

LAPORAN MAGANG
Web ACC Whistle
PT. Astra Sedaya Finance (Astra Credit Companies)



Dipersiapkan Oleh :
Jonathan Wahyu Nugraha Pinulung/140707910

PROGRAM STUDI TEKNIK INFORMATIKA
FAKULTAS TEKNOLOGI INDUSTRI
UNIVERSITAS ATMA JAYA YOGYAKARTA
2019



No : 042 / HC-RM / RMS / II / 2019

Jakarta, 4 Februari 2019

Perihal : Surat keterangan ACC STARSHIP (Digital Workplace – Techno Center)

Dengan hormat,

Dengan surat ini, kami menyatakan bahwa mahasiswa dengan nama sebagai berikut :

Nama	Universitas	Jurusan
Jonathan Wahyu Nugraha	Universitas Atma Jaya Yogyakarta	Teknik Informatika

Telah **MENYELESAIKAN** program Star Internship at ACC (Starship) selama 6 bulan, sejak **16 Juli 2018 – 31 Desember 2018** di bagian **Information Technology** dengan tugas melakukan development dan implementation terhadap beberapa project digital dari ACC.

Demikian pemberitahuan kami, atas perhatiannya kami ucapkan terima kasih.

Hormat kami,

PT. ASTRA SEDAYA FINANCE

Roberto Sontani

HC Recruitment Management

HALAMAN PENGESAHAN

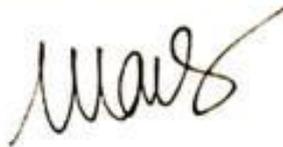
Laporan Magang

Laporan ini telah diperiksa dan disetujui

Pada tanggal :11 Februari 2019

Oleh :

Dosen pembimbing,



Martinus Maslim, S.T., M.T

Pembimbing Lapangan,



Adamono Awamiti

KATA PENGANTAR

Magang adalah salah satu dari mata kuliah wajib yang harus ditempuh oleh setiap mahasiswa Fakultas Teknik Industri, program studi Teknik Informatika, Universitas Atma Jaya Yogyakarta sebagai salah satu syarat untuk mengambil mata kuliah Tugas Akhir (TA).

Melalui magang diharapkan mahasiswa dapat memperluas pengetahuan dan pemahaman mengenai disiplin ilmu disertai penerapannya secara nyata yang belum pernah didapatkan selama kuliah, mengingat dunia kerja saat ini membutuhkan tenaga kerja yang berpengalaman bagi sebuah perusahaan. Maka laporan ini dibuat untuk menyimpulkan pengalaman dan pengetahuan yang didapat selama penulis melakukan magang.

Laporan ini dapat penulis selesaikan dengan baik dan tepat waktu karena tidak terlepas dari bantuan beberapa pihak, oleh karena itu penulis mengucapkan terimakasih kepada:

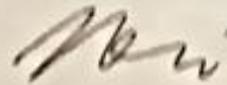
1. Tuhan Yesus Kristus, yang selalu menyertai dan mencurahkan berkat, kesehatan, kasih karunia, anugrah, dan pengetahuan yang melimpah kepada penulis
2. Orang tua, yang selalu tiada henti memberikan doa, semangat, dukungan dan motivasi selama penulis melakukan studi
3. Bapak Martinus Maslim, S.T., M.T selaku Kepala Program Studi Teknik Informatika Universitas Atma Jaya Yogyakarta dan selaku dosen pembimbing magang yang selalu memberikan nasehat dan arahan pada penulis selama menjalankan magang
4. Seluruh dosen pengajar, staff, dan karyawan Universitas Atma Jaya Yogyakarta
5. Mas Novandy Pradana, selaku pembimbing lapangan magang pertama
6. Mbak Adamono Awamiti, selaku pembimbing lapangan magang kedua
7. Para pegawai Astra Credit Companies yang telah menerima penulis dengan baik dan dapat berkerja sama dengan baik selama magang
- 8.

8. Teman – teman yang telah membantu dan mendukung penulis dalam menyelesaikan laporan ini.
9. Kepada semua pihak yang telah berkenan memberikan bantuan dan dukungan serta kerja sama yang baik sehingga laporan ini dapat diselesaikan dengan lancar.

Akhir kata, penulis menyadari bahwa pelaksanaan magang dan penyusunan laporan ini masih belum sempurna. Oleh karena itu, kritik dan saran yang membangun penulis harapkan, semoga penyusunan laporan ini bermanfaat bagi semua pihak.

Yogyakarta, 11 Februari 2019

Penulis,



Jonathan Wahyu Nugraha Pinulung

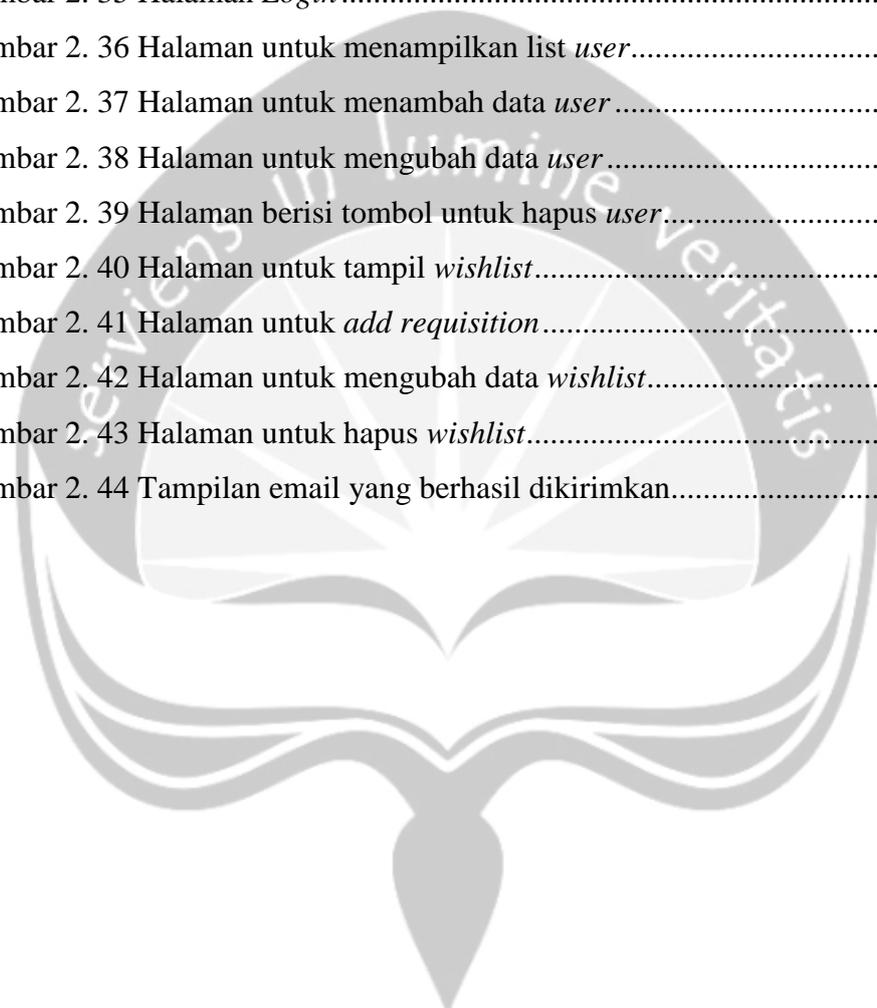
DAFTAR ISI

LAPORAN MAGANG	1
HALAMAN PENGESAHAN.....	3
KATA PENGANTAR.....	4
DAFTAR ISI.....	6
DAFTAR GAMBAR.....	7
BAB I PENDAHULUAN.....	9
1.1. Sekilas Perusahaan	9
1.2. Sejarah Perusahaan.....	9
1.3. Visi, Misi dan Nilai Perusahaan	11
1.4. Struktur Organisasi.....	12
1.5. Deskripsi Tugas Dalam Struktur Organisasi	13
1.6. Departemen IT dalam Perusahaan.....	14
BAB II PELAKSANAAN MAGANG	15
2.1. Penjelasan Logbook	15
2.2. Hasil Pekerjaan Secara Umum	62
2.3. Bukti Hasil Pekerjaan	68
BAB III HASIL PEMBELAJARAN	85
3.1. Manfaat Magang.....	85
3.2. Penerapan Ilmu dalam Magang.....	85
BAB IV KESIMPULAN.....	87

DAFTAR GAMBAR

Gambar 1. 1 Struktur Organisasi.....	12
Gambar 2. 1 Logo UniPaas.....	19
Gambar 2. 2 Form dari UniPaas	20
Gambar 2. 3 Logic yang dilakukan dalam sebuah form di UniPaas.....	20
Gambar 2. 4 Logo PHP	20
Gambar 2. 5 Logo Git.....	21
Gambar 2. 6 Logo Laravel	21
Gambar 2. 7 Penerapan penggunaan MVC pada laravel	22
Gambar 2. 8 Logo OutSystems	35
Gambar 2. 9 Contoh <i>slide button</i>	49
Gambar 2. 10 Contoh <i>breadcrumbs</i>	50
Gambar 2. 11 <i>Design</i> data yang digunakan untuk <i>requisition</i>	58
Gambar 2. 12 Halaman Tampil <i>List PIC</i> Speakup.....	68
Gambar 2. 13 Halaman <i>Create PIC</i> baru	68
Gambar 2. 14 Halaman <i>Edit PIC</i>	69
Gambar 2. 15 Halaman <i>Delete PIC</i>	69
Gambar 2. 16 Halaman List Pelaporan	70
Gambar 2. 17 Halaman home dengan fungsi <i>search</i>	70
Gambar 2. 18 Halaman Detail Laporan	71
Gambar 2. 19 Tampilan Email untuk Pembuat Laporan	71
Gambar 2. 20 Tampilan Email Aktifasi <i>User</i>	72
Gambar 2. 21 Tampilan Email <i>Approval</i> Pelaporan	72
Gambar 2. 22 Tampilan Email <i>Approval</i> Penambahan PIC	73
Gambar 2. 23 Tampilan Email Lupa Password	73
Gambar 2. 24 Halaman untuk Mengubah PIC	74
Gambar 2. 25 Halaman untuk melihat data semua <i>player</i>	74
Gambar 2. 26 Halaman untuk menyimpan <i>player</i> baru	75
Gambar 2. 27 Halaman untuk mengubah data <i>player</i>	75
Gambar 2. 28 Halaman <i>list player</i> setelah berhasil menghapus data.....	76
Gambar 2. 29 Halaman untuk menampilkan seluruh data <i>point</i>	76

Gambar 2. 30 Halaman untuk menambah data <i>point</i>	77
Gambar 2. 31 Halaman untuk mengubah data <i>point</i>	77
Gambar 2. 32 Halaman <i>list point</i> setelah berhasil menghapus data	78
Gambar 2. 33 Halaman untuk menampilkan leaderboard.....	78
Gambar 2. 34 Halaman untuk memanggil fungsi <i>reset</i>	79
Gambar 2. 35 Halaman <i>Login</i>	79
Gambar 2. 36 Halaman untuk menampilkan list <i>user</i>	80
Gambar 2. 37 Halaman untuk menambah data <i>user</i>	80
Gambar 2. 38 Halaman untuk mengubah data <i>user</i>	81
Gambar 2. 39 Halaman berisi tombol untuk hapus <i>user</i>	81
Gambar 2. 40 Halaman untuk tampil <i>wishlist</i>	82
Gambar 2. 41 Halaman untuk <i>add requisition</i>	82
Gambar 2. 42 Halaman untuk mengubah data <i>wishlist</i>	83
Gambar 2. 43 Halaman untuk hapus <i>wishlist</i>	83
Gambar 2. 44 Tampilan email yang berhasil dikirimkan.....	84



BAB I

PENDAHULUAN

1.1. Sekilas Perusahaan

Astra Credit Companies atau biasa disingkat dengan ACC adalah perusahaan pembiayaan mobil dan alat berat. Sesuai dengan Peraturan Otoritas Jasa Keuangan No. 29/POJK.05/2014 ACC melakukan perluasan usaha di bidang Pembiayaan Investasi, Pembiayaan Modal Kerja, Pembiayaan Multiguna dan Sewa Operasi (Operating Lease), baik dengan skema konvensional maupun syariah.

1.2. Sejarah Perusahaan

PT Astra Sedaya Finance yang merupakan cikal bakal ACC berdiri pada 15 Juli 1982 dengan nama PT Rahardja Sedaya, didirikan guna mendukung bisnis otomotif kelompok Astra. Di tahun 1990, PT Rahardja Sedaya berganti nama menjadi PT Astra Sedaya Finance. Dalam perkembangannya, PT Astra Sedaya Finance memiliki penyertaan saham pada perusahaan asosiasi, yaitu PT Swadharma Bhakti Sedaya Finance, PT Pratama Sedaya Finance, PT Staco Estika Sedaya Finance dan PT Astra Auto Finance yang semuanya telah terdaftar dan diawasi oleh Otoritas Jasa Keuangan (OJK) serta mendapatkan izin dari Departemen Keuangan Republik Indonesia Direktorat Jenderal Lembaga Keuangan dengan nomor izin usaha sebagai berikut:

- PT Astra Sedaya Finance dengan nomor izin usaha 1093/KMK.013/1989
- PT Swadharma Bhakti Sedaya Finance dengan nomor izin usaha 1095/KMK.013/1989
- PT Astra Auto Finance dengan nomor izin usaha 437/KMK.017/1995
- PT Staco Estika Sedaya Finance dengan nomor izin usaha 590/KMK.013/1990
- PT Pratama Sedaya Finance dengan nomor izin usaha 1258/KMK.013/1989

Sejak tahun 1994, PT Astra Sedaya Finance dan perusahaan asosiasinya mengembangkan merek Astra Credit Companies untuk mendukung usahanya. ACC berkomitmen penuh untuk meningkatkan layanan pada masyarakat. ACC menyediakan fasilitas pembiayaan untuk pembelian mobil dan alat berat dalam kondisi baru ataupun bekas serta fasilitas Pembiayaan Investasi, Pembiayaan Modal Kerja, Pembiayaan Multiguna dan Sewa Operasi (Operating Lease). ACC juga mendukung penjualan mobil melalui jaringan dealer, showroom maupun perseorangan di seluruh wilayah Indonesia. Jaringan ACC tersebar di hampir seluruh kota besar di Indonesia. Saat ini ACC memiliki 75 kantor cabang yang tersebar di 59 kota di Indonesia, dan akan terus bertambah.

Astra Credit Companies selalu mempertahankan reputasinya sebagai perusahaan pembiayaan terkemuka di Indonesia. Saat krisis ekonomi melanda Indonesia di tahun 1998, ACC dapat melewati krisis ini dengan baik. Termasuk juga dapat melunasi pinjaman sindikasi pada tahun 1999, tanpa restrukturisasi.

Pada tahun 2000, ACC mulai melakukan penerbitan obligasi dengan rating A- dari PT Pemeringkat Efek Indonesia. Saat ini ACC telah meraih peringkat AAA (idn) Stable Outlook dari PT Fitch Ratings Indonesia dan peringkat AAA dari PT Pemeringkat Efek Indonesia. ACC sebagai perusahaan pembiayaan terbesar dan terkemuka di Indonesia, selalu membayarkan nilai pokok hutang dan bunga tepat pada waktunya.

1.3. Visi, Misi dan Nilai Perusahaan

Visi

Become the 1st Choice Financing Company with Total Solution.

Misi

To Promote Credit for a Better Living.

Nilai

- **Integrity**

Berani mentaati peraturan berlandaskan asas dan etika yang berlaku serta menunjukkan sikap profesional dan bertanggung jawab.

- **Teamwork**

Bersinergi melalui interaksi yang positif dan terbuka, dengan komitmen mencapai target perusahaan.

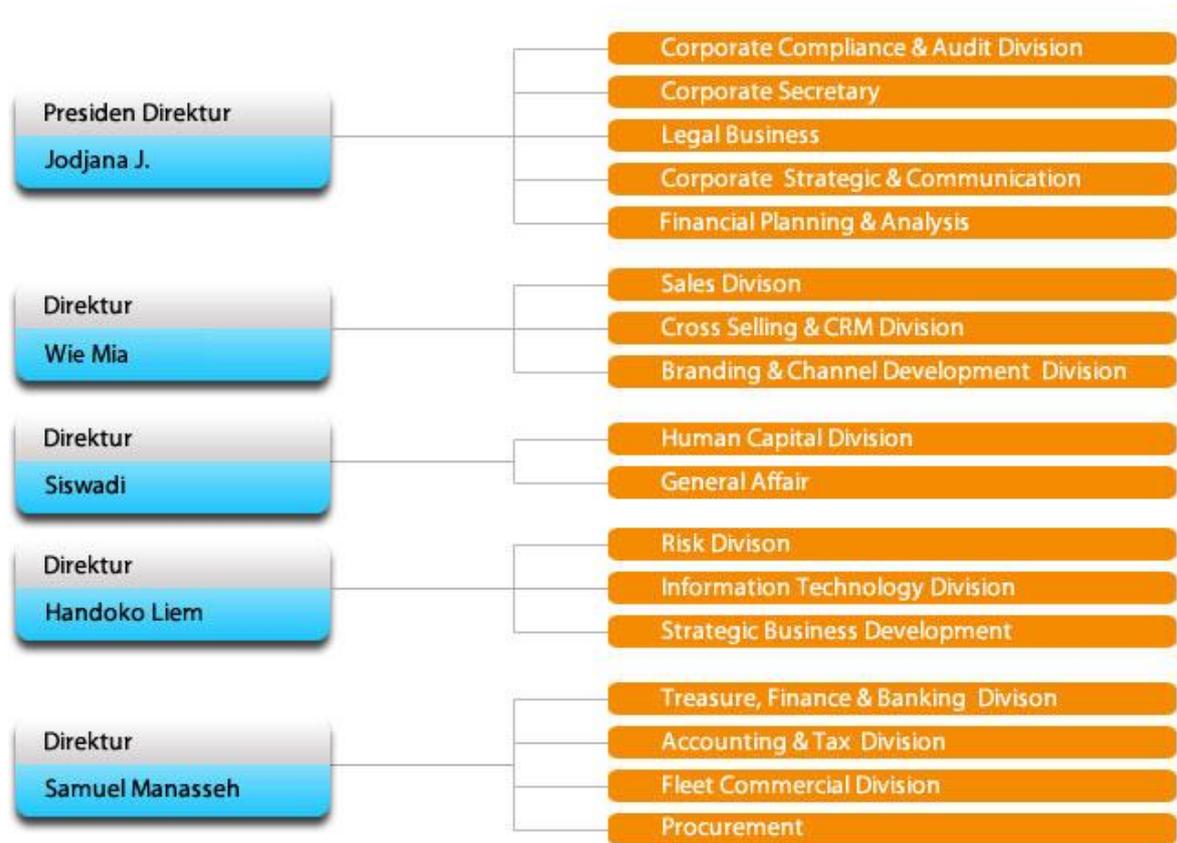
- **Quality**

Proses yang cepat dan akurat dengan mentalitas perbaikan secara terus menerus untuk hasil yang terukur dan terbaik.

- **Customer Satisfaction**

Memberikan pengalaman terbaik bagi pelanggan melalui pelayanan yang handal dan terpercaya.

1.4. Struktur Organisasi



Gambar 1. 1 Struktur Organisasi

1.5.Deskripsi Tugas Dalam Struktur Organisasi

Tugas Corporate Compliance & Audit Division:

- Menjaga keberlangsungan tata kelola yang baik.

Tugas Corporate Secretary:

- Memastikan kelancaran komunikasi antara perusahaan dengan pemangku kepentingan.

Tugas Legal Business:

- Mengelola perizinan, dokumen, HaKI, ketenagakerjaan dan masalah hukum di perusahaan.

Tugas Corporate Strategic & Communication:

- Menjaga dan membangun citra positif perusahaan di mata publik.

Tugas Financial Planning & Analysis:

- Perancangan keuangan dan mengurus aliran kas masuk dan keluar.

Tugas Sales Division:

- Melakukan penjualan dan pemasaran produk.

Tugas Cross Selling & CRM Division:

- Mengelola hubungan korporasi dan pelanggan pada level bisnis.

Tugas Branding & Channel Development Division:

- Membangun dan menjaga identitas perusahaan.

Tugas Human Capital Division:

- Memperkerjakan staff untuk menyelesaikan tugas dan mendorong agar staff tersebut menjadi inovatif.

Tugas General Affair:

- Mendukung kegiatan operasional perusahaan melalui pengadaan barang dan jasa yang dibutuhkan.

Tugas Risk Division:

- Mengarahkan dan menjaga perusahaan dari risiko risiko yang dapat mengganggu pencapaian sasaran organisasi.

Tugas Information Technology Division:

- Membuat dan bertanggung jawab dengan pembuatan dan pengembangan aplikasi bisnis.

Tugas Strategic Business Development:

- Mengatur strategi bisnis pada perusahaan agar dapat mencapai tujuan organisasi.

Tugas Treasure, Finance & Banking Division:

- Mengatur optimalisasi penggunaan dana kas perusahaan.

Tugas Accounting & Tax Division:

- Mengatur dan mengelola keuangan dan perpajakan perusahaan.

Tugas Fleet Commercial Division:

- Mengelola dan mengawasi pelaksanaan operasional penjualan untuk produk bersifat fleet atau banyak dan besar.

Tugas Procurement:

- Mengelola pengadaan dan perawatan barang.

1.6. Departemen IT dalam Perusahaan

Astra Credit Companies mempunyai departemen Teknologi Informasi yang bertugas dalam pembangunan, pengelolaan dan penyempurnaan sistem atau aplikasi yang digunakan untuk mendukung dan mempermudah proses bisnis pada perusahaan.

Astra Credit Companies sendiri membagi departemen Teknologi Informasi yang ada menjadi 2 yaitu bagian yang mengurus sistem atau aplikasi yang berhubungan dengan Core Business dan yang diluar Core Business itu sendiri.

BAB II

PELAKSANAAN KERJA PRAKTEK

2.1. Penjelasan Logbook

Senin, 16 Juli 2018

Pada hari pertama Kerja Praktek ini, pihak ACC melakukan acara *welcoming* peserta magang yang dilaksanakan di *Head Office* ACC yang berada di Jl. TB Simatupang No.90. *Welcoming* tersebut dimulai dengan perkenalan diri peserta magang kepada berbagai perwakilan dari divisi - divisi IT (*Information Technology*), *division head* IT, serta direktur IT ACC yang turut menyampaikan ucapan selamat datang dalam program magang ini. Pada *welcoming* ini peserta magang juga diperkenalkan kepada perwakilan dari IT ACC yang akan mengawasi kinerja para peserta magang saat di Jogja. Kegiatan dilanjutkan dengan *briefing* singkat mengenai kegiatan dan pembagian tugas atau project yang akan peserta ikuti selama kegiatan magang. Terdapat 2 buah proyek yang masing – masing akan ditangani oleh 2 peserta magang. Proyek yang peserta dapatkan adalah proyek web dengan nama Speakup yang menggunakan *framework* Laravel nantinya.

Proyek SpeakUp ini bertujuan sebagai wadah para pegawai ACC untuk melakukan pengaduan terhadap temuan kecurangan yang mungkin terjadi. Melalui wadah ini, Departemen *Complaint* nantinya dapat mengetahui kecurangan yang terjadi secara lebih mudah dan dapat memutuskan langkah yang akan diambil menanggapi laporan yang ditemukan. Setelah penjelasan singkat mengenai kedua proyek yang akan ditangani oleh para peserta magang, acara dilanjutkan dengan melakukan penandatanganan kontrak dan pengumpulan beberapa dokumen.

Selasa, 17 Juli 2018

Hari berikutnya peserta magang mendapatkan materi training yang dibuka dengan pengenalan mengenai Astra secara luas. Dimulai dari sejarah

dan bidang bisnis yang Astra lakukan. Selain itu peserta magang juga diperkenalkan kepada beberapa perusahaan besar yang bekerjasama dengan Astra. Setiap aspek bisnis yang dikelola Astra memiliki anak perusahaannya sendiri. Astra Credit Companies sendiri merupakan salah satu lini bisnis Astra dalam bidang *finance*.

Kemudian kegiatan magang dilanjutkan dengan materi *IT Governance* di ACC. Ada pula materi yang peserta dapat melalui kegiatan tersebut adalah mengenai metode pengendalian kualitas dan *testing* yang dilakukan oleh divisi IT di ACC. Orang yang menangani *testing* ini sendiri disebut dengan QC (*Quality Control*). Para peserta magang diberikan pengajaran apa tujuan dari *testing*, sejauh mana *testing* dilakukan dan dokumentasi hasil dari *testing*. *Testing* sendiri dibedakan menjadi 3 tahap, yaitu UT, SIT dan UAT. UT (*Unit Testing*) adalah *testing* yang dilakukan pada waktu *developing* yang dilakukan oleh *developer*. SIT (*Sistem Integration Testing*) adalah *testing* yang dilakukan setelah selesai masa *developing* dan sistem akan di-*test* oleh QC, dari SIT ini jika masih ditemukan *bug* maka sistem akan dikembalikan ke *developer* untuk dilakukan *fixing* (perbaikan). Sedangkan UAT (*User Acceptance Testing*) adalah *testing* yang dilakukan oleh *user*, *testing* ini dilakukan jika tidak lagi ditemukan *bug* setelah SIT. Dalam *testing* terdapat 4 jenis *error* atau *bug* yang ditemukan, yaitu *major*, *medium*, *minor*, dan *content*. Dari keempat jenis *bug* ini memiliki tingkat prioritas dan seberapa besar dampak *error* terhadap sistem yang dibangun. Secara berurutan dari *major* memiliki tingkat prioritas dan dampak *error* yang paling besar, sedangkan untuk *content* tingkat prioritas dan dampak *error* yang kecil seperti adanya salah tulisan pada tampilan sistem.

Rabu, 18 Juli 2018

Pada hari Rabu, peserta kembali mengikuti training dengan materi *IT Infrastructure* dan *IT Architecture* di ACC. *IT Architecture* adalah *blueprint* atau gambaran dari sebuah sistem yang akan dibangun. *IT Infrastructure* ialah barang atau sistem dan berbagai hal lain yang menjadi pondasi IT suatu perusahaan. Adapun infrastruktur mencakup *database*, *operating system*,

visualization, server, storage dan juga *networking*. Terdapat 3 buah aspek yang diperhatikan dalam *IT Infrastructure*, yaitu *Availability, Maintenance* dan juga *Security*. *Availability* berarti memperhatikan kemampuan kerja sistem dalam keadaan apapun. *Maintenance* berarti memperhatikan perawatan dan pembaharuan sebuah pondasi, agar dapat digunakan untuk sistem yang baru. *Security* berarti memperhatikan tingkat keamanan dari sistem yang ada dari ACC.

Peserta juga mendapatkan pengetahuan pembagian bisnis proses yang ada di ACC serta peran IT dalam setiap bisnis proses yang dilakukan. Ada empat bagian yang ada dalam pembagian bisnis proses pada ACC yaitu AA (*Account Acquisition*), AM – COLL (*Account Maintenance - Collection*), BO (*Back Office*) dan EIS (*Enterprise Information Sistem*). Pada hari ini peserta mendapatkan training mengenai proses bisnis AA. AA sendiri merupakan proses bisnis paling awal, dimana AA adalah proses bisnis yang meliputi dari bagaimana ACC mencari hingga mendapatkan dan menyetujui *customer* yang ada. Pendataan data pribadi *customer* hingga mencapai kontrak.

Kamis, 19 Juli 2018

Pada hari Kamis, peserta mendapatkan materi *training* mengenai proses bisnis bagian AM – COLL dan BO. Dimana AM (*Account Maintenance*) adalah bagian yang mengurus proses pembayaran angsuran *customer*. Pembayaran melalui ACC ini dapat dilakukan melalui *Direct Payment* atau melakukan pembayaran secara langsung kepada ACC, *Maintenance Kontrak* atau melakukan perubahan kontrak dalam kondisi tertentu, ataupun dengan cara lainnya. Para *customer* juga dapat melakukan pembayaran angsuran tanpa harus datang langsung ke ACC. Pembayaran dapat dilakukan dengan *Transfer, H2H (Host to Host)* melakukan pembayaran melalui bank, dan ACC yes! yang merupakan sebuah aplikasi mobile untuk pembayaran angsuran.

Selain proses bisnis AM, terdapat COLL (*Collection*) yang mengurus bagian penarikan. COLL sebenarnya adalah proses bisnis yang akan dilakukan hanya jika *customer* terlambat melakukan pembayaran angsurannya. COLL

sendiri dibedakan menjadi 4, yaitu DC, ARHO, ARRO, dan REMO. DC sendiri dilakukan melalui telepon saat *customer* terlambat membayar selama 1-7 hari. ARHO dilakukan dengan cara mendatangi rumah dari *customer* saat terlambat membayar selama 8-30 hari. Pada proses ARRO barang *customer* sesuai dengan kontrak akan ditarik jika terlambat membayar selama 31-60 hari. Sedangkan REMO akan dilakuakn penarikan paksa menggunakan pihak ketiga.

Proses bisnis BO (*Back Office*) adalah bagian yang mengurus tentang pencarian dana ataupun hal lainnya yang juga berhubungan dengan pihak ketiga. Adapun terdapat 4 jenis dari BO yaitu *Join Finance*, *Orafin*, *Collateral* dan *Subrogasi*. Pada hari ini pula peserta mendapatkan pelatihan mengenai *IT Security* yang ada pada ACC. *IT Security* tentu saja bekerja untuk menjaga semua sistem yang ada didalam ACC, didalamnya terdapat 3 buah prinsip utama yaitu *Availability* (ketersediaan), *Confidentiality* (kerahasiaan), *Integrity* (Akurat). Dalam menjaga sistem tersebut, ACC menggunakan beberapa *security device* seperti *Firewall*, *Antispam*, *Antivirus*, *MDM (Mobile Device Management)*, *Identity Management*, dan *WAF (Web Application Firewall)*.

Jumat, 20 Juli 2018

Pada hari Jumat, peserta mendapatkan materi mengenai EIS (*Enterprise Information System*) yang dimana mengurus pelaporan dan pencatatan mengenai kegiatan bisnis yang ACC lakukan. EIS sendiri memiliki urutan proses yang dibedakan menjadi 2 bagian yaitu OLTP (*Online Transaction Process*) dan OLAP (*Online Analytical Process*). Pada OLTP sendiri terdapat ETL (*Extract Transform Load*) yang bertugas untuk menyatukan semua data yang akan digunakan ke tahap *Staging*, yang nanti data yang sudah matang akan disimpan ke dalam *Data Warehouse*. Lalu bagian kedua yaitu OLAP, terdapat BI yang bertugas untuk menyusun laporan atas data matang yang tersimpan di *Data Warehouse*. Melalui EIS, terbentuk proses CRM (*Customer Relationship Management*) yang bertujuan untuk terus menjaga dan membangun hubungan dengan para *customer* maupun calon *customer*. Di CRM terdapat 4 cara yaitu, *Tele Service (Inbound Service)*, *Tele Marketing (Outbound Service)*, *CRS*

(*Customer Response Sistem*), dan CCRP (*Customer Complaint Resolution Progress*).

Peserta magang juga mendapatkan materi SDLC (*System Development Life Cycle*). SDLC adalah penentuan bagaimana kerja masa *development* sebuah sistem. Dalam SDLC, terbagi menjadi 3 yaitu *Traditional (Waterfall)* yang menunjukkan bahwa setiap proses dalam *development* harus dituntaskan terlebih dulu sebelum masuk ke dalam proses selanjutnya, *Object (Spiral)* dimana proses dalam *development* dapat kembali ke proses sebelumnya dengan memperhitungkan selesainya proses, *Agile (Scrum)* dimana proses dalam *development* dapat kembali ke proses sebelumnya dengan memperhitungkan timeline yang berarti setiap proses memiliki jadwalnya yang harus masing – masing. Nantinya para peserta magang akan menggunakan Scrum selama pembuatan sistem.

Senin, 23 Juli 2018

Pada hari ini peserta magang kembali melakukan *training* dengan materi standarisasi pemrograman di perusahaan. Peserta magang mempelajari ada standar pemberian nama variabel dan penulisan kode program di sebuah perusahaan guna dalam melakukan standarisasi program dan mempermudah setiap *programmer* membaca program maupun melakukan pengembangan sistem. Peserta magang hari ini juga diperkenalkan dengan *tool* pemrograman Magic Unipaas. Magic Unipaas merupakan *tools* pemrograman yang kerap kali dipercaya dalam pengembangan sistem *core* bisnis ACC.



Gambar 2. 1 Logo UniPaas

Selasa, 24 Juli 2018

Training pada hari ini akan dilanjutkan dengan *training* pemrograman menggunakan UniPaas. Dilanjutkan dengan mengikuti *test* yang diberikan untuk menguji kemampuan peserta magang dalam memahami materi *training* yang diberikan.

Verifikasi Kontrak

No Aggr : 01100103001723633
 St Aggr : CPRE OVD
 No Police : B 1263EFM

Warning

CUSTOMER MEMILIKI MULTIKONTRAK MENUNGGAK DAN WO, CUSTOMER DIWAJIBKAN MEMBAYAR KEWAJIBAN JIKA MAU MENGAMBIL BPKB

STATUS	NO AGGREMENT	TYPE MOBIL	NOMOR POLISI	ANGSURAN KE	OVERDUE	SALDO AR
CPRE	06100106009536050	FEROZA	B 2467RI	36	120	0

OK

Gambar 2. 2 Form dari UniPaas

```

6 El Control Verification of: v:no aggr Cnd: 40 VP PILOTING T
7 Verify Error 0 No Aggr harus diisi..! Display in Box Cnd: 1 Trim(v:no aggr) = *
8 Call SubTask 2 Cel. Aggr Cnd: 2 Trim(v:no aggr) <
9 Verify Error 0 No Aggr tidak valid..! Display in Box Cnd: 2 Trim(v:no aggr) <
10
11 Call SubTask 8 cek. tpd
12 Call SubTask 6 cek. status
13 Call SubTask 3 cek. ovd 1 Cnd: 13 NOT(vf:found aggr)
14 Call SubTask 7 cek. status 1 Cnd: 13 NOT(vf:found aggr)
15
16 Block If 32 ==91518_TEDDY==
17 (Trim(v:no aggr) < * AND VP PILOTING TPS
18 panggil procedure PRC_GEN_MULT_OVD
19 Call SubTask 12 call prc_gen_multi_ovd [1 Arguments]
20 Call SubTask 13 count tmp [1 Arguments]
21 panggil mod-transaction Notif Multikontrak Overdue
22 Block If 41 (IF(V TMP;'TRUE';'FALSE') = 'TRUE'
23 Call By Name "SMOD-TRAN"%Trim(ECF_VERS) [1 Arguments] Cnd: 40 VP PILOTING TPS
24 Block End )
25 cek. sudah ada approval/bim
26 Block End )
  
```

Gambar 2. 3 Logic yang dilakukan dalam sebuah form di UniPaas

Rabu, 25 Juli 2018

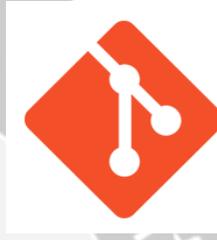
Pada hari ini peserta magang mendapat materi pelatihan pemrograman PHP. Pelatihan ini berguna untuk project magang yang akan penulis kerjakan. Pelatihan pemrograman PHP ini dilanjutkan dengan implementasi penggunaan framework yang nantinya akan digunakan yaitu Laravel. Pada pelatihan hari ini para peserta magang diajari mengenai konsep pemrograman di Laravel, yang biasa disebut MVC (*Model, View, Controller*). Model berkaitan dengan inialisasi data yang akan digunakan atau diolah dalam sistem. *View* berkaitan dengan *front-end* atau tampilan yang akan dilihat oleh *user* dari sistem yang

ada. *Controller* adalah back-end atau kumpulan fungsi ataupun prosedur yang digunakan untuk mengolah data dari *front-end*. Pada pelatihan ini para peserta magang juga diajari mengenai bentuk relasi yang ada dalam setiap model.

Selain pelatihan mengenai pemrograman php, para peserta magang juga diajarkan mengenai penggunaan git. Dimana git ini digunakan untuk *version control* dari sebuah sistem yang bisa dikerjakan oleh banyak orang yang memiliki akses ke sistem tersebut. Melalui git sistem laravel yang akan dibangun akan dapat diakses oleh beberapa orang yang telah memiliki *Git Key*.



Gambar 2. 4 Logo PHP



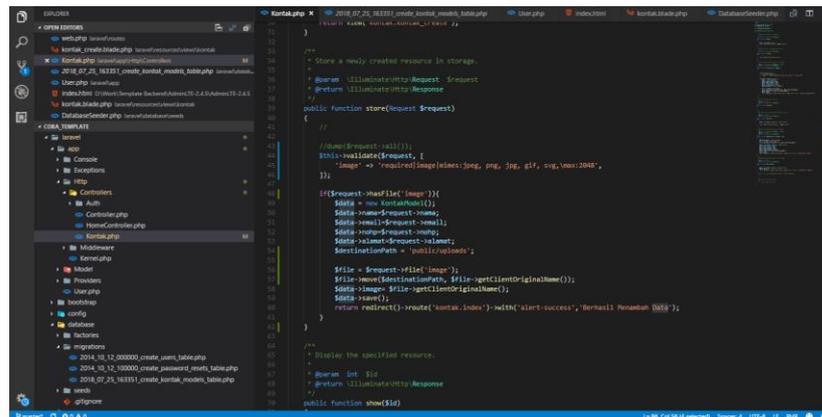
Gambar 2. 5 Logo Git

Kamis, 26 Juli 2018

Pada hari ini peserta magang kembali melanjutkan *training* penggunaan *framework* Laravel. Setelah sesi pelatihan selesai peserta magang melanjutkan dengan mengikuti *test* dan ujian mengenai materi pemrograman PHP dengan menggunakan *framework* Laravel.



Gambar 2. 6 Logo Laravel



Gambar 2. 7 Penerapan penggunaan MVC pada laravel

Jumat, 27 Juli 2018

Pada terakhir *training* di Jakarta ini, pesesrta mendapatkan kesempatan untuk melakukan *briefing* mengenai *Technical Design* proyek yang akan dikerjakan nanti. Penulis mendapatkan tugas SpeakUp. SpeakUp adalah merupakan sebuah wadah untuk menyampaikan penemuan pelanggaran yang dilakukan oleh karyawan ACC melalui web.

Senin, 30 Juli 2018

Pada hari ini penulis sudah melakukan pengerjaan *remote* proyek yang diberi nama *Digital Workplace*. Peserta magang melakukan kegiatan *development* di Ekologi (Yogyakarta), dimana kegiatan ini dimulai dengan melakukan *Daily Scrum* dengan masing – masing tim yang menangani proyek yang berada di Jakarta. *Daily Scrum* ini guna mendiskusikan proses bisnis masing – masing proyek. Seperti proyek SpeakUp yang ditangani oleh penulis, memiliki proses bisnis dimana complaint dapat dibuat oleh setiap orang yang sudah menjadi member dari sistem ini. Tiket yang masuk akan memiliki 3 buah status yaitu REGS (Registrasi), OPEN (Tiket telah di terima oleh PIC (*Person In Charge*)), CLOS (Tiket telah ditanggapi). Administrator dapat merubah status REGS menjadi OPEN, sedangkan PIC dapat mengubah status dari OPEN menjadi CLOS.

Setelah mendapatkan penjelasan mengenai proses bisnis dari sistem yang dibangun, peserta magang dijelaskan mengenai pembagian *task* dalam membangun sistem SpeakUp ini. Dimana setiap *task* memiliki *mandays* nya masing – masing. Pada hari ini penulis mengerjakan *task* pembuatan PIC tiket. PIC sendiri adalah orang yang bertanggung jawab terhadap departemen tertentu, yang nantinya bertugas untuk menerima tiket dan memberikan tanggapan atas tiket yang disampaikan. Data PIC yang di perlu disimpan adalah nama, email, nomor telepon, dan *department*.

Selasa, 31 Juli 2018

Pada hari ini penulis melakukan *Daily Scrum Meeting* menggunakan *video conference* Zoom untuk membahas dan berdiskusi mengenai *Technical Design*. Setelah itu penulis melanjutkan pembuatan PIC dengan perubahan form yang nanti akan digunakan untuk memasukkan data baru dan juga melihat detail dari data PIC yang sudah ada. Setelah *user* selesai mengisi data PIC, *user* dapat mengklik tombol tambah, jika tidak ingin menyimpannya dapat mengklik tombol batal. Selama pembuatan penulis merasa kesusahan karena masih asing dengan laravel. Setelah pembuatan, penulis mencoba melakukan UT untuk mengetahui apakah sistem sudah berjalan dengan baik atau belum. Selain memastikan data yang diinputkan sudah masuk ke dalam *database*, penulis juga memastikan bahwa data yang tersimpan di *database* sudah melalui validasi yang diberikan.

Rabu, 1 Agustus 2018

Pada hari ini penulis melakukan *video conference Daily Scrum* lalu dilanjutkan dengan pengerjaan *task* PIC untuk fitur *Edit*. Pada saat melakukan pembuatan fitur *edit*, penulis menemukan suatu masalah proses bisnis, apakah PIC dapat melakukan pelaporan juga atau tidak. Setelah berdiskusi dan memastikan bahwa PIC juga dapat melakukan pelaporan, maka ada perubahan pada *database*. Pada tabel PIC ditambahkan kolom id *user* yang digunakan untuk menampung data pelapor pada tabel tiket pelaporan, jika PIC tidak

memiliki id *user* maka secara logika pic bukanlah *user* dan seharusnya tidak dapat *login* ke dalam sistem dan tidak dapat melakukan pelaporan.

Setelah berhasil menyimpan data PIC dengan *database* dan validasi yang baru, maka penulis melanjutkan pembuatan sistem PIC dengan fitur *Edit*. Sebelum dapat masuk ke dalam form *edit*, penulis membuat tampilan dari keseluruhan data PIC. Tampilan dari *list* data PIC menggunakan *Yajra Datatable*, yang menjadi hal baru lagi untuk penulis. Setelah mengetahui cara penggunaannya melalui contoh yang lain, pembuatan *list* data PIC selesai pada hari ini.

Kamis, 2 Agustus 2018

Pada hari ini penulis melanjutkan pengerjaan task PIC untuk fitur *Edit* dan *Delete*. Untuk fitur *edit*, *user* diminta untuk memilih salah satu data PIC yang ingin diubah, dengan mengklik tombol *edit* yang ada di baris PIC. Setelah mengklik tombol *edit*, akan ditampilkan form untuk *edit* yang bentuknya sama seperti pada saat pembuatan PIC baru hanya berbeda *method*. Didalamnya terdapat 2 tombol yaitu Perbarui dan Batal. Jika sudah selesai melakukan perubahan dan akan menyimpannya maka *user* diminta untuk mengklik tombol Perbarui, jika tidak ingin menyimpan perubahan maka *user* dapat mengklik tombol batal.

Sedangkan untuk melakukan *Delete*, *user* dapat melakukannya dengan mengklik tombol *delete* pada baris data PIC yang ingin dihapus, maka data akan berhasil dihapus. Data yang berhasil dihapus tidak akan muncul pada *datatable*. Perlu diketahui, *delete* pada sistem ini dilakukan dengan cara *soft-delete* dimana data tidak hilang dari *database* hanya saja pada *database* terdapat *field Deleted_at* yang berisi tanggal saat data tersebut di-*delete*.

Jumat, 3 Agustus 2018

Pada hari ini penulis kembali melakukan penyempurnaan program untuk *task* PIC, penyempurnaan ini dimaksudkan untuk menyesuaikan *standard coding* yang ada, sehingga jika *developer* lain ingin melakukan perubahan tidak

kesusahan dalam memahaminya. Setelah melakukan penyesuaian *coding* pada *task* PIC, penulis melakukan *Unit Testing* untuk mengetahui proses bisnis mana yang masih terdapat *bug*.

Senin, 6 Agustus 2018

Pada hari ini penulis kembali melakukan *Daily Scrum Meeting* menggunakan *video conference* untuk memberikan laporan mengenai *task* yang sudah penulis kerjakan. Melalui *Daily Scrum Meeting* ini didapati bahwa proses bisnis yang ada sedikit berubah. Perubahan ini berkenaan pada email PIC saat melakukan *edit*. Perubahan ini berdampak pada *database* yang ada. Pada tabel PIC akan ditambahkan 1 *field* baru yang berisikan data id *user* dari PIC. Dengan kata lain saat melakukan pembuatan PIC, data akan tersimpan dalam tabel PIC dan tabel *user*. Pada saat *edit* PIC, jika email yang dimasukkan berbeda maka akan data PIC di tabel PIC akan diperbarui sedangkan pada *user* akan dibuat sebuah *record* baru. Dengan adanya penambahan *user* baru, maka *field* id *user* di data PIC akan di-*update* menjadi id *user* dengan email yang baru.

Selasa, 7 Agustus 2018

Penulis kembali melanjutkan *fixing* pada *task* PIC. Karena pada hari sebelumnya akan terdapat 2 buah data dari pengguna yang sama, dengan *group user* yang sama hanya berbeda email. Hal ini dapat membuat terbentuknya 2 buah PIC dengan pemilik yang sama, namun salah satunya tidak terdaftar pada table PIC. Ini berarti ada data aktif yang bersifat 'sampah'. Jika penerapan ini dilanjutkan, dan terjadi perubahan email berkali – kali maka akan terbentuk data – data sampah yang lebih banyak. Maka dari itu, penulis mengusulkan untuk menghapus data PIC dengan email lama pada tabel *user*. Dengan menghapus data menggunakan *soft delete* maka data tersebut akan dianggap tidak aktif, namun tidak menghilangkan data tersebut dari *database* nya. Dengan perubahan itu maka proses bisnis pada saat melakukan *edit* akan ditambahkan dengan fungsi menghapus data PIC dengan email lama pada table *user*.

Rabu, 8 Agustus 2018

Penulis melanjutkan magang dengan pembuatan *task develop List* Tiket. Sebelum membuat *list* tiket, penulis wajib mengetahui proses bisnis yang ada dalam pembuatan Tiket. Tiket yang dibuat memiliki beberapa data yang penting seperti id tiket, id *user*, kategori, sla dan juga status tiket. *User* meminta agar *list* tiket yang dapat ditampilkan setiap *group user* berbeda – beda. Saat *user* yang *login* adalah administrator maka akan menampilkan semua *list* data tiket tanpa terkecuali. Jika *user* yang *login* adalah PIC maka akan ditampilkan *list* data tiket dengan id pic sama dengan *user* yang *login* dan juga *list* data tiket dengan id *user* (pembuat) sama dengan *user* yang *login*. Sedangkan untuk *user member* maka akan ditampilkan semua *list* tiket yang pernah dibuat oleh *user* tersebut. Status pada tiket dibagi menjadi 3 yaitu REGS (saat pertama kali tiket dibuat), PEND (saat tiket sudah diassignkan ke PIC terkait), OPEN (saat PIC menerima tiket pengaduan) dan CLOS (saat tiket sudah selesai diatasi). PIC dapat melihat *list* tiket yang ditujukan kepadanya hanya pada saat status tiket sudah berubah menjadi OPEN. Pada saat status tiket berubah menjadi OPEN juga, sla akan mulai dihitung mundur dengan satuan hari. Dengan proses bisnis seperti ini, penulis membuat *list* data tiket menggunakan *datatable* seperti yang dilakukan di PIC, dimana pada *field action* akan menampilkan detail dari tiket tersebut.

Kamis, 9 Agustus 2018

Penulis melanjutkan *task* pembuatan *list* tiket terutama pada bagian detail tiketnya. Penulis menggunakan form yang mirip dengan saat pembuatan pic, dimana pada form teks dibuat *disable* sehingga tidak dapat diubah. Adapun pada form detail data yang ditampilkan ialah, Id Tiket, Tanggal Pembuatan, Id *User*, Kategori, Sub Kategori, PIC, Deskripsi Tiket, *File attachment*, *comment*. *User* yang dapat memberikan *comment* berbeda – beda sesuai dengan status tiket pada saat itu. Penulis menemukan beberapa permasalahan yang mungkin akan terjadi, seperti bagaimana jika *comment* bisa diberikan lebih dari 1 kali oleh 1 *user*, lalu bagaimana menampilkan *history* saat tiket berubah status

tiketnya. Maka dari itu dilakukanlah *Daily Scrum* untuk menyampaikan masalah yang didapatkan dan keputusan apa yang harus diambil. Pertama – tama, penulis diminta untuk berfokus pada perubahan status tiket. Maka diambil keputusan untuk menampilkan *history* perubahan status tiket dengan cara menampilkan *dropdown list* dari setiap tiket, dimana *dropdown* ini berisi data tiket yang sama hanya berbeda pada status tiketnya. Untuk mengimplementasikannya, penulis membuat proses bisnis baru dimana sistem akan menambahkan data tiket dengan data yang sama setiap kali terjadi perubahan status tiket. Setelah berhasil *list* data tiket akan menampilkan status tiket terbaru dan juga *history* pada *dropdown*-nya tanpa memiliki tombol detail, tombol detail hanya terdapat pada *list* tiket utama. Pada *list* tiket utama juga terdapat tombol *bar* sebagai toogle dari *dropdown*.

Jumat, 10 Agustus 2018

Pada hari ini penulis melakukan *Daily Scrum* dan mendapat pemberitahuan bahwa akan dilakukan *migration version* laravel dari 5.6 menjadi 5.4. Karena penulis masih memiliki beberapa *task* yang harus diselesaikan, *migration version* dilakukan oleh rekan penulis. Pada hari ini penulis meneruskan pembuatan *list* dan detail tiket, menangani masalah saat *comment* bisa dilakukan lebih dari 2 kali.

Senin, 13 Agustus 2018

Penulis beserta tim magang lainnya melakukan *testing migrasi* versi laravel dikarenakan kebutuhan *server* nantinya. Penulis juga menemukan beberapa perubahan pada form detail, dimana yang sebelumnya menggunakan form teks kali ini diubah kedalam bentuk tabel. Data – data yang ditampilkan sama, kecuali pada bagian comment. Pada database sendiri *field comment* pada tabel tiket sudah dihapus dan dibuat kedalam tabel baru yaitu detail tiket. Dengan menggunakan detail tiket, pemberian *comment* dapat dilakukan berkali – kali dan dapat ditampilkan semuanya sesuai dengan id tiket yang sesuai. Cara kerjanya menjadi seperti ini, saat *user* memberikan *comment* maka akan dibuat

record baru pada detail tiket dengan id tiket yang sesuai yang berisi id *user* pembuat *comment* dan juga *comment* yang diberikan. Pada hari ini juga diberitahukan perubahan proses bisnis, dimana Administrator hanya dapat merubah status tiket dari REGS menjadi PEND dan merubah OPEN menjadi CLOS, PIC dapat merubah status tiket dari PEND menjadi OPEN, sedangkan *user* hanya dapat memberikan *comment* pada saat REGS dan CLOS saja.

Selasa, 14 Agustus 2018

Pada hari ini penulis mencoba menyelesaikan *task list* dan detail tiket. Penulis mulai menambahkan fungsi untuk menghitung mundur SLA pada saat status tiket berubah menjadi OPEN, yang nantinya akan ditampilkan di dalam *datatable*. Lalu pada detail tiket penulis membuat setiap *group user* hanya dapat mengubah status tiket sesuai urutan proses bisnis, penulis berniat untuk mengurangi kesalahan pemilihan status tiket.

Rabu, 15 Agustus 2018

Pada hari ini penulis melakukan penambahan *history comment* pada detail tiket, menggunakan perulangan berdasar id tiket yang sesuai. Lalu dilanjutkan dengan melakukan *testing* dan memperbaiki jika menemukan *bug*, mengingat bahwa sebelumnya telah dilakukan migrasi versi laravel. *Testing* yang dilakukan meliputi *task* PIC, kategori dan tiket.

Kamis, 16 Agustus 2018

Pada hari ini kembali diadakan *Daily Scrum* untuk membahas migrasi dan temuan temuan *error* dan *bug* yang ada disaat migrasi. Penulis kembali melanjutkan untuk *task Search bar* yang ada di *header*. Melalui *search bar* ini *user* yang *login* dapat mencari tiket yang ingin dilihatnya dengan memasukkan id tiket saja. Saat mencoba untuk membuat *search*, penulis menemukan *bug* yang dapat mempengaruhi proses bisnis yang ada. *Bug* ini ditemukan jika sebuah tiket dengan kategori tertentu, dimana kategori yang telah terdaftar dihapus ataupun diubah maka akan terjadi kerusakan pada data tiket. Lalu

setelah dibentuk tabel baru yaitu detail tiket, penulis berniat untuk memasukkan *comment* pada saat terjadi perubahan status tiket. Pada penampilan *comment* terdapat juga *User* dan tanggal pembuatan *comment*, termasuk perubahan status tiket.

Senin, 20 Agustus 2018

Pada hari ini peserta mencoba untuk mengatasi *bug* yang sebelumnya ditemukan yaitu saat melakukan perubahan pada kategori yang telah di-*assign* ke dalam sebuah tiket. Dimana sistem dibuat dapat mencegah perubahan atau penghapusan kategori dan juga pic, saat masih ada tiket dengan kategori atau pic yang masih belum CLOS. Penulis membuat sistem agar menampilkan informasi dalam bentuk *message* kepada user, untuk mengetahui tiket mana saja yang masih aktif atau belum CLOS. Lalu, penulis mencoba untuk membuat tampilan baru pada data *list* tiket sesuai dengan permintaan dari *user*. *User* meminta agar penampilan *list* tiket untuk PIC dibedakan dengan cara membuatnya di dalam sebuah *tab*. *Tab* pertama berisi data *list* tiket buatan PIC, *tab* kedua berisi data *list* tiket yang harus ditangani PIC.

Selasa, 21 Agustus 2018

Pada hari ini peserta melanjutkan pengerjaan *task Search* Tiket. Dimana *user* memberi permintaan agar *user* sebagai member hanya dapat mencari tiket yang dibuat olehnya, sedangkan pic hanya dapat mencari tiket yang dibuat olehnya dan tiket yang ditujukan kepadanya, sedangkan untuk administrator dapat mencari semua tiket tanpa terkecuali.

Setelah menyelesaikan *task Search* Tiket, penulis melanjutkan dengan mengerjakan *task* pengiriman email kepada PIC saat status tiket berubah menjadi OPEN. Dimana email ini berisi *link* yang nanti akan diarahkan ke detail tiket yang ditujukan kepada PIC. Tidak hanya mengirimkan email pada saat perubahan tiket saja, penulis mencoba untuk membuat pengiriman email pada saat pic dibuat. Email ini berisi *link* yang nanti meminta PIC untuk mengubah *password*-nya sesuai dengan keinginan PIC.

Kamis, 23 Agustus 2018

Pada hari ini peserta melakukan perubahan pada email yang pada hari sebelumnya telah dibuat dan melakukan *Daily Scrum* untuk melaporkan hasil kerja dan masalah – masalah yang ditemukan. Pada saat *Daily Scrum*, penulis diminta untuk membuat *alert* pada saat *update* ataupun *delete* data yang berpengaruh pada tiket. *Alert* ini meminta persetujuan dari *user* untuk melakukan *update/delete* atau tidak. Jika *user* setuju untuk mengubah atau menghapus data yang dipilihnya maka sistem akan menjalankan fungsi, dimana fungsi ini juga akan melakukan pengecekan apakah tiket dengan data tersebut sudah CLOS semua atau belum jika belum maka sistem akan mencegah fungsi *update/delete*. Jika *user* tidak setuju untuk mengubah atau menghapus data yang dipilihnya maka akan kembali ke form sebelumnya dan tidak melakukan apapun.

Jumat, 24 Agustus 2018

Pada hari ini penulis kembali melanjutkan pengerjaan *task* dan melakukan penyempurnaan program untuk siap di *test* oleh *Quality Control* nantinya. Penulis juga menyempurnakan *alert* dan *message*, dimana *message* yang ditampilkan akan menampilkan *list* id tiket mana saja yang berkaitan dengan data pic ataupun kategori yang akan diubah atau dihapus dan juga masih dalam status belum CLOS. Penulis juga memberikan validasi pada PIC yang membedakan saat *create* dan *edit* agar tidak ada duplikasi email.

Senin, 27 Agustus 2018

Pada hari ini peserta beserta tim magang melakukan migrasi program ke *cloud server development* Alibaba agar *Quality Control* dapat melakukan *test* program langsung. Peserta juga mendapatkan beberapa *bug* atau *error* yang harus diselesaikan selama proses migrasi tersebut, beberapa *bug* tergolong dalam *bug* tampilan seperti perubahan bahasa dari bahasa inggris ke bahasa indonesia.

Selasa, 28 Agustus 2018

Pada hari ini penulis mengawasi proses SIT yang dilakukan oleh QC. Penulis juga akan menangani setiap *bug* ataupun *request* yang ada selama SIT berlangsung. Pada hari ini penulis menangani *bug* yang didapat pada saat melakukan *register* maupun *login*. Sebagian besar dari *bug* yang didapat mengarah pada validasi dari model yang ada. Selain itu terdapat juga *bug* dimana terdapat tombol yang belum memiliki fungsi. Dari perubahan validasi ini, terbentuklah *error handling* yang dapat memberikan informasi kesalahan apa yang telah dilakukan *user*.

Rabu, 29 Agustus 2018

Pada hari ini penulis beserta tim magang lainnya kembali memperbaiki laporan *bug* dan *error* yang ditemukan oleh tim *Quality Control*. Terdapat *bug* dimana pada saat *user* yang registrasi belum melakukan aktivasi, sudah dapat melakukan perubahan *password*. Untuk menangani *bug* tersebut penulis membuat ketentuan bahwa *user* tidak dapat merubah *password* apabila belum aktif, yang memaksa *user* untuk mengaktifkan id nya sebelum dapat merubah *password*-nya.

Kamis, 30 Agustus 2018

Pada hari ini ditemukan *bug* tentang kosongnya deskripsi kategori setelah dilakukan *update*. Pada saat *update* pun deskripsi sebelumnya tetap muncul, hanya ketika pada saat tidak ada perubahan pada bagian deskripsi maka deskripsi sebelumnya tidak akan tersimpan. Penulis menangani hal tersebut dengan cara menampung deskripsi sebelumnya agar tetap dapat disimpan sekalipun tidak ada perubahan.

Jumat, 31 Agustus 2018

Pada hari ini QC memberikan permintaan untuk menghilangkan tombol *order data* pada *datatable list* tiket kolom *action*. Permintaan ini berkaitan

karena tidak berpengaruhnya tombol *order* data ini terhadap kolom *action*. Untuk itu perlu dihilangkan. Lalu terdapat permintaan untuk membuat *paging* pada *follow up* yang ada di form detail tiket.

Senin, 3 September 2018

Pada hari ini penulis melakukan perbaikan untuk *bug* pada saat melakukan *create*, *edit* ataupun *delete user* dan PIC, dimana data dari kedua tabel ini akan disinkronisasikan. Sehingga proses bisnis dari pic dan *user* adapun seperti :

- Pada saat melakukan *create* PIC, akan dilakukan pengecekan terhadap email PIC yang diinputkan. Jika terdapat dalam tabel PIC dan juga *user* dan dalam keadaan aktif, maka *create* akan gagal. Sedangkan jika data terdapat pada tabel PIC dan *user* dalam keadaan *non-aktif*, maka *create* akan berhasil dengan mengaktifkan data tersebut dan mengubah data nama, *group*, dan *department* sesuai dengan yang diinputkan. Jika belum ada *create* dapat langsung dilakukan.
- Pada saat melakukan *edit user* dengan *group* pic, maka data yang ada dalam tabel pic akan *terupdate* juga. Jika *edit user* ini mengubah *group* dari PIC menjadi member atau administrator, maka data yang ada dalam tabel PIC akan dihapus. Jika email *user* dengan *group* PIC diubah, maka akan memanggil fungsi *create*.
- Pada saat melakukan *delete user* dengan *group* pic, maka data yang sesuai di dalam tabel PIC akan dihapus juga.

Selasa, 4 September 2018

Pada hari ini penulis masih melanjutkan perubahan proses bisnis yang terjadi pada tabel *user* dan PIC. Ditambah dengan melakukan pengecekan dampak perubahan PIC ini terhadap tabel kategori dan juga tiket.

Rabu, 5 September 2018

Pada hari ini penulis memberikan laporan bahwa perubahan proses bisnis yang dilakukan sebelumnya sudah selesai dan tidak ada *error* yang ditemukan, dan meminta QC untuk membantu melakukan *testing*. Sembari menunggu dilakukannya *testing* yang dilakukan oleh QC, penulis melakukan *supporting* kepada rekan kerja yang sedang mengerjakan *request* atau permintaan dari SIT, yaitu perubahan tampilan datatable kategori, pic dan *user* yang *error* ketika dilakukan *sorting*.

Kamis, 6 September 2018

Pada hari ini penulis siap untuk melakukan *support* terhadap UAT. Setelah UAT selesai dilakukan, penulis mengerjakan beberapa *request* UAT yang disampaikan QC secara pribadi. Adapun *request* ini adalah menyembunyikan deskripsi dari tiket saat kategori dari tiket tersebut adalah Fraud atau Etika Bisnis.

Jumat, 7 September 2018

Pada hari ini penulis melakukan *Daily Scrum* untuk membicarakan mengenai hasil dari UAT, karena pada hari sebelumnya penulis hanya diberikan 1 buah *request* dari QC. Pada saat *Daily Scrum* diberikan *list* dari *request* yang diminta oleh user setelah melakukan UAT. Adapun request tersebut mayoritas berhubungan dengan tampilan dari web, seperti penggunaan bahasa yang belum semuanya menggunakan bahasa Indonesia, atau pemberian *caption* yang kurang tepat.

Senin, 10 September 2018

Pada hari ini penulis melakukan pengerjaan *request* dari UAT yang pada hari Jumat lalu diberikan. Adapun *request medium* yang dikerjakan oleh penulis adalah merubah panel untuk melakukan *searching* tiket. User meminta agar *clue* yang ditampilkan dirubah dan bentuk dari kolom diubah menjadi lebih kecil. Dengan melakukan perubahan ini, penulis berinisiatif untuk melakukan *testing* untuk *searching* tiket.

Rabu, 12 September 2018

Pada hari ini penulis melakukan *Daily Scrum*. Penulis mendapatkan laporan dari QC jika *request* saat menyembunyikan deskripsi tiket pada saat kategori Fraud atau Etika Bisnis masih gagal, justru semua deskripsi dengan kategori apapun tidak tampil. Maka dari itu, penulis melakukan perbaikan untuk *request* tersebut. Pada saat *Daily Scrum*, penulis dan rekan kerja menemukan *request* yang tidak perlu, seperti menghitung kelebihan ukuran *file* yang *upload*, selain diperlukan *effort* yang besar, pengguna nantinya juga tidak memerlukan informasi seberapa besar kelebihan ukuran dari filenya. Sehingga diputuskan agar *request* tersebut dapat diabaikan.

Kamis, 13 September 2018

Pada hari ini penulis kembali melanjutkan pengerjaan *request search* tiket sekaligus melakukan *testing*. Penulis menemukan *bug*, dimana administrator tidak dapat mencari tiket dengan id pembuat bukan administrator. Penulis akhirnya melakukan *fixing* terhadap temuan *bug* ini, sekaligus merapikan *coding* untuk *search* tiket.

Jumat, 14 September 2018

Pada hari ini penulis bersama rekan melakukan pengecekan terhadap semua *request* UAT yang telah dikerjakan. Karena tidak ditemukan *bug* lagi dari perubahan *request* UAT dan proses bisnis lainnya, penulis melaporkan

bahwa UAT sudah selesai dikerjakan seluruhnya dan siap untuk diimplementasikan.

Senin, 17 September 2018

Pada hari ini penulis mendapatkan permintaan terbaru dari *user*, dimana saat pembuatan pengaduan akan diminta persetujuan dari *Division Head* yang menentukan apakah tiket ini dapat ditindaklanjuti atau tidak. Untuk itu penulis memberikan usul untuk menambahkan kembali status pada tiket pelaporan yang menunjukkan bahwa tiket pelaporan saat ini menunggu persetujuan dari *Division Head*. Oleh karena itu maka akan dibentuk sebuah *role* user baru yaitu *Authorizer* yang bertugas sebagai *Division Head* yang memberikan persetujuan terhadap tiket pelaporan yang diterima. Karena perubahan ini, status dari tiket memiliki alur yang berbeda pula. Laporan pertama kali diterima sistem akan memiliki status REGS, dimana tiket pelaporan ini dapat dilihat oleh administrator dan *Authorizer*. Jika *Authorizer* menyetujui pelaporan tersebut untuk ditindaklanjuti maka status akan berubah menjadi OPEN yang nantinya akan diterima oleh PIC, jika tidak disetujui maka status akan berubah menjadi CLOS dimana pelapor harus membuat laporan baru.

Selasa, 18 September 2018

Pada hari ini penulis kembali melanjutkan permintaan user untuk membuat persetujuan dari *Division Head* pada tiket pelaporan, sekaligus melakukan *testing*. Setelah selesai melakukan *testing*, penulis dan rekan – rekan magang melanjutkan pekerjaan dengan mempelajari OutSystems. OutSystems sendiri adalah *platform low-code* yang dapat digunakan untuk membuat sebuah web ataupun *mobile*. Kami diminta untuk mempelajari cara pembuatan aplikasi *mobile* menggunakan OutSystems



Gambar 2. 8 Logo OutSystems

Rabu, 19 September 2018

Penulis kembali mempelajari OutSystems. OutSystems sendiri memberikan beberapa *tutorial* yang dapat dilakukan oleh para penggunanya. *Tutorial* ini ada berbagai macam, antara lain membuat sebuah aplikasi *mobile* dalam waktu 5 menit. Dimana para *user* dapat mengunduh *template* yang telah diberikan oleh OutSystems dan mengikuti arahan yang ada secara otomatis, satu per satu. Terdapat juga *guided* untuk menjadi *mobile developer*. Didalamnya terdapat beberapa tahapan pengerjaan yang dikelompokkan menjadi 4 golongan berurutan. Yang pertama *intro*, *advanced*, *intermediate I*, dan *intermediate II*. Setiap tahapannya memiliki sub-bab yang menjelaskan secara detail cara pembuatan *mobile apps* mulai dari awal. Penulis memilih untuk melakukan *tutorial* membuat aplikasi *mobile* dalam waktu 5 menit dan memulai *guided mobile developer*.

Kamis, 20 September 2018

Sambil mempelajari OutSystems, penulis masih melakukan *supporting* kepada proyek SpeakUp. Tidak jarang penulis meninggalkan pembelajaran OutSystems untuk menangani SpeakUp. Pada hari ini penulis mencoba untuk mempelajari basis dari penggunaan OutSystems. Karena OutSystems adalah *low-code platform* pembuatan aplikasinya akan lebih difokuskan kepada *drag and drop*, melakukan perubahan terhadap *setting* setiap komponen yang ada seperti pemberian fungsi, nama, id, bentuk dan lain – lain. Selain itu OutSystems dapat digabungkan menggunakan javascript, untuk menentukan *style* yang akan dimiliki sebuah komponen. Untuk pembuatan fungsinya sendiri menggunakan cara yang sama dalam bentuk *flowchart*.

Jumat, 21 September 2018

Pada hari ini penulis melanjutkan mempelajari OutSystems tahap awal, yaitu pengenalan terhadap beberapa tombol bantu dan fungsi yang ada pada OutSystems, serta pembuatan data pada OutSystems. Tombol yang sering digunakan dan dibutuhkan adalah tombol “*1-Click Publish*”. Tombol ini

berfungsi untuk melakukan *debug* kepada hasil pengerjaan dan dapat dilakukan *test* melalui *mobile* maupun *web*. Project pada OutSystems dibagi menjadi 4 layer yaitu *Processes*, *Interface*, *Logic*, dan *Data*.

Processes sendiri berisi beberapa proses bisnis dan juga *event* yang akan ditampilkan di *layer Interface*. *Interface* sendiri merupakan *front-end* dari proyek yang sedang dibuat, disinilah *developer* dapat menyisipkan javascript. *Logic* berisi fungsi ataupun aksi yang dikerjakan pada *device* ataupun pada *server*, *integrasi*, *roles* dan *exceptions*. Sedangkan *Data* berisi struktur dari data yang akan digunakan pada proyek tersebut.

Senin, 24 September 2018

Pada hari ini penulis melanjutkan mempelajari OutSystems. *Guided* selanjutnya adalah penjelasan mengenai tipe dan perbedaan aplikasi yang dibuat oleh OutSystems dan modul aplikasi. Aplikasi yang dapat dibuat menggunakan OutSystem ada 2 yaitu *Web* dan *Mobile App*. Perbedaan dari kedua aplikasi ini terdapat pada beberapa aspek, seperti *client* dimana *web app* menggunakan browser sedangkan *mobile app* akan menjadi aplikasi pada sebuah *device*, *logic* pada *web* menggunakan *logic* yang ada pada *server* sedangkan *mobile* terdapat 2 buah *logic* yaitu pada *client* ataupun pada *server*, untuk *request* web bersifat berkesinambungan, sedangkan pada *mobile* kebalikannya, dan yang terakhir pada interaksinya aplikasi *web* menggunakan metode *request-response*, sedangkan *mobile* menggunakan *reactive*.

Modul aplikasi adalah bagian – bagian dan keseluruhan proyek dan memiliki 1 aspek yang penting untuk menjalankan fungsinya. Sebagai contoh dalam pembuatan proyek *mobile* akan terdapat setidaknya 2 buah modul, yaitu *core* yang menangani perubahan yang terjadi pada *server* (biasanya tidak memiliki *layer interface* dan fungsi berada pada *server*) dan juga *mobile* yang menangani segala perubahan yang terjadi pada *client* (biasanya memiliki *layer interface* dan fungsi berada pada *client*).

Selasa, 25 September 2018

Pada hari ini penulis melanjutkan mempelajari OutSystems sekaligus melanjutkan *supporting* pada proyek SpeakUp. *Guided* OutSystems akan mengajarkan tahap – tahap dari awal pembuatan aplikasi *mobile To Do*. Dimana aplikasi ini akan berfungsi untuk membuat, menampilkan, mengubah rencana kegiatan dari *user*. Pada *Guided* ini penulis akan membuat sebuah modul *core* untuk proyek *To Do*. Modul ini akan digunakan untuk melakukan *testing login* menggunakan *credential* dari OutSystems.

Lalu penulis diminta untuk menjelaskan kembali alur pembuatan tiket pelaporan yang ada, termasuk beberapa status baru yang ada. Status yang awalnya REGS, PEND, OPEN, REJ dan CLOS, berubah tampilannya menjadi *Register, Pending, Approve, Reject* dan *Close* untuk hasil datanya tetap sama.

Rabu, 26 September 2018

Pada hari ini penulis mendapatkan beberapa permintaan untuk proyek SpeakUp, untuk mengubah beberapa bentuk email yang dikirimkan melalui web SpeakUp serta beberapa penulisan pada web.

Guided OutSystems yang dibaca oleh penulis adalah mengenai data yang ada dalam OutSystems seperti *entity, atribut, tipe data, entity actions* (beberapa fungsi yang dalam *database* OutSystems), *static entity*, relasi data, dan *structure (temporary entities)*.

Kamis, 27 September 2018

Pada hari ini penulis kembali melanjutkan mempelajari OutSystems yang memasuki tahap cara mengambil atau mengolah data. Dalam pengambilan data ini dapat dilakukan dengan 2 cara yaitu menjadi *aggregate* (dibentuk kedalam bentuk *list*) dan *SQL Queries* (menggunakan *query sql*). Penggunaan *aggregate* biasanya ditemukan pada pengolahan data pada *device* dan *server*, sedangkan untuk *SQL* akan ditemukan pada pengolahan data pada *server*.

Jumat, 28 September 2018

Pada hari ini penulis kembali mendapatkan informasi bahwa ada sebuah *bug* pada proyek SpeakUp. *Bug* ini menyangkut pada perubahan email pada PIC. Saat email PIC diubah, maka akan dibuatkan sebuah *record* baru untuk PIC yang sama dengan email terbaru dan email sebelumnya akan dihapus dari table PIC namun data *role* di table User tidak berubah, sehingga menimbulkan *error* ketika selesai melakukan *login*. Oleh karena itu penulis memperbaiki alur dengan cara saat terdapat perubahan email pada *user* PIC maka data sebelumnya pada tabel PIC akan dihapus dan pada table User *role*-nya akan diubah menjadi member.

Senin, 1 Oktober 2018

Pada hari ini penulis melanjutkan *supporting* terhadap proyek SpeakUp. Penulis juga melakukan *testing* terhadap *update* terbaru yang diberikan oleh BSA (*Business Software Alliance*) proyek ini. Update ini berkaitan dengan saat administrator menghapus *user* dengan *role* PIC melalui table *user*. Data PIC tersebut selain dihapus dari tabel *User* juga harus dihapus dari table PIC.

Selasa, 2 Oktober 2018

Pada hari ini penulis mendapatkan *request* dari *user* untuk mengubah beberapa deskripsi ataupun nama – nama kolom yang ada dalam web SpeakUp. Ditambah juga dengan permintaan merubah beberapa format email yang dikirimkan saat melakukan aktivasi user, reset password pic, ubah *password* *user*, pembuatan tiket pelaporan, perubahan status tiket pelaporan, pembuatan PIC baru, dan penunjukan PIC pada kategori tertentu.

Rabu, 3 Oktober 2018

Pada hari ini penulis melanjutkan mempelajari OutSystems tahap pembuatan *entities* untuk proyek aplikasi *mobile* To Do yang pernah diajarkan pada *guided* sebelumnya. Dalam pembuatan *entities* ini beberapa yang perlu diperhatikan ialah seperti nama tunggal dan jamak dari sebuah *entity*, lalu

mementukan *attribute* mana yang menjadi *primary key* dari sebuah *entity*. Dalam pembuatan *entity* di OutSystems dapat dimasukkan sebuah *attribute* yang menjadi *attribute* di *entity* lainnya, dengan kata lain membuat relasi antar *entity* dengan sangat mudah.

Kamis, 4 Oktober 2018

Penulis melanjutkan mempelajari OutSystems bagian tema dan model pemrograman dari OutSystems, khususnya aplikasi *mobile*. Pada bagian tema, dijelaskan bagian – bagian dari *layer interface* yang dapat digunakan pada aplikasi *mobile*. Sedangkan pada bagian model pemrograman dijelaskan cara kerja dari aplikasi *mobile* berdasarkan hubungan antara ketiga *layer* yang ada dalam OutSystems.

Lalu pada bagian *guided* untuk membuat tampilan dari aplikasi *mobile*, seperti bentuk *splash screen* setelah berhasil masuk kedalam aplikasi dan juga menu – menu yang ada. Terdapat 2 cara penampilan menu. Yang pertama melalui *side bar*, menu akan ditampilkan dalam *list* vertikal dan yang kedua melalui *bottom bar* dimana menu akan ditampilkan secara *horizontal*. Setiap menu ini akan mengarahkan kepada halaman tertentu sesuai dengan nama menu yang ditampilkan.

Jumat, 5 Oktober 2018

Pada hari ini penulis kembali mendapatkan *request* dari *user* untuk menambahkan *alert* pada saat membuat tiket pelaporan. *Alert* ini muncul dalam 2 kondisi, yang pertama jika tiket pelaporan gagal dibuat karena terdapat kesalahan input dan yang kedua jika tiket pelaporan berhasil dibuat.

Pada saat ada form wajib yang tidak diisi oleh *user* maka ketika memilih tombol “Laporkan” akan muncul alert bahwa tiket pelaporan gagal dibuat karena masih terdapat form yang kosong. Jika semua sudah sesuai dengan yang diminta maka setelah memilih tombol “Laporkan”, muncul *alert* bahwa tiket pelaporan berhasil disimpan lalu laporan akan tersimpan di *database*.

Senin, 8 Oktober 2018

Pada hari ini penulis mendapatkan *request* kembali mengenai *alert* yang dibuat. Sebelumnya *alert* gagal akan muncul jika ada form yang kosong, namun jika terdapat format file yang tidak diijinkan diunggah, *alert* tidak muncul. Oleh karena itu user meminta agar *alert* gagal tetap muncul saat mendeteksi format *file* yang diunggah tidak sesuai.

Alert gagal yang muncul memberikan informasi bahwa format file yang diunggah salah dan meminta *user* untuk mengupload file yang diijinkan untuk diunggah.

Selasa, 9 Oktober 2018

Alert yang dibuat sebelumnya dirasa kurang baik jika alur *alert* berhasil muncul sebelum data benar – benar berhasil disimpan di *database*. Oleh karena itu BSA penulis meminta agar *alert* dimunculkan setelah tiket pelaporan tersimpan ke dalam *database*.

Oleh karena itu penulis memberikan status terhadap *link* saat melakukan pengisian form pelaporan yang menyatakan bahwa data berhasil disimpan. Lalu saat kembali ke *link* tersebut dengan status menampilkan *success*, maka *alert* berhasil disimpan akan dimunculkan.

Rabu, 10 Oktober 2018

Pada hari ini penulis melakukan *testing* ulang terhadap *alert* saat melakukan pembuatan tiket pelaporan. Penulis menemukan *bug* pada saat melakukan pembuatan tiket pelaporan oleh *guest*, tanpa *login*. Saat form masih kosong *alert* gagal tidak muncul.

Penulis menemukan kesalahan pada pemberian nama form yang berbeda, sehingga saat dilakukan pengecekan *error* tersebut lolos.

Kamis, 11 Oktober 2018

Pada hari ini penulis kembali melakukan *testing* saat melakukan pembuatan tiket pelaporan sebagai *guest*. Saat tiket berhasil dibuat dan data

tersimpan di dalam *database*, namun kolom nama pembuatnya tidak muncul saat ditampilkan dalam *datatable report*. Bug ini terjadi karena sebelumnya kolom tersebut hanya diberikan data dari kolom *username*, dimana kolom *username* ini berisi data dari *user* yang telah terdaftar didalam SpeakUp. Sedangkan selain itu disimpan ke dalam kolom *email_guest*.

Oleh karena itu penulis memperbaiki relasi yang telah dibuat, lalu melakukan pengecekan terhadap data dari *database* apakah memiliki data *email_guest* atau tidak. Jika tidak memiliki maka kolom nama pembuat di *report* akan berisi nama *user* yang terdaftar. Jika memiliki data maka kolom nama pembuat akan berisi email *guest*.

Jumat, 12 Oktober 2018

Pada hari ini penulis diberikan *request* baru untuk mengubah beberapa deskripsi dari form yang ada di web. Selain itu penulis diminta untuk membuat sebuah *link* yang mengarah pada halaman alur kerja dari web SpeakUp. Saat link dipilih maka akan mengarah ke halaman khusus alur kerja, dimana akan menampilkan gambar alur kerja dari SpeakUp.

Selain itu penulis juga mendapatkan *request* dalam perubahan alur status tiket. Dimana status pada tiket pelaporan akan berubah terdiri dari REGS, PEND, OPEN, CLOS, dan REJ. REGS adalah status pertama saat tiket pelaporan berhasil dibuat dan tiket dapat dilihat oleh administrator. PEND adalah status setelah administrator sudah melakukan pengecekan dan meminta persetujuan dari *Division Head*. OPEN adalah status jika tiket pelaporan disetujui oleh *Division Head* dan tiket pelaporan akan diberikan kepada PIC. CLOS adalah status dimana tiket pelaporan yang telah diatasi PIC dengan pengawasan administrator telah selesai dikerjakan. REJ adalah status saat tiket pelaporan yang sebelumnya pada status PEND ditolak oleh *Division Head*. Untuk itu diperlukan beberapa email baru untuk memberikan informasi kepada pelapor bahwa tiket pelaporannya disetujui atau tidak.

Senin, 15 Oktober 2018

Pada hari ini penulis mendapatkan *request* untuk menambahkan persetujuan dari *authorizer* saat melakukan penambahan PIC dan juga perubahan email PIC (karena mengingat jika email PIC berubah maka akan dibuat sebuah data baru), serta saat melakukan penambahan dan perubahan pada sub kategori. Sehingga saat administrator melakukan fungsi diatas, maka web akan langsung mengirimkan sebuah email untuk *authorizer* melakukan *approval*.

Penulis membuat email *authorizer* untuk melakukan *approval* dengan memasukkan *link* ke dalam button email yang mengarah ke halaman untuk melakukan editing sub kategori maupun PIC. Di dalamnya *authorizer* dapat memberikan *approval* nya dengan cara mengubah *flag active* yang ada. Jika *flag active* nya bernilai *yes*, maka *authorizer* setuju dengan permintaan pembuatan atau pun perubahan pic dan sub kategori. Jika bernilai *no*, maka *authorizer* tidak setuju.

Karena perubahan pada PIC maupun sub kategori dapat berdampak pada alur kerja pembuatan laporan, maka akan dilakukan pengecekan pada saat akan memilih PIC maupun sub kategori. Pengecekan ini akan menampilkan semua data yang memiliki *flag active yes*.

Selasa, 16 Oktober 2018

Pada hari ini penulis melakukan *testing* terhadap perubahan yang terjadi hari sebelumnya. Penulis juga melakukan pengecekan terhadap *datatables* dari PIC maupun sub kategori. Karena semua data masih ditampilkan maka penulis berinisiatif untuk mengubah struktur *datatable*, dimana *flag active* akan ditampilkan. Jika data memiliki *flag active* dengan nilai *no*, maka baris tersebut akan dibuat berwarna merah. Jika *flag active* bernilai *yes* maka tidak ada perubahan warna pada baris tersebut. Jika *authorizer* belum melakukan *approval*, maka kolom *flag active* akan berisi *pending*. Selain itu, penulis juga diminta untuk mengubah beberapa kata – kata SpeakUp menjadi ACC Whistle.

Rabu, 17 Oktober 2018

Pada hari ini penulis melakukan *Daily Scrum* untuk melaporkan hasil perubahan yang dilakukan oleh penulis. Penulis memperlihatkan hasil perubahan sesuai dengan permintaan di hari sebelumnya. BSA akhirnya menyetujui hasil dari perubahan ini dan proyek ACC Whistle ini berlanjut masuk ke dalam proses PIR (*Project Implementation Review*). PIR ini adalah tahap yang dilakukan setelah proyek selesai dikerjakan. Bertujuan untuk mengukur apakah proyek yang dikerjakan sesuai atau memenuhi tujuan dari dibuatnya proyek ini. Selain itu juga berguna untuk mengukur efisiensi dari penggunaan hasil proyek ini dan memastikan bahwa perusahaan mendapatkan untung dari penerapan hasil proyek ini.

Kamis, 18 Oktober 2018

Setelah masuk ke dalam proses PIR, maka penulis melakukan mempelajari kembali OutSystems dari awal. Penulis memulai dengan pembuatan *database* dan juga *block* untuk membuat aplikasi *mobile* menggunakan OutSystems.

Jumat, 19 Oktober 2018

Pada hari ini penulis melanjutkan mempelajari OutSystems pada bagian pembuatan tema dan UI (*User Interface*) pada aplikasi *mobile*. Pada bagian tema, dijelaskan bagian – bagian dari *layer interface* yang dapat digunakan pada aplikasi *mobile*. Pada UI nya akan diajarkan mengenai UI *default* yang biasanya dimiliki oleh aplikasi *mobile*, seperti tampilan *loading* saat aplikasi pertama kali terbuka, UI *login* dan juga *register* bagi *user* yang menggunakan aplikasi *mobile* ini.

Senin, 22 Oktober 2018

Pada hari ini penulis melanjutkan mempelajari OutSystems untuk bagian selanjutnya yaitu membuat *list* dari data yang ada dalam aplikasi *mobile*. *List*

ini nantinya akan melakukan terlebih dulu apakah hasil dari mengambil data dari *database* yang tersambung dengan aplikasi. Jika terjadi *error* pada saat mengambil data maka akan ditampilkan informasi bahwa terjadi *error*, jika data yang diambil kosong akan ditampilkan informasi bahwa *database* kosong, sedangkan jika berhasil mengambil data maka akan ditampilkan langsung dalam bentuk *list*.

Selain membuat *list* ini, penulis juga membuat halaman untuk detail dari data yang ditampilkan. Pada saat memilih salah satu data dari *list* maka akan menampilkan detail dari data yang dipilih.

Selasa, 23 Oktober 2018

Pada hari ini, penulis melanjutkan mempelajari OutSystems pada tahap membuat validasi input. Validasi ini bertujuan agar inputan yang dibuat memiliki format yang sesuai dengan kolom dari *database*. Validasi input ini ditujukan pada saat membuat atau menambah ataupun mengubah data untuk *database* yang dipilih. Dari hasil validasi ini dapat diberikan sebuah kondisi dimana pada saat terjadi kesalahan format maka aplikasi akan melemparkan informasi *error*.

Rabu, 24 Oktober 2018

Pada hari ini, penulis melanjutkan kembali mempelajari OutSystems yang memasuki tahap pembuatan *style sheet*. Pada tahap ini *developer* diminta untuk fokus terhadap tampilan dari aplikasi. Adapun pada tahap ini penulis diajarkan untuk mengubah judul dari halaman aplikasi yang sedang dibuka sesuai dengan *flowchart* yang telah dibuat. Lalu menggunakan CSS(*Cascading Style Sheet*) untuk merubah tampilan dari sebuah tombol ataupun merubah warna dari sebuah *icon*.

Kamis, 25 Oktober 2018

Pada hari ini penulis melanjutkan mempelajari OutSystems yang memasuki tahap melakukan *generate* aplikasi *mobile*. Aplikasi dari OutSystems

ini dapat *degenerate* kedalam 2 buah *platform*, yaitu Android dan iOS. Perlu diperhatikan setiap kali ingin menggunakan aplikasi hasil dari *generate user* harus sudah terdaftar sebagai *user* yang dapat mengakses proyek OutSystems ini, jika tidak maka aplikasi hanya dapat diunduh namun gagal untuk dibuka.

Jumat, 26 Oktober 2018

Penulis kembali melanjutkan mempelajari OutSystems yang memasuki tahap pembuatan *Block*. *Block* adalah sebuah *container* yang dapat menampung *component*. Sebagai contoh, pada *list* yang dibuat sebelumnya hanya berisi sebuah *component* teks, namun dengan menggunakan sebuah *block developer* dapat membuat sebuah tampilan dari setiap *list* yang ada dengan menambahkan beberapa *component* yang dikira diperlukan untuk memberikan informasi yang mudah dimengerti oleh user.

Pada hari ini juga penulis bertemu dengan tim dari Jakarta untuk membahas sebuah proyek baru. Penulis mendapatkan proyek baru yang bernama Family Games. Proyek ini digunakan untuk menjalankan sebuah permainan pada sebuah acara di ACC, permainan ini berjalan dengan cara setiap user yang sudah terdaftar harus mengumpulkan *point* sebanyak mungkin dengan melakukan scan QR (*Quick Response*) Code menggunakan *smartphone* yang sudah terpasang aplikasi *mobile* yang terintegrasi dengan web. Dalam proyek ini penulis akan membuat web *backend* yang digunakan untuk menyimpan data user dan *smartphone* yang akan digunakan, dan juga melakukan *generate QR Code* yang berisikan *point* maupun tantangan. Melalui web ini juga admin dapat menentukan kapan permainan berhenti dan juga selesai dengan menonaktifkan seluruh QR Code yang tersebar. Proyek ini diharapkan dapat digunakan 2 minggu setelah hari ini.

Senin, 29 Oktober 2018

Pada hari ini penulis memulai pembuatan proyek baru yaitu proyek Family Games. Karena rentang waktu yang diberikan sangat cepat penulis dan rekan penulis kembali memastikan alur dari proyek ini. Penulis dan rekan

penulis juga membahas mengenai bagaimana tampilan yang diinginkan dari sisi web dan juga *mobile*, dan memutuskan untuk menggunakan sebuah *template* untuk bagian web agar pengerjaan data dapat dikerjakan lebih cepat. Sedangkan pada bagian *mobile* akan dibentuk sesuai dengan *Technical Design* yang diberikan

Selasa, 30 Oktober 2018

Pada hari ini penulis melanjutkan pembuatan proyek Family Games dengan membuat menu – menu yang nanti akan digunakan. Karena penulis menggunakan *platform* Laravel maka menu dan fungsi pertama yang dibuat ialah *Login* ke dalam web Family Games. Untuk melakukan *login* ini, penulis membuat database lokal sementara yang berisi *username* dan *password* untuk melakukan login. Setelah membuat fungsi *login*, penulis melanjutkan dengan membuat menu *Player Registration*.

Pada menu ini, dijelaskan di dalam *Technical Design* bahwa pengguna dapat melakukan CRUD (*Create, Read, Update, Delete*) *player*. Beberapa data *player* yang akan disimpan seperti *username, password, team_name, imei_devices, login_status*. *Player* adalah setiap tim yang akan mengikuti acara Family Games nantinya dan diwajibkan mendaftarkan tim beserta *imei device* yang digunakan untuk bermain nantinya, sehingga setiap tim hanya memiliki 1 buah *device* yang dapat digunakan.

Rabu, 31 Oktober 2018

Pada hari ini penulis beserta rekan penulis melakukan *video conference* dengan tim di Jakarta yang bertanggung jawab terhadap proyek Family Games ini. Kami dijelaskan mengenai alur kerja dari penggunaan web dan aplikasi mobile Family Games ini. Alur permainan dari Family Games ini ialah, setiap tim berlomba untuk mendapatkan point sebanyak – banyak selama waktu yang ditentukan oleh panitia. Tim yang dibentuk diharapkan sudah terdaftar sebagai *Player* dan dapat melakukan *login* menggunakan aplikasi *mobile*. Panitia akan menyebarkan beberapa *QR Code* yang berisi point maupun tantangan yang

dapat didapatkan oleh *Player* dengan cara scan QR Code yang ada menggunakan aplikasi *mobile* yang dibuat. Tim yang mendapatkan point terbanyak sampai habis waktu akan menjadi pemenang. Karena membutuhkan interaksi antara web dan aplikasi *mobile* maka tim di Jakarta akan menangani pada bagian pembuatan API (*Application Program Interface*). Sehingga semua data yang akan digunakan saat Family Games akan diakses menggunakan API yang dibuat tim di Jakarta. Penulis juga memberikan saran untuk menghilangkan fungsi *login* pada web, karena user yang akan menggunakan web admin atau panitia penyelenggara Family Games dan saran dari penulis diterima oleh seluruh tim.

Selain itu penulis juga diberikan penjelasan lebih lanjut apa kegunaan dari setiap menu web yang ada. Menu *Home* digunakan untuk menampilkan data perolehan *point* sementara dari *player*. Menu *Player* digunakan untuk melakukan CRUD *player* seperti yang dijelaskan diatas. Menu *point* digunakan untuk melakukan CRUD *point* dan juga mengubah *Global Status*. *Point* yang digunakan pada Family Games ini terdapat 2 jenis, yaitu *Special* dan *Non-Special*. *Point Non-Special* hanya berisi *point* saja. *Point Special* berisi tantangan yang harus dikerjakan oleh *Player*, jika berhasil menyelesaikan tantangan maka panitia akan memberikan QR Code yang berisi *point* yang didapat. *Global Status* digunakan sebagai acuan apakah waktu permainan sudah habis atau belum dengan cara mengubah data *flag_free* seluruh *point* sehingga tidak dapat discan lagi. Menu *Reset Data* digunakan untuk melakukan penghapusan data perolehan *point* dari setiap *Player*.

Setelah melakukan *video conference*, penulis kembali melanjutkan pengerjaan web Family Games. Penulis menghilangkan fungsi *login* dari web sehingga saat pertama kali web dibuka akan memunculkan menu *home* yang menampilkan perolehan *point* sementara. Karena API yang dimaksud belum dibuat maka penulis dianjurkan untuk mencobanya melalui data lokal terlebih dahulu.

Kamis, 1 November 2018

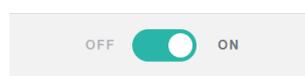
Penulis melanjutkan pengerjaan proyek Family Games, dimulai dengan membuat menu *Point*. Pada pembuatan menu *point* ini, penulis masih menggunakan *database* lokal. Penulis membuat fungsi CRUD untuk menu *Point*, mencoba apakah data yang dibuat dan diubah dapat tersimpan dengan baik di *database*. Penulis juga memperbaiki tampilan halaman *create Point* maupun *update Point* agar lebih dinamis sesuai dengan jenis dari *point* yang dibuat. Jika jenis *pointnya Special* maka akan muncul *field Point* dan *radio button* untuk memberikan nilai *flag_free* pada *Point* tersebut. Jika jenis *pointnya Non-Special* maka akan muncul *field Question* dan *radio button flag_free*.

Jumat, 2 November 2018

Pada hari ini penulis merapikan tampilan halaman *Point* agar lebih dinamis terhadap jenis *Point* yang dipilih. Penulis juga mencoba berbagai *package* untuk membuat *QR Code* pada laravel. Untuk *generate QR Code* pada web Family Games ini, penulis menggunakan *package simple-qrcode*. *Package* ini dapat membuat *QR Code* dengan besaran yang dapat ditentukan oleh *developer* dan juga dapat mengubah hasil *QR Code* kedalam format *image* tertentu. Penulis membuat agar *QR Code* yang dihasilkan memiliki format *png* dengan ukuran 300 yang berisi *id* dari *Point* yang ditunjuk.

Senin, 5 November 2018

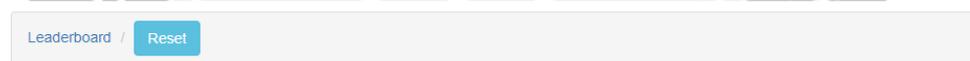
Pada hari ini penulis dengan rekan mencoba melakukan *scanning QR Code* yang dibuat di web Family Games menggunakan aplikasi *mobile Family Games* yang dibuat oleh rekan penulis. Setelah berhasil dibaca, maka penulis melanjutkan pengerjaan dengan melakukan pembuatan tampilan untuk mengubah *Global Status*. Penulis mencoba untuk menggunakan *slide button* pada bootstrap untuk melakukan perubahan *Global Status*.



Gambar 2. 9 Contoh *slide button*

Selasa, 6 November 2018

Pada hari ini penulis melanjutkan pembuatan web Family Games dengan membuat menu *Reset Data*. *Reset Data* digunakan untuk menghapus semua data transaksi atau data *Point* yang didapat oleh *Player* saat bermain Family Games. Karena untuk melakukan transaksi ini dilakukan melalui mobile maka penulis berinisiatif untuk melakukan *seeding*, yaitu pengisian data ke tabel secara otomatis sesuai dengan jenis kolomnya. Dengan adanya beberapa data pada tabel transaksi *Point*, maka penulis dapat memastikan bahwa fungsi *reset* data dapat berjalan dengan lancar. Selain memberikan menu *Reset Data* sendiri, penulis juga memberikan fungsi ini pada *breadcrumbs navigation* pada saat menampilkan data perolehan *Point* sementara.



Gambar 2. 10 Contoh *breadcrumbs*

Rabu, 7 November 2018

Pada hari ini penulis menunggu API yang dibuat untuk web Family Games. Sembari menunggu, penulis kembali memperbaiki tampilan pada menu *Point*. Alur dari pembuatan *Point* sebelumnya sebagai berikut, *user* masuk ke halaman untuk menyimpan *Point* baru lalu kembali ke halaman *list Point* dan harus masuk kedalam halaman *edit Point* untuk melihat *QR Code* yang terbentuk. Penulis akhirnya mengubah alur pembuatan *Point* menjadi menampilkan *QR Code* terlebih dulu setelah berhasil menyimpan data *Point*. Penulis juga merubah data yang disimpan di dalam *QR Code* menjadi id dari data *Point* yang ditunjuk.

Kamis, 8 November 2018

Pada hari ini penulis masih menunggu API yang akan digunakan pada web Family Games ini. Penulis akhirnya melakukan *testing* untuk menu – menu yang telah dibuat. Penulis menemukan error pada bagian perubahan Status

Global yang pada tampilannya menggunakan *slide button*. Penulis menemukan bahwa data dari Status Global tidak berubah. Akhirnya penulis mencoba mengubah tampilan untuk mengubah Status Global menggunakan *radio button*, dan fungsi berjalan dengan baik.

Jumat, 9 November 2018

Pada hari ini penulis bersama rekan penulis melakukan *video conference* dengan tim di Jakarta untuk melaporkan hasil pengerjaan. Kami melaporkan bahwa baik web maupun *mobile* untuk Family Games sudah siap untuk digunakan, hanya menunggu API yang dibuat oleh tim di Jakarta. Karena API belum dibuat sama sekali, kami diminta untuk membuat *list* apa saja fungsi API yang dibutuhkan. Adapun untuk web sendiri terdapat beberapa API yang diperlukan, seperti menampilkan semua *Player*, menampilkan data detail *Player*, menghapus *Player*, mengubah data *Player*, pengecekan ketersediaan *Player*, pembuatan *Player* baru, menampilkan semua *Point*, menampilkan data detail *Point*, menghapus *Point*, mengubah data *Point*, pembuatan *Point* baru, mengetahui Status Global terbaru, mengubah Status Global, melihat keseluruhan perolehan *Point* sementara dan melakukan *reset* data.

Senin, 12 November 2018

Pada hari ini penulis melanjutkan pengerjaan web Family Games dengan mencoba melakukan *consume* API karena pada hari ini juga beberapa API sudah diberikan. Penulis menggunakan *package* Guzzle untuk melakukan *consume* API. Penulis mencoba untuk *consume* API yang berfungsi menyimpan *Player* baru ke dalam *database* pusat. Karena terhalang masalah ijin untuk melihat *database*, maka penulis menunggu kabar dari tim Jakarta apakah penggunaan API sudah berhasil atau belum. Setelah mendapatkan kabar bahwa data berhasil disimpan, penulis melanjutkan penggunaan API pada saat menyimpan data *Point*.

Selasa, 13 November 2018

Pada hari ini penulis kembali mendapatkan beberapa API, seperti menampilkan semua data *Point* dan *Player*, melakukan *delete Point* dan *Player*. Penulis melakukan *consume* API yang disebutkan diatas sekaligus memastikan apakah semua fungsi berjalan dengan baik karena sekarang penulis dapat melihat data yang dibuatnya melalui API yang diberikan.

Rabu, 14 November 2018

Pada hari ini penulis mendapatkan beberapa API untuk bagian *update* data, baik data *Player*, *Point* maupun Status Global. Setiap melakukan *update Player* maupun *Point* diperlukan API yang memiliki hasil balikan detail dari data yang dipilih melalui *list*. Sedangkan untuk Status Global, diperlukan API yang memberikan kondisi Status Global saat ini.

Kamis, 15 November 2018

Pada hari ini penulis diberikan beberapa API yaitu untuk melakukan reset data dan juga menampilkan perolehan *Point* sementara. Penulis juga mendapatkan permintaan dari rekan penulis untuk membuat data yang disimpan pada QR Code bukan lagi nomor id namun id string unik dari *Point*. Ini dimaksudkan untuk menyesuaikan inputan *body* API yang digunakan di aplikasi *mobile* yang meminta inputan id *string* unik. Hal ini juga menyesuaikan dengan contoh data yang sudah tersimpan pada *database* utama.

Jumat, 16 November 2018

Pada hari ini, penulis dan rekan penulis mencoba permainan Family Games ini. Mulai dari melakukan pendaftaran *Player* menggunakan *device* dari rekan penulis, membuat *Point*, melakukan scanning hingga menampilkan *point* sementara hasil dari melakukan *scanning*. Kami juga melakukan *video conference* dengan tim di Jakarta untuk melaporkan bahwa web maupun aplikasi mobile untuk Family Games sudah siap digunakan (tidak ada error) dan melakukan *testing* bersama tim di Jakarta.

Senin, 19 November 2018

Pada hari ini, penulis mempelajari OutSystems karena akan di-assign ke proyek ACC Me. OutSystems yang dipelajari adalah pada modul *mobile* karena sebelumnya hanya sempat mempelajari modul web. ACC Me sendiri adalah sebuah aplikasi yang menangani bagian sales atau proses AA. Karena ada pada ranah awal sebuah transaksi di ACC dilakukan, maka sasaran pengguna dari aplikasi ini adalah para sales yang akan menyimpan data para calon pelanggan.

Rabu, 21 November 2018

Pada hari ini, penulis diberikan waktu untuk mempelajari alur kerja dari penggunaan ACC Me. Pada hari ini, penulis mendapatkan kesulitan karena tidak memiliki izin untuk mengunduh aplikasi ini karena aplikasi tersebut belum diregistrasikan untuk *smartphone* dengan sistem operasi yang penulis gunakan yaitu iOS. Pada hari ini penulis mencoba mencari cara agar dapat mengunduh aplikasi yang telah diberikan dengan format ipa (*iOS App Store Package*).

Kamis, 22 November 2018

Pada hari ini penulis kembali mencari cara agar dapat mengunggah aplikasi *mobile* dari ACC Me untuk iOS. Pada hari itu juga penulis dibantu dari tim yang di Jakarta dengan cara mendaftarkan id dari *smartphone* yang dimiliki oleh penulis. Sembari menunggu hasil setelah mendaftar, penulis melanjutkan untuk mencoba mengunggah aplikasi melalui cara lain.

Jumat, 23 November 2018

Pada hari ini penulis masih mengalami permasalahan yang sama dari beberapa hari yang lalu, berinisiatif untuk mencoba memahami penggunaan ACC Me dengan meminjam *smartphone* rekan penulis yang memiliki sistem operasi Android. Pada hari yang sama penulis mencoba melakukan tahapan yang diberikan oleh tim di Jakarta untuk memahami aplikasi ACC Me. Pada saat memahami alur penggunaan aplikasi tersebut, penulis mendapat kabar

bahwa proyek aplikasi ACC Me ini dirasa terlalu besar untuk menggunakan tenaga karyawan magang. Oleh karena itu, penulis akan di-assign sebuah proyek lainnya.

Senin, 26 November 2018

Pada hari ini, penulis melakukan *video conference* dengan tim proyek baru yang akan diassignkan kepada penulis. Proyek yang akan diberikan kepada penulis ialah pembuatan Requisition Web. Requisition Web sendiri digunakan untuk melakukan *request* dari sebuah divisi ke divisi lain. Pengerjaan proyek ini akan menggunakan *framework* Laravel dengan PHP versi 5.4. Pada *video conference*, penulis diberi tahu mengenai alur kerja dari web ini dan menanyakan apa saja yang kira – kira akan dibutuhkan dalam pengerjaan web ini nantinya.

Rabu, 28 November 2018

Pada hari ini, penulis memulai pengerjaan proyek dengan membuat tampilan dari web yang akan dibentuk. Untuk memastikan apakah tampilan yang diajukan oleh penulis mendapat persetujuan dari tim yang ada di Jakarta, penulis memberikan gambar potongan dari tampilan web yang akan dibentuk. Setelah mendapatkan persetujuan dari tim di Jakarta, penulis memulai pembuatan tampilan awal dari requisition web ini dengan awalan membuat tampilan menu dan juga *login* user. Sembari membuat tampilan dari *requisition web*, penulis juga mempelajari pengambilan data yang akan digunakan.

Kamis, 29 November 2018

Pada hari ini, penulis melanjutkan pengerjaan *requisition web* dengan membuat tampilan untuk menu *Add Requisition*. Pada menu *add requisition* ini dibagi menjadi 2 sub bagian yaitu *header* dan *body*. Pada *header*, akan berisi data dari *user* yang *login* untuk melakukan permintaan. Sedangkan pada *body*, berisi data permintaan yang diinginkan oleh *user*. Penulis diberikan *design* dari

halaman *Add Requisition* ini dan mencoba untuk mengikuti sesuai dengan design yang diberikan.

Jumat, 30 November 2018

Pada hari ini penulis kembali melanjutkan pembuatan tampilan untuk menu *Add Requisition*. Pada hari ini juga penulis melakukan *video conference* dengan tim di Jakarta untuk membahas tentang penggunaan data untuk Web Requisition ini. Penulis diberikan tampilan data mana saja yang akan digunakan untuk membuat sebuah permintaan dan bagaimana alur untuk mendapatkan data yang diinginkan.

Senin, 3 Desember 2018

Penulis yang telah usai mengerjakan pembuatan tampilan untuk menu *Add Requisition*, melanjutkan pengerjaan Web Requisition dengan membuat data *master user*. Data master ini akan menyimpan data dari *user* yang hendak menggunakan web ini. Penulis membuat sebuah *database* baru untuk menyimpan data *master* ini yang nantinya akan dimigrasi melalui *framework* Laravel. Penulis mencoba untuk memasukkan data *dummy* sebagai administrator yang bekerja untuk melakukan pengelolaan data *master user* nantinya. Penulis mencoba untuk melakukan login menggunakan data *dummy* administrator yang baru saja dibuat.

Selasa, 4 Desember 2018

Pada hari ini, penulis mendapatkan informasi terbaru dari tim Jakarta. Informasi ini meminta penulis untuk mengubah beberapa alur kerja dari pengelolaan data *user* di Web Requisition. Data *user* yang dapat dibuat dan dikelola harus melalui pengecekan dari tabel yang diberikan oleh tim. Tabel ini merupakan *clone* dari database *master user* yang ada di *database* pusat. Penulis diminta agar saat membuat *user* sebagai pengguna Web Requisition ini dilakukan pengecekan, apakah *user* yang dimaksud sudah terdaftar sebagai karyawan di Astra Credit Companies atau tidak. Jika data *user* yang diinputkan

sesuai dengan data karyawan maka inputan akan disimpan, jika bukan salah satu karyawan maka inputan akan gagal dan tidak akan disimpan.

Rabu, 5 Desember 2018

Pada hari ini, penulis kembali melanjutkan pengerjaan Web Requisition untuk menu *Master User*. Melalui menu ini, administrator dapat melakukan pengelolaan data *master user*. Administrator dapat menyimpan data karyawan sebagai *user* Web Requisition, melakukan perubahan data hingga menghapus data *user*. Pada hari ini penulis memulai pengerjaan menu ini dengan membuat tampilannya terlebih dulu. Adapun data yang perlu diinputkan melalui menu ini ialah npk, nama, *password* dan juga *role*. *Role user* dibedakan menjadi 2, yaitu admin dan *user*. *User* dengan *role* admin nantinya dapat mengakses semua menu, sedangkan untuk *user* dengan *role user* hanya dapat mengakses menu Add Requisition.

Kamis, 6 Desember 2018

Pada hari ini, penulis memperbaiki tampilan dari menu *master user* sekaligus melakukan pengecekan terhadap fungsi – fungsi yang ada di menu *master user*. Setelah fungsi menambah *user* baru berhasil dilakukan, penulis mencoba untuk melakukan *login* menggunakan data *user* yang baru saja dibuat. Penulis juga melakukan pengecekan saat melakukan *edit* maupun *delete* data *user* yang ada dan fungsi dapat berjalan dengan baik. Penulis melanjutkan dengan kembali mempelajari cara pengambilan data yang akan dibutuhkan pada *header requisition*.

Jumat, 7 Desember 2018

Pada hari ini, penulis meneruskan pengerjaan Web Requisition pada bagian tampilan detail *requisition*. Penulis mendapatkan permintaan agar tampilan di detail *requisition* tidak terlalu besar namun tetap dapat dibaca dengan baik oleh *user*. Penulis juga mendapatkan permintaan untuk menangani jika terdapat lebih dari 1 data detail yang akan dikirimkan. Penulis mendapatkan

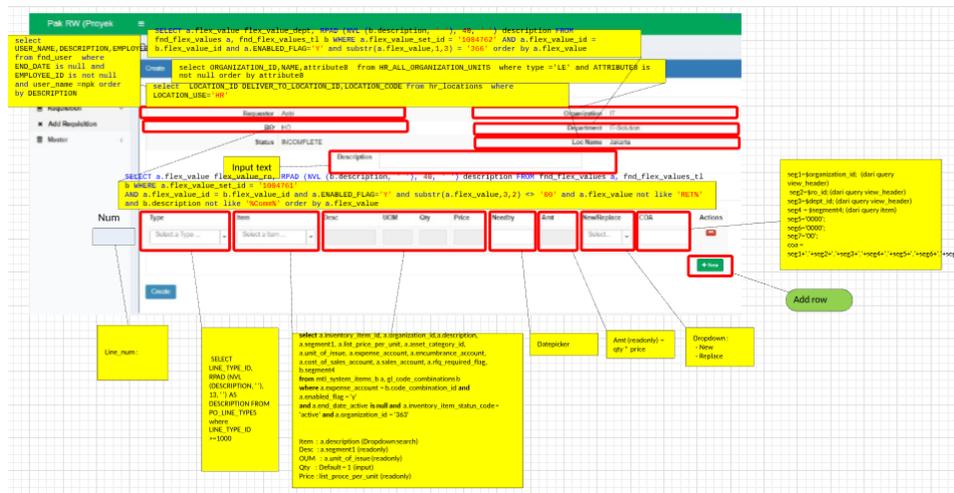
2 buah pilihan, yang pertama data *requisition* akan ditampung ke dalam sebuah table penampung sementara bernama *wishlist* atau pilihan kedua yaitu dengan membuat tampilan form detail dapat ditambahkan atau digandakan.

Senin, 10 Desember 2018

Penulis hari ini kembali berfokus pada penanganan saat terdapat lebih dari 1 detail *requisition* yang harus dikirimkan. Penulis mencoba untuk membuat agar web dapat menggandakan form detail *requisition* dan mengirimkannya secara langsung melalui API. Setelah berhasil melakukan penggandaan form detail melalui sebuah tombol, penulis melanjutkan untuk merapikan ataupun merapatkan setiap kolom pengisian untuk form detail karena tim masih merasa bahwa tampilan dari form detail masih terlalu renggang, sehingga membutuhkan banyak *space* dalam 1 halaman.

Selasa, 11 Desember 2018

Setelah berhasil menggandakan form detail *requisition*, penulis kembali melakukan pengecekan terhadap data – data yang dapat diinputkan di menu *Add Requisition* ini. Sebenarnya pada form *header*, kolom – kolom yang ada seharusnya langsung terisi sesuai dengan data mengenai *user*. Namun, tim memberikan saran kepada penulis untuk membuat kolom – kolom tersebut kosong dan menjadi inputan yang harus diinput oleh user setiap kali ingin melakukan permintaan. Data – data yang diambil ini tidak hanya untuk ditampilkan pada form *header* saja, namun pada form detail juga perlu ditampilkan. Ada beberapa data yang nantinya digunakan untuk menghasilkan data unik baru untuk setiap *requisition* yang dibuat.



Gambar 2. 11 Design data yang digunakan untuk requisition

Rabu, 12 Desember 2018

Pada hari ini, melakukan pengecekan apakah data yang ditampilkan dari form requisition sudah sesuai dengan yang diinginkan. Terutama untuk id unik yang terbentuk dari beberapa data yang dipilih. Pada hari ini juga, penulis melakukan *video conference* dengan tim Jakarta untuk memastikan data – data yang digunakan sesuai dengan yang diharapkan. API yang digunakan untuk mengirimkan permintaan dan menyimpannya ke dalam *database* pusat yang dikerjakan oleh tim Jakarta, masih dalam proses.

Kamis, 13 Desember 2018

Pada hari ini, penulis hanya menunggu API yang dikerjakan oleh tim Jakarta. Sembari menunggu API yang akan digunakan jadi, penulis diminta untuk merapikan kembali tampilan yang dirasa masih terlalu renggang. Penulis juga diminta untuk mempelajari prosedur yang digunakan API untuk menyimpan data *requisition*, sekaligus mencari apakah ada kesalahan pada prosedur tersebut atau tidak.

Jumat, 14 Desember 2018

Penulis pada hari ini menemukan sebuah *bug* pada saat menekan tombol yang menggandakan form detail yang telah terisi sebuah data, hasil penggandaan juga memiliki pilihan yang sama dengan form detail sebelumnya. Penulis mencari cara lain untuk menyelesaikan *bug* ini. Penulis juga mengabarkan *bug* ini kepada tim Jakarta. Solusi – solusi yang diberikan dan yang penulis temukan dari *website*, tidak dapat berjalan sesuai yang diharapkan. Akhirnya penulis mengusulkan untuk menggunakan cara lain, yaitu dengan menyimpan permintaan tersebut ke dalam sebuah tabel sementara bernama *wishlist*. Setelah tersimpan di *wishlist*, nantinya setiap data yang ada di *wishlist* dapat langsung disimpan ke database pusat menggunakan API yang sedang di kerjakan. Penggunaan cara ini jauh lebih mudah, namun alur kerja yang perlu dilakukan akan bertambah.

Senin, 17 Desember 2018

Setelah mengusulkan perubahan alur kerja untuk menyimpan data *requisition*, penulis pada hari ini mencoba untuk merealisasikannya. Penulis memulai dengan membuat table *wishlist* dengan kolom – kolom sesuai dengan data inputan *requisition*. Dengan alur kerja yang berbeda ini, penulis berfikir bahwa *wishlist* yang disimpan sementara oleh *user*, dapat dilakukan perubahan maupun dihilangkan. Pada hari ini juga penulis membuat fungsi CRUD untuk *wishlist*.

Selasa, 18 Desember 2018

Hari sebelumnya, penulis membuat fungsi CRUD untuk *wishlist* di beberapa halaman yang berbeda, namun hari ini penulis mendapatkan permintaan dari tim Jakarta agar halaman yang digunakan tidak terlalu banyak. Penulis akhirnya memasukkan *datatable* yang menampilkan semua data *wishlist* dari *user* yang *login* ke dalam halaman *Add Requisition*. Sehingga tampilan halaman *Add Requisition* berisi form *header*, form detail dan juga *datatable wishlist*.

Rabu, 19 Desember 2018

Pada hari ini, penulis melakukan *video conference* dengan BSA karena selama ini *video conference* hanya dengan tim *developer* Jakarta. Saat melakukan *video conference* penulis diminta untuk melakukan *demo* Web Requisition, dimulai dari menu *master user*, hingga *add requisition*. Hasil dari *video conference*, BSA menginginkan agar tampilan dibuat lebih rapat lagi dan juga tampilan untuk *create* dan *edit wishlist* dimasukkan ke dalam 1 halaman saja yaitu halaman pada menu *Add Requisition*. Sehingga pada saat tombol *edit* pada sebuah baris *wishlist*, tidak akan berpindah halaman hanya kembali mengisi form *header* dan form detail dengan data sesuai dengan baris yang dipilih. BSA juga menemukan kesalahan pada id unik yang terbentuk dan meminta penulis untuk melakukan pengecekan apakah sudah sesuai dengan design yang diberikan.

Kamis, 20 Desember 2018

Pada hari ini, penulis meneruskan pembuatan permintaan dari BSA untuk membuat fungsi *create* dan *edit wishlist* pada sebuah halaman saja. Penulis juga diminta oleh tim *developer* Jakarta untuk menyesuaikan struktur data dari tabel *wishlist* dengan data yang dibutuhkan prosedur untuk memanggil API.

Jumat, 21 Desember 2018

Pada hari ini, penulis masih meunggu API untuk menyimpan data *requisition* ke *database* pusat, sembari melakukan pengecekan terhadap permintaan dari BSA di *video conference* sebelumnya. Penulis juga menyiapkan kode untuk *consume* API nanti. Dimana API akan dipanggil berulang sesuai dengan jumlah data *wishlist* yang ada. Penulis juga membuat halaman untuk mengirimkan email. Pada halaman tersebut *user* diminta untuk memasukkan inputan mengenai alamat email yang akan mendapatkan pesan *requisition*.

Rabu, 26 Desember 2018

Pada siang hari, penulis baru mendapatkan API untuk menyimpan *wishlist requisition*. Penulis melakukan pengecekan pertama menggunakan Postman dan dinyatakan berhasil, tidak ada *error* saat menggunakan API. Penulis melanjutkan untuk mencobanya menggunakan web. Pada saat data yang ada dalam *wishlist* hanya ada 1 buah, API berjalan dengan baik. Kemudian penulis mencoba untuk menggunakan API dengan kondisi terdapat beberapa *wishlist* dan tidak ditemukan *error*, hanya saja pada saat dilakukan pengecekan terhadap database pusat, hanya 1 *requisition* yang tersimpan.

Kamis, 27 Desember 2018

Setelah mendapatkan *error* API yang diberikan, penulis berdiskusi dengan tim *developer* Jakarta untuk memastikan bahwa alur untuk menggunakan API sudah benar. Sembari menunggu perbaikan, penulis mencari apa yang membuat hanya 1 data yang tersimpan.

Jumat, 28 Desember 2018

Akhirnya penulis menemukan yang membuat API tidak bekerja baik, terdapat 1 buah data yang seharusnya tidak digunakan sebagai *body* API. Penulis segera melaporkan temuan *error* ini kepada tim *developer* di Jakarta. Penulis kembali menunggu adanya perbaikan pada prosedur API yang diberikan.

Senin, 31 Desember 2018

Pada hari ini, penulis mendapatkan perbaikan API yang sesuai dengan prosedur yang diberikan. Penulis langsung mencoba menggunakan API yang diberikan. Dengan kondisi yang sama dengan sebelumnya, API sukses menyimpan data *wishlist* saat hanya terdapat 1 buah data. Namun pada saat terdapat lebih dari 1 data *wishlist*, API menampilkan *error*. *Error* ini dikarenakan data *line_num* dari *wishlist* ada yang tidak bernilai 1 sehingga

mengeluarkan *return error*. Penulis segera memberikan informasi kepada tim *developer* Jakarta bahwa terdapat *error* di API pada bagian *line_num*.

2.2. Hasil Pekerjaan Secara Umum

Selama magang di Astra Credit Companies(ACC) pada tanggal 16 Juli hingga 31 Desember 2018 ada beberapa kegiatan yang penulis lakukan. Selama melakukan magang ini, penulis bekerja mengikuti jadwal kerja dari kantor pusat ACC di Jakarta yaitu hari Senin hingga Jumat dengan akumulasi waktu total bekerja dalam 1 harinya selama 8 jam.

Saat memulai magang di ACC ini, penulis dengan rekan – rekan penulis diberikan waktu 2 minggu selama hari kerja untuk mengikuti masa *training* yang dilaksanakan di kantor IT ACC yang berada di ACC cabang Fatmawati, Jakarta. Penulis beserta rekan – rekan penulis ang mengikuti magang, langsung dibagi kedalam 2 kelompok yang beranggotakan 2 orang magang untuk di-*assign* ke dalam proyek. Pada magang ini, penulis berkelompok dengan salah satu rekan bernama Agung Wira untuk mengerjakan proyek web bernama Speakup yang menggunakan *platform* Laravel. Pada masa *training* ini, penulis dan rekan – rekan penulis diberikan wawasan mengenai sejarah Astra hingga cara kerja dari anak perusahaan Astra International ini. Tidak hanya itu, pada minggu pertama *training* penulis dan rekan – rekan juga diberi wawasan mengenai berbagai departemen yang menangani masing – masing proses bisnis yang ada. Pada minggu kedua, penulis dan rekan – rekan diberikan pelatihan mengenai *platform* yang menjadi tugas dari setiap kelompok anggota magang ini. Adapun kami diajarkan mengenai UniPaas dan juga Laravel.

Setelah mengikuti *training*, penulis kembali ke Jogja untuk mengerjakan proyek yang telah diberikan. Dalam pengerjaannya penulis didampingi oleh seorang pembimbing dari IT ACC. Proyek *Speakup* ini dikerjakan dengan metode waterfall yang dibagi ke dalam beberapa tahap, antara lain *Development*, *System Integration Test (SIT)*, *User Acceptance Testing (UAT)*, *Implementation*, dan *Project Implementation Review (PIR)*. Adapun beberapa orang yang bertanggung jawab dalam proyek ini adalah penulis dan rekan Agung Wira sebagai *developer*,

Project Manager (PM), Business and System Analyst (BSA), Quality Control (QC), dan User.

Tahap *development* dilakukan sejak tanggal 30 Juli 2018 hingga 23 Agustus 2018. Setiap fungsi yang akan dibuat untuk proyek web Speakup ini telah dibagi oleh PM kepada peserta magang yaitu penulis dan rekan Agung Wira. Adapun fungsi yang menjadi tugas yang dikerjakan oleh penulis adalah *Create Backend PIC, Table & Search List* Pelaporan, *Push Email*, dan *Enhance Master Categories*.

Tugas *Create Backend PIC* merupakan tugas dimana saya harus membuat fungsi pengelolaan data PIC (*Person In Charge*). Pengelolaan data ini berupa tampil data PIC, buat data PIC baru, ubah data PIC, cari data PIC, dan hapus data PIC. Fungsi pengelolaan ini hanya dapat diakses oleh pengguna yang bertindak sebagai administrator. Adapun saat melakukan pengelolaan PIC akan berdampak kepada table *User* yang berisi data – data dari pengguna yang sudah terdaftar.

Tugas *Table & Search List* Pelaporan adalah tugas yang dimandatkan ke penulis untuk membuat halaman dimana dapat menampilkan data – data pelaporan sesuai dengan pengguna yang masuk saat itu. Setiap pengguna yang memiliki *role* yang berbeda akan memiliki hasil tampilan pelaporan yang berbeda, sesuai dengan keterkatiannya masing – masing. Selain itu penulis juga diminta untuk memberikan memberikan fungsi untuk mencari pelaporan milik pengguna dengan memasukkan nomor id pelaporan.

Tugas *Push Email* merupakan tugas dimana penulis harus membuat fungsi untuk mengirimkan email kepada pengguna yang dituju. Adapun email yang dikirimkan pada saat *Sign Up* untuk melakukan aktivasi agar dapat melakukan *Login*, sedangkan pada saat *Forgot Password* untuk merubah password dari user dengan email yang bersangkutan, pengiriman email setelah berhasil melakukan *Create* Pelaporan untuk memberikan informasi kepada PIC yang bersangkutan dan juga kepada pengguna yang membuat pelaporan, pengiriman email setelah *Create PIC, Edit User* menjadi PIC digunakan untuk merubah password dari PIC tersebut.

Tugas *Enhance Master Categories* merupakan tugas dimana penulis harus membuat mengembangkan fungsi pengelolaan kategori yang sebelumnya dibuat oleh rekan penulis. Pengembangan ini dilakukan setelah diketahui bahwa terdapat proses bisnis yang dapat merusak data dari kategori maupun PIC. Proses bisnis yang dimaksud adalah saat kategori diubah PIC atau didelete, dan atau pada saat PIC dihapus Penulis diminta untuk membuat fungsi untuk menangani proses bisnis tersebut.

Setelah menyelesaikan tahap *development*, pengerjaan proyek dilanjutkan kepada tahap kedua yaitu SIT yang dilaksanakan pada tanggal 24 Agustus 2018 hingga 7 September 2018. Tahap SIT merupakan tahap testing yang dilakukan oleh QC pada *server* dengan *environment* hasil dari *development*. Pada tahap ini *developer* diharapkan selalu *stand by* apabila ditemukan *error* oleh QC dapat segera diperbaiki. Tak menutup kemungkinan juga jika ada perubahan – perubahan yang diinginkan pada saat tahap ini.

Tahap selanjutnya adalah tahap UAT yang dilaksanakan dari tanggal 10 September 2018 – 14 September 2018. Tahap UAT ini merupakan tahap dimana hasil pengerjaan diperlihatkan BSA kepada *User* yang meminta dibentuknya sebuah aplikasi ini. Dalam tahap ini akan dilakukan pengecekan oleh *User* untuk menilai apakah sudah sesuai yang diinginkan atau belum. Tidak menutup kemungkinan juga pada tahap ini *User* meminta beberapa tambahan pada aplikasinya.

Tahap *Implementation* merupakan tahap dimana aplikasi yang telah dibuat akan dipindahkan pada *server production* untuk diimplementasi. Pada tahap ini BSA, PM dan *User* melakukan diskusi mengenai implementasi aplikasi, seperti *domain* yang digunakan, *server* apa yang dipakai. *Developer* tetap dalam bersiap jika ditemukan *bug* atau permintaan perubahan dari *User*.

Tahap setelah *Implementation* adalah tahap PIR yang berlangsung pada tanggal 17 September 2018 hingga 12 Oktober 2018. Tahap ini merupakan masa garansi bagi proyek yang telah diimplementasi. Masa garansi ini berguna ketika ditemukan *bug* oleh pengguna. Pada tahap ini juga ada beberapa perubahan

permintaan atau bahkan penambahan permintaan yang diminta dari *user* dan jika hal ini terjadi akan dimasukkan ke dalam tahap *Change Request* atau *Additional Request*. Adapun yang terjadi pada proyek *SpeakUp* ini, penulis diminta untuk menambahkan sebuah proses bisnis pada saat membuat laporan. Dimana laporan yang dibuat dibutuhkan persetujuan dari *Super Admin* agar dapat diberikan tindak lanjut.

Setelah proyek *SpeakUp* selesai dan telah diimplementasi, penulis mendapatkan proyek lainnya berupa membuat aplikasi web *Family Games*. Dengan tahap yang sama seperti proyek sebelumnya, proyek web *Family Games* hanya dikerjakan dalam waktu 2 minggu saja. Kali ini penulis bekerja sama dengan seorang rekan yang akan mengerjakan pada platform mobile. Proyek ini akan digunakan untuk bermain game dengan cara melakukan scan QR Code. Web yang akan dibuat penulis digunakan sebagai *backend* dari *Family Games* yang memiliki fungsi untuk mengatur data – data yang akan digunakan dalam permainan.

Tahap *development* dilakukan sejak tanggal 1 November 2018 hingga 13 November 2018. Adapun fungsi yang menjadi tugas yang dikerjakan oleh penulis adalah *Create Backend Player*, *Create Backend Point*, *Leaderboard*, *Reset Data*.

Tugas *Create Backend Player* merupakan tugas yang meminta penulis untuk membuat fungsi CRUD (*Create*, *Read*, *Update*, dan *Delete*) data *player*. Setiap *player* memiliki data yang harus diinputkan adalah *username*, *password*, *team name*, *IMEI Device*. Setiap fungsi ini hanya dapat diakses oleh semua panitia pada *family games*.

Tugas *Create Backend Point* adalah tugas yang harus dibuat penulis untuk membuat fungsi CRUD pada data *point*. *Point* dibedakan menjadi 2 buah jenis, yaitu spesial atau tidak. Setiap *point* yang spesial maka akan memiliki tantangan didalamnya, selain itu maka hanya berisi data berapa besar *point* yang ada. Setiap melakukan pembuatan atau perubahan pada data *point* maka web akan *generate* data tersebut ke dalam bentuk QR Code.

Tugas *Leaderboard* merupakan tugas dimana penulis diminta untuk membuat sebuah halaman yang menampilkan perolehan *point* sementara selama permainan berlangsung. *Leaderboard* ini hanya dapat diakses oleh panitia melalui web.

Tugas *Reset Data* merupakan tugas dimana penulis diminta untuk membuat sebuah fungsi yang dapat melakukan penghapusan semua data transaksi *point* yang pernah tersimpan. Fungsi ini akan digunakan pada saat akan dilakukan permainan baru, sehingga aplikasi akan menghapus semua data yang ada menjadi kembali kosong.

Setelah tahap *development* selesai, maka dilanjutkan ke tahap SIT dan UAT secara bersamaan, mengingat proyek ini dibutuhkan sesegera mungkin. Tahap ini dilaksanakan pada tanggal 14 November 2018 hingga 16 November 2018. Pada tahap ini tidak menggunakan QC karena penulis dan rekan, serta developer yang berada di Jakarta beserta dengan *user* melakukan pengecekan secara bersama – sama.

Tahap selanjutnya adalah tahap *Implementation*. Tahap ini dilakukan sesuai dengan dilakukannya *Family Games* ini yaitu pada tanggal 18 November 2018. Penulis dan rekan diminta untuk berjaga – jaga jika terjadi error yang harus segera ditangani.

Proyek selanjutnya yang penulis kerjakan adalah proyek web Pak RW (*Project Application Requisition Web*). Pada proyek kali ini, penulis hanya mengerjakannya sendiri. Pada proyek ini penulis melakukannya dengan tahapan yang sama seperti proyek – proyek sebelumnya.

Tahap *development* dikerjakan pada tanggal 3 Desember 2018 hingga 28 Desember 2018. Adapun tugas yang harus dikerjakan penulis pada tahap *development* ini antara lain *Login*, *Master User*, *Submit Requisition*, dan *Send email*,

Tugas *Login* merupakan tugas yang harus dibuat penulis untuk memasukkan fungsi untuk masuk ke dalam aplikasi. Setiap orang yang masuk ke dalam aplikasi ini akan berpengaruh terhadap *form requisition* yang akan dibuat nantinya.

Tugas *Master User* adalah tugas yang diberikan kepada penulis untuk mengelola data setiap *user* yang dapat mengakses aplikasi ini. Adapun *user* yang dimasukkan adalah *user* yang sudah terdaftar sebagai pegawai di ACC. *User* yang sudah tersimpan dapat diubah ataupun dihapus. Adapun menu ini hanya dapat diakses oleh *user* yang memiliki peran sebagai administrator.

Tugas *Submit Requisition* merupakan tugas kepada penulis untuk melakukan pengelolaan permintaan. Pada menu ini terdapat form yang harus diisi oleh *user* yang dibagi ke dalam 2 buah bagian, yaitu *form header* dan *form body*. *Form header* akan berisi data dari *user* yang sedang menggunakan aplikasi serta deskripsi alasan *user* membuat permintaan ini. *Form body* akan berisi detail dari permintaan *user*.

Tugas *Send Email* adalah tugas yang harus dikerjakan penulis untuk melakukan pengiriman email kepada orang yang dapat mengabulkan permintaan dari *user*. Fungsi ini akan ditampilkan saat *user* ingin mengirimkan semua permintaannya agar dapat disimpan ke dalam database pusat. Setiap data permintaan yang telah tersimpan nantinya diharapkan dapat selalu dipantau oleh sang pembuat.

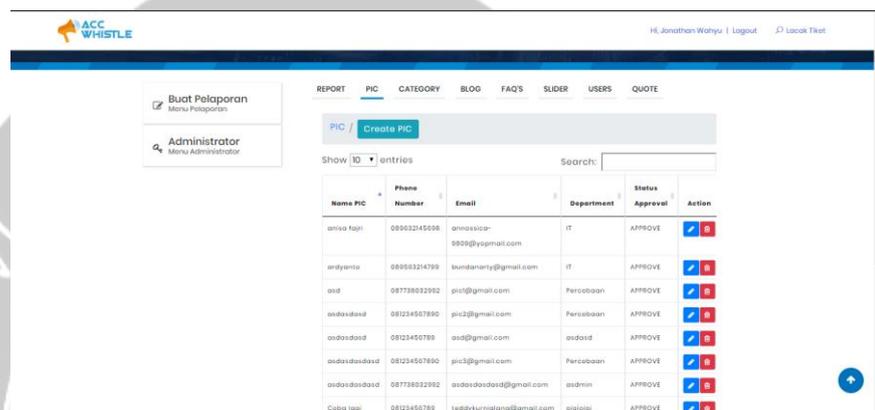
Setelah tahap *development* selesai, proyek memasuki tahap SIT. Pada tahap ini hanya berlangsung pada tanggal 29 Desember 2018 hingga 30 Desember 2018. Pada tahap ini *user* atau orang yang meminta agar aplikasi ini juga ikut melakukan *testing*. Tahap ini hanya dilakukan sebentar karena aplikasi ini akan dilanjutkan oleh *developer* lain. SIT dianggap telah selesai dan tidak ada banyak perubahan yang diinginkan oleh *user*.

2.3. Bukti Hasil Pekerjaan

Selama saya melakukan magang di Astra Credit Companies saya mengerjakan proyek SpeakUp. Berikut bukti hasil pekerjaan saya pada proyek tersebut :

1. Create Backend PIC

a. List PIC

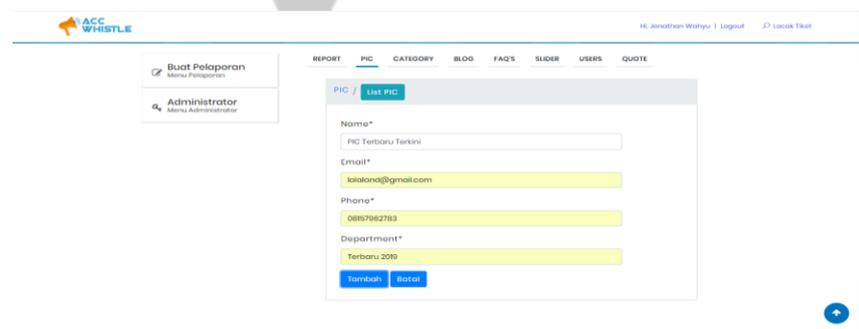


Nama PIC	Phone Number	Email	Department	Status Approval	Action
anisa hani	089032145008	anissaca-0809@yagmail.com	IT	APPROVE	+ -
andyanita	088563214789	sundanarty@gmail.com	IT	APPROVE	+ -
and	08778032992	pic1@gmail.com	Percobaan	APPROVE	+ -
andandand	08124507890	pic2@gmail.com	Percobaan	APPROVE	+ -
andandand	0812450789	and@gmail.com	andand	APPROVE	+ -
andandandand	08124507890	pic3@gmail.com	Percobaan	APPROVE	+ -
andandandand	08778032992	andandandand@gmail.com	admin	APPROVE	+ -
Coba lagi	0812450789	teddykurniatama@gmail.com	analisa	APPROVE	+ -

Gambar 2. 12 Halaman Tampil *List* PIC Speakup

Gambar diatas menampilkan halaman dari *list* PIC yang terdaftar dalam speakup. Halaman ini hanya dapat diakses oleh administrator ataupun authorizer. Melalui halaman ini pengguna dapat mengakses halaman untuk melakukan tambah, *edit* dan *delete* PIC sesuai dengan tombol yang ada

b. Create



ACC WHISTLE

Hi, Jonathan Wahyu | Logout | Lacak Tiket

Buat Pelaporan
Menu Pelaporan

Administrator
Menu Administrator

REPORT PIC CATEGORY BLOG FAQ'S SLIDER USERS QUOTE

PIC / Create PIC

Show 10 entries Search:

Nama*
PIC Terbaru Terkini

Email*
kaldana@gmail.com

Phone*
08857062783

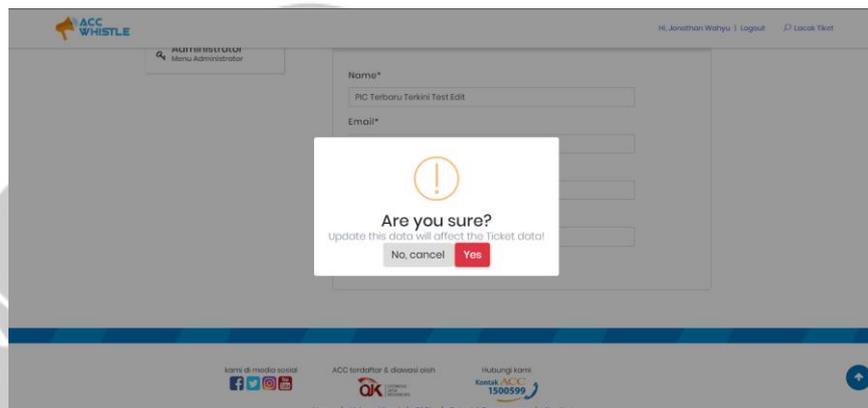
Department*
Terbaru 2019

Tambah Batal

Gambar 2. 13 Halaman *Create* PIC baru

Gambar diatas menampilkan halaman create PIC yang dapat diakses dengan memilih tombol *create* PIC pada halaman *list* PIC (re : Gambar 2.12). Adapun data yang harus dilengkapi adalah nama, email, nomor telepon, dan departemen tempat bekerja.

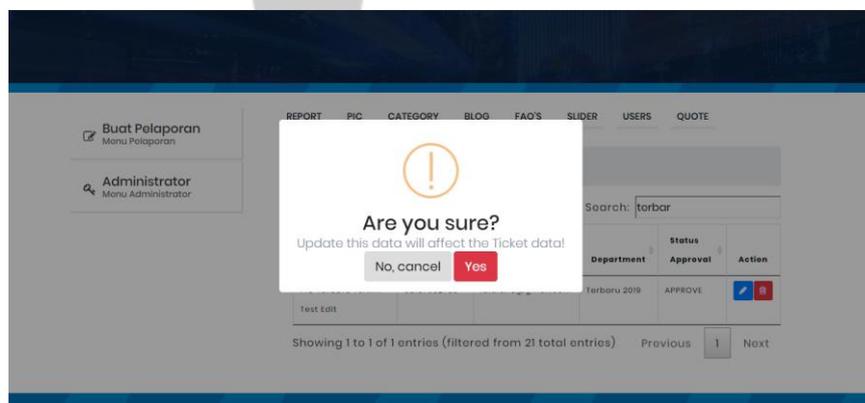
c. Edit



Gambar 2. 14 Halaman *Edit* PIC

Pada halaman sesuai gambar diatas, digunakan untuk melakukan perubahan data PIC. Halaman ini dapat diakses dengan menekan tombol edit yang berada di tiap PIC yang berada pada *List*. Sama seperti saat melakukan *create* PIC, data yang bisa diubah adalah nama, email, nomor telepon, dan juga departemen.

d. Delete

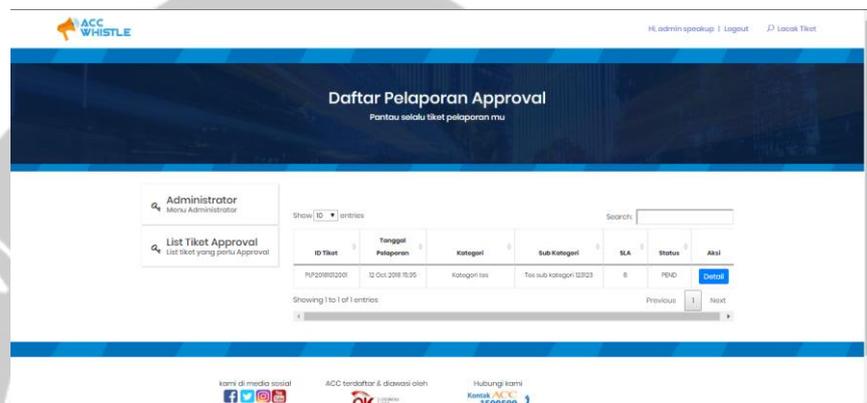


Gambar 2. 15 Halaman *Delete* PIC

Fungsi delete ini dapat diakses dengan menekan tombol *delete* yang ada disebelah kanan tombol *edit* data PIC. Setiap ingin melakukan penghapusan data PIC, aplikasi akan meminta konfirmasi dari user apakah ingin menghapus data tersebut atau tidak.

2. Table & Search List Pelaporan

a. List Pelaporan



Gambar 2. 16 Halaman List Pelaporan

Halaman pada gambar diatas merupakan halaman untuk menampilkan semua data pelaporan yang telah tersimpan di dalam aplikasi. Adapun halaman ini dapat diakses oleh pembuat laporan, PIC yang bertanggung jawab dengan kategori dari laporan tersebut, administrator, dan juga authorizer yang dapat melihat laporan mana yang membutuhkan *approval*.

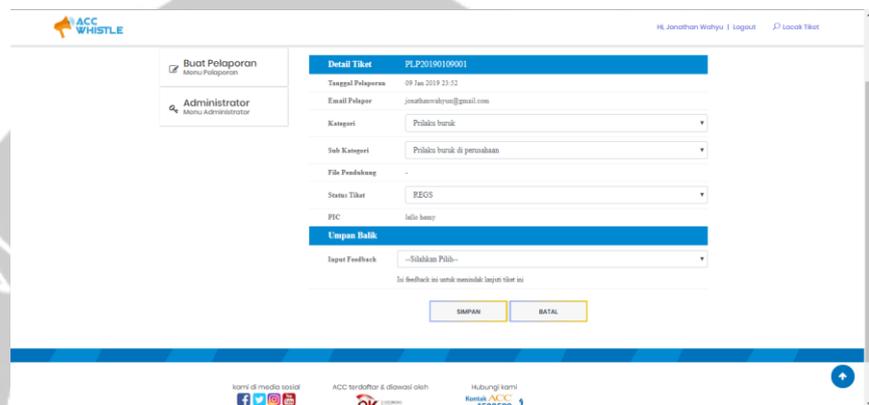
b. Search Pelaporan



Gambar 2. 17 Halaman home dengan fungsi *search*

Gambar diatas memperlihatkan fungsi *search* laporan yang dapat diakses di halaman mana saja. Untuk menggunakan fungsi ini, pengguna cukup menekan tulisan “Lacak Tiket” yang berada di *navigation bar*. Pengguna hanya dapat mencari laporan yang berkaitan dengan si pengguna.

c. Detail Pelaporan

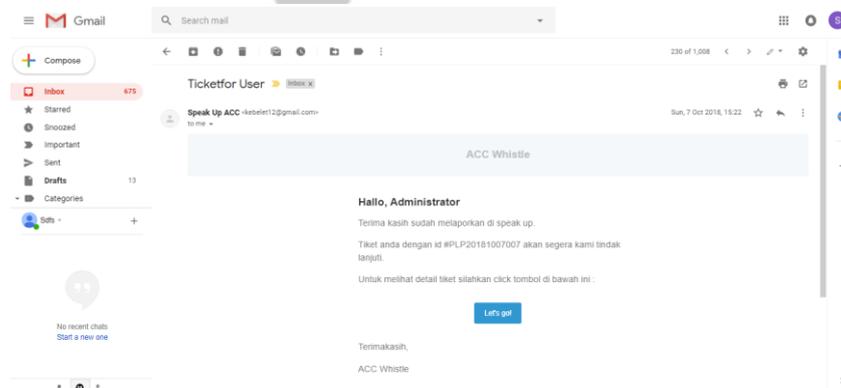


Gambar 2. 18 Halaman Detail Laporan

Pada gambar diatas, menampilkan halaman detail dari sebuah laporan. Untuk mengakses halaman ini pengguna dapat menekan tombol detail yang ada pada *list* laporan ataupun dapat langsung diakses setelah melakukan fungsi *search* dan menemukan hasilnya.

3. Push Email

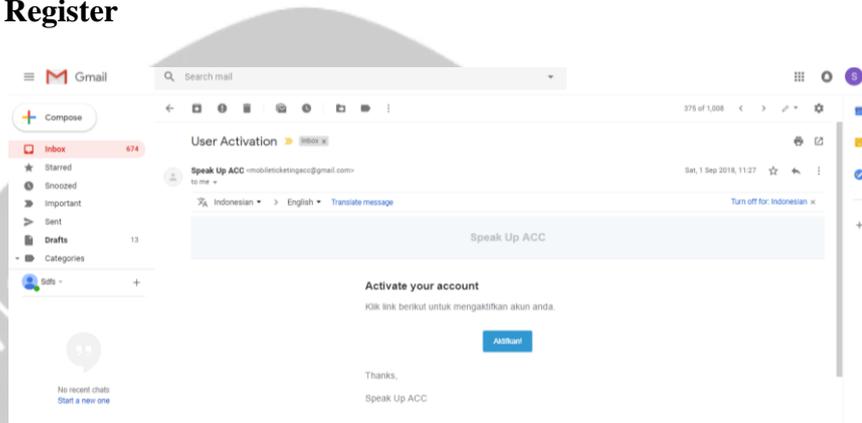
a. Create Pelaporan



Gambar 2. 19 Tampilan Email untuk Pembuat Laporan

Gambar diatas menampilkan email yang diterima setelah pengguna melakukan pelaporan menggunakan SpeakUp. Melalui tombol yang berada dalam email tersebut, pengguna akan diarahkan ke halaman detail dari pelaporan yang dibuatnya. Setiap pelaporan yang dibuat dapat dilihat oleh user melalui menu “Daftar Pelaporan”.

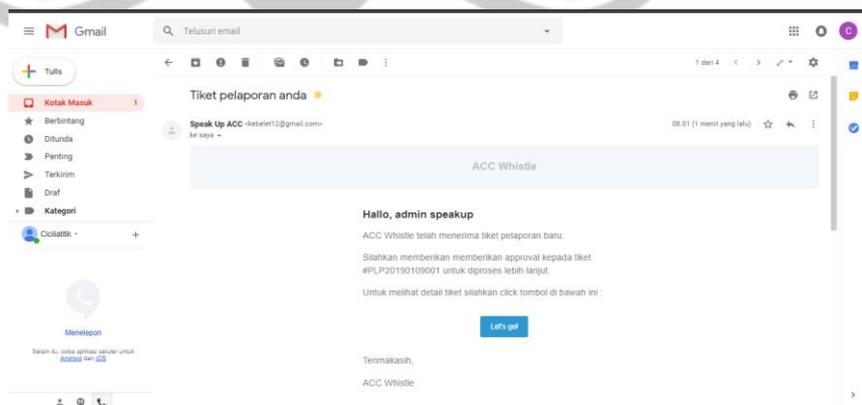
b. Register



Gambar 2. 20 Tampilan Email Aktifasi *User*

Email pada gambar diatas diberikan pada saat *user* telah berhasil melakukan registrasi. Setiap *user* yang ingin menggunakan aplikasi SpeakUp ini harus melakukan aktivasi terlebih dulu.

c. Approval Pelaporan

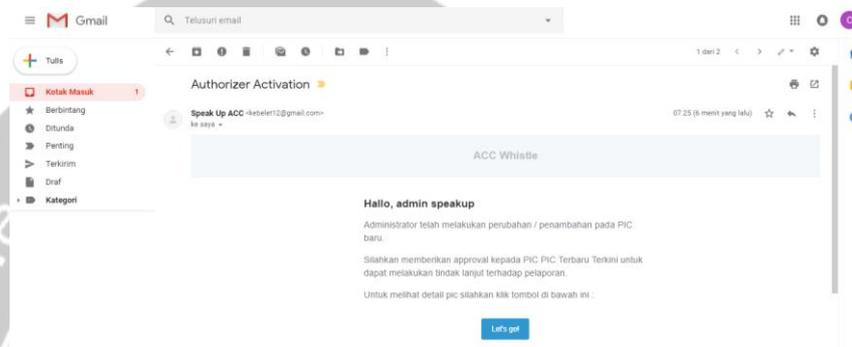


Gambar 2. 21 Tampilan Email *Approval* Pelaporan

Email pada gambar diatas diberikan kepada *Authorizer* untuk melakukan *approval* kepada setiap pelaporan yang masuk. Tombol

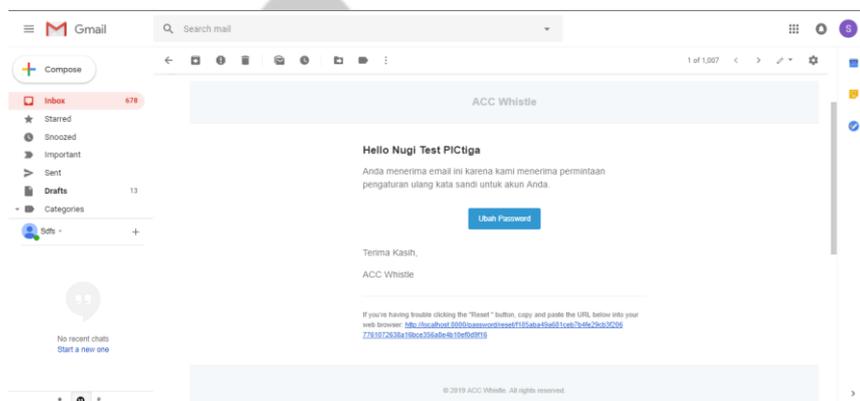
pada email itu akan mengarahkan pengguna kepada halaman detail dari laporan tersebut. Setiap laporan memiliki status masing – masing. Setiap status memiliki maksud tersendiri dan hanya beberapa pengguna saja yang dapat merubah status tersebut sesuai dengan alur bisnisnya.

d. Approval Authorizer (Super admin)



Gambar 2. 22 Tampilan Email *Approval* Penambahan PIC Pada gambar diatas ditampilkan email kepada *Authorizer* saat ada penambahan atau perubahan email dari PIC yang ada di SpeakUp yang dilakukan oleh administrator. Melalui email ini *Authorizer* dapat mengaktifkan PIC yang baru saja dibuat ataupun diubah.

e. Lupa Password

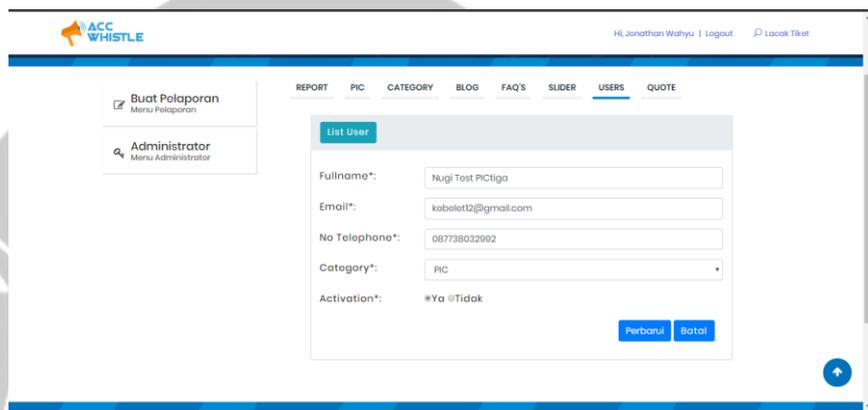


Gambar 2. 23 Tampilan Email Lupa Password

Email diatas didapatkan oleh setiap *user* yang menggunakan fungsi “Lupa Password”. Melalui email ini *user* akan diarahkan ke halaman ubah sandi dimana *user* diminta untuk memasukkan data email dan *password*-nya yang baru

4. Enhance Master Categories

a. Edit PIC Kategori tertentu



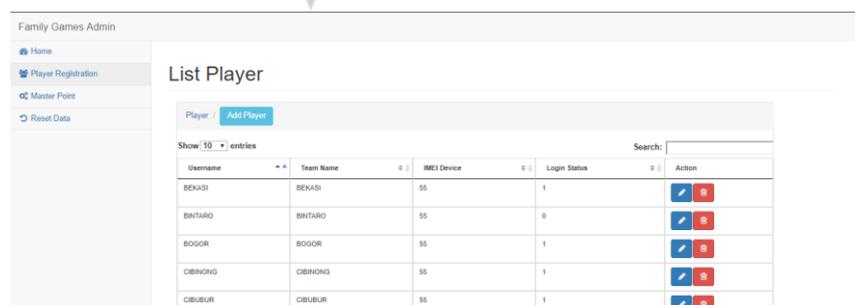
The screenshot shows a web application interface for ACC WHISTLE. The top navigation bar includes the logo and user information (Hi, Jonathan Wahyu | Logout) and a search icon (Lacak Tiket). The main menu includes REPORT, PIC, CATEGORY, BLOG, FAQ'S, SLIDER, USERS, and QUOTE. The 'USERS' menu is active, displaying a 'List User' form. The form contains the following fields: Fullname* (Nugi Test PICTiga), Email* (kabelet2@gmail.com), No Telephone* (087738032992), Category* (PIC), and Activation* (Ya/Tidak). There are 'Perbarui' and 'Batal' buttons at the bottom right of the form.

Gambar 2. 24 Halaman untuk Mengubah PIC

Halaman ubah PIC ini dilakukan pada menu *User*. Ini diperlukan karena data PIC berhubungan dengan data pada tabel *User*. Jika ada PIC berubah email dan data tidak ada pada tabel *User* maka akan terbentuk *User* baru.

5. Create BackEnd Player

a. List Player



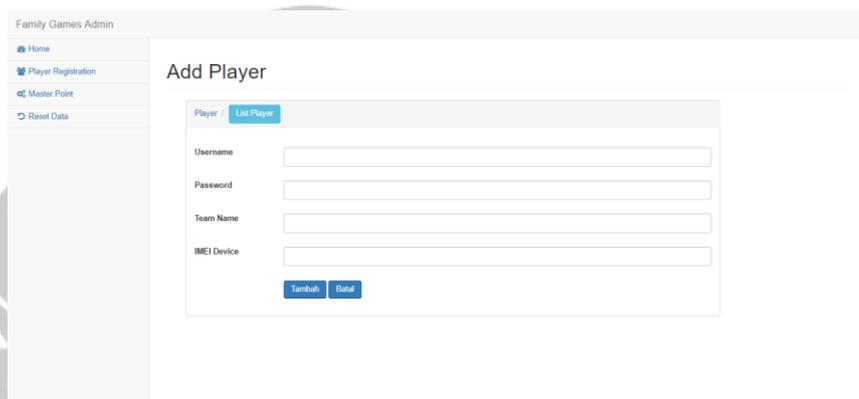
The screenshot shows the 'Family Games Admin' interface. The left sidebar contains navigation links: Home, Player Registration, Master Point, and Reset Data. The main content area is titled 'List Player' and features a table with the following data:

Username	Team Name	IMEI Device	Login Status	Action
BEKASI	BEKASI	55	1	[Edit] [Delete]
BINTARO	BINTARO	55	0	[Edit] [Delete]
BOGOR	BOGOR	55	1	[Edit] [Delete]
CIBINONG	CIBINONG	55	1	[Edit] [Delete]
CIBURUR	CIBURUR	55	1	[Edit] [Delete]

Gambar 2. 25 Halaman untuk melihat data semua *player*

Gambar diatas adalah tampilan halaman web yang berguna untuk memperlihatkan semua data *player* yang sudah disimpan. Setiap *player* yang sudah terdaftar pada sistem ini, dapat melakukan *login* menggunakan aplikasi mobile yang dibuat oleh rekan penulis.

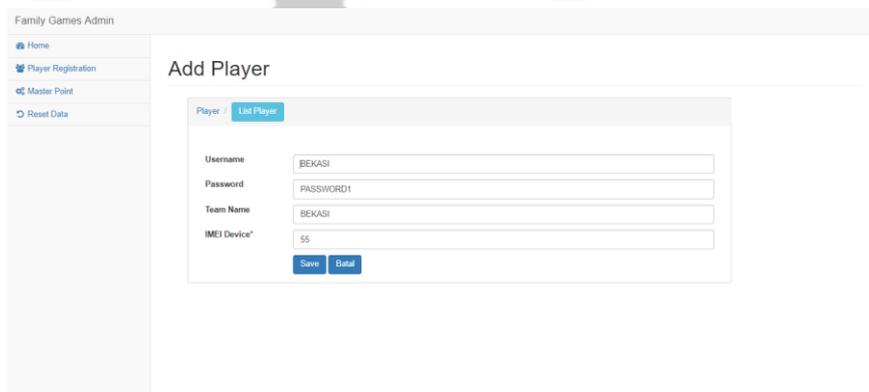
b. Create Player



The screenshot shows the 'Family Games Admin' interface. On the left is a sidebar menu with options: Home, Player Registration, Master Point, and Reset Data. The main content area is titled 'Add Player'. Below the title, there is a sub-header 'Player / List Player'. The form contains four input fields: 'Username', 'Password', 'Team Name', and 'IMEI Device'. At the bottom of the form are two buttons: 'Tambah' (Add) and 'Batal' (Cancel).

Gambar 2. 26 Halaman untuk menyimpan *player* baru
Di atas adalah gambar yang menunjukkan halaman pada saat ingin menyimpan data *player* baru. Setiap *player* di *Family Day Games* ini harus didaftarkan terlebih dulu melalui menu ini oleh panitia sebelum permainan dimulai.

c. Edit Player

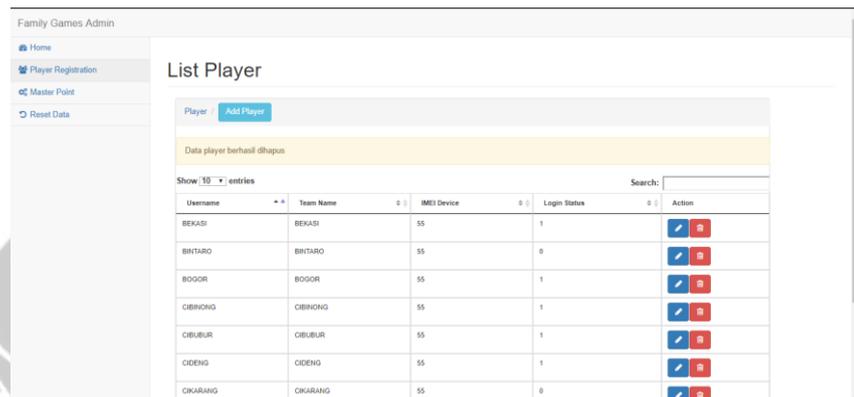


The screenshot shows the 'Family Games Admin' interface. On the left is a sidebar menu with options: Home, Player Registration, Master Point, and Reset Data. The main content area is titled 'Add Player'. Below the title, there is a sub-header 'Player / List Player'. The form contains four input fields with pre-filled data: 'Username' (BEKASI), 'Password' (PASSWORD1), 'Team Name' (BEKASI), and 'IMEI Device*' (55). At the bottom of the form are two buttons: 'Save' and 'Batal' (Cancel).

Gambar 2. 27 Halaman untuk mengubah data *player*

Gambar diatas adalah halaman pada saat panitia ingin mengubah data *player* yang sudah ada. Panitia hanya dapat mengubah data ini pada saat status *login* dari *player* tidak aktif.

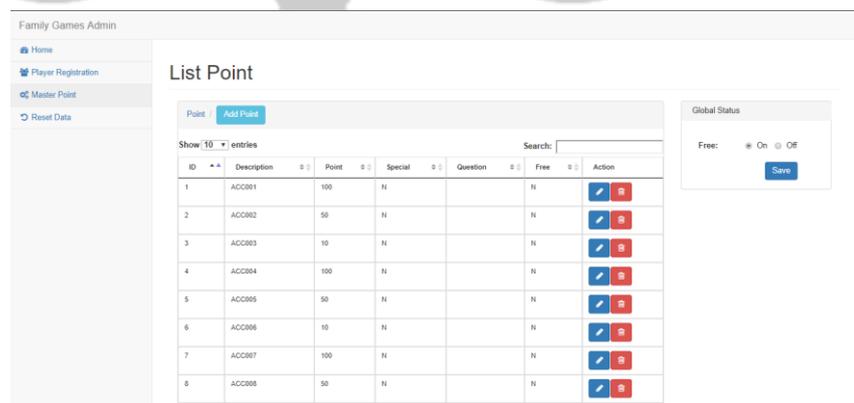
d. Delete Player



Gambar 2. 28 Halaman *list player* setelah berhasil menghapus data Gambar diatas adalah halaman yang menampilkan setelah sebuah data *player* berhasil dihapus. Karena penggunaan aplikasi ini dituntut untuk mudah dan cepat, disepakati untuk tidak memberikan *alert message*.

6. Create BackEnd Point

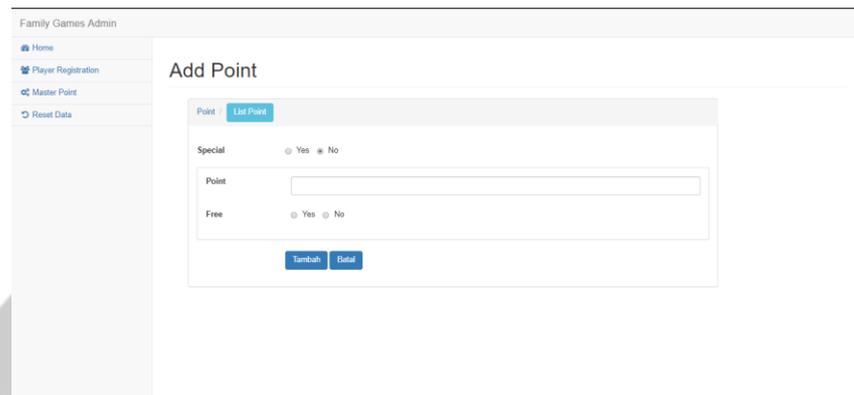
a. List Point



Gambar 2. 29 Halaman untuk menampilkan seluruh data *point* Gambar diatas digunakan untuk menampilkan seluruh data *point* yang telah tersimpan pada sistem Pak RW. Untuk menghindari

kecurangan, terdapat fungsi ubah Global Status yang menentukan apakah *point* dapat di scan atau tidak.

b. Create Point

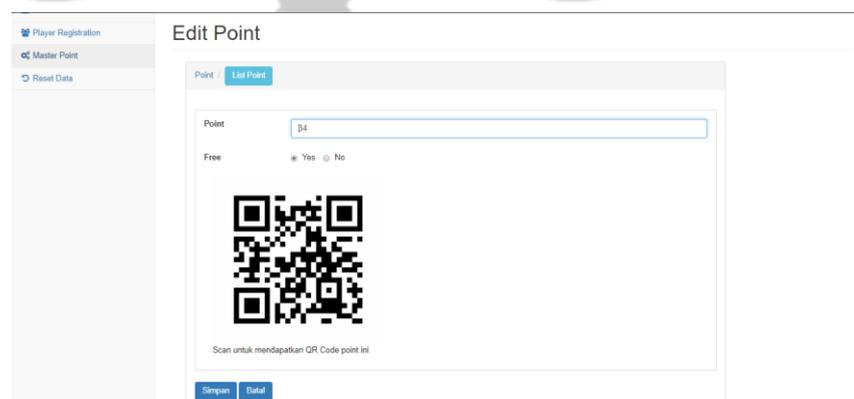


The screenshot shows the 'Add Point' form in the Family Games Admin interface. The form includes a sidebar with navigation options: Home, Player Registration, Master Point, and Reset Data. The main content area is titled 'Add Point' and contains a 'Point' section with a 'List Point' button. Below this, there are two radio button options: 'Special' (Yes/No) and 'Free' (Yes/No). A text input field for the point value is also present. At the bottom, there are 'Tambah' and 'Batal' buttons.

Gambar 2. 30 Halaman untuk menambah data *point*

Pada saat ingin menyimpan *point* baru, akan diminta pilihan apakah *point* yang akan dibuat spesial atau tidak. Spesial adalah *point* yang hanya berisi pertanyaan, sedangkan tidak spesial adalah *point* yang hanya berisi besar *point*. Setiap selesai melakukan penyimpanan data *point* baru maka sistem akan langsung membentuk sebuah *QR Code* yang berisi data dari *point* yang dibuat.

c. Edit Point

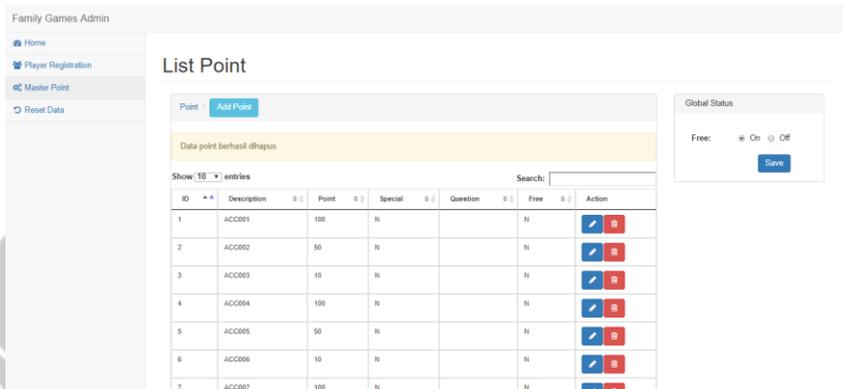


The screenshot shows the 'Edit Point' form in the Family Games Admin interface. The form includes a sidebar with navigation options: Player Registration, Master Point, and Reset Data. The main content area is titled 'Edit Point' and contains a 'Point' section with a 'List Point' button. Below this, there is a text input field for the point value (containing '14') and a 'Free' radio button option (Yes/No). A QR code is displayed below the input fields. At the bottom, there are 'Simpan' and 'Batal' buttons.

Gambar 2. 31 Halaman untuk mengubah data *point*

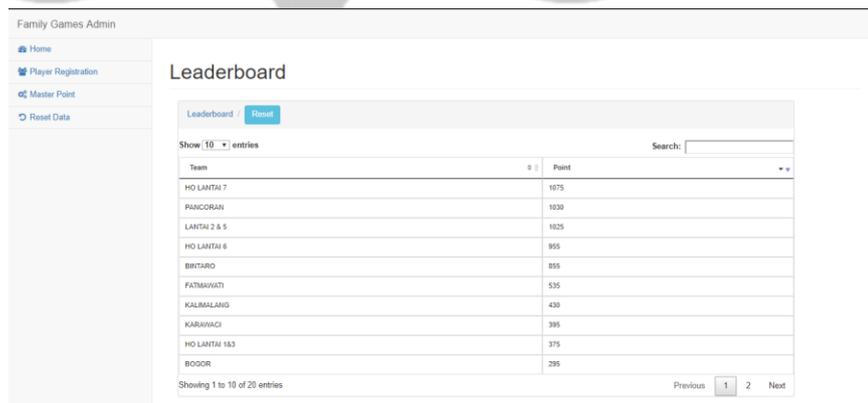
Gambar diatas adalah tampilan pada saat panitia ingin mengubah data dari *point*. Untuk mengakses fungsi ini panitia hanya perlu menekan tombol biru berlogo pensil pada *list point*.

d. Delete Point



Gambar 2. 32 Halaman *list point* setelah berhasil menghapus data Gambar diatas adalah tampilan pada saat panitia berhasil menghapus data *point*. Untuk mengakses fungsi ini, panitia cukup menekan tombol merah berlogo tong sampah. Tidak digunakannya *alert message* karena sistem ini dibutuhkan dapat bekerja secara cepat dan sederhana.

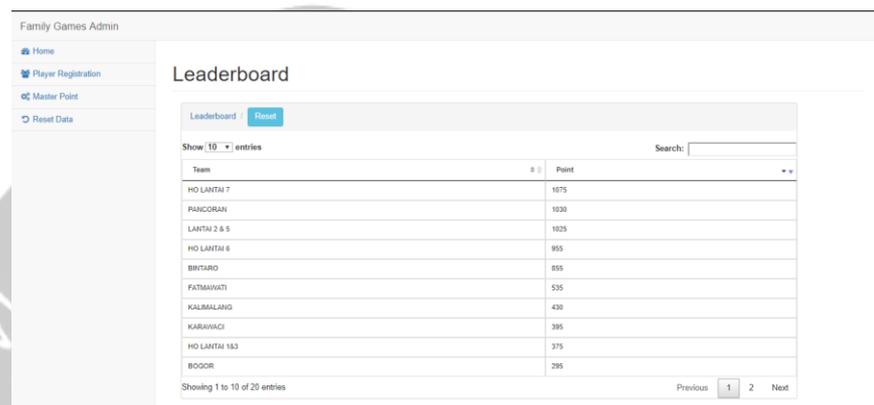
7. Leaderboard



Gambar 2. 33 Halaman untuk menampilkan leaderboard

Gambar diatas adalah tampilan untuk memperlihatkan perolehan *point* setiap *player* yang mengikuti acara. Setiap *player* berhasil melakukan *scan* data transaksi tersebut akan tersimpan ke dalam *database* dan ditampilkan di halaman ini.

8. Reset Data

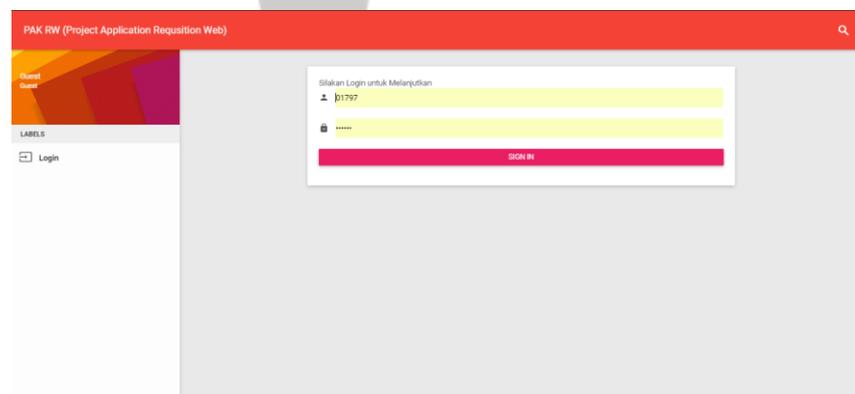


The screenshot shows the 'Family Games Admin' interface. On the left is a sidebar menu with options: Home, Player Registration, Master Point, and Reset Data. The main content area is titled 'Leaderboard' and features a 'Reset' button. Below the button is a table with columns for 'Team' and 'Point'. The table lists several teams and their corresponding point values. At the bottom of the table, it indicates 'Showing 1 to 10 of 20 entries' and includes pagination controls for 'Previous', '1', '2', and 'Next'.

Team	Point
HO LANTAI 7	1075
PANCORAN	1030
LANTAI 2 & 5	1025
HO LANTAI 6	955
BINTUJO	955
FATMAHATI	535
KALIMALANG	430
KARAWAJI	395
HO LANTAI 1&3	375
BIGGOR	285

Gambar 2. 34 Halaman untuk memanggil fungsi *reset* Tombol biru bertuliskan *Reset* pada halaman *leaderboard* berfungsi untuk menghilangkan semua data transaksi hasil *scan* yang disimpan di *database*. Fungsi ini akan membantu para panitia jika suatu saat akan melakukan permainan yang sama di waktu yang berbeda, sehingga permainan harus dimulai dari awal.

9. Login



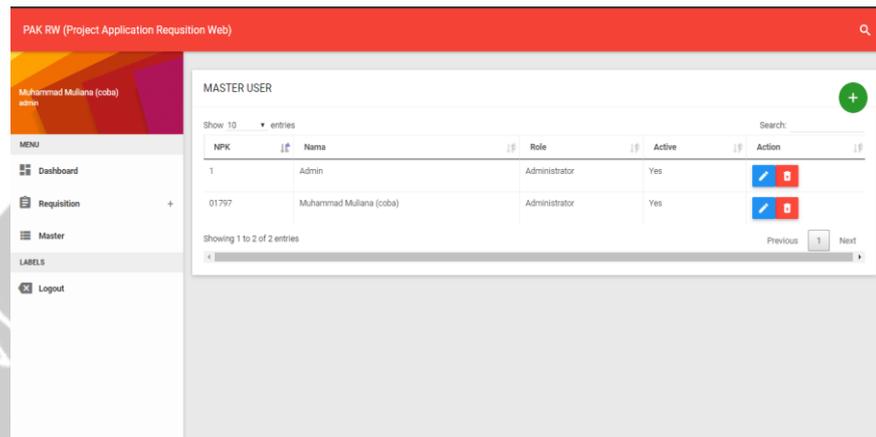
The screenshot shows the login page for 'PAK RW (Project Application Requestion Web)'. The page has a red header with the site name and a search icon. On the left is a sidebar with a 'Login' button. The main content area contains a login form with the text 'Silakan Login untuk Melanjutkan'. The form includes a username field with the value '01797', a password field with masked characters, and a red 'SIGN IN' button.

Gambar 2. 35 Halaman *Login*

Halaman diatas adalah halaman awal saat mengakses web Pak RW. Pada halaman ini *user* diminta untuk melakukan login terlebih dulu agar dapat mengakses menu web Pak RW.

10. Master User

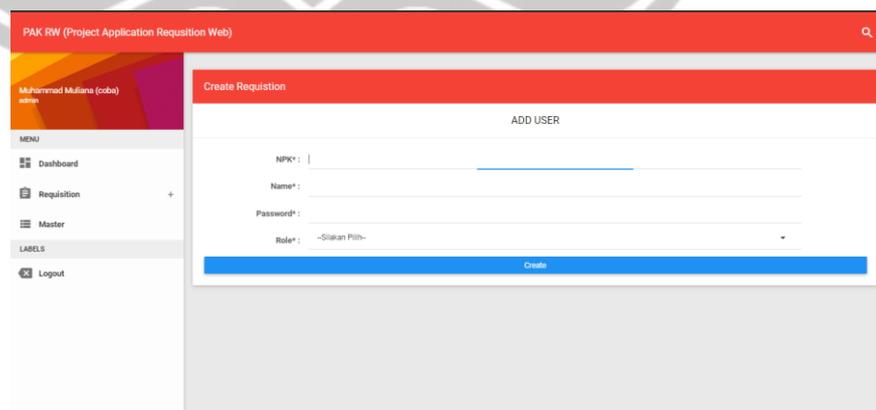
a. List User



Gambar 2. 36 Halaman untuk menampilkan list *user*

Gambar diatas menunjukkan halaman dari list *user* yang sudah terdaftar di sistem Pak RW. Halaman ini hanya dapat diakses oleh *user* yang memiliki *role* sebagai administrator.

b. Create User

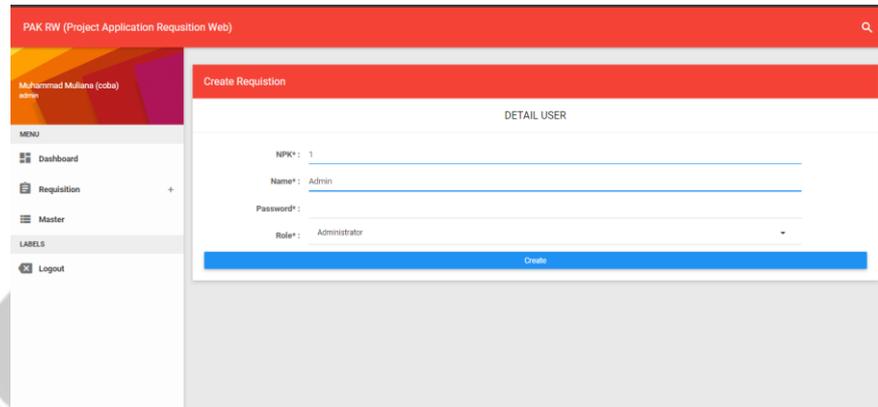


Gambar 2. 37 Halaman untuk menambah data *user*

Gambar diatas menampilkan halaman untuk menambah data *user*. *User* yang dapat ditambahkan merupakan *user* yang sudah terdaftar

sebagai pegawai di ACC. Untuk menu ini hanya dapat diakses oleh administrator

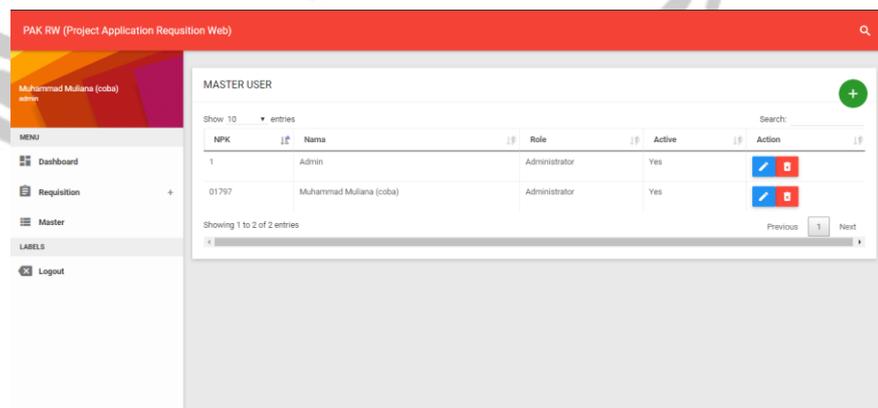
c. Edit User



Gambar 2. 38 Halaman untuk mengubah data *user*

Halaman ini juga hanya dapat diakses oleh administrator. Gunanya adalah untuk melakukan perubahan terhadap data *user* yang sudah ada.

d. Delete User



Gambar 2. 39 Halaman berisi tombol untuk hapus *user*

Pada gambar diatas dapat kita lihat tombol merah yang berada di list *user*. Tombol tersebut berfungsi untuk menghapus data *user* sesuai dengan baris yang dipilih.

11. Submit Requisition

a. List

The screenshot shows the 'Wishlist' page in the PAK RW system. The page title is 'PAK RW (Project Application Requisition Web)'. The user is logged in as 'Muhammad Muliana (coba) admin'. The page displays a table with one entry in the wishlist:

ID	COA	Item	UOM	Qty	Price	Condition	Total	Action
1	01.2102.36621.1812110001.0000.0000.00	RENOVATION-4 TH	Unit	3	75,000,000	Item	225,000,000	[Edit] [Delete]

Below the table, it shows 'Showing 1 to 1 of 1 entries' and pagination controls. A 'Confirm Request' button is located at the bottom of the page.

Gambar 2. 40 Halaman untuk tampil *wishlist*

Gambar diatas adalah tampilan dari menu *Add Requisition* pada bagian bawah, yang menampilkan *wishlist* yang sudah ada. *Wishlist* yang ditampilkan hanyalah *wishlist* yang dibuat oleh *user* yang sedang login. Menu ini dapat diakses oleh semua *user* yang sudah terdaftar di aplikasi web Pak RW.

b. Create

The screenshot shows the 'Create Requisition' form in the PAK RW system. The page title is 'PAK RW (Project Application Requisition Web)'. The user is logged in as 'Muhammad Muliana (coba) admin'. The form includes the following fields:

- Requester: MULIANA, MUHAMMAD
- Organization: ASTRA SEDAYA FINANCE, PT
- Department: Reg Retail Sales Head OKI
- Location: DIREKSTERNAL
- Description: coba
- Status: INCOMPLETE

Below the form, it displays a table with one entry:

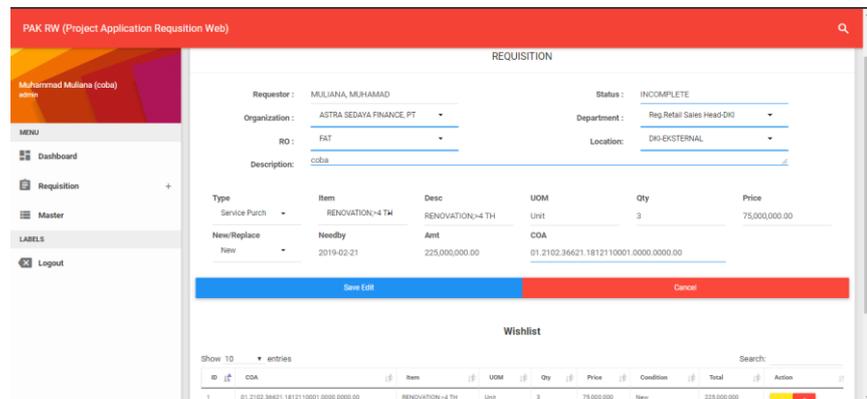
Type	Item	Desc	UOM	Qty	Price
Service Purch	REPAIR & MAINTENJ	REPAIR & MAINTENAN	Unit	10	5,000,000.00

The form includes a 'Add to Wishlist' button and a 'Cancel' button.

Gambar 2. 41 Halaman untuk *add requisition*

Gambar diatas menampilkan *form* untuk melakukan pembuatan permintaan. Setiap data yang dimasukkan nantinya akan ditampung terlebih dulu sebagai *wishlist* yang bisa dilihat pada *list* dibawahnya.

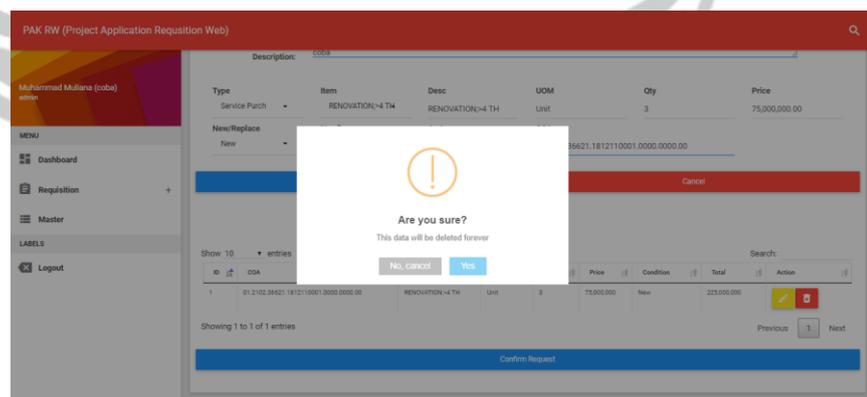
c. Edit



Gambar 2. 42 Halaman untuk mengubah data *wishlist*

Gambar diatas merupakan halaman yang sama, hanya memiliki fungsi yang berbeda yaitu untuk melakukan perubahan pada data *wishlist* yang dipilih. Untuk masuk kepada fungsi ini, *user* cukup menekan tombol edit berwarna kuning untuk mengubah data *wishlist* sesuai dengan barisnya masing – masing. Pada fungsi ini hanya *from detail* saja yang bisa diubah datanya.

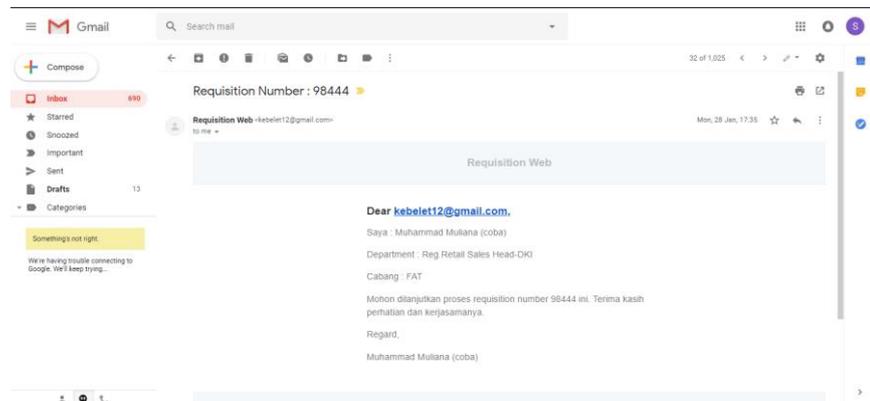
d. Delete



Gambar 2. 43 Halaman untuk hapus *wishlist*

Gambar diatas menampilkan *alert message* saat *user* ingin melakukan penghapusan data *wishlist*. Untuk mengakses fungsi ini, *user* cukup menekan tombol hapus berwarna merah pada *wishlist* yang ingin dihapus sesuai dengan barisnya.

12. Send Email



Gambar 2. 44 Tampilan email yang berhasil dikirimkan
Diatas adalah tampilan dari email yang dikirimkan setelah *wishlist* dirasa telah mencukupi. Untuk melakukan pengiriman ini, *user* cukup menekan tombol *confirm request* lalu mengisikan email tujuan.

BAB III

HASIL PEMBELAJARAN

3.1. Manfaat Magang

Pengalaman penulis ketika menjalani magang di Astra Credit Companies merupakan pengalaman yang sangat berharga. Magang ini menjadi sarana penulis untuk menambah ilmu, menambah relasi, dan menambah pandangan penulis terhadap perkembangan teknologi informasi yang ada pada dunia kerja terlebih pada Astra Credit Companies. Selama melaksanakan magang, penulis mengenal mengenai bisnis proses dari ACC dan teknologi – teknologi IT yang sedang diterapkan.

Pelaksanaan magang ini juga menambah pengetahuan penulis dalam melakukan *development* proyek berbasis web seperti proyek SpeakUp. Penulis dapat mengikuti proses dari awal pembuatan, melakukan *test* hingga pada akhirnya diimplementasi. Dalam proyek yang dikerjakan secara tim ini juga penulis dilatih untuk bisa berkomunikasi dengan baik dalam penyampaian perkembangan dan masalah yang dihadapi karena pengerjaan proyek dilakukan secara *remote* dengan tim di Jakarta.

Diharapkan juga dengan melakukan magang di Astra Credit Companies dapat membuat hubungan yang baik antara Atma Jaya Yogyakarta dengan Astra Credit Companies agar mempermudah kedua pihak untuk melakukan kerjasama.

3.2. Penerapan Ilmu dalam Magang.

Beberapa pengetahuan saat kuliah yang diterapkan pada pelaksanaan magang adalah sebagai berikut :

1. Pemrograman Web

Penggunaan ilmu dalam pemrograman web sangat membantu untuk memahami dasar – dasar bagaimana aplikasi berbasis web bekerja

2. Basis Data

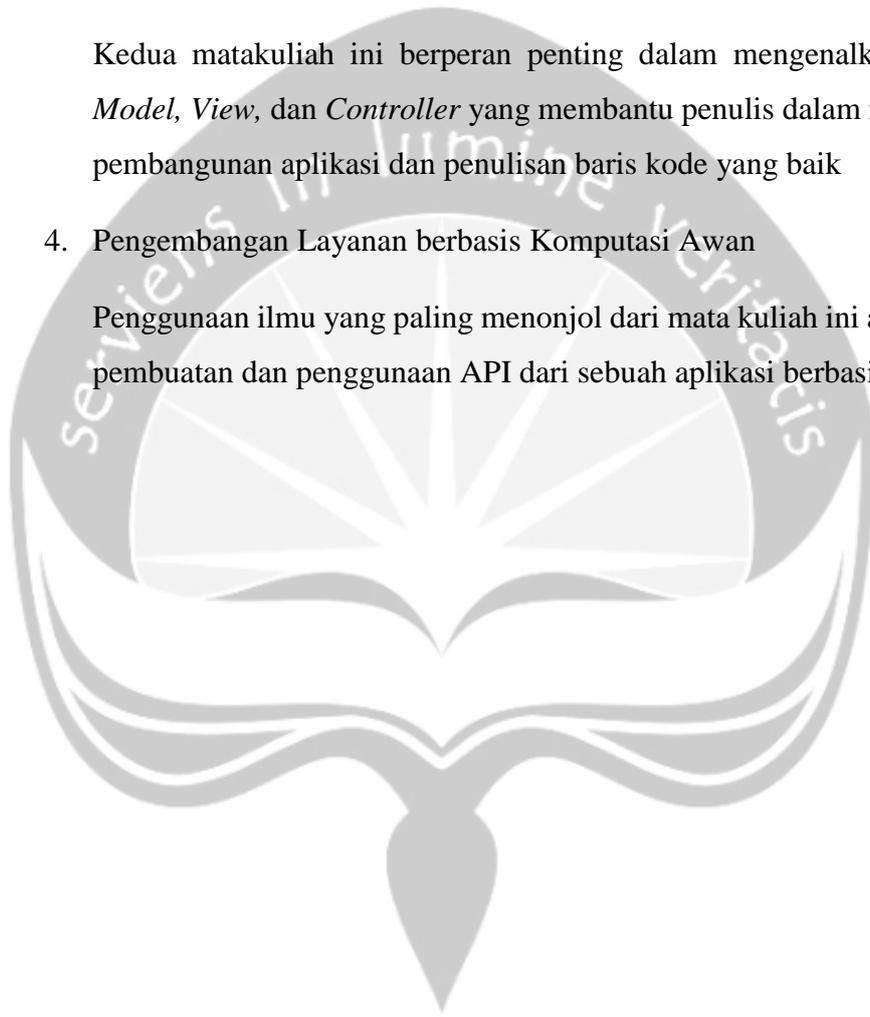
Pada proyek yang penulis kerjakan, basis data berperan penting untuk mengetahui bagaimana kerja *database* yang digunakan. Dalam pengerjaannya juga jenis basis data yang digunakan adalah SQL

3. Pemrograman Berbasis Objek dan Pemrograman Visual

Kedua matakuliah ini berperan penting dalam mengenalkan konsep *Model*, *View*, dan *Controller* yang membantu penulis dalam memahami pembangunan aplikasi dan penulisan baris kode yang baik

4. Pengembangan Layanan berbasis Komputasi Awan

Penggunaan ilmu yang paling menonjol dari mata kuliah ini adalah cara pembuatan dan penggunaan API dari sebuah aplikasi berbasis web.



BAB IV

HASIL PEMBELAJARAN

Dengan diadakannya mata kuliah magang ini, mahasiswa Teknik Informatika Universitas Atma Jaya Yogyakarta mampu mengenal lebih jauh mengenai dunia kerja yang berada di luar universitas. Penulis yang melaksanakan magang di Astra Credit Companies merasakan banyaknya ilmu yang diperoleh selama magang baik secara *hardskill* maupun *softskill* yang didapat secara langsung maupun tidak langsung.

Hardskill yang terasah adalah mengenai pembuatan program dengan *platform* yang belum pernah saya alami yaitu Laravel. Sedangkan *softskill* saya yang terasah berdasarkan keterlibatan saya dalam proyek Speakup, Family Games, dan Pak RW adalah belajar untuk bekerjasama dengan tim yang terdiri dari anggota yang belum dikenal, bukan hanya saling melengkapi pada saat *coding* namun juga melatih komunikasi yang baik untuk kelancaran sebuah proyek. Selain itu penulis juga merasakan tekanan yang besar untuk memberikan hasil sebaik mungkin dalam proyek – proyek yang diberikan serta merasa tertantang dengan cara kerja jarak jauh yang diterapkan pada magang kali ini.

Mata kuliah magang yang penulis lakukan sangatlah mendukung dalam proses belajar yang sedang dijalankan. Dengan adanya magang ini, mahasiswa dilatih agar dapat merasakan dunia kerja, menambah pengalaman kerja, dan menambah portofolio yang bisa digunakan mahasiswa sebagai bekal ketika lulus dari universitas.