



UNIVERSITAS INDONESIA

**Pengembangan *Application Programming Interface* (API) untuk
Aplikasi Kudo Companion (Field Officer App)**

LAPORAN KERJA PRAKTIK

ADRIANUS SAGA EKAKRISTI

1306409476

FAKULTAS ILMU KOMPUTER

PROGRAM STUDI ILMU KOMPUTER

DEPOK

NOVEMBER 2016

HALAMAN PERSETUJUAN DOSEN MATA KULIAH KERJA PRAKTIK

Laporan ini diajukan oleh :

Nama : Adrianus Saga Ekakristi

NPM : 1306409476

Program Studi : Ilmu Komputer

Judul Kerja Praktik : Pengembangan *Application Programming Interface (API)* untuk Aplikasi *Kudo Companion (Field Officer App)*

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.Sc.)

Ditetapkan di : Depok

Tanggal : 24 November 2016

ABSTRAK

Nama : Adrianus Saga Ekakristi
Program Studi : Ilmu Komputer
Judul : Pengembangan *Application Programming Interface* (API)
untuk Aplikasi Kudo Companion (Field Officer App)

Laporan ini berisi hasil kerja praktik yang telah dilakukan penulis antara Juni hingga Agustus 2016. Laporan kerja praktik ini disusun sebagai pemenuhan mata kuliah Kerja Praktik pada semester Gasal 2016/2017. Penulis melaksanakan kerja praktik pada perusahaan PT. Kudo Teknologi Indonesia sebagai *back-end API developer intern*. Penulis membangun API yang digunakan dalam proyek Kudo Companion atau Field Officer App. API ini akan dikonsumsi oleh aplikasi *front-end* Field Officer App, yaitu *mobile app* dan *web app*. Field Officer App itu sendiri akan digunakan oleh *field officer* Kudo untuk memantau agen Kudo. Penulis belajar banyak hal selama melaksanakan kerja praktik di Kudo, baik itu hal teknis maupun non-teknis.

Kata kunci: Kudo, *back-end*, *application programming interface*

DAFTAR ISI

HALAMAN PERSETUJUAN DOSEN MATA KULIAH KERJA PRAKTIK	i
ABSTRAK	ii
DAFTAR ISI.....	iii
DAFTAR GAMBAR	v
DAFTAR LAMPIRAN.....	vi
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 ISI.....	6
2.1. Pekerjaan dalam Kerja Praktik	6
2.1.1. Latar Belakang Pekerjaan	6
2.1.2. Tinjauan Pustaka.....	7
2.1.3. Metodologi dan Timeline.....	7
2.1.4. Teknologi yang Digunakan.....	8
2.1.5. Pelaksanaan dan Hasil Kerja Praktik	10
2.2. Analisis	11
2.2.1. Kesesuaian dan Perbedaan dengan KAKP	11
2.2.2. Kendala dalam Pelaksanaan Kerja Praktik	13
2.2.3. Relevansi dengan Perkuliahan di Fasilkom UI.....	13
BAB 3 PENUTUP.....	16

3.1. Kesimpulan	16
3.2. Saran	16
DAFTAR REFERENSI	17

DAFTAR GAMBAR

Gambar 1 Struktur.....	5
------------------------	---

DAFTAR LAMPIRAN

LAMPIRAN 1 KERANGKA ACUAN KERJA PRAKTIK	viii
LAMPIRAN 2 LOG KERJA PRAKTIK.....	ix

BAB 1

PENDAHULUAN

1.1. Proses Pencarian Kerja Praktik

Pada awal Semester Genap 2015/2016, penulis mulai mencari tempat magang untuk memenuhi mata kuliah Kerja Praktik. Sebagai persiapan, penulis meminta transkrip nilai penulis yang terbaru melalui Sistem Informasi Surat Fakultas Ilmu Komputer (Fasilkom) Universitas Indonesia (UI). Transkrip nilai digunakan untuk beberapa perusahaan yang meminta dalam proses pendaftaran. Transkrip nilai tersebut dapat penulis ambil dalam waktu 2 hari di Sekretariat Fasilkom UI.

Perusahaan pertama dimana penulis mendaftar adalah PT. Global Digital Prima (GDP) Venture divisi teknologi yang lebih sering dikenal sebagai GDP Labs. Penulis mengenal GDP Labs dari beberapa seminar yang dibawakan oleh Bapak On Lee, *Chief Tehnology Officer* dari perusahaan ini. Untuk pendaftaran *internship* di GDP Labs, penulis perlu membuat *cover letter* berisi alasan, latar belakang, dan motivasi untuk magang di perusahaan tersebut. Langkah pertama penulis mendaftar di GDP Labs adalah mengirim *email* kepada Bapak On Lee pada 22 Februari 2016 dan melampirkan *cover letter* dan *curriculum vitae* (CV) pada *email* tersebut. Beberapa jam kemudian Bapak On Lee membalas *email* penulis dan memberitahu bahwa HR dari GDP Labs akan menghubungi penulis. Penulis kemudian mendapatkan *email* dari Ibu Stella Kurniawan bahwa Bapak Setiawan akan menghubungi penulis untuk *online coding test*. Hingga sebulan kemudian, penulis tidak dihubungi kembali. Ketika penulis mencoba menghubungi, Bapak Setiawan mengatakan bahwa *platform* yang digunakan untuk *online coding test* sedang mengalami migrasi dan *testing* sehingga belum siap. Pada tanggal 23 Maret 2016, penulis mendapatkan *email* berisi *link* menuju tempat *test*, yaitu olympia.id dan menjalani *test* tersebut. Pada tanggal 31 Maret 2016, penulis mendapatkan kabar berupa *email* bahwa pendaftaran *internship* penulis di GDP Labs tidak diterima.

Penulis juga mendaftar program *internship* pada perusahaan Bukalapak pada awal April 2016. Pada tanggal 14 April 2016, salah satu pegawai divisi *Recruitment Specialist* Bukalapak mengirimkan penulis *email* beserta dengan soal test yang harus penulis kerjakan dan kumpulkan 2 hari kemudian. Pada tanggal 3 Mei 2016, Bukalapak mengirimkan penulis *email* bahwa pendaftaran *internship* penulis di Bukalapak tidak diterima.

Selain itu, penulis juga mendaftar pada perusahaan Blibli dengan mengirimkan *email* beserta CV kepada salah satu *email recruitment* IT Blibli. Namun, Blibli tidak membalas pendaftaran tersebut.

Pada pertengahan April 2016, penulis mendaftar program magang pada perusahaan YesBoss dengan mengirimkan *email* berisi CV dan *application letter* sesuai proses pendaftaran yang tertera pada situs resmi YesBoss. YesBoss baru membalas *email* penulis pada awal Juni 2016 dan memberitahukan bahwa penulis tidak diterima.

Pada akhir April 2016, penulis mendaftar program magang pada perusahaan Dekoruma. Penulis mengetahui kesempatan magang tersebut dari salah satu alumni Fasilkom UI. Penulis mendaftar dengan cara mengirimkan *email* beserta CV kepada salah satu developer Dekoruma. Kemudian, penulis dipanggil untuk *interview* dan *test* di kantor Dekoruma dua hari kemudian. Namun, penulis tidak diterima pada perusahaan tersebut.

Pada akhir April 2016, penulis mendaftar program *internship* pada perusahaan Kios Untuk Dagang Online (Kudo) yang dinamakan Kudo Class Academy 2016. Penulis mendapatkan informasi program *internship* ini dari berbagai sumber, seperti LinkedIn, poster yang tersebar di media komunikasi, dan teman-teman Fasilkom UI yang juga sedang mencari tempat kerja praktik. Penulis mendaftar dengan cara mengirimkan *email* kepada *email* resmi *Human Capital* (HC) Kudo beserta CV. Penulis mendaftar pada divisi *back-end application programming interface* (API). Kurang lebih seminggu kemudian, penulis menerima *email* berisi undangan wawancara pada 4 Mei 2016 di kantor Kudo yang terletak di Gedung Grand Lucky, Radio Dalam, Jakarta

Selatan. Penulis menghadiri wawancara tersebut dan ditanyakan beberapa hal terkait teknis maupun non-teknis. Wawancara non-teknis dilakukan bersama dengan seorang dari divisi HC. Wawancara teknis dilakukan bersama dengan seorang dari divisi dimana saya mendaftar, yaitu divisi *back-end* API. Sekitar seminggu kemudian, penulis dikabari oleh salah seorang dari divisi HC Kudo melalui telepon bahwa penulis diterima pada program *internship* yang dimulai pada 13 Juni 2016. Penulis memilih Kudo sebagai tempat kerja praktik dan tidak meneruskan pencarian.

1.2. Tempat Kerja Praktik

Bagian ini akan membahas mengenai profil tempat kerja praktik serta posisi penulis dan divisi dimana penulis magang dalam struktur organisasi perusahaan.

1.2.1. Profil Tempat Kerja Praktik

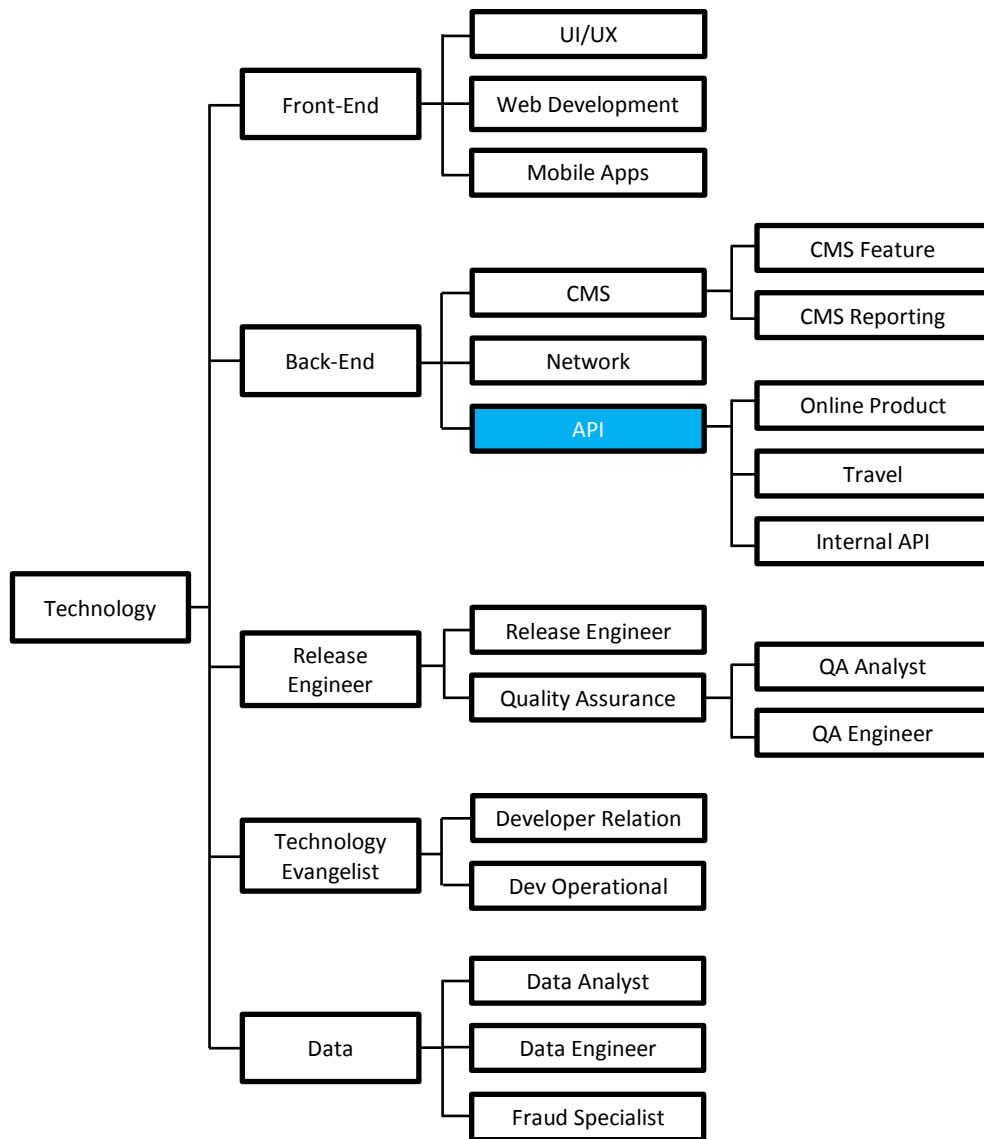
Kios Untuk Dagang Online atau Kudo adalah sebuah perusahaan yang menyediakan aplikasi dalam berbagai platform yang menghubungkan pedagang, baik itu *seller* ataupun *reseller* produk, untuk memasarkan usaha mereka kepada masyarakat melalui agen-agen Kudo. Konsep *marketplace* pada Kudo adalah *online-to-offline*, yaitu memasarkan produk *online* yang dapat berasal dari *marketplace online* seperti Bukalapak, Elevenia, Lazada, dan sebagainya, kepada masyarakat yang tidak memiliki akses *online*. Oleh karena itu, Kudo memiliki agen-agen Kudo, yang mana masyarakat dapat membeli produk yang tersedia secara *online* dan melakukan pembayaran kepada agen tersebut. Target pasar utama dari Kudo adalah masyarakat yang tidak memiliki akses terhadap platform belanja *online*, seperti masyarakat menengah kebawah di kota kecil. Kudo memiliki visi untuk menciptakan pengusaha-pengusaha digital untuk memajukan Indonesia. Seluruh masyarakat Indonesia memiliki kesempatan menjadi agen Kudo.

Kudo didirikan pada tahun 2014 oleh Albert Lucius sebagai *Chief Executive Officer* (CEO) serta merangkap sebagai *Chief Technology Officer* (CTO), dan Agung Nugroho sebagai *Chief Operating Officer* (COO) [1]. Melihat pada sejarah berkembangnya Kudo, produk-produk Kudo mengalami evolusi yang signifikan untuk mencapai visi

perusahaan. Produk pertama Kudo diluncurkan pada Desember 2014, yaitu Kudobox 1, yaitu *tablet* dalam *box* yang diletakkan pada tempat publik. Perangkat tersebut dapat digunakan pengunjung untuk berbelanja *online* [1]. Pada Januari 2015, Kudo meluncurkan Kudobox 2 yang berbentuk *tablet* yang dapat dibawa oleh agen Kudo perorangan [1]. Pada Maret 2016, Kudo meluncurkan aplikasi *mobile app* untuk mempermudah agen Kudo dan meningkatkan jumlah pelanggan [1]. Pada Mei 2016, Kudo sudah memiliki *web app* sehingga semua orang dapat menjadi agen Kudo tanpa aplikasi *mobile* Kudo [1].

1.2.2. Posisi Penempatan Pelaksana Kerja Praktik dalam Struktur Organisasi

Penulis memiliki posisi sebagai *back-end API developer intern* yang berada dibawah divisi API yang merupakan bagian dari divisi besar *back-end*. Divisi *back-end* adalah salah satu divisi dari bidang *technology* Kudo yang dikepalai oleh CTO. Struktur organisasi bidang *technology* Kudo dapat dilihat pada Gambar 1.



Gambar 1. Struktur Organisasi Divisi Technology Kudo dan Posisi Penulis saat Kerja Praktik

BAB 2

ISI

2.1. Pekerjaan dalam Kerja Praktik

Bagian ini akan membahas latar belakang, tinjauan pustaka untuk teknologi yang digunakan, dan hasil pekerjaan penulis pada kerja praktik.

2.1.1. Latar Belakang Pekerjaan

Kudo menciptakan produk yang siap melayani *user* dalam jumlah yang besar dengan aman dan stabil. Untuk memenuhi hal tersebut, sistem aplikasi *web* Kudo harus memiliki arsitektur yang *reliable* dan *scalable*. Salah satu contoh arsitektur sistem yang baik adalah arsitektur yang modular, seperti pemisahan sisi *back-end* yang bertanggung jawab kepada logika aplikasi serta pengolahan data, dan *front-end* yang fokus kepada tampilan aplikasi kepada *user*. Divisi *back-end* Kudo bertanggung jawab pada *layer back-end* pada sistem Kudo, yaitu *network* pada *layer* paling dasar, *content management system* (CMS) untuk mengolah dan memanipulasi data, dan API yang bertanggung jawab menyediakan *interface* untuk melayani *request* informasi ataupun data yang dibutuhkan berbagai aplikasi di Kudo.

Field officer (FO) di Kudo memiliki tanggung jawab untuk mengawasi kinerja agen serta membantu kendala teknis maupun non-teknis yang dihadapi agen dalam berjualan dengan aplikasi Kudo. Aplikasi untuk membantu pekerjaan FO sudah pernah dibuat sebelumnya, namun tidak digunakan kembali dikarenakan pergeseran *platform* aplikasi Kudo menjadi *mobile app*. Kudo sebagai sebuah perusahaan juga perlu memantau kinerja FO di lapangan. Perlunya aplikasi untuk membantu dan memantau FO inilah yang menjadi alasan dibuatnya aplikasi Kudo Companion, atau dikenal juga sebagai Kudo Field Officer App (FO App). Pada *internship* kali ini, penulis mendapatkan kesempatan untuk mengimplementasi API untuk aplikasi FO App.

2.1.2. Tinjauan Pustaka

Bagian ini akan membahas beberapa konsep penting terkait teknis pekerjaan penulis pada masa kerja praktik.

1. *Application Programming Interface (API)*

Application programming interface, atau *application program interface*, atau API adalah sekumpulan *code* yang memungkinkan dua buah aplikasi perangkat lunak untuk berkomunikasi dengan satu sama lain [2]. API digunakan dua buah sistem yang berbeda untuk bertukar, meminta, atau mengirim data. Dalam konteks *web development*, API yang disediakan melalui *network* disebut *web API*. Umumnya, implementasi *web API* memiliki bentuk *request-response* dalam *layer* HTTP. Beberapa tipe *web API* adalah *simple object access protocol* (SOAP) dan *representational state transfer* (REST).

2. *Representational State Transfer (REST) API*

Representational state transfer atau REST API adalah sebuah *architectural style* untuk mendefinisikan *web API*. REST API memiliki ide dasar mengakses *resource* dengan menjelajahi *link*. REST API mendefinisikan standar cara berkomunikasi dua buah sistem aplikasi perangkat lunak melalui *request* dan *response* HTTP. Beberapa karakteristik yang dimiliki REST API adalah terjadi antara *client-server*, *stateless*, *uniform interface* atau seluruh *resource* dapat diakses dengan *interface* yang *generic*, dan dapat terdiri dari beberapa *component* dalam setiap *layer*.

2.1.3. Metodologi dan Timeline

Developer Kudo menggunakan *software development life cycle* (SDLC) bertipe *scrum*. Umumnya, tim API, yaitu penulis, akan membuat *design endpoint* beserta detail *format request* dan *response* setelah *mockup* selesai dikerjakan oleh tim divisi UI/UX. Setelah tim API membuka API pada *test environment*, tim dari divisi lain, seperti *mobile app* dan *web development*, dapat mulai menggunakan API untuk aplikasi yang sedang mereka kerjakan masing-masing. Namun tidak jarang terjadi perubahan terhadap *mockup* oleh divisi UI/UX setelah berdiskusi dengan berbagai pihak, termasuk *user*.

Oleh karena itu, perubahan terhadap *design* dan implementasi API juga dapat terjadi di saat berjalannya *sprint*. Dalam perubahan tersebut, tim API juga perlu mempertimbangkan berbagai faktor seperti *feasibility* dan dampak terhadap performa aplikasi.

Pada pengerjaan proyek ini, terdapat *sprint-sprint* untuk target pengerjaan *feature* API untuk FO App. *Feature* yang dikerjakan pada *sprint* pertama dari 27 Juni hingga 3 Juli adalah *authentication* atau *login*. *Feature* yang dikerjakan pada *sprint* kedua dari 4 Juli hingga 17 Juli adalah absensi kunjungan FO terhadap agen Kudo. *Feature* yang dikerjakan pada *sprint* ketiga antara 18 Juli hingga 31 Juli adalah *list* agen, *detail* profil agen, dan *update* profil agen. Setelah *sprint* ketiga, akan diadakan *mid presentation* untuk proyek FO App yang akan dihadiri CEO serta *lead* dari seluruh divisi *technology* Kudo. Pada presentasi ini, akan dilaporkan seluruh *progress* pengerjaan proyek FO App, termasuk dari sisi API. Pada *sprint* keempat antara 1 Agustus hingga 7 Agustus, *Feature* yang dikerjakan adalah *forgot password* dan *reset password* akun FO. Setelah *sprint* keempat, akan diadakan *final presentation* proyek FO App yang dihadiri CEO, *lead* seluruh divisi *technology*, serta divisi *sales*. Waktu selanjutnya hingga masa kerja praktik selesai digunakan untuk *refactoring*, *bug fixing*, serta mempersiapkan *deployment* ke *environment staging* dan *production*.

2.1.4. Teknologi yang Digunakan

Dalam pekerjaan penulis dalam divisi *back-end* API, terdapat beberapa *tools* dan *framework* yang digunakan dalam mengerjakan proyek FO App. Beberapa teknologi yang digunakan penulis dalam masa kerja praktik adalah bahasa pemrograman PHP, Laravel, MariaDB, Postman.

PHP adalah bahasa pemrograman untuk *web* yang relatif mudah dipahami *programmer* pemula, bahkan masyarakat umum yang belum punya pengalaman pemrograman. Karakteristik mudah dipahami inilah yang membuat bahasa pemrograman PHP masih berkembang dan digunakan masyarakat luas. *Developer* Kudo menggunakan bahasa pemrograman PHP, terutama *framework* Laravel dan Lumen.

Laravel merupakan salah satu *framework* untuk membangun aplikasi *web* dalam bahasa pemrograman PHP. Laravel memiliki *architectural pattern* *model-view-controller* (MVC). Untuk mengelola data, Laravel memiliki *class model* yang merepresentasikan *table* dalam *database*. Hal ini merupakan realisasi konsep *object-relational mapping* (ORM) dalam Laravel. Dengan ORM, *developer* dapat lebih mudah memanipulasi data dalam bentuk *class* dan *object*, bukan dengan menuliskan SQL. SQL akan disusun oleh *model* Laravel dan mengeksekusinya pada *database*. *Model* pada Laravel juga dapat digunakan untuk mendefinisikan *relationship* antar *table* yang nantinya dapat mempermudah berbagai macam proses *join table*. Selain itu, Laravel juga menggunakan *migration* untuk mengelola *versioning* dari *blueprint* struktur *database*. Setiap *migration* pada Laravel berisi suatu perubahan struktur *database*, baik itu menambah *table*, mengubah *field* dari *table*, atau menghapus *table*. Secara garis besar, proses pengolahan *request* menjadi *response* pada Laravel dimulai dari *routing*. *Routing* akan mendefinisikan fungsi pada *controller* yang akan mendapatkan *request* dan menyusun bentuk *response*, baik itu data dalam format JSON ataupun hasil *render* sebuah *view*. *Controller* pada Laravel adalah bagian berisi *function* yang berisi logika pengolahan *request* dan penyusunan *response*. *Controller* juga dapat mengakses data dengan menggunakan *model*. Laravel menyediakan *templating* sebagai bagian *view* dalam MVC. *Templating* pada Laravel menggunakan Blade, yaitu salah satu cara mendefinisikan struktur *layout* dan bagaimana suatu data ditampilkan dalam *view*. Dalam pengerjaan API oleh penulis, *templating* Blade ini tidak digunakan karena API FO App mengembalikan data dalam format JSON. Dependensi *module* ataupun *library* pada suatu *project* Laravel dikelola oleh Composer. API FO App membutuhkan dependensi Guzzle untuk melakukan *request* HTTP terhadap API Kudo lain.

Database yang digunakan dalam proyek Kudo ini adalah MariaDB. MariaDB memiliki kemiripan yang tinggi dengan MySQL. MariaDB bersifat *open-source*. Secara garis besar, MariaDB adalah salah satu jenis *relational database management system* (RDBMS). MariaDB digunakan untuk menyimpan data dalam bentuk *table* yang dapat

memiliki *relationship*, *key*, dan properti lainnya. Penulis tidak banyak melakukan optimasi, seperti *indexing*, pada *database* Kudo.

Postman adalah suatu aplikasi yang dapat mengirim *request* HTTP dalam berbagai bentuk. Postman memudahkan penggunaanya untuk mengirim *request* HTTP dengan berbagai tipe seperti POST, GET, PUT, DELETE, dan sebagainya. Penulis menggunakan aplikasi Postman dalam *browser* Google Chrome untuk *development* dan uji coba API yang sudah dibuat, baik itu dalam *environment local*, *test server*, maupun *production server*. Postman relatif lebih mudah digunakan daripada metode lain, seperti curl pada Linux.

Sebagai tambahan, untuk mengevaluasi jalannya aplikasi *mobile app* Kudo, baik itu *mobile app* FO App maupun *mobile app* produk Kudo, penulis menggunakan Genymotion. Genymotion adalah salah satu *emulator* Android yang relatif mudah digunakan. Penulis mencoba aplikasi *mobile app* untuk mempelajari dan mengevaluasi kekurangan atau hal yang dapat diperbaiki dari API yang digunakan oleh *mobile app*. Hal ini terkait dengan performa yang diberikan API dan beban yang diterima *server*.

Terdapat beberapa *tools* pendukung lain yang digunakan penulis dalam masa kerja praktik, seperti JIRA dan PHPMyAdmin.

2.1.5. Pelaksanaan dan Hasil Kerja Praktik

Dalam *internship* ini, penulis membangun API untuk aplikasi FO App yang terdiri dari beberapa *endpoint*. API ini dibangun dari awal. Secara garis besar, penulis membangun sebuah *web service*. API ini dibangun dengan *framework* Laravel. *Versioning* pada *project* ini menggunakan BitBucket, yaitu *hosting service* untuk *version control*, yang salah satunya adalah Git. Penulis menggunakan *environment local* dan *environment test server* untuk *development*. *Server* untuk API FO App diurus oleh penyelia penulis dan *head* dari divisi *back-end*, sehingga penulis sering berkoordinasi dengan kedua pihak tersebut.

Terdapat beberapa *feature* yang dikerjakan. *Feature* pertama adalah *authentication* yang terdiri dari *login*, *forgot password*, dan *reset password*. *Feature* kedua adalah absensi kunjungan FO terhadap agen. *Feature* ini terdiri dari *request* QR Code oleh aplikasi agen Kudo, absensi dengan akses internet, dan absensi tanpa akses internet. *Feature* ketiga adalah penampilan data agen. *Feature* ini terdiri dari *list* agen, *detail* data sebuah agen, statistik agen, *update* data agen, *search* terhadap agen berdasarkan *keyword*, dan *endpoint* pendukung lainnya. *Feature* terakhir adalah beberapa *feature* kecil pendukung aplikasi *mobile*, seperti informasi valid atau tidaknya *session*, informasi akun yang sedang *login*, *accomplishment* target kunjungan oleh akun yang sedang *login*, dan sebagainya.

API yang penulis kerjakan perlu mengakses data dengan ukuran yang sangat besar. Penulis perlu mengimplementasi algoritma se-efisien mungkin agar tercapai skalabilitas untuk jangka panjang.

Untuk beberapa *feature*, seperti *feature forgot* dan *reset password* akun FO, API yang penulis kerjakan perlu menggunakan *mail service*. Oleh karena itu, penulis juga berkoordinasi dengan divisi *network* yang menyediakan dan mengkoordinasikan *server* serta *domain* yang digunakan untuk *mail*.

2.2. Analisis

Bagian ini akan membahas mengenai kesesuaian pelaksanaan kerja praktik dengan KAKP, kendala atau hambatan dalam kerja praktik, serta relevansi dengan perkuliahan di Fasilkom UI.

2.2.1. Kesesuaian dan Perbedaan dengan KAKP

Terdapat ketidaksesuaian pelaksanaan kerja praktik dengan KAKP, yaitu perbedaan *timeline* yang direncanakan dengan pengerjaan sesungguhnya. Pada *timeline* di KAKP, minggu kedua digunakan penuh untuk tahap *planning* untuk rancangan dan spesifikasi setiap *endpoint* API yang dibutuhkan untuk proyek FO App. Namun pada pelaksanaan sesungguhnya, penulis menyelesaikan lebih rancangan *endpoint* lebih dulu, selesai mengimplementasi *feature authentication*, termasuk *forgot* dan *reset password*, dan

mulai mengimplementasi *feature* data agen. Minggu ketiga, yang pada KAKP direncanakan untuk implementasi *feature authentication*, pada pelaksanaan sesungguhnya digunakan untuk menyelesaikan sebagian besar implementasi *feature* data agen dan absensi. Pada masa ini, sebagian besar *feature* sudah terimplementasi, sehingga minggu-minggu selanjutnya lebih banyak digunakan untuk berbagai revisi terkait penambahan validasi parameter *request*, *bug fixing*, memperbaiki performa *query* terhadap *database*, dan mulai menyusun dokumentasi. Revisi juga banyak dilakukan terkait integrasi dengan API Kudo lainnya. Tujuan dari integrasi ini adalah memperbaiki performa sehingga beban terhadap *database server* lebih rendah.

Perbedaan lain antara *timeline* KAKP dengan pelaksanaan sesungguhnya adalah pergeseran *timeline* untuk *feature forgot password* menjadi paling akhir. Perpindahan ini dilakukan ditengah masa kerja praktik berlangsung. Hal ini merupakan keputusan *lead* untuk proyek *intern* dengan mempertimbangkan kesulitan serta prioritas pentingnya *feature* lain. Perubahan ini relatif tidak mengganggu penulis pada pelaksanaan kerja praktik.

Pengerjaan *feature* absensi juga mengalami banyak perubahan ditengah *sprint* berlangsung. Hal ini dikarenakan *feature* absensi adalah *feature* yang memiliki prioritas tertinggi dalam proyek FO App ini. Penulis dan tim *intern* divisi lain yang berada pada proyek FO App ini diharapkan mengusulkan suatu prosedur untuk absensi kunjungan FO kepada agen. Prosedur ini mendapat banyak usulan dari berbagai pihak agar lebih baik lagi, terutama setelah *mid presentation*. Masukkan tersebut antara lain seperti kasus ketika *user* atau FO berada pada daerah dengan koneksi yang buruk, bagaimana sistem FO App sebelumnya mengatasi masalah kecurangan, dan sebagainya.

Penulis juga melakukan *refactoring*, *bug fixing*, dan perubahan *code* untuk *security* yang lebih baik di waktu yang tersebar pada setiap *sprint*. Aktivitas tersebut direncanakan pada *timeline* di KAKP pada minggu-minggu terakhir setelah *final presentation*. Penulis merasa perlu memprioritaskan hal ini agar *endpoint* yang sudah

dibuka oleh API dapat diuji oleh tim divisi *quality assurance* atau digunakan dengan baik oleh tim *front-end*.

2.2.2. Kendala dalam Pelaksanaan Kerja Praktik

Terdapat beberapa kendala yang dihadapi penulis dalam menjalani kerja praktik.

Salah satu kendala adalah ukuran data yang digunakan oleh aplikasi di Kudo sangatlah besar. Hal ini menyebabkan penulis perlu berhati-hati mengimplementasi *feature* yang memerlukan data dari *database*, terutama data yang merupakan gabungan dari lebih dari satu *table*. Penyusunan *query* yang tidak tepat, seperti *join table* yang berlebihan dapat menyebabkan *service API* atau *database* yang berhenti bekerja. Dari kendala ini, penulis belajar bahwa implementasi aplikasi dalam skala *enterprise* memerlukan ketelitian dan perhatian pada berbagai aspek, seperti *scalability*.

Penulis juga perlu mempelajari beberapa *protocol* atau *library* yang digunakan *developer API* Kudo, seperti OAuth, Guzzle, dan sebagainya. Mengingat dalam implementasi API itu sendiri dapat muncul berbagai bug atau ketidaksesuaian, penulis merasa perlu mempelajari komponen-komponen tersebut dengan cepat agar memiliki waktu yang cukup untuk hal-hal yang tidak terduga.

Kendala lain yang dihadapi penulis adalah penulis membutuhkan waktu untuk mempelajari karakteristik implementasi aplikasi yang digunakan oleh *developer* perusahaan. Hal ini mencakup hal teknis yang detail seperti *format response endpoint*, penggunaan *helper class*, struktur *project*, penggunaan *model*, struktur *conditional*, penggunaan *migration*, dokumentasi, dan sebagainya. Proyek FO App yang dikerjakan *intern* pada akhirnya akan diwariskan kepada *developer* Kudo. Penulis merasa perlu untuk memastikan bahwa proyek tersebut dapat dengan mudah diadopsi dan dikembangkan lebih lanjut oleh *developer* Kudo.

2.2.3. Relevansi dengan Perkuliahan di Fasilkom UI

Terdapat beberapa mata kuliah di Fasilkom UI yang terkait dan membantu penulis dalam melaksanakan kerja praktik. Mata kuliah tersebut mencakup Layanan Aplikasi

Web, Perancangan dan Pemrograman Web, Basis Data, dan Proyek Perangkat Lunak. Terdapat beberapa mata kuliah lain yang bersifat sebagai pilar dasar tentang bagaimana cara berpikir yang benar dalam membangun sebuah aplikasi, seperti Struktur Data dan Algoritma, Dasar Dasar Pemrograman. Mata kuliah seperti Sistem Operasi dan Pemrograman Sistem juga membantu penulis dalam hal-hal teknis menggunakan *environment* Linux, *shell*, dan sebagainya.

Penulis memiliki pekerjaan membangun sebuah web service. Oleh karena itu, mata kuliah terkait teknologi aplikasi *web* sangat membantu penulis dalam menjalani kerja praktik. Layanan Aplikasi Web (LAW) adalah mata kuliah yang menurut penulis paling relevan. Mata kuliah LAW mempelajari dasar membangun *web service*, baik itu teori atau praktek. Dalam mata kuliah LAW, penulis mempelajari tipe *web service* seperti SOAP dan REST, serta implementasi *web service* menggunakan Java EE. Meskipun bahasa pemrograman yang digunakan berbeda dengan yang digunakan pada kerja praktik, penulis memiliki dasar teori yang cukup untuk mengerti tujuan dari *web service* yang akan dibangun. Penulis juga terbantu dengan mata kuliah Perancangan dan Pemrograman Web (PPW). PPW mempelajari kemampuan dasar membangun aplikasi *web* dengan berbagai bahasa pemrograman, termasuk PHP *native* ataupun menggunakan *framework*. Karena *database* merupakan salah satu komponen penting dalam aplikasi *web*, mata kuliah Basis Data (Basdat) juga sangat membantu penulis dalam masa kerja praktik. Dalam mata kuliah Basdat, penulis mempelajari bagaimana data disimpan dalam *table*, properti-properti dalam *table*, *relationship* antar *table*, penyusunan SQL, operasi *join table*, dan sebagainya. Dalam membangun API untuk aplikasi baru FO App, penulis juga merancang beberapa *table*. Selain itu, ilmu dari Basdat juga berguna dalam menyusun bagaimana aplikasi mengakses data dari *database* dengan baik dan benar. Dalam mata kuliah Proyek Perangkat Lunak (PPL), penulis juga membangun sebuah aplikasi *web* yang termasuk aspek *front-end* dan *back-end* dengan Laravel. Penulis pertama kali mendapat pengalaman langsung dengan *framework* Laravel pada mata kuliah PPL ini.

Mata kuliah pada tahun pertama dan kedua berkontribusi dalam membangun dasar berpikir yang baik dalam *programming*. Dalam mata kuliah Struktur Data dan Algoritma (SDA), penulis mempelajari mengenai kompleksitas algoritma. Hal ini sangat berguna untuk membangun aplikasi yang *scalable* sehingga memiliki performa yang baik meskipun mengolah data yang besar. Mata kuliah *programming* yang paling dasar, yaitu Dasar Dasar Pemrograman (DDP), juga berkontribusi dalam membangun cara berpikir serta logika yang baik dalam *programming*.

Meskipun tidak signifikan, mata kuliah seperti Sistem Operasi (OS) dan Pemrograman Sistem (Sysprog) juga berkontribusi dalam mengajarkan penulis bagaimana bekerja menggunakan *shell* dan bagaimana berhadapan dengan *environment server* yang menggunakan distribusi Linux.

BAB 3

PENUTUP

3.1. Kesimpulan

Program internship pada perusahaan Kudo memberikan pengalaman kerja praktik yang berharga. Penulis belajar banyak hal baru baik itu secara teknis maupun non-teknis. Dalam hal teknis, penulis belajar bagaimana membangun aplikasi yang riil akan digunakan dalam perusahaan dan kepada *user*. Pembelajaran dalam hal non-teknis mencakup mengamati budaya *startup*, menganalisa dengan baik kebutuhan *user*, serta berkomunikasi dengan *developer* lain.

3.2. Saran

Pengalaman kerja praktik yang telah dilalui oleh mahasiswa tahun ketiga dalam perkuliahan di Fasilkom sangatlah berharga. Pengalaman tidak hanya mencakup pengalaman mencari tempat kerja praktik, namun juga bagaimana karakteristik pekerjaan dalam suatu divisi bidang tertentu, materi kuliah yang membantu pekerjaan dalam perusahaan, ataupun pengalaman berinteraksi dengan *client*. Ada baiknya pengalaman berharga ini juga dapat dibagikan kepada mahasiswa tahun-tahun sebelumnya. Adanya suatu wadah untuk saling berbagi pengalaman *internship* sangatlah penting, baik itu dalam bentuk sebuah acara terbuka dalam fakultas ataupun media digital untuk mendokumentasikan intisari pengalaman kerja praktik ini.

DAFTAR REFERENSI

- [1] PT. KUDO Teknologi Indonesia. (2016). *APA ITU KUDO*. [online] Available: <https://kudo.co.id/apa-itu-kudo>. [Accessed 21 Nov. 2016].
- [2] Rouse, Margaret. (2014). *Application Program Interface (API)*. [online] Available: <http://searchexchange.techtarget.com/definition/application-program-interface>. [Accessed 21 Nov. 2016].

LAMPIRAN 1
KERANGKA ACUAN KERJA PRAKTIK

LAMPIRAN 2
LOG KERJA PRAKTIK