



UNIVERSITAS INDONESIA

***SOFTWARE ENGINEER OF THAILAND BANK STATEMENT
FETCHER FOR TRAVELOKA PAYMENT MISSION***

LAPORAN KERJA PRAKTIK

WIRATMIKA

1306402873

**FAKULTAS ILMU KOMPUTER
PROGRAM STUDI ILMU KOMPUTER
DEPOK
OKTOBER 2016**

HALAMAN PERSETUJUAN DOSEN MATA KULIAH KERJA PRAKTIK

Laporan ini diajukan oleh :
Nama : Wiratmika
NPM : 1306402873
Program Studi : Ilmu Komputer
Judul Kerja Praktik : *Software Engineer of Thailand Bank Statement
Fetcher for Traveloka Payment Mission*

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,

Rahmad Mahendra, S.Kom., M.A., M.Sc.

Ditetapkan di :

Tanggal :

ABSTRAK

Penulis melakukan kerja praktik dalam rangka memenuhi mata kuliah kerja praktik semester ganjil 2016/2017 Fakultas Ilmu Komputer Universitas Indonesia. Pelaksanaan kerja praktik dilakukan di Traveloka pada tanggal 13 Juli hingga 12 September 2016. Penulis ditempatkan di *product team* bagian *payment mission* untuk membuat beberapa *fetcher* bank di Thailand dalam rangka peningkatan efisiensi operasional Traveloka. Tulisan ini akan membahas antara lain proses pencarian kerja praktik, pekerjaan yang dibuat, ilmu yang dipelajari, serta kendala yang ditemukan. Penulis memanfaatkan banyak teknologi dan bantuan untuk menyelesaikan tugas tersebut. Secara umum, pelaksanaan kerja praktik sesuai dengan kerangka acuan kerja praktik, namun memiliki beberapa perbedaan di jadwal kerja. *Fetcher* yang dibuat oleh penulis akan digunakan untuk jangka panjang dalam menunjang proses bisnis dan ekspansi Traveloka ke depannya.

Kata kunci: *bank statement fetcher*, kerja praktik, *software engineering*, Traveloka

DAFTAR ISI

HALAMAN PERSETUJUAN DOSEN MATA KULIAH KERJA PRAKTIK	i
ABSTRAK	ii
DAFTAR ISI.....	iii
DAFTAR LAMPIRAN.....	v
BAB 1 PENDAHULUAN	1
1.1. Proses Pencarian Kerja Praktik.....	1
1.2. Tempat Kerja Praktik.....	3
1.2.1. Profil Tempat Kerja Praktik.....	3
1.2.2. Posisi Penempatan Pelaksana Kerja Praktik dalam Struktur Organisasi ..	4
BAB 2 PELAKSANAAN KERJA PRAKTIK	5
2.1. Latar Belakang Pekerjaan	5
2.2. Tinjauan Pustaka.....	6
2.3. Metodologi.....	9
2.4. Teknologi.....	9
2.5. Proyek Kerja dan Deliverables	11
2.6. Non-Teknis	13
BAB 3 ANALISIS PEKERJAAN KERJA PRAKTIK.....	14
3.1. Kesesuaian dan Perbedaan dengan KAKP	14
3.2. Kendala dalam Kerja Praktik dan Cara Penanganannya	15
3.3. Pembelajaran Soft Skill selama Kerja Praktik.....	15
3.4. Penilaian Individu terhadap Tempat Kerja Praktik	17
3.5. Relevansi dengan Perkuliahan di Fasilkom UI.....	17

BAB 4 PENUTUP	19
4.1. Kesimpulan	19
4.2. Saran	20
DAFTAR REFERENSI	21
LAMPIRAN 1 KERANGKA ACUAN KERJA PRAKTIK	viii
LAMPIRAN 2 LOG KERJA PRAKTIK.....	ix

DAFTAR LAMPIRAN

[Daftar Lampiran, ditulis serupa dengan daftar isi]

BAB 1

PENDAHULUAN

1.1. Proses Pencarian Kerja Praktik

Penulis bercita-cita untuk merintis perusahaan yang mengembangkan produk berbasis teknologi khususnya perangkat lunak. Untuk mempersiapkan hal tersebut, sejak awal penulis berniat untuk mencari pekerjaan sebagai *software engineer* terutama untuk kerja praktik agar penulis bisa belajar mengenai pengembangan perangkat lunak langsung dari praktisi industri.

Ada beberapa kriteria spesifik mengenai perusahaan yang dicari oleh penulis, yaitu:

- Memandang teknologi sebagai bagian dari nilai utama perusahaan, bukan sekedar alat bantu untuk mencapai tujuan bisnis
- Berukuran menengah, memiliki setidaknya 50 karyawan namun bukan korporat yang sudah terlalu mapan
- Memiliki produk yang signifikan di mata penulis, dengan pasar dan masalah konkret yang tervalidasi
- Memiliki budaya kerja yang kondusif dan progresif
- Memiliki visi jangka panjang yang jelas
- Didirikan oleh orang-orang dengan reputasi dan rekam jejak yang baik

Tujuan utama dari kriteria-kriteria tersebut adalah penulis ingin mendapat tanggung jawab dan pengalaman teknis sebanyak mungkin namun tetap dengan proporsional dengan kapabilitas penulis. Selain itu, perusahaan yang berukuran menengah biasanya sedang dalam fase pertumbuhan yang cepat sehingga bisa memberikan ilmu dan potensi pengembangan diri yang seharusnya paling besar.

Dimulai sejak Maret 2016, penulis melamar di lima perusahaan yang sesuai dengan kriteria tersebut. Kelima perusahaan tersebut adalah Traveloka, Veritrans, Tokopedia,

Bukalapak, dan Ninja Van. Penulis mengetahui seluruh perusahaan tersebut karena pernah memakai atau mendengar produk mereka dari berbagai sumber, khususnya media *online*. Seluruh pendaftaran penulis dilakukan melalui situs perusahaan masing-masing atau melalui surat elektronik (surel).

Ninja Van adalah perusahaan rintisan berbasis di Singapura yang bergerak di bidang logistik *last-mile*. Penulis tertarik ke Ninja Van karena penulis juga memiliki keinginan untuk bekerja di Singapura, dan dari semua perusahaan teknologi Singapura yang ditemukan oleh penulis, Ninja Van merupakan pilihan yang paling menarik. Hal itu karena ukurannya sesuai dengan kriteria penulis dan mereka memanfaatkan teknologi sebagai nilai tambah utama produk mereka. Sayangnya, Ninja Van merespons penulis dengan menyatakan bahwa mereka tidak menerima karyawan magang dari luar negeri.

Bukalapak adalah perusahaan *online marketplace* yang berbasis di Kemang. Meski sudah mendaftar melalui situs mereka, Bukalapak tidak memberi respons apa pun, dan penulis menduga posisi *software engineer* sudah penuh terisi. Sementara itu, Tokopedia, perusahaan kompetitor Bukalapak yang berbasis di Slipi, menawarkan kesempatan wawancara untuk penulis. Setelah melalui proses wawancara yang terdiri dari pertanyaan kepribadian dan teknis, penulis mendapat tawaran kerja praktik sebagai *software engineer intern*. Pada saat itu, penulis langsung menerima tawaran dari Tokopedia.

Veritrans adalah perusahaan *payment gateway* yang berbasis di Jalan Sudirman. Veritrans memberikan respons melalui surel dan menawarkan penulis untuk mengikuti proses seleksi lebih lanjut melalui tes di tempat. Namun, karena lebih memilih Tokopedia, penulis tidak mengikuti tes tersebut.

Traveloka adalah perusahaan *online travel agent* yang memiliki kantor di gedung yang sama dengan Tokopedia. Beberapa minggu setelah menerima tawaran dari Tokopedia, penulis dihubungi oleh *human resources* Traveloka dan menjalani wawancara teknis singkat melalui telepon. Setelah itu, penulis diberi kesempatan untuk menjalani proses wawancara tatap muka di Traveloka meskipun penulis sudah menyatakan ke mereka

bahwa penulis sudah diterima di Tokopedia. Proses wawancara di Traveloka nyaris tidak memiliki pertanyaan mengenai kepribadian dan sangat menitik beratkan pada pertanyaan teknis.

Pada akhirnya, penulis juga mendapat tawaran kerja praktik di Traveloka dengan posisi yang sama, yaitu sebagai *software engineer intern*. Setelah berkontemplasi mengenai kelebihan dan kekurangan kedua pilihan yang ada serta berdiskusi dengan orang tua, penulis memutuskan untuk melepas posisi di Tokopedia dan menerima tawaran dari Traveloka.

1.2. Tempat Kerja Praktik

Penulis melakukan kerja praktik selama tiga bulan di Traveloka, yang menjalankan bisnis sebagai PT Trinusa Travelindo.

1.2.1. Profil Tempat Kerja Praktik

Traveloka adalah layanan *online travel agent* di Asia Tenggara yang menyediakan layanan pemesanan tiket pesawat dan hotel secara *online*. Traveloka memiliki basis operasional di Jakarta, dan didirikan pada tahun 2016 oleh tiga orang *engineer* di Silicon Valley, yaitu Ferry Unardi, Derianto Kusuma, dan Albert Zhang. Selain di Indonesia, pada saat tulisan ini dibuat, Traveloka sudah beroperasi di beberapa negara di Asia Tenggara, yaitu Filipina, Malaysia, Singapura, Thailand, dan Vietnam, dengan lebih dari 1000 karyawan.

Traveloka memiliki visi jangka panjang untuk menjadi *mobility company*, yaitu mengubah cara manusia melakukan mobilisasi dengan memanfaatkan teknologi. Kantor pusat Traveloka terletak di gedung Wisma 77, dengan kantor cabang lokal di Bali. Traveloka memiliki struktur organisasi yang bersifat terdesentralisasi dan terdiri dari berbagai tim dengan *area of concern* yang berbeda-beda. Setiap tim memiliki struktur dan peran anggota yang berbeda-beda tergantung dengan kebutuhan dan tugas yang diemban.

1.2.2. Posisi Penempatan Pelaksana Kerja Praktik dalam Struktur Organisasi

Selama menjalani kerja praktik di Traveloka sebagai *software engineer intern*, penulis ditempatkan di *product team* bagian *payment mission*, yaitu tim yang membuat aplikasi dan layanan yang berkaitan dengan penerimaan pembayaran dari pengguna. Tujuan dari *payment mission* adalah untuk mendukung salah satu pilar bisnis Traveloka, yaitu pembayaran, dengan cara membuat proses pembayaran menjadi mudah dan memiliki friksi serendah mungkin.

Lebih spesifiknya lagi, penulis bekerja untuk *payment acquiring*, yaitu tim yang bertanggung jawab dalam menambah dan mengelola kanal pembayaran untuk memungkinkan pengguna membayar dengan cara yang lebih bervariasi. Dari sudut pandang bisnis, *payment acquiring* merupakan salah satu kepentingan yang paling krusial karena merupakan faktor prasyarat dalam membuka pasar baru, atau dengan kata lain, ekspansi ke luar negeri.

BAB 2

PELAKSANAAN KERJA PRAKTIK

2.1. Latar Belakang Pekerjaan

Di dalam aplikasi Traveloka, baik *web*, *mobile*, maupun aplikasi *smartphone*, terdapat banyak metode pembayaran yang disesuaikan dengan kanal yang tersedia di suatu negara, seperti transfer bank, kartu kredit, dan gerai ritel. Transfer bank merupakan metode pembayaran yang paling populer karena mayoritas pengguna Traveloka memiliki rekening bank. Selain itu, transfer antar bank yang sama tidak terkena biaya, sehingga biaya operasionalnya lebih rendah dibanding metode lain seperti kartu kredit yang harus membayar biaya untuk pihak ketiga. Karena proposisi tersebut, membuka kanal pembayaran dengan metode transfer bank di seluruh negara tempat Traveloka beroperasi menjadi salah satu prioritas dari *payment acquiring*. Pilihan bank yang beragam akan menjadi nilai lebih di mata pengguna.

Pada saat tulisan ini dibuat, Traveloka sudah menerima sangat banyak pembayaran melalui transfer. Penulis pernah mendapat informasi bahwa terdapat sekitar puluhan ribu transaksi transfer per hari ketika penulis sedang melakukan kerja praktik. Angka tersebut akan terus bertambah dengan semakin banyaknya pengguna dan negara tempat Traveloka beroperasi. Masalah utama dari pemrosesan transfer dalam konteks penunjang layanan *online* saat ini adalah, bank tidak memiliki *application programming interface* (API) atau metode apapun bagi pihak luar untuk berkomunikasi dengan sistem mutasi mereka. Untuk itu, mengecek mutasi dan mendeteksi pembayaran pengguna harus dilakukan secara manual oleh operator manusia dari tim operasional Traveloka.

Untuk membuat proses tersebut lebih efisien dan mempermudah tugas tim operasional, Traveloka mengembangkan sistem *fetcher* dengan menggunakan teknologi *in-house* buatan sendiri untuk mengambil data mutasi transaksi bank dan mengolahnya untuk dimasukkan ke dalam sistem secara otomatis, secara hampir *real-time*. Setelah data

pembayaran masuk, maka tagihan yang bersesuaian juga akan diproses secara otomatis sehingga pengguna akan langsung mendapat tiket atau *voucher* hotel langsung setelah mereka melakukan transfer.

Pada situasi normal yang diharapkan, waktu dari pengguna selesai melakukan transfer hingga mendapat konfirmasi dari sistem adalah sekitar satu menit, dengan asumsi tidak terdapat kendala pada sistem, baik milik Traveloka (sistem *payment* dan *booking*) maupun maskapai atau hotel yang bersangkutan.

Sejak pertama kali diluncurkan, performa *fetcher* Traveloka mengalami fluktuasi dalam hal availabilitas dan waktu respons, baik karena faktor internal seperti pengubahan arsitektur *fetcher* maupun faktor eksternal seperti ketersediaan situs *internet banking* masing-masing bank. *Fetcher* terus diperbaiki dan dikembangkan untuk meningkatkan performa dan kapabilitasnya.

Pada saat penulis melakukan kerja praktik, *fetcher* Traveloka sudah diimplementasikan untuk hampir seluruh bank besar di Indonesia. Tugas penulis adalah untuk membuat sistem yang serupa untuk bank-bank besar di Thailand. Ranah pekerjaan penulis adalah mempelajari situs *internet banking* beberapa bank di Thailand beserta keterbatasannya, mengadaptasi *fetcher* untuk mengambil dan mengolah mutasi rekening Traveloka, dan melakukan uji coba untuk sistem yang telah dibuat.

2.2. Tinjauan Pustaka

Selama satu setengah minggu pertama kerja praktik, penulis mempelajari konvensi penulisan kode, *codebase* dan arsitektur sistem yang sudah berjalan, serta alur kerja pengembangan perangkat lunak yang digunakan oleh Traveloka.

Traveloka menggunakan layanan *knowledge management* yaitu Confluence. Confluence adalah perangkat lunak kolaborasi tim untuk korporat yang dikembangkan dan dipasarkan oleh Atlassian [1]. Di dalam Confluence, terdapat seluruh informasi yang terkait dengan Traveloka, antara lain prosedur standar rekayasa perangkat lunak, informasi bisnis, alur teknis produk, hingga perencanaan untuk pengembangan produk

baru. Penulis memanfaatkan Confluence untuk mempelajari informasi sebanyak mungkin khususnya dalam segi teknis produk.

Selain informasi yang terdapat di Confluence, penulis juga mempelajari hal lain yang terkait dengan pembuatan perangkat lunak terutama dengan konsiderasi arsitektur yang digunakan oleh Traveloka. Poin-poin tersebut adalah:

1. *Asynchronous programming*

Hal utama yang telah dipelajari oleh penulis adalah mengenai *async programming*, karena penulis belum pernah mendalaminya sebelumnya. *Async programming* memungkinkan kita untuk menulis kode yang tidak memblok proses lain, di mana sebuah *method* yang berpotensi menghabiskan waktu lama bisa dilewatkan terlebih dahulu sehingga proses lain bisa berjalan. Apabila *method* tersebut telah selesai dijalankan, barulah proses lain yang disebut *callback* akan dijalankan.

Meski konsepnya sederhana, namun pengimplementasiannya dalam bahasa Java tidak begitu mudah. Bagi yang jarang menggunakannya, sintaksnya akan terasa aneh karena tidak begitu sederhana. Terkadang sebuah *async method* bisa merupakan *callback* dari *async method* lain sehingga berpotensi menghasilkan kode yang mengandung *callback hell*.

Async programming diperlukan untuk *fetcher* Traveloka karena operasinya melibatkan koneksi dengan server lain menggunakan protokol *hypertext transfer protocol* (HTTP). Dengan menggunakan *async method*, ketika *fetcher* sedang mengambil data dari situs *internet banking*, proses lainnya tidak terhenti.

2. *Distributed network architecture* dan *remote procedure call*

Besarnya jumlah pengguna Traveloka mengharuskan sistem yang memiliki kapasitas dan skalabilitas yang tinggi. Untuk memudahkan pengembangan dan pemeliharaan, layanan Traveloka dipecah menjadi banyak subkomponen yang saling berkomunikasi satu sama lain dengan menggunakan *remote procedure call* (RPC). Dengan begitu, satu

subkomponen bisa dipecah lagi menjadi beberapa *server* di balik *load balancer* untuk menjamin *availabilitas*, *redundansi*, dan *performa*.

Untuk itu, meskipun masih berada di dalam satu *repository*, *codebase* Traveloka dipecah menjadi beberapa komponen besar. Terdapat dua komponen utama yang dipelajari dan disentuh oleh penulis, yaitu PG dan TAP.

PG adalah komponen *payment gateway* yang bertugas untuk melayani seluruh kebutuhan terkait pembayaran. Karena transfer bank berada di dalam konteks pembayaran, maka sistem *fetcher* berada di dalam PG. Selain *fetcher*, seluruh metode pembayaran lain juga berada di dalam PG. TAP adalah platform untuk mengelola operasional sehari-hari Traveloka, dan sistem *fetcher* memerlukannya agar proses *fetcher* bisa dijalankan dan dikonfigurasi oleh tim operasional Traveloka yang mengurus pembayaran.

3. Hypertext transfer protocol

Karena *fetcher* melibatkan banyak interaksi dengan server lain menggunakan HTTP, maka penulis juga mendalami lebih lanjut konsep terkait HTTP. Konsep-konsep dalam protokol HTTP yang dipelajari oleh penulis adalah mengenai pola *request-response*, *method*, *header*, *body*, *status code*, *cookie*, dan *session*. Selain itu, *fetcher* juga melibatkan penggunaan *hypertext markup language* (HTML) dan *scraping* menggunakan *library* jsoup.

4. Software Development Life Cycle (SDLC)

SDLC adalah sebuah istilah yang mencakup banyak hal. Secara definisi, SDLC adalah model konseptual yang menjelaskan tahap-tahap yang ada dalam proyek pengembangan sistem informasi, dari studi *feasibilitas* awal hingga pemeliharaan aplikasi yang sudah selesai [2]. Agar kode yang baru dibuat bisa dipakai di Traveloka, kode tersebut harus melalui serangkaian proses yang terstandarisasi. Apa yang dipelajari oleh penulis mengenai SDLC lebih mengacu ke implementasi dan justifikasi di baliknya.

2.3. Metodologi

Product team Traveloka menerapkan prinsip *agile* tanpa menggunakan kerangka kerja tertentu secara murni. Hal tersebut menyebabkan perencanaan kerja bersifat fleksibel dan tentatif, mengikuti kebutuhan bisnis dan proses pengembangan produk yang sedang berjalan. Traveloka menggunakan Confluence dan Google Docs untuk proses *design review* serta Asana untuk melacak pekerjaan dan koordinasi antar tim. Ketika penulis melakukan kerja praktik, Traveloka sedang merencanakan untuk beralih menggunakan JIRA.

Tim *payment* tidak mengadakan rapat formal rutin, melainkan hanya membuat laporan tiap minggu yang berisi hal yang sudah dilaksanakan, kendala yang dihadapi, dan hal yang akan dikerjakan. Laporan tersebut dikirim melalui surel ke milis tim *payment* agar bisa dibaca oleh semua orang. Namun, sesekali tim *payment* juga melakukan pertemuan langsung untuk berbagi ilmu dan pengalaman yang dipelajari oleh sesama anggota tim.

Setiap kode yang dibuat harus melalui proses *code review* terlebih dahulu sebelum bisa dimasukkan ke *branch develop*. Code review tidak hanya bersifat basa-basi, melainkan benar-benar dilakukan menyeluruh di mana kode tidak akan bisa dimasukkan apabila belum di-review dan disetujui oleh reviewers. Hal ini diperlukan untuk mejmain kebenaran dan konssistensi kode serta engineer lain dari tim yang sama bisa saling mengetahui pekerjaan satu sama lain.

2.4. Teknologi

Sebelum melakukan kerja praktik, penulis sudah mengetahui sedikit mengenai gambaran *technology stack* yang digunakan Traveloka setelah membaca tulisan CTO Traveloka, Derianto Kusuma, di Quora. Meski tulisan tersebut ditulis beberapa tahun yang lalu, banyak informasi yang masih valid sampai ketika penulis melakukan kerja praktik.

Traveloka menggunakan bahasa pemrograman Java untuk hampir seluruh kode yang dibuat, baik *backend* maupun *frontend*. *Template engine* yang digunakan untuk

menghasilkan HTML adalah Apache Velocity. Dulu, Traveloka menggunakan *framework* Play versi Java, akan tetapi, karena *framework* tersebut tidak sesuai dalam memenuhi kebutuhan yang ada, maka Traveloka sekarang sudah beralih menggunakan *framework* buatan Traveloka sendiri.

Database management system utama yang digunakan adalah MongoDB karena banyak data Traveloka yang bersifat sesaat, dengan *accessor* yang sudah diimplementasikan dalam bahasa Java. Ketika penulis melakukan kerja praktik, sedang terjadi proses migrasi ke PostgreSQL untuk menangani data *transaksional* untuk menjamin integritas data dengan lebih baik.

Traveloka banyak menggunakan *library* dan perangkat lunak *open source*. Misalnya, Jetty digunakan sebagai *embedded server* baik dalam lingkungan *development*, *staging*, maupun *production*. Git digunakan sebagai *version control system* untuk seluruh *repository* yang ada.

Codebase utama Traveloka sangat besar sehingga memerlukan waktu yang lama untuk dikompilasi (30 hingga 60 menit). Untuk itu, terdapat beberapa *remote build server* untuk menjalankan proses *build* baik untuk keperluan *staging* maupun *production*. Untuk mengatur server tersebut, dulunya Traveloka menggunakan GoCD. Namun, ketika penulis melakukan kerja praktik, tim *devops* Traveloka melakukan migrasi untuk beralih ke Jenkins. *Integrated development environment* standar yang digunakan oleh *software engineer* adalah IntelliJ IDEA, kecuali untuk pengembangan aplikasi *smartphone*.

Layanan *repository hosting* yang digunakan dulunya adalah Bitbucket, namun sekarang sudah dipindah ke Phabricator yang dijalankan di *server* internal Traveloka karena Phabricator memiliki fitur dan alur *code review* yang lebih baik. *Repository* yang ada di Bitbucket masih beroperasi, namun hanya sebagai *mirror*.

Ansible digunakan untuk mengelola seluruh *server staging* dan *production*. Perangkat tersebut diperlukan karena terdapat banyak *server* yang dioperasikan oleh Traveloka,

sehingga proses yang umum dilakukan seperti *deployment* dikemas dalam kode yang bisa digunakan berulang kali. Kode itu digunakan untuk melakukan operasi ke banyak *server* secara simultan.

Selain teknologi yang sudah disebutkan di atas, ada juga beberapa perkakas yang dibuat oleh Traveloka sendiri khususnya skrip yang berkaitan dengan keperluan *devops*. Ada juga hal lain yang sudah diabstraksikan untuk langsung digunakan seperti berbagai layanan Amazon Web Services, namun penulis tidak berhubungan langsung dengan hal tersebut.

2.5. Proyek Kerja dan Deliverables

Hari pertama kerja praktik diisi oleh sesi pengenalan oleh karyawan yang baru masuk serta penjelasan singkat mengenai profil perusahaan. Setelah itu, setiap orang mendapat pembagian inventori pinjaman, di mana penulis dan sebagian besar *software engineer intern* lainnya mendapat *notebook* Lenovo ThinkPad yang sudah berisi sistem operasi Ubuntu.

Devops Traveloka membuat skrip yang akan mengkonfigurasi komputer secara otomatis. Hal yang dilakukan skrip tersebut antara lain memasang dan mengkonfigurasi perkakas *system-wide*, memasang dan mengkonfigurasi IntelliJ IDEA, dan mengunduh *repository* utama Traveloka. Meski proses dan tujuannya cukup jelas, namun menjalankan skrip tersebut hingga tuntas cukup sulit karena banyaknya masalah yang ditemukan baik oleh penulis maupun *software engineer* lain. Sebagian masalah tersebut biasanya berkaitan dengan koneksi internet yang kurang memadai atau masalah *permission* pada beberapa direktori.

Setiap karyawan baru Traveloka dianjurkan untuk membaca panduan *onboarding* yang terdapat di Confluence. Setiap posisi memiliki dokumen *onboarding* yang berbeda-beda tergantung kemampuan yang diharapkan serta pengetahuan dan perkakas yang sekiranya akan diperlukan. Di *backend engineer onboarding*, penulis membaca banyak informasi yang berkaitan dengan proses pengembangan perangkat lunak yang digunakan di Traveloka. Pada hari ketiga, penulis menghadiri pertemuan dengan

product manager untuk *payment acquiring*, Budi Waskita, untuk mendapat penjelasan mengenai tugas yang harus dikerjakan penulis.

Setelah mendapat penjelasan mengenai *fetcher* dan garis besar pekerjaan yang harus dibuat penulis, pada awalnya, penulis tidak memiliki bayangan bagaimana teknis implementasinya sama sekali. Penulis harus mempelajari terlebih dahulu sistem *fetcher* yang sudah berjalan dengan cara membaca kode untuk *fetcher* bank Thailand yang sudah pernah diimplementasikan sebelumnya.

Setelah mempelajari kode tersebut, penulis mulai mengembangkan *fetcher* untuk bank yang harus dibuat oleh penulis dengan cara melakukan uji coba berkali-kali. Hal ini dikarenakan situs *internet banking* setiap bank berbeda-beda dan sebagian besar memiliki perilaku yang sulit diprediksi. Satu-satunya cara yang paling efektif adalah dengan langsung mencoba menjalankan kodenya ke situs *internet banking* dengan rekening *dummy* atau rekening operasional agar bisa melihat langsung respons dan hasilnya.

Fetcher pertama buatan penulis selesai sekitar sebulan setelah penulis melakukan kerja praktik. Hanya dalam beberapa hari setelah peluncurannya, angka transaksi transfer untuk bank tersebut melonjak tinggi karena bank tersebut merupakan bank dengan jumlah nasabah ketiga terbesar di Thailand. Pada akhirnya, penulis berhasil menyelesaikan lima *fetcher* untuk lima bank yang berbeda sesuai dengan jadwal.

Dengan dorongan dan koordinasi dari beberapa *software engineer* lain yang lebih senior, penulis berhasil membuat *building blocks* untuk pengembangan *fetcher* selanjutnya. *Building blocks* tersebut adalah prosedur terstruktur dan terstandarisasi dalam pengembangan *fetcher*, yaitu diagram *nodes* dalam alur melakukan *scraping* serta struktur *async method* yang lebih elegan untuk mencegah *callback hell*. Hal itu diharapkan akan memudahkan pengembangan *fetcher* ke depannya khususnya bagi *software engineer* baru. Selain itu, diharapkan kode implementasi dan dokumentasinya lebih rapi karena sudah memiliki standar.

Selain membuat *fetcher* beserta *building block*nya, penulis juga mendokumentasikan kode dan seluruh pekerjaan penulis ke dalam Confluence agar pembuat *fetcher* berikutnya bisa mengetahui apa kendala dan potensi masalah yang ada selama pembuatan *fetcher*. Selama belum ada sistem yang akan menggantikan *fetcher* (seperti API dari bank) atau perombakan masif dari infrastruktur *fetcher* yang ada, *fetcher* dan *building blocks* buatan penulis akan terus digunakan dan semakin meningkat penggunaan serta manfaatnya.

Sayangnya, penulis tidak bisa mencantumkan sampel kode, diagram, ataupun dokumen lain yang sudah dibuat karena merupakan konten perusahaan yang bersifat konfidensial. Traveloka secara umum mempunyai kecenderungan untuk berbagi informasi mengenai teknologi yang diterapkan di Traveloka, kecuali untuk hal yang berkaitan dengan proses bisnis. Karena *fetcher* merupakan salah satu poin kelebihan bisnis kompetitif yang dimiliki oleh Traveloka, teknologi tersebut tidak bisa dibagikan ke pihak luar mana pun.

2.6. Non-Teknis

Selama menjalani kerja praktik, penulis mendapat cukup banyak waktu untuk mempelajari berbagai hal, baik yang berkaitan langsung dengan pekerjaan atau pengetahuan lainnya yang tidak berkaitan. Sebagai contoh, apabila penulis sedang menunggu proses *remote build*, penulis sering meluangkan waktu dengan cara membaca artikel-artikel menarik di Confluence untuk mempelajari proses bisnis dan perencanaan produk masa depan Traveloka.

Traveloka memiliki jam kerja fleksibel untuk hampir seluruh divisi kecuali tim operasional. Penulis bekerja dari sekitar pukul 10 pagi hingga pukul 6 petang. Penulis dibimbing oleh seluruh *software engineer* dari tim *payment* dan mendapat seorang mentor untuk konsultasi dan pengawasan, yaitu David Bayu Ananda. Setiap minggu, penulis bertemu dengan penyelia, Kemal Darmawan, untuk membicarakan perkembangan, kendala, dan rencana penulis mengenai pekerjaan dan pengembangan diri penulis selama menjalani kerja praktik.

Secara umum, lingkungan kerja Traveloka sangat profesional namun tidak hierarkis sehingga siapa pun bisa saling berinteraksi tanpa adanya kecanggungan. Apabila penulis mengalami kendala di luar ranah *tim payment*, penulis bisa langsung menghubungi orang yang relevan tanpa pernah mendapat penolakan.

BAB 3

ANALISIS PEKERJAAN KERJA PRAKTIK

3.1. Kesesuaian dan Perbedaan dengan KAKP

Secara umum, pekerjaan yang dilakukan oleh penulis sesuai dengan apa yang sudah direncanakan di kerangka acuan kerja praktik (KAKP). Ranah pekerjaan dan langkah yang penulis lakukan hampir sama dengan apa sudah tercantum di KAKP.

Perbedaan yang cukup drastis adalah jadwal kerja yang disebabkan oleh dua faktor utama. Pertama, pada saat perancangan KAKP, penyelia belum bisa memberikan gambaran yang cukup mendetil mengenai jadwal kerja penulis. Kedua, penulis baru mengetahui setelah KAKP disetujui bahwa setiap *fetcher* sudah memiliki tenggat waktu peluncuran masing-masing, tidak seperti perkiraan penulis di mana *fetcher* bisa dikerjakan secara paralel. Oleh karena itu, *fetcher* pertama selesai ketika penulis memperkirakan masih melakukan uji coba dan pengembangan *fetcher* secara keseluruhan. Meskipun begitu, tanggal berakhirnya kerja praktik penulis sama dengan apa yang sudah direncanakan.

Selain itu, penulis tidak sempat mencari tahu keterbatasan situs *internet banking* yang dibuat *fetcher*-nya oleh penulis. Beberapa situs *internet banking* seperti Bank BCA hanya mampu menampilkan sejumlah transaksi dalam satu hari, dan transaksi yang berlebih tidak bisa ditampilkan. Untuk mengatasi keterbatasan tersebut, Traveloka memiliki beberapa rekening untuk bank yang sama. Karena angka transaksi di Thailand belum sebanyak transaksi di Indonesia dan mengujinya tidak memungkinkan, maka penulis tidak sempat mencari tahu keterbatasannya.

3.2. Kendala dalam Kerja Praktik dan Cara Penanganannya

Selama pembuatan *fetcher*, penulis sempat menemui beberapa kendala, baik karena kesalahan atau kurangnya pengalaman penulis maupun faktor eksternal yang tidak mungkin penulis atasi sendiri. Namun dari semua kendala tersebut, penulis tidak menemui kendala yang benar-benar menghambat pekerjaan, di mana sebagian besar hanya masalah-masalah kecil yang tidak perlu penanganan khusus. Kendala yang dihadapi oleh penulis justru menambah pengalaman dan meningkatkan kemampuan penyelesaian masalah penulis.

Salah satu hal yang menyulitkan pekerjaan penulis adalah setiap situs *internet banking* memiliki pola dan cara kerja yang sangat berbeda satu sama lain. Hal itu menyebabkan pekerjaan penulis mengharuskan adanya banyak *trial and error*, di mana setiap *fetcher* harus menyesuaikan karakteristik masing-masing situs *internet banking*. Hal tersebut diperparah dengan durasi proses *build* yang cukup lama baik di komputer penulis maupun di *remote build server*. Pada kasus di mana pengujian harus dilakukan di *server staging* sehingga *full build* harus dilakukan, hal ini bisa sangat memakan waktu.

Kendala lainnya adalah *repository* kode yang masih bersifat tunggal menyebabkan pekerjaan setiap orang terhubung satu sama lain, khususnya di *branch develop*. Adanya fitur atau komponen yang baru dibuat atau diubah terkadang memerlukan konfigurasi baru yang harus dikonfigurasi secara manual di *database*.

Selain itu, sering terjadi kasus di mana kode tidak bisa dikompilasi karena adanya kesalahan yang dibuat oleh tim lain. Ada juga kasus langka di mana tim *devops* yang bertugas dalam mengelola prosedur rilis mengubah kode tanpa memberitahukan seluruh pihak yang berkepentingan, sehingga proses rilis tim *payment* sempat terkendala selama beberapa hari. Kedua masalah tersebut bisa diatasi dengan cara mengkomunikasikannya dengan tim lain atau penanggung jawab yang bersangkutan.

3.3. Pembelajaran Soft Skill selama Kerja Praktik

Ada beberapa *soft skill* yang penulis pelajari selama menjalani kerja praktik di Traveloka. Pembelajaran yang paling dirasakan adalah beban tanggung jawab atas

tugas yang diberikan terasa sangat nyata. Setiap karyawan Traveloka, khususnya bagi *product team*, memiliki tanggung jawab individu yang disesuaikan dengan perannya. Setiap karyawan juga mendapat perlakuan yang sama dan diharapkan bisa belajar secara mandiri. Oleh karena itu, tugas yang diemban oleh karyawan magang memiliki bobot dan tanggung jawab yang sederajat dengan karyawan biasa.

Sebagai contoh, penulis pernah dihubungi pada akhir pekan dikarenakan terdapat masalah pada salah satu *fetcher*. Meski akhirnya masalah tersebut bisa diselesaikan hanya dengan mengubah konfigurasi, insiden tersebut menunjukkan bahwa sistem yang dibuat oleh penulis merupakan bagian dari sistem *mission-critical* yang memiliki peran penting bagi operasional Traveloka.

Untuk itu, setiap pekerjaan harus dikerjakan dengan sungguh-sungguh dan sepenuh hati. Pernah ada masalah di *server production* ketika penulis ingin melakukan *deployment* salah satu *fetcher* baru. Karena jadwal rilis harus tetap dipenuhi, penulis harus menunggu sampai larut malam sampai masalah diselesaikan oleh *release engineer*. Apabila pada saat itu peluncuran *fetcher* diundur, akan ada banyak pemegang kepentingan yang terkena dampaknya, khususnya *product manager* yang bertanggung jawab atas *payment* serta tim operasional dan pemasaran di kantor Thailand.

Pembelajaran penting lainnya adalah mau bertanya kepada orang lain yang lebih mengerti tanpa keraguan. Meski bantuan mentor memang sangat signifikan, namun ada kalanya penulis menemukan masalah yang hanya bisa dipecahkan oleh orang lain yang lebih mengerti konteksnya, sering kali dengan hal yang berhubungan dengan *devops*. Traveloka menggunakan Slack sebagai perkakas komunikasi internal untuk menghubungkan seluruh karyawan yang ada di Traveloka dari seluruh jajaran, sehingga siapa pun bisa dihubungi dengan mudah.

Penulis juga mempelajari mengenai pengelolaan waktu. Traveloka memberikan otonomi dan kebebasan waktu bagi karyawannya dalam mengerjakan pekerjaan. Apabila waktu yang diberikan tersebut tidak dimanfaatkan dengan baik, maka pekerjaan bisa tidak selesai dan berpotensi merugikan banyak pihak.

Seluruh sistem dan Confluence di Traveloka menggunakan bahasa Inggris formal. Komunikasi antar tim khususnya dalam bentuk tertulis juga menggunakan bahasa Inggris. Untuk itu, kemampuan berbahasa dan memahami konteks sangat berguna di mana pun, tidak terkecuali di Traveloka.

3.4. Penilaian Individu terhadap Tempat Kerja Praktik

Traveloka merupakan tempat untuk melakukan kerja praktik yang sesuai dengan kriteria dan harapan penulis. Lingkungan kerja di Traveloka sangat kondusif untuk belajar dan berkarya, di mana struktur organisasinya cenderung datar sehingga tidak ada politik dalam kantor. Semua orang harus bertanggung jawab atas pekerjaannya, sehingga semua orang harus merespons ketika mendapat pertanyaan dari tim lain dari ranah yang berbeda. Penulis pernah memanfaatkan kesempatan ini untuk bertanya banyak kepada *product manager* di tim *payment* terkait dengan seluk beluk sistem pembayaran yang ada di Traveloka. Budaya kerja Traveloka tidak pernah memandang individu sebelah mata, namun lebih melihat hasil (*result-oriented*) dan menghargai keberhasilan (*merit-based*).

Akan tetapi, Traveloka juga memiliki beberapa ketidak sempurnaan bagi penulis. Penulis merasa Traveloka sebenarnya sudah terhitung terlalu mapan, sehingga penulis merasa sebagai individu hanya merupakan satu bagian kecil dari roda bisnis yang sangat besar. Meski tim penulis sudah menyatakan bahwa kontribusi yang diberikan penulis memiliki dampak yang besar dan berjangka panjang, namun penulis tetap merasa bahwa penulis belum menambah nilai yang nyata bagi perusahaan.

3.5. Relevansi dengan Perkuliahan di Fasilkom UI

Pekerjaan penulis memiliki beberapa kaitan dengan perkuliahan program studi penulis, yaitu ilmu komputer. Berikut merupakan penjabaran spesifik mengenai mata kuliah yang memiliki relevansi:

1. Dasar-Dasar Pemrograman dan Struktur Data & Algoritma

DDP dan SDA memberikan pengetahuan dasar mengenai pemrograman yang diperlukan oleh *software engineer* dalam konteks apapun. Untungnya, bahasa yang digunakan oleh Traveloka dan kedua perkuliahan tersebut adalah sama yaitu Java, sehingga penulis tidak merasakan kesulitan dalam beradaptasi.

2. Perancangan & Pemrograman Web dan Jaringan Komputer

Seperti yang sudah penulis jabarkan sebelumnya, *fetcher* banyak menyinggung konsep yang berkaitan dengan internet. Tidak hanya itu, penulis juga sempat memperbaiki halaman pengaturan *fetcher* untuk tim operasional serta memperbaiki kode JavaScript yang bermasalah dan menyebabkan *fetcher* tidak berjalan dengan semestinya. *Fetcher* juga memerlukan penyimpanan data di *database*, namun ilmu mengenai *database* yang penulis pelajari selama perkuliahan tidak langsung diaplikasikan karena Traveloka menggunakan *database* NoSQL dan penulis cukup menggunakan *accessor* yang sudah ada serta siap digunakan.

3. Proyek Perangkat Lunak

Di mata kuliah PPL *mentorship* semester genap 2015/2016, terdapat materi dan praktek mengenai testing. Hal tersebut diaplikasikan penulis ketika mengembangkan *fetcher* dengan cara membuat *automated unit testing* agar uji coba dan pengecekan hasil *fetch* bisa berjalan secara otomatis.

4. Pemrograman Sistem

Selama membuat, menguji coba, dan melakukan *deployment fetcher*, penulis sangat mengandalkan UNIX *shell* untuk memenuhi berbagai kebutuhan. Mata kuliah pemrograman sistem pada dasarnya memperkenalkan konsep dan pemakaian *terminal* ke mahasiswa. Penulis juga sering melakukan koneksi SSH ke *server* untuk melakukan beberapa pekerjaan administratif seperti melihat *error log* dan mengambil *database production* untuk digunakan di *server development* atau *staging*.

BAB 4

PENUTUP

4.1. Kesimpulan

Penulis melakukan kerja praktik selama tiga bulan di Traveloka dari 13 Juni hingga 12 September 2016. Selama waktu tersebut, penulis ditempatkan di *product team* bagian *payment mission* untuk mengembangkan beberapa *fetcher* untuk bank di Thailand dalam rangka meningkatkan efisiensi operasional transaksi pembayaran melalui transfer.

Metodologi yang digunakan selama penulis melakukan kerja praktik menerapkan prinsip *agile* tanpa menggunakan kerangka kerja tertentu. Dalam rangka mendukung pengembangan *fetcher*, penulis mempelajari lebih dalam beberapa konsep seperti *asynchronous programming*, *distributed network architecture*, *remote procedure call*, *hypertext transfer protocol*, dan *software development life cycle*. Penulis memanfaatkan beberapa teknologi yang digunakan di internal Traveloka seperti bahasa pemrograman Java, MongoDB, Git, GoCD, Jenkins, IntelliJ IDEA, Phabricator, dan Ansible.

Penulis berhasil menyelesaikan lima *fetcher* untuk lima bank berbeda di Thailand dengan tepat waktu. Tidak hanya itu, penulis juga membuat dokumentasi atas pekerjaan penulis serta *building blocks* untuk pengembangan *fetcher* ke depannya. Secara umum, penulis merasa puas telah melakukan kerja praktik di Traveloka karena budaya perusahaan, bobot pekerjaan, dan ilmu yang didapat cukup sesuai dengan ekspektasi penulis. Penulis mendapat pengalaman yang tidak bisa didapat di perkuliahan, dan segala hasil pekerjaan penulis mendapat apresiasi karena akan bermanfaat bagi proses bisnis Traveloka dalam jangka panjang.

Tidak hanya pengetahuan teknis, penulis juga mempelajari berbagai kemampuan non-teknis khususnya mengenai rasa tanggung jawab karena penulis mendapat tanggung

jawab yang sederajat dengan karyawan biasa. Kesempatan tersebut belum tentu bisa didapat oleh penulis apabila melakukan kerja praktik di tempat lain.

4.2. Saran

Penulis memiliki beberapa saran untuk tempat kerja praktik. Seperti yang sudah dijabarkan sebelumnya, salah satu poin yang penulis kurang sukai dari Traveloka adalah ukurannya yang terlalu besar. Hal ini menyebabkan penulis sempat merasa kehilangan arah pada awal pekerjaan karena tidak ada pelatihan atau sesi khusus bagi karyawan baru. Dampak yang sangat terasa adalah penulis tidak memiliki pengetahuan garis besar mengenai Traveloka, seperti tim-tim yang ada di perusahaan dan bagaimana cara kerja sistem-sistem yang sudah berjalan. Meskipun pada akhirnya penulis bisa mempelajari hal tersebut secara otodidak, alangkah baiknya apabila ada penjelasan mengenai hal tersebut untuk memperlancar karyawan baru dalam beradaptasi dengan lingkungan kerja Traveloka.

Penulis ingin memberi beberapa saran kepada mahasiswa yang akan melakukan kerja praktik ke depannya. Carilah pekerjaan berdasarkan apa yang ingin dilakukan, bukan berdasarkan masukan orang lain apa yang harus dilakukan. Untuk memudahkan proses pencarian tempat kerja praktik, daftarlal sedini mungkin, karena jika mendekati liburan persaingan akan semakin sulit karena kuota yang tersedia semakin berkurang.

Untuk dosen kerja praktik dan fakultas, alangkah baiknya apabila persetujuan KAKP dilakukan lebih awal. Format *log* kerja praktik juga seharusnya diberikan lebih awal agar peserta memiliki gambaran seperti apa jurnal yang harus dibuat. Selain itu, presentasi kerja praktik seharusnya dipublikasikan dengan lebih baik sehingga mahasiswa yang belum melakukan kerja praktik bisa ikut datang ke presentasi untuk mengetahui perusahaan dan pekerjaan yang cocok bagi mereka. Apabila memungkinkan, seharusnya orientasi peserta presentasi kerja praktik seharusnya tidak hanya mahasiswa yang sedang mengambil mata kuliah kerja praktik, melainkan bagi mereka yang belum pernah melakukan kerja praktik sama sekali.

DAFTAR REFERENSI

[1] *Confluence - Team Collaboration Software | Atlassian*. (2016). Diakses 4 Oktober 2016, dari <https://www.atlassian.com/software/confluence>.

[2] *What is systems development life cycle (SDLC)? - Definition from WhatIs.com*. (2016). Diakses 4 Oktober 2016, dari <http://searchsoftwarequality.techtarget.com/definition/systems-development-life-cycle>.

LAMPIRAN 1

KERANGKA ACUAN KERJA PRAKTIK

[Selipkan KAKP pada halaman-halaman setelah halaman ini.

**Gunakan PDFMerge atau lainnya untuk melakukan hal
tersebut/*rearrange* halaman-halaman pada PDF]**

LAMPIRAN 2
LOG KERJA PRAKTIK

[Selipkan LOG KP pada halaman-halaman setelah halaman ini.

**Gunakan PDFMerge atau lainnya untuk melakukan hal
tersebut/*rearrange* halaman-halaman pada PDF]**