APLIKASI DELIVERY ORDER MINIMARKET BERBASIS WEB DAN ANDROID



Disusun oleh:

Aprilia Nisa' Ulya

3.34.15.1.04

Dinar Rhamiya Pramu Zuko 3.34.15.1.08

PROGRAM STUDI D3 TEKNIK INFORMATIKA JURUSAN TEKNIK ELEKTRO POLITEKNIK NEGERI SEMARANG 2018

APLIKASI DELIVERY ORDER MINIMARKET BERBASIS WEB DAN ANDROID



Tugas akhir ini disusun untuk melengkapi sebagian persyaratan menjadi Ahli Madya

Disusun oleh

Aprilia Nisa' Ulya

3.34.15.1.04

Dinar Rhamiya Pramu Zuko 3.34.15.1.08

PROGRAM STUDI D3 TEKNIK INFORMATIKA JURUSAN TEKNIK ELEKTRO POLITEKNIK NEGERI SEMARANG 2018

PERNYATAAN KEASLIAN TUGAS AKHIR

Yang bertanda tangan di bawah ini, saya :

Nama / NIM / Kelas :

Aprilia Nisa' Ulya / 3.34.15.1.04 / IK 3B

Dinar Rhamiya Pramu Zuko / 3.34.15.1.08 / IK 3B

Jurusan : Teknik Elektro

Program Studi : Teknik Informatika

Menyatakan bahwa Tugas Akhir dengan judul APLIKASI DELIVERY ORDER

MINIMARKET BERBASIS WEB DAN ANDROID yang dibuat untuk melengkapi sebagian persyaratan menjadi Ahli Madya pada Program Studi Teknik Informatika, Jurusan Teknik Elektro Politeknik Negeri Semarang, bukan merupakan tiruan atau duplikasi dari Tugas Akhir yang sudah dipublikasikan dan pernah dipakai untuk mendapatkan gelar Ahli Madya di lingkungan Politeknik Negeri Semarang maupun di Perguruan Tinggi atau Instansi manapun, kecuali bagian yang sumber informasinya dicantumkan sebagaimana mestinya.

Semarang, 15 Agustus 2018

Mahasiswa, Mahasiswa,

Aprilia Nisa' Ulya Dinar Rhamiya Pramu Zuko

NIM. 3.34.15.1.04 NIM. 3.34.15.1.08

HALAMAN PERSETUJUAN

Tugas akhir/skripsi dengan judul **APLIKASI** *DELIVERY ORDER* **MINIMARKET BERBASIS WEB DAN ANDROID** dibuat untuk melengkapi sebagian persyaratan menjadi Ahli Madya/Sarjana Terapan pada Program Studi D3-Teknik Informatika, Jurusan Teknik Elektro Politeknik Negeri Semarang dan disetujui untuk diajukan dalam sidang ujian tugas akhir/skripsi.

Semarang, 15 Agustus 2018

Pembimbing I,

Pembimbing II,

Mardiyono S.Kom, M.Sc

Slamet Handoko, S.Kom, M.Kom

NIP. 197403112000121001

NIP. 197501302001121001

Mengetahui,

Ketua Program Studi Teknik Informatika

<u>Sukamto, S.Kom, M.T</u> NIP. 197101172003121001

HALAMAN PENGESAHAN

Tugas akhir/skripsi dengan judul **APLIKASI** *DELIVERY ORDER* **MINIMARKET BERBASIS WEB DAN ANDROID** telah dipertahankan dalam ujian wawancara dan diterima sebagai syarat untuk menjadi Ahli Madya/Sarjana Terapan pada Program Studi D3 — Teknik Informatika Jurusan Teknik Elektro Politeknik Negeri Semarang pada tanggal 15 Agustus 2018.

Tim Penguji

Penguji 1, Penguji 2, Penguji 3,

Prayitno, SST., M.T. Sukamto, S.Kom, M.T. Liliek Triyono, S.T., M.Kom.

NIP. 198504102014041002 NIP. 197101172003121001 NIP. 198404202015041003

Ketua, Sekretaris,

 Mardiyono,S.Kom.,M.Sc.
 Netty Nurdiyani, Dra, M.Hum.

 NIP. 197403112000121001
 NIP. 196107101988112001

Mengesahkan, Ketua Jurusan Teknik Elektro

<u>Dr. Amin Suharjono,S.T.,M.T.</u> NIP. 197210271999031002

KATA PENGANTAR

Puji syukur kehadirat Allah SWT yang telah memberikan rahmat, berkah, dan hidayah-Nya sehingga penulis dapat menyelesaikan Tugas Akhir dengan judul Aplikasi *Delivery Order* Minimarket Berbasis Web dan Android. Tugas Akhir ini disusun untuk melengkapi syarat akhir studi Program Diploma III pada Program Studi Teknik Informatika Jurusan Elektro Politeknik Negeri Semarang.

Dalam menyelesaikan Tugas Akhir ini penulis banyak memperolah bantuan dari berbagai pihak. Oleh karena itu pada kesempatan ini penulis ingin menyampaikan ucapan terimakasih kepada:

- 1. Bapak Ir. Supriyadi, M.T. selaku Direktur Politeknik Negeri Semarang.
- 2. Bapak Dr. Amin Suharjono, S.T., M.T. selaku Ketua Jurusan Teknik Elektro Politeknik Negeri Semarang.
- 3. Bapak Sukamto, S.Kom., M.T. selaku Ketua Program Studi Teknik Informatika Politeknik Negeri Semarang.
- 4. Bapak Budi Suyanto S.Kom., M.Eng. selaku wali kelas Teknik Informatika B angkatan 2015.
- 5. Bapak Mardiyono S.Kom., M.Sc. dan Bapak Slamet Handoko, S.Kom., M.Kom. selaku dosen pembimbing yang telah banyak membantu, memberikan dukungan dan bantuan berupa solusi, ide pengembangan, serta masukan yang berguna bagi penulis.
- 6. Seluruh dosen pengajar dan staf laboratorium komputer Program Studi Teknik Informatika Politeknik Negeri Semarang yang turut memberikan bantuan dan dukungan.
- 7. Kedua orang tua dan keluarga dari penulis yang selalu memberikan dukungan dan doa yang tulus.
- 8. Teman-teman seperjuangan yang turut membantu dalam proses mengerjakan tugas akhir ini serta memberikan motivasi kepada penulis.

vi

Penulis menyadari bahwa dalam penyusunan laporan Tugas Akhir ini masih jauh dari kesempurnaan. Oleh karena itu kritik dan saran yang membangun dibutuhkan untuk penyempurnaan laporan Tugas Akhir ini.

Semarang, 15 Agustus 2018

Penulis

ABSTRAK

Aprilia Nisa' Ulya dan Dinar Rhamiya Pramu Zuko, "Aplikasi Delivery Order Minimarket Berbasis Web dan Android", Tugas Akhir DIII Jurusan Teknik Elektro Politeknik Negeri Semarang, di bawah bimbingan Mardiyono, S. Kom., M. Sc. dan Slamet Handoko, S.Kom, M.Kom., Agustus 2018, 231 halaman.

Layanan pesan antar pada toko yang masih menggunakan sistem messenger yaitu melakukan pesanan lewat sosial media seperti WhatsApp, Line, BBM dan pembeli tidak bisa secara langsung melihat jenis produk beserta harga-harganya. Pemesanan seperti itu dirasa kurang efektif karena pesan yang masuk akan tumpang tindih dengan pesan sebelumnya dan pembeli juga tidak bisa memonitor status pesanan. Tujuan pembuatan aplikasi ini untuk membangun aplikasi yang dapat mempermudah sistem pemesanan produk pada minimarket sehingga pelanggan yang tidak sempat datang ke toko bisa berbelanja melalui aplikasi. Aplikasi ini menyediakan fitur pesan layan antar meliputi pembelian produk kurang dari atau sama dengan 50 ribu, melakukan pesanan dalam radius kurang dari atau sama dengan 20 kilometer dari lokasi toko, melihat detail produk, fitur lacak posisi kurir, fitur promo produk, fitur monitor status pesanan, dan fitur tracking rute kurir. Aplikasi Delivery Order Minimarket Berbasis Web dan Android dibuat menggunakan metode Waterfall dimana setiap langkah penelitian dilakukan secara berurutan mulai dari tahap analisis, perancangan sistem, penulisan kode program, dan pengujian sistem. Pengujian dilakukan mengunakan dua cara yaitu pengujian fungsional dan pengujian kepuasan pengguna menggunakan metode kuisioner. Aplikasi web delivery order minimarket telah diuji pada tiga browser yang berbeda dengan hasil semua fungsi dapat berjalan dengan baik. Aplikasi android delivery order minimarket telah diuji pada tiga sistem operasi android yang berbeda dengan hasil semua fungsi berjalan dengan baik, dan pengujian kepada 10 responden dan didapatkan hasil presentase sebesar 81.6% yang berarti pengguna merasa sangat puas dengan sistem ini.

Kata kunci: Android, Delivery Order, Minimarket, Web, Tracking

DAFTAR ISI

JUDI	UL	i
PERI	NYATAAN KEASLIAN TUGAS AKHIR	ii
PERS	SETUJUAN	iii
PEN	GESAHAN	iv
KAT	A PENGANTAR	v
ABS'	TRAK	vii
DAF	TAR ISI	viii
DAF	TAR GAMBAR	X
DAF	TAR TABEL	xiii
DAF	TAR LAMPIRAN	xiv
BAB	I PENDAHULUAN	1
1.	1 Latar Belakang	1
1.	2 Perumusan Masalah	2
1.	3 Manfaat	3
1.	4 Tujuan	3
1.	5 Batasan Masalah	3
1.	6 Sistematika Penulisan	4
BAB	II TINJAUAN PUSTAKA	5
2.	1. Penelitian Terkait	5
2.	2. Teknologi yang Digunakan	10
2.	2.1 MySQL	10
2.	2.2 Android	11
2.	2.3 Laravel	12
BAB	III KEGIATAN PELAKSANAAN	13
3.	1 Analisis Kebutuhan	14
3.	1.1. Analisis Kebutuhan Perangkat Keras	14
3.	1.2. Analisis Kebutuhan Perangkat Lunak	14
3.	2 Perancangan Sistem	15
3.	2.1 Gambaran Umum Sistem	15

3.2.2 Pemodelan Sistem	16
3.2.2.1Data flow Diagram (DFD)	16
3.2.2.2Entitiy Relationship Diagram (ERD)	22
3.2.2.3 Sequence Diagram	24
3.2.3 Perancangan <i>Database</i>	24
3.2.3.1 Relasi Tabels	24
3.2.3.2Struktur Tabel Basis Data	25
3.2.4 Perancangan Antarmuka Sistem	27
3.2.4.1 Perancangan Antarmuka Website	27
3.2.4.2 Perancangan Antarmuka Android	37
3.3 Penulisan Kode Program	49
3.3.1 Penulisan Kode Program Website	49
3.3.2 Penulisan Kode Program Android	56
3.4 Pengujian Sistem	62
3.4.1 Pengujian Fungsional Sistem	62
3.4.1.1Rancangan Uji Website Admin	62
3.4.1.2Rancangan Uji Website Manajer	64
3.4.1.3Rancangan Uji Android Pelanggan	65
3.4.1.4Rancangan Uji Android Kurir	67
3.4.2 Pengujian Oleh Pengguna	68
3.5 Pemeliharaan Aplikasi	70
BAB IV ANALISIS DAN PEMBAHASAN	71
4.1 Analisis Fungsional Sistem	71
4.1.1 Web GENmart	71
4.1.2 Android GENmart	90
4.2 Analisis Kepuasan Pengguna	108
BAB V KESIMPULAN	111
DAFTAR PUSTAKA	112
LAMPIRAN	113

DAFTAR GAMBAR

Gambar 3. 1 Model Waterfall Pressman	13
Gambar 3. 2 Gambaran Umum Sistem	15
Gambar 3. 3 Diagram konteks aplikasi delivery order minimarket	17
Gambar 3. 4 Diagram level 0 aplikasi delivery order minimarket	19
Gambar 3. 5 Diagram level 1 proses pendataan	20
Gambar 3. 6 Diagram level 1 Proses Pemesanan	21
Gambar 3. 7 Diagram level 1 Proses Pelaporan	22
Gambar 3. 8 ERD aplikasi delivery order minimarket	23
Gambar 3. 9 Sequence diagram delivery order minimarket	24
Gambar 3. 10 Relasi Tabel Delivery order Minimarket	25
Gambar 3. 11 Halaman login	27
Gambar 3. 12 Dashboard admin toko	28
Gambar 3. 13 Halaman master <i>user</i>	29
Gambar 3. 14 Halaman tambah <i>user</i>	29
Gambar 3. 15 Halaman edit <i>user</i>	30
Gambar 3. 16 Halaman master produk	31
Gambar 3. 17 Halaman tambah produk	31
Gambar 3. 18 Halaman edit produk	32
Gambar 3. 19 Halaman data transaksi	33
Gambar 3. 20 Dashboard manajer	33
Gambar 3. 21 Halaman laporan transaksi	34
Gambar 3. 22 Halaman laporan transaksi harian	35
Gambar 3. 23 Halaman laporan transaksi bulanan	35
Gambar 3. 24 Halaman laporan harian pdf	36
Gambar 3. 25 Halaman bulanan pdf	37
Gambar 3. 26 Halaman <i>login</i> pelanggan	38
Gambar 3. 27 Halaman home	39
Gambar 3. 28 navigasi Bar	40
Gambar 3. 29 Halaman makanan	41
Gambar 3. 30 Halaman minuman	41

Gambar 3. 31 Halaman kesehatan	42
Gambar 3. 32 Halaman rumah tangga	42
Gambar 3. 33 Halaman pesanan saya	43
Gambar 3. 34 Halaman detail produk	44
Gambar 3. 35 Halaman troli saya	44
Gambar 3. 36 Halaman login kurir	45
Gambar 3. 37 Halaman Pengiriman	46
Gambar 3. 38 Navigasi bar	47
Gambar 3. 39 Halaman maps	48
Gambar 4. 1 Halaman <i>log in</i> admin toko	72
Gambar 4. 2 Dashboard admin toko	73
Gambar 4. 3 Halaman master <i>user</i>	74
Gambar 4. 4 Halaman tambah <i>user</i>	74
Gambar 4. 5 Berhasil tambah <i>user</i>	75
Gambar 4. 6 Halaman edit <i>user</i>	76
Gambar 4. 7 Berhasil edit <i>user</i>	76
Gambar 4. 8 Halaman hapus <i>user</i>	77
Gambar 4. 9 Berhasil hapus <i>user</i>	77
Gambar 4. 10 Halaman master produk	78
Gambar 4. 11 halaman tambah produk	79
Gambar 4. 12 Berhasil tambah produk	79
Gambar 4. 13 Halaman edit produk	80
Gambar 4. 14 Berhasil edit produk	80
Gambar 4. 15 Halaman hapus produk	81
Gambar 4. 16 Berhasil hapus produk	81
Gambar 4. 17 halaman stock warning	82
Gambar 4. 18 Halaman data transaksi	83
Gambar 4. 19 Halaman hapus transaksi	83
Gambar 4. 20 Berhasil hapus transaksi	84
Gambar 4. 21 Halaman log in manajer	84
Gambar 4. 22 Dashboard manajer	85
Gambar 4, 23 Halaman data transaksi	86

Gambar 4. 24 Halaman data laporan harian	86
Gambar 4. 25 Halaman laporan harian pdf	87
Gambar 4. 26 Halaman data laporan bulanan	88
Gambar 4. 27 Halaman laporan bulan pdf	89
Gambar 4. 28 Tampilan pengujian halaman login	91
Gambar 4. 29 Tampilan pengujian menu utama	91
Gambar 4. 30 Tampilan pengujian navigasi bar	92
Gambar 4. 31 Tampilan pengujian detail produk	93
Gambar 4. 32 Tampilan pengujian jumlah pembelian	93
Gambar 4. 33 Tampilan pengujian troli	94
Gambar 4. 34 Tampilan <i>form</i> pengiriman	94
Gambar 4. 35 Tampilan kategori makanan	95
Gambar 4. 36 Tampilan kategori minuman	96
Gambar 4. 37 Tampilan kategori kesehatan	96
Gambar 4. 38 Tampilan kategori rumah tangga	97
Gambar 4. 39 Menu pesanan saya	98
Gambar 4. 40 Status pickup barang	98
Gambar 4. 41 Status dalam pengiriman	99
Gambar 4. 42 Lacak posisi kurir	99
Gambar 4. 43 Status telah diterima	100
Gambar 4. 44 Tampilan pengujian halaman login	101
Gambar 4. 45 Notifikasi kurir	101
Gambar 4. 46 Status sedang diproses	102
Gambar 4. 47 Pop up pickup barang	102
Gambar 4. 48 Status <i>pickup</i> barang	103
Gambar 4. 49 Pop up dalam pengiriman	104
Gambar 4. 50 Status dalam pengiriman	104
Gambar 4. 51 Pop up telah diterima	105
Gambar 4. 52 Menu <i>history</i>	106
Gambar 4. 53 Menu map	106
Gambar 4. 54 Tracking rute	107

DAFTAR TABEL

Tabel 2. 1 Perbandingan aplikasi delivery order minimarket dengan aplikasi
delivery order yang sudah ada
Tabel 3. 1 Tabel <i>Users</i>
Tabel 3. 2 Tabel Produks
Tabel 3. 3 Tabel Pesanans
Tabel 3. 4 Hasil pengujian fungsional sistem
Tabel 3. 5 Aspek Tingkat Kepuasan Pelanggan
Tabel 3. 6 Indikator Penilaian
Tabel 3. 7 Indikator Kepuasan Pemakai Aplikasi
Tabel 4. 1 Hasil pengujian fungsional web <i>delivery order minimarket</i> dengan 3
browser berbeda
Tabel 4. 2 Hasil pengujian fungsional aplikasi android GENmart perbelanjaan pada
sistem operasi yang berbeda
Tabel 4. 3 Hasil pengujian fungsional aplikasi android GENmart kurir pada sistem
operasi yang berbeda
Tabel 4. 4Hasil penilaian kuesioner

DAFTAR LAMPIRAN

Lampiran 1	Kode Program Web	. 113
Lampiran 2	Kode Program Android	. 157

BAB I PENDAHULUAN

1.1 Latar Belakang

Layanan pesan antar pada toko yang masih menggunakan sistem *messenger* yaitu melakukan pesanan lewat sosial media seperti *WhatsApp, Line, BBM* dan pembeli tidak bisa secara langsung melihat jenis produk beserta harga-harganya. Pemesanan seperti itu dirasa kurang efektif karena pesan yang masuk akan tumpang tindih dengan pesan sebelumnya dan pembeli juga tidak bisa memonitor status pesanan. Maka dari itu, sangat diperlukan sebuah aplikasi yang dapat mempermudah sistem pemesanan produk di toko, memiliki *User Interface* yang mudah dipahami oleh pengguna, dan dapat mempermudah pengelolaan minimarket.

Banyak inovasi aplikasi pengelolaan toko yang sudah dikembangkan. Salah satu contoh adalah aplikasi layanan pesan antar makanan pada sistem operasi android. Dengan aplikasi tersebut, pemesanan dapat dilakukan dengan mudah dan menghemat waktu serta pemesan dapat menggunakan layanan pengiriman ke lokasi pemesan sehingga pemesan dapat membeli makanan tanpa harus keluar rumah atau kantor (Akbar, 2014). Namun kelemahan dari aplikasi ini adalah belum adanya *tracking* pesanan berbasis lokasi sehingga kurir akan kesulitan menemukan lokasi pemesan. Aplikasi selanjutnya yaitu aplikasi layanan pesan antar makanan di toko roti Vania *bakery*. Aplikasi ini mempunyai fitur layanan pesan antar melalui web ataupun aplikasi android sehingga konsumen dapat memesan roti (Antonia, Budhi,

& Purba, 2016). Namun pada aplikasi ini belum terdapat fitur lokasi untuk mengetahui lokasi pasti pemesan dan untuk melacak kurir yang mengantarkan pesanannnya. Aplikasi layanan pesan antar makanan yang diterapkan di area

Telkom University digunakan kepada mahasiswa untuk mempermudah memesan makanan di rumah makan sekitar Telkom University. Pemesanan pada aplikasi ini dilakukan melalui web dan sistem akan otomatis mengirim sms gateway atau konfirmasi ke rumah makan yang berkaitan. Kelebihan dari aplikasi ini adalah menggunakan sms gateway dengan tools Gammu yang dapat memberi sms langsung kepada rumah makan bersangkutan (Samudra, 2015). Namun pada aplikasi ini belum ada aplikasi mobile dan belum ada fitur lokasinya, sehingga saat

melakukan pengantaran akan sedikit kesulitan apabila belum mengetahui lokasi pastinya. Aplikasi pesan antar di Wulan's cakery sudah menggunakan teknologi Google Maps sehingga memudahkan kurir untuk mencari rute ke lokasi pemesan (Hastutik & Sediyono, 2015). Namun aplikasi ini tidak berbasis web serta antarmuka pengguna yang masih sederhana belum adanya ikon pada lacak kurir. Aplikasi layanan pada rumah makan podoteko, merupakan aplikasi yang nantinya bisa membantu mempermudah masyarakat untuk memesan produk di restoran atau rumah makan podoteko serta dapat mempermudah kinerja administrator ataupun pemilik dalam proses pengolahan data pesan antar makan dan minuman (Purnomo Rahmat, & Nurdin Akbar, 2017). Namun aplikasi ini masih ada kekurangan yaitu fitur tracking lokasi restoran atau rumah makan dari lokasi pengguna yang memanfaatkan GPS, Google Maps dan belum adanya fitur notifikasi berupa bunyi untuk memberikan informasi ke admin bahwa ada pesanan masuk. Aplikasi yang sudah dibuat sudah menyediakan fitur delivery order melalui web dan android sehingga mempermudah untuk pemesanan produk, namun aplikasi sebelumnya belum mempunyai fitur tracking rute, fitur lokasi, serta monitor status pesanan.

Berdasarkan permasalahan diatas, maka Tugas Akhir ini membahas tentang pembangunan aplikasi *delivery order* minimarket berbasis web dan android yang dapat mempermudah sistem pemesanan produk pada minimarket. Aplikasi ini selain menyediakan fitur standart pesan layan antar meliputi pembelian produk kurang dari atau sama dengan 50 ribu, melakukan pesanan dalam radius kurang dari atau sama dengan 20 kilometer dari lokasi toko, melihat detail produk, fitur lacak posisi kurir, fitur promo produk, fitur monitor status pesanan, dan fitur *tracking rute* kurir. Hal ini tentu berguna bagi pelanggan yang tidak sempat datang ke toko untuk berbelanja maupun bagi pelanggan yang akan berbelanja dengan jumlah yang banyak agar tidak kerepotan membawanya.

1.2 Perumusan Masalah

Berdasarkan latar belakang yang telah diuraikan, dapat disimpulkan bahwa terdapat beberapa permasalahan yaitu :

- 1) Bagaimana membuat aplikasi *delivery order* minimarket berbasis web dan android.
- 2) Bagaimana menguji aplikasi *delivery order* minimarket berbasis web dan android secara fungsional dan uji kepuasan pengguna.

1.3 Manfaat

Manfaat dari aplikasi yang akan diterapkan pada minimarket diharapkan memberikan nilai tambah, diantaranya :

- 1) Memudahkan minimarket untuk mengelola layanan pesan antar.
- 2) Memudahkan admin dalam mengelola data pada minimarket.
- 3) Memudahkan pelanggan dalam memesan produk dengan tampilan antarmuka pengguna yang lebih menarik dengan berbagai data produk dan promo.
- 4) Memudahkan pelanggan dalam menggunakan layanan pesan antar.
- 5) Memudahkan kurir dalam mengelola pesanan.
- 6) Memudahkan kurir dalam mengantar pesanan pelanggan.
- 7) Memudahkan manajer dalam memantau transaksi layanan pesan antar dengan sajian berupa dokumen laporan.

1.4 Tujuan

Pembuatan aplikasi delivery order minimarket ini bertujuan untuk :

- 1) Membuat aplikasi *delivery order* minimarket berbasis web dan android.
- 2) Menguji aplikasi *delivery order* minimarket secara fungsional dan uji kepuasan pengguna.

1.5 Batasan Masalah

Dalam pembuatan Tugas Akhir ini, masalah yang akan dibahas terbatas pada:

- 1) Aplikasi ini hanya meliputi pembelian dengan sistem *delivery order* ke tempat tujuan.
- 2) Dalam implementasinya, aplikasi ini berbasis android untuk pelanggan dan kurir, serta aplikasi berbasis web untuk halaman admin toko dan manajer.
- 3) Entitas yang terlibat dalam aplikasi *delivery order* minimarket adalah admin toko, manajer, kurir, dan pelanggan.

1.6 Sistematika Penulisan

Sistematika penulisan ini terdiri dari beberapa bab dan sub bab sebagai berikut:

BAB I PENDAHULUAN

Bab pendahuluan terdiri dari beberapa sub bab, yang berisi latar belakang masalah, rumusan masalah, yang menjadi kajian program ini, tujuan yang hendak dicapai, manfaat, metode penelitian yang digunakan, serta sistematika penulisan.

BAB II TINJAUAN PUSTAKA

Bab ini berisi tentang penelitian-penelitian yang terkait dan perbandingan antara penelitian dan aplikasi yang dibuat.

BAB III KEGIATAN PELAKSANAAN

Bab ini membahas kegiatan pelaksanaan pembuatan aplikasi delivery order.

BAB IV ANALISIS DAN PEMBAHASAN

Bab ini membahas analisis dan pembahasan aplikasi pada website dan perangkat android.

BAB V KESIMPULAN

Kesimpulan merupakan pernyataan singkat dan tepat yang dijabarkan dari hasil analisis dan pembahasan, tanpa perlu penulisan saran dan pengembangan lebih lanjut.

DAFTAR PUSTAKA

LAMPIRAN

BAB II TINJAUAN PUSTAKA

Bab ini membahas penelitian terkait dari aplikasi *delivery order* minimarket yang dibuat serta teknologi yang digunakan dalam pembuatan aplikasi *delivery order* minimarket. Dibahas pula perbandingan antara aplikasi *delivery order* minimarket yang dibuat dengan aplikasi yang sudah ada sebelumnya.

2.1. Penelitian Terkait

Sebelumnya sudah ada beberapa penelitian terkait mengenai aplikasi *delivery order* yang dibuat pada Tugas Akhir ini. Terdapat 10 makalah yang dibahas pada bagian ini meliputi layanan pesan antar makanan (Hendra Prabowo, Yulina, & Sari, 2016), layanan pesan antar pada Vania *bakery* (Antonia, Budhi, & Purba, 2016), layanan pesan antar *laundry* (Exstrilia, Andjarwirawan, & Dewi, 2017), layanan pesan antar makanan di area *Telkom University* (Samudra, 2015), *delivery order* pada Wulan's *cakery* (Hastutik & Sediyono, 2015), layanan pesan antar *Pizza Hut* (Maimunah, Supriyanti, & Hendrian, 2017), layanan pesan antar restoran (Akbar, 2014), layanan pesan antar rumah makan Podoteko (Purnomo Rahmat, & Nurdin Akbar, 2017), layanan pesan antar katering (Dewi Srie,& Arnie Rintana, 2014), layanan pesan antar toko bunga (Sanjaya Kelvin, Purba Kristo, Andjarwirawan Justinus, 2014). Masingmasing memiliki fitur unggulan yang berbeda-beda.

Aplikasi *delivery order* sudah banyak diterapkan pada restoran makanan. Salah satunya adalah aplikasi *Food Deliver* yang dibangun untuk mempermudah konsumen memesan makanan melalui *smartphone* android. Kelebihan dari aplikasi ini yaitu memiliki fitur lokasi atau maps untuk mempermudah kurir mengantar makanan kepada pemesan serta pada aplikasi ini dapat melakukan pesan antar dari beberapa restoran (Hendra Prabowo, Yulina, & Sari, 2016). Namun walaupun aplikasi ini sudah memiliki fitur maps, aplikasi *food deliver* belum memiliki fitur lacak posisi kurir dan belum memiliki fitur produk promo yang tampil dimuka serta belum adanya monitoring status pesanan pada *smartphone* android.

Aplikasi layanan pesan antar juga sudah mulai diterapkan pada beberapa toko roti contohnya *Vania bakery*. Aplikasi layanan pesan antar berbasis web dan android ini memmeberikan kemudahan kepada pelanggan untuk memesan roti melalui *smartphone*. Aplikasi layanan pesan antar Vania *bakery* memiliki kelebihan yaitu adanya fitur pemesanan melalui web dan android (Antonia, Budhi, & Purba, 2016). Namun aplikasi ini belum memiliki fitur lokasi atau lacak kurir sehingga masih menyulitkan dalam pengiriman.

Tidak hanya diterapkan pada restoran saja, aplikasi layanan pesan antar juga diterapkan pada usaha *laundry* di daerah Siwalankerto, Surabaya. Aplikasi layanan pesan antar *laundry* ini memudahkan pelanggan untuk ambil dan antar *laundry* di tempat. Aplikasi android layanan pesan antar *laundry* ini dibangun dengan fitur *Google Maps* untuk mengetahui lokasi *laundry* terdekat (Exstrilia, Andjarwirawan, & Dewi, 2017). Namun aplikasi ini belum memiliki aplikasi berbasis web untuk pengelolaan admin dan belum memiliki fitur lacak posisi.

Aplikasi delivery order berbasis web juga sudah diterapkan di area Telkom University. Aplikasi ini dibuat dengan tujuan mempermudah mahasiswa untuk memesan makanan di rumah makan sekitar kampus. Pemesanan pada aplikasi ini dilakukan melalui web kemudian sistem akan secara otomatis mengirimkan sms gateway kepada rumah makan terkait untuk melayani pesanan konsumen dan melakukan delivery ke tempat pemesan (Samudra, 2015). Namun aplikasi ini belum memiliki aplikasi delivery order berbasis android sehingga masih kurang efektif jika konsumen harus membuka web terlebih dahulu ketika ingin memesan makanan.

Penerapan aplikasi *delivery order* berbasis android dan *Google Maps* pada toko roti juga diterapkan pada Wulan's *cakery*. Aplikasi ini dibangun seiring meningkatnya jumlah konsumen dan kesulitan kurir untuk mengetahui rute lokasi ke tempat pemesan. Dengan adanya aplikasi *delivery order* roti berbasis android, pembeli dapat memesan melalui aplikasi dan pesanan akan diantar oleh kurir. Kurir akan mengantar ke lokasi pemesan dengan rute yang tersedia di aplikasi (Hastutik &

Sediyono, 2015). Namun aplikasi *delivery order* Wulan's *Cakery* belum memiliki aplikasi *web based* dan belum memiliki antarmuka pengguna yang menarik yaitu belum memiliki ikon kurir pada fitur lacak.

Pada perusahaan-perusahaan makanan besar seperti *Pizza Hut* juga sudah mulai menerapkan aplikasi layanan pesan antar berbasis android. Dengan adanya persaingan di dunia bisnis, para pelaku bisnis saling berlomba-lomba dalam meningkatkan jangkauan pasar melalui kemajuan teknologi. Outlet *Pizza Hut* sudah menggunakan aplikasi layanan pesan antar *pizza* melalui smartphone android (Maimunah, Supriyanti, & Hendrian, 2017). Namun berdasarkan survey penggunaan, aplikasi layanan pesan antar *Pizza Hut* belum memiliki fitur lacak posisi kurir yang mengantar.

Aplikasi layanan pesan antar juga dibuat pada restoran makanan untuk mempermudah proses transaksi pemesanan dan *delivery*. Aplikasi ini merupakan sistem informasi dengan *mobile device* yang menyediakan proses pemesanan menu makanan restoran yang dilengkapi dengan tampilan gambar dan daftar harga yang sesuai dengan jenis makanan yang tersedia. Proses pengiriman makanan dilakukan secara manual oleh kurir yang bertugas. Dengan demikian pelanggan yang menggunakan sistem ini dapat lebih mudah melakukan proses pemesanan serta dapat menghemat biaya (Akbar, 2014). Namun aplikasi ini masih ada kekurangan yaitu belum adanya penggunaan sistem *mapping* agar kurir yang mengantar dapat ditracking oleh pemesan.

Aplikasi layanan pesan antar lain juga diterapkan pada rumah makan Podoteko. Aplikasi ini merupakan sistem yang nantinya bisa membantu mempermudah masyarakat untuk memesan produk di restoran atau rumah makan Podoteko serta dapat mempermudah kinerja administrator ataupun pemilik dalam proses pengolahan data pesan antar makan dan minuman (Purnomo Rahmat, & Nurdin Akbar, 2017). Namun aplikasi ini masih ada kekurangan yaitu fitur *tracking* lokasi restoran atau rumah makan dari lokasi pengguna yang memanfaatkan GPS dan

Google Map dan belum adanya fitur notifikasi berupa bunyi untuk memberikan informasi ke admin bahwa ada pesanan masuk.

Perusahaan katering juga sudah mulai menggunakan aplikasi untuk kemudahan sistem layanan pesan antar pada usaha kateringnya. Salah satunya aplikasi layanan pesan antar yang diterapkan pada usaha katering rumah makan Srie. Aplikasi ini berbasis web yang mampu menampilkan daftar menu yang tersedia dan memberikan fasilitas pemesanan bagi pelanggan sehingga memudahkan pelanggan dalam menentukan paket katering yang diinginkan dan memesannya. Aplikasi ini menyediakan informasi yang lengkap bagi pelanggan tentang ketersediaan pemesanan, menu, dan harga makanan tanpa harus menempatkan pegawai yang harus selalu siap sedia menjawab panggilan pelanggan (Dewi Srie,& Arnie Rintana, 2014). Namun aplikasi ini masih ada kekurangan yaitu belum adanya aplikasi berbasis android yang digunakan, sehingga masih menyusahkan pembeli jika ingin memesan makanan.

Selain diterapkan pada berbagai restoran makanan, aplikasi *delivery order* juga diterapkan pada toko bunga. Salah satunya aplikasi yang diterapkan pada toko bunga X. Pembuatan aplikasi layanan pesan antar pada toko bunga X menggunakan HTML 5 dan dibantu kompilasi oleh Phonegap. Aplikasi ini menggunakan fitur chat yang mempermudah melakukan pemesanan bunga di daerah Surabaya dan aplikasi ini menggunakan *Google Map* yang bertujuan untuk mempermudah melacak posisi tempat pengiriman Bunga (Sanjaya Kelvin, Purba Kristo, Andjarwirawan Justinus, 2014). Namun aplikasi ini masih memiliki desain yang kurang menarik, sehingga belum menarik minat konsumen untuk menggunakan aplikasi ini.

Setelah membahas beberapa aplikasi terkait yang sudah pernah dibuat dan dipublikasikan pada jurnal ilmiah, selanjutnya akan ditunjukkan perbandingan antara aplikasi yang sudah pernah dibuat dengan aplikasi *delivery order* pada minimarket yang dibuat pada Tugas Akhir ini. Perbandingan tersebut dapat dilihat pada Tabel 2.1.

Tabel 2. 1 Perbandingan aplikasi *delivery order* minimarket dengan aplikasi *delivery order* yang sudah ada

Aplikasi yang telah dibuat	Fitur aplikasi yang telah dimiliki			
	Web Based	Aplikasi Android	GPS	Tracking
Aplikasi <i>Food Deliver</i> Online Berbasis Android dan Web				
Aplikasi <i>Delivery order</i> Toko Roti Vania Bakery				
Aplikasi <i>Delivery order</i> dengan Menggunakan HTML5 Berbasis Mobile				
Aplikasi Delivery Makanan berbasis Web di Area Telkom <i>University</i>				
Aplikasi <i>Delivery order</i> Berbasis Android Menggunakan Teknologi <i>Google Maps</i> dan GPS				
Aplikasi Sistem Order Online Berbasis Mobile Android pada Outlet Pizza Hut <i>Delivery</i>	X			
Pembuatan Aplikasi Layanan Pesan Antar Makanan Pada Sistem Operasi Android				
Aplikasi Layanan <i>Delivery Order</i> Berbasis Web Pada Rumah Makan Podoteko		X	X	X
Aplikasi Layanan Pemesanan Katering Pada Rumah Makan Srie Berbasis Web		X	X	X
Pembuatan Aplikasi <i>Delivery</i> Order Bunga pada Toko X Berbasis Mobile	X		X	X
Aplikasi Delivery order Minimarket Berbasis Web dan Android				

Berdasarkan Tabel 2.1, kelebihan aplikasi *delivery order* minimarket ini adalah memiliki aplikasi berbasis web yang dikelola oleh admin, serta sudah adanya dengan fitur lacak posisi kurir, fitur promo produk dan fitur *delivery order*, serta fitur monitor status pesanan.

2.2. Teknologi yang Digunakan

Teknologi yang digunakan pada pembuatan aplikasi *delivery order* minimarket diantaranya adalah MySQL, Android, Laravel, Location Based System. Penjelasan teknologi yang digunakan adalah sebagai berikut:

2.2.1 **MySQL**

MySQL adalah salah satu aplikasi RDBMS (*Relational Database Management System*). Pengertian sederhana RDBMS adalah aplikasi database yang menggunakan prinsip relasional. MySQL bersifat gratis dan *open source*, artinya setiap orang boleh menggunakan dan mengembangkan aplikasi ini. Namun walaupun gratis, MySQL di *support* oleh ribuan programmer dari seluruh dunia dan merupakan sebuah aplikasi RDBMS yang lengkap, cepat, dan reliabel.

SQL (*Structured Query Language*) adalah bahasa pemrograman khusus yang digunakan untuk memanajemen data dalam RDBMS. SQL biasanya berupa perintah sederhana yang berisi instruksi-instruksi untuk manipulasi data. Perintah SQL ini sering juga disingkat dengan sebutan 'query'.

Jenis-jenis perintah SQL

Perintah atau instruksi SQL dapat dikelompokkan berdasarkan jenis dan fungsinya. Terdapat 3 jenis perintah dasar SQL yaitu *Data Definition Language*, *Data Manipulation Language* dan *Data Control Language*.

- 1) Data Definition Language (DDL) adalah jenis instruksi SQL yang berkaitan dengan pembuatan struktur tabel maupun database. Termasuk diantaranya: CREATE, DROP, ALTER, dan RENAME.
- 2) Data Manipulation Language (DML) adalah jenis instruksi SQL yang berkaitan dengan data yang ada dalam tabel, tentang bagaiman menginput, menghapus, memperbaharui serta membaca data yang tersimpan di dalam

- database. Contoh perintah SQL untuk DML : SELECT, INSERT, DELETE, dan UPDATE.
- 3) Data Control Language (DCL) adalah jenis instruksi SQL yang berkaitan dengan manajemen hak akses dan pengguna (user) yang dapat mengakses database maupun tabel. Termasuk diantaranya: GRANT dan REVOKE.

Pada pembuatan Aplikasi *Delivery order Minimarket*, MySQL digunakan sebagai database server yang akan menyimpan semua data-data yang dibutuhkan pada aplikasi.

2.2.2 Android

Android merupakan subset perangkat lunak untuk perangkat mobile yang meliputi sistem operasi, *middleware*, dan aplikasi inti yang dirilis oleh Google. Android adalah sistem operasi bergerak (*mobile operating system*) yang mengadopsi sistem operasi linux, namun telah dimodifikasi. Android diambil alih oleh Google pada tahun 2005 dari Android,Inc sebagai bagian strategi utuk mengisi pasar sistem operasi bergerak. Google mengambil alih seluruh hasil kerja Android termasuk tim yang mengembangkan Android. (Suprianto & Agustina, 2012).

1) Arsitektur Android

Secara sederhana arsitektur Android merupakan sebuah kernel Linux dan sekumpulan pustaka *Java* dalam suatu *framework* yang mengatur dan menyediakan alur proses aplikasi.

2) XML

XML merupakan singkatan dari eXtensible Markup language. XML adalah sekumpulan aturan-aturan yang mendefinisikan suatu sintaks yang digunakan untuk menjelaskan dan mendeskripsikan teks atau data dalam sebuah dokumen melalui penggunaan tag. XML terletak pada inti Web Service yang digunakan untuk mendeskripsikan data. Fungsi utama dari XML adalah komunikasi antar aplikasi, integrasi data, dan aplikasi-aplikasi yang berbeda dengan partner luaran. Dengan standarisasi XML, aplikasi-aplikasi yang berbeda dapat dengan mudah berkomunikasi antar satu dengan yang lain (Akhmad Dharma Kasman, 2013).

Pada pembuatan aplikasi *delivery order* minimarket, Android digunakan sebagai device yang akan diinstal aplikasi utama kemudian aplikasinya dibuat menggunakan software *Android Studio*.

2.2.3 Laravel

Laravel adalah sebuah framework PHP yang dirilis dibawah lisensi MIT, dan sedang *happening* baru-baru ini karena berbagai kemudahannya. Laravel adalah framework pengembangan website yang mengadopsi konsep MVC (Model View Controller) yang tentunya memudahkan dan mengefesiensi waktu dan biaya pembuatan serta pemeliharaan sistem. Laravel juga menyediakan sintaks yang ekspresif, efektif, jelas dan dapat menghemat waktu pengerjaan. Konsep MVC dibedakan yaitu model sebagai sesuatu yang mewakili struktur data yang membantu dalam pengelolaan basis data. Kemudian view adalah bagian yang mengatur view atau tampilan kepada pengguna. Dan controller adalah sesuatu yang menjembatani antara model dan view.

Pada pembuatan aplikasi *delivery order* minimarket, framework Laravel digunakan untuk koding aplikasi *web based* atau untuk membuat halaman admin sebagai pengelola aplikasi.

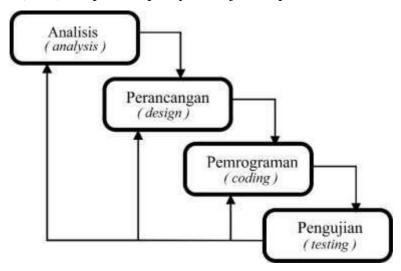
2.2.4 Location Based System

Pemanfaatan aplikasi dengan menggunakan Sistem Layanan Berbasis Lokasi, atau lebih dikenal dengan *Location based Service* (LBS) dapat menggabungkan antar proses dari layanan mobile dengan posisi geografis dari penggunanya. Posisi pemesan layanan dan pengantar layanan dapat ditentukan dengan mudah dengan adanya LBS. Dengan adanya LBS pula pelacakan pengantar layanan dapat dilakukan dengan mudah. Aplikasi layanan pesan antar barang ini dapat memberikan kemudahan kepada pihak penyedia layanan dan mereka yang menggunakan layanannya.

Pada pembuatan Aplikasi *Delivery order Minimarket*, *Location Based System* digunakan untuk membuat fitur maps dan tracking pada aplikasi androidnya.

BAB III KEGIATAN PELAKSANAAN

Bab ini membahas tahapan-tahapan yang dilakukan pada pembuatan Tugas Akhir. Metode yang digunakan pada pembuatan tugas akhir aplikasi *delivery order* minimarket berbasis web dan android adalah metode *Waterfall* menurut Rosa dan Shalahuddin (2011), adapun tahapannya ditunjukkan pada Gambar 3.1.



Gambar 3. 1 Model Waterfall Rosa dan Shalahudin

Gambar 3.1 merupakan metode *Waterfall* Rosa dan Shalahudin yang digunakan dalam pembuatan aplikasi *delivery order* minimarket dan memiliki 4 tahap yaitu analisis, perancangan, pemrograman, dan pengujian. Tahap analisis meliputi analisis kebutuhan yang dijadikan bahan dalam pembuatan aplikasi *delivery order* minimarket. Tahap perancangan digunakan untuk mendefinisikan arsitektur sistem secara keseluruhan, hasil dari tahap ini berupa desain sistem yang membantu dalam menentukan perangkat keras dan sistem persyaratan. Tahap pemrograman merupakan tahap pembuatan aplikasi *delivery order* minimarket baik dari segi sistem maupun alat sesuai dengan perancangan sistem dari tahap sebelumnya. Tahap pengujian merupakan tahap yang terakhir, aplikasi *delivery order* minimarket yang sudah dibuat diuji fungsionalitasnya untuk mengecek apakah ditemukan kesalahan atau tidak yang kemudian akan dilakukan perbaikan agar aplikasi *delivery order* minimarket dapat bekerja dengan baik sesuai dengan fungsinya. Alasan menggunakan metode *waterfall* Rosa dan Shaladin karena proses pembuatan aplikasi ini sampai ke tahap pengujian saja.

3.1 Analisis Kebutuhan

Analisis Kebutuhan merupakan tahap awal dalam melakukan perancangan sistem dengan cara mengidentifikasi kebutuhan perangkat lunak, perangkat keras yang dibutuhkan untuk membangun aplikasi dan untuk menjalankan aplikasi serta data yang dibutuhkan berupa informasi yang diperoleh dengan melakukan wawancara terhadap pemilik toko minimarket untuk mendapatkan data yang benar.

3.1.1. Analisis Kebutuhan Perangkat Keras

Spesifikasi perangkat keras minimal yang digunakan untuk membangun aplikasi *delivery order* minimarket berbasis web dan android diantaranya adalah:

- 1) Komputer
 - a) Sistem Operasi minimal Windows 7
 - b) Prosesor Intel i3
 - c) Memori RAM 2GB
 - d) HDD 500GB
- 2) Smartphone
 - a) Sistem Operasi minimal Android 5.0
 - b) Memori RAM minimal 1 GB
 - c) Memori Internal 8 GB

3.1.2. Analisis Kebutuhan Perangkat Lunak

Perangkat lunak untuk membangun aplikasi *delivery order* minimarket berbasis web dan android diantaranya adalah:

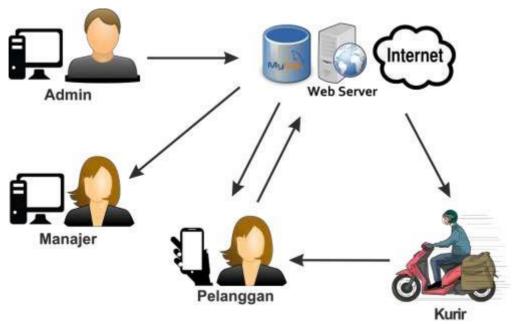
- a) Android Studio
- b) Browser (Mozila Firefox atau Google Chrome)
- c) Sistem Operasi Windows
- d) Sublime
- e) XAMPP
- f) PHP (Hypertext Preprocessor)

3.2 Perancangan Sistem

Perancangan sistem yang digunakan yaitu menggunakan DFD atau *Data Flow Diagram*, ERD atau *Entitiy Relationship Diagram* sebagai media untuk menyampaikan rancangan sistem aplikasi tersebut, dan *Diagram Squence*. *Data Flow Diagram* (DFD) adalah suatu diagram yang menggunakan notasi-notasi untuk menggambarkan arus dari data sistem, yang penggunaannya sangat membantu untuk memahami sistem secara logika, terstruktur dan jelas. DFD merupakan alat bantu dalam menggambarkan atau menjelaskan sistem yang sedang berjalan secara logis (Setyo & Imam, 2015). *Entity Relationship diagram* (ERD) merupakan teknik yang digunakan untuk memodelkan kebutuhan data dari suatu sistem (Brady & Loonam, 2010). *Sequence Diagram* untuk memperlihatkan interaksi antar obyek dalam perintah yang berurut dan mendefinisikan urutan kejadian yang dapat menghasilkan *output* yang diinginkan.

3.2.1 Gambaran Umum Sistem

Aplikasi *delivery order* minimarket merupakan sistem aplikasi berbasis web dan android dimana antarmuka fungsi-fungsi dari android dan website dibangun menggunakan *script* HTML, PHP, dan Java. Sedangkan untuk penyimpanan data menggunakan *database* MySQL seperti terlihat pada Gambar 3.2.



Gambar 3. 2 Gambaran Umum Sistem

Prinsip kerja dari aplikasi ini secara umum adalah menyediakan layanan *delivery orer* untuk pelanggan yang berbelanja pada sebuah minimarket dengan total belanja lebih dari atau sama dengan Rp. 50.000 dan dengan radius jarak kurang dari atau sama dengan 20 kilometer dari lokasi toko. Pada halaman muka android terdapat produk-produk apa saja yang dijual pada minimarket tersebut sehingga memudahkan pelanggan untuk melakukan pemesanan barang pada minimarket. Fitur lainnya adalah monitoring status pesanan yang memudahkan pelanggan mengetahui status pesanannya. Terdapat dua aplikasi android yaitu aplikasi perbelanjaan yang digunakan oleh *user* dan aplikasi *delivery* yang digunakan oleh kurir. Kemudian pada aplikasi website hanya digunakan oleh admin dan manajer saja. Admin memiliki hak akses untuk mengatur data-data yang ada pada aplikasi. Kemudian manager hanya dapat mengakses laporan penjualannya saja.

3.2.2 Pemodelan Sistem

Perancangan aplikasi *delivery order* minimarket menggunakan desain pemodelan DFD sebagai media untuk menyampaikan rancangan sistem. Selain itu juga akan ditampilkan ERD dan *Sequence Diagram* dari aplikasi *delivery order* minimarket yang telah dibuat.

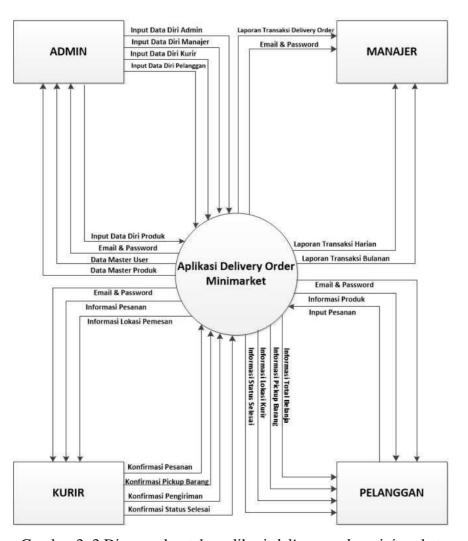
3.2.2.1 Data flow Diagram (DFD)

a) Diagram Konteks

Diagram Konteks menggambarkan satu lingkaran besar yang dapat mewakili seluruh proses yang terdapat di dalam suatu sistem. Semua entitas eksternal yang ditunjukkan pada diagram konteks berikut aliran-aliran data utama menuju dan dari sistem. Diagram ini sama sekali tidak memuat penyimpanan data dan tampak sederhana untuk diciptakan. Diagram konteks aplikasi *delivery order* minimarket berbasis Web dan Android dapat dilihat pada Gambar 3.3.

Berdasarkan Gambar 3.3 terdapat 4 entitas yang berinteraksi dengan aplikasi delivery order minimarket yaitu admin, manajer, pelanggan, dan kurir. Aliran data pada aplikasi berupa proses *input* dan *output* keseluruhan entitas ke dalam sistem. Dapat kita lihat bahwa admin dapat mengelola data diri *user* yang menggunakan aplikasi, diantaranya admin itu sendiri, manajer, pelanggan, dan kurir. Kemudian

admin juga dapat mengelola data produk dengan menambahkan produk baru, mengupdate data produk, dan menghapus data produk yang tidak dibutuhkan lagi. Kemudian manajer akan mendapatkan e-mail dan password yang digunakan untuk login ke dalam aplikasi, laporan data transaksi keseluruhan, laporan transaksi harian, dan laporan transaksi bulanan. Pelanggan akan mendapatkan e-mail dan password yang digunakan untuk login ke dalam aplikasi, pelanggan mendapatkan informasi produk, memasukkan pesanan ke dalam aplikasi, mendapatkan informasi status pesanan, dan pelanggan dapat melacak posisi kurir saat mengantar pesanannya. Kemudian kurir mendapatkan e-mail dan password yang digunakan untuk login ke dalam sistem, informasi pemesanan, kurir juga dapat mengkonfirmasi status pesanan pelanggan, dan dapat melakukan pengiriman ke lokasi pemesan dengan menggunakan fitur maps.

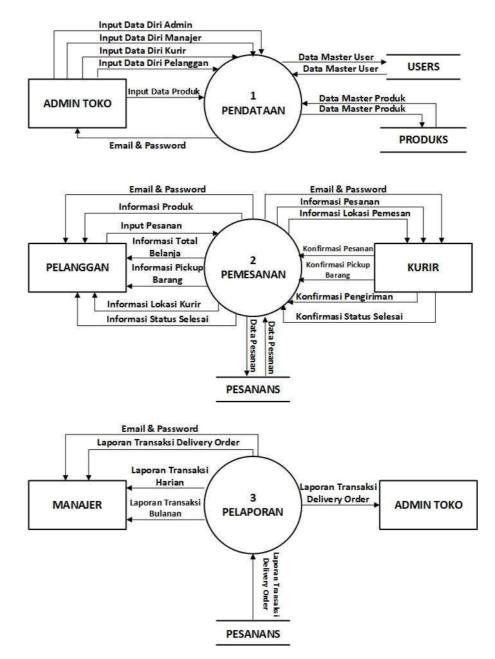


Gambar 3. 3 Diagram konteks aplikasi delivery order minimarket

b) Diagram level 0

Diagram level 0 merupakan pemecah dari diagram konteks, diagram ini memuat penyimpanan data. Diagram level 0 dibagi menjadi 3 proses yaitu pendataan, pemesanan, dan pelaporan serta 4 entitas yaitu admin toko, manajer, pelanggan, dan kurir. Diagram level 0 memiliki 3 tabel yaitu users, produks, dan pesanans. Diagram level 0 Aplikasi *delivery order* minimarket dapat dilihat pada Gambar 3.4.

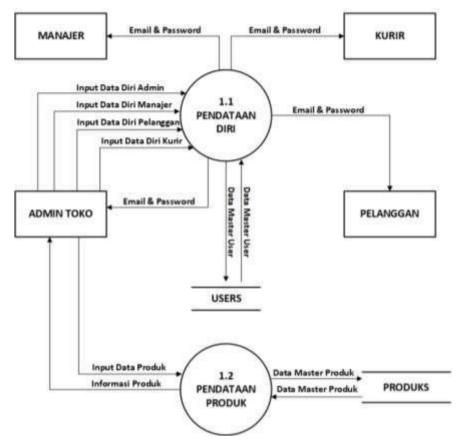
Berdasarkan Gambar 3.4 diagram konteks dipecah menjadi diagram level 0. Terdapat 3 proses utama pada aplikasi yaitu proses 1 pendataan, proses 2 pemesanan, dan proses 3 pelaporan. Pada proses 1 pendataan admin toko melakukan input data ke user yang menggunakan aplikasi dan melakukan input data produk. Kemudian data tersebut akan tersimpan pada penyimpanan users dan produks. Pada proses 2 pemesanan entitas yang terlibat adalah pelanggan dan kurir. Pelanggan akan mendapatkan e-mail dan password untuk login, kemudian setelah *login* pelanggan akan mendapatkan informasi produk, kemudian pelanggan melakukan *input* pesanan dan akan mendapatkan total belanja. Setelah pesanan berhasil masuk ke dalam aplikasi, pelanggan menunggu konfirmasi kurir agar pesanannya diproses dan diantar ke alamat yang di *inputkan* saat melakukan pemesanan. Kemudian kurir akan menerima notifikasi pesanan baru dan akan melihat detail pesanan. Kurir dapat mengelola pesanan pelanggan dengan mengkonfirmasi status pesanan. Status pesanan pada aplikasi pelanggan akan berubah sesuai konfirmasi dari kurir. Apabila kurir sudah mengkonfirmasi status pesanan menjadi "dalam pengiriman" maka pada aplikasi pelanggan dapat melacak posisi kurir, dan kurir dapat menggunakan fitur maps untuk mengantar pesanan sampai ke tempat tujuan. Kemudian pada proses 3 pelaporan entitas yang terlibat adalah manajer dan admin toko. Setelah mendapatkan e-mail dan password yang digunakan untuk login, manajer dapat login ke dalam web untuk mengakses laporan transaksi delivery order yang terjadi pada aplikasi baik pesanan yang sedang diproses maupun yang sudah diterima. Manajer juga dapat mengakses laporan transaksi harian dan laporan transaksi bulanan. Kemudian admin toko dapat mengakses laporan transaksi pada aplikasi dan dapat menghapus data transaksi yang sudah tidak dibutuhkan lagi.



Gambar 3. 4 Diagram level 0 aplikasi delivery order minimarket

c) Diagram level 1 Proses Pendataan

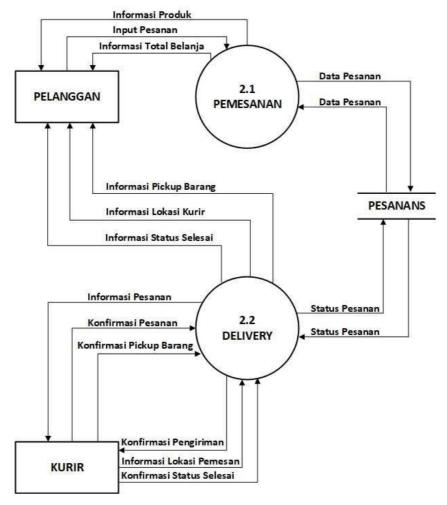
Diagram level 1 merupakan dekomposisi dari diagram level 0. DFD level 1 ini menjabarkan lebih rinci dari DFD level 0. Setiap proses pada DFD level 0 dapat dimodelkan secara lebih terperinci menjadi sebuah DFD lagi. Gambar 3.5 merupakan DFD level 1 proses 1 menjabarkan proses awal yaitu Pendataan masing-masing *user* yang ada pada aplikasi dan pendataan produk. Diagram 1 proses pendataan dapat dilihat pada Gambar 3.5.



Gambar 3. 5 Diagram level 1 proses pendataan

Gambar 3.5 merupakan gambar dari Diagram 1 proses pendataan yang terdiri dari 2 proses yaitu pendataan diri dan pendataan produk. Entitas admin toko akan melakukan pendataan semua *user* yang menggunakan aplikasi *delivery order* minimarket yaitu admintoko, manajer, kurir, dan pelanggan yang akan tersimpan pada tabel users, kemudian entitas yang lain seperti manajer, pelanggan, dan kurir hanya akan mendapatkan *e-mail* dan *password* untuk melakukan login pada aplikasi. Kemudian proses yang kedua yaitu pendataan produk yang dilakukan oleh admin toko dan akan tersimpan pada tabel produks. d) Diagram level 1 Proses Pemesanan

Diagram 1 proses pemesanan menjelaskan tentang proses pemesanan oleh pelanggan dan *delivery* yang akan dilakukan oleh kurir. Diagram level 1 proses pemesanan dapat dilihat pada Gambar 3.6.



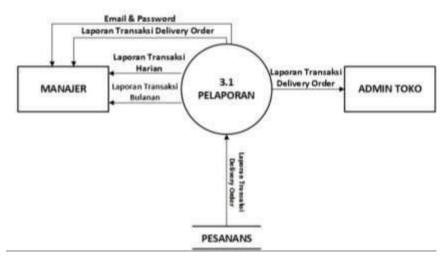
Gambar 3. 6 Diagram level 1 Proses Pemesanan

Gambar 3.6 adalah gambar diagram proses pemesanan. Terdapat 2 proses yaitu pemesanan, dan *delivery*. Entitas pelanggan akan melakukan pemesanan pada aplikasi android kemudian setelah mendapatkan total belanja pelanggan mengkonfirmasi untuk melakukan proses pemesanan selanjutnya dan pembayaran dilakukan secara *cash* ketika pesanan sudah sampai. Kemudian setelah itu akan masuk ke proses *delivery* yang dilakukan oleh kurir ke tempat pelanggan memesan. Pada proses ini pelanggan dapat memonitor status pesanan, pada saat status pesanan "dalam pengiriman", pelanggan dapat melacak posisi kurir. Entitas kurir akan menerima detail pesanan berupa notifikasi kemudian kurir akan mengkonfirmasi status pesanan pelanggan. Pada saat status pesanan "dalam pengiriman", kurir dapat menggunakan fitur *maps* untuk mengantar pesanan pelanggan ke alamat tujuan,

kemudian mengkonfirmasi status pesanan menjadi telah diterima setelah pelanggan melakukan pembayaran secara *cash*.

e) Diagram level 1 Proses Pelaporan

Diagram level 1 proses pelaporan akan menjelaskan proses pelaporan yang dapat diakses oleh manajer dan admin toko. Diagram 1 proses pelaporan dapat dilihat pada Gambar 3.7.

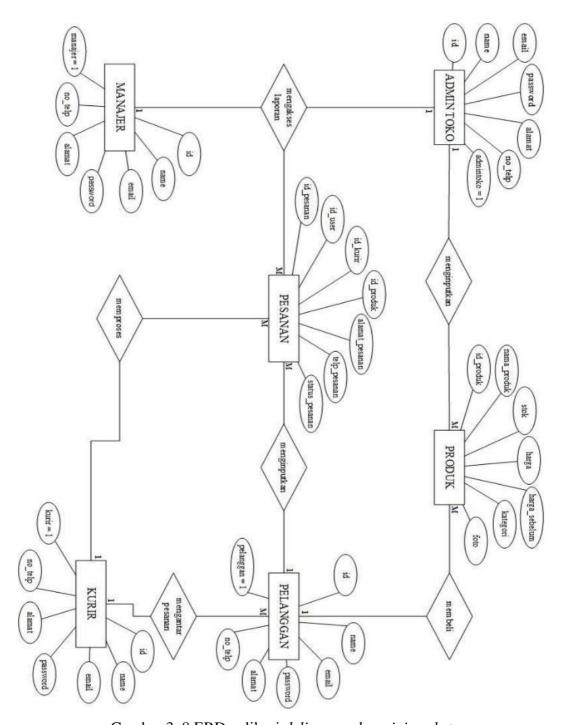


Gambar 3. 7 Diagram level 1 Proses Pelaporan

Pada Gambar 3.7 entitas yang terlibat adalah manajer dan admin toko yang dapat mengakses laporan transaksi *delivery order* minimarket. Setelah mendapat *e-mail* dan *password*, manajer melakukan *login* ke dalam web. Kemudian manajer dapat mengakses data transaksi *delivery order* minimarket secara keseluruhan, laporan transaksi harian, dan laporan transaksi bulanan. Entitas admintoko dapat mengakses laporan transaksi *delivery order* secara keseluruhan kemudian admintoko dapat menghapus data transaksi yang sudah tidak dibutuhkan lagi.

3.2.2.2 Entitiy Relationship Diagram (ERD)

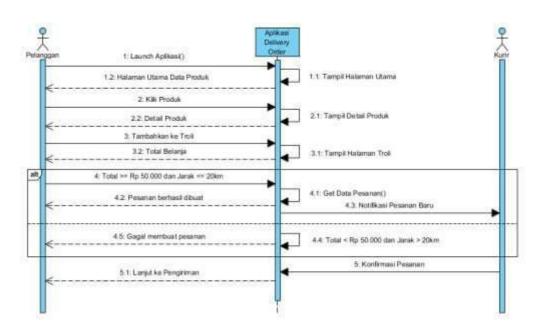
ERD menggambarkan relasi masing masing entitas dalam aplikasi *delivery order* minimarket berbasis web dan android dan digunakan untuk memodelkan kebutuhan data dari dalam aplikasi *delivery order* minimarket berbasis web dan android. Rancangan ERD dari aplikasi *delivery order* minimarket ditunjukkan pada gambar 3.8.



Gambar 3. 8 ERD aplikasi delivery order minimarket

3.2.2.3 Sequence Diagram

Sequence diagram digunakan untuk menggambarkan interaksi antar objek di dalam dan di sekitar aplikasi delivery order minimarket berbasis web dan android. Sequence Diagram biasa digunakan untuk memperlihatkan interaksi antar obyek dalam perintah yang berurut dan mendefinisikan urutan kejadian yang dapat menghasilkan output yang diinginkan. Berikut merupakan sequence diagram pemesanan dari aplikasi delivery order minimarket berbasis web dan android ditunjukkan pada gambar 3.9.



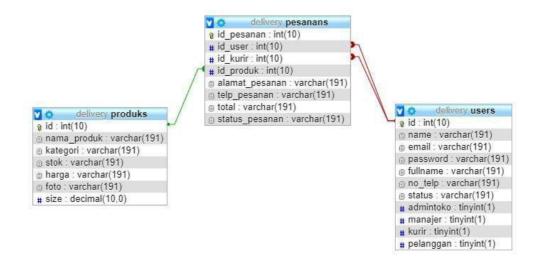
Gambar 3. 9 Sequence diagram delivery order minimarket

3.2.3 Perancangan Database

Perancangan *Database* merupakan proses penyusunan atribut dari masing masing entitas beserta tipe datanya, serta menentukan relasi dari masing masing entitas dalam bentuk relasi antar tabel.

3.2.3.1 Relasi Tabels

Relasi basis data menunjukkan suatu mekanisme yang digunakan untuk mengelola atau mengorganisasi data secara fisik. Adapun keterkaitan antar tabel yang digunakan dalam perancangan sistem dapat dilihat pada Gambar 3.10.



Gambar 3. 10 Relasi Tabel Delivery order Minimarket

Pada Gambar 3.10. tabel pesanans berelasi dengan tabel-tabel induk yang akan diambil datanya. *Field* id pada tabel produks berelasi dengan *field* id_produk pada tabel pesanans, kemudian *field* id pada tabel *users* berelasi dengan *field* id_user pada tabel pesanans. Relasi ini berfungsi pada saat pengambilan data yang akan digunakan pada transaksi *delivery order*.

3.2.3.2 Struktur Tabel Basis Data

Struktur tabel basis data memuat tabel-tabel yang dibuat untuk menyimpan data. Berikut merupakan tabel yang digunakan untuk membuat aplikasi *delivery order* minimarket.

1) Users

Tabel 3.1 merupakan tabel *users* yang digunakan untuk menyimpan semua datadata user atau pengguna aplikasi.

Tabel 3. 1 Tabel *Users*

Field	Tipe	Panjang
#Id	Int	10 (Auto Increment)
Name	varchar	191
Email	varchar	191
Password	varchar	191
Fullname	varchar	191
Alamat	varchar	191

no_telp	varchar	191
Status	varchar	191
admintoko	Tinyint	1
manajer	Tinyint	1
Kurir	Tinyint	1
Pelanggan	Tinyint	1

Semua *user* yang telah didaftarkan pada tabel *users* dapat melakukan login pada aplikasi, hak akses akan dibedakan dengan menggunakan field admintoko, manajer, pelanggan, kurir. Apabila admintoko bernilai satu sedangkan manajer, kurir, dan pelanggan bernilai 0 maka *user* tersebut memiliki hak akses login sebagai admintoko begitupula sebaliknya.

2) Produks

Tabel 3.2 merupakan tabel produks yang digunakan untuk menyimpan data-data produk seperti nama produk, harga, stok, kategori, dan foto yang nantinya akan ditampilkan pada aplikasi android untuk di *order* oleh pelanggan. Tabel 3. 2 Tabel Produks

Field	Tipe	Panjang
#Id	Int	10 (Auto Increment)
nama_produk	Varchar	191
kategori	Varchar	191
Stok	Varchar	191
harga_sebelum	Varchar	191
Harga	Varchar	191
Foto	Varchar	191
Size	Decimal	10,0

Pada tabel 3.2 terdapat *field* harga_sebelum. *Field* tersebut akan digunakan ketika admin menginputkan produk dengan kategori "Promo" maka *field* harga sebelum diisi harga produk awal sebelum promo dan pada *field* harga diisi harga setelah promo.

3) Pesanans

Tabel 3.3 merupakan tabel pesanans yang akan digunakan untuk menyimpan datadata transaksi dan data-data pada tabel tersebut akan digunakan sebagai laporan transaksi *delivery order*.

Tabel 3. 3 Tabel Pesanans

Field	Tipe	Panjang
#id_pesanan	int	10 (Auto Increment)
id_user*	varchar	191
id_kurir*	varchar	191
id_produk*	varchar	191
alamat_pesanan	varchar	191
telp_pesanan	varchar	191
Total	varchar	191
status_pesanan	varchar	191

3.2.4 Perancangan Antarmuka Sistem

Pada bagian ini akan dijelaskan tentang perancangan tampilan dari aplikasi android yang berjalan pada *smartphone* dan *website* berbasis web.

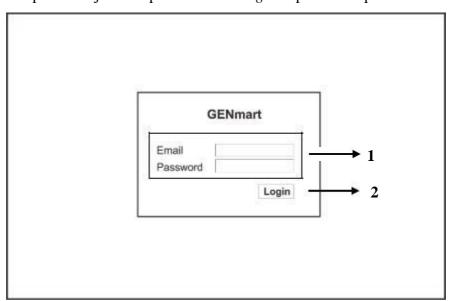
3.2.4.1 Perancangan Antarmuka Website

Pada bagian ini akan dijelaskan rancangan antarmuka yang tampil pada website.

Diantaranya adalah sebagai berikut :

1) Halaman login

Halaman ini merupakan halaman *login* utama untuk masuk ke dalam aplikasi web admin maupun manajer. Tampilan halaman *login* dapat dilihat pada Gambar 3.11.

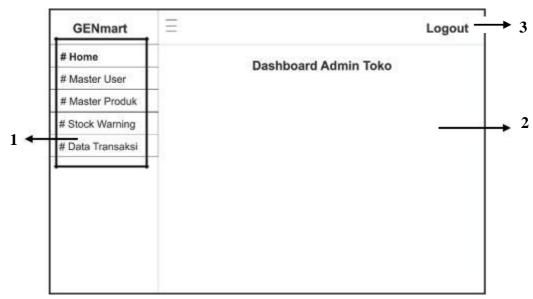


Gambar 3. 11 Halaman *login*

Gambar 3.11 adalah tampilan halaman *login*. Halaman ini akan muncul apabila membuka url http://antargen.informatikapolines.com . Bagian 1 merupakan *textbox* untuk mengisi *email* dan *password login*. Bagian 2 merupakan *button login* untuk masuk ke halaman web. Tampilan web akan dibedakan berdasarkan hak akses *user* saat *login*.

2) Dashboard admin toko

Halaman *dashboard* admin toko akan tampil ketika *user* telah *login* sebagai admin toko. Tampilan halaman *dashboard* admin toko dapat dilihat pada Gambar 3.12.

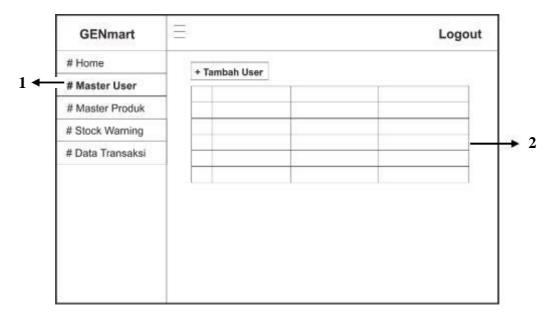


Gambar 3. 12 Dashboard admin toko

Gambar 3.12 merupakan *dashboard* admin toko. Bagian 1 merupakan menu-menu yang dapat dikelola di halaman web admin toko. Bagian 2 merupakan *dashboard* selamat datang ketika berhasil login. Dan bagian 3 merupakan *button logout* untuk keluar dari aplikasi web admin toko.

3) Halaman master *user*

Halaman master *user* adalah halaman yang menampilkan data-data *user* yang dapat menggunakan aplikasi *delivery order* minimarket. Tampilan halaman master *user* dapat dilihat pada Gambar 3.13.

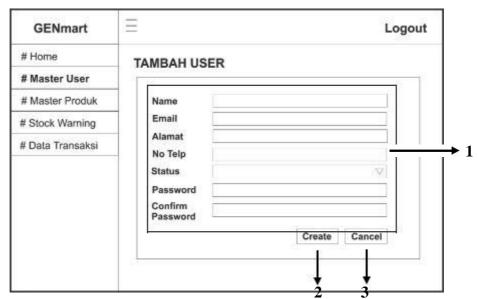


Gambar 3. 13 Halaman master user

Gambar 3.13 merupakan tampilan halaman master *user*. Halaman ini akan muncul apabila *user* admin toko mengklik menu Master *User*. Bagian 1 merupakan menu master *user* yang diklik. Bagian 2 merupakan master data *user* yang tampil. Bagian 3 merupakan *button* tambah *user* untuk menambahkan *user* / pengguna baru.

4) Halaman tambah user

Halaman tambah *user* merupakan halaman untuk menambah *user* baru. Halaman ini akan tampil ketika *user* mengklik *button* tambah *user* pada halaman master *user*. Tampilan halaman tambah *user* dapat dilihat pada Gambar 3.14.

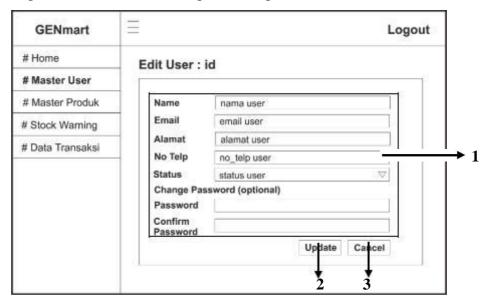


Gambar 3. 14 Halaman tambah user

Gambar 3.14 merupakan tampilan halaman tambah *user*. Bagian 1 merupakan *form* pengisian data *user*. Bagian 2 merupakan *button create* yang akan menyimpan data-data *user* ke *database*. Bagian 3 merupakan *button cancel* apabila akan membatalkan membatalkan membatalkan button cancel

5) Halaman edit *user*

Halaman edit *user* merupakan halaman yang digunakan untuk mengedit data-data *user*. Tampilan halaman edit *user* dapat dilihat pada Gambar 3.15.

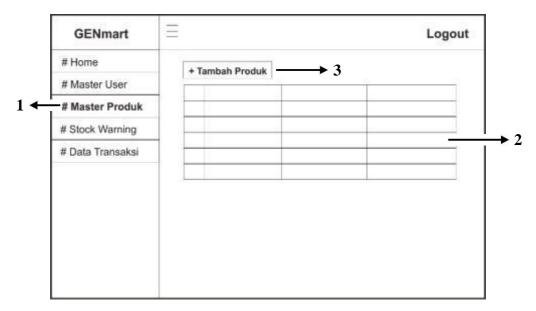


Gambar 3. 15 Halaman edit user

Gambar 3.15 merupakan tampilan halaman untuk mengedit data *user*. Bagian 1 merupakan *form update* data *user* yang sudah berisi data-data sebelumnya. Bagian 2 merupakan *button update* yang akan mengganti data *user* di *database*. Bagian 3 merupakan *button cancel* untuk membatalkan edit data *user*.

6) Halaman master produk

Halaman master produk merupakan halaman yang menampilkan master data produk yang dijual di minimarket terkait. Produk-produk yang ada di halaman master produk akan tampil pada aplikasi android *delivery order*. Tampilan halaman master produk dapat dilihat pada Gambar 3.16.

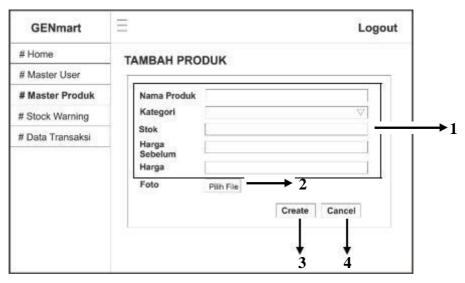


Gambar 3. 16 Halaman master produk

Gambar 3.16 merupakan tampilan halaman master produk. Halaman ini akan muncul apabila *user* admin toko mengklik menu Master Produk. Bagian 1 merupakan menu master produk yang diklik. Bagian 2 merupakan master data produk yang tampil. Bagian 3 merupakan *button* tambah produk untuk menambahkan produk baru.

7) Halaman tambah produk

Halaman tambah produk merupakan halaman yang digunakan untuk menambahkan data produk baru. Tampilan halaman tambah produk dapat dilihat pada Gambar 3.17.



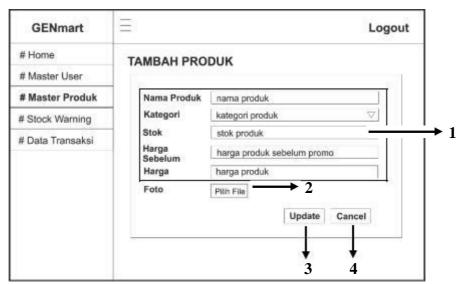
Gambar 3. 17 Halaman tambah produk

Gambar 3.17 merupakan tampilan halaman tambah produk. Bagian 1 merupakan *form* untuk mengisi data produk baru. Bagian 2 adalah *button* untuk menambahkan foto produk. Bagian 3 merupakan *button create* yang akan menambahkan data produk ke database. Dan bagian 4 merupakan *button cancel* untuk membatalkan tambah produk.

8) Halaman edit produk

Halaman edit produk merupakan halaman yang digunakan untuk mengedit data produk. Pada halaman ini sudah tampil data-data produk

yang akan diedit. Tampilan halaman edit produk dapat dilihat pada Gambar 3.18.



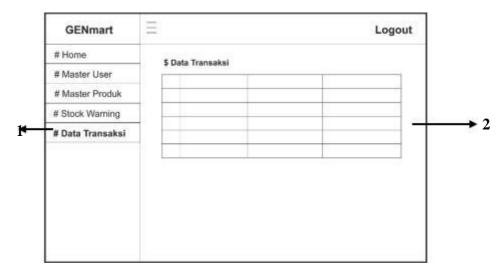
Gambar 3. 18 Halaman edit produk

Gambar 3.18 merupakan tampilan halaman untuk mengedit data produk. Bagian 1 merupakan form untuk mengedit data-data produk sebelumnya. Bagian 2 merupakan *button* untuk mengganti foto produk. Bagian 3 merupakan *button update* untuk mengganti data produk di *database*. Dan bagian 4 merupakan *button cancel* untuk membatalkan edit produk

9) Halaman data transaksi

Halaman data transaksi merupakan halaman yang menampilkan data-data transaksi yang terjadi pada aplikasi *delivery order* minimarket. Baik transaksi yang sedang berjalan maupun yang sudah selesai. Data ini tampil pada halaman web admin toko agar admin toko dapat mengelola data transaksi apabila sewaktu-waktu ada

transaksi yang akan dihapus. Tampilan halaman data transaksi dapat dilihat pada Gambar 3.19.

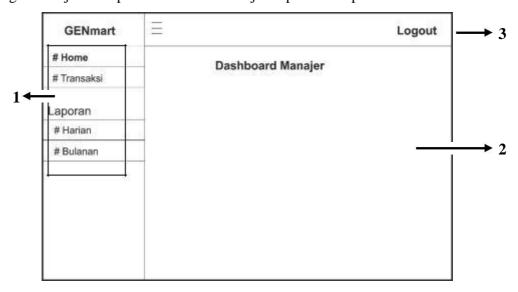


Gambar 3. 19 Halaman data transaksi

Gambar 3.19 merupakan tampilan halaman data transaksi. Bagian 1 merupakan menu Data Transaksi yang diklik oleh admin toko sehingga menampilkan halaman tersebut. Bagian 2 merupakan data-data transaksi yang telah terjadi pada aplikasi *delivery order* minimarket, baik transaksi yang sedang berlangsung maupun transaksi yang telah selesai.

10) Dashboard manajer

Halaman *dashboard* manajer merupakan halaman yang akan tampil jika *user login* sebagai manajer. Tampilan dashboard manajer dapat dilihat pada Gambar 3.20.

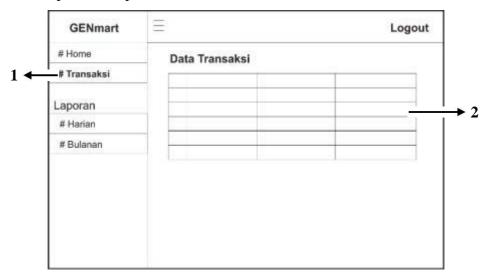


Gambar 3. 20 Dashboard manajer

Gambar 3.20 merupakan halaman yang tampil pada saat *user login* sebagai manajer. Bagian 1 merupakan menu-menu yang dapat dikelola oleh manajer. Bagian 2 merupakan *dashboard* yang berisi selamat datang ketika manajer berhasil login. Dan bagian 3 merupakan *button logout* untuk keluar dari halaman web manajer.

11) Halaman laporan transaksi

Halaman laporan transaksi merupakan halaman yang menampilkan data-data transaksi yang terjadi pada aplikasi *delivery order*. Data-data ini digunakan sebagai laporan harian dan bulanan kepada manajer. Tampilan halaman laporan transaksi dapat dilihat pada Gambar 3.21.

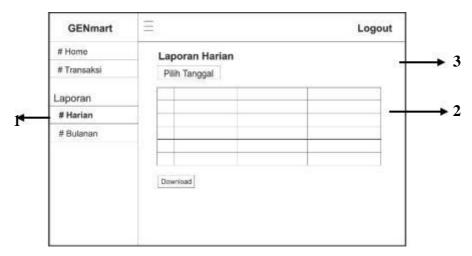


Gambar 3. 21 Halaman laporan transaksi

Gambar 3.21 merupakan tampilan laporan transaksi yang akan tampil ketika manajer mengklik menu transaksi. Bagian 1 merupakan menu transaksi yang diklik. Bagian 2 merupakan *field* data-data transaksi yang terjadi pada aplikasi *delivery order* minimarket. Baik yang belum diproses maupun yang sudah selesai.

12) Halaman laporan transaksi harian

Halaman laporan transaksi harian merupakan halaman yang menampilkan datadata transaksi *delivery order* perhari. *User* manajer dapat memilih tanggal yang akan dicek laporannya. Kemudian manajer dapat me*download* laporan harian tersebut dalam bentuk dokumen pdf. Tampilan halaman laporan transaski harian dapat dilihat pada Gambar 3.22.

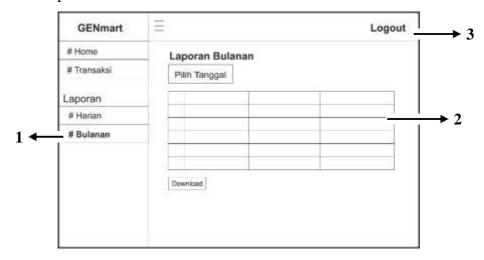


Gambar 3. 22 Halaman laporan transaksi harian

Gambar 3.22 merupakan tampilan transaksi *delivery oder* harian yang dapat dipantau oleh manajer. Bagian 1 merupakan menu harian yang diklik oleh manajer. Bagian 2 merupakan data-data transaksi harian yang tampil pada hari saat manajer mengakses. Dan bagian 3 merupakan *button* untuk me*download* laporan harian dalam bentuk dokumen pdf.

13) Halaman laporan transaksi bulanan

Halaman laporan transaksi bulanan merupakan halaman yang menampilkan datadata transaksi *delivery order* perbulan. *User* manajer dapat memilih tanggal yang akan dicek laporannya. Kemudian manajer dapat me*download* laporan bulanan tersebut dalam bentuk dokumen pdf. Tampilan halaman laporan transaski bulanan dapat dilihat pada Gambar 3.23.

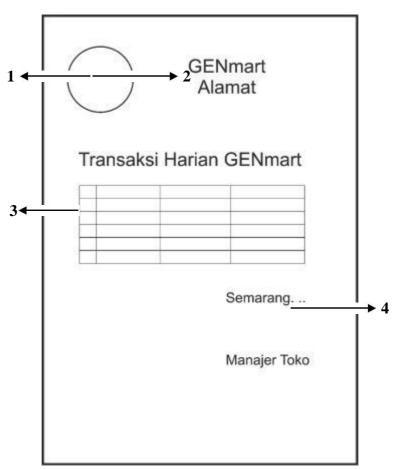


Gambar 3. 23 Halaman laporan transaksi bulanan

Gambar 3.23 merupakan tampilan transaksi *delivery oder* bulanan yang dapat dipantau oleh manajer. Bagian 1 merupakan menu bulanan yang diklik oleh manajer. Bagian 2 merupakan data-data transaksi bulanan yang tampil pada bulan saat manajer mengakses. Dan bagian 3 merupakan *button* untuk me*download* laporan bulanan dalam bentuk dokumen pdf.

14) Halaman harian pdf

Halaman harian pdf merupakan laporan harian yang sudah berhasil diunduh dan tampil dalam bentuk pdf. Tampilan halaman harian pdf dapat dilihat pada Gambar 3.24.

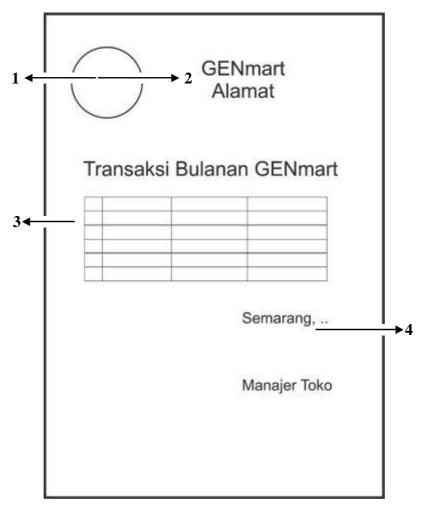


Gambar 3. 24 Halaman laporan harian pdf

Gambar 3.24 terdiri dari beberapa bagian. Bagian 1 merupakan logo minimarket, bagian 2 merupakan *header* laporan yang berisi motto dan alamat minimarket, bagian 3 merupakan laporan transaksi harian yang dipilih, dan bagian 4 merupakan pengesahan laporan.

15) Halaman bulanan pdf

Halaman bulanan pdf merupakan laporan bulanan yang sudah berhasil diunduh dan tampil dalam bentuk pdf. Tampilan halaman bulanan pdf dapat dilihat pada Gambar 3.25.



Gambar 3. 25 Halaman bulanan pdf

Gambar 3.25 terdiri dari beberapa bagian. Bagian 1 merupakan logo minimarket, bagian 2 merupakan *header* laporan yang berisi motto dan alamat *minimarket*, bagian 3 merupakan laporan transaksi bulanan yang dipilih, dan bagian 4 merupakan pengesahan laporan.

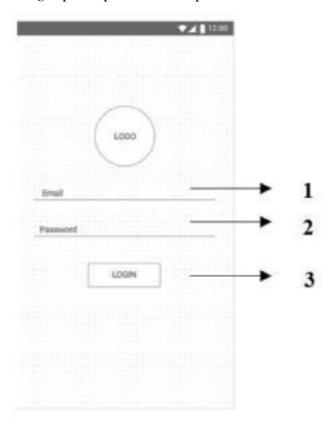
3.2.4.2 Perancangan Antarmuka Android

Pada bagian ini akan dijelaskan tentang perancangan tampilan dari aplikasi android yang berjalan pada *smartphone* android untuk bagian perbelanjaan dan kurir.

a) Desain Aplikasi Pelanggan

1) Halaman *login*

Halaman *Login* digunakan saat pengguna *login* ke dalam aplikasi. Pengguna harus mengisi *e-mail* dan *password* yang sesuai untuk masuk ke dalam aplikasi. Tampilan halaman *login* pada aplikasi dilihat pada Gambar 3.26.



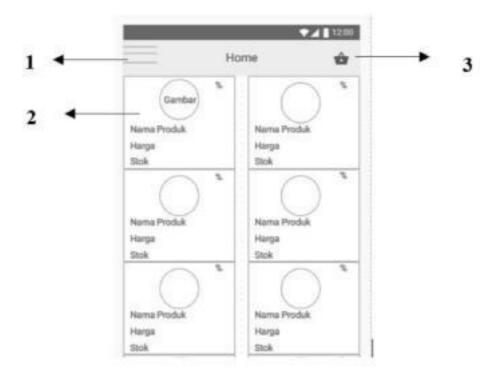
Gambar 3. 26 Halaman *login* pelanggan

Gambar 3.26 adalah tampilan halaman *login*. Pada halaman *login* terdapat 3 bagian diantaranya yaitu bagian 1 merupakan form *e-mail* yang digunakan untuk memasukkan *e-mail* pengguna/*user* yang telah terdaftar di admin, bagian 2 merupakan form *password* digunakan untuk memasukkan *password* pengguna serta bagian 3 merupakan tombol *login* yang berfungsi untuk masuk ke dalam sistem apabila *e-mail* dan *password* benar.

2) Halaman home

Halaman *home* merupakan halaman yang akan muncul jika pengguna berhasil *login* ke dalam aplikasi. Pada halaman ini terdapat *navigasi bar* yang berisi menu pilihan

yang dapat diakses oleh pengguna. Tampilan halaman *home* pada aplikasi dapat dilihat pada Gambar 3.27.

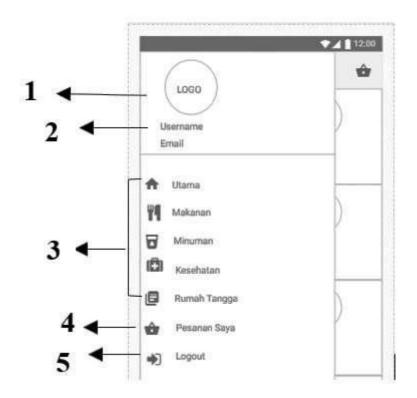


Gambar 3. 27 Halaman home

Gambar 3.27 merupakan tampilan yang akan muncul ketika *login* berhasil. Dalam halaman *home*, terdapat 3 bagian yaitu *navigasi bar* yang ditunjukkan pada bagian 1, *navigasi bar* berfungsi untuk menampilkan berbagai menu yang tersedia dalam aplikasi. Bagian 2 berisi produk yang sedang promo. Bagian 3 merupakan troli yang berisi produk yang sudah dipilih ketika berbelanja.

3) Navigasi bar

Navigasi bar berisi beberapa menu yang tersedia dalam aplikasi. Menu tersebut berupa halaman – halaman yang sudah terkategori pada aplikasi. Tampilan navigasi bar pada aplikasi dapat dilihat pada Gambar 3.28.



Gambar 3. 28 navigasi Bar

Gambar 3.28 merupakan tampilan pada *navigasi bar*. Dalam *navigasi bar*, terdapat 4 bagian yaitu gambar logo yang ditunjukkan pada bagian 1. Bagian 2 berisi *username* dan *e-mail* dari pelanggan setelah berhasil *login*. Bagian 3 merupakan beberapa menu yang ada di dalam aplikasi yang sudah terkategori. Bagian 4 berisi *history* dan status dari produk yang sudah selesai melakukan transaksi berbelanja. Bagian 5 merupakan menu *logout* yang digunakan untuk keluar dari aplikasi.

4) Halaman makanan

Halaman makanan merupakan halaman yang menampilkan produk – produk dalam kategori makanan yang tersimpan dalam *database*, pada halaman makanan terdapat nama produk, harga produk, dan jumlah stok produk. Tampilan halaman makanan pada aplikasi dapat dilihat pada Gambar 3.29.



Gambar 3. 29 Halaman makanan

5) Halaman minuman

Halaman minuman merupakan halaman yang menampilkan produk – produk dalam kategori minuman yang tersimpan dalam *database*, pada halaman minuman terdapat nama produk, harga produk, dan jumlah stok produk. Tampilan halaman minuman pada aplikasi dapat dilihat pada Gambar 3.30.



Gambar 3. 30 Halaman minuman

6) Halaman kesehatan

Halaman kesehatan merupakan halaman yang menampilkan produk – produk dalam kategori kesehatan yang tersimpan dalam *database*, pada halaman kesehatan

terdapat nama produk, harga produk, dan jumlah stok produk. Tampilan halaman kesehatan pada aplikasi dapat dilihat pada Gambar 3.31



Gambar 3. 31 Halaman kesehatan

7) Halaman rumah tangga

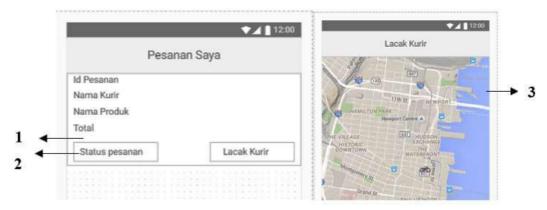
Halaman rumah tangga merupakan halaman yang menampilkan produk — produk dalam kategori rumah tangga yang tersimpan dalam *database*, pada halaman rumah tangga terdapat nama produk, harga produk, dan jumlah stok produk. Tampilan halaman rumah tangga pada aplikasi dapat dilihat pada Gambar 3.32.



Gambar 3. 32 Halaman rumah tangga

8) Halaman pesanan saya

Halamaan pesanan saya merupakan halaman yang berisi produk setelah melakukan proses pemesanan dan berisi status pesanan produk, dan tombol lacak kurir. Tampilan halaman pesanan saya pada aplikasi dapat dilihat pada Gambar 3.33.

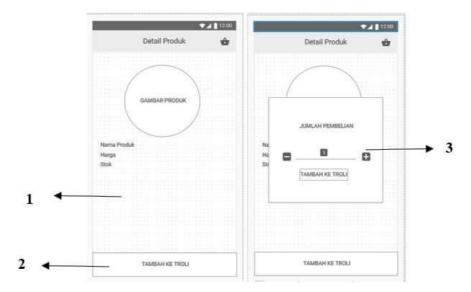


Gambar 3. 33 Halaman pesanan saya

Gambar nomor 1 pada Gambar 3.33 merupakan rancangan antar muka untuk menampilkan data *history* pesanan. Gambar nomor 2 merupakan status pesanan, dari situlah pengguna bisa melihat status pesanan dari produk tersebut, jadi nantinya ada empat jenis status pesanan pertama "sedang diproses" artinya produk tersebut sedang di proses menuju aplikasi dari kurir, yang keduas "pick up barang" artinya produk tersebut sedang dikemas oleh kurir, yang ketiga "dalam pengiriman" artinya produk tersebut telah diantar menuju rumah pelanggan, dan yang terakhir "telah diterima" yang berarti produk tersebut sudah diterima oleh pelanggan. Gambar nomor 3 merupakan rancangan antar muka untuk halaman lacak kurir, jadi ketika tombol lacak kurir di klik maka pelanggan bisa melihat posisi kurir.

9) Halaman detail produk

Halaman detail produk merupakan halaman yang berisi keterangan dari produk setelah anda mengklik produk yang anda beli. Gambar 3.34 merupakan tampilan dari halaman detail produk.



Gambar 3. 34 Halaman detail produk

Gambar nomor 1 pada gambar 3.34 terdapat informasi mengenai produk dan gambar produk. Gambar nomor 2 merupakan tombol tambah ke troli yang nantinya jika pelanggan berminat untuk membeli produk tersebut, klik saja tombol tambah troli. Gambar nomor 3 merupakan masukan jumlah produk yang akan dibeli oleh pelanggan, setelah pelanggan memasukkan jumlah produk maka produk akan masuk di dalam troli.

10) Halaman troli saya

Halaman troli merupakan halaman yang berisi keterangan jumlah total yang harus dibayar dari perbelanjaan pelanggan. Gambar 3.35 merupakan tampilan dari halaman troli.



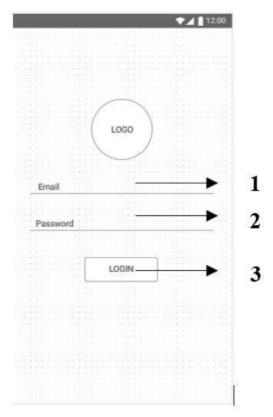
Gambar 3. 35 Halaman troli saya

Pada Gambar 3.35 pada nomor 1 merupakan keterangan dari biaya perbelanjaan pelanggan. Sedangkan pada nomor 2 merupakan *icon* tempat sampah yang artinya jika pelanggan salah membeli produk, pelanggan bisa mengklik *icon* tersebut. Pada nomor 3 terdapat tombol lanjutkan ke pengiriman yang artinya jika produk tersebut sudah benar-benar dibeli nantinya akan di proses ke pengiriman. Pada nomor 4 merupakan form pengiriman setelah pelanggan mengklik tombol lanjutkan ke pengiriman, pelanggan akan mengisi form pengiriman tersebut. Di form pengiriman pelanggan bisa memilih kurir, memasukkan alamat pengiriman, dan memasukkan nomor telefon setelah itu klik tombol kirim untuk melanjutkan proses pengiriman yang nantinya produk yang anda beli akan masuk di notifikasi dari aplikasi kurir.

b) Desain Aplikasi Kurir

1) Halaman login

Halaman *login* digunakan saat kurir *login* ke dalam aplikasi. Kurir harus mengisi *e-mail* dan *password* yang sesuai untuk masuk ke dalam aplikasi. Tampilan halaman *login* pada aplikasi dilihat pada Gambar 3.36.



Gambar 3. 36 Halaman login kurir

Gambar 3.36 adalah tampilan halaman *login*. Pada halaman *login* terdapat 3 bagian diantaranya yaitu bagian 1 merupakan form *e-mail* yang digunakan untuk memasukkan *e-mail* kurir yang telah terdaftar di admin, bagian 2 merupakan form *password* digunakan untuk memasukkan *password* pengguna serta bagian 3 merupakan tombol *login* yang berfungsi untuk masuk ke dalam sistem apabila *e-mail* dan *password* benar.

2) Halaman pengiriman

Halaman pengiriman merupakan halaman yang akan muncul jika pengguna berhasil *login* ke dalam aplikasi. Pada halaman ini terdapat *navigasi bar* yang berisi menu pilihan yang dapat diakses oleh kurir, status pesanan dan *maps*. Tampilan halaman pengiriman pada aplikasi dapat dilihat pada Gambar 3.37.



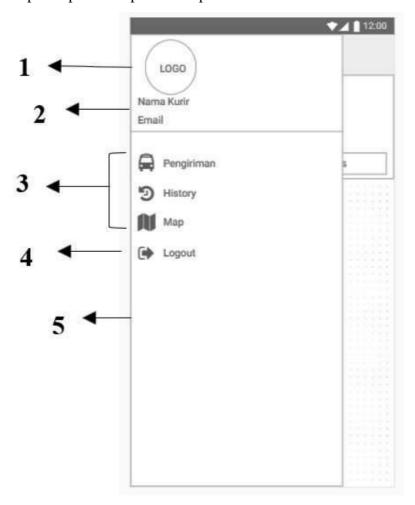
Gambar 3. 37 Halaman Pengiriman

Gambar 3.37 merupakan tampilan yang akan muncul ketika *login* berhasil. Dalam halaman pengiriman, terdapat 4 bagian yaitu *navigasi bar* yang ditunjukkan pada nomor 1, *navigasi bar* berfungsi untuk menampilkan berbagai menu yang tersedia dalam aplikasi kurir. Nomor 2 berisi detail produk pembeli yang akan diantar. Nomor 3 merupakan tombol status pesanan, kurir akan memencet tombol tersebut sesuai kondisi yang ada. Di dalam status pesanan ada empat jenis status pesanan pertama "sedang diproses" artinya produk tersebut sedang di proses menuju aplikasi dari kurir, yang kedua "*pick up* barang" artinya produk tersebut sudah dikemas oleh kurir, yang ketiga "dalam pengiriman" artinya produk tersebut telah diantar menuju rumah pelanggan, dan yang terakhir "telah diterima" yang berarti

produk tersebut sudah diterima oleh pelanggan. Gambar nomor 4 merupakan tombol *maps*, jika dipencet nantinya map tersebut akan melakukan tarcking menuju kerumah pelanggan.

3) Navigasi bar

Navigasi bar berisi beberapa menu yang tersedia dalam aplikasi. Menu tersebut berupa halaman – halaman yang sudah terkategori pada aplikasi. Tampilan navigasi bar pada aplikasi dapat dilihat pada Gambar 3.38.



Gambar 3. 38 Navigasi bar

Gambar 3.38 merupakan tampilan pada *navigasi bar*. Dalam *navigasi bar*, terdapat 4 bagian yaitu gambar logo yang ditunjukkan pada bagian 1. Bagian 2 berisi *username* dan *e-mail* dari kurir setelah berhasil *login*. Bagian 3 merupakan beberapa menu yang ada di dalam aplikasi yang sudah terkategori. Bagian 4 merupakan menu *logout* yang digunakan untuk keluar dari aplikasi kurir.

4) Halaman history

Didalam halaman *history* berisi produk-produk yang pernah diantar oleh kurir. Didalam halaman tersebut tampilannya sama pada halaman pengiriman tetapi terdapat perbedaan status pengiriman saja, di halaman *history* status pengirimannya berubah menjadi telah diterima karena produk tersebut tersebut sudah diterima oleh pelanggan.

5) Halaman maps

Pada halaman *maps* berisi *tracking rute* rumah pelanggan sehingga mempermudah kurir dalam mengantar pesanan. *Maps* ini akan masuk ke dalam *google maps*. Tampilan *maps* pada aplikasi dapat dilihat pada Gambar 3.39.



Gambar 3. 39 Halaman maps

Pada gambar 3.39 merupakan halaman *maps* dari aplikasi kurir yang berisi *tracking rute* menuju alamat pelanggan.

3.3 Penulisan Kode Program

Dibawah ini akan diperlihatkan penulisan kode program android dan website.

3.3.1 Penulisan Kode Program Website

Pada bagian ini akan ditunjukkan beberapa bagian koding website untuk mengelola halaman-halaman tertentu. Diantaranya sebagai berikut.

1) Halaman login

Halaman *login* merupakan halaman yang akan tampil saat user mengakses url http://antarger.informatikapolines.com . Halaman *login* digunakan untuk mesuk ke dalam sistem. Berikut kode program untuk *login* ke sistem :

```
<!-- /.login-logo -->
   <div class="login-box-body">
            {{
trans('adminite::adminite.login message') }}
            <form action="{{ url(config('adminlte.login url',</pre>
'login')) }}" method="post">
                {!! csrf field() !!}
                <div class="form-group has-feedback {{</pre>
$errors->has('email') ? 'has-error' : '' }}">
                    <input type="email" name="email"</pre>
class="form-control" value="{{ old('email') }}"
                           placeholder="{{
trans('adminlte::adminlte.email') }}">
                    <span class="glyphicon glyphicon-</pre>
envelope form-control-feedback"></span>
                    @if ($errors->has('email'))
                        <span class="help-block">
                             <strong>{{ $errors->first('email')}
}}</strong>
                        </span>
                    @endif
                </div>
                <div class="form-group has-feedback {{</pre>
$errors->has('password') ? 'has-error' : '' }}">
                    <input type="password" name="password"</pre>
class="form-control"
                           placeholder="{{
trans('adminlte::adminlte.password') }}">
                    <span class="glyphicon glyphicon-lock</pre>
form-control-feedback"></span>
                    @if ($errors->has('password'))
                        <span class="help-block">
                            <strong>{{ $errors-
>first('password') }}</strong>
                        </span>
                    @endif
                </div>
                <div class="row">
                    <div class="col-xs-8">
                        <div class="checkbox icheck">
```

```
<label>
                                 <input type="checkbox"</pre>
name="remember"> {{ trans('adminIte::adminIte.remember me') }}
                            </label>
                         </div>
                     </div>
                     <!-- /.col -->
                     <div class="col-xs-4">
                         <button type="submit"</pre>
                                 class="btn btn-primary btn-block
btn-flat">{{ trans('adminlte::adminlte.sign in')
                    }}</button> </div>
                     <!-- /.col -->
                 </div>
            </form>
        </div>
        <!-- /.login-box-body -->
    </div><!-- /.login-box -->
```

Pada bagian controller kode programnya sebagai berikut :

```
public function index()
{
    return view('home');
}
```

Kode program pada *controller* diatas berfungsi untuk menjalankan proses login dalam aplikasi. Yang mana setelah login akan diarahakan ke halaman *home* atau dashboard masing-masing hak akses pengguna.

Dan routing diatur pada file web.php. Kode programnya sebagai berikut :

```
Route::get('/', function () {
    return view('welcome');
});
Route::get('/home', 'HomeController@index');
```

Routing digunakan untuk mengalihkan ke halaman *login* ketika pengguna mengetikkan url antarggen.informatikapolines.com .

2) Halaman master data

Halaman master data akan menampilkan data-data master yang digunakan oleh aplikasi yaitu master data user dan master data produk. Beriku kode program untuk menampilkan master data :

```
No
              Nama
              Email
              Status
              Action
            </thead>
          <?php $no=1; ?>
          @foreach ($users as $user)
            { $no++ } } 
              {{$user-> name}}
              {{$user-> email}}
              @if ( $user->admintoko == 1)
                 Admin Toko
              @elseif ( $user->admintransaksi == 1)
                 Admin Transaksi
              @elseif ( $user->manajer == 1)
                 Manajer
              @elseif ( $user->kurir == 1)
                 Kurir
              @else
                 Pelanggan
              @endif
              class=""
              <form
                             action="{{ url('masteruser')
}}/{{ $user->id }}" method="post">
              <input type="hidden" name=" token" value="{{</pre>
csrf token() }}">
              {{method field('DELETE')}}
              <a href="{{ url('masteruser') }}/{{ $user->id}
}}/edit" class="btn btn-warning"><span class="fa fa-pencil"</pre>
aria-hidden="true"></span> Edit</a>
              <button class="btn btn-danger" onclick="var x = \frac{1}{2}
confirm('Yakin akan menghapus data ini ?');
              if(x){return true;} else {return false;}">
                       class="fa
                                     fa-trash"
              <span
                                                   aria-
hidden="true"></span> Hapus</button>
              </form>
              @endforeach
          </div>
```

Kode program diatas adalah sebagai *view* atau desain pada web yang akan menampilkan data-data dalam bentuk tabel disertai beberapa *button* untuk mengelola data tersebut seperti *button* edit dan *button* hapus.

Pada bagian controller kode programnya sebagai berikut :

```
return view('admintoko.masteruser', compact('users'));
}
```

Kode program pada *controller* diatas digunakan untuk mengatur setelah klik menu master data tersebut akan tampil halaman yang dituju.

Kemudian routing pada web.php diatur sebagai berikut :

```
Route::resource('/masteruser', 'MasterUserController');
```

Rouiting digunakan agar setelah menu diklik akan tertuju ke halaman yang benar.

3) Halaman data transaksi

Halaman data transaksi akan menampilkan data-data transaksi yang telah terjadi pada aplikasi. Kode programnya adalah sebagai berikut :

```
<div class="table-responsive">
      <table id="table id" class="table table-striped table-
hover">
         <thead>
              No
                        Pemesan
                        Kurir
                        Pesanan
                        Alamat
                        Telp Pemesan
                        Total
                        Status
                   </thead>
          @php
                   no=1;
              @endphp
         @foreach ($pesanans as $items)
              \langle t.r \rangle
                        {{ $no++ }}
                        {{ $items->namapemesan }}
                        {{ $items->namakurir }}
                        {{ $items->nama produk }}
                        { {
                                 $items->alamat pesanan
} 
                        { {
                                    $items->telp pesanan
} < < td>
                        {{ $items->total }}
                        @if($items->status pesanan == '3')
                        <span class="badge"
                                              badge-
success">Telah Diterima</span>
                        @elseif($items->status pesanan
'2')
```

```
<span
                                   class="badge
                                                badge-
primary">Dalam Pengiriman</span>
                        @elseif($items->status pesanan
'1')
                        <span class="badge badge-
warning">Pickup Barang</span>
                        @else
                        <span class="badge badge-
secondary">Sedang
                        Diprose
                         s</span
                         >
                         @endif
               @endforeach
```

Kode program diatas akan menampilkan data-data transaksi yang terjadi pada aplikasi baik transaksi yang sedang diproses maupun transaksi yang sudah selesai. Pada controller kode programnya adalah sebagai berikut :

```
public function pesanan()
        $pesanans = DB::table('pesanans')
            ->join('users', 'pesanans.id user', '=', 'users.id')
            ->join('users as kurir', 'pesanans.id kurir', '=',
'kurir.id')
            ->join('produks', 'pesanans.id produk', '=',
'produks.id')
            ->select('pesanans.*', 'users.name as namapemesan',
'kurir.name as namakurir', 'produks.nama_produk')
            ->get();
        $pesanans = \App\Pesanan::paginate(15);
        return view('manajer.pesanan', compact('pesanans'));
```

Function pesanans adalh fungsi yang digunakan untuk mengontrol data-data yang akan tampil pada halaman transaksi, data-data diambil dari beberapa tabel seperti tabel users untuk mengambil data pelanggan dan kurir, kemudian tabel produks untuk mengambil data produk yang ada dalam transaksi. Pengambilan data-data tersebut dilakukan dengan cara inner join.

Dan routing web.php kode programnya adalah sebagai berikut:

```
Route::get('/manajer', function ()
    return view('manajer.index');
});
   Route::get('manajer/pesanan', 'ManajerController@pesanan');
```

Routing tersebut diatas akan mengarahkan ke halaman data transaksi setelah menu diklik.

4) Laporan dalam dokumen pdf

Laporan transaksi akan dicetak dalam bentuk dokumen pdf. Laporan ini dapat diunduh oleh manajer yang nantinya akan disahkan oleh manajer dan disimpan sebagai arsip. Berikut kode program untuk menampilkan laporan.

```
<body style="font-size: 12px;">
     <img src="{{ url('logo.png') }}" alt="Logo" style="height:</pre>
80px; width: 150px; float: left; margin-top: -15px;">
     <center><h1 style="margin-right: 75px;">GENmart
<i>"Belanja Mudah dan Berkah"</i></hl></center>
          <h2>Jalan Dinar Asri
Raya B-1, Meteseh-Tembalang</h2>
      </center>
_____
          <?php
          $myDateTime = new DateTime($tql);
          $newDateString = $myDateTime->format('d M
          Y'); $tanggal = $myDateTime->format('d M Y');
      ?>
          <br><br><center>
          <b>LAPORAN TRANSAKSI PESANAN GENmart TANGGAL
{{strtoupper($newDateString)}}</b>
      </center>
          <br>
          <thead>
               \langle t.r \rangle
                 No
                         Pemesan
                         Kurir
                         Pesanan
                         Alamat
                         Telp Pemesan
                         Total
               </thead>
           @php
                 $no=1;
                 totalsemua = 0;
             @endphp
             @foreach ($laporan as $items)
             {{ $no++ }}
                         {{ $items->namapemesan }}
                         {{ $items->namakurir }}
                         {{ $items->nama produk }}
                                   $items->alamat pesanan
                         { {
} 
                         { {
                                     $items->telp pesanan
} < < td>
```

```
Rp.{{number format($items-
>total) } } 
              @php
                 $totalsemua += $items->total;
         @endphp
         @endforeach
         <b>Jumlah Total</b>
   <b>Rp.{{number format($totalsemua)}}</b></td
              > 
        <br><br><br>>
  Semarang, {{$tanggal}}<br>
  Manajer
GENmart,
  <br>
  <br>
  <br>
  <br>
  <br>
                               style="float:
  <p
right;">(.....)</p
</body>
```

Kode program diatas merupakan format laporan yang akan tampil pada dokumen pdf. Laporan tersebut berisi header dari *minimarket*, data-data transaksi, dan bagian pengesahan oleh manajer.

Kemudian pada *controller* menggunakan fungsi dibawah :

```
public function harianpdf($id pesanan)
    {
        $laporan = DB::table('pesanans')
            ->join('users', 'pesanans.id user', '=', 'users.id')
            ->join('users as kurir', 'pesanans.id_kurir', '=',
'kurir.id')
            ->join('produks', 'pesanans.id_produk', '=',
'produks.id')
            ->select('pesanans.*', 'users.name as namapemesan',
'kurir.name as namakurir', 'produks.nama_produk')
            ->whereRaw('pesanans.created at LIKE "%' .
$id_pesanan . '%"')
            ->get();
        $pdf = PDF::loadView('laporan.harian-pdf', ['laporan' =>
$laporan, 'tgl' => $id pesanan]);
       return $pdf->download('laporan-harian.pdf');
```

Fungsi diatas digunakan saat manajer memilih tanggal transaksi yang akan diunduh sehingga akan menampilkan transaksi di tanggal yang dipilih. Kemudian diroutingkan menggunakan kode program berikut:

```
Route::get('manajer/laporan-harian-pdf/{id_pesanan}',
'ManajerController@harianpdf')->name('laporan.harianpdf');
```

Kode program diatas akan berfungsi ketika manajer mengklik tombol *download*, maka akan muncul popup untuk mengunduh laporan.

3.3.2 Penulisan Kode Program Android

Pada bagian ini akan ditunjukkan beberapa bagian koding android untuk mengelola halaman-halaman tertentu. Diantaranya sebagai berikut.

1) Halaman login.java

Login.java merupakan kode progam untuk menginisialisasi berbagai form yang tersedia pada tampilan Login. Berikut kode programnya:

```
public class loginutama extends AppCompatActivity
    { TextView email, password;
    @Override
   protected void onCreate(Bundle savedInstanceState)
        { super.onCreate(savedInstanceState);
        setContentView(R.layout.activity loginutama);
        if (isLogin()){
            Intent intent = new
Intent(loginutama.this, MainActivity.class);
            startActivity(intent); finish();
        email=(TextView) findViewById(R.id.txt email);
        password=(TextView) findViewById(R.id.txt password);
    public void MainActivity (View view) {
login(email.getText().toString(),password.getText().toString());
        public void login(String email, String password) {
        APIService service =
RetrofitClient.getClient().create(APIService.class);
        Call<LoginResponse> userCall = service.userLogin(email,
password);
        userCall.enqueue(new Callback<LoginResponse>()
             { @Override
            public void onResponse(Call<LoginResponse>
call, Response<LoginResponse> response) {
                 if (response.code() == 401) {
                    Toast.makeText(loginutama.this, "Login Gagal",
Toast.LENGTH SHORT).show();
```

```
} else {
 Login login = response.body().getData();
                    saveSession(login.getId(),
login.getName(), login.getEmail(),login.getId());
                    Intent intent = new
Intent(loginutama.this, MainActivity.class);
                    startActivity(intent);
                    finish();
            @Override
            public void onFailure(Call<LoginResponse> call, Throwable
t) {
                Toast.makeText(loginutama.this, "Gagal Terhubung ke
Server", Toast.LENGTH SHORT).show();
        });
   public void saveSession(String id, String name, String
email, String loginId) {
        SharedPreferences.Editor editor =
getSharedPreferences("SESSION", MODE_PRIVATE).edit();
        editor.putString("ID", id );
        editor.putString("NAME", name);
        editor.putString("EMAIL", email );
        editor.putString("ID", id );
        editor.apply();
    public boolean isLogin () {
        SharedPreferences preferences =
getSharedPreferences("SESSION", MODE PRIVATE);
        return !preferences.getString("ID", "").equals("");
}
```

2) Kode program *MainActivity.java*

Setelah proses *login* maka selanjutnya masuk ke halaman utama yang di dalam android studio dinamakan *mainactivity.java*. Berikut kode programnya :

```
drawer.addDrawerListener(toggle);
        toggle.syncState();
        NavigationView navigationView = (NavigationView)
findViewById(R.id.nav_view);
        navigationView.setNavigationItemSelectedListener(this);
        displaySelectedScreen(R.id.nav utama);
        SharedPreferences preferences =
getSharedPreferences("SESSION", MODE PRIVATE)
        View headerView = navigationView.getHeaderView(0);
        TextView headerNama = (TextView)
headerView.findViewById(R.id.header nama);
        TextView headerEmail = (TextView)
headerView.findViewById(R.id.header email);
        headerNama.setText(preferences.getString("NAME",
"").toString());
        headerEmail.setText(preferences.getString("EMAIL",
"").toString());
    public void displaySelectedScreen(int itemId) {
        //creating fragment object
        Fragment fragment = null;
        Bundle bundle = new Bundle();
        switch (itemId) {
            case R.id.nav utama:
                fragment = new HalamanUtama();
                break;
            case R.id.nav makanan:
                fragment = new HalamanMakanan();
                break;
            case R.id.nav minuman:
                fragment = new HalamanMinuman();
                break;
            case R.id.nav obat:
                fragment = new HalamanKesehatan();
                break;
            case R.id.nav_pesanan:
                startActivity(new Intent(MainActivity.this,
PesananSayaActivity.class));
                break;
            case R.id.nav_logout:
                //Hapus session login dari shared preference
                SharedPreferences.Editor editor =
getSharedPreferences("SESSION", MODE PRIVATE).edit();
                editor.clear();
                editor.apply();
                Troli.deleteAll(Troli.class);
                Troli.executeQuery("DELETE FROM SQLITE SEQUENCE WHERE
NAME = 'TROLI'");
                Intent intent = new Intent (MainActivity.this,
loginutama.class);
                startActivity(intent);
                finish();
                break;
            case R.id.nav_alat:
                fragment = new HalamanKRT();
                break:
```

```
if (fragment != null) {
            FragmentTransaction ft =
getSupportFragmentManager().beginTransaction();
            ft.replace(R.id.content_frame,
            fragment); ft.commit();
       DrawerLayout drawer = (DrawerLayout)
findViewById(R.id.drawer layout);
       drawer.closeDrawer(GravityCompat.START);
    @Override
   public void onBackPressed() {
        DrawerLayout drawer = (DrawerLayout)
findViewById(R.id.drawer_layout);
        if (drawer.isDrawerOpen(GravityCompat.START))
            { drawer.closeDrawer(GravityCompat.START);
        } else {
            Intent intent = new Intent(Intent.ACTION MAIN);
            intent.addCategory(Intent.CATEGORY HOME);
            intent.setFlags(Intent.FLAG ACTIVITY NEW TASK);
            startActivity(intent);
            //super.onBackPressed();
    }
    @Override
       getMenuInflater().inflate(R.menu.main,
       menu); return true;
    @Override
   public boolean onOptionsItemSelected(MenuItem item) {
               int id = item.getItemId();
        if (id == R.id.action cart) {
            Intent intent = new Intent (MainActivity.this,
TroliActivity.class);
           startActivity(intent);
           return true;
       return super.onOptionsItemSelected(item);
   @Override
   public boolean onNavigationItemSelected(@NonNull MenuItem item) {
       displaySelectedScreen(item.getItemId());
       return false;
}
```

3) Kode program PosisiKurirActivity.java

Halaman posisi kurir merupakan halaman pada map lacak kurir, pada halaman ini pelanggan bisa mengetahui posisi kurir berada dimana. Berikut adalah kode program PosisiKurirActivity.java:

```
@Override
    protected void onCreate(Bundle savedInstanceState)
        { super.onCreate(savedInstanceState);
        setContentView(R.layout.activity_posisi_kurir);
        SupportMapFragment mapFragment = (SupportMapFragment)
getSupportFragmentManager()
                .findFragmentById(R.id.map);
        mapFragment.getMapAsync(this);
        idKurir = getIntent().getStringExtra("ID_KURIR");
    @Override
    public void onMapReady(GoogleMap googleMap) {
        mMap = googleMap;
        mMap.setMaxZoomPreference(16);
        loginToFirebase();
        Toast toast = Toast.makeText(this, "Mohon tunggu beberapa saat,
kami sedang melacak posisi kurir anda", Toast.LENGTH_LONG);
        TextView v = (TextView)
toast.getView().findViewById(android.R.id.message);
        if( v != null) v.setGravity(Gravity.CENTER);
        toast.show();
    private void loginToFirebase() {
        String email = "admin@admin.com";
        String password = "admin123";
        FirebaseAuth.getInstance().signInWithEmailAndPassword( email,
                password).addOnCompleteListener(new
OnCompleteListener<AuthResult>() {
            @Override
            public void onComplete(Task<AuthResult> task) {
                if (task.isSuccessful()) {
                    subscribeToUpdates();
                    Log. d(TAG, "firebase auth success");
                } else {
                    Log.d(TAG, "firebase auth failed");
            }
        });
    }
    private void subscribeToUpdates() {
        DatabaseReference ref =
FirebaseDatabase.getInstance().getReference(idKurir);
        ref.addChildEventListener(new ChildEventListener() {
            @Override
            public void onChildAdded(DataSnapshot dataSnapshot,
String previousChildName) {
                setMarker(dataSnapshot);
            @Override
            public void onChildChanged(DataSnapshot
dataSnapshot, String previousChildName) {
                setMarker(dataSnapshot);
            @Override
            public void onChildMoved(DataSnapshot dataSnapshot,
String previousChildName) {
            }
```

```
public void onChildRemoved(DataSnapshot dataSnapshot) {
            @Override
            public void onCancelled(DatabaseError error) {
                Log. d(TAG, "Failed to read value.",
error.toException());
        });
    private void setMarker(DataSnapshot dataSnapshot) {
        String key = dataSnapshot.getKey();
        HashMap<String, Object> value = (HashMap<String, Object>)
dataSnapshot.getValue();
        double lat =
Double.parseDouble(value.get("latitude").toString());
        double lng =
Double.parseDouble(value.get("longitude").toString());
        LatLng location = new LatLng(lat, lng);
        if (!mMarkers.containsKey(key)) {
            Marker markerIcon = mMap.addMarker(new MarkerOptions()
                    .title("Kurir Anda")
                    .position(location)
                    .icon(bitmapDescriptorFromVector(this,
R.drawable.ic_directions_car)));
            markerIcon.showInfoWindow();
            mMarkers.put(key, markerIcon);
        } else {
            mMarkers.get(key).setPosition(location);
        LatLngBounds.Builder builder = new LatLngBounds.Builder();
        for (Marker marker : mMarkers.values()) {
            builder.include(marker.getPosition());
mMap.animateCamera(CameraUpdateFactory.newLatLngBounds(builder.build(),
300));
   private BitmapDescriptor
bitmapDescriptorFromVector(Context context, int vectorResId) {
        Drawable vectorDrawable =
ContextCompat.getDrawable(context, vectorResId);
        vectorDrawable.setBounds(0, 0,
vectorDrawable.getIntrinsicWidth(),
vectorDrawable.getIntrinsicHeight());
        Bitmap bitmap =
Bitmap.createBitmap(vectorDrawable.getIntrinsicWidth(),
vectorDrawable.getIntrinsicHeight(), Bitmap.Config.ARGB 8888);
        Canvas canvas = new Canvas(bitmap);
        vectorDrawable.draw(canvas);
        return BitmapDescriptorFactory.fromBitmap(bitmap);
}
```

3.4 Pengujian Sistem

Pengujian sistem adalah proses untuk memastikan keberhasilan sistem yang dibuat. Pengujian aplikasi *delivery order* minimarket dilakukan dengan dua cara yaitu pengujian secara fungsional dan pengujian oleh pengguna.

3.4.1 Pengujian Fungsional Sistem

Pengujian aplikasi dilakukan pada minimarket GENmart yang berlokasi di Jalan Dinar Asri Raya B-1, Tembalang. Pengujian aplikasi *delivery order* minimarket dibagi menjadi 4 bagian yaitu pengujian website admin, pengujian website manajer, pengujian aplikasi android pelanggan, dan pengujian aplikasi android kurir.

3.4.1.1 Rancangan Uji Website Admin

- 1) Uji Login
 - i) Tujuan : Melihat tampilan awal dan syarat untuk masuk ke *website*.
 - ii) Cara Menguji : Masuk ke Aplikasi GENmart dengan mengakses URL http://antargen.informatikapolines.com ke web browser kemudian masukkan username dan password yang telah terdaftar kemudian klik Login.
- 2) Uji Home
 - i) Tujuan : Melihat tampilan setelah *login*.
 - ii) Cara Menguji : setelah *login* akan diarahkan ke tampilan menu *home* atau *dashboard*.
- 3) Uji Data *User*
 - *i*) Tujuan : Menampilkan data *user*.
 - ii) Cara Menguji : Dengan mengklik dibagian menu master *user*.
- 4) Uji Tambah *User*
 - i) Tujuan : Menambahkan data *user*.
 - ii) Cara Menguji: Dengan mengklik *button* tambah *user* diatas tabel data *user* maka akan tampil *form input* data, kemudian isi *form* tersebut dan klik pada *button Create*.
- 5) Uji Edit *User*
 - i) Tujuan : Mengubah data *user* ketika ada perubahan data.

ii) Cara Menguji: Pilih data *user* yang akan diedit, kemudian klik *button* edit. Akan muncul form data-data *user* sebelumnya kemudian ubah pada *field* yang akan diubah, kemudian klik *button Update*.

6) Uji Hapus *User*

- i) Tujuan : Menghapus data *user* yang sudah tidak dibutuhkan.
- ii) Cara Menguji : Pilih data *user* yang akan dihapus, lalu klik *button* Hapus. Akan muncul popup konfirmasi hapus data kemudian klik Ok.

7) Uji Data Produk

- i) Tujuan : Menampilkan data produk.
- ii) Cara Menguji : Dengan mengklik di bagian menu Master Produk.

8) Uji Tambah Produk

- i) Tujuan : Menambahkan data produk
- ii) Cara Menguji: Dengan mengklik *button* tambah produk diatas tabel data produk maka akan tampil form input data, kemudian isi form tersebut dan klik pada *button Create*.

9) Uji Edit Produk

- i) Tujuan : Mengubah data *user* jika ada perubahan data.
- ii) Cara Menguji: Pilih data produk yang akan diedit, kemudian klik *button* edit. Akan muncul form data-data produk sebelumnya kemudian ubah pada *field* yang akan diubah, kemudian klik *button Update*.

10) Uji Hapus Produk

- i) Tujuan : Menghapus data produk yang sudah tidak dibutuhkan lagi.
- ii) Cara Menguji : Pilih data produk yang akan dihapus, lalu klik buttonHapus. Akan muncul popup konfirmasi hapus data kemudian klik Ok.

11) Uji Data Transaksi

- i) Tujuan : Menampilkan data transaksi.
- ii) Cara Menguji : Dengan mengklik dibagian menu Data Transaksi.

12) Uji Hapus Transaksi

- i) Tujuan: Menghapus data transaksi yang sudah tidak dibutuhkan lagi.
- ii) Cara Menguji : Pilih transaksi yang akan dihapus lalu klik *button* hapus, kemudian akan muncul popup konfirmasi hapus data kemudian klik Ok.

3.4.1.2 Rancangan Uji Website Manajer

- 1) Uji *Login*
 - Tujuan : Melihat tampilan awal website dan syarat untuk masuk ke sistem.
 - ii) Cara Menguji : Masuk ke Aplikasi GENmart dengan mengakses URL http://antargen.informatikapolines.com ke web browser kemudian masukkan username dan password yang telah terdaftar kemudian klik Login.
- 2) Uji Home
 - i) Tujuan : Untuk melihat tampilan *home* setelah *login* sebagai manajer.
 - ii) Cara Menguji : Setelah login akan diarahkan ke tampilan menu *home* pada *website* manajer
- 3) Uji Data Transaksi
 - i) Tujuan : Menampilkan data transaksi.
 - ii) Cara Menguji : Dengan mengklik menu Data Transaksi.
- 4) Uji Laporan Harian
 - i) Tujuan : Menampilkan transaksi harian.
 - ii) Cara Menguji : Dengan mengklik menu Harian, maka akan muncul data transaksi pada saat hari manajer mengakses.
- 5) Uji Pilih Tanggal Harian
 - i) Tujuan : Memilih tanggal untuk menampilkan data transaksi.
 - ii) Cara Menguji : Dengan klik di bagian *datepicker* pada halaman harian kemudian pilih tanggal sesuai kebutuhan. Lalu akan muncul data transaksi di hari yang dipilih.
- 6) Uji Download Laporan Harian
 - i) Tujuan : Medownload laporan harian dalam bentuk dokumen pdf.
 - ii) Cara Menguji : Dengan mengklik button download lalu akan muncul popup untuk download laporan. Hasil dapat dilihat setelah proses download selesai.
- 7) Uji Laporan Harian PDF
 - i) Tujuan : Menampilkasn laporan harian dalam bentuk dokumen pdf.

- ii) Cara Menguji : Dengan membuka dokumen yang sudah berhasil diunduh.
- 8) Uji Laporan Bulanan
 - i) Tujuan : Menampilkan transaksi bulanan.
 - ii) Cara Menguji : Dengan mengklik menu Bulanan, maka akan muncul data transaksi bulanan pada saat diakses.
- 9) Uji Pilih Bulan
 - i) Tujuan : Memilih bulan untuk menampilkan data transaksi.
 - ii) Cara Menguji : Dengan klik di bagian *datepicker* pada halaman bulanan kemudian pilih bulan sesuai kebutuhan. Lalu akan muncul data transaksi di bulan yang dipilih.
- 10) Uji Download Laporan Bulanan
 - i) Tujuan : Medownload laporan bulanan dalam bentuk dokumen pdf.
 - ii) Cara Menguji : Dengan mengklik button download lalu akan muncul popup untuk download laporan. Hasil dapat dilihat setelah proses download selesai.
- 11) Uji Laporan Bulanan PDF
 - i) Tujuan : Menampilkan laporan bulanan dalam bentuk dokumen pdf.
 - ii) Cara Menguji : Dengan membuka file yang berhasil diunduh.

3.4.1.3 Rancangan Uji Android Pelanggan

- 1) Uji Login
 - i) Tujuan : Syarat untuk masuk ke aplikasi.
 - ii) Cara Menguji : Masukkan *e-mail* dan *password* yang telah di daftarkan oleh admin, kemudian klik tombol *Login*.
- 2) Uji Utama
 - i) Tujuan : Melihat halaman awal apabila login sukses.
 - ii) Cara Menguji : Klik pada bagian menu utama.
- 3) Uji Halaman Makanan
 - i) Tujuan : Melihat data produk makanan
 - ii) Cara Menguji : Klik pada bagian menu Makanan.
- 4) Uji Halaman Minuman
 - i) Tujuan : Melihat data produk minuman.

- ii) Cara Menguji : Klik pada bagian menu Minuman.
- 5) Uji Halaman Kesehatan
 - i) Tujuan : Melihat data produk Kesehatan
 - ii) Cara Menguji : Klik pada bagian menu Kesehatan
- 6) Uji Halaman Rumah Tangga
 - i) Tujuan : Melihat data produk Kesehatan
 - ii) Cara Menguji : Klik pada bagian menu Kesehatan
- 7) Uji Halaman Pesanan Saya
 - i) Tujuan : Melihat daftar perbelanjaan yang sudah berhasil dibeli
 - ii) Cara Menguji : Klik pada bagian menu pesanan saya
- 8) Uji Lacak Posisi Kurir
 - i) Tujuan : Melihat keberadaan posisi kurir
 - ii) Cara Menguji : Klik pada Posisi Kurir
- 9) Uji Logout
 - i) Tujuan : Keluar dari aplikasi
 - ii) Cara Menguji : Klik pada bagian menu logout
- 10) Uji Detail Produk
 - i) Tujuan : Untuk melihat nama produk, harga dan stok produk
 - ii) Cara Menguji: Tekan salah satu produk yang muncul pada kotak dialog
- 11) Uji Jumlah Pembelian
 - i) Tujuan : Untuk menambahkan jumlah produk yang akan dibeli
 - ii) Cara Menguji : Tekan tombol tambah ke troli pada detail produk lalu masukkan jumlah produk setelah itu tekan tombol tambah ke troli
- 12) Uji Troli
 - Tujuan : Melihat produk apa saja yang sudah berhasil dimasukkan kedalam troli, melihat total jumlah perbelanjaan, dan setelah itu melanjutkan ke pengiriman
 - ii) Cara Menguji : Tekan icon troli pada aplikasi yang berada pada pojok kanan atas
- 13) Uji Pengiriman
 - Tujuan : Untuk mengisi alamat pengiriman, nomor telefon dan memilih kurir

ii) Cara Menguji : Klik pada icon troli, lalu klik tombol lanjutkan ke pengiriman, setelah itu isi form pengiriman dan klik tombol kirim.

3.4.1.4 Rancangan Uji Android Kurir

- 1) Uji Login
 - i) Tujuan : Syarat untuk masuk ke aplikasi.
 - ii) Cara Menguji : Masukkan *e-mail* dan *password* yang telah di daftarkan oleh admin, kemudian klik tombol *Login*.
- 2) Uji Pengiriman
 - i) Tujuan : Untuk memperbarui status pesanan
 - ii) Cara Menguji : Klik pada status pesanan, setelah itu perbarui status sesuai dengan kondisi yang ada dengan cara klik saja status pesanan tersebut
- 3) Uji *Maps*
 - i) Tujuan : Melihat tracking rute rumah pelanggan
 - ii) Cara Menguji : Klik pada bagian menu *Map*.
- 4) Uji Histori
 - i) Tujuan : Melihat data produk-produk yang pernah diantar oleh kurir.
 - ii) Cara Menguji : Klik pada bagian menu Histori
- 5) Uji Logout
 - i) Tujuan : Keluar dari aplikasi
 - ii) Cara Menguji : Klik pada bagian menu logout.

Pengujian aplikasi *delivery order* minimarket berbasis web dan android dapat dilihat pada Tabel 3.4

Tabel 3. 4 Hasil pengujian fungsional sistem

No.	Nama Uji	Hasil
1	Uji log in	Masing-masing <i>user</i> bisa mengakses halamannya
		sesuai dengan hak aksesnya. Dan user yang belum
		terdaftar tidak dapat masuk ke dalam <i>aplikasi</i> .
2	Uji manajemen data	Admin toko dapat melakukan tambah, edit, dan
	user pada admin toko	hapus data akun <i>login user</i> tanpa <i>error</i> .

3	Uji manajemen data	Admin toko dapat melakukan tambah, edit, dan		
	produk pada admin	hapus data produk tanpa <i>error</i> .		
	toko			
4	Uji manajemen data	Admin toko dapat menghapus data transaksi yang		
	transaksi pada admin	sudah tidak dibutuhkan lagi tanpa <i>error</i> .		
	toko			
5	Uji laporan transaksi	Manajer dapat mengakses data transaksi harian		
	pada manajer	dan bulanan, serta dapat mengunduh laporan		
		tersebut dalam bentuk dokumen pdf tanpa error.		
6	Uji pemesanan pada	Pelanggan dapat melakukan pemesanan produk		
	pelanggan	pada aplikasi dengan minimal pembelian Rp.		
		50.000,- dengan benar.		
7	Uji fitur delivery order	Pelanggan dapat menggunakan fitur delivery		
	pada pelanggan	order setelah pesanan masuk ke dalam sistem.		
8	Uji lacak posisi kurir	Pelanggan dapat melacak posisi kurir pada saat		
	pada pelanggan	kurir mengantar pesanannya tanpa <i>error</i> .		
9	Uji monitoring status	Pelanggan dapat memonitor status pesanannya		
	pesanan pada	dengan benar.		
	pelanggan			
10	Uji konfirmasi status	Kurir dapat mengkonfirmasi status pesanan		
	pesanan pada kurir	pelanggan pada saat menerima notifikasi pada		
		aplikasi kurir dengan benar.		
11	Uji <i>maps</i> pada kurir	Kurir dapat mengirim pesanan ke lokasi		
		pelanggan dengan benar dengan menggunakan		
		fitur <i>maps</i> yang tersedia pada aplikasi kurir.		

3.4.2 Pengujian Oleh Pengguna

Untuk dapat mengetahui tingkat kepuasan pengguna, maka dilakukan ujicoba secara langsung oleh pengguna. Berikut beberapa aspek yang menjadi bahan pertimbangan aspek tingkat kepuasan pelanggan dapat dilihat pada tabel 3.5, indikator peniliaian dapat dilihat pada Tabel 3.6, tingkat kepuasan pengguna pemakai aplikasi yang dapat dilihat pada Tabel 3.7.

Tabel 3. 5 Aspek Tingkat Kepuasan Pelanggan

No	Aspek			
1	Apakah anda setuju aplikasi GENmart berbasis web dan android sudah			
	memiliki antarmuka pengguna yang menarik dan mudah dipahami?			
2	Apakah anda setuju bahwa aplikasi GENmart mempermudah admin			
	untuk mengelola data-data di toko ?			
3	Apakah anda setuju bahwa aplikasi GENmart mempermudah admin			
	dalam mengelola stok produk karena sudah ada fitur stock warning?			
4	Apakah anda setuju bahwa aplikasi GENmart mempermudah manajer			
	dalam hal pelaporan transaksi ?			
5	Apakah anda setuju bahwa aplikasi GENmart mempermudah pelanggan			
	dalam melakukan perbelanjaan dengan sistem delivery?			
6	Apakah anda setuju bahwa aplikasi GENmart mempermudah pelanggan			
	untuk melacak posisi kurir saat mengantar pesanannya ?			
7	Apakah anda setuju bahwaaplikasi GENmart mempermudah kurir			
	dalam mengelola pesanan yang harus diantar ?			
8	Apakah anda setuju bahwa aplikasi GENmart mempermudah kurir			
	untuk mengantar pesanan sampai tujuan ?			
9	Apakah anda setuju bahwa aplikasi GENmart mempermudah proses			
	transaksi delivery order?			
10	Apakah anda setuju jika aplikasi GENmart ini diterapkan dan digunakan			
	oleh <i>minimarket</i> GENmart untuk memperluas jangkauan pasar ?			

Tabel 3. 6 Indikator Penilaian

Indikator	Nilai
Tidak Setuju	1
Kurang Setuju	2
Cukup	3
Setuju	4
Sangat Setuju	5

Tabel 3. 7 Indikator Kepuasan Pemakai Aplikasi

Indikator	Nilai
Sangat Memuaskan	81% - 100%
Memuaskan	61% - 80%
Cukup Memuaskan	41% - 60%
Tidak Memuaskan	21% - 40%
Sangat Tidak Memuaskan	0 -20%

Aspek tingkat kepuasan pengguna tersebut diperoleh melalui hasil perhitungan dengan rumus:

Presentase kepuasan pengguna (%) = _____

3.5 Pemeliharaan Aplikasi

Tahap terakhir adalah pemeliharaan. Aplikasi ini akan diunggah ke *Google Play* untuk memudahkan pengunduhan dan *update* aplikasi. Perubahan tersebut bisa terjadi karena masih adanya kesalahan pada sistem maupun karena pengguna membutuhkan perkembangan fungsional yang lebih baik.

BAB IV ANALISIS DAN PEMBAHASAN

Bab ini membahas analisis aplikasi yang telah dibuat dan mengujinya secara langsung kepada pengguna. Pengujian fungsional web ditunjukkan dengan hasil uji pada beberapa *platform* web diantaranya Google Chrome, Mozilla Firefox, dan Internet Explorer. Sedangkan aplikasi android diuji pada 3 versi operasi sistem android yaitu Kitkat, Lollipop, dan Nugget. Hasil pengujian kuisoner kepada responden juga dibahas pada bagian akhir bab ini.

4.1 Analisis Fungsional Sistem

Pengujian fungsional aplikasi *delivery order* pada minimarket GENmart dijelaskan sesuai analisis dari perancangan yang dibuat sebelumnya. Pengujian fungsionalitas menggunakan web dan android. Aplikasi web terdiri dari admin toko dan manajer. Web pada admin toko meliputi fitur halaman *login*, *dashboard*, manajemen user, manajemen produk, *stock warning*, dan manajemen transaksi. Sedangkan web pada manajer meliputi fitur halaman *login*, *dashboard*, dan informasi manajemen transaksi. Aplikasi android terdiri dari pelanggan dan kurir. Aplikasi android pada pelanggan meliputi fitur *login*, kategori produk, dan menu pesanan. Aplikasi android pada kurir meliputi fitur *login*, menu pengiriman, menu histori, dan menu map.

4.1.1 Web GENmart

Web aplikasi *delivery order* minimarket GENmart akan menampilkan 2 jenis halaman pengguna yaitu admin toko dan manajer.

1) Halaman *log in* admin toko

Halamn *log in* merupakan halaman awal sebelum *user* masuk ke dalam aplikasi dan mengakses menu-menu yang ada. Halaman *login* akan muncul ketika *user* menmbuak *url* http://antargen.informatikapolines.com. Pengujian fungsionalitas pada proses ini dapat dilihat pada Gambar 4.1.

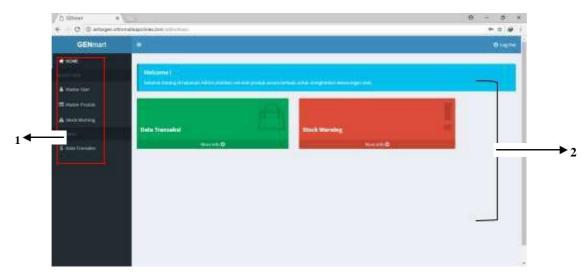


Gambar 4. 1 Halaman log in admin toko

Gambar 4.1 terdiri dari beberapa bagian. Bagian 1 adalah *form e-mail* yang digunakan untuk mengisi *e-mail user* yang sudah didaftarkan oleh admin, kemudian bagian 2 adalah *form password* digunakan untuk menginputkan *password user* dan bagian 3 adalah tombol *Sign In* yang berfungsi membawa *user* masuk ke halaman sesuai hak akses masing-masing. Namun apabila *e-mail* dan *password* yang dimasukkan salah, maka *user* akan tetap berada dihalaman *Login*. Proses ini sudah dapat berjalan dengan baik, *user* admin toko akan dibawa menuju menu untuk manajemen akun *user*, produk, dan data transaksi. *User* manajer akan dibawa menuju menu laporan transaksi harian dan transaksi bulanan.

2) Dashboard admin toko

Halaman *dashboard* adalah halaman yang tampil setelah *user* admin toko berhasil *login* ke dalam aplikasi web. Halaman *dashboard* admin menampilkan opsi menu *stock warning* dan transaksi yang berfungsi menuju kehalaman tersebut selain dengan memilih menu pada *sidebar*. Pengujian fungsionalitas pada proses ini dapat dilihat pada Gambar 4.2.

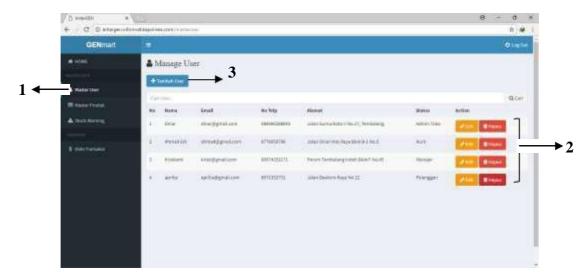


Gambar 4. 2 Dashboard admin toko

Gambar 4.2 terdiri dari beberapa bagian. Bagian 1 merupakan menu-menu yang dapat dikelola oleh admin toko yaitu menu kelola master *user* berfungsi untuk mengelola data *user*, master produk berfungsi untuk mengelola data produk, *stock warning* berfungsi untuk melihat produk yang stoknya hampir habis, dan kelola data transaksi yang berfungsi untuk menampilkan data transaksi pada aplikasi, bagian 2 merupakan halaman awal ketika berhasil *login* sebagai admin, bagian ini berfungsi sebagai menu opsional untuk masuk ke halaman data transaksi dan *stock warning*. Proses yang ada pada halaman ini sudah dapat berjalan dengan baik.

3) Halaman master *user*

Halaman master *user* merupakan halaman yang berfungsi untuk mengelola akun *user* yang akan menggunakan aplikasi *delivery order* minimarket. Pengujian fungsionalitas pada proses ini dapat dilihat pada Gambar 4.3

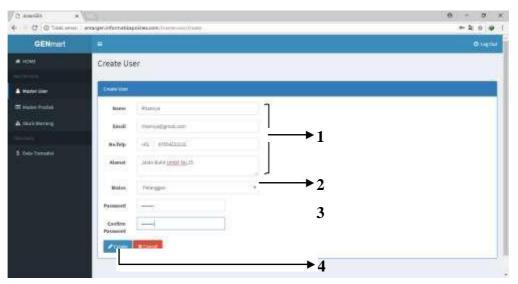


Gambar 4. 3 Halaman master user

Gambar 4.3 terdiri dari beberapa bagian. Bagian 1 adalah menu master *user* yang diklik berfungsi untuk mengelola data *user*, bagian 2 adalah data master *user* digunakan untuk menampilkan data *user*, dan bagian 3 adalah tombol tambah *user* yang digunakan untuk menambahkan *user* baru. Proses yang ada pada halaman ini sudah dapat berjalan dengan baik.

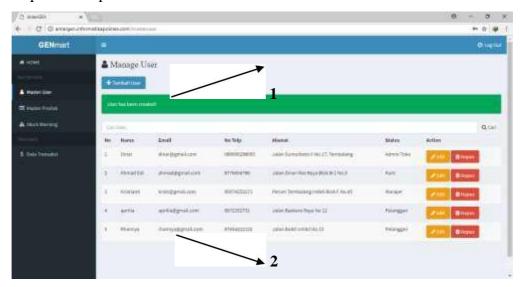
4) Halaman tambah *user*

Halaman tambah *user* adalah halaman yang digunakan untuk menambahkan *user* baru. Pengujian fungsionalitas pada proses halaman ini dapat dilihat pada Gambar 4.4.



Gambar 4. 4 Halaman tambah user

Gambar 4.4 terdiri dari beberapa bagian. Bagian 1 merupakan *textbox* yang digunakan untuk menambahkan data-data *user*, bagian 2 merupakan *dropdown* yang digunakan untuk memilih status *user*, pemilihan status ini berfungsi untuk membedakan hak akses *log in* pada aplikasi. Bagian 3 merupakan *textbox password* yang digunakan untuk mengisi *password* yang digunakan untuk *login* kedalam aplikasi, bagian 4 merupakan tombol *Create* yang berfungsi untuk menyimpan data ke database. Kemudian jika berhasil menginputkan akan diarahkan ke halaman master *user* dan akan muncul notifikasi. Notifikasi dan data baru dapat dilihat pada Gambar 4.5

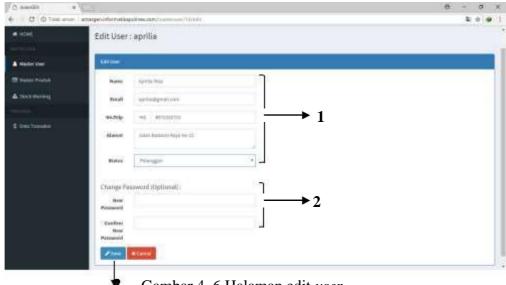


Gambar 4. 5 Berhasil tambah user

Gambar 4.5 akan tampil setelah berhasil menginputkan *user* baru. Terdapat beberapa bagian, bagian 1 merupakan notifikasi yang berfungsi untuk sebagai tanda berhasil menambahkan *user*, bagian 2 merupakan data *user* yang baru ditambahkan. Proses yang ada pada halaman ini sudah dapat berjalan dengan baik.

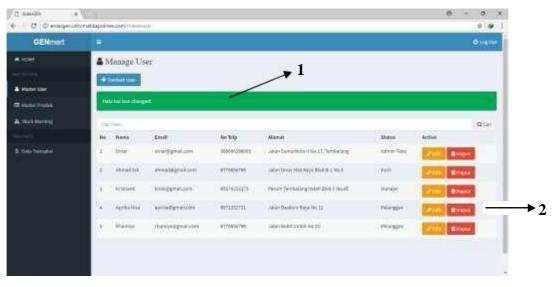
5) Halaman edit user

Menu kedua yang dapat dikelola oleh admin toko adalah menu master produk yang digunakan untuk mengubah data *user* jika ada perubahan. Pengujian fungsionalitas pada proses ini dapat dilihat pada Gambar 4.6.



Gambar 4. 6 Halaman edit *user*

Gambar 4.6 terdiri dari beberapa bagian. Bagian 1 merupakan *textbox* yang digunakan untuk mengganti data *user* sebelumnya, bagian 2 merupakan *textbox* yang digunakan untuk mengganti *password* tetapi hal ini bersifat opsional, dan bagian 3 merupakan tombol *Save* yang berfungsi untuk menyimpan data baru. Jika sudah berhasil maka akan masuk ke halaman master *user* dan muncul notifikasi. Notifikasi dan data baru dapat dilihat pada Gambar 4.7.

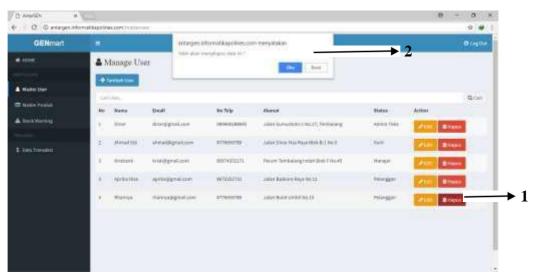


Gambar 4. 7 Berhasil edit user

Gambar 4.7 akan tampil ketika berhasil mengedit data *user*. Terdapat beberapa bagian, bagian 1 merupakan notifikasi yang berfungsi sebagai tanda berhasil mengedit, bagian 2 merupakan data yang baru diedit. Proses pada menu edit *user* sudah dapat berjalan dengan baik.

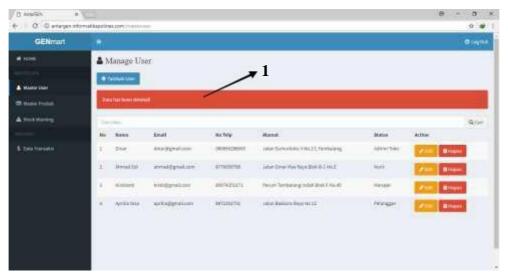
6) Halaman hapus *user*

Menu selanjutnya adalah hapus *user* yang digunakan untuk menghapus *user* yang sudah tidak dibutuhkan lagi. Pengujian fungsionalitas pada proses ini dapat dilihat pada Gambar 4.8.



Gambar 4. 8 Halaman hapus *user*

Gambar 4.8 terdiri dari beberapa bagian, bagian 1 adalah tombol hapus yang digunakan untuk menghapus data *user*,bagian 2 merupakan *popup* yang muncul ketika tombol hapus diklik. Kemudian jika diklik *ok* maka data akan terhapus dan akan muncul notifikasi yang dapat dilihat pada Gambar 4.9.



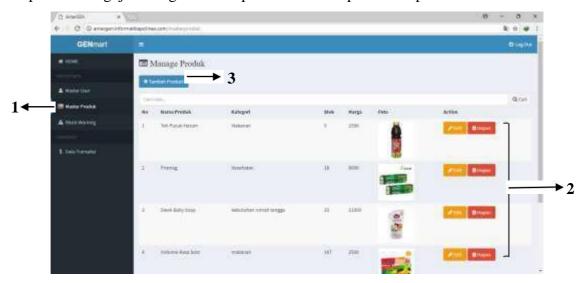
Gambar 4. 9 Berhasil hapus *user*

Gambar 4.9 merupakan notifikasi jika berhasil menghapus *user*. Bagian 1 merupakan notifikasi yang berfungsi sebagai tanda berhasil hapus data. Dan pada

halaman master *user* sudah tidak ada lagi data tersebut. Proses pada menu hapus *user* sudah dapat berjalan dengan baik.

7) Halaman master produk

Menu selanjutnya yang dapat dikelola oleh admin toko adalah menu master produk yang berfungsi untuk mengelola data-data produk yang dapat dibeli pada aplikasi. Pengujian fungsionalitas pada menu inidapat dilihat pada Gambar 4.10.

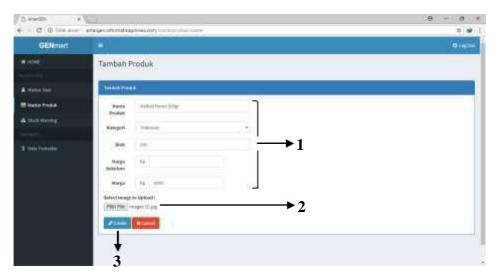


Gambar 4. 10 Halaman master produk

Gambar 4.10 terdiri dari beberapa bagian, bagian 1 merupakan menu master produk yang sedang aktif, bagian 2 merupakan data-data produk, dan bagian 3 merupakan tombol tambah produk yang berfungsi untuk menambahkan produk baru. Pengujian fungsionalitas pada menu ini sudah dapat berjalan dengan baik.

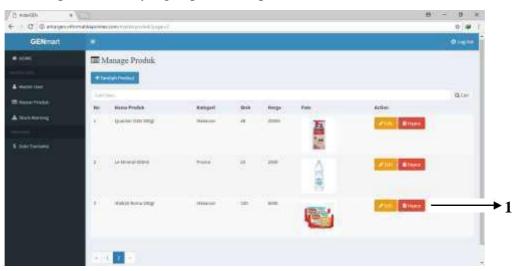
8) Halaman tambah produk

Halaman tambah produk merupakan halaman yang digunakan untuk menambahkan produk baru. Pengujian fungsionalitas pada menu tambah produk dapat dilihat pada Gambar 4.11.



Gambar 4. 11 halaman tambah produk

Gambar 4.11 terdiri dari beberapa bagian. Bagian 1 merupakan *textbox* yang digunakan untuk menambahkan data produk baru, bagian 2 merupakan tombol yang digunakan untuk menambahkan foto produk, bagian 3 merupakan tombol *Create* yang berfungsi untuk menyimpan produk ke database. Kemudian jika berhasil menyimpan maka akan masuk ke halaman master produk dan akan muncul data produk baru yang dapat dilihat pada Gambar 4.12.

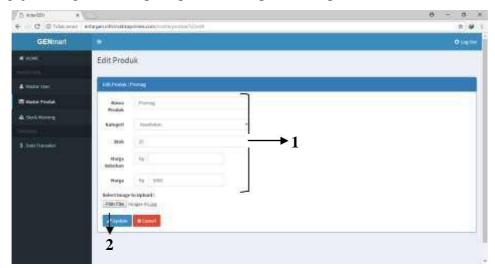


Gambar 4. 12 Berhasil tambah produk

Gambar 4.12 bagian 1 merupakan data produk yang baru saja berhasil diinputkan. Pengujian pada proses ini sudah dapat berjalan dengan baik.

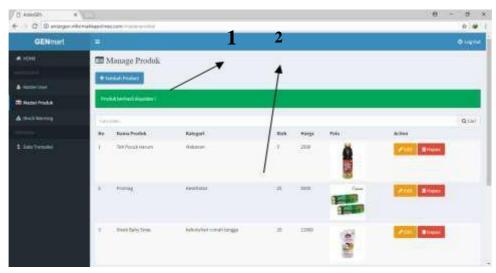
9) Halaman edit produk

Halaman edit produk digunakan untuk mengubah data produk yang perlu diubah. Pengujian fungsionalitas pada proses ini dapat dilihat pada Gambar 4.13.



Gambar 4. 13 Halaman edit produk

Gambar 4.13 terdiri dari beberapa bagian. Bagian 1 merupakan *form* untuk mengubah data produk, bagian 2 merupakan tombol *Update* yang berfungsi untuk menyimpan perubahan data. Kemudian setelah berhasil mengubah data akan muncul notifikasi yang dapat dilihat pada Gambar 4.14.

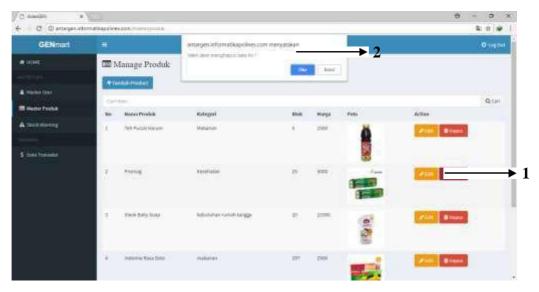


Gambar 4. 14 Berhasil edit produk

Gambar 4.14 terdiri dari beberapa bagian. Bagian 1 merupakan notifikasi berhasil mengubah data produk, bagian 2 merupakan perubahan data pada produk. Proses pada menu edit produk sudah dapat berjalan dengan baik.

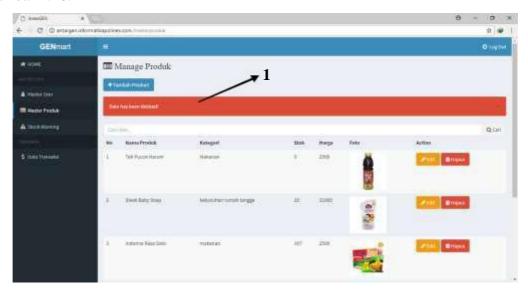
10) Halaman hapus produk

Halaman hapus produk digunakan untuk menghapus data produk yang sudah tidak dibutuhkan lagi. Pengujian fungsionalitas pada menu hapus produk dapat dilihat pada Gambar 4.15.



Gambar 4. 15 Halaman hapus produk

Gambar 4.15 terdiri dari beberapa bagian. Bagian 1 merupakan tombol hapus untuk menghapus data produk yang dipilih, bagian 2 merupakan *pop up* yang muncul ketika tombol hapus diklik. Jika diklik ok maka akan masuk ke halaman master produk dan muncul notifikasi berhasil hapus data yang dapat dilihat pada Gambar 4.16.

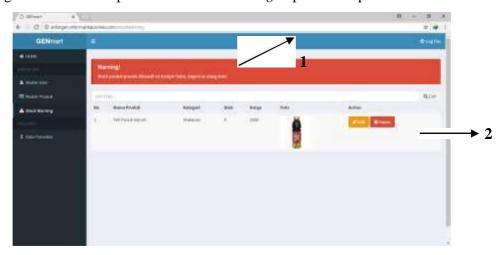


Gambar 4. 16 Berhasil hapus produk

Gambar 4.16 bagian 1 merupakan notifikasi berhasil menghapus data produk. Pada menu master produk sudah tidak ada lagi data promag karena baru saja dihapus. Proses pada menu ini sudah dapat berjalan dengan baik.

11) Halaman stock warning

Halaman *stock warning* digunakan untuk mengingatkan kepada admin toko tentang produk-produk yang stoknya hampir kosong untuk segera ditambah stok. Pengujian fungsionalitas sistem pada menu *stock warning* dapat dilihat pada Gambar 4.17.

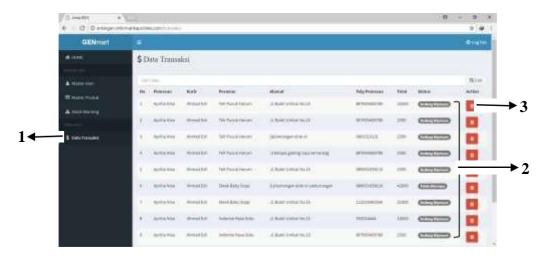


Gambar 4. 17 halaman stock warning

Gambar 4.17 terdiri dari beberapa bagian. Bagian 1 merupakan peringatan untuk menambah stok produk yang hampir habis, dan bagian 2 merupakan data produk yang hampir habis. Proses pada menu *stock warning* sudah dapat berjalan dengan baik.

12) Halaman data transaksi

Menu selanjutnya yang dikelola oleh admin toko adalah menu data transaksi. halaman ini digunakan untuk melihat data-data transaksi yang terjadi pada aplikasi dan digunakan untuk menghapus data transaksi yang sudah tidak dibutuhkan lagi. Pengujian fungsionalitas pada menu ini dapat dilihat pada Gambar 4.18.

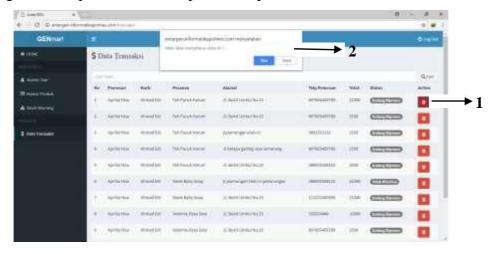


Gambar 4. 18 Halaman data transaksi

Gambar 4.18 terdiri dari beberapa bagian. Bagian 1 adalah menu data transaksi yang aktif, bagian 2 adalah data-data transaksi yang terjadi pada aplikasi baik yang sedang diproses maupun transaksi yang sudah diselesai, dan bagian 3 merupakan tombol hapus yang digunakan untuk menghapus data transaksi yang sudah tidak diperlukan. Proses pada menu ini sudah dapat berjalan dengan baik.

13) Halaman hapus data transaksi

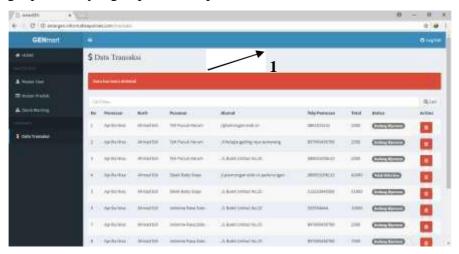
Menu selanjutnya adalah menu hapus data transaksi yang digunakan untuk menghapus data transaksi yang sudah tidak dibuthkan lagi. Pengujian fungsionalitas pada menu ini dapat dilihat pada Gambar 4.19.



Gambar 4. 19 Halaman hapus transaksi

Gambar 4.19 terdiri dari beberapa bagian. Bagian 1 adalah tombol hapus yang berfungsi untuk menghapus data transaksi, bagian 2 merupakan *pop up* yang

muncul ketika tombol hapus diklik. Jika diklik *ok* akan muncul notifikasi berhasil menghapus data yang dapat dilihat pada Gambar 4.20.



Gambar 4. 20 Berhasil hapus transaksi

Gambar 4.20 bagian 1 adalah notifikasi berhasil menghapus data. Data yang sudah dihapus tidak akan tampil lagi pada tabel data transaksi. proses pada menu hapus data transaksi ini sudah dapat berjalan dengan baik.

14) Halaman login manajer

Halaman *login* manajer digunakan untuk membawa *user* yang masuk sebagai manajer ke halaman yang dapat dikelola oleh manajer. Tampilan *login* dapat dilihat pada Gambar 4.21.

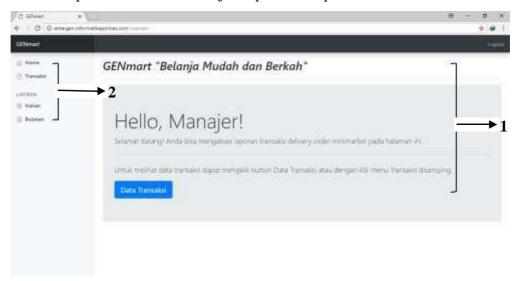


Gambar 4. 21 Halaman *log in* manajer

Gambar 4.21 terdiri dari beberapa bagian. Bagian 1 merupakan *textbox* untuk mengisi *e-mail* manajer, bagian 2 merupakan *textbox* untuk mengisi *password* yang sesuai dengan *e-mail*nya, dan bagian 3 adalah tombol *login* untuk masuk ke dalam aplikasi web manajer. Proses pada menu *login* manajer ini sudah dapat berjalan dengan baik.

15) Dashboard manajer

Dashboard manajer digunakan sebagai halaman awal yang tampil ketika *user* berhasil *login* sebagai manajer. Pada dashboard terdapat tombol data transaksi yang digunakan untuk melihat transaksi dengan opsi lain selain mengklik menu transaksi. Tampilan *dashboard* manajer dapat dilihat pada Gambar 4.22.

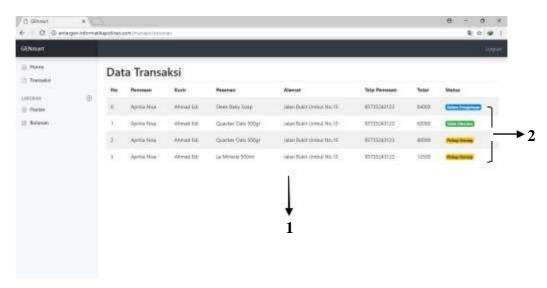


Gambar 4. 22 Dashboard manajer

Gambar 4.22 terdiri dari beberapa bagaian. Bagian 1 adalah halaman ketika berhasil *login*, bagian 2 merupakan menu yang dapat dikelola oleh manajer. Proses pada menu ini sudah dapat berjalan dengan baik.

16) Halaman data transaksi

Menu selanjutnya yang dapat diakses oleh manajer adalah halaman data transaksi. Halaman ini digunakan untuk menampilkan data-data transaksi yang terjadi pada aplikasi android, baik data yang sedang diproses maupun yang telah selesai. Pengujian fungsionalitas pada menu transaksi dapat dilihat pada Gambar 4.23.

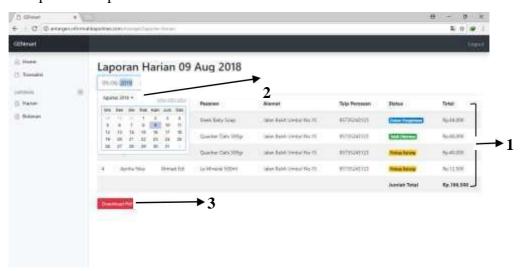


Gambar 4. 23 Halaman data transaksi

Gambar 4.23 bagian 1 merupakan data transaksi yang tampil sebagai laporan untuk manajer, bagian 2 merupakan status transaksi pada aplikasi berupa sedang diproses, dalam pengiriman, atau telah diterima. Ditandai dengan indikator warna dan keterangannya. Proses pada menu ini sudah dapat berjalan dengan baik.

17) Halaman data laporan harian

Menu selanjutnya adalah menu laporan harian yang berfungsi untuk menampilkan laporan transaksi harian. Pada halaman ini manajer dapat memilh tanggal berapa ingin ditampilkan laporan transaksinya. Pengujian fungsionalitas pada menu harian dapat dilihat pada Gambar 4.24.

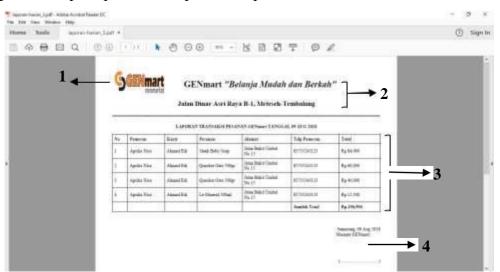


Gambar 4. 24 Halaman data laporan harian

Gambar 4.24 terdiri dari beberapa bagian. Bagian 1 merupakan data transaksi yang tampil pada hari saat manajer mengakses web, bagian 2 merupakan *datepicker* yang digunakan untuk memilih tanggal transaksi, dan bagian 3 merupakan tombol download pdf yang berfungsi untuk mengunduh laporan harian dalam bentuk dokumen pdf. Proses pada menu laporan harian ini sudah dapat berjalan dengan baik.

18) Halaman laporan harian pdf

Halaman laporan harian pdf adalah halaman yang digunakan untuk menampilkan data transaksi harian dalam bentuk pdf. Halaman ini akan tampil setelah manajer mengunduh data trasaksi harian sesuai tanggal yang dipilih. Pengujian fungsionalitas pada proses ini dapat dilihat pada Gambar 4.25.

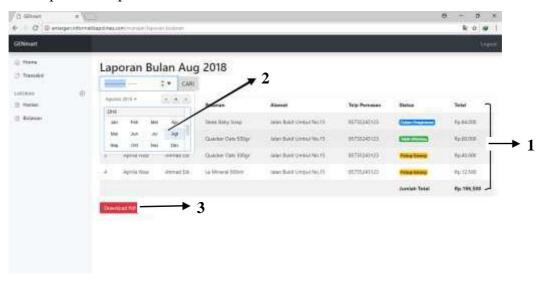


Gambar 4. 25 Halaman laporan harian pdf

Gambar 4.25 terdiri dari beberapa bagian. Bagian 1 merupakan logo dari minimarket yang menerapkan aplikasi *delivery order* yaitu GENmart minimarket, bagian 2 merupakan simbol dari minimarket GENmart itu sendiri beserta alamat tokonya, bagian 3 merupakan data transaksi harian yang dipilih dan diunduh, dan bagian 4 merupakan pengesahan oleh manajer GENmart. Proses pada halaman laporan harian pdf ini sudah berjalan dengan baik.

19) Halaman data laporan bulanan

Menu selanjutnya adalah menu laporan bulanan yang digunakan untuk menampilkan data transaksi bulanan, pada menu ini manajer dapat memilh bulan yang ingin ditampilkan laporan transaksinya. Pengujian fungsionalitas pada menu bulanan dapat dilihat pada Gambar 4.26.

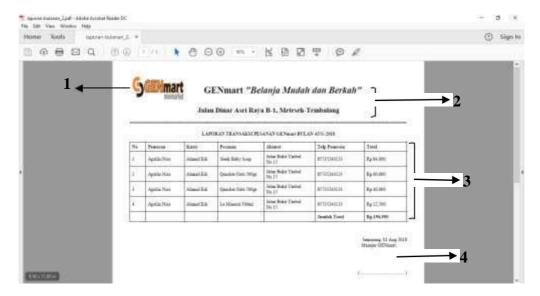


Gambar 4. 26 Halaman data laporan bulanan

Gambar 4.26 terdiri dari beberapa bagian. Bagian 1 merupakan data transaksi yang tampil pada bulan saat manajer mengakses web, bagian 2 merupakan *datepicker* yang berfungsi untuk memilih bulan transaksi, dan bagian 3 merupakan tombol download pdf untuk mengunduh laporan bulanan dalam bentuk dokumen pdf. Proses pada menu laporan bulanan ini sudah dapat berjalan dengan baik.

20) Halaman laporan bulanan pdf

Halaman laporan bulanan pdf adalah halaman yang digunakan untuk menampilkan data transaksi bulanan dalam bentuk pdf. Halaman ini akan tampil setelah manajer mengunduh data trasaksi harian sesuai tanggal yang dipilih. Pengujian fungsionalitas pada proses ini dapat dilihat pada Gambar 4.27.



Gambar 4. 27 Halaman laporan bulan pdf

Gambar 4.27 terdiri dari beberapa bagian. Bagian 1 merupakan logo dari minimarket yang menerapkan aplikasi *delivery order* yaitu GENmart minimarket, bagian 2 merupakan simbol dari minimarket GENmart itu sendiri beserta alamat tokonya, bagian 3 merupakan data transaksi bulanan yang dipilih dan diunduh, dan bagian 4 merupakan pengesahan oleh manajer GENmart. Proses pada halaman laporan bulanan pdf ini sudah berjalan dengan baik.

Analisi fugsional web aplikasi *delivery order* minimarket juga dilakukan dengan menggunakan web browser yang berbeda yaitu melalui Mozilla Firefox, Google Chrome, dan Internet Explorer. Hasil dari iji fungsional dengan 3 browser yang berbeda dapat dilihat pada Tabel 4.1.

Tabel 4. 1 Hasil pengujian fungsional web *delivery order minimarket* dengan 3 browser berbeda.

No	Pengujian	Internet	Mozilla	Google
		Explorer	Firefox	Chrome
1	Log in admintoko	Berhasil	Berhasil	Berhasil
2	Dashboard admintoko	Berhasil	Berhasil	Berhasil
3	Halaman master <i>user</i>	Berhasil	Berhasil	Berhasil
4	Halaman tambah <i>user</i>	Berhasil	Berhasil	Berhasil
5	Halaman edit user	Berhasil	Berhasil	Berhasil
6	Halaman hapus user	Berhasil	Berhasil	Berhasil
7	Halaman mastetr produk	Berhasil	Berhasil	Berhasil

8	Halaman tambah produk	Berhasil	Berhasil	Berhasil
9	Halaman edit produk	Berhasil	Berhasil	Berhasil
10	Halaman hapus produk	Berhasil	Berhasil	Berhasil
11	Halaman stock warning	Berhasil	Berhasil	Berhasil
12	Halaman data transaksi	Berhasil	Berhasil	Berhasil
13	Halaman hapus data transaksi	Berhasil	Berhasil	Berhasil
14	Log in manajer	Berhasil	Berhasil	Berhasil
15	Dashboard manajer	Berhasil	Berhasil	Berhasil
16	Halaman laporan transaksi	Berhasil	Berhasil	Berhasil
17	Halaman transaksi harian	Berhasil	Berhasil	Berhasil
18	Halaman transaksi harian pdf	Berhasil	Berhasil	Berhasil
19	Halaman transaksi bulanan	Berhasil	Berhasil	Berhasil
20	Halaman transaksi bulanan pdf	Berhasil	Berhasil	Berhasil

Tabel 4.1 memperlihatkan hasil pengujian fungsionalitas aplikasi *delivery order* minmarket berbasis web pada *user* admintoko dan manajer, dan terdapat 20 fungsi yang sudah diuji pada 3 browser yaitu Google Chrome, Mozilla Firefox, dan Internet Explorer. Dapat disimpulkan bahwa aplikasi *delivery order* minimarket berbasis web untuk halaman kelola admin toko dan manajer sudah berjalan baik di beberapa *web browser*.

4.1.2 Android GENmart

Aplikasi android GENmart dibagi menjadi dua aplikasi yaitu aplikasi pelanggan dan aplikasi kurir.

4.1.2.1 Android GENmart Pelanggan

Aplikasi GENmart Pelanggan pada android berfungsi untuk melakukan pemesanan produk dengan menggunakan *smartphone*. Terdapat 7 buah menu yang tersedia, yaitu halaman *login*, menu utama, kategori makanan, kategori minuman, kategori kesehatan, kategori rumah tangga, dan menu pesanan saya.

1) Halaman *Login*

Halaman *Login* digunakan untuk halaman awal sebelum pelanggan masuk ke aplikasi dan mengakses menu – menu yang ada. Pengujian fungsional pada proses ini dapat dilihat pada Gambar 4.28.

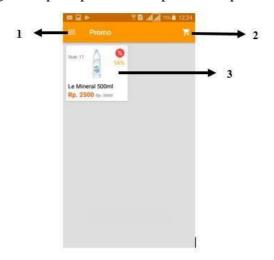


Gambar 4. 28 Tampilan pengujian halaman login

Gambar 4.28 adalah tampilan halaman *login* yang akan muncul setelah aplikasi GENmart dibuka. Pada halaman *login* terdapat 3 bagian diantaranya yaitu bagian 1 merupakan *e-mail* yang digunakan untuk memasukkan *e-mail* pelanggan yang telah terdaftar, bagian 2 merupakan *password* digunakan untuk memasukkan *password* pelanggan serta bagian 3 merupakan tombol *login* yang berfungsi untuk masuk ke dalam aplikasi apabila *e-mail* dan *password* benar.

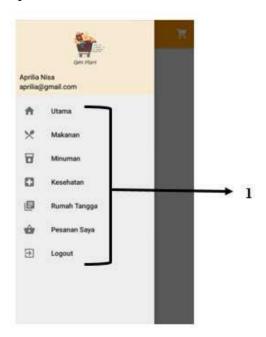
2) Menu Utama

Menu Utama digunakan untuk menampilkan produk yang sedang promo pada minimarket GENmart. Pada menu ini terdapat beberapa bagian seperti *navigasi bar*, produk yang sedang promo dan *icon* troli untuk menyimpan produk yang dipilih. Pengujian fungsional pada proses ini dapat dilihat pada Gambar 4.29.



Gambar 4. 29 Tampilan pengujian menu utama

Gambar 4.29 merupakan tampilan yang akan muncul ketika *login* berhasil. Dalam menu utama, terdapat 3 bagian yaitu *navigasi bar* yang ditunjukkan pada bagian 1, *navigasi bar* berfungsi untuk menampilkan berbagai menu yang tersedia dalam aplikasi. Bagian 2 merupakan *icon* troli yang digunakan untuk menyimpan produk yang dipilih. Bagian 3 merupakan daftar produk yang sedang promo. Untuk *navigasi bar* dapat dilihat pada Gambar 4.30.



Gambar 4. 30 Tampilan pengujian navigasi bar

Gambar 4.30 nomor 1 berfungsi untuk menampilkan menu-menu yang tersedia di dalam aplikasi, terdapat logo dari aplikasi GENmart selain itu secara otomatis akan menampilkan nama dan *e-mail* dari *username* yang telah *login* ke dalam aplikasi. Sedangkan untuk melihat detail dari produk dilakukan dengan cara klik satu kali produk yang dipilih. Untuk detail produk dapat dilihat pada Gambar 4.31.



Gambar 4. 31 Tampilan pengujian detail produk

Pada Gambar 4.31 nomer 1 merupakan detail produk atau keterangan produk yang terdiri dari nama produk, harga produk dan jumlah stok produk. Nomor 2 merupakan tombol tambah ke troli yang digunakan untuk menambahkan jumlah produk pembelian. Untuk tampilan jumlah pembelian dapat dilihat pada Gambar 4.32.



Gambar 4. 32 Tampilan pengujian jumlah pembelian

Gambar 4.32 merupakan tampilan jumlah pembelian yang akan dibeli. Setelah menambahkan jumlah pembelian produk, maka produk tersebut akan tersimpan ke dalam troli yang disimbolkan dengan *icon* troli di aplikasi. Untuk tampilan troli dapat dilihat pada Gambar 4.33.



Gambar 4. 33 Tampilan pengujian troli

Gambar 4.33 nomer 1 merupakan *icon* tempat sampah yang digunakan untuk menghapus daftar perbelanjaan. Sedangkan nomor 2 merupakan jumlah total yang perbelanjaan. Proses pembelian di aplikasi dilakukan kurang dari atau sama dengan 50 ribu rupiah, jika pembelian kurang dari 50 ribu rupiah maka tidak bisa melanjutkan ke pengiriman. Nomor 3 merupakan tombol untuk lanjutkan ke pengiriman yang berisi inputan form pengiriman. Untuk tampilan form pengiriman dapat dilihat pada Gambar 4.34.



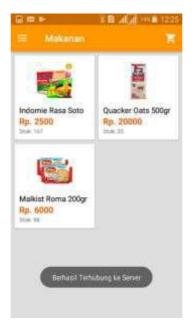
Gambar 4. 34 Tampilan *form* pengiriman

Gambar 4.34 merupakan tampilan *form* pengiriman yang berisi pilih kurir, alamat pengiriman dan telefon pelanggan. Pelanggan diharuskan mengisi form tersebut

agar produk yang sudah dipilih dapat diantar oleh kurir. Jarak yang bisa diantar oleh kurir yaitu kurang dari atau sama dengan 20 km dari lokasi toko. Setelah selesai mengisi *form* pengiriman tekan tombol kirim yang ditujukkan pada nomor 1.

3) Kategori Makanan

Kategori makanan digunakan untuk menampilkan daftar produk yang masuk ke dalam kategori makanan, semua produk makanan yang dijual ditoko berada dalam kategori makanan. Pengujian fungsional pada proses ini dapat dilihat pada Gambar 4.35.



Gambar 4. 35 Tampilan kategori makanan

Pada kategori makanan informasi mengenai makanan dapat diperoleh. Informasi tersebut berupa nama produk, harga produk dan jumlah stok produk.

4) Kategori Minuman

Kategori minuman digunakan untuk menampilkan daftar produk yang masuk ke dalam kategori minuman, semua produk minuman yang dijual ditoko berada dalam kategori minuman. Pengujian fungsional pada proses ini dapat dilihat pada Gambar 4.36.



Gambar 4. 36 Tampilan kategori minuman

Pada kategori minuman informasi mengenai minuman dapat diperoleh. Informasi tersebut berupa nama produk, harga produk dan jumlah stok produk.

5) Kategori Kesehatan

Kategori kesehatan digunakan untuk menampilakn daftar kesehatan yang masuk ke dalam kategori kesehatan, semua produk kesehatan yang dijual ditoko berada dalam kategori kesehatan. Pengujian fungsional pada proses ini dapat dilihat pada Gambar 4.37.

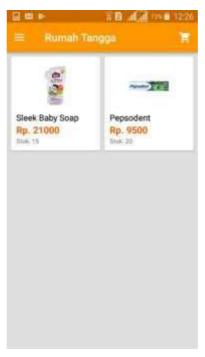


Gambar 4. 37 Tampilan kategori kesehatan

Pada kategori kesehatan informasi mengenai kesehatan dapat diperoleh. Informasi tersebut berupa nama produk, harga produk dan jumlah stok produk.

6) Kategori Rumah Tangga

Kategori rumah tangga digunakan untuk menampilan daftar produk yang masuk ke dalam kategori rumah tangga, semua produk rumah tangga yang dijual ditoko berada dalam kategori rumah tangga. Pengujian fungsional pada proses ini dapat dilihat pada Gambar 4.38.



Gambar 4. 38 Tampilan kategori rumah tangga

Pada kategori rumah tangga informasi mengenai rumah tangga dapat diperoleh. Informasi tersebut berupa nama produk, harga produk dan jumlah stok produk.

7) Menu Pesanan Saya

Menu pesanan saya digunakan untuk menampilkan barang - barang setelah melakukan transaksi pembelian produk. Pengujian fungsional pada proses ini dapat dilihat pada Gambar 4.39.



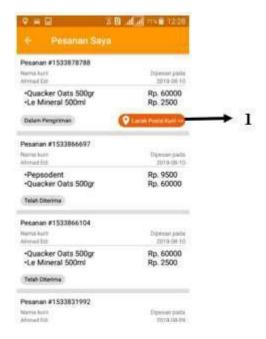
Gambar 4. 39 Menu pesanan saya

Gambar 4.39 merupakan tampilan menu pesanan saya yang berisi status pesanan produk dari pelanggan dan menampilkan data *history* pesanan. Pada Gambar 4.39 produk quaker oat dan le mineral berstatus "sedang diproses" artinya barang tersebut sedang di proses menuju aplikasi kurir yang nantinya kurir akan mendapatkan notifikasi pesanan. Lalu jika pesanan tersebut sudah diterima oleh kurir maka status dari barang akan berubah menjadi "*pickup* barang". Untuk tampilan *pickup* barang dapat dilihat pada Gambar 4.40.



Gambar 4. 40 Status pickup barang

Gambar 4.40 merupakan tampilan status *pickup* barang artinya pesanan produk sedang dikemas oleh kurir. Setelah itu produk diantar menuju rumah pelanggan yang mempunyai status "dalam pengiriman". Untuk tampilan dalam pengiriman barang dapat dilihat pada Gambar 4.41.



Gambar 4. 41 Status dalam pengiriman

Gambar 4.41 merupakan tampilan status dalam pengiriman oleh kurir. Pada proses pengiriman produk, pelanggan bisa melihat posisi kurir dengan cara menekan tombol lacak posisi kurir yang ditujukkan pada nomer 1. Untuk tampilan posisi lacak kurir dapat dilihat pada Gambar 4.42.



Gambar 4. 42 Lacak posisi kurir

Gambar 4.42 merupakan tampilan posisi lacak kurir untuk mengetahui posisi kurir. Lalu setelah produk sudah sampai ke pelanggan maka status produk akan berubah menjadi "telah diterima". Untuk tampilan status telah diterima dapat dilihat pada Gambar 4.43.



Gambar 4. 43 Status telah diterima

Gambar 4.43 merupakan tampilan status telah diterima yang berarti produk tersebut sudah diterima oleh pelanggan. Lalu setelah itu pelanggan dapat memberikan rating kepada kurir setelah menerima barang tersebut.

4.1.2.2 Android GENmart Kurir

Aplikasi GENmart kurir pada android berfungsi untuk melakukan pengiriman produk dengan menggunakan *smartphone*. Terdapat 4 buah menu yang tersedia, yaitu halaman *login*, menu pengiriman, menu *history*, dan menu map.

1) Halaman *login*

Halaman *login* digunakan untuk halaman awal sebelum kurir masuk ke aplikasi dan mengakses menu – menu yang ada. Pengujian fungsional pada proses ini dapat dilihat pada Gambar 4.44.

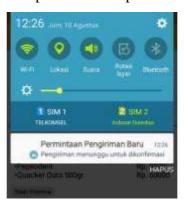


Gambar 4. 44 Tampilan pengujian halaman login

Gambar 4.44 adalah tampilan halaman *login* yang akan muncul setelah aplikasi GENmart dibuka. Pada halaman *login* terdapat 3 bagian diantaranya yaitu bagian 1 merupakan *e-mail* yang digunakan untuk memasukkan *e-mail* kurir yang telah terdaftar, bagian 2 merupakan *password* digunakan untuk memasukkan *password* kurir serta bagian 3 merupakan tombol Login yang berfungsi untuk masuk ke dalam aplikasi apabila *e-mail* dan *password* benar.

2) Menu pengiriman

Menu pengiriman digunakan untuk menampilkan proses pengiriman kurir. Jika ada pemesanan dari pelanggan maka di aplikasi kurir akan terdapat notifikasi pemesan. Untuk tampilan notifikasi pemesanan dapat dilihat pada Gambar 4.45.



Gambar 4. 45 Notifikasi kurir

Gambar 4.45 merupakan tampilan notifikasi pesanan dari pelanggan. Setelah itu kurir mengeklik notifikasi tersebut maka akan muncul status "sedang diproses". Untuk tampilan sedang diproses dapat dilihat pada Gambar 4.46.



Gambar 4. 46 Status sedang diproses

Gambar 4.46 merupakan tampilan status sedang diproses, lalu kurir menerima pesanan tersebut dengan mengklik status sedang diproses, maka akan muncul tampilan *pop up* pickup barang. Untuk tampilan *pop up* pickup barang dapat dilihat pada Gambar 4.47.



Gambar 4. 47 Pop up pickup barang

Gambar 4.47 merupakan tampilan *pop up pickup* barang setelah mengklik status sedang diproses. Nomor 1 merupakan tombol *message* yang digunakan untuk mengirim pesan ke pelanggan. Nomor 2 merupakan tombol *call* yang digunakan untuk menelfon pelanggan. Nomor 3 merupakan tombol *pick up* barang yang artinya produk tersebut akan dikemas oleh kurir. Lalu tampilan pengiriman akan berubah status menjadi sedang diproses. Untuk tampilan status *pickup* barang dapat dilihat pada Gambar 4.48.



Gambar 4. 48 Status pickup barang

Setelah kurir selesai mengkemas produk pemesanan pelanggan, maka prduk siap untuk diantar. Untuk memperbarui status pengiriman maka kurir bisa mengklik status *pickup* barang. Setelah itu akan muncul *pop up* dalam pengiriman. Untuk tampilan *pop up* dalam pengiriman dapat dilihat pada Gambar 4.49.



Gambar 4. 49 Pop up dalam pengiriman

Gambar 4.49 merupakan tampilan *pop up* dalam pengiriman setelah mengklik status *pickup* barang. Nomor 1 merupakan tombol dalam pengiriman yang artinya produk tersebut sedang dikirim oleh kurir. Lalu tampilan *pickup* barang akan berubah status menjadi dalam pengiriman. Untuk tampilan status dalam pengiriman dapat dilihat pada Gambar 4.50.



Gambar 4. 50 Status dalam pengiriman

Setelah kurir selesai mengantar produk pemesanan pelanggan, maka produk telah diterima oleh pelanggan. Untuk memperbarui status pengiriman maka kurir bisa mengklik status dalam pengiriman. Setelah itu akan muncul *pop up* telah diterima. Untuk tampilan *pop up* telah diterima dapat dilihat pada Gambar 4.51.



Gambar 4. 51 Pop up telah diterima

Gambar 4.51 merupakan tampilan *pop up* telah diterima setelah mengklik status dalam pengiriman. Nomor 1 merupakan tombol telah diterima yang artinya produk tersebut telah diterima oleh pelanggan.

3) Menu history

Didalam menu *history* digunakan untuk menampilkan barang - barang yang pernah diantar oleh kurir. Didalam halaman tersebut tampilannya sama pada menu pengiriman tetapi terdapat perbedaan status pengiriman saja, di halaman *history* status pengirimannya berubah menjadi telah diterima karena produk tersebut sudah selesai diantar oleh kurir dan diterima oleh pelanggan. Untuk tampilan menu *history* dapat dilihat pada Gambar 4.52.



Gambar 4. 52 Menu history

Gambar 4.52 merupakan tampilan menu *history* yang berisi produk yang sudah diantar oleh kurir dengan ditujukkan status "telah diterima" oleh pelanggan

4) Menu Map

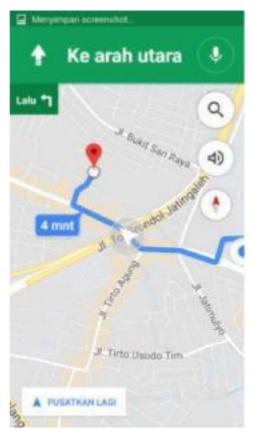
Menu map digunakan untuk menampilkan status pesanan dan digunakan untuk *tracking rute* yang disimbolkan pada tombol maps. Untuk tampilan menu map dapat dilihat pada Gambar 4.53.



Gambar 4. 53 Menu map

Pada Gambar 4.53 merupakan tampilan menu map. Sedangak nomor 1 merupakan tombol maps yang berisi *tracking rute* alamat pemesanan. Pada map ini kurir dapat

mengetahui jalan menuju alamat pemesanan. Untuk Untuk tampilan map dapat dilihat pada Gambar 4.54.



Gambar 4. 54 Tracking rute

Analisis fungsional aplikasi android GENmart juga dilakukan dengan menggunakan menggunakan sistem operasi yang berbeda yaitu melalui Kitkat, Lollipop, Nuogat. Hasil dari uji fungsional dengan 3 sistem operasi yang berbeda dapat dilihat pada tabel 4.2 dan tabel 4.3.

Tabel 4. 2 Hasil pengujian fungsional aplikasi android GENmart perbelanjaan pada sistem operasi yang berbeda

No.	Pengujian	Kitkat	Lollipop	Nougat
1.	Log in sistem	Berhasil	Berhasil	Berhasil
2.	Pengoperasian menu utama	Berhasil	Berhasil	Berhasil
3.	Pengoperasian kategori makanan	Berhasil	Berhasil	Berhasil
4.	Pengoperasian kategori minuman	Berhasil	Berhasil	Berhasil
5.	Pengoperasian kategori kesehatan	Berhasil	Berhasil	Berhasil
6.	Pengoperasian kategori rumah tangga	Berhasil	Berhasil	Berhasil

7.	Pengoperasian menu rumah pesanan	Berhasil	Berhasil	Berhasil
	saya			

Tabel 4.2 memperlihatkan hasil pengujian fungsionalitas aplikasi *delivery order* minmarket berbasis web pada aplikasi pelanggan, dan terdapat 7 fungsi yang sudah diuji pada 3 sistem operasi android yaitu Kitkat, Lollipop, dan Nougat. Dapat disimpulkan bahwa aplikasi *delivery order* minimarket berbasis andorid untuk aplikasi perbelanjaan sudah berjalan baik di beberapa jenis operasi android.

Tabel 4. 3 Hasil pengujian fungsional aplikasi android GENmart kurir pada sistem operasi yang berbeda

No.	Pengujian	Kitkat	Lollipop	Nougat
1.	Log in sistem	Berhasil	Berhasil	Berhasil
2.	Pengoperasian menu pengiriman	Berhasil	Berhasil	Berhasil
3.	Pengoperasian menu histori	Berhasil	Berhasil	Berhasil
4.	Pengoperasian menu map	Berhasil	Berhasil	Berhasil

Tabel 4.3 memperlihatkan hasil pengujian fungsionalitas aplikasi *delivery order* minmarket berbasis web pada aplikasi kurir, dan terdapat 4 fungsi yang sudah diuji pada 3 sistem operasi android yaitu Kitkat, Lollipop, dan Nougat. Dapat disimpulkan bahwa aplikasi *delivery order* minimarket berbasis andorid untuk aplikasi kurir sudah berjalan baik di beberapa jenis operasi android.

4.2 Analisis Kepuasan Pengguna

Pada tahapan ini dilakukan proses pengujian aplikasi yang sudah jadi kepada pengguna yang sebenarnya. Pengujian dilakukan pada sebuah minimarket dengan 10 responden. Minimarket tersebut adalah GENmart yang berlokasi di Jalan Dinar Asri Raya B-1, Meteseh – Tembalang. Pada pengujian ini pihak yang terlibat yaitu pemilik toko sebagai admin toko, manajer toko, kurir, dan pelanggan GENmart. Hasil kuesioner analisis kepuasan pengguna dapat dilihat pada Tabel 4.4.

Tabel 4. 4Hasil penilaian kuesioner

No	Nama	Posisi	Pertanyaan								Total		
			1	2	3	4	5	6	7	8	9	10	
1	Ramlah Nur	Admin	4	4	4	4	5	3	4	4	5	4	41
	Hamidin	Toko											

2	Tutik	Admin	4	5	4	5	4	4	3	4	4	4	41
	Rahayu	Toko											
3	Kristianti	Manajer	4	5	5	4	4	4	4	3	3	4	40
4	Yanti	Manajer	4	4	5	4	4	3	3	4	4	4	39
5	Ahmad Edi	Kurir	4	4	4	3	5	4	4	5	4	5	42
6	Maesaroh	Kurir	4	4	4	4	4	3	5	5	4	5	42
7	Shofa	Pelanggan	4	3	3	4	4	4	4	5	4	5	40
8	Sumarni	Pelanggan	4	4	3	4	4	4	4	5	4	5	41
9	Dewi	Pelanggan	4	4	4	4	3	5	5	4	4	5	42
10	Maulida	Pelanggan	4	4	4	4	4	4	4	4	4	4	40

Berikut rekap perhitungan skor menggunakan panduan Tabel 4.4.

Total nilai kepuasan pengguna = 408

Total nilai maksimal indikator penilaian = 500

Presentasi kepuasan pengguna (%) ==
$$\times 100\%$$

= $\frac{408}{81,6\%} \times 100\%$

Dari hasil kuesioner dapat diambil kesimpulan bahwa 81,6 % menyatakan bahwa aplikasi yang dibuat sesuai dengan yang diharapkan dengan indikator pengguna menunjukkan "Sangat Puas".

Berdasarkan hasil pengujian, semua fitur aplikasi *delivery order* minimarket pada fungsi kelola data di admin sudah berjalan dengan baik. Fungsi kelola data di admin meliputi kelola data *user*, produk, data transaksi, dan *stock warning*. Terdapat kesamaan fitur pada aplikasi web based dengan aplikasi layanan pesan antar pada toko makanan (Hendra Prabowo, Yulina, & Sari, 2016), aplikasi layanan pesan antar pada Vania Bakery (Antonia, Budhi, & Purba, 2016). Namun semua fitur tersebut belum memiliki fitur *stock warning* dan fitur kelola data transaksi. Sedangkan fungsi kelola data pada aplikasi *delivery order* GENmart sudah tersedia fitur *stock warning* dan kelola data transaksi.

Aplikasi pesan antar berbasis android memiliki kesamaan dengan aplikasi *delivery* order Vania bakery (Antonia, Budhi, & Purba, 2016), aplikasi layanan pesan antar rumah makan Podoteko (Purnomo Rahmat, & Nurudin Akbar, 2017), aplikasi layanan pesan antar restoran (Akbar, 2014). Namun aplikasi tersebut belum memiliki fitur promo pada halaman muka aplikasi. Sedangkan pada aplikasi delivery order GENmart minimarket produk promo ditampilkan pada halaman muka.

Fitur lacak kurir atau lokasi merupakan fitur yang dimiliki oleh pelanggan untuk melacak posisi kurir ketika sedang mengantar pesanan. Terdapat kesamaan fitur lokasi dengan aplikasi delivery order pada toko bunga (Sanjaya Kelvin, Purba Kristo, Andjarwirawan Justinus, 2014), aplikasi delivery order pada toko Wulan's cakery (Hatutik & Sediyono, 2015). Namun fitur lacak pada aplikasi tersebut belum memiliki icon kurir yang menarik. Sedangkan pada aplikasi delivery order GENmart minimarket memiliki fitur lacak kurir dengan UI yang menarik dan logo kurir menggunakan simbol motor. Fitur monitor status pesanan adalah fitur pelanggan untuk mengetahui status pesanannya. Ada 4 macam status pesanan, yang pertama adalah "sedang diproses" artinya pesanan sudah masuk berupa notifikasi di smartphone kurir, kemudian "pickup barang" artinya kurir telah menerima pesanan dan mengkonfirmasi mulai menyiapkan barang pesanan, selanjutnya adalah "dalam pengiriman" artinya kurir sudah mengirim pesanan ke alamat pelanggan, dalam status ini akan muncul fitur maps untuk mengetahui rute lokasi tujuan dan pada aplikasi pelanggan akan muncul menu lacak kurir, yang terakhir adalah "telah diterima" artinya pesanan telah diterima oleh pelanggan dan status pesanan selesai. Fitur monitor status pesanan hanya ada pada aplikasi delivery order GENmart minimarket dan belum ada di aplikasi yang sudah pernah dibuat.

BAB V KESIMPULAN

Berdasarkan hasil pembuatan Tugas Akhir "Aplikasi *Delivery Order* Minimarket Berbasis Web dan Android", maka dapat diambil kesimpulan sebagai berikut :

- 1) GENmart: Aplikasi *Delivery Order* Minimarket Berbasis Web dan Android sudah berhasil dibuat dengan aplikasi web dan android. Aplikasi web GENmart telah diuji pada tiga browser yang berbeda dengan hasil semua fungsi dapat berjalan dengan baik, Aplikasi android GENmart telah diuji pada 3 sistem operasi android yang berbeda dengan hasil semua fungsi berjalan dengan baik.
- 2) Berdasarkan hasil pengujian kepuasan pengguna 10 responden didapatkan hasil presentase tingkat kepuasan pengguna sebesar 81.6% yang berarti bahwa pengguna merasa sangat puas dengan sistem ini.

DAFTAR PUSTAKA

- Akbar, M., Satoto, K. I. and Isnanto, R. R. (2014) 'Sistem Operasi Android', *Transmisi*, 16(4), pp. 171–174.
- Antonio, M., Budhi, G. S. and Purba, K. R. (no date) 'Pembuatan Aplikasi *Delivery order* Toko Roti Vania Bakery', pp. 3–7.
- Content, M. C. and Informatika, M. (2015) '(1) APLIKASI DELIVERY MAKANAN BERBASIS WEB DI AREA', 1(1), pp. 180–185.
- Dewi, S. P. and Arnie, R. (2014) 'Aplikasi Layanan Pemesanan Katering Pada Rumah Makan Srie Berbasis Web', *JUTISI*, 3(3), pp. 579–652. Available at: sarrypurnama@gmail.com.
- Exstrilia, G., Andjarwirawan, J. and Dewi, L. P. (no date) 'Pembuatan Aplikasi *Delivery order* dengan Menggunakan HTML5 Berbasis Mobile'.
- Hastutik, P. and Sediyono, E. (2017) 'Perancangan Aplikasi *Delivery order*Berbasis Android Menggunakan Teknologi Google Maps dan GPS
 Perancangan Aplikasi *Delivery order* Berbasis Android Menggunakan
 Teknologi Google Maps dan GPS', (672013221).
- Maimunah, D. (2017) 'Aplikasi Sistem Order Online Berbasis Mobile Android Pada Outlet Pizza Hut Delivery', in *Seminar Nasional Teknologi Informasi dan Multimedia* 2017, pp. 4–5.
- Prabowo, H., Yulina, S. and Sari, R. P. (no date) 'Aplikasi Food Deliver Online Berbasis Android dan Web'.
- Purnomo, R. and Nurdin, A. (2017) 'Aplikasi layanan *delivery order* berbasis web pada rumah makan podoteko', 3(2), pp. 23–30.
- Sanjaya, K., Purba, K. R. and Andjarwirawan, J. (no date) 'Pembuatan Aplikasi *Delivery order* Bunga pada Toko X Berbasis Mobile', pp. 2–6.

LAMPIRAN

Lampiran 1 Kode Program Web

A. MODEL

1. User.php

```
<?php
namespace App;
use Illuminate\Notifications\Notifiable;
use Illuminate\Foundation\Auth\User as Authenticatable;
class User extends Authenticatable
   use Notifiable;
   protected $table = 'users';
   public $timestamps = false;
   protected $fillable = [
       'name', 'email', 'password', 'fullname', 'alamat', 'no telp',
'token', 'status'
   ];
    protected $hidden = [
        'password', 'remember_token',
    public function isAdminToko()
        if($this->admintoko==1) {
           return true;
        }return false;
    public function isManajer()
        if($this->manajer==1) {
           return true;
        }return false;
   public function isKurir()
        if($this->kurir==1) {
           return true;
        }return false;
    }
    public function isPelanggan()
        if($this->pelanggan==1) {
           return true;
        }return false;
    public function pesanan()
        return $this->hasMany('App\Pesanan', 'id user');
```

2. Produk.php

```
<?php
namespace App;
use Illuminate\Database\Eloquent\Model;</pre>
```

```
class Produk extends Model
{
    protected $primaryKey = 'id';
    protected $fillable = [
        'id', 'nama_produk', 'kategori', 'stok', 'harga_sebelum',
    'harga', 'foto', 'size'
    ];

    protected $dates = [
        'created_at',
        'updated_at',
    ];

    public function pesanan()
    {
        return $this->hasMany('App\Pesanan', 'id_produk');
    }
}
```

3. Pesanan.php

```
<?php
namespace App;
use Illuminate\Database\Eloquent\Model;
class Pesanan extends Model
{
    protected $primaryKey = 'id';
    protected $fillable = [
'id_user', 'id_kurir', 'id_produk', 'alamat_pesanan', 'telp_pesanan', 'total', 'status_pesanan'
    ];
    protected $dates = [
      'created at',
       'updated_at',
    public function produk()
        return $this->belongsTo('App\Produk', 'id');
    public function user()
        return $this->belongsTo('App\User', 'id');
```

B. VIEW

1. welcome.blade.php

```
<a href="{{ url(config('adminlte.dashboard url',
'home')) }}">{!! config('adminlte.logo', '<b>Admin</b>LTE') !!}</a>
        </div>
        <!-- /.login-logo -->
        <div class="login-box-body">
            { {
trans('adminlte::adminlte.login message') }}
            <form action="{{ url(config('adminlte.login url',</pre>
'login')) }}" method="post">
                {!! csrf field() !!}
                <div class="form-group has-feedback {{</pre>
$errors->has('email') ? 'has-error' : '' }}">
                    <input type="email" name="email"</pre>
class="form-control" value="{{ old('email') }}"
                          placeholder="{{
trans('adminlte::adminlte.email') }}">
                    <span class="glyphicon glyphicon-envelope</pre>
form-control-feedback"></span>
                    @if ($errors->has('email'))
                        <span class="help-block">
<strong>{{ $errors->first('email') }}</strong>
                        </span>
                    @endif
                </div>
                <div class="form-group has-feedback {{</pre>
$errors->has('password') ? 'has-error' : '' }}">
                    <input type="password" name="password" class="form-</pre>
control"
                           placeholder="{{
trans('adminite::adminite.password') }}">
                    <span class="glyphicon glyphicon-lock form-control-</pre>
feedback"></span>
                    @if ($errors->has('password'))
                        <span class="help-block">
                      <strong>{{ $errors->first('password') }}</strong>
                        </span>
                    @endif
                                         </div>
                <div class="row"> <div class="col-xs-4">
                        <button type="submit"</pre>
                                class="btn btn-primary btn-block btn-
flat">{{ trans('adminite::adminite.sign in')
                    <!-- /.col -->
                </div>
            </form>
        </div>
        <!-- /.login-box-body -->
    </div><!-- /.login-box -->
@stop
@section('adminite js')
    <script src="{{</pre>
asset('vendor/adminite/plugins/iCheck/icheck.min.js') }}"></script>
    <script>
        $(function () {
            $('input').iCheck({
                checkboxClass: 'icheckbox square-blue',
                radioClass: 'iradio_square-blue',
                increaseArea: '20%' // optional
        });
    </script>
    @yield('js')
@stop
```

2. admintoko (index.blade.php)

```
@extends('adminlte::page')
@section('title', 'GENmart')
@section('content header')
    <!--<h1>Admin Toko</h1>-->
@section('content')
    <div class="callout callout-info">
        <h4>Welcome !</h4>
        Selamat Datang di halaman Admin silahkan cek stok
produk secara berkala untuk menghindari kekosongan stok.
      </div>
      <!-- Transaksi -->
      <div class="row">
        <div class="col-lg-5 col-xs-6">
          <!-- small box -->
          <div class="small-box bg-green">
            <div class="inner">
             <br><br><br><br><
              <h4><b>Data Transaksi</b></h4>
            </div>
            <div class="icon">
              <i class="ion ion-bag"></i></i>
            </div>
            <a href="{{ url('transaksi') }}" class="small-box-
footer">More info <i class="fa fa-arrow-circle-right"></i></a>
          </div>
        </div>
       <!-- Stock Warning -->
       <div class="row">
        <div class="col-lg-5 col-xs-6">
          <!-- small box -->
          <div class="small-box bg-red">
            <div class="inner">
             <br><br><br><br>>
              <h4><b>Stock Warning</b></h4>
            </div>
            <div class="icon">
              <i class="ion ion-alert"></i></i>
            </div>
            <a href="{{ url('stockwarning') }}" class="small-box-</pre>
footer">More info <i class="fa fa-arrow-circle-right"></i></a>
          </div>
        </div>
```

3. admintoko (masteruser.blade.php)

```
<a href="{{ url('masteruser') }}" class="close"</pre>
data-dismiss="alert" aria-label="close">×</a>
         @endif
       @endforeach
     </div>
<!-- Search -->
       {!! Form::open(['method'=>'GET',
'url'=>'cariuser', 'role'=>'cari']) !!}
       <div class="input-group custom-search-form">
          <input type="text" class="form-control" name="cari"</pre>
placeholder="Cari User..">
              <span class="input-group-btn">
                  <button class="btn btn-default"</pre>
type="submit"><i class="fa fa-search"></i> Cari</button>
       </div>
       {!! Form::close() !!}<!--->
       <div class="table-responsive">
         \langle t.r \rangle
              No
              Nama
              Email
              No Telp
              Alamat
              Status
              Action
            </thead>
           <?php $no=1; ?>
           @foreach ($users as $user)
            { $no++ } } 
              {{$user-> name}}
              {{$user-> email}}
              \t  { \{ suser-> no telp \} } 
              {{$user-> alamat}}
              @if ( $user->admintoko == 1)
                  Admin Toko
              @elseif ( $user->manajer == 1)
                  Manajer
              @elseif ( $user->kurir == 1)
                  Kurir
              @else
                  Pelanggan
              @endif
              <form class="" action="{{ url('masteruser')}</pre>
<input type="hidden" name=" token"</pre>
value="{{ csrf token() }}">
              {{method_field('DELETE')}}
              <a href="{{ url('masteruser') }}/{{ $user->id}
}}/edit" class="btn btn-warning"><span class="fa fa-pencil" aria-</pre>
hidden="true"></span> Edit</a>
              <button class="btn btn-danger" onclick="var x</pre>
= confirm('Yakin akan menghapus data ini ?');
              if(x){return true;} else {return false;}">
<span class="fa fa-trash" aria-hidden="true"></span>
Hapus</button>
              </form>
```

4. admintoko(tambahuser.blade.php)

```
@extends('adminlte::page')
@section('title', 'GENmart')
@section('content header')
    <h1>Create User</h1><br>
@section('content')
@if(count($errors) > 0)
    <div class="alert alert-danger">
      <111>
        @foreach ($errors->all() as $error)
          {{ $error }}
        @endforeach
      </div>
@endif
<div class="panel panel-primary">
    <div class="panel-heading">Create
        User</div> <div class="panel-body">
        <form class="form-horizontal" role="form" method="POST"</pre>
action="{{ url('masteruser') }}">
              <div class="form-group{{ $errors->has('name') ? ' has-
error' : '' }}">
              <label for="name" class="col-md-1 control-</pre>
label">Name</label>
              <div class="col-md-4">
                <input id="name" type="text" class="form-control"</pre>
name="name" value="{{ old('name') }}">
                     @if ($errors->has('name'))
                         <span class="help-block">
                         <strong>{{ $errors->first('name')}
                         }}</strong> </span>
                     @endif
                </div>
            </div>
            <div class="form-group{{ $errors->has('email') ? ' has-
error' : '' }}">
                <label for="email" class="col-md-1 control-</pre>
label">Email</label>
                <div class="col-md-4">
                <input id="email" type="email" class="form-control"</pre>
name="email" value="{{ old('email') }}">
                    @if ($errors->has('email'))
                         <span class="help-block">
                         <strong>{{ $errors->first('email')}
                         }}</strong> </span>
                     @endif
                </div>
            </div>
            <div class="form-group{{ $errors->has('no_telp') ? ' has-
error' : '' }}">
                <label for="no_telp" class="col-md-1 control-</pre>
label">No.Telp</label>
                <div class="col-md-4">
                <div class="input-group">
                     <div class="input-group-addon">+62</div>
                     <input type="text" class="form-control"</pre>
id="no_telp" name="no_telp" value="{{ old('no telp') }}">
```

```
</div>
                     @if ($errors->has('no_telp'))
                         <span class="help-block">
              <strong>{{ $errors->first('no telp')}
                         }}</strong> </span>
                     @endif
                </div>
            </div>
            <div class="form-group{{ $errors->has('alamat') ? ' has-
error' : '' }}">
                <label for="alamat" class="col-md-1</pre>
control-label">Alamat</label>
                <div class="col-md-4">
                <textarea id="alamat" type="text" class="form-</pre>
control" name="alamat" value="{{ old('alamat') }}"></textarea>
                     @if ($errors->has('alamat'))
                         <span class="help-block">
                      <strong>{{ $errors->first('alamat')}
                         }}</strong> </span>
                     @endif
                </div>
            </div>
            <div class="form-group{{ $errors->has('status') ? ' has-
error' : '' }}">
                <label for="status" class="col-md-1</pre>
control-label">Status</label>
                <div class="col-md-4">
                <select class="form-control" id="status" class="form-</pre>
control" name="status">
                     <option value="">Pilih ..</option>
                     <option>Admin Toko</option>
                     <option>Manajer</option>
                     <option>Kurir</option>
                     <option>Pelanggan</option>
                </select>
                @if ($errors->has('status'))
                     <span class="help-block">
                     <strong>{{ $errors->first('status') }}</strong>
                     </span>
                @endif
                   </div>
            </div>
            <div class="form-group{{ $errors->has('password') ? ' has-
error' : '' }}">
                <label for="password" class="col-md-1 control-</pre>
label">Password</label>
                <div class="col-md-3">
                <input id="password" type="password" class="form-</pre>
control" name="password">
                @if ($errors->has('password'))
                     <span class="help-block">
     <strong>{{ $errors->first('password') }}</strong>
                                                                 </span>
                @endif
                </div>
            </div>
            <div class="form-group{{ $errors-</pre>
>has('password_confirmation') ? ' has-error' : '' }}">
                <label for="password-confirm" class="col-md-1</pre>
control-label">Confirm Password</label>
                <div class="col-md-3">
                <input id="password-confirm" type="password"</pre>
class="form-control" name="password_confirmation">
                @if ($errors->has('password confirmation'))
                     <span class="help-block">
         <strong>{{ $errors->first('password_confirmation') }}</strong>
```

```
</span>
                @endif
                </div>
            </div>
       <input type="hidden" name=" token" value="{{ csrf token() }}">
                <button type="submit" class="btn btn-primary">
                <span class="fa fa-pencil" aria-hidden="true"></span>
                    Create
                </button>
                <a href="{{ url('/masteruser') }}" type="button"
class="btn btn-danger"><span class="fa fa-close" aria-
hidden="true"></span> Cancel</a>
         </form>
    </div>
</div>
@endsection
```

5. admintoko(edituser.blade.php)

```
@extends('adminlte::page')
@section('title', 'AntarGEN')
@section('content header')
    <h1>Edit User : {{ $user->name }}</h1><br>
@st.op
@section('content')
@if(count($errors) > 0)
    <div class="alert alert-danger">
      <111>
        @foreach ($errors->all() as $error)
          {{ $error }}
        @endforeach
      </div>
@endif
<div class="panel panel-primary">
      <div class="panel-heading"></div>
       <div class="panel-body">
        <form enctype="multipart/form-data" class="form-horizontal"</pre>
role="form" method="POST" action="{{ url('masteruser') }}/{{ $user-
>id }}">
       <div class="form-group{{ $errors->has('fullname') ? ' has-
error' : '' }}">
            <label for="name" class="col-md-1 control-</pre>
label">Name</label>
            <div class="col-md-4">
            <input id="name" type="text" class="form-control"</pre>
@if ($errors->has('name'))
                    <span class="help-block">
                    <strong>{{ $errors->first('name')}
                    }}</strong> </span>
                @endif
            </div>
        </div>
<div class="form-group{{ $errors->has('email') ? ' has-error' : '' }}">
            <label for="email" class="col-md-1 control-</pre>
label">Email</label>
            <div class="col-md-4">
            <input id="email" type="email" class="form-control"</pre>
name="email" value="{{ $user->email }}">
                @if ($errors->has('email'))
                    <span class="help-block">
                    <strong>{{ $errors->first('email') }}</strong>
                    </span>
                @endif
```

```
</div>
        </div>
<div class="form-group{{ $errors->has('no_telp') ? ' has-error' :
'' } } ">
            <label for="no telp" class="col-md-1 control-</pre>
label">No.Telp</label>
            <div class="col-md-4">
            <div class="input-group">
               <div class="input-group-addon">+62</div>
               <input id="no telp" type="text" class="form-</pre>
control" name="no_telp" value="{{ $user->no_telp }}">
            </div>
               @if ($errors->has('no telp'))
                   <span class="help-block">
                   <strong>{{ $errors->first('no telp')}
                    }}</strong> </span>
               @endif
            </div>
        </div>
       <div class="form-group{{ $errors->has('alamat') ? ' has-error'
: '' }}">
           <label for="alamat" class="col-md-1 control-</pre>
label">Alamat</label>
            <div class="col-md-4">
            <textarea id="alamat" type="text" class="form-control"</pre>
@if ($errors->has('alamat'))
                   <span class="help-block">
                   <strong>{{ $errors->first('alamat') }}</strong>
                   </span>
               @endif
            </div>
        </div>
<div class="form-group{{ Serrors->has('status') ? ' has-error' : ''
<label for="status" class="col-md-1 control-</pre>
label">Status</label>
           <div class="col-md-4">
            <select class="form-control" id="status"</pre>
class="form-control" name="status">
               <option value="">Pilih..</option>
               <option>Admin Toko</option>
               <option>Manajer
               <option>Kurir</option>
               <option>Pelanggan</option>
            </select>
            @if ($errors->has('status'))
               <span class="help-block">
               <strong>{{ $errors->first('status') }}</strong>
               </span>
            @endif
              </div>
        </div>
        <br>
        <h4>Change Password (Optional) :</h4>
            <div class="form-group{{ $errors->has('password') ? ' has-
error' : '' }}">
               <label for="password" class="col-md-1 control-</pre>
label">New Password</label>
               <div class="col-md-4">
@if ($errors->has('password'))
                       <span class="help-block">
               <strong>{{ $errors->first('password') }}</strong>
```

```
</span>
                     @endif
                </div>
            </div>
            <div class="form-group{{ $errors-</pre>
>has('password_confirmation') ? ' has-error' : '' }}">
                <label for="password-confirm" class="col-md-1</pre>
control-label">Confirm New Password</label>
                <div class="col-md-4">
                <input id="password-confirm" type="password"</pre>
class="form-control" name="password confirmation">
                     @if ($errors->has('password_confirmation'))
                         <span class="help-block">
   <strong>{{ $errors->first('password confirmation')}
                         }}</strong> </span>
                     @endif
                </div>
            </div>
            <input type="hidden" name=" method" value="PUT">
   <input type="hidden" name=" token" value="{{ csrf token() }}">
             <button type="submit" class="btn btn-primary">
            <span class="fa fa-pencil" aria-</pre>
              hidden="true"></span> Save
            </button>
            <a href="{{ url('/masteruser') }}" type="button"
class="btn btn-danger"><span class="fa fa-close" aria-
hidden="true"></span> Cancel</a>
            </form>
        </div>
    </div>
@endsection
```

6. admintoko (masterproduk.blade.php)

```
@extends('adminlte::page')
@section('title', 'GENmart')
@section('content header')
   <h1><i class="fa fa-list-alt"> Manage Produk</i></h1><br> <a
   href="{{ url('masterproduk/create') }}" class="btn btn-
primary"><i class="fa fa-plus" aria-hidden="true"></i> Tambah
Product</a></h2>
@stop
@section('content')
  <div class="flash-message">
   @foreach (['danger', 'warning', 'success', 'info'] as $pesan)
      @if(Session::has('alert-' . $pesan))
       {{ Session::get('alert-' . $pesan) }}
       <a href="{{ url('masterproduk') }}" class="close"
data-dismiss="alert" aria-label="close">×</a>
      @endif
   @endforeach
  </div>
  <!-- Search -->
        {!! Form::open(['method'=>'GET', 'url'=>'cariproduk',
'role'=>'cari']) !!}
       <div class="input-group custom-search-form">
           <input type="text" class="form-control" name="cari"</pre>
placeholder="Cari User..">
                <span class="input-group-btn">
                   <button class="btn btn-default"</pre>
type="submit"><i class="fa fa-search"></i> Cari</button>
                </span>
        </div>
        {!! Form::close() !!}
```

```
<div class="table-responsive">
    <thead>
        No
         Nama Produk
         Kategori
         Stok
         Harga
         Foto
         Action
        </thead>
      <?php $no=1; ?>
      @foreach ($produks as $produk)
         {{ $no++ }}
         {{ $produk->nama_produk }}
         {{ $produk->kategori }}
         {{ $produk->stok }}
         {{ $produk->harga }}
         <!--<img src="{{ asset('/storage/app/images/produk/')
<img src="{{</pre>
url('http://antargen.informatikapolines.com/laravel/storage/app/images/
produk')}}/{{ $produk->foto }}" width="100px" heigth="100px" alt="">
         <form class="" action="{{ url('masterproduk') }}/{{
$produk->id }}" method="post">
             <input type="hidden" name="_token"</pre>
value="{{ csrf token() }}">
             {{method field('DELETE')}}
             <a href="{{ url('masterproduk') }}/{{ $produk-</pre>
>id }}/edit" class="btn btn-warning"><span class="fa fa-pencil"</pre>
aria-hidden="true"></span> Edit</a>
             <button class="btn btn-danger" onclick="var x</pre>
= confirm('Yakin akan menghapus data ini ?');
             if(x){return true;} else {return false;}">
             <span class="fa fa-trash" aria-hidden="true"></span>
Hapus</button>
             </form>
         @endforeach
      {{ $produks->links() }}
 </div>
@stop
```

7. admintoko(tambahproduk.blade.php)

```
@endforeach
      </div>
@endif
<div class="panel panel-primary">
    <div class="panel-heading">Tambah Produk</div>
        <div class="panel-body">
        <form class="form-horizontal" role="form" method="POST"</pre>
action="{{ route('masterproduk.store') }}" enctype="multipart/form-
data">
            {{ csrf field() }}
             <div class="form-group{{ $errors->has('nama produk') ? '
has-error' : '' }}">
             <label for="nama produk" class="col-md-1 control-</pre>
label">Nama Produk</label>
             <div class="col-md-4">
                <input id="nama_produk" type="text" class="form-</pre>
control" name="nama produk" value="{{ old('nama produk') }}">
                    @if ($errors->has('nama produk'))
                        <span class="help-block">
             <strong>{{ $errors->first('nama produk') }}</strong>
                        </span>
                    @endif
                </div>
            </div>
            <div class="form-group{{ $errors->has('kategori') ? ' has-
error' : '' }}">
                <label for="kategori" class="col-md-1 control-</pre>
label">Kategori</label>
                <div class="col-md-4">
                <select class="form-control" id="kategori" class="form-</pre>
control" name="kategori">
                    <option>Pilih ..
                    <option value="Makanan">Makanan
                    value="Minuman">Minumanoption
                    value="Kesehatan">Kesehatan
                    value="Kebutuhan Rumah Tangga">Kebutuhan
Rumah Tangga</option>
                    <option value="Promo">Promo</option>
                </select>
                @if ($errors->has('kategori'))
                    <span class="help-block">
                    <strong>{{ $errors->first('kategori') }}</strong>
                    </span>
                @endif
                </div>
            </div>
            <div class="form-group{{ $errors->has('stok') ? ' has-
error' : '' }}">
                <label for="stok" class="col-md-1</pre>
control-label">Stok</label>
                <div class="col-md-4">
                <input id="stok" type="stok" class="form-control"</pre>
name="stok" value="{{ old('stok') }}">
                    @if ($errors->has('stok'))
                        <span class="help-block">
                        <strong>{{ $errors->first('stok')}
                        }}</strong> </span>
                    @endif
                </div>
            </div>
            <div class="form-group{{ $errors->has('harga sebelum') ? '
has-error' : '' }}">
                    <label for="harga sebelum" class="col-md-1 control-</pre>
label">Harga Sebelum</label>
```

```
<div class="col-md-6">
                           <div class="col-md-6 input-group">
                           <div class="input-group-addon">Rp</div>
                           <input type="text" class="form-control"</pre>
id="harga sebelum" name="harga sebelum" value="{{
old('harga sebelum') }}">
                           </div>
                           @if ($errors->has('harga_sebelum'))
                           <span class="help-block">
             <strong>{{ $errors->first('harga sebelum') }}</strong>
                           </span>
                           @endif
                           </div>
              </div>
            <div class="form-group{{ $errors->has('harga') ? ' has-
error' : '' }}">
                    <label for="harga" class="col-md-1 control-</pre>
label">Harga</label>
                           <div class="col-md-6">
                           <div class="col-md-6 input-group">
                           <div class="input-group-addon">Rp</div>
                           <input type="text" class="form-control"</pre>
id="harga" name="harga" value="{{ old('harga')
                           }}"> </div>
                           @if ($errors->has('harga'))
                           <span class="help-block">
                                         <strong>{{ $errors-
>first('harga') }}</strong>
                           </span>
                           @endif
                           </div>
             </div>
                      <label>Select Image to Upload :</label>
                      <input type="file" name="foto" id="foto">
                      <hr>
        <input type="hidden" name=" token" value="{{ csrf token() }}">
                <button type="submit" class="btn btn-primary">
                <span class="fa fa-pencil" aria-hidden="true"></span>
                    Create
                </button>
                <a href="{{ url('/masterproduk') }}"
type="button" class="btn btn-danger"><span class="fa fa-close"
aria-hidden="true"></span> Cancel</a>
         </form>
    </div>
</div>
@endsection
```

8. admintoko(editproduk.blade.php)

```
<div class="panel panel-primary">
<div class="panel-heading">Edit Produk : {{ $produk->nama produk
}}</div>
        <div class="panel-body">
        <form class="form-horizontal" role="form"</pre>
method="POST" action="{{ url('masterproduk') }}/{{ $produk->id}
}}" enctype="multipart/form-data">
            {{ csrf field() }}
             <div class="form-group{{ $errors->has('nama produk') ? '
has-error' : '' }}">
             <label for="nama produk" class="col-md-1 control-</pre>
label">Nama Produk</label>
             <div class="col-md-4">
                <input id="nama_produk" type="text" class="form-</pre>
control" name="nama_produk" value="{{ $produk->nama_produk }}">
                    @if ($errors->has('nama_produk'))
                         <span class="help-block">
             <strong>{{ $errors->first('nama produk') }}</strong>
                         </span>
                    @endif
                </div>
            </div>
            <div class="form-group{{ $errors->has('kategori') ? ' has-
error' : '' }}">
                <label for="kategori" class="col-md-1 control-</pre>
label">Kategori</label>
                <div class="col-md-4">
                <select class="form-control" id="kategori" class="form-</pre>
control" name="kategori" value="{{ $produk->kategori }}>
                    <option value="">Pilih ..</option>
                    <option>Makanan
                    <option>Minuman</option>
                    <option>Kesehatan
                    <option>Kebutuhan Rumah Tangga</option>
                    <option>Promo</option>
                </select>
                @if ($errors->has('kategori'))
                    <span class="help-block">
                    <strong>{{ $errors->first('kategori') }}</strong>
                    </span>
                @endif
                </div>
            </div>
            <div class="form-group{{ $errors->has('stok') ? ' has-
error' : '' }}">
                <label for="stok" class="col-md-1</pre>
control-label">Stok</label>
                <div class="col-md-4">
                <input id="stok" type="stok" class="form-control"</pre>
name="stok" value="{{ $produk->stok }}">
                    @if ($errors->has('stok'))
                         <span class="help-block">
                        <strong>{{ $errors->first('stok')}
                         }}</strong> </span>
                    @endif
                </div>
            </div>
            <div class="form-group{{ $errors->has('harga sebelum') ? '
has-error' : '' }}">
                    <label for="harga_sebelum" class="col-md-1 control-</pre>
label">Harga Sebelum</label>
                           <div class="col-md-6">
                           <div class="col-md-6 input-group">
                           <div class="input-group-addon">Rp</div>
```

```
<input type="text" class="form-control" id="harga sebelum"</pre>
name="harga_sebelum" value="{{ $produk->harga_sebelum }}">
                           </div>
                            @if ($errors->has('harga sebelum'))
                            <span class="help-block">
             <strong>{{ $errors->first('harga sebelum') }}</strong>
                            </span>
                            @endif
                            </div>
             </div>
            <div class="form-group{{ $errors->has('harga') ? ' has-
error' : '' }}">
                    <label for="harga" class="col-md-1</pre>
control-label">Harga</label>
                            <div class="col-md-6">
                            <div class="col-md-6 input-group">
                            <div class="input-group-addon">Rp</div>
                            <input type="text" class="form-control"</pre>
id="harga" name="harga" value="{{ $produk->harga
                            }}"> </div>
                            @if ($errors->has('harga'))
                            <span class="help-block">
                     <strong>{{ \ \serrors-\rightarrors('harga') \} \rightarrors>
                            </span>
                            @endif
                            </div>
              </div>
                      <label>Select Image to Upload :</label>
 <input type="file" name="foto" id="foto" value="{{ $produk->foto }}">
                      <br>
    <input type="hidden" name=" token" value="{{ csrf token() }}">
               <button type="submit" class="btn btn-primary">
                 <span class="fa fa-pencil" aria-hidden="true"></span>
                     Update
                 </hitton>
<a href="{{ url('/masterproduk') }}" type="button" class="btn btn-</pre>
danger"><span class="fa fa-close" aria-hidden="true"></span> Cancel</a>
         </form>
    </div>
</div>
@endsection
```

9. admintoko(stockwarning.blade.php)

```
@extends('adminlte::page')
@section('title', 'GENmart')
@section('content')
<div class="callout callout-danger">
   <h4>Warning!</h4>
     Stock produk-produk dibawah ini hampir habis. Segera isi ulang
stok !
</div>
   <div class="flash-message">
       @foreach (['danger', 'warning', 'success', 'info'] as $pesan)
         @if(Session::has('alert-' . $pesan))
         {{ Session::get('alert-' . $pesan) }}
           <a href="{{ url('stockwarning') }}" class="close"
data-dismiss="alert" aria-label="close">×</a>
         @endif
       @endforeach
     </div>
<!-- Search -->
       {!! Form::open(['method'=>'GET',
'url'=>'caristock', 'role'=>'cari']) !!}
```

```
<div class="input-group custom-search-form">
           <input type="text" class="form-control" name="cari"</pre>
placeholder="Cari User..">
               <span class="input-group-btn">
                  <button class="btn btn-default"</pre>
type="submit"><i class="fa fa-search"></i> Cari</button>
               </span>
       </div>
      {!! Form::close() !!}
    <div class="table-responsive">
     <table id="table id" class="table table-striped table-
       hover"> <thead>
         \langle t.r \rangle
           No
           Nama Produk
           Kategori
           Stok
           Harga
           Foto
           Action
         </thead>
       <?php $no=1; ?>
       @foreach ($produks as $produk)
         { { $no++ } }
           {{ $produk->nama_produk }}
           {{ $produk->kategori }}
           {{ $produk->stok }}
           {{ $produk->harga }}
           >
             <!-- <img src="{{ asset('/storage/app/images/produk/')
<img src="{{
url('http://antargen.informatikapolines.com/laravel/storage/app/images/
produk')\}\}/\{\{ \ produk->foto \ \}\}" \ width="100px" \ height="100px" \ alt="">
           <form class="" action="{{ url('masterproduk') }}/{{ $produk->id}}  
} " method="post">
 <input type="hidden" name=" token" value="{{ csrf token() }}">
               {{method_field('DELETE')}}
<a href="{{ url('masterproduk') }}/{{ produk-id }}/edit"
class="btn btn-warning"><span class="fa fa-pencil" aria-
hidden="true"></span> Edit</a>
               <button class="btn btn-danger" onclick="var x</pre>
= confirm('Yakin akan menghapus data ini ?');
              if(x){return true;} else {return false;}">
<span class="fa fa-trash" aria-hidden="true"></span>
Hapus</button>
               </form>
           @endforeach
       {{ $produks->links() }}
 </div>
@stop
```

10. manajer(index.blade.php)

```
<div class="d-flex justify-content-between flex-wrap flex-md-
nowrap align-items-center pt-3 pb-2 mb-3 border-bottom">
            <hl class="h2" align="center"><i>GENmart "Belanja Mudah dan
Berkah"</i></h1>
          </div>
          <div class="jumbotron">
  <h3 class="display-4">Hello, Manajer!</h3>
  Selamat datang! Anda bisa mengakses laporan
transaksi delivery order minimarket pada halaman ini.
  <hr class="my-4">
  Untuk melihat data transaksi dapat mengklik button
Data Transaksi atau dengan klik menu Transaksi disamping.
  <a class="btn btn-primary btn-lg" href="{{url('manajer/pesanan')}}"</pre>
role="button">Data Transaksi</a>
</div>
         <!-- <canvas class="my-4 w-100" id="myChart" width="900"
height="380"></canvas> -->
        </main>
@stop
```

11. manajer(pesanan.blade.php)

```
@extends('layouts.app2')
@section('content')
<main role="main" class="col-md-9 ml-sm-auto col-lg-10 px-4">
   <br><h2>Data Transaksi</h2>
   <div class="table-responsive">
      <thead>
          \langle t.r \rangle
                No
                Pemesan
                Kurir
                Pesanan
                Alamat
                Telp Pemesan
                Total
                Status
          </thead>
         @php
                   po = 1;
                @endphp
         @foreach ($pesanans as $items)
          \langle tr \rangle
                {{ $no++ }}
                {{ $items->namapemesan }}
                {{ $items->namakurir }}
                {{ $items->nama produk }}
                {{ $items->alamat pesanan }}
                {{ $items->telp_pesanan }}
                {{ $items->total }}
                @if($items->status_pesanan == '3')
<span class="badge badge-success">Telah Diterima</span>
                @elseif($items->status_pesanan == '2')
<span class="badge badge-primary">Dalam Pengiriman</span>
                @elseif($items->status pesanan == '1')
<span class="badge badge-warning">Pickup
                Barang</span> @else
<span class="badge badge-secondary">Sedang Diproses</span>
                @endif
          @endforeach
```

12. manajer(terlaris.blade.php)

```
@extends('layouts.app2')
@section('content')
<main role="main" class="col-md-9 ml-sm-auto col-lq-10 px-4">
  <br><h2>Data Produk Terlaris</h2>
  <div class="table-responsive">
     <thead>
         No
              Produk Terlaris
              Total Penjualan
         </thead>
        @php
                 no = 1;
              @endphp
        @foreach ($pesanans as $items)
         { { no++ } }
              {{ $items->nama_produk }}
              {{ $items->count }} buah
         @endforeach
        </div>
  </main>
@endsection
```

13. laporan(harian.blade.php)

```
@extends('layouts.app2')
@section('content')
<main role="main" class="col-md-9 ml-sm-auto col-lq-10 px-4">
      <?php
      $source = $tgl;
      $date = new DateTime($source);
    <br><h2>Laporan Harian {{$date->format('d M Y')}}</h2>
    <div class="row" style="padding-left:15px;padding-
bottom:15px;"> {{-- Form pencarian harian --}}
        <form class="form-inline my-2 my-lg-0 mr-lg-2"</pre>
action="{{ route('laporan.harian') }}" method="GET">
            <div class="input-group">
                <input class="form-control" name="term" type="date"</pre>
placeholder="Search">
                <span class="input-group-append">
                                  <button class="btn btn-default"</pre>
type="submit">CARI</button>
                </span>
            </div>
             </form>
             {{-- / End Form pencarian harian --}}
    <div class="table-responsive">
        <thead>
```

```
No
               Pemesan
               Kurir
               Pesanan
               Alamat
               Telp Pemesan
               Status
               Total
          </thead>
         @php
                     no=1;
                    totalsemua = 0;
               @endphp
         @foreach ($laporans as $items)
            {{ $no++ }}
               {{ $items->namapemesan }}
               {{ $items->namakurir }}
               \t  { { $items->nama produk } } 
               {{ $items->alamat pesanan }}
               {{ $items->telp pesanan }}
               @if($items->status_pesanan == '3')
<span class="badge badge-success">Telah Diterima</span>
               @elseif($items->status_pesanan == '2')
<span class="badge badge-primary">Dalam Pengiriman</span>
               @elseif($items->status_pesanan == '1')
<span class="badge badge-warning">Pickup
               Barang</span> @else
<span class="badge badge-secondary">Sedang Diproses</span>
               @endif
Rp.{{number_format($items->total)}}
                     @php
                     $totalsemua += $items->total;
               @endphp
               @endforeach
          <b>Jumlah Total</b>
               <b>Rp.{{number format($totalsemua)}}</b>
          <a href="{{route('laporan.harianpdf', $tgl)}}" class="btn btn-</pre>
danger btn-sm"><span class="fa fa-file">&nbsp;Download Pdf</span></a>
</main>
@endsection
```

14. laporan(harian-pdf.blade.php)

```
width: 100%;
              border-collapse: collapse;
           td,th {
              text-align: left;
              padding: 5px;
              border:1px solid;
     </style>
</head>
<body style="font-size: 12px;">
     <img src="{{ url('logo.png') }}" alt="Logo" style="height:</pre>
80px; width: 150px; float: left; margin-top: -15px;">
     <center><h1 style="margin-right: 75px;">GENmart <i>"Belanja
Mudah dan Berkah"</i></hl></center>
      <center>
          <h2>Jalan Dinar Asri Raya
B-1, Meteseh-Tembalang</h2>
      </center>
______
           <?php
          $myDateTime = new DateTime($tgl);
          $newDateString = $myDateTime->format('d M
          Y'); $tanggal = $myDateTime->format('d M Y');
           <br><br><center>
          <br/>
<br/>b>LAPORAN TRANSAKSI PESANAN GENmart
TANGGAL {{strtoupper($newDateString)}}</b>
      </center>
           <hr>>
           <thead>
               >
                No
                Pemesan
                Kurir
                Pesanan
                Alamat
                Telp Pemesan
                Total
               </thead>
            @php
                totalsemua = 0;
             @endphp
             @foreach ($laporan as $items)
             \langle t.r \rangle
                {{ $no++ }}
                {{ $items->namapemesan }} {{
                $items->namakurir }} {{ $items-
                >nama produk }} {{ $items-
                >alamat_pesanan }} {{ $items-
                >telp_pesanan }}
                Rp.{{number format($items->total)}}
           @php
                         $totalsemua += $items->total;
             @endphp
             @endforeach
```

```
<b>Jumlah Total</b>
   <b>Rp.{{number format($totalsemua)}}</b>
            <br><br><
  Semarang, {{$tanggal}}<br>
  Manajer GENmart,
  <br>
  <br>
  <br>
  \langle br \rangle
  <hr>>
 (.....)
</body>
</html>
```

15. laporan(bulanan.blade.php)

```
@extends('layouts.app2')
@section('content')
<main role="main" class="col-md-9 ml-sm-auto col-lg-10 px-4">
     <?php
     $source = $tgl;
     $date = new DateTime($source);
   2>
   <br><h2>Laporan Bulan {{$date->format('M Y')}}</h2>
   <div class="row" style="padding-left:15px;padding-bottom:15px;">
           {{-- Form pencarian harian --}}
       <form class="form-inline my-2 my-lg-0 mr-lg-2"</pre>
action="{{ route('laporan.bulanan') }}" method="GET">
          <div class="input-group">
              <input class="form-control" name="term"</pre>
type="month" placeholder="Search">
             <span class="input-group-append">
                             <button class="btn btn-default"</pre>
type="submit">CARI</button>
              </span>
          </div>
           </form>
           {{-- / End Form pencarian harian --}}
   </div>
   <div class="table-responsive">
      <thead>
           No
                 Pemesan
                 Kurir
                 Pesanan
                 Alamat
                 Telp Pemesan
                 Status
                 Total
                 </thead>
          @php
                       no=1;
                       $totalsemua = 0;
                 @endphp
```

```
@foreach ($laporans as $items)
            {{ $no++ }}
                {{ $items->namapemesan }}
                {{ $items->namakurir }}
                {{ $items->nama produk }}
                {{ $items->alamat_pesanan }}
                {{ $items->telp_pesanan }}
                @if($items->status pesanan == '3')
<span class="badge badge-success">Telah Diterima</span>
                @elseif($items->status_pesanan == '2')
<span class="badge badge-primary">Dalam Pengiriman</span>
                @elseif($items->status_pesanan == '1')
<span class="badge badge-warning">Pickup Barang</span>
                @else
<span class="badge badge-secondary">Sedang
                Diproses</span> @endif
Rp. {{number format($items-
          >total) }  
                @php
                     $totalsemua += $items->total;
                @endphp
                @endforeach
          <t.r>
                <b>Jumlah Total</b>
                <b>Rp.{{number_format($totalsemua)}}</b>
          <a href="{{route('laporan.bulananpdf', $tgl)}}" class="btn btn-</pre>
danger btn-sm"><span class="fa fa-file">&nbsp;Download Pdf</span></a>
</main>
@endsection
```

16. laporan(bulanan-pdf.blade.php)

```
<!DOCTYPE html>
<html lang="en">
<head>
       <meta charset="UTF-8">
       <title>Laporan Transaksi Bulanan</title>
       <style>
              table {
                    width: 100%;
                  border-collapse: collapse;
              td,th {
                  text-align: left;
                  padding: 5px;
                  border:1px solid;
       </style>
</head>
<body style="font-size: 12px;">
       <img src="{{ url('logo.png') }}" alt="Logo" style="height:</pre>
80px; width: 150px; float: left; margin-top: -15px;">
       <center><h1 style="margin-right: 75px;">GENmart <i>"Belanja
Mudah dan Berkah"</i></hl></center>
        <center>
```

```
<h2>Jalan Dinar Asri Raya
B-1, Meteseh-Tembalang</h2>
     </center>
______
        $myDateTime = new DateTime($tql);
        $newDateString = $myDateTime->format('M Y');
        $tanggal = $myDateTime->format('d M Y');
         <br><br><center>
        <br/> <b>LAPORAN TRANSAKSI PESANAN GENmart BULAN
{{strtoupper($newDateString)}}</b>
     </center>
         <hr>>
         <thead>
            No
             Pemesan
             Kurir
             Pesanan
             Alamat
             Telp Pemesan
             Total
             </thead>
          @php
             no=1;
             $totalsemua = 0;
           @endphp
           @foreach ($laporan as $items)
           {{ $no++ }}
             {{ $items->namapemesan }} {{
             $items->namakurir }} {{ $items-
             >nama produk } }  {{ $items-
             >alamat_pesanan }} {{ $items-
             >telp pesanan }}
             Rp.{{number_format($items->total)}}
         @php
                     $totalsemua += $items->total;
           @endphp
           @endforeach
           <b>Jumlah Total</b>
             <b>Rp.{{number format($totalsemua)}}</b>
         <hr><hr><hr>>
  Semarang, {{$tanggal}}<br>
  Manajer GENmart,
  <hr>
  <br>
  <hr>
  <br>
```

C. CONTROLLER

1. HomeController.php

```
<?php
namespace App\Http\Controllers;
use Illuminate\Http\Request;
class HomeController extends Controller
{
    public function __construct()
    {
        $this->middleware('user');
    }
public function index()
    {
        return view('home');
    }
}
```

2. AdminTokoController.php

```
<?php
namespace App\Http\Controllers;
use Illuminate\Http\Request;
use App\Pesanan;
use App\Produk;
use App\User;
use App\Kurir;
use DB;
use Carbon;
use Auth;
class AdminTokoController extends Controller
    public function construct()
        $this->middleware('admintoko');
    public function index()
        return view('admintoko.index',
array('admintoko'=> Auth::admintoko()));
    public function pesanan()
        $pesanans = DB::table('pesanans')
            ->join('users', 'pesanans.id user', '=', 'users.id')
            ->join('users as kurir', 'pesanans.id_kurir', '=',
'kurir.id')
            ->join('produks', 'pesanans.id_produk', '=', 'produks.id')
            ->select('pesanans.*', 'users.name as namapemesan',
'kurir.name as namakurir', 'produks.nama produk')
            ->paginate(10);
        return view('admintoko.pesanan', compact('pesanans', 'users',
'produks'));
   }
    public function destroy(Request $request, $id pesanan)
```

```
$pesanans = Pesanan::where('id pesanan', $id pesanan);
        $pesanans->delete();
        $request->session()->flash('alert-danger', 'Data has
been deleted!');
        return redirect('/transaksi');
    public function search(Request $request)
        $cari = $request->get('cari');
        // $pesanans = Pesanan::where('nama_produk',
'LIKE', '%'.$cari.'%')->paginate(5);
        $pesanans = DB::table('pesanans')
             ->join('users', 'pesanans.id_user', '=', 'users.id')
             ->join('users as kurir', 'pesanans.id_kurir', '=',
'kurir.id')
             ->join('produks', 'pesanans.id_produk', '=', 'produks.id') -
>select('pesanans.*', 'users.name as namapemesan',
'kurir.name as namakurir', 'produks.nama_produk') -
             >where('kurir.name', 'LIKE', '%'.$cari.'%') -
             >orWhere('users.name', 'LIKE', '%'.$cari.'%')
             ->paginate(10);
        return view('admintoko.pesanan', compact('produks',
'users', 'pesanans'));
   }
```

3. MasterProdukController.php

```
<?php
namespace App\Http\Controllers;
use Illuminate\Http\Request;
use Illuminate\Support\Facades\Storage;
use App\Produk;
use App\Kategori;
use Auth:
use DB;
use Carbon;
class MasterProdukController extends Controller
    public function construct()
    {
        $this->middleware('admintoko');
    }
    public function index(Request $request)
        $produks = Produk::all();
        $produks = \App\Produk::paginate(5);
        return view('admintoko.masterproduk', compact('produks'));
    public function create()
        return view('admintoko.tambahproduk', compact('produk'));
    public function store(Request $request)
         $this->validate($request, [
            'nama_produk' => 'required',
            'stok' => 'required',
            'harga' => 'required',
```

```
]);
        $size = $request->foto->getClientSize();
        $produk = new Produk;
        $produk->nama produk = $request->nama produk;
        $produk->kategori = $request->kategori;
        $produk->stok = $request->stok;
        $produk->harga sebelum = $request->harga sebelum;
        $produk->harga = $request->harga;
        $request->foto->store('/images/produk');
        $path = $request->foto->hashName();
        $produk->foto = $path;
        $produk->size = $size;
        $produk->save();
        $request->session()->flash('alert-success', 'Produk
berhasil ditambahkan !');
        return redirect('/masterproduk');
    public function edit($id)
        $produk = Produk::where('id',$id)->first();
        if(!$produk){
            abort (503);
        return view('admintoko.editproduk', compact('produk'));
    public function update(Request $request, $id)
        $this->validate($request, [
            'nama_produk' => 'required',
            'stok' => 'required',
            'harga' => 'required',
            ]);
        $size = $request->foto->getClientSize();
        $produk = Produk::where('id', $id)->first();
        // $produk->$request->all();
        $produk->nama_produk = $request->nama_produk;
        $produk->kategori = $request->kategori;
        $produk->stok = $request->stok;
        $produk->harga sebelum = $request->harga sebelum;
        $produk->harga = $request->harga;
        if($request->hasFile('foto'))
            $file = $request->file('foto');
            $fileName = $file->getClientOriginalName();
            $destinationPath =
public path().'/storage/app/images/produk';
            $file->move($destinationPath,$fileName);
            $produk->foto = $fileName ;
        }
        $produk->save();
        $request->session()->flash('alert-success', 'Produk
berhasil diupdate !');
        return redirect('/masterproduk');
    public function destroy(Request $request, $id)
        $produks = Produk::where('id', $id);
```

```
$produks->delete();
        $request->session()->flash('alert-danger', 'Data has
been deleted!');
        return redirect('/masterproduk');
    public function stockwarning()
        $produk = DB::table('produks')
                    ->where('stok', '<=', 5)
                    ->paginate(10);
        return view('admintoko.stockwarning')->with('produks',
$produk);
    public function search(Request $request)
        $cari = $request->get('cari');
        $produks = Produk::where('nama produk', 'LIKE', '%'.$cari.'%')-
>paginate(5);
        return view('admintoko.masterproduk', compact('produks'));
    public function searchstock(Request $request)
        $cari = $request->get('cari');
        $produks = Produk::where('nama_produk', 'LIKE', '%'.$cari.'%')
                    ->orWhere('stok', '<=', 5)
                    ->paginate(5);
        return view('admintoko.stockwarning', compact('produks'));
    }
```

4. MasterUserController.php

```
<?php
namespace App\Http\Controllers;
use Illuminate\Http\Request;
use App\User;
use App\Http\Requests;
class MasterUserController extends
Controller {
   public function construct()
        $this->middleware('admintoko');
    public function index()
        $users = User::all();
        $users = \App\User::paginate(5);
        return view('admintoko.masteruser', compact('users'));
    public function create()
        return view('admintoko.tambahuser', compact('users'));
    public function store(Request $request)
        $this->validate($request, [
            'name' => 'required|max:30|unique:users,name',
            'email' => 'required',
            'no telp' => 'required',
```

```
'alamat' => 'required',
            'password' => 'required|min:6|confirmed',
            'status' => 'required',
        ]);
        $user = new User;
        $user->name = $request->name;
        $user->email = $request->email;
        $user->no telp = $request->no telp;
        $user->alamat = $request->alamat;
        if ($request->status == 'Admin Toko') {
            $user->admintoko = 1;
            suser->manajer = 0;
            suser->kurir = 0;
            $user->pelanggan = 0;
        elseif ($request->status == 'Manajer') {
            $user->admintoko = 0;
            $user->manajer = 1;
            suser->kurir = 0;
            $user->pelanggan = 0;
        elseif ($request->status == 'Kurir') {
           $user->admintoko = 0;
            suser->manajer = 0;
            suser->kurir = 1;
            $user->pelanggan = 0;
        elseif ($request->status == 'Pelanggan')
            { $user->admintoko = 0; $user-
            >manajer = 0;
            suser->kurir = 0;
            $user->pelanggan = 1;
        else {
            $user->admintoko = 0;
            $user->manajer = 0;
            suser->kurir = 0;
            $user->pelanggan = 0;
        $user->password=bcrypt($request->password);
       $request->session()->flash('alert-success', 'User has been
created!');
       return redirect('/masteruser');
   public function edit($id)
        $user = User::where('id',$id)->first();
       if(!$user){
            abort (503);
        return view('admintoko.edituser', compact('user'));
   public function update(Request $request, $id)
        $user = User::where('id',$id)->first();
        $user->name = $request->name;
        $user->email = $request->email;
        $user->no_telp = $request->no_telp;
        $user->alamat = $request->alamat;
        $user->status = $request->status;
        if ($request->status == 'Admin Toko') {
```

```
suser->admintoko = 1;
            suser->manajer = 0;
            suser->kurir = 0;
            $user->pelanggan = 0;
        elseif ($request->status == 'Manajer') {
            $user->admintoko = 0;
            $user->manajer = 1;
            suser->kurir = 0;
            $user->pelanggan = 0;
        elseif ($request->status == 'Kurir') {
            $user->admintoko = 0;
            $user->manajer = 0;
            suser->kurir = 1;
            $user->pelanggan = 0;
        elseif ($request->status == 'Pelanggan')
            { $user->admintoko = 0; $user-
            >manajer = 0;
            suser->kurir = 0;
            $user->pelanggan = 1;
        else {
            $user->admintoko = 0;
            suser->manajer = 0;
            suser->kurir = 0;
            $user->pelanggan = 0;
        $user->password=bcrypt($request->password);
        $user->save();
        $request->session()->flash('alert-success', 'Data has
ben changed!');
       return redirect('/masteruser');
    public function destroy (Request $request, $id)
        $user = User::where('id', $id);
        $user->delete();
        $request->session()->flash('alert-danger', 'Data has
been deleted!');
        return redirect('/masteruser');
    public function search(Request $request)
        $cari = $request->get('cari');
        $users = User::where('name', 'LIKE',
'%'.$cari.'%')->paginate(5);
        return view('admintoko.masteruser', compact('users'));
    }
```

5. ProdukController.php

```
<?php
namespace App\Http\Controllers;
use Illuminate\Http\Request;
use App\Produk;
use Illuminate\Support\Facades\DB;

class ProdukController extends Controller
{</pre>
```

```
public function index()
    $produks = Produk::orderBy('stok',
                'desc') ->where('stok', '!=',
                0) ->get();
    return response()->json([
    'produks' => $produks
], 200);
public function store(Request $request)
    $produk = new Produk;
    $produk->nama_produk = $request-
    >nama produk; $produk->kategori = $request-
    >kategori; $produk->stok = $request->stok;
    $produk->harga = $request->harga;
    $produk->harga sebelum = $request-
    >harga_sebelum; //Harus koneksi ke internet
    $produk->foto = '';
    $produk->size = '';
    $produk->save();
    return response()->json([
        'pesan' => 'Data berhasil disimpan'
    ], 200);
}
public function show($id)
    $produk = Produk::find($id);
    return response()->json([
    'pesan' => 'berhasil',
    'produk' => $produk
], 200);
public function update(Request $request, $id)
    $produk = Produk::find($id);
    produk->nama_produk = produk-
    >nama produk; $produk->kategori = $request-
    >kategori; $produk->stok = $request->stok;
    $produk->harga = $request->harga;
    $produk->harga sebelum = $request-
    >harga_sebelum; //Harus koneksi ke internet
$produk->foto = '';
    $produk->size = '';
    $produk->save();
    return response()->json([
        'pesan' => 'Data berhasil diubah'
    1, 200);
}
public function destroy($id)
    $produk = Produk::find($id);
    $produk->delete();
    return response()->json([
        'pesan' => 'Data berhasil dihapus'
    ]);
}
public function makanan()
```

```
$produks = DB::table('produks')->where('kategori',
'makanan') ->where('stok', '>', 0) ->select('nama_produk', 'kategori',
'stok', 'harga', 'id', 'foto') ->limit(10) ->get();
        return response()->json([
             'pesan' => 'Data Makanan',
             'produks' => $produks], 200);
    public function minuman()
        $produks = DB::table('produks')->where('kategori',
'minuman')->where('stok', '>', 0)->select('nama_produk', 'kategori',
'stok', 'harga', 'id', 'foto')->limit(10)->get();
        return response()->json([
             'pesan' => 'Data Minuman',
             'produks' => $produks], 200);
    public function kesehatan()
        $produks = DB::table('produks')->where('kategori',
'kesehatan')->where('stok', '>', 0)->select('nama_produk',
'kategori', 'stok', 'harga', 'id', 'foto')->limit(10)->get();
        return response()->json([
             'pesan' => 'Data Produk Kesehatan',
             'produks' => $produks], 200);
    public function krt()
        $produks = DB::table('produks')->where('kategori', 'kebutuhan
rumah tangga')->where('stok', '>', 0)->select('nama_produk', 'kategori',
'stok', 'harga', 'id', 'foto')->limit(10)->get();
        return response()->json([
             'pesan' => 'Data Kebutuhan Rumah Tangga',
             'produks' => $produks], 200);
    public function promo()
        // $produks = Produk::where('kategori', 'promo')-
>where('stok', '<>', 0)->select('nama produk', 'kategori', 'stok',
'harga', 'harga_sebelum', 'id', 'foto')->limit(10)->get();
        $produks = DB::table('produks')->where('kategori',
'promo')->where('stok', '>', 0)->select('nama produk', 'kategori',
'stok', 'harga', 'harga_sebelum', 'id', 'foto')->limit(10)->get();
        return response()->json([ 'pesan'
            => 'Data Produk Promo',
             'produks' => $produks], 200);
```

6. UserController.php

```
<?php
namespace App\Http\Controllers;
use Illuminate\Http\Request;
use App\User;
use App\Transformers\UserTransformer;
use Auth;

class UserController extends Controller
{
    public function index()
    {
        $users = User::all();
}</pre>
```

```
return response()->json([
    'pesan' => 'berhasil',
    'users' => $users
], 200);
public function show($id)
    $user = User::find($id);
   return response()->json([
    'pesan' => 'berhasil',
'user' => $user
], 200);
public function users(User $user)
    $users = $user->all();
    return fractal()
        ->collection($users)
        ->transformWith(new UserTransformer)
        ->toArray();
}
public function getAllKurir(){
    $users = User::where('kurir', 1)->get();
    return fractal()
        ->collection($users)
        ->transformWith(new UserTransformer)
        ->toArray();
```

7. ManajerController.php

```
<?php
namespace App\Http\Controllers;
use Illuminate\Http\Request;
use App\Pesanan;
use App\Produk;
use App\User;
use App\Kurir;
use DB;
use Carbon;
use PDF;
use Auth;
class ManajerController extends Controller
    public function construct()
        $this->middleware('manajer');
    public function index()
       return view('manajer.index', array('manajer'=>
Auth::manajer());
   public function pesanan()
        $pesanans = DB::table('pesanans')
            ->join('users', 'pesanans.id user', '=', 'users.id')
```

```
->join('users as kurir', 'pesanans.id kurir', '=',
'kurir.id')
            ->join('produks', 'pesanans.id_produk', '=', 'produks.id')
            ->select('pesanans.*', 'users.name as namapemesan',
'kurir.name as namakurir', 'produks.nama_produk')
            ->paginate(15);
        // $pesanans = \App\Pesanan::paginate(15);
       return view('manajer.pesanan', compact('pesanans'));
   public function terlaris()
        $pesanans = DB::select(DB::raw(
            "SELECT a.id_produk, b.nama_produk, count(a.id_produk) AS
count FROM pesanans as a
            JOIN produks as b ON a.id_produk = b.id GROUP BY
id produk HAVING count > 1 ORDER BY count DESC"));
        // $pesanans = \App\Pesanan::paginate(15);
       return view('manajer.terlaris', compact('pesanans'));
   public function harian(Request $request)
        if (($tgl = $request->get('term'))) {
            $tgl = $request->get('term');
            $laporan = DB::table('pesanans')
                ->join('users', 'pesanans.id user', '=', 'users.id')
                ->join('users as kurir', 'pesanans.id kurir', '=',
'kurir.id')
                ->join('produks', 'pesanans.id produk', '=',
'produks.id')
                ->select('pesanans.*', 'users.name as namapemesan',
'kurir.name as namakurir', 'produks.nama_produk')
                ->whereRaw('pesanans.created at LIKE "%' . $tgl . '%"')
                ->get();
        } else {
            $tql = date('Y-m-d');
            $laporan = DB::table('pesanans')
                ->join('users', 'pesanans.id_user', '=', 'users.id')
                ->join('users as kurir', 'pesanans.id kurir', '=',
'kurir.id')
                ->join('produks', 'pesanans.id_produk', '=',
'produks.id')
                ->select('pesanans.*', 'users.name as namapemesan',
'kurir.name as namakurir', 'produks.nama produk')
                ->whereRaw('pesanans.created at LIKE "%' . $tgl . '%"')
                ->get();
       return view('laporan.harian', ['tql' => $tql, 'laporans' =>
$laporan]);
   }
   public function harianpdf($id pesanan)
        $laporan = DB::table('pesanans')
            ->join('users', 'pesanans.id user', '=', 'users.id')
            ->join('users as kurir', 'pesanans.id kurir', '=',
'kurir.id')
            ->join('produks', 'pesanans.id_produk', '=', 'produks.id')
            ->select('pesanans.*', 'users.name as namapemesan',
'kurir.name as namakurir', 'produks.nama_produk')
            ->whereRaw('pesanans.created at LIKE "%' . $id pesanan .
18"1)
            ->get();
```

```
$pdf = PDF::loadView('laporan.harian-pdf', ['laporan' =>
$laporan, 'tgl' => $id pesanan]);
       return $pdf->download('laporan-harian.pdf');
   public function bulanan(Request $request)
        if (($tql = $request->get('term'))) {
            $tgl = $request->get('term');
            $laporan = DB::table('pesanans')
                ->join('users', 'pesanans.id_user', '=', 'users.id')
                ->join('users as kurir', 'pesanans.id kurir', '=',
'kurir.id')
                ->join('produks', 'pesanans.id produk', '=',
'produks.id')
                ->select('pesanans.*', 'users.name as namapemesan',
'kurir.name as namakurir', 'produks.nama produk')
                ->whereRaw('pesanans.created_at LIKE "%' . $tgl . '%"')
                ->get();
        } else {
            t= date('Y-m');
            $laporan = DB::table('pesanans')
                ->join('users', 'pesanans.id_user', '=', 'users.id')
                ->join('users as kurir', 'pesanans.id kurir', '=',
'kurir.id')
                ->join('produks', 'pesanans.id produk', '=',
'produks.id')
                ->select('pesanans.*', 'users.name as namapemesan',
'kurir.name as namakurir', 'produks.nama produk')
                ->whereRaw('pesanans.created_at LIKE "%' . $tgl . '%"')
       return view('laporan.bulanan', ['tgl' => $tgl, 'laporans'
=> $laporan]);
   public function bulananpdf($id pesanan)
        $laporan = DB::table('pesanans')
            ->join('users', 'pesanans.id_user', '=', 'users.id')
            ->join('users as kurir', 'pesanans.id_kurir', '=',
'kurir.id')
            ->join('produks', 'pesanans.id_produk', '=', 'produks.id')
            ->select('pesanans.*', 'users.name as namapemesan',
'kurir.name as namakurir', 'produks.nama produk')
            ->whereRaw('pesanans.created_at LIKE "%' . $id pesanan .
1811)
            ->get();
        $pdf = PDF::loadView('laporan.bulanan-pdf', ['laporan'
=> $laporan, 'tgl' => $id pesanan]);
        return $pdf->download('laporan-bulanan.pdf');
```

PesananController.php

```
<?php
namespace App\Http\Controllers;
use Illuminate\Http\Request;
use App\Pesanan;
use App\Produk;
use App\User;
use App\User;
use App\Kurir;
use App\Transformers\UserTransformer;
use DB;
use Carbon;</pre>
```

```
use PDF;
use Auth;
class PesananController extends Controller
    function sendNotification($target)
        #API access key from Google API's Console
        define('FIREBASE API KEY', "AAAAPYF231w:APA91bHRKcvGLpPcIvalP-
2C-9Qd6gpP1fqTg5FujvIHnRkjQRfI6Aao-
KPqOSolhDo1b3TgRI GImT3P3bxHgb MBEoTehDE7rD1ETOL1HYGupAPuZu1rdHBSaXvQmo
fHWKELXcxhZpFjrscbjyJUraJaIOcaKYEQ");
        $data = array
            'title' => "Pengiriman Baru",
             'judul pengumuman' => "Permintaan pengiriman baru",
        );
        $fields = array
                        => '/topics/' . $target,
             'to'
            'data' => $data
        $headers = array
            'Authorization: key=' . FIREBASE API KEY,
            'Content-Type: application/json'
        );
        #Send Reponse To FireBase Server
        $ch = curl init();
        curl setopt ($ch, CURLOPT URL,
'https://fcm.googleapis.com/fcm/send');
        curl_setopt( $ch,CURLOPT_POST, true );
        curl_setopt( $ch,CURLOPT_HTTPHEADER, $headers );
curl_setopt( $ch,CURLOPT_RETURNTRANSFER, true );
        curl setopt( $ch,CURLOPT SSL VERIFYPEER, false );
        curl_setopt( $ch,CURLOPT_POSTFIELDS, json_encode( $fields ) );
        $result = curl_exec($ch);
        curl close( $ch );
    }
    public function construct()
        if (request()->segment(1) != "api") {
            $this->middleware('admintransaksi');
    public function index()
        $pesanans = DB::table('pesanans')
            ->join('users as pelanggan', 'pesanans.id_user',
'=', 'pelanggan.id')
            ->join('users as kurir', 'pesanans.id_kurir', '=',
'kurir.id')
            ->join('produks', 'pesanans.id_produk', '=', 'produks.id') -
            >select('pesanans.*', 'pelanggan.name as nama pelanggan',
'kurir.name as nama kurir', 'produks.nama produk')
            ->orderBy('updated at', 'desc')
```

```
->where('id user', $id)
            ->get();
       return view('admintransaksi.pesanan', compact('pesanan',
'user', 'produk', 'kurir'));
   }
   public function create()
   public function troli(Request $request)
       $pesanan = new Pesanan;
       $pesanan->id produk = $request->id produk;
       $pesanan->total = $request->total;
       $produk = Produk::find($request->id produk);
       $produk->stok = $produk->stok - $request->jumlah;
       $produk->save();
       return response()->json([
            'pesan' => 'Barang masuk ke troli'
            ], 200);
   public function untroli(Request $request)
       $pesanan = new Pesanan;
       $pesanan->id_produk = $request->id produk;
       $pesanan->total = $request->total;
       $produk = Produk::find($request->id produk);
       $produk->stok = $produk->stok + $request->jumlah;
       $produk->save();
       return response()->json([
            'pesan' => 'Barang dihapus dari troli'
            ], 200);
   public function store(Request $request)
       $pesanan = new Pesanan;
       $pesanan->id_user = $request->id_user;
        $pesanan->id kurir = $request->id kurir;
       $pesanan->id produk = $request->id produk;
       $pesanan->alamat pesanan = $request->alamat pesanan;
       $pesanan->telp pesanan = $request->telp pesanan;
       $pesanan->total = $request->total;
       $pesanan->status_pesanan = $request->status_pesanan;
        if($pesanan->save()){
            $produk = Produk::find($request->id produk);
            $produk->stok = $produk->stok - $request->jumlah;
            $produk->save();
            $this->sendNotification("kurir-" . $request->id_kurir);
            return response()->json([
                'status' => '1',
```

```
'pesan' => 'Pesanan berhasil ditambahkan'
                ], 200);
        return response()->json([
             'status' => '0',
             'pesan' => 'Pesanan gagal ditambahkan'
            ], 200);
    public function edit($id)
    {
    public function update(Request $request, $id)
        //
    public function destroy($id)
        //
    public function getPesanansByUser($id)
        $pesanans = DB::table('pesanans')
            ->where('id user', $id)
            ->orderBy('created at', 'desc')
            ->distinct()
            ->get(['created_at']);
        $json = array();
        $id_pesanan_arr = array();
        $produk arr = array();
        $total arr = array();
        $jumlah arr = array();
        for (\$i=0; \$i<count(\$pesanans); \$i++) {
            $created at = $pesanans[$i]->created at;
            $pesanan = DB::table('pesanans')
                ->join('users as pelanggan', 'pesanans.id_user', '=',
'pelanggan.id')
                ->join('users as kurir', 'pesanans.id_kurir', '=',
'kurir.id')
// ->join('users as alamat_pesanan',
'pesanans.alamat_pesanan', '=', 'alamat_pesanan.alamat')
                 // ->join('users as telp_pesanan',
'pesanans.telp_pesanan', '=', 'telp_pesanan.no_telp')
                ->join('produks', 'pesanans.id produk', '=',
'produks.id')
                ->where('pesanans.created at', $created at)
                ->get(['pesanans.*', 'pelanggan.name as
nama_pelanggan', 'kurir.name as nama_kurir', 'produks.harga',
'produks.harga_sebelum', 'produks.nama_produk', 'produks.kategori']);
            if (count($pesanan) > 0) {
                for ($j=0; $j<count($pesanan); $j++) {</pre>
                     array_push($id_pesanan_arr, $pesanan[$j]-
>id pesanan);
                     array_push($produk_arr, $pesanan[$j]->nama_produk);
                     array push($total arr, $pesanan[$j]->total);
                     array_push($jumlah_arr, $pesanan[$j]->total /
$pesanan[$j]->harga);
```

```
if ($pesanan[$j]->kategori == "Promo") {
                    //
                           array_push($jumlah_arr, $pesanan[$j]->total
/ $pesanan[$j]->harga sebelum);
                    // } else {
                           array_push($jumlah_arr, $pesanan[$j]->total
                    11
/ $pesanan[$j]->harga);
                    // }
                $temp['kode pesanan'] = strtotime($pesanan[0]-
>created at);
                $temp['id user'] = $pesanan[0]->id user;
                $temp['nama_pelanggan'] = $pesanan[0]->nama_pelanggan;
                $temp['id kurir'] = $pesanan[0]->id_kurir;
                $temp['nama_kurir'] = $pesanan[0]->nama_kurir;
                $temp['telepon'] = $pesanan[0]->telp_pesanan;
                $temp['alamat'] = $pesanan[0]->alamat pesanan;
                $temp['id_pesanan'] = $id_pesanan_arr;
                $temp['nama produk'] = $produk arr;
                $temp['total'] = $total arr;
                $temp['jumlah'] = $jumlah arr;
                $temp['status'] = $pesanan[0]->status pesanan;
                $temp['created at'] = $pesanan[0]->created at;
                $temp['updated at'] = $pesanan[0]->updated at;
                array push($json, $temp); $id pesanan arr =
                array();
                $produk_arr = array();
                $total_arr = array();
                $jumlah arr = array();
        return response()->json(["pesanans"=>$json], 200);
    public function getPesanansByKurir($id)
         $pesanans = DB::table('pesanans')
            ->where('id_kurir', $id)
            ->whereIn('status pesanan', ['0','1','2'])
            ->orderBy('created_at', 'desc')
            ->distinct()
            ->get(['created at']);
        $json = array();
        $id pesanan arr = array();
        $produk arr = array();
        $total arr = array();
        $jumlah arr = array();
        for ($i=0; $i < count($pesanans); $i++) {</pre>
            $created at = $pesanans[$i]->created at;
            $pesanan = DB::table('pesanans')
                ->join('users as pelanggan', 'pesanans.id_user', '=',
'pelanggan.id')
                ->join('users as kurir', 'pesanans.id_kurir', '=',
'kurir.id')
                // ->join('users as alamat pesanan',
'pesanans.alamat_pesanan', '=', 'alamat_pesanan.alamat')
                // ->join('users as telp pesanan',
'pesanans.telp_pesanan', '=', 'telp_pesanan.no_telp')
                ->join('produks', 'pesanans.id_produk', '=',
'produks.id')
                ->where('pesanans.created at', $created at)
```

```
->get(['pesanans.*', 'pelanggan.name as nama_pelanggan', 'kurir.name as nama_kurir', 'produks.harga',
'produks.harga_sebelum', 'produks.nama_produk', 'produks.kategori']);
            if (count($pesanan) > 0) {
                for ($j=0; $j<count($pesanan); $j++) {</pre>
                     array push ($id pesanan arr, $pesanan[$j]-
>id pesanan);
                     array_push($produk_arr, $pesanan[$j]->nama_produk);
                     array push($total arr, $pesanan[$j]->total);
                     array_push($jumlah_arr, $pesanan[$j]->total /
$pesanan[$j]->harga);
                     // if ($pesanan[$j]->kategori == "Promo"){
                     //
                           array push($jumlah arr, $pesanan[$j]->total
/ $pesanan[$j]->harga sebelum);
                     // } else {
                     //
                            array push($jumlah arr, $pesanan[$j]->total
/ $pesanan[$j]->harga);
                    // }
                $temp['kode pesanan'] = strtotime($pesanan[0]-
>created at);
                $temp['id user'] = $pesanan[0]->id_user;
                $temp['nama_pelanggan'] = $pesanan[0]->nama_pelanggan;
                $temp['id_kurir'] = $pesanan[0]->id_kurir;
                $temp['nama_kurir'] = $pesanan[0]->nama_kurir;
                $temp['telepon'] = $pesanan[0]->telp pesanan;
                $temp['alamat'] = $pesanan[0]->alamat pesanan;
                $temp['id_pesanan'] = $id_pesanan_arr;
                $temp['nama produk'] = $produk arr;
                $temp['total'] = $total_arr;
                $temp['jumlah'] = $jumlah_arr;
                $temp['status'] = $pesanan[0]->status pesanan;
                $temp['created_at'] = $pesanan[0]->created_at;
                $temp['updated at'] = $pesanan[0]->updated at;
                array push($json, $temp); $id pesanan arr =
                array();
                $produk arr = array();
                $total_arr = array();
                $jumlah arr = array();
        }
        return response()->json(["pesanans"=>$json], 200);
    public function getHistoriPesanansByKurir($id)
         $pesanans = DB::table('pesanans')
            ->where('id kurir', $id)
            ->whereIn('status pesanan', ['3'])
            ->orderBy('created at', 'desc')
            ->distinct()
            ->get(['created_at']);
        $json = array();
        $id_pesanan_arr = array();
        produk_arr = array();
        $total arr = array();
        $jumlah_arr = array();
        for ($i=0; $i<count($pesanans); $i++) {</pre>
            $created_at = $pesanans[$i]->created at;
```

```
$pesanan = DB::table('pesanans')
                 ->join('users as pelanggan', 'pesanans.id_user', '=',
'pelanggan.id')
                 ->join('users as kurir', 'pesanans.id kurir', '=',
'kurir.id')
// ->join('users as alamat_pesanan',
'pesanans.alamat_pesanan', '=', 'alamat_pesanan.alamat')
                 // ->join('users as telp_pesanan',
'pesanans.telp pesanan', '=', 'telp pesanan.no telp')
                 ->join('produks', 'pesanans.id produk', '=',
'produks.id')
                 ->where('pesanans.created at', $created at)
                 ->get(['pesanans.*', 'pelanggan.name as
nama_pelanggan', 'kurir.name as nama_kurir',
'produks.harga', 'produks.nama_produk']);
             if (count($pesanan) > 0) {
                 for ($j=0; $j<count($pesanan); $j++) {</pre>
                      array_push($id_pesanan_arr, $pesanan[$j]-
>id pesanan);
                      array_push($produk_arr, $pesanan[$j]->nama_produk);
                     array_push($total_arr, $pesanan[$j]->total);
array_push($jumlah_arr, $pesanan[$j]->total /
$pesanan[$j]->harga);
                 $temp['kode pesanan'] = strtotime($pesanan[0]-
>created at);
                 $temp['id user'] = $pesanan[0]->id user;
                 $temp['nama_pelanggan'] = $pesanan[0]->nama_pelanggan;
                 $temp['id_kurir'] = $pesanan[0]->id_kurir;
                 $temp['nama_kurir'] = $pesanan[0]->nama_kurir;
                 $temp['telepon'] = $pesanan[0]->telp_pesanan;
                 $temp['alamat'] = $pesanan[0]->alamat pesanan;
                 $temp['id_pesanan'] = $id_pesanan_arr;
                 $temp['nama produk'] = $produk arr;
                 $temp['total'] = $total arr;
                 $temp['jumlah'] = $jumlah arr;
                 $temp['status'] = $pesanan[0]->status pesanan;
                 $temp['created at'] = $pesanan[0]->created at;
                 $temp['updated_at'] = $pesanan[0]->updated_at;
                 array push($json, $temp); $id pesanan arr =
                 array();
                 $produk_arr = array();
                 $total_arr = array();
                 $jumlah_arr = array();
             }
        return response()->json(["pesanans"=>$json], 200);
    public function updateStatusPesanan($kode, Request $request){
         $datetime = date('Y-m-d H:i:s', $kode);
         if (Pesanan::where('created at', $datetime)-
>update(['status_pesanan' => $request->status])) {
             return 'success';
         } else {
             return 'failed';
         }
    }
```

D. KONFIGURASI

1. App.php

```
'providers' => [
        * Laravel Framework Service Providers...
       Illuminate\Auth\AuthServiceProvider::class,
       Illuminate\Broadcasting\BroadcastServiceProvider::class,
       Illuminate\Bus\BusServiceProvider::class,
       Illuminate\Cache\CacheServiceProvider::class,
Illuminate\Foundation\Providers\ConsoleSupportServiceProvider::class,
       Illuminate\Cookie\CookieServiceProvider::class,
       Illuminate\Database\DatabaseServiceProvider::class,
       Illuminate\Encryption\EncryptionServiceProvider::class,
       Illuminate\Filesystem\FilesystemServiceProvider::class,
Illuminate\Foundation\Providers\FoundationServiceProvider::class,
       Illuminate\Hashing\HashServiceProvider::class,
       Illuminate\Mail\MailServiceProvider::class,
       Illuminate\Notifications\NotificationServiceProvider::class,
       Illuminate\Pagination\PaginationServiceProvider::class,
       Illuminate\Pipeline\PipelineServiceProvider::class,
       Illuminate\Queue\QueueServiceProvider::class,
       Illuminate\Redis\RedisServiceProvider::class,
       Illuminate\Auth\Passwords\PasswordResetServiceProvider::class,
       Illuminate\Session\SessionServiceProvider::class,
       Illuminate\Translation\TranslationServiceProvider::class,
       Illuminate\Validation\ValidationServiceProvider::class,
       Illuminate\View\ViewServiceProvider::class,
       JeroenNoten\LaravelAdminLte\ServiceProvider::class,
       Barryvdh\DomPDF\ServiceProvider::class,
       Collective\Html\HtmlServiceProvider::class,
        * Package Service Providers...
       Laravel\Tinker\TinkerServiceProvider::class,
        * Application Service Providers...
       // App\Providers\BroadcastServiceProvider::class,
       App\Providers\EventServiceProvider::class,
       App\Providers\RouteServiceProvider::class,
       Spatie\Fractal\FractalServiceProvider::class,
   ],
             _____
   | Class Aliases
   |-----
   | This array of class aliases will be registered when this
application
   \mid is started. However, feel free to register as many as you wish
   as | the aliases are "lazy" loaded so they don't hinder
   performance. |
```

```
'aliases' => [
        'App' => Illuminate\Support\Facades\App::class, 'Artisan'
        => Illuminate\Support\Facades\Artisan::class, 'Auth' =>
        Illuminate\Support\Facades\Auth::class, 'Blade' =>
        Illuminate\Support\Facades\Blade::class, 'Broadcast' =>
        Illuminate\Support\Facades\Broadcast::class, 'Bus' =>
        Illuminate\Support\Facades\Bus::class,
        'Cache' => Illuminate\Support\Facades\Cache::class,
        'Config' => Illuminate\Support\Facades\Config::class,
        'Cookie' => Illuminate\Support\Facades\Cookie::class,
        'Crypt' => Illuminate\Support\Facades\Crypt::class, 'DB'
        => Illuminate\Support\Facades\DB::class, 'Eloquent' =>
        Illuminate\Database\Eloquent\Model::class, 'Event' =>
       Illuminate\Support\Facades\Event::class, 'File' =>
Illuminate\Support\Facades\File::class, 'Gate' =>
        Illuminate\Support\Facades\Gate::class, 'Hash' =>
        Illuminate\Support\Facades\Hash::class, 'Lang' =>
        Illuminate\Support\Facades\Lang::class, 'Log' =>
        Illuminate\Support\Facades\Log::class,
        'Mail' => Illuminate\Support\Facades\Mail::class,
        'Notification' =>
Illuminate\Support\Facades\Notification::class,
        'Password' => Illuminate\Support\Facades\Password::class,
        'Queue' => Illuminate\Support\Facades\Queue::class,
        'Redirect' => Illuminate\Support\Facades\Redirect::class,
        'Redis' => Illuminate\Support\Facades\Redis::class,
        'Request' => Illuminate\Support\Facades\Request::class,
        'Response' => Illuminate\Support\Facades\Response::class,
        'Route' => Illuminate\Support\Facades\Route::class,
        'Schema' => Illuminate\Support\Facades\Schema::class,
        'Session' => Illuminate\Support\Facades\Session::class,
        'Storage' => Illuminate\Support\Facades\Storage::class,
        'URL' => Illuminate\Support\Facades\URL::class, 'Validator'
        => Illuminate\Support\Facades\Validator::class, 'View' =>
        Illuminate\Support\Facades\View::class, 'Fractal' =>
        Spatie\Fractal\FractalFacade::class,
        'PDF' => Barryvdh\DomPDF\Facade::class,
        'Form' => Collective\Html\FormFacade::class,
        'Html' => Collective\Html\HtmlFacade::class,
```

Adminite.php

```
'menu' =>
        [
            'text'
                           => 'HOME',
            'url'
                           => 'admintoko',
                           => 'home',
             'icon'
        'MASTER DATA',
        Γ
            'text' => 'Master User',
             'url' => 'masteruser',
             'icon' => 'user',
        ],
        Γ
            'text' => 'Master Produk',
             'url' => 'masterproduk',
             'icon' => 'list-alt',
        ],
        Γ
            'text' => 'Stock Warning',
```

```
'url' => 'stockwarning',
    'icon' => 'warning',
],
'TRANSAKSI',
[
    'text' => 'Data Transaksi',
    'url' => 'transaksi',
    'icon' => 'dollar',
],
],
```

3. .env

```
APP_NAME=Laravel
APP_ENV=local
APP_KEY=base64:ZgAleYLNu8SFDIXrabYSGoDlVP1i6N7qkI6AM00cdkg=
APP_DEBUG=true
APP_LOG_LEVEL=debug
APP_URL=http://antargen.informatikapolines.com

DB_CONNECTION=mysql
DB_HOST=127.0.0.1
DB_PORT=3306
DB_DATABASE=u5293999_delivery
DB_USERNAME=u5293999_delivery
DB_PASSWORD=delivery123
```

4. web.php

```
<?php
Route::get('/', function () {
    return view('welcome');
Route::get('/home', 'HomeController@index');
Auth::routes();
Route::auth();
Route::get('logout',
'\App\Http\Controllers\Auth\LoginController@logout');
Route::get('admintoko', ['middleware' => ['web', 'auth',
'admintoko'], function() {
      return view('/admintoko', 'AdminTokoController');
}]);
//ADMINTOKO
Route::resource('/admintoko', 'AdminTokoController');
Route::get('/admintoko', function () {
    return view('admintoko.index');
});
Route::get('/transaksi', 'AdminTokoController@pesanan');
Route::delete('/transaksi/{id pesanan}',
'AdminTokoController@destroy');
Route::resource('/masteruser', 'MasterUserController');
Route::resource('/masterproduk', 'MasterProdukController');
Route::post('/masterproduk/{id}', 'MasterProdukController@update');
Route::get('/stockwarning', 'MasterProdukController@stockwarning');
Route::get('/cariuser', 'MasterUserController@search');
Route::get('/cariproduk', 'MasterProdukController@search');
Route::get('/caristock', 'MasterProdukController@searchstock');
Route::get('/caritransaksi', 'AdminTokoController@search');
//MANAJER
Route::get('/manajer', function () {
    return view('manajer.index');
```

```
Route::get('manajer/pesanan', 'ManajerController@pesanan');
Route::get('manajer/terlaris', 'ManajerController@terlaris');
Route::get('manajer/laporan-harian', 'ManajerController@harian')-
>name('laporan.harian');
Route::get('manajer/laporan-harian-pdf/{id_pesanan}',
'ManajerController@harianpdf')->name('laporan.harianpdf');
Route::get('manajer/laporan-bulanan', 'ManajerController@bulanan')-
>name('laporan.bulanan');
Route::get('manajer/laporan-bulanan-pdf/{id_pesanan}',
'ManajerController@bulananpdf')->name('laporan.bulananpdf');
Route::get('logout', '\App\Http\Controllers\ManajerController@logout');
//KURIR
Route::resource('/kurir', 'KurirController');
Route::get('/kurir', function () {
    return view('kurir.index');
});
```

Lampiran 2 Kode Program Android

A. KODE PROGRAM PELANGGAN

AndroidManifest.Java

```
<?xml version="1.0" encoding="utf-8"?>
<manifest xmlns:android="http://schemas.android.com/apk/res/android"</pre>
    package="com.example.asus.tugasakhir2">
    <uses-permission android:name="android.permission.INTERNET" />
    <uses-permission android:name="android.permission.GET ACCOUNTS" />
    <uses-permission android:name="android.permission.READ_PROFILE" />
    <uses-permission android:name="android.permission.READ_CONTACTS" />
    <uses-permission</pre>
android:name="android.permission.ACCESS_NETWORK_STATE"
    /> <uses-permission
android:name="android.permission.WRITE EXTERNAL STORAGE"
    /> <uses-permission
android:name="android.permission.ACCESS_FINE_LOCATION"
    /> <uses-permission
android:name="android.permission.ACCESS COARSE LOCATION" />
    <uses-feature
        android:name="android.hardware.location.network"
        android:required="false" />
    <uses-feature
        android:name="android.hardware.location.gps"
        android:required="false" />
    <application</pre>
        android:name="com.orm.SugarApp"
        android:allowBackup="true"
        android:icon="@mipmap/ic_launcher"
        android:label="@string/app name"
        android:roundIcon="@mipmap/ic_launcher_round"
        android:supportsRtl="true"
        android: theme="@style/AppTheme">
        <meta-data
            android:name="DATABASE"
            android:value="antargen.db" />
        <meta-data
            android:name="VERSION"
            android:value="1" />
        <meta-data
            android:name="QUERY LOG"
            android:value="true" />
        <activity
            android:name=".activities.splashscreen"
            android: theme="@style/AppTheme.NoActionBar.Fullscreen">
            <intent-filter>
                <action android:name="android.intent.action.MAIN" />
                <category
android:name="android.intent.category.LAUNCHER" />
            </intent-filter>
        </activity>
        <activity
            android:name=".activities.MainActivity"
            android:label="@string/title activity main"
            android:theme="@style/AppTheme.NoActionBar" />
        <activity
```

```
android:name=".activities.loginutama"
            android: theme="@style/AppTheme.NoActionBar.Fullscreen"/>
        <activity
            android:name=".activities.ProdukDetailActivity"
            android: theme="@style/AppTheme.NoActionBar" />
        <activity
            android:name=".activities.TroliActivity"
            android:theme="@style/AppTheme.NoActionBar" />
        <activity
            android:name=".activities.PesananSayaActivity"
            android:theme="@style/AppTheme.NoActionBar" />
        <meta-data
            android:name="com.google.android.geo.API KEY"
            android:value="@string/google_maps_key" />
        <activity
            android:name=".activities.PosisiKurirActivity"
            android:label="@string/title_activity_posisi_kurir" />
</manifest>
```

PesananAdapter.Java

```
package com.example.asus.tugasakhir2.adapter;
import android.app.Dialog;
import android.content.Context;
import android.content.Intent;
import android.content.SharedPreferences; import
android.support.v7.widget.RecyclerView; import
android.util.Log;
import android.view.LayoutInflater;
import android.view.View;
import android.view.ViewGroup;
import android.view.Window;
import android.widget.Button;
import android.widget.RatingBar;
import android.widget.TextView;
import com.example.asus.tugasakhir2.R;
import com.example.asus.tuqasakhir2.activities.PesananSayaActivity;
import com.example.asus.tugasakhir2.activities.PosisiKurirActivity;
import com.example.asus.tugasakhir2.models.Pesanan;
import com.example.asus.tugasakhir2.models.PesananResponse;
import com.example.asus.tugasakhir2.models.RatingTable;
import java.util.List;
import retrofit2.Callback;
import static android.content.Context.MODE PRIVATE;
public class PesananAdapter extends
RecyclerView.Adapter<PesananAdapter.MyViewHolder>
    { private List<Pesanan> pesananList;
    Context mcontext;
    public class MyViewHolder extends RecyclerView.ViewHolder
        { TextView mTvKode;
        TextView mTvKurir;
        TextView mTvTanggal;
```

```
TextView mTvProduk;
        TextView mTvJumlah;
        TextView mTvStatus;
        TextView mLacakKurir;
        public MyViewHolder(View view) {
            super(view);
            mTvKode = (TextView) itemView.findViewById(R.id.tv kode);
            mTvKurir = (TextView) itemView.findViewById(R.id.tv kurir);
            mTvTanggal = (TextView)
itemView.findViewById(R.id.tv tanggal);
            mTvProduk = (TextView)
itemView.findViewById(R.id.tv_produk);
            mTvJumlah = (TextView)
itemView.findViewById(R.id.tv_jumlah);
            mTvStatus = (TextView)
itemView.findViewById(R.id.tv status);
            mLacakKurir = (TextView)
itemView.findViewById(R.id.lacak_kurir);
        }
   public PesananAdapter(List<Pesanan> pesananList, Context context)
        { this.pesananList = pesananList;
        this.mcontext = context;
    @Override
   public MyViewHolder onCreateViewHolder(ViewGroup parent, int
viewType) {
        View itemView = LayoutInflater.from(parent.getContext())
                .inflate(R.layout.pesanan_item, parent, false);
        return new MyViewHolder(itemView);
    }
    @Override
    public void onBindViewHolder(final MyViewHolder holder, final int
position) {
        final Pesanan pesanan = pesananList.get(position);
        final Context mContext = holder.mTvKode.getContext();
        final float[] rating = {0};
        holder.mTvKode.setText("Pesanan #" + pesanan.getKodePesanan());
        holder.mTvKurir.setText(pesanan.getNamaKurir());
        holder.mTvTanggal.setText(pesanan.getUpdatedAt().substring(0,
10));
        holder.mTvProduk.setText("");
        String valueProduk = "";
        for (int i = 0; i < pesanan.getNamaProduk().size(); i++) {</pre>
            if (i == pesanan.getNamaProduk().size() - 1){
                valueProduk = valueProduk + "•" +
pesanan.getNamaProduk().get(i);
            } else {
                valueProduk = valueProduk + "•" +
pesanan.getNamaProduk().get(i) + "\n";
            }
        holder.mTvProduk.setText(valueProduk);
        String valueTotal = "";
        for (int i = 0; i < pesanan.getTotal().size(); i++)</pre>
            { if (i == pesanan.getTotal().size() - 1) {
                valueTotal = valueTotal + "Rp. "
+ pesanan.getTotal().get(i);
```

```
else {
                valueTotal = valueTotal + "Rp. "
+ pesanan.getTotal().get(i) + "\n";
            }
        }
        holder.mTvJumlah.setText(valueTotal);
        holder.mLacakKurir.setVisibility(View.GONE);
        //set padding
        int paddingDp = 8;
        float density =
mContext.getResources().getDisplayMetrics().density; int
        paddingPixel = (int) (paddingDp * density);
        holder.mLacakKurir.setPadding(0,0,paddingPixel,0)
        if (pesanan.getStatus().toString().equals("0")) {
            holder.mTvStatus.setText("Sedang Diproses");
        } else if (pesanan.getStatus().toString().equals("1")) {
            holder.mTvStatus.setText("Pickup Barang");
        } else if (pesanan.getStatus().toString().equals("2")){
            holder.mTvStatus.setText("Dalam Pengiriman");
            holder.mLacakKurir.setText("Lacak Posisi Kurir >>");
            holder.mLacakKurir.setVisibility(View.VISIBLE);
            holder.mLacakKurir.setCompoundDrawablesWithIntrinsicBounds(
R.drawable.ic_lokasi, 0, 0, 0);
            holder.mLacakKurir.setOnClickListener(new
View.OnClickListener() {
                @Override
                public void onClick(View view) {
                    Intent intent = new Intent(mContext,
PosisiKurirActivity.class);
                     intent.putExtra("ID KURIR", pesanan.getIdKurir());
                    mContext.startActivity(intent);
            });
        else if (pesanan.getStatus().toString().equals("3")) {
            holder.mTvStatus.setText("Telah Diterima");
            holder.mLacakKurir.setVisibility(View.VISIBLE);
            \verb|holder.mLacakKurir.setCompoundDrawablesWithIntrinsicBounds|| (
R.drawable.ic_star, 0, 0, 0);
            holder.mLacakKurir.setText("Berikan Rating");
            holder.mLacakKurir.setOnClickListener(new
View.OnClickListener() {
                @Override
                public void onClick(View view) {
                    final Dialog rankDialog = new Dialog(mContext);
rankDialog.requestWindowFeature(Window.FEATURE NO TITLE);
                    rankDialog.setContentView(R.layout.dialog rating)
                     ; rankDialog.setCancelable(true);
                    RatingBar ratingBar =
(RatingBar) rankDialog.findViewById(R.id.dialog_ratingbar);
                     //Cek Nilai SharedPreferences
                    String skor =
String.valueOf(pesanan.getKodePesanan());
                    SharedPreferences preferences =
mContext.getSharedPreferences("RATING", MODE PRIVATE);
                    if (!preferences.getString(skor, "").equals("")){
                         rating[0] =
Float.parseFloat(preferences.getString(skor, ""));
```

```
ratingBar.setRating(rating[0]);
                    //Listener RatingBarChange
                    ratingBar.setOnRatingBarChangeListener(new
RatingBar.OnRatingBarChangeListener() {
                        @Override
                        public void onRatingChanged(RatingBar
ratingBar, float v, boolean b) {
                            rating[0] = v;
                    });
                    TextView text = (TextView)
rankDialog.findViewById(R.id.rank_dialog_text1);
                    text.setText(pesanan.getNamaKurir());
                    Button updateButton = (Button)
rankDialog.findViewById(R.id.rank dialog button);
                    updateButton.setOnClickListener(new
View.OnClickListener() {
                         @Override
                        public void onClick(View v) {
                            String skor =
String.valueOf(pesanan.getKodePesanan());
                            SharedPreferences.Editor editorRating =
mContext.getSharedPreferences("RATING", MODE_PRIVATE).edit();
                            editorRating.putString(skor,
String.valueOf(rating[0]));
                            editorRating.apply();
                            rankDialog.dismiss();
                    });
                    rankDialog.show();
            });
        }
    @Override
    public int getItemCount() {
        return pesananList.size();
```

ProdukAdapter.Java

```
package com.example.asus.tugasakhir2.adapter;
import android.content.Context;
import android.content.Intent;
import android.graphics.Paint;
import android.support.v7.widget.CardView;
import android.support.v7.widget.RecyclerView;
import android.view.LayoutInflater; import
android.view.View;
import android.view.ViewGroup;
```

```
import android.widget.ImageView;
import android.widget.TextView;
import com.example.asus.tugasakhir2.R;
import com.example.asus.tugasakhir2.activities.ProdukDetailActivity;
import com.example.asus.tugasakhir2.config.URLConfig;
import com.example.asus.tugasakhir2.models.Produk;
import com.squareup.picasso.Picasso;
import java.util.List;
public class ProdukAdapter extends
RecyclerView.Adapter<ProdukAdapter.MyViewHolder>{
   private List<Produk> produkList;
   public class MyViewHolder extends RecyclerView.ViewHolder
        { TextView tnama, tharga, tstok;
        ImageView tfoto;
        CardView tcontainer;
        public MyViewHolder(View view) {
            super(view);
            tnama = (TextView) itemView.findViewById(R.id.Nama Produk);
            tharga = (TextView) itemView.findViewById(R.id.Harga);
            tstok = (TextView) itemView.findViewById(R.id.Stok); tfoto
            = (ImageView) itemView.findViewById(R.id.gambar);
            tcontainer = (CardView)
itemView.findViewById(R.id.container);
       }
    }
    public ProdukAdapter(List<Produk> produkList)
        { this.produkList = produkList;
    @Override
   public ProdukAdapter.MyViewHolder onCreateViewHolder(ViewGroup
parent, int viewType) {
        View itemView = LayoutInflater.from(parent.getContext())
                .inflate(R.layout.produk_item, parent, false);
        return new ProdukAdapter.MyViewHolder(itemView);
    }
    @Override
   public void onBindViewHolder(final
ProdukAdapter.MyViewHolder holder, final int position) {
        final Context Context = holder.tcontainer.getContext();
        final Produk produk = produkList.get(position);
        holder.tnama.setText(produk.getNamaProduk());
        holder.tharga.setText("Rp. " + produk.getHarga());
        holder.tstok.setText("Stok: " + produk.getStok());
        holder.tcontainer.setOnClickListener(new View.OnClickListener()
            @Override
            public void onClick(View view) {
                Intent intent = new Intent(Context,
ProdukDetailActivity.class);
                intent.putExtra("NAMA", produk.getNamaProduk());
                intent.putExtra("HARGA", produk.getHarga());
                intent.putExtra("STOK", produk.getStok());
                intent.putExtra("KATEGORI", produk.getKategori());
                intent.putExtra("FOTO", produk.getFoto());
                intent.putExtra("ID", produk.getId_produk());
```

```
Context.startActivity(intent);
}
});

if (!produk.getFoto().equals("")){
    String url = URLConfig.FOTO_URL +
produk.getFoto().replaceAll(" ", "%20");
    Picasso.get().load(url).into(holder.tfoto);
}

@Override
public int getItemCount() {
    return produkList.size();
}
```

PromoAdapter.Java

```
package com.example.asus.tugasakhir2.adapter;
import android.content.Context;
import android.content.Intent;
import android.graphics.Paint;
import android.support.v7.widget.CardView;
import android.support.v7.widget.RecyclerView;
import android.view.LayoutInflater; import
android.view.View;
import android.view.ViewGroup;
import android.widget.ImageView;
import android.widget.TextView;
import com.example.asus.tugasakhir2.R;
import com.example.asus.tugasakhir2.activities.ProdukDetailActivity;
import com.example.asus.tugasakhir2.config.URLConfig;
import com.example.asus.tugasakhir2.models.ProduksItem;
import com.squareup.picasso.Picasso;
import java.util.List;
public class PromoAdapter extends
RecyclerView.Adapter<PromoAdapter.MyViewHolder> {
    private List<ProduksItem> promoList;
    public class MyViewHolder extends RecyclerView.ViewHolder {
        TextView tnama, tharga, tstok ,thargaturun,tdiskon;
        ImageView tfoto;
        CardView tcontainer;
        public MyViewHolder(View itemView) {
            super(itemView);
            tdiskon = (TextView)
            itemView.findViewById(R.id.tvdiskon); tnama = (TextView)
itemView.findViewById(R.id.pNama_Produk);
            tharga = (TextView) itemView.findViewById(R.id.pHarga);
            thargaturun = (TextView)
itemView.findViewById(R.id.pHargaTurun);
            tstok = (TextView) itemView.findViewById(R.id.pStok); tfoto
            = (ImageView) itemView.findViewById(R.id.pgambar);
            tcontainer = (CardView)
```

```
itemView.findViewById(R.id.pcontainer);
       }
   public PromoAdapter(List<ProduksItem> produksPromo) {
        this.promoList = produksPromo;
    @Override
    public MyViewHolder onCreateViewHolder(ViewGroup parent, int
viewType) {
        View itemView = LayoutInflater.from(parent.getContext())
                .inflate(R.layout.promo_item, parent, false);
        return new PromoAdapter.MyViewHolder(itemView);
    @Override
   public void onBindViewHolder(MyViewHolder holder, int position) {
        final Context mContext = holder.tcontainer.getContext();
        final ProduksItem produk = promoList.get(position);
        //set flag untuk strike
        holder.tharga.setPaintFlags(holder.tharga.getPaintFlags() |
Paint.STRIKE_THRU_TEXT_FLAG);
        //end
        //mengubah stok ke string
        final int stok = produk.getStok();
        final String strstok = String.valueOf(stok).toString().trim();
        //hitung diskon
        int hargaAsli = Integer.parseInt(produk.getHargaSebelum());
        int hargaTurun = Integer.parseInt(produk.getHarga()); int
        diskon = ((hargaAsli - hargaTurun) * 100)/hargaAsli;
        final String harga = String.valueOf(hargaTurun).trim();
        final String id = String.valueOf(produk.getId()).trim();
        holder.tnama.setText(produk.getNamaProduk());
        holder.tharga.setText("Rp. " + produk.getHargaSebelum());
        holder.thargaturun.setText("Rp. " + harga);
        holder.tdiskon.setText(String.valueOf(diskon) + "%");
        holder.tstok.setText("Stok: " + stok);
        holder.tcontainer.setOnClickListener(new View.OnClickListener()
{
            @Override
            public void onClick(View view) {
                Intent promointent = new Intent(mContext,
ProdukDetailActivity.class);
                promointent.putExtra("NAMA", produk.getNamaProduk());
                promointent.putExtra("HARGA", harga);
                promointent.putExtra("STOK", strstok);
                promointent.putExtra("KATEGORI", produk.getKategori());
                promointent.putExtra("FOTO", produk.getFoto());
                promointent.putExtra("ID", id);
                mContext.startActivity(promointent);
        });
        if (!produk.getFoto().equals("")){
            String url = URLConfig. FOTO URL +
produk.getFoto().replaceAll(" ", "%20");
            Picasso.get().load(url).into(holder.tfoto);
```

```
@Override
public int getItemCount() {
    return promoList.size();
}
```

TroliAdapter.Java

```
package com.example.asus.tugasakhir2.adapter;
import android.content.Context;
import android.support.v7.widget.RecyclerView;
import android.util.Log;
import android.view.LayoutInflater;
import android.view.View;
import android.view.ViewGroup;
import android.widget.ImageView;
import android.widget.RelativeLayout;
import android.widget.TextView;
import android.widget.Toast;
import com.example.asus.tugasakhir2.R;
import com.example.asus.tugasakhir2.activities.TroliActivity;
import com.example.asus.tugasakhir2.config.URLConfig; import
com.example.asus.tugasakhir2.models.Troli;
import com.example.asus.tuqasakhir2.network.APIService;
import com.example.asus.tugasakhir2.network.RetrofitClient;
import com.google.gson.JsonObject; import
com.squareup.picasso.Picasso;
import java.util.List;
import retrofit2.Call;
import retrofit2.Callback;
import retrofit2.Response;
public class TroliAdapter extends
RecyclerView.Adapter<TroliAdapter.MyViewHolder>{
    private List<Troli> troliList;
    Context mContext;
    public class MyViewHolder extends RecyclerView.ViewHolder
        { TextView tnama, tharga, tkategori, ttotal;
        RelativeLayout tcontainer;
        ImageView tfoto, tdelete;
        public MyViewHolder(View view) {
            super(view);
            tnama = (TextView) itemView.findViewById(R.id.nama);
            tkategori = (TextView)
itemView.findViewById(R.id.kategori);
            tharga = (TextView) itemView.findViewById(R.id.harga);
            ttotal = (TextView) itemView.findViewById(R.id.total);
            tcontainer = (RelativeLayout)
itemView.findViewById(R.id.container);
            tfoto = (ImageView) itemView.findViewById(R.id.foto);
            tdelete = (ImageView) itemView.findViewById(R.id.delete);
```

```
}
    public TroliAdapter(List<Troli> troliList, Context troliActivity)
        { this.troliList = troliList;
        this.mContext = troliActivity;
    @Override
    public TroliAdapter.MyViewHolder
onCreateViewHolder(ViewGroup parent, int viewType) {
        View itemView = LayoutInflater.from(parent.getContext())
                .inflate(R.layout.troli_item, parent, false);
        return new TroliAdapter.MyViewHolder(itemView);
    }
    @Override
    public void onBindViewHolder(final TroliAdapter.MyViewHolder
holder, final int position) {
        final Context mContext = holder.tcontainer.getContext();
        final Troli item = troliList.get(position);
        holder.tnama.setText(item.getNama());
        holder.tkategori.setText(item.getKategori());
        holder.tharga.setText(item.getJumlah() + " x Rp. " +
item.getHarga());
        holder.ttotal.setText("Rp. " +
(Integer.parseInt(item.getJumlah()) *
Integer.parseInt(item.getHarga())));
        if (!item.getFoto().equals("")){
            String url = URLConfig.FOTO_URL +
item.getFoto().replaceAll(" ", "%20");
            Picasso.get().load(url).into(holder.tfoto);
        holder.tdelete.setOnClickListener(new View.OnClickListener()
            { @Override
            public void onClick(View view) {
                troliList.remove(position);
                notifyItemRemoved(position);
removeTroli(String.valueOf(item.getId produk()),item.getJumlah());
                \label{log.d} \mbox{Log.d("kurang", "onClick: "+item.getId()+" jumlhah}
:"+item.getJumlah());
                notifyItemRangeChanged(position, troliList.size());
                Troli troli = Troli.findById(Troli.class,
item.getId());
                troli.delete():
                if (mContext instanceof TroliActivity) {
                     ((TroliActivity) mContext).updateTotal(troliList);
        });
    private void removeTroli(String id, String jumlah) {
        APIService service =
RetrofitClient.getClient().create(APIService.class);
        Call<JsonObject> troliRmvCall = service.removeTroli(id,jumlah);
        troliRmvCall.enqueue(new Callback<JsonObject>() {
            @Override
            public void onResponse(Call<JsonObject> call,
Response<JsonObject> response) {
```

```
Toast.makeText(mContext, "Stok dikembalikan",
Toast.LENGTH_SHORT).show();

@Override
    public void onFailure(Call<JsonObject> call, Throwable t) {

    }
});
}

@Override
public int getItemCount() {
    return troliList.size();
}
```

Loginutama.java

```
package com.example.asus.tugasakhir2.activities;
import android.content.Intent;
import android.content.SharedPreferences;
import android.os.Bundle;
import android.support.v7.app.AppCompatActivity;
import android.view.View; import
android.widget.TextView;
import android.widget.Toast;
import com.example.asus.tugasakhir2.R;
import com.example.asus.tugasakhir2.models.Login;
import com.example.asus.tugasakhir2.models.LoginResponse;
import com.example.asus.tugasakhir2.network.APIService;
import com.example.asus.tugasakhir2.network.RetrofitClient;
import retrofit2.Call;
import retrofit2.Callback;
import retrofit2.Response;
public class loginutama extends AppCompatActivity
    { TextView email, password;
    @Override
    protected void onCreate(Bundle savedInstanceState)
        { super.onCreate(savedInstanceState);
        setContentView(R.layout.activity_loginutama);
        if (isLogin()){
            Intent intent = new
Intent(loginutama.this, MainActivity.class);
            startActivity(intent); finish();
        }
        email=(TextView) findViewById(R.id.txt_email);
        password=(TextView) findViewById(R.id.txt_password);
    public void MainActivity (View view) {
login(email.getText().toString(),password.getText().toString());
    }
        public void login(String email, String password) {
```

```
//Retrofit call
        APIService service =
RetrofitClient.getClient().create(APIService.class);
        Call<LoginResponse> userCall = service.userLogin(email,
password);
        userCall.enqueue(new Callback<LoginResponse>() {
            //Fungsi ini akan dieksekusi ketika ada respon dari
            server @Override
            public void onResponse(Call<LoginResponse>
call, Response<LoginResponse> response) {
                //Cek apakah kode respon server 401
                if (response.code() == 401) {
                    Toast.makeText(loginutama.this, "Login Gagal",
Toast. LENGTH SHORT) . show();
                } else {
                     //Ambil data dari respon server
                    Login login = response.body().getData();
                    //Memanggil fungsi simpan session login
dengan parameter nama dan email
                    saveSession(login.getId(),
login.getName(), login.getEmail(),login.getId());
                     //Intent untuk pindah activity ke
                    MainActivity Intent intent = new
Intent(loginutama.this, MainActivity.class);
                    startActivity(intent);
                    finish();
            @Override
            public void onFailure(Call<LoginResponse> call, Throwable
t) {
                Toast.makeText(loginutama.this, "Gagal Terhubung ke
Server", Toast. LENGTH_SHORT) . show();
        });
    }
    public void saveSession(String id, String name, String
email, String loginId){
        SharedPreferences.Editor editor =
getSharedPreferences("SESSION", MODE_PRIVATE).edit();
        editor.putString("ID", id );
        editor.putString("NAME", name );
        editor.putString("EMAIL", email );
        editor.putString("ID", id );
        editor.apply();
    public boolean isLogin () {
        SharedPreferences preferences =
getSharedPreferences("SESSION", MODE_PRIVATE);
        return !preferences.getString("ID", "").equals("");
```

MainActivity.Java

```
package com.example.asus.tugasakhir2.activities;
import android.content.Intent;
```

```
import android.content.SharedPreferences;
import android.os.Bundle;
import android.support.annotation.NonNull;
import android.support.design.widget.NavigationView;
import android.support.v4.app.Fragment;
import android.support.v4.app.FragmentTransaction;
import android.support.v4.view.GravityCompat; import
android.support.v4.widget.DrawerLayout; import
android.support.v7.app.ActionBarDrawerToggle; import
android.support.v7.app.AppCompatActivity; import
android.support.v7.widget.Toolbar; import
android.view.Menu;
import android.view.MenuItem;
import android.view.View;
import android.widget.TextView;
import android.widget.Toast;
import com.example.asus.tugasakhir2.R;
import com.example.asus.tugasakhir2.fragments.HalamanKRT; import
com.example.asus.tuqasakhir2.fragments.HalamanKesehatan; import
com.example.asus.tugasakhir2.fragments.HalamanMakanan; import
com.example.asus.tugasakhir2.fragments.HalamanMinuman; import
com.example.asus.tugasakhir2.fragments.HalamanUtama; import
com.example.asus.tugasakhir2.models.Troli;
public class MainActivity extends AppCompatActivity
        implements NavigationView.OnNavigationItemSelectedListener {
    @Override
    protected void onCreate(Bundle savedInstanceState)
        { super.onCreate(savedInstanceState);
        setContentView(R.layout.activity_main);
        Toolbar toolbar = (Toolbar) findViewById(R.id.toolbar);
        setSupportActionBar(toolbar);
        DrawerLayout drawer = (DrawerLayout)
findViewById(R.id.drawer layout);
        ActionBarDrawerToggle toggle = new ActionBarDrawerToggle(
                this, drawer, toolbar, R.string.navigation_drawer_open,
R.string.navigation drawer close);
        drawer.addDrawerListener(toggle);
        toggle.syncState();
        NavigationView navigationView = (NavigationView)
findViewById(R.id.nav_view);
        navigationView.setNavigationItemSelectedListener(this);
        displaySelectedScreen(R.id.nav_utama);
        //Ambil session login dari shared preference
        SharedPreferences preferences =
getSharedPreferences("SESSION", MODE PRIVATE);
        //Ubah TextView header sesuai nama dan email di session login
        View headerView = navigationView.getHeaderView(0); TextView
        headerNama = (TextView)
headerView.findViewById(R.id.header nama);
        TextView headerEmail = (TextView)
headerView.findViewById(R.id.header_email);
        headerNama.setText (preferences.getString ("NAME",
"").toString());
        headerEmail.setText(preferences.getString("EMAIL",
"").toString());
```

```
public void displaySelectedScreen(int itemId) {
        //creating fragment object
        Fragment fragment = null;
        Bundle bundle = new Bundle();
        switch (itemId) {
            case R.id.nav utama:
                fragment = new HalamanUtama();
                break;
            case R.id.nav makanan:
                fragment = new HalamanMakanan();
                break;
            case R.id.nav_minuman:
                fragment = new HalamanMinuman();
                break;
            case R.id.nav_obat:
                fragment = new HalamanKesehatan();
                break;
            case R.id.nav_pesanan:
                startActivity (new Intent (MainActivity.this,
PesananSayaActivity.class));
                break;
            case R.id.nav logout:
                //Hapus session login dari shared preference
                SharedPreferences.Editor editor =
getSharedPreferences("SESSION", MODE_PRIVATE).edit();
                editor.clear();
                editor.apply();
                //Hapus tabel troli dari sqlite
                Troli.deleteAll(Troli.class);
                Troli.executeQuery("DELETE FROM SQLITE SEQUENCE WHERE
NAME = 'TROLI'");
                //Intent untuk pindah activity ke LoginActivity
                Intent intent = new Intent(MainActivity.this,
loginutama.class);
                startActivity(intent);
                finish();
                break;
            case R.id.nav alat:
                fragment = new HalamanKRT();
                break:
        }
        if (fragment != null) {
            FragmentTransaction ft =
getSupportFragmentManager().beginTransaction();
            ft.replace (R.id.content frame,
            fragment); ft.commit();
        DrawerLayout drawer = (DrawerLayout)
findViewById(R.id.drawer_layout);
        drawer.closeDrawer(GravityCompat.START);
    @Override
    public void onBackPressed() {
        DrawerLayout drawer = (DrawerLayout)
findViewById(R.id.drawer_layout);
        if (drawer.isDrawerOpen(GravityCompat.START))
            { drawer.closeDrawer(GravityCompat.START);
        } else {
            Intent intent = new Intent(Intent.ACTION_MAIN);
```

```
intent.addCategory(Intent.CATEGORY HOME);
            intent.setFlags(Intent.FLAG_ACTIVITY_NEW_TASK);
            startActivity(intent);
            //super.onBackPressed();
        }
    }
    @Override
   public boolean onCreateOptionsMenu(Menu menu) {
        // Inflate the menu; this adds items to the action bar if it is
present.
       getMenuInflater().inflate(R.menu.main,
       menu); return true;
    @Override
   public boolean onOptionsItemSelected(MenuItem item) {
        int id = item.getItemId();
        //Ketika icon troli di klik
        if (id == R.id.action cart) {
            Intent intent = new Intent (MainActivity.this,
TroliActivity.class);
            startActivity(intent);
            return true;
       return super.onOptionsItemSelected(item);
    @Override
   public boolean onNavigationItemSelected(@NonNull MenuItem item) {
       displaySelectedScreen(item.getItemId());
       return false;
}
```

PesananSayaActivity.Java

```
package com.example.asus.tugasakhir2.activities;
import android.content.SharedPreferences;
import android.os.Bundle;
import android.support.v7.app.AppCompatActivity; import
android.support.v7.widget.DefaultItemAnimator; import
android.support.v7.widget.DividerItemDecoration; import
android.support.v7.widget.LinearLayoutManager; import
android.support.v7.widget.RecyclerView; import
android.support.v7.widget.Toolbar;
import android.widget.Toast;
import com.example.asus.tugasakhir2.R;
import com.example.asus.tuqasakhir2.adapter.PesananAdapter;
import com.example.asus.tugasakhir2.models.Pesanan;
import com.example.asus.tugasakhir2.models.PesananResponse;
import com.example.asus.tugasakhir2.network.APIService;
import com.example.asus.tugasakhir2.network.RetrofitClient;
import java.util.List;
import retrofit2.Call;
```

```
import retrofit2.Callback;
import retrofit2.Response;
public class PesananSayaActivity extends AppCompatActivity {
    RecyclerView recyclerView;
    PesananAdapter adapter;
    @Override
    protected void onCreate(Bundle savedInstanceState) {
        super.onCreate(savedInstanceState);
        setContentView(R.layout.activity pesanan saya);
        Toolbar toolbar = (Toolbar) findViewById(R.id.toolbar);
        setSupportActionBar(toolbar);
        getSupportActionBar().setDisplayHomeAsUpEnabled(true);
        getSupportActionBar().setDisplayShowHomeEnabled(true);
        getSupportActionBar().setTitle("Pesanan Saya");
        //Deklarasi dan konfigurasi recyclerview
        recyclerView = (RecyclerView)
findViewById(R.id.recycler pesanan);
        RecyclerView.LayoutManager mLayoutManager = new
LinearLayoutManager(this);
        recyclerView.setLayoutManager(mLayoutManager);
        recyclerView.setItemAnimator(new DefaultItemAnimator());
        recyclerView.addItemDecoration(new DividerItemDecoration(this,
LinearLayoutManager.VERTICAL));
        SharedPreferences preferences =
getSharedPreferences("SESSION", MODE_PRIVATE)
        loadPesanan(Integer.parseInt(preferences.getString("ID", "")));
    private void loadPesanan(int id) {
        APIService service =
RetrofitClient.getClient().create(APIService.class);
        Call<PesananResponse> userCall = service.getPesanan(id);
        userCall.enqueue(new Callback<PesananResponse>()
            { @Override
            public void onResponse(Call<PesananResponse> call,
Response<PesananResponse> response) {
                if (response.isSuccessful()){
                    List<Pesanan> pesanans =
response.body().getPesanans();
                    adapter = new
PesananAdapter (pesanans, PesananSayaActivity.this);
                    recyclerView.setAdapter(adapter);
            @Override
            public void onFailure(Call<PesananResponse> call, Throwable
t) {
                Toast.makeText(PesananSayaActivity.this,
"Gagal Terhubung Ke Server", Toast. LENGTH_SHORT).show();
        });
    }
    @Override
```

```
onBackPressed();
    return true;
}

@Override
public void onBackPressed() {
    super.onBackPressed();
}
```

PosisiKurirActivity.Java

```
package com.example.asus.tugasakhir2.activities;
import android.content.Context;
import android.graphics.Bitmap;
import android.graphics.Canvas;
import android.graphics.drawable.Drawable;
import android.os.Bundle;
import android.support.v4.app.FragmentActivity;
import android.support.v4.content.ContextCompat;
import android.util.Log; import
android.view.Gravity;
import android.widget.TextView;
import android.widget.Toast;
import com.example.asus.tugasakhir2.R;
import com.google.android.gms.maps.CameraUpdateFactory;
import com.google.android.gms.maps.GoogleMap;
import com.google.android.gms.maps.OnMapReadyCallback; import
com.google.android.gms.maps.SupportMapFragment; import
com.google.android.gms.maps.model.BitmapDescriptor; import
com.google.android.gms.maps.model.BitmapDescriptorFactory;
import com.google.android.gms.maps.model.LatLng;
import com.google.android.gms.maps.model.LatLngBounds;
import com.google.android.gms.maps.model.Marker; import
com.google.android.gms.maps.model.MarkerOptions; import
com.google.android.gms.tasks.OnCompleteListener; import
com.google.android.gms.tasks.Task;
import com.google.firebase.auth.AuthResult;
import com.google.firebase.auth.FirebaseAuth;
import com.google.firebase.database.ChildEventListener;
import com.google.firebase.database.DataSnapshot; import
com.google.firebase.database.DatabaseError; import
com.google.firebase.database.DatabaseReference; import
com.google.firebase.database.FirebaseDatabase;
import java.util.HashMap;
public class PosisiKurirActivity extends FragmentActivity
implements OnMapReadyCallback {
    private static final String TAG =
PosisiKurirActivity.class.getSimpleName();
    private HashMap<String, Marker> mMarkers = new HashMap<>();
    private GoogleMap mMap;
    private String idKurir;
    @Override
    protected void onCreate(Bundle savedInstanceState)
        { super.onCreate(savedInstanceState);
        setContentView(R.layout.activity_posisi_kurir);
                SupportMapFragment mapFragment = (SupportMapFragment)
```

```
getSupportFragmentManager()
                .findFragmentById(R.id.map);
        mapFragment.getMapAsync(this);
        idKurir = getIntent().getStringExtra("ID_KURIR");
    @Override
    public void onMapReady(GoogleMap googleMap) {
        // Authenticate with Firebase when the Google map is loaded
        mMap = googleMap;
        mMap.setMaxZoomPreference(16)
        ; loginToFirebase();
        Toast toast = Toast.makeText(this, "Mohon tunggu beberapa saat,
kami sedang melacak posisi kurir anda", Toast.LENGTH LONG);
        TextView v = (TextView)
toast.getView().findViewById(android.R.id.message);
        if( v != null) v.setGravity(Gravity.CENTER);
        toast.show();
    private void loginToFirebase() {
        String email = "admin@admin.com";
        String password = "admin123";
        // Authenticate with Firebase and subscribe to updates
        FirebaseAuth.getInstance().signInWithEmailAndPassword(
                email, password).addOnCompleteListener(new
OnCompleteListener<AuthResult>() {
            @Override
            public void onComplete(Task<AuthResult> task) {
                if (task.isSuccessful()) {
                    subscribeToUpdates();
                    Log.d(\mathit{TAG}, "firebase auth success");
                } else {
                    Log. d(TAG, "firebase auth failed");
            }
        });
    private void subscribeToUpdates() {
        DatabaseReference ref =
FirebaseDatabase.getInstance().getReference(idKurir);
        ref.addChildEventListener(new ChildEventListener() {
            @Override
            public void onChildAdded(DataSnapshot dataSnapshot,
String previousChildName) {
                setMarker(dataSnapshot);
            @Override
            public void onChildChanged(DataSnapshot
dataSnapshot, String previousChildName) {
                setMarker(dataSnapshot);
            }
            @Override
            public void onChildMoved(DataSnapshot dataSnapshot,
String previousChildName) {
            @Override
            public void onChildRemoved(DataSnapshot dataSnapshot) {
            @Override
```

```
public void onCancelled(DatabaseError error) {
                Log.d(TAG, "Failed to read value.",
error.toException());
        });
    private void setMarker(DataSnapshot dataSnapshot) {
        String key = dataSnapshot.getKey();
        HashMap<String, Object> value = (HashMap<String, Object>)
dataSnapshot.getValue();
        double lat
Double.parseDouble(value.get("latitude").toString());
        double lng =
Double.parseDouble(value.get("longitude").toString());
        LatLng location = new LatLng(lat, lng);
        if (!mMarkers.containsKey(key)) {
            Marker markerIcon = mMap.addMarker(new MarkerOptions()
                    .title("Kurir Anda")
                    .position(location)
                    .icon(bitmapDescriptorFromVector(this,
R.drawable.ic directions car)));
            markerIcon.showInfoWindow();
            mMarkers.put(key, markerIcon);
        } else {
            mMarkers.get(key).setPosition(location);
        LatLngBounds.Builder builder = new LatLngBounds.Builder();
        for (Marker marker : mMarkers.values()) {
            builder.include(marker.getPosition());
        }
mMap.animateCamera(CameraUpdateFactory.newLatLnqBounds(builder.build(),
300));
    private BitmapDescriptor
bitmapDescriptorFromVector(Context context, int vectorResId) {
        Drawable vectorDrawable =
ContextCompat.getDrawable(context, vectorResId);
        vectorDrawable.setBounds(0, 0,
vectorDrawable.getIntrinsicWidth(),
vectorDrawable.getIntrinsicHeight());
        Bitmap bitmap =
Bitmap.createBitmap(vectorDrawable.getIntrinsicWidth(),
vectorDrawable.getIntrinsicHeight(), Bitmap.Config.ARGB 8888);
        Canvas canvas = new Canvas(bitmap);
        vectorDrawable.draw(canvas);
        return BitmapDescriptorFactory.fromBitmap(bitmap);
```

ProdukDetailActivity.Java

```
package com.example.asus.tugasakhir2.activities;

import android.app.Dialog;
import android.content.Intent;
import android.os.Bundle;
import android.support.v7.app.AppCompatActivity;
import android.support.v7.widget.Toolbar;
import android.util.Log;
import android.view.Menu;
```

```
import android.view.MenuItem;
import android.view.View;
import android.view.Window;
import android.widget.Button;
import android.widget.EditText;
import android.widget.ImageView;
import android.widget.TextView;
import android.widget.Toast;
import com.example.asus.tugasakhir2.R;
import com.example.asus.tugasakhir2.config.URLConfig;
import com.example.asus.tugasakhir2.models.Troli; import
com.example.asus.tugasakhir2.network.APIService; import
com.example.asus.tugasakhir2.network.RetrofitClient; import
com.google.gson.JsonObject; import
com.squareup.picasso.Picasso;
import retrofit2.Call;
import retrofit2.Callback;
import retrofit2.Response;
public class ProdukDetailActivity extends AppCompatActivity {
    private Toolbar mActionBarToolbar;
   private String nama, harga, stok, kategori, foto,
    id; private ImageView mFotoProduk; private TextView
    mNamaProduk;
    private TextView mHargaProduk;
    private TextView mStokProduk;
   private Button mButtonTroli;
    @Override
   protected void onCreate(Bundle savedInstanceState) {
        super.onCreate(savedInstanceState);
        setContentView(R.layout.activity_produk_detail); Toolbar
        toolbar = (Toolbar) findViewById(R.id.toolbar);
        setSupportActionBar(toolbar);
        getSupportActionBar().setDisplayHomeAsUpEnabled(true);
        getSupportActionBar().setDisplayShowHomeEnabled(true);
        getSupportActionBar().setTitle("Detail Produk");
        initView();
        if (getIntent().getExtras() != null) {
            //Ambil intent extra dan simpan pada variabel
            nama = getIntent().getStringExtra("NAMA"); harga
            = getIntent().getStringExtra("HARGA"); stok =
            getIntent().getStringExtra("STOK"); kategori =
            getIntent().getStringExtra("KATEGORI"); foto =
            getIntent().getStringExtra("FOTO"); id =
            getIntent().getStringExtra("ID");
            //Menampilkan detail produk
            mNamaProduk.setText(nama);
            mHargaProduk.setText("Rp. " + harga);
            mStokProduk.setText("Stok: " + stok);
            String url = URLConfig.FOTO_URL + foto.replaceAll(" ",
"%20");
            Picasso.get().load(url).into(mFotoProduk);
            //Ketika button Tambah Ke Troli ditekan
            mButtonTroli.setOnClickListener(new View.OnClickListener()
```

```
@Override
                public void onClick(View view) {
                    //Memanggil fungsi showDialog
                    showDialog();
            });
        }
    }
    //Fungsi deklarasi view yang ada pada
    layout private void initView() {
        mFotoProduk = (ImageView) findViewById(R.id.foto_produk);
        mNamaProduk = (TextView) findViewById(R.id.nama produk);
        mHargaProduk = (TextView) findViewById(R.id.harga_produk);
        mStokProduk = (TextView) findViewById(R.id.stok_produk);
        mButtonTroli = (Button) findViewById(R.id.button troli);
    //Memunculkan popup untuk konfirmasi jumlah
    pembelian private void showDialog() {
        //Membuat dialog dengan kustom layout final Dialog
        dialog = new Dialog(this);
        dialog.requestWindowFeature(Window.FEATURE NO TITLE);
        dialog.setContentView(R.layout.dialog jumlah beli);
        Button min = (Button) dialog.findViewById(R.id.button_min);
        Button plus = (Button) dialog.findViewById(R.id.button_plus);
        Button troli = (Button) dialog.findViewById(R.id.button troli);
        final EditText jumlah = (EditText)
dialog.findViewById(R.id.jumlah);
        //Ketika button minus(-) ditekan
        min.setOnClickListener(new View.OnClickListener() {
            @Override
            public void onClick(View view) {
                //Cek supaya jumlah beli tidak boleh dibawah satu
                if (Integer.parseInt(jumlah.getText().toString()) >
                    1) { //Jumlah berli sekarang dikurang satu
                    int jumlah beli =
Integer.parseInt(jumlah.getText().toString()) - 1;
                    jumlah.setText(String.valueOf(jumlah beli));
            }
        });
        //Ketika button plus(+) ditekan
            plus.setOnClickListener(new View.OnClickListener() {
                @Override
                public void onClick(View view) {
                    if (Integer.parseInt(jumlah.getText().toString()) <</pre>
Integer.parseInt(stok)) {
                         //Jumlah beli sekarang ditambah satu
                        int jumlah beli =
Integer.parseInt(jumlah.getText().toString()) + 1;
                         jumlah.setText(String.valueOf(jumlah beli));
                        Log.d("jumlahbeli",
String.valueOf(jumlah beli));
                        Log.d("jumlahstok", stok);
                        boolean cek = jumlah beli <
Integer.parseInt(stok);
if (Integer.parseInt(jumlah.getText().toString())
== Integer.parseInt(stok)){
```

```
Toast.makeText(ProdukDetailActivity.this,
"Maksimal Stok", Toast. LENGTH_SHORT) . show();
                    }
                }
            });
        troli.setOnClickListener(new View.OnClickListener() {
            @Override
            public void onClick(View view) {
                if (Integer.parseInt(jumlah.getText().toString())
<= Integer.parseInt(stok)) {
                     //Simpan detail pembelian di sqlite
                    Troli item = new Troli(id, nama, kategori,
stok, harga, jumlah.getText().toString(), foto);
                    Troli.save(item);
                    Log.d("cek", "onClick:
"+getIntent().getStringExtra("ID")+" ,
"+Integer.parseInt(jumlah.getText().toString()));
\verb|addTroli(getIntent().getStringExtra("ID"), Integer.|parseInt(jumlah.getTe)||
xt().toString()));
                    dialog.dismiss();
                    Toast.makeText(ProdukDetailActivity.this, "Produk
Anda Berhasil Dimasukkan Ke Troli", Toast. LENGTH SHORT). show();
                }else {
                    Toast.makeText(ProdukDetailActivity.this, "Jumlah
Melebihi stok", Toast.LENGTH_SHORT).show();
        });
        //Tampilkan popup
        dialog.show();
    private void addTroli(String id, int total) {
        APIService service =
RetrofitClient.getClient().create(APIService.class);
        Call<JsonObject> troliCall = service.addTroli(id,
String.valueOf(total));
        troliCall.enqueue(new Callback<JsonObject>() {
            @Override
            public void onResponse(Call<JsonObject> call,
Response<JsonObject> response) {
                Toast.makeText(ProdukDetailActivity.this,
"Produk Berkurang", Toast. LENGTH_SHORT).show();
            @Override
            public void onFailure(Call<JsonObject> call, Throwable t) {
        });
    //Fungsi menampilkan icon di actionbar
    @Override
    public boolean onCreateOptionsMenu(Menu menu) {
```

```
getMenuInflater().inflate(R.menu.main,
       menu); return true;
   @Override
   public boolean onOptionsItemSelected(MenuItem item) {
        int id = item.getItemId();
       if (id == R.id.action cart) {
            Intent intent = new
Intent(ProdukDetailActivity.this, TroliActivity.class);
            startActivity(intent);
            return true;
       return super.onOptionsItemSelected(item);
    @Override
   public boolean onSupportNavigateUp() {
       onBackPressed();
       return true;
   @Override
   public void onBackPressed() {
       super.onBackPressed();
```

TroliActivity.Java

```
package com.example.asus.tugasakhir2.activities;
import android.app.Dialog;
import android.app.ProgressDialog;
import android.content.Intent;
import android.content.SharedPreferences;
import android.os.Bundle;
import android.os.Handler;
import android.support.v7.app.AppCompatActivity; import
android.support.v7.widget.DefaultItemAnimator; import
android.support.v7.widget.DividerItemDecoration; import
android.support.v7.widget.LinearLayoutManager; import
android.support.v7.widget.RecyclerView; import
android.support.v7.widget.Toolbar;
import android.util.Log;
import android.view.View;
import android.view.Window;
import android.widget.ArrayAdapter;
import android.widget.Button;
import android.widget.CompoundButton;
import android.widget.EditText;
import android.widget.RadioButton;
import android.widget.RadioGroup;
import android.widget.Spinner;
import android.widget.TextView;
import android.widget.Toast;
import com.example.asus.tugasakhir2.R;
import com.example.asus.tugasakhir2.adapter.TroliAdapter;
import com.example.asus.tugasakhir2.models.Kurir;
import com.example.asus.tugasakhir2.models.KurirResponse;
import com.example.asus.tugasakhir2.models.Troli;
```

```
import com.example.asus.tugasakhir2.network.APIService; import
com.example.asus.tugasakhir2.network.RetrofitClient; import
com.example.asus.tugasakhir2.route.Distance; import
com.example.asus.tugasakhir2.route.Duration; import
com.example.asus.tugasakhir2.route.EndLocation; import
com.example.asus.tugasakhir2.route.LegsItem; import
com.example.asus.tugasakhir2.route.ResponseRoute; import
com.example.asus.tuqasakhir2.route.network.ApiServices; import
com.example.asus.tugasakhir2.route.network.InitLibrary; import
com.example.asus.tugasakhir2.test.User;
import com.example.asus.tugasakhir2.test.UserResponse;
import com.google.android.gms.maps.model.LatLng;
import com.google.gson.JsonObject; import
com.orm.SugarContext;
import java.util.ArrayList;
import java.util.HashMap;
import java.util.List;
import retrofit2.Call;
import retrofit2.Callback;
import retrofit2.Response;
public class TroliActivity extends AppCompatActivity {
    //Deklarasi variabel global
    private RecyclerView recyclerView;
    private TroliAdapter adapter;
    List<Troli> trolis = new ArrayList<>();
    //private ArrayList<User> listUser;
    public String Telepon;
    public Integer finalTotal = 0;
    public String Alamat;
    public String TeleponKirim;
    public String AlamatKirim;
    public int TeleponLain = 0 ;
    public int AlamatLain = 0 ;
    private TextView mTotal;
    private Button mButtonPengiriman;
    private List<String> namaKurir = new ArrayList<String>();
    private HashMap<String, Integer> idKurir = new HashMap<String,</pre>
Integer>();
    RadioButton rbAlmtLainnya;
    String jarakTempuh;
    int jarakTempuhNilai;
    String durasi;
    int durasiNilai;
    String alamatBeli;
    ProgressDialog progressBar;
    private int progressBarStatus = 0;
    private Handler progressBarHandler = new Handler();
    int response cek =0;
    @Override
    protected void onCreate(Bundle savedInstanceState)
        { super.onCreate(savedInstanceState);
        setContentView(R.layout.activity_troli);
        Toolbar toolbar = (Toolbar) findViewById(R.id.toolbar);
        setSupportActionBar(toolbar);
         RadioButton rbAlmtSekarang = (RadioButton)
findViewById(R.id.rb_al_sekarang);
         rbAlmtLainnya = (RadioButton)
```

```
findViewById(R.id.rb al lainnya);
         RadioButton rbTelpSekarang = (RadioButton)
findViewById(R.id.rb telp sekarang);
         RadioButton rbTelpLainnya = (RadioButton)
findViewById(R.id.rb_telp_lainnya);
        //Menampilkan icon panah kembali di action bar
        getSupportActionBar().setDisplayHomeAsUpEnabled(true);
        getSupportActionBar().setDisplayShowHomeEnabled(true);
        //Mengganti judul action bar
        getSupportActionBar().setTitle("Troli Saya");
        //Memanggil fungsi initView
        initView();
        loadDataUser();
        //Deklarasi dan konfigurasi recyclerview
        recyclerView = (RecyclerView)
findViewById(R.id.recycler troli);
        RecyclerView.LayoutManager mLayoutManager = new
LinearLayoutManager(this);
        recyclerView.setLayoutManager(mLayoutManager);
        recyclerView.setItemAnimator(new DefaultItemAnimator());
        recyclerView.addItemDecoration(new DividerItemDecoration(this,
LinearLayoutManager.VERTICAL));
        //Ambil data dari tabel Troli di sqlite
        SugarContext.init(TroliActivity.this);
        trolis = Troli.listAll(Troli.class);
        //Tampilkan item troli di recyclerview
        adapter = new TroliAdapter(trolis, this);
        recyclerView.setAdapter(adapter);
        //Hitung total keseluruhan harga item
        Integer total = 0;
        for (int 1 = 0; 1 < trolis.size(); 1++) {</pre>
            Troli item = trolis.get(l);
            total = total + Integer.parseInt(item.getJumlah()) *
Integer.parseInt(item.getHarga());
        mTotal.setText("Rp. " + total);
        finalTotal = total;
        mButtonPengiriman.setOnClickListener(new View.OnClickListener()
            @Override
            public void onClick(View view) {
                if (!mTotal.getText().equals("Rp. 0") && finalTotal
>=50000) {
                    //menampilkan dialog form pengiriman
                    showDialogPengiriman();
                }else if(!mTotal.getText().equals("Rp. 0") &&
finalTotal < 50000 ) {
                    Toast.makeText(TroliActivity.this, "Minimal Belanja
Adalah Rp 50.000", Toast. LENGTH SHORT).show();
                }else if(mTotal.getText().equals("Rp. 0")){
                    Toast.makeText(TroliActivity.this, "Troli Belanjaan
Anda Kosong", Toast.LENGTH_SHORT).show();
            }
        });
```

```
//Fungsi deklarasi view yang ada pada
    layout private void initView() {
        mTotal = (TextView) findViewById(R.id.total);
        mButtonPengiriman = (Button)
findViewById(R.id.button pengiriman);
    private void showDialogPengiriman() { //Membuat dialog
        dengan kustom layout final Dialog dialog = new
        Dialog(this);
        dialog.requestWindowFeature(Window. FEATURE NO TITLE);
        dialog.setContentView(R.layout.dialog pengiriman);
        final Spinner mKurir = (Spinner)
dialog.findViewById(R.id.spn kurir);
        loadDataUser();
        loadListKurir(mKurir);
        final EditText mAlamat = (EditText)
dialog.findViewById(R.id.et alamat);
        final EditText mTelepon = (EditText)
dialog.findViewById(R.id.et_telepon);
        final RadioGroup rgAlamat = (RadioGroup)
dialog.findViewById(R.id.rg alamat);
        final RadioGroup rgTelp = (RadioGroup)
dialog.findViewById(R.id.rg telp);
        final RadioButton rbAlmtSekarang = (RadioButton)
dialog.findViewById(R.id.rb_al_sekarang);
        final RadioButton rbAlmtLainnya = (RadioButton)
dialog.findViewById(R.id.rb_al_lainnya);
        final RadioButton rbTelpSekarang = (RadioButton)
dialog.findViewById(R.id.rb telp sekarang);
        final RadioButton rbTelpLainnya = (RadioButton)
dialog.findViewById(R.id.rb_telp_lainnya);
        final TextView tvAlamat = dialog.findViewById(R.id.tvalamat);
        final TextView tvAlamatContent =
dialog.findViewById(R.id.tvalamatsekarangcontent);
         final TextView tvTeleponContent :
dialog.findViewById(R.id.tvteleponsekarangcontent);
        //Menyembunyikan editext alamat dan telepon lainnya
        \verb| mAlamat.setVisibility(View.$\textit{GONE}$)|;
        mTelepon.setVisibility(View.GONE);
        //Mengisi nilai textview alamatcontent & teleponcontent dari
server
        try {
            tvAlamatContent.setText(Alamat.toString().trim());
            tvTeleponContent.setText(Telepon.toString().trim());
        } catch (Exception e) {
            e.printStackTrace();
            tvAlamatContent.setVisibility(View.GONE);
            tvTeleponContent.setVisibility(View.GONE);
        }
        //radiogroup alamat dan telepon
        rgAlamat.setOnCheckedChangeListener(new
RadioGroup.OnCheckedChangeListener() {
            @Override
            public void onCheckedChanged(RadioGroup radioGroup, int i)
```

```
switch (i) {
                    case R.id.rb_al_lainnya:
                        mAlamat.setVisibility(View.VISIBLE);
                        tvAlamatContent.setVisibility(View.GONE);
                        AlamatLain =1;
                        break;
                    case R.id.rb al sekarang:
                        AlamatLain =0;
                        mAlamat.setVisibility(View.GONE);
                        tvAlamatContent.setVisibility(View.VISIBLE);
                        break;
                }
        });
        rgTelp.setOnCheckedChangeListener(ne
w RadioGroup.OnCheckedChangeListener() {
            @Override
            public void onCheckedChanged(RadioGroup radioGroup, int i)
                switch (i) {
                    case R.id.rb telp lainnya:
                        mTelepon.setVisibility(View.VISIBLE);
                        TeleponLain = 1;
                        tvTeleponContent.setVisibility(View.GONE);
                        break;
                    case R.id.rb_telp_sekarang:
                        TeleponLain = 0;
                        mTelepon.setVisibility(View.GONE);
                        tvTeleponContent.setVisibility(View.VISIBLE);
                        break;
                }
        });
        Button mKirim = (Button) dialog.findViewById(R.id.btn kirim);
        mKirim.setOnClickListener(new View.OnClickListener()
            { @Override
            public void onClick(View view) {
                //Melakukan pengecekan apakah menggunakan
Radio Sekarang atau Lainnya
                //Mengisikan nilai TeleponKirim & Alamat Kirim yg akan
dipesan
                if (TeleponLain ==1) {
                     //Jika menggunakan Telepon Lainnya , mengambil dari
EditText
                    TeleponKirim = mTelepon.getText().toString();
                }else {
                     //Jika menggunakan Telepon Sekarang ,mengambil dari
Server
                    TeleponKirim = Telepon;
                if (AlamatLain == 1) {
                     //Jika menggunakan Alamat Lainnya ,mengambil dari
Edit.Text.
                    AlamatKirim = mAlamat.getText().toString().trim();
                }else {
                     //Jika menggunakan Alamat Sekarang ,mengambil dari
```

```
Server
                      AlamatKirim = Alamat;
                  // Membuat Progress Bar & New Thread untuk proses
retrofit get routes
                 progressBar = new ProgressDialog(view.getContext());
                 progressBar.setMessage("Please Wait ...");
                 progressBar.setCancelable(false);
progressBar.setProgressStyle(ProgressDialog.STYLE_HORIZONTAL);
                 progressBar.setProgress(0);
                 progressBar.setMax(100);
                 progressBar.show();
                 new Thread(new Runnable() {
                      public void run() {
                           //Proses Retrofit get data routes
                          while (progressBarStatus < 100) {</pre>
                               progressBarStatus = doOperation();
                               try {Thread.sleep(1000);} catch
(InterruptedException e) {e.printStackTrace();}
                              progressBarHandler.post(new Runnable() {
                                   public void run() {
progressBar.setProgress(progressBarStatus);
                               });
                           }
                          if (progressBarStatus >= 100) {
                               try {
                                   Thread. sleep (1000);
                               } catch (InterruptedException e) {
                                   e.printStackTrace();
                               progressBar.dismiss();
                               progressBarStatus =0;
                               //masuk ke ui thread
                               TroliActivity.this.runOnUiThread(new
Runnable() {
                                   public void run() {
                                       if (response_cek == 100 &&
jarakTempuhNilai >= 20500) {
                                            showDialogJarakTempuh();
                                        }
 \textbf{else if(response\_cek} \ = \ 100 \ \&\& \ \textbf{jarakTempuhNilai} \ > \ 0 \ \&\& \ \textbf{jarakTempuhNilai} \ < \ 20500) \ \{ \ \textbf{secondary} \ \} 
SugarContext.init(TroliActivity.this);
                                            trolis =
Troli.listAll(Troli.class);
                                            for (int 1 = 0; 1 <
trolis.size(); l++) {
                                                 SharedPreferences
preferences = getSharedPreferences("SESSION", MODE PRIVATE);
```

```
Troli item = trolis.get(l);
                                             addPesanan (
preferences.getString("ID", ""),
String.valueOf(idKurir.get(mKurir.getSelectedItem().toString())),
item.getId produk(),
AlamatKirim.toString().trim(),
String.valueOf(Integer.parseInt(item.getJumlah())
* Integer.parseInt(item.getHarga())),
                                                     "0",
TeleponKirim.toString().trim(),
                                                     item.getJumlah());
                                         response cek = 0 ;
                                         jarakTempuhNilai = 0;
                                     else if (response_cek == 404 ) {
                                         //Koneksi retrofit gagal
Toast.makeText(TroliActivity.this, "Koneksi Gagal",
Toast. LENGTH SHORT) . show();
                                     else if(response cek == 105) {
                                         //alamat pengiriman tidak
ditemukan
Toast.makeText(TroliActivity.this, "Alamat tidak
ditemukan", Toast.LENGTH SHORT).show();
                                     Thread.interrupted();
                            });
                        }
                }).start();
                dialog.dismiss();
        });
        dialog.show();
    public int doOperation() {
        //Proses Retrofit get data routes
        getRoutes(AlamatKirim);
        return 100;
    //Dialog Jarak Tempuh
    private void showDialogJarakTempuh() {
        final Dialog dialog2 = new Dialog(TroliActivity.this);
        dialog2.requestWindowFeature(Window. FEATURE NO TITLE);
```

```
dialog2.setContentView(R.layout.dialog jarak pengiriman);
        dialog2.show();
        response_cek = 0 ;
        //init view
        TextView tvAlamatPesan = (TextView)
dialog2.findViewById(R.id.tv_alamat_pesan);
        TextView tvJarak = (TextView)
dialog2.findViewById(R.id.tv jarak);
        TextView tvDurasi = (TextView)
dialog2.findViewById(R.id.tv durasi);
        Button btTutup = (Button) dialog2.findViewById(R.id.bt tutup);
        if (!alamatBeli.isEmpty() && !jarakTempuh.isEmpty() &&
!durasi.isEmpty()) {
            Log. d("suksesStatus", String. valueOf(!alamatBeli.isEmpty()
&& !jarakTempuh.isEmpty() && !durasi.isEmpty()));
            tvAlamatPesan.setText(alamatBeli.toString().trim());
            tvJarak.setText(jarakTempuh);
            tvDurasi.setText(durasi);
        }
        btTutup.setOnClickListener(new View.OnClickListener()
            { @Override
            public void onClick(View view) {
                dialog2.dismiss();
                jarakTempuhNilai = 0;
        });
    }
    //mendapatkan route ,distance dari api routes google maps
    private void getRoutes(final String alamatPembeli) {
        final TextView tvJarak = (TextView)
findViewById(R.id.tvalamat);
        String API KEY = "AlzaSyCkMKgyPRcTK0Eb-2KbWUcm QzdYHTBETY";
        String lokasiAwal = "Jl. Dinar Asri Raya/Dinar MAS Raya,
Meteseh"; // Alamat Toko
        String lokasiAkhir = AlamatKirim; // Alamat Pembeli
        // Panggil Retrofit
        ApiServices api = InitLibrary.getInstance();
        // Siapkan request
        Call<ResponseRoute> routeRequest =
api.request route(lokasiAwal, lokasiAkhir, API KEY);
        // \overline{k}irim request
        routeRequest.enqueue(new Callback<ResponseRoute>()
            { @Override
            public void onResponse(Call<ResponseRoute>
call, Response<ResponseRoute> response) {
                if (response.isSuccessful()){
                     // tampung response ke variable
                    ResponseRoute dataDirection =
                    response.body(); response_cek =105;
```

```
try{
                        LegsItem dataLegs =
dataDirection.getRoutes().get(0).getLegs().get(0);
                        Distance dataDistance = dataLegs.getDistance();
                        Duration dataDuration = dataLegs.getDuration();
                        // Dapatkan jarak dan
                        waktu response cek = 100 ;
                        jarakTempuh = dataDistance.getText();
                        jarakTempuhNilai = dataDistance.getValue();
                        durasi = dataDuration.getText(); durasiNilai
                         dataLegs.getEndAddress();
                        Log.d("sukses",
dataDistance.getText()+dataDuration.getText()+dataLegs.getEndAddress()+
dataLegs.getStartAddress());
                        Toast.makeText(TroliActivity.this, "Jarak :
"+dataDistance.getText() + " ( "+dataDistance.getValue()+" m )",
Toast. LENGTH SHORT) . show();
                    }catch (IndexOutOfBoundsException e) {
                        e.printStackTrace();
                    Log. d("suksesStatusResponse1",
String.valueOf(response_cek));
                }else {
                    response_cek =404;
            }
            @Override
            public void onFailure(Call<ResponseRoute> call, Throwable
t) {
                t.printStackTrace();
                Log.d("gagal", "gagal");
                response_cek =404;
            }
        });
    //mengambil data alamat dan telepon
    private void loadDataUser() {
        SharedPreferences preferences =
getSharedPreferences("SESSION", MODE PRIVATE)
        String id = preferences.getString("ID", "");
       APIService requestData =
RetrofitClient.getClient().create(APIService.class);
       Call<UserResponse> userRespos = requestData.getAlamat(id);
        userRespos.enqueue (new Callback<UserResponse>()
            { @Override
            public void onResponse(Call<UserResponse> call,
Response<UserResponse> response) {
               if (response.isSuccessful()){
                    Alamat = (String)
```

response.body().getUser().getAlamat();

```
Telepon = (String)
  response.body().getUser().getNoTelp();
             }
             @Override
            public void onFailure(Call<UserResponse> call, Throwable t)
{
        });
    public void loadListKurir(final Spinner mKurir) {
        APIService service =
RetrofitClient.getClient().create(APIService.class);
        Call<KurirResponse> userCall = service.getAllKurir();
        userCall.enqueue(new Callback<KurirResponse>()
             { @Override
            public void onResponse(Call<KurirResponse>
call, Response<KurirResponse> response) {
                 if (response.isSuccessful()){
                     List<Kurir> kurirs = response.body().getData();
                     namaKurir.add("Pilih Kurir");
                     idKurir.put("Pilih Kurir", 0);
                     for (int i=0; i<kurirs.size(); i++) {</pre>
                         namaKurir.add(kurirs.get(i).getName());
                         idKurir.put(kurirs.get(i).getName(),
kurirs.get(i).getId());
                     ArrayAdapter<String> dataAdapter = new
ArrayAdapter<String>(TroliActivity.this,
android.R.layout.simple_spinner_item, namaKurir);
dataAdapter.setDropDownViewResource(android.R.layout.simple spinner dro
pdown_item);
                     mKurir.setAdapter(dataAdapter);
                 }
            @Override
            public void onFailure(Call<KurirResponse> call, Throwable
t) {
        });
    public void addPesanan(String id_user, String id_kurir, String
id_produk, String alamat_pesanan, String total, String
status_pesanan, String telp_pesanan, String jumlah) {
        mButtonPengiriman.setEnabled(false);
        mButtonPengiriman.setText("memproses...");
        APIService service =
{\tt RetrofitClient.} \textit{getClient()}. \texttt{create(APIService.} \textbf{class);}
        Call<JsonObject> userCall = service.addPesanan(id user,
id_kurir, id_produk, alamat_pesanan, total, status_pesanan,
telp_pesanan, jumlah);
        userCall.enqueue(new Callback<JsonObject>() {
```

```
@Override
            public void onResponse(Call<JsonObject> call,
Response<JsonObject> response) {
                if (response.isSuccessful()){
                    mButtonPengiriman.setEnabled(true);
                    mButtonPengiriman.setText("lanjutkan ke
pengiriman");
                    Troli.deleteAll(Troli.class);
                    Toast.makeText(TroliActivity.this, "Pesanan
Anda Telah Kami Proses", Toast. LENGTH_LONG).show();
                    TroliActivity.this.recreate();
            }
            @Override
            public void onFailure(Call<JsonObject> call, Throwable t) {
                mButtonPengiriman.setEnabled(true);
                mButtonPengiriman.setText("lanjutkan ke pengiriman");
                Toast.makeText(TroliActivity.this, "Gagal Tehubung Ke
Server", Toast. LENGTH SHORT). show();
            }
        });
public void updateTotal(List<Troli>
        troliList) { Integer total = 0;
        for (int 1 = 0; 1 < troliList.size(); 1++)</pre>
            { Troli item = troliList.get(l);
            total = total + Integer.parseInt(item.getJumlah()) *
Integer.parseInt(item.getHarga());
        mTotal.setText("Rp. " + total);
        finalTotal = total;
    //Fungsi ketika icon panah kembali
    ditekan @Override
    public boolean onSupportNavigateUp() {
        onBackPressed();
        return true;
    @Override
    public void onBackPressed() {
        Intent intent = new Intent(TroliActivity.this,
MainActivity.class);
        startActivity(intent);
        finish();
        super.onBackPressed();
}
```

URLConfig.Java

```
package com.example.asus.tugasakhir2.config;
public class URLConfig {
    public static final String BASE_URL =
    "http://antargen.informatikapolines.com/";
    public static final String FOTO_URL = BASE_URL +
    "laravel/storage/app/images/produk/";
}
```

APIService.Java

```
package com.example.asus.tugasakhir2.network;
import com.example.asus.tugasakhir2.models.KurirResponse;
import com.example.asus.tugasakhir2.models.LoginResponse;
import com.example.asus.tugasakhir2.models.PesananResponse;
import com.example.asus.tugasakhir2.models.ProdukResponse;
import com.example.asus.tugasakhir2.models.PromoResponse;
import com.example.asus.tugasakhir2.test.UserResponse;
import com.google.gson.JsonObject;
import retrofit2.Call;
import retrofit2.http.Field;
import retrofit2.http.FormUrlEncoded;
import retrofit2.http.GET;
import retrofit2.http.POST;
import retrofit2.http.Path;
 * Created by ASUS on 5/24/2018.
public interface APIService {
    @FormUrlEncoded
    @POST("api/auth/login")
    Call<LoginResponse> userLogin(
            @Field("email") String email,
            @Field("password") String password);
    @GET("api/user/{id}")
    Call<UserResponse> getAlamat(@Path("id") String id);
    @FormUrlEncoded
    @POST("api/produk/troli")
    Call<JsonObject> addTroli(
            @Field("id produk") String id produk,
            @Field("jumlah") String total);
    @FormUrlEncoded
    @POST("api/produk/untroli")
    Call<JsonObject> removeTroli(
            @Field("id produk") String id produk,
            @Field("jumlah") String total);
    @GET("api/produk")
    Call<ProdukResponse> getProduk();
    @GET("api/promo")
    Call<PromoResponse> getPromo();
    @GET("api/makanan")
    Call<ProdukResponse> getMakanan();
    @GET("api/minuman")
    Call<ProdukResponse> getMinuman();
    @GET("api/kesehatan")
    Call<ProdukResponse> getKesehatan();
    @GET("api/krt")
    Call<ProdukResponse> getKrt();
    @FormUrlEncoded
    @POST("api/pesanan")
```

```
Call<JsonObject> addPesanan(
    @Field("id_user") String id_user,
    @Field("id_kurir") String id_kurir,
    @Field("id_produk") String id_produk,
    @Field("alamat_pesanan") String alamat_pesanan,
    @Field("total") String total,
    @Field("status_pesanan") String status_pesanan,
    @Field("telp_pesanan") String telp_pesanan,
    @Field("jumlah") String jumlah);

@GET("api/kurir")
Call<KurirResponse> getAllKurir();

@GET("api/pesanan/{id}")
Call<PesananResponse> getPesanan(@Path("id") int id);
}
```

RetrofitClient.Java

```
package com.example.asus.tugasakhir2.network;
import android.content.Context;
import com.example.asus.tugasakhir2.config.URLConfig;
import com.google.gson.Gson; import
com.google.gson.GsonBuilder;
import retrofit2.Retrofit;
import retrofit2.converter.gson.GsonConverterFactory;
public class RetrofitClient {
    public static final String BASE_URL = URLConfig.BASE URL;
    private static Retrofit retrofit = null;
    // variable to hold context
    private static Context context;
    public static Retrofit getClient() {
        if (retrofit==null) {
            Gson gson = new GsonBuilder()
                    .setLenient()
                     .create();
            retrofit = new Retrofit.Builder()
                    .baseUrl(BASE_URL)
.addConverterFactory(GsonConverterFactory.create(gson))
        return retrofit;
}
```

Activity_loginutama.xml

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

```
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"
    xmlns:fancy="http://schemas.android.com/apk/res-auto"
    android:layout_width="match_parent"
    android:layout height="match parent"
    android:background="#f7ebd3"
    tools:context="com.example.asus.tugasakhir2.activities.loginutama">
    <LinearLayout</pre>
        android:layout_width="match_parent"
        android: layout height="wrap content"
        android:layout_centerHorizontal="true"
        android:layout centerVertical="true"
        android:orientation="vertical"
        android:padding="28dp"
        android:background="#f7ebd3" >
        <ImageView</pre>
            android:id="@+id/logo"
            android:layout_width="100dp"
            android:layout_height="100dp"
            android:layout gravity="center"
            android:background="@drawable/icon4" />
        <com.rengwuxian.materialedittext.MaterialEditTex</pre>
            t android:layout_marginLeft="16dp"
            android:id="@+id/txt email"
            android:layout marginTop="16dp"
            android:layout_marginRight="16dp"
            android:layout width="match parent"
            android:layout_height="wrap_content"
            app:met floatingLabel="highlight"
            android:hint="Email"/>
        <com.rengwuxian.materialedittext.MaterialEditTex</pre>
            t android:layout marginLeft="16dp"
            android:inputType="textPassword"
            android:id="@+id/txt password"
            android:layout marginTop="8dp"
            android:layout marginRight="16dp"
            android:layout width="match parent"
            android:layout_height="wrap_content"
            android:layout marginBottom="16dp"
            app:met_floatingLabel="highlight"
            android:hint="Password"/>
        <Button
            android:id="@+id/btn login"
            android:layout_width="wrap_content"
            android:layout height="wrap content"
            android:layout gravity="center"
            android:layout marginBottom="15dp"
            android:text="Login"
            android:onClick="MainActivity"
            android:background="#FFFF6600"
            android:textColor="#fff"
            android:textStyle="bold" />
    </LinearLayout>
</RelativeLayout>
```

```
<?xml version="1.0" encoding="utf-8"?>
<android.support.v4.widget.DrawerLayout</pre>
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/drawer layout"
    android: layout width="match parent"
    android: layout height="match_parent"
    android: fitsSystemWindows="true"
    tools:openDrawer="start">
    <include
        layout="@layout/app_bar_main"
        android:layout width="match parent"
        android:layout_height="match_parent" />
    <android.support.design.widget.NavigationView</pre>
        android:id="@+id/nav view"
        android:layout width="wrap content"
        android:layout height="match parent"
        android:layout_gravity="start"
        android:fitsSystemWindows="true"
        app:headerLayout="@layout/nav header main"
        app:menu="@menu/activity_main_drawer" />
</android.support.v4.widget.DrawerLayout>
```

Activity_pesanansaya.xml

```
<?xml version="1.0" encoding="utf-8"?>
<LinearLayout</pre>
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:layout width="match parent"
    android:layout_height="match_parent"
    android:orientation="vertical"
    android:background="@color/grey"
    tools:context=".activities.PesananSayaActivity">
    <android.support.design.widget.AppBarLayout</pre>
        android:id="@+id/appbar"
        android:layout width="match parent"
        android:layout height="?attr/actionBarSize"
        android:orientation="vertical"
        android: theme="@style/ThemeOverlay.AppCompat.Dark.ActionBar">
        <android.support.v7.widget.Toolbar</pre>
            xmlns:android="http://schemas.android.com/apk/res/android"
            android:id="@+id/toolbar"
            android:layout_width="match_parent"
            android:layout_height="?attr/actionBarSize">
        </android.support.v7.widget.Toolbar>
    </android.support.design.widget.AppBarLayout>
    <android.support.v7.widget.RecyclerView</pre>
        android:id="@+id/recycler_pesanan"
        android:layout width="match parent"
        android:layout_height="match_parent"/>
</LinearLayout>
```

Activity_troli.xml

```
<?xml version="1.0" encoding="utf-8"?>
<LinearLayout</pre>
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:layout_width="match_parent"
    android:layout height="match parent"
    android:orientation="vertical"
    android:background="@color/grey"
    tools:context=".activities.PesananSayaActivity">
    <android.support.design.widget.AppBarLayout</pre>
        android:id="@+id/appbar"
        android:layout_width="match_parent"
        android:layout height="?attr/actionBarSize"
        android:orientation="vertical"
        android: theme="@style/ThemeOverlay.AppCompat.Dark.ActionBar">
        <android.support.v7.widget.Toolbar</pre>
            xmlns:android="http://schemas.android.com/apk/res/android"
            android:id="@+id/toolbar"
            android:layout width="match parent"
            android:layout height="?attr/actionBarSize">
        </android.support.v7.widget.Toolbar>
    </android.support.design.widget.AppBarLayout>
    <android.support.v7.widget.RecyclerView</pre>
        android:id="@+id/recycler pesanan"
        android:layout_width="match_parent"
        android:layout_height="match_parent"/>
</LinearLayout>
```

Troli item.xml

```
<?xml version="1.0" encoding="utf-8"?>
< Relative Layout
    android:id="@+id/container"
    xmlns:android="http://schemas.android.com/apk/res/android"
    android: orientation="horizontal"
android:layout_width="match_parent"
    android:layout height="wrap content">
    <ImageView</pre>
        android:id="@+id/foto"
        android:layout_width="100dp"
        android:layout height="100dp"
        android:src="@drawable/ic_shopping_basket"/>
    <LinearLayout</pre>
        android:layout width="match parent"
        android:layout height="wrap content"
        android:layout toRightOf="@+id/foto"
        android:layout toEndOf="@+id/foto"
        android:orientation="vertical"
        android:layout margin="8dp">
        <TextView
            android:id="@+id/nama"
            android:layout_width="wrap_content"
```

```
android:layout height="wrap content"
            android:textColor="@color/black"
            android:textSize="18dp"
            android:text="Nama Produk"/>
        <TextView
            android:id="@+id/kategori"
            android:layout_width="wrap_content"
            android:layout height="wrap content"
            android:textSize="14dp"
            android:text="Tipe Produk"/>
        <TextView
            android:id="@+id/harga"
            android:layout_width="wrap_content"
            android:layout_height="wrap_content"
            android:textColor="@color/orange"
            android:textStyle="bold"
            android:textSize="16dp"
            android:text="8 x Rp. 2000"/>
        <TextView
            android:id="@+id/total"
            android:layout width="wrap_content"
            android:layout height="wrap content"
            android:textColor="@color/orange"
            android:textStyle="bold"
            android:textSize="18dp"
            android:text="Rp. 16000"/>
    </LinearLayout>
    <ImageView</pre>
        android:id="@+id/delete"
        android:layout width="36dp"
        android:layout height="36dp"
        android:layout_alignParentEnd="true"
        android:layout_alignParentRight="true"
        android:layout margin="8dp"
        android: tint="@color/colorPrimary"
        android:src="@drawable/ic_delete"/>
</RelativeLayout>
```

B. KODE PROGRAM KURIR

AndroidManifest.xml

```
<intent-filter>
                <action android:name="android.intent.action.MAIN" />
                <category</pre>
android:name="android.intent.category.LAUNCHER" />
            </intent-filter>
        </activity>
        <activity
            android:name=".activities.HalamanUtama"
            android:label="@string/title activity halaman utama"
            android: theme="@style/AppTheme.NoActionBar" />
        <service
            android:name=".services.TrackerService"
            android:enabled="true"
            android:exported="true" />
        <!-- Firebase Notifications -->
        <service android:name=".services.MyFirebaseMessagingService">
            <intent-filter>
                <action
android:name="com.google.firebase.MESSAGING_EVENT" />
            </intent-filter>
        </service>
        <service android:name=".services.MyFirebaseInstanceIDService">
            <intent-filter>
                <action
android:name="com.google.firebase.INSTANCE ID EVENT" />
            </intent-filter>
        </service>
        <!-- ./Firebase Notifications -->
        <service android:name=".services.RefreshFirebaseTokenService"</pre>
/>
    </application>
</manifest>
```

HistoriAdapter.Java

```
package id.web.proditipolines.kurir2.adapters;
import android.app.Dialog;
import android.content.Context;
import android.content.Intent;
import android.net.Uri;
import android.support.v7.widget.RecyclerView;
import android.view.LayoutInflater; import
android.view.View;
import android.view.ViewGroup;
import android.view.Window;
import android.widget.Button;
import android.widget.LinearLayout;
import android.widget.TextView;
import java.util.List;
import id.web.proditipolines.kurir2.R;
import id.web.proditipolines.kurir2.fragments.HistoriFragment;
import id.web.proditipolines.kurir2.models.Pengiriman;
import id.web.proditipolines.kurir2.network.APIService;
import id.web.proditipolines.kurir2.network.RetrofitClient;
import retrofit2.Call;
import retrofit2.Callback;
```

```
import retrofit2.Response;
public class HistoriAdapter extends
RecyclerView.Adapter<HistoriAdapter.MyViewHolder> {
    private List<Pengiriman> historiList;
    private HistoriFragment fragment;
    public class MyViewHolder extends RecyclerView.ViewHolder
        { TextView mTvKode;
        TextView mTvKurir;
        TextView mTvTanggal;
        TextView mTvProduk;
        TextView mTvJumlah;
        TextView mTvStatus;
        LinearLayout container;
        public MyViewHolder(View view) {
            super(view);
            mTvKode = (TextView) itemView.findViewById(R.id.tv kode);
            mTvKurir = (TextView) itemView.findViewById(R.id.tv_kurir);
            mTvTanggal = (TextView)
itemView.findViewById(R.id.tv tanggal);
            mTvProduk = (TextView)
itemView.findViewById(R.id.tv produk);
            mTvJumlah = (TextView)
itemView.findViewById(R.id.tv jumlah);
            mTvStatus = (TextView)
  itemView.findViewById(R.id.tv status);
            container = (LinearLayout)
  itemView.findViewById(R.id.container);
    public HistoriAdapter(List<Pengiriman> historiList,
HistoriFragment fragment) {
        this.historiList = historiList;
        this.fragment = fragment;
    @Override
    public MyViewHolder onCreateViewHolder(ViewGroup parent, int
viewType) {
        View itemView = LayoutInflater.from(parent.getContext())
                .inflate(R.layout.pengiriman_item, parent, false);
        return new MyViewHolder(itemView);
    }
    @Override
    public void onBindViewHolder(final MyViewHolder holder, final int
position) {
        final Pengiriman pengiriman = historiList.get(position);
        final Context mContext = holder.mTvKode.getContext();
        holder.mTvKode.setText("Pengiriman #" +
pengiriman.getKodePesanan());
        holder.mTvKurir.setText(pengiriman.getNamaPelanggan());
holder.mTvTanggal.setText(pengiriman.getUpdatedAt().substring(0, 10));
        holder.mTvProduk.setText("");
        String valueProduk = "";
        for (int i = 0; i < pengiriman.getNamaProduk().size(); i++) {</pre>
            if (i == pengiriman.getNamaProduk().size() - 1) {
                valueProduk = valueProduk + "•" +
```

```
pengiriman.getNamaProduk().get(i);
            } else {
                valueProduk = valueProduk + "•" +
pengiriman.getNamaProduk().get(i) + "\n";
        holder.mTvProduk.setText(valueProduk);
        String valueJumlah = "";
        for (int i = 0; i < pengiriman.getJumlah().size(); i++)</pre>
            { if (i == pengiriman.getTotal().size() - 1) {
                valueJumlah = valueJumlah + "Qty: "
+ pengiriman.getJumlah().get(i);
            } else {
                valueJumlah = valueJumlah + "Qty: " +
pengiriman.getJumlah().get(i) + "\n";
            }
        holder.mTvJumlah.setText(valueJumlah);
        if (pengiriman.getStatus().equals("0")){
            holder.mTvStatus.setText("Sedang Diproses");
        } else if (pengiriman.getStatus().equals("1")){
        holder.mTvStatus.setText("Pickup Barang"); }
        else if (pengiriman.getStatus().equals("2")){
           holder.mTvStatus.setText("Dalam Pengiriman");
        } else {
            holder.mTvStatus.setText("Telah Diterima");
        final String finalValueProduk = valueProduk; final String
        finalValueJumlah = valueJumlah;
        holder.container.setOnClickListener(new View.OnClickListener()
{
            @Override
            public void onClick(View view) {
                //Membuat dialog dengan kustom layout
                final Dialog dialog = new Dialog(mContext);
                dialog.requestWindowFeature(Window.FEATURE NO TITLE);
                dialog.setContentView(R.layout.dialog pengiriman);
                TextView nama = (TextView)
dialog.findViewById(R.id.nama);
                TextView telepon = (TextView)
dialog.findViewById(R.id.telepon);
                TextView alamat = (TextView)
dialog.findViewById(R.id.alamat);
                TextView produk = (TextView)
dialog.findViewById(R.id.produk);
                TextView jumlah = (TextView)
dialog.findViewById(R.id.jumlah);
                TextView tagihan = (TextView)
dialog.findViewById(R.id.tagihan);
                Button message = (Button)
dialog.findViewById(R.id.message);
                Button call = (Button) dialog.findViewById(R.id.call);
                final Button aksi = (Button)
dialog.findViewById(R.id.btn_aksi);
                nama.setText(": " + pengiriman.getNamaPelanggan());
                telepon.setText(": " + pengiriman.getTelepon());
                alamat.setText(": " + pengiriman.getAlamat());
                produk.setText(finalValueProduk);
                jumlah.setText(finalValueJumlah); Integer
                totalTagihan = 0;
                for (int i = 0; i < pengiriman.getTotal().size(); i++)</pre>
```

```
totalTagihan = totalTagihan +
Integer.parseInt(pengiriman.getTotal().get(i));
                tagihan.setText("Total Tagihan: Rp. "
+ String.valueOf(totalTagihan));
                message.setOnClickListener(new View.OnClickListener() {
                    @Override
                    public void onClick(View view) {
                        Intent smsIntent = new
Intent(Intent.ACTION VIEW);
                        smsIntent.setData(Uri.parse("smsto:" +
pengiriman.getTelepon());
                        mContext.startActivity(smsIntent);
                });
                call.setOnClickListener(new View.OnClickListener()
                    { @Override
                    public void onClick(View view) {
                        Intent intent = new Intent(Intent.ACTION DIAL,
Uri.fromParts("tel", pengiriman.getTelepon(), null));
                        mContext.startActivity(intent);
                });
                aksi.setVisibility(View.GONE);
                dialog.show();
        });
    }
    @Override
    public int getItemCount() {
        return historiList.size();
   public void updateStatusPesanan(final Dialog dialog, int kode,
String status) {
        APIService service =
RetrofitClient.getClient().create(APIService.class);
        Call<String> userCall = service.updateStatusPesanan(kode,
status);
        userCall.enqueue(new Callback<String>() {
            @Override
            public void onResponse(Call<String> call, Response<String>
response) {
                dialog.dismiss();
                fragment.onRefresh();
            @Override
            public void onFailure(Call<String> call, Throwable t) {
                dialog.dismiss();
                fragment.onRefresh();
        });
    }
}
```

PengirimanAdapter.Java

```
package id.web.proditipolines.kurir2.adapters;
import android.app.Dialog;
import android.content.Context;
import android.content.Intent;
import android.net.Uri;
import android.support.v7.widget.CardView;
import android.support.v7.widget.RecyclerView;
import android.view.LayoutInflater; import
android.view.View;
import android.view.ViewGroup;
import android.view.Window;
import android.widget.Button;
import android.widget.LinearLayout;
import android.widget.RelativeLayout;
import android.widget.TextView;
import android.widget.Toast;
import java.net.URI;
import java.util.List;
import id.web.proditipolines.kurir2.R;
import id.web.proditipolines.kurir2.fragments.PengirimanFragment;
import id.web.proditipolines.kurir2.models.Pengiriman;
import id.web.proditipolines.kurir2.network.APIService;
import id.web.proditipolines.kurir2.network.RetrofitClient;
import id.web.proditipolines.kurir2.services.TrackerService;
import retrofit2.Call;
import retrofit2.Callback;
import retrofit2.Response;
public class PengirimanAdapter extends
RecyclerView.Adapter<PengirimanAdapter.MyViewHolder>
    { private List<Pengiriman> pengirimanList;
    private PengirimanFragment fragment;
    public class MyViewHolder extends RecyclerView.ViewHolder
        { TextView mTvKode;
        TextView mTvKurir;
        TextView mTvTanggal;
        TextView mTvProduk;
        TextView mTvJumlah;
        TextView mTvStatus;
        TextView mTvMaps;
        LinearLayout container;
        CardView cvMaps;
        public MyViewHolder(View view) {
            super(view);
            mTvKode = (TextView) itemView.findViewById(R.id.tv kode);
            mTvKurir = (TextView) itemView.findViewById(R.id.tv kurir);
            mTvTanggal = (TextView)
itemView.findViewById(R.id.tv tanggal);
            mTvProduk = (TextView)
itemView.findViewById(R.id.tv produk);
            mTvJumlah = (TextView)
itemView.findViewById(R.id.tv_jumlah);
            mTvStatus = (TextView)
itemView.findViewById(R.id.tv status);
            mTvMaps = (TextView) itemView.findViewById(R.id.tvMaps);
            container = (LinearLayout)
itemView.findViewById(R.id.container);
            cvMaps = (CardView) itemView.findViewById(R.id.cvMaps);
```

```
public PengirimanAdapter(List<Pengiriman>
pengirimanList, PengirimanFragment fragment) {
        this.pengirimanList = pengirimanList;
        this.fragment = fragment;
    @Override
   public MyViewHolder onCreateViewHolder(ViewGroup parent, int
viewType) {
        View itemView = LayoutInflater.from(parent.getContext())
                .inflate(R.layout.pengiriman_item, parent, false);
        return new MyViewHolder(itemView);
    @Override
    public void onBindViewHolder(final MyViewHolder holder, final int
position) {
        final Pengiriman pengiriman = pengirimanList.get(position);
        final Context mContext = holder.mTvKode.getContext();
        holder.mTvKode.setText("Pengiriman #" +
pengiriman.getKodePesanan());
        final String alamatMaps = pengiriman.getAlamat();
        holder.mTvKurir.setText(pengiriman.getNamaPelanggan())
holder.mTvTanggal.setText(pengiriman.getUpdatedAt().substring(0, 10));
        holder.mTvProduk.setText("");
        String valueProduk = "";
        for (int i = 0; i < pengiriman.getNamaProduk().size(); i++) {</pre>
            if (i == pengiriman.getNamaProduk().size() - 1) {
                valueProduk = valueProduk + "•" +
pengiriman.getNamaProduk().get(i);
            } else {
                valueProduk = valueProduk + "•" +
pengiriman.getNamaProduk().get(i) + "\n";
        holder.mTvProduk.setText(valueProduk);
        String valueJumlah = "";
        for (int i = 0; i < pengiriman.getJumlah().size(); i++)</pre>
            { if (i == pengiriman.getTotal().size() - 1) {
                valueJumlah = valueJumlah + "Qty: "
+ pengiriman.getJumlah().get(i);
            } else {
                valueJumlah = valueJumlah + "Qty: " +
pengiriman.getJumlah().get(i) + "\n";
            }
        }
        holder.mTvJumlah.setText(valueJumlah);
        if (pengiriman.getStatus().equals("0")){
            holder.mTvStatus.setText("Sedang Diproses");
        } else if (pengiriman.getStatus().equals("1")){
        holder.mTvStatus.setText("Pickup Barang"); }
        else if (pengiriman.getStatus().equals("2")){
            holder.mTvStatus.setText("Dalam Pengiriman");
        } else {
            holder.mTvStatus.setText("Telah Diterima");
```

```
final String finalValueProduk = valueProduk;
        final String finalValueJumlah = valueJumlah;
        holder.cvMaps.setOnClickListener(new View.OnClickListener() {
            @Override
            public void onClick(View view) {
                /*Deklarasi variable*/
                Button btn navigasi;
                String goolgeMap = "com.google.android.apps.maps"; //
identitas package aplikasi google masps android
                Uri gmmIntentUri;
                Intent mapIntent;
                String masjid_agung_demak = "-
6.894649906672214,110.63718136399984"; // koordinat Masjid Agung Demak
                /*Deklarasi variable*/
                // Buat Uri dari intent string. Gunakan hasilnya untuk
membuat Intent.
                gmmIntentUri = Uri.parse("google.navigation:q=" +
alamatMaps);
                // Buat Uri dari intent gmmIntentUri. Set action =>
ACTION VIEW
                mapIntent = new Intent (Intent. ACTION VIEW,
gmmIntentUri);
                // Set package Google Maps untuk tujuan aplikasi yang
di Intent yaitu google maps
                mapIntent.setPackage(goolgeMap);
(mapIntent.resolveActivity(mContext.getPackageManager()) != null) {
                    mContext.startActivity(mapIntent);
                } else {
                    Toast.makeText(mContext, "Install Aplikasi Google
Maps Terlebih dahulu", Toast.LENGTH_SHORT).show();
        });
        holder.container.setOnClickListener(new View.OnClickListener()
            @Override
            public void onClick(View view) {
                //Membuat dialog dengan kustom layout
                final Dialog dialog = new Dialog(mContext);
                dialog.requestWindowFeature(Window. FEATURE NO TITLE);
                dialog.setContentView(R.layout.dialog pengiriman);
                TextView nama = (TextView)
dialog.findViewById(R.id.nama);
                TextView telepon = (TextView)
dialog.findViewById(R.id.telepon);
                TextView alamat = (TextView)
dialog.findViewById(R.id.alamat);
                TextView produk = (TextView)
dialog.findViewById(R.id.produk);
                TextView jumlah = (TextView)
dialog.findViewById(R.id.jumlah);
                TextView tagihan = (TextView)
dialog.findViewById(R.id.tagihan);
                Button message = (Button)
dialog.findViewById(R.id.message);
                Button call = (Button) dialog.findViewById(R.id.call);
```

```
final Button aksi = (Button)
dialog.findViewById(R.id.btn_aksi);
                nama.setText(": " + pengiriman.getNamaPelanggan());
                telepon.setText(": " + pengiriman.getTelepon());
                alamat.setText(": " + pengiriman.getAlamat());
                produk.setText(finalValueProduk);
                jumlah.setText(finalValueJumlah); Integer
                totalTagihan = 0;
                for (int i = 0; i < pengiriman.getTotal().size(); i++)</pre>
{
                    totalTagihan = totalTagihan +
Integer.parseInt(pengiriman.getTotal().get(i));
                }
                tagihan.setText("Total Tagihan: Rp. "
+ String.valueOf(totalTagihan));
                message.setOnClickListener(new View.OnClickListener() {
                    @Override
                    public void onClick(View view) {
                         Intent smsIntent = new
Intent(Intent.ACTION VIEW);
                         smsIntent.setData(Uri.parse("smsto:" +
pengiriman.getTelepon());
                        mContext.startActivity(smsIntent);
                });
                call.setOnClickListener(new View.OnClickListener()
                     { @Override
                    public void onClick(View view) {
                        Intent intent = new Intent(Intent.ACTION_DIAL,
Uri.fromParts("tel", pengiriman.getTelepon(), null));
                        mContext.startActivity(intent);
                });
                if (pengiriman.getStatus().equals("0")){
                    aksi.setText("Pickup Barang");
                    aksi.setOnClickListener(new View.OnClickListener()
{
                         @Override
                         public void onClick(View view) {
                             aksi.setEnabled(false);
                             aksi.setText("Memproses...");
                             updateStatusPesanan(dialog,
pengiriman.getKodePesanan(), "1");
                     });
                } else if (pengiriman.getStatus().equals("1")) {
                    aksi.setText("Dalam Pengiriman");
                    aksi.setOnClickListener(new View.OnClickListener()
{
                         @Override
                         public void onClick(View view) {
                             aksi.setEnabled(false);
                             aksi.setText("Memproses...");
                             mContext.startService(new Intent(mContext,
TrackerService.class));
                             updateStatusPesanan (dialog,
pengiriman.getKodePesanan(), "2");
                    });
                } else {
```

```
aksi.setText("Telah Diterima");
                    aksi.setOnClickListener(new View.OnClickListener()
{
                         @Override
                        public void onClick(View view) {
                             aksi.setEnabled(false);
                             aksi.setText("Memproses...");
                            mContext.stopService(new Intent(mContext,
TrackerService.class));
                            updateStatusPesanan (dialog,
pengiriman.getKodePesanan(), "3");
                    });
                dialog.show();
        });
    }
    @Override
    public int getItemCount() {
        return pengirimanList.size();
    public void updateStatusPesanan(final Dialog dialog, int kode,
String status) {
        APIService service =
RetrofitClient.getClient().create(APIService.class);
        Call<String> userCall = service.updateStatusPesanan(kode,
status);
        userCall.enqueue(new Callback<String>() {
            @Override
            public void onResponse(Call<String> call, Response<String>
response) {
                dialog.dismiss();
                fragment.onRefresh();
            @Override
            public void onFailure(Call<String> call, Throwable t) {
                dialog.dismiss();
                fragment.onRefresh();
        });
    }
}
```

Halaman Utama.Java

```
package id.web.proditipolines.kurir2.activities;

import android.Manifest;
import android.content.Intent;
import android.content.SharedPreferences;
import android.content.pm.PackageManager;
import android.location.LocationManager;
import android.os.Bundle;
import android.support.design.widget.FloatingActionButton;
```

```
import android.support.design.widget.NavigationView;
import android.support.design.widget.Snackbar;
import android.support.v4.app.ActivityCompat; import
android.support.v4.app.Fragment;
import android.support.v4.app.FragmentTransaction;
import android.support.v4.content.ContextCompat;
import android.support.v4.view.GravityCompat; import
android.support.v4.widget.DrawerLayout; import
android.support.v7.app.ActionBarDrawerToggle; import
android.support.v7.app.AppCompatActivity; import
android.support.v7.widget.Toolbar; import
android.util.Log;
import android.view.Menu;
import android.view.MenuItem;
import android.view.View;
import android.widget.TextView;
import android.widget.Toast;
import id.web.proditipolines.kurir2.R;
import id.web.proditipolines.kurir2.fragments.HistoriFragment;
import id.web.proditipolines.kurir2.fragments.PengirimanFragment;
import
id.web.proditipolines.kurir2.services.RefreshFirebaseTokenService;
public class HalamanUtama extends AppCompatActivity
        implements NavigationView.OnNavigationItemSelectedListener {
    private static final int PERMISSIONS REQUEST = 1;
    @Override
    protected void onCreate(Bundle savedInstanceState) {
        super.onCreate(savedInstanceState);
        setContentView(R.layout.activity_halaman_utama); Toolbar
        toolbar = (Toolbar) findViewById(R.id.toolbar);
        setSupportActionBar(toolbar);
        // Check GPS is enabled LocationManager
        lm = (LocationManager)
getSystemService(LOCATION SERVICE);
        if (!lm.isProviderEnabled(LocationManager.GPS PROVIDER)) {
            Toast.makeText(this, "Mohon nyalakan GPS anda terlebih
dahulu", Toast.LENGTH LONG) .show();
            finish();
        //Ambil session login dari shared preference
        SharedPreferences preferences =
getSharedPreferences("SESSION", MODE_PRIVATE);
        if (preferences.getString("TOPIC", "").equals("")) {
            Intent intent = new Intent(getApplicationContext(),
RefreshFirebaseTokenService.class);
            startService(intent);
            Log.d("asdf", "if exec");
int permission = ContextCompat.checkSelfPermission(this,
               Manifest.permission.ACCESS_FINE_LOCATION);
        if (permission != PackageManager.PERMISSION GRANTED)
            { ActivityCompat.requestPermissions(this,
                    new
String[] {Manifest.permission. ACCESS FINE LOCATION},
                    PERMISSIONS_REQUEST);
        DrawerLayout drawer = (DrawerLayout)
```

```
findViewById(R.id.drawer layout);
        ActionBarDrawerToggle toggle = new ActionBarDrawerToggle(
                this, drawer, toolbar, R.string.navigation drawer open,
R.string.navigation_drawer_close);
        drawer.addDrawerListener(toggle);
        toggle.syncState();
        NavigationView navigationView = (NavigationView)
findViewById(R.id.nav view);
        navigationView.setNavigationItemSelectedListener(this);
        displaySelectedScreen(R.id.nav_pengiriman);
        //Ubah TextView header sesuai nama dan email di session login
        View headerView = navigationView.getHeaderView(0); TextView
        headerNama = (TextView)
headerView.findViewById(R.id.header nama);
        TextView headerEmail = (TextView)
headerView.findViewById(R.id.header email);
        headerNama.setText(preferences.getString("NAME", ""));
        headerEmail.setText(preferences.getString("EMAIL", ""));
    @Override
    public void onBackPressed() {
        DrawerLayout drawer = (DrawerLayout)
findViewById(R.id.drawer_layout);
        if (drawer.isDrawerOpen(GravityCompat.START))
            { drawer.closeDrawer(GravityCompat.START);
        } else {
            super.onBackPressed();
        }
    }
    @Override
    public boolean onCreateOptionsMenu(Menu menu) {
        // Inflate the menu; this adds items to the action bar if it is
present.
        getMenuInflater().inflate(R.menu.halaman utama, menu);
        return true;
    }
    @Override
    public boolean onOptionsItemSelected(MenuItem item) {
int id = item.getItemId();
        //noinspection SimplifiableIfStatement
        if (id == R.id.action_settings) {
            return true;
        return super.onOptionsItemSelected(item);
    @SuppressWarnings("StatementWithEmptyBody")
    @Override
   public boolean onNavigationItemSelected(MenuItem item) {
        displaySelectedScreen(item.getItemId());
        return false;
    public void displaySelectedScreen(int itemId) {
        //creating fragment object
        Fragment fragment = null;
```

```
Bundle bundle = new Bundle();
        switch (itemId) {
            case R.id.nav_pengiriman:
                fragment = new PengirimanFragment();
                break;
            case R.id.nav histori:
                fragment = new HistoriFragment();
                break;
            case R.id.nav map:
                fragment = new PengirimanFragment();
                break;
            case R.id.nav_logout:
                //Hapus session login dari shared preference
                SharedPreferences.Editor editor =
getSharedPreferences("SESSION", MODE_PRIVATE).edit();
                editor.clear();
                editor.apply();
                //Intent untuk pindah activity ke LoginActivity
                Intent intent = new Intent(HalamanUtama.this,
LoginActivity.class);
                startActivity(intent);
                finish();
                break:
        }
        if (fragment != null) {
            FragmentTransaction ft =
getSupportFragmentManager().beginTransaction();
            ft.replace (R.id.content frame,
            fragment); ft.commit();
        DrawerLayout drawer = (DrawerLayout)
findViewById(R.id.drawer layout);
        drawer.closeDrawer(GravityCompat.START);
    @Override
    public void onRequestPermissionsResult(int requestCode,
String[] permissions, int[]
            grantResults) {
        if (requestCode != PERMISSIONS_REQUEST && grantResults.length
                && grantResults[0] ==
PackageManager.PERMISSION_GRANTED) { Toast.makeText(this,
            "Mohon aktifkan GPS sebelum
menggunakan aplikasi ini.", Toast. LENGTH LONG).show();
            finish();
```

LoginActivity.Java

```
package id.web.proditipolines.kurir2.activities;

import android.content.Intent;
import android.content.SharedPreferences;
import android.os.Bundle;
import android.support.v7.app.AppCompatActivity;
import android.util.Log; import android.view.View;
import android.widget.TextView;
```

```
import android.widget.Toast;
import id.web.proditipolines.kurir2.R;
import id.web.proditipolines.kurir2.models.Login;
import id.web.proditipolines.kurir2.models.LoginResponse;
import id.web.proditipolines.kurir2.network.APIService;
import id.web.proditipolines.kurir2.network.RetrofitClient;
import retrofit2.Call;
import retrofit2.Callback;
import retrofit2.Response;
public class LoginActivity extends AppCompatActivity {
    TextView email, password;
    @Override
   protected void onCreate(Bundle savedInstanceState)
        { super.onCreate(savedInstanceState);
        setContentView(R.layout.activity login);
        //Cek apakah sudah login, jika sudah maka langsung diarahkan
ke MainActivity
        if (isLogin()){
            //Intent untuk pindah activity ke
            MainActivity Intent intent = new
Intent (LoginActivity.this, HalamanUtama.class);
            startActivity(intent);
            finish();
        }
        //Deklarasi view pada layout
        email=(TextView) findViewById(R.id.txt email);
        password=(TextView) findViewById(R.id.txt password);
    public void HalamanUtama(View view) {
        //Memanggil fungsi login
login(email.getText().toString(),password.getText().toString());
    //Fungsi login dengan parameter inputan user berupa email
dan password
   public void login(String email, String password)
        { //Retrofit call
        APIService service =
RetrofitClient.getClient().create(APIService.class);
        Call<LoginResponse> userCall =
service.userLogin(email, password);
        userCall.enqueue(new Callback<LoginResponse>() {
            //Fungsi ini akan dieksekusi ketika ada respon dari
            server @Override
            public void onResponse(Call<LoginResponse>
call, Response<LoginResponse> response) {
                //Cek apakah kode respon server 401
                if (response.code() == 401) {
                    Toast.makeText(LoginActivity.this, "Login Gagal",
Toast. LENGTH SHORT) . show();
                } else {
                     //Ambil data dari respon server
                    Login login = response.body().getData();
                    //Memanggil fungsi simpan session login
dengan parameter nama dan email
```

```
saveSession(login.getId(), login.getName(),
login.getEmail());
                    //Intent untuk pindah activity ke
                    MainActivity Intent intent = new
Intent (LoginActivity.this, HalamanUtama.class);
                    startActivity(intent);
                    finish();
            //Fungsi ini akan dieksekusi ketika tidak dapat terhubung
ke server
            public void onFailure(Call<LoginResponse> call, Throwable
t) {
                Toast.makeText(LoginActivity.this, "Gagal Terhubung
ke Server", Toast.LENGTH SHORT).show();
                Log.d("asdf", t.toString());
        });
    //Fungsi simpan session login ke shared preference
   public void saveSession(String id, String name, String
        email) { SharedPreferences.Editor editor =
getSharedPreferences ("SESSION",
        MODE PRIVATE) .edit(); editor.putString("ID",
        id ); editor.putString("NAME", name );
        editor.putString("EMAIL", email );
        editor.apply();
public boolean isLogin (){
        SharedPreferences preferences =
getSharedPreferences("SESSION", MODE PRIVATE)
        return !preferences.getString("ID", "").equals("");
    }
}
```

URLConfig.Java

```
package id.web.proditipolines.kurir2.config;

public class URLConfig {

   public static final String BASE_URL =
   "http://antargen.informatikapolines.com/";
}
```

APIService.Java

```
package id.web.proditipolines.kurir2.network;

import id.web.proditipolines.kurir2.models.LoginResponse;
import id.web.proditipolines.kurir2.models.PengirimanResponse;
import retrofit2.Call;
import retrofit2.http.Field;
import retrofit2.http.FormUrlEncoded;
```

```
import retrofit2.http.GET;
import retrofit2.http.POST;
import retrofit2.http.Path;
 * Created by ASUS on 5/24/2018.
public interface APIService {
    @FormUrlEncoded
    @POST("api/auth/login")
    Call<LoginResponse> userLogin(
            @Field("email") String email,
            @Field("password") String password);
    @GET("api/pesanan/kurir/{id}")
    Call<PengirimanResponse> getPengiriman(@Path("id") int id);
    @GET("api/pesanan/kurir/histori/{id}")
    Call<PengirimanResponse> getHistoriPengiriman(@Path("id") int id);
    @FormUrlEncoded
    @POST("api/pesanan/status/{kode}")
    Call<String> updateStatusPesanan(@Path("kode") int kode,
                                     @Field("status") String status);
}
```

RetrofitClient.Java

```
package id.web.proditipolines.kurir2.network;
import android.content.Context;
import com.google.gson.Gson;
import com.google.gson.GsonBuilder;
import id.web.proditipolines.kurir2.config.URLConfig;
import retrofit2.Retrofit;
import retrofit2.converter.gson.GsonConverterFactory;
public class RetrofitClient {
    public static final String BASE URL = URLConfig.BASE URL;
    private static Retrofit retrofit = null;
    // variable to hold context
    private static Context context;
    public static Retrofit getClient() {
        if (retrofit==null) {
            Gson qson = new GsonBuilder()
                    .setLenient()
                    .create();
            retrofit = new Retrofit.Builder()
                    .baseUrl(BASE URL)
.addConverterFactory(GsonConverterFactory.create(gson))
                    .build();
        return retrofit;
```

```
}
```

Activity_halaman_utama.xml

```
<?xml version="1.0" encoding="utf-8"?>
<android.support.v4.widget.DrawerLayout</pre>
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/drawer layout"
    android:layout_width="match_parent"
    android:layout height="match parent"
    android:fitsSystemWindows="true"
    tools:openDrawer="start">
    <include</pre>
        layout="@layout/app bar halaman utama"
        android:layout width="match parent"
        android:layout_height="match_parent" />
    <android.support.design.widget.NavigationView</pre>
        android:id="@+id/nav view"
        android:layout width="wrap content"
        android:layout height="match parent"
        android:layout gravity="start"
        android:fitsSystemWindows="true"
        app:headerLayout="@layout/nav_header_halaman_utama"
        app:menu="@menu/activity_halaman_utama_drawer" />
</android.support.v4.widget.DrawerLayout>
```

Activity_login.xml

```
<?xml version="1.0" encoding="utf-8"?>
<RelativeLayout
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: layout width="match parent"
    android:layout height="match parent"
    android:background="#f7ebd3"
tools:context="id.web.proditipolines.kurir2.activities.LoginActivity">
    <LinearLayout</pre>
        android:layout width="match parent"
        android:layout_height="wrap_content"
        android:layout centerHorizontal="true"
        android:layout_centerVertical="true"
        android:orientation="vertical"
        android:padding="28dp"
        android:background="#f7ebd3" >
        <ImageView</pre>
            android:id="@+id/logo"
            android:layout width="100dp"
            android:layout_height="100dp"
            android:layout gravity="center"
            android:background="@drawable/icon4" />
```

```
<TextView
            android:id="@+id/TextView1"
            android:layout width="wrap content"
            android:layout_height="wrap_content"
            android:layout_gravity="center"
            android:layout marginBottom="5dp"
            android:layout marginTop="5dp"
            android:text=""
            android:textSize="20dip"
            android:textStyle="bold" />
        <com.rengwuxian.materialedittext.MaterialEditTex</pre>
            t android:layout marginLeft="16dp"
            android:id="@+id/txt email"
            android:layout_marginTop="16dp"
            android:layout_marginRight="16dp"
            android:layout width="match parent"
            android:layout height="wrap content"
            app:met floatingLabel="highlight"
            android:hint="Email"/>
        <com.rengwuxian.materialedittext.MaterialEditTex</pre>
            t android:layout marginLeft="16dp"
            android:inputType="textPassword"
            android:id="@+id/txt password"
            android:layout_marginTop="8dp"
            android:layout_marginRight="16dp"
            android:layout width="match parent"
            android:layout_height="wrap_content"
            android:layout marginBottom="16dp"
            app:met_floatingLabel="highlight"
            android:hint="Password"/>
        <Button
            android:id="@+id/btn login"
            android:layout width="wrap content"
            android:layout height="wrap content"
            android:layout_gravity="center"
            android:layout marginBottom="15dp"
            android:text="Login"
            android:onClick="HalamanUtama"
            android:background="#FFFF6600"
            android:textColor="#fff"
            android:textStyle="bold" />
    </LinearLayout>
</RelativeLayout>
```

App_bar_halaman_utama.xml

```
<?xml version="1.0" encoding="utf-8"?>
<android.support.design.widget.CoordinatorLayout
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:layout_width="match_parent"
    android:layout_height="match_parent"

tools:context="id.web.proditipolines.kurir2.activities.HalamanUtama">
    <android.support.design.widget.AppBarLayout
        android:layout_width="match_parent"
        android:layout_width="match_parent"
        android:layout_height="wrap_content"
        android:theme="@style/AppTheme.AppBarOverlay">
```

Dialog_pengiriman.xml

```
<?xml version="1.0" encoding="utf-8"?>
<LinearLayout</pre>
xmlns:android="http://schemas.android.com/apk/res/android"
    android:orientation="vertical" android:layout_width="300dp"
    android:layout_height="match_parent" android:padding="8dp">
    <TextView
        android:layout width="match parent"
        android:layout height="wrap_content"
        android:textColor="@color/black"
        android:textSize="18sp"
        android:text="Detail Pelanggan"/>
    <LinearLayout</pre>
       android:layout_width="match_parent"
        android:layout height="wrap content"
        android:orientation="horizontal">
        <TextView
            android:layout width="0dp"
            android:layout_height="wrap_content"
            android:layout weight="1"
            android:textColor="@color/black"
            android:text="Nama"/>
        <TextView
            android:id="@+id/nama"
            android:layout_width="0dp"
            android:layout height="wrap content"
            android:layout weight="3"
            android:textColor="@color/black"
            android:text=": Sumarno Wibowo"/>
    </LinearLayout>
    <LinearLayout</pre>
       android:layout width="match parent"
        android:layout height="wrap_content"
        android: orientation="horizontal">
        <TextView
            android:layout width="0dp"
            android:layout height="wrap content"
            android:layout_weight="1"
            android:textColor="@color/black"
```

```
android:text="Telepon"/>
    <TextView
        android:id="@+id/telepon"
        android:layout_width="0dp"
        android:layout height="wrap content"
        android:layout_weight="3"
        android:textColor="@color/black"
        android:text=": 08461424647"/>
</LinearLayout>
<LinearLayout</pre>
   android:layout_width="match_parent"
   android:layout_height="wrap_content"
   android:orientation="horizontal">
    <TextView
        android:layout width="0dp"
        android:layout height="wrap content"
        android:layout_weight="1"
        android:textColor="@color/black"
        android:text="Alamat"/>
    <TextView
        android:id="@+id/alamat"
        android:layout width="0dp"
        android:layout_height="wrap_content"
        android:layout_weight="3"
        android:textColor="@color/black"
        android:text=": Jalan Joglo Raya No. 100"/>
</LinearLayout>
<View
   android:layout width="match parent"
   android:layout_height="2dp"
   android:layout marginTop="8dp"
   android:background="@color/grey"/>
<LinearLayout</pre>
   android:layout_width="match_parent"
    android:layout_height="wrap_content"
    android:layout marginTop="8dp"
   android:orientation="horizontal">
    <LinearLayout</pre>
        android:layout_width="0dp"
        android:layout_height="wrap_content"
        android:layout weight="1"
        android:gravity="center_horizontal">
        <Button
            android:id="@+id/message"
            android:layout width="wrap content"
            android:layout height="34dp"
            android:paddingLeft="8dp"
            android:paddingStart="8dp"
            android:paddingEnd="8dp"
            android:paddingRight="8dp"
            android:background="@drawable/rectangle_shape_yellow"
            android:drawableLeft="@drawable/ic message"
            android:drawableStart="@drawable/ic message"
            android:drawablePadding="4dp"
            android:textSize="14sp"
```

```
android:textColor="@color/white"
            android:text="Message"/>
    </LinearLayout>
    <LinearLayout</pre>
        android:layout width="0dp"
        android:layout_height="wrap_content"
        android:layout weight="1"
        android:gravity="center horizontal">
        <Button
            android:id="@+id/call"
            android:layout width="wrap content"
            android:layout height="36dp"
            android:paddingLeft="8dp"
            android:paddingStart="8dp"
            android:paddingEnd="8dp"
            android:paddingRight="8dp"
            android:background="@drawable/rectangle shape green"
            android:drawableLeft="@drawable/ic call"
            android:drawableStart="@drawable/ic call"
            android:drawablePadding="4dp"
            android:textColor="@color/white"
            android:text="Call"/>
    </LinearLayout>
</LinearLayout>
<View
   android:layout_width="match_parent"
   android:layout_height="2dp"
    android:layout marginTop="8dp"
    android:background="@color/grey"/>
<TextView
   android:layout width="match parent"
   android: layout height="wrap content"
   android:layout_marginTop="8dp"
   android:textColor="@color/black"
   android:textSize="18sp"
   android:text="Detail Barang"/>
<RelativeLayout
    android: layout width="match parent"
   android:layout height="wrap content">
    <TextView
        android:id="@+id/produk"
        android:layout width="250dp"
        android:layout height="wrap content"
        android:textColor="@color/black"
        android:ellipsize="end"
        android:textSize="16sp"
        android:text="-Indomie Goreng"/>
    <TextView
        android:id="@+id/jumlah"
        android:layout width="wrap content"
        android:layout_height="wrap_content"
        android:layout_alignParentEnd="true"
        android:layout alignParentRight="true"
        android:textColor="@color/black"
        android:textSize="16sp"
```

```
android:text="Qty: 5"/>
    </RelativeLayout>
    <TextView
        android:id="@+id/tagihan"
        android:layout_width="match_parent"
        android:layout_height="wrap_content"
        android:layout marginTop="8dp"
        android:textColor="@color/black"
        android:textSize="18dp"
         android:textStyle="bold"
        android:text="Total Tagihan: Rp. 50000"/>
    <Button
        android:id="@+id/btn_aksi"
        android:layout_width="match_parent"
android:layout_height="wrap_content"
android:layout_marginTop="16dp"
        android:background="@color/orange"
        android:textColor="@color/white"
        android:textStyle="bold"
        android:textSize="18sp"
        android:text="pickup barang"/>
</LinearLayout>
```