

Pembuatan Aplikasi *Point of Sale* Toko Cabang Perusahaan Torani Menggunakan *Framework CodeIgniter*

Jodhi Sugihartono, Kodrat Iman Satoto, Eko Didik Widiyanto
Program Studi Sistem Komputer, Fakultas Teknik, Universitas Diponegoro
Jl. Prof. Sudharto, Tembalang, Semarang, Indonesia
E-mail : jodhisugihartono25@gmail.com

Abstrak – Dalam Industri yang semakin pesat perkembangannya, proses transaksi dan keluar masuknya barang perlu dicatat, ini diperlukan untuk mempermudah suatu perusahaan dalam mengontrol jumlah nilai pemasukan dan pengeluaran. Data fisik merupakan sesuatu yang mudah hilang, dimana dalam pendataan juga dimungkinkan adanya *Human Error*, salah satu akibatnya adalah dapat menyebabkan suatu perbedaan data, kesalahan pendataan, kekurangan pemesanan, dan kesalahan perhitungan jumlah pemasukan dan pengeluaran. Aplikasi ini dapat membantu perusahaan dalam pendataan barang dan transaksi menjadi lebih mudah dilakukan. Pemesanan barang pun dapat dilakukan melalui aplikasi tanpa harus memesan dengan menggunakan telepon. Aplikasi ini dibuat dengan menggunakan bahasa pemrograman PHP, Jquery, database MySQL, dan *framework CodeIgniter*.

Kata Kunci : Point of Sale, PHP, MySQL, CodeIgniter.

I. PENDAHULUAN

Dalam perkembangan teknologi yang pesat, komunikasi dan pertukaran data yang cepat sangat diperlukan oleh suatu lembaga maupun. Dengan adanya aplikasi *point of sale* ini dimungkinkan dapat melakukan suatu pekerjaan pendataan maupun transaksi barang dengan cepat, tepat dan lebih efektif serta efisien.

Pada suatu perusahaan yang memiliki relasi kerjasama dengan pihak luar lainnya, khususnya untuk perusahaan yang bergerak di bidang industri perdagangan, jumlah keluar masuknya barang dan proses pemesanan barang yang habis di setiap toko cabang yang dimiliki perusahaan perlu didata sehingga setiap transaksi yang dilakukan dapat diawasi dan dicek dengan mudah oleh toko itu sendiri maupun oleh perusahaan. Pendataan barang dan transaksi jual beli secara manual dapat dilakukan, namun ini memiliki banyak kekurangan. Faktor *human error* merupakan suatu faktor yang dapat mengakibatkan banyak masalah, seperti terjadinya kesalahan pendataan, kekurangan pemesanan, kesalahan penghitungan pemasukan dan pengeluaran, dan lain sebagainya, yang mana ini dapat menyebabkan kerugian bagi perusahaan tersebut.

Aplikasi *point of sale* ini dibuat untuk meminimalisasi pendataan barang dan transaksi secara manual serta penggunaan telepon untuk melakukan pemesanan terhadap distributor. Dengan adanya aplikasi ini, maka setiap pendataan barang di toko dan pemesanan barang yang habis di toko dapat dilakukan secara cepat dengan mengacu pada database yang ada pada perusahaan, sehingga ini akan membuat pekerja melakukan pekerjaan secara efektif dan efisien.

Sistem *Point of Sale* (POS) adalah sebuah sistem aplikasi yang diterapkan pada bisnis minimarket ataupun pertokoan untuk menangani pengolahan data transaksi pembelian (*purchases*), transaksi penjualan eceran (*retails*), transaksi hutang (*liabilities*), transaksi retur pembelian (*purchase returns*), dan pelaporan transaksi (*reporting*) yang secara umum penting dibutuhkan dalam pengambilan keputusan strategik oleh para pebisnis swalayan, organisasi, atau perusahaan yang berskala kecil dan menengah.^[4]

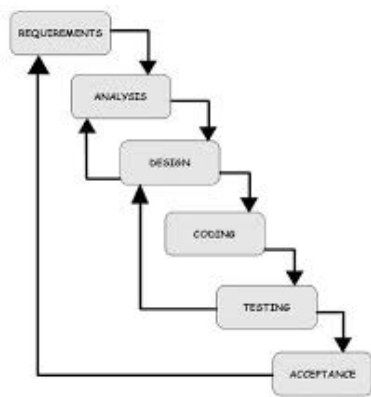
Manfaat dari sistem *point of sale* untuk sebuah organisasi salah satunya adalah untuk mengolah transaksi-transaksi, mengurangi biaya dan menghasilkan pendapatan sebagai salah satu produk atau pelayanan mereka. Perusahaan menggunakan sistem *point of sale* untuk mempertahankan persediaan pada tingkat rendah agar konsisten dengan jenis barang yang tersedia.^[1]

Tujuan dilakukan pembuatan aplikasi *point of sale* penelitian dalam Tugas Akhir ini adalah untuk merancang sistem pendataan barang yang dapat digunakan untuk memudahkan pengecekan jumlah stok barang di toko, merancang sistem pemesanan barang di toko cabang untuk memudahkan pemesanan barang yang habis di toko, merancang sistem kasir untuk memudahkan proses transaksi jual beli dengan pelanggan toko, dan membagi sistem menjadi 3 bagian menu untuk membagi tugas antara admin, pegawai, dan petugas kasir secara terstruktur.

Dalam pembuatan Tugas Akhir ini pembahasan masalah memiliki batasan permasalahan, antara lain pembuatan aplikasi *point of sale* ditujukan pada lingkup toko cabang perusahaan pemasok bahan olahan ikan mentah Torani Cirebon, pembuatan aplikasi *point of sale* toko cabang perusahaan Torani menggunakan *framework CodeIgniter* dan database MySQL, dan Hasil aplikasi akan disimulasikan di *web browser* secara *offline* maupun *online*.

II. METODE PENELITIAN

Sebuah sistem yang baik tidak lepas dari sebuah perancangan yang matang. Layaknya membuat sebuah bangunan, tentulah dibutuhkan sebuah *blueprint* agar bangunan yang akan dibuat nantinya dapat tepat sasaran dan tepat guna. Metode yang digunakan dalam perancangan aplikasi tugas akhir ini adalah metode *Waterfall*. Metode *Waterfall* terdiri dari spesifikasi (*requirements*), analisis (*analysis*), perancangan (*design*), implementasi (*implementation*), dan pengujian (*testing*). Proses dalam metode *Waterfall* digambarkan pada Gambar 1.



Gambar 1. Tahapan metode *Waterfall*

Tahap pertama yang dilakukan adalah spesifikasi kebutuhan, yang bertujuan untuk mengetahui masalah yang sedang terjadi yang ada pada toko cabang perusahaan Torani.

Tahap kedua adalah analisa kebutuhan, yang dilakukan untuk mencari solusi yang tepat untuk memecahkan masalah yang ada pada toko cabang perusahaan Torani seperti yang diketahui pada spesifikasi kebutuhan.

Tahap ketiga adalah perancangan aplikasi. Hasil analisa yang telah didapat sebelumnya dituangkan dalam perancangan untuk menentukan fungsi-fungsi apa saja yang nantinya terdapat dalam aplikasi tersebut untuk memenuhi kebutuhan aplikasi tertentu.

Setelah melewati tahap perancangan, pada tahap keempat, hasil dari rancangan yang sudah ada tersebut diimplementasikan ke dalam kode-kode program sehingga terbentuklah suatu aplikasi jadi yang siap pakai.

Tahap kelima adalah pengujian. Metode pengujian yang digunakan untuk menguji sistem adalah dengan menggunakan metode pengujian *Black Box*. Metode pengujian ini akan menguji seluruh komponen dan fungsionalitas sistem apakah perangkat lunak yang dibuat telah dapat berjalan dengan benar dan sesuai dengan tujuan yang ingin dicapai. Pengujian dilakukan terhadap aplikasi untuk mengidentifikasi sekaligus memperbaiki kesalahan-kesalahan yang ada. Hasil dari tahapan ini adalah aplikasi yang sudah terbebas dari kesalahan-kesalahan, dapat berjalan sesuai dengan fungsinya, dan siap digunakan.^[2]

III. HASIL DAN PENGUJIAN

Setelah melakukan wawancara dan pengamatan secara langsung dengan pihak terkait terhadap keadaan di toko Torani, maka didapat hasil spesifikasi kebutuhan dimana proses kerja dibagi menjadi beberapa tahapan yang tertuang dalam tabel berikut :

Tabel 1. Kesimpulan proses bisnis

Proses	Informasi	Kendala
Pendaftaran karyawan baru	Kode karyawan, nama lengkap, <i>username</i> , <i>password</i> , tanggal lahir, alamat lengkap, nomor telepon, jabatan	Proses pendaftaran masih manual dan pencarian data karyawan masih dilakukan dengan mencari data secara manual
Penambahan Produk Baru dan Penambahan Stok Barang	Kode produk, nama produk, detail produk, harga produk, stok produk	Proses pendataan barang masih manual dengan menggunakan buku catatan

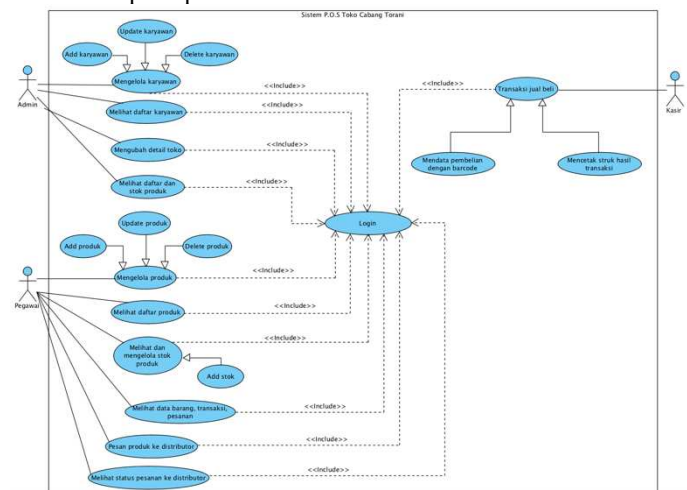
Pemesanan barang ke distributor	Kode produk, nama produk, jumlah produk	Pemesanan ke distributor masih dilakukan via telepon
Transaksi kasir dengan konsumen	Kode produk, nama produk, harga produk, jumlah, total harga	Proses transaksi masih dilakukan dengan manual dan perhitungan barang dan harga masih dihitung satu per satu

Untuk memahami sifat dan fungsi sistem yang akan dibangun, analisa yang dilakukan harus memahami domain informasi, pembagian fungsi, dan tingkah laku yang diperlukan dan mengacu pada tahap sebelumnya. Sistem yang dibangun akan dapat digunakan oleh 3 aktor, yaitu administrator, pegawai, dan kasir. Deskripsi dari masing-masing aktor dapat dilihat adalah sebagai berikut :

Tabel 2. Deskripsi masing-masing aktor

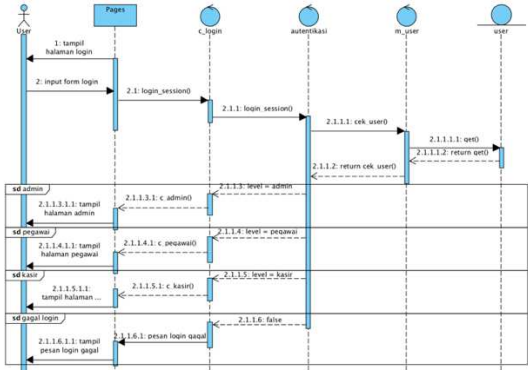
No	Aktor	Deskripsi
1	Admin	Admin atau selaku manajer memiliki hak akses tertinggi untuk mengelola data master yang ada dalam <i>database</i> , memantau produk dan jumlah stok produk, dan mengubah detail toko cabang sebagai data utama untuk alamat pemesanan kepada distributor.
2	Pegawai	Pegawai yang merupakan karyawan pendataan barang dapat melakukan penambahan atau perubahan suatu produk, menambahkan stok suatu produk, dan memesan produk kepada distributor. Pegawai juga dapat melihat stok barang masuk dan keluar dari toko dan memantau laporan pemasukan dan pengeluaran.
3	Kasir	Kasir hanya dapat melakukan transaksi menggunakan <i>barcode scanner</i> untuk memasukkan data barang yang dibeli oleh konsumen ke dalam sistem.

Setelah mendeskripsikan prosedur-prosedur pada sistem yang akan digunakan oleh toko Torani, pihak toko Torani Cirebon menginginkan kondisi dimana terdapat pembagian untuk masing-masing aktor yang terlibat dalam menggunakan sistem. Setiap pembagian untuk masing-masing aktor memiliki menu dan fungsi yang berbeda sesuai dengan kebutuhan masing-masing aktor seperti yang digambarkan dalam diagram *use case* seperti pada Gambar 2.



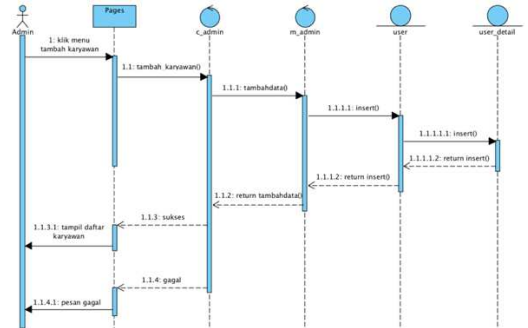
Gambar 2. Diagram *use case*

Dari diagram *use case* yang telah digambarkan di atas, proses kerja dari masing-masing aktor dapat dijelaskan melalui *sequence diagram*. Berikut adalah *sequence diagram login* yang ditunjukkan pada Gambar 3.



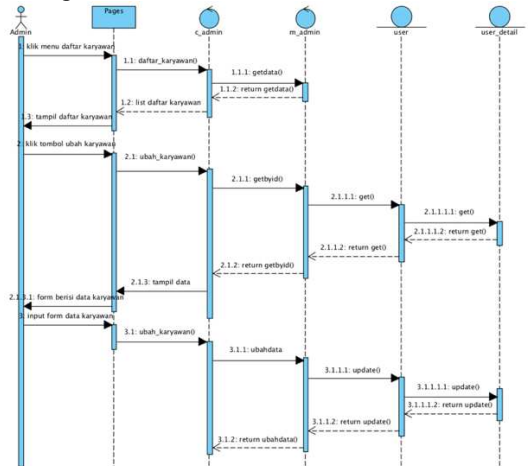
Gambar 3. *Sequence diagram login*

Berikut adalah *sequence diagram add karyawan* yang ditunjukkan pada Gambar 4.



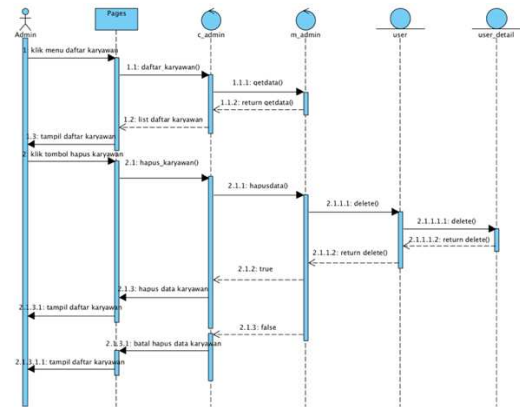
Gambar 4. *Sequence diagram add karyawan*

Berikut adalah *sequence diagram update karyawan* yang ditunjukkan pada Gambar 5.



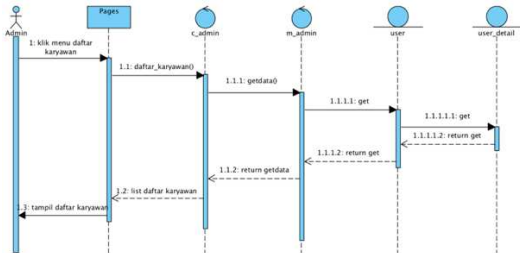
Gambar 5. *Sequence diagram update karyawan*

Berikut adalah *sequence diagram delete karyawan* yang ditunjukkan pada Gambar 6.



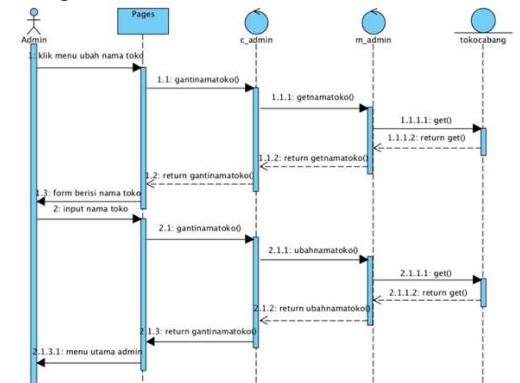
Gambar 6. *Sequence diagram delete karyawan*

Berikut adalah *sequence diagram melihat daftar karyawan* yang ditunjukkan pada Gambar 7.



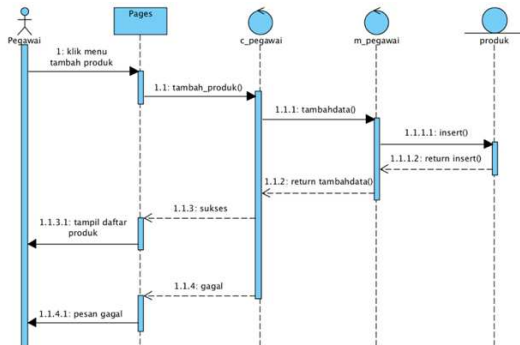
Gambar 7. *Sequence diagram lihat daftar karyawan*

Berikut adalah *sequence diagram mengubah nama toko* yang ditunjukkan pada Gambar 8.



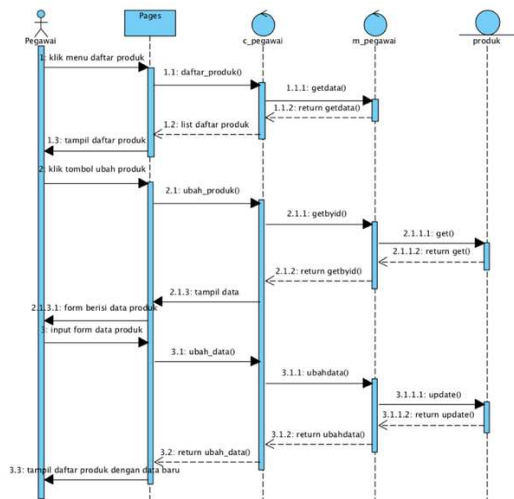
Gambar 8. *Sequence diagram ubah nama toko*

Berikut adalah *sequence diagram add produk* yang ditunjukkan pada Gambar 9.



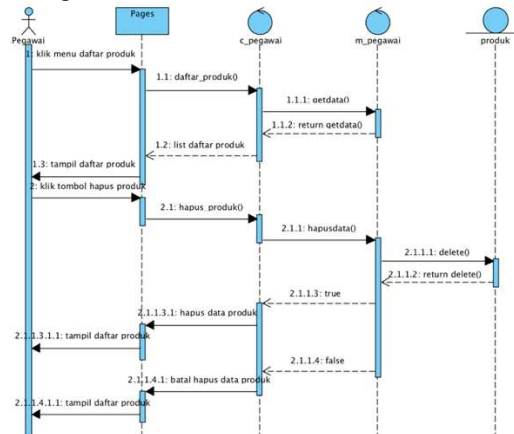
Gambar 9. *Sequence diagram add produk*

Berikut adalah *sequence diagram update produk* yang ditunjukkan pada Gambar 10.



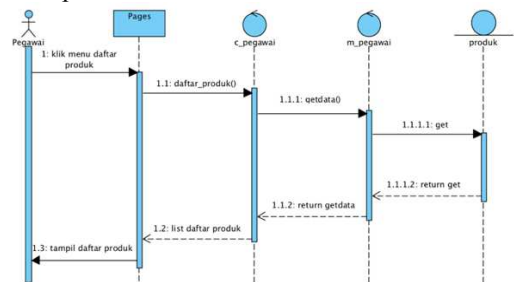
Gambar 10. Sequence diagram update produk

Berikut adalah sequence diagram delete produk yang ditunjukkan pada Gambar 11.



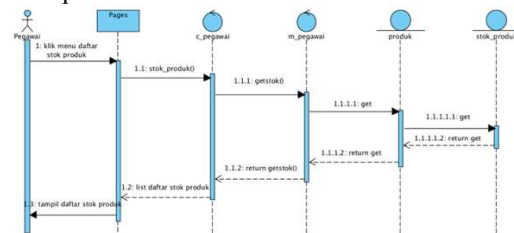
Gambar 11. Sequence diagram delete produk

Berikut adalah sequence diagram lihat daftar produk yang ditunjukkan pada Gambar 12.



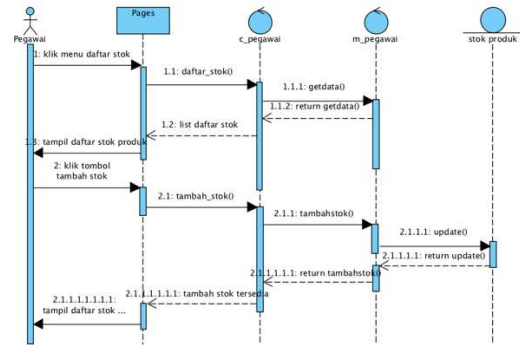
Gambar 12. Sequence diagram lihat daftar produk

Berikut adalah sequence diagram lihat stok produk yang ditunjukkan pada Gambar 13.



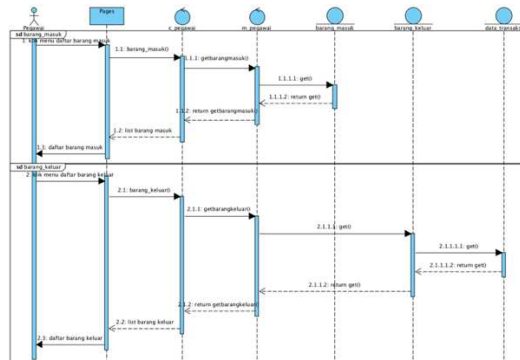
Gambar 13. Sequence diagram lihat stok produk

Berikut adalah sequence diagram add stok yang ditunjukkan pada gambar Gambar 14.



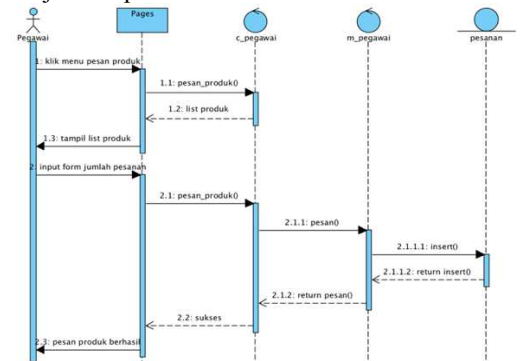
Gambar 14. Sequence diagram add stok

Berikut adalah sequence diagram melihat data barang masuk dan data transaksi (barang keluar) yang ditunjukkan pada Gambar 15.



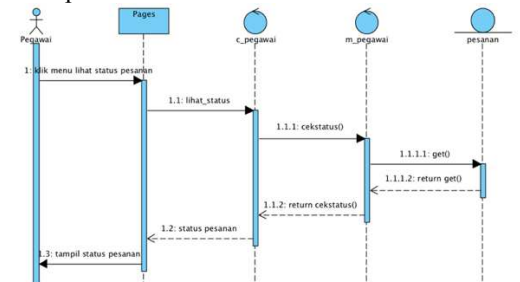
Gambar 15. Sequence diagram lihat data barang masuk dan data transaksi (barang keluar)

Berikut adalah sequence diagram pesan produk ke distributor yang ditunjukkan pada Gambar 16.



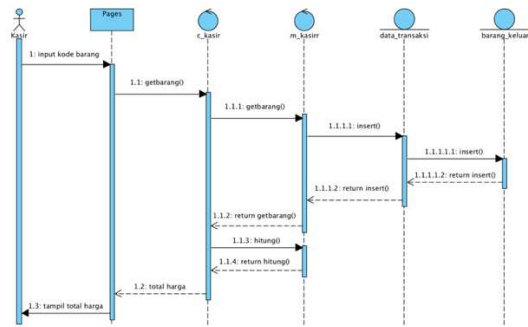
Gambar 16. Sequence diagram pesan produk ke distributor

Berikut adalah sequence diagram melihat status pesanan yang ditunjukkan pada Gambar 17.



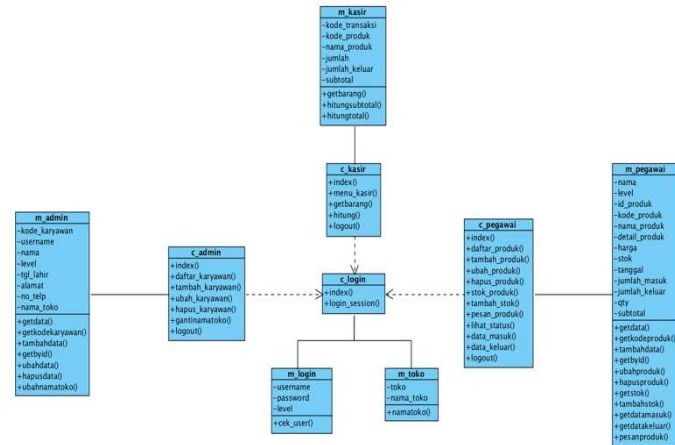
Gambar 17. Sequence diagram lihat status pesanan

Berikut adalah sequence diagram mendata pembelian yang ditunjukkan pada Gambar 18.



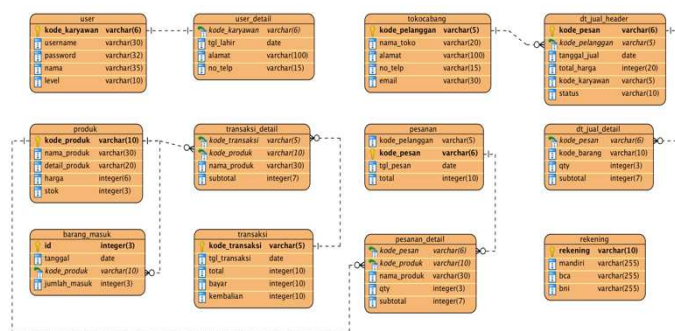
Gambar 18. Sequence diagram mendata barang pembelian

Class diagram menggambarkan struktur dan deskripsi *class*, *package*, dan objek beserta hubungan satu sama lain seperti *containment*, pewarisan, asosiasi, dan lain-lain. Class diagram berfungsi untuk menjelaskan tipe dari objek sistem dan hubungannya dengan objek yang lain. Objek adalah nilai tertentu dari setiap attribute kelas *entity*. Berikut kelas diagram dari sistem *point of sale* toko Torani yang akan dibangun seperti ditunjukkan pada Gambar 19.



Gambar 19. Class diagram

Setelah perancangan perangkat lunak selesai dibuat, langkah selanjutnya adalah merancang basis data yang akan digunakan untuk menyimpan seluruh data yang akan digunakan oleh sistem.



Gambar 20. Entity Relationship Diagram rancangan basis data

Dari *Entity Relationship Diagram* yang ditunjukkan pada Gambar 20 di atas, dapat diperoleh rancangan basis data dari *class-class* yang ada. Basis data yang digambarkan di atas telah melalui proses normalisasi sehingga dapat mengatasi terjadinya redundansi dan duplikasi data.^[5]

Pembuatan basis data dilakukan menggunakan bahasa SQL (*Structure Query Language*), dimana basis data atau DBMS (*Database Management System*) yang digunakan adalah MySQL. Berikut adalah struktur dari tabel-tabel yang diimplementasikan dalam basis data sesuai dengan rancangan sebelumnya.

Struktur dari tabel *user* dijelaskan pada Tabel 3.

Tabel 3. Struktur tabel *user*

Field	Jenis	Keterangan
kode_karyawan	varchar(6)	Primary key
username	varchar(30)	Username yang digunakan untuk login ke sistem
password	varchar(32)	-
nama	varchar(35)	-
level	enum('Admin', 'Pegawai', 'Kasir')	Level user

Struktur dari tabel *user_detail* dijelaskan pada Tabel 4.

Tabel 4. Struktur tabel *user_detail*

Field	Jenis	Keterangan
kode_karyawan	varchar(6)	Foreign key yang terhubung dengan tabel user
tgl_lahir	Date	Tanggal lahir menggunakan format tanggal
alamat	varchar(100)	-
no_telp	varchar(15)	-

Struktur dari tabel *tokocabang* dijelaskan pada Tabel 5.

Tabel 5. Struktur tabel *tokocabang*

Field	Jenis	Keterangan
kode_pelanggan	varchar(6)	Primary key
nama_toko	varchar(20)	-
alamat	varchar(100)	-
no_telp	varchar(15)	-
email	varchar(30)	-

Struktur dari tabel *produk* dijelaskan pada Tabel 6.

Tabel 6. Struktur tabel *produk*

Field	Jenis	Keterangan
kode_produk	varchar(10)	Primary key
nama_produk	varchar(30)	-
detail_produk	varchar(20)	-
harga	int(6)	-
stok	int(3)	-

Struktur dari tabel *barang_masuk* dijelaskan pada Tabel 7.

Tabel 7. Struktur tabel *barang masuk*

Field	Jenis	Keterangan
id	int(3)	Primary key
tanggal	date	-
kode_produk	varchar(10)	Foreign key yang menghubungkan dengan tabel produk
jumlah_masuk	int(3)	-

Struktur dari tabel *transaksi* dijelaskan pada Tabel 8.

Tabel 8. Struktur tabel *transaksi*

Field	Jenis	Keterangan
kode_transaksi	varchar(5)	Primary key
tgl_transaksi	date	-
total	int(10)	-
bayar	int(10)	-
kembalian	int(10)	-

Struktur dari tabel transaksi_detail dijelaskan pada Tabel 9.

Tabel 9. Struktur tabel transaksi_detail

Field	Jenis	Keterangan
kode_transaksi	varchar(5)	<i>Foreign key</i> yang menghubungkan dengan tabel transaksi
kode_produk	varchar(10)	<i>Foreign key</i> yang menghubungkan dengan tabel produk
nama_produk	varchar(30)	-
subtotal	int(7)	-

Struktur dari tabel pesanan dijelaskan pada Tabel 10.

Tabel 10. Struktur tabel pesanan

Field	Jenis	Keterangan
kode_pelanggan	varchar(5)	-
kode_pesanan	varchar(6)	<i>Primary key</i>
tgl_pesanan	date	-
total	int(10)	-

Struktur dari tabel pesanan_detail dijelaskan pada Tabel 11.

Tabel 11. Struktur tabel pesanan_detail

Field	Jenis	Keterangan
kode_pesanan	varchar(6)	<i>Foreign key</i> yang menghubungkan dengan tabel pesanan
kode_produk	varchar(10)	<i>Foreign key</i> yang menghubungkan dengan tabel produk
nama_produk	varchar(30)	-
qty	int(10)	-
subtotal	int(10)	-

Struktur dari tabel dt_jual_header dijelaskan pada Tabel 12.

Tabel 12. Struktur tabel dt_jual_header

Field	Jenis	Keterangan
kode_pesanan	varchar(6)	<i>Primary key</i>
kode_pelanggan	varchar(10)	<i>Foreign key</i> yang menghubungkan dengan tabel tokocabang
tanggal_jual	date	-
total_harga	int(20)	-
kode_karyawan	varchar(5)	Merupakan <i>field</i> yang hanya dapat diisi oleh pihak distributor
status	enum('PENDING', 'DIPROSES', 'DITERIMA', 'BELI DITEMPAT')	Status dari barang pesanan

Struktur dari tabel dt_jual_detail dijelaskan pada Tabel 13.

Tabel 13. Struktur tabel dt_jual_detail

Field	Jenis	Keterangan
kode_pesanan	varchar(6)	<i>Primary key</i>
kode_barang	varchar(10)	-
qty	int(3)	-
subtotal	int(7)	-

Struktur dari tabel dt_rekening dijelaskan pada Tabel 14.

Tabel 14. Struktur tabel dt_rekening

Field	Jenis	Keterangan
rekening	varchar(10)	<i>Primary key</i>
mandiri	text	-
bca	text	-
bni	text	-

Setelah *database* selesai dibentuk, langkah selanjutnya yaitu implementasi program. Pada implementasi program, dibagi menurut kategori dan fungsi sesuai rancangan sebelumnya. Dalam pengimplementasian desain *user interface* serta pengambilan data dalam *database*, digunakan *model* dan *controller* yang menampung variabel maupun *method* yang bersifat *public* sehingga dapat digunakan di seluruh *class* dan kemudian hasilnya ditampilkan oleh *module view*.^[3]

Langkah pertama yang dilakukan adalah membuat autentikasi login. Autentikasi dibuat untuk pengenalan *user* dengan aplikasi, apabila *user* dikenali, maka *user* tersebut dapat mengakses aplikasi sesuai dengan *rol*nya.

Pustaka autentikasi menggunakan sebuah tabel di dalam *database*. Tabel yang digunakan adalah tabel *user*. Pustaka ini juga menggunakan sebuah model untuk melakukan interaksi data dengan *database*. Berikut adalah potongan kode dari fungsi login() yang berfungsi untuk melakukan autentikasi login pada aplikasi ditunjukkan pada Gambar 33.

```
public function login_session() {
    $data['namatoko'] = $this->m_toko->namatoko();

    $this->form_validation->set_message('required', '%s harus diisi');

    $this->form_validation->set_rules('username', 'Username', 'trim|required');
    $this->form_validation->set_rules('password', 'Password', 'trim|required');

    if($this->form_validation->run() == FALSE) {
        $this->load->view('login', $data);
    } else {
        $hasil = $this->m_login->cek_user($data);
        if ($hasil->num_rows() == 1) {
            foreach ($hasil->result() as $sess) {
                $sess_data['uid'] = $sess->uid;
                $sess_data['username'] = $sess->username;
                $sess_data['nama'] = $sess->nama;
                $sess_data['level'] = $sess->level;
                $this->session->set_userdata($sess_data);
            }
            if ($this->session->userdata('level') == 'Admin') {
                redirect('admin/c_admin');
            }
            elseif ($this->session->userdata('level') == 'Pegawai') {
                redirect('pegawai/c_pegawai');
            }
            elseif ($this->session->userdata('level') == 'Kasir') {
                redirect('kasir/c_kasir');
            }
            else {
                echo "<script>alert('Login gagal: Username atau password salah!');history.go(-1);</script>";
            }
        }
    }
}
```

Gambar 21. Fungsi login()

Berikut adalah potongan kode dari fungsi cek_user() yang berfungsi untuk mengecek *username* dan *password* di dalam *database* ditunjukkan pada Gambar 34.

```
public function cek_user() {
    $data = array(
        'username' => $this->input->post('username', TRUE),
        'password' => md5($this->input->post('password', TRUE)),
    );

    $query = $this->db->get_where('user', $data);
    return $query;
}
```

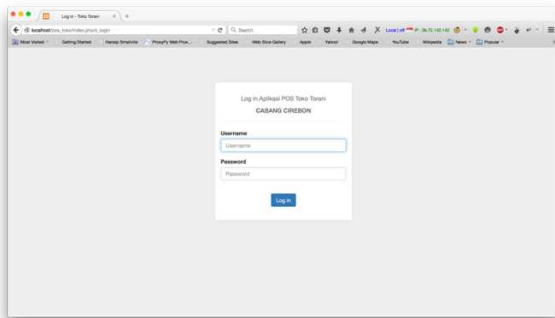
Gambar 22. Fungsi cek_user()

Berikut adalah potongan kode dari fungsi logout() yang berfungsi untuk dan mengakhiri *session login* dan keluar dari aplikasi ditunjukkan pada Gambar 35.

```
public function logout() {
    $this->session->unset_userdata('username');
    $this->session->unset_userdata('level');
    session_destroy();
    redirect('c_login');
}
```

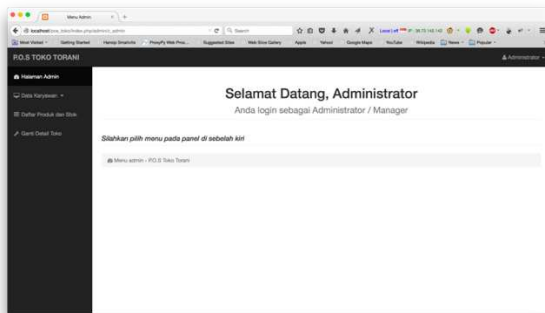
Gambar 23. Fungsi logout()

Selanjutnya adalah membuat halaman *login* yang merupakan halaman awal untuk masuk ke dalam aplikasi seperti yang digambarkan pada Gambar 36.



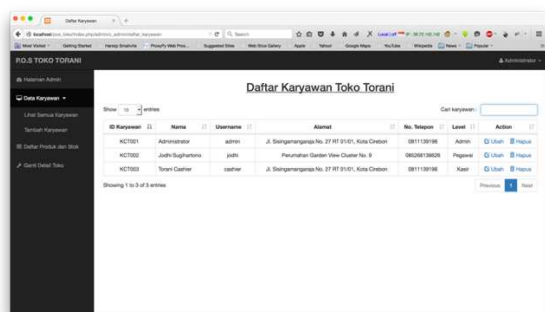
Gambar 24. Tampilan halaman login

Selanjutnya membuat halaman utama yang merupakan halaman yang pertama ditampilkan setelah *login* seperti yang digambarkan pada Gambar 37.



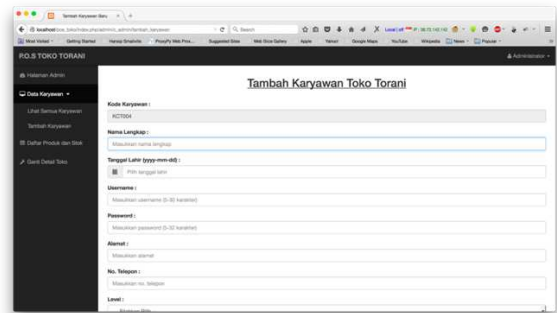
Gambar 25. Tampilan halaman utama

Selanjutnya membuat halaman daftar karyawan yang merupakan halaman yang berfungsi untuk menampilkan seluruh data karyawan seperti yang digambarkan pada Gambar 38.



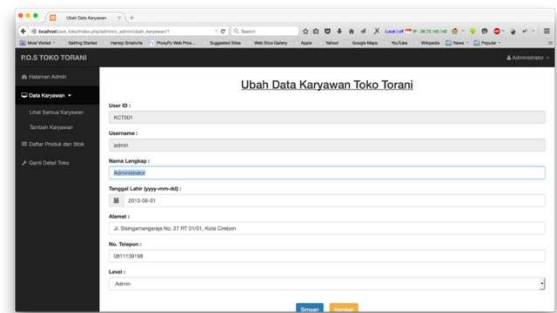
Gambar 26. Tampilan halaman daftar karyawan

Selanjutnya membuat halaman tambah karyawan yang merupakan halaman untuk menambah karyawan baru seperti yang digambarkan pada Gambar 39.



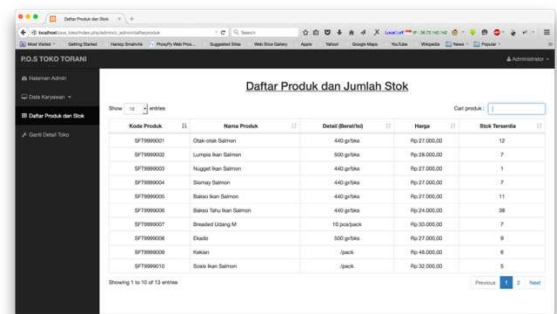
Gambar 27. Halaman tambah karyawan baru

Selanjutnya adalah membuat halaman ubah data karyawan yang merupakan halaman yang berfungsi untuk menampilkan dan mengubah data karyawan seperti yang digambarkan pada Gambar 40.



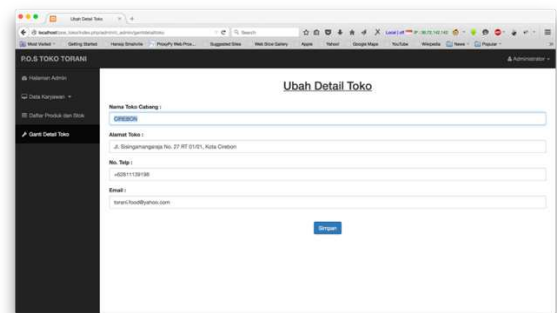
Gambar 28. Halaman ubah data karyawan

Selanjutnya adalah membuat halaman daftar produk dan stok (admin) yang merupakan halaman yang berfungsi untuk menampilkan daftar produk dan stok seperti yang digambarkan pada Gambar 41.



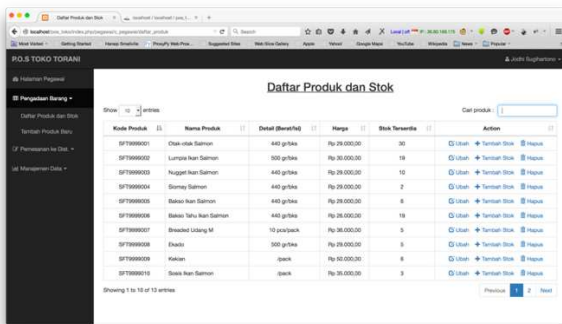
Gambar 29. Halaman daftar produk dan stok

Selanjutnya adalah membuat halaman ubah detail toko yang merupakan halaman untuk mengubah detail toko yang akan digunakan sebagai alamat pengiriman pesanan seperti yang digambarkan pada Gambar 42.



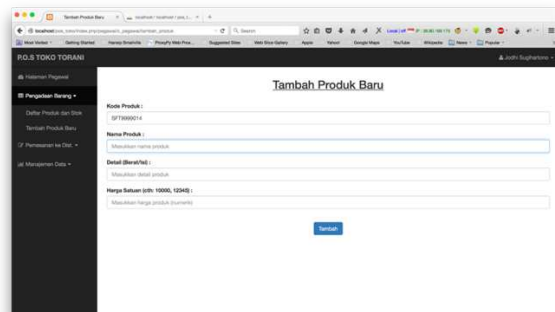
Gambar 30. Halaman ubah detail toko

Selanjutnya adalah membuat halaman daftar produk dan stok (pegawai) yang merupakan Halaman yang berfungsi untuk menampilkan daftar produk dan stok seperti yang digambarkan pada Gambar 43.



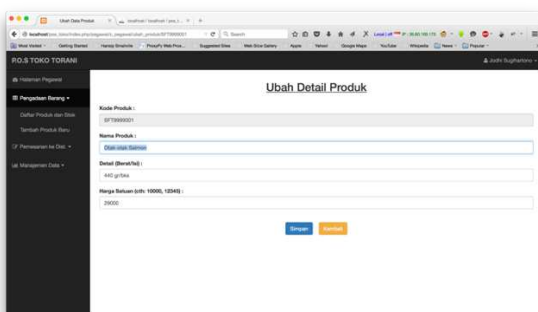
Gambar 31. Halaman daftar produk dan stok

Selanjutnya adalah membuat halaman tambah produk baru yang merupakan halaman untuk menambah produk baru seperti yang digambarkan pada Gambar 44.



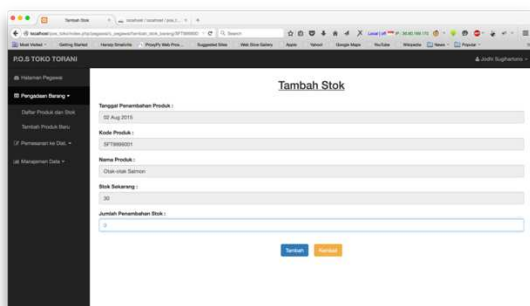
Gambar 32. Halaman tambah produk baru

Selanjutnya adalah Membuat halaman ubah detail produk yang merupakan halaman yang berfungsi untuk menampilkan dan mengubah detail produk seperti yang digambarkan pada Gambar 45.



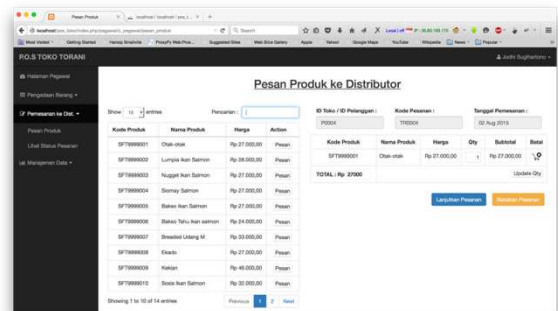
Gambar 33. Halaman ubah detail produk

Selanjutnya adalah membuat halaman tambah stok yang merupakan halaman untuk menambah jumlah stok produk seperti yang digambarkan pada Gambar 46.



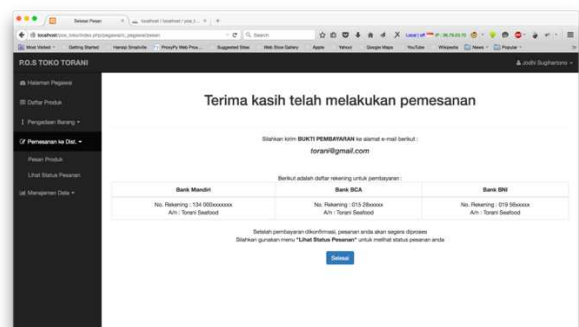
Gambar 34. Halaman tambah stok

Selanjutnya adalah membuat halaman pesan produk ke distributor yang berfungsi untuk melakukan pemesanan produk kepada distributor dan terhubung langsung dengan database distributor seperti yang digambarkan pada Gambar 47.



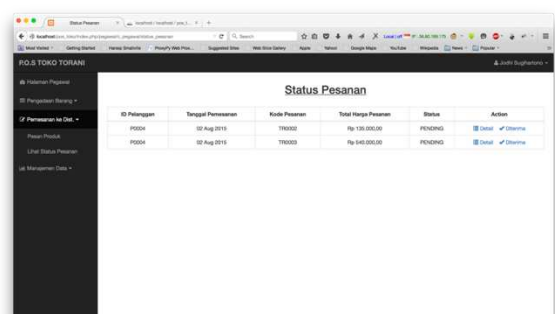
Gambar 35. Halaman pesan produk

Selanjutnya adalah membuat halaman selesai pesan yang merupakan halaman yang berisi daftar rekening untuk pembayaran yang tersimpan di dalam database distributor seperti yang digambarkan pada Gambar 48.



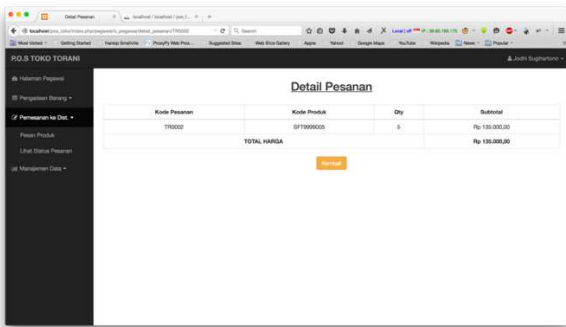
Gambar 36. Halaman selesai pesan

Selanjutnya adalah membuat halaman lihat status pesanan yang merupakan halaman untuk melihat status pesanan yang sedang berjalan seperti yang digambarkan pada Gambar 49.



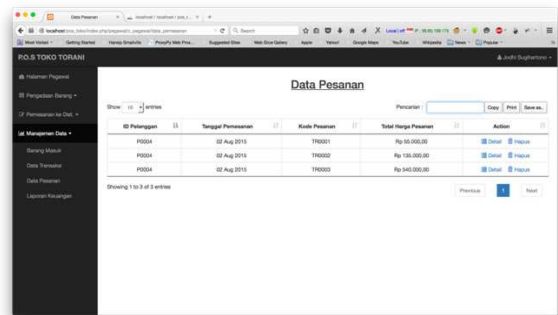
Gambar 37. Halaman lihat status pesanan

Selanjutnya adalah membuat halaman detail pesanan yang merupakan halaman untuk melihat detail pesanan yang sedang berlangsung seperti yang digambarkan pada Gambar 50.



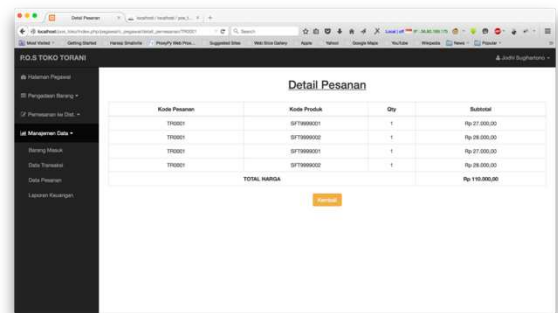
Gambar 38. Halaman detail pesanan

Selanjutnya adalah Membuat halaman data pesanan yang merupakan halaman untuk melihat seluruh data pesanan yang pernah dilakukan seperti yang digambarkan pada Gambar 54.



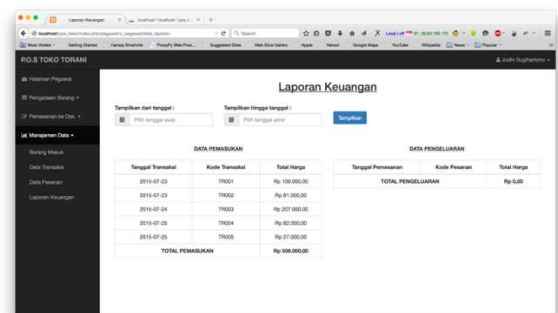
Gambar 42. Halaman data pesanan

Selanjutnya adalah membuat halaman detail pesanan yang merupakan halaman untuk melihat detail pesanan tertentu seperti yang digambarkan pada Gambar 55.



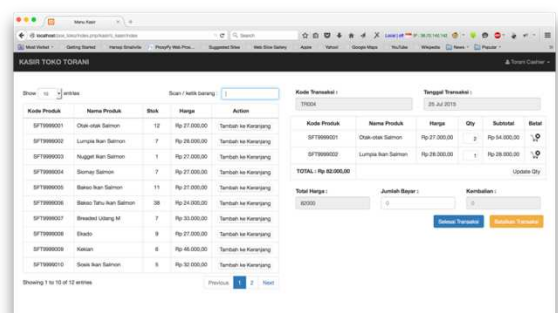
Gambar 43. Halaman detail pesanan

Selanjutnya adalah membuat halaman laporan keuangan yang merupakan alaman untuk melihat jumlah pemasukan dan pengeluaran berdasarkan tanggal atau bulan tertentu seperti yang digambarkan pada Gambar 56.



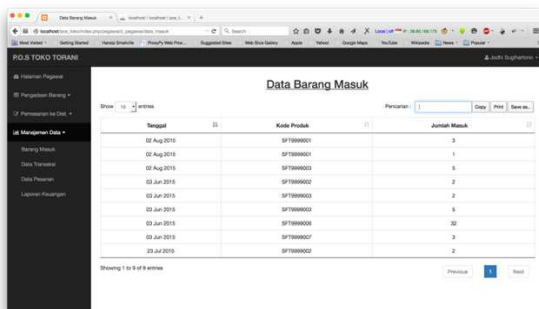
Gambar 44. Halaman laporan keuangan

Selanjutnya adalah membuat halaman menu kasir yang merupakan halaman utama kasir untuk melakukan transaksi jual beli seperti yang digambarkan pada Gambar 57.



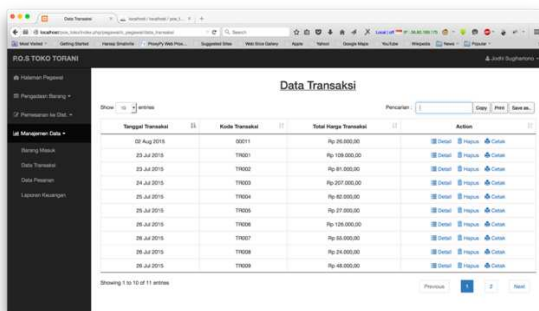
Gambar 45. Halaman menu kasir

Selanjutnya adalah membuat halaman data barang masuk yang merupakan halaman untuk melihat daftar dan jumlah barang masuk seperti yang digambarkan pada Gambar 51.



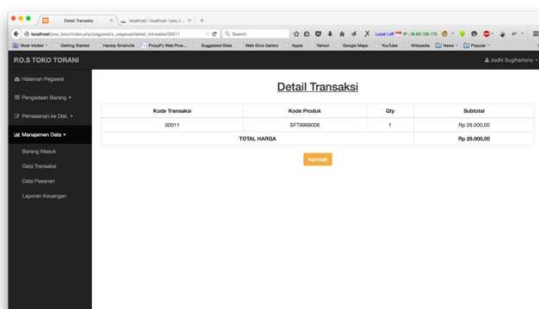
Gambar 39. Halaman data barang masuk

Selanjutnya adalah membuat halaman data transaksi yang merupakan halaman untuk melihat data transaksi yang pernah dilakukan oleh kasir seperti yang digambarkan pada Gambar 52.



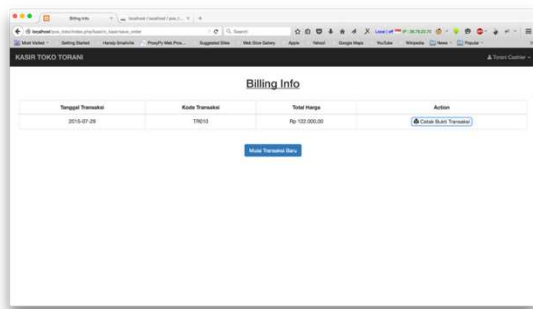
Gambar 40. Halaman data transaksi

Selanjutnya adalah membuat halaman detail transaksi yang merupakan halaman untuk melihat detail transaksi tertentu seperti yang digambarkan pada Gambar 53.



Gambar 41. Halaman detail transaksi

Selanjutnya adalah membuat halaman *billing info* yang merupakan halaman untuk menyimpan data transaksi dan untuk mencetak bukti transaksi seperti yang digambarkan pada Gambar 58.



Gambar 46. Halaman *billing info*

Selanjutnya adalah membuat halaman cetak bukti transaksi yang merupakan halaman berbentuk *print preview* dari bukti transaksi yang siap cetak seperti yang digambarkan pada Gambar 59.



Gambar 47. Halaman cetak bukti transaksi

Proses pengujian dilakukan dengan menggunakan metode *black box*. Metode *black box* menguji aplikasi dengan berfokus pada spesifikasi fungsional dari perangkat lunak. Bentuk pengujian dilakukan dengan memeriksa kondisi input dan melakukan pengetesan pada spesifikasi fungsional dengan langsung menggunakan aplikasi.

Tabel 15. Tabel pengujian

Nama Pengujian	Bentuk Pengujian	Hasil Yang Diharapkan	Hasil Pengujian
Pengujian login sukses sebagai admin	Mengisi <i>field username</i> dan <i>password</i> yang sudah terdaftar sebagai admin	Masuk ke menu utama admin dengan tampilan menu admin	Berhasil
Pengujian login sukses sebagai pegawai	Mengisi <i>field username</i> dan <i>password</i> yang sudah terdaftar sebagai pegawai	Masuk ke menu utama admin dengan tampilan menu pegawai	Berhasil
Pengujian login sukses sebagai kasir	Mengisi <i>field username</i> dan <i>password</i> yang sudah terdaftar sebagai kasir	Masuk ke menu utama admin dengan tampilan menu kasir	Berhasil
Pengujian login gagal	Mengisi <i>field</i> dan <i>username</i> dan <i>password</i> yang tidak terdaftar	Menampilkan pemberitahuan login gagal	Berhasil

Pengujian login salah	Mengosongkan <i>field username</i> atau <i>password</i>	Menampilkan pemberitahuan bahwa pengisian <i>field</i> tidak memenuhi ketentuan	Berhasil
Pengujian logout aplikasi	Memilih menu dropdown pada nama <i>user</i> di pojok kanan atas dan pilih logout	Keluar dari aplikasi dan kembali ke menu login	Berhasil
Pengujian tampilan halaman admin	Memilih menu halaman admin	Menampilkan halaman utama admin	Berhasil
Pengujian membuka tampilan daftar karyawan	Memilih menu dropdown data karyawan dan pilih menu lihat semua karyawan	Menampilkan halaman yang berisi tabel daftar karyawan	Berhasil
Pengujian membuka tampilan tambah karyawan	Memilih menu dropdown data karyawan dan pilih menu tambah karyawan	Menampilkan halaman yang berisi form input untuk mengisi data karyawan baru	Berhasil
Pengujian membuka tampilan daftar produk dan stok	Memilih menu daftar produk dan stok	Menampilkan halaman yang berisi tabel daftar produk dan stok yang tersedia	Berhasil
Pengujian membuka tampilan ubah detail toko	Memilih menu ganti detail toko	Menampilkan halaman yang berisi form berisi detail toko	Berhasil
Pengujian tampilan halaman pegawai	Memilih menu halaman pegawai	Menampilkan halaman utama pegawai	Berhasil
Pengujian membuka tampilan daftar produk	Memilih menu daftar produk	Menampilkan halaman yang berisi tabel daftar produk dan detailnya	Berhasil
Pengujian membuka tampilan tambah produk baru	Memilih menu dropdown pengadaan barang dan pilih menu tambah produk baru	Menampilkan halaman yang berisi form input untuk menambah produk baru	Berhasil
Pengujian membuka tampilan pesan produk ke distributor	Memilih menu dropdown pemesanan ke distributor dan pilih menu pesan produk	Menampilkan halaman yang berisi tabel daftar produk yang disediakan distributor beserta detail pesanan	Berhasil
Pengujian membuka tampilan lihat status pesanan	Memilih menu dropdown pemesanan ke distributor dan pilih menu lihat status pesanan	Menampilkan halaman yang berisi tabel daftar pesanan dan status pesanan	Berhasil
Pengujian membuka tampilan barang masuk	Memilih menu dropdown manajemen data dan pilih menu barang masuk	Menampilkan halaman yang berisi tabel barang masuk	Berhasil

Pengujian membuka tampilan data transaksi	Memilih menu <i>dropdown</i> manajemen data dan pilih menu data transaksi	Menampilkan halaman yang berisi tabel data transaksi	Berhasil
Pengujian membuka tampilan data pesanan	Memilih menu <i>dropdown</i> manajemen data dan pilih menu data pesanan	Menampilkan halaman yang berisi tabel data pesanan	Berhasil
Pengujian membuka tampilan laporan keuangan	Memilih menu <i>dropdown</i> manajemen data dan pilih menu laporan keuangan	Menampilkan halaman yang berisi tabel total pemasukan dan pengeluaran	Berhasil
Pengujian membuka tampilan menu kasir	<i>Login</i> menggunakan akun dengan <i>level</i> kasir	Menampilkan halaman yang berisi tabel daftar produk beserta detail transaksi	Berhasil
Pengujian tombol tambah ke keranjang	Menekan tombol tambah ke keranjang	Menampilkan data produk yang dipilih beserta <i>quantity</i> dan harga produk yang dipilih	Berhasil
Pengujian tombol <i>update qty</i>	Menekan tombol <i>update qty</i> pada pojok kanan bawah tabel transaksi	Menjumlahkan harga produk yang dipilih sesuai dengan <i>quantity</i> yang dipesan	Berhasil
Pengujian tombol hapus	Menekan tombol hapus pada bagian kanan tabel transaksi	Menghapus data transaksi dari daftar tabel transaksi	Berhasil
Pengujian tombol batalkan transaksi	Menekan tombol batalkan transaksi	Menghapus semua data transaksi yang ditampilkan	Berhasil
Pengujian sukses <i>submit form input</i>	Mengisi <i>form input</i> dengan lengkap dan sesuai format	Menyimpan data <i>input</i>	Berhasil
Pengujian gagal <i>submit form input</i>	Mengosongkan sebagian atau semua <i>form input</i>	Menampilkan pemberitahuan bahwa <i>field</i> tidak boleh kosong	Berhasil
Pengujian kesalahan pengisian <i>form input</i>	Mengisi <i>form input</i> dengan format yang tidak sesuai	Menampilkan pemberitahuan bahwa pengisian <i>field</i> tidak memenuhi ketentuan	Berhasil
Pengujian tombol kembali	Menekan tombol kembali	Kembali ke halaman sebelumnya	Berhasil
Pengujian tombol cetak bukti transaksi	Menekan tombol cetak bukti transaksi	Menampilkan <i>print preview</i> dari transaksi terakhir dan siap cetak	Berhasil

IV. KESIMPULAN DAN SARAN

Dari hasil pengujian dan analisis aplikasi *point of sale* toko cabang perusahaan Torani, maka dapat disimpulkan hal-hal sebagai berikut :

1. Berdasarkan pengujian, *user* admin / manajer dapat melakukan aksi pengelolaan karyawan toko, seperti lihat, tambah, ubah data, dan hapus data karyawan. Admin juga dapat melihat daftar produk dan jumlah stok yang tersedia untuk memantau persediaan barang di toko.
2. Berdasarkan pengujian, *user* pegawai dapat melakukan aksi pengelolaan barang di toko, seperti lihat, tambah barang baru, tambah stok, ubah data, dan hapus data barang. Pegawai juga dapat melakukan pemesanan barang kepada distributor dengan menggunakan menu pesan yang terhubung dengan internet sehingga dapat terkoneksi dengan *database* distributor.
3. Berdasarkan pengujian, *user* kasir dapat melakukan aksi transaksi jual beli dengan baik, seperti memilih barang, mendata barang yang dibeli, menghitung jumlah harga secara otomatis, dan mencetak bukti transaksi.

Berdasarkan pengujian terhadap aplikasi *point of sale* toko cabang perusahaan Torani yang telah dibuat, maka dapat diberikan beberapa saran sebagai berikut :

1. Aplikasi *point of sale* toko cabang perusahaan Torani ini dapat dikembangkan lebih lanjut dengan menambahkan fitur seperti logika *fuzzy* untuk mengecek barang apa yang laris dan tidak serta menghitung berapa banyak jumlah barang yang harus dipesan ke distributor, dan kasir yang *full* menggunakan sistem *barcode*.
2. Tampilan hasil cetak bukti transaksi dapat dikembangkan dan disesuaikan dengan bentuk *printer* untuk kasir yang menggunakan kertas gulung berukuran kecil.
3. Aplikasi *point of sale* toko cabang dapat diintegrasikan ke dalam sebuah *web server* bersama aplikasi distributor sehingga data dapat menjadi lebih terpusat pada satu *database* saja.

DAFTAR PUSTAKA

- [1] Ladjamudin, Al-Bahra Bin, *Analisis dan Desain Sistem Informasi*, Penerbit Graha Ilmu, Yogyakarta, 2005.
- [2] Sholih, *Pemodelan Sistem Informasi Berorientasi Obyek dengan UML*, Penerbit Graha Ilmu, Yogyakarta, 2006.
- [3] Andi, *Mudah dan Cepat Membuat Website dengan Codeigniter*, Penerbit Andi, Yogyakarta, 2011.
- [4] SE, Tommy, *Membuat Sistem Aplikasi Minimarket (Point of Sales) dengan VB 6 & MYSQL 5*, Penerbit Elex Media Komputindo, Jakarta, 2008.
- [5] Solichin, Achmad, *MySQL dari Pemula Hingga Mahir*, Universitas Budi Luhur, Jakarta, 2010.