



### **Hak cipta dan penggunaan kembali:**

Lisensi ini mengizinkan setiap orang untuk menggubah, memperbaiki, dan membuat ciptaan turunan bukan untuk kepentingan komersial, selama anda mencantumkan nama penulis dan melisensikan ciptaan turunan dengan syarat yang serupa dengan ciptaan asli.

### **Copyright and reuse:**

This license lets you remix, tweak, and build upon work non-commercially, as long as you credit the origin creator and license it on your new creations under the identical terms.

## BAB III

### PELAKSANAAN KERJA MAGANG

#### 3.1 Kedudukan dan Koordinasi

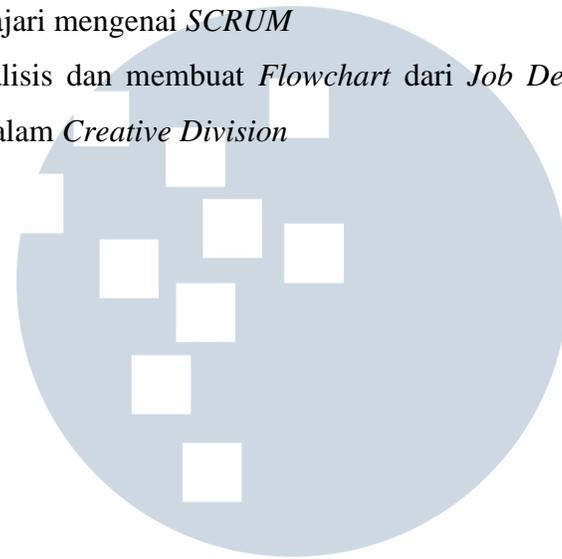
Pada kerja magang yang dilakukan di PT Raksasa Laju Lintang, penulis bekerja sebagai *Business Analyst*. *Business Analyst* termasuk ke dalam tim *Developer* yang tergabung ke dalam divisi tim kreatif atau *Creative Division* perusahaan. Adapun bagian-bagian lain yang termasuk ke dalam *Creative Division* adalah *Web Content*, *News*, *Search Engine Optimization*, *Graphic Design*, dan *Creative Multimedia*. *Creative Division* adalah divisi dalam perusahaan yang bertugas untuk menjalankan dan mengelola sistem dan website *e-commerce* untuk kegiatan operasional perusahaan sehari-hari. Penulis sebagai *Business Analyst* mempunyai *Job Description* untuk membantu tim *Developer* dalam mengembangkan sistem yang akan digunakan dengan membantu dalam melakukan analisis mengenai beberapa kebutuhan sistem.

#### 3.2 Tugas Yang Dilakukan

Selama melakukan kerja magang selama 2 bulan, berikut ini adalah tugas-tugas yang dikerjakan oleh penulis sebagai *Business Analyst* :

- Membuat *Entity Relationship Diagram* dari *database* pada sistem perusahaan yang sedang berjalan / sedang digunakan
- Membuat *Flowchart* dari proses order yang dilakukan customer
- Membuat *User Manual* dari *Dashboard Admin* yang ada pada sistem

- Mempelajari, menganalisis, dan tes penggunaan *CMS* berbasis *framework* *Laravel* untuk *E-Commerce*
- Menganalisis kebutuhan serta membuat *Entity Relationship Diagram* untuk tabel *vendor\_verification*
- Mempelajari mengenai *SCRUM*
- Menganalisis dan membuat *Flowchart* dari *Job Description* masing-masing bagian dalam *Creative Division*



UMN  
UNIVERSITAS  
MULTIMEDIA  
NUSANTARA

### 3.3 Uraian Pelaksanaan Kerja Magang

#### 3.3.1 Time Table Pelaksanaan Kerja Magang

Pekerjaan	Minggu ke-							
	1	2	3	4	5	6	7	8
Membuat <i>ERD</i> dari sistem yang sedang berjalan, Membuat <i>Flowchart</i> dari proses order <i>Customer</i>								
Mempelajari <i>OctoberCMS</i> dan penggunaannya untuk <i>E-Commerce</i>								
Mempelajari mengenai <i>CMS</i> yang menggunakan <i>Laravel framework</i>								
Mempelajari dan tes implementasi penggunaan <i>PyroCMS</i> untuk <i>E-Commerce</i>								
Menganalisis kebutuhan data untuk proses <i>vendor verification</i>								
Membuat <i>ERD</i> dan <i>User Manual</i> untuk tabel data <i>vendor verification</i>								
Mempelajari mengenai <i>SDLC</i> dan <i>SCRUM</i>								
Mempelajari dan menyusun <i>FlowChart</i> untuk <i>Job Description</i> tiap divisi								

Tabel di atas diambil dari bagian 1.3.3, dengan tujuan untuk memudahkan pembacaan *timeline* secara garis besar. Adapun *timeline* di atas akan dijelaskan lebih lanjut di halaman berikutnya.

Minggu 1: Penulis membuat *ERD* dari daftar tabel yang ada pada database sistem yang sedang berjalan, agar dapat memudahkan dilakukannya analisis untuk pengembangan sistem. Adapun yang diperlukan dalam *ERD* tersebut adalah relasi antar tabel yang ada, sehingga dapat diketahui kira-kira apa yang kurang , ataupun apa yang tidak diperlukan dari tabel yang ada. Serta penulis juga diminta untuk menyusun *flowchart* dari proses order yang dilakukan oleh *Corporate Customer*.

Minggu 2: Ralali berencana untuk mengganti sistem yang ada, maka penulis diminta untuk mempelajari salah satu *CMS* yang baru dan berbasis *framework Laravel*, yaitu OctoberCMS. Penulis mempelajari cara instalasi dan cara menggunakannya, serta cara penggunaan OctoberCMS tersebut untuk *E-Commerce*. Ternyata OctoberCMS masih belum memiliki *template* khusus untuk *E-Commerce*.

Minggu 3 & 4: Karena OctoberCMS masih baru dan belum memiliki *template* khusus untuk *E-Commerce*, maka penulis diminta untuk melakukan *research* mengenai jenis-jenis *CMS* yang menggunakan *framework Laravel*. Maka penulis menemukan salah satunya yaitu PyroCMS, sehingga penulis mempelajari *CMS* tersebut beserta penggunaannya untuk *E-Commerce*. Ternyata PyroCMS memiliki *platform* yang dapat membantu pembuatan *E-Commerce*, yang dinamakan FireSale.

Minggu 5 & 6: Ralali menginginkan bahwa setiap *vendor* yang akan atau telah menjalin kerjasama dengan Ralali, akan dibuatkan tabel dalam *database* untuk menyimpan data dokumen dan surat-surat untuk verifikasi. Maka penulis diminta untuk melakukan wawancara dengan divisi Buyer, untuk mendapatkan data apa saja yang diperlukan, dan didapatkan data dokumen verifikasi yang diperlukan yaitu dokumen MOU, NPWP, SIUP, Surat Keterangan Domisili, dan Surat Penunjukan. Maka penulis pun menyusun *ERD* serta *User Manualnya* untuk diserahkan kepada tim Developer.

Minggu 7: Pada minggu ke 7 ini, penulis diminta untuk mempelajari mengenai *System Development Life Cycle* (SDLC), dan *SCRUM*. Setelah dipelajari, penulis mengetahui bahwa SCRUM merupakan salah satu bagian dari model dalam SDLC, yaitu *Agile Model*. Setelah itu, penulis melakukan *sharing* kepada tim Developer mengenai apa yang telah penulis pelajari.

Minggu 8: Ralali akan segera membuka kantor cabang yang baru, dan tentu saja akan meng-*hire* karyawan baru pula. Oleh sebab itu, untuk mempermudah pembelajaran karyawan baru mengenai cara melakukan kegiatan operasional sehari-hari, maka penulis diminta untuk melakukan wawancara pada masing-masing divisi untuk mendapatkan *Job Description* masing-masing. Setelah itu penulis menyusun *Flowchart* dari *Job Description* yang ada, sehingga dapat digunakan sebagai pedoman kerja di kantor yang baru nanti.



### 3.3.2 Pembuatan ERD Sistem Yang Sedang Berjalan

Ini adalah tugas pertama yang dikerjakan oleh penulis saat magang. Tim *Developer* berencana untuk mengganti sistem yang sedang berjalan menjadi sistem yang baru di masa mendatang, oleh karena itu diperlukan analisis mengenai tabel-tabel di dalam *database* yang digunakan saat ini. Setelah mengambil daftar tabel yang ada di dalam sistem, ternyata hasil *extract* data nya hanya berupa list tabel-tabel yang ada beserta strukturnya, tanpa diketahui *relationship* antar tabel. Oleh sebab itu, maka penulis diminta untuk menganalisis *relationship* antar tabel dari semua tabel yang ada, menganalisis mana yang merupakan *primary key* dan *foreign key*, serta menampilkan hasil ERD nya.

Untuk melakukan hal ini, penulis menggunakan bantuan *software* XAMPP beserta database PHPMyAdmin yang merupakan bawaannya. Dari daftar tabel yang ada, penulis membuatnya beserta strukturnya ke dalam PHPMyAdmin. Setelah seluruh tabel selesai dibuat, penulis melakukan analisis tabel mana saja yang akan saling berhubungan, dengan menganalisis *primary key* dan *foreign key* nya. Ini membutuhkan waktu yang cukup lama karena dikerjakan secara manual. Setelah semua *primary key* dan *foreign key* dibuat, maka penulis menampilkan ERD nya melalui menu *Designer View* pada XAMPP.

Rancangan ERD yang dibuat tidak dapat dilampirkan, dengan tujuan kerahasiaan informasi perusahaan.

### 3.3.3 Flowchart Proses Order Customer

Penulis diminta untuk menganalisis proses bisnis perusahaan, yaitu proses order yang dilakukan oleh customer kepada perusahaan, terutama *Corporate customer* karena PT Raksasa Laju Lintang merupakan perusahaan B2B sehingga sebagian besar *customer*-nya merupakan *Corporate* atau perusahaan lain. Setelah dianalisis, penulis membuat *flowchart* proses order tersebut untuk dapat dianalisis lebih lanjut mengenai proses yang kurang tepat dan harus diubah. Adapun proses tersebut secara umum adalah :

- *Customer* mengirimkan *Inquiry* produk kepada perusahaan
- Perusahaan akan membalasnya dengan memberikan *Quotation* (daftar barang, jumlah, serta harga yang ditawarkan) untuk *Inquiry* tersebut
- Setelah menerima *Quotation* dan sepakat, maka *customer* akan membuat dan mengirimkan *Purchase Order* (PO) kepada perusahaan
- Perusahaan akan membuat *Invoice* atau tagihan berdasarkan PO yang diterima, lalu dikirim kepada *customer*
- Setelah menerima *Invoice*, *customer* melakukan pembayaran, dan setelah itu mengirimkan bukti pembayaran kepada perusahaan
- Setelah menerima bukti pembayaran, perusahaan akan melakukan validasi terhadap pembayaran yang telah dilakukan apakah sudah sesuai atau belum
- Jika sudah sesuai, maka perusahaan akan memproses dan mengirimkan barang pesanan *customer*

*Flowchart* Proses Order *Customer* dilampirkan pada Lampiran 1.

### 3.3.4 User Manual Dashboard Admin Sistem

Pada sistem terdapat halaman *back-end* yang digunakan untuk *administrator* sistem. Halaman *backend* ini berisi *dashboard* data penjualan, order, dan *product* yang tersedia, serta dapat digunakan untuk mengelola data *customer*, data order, dan data *product* yang ada. Segala sesuatu yang dilakukan pada halaman *backend* ini bersifat *real-time* sehingga apapun yang dilakukan akan mempengaruhi sistem secara langsung. Karena bersifat *real-time*, maka diperlukan pemahaman mengenai cara penggunaan halaman *backend* ini untuk mencegah terjadinya atau dilakukannya kesalahan yang dapat berakibat fatal bagi perusahaan, misalnya order yang belum dibayar tanpa sengaja diubah menjadi *complete*. Oleh karena itu, penulis diminta untuk mempelajari penggunaan halaman *backend* admin tersebut tanpa mengubah data yang ada, dan menyusun user manual penggunaannya untuk dapat menjadi pedoman bagi *user* yang akan menggunakannya.

*User Manual* yang dibuat Rancangan tidak dapat dilampirkan, dengan tujuan kerahasiaan informasi perusahaan.



### 3.3.5 Penggunaan CMS Berbasis Framework Laravel

Oleh karena adanya kemungkinan untuk mengganti sistem di masa mendatang, maka akan terjadi kemungkinan juga digantinya CMS (*Content Management System*) yang digunakan untuk mengelola website *e-commerce* perusahaan. Pada sistem yang sedang berjalan, CMS yang digunakan adalah OpenCart, namun penulis diminta untuk mempelajari dan menganalisis jenis CMS yang menggunakan *framework* Laravel, terutama OctoberCMS dan PyroCMS.

Berikut ini merupakan jenis-jenis CMS yang menggunakan / berbasis framework Laravel:

- CMS yang sudah bersifat Final Release dan sudah stabil :
  1. Laravel Bootstrap
  2. TypiCMS Bootstrap starter
  3. Pongo CMS v1
  4. Wardrobe CMS (blogging)
  5. CodeSleeve
  
- CMS yang masih dalam tahap pengembangan, atau masih bersifat Beta Release :
  - PongoCMS v2
  - October CMS
  - Larapress
  - Coanda
  - PyroCMS

### 3.3.5.1 OctoberCMS untuk E-Commerce

Octobercms ini adalah CMS yang masih tergolong baru, maka belum ada *Themes* ataupun *Template* khusus *e-commerce* yang dapat di download, seperti WooCommerce pada Wordpress sebagai contoh, maka yang dapat digunakan sebagai dasar membangun *website e-commerce* menggunakan Octobercms adalah menggunakan beberapa plugin yang sudah ada / tersedia dan dapat di-download melalui *website* [octobercms.com](http://octobercms.com). Adapun beberapa plugin ini bersifat *Free*, namun juga banyak yang bersifat berbayar.

Hal pertama yang harus kita lakukan untuk memudahkan instalasi plugin-plugin yang diperlukan adalah, kita harus meng-*attach file* octobercms yang kita install dalam komputer kita ke dalam *file project* pada [octobercms.com](http://octobercms.com)

Langkah-langkahnya secara singkat adalah sebagai berikut :

- Buat *account* baru melalui *website* [octobercms.com](http://octobercms.com)
- Lalu, buat New Project setelah kita melakukan *login*
- Setelah *Project* yang baru dibuat, maka akan muncul kode unik dari *Project* tersebut
- Copy kode unik tersebut
- Lalu, masuk ke halaman *backend* admin dengan alamat [localhost/octobercms/backend](http://localhost/octobercms/backend)
- Lalu masuk ke menu System, dan pilih sub-menu Updates, dan klik tulisan “Attach to project”
- *Paste* / masukkan kode unik dari Project yang telah kita copy tadi, maka instalasi dan proses attach akan dilakukan secara otomatis

Jika file CMS kita sudah ter-*attach* pada project, maka kita hanya tinggal memilih *plugin* yang kita inginkan, dan klik “Add to project”, dan *plugin* tersebut secara otomatis akan masuk ke dalam file CMS di localhost.

Dari beberapa *plugin* yang ada, di bawah ini adalah contoh *plugin – plugin* yang dapat digunakan untuk membangun *website* e-commerce :

1. **FrontEnd User Management**, *plugin* ini digunakan agar kita dapat membuat pembagian *privilege* / hak akses terhadap *front-end user* yang akan menggunakan *website*
2. **Gallery**, *plugin* ini digunakan untuk membuat *gallery* gambar, ataupun membuat *slideshow* kumpulan–kumpulan gambar yang dapat digunakan untuk menunjukkan produk perusahaan
3. **Google Analytic**, seperti pada umumnya *plugin* ini digunakan untuk memberikan data *statistic* dalam penggunaan maupun pengaksesan *website* yang dibangun
4. **MenuList**, *plugin* ini digunakan untuk membuat struktur menu yang dapat digunakan dalam bagian *front-end website* yang dibangun.
5. **OctoMail**, adalah *plugin* yang dapat digunakan untuk membuat form seperti *Contact Us* pada halaman *front-end website*, sehingga admin *backend* akan mendapatkan email notifikasi mengenai *Contact* yang dikirimkan oleh *customer* melalui bagian *front-end*
6. **OctoSlider**, *plugin* ini digunakan untuk membuat *slider* gambar / foto yang dapat ditampilkan di halaman *front-end website*
7. **ProBlog**, adalah *plugin* yang cukup *powerful* dimana dengan menggunakan *plugin* ini kita dapat membuat postingan blog di dalam *website* yang kita bangun, dan *plugin* ini juga dilengkapi dengan *tools* untuk melakukan *Search Engine Optimization* (SEO)
8. **ToDo List**, adalah *plugin* yang digunakan untuk membuat *Memo* atau *List* mengenai hal-hal apa yang harus dilakukan oleh bagian tertentu dalam perusahaan pada waktu tertentu

### 3.3.5.2 PyroCMS untuk E-Commerce

Seperti OctoberCMS, PyroCMS juga merupakan salah satu CMS yang menggunakan *framework* Laravel, dan juga masih tergolong baru. Walaupun belum mempunyai *theme / template* yang dapat digunakan untuk *e-commerce*, namun PyroCMS memiliki *add-on platform* yang dapat digunakan untuk *e-commerce*, *platform* ini dinamakan FireSale. Secara singkat cara kerja *platform* ini bukanlah mengubah tampilan *website* kita menjadi *e-commerce*, namun menambahkan fitur-fitur *e-commerce* pada halaman backend *website* (halaman admin).

FireSale juga tersedia dalam beberapa jenis, yang membedakan hanyalah kelengkapan fitur yang disediakan. Untuk FireSale yang free atau gratis hanya terbatas fitur tertentu seperti menambahkan produk, brand, dsb. Namun untuk FireSale yang berbayar akan tersedia fitur lengkap hingga fitur untuk membantu SEO (Search Engine Optimization). Dalam hal ini, penulis hanya diminta untuk menganalisis dan mempelajari penggunaan FireSale yang free. Setelah menganalisis dan mempelajarinya, maka penulis kemudian membuat User Manual cara penggunaannya, yang disertakan pada lampiran laporan magang ini.

*User Manual* penggunaan *FireSale* pada PyroCMS dilampirkan pada Lampiran 2.

### 3.3.6 Analisis Kebutuhan dan Membuat ERD dari tabel vendor\_verification

PT Raksasa Laju Lintang merupakan perusahaan yang tergolong baru, oleh sebab itu belum seluruh bagian dalam perusahaannya lengkap. Salah satu bagian yang belum ada adalah bagian *Legal*. Salah satu tugas bagian *Legal* adalah menerima, memvalidasi, dan mengarsip dokumen-dokumen dan surat-surat keterangan yang digunakan untuk verifikasi, salah satunya adalah untuk verifikasi vendor.

Setelah mengadakan *meeting* dengan tim *Developer*, maka diputuskan untuk menambah tabel dan fitur pada sistem untuk menyimpan data untuk melakukan verifikasi terhadap suatu vendor. Sebagai contoh untuk mengecek apakah vendor A sudah melengkapi dokumen dan surat-surat verifikasi yang sah, atau ada dokumen apa saja yang masih belum lengkap? Hal ini dapat dibantu dengan adanya bagian dalam *database* yang menyimpan data-data yang diperlukan tersebut. Maka penulis diminta untuk bertanya dan mengumpulkan informasi kepada bagian / divisi *Buyer* (*Purchasing*), untuk mengetahui data-data apa saja yang harus dimasukkan ke dalam database. Hal ini dilakukan karena belum adanya divisi *Legal* di dalam perusahaan, sehingga divisi yang dapat mengetahui informasi terkait adalah divisi *Buyer*. Adapun proses singkat pendaftaran vendor perusahaan adalah sebagai berikut :

- *Buyer* akan menghubungi berbagai vendor untuk menawarkan kerjasama dalam transaksi jual beli, baik dalam jangka panjang maupun jangka pendek
- Jika vendor yang terkait berminat, maka buyer akan bertemu dengan vendor secara langsung untuk membahas kesepakatan dan perjanjian yang berlaku
- Lalu Buyer akan menyusun dan membuat dokumen MOU yang akan ditandatangani oleh kedua pihak, sebagai bukti sah dimulainya kerjasama yang berlaku.
- Setelah itu, vendor akan memberikan dokumen dan surat-surat verifikasi seperti SIUP, NPWP, dsb yang sah kepada bagian Buyer

- Buyer akan menyerahkan dokumen-dokumen tersebut kepada bagian Legal untuk dicek dan divalidasi
- Jika sudah valid, maka dokumen-dokumen tersebut akan diarsip dan disimpan selama kerjasama masih berlangsung.

Dari proses singkat diatas, salah satu masalah yang terjadi pada PT Raksasa Laju Lintang, selain belum adanya divisi Legal, dan pencatatan data vendor dan dokumen verifikasinya yang masih dilakukan secara manual menggunakan Microsoft Excel, sehingga tim *Developer* memutuskan perlu untuk diinput ke dalam sistem. Hal ini untuk mencegah hilangnya data dan dokumen penting, serta memudahkan untuk pengambilan (retrieve) informasi yang diperlukan. Setelah mengumpulkan informasi dan menganalisis, maka penulis merancang ERD untuk tabel *vendor\_verification* yang terdapat pada lampiran laporan magang ini. ‘

Adapun data-data yang diperlukan untuk vendor verification ini tidak bisa dijelaskan dan dilampirkan, dengan tujuan kerahasiaan informasi perusahaan.



### 3.3.7 SCRUM

Setelah menyelesaikan pekerjaan yang diminta, maka penulis diminta untuk melakukan pembelajaran mengenai SCRUM untuk menambah informasi dan pengetahuan. Setelah mempelajari tentang SCRUM, penulis melakukan sharing atau presentasi singkat mengenai topik tersebut.

SCRUM adalah sebuah kerangka kerja dimana orang - orang dapat menyelesaikan permasalahan kompleks yang senantiasa berubah, di mana pada saat bersamaan menghasilkan produk dengan nilai setinggi mungkin secara kreatif dan produktif. SCRUM bukan menyediakan deskripsi rinci tentang bagaimana tahapan pengembangan yang harus dilakukan seperti metodologi lainnya, namun SCRUM dilakukan untuk membantu Tim agar dapat mengetahui bagaimana cara terbaik untuk melakukannya.

SCRUM menggunakan pendekatan *iterative* dan *incremental* untuk meningkatkan prediktabilitas dan mengendalikan resiko , serta mempunyai 3 pilar yaitu :

- Transparansi, aspek penting dari proses yang berjalan dapat ditinjau oleh pihak-pihak yang berhubungan
- Inspeksi, Peninjauan berkala yang bertujuan mendeteksi perubahan atau masalah yang mungkin terjadi
- Adaptasi, proses yang sedang terjadi dapat menyesuaikan dengan perubahan yang terjadi

SCRUM memiliki 3 komponen utama, yaitu Artefak, Tim SCRUM, dan acara SCRUM. Sebelum membahas lebih lanjut mengenai ketiga komponen tersebut, kita harus terlebih dahulu mengenal *Sprint*. *Sprint* adalah perulangan yang dilakukan di dalam kerangka kerja hingga mencapai waktu yang telah ditentukan ( *deadline* ), dan hingga produk yang dikembangkan telah memenuhi kebutuhan yang diinginkan atau

hingga proyek tidak didanai lagi. Atau dengan kata lain *Sprint* dapat disebut sebagai sebuah *cycle* dalam SCRUM.

### 3.3.7.1 Artefak SCRUM

Dalam SCRUM, dikenal ada 3 artefak, yaitu :

- *Product Backlog*, adalah daftar terurut, dari setiap hal yang kemungkinan dibutuhkan di dalam produk, dan juga merupakan sumber utama, dari daftar kebutuhan mengenai semua hal yang perlu dilakukan terhadap produk. *Product Backlog* bersifat dinamis; dapat berubah agar produk dapat menjadi semakin layak, kompetitif di pasar, dan bermanfaat bagi penggunanya.
- *Sprint Backlog*, adalah sekumpulan *item Product Backlog* yang telah dipilih untuk dikerjakan di dalam satu *Sprint*, dan di dalamnya berisi daftar cara pengerjaannya secara lengkap, dan dapat berbentuk to-do list
- *Product Increment*, adalah potongan produk yang dikembangkan oleh Tim Pengembang dalam satu *Sprint* yang dapat digunakan oleh pengguna, atau dapat juga dikatakan sebagai gabungan dari semua *item Product Backlog* yang diselesaikan pada *Sprint*.

U N I V E R S I T A S  
M U L T I M E D I A  
N U S A N T A R A

### 3.3.7.2 Tim SCRUM

Tim SCRUM terdiri dari 3 bagian, yaitu :

- *Product Owner*, yaitu orang yang bertanggung-jawab terhadap suksesnya pengembangan produk, pendefinisian *Product Backlog*, dan bertanggung-jawab untuk memaksimalkan nilai dari produk yang dikembangkan.
- Tim Pengembang, yaitu semua orang-orang yang bertugas untuk mengembangkan product. Tim Pengembang bisa memiliki semua keahlian yang dibutuhkan untuk mengembangkan produk utuh, yang tidak terbatas dari *business analyst, software engineer, tester, technical writer*, dsb.
- SCRUM Master, adalah orang yang bertanggung jawab untuk memastikan SCRUM telah dipahami dan dilaksanakan, serta membantu setiap pihak di luar Tim SCRUM untuk dapat melakukan interaksi yang bermanfaat dengan Tim SCRUM, sehingga bisa memaksimalkan nilai yang dihasilkan oleh Tim SCRUM.

### 3.3.7.3 Acara SCRUM (SCRUM Events)

Dalam SCRUM, dikenal ada 4 acara atau events yang harus dilakukan :

- *Sprint Planning*, adalah pertemuan (meeting) untuk merencanakan pekerjaan yang akan dilakukan oleh Tim Pengembang selama satu *Sprint*. Hasil dari *Sprint Planning* adalah *Sprint Backlog* dan *Sprint Goal*.
- *Daily SCRUM Meeting*, adalah meeting yang dilakukan untuk melakukan evaluasi terhadap pekerjaan yang sudah dilakukan di hari sebelumnya, dan merencanakan pekerjaan yang akan dilakukan hari ini. *Daily SCRUM Meeting* dilakukan setiap hari hingga *Sprint* berakhir.

- *Sprint Review*, adalah *meeting* yang dilakukan pada akhir *Sprint*, untuk menginspeksi *product increment* yang telah dikembangkan selama satu *Sprint* tersebut, dan akan membahas perencanaan dan strategi untuk *Sprint* berikutnya.
- *Sprint Retrospective*, adalah *meeting* Tim SCRUM untuk meninjau dirinya sendiri dan membuat perencanaan mengenai peningkatan yang akan dilakukan di *Sprint* berikutnya

### 3.3.8 Flowchart Job Description Setiap Divisi Tim Creative

Seperti yang sudah penulis katakan sebelumnya, PT Raksasa Laju Lintang merupakan perusahaan yang masih tergolong baru, dan baru memiliki 1 kantor. Maka PT Raksasa Laju Lintang mulai mengembangkan dan memperluas usahanya dengan membuka kantor cabang baru yang berlokasi di Jelambar. Seiring dengan dibukanya kantor yang baru, tentu saja akan dicari dan direkrut karyawan-karyawan baru, khususnya untuk ditempatkan di kantor tersebut. Oleh karena itu, pihak manajemen perusahaan meminta kepada Tim *Creative* untuk merangkumkan *Job Description* masing-masing dari setiap Divisi yang ada, agar dapat digunakan sebagai panduan kerja bagi karyawan-karyawan baru di kantor yang baru.

Rangkuman *Job Description* dibuat sendiri oleh masing-masing karyawan pada setiap divisi, yang kemudian dikumpulkan dan diserahkan kepada Head dari Tim *Creative*, lalu disusun dalam dokumen yang rapi. Lalu penulis-pun diminta untuk menganalisis masing-masing *Job Description* yang ada, dan membuat *Flowchart* urutan pekerjaan dari *Job Description* tiap divisi tersebut.

Proses yang dilakukan oleh penulis dalam membuat *Flowchart* tersebut adalah:

- Membaca dan menganalisis *list Job Description* yang ada
- Melakukan wawancara singkat dengan tiap divisi yang ada untuk mendapatkan informasi dan keterangan lebih lanjut mengenai *Job Description*
- Membuat *Flowchart* berdasarkan informasi yang didapatkan
- Menunjukkan atau mempresentasikan secara singkat hasil pembuatan *Flowchart* kepada divisi yang bersangkutan
- Melakukan revisi atau perbaikan jika diperlukan
- Merangkum setiap *Flowchart* ke dalam satu folder dan diserahkan kepada *Head Tim Creative*

Adapun *flowchart* yang dibuat oleh penulis akan disertakan di dalam Lampiran 3 dari laporan magang ini.



### 3.3.9 Kendala Dalam Pelaksanaan Kerja Magang

Dalam melaksanakan program kerja magang, penulis menghadapi beberapa kendala berikut ini :

- Jam kerja yang bersifat *strict* / tegas, pada kontrak kerja yang disepakati dengan perusahaan, penulis tidak boleh terlambat, atau dengan kata lain tidak boleh sampai di kantor melewati pukul 08.30, dan apabila terlambat hingga 2 kali selama masa kerja magang, maka kontrak kerja tersebut akan dicabut dan dibatalkan.
- Penulis menemukan sedikit kesulitan dalam mengumpulkan informasi untuk analisis kebutuhan database *vendor\_verification*, karena tidak adanya divisi *Legal*, sehingga penulis hanya dapat mengumpulkan informasi dari divisi *Buyer*, dimana divisi *Buyer*-pun tidak terlalu paham dan yakin mengenai dokumen-dokumen yang dibutuhkan.

### 3.3.10 Solusi Untuk Mengatasi Kendala

Untuk mengatasi kendala-kendala seperti yang disebutkan di atas, maka berikut ini adalah solusi dan hal-hal yang dilakukan oleh penulis :

- Disiplin dalam waktu, penulis mengatur jam tidur yang baik, agar tidak terlambat bangun pagi, sehingga tidak terlambat untuk sampai di kantor pula.
- Menggunakan Reminder, untuk mengingatkan penulis akan jadwal hari ini, besok, dan seterusnya, sehingga penulis dapat mengatur waktu dalam melakukan aktivitas yang lain.
- Mencermati dan menganalisis dokumen yang ditunjukkan oleh divisi Buyer untuk mendapatkan informasi yang diperlukan, serta memahasnya dengan tim Developer sehingga tidak terjadi kesalahan dalam analisis yang dilakukan