

**LAPORAN KERJA PRAKTEK**  
**Future Program PT. Global Digital Niaga**  
**( Blibli.com )**



**Dipersiapkan oleh:**  
**Satya Syahputra / 150708332**

**Program Studi Teknik Informatika**  
**Fakultas Teknologi Industri**  
**Universitas Atma Jaya Yogyakarta**  
**2018**

Jakarta, September 14<sup>th</sup>, 2018

**Letter of Recommendation for: Mr. Satya Syahputra**

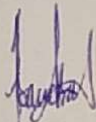
**TO WHOM IT MAY CONCERN :**

This certifies that Mr. Satya Syahputra had done apprentice program at PT. Global Digital Niaga from August 1<sup>st</sup>, 2018 until September 14<sup>th</sup>, 2018.

During the course of his apprentice, Mr. Satya Syahputra proved himself to be a hard worker. We are pleased to say that Mr. Satya Syahputra is a team player, a kind and sincere individual. He also carried out his responsibilities very well and could get along with the rest of the employees.

On behalf of PT. Global Digital Niaga, we thank Mr. Satya Syahputra for contributing to company's excellence and wish him the best for many years to come.

Your faithfully,



**Sandra Kumalasari**

Sr.VP People Operations & General Services

**HALAMAN PENGESAHAN**  
Laporan Kerja Praktek  
Future Program PT. Global Digital Niaga  
( Blibli.com )



Laporan ini telah diperiksa dan disetujui  
Pada tanggal : 2018

Oleh :

**Dosen Pembimbing,**

( Martinus Maslim, S.T., M.T. )

**Pembimbing Lapangan,**

( Daniel Hutama Putra )

## KATA PENGANTAR

Puji syukur penulis panjatkan kepada Tuhan Yang Maha Esa yang telah memberikan rahmat dan karunia-Nya, sehingga penulis dapat menyelesaikan laporan kerja praktek ini. Kerja Praktek 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 Kerja Praktek diharapkan mahasiswa dapat memperluas pengetahuan dan pemahaman mengenai disiplin ilmu disertai penerapannya secara nyata. Mengingat dunia kerja saat ini membutuhkan tenaga kerja yang berpengalaman bagi sebuah perusahaan. Laporan ini dibuat berdasarkan pengalaman yang didapatkan dari kerja praktek yang telah ditempuh dari tanggal 1 Agustus 2018 sampai dengan 14 September 2018 di PT. Global Digital Niaga ( Blibli.com ).

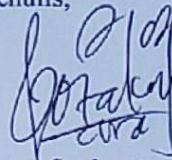
Penyusunan laporan ini tidak terlepas dari bantuan beberapa pihak, oleh karena itu penulis hendak mengucapkan terimakasih kepada:

1. Tuhan Yesus Kristus, yang selalu menyertai dan mencurahkan berkat, 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 kerja praktek.
3. Bapak Martinus Maslim, S.T., M.T., selaku dosen pembimbing yang telah membimbing penulis selama pelaksanaan kerja praktek maupun penyusunan laporan.
4. Kak Daniel Utama Putra, selaku pembimbing lapangan yang telah membimbing penulis selama pelaksanaan kerja praktek.
5. Seluruh teman yang tidak dapat disebutkan satu persatu yang telah memberikan bantuan kepada penulis selama pelaksanaan kerja praktek.

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

Yogyakarta, 14 September 2018

Penulis,



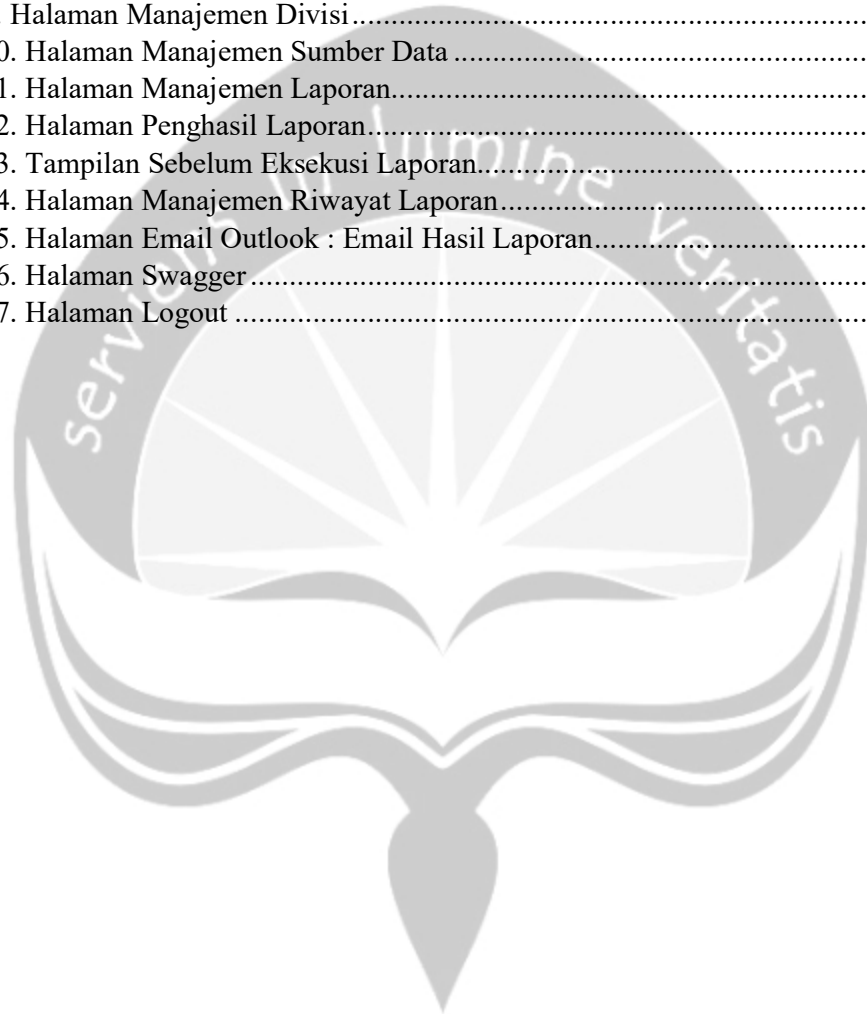
Satya Syahputra

# DAFTAR ISI

<b>HALAMAN PENGESAHAN</b> .....	<b>3</b>
<b>KATA PENGANTAR</b> .....	<b>4</b>
<b>DAFTAR ISI</b> .....	<b>6</b>
<b>DAFTAR GAMBAR</b> .....	<b>7</b>
<b>BAB I PENDAHULUAN</b> .....	<b>8</b>
1.1    Sekilas Perusahaan.....	8
1.2    Sejarah Perusahaan.....	8
1.3    Visi dan Misi Perusahaan.....	9
1.4    Struktur Organisasi.....	10
1.5    Deskripsi Tugas Struktur Organisasi Perusahaan.....	12
1.6    Departemen Teknologi Informasi(TI) dalam perusahaan.....	13
1.7    Deskripsi Future Program PT. Global Digital Niaga (Blibli.com).....	17
<b>BAB II PELAKSANAAN KERJA PRAKTEK</b> .....	<b>18</b>
2.1    Penjelasan <i>Logbook</i> .....	18
2.2    Hasil Pekerjaan Secara Umum.....	32
2.3    Bukti Hasil Pekerjaan.....	34
<b>BAB III HASIL PEMBELAJARAN</b> .....	<b>40</b>
3.1    Manfaat Kerja Praktek.....	40
3.2    Penerapan Ilmu dalam Kerja Praktek.....	40
<b>BAB IV KESIMPULAN</b> .....	<b>42</b>
<b>LAMPIRAN</b> .....	<b>43</b>

## DAFTAR GAMBAR

Gambar 1. Bagan Struktur Organisasi : Inti .....	10
Gambar 2. Bagan Struktur Organisasi : Departemen Teknologi .....	10
Gambar 3. Bagan Struktur Organisasi : Services .....	11
Gambar 4. Bagan Struktur Organisasi : Commerce & QA .....	11
Gambar 5. Tampilan Pengguna Sebelum di Perbarui .....	21
Gambar 6. Tampilan Pengguna Setelah di Perbarui .....	21
Gambar 7. Slides Presentasi Ujian Future Program .....	23
Gambar 8. Halaman login sistem Blibli.com .....	34
Gambar 9. Halaman Manajemen Divisi .....	35
Gambar 10. Halaman Manajemen Sumber Data .....	35
Gambar 11. Halaman Manajemen Laporan .....	36
Gambar 12. Halaman Penghasil Laporan .....	37
Gambar 13. Tampilan Sebelum Eksekusi Laporan .....	37
Gambar 14. Halaman Manajemen Riwayat Laporan .....	38
Gambar 15. Halaman Email Outlook : Email Hasil Laporan .....	38
Gambar 16. Halaman Swagger .....	39
Gambar 17. Halaman Logout .....	39



# BAB I

## PENDAHULUAN

### 1.1 Sekilas Perusahaan

Belakangan ini kita dapat menemukan banyak inovasi pada jenis bisnis di Indonesia, salah satu perkembangan yang kita dapat rasakan dampaknya ialah hadirnya *e-commerce* sebagai warna baru pada dunia bisnis di Indonesia. Menurut Loudon (1998) *E-commerce* atau kependekan dari *electronic commerce* (perdagangan secara elektronik), suatu proses transaksi yang dilakukan oleh pembeli dan penjual dalam membeli dan menjual berbagai produk secara elektronik dari perusahaan ke perusahaan lain dengan menggunakan komputer sebagai perantara transaksi bisnis yang dilakukan.. Salah satu perusahaan *e-commerce* yang hadir di Indonesia adalah Blibli.com.

Blibli.com adalah perusahaan *e-commerce* yang ada di Indonesia dengan konsep *online mall* melalui 11 kategori pilihan yaitu Fashion Wanita, Fashion Pria, Kesehatan & Kecantikan, Ibu & Anak, Kuliner, Handphone & Tablet, Komputer & Laptop, Kamera, Peralatan Elektronik, Hobi & Olahraga, dan Otomotif. Dengan konsep tersebut Blibli.com berharap masyarakat dapat berbelanja dengan mudah, menyenangkan, dimanapun, dan kapanpun. Blibli adalah produk pertama PT Global Digital Niaga yang merupakan anak perusahaan Djarum dibidang digital.

### 1.2 Sejarah Perusahaan

Saat ini Blibli.com telah menjadi *startup e-commerce* yang terkenal dan diminati oleh banyak kalangan. Dengan konsep sebagai *mall online*, Blibli.com memberikan berbagai kemudahan bagi para pedagang dan pembeli untuk menjual dan membeli barang. Pedagang dapat dengan mudah memasukkan, memasarkan dan menjual produk, sedangkan pembeli dapat dengan mudah mencari barang yang diinginkan. Didirikan pada 25 Juli 2011 oleh PT. Global Digital Niaga (GDN) yang merupakan anak perusahaan PT. Global Digital Prima (GDP) milik Martin Hartono – putra pemilik Djarum Group, Budi Hartono. PT. Global Digital Prima (GDP) adalah bagian dari perusahaan produsen rokok terbesar di Indonesia Djarum Group yang juga menjadi pemegang saham beberapa bisnis *online* terkemuka seperti situs *e-commerce*



Kaskus, serta beberapa situs lokal seperti lintasberita.com, krazymarket.com, dailysocial.net, dan perusahaan inkubator bisnis online bernama Merah Putih Inc. Keberadaan Blibli.com dengan nilai investasi 10 juta dolar menjadi pertanda keseriusan dan eksistensi Djarum Group dalam bisnis *online*.

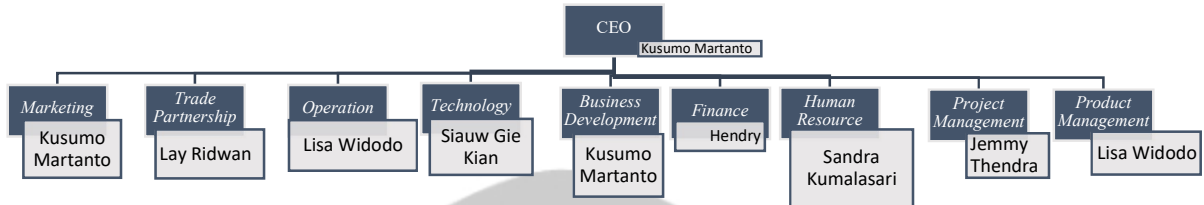
Blibli.com dipelopori tenaga-tenaga muda profesional dengan latar pendidikan rata-rata lulusan dari mancanegara. Salah satu sosok penting dibalik berdirinya Blibli.com adalah Kusumo Martanto, lulusan S-2 Teknik Industri dari *Georgia Institute of Technology*, Amerika Serikat yang kini menjabat *CEO* PT. Global Digital Niaga. Beliau dipercaya untuk memimpin Blibli.com setelah sekian lama bersekolah dan bekerja di Amerika Serikat.

### **1.3 Visi dan Misi Perusahaan**

- a. Visi Perusahaan PT Global Digital Niaga (Blibli.com) :  
*To be 'No 1' e-commerce in Indonesia with the most loyal and satisfied customer*
- b. Misi Perusahaan PT Global Digital Niaga (Blibli.com) :  
*Empowering generations and bringing the best e-commerce experience for a better life.*

## 1.4 Struktur Organisasi

### a. Struktur Inti



*Gambar 1. Bagan Struktur Organisasi : Inti*

**Gambar 1.1** merupakan gambar struktur organisasi inti dan nama pemegang jabatan yang berada di Blibli.com.

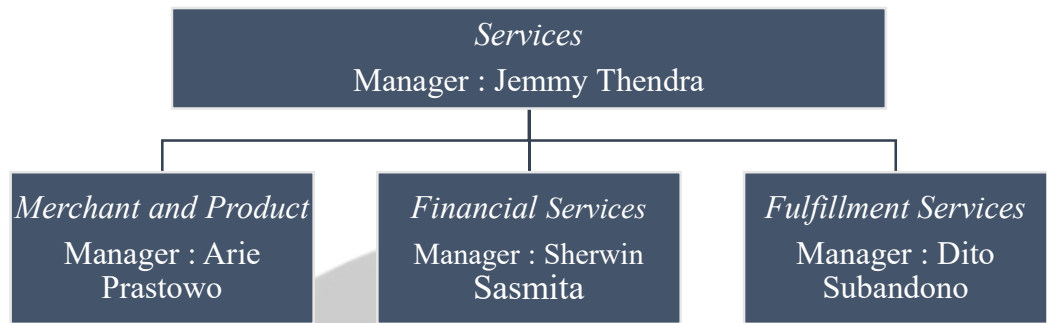
### b. Departemen Teknologi



*Gambar 2. Bagan Struktur Organisasi : Departemen Teknologi*

**Gambar 1.2** merupakan bagan struktur Departemen IT dan pemegang jabatannya dalam Blibli.com. Bertanggung jawab dalam kelangsungan teknologi dalam Blibli.com.

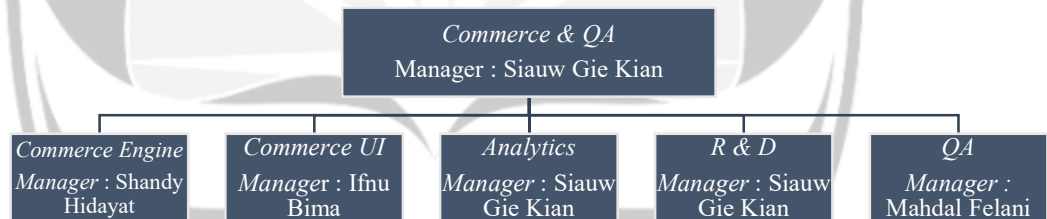
**c. Layanan**



**Gambar 3. Bagan Struktur Organisasi : Services**

**Gambar 1.3** merupakan bagan struktur Layanan beserta orang yang menduduki jabatan tersebut. Bagian Layanan bertanggung jawab dalam mengurus teknologi, layanan dan data yang berhubungan dengan pedagang, finansial, dan logistik.

**d. Commerce and QA (Quality Assurance)**



**Gambar 4. Bagan Struktur Organisasi : Commerce & QA**

**Gambar 1.4** merupakan bagan struktur *Commerce and QA* beserta orang yang menduduki jabatan tersebut. Bagian *Commerce and QA* bertanggung jawab dalam mengurus teknologi, layanan dan data yang berhubungan dengan *User* dalam hal ini pelanggan.

## 1.5 Deskripsi Tugas Struktur Organisasi Perusahaan

a. *Chief Executive Officer (CEO)*

Merupakan eksekutif dengan ranking tertinggi pada perusahaan. Tanggung jawab utama dari CEO adalah membuat keputusan perusahaan, mengatur operasi, dan sumber perusahaan secara keseluruhan, serta menjadi titik utama komunikasi antara jajaran direktur dengan operasi perusahaan.

b. *Marketing*

Merupakan bagian perusahaan yang memiliki tanggung jawab dalam memasarkan produk / jasa dari Blibli.com, menarik konsumen, melakukan riset terhadap pasar saat ini, serta *sales*.

c. *Trade Partnership*

Merupakan bagian perusahaan yang memiliki tanggung jawab dalam mengelola hubungan dengan pihak luar. Salah satu contohnya adalah hubungan antara Blibli.com dengan pihak jasa pengiriman untuk kurir yang mengantarkan pesanan.

d. *Operation*

Merupakan bagian perusahaan yang bertanggung jawab pada semua aktivitas operasional perusahaan, mulai dari perencanaan proses hingga bertanggung jawab pada hasil akhir proses.

e. *Technology*

Merupakan bagian perusahaan yang bertanggung jawab dalam pembangunan, pemeliharaan dan pembaruan teknologi yang digunakan. Di Blibli.com terbagi menjadi 2 bagian, yaitu untuk sisi *software* dan *infrastructure*.

f. *Business Development*

Merupakan bagian perusahaan yang bertanggung jawab dalam mengembangkan bisnis perusahaan.

g. *Finance*

Merupakan bagian perusahaan yang bertanggung jawab dalam mengatur keuangan, akuntansi, investasi dari Blibli.com.

h. *Human Resources*

Merupakan bagian perusahaan yang bertanggung jawab dalam mengelola Sumber Daya Manusia dari Blibli.com baik untuk seleksi, penerimaan dan pelatihannya.

i. *Project Management*

Merupakan bagian perusahaan yang bertanggung jawab dalam mengelola proyek – proyek yang sedang dikerjakan dan dikembangkan Blibli.com.

j. *Product Management*

Merupakan bagian perusahaan yang bertanggung jawab dalam mempertahankan visi produk dan *roadmap* perusahaan, menciptakan hubungan yang baik dengan tim dan *stakeholders*, serta meningkatkan perkembangan produk yang ada.

## 1.6 Departemen Teknologi Informasi(TI) dalam perusahaan

1. *Chief Technology Officer (CTO)*

Merupakan bagian dari Departemen IT Blibli yang bertanggung jawab akan kesesuaian dan kemajuan teknologi membantu dan searah terhadap strategi bisnisnya. Selain itu CTO juga bertugas menentukan *platform* dan *technical design* dalam teknologi Blibli.com.

2. *Infrastruktur dan Operasi*

Merupakan bagian yang mengatur perangkat keras maupun jaringan (baik untuk internal maupun untuk eksternal) Blibli.com.

Dalam Bidang ini terdapat beberapa tim utama, yaitu:

a. *GRC (Governance, Risk, Compliance)* Bertanggung jawab dalam mengatur kebijakan (berkaitan dengan UU ITE, Resiko, Peraturan perbankan) serta teknikal ( Memberikan edukasi *Secure Coding* terhadap *developer-developer* di Blibli.com agar dapat melakukan pengembangan produk secara aman.)

b. *Sys Admin (System Admin)* Bertanggung jawab dalam :

1. *Server*

Mengurus server yang digunakan oleh blibli, baik dalam instalasi maupun pengembangannya.

2. *Sistem Operasi*

3. *Alert*

- c. *Otomation* (Sedang direncanakan)  
Untuk mengotomatisasi beberapa hal yang memakan cukup banyak waktu ketika dilakukan secara manual.
- d. *SRE (Side Reliability Engineer)* (Sedang direncanakan)  
Melakukan *troubleshooting* atas kesalahan-kesalahan yang terjadi di infrastruktur Blibli.com .
- e. *Network*  
Bertanggung jawab terhadap data center Blibli.com dan *campus network* (untuk kantor).
- f. *Team Support* (Sedang direncanakan)  
Untuk mengurus hal-hal berkaitan dengan masalah sehari-hari seperti printer yang tidak mau terkoneksi, jaringan internet yang tiba-tiba terputus dan sebagainya.

### 3. Layanan

#### a. *Merchant and Product*

Merupakan divisi yang bertanggung jawab di *back office process*. Pada bagian ini bertanggung jawab di bagian urusan pedagang dan pengembalian produk. Untuk bagian Pedagang, bertugas dalam mengurus aplikasi dan menambah fitur baru (jika sekiranya dibutuhkan).

#### b. *Financial Services*

Disebut juga divisi internal, merupakan divisi yang bertanggung jawab dalam pembuatan aplikasi untuk keperluan internal blibli.com. Merupakan *Back-End* dari aplikasi Blibli.com namun lebih belakang dengan *Core Commerce (Customer Facing)*. Terdapat beberapa hal penting dalam divisi ini yaitu :

##### 1. Xfinance

Aplikasi buatan blibli untuk mengatur dan mencatat *cash flow*, pengembalian produk, pembayaran kepada pedagang yang digunakan dibagian finansial. Merupakan jembatan antar divisi yang membutuhkan data finansial.

##### 2. OpenBravo

Aplikasi ERP (*Enterprise Resources Planning*) yang digunakan dengan kostumisasi untuk mengorganisasikan akuntansi,

*Warehousing Supply Chain Management, Purchase, Sale dan Inventory.*

### 3. IDV

*Integrated Data Viewer*, sebagai penampil status untuk memudahkan *Customer Service* dalam melihat status baik pengiriman barang maupun pembayaran.

#### c. *Fulfillment Services*

Disebut juga Divisi *Supply Chain Management* merupakan divisi yang bertanggung jawab terhadap pengkoordinasian pemindahtanganan barang dari pedagang/perusahaan ke tangan pelanggan. Terdapat 4 tim dalam divisi ini :

##### 1. Xfraud

Untuk memastikan kebenaran pemesanan, alamat, barang dan sebagainya.

##### 2. Logistik

Bertanggungjawab terhadap sistem dalam pencatatan *traffic* bar .

##### 3. Stockholm

Bertanggung jawab terhadap stok barang yang terdapat di gudang.

##### 4. TMS (*Transportation Management System*),

Bertanggung jawab terhadap pengiriman barang, menggunakan layanan dari mana.

### 4. *Commerce and Quality Assurance*

#### a. *Commerce Engine*

Bertugas mengurus area *Back end* dari website dan aplikasi *mobile* Blibli.com. Memiliki divisi bernama *Development*. Untuk produk yang sudah dihasilkan dan telah berjalan hingga saat ini adalah : XPayment, XInventory, XCommerce dan Xcart.

Divisi ini memiliki 3 tugas utama :

1. *Refactoring*

Untuk memindahkan / menulis kembali dari satu bahasa ke bahasa lain, yang lebih baru dan memiliki keuntungan lebih.

2. *New Features*

Menambahkan fitur baru, misalnya *payment gateway*, *payment router* dan *shipping service*.

3. *New Product line*

Mengembangkan produk / layanan baru , seperti :

- a. Xrail ( tiket kereta api)
- b. Xpulsa
- c. Xwallet

b. *Commerce User Interface*

Bertugas mengembangkan antar muka (menerjemahkan hasil desain tim UI-UX ke dalam bentuk kode) dari website dan aplikasi mobile Blibli.com. Selain itu memastikan pengguna akhir tidak mengalami kesulitan ketika menggunakannya.

c. *Analytics*

Bertanggung jawab dalam menganalisis, selain bertugas mengumpulkan, mengolah dan menyajikan data, divisi ini juga memanfaatkannya untuk keperluan pengambilan keputusan dan peningkatan rekomendasi produk di web Blibli.com.

d. *Research and Development*

Bertanggung jawab dalam melaksanakan riset terhadap data yang ada dan menggunakannya untuk mengembangkan produk atau fitur teknologi Blibli.com.

e. *Quality Assurance*

Merupakan Divisi yang bertanggung jawab terhadap kualitas suatu aplikasi serta memastikan bahwa aplikasi tersebut telah layak dipakai , tanpa ada masalah yang berarti (karena kesalahan dalam proses pembuatan).



### 1.7 Deskripsi Future Program PT. Global Digital Niaga (Blibli.com)

*Future Program* dikeluarkan pada tahun 2016, program ini diperuntukkan bagi para mahasiswa-mahasiswi yang mengambil jurusan teknologi (Teknik Informatika, Sistem Informasi, Ilmu Komputer, dan sejenisnya) yang telah sedang duduk di semester 4. Program ini ditujukan untuk mempersiapkan para wisudawan ketika berhadapan dengan dunia kerja nantinya. Selain itu, program ini juga dimaksudkan untuk membentuk mental dan sikap. Hingga saat ini, terdapat beberapa universitas yang terlibat dalam kerja sama program *future*, yaitu :

1. Universitas Atma Jaya Yogyakarta
2. Universitas Duta Wacana Yogyakarta
3. Universitas Gadjah Mada Yogyakarta
4. Universitas Maranatha Bandung
5. Institut Teknologi Bandung
6. Politeknik Elektronika Negeri Surabaya
7. Universitas Brawijaya Malang
8. Universitas Bina Nusantara
9. Institut Teknologi Sepuluh Nopember (ITS)
10. Universitas Indonesia

Hingga saat ini, program ini telah memiliki 3 batch. Kegiatan batch yang pertama berlangsung pada Agustus 2016, sedangkan yang kedua berlangsung pada Agustus 2017, dan ketiga berlangsung pada Agustus 2018. Peserta dari program ini dibagi menjadi 2 jurusan, yaitu Developer dan Infrastruktur. Pada jurusan Developer peserta program akan diajar dan dilatih untuk menyelesaikan masalah dengan prinsip desain dan implementasi rekayasa perangkat lunak. Pada jurusan Infrakstruktur peserta program akan diajar tentang dasar-dasar jaringan, *server*, dan keamanan sistem. Seluruh peserta program dilatih untuk menjadi individu yang siap dalam dunia pekerjaan dan mampu bekerja didalam sebuah tim.

## BAB II

### PELAKSANAAN KERJA PRAKTEK

#### 2.1 Penjelasan *Logbook*

##### Rabu, 01 Agustus 2018 ( 08.00 – 18.00 )

Pada hari pertama saya pertama sebelum memulai kegiatan di kantor yaitu menemui *Human Resource Development* (HRD) dari Blibli.com untuk melakukan registrasi *fingerprint*. *Fingerprint* ini berguna untuk melakukan presensi masuk dan keluar kerja di Blibli.com. Sistem *fingerprint* untuk anak magang ini sendiri kebetulan berlaku semenjak akhir juli sehingga belum lama di terapkan. Sebelumnya presensi menggunakan catatan tertulis menggunakan lembar presensi yang diberikan oleh Blibli sendiri.

Selanjutnya melakukan *setting up* komputer yang digunakan dengan melakukan pemasangan alat – alat pengembangan serta *development kit* yang dibutuhkan. Pada hari pertama saya masih sedikit bingung dengan pemasangan aplikasi yang dibutuhkan, karena memerlukan setting khusus pada *certificate* untuk melakukan koneksi antara *Java Development Kit* (JDK) dengan internet. Sehingga pada hari pertama saya menggunakan laptop saya sendiri untuk mengerjakan sistem yang sedang saya kembangkan, yaitu Sleman Project.

##### Kamis, 02 Agustus 2018 ( 09.00 – 18.00 )

Pada hari kedua saya mulai mengerjakan fitur *Queue* untuk sistem Sleman Project. Sleman Project merupakan sistem untuk membuat laporan yang sedang dikembangkan di Blibli untuk penggunaan internal. Latar belakang sistem ini ialah karena sistem sebelumnya yaitu Jasper Report tidak berfungsi dengan baik dan sangat tidak mendukung kinerja yang baik terhadap basis data. Salah satu masalah yang terjadi pada Jasper Report adalah melakukan eksekusi tanpa melakukan antrian sehingga ketika banyak sekali orang yang melakukan eksekusi, akan membuat basis data menjadi *down*.

Oleh sebab itu Sleman Project yang akan menjadi pengganti dari Jasper Report diberi fitur *queue*. *Queue* yang saya kembangkan ini masih menggunakan cara yang sederhana, yaitu dengan melakukan cek terhadap status dari *row* eksekusi (pada tabel *Job Generate*) yang ada pada basis data. *Row* akan dieksekusi ketika status nya masih

*waiting* dan tidak ada status yang sedang *running* sehingga perlu melakukan cek pada kedua status tersebut sebelum menjalankan eksekusi. Kemudian setelah di eksekusi akan memiliki hasil yaitu *done* atau *failed*.

#### **Jum'at, 03 Agustus 2018 ( 08.00 – 18.00 )**

Pada hari ketiga saya mengerjakan fitur unduh berkas pada sistem Sleman Project. Tipe balikan yang diminta pada fitur unduh berkas ini adalah *ByteArray* dari berkas *\*.csv* yang berhasil di hasilkan oleh sistem supaya dapat diolah dan di unduh pada bagian *front-end*. Saya mencoba mempelajarinya dengan melakukan browsing tentang *input/output* pada java. Setelah melakukan pencarian, saya mendapatkan beberapa referensi untuk membaca file *\*.csv*.

Hasil eksekusi dibaca oleh sistem dengan menggunakan tipe data *File* yang kemudian membuat instance baru dengan satu parameter yang berisi *path* / lokasi berkas tersebut disimpan beserta nama berkas tersebut. Kemudian melalui sistem di ubah menjadi *ByteArray* dan di *return*. Untuk membaca nilai *bytes* dari berkas, saya menggunakan *FileInputStream*. Kemudian untuk melakukan tugas tersebut diperlukan *syntax try catch* untuk mencegah kegagalan berupa *exception* seperti *null pointer exception*.

#### **Senin, 06 Agustus 2018 ( 09.00 – 19.00 )**

Pada hari keempat saya mengerjakan *unit test* untuk *package service* pada sistem Sleman Project. Kelas yang saya buat untuk *unit test* yaitu *DatasourceServiceImpl* dan *DivisiServiceImpl*. Untuk melakukan pengujian ini yang menjadi standarnya adalah *line coverage*. Karena apabila *line coverage* sudah terpenuhi biasanya *class* dan *method coverage* sudah pasti dijalankan.

Yang digunakan untuk membantu melakukan pengujian ini adalah *jUnit 4* dan *Mockito/PowerMockito*. Pada *jUnit* sendiri merupakan dasar dari pengujian sedangkan *Mockito/PowerMockito* ini sebagai *support* untuk mempermudah melakukan pengujian seperti membuat *mock data*, *return* buatan dan verifikasi. Langkah pembuatan *unit test* sering kali tidak berubah, yaitu :

1. Melakukan *mock* atau pemalsuan pada kelas yang dibutuhkan.
2. Menganalisis atau menentukan, ketika suatu *method* di panggil, akan membalikan apa.

3. Menuliskan hasil analisis tersebut dengan menggunakan *when(...).thenReturn(...)*.
4. Membuat *response expected output*.
5. Memanggil *method* yang akan di uji dan ditampung hasil balikkannya.
6. Membandingkan hasilnya dengan *expected* yang sudah dibuat dengan menggunakan *assert*.
7. Setelah itu melakukan *verify* terhadap *method – method* yang dipanggil berdasarkan analisis tadi.

Kemudian untuk standar pengujian yaitu 95% *line coverage*, berdasarkan standar yang digunakan oleh Blibli.

#### **Selasa, 07 Agustus 2018 ( 09.00 – 20.00 )**

Pada hari kelima saya mengerjakan *unit test* untuk *package service* pada sistem Sleman Project. Kelas yang saya buat untuk *unit test* yaitu *UserServiceImpl* dan *ReportServiceImpl*. Untuk melakukan pengujian ini yang menjadi standarnya adalah *line coverage*. Karena apabila *line coverage* sudah terpenuhi biasanya *class* dan *method coverage* sudah pasti dijalankan.

Yang digunakan untuk membantu melakukan pengujian ini adalah *jUnit 4* dan *Mockito/PowerMockito*. Pada *jUnit* sendiri merupakan dasar dari pengujian sedangkan *Mockito/PowerMockito* ini sebagai *support* untuk mempermudah melakukan pengujian seperti membuat *mock data*, *return* buatan dan verifikasi. Langkah pembuatan *unit test* sama seperti sebelumnya mulai dari *mocking* sampai melakukan *verify*.

Kemudian untuk standar pengujian yaitu 95% *line coverage*, berdasarkan standar yang digunakan oleh Blibli.

#### **Rabu, 08 Agustus 2018 ( 08.00 – 18.00 )**

Pada hari keenam saya mengerjakan *unit test* untuk *package service* pada sistem Sleman Project. Kelas yang saya buat untuk *unit test* yaitu *RoleServiceImpl* dan *JobGenerateServiceImpl*. Untuk melakukan pengujian ini yang menjadi standarnya adalah *line coverage*. Karena apabila *line coverage* sudah terpenuhi biasanya *class* dan *method coverage* sudah pasti dijalankan.

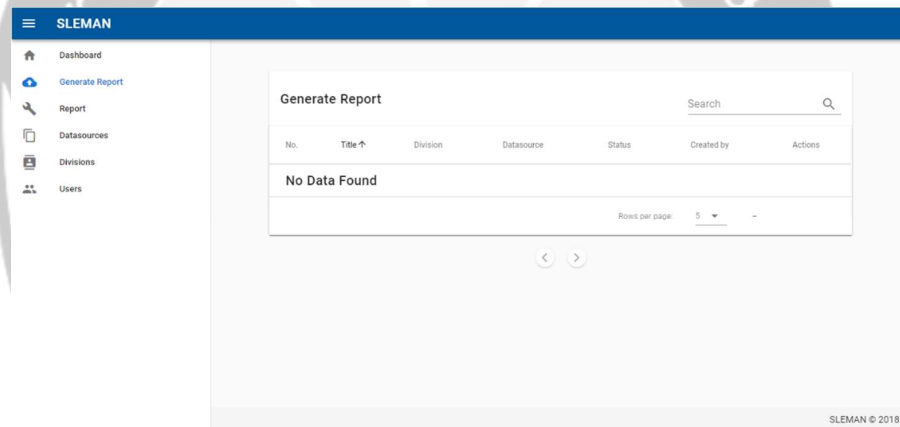
Yang digunakan untuk membantu melakukan pengujian ini adalah *jUnit 4* dan *Mockito/PowerMockito*. Pada *jUnit* sendiri merupakan dasar dari pengujian sedangkan

Mockito/PowerMockito ini sebagai *support* untuk mempermudah melakukan pengujian seperti membuat *mock data*, *return* buatan dan verifikasi. Langkah pembuatan *unit test* sama seperti sebelumnya mulai dari *mocking* sampai melakukan *verify*.

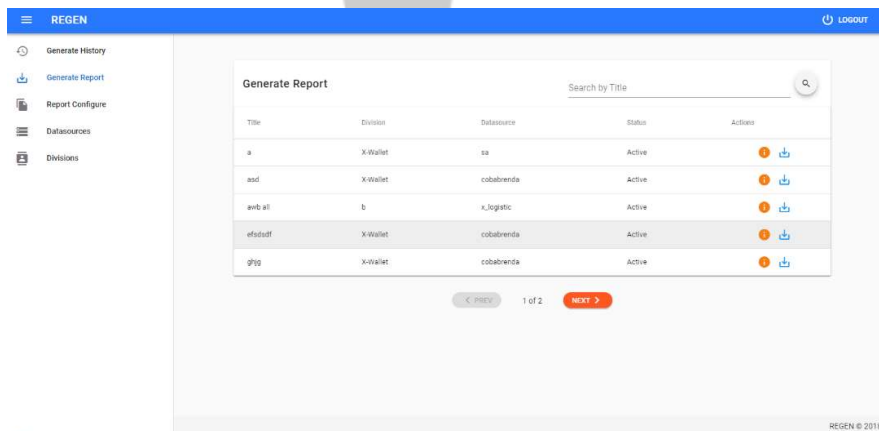
Kemudian untuk standar pengujian yaitu *95% line coverage*, berdasarkan standar yang digunakan oleh Bibli.

### **Kamis, 09 Agustus 2018 ( 08.00 – 20.00 )**

Pada hari ketujuh saya membetulkan Spesifikasi Kebutuhan Perangkat Lunak (SKPL) dari Sleman Project, karena melakukan perubahan tampilan *user interface*. Yang diperbarui ialah pada bagian gambar *user interface* ditambahkan *screenshot* dari tampilan saat ini. Hari ini saya hanya mengerjakan itu saja, karena teman tim saya mengambil alih beberapa *unit test package controller* sehingga saya memberikan arahan pada teman tim saya yang sedang mengerjakan *unit test*.



**Gambar 5. Tampilan Pengguna Sebelum di Perbarui**



**Gambar 6. Tampilan Pengguna Setelah di Perbarui**

**Jum'at, 10 Agustus 2018 ( 08.00 – 20.00 )**

Pada hari kedelapan saya mengerjakan *unit test* untuk *package controller* pada sistem Sleman Project. Kelas yang saya buat untuk *unit test* yaitu *ReportServiceImpl* dan *JobGenerateServiceImpl*. Untuk melakukan pengujian ini yang menjadi standarnya adalah *line coverage*. Karena apabila *line coverage* sudah terpenuhi biasanya *class* dan *method coverage* sudah pasti dijalankan.

Yang digunakan untuk membantu melakukan pengujian ini adalah *jUnit 4* dan *MockMvc*. *MockMvc* ini digunakan untuk melakukan request pada path url tertentu pada project sehingga menjadi seperti memanggil *Representation State (REST) Application Programming Interface (API) / REST API*. Pada *unit test controller* kali ini lebih mudah dibandingkan service dikarenakan *mock* data yang dibutuhkan tidak sebanyak *service* dan yang di verifikasi hanya *status code* nya saja. Langkah untuk melakukan pengujian ini yaitu :

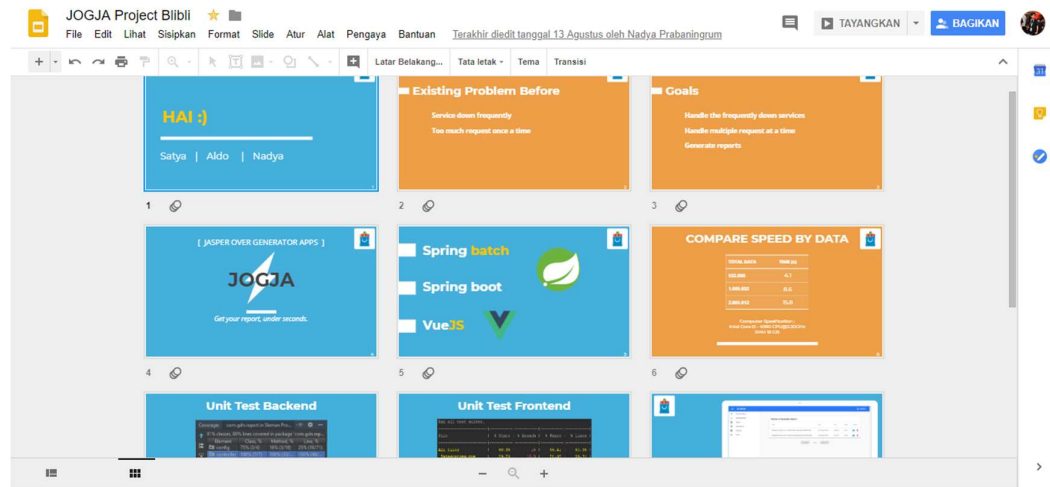
1. Melakukan *mock* atau pemalsuan pada kelas yang dibutuhkan.
2. Menganalisis penggunaan method HTTP *request* yang digunakan pada url yang akan dipanggil. Contoh *method* tersebut : GET, POST, PUT, DELETE.
3. Apabila dibutuhkan *parameter request* seperti *PathVariable*, *ParamRequest*, *RequestBody* dll maka perlu menambahkan ke dalam *builder MockMvc* tersebut.
4. Setelah itu ditambah lagi *verify* pada *builder* tersebut pada akhir untuk perbandingan *status code* yang dihasilkan, apabila berhasil maka akan memberikan *status code 200*.

Kemudian untuk standar pengujian yaitu *95% line coverage*, berdasarkan standar yang digunakan oleh Blibli.

**Senin, 13 Agustus 2018 ( 09.00 – 19.00 )**

Pada hari kesembilan saya mempersiapkan presentasi Sleman project kepada Bp Siaw Gie Kian dan rekan – rekannya. Serta mempersiapkan program untuk dipresentasikan kepada para juri. Pada hari ini saya hanya mempersiapkan diri untuk melakukan ujian *Future Program* pada esok hari. Serta kami (saya dan rekan saya Nadya) melakukan penggantian nama project menjadi *Jogja Project* dengan alasan karena Sleman tidak merepresentasikan apa yang kami kerjakan. Oleh karena itu

berubah menjadi Jogja, yang bila dijabarkan akan menjadi Jasper Over Generator (JOG).



*Gambar 7. Slides Presentasi Ujian Future Program*

**Selasa, 14 Agustus 2018 ( 09.00 - 19.00 )**

Pada hari kesepuluh saya maju mempresentasikan Sleman project kepada Bp Siaw Gie Kian dan rekan – rekannya. Yang ditanyakan dan dipertimbangkan untuk penilaian ialah pemahaman konsep tentang teknologi yang digunakan, serta konsep dasar pemrograman yang sering digunakan. Pada konteks ini saya menjelaskan tentang alur Spring Batch.

Setelah itu pada sore hari diumumkan apakah lulus ujian atau tidak. Saya bersyukur masih diberi kesempatan untuk menjadi bagian dari *Future Program*.

**Rabu, 15 Agustus 2018 ( 09.00 - 21.00 )**

Pada hari kesebelas saya diajarkan tentang *seven habits* yaitu *Synergize* dan *Sharpen the Saw*. Pada sesi ini kami di perlihatkan suatu negara yang msasyarakatnya merupakan dari golongan dan etnis yang berbeda serta agama yang berbeda, namun sangat aktif untuk saling membantu. Masyarakat pada negara itu percaya bahwa mereka dapat menjadi lebih berkembang dengan saling membantu, terbuka dengan yang lain, tidak egois akan diri sendiri. Hal ini disebut sinergi, sinergi ini membuat suatu tim atau kelompok menjadi lebih kuat dan akan menghasilkan suatu yang lebih luar biasa. Kemudian diberikan cuplikan video tentang

Setelah itu saya menuju lokasi bootcamp untuk bertemu dengan *junior* dan saling berbagi pengalaman dengan para *junior*. Saat sore hari saya diberikan materi

berbentuk permainan yang mengajarkan tentang kerjasama dan untuk saling memahami antar anggota tim.

**Kamis, 16 Agustus 2018 ( 09.00 - 17.00 )**

Pada hari kedua belas saya diajarkan kembali tentang *Synergize* dan *Sharpen the Saw*. Yang di mana saya diajarkan bekerja sama untuk memecahkan suatu masalah menggunakan metode *fishbone*, *brainstroming* dan *five why*. Kemudian kami diberi pembelajaran untuk memecahkan masalah pribadi dan tim.

Pada metode *fishbone* bekerja dengan cara menentukan kategori yang membuat masalah tersebut ada, kemudian memberikan cabang – cabang kecil yang merupakan masalah yang lebih detil lagi dari kategori yang ditentukan. Setelah itu menghitung banyaknya cabang kecil disetiap kategori tersebut. Ketika suatu kategori tersebut memiliki cabang kecil paling banyak, maka kita dapat menentukan prioritas nya mulai dari yang paling banyak.

Pada metode *brainstorming* bekerja dengan cara kami berdiskusi untuk saling memberikan ide tanpa saling mengomentari. Disini kami hanya saling menuangkan ide dalam kertas sampai sekiranya sudah tidak ada ide lagi untuk di tuangkan. Baru setelah itu di diskusikan bersama sama dengan memilah ide - ide yang dapat dibilang cocok untuk menyelesaikan masalah yang sedang dihadapi. Dengan itu baru bisa di kerucutkan lagi dengan memberikan prioritas dan eksekusi ide.

Pada metode *five why* bekerja dengan cara menanyakan pada diri sendiri dengan menggunakan *why*. Contoh: Saya telat masuk kelas. Saya akan mencari tahu dimulai dengan “Mengapa saya telah masuk kelas?” karena kesiangan, “Mengapa saya kesiangan?” karena saya mematikan alarm dan tertidur lagi, “Mengapa saya mematikan alarm dan tertidur lagi?” karena saya masih mengantuk, “Mengapa saya masih mengantuk?” karena saya kurang waktu tidur, “Mengapa saya kurang waktu tidur?” karena saya tidur larut malam, “Mengapa saya tidur larut malam?” karena saya main *video game*. Maka dengan begitu akan didapatkan jawaban bahwa saya terlambat karena saya main *video game* sampai larut malam. Tujuan dari *five why* ini adalah mengerucutkan masalah berdasarkan sebab akibat nya.



**Jum'at 17 Agustus 2018 ( 09.00 - 18.00 )**

Pada hari ketiga belas saya diajarkan materi android. Hal yang diajarkan pada kelas ini ialah mengenai Activity, RecyclerView dan Retrofit. Untuk melakukan sesi praktek ini, kami akan membuat suatu aplikasi sederhana yaitu Aplikasi Movie.

Yang pertama diajarkan adalah pembuatan tampilan menggunakan xml. Sebelum kami memulai pembuatan, kami dijelaskan akan penggunaan layout yang cocok untuk pembuatannya. Disini kami di jelaskan tentang LinearLayout, GridLayout, RelativeLayout dan ConstraintsLayout. Sesuai penjelasannya, LinearLayout cocok untuk tampilan yang berbentuk daftar secara vertical maupun horizontal. GridLayout cocok untuk tampilan yang akan dibuat seperti tabel namun bukan tabel, layout ini membentuk kotak kotak. RelativeLayout cocok untuk tampilan yang kompleks, sehingga mudah untuk menempatkan posisi yang tepat. ConstraintsLayout cocok untuk tampilan yang kompleks seperti RelativeLayout, namun lebih di dukung dengan adanya bias yang digunakan untuk menentukan posisi suatu *item*, sehingga hasilnya pun akan lebih *responsive* pada ukuran layar yang berbeda – beda.

Setelah itu kami diajarkan menggunakan RecyclerView merupakan tampilan yang berbentuk *List*, *Grid* atau *Staggered*. RecyclerView ini sungguh membantu developer untuk membuat tampilan yang memiliki banyak item. Pada kali ini kami mempelajari membuat tampilan tersebut dengan mendapatkan data dari suatu API yang didapatkan dari themoviedb. Untuk melakukan pengaturan tampilan pada RecyclerView, diperlukan *adapter*. *Adapter* ini digunakan sebagai pengolah data untuk ditampilkan pada layar pengguna.

Selanjutnya kami diberikan cara untuk mengambil data dari API yang tersedia dengan menggunakan Retrofit yang merupakan HttpClient yang digunakan untuk menggunakan services REST. Konfigurasi yang kami butuhkan untuk menggunakan retrofit ialah dengan membuat *instance* retrofit pada *project* kita. Kemudian memberikan *base url* supaya mudah bagi kita untuk melakukan perubahan apabila lokasi *service* berubah. Setelah itu membuat sebuah kelas *interface* yang digunakan untuk melakukan akses pada API, pada setiap *method* pada *interface* ini akan diberikan setidaknya satu *method* HTTP *Request* dengan menuliskan nama anotasinya. Pada kali ini kami hanya menggunakan GET saja, karena hanya mengambil data dari *services*.

**Sabtu, 18 Agustus 2018 ( 09.00 - 18.00 )**

Pada hari keempat belas saya diajarkan materi *Automated Testing*, menggunakan *Cucumber Java Virtual Machine*. Hal yang diajarkan pada kelas ini ialah mengenai pembuatan *project test* dengan menggunakan maven *project*. Yang di uji ialah tampilan dari suatu halaman web dan *API service*.

*Automated testing* ini digunakan untuk melakukan tugas – tugas secara otomatis dengan menghubungkan *item* pada tampilan dengan *project testing* yang kita gunakan. Pada sesi ini kami membuat sistem testing yang melakukan pengisian pada suatu form yang lengkap berisi *text field*, *combo box*, *radio button* dan *button*. Selanjutnya akan membuat *feature* yang berisi tugas – tugas yang harus dilakukan, pada *feature* ini juga berisi Skenario alur tugas yang harus dijalankan, jadi akan lebih jelas bagaimana tugas tersebut dilakukan. Selanjutnya membuat kelas yang berisi skenario yang dibuat tadi dan kemudian untuk menyambungkan *item* pada tampilan dengan sistem *automated testing* dengan menggunakan *xpath* dari *item* tersebut. Setelah itu melakukan pengkodean java biasa diapakan dan diberi nilai apa pada setiap *item* yang dikaitkan tersebut. Dari sini dapat dilakukan untuk mengatur semua *item* yang sudah dikaitkan.

**Minggu, 19 Agustus 2018 ( 09.00 - 18.00 )**

Pada hari kelima belas saya diajarkan materi *Perfomance Testing* dan *User Experience (UX)*. Hal yang diajarkan pada *Perfomance Testing* ialah penggunaan *jMeter* untuk melakukan request pada suatu alamat web / *API*. Perfoma yang dimaksud ialah ketika ada request yang banyak sekaligus atau bertahap dalam beberapa detik dan dapat dilihat berapa kali *request* berhasil atau gagal. Selain itu dapat juga diatur parameter yang diperlukan dan method *HTTP Request*. Hal yang kami lakukan adalah menggunakan *request* sederhana dengan inputan dua *field*. Kemudian menambahkan *item* yang digunakan untuk melakukan request namanya adalah “*HTTP Request*”, disini tinggal memasukkan url yang akan digunakan untuk *request* dan parameter yang dimasukkan. Setelah itu kita juga dapat melakukan berapa kali *request* tersebut akan dilakukan juga mungkin melakukan *request* setiap detiknya.

Kemudian pada *UX* saya belajar untuk lebih memahami pengguna dan menentukan target pengguna dengan menentukan persona, menerapkan konsep, memberikan solusi dan melakukan eksekusi. Untuk hal ini kami dibagi menjadi beberapa kelompok dan diminta untuk membuat suatu solusi untuk masalah pada masyarakat / lingkungan saat ini. Alur dari permainan ini adalah ketika kami sudah

berkelompok dan mendapatkan solusi dari suatu masalah maka selanjutnya adalah melakukan *brainstroming* tentang apa saja yang akan di lakukan supaya bisa memecahkan masalah yang sedang terjadi dan memilih ide – ide yang cocok. Kemudian membuat rencana untuk melakukan eksekusi ide – ide yang dipilih tadi dan terakhir melakukan eksekusi dengan menceritakan alur penggunaannya.

#### **Senin, 20 Agustus 2018 (09.00 - 18.00)**

Pada hari keenam belas saya kembali beraktifitas dikantor. Saya memikirkan masalah yang terjadi pada masyarakat / perusahaan / sosial untuk *Future Program*, fase selanjutnya. Serta sambil menunggu mentor saya untuk memasukkan saya ke dalam tim mana. Ide yang bayangkan adalah pembuatan sosial media khusus informasi sekitar.

Sosial media khusus informasi sekitar yang saya bayangkan seperti penggunaan media sosial Facebook, yaitu dengan masuk ke dalam grup atau topik untuk berbagi informasi sesuai seperti info cegatan, bencana alam, lowongan kerja dan sebagainya. Kemudian pada hal ini setiap pengguna dapat memiliki akun yang digunakan sebagai identitas. Serta untuk tampilannya akan menggunakan gaya *google material* yaitu berbentuk *card* setiap feednya.

#### **Selasa, 21 Agustus 2018 (08.00 - 18.00)**

Pada hari ketujuh belas saya tidak dimasukkan kedalam tim, namun membentuk tim baru sendiri untuk melanjutkan Jogja Project. Pada Jogja Project, masih perlu di revisi beberapa bagian yaitu bagian antrian diminta untuk menggunakan Rabbit MQ / Spring Kafka dan menambahkan *Login LDAP* sesuai keinginan user setelah saya ujian pada tanggal 14 Agustus. Kemudian saya bertemu lagi dengan user untuk memastikan lagi teknologi apa yang akan di pakai dan darimana saya bisa mendapat kan referensi untuk melakukan integrase dengan *services* yang ada di dalam kantor. Karena untuk melakukan *login* menggunakan Central Authentication Security (CAS) yang merupakan *service internal* dan pengiriman email menggunakan X-Message yang merupakan *service internal*.

#### **Kamis, 23 Agustus 2018 (08.00 – 21.00).**

Pada hari kedelapan belas saya mempelajari materi Rabbit MQ. Rabbit MQ ini merupakan salah satu message broker untuk melakukan manajemen antrian pesan yang menggunakan Advanced Messaging Queuing Procotol (AMQP). Rabbit MQ ini akan

menggantikan peran antrian *Job Generate* yang menggunakan status basis data yang saat ini digunakan pada sistem Jogja Project. Pada hari ini saya mencoba melakukan penginstalan RabbitMQ pada komputer saya dan membaca – baca dahulu tentang RabbitMQ juga menonton beberapa video dari youtube untuk memastikan saya menggunakan RabbitMQ dengan benar. Beberapa yang saya dapatkan hari ini ialah penggunaan yang cocok untuk Jogja Project adalah dengan menggunakan direct message yang menggunakan route untuk menghubungkan secara langsung antara *producer* dan *consumer*. Pada RabbitMQ *producer* merupakan *client* yang melakukan mengirimkan pesan sedangkan *consumer* merupakan *client* yang mendapatkan pesan dari *producer*.

**Jum'at, 24 Agustus 2018 (18.00 – 20.00)**

Pada hari kesembilan belas saya melanjutkan mempelajari Rabbit MQ. Saya juga mulai memasukkan Rabbit MQ ke dalam Jogja Project. Pada hal ini saya hanya membuat dua kelas baru yaitu *JobGenerateProducer* dan *JobGenerateConsumer*. Metode yang saya gunakan untuk melakukan eksekusi pada RabbitMQ adalah dengan mengirimkan id dari *job generate* yang sudah disimpan di dalam basis data. Kemudian *consumer* mendapatkan id yang diberikan oleh *producer* yang kemudian mengeksekusi hingga *job* selesai baru akan melakukan *consume* lagi.

**Senin, 27 Agustus 2018 (08.00 - 20.00)**

Pada hari kedua puluh saya mengerjakan Central Authentication Security (CAS). CAS ini digunakan untuk melakukan login pada sistem yang terdapat di Blibli. Untuk menggunakannya dibutuhkan pengaturan pada maven (*dependency tools*) untuk melakukan akses pada repository yang dimiliki oleh Blibli, karena CAS ini harus dipasangkan pada *project* dengan menambahkan *dependency com.gdn.cas* pada maven. Setelah menambahkan CAS pada *project*, saya tidak langsung melakukan implementasi, namun saya mempelajari dahulu dari kode *dependency* CAS tersebut.

Selama seharian ini saya masih belajar untuk melakukan konfigurasi CAS dengan memberikan hak akses *path* sesuai dengan peran yang dimiliki. Selain memberikan hak akses, juga melakukan konfigurasi *properties* seperti *host*, *expired time*, *token*, *system name* dan mode yang digunakan untuk melakukan koneksi dengan

*environment* CAS ini berada, karena Blibli sendiri memiliki 4 *environment* yaitu UAT-A, UAT-B, DEV1 dan DEV2. Saat ini Jogja Project terhubung dengan UAT-A namun saya masih belum memiliki informasi tentang *environment* tersebut dengan baik. Jadi saya mempelajari sistem lain yang menggunakan CAS.

**Selasa, 28 Agustus 2018 (08.00 - 18.00)**

Pada hari kedua puluh satu saya menunggu pembuatan user pada sistem CAS, yang hak akses nya belum saya dapati dari DevOps. Selagi menunggu, mentor meminta untuk mengganti nama project, karena nama Jogja sudah dipakai oleh sistem lain pada Blibli. Oleh karena itu saya hanya melakukan *browsing* nama – nama kota yang sekiranya berhubungan dengan sistem saya. Kemudian saya menemukan beberapa nama kota yang cukup berhubungan dengan dengan sistem ini, yaitu Reporoa(Report) dan Jasin(Jasper Report).

**Rabu, 29 Agustus 2018 (08.00 - 18.00)**

Pada hari kedua puluh dua saya masih menunggu pembuatan user pada sistem CAS, yang hak akses nya belum saya dapati dari DevOps. Saya mengajukan kedua nama kota yang saya dapatkan kemarin dan ditolak. Akhirnya saya melakukan *browsing* kembali, dan saya mendapatkan nama kota di Jerman yaitu Regen (Report Generator). Setelah itu saya segera laporkan kepada mentor dan diterima.

**Kamis, 30 Agustus 2018 (08.00 - 20.00)**

Pada hari kedua puluh tiga saya dipertemukan dengan bagian infrastruktur untuk membahas mengenai hak akses untuk membuat pengguna pada CAS. Setelah mendapatkan hak akses saya membuat pengguna baru pada *environment* sistem saya, yaitu UAT-A. Kemudian saya mencoba – coba untuk mempelajari tentang penggunaan sistem CAS dari sisi developer/administrator. Dan pada saat itu saya mencoba untuk melakukan koneksi secara langsung dan melakukan *login*. Saat awal *login*, sistem saya tidak melakukan *redirect* dengan benar, kemudian saya mencoba melakukan cek pada konfigurasi properties dan mencoba mengubahnya dari *redirect* ke dalam back-end menjadi *redirect* ke front-end. Setelah di jalankan kembali, sudah bekerja dengan

baik. Walaupun sudah berjalan dengan baik, saya sampai saat ini saya masih bingung dengan cara kerja didalam CAS itu sendiri dan belum memahami apa yang dimaksud.

**Jum'at 31 Agustus 2018 (08.00 - 20.00)**

Pada hari kedua puluh empat saya masih mencoba – coba memahami sistem CAS yang ada pada Blibli. Hal ini amat saya bingung karena kode yang dibuat masih belum bisa diterima oleh nalar saya. Saya pun mencoba menanyakan pada beberapa rekan di Blibli, namun saya tidak mendapatkan jawaban yang pasti. Oleh karena itu saya mencoba melihat project lain yang menggunakan sistem CAS yang sama. Sayang sekali setelah saya mencari, ternyata kebanyakan masih menggunakan sistem CAS yang lama. Sedangkan yang saya gunakan ternyata merupakan versi dua. Akhirnya saya abaikan dulu masalah ini dan bersiap untuk mendapatkan tugas selanjutnya.

**Senin 3, September 2018 (08.00 – 20.00).**

Pada hari kedua puluh lima saya menambahkan swagger pada Regen Project yang saya buat. Swagger adalah *plugin* yang digunakan untuk membantu developer untuk melihat RESTful API yang terdapat pada sistem. Swagger ini akan menampilkan *path request* yang tersedia dalam sistem sehingga menjadi mudah untuk digunakan oleh *developer*. Dengan menggunakan swagger, *developer* tidak perlu lagi menggunakan POST Man atau semacamnya karena fitur tersebut sudah didapatkan didalam Swagger. Untuk melakukan akses terhadap swagger hanya perlu masuk ke dalam halaman swagger dengan path “<http://host/swagger-ui.html>”. Untuk menggunakannya maka pertama – tama memilih kelas yang memiliki kumpulan *request*, setiap *request* ini biasanya merepresentasikan satu *method* pada kelas tersebut. Setelah memilih kelas, kemudian memilih *request* yang ingin dijalankan, setelah itu akan disediakan tempat untuk memasukan *parameter/request body/path variable* apabila ada. Apabila sudah di isi kemudian tinggal menekan tombol *Try it out !*. Setelah itu dibawahnya akan menampilkan hasil dari *request*.

**Selasa 4 - 5, September 2018 (08.00 – 19.00)**

Pada hari kedua puluh enam dan kedua puluh tujuh saya menunggu tugas X-Message, karena saya sendiri belum mengerti mengenai X-Message dan masih takut/ragu untuk langsung mencoba.

**Kamis 6, September 2018 (09.00 – 19.00)**

Pada hari kedua puluh delapan saya mendapatkan tugas X-Message yang berguna untuk mengirimkan pesan menggunakan email, sms dan whatsapp. Namun pada saat ini saya hanya menggunakan email saja. Dan tidak terlalu banyak yang saya lakukan pada hari ini, hanya mencari – cari pengaturan *environment* X-Message pada UAT-A. Kemudian saya mendapatkan pengaturan tersebut di dalam *config tracker* milik Blibli.

Kemudian saya menambahkan *dependency* X-Message ke dalam maven Regen Project. Setelah itu saya membukanya dan mempelajari setiap kelas yang ada. Di dalam sana terdapat kelas tiga kelas utama untuk mengirimkan sms, email dan whatsapp. Setiap macam nya menggunakan *queuing* untuk mengirimkan pesan menggunakan RabbitMQ.

**Jum'at 7 & 10, September 2018 (09.00 – 20.00)**

Pada hari kedua puluh sembilan dan ketiga puluh saya memulai melakukan implementasi X-Message menggunakan RabbitMQ untuk melakukan *queue* pengiriman email yang berada pada *environment* UAT-A. Saya menambahkan implementasi *service* pada Regen Project. Namun saya masih tidak mengetahui basis data yang digunakan untuk melakukan penyimpanan template body dan setelah saya tanyakan kepada salah satu *user* saya (karena kebetulan mentor saya sedang cuti) bagaimana cara menambahkan *template* tersebut. Beliau mengatakan bahwa hanya *Database Admin* (DBA) yang dapat menambahkan / membuatnya. Oleh karena itu saya menghubungi mentor saya, namun tidak ada jawaban sehingga saya menunda tugas ini.

### **Rabu 12, September 2018 (09.00 – 19.00)**

Pada hari ketiga puluh satu saya melanjutkan melakukan implementasi x-message dengan menambahkan *template* pengiriman e-mail ke basis data yang ternyata untuk melakukan akses pada basis data *template* dapat diakses langsung oleh *developer*. Segera pada saat itu saya menambahkan *template* pada basis data tersebut. Setelah itu saya mencari konfigurasi untuk *properties* yang dibutuhkan seperti koneksi ke RabbitMQ x-message dan format pengiriman email. Pada config tracker terdapat konfigurasi untuk x-message, namun *queue* dan *exchange* yang digunakan masih tidak tepat. Kemudian saya menanyakan kepada rekan saya kembali dan mendapatkan nama *queue* dan *exchange* yang tepat. Setelah itu saya mencoba mengirimkan email dan berhasil di implementasikan dengan beberapa *error* kecil yaitu format unduhan belum csv.

### **Kamis 13, September 2018 (09 – 19.00)**

Pada hari ketiga puluh dua saya membetulkan *error* kecil yaitu mengubah *format download* menjadi csv. Pada bagian ini ternyata saya hanya perlu mengubah *produces* dari “*application/json*” menjadi “*text/csv*”, setelah itu file berhasil di unduh berupa csv. Serta proyek ini segera dilakukan *deployment* pada *server* dan hanya perlu menunggu mentor saya membuat dokumen *request* untuk *deploy* pada *server*.

## **2.2 Hasil Pekerjaan Secara Umum**

Berdasarkan yang sudah saya kerjakan yaitu Regen Project (nama terakhir untuk sistem, sebelumnya Sleman / Jogja). Hasil yang saya kerjakan adalah sebuah sistem Report Generator. Yang dimana sistem ini melakukan eksekusi terhadap suatu query untuk menghasilkan suatu berkas berekstensi \*.csv. Pada sistem ini saya lebih fokus pada Back-End yang dimana saya membuat bisnis logik dari sistem Regen ini.

Teknologi inti yang digunakan untuk melakukan proses menghasilkan laporan ini adalah Spring Batch untuk melakukan *Batch Processing*. Hal ini menjadikan sistem dapat melakukan eksekusi dengan cepat karena melakukan eksekusi secara paralel, selain itu juga dapat menghemat memori karena saat melakukan eksekusi dilakukan secara kelompok, dengan kata lain data yang sangat banyak dipotong – potong menjadi beberapa bagian untuk di eksekusi.

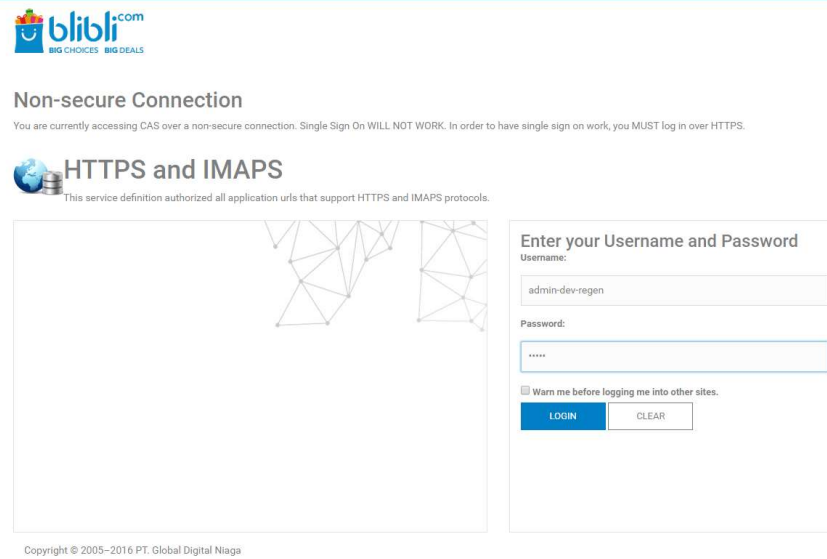


Selain itu untuk mendukung kinerja sistem ini, diberi kemampuan untuk melakukan antrian terhadap setiap laporan yang di eksekusi. Teknologi yang digunakan untuk melakukan antrian ini adalah RabbitMQ. RabbitMQ akan melakukan eksekusi atau lebih sering di sebut dengan *consumer*, setiap ada yang melakukan *request* atau lebih sering di sebut dengan *producer*. Sehingga secara singkat kerja RabbitMQ ini seperti orang yang memberikan makan dan orang yang memakan, setiap ada makanan maka akan dimakan.

Pada sisi *internal* menggunakan *Central Authentication Security (CAS)* sebagai jalur sistem untuk melakukan autentikasi. CAS ini tersedia pada internal blibli, sehingga tinggal melakukan implementasi saja ke Regen. Bagian yang sulit dari melakukan implementasi CAS ialah minimnya dokumentasi dan hanya ada referensi berbentuk project yang sudah ada sehingga sulit untuk dipelajari.

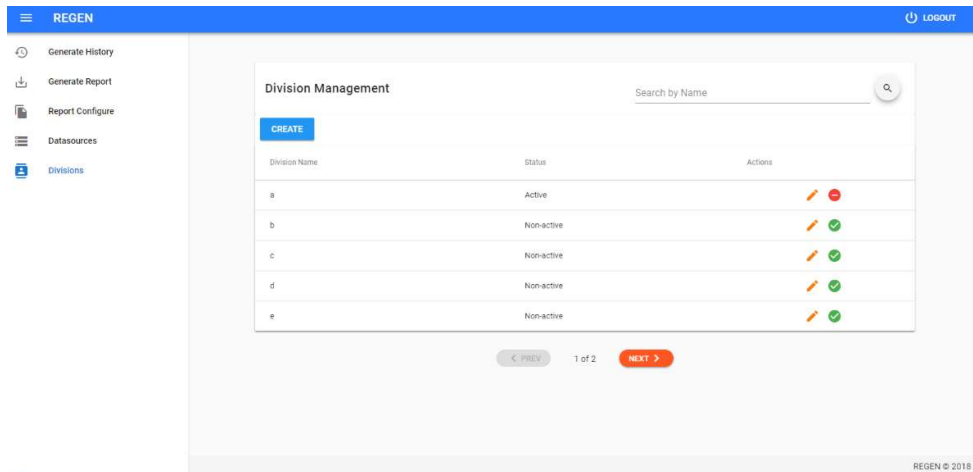
Selanjutnya ada X-Message yang merupakan *services* yang tersedia pada Blibli yang digunakan sebagai pengiriman email, sms, whatsapp dan lain – lain. Pada X-*message* ini saya hanya menggunakan pada bagian pengiriman e-mail saja. *Service* ini saya gunakan untuk memberikan notifikasi pengguna melalui email mengenai status (berhasil/gagal) laporan yang di hasilkan oleh sistem. Apabila berhasil maka pengguna dapat melakukan unduh berkas melau email, sedangkan apabila gagal maka pengguna hanya diberikan notifikasi bahwa laporan gagal dihasilkan.

## 2.3 Bukti Hasil Pekerjaan



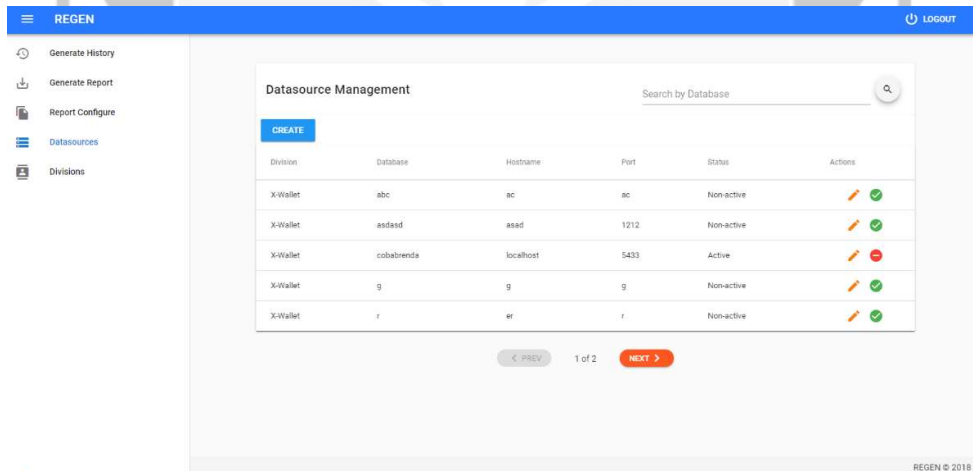
**Gambar 8. Halaman login sistem Blibli.com**

Halaman diatas merupakan halaman utama untuk melakukan akses pada sistem yang terdapat pada internal Blibli. Karena semua sistem melakukan login pada satu satu tempat ini, oleh sebab itulah disebut dengan nama Centra Authentication Security (CAS). Untuk semua pengguna pun biasanya memiliki akun yang sama, untuk melakukan *login*. Jadi setiap pengguna bisa memiliki satu akun yang sama.



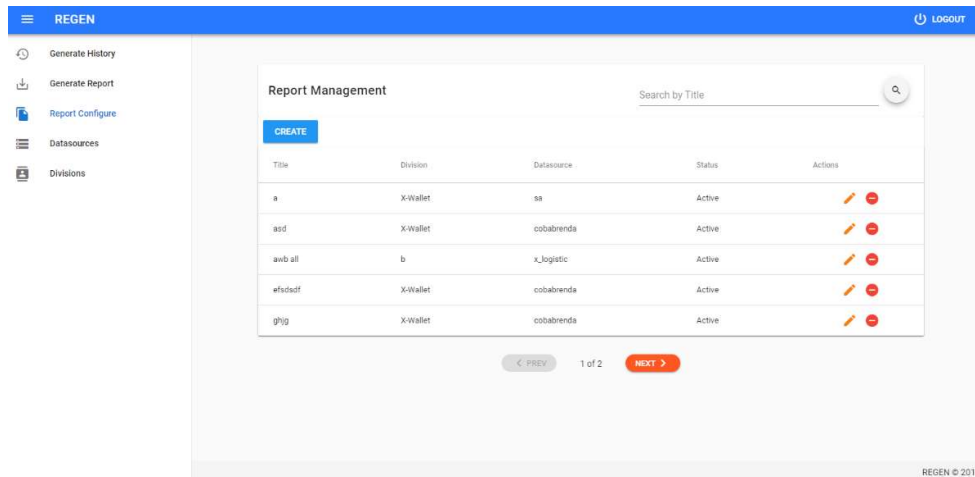
*Gambar 9. Halaman Manajemen Divisi*

Halama ini adalah halaman yang digunakan untuk melakukan manajemen divisi. Divisi yang dimaksud adalah departemen pada Blibli seperti finansial, pemasaran, bisnis dan lain – lain. Pada divisi ini hanya digunakan sebagai penanda sumber data dan laporan tersebut untuk divisi mana.



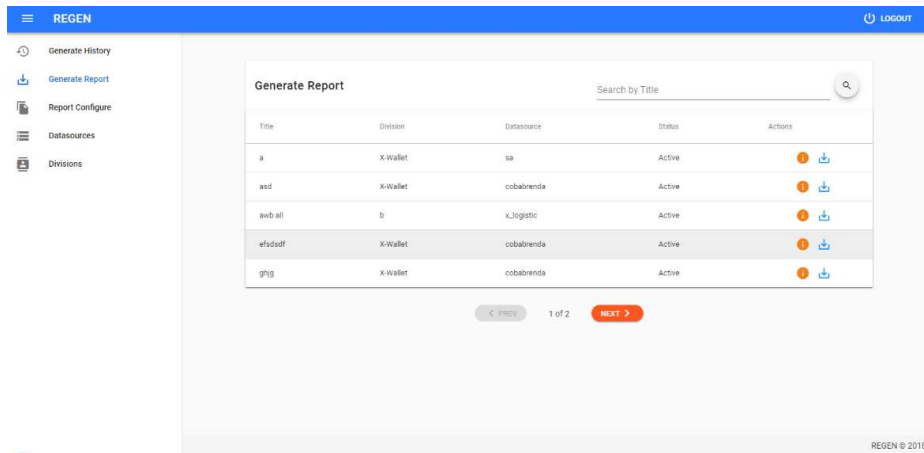
*Gambar 10. Halaman Manajemen Sumber Data*

Halaman ini adalah halaman yang digunakan untuk melakukan manajemen sumber data. Sumber data yang dimaksud adalah basis data yang digunakan sehingga isi yang diperlukan sesuai dengan pengaturan sumber data. Sumber data ini berguna sebagai penghubung untuk melakukan penghasilan berkas.



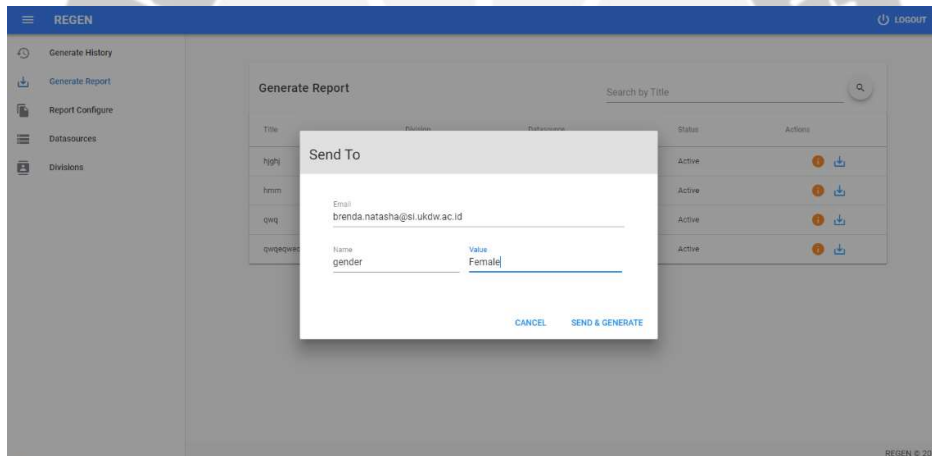
**Gambar 11. Halaman Manajemen Laporan**

Halaman ini adalah halaman yang digunakan untuk melakukan manajemen laporan. Laporan akan dibuat berdasarkan *query* sehingga *developer* nantinya wajib memasukkan *query* untuk dilakukan eksekusi serta menambahkan parameter seperti *query* biasanya menggunakan “?”. Kemudian pada pemberian parameter juga akan diatur nama parameter dan urutan parameter tersebut supaya bila lebih dari satu parameter dapat diatur dengan mudah.



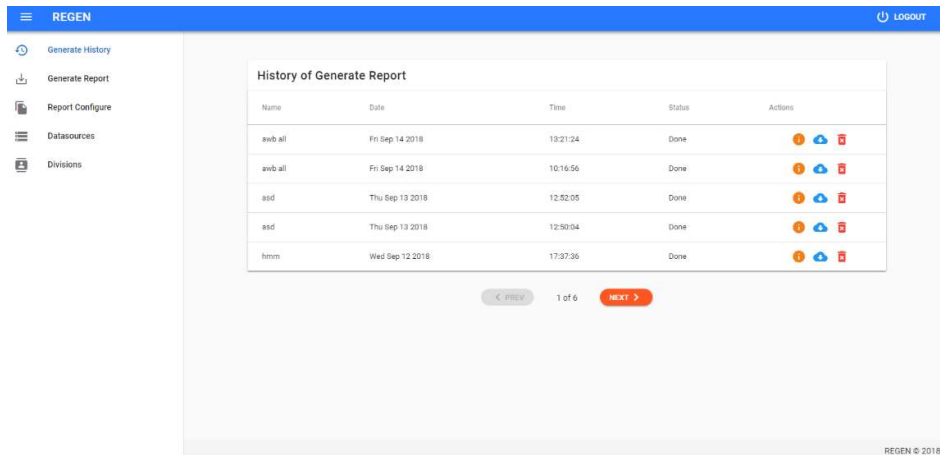
**Gambar 12. Halaman Penghasil Laporan**

Halaman ini adalah halaman yang digunakan untuk menghasilkan berkas dengan menekan tombol “Generate” dan juga memberikan informasi tentang parameter apa yang perlu dimasukkan.



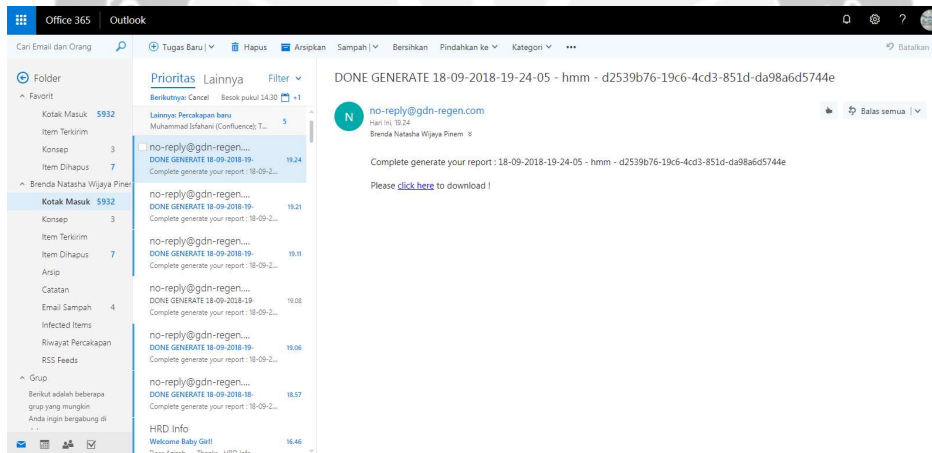
**Gambar 13. Tampilan Sebelum Eksekusi Laporan**

Tampilan ini adalah tampilan yang digunakan untuk menghasilkan berkas dan mengirimkan melalui email. Berkas akan dikirimkan sesuai dengan inputan email pada formulir. Kemudian dilanjutkan dengan pengisian parameter apabila ada parameter yang perlu dimasukkan, sebelah kiri adalah nama parameter dan sebelah kanan adalah value parameter tersebut.



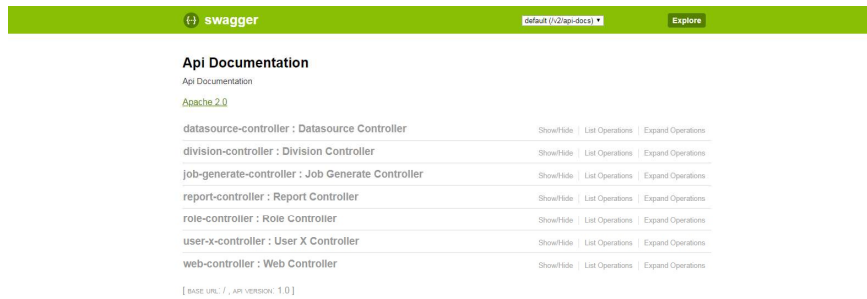
**Gambar 14. Halaman Manajemen Riwayat Laporan**

Halaman ini digunakan untuk mengunduh hasil laporan dan menampilkan parameter yang dimasukkan saat melakukan eksekusi laporan. Hasil laporan dapat diunduh dan nantinya akan berformat \*.csv. Kemudian untuk admin juga nanti dapat melakukan penghapusan pada hasil laporan yang sudah dieksekusi.



**Gambar 15. Halaman Email Outlook : Email Hasil Laporan**

Halaman ini adalah halaman email outlook milik *Microsoft*. Pada email akan dikirimkan hasil laporan yang sudah dieksekusi dan dapat di untuk dengan menekan tombol “*click here*” pada email. Setelah menekan tombol maka secara otomatis akan mengunduh berkas ke dalam bentuk \*.csv.



**Gambar 16. Halaman Swagger**

Halaman ini adalah halaman untuk menggunakan *plugin* swagger. Swagger ini berguna dan membantu developer supaya dapat mencoba api yang sudah dibuat. Swagger mendukung komunikasi menggunakan HTTP *Request* seperti GET, POST, PUT, DELETE dan lain – lain serta *request body* dan parameter.



**Gambar 17. Halaman Logout**

Halaman ini adalah halaman *logout* yang disediakan oleh CAS. Apabila halaman logout menampilkan seperti ini maka sistem sudah berhasil melakukan logout dari Regen Project maupun dari CAS sendiri.

## **BAB III**

### **HASIL PEMBELAJARAN**

#### **3.1 Manfaat Kerja Praktek**

Selama menjalankan kerja praktek di Blibli, saya rasakan banyak hal yang bermanfaat bagi diri saya. Ada tiga aspek manfaat yang saya dapatkan, yaitu sikap, kerja sama dan teknologi. Saya menjadi mengerti bagaimana sikap yang seharusnya saya tanggapi saat melakukan aktifitas di dalam kantor, karena orang – orang datang untuk mengerjakan pekerjaan sehingga lebih sering focus dengan pekerjaannya. Namun bersantai juga penting, oleh karena itu saat siang hari kami beristirahat, bercanda, olahraga ataupun membaca buku. Selain itu juga saat melakukan percakapan dengan atasan atau orang lebih di hormati, di Blibli tidak terlalu banyak orang yang benar – benar seperti bos, di Blibli lebih sering berkomunikasi dengan santai namun tetap sopan. Dari situ saya jadi mengerti kapan saya harus bersikap. Kemudian kerja sama, hal ini sangat terasa karena di Blibli sendiri memiliki tim – tim yang berguna untuk mengembangkan suatu sistem. Saya mendapatkan tim yang merupakan mentor saya dan dua orang lainnya yaitu Brenda dan Nadya. Dari kerja sama ini saya mempelajari bagaimana cara menyatukan pemikiran supaya tidak terjadi *miss* saat sistem sudah jadi. Kemudian teknologi, pada bagian ini tentu saja saya mendapatkan banyak pengetahuan tentang teknologi baru yang belum tentu saya dapatkan di kampus. Serta tantangan untuk mempelajari hal baru yang benar – benar masih awam tentang hal tersebut.

#### **3.2 Penerapan Ilmu dalam Kerja Praktek**

Ilmu yang dapat saya terapkan dalam kerja praktek ini cukup banyak di antaranya adalah materi tentang pemrograman berorientasi objek dan pemrograman web. Bahasa yang digunakan untuk melakukan pengembangan aplikasi yaitu Java sebagai back-end dan HTML, CSS dan JS sebagai front-end. Dari dasar – dasar pelajaran yang di dapat saat kuliah sungguh diterapkan, namun dalam versi yang lebih profesional yaitu dengan menggunakan *framework, boiler plate dan plug-ins*.

Pemrograman Berorientasi Objek (PBO) adalah mata kuliah yang mempelajari tentang data yang dipandang sebagai objek – objek untuk menampung nilai – nilai nya. Bahasa pemrograman yang menggunakan PBO salah satu contohnya adalah Java. Mulai dari kelas, objek sampai konektivitas ke dalam basis data. Hampir seluruh materi



yang saya pelajari di kampus, digunakan pada saat membuat Regen Project. Pada saat melakukan pengerjaan *project*, hal yang saya dapatkan lebih ialah mengenai *design pattern*, *properties configuration* dan sebagainya.

Mata kuliah Basis Data juga diterapkan pada saat saya mengerjakan Regen Project dengan menggunakan PostgreSQL. Penerapan yang digunakan tidak jauh berbeda dengan yang di ajarkan saat berada di kampus. Karena PostgreSQL merupakan basis data yang menggunakan SQL atau yang kita sebut dengan relasi. Namun pada kantor saya juga melihat ada beberapa perbedaan seperti penggunaan MongoDB yang merupakan NoSQL, dimana setiap tabel tidak berhubungan sehingga bentuknya menjadi sangat sederhana dan flat seperti menyimpan *Javascript Object Notation* (JSON).



## **BAB IV**

### **KESIMPULAN**

Kerja praktek tidak hanya datang ke dalam suatu instansi dan asal mengerjakan tugas yang di dapatkan disana. Namun juga sebagai tempat kita memahami dunia kerja orang – orang yang bergerak dalam bidang Teknologi Informasi (TI). Selain itu juga untuk lebih mengenal sekitar seperti pergaulan dan kerja sama tim yang belum tentu didapatkan saat berkuliah di kampus. Teknologi yang digunakan oleh instansi pun menjadi pengetahuan kita akan luasnya bidang teknologi informasi yang luar biasa. Karena banyak sekali yang dapat kita temui pada instansi tersebut.

Tantangan bagi orang yang bergerak pada bidang teknologi informasi untuk bisa mengikuti perkembangan jaman, yang dapat merubah gaya pengkodean menjadi lebih mudah dibaca orang lain atau yang disebut dengan *clean code*. Hal ini menjadi salah satu bagian dari aspek kerja sama, supaya dapat sistem yang dibangun dapat dilanjutkan oleh orang lain dengan mudah oleh orang lain.

Kerja sama tim akan sangat penting ketika mengerjakan project, oleh karena itu perlu melakukan sesi berbagi tentang apa yang dikerjakan. Contoh saja menggunakan daily scrum akan membantu kerja project karena setiap hari tim berbagi akan apa yang sedang dikerjakan dan kesulitan apa yang di alami sehingga rekan tim dapat menyalurkan solusi untuk menyelesaikan masalah tersebut. Apabila hubungan antar individu di dalam tim tidak ada masalah, maka kinerja tim akan menjadi lebih baik.

## LAMPIRAN

```
package com.satyadara.regen.config;

import lombok.Getter;
import lombok.Setter;
import org.springframework.batch.core.Job;
import org.springframework.batch.core.Step;
import org.springframework.batch.core.configuration.annotation.JobBuilderFactory;
import org.springframework.batch.core.configuration.annotation.StepBuilderFactory;
import org.springframework.batch.item.database.JdbcCursorItemReader;
import org.springframework.batch.item.file.FlatFileItemWriter;
import org.springframework.batch.item.file.transform.PassThroughLineAggregator;
import org.springframework.beans.factory.annotation.Autowired;
import org.springframework.beans.factory.annotation.Value;
import org.springframework.context.annotation.Configuration;
import org.springframework.core.io.FileSystemResource;

import javax.sql.DataSource;
import java.sql.ResultSet;
import java.sql.ResultSetMetaData;
import java.sql.SQLException;
import java.util.UUID;

@Configuration
public class JobConfiguration {
    @Autowired
    public JobBuilderFactory jobBuilderFactory;
    @Autowired
    public StepBuilderFactory stepBuilderFactory;
    @Getter
    @Setter
    private int chunkSize;
    @Getter
    @Setter
    private DataSource dataSource;
    @Getter
    @Setter
    private String sql;
    @Getter
    @Setter
    private String fileName;
    @Value("${regen.generate.path-file}")
    public String pathFile;
    @Value("${regen.generate.separator}")
    public String separator;

    public JdbcCursorItemReader<Object> reader() {
        JdbcCursorItemReader<Object> reader = new JdbcCursorItemReader<>();
        reader.setSql(getSql());
        reader.setRowMapper((resultSet, row) -> (row == 1) ?
            recursiveRowOfColumnName(resultSet, 1) +
recursiveRowOfData(resultSet, 1) :
            recursiveRowOfData(resultSet, 1));
        reader.setDataSource(getDataSource());
        reader.setFetchSize(getChunkSize());

        return reader;
    }

    public FlatFileItemWriter<Object> writer() throws Exception {
        FlatFileItemWriter<Object> writer = new FlatFileItemWriter<>();
        writer.setLineAggregator(new PassThroughLineAggregator<>()); //will run
toString
        String outputPath = pathFile + getFileName() + ".csv";
        writer.setResource(new FileSystemResource(outputPath));
        writer.afterPropertiesSet();
    }
}
```

```

        return writer;
    }

    public Step steps() throws Exception {
        return stepBuilderFactory.get(UUID.randomUUID().toString())
            .<Object, Object>chunk(getChunkSize())
            .reader(reader())
            .writer(writer())
            .build();
    }

    public Job myJob() throws Exception {
        return jobBuilderFactory.get(UUID.randomUUID().toString())
            .start(steps())
            .build();
    }

    public String recursiveRowOfColumnName(ResultSet resultSet, int column) throws
    SQLException {
        ResultSetMetaData metaData = resultSet.getMetaData();
        String columnName = metaData.getColumnName(column).replace(',', ' ');
        return (metaData.getColumnCount() == column) ?
            columnName + "\n" : columnName + separator +
            recursiveRowOfColumnName(resultSet, column + 1);
    }

    public String recursiveRowOfData(ResultSet resultSet, int column) throws
    SQLException {
        String columnResult = resultSet.getObject(column).toString().replace(',', ' ');
        return (resultSet.getMetaData().getColumnCount() == column) ?
            columnResult : columnResult + separator +
            recursiveRowOfData(resultSet, column + 1);
    }
}

```

Kode inti program yang digunakan untuk melakukan batch processing.