

LAPORAN KERJA PRAKTEK

Project Mobile Aumi Revi

PT Multipolar Technology Tbk



Dipersiapkan oleh:

Verdy Harmawan / 140707861

PROGRAM STUDI TEKNIK INFORMATIKA

FAKULTAS TEKNOLOGI INDUSTRI

UNIVERSITAS ATMA JAYA YOGYAKARTA

2017

HALAMAN PENGESAHAN
Laporan Kerja Praktek



Laporan ini telah diperiksa dan disetujui

Pada tanggal : 18 Agustus 2017

Oleh :

Dosen Pembimbing

A handwritten signature in black ink, appearing to be "Stephanie Pamela Adithama".

(Stephanie Pamela Adithama, S.T., M.T.)

Pembimbing Lapangan

A handwritten signature in black ink, appearing to be "Hans Kristianto".

(Hans Kristianto)

Kata Pengantar

Puji syukur penulis panjatkan kepada Tuhan Yang Maha Esa yang telah melimpahkan rahmatNya, sehingga penulis dapat melaksanakan Kerja Praktek serta dapat menyelesaikan laporannya tepat waktu dan tanpa adanya halangan yang berarti. Laporan Kerja Praktek ini disusun berdasarkan apa yang telah penulis lakukan pada saat dilapangan yakni pada PT. Multipolar Technology dari tanggal 6 Juli 2017 sampai dengan 18 Agustus 2017. Kerja praktek ini merupakan syarat wajib yang harus ditempuh dalam Program Studi Teknik Informatika Universitas Atmajaya Yogyakarta.

Dalam penyusunan laporan hasil Kerja Praktek ini penulis banyak mendapatkan bantuan dari berbagai pihak, oleh sebab itu penulis ingin mengungkapkan rasa terima kasih kepada Ibu Stephanie Pamela Adhitama S.T. M.T.. selaku dosen pembimbing. Bapak Hans Kristianto selaku pembimbing lapangan yang telah banyak memberikan arahan dan masukan kepada saya dalam melaksanakan Kerja Praktek. Dan Bapak Ngo Yoki Uray selaku ketua dari tim sekaligus yang menerima saya untuk melaksanakan kerja praktek. Tidak lupa juga penulis mengucapkan terima kasih kepada pembimbing lapangan kami yang lain yaitu Bapak Jonathan Tandharma Tjhin, dan Bapak Kelvin Martinez serta pihak – pihak terkait lainnya yang telah banyak membantu baik itu untuk pelaksanaan Kerja Praktek maupun dalam penyelesaian Laporan Kerja Praktek ini.

Penulis juga menyadari, bahwa laporan ini masih jauh dari kesempurnaan. Baik isi maupun tata tulisan. Oleh sebab itu, kritik dan saran dari berbagai arah yang bersifat konstruktif, sangat penulis harapkan demi kesempurnaan laporan ini. Akhir kata penulis berharap, semoga laporan ini tetap ada manfaatnya khususnya bagi penulis pribadi dan tentunya bagi para pembaca pada umumnya.

Yogyakarta, 15 Agustus 2017

Penulis,


Verdy Harmawan

DAFTAR ISI

| | |
|--|------------|
| HALAMAN JUDUL | i |
| HALAMAN PENGESAHAN | ii |
| KATA PENGANTAR | iii |
| DAFTAR ISI | iv |
| 1. BAB 1 PENDAHULUAN | 1 |
| 1.1. Sekilas Perusahaan | 1 |
| 1.2. Sejarah Perusahaan | 3 |
| 1.3. Visi, Misi dan Tujuan Perusahaan | 5 |
| 1.4. Struktur Organisasi | 7 |
| 1.5. Deskripsi Tugas Struktur Organisasi | 10 |
| 1.6. Department TI dalam Perusahaan | 13 |
| 2. BAB II PELAKSANAAN KERJA PRAKTEK | 15 |
| 2.1. Penjelasan Logbook | 15 |
| 2.2. Hasil Pekerjaan Secara Umum | 24 |
| 2.3. Bukti Hasil Pekerjaan | 26 |
| 3. BAB III HASIL PEMBELAJARAN | 42 |
| 3.1. Manfaat Kerja Praktek | 42 |
| 3.2. Penerapan Ilmu dalam Kerja Praktek | 43 |
| 4. BAB IV KESIMPULAN | 45 |
| 4.1. Kesimpulan | 45 |
| 4.2. Saran | 45 |
| 5. BAB V LAMPIRAN | 46 |

DAFTAR GAMBAR

| | |
|--|----|
| Gambar 1.1. Struktur Organisasi(1) | 7 |
| Gambar 1.2. Struktur Organisasi(2) | 8 |
| Gambar 1.3. Struktur Organisasi(3) | 9 |
| Gambar 2.1. Package bagian BO | 26 |
| Gambar 2.2. Package bagian Service | 27 |
| Gambar 2.3. Service Constants | 28 |
| Gambar 2.4. MySQL Workbench | 29 |
| Gambar 2.5. Command Prompt bagian BO | 29 |
| Gambar 2.6. Command Prompt bagian service | 30 |
| Gambar 2.7. Advanced Rest Client (ARC) | 31 |
| Gambar 2.8. Tampil Data pada ARC | 32 |
| Gambar 2.9. Projek Ionic pada VS code | 32 |
| Gambar 2.10. Syntax untuk menjalankan ionic | 33 |
| Gambar 2.11. Code untuk mengambil data pada rest | 34 |
| Gambar 2.12. Code untuk deklarasi halaman | 35 |
| Gambar 2.13. Code untuk navigasi | 35 |
| Gambar 2.14. Code untuk menerima passing data | 36 |
| Gambar 2.15. Code untuk autentikasi data fingerprint | 36 |
| Gambar 2.16. Code untuk autentikasi data NFC | 37 |
| Gambar 2.17. Code untuk mengaktifkan method NFC | 38 |
| Gambar 2.18. Code untuk memanggil method | 38 |
| Gambar 2.19. Code untuk membuat loading | 38 |
| Gambar 2.20. Code untuk menggunakan plugin | 38 |
| Gambar 2.21. Code untuk login | 39 |
| Gambar 2.22. Code untuk membuat toast | 40 |
| Gambar 2.23. Log file | 40 |
| Gambar 2.24. Data pada bitbucket | 41 |

BAB I

PENDAHULUAN

1.1. Sekilas Perusahaan

Berdiri pada tahun 1975, PT Multipolar Technology Tbk (MLPT) adalah *System Integrator* terkemuka di Indonesia, dan merupakan salah satu perusahaan pertama yang memperoleh sertifikasi ISO 9001. Tanggal 8 Juli 2013 merupakan momen bersejarah dalam rekam jejak PT Multipolar Technology Tbk, yang resmi tercatat di Bursa Efek Indonesia (BEI) dengan kode saham MLPT, dan terdaftar sebagai sekuritas syariah. Pemegang saham mayoritas Multipolar Technology adalah PT Multipolar Tbk (MLPL), perusahaan induk terkemuka yang berfokus di segmen ritel dan TMT (Teknologi, Multimedia dan Telekomunikasi) dengan 80% kepemilikan saham dan sisanya 20% dimiliki oleh publik.

Multipolar Technology bersama anak usahanya secara berkesinambungan memberikan layanan menyeluruh dan optimal bagi bisnis konsumennya: PT VisioNet Data Internasional (VDI) dengan fokus pada Business Process Managed Services, dan PT Graha Teknologi Nusantara (GTN) yang berfokus pada Layanan Pengelolaan dan Pengoperasian *Data Center*. Dengan pengalaman yang luas PT Multipolar telah mengembangkan kompetensi dalam area utamanya yaitu :

1. Perangkat Keras dan Layanan Integrasinya

Menyediakan dan melakukan pengaturan perangkat keras Teknologi Informasi (TI) dan perangkat jaringan bagi bisnis konsumennya; memastikan peralihan yang mulus termasuk pengintegrasian dengan infrastruktur yang sudah ada. Dalam hal ini meliputi infrastruktur jaringan, *server enterprise* termasuk *midrange* dan *mainframe*, *storage enterprise*, dan solusi *self-service*/layanan mandiri.

2. Sistem Aplikasi dan Layanan Implementasinya

PT Multipolar memiliki kerangka solusi yang terintegrasi dengan baik dan layanan implementasinya guna mendukung bisnis konsumennya; antara lain Solusi Perbankan, Solusi *High Availability*, Solusi *Information Management*, *Business Analytics*, Solusi *Enterprise Business*, *Service-Oriented Architecture* (SOA), Solusi Virtualisasi, dan Sistem *Enterprise Project Management* (EPM).

3. *IT Consulting Services*

Tim konsultan TI PT Multipolar terdiri dari individu-individu berbakat yang sangat menguasai bidangnya, memiliki pengetahuan global yang luas, dan berpengalaman di bidang konsultan dengan berdasarkan pada budaya profesionalisme yang tinggi, mengadopsi praktik bisnis terbaik, sikap ‘optimis’, dan terbuka terhadap gagasan baru. PT Multipolar akan membantu konsumennya untuk mengidentifikasi solusi yang paling tepat dalam upaya mencapai tujuan perusahaan. Layanan ini dikategorikan kedalam tiga area, yaitu *IT Strategy/IT Planning Consulting Services*, *IT Project Management Consulting Services*, dan *IT Operations Consulting Services*.

4. *Business Process Managed Services*

Layanan ini memiliki cakupan layanan berskala nasional dengan lebih dari 125 titik layanan. PT Multipolar menawarkan solusi alternatif berbasis TI yang ekonomis untuk area *Field Operation Managed Services*, *Branch IT Services*, *Merchant IT Services*, *IT Operation Managed Services*, *IT Application Managed Services*, *Contact Center Services*, dan *Mobile & Payment Services*.

Multipolar Technology adalah mitra terpercaya dari penyedia TI terkemuka di dunia seperti IBM, Cisco, NCR, VMware, Oracle, and Microsoft. Profesionalisme Multipolar Technology telah diakui dengan level kemitraan seperti IBM Premier Partner, Cisco Gold Partner, dan VMware Premier Partner. Multipolar Technology juga telah disertifikasi dengan sertifikasi *Specialty* dari IBM, dan sertifikasi *Authorized Technology Provider* (ATP) dari Cisco, yang membuktikan

kapabilitas dan fokus Multipolar Technology dalam mengembangkan kompetensi teknisnya.

1.2. Sejarah Perusahaan



Sejarah berdirinya usaha menggambarkan kepada pihak-pihak lain yang berhubungan dengan perusahaan ataupun konsumen mengenai dasar atau landasan usaha ini berdiri, apakah cukup kuat secara pengalaman dan keutuhan individu yang terlibat didalamnya. Sejarah menceritakan perjalanan singkat perusahaan, seluruh kejadian yang pernah dialami dalam perkembangan perusahaan.

Awalnya, PT Multipolar Tbk didirikan sebagai peritel elektronik yang saat ini berubah menjadi penyedia solusi informasi dan teknologi (IT). Inovasi PT Multipolar dimulai pada tahun 1975 sebagai pengecer elektronik. Perkembangan Multipolar Technology di bidang *Systems integrator* merupakan langkah yang tepat. Dimulai dari kepemimpinan di sektor perbankan dan keuangan, PT Multipolar telah berkembang dengan pesat dan menyediakan layanan di sektor bisnis lainnya untuk membantu menyelaraskan bisnis konsumennya. Dengan kemampuan yang mengacu pada praktik-praktik bisnis terbaik, PT Multipolar memiliki posisi terbaik untuk menawarkan solusi yang inovatif. Berikut adalah sejarah dari tahun ke tahun dari PT Multipolar :

Tahun 1975 – 1982

- Berdirinya Multipolar.
- Pengecer elektronik.
- Produsen dan penyalur PC.
- Pionir Komputerisasi perbankan dan keuangan.

Tahun 1986 – 1989

- Mitra generasi pertama IBM di Indonesia untuk sektor perbankan.
- Perusahaan TI Indonesia pertama tercatat di BEI (PT Multipolar Tbk/MLPL).

Tahun 1990 – 1997

- Fokus pada layanan *IT System Integration*.
- Perusahaan TI pertama yang memperoleh sertifikasi ISO 9001 untuk perangkat keras, perangkat lunak dan layanan profesional (1997).

Tahun 2002 – 2010

- Membentuk unit bisnis layanan *IT consulting*.
- Memperkenalkan layanan *IT Outsourcing*.
- Pengembangan dari Unit Bisnis *IT Business Group* MLPL menjadi perusahaan dengan fokus di bidang TI.

Tahun 2011 – 2012

- Fokus pada industri strategis.
- Mengembangkan kompetensi baru untuk memenuhi kebutuhan pasar.

Tahun 2013 – Saat ini

- IPO : Tercatat di BEI pada Juli 2013 (Kode saham : MLPT).
- Sebagai salah satu perusahaan penyedia dan pengelola teknologi informasi terbaik dengan fokus pada solusi dan jasa.

- Terus menjadi pemimpin industri yang kuat dalam bidang *system integration* dengan senantiasa menawarkan produk – produk terbaik yang saling melengkapi portofolio produk saat ini.
- Mengembangkan peluang-peluang bisnis yang baru.
- Mempertahankan pertumbuhan laba usaha secara berkelanjutan.

Walaupun Multipolar Technology memulai usahanya sebagai peritel elektronik, saat ini Multipolar Technology juga fokus dalam mengembangkan bisnisnya untuk menjadi pemain besar dalam industri ritel melalui kepemilikannya pada PT Matahari Putra Prima Tbk (MPPA) dan Robbinz Department Store (“RDS”) di Cina. Saat ini, Multipolar memiliki jaringan ritel yang sangat luas yang tersebar di Indonesia dan Cina. Di Indonesia, Multipolar melalui MPPA mengoperasikan 63 Hypermart, 24 Foodmart dan 63 Boston HBC’s. Sedangkan di Cina, Multipolar telah membuka enam toko ritel yang terdiri dari dua Hypermart di Tianjin dan Suzhou dan empat Department Store di Tianjin, Chengdu, Yangzhou, dan Suzhou. Selain itu, Multipolar masih akan terus melakukan pengembangan pada lini bisnis IT dan multimediana dan terus mencari peluang investasi lainnya untuk mencapai portfolio bisnis yang lebih strategis.

Multipolar juga merupakan perusahaan TI pertama yang tercatat di Bursa Efek Indonesia dan memperoleh sertifikasi ISO 9001 dalam bidang perangkat keras, perangkat lunak, dan jasa profesional lainnya. Aliansi dengan penyedia solusi TI kelas dunia seperti IBM, Cisco, Microsoft, NCR, Oracle.

1.3. Visi, Misi dan Tujuan Perusahaan

Visi

"Menjadi penyedia jasa teknologi terkemuka, yang memberikan nilai berkesinambungan bagi para pemangku kepentingan dan hasil positif bagi kehidupan"

Misi

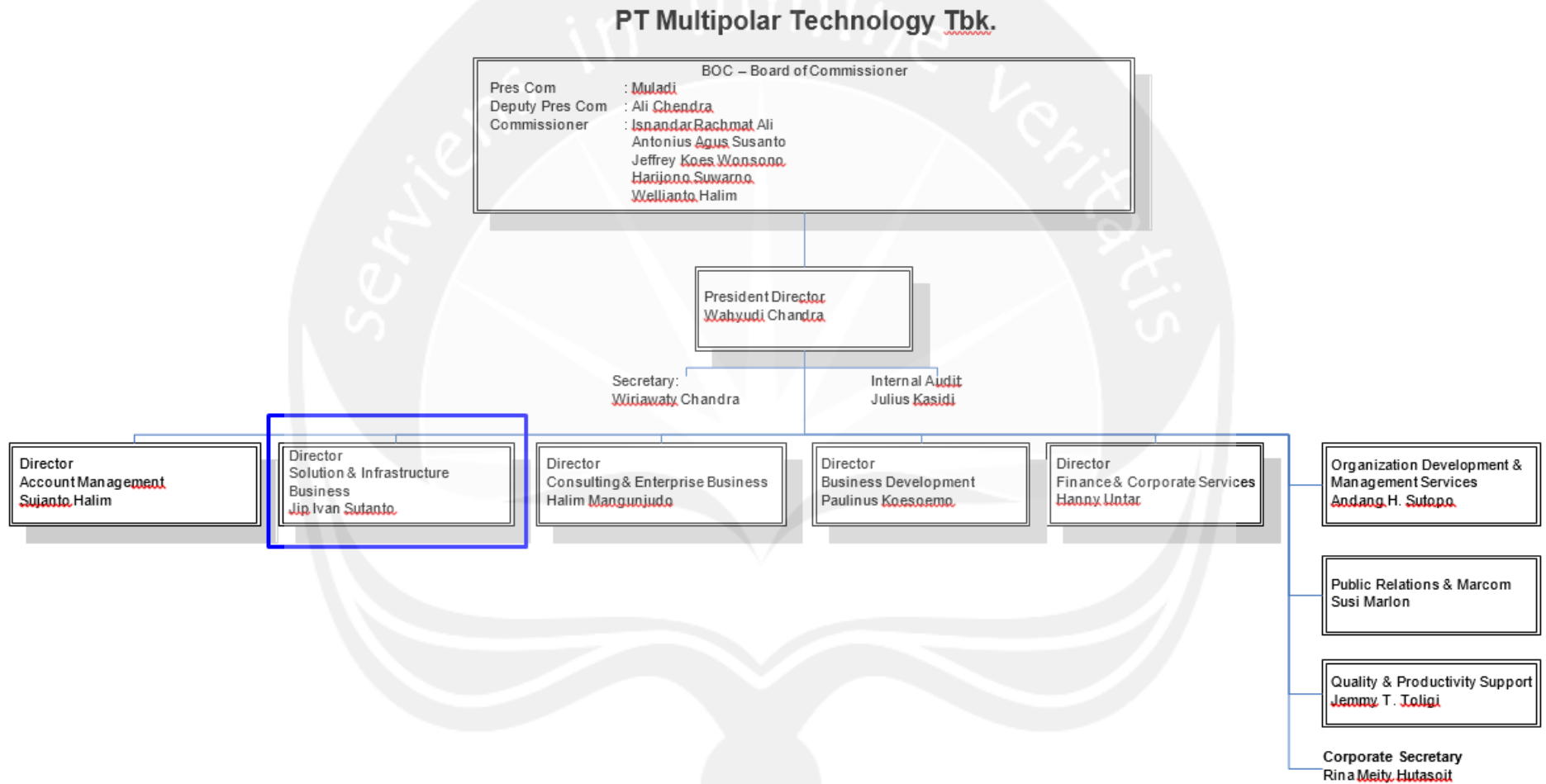
- Menjadi mitra terpercaya melalui penyediaan solusi yang terbaik.
- Menyediakan Sumber Daya Manusia yang berkualitas dengan mengacu pada filosofi Perseroan, guna menjamin pertumbuhan yang berkesinambungan.

Filosofi Perusahaan

Karyawan PT Multipolar menjunjung kode etik profesional demi terciptanya keunggulan tata laksana dan operasional. Program pelatihan karyawan dan standar mutu internal memastikan tercapainya standar mutu yang dibutuhkan untuk mendorong kepuasan pelanggan. PT Multipolar menyediakan lingkungan yang memperhatikan keseimbangan beban kerja sehingga memungkinkan karyawan PT Multipolar menikmati keseimbangan dalam pekerjaan dan kehidupan. Ini semua tercermin dalam gambar di bawah ini:

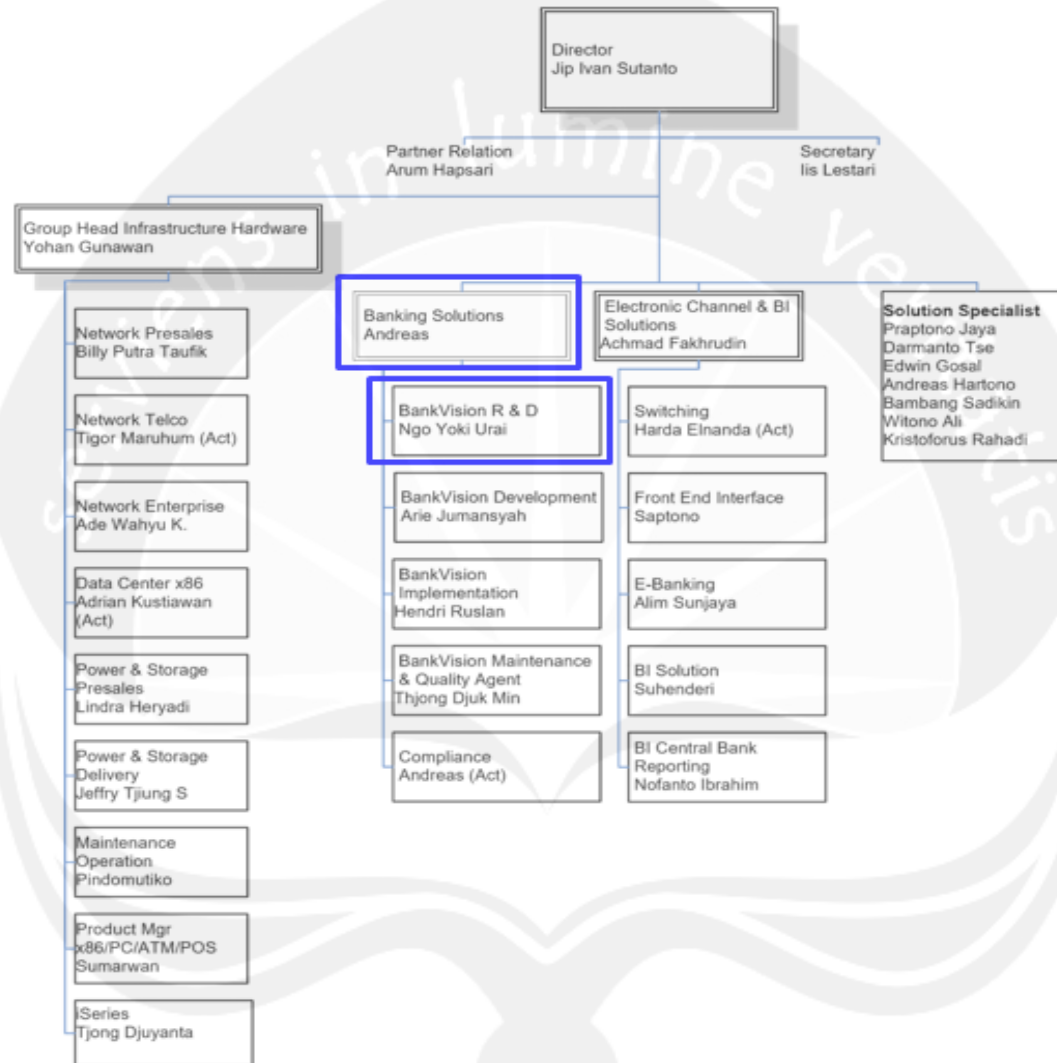


1.4. Struktur Organisasi



Gambar 1.1. Struktur Organisasi(1)

Solution & Infrastructure Business



Gambar 1.2. Struktur Organisasi(2)



Gambar 1.3. Struktur Organisasi(3)

1.5. Deskripsi Tugas Dalam Struktur Organisasi

1. Board of Commissioner

Sekelompok orang yang dipilih atau ditunjuk untuk mengawasi kegiatan suatu perusahaan atau organisasi.

2. President Director

Pimpinan tertinggi dalam suatu perusahaan. Pimpinan tertinggi ini memiliki tanggung jawab dalam memimpin dan mengarahkan perusahaan.

3. Director Account Management

Bertanggung jawab dalam merencanakan, mengarahkan, memonitor dan mengevaluasi sistem kerja akunting untuk pengelolaan data keuangan dan neraca Rugi – Laba perusahaan.

4. Director Solution & Infrastructure Business

- Group Head Infrastructure Hardware

Bertanggung jawab menangani permasalahan ataupun pengembangan pada bagian hardware.

- Banking Solutions

Bertanggung jawab dan mengatur segala sesuatu di dalam penyediaan solusi sistem perbankan dan bertanggung jawab melaporkan hasil kerja divisi kepada director solution & infrastructure.

- BankVision R & D (Research & Development)

Bertugas untuk mengembangkan produk, dan mengembangkan fitur-fitur baru. Dan melakukan riset mengenai teknologi baru yang digunakan untuk pengembangan aplikasi.

- BankVision Development

Bertugas untuk mengembangkan produk sesuai dengan permintaan dari customer (customer request), dan memperbaiki kesalahan yang terjadi (fixing bug).

- **BankVision Implementation**
Bertugas untuk mengimplementasikan produk yang telah dibuat oleh BankVision ke customer baru.
 - **BankVision Maintenance & Quality Agent**
Bertugas untuk melakukan maintenance terhadap produk dari BankVision dari customer yang sudah ada.
 - **Compliance**
Bertugas sebagai penghubung antara BankVision dengan customer.
 - **Electronic Channel & BI Solutions**
Bertanggung jawab menangani permasalahan ataupun pengembangan di bagian ATM dan Internet Banking.
 - **Solution Specialist**
Bertanggung jawab menangani produk-produk khusus diluar dari banking seperti ERP dan CRM.
5. **Director Consulting & Enterprise Business**
Bertugas untuk menentukan perencanaan dan bisnis strategi untuk meningkatkan kemampuan sumber daya perusahaan dalam mewujudkan competitive advantage.
 6. **Director Business Development**
Bertanggung jawab untuk mengembangkan bisnis yang dimiliki oleh perusahaan.
 7. **Director Finance & Corporate Service**
Bertugas untuk menangani layanan-layanan yang disediakan perusahaan kepada pemegang saham dan menangani pelaporan finansial dari perusahaan.
 8. **Organization Development & Management Services**
Bertugas untuk mengatur seluruh kegiatan yang berkaitan dengan layanan pengembangan dan pengelolaan perusahaan.
 9. **Public Relations & Marcom**

Bertugas untuk merencanakan dan mengembangkan komunikasi yang lebih bersifat korporat, membina hubungan baik dengan pihak eksternal perusahaan, sekaligus juga bertanggung jawab terhadap lancarnya komunikasi di tingkat internal perusahaan.

10. Quality & Productivity Support

Bertugas untuk menangani tingkat kualitas dan produktivitas yang dimiliki oleh perusahaan.

11. Corporate Secretary

Bertugas sebagai penghubung antara dewan direksi dengan para pemegang saham di perusahaan.

1.6. Department TI dalam Perusahaan

Department TI dalam PT. Multipolar ini berada pada :

1. Director Solution & Infrastructure Business

- Group Head Infrastructure Hardware
Bertanggung jawab menangani permasalahan ataupun pengembangan pada bagian hardware.
- Banking Solutions
Bertanggung jawab dan mengatur segala sesuatu di dalam penyediaan solusi sistem perbankan dan bertanggung jawab melaporkan hasil kerja divisi kepada director solution & infrastructure.
 - BankVision R & D (Research & Development)
Bertugas untuk mengembangkan produk, dan mengembangkan fitur-fitur baru. Dan melakukan riset mengenai teknologi baru yang digunakan untuk pengembangan aplikasi.
 - BankVision Development
Bertugas untuk mengembangkan produk sesuai dengan permintaan dari customer (customer request), dan memperbaiki kesalahan yang terjadi (fixing bug).
 - BankVision Implementation
Bertugas untuk mengimplementasikan produk yang telah dibuat oleh BankVision ke customer baru.
 - BankVision Maintenance & Quality Agent
Bertugas untuk melakukan maintenance terhadap produk dari BankVision dari customer yang sudah ada.
 - Compliance
Bertugas sebagai penghubung antara BankVision dengan customer.
- Electronic Channel & BI Solutions

Bertanggung jawab menangani permasalahan ataupun pengembangan di bagian ATM dan Internet Banking.

- Solution Specialist

Bertanggung jawab menangani produk-produk khusus diluar dari banking seperti ERP dan CRM.



BAB II

PELAKSANAAN KERJA PRAKTEK

2.1. Penjelasan Logbook

Kamis 6 Juli 2017

Pada hari ini penulis datang ke PT Multipolar Technology untuk mengisi formulir dan menandatangani kontrak kerja praktek dengan pihak *HRD (Human Resource Development)* perusahaan. Disini penulis juga dijelaskan tentang peraturan perusahaan, pakaian yang harus digunakan selama kerja praktek dan waktu kerja. Di perusahaan ini jam operasional di mulai pukul 08.30 – 17.30 dengan jam istirahat pada pukul 12.00 – 13.00. Setelah selesai mendengarkan penjelasan dan menandatangani kontrak, kami diajak untuk ke lantai 4 dimana tempat kami bekerja, sampai di lantai 4 kami berkenalan terlebih dahulu dengan semua karyawan yang berada di lantai 4. Setelah itu kami bertemu dengan pembimbing kami yaitu Pak Hans, dan ketua dari tim yang menerima kami untuk kerja praktek di PT Multipolar Technology yaitu Pak Yoki. Pertama kami dijelaskan tentang proyek apa yang akan kami kerjakan selama disana. Tetapi sebelumnya kami diberikan latihan-latihan terlebih dahulu agar kami dapat belajar sebelum mengerjakan proyek yang sebenarnya. Hari pertama ini penulis mengerjakan instal java, instal android studio, dan membuat aplikasi sederhana yaitu hello world, dan aplikasi sign up pada android studio. Lalu penulis juga mendokumentasikan apa yang telah dikerjakan tersebut.

Jumat 7 Juli 2017

Pada hari ini penulis diajak untuk mengikuti *sharing session*. *Sharing session* disini adalah mengikuti pembelajaran untuk membuat sesuatu atau mengikuti *tutorial* dari satu orang yang mempelajari hal tersebut. Pada *sharing session* kali ini yang dipelajari adalah struts pada java. Setelah itu penulis mempersiapkan tools-tools yang akan digunakan untuk mengerjakan tugas yang diberikan untuk menggunakan ionic

framework. Ionic framework ini menggunakan beberapa *software* yang perlu untuk di instal yaitu git, node.js, dan vs code. Lalu penulis juga setting *software-software* tersebut agar dapat berjalan. *Setting software* tersebut membutuhkan waktu cukup lama agar dapat cocok antara java dan *software* tersebut. Ionic framework merupakan framework yang digunakan untuk aplikasi mobile.

Senin 10 Juli 2017

Pada hari ini pembimbing lapangan memberikan tugas untuk mengerjakan aplikasi sederhana pada ionic yaitu membuat hello world dan membuat tampilan untuk latihan seperti yang diberikan oleh pembimbing. Untuk membuat tampilan ini penulis menggunakan vs code untuk pengerjaannya. Coding pada ionic framework ini seperti coding pada web karena untuk tampilan terdapat di bagian html. Pada ionic terdapat scss yang penggunaannya sama dengan css. Lalu ada juga *ts(typescript)* yang berisi *logic* dari halaman tersebut. Tampilan yang dikerjakan ini berisi menu yang mempunyai fungsionalitas *barcode scanner*, *nfc*, dan *fingerprint*. Tampilan proyek ionic pada vs code dapat dilihat pada gambar 2.9.

Selasa 11 Juli 2017

Pada hari ini pembimbing lapangan memberikan tugas untuk melanjutkan latihan yang sebelumnya yaitu mengerjakan untuk perpindahan halaman atau navigasi dari halaman yang satu ke halaman yang lain. Lalu penulis mengerjakan fungsionalitas untuk membaca *barcode* dari kamera pada *smartphone* menggunakan fungsi *barcode scanner* dan mencari cara menggunakan sensor *nfc* dan *fingerprint* pada ionic.

Rabu 12 Juli 2017

Pada hari ini pembimbing lapangan memberikan tugas untuk kembali melanjutkan latihan sebelumnya yaitu mengerjakan fungsionalitas untuk *nfc* dan *fingerprint* pada ionic. Setelah itu penulis mengerjakan untuk membuat *layout* atau

tampilan untuk projek mobile aumi revi pada ionic sesuai dengan ketentuan yang diberikan oleh pembimbing.

Kamis 13 Juli 2017

Pada hari ini pembimbing lapangan memberikan tugas untuk membuat git/bitbucket untuk dapat menyatukan dengan apa yang dikerjakan oleh penulis dan kelompok dari penulis. Penggunaan git/bitbucket dapat dilihat pada gambar 2.24. Lalu setelah itu penulis melanjutkan projek yaitu membuat navigasi pada *layout* yang dibuat sebelumnya dan perbaikan pada bagian *list* yaitu *tabs* yang masih belum sesuai dengan yang diinginkan. Cara untuk navigasi pada *layout* dapat dilihat pada gambar 2.12.

Jumat 14 Juli 2017

Pada hari ini pembimbing lapangan memberikan tugas untuk memperbaiki *layout* yang masih kurang, dan memberi warna pada *button*, dan juga mematikan fungsi *back button* karena pada aplikasi sudah memiliki *button* untuk kembali.

Senin 17 Juli 2017

Pada hari ini pembimbing lapangan memberikan tugas merapikan *button*, *tabs/navigation bar*, *list*, dan lain lain agar semuanya rata. Setelah itu penulis mengerjakan fungsi *fingerprint* untuk validasi. Penggunaan fungsi *fingerprint* dapat dilihat pada gambar 2.15.

Selasa 18 Juli 2017

Pada hari ini pembimbing lapangan memberikan tugas untuk mencari cara menggunakan *NFC* pada ionic. Lalu setelah itu mencoba menggunakannya pada projek. Fungsi *NFC* ini juga digunakan untuk validasi. Penulis membutuhkan waktu yang cukup lama untuk mencari fungsi *NFC* agar dapat digunakan pada projek ionic tersebut. Penggunaan fungsi *NFC* dapat dilihat pada gambar 2.16 dan 2.17.

Rabu 19 Juli 2017

Pada hari ini pembimbing lapangan memberikan tugas yang sama seperti hari sebelumnya yaitu menggunakan fungsi *NFC* pada projek ionic. Penulis juga merapikan tampilan yang masih kurang.

Kamis 20 Juli 2017

Pada hari ini pembimbing lapangan memberikan tugas untuk memperbaiki fungsi *NFC* yang telah dikerjakan sebelumnya, dan juga penulis diajarkan cara menggunakan *rest web service*. Lalu penulis ditugaskan untuk mencoba menggunakan *rest web service*. Untuk tampilan dari *rest web service* pada eclipse dapat dilihat pada gambar 2.1-2.3.

Jumat 21 Juli 2017

Pada hari ini pembimbing lapangan memberikan tugas untuk menginstal *software* yang dibutuhkan untuk menggunakan *rest web service*, yaitu mysql workbench, eclipse, dan advance rest client. Saat menginstal *software-software* tersebut membutuhkan waktu yang cukup lama, karena terjadi error yang harus diselesaikan satu per satu. Setelah selesai menginstal, penulis mengerjakan bagian *login* pada ionic, agar user yang *login* pada projek ionic hanya *user* yang ada pada *database* di rest. Untuk tampilan *rest web service* pada eclipse dapat dilihat pada gambar 2.1-2.3. Untuk tampilan mysqlworkbench dapat dilihat pada gambar 2.4. Untuk tampilan *advance rest client* dapat dilihat pada gambar 2.7.

Senin 24 Juli 2017

Pada hari ini pembimbing lapangan memberikan tugas untuk membuat *package* pada rest yang digunakan untuk mengerjakan *bo (logic)* dan *service* (penghubung antara rest dan ionic). Setelah itu penulis mengerjakan *bo* dan *service* untuk bagian *get approval request list* yang berfungsi untuk memasukkan dan menampilkan data yang ada pada rest di bagian *request list* sesuai dengan ketentuan

parameter yang dibutuhkan. Untuk bo dan service yang penulis kerjakan dapat dilihat pada gambar 2.1 dan 2.2.

Selasa 25 Juli 2017

Pada hari ini pembimbing lapangan memberikan tugas untuk menyelesaikan semua bo dan service. Penulis memiliki bagian pada bo dan service get approval request list, get approval request detail, dan submit approval. Fungsi dari get approval request detail adalah menampilkan detail dari request list yang sebelumnya, dan submit approval digunakan untuk menampilkan detail dari list yang dipilih dan sudah di *approve*.

Rabu 26 Juli 2017

Pada hari ini pembimbing lapangan memberikan tugas untuk menyelesaikan *login* pada ionic dengan mengambil data yang sudah dibuat di bo dan service pada rest. Setelah penulis selesai menyelesaikan *login* tersebut, lalu penulis mencoba untuk *passing data* dari satu halaman ke halaman yang lain pada ionic. Karena setiap halaman memiliki parameter-parameter yang harus dibawa dari halaman sebelumnya agar data dapat ditampilkan. Untuk bagian dari code *login* dapat dilihat pada gambar 2.21. Untuk bagian dari code *passing data* dapat dilihat pada gambar 2.14.

Kamis 27 Juli 2017

Pada hari ini pembimbing lapangan memberikan tugas untuk mengerjakan *passing data*, lalu menampilkan data pada request list, sesuai dengan bo dan service yang telah dibuat. Untuk menampilkan request list maka diperlukan data yang dibawa dari *login* yaitu userid, sehingga kita harus *passing data* dari *login* untuk mengambil data json untuk userid, lalu dibagian request list penulis membandingkan dengan data yang ada, jika userid yang *login* tersedia, maka data request list akan ditampilkan pada list. Setelah itu menyelesaikan semua *passing data* di halaman yang membutuhkan data dari halaman sebelumnya. Lalu menampilkan data detail request list.

Jumat 28 Juli 2017

Pada hari ini pembimbing lapangan memberikan tugas untuk menambahkan data pada list beserta detailnya, yang sebelumnya hanya tersedia satu data. Sehingga penulis menambahkan hingga tiga data. Setelah itu penulis mencoba menampilkan data yang ditambahkan tersebut pada list yang tersedia. Lalu penulis memberi *loading* pada setiap perpindahan halaman. Untuk potongan code *loading* dapat dilihat pada gambar 2.19.

Senin 31 Juli 2017

Pada hari ini pembimbing lapangan memberikan tugas untuk menggunakan *loop* saat mengakses dan menampilkan data yang diambil dari rest yang berupa *array json* di bagian halaman approval request list. Setelah data berhasil ditampilkan menggunakan *loop*, penulis mengerjakan bagian klik data yang dipilih untuk menampilkan detailnya di halaman yang telah ditentukan. Setelah selesai, penulis membuat halaman baru pada projek ionic yang digunakan jika user memilih *reject*. Sehingga penulis membuat beberapa halaman reject sesuai dengan yang dibutuhkan. Untuk potongan code untuk mengambil data dari rest dapat dilihat pada gambar 2.11.

Selasa 1 Agustus 2017

Pada hari ini penulis memperbaiki history list yang masih *error*, karena hanya muncul satu data. Setelah itu penulis mencari *output* data dari *fingerprint* pada *smartphone*.

Rabu 2 Agustus 2017

Sama seperti hari sebelumnya penulis masih mencari *output* dari data *fingerprint* yang ada pada *smartphone*. Penulis menemukan hasil data *fingerprint* yang masih terenkripsi dan tidak dapat dibuka. Untuk mencari output dari *fingerprint* ini membutuhkan waktu yang cukup lama.

Kamis 3 Agustus 2017

Pada hari ini pembimbing lapangan memberikan tugas untuk memperbaiki kembali *error NFC* yang hanya bisa digunakan satu kali, dan jika keluar ke *page* yang lain, sensor *NFC* tidak dapat digunakan lagi.

Jumat 4 Agustus 2017

Sama seperti hari sebelumnya penulis melanjutkan untuk memperbaiki *NFC*. Penulis berhasil memperbaiki *error* tersebut sehingga data *NFC* sudah dapat terbaca berulang kali.

Selasa 8 Agustus 2017

Pada hari ini penulis mencoba untuk membandingkan data *NFC* dari rest yang dibandingkan dengan data *NFC* dari tagnya. Penulis berhasil membandingkan data tersebut. Lalu penulis memberikan *sharing session*, dimana penulis mengajarkan *basic* dari ionic, yang memiliki judul tutorial dan navigasi data pada ionic ke pegawai/karyawan yang ada di PT Multipolar. Disini penulis memberikan tutorial yang setelah itu memberikan penjelasan sekaligus mengerjakan bersama-sama apa yang penulis berikan di tutorial tersebut.

Rabu 9 Agustus 2017

Pada hari ini penulis mencoba untuk mencari cara agar jika data pada *NFC tag* sama dengan data pada *NFC* di rest maka otomatis akan berpindah ke halaman selanjutnya tanpa perlu diklik. Setelah itu penulis mengikuti *sharing session* yang diajarkan oleh teman kelompok dari penulis dengan topik *tutorial validasi* dan penggunaan *scss* pada ionic.

Kamis 10 Agustus 2017

Pada hari ini penulis mencoba mencari cara agar data pada *fingerprint* di *smartphone* dapat diakses, dan dapat dilihat datanya. Lalu penulis memberikan *sharing session* dengan topik *tutorial navigasi* dan menu pada ionic.

Jumat 11 Agustus 2017

Pada hari ini penulis mencoba memperbaiki perpindahan halaman otomatis pada halaman *NFC* ke halaman selanjutnya, tetapi masih ada sedikit *error* yaitu saat *nfc tag* di tap ke smartphone di halaman selain *NFC* masih error. Setelah itu penulis mengikuti *sharing session* yang diajarkan oleh teman kelompok dari penulis dengan topik *tutorial* penggunaan *barcode scanner* pada ionic.

Senin 14 Agustus 2017

Pada hari ini penulis *pull project* pada bitbucket yang sudah diubah tampilannya pada bitbucket, lalu memasukkan apa yang telah dikerjakan oleh penulis ke project. Setelah itu penulis berhasil memperbaiki *error NFC* yang sebelumnya, dan halaman sudah dapat berpindah secara otomatis tanpa perlu diklik, dan *NFC* sudah berjalan hanya di halaman bagian *NFC*, dan tidak memiliki *error*. Lalu perbaiki *error* pada *push/pop page* di halaman-halaman tertentu dan menghapus *button* yang tidak digunakan yaitu *button* submit di halaman *NFC* dan *Fingerprint*.

Selasa 15 Agustus 2017

Pada hari ini penulis merapikan tampilan bagian halaman *NFC* yaitu memberikan pesan jika *nfc tag* tidak sesuai dengan data *nfc* yaitu *NFC tag is incorrect*. Lalu menambahkan toast, dan mengganti warna pada background dan text di toast tersebut. Toast tersebut terletak di bagian halaman login, yang berfungsi jika user salah memasukkan username atau password maka akan muncul toast dengan tulisan *Username or Password is incorrect*.

Rabu 16 Agustus 2017

Pada hari ini penulis mengikuti *meeting* untuk menunjukkan hasil dari yang penulis dan teman kelompok dari penulis kerjakan. Lalu penulis mengerjakan kunci *orientation* agar tampilan selalu *potrait*. Setelah itu penulis menghapus code yang

tidak terpakai, seperti *import* yang tidak pernah dipakai, dan lain lain, selain itu penulis juga merapikan code yang masih berantakan.

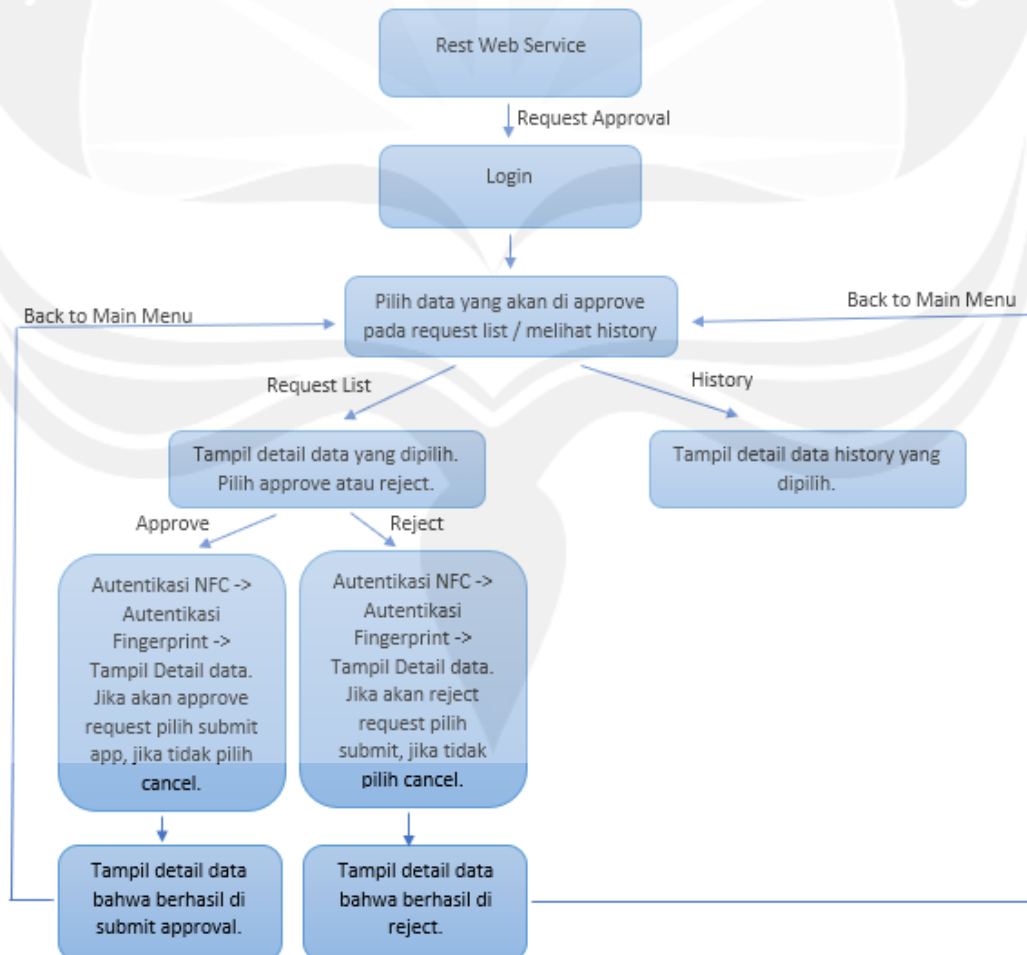
Jumat 18 Agustus 2017

Pada hari ini penulis mendokumentasikan apa yang dikerjakan oleh penulis di word.



2.2 Hasil Pekerjaan Secara Umum

Selama 30 hari kerja penulis melaksanakan kerja praktek, berbagai hal dan pengalaman telah penulis dapatkan ataupun selesai penulis kerjakan. Dalam kerja praktek ini penulis mengerjakan sebuah projek mobile dengan nama aumi revi. Projek aumi revi ini dibuat dengan menggunakan ionic framework yang terkoneksi dengan *rest web service*. Ionic Framework adalah *framework* yang digunakan untuk membangun aplikasi *mobile hybrid* dengan menggunakan html5, css, dan typescript. Karena *hybrid* maka aplikasi ini dapat digunakan di berbagai *platform mobile*. Sedangkan *rest web service* digunakan untuk mengirim data dari *server* yang akan diterima dan digunakan oleh aumi revi. Aumi revi adalah sebuah aplikasi *mobile* yang digunakan oleh *supervisor* untuk melakukan *approval* dari *request* yang diminta oleh *teller*. Cara kerja dari aumi revi dapat dilihat seperti gambar dibawah ini :

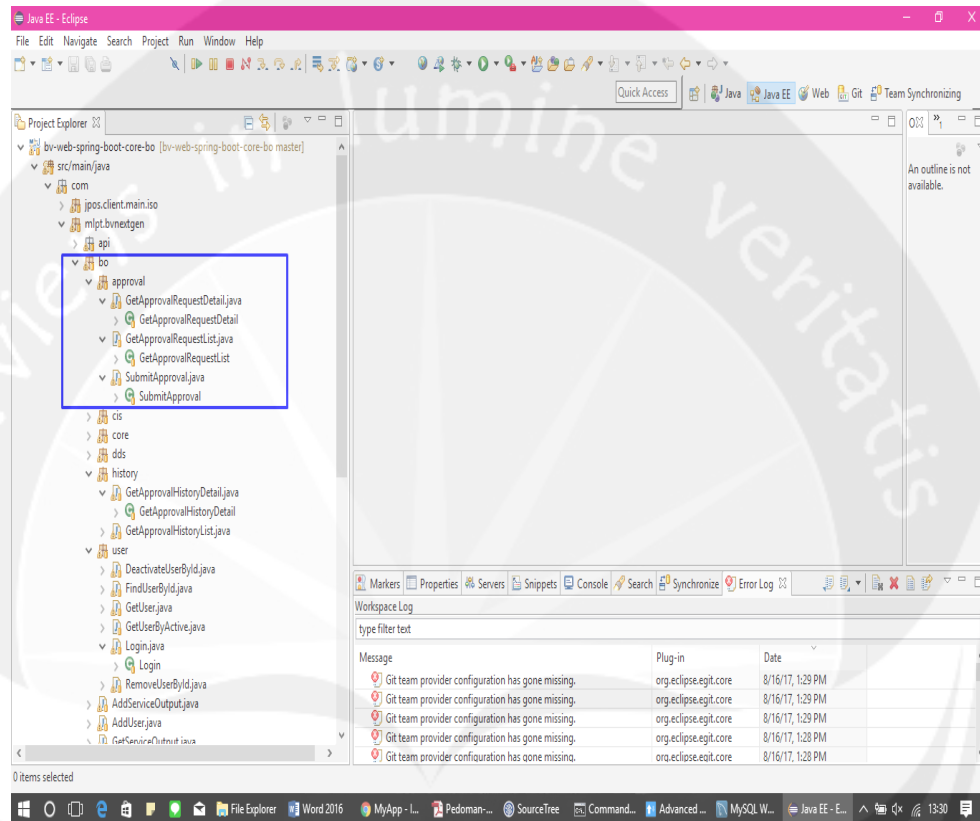


Proses bisnis dari aumi revi seperti berikut : setiap *teller* memiliki *limit*, *limit* tersebut dapat dilewati, tetapi *teller* harus terlebih dahulu meminta *approval* dari *supervisor*. *Supervisor* *approve* permintaan tersebut dengan menggunakan aumi revi. Dalam aumi revi ini, *supervisor* menjalankan langkah-langkah seperti pada gambar diatas. Setelah selesai maka *teller* sudah mendapat *approve* dari *supervisor* dan dapat melewati *limit* tersebut.

Penulis mengerjakan bagian fungsionalitas dari aumi revi. Seperti *NFC*, cara kerja autentikasi *NFC* pada aumi revi adalah *supervisor* akan menempelkan *NFC tag* ke *smartphone*, maka saat *NFC tag* tertempel akan mengecek apakah data *NFC tag* sudah sama dengan data *NFC tag* pada *database*. Jika sudah sama maka autentikasi *NFC* berhasil dan langsung menuju halaman selanjutnya. Data yang dicocokkan pada *NFC* adalah id dari *NFC*, karena setiap *NFC tag* memiliki id yang berbeda-beda. Lalu untuk cara kerja autentikasi *Fingerprint* pada aumi revi adalah *supervisor* harus melakukan autentikasi *NFC* pada sebelumnya, jika sudah berhasil maka pada autentikasi *fingerprint* ini *supervisor* diminta untuk menempelkan jari pada sensor *fingerprint*. Data *fingerprint* yang dibandingkan ini adalah data dari jari yang ditempelkan dengan data *fingerprint* yang ada pada *smartphone*. Jika sudah sama maka akan otomatis menuju ke halaman selanjutnya. Sehingga fungsi autentikasi *NFC* dan *fingerprint* ini digunakan untuk keamanan agar data yang dipakai benar-benar milik *supervisor* yang memiliki data *NFC* dan *fingerprint* yang terdaftar. Untuk potongan code bagian *NFC* dan *Fingerprint* dapat dilihat pada gambar 2.15.-2.17.

2.3 Bukti Hasil Pekerjaan

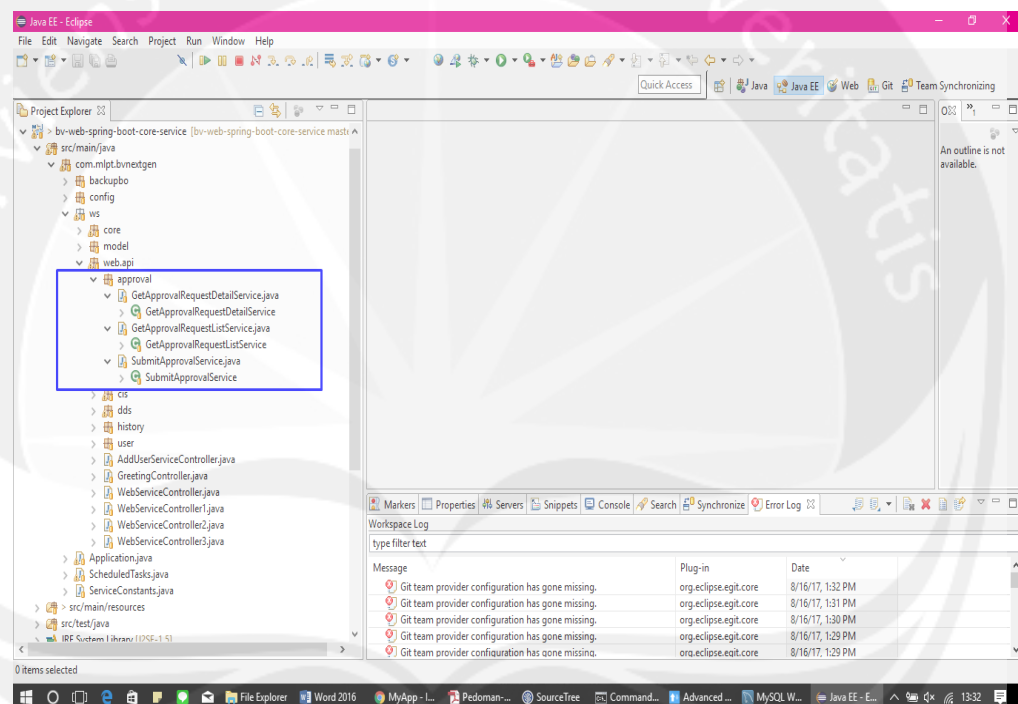
2.3.1 Bagian Rest Web Service pada eclipse



Gambar 2.1. Package bagian BO

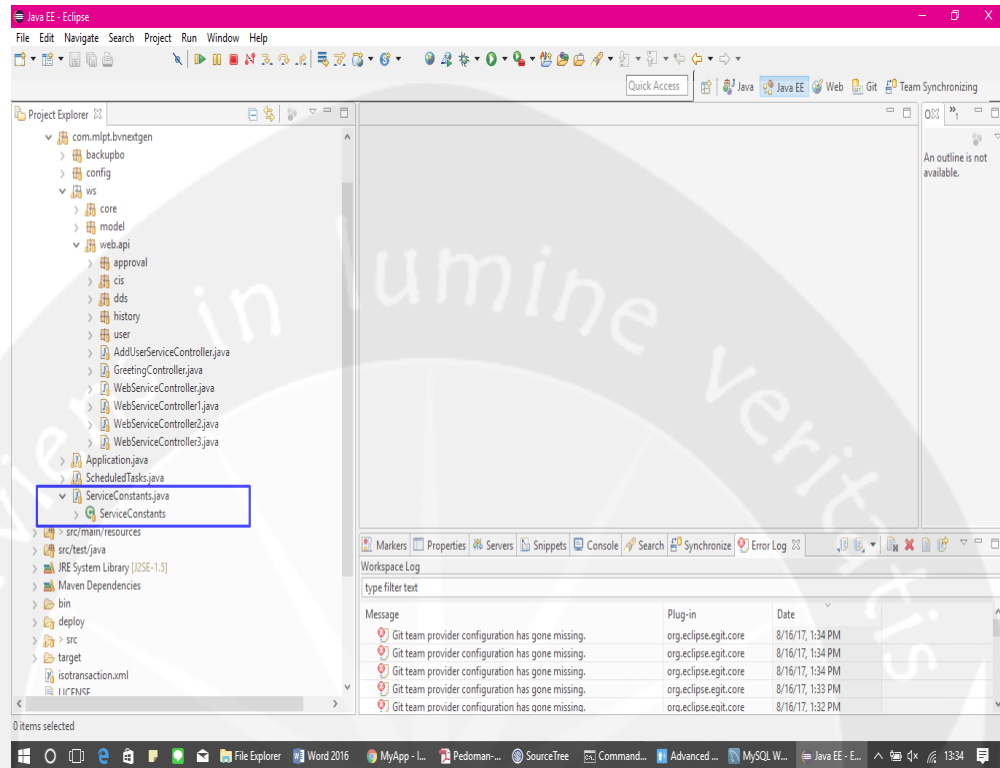
Gambar 2.1 merupakan *package* bagian BO yang dibuat oleh penulis yang merupakan bagian approval, BO berisi *logic* yang akan dikerjakan. Pada bagian BO GetApprovalRequestList berisi *logic* untuk menerima inputan *user* yaitu userid untuk dibandingkan dengan userid yang tersedia, jika sama maka tampilkan data-data yang ada di list yang telah penulis buat. Lalu bagian BO GetApprovalRequestDetail berisi *logic* untuk menerima inputan *user* yaitu userid, transactiontypeid, transactionid, dan traceid, untuk dibandingkan dengan userid, transactiontypeid, transactionid, dan traceid yang tersedia. Jika sama maka akan ditampilkan detail datanya yang telah penulis buat

sebelumnya yang sesuai dengan *userid*, *transactiontypeid*, *transactionid*, dan *traceid* tersebut. Lalu bagian *SubmitApproval* hampir sama dengan bagian *GetApprovalRequestDetail* yaitu menerima inputan *user* yaitu *userid*, *transactiontypeid*, *transactionid*, *traceid*, dan *statussubmit* yang inputannya harus *approved*. Setelah dibandingkan dengan data yang tersedia, maka jika sama akan ditampilkan data detailnya yang telah dibuat penulis.



Gambar 2.2. Package bagian Service

Gambar 2.2 merupakan *package* bagian service yang dibuat oleh penulis yang merupakan bagian approval, service merupakan penghubung *rest web service* agar dapat diakses oleh projek ionic yang penulis kerjakan. Pada bagian service *GetApprovalRequestListService*, *GetApprovalRequestDetailService*, dan *SubmitApprovalService* isinya hampir sama yaitu menampilkan logic yang dibuat oleh penulis di bagian BO ke dalam *URL* yang dibuat di service constant.

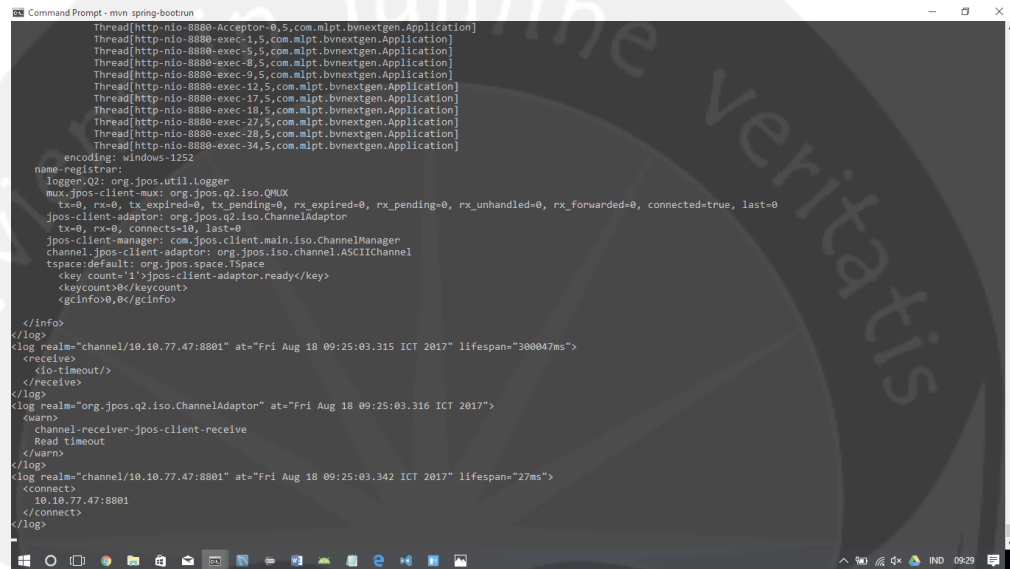


Gambar 2.3. Service Constants

Gambar 2.3 merupakan bagian service constants, service constants berisi url dari service-service yang telah dibuat sebelumnya, sehingga url tersebut dapat diakses.

2.3.2. Proses mengaktifkan/menyalakan rest web service

Gambar 2.5 merupakan *screenshot* dari command prompt, untuk menjalankan bagian BO yang sudah dibuat sebelumnya. Caranya adalah arahkan cmd ke direktori bagian BO, lalu ketikkan mvn clean package install jika tidak ada error maka muncul *build success*.

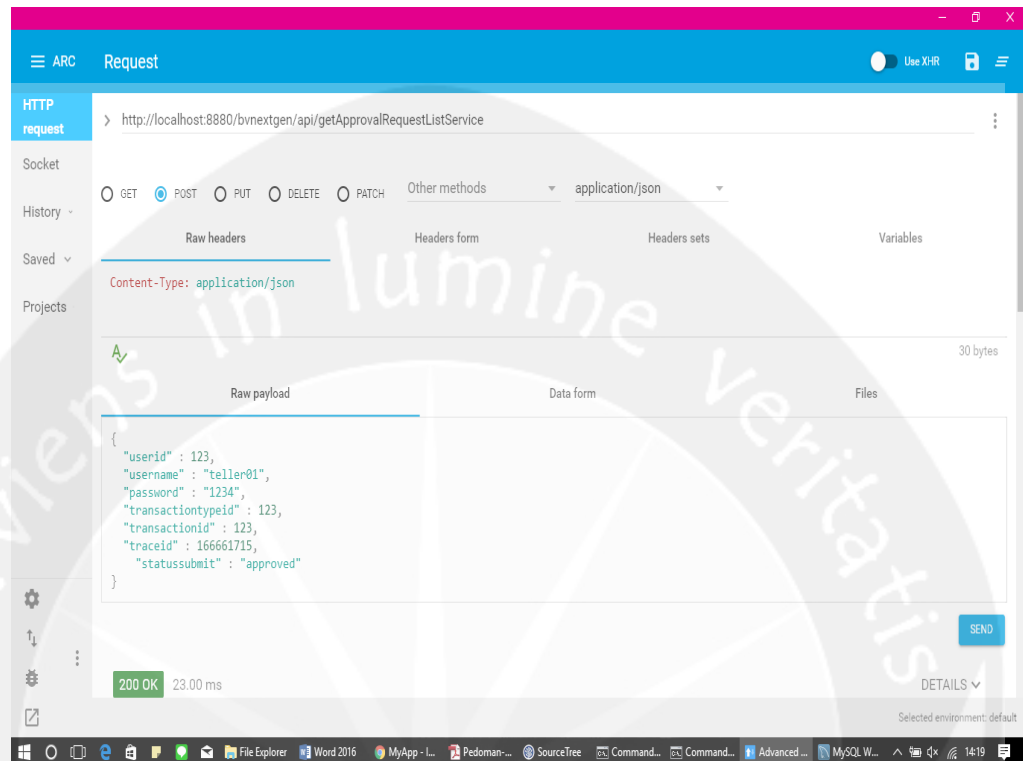


```
Command Prompt - mvn spring-boot:run
Thread[http-nio-8880-Acceptor-0,5,com.mlpt.bnnextgen.Application]
Thread[http-nio-8880-exec-1,5,com.mlpt.bnnextgen.Application]
Thread[http-nio-8880-exec-5,5,com.mlpt.bnnextgen.Application]
Thread[http-nio-8880-exec-8,5,com.mlpt.bnnextgen.Application]
Thread[http-nio-8880-exec-9,5,com.mlpt.bnnextgen.Application]
Thread[http-nio-8880-exec-12,5,com.mlpt.bnnextgen.Application]
Thread[http-nio-8880-exec-17,5,com.mlpt.bnnextgen.Application]
Thread[http-nio-8880-exec-18,5,com.mlpt.bnnextgen.Application]
Thread[http-nio-8880-exec-27,5,com.mlpt.bnnextgen.Application]
Thread[http-nio-8880-exec-28,5,com.mlpt.bnnextgen.Application]
Thread[http-nio-8880-exec-34,5,com.mlpt.bnnextgen.Application]
encoding: windows-1252
name=registran:
  logger.q2: org.jpos.util.Logger
  mux.jpos-client-mux: org.jpos.q2.iso.QMUX
    tx=0, rx=0, tx_expired=0, tx_pending=0, rx_expired=0, rx_pending=0, rx_unhandled=0, rx_forwarded=0, connected=true, last=0
  jpos-client-adaptor: org.jpos.q2.iso.ChannelAdaptor
    tx=0, rx=0, connects=10, last=0
  jpos-client-manager: com.jpos.client.main.iso.ChannelManager
  channel.jpos-client-adaptor: org.jpos.iso.channel.ASCIIChannel
  tspace:default: org.jpos.space.TSpace
    <key count=1>jpos-client-adaptor.ready</key>
    <keycount>0</keycount>
    <ginfo>0</ginfo>
</Info>
</log>
<log realm="channel/10.10.77.47:8801" at="Fri Aug 18 09:25:03.315 ICT 2017" lifespan="300047ms">
  <receive>
    <io-timeout/>
  </receive>
</log>
<log realm="org.jpos.q2.iso.ChannelAdaptor" at="Fri Aug 18 09:25:03.316 ICT 2017">
  <warn>
    channel-receiver-jpos-client-receive
    Read timeout
  </warn>
</log>
<log realm="channel/10.10.77.47:8801" at="Fri Aug 18 09:25:03.342 ICT 2017" lifespan="27ms">
  <connect>
    10.10.77.47:8801
  </connect>
</log>
```

Gambar 2.6. Command Prompt bagian service

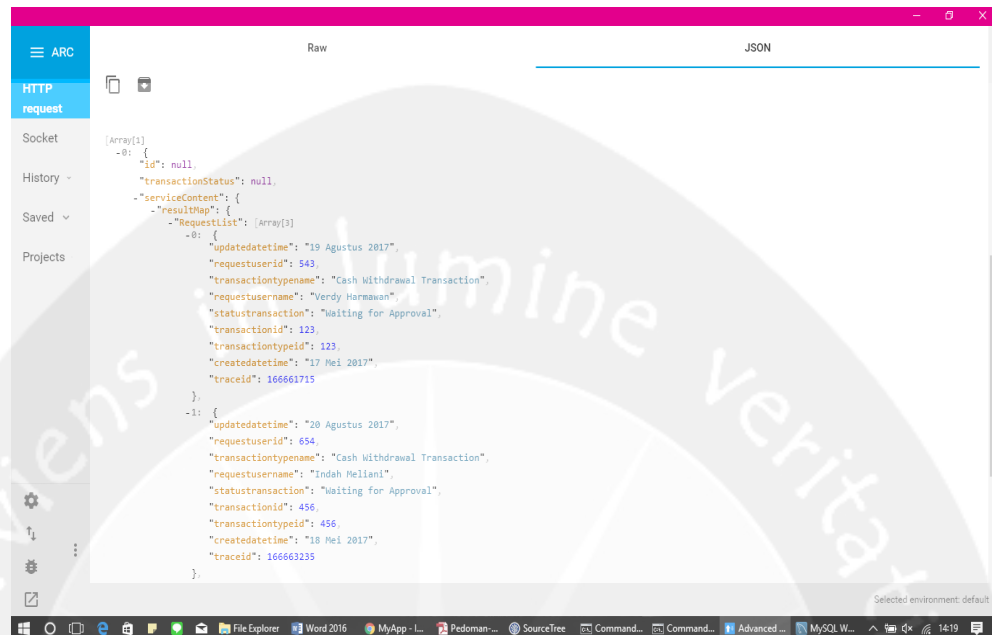
Gambar 2.6 merupakan *screenshot* dari command prompt, untuk menjalankan bagian service yang sudah dibuat sebelumnya. Caranya adalah arahkan cmd ke direktori bagian service, lalu ketikkan mvn spring-boot:run. Jika *rest web service* akan digunakan maka service ini harus diaktifkan, jika tidak maka data pada *rest web service* tidak dapat diakses.

2.3.3. Pengecekan BO dan Service



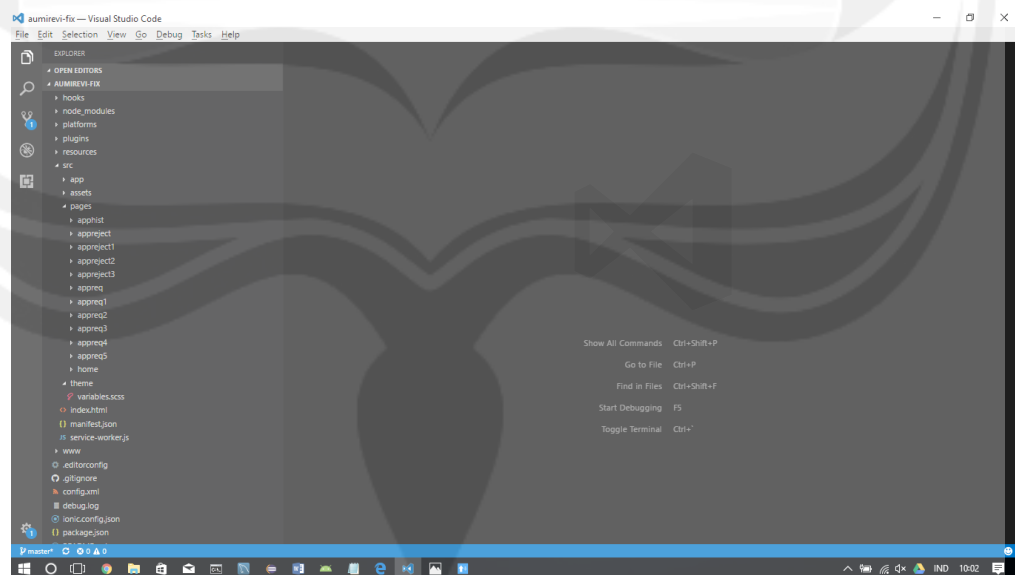
Gambar 2.7. Advanced Rest Client (ARC)

Gambar 2.7 merupakan *screenshot* dari aplikasi Advanced Rest Client yang digunakan untuk mencoba apakah BO dan Service yang dibuat di eclipse sudah benar atau masih error. Caranya adalah dengan memasukkan *URL* dibagian atas sesuai dengan *URL* yang dibuat. Lalu masukkan data-data inputan yang dibutuhkan contohnya pada bagian `getApprovalRequestList` membutuhkan `userid`, maka harus ada inputan `userid`. Setelah itu dapat klik `send`, jika sudah benar maka akan ditampilkan datanya seperti berikut :



Gambar 2.8. Tampil Data pada ARC

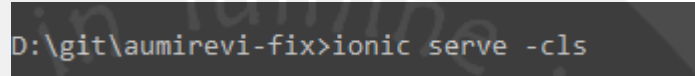
2.3.4. Proyek mobile aumi revi dengan ionic framework pada VS code



Gambar 2.9. Proyek Ionic pada VS code

Gambar 2.9 merupakan *screenshot* dari Vscod, dibagian kiri terlihat halaman-halaman yang telah dibuat yaitu di dalam direktori src->pages. Setiap page terdiri dari 3 file yaitu .html, .scss, .ts(typescript). .html adalah

bagian yang ditampilkan di layar. Lalu `.scss` adalah bagian yang digunakan untuk memodifikasi tampilan, seperti ubah warna *background*, ubah ukuran *font*, dan lain lain. Lalu `.ts` adalah bagian *logic* atau apa yang dikerjakan oleh halaman tersebut.



```
D:\git\aumirevi-fix>ionic serve -cls
```

Gambar 2.10. Syntax untuk menjalankan ionic

Gambar 2.10 merupakan *screenshot* dari command prompt yang digunakan untuk menjalankan/test aplikasi ionic yang telah dibuat, jika berhasil maka projek yang dibuat akan ditampilkan di browser dengan tampilan *smartphone*. Untuk menjalankan aplikasi ionic, juga dapat langsung ke *smartphone* android dengan syntax `ionic cordova run android`, maka aplikasi yang dibuat akan terinstal di *smartphone* yang tersambung dengan laptop. Lalu untuk membuat file `.apk` dapat digunakan dengan syntax `ionic cordova build android`.

```

return new Promise((resolve, reject) => {
  let body = JSON.stringify({userid: this.getid(), transactiontypeid: this.gettranstypeid(),
    transactionid: this.gettransid(), traceid: this.gettraceid()
  });
  let headers = new Headers({
    "Content-Type": "application/json", "Accept": "application/json"
  });
  let options = new RequestOptions({
    headers: headers
  });
  this.showLoader();
  this.http.post(apiUrl+'getApprovalRequestDetailService', body, options)
    .subscribe((res) => {
      if(res.json())
      {
        this.loading.dismiss();
        //ambil data json
        this.dataaccountfrom=res.json()["0"].serviceContent.resultMap.accountfrom;
        this.dataaccountto=res.json()["0"].serviceContent.resultMap.accountto;
        this.dataamount=res.json()["0"].serviceContent.resultMap.amount;
        this.datacreatedatime=res.json()["0"].serviceContent.resultMap.createdatime;
        this.datacurrcode=res.json()["0"].serviceContent.resultMap.currcode;
        this.dataremark=res.json()["0"].serviceContent.resultMap.remark;
        this.datarequestuserid=res.json()["0"].serviceContent.resultMap.requestuserid;
        this.datarequestusername=res.json()["0"].serviceContent.resultMap.requestusername;
        this.datastatustransaction=res.json()["0"].serviceContent.resultMap.statustransaction;
        this.datatraceid=res.json()["0"].serviceContent.resultMap.traceid;
        this.datatransactionid=res.json()["0"].serviceContent.resultMap.transactionid;
        this.datatransactiontypeid=res.json()["0"].serviceContent.resultMap.transactiontypeid;
        this.datatransactiontypename=res.json()["0"].serviceContent.resultMap.transactiontypename;
        this.dataupdatedatime=res.json()["0"].serviceContent.resultMap.updatedatime;
        resolve(res.json());
        console.log(res.json());
      }
    }, (err) => {
      this.loading.dismiss();
      reject(err);
      console.log("Error");
    });
});

```

Gambar 2.11. Code untuk mengambil data pada rest

Gambar 2.11 merupakan *screenshot* dari code proyek yaitu bagian untuk mengakses data yang ada pada *rest web service* di proyek ionic.

```

9
10 import { MyApp } from '../app.component';
11 import { HomePage } from '../pages/home/home';
12 import { AppreqPage } from '../pages/appreq/appreq';
13 import { Appreq1Page } from '../pages/appreq1/appreq1';
14 import { Appreq2Page } from '../pages/appreq2/appreq2';
15 import { Appreq3Page } from '../pages/appreq3/appreq3';
16 import { Appreq4Page } from '../pages/appreq4/appreq4';
17 import { Appreq5Page } from '../pages/appreq5/appreq5';
18 import { ApphistPage } from '../pages/apphist/apphist';
19 import { AuthService } from '../pages/home/authservice';
20 import { ApprejectPage } from '../pages/appreject/appreject';
21 import { Appreject1Page } from '../pages/appreject1/appreject1';
22 import { Appreject2Page } from '../pages/appreject2/appreject2';
23 import { Appreject3Page } from '../pages/appreject3/appreject3';
24 @NgModule({
25   declarations: [
26     MyApp,
27     HomePage,
28     AppreqPage,
29     Appreq1Page,
30     Appreq2Page,
31     Appreq3Page,
32     Appreq4Page,
33     Appreq5Page,
34     ApphistPage,
35     ApprejectPage,
36     Appreject1Page,
37     Appreject2Page,
38     Appreject3Page,
39   ],
40   imports: [
41     BrowserModule,
42     HttpClientModule,
43     IonicModule.forRoot(MyApp)
44   ],
45   bootstrap: [IonicApp],
46   entryComponents: [
47     MyApp,
48     HomePage,
49     AppreqPage,
50     Appreq1Page,
51     Appreq2Page,
52     Appreq3Page,
53     Appreq4Page,
54     Appreq5Page,
55     ApphistPage,
56     ApprejectPage,
57     Appreject1Page,

```

Gambar 2.12. Code untuk deklarasi halaman

Gambar 2.12 merupakan *screenshot* dari code proyek yaitu bagian `app.module.ts` yang berisi deklarasi dari halaman-halaman yang digunakan pada proyek dan juga deklarasi dari *plugin* yang digunakan.

```

loadappreq(i){
  this.transypeidreq=this.datareq["0"].serviceContent.resultMap.RequestList[i].transactiontypeid;
  this.transidreq=this.datareq["0"].serviceContent.resultMap.RequestList[i].transactionid;
  this.traceidreq=this.datareq["0"].serviceContent.resultMap.RequestList[i].traceid;

  this.navCtrl.push(Appreq1Page, {param1: this.getid(), param2: this.transypeidreq, param3: this.transidreq, param4: this.traceidreq, paramfc: this.getnfc()});
}

```

Gambar 2.13. Code untuk navigasi

Gambar 2.13 merupakan *screenshot* dari code proyek yang digunakan untuk navigasi atau berpindah ke halaman yang lain, pada code tersebut maksudnya adalah akan berpindah ke halaman `appreq1page` dengan membawa parameter yang akan digunakan untuk *passing data*. Jika tidak ingin membawa parameter untuk berpindah halaman, maka bagian parameter dapat dihapus.

Lalu untuk kembali ke halaman sebelumnya dapat menggunakan *code* `this.navCtrl.pop({});`.

```
this.parameter1 = navParams.get('param1');
this.parameter2 = navParams.get('param2');
this.parameter3 = navParams.get('param3');
this.parameter4 = navParams.get('param4');
this.parameternfc=navParams.get('paramnfc');
this.setid(this.parameter1);
this.settranstypeid(this.parameter2);
this.settransid(this.parameter3);
this.settraceid(this.parameter4);
this.getid();
this.gettranstypeid();
this.gettransid();
this.gettraceid();
this.setnfc(this.parameternfc);
this.getnfc();
```

Gambar 2.14. Code untuk menerima passing data

Gambar 2.14 merupakan *screenshot* dari *code* proyek yang digunakan untuk menerima parameter/data yang dibawa dari halaman sebelumnya, contohnya data yang dibawa pada halaman sebelumnya yaitu param1, akan disimpan di variabel parameter1, sehingga parameter1 sudah berisi data yang sama dengan param1.

```
public showFingerprintAuthDlg(){
    this.androidFingerprintAuth.isAvailable()
        .then((result)=> {
            if(result.isAvailable){
                // it is available
                this.androidFingerprintAuth.encrypt({ clientId: 'myAppName', username: 'myUsername', password: 'myPassword' })
                    .then(result => {
                        if(result.withFingerprint) {
                            console.log('Successfully encrypted credentials.');
```

Gambar 2.15. Code untuk autentikasi data fingerprint

Gambar 2.15 merupakan *method* yang digunakan untuk mencocokkan antara data *fingerprint* yang ada di *device* dengan *fingerprint sensor*, jika sama maka akan lanjut ke halaman selanjutnya secara otomatis, jika tidak sama maka *user* akan diminta mengulangi kembali sampai sesuai dengan data *fingerprint* pada *device*.

```
onNfc: function(nfcEvent) {
  console.log("nfc is on");
  var tag = nfcEvent.tag;
  if (!document.getElementById('messageDiv'))
  {
  }
  else
  {
    var nfcvar=$('#varnfc').text();
    var nfcvar1=$('#varnfc1').text();
    console.log("data nfc = " + nfcvar);
    console.log("data nfc 1 = " + nfcvar1);
    if(nfcvar=="0" && nfcvar1=="")
    {
      console.log("masuk if");
      messageDiv.innerHTML = nfc.bytesToHexString(tag.id);

      if(document.getElementById('btnvalidate1'))
      {
        console.log("mau ngeklik ni");
        document.getElementById('btnvalidate1').click();
      }
      else
      {
        console.log("mau klikbtnvalide ni");
        document.getElementById('btnvalidate').click();
      }
    }else
    {
      console.log("masuk else bung");
    }
  }
},
```

Gambar 2.16. Code untuk autentikasi data NFC

Gambar 2.16 merupakan *method* yang digunakan untuk mencocokkan antara data pada *NFC tag* dengan data di *rest web service*. Jika *NFC* di tap di *device* maka otomatis akan menjalankan *method* ini, lalu jika datanya sesuai akan otomatis lanjut ke halaman berikutnya.

```
<script type="text/javascript" src="js/index.js"></script>
<script type="text/javascript">
  app.initialize();
</script>
```

Gambar 2.17. Code untuk mengaktifkan method NFC

Gambar 2.17 merupakan *script* yang digunakan untuk mengaktifkan *method NFC* agar dapat digunakan.

```
<button ion-button (click)="loadlogout();" >
```

Gambar 2.18. Code untuk memanggil method

Gambar 2.18 merupakan *code* pada bagian.html yaitu pada *button* yang digunakan untuk memanggil *method* yang telah dibuat.

```
showLoader(){
  this.loading = this.loadingCtrl.create({content: 'Authenticating...'});
  this.loading.present();
}
```

Gambar 2.19. Code untuk membuat loading

Gambar 2.19 merupakan *method* yang digunakan untuk membuat *loading*, lalu menampilkannya dan tulisan dari *loading* dapat diganti sesuai dengan yang diinginkan.

```
ionic cordova plugin add cordova-plugin-android-fingerprint-auth
npm install --save @ionic-native/android-fingerprint-auth
```

Gambar 2.20. Code untuk menggunakan plugin

Gambar 2.20 merupakan *syntax* yang digunakan untuk menginstal atau menambahkan *plugin* yang akan digunakan di proyek, contoh nya adalah *plugin fingerprint*. *Syntax* tersebut diketikan di command prompt yang sudah berada di direktori proyek ionic.

```
login(credentials) {
  return new Promise((resolve, reject) => {
    let body = JSON.stringify({
      username: credentials.username,
      password: credentials.password
    });
    let headers = new Headers({
      "Content-Type": "application/json",
      "Accept": "application/json"
    });
    let options = new RequestOptions({
      headers: headers
    });

    this.http.post(apiUrl+'loginService', body, options)
      .subscribe((res) => {
        if(res.json()["0"].serviceContent.resultMap.status)
        {
          resolve(res.json());
          console.log(res.json());
        }
        else
        {
          this.showToast();
        }
      }, (err) => {
        reject(err);
        console.log("Error");
      });
  });
}
```

Gambar 2.21. Code untuk login

Gambar 2.21 merupakan *method* yang digunakan untuk login, dilakukan pengecekan yaitu username , dan password yang diinputkan dengan data username , dan password yang ada pada *rest web service*, jika statusnya *true* maka akan lanjut ke halaman selanjutnya. Jika salah akan muncul peringatan.

```

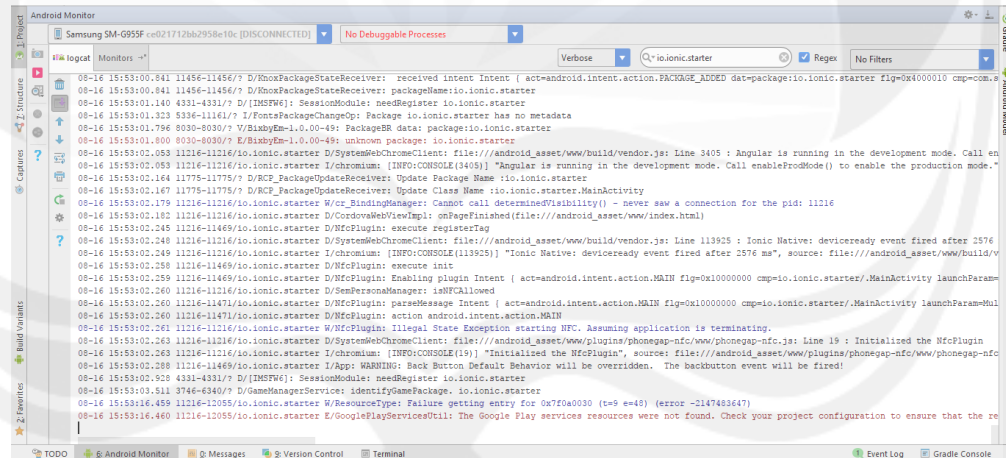
showToast() {
  let toast = this.toastCtrl.create({
    message: 'Username or Password is incorrect',
    duration: 1500,
    position: 'top',
    cssClass: "toastcss",
  });
  toast.present(toast);
}

```

Gambar 2.22. Code untuk membuat toast

Gambar 2.22 merupakan *method* yang digunakan untuk membuat toast/pesan peringatan. Isinya adalah message yang akan ditampilkan, lalu *duration* adalah waktu lamanya toast ditampilkan, *position* adalah tempat toast ditampilkan, dan *cssClass* adalah modifikasi tampilan toast.

2.3.5. Melihat log pada smartphone menggunakan android studio



Gambar 2.23. Log file

Gambar 2.23 merupakan *screenshot log* dari android studio. *Log* ini digunakan untuk mencari error, dan mengecek jalannya aplikasi. Dengan *log* ini maka penulis dapat mengetahui terdapat error dibagian mana dan dapat lebih mudah untuk memperbaikinya.

2.3.5. Menyatukan file dengan BitBucket

| Author | Commit | Message | Date | Builds |
|-------------------|---------|--|------------|--------|
| Verdy | c44f39 | rapiin code & buang code | 2 days ago | |
| Verdy | 0bba49 | add plugins | 2 days ago | |
| Verdy | 7cefa9b | add Index js for nfc | 2 days ago | |
| Verdy | c081dcc | Initial Commit | 3 days ago | |
| Verdy | db9502e | Initial Commit | 3 days ago | |
| Indah Meliani ... | c7b0fa3 | layout seluruh form (color) v.1 | 2017-08-03 | |
| Verdy | a49682f | perbaikan error tampil data pada list | 2017-08-01 | |
| Indah Meliani ... | a7018ad | tampil data di history detail dan klik di history list | 2017-08-01 | |
| Verdy | 6ec73d2 | buat halaman-halaman reject, tampil dan klik pada appreqlist | 2017-07-31 | |
| Verdy | 83c337a | tambah loading | 2017-07-28 | |
| Indah Meliani ... | d1f61c2 | submit approval | 2017-07-28 | |
| Verdy Harmawan | 1f2f328 | approval history (list + detail) (indah) | 2017-07-27 | |
| Verdy | f9666f9 | halaman appreq4 tampil detail | 2017-07-27 | |
| | 93900bf | tampil detail approval request | 2017-07-27 | |

Gambar 2.24. Data pada bitbucket

Gambar 2.24 merupakan file-file yang telah dicommit di dalam bitbucket, yang dapat diakses oleh tim.

BAB III

HASIL PEMBELAJARAN

3.1 Manfaat Kerja Praktek

Penulis dapat mengambil beberapa manfaat dari pelaksanaan kerja praktek yang telah dilakukan oleh penulis, yaitu:

Manfaat bagi penulis:

1. Penulis dapat mengetahui dan memahami aplikasi ilmu yang telah didapat di perusahaan dalam proses pengerjaan proyek yang diberikan.
2. Penulis dapat mengetahui produktivitas dan sistem kerja di dalam perusahaan.
3. Penulis dapat memahami dan mengerti secara langsung sistem kerja perusahaan dan terjun langsung atau ikut serta dalam proses kerja di perusahaan.
4. Penulis dapat mengenal dan membiasakan diri pada saat pelaksanaan kerja praktek terhadap suasana kerja sebenarnya sehingga dapat membangun etos kerja yang baik, serta sebagai upaya untuk memperluas wawasan kerja bagi penulis.
5. Penulis mampu menerapkan dan membandingkan bagaimana disiplin ilmu yang didapat dalam dunia pendidikan dengan kenyataan di lapangan kerja.
6. Penulis dapat menambah kreatifitas diri dan mengasah kemampuan yang dimiliki dalam menyelesaikan masalah atau tugas yang diberikan sehingga dapat belajar untuk bisa lebih baik lagi.
7. Penulis dapat memperluas jalinan persaudaraan dengan orang-orang baru di perusahaan dan belajar bekerja dengan baik didalam tim.

Manfaat bagi Universitas Atma Jaya Yogyakarta :

1. Memperoleh masukan terkait isu-isu terkini dalam dunia kerja serta usaha sebagai bahan pengembangan penelitian dan pendidikan.

2. Meningkatkan dan memperluas jaringan kerjasama (network) dengan institusi bisnis baik swasta, BUMN, dan sebagainya, khususnya PT. Multipolar Technology Tbk (MLPT).
3. Universitas dapat meningkatkan kualitas lulusannya melalui pengalaman kerja praktek mahasiswa.
4. Universitas semakin dikenal di dunia industri.

Manfaat bagi PT. Multipolar Technology Tbk (MLPT) :

1. Menjalin kerjasama yang baik antara pihak perusahaan dengan lembaga pendidikan khususnya Universitas Atma Jaya Yogyakarta.
2. Perusahaan mendapat saran dan kritik yang membangun dari mahasiswa-mahasiswa yang melakukan kerja praktek.

3.2. Penerapan Ilmu dalam Kerja Praktek

Beberapa pengetahuan perkuliahan yang diaplikasikan pada pelaksanaan kerja praktek sebagai berikut :

1. **Praktikum Aplikasi Komputer**

Penggunaan ilmu dalam Praktikum Aplikasi Komputer yaitu bagaimana menggunakan program microsoft word yang digunakan penulis untuk pembuatan dokumentasi dan jurnal dari yang dikerjakan oleh penulis.

2. **Praktikum Aplikasi Mobile**

Penggunaan ilmu dalam Praktikum Aplikasi Mobile dimana penulis mengerjakan latihan awal yang diberikan oleh pembimbing yaitu penggunaan *activity* dan *intent* dengan menggunakan android studio, dan pada projek mobile aumi revi menggunakan android studio untuk melihat log.

3. **Pemrograman Berorientasi Objek**

Penggunaan ilmu dalam Pemrograman Berorientasi Objek dimana penulis menggunakan bahasa pemrograman java yang digunakan pada

penggunaan *rest web service* menggunakan eclipse seperti yang dipelajari di mata kuliah ini.

4. Sistem Operasi

Penggunaan ilmu dalam Sistem Operasi yaitu ketika penulis menggunakan windows untuk mengatur environment variables yang dibutuhkan, dan menggunakan command prompt, serta program-program lain yang ada di windows.

5. Pemrograman Web

Penggunaan ilmu dalam Pemrograman Web yaitu penulis mengerjakan projek mobile aumi revi menggunakan ionic framework. Karena ionic framework ini merupakan framework yang digunakan untuk mobile hybrid yang berbasis HTML5, maka penggunaannya hampir sama dengan pemrograman web, seperti ada bagian html untuk apa yang ditampilkan di layar, scss seperti css pada pemrograman web yaitu untuk memodifikasi tampilan.

BAB IV

KESIMPULAN

4.1. Kesimpulan

Pelaksanaan Kerja Praktek telah penulis jalani selama 30 hari kerja. Selama menjalani Kerja Praktek di PT Multipolar Technology penulis telah mencapai tujuan Kerja Praktek yang sudah diuraikan dengan menunjukkan perilaku yang professional di dunia kerja, maupun menganalisis sebuah masalah dan menemukan solusi terbaik dari masalah tersebut, dan mampu mempraktikkan ilmu yang dimiliki di dunia nyata, penulis juga dituntut untuk mampu beradaptasi dengan lingkungan kerja dan mencari pengetahuan tambahan saat menghadapi masalah yang belum pernah ditemui sebelumnya.

Dalam kerja praktek ini, penulis diberikan kesempatan untuk mengerjakan proyek mobile aumi revi. Dalam mengerjakan proyek tersebut penulis banyak belajar, dan mencari pengetahuan-pengetahuan baru. Kerja praktek ini juga membuat penulis lebih memahami tentang dunia kerja yaitu tentang kerja sama di dalam satu tim, kedisiplinan, dan tanggung jawab terhadap tugas yang diberikan. Penulis juga menyadari pentingnya ilmu yang di dapat selama perkuliahan sebagai referensi yang cukup membantu dalam dunia kerja.

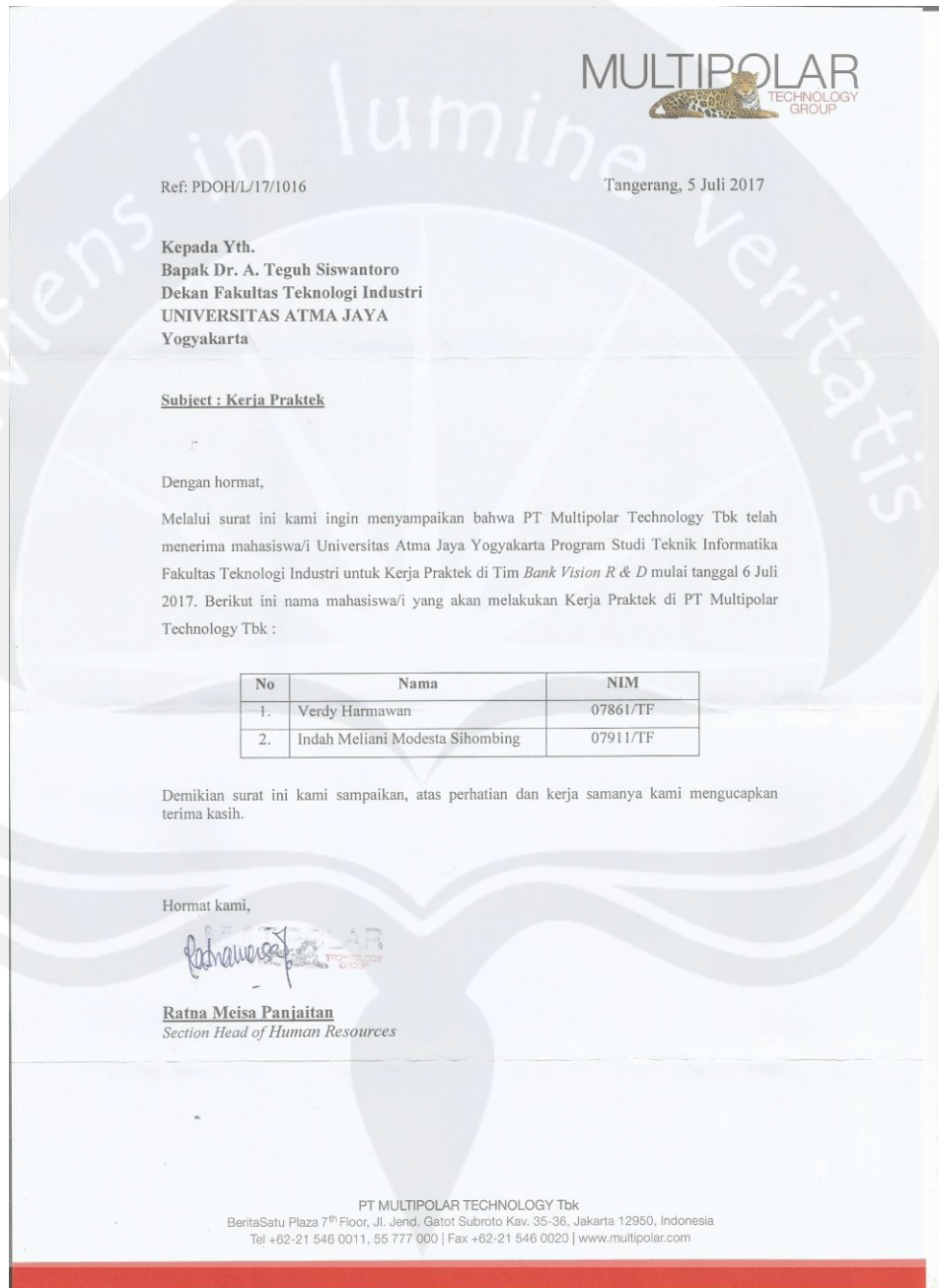
4.2. Saran

Beberapa saran yang dapat diambil dari kerja praktek ini adalah sebagai berikut :

1. Untuk perusahaan, pada proyek mobile aumi revi dapat dikembangkan lebih lanjut khususnya di bagian koneksi dengan database. Sehingga dapat menjadi lebih baik.
2. Untuk universitas, dapat memperbanyak kerja sama dengan perusahaan-perusahaan *IT* sehingga mahasiswa lebih mudah dalam mencari perusahaan untuk kerja praktek.
3. Untuk mahasiswa, dapat mengikuti kerja praktek dengan sebaik-baiknya agar mendapatkan ilmu yang banyak, dan menjaga nama baik universitas.

LAMPIRAN

1. Surat Keterangan dari Perusahaan



MULTIPOLAR
TECHNOLOGY
GROUP

Ref: PDOH/L/17/1016 Tangerang, 5 Juli 2017

Kepada Yth.
Bapak Dr. A. Teguh Siswanto
Dekan Fakultas Teknologi Industri
UNIVERSITAS ATMA JAYA
Yogyakarta

Subject : Kerja Praktek

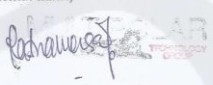
Dengan hormat,

Melalui surat ini kami ingin menyampaikan bahwa PT Multipolar Technology Tbk telah menerima mahasiswa/i Universitas Atma Jaya Yogyakarta Program Studi Teknik Informatika Fakultas Teknologi Industri untuk Kerja Praktek di Tim *Bank Vision R & D* mulai tanggal 6 Juli 2017. Berikut ini nama mahasiswa/i yang akan melakukan Kerja Praktek di PT Multipolar Technology Tbk :

| No | Nama | NIM |
|----|---------------------------------|----------|
| 1. | Verdy Harmawan | 07861/TF |
| 2. | Indah Meliani Modesta Sihombing | 07911/TF |

Demikian surat ini kami sampaikan, atas perhatian dan kerja samanya kami mengucapkan terima kasih.

Hormat kami,


Ratna Meisa Paniaitan
Section Head of Human Resources

PT MULTIPOLAR TECHNOLOGY Tbk
BeritaSatu Plaza 7th Floor, Jl. Jend. Gatot Subroto Kav. 35-36, Jakarta 12950, Indonesia
Tel +62-21 546 0011, 55 777 000 | Fax +62-21 546 0020 | www.multipolar.com

Gambar 4.1. Surat penerimaan kerja praktek

No. Ref: PDOH/L/17/1218

Tangerang, 18 Agustus 2017

Kepada Yth,
Bapak Dr. A. Teguh Siswanto
Dekan Fakultas Teknologi Industri
UNIVERSITAS ATMA JAYA
Yogyakarta

Subject : Penyelesaian Kerja Praktek

Dengan hormat,

Melalui surat ini kami ingin menyampaikan bahwa mahasiswa/i Universitas Atma Jaya Yogyakarta Program Studi Teknik Informatika Fakultas Teknologi Industri, telah menyelesaikan Kerja Praktek di Tim *Bank Vision R & D* sejak tanggal 6 Juli 2017 s/d 18 Agustus 2017. Berikut ini daftar nama mahasiswa/i yang telah menyelesaikan kerja praktek di PT Multipolar Technology Tbk :

| No | Nama | NIM |
|----|---------------------------------|----------|
| 1. | Verdy Harmawan | 07861/TF |
| 2. | Indah Meliani Modesta Sihombing | 07911/TF |

Demikian surat ini kami sampaikan, atas perhatian dan kerja samanya kami mengucapkan terima kasih.

Hormat kami,



Ratna Meisa Panjaitan
Section Head of Human Resources

Gambar 4.2. Surat bukti kerja praktek

2. Surat Penilaian dari Perusahaan


FORM PENILAIAN KERJA PRAKTEK

Nama Mahasiswa : Verdy Harmawan
NIM : 140707861
Tempat Pelaksanaan : PT. Multipolar Technology Tbk.
Waktu Pelaksanaan : 06-07-2017 s/d 18-08-2017

| NO | ASPEK YANG DINILAI | NILAI (0-100) |
|-------------|--|---------------|
| 1 | Kemampuan Teknis di Bidang IT | 98 |
| 2 | Kemampuan Bekerja Sama dalam Tim | 98 |
| 3 | Penempatan Diri dalam Lingkungan Kerja | 100 |
| 4 | Kedisiplinan | 100 |
| RATA-RATA : | | 99 |

Komentar :
Kejelian melihat keyword yang dibutuhkan untuk problem solving dan googling perlu ditingkatkan lagi. Ketekunan, motivasi, dan kedisiplinan harus tetap dipertahankan saat masuk ke dunia kerja nanti!

Nama Pembimbing : Hans Kristianto
Posisi/Jabatan : Application Developer (SIBVR) / Project Leader (Aum: Ravi)
No. Handphone : 0878 8912 8755
Alamat Email : hans.kristianto@multipolar.com

Tangerang, ... 18 Agustus 2017...
Pembimbing Lapangan,

(Hans Kristianto

Gambar 4.3. Surat penilaian kerja praktek