



KERJA PRAKTIK - KI141330

CR Tool: Android Mobile Application

PT Kreator Solusi Informasi

Jalan Ngagel Jaya Selatan, Ruko Manyar Megah Indah

Plaza Blok F No. 4, Surabaya, Indonesia

Periode: 13 Juni - 13 Agustus 2016

Oleh:

Gian Sebastian Anjasmara

5113100132

Julio Anthony Leonard

5113100148

Pembimbing Jurusan

Prof. Ir. Handayani Tjandrasa, M.Sc., Ph.D.

Pembimbing Lapangan

Budi Condro.

JURUSAN TEKNIK INFORMATIKA

Fakultas Teknologi Informasi

Institut Teknologi Sepuluh Nopember

Surabaya 2016

RBTC



KERJA PRAKTIK - KI141330

CR Tool: Android Mobile Application

PT Kreator Solusi Informasi

Jalan Ngagel Jaya Selatan, Ruko Manyar Megah Indah

Plaza Blok F No. 4, Surabaya, Indonesia

Periode: 13 Juni - 13 Agustus 2016

Oleh:

Gian Sebastian Anasmara

5113100132

Julio Anthony Leonard

5113100148

Pembimbing Jurusan

Prof. Ir. Handayani Tjandrasa, M.Sc., Ph.D.

Pembimbing Lapangan

Budi Condro.

JURUSAN TEKNIK INFORMATIKA

Fakultas Teknologi Informasi

Institut Teknologi Sepuluh Nopember

Surabaya 2016

[Halaman ini sengaja dikosongkan]

RBTC

LEMBAR PENGESAHAN

KERJA PRAKTIK

CR Tool: Android Mobile Application

Oleh:

Gian Sebastian A.

5113100132

Julio Anthony Leonard

5113100148

Disetujui oleh Pembimbing Kerja Praktik:

1. Prof. Ir. Handayani Tjandrasa, M.Sc.,
Ph.D.
NIP.
(Pembimbing Jurusan)

2. Budi Condro.
NIP.
(Pembimbing Lapangan)

**SURABAYA
SEPTEMBER, 2016**

[Halaman ini sengaja dikosongkan]

RBTC

**Implementasi CR Tool:
Android Mobile Application**

Nama Mahasiswa : Gian Sebastian Anjasmara
NRP : 5113100132
Nama Mahasiswa : Julio Anthony Leonard
NRP : 5113100148
Jurusan : Teknik Informatika FTIf-ITS
Pembimbing Jurusan : Prof. Ir. Handayani Tjandrasa,
M.Sc., Ph.D.
Pembimbing Lapangan : Budi Condro.

RBTC

ABSTRAK

PT Axioo sebagai salah satu perusahaan yang bergerak di bidang IT di Indonesia terbagi menjadi beberapa divisi dalam struktur organisasinya, salah satunya yaitu divisi *sales* dan divisi Customer Relations (CR). Divisi CR adalah divisi yang bertanggung jawab melaporkan *sales out* (hasil penjualan *dealer* ke *end-user*) dari masing-masing outlet yang dibawahinya. Hingga saat ini proses pelaporan *sales out* dilakukan melalui aplikasi berbasis web, salah satu permasalahan dari proses tersebut adalah kurangnya sistem monitoring terhadap aktivitas dari CR. Selain itu, mengingat aktivitas CR memiliki tingkat mobilitas yang tinggi, dibutuhkan jenis aplikasi selain aplikasi web yang lebih mudah digunakan ketika faktor mobilitas menjadi pertimbangan.

CR Tool adalah sebuah aplikasi berbasis Android yang dikembangkan untuk memenuhi kebutuhan bisnis dari divisi CR dan divisi *sales* PT Axioo. CR Tool digunakan oleh CR agar dapat melakukan pelaporan *sales out*, melihat laporan *sales out* yang pernah dilaporkannya dan melihat pencapaiannya selama periode tertentu. Selain itu, CR Tool dapat mempermudah divisi *sales* untuk melakukan monitoring masing-masing CR.

Pada laporan kerja praktik ini, penulis akan menguraikan secara garis besar pengerjaan CR Tool yang secara langsung berhadapan dengan *user* / CR sebagai aktor utama. Metodologi yang digunakan adalah dengan melakukan perumusan masalah, studi literatur, analisis dan perancangan sistem, implementasi sistem, pengujian serta evaluasi sistem.

Berdasarkan hasil uji coba dan evaluasi menunjukkan bahwa CR Tool telah berhasil memenuhi semua kebutuhan sistem.

Kata kunci:

PT Axioo; CR Tool; Android; Customer Relations; sales out; laporan; pencapaian; monitoring

KATA PENGANTAR

Puji dan syukur kami panjatkan kepada Allah Yang Maha Besar karena atas berkat limpahan rahmat dan lindungan-Nya kami dapat melaksanakan salah satu kewajiban kami sebagai mahasiswa Teknik Informatika ITS yaitu Kerja Praktik (KP).

Kami menyadari masih terdapat banyak kekurangan baik dalam pelaksanaan kerja praktik maupun penyusunan buku laporan ini, namun kami berharap buku laporan ini dapat menambah wawasan pembaca dan dapat menjadi sumber referensi. Kami mengharapkan kritik dan saran yang membangun untuk kesempurnaan penulisan buku laporan ini.

Melalui laporan ini kami juga ingin menyampaikan rasa terima kasih kepada kepada orang-orang yang telah membantu dalam pelaksanaan kerja praktik hingga penyusunan laporan Kerja praktik baik secara lnsung maupun tidak langsung. Orang-orang tersebut antara lain adalah:

1. Kedua orang tua penulis.
2. Prof. Ir. Handayani Tjandrasa, M.Sc., Ph.D., selaku dosen pembimbing kerja praktik yang telah membimbing penulis selama kerja praktik berlangsung.
3. Bapak Radityo Anggoro, S.Kom., M.Sc., Dr.Eng., selaku koordinator kerja praktik.
4. Bapak Budi Condro S.SI., selaku pembimbing lapangan kami selama kerja praktik yang telah memberikan bimbingan serta ilmunya kepada penulis.

Surabaya, Agustus 2016

Gian Sebastian Anjasmara, Julio Anthony Leonard

[Halaman ini sengaja dikosongkan]

RBTC

DAFTAR ISI

LEMBAR PENGESAHAN.....	v
ABSTRAK	viii
KATA PENGANTAR.....	ix
DAFTAR ISI	xi
DAFTAR GAMBAR	xv
DAFTAR TABEL	xvii
DAFTAR KODE SUMBER	xix
1 BAB I PENDAHULUAN	21
1.1. Latar Belakang	21
1.2. Tujuan.....	22
1.3. Manfaat.....	22
1.4. Rumusan Permasalahan.....	22
1.5. Lokasi dan Waktu Kerja Praktik	23
1.6. Metodologi Kerja Praktik	23
1.7. Sistematika Laporan	25
2 BAB II PROFIL PERUSAHAAN	27
2.1. Sejarah Perusahaan.....	27
2.2. Visi dan Misi Perusahaan	28
2.3. Struktur Organisasi.....	28
2.4. Mobile Apps & Content	29
3 BAB III TINJAUAN PUSTAKA.....	31
3.1. Android Studio	31
3.2. Microsoft SQL Server 2016 Express.....	31
3.3. Draw.io	31
3.4. Volley	32
3.5. Google Play Services - Location	32
3.6. Barcode Scanner.....	32
3.7. MP Android Chart	32
3.8. PHP (Hypertext Processor)	32
3.9. JSON (JavaScript Object Notation).....	33
3.10. REST (REpresentational State Transfer) & RESTFul Web Services	33

3.11.	XML (eXtensible Markup Language)	33
3.12.	Java	34
3.13.	CodeIgniter	34
4	BAB IV ANALISIS DAN PERANCANGAN SISTEM.....	35
4.1.	Analisis Sistem	35
4.1.1.	Definisi Umum Aplikasi.....	35
4.1.2.	Analisis Kebutuhan Fungsional.....	35
4.2.	Perancangan Sistem.....	36
4.2.1.	Perancangan Basis Data.....	36
4.2.2.	Desain Sistem	37
5	BAB V IMPLEMENTASI SISTEM	45
5.1.	Tampilan Antarmuka User	45
5.1.1.	View Login.....	45
5.1.2.	View Home.....	46
5.1.3.	View Dialog Check In	47
5.1.4.	View Input SN.....	48
5.1.5.	View Scan SN.....	49
5.1.6.	View Report.....	50
5.1.7.	View Achievement	51
5.1.8.	View Sales Out	52
5.2.	Implementasi Antarmuka dan Logic	54
5.2.1.	Implementasi Sistem <i>Login</i>	54
5.2.2.	Implementasi Sistem <i>Check In</i>	60
5.2.3.	Implementasi Sistem Pelaporan <i>Sales Out</i>	77
5.2.4.	Implementasi Sistem Menampilkan Laporan <i>Sales Out</i>	90
5.2.5.	Implementasi Sistem Menampilkan Pencapaian CR	128
5.3.	Implementasi Web Services	140
5.3.1.	Implementasi Controller Services.php.....	140
5.3.2.	Implementasi Model Services_model.php.....	148
	BAB VI PENGUJIAN DAN EVALUASI	153
6	153	
6.1.	Tujuan Pengujian	153
6.2.	Kriteria Pengujian.....	153

6.3.	Lingkungan Pengujian.....	154
6.4.	Skenario Pengujian.....	154
6.5.	Evaluasi Pengujian	155
7	BAB VII KESIMPULAN DAN SARAN	157
7.1.	Kesimpulan.....	157
7.2.	Saran.....	157
	DAFTAR PUSTAKA.....	159
	LAMPIRAN	161
	BIODATA PENULIS I	163
	BIODATA PENULIS II.....	165

RBTC

[Halaman ini sengaja dikosongkan]

RBTC

DAFTAR GAMBAR

Gambar 2.3-1 Struktur Organisasi.....	29
Gambar 4.1.2-1 Diagram Kasus Penggunaan.....	36
Gambar 4.2.1-1 Physical Data Model (PDM) CR Tool	37
Gambar 4.2.2-1 Diagram Aktivitas <i>Check In</i>	38
Gambar 4.2.2-2 Diagram Aktivitas Pelaporan <i>Sales Out</i>	40
Gambar 4.2.2-3 Diagram Aktivitas Menampilkan Laporan <i>Sales Out</i>	42
Gambar 4.2.2-4 Diagram Aktivitas Menampilkan Pencapaian CR	44
Gambar 5.1.1-1 View Login.....	45
Gambar 5.1.2-1 View Home	46
Gambar 5.1.3-1 View Dialog Check In.....	47
Gambar 5.1.4-1 View Input SN.....	48
Gambar 5.1.5-1 View Scan SN	49
Gambar 5.1.6-1 View Report (a), View Memilih Rentang Tanggal (b)	50
Gambar 5.1.7-1 View Achievement (a), View Detil Achievement (b)	51
Gambar 5.1.8-1 View Sales Out (a)(b)(c)(d).....	53

[Halaman ini sengaja dikosongkan]

RBTC

DAFTAR TABEL

Tabel 6.5-1 Hasil Pengujian	155
-----------------------------------	-----

RBTC

[Halaman ini sengaja dikosongkan]

RBTC

DAFTAR KODE SUMBER

Kode Sumber 5.2.1-1 activity_login.xml	55
Kode Sumber 5.2.1-2 LoginActivity.java	58
Kode Sumber 5.2.1-3 SessionManager.java.....	60
Kode Sumber 5.2.2-1 activity_home.xml.....	61
Kode Sumber 5.2.2-2 HomeActivity.java	69
Kode Sumber 5.2.2-3 GoogleLocationRequest.java	74
Kode Sumber 5.2.2-4 fragment_select_outlet_dialog.xml	75
Kode Sumber 5.2.2-5 SelectOutletDialogFragment.java	77
Kode Sumber 5.2.3-1 activity_sales_out.xml.....	78
Kode Sumber 5.2.3-2 fragment_sales_out_input.xml.....	81
Kode Sumber 5.2.3-3 SalesOutActivity.java	84
Kode Sumber 5.2.3-4 SalesOutListener.java	84
Kode Sumber 5.2.3-5 SalesOutInputFragment.java.....	88
Kode Sumber 5.2.3-6 SalesOutScanFragment.java	90
Kode Sumber 5.2.4-1 activity_report.xml.....	90
Kode Sumber 5.2.4-2 ReportActivity.java	92
Kode Sumber 5.2.4-3 fragment_report_main.xml	94
Kode Sumber 5.2.4-4 ReportMainFragment.java	99
Kode Sumber 5.2.4-5 activity_report_sales_out.xml	100
Kode Sumber 5.2.4-6 ReportSalesOutActivity.java.....	108
Kode Sumber 5.2.4-7 ViewPagerAdapter.java	109
Kode Sumber 5.2.4-8 adapter_history_sales_out_row.xml	110
Kode Sumber 5.2.4-9 HistorySalesOutAdapter.java.....	111
Kode Sumber 5.2.4-10 fragment_report_sales_out.xml.....	112
Kode Sumber 5.2.4-11 ReportSalesOutByDateFragment.java .	115
Kode Sumber 5.2.4-12 ReportSalesOutByOutletFragment.java	118
Kode Sumber 5.2.4-13 fragment_history_sales_out.xml	118
Kode Sumber 5.2.4-14 ReportHistorySalesOutFragment.java .	123
Kode Sumber 5.2.4-15 fragment_serial_number_sales_out.xml	124

Kode	Sumber	5.2.4-16
ReportSerialNumberSalesOutFragment.java.....		
		127
Kode Sumber 5.2.5-1	fragment_report_track_record.xml 129
Kode Sumber 5.2.5-2	ReportTrackRecordFragment.java 135
Kode Sumber 5.2.5-3	fragment_track_record_detail_dialog.xml 137
Kode Sumber 5.2.5-4	TrackRecordDialogFragment.java 139
Kode Sumber 5.3.1-1	Services.php 147
Kode Sumber 5.3.2-1	Services_model.php 151

RBTC

BAB I

PENDAHULUAN

1.1. Latar Belakang

Teknologi menawarkan banyak kemudahan dalam dunia usaha. Persebaran informasi yang cepat dan kemudahan dalam melakukan aktivitas dapat membantu meningkatkan efisiensi terlaksananya proses-proses bisnis yang terlibat dalam suatu aktivitas tersebut. Seiring dengan berkembangnya dunia usaha di Indonesia, dibutuhkan teknologi penunjang yang tepat guna dan dapat mengelola dengan baik sumber daya yang digunakan.

PT Axioo sebagai salah satu perusahaan yang bergerak di bidang IT di Indonesia terbagi menjadi beberapa divisi dalam struktur organisasinya, salah satunya yaitu divisi *sales* dan divisi Customer Relations (CR). Divisi *sales* membawahi divisi CR dan mengelola masing-masing CR yang menjadi tanggung jawabnya. Sedangkan divisi CR adalah divisi yang bertanggung jawab melaporkan *sales out* (hasil penjualan *dealer* ke *end-user*) dari masing-masing outlet yang dibawahinya. Hingga saat ini proses pelaporan *sales out* dilakukan melalui aplikasi berbasis web, salah satu permasalahan dari proses tersebut adalah kurangnya sistem monitoring terhadap aktivitas dari CR. Proses bisnis yang benar pada saat pelaporan *sales out* mewajibkan CR untuk mengunjungi outlet yang dikelolanya secara langsung dan rutin. Selain itu, mengingat aktivitas CR memiliki tingkat mobilitas yang tinggi, dibutuhkan jenis aplikasi selain aplikasi web yang lebih mudah digunakan ketika faktor mobilitas menjadi pertimbangan.

CR Tool adalah sebuah aplikasi berbasis Android yang dikembangkan untuk menjawab permasalahan di atas sekaligus memenuhi kebutuhan bisnis dari divisi CR dan divisi *sales*. CR Tool digunakan oleh CR agar dapat melakukan

pelaporan *sales out*, melihat laporan *sales out* yang pernah dilaporkannya dan melihat pencapaiannya selama periode tertentu. Selain itu, CR Tool dapat mempermudah divisi *sales* untuk melakukan monitoring masing-masing CR.

1.2. Tujuan

Tujuan kerja praktik ini adalah mengimplementasikan CR Tool guna memenuhi kebutuhan bisnis dari divisi CR dan divisi *sales*. Kebutuhan tersebut antara lain adalah:

- Membantu CR dalam melakukan pelaporan *sales out* di masing-masing outlet yang dikelolanya.
- Mempermudah divisi *sales* untuk melakukan monitoring masing-masing CR.

1.3. Manfaat

Berikut manfaat yang diperoleh melalui kerja praktik dalam pengembangan CR Tool:

- Memperdalam pemahaman tentang proses perancangan perangkat lunak berbasis mobile.
- Mengetahui kondisi nyata dari kasus-kasus yang dipelajari saat kuliah.
- Melatih berpikir secara tepat untuk menyelesaikan suatu masalah.
- Berpikir inovatif dan solutif untuk memberikan inovasi baru terhadap suatu solusi permasalahan.

1.4. Rumusan Permasalahan

Berikut rumusan masalah dalam pelaksanaan kerja praktik:

- Bagaimana merancang sistem CR Tool agar dapat memenuhi semua kebutuhan yang ditentukan.
- Bagaimana menerapkan bahasa pemrograman XML sebagai *frontend* / *user interface* dari CR Tool agar mudah dipahami dan digunakan oleh *user*.

- Bagaimana menerapkan bahasa pemrograman Java sebagai *backend* dari CR Tool agar aplikasi berjalan sesuai fungsi yang diinginkan.
- Bagaimana menerapkan bahasa pemrograman PHP dan JSON untuk membuat *web service* dari CR Tool.

1.5. Lokasi dan Waktu Kerja Praktik

Kerja praktik dilaksanakan pada waktu dan tempat sebagai berikut:

Lokasi : PT Kreator Solusi Informasi
 Alamat : Jalan Ngagel Jaya Selatan, Ruko Manyar
 Megah Indah Plaza Blok F No. 4, Surabaya,
 Indonesia
 Waktu : 13 Juni – 13 Agustus 2016

1.6. Metodologi Kerja Praktik

1. Perumusan Masalah

Untuk mengetahui domain dan fungsionalitas, dijelaskan secara rinci bagaimana sistem yang harus dibuat. Penjelasan oleh pembimbing kerja praktik kali ini menghasilkan kebutuhan yang harus dipenuhi oleh sistem dan gambaran sistem secara garis besar. Setelah mendapatkan gambaran sistem, diskusi lebih lanjut dilakukan guna menentukan rancangan serta *tools* pendukung pembuatan sistem.

2. Studi Literatur

Pada tahap ini, setelah ditentukannya *database*, bahasa pemrograman sampai dengan *framework* beserta *tools* tambahan yang digunakan, dilakukan studi literatur lanjut mengenai bagaimana penggunaannya dalam membangun sistem sesuai yang diharapkan.

Secara garis besar, karena sistem adalah *mobile-based* maka digunakan Android Studio sebagai Integrated

Development Environment (IDE) yang menggunakan bahasa pemrograman XML dan Java untuk pengembangan sistem. Sedangkan untuk *web services*, digunakan *framework* CodeIgniter dengan bahasa pemrograman PHP. Kemudian untuk DBMS digunakan Microsoft SQL Server Express Edition sesuai permintaan pembimbing kerja praktik.

3. Analisis dan Perancangan Sistem

Langkah ini meliputi penjelasan awal tentang sistem. Bagaimana cara kerja sistem dengan skenario tertentu. Dari penjelasan awal telah didapatkan beberapa kebutuhan fungsional secara garis besar. Kemudian dilanjutkan dengan memperjelas dan menspesifikan kebutuhan-kebutuhan tersebut. Dibuatlah diagram kasus penggunaan dan aktivitas yang mewakili skenario-skenario untuk penggunaan sistem CR Tool.

4. Implementasi Sistem

Implementasi sistem didasarkan oleh perancangan dan analisis sebelumnya. Semua didasari pada rancangan kasus penggunaan dan penentuan *tools* yang telah dilakukan sebelumnya. Penentuan tipe data saat pembuatan database disesuaikan juga dengan kebutuhan.

Pengerjaan dilakukan dengan progres setiap 7 hari. Jika ada masukan maka akan segera diaplikasikan karena pengerjaan dengan metode *agile programming* yang memungkinkan melakukan perubahan dengan cepat.

5. Pengujian dan Evaluasi

Pengujian dilakukan dengan menguji fitur dan antarmuka sistem CR Tool. Kesesuaian sistem dengan kebutuhan akan menentukan keberhasilan dari pengujian. Sehingga bisa di-evaluasi apakah sistem sudah sesuai dengan tujuan dibuatnya CR Tool itu sendiri atau belum.

6. Kesimpulan dan Saran

Setelah melakukan pengujian, kesimpulan dan saran yang didapatkan untuk aplikasi CR Tool antara lain adalah:

- Perancangan dan implementasi sistem CR Tool telah memenuhi semua kebutuhan yang ditentukan.
- *Frontend / user interface* dari CR Tool yang dibuat menggunakan bahasa pemrograman XML cukup mudah dipahami dan digunakan oleh *user*.
- *Backend* dari CR Tool yang dibuat menggunakan bahasa pemrograman Java telah berjalan sesuai fungsi yang diinginkan.
- *Web services* dari CR Tool yang dibuat menggunakan bahasa pemrograman PHP dan JSON telah berjalan dengan cukup handal ketika layanannya dibutuhkan oleh aplikasi.
- Saran mengenai *frontend* harus di-optimasi agar lebih *user friendly* pada seluruh Smartphone dengan *platform* Android.

1.7. Sistematika Laporan

Laporan kerja praktik ini terdiri dari 7 bab, dengan rincian sebagai berikut:

1. Bab I : Pendahuluan

Bab ini berisi tentang latar belakang masalah, tujuan, manfaat, rumusan masalah, lokasi dan waktu kerja praktik, metodologi dan sistematika laporan.

2. Bab II : Profil Perusahaan

Bab ini berisi tentang PT Kreator Solusi Informasi, membahas tentang sejarah perusahaan, visi dan misi,

struktur organisasi dan divisi tempat dilaksanakannya kerja praktek.

3. Bab III : Tinjauan Pustaka

Bab ini berisi penjelasan pustaka-pustaka yang digunakan dalam pembuatan aplikasi.

4. Bab IV : Analisis dan Perancangan Sistem

Bab ini membahas tentang proses analisa kebutuhan berdasarkan kondisi yang sesungguhnya dan perancangannya yang meliputi desain aplikasi yang akan dikembangkan. Proses analisa dan desain aplikasi menghasilkan daftar fitur dan diagram alur aplikasi.

5. Bab V : Implementasi Sistem

Bab ini membahas tentang tampilan antarmuka *user*, implementasi antarmuka dan *logic*, dan implementasi *web services*.

6. Bab VI : Pengujian dan Evaluasi

Bab ini membahas tentang lingkungan pengujian, skenario pengujian dan evaluasi pengujian setelah aplikasi selesai dikembangkan.

7. Bab VII : Kesimpulan dan Saran

Bab ini berisi tentang kesimpulan dan saran yang didapatkan dari tugas selama kerja praktik.

BAB II

PROFIL PERUSAHAAN

2.1. Sejarah Perusahaan

Axioo adalah merek kelas dunia dari produsen perangkat ICT dengan kemampuan desain rumah dan fasilitas produksi yang terintegrasi di Indonesia. PT Axioo memasuki pasar untuk teknologi komputasi portabel di Indonesia pada tahun 2004, yang menawarkan persimpangan unik antara kinerja, estetika dan aksesibilitas. Axioo terus mengembangkan diri sebagai produsen perangkat teknologi dan menjadi bagian dari sejarah perkembangan teknologi di Indonesia dengan menjadi merek lokal pertama yang meluncurkan Notebook di Indonesia. Sejak diperkenalkannya sistem operasi baru untuk ponsel pada tahun 2009 silam yang bernama Android, Axioo yang awalnya sebagai produsen Notebook pun mulai berinvestasi ke pasar Smartphone dan Tablet untuk melebarkan sayap bisnis dengan diversifikasi produk.

Inovasi terus dilakukan Axioo untuk memberikan produk terbaiknya ke pasar Indonesia. Hal ini dibuktikan dengan banyaknya varian perangkat dan produk mulai dari *low-end* hingga *high-end* yang telah diluncurkan sampai saat ini. Melalui berbagai produk yang telah dipasarkan di seluruh Indonesia, Axioo pada tahun 2015 mendapatkan penghargaan sebagai *The 1st Champion of Indonesia Original Brand 2015*, produk kategori Local Tablet. Ekonomis namun inovatif, sederhana namun luwes, model yang beragam serta garis kinerja tinggi dari produk-produk berkualitas dengan cepat meluncurkan nama Axioo di Indonesia. Produk utama Axioo hingga tahun 2016 adalah Notebook, Smartphone, Tablet serta Desktop PC dan Peripherals.

PT Axioo memiliki anak perusahaan yang bergerak di bidang *software development* yaitu PT Kreator. Kreator adalah

perusahaan yang terdiri dari sekumpulan profesional di bidang teknologi informasi yang telah berdiri sejak tahun 2009 di Surabaya. Konsep Kreator adalah mempermudah bisnis dengan bisnis proses yang lebih baik sehingga bisnis tumbuh menjadi bisnis yang lebih sehat dan kompetitif. Pada Kreator kualitas adalah hal yang sangat diutamakan, karena tanpa suatu bukti kualitas, pelanggan tidak akan mendapatkan *service excellent* yang diharapkan.

Kreator fokus dalam membangun bisnis di area UKM terutama pada sektor retail, industri, dan edukasi. Memperhatikan bahwa pada segmen-segmen tersebut sangat dibutuhkan adanya sistem informasi yang mudah digunakan, efisien, harga yang terjangkau, dan layanan yang cepat baik mulai dari *pre-sales* hingga *post implementation*. Kreator telah menuai sukses di segmen market usaha kecil menengah ditandai dengan telah digunakannya jasa Kreator di berbagai penjuru nusantara. Jasa tersebut berupa perangkat lunak yang telah dihasilkan oleh Kreator, diantaranya terdapat myBiz dan SAP Business One.

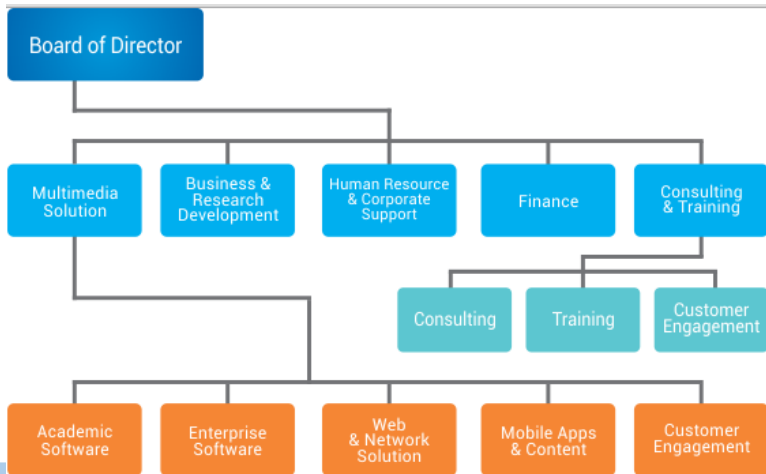
2.2. Visi dan Misi Perusahaan

Visi dari Kreator adalah menjadi perusahaan software development yang dinamis dengan didukung tenaga profesional di bidangnya untuk memberikan solusi terbaik bagi channel kami dan masyarakat.

Misi dari Kreator yaitu menjadi perusahaan yang tumbuh secara sehat melalui produk-produk solusi IT yang dapat diterima oleh masyarakat dan memotivasi seluruh elemen didalamnya untuk berkreasi secara berkesinambungan dalam mengirimkan solusi terbaik sesuai dengan perkembangan teknologi dan kebutuhan masyarakat.

2.3. Struktur Organisasi

Struktur organisasi dari Kreator dapat dilihat pada gambar dibawah



Gambar 4.1.1-1 Struktur Organisasi

2.4. Mobile Apps & Content

Pada kesempatan kerja praktik kali ini, kami mendapatkan pekerjaan pada divisi Mobile Apps & Content dan mengembangkan aplikasi mobile berbasis Android. Skill yang menjadi persyaratan untuk masuk ke divisi ini antara lain: Object Oriented Programming, Java, XML, Dokumentasi, Desain, Github, dan *Communication Skill*.

[Halaman ini sengaja dikosongkan]

RBTC

BAB III

TINJAUAN PUSTAKA

3.1. Android Studio

Android Studio [1] adalah Integrated Development Environment (IDE) resmi untuk Android. Tujuan dibangunnya Android Studio adalah untuk mempercepat pengembangan aplikasi dan membantu membangun aplikasi berkualitas tinggi untuk setiap perangkat Android. Android Studio menawarkan banyak fitur yang dapat meningkatkan produktivitas ketika membangun aplikasi Android, diantaranya adalah sistem pembangunan yang fleksibel berdasarkan Gradle, adanya emulator yang cepat dan kaya fitur, Instant Run untuk *deploy* perubahan pada aplikasi yang sedang berjalan tanpa membangun APK baru, integrasi dengan GitHub, dll.

3.2. Microsoft SQL Server 2016 Express

Microsoft SQL Server Express [2] adalah versi Relational Database Management System (RDBMS) milik Microsoft yang bebas diunduh, didistribusikan dan digunakan. Fitur-fitur yang ditawarkan hampir sama dengan versi berbayar, namun memiliki beberapa keterbatasan teknis dalam hal besar database (10 GB), spesifikasi perangkat keras (CPU tunggal, RAM 1 GB) dan tidak ada servis SQL Server Agent. Versi ini ditujukan khusus untuk aplikasi *embedded* dan berskala kecil.

3.3. Draw.io

Draw.io [3] adalah sebuah aplikasi diagram berbasis web yang dibangun di atas mxGraph, yang terintegrasi dengan Google Drive, OneDrive dan Dropbox. Dengan teknik *drag-n-drop* sederhana, aplikasi yang mudah digunakan ini menyediakan metode untuk merancang desain diagram yang dapat digunakan secara gratis oleh setiap orang.

3.4. Volley

Volley [4] adalah *networking library* yang dikembangkan oleh Google dan diperkenalkan pada Google I/O 2013. Volley dikembangkan karena ketiadaan *networking class* di Android yang dapat bekerja tanpa mengganggu pengalaman pengguna.

3.5. Google Play Services - Location

Google Play Services [5] adalah kumpulan *background services* dan APIs untuk perangkat Android. Servis yang tersedia antara lain Google Play Game, Location, Maps, dll. Location APIs adalah servis yang berhubungan dengan teknologi lokasi, diantaranya menyediakan Geofencing APIs untuk penjadwalan tindakan khusus pada pengguna yang memasuki atau meninggalkan batas-batas geografis tertentu, dan Fused Location Provider untuk memperoleh informasi lokasi pengguna dengan penggunaan daya seminimal mungkin.

3.6. Barcode Scanner

Barcode Scanner [6] adalah *library* Android buatan dm77 yang menyediakan kemudahan dalam membuat *view* untuk melakukan pemindaian *barcode*. *Library* ini berbasis ZXing dan ZBar yang bersifat gratis dan *extensible*.

3.7. MP Android Chart

MP Android Chart [7] adalah *library* Android buatan Philipp Jahoda yang merupakan *view / graph library*, mendukung grafik berbentuk *line, bar, pie, radar, bubble* dan *candlestick* serta adanya fitur *scrolling, dragging*, dan animasi.

3.8. PHP (Hypertext Processor)

PHP [8] adalah bahasa *scripting server-side* yang dirancang untuk pengembangan web namun juga digunakan sebagai bahasa pemrograman yang bersifat general. PHP

disebut bahasa pemrograman *server-side* karena PHP diproses pada komputer server. PHP dapat digunakan secara gratis dan bersifat *open source*. PHP ini digunakan untuk membangun RESTful *web services* yang menggunakan protokol HTTP dalam komunikasi data.

3.9. JSON (JavaScript Object Notation)

JSON [9] adalah format pertukaran data yang ringan, mudah dibaca dan ditulis oleh manusia, serta mudah diterjemahkan dan digenerate oleh komputer. JSON digunakan sebagai format pertukaran data dalam RESTful *web services*.

3.10. REST (REpresentational State Transfer) & RESTful Web Services

REST [10] adalah arsitektur standar untuk web yang menggunakan protokol HTTP dalam komunikasi data dan umumnya menggunakan format JSON dalam pertukaran data. *Web services* adalah kumpulan protokol dan standar yang digunakan untuk pertukaran data antar aplikasi. *Web services* yang berbasis REST disebut dengan RESTful *web services*. Pada RESTful *web services*, sebuah REST *Server* menyediakan akses ke sumber daya (misal *database*) dan sebuah REST *Client* mengakses dan mengolah sumber daya tersebut.

3.11. XML (eXtensible Markup Language)

XML [11] adalah bahasa *markup* yang mendefinisikan sebuah kumpulan peraturan untuk dokumen *encoding* dalam format yang bersifat *human-readable* dan *machine-readable*. XML digunakan untuk membuat *user interface layouts* pada aplikasi Android. Keuntungan menggunakan XML dalam membuat *user interface* aplikasi Android adalah mudah untuk dibaca dan dimanipulasi baik oleh manusia maupun komputer, selain itu memungkinkan adanya pemisahan antara penyajian aplikasi dengan kode yang mengontrol perilakunya.

3.12. Java

Java [12] adalah bahasa pemrograman yang bersifat *class-based*, *object-oriented*, dan dirancang secara spesifik agar memiliki ketergantungan implementasi seminimal mungkin. Java dimaksudkan untuk memungkinkan pengembang aplikasi melakukan “write once, run anywhere”, yang berarti kode Java yang telah di-*compile* dapat berjalan pada semua *platform* yang mendukung Java tanpa melakukan *compile* ulang. Java digunakan untuk membangun *backend* pada aplikasi Android. Keuntungan menggunakan Java dalam membangun *backend* aplikasi Android adalah merupakan bahasa pemrograman yang cukup dikenal para pengembang dan industri, berjalan pada Virtual Machine (VM) sehingga tidak perlu melakukan *compile* ulang untuk setiap Smartphone dan memiliki koleksi *development tools* dalam jumlah yang cukup besar.

3.13. CodeIgniter

CodeIgniter [13] adalah kerangka kerja PHP yang kuat dengan konsumsi memori yang rendah, dibangun untuk pengembang yang membutuhkan *toolkit* yang sederhana dan elegan untuk membuat aplikasi web. CodeIgniter menggunakan arsitektur Model-View-Controller (MVC), namun hanya kelas Controller yang wajib diimplementasikan, kelas View dan Model bersifat opsional.

BAB IV

ANALISIS DAN PERANCANGAN SISTEM

4.1. Analisis Sistem

4.1.1. Definisi Umum Aplikasi

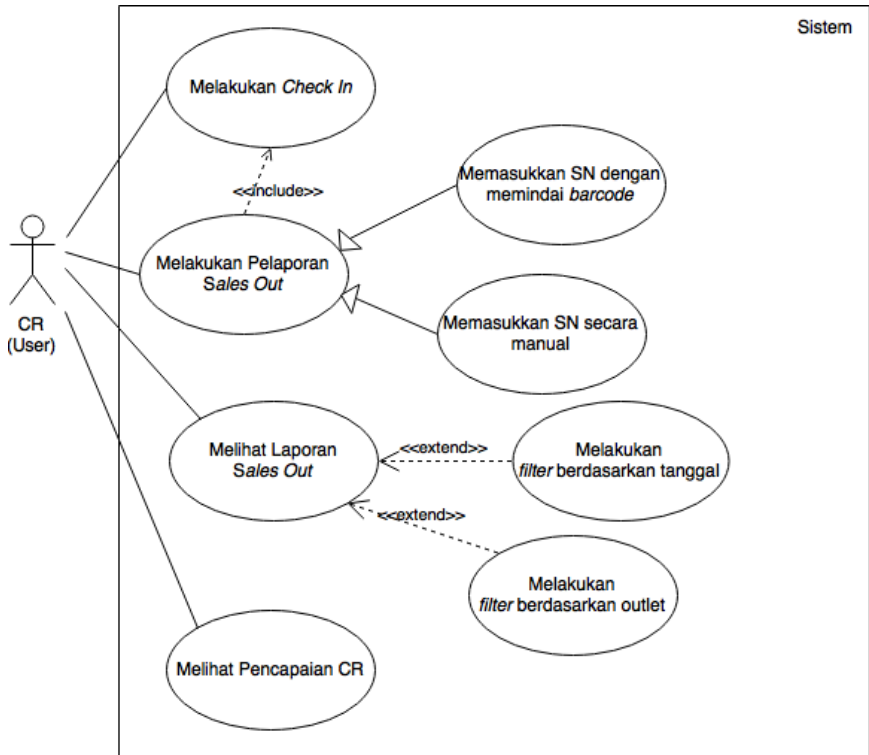
Secara umum, CR Tool merupakan sebuah aplikasi berbasis Android yang membantu CR (*user*) dalam melakukan pelaporan *sales out* di masing-masing outlet yang dikelolanya dan mempermudah divisi *sales* untuk melakukan monitoring masing-masing CR. CR Tool digunakan oleh CR agar dapat melakukan pelaporan *sales out*, melihat laporan *sales out* yang pernah dilaporkannya dan melihat pencapaiannya selama periode tertentu.

4.1.2. Analisis Kebutuhan Fungsional

Beberapa kebutuhan fungsional dalam pengembangan aplikasi CR Tool antara lain :

- a. **(F-001) Melakukan *Check In***
CR dapat melakukan *check in* pada outlet yang ingin dilaporkan *sales out* nya.
- b. **(F-002) Melakukan Pelaporan *Sales Out***
CR dapat melakukan pelaporan *sales out* dengan memasukkan Serial Number (SN) dari produk yang di *sales out*. SN dapat dimasukkan secara manual atau dengan memindai *barcode* milik produk.
- c. **(F-003) Melihat Laporan *Sales Out***
CR dapat melihat laporan *sales out* (SN dari produk yang di *sales out* dan status pelaporan) yang telah dilaporkannya selama periode tertentu. Laporan yang ditampilkan dapat di-*filter* berdasarkan tanggal atau outlet.
- d. **(F-004) Melihat Pencapaian CR**
CR dapat melihat pencapaiannya selama 6 bulan terakhir. Pencapaian tersebut berupa jumlah pelaporan *sales out* yang dikategorikan berdasarkan status

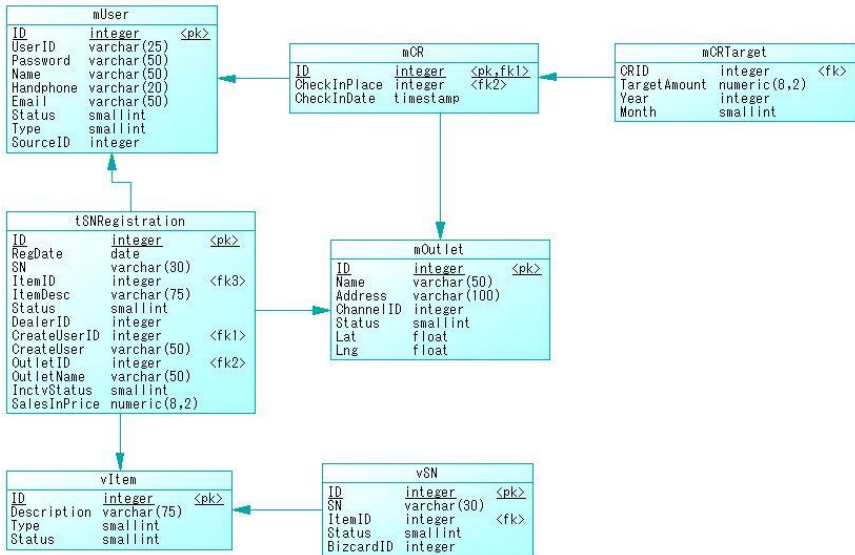
pelaporan (*submitted, received, approved, retur*) dan jumlah target (Rp. dan %) yang dicapai pada bulan tersebut.



Gambar 4.1.2-1 Diagram Kasus Penggunaan

4.2. Perancangan Sistem

4.2.1. Perancangan Basis Data



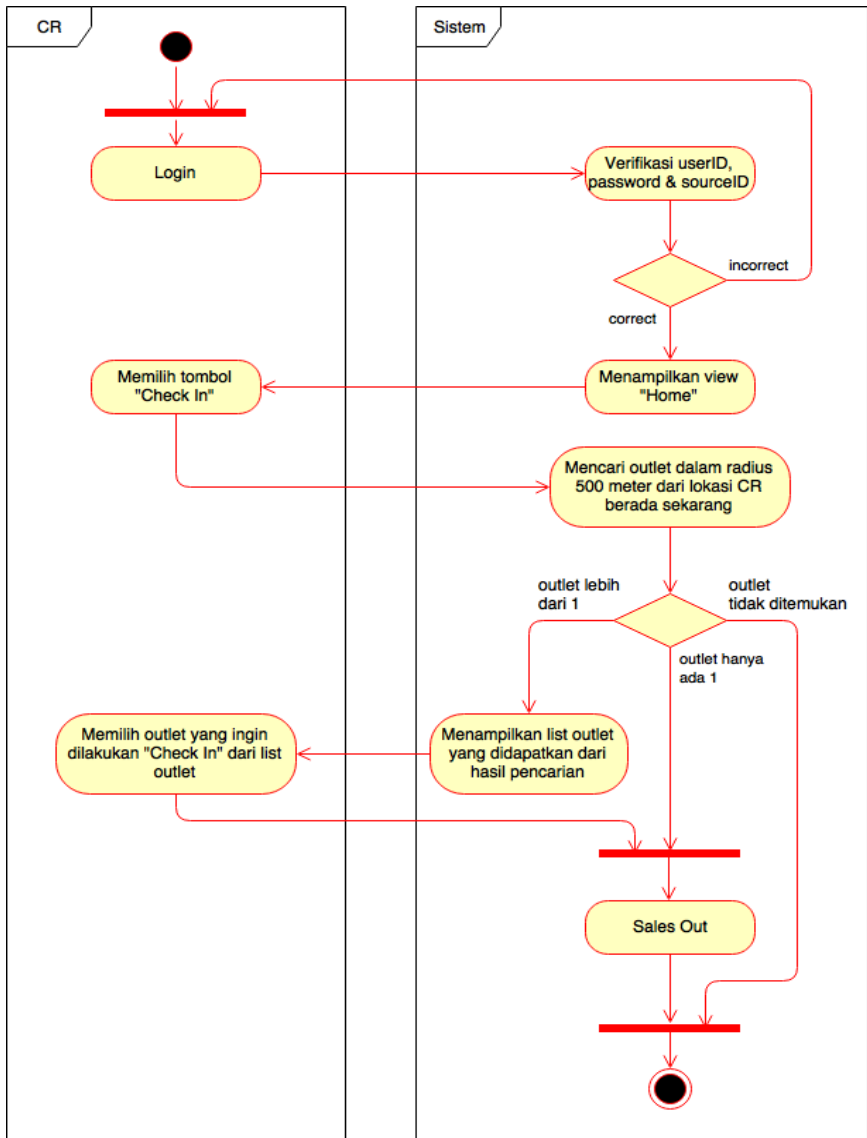
Gambar 4.2.1-1 Physical Data Model (PDM) CR Tool

4.2.2. Desain Sistem

Desain sistem digunakan untuk mengetahui jalannya proses bisnis pada suatu aplikasi sehingga pengembang aplikasi dapat dengan mudah melakukan perubahan. Desain sistem yang digunakan adalah *Use Case Diagram* dan *Activity Diagram*. *Use Case Diagram* menunjukkan proses bisnis yang dikerjakan sistem sedangkan *Activity Diagram* menunjukkan bagaimana sistem melakukan respon terhadap *input* dari *user*.

4.2.2.1. Desain Sistem Check In

Pada fitur ini, CR dapat melakukan *check in* pada outlet yang ingin dilaporkan *sales out* nya. Untuk lebih jelasnya mengenai alur *check in* akan dijelaskan pada diagram aktivitas di bawah ini :

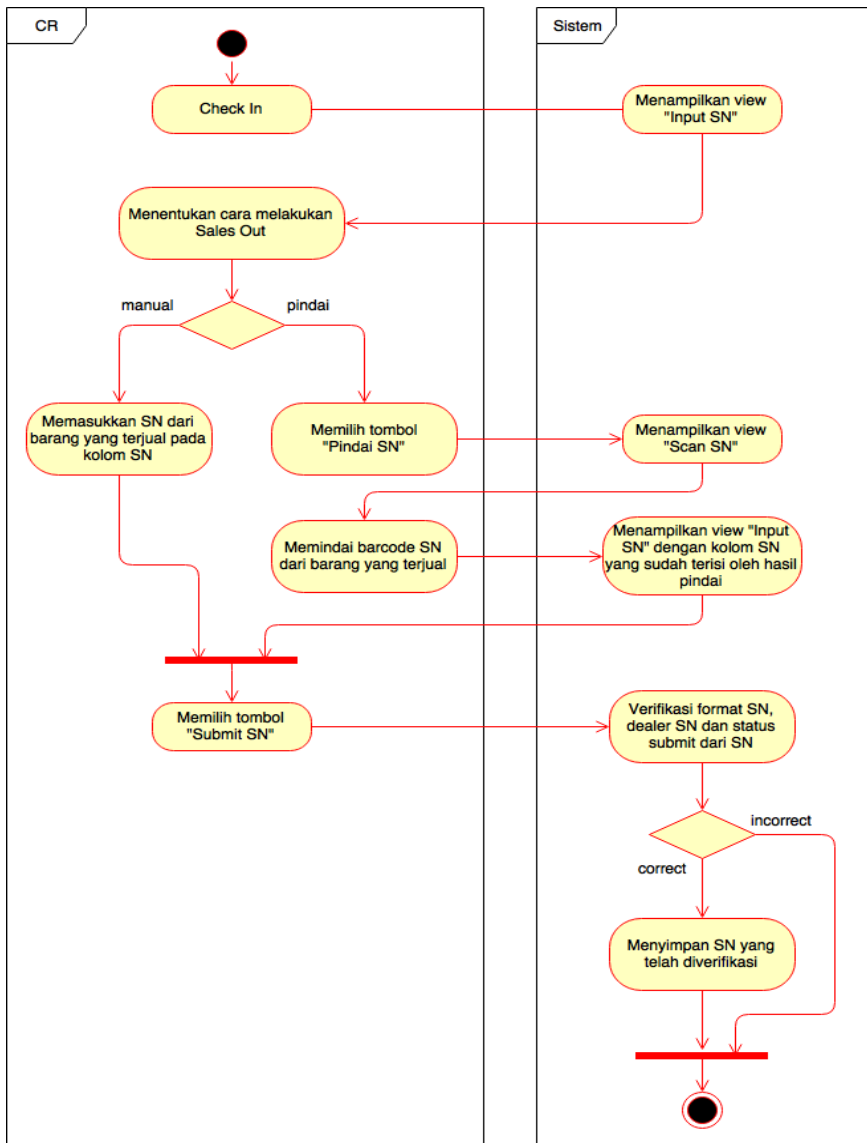


Gambar 4.2.2-1 Diagram Aktivitas *Check In*

4.2.2.2. Desain Sistem Pelaporan *Sales Out*

Pada fitur ini, CR dapat melakukan pelaporan *sales out* dengan memasukkan Serial Number (SN) dari produk yang di *sales out*. SN dapat dimasukkan secara manual atau dengan memindai *barcode* milik produk. Untuk lebih jelasnya mengenai alur pelaporan *sales out* akan dijelaskan pada diagram aktivitas di bawah ini :

RBTC

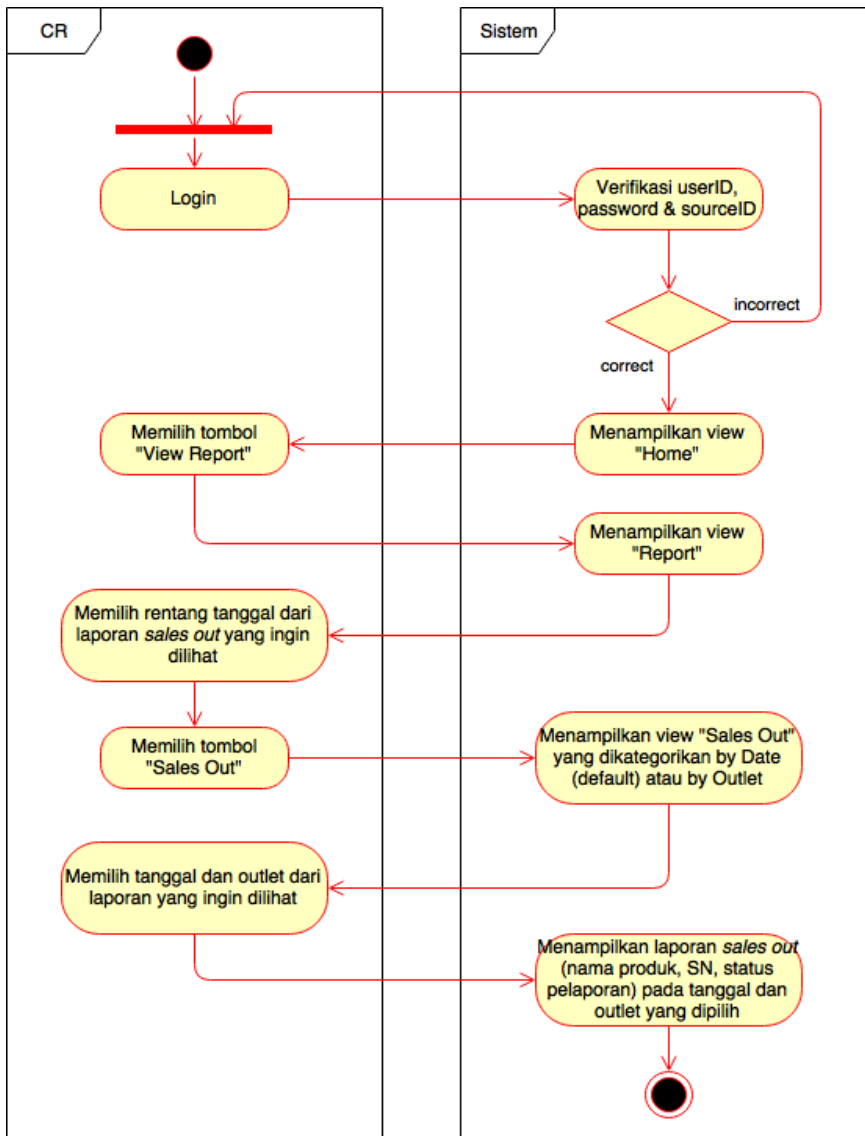


Gambar 4.2.2-2 Diagram Aktivitas Pelaporan *Sales Out*

4.2.2.3. Desain Sistem Menampilkan Laporan *Sales Out*

Pada fitur ini, CR dapat melihat laporan *sales out* (SN dari produk yang di *sales out* dan status pelaporan) yang telah dilaporkannya selama periode tertentu. Laporan yang ditampilkan dapat dapat di-*filter* berdasarkan tanggal atau outlet. Untuk lebih jelasnya mengenai alur menampilkan laporan *sales out* akan dijelaskan pada diagram aktivitas di bawah ini :

RBTC

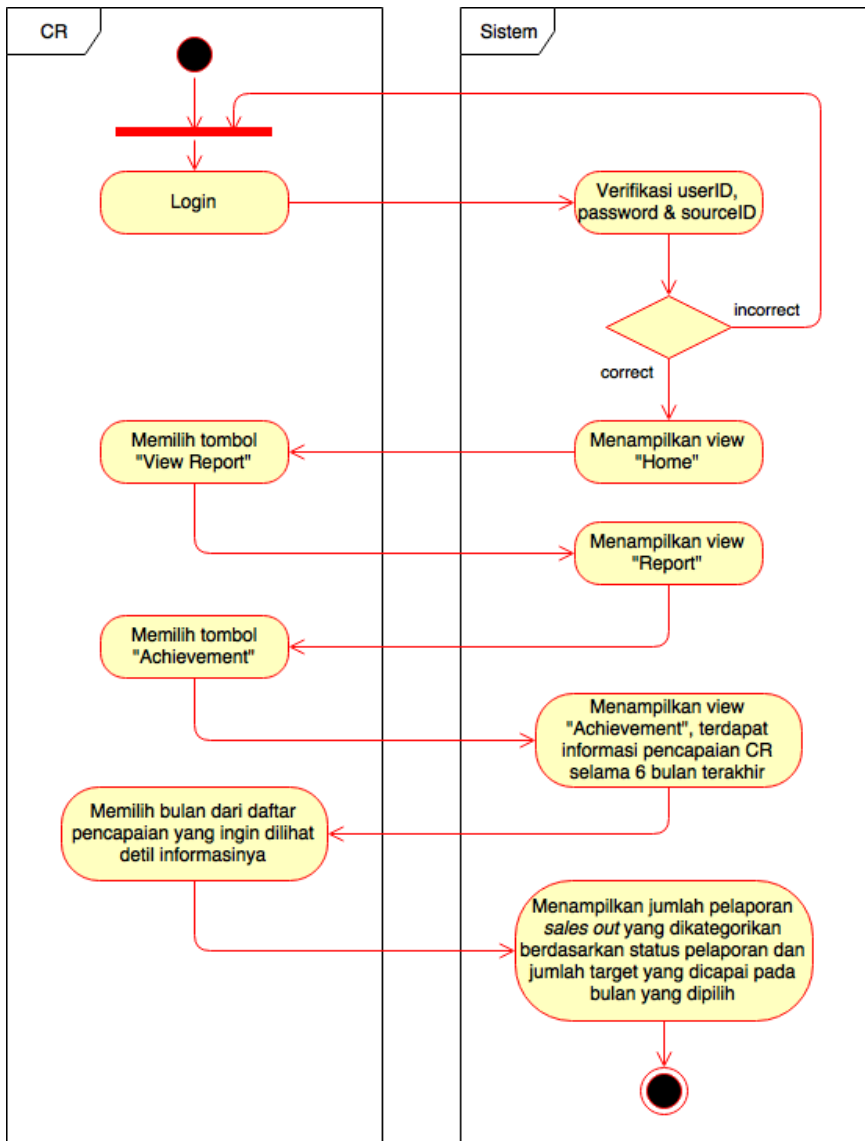


Gambar 4.2.2-3 Diagram Aktivitas Menampilkan Laporan *Sales Out*

4.2.2.4. Desain Sistem Menampilkan Pencapaian CR

Pada fitur ini, CR dapat melihat pencapaiannya selama 6 bulan terakhir. Pencapaian tersebut berupa jumlah pelaporan *sales out* yang dikategorikan berdasarkan status pelaporan (*submitted, received, approved, retur*) dan jumlah target (Rp. dan %) yang dicapai pada bulan tersebut. Untuk lebih jelasnya mengenai alur menampilkan pencapaian CR akan dijelaskan pada diagram aktivitas di bawah ini :

RBTC



Gambar 4.2.2-4 Diagram Aktivitas Menampilkan Pencapaian CR

BAB V

IMPLEMENTASI SISTEM

Bab ini membahas tentang implementasi dari perancangan sistem. Bahasa pemrograman yang digunakan adalah XML, Java, dan PHP.

5.1. Tampilan Antarmuka *User*

5.1.1. View Login

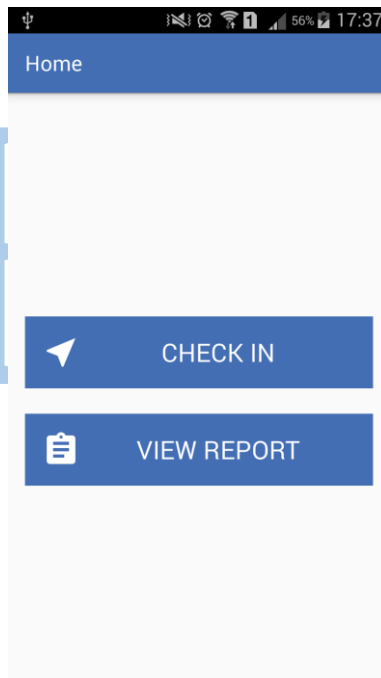
Merupakan antarmuka paling awal saat membuka aplikasi CR Tool. CR harus melakukan *login* dengan memasukkan *username* dan *password* yang benar agar dapat mengakses fitur-fitur dari CR Tool.



Gambar 5.1.1-1 View Login

5.1.2. View Home

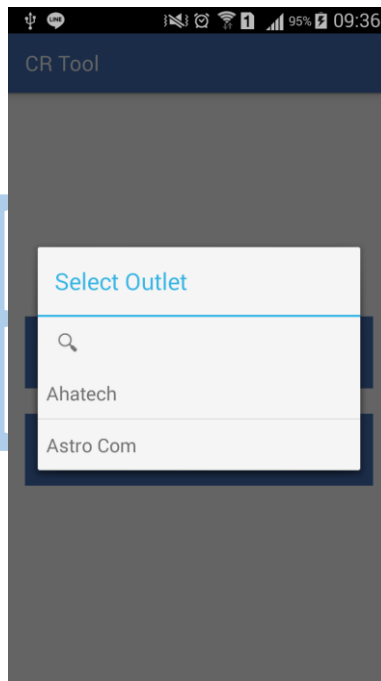
Merupakan antarmuka utama yang ditampilkan setelah CR berhasil melakukan *login*. Pada *view* Home terdapat dua buah tombol yang dapat dipilih, yaitu tombol “CHECK IN” dan tombol “VIEW REPORT”. Tombol “CHECK IN” dipilih jika CR ingin melakukan *check in* ke outlet yang ingin dilaporkan *sales out* nya. Tombol “VIEW REPORT” dipilih jika CR ingin melihat laporan *sales out* dan pencapaiannya.



Gambar 5.1.2-1 View Home

5.1.3. View Dialog Check In

Merupakan antarmuka yang ditampilkan setelah CR memilih tombol “CHECK IN” pada *view* Home dan terdapat lebih dari satu outlet yang berada pada radius 500 meter dari posisi CR berada. CR dapat memilih outlet yang ingin dilakukan *check in* dari daftar outlet yang ditampilkan.



Gambar 5.1.3-1 View Dialog Check In

5.1.4. View Input SN

Merupakan antarmuka yang ditampilkan setelah CR berhasil melakukan *check in*. CR dapat memasukkan SN dari barang yang terjual secara manual pada kolom SN atau dengan memilih tombol “SCAN” untuk memindai *barcode* SN dari barang yang terjual. CR dapat melakukan pelaporan *sales out* dengan memilih tombol “SUBMIT SN” setelah kolom SN terisi.

Serial Number

SUBMIT SN

Ketentuan

- Setelah submit SN, silahkan mengirimkan kartu garansi ke kantor TDI
Ibu Frita/Romi
JL. Ngagel Jaya Selatan blok H7 Kompleks RMI
(ex. Kebun Bibit -Bratang)
Surabaya 60283
- SN / IMEI akan di-approve setelah kartu garansi diterima
- Semua pertanyaan yang berkaitan dengan klaim harap ditujukan ke admintrp@terra.co.id

Gambar 5.1.4-1 View Input SN

PT JNE (e-connote)

05/09/2016 12:36:31

Layanan : YES 15

Jawa Kriman : DOC

No. e-Connote : JD04801673104016

Stampel : TIDAK DIASURANSIKAN

Keterangan	Jml	Berat Asli	Dimensi L	H	W	Berat Volume
DOC	1	1.00				0.1

Daya Kirim : 1 IDR

Berat : 1.00

25,000.00

TE (e-connote) Halaman 1 of 1

03-09-2016 12:36:31

LEMBAR UNTUK PENGIRIM

Baris: 1 of 1

Yes/No: YES/No

Barcode: B2DH081673154616

Stamp: TIDAK DIASURANSIKAN

Keterangan	Jml	Baris	Ases	L	Dirangs	W	Baris	Kuant
DOC	1	1.00						5.1

Baris: 1 of 1

Biaya Kirim : IDR 25,000.00

Biaya Lain-Lain : IDR 0.00

Admin Asuransi : IDR 0.00

Total Biaya : IDR 25,000.00

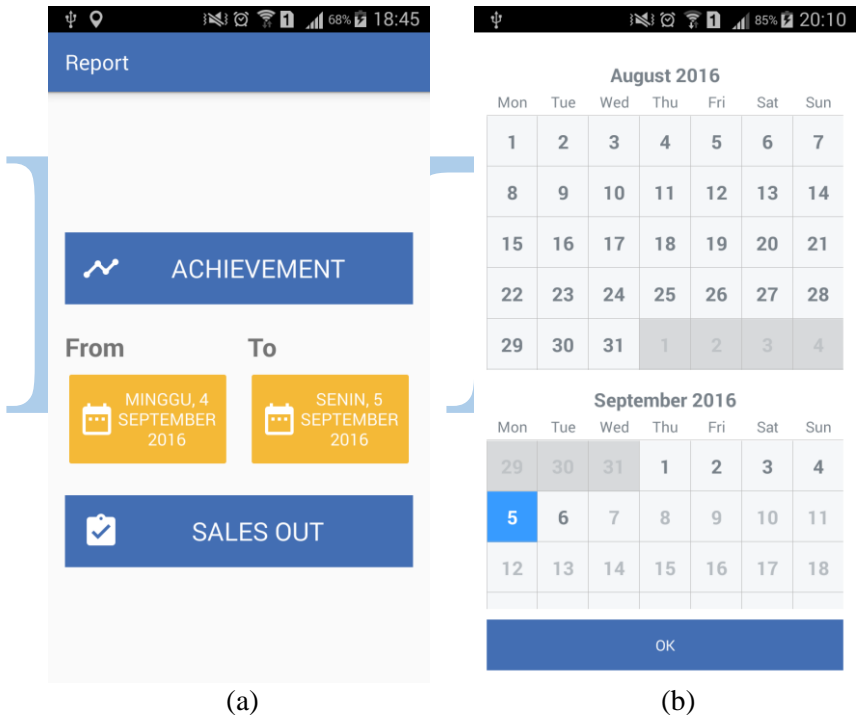
Petugas Tm Pengirim

(TINI KIRI)

03-09-2016 12:37

5.1.6. View Report

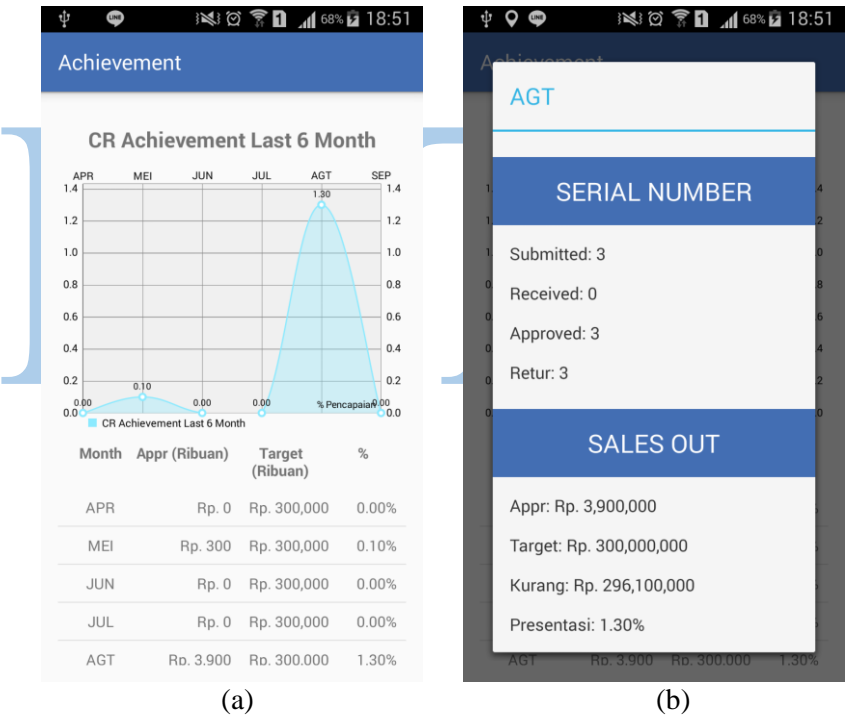
Merupakan antarmuka yang ditampilkan setelah CR memilih tombol “VIEW REPORT” pada *view* Home. Pada *view* Report terdapat beberapa tombol yang dapat dipilih, yaitu tombol “ACHIEVEMENT” untuk melihat pencapaian CR, tombol “SALES OUT” untuk melihat laporan *sales out* dan dua tombol untuk memilih rentang tanggal dari laporan *sales out* yang ingin dilihat.



Gambar 5.1.6-1 View Report (a), View Memilih Rentang Tanggal (b)

5.1.7. View Achievement

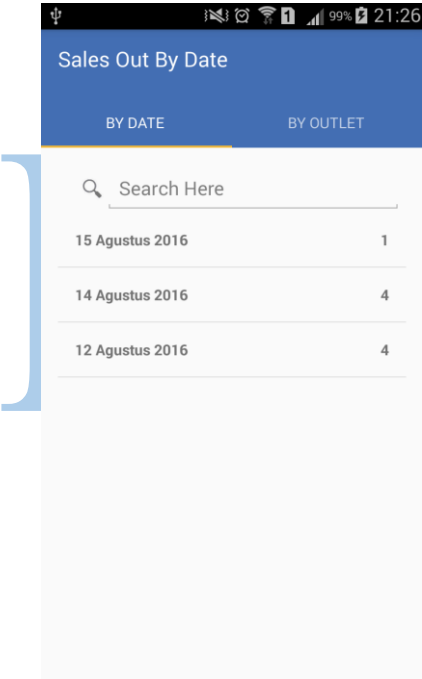
Merupakan antarmuka yang ditampilkan setelah CR memilih tombol “ACHIEVEMENT” pada *view* Report. Pada *view* Achievement ditampilkan pencapaian CR selama 6 bulan terakhir dalam rupa grafik dan tabel. Detil pencapaian tersebut berupa jumlah pelaporan *sales out* yang dikategorikan berdasarkan status pelaporan (*submitted*, *received*, *approved*, *retur*) dan jumlah target (Rp. dan %) yang dicapai pada bulan tersebut.



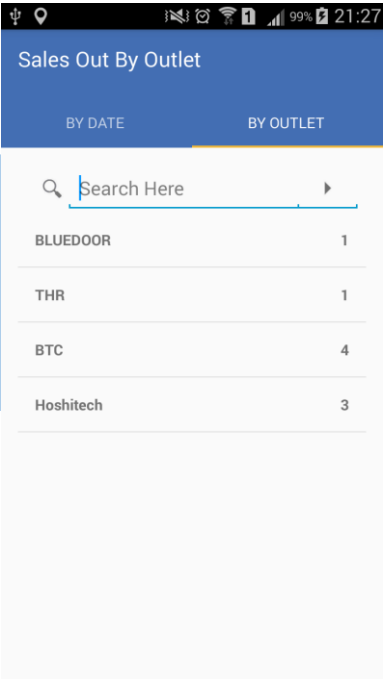
Gambar 5.1.7-1 View Achievement (a), View Detil Achievement (b)

5.1.8. View Sales Out

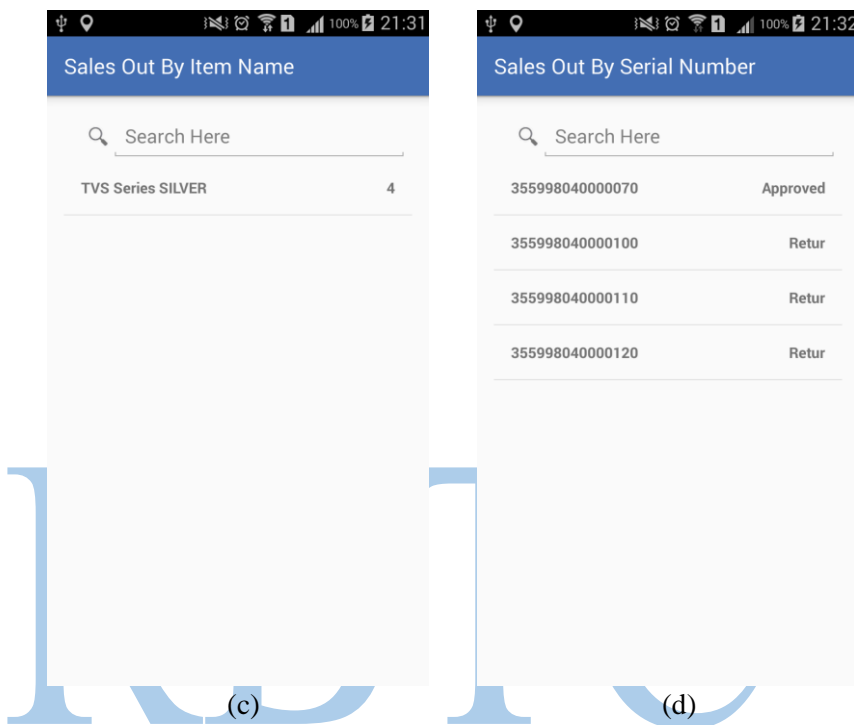
Merupakan antarmuka yang ditampilkan setelah CR memilih tombol “SALES OUT” pada *view* Report. Pada *view* Sales Out ditampilkan laporan *sales out* (SN dari produk yang di *sales out* dan status pelaporan) yang telah dilaporkan CR selama rentang periode yang dipilih pada *view* Report. Laporan yang ditampilkan dapat dapat di-*filter* berdasarkan tanggal atau outlet.



(a)



(b)



Gambar 5.1.8-1 View Sales Out (a)(b)(c)(d)

5.2. Implementasi Antarmuka dan Logic

5.2.1. Implementasi Sistem *Login*

Berikut merupakan potongan dari kode antarmuka dan *logic* yang digunakan untuk implementasi sistem *login*.

1. **activity_login.xml** digunakan untuk membangun tampilan antarmuka dari *view login*.

```
<?xml version="1.0" encoding="utf-8"?>
<LinearLayout
xmlns:android="http://schemas.android.com/apk/res/android"
xmlns:tools="http://schemas.android.com/tools"
android:layout_width="match_parent"
android:layout_height="match_parent"
android:background="@android:color/white"
android:orientation="vertical"
android:paddingBottom="@dimen/activity_vertical_margin"
android:paddingLeft="@dimen/activity_horizontal_margin"
android:paddingRight="@dimen/activity_horizontal_margin"
android:paddingTop="@dimen/activity_vertical_margin"
tools:context=". Activity.LoginActivity">

    <ImageView
        android:layout_width="wrap_content"
        android:layout_height="wrap_content"
        android:contentDescription="@string/deskripsi_logo"
        android:src="@drawable/logo_kreator" />

    <EditText
        style="@style/ShortInput"
        android:id="@+id/usernameLogin"
        android:hint="@string/username"
        android:inputType="text" />

    <EditText
        style="@style/ShortInput"
        android:id="@+id/passwordLogin"
        android:hint="@string/password"
```

```

        android:inputType="textPassword" />

<Button
    android:id="@+id/buttonLogin"
    style="@style/Widget.AppCompat.Button.Colored"
    android:layout_marginTop="@dimen/activity_margin_top"
    android:onClick="checkLogin"
    android:textSize="@dimen/font_small_0"
    android:text="@string/login_CRTool"
    android:layout_width="match_parent"
    android:layout_height="wrap_content" />

</LinearLayout>

```

Kode Sumber 5.2.1-1 activity_login.xml

2. **LoginActivity.java** berisi *logic* dari *view* login yang berfungsi untuk melakukan *login* ke *server*. Fitur ini juga dilengkapi dengan enkripsi md5.

```

package com.kreators.crtoolv1.Activity;

public class LoginActivity extends AppCompatActivity {
    private SessionManager session;
    private VolleyManager volleyManager;
    private ProgressDialog pd;
    private EditText editTextUsername, editTextPassword;
    private String username, password;
    private Button buttonLogin;

    @Override
    protected void onCreate(Bundle savedInstanceState) {
        super.onCreate(savedInstanceState);
        setContentView(R.layout.activity_login);
        initialization();
        session.checkLoginSession(LoginActivity.this);
    }
}

```

```

private void checkLogin() {
    pd.setTitle(Constant. loginDialog);
    pd.show();
    GetVolleyRequest request = new GetVolleyRequest(Uri. LOG_IN);
    request.putParams(Protocol. USERNAME, username);
    request.putParams(Protocol. PASSWORD, password);
    request.setListener(new VolleyListener() {
        @Override
        public void onSuccess(VolleyRequest request, JSONArray
result) {
            try {
                JSONObject response;
                int crID=0;
                boolean status=false;
                String message="";
                for(int i = 0; i < result.length(); i++) {
                    response = result.getJSONObject(i);
                    message = response.getString("message");
                    status = response.getBoolean("status");
                    if(response.has("ID")) crID =
response.getInt("ID");
                }
                if (pd != null) {
                    pd.dismiss();
                }
                Toast. makeText(LoginActivity. this, message,
Toast. LENGTH_SHORT).show();

                if(status) {
                    session.createLoginSession(crID, username);
                    Intent intent = new
Intent(LoginActivity. this, HomeActivity. class);
                    startActivity(intent);
                    finish();
                }

            } catch (JSONException e) {
                e.printStackTrace();
            }
        }
    });
}

```



```

    }
}

@Override
public void onError(VolleyRequest request, String
errorMessage) {
    Toast.makeText(LoginActivity.this, errorMessage,
Toast.LENGTH_SHORT).show();
    if (pd != null) {
        pd.dismiss();
    }
}
});
volleyManager.createRequest(request, Protocol.POST);
}

public void initialization() {
    setTitle("Login");
    editTextUsername = (EditText)
findViewById(R.id.usernameLogin);
    editTextPassword = (EditText)
findViewById(R.id.passwordLogin);
    volleyManager =
VolleyManager.getInstance(getApplicationContext());
    session = new SessionManager(getApplicationContext());
    pd = new ProgressDialog(this);
    pd.setMessage("Please wait.");
    pd.setCancelable(false);
    pd.setIndeterminate(true);
    buttonLogin = (Button) findViewById(R.id.buttonLogin);

    buttonLogin.setOnClickListener(new View.OnClickListener() {
        @Override
        public void onClick(View v) {
            username = editTextUsername.getText().toString();
            password =
md5(editTextPassword.getText().toString());
            Log.d("a", "a");

```

```

        if (!username.equals("") && !password.equals("")) {
            checkLogin();
        } else {
            Toast.makeText(LoginActivity.this, "Fill
Username or Password", Toast.LENGTH_LONG).show();
        }
    }
});
}

public static String md5(String string) {
    byte[] hash;

    try {
        hash =
MessageDigest.getInstance("MD5").digest(string.getBytes("UTF-8"));
    } catch (NoSuchAlgorithmException e) {
        throw new RuntimeException("Huh, MD5 should be
supported?", e);
    } catch (UnsupportedEncodingException e) {
        throw new RuntimeException("Huh, UTF-8 should be
supported?", e);
    }

    StringBuilder hex = new StringBuilder(hash.length * 2);

    for (byte b : hash) {
        int i = (b & 0xFF);
        if (i < 0x10) hex.append('0');
        hex.append(Integer.toHexString(i));
    }

    return hex.toString();
}
}

```

Kode Sumber 5.2.1-2 LoginActivity.java

3. SessionManager.java berisi *logic* untuk menyimpan *session* dari login.

```
package com.kreators.crtoolv1.Commons;

public class SessionManager {
    SharedPreferences pref;
    Editor editor;
    Context _context;

    public SessionManager(Context context) {
        this._context = context;
        pref = _context.getSharedPreferences(Protocol. PREF_NAME,
Protocol. PRIVATE_MODE);
        editor = pref.edit();
    }

    public void createLoginSession(int crID, String name) {
        editor.putBoolean(Protocol. IS_LOGIN, true);
        editor.putString(Protocol. USERID, String. valueOf(crID));
        editor.putString(Protocol. USERNAME, name);
        editor.commit();
    }

    public void checkLoginSession(LoginActivity loginActivity) {
        if(this.isLoggedIn()) {
            Intent i = new Intent(_context, HomeActivity.class);
            i.addFlags(Intent.FLAG_ACTIVITY_CLEAR_TOP);
            i.setFlags(Intent.FLAG_ACTIVITY_NEW_TASK);
            _context.startActivity(i);
            loginActivity.finish();
        }
    }

    public boolean isLoggedIn() {
        return pref.getBoolean(Protocol. IS_LOGIN, false);
    }

    public void logoutUser() {
```

```

        editor.clear();
        editor.commit();
        Intent i = new Intent(_context, LoginActivity.class);
        i.addFlags(Intent.FLAG_ACTIVITY_CLEAR_TOP);
        i.setFlags(Intent.FLAG_ACTIVITY_NEW_TASK);
        _context.startActivity(i);
    }

    public HashMap<String, String> getUserDetails() {
        HashMap<String, String> user = new HashMap<String,
String>();
        user.put(Protocol.USERID, pref.getString(Protocol.USERID,
null));
        user.put(Protocol.USERNAME,
pref.getString(Protocol.USERNAME, null));
        return user;
    }
}

```

Kode Sumber 5.2.1-3 SessionManager.java

5.2.2. Implementasi Sistem *Check In*

Berikut merupakan potongan dari kode antarmuka dan *logic* yang digunakan untuk implementasi sistem *check in*.

1. **activity_home.xml** digunakan untuk membangun tampilan antarmuka dari *view* Home.

```

<?xml version="1.0" encoding="utf-8"?>
<LinearLayout xmlns:android="http://schemas.android.com/apk/res/android"
    xmlns:tools="http://schemas.android.com/tools"
    android:layout_width="match_parent"
    android:layout_height="match_parent"
    android:gravity="center"
    android:paddingBottom="@dimen/activity_vertical_margin"

```

```

        android:paddingLeft="@dimen/activity_horizontal_margin"
        android:paddingRight="@dimen/activity_horizontal_margin"
        android:paddingTop="@dimen/activity_vertical_margin"
        android:orientation="vertical"
        tools:context=".Activity.HomeActivity">

```

```
<Button
```

```

    style="@style/DarkButton"
    android:drawableLeft="@drawable/ic_near_me_white_36dp"
    android:onClick="checkIn"
    android:text="@string/checkin_CRTool" />

```

```
<Button
```

```

    style="@style/DarkButton"
    android:drawableLeft="@drawable/ic_assignment_white_36dp"
    android:onClick="viewReport"
    android:text="@string/report_button" />

```

```
</LinearLayout>
```

Kode Sumber 5.2.2-1 activity_home.xml

2. **HomeActivity.java** berisi *logic* dari *view* Home. Berfungsi sebagai kelas yang memanggil kelas **GoogleLocationRequest.java**, menerima lokasi CR dari **GoogleLocationRequest.java**, menggunakan lokasi CR untuk mencari outlet terdekat, memanggil kelas **SelectOutletDialogFragment.java**, menerima outlet yang ingin dilakukan *check in* dari **SelectOutletDialogFragment.java**, memanggil kelas **SalesOutActivity.java** dan kelas **ReportActivity.java**.

```

package com.kreators.crtoolv1.Activity;

public class HomeActivity extends AppCompatActivity implements
SelectOutletDialogFragment.MyDialogFragmentListener {
    public static final String TAG = HomeActivity.class.getSimpleName();
    private static final int PERMISSION_LOCATION_REQUEST_CODE = 1;
    private static final int REQUEST_CHECK_SETTINGS = 0x1;
    private VolleyManager volleyManager;

```

```

private GoogleLocationRequest googleLocationRequest;
private double curLat;
private double curLng;
private ProgressDialog pd;
private SimpleDateFormat sdf;
private SessionManager sessionManager;
private HashMap<String, String> user = new HashMap<>();

@Override
protected void onCreate(Bundle savedInstanceState) {
    super.onCreate(savedInstanceState);
    setContentView(R.layout.activity_home);
    setTitle("Home");
    setUpGoogleLocationRequest();
    initialization();
}

@Override
protected void onStop() {
    super.onStop();
    if (googleLocationRequest.getConnectionStatus()) {
        googleLocationRequest.removeLocationUpdates();
        googleLocationRequest.setGoogleAPIConnection(false);
    }
    volleyManager.cancelPendingRequests(Protocol.GET);
}

@Override
public void onRequestPermissionsResult(int requestCode, String[]
permissions, int[] grantResults) {
    switch (requestCode) {
        case PERMISSION_LOCATION_REQUEST_CODE:
            if (grantResults.length > 0 && grantResults[0] ==
PackageManager.PERMISSION_GRANTED) {
            } else {
                Toast.makeText(this, Constant.needLocation,
Toast.LENGTH_SHORT).show();
            }
    }
}

```

```

        break;
    }
}

@Override
protected void onActivityResult(int requestCode, int resultCode,
Intent data) {
    switch (requestCode) {
        case REQUEST_CHECK_SETTINGS:
            switch (resultCode) {
                case Activity.RESULT_OK:
                    googleLocationRequest.startLocationUpdates();
                    break;
                case Activity.RESULT_CANCELED:
                    break;
            }
            break;
    }
}

@Override
public void onReturnValue(final Outlet outlet) {
    pd.setTitle(Constant.checkinDialog);
    pd.show();
    Date dt = new Date();
    String currentTime = sdf.format(dt);
    GetVolleyRequest request = new
GetVolleyRequest(Uri.POST_CHECK_IN_OUTLET);
    request.putParams(Protocol.CRID, user.get(Protocol.USERID));
    request.putParams(Protocol.OUTLETID,
String.valueOf(outlet.getOutletID()));
    request.putParams(Protocol.CUR_DATE_TIME, currentTime);
    request.setListener(new VolleyListener() {
        @Override
        public void onSuccess(VolleyRequest request, JSONArray
result) {
            try {
                JSONObject jsonObject = result.getJSONObject(0);

```

```

        if (pd != null) {
            pd.dismiss();
        }

        if (jsonObject.getBoolean("status")) {
            Intent intent = new Intent(HomeActivity.this,
SalesOutActivity.class);
            intent.putExtra(Protocol.OUTLETID,
String.valueOf(outlet.getOutletID()));
            intent.putExtra(Protocol.SN_OUTLET_NAME,
outlet.getOutletName());
            startActivity(intent);
        } else {
            Toast.makeText(HomeActivity.this,
jsonObject.getString("message"), Toast.LENGTH_SHORT).show();
        }
    } catch (JSONException e) {
        e.printStackTrace();
    }
}

@Override
public void onError(VolleyRequest request, String
errorMessage) {
    Toast.makeText(HomeActivity.this, errorMessage,
Toast.LENGTH_SHORT).show();
    if (pd != null) {
        pd.dismiss();
    }
}

});
volleyManager.createRequest(request, Protocol.POST);
}

private void storeCurrentLocation(Location location) {
    curLat = location.getLatitude();
    curLng = location.getLongitude();
}

```



```

    }

    private void searchNearestOutlet() {
        pd.setTitle(Constant. searchDialog);
        pd.show();
        GetVolleyRequest request = new
GetVolleyRequest(Uri. GET_CHECK_IN_OUTLET);
        request.putParams(Protocol. LONGITUDE, String. valueOf(curLng));
        request.putParams(Protocol. LATITUDE, String. valueOf(curLat));
        request.setListener(new VolleyListener() {
            @Override
            public void onSuccess(VolleyRequest request, JSONArray
result) {
                try {
                    List<Outlet> nearestOutlet= new ArrayList<>();
                    JSONObject outletObj;

                    for(int i = 0; i < result.length(); i++) {
                        outletObj = result.getJSONObject(i);
                        if(outletObj.has("status")) {
                            break;
                        }

                        Outlet outlet = new Outlet();
                        outlet.setOutletID(outletObj.getInt("ID"));

outlet.setOutletName(outletObj.getString("Name"));
                        nearestOutlet.add(outlet);
                    }

                    if (pd != null) {
                        pd.dismiss();
                    }

                    if(nearestOutlet.size() > 0) {
                        if(nearestOutlet.size() > 1){
                            showDialog(nearestOutlet);
                        }
                    }
                }
            }
        });
    }
}

```

```

        else {
            onReturnValue(nearestOutlet.get(0));
        }
    }
    else {
        Toast.makeText(HomeActivity.this,
Constant.noOutlets, Toast.LENGTH_LONG).show();
    }

    } catch (JSONException e) {
        e.printStackTrace();
    }
}

@Override
public void onError(VolleyRequest request, String
errorMessage) {
    Toast.makeText(HomeActivity.this, errorMessage,
Toast.LENGTH_SHORT).show();
    if (pd != null) {
        pd.dismiss();
    }
}

));
volleyManager.createRequest(request, Protocol.GET);
}

private void showDialog(List<Outlet> nearestOutlet) {
    FragmentTransaction ft =
getFragmentManager().beginTransaction();
    Fragment prev = getFragmentManager().findFragmentByTag("Select
Outlet");
    if (prev != null) {
        ft.remove(prev);
    }
    SelectOutletDialogFragment s0 =
SelectOutletDialogFragment.newInstance(nearestOutlet);
    s0.show(ft, "Select Outlet");
}

```

```

    }

    private void initialization() {
        volleyManager =
VolleyManager.getInstance(getApplicationContext());

        pd = new ProgressDialog(this);
        pd.setMessage(Constant.msgDialog);
        pd.setCancelable(false);
        pd.setIndeterminate(true);

        sdf = new SimpleDateFormat(Constant.SYSTEM_DATE_COMPLETE);

        sessionManager = new SessionManager(getApplicationContext());
        user = sessionManager.getUserDetails();
    }

    private void setUpGoogleLocationRequest() {

        if ( ContextCompat.checkSelfPermission( this,
android.Manifest.permission.ACCESS_COARSE_LOCATION ) !=
PackageManager.PERMISSION_GRANTED &&
            ContextCompat.checkSelfPermission(this,
android.Manifest.permission.ACCESS_FINE_LOCATION) !=
PackageManager.PERMISSION_GRANTED) {

            ActivityCompat.requestPermissions(this, new
String[] {android.Manifest.permission.ACCESS_COARSE_LOCATION,
android.Manifest.permission.ACCESS_FINE_LOCATION},
PERMISSION_LOCATION_REQUEST_CODE);

        }

        googleLocationRequest = new GoogleLocationRequest(this);
        googleLocationRequest.setListener(new GoogleLocationListener() {
            @Override
            public void onConnected(GoogleLocationRequest request) {
                if (pd != null) {
                    pd.dismiss();
                }
            }
        });
    }

```

```

    }
}

@Override
public void onConnectionSuspended(GoogleLocationRequest
request, String errorMessage) {
    if (pd != null) {
        pd.dismiss();
    }
    Toast.makeText(HomeActivity.this, errorMessage,
Toast.LENGTH_LONG).show();
}

@Override
public void onConnectionFailed(GoogleLocationRequest request,
String errorMessage) {
    if (pd != null) {
        pd.dismiss();
    }
    Toast.makeText(HomeActivity.this, errorMessage,
Toast.LENGTH_LONG).show();
}

@Override
public void onLocationChanged(GoogleLocationRequest request,
Location location) {
    storeCurrentLocation(location);
    searchNearestOutlet();
}
});
}

public void checkIn(View view) {
    if (googleLocationRequest.getConnectionStatus()) {
        googleLocationRequest.checkLocationSettings();
    } else {
        pd.setTitle(Constant.connectDialog);
        pd.show();
    }
}

```

```

        googleLocationRequest.setGoogleAPIConnection(true);
    }
}

public void viewReport(View view) {
    Intent intent = new Intent(this, ReportActivity.class);
    startActivity(intent);
}

@Override
public void onBackPressed() {
    AlertDialog.Builder builder = new AlertDialog.Builder(this,
R.style.MyDialogTheme);
    builder.setCancelable(false);
    builder.setMessage(Constant.exitApp);
    builder.setPositiveButton(Constant.YES,
        new DialogInterface.OnClickListener() {
            @Override
            public void onClick(DialogInterface dialog, int id)
        {
            dialog.dismiss();
            HomeActivity.this.finish();
        }
    ));
    builder.setNegativeButton(Constant.NO,
        new DialogInterface.OnClickListener() {
            public void onClick(DialogInterface dialog, int id)
        {
            dialog.cancel();
        }
    ));
    builder.create().show();
}
}

```

Kode Sumber 5.2.2-2 HomeActivity.java

3. **GoogleLocationRequest.java** adalah kelas yang mengimplementasikan Location APIs dari *library* Google Play Services. Berfungsi untuk mendapatkan dan memperbaharui lokasi dari CR secara *real time* menggunakan teknologi Global Positioning System (GPS), mengirimkan lokasi yang didapat melalui fungsi *listener* yang diimplementasikan di **HomeActivity.java**.

```
package com.kreators.crtoolv1.Network;

public class GoogleLocationRequest implements
    GoogleApiClient.ConnectionCallbacks,
    GoogleApiClient.OnConnectionFailedListener,
    LocationListener {

    private static final int REQUEST_CHECK_SETTINGS = 0x1;

    private Context mContext;
    private GoogleLocationListener mListener;
    private GoogleApiClient googleApiClient;
    private LocationRequest mLocationRequest;

    private ProgressDialog locRequestProgress;
    private Handler pdCanceller;
    private Runnable progressRunnable;

    public GoogleLocationRequest(Context context) {

        mContext = context;

        googleApiClient = new GoogleApiClient.Builder(context).
            addConnectionCallbacks(this).
            addOnConnectionFailedListener(this).
            addApi(LocationServices.API).
            build();

        mLocationRequest = LocationRequest.create()
            .setPriority(LocationRequest.PRIORITY_HIGH_ACCURACY)
```

```

        .setSmallestDisplacement(10) // 10 meters
        .setInterval(60 * 1000)      // 60 seconds, in
milliseconds
        .setFastestInterval(10 * 1000); // 10 second, in
milliseconds

        locRequestProgress = new ProgressDialog(context);
        locRequestProgress.setTitle(Constant. searchDialog);
        locRequestProgress.setMessage(Constant. msgDialog);
        locRequestProgress.setCancelable(true);
        locRequestProgress.setOnCancelListener(new
DialogInterface.OnCancelListener() {
            @Override
            public void onCancel(DialogInterface dialog) {
                Toast.makeText(mContext, Constant. noLocation,
Toast. LENGTH_SHORT).show();
                removeLocationUpdates();
            }
        });

        pdCanceller = new Handler();
        progressRunnable = new Runnable() {
            @Override
            public void run() {
                locRequestProgress.cancel();
            }
        };
    }

    @Override
    public void onConnected(Bundle bundle) {
        if (mListener != null) {
            mListener.onConnected(this);
            checkLocationSettings();
        }
    }

    @Override

```

```

public void onConnectionSuspended(int errorCode) {
    if (mListener != null) {
        String errorMessage = Constant.locSuspend;
        mListener.onConnectionSuspended(this, errorMessage);
    }
}

@Override
public void onConnectionFailed(ConnectionResult connectionResult) {
    if (mListener != null) {
        int errorCode = connectionResult.getErrorCode();
        String errorMessage = Constant.locFailed;
        if (errorCode == 2) {
            errorMessage = Constant.updateServices;
        }
        mListener.onConnectionFailed(this, errorMessage);
    }
}

@Override
public void onLocationChanged(Location location) {
    if (locRequestProgress != null) {
        locRequestProgress.dismiss();
        pdCanceller.removeCallbacks(progressRunnable);
    }
    if (mListener != null) {
        mListener.onLocationChanged(this, location);
    }
}

public void setListener(GoogleLocationListener listener) { mListener
= listener; }

public void setGoogleAPIConnection(boolean connect) {
    if (connect) {
        googleApiClient.connect();
    } else {
        googleApiClient.disconnect();
    }
}

```



```

    }
}

public boolean getConnectionStatus() {
    return googleApiClient.isConnected();
}

public void checkLocationSettings() {
    LocationSettingsRequest.Builder builder = new
LocationSettingsRequest.Builder()
        .addLocationRequest(mLocationRequest);
    builder.setAlwaysShow(true);
    PendingResult<LocationSettingsResult> result =
LocationServices.SettingsApi.checkLocationSettings(googleApiClient,
        builder.build());
    result.setResultCallback(new
ResultCallback<LocationSettingsResult>() {
        @Override
        public void onResult(LocationSettingsResult result) {
            final Status status = result.getStatus();
            final LocationSettingsStates state =
result.getLocationSettingsStates();
            switch (status.getStatusCode()) {
                case LocationSettingsStatusCodes.SUCCESS:
                    startLocationUpdates();
                    break;
                case
LocationSettingsStatusCodes.RESOLUTION_REQUIRED:
                    try {
                        status.startResolutionForResult((Activity)
mContext, REQUEST_CHECK_SETTINGS);
                    } catch (IntentSender.SendIntentException e) {
                    }
                    break;
                case
LocationSettingsStatusCodes.SETTINGS_CHANGE_UNAVAILABLE:
                    break;
            }
        }
    });
}

```

```

    }
    });
}

    public void startLocationUpdates() {
        if (ContextCompat.checkSelfPermission(mContext,
            android.Manifest.permission.ACCESS_COARSE_LOCATION)
            == PackageManager.PERMISSION_GRANTED) {

            //lastLocation equivalent to user's current location
            Location lastLocation =
                LocationServices.FusedLocationApi.getLastLocation(googleApiClient);

            if(lastLocation == null){
                locRequestProgress.show();
                pdCancellable.postDelayed(progressRunnable, 10000);

                LocationServices.FusedLocationApi.requestLocationUpdates(googleApiClient,
                    mLocationRequest, this);
            } else {
                if (mListener != null) {
                    mListener.onLocationChanged(this, lastLocation);
                }
            }
        }
    }

    public void removeLocationUpdates() {

        LocationServices.FusedLocationApi.removeLocationUpdates(googleApiClient,
            this);
    }
}

```

Kode Sumber 5.2.2-3 GoogleLocationRequest.java

4. **fragment_select_outlet_dialog.xml** digunakan untuk membangun tampilan antarmuka dari *view* dialog Check In.

```
<?xml version="1.0" encoding="utf-8"?>

<LinearLayout
xmlns:android="http://schemas.android.com/apk/res/android"
    android:layout_width="match_parent"
    android:layout_height="match_parent"
    android:orientation="vertical">

    <SearchView
        android:id="@+id/svSearchOutlet"
        android:layout_width="match_parent"
        android:layout_height="wrap_content" />

    <ListView
        android:id="@+id/lvListViewOutlet"
        android:layout_width="match_parent"
        android:layout_height="match_parent"/>

</LinearLayout>
```

Kode Sumber 5.2.2-4 fragment_select_outlet_dialog.xml

5. **SelectOutletDialogFragment.java** berisi *logic* dari *view* dialog Check In. Berfungsi sebagai kelas yang menerima dan menampilkan daftar outlet yang dapat dilakukan *check in*, mengirimkan outlet yang dipilih melalui fungsi *listener* yang diimplementasikan di **HomeActivity.java**.

```
package com.kreators.crtoolv1.Fragment.Dialog;

public class SelectOutletDialogFragment extends DialogFragment
implements AdapterView.OnItemClickListener {

    private OutletAdapter outletAdapter;
    private List<Outlet> mOutlets;

    public static SelectOutletDialogFragment newInstance(List<Outlet>
```

```

nearestOutlet() {
    SelectOutletDialogFragment f = new SelectOutletDialogFragment();

    Bundle args = new Bundle();
    args.putParcelableArrayList(Protocol.NEAREST, (ArrayList<?
extends Parcelable>) nearestOutlet);
    f.setArguments(args);

    return f;
}

@Override
public void onCreate(Bundle savedInstanceState) {
    super.onCreate(savedInstanceState);
    mOutlets =
getArguments().getParcelableArrayList(Protocol.NEAREST);
}

@Override
public View onCreateView(LayoutInflater inflater, ViewGroup
container, Bundle savedInstanceState) {
    View rootView =
inflater.inflate(R.layout.fragment_select_outlet_dialog, container,
false);

    getDialog().setTitle("Select Outlet");

    ListView lv = (ListView)
rootView.findViewById(R.id.lvListViewOutlet);
    SearchView sv = (SearchView)
rootView.findViewById(R.id.svSearchOutlet);

    outletAdapter = new OutletAdapter(getActivity(), mOutlets);
    lv.setAdapter(outletAdapter);
    lv.setOnItemClickListener(this);

    sv.setQueryHint("Search.");
    sv.setOnQueryTextListener(new SearchView.OnQueryTextListener() {

```

```

        @Override
        public boolean onQueryTextSubmit(String txt) {
            return false;
        }

        @Override
        public boolean onQueryTextChange(String txt) {
            outletAdapter.getFilter().filter(txt);
            return false;
        }
    });

    return rootView;
}

@Override
public void onItemClick(AdapterView<?> adapter, View arg1, int
position, long arg3) {
    Outlet item = (Outlet) adapter.getItemAtPosition(position);
    MyDialogFragmentListener activity = (MyDialogFragmentListener)
getActivity();
    activity.onReturnValue(item);
    dismiss();
}

public interface MyDialogFragmentListener {
    void onReturnValue(Outlet outlet);
}
}

```

Kode Sumber 5.2.2-5 SelectOutletDialogFragment.java

5.2.3. Implementasi Sistem Pelaporan *Sales Out*

Berikut merupakan potongan dari kode antarmuka dan *logic* yang digunakan untuk implementasi sistem pelaporan *sales out*.

1. **activity_sales_out.xml** digunakan sebagai wadah untuk membangun tampilan antarmuka dari *view* Input SN dan *view* Scan SN.

```
<?xml version="1.0" encoding="utf-8"?>
<FrameLayout xmlns:android="http://schemas.android.com/apk/res/android"
    xmlns:tools="http://schemas.android.com/tools"
    android:id="@+id/sales_out_activity"
    android:layout_width="match_parent"
    android:layout_height="match_parent"
    tools:context=". Activity. SalesOutActivity">

</FrameLayout>
```

Kode Sumber 5.2.3-1 activity_sales_out.xml

2. **fragment_sales_out_input.xml** digunakan untuk membangun tampilan antarmuka dari *view* Input SN.

```
<LinearLayout xmlns:android="http://schemas.android.com/apk/res/android"
    xmlns:tools="http://schemas.android.com/tools"
    android:layout_width="match_parent"
    android:layout_height="match_parent"
    android:orientation="vertical"
    android:paddingBottom="@dimen/activity_vertical_margin"
    android:paddingLeft="@dimen/activity_horizontal_margin"
    android:paddingRight="@dimen/activity_horizontal_margin"
    android:paddingTop="@dimen/activity_vertical_margin"
    tools:context=". Fragment. SalesOutInputFragment">

    <LinearLayout
        android:layout_width="match_parent"
        android:layout_height="0dp"
        android:layout_weight="1"
        android:gravity="right"
        android:orientation="horizontal">

        <Button
            android:id="@+id/scanSN"
            style="@style/Widget.AppCompat.Button.Colored"
            android:layout_width="wrap_content"
```

```

        android:layout_height="wrap_content"

android:drawableLeft="@drawable/ic_settings_overscan_white_36dp"
        android:text="@string/scan_sn" />

</LinearLayout>

<LinearLayout
    android:layout_width="match_parent"
    android:layout_height="0dp"
    android:layout_weight="4"
    android:gravity="center"
    android:orientation="vertical">

    <EditText
        android:id="@+id/textSN"
        style="@style/ShortInput"
        android:hint="@string/SN" />

    <Button
        android:id="@+id/btnInputSN"
        style="@style/Widget.AppCompat.Button.Colored"
        android:layout_width="match_parent"
        android:layout_height="wrap_content"
        android:padding="@dimen/padding_in_a_box"
        android:text="@string/submit_sn"
        android:textSize="@dimen/font_medium_0" />

</LinearLayout>

<ScrollView
    android:layout_width="match_parent"
    android:layout_height="0dp"
    android:layout_weight="4"
    android:isScrollContainer="false">

    <LinearLayout
        android:layout_width="match_parent"

```

```

        android:layout_height="wrap_content"
        android:orientation="vertical">

        <TextView
            android:layout_width="match_parent"
            android:layout_height="wrap_content"
            android:text="Ketentuan"
            android:textSize="16dp" />

        <View style="@style/Divider" />

        <TableLayout
            android:layout_width="match_parent"
            android:layout_height="wrap_content">

            <TableRow
                android:layout_width="wrap_content"
                android:layout_height="wrap_content">

                <TextView
                    android:layout_width="match_parent"
                    android:layout_height="wrap_content"
                    android:text="@string/bullet" />

                <TextView
                    android:layout_width="match_parent"
                    android:layout_height="wrap_content"
                    android:layout_weight="1"
                    android:text="@string/ketentuan1" />
            </TableRow>

            <TableRow
                android:layout_width="wrap_content"
                android:layout_height="wrap_content">

                <TextView
                    android:layout_width="match_parent"
                    android:layout_height="wrap_content"

```



```

        android:text="@string/bullet" />

        <TextView
            android:layout_width="match_parent"
            android:layout_height="wrap_content"
            android:layout_weight="1"
            android:text="@string/ketentuan2" />
    </TableRow>

    <TableRow
        android:layout_width="wrap_content"
        android:layout_height="wrap_content">

        <TextView
            android:layout_width="match_parent"
            android:layout_height="wrap_content"
            android:text="@string/bullet" />

        <TextView
            android:layout_width="match_parent"
            android:layout_height="wrap_content"
            android:layout_weight="1"
            android:text="@string/ketentuan3" />
    </TableRow>
</TableLayout>

</LinearLayout>

</ScrollView>

</LinearLayout>

```

Kode Sumber 5.2.3-2 fragment_sales_out_input.xml

3. **SalesOutActivity.java** berisi *logic* yang fungsi utamanya untuk menghubungkan antar kelas *fragment* yaitu **SalesOutInputFragment.java** dan **SalesOutScanFragment.java**.

```

package com.kreators.crtoolv1.Activity;

public class SalesOutActivity extends AppCompatActivity implements
SalesOutListener {

    private String curOutletID;

    @Override
    protected void onCreate(Bundle savedInstanceState) {
        super.onCreate(savedInstanceState);
        setContentView(R.layout.activity_sales_out);

        getSupportActionBar().setDisplayHomeAsUpEnabled(true);

        if (findViewById(R.id.sales_out_activity) != null) {

            if (savedInstanceState != null) {
                return;
            }

            SalesOutInputFragment salesOutInputFragment = new
SalesOutInputFragment();

            curOutletID = getIntent().getStringExtra(Protocol.OUTLETID);
            String curOutletName =
getIntent().getStringExtra(Protocol.SN_OUTLET_NAME);

            if (curOutletID != null && curOutletName != null) {
                setTitle(curOutletName);

                Bundle b = new Bundle();
                b.putString(Protocol.OUTLETID, curOutletID);
                salesOutInputFragment.setArguments(b);
            }

            getSupportFragmentManager().beginTransaction().add(R.id.sales_out_acti
ty, salesOutInputFragment).commit();

```

```

    }

}

@Override
public boolean onOptionsItemSelected(MenuItem item) {
    switch (item.getItemId()) {
        case android.R.id.home:
            this.finish();
            return true;
        default:
            return super.onOptionsItemSelected(item);
    }
}

public void onScanButtonClick() {
    SalesOutScanFragment salesOutScanFragment = new
SalesOutScanFragment();

    FragmentTransaction transaction =
getSupportFragmentManager().beginTransaction();

    transaction.replace(R.id.sales_out_activity,
salesOutScanFragment);

    transaction.addToBackStack(null);

    transaction.commit();
}

@Override
public void handleResult(Result rawResult) {
    if(rawResult.getContents() == null){
        Toast.makeText(this, Constant.noContents,
Toast.LENGTH_LONG).show();
    } else {
        String scanContent = rawResult.getContents();

```

```

        getSupportFragmentManager().popBackStack();

        SalesOutInputFragment salesOutInputFragment = new
SalesOutInputFragment();

        Bundle b = new Bundle();
        b.putString(Protocol.CONTENT, scanContent);
        b.putString(Protocol.OUTLETID, curOutletID);
        salesOutInputFragment.setArguments(b);

        FragmentTransaction transaction =
getSupportFragmentManager().beginTransaction();
        transaction.replace(R.id.sales_out_activity,
salesOutInputFragment);
        transaction.commit();
    }
}

```

Kode Sumber 5.2.3-3 SalesOutActivity.java

4. **SalesOutListener.java** adalah kelas *interface* yang berfungsi sebagai *listener* untuk *event* yang terjadi di kelas *fragment*. Kelas ini diimplementasikan di **SalesOutActivity.java**.

```

package com.kreators.crtoolv1.Fragment.Listener;

public interface SalesOutListener {
    void onScanButtonClick();
    void handleResult(Result result);
}

```

Kode Sumber 5.2.3-4 SalesOutListener.java

5. **SalesOutInputFragment.java** berisi *logic* dari *view* Input SN. Berfungsi sebagai kelas yang menerima SN

dari **SalesOutActivity.java**, memverifikasi dan mengirimkan SN yang dimasukkan CR ke server.

```
package com.kreators.crtoolvl.Fragment;

public class SalesOutInputFragment extends Fragment {

    private String outletID;
    private TextView textSN;
    private SalesOutListener activityCallback;
    private VolleyManager volleyManager;
    private ProgressDialog pd;
    private SimpleDateFormat sdf;
    private SessionManager sessionManager;
    private HashMap<String, String> user = new HashMap<>();

    @Override
    public void onAttach(Context context) {
        super.onAttach(context);
        if (context instanceof SalesOutListener) {
            activityCallback = (SalesOutListener) context;
        } else {
            throw new RuntimeException(context.toString()
                + " must implement SalesOutListener");
        }
    }

    @Override
    public View onCreateView(LayoutInflater inflater, ViewGroup
container,
                                Bundle savedInstanceState) {
        // Inflate the layout for this fragment
        View view = inflater.inflate(R.layout.fragment_sales_out_input,
container, false);

        textSN = (TextView) view.findViewById(R.id.textSN);

        Bundle bundle = getArguments();
```

```

        if(bundle != null) {
            String contentSN = bundle.getString(Protocol. CONTENT);
            outletID = bundle.getString(Protocol. OUTLETID);
            if(contentSN != null) {
                textSN.setText(contentSN);
            }
        }

        final Button btnSN = (Button)
view.findViewById(R.id. btnInputSN);
        btnSN.setOnClickListener(new View.OnClickListener() {
            public void onClick(View v) {
                String input = textSN.getText().toString();
                if( (input.matches("002[0-9A-Z]+") && input.length() >=
17 && input.length() <= 20) ||
                    (input.matches("3[0-9A-Z]+") && input.length()
== 15) ){
                    inputSNButtonClicked(input);
                } else {
                    Toast.makeText(getActivity(), Constant. wrongInput,
Toast. LENGTH_SHORT).show();
                }
            }
        });

        final Button scanSN =
            (Button) view.findViewById(R.id. scanSN);
        scanSN.setOnClickListener(new View.OnClickListener() {
            public void onClick(View v) {
                scanButtonClicked();
            }
        });

        initialization();

        return view;
    }

```

```

@Override
public void onDestroyView() {
    super.onDestroyView();
    textSN.setText("");
}

private void inputSNButtonClicked(String SN) {
    pd.show();

    Date dt = new Date();
    String currentDate = sdf.format(dt);

    GetVolleyRequest request = new
GetVolleyRequest(Uri.SALES_OUT_SN);
    request.putParams(Protocol.CRID, user.get(Protocol.USERID));
    request.putParams(Protocol.OUTLETID, outletID);
    request.putParams(Protocol.SN, SN);
    request.putParams(Protocol.CUR_DATE, currentDate);
    request.setListener(new VolleyListener() {
        @Override
        public void onSuccess(VolleyRequest request, JSONArray
result) {
            try {
                JSONObject jsonObject = result.getJSONObject(0);
                String message = jsonObject.getString("message");

                if (pd != null) {
                    pd.dismiss();
                }

                Toast.makeText(getActivity(), message,
Toast.LENGTH_SHORT).show();
                textSN.setText("");
            } catch (JSONException e) {
                e.printStackTrace();
            }
        }
    })
}

```

```

        @Override
        public void onError(VolleyRequest request, String
errorMessage) {
            Toast.makeText(getActivity(), errorMessage,
Toast.LENGTH_SHORT).show();
            if (pd != null) {
                pd.dismiss();
            }
        }
    });
    volleyManager.createRequest(request, Protocol.POST);
}

private void scanButtonClicked() {
    activityCallback.onScanButtonClick();
}

private void initialization() {
    volleyManager =
VolleyManager.getInstance(getActivity().getApplicationContext());

    pd = new ProgressDialog(getActivity());
    pd.setMessage(Constant.msgDialog);
    pd.setTitle(Constant.submitDialog);
    pd.setCancelable(false);
    pd.setIndeterminate(true);

    sdf = new SimpleDateFormat(Constant.SYSTEM_DATE_STANDART);

    sessionManager = new SessionManager(getActivity());
    user = sessionManager.getUserDetails();
}
}

```

Kode Sumber 5.2.3-5 SalesOutInputFragment.java

6. **SalesOutScanFragment.java** berisi *logic* dari *view* Scan SN. Berfungsi sebagai kelas yang memasang *default view*

dari *library* Barcode Scanner sebagai *view* Scan SN, memindai *barcode* SN dari produk yang terjual dan mengirimkan hasilnya ke **SalesOutActivity.java** melalui **SalesOutListener.java**.

```
package com.kreators.crtoolv1.Fragment;

public class SalesOutScanFragment extends Fragment implements
ZBarScannerView.ResultHandler {

    private SalesOutListener activityCallback;
    private ZBarScannerView mScannerView;

    @Override
    public void onAttach(Context context) {
        super.onAttach(context);
        if (context instanceof SalesOutListener) {
            activityCallback = (SalesOutListener) context;
        } else {
            throw new RuntimeException(context.toString()
                + " must implement SalesOutListener");
        }
        mScannerView = new ZBarScannerView(context);
        mScannerView.setResultHandler(this);
    }

    @Override
    public View onCreateView(LayoutInflater inflater, ViewGroup
container,
                                Bundle savedInstanceState) {
        return mScannerView; // Set the scanner view as
the content view
    }

    @Override
    public void onResume() {
        super.onResume();
        mScannerView.startCamera(); // Start camera on resume
    }
}
```

```

@Override
public void onPause() {
    super.onPause();
    mScannerView.stopCamera(); // Stop camera on pause
}

@Override
public void handleResult(Result result) {
    activityCallback.handleResult(result);
}
}

```

Kode Sumber 5.2.3-6 SalesOutScanFragment.java

5.2.4. Implementasi Sistem Menampilkan Laporan *Sales Out*

Berikut merupakan potongan dari kode antarmuka dan *logic* yang digunakan untuk implementasi sistem menampilkan laporan *sales out*.

1. **activity_report.xml** digunakan untuk tampilan activity dari *view report*.

```

<?xml version="1.0" encoding="utf-8"?>
<FrameLayout
xmlns:android="http://schemas.android.com/apk/res/android"
xmlns:tools="http://schemas.android.com/tools"
android:id="@+id/reportActivity"
android:layout_width="match_parent"
android:layout_height="match_parent"
tools:context=".Activity.ReportActivity">

</FrameLayout>

```

Kode Sumber 5.2.4-1 activity_report.xml

2. **ReportActivity.java** merupakan *logic* dari *activity view report*. *Logic* ini berfungsi untuk mengarahkan pengguna menuju menampilkan laporan *sales out* atau menampilkan pencapaian per 6 bulan.

```
package com.kreators.crtoolvl.Activity;

public class ReportActivity extends AppCompatActivity implements
ReportMainFragment.ReportMainListener {

    @Override
    protected void onCreate(Bundle savedInstanceState) {
        setTitle("Report");
        super.onCreate(savedInstanceState);
        setContentView(R.layout.activity_report);
        if (findViewById(R.id.reportActivity) != null) {
            if (savedInstanceState != null) {
                return;
            }
            ReportMainFragment reportMainFragment= new
ReportMainFragment();

            getSupportFragmentManager().beginTransaction().add(R.id.reportActivi
ty, reportMainFragment).commit();
        }

        public void onReportTrackRecordButtonClick(String dateFrom,
String dateTo) {
            ReportTrackRecordFragment reportTrackRecordFragment = new
ReportTrackRecordFragment();
            FragmentTransaction transaction =
getSupportFragmentManager().beginTransaction();
            transaction.replace(R.id.reportActivity,
reportTrackRecordFragment);
            transaction.addToBackStack(null);
            transaction.commit();
        }
    }
}
```

```

        public void onReportSalesOutButtonClick(String dateFrom, String
dateTo) {
            Intent intent = new Intent(this,
ReportSalesOutActivity.class);
            DateParameter dateArgs = new DateParameter(dateFrom, dateTo);
            intent.putExtra(Protocol.DATE_PARAMETER, dateArgs);
            startActivity(intent);
        }
    }
}

```

Kode Sumber 5.2.4-2 ReportActivity.java

3. **fragment_report_main.xml** digunakan untuk membangun tampilan antarmuka utama dari *view report main*. Antarmuka ini memfasilitasi pengguna untuk dapat melihat laporan *sales out* dan pencapaian 6 bulanan.

```

<?xml version="1.0" encoding="utf-8"?>
<LinearLayout
xmlns:android="http://schemas.android.com/apk/res/android"
xmlns:tools="http://schemas.android.com/tools"
android:layout_width="match_parent"
android:layout_height="match_parent"
android:gravity="center"
android:orientation="vertical"
android:padding="@dimen/activity_horizontal_margin"
tools:context=". Fragment. ReportMainFragment">

    <Button
        android:id="@+id/btnReportTrackRecord"
        style="@style/DarkButton"
        android:layout_marginBottom="@dimen/activity_margin_bottom"
        android:drawableLeft="@drawable/ic_timeline_white_36dp"

        android:text="@string/report_track_record" />

    <LinearLayout
        android:layout_width="match_parent"
        android:layout_height="wrap_content"

```

```

        android:orientation="horizontal">

        <TextView
            android:layout_height="wrap_content"
            android:layout_width="@dimen/zero_weight"

android:layout_marginRight="@dimen/activity_horizontal_margin0.5"
            android:layout_weight="1"
            android:textSize="@dimen/font_medium_0"
            android:textStyle="bold"
            android:text="@string/date_from" />

        <TextView
            android:layout_height="wrap_content"
            android:layout_width="@dimen/zero_weight"

android:layout_marginLeft="@dimen/activity_horizontal_margin0.5"
            android:layout_weight="1"
            android:textSize="@dimen/font_medium_0"
            android:textStyle="bold"
            android:text="@string/date_to" />
    </LinearLayout>

    <LinearLayout
        android:layout_width="match_parent"
        android:layout_height="@dimen/tanggal_box_height"
        android:orientation="horizontal">

        <Button
            android:layout_height="match_parent"
            android:id="@+id/btnDateFrom"
            style="@style/Widget.AppCompat.Button.Colored"
            android:layout_width="@dimen/zero_weight"

android:layout_marginTop="@dimen/activity_horizontal_margin0.25"

android:layout_marginRight="@dimen/activity_horizontal_margin0.5"
            android:layout_weight="1"

```

```

android:drawableLeft="@drawable/ic_date_range_white_36dp"
    android:textSize="@dimen/font_small_0" />

    <Button
        android:layout_height="match_parent"

android:layout_marginTop="@dimen/activity_horizontal_margin0.25"
    android:id="@+id/btnDateTo"
    style="@style/Widget.AppCompat.Button.Colored"
    android:enabled="false"
    android:layout_width="@dimen/zero_weight"

android:layout_marginLeft="@dimen/activity_horizontal_margin0.5"
    android:layout_weight="1"

android:drawableLeft="@drawable/ic_date_range_white_36dp"
    android:textSize="@dimen/font_small_0" />
</LinearLayout>

    <Button
        android:id="@+id/btnReportSalesOut"
        style="@style/DarkButton"

android:drawableLeft="@drawable/ic_assignment_turned_in_white_36dp"
    android:text="@string/report_sales_out" />

</LinearLayout>

```

Kode Sumber 5.2.4-3 fragment_report_main.xml

4. **ReportMainFragment.java** merupakan logic dari fragment report main yang berisi implementasi dari ReportActivity.

```

package com.kreators.crtoolv1.Fragment;

public class ReportMainFragment extends Fragment {
    private ReportMainListener activityCallback;

```

```

private View view;
private Button btnFrom, btnTo, btnTR, btnS0;
private Date date1, date2;
private long today, date1long, date2long;
private Calendar toDate, fromDate, pastFiveYear, todayDate;
private static final SimpleDateFormat dateStandartFormatter =
new SimpleDateFormat(Constant.SYSTEM_DATE_STANDART, Locale.US);
private String dateFrom, dateTo;

@Override
public View onCreateView(LayoutInflater inflater, ViewGroup
container, Bundle savedInstanceState) {
    view = inflater.inflate(R.layout.fragment_report_main,
container, false);
    bind();
    setDefaultLayout();
    return view;
}

private void bind() {
    btnFrom = (Button) view.findViewById(R.id.btnDateFrom);
    btnTo = (Button) view.findViewById(R.id.btnDateTo);
    btnTR = (Button)
view.findViewById(R.id.btnReportTrackRecord);
    btnS0 = (Button) view.findViewById(R.id.btnReportSalesOut);
}

private void setDefaultLayout() {
    fromDate = Calendar.getInstance();
    toDate = Calendar.getInstance();
    pastFiveYear = Calendar.getInstance();
    todayDate = Calendar.getInstance();
    fromDate.add(Calendar.YEAR, -5);
    toDate.add(Calendar.DATE, 1);
    todayDate.add(Calendar.DATE, 2);
    pastFiveYear.add(Calendar.YEAR, -5);
}

```

```

        date1long = fromDate.getTimeInMillis();
        date2long = toDate.getTimeInMillis();
        today = Calendar.getInstance().getTimeInMillis();
        date1 = new Date(date1long);
        date2 = new Date(date2long);

btnFrom.setText(IndoCalendarFormat.getFullDate(date1.getTime()));

btnTo.setText(IndoCalendarFormat.getFullDate(date2.getTime()));
    }
    @Override
    public void onActivityCreated(Bundle savedInstanceState) {
        super.onActivityCreated(savedInstanceState);
        btnFrom.setOnClickListener(new View.OnClickListener() {
            @Override
            public void onClick(View v) {
                showCalendarInDialog(btnFrom, date1, true);
            }
        });
        btnTo.setOnClickListener(new View.OnClickListener() {
            @Override
            public void onClick(View v) {
                showCalendarInDialog(btnTo, date2, false);
            }
        });
        btnTR.setOnClickListener(new View.OnClickListener() {
            public void onClick(View v) {
                dateFrom = dateStandartFormatter.format(new
Date(date1.getTime()));
                dateTo =dateStandartFormatter.format(new
Date(date2.getTime()));
                reportTrackRecordButtonClicked(dateFrom, dateTo);
            }
        });
        btnS0.setOnClickListener(new View.OnClickListener() {
            public void onClick(View v) {
                dateFrom = dateStandartFormatter.format(new
Date(date1.getTime()));

```



```

        dateTo =dateStandartFormatter.format(new
Date(date2.getTime()));

        if(btnTo.isEnabled())
reportSalesOutButtonClicked(dateFrom,dateTo); else{
        Toast.makeText(getActivity(),
Constant.selectDateTo, Toast.LENGTH_SHORT).show();
    }
    }
});
}

private void showCalendarInDialog(final Button btn, final Date
date, final boolean isFrom) {
    final Dialog dg = new Dialog(getActivity(),
R.style.FullscreenAppCompatDialogTheme);
    dg.setContentView(R.layout.dialog_calendar);
    final CalendarPickerView dialogView = (CalendarPickerView)
dg.findViewById(R.id.calendarView);

    dialogView.setCustomDayView(new DefaultDayViewAdapter());

    if(isFrom) {
        dialogView.init(pastFiveYear.getTime(),toDate.getTime())
        .inMode(CalendarPickerView.SelectionMode.SINGLE)
        .withSelectedDate(new Date(today));
    } else {
        dialogView.init(fromDate.getTime(), todayDate.getTime())
        .inMode(CalendarPickerView.SelectionMode.SINGLE)
        .withSelectedDate(new Date(today));
    }
    Button btnOK = (Button) dg.findViewById(R.id.btnOK);
    btnOK.setOnClickListener(new View.OnClickListener() {
        @Override
        public void onClick(View v) {
            if(isFrom) {
                fromDate.setTime(dialogView.getSelectedDate());
                btnTo.setEnabled(true);
            }
        }
    });
}

```

```

        } else {
            toDate.setTime(dialogView.getSelectedDate());
        }

        if(fromDate.getTimeInMillis() !=
toDate.getTimeInMillis()) {

btn.setText(IndoCalendarFormat.getFullDate(dialogView.getSelectedDate().getTime()));

date.setTime(dialogView.getSelectedDate().getTime());
        dg.dismiss();
        } else {
            Toast.makeText(getActivity(),
Constant.dontSelectSameDate, Toast.LENGTH_SHORT).show();
        }
    }
});
dg.setOnShowListener(new DialogInterface.OnShowListener() {
    @Override
    public void onShow(DialogInterface dialog) {
        dialogView.fixDialogDimens();
    }
});
dg.show();
}

public void reportTrackRecordButtonClicked(String dateFrom,
String dateTo) {
    activityCallback.onReportTrackRecordButtonClick(dateFrom,
dateTo);
}

public void reportSalesOutButtonClicked(String dateFrom, String
dateTo) {
    activityCallback.onReportSalesOutButtonClick(dateFrom, dateTo);
}

```

```

    }

    public interface ReportMainListener {
        void onReportTrackRecordButtonClick(String dateFrom, String
dateTo);
        void onReportSalesOutButtonClick(String dateFrom, String
dateTo);
    }

    @Override
    public void onAttach(Context context) {
        super.onAttach(context);
        if (context instanceof ReportMainListener) {
            activityCallback = (ReportMainListener) context;
        } else {
            throw new RuntimeException(context.toString()
                + " must implement ReportMainListener");
        }
    }
}
]

```

Kode Sumber 5.2.4-4 ReportMainFragment.java

5. **activity_report_sales_out.xml** digunakan untuk membangun tampilan antarmuka dari *activity report sales out*. Antarmuka ini terdiri dari 2 tab yang berisi *report sales out by date* dan *report sales out by outlet*.

```

<?xml version="1.0" encoding="utf-8"?>
<LinearLayout
xmlns:android="http://schemas.android.com/apk/res/android"
xmlns:app="http://schemas.android.com/apk/res-auto"
xmlns:tools="http://schemas.android.com/tools"
android:id="@+id/reportSalesOutMainContent"
android:layout_width="match_parent"
android:layout_height="match_parent"
android:fitsSystemWindows="true"
tools:context=".Activity.ReportSalesOutActivity"
android:orientation="vertical">

```

```

        <android.support.v7.widget.Toolbar
xmlns:android="http://schemas.android.com/apk/res/android"
        android:id="@+id/toolbarDetailHotel"
        android:layout_width="match_parent"
        android:layout_height="wrap_content"
        android:theme="@style/ThemeOverlay.AppCompat.Dark"
        android:background="@color/colorPrimary"
        android:elevation="4dp">
        </android.support.v7.widget.Toolbar>

        <android.support.design.widget.AppBarLayout
        android:id="@+id/appbar"
        android:layout_width="match_parent"
        android:layout_height="wrap_content"
        android:paddingTop="@dimen/appbar_padding_top"
        android:theme="@style/AppTheme.AppBarOverlay">

                <android.support.design.widget.TabLayout
        android:id="@+id/tabs"
        android:layout_width="match_parent"
        android:layout_height="wrap_content" />

        </android.support.design.widget.AppBarLayout>

        <android.support.v4.view.ViewPager
        android:id="@+id/activityDetailHotel"
        android:layout_width="match_parent"
        android:layout_height="match_parent"
        app:layout_behavior="@string/appbar_scrolling_view_behavior"
/>

</LinearLayout>

```

Kode Sumber 5.2.4-5 activity_report_sales_out.xml

6. **ReportSalesOutActivity.java** merupakan *logic* dari activity *view report*. *Logic* ini berfungsi untuk mengarahkan pengguna menuju menampilkan laporan *sales out* berdasarkan outlet atau berdasarkan tanggal

```
package com.kreators.crtoolvl.Activity;

public class ReportSalesOutActivity extends AppCompatActivity
implements
ReportSalesOutByDateFragment.ReportSalesOutByDateListener,
ReportSalesOutByOutletFragment.ReportSalesOutByOutletListener {
    private ProgressDialog pd;
    private SessionManager session;
    private VolleyManager volleyManager;
    private Toolbar toolbar;
    private ViewPagerAdapter adapter;
    private TabLayout tabLayout;
    private ViewPager viewPager;
    private DateParameter dateParameter;
    private HashMap<String, String> userLogin = new HashMap<>();
    private List<SalesOutReport> salesOutReportList = new
ArrayList<>();
    private ArrayList<Report> crOutletSalesOutList = new
ArrayList<>();
    private ArrayList<Report> crDateSalesOutList = new
ArrayList<>();
    private static final SimpleDateFormat dateStandartFormatter =
new SimpleDateFormat(Constant.SYSTEM_DATE_STANDART, Locale.US);
    private String[] tabsTitles = {Constant.fragmentTitleSalesOut +
Constant.fragmentTitleByDate, Constant.fragmentTitleSalesOut +
Constant.fragmentTitleByOutlet};
    protected FragmentManager fragmentManager;

    @Override
    protected void onCreate(Bundle savedInstanceState) {
        super.onCreate(savedInstanceState);
        setContentView(R.layout.activity_report_sales_out);
    }
}
```

```

        retrieve();
        setUpData();
    }

    private void retrieve() {
        dateParameter =
getIntent().getExtras().getParcelable(Protocol. DATE_PARAMETER);
        volleyManager =
VolleyManager.getInstance(getApplicationContext());
        session = new SessionManager(getApplicationContext());
        userLogin = session.getUserDetails();
        pd = new ProgressDialog(this);
        pd.setMessage("Please wait.");
        pd.setCancelable(false);
        pd.setIndeterminate(true);
    }

    public void bind() {
        fragmentManager = getSupportFragmentManager();
        viewPager = (ViewPager)
findViewById(R.id. activityDetailHotel);
        setupViewPager(viewPager);
        toolbar = (Toolbar) findViewById(R.id. toolbarDetailHotel);
        setSupportActionBar(toolbar);
        tabLayout = (TabLayout) findViewById(R.id. tabs);
        tabLayout.setupWithViewPager(viewPager);
    }

    private void setupViewPager(ViewPager viewPager) {
        adapter = new ViewPagerAdapter(fragmentManager);
        ReportSalesOutByDateFragment reportSalesOutByDateFragment =
new ReportSalesOutByDateFragment();
        Bundle argsDate = new Bundle();

argsDate.putSerializable(Protocol. SN_DATE, crDateSalesOutList);
        reportSalesOutByDateFragment.setArguments(argsDate);
        adapter.addFragment(reportSalesOutByDateFragment,
Constant. fragmentTitleByDate);
        ReportSalesOutByOutletFragment

```

```

reportSalesOutByOutletFragment = new
ReportSalesOutByOutletFragment();
    Bundle argsOutlet = new Bundle();

argsOutlet.putSerializable(Protocol.SN_OUTLET_NAME, crOutletSalesOutL
ist);

    reportSalesOutByOutletFragment.setArguments(argsOutlet);

adapter.addFragment(reportSalesOutByOutletFragment, Constant.fragment
TitleByOutlet);
    viewPager.addOnPageChangeListener(new
ViewPager.OnPageChangeListener() {
        @Override
        public void onPageSelected(int position) {
            // TODO Auto-generated method stub
            toolbar.setTitle(tabsTitles[position]);
        }

        @Override
        public void onPageScrolled(int arg0, float arg1, int
arg2) {
            // TODO Auto-generated method stub
        }
        @Override
        public void onPageScrollStateChanged(int pos) {
            // TODO Auto-generated method stub
        }
    });
    viewPager.setAdapter(adapter);
}

@Override
public void adapterSalesOutByDateButtonClick(Report dateClicked)
{
    Intent intent = new Intent(this,
DetailReportSalesOutActivity.class);
    Bundle args = new Bundle();

```

```

args.putSerializable(Protocol. FRAGMENT_TAG, Constant. inflateFragmentByOutlet);
    args.putSerializable(Protocol. SN_DATE,
dateClicked.getDate());
    args.putParcelableArrayList(Protocol. SO_REPORT, (ArrayList<?
extends Parcelable>) salesOutReportList);
    intent.putExtras(args);
    startActivity(intent);
    finish();
}

@Override
public void adapterSalesOutByOutletButtonClick (Report
outletClicked) {
    Intent intent = new Intent(this,
DetailReportSalesOutActivity.class);
    Bundle args = new Bundle();
    args.putSerializable(Protocol. FRAGMENT_TAG,
Constant. inflateFragmentByDate);
    args.putSerializable(Protocol. SN_OUTLET_NAME,
outletClicked.getDate());
    args.putParcelableArrayList(Protocol. SO_REPORT, (ArrayList<?
extends Parcelable>) salesOutReportList);
    intent.putExtras(args);
    startActivity(intent);
}

private void setUpData() {
    pd.setTitle(Constant. salesOutDialog);
    pd.show();
    GetVolleyRequest request = new
GetVolleyRequest(Uri. GET_SALES_OUT_REPORT);

    request.putParams(Protocol. USERID, userLogin.get(Protocol. USERID));
    request.putParams(Protocol. DATE_FROM, dateParameter.getDateFrom());

```



```

request.putParams(Protocol. DATE_TO, dateParameter.getDateTo());
request.setListener(new VolleyListener() {
    @Override
    public void onSuccess(VolleyRequest request, JSONArray
result) {
        try {
            SalesOutReport salesOutReport;
            JSONObject response;
            boolean status = true;
            String message = "";

            for(int i = 0; i < result.length(); i++) {
                response = result.getJSONObject(i);

                if(response.has(Protocol. STATUS)) {
                    status =
response.getBoolean(Protocol. STATUS);
                    message =
response.getString(Protocol. MESSAGE);
                    break;
                }

                salesOutReport = new
SalesOutReport(response.getLong(Protocol. SN), response.getString(Prot
ocol. SN_OUTLET_NAME),

response.getString(Protocol. SN_DATE),
response.getString(Protocol. SN_ITEM_DESC),

response.getInt(Protocol. SN_STATUS));
                salesOutReportList.add(salesOutReport);
            }

            if (pd != null) {
                pd.dismiss();
            }
        }
    }
}

```

```

        if (status) {
            getCROutletSalesOut();
            getCRDateSalesOut();
            bind();
        } else {
            ReportSalesOutActivity.this.finish();
            Toast.makeText(ReportSalesOutActivity.this,
message, Toast.LENGTH_SHORT).show();
        }

    } catch (JSONException e) {
        e.printStackTrace();
    } catch (ParseException e) {
        e.printStackTrace();
    }
}

@Override
public void onError(VolleyRequest request, String
errorMessage) {
    finish();
    Toast.makeText(ReportSalesOutActivity.this,
errorMessage, Toast.LENGTH_SHORT).show();
    if (pd != null) {
        pd.dismiss();
    }
}
});
volleyManager.createRequest(request, Protocol.GET);
}

private void getCROutletSalesOut () {
    int i, count;
    HashMap<String, String> hashSet = new HashMap<>();

    for(i=0; i < salesOutReportList.size(); i++) {
        if

```

```

(hashSet.get(salesOutReportList.get(i).getOutletName()) == null) {
    count = 1;
} else {
    count =
Integer. valueOf(hashSet.get(salesOutReportList.get(i).getOutletName(
))) + 1 ;
}

hashSet.put(salesOutReportList.get(i).getOutletName(), String. valueOf
(count));
}

    Iterator hashSetIterator = hashSet.keySet().iterator();
    while(hashSetIterator.hasNext()) {
        String key=(String)hashSetIterator.next();
        String value=hashSet.get(key);
        crOutletSalesOutList.add(new Report(key, value));
    }
    hashSet.clear();
}

private void getCRDateSalesOut () throws ParseException {
    int i, count;
    Calendar calendar = Calendar.getInstance();
    Date date;
    HashMap<String, String> hashSet = new HashMap<>();
    for(i=0; i < salesOutReportList.size(); i++) {
        date =
dateStandartFormatter.parse(salesOutReportList.get(i).getPostDate())
;
        calendar.setTime(date);
        if
(hashSet.get(IndoCalendarFormat. getDate(calendar.getTimeInMillis()))
== null) {
            count = 1;
        } else {
            count =
Integer. valueOf(hashSet.get(IndoCalendarFormat. getDate(calendar.getT
imeInMillis())) + 1 ;

```

```

    }

    hashSet.put(IndoCalendarFormat. getDate(calendar.getTimeInMillis()),String. valueOf(count));
    }

    Iterator hashSetIterator = hashSet.keySet().iterator();
    while(hashSetIterator.hasNext()) {
        String key=(String)hashSetIterator.next();
        String value=hashSet.get(key);
        orDateSalesOutList.add(new Report(key, value));
    }
    hashSet.clear();
}
}
}

```

Kode Sumber 5.2.4-6 ReportSalesOutActivity.java

7. **ViewPagerAdapter.java** view pager adapter berfungsi untuk membuat *tabbed layout*, yaitu *layout* yang dapat berubah dengan slide.

```

package com.kreators.crtoolv1.Fragment.Adapter;

public class ViewPagerAdapter extends FragmentPagerAdapter {
    private final List<Fragment> mFragmentManager = new ArrayList<>();
    private final List<String> mFragmentManagerTitleList = new
ArrayList<>();
    public ViewPagerAdapter (FragmentManager manager) {
        super (manager);
    }

    public void addFragment(Fragment fragment, String title) {
        mFragmentManager.add(fragment);
        mFragmentManagerTitleList.add(title);
    }

    @Override
    public CharSequence getPageTitle(int position) {

```

```

        return mFragmentTitleList.get(position);
    }

    @Override
    public Fragment getItem(int position) {
        return mFragmentManager.get(position);
    }

    @Override
    public int getCount() {
        return mFragmentManager.size();
    }
}

```

Kode Sumber 5.2.4-7 ViewPagerAdapter.java

8. **adapter_history_sales_out_row.xml** antarmuka ini digunakan untuk membangun tampilan konsisten *adapter* dari data yang ditampilkan.

```

<?xml version="1.0" encoding="utf-8"?>
<LinearLayout
    xmlns:android="http://schemas.android.com/apk/res/android"
    android:orientation="horizontal"
    android:layout_width="match_parent"
    android:padding="@dimen/activity_horizontal_margin"
    android:layout_height="match_parent">

    <TextView
        android:id="@+id/txtName"
        android:layout_width="wrap_content"
        android:layout_height="wrap_content"
        android:textStyle="bold"
        android:text="Laptop lenovo"/>

    <TextView
        android:id="@+id/txtSize"
        android:layout_width="match_parent"
        android:gravity="end"
        android:layout_height="wrap_content"

```

```

        android:textStyle="bold"
        android:text="10"/>
</LinearLayout>

```

Kode Sumber 5.2.4-8 adapter_history_sales_out_row.xml

9. **HistorySalesOutAdapter.java** adapter ini berfungsi untuk memetakan data sales out yang ingin ditampilkan ke fragment.

```

package com.kreators.crtoolv1.Fragment.Adapter;

public class HistorySalesOutAdapter extends BaseAdapter {

    public Context context;
    public ArrayList<Report> snArrayList;

    public HistorySalesOutAdapter(Context context, ArrayList<Report>
historyArrayList) {
        super();
        this.context = context;
        this.snArrayList = historyArrayList;
    }

    public class HistoryHolder
    {
        TextView name;
        TextView size;
    }

    @Override
    public int getCount() {
        return snArrayList.size();
    }

    @Override
    public Object getItem(int position) {
        return snArrayList.get(position);
    }
}

```

```

@Override
public long getItemId(int position) {
    return position;
}

@Override
public View getView(int position, View convertView, ViewGroup
parent) {
    HistoryHolder holder;
    if (convertView == null) {
        convertView =
LayoutInflater.from(context).inflate(R.layout.adapter_histroy_sales_
out_row, parent, false);
        holder = new HistoryHolder();
        holder.name = (TextView)
convertView.findViewById(R.id.txtName);
        holder.size = (TextView)
convertView.findViewById(R.id.txtSize);
        convertView.setTag(holder);
    }
    else {
        holder = (HistoryHolder) convertView.getTag();
    }

    holder.name.setText(snArrayList.get(position).getDate());
    holder.size.setText(snArrayList.get(position).getSN());

    return convertView;
}
}

```

Kode Sumber 5.2.4-9 HistorySalesOutAdapter.java

10. **fragment_report_sales_out.xml** antarmuka ini digunakan untuk membangun tampilan konsisten dari *report* data yang ingin ditampilkan.

```

<LinearLayout
xmlns:android="http://schemas.android.com/apk/res/android"
xmlns:tools="http://schemas.android.com/tools"
android:layout_width="match_parent"
android:layout_height="match_parent"
android:padding="@dimen/activity_horizontal_margin"
android:orientation="vertical"
tools:context=".Fragment.ReportSalesOutByDateFragment">

    <SearchView
        android:id="@+id/svSearchSN"
        android:layout_width="match_parent"
        android:layout_height="wrap_content" />

    <ListView
        android:id="@+id/lvListViewSN"
        android:layout_width="match_parent"
        android:layout_height="match_parent"/>

</LinearLayout>

```

Kode Sumber 5.2.4-10 fragment_report_sales_out.xml

- 11. ReportSalesOutByDateFragment.java** *fragment* ini berfungsi untuk menampilkan laporan penjualan berdasarkan tanggal.

```

package com.kreators.crtoolv1.Fragment;

public class ReportSalesOutByDateFragment extends Fragment
implements SearchView.OnQueryTextListener {
    private ReportSalesOutByDateListener activityCallBack;
    private SearchView mSearchView;
    private ListView mListView;
    private HistorySalesOutAdapter reportAdapter;
    private List<Report> crDateSalesOutList = new ArrayList<>();
    private InputMethodManager imm;
    private View v;
}

```



```

        @Override
        public View onCreateView(LayoutInflater inflater, ViewGroup
container, Bundle savedInstanceState) {
            v = inflater.inflate(R.layout. fragment_report_sales_out,
container, false);
            retrieve();
            bind();
            setupSearchView();
            return v;
        }

        @Override
        public void onActivityCreated(Bundle savedInstanceState) {
            super.onActivityCreated(savedInstanceState);
            imm =
(InputMethodManager) getActivity().getSystemService(Context. INPUT_MET
HOD_SERVICE);
            imm.hideSoftInputFromWindow(getView().getWindowToken(), 0);
            mListView.setOnItemClickListener(new
AdapterView.OnItemClickListener() {
                @Override
                public void onItemClick(AdapterView<?> parent, View
view, int position, long id) {
                    adapterSalesOutByDateButtonClick(crDateSalesOutList.get(position));
                }
            });
        }

        public void retrieve() {
            getActivity().setTitle(Constant. fragmentTitleSalesOut +
Constant. fragmentTitleByDate);
            crDateSalesOutList = (List<Report>)
getArguments().getSerializable(Protocol. SN_DATE);
        }

```

```

private void bind() {
    mSearchView=(SearchView) v.findViewById(R.id.svSearchSM);
    mListview=(ListView) v.findViewById(R.id.lvListViewSM);
    reportAdapter =new HistorySalesOutAdapter(getActivity(),
(ArrayList<Report>) crDateSalesOutList);
    mListview.setAdapter(reportAdapter);
    mListview.setTextFilterEnabled(true);
}

private void setupSearchView() {
    mSearchView.setIconifiedByDefault(false);
    mSearchView.setOnQueryTextListener(this);
    mSearchView.setSubmitButtonEnabled(true);
    mSearchView.setQueryHint("Search Here");
}

@Override
public boolean onQueryTextChange(String newText) {
    if (TextUtils.isEmpty(newText)) {
        mListview.clearTextFilter();
    } else {
        mListview.setFilterText(newText);
    }
    return true;
}

@Override
public boolean onQueryTextSubmit(String query) {
    return false;
}

@Override
public void onAttach(Context context) {
    super.onAttach(context);
    if (context instanceof ReportSalesOutByDateListener) {
        activityCallback = (ReportSalesOutByDateListener)
context;
    } else {

```

```

        throw new RuntimeException(context.toString()
            + " must implement ReportMainListener");
    }
}
public void adapterSalesOutByDateButtonClick (Report dateClicked)
{
activityCallback.adapterSalesOutByDateButtonClick (dateClicked) ;
}
public interface ReportSalesOutByDateListener {
    void adapterSalesOutByDateButtonClick (Report dateClicked) ;
}
}

```

Kode Sumber 5.2.4-11 ReportSalesOutByDateFragment.java

12. ReportSalesOutByOutletFragment.java *fragment* ini berfungsi untuk menampilkan laporan penjualan berdasarkan outlet.

```

package com.kreators.crtoolv1.Fragment;

public class ReportSalesOutByOutletFragment extends Fragment
implements SearchView.OnQueryTextListener {
    private ReportSalesOutByOutletListener activityCallback;
    private SearchView mSearchView;
    private ListView mListView;
    private HistorySalesOutAdapter reportAdapter;
    private List<Report> crOutletSalesOutList = new ArrayList<>();
    private InputMethodManager imm;
    private View v;

    @Override
    public View onCreateView(LayoutInflater inflater, ViewGroup
container, Bundle savedInstanceState) {
        v = inflater.inflate(R.layout.fragment_report_sales_out,
container, false);
        retrieve();
        bind();
    }
}

```

```

        setupSearchView();
        return v;
    }

    @Override
    public void onCreate(Bundle savedInstanceState) {
        super.onCreate(savedInstanceState);

        imm =
        (InputMethodManager) getActivity().getSystemService(Context.INPUT_METHOD_SERVICE);
        imm.hideSoftInputFromWindow(getView().getWindowToken(), 0);

        mListView.setOnItemClickListener(new
        AdapterView.OnItemClickListener() {
            @Override
            public void onItemClick(AdapterView<?> parent, View
            view, int position, long id) {

                adapterSalesOutByOutletButtonClick(crOutletSalesOutList.get(position));
            }
        });
    }

    private void retrieve() {
        getActivity().setTitle(Constant.fragmentTitleSalesOut +
        Constant.fragmentTitleByOutlet);
        crOutletSalesOutList = (List<Report>)
        getArguments().getSerializable(Protocol.SN_OUTLET_NAME);
    }

    private void bind() {
        mSearchView=(SearchView) v.findViewById(R.id.svSearchSN);
        mListView=(ListView) v.findViewById(R.id.lvListViewSN);
        reportAdapter =new HistorySalesOutAdapter(getActivity(),
        (ArrayList<Report>) crOutletSalesOutList);
    }

```

```

        mListView.setAdapter(reportAdapter);
        mListView.setTextFilterEnabled(true);
    }

    private void setupSearchView() {
        mSearchView.setIconifiedByDefault(false);
        mSearchView.setOnQueryTextListener(this);
        mSearchView.setSubmitButtonEnabled(true);
        mSearchView.setQueryHint("Search Here");
    }

    @Override
    public boolean onQueryTextChange(String newText) {
        if (TextUtils.isEmpty(newText)) {
            mListView.clearTextFilter();
        } else {
            mListView.setFilterText(newText);
        }
        return true;
    }

    @Override
    public boolean onQueryTextSubmit(String query) {
        return false;
    }

    @Override
    public void onAttach(Context context) {
        super.onAttach(context);
        if (context instanceof ReportSalesOutByOutletListener) {
            activityCallBack = (ReportSalesOutByOutletListener)
context;
        } else {
            throw new RuntimeException(context.toString()
                + " must implement ReportMainListener");
        }
    }

    public void adapterSalesOutByOutletButtonClick(Report

```

```

outletClicked) {

activityCallBack.adapterSalesOutByOutletButtonClick (outletClicked) ;
    }
    public interface ReportSalesOutByOutletListener {
        void adapterSalesOutByOutletButtonClick (Report
outletClicked) ;
    }
}

```

Kode Sumber 5.2.4-12 ReportSalesOutByOutletFragment.java

- 13. fragment_history_sales_out.xml** antarmuka ini digunakan untuk membangun tampilan dari laporan berdasarkan nama barang.

```

<LinearLayout
    xmlns:android="http://schemas.android.com/apk/res/android"
    xmlns:tools="http://schemas.android.com/tools"
    android:layout_width="match_parent"
    android:padding="@dimen/activity_horizontal_margin"
    android:layout_height="match_parent"
    android:orientation="vertical"
    tools:context=".Fragment.ReportHistorySalesOutFragment">

    <SearchView
        android:id="@+id/svSearchHistorySalesOut"
        android:layout_width="match_parent"
        android:layout_height="wrap_content" />

    <ListView
        android:id="@+id/lvListViewHistorySalesOut"
        android:layout_width="match_parent"
        android:layout_height="match_parent" />

</LinearLayout>

```

Kode Sumber 5.2.4-13 fragment_history_sales_out.xml

14. ReportHistorySalesOutFragment.java *fragment* ini berfungsi untuk menampilkan laporan penjualan berdasarkan nama barang.

```
package com.kreators.crtoolv1.Fragment;

public class ReportHistorySalesOutFragment extends Fragment
implements SearchView.OnQueryTextListener {
    private View v;
    private static final SimpleDateFormat dateStandartFormatter =
new SimpleDateFormat(Constant.SYSTEM_DATE_STANDART, Locale.US);
    private SearchView mSearchView;
    private ListView mListView;
    private HistorySalesOutAdapter snAdapter;
    private HistorySalesOutListener activityCallBack;
    private String dateSelected, outletSelected;
    private List<SalesOutReport> salesOutReportList = new
ArrayList<>();
    private List<Report> itemName = new ArrayList<>();
    private ArrayList<String> crItemNameList = new ArrayList<>();
    private InputMethodManager imm;

    @Override
    public View onCreateView(LayoutInflater inflater, ViewGroup
container, Bundle savedInstanceState) {
        v = inflater.inflate(R.layout.fragment_history_sales_out,
container, false);
        retrieve();
        try {
            setUpData();
        } catch (ParseException e) {
            e.printStackTrace();
        }
        bind();
        setupSearchView();
        return v;
    }

    private void retrieve() {
```

```

        getActivity().setTitle(Constant. fragmentTitleSalesOut +
Constant. fragmentTitleName);
        dateSelected = (String)
getArguments().getSerializable(Protocol. SN_DATE);
        outletSelected = (String)
getArguments().getSerializable(Protocol. SN_OUTLET_NAME);
        salesOutReportList =
getArguments().getParcelableArrayList(Protocol. SO_REPORT);
    }

    private void setUpData() throws ParseException {
        int num;
        Calendar calendar = Calendar.getInstance();
        Date date;
        itemName.clear();
        crItemNameList.clear();
        for (num=0; num<salesOutReportList.size(); num++) {

if(salesOutReportList.get(num).getOutletName().equals(outletSelected
)) {
            date =
dateStandartFormatter.parse(salesOutReportList.get(num).getPostDate(
));
            calendar.setTime(date);
            if
(IndoCalendarFormat. getDate(calendar.getTimeInMillis()).equals(dateS
elected)) {

crItemNameList.add(salesOutReportList.get(num).getItemDesc());
            }
        }
    }

    HashMap<String, String> hashSet = new HashMap<String,
String>();
    for (num=0; num<crItemNameList.size(); num++) {

hashSet.put(crItemNameList.get(num), String. valueOf(Collections. frequ

```



```

ency(crItemNameList, crItemNameList.get(num)));
    }

    Iterator hashSetIterator = hashSet.keySet().iterator();
    while(hashSetIterator.hasNext()) {
        String key=(String)hashSetIterator.next();
        String value=(String)hashSet.get(key);
        itemName.add(new Report(key,value));
    }
}

@Override
public void onActivityCreated(Bundle savedInstanceState) {
    super.onActivityCreated(savedInstanceState);
    imm =
(InputMethodManager) getActivity().getSystemService(Context.INPUT_METHOD_SERVICE);
    imm.hideSoftInputFromWindow(getView().getWindowToken(), 0);

    mListView.setOnItemClickListener(new
AdapterView.OnItemClickListener() {
        @Override
        public void onItemClick(AdapterView<?> parent, View
view, int position, long id) {

adapterHistorySalesOutButtonClick(dateSelected, outletSelected, itemName.
get(position).getDate());
        }
    });
}

private void bind() {
    mSearchView=(SearchView)
v.findViewById(R.id.svSearchHistorySalesOut);
    mListView=(ListView)
v.findViewById(R.id.lvListViewHistorySalesOut);
    snAdapter=new HistorySalesOutAdapter(getActivity(),

```

```

(ArrayList<Report>) itemName);
    mListView.setAdapter(snAdapter);
    mListView.setTextFilterEnabled(true);
}

private void setupSearchView() {
    mSearchView.setIconifiedByDefault(false);
    mSearchView.setOnQueryTextListener(this);
    mSearchView.setSubmitButtonEnabled(true);
    mSearchView.setQueryHint("Search Here");
}

@Override
public boolean onQueryTextChange(String newText) {
    if (TextUtils.isEmpty(newText)) {
        mListView.clearTextFilter();
    } else {
        mListView.setFilterText(newText);
    }
    return true;
}

@Override
public boolean onQueryTextSubmit(String query) {
    return false;
}

@Override
public void onAttach(Context context) {
    super.onAttach(context);
    if (context instanceof HistorySalesOutListener) {
        activityCallBack = (HistorySalesOutListener) context;
    } else {
        throw new RuntimeException(context.toString()
            + " must implement
OnFragmentInteractionListener");
    }
}
}

```

```

@Override
public void onDetach() {
    super.onDetach();
    activityCallBack=null;
}

    public void adapterHistorySalesOutButtonClick (String date,
String outlet, String name) {

activityCallBack.adapterHistorySalesOutButtonClick (date, outlet, name)
;
    }

    public interface HistorySalesOutListener {
        void adapterHistorySalesOutButtonClick (String date, String
outlet, String name);
    }
}

```

Kode Sumber 5.2.4-14 ReportHistorySalesOutFragment.java

15. **fragment_serial_number_sales_out.xml** antarmuka ini digunakan untuk membangun tampilan dari laporan berdasarkan serial number barang.

```

<LinearLayout
    xmlns:android="http://schemas.android.com/apk/res/android"
    xmlns:tools="http://schemas.android.com/tools"
    android:layout_width="match_parent"
    android:padding="@dimen/activity_horizontal_margin"
    android:layout_height="match_parent"
    android:orientation="vertical"
    tools:context=". Fragment. ReportHistorySalesOutFragment">

    <SearchView
        android:id="@+id/svSearchSerialNumberSalesOut"
        android:layout_width="match_parent"
        android:layout_height="wrap_content" />

    <ListView

```

```

        android:id="@+id/lvListViewSerialNumberSalesOut"
        android:layout_width="match_parent"
        android:layout_height="match_parent" />

```

```
</LinearLayout>
```

Kode Sumber 5.2.4-15 fragment_serial_number_sales_out.xml

16. ReportSerialNumberSalesOutFragment.java *fragment*

ini berfungsi untuk menampilkan laporan penjualan berdasarkan status serial number.

```

package com.kreators.crtoolv1.Fragment;

public class ReportSerialNumberSalesOutFragment extends Fragment
implements SearchView.OnQueryTextListener {
    private View v;
    private SearchView mSearchView;
    private static final SimpleDateFormat dateStandartFormatter =
new SimpleDateFormat(Constant.SYstem_DATE_STANDART, Locale.US);
    private ListView mListView;
    private ArrayList<Report> crReport = new ArrayList<>();
    private HistorySalesOutAdapter snAdapter;
    private String dateSelected, outletSelected, itemSelected;
    private List<SalesOutReport> salesOutReportList = new
ArrayList<>();
    private InputMethodManager imm;

    @Override
    public View onCreateView(LayoutInflater inflater, ViewGroup
container, Bundle savedInstanceState) {
        v =
inflater.inflate(R.layout.fragment_serial_number_sales_out,
container, false);
        retrieve();
        try {
            setUpData();
        } catch (ParseException e) {
            e.printStackTrace();
        }
    }

```

```

        bind();
        setupSearchView();
        return v;
    }

    private void retrieve() {
        getActivity().setTitle(Constant.fragmentTitleSalesOut +
Constant.fragmentTitleSerialNumber);
        dateSelected = (String)
getArguments().getSerializable(Protocol.SN_DATE);
        outletSelected = (String)
getArguments().getSerializable(Protocol.SN_OUTLET_NAME);
        itemSelected = (String)
getArguments().getSerializable(Protocol.SN_ITEM_DESC);
        salesOutReportList =
getArguments().getParcelableArrayList(Protocol.SO_REPORT);
    }

    private void setUpData() throws ParseException {
        int num;
        Calendar calendar = Calendar.getInstance();
        Date date;
        for (num=0; num<salesOutReportList.size(); num++) {

if(salesOutReportList.get(num).getOutletName().equals(outletSelected
)) {
            date =
dateStandartFormatter.parse(salesOutReportList.get(num).getPostDate(
));
            calendar.setTime(date);
            if
(IndoCalendarFormat.getDate(calendar.getTimeInMillis()).equals(dateS
elected)) {

if(salesOutReportList.get(num).getItemDesc().equals(itemSelected)) {
                int status;
                status =
salesOutReportList.get(num).getInctvStatus();

```

```

        String statusInString = "";
        switch(status) {
            case 0:
                statusInString = "Submitted";
                break;
            case 1:
                statusInString = "Received";
                break;
            case 2:
                statusInString = "Approved";
                break;
            case 3:
                statusInString = "Retur";
                break;
            default:
                break;
        }
        crReport.add(new
Report(String.valueOf(salesOutReportList.get(num).getSN()), statusInS
tring));
    }
}
}
}

private void bind() {
    mSearchView=(SearchView)
v.findViewById(R.id.svSearchSerialNumberSalesOut);
    mListView=(ListView)
v.findViewById(R.id.lvListViewSerialNumberSalesOut);
    snAdapter=new HistorySalesOutAdapter(getActivity(),
crReport);
    mListView.setAdapter(snAdapter);
    mListView.setTextFilterEnabled(true);
}

private void setupSearchView() {

```

```

        mSearchView.setIconifiedByDefault(false);
        mSearchView.setOnQueryTextListener(this);
        mSearchView.setSubmitButtonEnabled(true);
        mSearchView.setQueryHint("Search Here");
    }

    @Override
    public boolean onQueryTextChange(String newText) {
        if (TextUtils.isEmpty(newText)) {
            mListView.clearTextFilter();
        } else {
            mListView.setFilterText(newText);
        }
        return true;
    }

    @Override
    public boolean onQueryTextSubmit(String query) {
        return false;
    }

    @Override
    public void onActivityCreated(Bundle savedInstanceState) {
        super.onActivityCreated(savedInstanceState);
        imm =
(InputMethodManager) getActivity().getSystemService(Context.INPUT_METHOD_SERVICE);
        imm.hideSoftInputFromWindow(getView().getWindowToken(), 0);
    }
}

```

Kode Sumber 5.2.4-16 ReportSerialNumberSalesOutFragment.java

5.2.5. Implementasi Sistem Menampilkan Pencapaian CR

Berikut merupakan potongan dari kode antarmuka dan *logic* yang digunakan untuk menampilkan pencapaian CR.

1. **fragment_report_track_record.xml** digunakan untuk membangun tampilan antarmuka dari *view track record*.

```
<?xml version="1.0" encoding="utf-8"?>

<LinearLayout
    android:layout_width="match_parent"
    android:layout_height="match_parent"
    android:orientation="vertical"
    android:padding="@dimen/activity_horizontal_margin"
    xmlns:android="http://schemas.android.com/apk/res/android"
    xmlns:tools="http://schemas.android.com/tools"
    tools:context=". Fragment.ReportTrackRecordFragment">
    <TextView
        android:id="@+id/tvTableHistoryTitle"
        android:layout_width="match_parent"
        android:layout_height="wrap_content"

        android:layout_margin="@dimen/activity_horizontal_margin"
        android:gravity="center"
        android:text="@string/track_record_table_title"
        android:textSize="@dimen/font_small_1"
        android:textStyle="bold" />

    <com.github.mikephil.charting.charts.LineChart
        android:id="@+id/chart"
        android:layout_width="match_parent"
        android:layout_height="250dp" />

    <LinearLayout
        android:layout_height="wrap_content"
        android:layout_width="match_parent"
        android:padding="@dimen/activity_horizontal_margin0.5"
```



```

        android:orientation="horizontal">
        <TextView
            android:text="@string/track_record_bulan"
            android:textStyle="bold"
            android:gravity="center"
            android:layout_weight="0.7"
            style="@style/TextViewTable"/>
        <TextView
            android:text="@string/track_record_appr"
            android:textStyle="bold"
            android:gravity="center"
            style="@style/TextViewTable"/>
        <TextView
            android:text="@string/track_record_target"
            android:textStyle="bold"
            android:gravity="center"
            style="@style/TextViewTable"/>
        <TextView
            android:text="@string/track_record_presentation"
            android:textStyle="bold"
            android:gravity="center"
            android:layout_weight="0.7"
            style="@style/TextViewTable"/>
    </LinearLayout>

    <ListView
        android:id="@+id/lvListViewTrackRecord"
        android:layout_width="match_parent"
        android:layout_height="match_parent" />
</LinearLayout>

```

Kode Sumber 5.2.5-1 fragment_report_track_record.xml

2. **ReportTrackRecordFragment.java** berisi *logic* dari *view track record* yang berfungsi untuk menampilkan pencapaian selama 6 bulan terakhir. Fitur ini juga dilengkapi dengan library MPChart untuk menampilkan grafik.

```

package com.kreators.crtoolv1.Fragment;

public class ReportTrackRecordFragment extends Fragment {

    private View v;
    private static final SimpleDateFormat sdf = new
SimpleDateFormat(Constant. SYSTEM_DATE_STANDART, Locale. US);
    private ListView mListView;
    private ArrayList<TrackRecord> trArrayList;
    private TrackRecordAdapter trAdapter;
    private List<TrackRecord> trackRecordList = new ArrayList<>();
    private SessionManager session;
    private VolleyManager volleyManager;
    private ProgressDialog pd;
    private String currentDate;
    private LineChart lineChart;
    private int monthNow;

    public View onCreateView(LayoutInflater inflater, ViewGroup
container, Bundle savedInstanceState) {
        v = inflater.inflate(R.layout.fragment_report_track_record,
container, false);
        initialization();
        fetchData();
        return v;
    }

    private void initialization() {
        getActivity().setTitle(Constant. fragmentTitleAchievement);
        volleyManager =
VolleyManager.getInstance(getActivity().getApplicationContext());
        session = new
SessionManager(getActivity().getApplicationContext());
        pd = new ProgressDialog(getActivity());
        pd.setMessage("Please wait.");
        pd.setCancelable(false);
        pd.setIndeterminate(true);
        mListView = (ListView)

```

```

v.findViewById(R.id.lvListViewTrackRecord);
    Date dt = new Date();
    currentDate = sdf.format(dt);
}

private void fetchData() {
    pd.setTitle(Constant.salesOutDialog);
    pd.show();
    GetVolleyRequest request = new
GetVolleyRequest(Uri.TRACK_RECORD_REPORT);

request.putParams(Protocol.USERID, session.getUserDetails().get(Proto
col.USERID));
    request.putParams(Protocol.CUR_DATE, currentDate);
    request.setListener(new VolleyListener() {
        @Override
        public void onSuccess(VolleyRequest request, JSONArray
result) {
            try {
                JSONObject response;
                TrackRecord trackRecord;

                for(int i = 0; i < result.length(); i++) {
                    response = result.getJSONObject(i);
                    trackRecord = new
TrackRecord(response.getLong(Protocol.PRICE), response.getString(Prot
ocol.SN_DATE), response.getInt(Protocol.SN_STATUS));
                    trackRecordList.add(trackRecord);
                }
                if (pd != null) {
                    pd.dismiss();
                }
                try {
                    setUpData();
                } catch (ParseException e) {
                    e.printStackTrace();
                }
            }
        }
    });
}

```

```

        } catch (JSONException e) {
            e.printStackTrace();
        }
    }

    @Override
    public void onError(VolleyRequest request, String
errorMessage) {
        Toast.makeText(getActivity(), errorMessage,
Toast.LENGTH_SHORT).show();
        if (pd != null) {
            pd.dismiss();
        }
    }
});
volleyManager.createRequest(request, Protocol.GET);
}

private void setUpData() throws ParseException {

    Calendar calendar = Calendar.getInstance();
    Date date;
    date = sdf.parse(currentDate);
    calendar.setTime(date);
    monthNow =
IndoCalendarFormat.getMonthIndex(calendar.getTimeInMillis());

    int num;
    for (num=0; num<trackRecordList.size(); num++) {
        date = sdf.parse(trackRecordList.get(num).getBulan());
        calendar.setTime(date);

trackRecordList.get(num).setBulan(IndoCalendarFormat.getMonthNameSho
rt(calendar.getTimeInMillis()));
    }

    TrackRecord trackRecordListDummy[] = new TrackRecord[6];
    int monthNowDummy = monthNow;

```

```

        for (num=5; num>=0; num--) {
            String bulanNow =
IndoCalendarFormat.getMonthSimpleNameFromIndex(monthNowDummy);
            monthNowDummy--;
            if (monthNowDummy == 0 ) monthNowDummy = 12;
            trackRecordListDummy[num] = new
TrackRecord(0, bulanNow, 0, 0, 0, 0);
        }

        for (TrackRecord i : trackRecordList) {
            int bulanIndex =
IndoCalendarFormat.getMonthIndexFromSimpleName(i.getBulan());
            num = bulanIndex - monthNow + 5;

trackRecordListDummy[num]. setPrice (trackRecordListDummy[num]. getPr ice
() + i. getPrice());
            int status = i. getStatus();
            switch (status) {
                case 0:

trackRecordListDummy[num]. setSubmitted (trackRecordListDummy[num]. get
Submitted() + 1);

                    break;
                case 1:

trackRecordListDummy[num]. setReceived (trackRecordListDummy[num]. getR
eceived() + 1);

                    break;
                case 2:

trackRecordListDummy[num]. setApproved (trackRecordListDummy[num]. getA
pproved() + 1);

                    break;
                case 3:

trackRecordListDummy[num]. setRetur (trackRecordListDummy[num]. getRetu
r() + 1);

```

```

        break;
    default:
        break;
    }
}
trackRecordList.clear();

for(int i=0;i<6;i++){
    trackRecordList.add(trackRecordListDummy[i]);
}
monthNow = monthNowDummy;

bind();
}

private void bind() {
    lineChart = (LineChart) v.findViewById(R.id.chart);
    lineChart.setDescription("% Pencapaian");
    ArrayList<Entry> entries = new ArrayList<>();
    ArrayList<String> labels = new ArrayList<String>();

    for(int i=0;i<trackRecordList.size();i++) {
        labels.add(trackRecordList.get(i).getBulan());
        entries.add(new
Entry((float)trackRecordList.get(i).getPrice()/3000000,i));
    }
    LineDataSet dataset = new LineDataSet(entries, "CR
Achievement Last 6 Month");
    LineData data = new LineData(labels, dataset);
    dataset.setDrawCubic(true);
    dataset.setDrawFilled(true);
    lineChart.setData(data);
    lineChart.animateY(3000);
    trAdapter=new TrackRecordAdapter(getContext(),
(ArrayList<TrackRecord>) trackRecordList);
    mLstView.setAdapter(trAdapter);
}

```

```

@Override
public void onActivityCreated(Bundle savedInstanceState) {
    super.onActivityCreated(savedInstanceState);
    mListView.setOnItemClickListener(new
AdapterView.OnItemClickListener() {
        @Override
        public void onItemClick(AdapterView<?> parent, View
view, int position, long id) {
            showTrackRecordDetails((TrackRecord)
parent.getItemAtPosition(position));
        }
    });
}

private void showTrackRecordDetails(TrackRecord
trackRecordDetails) {
    FragmentTransaction ft =
getActivity().getFragmentManager().beginTransaction();
    android.app.Fragment prev =
getActivity().getFragmentManager().findFragmentByTag("Report Track
Record");
    if (prev != null) {
        ft.remove(prev);
    }
    TrackRecordDialogFragment TR =
TrackRecordDialogFragment.newInstance(trackRecordDetails);
    TR.show(ft, "Report Track Record");
}
}

```

Kode Sumber 5.2.5-2 ReportTrackRecordFragment.java

3. **fragment_track_record_detail_dialog.xml** digunakan untuk membangun tampilan antarmuka dari *view* detail *track record* bulanan.

```

<?xml version="1.0" encoding="utf-8"?>
<ScrollView
    xmlns:android="http://schemas.android.com/apk/res/android"

```

```

android:layout_width="match_parent"
android:layout_height="match_parent">
<LinearLayout
    android:layout_width="match_parent"
    android:layout_height="wrap_content"
    android:orientation="vertical">
    <Button
        android:text="@string/SN"
        style="@style/DarkButton"/>
    <TextView
        android:id="@+id/tvSubmittedTrackRecordDetail"
        style="@style/SmallTextView"/>
    <TextView
        android:id="@+id/tvReceivedTrackRecordDetail"
        style="@style/SmallTextView"/>
    <TextView
        android:id="@+id/tvApprovedTrackRecordDetail"
        style="@style/SmallTextView"/>
    <TextView
        android:id="@+id/tvReturnedTrackRecordDetail"
        style="@style/SmallTextView"/>

    <Button
        android:text="@string/report_sales_out"
        style="@style/DarkButton"/>
    <TextView
        android:id="@+id/tvJumlahTrackRecordDetail"
        style="@style/SmallTextView"/>
    <TextView
        android:id="@+id/tvTargetTrackRecordDetail"
        style="@style/SmallTextView"/>
    <TextView
        android:id="@+id/tvKurangTrackrecordDetail"
        style="@style/SmallTextView"/>
    <TextView
        android:id="@+id/tvPresentasiTrackRecordDetail"

android:layout_marginBottom="@dimen/activity_horizontal_margin"

```



```

        style="@style/SmallTextView"/>

    </LinearLayout>
</ScrollView>

```

Kode Sumber 5.2.5-3 fragment_track_record_detail_dialog.xml

4. **TrackRecordDialogFragment.java** berisi *logic* dari *view track record* yang berfungsi untuk menampilkan detail penjualan dari bulan yang dipilih.

```

package com.kreators.crtoolv1.Fragment.Dialog;

public class TrackRecordDialogFragment extends DialogFragment {

    private TrackRecord mTrackRecord, trackRecord;
    private TextView
jumlah, approve, received, submitted, retur, target, kurang, presentasi;
    private View rootView;

    public static TrackRecordDialogFragment newInstance(TrackRecord
trackRecordDetails) {
        TrackRecordDialogFragment f = new
TrackRecordDialogFragment();
        Bundle args = new Bundle();
        args.putSerializable(Protocol.TRACK_RECORD_DETAILS,
trackRecordDetails);
        f.setArguments(args);
        return f;
    }

    @Override
    public void onCreate(Bundle savedInstanceState) {
        super.onCreate(savedInstanceState);
        mTrackRecord = (TrackRecord)
getArguments().getSerializable(Protocol.TRACK_RECORD_DETAILS);
    }

    @Override

```

```

    public View onCreateView(LayoutInflater inflater, ViewGroup
container, Bundle savedInstanceState) {
        rootView =
inflater.inflate(R.layout.fragment_track_record_detail_dialog,
container, false);
        bind();
        trackRecord = mTrackRecord;
        getDialog().setTitle(trackRecord.getBulan());
        jumlah.setText("Apr: " +
IndoCurrencyFormat. transformIntegerToRupiah(trackRecord.getPrice()))
;

        target.setText("Target: " +
IndoCurrencyFormat. transformIntegerToRupiah(300000000));
        kurang.setText("Kurang: " +
IndoCurrencyFormat. transformIntegerToRupiah(300000000 -
trackRecord.getPrice()));
        presentasi.setText("Presentasi: " +
String. format("%.2f", ((double)trackRecord.getPrice()/3000000))+"%");
        submitted.setText("Submitted: " +
String. valueOf(trackRecord.getSubmitted()));
        received.setText("Received: " +
String. valueOf(trackRecord.getReceived()));
        approve.setText("Approved: " +
String. valueOf(trackRecord.getApproved()));
        retur.setText("Retur: " +
String. valueOf(trackRecord.getRetur()));
        return rootView;
    }

    private void bind() {
        jumlah = (TextView)
rootView.findViewById(R.id. tvJumlahTrackRecordDetail);
        target = (TextView)
rootView.findViewById(R.id. tvTargetTrackRecordDetail);
        kurang = (TextView)
rootView.findViewById(R.id. tvKurangTrackrecordDetail);
        presentasi = (TextView)
rootView.findViewById(R.id. tvPresentasiTrackRecordDetail);
        submitted = (TextView)

```

```

rootView.findViewById(R.id.tvSubmittedTrackRecordDetail);
    received = (TextView)
rootView.findViewById(R.id.tvReceivedTrackRecordDetail);
    approve = (TextView)
rootView.findViewById(R.id.tvApprovedTrackRecordDetail);
    retur = (TextView)
rootView.findViewById(R.id.tvReturnedTrackRecordDetail);
    }
}

```

Kode Sumber 5.2.5-4 TrackRecordDialogFragment.java

RBTC

5.3. Implementasi *Web Services*

Implementasi *web services* berbasis REST dilakukan menggunakan *framework* CodeIgniter.

5.3.1. Implementasi Controller *Services.php*

Berikut merupakan potongan dari kode yang merupakan kumpulan fungsi yang diimplementasikan pada *web services* agar dapat dipanggil oleh CR Tool. Jenis fungsi HTTP yang diimplementasikan adalah GET dan POST. Contoh kasus jika CR ingin mendapatkan data outlet terdekat, maka sistem CR Tool melakukan *request* GET ke *web services*, kemudian fungsi yang bersangkutan (`outlet_get()`) pada Controller **Services.php** akan memproses dan memanggil fungsi yang bersangkutan (`getOutlet()`) pada Model **Services_model**, kemudian hasil pemrosesan akan dikembalikan sebagai *response* ke CR Tool.

```
<?php
defined('BASEPATH') OR exit('No direct script access
allowed');

// This can be removed if you use __autoload() in
config.php OR use Modular Extensions
require_once APPPATH . '/libraries/REST_Controller';

class Services extends REST_Controller
{
    public function __construct()
    {
        parent::__construct();
        $this->load->helper('url_helper');
        $this->load->model('Services_model');
    }

    public function outlet_get()
    {
        $IDOutlet = $this->get('id');
        $lat = $this->get('lat');
        $lng = $this->get('lng');
```

```

        if($IDOutlet === NULL)
        {
            if($lat && $lng)
            {
                $outlets = $this->Services_model-
>getOutlet(0, $lat, $lng);
            }
            else
            {
                $outlets = $this->Services_model-
>getOutlet();
            }

            if($outlets)
            {
                $this->response($outlets,
REST_Controller::HTTP_OK); // OK (200) being the HTTP
response code
            }
            else
            {
                $this->response([[
                    'status' => FALSE,
                    'message' => 'No outlets were found'
                ]]);
            }
        }

        $IDOutlet = (int) $IDOutlet;

        if ($IDOutlet <= 0)
        {
            $this->response(NULL,
REST_Controller::HTTP_BAD_REQUEST); // BAD_REQUEST (400)
being the HTTP response code
        }
        else
        {
            $outlet = $this->Services_model-
>getOutlet($IDOutlet);

            if($outlet)
            {

```

```

                                $this->response($outlet,
REST_Controller::HTTP_OK);
                                }
                                else
                                {
                                    $this->response([[
                                        'status' => FALSE,
                                        'message' => 'The specified outlet
were not found'
                                    ]], REST_Controller::HTTP_NOT_FOUND); //
NOT_FOUND (404) being the HTTP response code
                                }
                            }
                        }

public function Soreport_get()
{
    $IDCR = $this->get('id');
    $DateFrom = $this->get('datefrom');
    $DateTo = $this->get('dateto');

    if($IDCR && $DateFrom && $DateTo)
    {
        $reports = $this->Services_model-
>getReport($IDCR, $DateFrom, $DateTo);

        if($reports)
        {
            $this->response($reports,
REST_Controller::HTTP_OK);
        }
        else
        {
            $this->response([[
                'status' => FALSE,
                'message' => 'No reports were found'
            ]]);
        }
    }
    else
    {
        $this->response(NULL,
REST_Controller::HTTP_BAD_REQUEST);
    }
}

```

```

    }

    public function achievement_get()
    {
        $IDCR = $this->get('id');
        $date = $this->get('date');

        if($IDCR && $date)
        {
            $reports = $this->Services_model-
>getAchievement($IDCR, $date);

            if($reports)
            {
                $this->response($reports,
REST_Controller::HTTP_OK);
            }
            else
            {
                $this->response([[
                    'status' => FALSE,
                    'message' => 'No achievements were
found'
                ]]);
            }
        }
    }

    public function SO_post()
    {
        $IDCR = $this->post('idCR');
        $SN = $this->post('SN');
        $IDOutlet = $this->post('idoutlet');
        $RegDate = $this->post('date');

        if($IDCR && $SN && $IDOutlet && $RegDate){
            $user = $this->Services_model-
>getUser($IDCR);
            $outlet = $this->Services_model-
>getOutlet($IDOutlet);
            $searched_SN = $this->Services_model-
>getSN($SN);

            if($user && $outlet && $searched_SN)

```

```

        {
            if($outlet[0]->ChannelID !=
$searched_SN[0]->BizcardID)
            {
                $this->response([[
                    'status' => FALSE,
                    'message' => 'SN belong to
another Dealer'
                ]]);
            }

            $submittedSN = $this->Services_model-
>getSNSubmitted($SN);

            if($submittedSN)
            {
                if($submittedSN[0]->CreateUserID ==
$user[0]->ID)
                {
                    $this->response([[
                        'status' => FALSE,
                        'message' => "SN already
submitted by you on ".$submittedSN[0]->RegDate.""
                    ]]);
                }
                else
                {
                    $this->response([[
                        'status' => FALSE,
                        'message' => "SN already
submitted by ".$submittedSN[0]->CreateUser." on
".$submittedSN[0]->RegDate.""
                    ]]);
                }
            }

            $item = $this->Services_model-
>getItem($searched_SN[0]->ItemID);

            $data = array(
                'CreateUserID' => $user[0]->ID,
                'CreateUser' => $user[0]->Name,
                'SN' => $searched_SN[0]->SN,
                'OutletID' => $outlet[0]->ID,

```



```

        'OutletName' => $outlet[0]->Name,
        'RegDate' => $RegDate,
        'ItemID' => $item[0]->ID,
        'ItemDesc' => $item[0]->Description,
        'InctvStatus' => 0,
        'Status' => 1,
        'DealerID' => $searched_SN[0]-
>BizcardID
    );

    $result = $this->Services_model-
>postSalesOut($data);

    if($result)
    {
        $this->response([[
            'status' => TRUE,
            'message' => 'SN submitted'
        ]], REST_Controller::HTTP_CREATED);
        // CREATED (201) being the HTTP response code
    }
    else
    {
        $this->response([[
            'status' => FALSE,
            'message' => 'Failed to submit
SN'

        ]]);
    }
}
else
{
    $this->response([[
        'status' => FALSE,
        'message' => 'SN not found'
    ]]);
}
}
else
{
    $this->response(NULL,
REST_Controller::HTTP_BAD_REQUEST);
}

```

```

    }

    public function login_post()
    {
        $username = $this->post('username');
        $password = $this->post('password');

        if($username && $password)
        {
            $user = $this->Services_model->getUser(0,
$username, $password);

            if($user)
            {
                $SourceID = $user[0]->SourceID;
                if($SourceID)
                {
                    $this->response([[
                        'status' => TRUE,
                        'message' => 'Login succeeded',
                        'ID' => $SourceID
                    ]], REST_Controller::HTTP_OK);
                }
                else
                {
                    $this->response([[
                        'status' => FALSE,
                        'message' => 'You are not
authorized'
                    ]]);
                }
            }
            else
            {
                $this->response([[
                    'status' => FALSE,
                    'message' => 'Wrong username or
password'
                ]]);
            }
        }
        else
        {

```

```

        $this->response(NULL,
REST_Controller::HTTP_BAD_REQUEST);
    }

    }

    public function checkin_post()
    {
        $IDCR = $this->post('idCR');
        $IDOutlet = $this->post('idoutlet');
        $CheckInDate = $this->post('datetime');

        if($IDCR && $IDOutlet && $CheckInDate)
        {
            $result = $this->Services_model-
>updateCR($IDCR, $IDOutlet, $CheckInDate);

            if($result)
            {
                $this->response([[
                    'status' => TRUE,
                    'message' => 'Check in succeeded'
                ]], REST_Controller::HTTP_CREATED);
            }
            else
            {
                $this->response([[
                    'status' => FALSE,
                    'message' => 'Failed to check in'
                ]]);
            }
        }
        else
        {
            $this->response(NULL,
REST_Controller::HTTP_BAD_REQUEST);
        }
    }
}

```

Kode Sumber 5.3.1-1 Services.php

5.3.2. Implementasi Model `Services_model.php`

Berikut merupakan potongan dari kode yang merupakan kumpulan fungsi yang dipanggil oleh Controller **Services.php** untuk mengakses dan melakukan operasi *database* seperti SELECT, INSERT dan UPDATE.

```
<?php
defined('BASEPATH') OR exit('No direct script access
allowed');

class Services_model extends CI_Model {
    public function __construct()
    {
        $this->load->database('default');
    }

    function getUser($sourceID = NULL, $username = NULL,
$password = NULL)
    {
        if($sourceID === NULL || $sourceID == 0)
        {
            if($username && $password)
            {
                $result = $this->db-
>get_where('dbo.mUser', array('userid' => $username,
'password' => $password))->result();
            }
        }
        else
        {
            $result = $this->db->get_where('dbo.mUser',
array('SourceID' => $sourceID))->result();
        }

        return $result;
    }

    function getOutlet($IDOutlet = NULL, $lat = NULL,
$lng = NULL) {
        if($IDOutlet === NULL || $IDOutlet == 0)
        {
            if($lat && $lng)
```

```

        {
            $result = $this->db->query("WITH
GreatCircleDistance AS
        (
            SELECT ID, Code, Name, ( 6371 *
acos( cos( radians(".$lat.") ) * cos( radians( Lat ) )
            * cos( radians( Lng ) -
radians(".$lng.") ) + sin( radians(".$lat.") ) *
sin(radians(Lat)) ) ) AS distance
            FROM dbo.mOutlet
        )
        SELECT ID, Code, Name
        FROM GreatCircleDistance
        WHERE distance < 0.5")->result();
        }
        else
        {
            $result = $this->db->get('dbo.mOutlet')-
>result();
        }
    }
    else
    {
        $result = $this->db-
>get_where('dbo.mOutlet', array('ID' => $IDOutlet))-
>result();
    }

    return $result;
}

function getReport($IDCR, $DateFrom, $DateTo) {
    $result = $this->db->query("
        SELECT SN, OutletName, RegDate, ItemDesc,
InctvStatus
        FROM dbo.tSNRegistration
        WHERE RegDate >= ".'"'.$DateFrom.'"'" AND
RegDate <= ".'"'.$DateTo.'"'" AND CreateUserID =
"'.$IDCR.'"")->result();

    return $result;
}

function getAchievement($IDCR, $date) {

```

```

        $result = $this->db->query("
            SELECT SalesInPrice, RegDate, InctvStatus
            FROM dbo.tSNRegistration
            WHERE RegDate >= DATEADD(month, -6,
                ".$date."') AND RegDate <= ".$date.'" AND
            CreateUserID = ".$IDCR.")->result();

        return $result;
    }

    function getSNSubmitted($SN) {
        $result = $this->db->query("
            SELECT CreateUserID, CreateUser,
            CONVERT(VARCHAR(20),RegDate,103) as RegDate
            FROM dbo.tSNRegistration s
            JOIN dbo.mUser u
            ON s.CreateUserID = u.ID
            WHERE SN = ".$SN.'" AND s.Status = 1
            AND Type IN (1,2)")->result();

        return $result;
    }

    function getSN($SN) {
        $result = $this->db->get_where('dbo.vSN',
        array('SN' => $SN))->result();

        return $result;
    }

    function getItem($ItemID) {
        $result = $this->db->get_where('dbo.vItem',
        array('ID' => $ItemID))->result();

        return $result;
    }

    function postSalesOut($data)
    {
        if($this->db->insert('dbo.tSNRegistration',
        $data))
        {
            return true;
        }
    }

```

```

    }

    function updateCR($IDCR, $IDOutlet, $CheckInDate)
    {
        $result = $this->db->query("
            UPDATE dbo.mCR
            SET CheckInPlace=". $IDOutlet.",
CheckInDate=". $CheckInDate."";
            WHERE ID=". $IDCR."");

        if($result)
        {
            return true;
        }
        else
        {
            return false;
        }
    }
}

```

Kode Sumber 5.3.2-1 Services_model.php

[Halaman ini sengaja dikosongkan]

RBTC

BAB VI

PENGUJIAN DAN EVALUASI

Pengujian dilakukan untuk memastikan kualitas perangkat lunak yang dikembangkan dan kesesuaian hasil eksekusi perangkat lunak dengan analisis dan perancangan perangkat lunak.

6.1. Tujuan Pengujian

Uji coba dilakukan terhadap sistem CR Tool guna mengetahui beberapa hal berikut:

- a. Menguji kebenaran dan ketepatan fungsionalitas keseluruhan sistem CR Tool.
- b. Menguji kesesuaian bahasa pemrograman dan *tools* yang digunakan dalam mengembangkan aplikasi *mobile* CR Tool.

6.2. Kriteria Pengujian

Penilaian atas pencapaian tujuan pengujian didapatkan dengan memperhatikan beberapa hasil yang diharapkan memenuhi kriteria berikut:

- a. Kemampuan sistem CR Tool dalam memenuhi kebutuhan fungsional.
- b. Kesesuaian dalam memenuhi kebutuhan non-fungsional sistem.
 - Kemudahan penggunaan *frontend*.
 - Aspek estetika *frontend*.

6.3. Lingkungan Pengujian

Lingkungan pengujian sistem pada pengerjaan kerja praktik ini dilakukan pada lingkungan dan alat kaku sebagai berikut:

<i>Processor</i>	: Lenovo Vibe K4 Note MediaTex MT6753(8 Core 1.3 GHz)
<i>Memory</i>	: 16.00 GB
<i>RAM</i>	: 3.00 GB
<i>Jenis Device</i>	: Smartphone
<i>Sistem Operasi</i>	: Android Marshmallow (6.0)

6.4. Skenario Pengujian

Skenario pengujian yang akan diujikan adalah melakukan peran sebagai CR / *user* yang akan melakukan pelaporan *sales out*, melihat laporan penjualan dan pencapaiannya. Langkah-langkah dari skenario sebagai berikut:

- User* membuka aplikasi CR Tool.
- User* masuk ke halaman utama aplikasi.
- User* melakukan *login* ke dalam sistem.
- User* memilih *check in*.
- User* melakukan pelaporan *sales out* dengan memasukkan *serial number* produk yang terjual lewat input text.
- User* melakukan pelaporan *sales out* dengan memindai *barcode serial number* dari produk yang terjual.
- User* melihat pencapaiannya per 6 bulan.
- User* melihat laporan *sales out* yang telah dilakukan berdasarkan tanggal atau outlet.

6.5. Evaluasi Pengujian

Hasil pengujian dihasilkan pengamatan lebih lanjut terhadap perilaku sistem aplikasi CR Tool terhadap kasus skenario uji coba. Pengujian dilakukan oleh 3 *user* untuk mencoba sistem yang telah diterapkan. Pada saat salah satu kasus uji coba, *user* diharuskan untuk memperbaharui versi dari Google Play Services-nya.

Kriteria pengujian	Hasil Pengujian
Log in dan menampilkan menu Check In dan Melihat Laporan	Terpenuhi
Melakukan pelaporan <i>sales out</i> baik dengan memasukkan <i>serial number</i> secara manual maupun dengan memindai <i>barcode</i>	Terpenuhi
Melihat pencapaian pegawai per 6 bulan	Terpenuhi
Melihat dan mencari laporan <i>sales out</i> yang telah dilakukan	Terpenuhi
Kehandalan <i>web services</i>	Terpenuhi
Kemudahan penggunaan antarmuka <i>frontend</i>	Terpenuhi

Tabel 6.5-1 Hasil Pengujian

Berdasarkan hasil pengujian yang telah ditampilkan dapat disimpulkan bahwa secara keseluruhan, aplikasi CR Tool telah memenuhi kriteria-kriteria yang telah disebutkan pada sub-bab sebelumnya.

[Halaman ini sengaja dikosongkan]

RBTC

BAB VII

KESIMPULAN DAN SARAN

7.1. Kesimpulan

Kesimpulan yang di dapat setelah melakukan serangkaian kegiatan kerja praktik di PT Kreator dalam perancangan sistem aplikasi *mobile* CR Tool diantaranya adalah sebagai berikut:

- Perancangan dan implementasi sistem CR Tool telah memenuhi semua kebutuhan yang ditentukan.
- *Frontend / user interface* dari CR Tool yang dibuat menggunakan bahasa pemrograman XML cukup mudah dipahami dan digunakan oleh *user*.
- *Backend* dari CR Tool yang dibuat menggunakan bahasa pemrograman Java telah berjalan sesuai fungsi yang diinginkan.
- *Web services* dari CR Tool yang dibuat menggunakan bahasa pemrograman PHP dan JSON telah berjalan dengan cukup handal ketika layanannya dibutuhkan oleh aplikasi.

7.2. Saran

Beberapa saran untuk pembangunan sistem aplikasi *mobile* CR Tool antara lain:

- Mengingat banyaknya spesifikasi layar Smartphone yang beragam, tampilan *frontend* dari CR Tool harus di-optimasi agar lebih *user friendly* pada seluruh Smartphone dengan *platform* Android.
- Perlu dilakukan penelitian terhadap segala kemungkinan pengembangan fitur sistem di masa mendatang.

[Halaman ini sengaja dikosongkan]

RBTC

DAFTAR PUSTAKA

- [1] Android Developer, “Everything You Need to Build on Android,” [Online].
Available :
<https://developer.android.com/studio/features.html>
[Accessed 4 September 2016].
- [2] Wikipedia, “SQL Server Express,” [Online].
Available :
https://en.wikipedia.org/wiki/SQL_Server_Express
[Accessed 4 September 2016].
- [3] JGraph, “Build Interactive Web Diagramming Apps,” [Online].
Available : <https://www.jgraph.com/index.html>
[Accessed 4 September 2016].
- [4] EnvatoTuts+, “An Introduction to Volley,” [Online].
Available : <http://code.tutsplus.com/tutorials/an-introduction-to-volley--cms-23800>
[Accessed 4 September 2016].
- [5] Wikipedia, “Google Play Services,” [Online].
Available :
https://en.wikipedia.org/wiki/Google_Play_Services
[Accessed 4 September 2016].
- [6] GitHub, “Barcode Scanner,” [Online].
Available : <https://github.com/dm77/barcodescanner>
[Accessed 4 September 2016].
- [7] GitHub, “MP Android Chart,” [Online].
Available : <https://github.com/PhilJay/MPAndroidChart>
[Accessed 4 September 2016].

- [8] Wikipedia, “PHP,” [Online].
Available : <https://en.wikipedia.org/wiki/PHP>
[Accessed 4 September 2016].
- [9] JSON, “Pengenalan JSON,” [Online].
Available : <http://www.json.org/json-id.html>
[Accessed 4 September 2016].
- [10] Tutorials Point, “RESTFul Web Services - Introduction,”
[Online].
Available :
http://www.tutorialspoint.com/restful/restful_introduction.htm
[Accessed 4 September 2016].
- [11] Wikipedia, “XML,” [Online].
Available : <https://en.wikipedia.org/wiki/XML>
[Accessed 4 September 2016].
- [12] Wikipedia, “Java,” [Online].
Available :
[https://en.wikipedia.org/wiki/Java_\(programming_language\)](https://en.wikipedia.org/wiki/Java_(programming_language))
[Accessed 4 September 2016].
- [13] CodeIgniter, “CodeIgniter Rocks,” [Online].
Available : <https://www.codeigniter.com/>
[Accessed 4 September 2016].

LAMPIRAN





BIODATA PENULIS I

Nama : Gian Sebastian Anjasmara

Tempat, Tgl Lahir : Bandung, 18 Mei 1995

Jenis Kelamin : Laki-Laki

Kewarganegaraan : Indonesia

Agama : Kristen Protestan

Status : Belum Menikah

Alamat Asal : Taman Rahayu I Blok D3 No. 18

Alamat Surabaya : Kertajaya Indah Regency Blok G5

Telepon : 081910031656

Email : gian13@mhs.if.its.ac.id

PENDIDIKAN FORMAL

2013 – Sekarang : Mahasiswa Teknik Informatika ITS

2010 – 2013 : SMA Santa Angela

2007 – 2010 : SMP Santa Angela

2001 – 2007 : SD Permata Harapan

KEMAMPUAN

- Web Programming (HTML, PHP, CSS, Javascript)
- Web Framework (CodeIgniter, Flask)
- Web Services (REST)
- Mobile Programming (Android Studio)
- Game Programming (Unity)
- Programming (C, C++, C#, Java, Python)
- Database Management System (Oracle, MySQL, Microsoft SQL Server Management Studio, MongoDB)
- Sistem Operasi (Windows 7, Windows 8, Windows 10, Linux Ubuntu)
- Software Perkantoran (Microsoft Office [MS.Word, MS.Excel, MS. Power Point])
- Bahasa (Indonesia, Inggris)

AKADEMIS

Kuliah : Jurusan Teknik Informatika – Fakultas Teknologi
Informasi, Institut Teknologi Sepuluh Nopember
Surabaya

Angkatan : 2013

Semester : 7 (Tujuh)

IPK : 3.75 (Semester 6)

BIODATA PENULIS II

Nama : Julio Anthony Leonard

Tempat, Tgl Lahir : Palu, 21 Agustus 1995

Jenis Kelamin : Laki-Laki

Kewarganegaraan : Indonesia

Agama : Katolik

Status : Belum Menikah

Alamat Asal : Bumi Tinggede Indah Permai II/E4,
Palu, Sulawesi Tengah

Alamat Surabaya : Keputih Tegal Timur No.7 , Surabaya

Telepon : 08563192092

Email : Julio.Leonard13@mhs.if.its.ac.id

Linkedin : <https://id.linkedin.com/in/julio-anthony-leonard-18b91ba4>

PENDIDIKAN FORMAL

2013 – Sekarang : Mahasiswa Teknik Informatika ITS

2010 – 2013 : SMA Negeri Model Terpadu Madani,
Palu

2007 – 2010 : SMP Katolik St.Paulus, Palu

2002 – 2007 : SD Bala Keselamatan, Palu

2001 – 2002 : SD Filadelfia, Makassar

KEMAMPUAN

- Web Programming (HTML, PHP, CSS, Javascript)
- Web Framework (Laravel)
- Mobile Programming (Android Studio)
- Database Management System (Oracle, MySQL)
- Programming (C, C++, Java, Python)
- Data Analytics (Matlab, Weka, Canopy, Hadoop)
- Sistem Operasi (Windows, Linux Ubuntu)
- Software Perkantoran (Microsoft Office [MS.Word, MS.Excel, MS. Power Point], Google Drive)
- Software Development (Github)
- Design (Corel, Photoshop)
- Bahasa (Indonesia, Inggris)

AKADEMIS

Kuliah : Jurusan Teknik Informatika – Fakultas Teknologi
Informasi, Institut Teknologi Sepuluh Nopember
Surabaya

Angkatan : 2013

Semester : 7 (Tujuh)

IPK : 3.77 (semester 6)