

REPUBLIK INDONESIA KEMENTERIAN HUKUM
DAN HAK ASASI MANUSIA

SURAT PENCATATAN
CIPTAAN

Dalam rangka perlindungan ciptaan di bidang ilmu pengetahuan, seni dan sastra berdasarkan Undang-Undang Nomor 28 Tahun 2014 tentang Hak Cipta, dengan ini menerangkan:

Nomor dan tanggal permohonan : EC00201947455, 24 Juli 2019

Pencipta

Nama : **Susliansyah, Ririn Restu Aria, , dkk**
Alamat : Jalan Margonda Raya, Gang Kemiri Sawah, No.86, Rt 03, Rw 015, Depok, Jawa Barat, 16423
Kewarganegaraan : Indonesia

Pemegang Hak Cipta

Nama : **Susliansyah, Ririn Restu Aria, , dkk**
Alamat : Jalan Margonda Raya, Gang Kemiri Sawah, No.86, Rt 03, Rw 015, Depok, 8, 16423
Kewarganegaraan : Indonesia
Jenis Ciptaan : **Program Komputer**
Judul Ciptaan : **Aplikasi Penjualan Alat Olah Raga Berbasis Website**
Tanggal dan tempat diumumkan untuk pertama kali di wilayah Indonesia atau di luar wilayah Indonesia : 24 Juli 2019, di Depok
Jangka waktu perlindungan : Berlaku selama 50 (lima puluh) tahun sejak Ciptaan tersebut pertama kali dilakukan Pengumuman.
Nomor pencatatan : 000148012

adalah benar berdasarkan keterangan yang diberikan oleh Pemohon.
Surat Pencatatan Hak Cipta atau produk Hak terkait ini sesuai dengan Pasal 72 Undang-Undang Nomor 28 Tahun 2014 tentang Hak Cipta.



a.n. MENTERI HUKUM DAN HAK ASASI MANUSIA
DIREKTUR JENDERAL KEKAYAAN INTELEKTUAL

Dr. Freddy Harris, S.H., LL.M., ACCS.
NIP. 196611181994031001

LAMPIRAN PENCIPTA

No	Nama	Alamat
1	Susliansyah	Jalan Margonda Raya, Gang Kemiri Sawah, No.86, Rt 03, Rw 015
2	Ririn Restu Aria	Jl.Kramat Jaya 6 No.20
3	Nur Hidayati	Griya Waringin Elok B.4/8 RT. 001 RW. 008 Waringin Jaya Bojonggede
4	Indra Riyana Rahadjeng	Jl. Madrasah No. 31A
5	Ratnawaty Marginingsih	Jl Pitara Raya No. 05 Rt 001/016 Pancoran Mas
6	Wiwik Widiyanti	Graha Dinatera No 13 Jl. Keadilan Raya Rangkapan Jaya Baru Pancoran Mas
7	Rosento	Jl. Camar 6 No. 20

LAMPIRAN PEMEGANG

No	Nama	Alamat
1	Susliansyah	Jalan Margonda Raya, Gang Kemiri Sawah, No.86, Rt 03, Rw 015
2	Ririn Restu Aria	Jl.Kramat Jaya 6 No.20
3	Nur Hidayati	Griya Waringin Elok B.4/8 RT. 001 RW. 008 Waringin Jaya Bojonggede
4	Indra Riyana Rahadjeng	Jl. Madrasah No. 31A
5	Ratnawaty Marginingsih	Jl Pitara Raya No. 05 Rt 001/016 Pancoran Mas
6	Wiwik Widiyanti	Graha Dinatera No 13 Jl. Keadilan Raya Rangkapan Jaya Baru Pancoran Mas
7	Rosento	Jl. Camar 6 No. 20

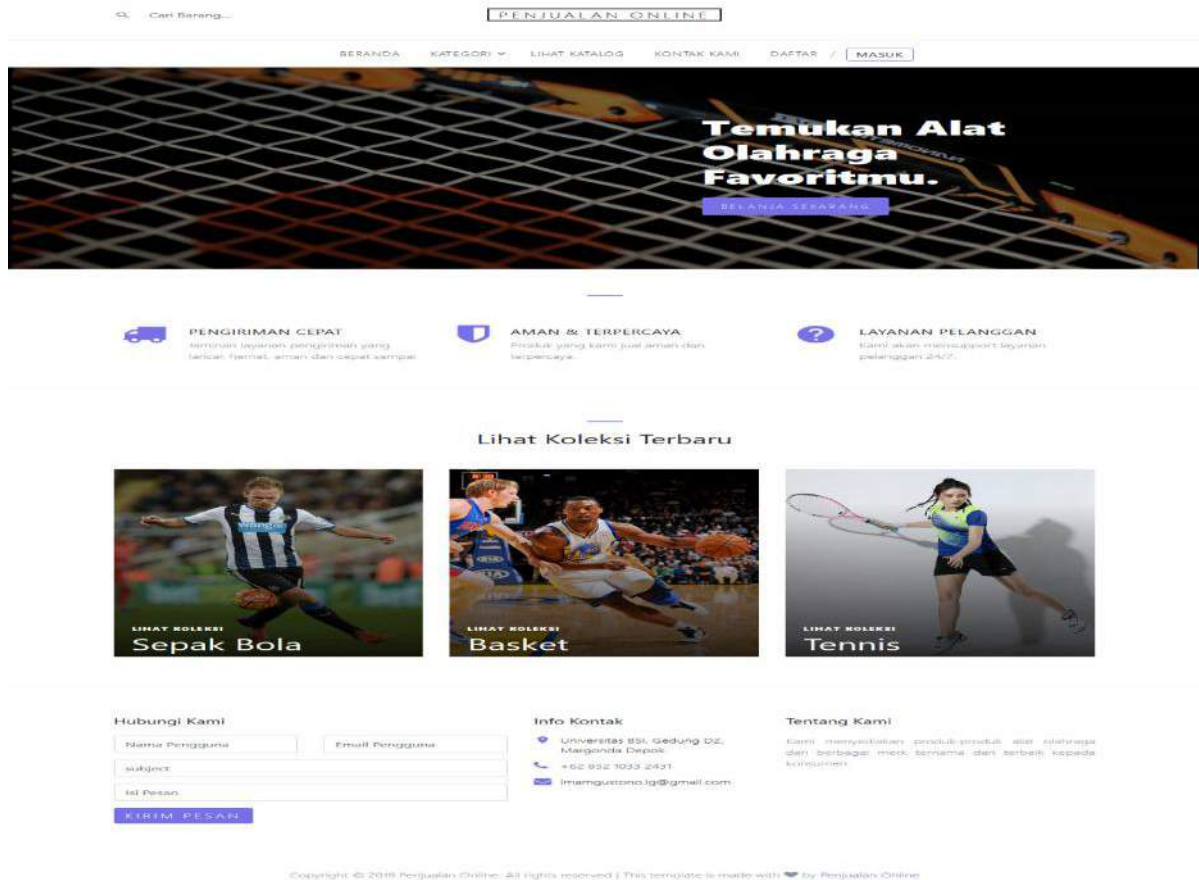


Manual Book

Hak cipta (Program Komputer)

APLIKASI PENJUALAN ALAT OLAHRAGA BERBASIS WEBSITE

APLIKASI PENJUALAN ALAT OLAHRAGA BERBASIS WEBSITE



Tim Penyusun:

Susliansyah
Ririn Restu Aria
Wiwik Widiyanti
Ratnawaty Marginingsih
Rosento
Indra Riyana Rahadjeng
Nur Hidayati

Jakarta

2019

KATA PENGANTAR

Dengan mengucapkan puji syukur kami panjatkan kepada Allah SWT yang mana telah memberikan banyak nikmat serta karunia-Nya kepada kami semua, sehingga kami berhasil menyelesaikan aplikasi program penjualan alat olah raga secara *online*. Kami sadar bahwa makalah yang kami buat masih jauh dari ambang kesempurnaan. Oleh karenanya, semua kritik dan saran yang bersifat membangun dari semua pihak sangat kami harapkan. Akhir kata, kami ucapkan terima kasih kepada seluruh pihak yang telah ikut turut membantu.

DAFTAR ISI

HALAMAN JUDUL.....	i
KATA PENGANTAR	ii
DAFTAR ISI	iii
BAB I PENDAHULUAN	1
BAB II ANALISA KEBUTUHAN	2
BAB III STRUKTUR NAVIGASI.....	3
BAB IV TAHAPAN PENGGUNAAN APLIKASI PROGRAM.....	5
BAB V <i>SOURCE CODE</i>	30

BAB I

PENDAHULUAN

Perkembangan teknologi dibidang informasi mendorong setiap perusahaan ataupun organisasi untuk selalu mengikuti perkembangannya, terutama yang ada hubungannya dengan kegiatan di perusahaan atau organisasi tersebut. Dalam suatu perusahaan terdapat berbagai bidang kegiatan, seperti penjualan, pembelian, persediaan maupun yang lainnya. Dan pada saat ini masih terdapat beberapa perusahaan yang menggunakan sistem secara konvensional dalam menjalankan sistem yang ada di perusahaannya. Hal ini tentunya dapat menimbulkan berbagai permasalahan dalam perusahaan, seperti: jika penyimpanan data tidak tersusun rapi, maka dapat mengakibatkan hilangnya dokumen. Dokumen yang masih ditulis menggunakan tulisan tangan, dapat mengakibatkan kesalahan dalam pembacaan data. Hal seperti ini tentunya akan berpengaruh dalam pembuatan laporannya, sehingga laporan yang dihasilkan menjadi tidak akurat data-datanya dan pembuatan laporannya pun membutuhkan waktu yang lama.

Oleh karena itu, dengan melihat kondisi tersebut, perlu adanya terobosan baru berupa pengembangan sistem, dengan cara pembuatan aplikasi program untuk dapat membantu menyelesaikan permasalahan yang ada di perusahaan tersebut. Salah satu trend sekarang adalah dengan pembuatan E-commerce. *E-Commerce* sendiri saat ini, adalah salah satu alternatif pilihan untuk sebuah perusahaan khususnya yang ada dibidang wiraswasta sebagai media informasi yang dapat memudahkan interaksi antara penjual dan si pengguna. Dengan adanya dukungan dari perkembangan teknologi informasi yang semakin canggih dan mudah untuk diakses, perkembangan *E-Commerce* pun semakin banyak diminati oleh kalangan orang.

Penggunaan *E-commerce* yang didukung dengan penggunaan aplikasi program, khususnya penjualan akan sangat membantu perusahaan dalam mengelola datanya. Berbagai keuntunganpun dapat diperoleh dengan adanya penggunaan aplikasi program penjualan, terutama untuk produk Olahraga, seperti kemudahan dalam melakukan transaksi mulai dari pemesanan produk olahraga sampai dengan pembayarannya.

BAB II

ANALISA KEBUTUHAN

A. Analisa Kebutuhan Fungsional

Analisis kebutuhan fungsional aplikasi *web profile* ditentukan berdasarkan spesifikasi kebutuhan interaksi pada aplikasi, yaitu interaksi antar sesama pengunjung, dan pengunjung-pengelola.

Tabel III. 1 Analisa Kebutuhan Fungsional

No.	Jenis Transaksi	Implementasi pada aplikasi web profile
1.	Pembeli dan Penjual	Komunikasi melalui kirim komentar dan email agar penjual mendapatkan <i>feedback</i> dari pembeli.

B. Analisa Pengguna

Analisis pengguna mencirikan siapa saja pengguna dari perangkat lunak yang dispesifikasikan dan apa saja haknya terhadap perangkat lunak tersebut. Aplikasi web profile yang dibangun memiliki dua jenis pengguna, yaitu :

1. Pembeli

Pada aplikasi *web profile* ini pengunjung merupakan orang yang melihat atau menggunakan aplikasi ini. Pengunjung sendiri dapat melakukan login ataupun mendaftar untuk menjadi member. Jadi pengunjung bisa melihat dan nantinya akan mendapatkan informasi dari web profile tersebut.

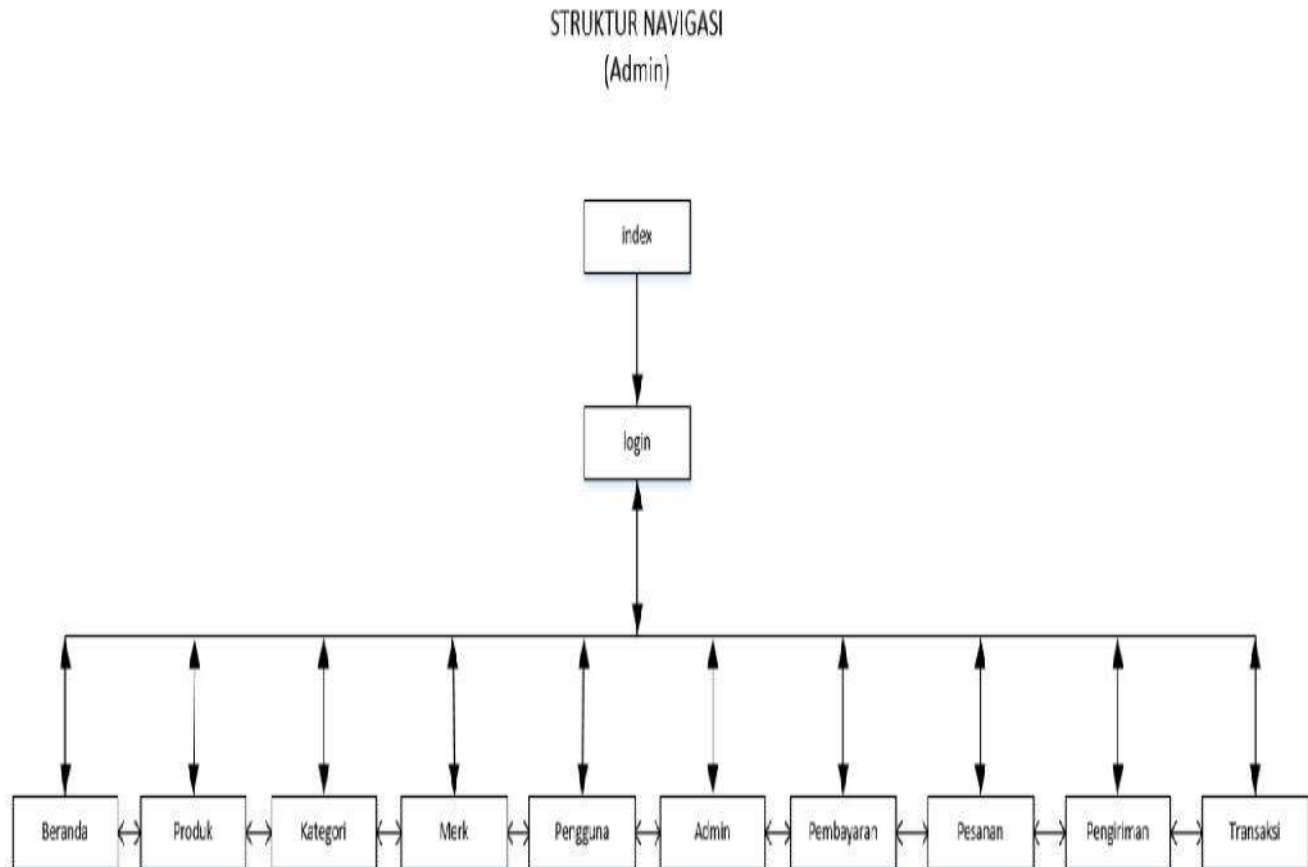
2. Penjualan

Penjualan merupakan orang yang bertanggung jawab mengelola data secara statis. Adapun syarat yang dibutuhkan untuk menjadi pengelola antara lain :

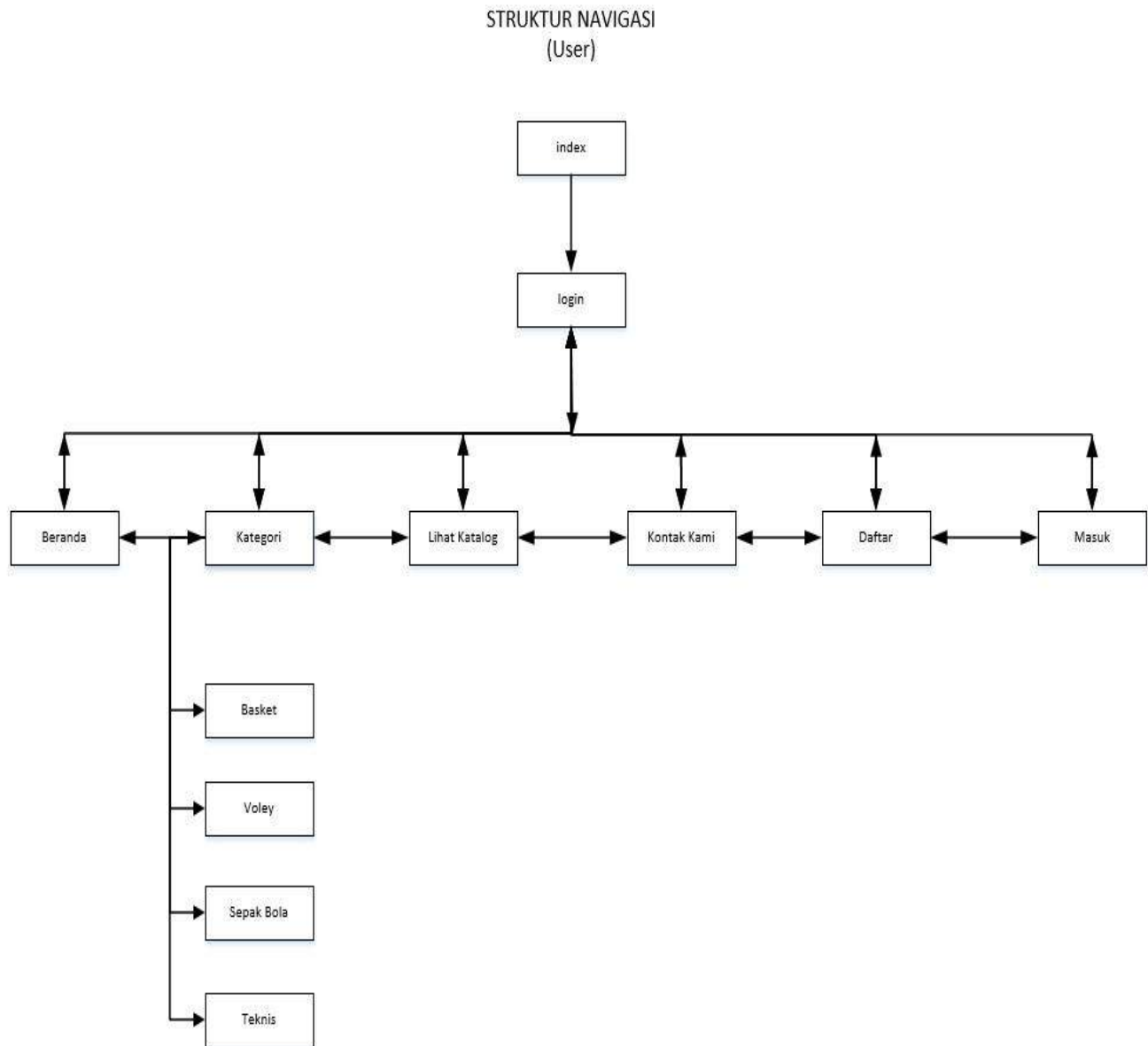
- a. Mampu menguasai teknik pemrograman PHP dan MySQL.
- b. Menguasa ilmu komputer.
- c. Mengerti dasar-dasar internet.

BAB III

STRUKTUR NAVIGASI



Gambar III. 1 Struktur Navigasi Admin



Gambar III. 2 Struktur Navigasi Pengguna

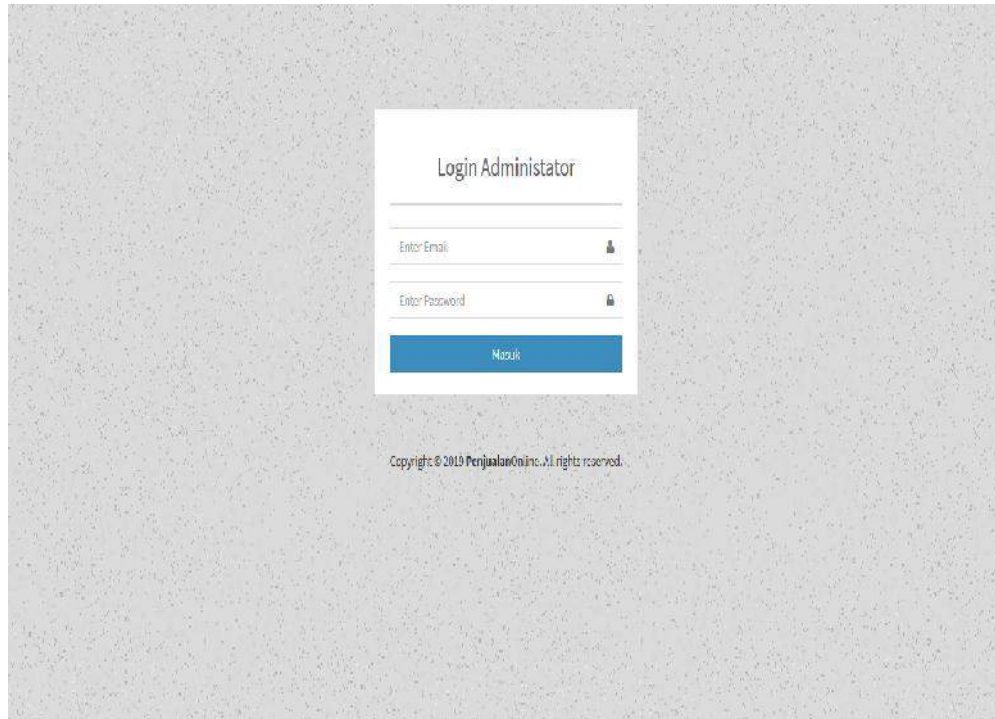
BAB IV

TAHAPAN PENGGUNAAN APLIKASI PROGRAM

A. Tahapan Dalam Penggunaan Halaman Admin

1. Halaman Login_Admin

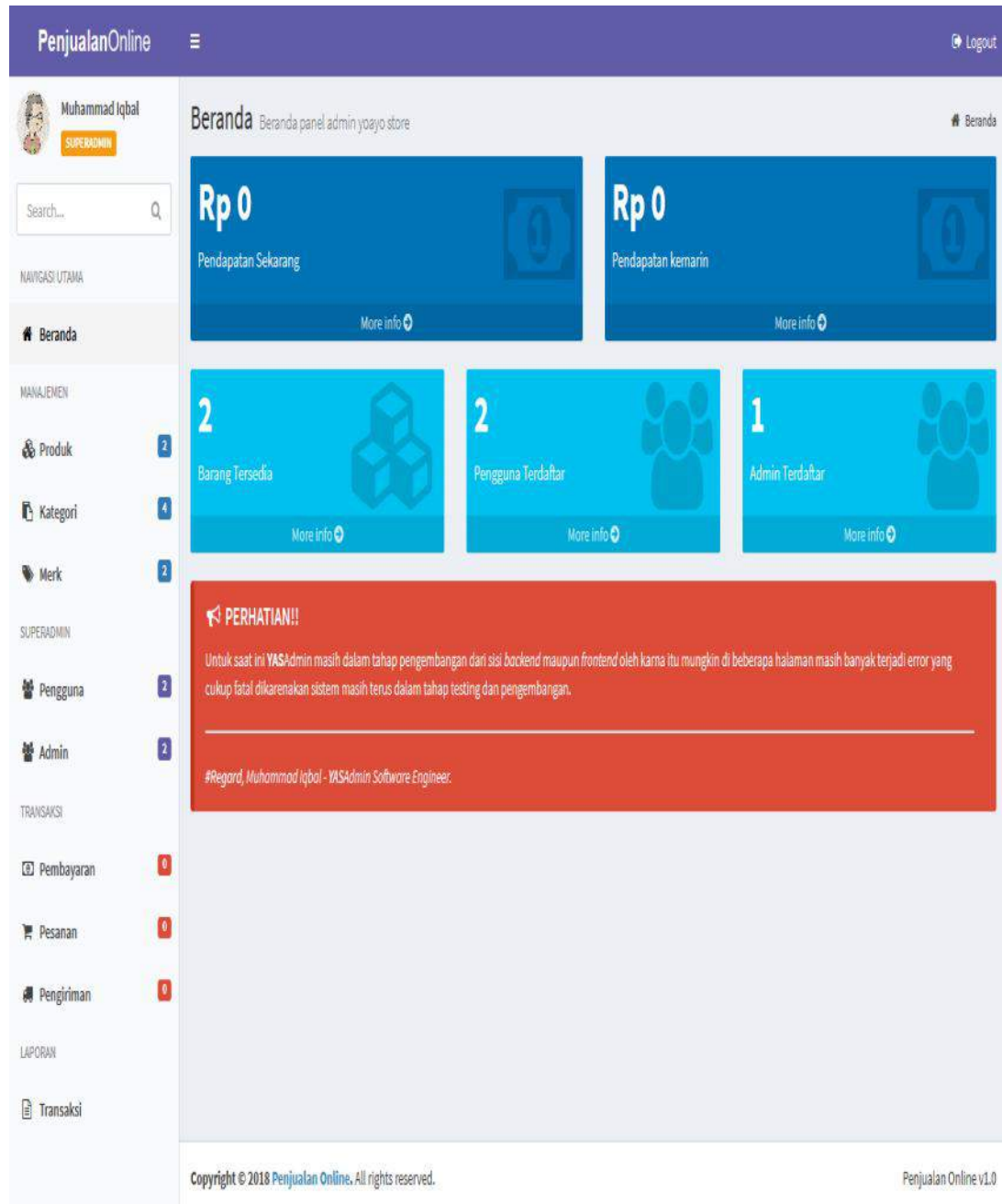
Admin harus memasukan user dan password untuk mengakses halaman login administrator.



Gambar III. 3 Halaman Login_Admin

2. Halaman Antar Muka Beranda_Admin

Admin dapat menambah, mengubah, dan menghapus produk.



Gambar III. 4 Halaman Antar Muka Beranda Admin

3. Halaman Antar Muka Data Prodak Admin

Admin dapat menambahkan prodak dengan cara memasukkan nama prodak, kategori, merk produk dan lain-lain.

The screenshot displays the 'Manajemen Produk' (Product Management) interface. The sidebar on the left contains navigation links: NAVIGASI UTAMA, Beranda, MANAJEMEN, Produk (2), Kategori (4), Merk (2), SUPERADMIN, Pengguna (2), Admin (2), TRANSAKSI, Pembayaran (9), Pesanan (9), Pengiriman (9), LAPORAN, and Transaksi. The top header shows the user 'Muhammad Iqbal' with a 'SUPERADMIN' role and a 'Logout' button. The main content area is titled 'Manajemen Produk' and includes a 'Form Input Produk' section. Below this is a 'Table Produk' with the following data:

ID Produk	Nama Produk	Kategori	Merk	Status	Tanggal Masuk	Aksi
BRG1810131	Wilson Highlight Original	Basket	Wilson	✓ Tersedia	2018-10-13 01:53:42	Action ▼
BRG1810242	Mikasa MG MV 2200 Gold	Volley	Mikasa	✓ Tersedia	2018-10-24 14:22:52	Action ▼

Below the table, it indicates 'Showing 1 to 2 of 2 entries' and provides pagination controls: Previous, 1, Next. The footer contains the copyright notice 'Copyright © 2018 Penjualan Online. All rights reserved.' and the version 'Penjualan Online v1.0'.

Gambar III. 5 Halaman Antar Muka Data Produk Admin

4. Halaman Antar Muka Kategori Admin

Disini Admin masukkan nama kategori prodak agar dapat menyimpan nama kategori prodak dan mengecek atau mencari nama kategori prodak.

Manajemen Kategori Halaman manajemen kategori pada prooduk

Form Input Kategori Produk

Nama Kategori

Silahkan masukan kategori produk

Check Nama kategori

Simpan kategori

Table Kategori

ID Kategori	Nama Kategori	Aksi
KTG1810131	Basket	Action
KTG1810172	Volley	Action
KTG1810173	Sepak Bola	Action
KTG1810214	Tennis	Action

Showing 1 to 4 of 4 entries

Previous 1 Next

Copyright © 2018 Penjualan Online. All rights reserved. Penjualan Online v1.0

Gambar III. 6 Halaman Antar Muka Kategori Admin

5. Halaman Antar Muka Merk Admin

Halaman yang digunakan untuk memasukan merk-merk produk dengan cara menginputkan nama merk yang ada pada form Input Merk Produk.

The screenshot displays the 'Manajemen Merk' (Brand Management) page in the 'PenjualanOnline' system. The interface includes a sidebar with navigation links, a main content area with a 'Form Input Merk Produk' and a 'Table Merk'.

Form Input Merk Produk

Nama Merk

Silahkan masukan merk produk

Check Nama Merk

Simpan Merk

Table Merk

ID Merk	Nama Merk	Aksi
MRK1810131	Wilson	Action
MRK1810242	Mikasa	Action

Showing 1 to 2 of 2 entries

Previous 1 Next

Copyright © 2018 Penjualan Online. All rights reserved. Penjualan Online v1.0

Gambar III. 7 Halaman Antar Muka Merk Admin

6. Halaman Antar Muka Pengguna Admin

Halaman yang digunakan untuk melihat pengguna yang sudah melakukan pembelian.

PenjualanOnline ≡ Logout

Muhammad Iqbal SUPERADMIN

Manajemen Akun Pengguna Halaman manajemen akun pengguna

Beranda > _superadmin > pengguna

Search...

NAVIGASI UTAMA

- Beranda

MANAJEMEN

- Produk 2
- Kategori 4
- Merk 2

SUPERADMIN

- Pengguna 2**
- Admin 2

TRANSAKSI

- Pembayaran 8
- Pesanan 8
- Pengiriman 8

LAPORAN

- Transaksi

Table Akun Pengguna

Search:

ID Pengguna	Nama Pengguna	Email	No Telepon	Aksi
PGN1809201	Muhammad Iqbal	miqbal.pengguna@email.com	082298277709	Action
PGN1811122	Dimas Pamungkas	dimas.pengguna@email.com	082298277709	Action

Showing 1 to 2 of 2 entries

Previous 1 Next

Copyright © 2018 **Penjualan Online**. All rights reserved. Penjualan Online v1.0

Gambar III. 8 Halaman Antar Muka Pengguna Admin

7. Halaman Antar Muka Akun Admin

Halaman yang digunakan untuk melihat dan mengelolah akun admin yang bisa akses masuk.

PenjualanOnline ≡ Logout

Manajemen Akun Admin Halaman manajemen akun admin

Beranda > Superadmin > Admin

Search...

NAVIGASI UTAMA

- Beranda

MANAJEMEN

- Produk 2
- Kategori 4
- Merk 2

SUPERADMIN

- Pengguna 2
- Admin 2**

TRANSAKSI

- Pembayaran 6
- Pesanan 6
- Pengiriman 6

LAPORAN

- Transaksi

Table Akun Admin + Buat Akun Admin

Search:

ID Admin	Nama Admin	SuperAdmin	Status Blokir	Aksi
ADM1809251	Muhammad Iqbal	✓ SuperAdmin	✗ Tidak	Action ▼
ADM1810162	Dimas Wahyu Pamungkas	✗ SuperAdmin	✓ Ya	Action ▼

Showing 1 to 2 of 2 entries

Previous 1 Next

Copyright © 2018 [Penjualan Online](#). All rights reserved. Penjualan Online v1.0

Gambar III. 9 Halaman Antar Muka Akun Admin

8. Halaman Antar Muka Tambah Admin

Halaman yang digunakan untuk melakukan penambahan akun admin.

The screenshot displays the 'Tambah Admin' modal form. The form is titled 'Tambah Admin' and has a close button (X) in the top right corner. It contains the following fields and labels:

- Nama Lengkap Admin**: A text input field with a placeholder 'Masukan name tanpa karakter khusus dan angka'.
- Email Aktif**: A text input field with a placeholder 'Silahkan masukan email aktif'.
- Foto Admin**: A file selection button labeled 'Telusuri...' with a placeholder 'Silahkan pilih foto admin'.
- Password Sementara**: A text input field with a placeholder 'Silahkan masukan password admin tanpa karakter khusus'.

At the bottom of the form, there are two buttons: 'Tutup' (Close) and 'Simpan Data Admin' (Save Admin Data).

The background shows the 'PenjualanOnline' dashboard. The sidebar on the left contains a search bar and a list of navigation links: 'NAVIGASI UTAMA', 'Beranda', 'MANAJEMEN', 'Produk', 'Kategori', 'Merk', 'SUPERADMIN', 'Pengguna', 'Admin', 'TRANSAKSI', 'Pembayaran', 'Pesanan', 'Pengiriman', 'LAPORAN', and 'Transaksi'. The main content area shows a table of admin accounts with columns for 'ID Admin', 'Status Blokir', and 'Aksi'. The table lists two admin accounts: 'ADM1809251' and 'ADM1810162'. The footer of the page contains the copyright notice 'Copyright © 2018 Penjualan Online. All rights reserved.' and the version 'Penjualan Online v1.0'.

Gambar III. 10 Halaman Antar Muka Tambah Admin

9. Halaman Antar Muka Pesanan Admin

Halaman yang digunakan untuk mengatur pesanan yang masuk.

PenjualanOnline ≡ Logout

Manajemen Pesanan Halaman manajemen segala pesanan Beranda Pesanan

Muhammad Iqbal
SUPERADMIN

Search...

NAVIGASI UTAMA

Beranda

MANAJEMEN

Produk 2

Kategori 4

Merk 2

SUPERADMIN

Pengguna 2

Admin 2

TRANSAKSI

Pembayaran 0

Pesanan 0

Pengiriman 0

LAPORAN

Transaksi

Table Pesanan Yang Sedang Di Proses

Search:

ID Pesanan	Nama Penerima	No Telepon	Status Pesanan	Aksi
No data available in table				

Showing 0 to 0 of 0 entries Previous Next

Table Pesanan Yang Siap Di Kirim

Search:

ID Pesanan	Nama Penerima	No Telepon	Layanan	Status Pesanan	Aksi
No data available in table					

Showing 0 to 0 of 0 entries Previous Next

Copyright © 2018 [Penjualan Online](#). All rights reserved. Penjualan Online v1.0

Gambar III. 11 Halaman Antar Muka Pesanan Admin

10. Halaman Antar Muka Data Pembayaran Admin

Halaman ini melakukan verifikasi pembayaran jika valid maka pembayaran di verifikasi dan pesanan akan langsung di proses.

The screenshot displays the 'Manajemen Pembayaran' (Payment Management) section of the 'PenjualanOnline' application. The interface is divided into a sidebar, a top header, and a main content area.

Top Header: 'PenjualanOnline' logo, a menu icon, and a 'Logout' button.

Sidebar: Contains navigation links for 'Muhammad Iqbal' (SUPERADMIN), 'Beranda', 'Produk' (2), 'Kategori' (4), 'Merk' (2), 'Pengguna' (2), 'Admin' (2), 'Pembayaran' (9), 'Pesanan' (9), 'Pengiriman' (9), and 'Transaksi'.

Main Content Area: Titled 'Manajemen Pembayaran' with the subtitle 'Halaman manajemen segala pembayaran pesanan'. It features two tables:

- Table Pembayaran Yang Belum Di Verifikasi:** A table with columns: ID Pesanan, Foto Bukti, Status Pembayaran, Tanggal Upload, Batas Waktu, and Aksi. It shows 'No data available in table' and 'Showing 0 to 0 of 0 entries'.
- Table Pembayaran Yang Sudah Di Verifikasi:** A table with the same columns as the first table. It also shows 'No data available in table' and 'Showing 0 to 0 of 0 entries'.

Both tables include a search bar and pagination controls (Previous, Next).

Footer: Copyright © 2018 Penjualan Online. All rights reserved. Penjualan Online v1.0

Gambar III. 12 Halaman Antar Muka Pembayaran Admin

11. Halaman Antar Muka Pengiriman Admin

Halaman tabel ini menampilkan di proses dan siap untuk dikirim.

PenjualanOnline Logout

Manajemen Pengiriman Halaman manajemen segala pengiriman pesanan Beranda > pengiriman

Search:

Table Data Pesanan Terkirim

ID Pesanan	Nama Penerima	No. Telepon	Layanan	No. Resi	Status Pesanan	Tanggal Di Kirim
No data available in table						

Showing 0 to 0 of 0 entries Previous Next

Table Data Pesanan Selesai

ID Pesanan	Nama Penerima	No. Telepon	Layanan	No. Resi	Status Pesanan	Tanggal Di Terima	Aksi
No data available in table							

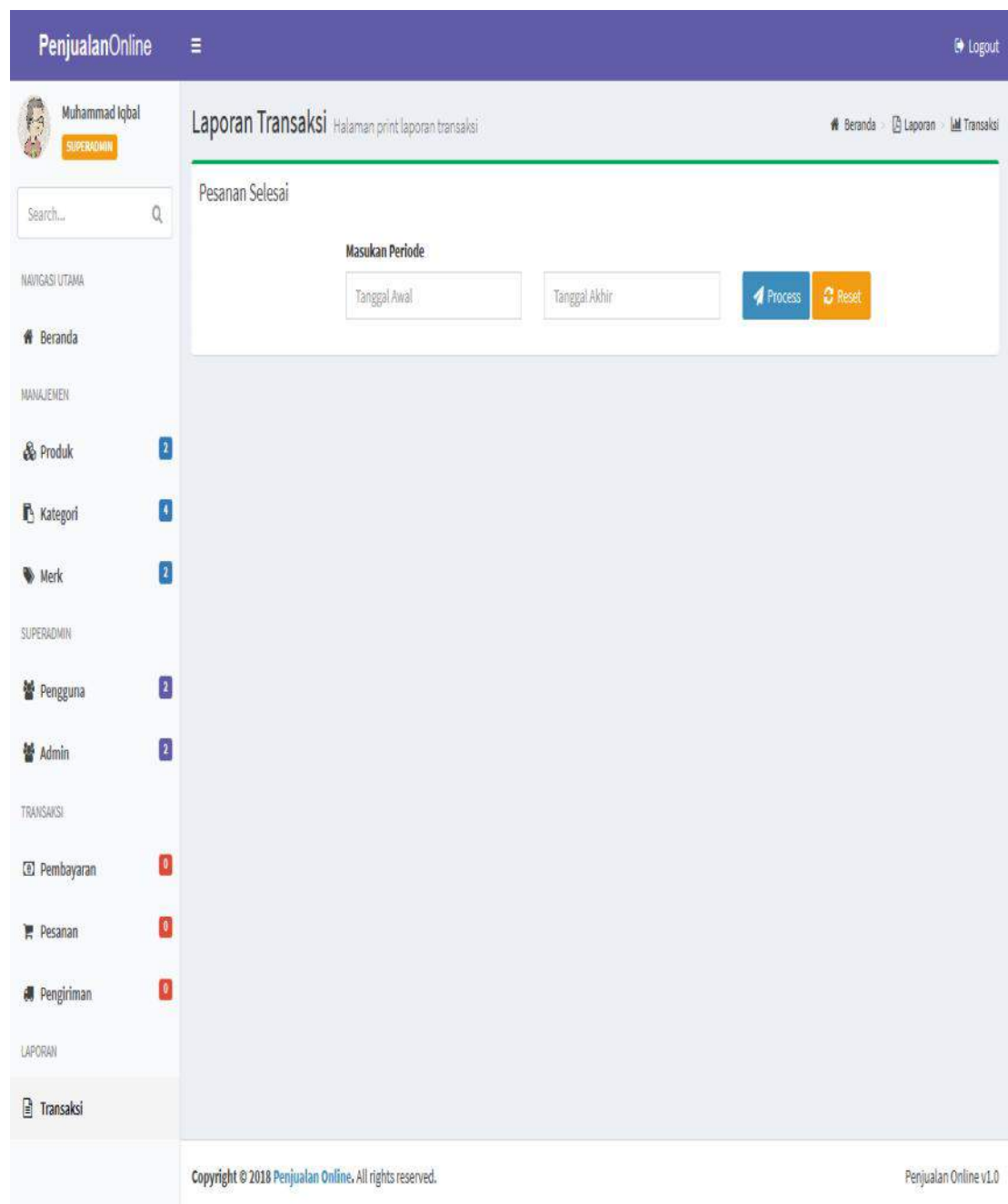
Showing 0 to 0 of 0 entries Previous Next

Copyright © 2018 **Penjualan Online**. All rights reserved. Penjualan Online v1.0

Gambar III. 13 Halaman Antar Muka Pengiriman Admin

12. Halaman Antar Muka Laporan Transaksi Admin

Halaman yang digunakan untuk melihat laporan transaksi selama perhari, perminggu, perbulan dan pertahun.



Gambar III. 14 Halaman Antar Muka Laporan Transaksi Admin

B. Tahapan Dalam Penggunaan Halaman Pengguna

1. Halaman Antar Muka Register_Pengguna

Halaman ini digunakan untuk melakukan registrasi data diri pembeli sebelum melakukan login jika pembeli tersebut belum memiliki akun.

The screenshot displays the 'Pendaftaran Akun Baru' (New Account Registration) page. At the top, there is a search bar labeled 'Cari Barang...' and a navigation menu with links: BERANDA, KATEGORI, LIHAT KATALOG, KONTAK KAMI, DAFTAR, and MASUK. Below the navigation bar, a breadcrumb trail shows 'Beranda / Daftar'. The main content area is titled 'Pendaftaran Akun Baru' and contains a registration form with the following fields:

- Nama Lengkap (Full Name)
- Password
- Jenis Kelamin (Gender) with a dropdown menu showing 'Pilih Jenis Kelamin..'
- Ulangi Password (Repeat Password)
- Email

A blue button labeled 'DAFTAR SEKARANG' (Register Now) is positioned below the form. Below the button, a link reads 'Sudah Punya Akun ? Silahkan Login' (Already have an account? Please login). The footer section includes three columns: 'Hubungi Kami' (Contact Us) with fields for 'Nama Pengguna' (Username), 'Email Pengguna' (User Email), 'subject', and 'Isi Pesan' (Message), followed by a 'KIRIM PESAN' (Send Message) button; 'Info Kontak' (Contact Info) with details for Universitas BSI, Gedung D2, Margonda Depok, phone number +62 852 1033 2431, and email imangustono.ig@gmail.com; and 'Tentang Kami' (About Us) with a brief description of the products offered.

Copyright © 2019 Penjualan Online. All rights reserved | This template is made with ♥ by Penjualan Online

Gambar III. 15 Halaman Antar Muka Registrasi Pengguna

2. Halaman Antar Muka Login Pengguna

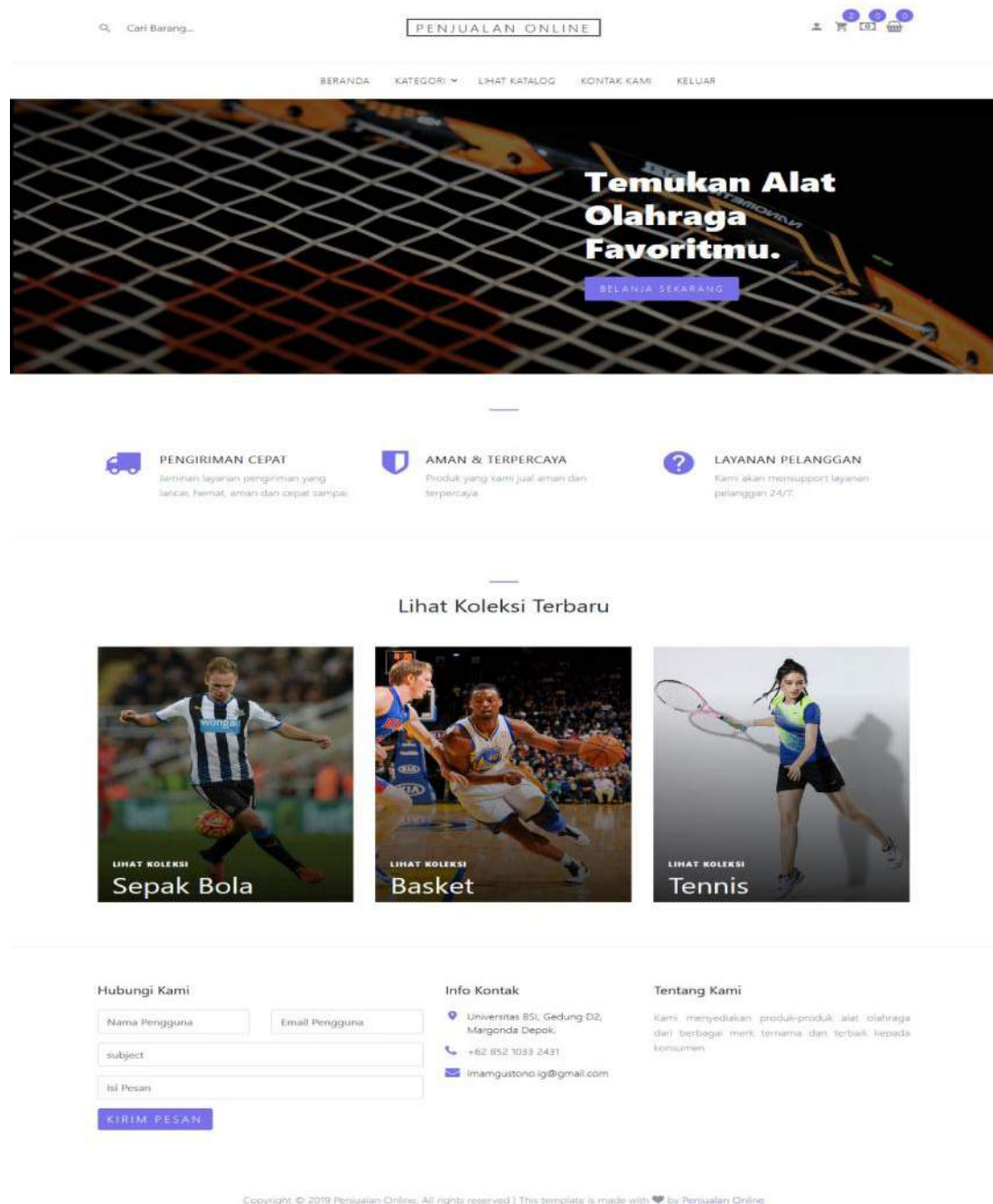
Pengguna harus memasukan Email dan password untuk mengakses halaman utama.

The screenshot displays the 'Masuk' (Login) page of an online store. At the top, there is a search bar labeled 'Cari Barang...' and a navigation bar with links: 'BERANDA', 'KATEGORI', 'LIHAT KATALOG', 'KONTAK KAMI', 'DAFTAR', and 'MASUK'. Below the navigation bar, a breadcrumb trail shows 'Beranda / Masuk'. The main content area is titled 'Masuk' and contains a login form with two input fields: 'Email' and 'Password'. A blue button labeled 'MASUK' is positioned below the fields. A link 'Lupa Password ? Klik Di Sini' is located below the button. The footer section includes three columns: 'Hubungi Kami' with input fields for 'Nama Pengguna', 'Email Pengguna', 'subject', and 'Isi Pesan', and a 'KIRIM PESAN' button; 'Info Kontak' with location, phone, and email details; and 'Tentang Kami' with a brief description. The footer also contains a copyright notice: 'Copyright © 2019 Penjualan Online. All rights reserved | This template is made with ♥ by Penjualan Online'.

Gambar III. 16 Halaman Antar Muka Login Pengguna

3. Halaman Antar Muka Beranda Pengguna

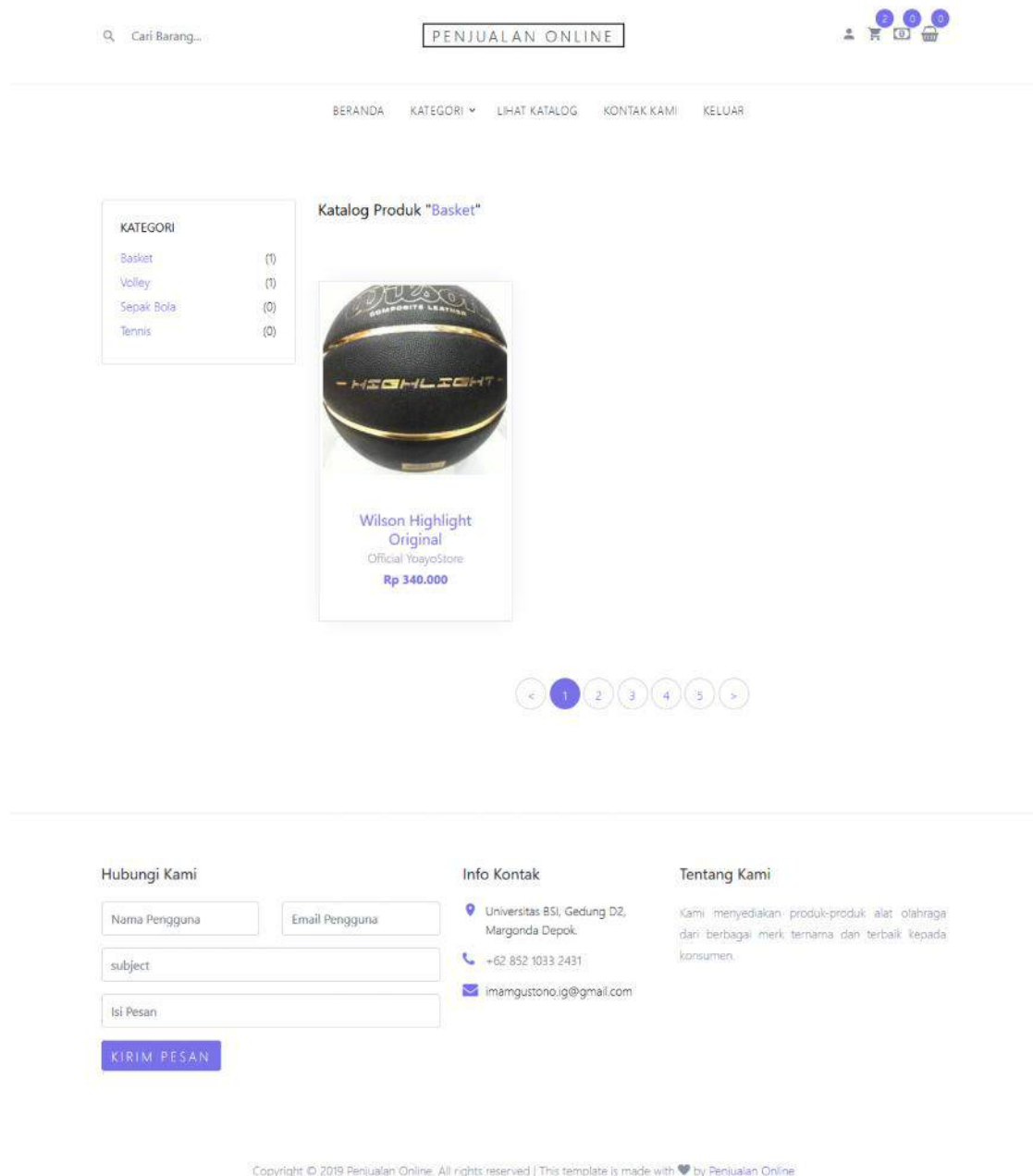
Halaman beranda ini digunakan sebagai tampilan awal yang pembeli lihat ketika mengakses website.



Gambar III. 17 Halaman Antar Muka Beranda Pengguna

4. Halaman Antar Muka Kategori Pengguna

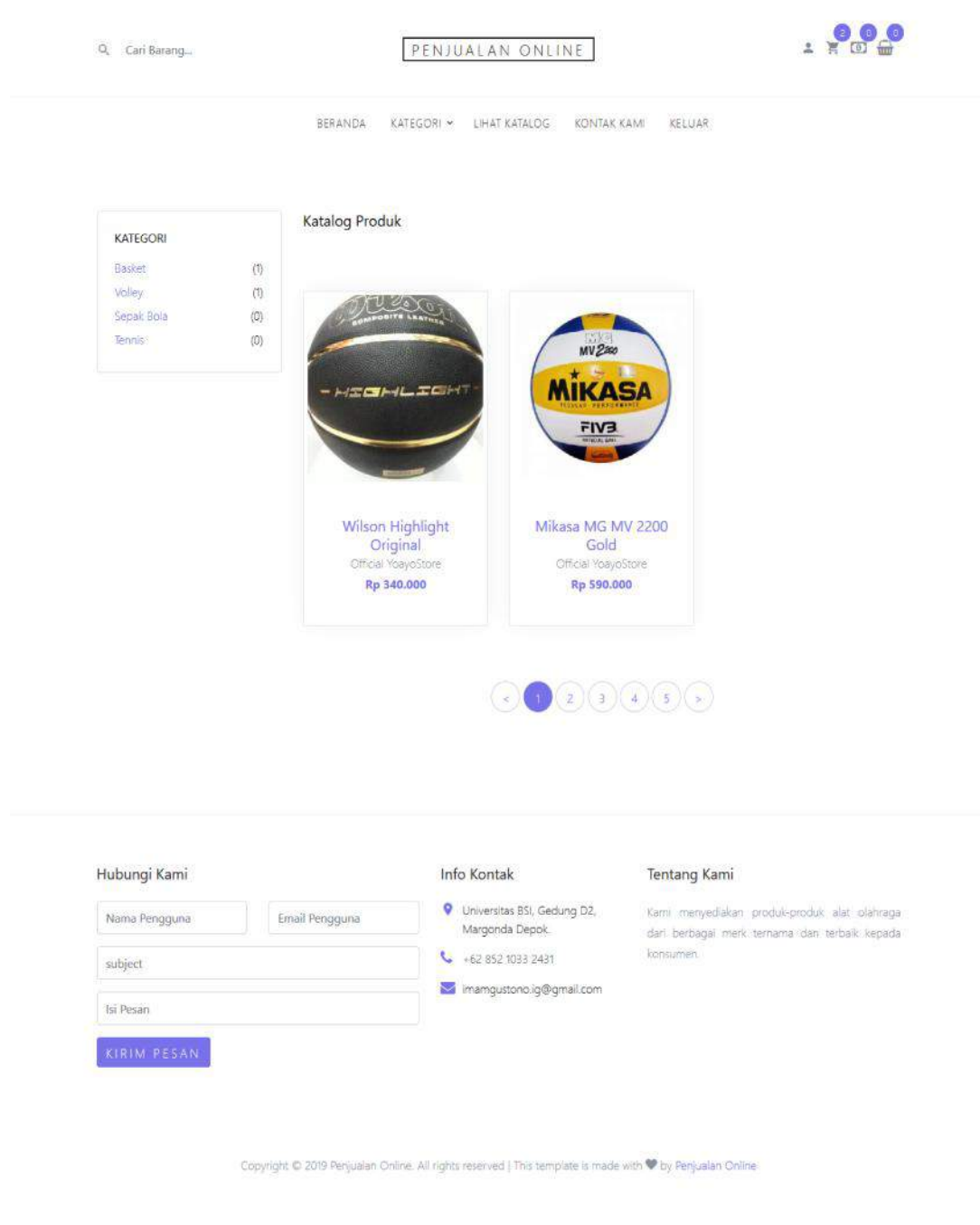
Halaman ini berfungsi untuk melihat kategori produk – produk apa saja yang dijual.



Gambar III. 18 Halaman Antar Muka Kategori Pengguna

5. Halaman Antar Muka Lihat Katalog Pengguna

Halaman ini berfungsi sebagai katalog dari produk-produk apa saja yang dijual.



Gambar III. 19 Halaman Antar Muka Lihat Katalog Pengguna

6. Halaman Antar Muka Detail Produk_Pengguna

Halaman ini digunakan pembeli untuk melihat detail dari produk yang akan di beli

Search Cari Barang...

PENJUALAN ONLINE

BERANDA KATEGORI LIHAT KATALOG KONTAK KAMI KELUAR

Beranda / Produk / Detail

Wilson Highlight Original

WILSON HIGHLIGHT ORIGINAL
 Warna: Black/Gold
 Harga Retail: IDR 399.000
 Harga kami: 340.000 -Hemat 59.000
 Size: 7
 Art #WTB068523
 100% Original
 Made in China
 Description: Composite Leather

Berat	500gram
Stok	93pcs
Status	Tersedia
Harga	Rp 340.000

1

TAMBAH KE KERANJANG

Hubungi Kami

Nama Pengguna
 Email Pengguna
 subject
 Isi Pesan
 KIRIM PESAN

Info Kontak

Universitas BSI, Gedung D2,
 Margonda Depok
 +62 852 1033 2431
 imamgustono.ig@gmail.com

Tentang Kami

Kami menyediakan produk-produk alat olahraga dari berbagai merk ternama dan terbaik kepada konsumen.

Copyright © 2019 Penjualan Online. All rights reserved | This template is made with ♥ by Penjualan Online

Gambar III. 20 Halaman Antar Muka Detail Produk_Pengguna

7. Halaman Antar Muka Keranjang Pengguna

Halaman ini digunakan pembeli untuk melihat produk apa saja yang akan di beli dan berapa jumlahnya sebelum masuk ke proses checkout

Search: Cari Barang... PENJUALAN ONLINE

BERANDA KATEGORI LIHAT KATALOG KONTAK KAMI KELUAR

Beranda / Keranjang

Gambar	Nama Produk	Harga	Jumlah	Subtotal	Hapus
	Wilson Highlight Original Berat Satuan : 500gram	Rp 340.000	- 1 + UPDATE	Rp 340.000	

LANJUTKAN BELANJA

TOTAL KERANJANG

Total Biaya Rp 340.000
Total biaya di atas belum termasuk ongkos kirim.

Pilih Alamat
 Pilih Alamat...

PROSES CHECKOUT

Hubungi Kami

Nama Pengguna

Email Pengguna

subject

Isi Pesan

KIRIM PESAN

Info Kontak

Universitas BSL Gedung D2,
Margonda Depok.

+62 852 1033 2431

imamgustono.ig@gmail.com

Tentang Kami

Kami menyediakan produk-produk alat olahraga dari berbagai merk ternama dan terbaik kepada konsumen.

Copyright © 2019 Penjualan Online. All rights reserved | This template is made with ♥ by Penjualan Online.

Gambar III. 21 Halaman Antar Muka Keranjang Pengguna

8. Halaman Antar Muka *Checkout* Pengguna

Halaman ini digunakan pembeli untuk memasukan data terkait alamat pengiriman, jenis kurir, dan bank apa yang akan dipakai untuk melakukan pembayaran.

Search: Cari Barang...

PENJUALAN ONLINE

BERANDA KATEGORI LIHAT KATALOG KONTAK KAMI KELUAR

Beranda / Keranjang / Checkout

KEMBALI

Detail Pengiriman

Informasi Penerima

Nama Lengkap *

Muhammad Iqbal

No Telepon *

082298277709

Alamat *

Lorem ipsum dolor sit amet, consectetur adipiscing elit, sed do eiusmod tempor.

Pilih Tujuan Pengiriman

Provinsi Kota

Service

Informasi Transfer Bank

Nama Bank Atas Nama No. Rekening

Pilih Bank

Catatan Pengiriman (Optional)

Detail Tagihan

Product	Total
Wilson Highlight Original x 1	Rp 340.000
Berat : 500gram - Total Berat : 500gram	
Subtotal Berat	500gram
Subtotal Biaya	Rp 340.000
Ongkos Kirim	

Transfer Bank

Silahkan Transfer Ke Rekening Di Bawah :

mandiri 12345678910 a/n nanda nugraha

Untuk saat ini kami hanya menggunakan rekening yang tertera di atas, jika anda transfer pembayaran selain menggunakan rekening di atas kami tidak bertanggung jawab.

PROSES PESANAN

Hubungi Kami

Nama Pengguna Email Pengguna

subject

Isi Pesan

KIRIM PESAN

Info Kontak

Universitas BSI, Gedung D2, Mangonda Depok

+62 852 1033 2431

imamgustono.ig@gmail.com

Tentang Kami

Kami menyediakan produk-produk alat olahraga dan berbagai merk ternama dan terbaik kepada konsumen.

Copyright © 2019 Penjualan Online. All rights reserved | This template is made with by Penjualan Online

Gambar III. 22 Halaman Antar Muka Checkout Pengguna

9. Halaman Antar Muka Daftar Pesanan Pengguna

Halaman ini digunakan untuk melihat daftar pesanan apa saja yang telah dilakukan oleh pengguna.

Cari Barang...

0

0

1

PENJUALAN ONLINE

BERANDA

KATEGORI

LIHAT KATALOG

KONTAK KAMI

KELUAR

Beranda / Pesanan

SUCCESS!! Pesanan Berhasil Di Simpan

Daftar Pesanan

RIWAYAT PESANAN

INFO PEMBAYARAN!

Silahkan Transfer Ke Rekening Di Bawah :

mandiri

12345678910 a/n nanda nurjanah

Untuk saat ini kami hanya menggunakan rekening yang tertera di atas,
jika anda transfer pembayaran selain rekening di atas kami tidak bertanggung jawab.

Note : Setelah transfer pembayaran pada rekening di atas di harapkan untuk segera mengupload bukti pembayaran.

No	Kode Pesanan	No. Invoice	Total Pembayaran	Status	Batalkan
#1	#PSN1907182 Lihat Detail Pesanan	UPLOAD BUKTI Upload bukti pembayaran.	Rp 415.000	Belum Di Proses	BATALKAN
Batas Waktu Pembayaran : 2019-07-19					

NOTE: Jika ingin mencabut status pembatalan silahkan hubungi kontak di bawah.

Hubungi Kami

Nama Pengguna

Email Pengguna

subject

Isi Pesan

KIRIM PESAN

Info Kontak

Universitas BSI, Gedung D2,
Margonda Depok.

+62 852 1033 2431

imamgustono.ig@gmail.com

Tentang Kami

Kami menyediakan produk-produk alat olahraga dari berbagai merk ternama dan terbaik kepada konsumen.

Copyright © 2019 Penjualan Online. All rights reserved | This template is made with by Penjualan Online.

Gambar III. 23 Halaman Antar Muka Daftar Pengguna

10. Halaman Antar Muka Upload Bukti Pembayaran Pengguna

Halaman ini digunakan pengguna untuk mengupload bukti pembayaran berupa gambar screenshot dari struk, informasi transfer berhasil.

YOAYO STORE

BERANDA KATEGORI LIHAT KATALOG KONTAK KAMI KELUAR

Beranda / Pembayaran / Upload Bukti

Upload Bukti Pembayaran

Bukti Pembayaran

Choose File No file chosen

Pastikan format foto yang di upload : jpg, jpeg, atau png

UPLOAD BUKTI

Telusuri... Tidak ada berkas dipilih.

Pastikan format foto yang di upload : jpg, jpeg, atau png

UPLOAD BUKTI

Hubungi Kami

Nama Pengguna

Email Pengguna

subject

Isi Pesan

KIRIM PESAN

Info Kontak

Universitas BSI, Gedung D2,
Margonda Depok.

+62 852 1033 2431

imamgustono.ig@gmail.com

Tentang Kami

Kami menyediakan produk-produk alat olahraga dari berbagai merk ternama dan terbaik kepada konsumen.

Gambar III. 24 Halaman Antar Muka Upload Bukti Pembayaran Pengguna

11. Halaman Antar Muka Riwayat Pesanan Pengguna

Halaman ini digunakan pembeli untuk melihat riwayat pesanan yang meliputi kode pesanan, no.invoice, Total pembayaran, Tanggal pesanan dan Tanggal diterima.

Search: Cari Barang... PENJUALAN ONLINE

BERANDA KATEGORI LIHAT KATALOG KONTAK KAMI KELUAR

Beranda / Pesanan / Riwayat

Riwayat Pesanan

No	Kode Pesanan	No. Invoice	Total Pembayaran	Tanggal Pesanan	Tanggal Diterima
#1	PSN1904041 Lihat Detail Pesanan	INV1904042	Rp 1.180.000	2019-04-04 16:08:50	2019-04-04 16:27:04

Hubungi Kami

Nama Pengguna

Email Pengguna

subject

Isi Pesan

KIRIM PESAN

Info Kontak

Universitas BSI, Gedung D2,
Margonda Depok.

+62 852 1033 2431

imamgustono.jg@gmail.com

Tentang Kami

Kami menyediakan produk-produk alat olahraga dari berbagai merk ternama dan terbaik kepada konsumen.

Copyright © 2019 Penjualan Online. All rights reserved | This template is made with ♥ by Penjualan Online

Gambar III. 25 Halaman Antar Muka Riwayat Pesanan Pengguna

12. Halaman Antar Muka Detail Pesanan Pengguna

Halaman ini digunakan pembeli untuk melihat detail pesanan yang meliputi informasi pembeli, informasi pengiriman, daftar produk yang dibeli dan rincian total biaya.

Cari Barang...

6

0

1

BERANDA

KATEGORI

LIHAT KATALOG

KONTAK KAMI

KELUAR

Beranda

/ Pesanan

/ Detail

Detail Informasi

Informasi Pengirim

From: YoayoStore

Universitas BSI Gedung D2,

Margonda Depok, Indonesia

No. Telepon: (123) 45678910

Email: info@yoayostore.com

Informasi Penerima

To: Muhammad Iqbal

Lorem ipsum dolor sit amet, consectetur

adipiscing elit, sed do eiusmod tempor

No. Telepon: 082238277709

Dikirim Ke: Kota, Jakarta Selatan, DKI Jakarta

Informasi Pembayaran

ID Pesanan:

PSN1904041

No. Rekening:

BCA 123456789 a/n Iqbal

Tanggal Upload:

2019-04-04 16:09:11

Daftar Pesanan

#No	Nama Barang	Harga	Subtotal
#1	Mikasa MG MV 2200 Gold x 2 Berat : 666gram	Rp 590.000	Rp 1.180.000
#2	Mikasa MG MV 2200 Gold x 2 Berat : 666gram	Rp 590.000	Rp 1.180.000

Subtotal Pesanan

Layanan	:	JNE (YES)
Total Berat	:	1332gram
Ongkir	:	9000
Total Biaya	:	Rp 2.360.000

Status Pesanan :

Selesai

Di Kirim Pada :

2019-04-04 16:04:38

Di Terima Pada :

2019-04-04 16:27:04

No. Resi Pengiriman :

C6K2H03789568816

Hubungi Kami

Nama Pengguna

Email Pengguna

subject

Isi Pesan

KIRIM PESAN

Info Kontak

Universitas BSI, Gedung D2,

Margonda Depok.

+62 852 1033 2431

imamgustono.ig@gmail.com

Tentang Kami

Kami menyediakan produk-produk alat olahraga dari

berbagai merk ternama dan terbaik kepada

konsumen.

Copyright © 2019 Penjualan Online. All rights reserved | This template is made with by Penjualan Online

Gambar III. 26 Halaman Antar Muka Detail Pesanan Pengguna

13. Halaman Antar Muka Invoice Pengguna

Halaman ini berfungsi untuk melihat invoice dari pesanan yang dilakukan oleh pengguna.

PENJUALAN ONLINE

Detail Informasi — No. Invoice : INV1904042

Informasi Pengirim	Informasi Penerima	Informasi Pembayaran
<i>From, YoayoStore</i> Universitas BSI Gedung D2, Margonda Depok, Indonesia No. Telepon: (123) 45678910 Email: info@yoayostore.com	<i>To, Muhammad Iqbal</i> Lorem ipsum dolor sit amet, consectetur adipiscing elit, sed do eiusmod tempor No. Telepon: 082298277709 Pengiriman : Kota. Jakarta Selatan, DKI Jakarta	ID Pesanan: PSN1904041 No. Rekening: BCA 123456789 a/n iqbal Tanggal Upload: 2019-04-04 16:09:11

Daftar Pesanan

#No	Nama Barang	Harga	Subtotal
#1	Mikasa MG MV 2200 Gold x 2 Berat : 666gram	Rp 590.000	Rp 1.180.000
#2	Mikasa MG MV 2200 Gold x 2 Berat : 666gram	Rp 590.000	Rp 1.180.000

Subtotal Pesanan

Layanan	:	JNE (YES)
Total Berat	:	1332gram
Ongkir	:	9000
Total Biaya	:	Rp 2.360.000

Gambar III. 27 Halaman Antar Muka Invoice Pengguna

BAB V

SOURCE CODE

A. Login

```
<?php

namespace App\Http\Controllers\Admin\Autentikasi;

use Validator;
use Illuminate\Http\Request;
use Illuminate\Support\Facades\DB;
use Illuminate\Support\Facades\Hash;
use Illuminate\Support\Facades\Storage;
use App\Http\Controllers\Controller;

class LoginController extends Controller
{
    public function index(Request $request) {
        if(!$request->session()->exists('email')) {
            return view('admin.autentikasi.login');
        } else {
            return redirect()->route('beranda_admin');
        }
    }

    public function login(Request $request) {
        if($request->has('simpan')) {
            $validasi = Validator::make($request->all(), [
                'email' => 'required|email',
                'password' => 'required|alpha_num'
            ]);

            if ($validasi->fails()) {
                return redirect()->route('login_admin')->with('fail', 'Email Atau Password Salah')->withInput();
            }

            $data = DB::table('tbl_admin')->where('email', $request->input('email'))->first();

            if(!empty($data) && Hash::check($request->input('password'), $data->password)) {
```

```

        if($data->diblokir) {
            return redirect()->route('login_admin')->with('fail', 'Akun Telah Di Blokir');
        } else {
            session([
                'id_admin'      => $data->id_admin,
                'email_admin'   => $data->email,
                'nama_admin'    => $data->nama_lengkap,
                'foto_admin'    => $data->foto,
                'superadmin'    => $data->superadmin
            ]);
            return redirect()->route('beranda_admin');
        }
    } else {
        return redirect()->route('login_admin')->with('fail', 'Email Atau Password Salah')->withInput();
    }
} else {
    return back()->withErrors('Terjadi Kesalahan Saat Login')->withInput();
}
}

public function logout(Request $request) {
    $request->session()->forget([
        'id_admin',
        'email_admin',
        'nama_admin',
        'foto_admin',
        'superadmin'
    ]);

    return redirect()->route('login_admin')->with('success', 'Anda telah logout');
}
}

```

B. Transaksi

```

<?php

namespace App\Http\Controllers\Admin\Laporan;

use Validator;
use Illuminate\Support\Facades\DB;
use Illuminate\Http\Request;
use App\Http\Controllers\Controller;

class TransaksiController extends Controller
{
    public function index(Request $request) {
        if (session()->has('email_admin')) {
            if(!empty($request->input('awal')) && !empty($request->input('akhir'))){
                $awal = $request->input('awal');
                $akhir = $request->input('akhir');

                $data = DB::table('tbl_pesanan')
                    ->where('status_pesanan', 5)
                    ->whereBetween('tanggal_pesanan', [$awal,
$akhir])
                    ->get();

                return view('admin.laporan.transaksi',
['data_pesanan' => $data]);
            }

            return view('admin.laporan.transaksi');
        } else {
            return redirect()->route('login_admin')-
>withErrors('Harus login terlebih dahulu');
        }
    }

    public function print_transaksi(Request $request) {
        if (session()->has('email_admin')) {
            $validasi = Validator::make($request->all(), [
                'tanggal_awal' => 'required|regex:/^[0-9\-
]*/|max:10',
                'tanggal_akhir' => 'required|regex:/^[0-9\-
]*/|max:10'
            ]);

```

```

        if($validasi->fails()){

            return back()->withErrors($validasi);

        }

        $tanggal_awal = $request->input('tanggal_awal');
        $tanggal_akhir = $request->input('tanggal_akhir');

        $data = DB::table('tbl_pesanan')
            ->where('status_pesanan', 5)
            ->whereBetween('tanggal_pesanan', [$tanggal_awal,
            $tanggal_akhir])
            ->get();

        return view('admin.laporan.print_transaksi',
        ['data_laporan' => $data]);

    } else {

        return redirect()->route('login_admin')-
        >withErrors('Harus login terlebih dahulu');

    }

}
}

```

C. Kategori

```

<?php

namespace App\Http\Controllers\Admin\Produk;

use DateTime;
use Validator;
use Illuminate\Http\Request;
use Illuminate\Support\Facades\DB;
use App\Http\Controllers\Controller;

class KategoriController extends Controller
{
    public function index(Request $request) {

        if($request->session()->exists('email_admin')) {

            $data = DB::table('tbl_kategori')->get();

            return view('admin.produk.kategori', ['data_kategori'
            => $data]);
        }
    }
}

```

```

    } else {
        return redirect()->route('login_admin');
    }
}

public function tambah_kategori(Request $request) {
    if($request->has('simpan')) {
        $validasi = Validator::make($request->all(), [
            'nama_kategori' => 'required|regex:/^[a-zA-Z\s]*$/|max:30'
        ]);
        if ($validasi->fails()) {
            return back()->withErrors($validasi);
        }
        if(DB::table('tbl_kategori')->where('nama_kategori',
            $request->input('nama_kategori'))->exists() == false) {
            DB::table('tbl_kategori')->insert([
                'id_kategori' => $this->set_id_kategori(),
                'nama_kategori' => $request->input('nama_kategori'),
            ]);
            return redirect()->route('kategori_produk')->with('success', 'kategori Produk Berhasil Di Tambah');
        } else {
            return back()->withErrors('Kategori tidak dapat di gunakan karna telah tersedia');
        }
    } else {
        return back()->withErrors('Terjadi Kesalahan Saat Menyimpan Harap Gunakan Tombol Simpan Untuk Menyimpan Data');
    }
}

public function edit_kategori(Request $request, $id_kategori)
{

```

```

        if($request->has('simpan')) {

            $validasi = Validator::make($request->all(), [
                'nama_kategori' => 'required|regex:/^[a-zA-Z\s]*$/|max:30'
            ]);

            if ($validasi->fails()) {

                return back()->withErrors($validasi);

            }

            if(DB::table('tbl_kategori')->where('nama_kategori',
                $request->input('nama_kategori'))->exists() == false) {

                DB::table('tbl_kategori')->where('id_kategori',
                    $id_kategori)
                    ->update(['nama_kategori' => $request-
                        >input('nama_kategori')]);

                return redirect()->route('kategori_produk')-
                    >with('success', 'Kategori Produk Berhasil Di Rubah');

            } else {

                return redirect()->route('kategori_produk')-
                    >withErrors('Kategori tidak dapat di gunakan karna telah
                    tersedia');

            }

        } else {

            return back()->withErrors('Terjadi Kesalahan Saat
                Menyimpan Harap Gunakan Tombol Simpan Untuk Menyimpan Data');

        }

    }

    public function hapus_kategori(Request $request, $id_kategori)
    {

        if($request->has('simpan')) {

            DB::table('tbl_kategori')->where('id_kategori',
                $id_kategori)->delete();

            return redirect()->route('kategori_produk')-
                >with('success', 'Kategori Produk Berhasil Di Hapus');

        } else {

            return back()->withErrors('Terjadi Kesalahan Saat

```

```

Menyimpan Harap Gunakan Tombol Simpan Untuk Menyimpan Data');

    }

}

public function get_kategori(Request $request) {

    $id_kategori = $request->input('id_kategori');

    $data = DB::table('tbl_kategori')->where('id_kategori',
$id_kategori)->first();

    return response()->json($data);
}

protected function set_id_kategori() {
    $data = DB::table('tbl_kategori')->max('id_kategori');

    if(!empty($data)) {

        $no_urut = substr($data, 9, 3) + 1;

        return 'KTG'.(new Datetime)->format('ymd').$no_urut;
    } else {
        return 'KTG'.(new Datetime)->format('ymd').'1';
    }
}
}

```

D. Merk

```

<?php

namespace App\Http\Controllers\Admin\Produk;

use DateTime;
use Validator;
use Illuminate\Http\Request;
use Illuminate\Support\Facades\DB;
use App\Http\Controllers\Controller;

class MerkController extends Controller
{
    public function index(Request $request) {

        if($request->session()->exists('email_admin')) {

            $data = DB::table('tbl_merk')->get();

            return view('admin.produk.merk', ['data_merk' =>
$data]);

```

```

    } else {
        return redirect()->route('login_admin');
    }
}

public function tambah_merk(Request $request) {
    if($request->has('simpan')) {
        $validasi = Validator::make($request->all(), [
            'nama_merk' => 'required|regex:/^[a-zA-Z\s]*$/|max:30'
        ]);
        if ($validasi->fails()) {
            return back()->withErrors($validasi);
        }
        if(DB::table('tbl_merk')->where('nama_merk', $request->input('nama_merk'))->exists() == false) {
            DB::table('tbl_merk')->insert([
                'id_merk' => $this->set_id_merk(),
                'nama_merk' => $request->input('nama_merk'),
            ]);
            return redirect()->route('merk_produk')->with('success', 'Merk Produk Berhasil Di Tambah');
        } else {
            return redirect()->route('merk_produk')->withErrors('Merk tidak dapat di gunakan karna telah tersedia');
        }
    } else {
        return back()->withErrors('Terjadi Kesalahan Saat Menyimpan Harap Gunakan Tombol Simpan Untuk Menyimpan Data');
    }
}

public function edit_merk(Request $request, $id_merk) {
    if($request->has('simpan')) {

```

```

        $validasi = Validator::make($request->all(), [
            'nama_merk' => 'required|regex:/^[a-zA-Z\s]*$/|max:30'
        ]);

        if ($validasi->fails()) {

            return back()->withErrors($validasi);

        }

        if(DB::table('tbl_merk')->where('nama_merk', $request->input('nama_merk'))->exists() == false) {

            DB::table('tbl_merk')->where('id_merk', $id_merk)
                ->update(['nama_merk' => $request->input('nama_merk')]);

            return redirect()->route('merk_produk')->with('success', 'Merk Produk Berhasil Di Rubah');

        } else {

            return back()->withErrors('Merk tidak dapat di gunakan karna telah tersedia');

        }

    } else {

        return back()->withErrors('Terjadi Kesalahan Saat Menyimpan Harap Gunakan Tombol Simpan Untuk Menyimpan Data');

    }

}

public function hapus_merk(Request $request, $id_merk) {

    if($request->has('simpan')) {

        DB::table('tbl_merk')->where('id_merk', $id_merk)->delete();

        return redirect()->route('merk_produk')->with('success', 'Merk Produk Berhasil Di Hapus');

    } else {

        return back()->withErrors('Terjadi Kesalahan Saat Menyimpan Harap Gunakan Tombol Simpan Untuk Menyimpan Data');

    }

}

```

```

    public function get_merk(Request $request) {
        $id_merk = $request->input('id_merk');

        $data = DB::table('tbl_merk')->where('id_merk', $id_merk)->first();
        echo response()->json($data);
    }

    public function check_merk(){
        $nama_merk = str_replace('%20', ' ', $_GET['nama_merk']);

        $data = DB::table('tbl_merk')->where('nama_merk', $nama_merk)->exists();

        return response()->json($data);
    }

    protected function set_id_merk() {
        $data = DB::table('tbl_merk')->max('id_merk');

        if(!empty($data)) {
            $no_urut = substr($data, 9, 3) + 1;

            return 'MRK'.(new Datetime)->format('ymd').$no_urut;
        } else {
            return 'MRK'.(new Datetime)->format('ymd').'1';
        }
    }
}

```

E. roduk

```

<?php

namespace App\Http\Controllers\Admin\Produk;

use DateTime;
use Validator;
use Illuminate\Http\Request;
use Illuminate\Support\Facades\DB;

```

```

use Illuminate\Support\Facades\Storage;
use App\Http\Controllers\Controller;

class ProdukController extends Controller
{
    public function index(Request $request) {
        if($request->session()->exists('email_admin')) {
            if(!empty($request->input('merk')) && !empty($request->input('kategori'))) {
                $id_kategori = DB::table('tbl_kategori')->where('nama_kategori', ucfirst($request->input('kategori')))->first();
                $id_merk = DB::table('tbl_merk')->where('nama_merk', ucfirst($request->input('merk')))->first();

                $data = DB::table('tbl_barang')
                    ->join('tbl_kategori', 'tbl_kategori.id_kategori', 'tbl_barang.id_kategori')
                    ->join('tbl_merk', 'tbl_merk.id_merk', 'tbl_barang.id_merk')
                    ->select('tbl_barang.*', 'tbl_kategori.*', 'tbl_merk.*')
                    ->where([
                        ['tbl_merk.id_merk', $id_merk->id_merk],
                        ['tbl_kategori.id_kategori', $id_kategori->id_kategori]
                    ])->get();
            } else if (!empty($request->input('kategori'))) {
                $id_kategori = DB::table('tbl_kategori')->where('nama_kategori', ucfirst($request->input('kategori')))->first();

                $data = DB::table('tbl_barang')
                    ->join('tbl_kategori', 'tbl_kategori.id_kategori', 'tbl_barang.id_kategori')
                    ->join('tbl_merk', 'tbl_merk.id_merk', 'tbl_barang.id_merk')
                    ->select('tbl_barang.*', 'tbl_kategori.*', 'tbl_merk.*')
                    ->where('tbl_kategori.id_kategori', $id_kategori->id_kategori)->get();
            } else if(!empty($request->input('merk'))) {
                $id_merk = DB::table('tbl_merk')->where('nama_merk', ucfirst($request->input('merk')))->first();

                $data = DB::table('tbl_barang')
                    ->join('tbl_kategori',

```

```

'tbl_kategori.id_kategori', 'tbl_barang.id_kategori')
    ->join('tbl_merk', 'tbl_merk.id_merk',
'tbl_barang.id_merk')
    ->select('tbl_barang.*',
'tbl_kategori.nama_kategori', 'tbl_merk.*')
    ->where('tbl_merk.id_merk', $id_merk-
>id_merk)->get();

    } else {

        $data = DB::table('tbl_barang')
            ->join('tbl_kategori',
'tbl_kategori.id_kategori', 'tbl_barang.id_kategori')
            ->join('tbl_merk', 'tbl_merk.id_merk',
'tbl_barang.id_merk')
            ->select('tbl_barang.*',
'tbl_kategori.nama_kategori', 'tbl_merk.nama_merk')
            ->get();

    }

    $merk = DB::table('tbl_merk')->get();
    $kategori = DB::table('tbl_kategori')->get();

    return view('admin.produk.produk', ['data_produk' =>
$data,
                                     'data_merk' =>
$merk,
                                     'data_kategori' =>
$kategori]);

    } else {

        return redirect()->route('login_admin')->with('fail',
'Harap Login Terlebih Dahulu');

    }

}

public function tambah_produk(Request $request) {

    if($request->has('simpan')) {

        $validasi = Validator::make($request->all(), [
            'nama_barang' => 'required|regex:/^[a-zA-Z0-9\s]*$/|max:50',
            'deskripsi_barang' => 'required',
            'berat_barang' => 'required|integer',
            'harga_satuan' => 'required|integer',
            'stok_barang' => 'required|integer',
            'foto_barang' =>
'required|mimes:jpg,jpeg,png'
        ]);
    }

```

```

        if ($validasi->fails()) {
            return back()->withErrors($validasi);
        }

        if(DB::table('tbl_barang')->where('nama_barang',
$request->input('nama_barang'))->exists() == false) {

            $id_barang = $this->set_id_barang();

            $extension = $request->file('foto_barang')-
>getClientOriginalExtension();

            $foto_produk = Storage::putFileAs(
                'public/produk/',
                $request->file('foto_barang'),
                $id_barang.'.'.$extension
            );

            DB::table('tbl_barang')->insert([
                'id_barang'          => $id_barang,
                'nama_barang'        => $request-
>input('nama_barang'),
                'id_kategori'        => $request-
>input('id_kategori'),
                'id_merk'           => $request-
>input('id_merk'),
                'deskripsi_barang'  => $request-
>input('deskripsi_barang'),
                'berat_barang'       => $request-
>input('berat_barang'),
                'harga_satuan'       => $request-
>input('harga_satuan'),
                'stok_barang'        => $request-
>input('stok_barang'),
                'foto_barang'        => basename($foto_produk),
            ]);

            return redirect()->route('list_produk')-
>with('success', 'Produk Berhasil DI Simpan');

        } else {

            return back()->withErrors('Produk tidak dapat di
simpan karna telah tersedia');
        }

    } else {

        return back()->withErrors('Terjadi Kesalahan Saat
Menyimpan Harap Gunakan Tombol Simpan Untuk Menyimpan Data');
    }
}

```

```

    }

    }

    public function edit_produk(Request $request, $id_barang) {

        if($request->has('simpan')) {

            $validasi = Validator::make($request->all(), [
                'nama_barang'    => 'required|regex:/^[a-zA-Z0-9\s]*$/|max:50',
                'berat_barang'    => 'required|integer',
                'harga_satuan'    => 'required|integer',
                'stok_barang'     => 'required|integer',
            ]);

            if ($validasi->fails()) {

                return back()->withErrors($validasi);

            }

            $data = DB::table('tbl_barang')->select('foto_barang')->where('id_barang', $id_barang)->first();

            if($request->hasFile('foto_barang')) {

                Storage::delete('public/produk/'.$data->foto_barang);

                $extension = $request->file('foto_barang')->getClientOriginalExtension();

                $save_foto = Storage::putFileAs(
                    'public/produk/',
                    $request->file('foto_barang'),
                    $id_barang.'.'.$extension
                );

                $foto_produk = basename($save_foto);

            }

            DB::table('tbl_barang')->where('id_barang', $id_barang)
                ->update([
                    'nama_barang'    => $request->input('nama_barang'),
                    'id_kategori'     => $request->input('id_kategori'),
                    'id_merk'         => $request->input('id_merk'),
                    'berat_barang'    => $request->input('berat_barang'),
                    'harga_satuan'    => $request->input('harga_satuan')
                ]);
        }
    }
}

```

```

>input('harga_satuan'),
        'stok_barang'    => $request-
>input('stok_barang'),
        'foto_barang'    => $request-
>hasFile('foto_barang') ? $foto_produk : $data->foto_barang,
    ]);

        return redirect()->route('list_produk')-
>with('success', 'Produk Berhasil DI Simpan');

    } else {

        return back()->withErrors('Terjadi Kesalahan Saat
Menyimpan Harap Gunakan Tombol Simpan Untuk Menyimpan Data');

    }
}

public function hapus_produk(Request $request, $id_barang) {

    $data = DB::table('tbl_barang')->where('id_barang',
$id_barang);

    Storage::delete('public/produk/' . $data->first()-
>foto_barang);

    $data->delete();

    return redirect()->route('list_produk')->with('success',
'Produk Berhasil Di Hapus');

}

public function get_barang() {

    $id_barang = $_GET['id_barang'];

    $data = DB::table('tbl_barang')->where('id_barang',
$id_barang)->first();

    return response()->json($data);

}

protected function set_id_barang() {
    $data = DB::table('tbl_barang')->max('id_barang');

    if(!empty($data)) {

        $no_urut = substr($data, 9, 3) + 1;

        return 'BRG' . (new Datetime)->format('ymd') . $no_urut;
    } else {
        return 'BRG' . (new Datetime)->format('ymd') . '1';
    }
}

```

```

    }
}

```

F. Invoice

```

<?php
namespace App\Http\Controllers\Admin\Transaksi;

use DateTime;
use Illuminate\Support\Facades\DB;
use Illuminate\Http\Request;
use App\Http\Controllers\Controller;

class InvoiceController extends Controller
{
    public function save_invoice($id_pesanan, $id_pengguna) {
        if(session()->has('email_admin')) {
            $data = DB::table('tbl_invoice')->insert([
                'id_invoice'    => $this->set_id_invoice(),
                'id_pesanan'    => $id_pesanan,
                'id_pengguna'   => $id_pengguna
            ]);
        } else {
            return redirect()->route('login_admin')->with('fail',
'Harap Login Terlebih Dahulu');
        }
    }

    public function delete_invoice($id_invoice) {
        if(session()->has('email_admin')) {
            $data = DB::table('tbl_invoice')->where([
                'id_invoice'    => $id_invoice,
            ])->delete();
        } else {
            return redirect()->route('login_admin')->with('fail',
'Harap Login Terlebih Dahulu');
        }
    }
}

```

```

protected function set_id_invoice() {
    $data = DB::table('tbl_invoice')->max('id_invoice');
    if(!empty($data)) {
        $no_urut = substr($data, 9, 3) + 1;
        return 'INV'.(new Datetime)->format('ymd').$no_urut;
    } else {
        return 'INV'.(new Datetime)->format('ymd').'1';
    }
}
}

```

G. Pembayaran

```

<?php
namespace App\Http\Controllers\Admin\Transaksi;

use Validator;
use Illuminate\Support\Facades\DB;
use Illuminate\Http\Request;
use App\Http\Controllers\Controller;
use App\Http\Controllers\Admin\Transaksi\InvoiceController;

class PembayaranController extends Controller
{
    public function index(Request $request) {
        if($request->session()->exists('email_admin')) {
            $data = DB::table('tbl_pembayaran as pembayaran')
                ->join('tbl_pesanan as pesanan',
                'pesanan.id_pesanan', 'pembayaran.id_pesanan')
                ->select('pembayaran.*', 'pesanan.status_pesanan',
                'pesanan.dibatalakan')
                ->where('pembayaran.selesai', 0)
                ->get();

            return view('admin.transaksi.pembayaran',
            ['data_pembayaran' => $data]);
        } else {
            return redirect()->route('login_admin')->with('fail',
            'Harap Login Terlebih Dahulu');
        }
    }
}

```

```

public function rubah_status(Request $request, $id_pesanan) {

    /**
     * STATUS PEMBAYARAN
     * =====
     * - 0 => Menunggu Verifikasi
     * - 1 => Telah Di Diterima
     */

    if($request->has('simpan') == true) {

        $pesanan = DB::table('tbl_pesanan')-
        >where('id_pesanan', $id_pesanan);
        $pembayaran = DB::table('tbl_pembayaran')-
        >where('id_pesanan', $id_pesanan);

        $invoice = new InvoiceController();

        if ($pembayaran->first()->status_pembayaran == 0) {

            $invoice->save_invoice($id_pesanan, $pesanan-
            >first()->id_pengguna);

        } else {

            $inv = DB::table('tbl_invoice')-
            >where('id_pesanan', $id_pesanan)->first();

            $invoice->delete_invoice($inv->id_invoice);

        }

        $data = $pesanan->update(['status_pesanan' =>
        $pesanan->first()->status_pesanan == 0 ? 1 : 0]);
        $pembayaran->update(['status_pembayaran' =>
        $pembayaran->first()->status_pembayaran == 0 ? 1 : 0]);

        return redirect()->route('pembayaran_admin')-
        >with('success', 'Pembayaran Dengan ID '.$id_pesanan.' Berhasil Di
        Update');

    } else {

        return back()->withErrors('Terjadi Kesalahan Saat
        Menyimpan Data');

    }

}

public function get_pembayaran(Request $request, $id_pesanan)
{

    if($request->session()->exists('email_admin')) {

```

```

        $data = DB::table('tbl_pembayaran')-
>where('id_pesanan', $id_pesanan)->first();

        return response()->json($data);

    } else {

        return redirect()->route('login_admin')->with('fail',
'Harap Login Terlebih Dahulu');

    }

}
}

```

H. Pengiriman

```

<?php

namespace App\Http\Controllers\Admin\Transaksi;

use DateTime;
use Validator;
use Illuminate\Support\Facades\DB;
use Illuminate\Http\Request;
use App\Http\Controllers\Controller;

class PengirimanController extends Controller
{
    public function index(Request $request) {

        if ($request->session()->exists('email_admin')) {

            $data = DB::table('tbl_pesanan')
                ->select(
                    'id_pesanan', 'nama_penerima',
                    'alamat_tujuan', 'layanan', 'tanggal_diterima',
                    'no_telepon', 'no_resi', 'status_pesanan',
                    'tanggal_dikirim'
                )->get();

            return view('admin.transaksi.pengiriman',
['data_pengiriman' => $data]);

        } else {

```

```

        return redirect()->route('login_admin')->with('fail',
'Harpas Login Terlebih Dahulu');

    }

}

public function batalkan_pesanan(Request $request,
$id_pesanan) {

    if($request->has('simpan') == true) {

        $data = DB::table('tbl_pesanan')->where('id_pesanan',
$id_pesanan);

        $status = $data->first()->dibatalkan;

        if($data->exists()) {

            $data->update([
                'dibatalkan'          => 1,
                'status_pesanan'      => 1,
                'no_resi'              => NULL,
                'tanggal_dikirim'     => NULL
            ]);

            return redirect()->route('pengiriman_admin')-
>with('success',
                $status == 0 ? 'Pesanan Berhasil Di Batalkan'
: 'Pembatalan Pesanan Berhasil Di Cabut');

        } else {

            return back()->withErrors('Terjadi Kesalahan Saat
Menyimpan Data');

        }

    } else {

        return back()->withErrors('Terjadi Kesalahan Saat
Menyimpan Data');

    }

}

public function selesai(Request $request, $id_pesanan) {

    if ($request->session()->exists('email_admin')) {

        $data = DB::table('tbl_pesanan')->where([
            'id_pesanan'              => $id_pesanan,
            'status_pesanan'          => 4

```

```

    });

    if(!empty($data->first())) {

        $data->update(['status_pesanan' => 5]);

        DB::table('tbl_pembayaran')
            ->where('id_pesanan', $id_pesanan)
            ->update(['selesai' => 1]);

    }

    return back()->with('success', 'Pesanan Dengan ID
'.'.$id_pesanan.' Telah Selesai');

    } else {

        return redirect()->route('login_admin')->with('fail',
'Harap Login Terlebih Dahulu');

    }

}
}
}

```

I. Pesanan

```

<?php

namespace App\Http\Controllers\Admin\Transaksi;

use DateTime;
use Validator;
use Illuminate\Support\Facades\DB;
use Illuminate\Http\Request;
use App\Http\Controllers\Controller;

class PesananController extends Controller
{
    /**
     * STATUS PESANAN
     * =====
     * - 0 => Belum Di Proses
     * - 1 => Telah Di Verifikasi
     * - 2 => Sedang Di Proses
     * - 3 => Telah Di Kirim
     * - 4 => Telah Di Terima
     * - 5 => Selesai
     */

    public function index(Request $request) {

        if($request->session()->exists('email_admin')) {

            $stat_label = [
                'bg-gray', 'label-info', 'bg-blue',
                // 'bg-navy', 'bg-orange', 'bg-green'
            ];
        }
    }
}

```

```

];

$stat_notif = [
    'Belum Di Proses', 'Sedang Di Proses', 'Siap
Dikirim',
    // 'Telah Di Kirim', 'Telah Di Terima', 'Selesai'
];

$data = DB::table('tbl_pesanan as pesanan')
    ->join('tbl_pembayaran as pembayaran',
'pembayaran.id_pesanan', 'pesanan.id_pesanan')
    ->select('pesanan.*', 'pembayaran.*')
    ->get();

return view('admin.transaksi.pesanan', [
    'data_pesanan' => $data,
    'stat_label'    => $stat_label,
    'stat_notif'    => $stat_notif
]);

} else {

    return redirect()->route('login_admin')->with('fail',
'Harap Login Terlebih Dahulu');

}

}

public function detail_pesanan(Request $request, $id_pesanan)
{

    if($request->session()->exists('email_admin')) {

        $status = [
            'Belum Di Proses',
            'Telah Di Verifikasi',
            'Sedang Di Proses',
            'Telah Di Kirim',
            'Telah Di Terima',
            'Selesai'
        ];

        $data_pesanan = DB::table('tbl_pesanan')-
>where('id_pesanan', $id_pesanan)->first();
        $data_pembayaran = DB::table('tbl_pembayaran')-
>where('id_pesanan', $id_pesanan)->first();
        $data_detail = DB::table('tbl_detail_pesanan')
            ->join('tbl_barang', 'tbl_barang.id_barang',
'tbl_detail_pesanan.id_barang')
            ->select('tbl_barang.*', 'tbl_detail_pesanan.*')
            ->where('tbl_detail_pesanan.id_pesanan',
$id_pesanan)->get();

```

```

        return view('admin.transaksi.detail_pesanan', [
            'data_pesanan' => $data_pesanan,
            'pembayaran'    => $data_pembayaran,
            'data_detail'   => $data_detail,
            'status'        => $status,
            'invoice'       => true
        ]);

    } else {

        return redirect()->route('login_admin')->with('fail',
'Harpap Login Terlebih Dahulu');

    }

}

public function edit_pesanan(Request $request, $id_pesanan) {

    if ($request->has('simpan')) {

        $validasi = Validator::make($request->all(), [

            'nama_penerima' => 'required|regex:/^[a-zA-Z\s]*$/|max:40',
            'alamat_tujuan' => 'required|string',
            'no_telepon'     => 'required|regex:/^[0-9\-\+]\s]*$/|max:18',

        ]);

        if($validasi->fails()) {

            return back()->withErrors($validasi);

        }

        $data = DB::table('tbl_pesanan')->where('id_pesanan',
$id_pesanan);

        if($data->first()->status_pesanan <= 2) {

            $data->update([

                'nama_penerima' => $request-
>input('nama_penerima'),
                'alamat_tujuan' => $request-
>input('alamat_tujuan'),
                'no_telepon'     => $request-
>input('no_telepon')

            ]);

            return redirect()->route('pesanan_admin')->

```

```

>with('success', 'Data Pengiriman Berhasil Di Simpan');

        } else {

            return back()->withErrors('Terjadi Kesalahan Saat
Menyimpan Data');

        }

    } else {

        return back()->withErrors('Terjadi Kesalahan Saat
Menyimpan Data Harap Gunakan Tombol Simpan');

    }

}

public function proses_pesanan(Request $request, $id_pesanan)
{

    if($request->has('simpan') == true) {

        $data = DB::table('tbl_pesanan')->where('id_pesanan',
$id_pesanan);

        $status = $data->first()->status_pesanan;

        if($status <= 2) {

            $data->update([
                'status_pesanan'    => $status == 1 ? 2 : 1,
            ]);

            return redirect()->route('pesanan_admin')-
>with('success', 'Status Berhasil DI Rubah');

        } else {

            return back()->withErrors('Terjadi Kesalahan Saat
Menyimpan Data');

        }

    } else {

        return back()->withErrors('Terjadi Kesalahan Saat
Menyimpan Data');

    }

}

public function kirim_pesanan(Request $request, $id_pesanan) {

    if($request->has('simpan') == true) {

```

```

        $validasi = Validator::make($request->all(), [
            'resi' => 'required|alpha_num|max:20'
        ]);

        if($validasi->fails()) {
            return back()->withErrors($validasi);
        }

        $data = DB::table('tbl_pesanan')->where('id_pesanan',
$id_pesanan);

        $status = $data->first()->status_pesanan;

        if($status <= 3 && $status > 1) {

            $data->update([
                'no_resi' => $status == 2 ?
$request->input('resi') : NULL,
                'status_pesanan' => $status == 2 ? 3 : 2,
                'tanggal_dikirim' => $status == 2 ? (new
DateTime)->format('Y-m-d H:m:s') : NULL
            ]);

            return redirect()->route('pesanan_admin')-
>with('success',
                $status == 2 ? 'Pesanan Berhasil DI Proses' :
'Proses Pesanan Berhasil Di Batalkan');

        } else {

            return back()->withErrors('Terjadi Kesalahan Saat
Menyimpan Data');

        }

    } else {

        return back()->withErrors('Terjadi Kesalahan Saat
Menyimpan Data');

    }

}

public function batalkan_pesanan(Request $request,
$id_pesanan) {

    if($request->has('simpan') == true) {

```

```

        $data = DB::table('tbl_pesanan')->where('id_pesanan',
$id_pesanan);

        $status = $data->first()->dibatalkan;

        if($data->first()->status_pesanan < 3) {

            $data->update([
                'dibatalkan' => $status == 0 ? 1 : 0,
            ]);

            return redirect()->route('pesanan_admin')-
>with('success',
                $status == 0 ? 'Pesanan Berhasil Di Batalkan'
: 'Pembatalan Pesanan Berhasil Di Cabut');

        } else {

            return back()->withErrors('Terjadi Kesalahan Saat
Menyimpan Data');

        }

    } else {

        return back()->withErrors('Terjadi Kesalahan Saat
Menyimpan Data');

    }

}

public function hapus_pesanan(Request $request, $id_pesanan) {

    if($request->has('simpan') == true) {

        $data = DB::table('tbl_pesanan')->where('id_pesanan',
$id_pesanan);

        if($data->first()->status_pesanan < 3) {

            DB::table('tbl_invoice')->where('id_pesanan',
$id_pesanan)->delete();

            $data->delete();

            return redirect()->route('pesanan_admin')-
>with('success', 'Pesanan Berhasil Di Hapus');

        } else {

            return back()->withErrors('Terjadi Kesalahan Saat
Menyimpan Data');

```

```
    }  
    } else {  
        return back()->withErrors('Terjadi Kesalahan Saat  
Menyimpan Data');  
    }  
}  
  
public function get_info_penerima(Request $request,  
$id_pesanan) {  
    if($request->session()->exists('email_admin')) {  
        $data = DB::table('tbl_pesanan')->where('id_pesanan',  
$id_pesanan)->first();  
        return response()->json($data);  
    } else {  
        return redirect()->route('login_admin')->with('fail',  
'Harap Login Terlebih Dahulu');  
    }  
}  
}
```