

LAPORAN KERJA PRAKTEK

Project Mobile AumiRevi

PT Multipolar Technology Tbk



Dipersiapkan oleh:

Indah Meliani Modesta Sihombing / 140707911

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

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

Pembimbing Lapangan

(Hans Kristianto)

Kata Pengantar

Puji syukur penulis panjatkan kepada kehadiran Tuhan Yang Maha Esa yang telah melimpahkan rahmat, kasih dan karuni-Nya 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 Tbk (MLPT) dari tanggal 06 Juli 2017 s/d 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 Orang tua serta segenap keluarga yang telah memberikan motivasi baik secara moril ataupun materil kepada saya, Ibu Stephanie Pamela A, S.T., M.T. selaku dosen pembimbing, Bapak Ngo Yoki Uray selaku Supervisor yang telah menerima saya untuk melakukan kerja praktek di PT. Multipolar Technology Tbk (MLPT) bagian Bank Vision R & D, Kak Hans Kristianto selaku pembimbing lapangan yang telah banyak memberikan arahan dan masukan kepada saya dalam melaksanakan Kerja Praktek, Kak Jonathan Tandarma Tjhin dan Kak Kelvin Martinez yang telah banyak membantu dalam pelaksanaan kerja praktek saya sehari-harinya. Tidak lupa juga penulis mengucapkan terima kasih kepada 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, 19 Agustus 2017

Penulis,



Indah Meliani Modesta Sihombing



Daftar Isi

Kata Pengantar	iii
Daftar Isi.....	v
Daftar Gambar.....	vii
BAB 1 PENDAHULUAN	1
1.1. Sekilas Perusahaan	1
1.2. Sejarah Perusahaan.....	3
1.3. Visi, Misi dan Filosofi Perusahaan.....	6
1.4. Struktur Oraganisasi	7
1.5. Deskripsi Tugas Dalam Struktur Organisasi	10
1.6. Department IT dalam Perusahaan	13
BAB 2 PELAKSANAAN KERJA PRAKTEK	16
2.1. Penjelasan Logbook	16
2.2. Hasil Pekerjaan Secara Umum	24
2.3. Bukti Hasil Pekerjaan	43
2.3.1. Bagian Eclipse Bo.....	43
2.3.2. Bagian Eclipse Service	47
2.3.3. Bagian Service Constants	49
2.3.4. Commit Project (Eclipse).....	50
2.3.5. Push Project (Eclipse).....	51
2.3.6. Pull Project (Eclipse)	52
2.3.7. Command Prompt Bagian Bo	53
2.3.8. Command Prompt Bagian Service.....	54
2.3.9. Command Prompt Ionic.....	55
2.3.10. MySQL Workbench.....	56
2.3.11. Advanced Rest Client Login.....	57
2.3.12. Advanced Rest Client Approval History List	58
2.3.13. Advanced Rest Client Approval History Detail	60
2.3.14. SorcerTree	62
2.3.15. Bitbucket.....	62
2.3.16. Project Ionic	64

2.3.17. Ionic (app.module.ts)	65
2.3.18. Ionic (config.xml)	65
2.3.19. Ionic (Index.html)	66
2.3.20. Tampilan Login (html & scss)	67
2.3.21. History List	68
2.3.22. History Detail.....	70
2.3.23. HTML seluruh tampilan ionic	71
2.3.24. SCSS diseluruh tampilan ionic	76
BAB 3 HASIL PEMBELAJARAN	78
3.1. Manfaat Kerja Praktek.....	78
3.2. Penerapan Ilmu dalam Kerja Praktek	79
BAB 4 Kesimpulan	81
Lampiran	83
1. Surat Keterangan Penerimaan dari Perusahaan	83
2. Surat Keterangan Penyelesaian dari Perusahaan	84
3. Surat Penilaian dari Perusahaan.....	85

Daftar Gambar

Gambar 1.1. Filosofi Perusahaan	6
Gambar 1.2. Struktur Organisasi (1)	7
Gambar 1.3. Struktur Organisasi (2)	8
Gambar 1.4. Struktur Organisasi (3)	9
Gambar 1.5. Diagram Sistem Kerja Aplikasi	25
Gambar 1.6. Halaman Login	26
Gambar 1.7. Halaman Main Menu (Request)	27
Gambar 1.8. Halaman Approval Request (1)	28
Gambar 1.9. Halaman NFC Authentication	29
Gambar 1.10. Halaman Fingerprint Authentication (1)	30
Gambar 1.11 Halaman Fingerprint Authentication (2)	31
Gambar 1.12. Halaman Approval Request(2)	32
Gambar 1.13. Halaman Approval Request(3)	33
Gambar 1.14. Halaman Approval Request(4)	34
Gambar 1.15. Halaman NFC Reject Authentication	35
Gambar 1.16. Halaman Fingerprint Reject Authentication(1)	36
Gambar 1.17. Halaman Fingerprint Reject Authentication(2)	37
Gambar 1.18. Halaman Approval to Reject(1)	38
Gambar 1.19. Halaman Approval to Reject(2)	39
Gambar 1.20. Halaman Approval Reject	40
Gambar 1.21. Halaman Main Menu (History)	41
Gambar 1.22. Halaman History Detail	42
Gambar 1.23. Eclipse Bo (1)	43
Gambar 1.24. Eclipse Bo (2)	43
Gambar 1.25. Eclipse Bo (3)	44
Gambar 1.26. Login.java	44
Gambar 1.27. GetApprovalList.java	45

Gambar 1.28. GetApprovalHistoryDetail.java	46
Gambar 1.29. LoginServiceController.java	47
Gambar 1.30. GetApprovalHistoryListService.java	48
Gambar 1.31. GetApprovalDetailService.java	48
Gambar 1.32. Service Constant (1)	49
Gambar 1.33. Service Constant (2)	49
Gambar 1.34. Commit Project (Eclipse)	50
Gambar 1.35. Push Project (Eclipse)	51
Gambar 1.36. Pull Project (Eclipse)	52
Gambar 1.37. Command Prompt Bo	53
Gambar 1.38. Command Prompt Service	54
Gambar 1.39. Command Prompt Ionic Test Browser	55
Gambar 1.40. Command Prompt Ionic Test Device	55
Gambar 1.41. MySQL Workbench	56
Gambar 1.42. Advanced Rest Client Login (1)	57
Gambar 1.43. Advanced Rest Client Login (2)	58
Gambar 1.44. Advanced Rest Client Approval History List (1)	58
Gambar 1.45. Advanced Rest Client Approval History List (2)	59
Gambar 1.46. Advanced Rest Client Approval History List (3)	60
Gambar 1.47. Advanced Rest Client Approval History Detail (1)	60
Gambar 1.48. Advanced Rest Client Approval History Detail (2)	61
Gambar 1.49. Source Tree	62
Gambar 1.50. Bitbucket (1)	62
Gambar 1.51. Bitbucket (2)	63
Gambar 1.52. Bitbucket (3)	63
Gambar 1.53. Project Ionic (1)	64
Gambar 1.54. Project Ionic (2)	64
Gambar 1.55. app.module.ts	65

Gambar 1.56. config.xml	65
Gambar 1.57. index.html	66
Gambar 1.58. login.html	67
Gambar 1.59. login.scss	67
Gambar 1.60. appreq.html	68
Gambar 1.61. appreq.scss	68
Gambar 1.62. appreq.ts	69
Gambar 1.63. apphist.html	70
Gambar 1.64. apphist.scss	70
Gambar 1.65. apphist.ts	71
Gambar 1.66. appreq1.html	72
Gambar 1.67. appreq2.html	72
Gambar 1.68. appreq3.html	73
Gambar 1.69. appreq4.html	73
Gambar 1.70. appreq5.html	74
Gambar 1.71. appreject.html	74
Gambar 1.72. appreject1.html	75
Gambar 1.73. appreject2.html	75
Gambar 1.74. appreject3.html	76
Gambar 1.75. scss ionic	77

BAB 1

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 TI (Teknologi Informasi) 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. Information Technology 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 (Information Technology) Strategy/IT (Information Technology) 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. 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. PT Multipolar memiliki perjalanan bisnis mulai dari awal berdiri hingga saat ini setiap tahunnya, yaitu;

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 (*Initial Public Offering*): 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 (International Organization for Standardization)* 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 Filosofi 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

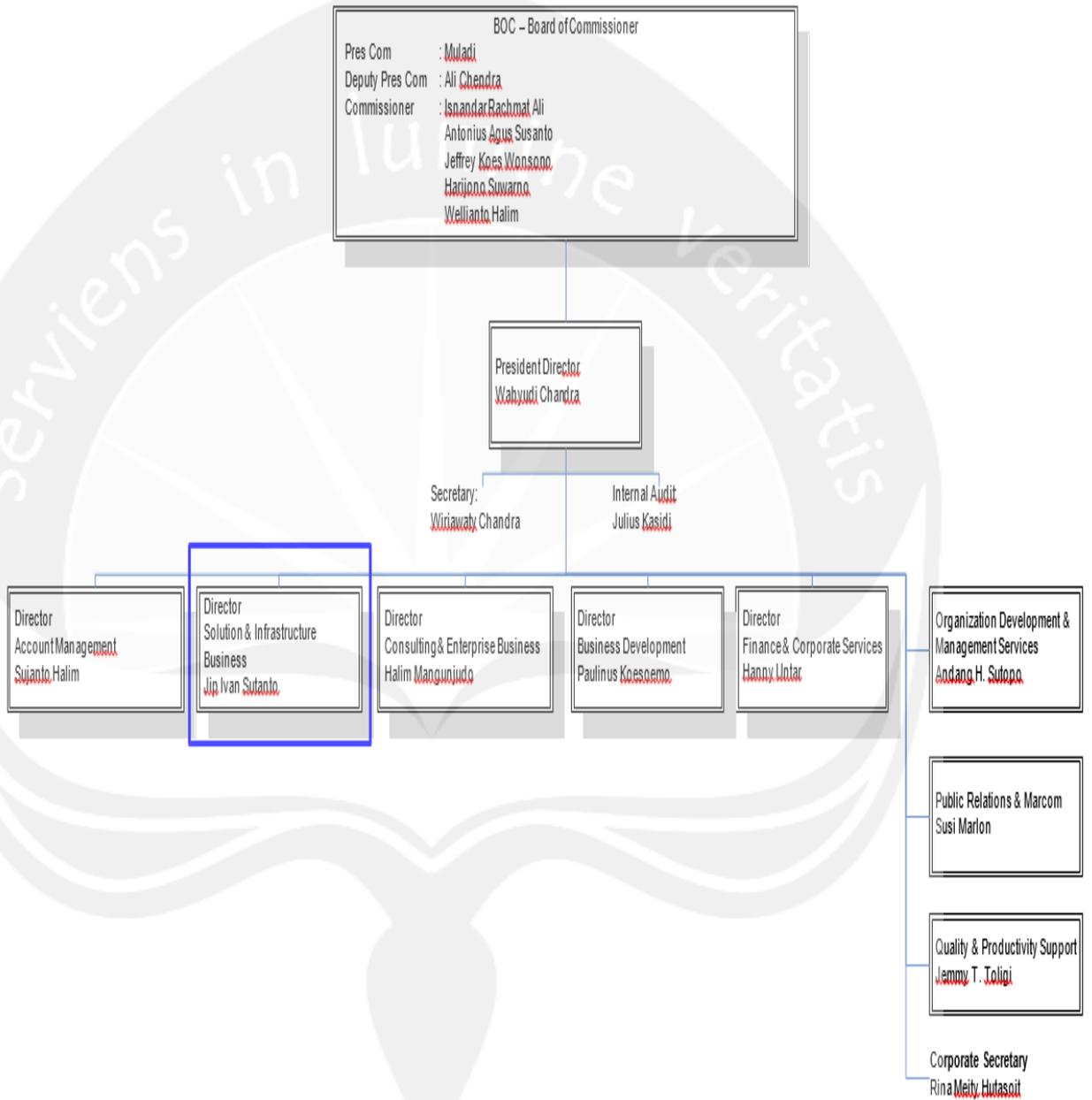
Karyawan PT Multipolar menjunjung kode etik profesional demi terciptanya keunggulan tata laksana dan operasional. Program pelatihan karyawan dan standar mutu internal memastikan tercapainya standar mutu yang dibutuhkan untuk mendorong kepuasan pelanggan. PT Multipolar menyediakan lingkungan yang memperhatikan keseimbangan beban kerja sehingga memungkinkan karyawan PT Multipolar menikmati keseimbangan dalam pekerjaan dan kehidupan. Ini semua tercermin dalam $M = IQ^2$, seperti pada gambar berikut :



Gambar 1.1. Filosofi Perusahaan

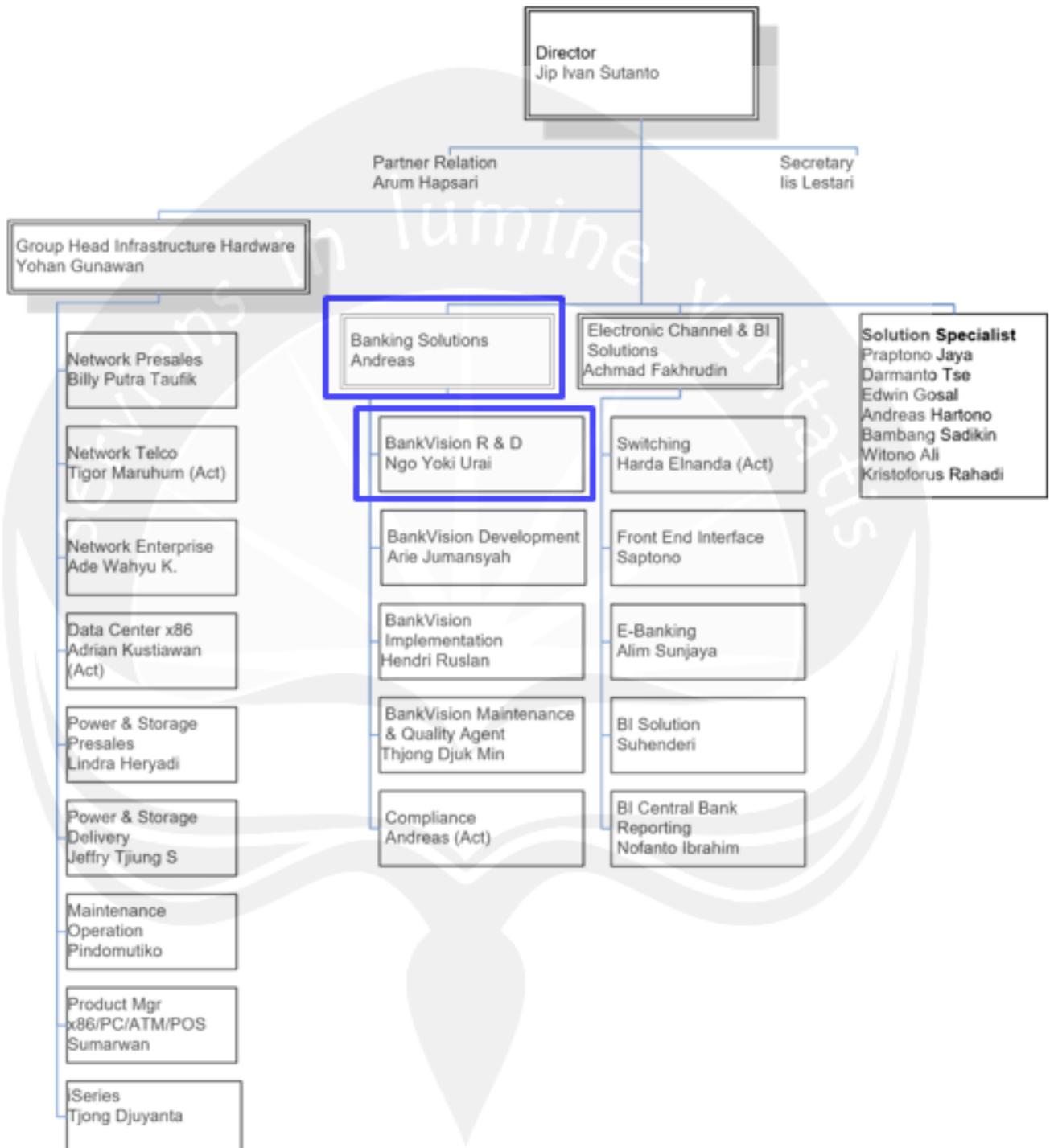
1.4. Struktur Organisasi

PT Multipolar Technology Tbk.

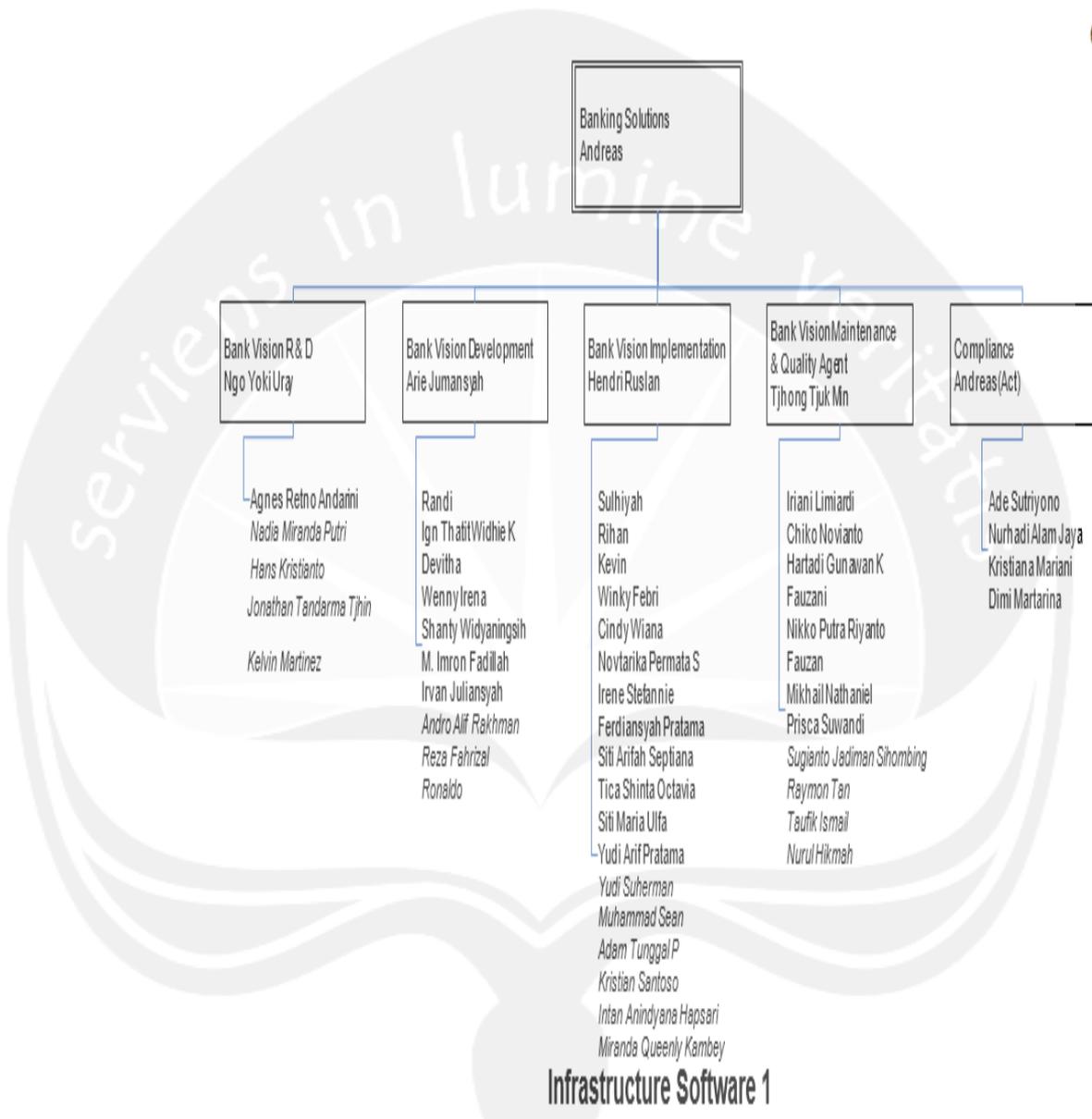


Gambar 1.2. Struktur Organisasi (1)

Solution & Infrastructure Business



Gambar 1.3. Struktur Organisasi (2)



Gambar 1.4. Struktur Organisasi (3)

1.5. Deskripsi Tugas Dalam Struktur Organisasi

1. BOC-Board of Commissioner:

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

2. Presiden Director

Pimpinan tertinggi dalam suatu perusahaan. Pimpinan tertinggi ini memiliki tanggung jawab dalam memimpin dan mengarahkan perusahaan, yaitu dalam hal:

- Menyusun strategi dan visi
- Menjalin hubungan dan kemitraan strategis
- Mengatur investasi, alokasi dan divestasi
- Memimpin direksi
- Memastikan bahwa prinsip tata kelola perusahaan benar-benar diterapkan dengan baik
- Membuat rencana pengembangan perusahaan dan usaha perusahaan dalam jangka pendek dan jangka panjang
- Bertanggung jawab penuh dalam melaksanakan tugasnya untuk kepentingan perseroan dalam mencapai maksud dan tujuannya
- Bertanggung jawab terhadap kemajuan perusahaan dan mengkoordinasikan seluruh kegiatan perusahaan
- Menerima dan memeriksa laporan dari masing-masing bagian

3. Secretary

Bertugas sebagai sekretaris dari President Director yang bersangkutan dan mengatur jadwal pertemuan dan janji dari President Director.

4. Internal Audit

Bertugas melakukan auditing perusahaan secara internal, melihat permintaan persetujuan proyek, mereview rencana proyek, dan melacak perkembangan dari sebuah proyek dan memberikan laporan hasil auditing kepada President Director.

5. Director Account Management

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

6. Director Solution & Infrastructure

a. Partner Relation

Bertugas dalam menjaga dan menjalin hubungan baik dengan semua partner atau mitra kerja dari perusahaan

b. Secretary

Bertugas sebagai sekretaris dari *Director Solution & Infrastructure* yang bersangkutan dan mengatur jadwal pertemuan dan janji dari *Director Solution & Infrastructure*.

c. Group Head Infrastructure Hardware

Bertugas dalam menangani permasalahan dan pengembangan infrastruktur dari *hardware*. Dan mengawasi divisi di bawahnya; *Network Presales, Network Telco, Network Enterprise, Data Center x86, Power & Storage Presales, Power & Storage Delivery, Maintenance Operation, Product Mgr x86/PC/ATM/POS, iSeries*).

d. Banking Solutions

Memiliki tugas – tugas seperti bertanggung jawab mengawasi dan memimpin semua sub divisi di bawahnya kemudian menerima dan menganalisi laporan setiap sub divisi di bawahnya. Bertanggung jawab dan mengatur segala sesuatunya di dalam penyediaan solusi sistem perbankan dan bertanggung jawab melaporkan hasil kerja divisi kepada *Director Solution & Infrastructure*.

- Bank Vision R & D

Bertugas dalam melakukan riset mengenai teknologi terbaru dan mengembangkan aplikasi dan teknologi yang ada dengan teknologi terbaru.

- **Bank Vision Development**

Bertugas dalam mengembangkan produk sesuai dengan permintaan dari *customer* (*customer request*), dan juga bertugas dalam melakukan *fixing bugs* dalam produk yang dibutuhkan *customer*.

- **Bank Vision Implementation**

Bertugas dalam pengimplementasian produk yang telah dibuat oleh Bank Vision pada *customer* baru.

- **Bank Vision Maintenance & Quality Agent**

Bertugas dalam maintenance produk dari Bank Vision yang digunakan oleh *customer* yang sudah ada (*customer* lama).

- **Compliance**

Bertugas sebagai penghubung dari Bank Vision dengan *customer*, seperti menerima *complaint* dari *customer* dan kemudian memberitahu ke bagian *maintenance*, jika kesalahan terjadi pada program, *maintenance* memberitahu ke bagian *development* untuk memperbaiki.

- e. **Electronic Channel & BI Solutions**

Bertugas dalam menangani permasalahan dan pengembangan pada perbankan seperti; *ATM (Automatic Teller Machine)*, *Internet Banking*, *Switching*, *Front End Interface*, *BI Solution*, dan *BI Central Bank Reporting*.

- f. **Solution Specialist**

Bertugas dalam menangani permasalahan dan pengembangan produk khusus di luar produk perbankan seperti; *ERP (Enterprise Resource Planning)* dan *CRM (Customer Relationship Management)*.

7. **Director Consulting & Enterprise Business**

Bertugas dalam menentukan perencanaan dan *business strategy* untuk *me-leverage* kemampuan sumber daya perusahaan dalam mewujudkan

competitive advantage guna pemenangan kompetisi dan pertumbuhan jangka panjang portofolio bisnis segmen *corporate (enterprise dan business)* pada lingkup perusahaan. Menerapkan rekomendasi dan memastikan *customer* menerima bantuan yang diperlukan, berhubungan dengan *customer* untuk selalu memberi informasi dari kemajuan dan untuk membuat keputusan yang relevan.

8. Director Business Development
Bertugas dan bertanggung jawab dalam mengembangkan bisnis yang ada pada perusahaan.
9. Director Finance & Corporate Services
Bertugas dalam menangani pengelolaan dana, pelaporan finansial dan perpajakan yang sesuai peraturan pemerintahan, legalitas perusahaan, dan layanan – layanan yang disediakan oleh perusahaan terhadap pemegang saham.
10. Organization Development & Management Services
Bertugas memimpin dan mengatur seluruh kegiatan yang berkaitan dengan layanan pengembangan dan pengelolaan perusahaan.
11. Public Relations & Marcom
Merencanakan dan mengembangkan komunikasi yang lebih bersifat korporat, membina hubungan baik dengan pihak eksternal perusahaan , sekaligus bertanggung jawab terhadap lancarnya komunikasi di tingkat internal perusahaan.
12. Quality & Productivity Support
Bertugas dalam menangani dan mengelola tingkat kualitas dan produktivitas yang dimiliki oleh perusahaan.
13. Corporate Secretary
Bertugas dalam menjembatani dewan direksi dengan para pemegang saham di dalam perusahaan.

1.6. Department IT dalam Perusahaan

Director Solution & Infrastructure

- a. Group Head Infrastructure Hardware

Bertugas dalam menangani permasalahan dan pengembangan infrastruktur dari *hardware*. Dan mengawasi divisi di bawahnya; *Network Presales, Network Telco, Network Enterprise, Data Center x86, Power & Storage Presales, Power & Storage Delivery, Maintenance Operation, Product Mgr x86/PC/ATM/POS, iSeries)*

b. Banking Solutions

Memiliki tugas – tugas seperti bertanggung jawab mengawasi dan memimpin semua sub divisi di bawahnya kemudian menerima dan menganalisis laporan setiap sub divisi di bawahnya. Bertanggung jawab dan mengatur segala sesuatunya didalam penyediaan solusi sistem perbankan dan bertanggung jawab melaporkan hasil kerja divisi kepada *Director Solution & Infrastructure*.

- Bank Vision R & D

Bertugas dalam melakukan riset mengenai teknologi terbaru dan mengembangkan aplikasi dan teknologi yang ada dengan teknologi terbaru.

- Bank Vision Development

Bertugas dalam mengembangkan produk sesuai dengan permintaan dari *customer (customer request)*, dan juga bertugas dalam melakukan *fixing bugs* dalam produk yang dibutuhkan *customer*.

- Bank Vision Implementation

Bertugas dalam pengimplementasian produk yang telah dibuat oleh Bank Vision pada *customer* baru.

- Bank Vision Maintenance & Quality Agent

Bertugas dalam *maintenance* produk dari Bank Vision yang digunakan oleh *customer* yang sudah ada (*customer* lama).

- Compliance

Bertugas sebagai penghubung dari Bank Vision dengan *customer*, seperti menerima *complaint* dari *customer* dan kemudian memberitahu ke bagian *maintenance*, jika kesalahan terjadi pada program, *maintenance* memberitahu ke bagian *development* untuk memperbaiki.

c. Electronic Channel & BI Solutions

Bertugas dalam menangani permasalahan dan pengembangan pada perbankan seperti; *ATM (Automatic Teller Machine)*, *Internet Banking*, *Switching*, *Front End Interface*, *BI Solution*, dan *BI Central Bank Reporting*.

d. Solution Specialist

Bertugas dalam menangani permasalahan dan pengembangan produk khusus di luar produk perbankan seperti; *ERP (Enterprise Resource Planning)* dan *CRM (Customer Relationship Management)*.

BAB 2

PELAKSANAAN KERJA PRAKTEK

2.1. Penjelasan Logbook

Kamis 06 Juli 2017

Pada hari ini penulis datang ke PT Multipolar Technology Tbk untuk mengisi formulir dan menandatangani kontrak kerja praktek dengan pihak *HRD (Human Resources Development)* perusahaan. Di sini kita 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. Setelah seluruh formulir dan kontrak diurus, penulis diajak oleh *HRD* ke lantai 4 untuk bertemu dengan *Supervisor* yang menerima kita dan dengan kakak-kakak yang akan menjadi tim dan yang akan membantu kita dalam proses kerja praktek kedepannya. Selanjutnya, penulis diperkenalkan dengan beberapa karyawan yang ada lantai 4. Dan penulis diberikan tugas oleh ketua tim sekaligus pembimbing lapangan yaitu Kak Hans Kristianto untuk menginstall java dan android studio, untuk proses *install* java dan android studio diminta untuk dibuat dokumentasinya (langkah-langkah *install*). Setelah proses *install* selesai, penulis diminta untuk membuat *project basic* android studio seperti *Hello World* dan sebuah aplikasi *sign up*, untuk *project* ini diminta juga dibuat dokumentasi dalam bentuk word.

Jumat 7 Juli 2017

Pada hari ini penulis diberikan tugas oleh pembimbing lapangan untuk melakukan penginstallan ionic framework dan setiap proses instalasi diminta untuk didokumentasikan dalam word. Setelah proses *install* selesai, penulis diberi tugas berikutnya yaitu membuat *project basic* yaitu aplikasi *hello world* dengan menggunakan ionic framework dan untuk pembuatan *project* tersebut juga didokumentasikan kedalam word.

Senin 10 Juli 2017

Pada hari ini pembimbing lapangan memberikan tugas untuk melakukan *setting* ionic framework, disini kita mencoba untuk mencocokkan java dengan cordova yang ada pada ionic, setelah proses *setting* selesai aplikasi

hello world yang telah dibuat diminta untuk di *deploy* pada *web browser* dan juga *device android*, proses *setting* dan *deploy* ini dimulai pada jam 08.30-12.00, diselingin dengan jam makan siang, setelah makan siang dilanjutkan jam 13.00-16.20. Pada pukul 16.30 penulis diberi tugas untuk membuat *mockup* untuk *login* dan juga menu dimana pada saat *login* berhasil halaman berpindah ke halaman menu dan aplikasi *login* dibuat menggunakan ionic framework.

Selasa 11 Juli 2017

Pada hari ini pembimbing lapangan memberikan tugas untuk mencoba membuat aplikasi dengan menggunakan *Ionic NFC (Near Field Communication) Reader, Ionic Barcode Scanner, OCR (Optical Character Recognition), dan Fingerprint*. Namun pada hari ini penulis hanya berhasil mengimplementasikan aplikasi *barcode scanner* pada *device*, dan aplikasi yang lain masih mengalami *error*.

Rabu 12 Juli 2017

Pada hari ini masih melanjutkan mencoba mengimplementasikan *ionic fingerprint*, namun hingga siang untuk *fingerprint* masih mengalami *error*. Dan setelah jam makan siang, pembimbing lapangan memberikan *mockup* sebuah aplikasi yang menjadi *project* yang akan kita kerjakan selama kerja praktek, dimana aplikasi ini digunakan pada bank untuk tindakan *approve* penyetoran uang oleh nasabah, dimana *teller* memiliki limit nominal uang untuk melakukan penyetoran dalam sehari, sehingga harus meminta *approve* dari *supervisor*, disinilah aplikasi ini digunakan. Untuk hari ini penulis membuat tampilan *login* dengan menggunakan ionic dan jquery agar tampilannya lebih rapi, kemudian membuat agar ketika *login* halamannya berpindah ke halaman *approval* dimana halaman *approval* berisi sebuah *navbar (navigation bar)* untuk *request* dan *history*, dan berikutnya penulis melakukan perbaikan untuk *navbar* dibagian *approval* tersebut.

Kamis 13 Juli 2017

Pada hari ini penulis melakukan perbaikan tampilan pada bagian *approval* yang belum rapi dan membuat tampilan *approval request*. Kemudian mencari cara bagaimana melakukan *push* dan *clone project* yang telah dibuat

ke bitbucket agar *project* tersebut dapat di *update* dan dapat saling berbagi tanpa harus menggunakan *flashdisk*. Selanjutnya penulis diminta untuk merapikan seluruh *button* yang ada di aplikasi tersebut.

Jumat 14 Juli 2017

Pada hari ini pembimbing lapangan memberi tugas agar seluruh *button* yang ada di tampilan *mobile* diberi warna. Selanjutnya saya membantu *partner* saya untuk mencari bagaimana cara mengunci tombol *back* pada *device* agar tombol *back* yang berfungsi hanya yang ada pada tampilan, selanjutnya mencari bagaimana cara menampilkan kembali konten yang ada pada halaman tersebut pada saat kita masuk kehalaman berikutnya dan kembali lagi kehalaman tersebut.

Senin 17 Juli 2017

Pada hari ini pembimbing lapangan memberi tugas untuk merapikan *layout* yang masih kurang rapi, dan melanjutkan kembali mencoba *ionic NFC* (*Near Field Communication*) yang masih mengalami *error*.

Selasa 18 Juli 2017

Pada hari ini pembimbing lapangan memberikan tugas yang sama seperti hari sebelumnya untuk melanjutkan kembali mencoba *ionic NFC* (*Near Field Communication*) yang masih mengalami *error*.

Rabu 19 Juli 2017

Pada hari ini pembimbing lapangan memberikan tugas yang sama seperti hari sebelumnya untuk melanjutkan kembali mencoba *ionic NFC* (*Near Field Communication*) yang masih mengalami *error*. Dan pada siang hari sekitar pukul 14.25 pembimbing lapangan memberikan tugas untuk mempelajari cara *restful service*. *REST* (*REpresentational State Transfer*) merupakan standar arsitektur komunikasi berbasis web yang sering diterapkan dalam pengembangan layanan berbasis web. Umumnya menggunakan *HTTP* (*Hypertext Transfer Protocol*) sebagai *protocol* untuk komunikasi data.

Kamis 20 Juli 2017

Pada hari ini pembimbing lapangan memberikan tugas yang sama seperti hari sebelumnya untuk mempelajari cara *restful service java*. Pada pukul

13.00-14.00 kita diundang untuk menghadiri *sharing* yaitu *sharing WELA*, di sini dijelaskan *project* yang sedang kita kerjakan secara detail. Setelah *sharing* selesai, pembimbing lapangan memberi tugas untuk melakukan *setting* beberapa *software* seperti; MySQL sebagai *server*, Advanced REST Client untuk melakukan *testing* data menggunakan *Uniform Resource Locator (URL)* yang telah dibuat di *service*, Java, Eclipse untuk *bo* dan *service*, dimana *bo* sebagai *logic* dan *service* sebagai *controller* nya, *maven (library)* dan *git* untuk menyambungkan ke bitbucket. Dan proses *setting* ini saya didamping dan dibimbing oleh Kak Jonathan Tandarma Tjhin.

Jumat 21 Juli 2017

Pada hari ini pembimbing lapangan memberikan tugas untuk membuat fungsi *login* java di eclipse bagian *bo login* dan mengetestnya pada *advanced REST client*.

Senin 24 Juli 2017

Pada hari ini pembimbing lapangan memberikan tugas untuk mengganti fungsi *login* java yang sebelumnya , menjadi fungsi *login* java yang bertipekan *boolean*. Setelah itu saya membantu *partner* saya mencari cari *rest login* menggunakan ionic, setelah selesai saya melanjutkan mengerjakan bagian *bo approval history list* dan *approval history list service* pada eclipse.

Selasa 25 Juli 2017

Pada hari ini pembimbing lapangan memberikan tugas masih yang sama yaitu mengerjakan *bo approval history list* dan *approval history list service* pada eclipse karena masih mengalami *error* pada saat mengetest data pada ARC. Setelah selesai saya lanjut mengerjakan *bo approval history detail* dan *approval history detail service*. Berikutnya mencoba *rest* menggunakan ionic pada bagian *approval history list*, dimana data yang telah dimasukkan harus tampil di aplikasi bagian *approval history list (navbar history)*.

Rabu 26 Juli 2017

Karena hari sebelumnya *rest* menggunakan ionic pada bagian *approval history list*, dimana data yang telah dimasukkan harus tampil di aplikasi bagian *approval history list(navbar history)* masih mengalami *error*,

pembimbing lapangan memberikan tugas untuk memperbaiki *error* nya, selanjutnya memperbaiki seluruh userid yang terdapat di *eclipse bo*, dimana seluruh tipe data dan data userid harus sama di seluruh kelas.

Kamis 27 Juli 2017

Karena hari sebelumnya terjadi perubahan data pada userid, hari ini pembimbing lapangan memberikan tugas untuk mencoba kembali *rest* menggunakan ionic pada bagian *approval history list*. Selanjutnya, karena data berbentuk *list* dan memanjang kesamping maka saya mencari bagaimana cara pindah baris pada data. Berikutnya, mengerjakan *rest* menggunakan ionic pada bagian *approval history* dan mengerjakan *rest* menggunakan ionic pada bagian *submit approval*, dimana di dua bagian ini hanya menampilkan detail dari data yang dipilih di bagian *approval history list*.

Jumat 28 Juli 2017

Karena hari sebelumnya *rest* menggunakan ionic pada bagian *submit approval* masih mengalami *error*, pembimbing lapangan memberikan tugas untuk memperbaiki *error* nya, selanjutnya mencoba kembali *ionic NFC*. Sebelumnya untuk data yang digunakan masih satu data, pembimbing lapangan memberi tugas untuk menambahkan beberapa data di bagian *approval history list* dan *approval history detail* pada *eclipse*. Setelah seluruh data yang dibutuhkan ditambahkan, data tersebut dipanggil di tampilan ionic nya. Untuk melihat *history detail* di sini ditambahkan *event* klik pada *history list*, agar detail yang ditampilkan pada halaman *approval history detail* hanya detail yang dipilih atau diklik pada halaman *approval history list*. Pada pukul 16.00 sampai selesai, tim saya mengadakan *sprint review*, pada saat *sprint review* ini setiap orang harus menjelaskan apa saja yang telah dikerjakan dalam seminggu.

Senin 31 Juli 2017

Pada hari ini pembimbing lapangan memberikan tugas untuk mencoba kembali menampilkan data yang telah ditambahkan dengan mengklik salah satu data pada *approval history list*, dan kemudian pada *approval history detail*, detail dari data tersebut ditampilkan. Dan diminta ketika kembali lagi

kebagian *approval history list* data yang dipilih tersebut tidak hilang atau berantakan.

Selasa 01 Agustus 2017

Pada hari ini pembimbing lapangan memberikan tugas untuk memberikan warna dan merapikan tampilan aplikasi pada halaman *login* dalam bentuk *CSS (Cascading Style Sheets)* dan *ionic*. Selanjutnya memberikan warna dan merapikan tampilan aplikasi pada bagian *approval* dalam bentuk *css* dan *ionic*.

Rabu 02 Agustus 2017

Pada hari ini pembimbing lapangan memberikan tugas untuk memberikan warna dan merapikan seluruh tampilan aplikasi dalam bentuk *css* dan *ionic*, tetapi disini bagian *header* untuk *icon logout* dan judul belum sejajar dan *active navbar* belum berwarna merah.

Kamis 03 Agustus 2017

Pada hari ini pembimbing lapangan memberikan tugas untuk memberikan *alert* pada setiap *button cancel* yang ada pada aplikasi. Berikutnya menyelesaikan *active navigation bar* ketika aktif menjadi berwarna merah untuk garis bagian bawahnya, selanjutnya saya mencari bagaimana cara agar di *header icon logout* dan judul menjadi sejajar. Kemudian *icon* yang terdapat *header* diganti dengan *image*, dan menyelesaikan bagian *header* dimana *image* dan judul menjadi sejajar pada seluruh *header* tampilan pada aplikasi.

Jumat 04 Agustus 2017

Pada hari ini pembimbing lapangan memberikan tugas untuk merapikan seluruh tampilan *ionic*, selanjutnya mencoba kembali *ionic fingerprint* untuk aplikasi tersebut. Pada pukul 16.00 sampai selesai tim saya mengadakan *sprint review*, pada saat *sprint review* ini setiap orang harus menjelaskan apa saja yang telah dikerjakan dalam seminggu.

Selasa 08 Agustus 2017

Pada hari ini pembimbing lapangan memberikan tugas untuk merapikan seluruh tampilan *ionic*, seperti membagi halamannya menggunakan *colom ionic* agar diseluruh *device* ketika di *deploy* tidak berantakan, disini

penulis didampingi dan dibimbing oleh kak Kelvin Martinez. Berikutnya pada pukul 16.00 sampai selesai saya mengikuti *sharing session* yang dibawakan oleh *partner* saya tentang *basic ionic*, navigasi halaman, dan menampilkan data yang diinputkan di halaman sebelumnya (halaman inputan) ke halaman *show data*.

Rabu 09 Agustus 2017

Pada hari ini pembimbing lapangan memberikan tugas untuk merapikan seluruh tampilan ionic, pada bagian *header* diseluruh tampilan, *style* judul seperti warna, ukuran, jenis dan posisinya serta memperbaiki kembali *button logout*, berikutnya merapikan *style* dari tulisan isi *approval request detail*, *approval request*, *approval history detail* dan *submit app*, seperti warna, ukuran, jenis dan posisinya. Selanjutnya sekitar pukul 16.00 sampai selesai saya membawakan *sharing session* tentang validasi ketika pindah halaman dan membuat tampilan lebih menarik dan rapi menggunakan *css* (memberi warna *background*, warna tulisan, jenis tulisan, memperbaiki letak konten, dan juga memberi warna *button*).

Kamis 10 Agustus 2017

Pada hari ini pembimbing lapangan memberikan tugas untuk mengganti tombol *logout* yang sebelumnya menggunakan *image* menjadi sebuah tombol yang didalamnya terdapat *icon* pada seluruh halaman tampilan. Berikutnya saya mencari bagaimana cara agar *list* yang ditampilkan sebaris panjang menjadi beberapa baris atau setiap *key* ditampilkan satu baris (*listnya* dienter). Selanjutnya sekitar pukul 16.00 sampai selesai saya mengikuti *sharing session* yang dibawakan oleh *partner* saya tentang pembuatan menu pada ionic.

Jumat 11 Agustus 2017

Pada hari ini pembimbing lapangan memberikan tugas yang sama seperti hari sebelumnya yaitu mencari bagaimana cara agar *list* yang ditampilkan sebaris panjang menjadi beberapa baris atau setiap *key* ditampilkan satu baris (*listnya* dienter). Setelah menemukan cara nya, hal tersebut diimplementasikan pada halaman *approval request list* dan halaman *approval history list*. Karena data yang ditampilkan masih data yang sembarangan, maka

saya mengganti data yang terdapat di eclipse, dan terakhir merapikan tampilan yang kurang rapi dan melakukan *clone project* ke bitbucket. Selanjutnya sekitar pukul 16.00 sampai selesai saya membawakan *sharing session* tentang *barcode scanner*.

Senin 14 Agustus 2017

Pada hari ini pembimbing lapangan memberikan tugas untuk mengecek kembali bagian *service constant* yang terdapat pada eclipse bagian *service* dan selanjutnya di *commit* dan di *push*.

Selasa 15 Agustus 2017

Pada hari ini pembimbing lapangan memberikan tugas untuk mengatur halaman *fingerprint* yaitu mengatur ukuran tulisan judul pada *header*, kemudian mengganti dan mengatur posisi tulisan perintah pada bagian gambar *fingerprint*.

Rabu 16 Agustus 2017

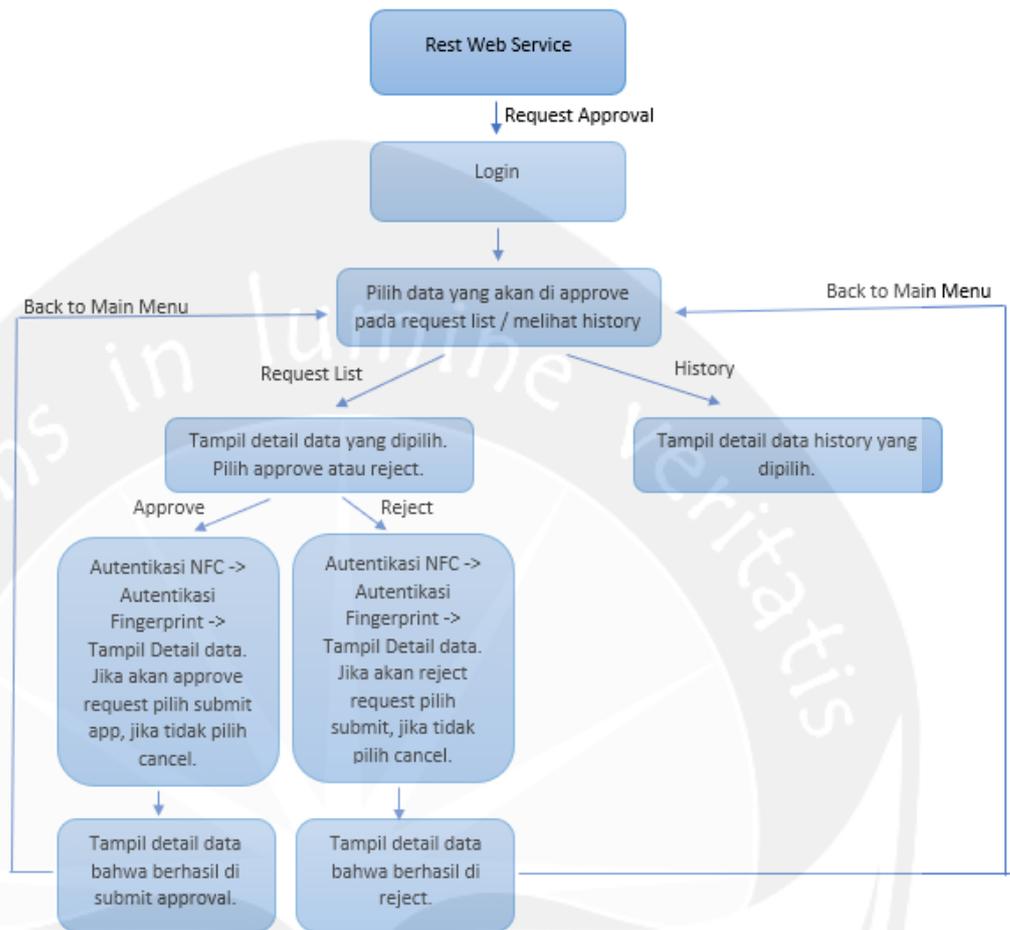
Pada hari ini penulis mengikuti rapat akhir untuk *project* yang dikerjakan bersama tim dan *supervisor*, pada rapat ini dibahas bagaimana *project* tersebut dijalankan (*flow*) serta apa yang harus diperbaiki untuk *finishing*. Selanjutnya penulis mengatur garis yang terdapat pada *approval request list* dan *approval history list*, serta membantu *partner* saya mencari bagaimana cara *lock orientation*.

Jumat 18 Agustus 2017

Pada hari ini pembimbing lapangan memberikan tugas untuk membuat dokumentasi dari *project* yang dikerjakan.

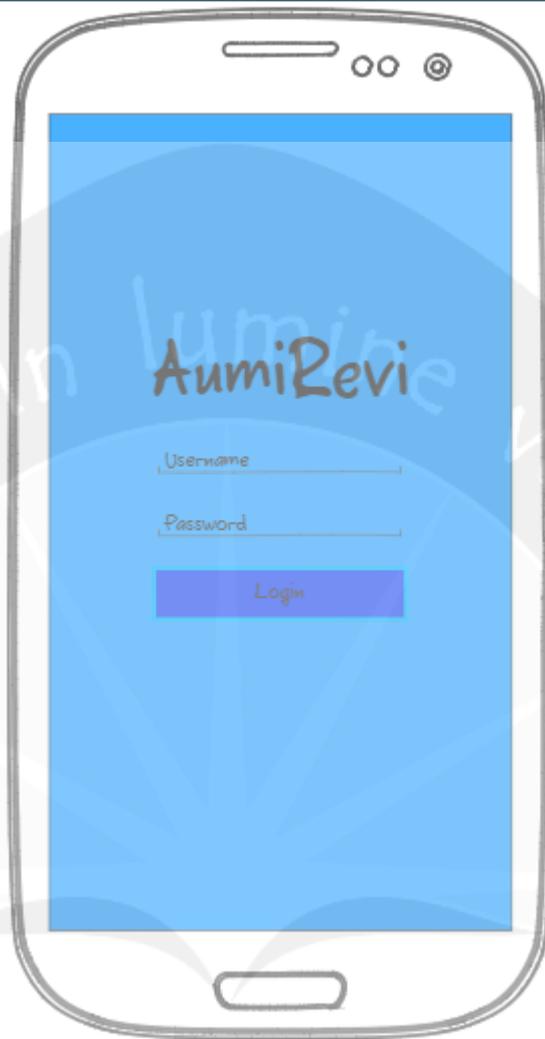
2.2. Hasil Pekerjaan Secara Umum

Selama satu bulan penulis melaksanakan kerja praktek, berbagai hal dan pengalaman telah penulis peroleh ataupun selesai penulis kerjakan. Disini penulis mengerjakan sebuah aplikasi *mobile* sebagai *project* untuk kerja praktek. Aplikasi ini diberi nama AumiRevi, dimana AumiRevi ini kedepannya akan digunakan oleh bank tepatnya pada bagian *supervisor* di bank tersebut. Untuk proses pembuatan aplikasi ini dibutuhkan beberapa *tools* dan *software* seperti Java jdk 1.7 untuk pembuatan *project* dan Java jdk 1.8 untuk *deploy* ke *device*, android studio digunakan untuk melihat *log* pada saat aplikasi dijalankan, ionic framework sebagai *tool* untuk pembuatan aplikasi (dapat dilihat pada gambar 1.53 sampai gambar 1.75), MySql sebagai *server* (dapat dilihat pada gambar 1.41), Advanced REST Client untuk melakukan *testing* data menggunakan *Uniform Resource Locator (URL)* yang telah dibuat di *service* (dapat dilihat pada gambar 1.42 sampai gambar 1.48), Eclipse untuk *bo* dan *service* (dapat dilihat pada gambar 1.23 sampai 1.36), dimana *bo* sebagai *logic* dan *service* sebagai *controller* nya, maven (*library*) dan git untuk menyambungkan ke bitbucket. Source Tree untuk melakukan *update code* yang dikerjakan pada ionic yang tersambung pada bitbucket (dapat dilihat pada gambar 1.49). Command prompt untuk mengaktifkan *bo*, *service* dan juga *project ionic* (dapat dilihat pada gambar 1.37 sampai gambar 1.40). Dan bitbucket sebagai *tool* untuk *share project* secara *online* dengan tim (dapat dilihat pada gambar 1.50 sampai gambar 1.52). Pada aplikasi ini penulis banyak bekerja pada bagian *FrontEnd/Layout*. Untuk sistem kerja dari aplikasi ini dapat dilihat pada diagram gambar 1.5 dan *mockup* aplikasi (dapat dilihat pada gambar 1.6 sampai gambar 1.22) beserta penjelasannya sebagai berikut;



Gambar 1.5. Diagram Sitem Kerja Aplikasi

1. Seorang *teller* pada sebuah bank memiliki limit jumlah uang untuk proses transaksi dalam sehari. Misalnya untuk transaksi penyimpanan uang oleh nasabah sebesar Rp 1.000.000.000, tetapi *teller* hanya boleh melakukan transaksi penyimpanan sebesar Rp 500.000.000, untuk dapat melanjutkan proses transaksi tersebut *teller* harus meminta *approve* dari *supervisor*, kemudian *teller* mengirim *request* kepada *supervisor*.
2. Disinilah aplikasi tersebut dijalankan, *supervisor* akan *login* ke aplikasi AumiRevi dengan *username* dan *password* yang dimilikinya. Untuk *code* dari tampilan dapat dilihat pada gambar 1.57 dan 1.58 dan untuk tampilannya dapat dilihat pada gambar 1.6 sebagai berikut:



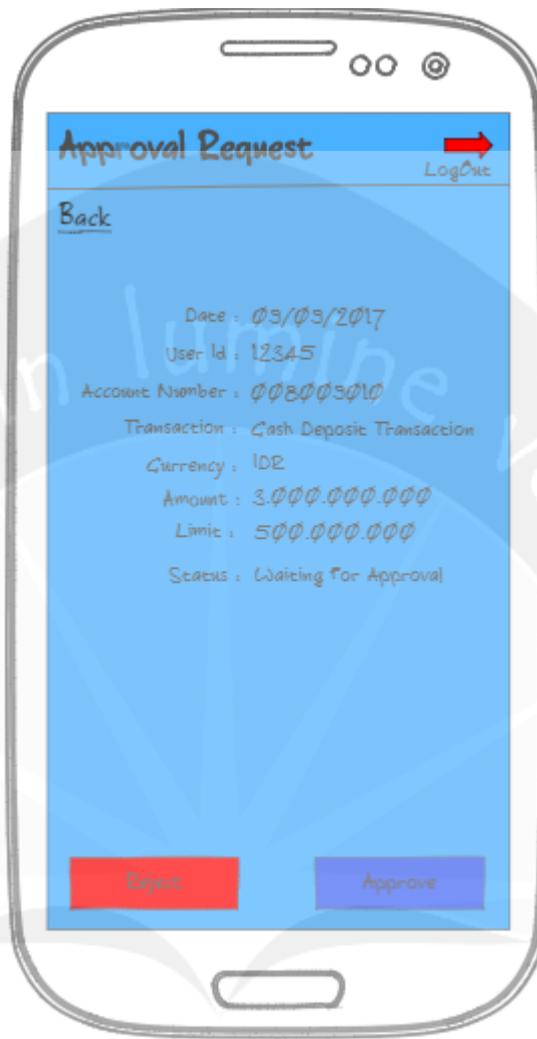
Gambar 1.6. Halaman Login

3. Aplikasi akan menampilkan *list* transaksi yang akan *diapprove* atau *direject* oleh *supervisor* tersebut. Tampilannya dapat dilihat pada gambar 1.7 sebagai berikut:



Gambar 1.7. Halaman Main Menu (Request)

4. *Supervisor* memilih transaksi mana yang akan diapprove, dan aplikasi akan menampilkan data detail dari *list* yang telah dipilih oleh *supervisor*. Untuk *code* dari tampilan dapat dilihat pada gambar 1.66 dan gambar 1.75 dan untuk tampilannya dapat dilihat pada gambar 1.8 sebagai berikut:



Gambar 1.8. Halaman Approval Request (1)

5. Selanjutnya *Supervisor* memilih apakah transaksi tersebut akan *direct* atau *approve*, jika *supervisor* memilih untuk melakukan *approve* maka selanjutnya aplikasi akan menampilkan proses selanjutnya yaitu *supervisor* diminta untuk melakukan *tag NFC* yang dimilikinya pada *smartphone* untuk proses *submit*. Untuk *code* dari tampilan dapat dilihat pada gambar 1.67 dan gambar 1.75 dan untuk tampilannya dapat dilihat pada gambar 1.9 sebagai berikut:



Gambar 1.9. Halaman NFC Authentication

6. Jika data NFC yang dimiliki oleh *supervisor* sama dengan *database* dan *NFC tag*nya berhasil aplikasi akan melanjutkan keproses berikutnya yaitu *supervisor* diminta untuk melakukan verifikasi dengan *fingerprint* untuk proses *submit*. Untuk *code* dari tampilan dapat dilihat pada gambar 1.68 dan gambar 1.75 dan untuk tampilannya dapat dilihat pada gambar 1.10 dan gambar 1.11 sebagai berikut:



Gambar 1.10. Halaman Fingerprint Authentication (1)



Gambar 1.11 Halaman Fingerprint Authentication(2)

7. Setelah verifikasi dengan *fingerprint* berhasil, aplikasi akan menampilkan detail data transaksinya kembali dengan status *in progress*. Dan *supervisor* diminta untuk memilih melanjutkan proses *approve* atau membatalkan proses tersebut, jika *supervisor* memilih untuk *cancel* maka *supervisor* cukup mengklik *button cancel* dan aplikasi akan memunculkan *message* untuk konfirmasi apakah proses tersebut benar-benar di *cancel*, jika *supervisor* memilih “No” aplikasi akan tetap menampilkan halaman tersebut, jika *supervisor* memilih “Yes” tampilannya kembali ke halaman *main menu*, dan jika *supervisor* memilih untuk melanjutkan proses *approve* maka *supervisor* cukup mengklik *button Submit App*. Untuk *code* dari

tampilan dapat dilihat pada gambar 1.69 dan 1.75 dan untuk tampilannya dapat dilihat pada gambar 1.12 dan gambar 1.13 sebagai berikut:

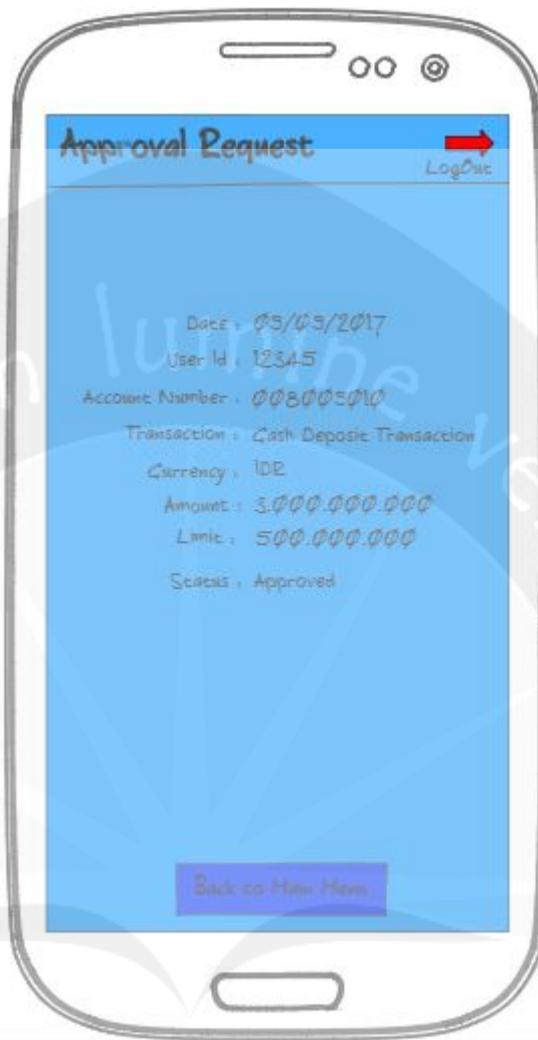


Gambar 1.12. Halaman Approval Request(2)



Gambar 1.13. Halaman Approval Request(3)

8. Setelah *supervisor* memilih untuk *Submit Approve*, aplikasi akan menampilkan detail data transaksinya kembali dengan status *approved*. Dan setelah selesai *supervisor* dapat mengklik *button "Back to Main Menu"*. Untuk *code* dari tampilan dapat dilihat pada gambar 1.70 dan 1.75 dan untuk tampilannya dapat dilihat pada gambar 1.14 sebagai berikut:



Gambar 1.14. Halaman Approval Request(4)

9. Pada proses no 5, *supervisor* memilih untuk melakukan *reject* maka selanjutnya aplikasi akan menampilkan proses selanjutnya yaitu *supervisor* diminta untuk melakukan *tag NFC* yang dimilikinya pada *smartphone* untuk proses *reject*. Untuk *code* dari tampilan dapat dilihat pada gambar 1.71 dan 1.58 dan untuk tampilannya dapat dilihat pada gambar 1.15 sebagai berikut:



Gambar 1.15. Halaman NFC Reject Authentication

10. Jika data *NFC* yang dimiliki oleh *supervisor* sama dengan *database* dan *NFC tag*nya berhasil aplikasi akan melanjutkan keproses berikutnya yaitu *supervisor* diminta untuk melakukan verifikasi dengan *fingerprint* untuk proses *reject*. Untuk *code* dari tampilan dapat dilihat pada gambar 1.72 dan 1.58 dan untuk tampilannya dapat dilihat pada gambar 1.16 dan gambar 1.17 sebagai berikut:



Gambar 1.16. Halaman Fingerprint Reject Authentication(1)



Gambar 1.17. Halaman Fingerprint Reject Authentication(2)

11. Setelah verifikasi dengan *fingerprint* berhasil, aplikasi akan menampilkan detail data transaksinya kembali dengan status *in progress*. Dan *supervisor* diminta untuk memilih melanjutkan proses *reject* atau membatalkan proses tersebut, jika *supervisor* memilih untuk *cancel* maka *supervisor* cukup mengklik *button cancel* dan aplikasi akan memunculkan *message* untuk konfirmasi apakah proses tersebut benar-benar di *cancel*, jika *supervisor* memilih “No” aplikasi akan tetap menampilkan halaman tersebut, jika *supervisor* memilih “Yes” tampilannya kembali ke halaman *main menu*, dan jika *supervisor* memilih untuk melanjutkan proses *reject* maka *supervisor* cukup mengklik *button submit reject*. Untuk *code* dari tampilan

dapat dilihat pada gambar 1.73 dan 1.58 dan untuk tampilannya dapat dilihat pada gambar 1.18 dan gambar 1.19 sebagai berikut:



Gambar 1.18. Halaman Approval to Reject (1)



Gambar 1.19. Halaman Approval to Reject (2)

12. Setelah *supervisor* memilih untuk *submit reject*, aplikasi akan menampilkan detail data transaksinya kembali dengan status *reject*. Dan setelah selesai *supervisor* dapat mengklik *button* “*Back to Main Menu*”. Untuk *code* dari tampilan dapat dilihat pada gambar 1.74 dan 1.58 dan untuk tampilannya dapat dilihat pada gambar 1.20 sebagai berikut:



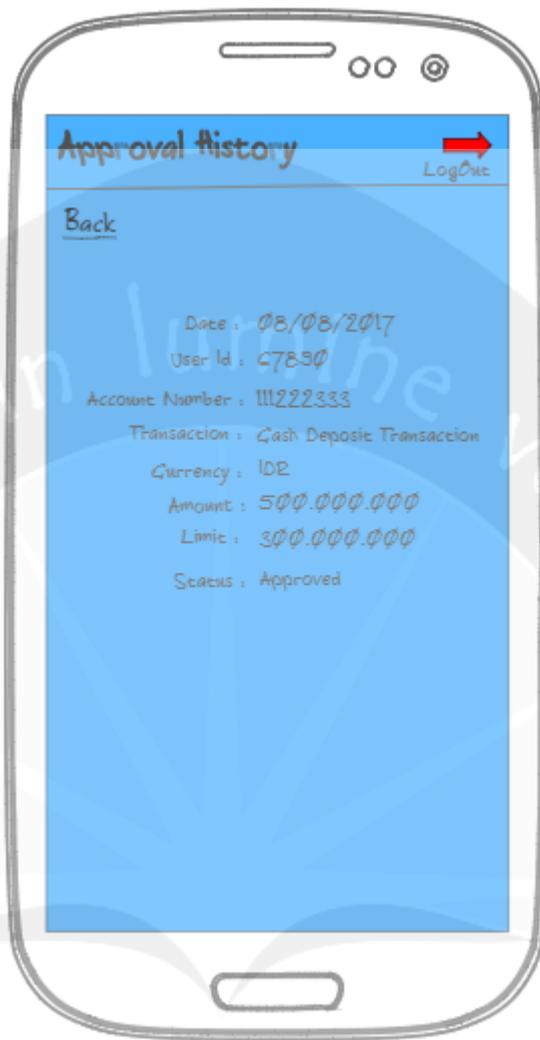
Gambar 1.20. Halaman Approval Reject

13. Jika *supervisor* ingin keluar dari aplikasi, *supervisor* cukup mengklik *icon logout* yang ada pada setiap halaman dibagian sebelah kanan atas. Dan jika pada setiap halaman *supervisor* ingin kembali kehalaman sebelumnya, *supervisor* cukup mengklik tulisan *back* dibagian kiri atas halaman.
14. *Supervisor* juga dapat melihat *history* dari proses *approve* dan *reject* yang telah dilakukan, dengan mengklik *tab history* pada *main menu*, aplikasi akan menampilkan *list* dari *history*. Untuk *code* dari tampilan dapat dilihat pada gambar 1.60, gambar 1.61 dan gambar 1.62 dan untuk tampilannya dapat dilihat pada gambar 1.21 sebagai berikut:



Gambar 1.21. Halaman Main Menu (History)

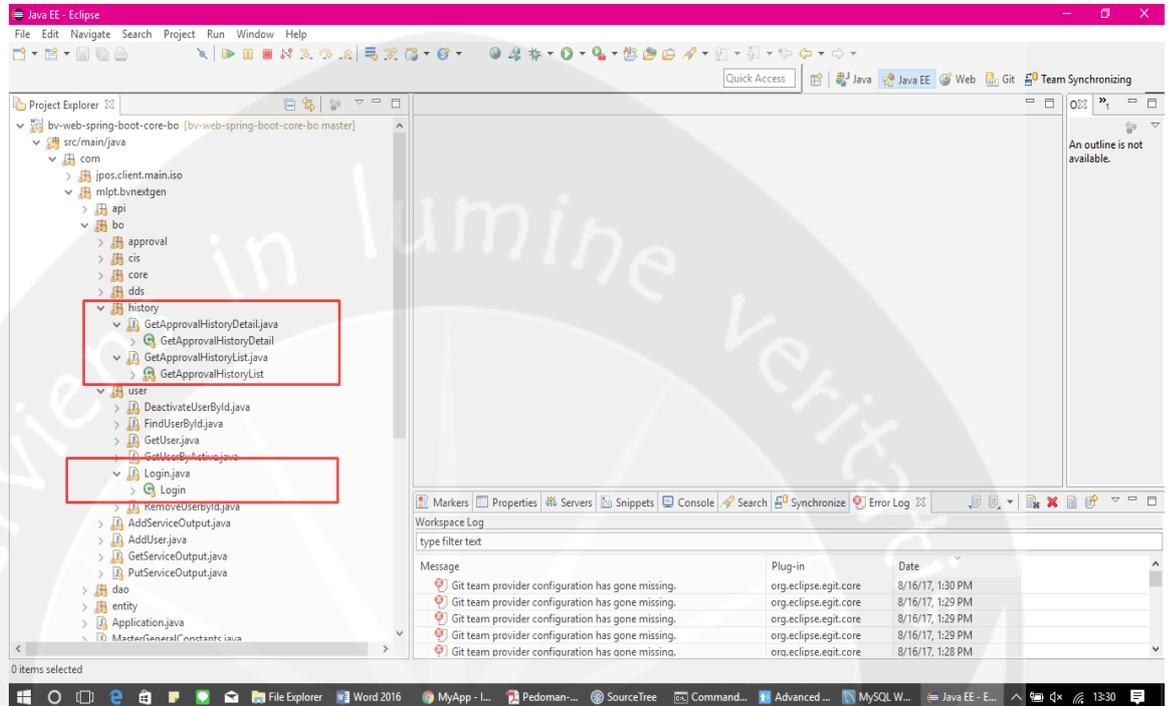
15. Untuk melihat detail dari *list history* tersebut, *supervisor* cukup mengklik salah satu *listnya* dan aplikasi akan menampilkan detailnya. Untuk *code* dari tampilan dapat dilihat pada gambar 1.63, gambar 1.64 dan gambar 1.65 dan untuk tampilannya dapat dilihat pada gambar 1.22 sebagai berikut:



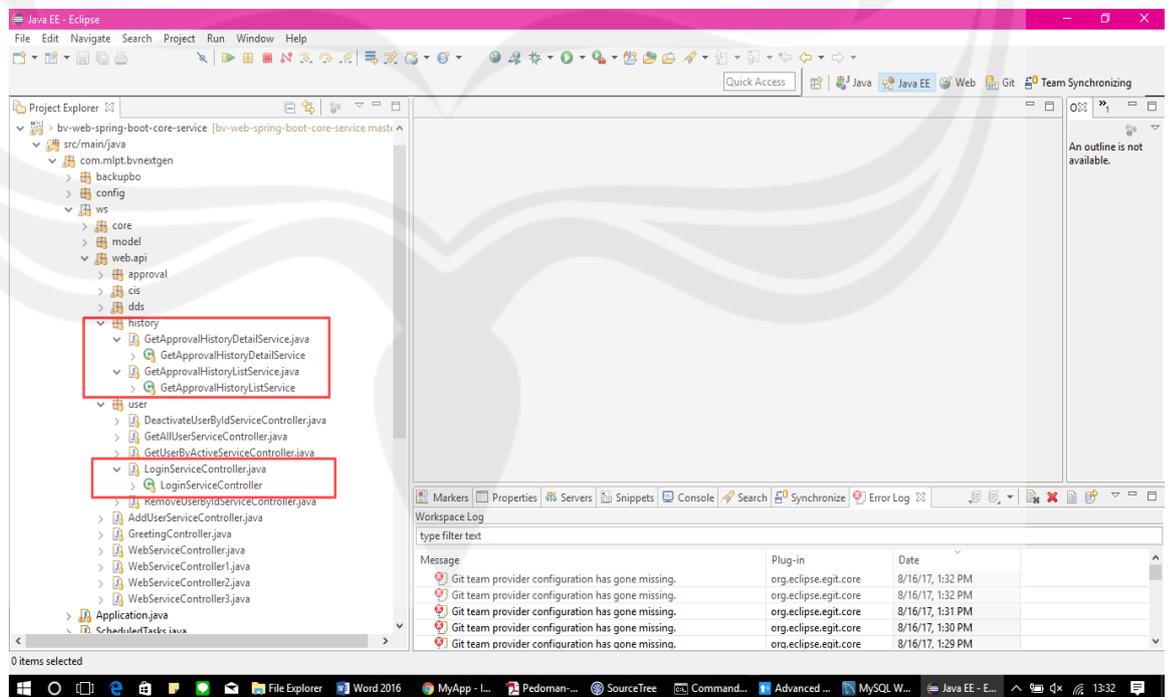
Gambar 1.22. Halaman History Detail

2.3. Bukti Hasil Pekerjaan

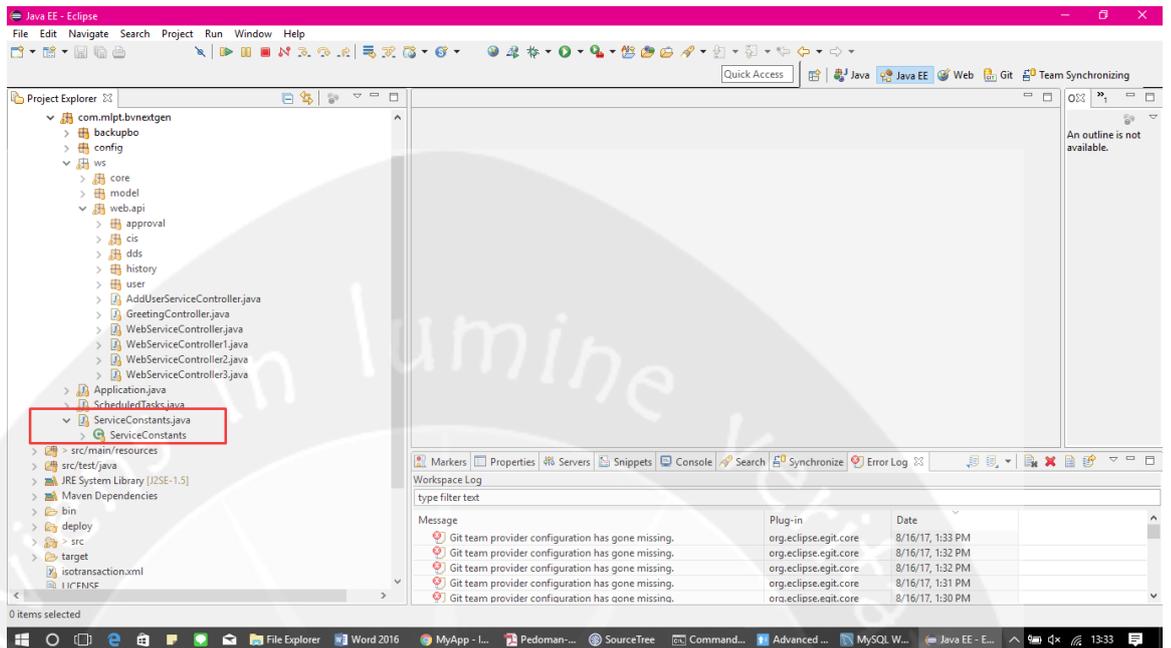
2.3.1. Bagian Eclipse Bo



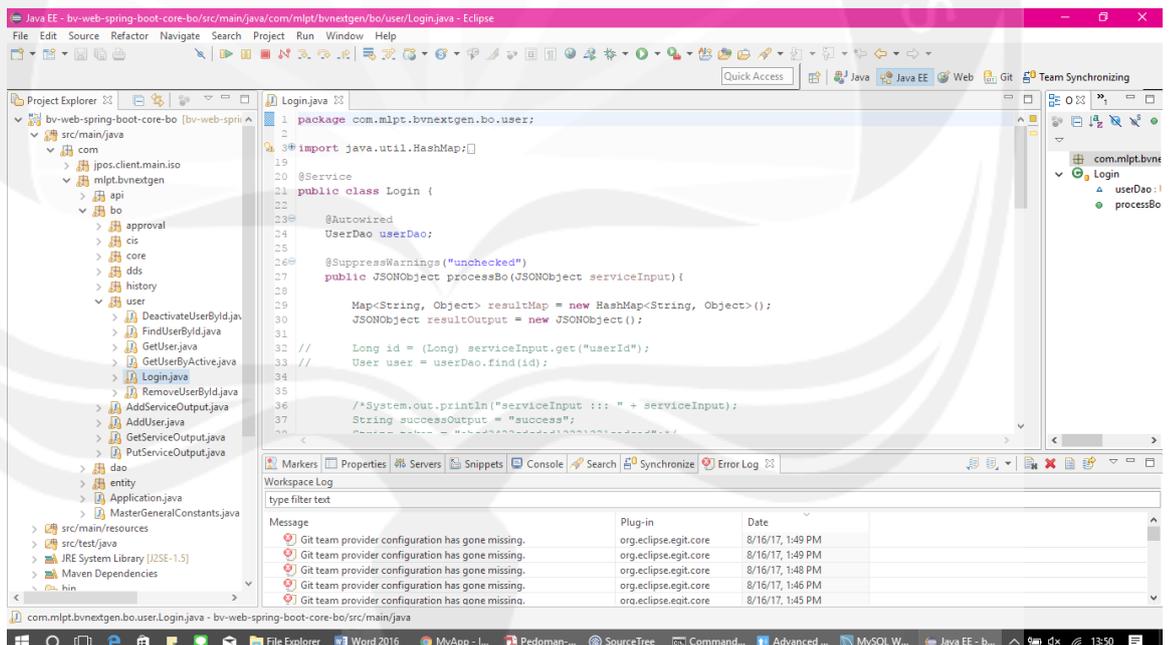
Gambar 1.23. Eclipse Bo (1)



Gambar 1.24. Eclipse Bo (2)

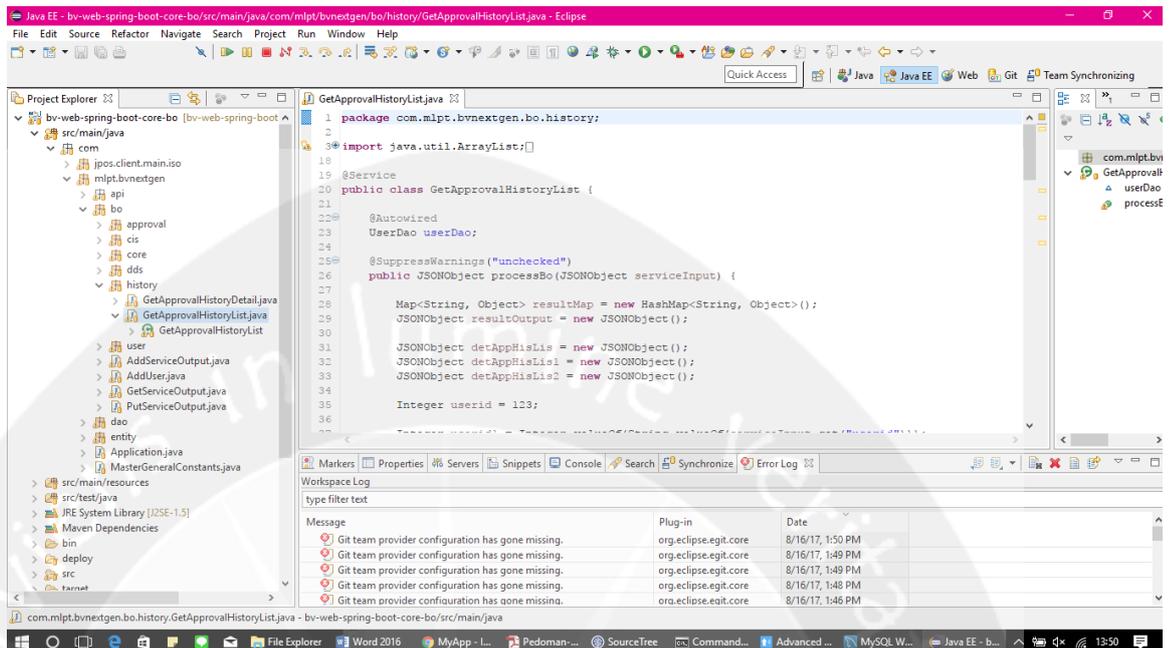


Gambar 1.25. Eclipse Bo (3)



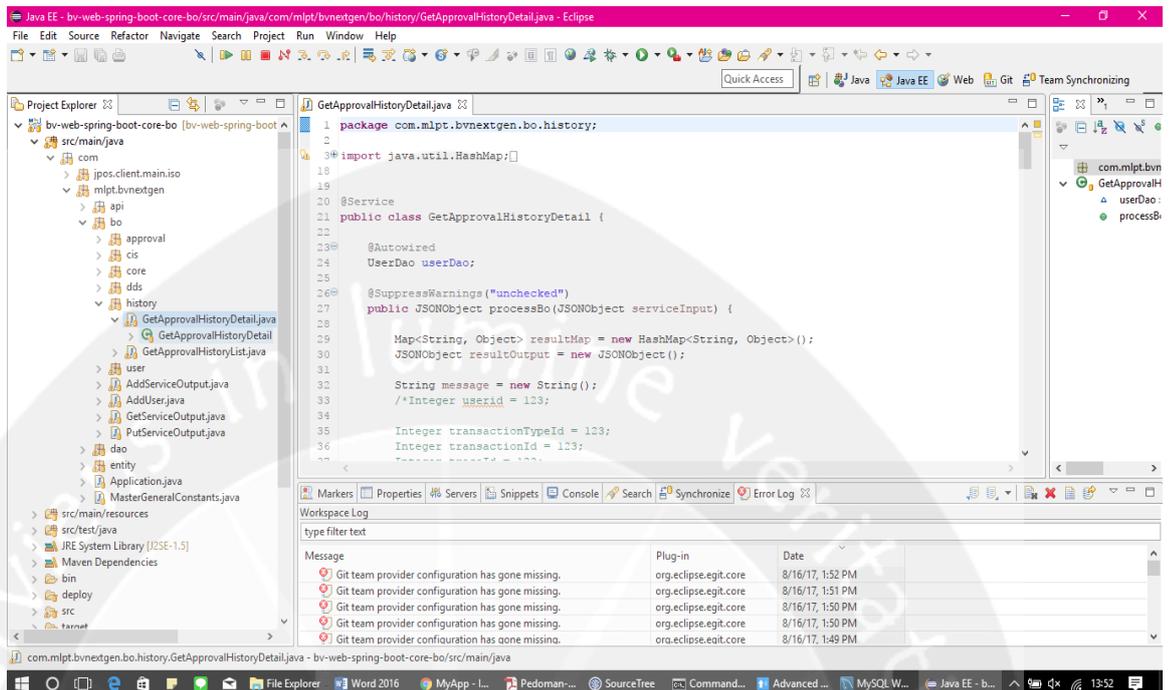
Gambar 1.26. Login.java

Untuk code Gambar 1.26. Login.java, merupakan *package* yang berisi *logic-logic* yang berfungsi untuk menerima inputan *user* yang digunakan untuk *login*, dimana inputan *user* berupa *username* dan *password* yang akan dibandingkan dengan *username* dan *password* yang ada untuk proses *login*.



Gambar 1.27. GetApprovalList.java

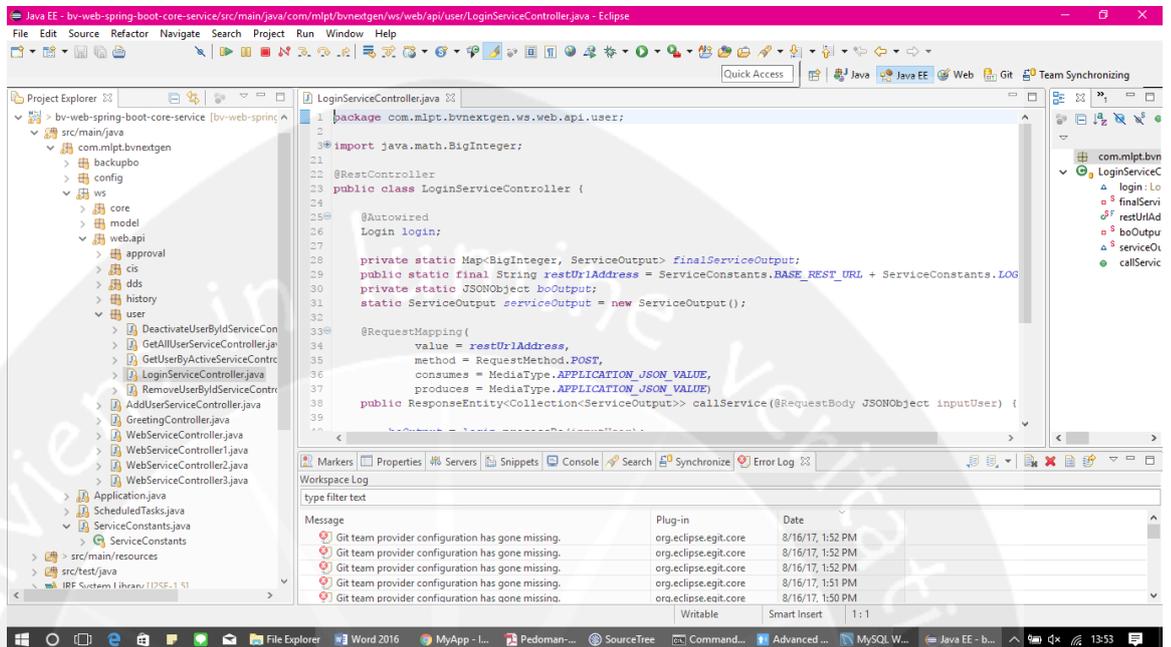
Untuk *code* Gambar 1.27. GetApprovalList.java, merupakan *package* yang berisi *logic-logic* yang berfungsi untuk menerima inputan *user* yang digunakan untuk menampilkan *list* dari *history approve* yang dilakukan oleh *user*. Dimana inputan *user* berupa *userid*, dimana *userid* ini akan dibandingkan dengan *userid* yang ada, jika *userid* cocok maka akan ditampilkan *listnya* dan jika inputan *userid* tidak cocok dengan yang ada maka akan ditampilkan sebuah pesan bahwa data yang memiliki *userid* tersebut tidak ada.



Gambar 1.28. GetApprovalHistoryDetail.java

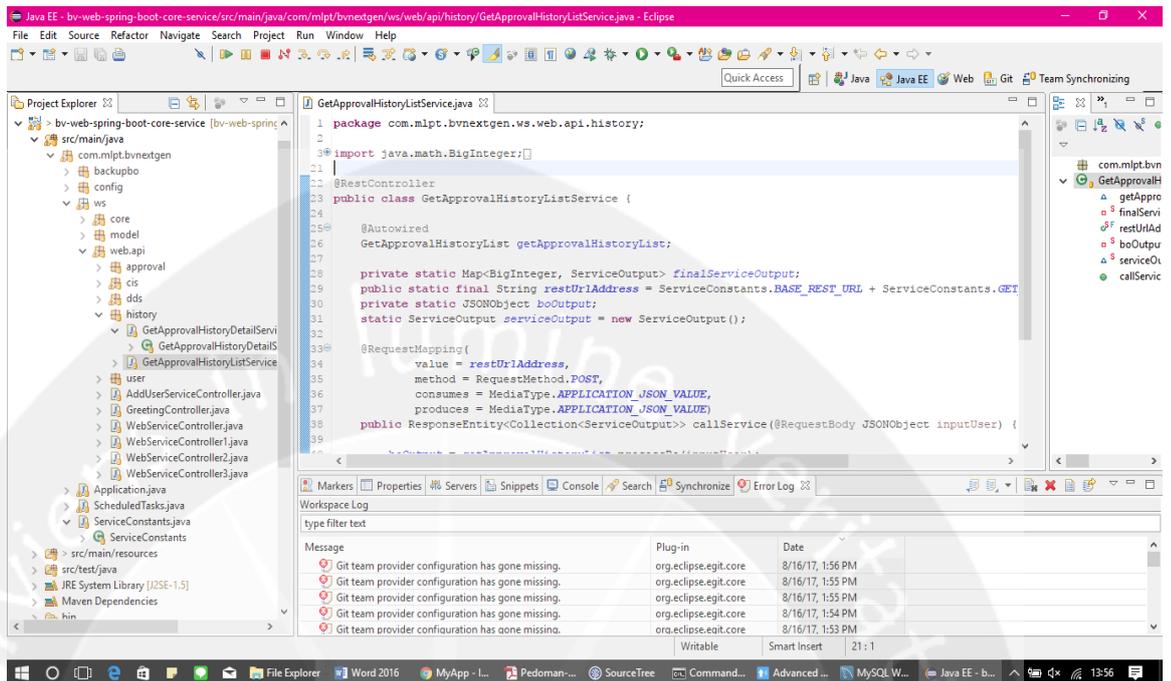
Untuk *code* Gambar 1.28. GetApprovalHistoryDetail.java merupakan *package* yang berisi *logic* yang berfungsi untuk menerima inputan *user* yang digunakan untuk menampilkan detail dari *history approve* yang dilakukan oleh *user*. Dimana inputan *user* berupa *userid*, *transactiontypeid*, *transactionid* dan *traceid*, inputan dari *user* tersebut akan dibandingkan dengan *userid*, *transactiontypeid*, *transactionid* dan *traceid* yang telah ada, jika data-data tersebut semuanya cocok maka detail dari *history approve* yang dibutuhkan akan ditampilkan.

2.3.2. Bagian Eclipse Service



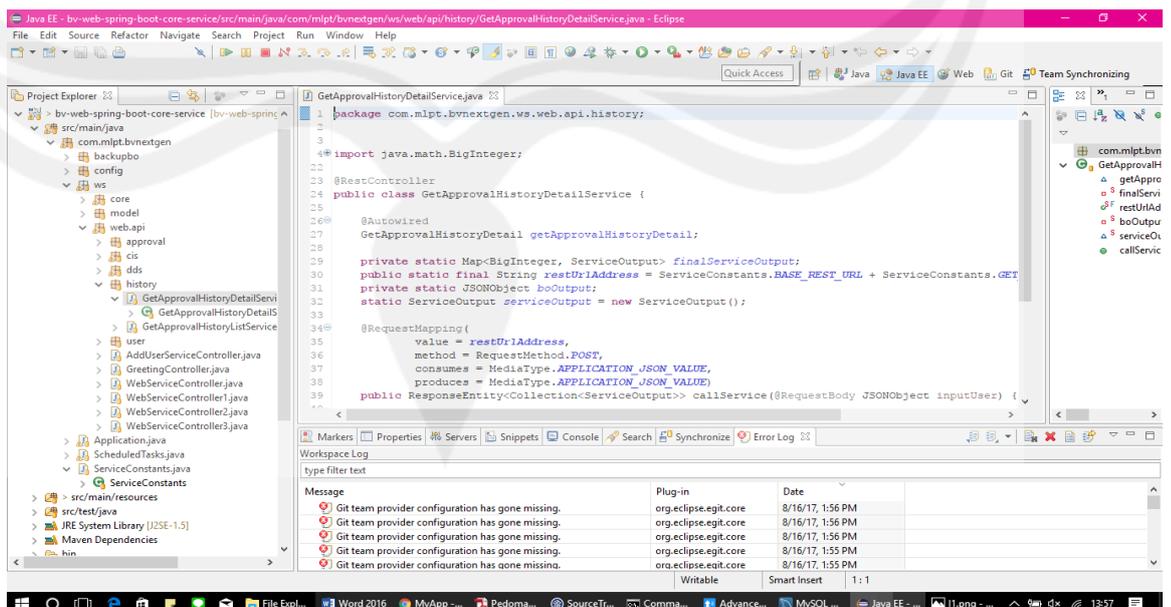
Gambar 1.29. LoginServiceController.java

Untuk *code* Gambar 1.29. LoginServiceController.java berisikan *code* yang digunakan sebagai penghubung rest web service dengan ionic, dimana ionic akan mengakses *bo(logic-logic)* dari *login* yang telah dibuat sebelumnya.



Gambar 1.30. GetApprovalHistoryListService.java

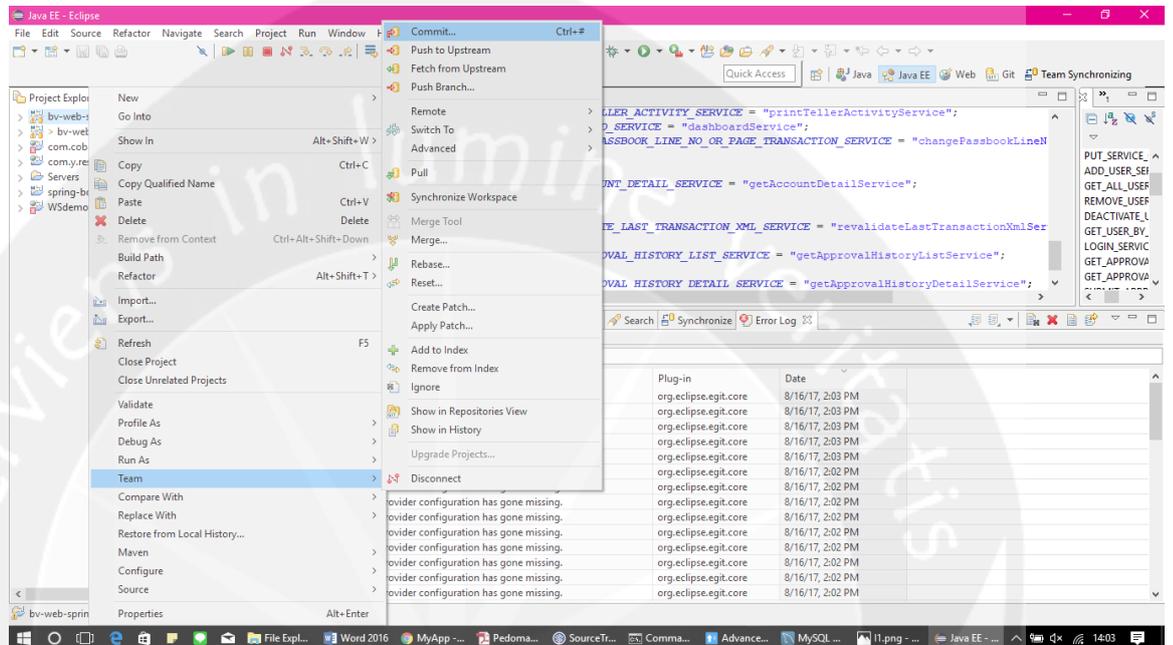
Untuk *code* Gambar 1.30. GetApprovalHistoryListService.java berisikan *code* yang digunakan sebagai penghubung rest web service dengan ionic, dimana ionic akan mengakses *bo(logic-logic)* dari GetApprovalHistoryList yang telah dibuat sebelumnya.



Gambar 1.31. GetApprovalDetailService.java

Gambar 1.32. Service Constant (1) dan Gambar 1.33. Service Constant (2) berisikan URL yang digunakan untuk mengakses rest web service.

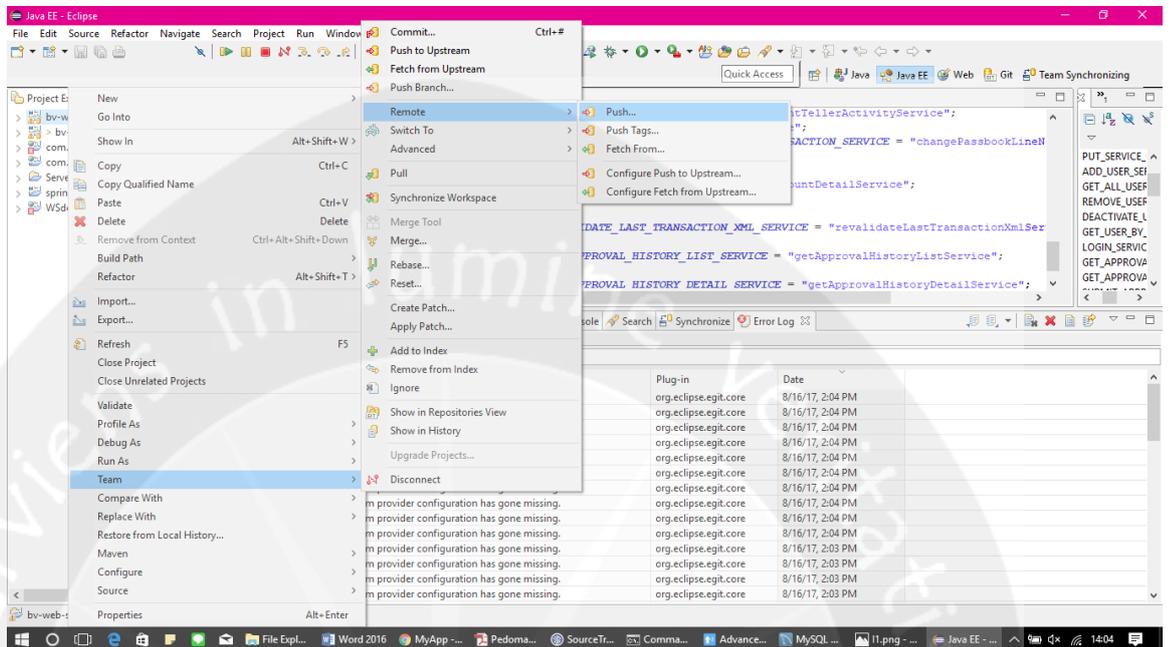
2.3.4. Commit Project (Eclipse)



Gambar 1.34. Commit Project (Eclipse)

Gambar 1.34. Commit Project (Eclipse) merupakan proses untuk *commit project*. *Commit* adalah proses menyimpan kode yang telah selesai di ubah ke repositori lokal. Untuk melakukan *commit* kita dapat menekan klik kanan pada proyek kita, kemudian klik menu *Team > Commit*, berikutnya isi kolom *commit message* dengan pesan *commit* yang sesuai dengan perubahan yang dilakukan.

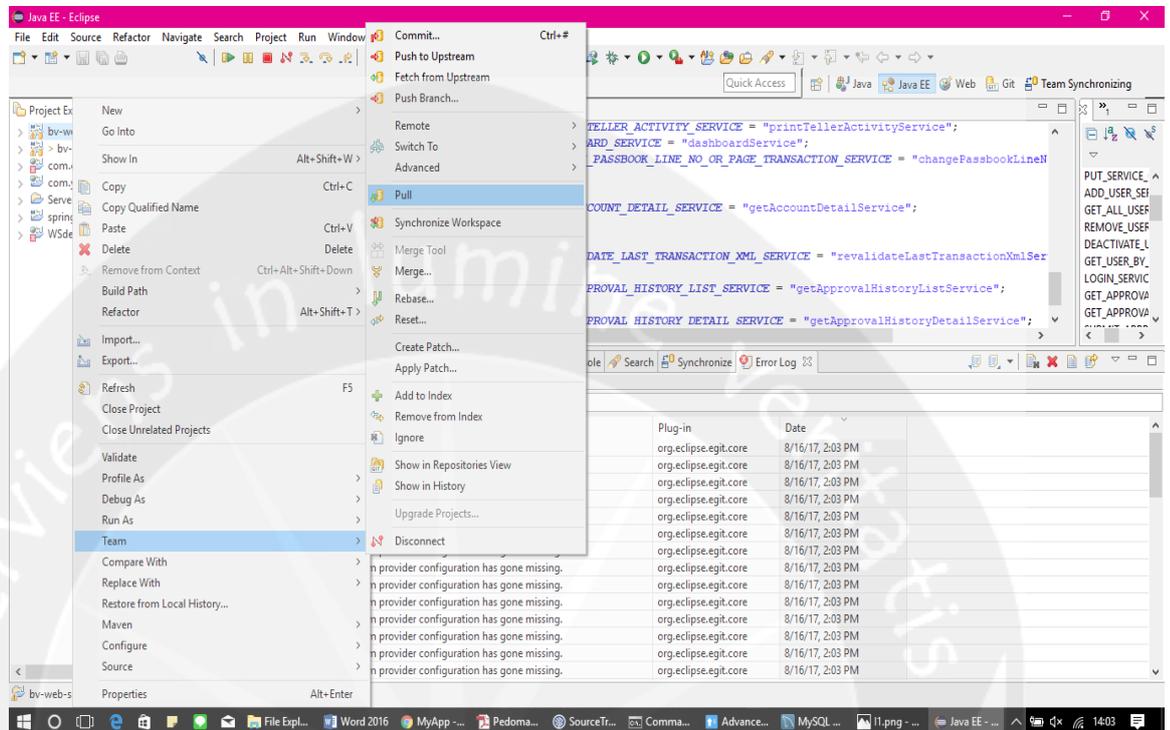
2.3.5. Push Project (Eclipse)



Gambar 1.35. Push Project (Eclipse)

Gambar 1.35. Push Project (Eclipse) merupakan proses untuk *push project* pada aplikasi eclipse. *Push* merupakan proses mengunggah data *commit* ke repositori publik dengan cara klik kanan pada *tree* proyek kemudian pilih menu *Team > Remote > Push*, pada URI isi dengan alamat ssh pada github, dan *next*. Pilih master -> *Add Spec* dan klik *finish*.

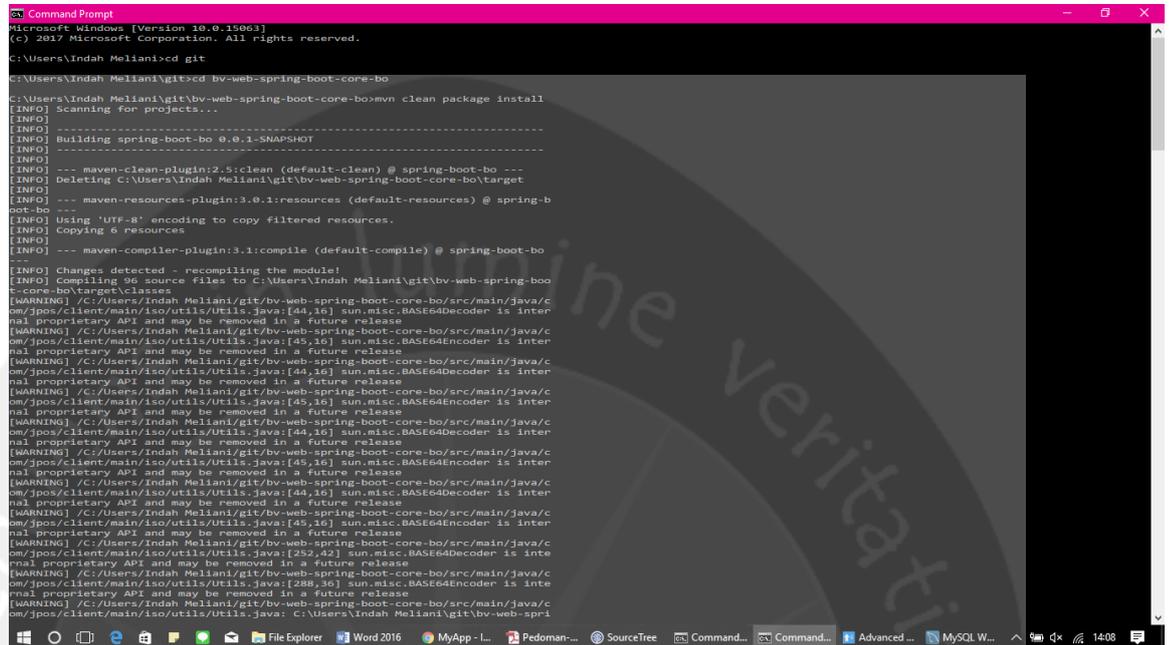
2.3.6. Pull Project (Eclipse)



Gambar 1.36. Pull Project (Eclipse)

Gambar 1.36. Pull Project (Eclipse) merupakan proses untuk *pull project* pada aplikasi eclipse. *Pull* merupakan proses untuk melakukan *update code* yang telah dirubah oleh *partner* ke *code* yang kita miliki, dengan cara klik kanan pada *tree* proyek kemudian pilih menu *Team > Pull*.

2.3.7. Command Prompt Bagian Bo



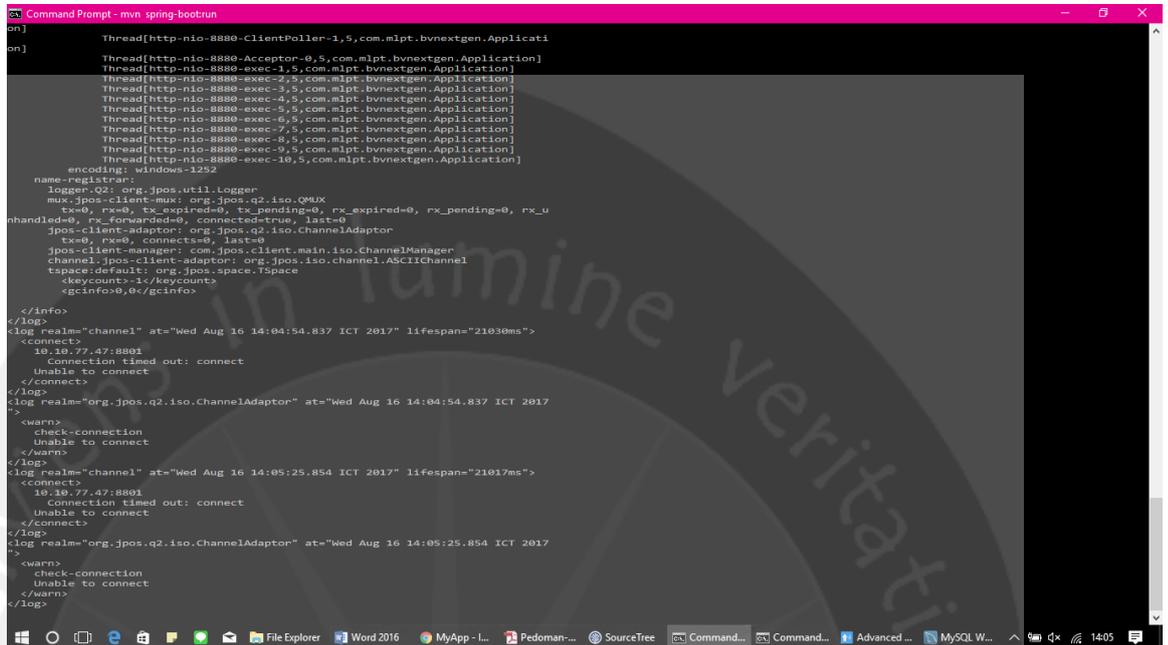
```
Microsoft Windows [Version 10.0.15063]
(c) 2017 Microsoft Corporation. All rights reserved.

C:\Users\Indah Meliani>cd git
C:\Users\Indah Meliani\git>cd bv-web-spring-boot-core-bo
C:\Users\Indah Meliani\git\bv-web-spring-boot-core-bo>mvn clean package install
[INFO] Scanning for projects...
[INFO]
[INFO] Building spring-boot-bo 0.0.1-SNAPSHOT
[INFO]
[INFO]
[INFO] --- maven-clean-plugin:2.5:clean (default-clean) @ spring-boot-bo ---
[INFO] Deleting C:\Users\Indah Meliani\git\bv-web-spring-boot-core-bo\target
[INFO]
[INFO] --- maven-resources-plugin:3.0.1:resources (default-resources) @ spring-bo
[INFO]
[INFO] --- maven-compiler-plugin:3.1:compile (default-compile) @ spring-boot-bo
[INFO] Changes detected - recompiling the module!
[INFO] Compiling 96 source files to C:\Users\Indah Meliani\git\bv-web-spring-bo
[WARNING] C:\Users\Indah Meliani\git\bv-web-spring-boot-core-bo\src/main/java/c
om\jpos\client\main\iso\utils\Utils.java:[44,16] sun.misc.BASE64Decoder is inter
nal proprietary API and may be removed in a future release
[WARNING] C:\Users\Indah Meliani\git\bv-web-spring-boot-core-bo\src/main/java/c
om\jpos\client\main\iso\utils\Utils.java:[45,16] sun.misc.BASE64Encoder is inter
nal proprietary API and may be removed in a future release
[WARNING] C:\Users\Indah Meliani\git\bv-web-spring-boot-core-bo\src/main/java/c
om\jpos\client\main\iso\utils\Utils.java:[44,16] sun.misc.BASE64Decoder is inter
nal proprietary API and may be removed in a future release
[WARNING] C:\Users\Indah Meliani\git\bv-web-spring-boot-core-bo\src/main/java/c
om\jpos\client\main\iso\utils\Utils.java:[45,16] sun.misc.BASE64Encoder is inter
nal proprietary API and may be removed in a future release
[WARNING] C:\Users\Indah Meliani\git\bv-web-spring-boot-core-bo\src/main/java/c
om\jpos\client\main\iso\utils\Utils.java:[49,16] sun.misc.BASE64Encoder is inter
nal proprietary API and may be removed in a future release
[WARNING] C:\Users\Indah Meliani\git\bv-web-spring-boot-core-bo\src/main/java/c
om\jpos\client\main\iso\utils\Utils.java:[49,16] sun.misc.BASE64Encoder is inter
nal proprietary API and may be removed in a future release
[WARNING] C:\Users\Indah Meliani\git\bv-web-spring-boot-core-bo\src/main/java/c
om\jpos\client\main\iso\utils\Utils.java:[49,16] sun.misc.BASE64Encoder is inter
nal proprietary API and may be removed in a future release
[WARNING] C:\Users\Indah Meliani\git\bv-web-spring-boot-core-bo\src/main/java/c
om\jpos\client\main\iso\utils\Utils.java:[252,42] sun.misc.BASE64Decoder is inte
rnal proprietary API and may be removed in a future release
[WARNING] C:\Users\Indah Meliani\git\bv-web-spring-boot-core-bo\src/main/java/c
om\jpos\client\main\iso\utils\Utils.java:[288,36] sun.misc.BASE64Encoder is inte
rnal proprietary API and may be removed in a future release
[WARNING] C:\Users\Indah Meliani\git\bv-web-spring-boot-core-bo\src/main/java/c
om\jpos\client\main\iso\utils\Utils.java: C:\Users\Indah Meliani\git\bv-web-spr
```

Gambar 1.37. Command Prompt Bo

Gambar 1.37. Command Prompt Bo merupakan command prompt yang digunakan untuk menjalankan Bo yang telah dibuat dengan cara mengetikkan *mvn clean package install* pada command prompt yang mengarah ke folder Bo.

2.3.8. Command Prompt Bagian Service

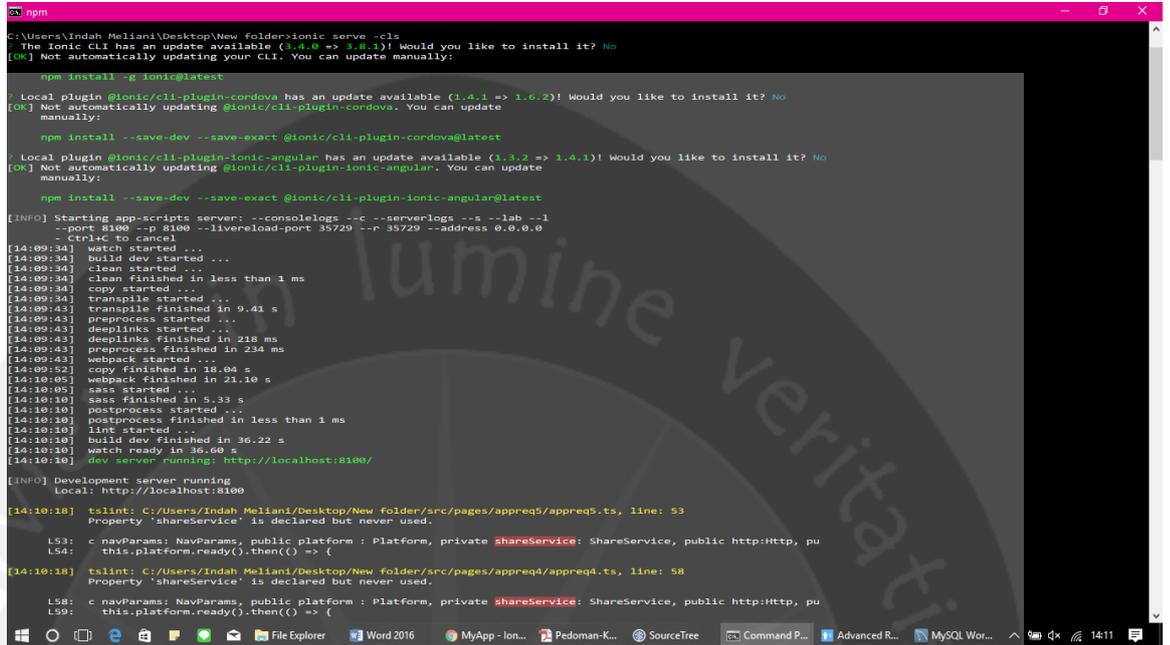


```
Command Prompt - mvn spring-boot:run
[sn]
Thread[http-nio-8880-ClientPoller-1,5,com.mipt.bvnextgen.Applicati
[sn]
Thread[http-nio-8880-Acceptor-0,5,com.mipt.bvnextgen.Application]
Thread[http-nio-8880-exec-1,5,com.mipt.bvnextgen.Application]
Thread[http-nio-8880-exec-2,5,com.mipt.bvnextgen.Application]
Thread[http-nio-8880-exec-3,5,com.mipt.bvnextgen.Application]
Thread[http-nio-8880-exec-4,5,com.mipt.bvnextgen.Application]
Thread[http-nio-8880-exec-5,5,com.mipt.bvnextgen.Application]
Thread[http-nio-8880-exec-6,5,com.mipt.bvnextgen.Application]
Thread[http-nio-8880-exec-7,5,com.mipt.bvnextgen.Application]
Thread[http-nio-8880-exec-8,5,com.mipt.bvnextgen.Application]
Thread[http-nio-8880-exec-9,5,com.mipt.bvnextgen.Application]
Thread[http-nio-8880-exec-10,5,com.mipt.bvnextgen.Application]
encoding: windows-1252
name:registram:
  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_u
  handled=0, rx_forwarded=0, connected=true, list=0
  jpos-client-adaptor: org.jpos.q2.iso.ChannelAdaptor
  tx=0, rx=0, connects=0, 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
  <keycount>-1/<keycount>
  <ginfo>0/<ginfo>
</info>
</log>
<log realm="channel" at="Wed Aug 16 14:04:54.837 ICT 2017" l1fespan="21039ms">
<connect>
  10:10:77:47:8801
  Connection timed out: connect
  Unable to connect
</connect>
</log>
<log realm="org.jpos.q2.iso.ChannelAdaptor" at="Wed Aug 16 14:04:54.837 ICT 2017">
<warn>
  check-connection
  Unable to connect
</warn>
</log>
<log realm="channel" at="Wed Aug 16 14:05:25.854 ICT 2017" l1fespan="21017ms">
<connect>
  10:10:77:47:8801
  Connection timed out: connect
  Unable to connect
</connect>
</log>
<log realm="org.jpos.q2.iso.ChannelAdaptor" at="Wed Aug 16 14:05:25.854 ICT 2017">
<warn>
  check-connection
  Unable to connect
</warn>
</log>
```

Gambar 1.38. Command Prompt Service

Gambar 1.38. Command Prompt Service merupakan command prompt yang digunakan untuk menjalankan *Service* yang telah dibuat dengan cara mengetikkan *mvn spring-boot:run* pada command Prompt yang mengarah ke folder *Service*, agar *service* tersebut dapat diakses menggunakan URL yang telah dibuat.

2.3.9. Command Prompt Ionic



```
C:\Users\Indah Meliani\Desktop\New folder>ionic serve -cls
The Ionic CLI has an update available (3.6.0 => 3.6.1)! Would you like to install it? No
[OK] Not automatically updating your CLI. You can update manually:

npm install -g ionic@latest

Local plugin @ionic/cli-plugin-cordova has an update available (1.4.1 => 1.6.2)! Would you like to install it? No
[OK] Not automatically updating @ionic/cli-plugin-cordova. You can update manually:

npm install --save-dev --save-exact @ionic/cli-plugin-cordova@latest

Local plugin @ionic/cli-plugin-ionic-angular has an update available (1.3.2 => 1.4.1)! Would you like to install it? No
[OK] Not automatically updating @ionic/cli-plugin-ionic-angular. You can update manually:

npm install --save-dev --save-exact @ionic/cli-plugin-ionic-angular@latest

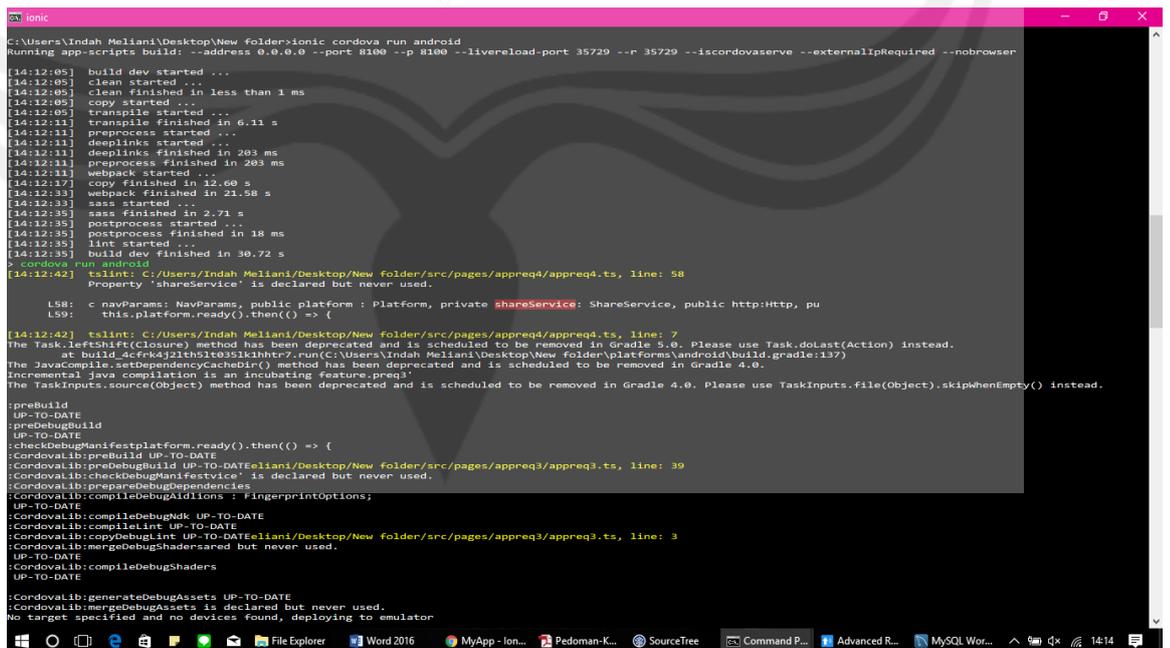
[INFO] Starting app-scripts server: --consolelogs --c --serverlogs --s --lab --l
--port 8100 --p 8100 --livereload-port 35729 --r 35729 --address 0.0.0.0
- Ctrl+C to cancel
[14:09:34] watch started ...
[14:09:34] build dev started ...
[14:09:34] clean started ...
[14:09:34] clean finished in less than 1 ms
[14:09:34] copy started ...
[14:09:34] transpile started ...
[14:09:43] transpile finished in 9.41 s
[14:09:43] preprocess started ...
[14:09:43] deeplinks started ...
[14:09:43] deeplinks finished in 218 ms
[14:09:43] preprocess finished in 234 ms
[14:09:43] webpack started ...
[14:09:52] copy finished in 18.04 s
[14:10:05] sass started ...
[14:10:10] sass finished in 5.33 s
[14:10:10] postprocess started ...
[14:10:10] postprocess finished in less than 1 ms
[14:10:10] lint started ...
[14:10:10] build dev finished in 36.22 s
[14:10:10] watch ready in 36.60 s
[14:10:10] dev server running: http://localhost:8100/

[INFO] Development server running
Local: http://localhost:8100

[14:10:18] tsLint: C:/Users/Indah Meliani/Desktop/New folder/src/pages/appreq5/appreq5.ts, line: 53
Property 'shareService' is declared but never used.
L53: c navParams: NavParams, public platform : Platform, private shareService: ShareService, public http:Http, pu
L54: this.platform.ready().then() => {
[14:10:18] tsLint: C:/Users/Indah Meliani/Desktop/New folder/src/pages/appreq4/appreq4.ts, line: 58
Property 'shareService' is declared but never used.
L58: c navParams: NavParams, public platform : Platform, private shareService: ShareService, public http:Http, pu
L59: this.platform.ready().then() => {
```

Gambar 1.39. Command Prompt Ionic Test Browser

Gambar 1.39. Command Prompt Ionic Test Browser merupakan command prompt yang digunakan untuk menjalankan *project ionic* yang telah dikerjakan di *browser*, dengan cara megetikkan *ionic serve-cls* pada command prompt yang telah mengarah ke folder *project ionic*.

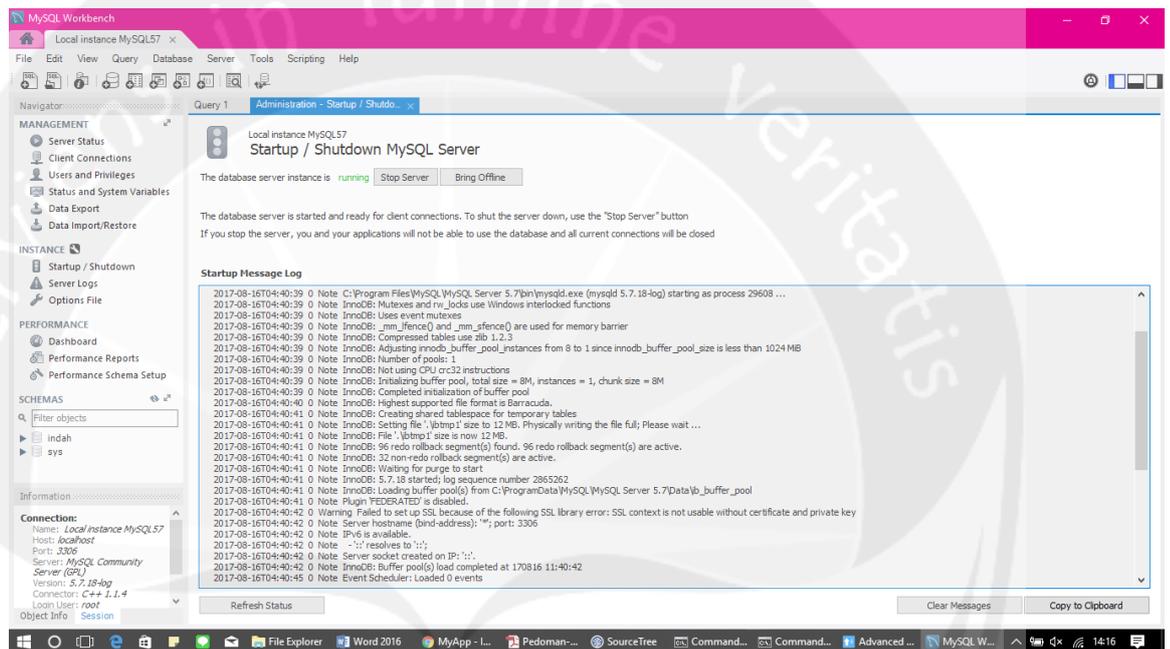


```
C:\Users\Indah Meliani\Desktop\New folder>ionic cordova run android
Running app-scripts build: --address 0.0.0.0 --port 8100 --p 8100 --livereload-port 35729 --r 35729 --iscordovaserve --externalipRequired --nobrowser
[14:12:05] build dev started ...
[14:12:05] clean started ...
[14:12:05] clean finished in less than 1 ms
[14:12:05] copy started ...
[14:12:05] transpile started ...
[14:12:11] transpile finished in 6.11 s
[14:12:11] preprocess started ...
[14:12:11] deeplinks started ...
[14:12:11] deeplinks finished in 203 ms
[14:12:11] preprocess finished in 203 ms
[14:12:11] webpack started ...
[14:12:17] copy finished in 12.60 s
[14:12:33] sass started ...
[14:12:35] sass finished in 2.71 s
[14:12:35] postprocess started ...
[14:12:35] postprocess finished in 18 ms
[14:12:35] lint started ...
[14:12:35] build dev finished in 30.72 s
> cordova run android
[14:12:42] tsLint: C:/Users/Indah Meliani/Desktop/New folder/src/pages/appreq4/appreq4.ts, line: 58
Property 'shareService' is declared but never used.
L58: c navParams: NavParams, public platform : Platform, private shareService: ShareService, public http:Http, pu
L59: this.platform.ready().then() => {
[14:12:42] tsLint: C:/Users/Indah Meliani/Desktop/New folder/src/pages/appreq4/appreq4.ts, line: 7
The Task.leftShift(Closure) method has been deprecated and is scheduled to be removed in Gradle 5.0. Please use Task.doLast(Action) instead.
at build_acFrK4j21th51t0351k1h7r.run(C:/Users/Indah Meliani/Desktop/New folder/platforms/android/build.gradle:137)
The JavaCompile.setDependencyCacheDir() method has been deprecated and is scheduled to be removed in Gradle 4.0.
Incremental java compilation is an incubating feature.
The TaskInputs.source(Object) method has been deprecated and is scheduled to be removed in Gradle 4.0. Please use TaskInputs.file(Object).skipWhenEmpty() instead.
:preBuild UP-TO-DATE
:preDebugBuild UP-TO-DATE
:preDebugBuild UP-TO-DATE
:checkDebugManifestplatform_ready().then() => {
:cordovaLib:preBuild UP-TO-DATE
:cordovaLib:preDebugBuild UP-TO-DATEIndah Meliani/Desktop/New folder/src/pages/appreq3/appreq3.ts, line: 39
:cordovaLib:checkDebugManifest' is declared but never used.
:cordovaLib:mergeDebugDependencies
:cordovaLib:compileDebugAidlions : FingerprintOptions;
UP-TO-DATE
:cordovaLib:compileDebugNdk UP-TO-DATE
:cordovaLib:compileLint UP-TO-DATE
:cordovaLib:copyDebugLint UP-TO-DATEIndah Meliani/Desktop/New folder/src/pages/appreq3/appreq3.ts, line: 3
:cordovaLib:mergeDebugShadersard but never used.
UP-TO-DATE
:cordovaLib:compileDebugShaders
UP-TO-DATE
:cordovaLib:generateDebugAssets UP-TO-DATE
:cordovaLib:mergeDebugAssets is declared but never used.
No target specified and no devices found, deploying to emulator
```

Gambar 1.40. Command Prompt Ionic Test Device

Gambar 1.40. Command Prompt Ionic Test Device merupakan command prompt yang digunakan untuk menjalankan *project ionic* yang telah dikerjakan di *device android*, dengan cara megetikkan *cordova run android* pada command prompt yang telah mengarah ke folder *project ionic*.

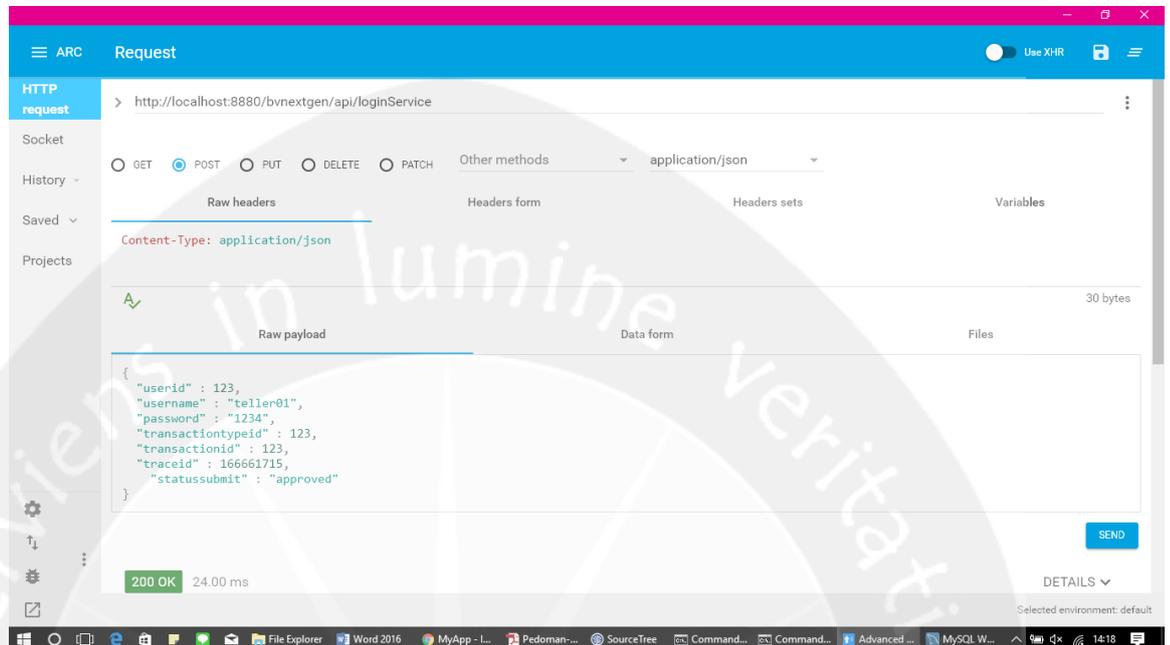
2.3.10. MySQL Workbench



Gambar 1.41. MySQL Workbench

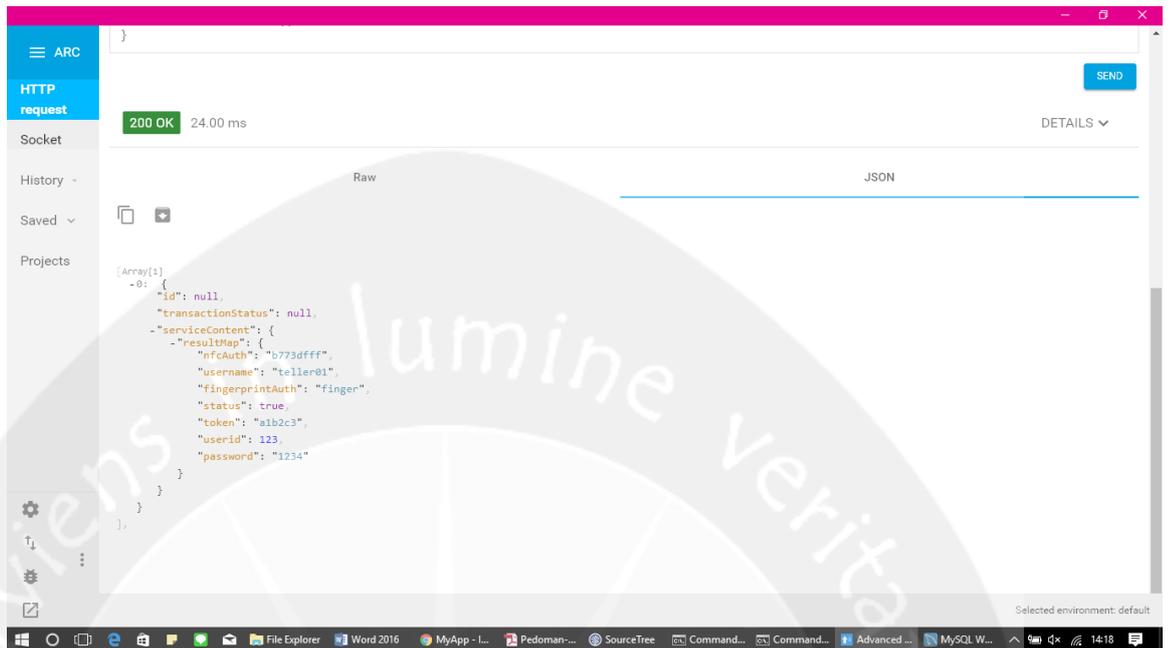
Gambar 1.41. MySQL Workbench merupakan MySQL Workbench yang digunakan untuk menjalankan *server* dengan cara mengklik *startup/shutdown* pada bagian *instance* di sisi sebelah kiri.

2.3.11. Advanced Rest Client Login



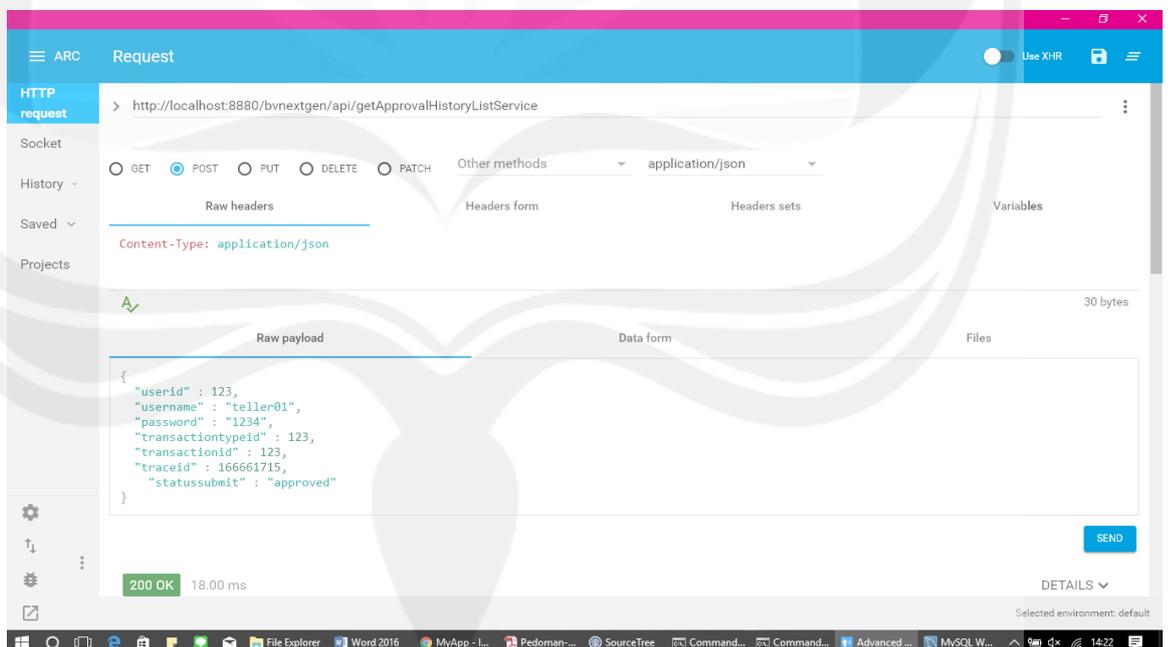
Gambar 1.42. Advanced Rest Client Login (1)

Gambar 1.42. Advanced Rest Client Login (1) adalah proses mengetest *Bo* dan *Service* yang telah dibuat, contohnya kita disini ingin mencoba fungsi *login* dapat diakses atau tidak dengan cara; masukkan URL yang telah dibuat yaitu `loginService` kemudian pilih yang `POST` dan inputkan data yang dibutuhkan untuk *login* beserta variabelnya seperti `"username" : "teller01"` dan `password : "1234"` kemudian klik `send`, dan inputan dicocokkan di *Bo* nya dan jika benar memberikan balikan bertipe *boolean* yaitu `true` dan data yang dimilikinya seperti pada Gambar 1.43 berikut:



Gambar 1.43. Advanced Rest Client Login (2)

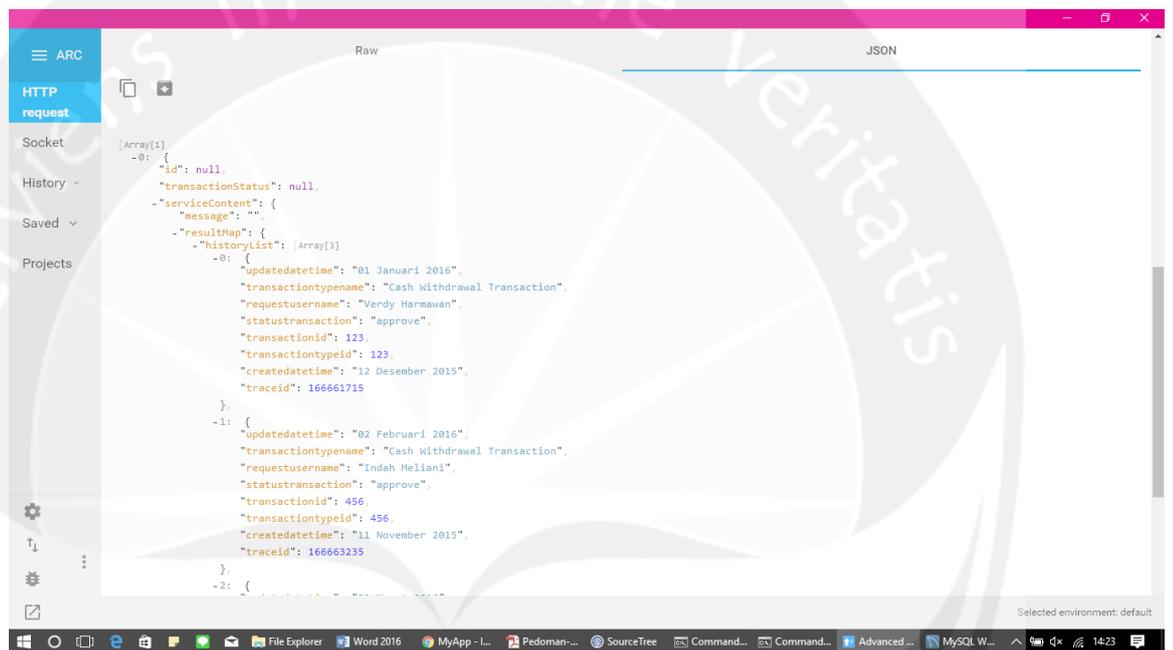
2.3.12. Advanced Rest Client Approval History List



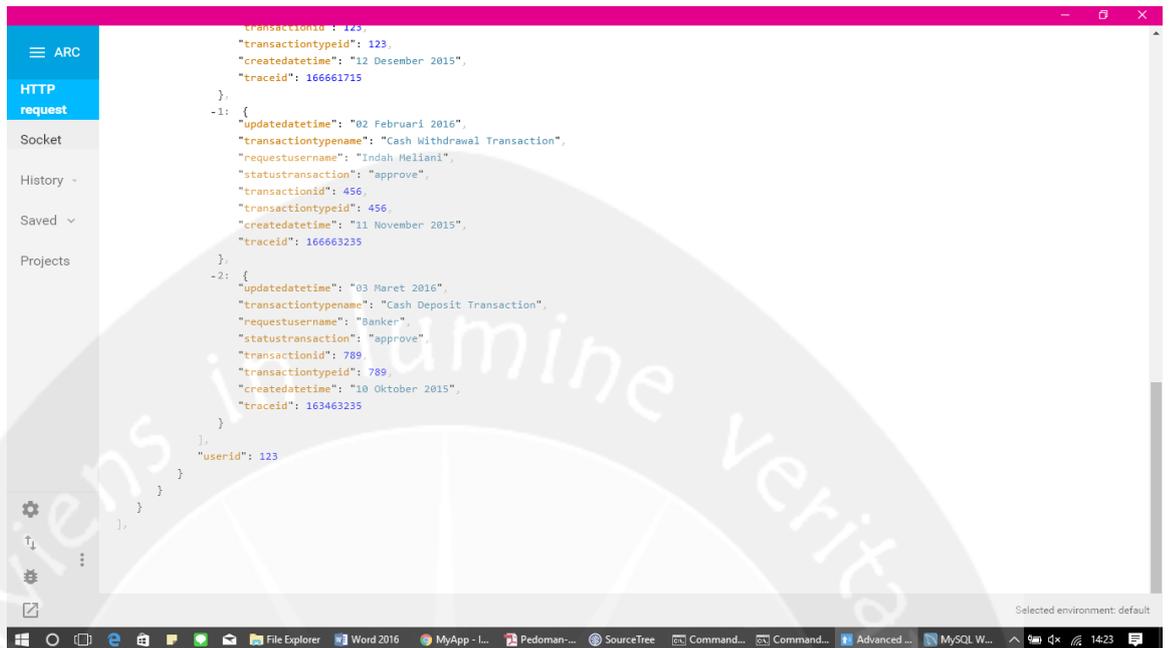
Gambar 1.44. Advanced Rest Client Approval History List (1)

Gambar 1.44. Advanced Rest Client Approval History List (1) merupakan proses untuk mengetest *Bo* dan *Service* yang telah dibuat, contohnya disini kita ingin mengetest fungsi *approval history list* dapat

diakses atau tidak dengan cara; masukkan URL yang telah dibuat yaitu `getApprovalHistoryListService` kemudian pilih yang “*POST*” dan inputkan data yang dibutuhkan untuk *approval history list* beserta variabelnya seperti “*userid*” : 123 kemudian klik *send*, dan inputan dicocokkan di *body* nya dan jika benar memberikan balikan yaitu data yang dimilikinya dengan *userid* tersebut seperti pada Gambar 1.45. dan Gambar 1.46 berikut :

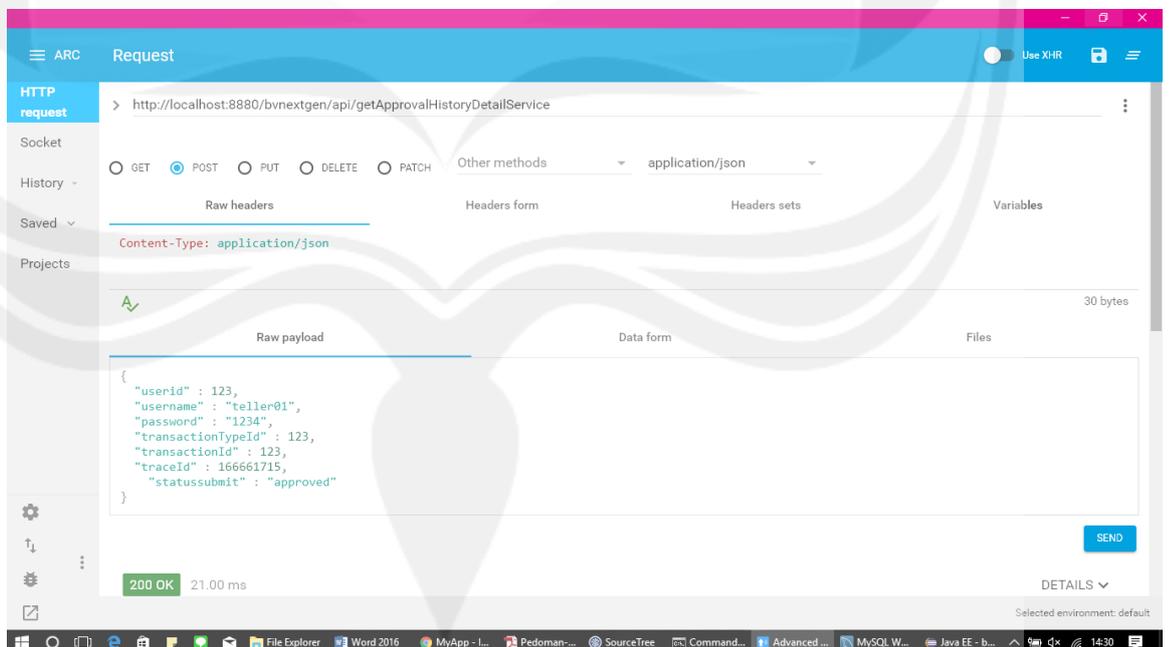


Gambar 1.45. Advanced Rest Client Approval History List (2)



Gambar 1.46. Advanced Rest Client Approval History List (3)

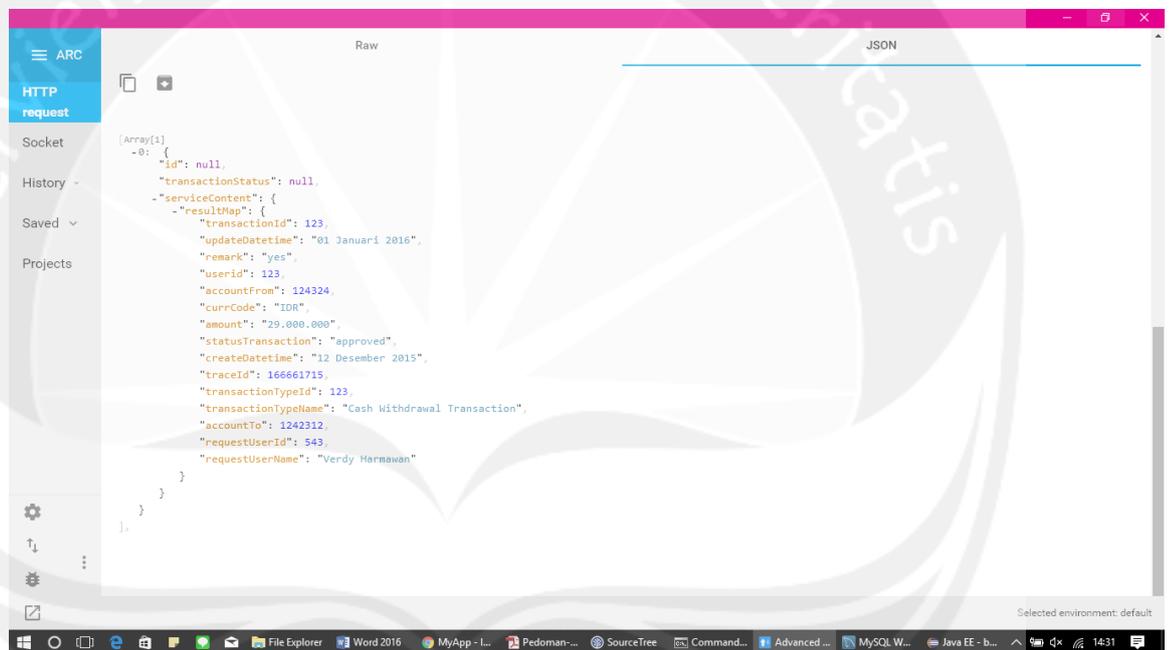
2.3.13. Advanced Rest Client Approval History Detail



Gambar 1.47. Advanced Rest Client Approval History Detail (1)

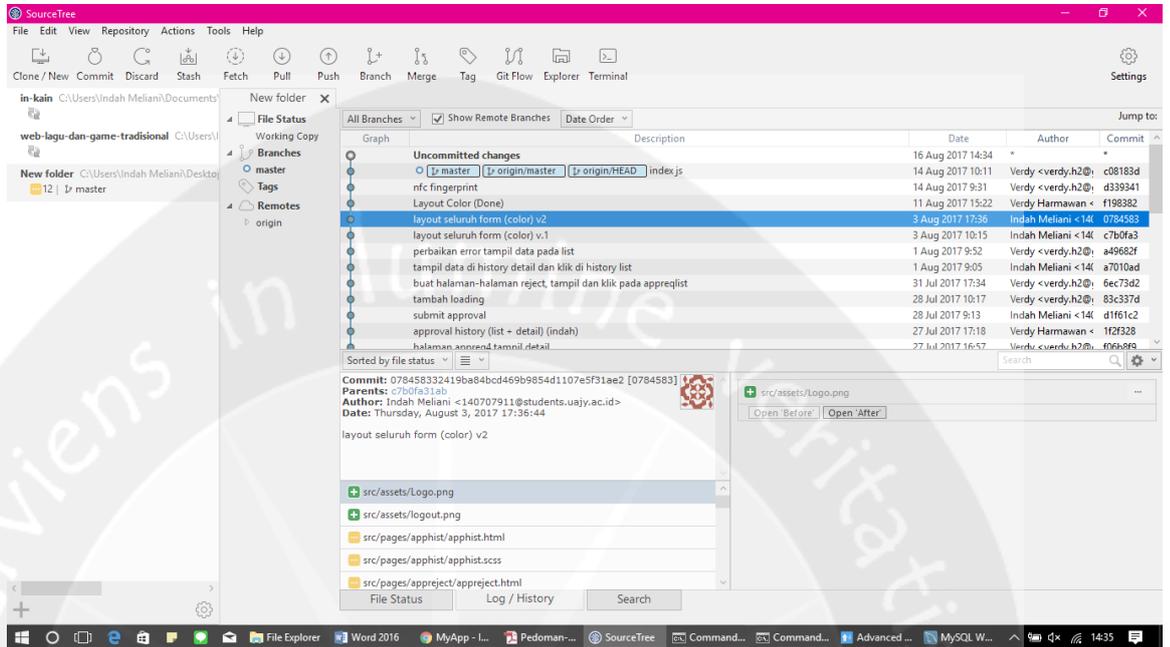
Gambar 1.47. Advanced Rest Client Approval History Detail (1) merupakan proses untuk mengetest *Bo* dan *Service* yang telah dibuat,

contohnya kita ingin mengetest fungsi *approval history detail* dapat diakses atau tidak dengan cara; masukkan URL yang telah dibuat yaitu `getApprovalHistoryDetailService` kemudian pilih yang “*POST*” dan inputkan data yang dibutuhkan untuk *approval history list* beserta variabelnya seperti “*userid*” : 123, *transactiontypeid* : 1234, “*transactionid*” : 123, dan “*traceid*” : 166661715 kemudian klik *send* dan inputan dicocokkan di *body* nya dan jika benar memberikan balikan yaitu data yang dimiliki *userid*, *transactiontypeid*, *transactionid*, dan *traceid* tersebut seperti pada Gambar 1.48 berikut :



Gambar 1.48. Advanced Rest Client Approval History Detail (2)

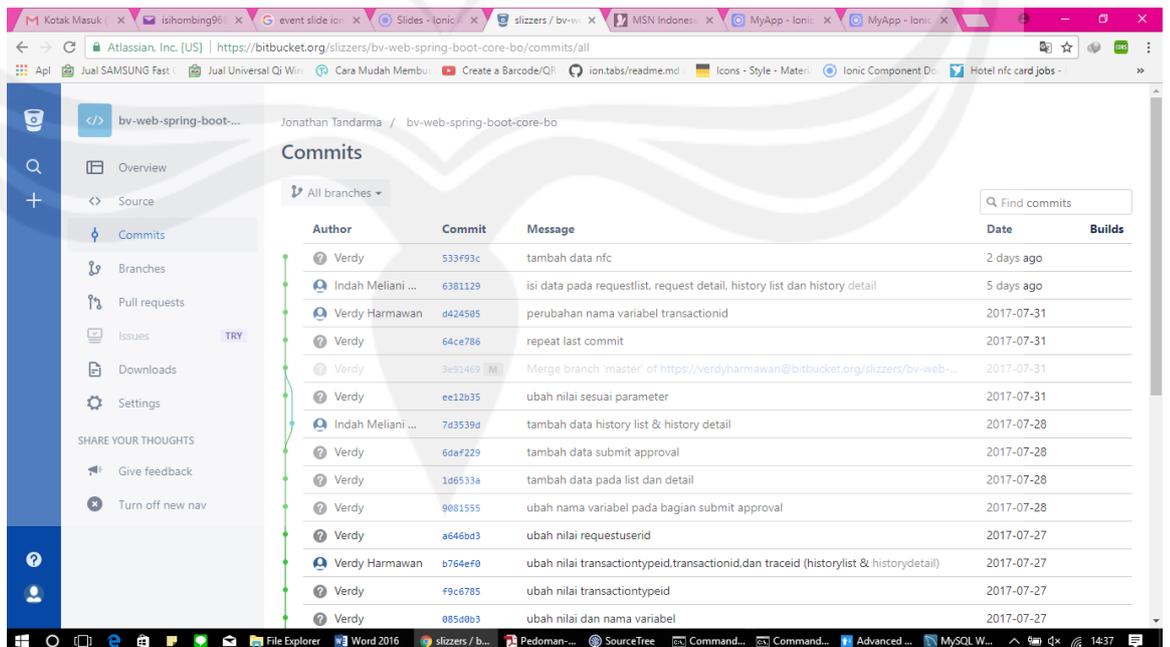
2.3.14. SorceTree



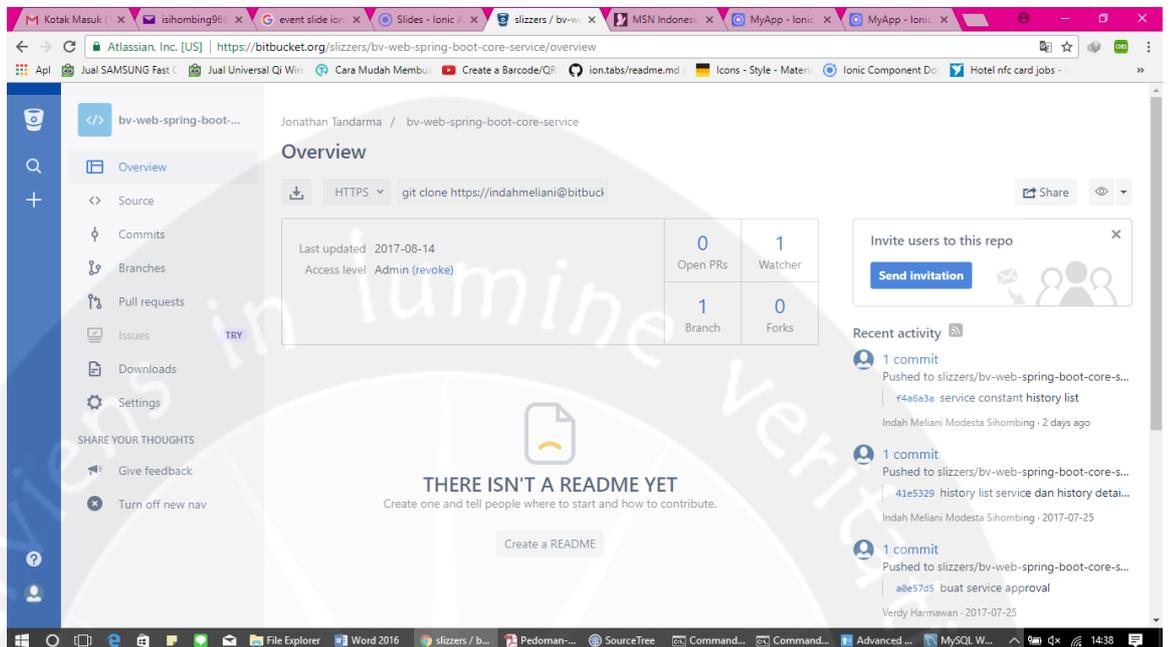
Gambar 1.49. Source Tree

Gambar 1.49 menunjukkan *tool* untuk *push* dan *pull project* dari dan ke bitbucket.

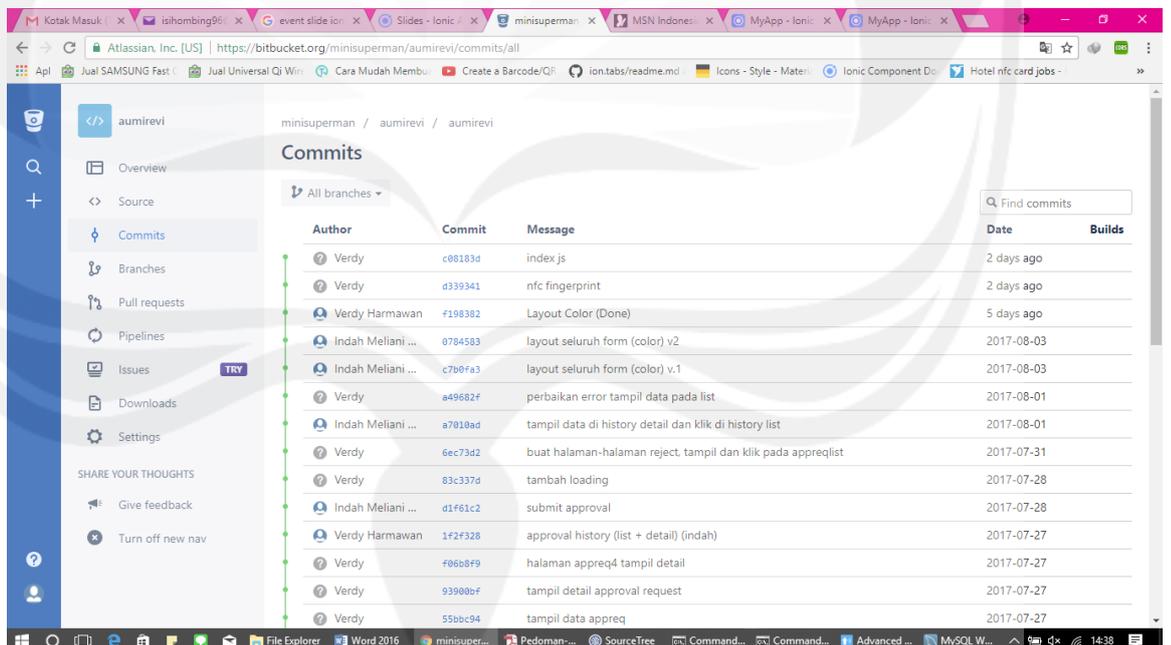
2.3.15. Bitbucket



Gambar 1.50. Bitbucket (1)



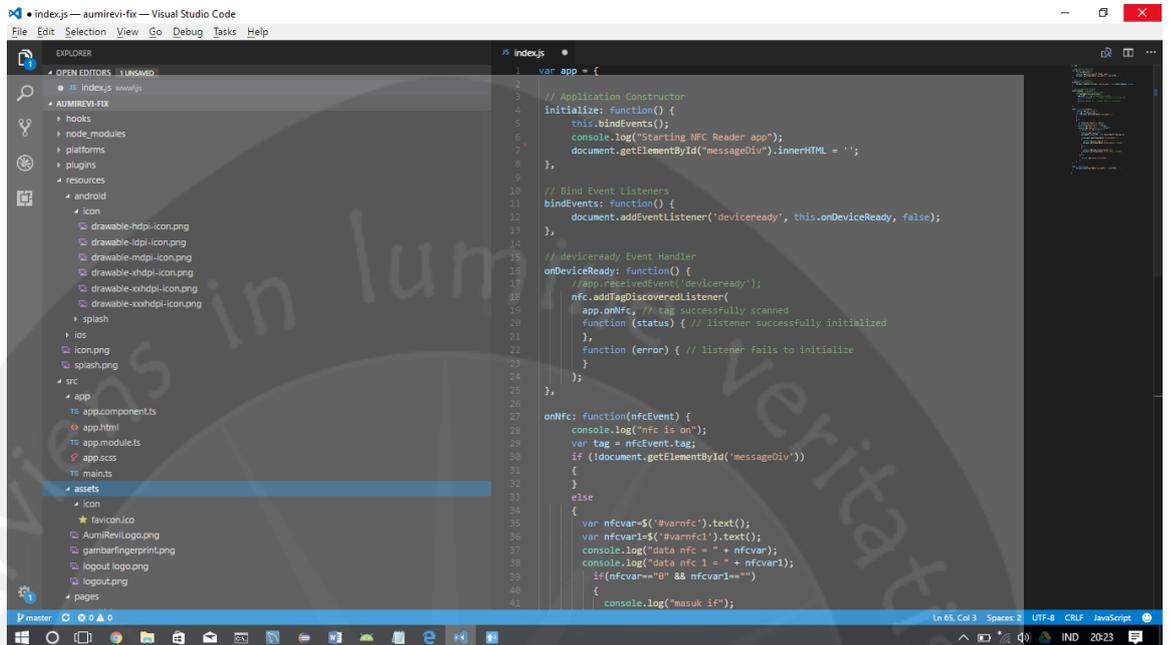
Gambar 1.51. Bitbucket (2)



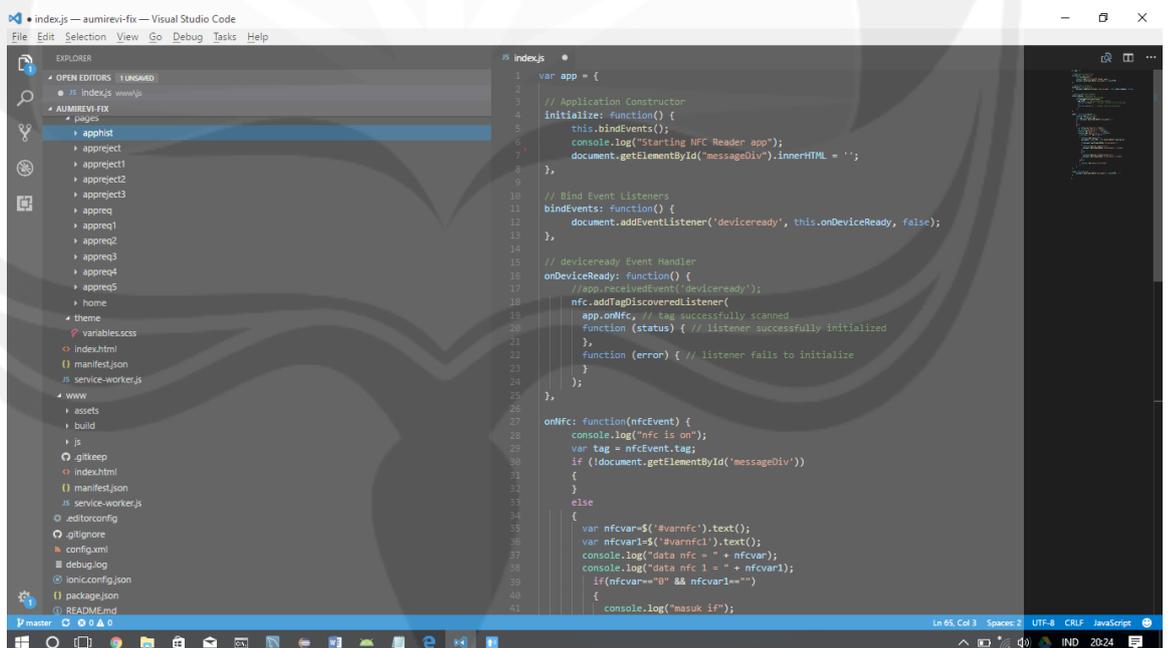
Gambar 1.52. Bitbucket (3)

Gambar 1.50 sampai gambar 1.52 merupakan *tool* yang digunakan untuk *share project* yang sedang dikerjakan baik *ionic*, *bo* dan *service* pada aplikasi *eclipse*.

2.3.16. Project Ionic

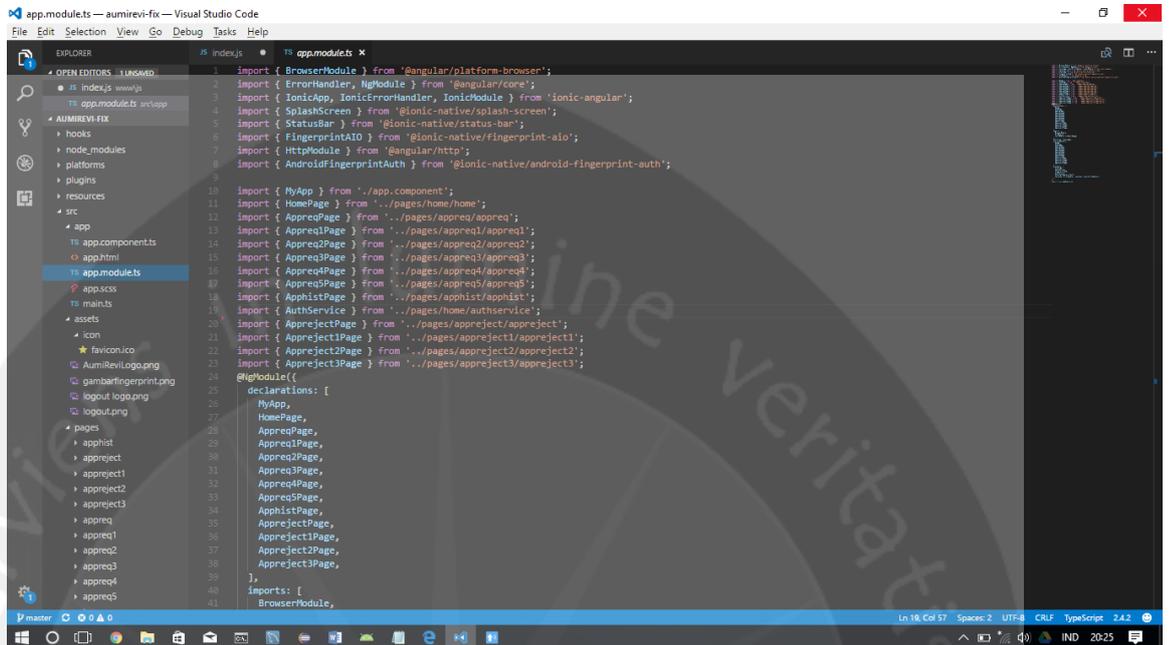


Gambar 1.53. Project Ionic (1)



Gambar 1.54. Project Ionic (2)

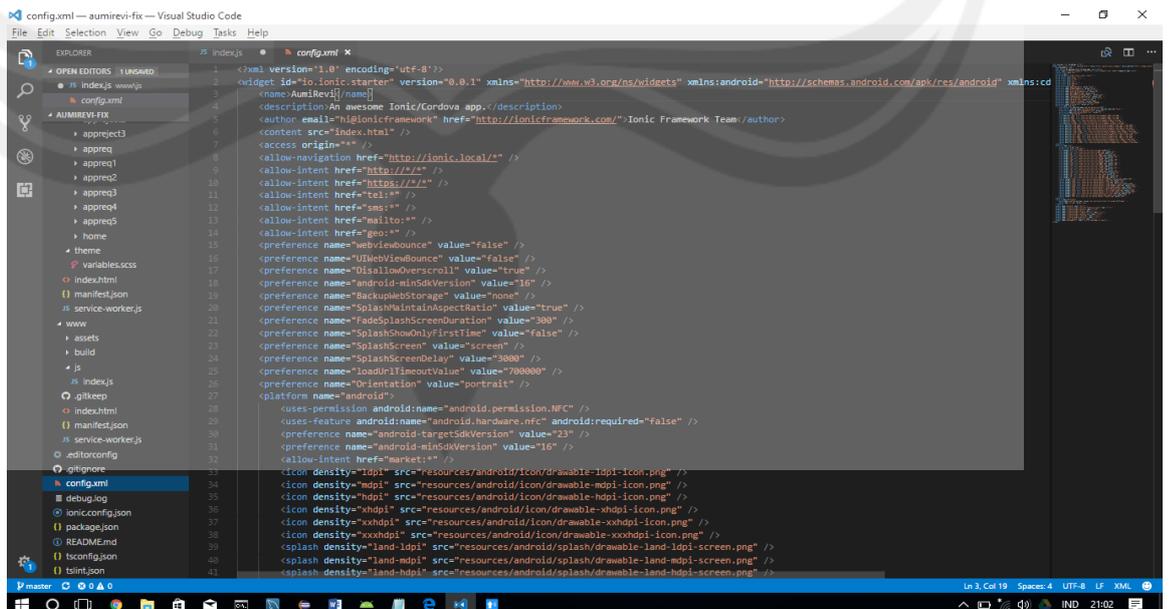
2.3.17. Ionic (app.module.ts)



Gambar 1.55. app.module.ts

Gambar 1.55 menunjukkan app.module.ts pada ionic yang digunakan untuk mendeklarasikan halaman yang terdapat pada *project* dan juga mendeklarasikan *provider* yang digunakan seperti *NFC* dan *Fingerprint*.

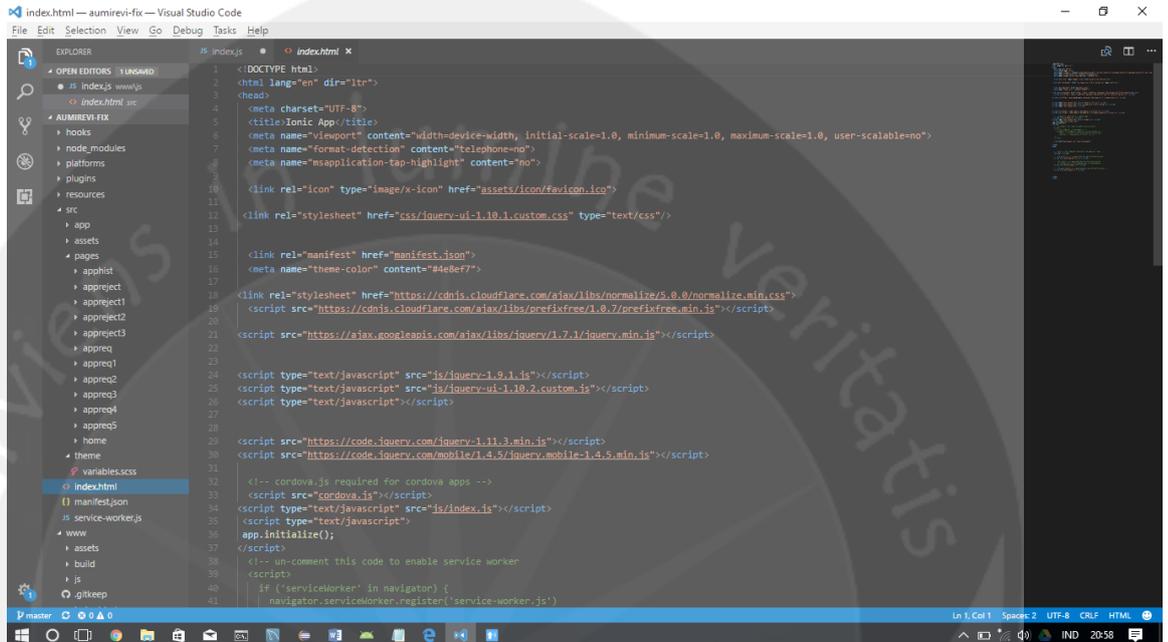
2.3.18. Ionic (config.xml)



Gambar 1.56. config.xml

Gambar 1.56 menunjukkan config.xml pada ionic yang berisikan *plugin-plugin* yang digunakan.

2.3.19. Ionic (Index.html)

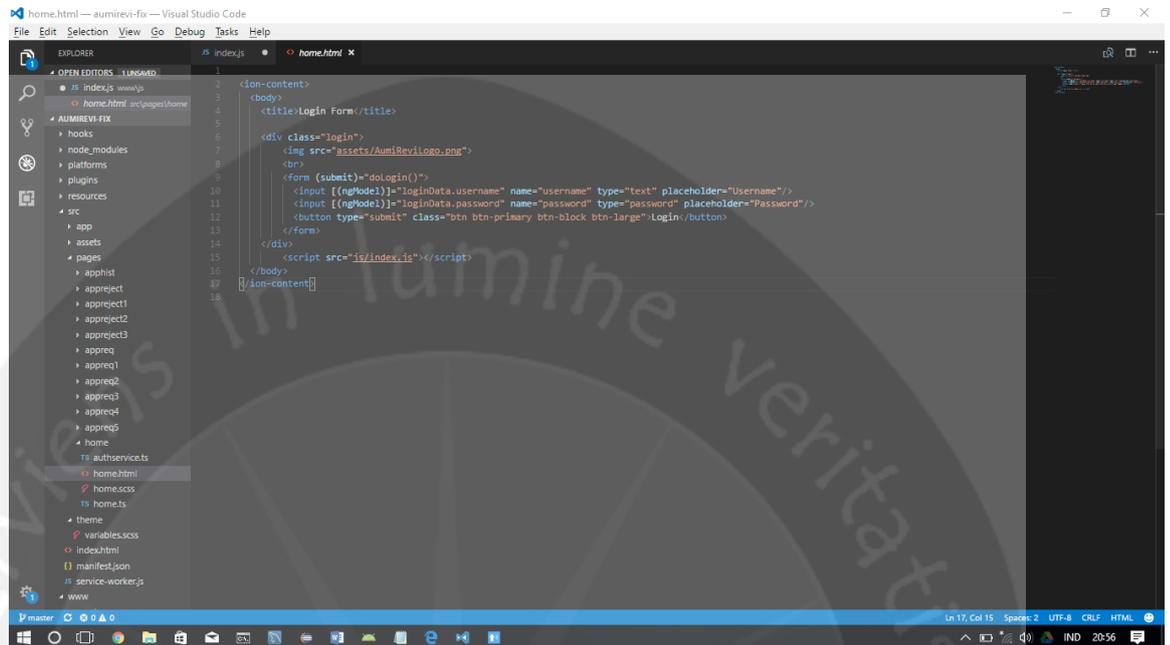


```
1 <!DOCTYPE html>
2 <html lang="en" dir="ltr">
3 <head>
4   <meta charset="UTF-8">
5   <title>Ionic App</title>
6   <meta name="viewport" content="width=device-width, initial-scale=1.0, minimum-scale=1.0, maximum-scale=1.0, user-scalable=no">
7   <meta name="format-detection" content="telephone=no">
8   <meta name="msapplication-tap-highlight" content="no">
9
10  <link rel="icon" type="image/x-icon" href="assets/icon/favicon.ico">
11
12  <link rel="stylesheet" href="css/jquery-ui-1.10.1.custom.css" type="text/css"/>
13
14  <link rel="manifest" href="manifest.json">
15  <meta name="theme-color" content="#4e8ef7">
16
17  <link rel="stylesheet" href="https://cdnjs.cloudflare.com/ajax/libs/normalize/5.0.0/normalize.min.css">
18  <script src="https://cdnjs.cloudflare.com/ajax/libs/jquery/1.10.7/jquery.min.js"></script>
19  <script src="https://ajax.googleapis.com/ajax/libs/jquery/1.7.1/jquery.min.js"></script>
20
21  <script type="text/javascript" src="js/jquery-1.9.1.js"></script>
22  <script type="text/javascript" src="js/jquery-ui-1.10.2.custom.js"></script>
23  <script type="text/javascript"></script>
24
25  <script src="https://code.jquery.com/jquery-1.11.3.min.js"></script>
26  <script src="https://code.jquery.com/mobile/1.4.5/jquery.mobile-1.4.5.min.js"></script>
27
28  <!-- cordova.js required for cordova apps -->
29  <script src="cordova.js"></script>
30  <script type="text/javascript" src="js/index.js"></script>
31  <script type="text/javascript">
32    app.initialize();
33  </script>
34  <!-- un-comment this code to enable service worker
35  <script>
36    if ('serviceWorker' in navigator) {
37      navigator.serviceWorker.register('service-worker.js')
38    }
39  </script>
40
41
```

Gambar 1.57. index.html

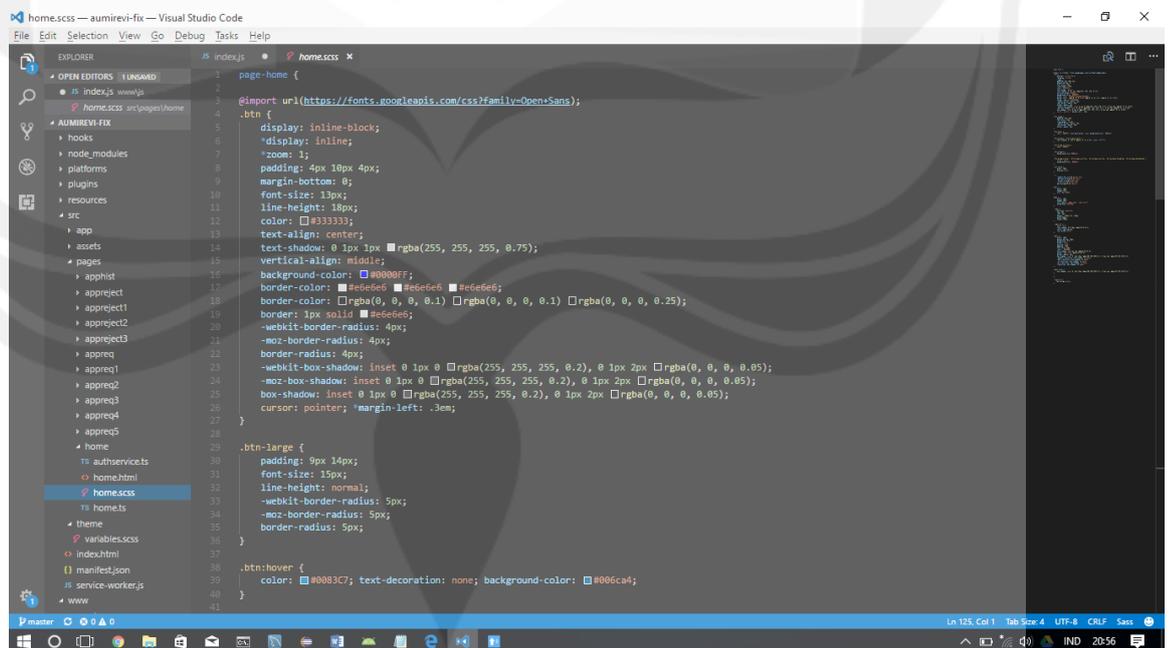
Gambar 1.57 menunjukkan *index* pada ionic yang berisikan *script-script* yang digunakan, seperti *script* untuk tampilan.

2.3.20. Tampilan Login (html & scss)



```
home.html — aumirevi-fix — Visual Studio Code
File Edit Selection View Go Debug Tasks Help
EXPLORER
  JS index.js www/js
  home.html src/pages/home
  AUMIREVI-FIX
  hooks
  node_modules
  platforms
  plugins
  resources
  src
  app
  assets
  pages
  apphit
  appreject
  appreject1
  appreject2
  appreject3
  appreq
  appreq1
  appreq2
  appreq3
  appreq4
  appreq5
  home
  authservices.ts
  home.html
  home.scss
  home.ts
  theme
  variables.scss
  index.html
  manifest.json
  service-worker.js
  www
  OPEN EDITORS
  home.html x
  1 <ion-content>
  2 <body>
  3 <title>Login Form</title>
  4 <div class="login">
  5 
  6 <br>
  7 <form (submit)="doLogin()">
  8 <input [(ngModel)]="loginData.username" name="username" type="text" placeholder="Username"/>
  9 <input [(ngModel)]="loginData.password" name="password" type="password" placeholder="Password"/>
  10 <button type="submit" class="btn btn-primary btn-block btn-large">Login</button>
  11 </form>
  12 </div>
  13 <script src="js/index.js"></script>
  14 </body>
  15 </ion-content>
```

Gambar 1.58. login.html



```
home.scss — aumirevi-fix — Visual Studio Code
File Edit Selection View Go Debug Tasks Help
EXPLORER
  JS index.js www/js
  home.html src/pages/home
  AUMIREVI-FIX
  hooks
  node_modules
  platforms
  plugins
  resources
  src
  app
  assets
  pages
  apphit
  appreject
  appreject1
  appreject2
  appreject3
  appreq
  appreq1
  appreq2
  appreq3
  appreq4
  appreq5
  home
  authservices.ts
  home.html
  home.scss
  home.ts
  theme
  variables.scss
  index.html
  manifest.json
  service-worker.js
  www
  OPEN EDITORS
  home.scss x
  1 page-home {
  2 @import url(https://fonts.googleapis.com/css?family=Open+Sans);
  3 .btn {
  4 display: inline-block;
  5 display: inline;
  6 *zoom: 1;
  7 padding: 4px 10px 4px;
  8 margin-bottom: 0;
  9 font-size: 13px;
  10 line-height: 18px;
  11 color: #333333;
  12 text-align: center;
  13 text-shadow: 0 1px 1px rgba(255, 255, 255, 0.75);
  14 vertical-align: middle;
  15 background-color: #0000ff;
  16 border-color: #666666 #666666 #666666;
  17 border: 1px solid #666666;
  18 border: 1px solid #666666;
  19 border: 1px solid #666666;
  20 -webkit-border-radius: 4px;
  21 -moz-border-radius: 4px;
  22 border-radius: 4px;
  23 -webkit-box-shadow: inset 0 1px 0 rgba(255, 255, 255, 0.2), 0 1px 2px rgba(0, 0, 0, 0.05);
  24 -moz-box-shadow: inset 0 1px 0 rgba(255, 255, 255, 0.2), 0 1px 2px rgba(0, 0, 0, 0.05);
  25 box-shadow: inset 0 1px 0 rgba(255, 255, 255, 0.2), 0 1px 2px rgba(0, 0, 0, 0.05);
  26 cursor: pointer; *margin-left: .3em;
  27 }
  28
  29 .btn-large {
  30 padding: 9px 14px;
  31 font-size: 15px;
  32 line-height: normal;
  33 -webkit-border-radius: 5px;
  34 -moz-border-radius: 5px;
  35 border-radius: 5px;
  36 }
  37
  38 .btn:hover {
  39 color: #0083c7; text-decoration: none; background-color: #006ca4;
  40 }
  41 }
```

Gambar 1.59. login.scss

Gambar 1.58 dan gambar 1.59 menunjukkan *project ionic* untuk tampilan halaman *login*, dimana html berisikan *code* yang digunakan untuk membentuk tampilannya/*user interface* pada *ionic*, *scss* berisikan *code*

untuk warna, *hover*, *line*, posisi pada tampilan, ukuran dari *button*, ukuran dari *box* inputan, *margin*, *effect* dan lain sebagainya yang digunakan pada halaman *login*.

2.3.21. History List

```

12 <ion-col col-10 style="font-size: 25px;color: #white; margin-top:7px; font-family: Lato;">Approval</ion-col>
13 <ion-col col-2 style="color: #white; float:right;">
14 <button ion-button color="danger" icon-only style="float:right; max-height:120px; max-width:80%" (click)="loadlogout()" >
15 <ion-icon name="log-out">/ion-icon>
16 </button>
17 </ion-col>
18 </ion-row>
19 </ion-grid>
20 </ion-toolbar>
21 </ion-header>
22 </ion-content>
23 <div id="background">
24 <ion-segment [(ngModel)]="show" [color]="isAndroid ? 'danger' : 'calm'">
25 <ion-segment-button value="Request" id="fcolor">
26 Request
27 </ion-segment-button>
28 <ion-segment-button value="History" id="fcolor">
29 History
30 </ion-segment-button>
31 </ion-segment>
32 </div>
33 </ion-content>
34 </ion-grid>
35 <ion-row>
36 <div [ngSwitch]="show">
37 <ion-list "ngSwitchCase"="Request">
38 <ion-item "ngOn"let realist of realist; let i = index" (click)="loadappreal(i)" style="min-width: 200px;">
39 <b2>{{realist}}><b>{{realistuser[i]}}><b>Trace Id {{realistid[i]}}><b>{{realistcreatedate[i]}}></b2>
40 </ion-item>
41 </ion-list>
42 </div>
43 </ion-row>
44 </ion-grid>
45 </ion-content>
46 </div>
47 </ion-grid>
48 </ion-content>
49 </ion-grid>
50 </ion-row>
51 </ion-grid>

```

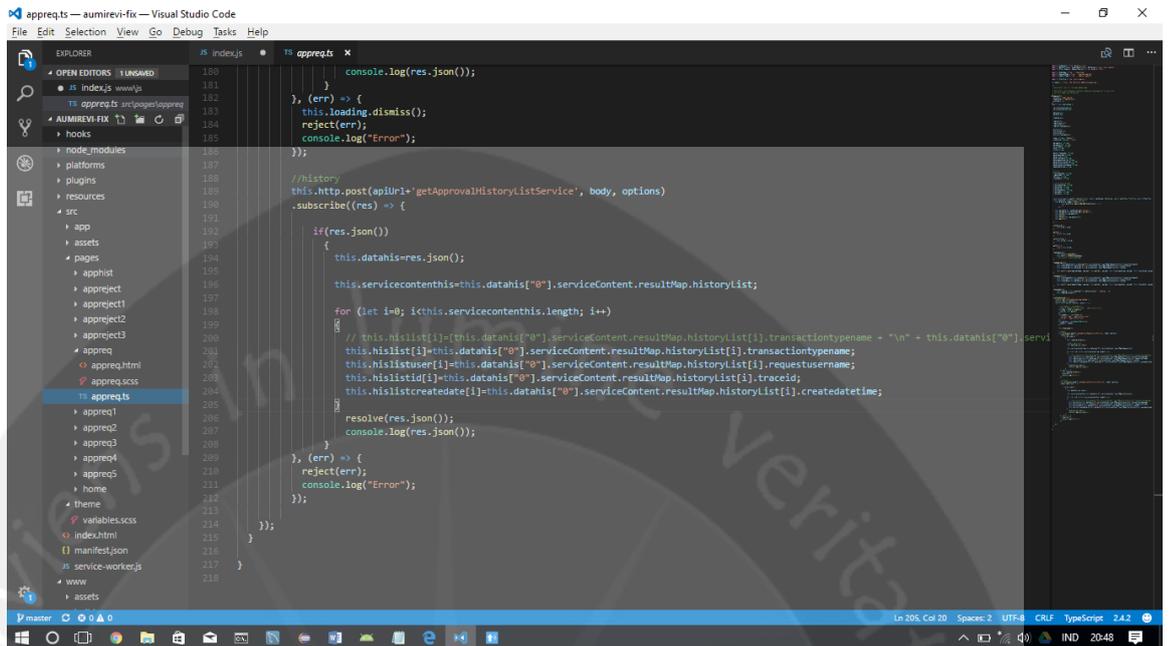
Gambar 1.60. appreq.html

```

1 page-appreq {
2   #btn: hover {
3     opacity: 1.0;
4   }
5   #background {
6     background-color: #87A7E8;
7   }
8   #fcolor {
9     color: #white;
10  }
11  .toolbar-background {
12    background-color: #87A7E8;
13  }
14  .line {
15    border-bottom-color: #white;
16    border-bottom-style: groove;
17  }
18  #button {
19    background-color: #red;
20  }
21  @import url(http://fonts.googleapis.com/css?family=Lato);
22 }

```

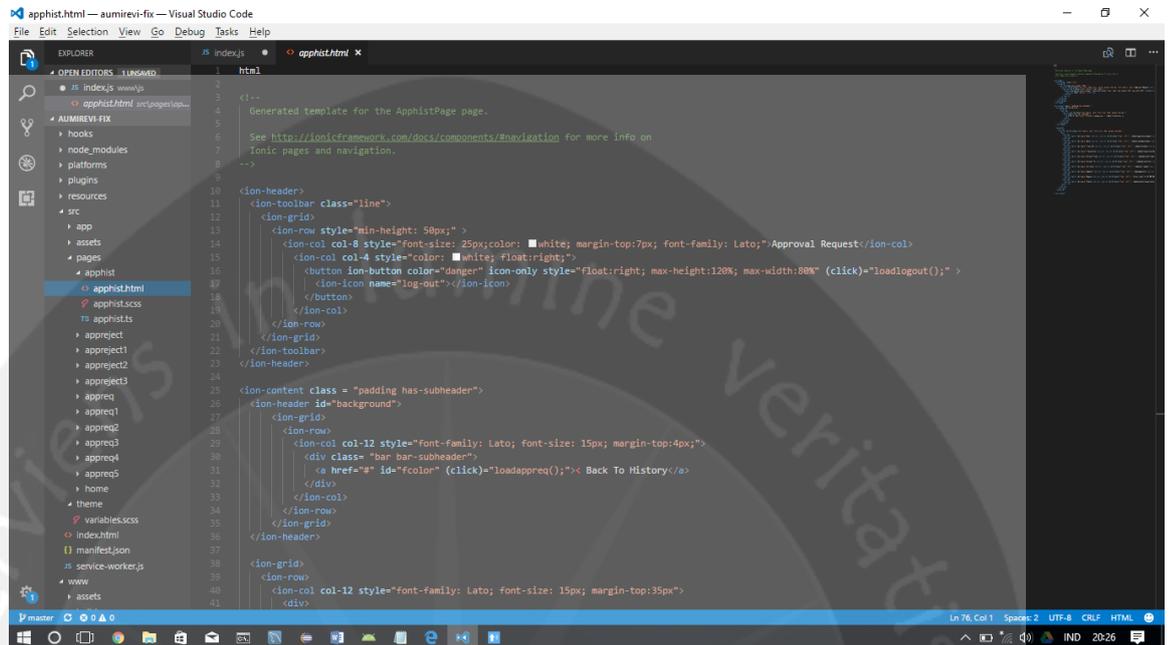
Gambar 1.61. appreq.scss



Gambar 1.62. appreq.ts

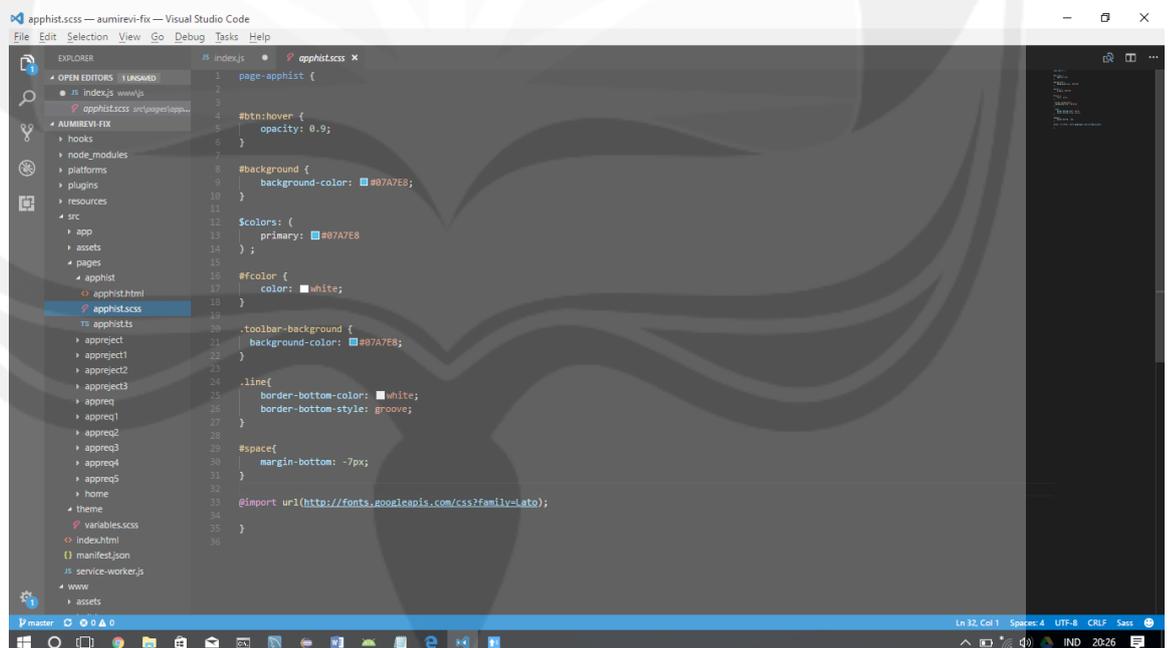
Gambar 1.60 sampai gambar 1.62 menunjukkan *project ionic* untuk tampilan halaman *history list*, dimana html berisikan *code* yang digunakan untuk membentuk tampilannya/*user interface* pada *ionic*, *css* berisikan *code* untuk warna, *hover*, *line*, posisi pada tampilan, dan *ts* berisikan *code logic* untuk menampilkan *list data* dari *history* pada *ionic*.

2.3.22. History Detail



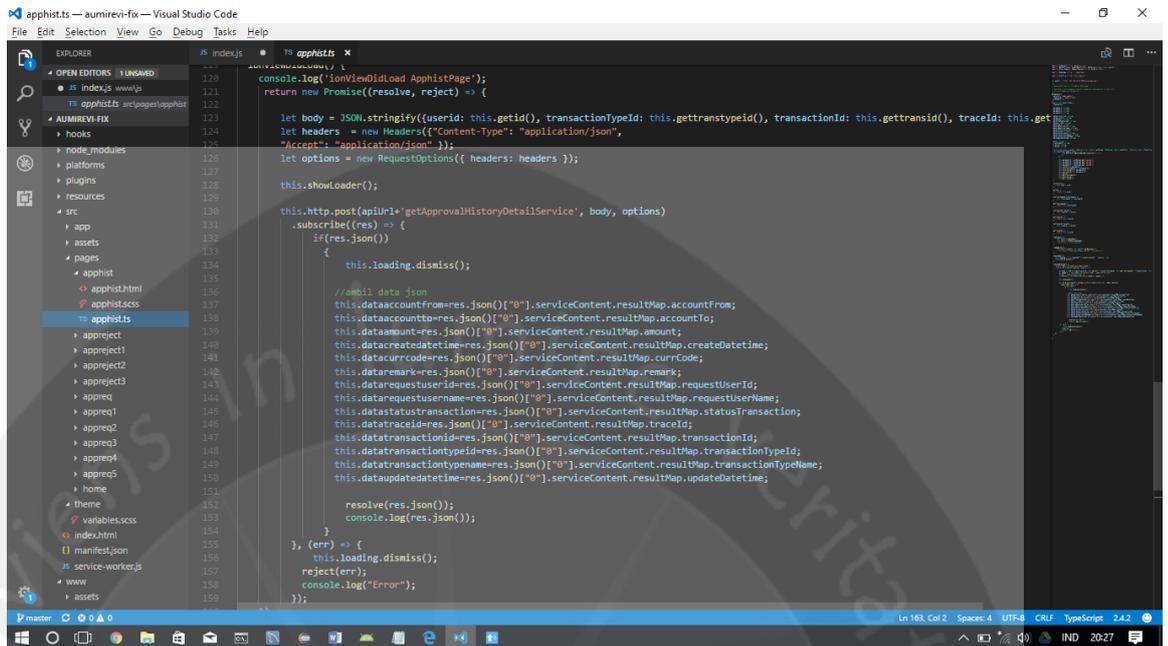
```
1  html
2
3  <!--
4  generated template for the ApphistPage page.
5
6  See http://ionicframework.com/docs/components/#navigation for more info on
7  Ionic pages and navigation.
8  -->
9
10 <ion-header>
11   <ion-toolbar class="line">
12     <ion-grid>
13       <ion-row style="min-height: 50px;" >
14         <ion-col col-8 style="font-size: 25px;color: white; margin-top:7px; font-family: Lato;">Approval Request</ion-col>
15         <ion-col col-4 style="color: white; float:right;">
16           <button ion-button color="danger" icon-only style="float:right; max-height:120px; max-width:80px" (click)="loadlogout();" >
17             <ion-icon name="log-out"></ion-icon>
18           </button>
19         </ion-col>
20       </ion-row>
21     </ion-grid>
22   </ion-toolbar>
23 </ion-header>
24
25 <ion-content class = "padding has-subheader">
26   <ion-header id="background">
27     <ion-grid>
28       <ion-row>
29         <ion-col col-12 style="font-family: Lato; font-size: 15px; margin-top:4px;">
30           <div class= "bar bar-subheader">
31             <a href="#" id="fcolor" (click)="loadappreq();">< Back To History</a>
32           </div>
33         </ion-col>
34       </ion-row>
35     </ion-grid>
36   </ion-header>
37
38   <ion-grid>
39     <ion-col col-12 style="font-family: Lato; font-size: 15px; margin-top:35px">
40       <div>
```

Gambar 1.63. apphist.html



```
1  page-apphist {
2
3    #btn: hover {
4      opacity: 0.9;
5    }
6
7    #background {
8      background-color: #07A7E8;
9    }
10
11    $colors: (
12      primary: #07A7E8
13    );
14
15    #fcolor {
16      color: white;
17    }
18
19    #toolbar-background {
20      background-color: #07A7E8;
21    }
22
23    .line{
24      border-bottom-color: white;
25      border-bottom-style: groove;
26    }
27
28    #space{
29      margin-bottom: -7px;
30    }
31
32    @import url(http://fonts.googleapis.com/css?family=Lato);
33  }
34
35
36
```

Gambar 1.64. apphist.scss

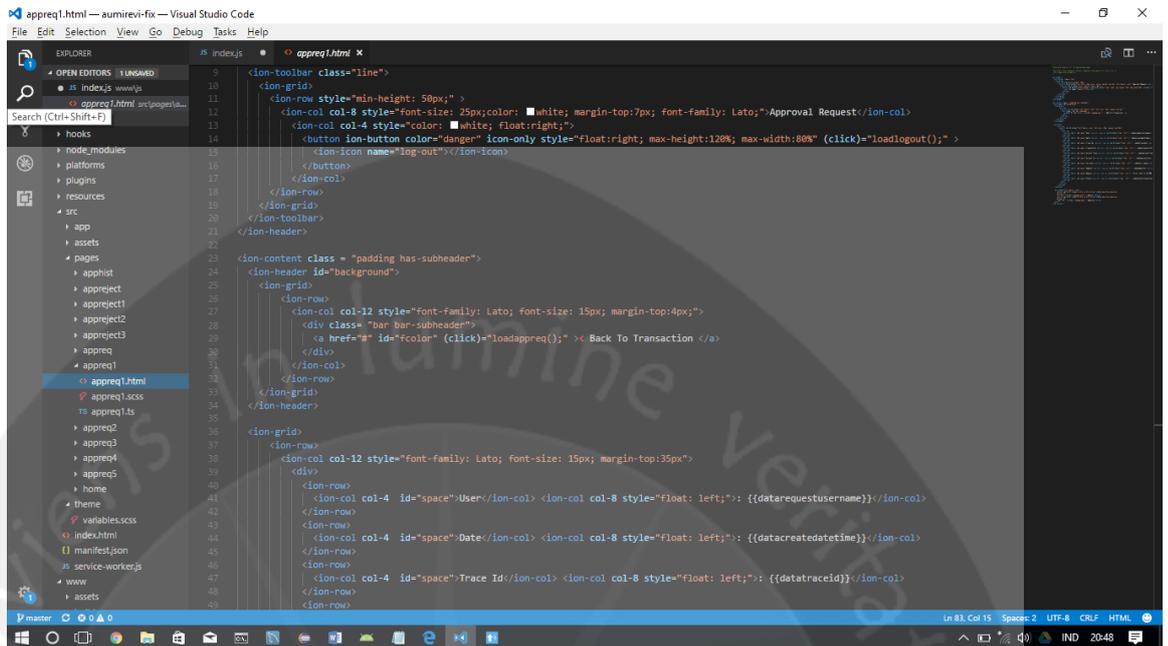


Gambar 1.65. apphist.ts

Gambar 1.63 sampai gambar 1.65 menunjukkan *project ionic* untuk tampilan halaman *history detail*, dimana html berisikan *code* yang digunakan untuk membentuk tampilannya/*user interface* pada *ionic*, *css* berisikan *code* untuk warna, *hover*, *line*, posisi pada tampilan, dan ts berisikan *code logic* untuk menampilkan data detail dari *history* yang telah dipilih *listnya* pada *history list* pada *ionic*.

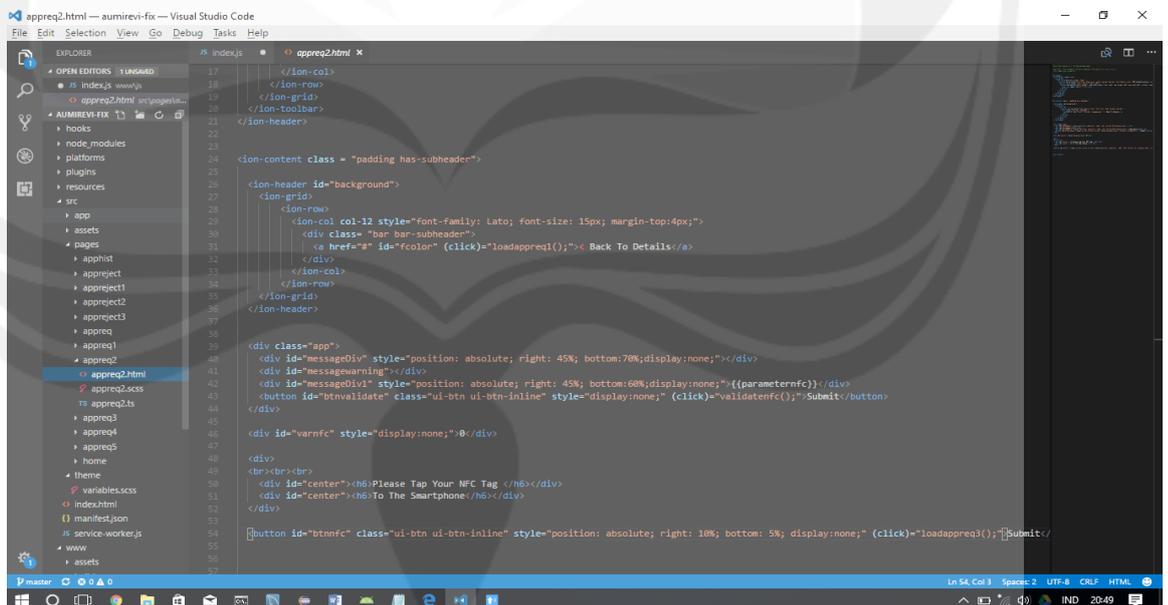
2.3.23. HTML seluruh tampilan ionic

Hypertext Markup Language (HTML) adalah sebuah bahasa markah yang digunakan untuk membuat sebuah halaman web, menampilkan berbagai informasi di dalam sebuah halaman tampilan.



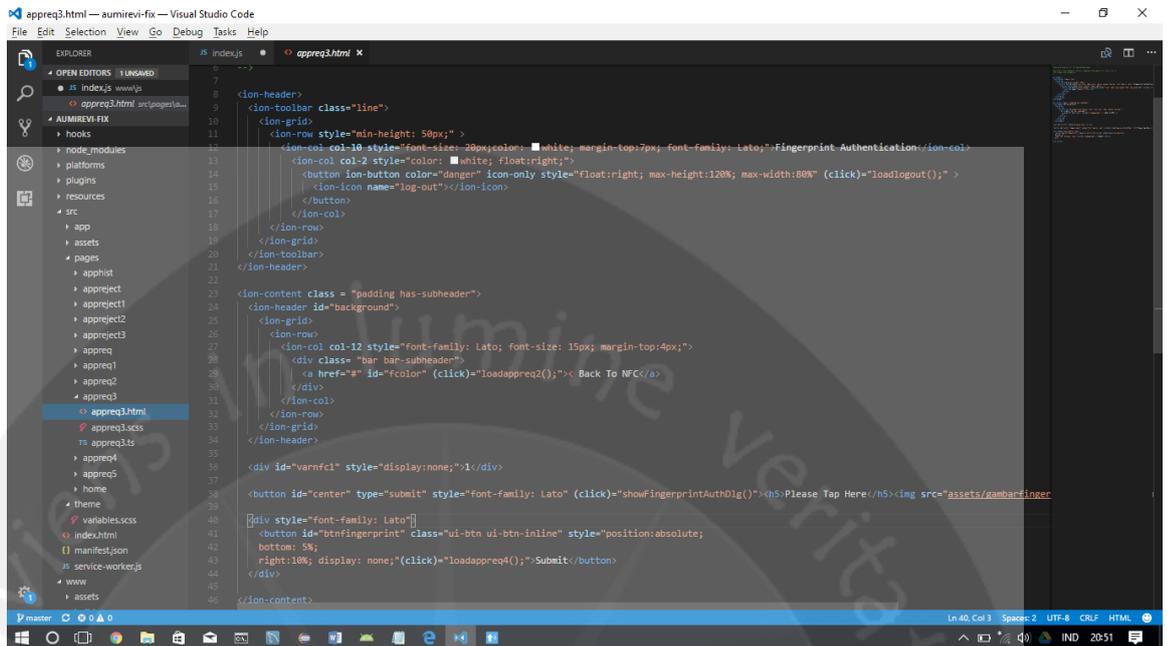
Gambar 1.66. appreq1.html

Gambar 1.66 menunjukkan *code* untuk tampilan detail data dari *request detail*.



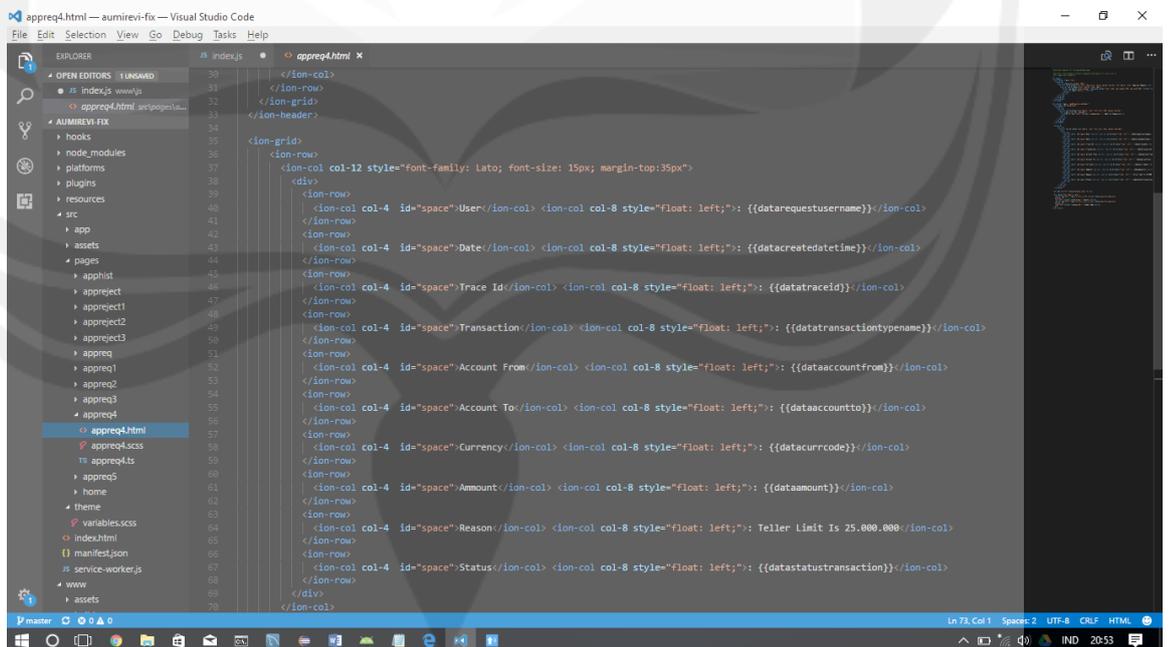
Gambar 1.67. appreq2.html

Gambar 1.67 menunjukkan *code* untuk tampilan *NFC*.



Gambar 1.68. appreq3.html

Gambar 1.68 menunjukkan *code* untuk tampilan *Fingerprint*.



Gambar 1.69. appreq4.html

Gambar 1.69 menunjukkan *code* untuk tampilan data yang akan di *submit*.

```

<ion-row>
  <ion-col col-4 id="space">User</ion-col> <ion-col col-8 style="float: left;"> {{datarequestername}}</ion-col>
</ion-row>
<ion-row>
  <ion-col col-4 id="space">Date</ion-col> <ion-col col-8 style="float: left;"> {{datacreatedatetime}}</ion-col>
</ion-row>
<ion-row>
  <ion-col col-4 id="space">Trace Id</ion-col> <ion-col col-8 style="float: left;"> {{datatraceid}}</ion-col>
</ion-row>
<ion-row>
  <ion-col col-4 id="space">Transaction</ion-col> <ion-col col-8 style="float: left;"> {{datatransactiontype}}</ion-col>
</ion-row>
<ion-row>
  <ion-col col-4 id="space">Account From</ion-col> <ion-col col-8 style="float: left;"> {{dataaccountfrom}}</ion-col>
</ion-row>
<ion-row>
  <ion-col col-4 id="space">Account To</ion-col> <ion-col col-8 style="float: left;"> {{dataaccountto}}</ion-col>
</ion-row>
<ion-row>
  <ion-col col-4 id="space">Currency</ion-col> <ion-col col-8 style="float: left;"> {{datacurrcode}}</ion-col>
</ion-row>
<ion-row>
  <ion-col col-4 id="space">Amount</ion-col> <ion-col col-8 style="float: left;"> {{dataamount}}</ion-col>
</ion-row>
<ion-row>
  <ion-col col-4 id="space">Reason</ion-col> <ion-col col-8 style="float: left;"> Teller Limit Is 25.000.000</ion-col>
</ion-row>
<ion-row>
  <ion-col col-4 id="space">Status</ion-col> <ion-col col-8 style="float: left;"> {{datastatustransaction}}</ion-col>
</ion-row>
</ion-grid>
</div>
<div id="varnfc3" style="display:none;">1</div>
<div style="font-family: Lato">
  <button id="btn1" class="btn btn-primary btn-block btn-large" style="position:absolute; bottom: 5%; left: 25%;(click)=""loadappreq();""Back to Main Menu</button>
</div>

```

Gambar 1.70. appreq5.html

Gambar 1.70 menunjukkan *code* untuk menampilkan data yang telah di *submit*.

```

<ion-toolbar class="line">
  <ion-grid>
    <ion-row style="min-height: 50px;">
      <ion-col col-10 style="font-size: 20px;color: white; margin-top:7px; font-family: Lato;">NFC Authentication Reject</ion-col>
      <ion-col col-2 style="color: white; float:right;">
        <button ion-button color="danger" ion-only style="float:right; max-height:120%; max-width:80%" (click)="loadlogout();" >
          <ion-icon name="log-out"></ion-icon>
        </button>
      </ion-col>
    </ion-row>
  </ion-grid>
</ion-toolbar>
</ion-header>
<ion-content class = "padding has-subheader">
  <ion-header id="background">
    <ion-grid>
      <ion-col col-12 style="font-family: Lato; font-size: 15px; margin-top:4px;">
        <div class= "bar bar-subheader">
          <a href="#" id="fcolor" (click)=""loadappreq();"">Back To Details</a>
        </div>
      </ion-col>
    </ion-grid>
  </ion-header>
  <div class="app">
    <div id="messageDiv1" style="position: absolute; right: 45%; bottom:70%; display:none;"></div>
    <div id="messageWarning"></div>
    <div id="messageDiv1" style="position: absolute; right: 45%; bottom:60%; display:none;">{{parameterfnc}}</div>
    <button id="btinvalidatel" class="ui-btn ui-btn-inline" style="display:none;" (click)=""validatenfc();""Submit</button>
  </div>
  <div id="varnfc" style="display:none;">0</div>
  <div style="font-family: Lato">
    <br><br>
    <div id="center"><h6>Please Tap Your NFC Tag </h6></div>
    <div id="center"><h6>To The Smartphone</h6></div>
  </div>

```

Gambar 1.71. appreject.html

Gambar 1.71 menunjukkan *code* untuk tampilan *NFC* pada proses *reject*.

```
<!--
Generated template for the Appreject1Page page.
See http://ionicframework.com/docs/components/#navigation for more info on
Ionic pages and navigation.
-->
<ion-header>
  <ion-toolbar class="line">
    <ion-grid>
      <ion-row style="min-height: 50px;">
        <ion-col col-10 style="font-size: 22px;color: white;margin-top:7px; font-family: Lato;">Fingerprint Authentication Reject</ion-col>
        <ion-col col-2 style="color: white; float:right;">
          <button ion-button color="danger" icon-only style="float:right; max-height:120%; max-width:80%" (click)="loadLogout();" >
            <ion-icon name="log-out"></ion-icon>
          </button>
        </ion-col>
      </ion-row>
    </ion-grid>
  </ion-toolbar>
</ion-header>
<ion-content class = "padding has-subheader">
  <ion-header id="background">
    <ion-grid>
      <ion-row>
        <ion-col col-12 style="font-family: Lato; font-size: 15px; margin-top:4px;">
          <div class="ban_bar_subheader">
            <a href="#" id="fcolor" (click)="loadAppreject();"> Back To NFC Reject</a>
          </div>
        </ion-col>
      </ion-row>
    </ion-grid>
  </ion-header>
  <div id="varnfc1" style="display:none;">1</div>
  <button id="center" type="submit" style="font-family: Lato; font-size: 15px" (click)="showFingerprintAuthDlg();">Please Tap Here</button>
  <div style="font-family: Lato">
    <button id="btnFingerprint" class="ui-btn ui-btn-inline" style="position:absolute;
```

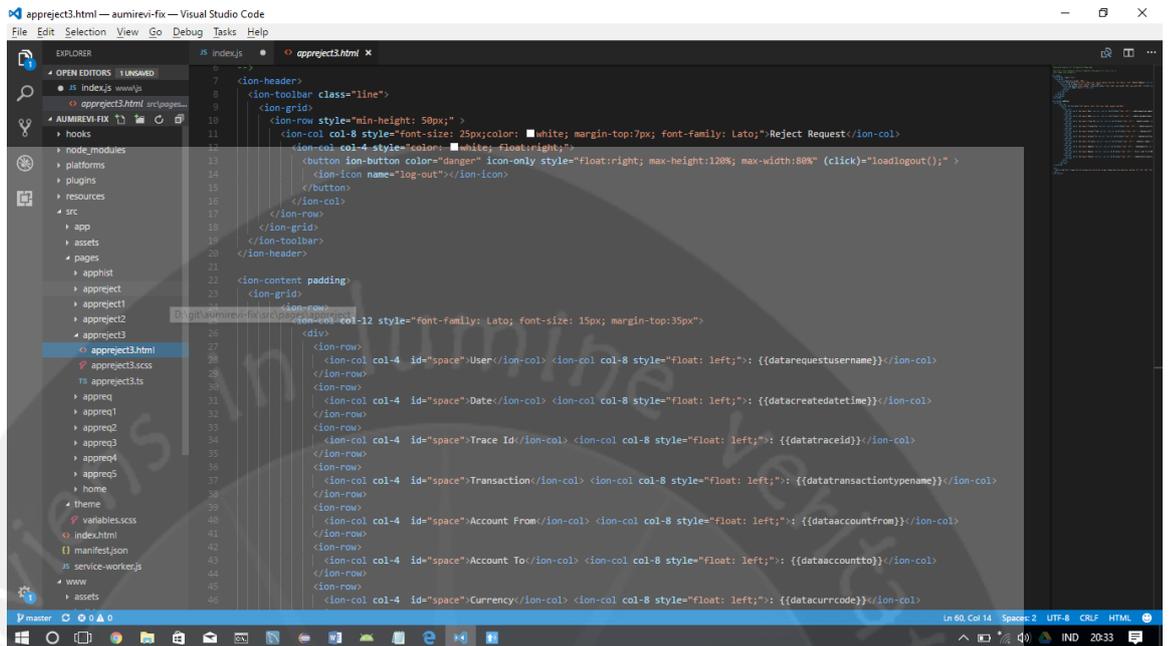
Gambar 1.72. appreject1.html

Gambar 1.72 menunjukkan *code* untuk tampilan *fingerprint* pada proses *reject*.

```
<ion-col col-4 id="space">date</ion-col> <ion-col col-8 style="float: left;">{{datacreatedatetime}}</ion-col>
</ion-row>
<ion-row>
  <ion-col col-4 id="space">Trace Id</ion-col> <ion-col col-8 style="float: left;">{{dataatraceid}}</ion-col>
</ion-row>
<ion-row>
  <ion-col col-4 id="space">Transaction</ion-col> <ion-col col-8 style="float: left;">{{datatransactiontypename}}</ion-col>
</ion-row>
<ion-row>
  <ion-col col-4 id="space">Account From</ion-col> <ion-col col-8 style="float: left;">{{dataaccountfrom}}</ion-col>
</ion-row>
<ion-row>
  <ion-col col-4 id="space">Account To</ion-col> <ion-col col-8 style="float: left;">{{dataaccountto}}</ion-col>
</ion-row>
<ion-row>
  <ion-col col-4 id="space">Currency</ion-col> <ion-col col-8 style="float: left;">{{datacurrcode}}</ion-col>
</ion-row>
<ion-row>
  <ion-col col-4 id="space">Amount</ion-col> <ion-col col-8 style="float: left;">{{dataamount}}</ion-col>
</ion-row>
<ion-row>
  <ion-col col-4 id="space">Reason</ion-col> <ion-col col-8 style="float: left;">Teller Limit Is 25.000.000</ion-col>
</ion-row>
<ion-row>
  <ion-col col-4 id="space">Status</ion-col> <ion-col col-8 style="float: left;">{{datastatustransaction}}</ion-col>
</ion-row>
</div>
<div id="varnfc2" style="display:none;">1</div>
<div style="font-family: Lato">
  <button id="btn1" class="ui-btn ui-btn-inline" style="position:absolute;
bottom: 5%;
left:10%" (click)="showDialog();">Cancel</button>
  <button id="btn2" class="ui-btn ui-btn-inline" style="position:absolute;
bottom: 5%;
right:10%" (click)="loadAppreject3();">Submit</button>
</div>
```

Gambar 1.73. appreject2.html

Gambar 1.73 menunjukkan *code* untuk tampilan *reject request* pada proses *reject*.

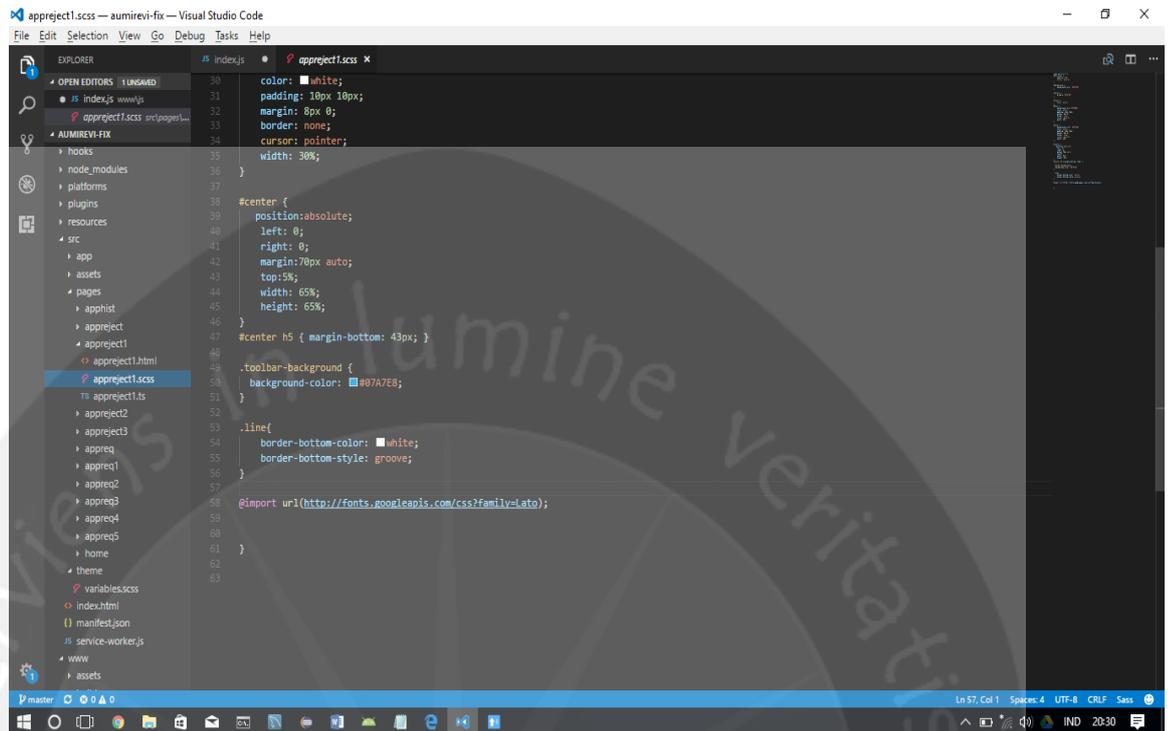


Gambar 1.74. appreject3.html

Gambar 1.74 menunjukkan *code* untuk menampilkan data yang telah di *reject*.

2.3.24. SCSS diseluruh tampilan ionic

SCSS merupakan sintaks yang paling umum digunakan yang merupakan superset dari CSS, yang berarti setiap sintak css yang berada di CSS3 bisa digunakan pada SCSS, tetapi bisa menggunakan fitur seperti yang ada pada SASS.



Gambar 1.75. scss ionic

Gambar 1.75 menunjukkan *code* scss pada ionic, dimana scss yang digunakan pada seluruh halaman hampir sama , dikarenakan menggunakan warna *background* yang sama diseluruhnya dan menggunakan warna tulisan yang sama pada semua *header* atau menggunakan warna dan ukuran yang sama untuk *button*.

BAB 3

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-kritik yang membangun dari mahasiswa-mahasiswa yang melakukan kerja praktek.

3.2. Penerapan Ilmu dalam Kerja Praktek

1. Pratikum Aplikasi Komputer

Penggunaan ilmu dalam Pratikum Aplikasi Komputer yaitu bagaimana menggunakan Microsoft Word dalam pembuatan jurnal dan dokumentasi untuk *project* yang dikerjakan dan dokumentasi *sharing session*.

2. Pratikum Aplikasi Mobile

Penggunaan ilmu dalam Pratikum Aplikasi Mobile yaitu pada saat penulis diminta untuk mengerjakan *project basic* dari android studio dan untuk melihat *log* aplikasi pada saat menjalankan aplikasi.

3. Pemrograman Web

Penggunaan ilmu dalam Pemrograman Web dimana penulis membuat tampilan dari aplikasi menggunakan ionic (html dan scss).

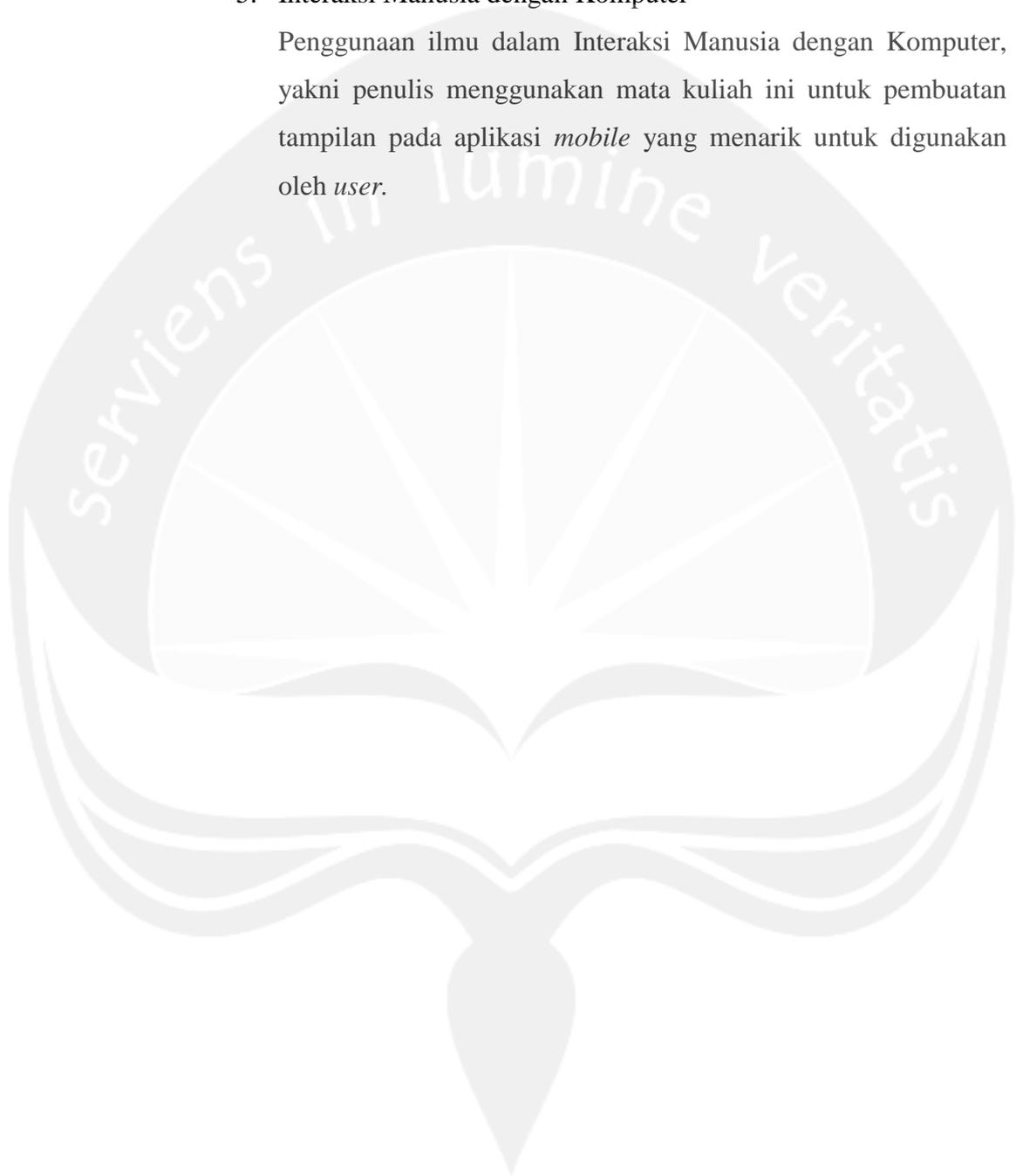
4. Pemrograman Berorientasi Objek

Penggunaan ilmu dalam Pemrograman Berorientasi Objek, yakni penulis menggunakan bahasa java untuk mendvelop aplikasi *mobile* AumiRevi dimana semua data dan fungsi di dibungkus dalam kelas-kelas atau objek-objek dan setiap objek dapat

menerima pesan, memproses data, dan mengirim pesan ke objek lainnya.

5. Interaksi Manusia dengan Komputer

Penggunaan ilmu dalam Interaksi Manusia dengan Komputer, yakni penulis menggunakan mata kuliah ini untuk pembuatan tampilan pada aplikasi *mobile* yang menarik untuk digunakan oleh *user*.



BAB 4

Kesimpulan Dan Saran

Kesimpulan

Setelah penulis melaksanakan Kerja Praktek selama satu bulan lebih di PT. Multipolar Technology Tbk (MLPT), dengan demikian penulis dapat memberi kesimpulan bahwa 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 dan didapatkan ke dunia kerja yang sebenarnya, penulis juga dituntut untuk mampu beradaptasi dengan lingkungan kerja dan mencari pengetahuan tambahan saat menghadapi masalah yang belum pernah ditemui sebelumnya.

Penulis juga mendapatkan pemahaman bahwa sesungguhnya pengetahuan dari universitas masih belum mencukupi untuk digunakan di dunia kerja, penulis harus terus belajar dan mencari pengetahuan – pengetahuan baru untuk menyelesaikan proyek yang dikerjakan. Di sisi lain, penulis juga merasa mendapat bantuan dari mata kuliah yang telah diberikan dan diajarkan oleh dosen sangat membantu dan berguna bagi penulis dalam pelaksanaan kerja praktek. Kerja praktek juga membuat penulis lebih memacu diri dalam membentuk kepribadian yang disiplin dan kooperatif, serta memberikan pengetahuan yang begitu luas tentang pembuatan sebuah aplikasi *mobile* untuk digunakan oleh perusahaan terkhususnya dalam perbankan.

Saran

Sebagai akhir dari penulisan laporan kerja praktek ini, penulis memberikan beberapa saran yang ditujukan kepada mahasiswa yang akan melakukan kerja praktek, jika ingin melakukan kerja praktek yang diwajibkan oleh Universitas maka sebaiknya dilaksanakan secepat-cepatnya jangan sampai membiarkan waktu terbuang percuma karena pada kenyataannya dalam pengajuan proposal kerja praktek juga dapat terjadi persaingan baik itu didalam

Universitas kita sendiri atau di luar Universitas kita, seperti keterbatasan jumlah mahasiswa yang mendaftar dalam suatu perusahaan oleh Universitas dan juga persaingan kita dengan mahasiswa dari Universitas-universitas lain yang ingin mendaftar kerja praktek ke perusahaan tersebut. Begitu juga dalam penyusunan laporan kerja praktek sebaiknya laporan disusun secara bersamaan dengan proses kerja praktek, karena bila laporan terlambat diselesaikan maka akan menyulitkan kita sendiri sebagai penyusun. Penulis juga menyarankan, ketika mencari perusahaan untuk pelaksanaan kerja praktek sebaiknya mencari perusahaan yang infrastrukturnya kuat dan bidangnya sesuai dengan yang kita sukai, serta dukungan dan komunikasi dari kampus diharapkan lebih ditingkatkan karena kerja praktek merupakan langkah awal bagi mahasiswa untuk mengenal dunia kerja secara langsung dengan membawa nama kampus.

Lampiran

1. Surat Keterangan Penerimaan 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

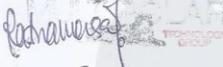
Dengan hormat,

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 Panjaitan
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

2. Surat Keterangan Penyelesaian dari Perusahaan



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

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

3. Surat Penilaian dari Perusahaan

FORM PENILAIAN KERJA PRAKTEK

Nama Mahasiswa : Indah Meliani Malesta Sihombing
NIM : 140707911
Tempat Pelaksanaan : PT. Multipolar Technology Tbk.
Waktu Pelaksanaan : 06 Juli 2017 - s/d 18 Agustus 2017

NO	ASPEK YANG DINILAI	NILAI (0-100)
1	Kemampuan Teknis di Bidang IT	97
2	Kemampuan Bekerja Sama dalam Tim	99
3	Penempatan Diri dalam Lingkungan Kerja	100
4	Kedisiplinan	100
RATA-RATA :		99

Komentar :
Kemampuan membaca error log dan develop UI sudah cukup bagus, tapi harus lebih ditingkatkan agar dapat membunt nilai jual dan kenyamanan penggunaan aplikasi lebih baik. Harus lebih sabar dalam bug tracing. Kedisiplinan dan kerjasama dalam tim harus dipertahankan saat masuk dunia kerja yang sebenarnya nanti!

Nama Pembimbing : Hans Kristianto
Posisi/Jabatan : Application Developer (SIBVR) / Project Leader (Aumi Ravi)
No. Handphone : 0878 89128755
Alamat Email : hans.kristianto@multipolar.com

Tangerang, 18 Agustus 2017
Pembimbing Lapangan,

(...Hans Kristianto.....)