



UNIVERSITAS INDONESIA

**Pengembangan VT-Vision sebagai Aplikasi Produktivitas Internal
PT. Midtrans**

LAPORAN KERJA PRAKTIK

Diajukan sebagai syarat kelulusan mata kuliah Kerja Praktik

KEVIN ADITYA PRABOWO

1306381616

FAKULTAS ILMU KOMPUTER

PROGRAM STUDI ILMU KOMPUTER

DEPOK

NOVEMBER 2016

HALAMAN PERNYATAAN ORISINALITAS

**Laporan Kerja Praktik ini adalah hasil karya saya sendiri,
dan semua sumber baik yang dikutip maupun dirujuk
telah saya nyatakan dengan benar.**

Nama: Kevin Aditya Prabowo

NPM: 1306381616

Tanda Tangan :

Tanggal : 22 November 2016

HALAMAN PERSETUJUAN DOSEN MATA KULIAH KERJA PRAKTIK

Laporan ini diajukan oleh :
Nama : Kevin Aditya Prabowo
NPM : 1306381616
Program Studi : Ilmu Komputer
Judul Kerja Praktik : Pengembangan *VT-Vision* sebagai Aplikasi
Produktivitas Internal PT. Midtrans

Telah berhasil diselesaikan laporan kerja praktik untuk fakultas dan dipresentasikan hasil kerja praktiknya dalam forum seminar kerja praktik sebagai persyaratan yang harus dipenuhi dalam mata kuliah Kerja Praktik.

Dosen Mata Kuliah Kerja Praktik,

Pelaksana Kerja Praktik,

(Meganingrum Arista, S.Kom., M.Kom.)

(Kevin Aditya Prabowo)

Ditetapkan di : Depok

Tanggal : 22 November

**HALAMAN PERNYATAAN PERSETUJUAN PUBLIKASI TUGAS AKHIR
UNTUK KEPENTINGAN AKADEMIS**

Sebagai sivitas akademik Universitas Indonesia, saya yang bertanda tangan di bawah ini:

Nama : Kevin Aditya Prabowo
NPM : 1306381616
Program Studi : Ilmu Komputer
Fakultas : Fakultas Ilmu Komputer
Jenis karya : Laporan Kerja Praktik

demi pengembangan ilmu pengetahuan, menyetujui untuk memberikan kepada Universitas Indonesia **Hak Bebas Royalti Noneksklusif** (*Non-exclusive Royalty-Free Right*) atas karya ilmiah saya yang berjudul :

Pengembangan VT-Vision sebagai Aplikasi Produktivitas Internal PT. Midtrans

beserta perangkat yang ada (jika diperlukan). Dengan Hak Bebas Royalti Noneksklusif ini Universitas Indonesia berhak menyimpan, mengalihmedia/formatkan, mengelola dalam bentuk pangkalan data (database), merawat, dan memublikasikan tugas akhir saya selama tetap mencantumkan nama saya sebagai penulis/pencipta dan sebagai pemilik Hak Cipta.

Demikian pernyataan ini saya buat dengan sebenarnya.

Dibuat di : Depok
Pada tanggal : 22 November 2016
Yang menyatakan

(Kevin Aditya Prabowo)

ABSTRAK

Nama : Kevin Aditya Prabowo
Program Studi : Ilmu Komputer
Judul : Pengembangan *VT-Vision* sebagai Aplikasi Produktivitas Internal
PT. Midtrans

Laporan kerja praktik ini adalah hasil dari kerja praktik penulis di PT. Midtrans Indonesia (Veritrans). Kerja praktik dilaksanakan selama 10 minggu, mulai tanggal 13 Juni 2016 sampai 26 Agustus 2016. Dalam kerja praktik, penulis memiliki peran sebagai *software engineering intern* dibawah pengawasan penyelia Rafeequl Rahman, *Vice President of Engineering* PT. Midtrans. Proyek kerja praktik ini dilakukan penulis bersama Dwiyan sebagai satu tim *scrum*, dengan pembagian tugas berdasarkan masing-masing *backlog* yang diambil. Proyek yang diberikan adalah pengembangan aplikasi *monitoring change request* dan *incident report*. Pengembangan yang dilakukan adalah dengan melakukan *refactoring* terhadap fitur-fitur yang sudah ada agar lebih efisien, dan penambahan fitur-fitur baru untuk memperlancar *business process* yang dilakukan.

Kata kunci : *change requests*, *Cucumber*, *incident reports*, *refactoring*, *Rspec*, *VT-Vision*

DAFTAR ISI

HALAMAN PERNYATAAN ORISINALITAS.....	1
HALAMAN PERSETUJUAN DOSEN MATA KULIAH KERJA PRAKTIK.....	2
HALAMAN PERNYATAAN PERSETUJUAN PUBLIKASI TUGAS AKHIR UNTUK KEPENTINGAN AKADEMIS.....	3
ABSTRAK.....	4
DAFTAR ISI.....	5
DAFTAR GAMBAR.....	8
DAFTAR LAMPIRAN.....	9
BAB 1 PENDAHULUAN.....	1
1.1. Proses Pencarian Kerja Praktik.....	1
1.2. Tempat Kerja Praktik.....	2
1.2.1. Profil Tempat Kerja Praktik.....	2
1.2.2. Posisi Penempatan Pelaksana Kerja Praktik dalam Struktur Organisasi.....	4
BAB 2 ISI.....	5
2.1. Pekerjaan dalam Kerja Praktik.....	5
2.1.1. Latar Belakang Pekerjaan.....	5
2.1.2. Metodologi.....	6
2.1.3. Tinjauan Pustaka.....	7
2.1.3.1. <i>Scrum</i>	7
2.1.3.2. <i>Unit Testing</i>	8
2.1.3.3. <i>Acceptance Testing</i>	8
2.1.3.4. <i>Refactoring</i>	9
2.1.4. Teknologi yang Digunakan.....	9

2.1.4.1.Git.....	9
2.1.4.2.Trello.....	10
2.1.4.3.Atom.....	10
2.1.4.4.Ruby on Rails.....	10
2.1.4.5.Heroku.....	11
2.1.4.6.Rspec.....	12
2.1.4.7.Cucumber.....	12
2.1.4.8.GitLab.....	13
2.1.4.9.GitKraken.....	13
2.1.4.10.Slack.....	14
2.1.5.Non-Teknis.....	14
2.1.6.Hasil Pekerjaan.....	15
2.1.6.1.Melakukan refactoring model approver.....	15
2.1.6.2.Melakukan refactoring sistem mailer Vision.....	15
2.1.6.3.Mengimplementasikan fitur “Clear all notifications”.....	16
2.1.6.4.Mengimplementasikan fitur “Duplicate change requests”.....	16
2.1.6.5.Perbaikan UI untuk change requests dan incident reports index.....	17
2.1.6.6.Pengimplementasian halaman “Notification”.....	17
2.1.6.7.Pengimplementasian fitur “Auto-complete” pada bagian “Comment” suatu change request.....	18
2.1.6.8.Pengimplementasian fitur “Quick Filter”.....	18
2.1.6.9.Pengimplementasian fitur “Change Request Lifeline”.....	19
2.1.6.10.Penyesuaian Unit Testing untuk setiap fitur yang diimplementasikan.....	19
2.1.6.11.Pembuatan Acceptance Testing untuk setiap fitur yang diimplementasikan.....	20
2.2.Analisis.....	20
2.2.1.Pelaksanaan Kerja Praktik.....	20
2.2.1.1.Kesesuaian dengan KAKP.....	20
2.2.1.2.Kendala yang dihadapi.....	21

2.2.1.3. Penilaian terhadap Tempat Kerja Praktik.....	21
2.2.2. Relevansi dengan Perkuliahan.....	22
2.2.2.1. Dasar-dasar Pemrograman (DDP) dan Struktur Data Algoritma (SDA).....	22
2.2.2.2. Perancangan dan Pembuatan Web (PPW).....	22
2.2.2.3. Rekayasa Perangkat Lunak (RPL) dan Proyek Perangkat Lunak (PPL).....	22
2.2.2.4. Penjaminan Mutu Perangkat Lunak (PMPL).....	23
2.2.2.5. Basis Data dan Basis Data Lanjut.....	23
BAB 3 PENUTUP.....	24
3.1. Kesimpulan.....	24
3.2. Saran.....	24
DAFTAR REFERENSI.....	26
LAMPIRAN 1 KERANGKA ACUAN KERJA PRAKTIK.....	28
LAMPIRAN 2 LOG KERJA PRAKTIK.....	29

DAFTAR GAMBAR

Gambar 1. Struktur Organisasi PT Midtrans.....	3
--	---

DAFTAR LAMPIRAN

LAMPIRAN 1 KERANGKA ACUAN KERJA PRAKTIK.....	28
LAMPIRAN 2 LOG KERJA PRAKTIK.....	29

BAB 1

PENDAHULUAN

1.1. Proses Pencarian Kerja Praktik

Pencarian kerja praktik pertama kali penulis lakukan berdasarkan saran dari kakak penulis yang sedang bekerja di Singapura. Perusahaan yang disarankan adalah J.P. Morgan, sebuah perusahaan yang bergerak di bidang korporat. Perusahaan tersebut membuka lowongan *internship* untuk program IT, dimana para *intern* nantinya akan fokus ke dalam optimisasi *business process* yang dimiliki perusahaan tersebut. Pengajuan *internship* untuk J.P. Morgan dilakukan pada bulan Agustus 2015, untuk program *internship* tahun 2016. Penulis mengajukan lamaran melewati *website* perusahaan tersebut, tetapi sayangnya pengajuan penulis ditolak.

Proses pencarian kerja praktik yang berikutnya penulis lakukan pada awal bulan Mei 2016. Penulis melakukan pencarian informasi kerja praktik dari teman-teman satu angkatan yang sudah mendapatkan kerja praktik maupun yang masih dalam proses pencarian. Penulis mendapatkan kabar dari saudari Nayana, bahwa ia mengundurkan diri dari PT Suitmedia, sehingga terdapat lowongan yang kosong. Penulis pun mencari tahu lebih lanjut dan mengirimkan CV ke saudari Nayana agar dapat diteruskan ke PT Suitmedia sebagai rekomendasi. Tepat dua hari setelah mengajukan CV, penulis diberikan kabar dari pihak HRD PT Suitmedia untuk melakukan proses wawancara di kantor PT Suitmedia di daerah Pejaten pada tanggal 19 Mei 2016. Penulis pun datang dan melakukan proses wawancara dan tes programming di tempat, dan setelah selesai, pihak HRD mengabarkan akan diberitahukan hasil seleksinya pada minggu yang mendatang.

Tetapi, masa tersebut adalah masa dimana penulis sedang mengikuti mata kuliah PPL - Mentorship dengan PT Midtrans. Pada salah satu sesi dimana penulis datang ke kantor PT Midtrans, HRD dari PT Midtrans menginformasikan bahwa

terdapat lowongan internship karena adanya pelamar yang mengundurkan diri. Mendengar hal tersebut, penulis merasa yakin untuk mengajukan diri untuk mengikuti program internship pada PT Midtrans. Setelah dilakukan wawancara, ternyata penulis langsung diterima dan telah ditentukan perannya dalam program internship pada bulan Juni 2016 nanti.

Esok harinya, penulis dihubungi oleh pihak HRD dari PT Suitmedia, dimana ternyata pengajuan internship penulis diterima oleh PT Suitmedia. Pada saat tersebut, penulis dengan berat hati memberitahukan kepada pihak HRD dari PT Suitmedia bahwa penulis telah diterima terlebih dahulu pada PT Midtrans, sehingga penulis harus dengan berat hati mengundurkan diri dari program internship pada PT Suitmedia, dan PT Suitmedia pun menerima pengunduran diri penulis.

1.2. Tempat Kerja Praktik

Pada subbab ini, penulis akan menjelaskan mengenai profil dari PT Midtrans serta penempatan penulis dalam struktur organisasi PT Midtrans selama kerja praktik

1.2.1. Profil Tempat Kerja Praktik

PT Midtrans merupakan perusahaan yang telah berdiri sejak tahun 2012. Perusahaan ini bergerak di bidang *online payment gateway* dengan moto *simple, reliable and secure*, dan telah melakukan kerja sama dengan beberapa bank terkemuka di Indonesia seperti BCA, BNI, Bank Mandiri, dan bank lainnya [1]. *Payment gateway* merupakan sebuah layanan yang bertujuan untuk mempermudah transaksi *online* dengan menjadi perantara antara penjual dengan bank, sehingga penjual tidak perlu repot mengurus administrasi dengan bank. Pembeli cukup melakukan pembelian barang penjual, dan sistem yang akan meneruskan transaksi tersebut ke bank, dan mendeteksi apakah transaksi tersebut berhasil atau gagal.

Selain *payment gateway*, PT Midtrans juga mengembangkan dua layanan lain yang dapat membantu *business process* dari penjual, yaitu *risk management* dan *chat*

commerce. *Risk management* bertujuan untuk menganalisis pola pembayaran pelanggan dan mendeteksi apakah ada kecurangan dalam setiap transaksi yang dilakukan [2]. Sedangkan, *chat commerce* adalah suatu aplikasi baru yang dikembangkan, dimana pembeli dapat melakukan *chat* dengan *customer service*, dimana data dari *chat* tersebut dapat dikonversikan menjadi data transaksi sehingga pembeli tidak perlu melakukan konfirmasi pembelian manual untuk setiap barang [3]. Aplikasi *chat commerce* ini diberi nama Prism.

PT Midtrans terdiri atas 8 divisi yang memiliki fungsinya masing-masing, dimana seluruhnya dikepalai oleh Ryu Kawano sebagai CEO. Struktur organisasi PT Midtrans dapat dilihat pada gambar berikut.



Gambar 1. Struktur Organisasi PT Midtrans

Sumber: *Organizational Chart 2016-updated 21 July.pdf*. Wu, Tannia.

1.2.2. Posisi Penempatan Pelaksana Kerja Praktik dalam Struktur Organisasi

Selama kerja praktik di PT Midtrans, penulis ditempatkan pada divisi *Engineering* yang diketuai oleh Rafeequl Rahman sebagai *software engineer*. Divisi ini berfokus pada pengembangan aplikasi yang bersifat internal maupun eksternal. Pada kerja praktik ini, *buddy* (mentor) dari penulis adalah Rafeequl Rahman sendiri sebagai penyelia dan bertanggung jawab penuh atas proyek yang penulis kerjakan.

BAB 2

ISI

2.1. Pekerjaan dalam Kerja Praktik

Pada subbab ini, penulis akan menjelaskan mengenai latar belakang pekerjaan yang diberikan dalam kerja praktik, metodologi, istilah, dan teknologi yang digunakan, kemampuan non-teknis yang didapatkan, serta hasil pekerjaan selama periode kerja praktik.

2.1.1. Latar Belakang Pekerjaan

Dalam pengembangan aplikasi dan layanannya, PT Midtrans menggunakan metodologi *agile*, dimana perubahan terhadap aplikasi yang sudah di-*deploy* maupun penambahan fitur-fitur baru akan terjadi secara progresif setiap hari demi meningkatkan kualitas pelayanan kepada pelanggan. Perubahan-perubahan ini disebut sebagai *change requests*, dimana pada tiap *change requests* akan terdapat tujuan dilakukannya perubahan dan detail-detail perubahan yang dilakukan. Selain itu, dikarenakan PT Midtrans menangani banyak transaksi setiap harinya, jika terjadi suatu hal yang dibutuhkan penanganan secara langsung, maka kasus-kasus tersebut harus dicatat agar diketahui penyebabnya dan bagaimana caranya untuk mencegah kasus tersebut terjadi lagi kedepannya. Hal ini disebut sebagai *incident reports*.

Atas dasar dua hal tersebut, PT Midtrans membutuhkan sebuah portal dimana *change requests* dan *incident reports* dapat dibuat dan dicatat untuk nantinya diulas oleh tim. Pada tahun 2015, kakak tingkat penulis yang melakukan internship di PT Midtrans telah membuat portal tersebut dengan nama Vision yang berbasis Ruby on Rails. Vision telah digunakan kurang lebih 1 tahun oleh PT Midtrans untuk menangani *change requests* dan *incident reports* agar *business process* PT Midtrans dapat berjalan dengan baik. Setelah 1 tahun pemakaian tersebut, beberapa *developer*

dan *scrum master* di PT Midtrans menganggap bahwa Vision masih dapat dikembangkan lebih lanjut untuk memenuhi kebutuhan sekarang.

Pada proyek kerja praktik ini, penulis bersama dengan Dwiyan ditugaskan untuk melakukan pengembangan terhadap aplikasi Vision tersebut. Pengembangan ini meliputi *refactoring model* dan *controller*, serta penambahan fitur-fitur seperti *searching*, *slack integration*, dan lainnya. Diharapkan dengan pengembangan ini, alur dari *change requests* dan *incident reports* dapat lebih optimal dan transparan. Selain menambahkan fitur, penulis dan Dwiyan juga ditugaskan untuk membuat *unit tests* dan *acceptance tests* untuk setiap fitur yang penulis dan tim implementasikan.

2.1.2. Metodologi

Pengembangan aplikasi Vision yang penulis dan Dwiyan kerjakan dilakukan dengan menerapkan metodologi *scrum*. Alasan penggunaan metodologi ini karena memang metodologi yang digunakan di PT Midtrans adalah *scrum* dan Vision dikembangkan tahun lalu juga dengan menggunakan *scrum*. Selain alasan tersebut, metodologi *scrum* digunakan karena dalam pengembangan aplikasi Vision, mungkin saja terdapat fitur-fitur yang bersifat urgent dan mendapatkan prioritas utama di *backlog* daripada fitur lain, dan mungkin saja muncul fitur baru yang dibutuhkan seiring berjalannya proses pengembangan ini.

Penerapan metodologi *scrum* dilakukan dengan beberapa kegiatan. Pada awalnya, penyelia membuat *backlog* pengembangan yang akan dilakukan di Trello, dimana penulis dan Dwiyan nantinya dapat mengambil fitur pengembangan yang akan dikerjakan. *Backlog* ini dapat diisi baik oleh penyelia maupun *developer* lain yang membutuhkan fitur baru di Vision. Setiap awal minggu, penulis bersama Dwiyan dan penyelia akan melakukan *sprint planning*, dimana kami menentukan fitur mana saja yang akan dikembangkan pada *sprint* ini. Setiap hari, kami melaporkan progres ke penyelia dan jika sudah ada fitur yang selesai, kami melakukan demo bersama-sama penyelia maupun dengan *developer* yang mengajukan fitur tersebut. Setelah demonstrasi dilakukan, penyelia akan melakukan *review* terhadap kode yang

telah diimplementasi, dan jika sudah disetujui, kami akan melakukan *deployment* ke server bersama-sama. Dan pada akhirnya, setiap 3 minggu sekali, penulis dan Dwiyan melakukan demonstrasi hasil pengembangan Vision kepada seluruh anggota divisi *Engineering* PT Midtrans, dengan tujuan mensosialisasikan perubahan yang terjadi dan meminta *feedback* terhadap fitur-fitur yang telah dikembangkan.

2.1.3. Tinjauan Pustaka

Berikut adalah beberapa istilah yang akan digunakan dalam laporan ini, beserta definisinya masing-masing.

2.1.3.1. *Scrum*

Scrum adalah sebuah *development framework* untuk *software* yang masuk ke dalam *agile development*. *Scrum* memiliki 4 nilai yang harus diaplikasikan dalam setiap penerapannya [4].

1. *Individuals and interactions over processes and tools*

Scrum adalah sebuah *team-based approach* untuk memberikan nilai-nilai ke dalam bisnis. Seluruh anggota tim bekerja sama untuk mencapai suatu tujuan, dengan mengedepankan interaksi yang efektif antar anggota tim

2. *Working software over comprehensive documentation*

Scrum membutuhkan sebuah *increment* dari sebuah produk sebagai hasil utama dari tiap *sprint*. *Increment* ini diharapkan sudah dalam kualitas layaknya sebuah *finished product*, meskipun mungkin belum memenuhi keseluruhan fungsionalitas yang diinginkan.

3. *Customer collaboration over contract negotiation*

Scrum adalah *framework* yang didesain untuk memfasilitasi kolaborasi. Anggota tim saling berkolaborasi untuk mencari cara terbaik untuk membuat *software* tersebut, dan product owner akan berkolaborasi dengan stakeholders

untuk memastikan fungsionalitas yang telah dibuat sudah sesuai dengan kebutuhan pasar.

4. *Responding to change over following a plan*

Tim yang menggunakan *scrum* membuat rencana baru untuk setiap *sprint*. Meskipun banyak tim lain yang melakukan perencanaan hingga akhir, tim *scrum* menghargai akan adanya perubahan *requirement*, sehingga tidak cuma mengikuti rencana dari awal sampai akhir secara buta.

2.1.3.2. *Unit Testing*

Unit testing adalah sebuah proses dalam pengembangan *software*, dimana sebuah bagian terkecil dari sebuah aplikasi, yang disebut sebagai unit, akan dites secara individu dan independen [5]. Artinya, unit tersebut tidak memiliki hubungan dengan unit lainnya, murni yang dites adalah fungsi-fungsi yang terdapat dalam unit tersebut.

Unit testing bertujuan untuk memastikan bahwa satuan unit terkecil pada sebuah program dapat melakukan fungsinya dengan baik. Meskipun *unit testing* membutuhkan waktu yang cukup lama dan *effort* yang cukup besar, tetapi jika dilakukan *unit testing*, *confidence* para *developer* terhadap aplikasi yang dibuat akan meningkat.

2.1.3.3. *Acceptance Testing*

Acceptance Testing adalah salah satu tingkat dalam *software testing* dimana sebuah sistem dites untuk akseptabilitasnya [6]. Tujuan dari tes ini adalah untuk mengevaluasi apakah sebuah sistem memenuhi *business requirements* dan menilai apakah sistem sudah siap untuk dilakukan *delivery*. Pada *acceptance test*, metode yang digunakan disesuaikan dengan *customer* dan *stakeholder*, sehingga setelah dilakukan *acceptance test*, *customer* dan *stakeholder* dapat menentukan apakah akan menerima sistem tersebut atau tidak.

2.1.3.4. *Refactoring*

Refactoring adalah suatu proses dimana *developer* melakukan restrukturisasi terhadap kode yang sudah ada, tetapi hasil restrukturisasi tersebut tidak mengubah *external behaviour* dari kode tersebut [7]. *Refactoring* dilakukan untuk mengoptimalkan kode dari sisi non-fungsional-nya, misalnya keterbacaan kode, penurunan kompleksitas kode yang mungkin mempercepat pemrosesan data, dan lain sebagainya. Tujuan utama *refactoring* biasanya tertuju pada 2 hal, yaitu:

1. *Maintainability*. Akan lebih mudah untuk memperbaiki *bugs* yang ada jika kode yang diimplementasi lebih mudah dibaca dan dimengerti alur prosesnya.
2. *Extensibility*. Akan lebih mudah untuk menambahkan fitur-fitur baru pada kode jika keseluruhan kode mengikuti suatu design pattern yang jelas.

2.1.4. Teknologi yang Digunakan

Dalam menjalankan kerja praktik ini, penulis menggunakan beberapa teknologi sebagai berikut.

2.1.4.1. Git

Git adalah sebuah *version control system* yang bersifat *free* dan *open source*, dan sudah umum digunakan dalam proses pengembangan software. Git pertama dikembangkan oleh Linus Torvalds pada tahun 2005 [8], dan Git ini sudah bersifat *fully distributed*, sehingga semua kontributor pada proyek dapat melihat keseluruhan *revision history* yang ada. Pada awalnya, *developer* akan melakukan *initial commit* untuk menginstantiasi *git project*, dan kemudian para *developer* lain melakukan *branching* untuk mengerjakan fiturnya masing-masing.

Pada kerja praktik ini, Git digunakan sebagai *version control* utama untuk pengembangan Vision, dimana beberapa fungsi utama yang digunakan adalah untuk melakukan *branching*, *merging*, dan *rebasing*.

2.1.4.2. Trello

Trello adalah sebuah *project management tool* yang dibuat oleh Fog Creek Software dan dirilis pada tahun 2011 [9]. Trello berfungsi untuk membantu developer dalam menyusun penjadwalan dan pengklasifikasian task yang akan dikerjakan. Product owner akan membuat task yang harus dikerjakan, dan kemudian seorang *developer* dapat mengambil suatu task tersebut untuk dikerjakan.

Pada kerja praktik ini, penulis bersama Dwiyan menggunakan Trello untuk melakukan pembagian tugas. Setelah mengambil tugas, kami mendefinisikan terlebih dahulu *subtask* masing-masing task dan barulah kami memulai mengerjakan *task* tersebut.

2.1.4.3. Atom

Atom adalah sebuah *text-editor* yang modern, dan dapat dikustomisasi tanpa mengubah *config file* sedikitpun [12]. Fitur-fitur yang ditawarkan oleh Atom akan membantu *developer* menaikkan produktivitasnya sehingga keseluruhan kualitas dari program dapat tercapai dengan lebih baik.

Penulis memilih menggunakan Atom karena setelah melakukan perbandingan dengan *text-editor* lainnya, Atom terasa lebih ringan dan untuk *browsing* antar *file* dalam *project* lebih mudah.

2.1.4.4. Ruby on Rails

Ruby on Rails adalah sebuah *framework* pengembangan web yang menggunakan bahasa *programming* Ruby sebagai basisnya. Sudah banyak aplikasi yang menggunakan Ruby on Rails sebagai *framework* pengembangannya, antara lain Github, Twitch, Airbnb, dan BaseCamp [10]. Ruby on Rails menggunakan *framework* MVC, yaitu *model*, *view*, dan *controller*. Penjelasan masing-masing adalah sebagai berikut [11]:

1. *Model*. Bertugas untuk mengatur data, *logic*, dan aturan-aturan pada sebuah objek.

2. *View*. Bertugas untuk merepresentasikan output dari informasi. Biasanya digunakan untuk menampilkan data dari *model*.
3. *Controller*. Bertugas untuk menerima input dan mengkonversikan input tersebut menjadi perintah untuk mengubah *view* atau *model*.

Vision menggunakan Ruby on Rails sebagai *framework* pengembangannya. Maka, penulis akan menggunakan Ruby on Rails sebagai *framework* pengembangan selama kerja praktik ini.

2.1.4.5. Heroku

Heroku adalah sebuah *cloud platform* yang menyediakan layanan berupa PaaS, atau *Platform as a Service*. Heroku mendukung beberapa *framework* pengembangan yang terkenal seperti Node.js, Ruby on Rails, Clojure, Python, dan beberapa *framework* lainnya. Untuk menggunakan layanan Heroku, developer cukup mendaftarkan aplikasi ke website Heroku dan melakukan ‘*git push heroku master*’. Setelah melakukan kedua hal tersebut, maka aplikasi kita akan langsung dapat diakses melalui sebuah URL yang telah ditentukan. *Developer* tidak perlu untuk melakukan pengaturan-pengaturan *virtual machine*, mereka cukup fokus kepada satu hal, yaitu pembuatan aplikasi itu sendiri.

Heroku telah banyak digunakan oleh perusahaan-perusahaan untuk *deploy* aplikasinya dikarenakan kemudahan dan simplisitas dari Heroku itu sendiri. Aplikasi-aplikasi yang di-*deploy* bisa berupa aplikasi eksternal maupun aplikasi produktivitas internal. Beberapa perusahaan tersebut antara lain Bonobos, Carbon Five, Macy’s, Roomer, dan PT Midtrans sendiri.

Pada kerja praktik ini, penulis menggunakan Heroku karena Vision memang di-*deploy* di Heroku, dan penulis telah memiliki pengalaman dalam menggunakan Heroku saat mengambil mata kuliah PPL.

2.1.4.6. Rspec

Rspec adalah tools yang dikembangkan oleh Steven Baker, Dave Asteks, Aslak Hellesøy, dan David Chelimsy pada awal 2005 yang digunakan untuk melakukan *automated testing* pada kode program yang berbasis bahasa Ruby. Rspec memiliki modul bernama Rspec-rails sehingga *developer* hanya perlu memasang modul tersebut dalam programnya. Rspec akan terintegrasi ke dalam kode program Ruby on Rails dan *developer* cukup membuat *testing* yang sesuai tanpa melakukan konfigurasi lebih lanjut.

Dalam kerja praktik ini, penulis menggunakan Rspec sebagai alat *testing* utama untuk melakukan *unit testing*. Setiap modul atau fitur yang penulis implementasikan hanya akan diterima oleh penyelia jika dan hanya jika *unit test* untuk modul atau fitur tersebut telah diimplementasikan dan berhasil.

2.1.4.7. Cucumber

Cucumber adalah tools untuk mempermudah *developer* melakukan *behaviour driven development* [14]. Fungsi utama Cucumber adalah untuk melakukan *automated acceptance testing*, dimana istilah yang digunakan *high-level language* sehingga dapat mudah dimengerti oleh pengguna dan *stakeholder*. *Acceptance test* yang dilakukan terdiri dari 3 komponen utama, yaitu:

1. *Given*. Bagian ini adalah bagian dimana ditentukan keadaan sebelum fitur yang dimaksud akan diuji. Contohnya, sebelum membeli baju di toko *online*, maka *user* harus melakukan *login* terlebih dahulu.
2. *When*. Bagian ini adalah bagian utama dari *acceptance test*. Pada bagian ini, *developer* menyimulasikan aksi-aksi yang dilakukan oleh pengguna untuk mencapai tujuan tertentu. Misal, untuk membeli baju di toko *online*, pengguna menekan tombol beli, kemudian pengguna memasukkan jumlah yang akan dibeli, dan seterusnya.

3. *Then*. Bagian ini adalah ekspektasi *developer* hasil apakah yang akan muncul jika kode di bagian *When* dijalankan.

Pada kerja praktik ini, selain menggunakan Rspec, penulis juga harus membuat *acceptance test* untuk fitur-fitur yang diimplementasikan.

2.1.4.8. GitLab

GitLab adalah sebuah aplikasi yang berfungsi sebagai Git *sharing repository*. Dengan GitLab, *developer* dapat melakukan *coding*, *testing*, dan *deploying* secara bersama-sama [15]. GitLab juga menyediakan manajemen Git yang dengan kontrol akses ke *repository* tersebut, *code reviews*, *issue tracking*, *activity feeds*, dan *wikis*. Yang membedakan GitLab dengan aplikasi Git *sharing repository* lainnya adalah GitLab memiliki fitur *continuous integration internal*. Sehingga, untuk setiap perubahan yang dilakukan oleh *developer* ke *master branch*, akan dilakukan *tests* yang telah dibuat sebelumnya secara otomatis, dan akan melaporkan apakah *tests* tersebut berhasil atau gagal.

Pada kerja praktik ini, penulis menggunakan GitLab sebagai Git *sharing repository* karena aplikasi *repository* Vision memang berada di GitLab. Pada GitLab tersebut, penulis dan Dwiyan memiliki tingkat akses sebagai *contributor* saja, sedangkan penyelia yang memiliki akses sebagai *owner* yang dapat melakukan *merge requests* ke *master branch*.

2.1.4.9. GitKraken

GitKraken adalah sebuah aplikasi GUI untuk melihat *repository* Git yang ada pada *remote server* dan *local*. Aplikasi ini memiliki fitur *auto-fetch* sehingga setiap perubahan yang terjadi (misal *developer* lain melakukan *push*) akan secara langsung terlihat pada *client* GitKraken-nya. Karena transparansi dan visualisasi data *repository* yang baik dan jelas, diharapkan produktivitas *developer* dapat meningkat.

Dalam kerja praktik ini, penulis menggunakan aplikasi GitKraken atas saran dari Dwiyan yang sudah pernah menggunakan GitKraken sebelumnya. Dengan

adanya GitKraken ini, penulis bersama Dwiyan merasa lebih terarah untuk keseluruhan *branch* pengembangannya.

2.1.4.10. Slack

Slack adalah sebuah aplikasi *messaging* yang ditujukan untuk tim *development* untuk mempermudah komunikasi intra dan ekstra tim dalam suatu instansi. Selain sebagai *messaging client*, pengguna Slack juga dapat melakukan pengunggahan file, pengunggahan *snippet* kode, dan melakukan *pin message*. Tetapi, fitur utama dari Slack adalah disediakannya API program, sehingga developer dapat menggunakan API tersebut untuk misalnya mengirim pesan otomatis dari program jika suatu hal terjadi.

Pada kerja praktik ini, penulis bersama Dwiyan menggunakan Slack sebagai media komunikasi utama dengan penyelia di dalam maupun di luar jam kerja.

2.1.5. Non-Teknis

Selama melaksanakan kerja praktik, penulis mendapatkan beberapa kemampuan non-teknis. Karena penyelia penulis memiliki jabatan yang tinggi di PT Midtrans, tentu penyelia akan sibuk untuk mengurus berbagai sektor pada divisi *engineering*. Maka dari itu, dibutuhkan kemampuan untuk melakukan manajemen waktu dengan baik, sehingga ketika penyelia menyediakan waktu untuk penulis, maka penulis dapat memaksimalkan penyampaian informasi dan progres dari proyek yang sedang dikerjakan. Selain itu, penulis juga dituntut untuk berpikir secara kreatif untuk menyelesaikan permasalahan yang diberikan oleh penyelia. Sehingga, solusi yang ditawarkan akan sesuai dengan spesifikasi, dan diharapkan akan seefektif dan seefisien mungkin. Kemampuan lain yang didapat adalah kemampuan untuk berkoordinasi dengan tim dan *product owner* ketika menemui masalah dalam pengimplementasian solusi, sehingga masalah akan cepat selesai dan sesuai dengan kesepakatan tim.

2.1.6. Hasil Pekerjaan

Dalam pengembangan aplikasi Vision ini, penulis melakukan pembagian tugas dengan saudara Dwiyani sesuai dengan backlog yang diambil oleh masing-masing orang. Berikut adalah hasil pengembangan Vision yang telah penulis lakukan.

2.1.6.1. Melakukan refactoring model approver

Pada setiap *change request* yang dibuat, agar *change request* tersebut dapat diproses, perlu adanya persetujuan terlebih dahulu dari beberapa orang yang ditunjuk sebagai *approver*. Tetapi, pada implementasi awalnya, sudah ditentukan 3 orang yang akan menjadi *approver* untuk semua *change request* yang dibuat. Namun, belum tentu orang tersebut memiliki relevansi dengan *change request* yang dibuat. Maka, muncul kebutuhan agar pada setiap pembuatan *change request*, pembuat dapat menentukan sendiri siapa yang akan menjadi *approver* untuk *change request* tersebut.

Maka, penulis melakukan perubahan terhadap model *approver*. Model tersebut akan dihubungkan ke model *user*, sehingga setiap *change request* akan memiliki beberapa *user* (yang memiliki *role* *approver*) yang akan diminta untuk melakukan *review* dan memberikan persetujuannya. Hal ini dilakukan dengan mengubah area input yang tadinya berupa teks menjadi *dropdown* yang terhubung dengan model *user*. Dengan perubahan ini, maka diharapkan *change request* akan lebih terkontrol untuk masing-masing divisinya dan tidak memberatkan beban di beberapa orang tertentu saja.

2.1.6.2. Melakukan refactoring sistem mailer Vision

Setiap sebuah *change request* dibuat, maka sistem Vision akan mengirimkan *e-mail* secara otomatis ke pembuat *change request* dan ke seluruh *approver* sebagai notifikasi. Tetapi, sistem *mailer* masih menggunakan konvensi yang lama yaitu mengirim *e-mail* ke semua *user* yang memiliki *role* berupa *approver*.

Maka, penulis melakukan perubahan pada sistem *mailer* tersebut. Karena pembuatan *change request* memiliki sistem penunjukkan *approver*, maka *e-mail* yang

dikirimkan hanyalah ke orang-orang yang telah ditunjuk sebagai *approver* tersebut, tidak lagi mengirim ke seluruh *approver*.

2.1.6.3. Mengimplementasikan fitur “Clear all notifications”

Pada *web-app* Vision, terdapat sistem notifikasi internal. Notifikasi ini berfungsi untuk memberitahukan apa saja yang terjadi pada sistem, misalnya terdapat *change request* baru, adanya perubahan terhadap suatu *change request*, dan lain sebagainya. Tetapi, untuk menghilangkan notifikasi tersebut, kita harus membuka link ke *change request* tersebut. Terdapat masalah ketika kita jarang menghapus notifikasi, maka notifikasi akan bertumpuk dan kita harus membuka setiap *change request* satu persatu.

Maka, penulis mengimplementasikan fitur untuk menghapus semua notifikasi. Hal ini dilakukan dengan membuat sebuah *method* yang akan membuat notifikasi yang terikat pada setiap *change request* berada pada kondisi sudah terbaca. Pengguna cukup menekan tombol tersebut, melakukan konfirmasi penghapusan, dan semua notifikasi akan terhapus.

2.1.6.4. Mengimplementasikan fitur “Duplicate change requests”

Karena sistem pengembangan yang bersifat *agile*, maka tentu saja akan terdapat perubahan-perubahan yang bersifat minor. Karena banyaknya *field* yang harus diisi dalam pembuatan sebuah *change request*, tentu akan menghabiskan banyak waktu jika untuk setiap *change request* harus dibuat dari awal. Selain itu, jika sebuah *change request* ditolak, maka *developer* akan lebih mudah untuk menggunakan *template change request* yang sebelumnya telah dibuat dan menambahkan beberapa perbaikan.

Maka, penulis mengimplementasikan fitur *duplicate change request*. Pengguna cukup menekan tombol *duplicate*, lalu pengguna akan diarahkan ke sebuah form pembuatan *change request* tetapi seluruh *input field*-nya sudah terisi dengan *change request* yang sebelumnya, kecuali *field* tanggal dan *approver*.

2.1.6.5. Perbaikan UI untuk *change requests* dan *incident reports index*

Dalam menelusuri aplikasi Vision, penulis menemukan beberapa hal secara tampilan yang dapat diperbaiki. Pertama, pada halaman *index* untuk *change request* dan *incident report*, ditampilkan daftar seluruh *change request* dengan menggunakan *pagination*. Tetapi, ketika menggunakan perangkat dengan resolusi layar yang kecil, daftar ini akan terlihat kecil dan butuh waktu yang cukup lama untuk mengakses opsi *pagination* yang berada di bawah halaman. Maka, penulis mengubah tampilan daftar *change request* tersebut sehingga daftar *change request* berada dalam sebuah area yang *scrollable*. Sehingga, halaman tidak bergerak sama sekali, hanya area *scrollable* saja dan opsi *pagination* akan selalu terlihat. Hal yang sama terjadi pada *sidebar filter* untuk melakukan *filtering* pada *change request* dan *incident report*. Penulis pun memberikan solusi yang sama dengan membuat *sidebar* tersebut menjadi *scrollable*. Selain itu, untuk menutup *sidebar* tersebut, pengguna harus menekan ikon silang yang berada pada pojok kanan atas. Agar lebih intuitif, penulis membuat jika pengguna melakukan klik pada bagian luar *sidebar*, maka *sidebar* akan tertutup. Hal ini lebih ke preferensi penulis dan penyelia, dan diterima cukup baik oleh para pengguna.

2.1.6.6. Pengimplementasian halaman “Notification”

Sistem notifikasi Vision mirip seperti Facebook, yang berada pada *navbar* dan ketika diklik, akan muncul jenis-jenis notifikasinya. Sistem penampilan data dengan bentuk *accordion* membuat keterbacaan setiap notifikasi rendah karena pada umumnya judul *change request* yang cukup panjang. Hal ini merupakan salah satu faktor yang membuat notifikasi jarang dibaca sehingga menumpuk.

Untuk mengatasi masalah tersebut, penulis mengimplementasikan sebuah halaman yang memang didedikasikan untuk menampilkan seluruh notifikasi yang ada. Jenis-jenis notifikasi akan dipisahkan ke beberapa *tab* dan jumlah masing-masing notifikasi akan ditunjukkan pada *tab* tersebut. Halaman notifikasi ini membuat notifikasi tidak terasa sempit sehingga lebih transparan dan mudah dibaca.

2.1.6.7. Pengimplementasian fitur “Auto-complete” pada bagian “Comment” suatu *change request*

Setiap *change request* memiliki sebuah *comment section*, dimana *developer* dan *approver* dapat berdiskusi mengenai *change request* tersebut. Tetapi, dampak dari *comment section* ini cukup kecil sehingga diskusi mengenai *change request* akan lama dibalas, padahal *change request* tersebut mungkin untuk memperbaiki sesuatu yang bersifat *urgent*. Sehingga, dibutuhkan sebuah notifikasi secara personal ke orang yang bersangkutan jika namanya disebutkan di dalam *comment section*.

Penulis mengimplementasikan fitur ini dengan menggunakan *AtWho.js*. *Javascript* tersebut berfungsi untuk memberikan *suggestions* ketika karakter ‘@’ diketik pada *comment section*. Maka, dengan mengetik karakter ‘@’, akan dimunculkan daftar *suggestions* yang berasal dari keseluruhan *user* pada sistem *Vision*. Kemudian, jika *comment* dikirim, maka sistem *backend* akan memberikan notifikasi melalui *Slack* ke orang yang bersangkutan, sehingga *comment* tersebut diharapkan akan cepat dilihat dan direspons.

2.1.6.8. Pengimplementasian fitur “Quick Filter”

Sebagai pengguna sistem *Vision*, jika pengguna melihat daftar seluruh *change request*, maka akan ada banyak *change request* yang mungkin tidak berhubungan dengan pengguna tersebut. Meskipun sudah terdapat *sidebar* untuk melakukan *filtering*, banyaknya *field* yang harus diisi membuat hal menjadi cukup rumit. Dibutuhkan sebuah cara cepat untuk melihat *change request* mana yang berhubungan dengan kita, *change request* yang butuh disetujui, dan lain sebagainya.

Maka, penulis mengimplementasikan fitur *quick filters* tersebut. Fitur ini memiliki 3 *filter* utama yaitu ‘*All*’, ‘*Relevant to me*’ dan ‘*Need my approvals*’. Dengan menekan salah satu tombol ini, maka daftar *change request* yang ditampilkan akan sesuai dengan *filter* yang dipilih.

2.1.6.9. Pengimplementasian fitur “Change Request Lifeline”

Pada PT Midtrans, sebuah *change request* memiliki alur hidupnya masing-masing. Misal, apakah *change request* baru dimasukkan, sudah disetujui, sudah diimplementasi ke *production*, dan lain sebagainya. Fitur yang sudah ada dianggap kurang jelas dalam merepresentasikan *journey* dari *change request* tersebut. *Journey* ini juga hanya dapat dilihat oleh pembuat *change request* dan *approver*, sehingga *developer* lain yang terkait tidak dapat mengetahui sudah sejauh apa proses *journey change request* tersebut. Kurangnya informasi mengenai langkah apa yang bisa diambil berikutnya, dan alasan-alasan lainnya memunculkan kebutuhan adanya perbaikan di fitur ini.

Penulis melakukan perbaikan dengan mengubah cara pemilihan *journey* dari *lifeline* tersebut. Misal, pada awalnya *change request* akan memiliki status *submitted*, kemudian, akan dapat dilihat oleh pembuat *change request* apa saja yang bisa dilakukan untuk ke status berikutnya. Kemudian, *lifeline* ini juga dibuat agar terlihat oleh seluruh *developer* (opsi memilih langkah selanjutnya hanya terlihat oleh pembuat *change request* dan *approver*). Perbaikan juga dilakukan secara tampilan agar terlihat lebih menonjol dan lebih mudah dibaca.

2.1.6.10. Penyesuaian Unit Testing untuk setiap fitur yang diimplementasikan

Untuk setiap fitur yang telah diimplementasikan, penulis diwajibkan oleh penyelia untuk membuat *unit testing* yang bersesuaian. *Unit testing* yang dibuat menggunakan Rspec sebagai basis *framework testing*. Vision telah memiliki beberapa *unit test* yang telah dibuat, sehingga penulis perlu menyesuaikan *unit tests* tersebut dengan sistem dan fitur-fitur yang baru. Untuk keseluruhan *unit testing*, diharapkan bahwa *code coverage* berada di atas 75%, sehingga banyak *unit test* yang harus dibuat untuk memenuhi *code coverage* tersebut.

2.1.6.11. Pembuatan Acceptance Testing untuk setiap fitur yang diimplementasikan

Acceptance testing adalah salah satu bentuk *testing* yang dibuat untuk memastikan bahwa program yang dibuat sesuai dengan spesifikasi dari konsumen dan konsumen dapat menerima atau menyetujui program tersebut. *Framework* yang digunakan adalah Cucumber. *Tests* yang dibuat dengan menggunakan Cucumber dan Capybara bersifat *behavioral*, sehingga seakan-akan konsumen memang sedang melakukan aksi tersebut (misal, mengetik “halo”, mengklik tombol “submit”). Dalam proses pembuatan *acceptance tests*, salah satu *plugin* yang sangat membantu adalah Capybara-screenshot. Jika terjadi kesalahan ketika *test* dijalankan, maka Capybara-screenshot akan menangkap layar terjadinya kesalahan tersebut, dan *screenshot* ini digunakan untuk menganalisis penyebab terjadinya kesalahan.

2.2. Analisis

Pada subbab ini, akan dijelaskan mengenai pelaksanaan kerja praktik, kendala-kendala yang ditemui, perbandingan hal yang telah dilakukan dengan KAKP yang telah ditentukan di awal, dan relevansi antara kerja praktik dengan mata kuliah yang telah diambil di Fasilkom.

2.2.1. Pelaksanaan Kerja Praktik

Dalam pelaksanaan kerja praktik, terdapat beberapa perbedaan yang terjadi dengan KAKP yang telah dibuat di awal. Hal ini dikarenakan prioritas proyek Vision lebih tinggi daripada proyek Procurement. Selain itu, penulis mengalami beberapa kendala yang membuat kinerja penulis dalam kerja praktik terhambat.

2.2.1.1. Kesesuaian dengan KAKP

Seperti telah dijelaskan sebelumnya, terjadi perubahan kerangka pekerjaan dari KAKP. Pada awalnya, penulis direncanakan akan mengerjakan dua buah proyek, yaitu Vision dan Procurement. Pembagian waktu pengerjaan tiap proyek pun telah dilakukan, 2 bulan pertama untuk Vision, 1 bulan terakhir untuk Procurement. Tetapi, seiring jalan, penyelia mengatakan bahwa lebih baik tim penulis fokus ke proyek

Vision terlebih dahulu, karena Vision digunakan sehari-hari dan banyak fitur yang harus diimplementasikan. Banyaknya *story* yang berdatangan dari para *developer* juga menjadi alasan mengapa ketidaksesuaian dengan KAKP ini terjadi.

2.2.1.2. Kendala yang dihadapi

Dalam melaksanakan kerja praktik, penulis menemui beberapa kendala yang mempengaruhi keseluruhan kerja praktik.

- Seperti yang telah dikatakan sebelumnya, *story* untuk Vision terus bertambah. Seselesainya penulis dengan sebuah fitur, akan ada sebuah *story* baru di Trello yang membuat *planning* setiap minggu membutuhkan analisis yang cukup mendalam mengenai prioritas masing-masing *story*.
- Penyelia penulis yang cukup sibuk membuat proses *review* menjadi terhambat. Karena terhambat, penulis menjadi bingung harus melakukan apa karena belum boleh mengambil *story* yang baru tanpa seizin penyelia.
- Kurangnya pengetahuan penulis mengenai cara melakukan *testing*, baik *unit test* maupun *acceptance test* membuat penulis lebih banyak belajar terlebih dahulu sebelum melakukan *testing*. Hal ini membuat proses *testing* berjalan agak lambat, meskipun seiring berjalannya kerja praktik terjadi peningkatan dalam pembuatan *testing* tersebut.

2.2.1.3. Penilaian terhadap Tempat Kerja Praktik

Menurut penulis, PT Midtrans merupakan tempat kerja praktik yang baik. *Work environment* di Midtrans sangat ramah, tidak adanya diskriminasi antara pekerja yang sedang magang maupun pekerja yang sudah *full-time*. Kepemimpinan dari CEO Midtrans, Ryu Kawano, sangat menginspirasi penulis bahwa seorang CEO tidak perlu membedakan tingkat dirinya dengan orang lain. Semua orang, dari karyawan, janitor, sampai staf sekuriti dianggap setara. Adanya acara setiap hari Jumat juga membuat antar staf saling berbaur satu sama lain.

Sebagai *developer* yang melakukan magang, penulis dapat belajar banyak dari seluruh karyawan, baik dari segi etika, *best practices*, dan *time management*. Proyek-proyek yang ditawarkan juga memacu para karyawan magang untuk berpikir kritis untuk memecahkan masalah dan memberikan solusi yang terbaik.

2.2.2. Relevansi dengan Perkuliahan

Dalam pelaksanaan kerja praktik, penulis menemukan beberapa relevansi dengan mata kuliah yang telah penulis ambil selama perkuliahan di Fasilkom UI.

2.2.2.1. Dasar-dasar Pemrograman (DDP) dan Struktur Data Algoritma (SDA)

Pada mata kuliah DDP, penulis belajar mengenai dasar-dasar dari pemrograman yang sangat penting untuk penulis dapat melakukan kerja praktik. Meskipun bahasa yang dipelajari berbeda, esensi dari program dan algoritma tidaklah berbeda. Konsep *Object Oriented Programming* (OOP) yang diajarkan membuat penulis mudah memahami mengenai permodelan objek di Ruby on Rails. Pada mata kuliah SDA, penulis belajar mengenai jenis-jenis struktur data, seperti *collection* yang penulis gunakan dalam menampung seluruh *user*, dan lain sebagainya.

2.2.2.2. Perancangan dan Pembuatan Web (PPW)

Pada mata kuliah PPW, penulis belajar mengenai dasar-dasar dari proses pembuatan suatu web yaitu HTML, CSS, dan Javascript. Penggunaan Javascript pada beberapa halaman merupakan hasil pembelajaran pada mata kuliah ini. Meskipun *framework* yang dipelajari berbeda, tetapi basis pengetahuan mengenai sebuah *framework* sudah cukup membantu dalam proses kerja praktik ini.

2.2.2.3. Rekayasa Perangkat Lunak (RPL) dan Proyek Perangkat Lunak (PPL)

Dalam mata kuliah RPL, penulis belajar mengenai metodologi *development*, seperti yang penulis gunakan saat melaksanakan kerja praktik yaitu *scrum*. Pembuatan berbagai diagram juga sangat membantu untuk dapat memvisualisasikan ide penulis ke tim dan penyelia. Dalam mata kuliah PPL, karena penulis menggunakan Ruby on Rails, maka tentu saja sangat membantu dalam pelaksanaan kerja praktik. Dasar-dasar yang didapat selama menjalani kuliah PPL, seperti melakukan *testing*,

microservices, dan lain sebagainya telah diaplikasikan secara efektif pada pelaksanaan kerja praktik.

2.2.2.4. Penjaminan Mutu Perangkat Lunak (PMPL)

Pada mata kuliah PMPL, penulis mempelajari mengenai berbagai macam *testing* dan metode-metode yang dilakukan untuk menjalankan *software quality assurance*. Sayangnya, pada mata kuliah tersebut lebih banyak ke teori, sehingga praktik langsungnya kurang didapat, dan lebih didapat pada kerja praktik.

2.2.2.5. Basis Data dan Basis Data Lanjut

Pada mata kuliah basis data, penulis belajar mengenai dasar-dasar penyimpanan data di komputer, yang sangat membantu penulis dalam memvisualisasikan permodelan objek-objek pada Ruby on Rails ke dalam tabel-tabel. Pada basis data lanjut, penulis belajar mengenai proses *indexing* pada tabel, untuk mempercepat *query* yang dilakukan. Hal ini digunakan terutama pada proses pencarian melalui Apache Solr.

BAB 3

PENUTUP

3.1. Kesimpulan

Penulis melakukan kerja praktik selama kurang lebih 10 minggu dengan tugas melakukan perbaikan dan penyempurnaan terhadap aplikasi Vision. Vision adalah sebuah aplikasi yang berfungsi sebagai sebuah portal untuk melakukan *change request* dan *incident report management*. Penulis telah berhasil mengimplementasikan *story* dan perbaikan yang diajukan, serta menawarkan solusi untuk masalah-masalah yang ada. Meskipun terdapat beberapa kendala dalam pelaksanaan kerja praktik, proses *development* tetap berlangsung dengan baik.

Kerja praktik yang penulis lakukan di PT Midtrans sangat membuka mata penulis mengenai dunia kerja. Sebelumnya, penulis masih sangat tertutup pengetahuannya mengenai apakah yang akan dilakukan setelah selesai dari kuliah nanti. Melakukan kerja praktik di PT Midtrans telah membuka peluang penulis dalam pengembangan diri, baik dari segi teknis maupun non-teknis. Banyak pengalaman baru yang telah penulis dapatkan dan akan menjadi dasar dari pengambilan keputusan kedepannya.

3.2. Saran

Penulis memiliki beberapa saran untuk pihak-pihak yang terkait dengan pelaksanaan kerja praktik periode ini:

1. Mahasiswa Fasilkom

- Gunakan seluruh koneksi untuk mencari internship
- Mulai mencari dari awal sehingga tidak kesusahan sendiri mencari tempat kerja praktik di akhir-akhir

- Pelajari TDD sedini mungkin, karena akan sangat berguna untuk pengembangan aplikasi kedepannya

2. Fasilkom

- Informasi mengenai kelas KP lebih baik dijelaskan sejak awal sehingga tidak ada misinformasi di tengah proses kerja praktik
- Samakan informasi mengenai KP dari masa KAKP hingga masa akhir kelas

3. PT Midtrans

- Program internship diteruskan dan dikembangkan agar mahasiswa dapat terus mengembangkan dirinya dengan melakukan proses magang
- Jabatan mentor kalau bisa jangan terlalu tinggi, karena sulit untuk berkomunikasi ketika mentor sedang sibuk

DAFTAR REFERENSI

- [1] "Midtrans - Solusi Untuk Bisnis Online." Midtrans - Solusi Untuk Bisnis Online. N.p., n.d. Web. 04 Dec. 2016. <<https://midtrans.com/about>>
- [2] "Midtrans - Atur Risiko Bisnis Online Dengan AEGIS." Midtrans - Atur Risiko Bisnis Online Dengan AEGIS. N.p., n.d. Web. 04 Dec. 2016. <<https://midtrans.com/aegis>>
- [3] "Prism - Chat That Sells, Literally." Prism. N.p., n.d. Web. 04 Dec. 2016. <<http://www.prismapp.io/>>
- [4] "Scrum Values." Scrum Values | Agile Manifesto | Scrum Principles - Scrum Alliance. N.p., n.d. Web. 04 Dec. 2016. <<https://www.scrumalliance.org/why-scrum/core-scrum-values-roles>>
- [5] "What Is Unit Testing? - Definition from WhatIs.com." SearchSoftwareQuality. N.p., n.d. Web. 04 Dec. 2016. <<http://searchsoftwarequality.techtarget.com/definition/unit-testing>>
- [6] "Acceptance Testing." Software Testing Fundamentals. N.p., n.d. Web. 04 Dec. 2016. <<http://softwaretestingfundamentals.com/acceptance-testing/>>
- [7] "Code Refactoring." Wikipedia. Wikimedia Foundation, n.d. Web. 04 Dec. 2016. <https://en.wikipedia.org/wiki/Code_refactoring>
- [8] "Git - Beginner to Advanced Commands and Git Workflows." Atlassian Git Tutorial. N.p., n.d. Web. 04 Dec. 2016. <<https://www.atlassian.com/git/>>
- [9] "About Trello." About Trello. N.p., n.d. Web. 04 Dec. 2016. <<https://trello.com/about>>
- [10] "Imagine What You Could Build If You Learned Ruby on Rails...." Ruby on Rails. N.p., n.d. Web. 04 Dec. 2016. <<http://rubyonrails.org>>

- [11] "Model View Controller." Wikipedia. Wikimedia Foundation, n.d. Web. 04 Dec. 2016. <<https://en.wikipedia.org/wiki/Model-view-controller>>
- [12] "A Hackable Text Editor for the 21st Century." Atom. N.p., n.d. Web. 04 Dec. 2016. <<https://atom.io>>
- [13] "RSpec." About RSpec. N.p., n.d. Web. 06 Dec. 2016. <<http://rspec.info/about/>>
- [14] Limited, Cucumber. "Simple, Human Collaboration." Cucumber. N.p., n.d. Web. 07 Dec. 2016. <<https://cucumber.io>>
- [15] "About Us." GitLab. N.p., n.d. Web. 07 Dec. 2016. <<https://about.gitlab.com/about/>>

LAMPIRAN 1
KERANGKA ACUAN KERJA PRAKTIK

LAMPIRAN 2
LOG KERJA PRAKTIK