

**LAPORAN KERJA PRAKTEK**  
**SISTEM ABSENSI**  
**DI PT. XYZ**



**Dipersiapkan oleh :**

**Paulina Sudharman / 140707789**

**Program Studi Teknik Informatika**

**Fakultas Teknologi Industri**

**Universitas Atma Jaya Yogyakarta**

**2017**

**HALAMAN PENGESAHAN**  
**Laporan Kerja Praktek**

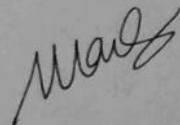
**Sistem Absensi**  
**PT. XYZ**

Laporan ini telah diperiksa dan disetujui  
pada tanggal, 31 Maret 2017

Oleh:

Dosen Pembimbing

Pembimbing Lapangan



**Martinus Maslim, S.T., M.T.**

**Nicolas Willianto Widjaja S. Kom**

## KATA PENGANTAR

Puji syukur kepada Tuhan Yang Maha Esa atas kasih dan karunia-Nya sehingga laporan pelaksanaan kerja praktek yang dilakukan penulis di PT. XYZ ini dapat tersusun.

Kerja Praktek ini merupakan salah satu mata kuliah yang wajib ditempuh di Program Studi Teknik Informatika Universitas Atma Jaya Yogyakarta. Laporan kerja praktek ini disusun sebagai laporan hasil kerja praktek penulis selama 30 hari di Divisi *Information Technology*.

Dengan selesainya laporan kerja praktek ini tidak terlepas dari bantuan banyak pihak yang telah memberikan masukan-masukan kepada penulis. Untuk itu penulis mengucapkan banyak terima kasih kepada:

1. Tuhan Yesus Kristus, yang selalu mencurahkan berkat, kasih karunia, anugerah dan pengetahuan-Nya kepada penulis.
2. Orang tua, dan adik penulis yang telah memberikan doa, semangat, dukungan, dan motivasi selama melakukan studi.
3. Bapak Martinus Maslim, selaku Dosen Pembimbing Kerja Praktek.
4. Bapak Andy Jaya, selaku Manager *Software Development 1*.
5. Ko Nicolas Willianto dan Ci Yunita Chandra, selaku Pembimbing Lapangan Kerja Praktek.
6. Para *staff* PT. XYZ yang telah menerima penulis dengan baik selama kerja praktek.
7. Kepada semua pihak yang telah berkenan memberikan bantuan dan dukungan serta kerja sama yang baik sehingga laporan ini dapat diselesaikan dengan lancar.

Penulis menyadari bahwa laporan kerja praktek ini masih jauh dari sempurna karena keterbatasan waktu dan pengetahuan yang dimiliki penulis. Oleh sebab itu,

segala kritik dan saran yang membangun sangat penulis harapkan. Akhir kata, semoga laporan kerja praktek ini dapat berguna dan bermanfaat bagi semua pihak.

Yogyakarta, 31 Maret 2017

Penulis,

Paulina Sudharman



## DAFTAR ISI

LAPORAN KERJA PRAKTEK.....	1
HALAMAN PENGESAHAN.....	2
KATA PENGANTAR .....	3
DAFTAR ISI.....	5
DAFTAR GAMBAR .....	6
BAB I.....	8
1.1. Sejarah Perusahaan.....	8
1.2. Sekilas Perusahaan .....	8
1.3. Visi, Misi dan Tujuan Perusahaan.....	9
1.4. Struktur Organisasi.....	9
1.5. Deskripsi Tugas Struktur Organisasi.....	9
1.6. Departemen TI dalam Perusahaan.....	14
BAB II.....	17
2.1 Penjelasan Logbook .....	17
2.2 Hasil Pekerjaan Secara Umum .....	46
2.3 Bukti Hasil Pekerjaan .....	47
2.3.1 Membuat Basis Data .....	47
BAB III .....	54
3.1 Manfaat Kerja Praktek.....	54
3.2 Penerapan Ilmu dalam Kerja Praktek .....	54
BAB IV .....	57
4.1 Kesimpulan.....	57

## DAFTAR GAMBAR

Gambar 1.1	Bagan Struktur Organisasi secara Umum .....	9
Gambar 1.2	Bagan Struktur Organisasi: Information Technology.....	14
Gambar 2.1.1	Tampilan PhpMyAdmin .....	18
Gambar 2.1.2	Tampilan MySQL Workbench .....	19
Gambar 2.1.3	Tampilan Data User .....	19
Gambar 2.1.4	Tampilan Data Toko .....	20
Gambar 2.1.5	Pembuatan Controller dan Model .....	21
Gambar 2.1.6	Kalender Absensi .....	22
Gambar 2.1.7	Tampilan Modal untuk Membuat Jadwal .....	22
Gambar 2.1.8	Foto Bersama saat Company Visit .....	23
Gambar 2.1.9	Modal untuk Membuat Jadwal .....	24
Gambar 2.1.10	Tampilan Kalender dengan View Hari .....	25
Gambar 2.1.11	Tampilan Tambah Toko .....	26
Gambar 2.1.12	Tampilan Login Pengguna .....	27
Gambar 2.1.13	Tampilan Register Pengguna .....	27
Gambar 2.1.14	Tampilan Data Pengguna .....	28
Gambar 2.1.15	Tampilan Aplikasi pada Android .....	29
Gambar 2.1.16	Tampilan Kelas Model Cabang di Android .....	30
Gambar 2.1.17	Tampilan Routing API .....	30
Gambar 2.1.18	Tampilan Fungsi Login di Android .....	31
Gambar 2.1.19	Tampilan Detil Pengguna .....	32

Gambar 2.1.20	Tampilan Kalender .....	34
Gambar 2.1.21	Fungsi Pembuat Marker .....	36
Gambar 2.1.22	Tampilan Detil Jadwal .....	37
Gambar 2.1.23	Tampilan Kalender Sebelum Dirubah .....	38
Gambar 2.1.24	Tampilan Kalender Setelah Dirubah .....	38
Gambar 2.1.25	Tampilan Basis Data untuk Tabel Tugas .....	40
Gambar 2.1.26	Tampilan Tambah Jadwal .....	41
Gambar 2.1.27	Tampilan Tampil Jadwal .....	41
Gambar 2.1.28	Tampilan Edit Jadwal .....	42
Gambar 2.1.29	Tampilan Tampil Data User .....	43
Gambar 2.1.30	Drop Down List Pilih Toko .....	44
Gambar 2.1.31	Drop Down List Pilih User .....	45
Gambar 2.3.1	MySQL Workbench di PT. XYZ .....	47
Gambar 2.3.2	Tampilan Kalender pada Website .....	48
Gambar 2.3.3	Tampilan Jadwal pada Website .....	48
Gambar 2.3.4	Tampilan Penambahan Jadwal dan Tugas .....	49
Gambar 2.3.5	Code API untuk Fungsi Login pada Android .....	50
Gambar 2.3.6	Tampilan Login pada Siabo .....	51
Gambar 2.3.7	Tampilan Kalender dan Jadwal pada Siabo .....	51
Gambar 2.3.8	Tampilan Detil Jadwal pada Siabo .....	52
Gambar 2.3.9	Google Developer Console .....	53

# **BAB I**

## **PENDAHULUAN**

### **1.1. Sejarah Perusahaan**

Berawal dari pemikiran untuk mempermudah penyediaan kebutuhan pokok sehari-hari karyawan, maka PT. XYZ mendirikan sebuah gerai. Sejalan pengembangan operasional toko, perusahaan tertarik untuk lebih mendalami dan memahami berbagai kebutuhan dan perilaku konsumen dalam berbelanja. Guna mengakomodasi tujuan tersebut, beberapa orang karyawan ditugaskan untuk mengamati dan meneliti perilaku belanja masyarakat. Kesimpulan yang didapat adalah bahwa masyarakat cenderung memilih belanja di gerai modern berdasarkan alasan kelengkapan pilihan produk yang berkualitas, harga yang pasti dan bersaing, serta suasana yang nyaman.

Pada mulanya dibentuk konsep penyelenggaraan gerai yang berlokasi di dekat hunian konsumen, menyediakan berbagai kebutuhan pokok maupun kebutuhan sehari-hari, melayani masyarakat umum yang bersifat majemuk, serta memiliki luas toko sekitar 200 m<sup>2</sup>. Seiring dengan perjalanan waktu dan kebutuhan pasar, gerai terus bertambah di berbagai kawasan perumahan, perkantoran, niaga, wisata, dan apartemen.

### **1.2. Sekilas Perusahaan**

PT. XYZ adalah jaringan minimarket yang menyediakan kebutuhan pokok dan kebutuhan sehari-hari. Saat ini PT. XYZ berkembang sangat pesat dengan jumlah gerai mencapai lebih dari 10.000 gerai dengan pasokan barang dagangan dari beberapa distributor yang menyediakan ribuan jenis produk.



### 1.3. Visi, Misi dan Tujuan Perusahaan

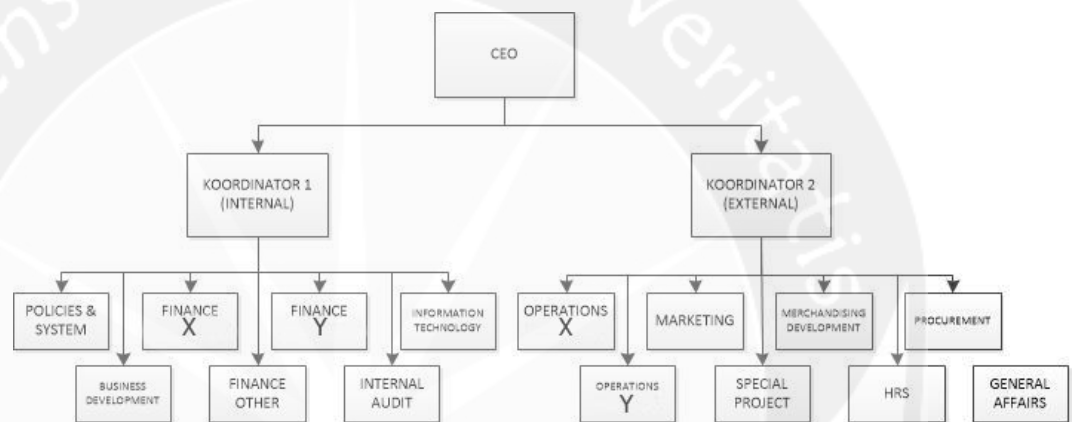
Visi PT. XYZ

- Menjadi perusahaan minimarket modern yang terbesar di Indonesia.

Misi PT. XYZ

- Meningkatkan kualitas layanan kepada konsumen terus menerus

### 1.4. Struktur Organisasi



**Gambar 1.1 Bagan Struktur Organisasi secara Umum**

Gambar 1.1 merupakan bagan struktur organisasi secara umum di PT. XYZ. Di bawah CEO PT. XYZ terdapat 2 koordinator yang bertanggung jawab atas kegiatan departemen-departemen yang ada di PT. XYZ. Di bawah koordinator tersebut, terdapat 15 departemen yang berhubungan dengan pihak internal dan pihak eksternal.

### 1.5. Deskripsi Tugas Struktur Organisasi

#### 1.5.1 Department of Policies and System

Departemen *policies and system* bertugas untuk mengelola peraturan-peraturan yang berjalan di perusahaan, agar peraturan dapat dijalankan dengan baik. Sistem yang berlaku di perusahaan akan dikontrol oleh departemen ini.

### **1.5.2 Department of Business Development**

Departemen bisnis bertugas untuk mengatur proses bisnis yang ada. Departemen bisnis akan menciptakan bisnis maupun produk baru, sehingga produk maupun bisnis tersebut dapat mengikuti perkembangan bisnis yang semakin maju.

### **1.5.3 Department of Finance Toko X**

Departemen keuangan Toko X akan bertugas untuk mengatur keuangan yang ada di toko X. Karena toko X berjumlah lebih dari 10.000, maka bagian keuangan X dikelola terpisah dengan jenis usaha milik PT. XYZ yang lain.

### **1.5.4 Department of Finance Toko Y**

Bagian keuangan Toko Y pun dikelola secara terpisah, dengan bisnis lainnya. Gerai Toko Y merupakan gerai berskala besar karena Toko Y merupakan pusat perkulakan, sehingga diperlukan departemen keuangan yang khusus mengelola keuangan Toko Y.

### **1.5.5 Department of Finance Other**

Jenis usaha yang dimiliki oleh PT. XYZ tidak hanya Toko X dan Toko Y saja, tetapi ada banyak bidang bisnis lainnya.

### **1.5.6 Department of Internal Audit**

Departemen audit memiliki tugas untuk memeriksa detil keuangan secara menyeluruh dari PT. XYZ, mencakup pemeriksaan transaksi keluar dan transaksi masuk. Departemen ini pula yang menyiapkan laporan Laba Rugi, dan memecahkan masalah keuangan. Semua departemen keuangan akan berhubungan langsung dengan departemen audit.

### **1.5.7 Department of Information Technology**

Departemen IT memiliki tugas untuk merencanakan dan mengembangkan semua sistem sesuai kebutuhan perusahaan, memelihara sistem informasi perusahaan yang terintegrasi dan meningkatkan kinerja perusahaan. Bagian IT terbagi kedalam 8 bagian yang mengelola Toko X, Toko Y, dan jenis badan usaha yang lainnya. Departemen IT akan mengembangkan *hardware* yang sudah dikembangkan oleh departemen *special project*.

### **1.5.8 Department of Operations Toko X**

Bertugas untuk mengatur operasional Toko X, seperti pada area supervisor. Departemen ini akan mengatur karyawan yang berada pada level kepala cabang di Toko X. Departemen ini pula yang akan membuat standar proses operasional, membuat strategi untuk mencapai suatu target.

### **1.5.9 Department of Operations Toko Y**

Bertugas untuk mengatur operasional Toko Y, seperti pada area supervisor. Departemen ini akan mengatur karyawan yang berada pada level kepala cabang di Toko Y. Departemen ini pula yang akan membuat standar proses operasional, membuat strategi untuk mencapai suatu target.

### **1.5.10 Department of Marketing**

Bagian ini bertugas untuk membuat penawaran produk kepada pelanggan, seperti pengadaan *event* di toko-toko Toko X, Toko O, dan Toko C, serta gerai Toko Y.

### **1.5.11 Department of Special Project**

Departemen Special Project bertugas untuk merancang custom *hardware* yang dibutuhkan oleh perusahaan. Departemen ini akan

membuat *hardware* yang akan diprogram oleh Departemen IT, sehingga dapat digunakan oleh anak perusahaan seperti Toko X, dan Toko Y.

#### **1.5.12 Department of Merchandising Development**

Departemen Merchandiser akan bertugas untuk mengatur *display* etalase toko, agar dapat memudahkan pembeli menemukan barang yang dibutuhkan. Dengan pengaturan *display* etalase akan meningkatkan penjualan yang ada di toko, karena dapat menarik pembeli untuk membeli barang yang sudah tertata rapi.

#### **1.5.13 Department of Human Resources and Service**

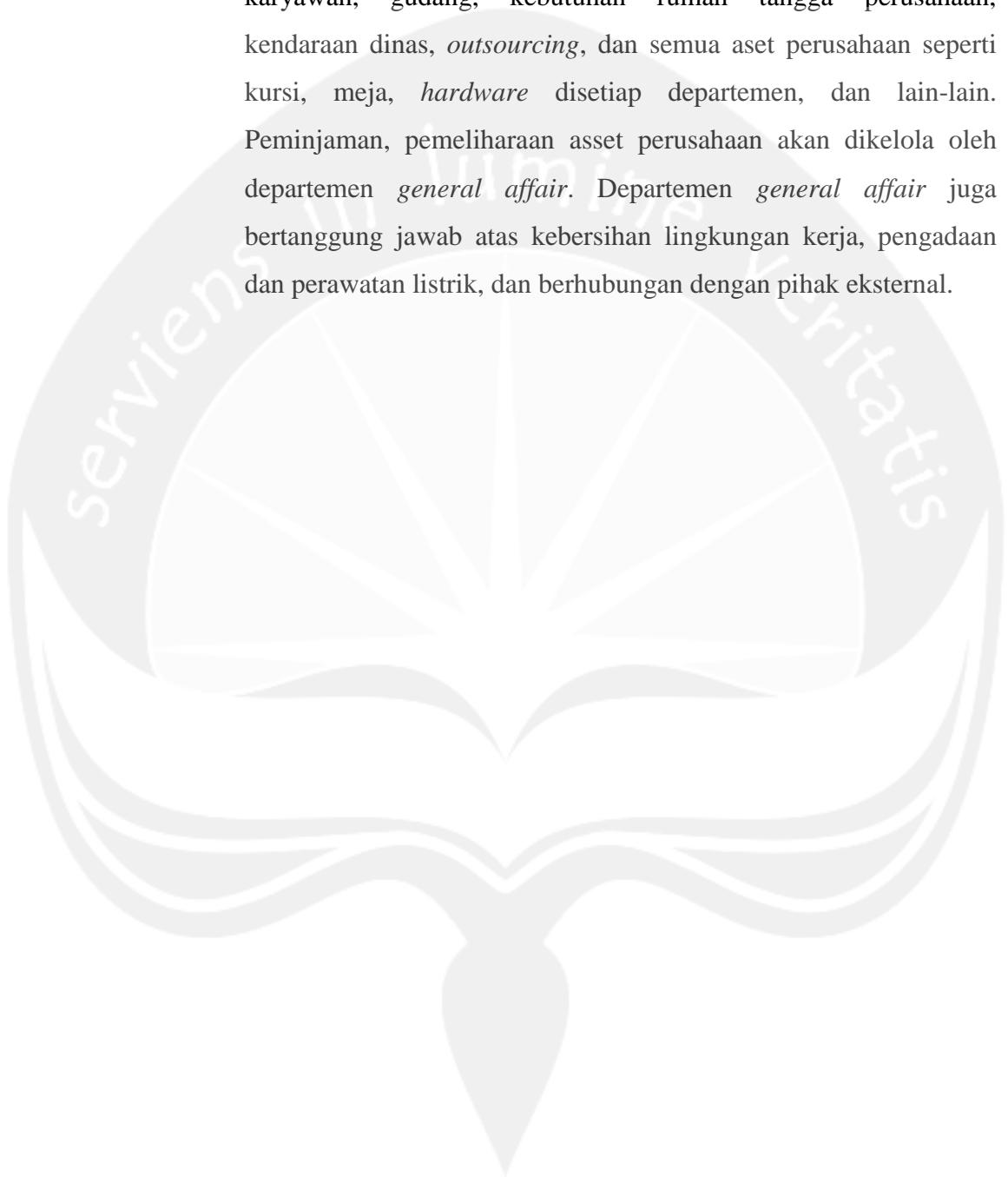
Departemen HRS bertanggung jawab secara penuh dalam sumber daya manusia diperusahaan, mulai dari perekrutan pegawai baru, dan kontrak kerjanya. Departemen HRS akan memetakan SDM yang sudah diterima diperusahaan sesuai dengan kemampuan karyawan dan kebutuhan perusahaan. Absensi karyawan, perhitungan gaji, bonus, tunjangan, menindak karyawan yang melanggar peraturan akan dilakukan oleh departemen ini, sehingga dapat sesuai dengan standar perusahaan.

#### **1.5.14 Department of Procurement**

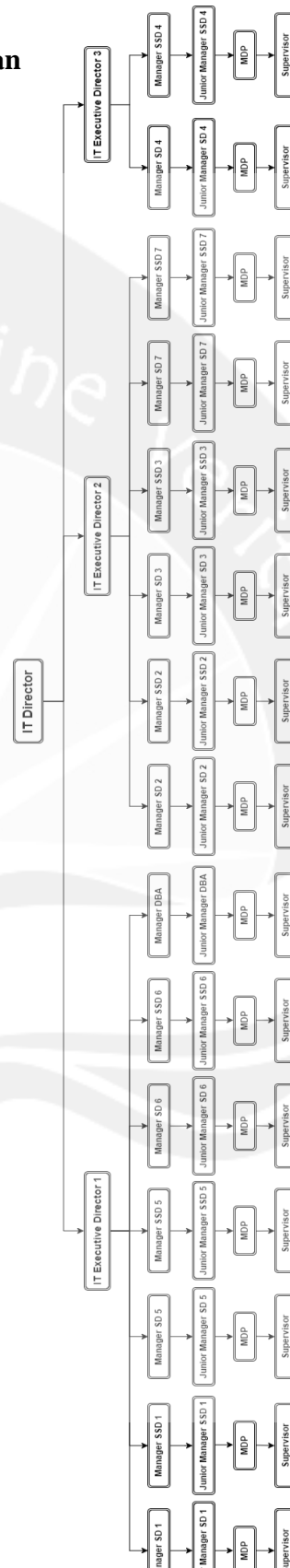
Departemen yang mengelola perusahaan barang yang dibutuhkan oleh perusahaan, agar proses bisnis dapat berjalan dengan lancar. Departemen Procurement yang akan berhubungan dengan *supplier*, mulai dari pemilihan *supplier*, negosiasi harga, memesan barang yang dibutuhkan dan mengatur pembelian rutin, serta mengatur kinerja *supplier*.

### **1.5.15 Department of General Affairs**

Department yang mengatur perawatan gedung, lahan parkir untuk karyawan, gudang, kebutuhan rumah tangga perusahaan, kendaraan dinas, *outsourcing*, dan semua aset perusahaan seperti kursi, meja, *hardware* disetiap departemen, dan lain-lain. Peminjaman, pemeliharaan aset perusahaan akan dikelola oleh departemen *general affair*. Departemen *general affair* juga bertanggung jawab atas kebersihan lingkungan kerja, pengadaan dan perawatan listrik, dan berhubungan dengan pihak eksternal.



## 1.6. Departemen TI dalam Perusahaan



Gambar 1.2 Bagan Struktur Organisasi: *Information Technology*

Gambar 1.2 merupakan bagan struktur organisasi bagian *Information Technology* yang mendukung kelangsungan teknologi yang ada di PT. XYZ. Terdapat 3 *Executive Director* yang membawahi 8 departemen. SD merupakan *Software Development* yang mengembangkan aplikasi yang akan digunakan oleh pihak internal maupun pihak eksternal PT. XYZ. SSD merupakan *Support Software Development* yang akan mendukung aplikasi yang digunakan oleh perusahaan, seperti *testing* atau menerima komplain dari pengguna.

Divisi IT terbagi menjadi 8 sub-divisi yaitu:

- **Software Development dan Support Software Development 1** yang bertanggung jawab atas POS Toko Y, *Back Office* Toko Y, POS Toko O, *Back Office* Toko O.
- **Software Development dan Support Software Development 2** yang bertanggung jawab atas Aplikasi HR dan *Payroll*.
- **Software Development dan Support Software Development 3** yang bertanggung jawab atas *Distribution Center (DC)*, dan *Logistik Head Office (HO)*.
- **Software Development dan Support Software Development 4** yang bertanggung jawab atas *Merchandising* Toko X, *Sewa Tempat Pemanjangan (STP)*, dan *Planogram* Toko X.
- **Software Development dan Support Software Development 5** yang bertanggung jawab atas *Arsip Legal*, *General Affair*, *Logistik IT*, *Procurement dan Budget*, *RTE* dan *Restoran*, *Building Management*, *Teknik Operational Bengkel*, *Operational B Plaza*.
- **Software Development dan Support Software Development 6** yang bertanggung jawab atas *Finance Accounting*, *Pajak ADP* dan *ACL*, *Finance Accounting Pajak Aset* dan *Burger B*, *Kuangan Mister A*, *Kuangan Mister B*, *Financial Accounting Toko Y (FAD)*
- **Software Development dan Support Software Development 7** yang bertanggung jawab atas *Aplikasi Toko Toko X*, *Back End* Toko X, *POS Mister A*.

- **IT Database Administrator** yang bertanggung jawab atas Basis Data Toko X, Toko Y, dan Mister. B.





## **BAB II**

### **PELAKSANAAN KERJA PRAKTEK**

#### **2.1 Penjelasan Logbook**

Berikut adalah penjelasan kegiatan yang saya kerjakan di PT. XYZ dari tanggal 4 Januari 2017 sampai dengan 14 Februari 2017.

##### **4 Januari 2017, Head Office PT. XYZ**

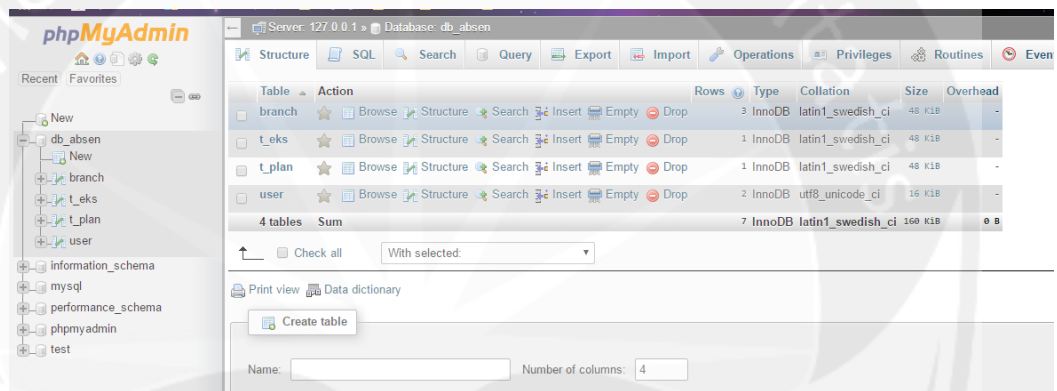
Merupakan hari pertama kerja praktek di PT. XYZ. *Head Office* PT. XYZ memiliki jam kerja dari hari Senin hingga hari Jum'at, dimulai pukul 08.00 WIB sampai dengan 17.00 WIB. Untuk hari Jum'at jam pulang kerja kantor adalah pukul 17.30 WIB karena perpanjangan waktu untuk karyawan yang akan melaksanakan sholat Jum'at.

Pada hari pertama saya perkenalan dengan Bapak Andy selaku *Manager Software Development 1*, tempat saya ditempatkan dan staff di *Software Development (SD) 1*, dan yang merupakan pembimbing lapangan saya Ko Nicolas dan Ci Yunita. *Software Development 1 (SD 1)* merupakan divisi yang bertanggung jawab dengan POS Toko Y, *Back Office* Toko Y, POS Toko O, *Back Office* Toko O.

Setelah berkenalan, saya diberi proyek oleh Pak Andy untuk membuat aplikasi absensi untuk karyawan lapangan Toko O, proyek ini adalah proyek yang saya kerjakan selama kerja praktek. Proyek yang diberikan ke saya berbasis platform web untuk *back-end* dan *front-endnya*, dan Android untuk aplikasinya. Program yang saya buat bertujuan untuk membantu karyawan dari PT. XYZ berkunjung ke minimarket Toko O, karyawan tersebut diharuskan untuk melaporkan kehadirannya dan mengerjakan tugas yang diberikan oleh atasannya menggunakan aplikasi tersebut. Sistem ini merupakan berbasis lokasi, untuk mengakses peta yang disediakan di dalam aplikasi dalam diakses

ketika pengguna mengaktifkan GPSnya. Dengan adanya aplikasi ini, karyawan yang ditugaskan untuk berkunjung ke Toko O tidak perlu melaporkan kehadirannya secara manual, cukup absen dari aplikasi tersebut

Untuk hari ini, saya merancang basis data yang diperlukan dalam sistem tersebut. Kemudian, saya membuat basis data yang sudah disetujui dengan Ko Nico menggunakan PhpMyAdmin. Untuk data toko, saya diberi 20 data toko untuk digunakan sebagai data *dummy* dalam basis data saya.



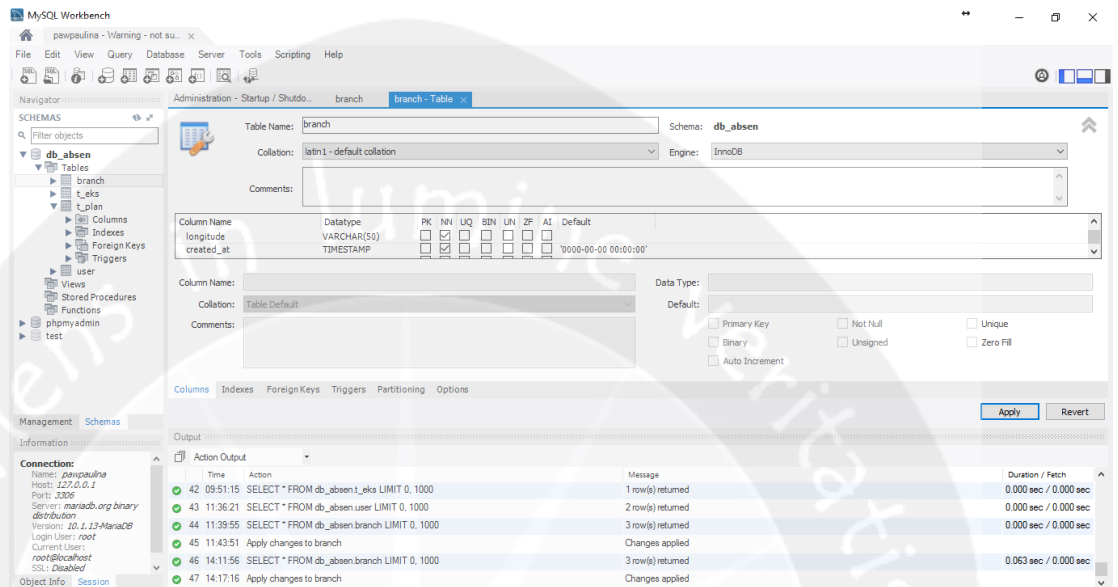
**Gambar 2.1.1 Tampilan PhpMyAdmin**

Gambar 2.1.1 merupakan tampilan dari PhpMyAdmin yang digunakan untuk mengelola basis data Siabo. Tabel yang sudah saya buat adalah tabel cabang, tabel eksekusi yang berguna untuk menyimpan data setelah pengguna absen, tabel plan untuk menyimpan data jadwal, dan tabel user/pengguna.

### **5 Januari 2017, Head Office PT. XYZ**

Pekerjaan yang saya lakukan hari ini adalah instalasi Laravel 5.3 yang kemudian diturunkan ke Laravel 5.1, karena *framework* yang digunakan di SD 1 adalah Laravel 5.1. Instalasi Laravel 5.1 saat ini saya menggunakan Composer. Sembari melakukan instalasi, saya memindahkan basis data yang saya buat di PhpMyAdmin ke MySQL Workbench, karena di PT. XYZ menggunakan MySQL untuk

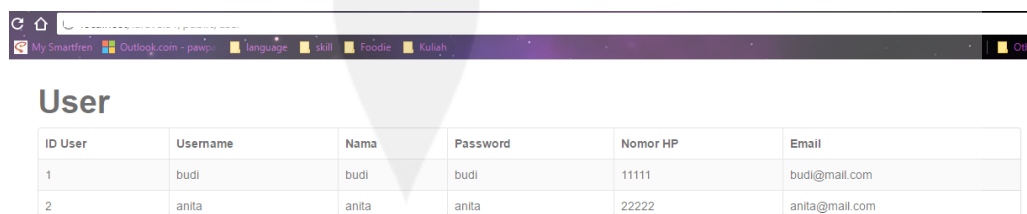
mengolah basis datanya. Lalu saya memodifikasi basis data seperti menambahkan kolom dan tabel baru.



**Gambar 2.1.2 Tampilan MySQL Workbench**

Gambar 2.1.2 merupakan tampilan dari MySQL Workbench yang digunakan di PT. XYZ.

Setelah Laravel 5.1 saya mulai membuat halaman untuk menampilkan data User dan Toko. *Framework* Laravel 5.1 merupakan hal baru bagi saya, karena saya belajar *Framework* Laravel 4 pada mata kuliah Pemrograman Web Lanjut, maka dari itu saya membuat tampilan dengan panduan dari Ko Nico dan Ci Yunita serta tutorial dari Google mengenai Laravel 5.1.



**Gambar 2.1.3 Tampilan Data User**



## Toko

ID Cabang	Kode Toko	Nama Toko	Alamat	Latitude	Longitude
1	F065	LEGOK	KP BABAKAN RT.03/01 BABAKAN LEGOK	-6.2985	106.5882222
1	F098	TEMBAGA	JL. TEMBAGA NO. 44 RT.07/09 BENCONGAN CURUG	-6.213972222	106.5966944
1	F117	WARINGIN KURUNG	JL.WARINGIN KURUNG No.44	-6.036333333	106.0845833

### Gambar 2.1.4 Tampilan Data Toko

Gambar 2.1.3 merupakan tampilan data pengguna yang ada. Tampilan ini dapat diakses di website, dan menggunakan *framework* Laravel 5.1. Data pengguna yang ditampilkan adalah ID pengguna, *username*, nama, *password*, nomor HP, dan *e-mail*. Gambar 2.1.4 merupakan tampilan data toko yang ada. Tampilan ini dapat diakses di *website*, dan menggunakan *framework* Laravel. Data toko yang ditampilkan adalah ID cabang, kode toko, nama toko, alamat, *latitude*, dan *longitude*.

### 6 Januari 2017, Head Office PT. XYZ

Hari ini saya melanjutkan pekerjaan saya dari hari sebelumnya, menambahkan tabel Cabang karena di setiap toko memiliki cabang tertentu, membuat halaman untuk menampilkan Kalender menggunakan *javascript* dari [fullcalendar.com](http://fullcalendar.com). Selain itu, saya membuat *model* dan *controller* dari setiap tabel yang ada di basis data, saya diajarkan untuk membuat *model* dan *controller* menggunakan *command line* agar pekerjaan semakin mudah dan efisien.

```
C:\WINDOWS\system32\cmd.exe
C:\xampp\htdocs\laravel51>php artisan make:controller KalenderController
Controller created successfully.
C:\xampp\htdocs\laravel51>php artisan make

[InvalidArgumentException]
Command "make" is not defined.
Did you mean one of these?
make:controller
make:middleware
make:migration
make:listener
make:provider
make:command
make:console
make:request
make:policy
make:seeder
make:event
make:model
make:test
make:job

C:\xampp\htdocs\laravel51>
```

**Gambar 2.1.5 Pembuatan Controller dan Model**

Gambar 2.1.5 merupakan tampilan cara pembuatan *controller* dan *model* menggunakan *command line*.

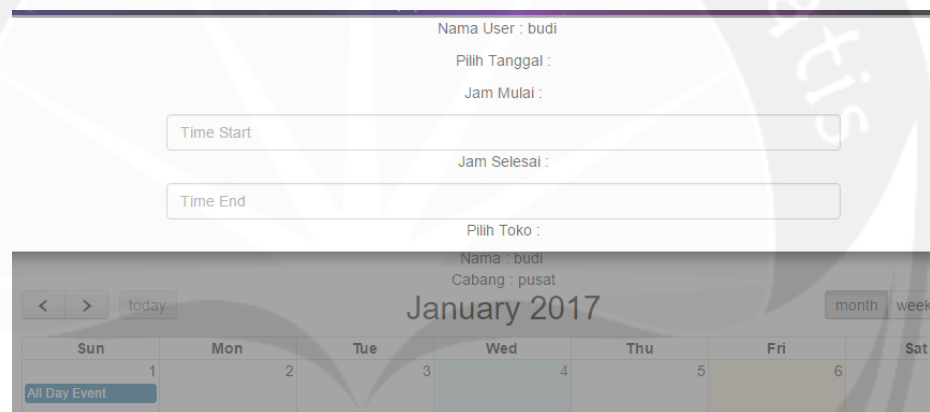
Saya membuat halaman untuk tampil kalender berdasarkan pengguna yang *login*, sehingga alamat yang diakses harus membawa ID pengguna yang sedang *login*. Di halaman kalender ini, akan ditampilkan jadwal-jadwal yang akan dilakukan pengguna tersebut, seperti kunjungan ke suatu Toko O. Halaman ini juga akan berfungsi untuk menambahkan jadwal karyawan tersebut, data yang dimasukkan untuk membuat suatu jadwal adalah tanggal, jam mulai, dan jam selesai sebagai batas keterlambatan karyawan, serta toko yang akan dijadwalkan untuk dikunjungi. Semua data dimasukkan menggunakan tampilan *modal* yang akan muncul jika pengguna mengklik tanggal yang diinginkan.

## Absensi

### Kalender User Cabang



**Gambar 2.1.6** Kalender Absensi



**Gambar 2.1.7** Tampilan Modal untuk Membuat Jadwal

Gambar 2.1.6 merupakan tampilan kalender yang akan menampilkan jadwal penggunaannya. Terdapat tampilan berdasarkan bulan dan hari di kalender tersebut. Gambar 2.1.7 merupakan tampilan *modal* yang muncul ketika akan membuat jadwal, dengan mengklik tanggal yang diinginkan.

### 9 Januari 2017, Head Office PT. XYZ

Hari ini saya mengikuti *Company Visit* ke *Head Office* PT. XYZ. Di *Head Office* ini saya dan teman-teman yang juga mengikuti kerja praktek di PT. XYZ dijelaskan tentang sejarah berdirinya perusahaan, dan anak perusahaan yang di bawah PT. XYZ. Jenis usaha yang

dimiliki PT. XYZ sangatlah beragam, hingga saat ini PT. XYZ memiliki 7 Sektor Bisnis dan akan terus berkembang.

Kami dijelaskan tentang jenis-jenis retail secara umum dan usaha retail yang dimiliki PT. XYZ seperti minimarket Toko X yang dimiliki oleh PT. XYZ itu sendiri maupun orang luar sebagai bisnis *franchise*, minimarket Toko O yang dapat dimiliki oleh koperasi maupun perorangan, serta Pusat Perkulakan yaitu Toko Y menjual barang secara grosir. Untuk bisnis *food and beverage* merupakan makanan olahan yang dijual di minimarket Toko X, dengan merek Y berupa suki, es krim, makanan, kopi. Sedangkan untuk *bakery*, PT. XYZ memiliki beberapa merek atau kerjasama yaitu Mister A, Mister B, Roti C, dan Roti D yang dijual di minimarket Toko X. Untuk restoran Jepang, PT. XYZ bekerja sama dengan restoran nomor 1 yang ada di Jepang.

Saat jam makan siang, kami semua diajak ke Mall of Indonesia, untuk makan siang. *Company visit* diakhiri dengan makan bersama di restoran milik PT. XYZ.



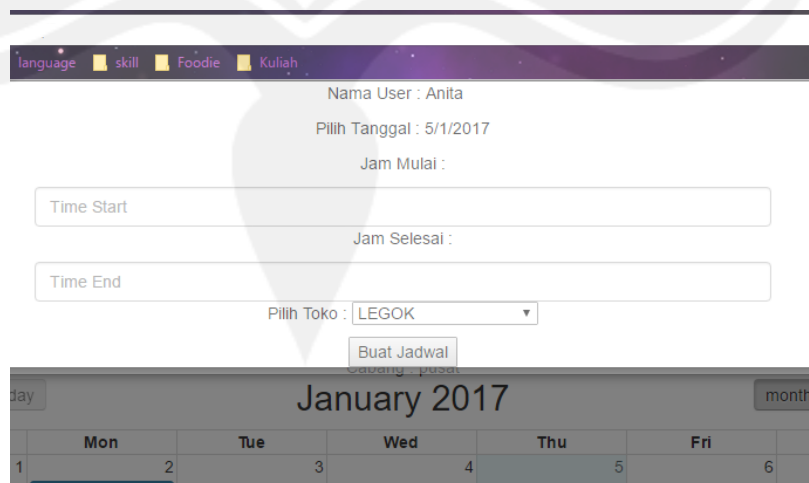
**Gambar 2.1.8 Foto Bersama di Salah Satu Restoran Milik PT. XYZ**

Gambar 2.1.8 foto bersama dengan teman-teman *internship*, dan HRD dari PT. XYZ. Dari ujung kiri Antony (SI UKDW), Sulis (TF UBAYA), Dwiki (TI UAJY), Ko Eka, Ci Astrid, Ci Sharron, Emma (TF Unika Soegijapranata), saya (TF UAJY), Michelle (TP Unika Soegijapranata), Widya (FE Univ. Surya), Christofer (SI UKDW).

### 10 Januari 2017, Head Office PT. XYZ

Pada hari ini, saya membuat data toko yang ada tampil di *modal* saat membuat jadwal menggunakan *drop down list*. Toko yang dipilih adalah toko yang merupakan cabang yang berada ditugaskan untuk karyawan tersebut. Pada bagian tanggal telah ditampilkan di *modal* sesuai tanggal yang dipilih, sehingga pengguna dapat melihat tanggal yang telah dipilih. Untuk pengisian jam, saya menggunakan *javascript timepicker*, sehingga pengguna tidak perlu menuliskan jam, cukup memilih dari desain yang berbentuk jam dan menit. Pada bagian akhir *modal* terdapat tombol untuk menyimpan jadwal yang sudah dimasukkan ke dalam basis data.

Untuk tampil jadwal di halaman kalender akan muncul sesuai tanggal yang ada disetiap jadwal. Pengguna dapat melihat jadwal yang ada dengan judul: nama toko dan kode toko.



language skill Foodie Kuliah

Nama User : Anita

Pilih Tanggal : 5/1/2017

Jam Mulai :

Time Start

Jam Selesai :

Time End

Pilih Toko : LEGOK

Buat Jadwal

January 2017

Mon	Tue	Wed	Thu	Fri	Sat
1	2	3	4	5	6

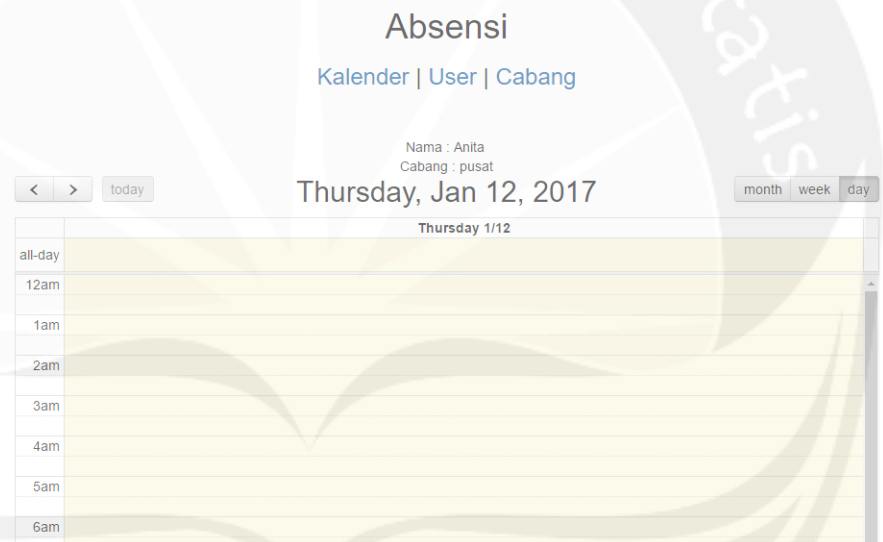
**Gambar 2.1.9 Modal untuk Membuat Jadwal**



Gambar 2.1.9 merupakan tampilan *modal* yang sudah dapat tampil seperti nama user, tanggal, jam mulai, jam selesai, toko yang dikunjungi.

### 11 Januari 2017, Head Office PT. XYZ

Hari ini saya merubah tampilan hari, pengguna juga dapat menambahkan jadwal dengan memblok jam yang dipilih sehingga *modal* dapat tampil. Fungsi *modal* yang ada sama seperti pada tampilan bulan, namun pada jam mulai dan jam selesai sudah terisi jam yang diblok sehingga pengguna cukup memilih tokonya saja.



### Gambar 2.1.10 Tampilan Kalender dengan View Hari

Gambar 2.1.10 merupakan tampilan hari pada kalender. Ketika pengguna memasukkan jadwal ditampilkan ini, pengguna cukup memilih tokonya saja. Pada gambar ini merupakan tampilan per hari pada kalender.

Saya juga membuat fungsi untuk merubah jadwal dengan cara memindahkan jadwal tersebut ke tanggal lain, sehingga pada basis data tanggal akan berubah sesuai tempat peletakan jadwal. Perubahan jadwal pun dapat dilakukan dengan *resize* jadwal, sehingga jadwal tidak hanya 1 hari, tetapi bisa lebih dari 1 hari.

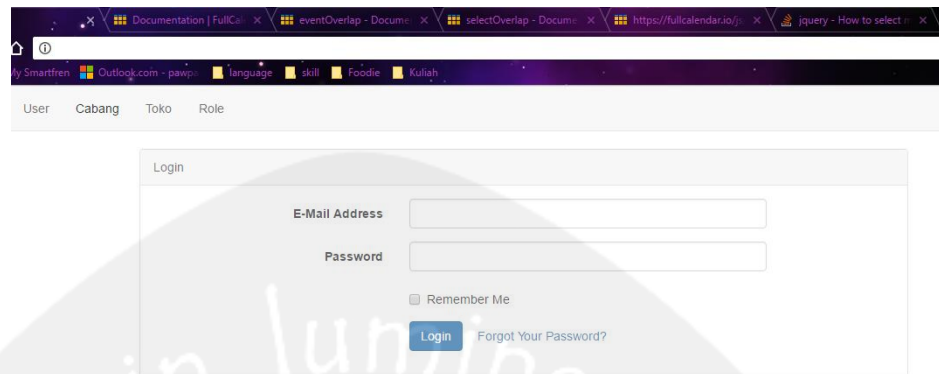
**12 Januari 2017, Head Office PT. XYZ**

Pada hari ini, saya menambahkan tabel baru yaitu tabel Role yang akan digunakan untuk *login*. Dengan menambahkan tabel Role, maka pada bagian *back-end* saya menambahkan model Role serta controllernya.

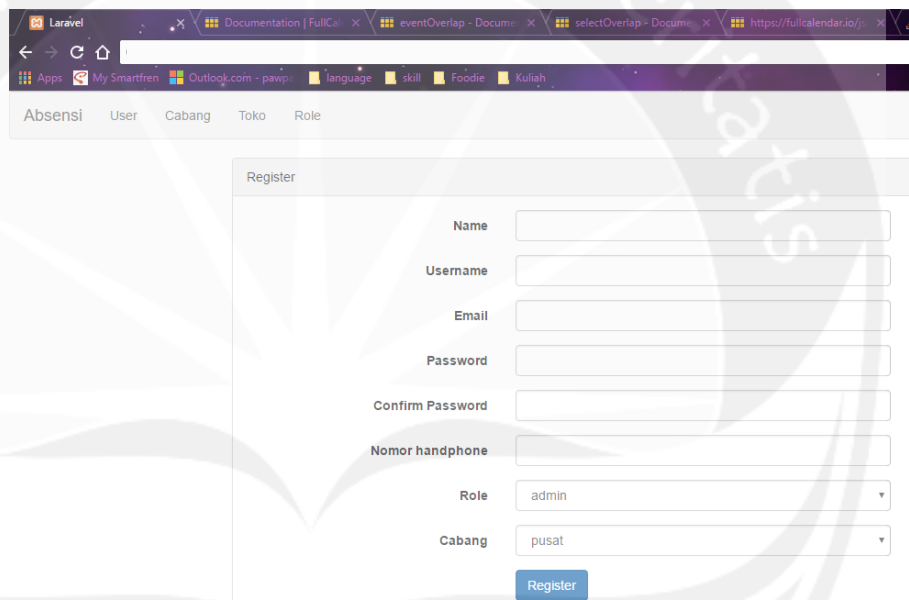
Setelah membuat *model* dan *controller* saya membuat fungsi dan tampilan tambah Role, Toko, Cabang Baru. Data yang dimasukkan untuk menambahkan Toko yaitu cabang, kode toko, nama toko, alamat toko, *latitude*, dan *longitude*. Untuk menambahkan Role cukup menambahkan nama role, karena ID akan *tergenerate* otomatis. Selain itu, saya juga menambahkan fungsi dan tampilan *login* dan *register*. Untuk *register*, pengguna harus memasukkan nama, *username*, *e-mail*, *password*, konfirmasi *password*, nomor HP, role yang dimiliki serta cabang. *Register* dan *Login* ini merupakan *scaffold* yang dapat diinstal pada Laravel 5.1, sehingga untuk tampilan *login* dan *register*, serta tampilan *login* sudah didapat dari *Scaffold Best Momo*. Untuk tampilan *register* saya merubah agar sesuai kebutuhan.

The screenshot shows a web browser window with three tabs: 'Laravel', 'Laravel', and 'php - Error when trying'. The address bar contains 'http://localhost:8080/'. Below the browser, the application interface is visible, featuring a 'Laravel Welcome' header. The main content area includes a dropdown menu for 'Cabang' with 'pusat' selected, and five input fields for 'Kode Toko', 'Nama Toko', 'Alamat Toko', 'Latitude', and 'Longitude'. A 'Tambah' button is located at the bottom of the form.

**Gambar 2.1.11 Tampilan Tambah Toko**



**Gambar 2.1.12 Tampilan Login Pengguna**



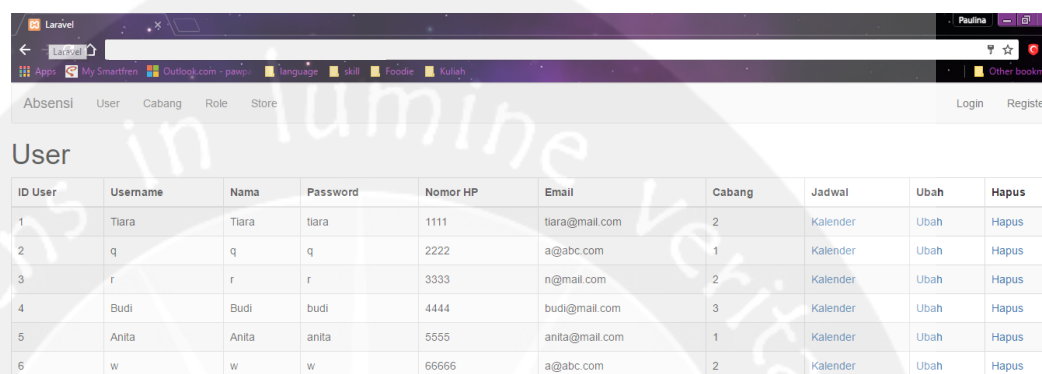
**Gambar 2.1.13 Tampilan Register Pengguna**

Gambar 2.1.11 merupakan tampilan untuk menambahkan toko baru ke dalam basis data. Gambar 2.1.12 merupakan tampilan Login Pengguna dari *Scaffold Best Momo*. Pengguna dapat *login* ke dalam sistem dengan memasukkan *e-mail* dan *password*. Gambar 2.1.13 merupakan tampilan *Register* Pengguna dari *Scaffold Best Momo*.

### **13 Januari 2017, Head Office PT. XYZ**

Hari ini saya membuat fungsi dan tampilan untuk ubah, dan hapus untuk Role, Toko, Cabang, dan User. Pengguna dapat merubah data suatu Role, Toko, Cabang dan User dengan mengklik tombol ubah

pada tabel tampil. Untuk hapus Role, Toko, Cabang dan User menggunakan *soft delete* sehingga data tidak hapus begitu saja di basis data, namun tidak akan muncul karena ada kolom *flag* untuk menandai jika data tersebut sudah dihapus.



ID User	Username	Nama	Password	Nomor HP	Email	Cabang	Jadwal	Ubah	Hapus
1	Tiara	Tiara	tiara	1111	tiara@mail.com	2	Kalender	Ubah	Hapus
2	q	q	q	2222	a@abc.com	1	Kalender	Ubah	Hapus
3	r	r	r	3333	n@mail.com	2	Kalender	Ubah	Hapus
4	Budi	Budi	budi	4444	budi@mail.com	3	Kalender	Ubah	Hapus
5	Anita	Anita	anita	5555	anita@mail.com	1	Kalender	Ubah	Hapus
6	w	w	w	66666	a@abc.com	2	Kalender	Ubah	Hapus

**Gambar 2.1.14 Tampilan Data Pengguna**

Gambar 2.1.14 merupakan tampilan untuk tampil user. Sudah terdapat pilihan untuk melihat kalender, ubah, dan hapus.

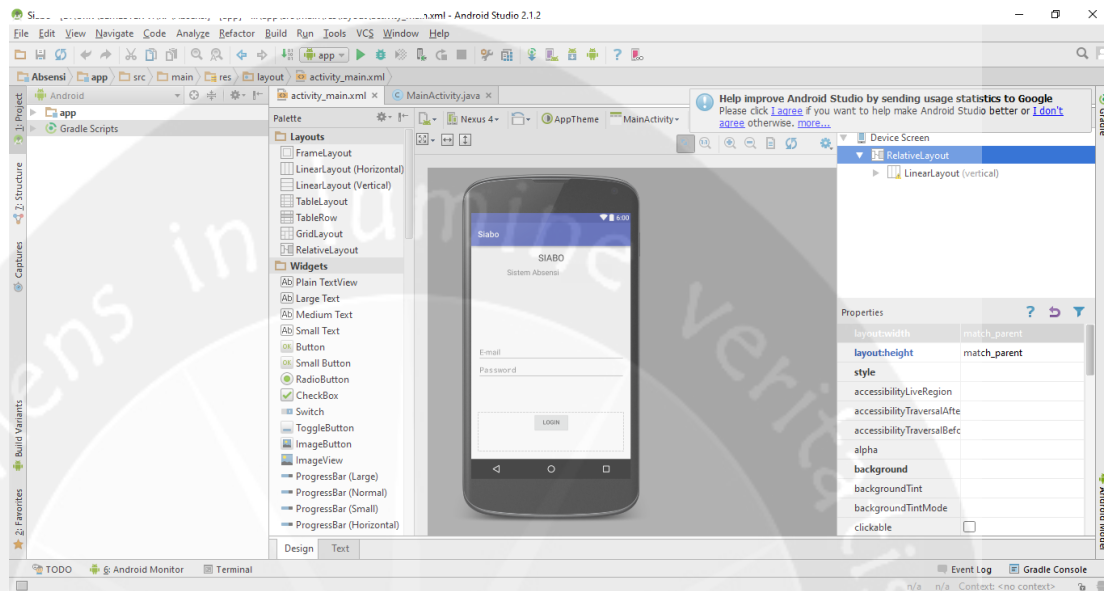
#### **16 Januari 2017, Head Office PT. XYZ**

Hari ini saya mencoba fungsi-fungsi yang sudah saya buat selama 2 minggu kemarin, dan memperbaiki *error* yang ada. Fungsi *drag*, *resize* kalender yang minggu sebelumnya tidak berfungsi, sehingga saya membetulkan fungsi tersebut sehingga dapat digunakan. Kemudian saya memperbaiki tampilan kalender, sehingga menjadi *user friendly* dan tidak membingungkan penggunanya.

#### **17 Januari 2017, Head Office PT. XYZ**

Hari ini saya memperbaharui SDK Android Studio karena SDK yang saya miliki tidak sama dengan SDK yang digunakan di perusahaan. Selain itu, saya mencoba mengupdate Android, namun gagal. Yang saya kerjakan hari ini adalah membuat rancangan yang ada pada aplikasi, dan saya membuat tampilan *login* menggunakan

Android Studio. Karena *login* berbasis *mobile*, maka saya membuat API untuk *login* yang menggunakan *token*.

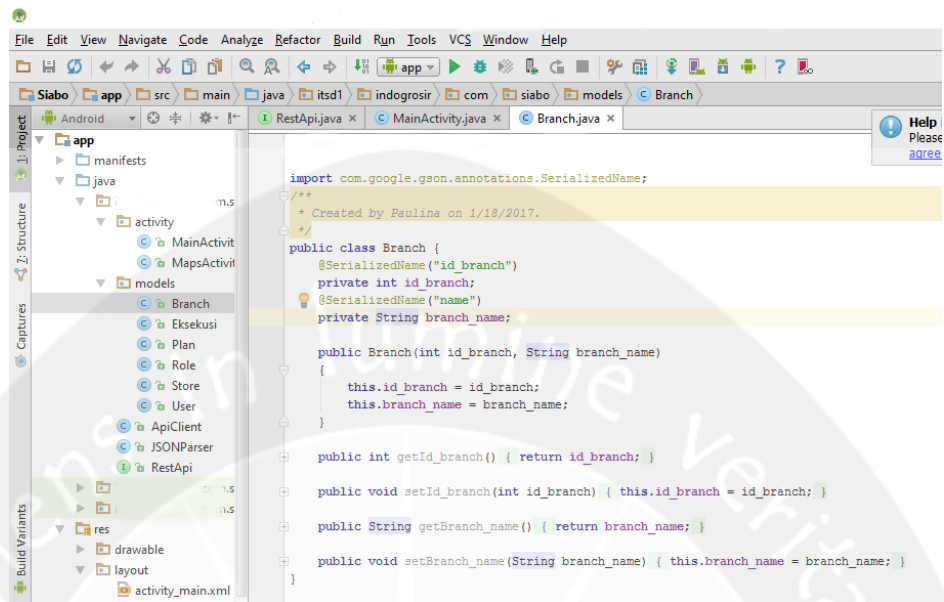


**Gambar 2.1.15 Tampilan Aplikasi pada Android**

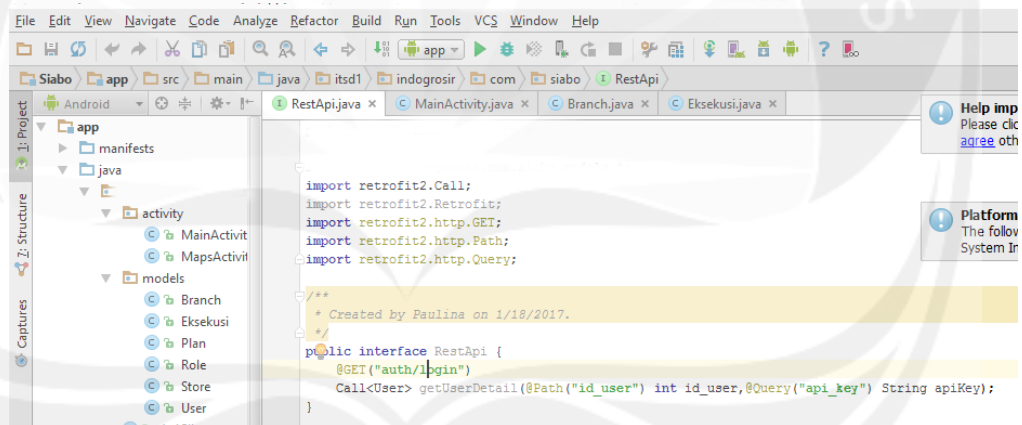
Gambar 2.1.15 merupakan tampilan aplikasi pada Android Studio.

**18 Januari 2017, Head Office PT. XYZ**

Hari ini saya membuat model di Android, model yang dibuat sama seperti pada *back-end*, sehingga saya membuat model untuk User, Role, Toko, Cabang, Planning, Eksekusi. Saya menggunakan Retrofit untuk pengaksesan API dari Android. Retrofit adalah *rest client* untuk Android dan Java, *library* ini menyediakan *framework* untuk *authentication* dan interaksi dengan API, dengan mengirimkan *request* menggunakan OkHttp. *Library* ini mengambil data dari API menjadi data berbentuk JSON atau data XML, ketika data dapat diambil maka akan dibawa ke bagian *response* untuk diproses lebih lanjut.



**Gambar 2.1.16 Tampilan Kelas Model Cabang di Android**



**Gambar 2.1.17 Tampilan Routing API**

Gambar 2.1.16 merupakan tampilan kelas Model Cabang yang digunakan pada Retrofit. Gambar 2.1.17 merupakan tampilan *routing* API. *Interface* ini berguna untuk menyambungkan URL dari aplikasi ke API.

**19 Januari 2017, Head Office PT. XYZ**

Hari ini saya belajar menggunakan Retrofit berdasarkan tutorial-tutorial dari Google. Saya mulai dari membuat fungsi untuk *login*

menggunakan Retrofit, data diambil menggunakan URL yang ada di API.

```
Call<Login> call = ra.getLogin(login);
call.enqueue(new Callback<Login>()
{
    @Override
    public void onResponse(Call<Login> call, Response<Login> response)
    {
        findViewById(R.id.loading_panel).setVisibility(View.GONE);
        token = response.body().getToken();
        Bundle b = new Bundle();
        b.putString("token", token);
        Intent i = new Intent(MainActivity.this, UserDetails.class);
        i.putExtras(b);
        startActivity(i);
    }
    @Override
    public void onFailure(Call<Login> call, Throwable t)
    {
        if(isOnline())
        {
            Log.d("Log", t.toString());
            findViewById(R.id.loading_panel).setVisibility(View.GONE);
            AlertDialog.Builder alertDialogBuilder = new AlertDialog.Builder(MainActivity.this);
            alertDialogBuilder.setMessage("Email atau Password Anda salah");
            alertDialogBuilder.setNegativeButton("OK", new DialogInterface.OnClickListener()
            {
                public void onClick(DialogInterface dialog, int which)
                {
                    .
                }
            });
        }
    }
});
```

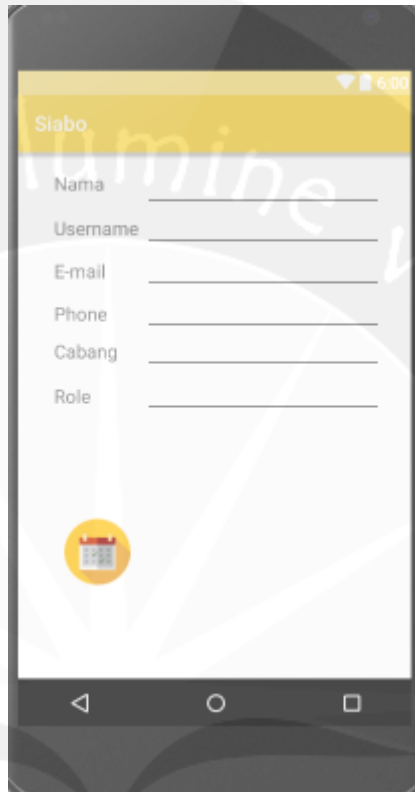
### Gambar 2.1.18 Tampilan Fungsi Login di Android

Gambar 2.1.18 merupakan tampilan penggunaan Retrofit yaitu memanggil fungsi `getLogin`, terdapat bagian `onResponse` dan `onFailure`. Jika data berhasil diambil maka akan masuk ke bagian `onResponse`, namun jika data gagal diambil maka akan masuk ke bagian `onFailure`. Contohnya pada penggunaan API untuk *login* pengguna, jika pengguna berhasil *login* maka akan masuk ke *activity* baru, jika pengguna tidak berhasil *login* maka akan muncul peringatan jika *e-mail* dan *passwordnya* salah. Selain itu, saya membuat tampilan-tampilan Android lainnya.

**20 Januari 2017, Head Office PT. XYZ**

Hari ini saya melanjutkan pekerjaan di hari sebelumnya, saya membuat tampilan untuk menampilkan data pengguna. Untuk menampilkan data pengguna, saya membuat fungsi API sesuai dengan pengguna yang sedang *login* pada saat itu. API tersebut diakses

menggunakan Retrofit, data yang sudah didapat dari API tersebut akan dimasukkan ke *textView* pada layout, sehingga data pengguna dapat ditampilkan di Android.



**Gambar 2.1.19 Tampilan Detil Pengguna**

Gambar 2.1.19 merupakan tampilan detil pengguna yang sedang *login* pada Android. Untuk keterangan nama, *username*, *e-mail*, sampai *role* dan inputannya saya menggunakan *TextView*, sedangkan untuk garis yang menunjukan ke pengguna jika bagian itu merupakan inputan adalah *CustomView* yang diberi border.

**23 Januari 2017, Head Office PT. XYZ**

Pada hari ini, saya membuat tampilan untuk kalender dalam aplikasi Android. Saya menggunakan *CalendarView* dapat menampilkan jadwal per penggunaanya. Fungsi untuk menampilkan jadwal per penggunaanya saya masih mempelajarinya dari Google karena harus mengambil data dari API yang sudah saya buat. Untuk

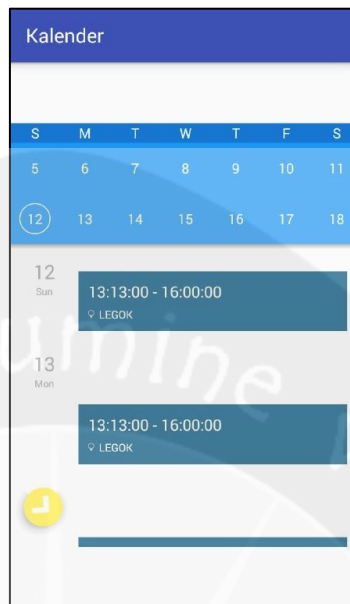


tampilan detil user pun saya perbaiki sehingga mudah dilihat oleh pengguna.

**24 Januari 2017, Head Office PT. XYZ**

Hari ini, saya mengganti *widget* *CalendarView* pada android dengan tampilan kalender dari *library*. *Widget* *CalendarView* sulit untuk dimodifikasi jika memasukkan *event* dari API yang akan dibuat. *Library* kalender yang saya gunakan dari github, merupakan *widget* yang *open source* dari <https://github.com/Tibolte/AgendaCalendarView>. Untuk mengetahui penggunaan dari *custom widget* ini, saya *mendownload* tutorial dari github dan dokumentasi yang dibuat oleh *developer* tersebut.

Saya memilih *widget* ini karena tampilan ke pengguna lebih nyaman, dan berwarna, selain itu untuk memasukkan, menampilkan data pada *AgendaCalendarView* lebih mudah karena ada dokumentasi yang sudah disediakan. Kemudian saya membuat API untuk mengambil data jadwal karyawan dari basis data. Data yang diambil menggunakan API, akan ditampilkan pada *widget* *AgendaCalendarView*, sesuai dengan tanggal yang ada di jadwal. Dalam *AgendaCalendarView*, terdapat tampilan per bulan maupun per hari, pengguna tidak perlu mencari jadwal pada hari ini karena tampilan *default* sesuai dengan hari ini.



**Gambar 2.1.20 Tampilan Kalender**

Gambar 2.1.20 merupakan tampilan kalender, dimana pengguna dapat melihat jadwal yang dimilikinya.

#### **25 Januari 2017, Head Office PT. XYZ**

Pada hari ini saya memperbaiki fungsi untuk tampil data jadwal di kalender aplikasi. Di hari sebelumnya, data jadwal yang tampil adalah semua jadwal dari berbagai pengguna pada satu pengguna yang *login*. Sehingga saya memperbaiki fungsi dari API, agar data jadwal yang ditampilkan sesuai dengan jadwal pengguna yang sedang *login*. Tampilan jadwal sudah dapat memunculkan untuk jadwal pengguna pada hari tersebut.

#### **26 Januari 2017, Head Office PT. XYZ**

Hari ini, saya masih memperbaiki fungsi untuk tampil jadwal di kalender aplikasi. Setelah mencoba memasukkan beberapa data untuk hari yang sama, ditemukan *bug* yaitu jadwal yang lebih dari satu yang tampil di hari yang sama tidak muncul di kalender. Maka dari itu, saya harus memperbaiki fungsi API kembali. Setelah diperbaiki, tampilan

jadwal pada kalender sudah dapat ditampilkan, meskipun dalam 1 hari terdapat banyak jadwal.

Selain itu, saya menambahkan alarm untuk memberitahukan jadwal yang akan datang kepada penggunanya. Alarm yang muncul kepada pengguna hanya berupa *toast* tanpa suara/musik. Saya belum menambahkan suara dan *push notification* pada alarm, karena belum diketahui apakah akan digunakan untuk memberitahukan penggunanya tentang jadwal yang dimiliki.

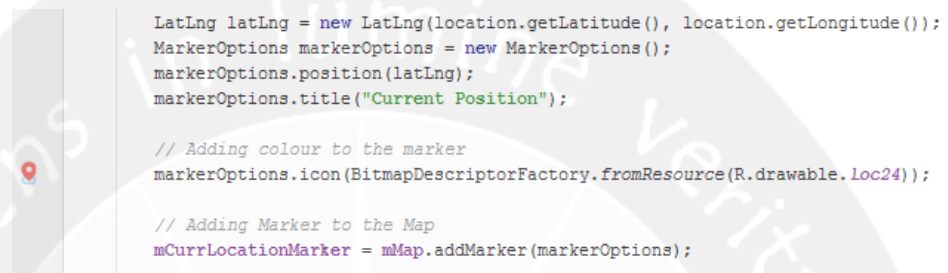
### **27 Januari 2017, Head Office PT. XYZ**

Pada hari ini, saya membuat tampilan peta yang akan digunakan untuk menampilkan lokasi terakhir pengguna dan lokasi toko yang dijadwalkan. Peta yang saya gunakan adalah peta yang disediakan dari Google menggunakan *free plan*, sehingga terdapat batasan untuk penggunaan API ini. Setelah aplikasi ini diluncurkan/*di live*, pilihan pada API tersebut akan diganti dengan pilihan yang dimiliki PT. XYZ agar batasan penggunaan API lebih besar.

Pertama, untuk dapat menampilkan peta di Android menggunakan API dari Google Maps, saya harus meminta API *key* dari *sand box* yang disediakan oleh Google. API *key* Google Maps dapat diakses melalui [console.developers.google.com](https://console.developers.google.com). Fitur untuk mendapatkan lokasi terakhir pengguna sudah saya pelajari dari mata kuliah Pemrograman Aplikasi Mobile, pada modul 8 yaitu modul *Geolocation*. Namun yang saya pelajari sedikit berbeda untuk penggunaan aplikasi Siabo, sehingga saya membutuhkan referensi lain dari github. Karena terlalu lama mempelajari referensi yang saya temukan, maka untuk fitur mendapatkan lokasi terakhir pengguna dan lokasi toko belum dapat dikerjakan.

### 30 Januari 2017, Head Office PT. XYZ

Hari ini saya melanjutkan pekerjaan kemarin, yaitu mendapatkan lokasi pengguna saat itu dengan memanfaatkan API dari Google Maps. Untuk menunjukkan lokasi pengguna, digunakan *marker*/penanda yang menandai lokasi pengguna pada peta.



#### Gambar 2.1.21 Fungsi Pembuat *Marker*

Gambar 2.1.21 merupakan fungsi untuk membuat *marker*/penanda lokasi pengguna pada peta. Untuk membuat penanda di peta, dibutuhkan *longitude* dan *latitude* dari pengguna, yang didapat dari API Google Maps. Jika pengguna bergeser ke tempat lain maka penanda pada peta akan berubah mengikuti lokasi pengguna.

### 31 Januari 2017, Head Office PT. XYZ

Hari ini saya membuat fungsi untuk menampilkan detail jadwal tertentu dari seorang pengguna. Jika pengguna mengklik suatu jadwal yang ada di *activity* kalender, maka akan muncul laman baru yaitu laman detail jadwal.



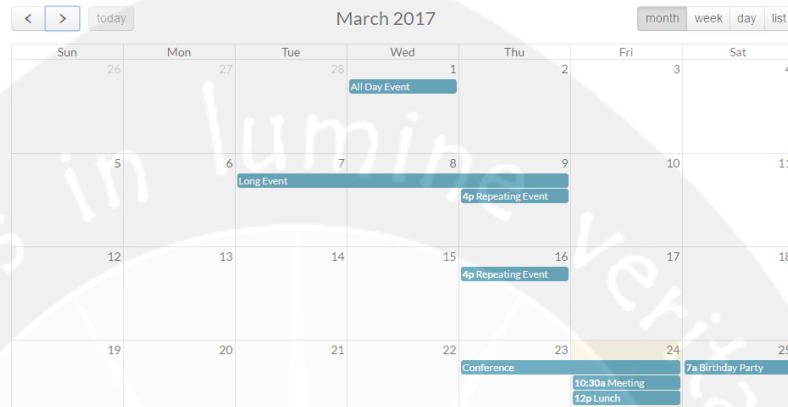
**Gambar 2.1.22 Tampilan Detil Jadwal**

Gambar 2.1.22 merupakan tampilan detil suatu jadwal yang dimiliki pengguna. Di dalam laman detil jadwal, terdapat data jadwal seperti toko yang dijadwalkan beserta alamatnya, jam berapa harus datang dan batas keterlambatannya. Untuk mendapatkan detil jadwal, aplikasi ini memerlukan akses ke basis data melalui API, sehingga saya membuat fungsi pada API.

#### **1 Februari 2017, Head Office PT. XYZ**

Pada hari ini, saya merubah tampilan dan fungsi kalender di *website*, karena permintaan Pak Andy yang mempertimbangkan kemudahan pengguna untuk menggunakan sistem ini. Sebelumnya, jadwal yang memiliki waktu lebih dari 1 hari (contoh: dari tanggal 2 Januari sampai 5 Januari) disimpan pada 1 data saja, akan dipisah menjadi data harian. Jika pengguna membuat jadwal baru yang lebih dari 1 hari (dari tanggal 2 Januari sampai 5 Januari), maka pada basis data akan menyimpan beberapa data sesuai tanggal yang dipilih (4 hari

terpilih, maka akan ada 4 data yang tersimpan). Tampilan juga akan berubah menjadi 1 garis 1 hari, tidak memanjang sebanyak hari yang dipilih.



**Gambar 2.1.23 Tampilan Kalender Sebelum Dirubah**

Gambar 2.1.23 merupakan tampilan kalender pada *website* sebelum fungsi *long event* dihilangkan. Saya ambil dari *fullcalendar.io*, karena saya tidak mendokumentasikannya.

Kemudian saya merubah tampilan detil jadwal di *website*, karena tidak ada lagi tanggal selesai pada detil jadwal. Sehingga detil jadwal yang ditampilkan adalah nama pengguna, toko yang dijadwalkan, tanggal penjadwalan, jam mulai dan jam selesai untuk rentang waktu yang diberikan pengguna untuk melakukan tugasnya.



**Gambar 2.1.24 Tampilan Kalender Setelah Dirubah**

Gambar 2.1.24 merupakan tampilan kalender pada *website* sesudah fungsi *long event* dihilangkan. Pada tiap jadwal juga sudah dirubah judulnya.

## 2 Februari 2017, Head Office PT. XYZ

Hari ini saya diberi tugas tambahan yaitu, setelah pengguna melakukan absensi pada aplikasi pengguna diharuskan melakukan sesuatu agar dapat diakui keasliannya. Tugas ini ditambahkan di fitur aplikasi ini, karena permintaan dari *management*. Saya membutuhkan tabel baru untuk menyimpan *list* tugas yang akan dikerjakan pengguna ke dalam basis data. Data yang disimpan di tabel adalah *id\_user* untuk ID pengguna yang memiliki jadwal tersebut, *id\_plan* untuk ID jadwal yang harus dikerjakan, *judul\_tugas*, *deskripsi\_tugas*, keterangan untuk memberi keterangan apakah sudah dikerjakan atau belum, dan *id\_bukti* untuk menyimpan ID bukti yang dikerjakan pengguna setelah *check in*. Dengan bertambahnya data yang disimpan saya pun juga harus merubah fungsi untuk menambahkan jadwal baru.

Ketika pengguna akan menambahkan jadwal baru di *website*, pengguna juga dapat memasukkan tugas apa yang harus dikerjakan anak buahnya. Pengguna cukup mengisikan judul tugas, dan deskripsi tugas yang akan dikerjakan. Dengan demikian, ID dari tabel *plan* menjadi *reference key* untuk tabel *to\_do* (tabel tugas) yang saya buat hari ini.



Table Name:

Collation:

Comments:

Column Name	Datatype	PK	NN	UQ	BIN	UN	ZF	AI	Default
id	INT(10)	<input checked="" type="checkbox"/>	<input checked="" type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input checked="" type="checkbox"/>	
judul_tugas	VARCHAR(50)	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	NULL
deskripsi_tugas	VARCHAR(200)	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	NULL
keterangan	VARCHAR(50)	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	NULL
id_bukti	INT(10)	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	NULL
id_user	INT(10)	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input checked="" type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	NULL
id_plan	INT(10)	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input checked="" type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	NULL
created_at	TIMESTAMP	<input type="checkbox"/>	<input checked="" type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	'0000-00-00 00:00:00'
updated_at	TIMESTAMP	<input type="checkbox"/>	<input checked="" type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	'0000-00-00 00:00:00'
deleted_at	TIMESTAMP	<input type="checkbox"/>	<input checked="" type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	'0000-00-00 00:00:00'
flag	INT(5)	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	NULL

**Gambar 2.1.25 Tampilan Basis Data untuk Tabel Tugas**

Gambar 2.1.25 merupakan tampilan basis data untuk tabel tugas, dan data yang akan disimpan beserta tipe datanya.

### **3 Februari 2017, Head Office PT. XYZ**

Hari ini saya masih melanjutkan perubahan untuk fungsi tambah tugas. Karena pada satu jadwal bisa terdapat banyak tugas, maka saya membuat fitur agar pengguna dapat menghapus *field* tugas di *modal* pembuatan jadwal. Dengan adanya fitur *field* tugas di *modal*, dapat membantu penggunaannya jika terjadi kesalahan sebelum menambahkan jadwal. Pada kalender saya menambahkan fungsi *on click*, sehingga pengguna dapat melihat detail jadwal di *website* ketika pengguna mengklik suatu jadwal di kalender.



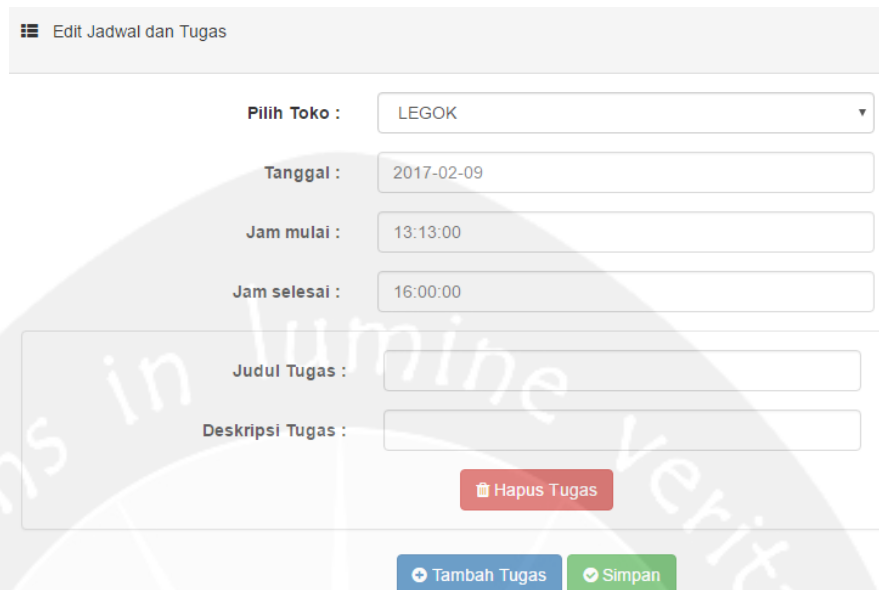
**Gambar 2.1.26 Tampilan Tambah Jadwal**

Gambar 2.1.26 merupakan tampilan modal untuk menambah jadwal, terdapat fitur tambah dan hapus tugas.

**6 Februari 2017, Head Office PT. XYZ**

Hari ini saya membuat fungsi ubah jadwal pada *website*, ketika pengguna ingin merubah suatu jadwal pengguna cukup mengklik tombol ubah pada modal detail jadwal.

**Gambar 2.1.27 Tampilan Tampil Jadwal**



☰ Edit Jadwal dan Tugas

Pilih Toko : LEGOK ▼

Tanggal : 2017-02-09

Jam mulai : 13:13:00

Jam selesai : 16:00:00

Judul Tugas :

Deskripsi Tugas :

Hapus Tugas

Tambah Tugas Simpan

















**Gambar 2.1.28 Tampilan Edit Jadwal**

Gambar 2.1.27 merupakan tampilan modal yang menampilkan detail suatu jadwal. Sedangkan gambar 2.1.28 merupakan tampilan edit jadwal, data sebelumnya akan ditampilkan pada tiap bagian tersebut. Data yang dapat dirubah adalah tanggal, jam mulai, dan jam selesai, serta tugas. Saya membuat halaman rubah jadwal, dengan menampilkan data-data sebelumnya, sehingga pengguna dapat tahu apa saja yang harus diubah. Di dalam edit jadwal, pengguna dapat menghapus maupun menambah tugas kepada bawahannya.

#### **7 Februari 2017, Head Office PT. XYZ**

Hari ini saya memperbaiki tampilan-tampilan yang ada di *website*. Saya memberi *icon* dan warna untuk setiap tombol, sehingga tampilannya tidak hanya berupa *link* saja. Saya juga mencari *bug* dan memperbaiki fungsi tambah tugas, ubah tugas, ubah jadwal, dan hapus jadwal.

## Data User

ID User	Username	Nama	Nomor HP	Email	Cabang	Jadwal	Tugas	Ubah	Hapus
7	Anita	Anita	33333	anita@mail.com	1				
8	Budi	Budi	2222	budi@mail.com	1				
9	ali	Ali	33333	ali@mail.com	4				
10	Rena	rena	3128923472	rena@mail.com	2				

### Gambar 2.1.29 Tampilan Tampil Data User

Gambar 2.1.29 merupakan tampilan *data user*, dengan *icon* yang sudah diperbaiki, yang sebelumnya berupa *link*.

#### 8 Februari 2017, Head Office PT. XYZ

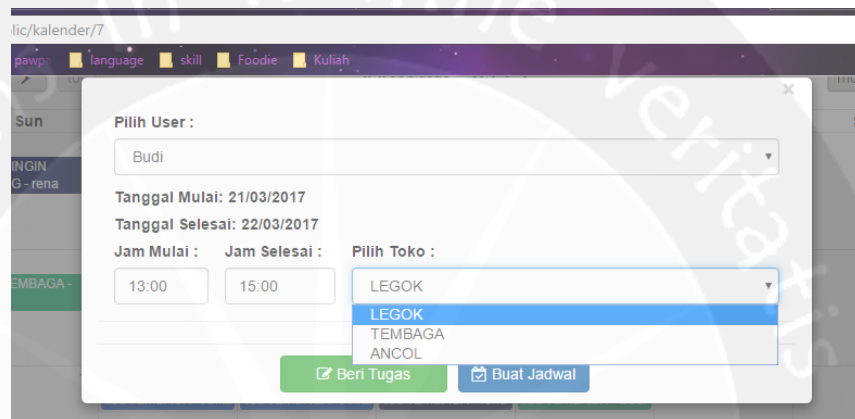
Hari ini saya mencoba semua fungsi yang ada di *website* dan menemukan beberapa *bug*. Fungsi-fungsi yang ada di *website* antara lain: tambah, ubah, hapus pengguna, toko, cabang, role. Setelah itu saya membuat API untuk menampilkan tugas-tugas pengguna di aplikasi, serta membuat tampilannya bersama dengan detail jadwal pengguna.

#### 9 Februari 2017, Head Office PT. XYZ

Hari ini saya merubah tampilan jadwal pada *website*. Pengguna yang sedang *login* merupakan atasan yang dapat melihat semua jadwal bawahannya, yang sebelumnya saya tampilkan adalah jadwal pengguna yang sedang *login*. Kemudian untuk menambahkan jadwal, juga saya ubah. Ketika pengguna ingin menambahkan jadwal, pengguna harus memilih bawahannya yang akan dijadwal dan diberikan tugas, nama bawahannya akan muncul menggunakan *drop down list*. Selain itu saya membuat tampilan pada aplikasi untuk menampilkan detail tugas pengguna, ketika pengguna mengklik tugas yang diinginkan.

## 10 Februari 2017, Head Office PT. XYZ

Hari ini saya masih memperbaiki fungsi tambah jadwal pada *website*. Saya membuat fungsi untuk menampilkan nama toko sesuai bagian karyawan yang dipilih oleh pengguna dengan *drop down list*. Toko yang ditampilkan merupakan toko yang terdaftar sesuai cabang yang menjadi bagian bawahannya.

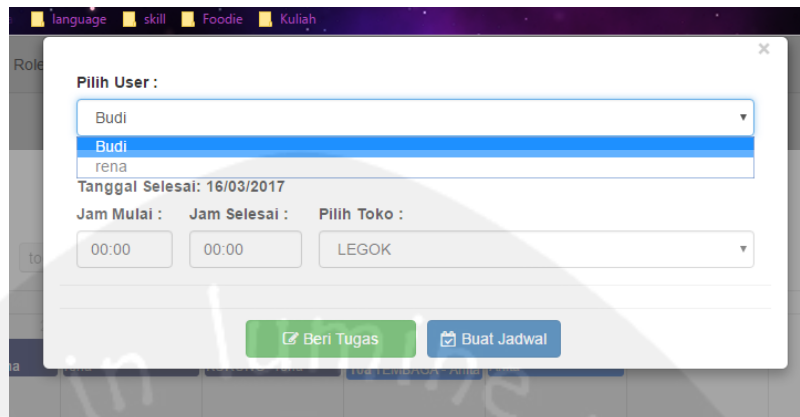


**Gambar 2.1.30 Drop Down List Pilih Toko**

Gambar 2.1.30 merupakan tampilan tambah jadwal dengan *drop down list* pada fungsi pilih toko. *Drop down list* dipilih karena memudahkan pengguna untuk memilih bawahan, dan toko, serta meminimalisir kesalahan yang akan dilakukan pengguna.

## 13 Februari 2017, Head Office PT. XYZ

Hari ini saya melanjutkan pekerjaan pada minggu sebelumnya. Tampilan bawahan yang akan dipilih pengguna yang sedang login, masih menampilkan nama pengguna itu sendiri di *drop down list*, sehingga saya memperbaiki fungsi tersebut. Pada tampilan kalender, judul jadwal yang ada diubah sesuai dengan nama karyawan yang dijadwalkan. Selain pada tampilan kalender, detil jadwal juga diubah dengan menampilkan nama karyawan yang memiliki jadwal tersebut.



**Gambar 2.1.31 Drop Down List Pilih User**

Gambar 2.1.31 merupakan tampilan tambah jadwal dengan *drop down list* pada fungsi pilih pengguna.

#### **14 Februari 2017, Head Office PT. XYZ**

Hari ini merupakan hari terakhir saya di *Head Office* PT. XYZ. Hari ini saya memperbaiki tampilan kalender di *website* dari hari sebelumnya. Di hari sebelumnya jadwal di kalender yang muncul hanya pada 1 bawahan saja, kemudian saya perbaiki lagi fungsinya agar dapat menampilkan nama bawahan yang lain, maupun bawahan yang masih memiliki bawahan lagi (Gambar 2.31). Selain itu, jadwal untuk tiap bawahan saya beri warna yang berbeda, sehingga dapat memudahkan pengguna untuk melihat jadwal yang ada.

## 2.2 Hasil Pekerjaan Secara Umum

Selama kerja praktek di PT. XYZ ada beberapa kegiatan yang saya lakukan di antaranya adalah:

1. Membuat Basis Data di *server*

Membuat basis data baru di *server* PT. XYZ menggunakan MySQL Workbench. Selain itu saya juga berkesempatan untuk mempelajari cara mengekspor maupun mengimpor basis data yang saya gunakan ke *server*.

2. Membuat Back End Project dengan Framework Laravel

Saya diberi tugas untuk membuat *website* yang digunakan sebagai *back end*, mulai dari menginstall Laravel 5.1 menggunakan *composer*, membuat fungsi tambah, ubah, tampil, dan hapus data. Selain itu saya juga belajar memperbaiki tampilan *website*, menggunakan HTML dan CSS. Saya juga belajar cara menggunakan *ajax*, untuk menampilkan data pada *modal*.

3. Membuat API Private

Untuk dapat mengakses data menggunakan Android, dibutuhkan API. API yang saya buat bersifat *private*, karena hanya digunakan untuk keperluan internal.

4. Membuat Aplikasi berbasis Android

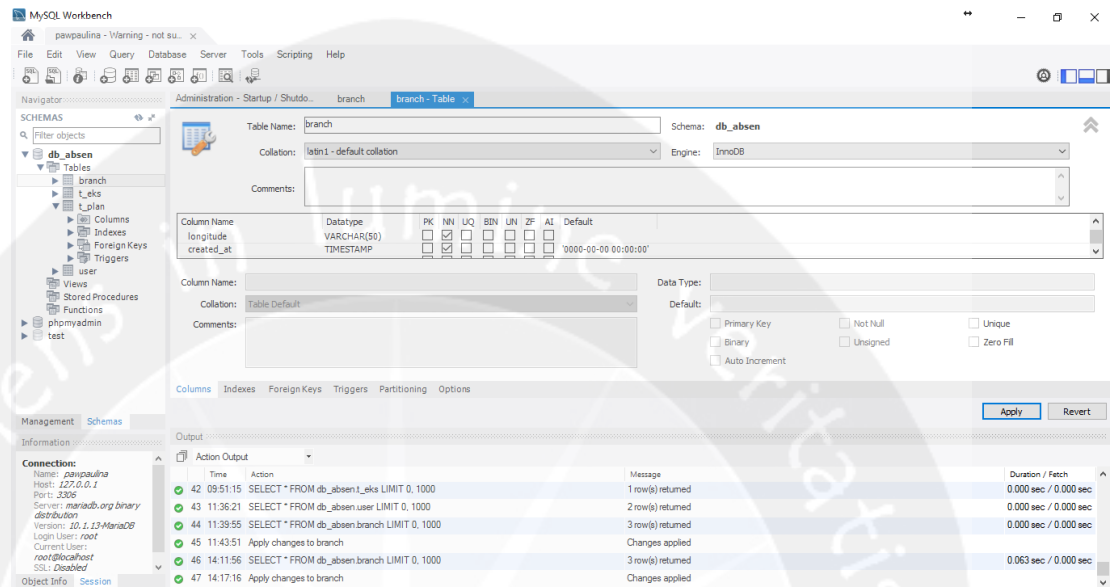
Perangkat seluler yang banyak digunakan di PT. XYZ, adalah Android. Selain banyak penggunaanya, pemasangan aplikasi di perangkatnya pun mudah dan murah.

5. Menggunakan API Public

Ketika pengguna mengakses peta pada aplikasi, saya menggunakan Google Maps API. Untuk menggunakan Google Maps API, dibutuhkan *key* yang diakses menggunakan Retrofit.

## 2.3 Bukti Hasil Pekerjaan

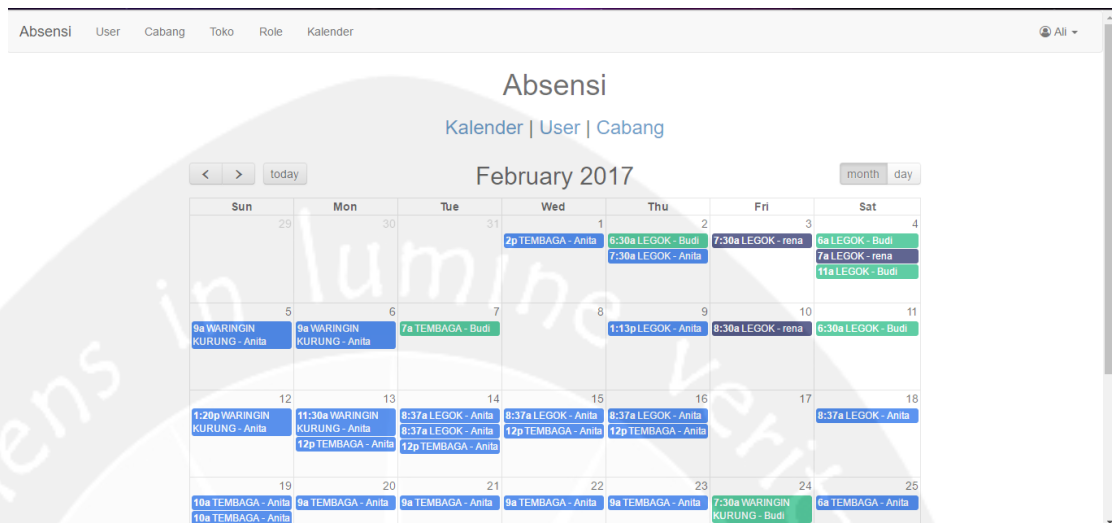
### 2.3.1 Membuat Basis Data



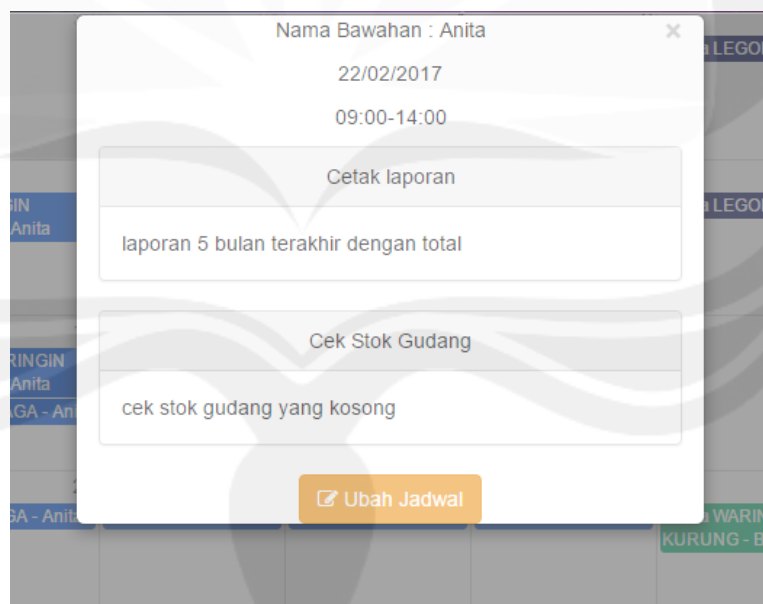
**Gambar 2.3.1 MySQL Workbench di PT. XYZ**

Pada gambar 2.3.1 merupakan tampilan MySQL Workbench, *software* pengelola basis data di PT. XYZ. Bagian kiri merupakan tampilan basis data yang ada di alamat tersebut. Untuk bagian kiri bawah terdapat keterangan mengenai alamat *server*, dan nama *server*. Pada gambar ini, *server* yang digunakan masih pada *server* lokal, belum diekspor ke *server* perusahaan.

### 2.3.2 Menggunakan javascript untuk menampilkan kalender

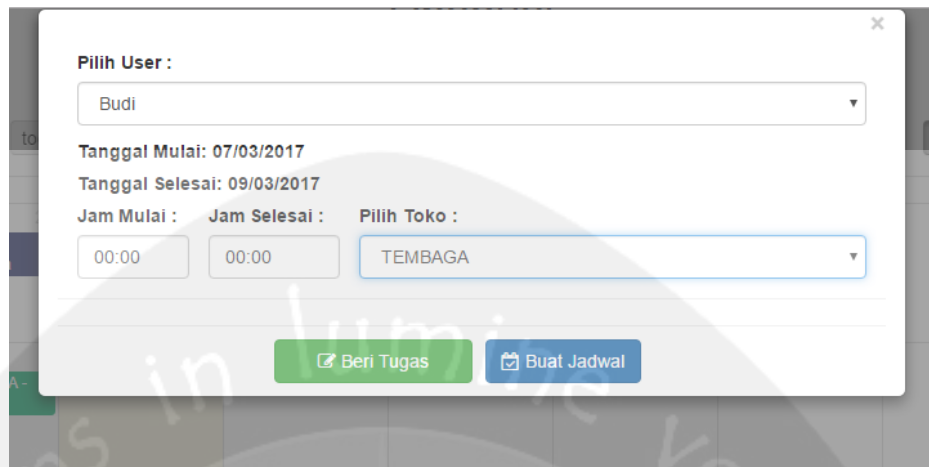


Gambar 2.3.2 Tampilan Kalender pada Website



Gambar 2.3.3 Tampilan Jadwal pada Website





Pilih User :

Budi

Tanggal Mulai: 07/03/2017

Tanggal Selesai: 09/03/2017

Jam Mulai : 00:00    Jam Selesai : 00:00    Pilih Toko : TEMBAGA

Beritugas    Buat Jadwal

**Gambar 2.3.4 Tampilan Penambahan Jadwal dan Tugas**

Pada Gambar 2.3.2 merupakan tampilan kalender yang akan digunakan karyawan yang memiliki anak buah. Anak buah yang memiliki jadwal akan tampil di kalender, dan memiliki warna yang berbeda. Selain jadwal, pengguna dapat pula melihat tugas anak buahnya seperti pada gambar 2.3.3. Tombol ubah jadwal pada gambar tersebut berguna untuk melakukan perubahan jadwal dan tugas, dan penghapusan jadwal maupun tugas. Pada gambar 2.3.4, pengguna dapat melakukan penambahan jadwal dan tugas yang akan diberikan kepada anak buahnya dengan memasukkan data nama karyawan, tanggal mulai dan tanggal selesai, jam mulai dan jam berakhirnya jadwal dan toko yang akan dijadwalkan.

### 2.3.3 Membuat API Private



```
Res.php x APIController.php x
\App\Http\Controllers\APIController cekEkeekusi

/**Untuk Login**/
public function authenticate(Request $request) {
    $credentials = $request->only(['email', 'password']);
    try {
        if (!$token = JWTAuth::attempt($credentials)) {
            return response()->json(['error' => 'invalid_credentials'], 403);
        }
    } catch (TokenExpiredException $e) {
        return response()->json(['token_expired'], $e->getStatusCode());
    } catch (TokenInvalidException $e) {
        return response()->json(['token_invalid'], $e->getStatusCode());
    } catch (JWTException $e) {
        return response()->json(['error' => 'could_not_create_token'], 500);
    } catch (\Exception $e) {
        return $e;
        return response()->json(['error' => 'could_not_create_token'], 500);
    }

    return response()->json(compact('token'));
}

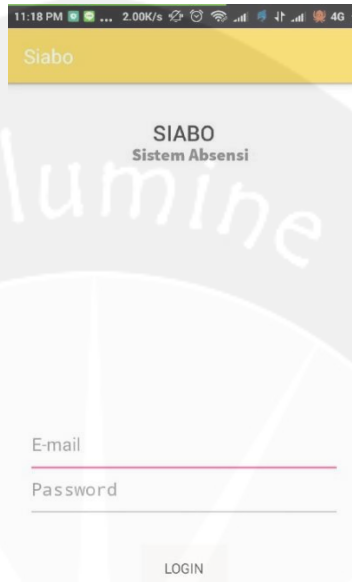
public function getAuthData($json)
{
    //Linux Server
    $response = explode("\n\n", $json, 2);
    if(count($response)<2)//Windows server
    {
        $response = explode("\r\n\r\n", $json, 2);
    }
    return json_decode($response[1]);
}

//return mandakaban jika gagal
```

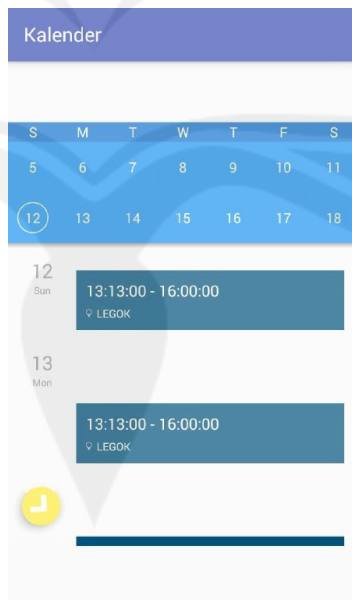
**Gambar 2.3.5 Code API Untuk Fungsi Login pada Android**

Pada Gambar 2.3.5, merupakan fungsi *login* yang ada Controller API yang digunakan pada aplikasi. API berfungsi untuk mengirimkan data-data yang akan diakses oleh pengguna aplikasi, dengan memanfaatkan koneksi yang ada pada jaringan yang sama. API bersifat *private* karena digunakan untuk keperluan internal pada PT. XYZ.

### 2.3.4 Membuat Aplikasi Berbasis Android



**Gambar 2.3.6 Tampilan Login pada Sistem**



**Gambar 2.3.7 Tampilan Kalender dan Jadwal pada Sistem**

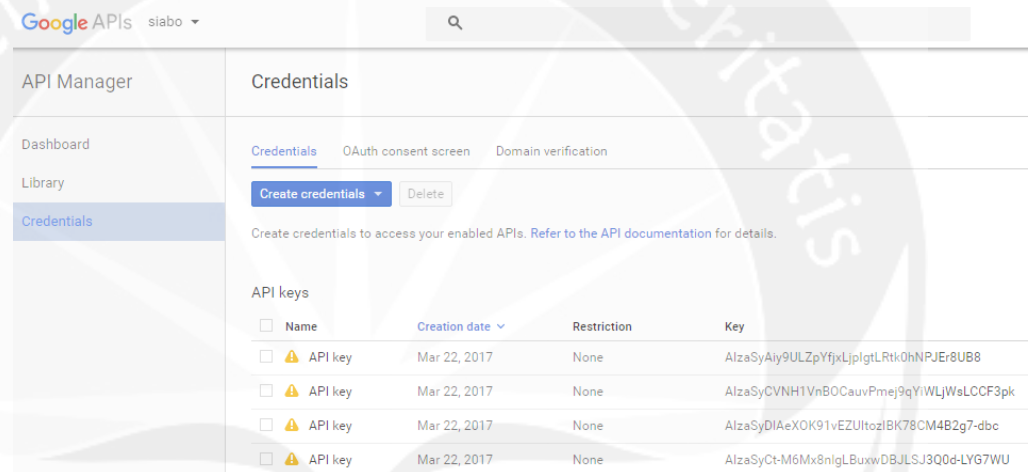


**Gambar 2.3.8 Tampilan Detil Jadwal pada Sistem**

Nama aplikasi berbasis Android yang saya buat adalah SIABO (Sistem Absensi untuk Toko O). Pengguna yang sudah didaftarkan melalui *website* dapat *login* ke aplikasi ini dengan memasukkan *e-mail* dan *password* yang dimiliki, seperti pada Gambar 2.3.6. Setelah *login* berhasil, pengguna akan dapat melihat jadwal yang diberikan atasan kepadanya pada Gambar 2.3.7. Pengguna dapat melihat detail jadwal dan detail tugas yang diberikan dengan mengklik pada jadwal yang diinginkan. Ketika melaksanakan tugas yang dijadwalkan, pengguna harus melakukan absensi di toko yang dimaksud. Absensi dapat dilakukan dengan mengklik tombol absen, tombol absen akan muncul sesuai dengan jadwal yang ada dan berada di radius 50 meter dari toko yang dijadwalkan seperti pada gambar yang terlihat di Gambar 2.3.8. Untuk memastikan bahwa pengguna telah mengunjungi toko yang dijadwalkan, pengguna perlu melakukan tugas tertentu yang telah diberikan atasan, seperti *upload foto*.

### 2.3.5 Menggunakan API Public

Terdapat fitur pencarian lokasi pengguna saat mengakses aplikasi dan lokasi toko yang dijadwalkan pada aplikasi. Sehingga aplikasi ini mengakses peta dari Google API. Untuk menggunakan Google Maps API, dibutuhkan *key* yang didapat dari *Google Developer*. *Key* tersebut digunakan untuk mengakses peta pada aplikasi menggunakan Retrofit. Laman untuk meminta API *key* dapat dilihat pada Gambar 2.3.9, dengan mengklik tombol *create credentials*.



Gambar 2.3.9 Google Developer Console

## **BAB III**

### **HASIL PEMBELAJARAN**

#### **3.1 Manfaat Kerja Praktek**

Manfaat yang saya dapatkan selama melakukan kerja praktek di PT. XYZ adalah:

1. Mendapatkan pengalaman tentang dunia kerja.
2. Melatih kemampuan berinteraksi dengan atasan maupun rekan kerja dalam perusahaan.
3. Melatih kemampuan beradaptasi dalam lingkungan kerja.
4. Mengamati proses bisnis yang ada di perusahaan.
5. Menambah pengetahuan yang tidak didapatkan ketika kuliah.
6. Mengembangkan ilmu yang sudah dipelajari saat kuliah, sesuai dengan proyek yang diterima.

#### **3.2 Penerapan Ilmu dalam Kerja Praktek**

Beberapa ilmu yang didapatkan saat kuliah dan diterapkan dalam kerja praktek di PT. XYZ ialah:

1. Penerapan ilmu Pemrograman Web dalam pembuatan tampilan *website* menggunakan HTML5, seperti membuat tabel, *form*, *javascript*, *button*.
2. Penerapan ilmu Pemrograman Web Lanjut dalam pembuatan *website* baik secara *back-end* menggunakan *framework* Laravel 5.1, menggunakan bahasa PHP.
3. Penerapan ilmu Algoritma dan Pemrograman seperti :
  - Dasar-dasar pemrograman
  - Konsep pemrograman /Modul Sekuens
  - Modul Pemilihan

- Modul Perulangan
- Modul Prosedur
- Modul Fungsi
- Modul Array

Modul-modul tersebut digunakan untuk memprogram *website* maupun Android yang memerlukan dasar pemrograman.

#### 4. Pemrograman Berorientasi Objek yaitu :

- ❖ Modul Kelas dan Objek seperti membuat kelas/model pada *website* maupun Android.
- ❖ Modul Pewarisan yang digunakan untuk *extend* kelas lain di Android.
- ❖ Modul Perancangan Kelas 2 yang menggunakan konsep *interface* pada Retrofit.
- ❖ Modul *Exception Handling* yang dipakai untuk menangkap kemungkinan *error* yang akan terjadi saat program berjalan.
- ❖ Modul *Objek Persistence* yaitu membuat kelas *entity*/kelas model.

#### 5. Pemrograman Aplikasi Mobile dalam pembuatan *website* maupun aplikasi Siabo, yaitu:

- Modul *Activity and Intent*, untuk *intent* baru tiap laman aplikasi.
- Modul *User Interface*, yang memperhatikan *user interface* pada aplikasi.
- Modul *Web API*, yang mempelajari cara menggunakan API ke aplikasi.
- Modul *Geolocation*, yang mempelajari cara mendapatkan lokasi pengguna saat mengakses aplikasi melalui peta. Menggunakan API dari Google Maps.

- Modul *Using Library*, *library* yang menambahkan *widget* kalender pada aplikasi.

6. Penerapan ilmu Basis Data, pada:

- Modul Basis Data Relasional
- Modul Aljabar Relasional
- Modul Analisis Perancangan Basis Data





## **BAB IV**

### **KESIMPULAN DAN SARAN**

#### **4.1 Kesimpulan**

Kesimpulan dari kerja praktek yang dilakukan dari tanggal 4 Januari 2017 sampai dengan tanggal 14 Februari 2017 adalah:

1. Penulis mendapatkan gambaran tentang dunia kerja yang sebenarnya, bagaimana departemen IT memiliki bagian yang penting dalam perusahaan dan proses bisnisnya.
2. Penulis dapat menerima ilmu baru dan mengembangkan ilmu yang sudah didapatkan selama kuliah di Universitas Atma Jaya Yogyakarta.
3. Aplikasi Siabo yang dapat digunakan oleh karyawan Toko O untuk melakukan absensi pada tiap toko yang dijadwalkan.