

LAPORAN MAGANG
PT. ASTRA SEDAYA FINANCE



Dipersiapkan oleh:

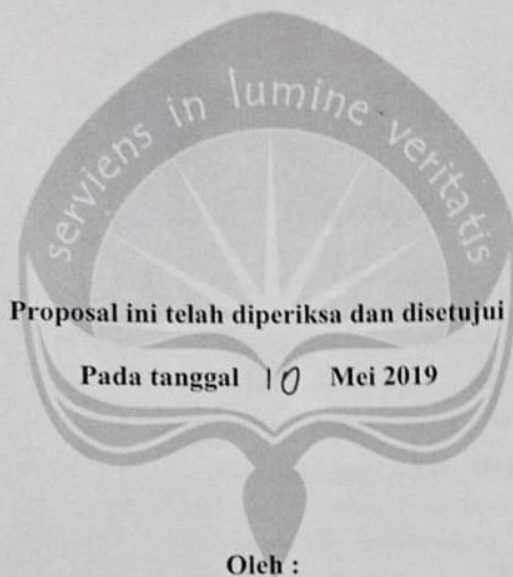
Kevin Antariksa

150708214

Program Studi Teknik Informatika
Fakultas Teknologi Industri
Universitas Atma Jaya Yogyakarta
2019

HALAMAN PENGESAHAN

Laporan Magang



Dosen Pembimbing

Joseph Eric S, S.Kom., MIT.

Pembimbing Lapangan

Julia Christie Feliciano D. C.

HALAMAN PENGESAHAN


Laporan Magang

Proposal ini telah diperiksa dan disetujui

Pada tanggal 10 Mei 2019

Oleh :

Dosen Pembimbing



Joseph Eric S, S.Kom., MIT.

Pembimbing Lapangan



Julia Christie Feliciano D. C.



No. 196/HC-RM / RMS / IV / 2019

Jakarta, 16 April 2019

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

Dengan hormat,

KATA PENGANTAR

Puji Syukur dan terima kasih penulis panjatkan ke hadirat Tuhan Yang Maha Esa atas segala rahmat dan karunia-Nya, sehingga penulis dapat menyelesaikan laporan magang ini. Penyusunan laporan magang ini tidak lepas dari bantuan berbagai pihak yang senantiasa mendukung penulis. Oleh sebab itu penulis memberikan ucapan terima kasih kepada :

1. Tuhan Yang Maha Esa, yang selalu menyertai dan melimpahkan rahmat dan karunia-Nya kepada penulis.
2. Orang tua, yang selalu mendukung penulis
3. Bapak Joseph Eric Samodra, S.Kom., MIT. selaku dosen pembimbing magang yang selama ini telah membimbing penulis selama magang dan penulisan laporan ini.
4. Ibu Awa dan Ibu Christie, selaku pembimbing lapangan yang telah membimbing penulis selama pelaksanaan magang.
5. Teman-teman PG (Wahyu, Dika, Bobby, Daniel, Komang, Ian, Verry, Travis) yang telah memberikan semangat kepada penulis untuk melaksanakan dan menyelesaikan magang.
6. Teman-teman kos TB V 5A yang selalu mendukung penulis..
7. Seluruh teman-teman dan saudara yang telah memberikan bantuan serta dukungan pada penulis.

Penyusunan laporan ini bertujuan untuk memenuhi persyaratan dalam menempuh mata kuliah magang. Laporan ini berisi dokumentasi pengetahuan, pengalaman, hingga hambatan yang penulis dapatkan selama melaksanakan magang. Magang dilaksanakan penulis pada tanggal 03 Januari 2019 - 07 April 2019 di PT. Astra Sedaya Finance (ACC – Astra Credit Company).

Penulis menyadari bahwa pelaksanaan magang dan penyusunan laporan ini masih jauh dari sempurna. Oleh karena itu, kritik dan saran yang membangun sangat penulis harapkan. Penulis berharap laporan ini akan memberikan manfaat kepada pembaca.

Yogyakarta, 16 April 2019

Penulis,
Kevin Antariksa



DAFTAR ISI

LAPORAN MAGANG	1
HALAMAN PENGESAHAN	2
KATA PENGANTAR	3
DAFTAR ISI	5
DAFTAR GAMBAR	6
BAB I PENDAHULUAN	9
1.1 Sekilas Perusahaan	9
1.2 Sejarah Perusahaan	10
1.3 Visi dan Misi Perusahaan	12
1.4 Struktur Organisasi	12
1.5 Deskripsi Tugas Struktur Organisasi Perusahaan	13
1.6 Departemen Teknologi Informasi(TI) dalam perusahaan	18
BAB II PELAKSANAAN MAGANG	19
2.1 Penjelasan Logbook	19
2.2 Hasil Pekerjaan Secara Umum	50
2.3 Bukti Hasil Pekerjaan	52
BAB III HASIL PEMBELAJARAN	74
3.1 Manfaat Magang	74
3.2 Penerapan Ilmu dalam Magang	75
BAB IV KESIMPULAN	76

DAFTAR GAMBAR

Gambar 1.1 Logo perusahaan Astra Sedaya Finance (ACC)	9
Gambar 1.2 Bagan struktur organisasi ACC	12
Gambar 2.1 Aplikasi OutSystem	22
Gambar 2.2 Aplikasi uniPaaS	23
Gambar 2.3 Aplikasi Software AG Designer.....	24
Gambar 2.4 Framework Laravel.....	25
Gambar 2.5 FortiClient VPN.....	27
Gambar 2.6 PL/SQL yang dibuat.....	28
Gambar 2.7 ACC BidMart.....	29
Gambar 2.8 Contoh API sederhana di OutSystem.....	30
Gambar 2.9 Contoh hasil GET API.....	32
Gambar 2.10 Contoh penggunaan Guzzle.....	33
Gambar 2.11 Tampilan yang dibuat	34
Gambar 2.12 Halaman tutorial CSS w3schools.....	39
Gambar 2.13 Salah satu dokumentasi API.....	52
Gambar 2.14 Salah satu dokumentasi API POST.....	52
Gambar 2.15 Mencoba POST API di Postman.....	53
Gambar 2.16 Hasil GET berdasarkan ID tertentu.....	53
Gambar 2.17 Hasil Delete berdasarkan ID tertentu	54
Gambar 2.18 Tampilan view data Balai Lelang + pagination	54

Gambar 2.19 Tampilan create data Balai Lelang	55
Gambar 2.20 Tampilan update data Balai Lelang	55
Gambar 2.21 Tampilan search data Balai Lelang	55
Gambar 2.22 Dokumentasi API Unit	56
Gambar 2.23 Hasil GET data Unit	56
Gambar 2.24 Hasil POST Unit	57
Gambar 2.25 Hasil GET berdasarkan ID tertentu	57
Gambar 2.26 Hasil DELETE UNIT	58
Gambar 2.27 Tampilan view Unit	58
Gambar 2.28 Upload file .xlsx	58
Gambar 2.29 Tampilan Create data Unit	59
Gambar 2.30 Tampilan Update data Unit	59
Gambar 2.31 Tampilan search data Unit	60
Gambar 2.32 Dokumentasi API Auction Event	61
Gambar 2.33 Hasil GET data Auction Event	61
Gambar 2.34 Hasil Generate No. LOT	62
Gambar 2.35 Hasil POST Auction Event	62
Gambar 2.36 Hasil GET berdasarkan ID tertentu	63
Gambar 2.37 Tampilan view Auction Event	63
Gambar 2.38 Tampilan create Auction Event	64
Gambar 2.39 Tampilan search data Auction Event	64

Gambar 2.40 Dokumentasi API Top Up	65
Gambar 2.41 Hasil GET data Top Up unverified	65
Gambar 2.42 Hasil GET data Top Up verified	66
Gambar 2.43 Hasil verify Top Up	66
Gambar 2.44 Hasil GET berdasarkan ID tertentu	67
Gambar 2.45 Tampilan view Top Up	67
Gambar 2.46 Tampilan verify Top Up	68
Gambar 2.47 Tampilan search data Top Up	68
Gambar 2.48 Dokumentasi API Penarikan.....	69
Gambar 2.49 Hasil GET data Penarikan unverified	69
Gambar 2.50 Hasil GET data Penarikan verified	70
Gambar 2.51 Hasil verify Penarikan	70
Gambar 2.52 Hasil GET berdasarkan ID tertentu	71
Gambar 2.53 Tampilan view Penarikan	71
Gambar 2.54 Tampilan verify Penarikan	72
Gambar 2.55 Tampilan search Penarikan	72

BAB I PENDAHULUAN

1.1 Sekilas Astra Credit Company

Astra Credit Companies adalah grup perusahaan pembiayaan yang telah berdiri sejak tahun 1982. Pada tahun 1994, PT Astra Sedaya Finance bersama PT Swadharma Bhakti Sedaya Finance (SBSF), PT Astra Auto Finance (AAF), PT Staco Estika Sedaya Finance (SESF) dan PT Pratama Sedaya Finance (PSF) bergabung dalam kelompok pembiayaan dengan brand Astra Credit Companies (ACC). ACC memiliki beragam produk pembiayaan, mulai dari pembiayaan mobil baru untuk produk Astra dan non Astra, pembiayaan mobil bekas melalui dealer maupun perorangan dan juga melayani pembiayaan Retail Commercial dan Fleet.



Gambar 1.1. Logo perusahaan Astra Credit Company

ACC memiliki jaringan yang luas meliputi 73 kantor cabang dan kantor pelayanan. Dengan tersedianya lebih dari 76.000 titik pembayaran di seluruh Indonesia akan membuat pelanggan ACC lebih mudah melakukan pembiayaan di ACC. ACC sebagai perusahaan multifinance yang solid dan telah menerima pengakuan berupa penghargaan dari berbagai pihak secara kontinyu. Di tahun 2015 ini ACC telah menerima penghargaan Top Brand Award, Contact Center Service Excellence Award, Service Quality Award dan Corporate Image Award. ACC juga mendapatkan penghargaan atas

partisipasi dalam Operasionalisasi mobil SiMOLEK pada program Literasi Keuangan Otoritas Jasa Keuangan.

1.2 Sejarah Singkat Astra Credit Company

Astra Credit Companies atau biasa di singkat 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. 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.

ACC 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 idAAA 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 dan Misi Astra Credit Company

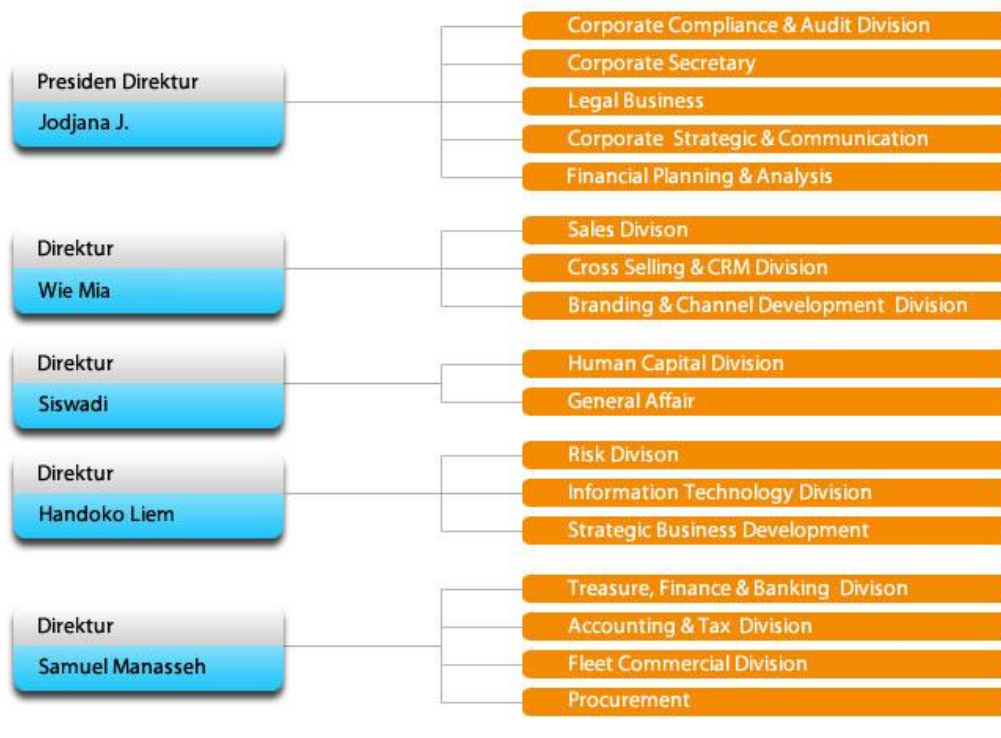
Visi

Become the 1st Choice Financing Company with Total Solution.

Misi

To Promote Credit for a better living.

1.4 Struktur Organisasi Astra Credit Company



Gambar 1.2. Struktur Organisasi Perusahaan Astra Credit Company

1.5 Deskripsi Tugas dari Struktur Organisasi Astra Credit

Company

1. Direksi

- a. Menyusun visi, misi dan nilai-nilai serta rencana strategis Perseroan dalam bentuk rencana korporasi (corporate plan) dan rencana bisnis (business plan)
- b. Menetapkan struktur organisasi Perseroan lengkap dengan rincian tugas setiap divisi dan unit usaha
- c. Mengelola sumber daya yang dimiliki Perseroan secara efektif dan efisien
- d. Membentuk sistem pengendalian internal Perseroan dan manajemen risiko
- e. Memperhatikan kepentingan yang wajar dari pemangku kepentingan Perseroan.

2. Corporate Audit & Compliance

- a. Meninjau kredibilitas dan obyektivitas laporan keuangan
- b. Meninjau tindakan yang telah diambil untuk mengidentifikasi dan mengendalikan risiko bisnis dan keuangan.
- c. Meninjau tindakan yang telah diambil untuk memastikan efektivitas pengendalian internal.
- d. Meninjau rencana dan pelaksanaan kegiatan fungsi Auditor Internal, Manajemen Risiko, dan Auditor Eksternal untuk menilai apakah risiko-risiko kunci telah dengan tepat dievaluasi dan dikendalikan.
- e. Meninjau obyektivitas dan independensi Auditor Internal dan Eksternal.
- f. Meninjau kecukupan monitoring ketaatan terhadap aturan dan etika bisnis.

3. Corporate Secretary

- a. Mengikuti perkembangan Pasar Modal khususnya peraturan-peraturan yang berlaku di Pasar Modal.

- b. Memberikan pelayanan informasi kepada masyarakat yang berkaitan dengan kondisi Perseroan.
- c. Memberikan masukan kepada Direksi dalam rangka mematuhi ketentuan dan peraturan di Pasar Modal.
- d. Menjadi penghubung antara Perseroan dengan otoritas dan regulator Pasar Modal serta penghubung antara Perseroan dengan masyarakat.

4. Legal Business

- a. menangani dokumen dan perizinan
- b. menangani permasalahan hukum, baik untuk masalah perdata maupun pidana
- c. melakukan penyesuaian-penyesuaian terhadap peraturan baru yang dikeluarkan oleh pemerintah yang berkaitan dengan operasional perusahaan

5. Corporate Strategic & Communication

- a. Untuk menyempurnakan profil dari perusahaan (*corporate branding*);
- b. Untuk meminimalkan perbedaan antara identitas perusahaan yang diinginkan dan fitur merek dari sebuah produk yang dihasilkan perusahaan;
- c. Untuk mendelegasikan tugas-tugas dalam komunikasi;
- d. Untuk merumuskan dan melaksanakan prosedur yang efektif untuk membuat keputusan mengenai urusan komunikasi;
- e. Untuk menggerakkan dukungan internal dan eksternal demi tujuan-tujuan perusahaan;
- f. Untuk berkoordinasi dengan perusahaan-perusahaan bisnis internasional.

6. Financial Planning & Analysis

- a. Mengatur dan mengelola keuangan secara tepat
- b. Memantau rencana keuangan

- c. Memberikan solusi terhadap masalah keuangan yang dihadapi

7. Sales Division

- a. Aktif mencari target
- b. Merekap data hasil penjualan
- c. Menjamin kepuasan pelanggan
- d. Mencari mitra kerja
- e. Menyusun strategi lanjutan

8. Cross Selling & CRM Division

- a. Meningkatkan hubungan antara perusahaan dengan pelanggan yang sudah ada untuk meningkatkan pendapatan perusahaan.
- b. Menyediakan informasi yang lengkap mengenai pelanggan untuk memaksimalkan jalinan hubungan pelanggan dengan perusahaan melalui penjualan secara *up selling* dan *cross selling*.
- c. Menggunakan informasi yang terintegrasi untuk menghasilkan pelayanan yang memuaskan dengan memanfaatkan informasi pelanggan untuk memenuhi kebutuhan pelanggan.
- d. Menghasilkan konsistensi dalam prosedur dan proses menyalurkan jawaban kepada pelanggan.

9. Branding & Channel Development Division

- a.

10. Human Capital Division

- a. Memimpin dan melaksanakan proses recruitment.
- b. Mengelola pelatihan & pengembangan.

11. General Affair

- a. Melakukan pengurusan seluruh perijinan yg dibutuhkan oleh perusahaan, menjaga hubungan baik dengan lingkungan

sekitar perusahaan dan dengan pemerintah daerah setempat, melakukan pencatatan dan pelaporan Asset Perusahaan, melakukan pemeliharaan dan perbaikan sarana kantor, dan memastikan ketersediaan kebutuhan kantor, serta pengamanan Fasilitas kantor dan asset perusahaan

12. Risk Division

- a. Memantau dan koreksi tindakan penyimpangan
- b. Mengawasi manajemen dan kultur pengendalian
- c. Identifikasi penilaian risiko

13. Information Technology Division

- a. Menilai inovasi teknologi baru dengan melakukan perbandingan (*benchmarking*), ujicoba dan analisis untuk mendapatkan teknologi yang sesuai dengan proses bisnis perusahaan
- b. Menganalisis sistem dan teknologi yang ada di perusahaan dengan cara *outsourcing* agar dapat mengetahui sistem dan teknologi yang aplikatif
- c. Membuat usulan dan saran-saran penyesuaian program dan strategi dibidang Sistem dan Teknologi Informasi yang dipandang perlu dengan cara menyampaikan hasil evaluasi dalam rangka pencapaian tujuan jangka panjang
- d. Merencanakan manajemen sistem informasi dengan membuat strategi, metodologi sistem informasi agar dapat diaplikasikan
- e. Mengelola penerapan teknologi baru menggunakan prinsip-prinsip manajemen perubahan dengan cara sosialisasi yang terencana untuk meminimalkan resistensi pengguna

14. Strategic Business Development

- a. Menciptakan ide
- b. Melakukan seleksi ide
- c. Mengembangkan konsep
- d. Melakukan sample atau uji coba

- e. Melakukan evaluasi program
- f. Mempelajari strategi pesaing
- g. Mengawasi program yang telah berjalan

15. Treasure Finance & Banking Division

- a. Mengambil sebagian atau sebuah data yang diberikan oleh bagian akunting ke dalam sebuah sistem secara komputerisasi untuk dilakukan *cash forecasting*.
- b. Mengelola modal kerja.
- c. Mengelola kas agar dana terjamin mencukupi kebutuhan kegiatan operasional sebuah perusahaan

16. Accounting & Tax Division

- a. Mencatat, memeriksa, dan melaporkan semua transaksi yang berkaitan dengan keuangan kantor.
- b. Menyusun rencana perpajakan untuk optimalisasi pajak
- c. Melakukan koordinasi dengan perusahaan afiliasi dan bagian terkait dalam melaksanakan hak dan kewajiban perpajakan
- d. Approval laporan pajak masa dan tahunan secara akurat dan tepat waktu
- e. Melakukan verifikasi transaksi perusahaan yang terkait aspek pajak
- f. Menangani audit pajak dan menyusun budget tahunan bagian pajak
- g. Update peraturan perpajakan untuk memastikan tax compliance

17. Fleet Commercial Division

- a. menghilangkan atau meminimalkan risiko yang terkait dengan investasi kendaraan,
- b. meningkatkan efisiensi, produktivitas
- c. mengurangi transportasi secara keseluruhan dan staf biaya

18. Procurement

- a. Perencanaan pembelian
- b. membuat prosedur standar pengadaan barang/jasa
- c. membuat spesifikasi barang/jasa yang dibutuhkan secara detail dengan informasi yang didapat dari departemen/bagian lain yang memintanya
- d. pencarian supplier/vendor yang tepat dengan melihat penawaran serta rekam jeaknya secara detail
- e. membuat perbandingan biaya pembelian dari supplier/vendor
- f. negosiasi harga, jangka waktu pembayaran (term of payment), pengiriman (shipping), dll.
- g. memutuskan pembelian dari supplier/vendor
- h. membuat kontrak
- i. melakukan kontrol jumlah persediaan di gudang
- j. menerima tagihan pembayaran dari vendor/supplier

1.6 Departemen IT dalam Perusahaan

Departemen IT atau dalam Astra Credit Company menyebutnya Information Technology Division. Divisi ini merupakan bagian yang berhubungan dengan semua yang berkaitan dengan IT, seperti membangun sistem baru yang digunakan dalam perusahaan tersebut, selain itu juga menganalisis sistem yang ada dan membuat usulan serta saran-saran penyesuaian program yang kemudian digunakan untuk menyempurnakan sistem yang ada. Selain itu, tugas lainnya adalah menilai inovasi teknologi terbaru dengan melakukan perbandingan, uji coba dan analisis untuk mendapatkan teknologi yang sesuai dengan proses bisnis perusahaan

BAB II

PELAKSANAAN MAGANG

2.1 Penjelasan Logbook

Kamis, 3 Januari 2019

Hari ini, saya dan 4 rekan saya (Elen, Valen, Desya, dan Budi) selaku magang batch 2 berkenalan dengan pembimbing lapangan kami, yaitu Ibu Awa. Ibu Awa disini adalah BSA (Business System Analyst) dari ACC Jakarta yang akan membimbing kami hingga Februari 2019. Pada hari pertama ini kami diperkenalkan secara singkat mengenai perusahaan ACC. Pada minggu pertama kami mendapatkan training terlebih dahulu. Traing pertama adalah Training arsitektur. Training ini dibawakan oleh mas Amri. Setelah itu ada training mengenai STLC yang dibawakan oleh Ibu Mifta STLC merupakan singkatan Software Testing Life Cycle. Pelatihan alur testing pada perangkat lunak penting agar developer dapat mengetahui proses testing yang harus dilewati.

Pelatihan berikutnya adalah Testing Methodology, yang dibawakan oleh Tulus Suyato. Pelatihan ini melanjutkan pembahasan sebelumnya, yaitu mengenai software testing. Pada pelatihan kali ini juga dijelaskan macam-macam testing. Pelatihan ketiga pada hari ini adalah pelatihan SDLC, Scrum, dan Waterfall yang dibawakan oleh Bapak Martin. SDLC merupakan singkatan Software Development Life Cycle. Pada pelatihan kali ini dijelaskan metodologi SDLC yang sering digunakan ACC, yaitu Scrum dan Waterfall. Dijelaskan masing-masing dari kedua metode tersebut. Yang terakhir pada hari ini adalah persiapan untuk magang, yaitu install software yang akan digunakan dan setting VPN agar dapat tersambung dengan server milik ACC. Persiapan ini dipandu oleh Astri.

Jumat, 4 Januari 2019

Hari ini kami mendapat pelatihan proses bisnis pada ACC. Pelatihan pertama adalah AA, yaitu AA (Account Acquisition), dan EIS (Enterprise

Information Sistem) Pelatihan ini dibawakan oleh Martin Asril. Pelatihan ini membahas 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.

Pelatihan kedua adalah pelatihan bisnis proses pada AM yaitu AM – COLL (Account Maintenance - Collection). Pelatihan ini dibawakan oleh Kevin Christian. Pelatihan kali ini membahas tentang 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,

Pelatihan ketiga adalah pelatihan BO yang dibawakan oleh Rudi. Pelatihan ini membahas tentang BO (Back Office). 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. Bagian keempat adalah training EIS (Enterprise Information Sistem), yang dibawakan oleh Eka Surya. Pada training EIS, diberitahukan mengenai

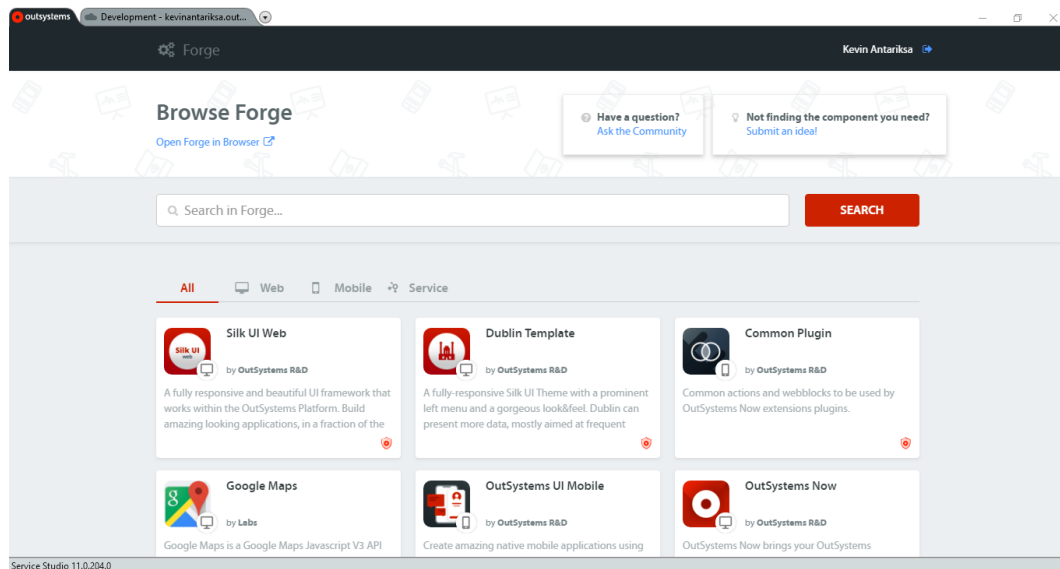
pelaporan dan pencatatan mengenai kegiatan bisnis yang ACC lakukan. EIS sendiri memiliki urutan proses yang dibedakan menjadi 2 bagian yaitu OLTP (Online Transaction Process) dan OLAP (Online Analytical Process). Pada OLTP sendiri terdapat ETL (Extract Transform Load) yang bertugas untuk menyatukan semua 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).

Training terakhir pada hari ini adalah training mengenai security di ACC. Training yang dibawakan oleh Robert menjelaskan pentingnya security pada sebuah system dan implementasi security pada ACC. Security ACC 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).

Senin, 07 Januari 2019

Hari ini kami menjalani training development program. Training hari ini membahas mengenai Outsystem. Outsystem adalah *platform* yang bisa membuat aplikasi dengan code yang sedikit. Pada hari ini kami diajari untuk membuat aplikasi pada OutSystem, Pelatihan yang diberikan berupa pelatihan penggunaan aplikasi OutSystem secara mendasar hingga membuat program sederhana. Pada akhir hari kami diuji seberapa kami dapat menangkap pelatihan yang sudah diberikan. Kami diberi tugas untuk membuat aplikasi sederhana untuk transaksi.

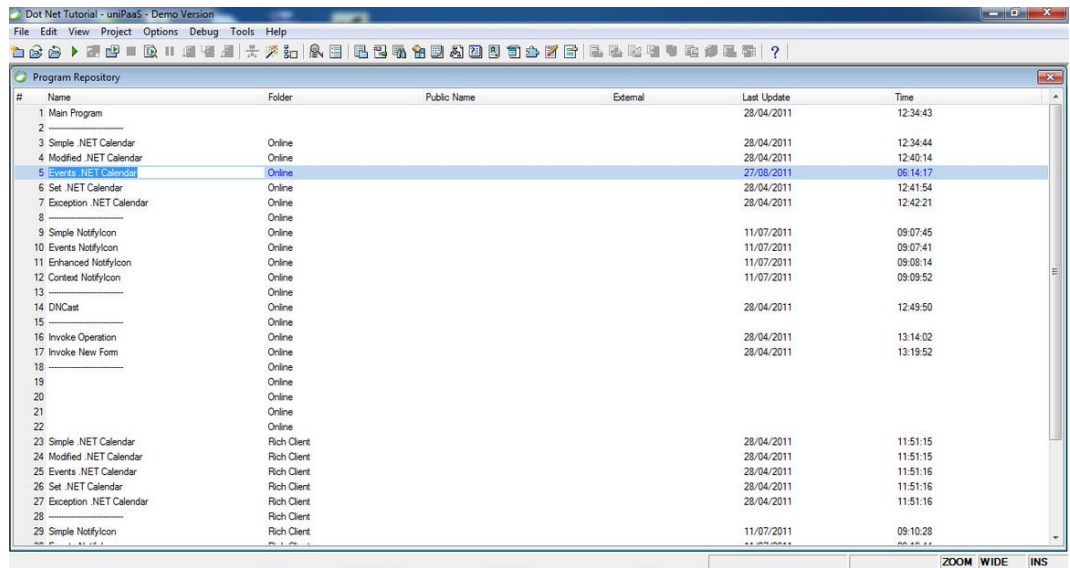
Saya hanya dapat menyelesaikan 80% dari tugas yang diberikan karena keterbatasan waktu. Tugas atau *quiz* ini yang akan diberikan hingga akhir training akan dimasukkan sebagai faktor penilaian



Gambar 2.1 Aplikasi OutSystem

Selasa, 08 Januari 2019

Hari ini kami diberikan training mengenai uniPaaS. uniPaaS, adalah sebuah aplikasi yang dibuat oleh Magic Software Enterprise. uniPaaS dapat membuat program dengan sangat sederhana (*drag&drop*) namun program tersebut dapat sangat *powerful*. uniPaaS juga tidak memerlukan banyak penulisan *code*. uniPaaS sedikit mirip dengan Visual Studio versi lama, namun uniPaaS lebih mempermudah penggunaanya (terutama pada bagian data). Menurut kata Ibu Astri(developer ACC di Jakarta) ACC masih menggunakan uniPaaS karena *core system*(transaksi, dll) yang ada di pusat dan cabang sudah diimplementasikan dah sudah dapat berjalan dengan baik, sehingga akan membutuhkan waktu yang sangat lama untuk mengganti uniPaaS. Pada hari ini kami diberi pelatihan mengenai uniPaaS secara mendalam, mulai dari cara membuat blank screen, tulisan-tulisan, membuat *form*, hingga cara memasukan data.



Gambar 2.2 Aplikasi uniPaaS

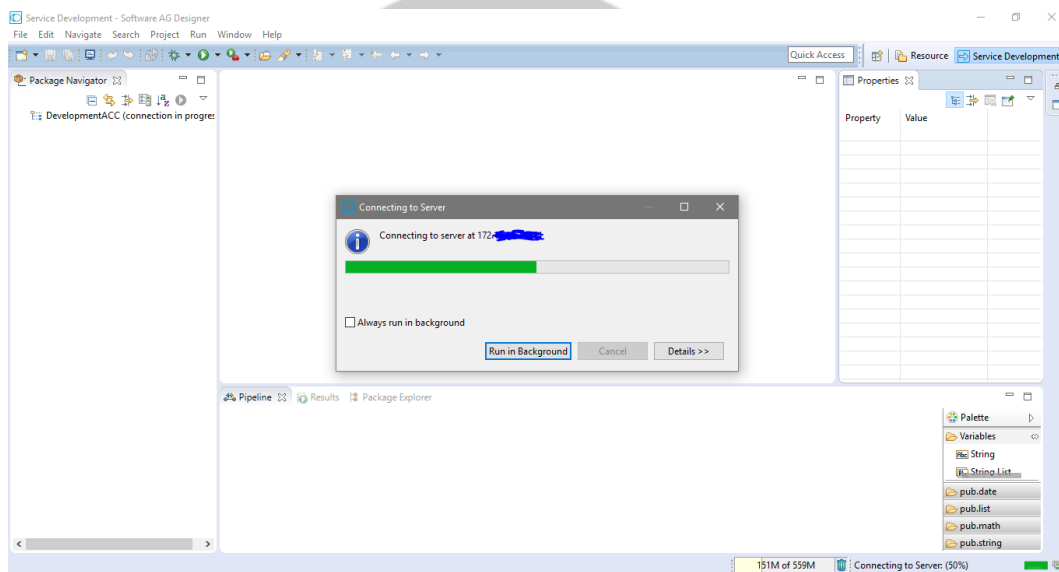
Rabu, 09 Januari 2019

Hari ini kami melanjutkan pelatihan uniPaaS. Lanjutan training yang diberikan adalah membuat tampilan lebih menarik dan penyimpanan data menggunakan PL SQL. Pada akhir hari ini, kami mendapat *quiz* untuk menguji pengetahuan dan kemampuan kami dalam menangkap training yang telah diberikan. *Quiz* yang diberikan berupa contoh kasus nyata, yaitu membuat sebuah program CRUD untuk cabang ACC, yang dapat menginput beberapa parameter. Input ini akan dimasukkan dalam database. Pada program juga diharuskan memiliki *exception* dan beberapa syarat lain. Pada hari ini saya dapat menyelesaikan *quiz* meskipun tidak sempurna (masih ada beberapa elemen untuk tampilan yang saya tidak mengerti).

Kamis, 10 Januari 2019

Hari ini kami diberikan training SOFIA. SOFIA merupakan Bahasa singkatan yang digunakan developer ACC. SOFIA yang sebenarnya adalah Software AG Designer. Salah satu kegunaan SOFIA adalah membuat database

berbasis SQL, dengan menggunakan *flowchart*. Jadi melalui SOFIA dapat membuat PL SQL secara GUI. Pelatihan yang kami dapatkan berupa pembuatan *flowchart* database untuk pengambilan dan pembuatan data. Pada akhir hari kami mendapat *quiz*. Pada *quiz* kali ini, kami diminta untuk membuat sebuah *flowchart* yang akan dapat menangani proses bisnis penjualan mobil. Pada *quiz* kali ini saya dapat menyelesaikan meskipun kurang pada beberapa bagian (kondisi dan sebagainya).



Gambar 2.3 Aplikasi Software AG Designer

Jumat, 11 Januari 2019

Hari ini kami mendapat training mengenai Laravel. Laravel merupakan *framework* PHP yang digunakan untuk mempermudah *developer* dalam membuat sebuah *website* berbasis PHP. Berdasarkan survey, Laravel adalah *framework* yang paling banyak digunakan dalam pembuatan *website*. Pada pelatihan kali ini, kami diajarkan untuk membuat web sederhana yang mampu menjalankan fungsi CRUD: create, read, update, delete. Untuk database yang digunakan, kami menggunakan MySQL. Lalu, dalam mengisi SQL dengan data *dummy* kami diajarkan menggunakan *faker*. Pada akhir hari, kami diberi *quiz* untuk melatih pemahaman

kami. *Quiz* ini kami disuruh untuk membuat web yang mampu menangani CRUD, login, terdapat halaman profil dan dapat mengupload gambar.



Gambar 2.4 Framework Laravel

Senin, 14 Januari 2019

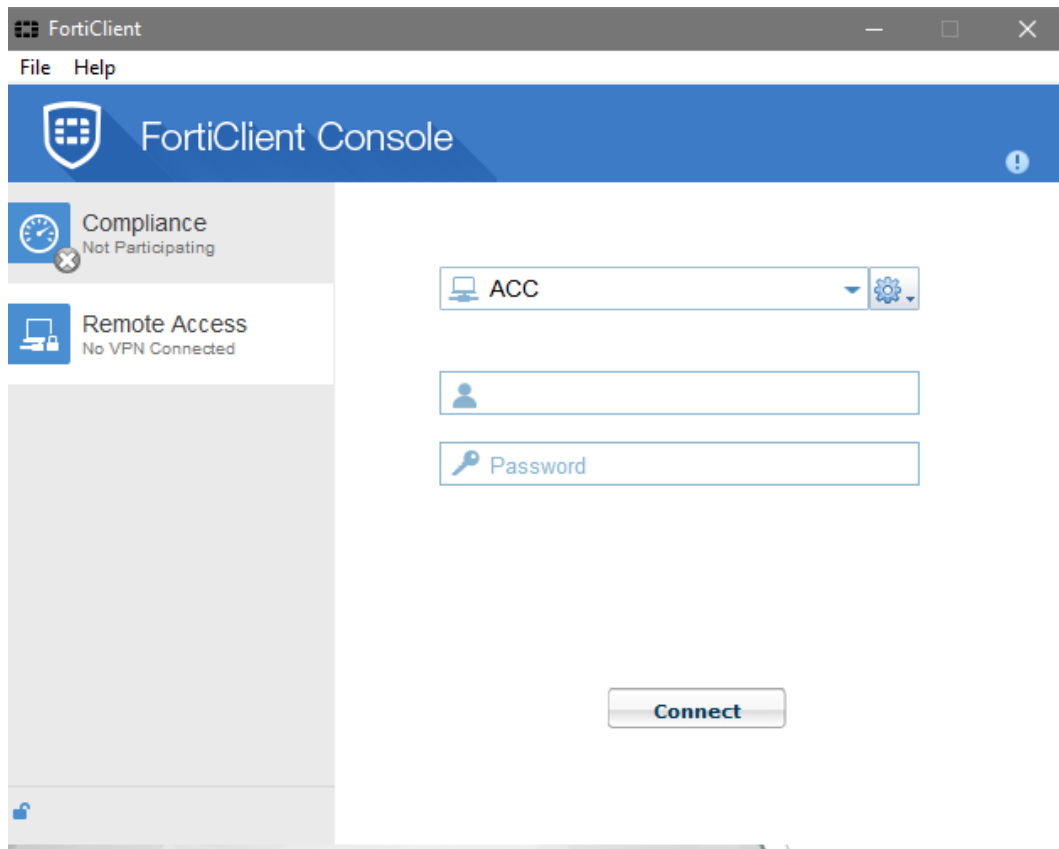
Hari ini kami menjalani training mandiri. Seperti yang sudah dijelaskan sebelum nya, OutSystem memiliki *channel* yang berisikan video tutorial untuk *Development Web* dan *Development Mobile Application*. Dalam satu rangkaian tutorial, biasanya terdapat 20 lebih video dengan durasi Training mulai dari 20 jam. Pada hari ini saya mengerjakan sejumlah 13 training yang berisikan tutorial dasar untuk membuat aplikasi web sederhana menggunakan Outsystem .Training saya lanjutkan dirumah meskipun saya tidak menyelesaikan seluruh video training tersebut. Yang saya dapatkan dari melihat video training tersebut adalah cara setup *environment* termasuk persiapan komponen-komponen dan *packages*, menggunakan template, membuat pop up, link navigasi, dan debug.

Selasa 15 Januari 2019

Hari ini, saya melanjutkan training mandiri OutSystem. Jika pada hari senin *platform* yang saya kerjakan adalah Web, maka hari ini saya melakukan pelatihan pada *platform* Mobile. Yang saya coba kerjakan pada hari ini adalah 15 video tutorial *mobile application* awal OutSystem. Dalam video tersebut saya mempelajari beberapa hal yang cukup mirip dengan *platform* web. Beberapa hal yang berbeda adalah cara deploy, *navigation bar*, keamanan PIN, tampilan-tampilan, dan hal yang lain. Hingga akhir hari ini saya masih belum bisa menyelesaikan seluruh tutorialnya.

Rabu, 16 Januari 2019

Hari ini berbeda dengan training sebelumnya. Pada hari ini kami diberikan training PL SQL. PL SQL merupakan Procedural Language SQL. Secara singkat PL SQL mampu memberikan data yang diminta hanya dengan memanggil nama prosedur SQL dan parameter-parameter yang dibutuhkan. Pada training ini kami dilatih untuk dapat membuat sebuah PL SQL pada komputer ACC pusat. Kami menggunakan VPN Forti Client dan software *remote* bawaan Windows untuk dapat menggunakan komputer ACC. Terlebih dahulu dibuat tabel yang merupakan tabel yang berisikan tentang merk mobil, nama mobil, tanggal *create/update*, dan user yang melakukan *create/update*. PL SQL yang dibuat adalah PL SQL untuk melakukan *select*, *create*, dan *update* pada tabel yang dibuat sebelumnya.



Gambar 2.5 FortiClient VPN

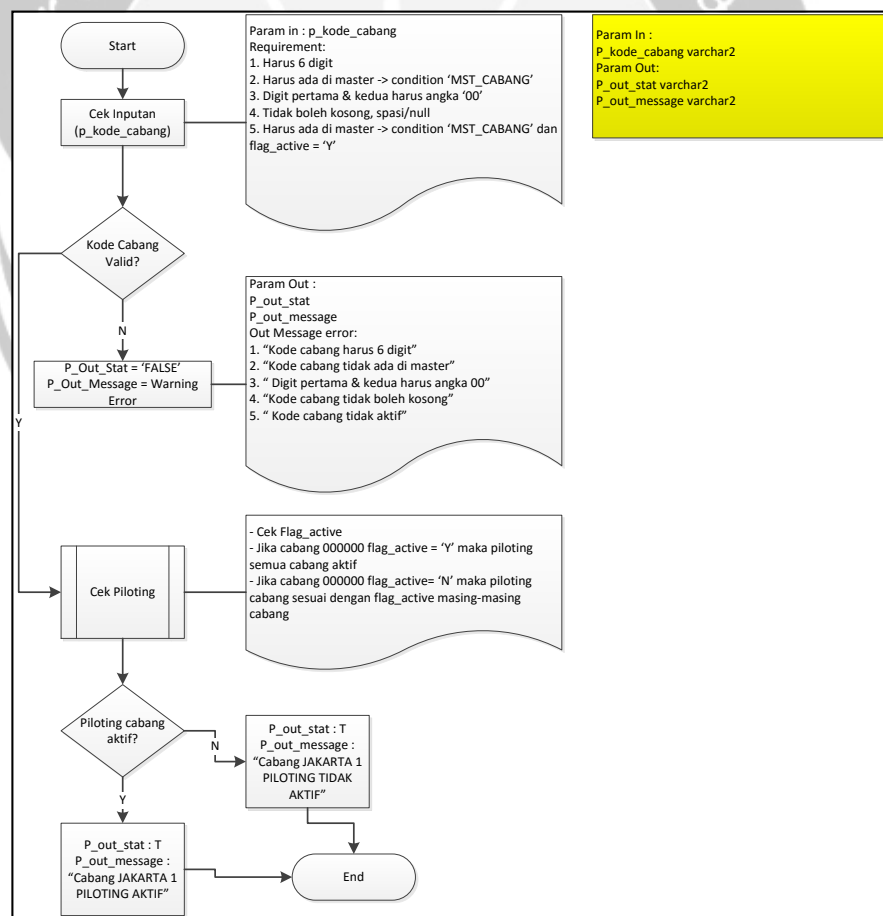
Kamis, 17 Januari 2019

Hari ini pelatihan PL/SQL dilanjutkan. Ibu Awa meminta untuk menyelesaikan PL/SQL yang kemarin belum selesai dan *submit* bagi yang sudah selesai. Karena saya sudah selesai, saya *submit* pekerjaan saya. Setelah saya submit saya memutuskan untuk kembali belajar PL/SQL. Pada sore hari kami diberikan pelatihan lanjutan. Kami diminta untuk membuat tabel, yang berisikan tentang cabang, kode cabang, status cabang, dan beberapa hal *credential* lain. Berikutnya kami diminta untuk melakukan insert data-data mengenai cabang. Setelah itu kami diminta untuk membuat fungsi untuk menampilkan nama cabang dan beberapa status lain berdasarkan input kode cabang. Prosedur yang diminta adalah prosedur untuk mengecek input kode cabang lalu mengecek status-status terkait dan

memberikan status pada cabang tersebut. Saya tidak berhasil menyelesaikan *quiz*, hanya tinggal pemberian kondisi/status pada cabang.

Jumat, 18 Januari 2019

Hari ini pelatihan SQL dan pelatihan OutSystem dilanjutkan secara mandiri karena supervisor kami Ibu Awa sedang berhalangan hadir. Saya melanjutkan PL/SQL saya namun masih mengalami kendala. Akhirnya saya mampu menyelesaikan *quiz* tersebut diberikan petunjuk oleh teman saya yang sudah selesai. Setelah *submit quiz* saya melanjutkan training OutSystem. Saya hanya melihat video tutorial tentang *environment* OutSystem dan fungsionalitas program(*shortcut*, kemampuan program).



Gambar 2.6 PL/SQL yang dibuat

Senin, 21 Januari 2019

Hari ini saat pembagian proyek, saya dan Desya mendapatkan proyek Outsystem-Laravel. Pada proyek kali ini, kami diminta untuk membuat Web di Laravel. Web yang dibuat adalah web untuk CMS (Content Management System) pada ACC BidMart. ACC BidMart merupakan aplikasi Android untuk proses pelelangan mobil-mobil bekas. CMS nya berbasis web dan masih ada di OutSystem versi ini merupakan versi yang belum di deploy(hosting). Guna efisiensi biaya, kami diminta untuk membuat versi Laravel dari CMS BidMart yang ada di OutSystem. Untuk data yang digunakan, kami membuat API yang akan menghubungkan data-data ACC dengan Laravel yang akan dibuat. *Environment* yang akan digunakan pada saat develop CMS BidMart adalah *environment* akun saya sendiri, belum diintegrasikan dengan *environment* CMS BidMart yang ada di ACC. *Environment* yang dimaksud disini adalah tempat untuk mengerjakan *Website*. Dalam satu akun OutSystem bisa ada banyak *environment* baik itu *environment* untuk web maupun mobile. Tugas pertama saya adalah *develop sub-menu* Balai Lelang dan Unit.

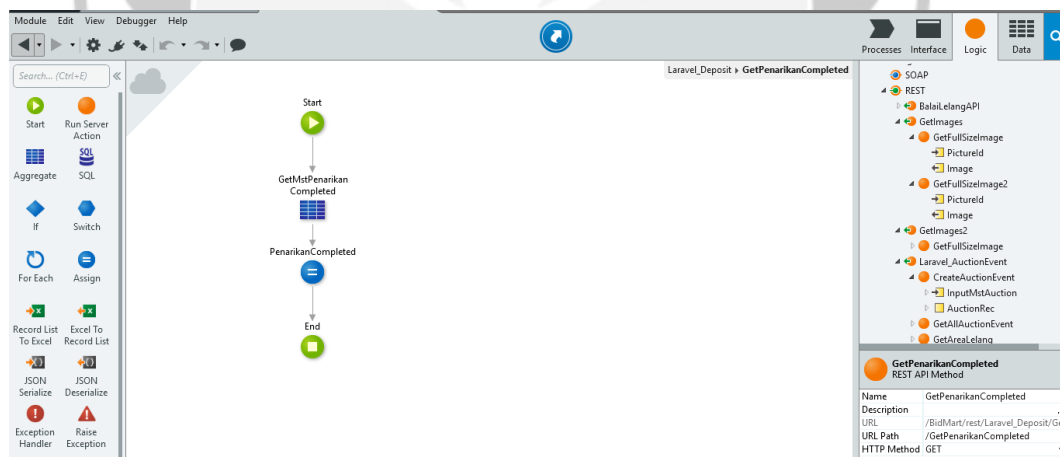


Gambar 2.7 ACC BidMart

Selasa, 22 Januari 2019

Hari ini saya memulai pengerjaan proyek. Pertama saya mencoba memahami mengenai API di OutSystem. API di OutSystem dapat mempermudah pekerjaan dalam proyek karena dapat diatur ketentuan-ketentuan seperti jumlah minimum. Selain itu pada OutSystem data dapat berintegrasi satu dengan yang lain dan dapat dikeluarkan menjadi data JSON. Membuat API dengan OutSystem lebih mudah dibandingkan saat saya membuat API menggunakan Laravel. Satu lagi keunggulan

API OutSystem adalah API sudah langsung dapat diakses melalui link yang disediakan (API tersedia secara online). Link untuk API nya juga sudah disediakan oleh OutSystem, sehingga sangat mudah untuk dicek dan digunakan. Pada hari ini saya juga mencoba memahami cara penggunaan API pada Laravel. Penggunaan API pada Laravel cukup mudah, terutama bila menggunakan *packages*, tidak sesulit saat pembuatan API nya.



Gambar 2.8 Contoh API sederhana di OutSystem

Rabu, 23 Januari 2019

Hari ini saya mulai mencoba membuat API untuk GET, POST, dan DELETE. Pada OutSystem, update dapat dijadikan satu dengan create, yaitu CreateOrUpdate. Hal ini sangat membantu karena dapat mengurangi duplikasi data.

Data yang sama, saat create, akan di cek terlebih dahulu, sehingga data yang sama tidak akan masuk ke database, namun data tersebut akan meng-update data yang sudah ada sebelumnya. Mempelajari OutSystem menurut saya gampang-gampang susah. Belum banyak tutorial yang tersedia, forum untuk melihat error (seperti stackoverflow) masih terbatas dan belum banyak yang dapat dilihat. Lalu dokumentasi yang cukup membingungkan memakan waktu lama dalam memahami. Video tutorial yang ada dalam websitenya juga tidak banyak membantu karena video tutorial itu kebanyakan video untuk membuat web/mobile application yang berkesinambungan. Beruntung bagi saya karena terdapat sebuah video YouTube yang menjelaskan pembuatan API untuk CRUD yang cukup membantu. Lalu saya mencoba mengimplementasikan video tersebut untuk membuat API pada *environment* lokal saya. API yang akan saya buat adalah API untuk Balai Lelang dan Unit.

Kamis, 24 Januari 2019

Setelah kemarin mencoba untuk membuat API melalui OutSystem, hari ini saya sudah bisa mendapatkan data GET. Data yang didapatkan merupakan data local, dan belum terkoneksi dengan database ACC. Database terdiri dari berbagai tabel dan tabel dibuat dengan cara yang sangat mudah. Tabel yang dibutuhkan, dibuat terlebih dahulu di Excel (dibuat dalam bentuk tabel dan dapat diisi data). Setelah itu tabel diupload ke OutSystem dan dapat digunakan. Hal lain yang saya lakukan hari ini adalah membuat API untuk CreateOrUpdate Balai Lelang lalu mengecek CreateOrUpdate Balai Lelang menggunakan Postman. Setelah berhasil, API Unit saya coba buat menggunakan OutSystem.


```
1 - [
2 -   {
3 -     "Id": 1,
4 -     "AreaLelang": "BANJARMASIN",
5 -     "CodeAreaLelang": "BJM"
6 -   },
7 -   {
8 -     "Id": 2,
9 -     "AreaLelang": "PEKANBARU",
10 -    "CodeAreaLelang": "PKB"
11 -   },
12 -   {
13 -     "Id": 3,
14 -     "AreaLelang": "ALL",
15 -     "CodeAreaLelang": "ALL"
16 -   },
17 -   {
18 -     "Id": 4,
19 -     "AreaLelang": "UJUNG PANDANG",
20 -     "CodeAreaLelang": "UJP"
21 -   },
22 -   {
23 -     "Id": 5,
24 -     "AreaLelang": "Jakarta Selatan",
25 -     "CodeAreaLelang": "DKI2"
26 -   },
27 - }
```

Gambar 2.9 Contoh hasil GET API

Jumat, 25 Januari 2019

Hari ini saya mencoba implementasi API yang telah saya buat ke dalam halaman Web. Pada hari ini, karena Laravel menggunakan MVC (Model View Controller) saya mengawali dengan membuat controller dan model untuk Balai Lelang dan Unit yang akan saya buat. Controller adalah bagian yang mengatur proses-proses yang diinginkan seperti create, update, view, dan delete. Setelah itu saya juga membuat halaman view pada Balai Lelang dan Unit. Selain itu hari ini juga saya menyesuaikan halaman web agar dapat menampilkan data-data sesuai API(read) dan menerima data untuk selanjutnya dimasukkan dalam database(create). Implementasi dimulai dengan membuat route-route pada web.php agar nantinya web dapat berjalan dengan baik dan sesuai harapan. Untuk implementasi get sangat mudah karena hanya perlu memanggil url, dan *decode* JSON agar dapat

ditampilkan. Saat ingin create, harus menggunakan fungsi `$client->request` dari *framework* Guzzle.

```
$client = new \GuzzleHttp\Client();  
$query= $request->search;  
  
$req = $client->get("https://kevinantariksa.outsystemscloud.com/API/rest/BalailelangAPI/SearchByQuery?Balailelang  
$response = $req->getBody()->getContents();  
$response = collect(json_decode($response,true));
```

Gambar 2.10 Contoh penggunaan Guzzle

Senin, 28 Januari 2019

Hari ini saya membuat halaman untuk menampilkan data-data Balai Lelang. Hal lain yang akan saya buat adalah *read* dan *create* Balai Lelang. Halaman *read*/tampilan hanyalah halaman untuk menampilkan (berisikan tabel dan data-data terkait) dan halaman *create* berisikan form untuk menginput data-data. Setelah saya selesai membuat *read* dan *create* Balai Lelang, saya juga akan menambahkan fungsi *read* dan *create* untuk Unit. Tidak terlalu sulit karena hanya sedikit berbeda dengan Balai Lelang. Menu Unit memiliki lebih banyak kolom di tabel tampilannya. Pada hari ini saya juga memperbaiki tampilan pada Balai Lelang dan Unit saya, yaitu dengan menggunakan Bootstrap (gambar yang ada dalam laporan adalah tampilan akhir setelah menggunakan Bootstrap. Pada hari ini juga penggunaan Bootstrap pada halaman Unit dan Balai Lelang telah selesai.

Balai Lelang

Master Management

Balai Lelang

Online Event

Unit

Master GCM

Auction Event

Deposit

Content Management

View History and Transaction

127.0.0.1:8000/viewBalaiLelang

Create New Balai Lelang

Upload New File

Search

Type balai code or description

Search

Reset

Balai code	Description	Address	E-mail	Contact	Is Active	Action
112	Balai Artha	Depok Fatmawati	artha@ac	Balai Lelang Artha / 081290634556	N	Delete - Edit
B022	Balai Lelang Larasati	Jl. Erlangga IV No. 6. Kebayoran Baru, Jakarta Selatan 12110. Indonesia DKI Jakarta	civlcmb@sharklasers.com	Balai Lelang Larasati / 7236075	N	Delete - Edit
B023	Balai Lelang	Jl. Erlangga IV No. 6. Kebayoran Baru, Jakarta	civlcmb@sharklasers.com	Balai Lelang Larasati /	N	Delete -

Gambar 2.11 Tampilan yang dibuat

Selasa, 29 Januari 2019

Hari ini saya menyadari keteledoran saya, yaitu kurang memasukan satu kolom dalam database. Saya melakukan perbaikan pada tabel Unit yang berimplikasi pada perubahan tampilan di halaman untuk menampilkan Unit dan membuat Unit. Untuk persiapan membuat fungsionalitas update, saya mencoba menampilkan ID untuk masing-masing Balai Lelang dengan cara meletakkan `{{array['Id']}}` disebuah kolom baru, yang nanti nya akan menjadi kolom *action (delete-update)*. Setelah ID ditampilkan dengan baik, maka dapat dibuat route untuk mengupdate, code nya adalah `updateBalaiLelang/{id}` yang akan mengarahkan pada `BalaiLelangController@edit'`). Selanjutnya adalah setelah berhasil masuk controller edit, controller edit tersebut akan mereturn data yang memenuhi `BalaiLelangId=id`. Lalu akan diarahkan ke halaman untuk mengupdate Balai Lelang. Pada halaman update tersebut akan ditampilkan data-data Balai Lelang dengan Id terkait di dalam kotak input, lalu data yang ingin diubah dapat diubah.

Rabu, 30 Januari 2019

Setelah read dan create dapat berjalan dengan baik dalam web yang saya buat, hal berikutnya yang saya kerjakan adalah membuat Update yang kurang lebih sama dan tombol delete yang berfungsi untuk menghapus data. Pada pembuatan halaman update, dibuat *form* dan *input box* yang sama dengan bagian create, namun dibedakan pada value *input* tersebut. Value diambil dari data JSON yang direturnkan melalui controller. Karena saya mereturnkan data-data JSON dari controller dengan nama *variable* 'array', maka contoh value untuk *input box* tahun adalah `{{ $array['Tahun'] }}`. Hari ini saya menambahkan API untuk mengedit dan menghapus data. API tersebut berfungsi untuk mengambil ID dari Balai Lelang maupun Unit, lalu akan memberikan *return* data-data terkait dari ID yang diambil.

Kamis, 31 Januari 2019

Hari ini saya menemukan beberapa kekurangan pada sistem yang saya buat. CRUD yang sudah saya buat mengalami kekurangan dibagian update, terutama karena update belum saya coba dengan postman. Untuk Create, Read, dan Delete sudah dapat berjalan semestinya. Pembetulan hanya pada OutSystem, karena web yang saya buat sepertinya tidak bermasalah. Setelah dilihat, ternyata permasalahannya adalah belum ada operator yang menyatakan bahwa ID yang di edit sama dengan ID yang ada di database, sehingga data tidak terbaca. Yang saya lakukan adalah menambahkan `Input.ID = BalaiLelang.ID` pada Balai Lelang dan `Input.ID = Unit.ID` untuk Unit. Setelah saya uji coba terlebih dahulu di Postman, saya implementasikan dan fungsional update telah dapat berjalan dengan baik.

Penambahan fitur search juga dilakukan hari ini. Untuk menambahkan fitur search, harus dibuat terlebih dahulu API untuk melakukan search. Konsep search ini adalah dengan melakukan GET data yang berdasarkan keyword yang diperlukan. Setelah saya membuat API search, hal berikutnya yang saya buat adalah form untuk search pada Unit dan Balai Lelang. Cara kerja search ini cukup simple,

setelah keyword pencarian diketik, tombol search akan mentrigger controller, lalu controller akan memproses data dan menampilkan data.

Jumat, 1 Februari 2019

Hari ini saya memperbaiki bootstrap sehingga sesuai dengan apa yang ada di gambar. Memperbaiki bootstrap memakan waktu agak lama karena Bootstrap tersebut merupakan hasil download template sehingga perlu memahami CSS yang ada dan mengimplementasikan pada code saya. Berikutnya saya juga menambahkan fitur untuk mereset search. Cara kerja dari tombol reset ini adalah ketika tombol ditekan, maka Controller akan mengakses API yang mengambil seluruh data Unit/Balai Lelang lalu dikembalikan ke halaman tampilan Balai Lelang/Unit.

Senin, 4 Februari 2019

Hari ini saya menyesuaikan CMS yang saya buat dengan CMS yang diminta, pada hari ini saya menambahkan beberapa exception, yaitu pada halaman create. Pada CMS BidMart milik ACC, create Unit dan create Balai Lelang. Exception yang ditambahkan adalah tidak diperbolehkannya create selama masih ada input wajib yang belum diisi, jumlah karakter, dan exception untuk mengecek file kosong pada saat upload file excel. Penambahan exception tidak memakan waktu lama karena tidak menggunakan JavaScript.

Rabu, 6 Februari 2019

Hari ini saya melakukan testing untuk memastikan Unit dan Balai Lelang saya sesuai dengan yang diinginkan. Testing yang saya lakukan meliputi mencoba menu-menu yang saya buat, yaitu halaman tampilan Unit dan Balai Lelang, halaman untuk membuat Unit dan Balai Lelang baru. Selain itu ada menu update

dan fungsionalitas untuk delete serta menu search. Beberapa bagian yang masih kurang sempurna, seperti exception yang belum terbaca jika menekan Search padahal input masih kosong. Setelah melakukan testing saya melengkapi data-data pada Balai Lelang saya seperti nama kota yang akan dibuka Balai Lelang baru.

Kamis, 7 Februari 2019

Hari ini pekerjaan saya ditesting dan terdapat kekurangan. Kekurangan yang akan saya kerjakan setelah ini adalah bagian upload excel dan upload gambar. Pada hari ini saya juga membuat pagination. Pagination sendiri berfungsi ketika data yang ada sangat banyak, dan ingin ditampilkan dalam jumlah tertentu, dapat digunakan pagination. Pagination yang saya buat, membatasi data tampil sebanyak hanya 10 data. Setelah itu data lain dapat dilihat di *page* yang lain.

Jumat, 8 Februari 2019

Hari ini saya diminta untuk menambahkan fitur upload gambar agak sesuai dengan CMS yang diminta. Submenu yang membutuhkan fitur ini adalah submenu Unit. Pada submenu Unit dibutuhkan upload gambar agar Unit yang akan dilelang, dapat diupload gambarnya sehingga memudahkan untuk pencocokan dan atau penawaran. Setelah diberitahu, saya mencari-cari tutorial untuk mengupload gambar dari Laravel. Saya memutuskan menunda belajar JavaScript untuk membuat update onclick pada API. Saya mencoba beberapa tutorial upload gambar namun hasilnya nihil. Hingga hari ini berakhir saya masih belum dapat menemukan tutorial yang pas untuk upload gambar.

Senin, 11 Februari 2019

Hari ini saya berusaha menyelesaikan upload gambar pada Unit. Setelah saya bertanya pada beberapa orang, biasanya untuk upload gambar ke database,

digunakan base64, yaitu data gambar diubah menjadi data teks. Saya sudah terlanjur memberi tipe data *Picture* pada tabel di database saya. Saya harus mengubah hal ini terlebih dahulu. Setelah selesai mengubah, saya mencari tutorial untuk base64. Namun saya masih belum dapat mengimplementasikannya. Setelah makan siang saya mendapat tutorial untuk mengupload gambar.

Upload gambar ini bekerja dengan cara mengambil file gambar yang diupload, dicocokkan tipe datanya, diberikan nama file tertentu(yang diinginkan atau yang dijadikan syarat), lalu akan disimpan dalam sebuah folder. Yang disimpan dalam database adalah lokasi gambar, dan saat ingin ditampilkan, sistem hanya perlu membaca lokasi gambar lalu menampilkannya. Hari ini saya dapat menyelesaikan hal ini.

Selasa, 12 Februari 2019

Hari ini saya melanjutkan pekerjaan pada halaman Unit. Pada halaman terdapat fungsionalitas untuk melakukan Upload file berformat .xlsx. Tugas saya adalah mencoba upload file excel berformat .xlsx ke dalam sehingga dapat terbaca dan dimasukkan dalam database. Saya mencoba mencari cara untuk upload dan dapat saya temukan di StackOverflow. Malahan ada tutorial yang bekerja seperti saat saya upload *file* gambar, yaitu mengupload ke folder dan memberikan format nama tertentu. Pencarian memakan waktu cukup lama, saya belum dapat menemukannya hari ini.

Rabu, 13 Februari 2019

Sebagai lanjutan dari hari sebelumnya, pada hari ini saya masih mencoba untuk mengupload data .xlsx melalui Laravel, lalu data yang diupload dapat dibaca sistem dan ditambahkan secara langsung ke API. Akhirnya saya mendapatkan code dari Stack Overflow, yang sesuai dengan yang saya harapkan. Cara kerja code tersebut adalah mengecek kesesuaian nama kolom lalu mengubah kolom-kolom

pada excel menjadi *request* (sejenis pada request dari form). Request tersebut lalu dapat dimasukkan ke database melalui API.

Kamis, 14 Februari 2019

Hari ini saya memastikan fungsi upload *file* Excel berjalan dengan baik, Saya mencoba testing pada halaman-halaman web yang sudah saya buat. Tes meliputi mencoba setiap route yang saya buat, mencoba input pada bagian create dan update, memastikan data yang dihapus telah dihapus. Setiap route yang saya coba sudah mengarahkan saya ke bagian yang memang saya inginkan. Mencoba input pada bagian create dan update meliputi uji coba karakter special, uji input kosong, input yang jumlah karakternya melebihi batas, upload file dengan format yang berbeda, dan input yang kosong.

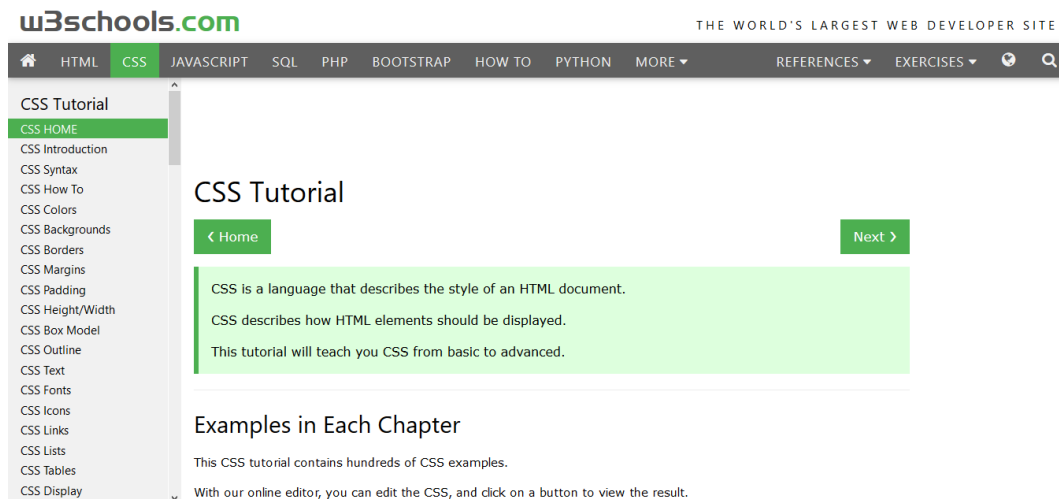
Jumat, 15 Februari 2019

Hari ini saya melihat hasil upload, ternyata saya terbalik dalam memasukan data ke kolom, hal ini dapat saya perbaiki. Setelah itu saya menunggu tugas berikutnya. Sembari menunggu tugas berikutnya saya memutuskan untuk membenahi tampilan pada CRUD Balai Lelang dan Unit. Pada bagian create-update Unit dan Balai Lelang saya atur besar kotak inputnya. Selain itu saya juga mengatur warna kotak pada bagian tampilan data Balai Lelang dan Unit.

Senin, 18 Februari 2019

Hari ini saya menyadari saya masih kurang baik pada CSS, pada hari ini saya mencoba mempelajari CSS. Saya belajar CSS dari W3School. Hal-hal yang saya pelajari merupakan basic CSS seperti padding, hover, mengatur jenis *font* dan mengatur warna text-background. Selain itu saya mempelajari tentang konsep `<div></div>` yang ‘mengurung’ code html agar memiliki style tertentu. CSS

tetap harus diatur, akan mempengaruhi halaman HTML tersebut berdasarkan ID, *name*, atau berdasarkan tag. Diluar syarat tersebut, CSS tidak akan mempengaruhi elemen HTML lain.



Gambar 2.12 Halaman tutorial CSS w3schools

Selasa, 19 Februari 2019

Hari ini saya baru memperoleh akses database OutSystem ACC melalui akun ACC Developer. Database yang saya pakai bukanlah database primer, tapi tabel-tabel dan relasi antar tabel sudah sama dengan database yang digunakan di sistem utama ACC. Karena saya baru saja mendapatkan akses, maka saya harus mengulang membuat API lagi, menyesuaikan Laravel dengan API saya yang saya buat. *Environment* BidMart CMS yang ada di ACC Dev sangatlah rumit, karena tidak sama dengan yang saya pelajari melalui tutorial. Hal ini yang membuat saya harus mempelajari dan melihat-lihat *environment* ACC Dev terlebih dahulu. Hal yang saya lihat adalah aliran data, database, dan kondisi-kondisi yang telah ditentukan harus dipenuhi sebelum melakukan operasi CRUD.

Rabu, 20 Februari 2019

Hari ini saya mencoba mengimplementasikan apa yang saya pelajari beberapa hari yang lalu. Saya mencoba menerapkan padding. Saya berusaha memindahkan letak tombol upload file menjadi terletak di kanan atas namun belum berhasil. Pada hari ini saya juga mempelajari lebih lagi mengenai base64. Karena masih buntu, saya memutuskan untuk kembali mempelajari *environment* ACC Developer. Kendala saya adalah *environment* BidMart pada ACC Dev sangatlah rumit, banyak hal yang harus saya perhatikan lebih lagi seperti relasi antar database (karena ACC BidMart terhubung dengan sistem Core ACC). Database yang digunakan dalam *develop* dan *deployment* berbeda, sehingga apa yang saya lakukan tidak mempengaruhi sistem *core* ACC. Selain itu saya juga mempelajari bahwa setiap operasi CRUD pada ACC BidMart harus memenuhi banyak syarat seperti user session dan hal-hal *credential* lain. Akhirnya saya memutuskan untuk mempelajari sembari membuat API yang sesuai dengan proses bisnis ACC.

Kamis, 21 Februari 2019

Hari ini, saya melanjutkan membuat API, lalu saya mencoba implementasi, yaitu mengganti link yang sudah ada pada controller dengan link baru. API yang sudah jadi hari ini adalah API untuk GET data Balai Lelang dan Unit. Setelah API jadi, saya coba ke halaman web saya dan berhasil. Setelah itu saya membuat create, update, dan delete untuk masing-masing Balai Lelang. Membuat API tidak memakan waktu lama karena saya sudah pernah membuatnya. Setelah itu saya masukan link-link API ke dalam projek saya. Setelah itu saya melakukan pengecekan dan ternyata API yang saya buat pada create dan update masih belum bisa digunakan. Setelah saya berkonsultasi dengan Ibu Christie, saya dimudahkan dengan cara menggunakan 'CreateOrUpdate' yang memenuhi bisnis proses yang sudah dibuat sebelumnya.

Jumat, 22 Februari 2019

Hari ini saya memperbaiki kesalahan pembuatan API saya kemarin. Setelah saya coba pada Postman dan setelah saya cek datanya pada *environment* ACC Dev, maka dapat dipastikan 'CreateOrUpdate' yang telah dibuat sudah berhasil. Pada hari ini, saya juga melakukan integrasi pada API yang saya buat. Integrasi meliputi penggunaan API pada web application saya. Pada hari ini integrasi kurang berjalan dengan baik karena kesalahan saya pada bagian controller. Setelah integrasi selesai, maka tugas saya pada bagian ini juga sudah selesai.

Senin, 25 Februari 2019

Ibu Christie memberikan tugas baru pada saya. Saya dipercayakan untuk membuat menu event lelang. Secara garis besar, menu event lelang berfungsi untuk membuat event pelelangan kendaraan pada suatu tempat di tanggal tertentu. Hari ini saya mempelajari alur untuk event lelang, dan mencoba membuat API untuk GET event lelang, sesuai dengan data Event lelang ACC Dev. Karena Event lelang membutuhkan banyak data seperti Lokasi(nama kota yang memiliki cabang ACC) dan data Balai Lelang, maka pembuatan API ini memakan waktu cukup lama. Pada hari ini juga saya menyelesaikan API untuk POST dan GET data, meskipun pada bagian POST belum sempurna.

Selasa, 26 Februari 2019

Hari ini saya mencoba API yang telah saya buat kemarin. API untuk POST akhirnya saya perbaiki (pada alur create data terdapat *variable* yang salah nama). Selain itu pada beberapa data yang saya coba, terdapat kesalahan input. Saya mencoba membenahi kesalahan tersebut. Pada hari ini juga saya mencoba API yang telah saya buat sebelumnya menggunakan Postman. Setelah kedua API saya coba, saya membuat halaman tampilan dan create untuk event lelang.

Rabu, 27 Februari 2019

Hari ini, saya melanjutkan membuat tampilan untuk *view* auction event. Yang saya buat hari ini adalah halaman untuk menampilkan seluruh data-data yang ada di API. API sendiri sudah saya buat dan saya coba kemarin. Setelah sudah jadi, saya mencoba untuk mempelajari JavaScript untuk menampilkan data Unit. JavaScript yang saya pelajari berhubungan dengan *event handler* OnClick.

Kamis, 28 Februari 2019

Hari ini saya membuat auction event, melanjutkan mempelajari create auction event. Pada proses create auction event ini sangat rumit, karena terintegrasi dengan banyak tabel di *core system* ACC. Saya pada akhirnya juga melihat sedikit flow pada *core system* untuk mempermudah pembuatan API saya. Hingga hari berakhir, saya masih mendapati error pada bagian generate nomor LOT dan bagian update unit history. Update unit history, tidak akan dapat berjalan apabila dalam proses belum didapatkan nomor LOT baru.

Jumat, 1 Maret 2019

Hari ini saya melanjutkan membuat API untuk auction event. API yang saya buat adalah API untuk create. API tersebut kemarin belum selesai, berusaha saya selesaikan hari ini. Namun, hasilnya masih sama, masih terdapat beberapa kesalahan pada bagian nomor LOT. Nomor LOT masih belum dapat dibuat. Lalu saya memutuskan untuk mencoba melakukan generate nomor LOT secara manual, namun saya masih belum dapat membuat nomor LOT secara benar.

Senin, 4 Maret 2019

Hari ini saya melanjutkan membuat tampilan dan berusaha membenahi beberapa kesalahan seperti untuk Unit yang dipilih masih belum masuk dalam tabel Unit History. Salah satu kendala dalam create auction adalah generate nomor LOT. Nomor LOT ini dimasukan ke dalam Unit sehingga Unit mendapatkan nomor LOT yang sesuai dengan kaidah penomoran ACC. Nomor LOT juga harus unik, namun tidak menggunakan random generator. Unik disini adalah nomornya urut dari A001-A999 berurut abjad hingga Z001-Z999. Setiap Unit yang sudah di *assign* ke Auction Event harus memiliki nomor LOT. Terakhir saya mencoba generate nomor LOT gagal, nomor LOT sudah ter *generate* namun belum masuk ke dalam data Unit. Pada hari ini saya mencoba menyesuaikan API saya agar sesuai dengan proses generate LOT

Selasa, 5 Maret 2019

Hari ini saya melanjutkan membuat auction event setelah ada stuck di bagian create. Saya membuat pada create, agar setelah salah satu Unit dipilih, tanggal Auction unit tersebut langsung masuk ke dalam kotak input. Untuk membuatnya, digunakan JQuery event on change. Sehingga ketika *checkbox* Unit dipilih, JQuery akan memanggil fungsi event on change yang didalamnya memanggil data-data Unit tersebut, lalu ditampilkan dalam kotak input terkait.

Rabu, 6 Maret 2019

Hari ini saya mencoba-coba API yang telah saya buat. Selain itu saya juga melakukan implementasi API pada auction event. Meskipun pada bagian create masih belum jadi, tapi API-API lain sudah dapat berjalan dengan baik. Setelah mencoba API, saya mengimplementasikan API saya ke dalam web yang saya buat. Setelah selesai, saya kembali mencoba menyelesaikan API create yang masih

belum selesai. Hari ini saya juga diberitahu untuk menghilangkan fitur *delete* pada Balai Lelang.

Jumat, 8 Maret 2019

Hari ini saya menyesuaikan tampilan yang saya buat karena pada contoh web ACC BidMart terdapat beberapa elemen yang harus menggunakan JavaScript. Karena pengetahuan JavaScript saya belum banyak, saya mempelajari tentang JavaScript. JavaScript yang saya pelajari adalah JS yang mendasar seperti *echo*, operasi hitung, array, dan JSON. Saya juga mempelajari tentang mengambil data API pada javascript dengan menggunakan ajax. Selain mempelajari JavaScript, saya tidak melakukan hal lain hari ini.

Senin, 11 Maret 2019

Pada hari ini saya diberikan tugas baru oleh Ibu Christie. Saya dipercayakan untuk membuat menu Deposit, yang memiliki *submenu* Top Up dan Penarikan. Top Up dan Penarikan disini merupakan salah satu bagian *account management* dimana setiap akun yang ingin mengikuti pelelangan kendaraan harus sudah Top Up ke akunnya dan ini digunakan sebagai tanda jadi apabila akun tersebut memenangkan pelelangan. Pelelangan dapat dilihat menggunakan aplikasi Android ACC BidMart, disana juga merupakan tempat membuat akun. Sedangkan penarikan merupakan fungsi apabila Balai Lelang atau pengguna ingin menarik saldonya yang ada di dalam aplikasi. Karena menu deposit ini berhubungan dengan uang yang banyak, maka API untuk CRUD nya sangat lah rumit karena perlu pengecekan banyak dan terhubung dengan banyak database. Pada hari ini saya mencoba-coba menu deposit dan penarikan. Uji coba memakan waktu lama karena juga harus menggunakan aplikasi Android untuk Top Up dan penarikan. Saya juga berusaha memahami alur data di *backend* menu Top Up maupun Penarikan.

Selasa, 12 Maret 2019

Hari ini saya berusaha membuat API untuk Top Up. API pertama yang saya buat adalah API untuk mendapatkan seluruh data Top Up. Seperti pada pekerjaan sebelumnya, untuk mendapatkan data yang dibutuhkan, diperlukan penggabungan beberapa tabel. API yang saya buat meliputi API untuk mengambil data Top Up yang belum diverifikasi, mengambil data Top Up yang sudah diverifikasi. Selain itu saya juga membuat API yang nantinya akan digunakan untuk melakukan search pada data Top Up, baik yang sudah diverifikasi maupun yang belum diverifikasi.

Rabu, 13 Maret 2019

Hari ini saya membuat tampilan untuk Top Up dan Penarikan. Dalam satu halaman tampilan Top Up yang saya buat, terdapat dua *tab* yang dapat dipilih. *Tab* pertama digunakan untuk melihat data Top Up yang belum diverifikasi, *tab* kedua digunakan untuk melihat data yang sudah diverifikasi. Pada bagian Top Up yang belum diverifikasi, terdapat tombol untuk verifikasi, yang mana bila ditekan akan mengarahkan ke halaman verifikasi untuk memverifikasi dan melihat status Top Up. Pada bagian Top Up yang telah diverifikasi, terdapat tombol untuk melihat detail Top Up yang telah diverifikasi.

Kamis, 14 Maret 2019

Hari ini saya membuat halaman tampilan dan *verify* untuk Top Up dan Penarikan. Halaman tampilan dibuat untuk melihat setiap Top Up dan Verifikasi yang telah dibuat secara mendetail. Halaman *verify* ini berfungsi sama seperti halaman tampilan Top Up – Penarikan, yang membedakan adalah terdapat tombol *verify*, *verify* ini akan memastikan bahwa Top Up dan Penarikan benar-benar ada dan valid. Apabila verifikasi telah berhasil, lalu pengguna akan diberitahukan melalui *push notification* yang dikirimkan ke gawai pengguna yang melakukan Top Up atau Penarikan.

Jumat, 15 Maret 2019

Hari ini saya mencoba API yang saya buat adalah API untuk Penarikan. Untuk API penarikan, yang dibuat sama seperti API Top Up kemarin, meliputi API untuk mendapatkan Penarikan yang belum diverifikasi, sudah diverifikasi, *search* Penarikan yang belum diverifikasi, dan *search* untuk Penarikan yang sudah diverifikasi. Selain itu saya juga mengimplementasikan kedua API yang sudah saya buat. Implementasi dilakukan dengan cara meletakkan *link* API ke DepositController. Pada DepositController saya juga membuat fungsi tampilan untuk masing-masing Top Up dan Penarikan.

Senin, 18 Maret 2019

Hari ini saya harus mempelajari JavaScript. JavaScript yang ingin saya buat memiliki konsep yang mirip dengan yang sebelumnya dimana ketika dropdown box mengalami perubahan, elemen tertentu yang saya inginkan akan mengikuti perubahan pada *dropdown box* nya. Perbedaanya adalah ketika saya mengubah *dropdown*, maka tabel yang menjadi target, akan terisi otomatis. Saya sudah menemukan cara untuk melakukan update tabel, namun ketika saya mencobanya, masih terdapat kegagalan. Akhirnya karena masih belum dapat diimplementasikan, saya mengembalikan tampilan menjadi seperti sebelumnya.

Selasa, 19 Maret 2019

Hari ini saya berdiskusi mengenai API create auction yang saya buat. Saya mendiskusikannya karena selama beberapa hari saya masih belum dapat menyelesaikan API create auction event. Setelah saya diskusikan dengan Bu Christie, ternyata API saya masih terdapat beberapa masalah terutama karena tidak mampu memenuhi persyaratan atau kondisi yang seharusnya. Beberapa tabel yang

saya tidak ketahui itu ada, tidak terupdate karena saya terpaksa melewati beberapa langkah karena adanya Error bila tidak saya lewati.

Rabu, 20 Maret 2019

Hari ini saya mencoba untuk melanjutkan *generate* nomor LOT. Saya melanjutkan *progress* saya beberapa waktu yang lalu. Saat ini saya mencoba menggunakan cara baru. Saya mencoba meniru cara *generate* nomor LOT pada Server Action. Pada akhirnya saya dapat melakukan *generate* nomor LOT dan dapat dimasukan dalam tabel Unit. Setelah itu saya coba untuk mengimplementasikan *generate* nomor LOT dalam create auction event saya. Meskipun belum sempurna, tapi *generate* nomor LOT sudah dapat berjalan, tinggal mencari kesalahan saya kenapa tabel History Unit belum bisa diupdate.

Kamis, 21 Maret 2019

Hari ini saya debugging auction event untuk memastikan tidak ada permasalahan lain, kecuali pada History Unit. Saya mencoba mengecek fungsionalitas pada search dan pagination. Pada bagian create auction event saya hanya mencoba untuk melihat apakah Unit dapat ditampilkan dan pada saat create saya mencoba apakah nomor LOT sudah berhasil ter-*generate*. Setelah itu saya melanjutkan mengerjakan deposit, dimana masih terdaapat kekurangan pada bagian verify.

Jumat, 22 Maret 2019

Hari ini saya memperbaiki auction event, terutama pada bagian pagination setelah search. Saya lupa untuk meletakkan pagination pada bagian search Perbaiki auction Event yang saya lakukan adalah nomor LOT. Setelah itu saya melanjutkan mengerjakan menu Deposit lagi. Kali ini saya mencoba melihat alur

dari penarikan, dimana terdapat reject. Saya melupakan fungsi reject pada penarikan. Saya juga mencoba membuat API untuk *reject* namun belum selesai.

Senin, 25 Maret 2019

Hari ini saya membuat API untuk cek nomor LOT. Nomor LOT sebelumnya digabungkan dengan proses auction, namun sekarang saya pisah karena proses API tidak dapat melakukan generate seperti pada *back-end* OutSystem Web. Saya berhasil membuat API yang dapat mengecek apakah nomor LOT tertentu telah digunakan atau belum. Caranya adalah dengan menggunakan *query* untuk mencari data nomor LOT yang sama dengan input. Saya mulai merasa badan saya kurang sehat, maka saya memutuskan untuk ijin pulang lebih awal.

Selasa, 26 Maret 2019

Hari Selasa saya tidak berangkat ke kantor, saya mengerjakan dari kos. Karena sedang dalam kondisi sakit, saya tidak dapat mengerjakan banyak hal. Saya hanya melanjutkan membuat API untuk *reject* Penarikan. Hingga akhir hari ini, saya hanya dapat menyelesaikan 40% dari API *reject*.

Rabu, 27 Maret 2019

Hari ini saya memutuskan untuk periksa ke dokter lalu check darah dan saya ternyata terkena Demam Berdarah. Saya dirawat di Rumah Sakit Bethesda Yogyakarta dari tanggal 27 Maret 2019 hingga tanggal 2 April 2019 (surat keterangan terlampir). Setelah keluar dari Rumah Sakit, saya diharuskan beristirahat di rumah hingga tanggal 5 April agar kondisi tubuh saya dapat membaik.

2.2 Hasil Pekerjaan Secara Umum

Berdasarkan magang yang telah saya lakukan, hasil yang saya kerjakan adalah sebuah Content Management System (CMS) untuk mengelola ACC BidMart. Sebagai salah satu perusahaan yang bekerja di bidang bisnis otomotif, terutama perkreditan. Dalam proses kredit, tidak semua kredit lancar hingga lunas. Untuk kendaraan yang tidak lunas kreditnya, ACC menjual ulang mobil atau kendaraan yang telah mereka tarik. ACC BidMart sendiri merupakan aplikasi berbasis Android, yang dapat digunakan untuk menghubungkan pembeli mobil dengan Balai Lelang yang telah bekerja sama dengan ACC. Dengan adanya CMS, pihak yang berkepentingan dapat memantau, melakukan *editing* dan mengatur aplikasi Android melalui CMS.

ACC sudah memiliki CMS yang sudah ada dan masih dalam tahap *development* di OutSystem, namun mereka ingin mencoba apakah mungkin API dibuat di OutSystem namun aplikasi webnya menggunakan *framework* Laravel. Kegunaan dari CMS BidMart disini adalah;

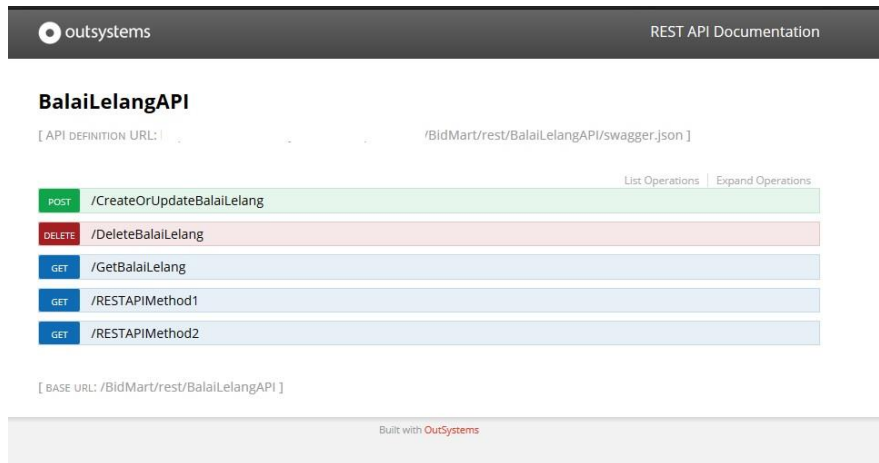
1. Memberikan informasi mengenai transaksi pembelian dan informasi penjualan
2. Mengatur konten-konten yang ada, seperti unit yang akan dijual, event lelang yang akan datang, dan hal yang berkaitan dengan itu.
3. Mengatur pengguna (admin dan pengguna mobile)
4. Mengatur event *auction*, beserta menampilkan hasil *auction*.
5. Menyimpan data akun bank untuk keperluan transaksi.
6. Deposit (Top Up dan Penarikan) untuk kepentingan ‘tanda jadi’ apabila memenangkan pelelangan, maupun menarik uang yang ada.
7. Manajemen konten yang akan ditampilkan pada aplikasi untuk menambah promo, push notification, rekomendasi, dan sebagainya.
8. Melihat transaksi yang telah dilakukan.

CMS tersebut dikembangkan dengan menggunakan bahasa pemrograman HTML dengan menggunakan *framework* Laravel. Untuk API, dibuat

menggunakan aplikasi OutSystem. Dengan demikian, apabila suatu saat dibutuhkan, *developer* hanya perlu memanggil alamat API.

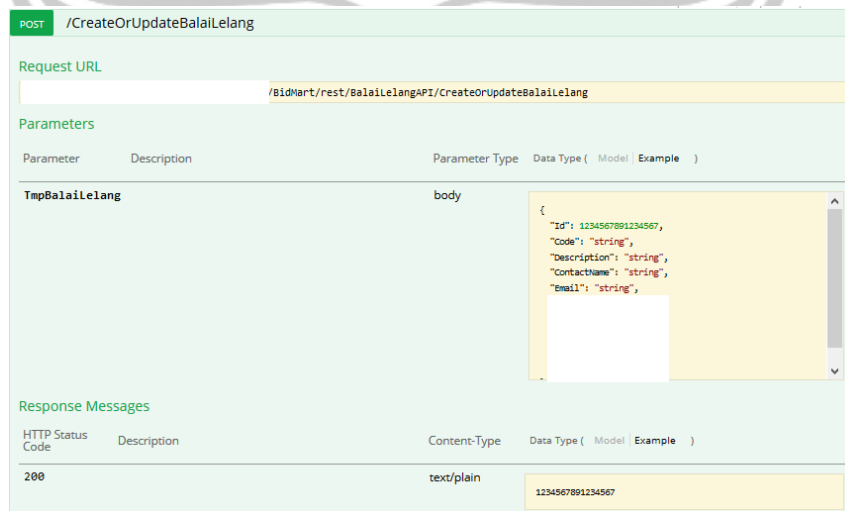


2.3 Bukti Hasil Pekerjaan



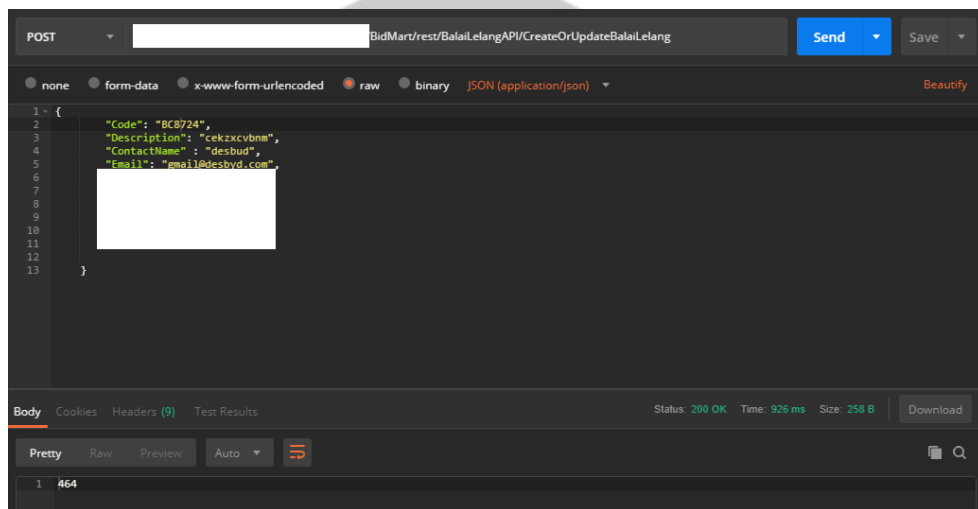
Gambar 2.13 Salah satu dokumentasi API

Saya membuat banyak API, baik untuk GET, POST, maupun DELETE. API tersebut digunakan untuk menu dan sub menu CMS. Total saya mengerjakan 4 submenu dan 1 menu. 4 Submenu terdiri dari: Balai Lelang (Master Management), Unit (Master Management), Top Up (Deposit), Penarikan (Deposit), dan 1 menu (Auction Event).

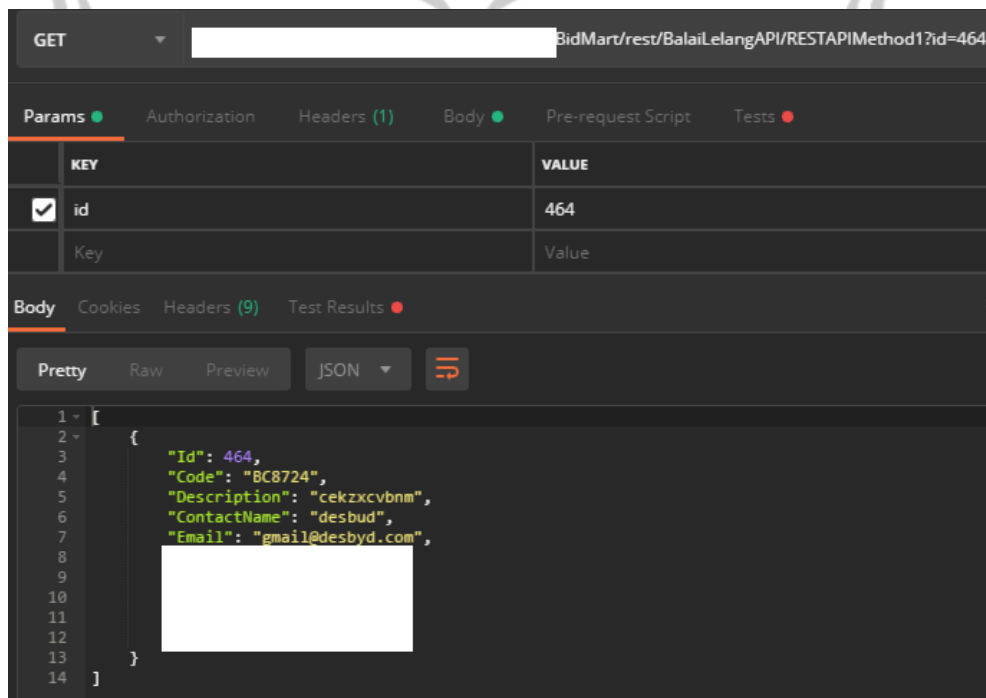


Gambar 2.14 Salah satu dokumentasi API POST

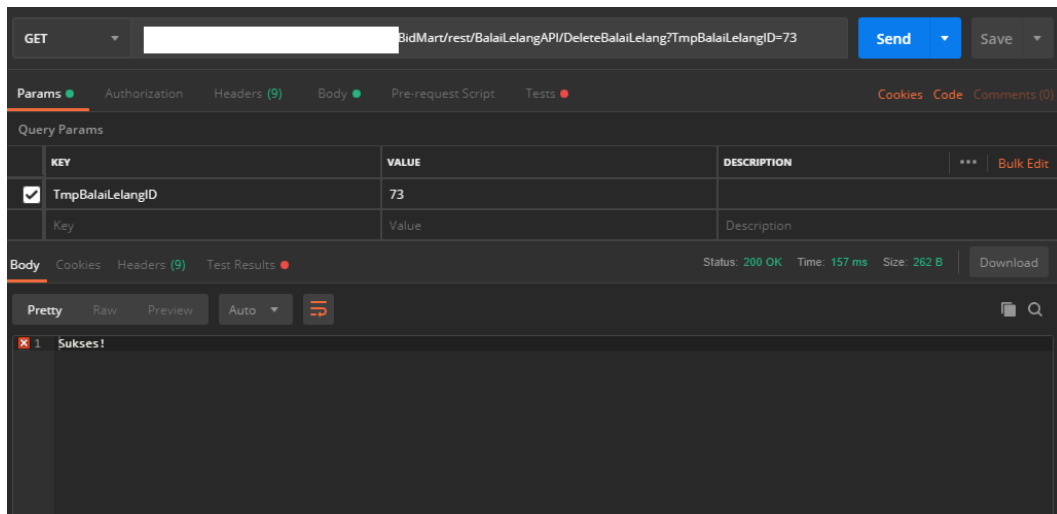
API yang saya buat apabila diakses, akan memiliki tampilan seperti pada gambar diatas. API POST tersebut apabila ingin digunakan harus menggunakan Postman atau melalui *packages* Guzzle (bila menggunakan Laravel). Berikut ini saya akan melampirkan seluruh pekerjaan saya, mulai dari pembuatan API hingga halaman web. URL dan beberapa bagian yang *confidential* sesuai kesepakatan, saya tutupi dengan kotak putih guna menjaga kerahasiaan data *core* ACC.



Gambar 2.15 Hasil POST API di Postman

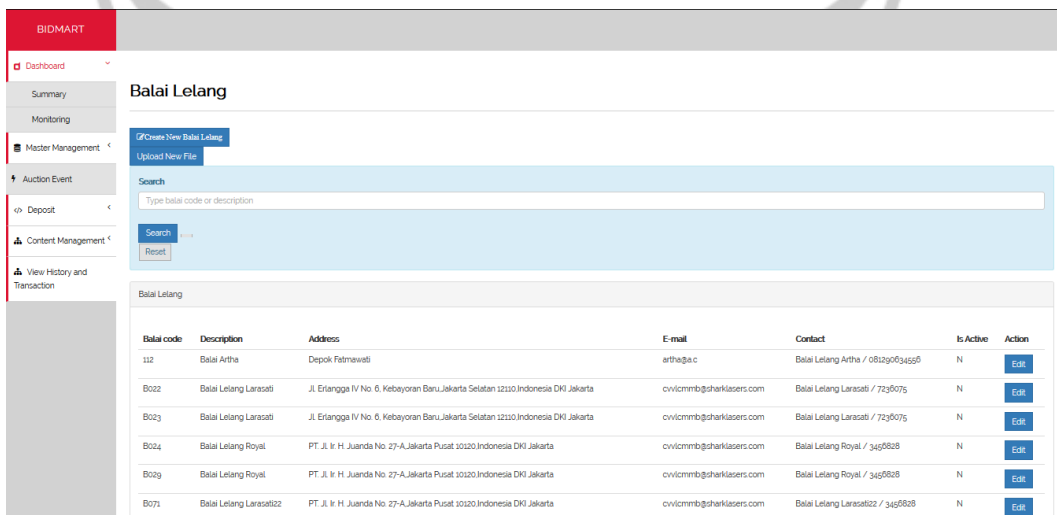


Gambar 2.16 Hasil GET berdasarkan ID tertentu



Gambar 2.17 Hasil DELETE berdasarkan ID tertentu

Pada Gambar diatas, apabila berhasil dilakukan POST dan data yang dimasukan memenuhi syarat (seperti jumlah karakter, dll) maka akan ditampilkan ID Balai Lelang baru tersebut. Apabila dicek ID tersebut dengan menggunakan API yang telah dibuat, maka akan ditampilkan data-data yang tadi telah dimasukan. Jika ingin menghapus, maka dapat digunakan API DELETE dengan input ID Balai Lelang.



Gambar 2.18 Tampilan view data Balai Lelang + pagination

Content Management * Deskripsi dari Balai Lelang

View History and Transaction

Contact Name:
Nama

Address:
Alamat dari Balai Lelang

Mail Number:
Nomor Mail

E-MAIL:
E-mail Balai Lelang

Phone Number:
Nomor telepon Balai Lelang

Facsimile Number:
Nomor Faksimile

Is Active?
 Yes
 No

Choose image from directory
 No file selected

Gambar 2.19 Tampilan *create* data Balai Lelang

View History and Transaction

Address:
gbswoodjndngsd

Mail Number:
147394

E-MAIL:
bugh@gmail.com

Phone Number:
0208777122

Facsimile Number:
1398230

Name:
bualangi

Is Active?
 Y
 N

Picture ID:

Gambar 2.20 Tampilan *update* data Balai Lelang

BIDMART

Dashboard

Master Management

Auction Event

Deposit

Content Management

View History and Transaction

Balai Lelang

Upload New File

Search: balai

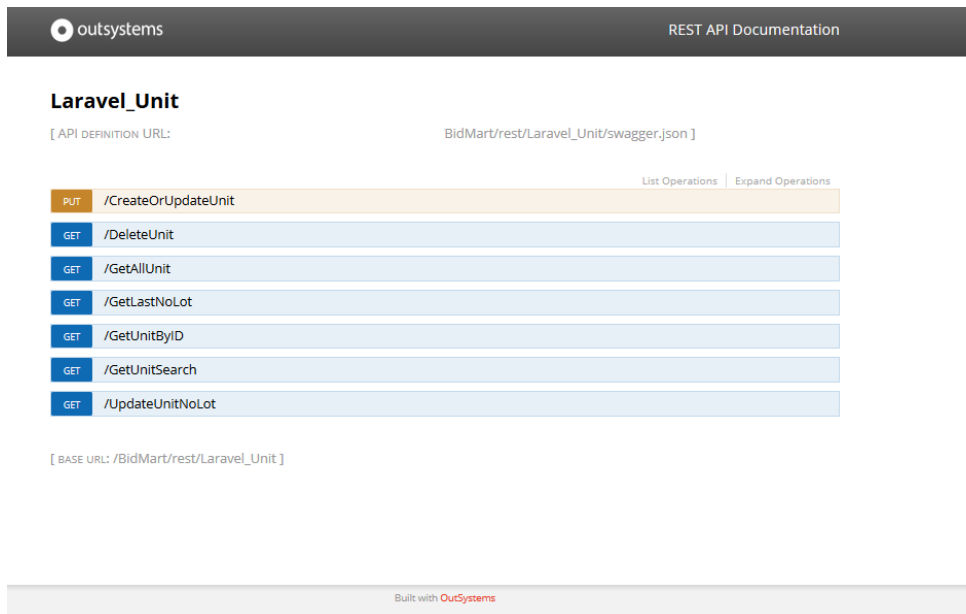
Search

Reset

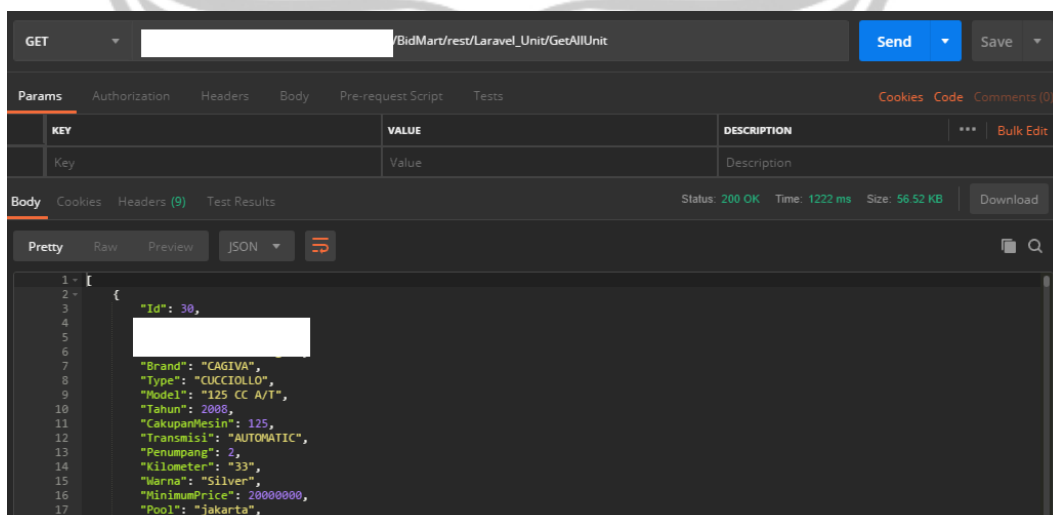
Balai code	Description	Address	E-mail	Contact	Is Active	Action
balai	balai lelang 123	jakarta	balai.lelang@gmail.com	balai lelang / 02174521	N	Edit

Gambar 2.21 Tampilan *search* data Balai Lelang + pagination

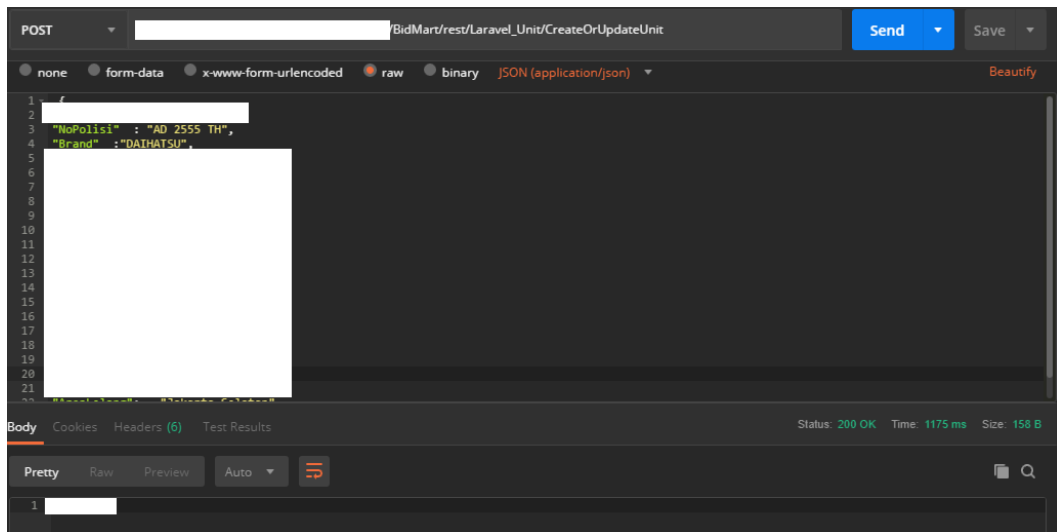
Pada gambar diatas memperlihatkan penggunaan dari API yang telah saya buat. API tersebut digunakan untuk memberikan *supply* data untuk aplikasi web yang saya buat. API tersebut juga dapat digunakan untuk membuat data, update, dan delete. Untuk CRUDS Balai Lelang sendiri, digunakan untuk mengatur data-data Balai Lelang yang bekerja sama dengan ACC.



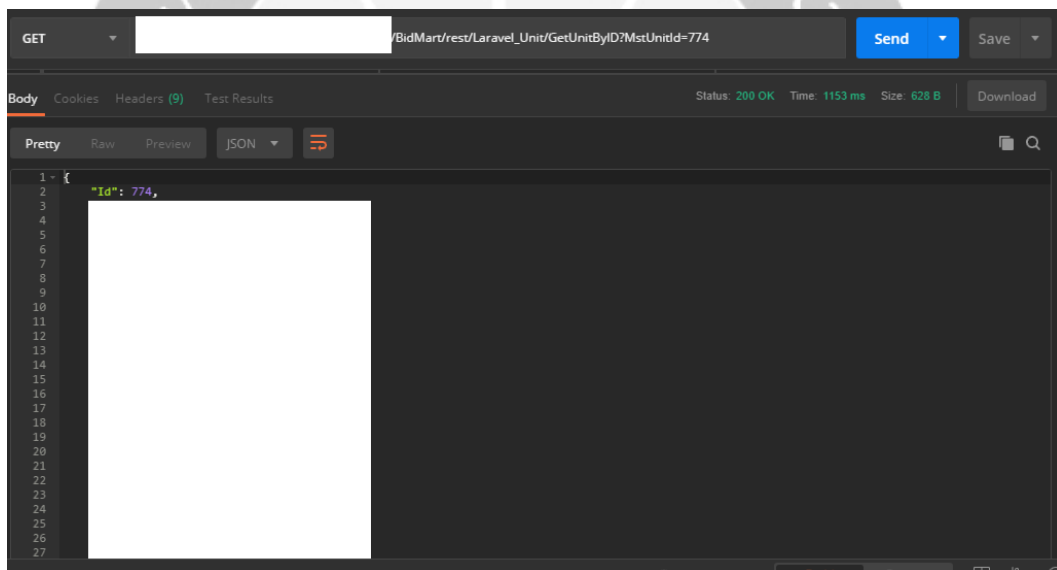
Gambar 2.22 Dokumentasi API Unit



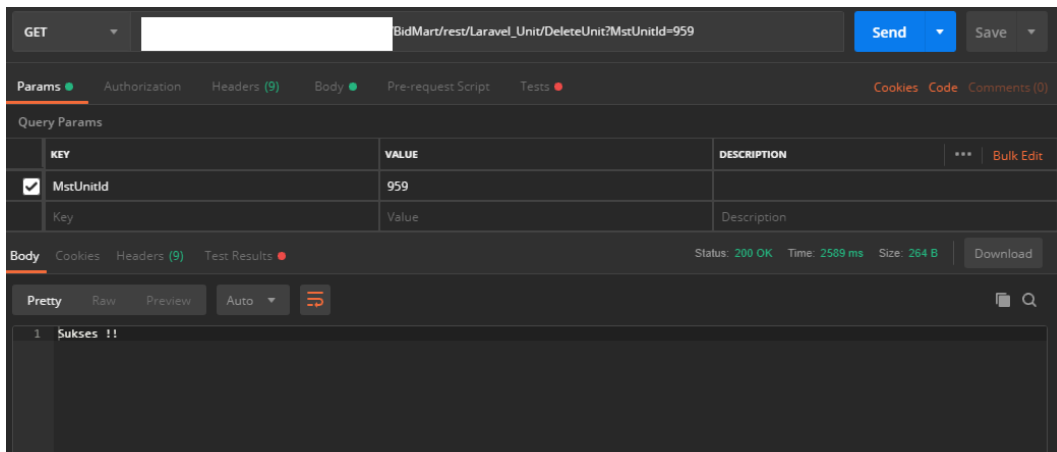
Gambar 2.23 Hasil GET data Unit



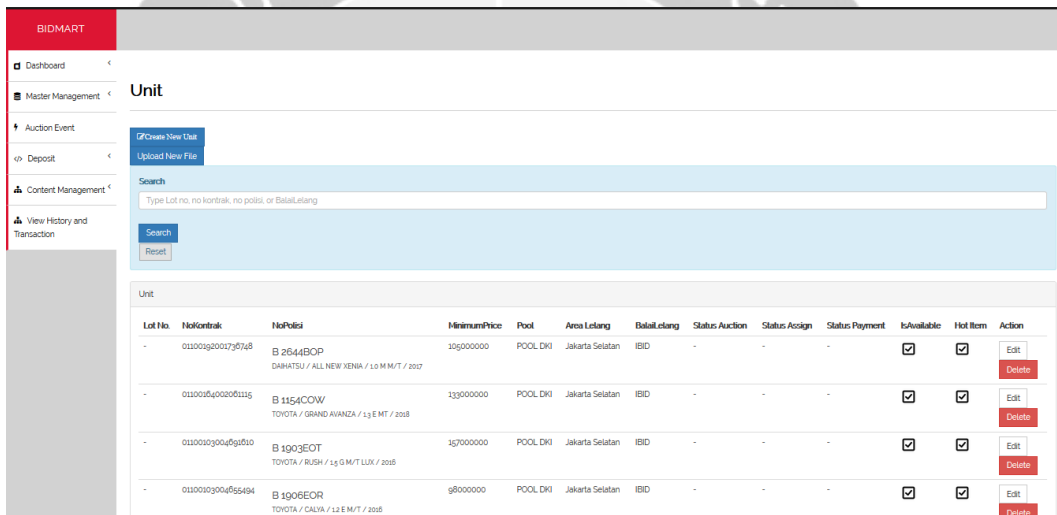
Gambar 2.24 Hasil POST Unit



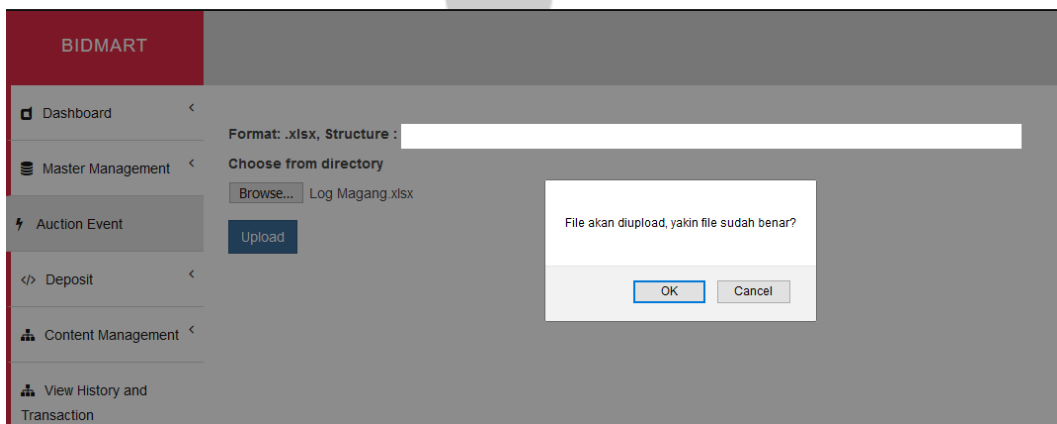
Gambar 2.25 Hasil GET berdasarkan ID tertentu



Gambar 2.26 Hasil DELETE Unit



Gambar 2.27 Tampilan view Unit



Gambar 2.28 Upload file .xlsx

BIDMART

- Dashboard
- Master Management
- Auction Event
- Deposit
- Content Management
- View History and Transaction

Create Unit

Nomor Kontrak

Nomor BPKB

Nomor Polisi Kendaraan

Kode Brand

Brand

Kode Type

Type

Kode Model

Gambar 2.29 Tampilan create data Unit

BIDMART

- Dashboard
- Master Management
- Auction Event
- Deposit
- Content Management
- View History and Transaction

Edit Unit

Nomor Kontrak

Nomor BPKB

Nomor Polisi Kendaraan

Kode Brand

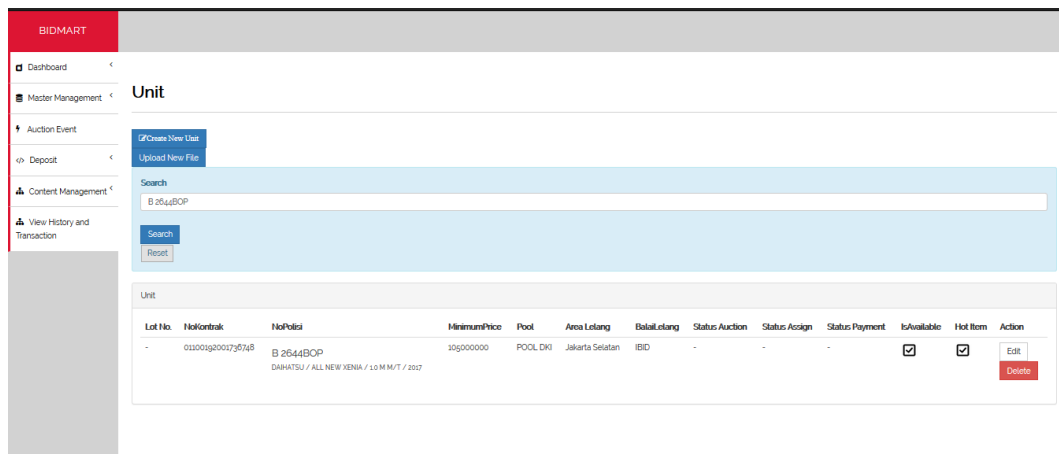
Brand

Kode Type

Type

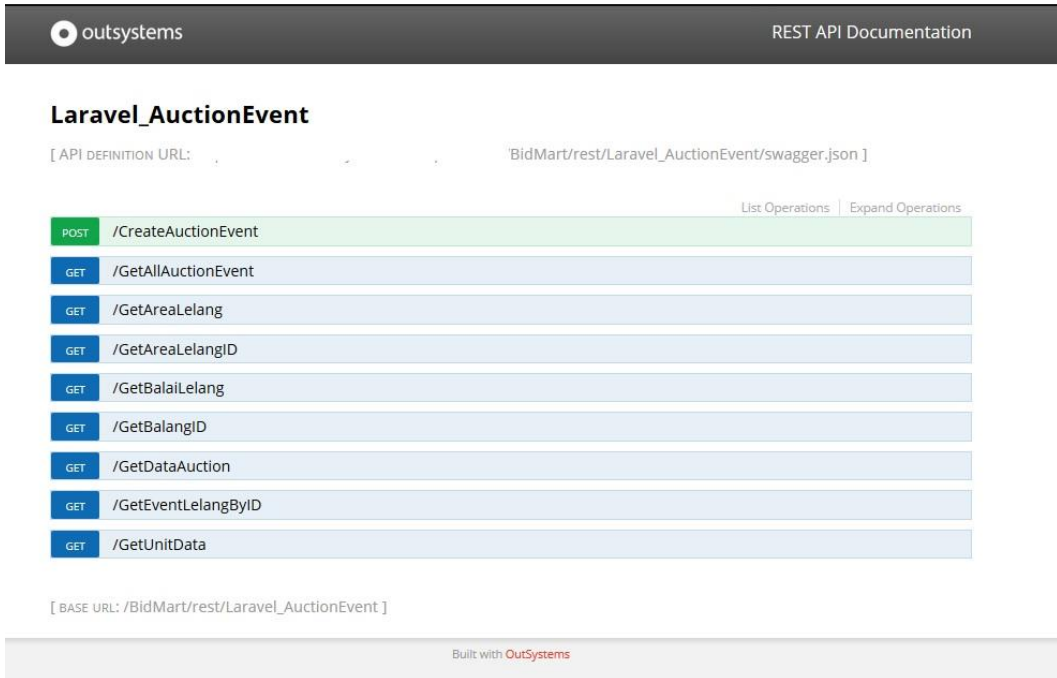
Kode Model

Gambar 2.30 Tampilan update data Unit

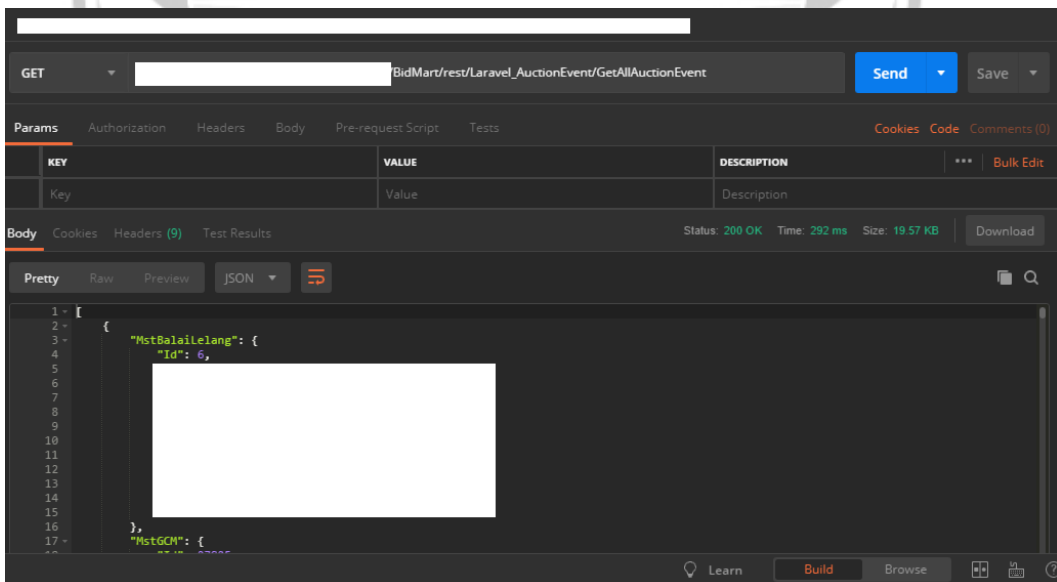


Gambar 2.31 Tampilan search data Unit

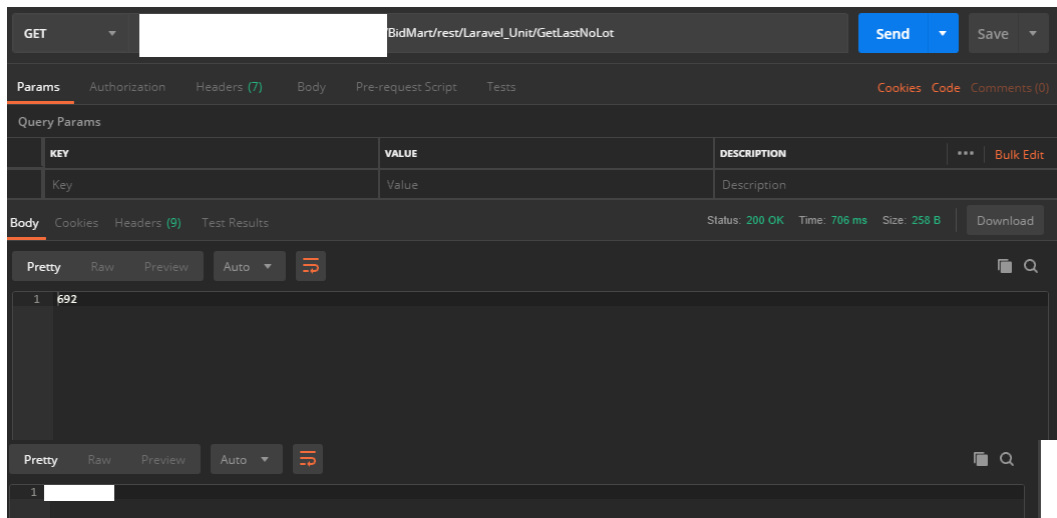
Pada sub-menu Unit ini API yang saya buat telah bisa diimplementasikan dengan baik. Kendala yang masih dihadapi adalah karena saya tidak mempunyai akses untuk melihat cabang aktif pada area lelang, sehingga ketika ada input area lelang yang tidak ada ditabel, akan bermasalah. Masalahnya adalah karena area lelang akan *diconvert* menjadi area lelang ID, yang berhubungan dengan tabel lain. Hal itu sudah saya antisipasi, yaitu dengan mereturnkan nilai 0 ke area lelang. Saya juga menambahkan fungsionalitas untuk upload berkas bertipe .xlsx, dimana jika format dan syarat kolom terpenuhi, maka data-data yang ada dalam berkas tersebut akan dilakukan *POST*. Ada bagian dimana gambar belum dapat ditampilkan. Ini berkaitan dengan format gambar base64 yang digunakan ACC. Saya masih belum menguasainya, sehingga saya diijinkan untuk melewati fitur yang berkaitan dengan gambar.



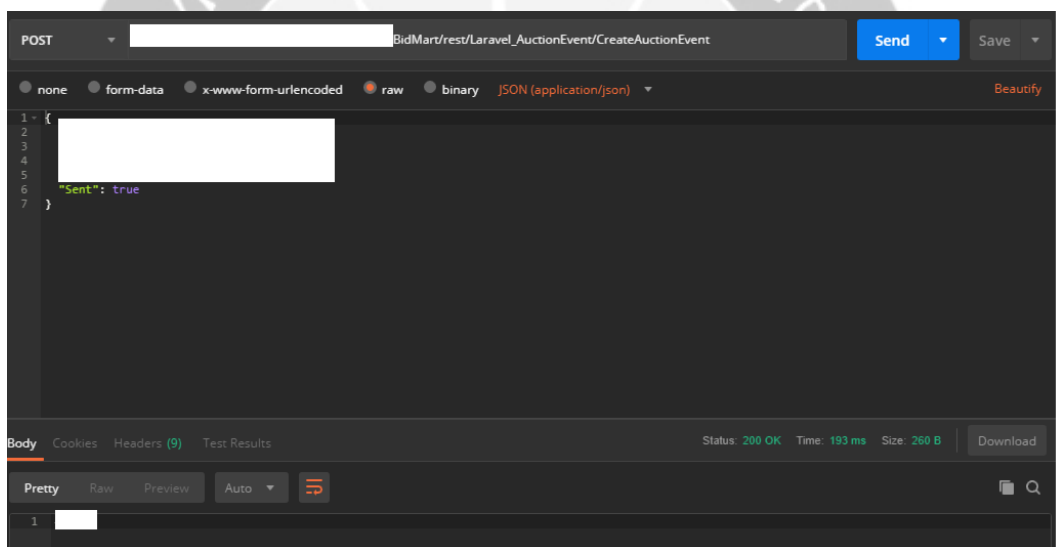
Gambar 2.32 Dokumentasi API Auction Event



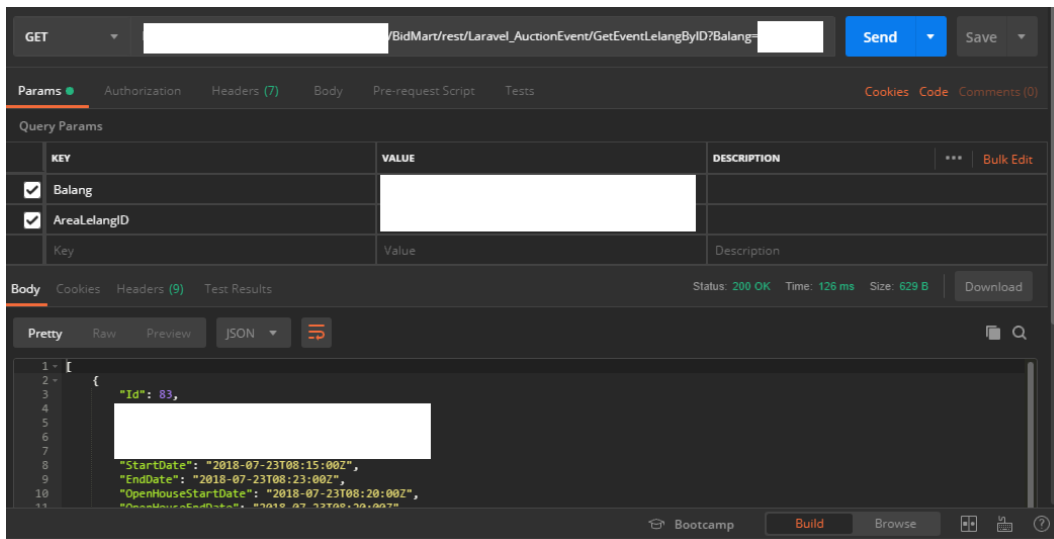
Gambar 2.33 Hasil GET Auction Event



Gambar 2.34 Hasil generate nomor LOT



Gambar 2.35 Hasil POST Auction Event



Gambar 2.36 Hasil GET berdasarkan ID tertentu

The screenshot shows the 'Auction Event' view in a web application. It includes a search bar and a table listing various auction events. The table has the following columns: EVENT NAME, LOT NO, NO KONTRAK, NO POLISI, MINIMUM PRICE, AREA, BALAI LELENG, OPEN HOUSE DATE, and EVENT.

EVENT NAME	LOT NO	NO KONTRAK	NO POLISI	MINIMUM PRICE	AREA	BALAI LELENG	OPEN HOUSE DATE	EVENT
CENTRALASIAVA	-	6666667	B6 401 TYN PATRIA KOMATSU / DIESEL GENSET / PL 250 / 2000	Rp 200000000	Jakarta Barat	Central Asia centralasia@gmail.com	15-Jul-2018 17:00:00 to 18-Jul-2018 17:00:00	15-Jul-2018 20:14:00 to 18-Jul-2018 17:00:00
Rolls royce fair	-	33333	B 1706 LON AMMAN / VIBRATING COMP / ASC100-RADFOOT / 2017	Rp 200000000	Jakarta Selatan	Central Asia centralasia@gmail.com	13-Jul-2018 03:05:00 to 13-Jul-2018 03:05:00	13-Jul-2018 03:04:00 to 14-Jul-2018 03:04:00
Mitsubishi Fair	-	12312	POPO019 MITSUBISHI / DELICA / 2.0 A/T / 2009	Rp 100000000	Jakarta Selatan	Trimita trimita@gmail.com	12-Jul-2018 10:28:00 to 20-Jul-2018 10:28:00	12-Jul-2018 10:28:00 to 20-Jul-2018 10:28:00
Rolls royce fair	-	12345	T123P DAEWOO / ESPERO / CD AUTOMATIC / 2012	Rp 120000000	Jakarta Selatan	Central Asia centralasia@gmail.com	13-Jul-2018 03:05:00 to 13-Jul-2018 03:05:00	13-Jul-2018 03:04:00 to 14-Jul-2018 03:04:00
Event Test Vero	-	34567890123	B 1550 ABC FERRARI / CALIFORNIA / LUXURY / 1998	Rp 100000000	Jakarta Selatan	IBID ibidjakarta@gmail.com	24-Jul-2018 04:15:00 to 28-Jul-2018 04:15:00	24-Jul-2018 04:15:00 to 28-Jul-2018 04:15:00
top star	-	5136514795898	qwerty2 CHERY / TIGGO / 2.0 SX / 2000	Rp 100000000	Jakarta Barat	star auction star@e.c	24-Jul-2018 11:45:10 to 24-Jul-2018 12:45:10	24-Jul-2018 11:45:10 to 24-Jul-2018 12:45:10
top star	-	5439097070001	OWERTY1 GREAT WALL / WINGLER / DC / 2018	Rp 100000000	Jakarta Barat	star auction star@e.c	24-Jul-2018 11:45:10 to 24-Jul-2018 12:45:10	24-Jul-2018 11:45:10 to 24-Jul-2018 12:45:10

Gambar 2.37 Tampilan view Auction Event

Create New Auction

Area Lelang: SEMARANG

Balai Lelang: Atria

Available unit

Action	Lot No	No Kontrak	No Polisi	Minimum Price	Pool	Area Lelang	Balai Lelang
<input type="checkbox"/>	-	82144	AD821JK / YAMAHA / YZF-R6 / 600 CC / 1245	500000000	POOL SOLO	SEMARANG	Atria

AuctionEvent: -- select an option --

Open House Date:

Start Date:

End Date:

Event Date:

Start Date:

End Date:

Simpan

Cancel

Gambar 2.38 Tampilan create Auction Event

BIDMART

Auction Event

Create New Auction

Search: honda

Search

Reset

Auction Event

EVENT NAME	LOT NO	NO KONTRAK	NO POLISI	MINIMUM PRICE	AREA	BALAI LELANG	OPEN HOUSE DATE	EVENT
testreturn	-	123456	B 1234 ABC HONDA / HR-V / 1.5 S M/T / 2018	Rp 150000000	DKI	JBA Indonesia jbaIndonesia@gmail.com	02-Aug-2018 06:15:17 to 02-Aug-2018 06:17:17	02-Aug-2018 06:14:17 to 02-Aug-2018 06:18:17

Gambar 2.39 Tampilan search data Auction Event

Menu Auction Event ini adalah menu untuk melakukan *assign* Unit ke Online Event. Setiap Unit yang akan dilelang harus dilakukan *assign* ke salah satu Online Event, untuk mendapatkan nomor LOT. Setelah didapatkan nomor LOT dan dilakukan Online Event, baru Unit tersebut dapat dilelang. Pada API dan web yang saya buat, sudah dilengkapi fungsi untuk melakukan fungsi CRUDS sederhana.

Laravel_Deposit

[API DEFINITION URL:

BidMart/rest/Laravel_Deposit/swagger.json]

List Operations | Expand Operations

GET	/GetPenarikanCompleted
GET	/GetPenarikanWaiting
GET	/GetTopUpCompleted
GET	/GetTopUpWaiting
GET	/SearchPenarikanCompleted
GET	/SearchPenarikanWaiting
GET	/SearchTopUpCompleted
GET	/SearchTopUpWaiting
GET	/VerifyPenarikan
GET	/VerifyTopUp
GET	/ViewSpecificPenarikanCompleted
GET	/ViewSpecificPenarikanWaiting
GET	/ViewSpecificTopUpCompleted
GET	/ViewSpecificTopUpWaiting

Gambar 2.40 Dokumentasi API Top Up

GET BidMart/rest/Laravel_Deposit/GetTopUpWaiting

Send Save

Params Authorization Headers (7) Body Pre-request Script Tests Cookies Code Comments (0)

Query Params

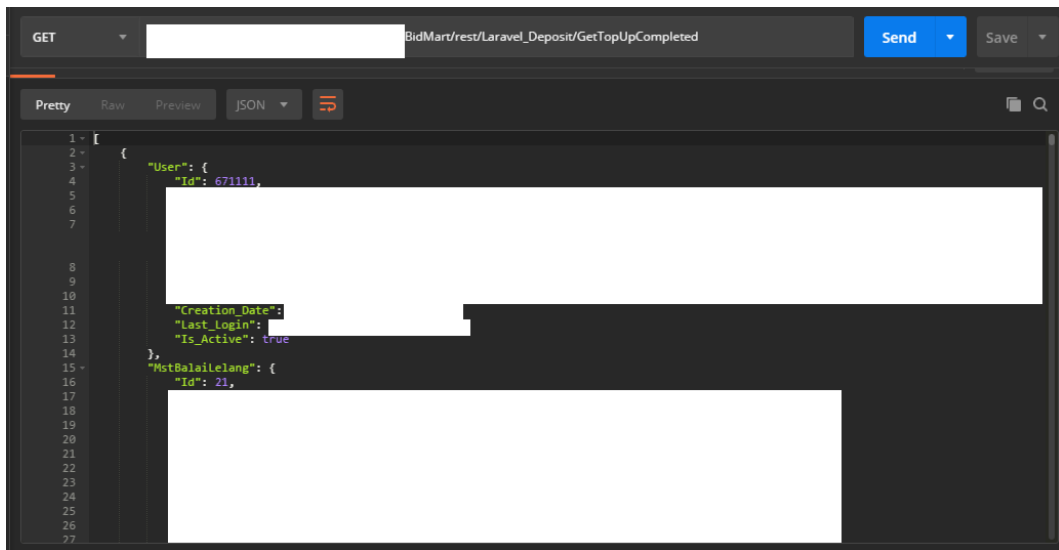
KEY	VALUE	DESCRIPTION	...	Bulk Edit
Key	Value	Description		

Body Cookies Headers (9) Test Results Status: 200 OK Time: 282 ms Size: 8.49 KB Download

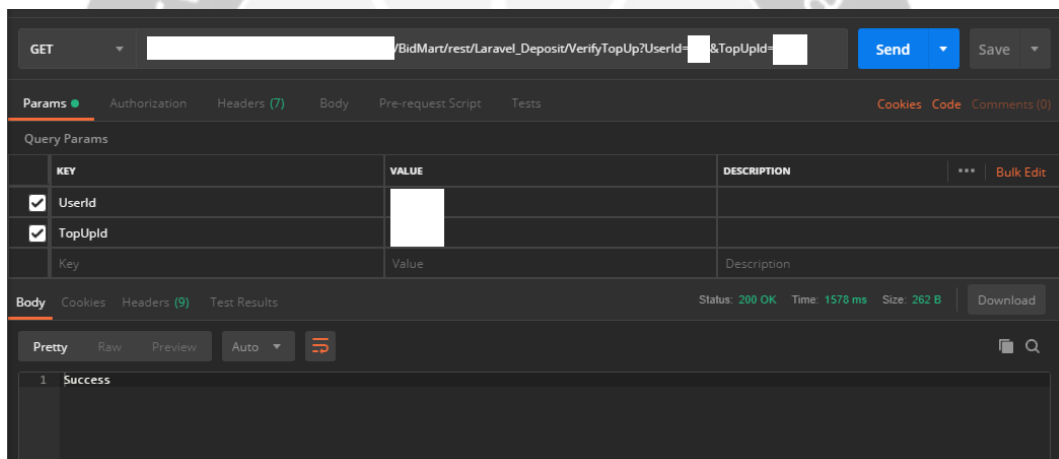
Pretty Raw Preview JSON

```
1 [
2   {
3     "User": {
4       "Id": 851,
5       "Creation Date":
6       "Last Login":
7       "Is Active": true
8     }
9   }
10
11
12
13
```

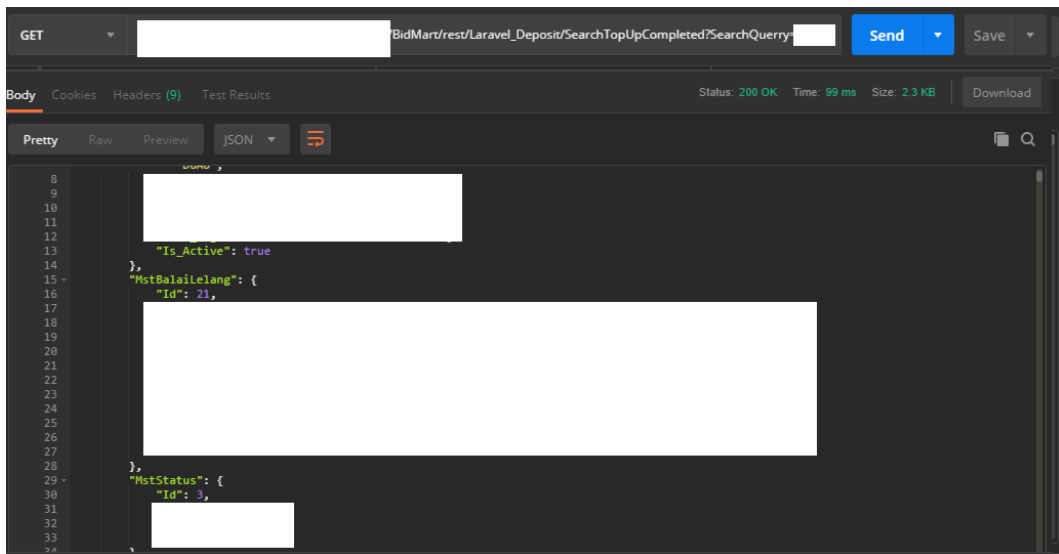
Gambar 2.41 Hasil GET data Top Up *unverified*



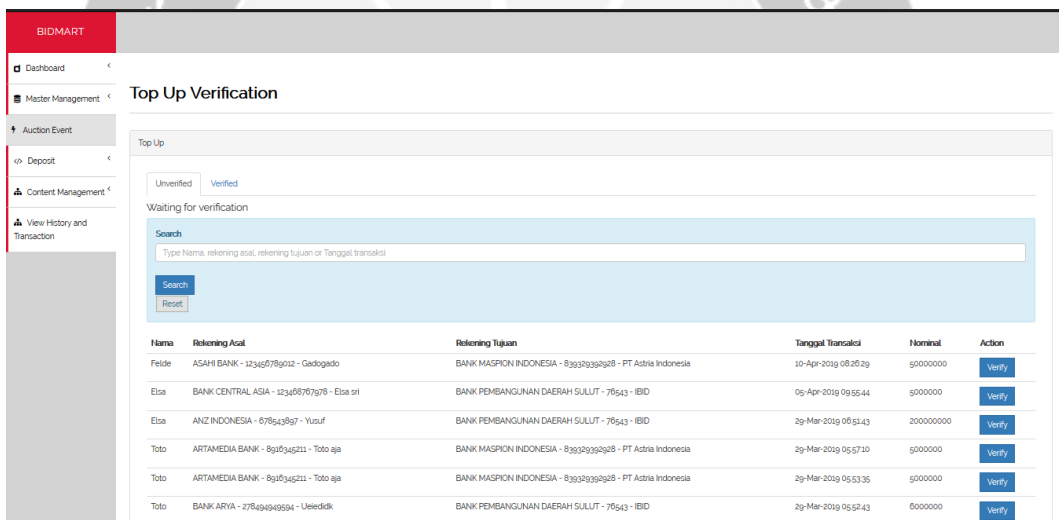
Gambar 2.42 Hasil GET data Top Up verified



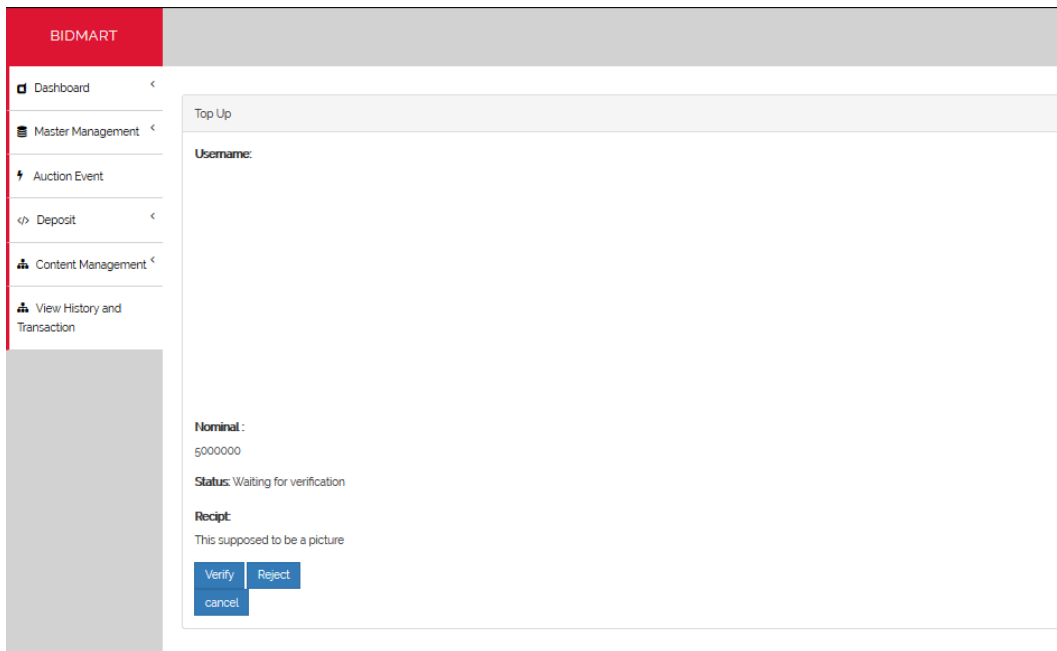
Gambar 2.43 Hasil verify Top Up



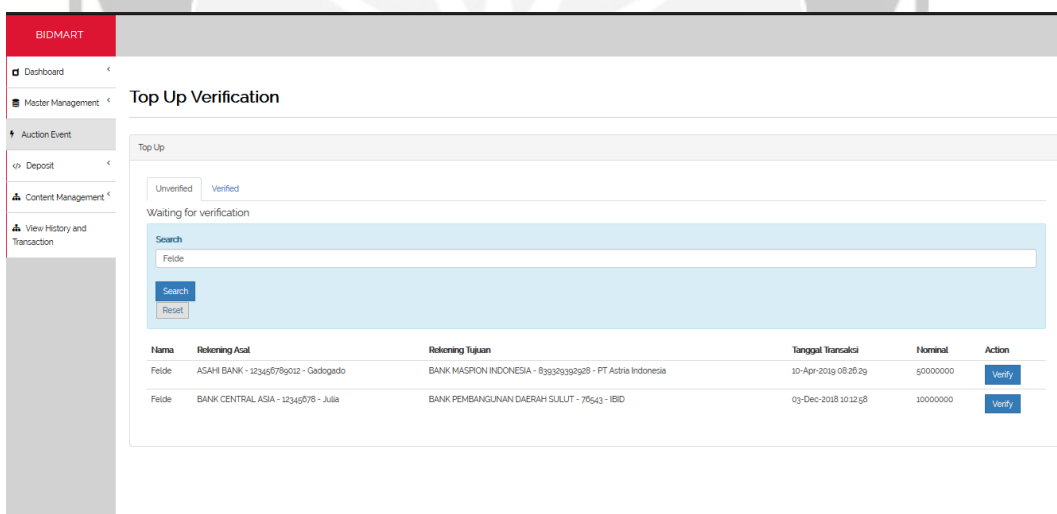
Gambar 2.44 Hasil GET berdasarkan ID tertentu



Gambar 2.45 Tampilan view Top Up



Gambar 2.46 Tampilan *verify* Top Up



Gambar 2.47 Tampilan *search* data Top Up

Menu Deposit disini adalah menu yang sangat krusial. Setiap pengguna yang ingin melakukan transaksi (melakukan *bidding*) secara online, wajib melakukan Top Up atau isi saldo. Saldo yang diisikan ini dapat ditarik kembali. Saldo yang diisikan ini berguna agar pengguna tidak ‘menghilang’ setelah memenangkan lelang. Top Up dilakukan melalui aplikasi Android, dimana

pengguna memasukan jumlah transfer dan untuk proses verifikasi Top Up itu sendiri dilakukan secara manual. Saya telah membuat API dan tampilan CRUDS untuk mendukung proses bisnis ini. Tidak banyak data yang saya bisa tampilkan, karena sekali lagi, data-data disini berhubungan dengan *core* ACC.

Laravel_Deposit

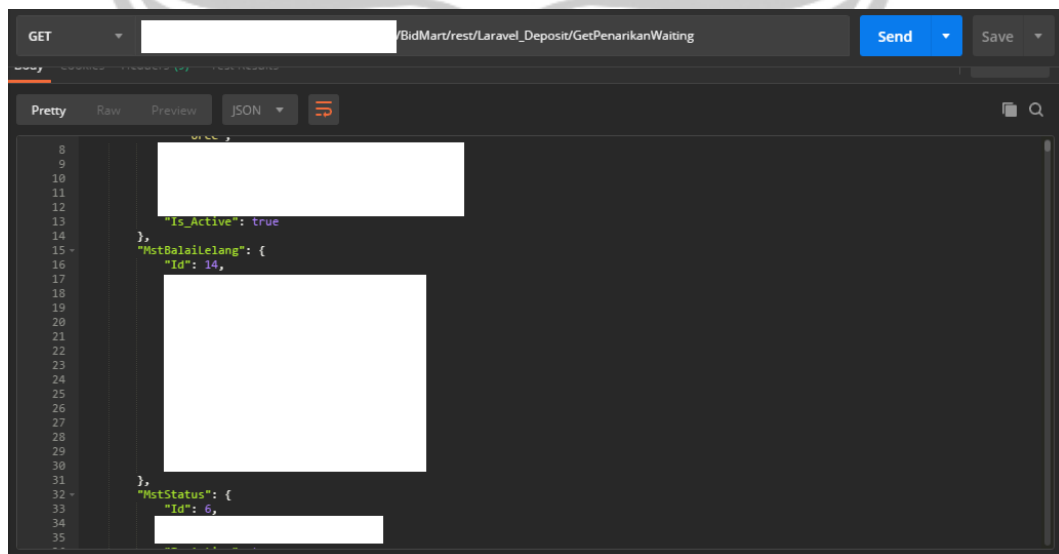
[API DEFINITION URL:

BidMart/rest/Laravel_Deposit/swagger.json]

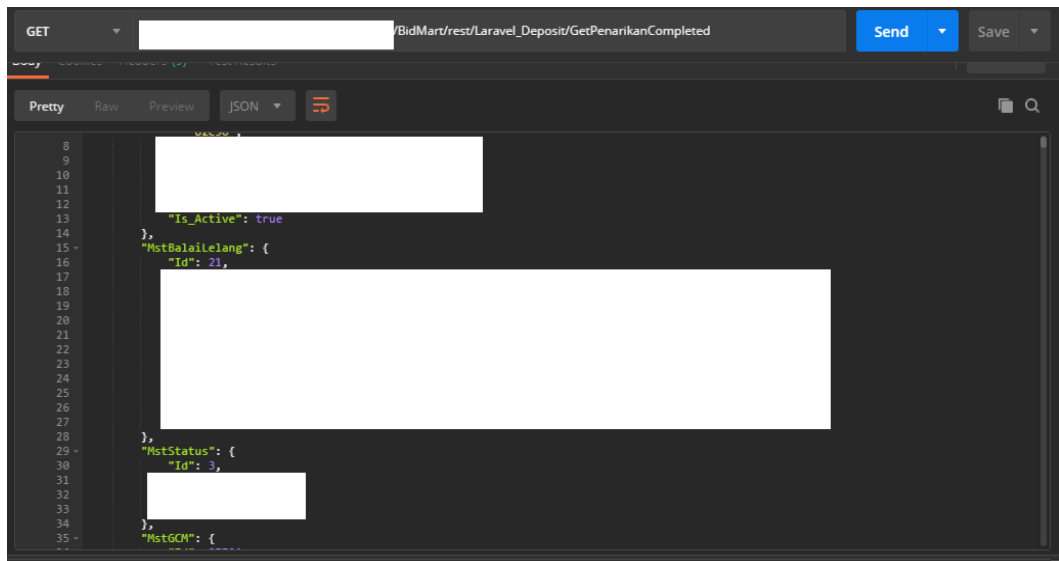
List Operations | Expand Operations

GET	/GetPenarikanCompleted
GET	/GetPenarikanWaiting
GET	/GetTopUpCompleted
GET	/GetTopUpWaiting
GET	/SearchPenarikanCompleted
GET	/SearchPenarikanWaiting
GET	/SearchTopUpCompleted
GET	/SearchTopUpWaiting
GET	/VerifyPenarikan
GET	/VerifyTopUp
GET	/ViewSpecificPenarikanCompleted
GET	/ViewSpecificPenarikanWaiting
GET	/ViewSpecificTopUpCompleted
GET	/ViewSpecificTopUpWaiting

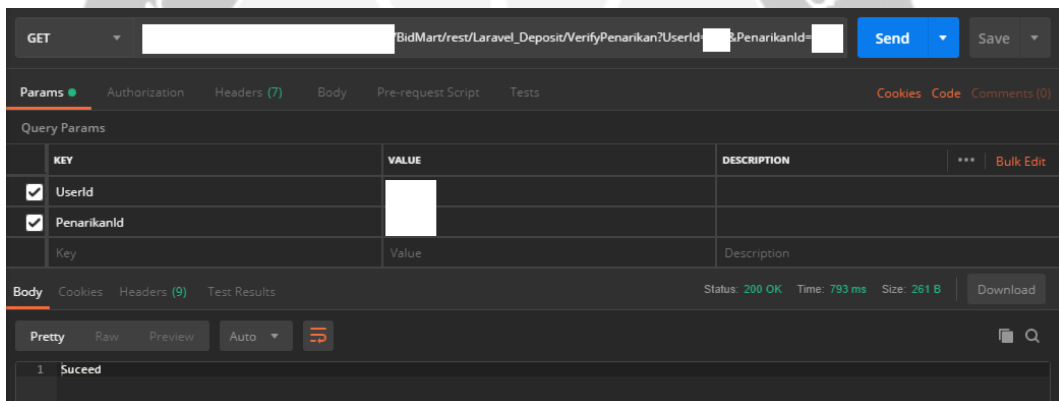
Gambar 2.48 Dokumentasi API Penarikan



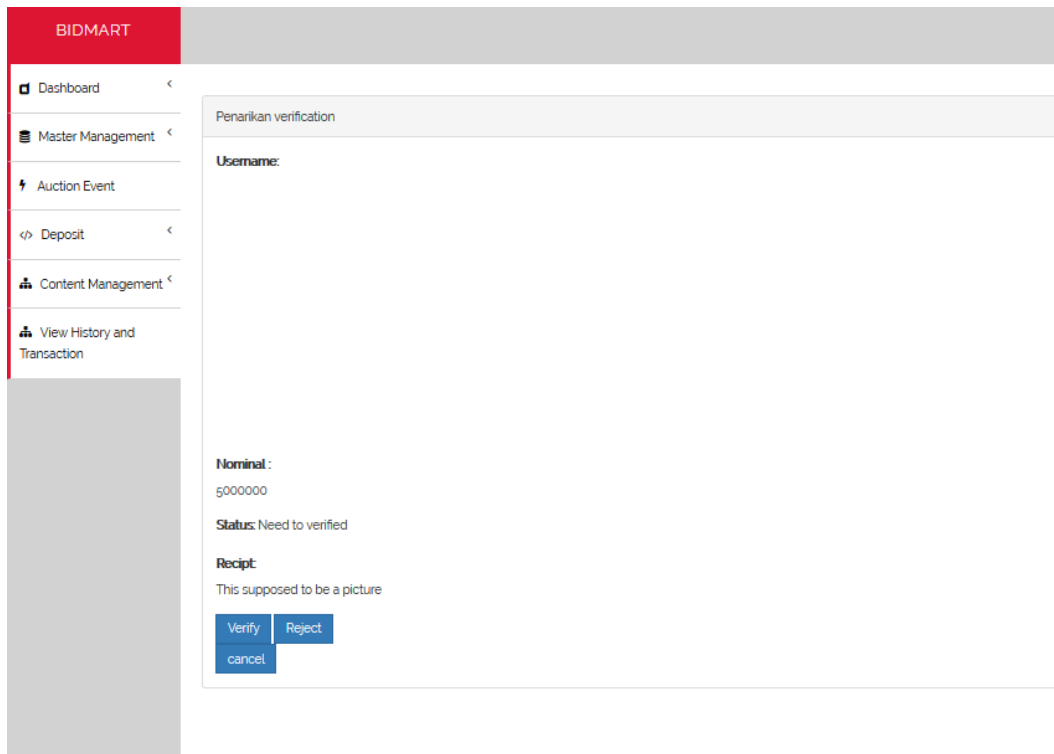
Gambar 2.49 Hasil GET Penarikan *unverified*



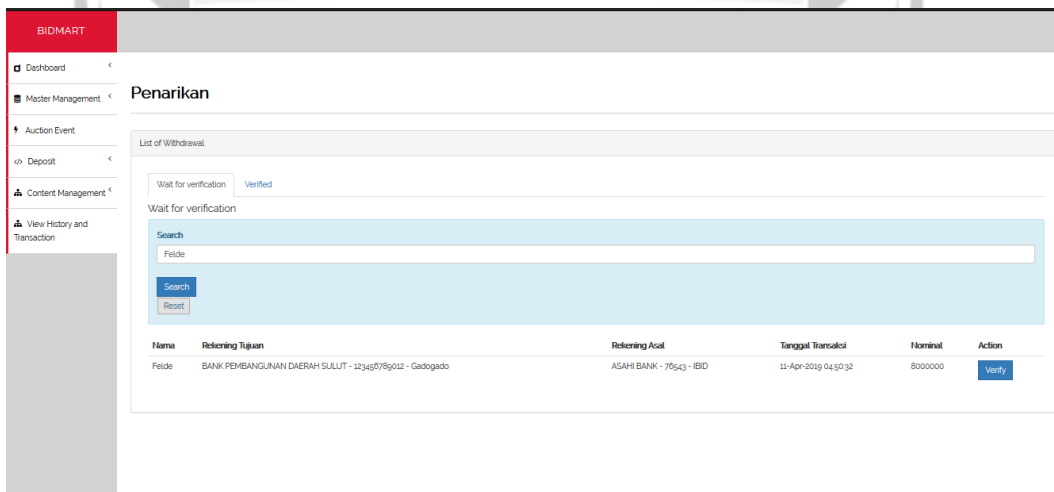
Gambar 2.50 Hasil GET Penarikan verified



Gambar 2.51 Hasil verify Penarikan



Gambar 2.54 Tampilan *verify* Penarikan



Gambar 2.55 Tampilan *search* Penarikan

Sub menu penarikan ini adalah bagian dari menu Deposit. Jika sebelumnya Top Up memastikan pengguna menyimpan ‘deposit’ uang didalam sistem, maka fungsi dari menu penarikan ini adalah untuk menarik uang yang telah disimpan di sistem. Uang yang akan ditarik oleh pengguna, terlebih dahulu harus melewati

proses verifikasi. Apabila penarikan yang dilakukan disetujui, maupun tidak disetujui, maka pengguna akan diberitahu melalui *push notification* pada gawainya. Proses verifikasi inilah yang saya buat, sistem CMS yang melakukan CRUDS pada penarikan, meskipun lebih tepatnya bukan *create* maupun *update* namun verifikasi.



BAB III

HASIL PEMBELAJARAN

3.1 Manfaat Magang

Magang memberikan sangat banyak manfaat berharga. Pertama, saya dapat merasakan suasana atau lingkungan kerja sesungguhnya sebagai seorang *developer*. Mengetahui lingkungan kerja dapat membuat saya menjadi lebih nyaman ketika benar-benar terjun ke dunia profesional nantinya.

Dengan melaksanakan magang, saya mengetahui bagaimana mengerjakan sebuah proyek yang digunakan pada sebuah perusahaan. Saya lebih memahami manfaat dari koordinasi dan kolaborasi dalam pengerjaan sebuah proyek. Meskipun dalam pengembangan CMS ini saya melakukannya secara perseorangan, namun tidak dapat terlepas dari bantuan rekan saya Desya dan bantuan dari Ibu Christie.

Karena kami bekerja secara *remote*, yang berarti ketika kami ada pertanyaan maupun saat ingin berkomunikasi kami harus menunggu BSA ACC di Jakarta meluangkan waktu untuk video call, terasa tidak mudah. Kegiatan *stand up meeting* yang diwakilkan dalam aplikasi Slack (menggunakan Geekbot), mengajarkan saya mengenai pentingnya kerja sama dan komunikasi. Komunikasi diperlukan untuk memastikan masing-masing anggota tim dapat mengerjakan tugasnya sesuai dengan yang diharapkan. Dengan adanya komunikasi yang baik, kendala yang dihadapi akan lebih mudah dicari penyelesaiannya.

Dalam pengerjaan proyek CMS, saya menyadari bahwa sebuah aplikasi yang telah selesai dibuat secara fungsionalitas masih dapat dikembangkan melalui CMS. Selain itu dengan dibuatnya API untuk setiap proses dalam CMS dapat berguna apabila API tersebut akan digunakan dalam aplikasi web/mobile lain. Penamaan *variable* dan penulisan *code* juga sangat penting karena bukan hanya kita yang akan menggunakan *code* itu. Pelaksanaan magang juga membuat saya mengetahui tentang teknologi-teknologi yang saat ini digunakan di bidang industri IT, setidaknya pada lingkup perusahaan tempat saya magang.

3.2 Penerapan Ilmu dalam Magang

Pelaksanaan magang ini dapat dilakukan dengan baik tidak terlepas dari peranan ilmu pengetahuan dan pengalaman yang didapatkan dalam perkuliahan. Pertama adalah kerja sama dalam sebuah tim. Sebagai mahasiswa, kerja sama tim terbentuk dari pengerjaan tugas perkuliahan yang diberikan secara berkelompok maupun keikutsertaan dalam berbagai organisasi.

Mata kuliah Pemrograman Basis Data yang telah mengajari mengenai PL/SQL (*Procedural Language for SQL*). PL/SQL ini akan sangat berguna dalam pengembangan aplikasi-aplikasi berbasis Desktop, terutama apabila database bersifat *credential*. PL/SQL ini juga sangat mempermudah *developer* dalam mengembangkan aplikasinya karena tinggal memanggil nama prosedur beserta parameternya saja. Saat magang, saya hanya menggunakan PL/SQL ini pada masa training karena saya tidak mendapat proyek berbasis Desktop.

Mata kuliah Pengembangan Aplikasi Web (PAW) yang banyak mengajari mengenai *framework* Laravel. Selain itu mata kuliah ini mengajari bahasa dasar yang digunakan dalam pengembangan aplikasi web seperti HTML, JS, CSS, dan PHP. Seluruh elemen yang saya sebutkan tadi merupakan komponen kunci dalam magang saya. Sebelum mengikuti magang, saya tidak akan pernah tahu bahwa JavaScript dan Laravel dapat dikombinasikan untuk membuat sebuah aplikasi yang sangat *powerful* dan *user friendly*. Saya juga baru mengerti apabila JavaScript dapat melakukan hal-hal ‘ajaib’ seperti *me-load* API. Saya merasa akan sangat berguna apabila hal ini diajarkan dalam mata kuliah PAW.

Mata kuliah Proyek Pengembangan Perangkat Lunak (P3L) memberikan pengalaman mengembangkan sebuah proyek perangkat lunak. Dalam pengembangan perangkat lunak pada mata kuliah ini, saya menggunakan bahasa pemrograman PHP-HTML dan *framework* Laravel. Hal ini tentu sangat membantu dalam pelaksanaan magang karena CMS yang saya kembangkan dalam magang ini juga menggunakan bahasa dan *framework* yang sama.

BAB IV

KESIMPULAN

Magang yang saya lakukan pada tanggal 3 Januari – 7 April 2019, untuk perusahaan ACC (PT. Astra Sedaya Finance) telah berjalan dengan baik dan lancar berkat dukungan dari berbagai pihak. Dalam pelaksanaan magang ini saya satu tim dengan Desya Kristianto. Saya ditugaskan untuk mengerjakan proyek *Content Management System* (CMS) untuk aplikasi BidMart. Proyek CMS BidMart telah selesai kami kembangkan dengan memiliki menu untuk mengatur elemen-elemen dalam proses pelelangan, fungsionalitas untuk memantau transaksi, *user management*, *history*, dan mengatur beberapa hal di aplikasi. Sedangkan fungsionalitas dari apa yang saya kerjakan kebanyakan hanya pada operasi CRUD (*Create, Read, Update, Delete*). Dalam pengerjaan CMS ini saya mengalami beberapa kendala. Namun, dengan adanya peranan pembimbing lapangan dan rekan-rekan yang magang di ACC, kendala-kendala tersebut dapat diatasi dengan baik.

Pelaksanaan magang memberikan banyak manfaat bagi saya yang akan melanjutkan ke dunia kerja setelah menyelesaikan perkuliahan. Dengan melaksanakan magang, saya mengetahui bagaimana suasana dan lingkungan kerja dibidang teknologi, pengalaman mengerjakan projek, serta pentingnya koordinasi, kerjasama, dan komunikasi. Selain itu, saya juga menyadari bahwa teknologi berkembang dengan sangat cepat. Oleh karena itu, diperlukan untuk selalu mempelajari teknologi-teknologi terbaru.

Pada pelaksanaan magang ini saya juga dapat menerapkan ilmu pengetahuan dan pengalaman yang saya dapatkan selama melaksanakan perkuliahan. Meskipun tidak seluruh ilmu dan pengetahuan yang dibutuhkan dalam magang ini diajarkan pada perkuliahan, namun dasar-dasar pengetahuan tersebut sebagian besar telah diajarkan. Saya juga menyadari bahwa saya harus selalu memperbarui pengetahuan, terbuka terhadap hal-hal baru, dan tidak hanya berpangku pada apa yang disampaikan pada bangku perkuliahan.