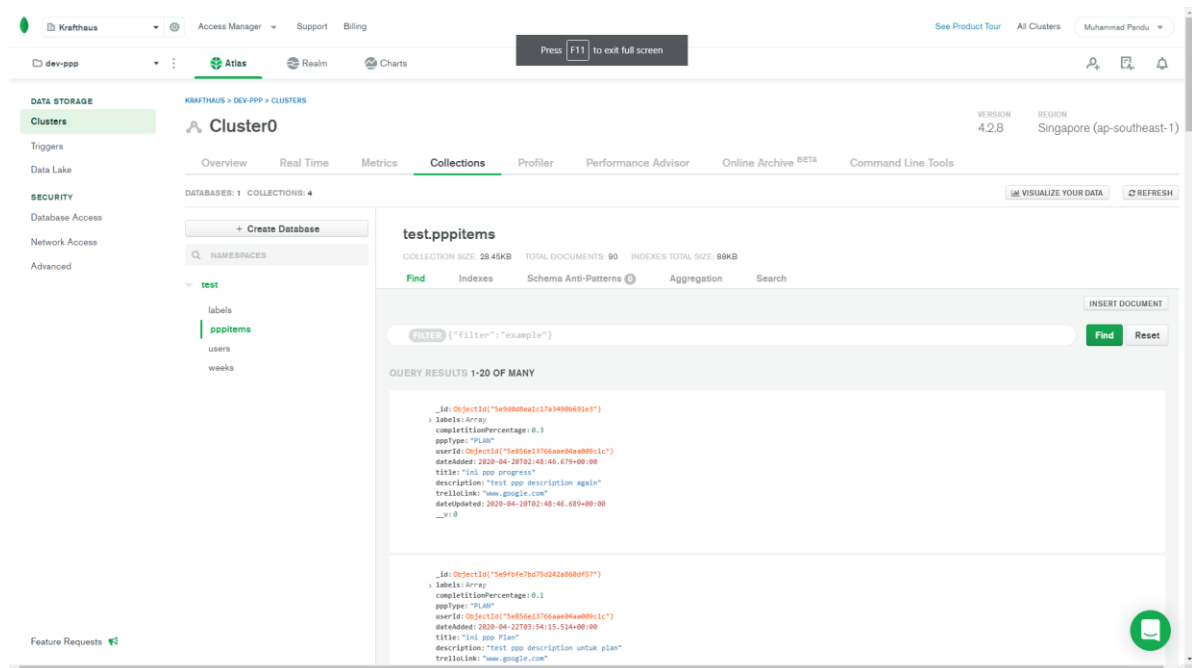
Gambar 3.1 Tangkapan layar Miro Board sebagai *platform* curah ide kolaboratif

Dalam diskusi *requirement*, diambil banyak keputusan penting terkait dengan fitur-fitur yang akan dikembangkan dari *Progress-Plan-Problem* App yang diharapkan mampu mengatasi permasalahan pelaporan *Progress-Plan-Problem* yang ada sebelumnya. Beberapa keputusan penting yang diambil seperti data apa saja yang harus ada dalam setiap laporan, alur penggunaan aplikasi, serta batasan-batasan lainnya.

### Mengembangkan basis data *Progress-Plan-Problem* App dengan MongoDB

Dari diskusi *requirement* *Progress-Plan-Problem* App sebelumnya, dari *Solution Architect* Krafthaus Indonesia memilih untuk menggunakan MongoDB sebagai basis data untuk menyimpan data yang akan digunakan. MongoDB menggunakan basis data dengan format JSON yang dioptimalkan untuk digunakan dalam RestAPI. Manajemen dari MongoDB dapat diakses dengan mudah melalui cloud dengan MongoDB Atlas. Untuk *Progress-Plan-Problem* App, dibuat 1 *cluster* basis data dengan 3 *collection* untuk menyimpan data *Progress-Plan-Problem*, data label/kategori, dan data pekan. Data pengguna diambil dari *cluster* lain yang telah dibuat sebelumnya untuk aplikasi internal Krafthaus Indonesia yang lain.

Pada gambar 3.2 ditunjukkan salah satu *cluster* basis data MongoDB yang digunakan untuk menyimpan *schema* dan *collection* data yang digunakan.



Gambar 3.2 Tangkapan layar salah satu *cluster* basis data di MongoDB Atlas

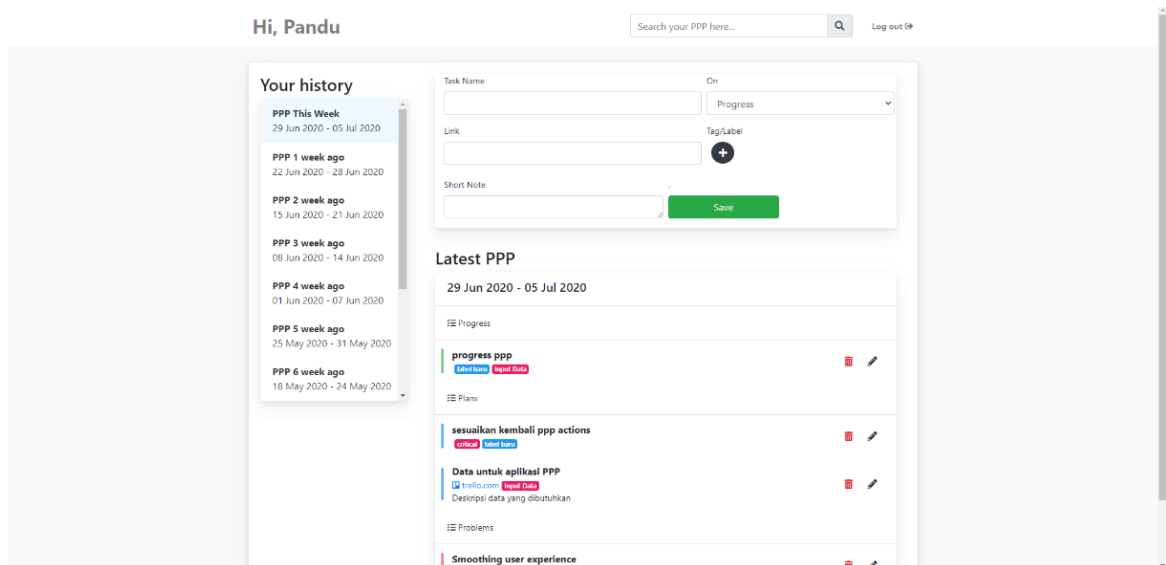
### Mengembangkan kerangka *back-end Progress-Plan-Problem App* dengan Express.js

Express.js merupakan *framework* aplikasi web berbasis Node.js digunakan untuk pengembangan *Progress-Plan-Problem App*. Express.js digunakan untuk *routing* data, akses HTTP, dan pembuatan *middleware*. Untuk menguji *routes* yang telah dibuat, digunakan aplikasi Postman untuk mensimulasikan berbagai *request* yang akan digunakan. Secara umum *back-end* dikategorisasi untuk memudahkan manajemen *routes* untuk membuat data, *routes* untuk mengambil data, dan *routes* untuk melakukan pencarian dengan *query*.

*Request* yang telah disusun, dikirim ke salah satu *endpoint back-end* yang ingin diuji dengan mengisi *endpoint* yang dimaksud pada kolom URL yang ada. Postman akan menerima *reply* dari *server* atas request yang dikirimkan sebelumnya, dan menampilkan isi dari *reply* tersebut di jendela di bawah konfigurasi *request*. Alur tersebut ditunjukkan pada gambar 3.3. Setiap *request* yang telah dibuat, dapat disimpan untuk bisa diuji di lain waktu. Postman juga mendukung untuk berbagi folder *request* dengan pengembang lain untuk saling berkolaborasi melakukan pengujian *endpoint* dari *back-end*.

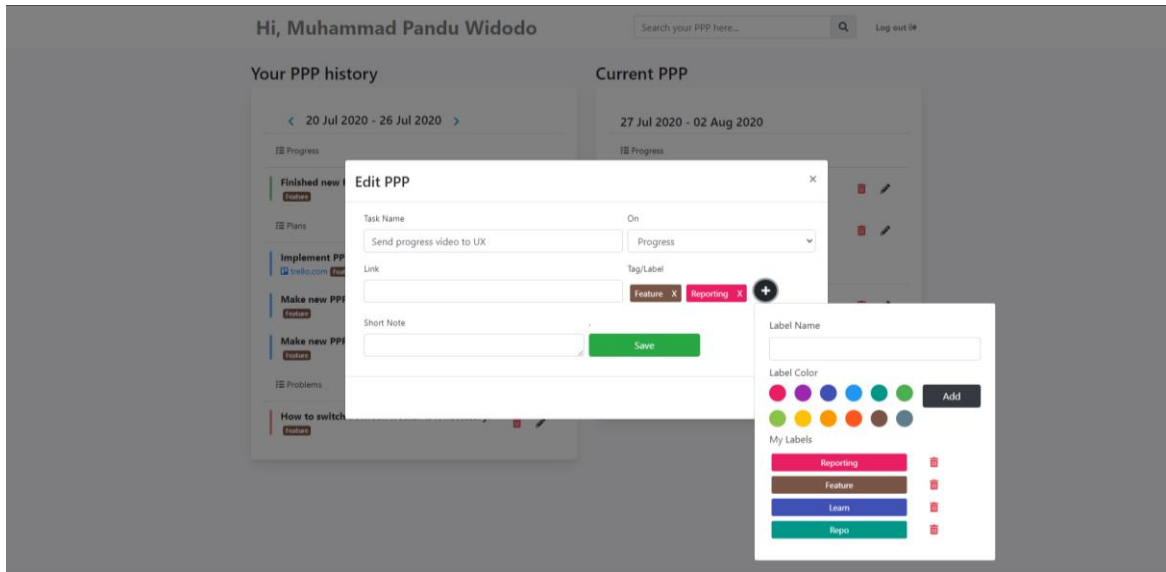


*Front-end Progress-Plan-Problem App* terdiri dari halaman dasbor utama yang ditunjukkan pada gambar 3.4, halaman pencarian, halaman komparasi per pekan, serta halaman pelaporan. Untuk dapat mencapai halaman dasbor, sebelumnya pengguna diperlukan untuk mengakses halaman dasbor dari Panel Krafthaus terlebih dahulu, yang isinya merupakan tautan ke berbagai aplikasi internal milik Krafthaus Indonesia yang dikembangkan sesuai kebutuhan perusahaan untuk mendukung kinerja karyawannya.



Gambar 3.4 Tampilan halaman dasbor utama *Progress-Plan-Problem App*

Dalam gambar 3.5 ditampilkan *pop-up modal* yang digunakan untuk menyunting data *Progress-Plan-Problem* yang ada. Terkhusus pada masukan dan sunting *Progress-Plan-Problem*, ada kolom khusus untuk memasukkan label. Label ini juga memiliki form sendiri yang berada di dalam form *Progress-Plan-Problem* untuk menambahkan label baru beserta kustomisasi warna label, dilanjutkan dengan memilih label dari daftar yang ada untuk ditambahkan ke setiap item *Progress-Plan-Problem*. Label menjadi fitur yang penting karena digunakan untuk mengkategorisasikan item *Progress-Plan-Problem* sesuai yang dikehendaki pengguna, juga dengan label tim yang digunakan bersama dalam satu tim seperti tim *Software Developer* untuk mengkoordinasikan *item* pelaporan yang terkait dari beberapa orang dalam satu tim tersebut.



Gambar 3.5 Tampilan modal pengisian & penyuntingan data *Progress-Plan-Problem*

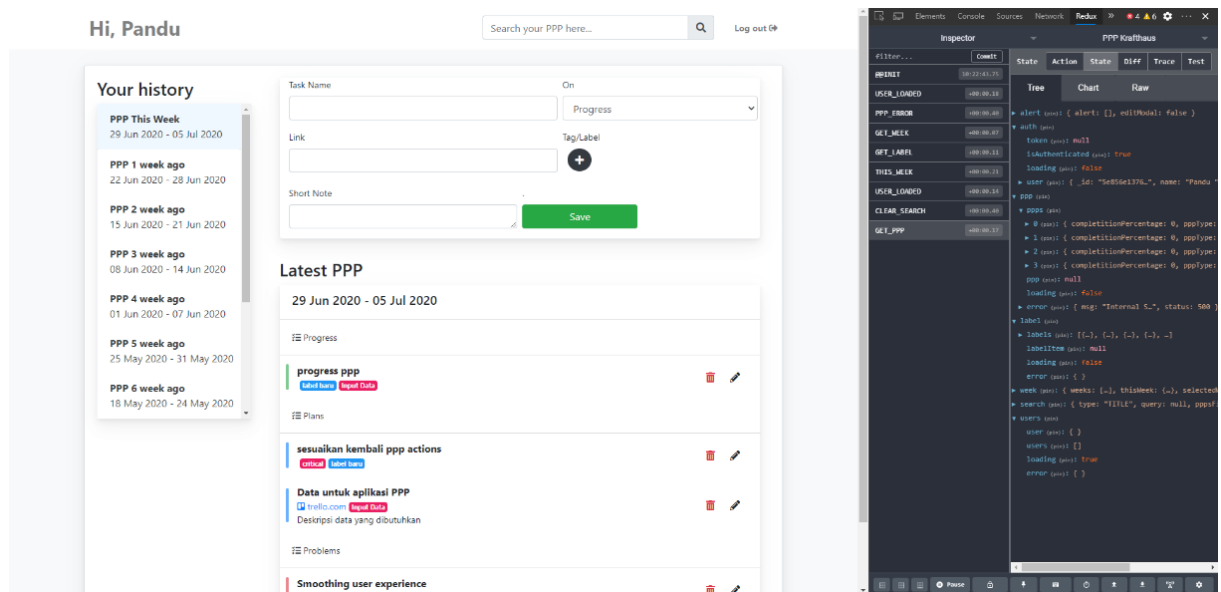
Setiap perubahan data yang terjadi pada aplikasi *Progress-Plan-Problem* App tidak memerlukan *reload* halaman karena antarmuka yang disusun dengan *framework* React.js. Sehingga setiap perubahan data, dari penambahan *Progress-Plan-Problem* baru, hasil sunting *Progress-Plan-Problem*, modifikasi parameter pencarian, dan yang lainnya ditampilkan secara *realtime* tanpa tampilan proses *loading*.

### **Membuat Redux Store untuk manajemen data pada *front-end* *Progress-Plan-Problem* App**

Redux Store adalah metode manajemen data yang digunakan oleh kerangka kerja React.js. Redux memudahkan pemantauan data dan variabel yang tersedia untuk dapat digunakan dalam antarmuka halaman yang disusun dengan React.js. Redux dari awal pembuatannya memang didesain untuk digunakan dengan komponen-komponen dalam React.js. Redux memudahkan untuk mengambil nilai-nilai variabel untuk digunakan dalam komponen sebagai *property*.

Setiap data variabel yang akan ditampilkan dalam tampilan antarmuka *Progress-Plan-Problem* App yang disusun dengan React.js, dikelola dalam sebuah Redux Store. Redux Store menjadi sebuah wadah yang digunakan untuk menyimpan berbagai data yang sebelumnya telah didefinisikan format dan jenis-jenisnya dengan membuat *reducers* terlebih dulu. Data dalam Redux Store dapat diakses oleh setiap komponen yang ada dalam antarmuka React.js. Dalam gambar 3.6 ditampilkan halaman dasbor dengan Redux DevTools di sebelah kanan layar untuk dapat memantau setiap perubahan data yang terjadi dalam Redux Store. Redux DevTools membantu pengembang untuk dapat melihat rincian dari setiap data yang ada dalam Redux Store.

Seperti pada gambar berikut juga terlihat data seperti ppp, user, week dan lain sebagainya. Juga ditampilkan *dispatch* yang sebelumnya telah dikirimkan.



Gambar 3.6 Halaman Dasbor dengan Redux DevTools di kanan layar

Data dalam Redux Store berubah setiap dilakukan *dispatch* untuk melakukan perubahan pada isi dari Redux Store yang dikirim oleh fungsi atau metode yang telah disusun. *Dispatch* biasa dikirimkan setelah melakukan *request* dan menerima data dari *server*, atau setelah melakukan perubahan pada komponen dalam aplikasi React.js. *Dispatch* akan menempatkan data yang diterima ke dalam posisi yang sesuai dalam Redux Store sesuai dengan yang didefinisikan pada *reducers*.

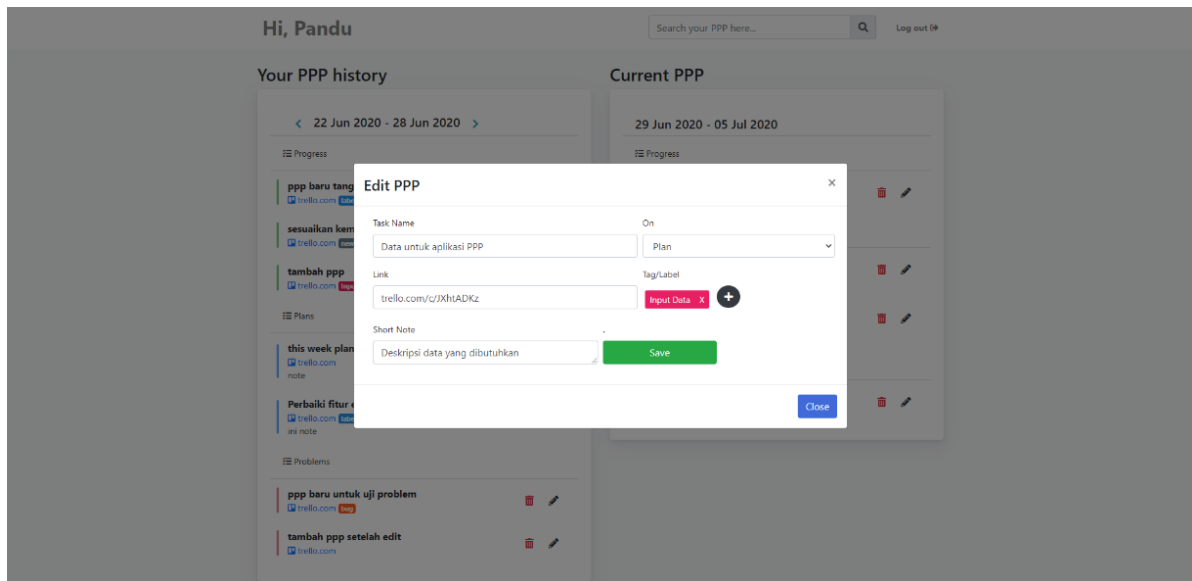
### Mengembangkan fitur CRUD *Progress-Plan-Problem* App

*Create, Read, Update, dan Delete* (CRUD) adalah 4 operasi dasar yang penting untuk dimiliki oleh hampir seluruh aplikasi web untuk memberikan pengguna cara untuk mengelola data yang ditampilkan pada web tersebut. Pada *Progress-Plan-Problem* App, CRUD yang dibuat adalah untuk data *Progress-Plan-Problem*, data label, serta data pekan. CRUD dibuat agar pengguna dapat menambahkan data baru, melihat, mengubah, atau menghapus data yang sudah ada, sehingga aplikasi dapat memenuhi fungsinya dan memecahkan masalah sesuai perancangan aplikasi tersebut.

Untuk transaksi data *Progress-Plan-Problem* yang merupakan data utama dalam aplikasi *Progress-Plan-Problem* App, digunakan form input di halaman dasbor seperti ditampilkan pada gambar 3.5 sebelumnya. Data *Progress-Plan-Problem* yang akan disimpan dalam MongoDB,



sebelum dikirim terlebih dulu diberikan *unique identifier* pada kolom `_id` sebagai *primary key* dengan menggunakan generator *unique identifier* UUID. Kemudian untuk melakukan penyuntingan data dilakukan dengan terlebih dahulu menekan icon pensil yang bermakna sunting pada sebelah kanan setiap *item Progress-Plan-Problem*. Kemudian untuk menghapus *item Progress-Plan-Problem* digunakan tombol tempat sampah yang bermakna hapus di sebelah kanan setiap *item Progress-Plan-Problem* juga.



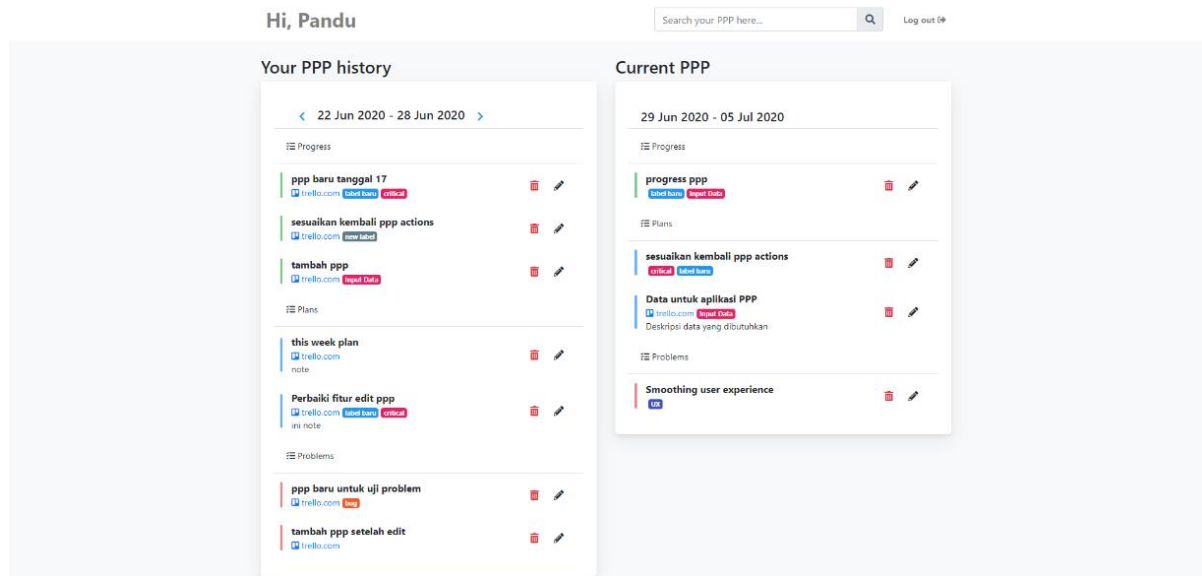
Gambar 3.7 Tampilan *pop-up modal* sunting *Progress-Plan-Problem*

Hasil isian form *Progress-Plan-Problem* baru akan langsung ditampilkan dalam daftar PPP pekan saat ini yang ada di bawahnya, sesuai dengan kategori *plan*, *progress* atau *problem* yang dipilih. *Progress-Plan-Problem* baru akan disimpan dalam basis data MongoDB sesuai dengan parameter yang dimasukkan dalam form. Dan Hasil dari perubahan data *Progress-Plan-Problem* setelah proses *editing* yang dilakukan dengan form *edit* pada gambar 3.7 juga langsung *ter-update* pada daftar *Progress-Plan-Problem* yang ada saat itu. Setiap proses ini terjadi tanpa perlu melakukan *reload* halaman karena menggunakan antarmuka berbasis Javascript dengan React.js.

### Mengembangkan fitur komparasi *Progress-Plan-Problem* per-pekan

Karena pelaporan *Progress-Plan-Problem* menjadi agenda pelaporan mingguan, dibutuhkan fitur untuk dapat melakukan komparasi *Progress-Plan-Problem* antar pekan. Hal ini dilakukan untuk dapat menemukan poin *Progress-Plan-Problem* yang mana yang sekiranya memiliki kemajuan atau kendala, sehingga pemetaan masalah dapat dilakukan dengan lebih mudah. Dengan membandingkan data PPP dari pekan saat ini dengan pekan-pekan sebelumnya,

untuk memantau apakah *Plan* dari suatu pekan sudah menjadi *Progress* di pekan setelahnya, atau *Problem* pada satu pekan mendapatkan *Progress* menuju pemecahannya di pekan setelahnya.



Gambar 3.8 Tampilan halaman komparasi *Progress-Plan-Problem* antar pekan

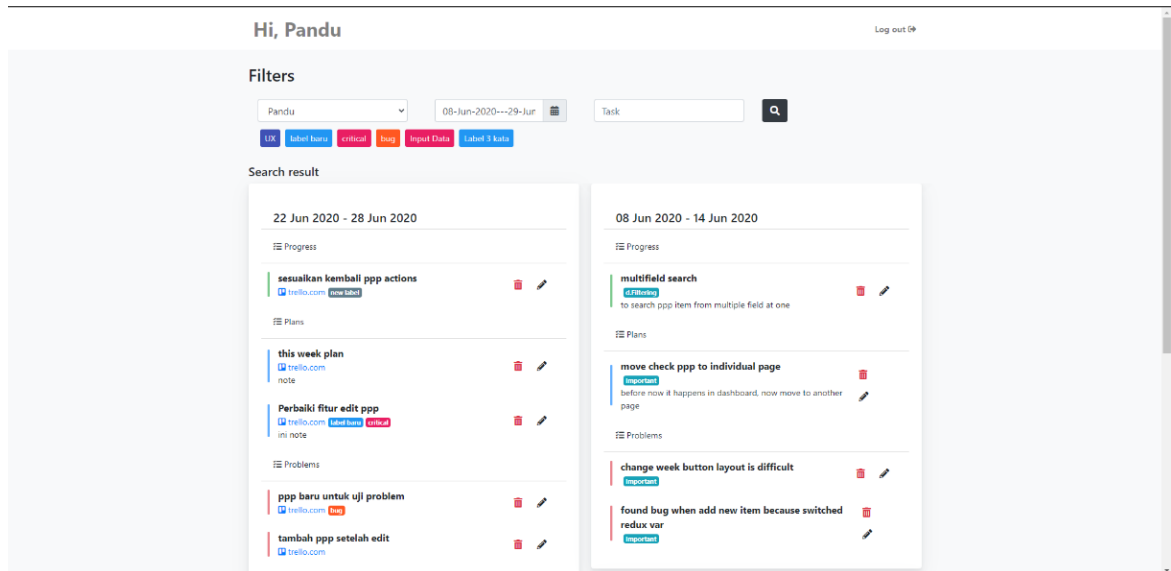
Halaman komparasi *Progress-Plan-Problem* pada gambar 3.8 menampilkan *Progress-Plan-Problem* pada pekan saat ini di sebelah kanan, dan *Progress-Plan-Problem* pada pekan sebelumnya di sebelah kiri. *Progress-Plan-Problem* pada pekan sebelumnya, dapat diganti ke 1 pekan sebelum atau setelahnya dengan menekan tombol panah yang ada di samping tanggal *Progress-Plan-Problem* tersebut. Dengan memanfaatkan Redux Store yang ada dan antarmuka berbasis Javascript, pergeseran antar pekan yang ingin dikomparasikan tidak memerlukan *reload* halaman dan transisi terjadi dengan cepat langsung pada halaman tersebut.

### Mengembangkan fitur pencarian *Progress-Plan-Problem*

Seiring semakin banyaknya data yang dikelola oleh aplikasi, maka metode pencarian menjadi sangat penting. Untuk dapat menemukan data spesifik yang diperlukan dengan memasukkan kata kunci tertentu pada *form* yang telah disediakan. Pencarian *Progress-Plan-Problem* dalam aplikasi ini memiliki empat parameter, yaitu nama pemilik *Progress-Plan-Problem*, judul, tanggal, dan label. Keempat parameter tersebut dapat digunakan secara terpisah atau bersamaan satu dengan lainnya.

Dengan banyaknya data *Progress-Plan-Problem* yang akan dibuat, serta tampilan yang ada untuk dapat fokus ke *Progress-Plan-Problem* dalam satu pekan saja atau komparasi yang hanya antar 2 pekan, maka diperlukan cara untuk dapat menelusuri data item *Progress-Plan-Problem*

yang spesifik berdasarkan kriteria tertentu. Pencarian dalam *Progress-Plan-Problem* App dikembangkan dalam satu halaman tersendiri untuk melakukan pencarian berdasarkan berbagai kriteria yang dibutuhkan. Adapun tampilan dari halaman pencarian dapat dilihat pada gambar 3.9.



Gambar 3.9 Halaman Pencarian *Progress-Plan-Problem*

Di bagian atas ditampilkan form berisi kolom masukan yang menjadi kriteria pencarian *Progress-Plan-Problem*. Kolom masukan terdiri dari masukan nama, *range* tanggal, judul, serta label. Nama digunakan ketika kita ingin menelusuri data *Progress-Plan-Problem* milik karyawan lainnya. *Range* tanggal digunakan untuk melakukan *filter* data *Progress-Plan-Problem* berdasarkan tanggal dibuatnya data *Progress-Plan-Problem* tersebut seperti antara tanggal 6 Juli hingga 17 Juli. Judul digunakan untuk menelusuri *Progress-Plan-Problem* berdasarkan judul yang dituliskan ketika membuat data *Progress-Plan-Problem* tersebut. Label digunakan untuk menelusuri *Progress-Plan-Problem* berdasarkan label yang disematkan pada setiap *item Progress-Plan-Problem*.

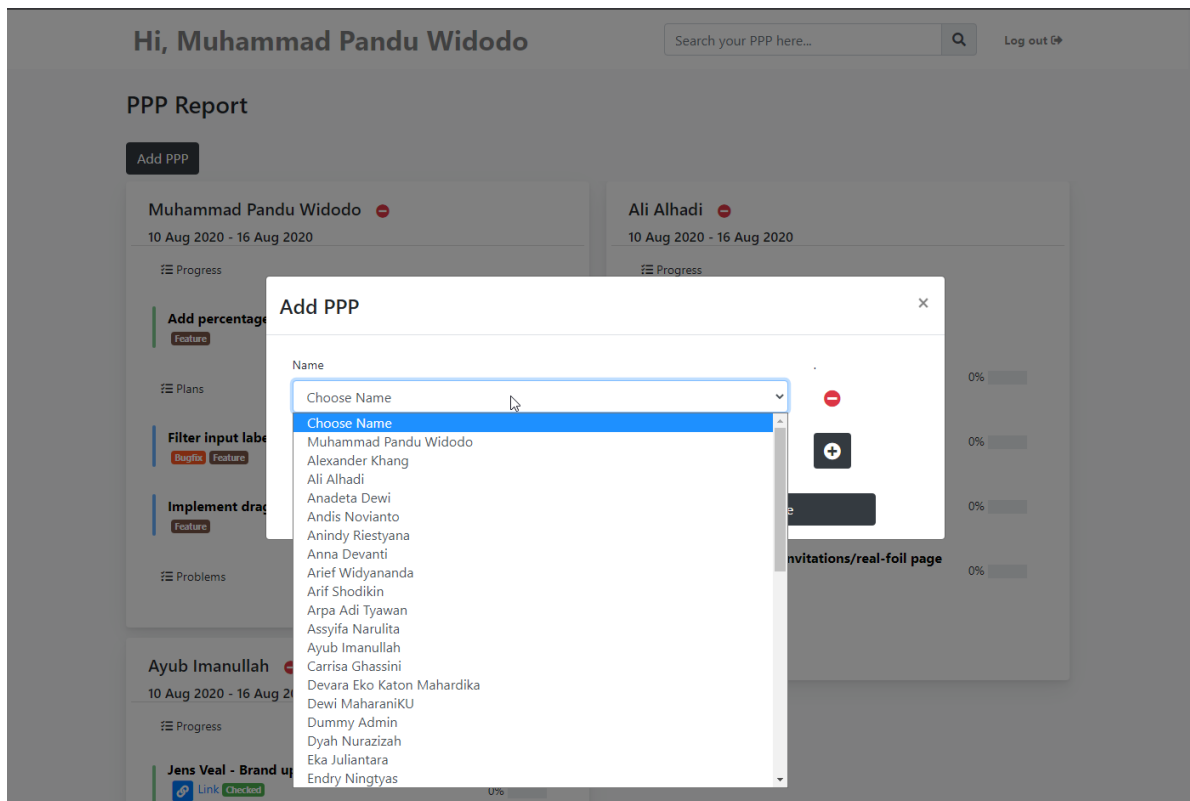
Kolom masukan pencarian dapat digunakan hanya satu kolom saja atau digunakan beberapa kolom bersamaan. Setelah mengisi kriteria pada kolom masukan kemudian klik tombol cari untuk melakukan penelusuran data *Progress-Plan-Problem*. Hasil pencarian akan ditampilkan di bawah form kriteria dengan tetap terbagi menjadi *Progress-Plan-Problem* setiap pekan.

### Mengembangkan fitur pelaporan *Progress-Plan-Problem* App

Dengan pertimbangan bahwa pelaporan *Progress-Plan-Problem* setiap karyawan Krafthaus Indonesia akan ada kaitannya dengan karyawan lainnya baik dalam satu tim pengembang maupun

dari tim lainnya sebagai bentuk kolaborasi dalam pekerjaan, maka dibutuhkan tampilan khusus untuk dapat membandingkan laporan *Progress-Plan-Problem* dari beberapa orang sekaligus.

Pada gambar 3.10 ditunjukkan tampilan halaman pelaporan dan proses untuk menambahkan laporan dari karyawan yang akan dilihat laporannya. Laporan setiap karyawan akan ditempatkan dalam kotak yang berbeda untuk bisa membandingkan laporan tersebut secara efisien tanpa perlu melakukan *scroll* halaman untuk menemukan *Progress-Plan-Problem* yang dicari, namun cukup melakukan *scroll* dalam kotak milik masing-masing pengguna.



Gambar 3.10 Halaman pelaporan *Progress-Plan-Problem*

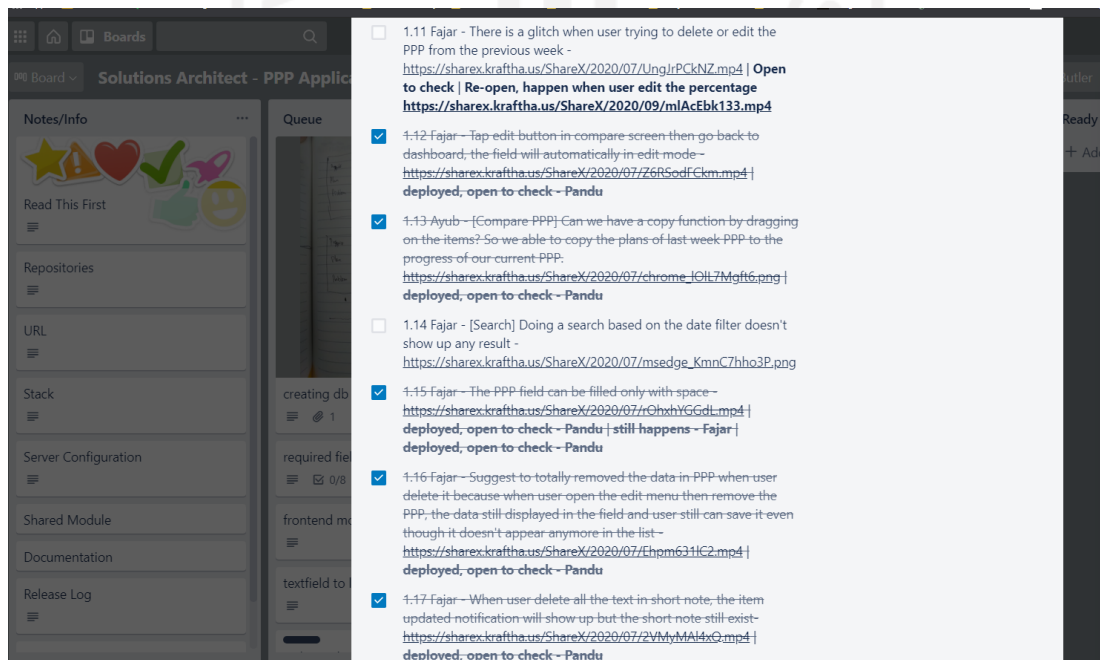
Halaman ini mendapatkan banyak masukan dan perubahan untuk menjadikan penggunaan fitur pelaporan menjadi lebih efisien dan mudah digunakan. Dikarenakan kebutuhan untuk memudahkan pengguna tersebut, maka menjadi tugas khusus bagi tim pengembang untuk mencari cara bagaimana mengkompilasi algoritma sederhana menjadi sebuah alur yang kompleks namun memudahkan pada sisi penggunaan antarmuka. Pada halaman pelaporan, dibutuhkan algoritma khusus untuk dapat mengambil data *Progress-Plan-Problem* yang sesuai dari Redux Store berdasarkan setiap nama pemilik dalam setiap kotak pelaporan, termasuk ketika dilakukan proses penambahan kotak baru di tengah proses pelaporan dan penghapusan kotak lama.

### 3.1.5 Pengendalian, Pengujian, dan Evaluasi Proyek

Untuk memastikan aplikasi yang dikembangkan berjalan dengan baik dan memenuhi fungsi sebagaimana yang dibutuhkan, dilakukan pengujian langsung kepada karyawan Krafthaus Indonesia. Pengujian dilakukan dengan melakukan observasi ketika karyawan menggunakan aplikasi dilanjutkan dengan melakukan wawancara mengenai pengalaman penggunaan aplikasi. Hasil dari pengujian dan wawancara berupa kendala dari fitur dan alur penggunaan aplikasi, serta saran mengenai *improvement* yang dapat ditambahkan.

Proses pengecekan dan pemantauan aktivitas pengerjaan proyek *Progress-Plan-Problem App* yang dilakukan oleh *software developer* menggunakan Trello dan Slack sebagai media komunikasi dengan tim *quality assurance* serta *project manager*. Seluruh karyawan Krafthaus Indonesia diberikan akses ke Trello Board pengembangan *Progress-Plan-Problem App* namun sebatas untuk dapat memantau perkembangan serta melaporkan adanya kendala atau *bug* yang ditemukan. Gitlab digunakan sebagai *platform versioning* dan *deployment* yang dipantau oleh *solution architect* dari Krafthaus Indonesia.

*Progress-Plan-Problem App* di-deploy di *server* milik Krafthaus untuk dapat diakses bersama dengan aplikasi Panel Krafthaus yang telah diimplementasikan sebelumnya. Setiap karyawan Krafthaus dapat menggunakan *Progress-Plan-Problem App* dengan menggunakan kredensial surel Krafthaus Indonesia yang telah dimiliki.



Gambar 3.11 Daftar pelaporan *bug* pada papan Trello pengembangan aplikasi

Pengujian *Progress-Plan-Problem* App dilakukan oleh tim *Quality Assurance* (QA) terhadap setiap fitur yang ada. Setiap *bug* yang ditemukan oleh tim QA akan dituliskan pada daftar *bug* di *board* Trello yang ditunjukkan pada gambar 3.11. *Board* Trello ini khusus disediakan untuk koordinasi pengembangan *Progress-Plan-Problem* App. QA akan menuliskan setiap *bug* dengan keterangan tempat terjadinya penyebab dan tangkapan layar bukti terjadinya *bug* tersebut. Keterangan nama anggota QA yang menemukan *bug* tersebut juga dituliskan untuk memudahkan *developer* jika perlu komunikasi lebih lanjut mengenai *bug* yang ditemukan.

*Developer* kemudian akan melakukan perbaikan terhadap *bug* yang dilaporkan tersebut dan melakukan *commit* kode baru dan *deploy* ulang aplikasi di *server*. Kemudian *developer* menghubungi kembali tim QA untuk mengkonfirmasi bahwa *bug* telah diperbaiki. Setelah *bug* dikonfirmasi selesai diperbaiki, *bug* bersangkutan yang ada di *board* Trello akan ditandai sebagai selesai.

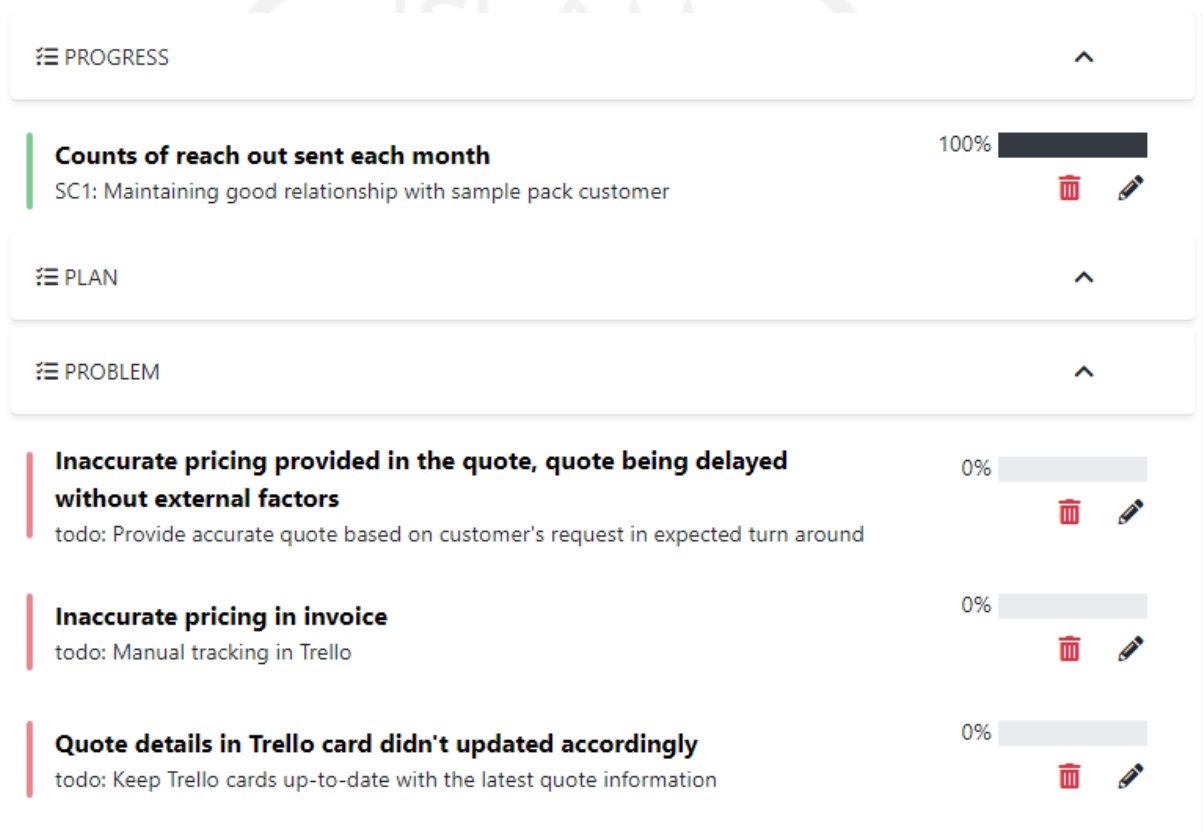
*Progress-Plan-Problem* App digunakan secara khusus oleh pihak manajemen Krafthaus Indonesia untuk membantu pemantauan OKR skala tim dan skala perusahaan. OKR disusun untuk meningkatkan kinerja perusahaan terhadap suatu *metric* tertentu, sehingga setiap karyawan yang dilibatkan dalam penugasan salah satu OKR akan mengisi jam kerjanya tidak hanya mengerjakan tugas rutin, namun juga mengembangkan strategi untuk meningkatkan pencapaian perusahaan.

Paperlust Account Team	2020	Passing Value	February	March	April	May	June	July	August
<b>TEAM OKR</b>									
Objective 1: Increase Sales Conversion (SC)									
<b>Key Performance Index</b>									
SC1 Maintaining good relationship with sample pack customer	100%	39.00%	100%	100%	100%	100%	100%	100%	100%
SC2 Approach existing customer for remarketing	100%	82.95%	100%	100%	100%	100%	100%	100%	100%
Provide accurate quote based on customer's request in SC3 expected turn around	87.50%	99.00%	100%	100%	100%	95.24%	100%	96.15%	
SC4 Provide accurate pricing for invoices	91%	100%	100%	100%	100%	100%	100%	100%	100%
Keep Trello cards up-to-date with the latest quote									
SC5 information	91%	99.67%	100%	100%	100%	99.07%	100%	99.54%	
<i>Average</i>	94%	84.12%	100%	100%	100%	98.86%	100%	99.14%	

Gambar 3.12 Pemodelan OKR pada tim Account di Paperlust

Tim Account di Krafthaus Indonesia menangani tugas berkaitan dengan *customer service* pada layanan Paperlust. Gambar 3.12 menunjukkan salah satu OKR yang disusun dalam tim Account Paperlust. Dapat dilihat satu *objective* tersebut memiliki 5 *key-result* yang masing-masing memiliki persentase ketercapaian dan *metric* pengukuran. Setiap karyawan yang pada tim Account Paperlust ditempatkan untuk fokus pada pengerjaan *key-result* tertentu.

Karyawan yang mendapat tugas untuk fokus pada satu atau lebih *key-result* akan menyisihkan sebagian jam kerjanya untuk fokus pada pencapaian *key-result* tersebut. Sesuai dengan jangka waktu pelaporan rutin mingguan di Krafthaus Indonesia, setiap karyawan akan melaporkan perencanaan, pencapaian, dan kendala yang dihadapi melalui *Progress-Plan-Problem* App. Gambar 3.13 menunjukkan laporan *Progress-Plan-Problem* dari salah satu karyawan yang fokus untuk menampilkan pencapaian dan kendala yang dihadapi. Dari laporan tersebut, pihak manajemen Krafthaus Indonesia dapat memantau kemajuan pencapaian *objective* yang sebelumnya telah disusun. Seperti ditunjukkan pada gambar 3.13, ada beberapa kendala yang menghambat pencapaian *objective* yang ditunjukkan pada gambar 3.12 sebelumnya.



Gambar 3.13 *Progress-Plan-Problem* salah satu karyawan tim Account Paperlust

### 3.1.6 Penutupan Proyek

Penutupan aktivitas pengerjaan pengembangan aplikasi *Progress-Plan-Problem* App masih belum dilakukan. Dengan masa magang dari penulis yang sudah selesai dan hanya tersisa beberapa *bug* yang tidak terlalu berdampak besar pada alur penggunaan aplikasi secara umum, pengembangan dari *Progress-Plan-Problem* App akan dilanjutkan oleh karyawan lain yang masih bekerja di Krafthaus Indonesia.

## BAB IV

### REFLEKSI PELAKSANAAN MAGANG

#### 4.1 Aplikasi Pelaporan *Progress-Plan-Problem* App Krafthaus Indonesia

*Progress-Plan-Problem* pada dasarnya sebagai teknik manajemen pelaporan status pekerjaan diterapkan oleh Krafthaus Indonesia untuk mengintegrasikan alur pelaporan pekerjaan karyawan dengan lebih mudah terutama untuk kelanjutannya yang akan di-*review* oleh *team leader* serta manajemen dari Krafthaus Indonesia. Tidak seperti daftar pekerjaan yang jauh mendetail, *Progress-Plan-Problem* cenderung singkat dengan setiap poinnya melaporkan pencapaian, perencanaan atau kendala dalam sebuah periode waktu pekerjaan.

*Progress-Plan-Problem* App dikembangkan sesuai dengan alur pengembangan perangkat lunak untuk membantu implementasi teknik manajemen pelaporan *Progress-Plan-Problem*. Tahap pertama adalah dengan menentukan *requirement* yang dibutuhkan dari aplikasi yang akan dikembangkan dengan diskusi dan *brainstorming* bersama pihak yang memiliki kepentingan. Kemudian dilakukan analisis dan desain dari solusi yang akan dikembangkan dalam bentuk menentukan teknologi yang akan digunakan serta membuat *prototype* aplikasi. Dilanjutkan dengan implementasi dari desain yang sebelumnya telah dibuat menjadi aplikasi yang dapat menjalankan berbagai fungsi dan memenuhi kebutuhan yang telah diidentifikasi sebelumnya. Setelah itu langkah terakhir yang dilakukan adalah melakukan evaluasi secara terus menerus terhadap aplikasi serta perbaikan dari *bug* dan *error* yang masih ada.

Apabila dilakukan komparasi dengan tanpa adanya *Progress-Plan-Problem* App, pada proses pelaporan dan koordinasi oleh tim manajemen Krafthaus Indonesia dengan berbagai *team leader* dalam Krafthaus Indonesia, setiap *team leader* dapat dengan mudah melaporkan hasil pencapaian setiap *team member*-nya untuk bisa berbagi laporan kemajuan dengan tim lainnya. Tidak lagi diperlukan pencatatan ulang oleh masing-masing *team leader* mengenai pencapaian, perencanaan atau kendala *team member*-nya karena dalam *Progress-Plan-Problem* App seluruh laporan setiap orang dapat dilihat oleh rekan kerja yang lain dengan fitur pencarian yang ada. Pelaporan dan koordinasi dilakukan dengan cukup melihat satu layar fitur pelaporan dari *Progress-Plan-Problem* App tanpa perlu membuka referensi dari berbagai aplikasi pencatatan tugas, *scrum board*, Trello, atau sejenisnya.



## 4.2 Manajemen Tugas dan Penentuan OKR

Aplikasi Pelaporan *Progress-Plan-Problem* App mengubah konsep *meeting* berbagai tim yang terdapat dalam tim Krafthaus Indonesia. Setiap tim melakukan *meeting* koordinasi *Progress-Plan-Problem* yang umumnya dilakukan setiap satu pekan sekali dalam hari yang ditentukan sendiri oleh tim tersebut. Dalam berjalannya *meeting*, dasbor *Progress-Plan-Problem* App selalu menjadi sumber referensi utama disamping Trello dan sumber pendukung lainnya untuk menginformasikan hasil pekerjaan, kendala serta perencanaan kepada seluruh anggota tim. Konsep *Progress-Plan-Problem* yang mengutamakan penulisan poin singkat dan spesifik yang menjelaskan setiap bagian pekerjaan dengan mudah disampaikan untuk dipahami oleh seluruh anggota tim dalam setiap *meeting* koordinasi.

Tugas sebagai dasar untuk menyelesaikan pekerjaan di organisasi, menjadi fokus utama dari setiap karyawan tanpa terkecuali. OKR lebih dikhususkan untuk dikelola oleh sisi manajemen perusahaan, yaitu *Managing Director* serta jajaran manajer. Untuk memastikan sinergi antara OKR dan tugas yang dilakukan masuk ke ranah kerja sisi manajemen perusahaan. OKR disusun sesuai dengan keadaan perusahaan mengenai apa yang ingin dicapai oleh perusahaan dalam kurun waktu tertentu. Praktik penerapan OKR umumnya menggunakan kurun waktu 3 bulan atau 1 kuartal untuk penyusunan, pelaksanaan dan evaluasi sebuah *objective*, untuk kemudian menyusun kembali OKR untuk 3 bulan berikutnya. OKR memberikan gambaran yang jelas dari visi dan misi perusahaan yang umumnya sangat umum dalam penafsirannya. Setiap *objective* disusun menyesuaikan keadaan perusahaan serta apa yang ingin dicapai, dengan tetap mengacu pada visi dan misi perusahaan.

Fitur pelaporan menjadi fitur yang berperan penting dalam rapat koordinasi mingguan antara pihak manajemen dengan *team leader* masing-masing, serta *briefing* harian setiap tim dalam Krafthaus Indonesia. Di Krafthaus Indonesia, setiap *role* pekerjaan dibuat sebagai sebuah tim yang akan mengerjakan tugas untuk membantu pekerjaan tim lainnya. Seperti tim *user experience specialist* membantu tim pengembang untuk menyusun konsep dan rancangan antarmuka halaman web. Setiap tim memiliki satu hari di setiap pekannya untuk melakukan evaluasi mingguan untuk membahas status pekerjaan masing-masing anggota timnya. Setiap dua pekan juga dilakukan evaluasi *sprint* karena Krafthaus Indonesia menerapkan kerangka *Scrum* sebagai kerangka manajemen proyek dan menetapkan dua pekan sebagai durasi satu *sprint*.

Manajemen Krafthaus Indonesia dan masing-masing *team leader* melakukan koordinasi setiap pekan untuk mendapatkan laporan yang jelas mengenai tugas yang dikerjakan setiap karyawan. Laporan yang didapat oleh manajemen kemudian dipastikan kesesuaiannya dengan

*objectives* perusahaan yang telah disusun dan setiap perubahan yang diperlukan disampaikan kembali pada karyawan melalui *team leader* masing-masing. Fitur pelaporan memberikan kemudahan ini dalam satu layar halaman web tidak seperti aplikasi manajemen tugas pada umumnya yang telah banyak digunakan.

### 4.3 Manfaat Magang untuk Mahasiswa

#### 4.3.1 Bekerja dalam tim profesional

Untuk mewujudkan tujuan perusahaan atau organisasi, tidak dapat dilakukan dengan usaha individu tetapi harus diusahakan dari kerjasama beberapa orang dalam tim. Berbagai tuntutan dalam tim dibuat untuk memastikan tim dapat memberikan hasil maksimal, tidak menghambat anggota tim lainnya. Beberapa diantaranya adalah mempelajari hal baru, menyelesaikan masalah dengan rapi dan kritis, serta berkonsultasi dengan rekan tim untuk menyelesaikan permasalahan yang ditemui.

#### 4.3.2 Mempelajari manajemen tim dan manajemen tugas

Terlibat dalam perusahaan pengembang aplikasi di masa pandemi menjadikan kegiatan perusahaan yang pada keadaan normal berlangsung di studio harus berpindah ke rumah masing-masing. Diperlukan usaha yang lebih untuk memastikan setiap bagian perusahaan tetap berfungsi sebagaimana mestinya dengan maksimal walaupun terpisah secara tempat. Koordinasi tim menjadi lebih intens melalui aplikasi perpesanan serta konferensi video untuk menjaga komunikasi, konsultasi kendala yang ditemui, dan koordinasi pembagian pekerjaan.

Untuk membantu manajemen dan pemantauan tugas digunakan *to-do list* pribadi masing-masing karyawan serta daftar *backlog* pekerjaan di papan Trello masing-masing tim. *Team leader* juga secara rutin memantau *backlog* masing-masing karyawan dalam timnya dan menanyakan apabila ada satu atau beberapa poin yang terlihat seperti macet atau belum ada kemajuan.

#### 4.3.3 Beradaptasi dengan budaya perusahaan

Budaya perusahaan merupakan mekanisme peraturan dan konsep tidak berwujud, individu bersatu ke dalam struktur sosial dan dapat digunakan oleh pimpinan perusahaan untuk mendukung perubahan strategis serta memengaruhi karyawan dan operasi organisasi (Agusinta et al., 2017). Tidak selamanya harus dengan hal-hal besar seperti ketetapan perusahaan atau mengadakan *event* tertentu. Tetapi budaya perusahaan dapat terlihat dari hal-hal sederhana seperti karyawan yang mengucapkan selamat pagi di grup Skype perusahaan setiap paginya, mengucapkan selamat

berakhir pekan atau liburan, serta memiliki grup yang terpisah untuk membahas hal-hal sehari-hari di luar topik mengenai pekerjaan. Krafthaus Indonesia juga memiliki rutinitas *Friday Sharing* yang memberikan kesempatan bagi karyawan secara bergantian untuk berbagi cerita, mengajarkan ilmu baru atau menghibur rekan-rekan kerjanya dengan mengadakan permainan-permainan sederhana untuk meningkatkan kekeluargaan antar karyawan.



## BAB V

### KESIMPULAN DAN SARAN

#### 5.1 Kesimpulan

Pengembangan *Progress-Plan-Problem* App berhasil memecahkan masalah yang menjadi latar belakang dibuatnya aplikasi tersebut. Setelah pengembangan dan pengujian, dilakukan wawancara dengan 5 karyawan yang memiliki peran sebagai team leader di Krafthaus Indonesia, dapat disimpulkan bahwa dengan menggunakan *Progress-Plan-Problem* App memberikan kemudahan untuk mengkoordinasikan setiap tugas karyawan dibawah bimbingan *team leader* masing-masing. Kemudian wawancara dengan manajemen Krafthaus Indonesia menunjukkan hal serupa bahwa penggunaan *Progress-Plan-Problem* App membantu koordinasi tugas menuju pemenuhan objectives perusahaan.

Susunan komponen MERN Stack yang berbasis Javascript memberikan kemudahan dibandingkan susunan komponen berbasis PHP, terutama dalam bidang manajemen basis data serta persiapan *server* dikarenakan MERN Stack yang menggunakan Node.js tidak terbatas oleh sistem operasi yang digunakan pada *server*. Pengembangan aplikasi dengan susunan komponen MERN Stack memudahkan pengembangan dengan cukup menggunakan basis bahasa Javascript untuk seluruh komponen aplikasi daripada susunan komponen berbasis PHP yang seringkali tetap memerlukan komponen Javascript untuk pengembangan *front-end*. Sehingga memberikan beban tambahan bagi tim pengembang untuk menambahkan anggota dengan keahlian PHP dan Javascript untuk melakukan integrasi antara kode PHP dan kode Javascript.

Program magang memberikan pengalaman secara langsung untuk terlibat aktif dalam industri teknologi informasi. Dengan terlibat langsung dalam pengembangan aplikasi di Krafthaus Indonesia, mahasiswa akan mendapat beberapa manfaat di antaranya:

- Pengalaman bekerjasama dalam sebuah tim pengembang aplikasi
- Mempelajari berbagai cara untuk manajemen tim pengembang aplikasi walaupun terpisah jarak dikarenakan kebijakan kerja dari rumah
- Mengamati dan mempelajari bagaimana kultur perusahaan, manajemen dan pengelolaan tim dalam sebuah perusahaan berbasis teknologi informasi untuk menjaga komunikasi, relasi dan kekeluargaan dalam perusahaan
- Mempelajari teknis keahlian pemrograman dalam skala industri. Memecahkan permasalahan yang lebih kompleks daripada apa yang dipelajari di kelas dalam bidang yang sesuai dengan proyek yang diberikan kepada mahasiswa.

## 5.2 Saran

Program Studi Informatika dalam menyusun penjaluran tugas akhir jalur magang dan jalur lainnya sebaiknya memberikan sosialisasi kepada mahasiswa yang berencana mengambil jalur tugas akhir. Sebagian besar mahasiswa mulai mempertimbangkan jalur tugas akhir yang akan ditempuh sejak awal semester 4 atau 5. Mahasiswa sering merasa kebingungan bagaimana cara untuk memilih penjaluran tugas akhir yang akan diambil. Sosialisasi ini sebaiknya memberikan informasi tanggal-tanggal penting dalam pemilihan jalur tugas akhir, persiapan yang perlu dilakukan oleh mahasiswa untuk jalur tugas akhir tertentu, dan akses ke formulir atau berkas administrasi yang perlu disiapkan oleh mahasiswa.

Mahasiswa yang mengambil jalur magang, sebaiknya membekali diri terlebih dulu dengan pengetahuan umum mengenai budaya perusahaan, etika profesi dan penerapannya. Suasana keseharian di perusahaan tentunya berbeda dengan kelas kuliah, sehingga mahasiswa dituntut untuk dapat beradaptasi dengan cepat dan mengatur komunikasi yang baik, aktif, dan sopan dengan rekan lainnya di perusahaan tempat magang.

Dalam pelaksanaan magang, seringkali mahasiswa akan mempelajari kemampuan baru di luar apa yang diajarkan di kelas sehingga mahasiswa diharapkan untuk dapat aktif mencari tahu terlebih dulu teknologi apa saja yang digunakan di tempat magang. Dengan mengetahui terlebih dulu teknologi yang digunakan di tempat magang, mahasiswa diharapkan untuk dapat mempersiapkan diri terlebih dulu dengan membiasakan tersebut, belajar dan berlatih menggunakan teknologi tersebut. Sehingga ketika pelaksanaan magang dimulai, mahasiswa dapat lebih cepat memberikan kontribusi dalam proyek yang dikerjakan.

Untuk mendapatkan manfaat sebesar mungkin dari OKR, diperlukan agar seluruh pihak yang terlibat dalam organisasi dari karyawan perusahaan hingga pihak manajemen untuk mengerti konsep dan tujuan dari OKR. Agar dapat memunculkan inspirasi dan meningkatkan performa karyawan, setiap karyawan harus memahami kaitannya antara pekerjaan yang sedang dikerjakan dengan kontribusinya terhadap pemenuhan tujuan perusahaan.

Pengembangan *Progress-Plan-Problem* App telah memenuhi tujuan dibuatnya aplikasi tersebut. Untuk pengembangan lebih lanjutnya dapat dilakukan integrasi dengan aplikasi lainnya yang sudah digunakan oleh perusahaan seperti Trello, serta dibuat fitur untuk otomasi pembuatan laporan dalam format *spreadsheet* atau *portable document format*.

## DAFTAR PUSTAKA

- Agusinta, L., Pahrudin, C., & Wildan, W. (2017). Budaya Perusahaan Dan Kinerja Karyawan Transportasi Udara. *Jurnal Manajemen Transportasi Dan Logistik*, 4(2), 123. <https://doi.org/10.25292/j.mtl.v4i2.97>
- Anyaeche, C. O., & Oluwole, L. (2012). Project Progress Evaluation : Methodology and Reporting Template. *J. of NIIE*, 3(January), 36–47. [https://www.researchgate.net/publication/257362879\\_Project\\_Progress\\_Evaluation\\_Methodology\\_and\\_Reporting\\_Template](https://www.researchgate.net/publication/257362879_Project_Progress_Evaluation_Methodology_and_Reporting_Template)
- Bhardwaj, P., & Kumar, D. (2018). A Review Paper on MERN Stack for Web. *International Journal of Innovative Research in Computer and Communication Engineering*, 6(4), 3615–3617. <https://doi.org/10.15680/IJIRCCE.2018.0604067>
- Doerr, J. (2018). Measure What Matters: How Google, Bono, and the Gates Foundation Rock the World with OKRs. In *Library of Congress Cataloging-in-Publication Data*. Penguin Publishing Group. <https://doi.org/10.1080/07347332.2020.1749212>
- Riss, U., Rickayzen, A., Maus, H., & van der Aalst, W. (2005). Challenges for business process and task management. *Journal of Universal Knowledge Management*, 2(January), 77–100.
- Tilkov, S., & Vinoski, S. (2010). Node.js: Using JavaScript to build high-performance network programs. *IEEE Internet Computing*, 14(6), 80–83. <https://doi.org/10.1109/MIC.2010.145>
- Trinkenreich, B., Santos, G., Barcellos, M. P., & Conte, T. (2019). Combining GQM+Strategies and OKR - Preliminary Results from a Participative Case Study in Industry. *Lecture Notes in Computer Science (Including Subseries Lecture Notes in Artificial Intelligence and Lecture Notes in Bioinformatics)*, 11915 LNCS(November), 103–111. [https://doi.org/10.1007/978-3-030-35333-9\\_7](https://doi.org/10.1007/978-3-030-35333-9_7)

LAMPIRAN

