BAB III

PELAKSANAAN KERJA MAGANG

3.1 Kedudukan dan Koordinasi

Kedudukan pada kerja magang ini adalah sebagai *Business System Analyst*. Pada PT Volante Teknologi Indonesia tugas dari *Business System Analyst* adalah untuk melakukan analisa terhadap proses bisnis dan proses kerja sistem yang ada. Analisa yang dilakukan disini dimulai dari menganalisa *user requirement* yang diberikan, perancangan *flow chart* dan *use case diagram*, perancangan *data structure*, melakukan testing sistem, perancangan *front end* dan *back end* sistem, perancangan user interface sistem, perancangan model bisnis, melakukan *market research*, dan perhitungan biaya sistem. Selain itu juga secara langsung ikut berpartisipasi dalam melakukan projek – projek yang sedang diterima oleh perusahaan. Selama melakukan pengerjaan tugas – tugas yang diberikan, interaksi dengan rekan – rekan kerja yang ditugaskan untuk mengerjakan projek tersebut dilakukan melalui sistem Slack, Trello maupun secara verbal.

3.2 Tugas yang Dilakukan

Tugas yang diberikan adalah untuk melakukan *research* untuk analisa projek yang diterima oleh PT Volante Teknologi Indonesia, membuat business model, ikut dalam membantu melakukan pengembangan sistem tersebut baik *front end* maupun *back end*, menghitung *cost of goods sold* serta *cost structure* dari sistem. Selama melakukan kerja magang di PT Volante Teknologi Indonesia selama tiga bulan, ada empat projek yang dikerjakan yaitu,

1. *Single Indentity Database* (SID) dan *Be GeMS Auto Tracking System* (BEATS)

Projek ini adalah projek pertama yang diberikan oleh PT Volante Teknologi Indonesia di awal periode kerja magang. Single Identity Database dan Be GeMS Auto Tracking System ini adalah sebuah projek dari PT Berau Coal yang merupakan salah satu perusahaan pertambangan milik Sinarmas Group. Pada perusahaan PT Berau Coal masih banyak beberapa hal yang bersifat manual seperti, pengisian formulir pelaporan seperti inspeksi, insiden, dan observasi masih bersifat manual dan pembuatan kartu lisensi tambang masih tidak terstruktur yang membuat satu orang harus membawa banyak kartu.

Dari permasalahan tersebut maka projek Single Identity Database memiliki tujuan utama yaitu untuk menyatukan banyak kartu tersebut sehingga satu orang hanya perlu membawa satu kartu saja dan membuat pendataan absensi pekerja. Tujuan lain dari projek itu adalah melihat performa pekerja, mendata pelanggaran yang dilakukan pekerja, mendata asset, dan lain – lain.

Projek Be GeMS Auto Tracking System berbeda dengan Single Identity Database, projek ini bertujuan untuk membuat pelaporan *inspection*, *observation*, *incident investigation*, dan *hazard report* menjadi mudah dengan diinput melalui sistem. Sehingga perusahaan bisa dengan cepat melakukan tindakan perbaikan (*corrective action*). Kedua projek ini akan dibuat menjadi satu kesatuan sistem dengan menu yang berbeda. Tujuan dari projek Single Identity Database dan Be GeMS Auto Tracking System digabung adalah

memudahkan pengguna dapat menggunakan kedua hal tersebut tanpa harus berganti – ganti sistem.

2. *Trading Station*

Projek ini dipercayakan oleh *client* untuk dibuatkan oleh PT Volante Teknologi Indonesia. Projek ini memiliki tujuan untuk mempersatukan para *trader* yang ada di Indonesia yang bersifat *web-based* dan *mobile*. Fitur – fitur yang dimiliki oleh Trading Station antara lain, posting, eChanger, mempersatukan broker – broker yang ada di Indonesia, *copy trade* orang lain, admin area, dan *marketplace* video tutorial, ebook, merchandise, dan apapun yang berkaitan dengan Forex.

3. *Operating Execution / Monitoring Management* (OE/MM)

Projek ketiga diberikan pada hari terakhir sebelum kontrak selesai. Projek ini diberikan oleh PT Berau Coal yang merupakan anak perusahaan dari Sinarmas Group. Projek ini memiliki tujuan utama untuk melakukan otomatisasi terhadap pekerjaan tambang mulai dari area tambang sampai ke tongkang khususnya dalam menghitung berat hasil tambang yang dibawa oleh truk tambang. Otomatisasi perhitungan tersebut akan disatukan dengan sistem jembatan timbang (*weighbridge*). Selain untuk otomatisasi *weighbridge* (*weighbridge automation*) ada beberapa fitur tambahan pada projek ini seperti *remote monitoring*, *fleet management*, *helicopter view system*, dan *interoperability*.

3.3 Uraian Pelaksanaan Kerja Magang

Dalam melaksanaan kerja magang, ada beberapa hal yang perlu dipersiapkan yang berguna untuk kelangsungan kerja magang. Pertama, mempelajari Angular JS yang merupakan sebuah *framework javascript* yang dirilis oleh Google. Angular JS dipelajari melalui media *e-learning* CodeAcademy (codecademy.com/learn/learn-angularjs).

Kedua, melakukan pendaftaran pada *website* Git Lab (gitlab.com) yang berguna untuk memulai melakukan pengembangan sistem yang sedang dikembangkan PT Volante Teknologi Indonesia. Setelah mendaftar selanjutnya diundang ke dalam grup volantech, selaku grup perusahaan yang berisi semua projek yang sedang maupun telah dikembangkan oleh perusahaan.

Ketiga, mengunduh Meteor (meteor.com/install) dan Git (git-scm.com/downloads). Meteor merupakan salah satu *framework javascript* yang digunakan oleh PT Volante Teknologi Indonesia untuk mengembangkan sistem sistem yang bersifat *real time*. Sedangkan Git berguna sebagai *version control system* yang dipakai para *developer* di PT Volante Teknologi Indonesia untuk menarik sistem melalui Git Lab dan mengembangkan sistem secara bersama – sama.

Keempat, mendaftar dan menggunakan sistem Slack (slack.com) dan Trello (trello.com) yang berguna sebagai sarana komunikasi antar individu di perusahaan. Setelah mengunduh sistem tersebut, selanjutnya diundang oleh pembimbing

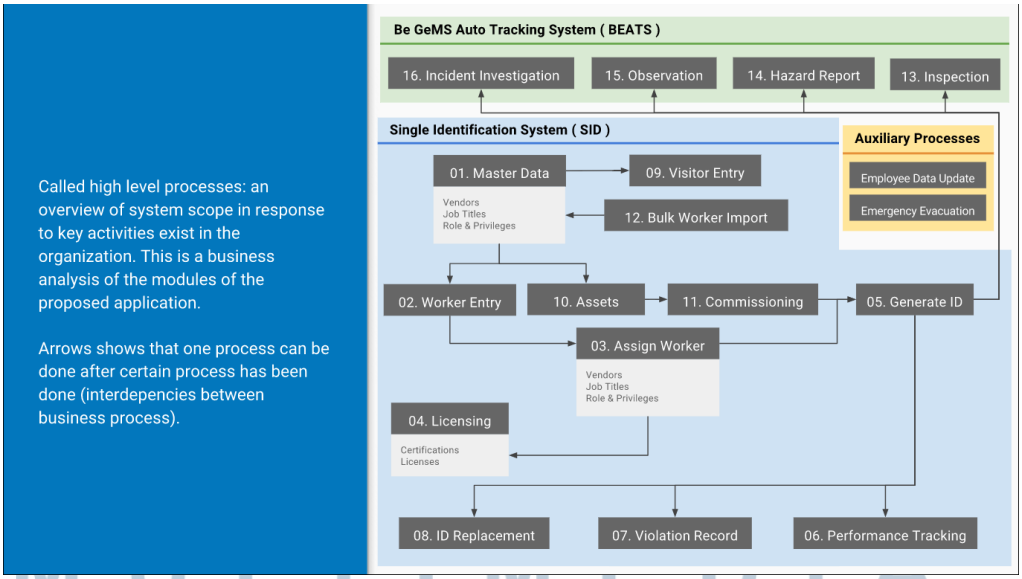
magang untuk masuk ke dalam grup atau *channel* perusahaan tersebut sehingga langsung dapat berkomunikasi dengan lebih baik.

Kelima, diberikan penjelasan dan contoh mengenai *work flow diagram*, *use case diagram*, dan *flow chart* yang akan dibuat agar mendapatkan gambaran.

3.3.1 Single Indentity Database (SID) dan Be GeMS Auto Tracking System (BEATS)

Hal pertama yang ditugaskan adalah melakukan analisa terhadap *user requirement* yang diberikan. Setelah dianalisa lalu dibuatkan *work flow diagram*, *use case diagram*, *flow chart*, dan *data structure* dari sistem *Single Identity Database* dan *Be GeMS Auto Tracking System*. Berikut adalah hasil dari analisa *user requirement* yang telah diberikan,

1. High Level Process



Gambar 3.1 High Level Process SID dan BEATS

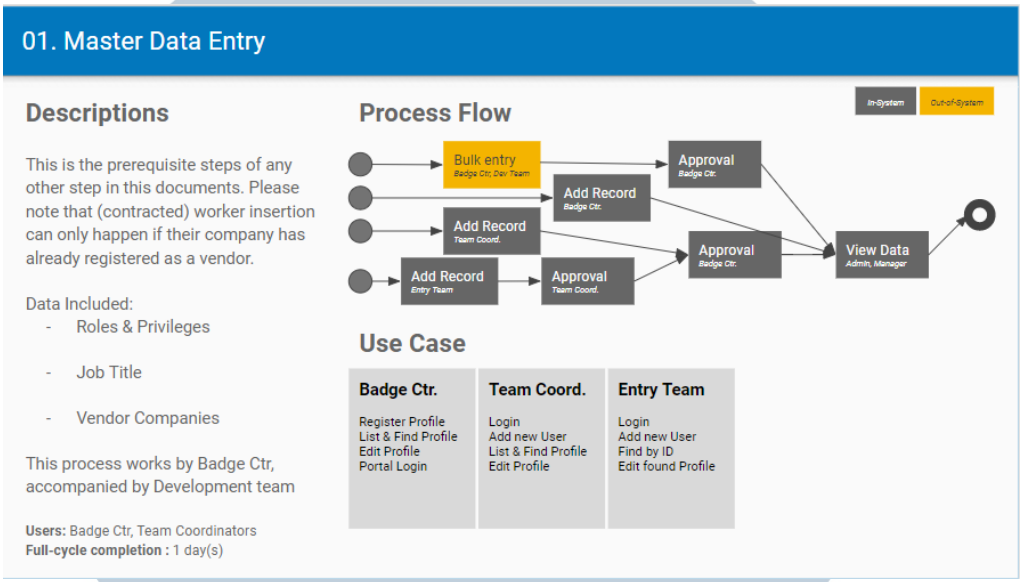
Gambar 3.1 menjelaskan secara garis besar kerja sistem *Single Identity Database* dan *Be GeMS Auto Tracking System*. Sistem ini akan menggunakan *database MongoDB*. Keputusan untuk menggunakan *MongoDB* sebagai *database* sistem dikarenakan *MongoDB* memiliki performa yang lebih cepat daripada *MySQL* dan *MongoDB* dapat mengoptimalisasi penggunaan *database* dengan memecahnya menjadi beberapa bagian (*auto-sharding*).

Single Identity Database memiliki dua belas modul yang memiliki proses berbeda satu sama lain. Modul – modul tersebut adalah *Master Data, Worker Entry, Assign Worker, Licensing, Generate ID, Performance Tracking, Violation Record, ID Replacement, Visitor Entry, Assets, Commissioning,* dan *Bulk Worker Import*.

Be GeMS Auto Tracking System memiliki modul yang lebih sedikit daripada *Single Identity Database*. Hal ini yang menjadi salah satu alasan dalam memutuskan untuk menggabung sistem *Single Indentity Database* dan *Be GeMS Auto Tracking System*. Modul pada *Be GeMS Auto Tracking System* ada empat, yaitu *Inspection, Observation, Hazard Report,* dan *Incident Investigation.*

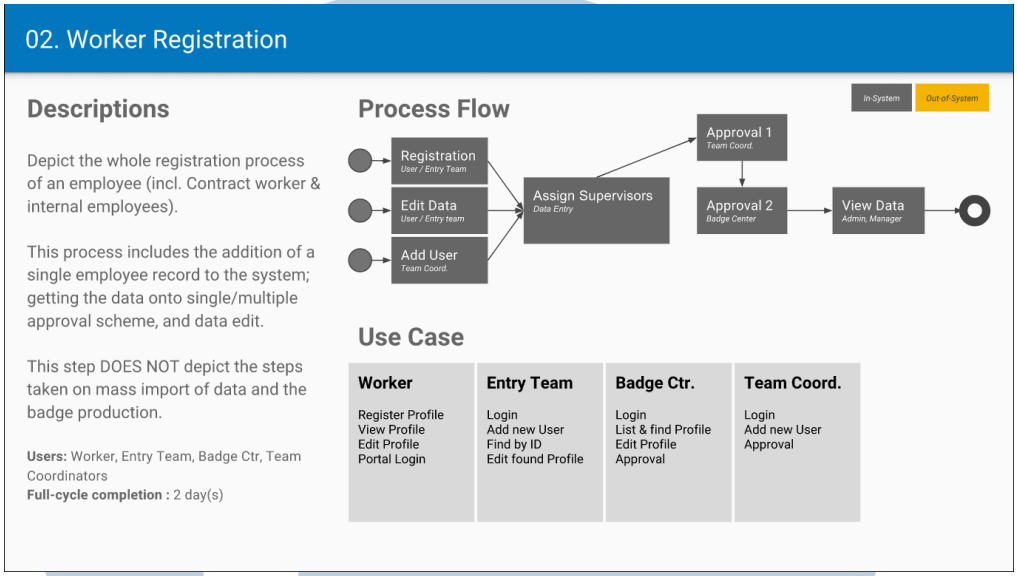
Selain modul – modul tersebut, terdapat juga modul bantuan yang disediakan pada sistem ini. Ada dua buah modul bantuan pada sistem ini, yaitu *Emergency Evacuation* dan *Employee Data Update*.

2. Use Case dan Process Flow



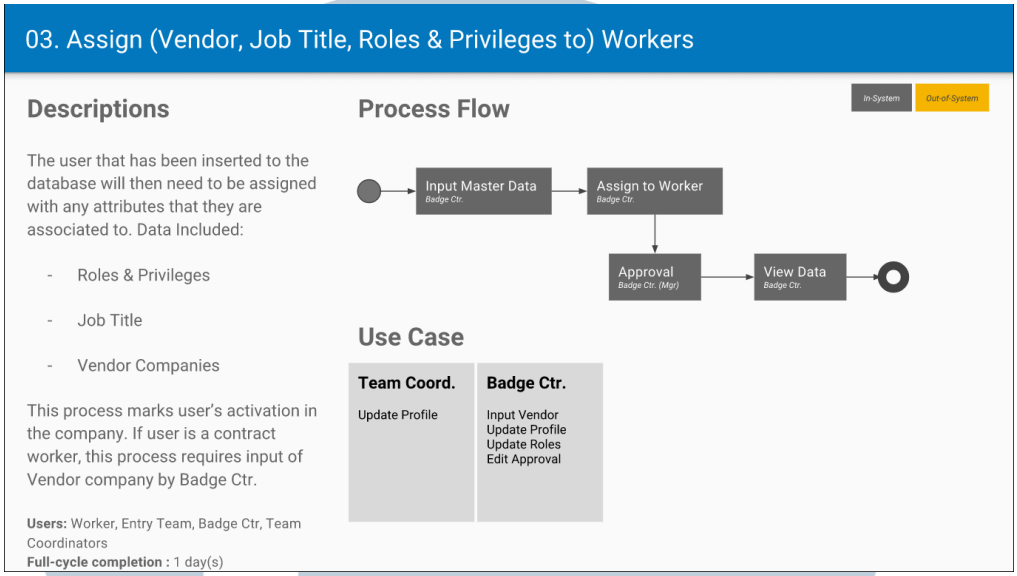
Gambar 3.2 Master Data Entry SID dan BEATS

Proses pada Gambar 3.2 ini menjelaskan alur dari memasukkan data ke dalam *master data*. Ada empat cara untuk memasukkan *record* ke *master data*. Pertama, dengan memasukkan data dalam jumlah besar (*bulk entry)* melalui *developer* yang harus disetujui oleh orang dari PT Berau Coal yang akan mengurus sistem ini (*badge center*). Kedua, di *input* oleh *badge center* itu sendiri. Ketiga, ketua dari setiap regu (*team coordinator*) menginput *record* tersebut yang harus disetujui oleh *badge center*. Keempat, PT Berau Coal akan menyewa seseorang untuk melakukan menginput data ke dalam sistem yang harus disetujui oleh dua pihak yaitu *team coordinator* dan *badge center*. Setelah disetujui maka data yang sudah diinput akan dapat diperlihatkan di dalam sistem.



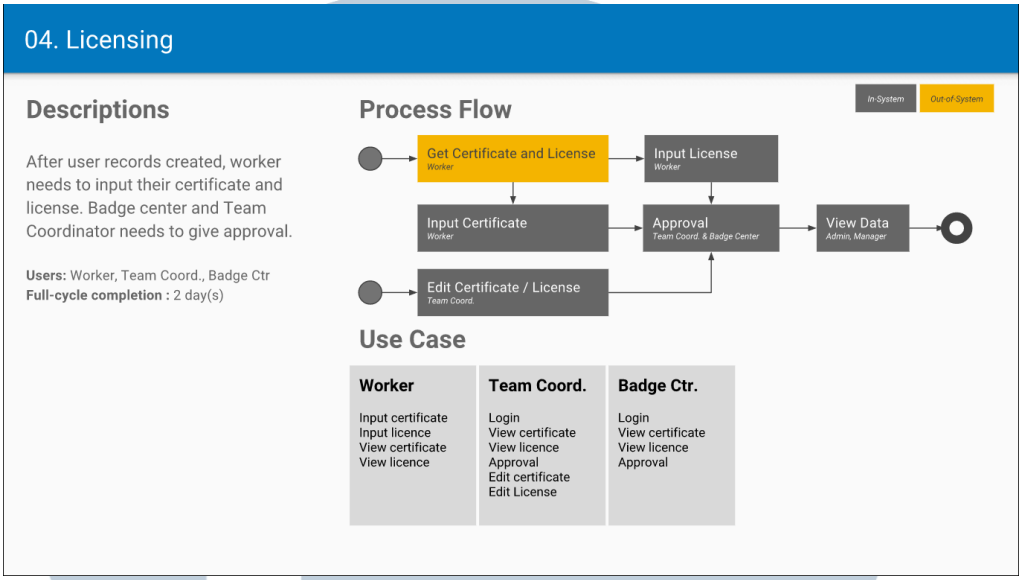
Gambar 3.3 Worker Registration SID dan BEATS

Worker Registration pada Gambar 3.3 memperlihatkan alur yang dilakukan oleh user atau pekerja dalam melakukan registrasi ke dalam sistem. Ada dua cara registrasi ke dalam sistem, pertama pekerja melakukan registrasi secara langsung melalui sistem yang harus disetujui oleh *team coordinator* dan *badge center*. Kedua, registrasi tersebut juga bisa dibantu oleh *team coordinator* dengan cara mengundang pekerja untuk masuk ke dalam sistem melalui sistem. Setelah diundang maka sistem akan mengirimkan email kepada pekerja untuk bergabung ke dalam sistem tersebut.



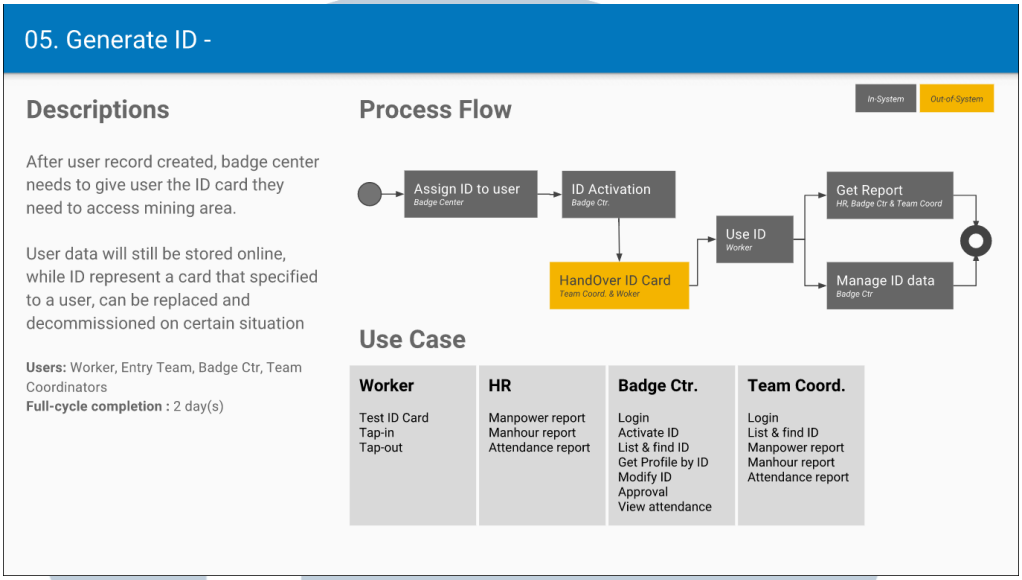
Gambar 3.4 Assign Workers SID dan BEATS

Gambar 3.4 menunjukan, setelah pekerja sudah melakukan registrasi ke dalam sistem maka pekerja tersebut harus ditentukan terlebih dahulu pekerjaan dan hak yang didapatkan oleh pekerja dalam sistem tersebut. Penentuan pekerjaan dan hak yang didapatkan di dalam sistem akan ditentukan oleh *badge center* dan akan disetujui oleh manajer dari *badge center*. Persetujuan hanya melalui manajer dari *badge center* berguna untuk meminimalisir kesalahan. Jika ada pekerja yang bukan pekerja tetap melainkan pekerja kontrak maka *badge center* juga harus menentukan perusahaan atau *vendor* pekerja kontrak itu berasal.



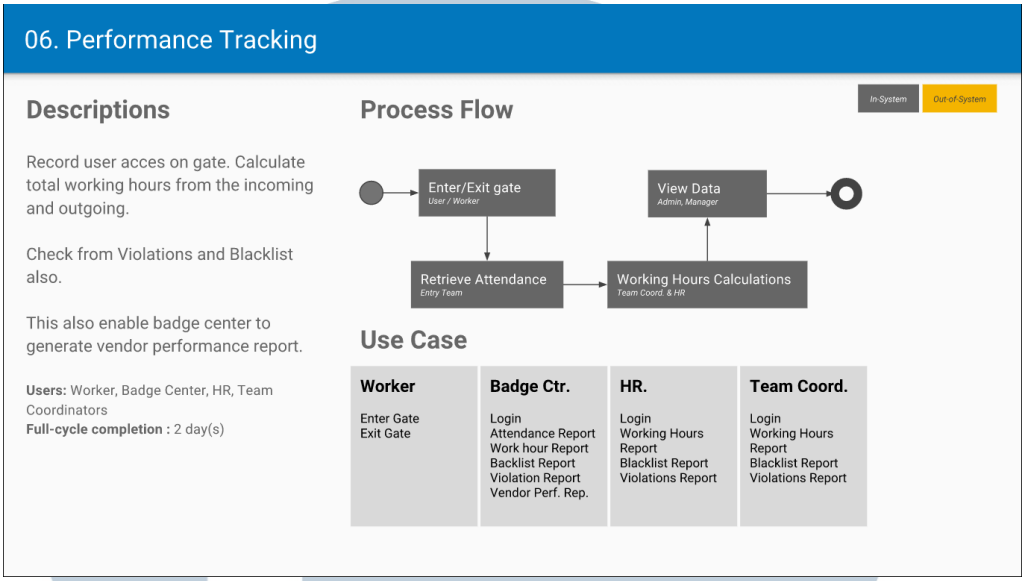
Gambar 3.5 Licensing SID dan BEATS

Licensing pada Gambar 3.5 diperlukan jika pekerja memerlukan izin untuk beroperasi di pertambangan. Contoh dari izin tersebut adalah SIMPER (Surat Izin Mengoperasikan Unit Perusahaan) yang diperlukan jika ingin mengemudikan kendaraan pertambangan. Cara kerja di sistem adalah sebagai berikut, pekerja memasukan informasi – informasi dari lisensi atau sertifikat yang telah dimiliki oleh pekerja ke dalam sistem. Lalu *team coordinator* dan *badge center* harus menyetujui bahwa pekerja sudah benar memasukan informasi dari lisensi atau sertifikat tersebut sehingga pekerja dapat menggunakannya di pertambangan.



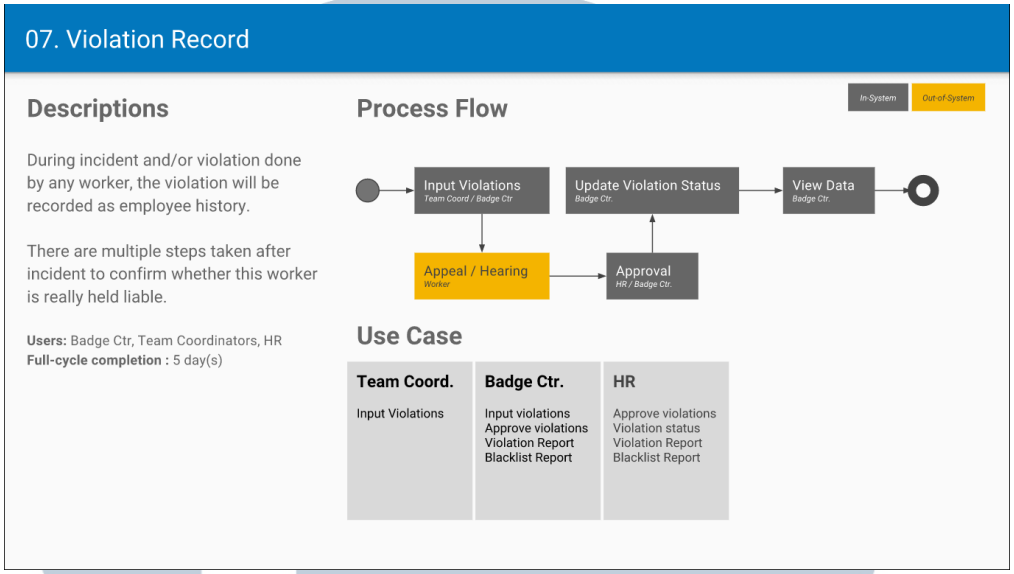
Gambar 3.6 Generate ID SID dan BEATS

Gambar 3.6 menjelaskan proses pembuatan kartu indentitas (*ID Card*) yang dapat dilakukan pada sistem ini, dimulai dari proses pembuatan membuat nomor unik setiap pekerja dan tempat untuk melakukan print kartu tersebut. Setiap *ID Card* yang sudah dibuat pun bisa dikelola, digantikan dan jika masa aktivasi sudah habis *ID Card* tersebut juga bisa dimatikan. Data penggunaan setiap *ID Card* yang dipakai oleh para pekerja juga akan disimpan dalam sistem ini sehingga akan menghasilkan laporan mengenai pemakaian yang dapat dilihat oleh departemen sumber daya manusia (*human resources*), *badge center*, dan juga *team coordinator*.



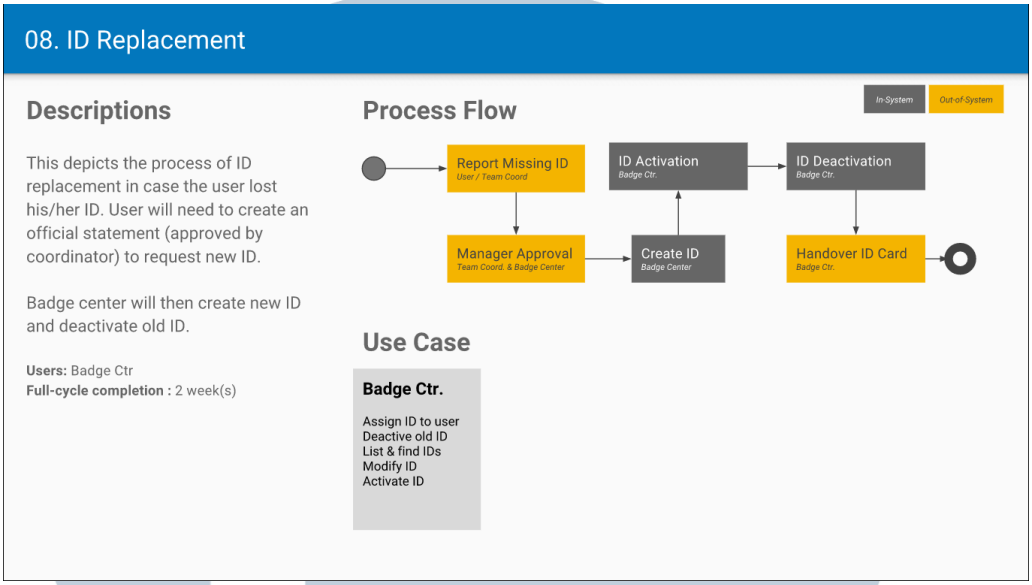
Gambar 3.7 Performance Tracking SID dan BEATS

Proses dari Gambar 3.7 adalah kehadiran pekerja pada area tambang yang dijadikan sebagai acuan lama pekerja bekerja di area tambang. Kegunaan dari proses ini adalah untuk melihat performa atau kinerja pekerja dalam area tambang. Setelah itu kehadiran tersebut akan dikalkulasikan untuk mendapatkan jam kerja pegawai yang nantinya dapat digunakan untuk pembayaran gaji pekerja maupun laporan perusahaan.



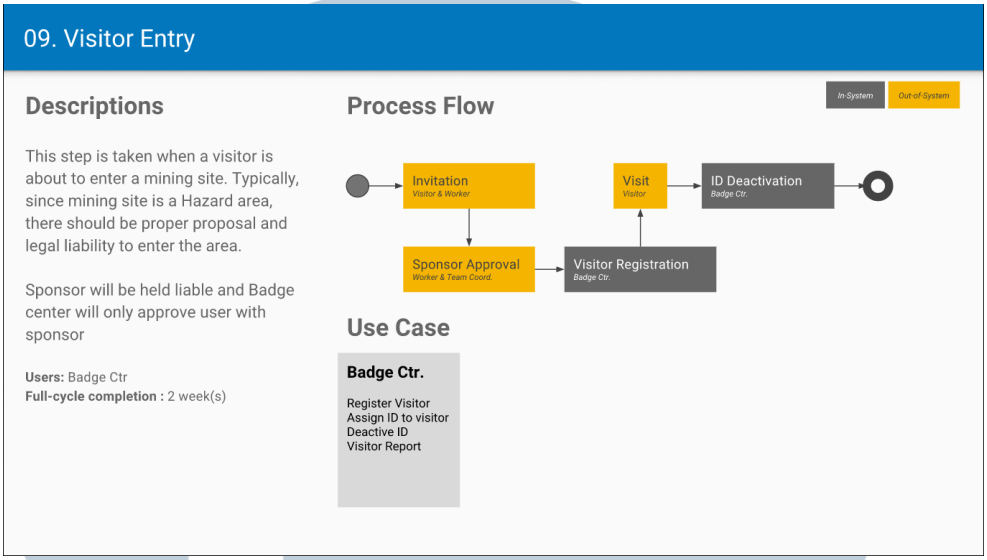
Gambar 3.8 Violation Record SID dan BEATS

Proses pada Gambar 3.8 ini *team coordinator* akan memasukan pelanggaran – pelanggaran yang telah dilakukan oleh para pekerja dalam pekerjaannya dan akan disetujui oleh *human resources* dan *badge center*. Pelanggaran tersebut akan masuk ke dalam riwayat pekerja dalam perusahaan tersebut. Dari hasil pelanggaran tersebut perusahaan juga melihat jika pekerja tersebut untuk menilai pekerja tersebut benar – benar bertanggung jawab dalam pekerjaannya atau tidak.



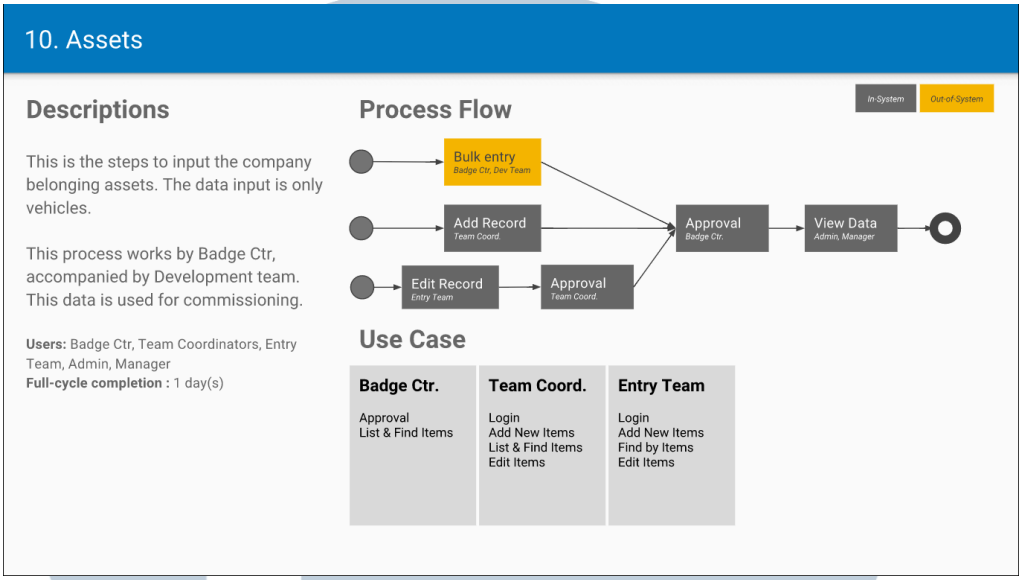
Gambar 3.9 ID Replacement SID dan BEATS

Seperti yang telah dijelaskan sebelumnya jika ID Card itu hilang dapat diganti dengan ID Card yang baru. Gambar 3.9 ini adalah proses ID Replacement yang bertujuan untuk meminta persetujuan pembuatan ID Card baru yang dilakukan secara manual baik ke team coordinator maupun kepada badge center. Setelah itu badge center akan membuatkan ID Card yang baru dan akan menonaktifkan ID Card yang lama.



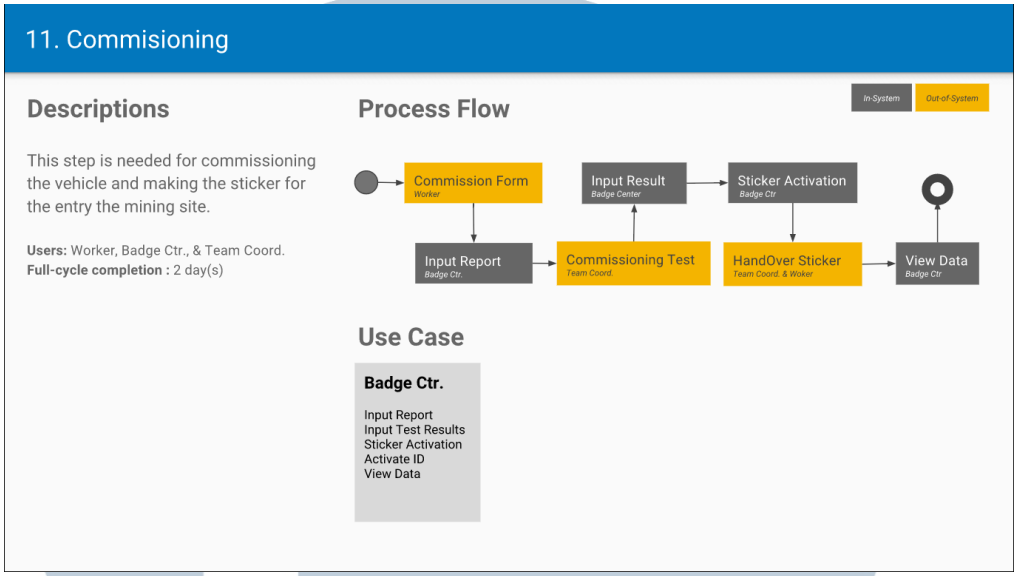
Gambar 3.10 Visitor Entry SID dan BEATS

Pada Gambar 3.10 ini jika pengunjung yang memiliki undangan juga akan dibuatkan akses *ID Card* untuk memasuki area pertambangan yang pembuatan *ID Card* nya juga akan diakomodasikan oleh sistem ini. Proses dari *Visitor Entry* tersebut hampir serupa dengan *Generate ID*, yang membedakannya adalah setelah pengunjung itu telah selesai berkunjung maka *ID Card* yang telah dibuat tadi pun akan dinonaktifkan untuk keamanan sehingga tidak dapat digunakan kembali.



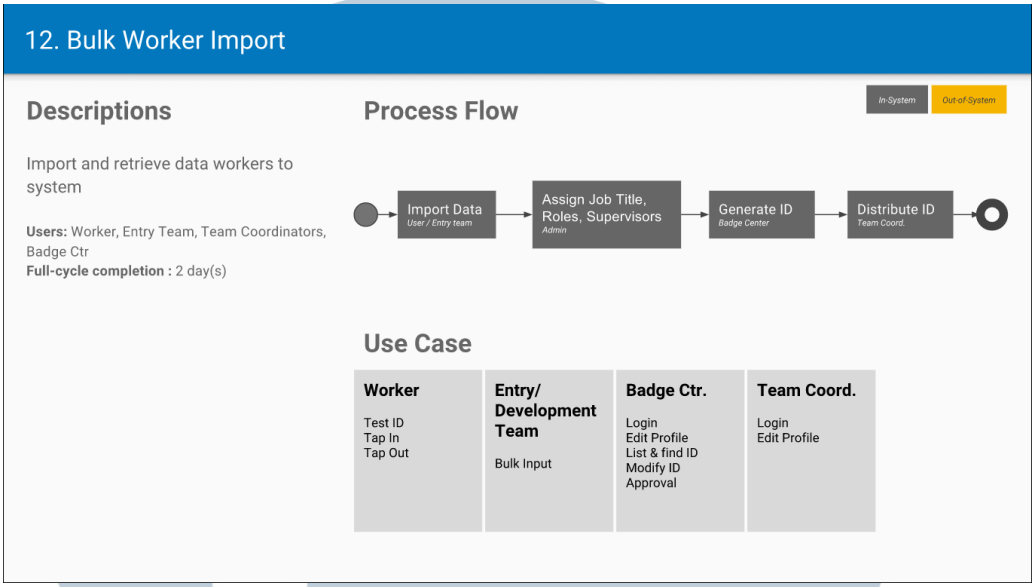
Gambar 3.11 Assets SID dan BEATS

Gambar 3.11 menjelaskan dapat dilakukannya pendataan setiap aset juga ada pada sistem ini. Kebanyakan aset yang didata pada sistem ini adalah kendaraan karena berhubungan dengan modul lisensi. Aset tersebut sebelumnya harus disetujui oleh *badge center* dahulu sebelum pengguna sistem dapat melihat semua data aset tersebut.



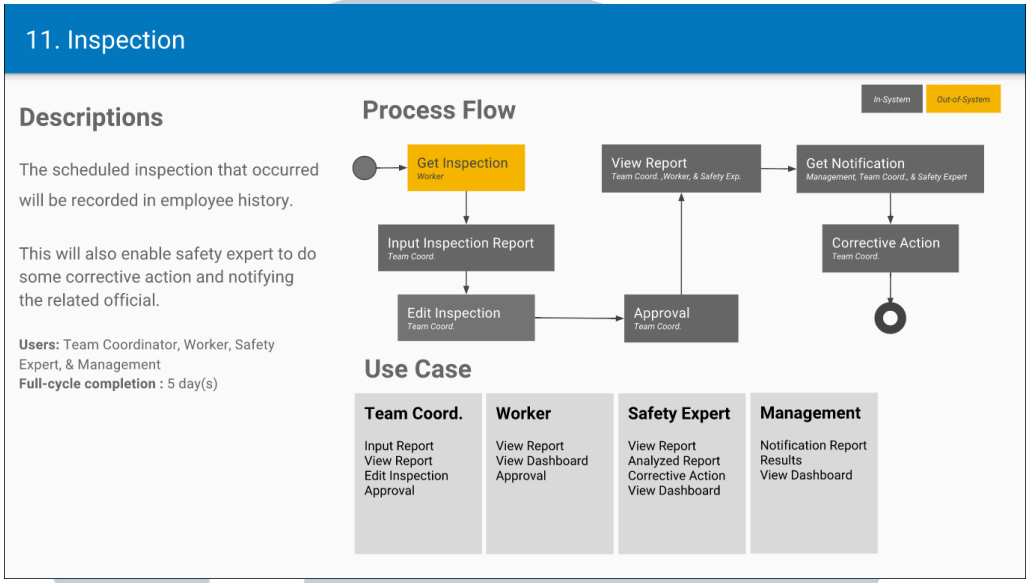
Gambar 3.12 Commissioning SID dan BEATS

Commissioning adalah serangkaian tahap pemeriksaan terhadap barang yang akan dioperasikan oleh pengguna. Proses *commissioning* pada Gambar 3.12 ini dimaksudkan untuk *commissioning* kendaraan atau aset yang dimiliki perusahaan. Pendataan pada sistem memudahkan pelaporan hasil *commissioning* yang telah dilakukan oleh pihak berwenang. Jika kendaraan tersebut sudah lolos dari tes *commissioning* ini maka badge center akan membuatkan sticker pada mobil sebagai bukti sudah lolos tes.



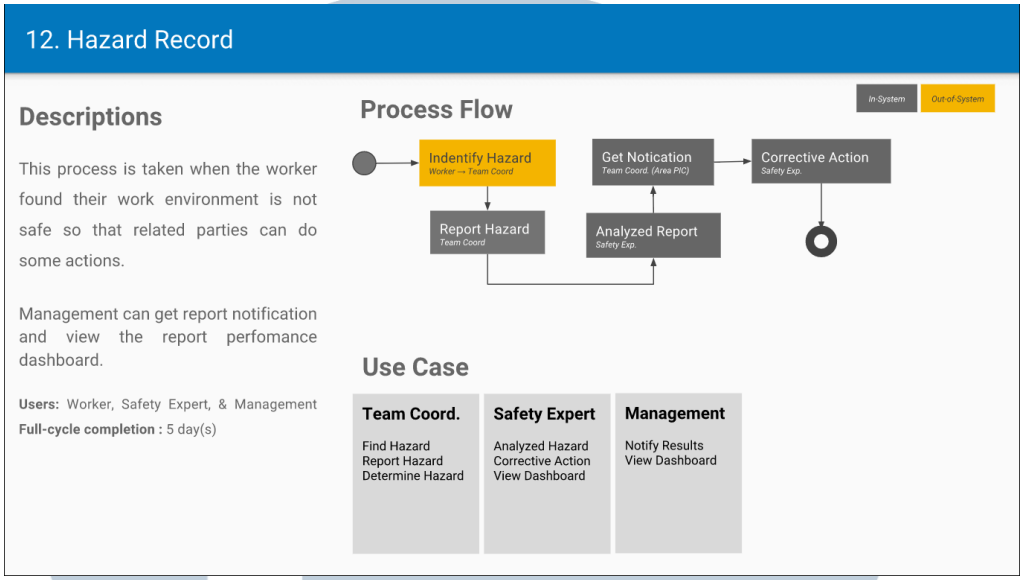
Gambar 3.13 Bulk Worker Import SID dan BEATS

Proses ini diperlukan pada saat memasukan data ke dalam sistem dalam jumlah besar. Biasanya digunakan untuk perpindahaan dari sistem yang lama ke sistem yang baru. Pada Gambar 3.13 diperlihatkan bahwa setelah data dimasukkan dapat langsung ditentukan hak, peran, *supervisor*, dan lain – lain. Setelah kedua proses itu selesai maka sistem dapat secara langsung membuatkan *ID Card* dan *ID Card* tersebut pun dapat dengan cepat didistribusikan dan digunakan secara langsung.



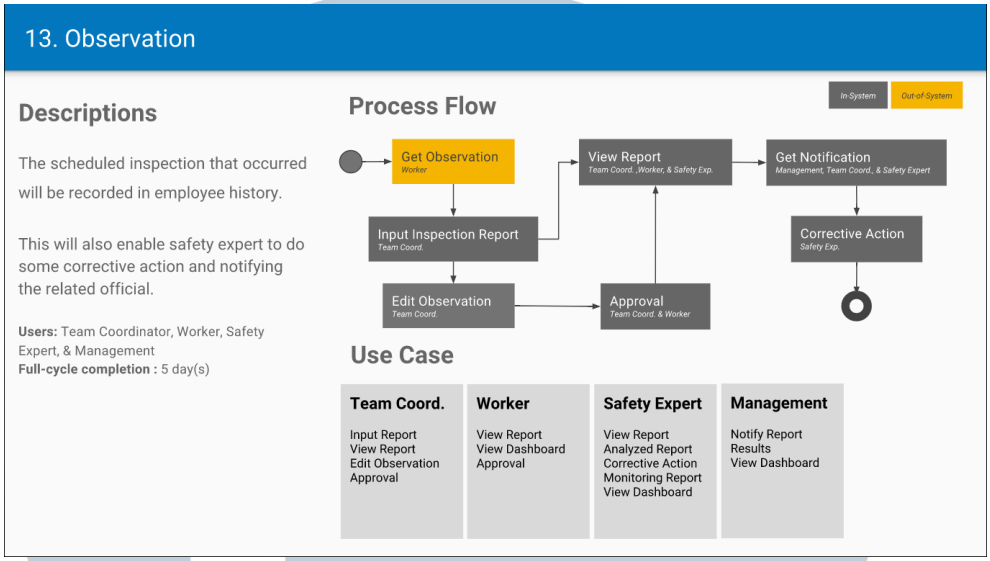
Gambar 3.14 Inspection SID dan BEATS

Pada Gambar 3.14 menunjukkan cara kerja proses *inspection* pada sistem ini. Setelah pihak berwenang melakukan inspeksi, Ia harus menginput hasil dari inspeksi tersebut ke dalam sistem. Hasil dari inspeksi itu akan dimasukkan sebagai riwayat pekerja dan akan memberi tahu *management*, *safety expert*, dan *team coordinator* bahwa inspeksi telah selesai dan sudah dapat dilihat hasilnya. Jika ada sesuatu masalah dalam inspeksi tersebut maka *safety expert* yang akan melakukan tindakan perbaikan terhadap masalah tersebut.



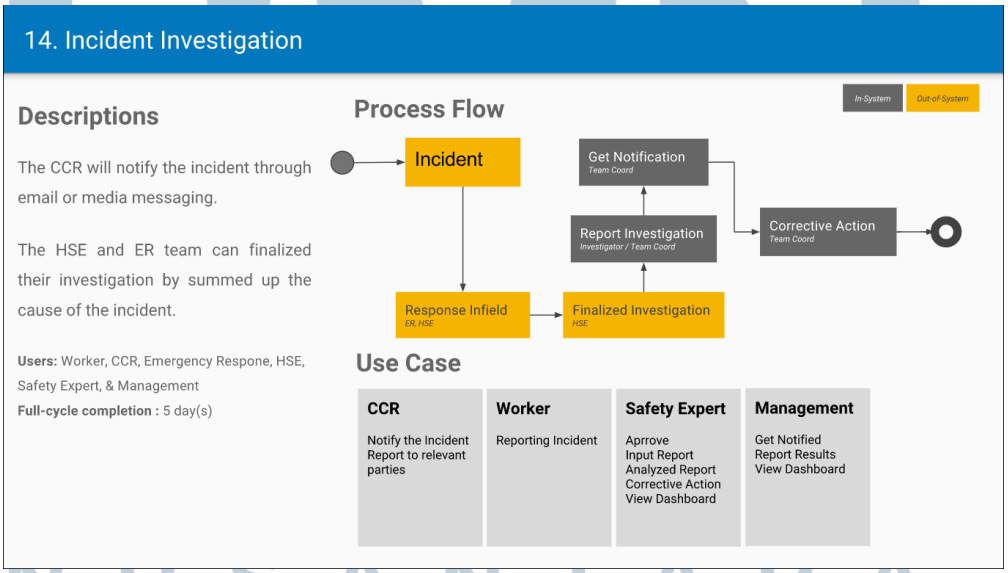
Gambar 3.15 Hazard Report SID dan BEATS

Kegunaan dari *hazard report* ini adalah untuk para pekerja melaporkan bahaya yang ditemukan di daerah tambang yang berpotensi menyebabkan kecelakaan. Gambar 3.15 menggambarkan cara pelaporan dari bahaya tersebut ke dalam sistem. Setelah pekerja melaporkan bahaya ke dalam sistem maka pihak berwenang yaitu *safety expert* akan memproses untuk melakukan *corrective action* yang diperlukan.



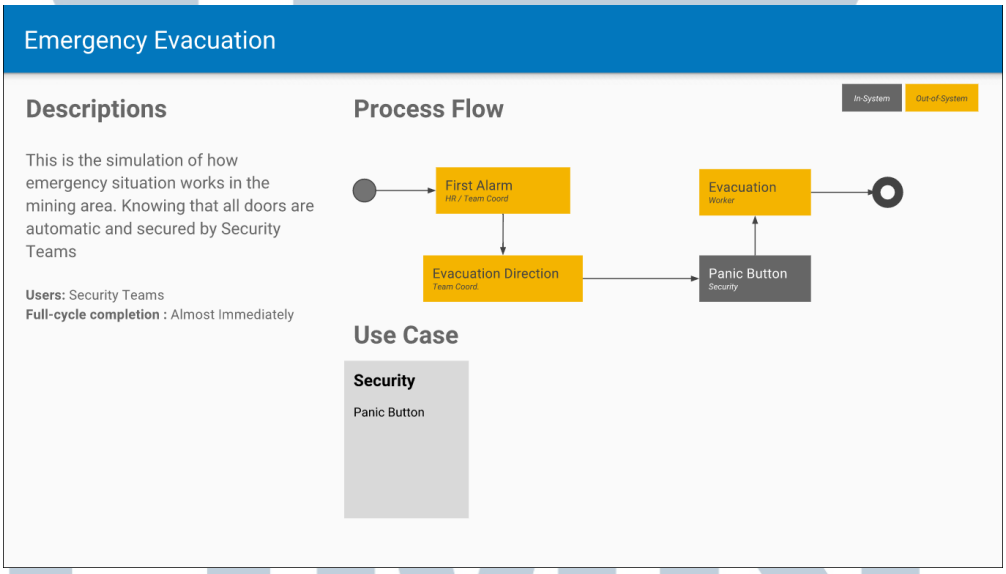
Gambar 3.16 Observation SID dan BEATS

Gambar 3.16 menejelaskan tahapan dalam pelaporan *observation* pada sistem ini serupa dengan tahap melakukan pelaporan *inspection*. Jika ada yang ingin mengedit hasil dari laporan tersebut maka diperlukan persetujuan dari dua pihak yaitu *team coordinator* dan pekerja itu sendiri.



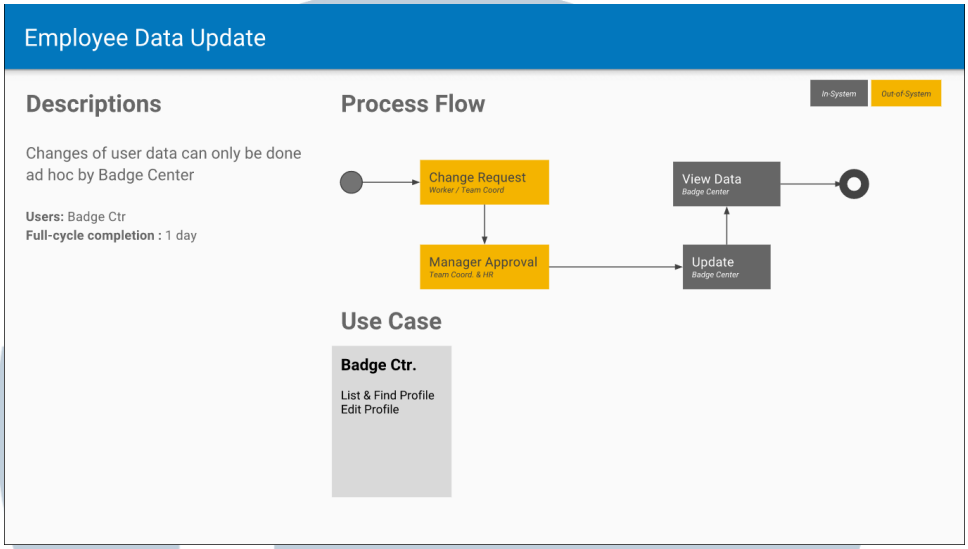
Gambar 3.17 Incident Investigation SID dan BEATS

Proses pada Gambar 3.17 menunjukan jika terjadi insiden pada area tambang maka sistem ini menyediakan modul untuk melakukan pelaporan tersebut pada modul *Incident Investigation*. Setelah terjadi pelaporan maka *Central Control Room* (CCR) menotifikasikan insiden tersebut kepada pihak yang mengurusinya yaitu *Health Safety and Environment* (HSE). Lalu HSE akan mengatasinya dan melaporkan tindakan yang dilakukan untuk mengatasi insiden tersebut pada sistem ini.



Gambar 3.18 Emergency Evacuation SID dan BEATS

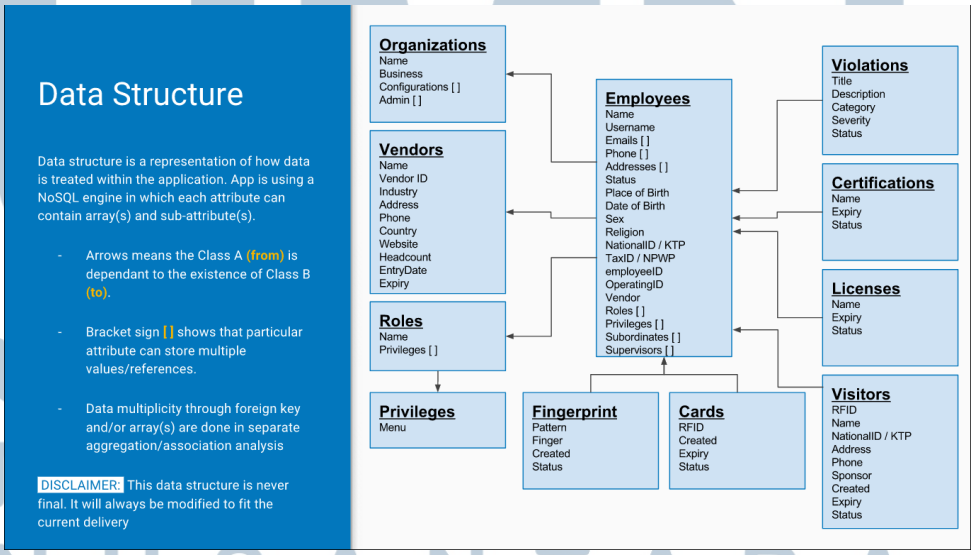
Proses pada Gambar 3.18 ini digunakan jika pada perusahaan tambang ingin mengintegrasikan sistem ini dengan sistem evakuasi darurat milik perusahaan tambang tersebut.



Gambar 3.19 Employee Data Update SID dan BEATS

Badge center mempunyai wewenang untuk mengganti data dari user yang telah mendaftar dalam sistem ini digambarkan pada Gambar 3.19 dengan persetujuan dari *team coordinator* atau *human resources.*

3. Data Structure

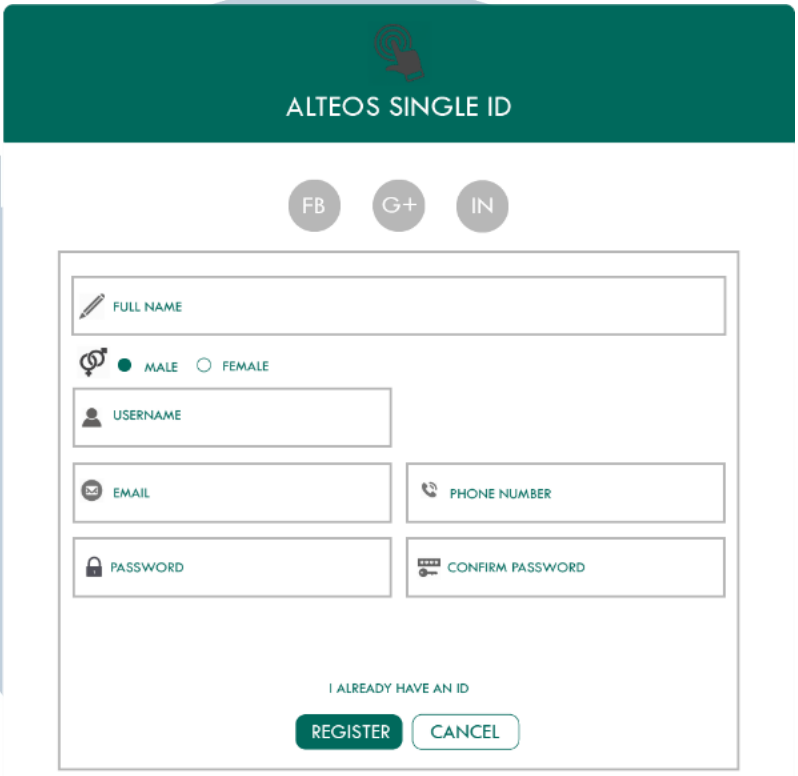


Gambar 3.20 Data Structure SID

Pada Gambar 3.20 diperlihatkan bagaimana struktur data dari *Single Identity Database* tersebut. Struktur data ini memperlihatkan bagaimana data akan diolah dalam database dari sistem ini. Anak panah menunjukan ketergantungan antara *class* pada database. Kurung siku ([ ]) menunjukan bahwa atribut tersebut dapat memiliki *value* yang lebih dari satu (*array*). Struktur data yang dibuat bukanlah struktur data yang final. Ada kemungkinan akan dimodifikasi kembali di masa depannya.

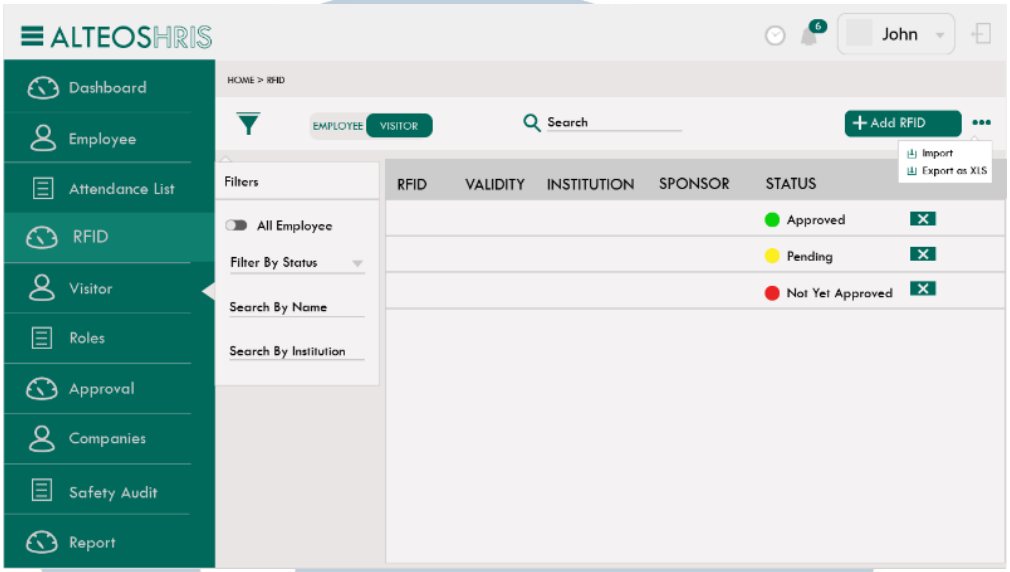
4. Mockup

Sebelum melakukan pengembangan sistem, diperlukan juga rancangan dari sistem yang akan dibuat. Rancangan ini berupa *mockup* yang dapat menjadi acuan sebagai pembuatan sistem. *Mockup* merupakan gambar model sistem secara detail. Tugas selanjutnya yang diberikan adalah pembuatan *mockup* sistem *Single Identity Database* dan *Be GeMS Auto Tracking System*. Pembuatan *mockup* ini akan menggunakan *Adobe Illustrator* dikarenakan sudah mempunyai pengentahuan pada aplikasi tersebut. Berikut adalah beberapa contoh dari *mockup* yang telah dibuat.



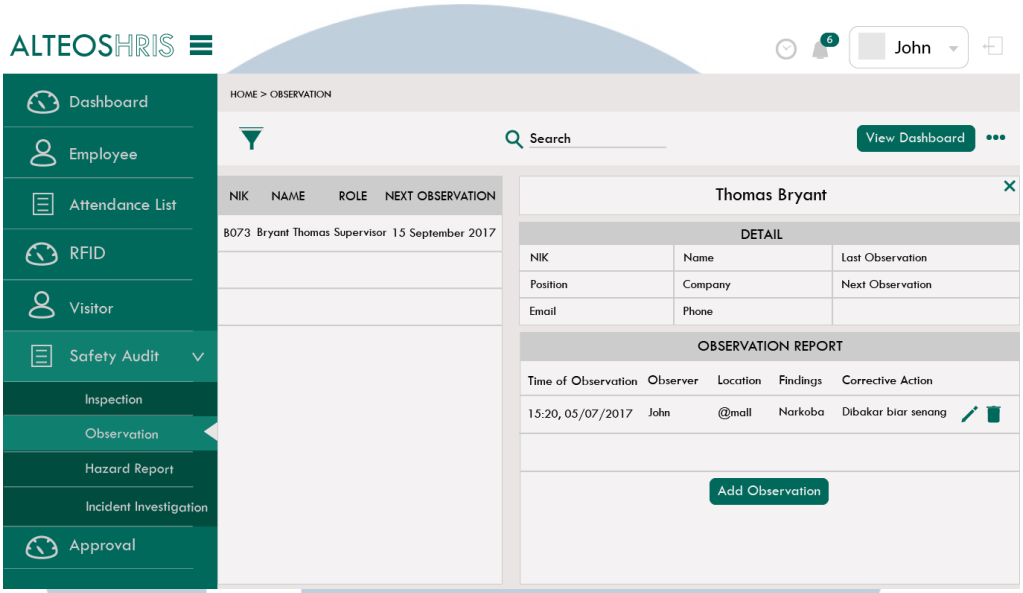
Gambar 3.21 Tampilan Mockup Register

Gambar 3.21 merupakan contoh dari tampilan untuk *user* melakukan registrasi ke dalam sistem. Seperti pada umumnya, *user* memasukan data diri mereka, *username*, dan *password*. *User* dapat menggunakan media sosial mereka melakukan registrasi. Media sosial yang dapat digunakan dalam registrasi disini adalah *Facebook*, *Gmail*, dan *Linked In*.



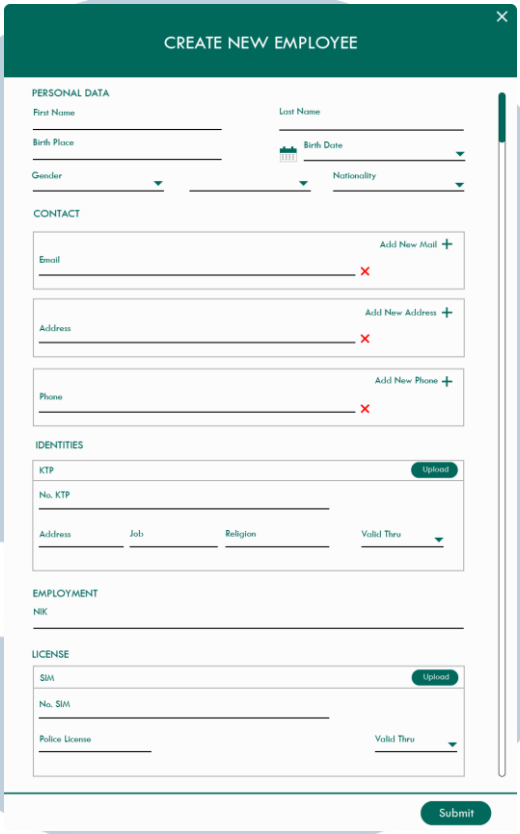
Gambar 3.22 Tampilan Mockup List RFID

Bagian kiri adalah menu utama dari sistem sedangkan bagian kanan adalah isi dari setiap menu tersebut. Di Gambar 3.22 mengambil contoh dari tampilan menu RFID. Pekerja atau *visitor* yang telah mendapatkan RFID akan terdaftar pada menu RFID. *Badge center* juga dapat menonaktifkan RFID pekerja atau *visitor* melalui tombol yang ada bagian kanan.



Gambar 3.23 Tampilan Mockup Observation Details

Sebelumnya merupakan contoh dari tampilan daftar sedangkan pada Gambar 3.23 merupakan tampilan *detail* dari menu *Observation*. Bagian *detail* akan muncul ketika *user* menekan nama pada daftar dan bagian *detail* tersebut akan keluar pada bagian kanan. Tujuan dari ditaruh bagian kanan adalah sehingga memudahkan *user* jika ingin melihat *detail* dari pekerja lain.



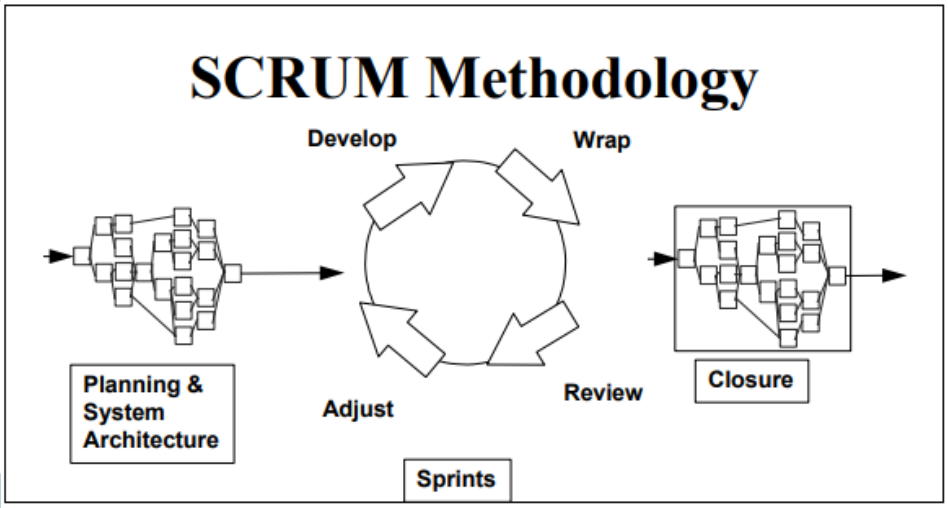
Gambar 3.24 Tampilan Mockup Form Employee

Jika *user* ingin membuat data pekerja yang baru atau mengedit data pekerja yang telah ada, maka *user* akan diberikan tampilan *form*. Gambar 3.24 merupakan contoh dari tampilan *form* tersebut pada menu *Employee*. Setelah selesai mengisi form tersebut maka user hanya perlu menekan tombol *submit* dan hasilnya akan muncul pada tampilan daftar.

5. Pengembangan Modul Licensing

Pengembangan aplikasi ini menggunakan metode SCRUM. SCRUM merupakan kerangka kerja yang digunakan untuk melakukan pengembangan sistem yang kompleks dan menghasilkan

sistem dengan nilai setinggi mungkin dan sekreatif mungkin. Pengembangan metode SCRUM digunakan jika ingin mengembangkan metode SCRUM digunakan untuk mengembangkan sistem dengan batas waktu yang sebentar. Pengembangan metode SCRUM terbagi menjadi tiga aktivitas utama yaitu *Planning* dan *System Architecture*, *Sprints*, dan *Closure*. (Schwaber, 1987).



Gambar 3.25 SCRUM Methodology

Sumber: (Schwaber, 1987)

Tahap *Planning dan System Architecture*, merupakan tahap pertama yang harus dilakukan dalam pengembangan sistem. *Planning* ini dilakukan dengan menganalisis *user requirement* dan estimasi waktu pengerjaan dan biaya yang dibutuhkan. Biasanya tahap *Planning* ini menggunakan *backlog* sebagai daftar kebutuhan *client* tersebut. *System Architecture*, merancang hasil analisa tersebut kedalam *high level process* (Schwaber, 1987). SCRUM

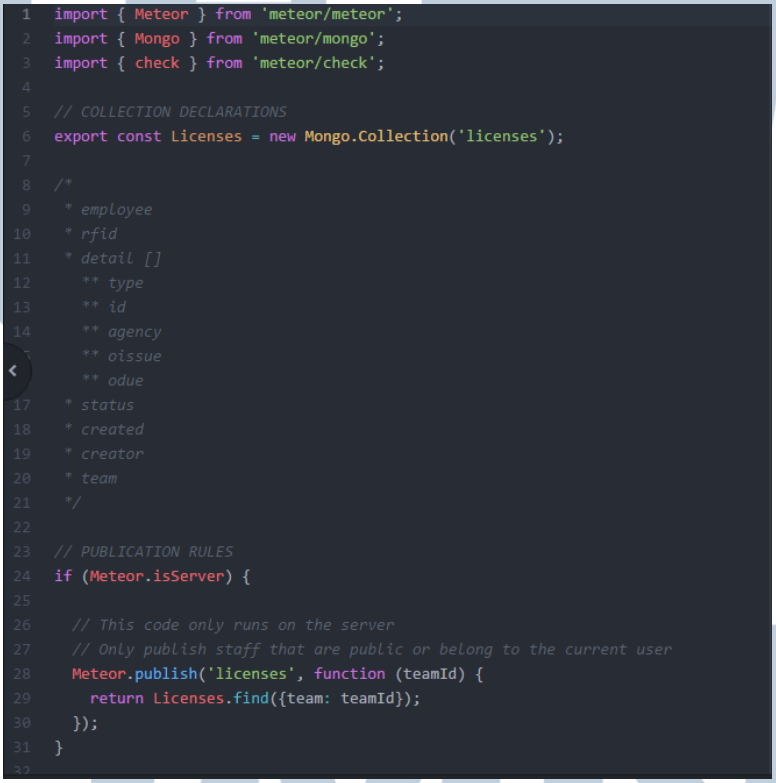
backlog dari PT. Volante Teknologi Indonesia dapat dilihat pada halaman Lampiran.

Tahap *Sprints* berarti pengembangan sistem sudah mulai dilakukan dengan memperhatikan waktu, kualitas, biaya, persyaratan, dan persaingan. Dalam tahap *Sprints*, pengembang melakukan empat kegiatan yaitu, *develop*, *wrap*, *review*, dan *adjust*. Kegiatan tersebut merupakan siklus yang dilakukan secara berulang hingga *backlog* diselesaikan (Schwaber, 1987).

Tahap *Closure* berarti sistem siap untuk diuji atau didemonstrasikan ke *client* yang bersangkutan. Sistem yang sudah siap digunakan harus dibuatkan dokumentasi guna mempermudah *user* dalam mempelajari sistem tersebut (Schwaber, 1987).

Sesuai yang sudah dijelaskan pertama – tama, pengembangan sistem ini menggunakan *framework AngularJS* dan *Meteor*. *AngularJS* digunakan untuk *front end*, sedangkan *Meteor* digunakan untuk *back end*. Pengembangan sistem ini menggunakan arsitektur *Model-View-Controller* (MVC). Model arsitektur MVC membagi pengembangan sistem menjadi tiga lapisan yaitu, meng-enkapsulasi data bersama dengan pemrosesan (*model*), mengisolasi dari proses manipulasi (*controller*) dan tampilan (*view*) untuk direpresentasikan pada sebuah *user interface* (Deacon, 2009).

Model, digunakan untuk mengelola informasi pada *back end.* Hanya model yang mengandung data dan fungsi yang berhubungan dengan pemrosesan data (Burbeck, 1992). Pada modul *Licensing*, pemrosesan data yang ada akan dimasukkan ke dalam *MongoDB.* Hasil dari *model Licensing* dapat dilihat pada Gambar 3.26



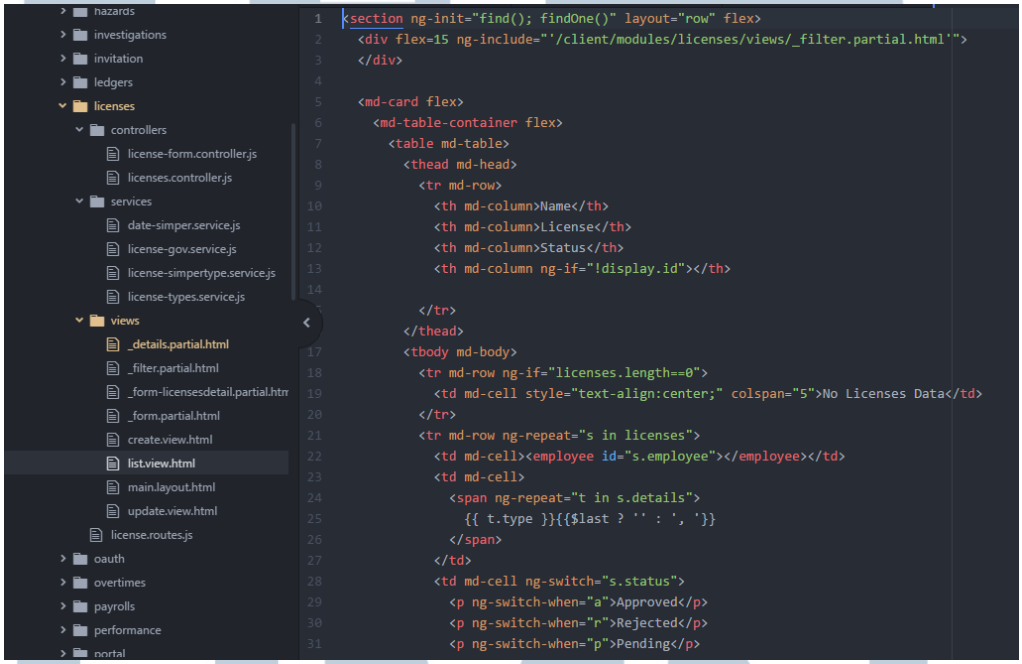
```
import { Meteor } from 'meteor/meteor';
import { Mongo } from 'meteor/mongo';
import { check } from 'meteor/check';

// COLLECTION DECLARATIONS
export const Licenses = new Mongo.Collection('licenses');

/*
 * employee
 * rfid
 * detail []
   ** type
   ** id
   ** agency
   ** oissue
   ** odue
 * status
 * created
 * creator
 * team
 */

// PUBLICATION RULES
if (Meteor.isServer) {

  // This code only runs on the server
  // Only publish staff that are public or belong to the current user
  Meteor.publish('licenses', function (teamId) {
    return Licenses.find({team: teamId});
  });
}
```

Gambar 3.26 Licensing Model

Model pada *Licensing* mengimport *Meteor* dan *Mongo* sama seperti pada modul lainnya. Setelah itu, dilakukan deklarasi *collection* ke *MongoDB* sehingga *MongoDB* dapat mengetahui *collection* yang digunakan pada *Licensing.*

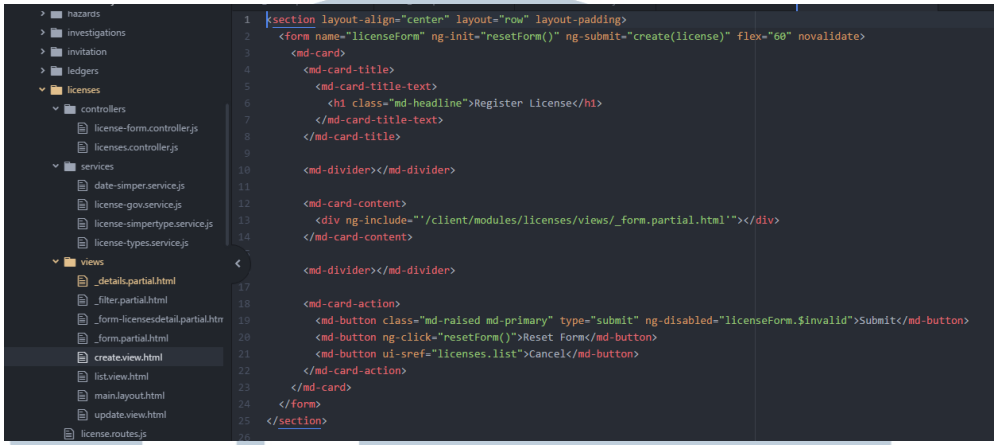
View, merupakan lapisan yang menampilkan *user interface* dari sistem tersebut. Jika *model* berubah maka *view* secara otomatis

akan ikut berubah mengikuti *model* tersebut (Burbeck, 1992). Pada modul Licensing terdapat empat tampilan utama, yaitu *list view*, *create view*, *update view*, dan *detail view*. Keempat tampilan tersebut dibagi ke dalam delapan parsial agar memudahkan pengembangan.



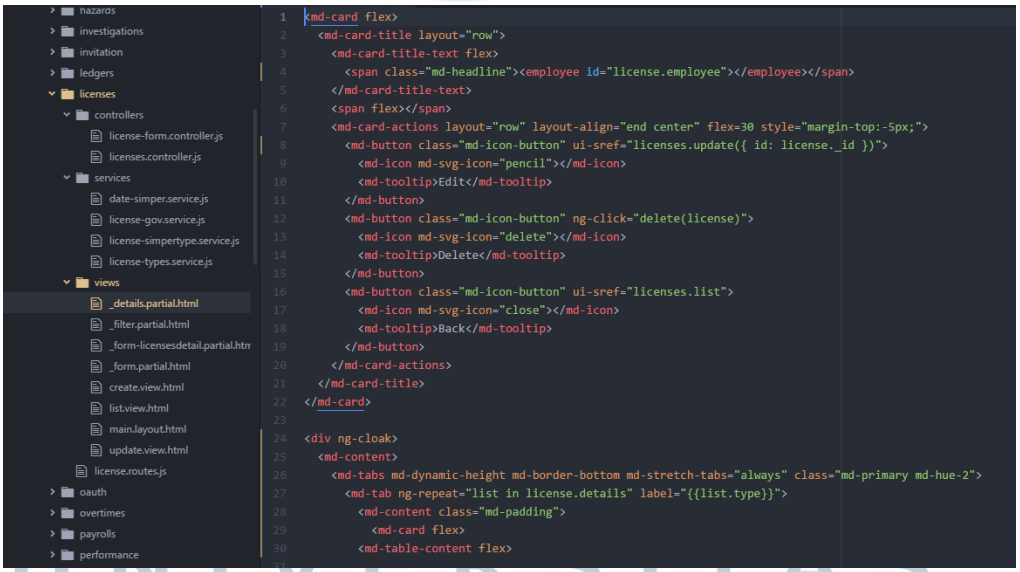
Gambar 3.27 List View Modul Licensing

Gambar 3.27 memperlihatkan *coding* untuk membuatkan daftar dari pekerja yang telah diberikan lisensi pada sistem. Untuk membuat daftar tersebut digunakan tabel (tr & td) sehingga menghasilkan daftar yang mudah dilihat.



Gambar 3.28 Create View Licensing

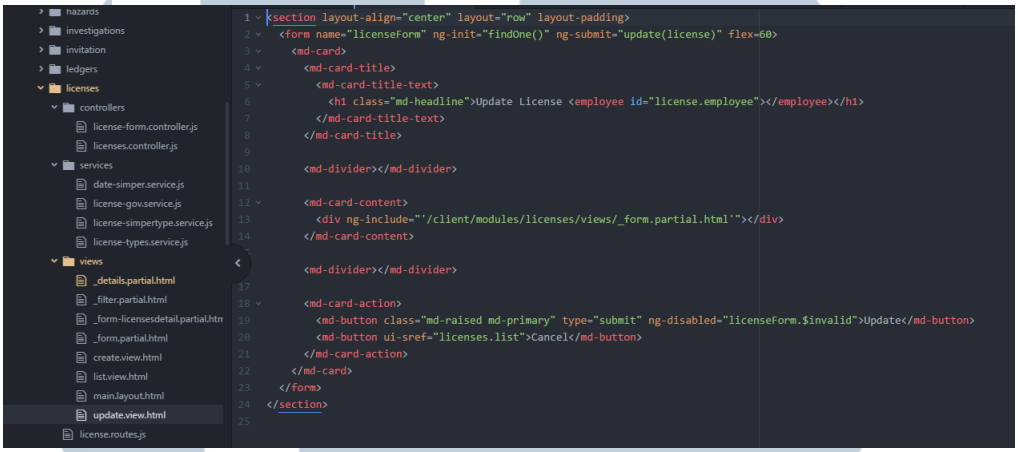
Untuk memberikan lisensi kepada pekerja maka coding disini menggunakan pengisian formulir (*form*). Isi dari *form* tersebut dipisah kedalam *file* baru pada *form partial* yang dapat dilihat pada Gambar 3.28.



Gambar 3.29 Detail View Modul Licensing

Coding di Gambar 3.29 menunjukan jika user ingin melihat detil dari lisensi yang dimiliki oleh pekerja. Adapun *button* untuk

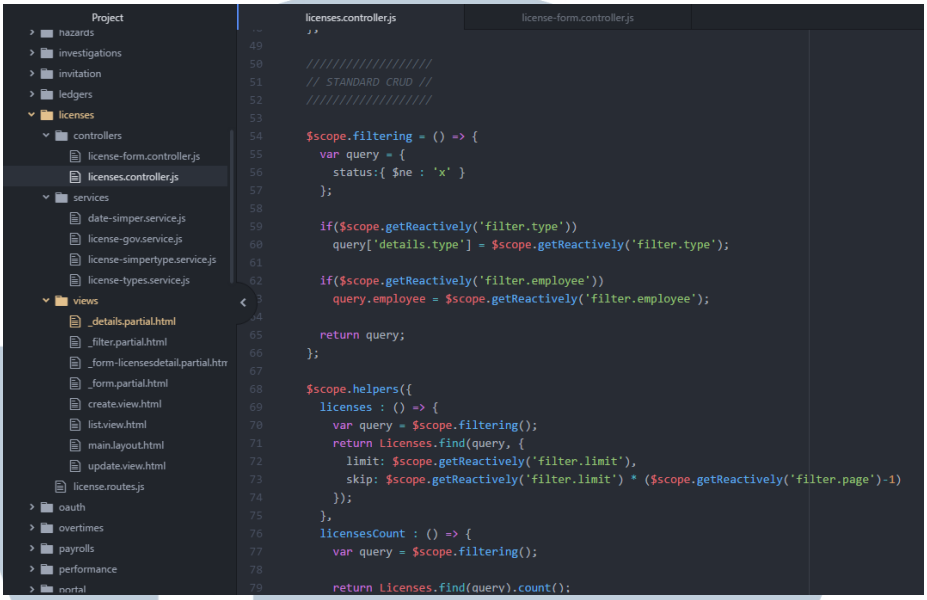
mempercepat proses jika user ingin menghapus atau mengubah detail tersebut.



Gambar 3.30 Update View Modul Licensing

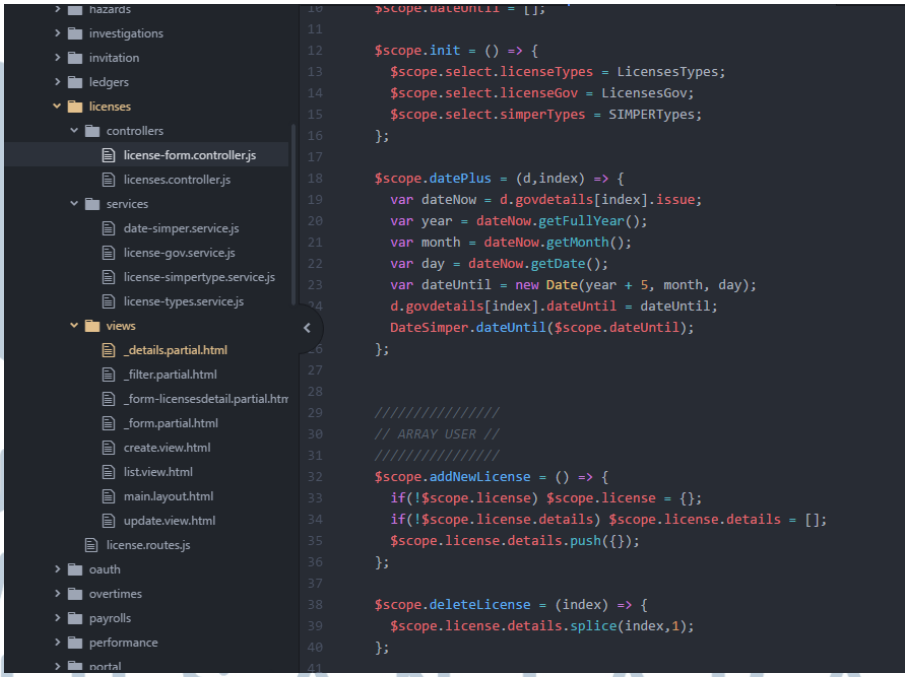
Update view memiliki *coding* yang hampir sama dengan *create view*. Perbedaannya, *update view* mengambil data lisensi pekerja yang telah dimasukan sebelumnya dengan menggunakan *ng-init*. *Coding* tersebut dapat dilihat pada Gambar 3.30.

Controller, menerima *input* dari pengguna dan menginstruksikan *model* dan *view* untuk melakukan aksi berdasarkan masukan tersebut. *Controller* bertanggung jawab untuk pemetaan aksi pengguna akhir terhadap respon aplikasi (Burbeck, 1992). Terdapat dua *controller* pada modul *Licensing*, yaitu *controller* utama dan controller untuk *form*. *Controller* utama berguna untuk *create*, *update*, *read*, dan *delete* (CRUD). *Controller* untuk *form* berguna untuk pengaturan otomatis batas tanggal dan *array*.



Gambar 3.31 License Controller Modul Licensing

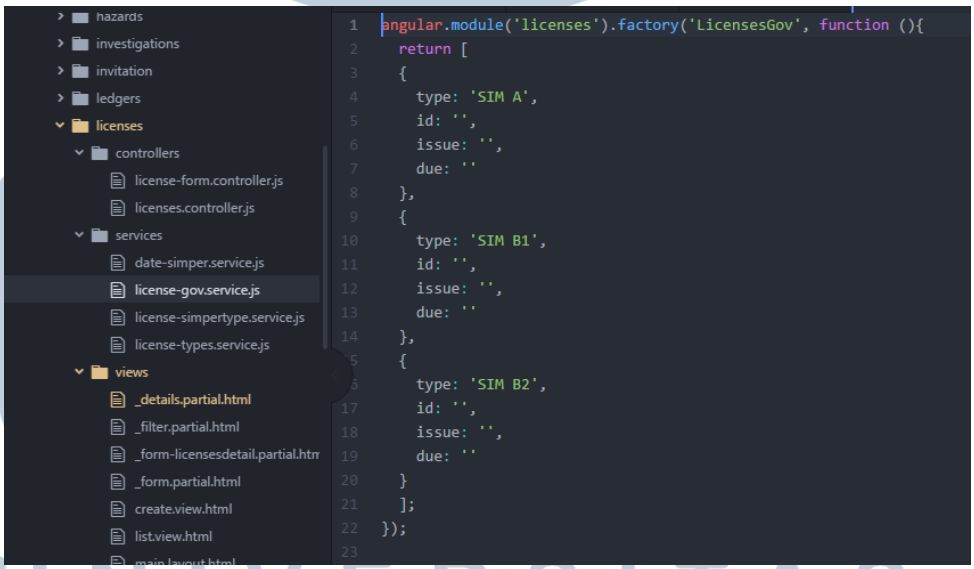
Gambar 3.31 merupakan *controller* dari *Licensing*. Pada *controller* ini berisi CRUD yang berguna untuk memasukan dan menampilkan data pada sistem.



Gambar 3.32 License Form Controller Modul Licensing

Controller pada Gambar 3.32 berguna untuk mengatur *form* sehingga mudah diisi oleh *user*. Pada Licensing *controller* digunakan untuk menambahkan tanggal secara otomatis untuk SIM dan menambahkan lisensi lebih dari satu untuk satu *user*.

Adapun juga *service layer* pada sistem ini. *Service layer* biasanya digunakan pada sistem perusahaan. *Layer* ini berguna untuk mendefinisikan batas sistem dengan cara membuatkan satu set operasi dan mengkoordinasikan respon. Membuat operasi tersebut dapat digunakan secara berulang – ulang dalam operasional (Stafford, 2002). *Service* tersebut digunakan pada modul *Licensing* untuk mempermudah pengembangan.



Gambar 3.33 SIM Service Licensing Modul Licensing

Gambar 3.33 merupakan contoh dari salah satu *service* yang digunakan di *Licensing*. Hasil dari *service* ini menghasilkan tiga

SIM dalam bentuk array tanpa *user* harus menginputnya secara manual.

PT Volante Teknologi Indonesia ingin membuat projek ini menjadi salah satu *software as a service* (SaaS) milik perusahaan. Fitur – fitur yang dimiliki oleh *Single Identity Database* dan *Be GeMS Auto Tracking System* dapat dibilang sangat serupa dengan *Human Resources Information System* (HRIS). Fitur fitur tersebut antara lain, *payroll*, *performance tracking*, *attendance*, *teams*, dan lain – lain. Tugas berikutnya adalah *cost of goods sold* dan *cost structure* dari HRIS.

6. Cost of Goods Sold

Cost of Goods Sold (COGS) adalah biaya yang diperlukan untuk menghasilkan produk sehingga produk tersebut siap untuk dijual. Jika pada konteks SaaS, maka COGS merupakan perhitungan seberapa besar data yang digunakan setiap user pada infrastruktur SaaS. Perhitungan data dihitung dari *record* pada setiap modul di HRIS. COGS untuk HRIS dapat dilihat pada Gambar 3.34.

RECORD	BYTES PER RE(	RECORD/MONT	RECORD/YEAR	YEARS	GB per User in X YEARS
P01 - Employees	1500	1	12	1	0.000016764
P01 - Overtime	150	3	36	1	0.000005029
P01 - Payroll	500	1	12	1	0.000005588
P01 - Reimbursemer	150	3	36	1	0.000005029
P01 - Teams	1200	2	24	1	0.000026822
P01 - Timeoff	150	2	24	1	0.000003353
P02 - Attendance	150	30	360	1	0.000050291
P02 - Projects	400	2	24	1	0.000008941
P02 - Tasks	400	50	600	1	0.000223517
P02 - Asset	500	1	12	1	0.000005588
P03 - Approvals	250	4	48	1	0.000011176
P03 - Audit	500	0.1	1.2	1	0.000000559
P03 - Commissionin(	350	1	12	1	0.000003912
P03 - Hazard	500	0.1	1.2	1	0.000000559
P03 - Investigation	500	0.1	1.2	1	0.000000559
P03 - License	350	1	12	1	0.000003912
P03 - RFID	250	2	24	1	0.000005588
P03 - Vendors	500	0.5	6	1	0.000002794
		Harga / perusahaan / 1 taon			$ 0.0114
		Harga / perusahaan / bulan			$ 0.000950
		Dalam Rupiah (kurs 13.600)			Rp 12.92

Gambar 3.34 Cost of Goods Sold HRIS

Semua record di HRIS akan didaftarkan pada tabel dan dihitung *bytes* per *record* pada modul – modul HRIS. Perhitungan dari *bytes* per *record* dibantu oleh pembimbing magang dengan menghitung *bytes* melalui *MongoDB.* Setelah itu, dihitung berapa *record* yang akan dihasilkan setiap bulannya. Setelah mendapatkan *record* setiap bulannya, maka *record* tersebut dikalikan dua belas untuk dijadikan satu tahun. Setelah semua hasil tersebut dikumpulkan maka dapat dihitung berapa *gigabytes* yang digunakan setiap *user* dalam setahun dengan rumus,

$$GB\ per\ User\ in\ X\ Years: \frac{Bytes\ per\ Record * Record\ Per\ Year}{(1024^3) * Years}$$

Rumus 3.1 GB per User in X Years

Setelah mendapatkan hasil perhitungan Rumus 3.1, maka data tersebut akan ditotal lalu dikali dengan $30. $30 merupakan harga dari *Amazon Web Service*, yaitu *hosting server* yang digunakan oleh PT Volante Teknologi Indonesia. Harga tersebut adalah harga setahunnya sehingga harus dibagi kembali dengan dua belas sehingga menjadi harga per bulan. Harga per bulan dalam US *dollar* tersebut akan dikalikan dengan kurs rupiah sebesar Rp. 13.600,00.

7. Cost Structure dan Break Even Point

Cost structure digunakan untuk mengetahui biaya yang dikeluarkan oleh PT Volante Teknologi Indonesia untuk menghasilkan sistem HRIS. *Cost structure* dari HRIS dapat dilihat pada Gambar 3.35.,

Cost Structure

No	Item			QTY	Unit	Cost/Unit		Price
1	Internet and Office		Office Rent and Wifi Connection Fee	1	year			Rp15.600.000
2	Electricity			1	year			Rp12.000.000
3	Research & Development			1	year			Rp254.562.000
4	Infrastructure		DB Server, Apps Server and ELB	2	month	Rp2.500.000	Rp5.000.000	
				10	month	Rp5.000.000	Rp50.000.000	
								Rp55.000.000
5	Misc (10%)		Overhead Costs	1				Rp33.716.200
			TOTAL					Rp370.878.200

Item Name	Qty	Cost	Sub Total	Tax	Sub Total + Tax
Project Lead	1	Rp10.000.000	Rp10.000.000	5%	Rp10.500.000
Programmer	4	Rp8.000.000	Rp32.000.000	5%	Rp33.600.000
Analyst	2	Rp8.000.000	Rp16.000.000	5%	Rp16.800.000
			Cost Per Month		Rp60.900.000

Cost Per Month	Rp60.900.000
Number of Month	4
Total	Rp243.600.000
PPh Pasal 23 4.5%	Rp10.962.000
Total + PPh	Rp254.562.000

Gambar 3.35 Cost Structure HRIS

Perhitungan ini dibantu oleh pembimbing magang dan sudah direvisi beberapa kali. Projek ini memiliki rentang waktu selama empat bulan untuk analisis dan pengembangan. Ada lima aspek yang menjadi hitungan dalam *cost structure ini*, yaitu sewa kantor dan internet, listrik, analisis dan pengembangan, serta biaya tak terduga (*overhead cost*). Analisis dan pengembangan dibagi menjadi satu *Project Lead*, empat *Developer*, dan dua *Analyst.* Gaji dari karyawan tersebut sudah ditambahkan PPh Pasal 23 sebesar 4.5%. Hasil tersebut akan ditotalkan sehingga biaya yang diperlukan adalah sebesar Rp. 370.878.200,00.

Jika sudah menghitung *cost structure* dan COGS maka dapat dihitung pula *break even point* (BEP) dari sistem HRIS. BEP adalah titik dimana pendapatan dari sistem sama dengan modal yang dikeluarkan PT Volante Teknologi Indonesia. Rumus dari BEP adalah,

$$Break\ Even\ Point : \frac{Fixed\ Cost}{(Unit\ Price - Unit\ Cost)}$$

Rumus 3.2 Break Even Point

Assume that the unit price is,						
Unit Price	Rp25.000		Rp50.000		Rp100.000	Rp150.000
Fixed Cost	Rp370.878.200					
Unit Cost	Rp12,92					
		Rumus BEP per Unit = Fixed Cost / (Unit Price - Unit Cost)				
Monthly Users	14843		7419		3709	2473
Monthly Clients	594		297		148	99
Yearly Users	1237		618		309	206
Yearly Clients	49		25		12	8

Gambar 3.36 Break Even Point HRIS

Fixed cost didapatkan dari total *cost structure* dan *unit cost* didapatkan dari total COGS. Ada empat tolak ukur pada *unit price*, yaitu sebesar Rp. 25.000, Rp. 50.000, Rp. 100.000, dan Rp. 150.000. Tujuan dibagi menjadi empat tersebut sehingga mengetahui kira – kira seberapa besar harga yang harus diberikan untuk menjual HRIS pada setiap *user*. Hasil dari BEP tersebut dapat dilihat pada Gambar 3.36.

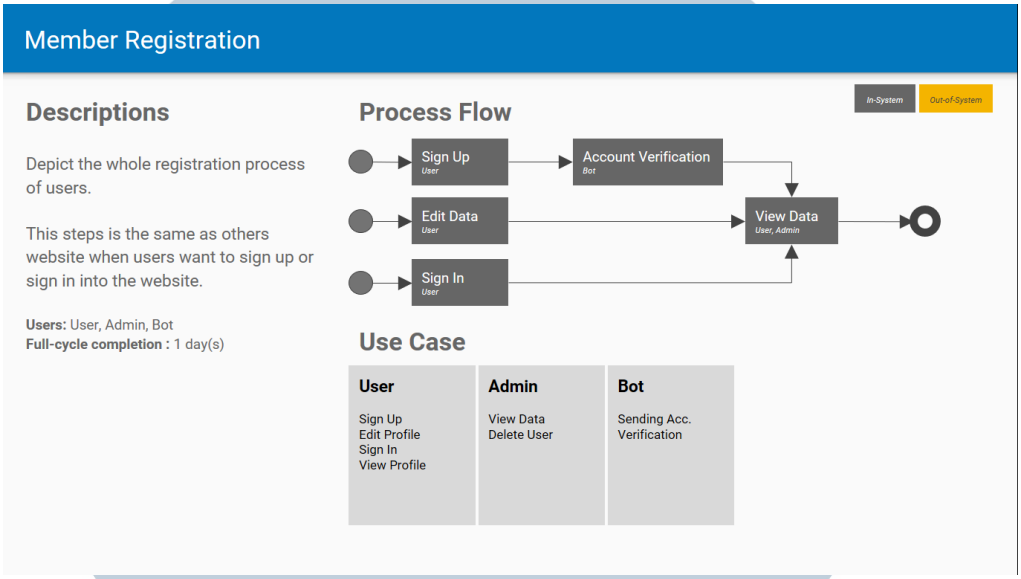
3.3.2 Trading Station

Hal yang ditugaskan adalah membantu menganalisis *user requirement* yang diberikan *client Trading Station*. Pada *user requirement* tersebut berisi fitur – fitur yang diinginkan client untuk ada pada *website Trading Station* tersebut. Antara lain fitur – fitur tersebut adalah *sign in* dengan *social media*, *free member*, *premium member*, *copy trading*, *marketplace*, broker, *eChanger*, *admin area*, kompetisi, *workshop*, dan lain – lain. *User requirement Trading Station* akan dilampirkan pada halaman Lampiran.

Dalam proses analisa, *user requirement* yang diberikan oleh *client* kurang jelas. Oleh sebab itu, perlu dibuat beberapa pertanyaan untuk *client* yang bersangkutan. Guna pertanyaan – pertanyaan tersebut ditujukan agar memperjelas *user requirement* sehingga projek yang dihasilkan sesuai dengan tujuan yang diingin *client*. Beberapa pertanyaan tersebut meliputi cara kerja *copy trading*, perbedaan *free member* dan *premium member*, cara kerja *marketplace*, pengadaan broker *trading*, fitur *eChanger*, dan cara kerja kompetisi *trading* dan *expert adviser*. Daftar pertanyaan secara rinci akan dilampirkan pada halaman Lampiran.

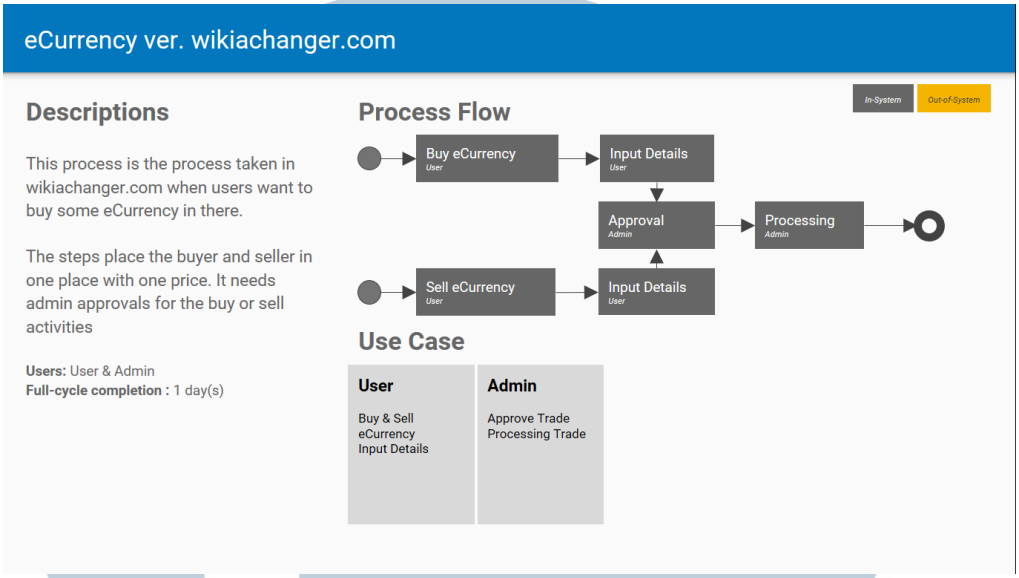
Dikarenakan *client* yang bersangkutan sulit dihubungi dan sampai kontrak magang ini selesai, pertanyaan belum terjawab maka dialihkan untuk membuatkan *use case* dan *process flow* untuk modul *Member Registration* dan *eChanger* terlebih dahulu.

1. *Use Case and Process Flow*



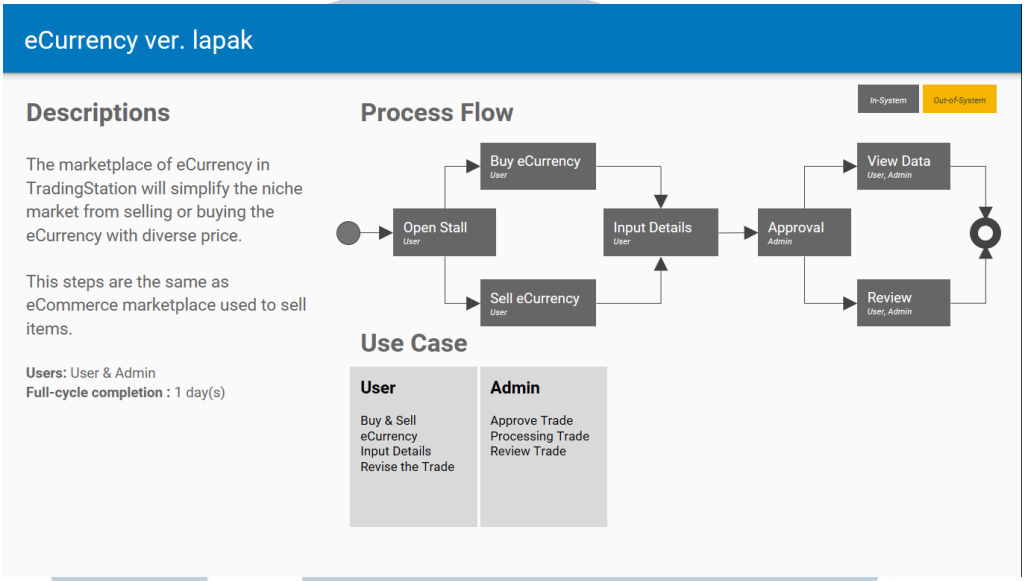
Gambar 3.37 Member Registration Trading Station

Gambar 3.37 merupakan alur dari melakukan registrasi pada *website Trading Station* ini. Proses ini banyak digunakan pada *website* – *website* lainnya dalam melakukan registrasi user. User yang melakukan registrasi akan dikirimkan verifikasi akun ke *email user* tersebut. Setelah melakukan verifikasi, *user* sudah *login* ke dalam *website Trading Station.*



Gambar 3.38 eCurrency ver. wikiachanger.com

Gambar 3.38 merupakan *use case* dan *process flow* menurut *website* wikiachanger.com yang kemungkinan serupa dengan *website Trading Station*. Pada proses ini pembelian dan penjualan *eCurrency* akan dijadikan pada satu tempat dengan harga yang sama. Sehingga tidak diperlukan *marketplace* dan memudahkan user dalam membeli atau menjual eCurrency. Setiap pembelian atau penjualan yang telah dilakukan harus memerlukan persetujuan admin untuk memproses transaksi.



Gambar 3.39 eCurrency ver. Stall

Use case dan *process flow* pada Gambar 3.39 merupakan hasil dari alur *eCurrency* ini dibuat sesuai dengan pengetahuan terhadap *user requirement* tersebut. Jika alur kerja sesuai dengan yang ada di *user requirement* maka pembelian atau penjualan akan dijadikan sebagai *marketplace*. Dengan sistem *marketplace* maka akan membuat satu barang akan memiliki kuantitas dan harga yang berbeda antara satu dengan yang lain. Hasil dari proses ini belum tentu sesuai dengan yang keinginan *client*.

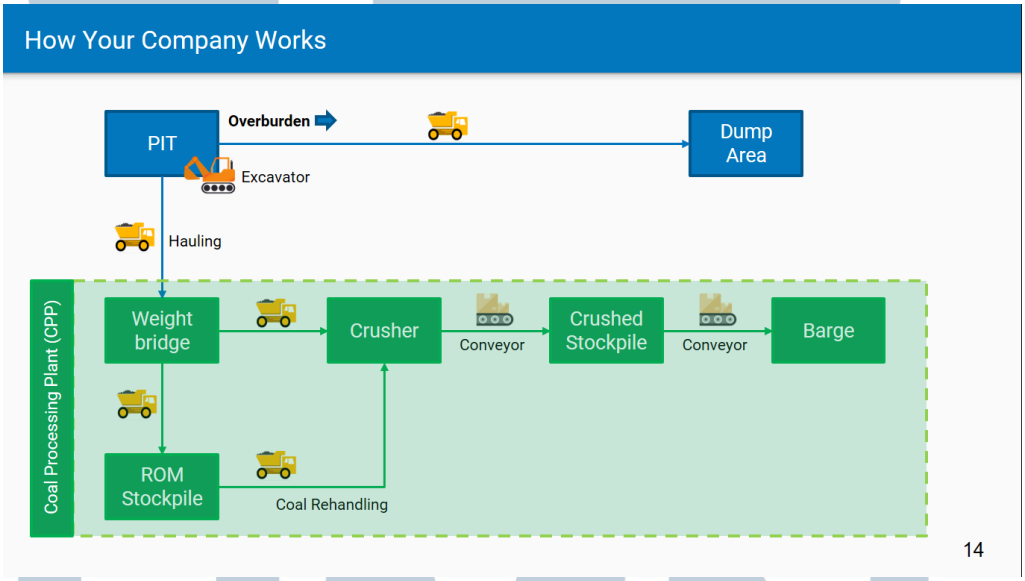
3.3.3 Operating Execution / Monitoring System

Projek ketiga yang ditugaskan adalah melakukan analisa mengenai alur kerja dari perusahaan tambang. Analisa yang dilakukan dimulai dari area penggalian tambang sampai hasil tambang tersebut sudah dapat diangkut ke tongkang (*barge*). Analisa tersebut dilakukan dengan bantuan

dari pembimbing magang, *website* mengenai pertambangan, dan *YouTube* untuk memperjelas alur kerja tersebut.

Setelah analisa sudah dilakukan, hasil dari analisa tersebut dituangkan ke dalam bentuk gambar sehingga dapat dijelaskan dengan lebih baik pada Gambar 3.40.

1. Alur pertambangan di PT Berau Coal



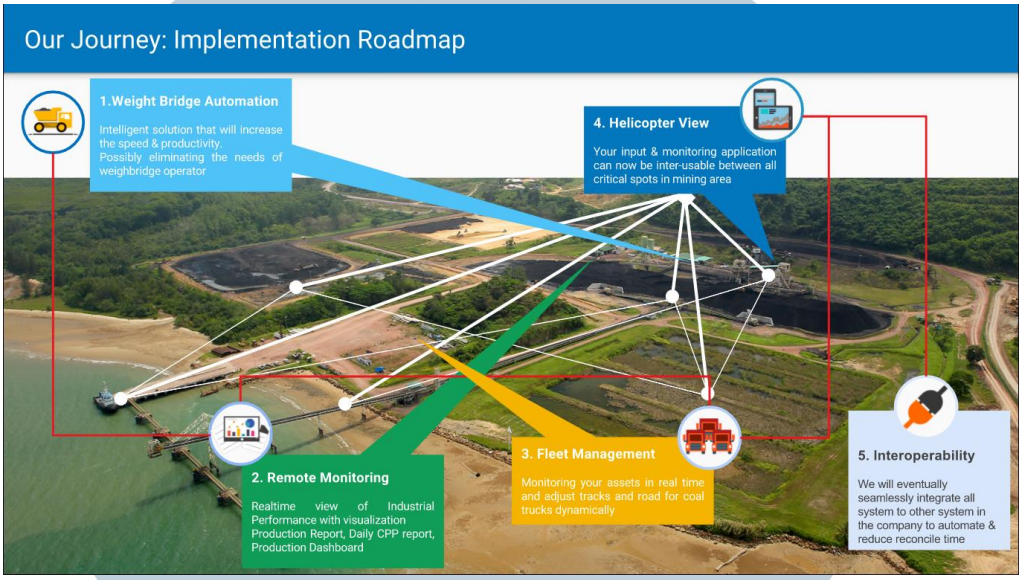
Gambar 3.40 Mining Workflow in PT Berau Coal

Alur kerja pertambangan dimulai dari area penggalian tambang (*Pit*). Tambang tersebut digali dengan menggunakan *excavator*. Hasil dari penggalian tersebut akan menghasilkan dua hal yaitu, pertama adalah lapisan tanah penutup (*overburden*) yang akan diangkut oleh truk dan dibuang di *dump area*. Kedua adalah hasil tambang atau batu bara yang akan diangkut dengan menggunakan truk untuk melakukan perhitungan berat dari batu bara yang

didapatkan di jembatan timbang (*weighbridge*). Setelah ditimbang di *weighbridge*, maka truk tersebut memiliki dua tujuan yaitu *ROM Stockpile* atau *crusher*. *ROM Stockpile* adalah tempat penyimpanan sementara batu bara. C*rusher* merupakan mesin untuk menghancurkan bongkahan besar batu bara menjadi butiran kecil. Batu bara yang berada di *ROM Stockpile* bisa diangkut ke *crusher* untuk dihancurkan. Batu bara yang sudah menjadi butiran kecil akan diangkut menggunakan *conveyor* dan ditaruh di tempat penyimpanan (*crushed stockpile*). Jika terjadi pemesanan, maka batu bara yang berada di *crushed stockpile* tersebut akan diangkut ke dalam *barge* di dermaga sehingga hasil batu bara tersebut dapat diantarkan.

Setelah analisa tersebut selesai, tugas selanjutnya adalah menganalisa sistem yang dapat diimplementasikan pada perusahaan tambang tersebut.

2. *Implementation Road*



Gambar 3.41 Implementation Roadmap OE/EM

Gambar 3.41 menjelaskan lima sistem yang akan diimplementasikan ke dalam lima fase. Kelima sistem tersebut yaitu,

Weightbridge Automation, bertujuan untuk melakukan otomisasi terhadap *weighbridge,* sehingga truk yang membawa hasil tambang tersebut dapat didata dengan mudah. Sistem ini memungkinkan pendataan tersebut secara otomatis tanpa memerlukan operator sama sekali. Hal ini dapat membantu meningkatkan kecepatan dalam pendataan hasil tambang.

Remote Monitoring, merupakan sistem yang akan digunakan untuk melakukan pengawasan secara waktu nyata (*real time*) terhadap kinerja dari perusahaan. Pengawasan tersebut akan dibantu dengan menggunakan *report* dan *dashboard,* sehingga memiliki

visualisasi yang baik dan mudah dilihat. Ada tiga bagian yang akan dijadikan indikator untuk pengawasan pada sistem ini, yaitu *production report, daily CPP report*, dan *production dashboard.*

Fleet Management, digunakan untuk melakukan pemantauan dan pengendalian terhadap aset yang dimiliki perusahaan, yaitu truk tambang. Sistem ini akan berjalan secara *real time* dan dapat membantu mengatur jalur truk secara dinamis. Sistem ini akan diimplementasikan pada fase ketiga dalam projek.

Helicopter View, adalah sebuah sistem yang membantu dalam memantau secara keseluruhan. Sistem ini ditujukan untuk *top level management. Top level management* dengan mudah dapat mengakses sistem ini dan membantu mengambil keputusan dengan cepat.

Interoperability, pada fase kelima, sistem ini akan dibuat untuk dapat diintegrasikan dengan sistem lain yang digunakan oleh perusahaan, seperti *SAP*, *Oracle*, *Microsoft*, dan lain – lain.

3.4 Kendala yang Dihadapi

Kendala pada saat magang sebagai *Business System Analyst* di PT Volante Teknologi Indonesia tidaklah banyak. Salah satu hal yang menjadi kendala utama adalah komunikasi yang kurang baik dalam hal pembagian tugas. Hal ini terlihat dari pendelegasian tugas yang kurang sesuai dengan *job desk* masing – masing

karyawan. Dengan kata lain, satu orang dapat mengerjakan beberapa pekerjaan yang berbeda dan tidak sesuai dengan bidangnya.

3.5 Solusi

Solusi dari kendala di atas adalah dengan memperbaiki cara komunikasi dan sistem koordinasi di dalam perusahaan tersebut. Koordinasi yang baik antar individu atau pun antar divisi juga dapat dibentuk dengan pengadaan rapat rutin setiap minggunya. Melalui rapat rutin ini, seluruh pekerja dapat mengevaluasi kinerja selama seminggu dan saling memberikan masukan. Selain itu, pembagian tugas kepada masing – masing orang juga harus tetap didasarkan pada kemampuan dan bidang masing – masing.