

Pengembangan Aplikasi Pelaporan *Progress-Plan-Problem* untuk Manajemen Tugas dan Penentuan OKR di Krafthaus Indonesia

Muhammad Pandu Widodo
Jurusan Informatika, Fakultas Teknologi Industri
Universitas Islam Indonesia
Yogyakarta
17523055@students.uui.ac.id

Andhik Budi Cahyono
Jurusan Informatika, Fakultas Teknologi Industri
Universitas Islam Indonesia
Yogyakarta
andhik.budi@uui.ac.id

Abstract—Krafthaus Indonesia menerapkan *framework Objective-Key Result (OKR)* untuk membantu penentuan arah & 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.

Keywords—Progress-Plan-Problem, MERN Stack, OKR, Perencanaan.

I. PENDAHULUAN

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 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 penting. 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 [1].

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 (*progress*) yang dimaksud di sini adalah hal apa saja yang telah berhasil dikerjakan dalam kurun waktu pelaporan. Target (*plan*) adalah perencanaan dari hal apa saja yang akan dikerjakan di kurun waktu pelaporan selanjutnya. Hambatan (*problem*) adalah kendala yang ditemui dan membutuhkan bantuan dari rekan kerja lainnya.

Metode pelaporan tugas 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. Namun ditemui beberapa kendala 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 permasalahan tersebut, Krafthaus Indonesia berinisiatif untuk mengembangkan aplikasi *inhouse* untuk memudahkan pelaporan tugas dengan metode PPP.

Berdasarkan uraian yang telah dikemukakan di atas, Krafthaus Indonesia memilih metode pelaporan tugas PPP sebagai metode pelaporan status pekerjaan dalam batas waktu tertentu. Selain itu, dalam implementasi 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 [2]. 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.

Pada makalah ini disajikan bagaimana pengembangan aplikasi web dengan teknologi *stack* komponen Javascript yang digunakan untuk manajemen pelaporan pekerjaan dengan menerapkan konsep PPP. Aplikasi ini tidak ditujukan untuk menggantikan aplikasi manajemen tugas teknis, *to-do list*, atau *scrum board*. Tetapi lebih kepada perencanaan dan pelaporan kemajuan atau kendala pada aspek strategis dari tugas teknis tersebut.

II. DASAR TEORI

A. 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 dalam kurun waktu pelaporan [3]. 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.

B. OKR

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 [4]. *Objective* adalah tujuan organisasi yang ingin dicapai, sementara *Key-Result* adalah langkah-langkah terukur dapat diukur untuk mencapai tujuan tersebut. Metode OKR membantu dalam mendefinisikan dan pemantauan dari tujuan serta hasil yang diharapkan, sehingga metode ini menjadi banyak digunakan di berbagai industri.

C. 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 [5]. Interaksi antar-komponen dalam MERN Stack diilustrasikan dalam gambar 1.



Gambar 1 Visualisasi *framework* MERN Stack

- MongoDB, 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.
- Express.js, 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.
- React.js, *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.
- Node.js, 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 [6]. 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.

III. METODOLOGI

Metodologi yang digunakan sesuai dengan alur pengembangan perangkat lunak. 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.

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.

IV. PEMBAHASAN DAN HASIL

A. Pembahasan

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. *Project Manager* juga berkoordinasi dengan *stakeholder* aplikasi ini yaitu *Managing Director* Krafthaus Indonesia.
- *Quality Assurance*, bertanggung jawab untuk memastikan implementasi solusi yang telah dikerjakan oleh *software developer* berjalan sesuai dengan kesepakatan yang telah dibuat sebelumnya.
- *Software Developer*, bertanggung jawab atas eksekusi dan implementasi rancangan kebutuhan dari aplikasi dalam bentuk kode program.

Pengembangan *Progress-Plan-Problem* App terbagi menjadi beberapa fitur dengan *timeline* pengembangan yang dirincikan dalam tabel 1.

TABEL 1 TIMELINE DAN FITUR PENGEMBANGAN PROGRESS-PLAN-PROBLEM APP

No	Aktivitas	Durasi
1	Diskusi <i>requirement</i> dari <i>Progress-Plan-Problem</i> App dengan melibatkan rekan-rekan pengembang dan User Experience Analyst dari berbagai tim yang ada dalam Krafthaus Indonesia. Diskusi mengenai <i>problem statement</i> , <i>ideation</i> , hingga perancangan solusi.	1 Pekan
2	Mengembangkan basis data <i>Progress-Plan-Problem</i> App dengan MongoDB. Dibuat 1 <i>cluster</i> basis data dengan 3 <i>collection</i> untuk menyimpan data <i>Progress-Plan-Problem</i> , data label/kategori, dan data pekan. Data pengguna diambil dari <i>cluster</i> lain yang telah dibuat sebelumnya untuk aplikasi internal Krafthaus Indonesia yang lain	1 Pekan
3	Mengembangkan kerangka <i>back-end</i> <i>Progress-Plan-Problem</i> App dengan Express.js terintegrasi dengan sistem Panel Krafthaus yang sebelumnya telah dikembangkan sebelumnya.	2 Pekan

4	Mengembangkan kerangka <i>front-end</i> <i>Progress-Plan-Problem</i> App dengan React.js. Komponen dan elemen penyusun halaman web yang dibuat di aplikasi <i>Progress-Plan-Problem</i> dikembangkan sebagai elemen yang dinamis dan mampu berdiri sendiri tanpa terlalu memengaruhi komponen yang lainnya.	1 Pekan
5	Membuat Redux Store untuk manajemen data pada <i>front-end</i> <i>Progress-Plan-Problem</i> App. Data dalam Redux Store berubah setiap dilakukan <i>dispatch</i> untuk melakukan perubahan pada isi dari Redux Store yang dikirim oleh fungsi atau metode yang telah disusun. <i>Dispatch</i> biasa dikirimkan setelah melakukan <i>request</i> dan menerima data dari <i>server</i> , atau setelah melakukan perubahan pada komponen dalam aplikasi React.js.	1 Pekan
6	Mengembangkan fitur CRUD <i>Progress-Plan-Problem</i> App. CRUD yang dibuat adalah untuk data <i>Progress-Plan-Problem</i> , 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.	1 Bulan
7	Mengembangkan fitur komparasi <i>Progress-Plan-Problem</i> per-pekan. Karena pelaporan <i>Progress-Plan-Problem</i> menjadi agenda pelaporan mingguan, dibutuhkan fitur untuk dapat melakukan komparasi <i>Progress-Plan-Problem</i> antar pekan. Hal ini dilakukan untuk dapat menemukan poin <i>Progress-Plan-Problem</i> yang mana yang sekiranya memiliki kemajuan atau kendala, sehingga pemetaan masalah dapat dilakukan dengan lebih mudah	1 Bulan
8	Mengembangkan fitur pencarian <i>Progress-Plan-Problem</i> . Untuk dapat menemukan data spesifik yang diperlukan dengan memasukkan kata kunci tertentu pada <i>form</i> yang telah disediakan. Pencarian <i>Progress-Plan-Problem</i> dalam aplikasi ini memiliki empat parameter, yaitu nama pemilik <i>Progress-Plan-Problem</i> , judul, tanggal, dan label. Keempat parameter tersebut dapat digunakan secara terpisah atau bersamaan satu dengan lainnya.	2 Pekan
9	Mengembangkan fitur pelaporan <i>Progress-Plan-Problem</i> App. Dengan pertimbangan bahwa pelaporan <i>Progress-Plan-Problem</i> setiap karyawan Krafthaus Indonesia akan ada kaitannya dengan karyawan lainnya baik dalam satu tim pengembang maupun	2 Bulan

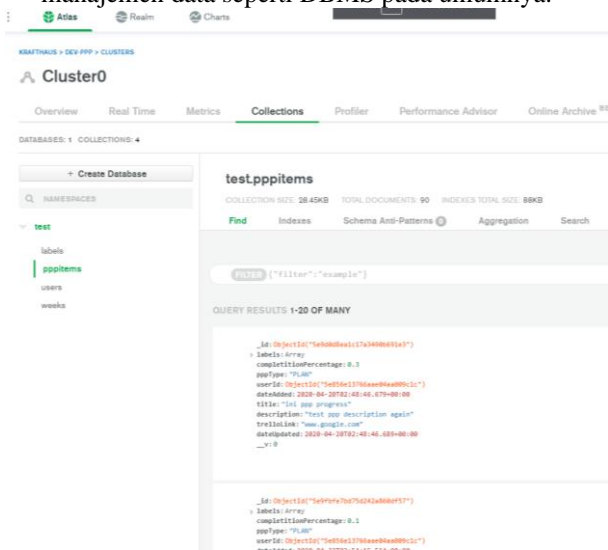
dari tim lainnya sebagai bentuk kolaborasi dalam pekerjaan, dibutuhkan tampilan khusus untuk dapat membandingkan laporan *Progress-Plan-Problem* dari beberapa orang sekaligus.

B. Hasil

1) *Progress-Plan-Problem* App

Berdasarkan pembahasan implementasi pengembangan *Progress-Plan-Problem* App menghasilkan hal berikut:

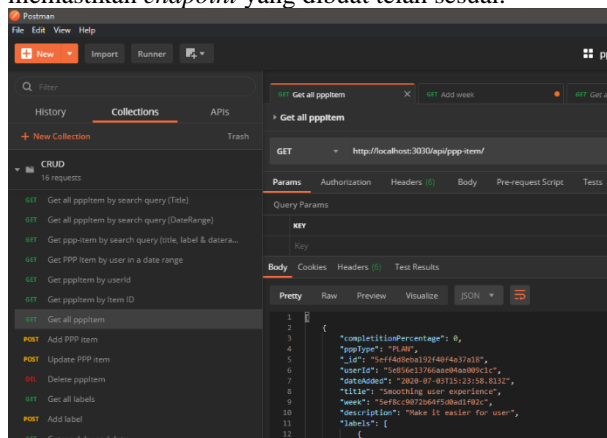
- Basis data MongoDB *Progress-Plan-Problem* App
Manajemen basis data MongoDB dilakukan pada platform MongoDB Atlas yang ditunjukkan pada gambar 2. MongoDB Atlas menyusun basis data di dalam *cluster* dan dapat digunakan untuk melakukan manajemen data seperti DBMS pada umumnya.



Gambar 2 Cluster basis data MongoDB

- Kerangka *back-end* *Progress-Plan-Problem* App dengan Express.js

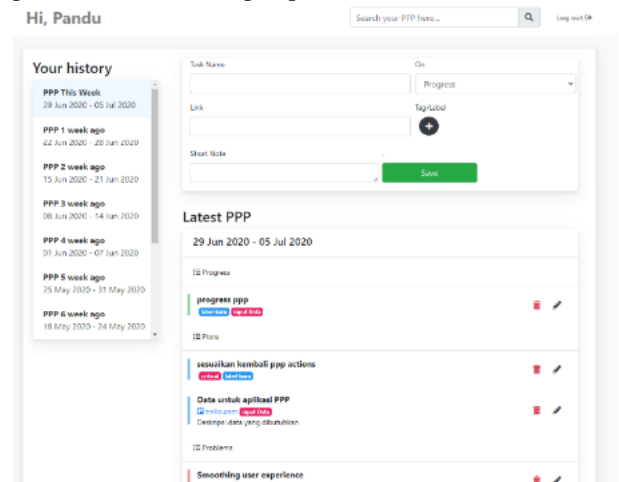
Request yang telah disusun, dikirim ke salah satu *endpoint back-end* yang ingin diuji dengan mengisi *endpoint* yang dimaksud pada kolom URL yang ada. Postman akan menerima *reply* dari *server* atas request yang dikirimkan sebelumnya, dan menampilkan isi dari *reply* tersebut di jendela di bawah konfigurasi *request*. Ini ditunjukkan pada gambar 3 untuk memastikan *endpoint* yang dibuat telah sesuai.



Gambar 3 Pengujian *back-end* dengan Postman

- Kerangka *front-end* *Progress-Plan-Problem* App dengan React.js

Front-end *Progress-Plan-Problem* App terdiri dari halaman dasbor utama yang ditunjukkan pada gambar 4, halaman pencarian, halaman komparasi pekan, serta halaman pelaporan.

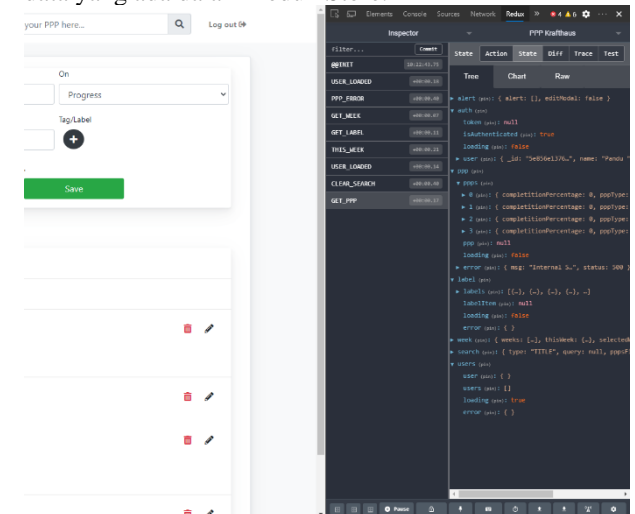


Gambar 4 Dasbor utama *Progress-Plan-Problem* App

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

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.

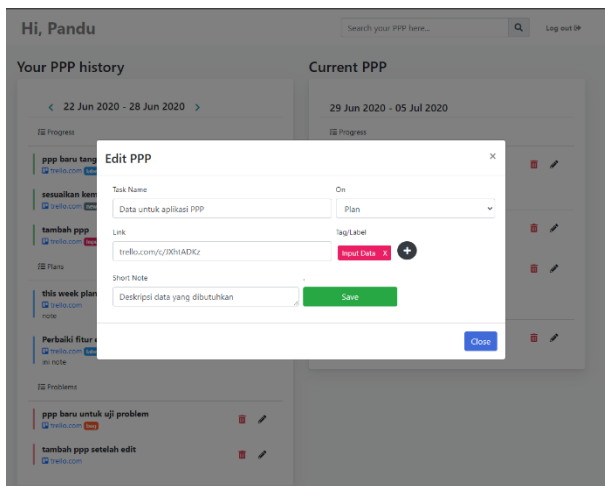
Dalam gambar 5 ditampilkan halaman dasbor dengan Redux DevTools di sebelah kanan layar untuk dapat memantau setiap perubahan data yang terjadi dalam Redux Store. Redux DevTools membantu pengembang untuk dapat melihat rincian dari setiap data yang ada dalam Redux Store.



Gambar 5 Analisis Redux DevTools di kanan layar

- Fitur CRUD *Progress-Plan-Problem* App

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 6 berikut.



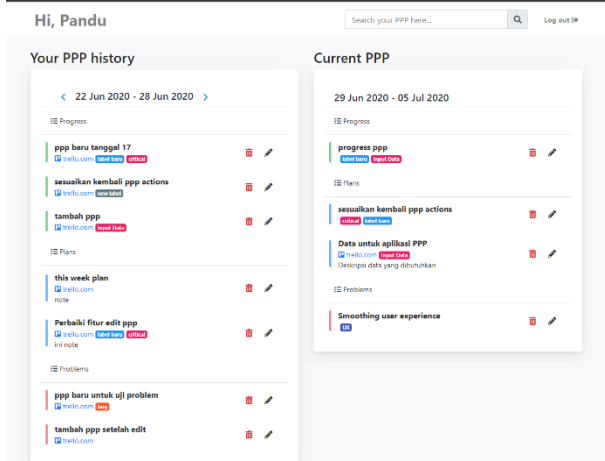
Gambar 6 Pop-up modal sunting Progress-Plan-Problem

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

f) Fitur komparasi *Progress-Plan-Problem* per-pekannya

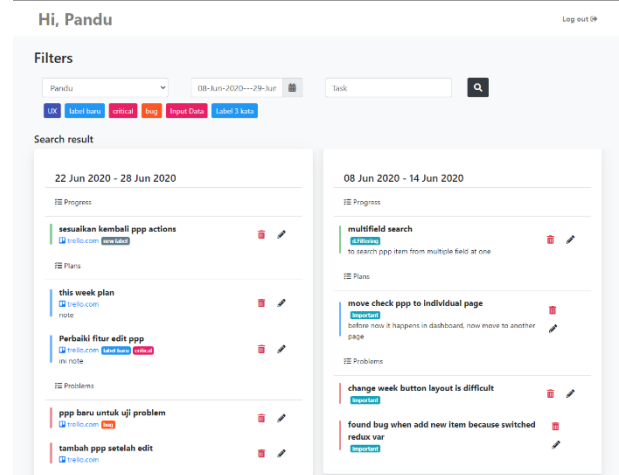
Karena pelaporan *Progress-Plan-Problem* menjadi agenda pelaporan mingguan, dibutuhkan fitur untuk dapat melakukan komparasi *Progress-Plan-Problem* antar pekan. Hal ini dilakukan untuk dapat menemukan poin *Progress-Plan-Problem* yang mana yang sekiranya memiliki kemajuan atau kendala, sehingga pemetaan masalah dapat dilakukan dengan lebih mudah. Halaman komparasi *Progress-Plan-Problem* pada gambar 7 menampilkan *Progress-Plan-Problem* pada pekan saat ini di sebelah kanan, dan *Progress-Plan-Problem* pada pekan sebelumnya di sebelah kiri.



Gambar 7 Halaman komparasi Progress-Plan-Problem antar pekan

g) Fitur pencarian *Progress-Plan-Problem*

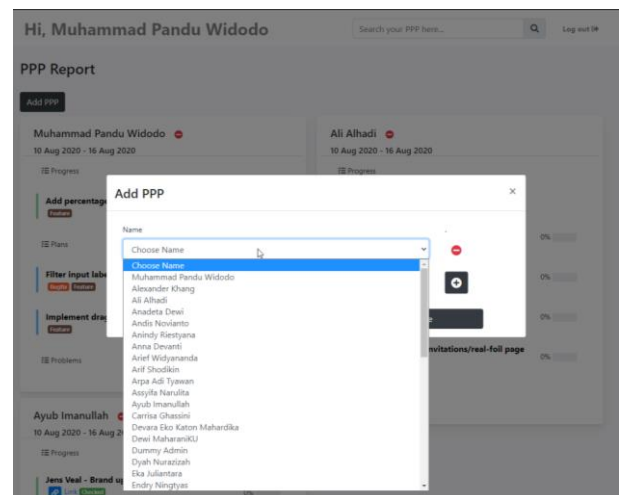
Pencarian dalam *Progress-Plan-Problem* App dikembangkan dalam satu halaman tersendiri yang dapat dilihat pada gambar 8. Di bagian atas ditampilkan form berisi kolom masukan yang menjadi kriteria pencarian *Progress-Plan-Problem*. Kolom masukan terdiri dari masukan nama, *range* tanggal, judul, serta label. Kolom masukan pencarian dapat menggunakan hanya satu kolom saja atau menggunakan beberapa kolom bersamaan.



Gambar 8 Halaman pencarian Progress-Plan-Problem

h) Fitur pelaporan *Progress-Plan-Problem* App

Fitur pelaporan dikembangkan dalam 2 tahap. Tahap pertama dengan alur yang sederhana dengan menambahkan *box* laporan langsung bersama dengan isi laporan milik karyawan yang diinginkan seperti ditunjukkan dalam gambar 9. Tetapi alur ini dianggap terlalu merepotkan dan kurang fleksibel.



Gambar 9 Halaman pelaporan Progress-Plan-Problem

Tahap kedua dengan alur baru di mana setiap *box* laporan ditambahkan dalam keadaan kosong, tetapi ditambahkan dengan *form* untuk memilih laporan karyawan yang diinginkan. Alur kedua ini dianggap lebih baik dan diimplementasikan secara tetap.

2) Manajemen Tugas dan OKR

Tugas sebagai dasar untuk menyelesaikan pekerjaan di organisasi, menjadi fokus utama dari setiap karyawan tanpa terkecuali. OKR lebih dikhususkan kepada sisi manajemen perusahaan, yaitu *Managing Director* serta jajaran manajer. Memastikan sinergi antara OKR dan tugas yang dilakukan masuk ke ranah kerja sisi manajemen 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. Manajemen mendapatkan laporan yang jelas mengenai tugas yang dikerjakan setiap karyawan untuk kemudian diselaraskan dengan *objectives* perusahaan melalui arahan yang disampaikan 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.

V. KESIMPULAN

Setelah dilakukan pengembangan, pengujian aplikasi, dan penjabaran sebelumnya, beberapa kesimpulan dapat ditarik adalah sebagai berikut:

- Dari hasil 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 dibanding 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 tim dengan keahlian PHP dan Javascript.

REFERENSI

- [1] U. Riss, A. Rickayzen, H. Maus, and W. van der Aalst, "Challenges for business process and task management," *J. Univers. Knowl. Manag.*, vol. 2, no. January, pp. 77–100, 2005.
- [2] B. Trinkenreich, G. Santos, M. P. Barcellos, and T. Conte, "Combining GQM+Strategies and OKR - Preliminary Results from a Participative Case Study in Industry," *Lect. Notes Comput. Sci. (including Subser. Lect. Notes Artif. Intell. Lect. Notes Bioinformatics)*, vol. 11915 LNCS, no. November, pp. 103–111, 2019, doi: 10.1007/978-3-030-35333-9_7.

- [3] J. Marovt, "Guide: the PPP framework (Progress , Plans , Problems)," 2020. <https://oneteamhq.com/progress-plans-problems/>.
- [4] J. Doerr, *Measure What Matters: How Google, Bono, and the Gates Foundation Rock the World with OKRs*. Penguin Publishing Group, 2018.
- [5] P. Bhardwaj and D. Kumar, "A Review Paper on MERN Stack for Web," *Int. J. Innov. Res. Comput. Commun. Eng.*, vol. 6, no. 4, pp. 3615–3617, 2018, doi: 10.15680/IJIRCCE.2018.0604067.
- [6] S. Tilkov and S. Vinoski, "Node.js: Using JavaScript to build high-performance network programs," *IEEE Internet Comput.*, vol. 14, no. 6, pp. 80–83, 2010, doi: 10.1109/MIC.2010.145.