



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/format-kan, 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, mulai tanggal 13 Juni 2016 sampai 26 Agustus 2016. Dalam kerja praktik, penulis memiliki peran sebagai *software engineering intern* dibawah pengawasan penyelia penulis yaitu Rafeequl Rahman, *Vice President of Engineering* di 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 kepada kami 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

DAFTAR GAMBAR

DAFTAR LAMPIRAN

BAB 1

PENDAHULUAN

1.1. Proses Pencarian Kerja Praktik

Pencarian kerja praktik pertama kali penulis lakukan atas dasar saran dari kakak penulis yang sedang bekerja di Singapura. Perusahaan yang disarnakan 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 pengoptimasian *business process* yang dimiliki perusahaan tersebut. Pengajuan 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 tahun depannya, yaitu 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, dan terdapat lowongan yang kosong. Penulis pun mencari tahu lebih lanjut dan mengirimkan CV ke saudara Nayana agar dapat diteruskan ke PT Suitmedia sebagai rekomendasi. Tepat dua hari setelah mengajukan CV, penulis diberikan kabar dari pihak HRD PT Suitmedia untuk datang ke 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 seleksinya pada minggu yang mendatang.

Tetapi, masa tersebut adalah masa dimana penulis sedang mengikut 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* yang kosong dikarenakan adanya pelamar yang mengundurkan diri. Mendengar hal tersebut, penulis memantapkan diri 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

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 motto *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 divis *Engineering* yang diketuai oleh Rafeequl Rahman. 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 dan Dwiyan lakukan.

BAB 2

ISI

2.1. Pekerjaan dalam 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 dilakukan 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 kami 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. Dan pada tiap harinya, 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 kedalam 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 membagi tugas mana yang akan dikerjakan oleh masing-masing orang. 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 *men-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 Chelimsky 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. Hasil Pekerjaan

2.2. Analisis

2.2.1. Pelaksanaan Kerja Praktik

2.2.2. Relevansi dengan Perkuliahan di Fasilkom UI

BAB 3

PENUTUP

3.1. Kesimpulan

3.2. Saran

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/>>

