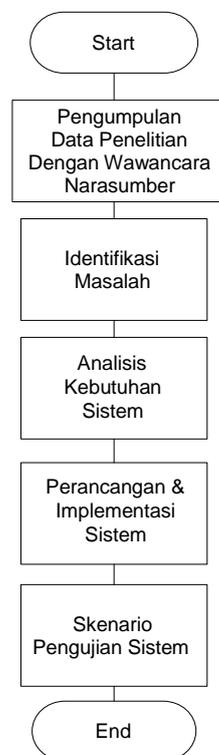


# BAB III

## METODOLOGI PENELITIAN

Proses penelitian rancang bangun aplikasi manajemen proyek teknologi informasi terbagi menjadi 5 tahapan yaitu : 1) Pengumpulan Data Penelitian Dengan Wawancara Narasumber. 2) Identifikasi Masalah. 3) Analisis Kebutuhan Sistem. 4) Perancangan & Implementasi Sistem. 5) Skenario Pengujian Sistem. Tahapan-tahapan penelitian diatas dapat dilihat pada Gambar 3.1.



Gambar 3. 1 Tahapan penelitian rancang bangun aplikasi

Berdasarkan Gambar 3. 1 dapat dijelaskan mengenai tahapan penelitian berawal dari pengumpulan data penelitian dengan melakukan wawancara pencarian alur bisnis perusahaan dalam rancang bangun aplikasi melalui Penanggung Jawab Operasionalnya yaitu Ibu Fitriati Prihatiningsih. Berdasarkan

informasi yang diperoleh dari beliau kemudian dilakukan pencarian teknologi terbaru yang relevan untuk digunakan, relevan dalam arti mudah untuk dipelajari dan dipahami oleh user yang masih awam dalam mengoperasikan sistem. Kemudian pada tahap kedua dilakukan penelusuran masalah yang terjadi dalam *monitoring* alur stok dalam tubuh perusahaan. Dari penelusuran masalah terkait kemudian lanjut ke tahap ketiga dan keempat yaitu merangkum setiap kebutuhan berdasarkan masalah yang ada serta merancang sistem yang akan dibuat dan diimplementasikan sesuai dengan kebutuhan bisnis perusahaan. Sedangkan langkah terakhir akan dilakukan skenario pengujian sistem untuk mengetahui apakah sistem yang dibuat dapat berjalan dengan baik sesuai kegunaannya.

### **3.1 Pengumpulan Data Terkait Penelitian**

Pada tahap ini dilakukan penelitian dari pertengahan Februari 2020 hingga aplikasi selesai dibuat. Pengumpulan data dilakukan dibawah naungan Laboratorium PPS (Pemrograman, Pengembangan, dan Strategi IT), Program Studi Teknik Informatika, Fakultas Ilmu Komputer Universitas Pembangunan Nasional “Veteran” Jawa Timur yang beralamat di Jl. Gunung Anyar 1, Surabaya dan perusahaan Showroom Permata Motor Semolowaru yang beralamat di Jl.Semolowaru Tengah 1 no.1A Surabaya.

Teknik pengumpulan data yang digunakan dalam penelitian ini yaitu studi literatur. Studi literatur yang dimaksud adalah belajar melalui jurnal penelitian sebelumnya di internet, mengikuti tutorial dasar rancang bangun aplikasi serta komponen UI (*User Interface*) menggunakan framework di

Youtube (Founder : Jawed, Steve and Hurley, 2005) dan mengikuti pembelajaran melalui situs Udemy (Founder : Biyani, Bali and Caglar, 2009).

### **3. 2        Identifikasi Masalah**

Manajemen pengelolaan stok mempunyai peranan penting dalam memantau keakuratan antara barang yang keluar (terbeli), barang yang siap jual di gerai atau toko, dengan barang yang baru masuk (datang) di gudang. Peranan penting manajemen pengelolaan stok dalam perusahaan adalah bagaimana dapat mengelola sumber daya yang ada untuk menghasilkan keuntungan semaksimal mungkin dengan meminimalisir resiko kerugian yang dapat menerpa suatu lini usaha. Dalam bidang Penjualan Motor Bekas peran manajemen pengelolaan stok terkait dengan proses awal dari pengadaan unit kendaraan yang baru diperoleh dan masuk gudang sampai pada proses penjualan yang melibatkan merek, tipe, tahun, dan bahkan warna kendaraan serupa yang hanya dibedakan oleh nomor polisi tiap kendaraan.

Perusahaan Permata Motor telah menjual ratusan kendaraan roda dua sejak berdiri pada 2018 mulai dari jenis motor bertransmisi matic hingga motor sport. Setiap proses pengadaan hingga penjualan kendaraan roda dua yang terjadi sejak perusahaan pertama berdiri masih dilakukan secara manual. Oleh karena dimonitor secara manual, maka kebutuhan akan dibuatnya aplikasi manajemen pengelolaan stok dalam perusahaan mejadi hal yang cukup mendesak bagi perusahaan dalam jangka panjang dengan semakin banyaknya permintaan daya beli masyarakat yang akan menjadi tujuan perusahaan untuk menyediakan semakin banyak pilihan kendaraan roda dua berkualitas dan pengelolaan sumber

daya manusia di dalam perusahaan serta menjadi acuan untuk memperkirakan kapan suatu unit sudah terlalu lama disimpan berada dalam toko/showroom.

Stok kendaraan perlu dimonitor dengan sistem yang dapat mencatat perubahan stok dengan tipe serupa namun dengan nomor polisi berbeda ketika tahap pengadaan hingga mencatat data penjualan. Tahap *pengadaan* merupakan tahap dimana stok kendaraan baru saja bertambah dan posisi unit kendaraan disimpan di gudang. Sedangkan pada tahap penjualan diperlukan untuk mengamati unit kendaraan yang berkurang dan sebagai bahan evaluasi ketika akan melakukan pengadaan unit kembali.

### **3.3 Analisis Kebutuhan**

Pada tahapan ini terbagi menjadi 2 bagian yaitu 1) Analisis Sistem dan 2) Analisis *User Requirement*. Pada penelitian ini analisis sistem bertujuan untuk mengumpulkan informasi yang dibutuhkan berdasarkan identifikasi masalah. Kemudian setelah sistem dianalisa dilakukan analisis *user requirement* untuk menjabarkan kebutuhan perusahaan secara detail.

#### **3.3.1 Analisis Sistem**

Berdasarkan identifikasi masalah yang sudah dijelaskan diatas maka sistem yang dibutuhkan adalah sebagai berikut :

1. Informasi unit baru datang di gudang setelah proses pengadaan unit yang terekam kedalam sistem.
2. Informasi unit yang sedang dilakukan proses restorasi.

3. Informasi yang siap jual dan sudah berada di dalam gerai atau showroom
4. Informasi ketika unit kendaraan terbeli sehingga stok terkini terekam kedalam sistem

### **3.3.2 Analisis *User Requirement***

Berdasarkan identifikasi masalah dan analisa sistem diatas maka dapat disimpulkan kebutuhan dasar aplikasi adalah sebagai berikut :

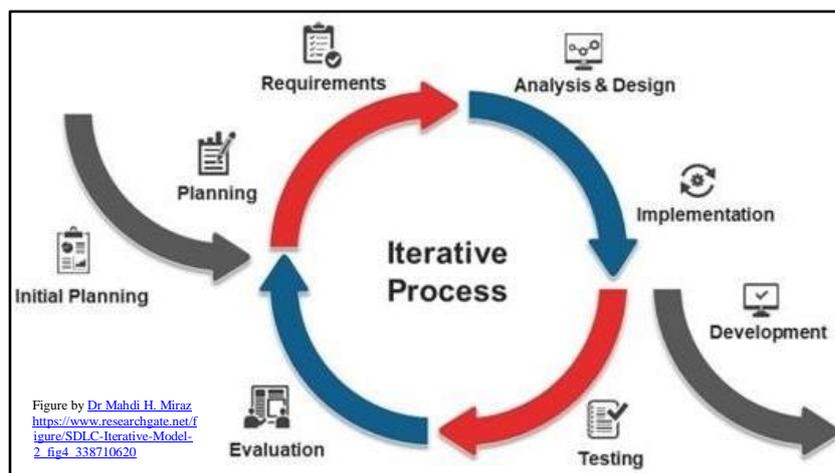
1. Sistem terdiri dari 3 jenis jabatan yaitu Admin Owner, dan Penjaga Toko.
2. Admin Owner dapat melihat, mengaktifkan, menonaktifkan, menghapus Penjaga Toko.
3. Admin Owner dapat memantau pekerjaan inputan data Penjaga Toko dalam rentang waktu tertentu.
4. Admin Owner dapat melakukan CRUD (Create, Read, Update, Delete) pada daftar motor yang ada
5. Admin Owner dapat melakukan CRUD (Create, Read, Update, Delete) pada daftar pegawai yang ada
6. Pegawai Toko dapat melakukan input data motor yang dicari calon pembeli yang telah melakukan walk-in kedalam gerai showroom sehingga menghasilkan Trend Motor paling dicari

### **3.4 Perancangan Sistem**

Pada tahap ini akan dijelaskan metodologi SDLC yang diterapkan selama penelitian dilakukan. Tahapan ini pula memiliki bertujuan melakukan perencanaan UML yang merupakan metodologi dalam mengembangkan sistem berorientasi objek dan juga merupakan alat untuk mendukung pengembangan

sistem (Hendini, 2016). Selain itu dilakukan pula perencanaan model sistem yang akan dibuat dengan output berupa desain akhir basis data berbentuk PDM, desain antarmuka, dan penerapan perhitungan dalam sistem. Tahapan yang dilakukan meliputi : 1) Arsitektur Framework CodeIgniter. 2) Perancangan Use Case Diagram. 3) Perancangan Activity Diagram. 4) Perancangan Class Diagram. 5) Perancangan Sequence Diagram. 6) Perancangan CDM (*Conceptual Data Model*). 7) Perancangan PDM (*Physical Data Model*). 8) Desain Antarmuka.

### 3.4.1 Metodologi SDLC Iterative



Gambar 3. 2 Pemodelan Metodologi SDLC *Iterative*

Pada penelitian ini metodologi SDLC yang digunakan dalam pengembangan aplikasi merupakan metodologi SDLC *Iterative*. Jenis metodologi SDLC *Iterative* dipilih untuk digunakan selama pengembangan aplikasi dikarenakan metodologi ini menyediakan kemudahan fleksibilitas yang tinggi namun juga menyediakan pilihan integrasi data yang cukup tinggi. Serta komunitas pengguna yang turut berperan aktif selama pengembangan aplikasi (Rizky Parlika, Moch. Fikri Hidayat, 2018). Mengingat pada penelitian ini

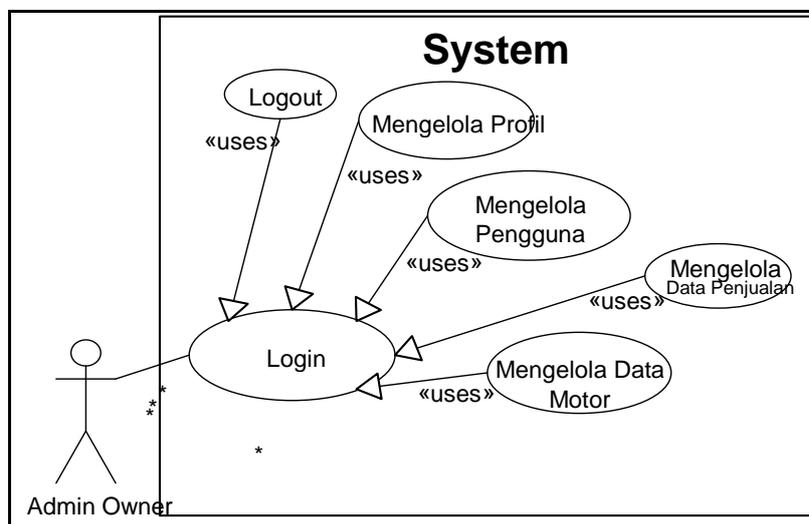
didasarkan pada kebutuhan dari komunitas pengguna Permata Motor Semolowaru, maka metodologi ini dirasa sesuai untuk digunakan.

### 3. 4. 2 Perancangan *Use Case Diagram*

*Use Case Diagram* merupakan model fungsional sebuah sistem yang menggunakan aktor dan *use case*. *Use Case Diagram* dibuat menggunakan *software* ciptaan Microsoft yaitu Microsoft Visio 2007. Dalam analisis *user requirement* yang sudah dijelaskan sebelumnya terdapat 2 aktor yang terlibat dalam sistem yaitu Admin Owner dan Pegawai Toko sebagai berikut.

#### 1. *Use Case Diagram* Admin Owner

*Use case diagram* Admin Owner merupakan hal-hal yang dapat dilakukan pengguna dengan jabatan Admin Owner di dalam sistem. Aktivitas yang dapat dilakukan pengguna dapat dilihat pada *use case diagram* Gambar 3. 3.



Gambar 3. 3 *Use Case Diagram* Admin Owner

Pada Gambar 3.3 menjelaskan *use case diagram* pengguna dengan jabatan Admin Owner yang terdiri dari aktor Admin Owner dan

7 *use case*. Pada gambar diatas terlihat bahwa *use case* Login merupakan syarat sebelum dijalankan *use case* lainnya seperti Mengelola Pengguna, Mengelola Profil, Mengelola Data Penjualan, Mengelola Data Motor, dan Logout. Relasi antara aktor dengan *use case* Login adalah asosiasi, sedangkan relasi antara Login dengan Mengelola Pengguna adalah include.

Pada *use case* Mengelola Profil terdiri dari mengubah profil dan mengubah password. Kemudian pada *use case* Mengelola Pengguna terdiri dari mengubah data profil dan jabatan pengguna. Lalu pada *use case* Mengelola Data Penjualan terdiri dari create input data motor paling dicari. Dan pada *use case* Mengelola Data Motor Admin Owner menambah data motor dan dapat mengubah status motor.

Tabel 3. 1 Deskripsi *Use Case Diagram* Login (Admin)

Nama <i>Use Case</i>	Login
Aktor yang terlibat	Admin Owner
Deskripsi singkat	Admin Owner akan login dengan menggunakan username dan password pada halaman login
Trigger	Admin Owner ingin masuk ke sistem
Precondition	Membuka web awal
Action	api.auth.login
Postcondition	Admin Owner masuk ke sistem dengan jabatan Admin Owner

Pada Tabel 3. 1 menjelaskan mengenai *use case diagram* login pada pengguna dengan jabatan Admin Owner. Untuk dapat masuk ke dalam sistem, pengguna harus memasukkan username dan password yang sudah terdaftar dalam sistem. Username dan password dibuat pada saat sistem pertama kali dirilis. Kemudian setelah sistem menyatakan

bahwa pengguna layak masuk ke sistem, pengguna akan diarahkan ke halaman dashboard sebagai Admin Owner.

Tabel 3. 2 Deskripsi *Use Case Diagram* Mengelola Profil (Admin)

Nama <i>Use Case</i>	Kelola Profile
Aktor yang terlibat	Admin Owner
Deskripsi singkat	Admin Owner dapat mengedit profil dan mengubah password sendiri pada halaman profile
Trigger	Admin Owner ingin mengelola profilnya
Precondition	Login sebagai Admin Owner dan membuka menu profile
Action	api.auth.update_profile
Postcondition	Data profile berhasil diubah atau password berhasil diubah

Pada Tabel 3. 2 menunjukkan *use case diagram* Mengelola Profil pada pengguna dengan jabatan Admin Owner. Admin Owner dapat mengubah nama dan identitas pada profilnya. Setelah sistem menyetujui perubahan maka sistem akan menampilkan pemberitahuan bahwa data profil berhasil diubah.

Tabel 3. 3 Deskripsi *Use Case Diagram* Logout (Admin)

Nama <i>Use Case</i>	Logout
Aktor yang terlibat	Admin Owner
Deskripsi singkat	Admin Owner akan logout
Trigger	Admin Owner ingin keluar dari sistem
Precondition	Login sebagai Admin Owner dan mengklik tombol logout
Action	api.auth.logout
Postcondition	Admin Owner keluar dari sistem

Pada Tabel 3. 3 menjelaskan *use case diagram* logout pada pengguna dengan jabatan Admin Owner. Ketika Admin Owner mengklik tombol logout maka sistem akan mengakhiri sesi dan

mengarahkan pengguna ke halaman login dengan menginformasikan bahwa pengguna telah keluar dari sistem.

Tabel 3. 4 Deskripsi *Use Case Diagram* Mengelola Pengguna (Admin)

Nama <i>Use Case</i>	Kelola Pengguna
Aktor yang terlibat	Admin Owner
Deskripsi singkat	Admin Owner dapat mengubah data, mengubah status, dan mengangkat pegawai baru
Trigger	Admin Owner ingin mengelola data pengguna lain
Precondition	Login sebagai Admin Owner dan membuka menu kelola pengguna
Action	api.user
Postcondition	Data pengguna berhasil diubah

Pada Tabel 3. 4 menjelaskan *use case diagram* Mengelola Pengguna pada pengguna dengan jabatan Admin Owner. Ketika Admin Owner mengklik menu kelola pengguna maka sistem akan mengarahkan ke halaman kelola pengguna. Pada halaman tersebut Admin Owner dapat mengubah, mengaktifkan, menonaktifkan, menghapus data serta mengubah jabatan pengguna.

Tabel 3. 5 Deskripsi *Use Case Diagram* Mengelola Data Penjualan (Admin)

Nama <i>Use Case</i>	Kelola Data Penjualan
Aktor yang terlibat	Admin Owner
Deskripsi singkat	Admin Owner dapat melakukan CRUD file informasi Data Penjualan
Trigger	Admin Owner ingin mengelola informasi Data Penjualan
Precondition	Login sebagai Admin Owner dan membuka menu kelola Data Penjualan
Postcondition	Data Penjualan berhasil diinput, Data Data Penjualan berhasil dipantau dan dievaluasi

Pada Tabel 3. 5 menjelaskan *use case diagram* Mengelola Data Penjualan pada pengguna dengan jabatan Admin Owner. Ketika Admin Owner mengklik menu kelola Data Penjualan maka sistem akan mengarahkan ke halaman kelola Data Penjualan.

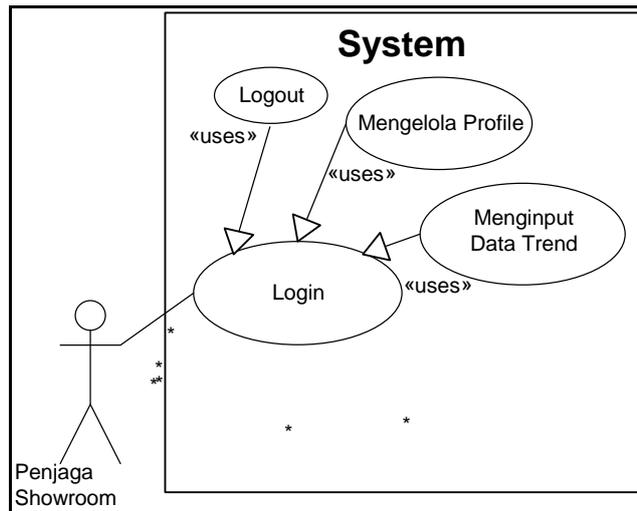
Tabel 3. 6 Deskripsi *Use Case Diagram* Mengelola data motor (Admin)

Nama <i>Use Case</i>	Kelola data motor
Aktor yang terlibat	Admin Owner
Deskripsi singkat	Admin Owner dapat menambah data motor dan mengatur data motor yang ada
Trigger	Admin Owner ingin mengelola data motor
Precondition	Login sebagai Admin Owner dan membuka menu kelola data motor
Action	api.master_project
Postcondition	Data proyek berhasil ditambah, Berhasil mengatur PIC proyek

Pada Tabel 3. 6 menjelaskan *use case diagram* Mengelola data motor pada pengguna dengan jabatan Admin Owner. Ketika Admin Owner mengklik menu kelola data motor maka sistem akan mengarahkan ke halaman kelola data motor. Pada halaman tersebut Admin Owner dapat melakukan penambahan dan mengatur data motor.

## 2. *Use Case Diagram* Penjaga Showroom

*Use case diagram* Penjaga Showroom merupakan hal-hal yang dapat dilakukan pengguna dengan jabatan Penjaga Showroom di dalam sistem. Aktivitas yang dapat dilakukan pengguna dapat dilihat pada *use case diagram* Gambar 3. 3.



Gambar 3. 4 *Use Case Diagram* Penjaga Showroom

Pada Gambar 3. 4 menjelaskan *use case diagram* pengguna dengan jabatan Penjaga Showroom yang terdiri dari aktor Penjaga Showroom dan 3 *use case*. Pada gambar diatas terlihat bahwa *use case* Login merupakan syarat sebelum dijalankan *use case* lainnya seperti Mengelola Profile, Menginput Data Trend, dan Logout. Relasi antara aktor dengan *use case*

Login adalah asosiasi, sedangkan relasi antara Login dengan Mengelola Profile adalah include. Pada *use case* Menginput Data Trend terdiri dari menambah Data Penjualan yang dicari oleh pengunjung walk-in ke showroom.

Tabel 3. 7 Deskripsi *Use Case Diagram* Login (Penjaga Showroom)

Nama <i>Use Case</i>	Login
Aktor yang terlibat	Penjaga Showroom
Deskripsi singkat	Penjaga Showroom akan login dengan menggunakan username dan password pada halaman login
Trigger	Penjaga Showroom ingin masuk ke sistem
Precondition	Membuka web awal

Action	api.auth.login
Postcondition	Penjaga Showroom masuk ke sistem dengan jabatan Penjaga Showroom

Pada Tabel 3. 7 menjelaskan mengenai *use case diagram* login pada pengguna dengan jabatan Penjaga Showroom. Untuk dapat masuk ke dalam sistem, pengguna harus memasukkan username dan password yang sudah terdaftar dalam sistem. Akun pengguna Penjaga Showroom ini sebelumnya telah terlebih dahulu didaftarkan oleh Admin Owner. Kemudian setelah sistem menyatakan bahwa pengguna layak masuk ke sistem, pengguna akan diarahkan ke halaman dashboard sebagai Penjaga Showroom.

Tabel 3. 8 Deskripsi *Use Case Diagram* Mengelola Profile

Nama <i>Use Case</i>	Kelola Profile
Aktor yang terlibat	Penjaga Showroom
Deskripsi singkat	Penjaga Showroom dapat mengedit profil dan mengubah pada halaman profile
Trigger	Penjaga Showroom ingin mengelola profilnya
Precondition	Login sebagai Penjaga Showroom dan membuka menu profile
Action	api.auth.update_profile
Postcondition	Data profile berhasil diubah

Pada Tabel 3. 8 menunjukkan *use case diagram* profile pada pengguna dengan jabatan Penjaga Showroom. Penjaga Showroom dapat mengubah profilnya. Setelah sistem menyetujui perubahan maka sistem akan menampilkan pemberitahuan bahwa data profil berhasil diubah.

Tabel 3. 9 Deskripsi *Use Case Diagram* Logout (Penjaga Showroom)

Nama <i>Use Case</i>	Logout
Aktor yang terlibat	Penjaga Showroom
Deskripsi singkat	Penjaga Showroom akan logout
Trigger	Penjaga Showroom ingin keluar dari sistem
Precondition	Login sebagai Penjaga Showroom dan mengklik tombol logout
Action	api.auth.logout
Postcondition	Penjaga Showroom keluar dari sistem

Pada Tabel 3. 9 menjelaskan *use case diagram* logout pada pengguna dengan jabatan Penjaga Showroom. Ketika Penjaga Showroom mengklik tombol logout maka sistem akan mengakhiri sesi dan mengarahkan pengguna ke halaman login dengan menginformasikan bahwa pengguna telah keluar dari sistem.

Tabel 3. 10 Deskripsi *Use Case Diagram* Menginput Data Data Penjualan (Penjaga Showroom)

Nama <i>Use Case</i>	Menginput Data Data Penjualan
Aktor yang terlibat	Penjaga Showroom
Deskripsi singkat	Penjaga Showroom dapat menambah data trend motot pengunjung ke Admin Owner
Trigger	Penjaga Showroom ingin menginput data trend nya
Precondition	Login sebagai Penjaga Showroom dan membuka menu Data Penjualan
Postcondition	Data Data Penjualan berhasil di input

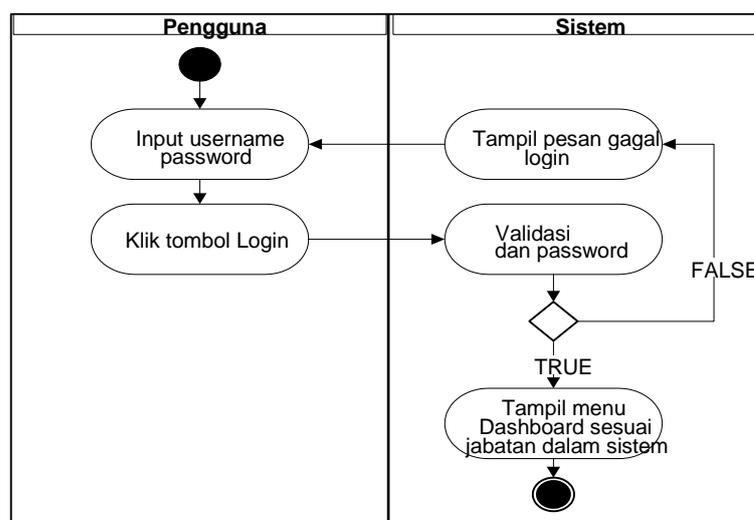
Pada Tabel 3. 10 menjelaskan *use case diagram* Menginput Data Data Penjualan pada pengguna dengan jabatan Penjaga Showroom. Data trend ini kedepannya akan dapat dilihat oleh Admin Owner dan digunakan sebagai bahan evaluasi saat akan melakukan pengadaan unit.

### 3. 4. 3 Perancangan *Activity Diagram*

*Activity Diagram* merupakan diagram yang menggambarkan aliran kerja atau aktivitas dari sebuah sistem atau proses bisnis. Fungsi *Activity Diagram* adalah untuk menggambarkan urutan aktivitas dalam sebuah proses maupun pada sistem berdasarkan proses yang terjadi dalam setiap *case* atau kondisi. *Activity Diagram* dibuat menggunakan *software* ciptaan Microsoft yaitu Microsoft Visio 2007. Berikut akan dijelaskan mengenai *Activity Diagram* dalam perancangan aplikasi manajemen proyek teknologi informasi.

#### 1. *Activity Diagram* Login

Aktivitas login merupakan aktivitas yang dilakukan oleh semua jabatan sebelum masuk ke sistem. Pada aktivitas login sistem memerlukan input berupa username dan password dari pengguna. Jika pengguna memasukkan input yang tidak sesuai dengan *database* yang terdapat dalam sistem maka pengguna tersebut tidak dapat masuk ke dalam sistem.

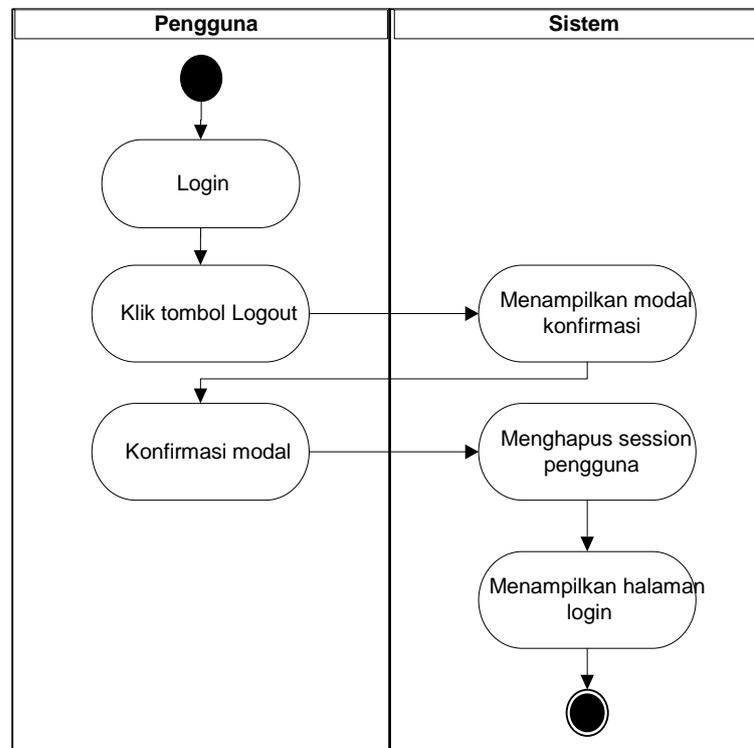


Gambar 3. 5 *Activity Diagram* Login

Gambar 3. 5 menggambarkan rangkaian proses pada setiap pengguna sebelum masuk ke dalam sistem yaitu dengan melakukan login. Setelah pengguna memasukkan input berupa username dan password lalu menekan tombol enter atau login maka sistem akan melakukan validasi terhadap input pengguna. Jika validasi sukses maka sistem akan mengarahkan pengguna pada halaman dashboard sesuai dengan jabatan yang terdaftar dalam sistem.

## 2. *Activity Diagram Logout*

Aktivitas logout merupakan aktivitas yang dilakukan oleh semua jabatan ketika akan keluar dari sistem. Pada aktivitas logout sistem akan menampilkan modal konfirmasi bahwa pengguna akan keluar dari sistem ketika pengguna mengklik tombol logout.



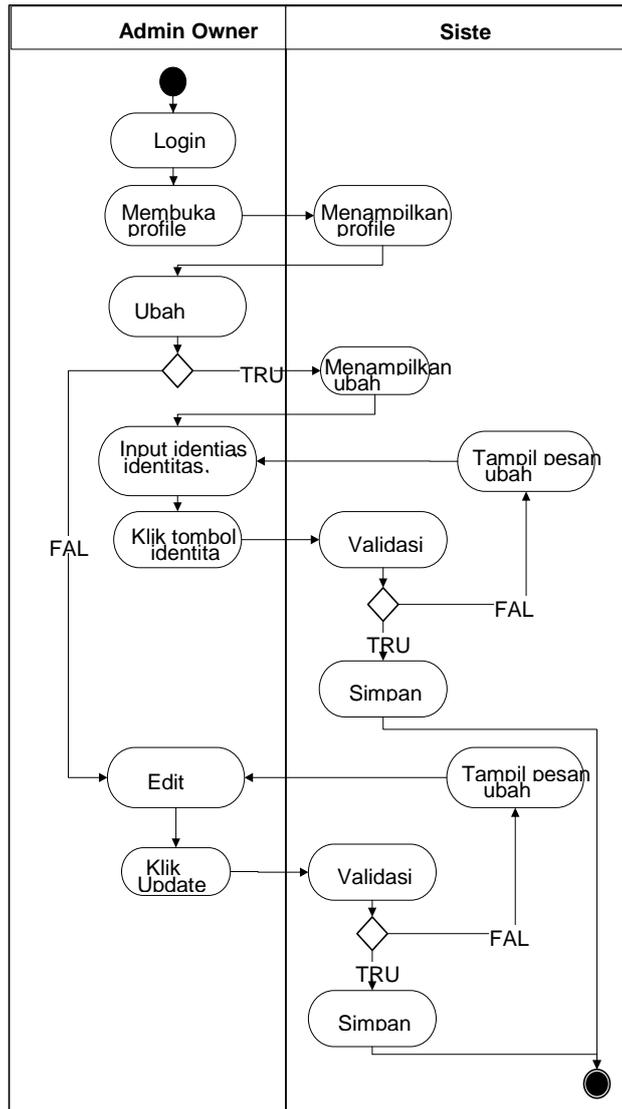
Gambar 3. 6 *Activity Diagram Logout*

Gambar 3. 6 menunjukkan proses yang dilalui pengguna sebelum keluar dari sistem atau logout. Setelah muncul konfirmasi modal apabila pengguna memilih “Ya” maka sistem akan menghapus session pengguna dari *database* dan menampilkan halaman login yang menandakan pengguna berhasil keluar dari sistem.

### 3. *Activity Diagram* Kelola Profile (Admin Owner)

Aktivitas pada menu profile terdiri dari 2 bagian yaitu ubah profile dan ubah password. Pada activity ini hanya dapat dilakukan oleh pengguna Admin Owner yang dapat mengelola profil dari masing-masing pegawai yang ada didalam sistem. Pada pegawai tidak disediakan menu ini agar Admin Owner dapat lebih mudah memantau dan mengontrol data pegawai. Sehingga pegawai tidak dapat merubah identitas mereka secara sepihak.

Gambar 3. 7 dibawah menggambarkan proses yang dilalui pengguna pada saat membuka halaman profile pengguna. Pada saat halaman profile pengguna ditampilkan akan ada 1 tombol yaitu Ubah Password dimana ketika pengguna menekan tombol tersebut maka sistem akan menampilkan form input password berupa password lama, password baru dan masukkan ulang password baru.

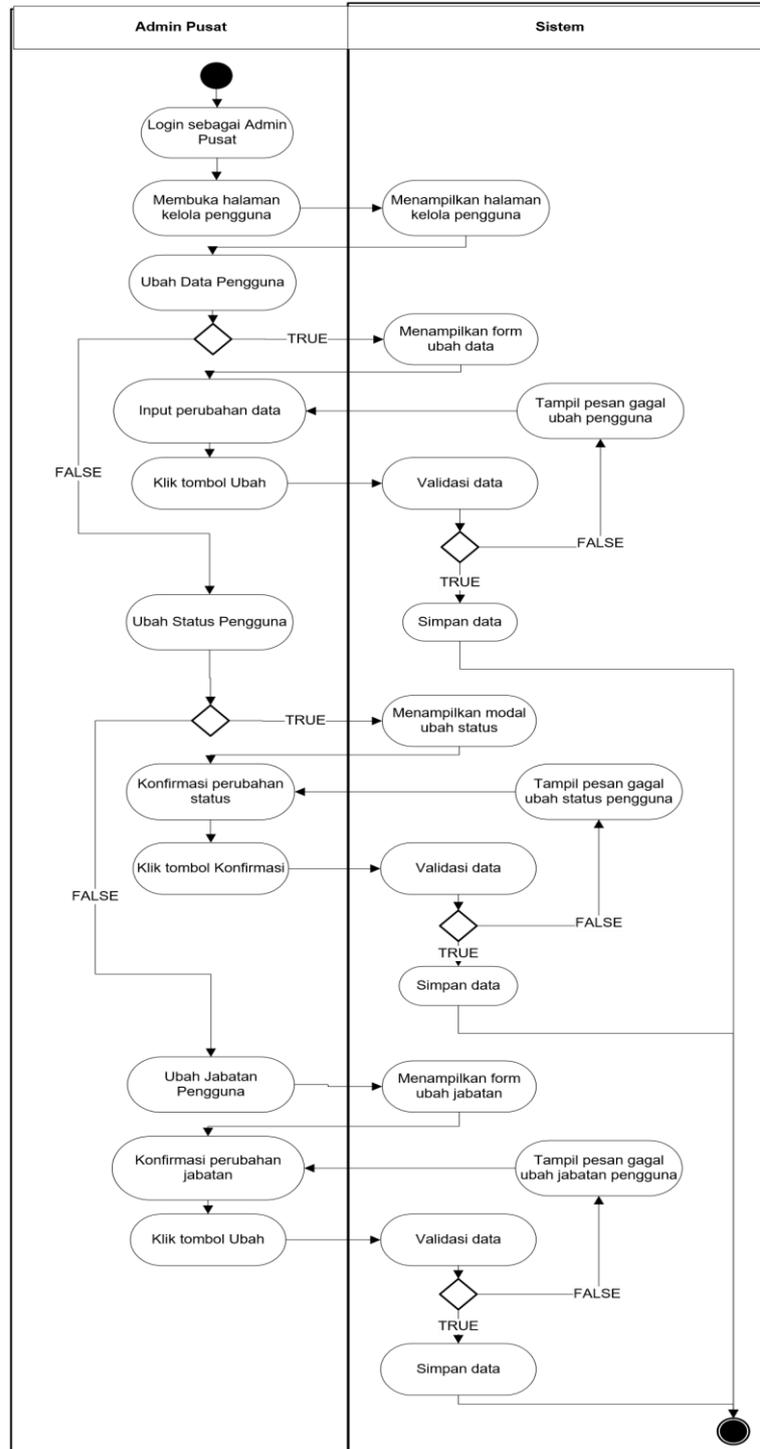


Gambar 3. 7 Activity Diagram Kelola Profile

Setelah pengguna mengisi 3 input tersebut dan menekan tombol Ubah Password kemudian sistem akan memvalidasi data dan menginformasikan apakah data berhasil disimpan. Sama halnya juga pada saat pengguna mengubah profilnya sistem pun akan melakukan pengecekan apakah profil yang diubah adalah profil yang valid dengan username yang berbeda antara satu pengguna pegawai dengan pegawai lainnya.

#### 4. Activity Diagram Kelola Pengguna (Admin Owner / Admin Pusat)

Pada aktivitas mengelola pengguna hanya tersedia bagi pengguna yang login dengan jabatan Admin Owner. Penjelasan selengkapnya dapat dilihat pada gambar berikut.



Gambar 3. 8 Activity Diagram Kelola Pengguna

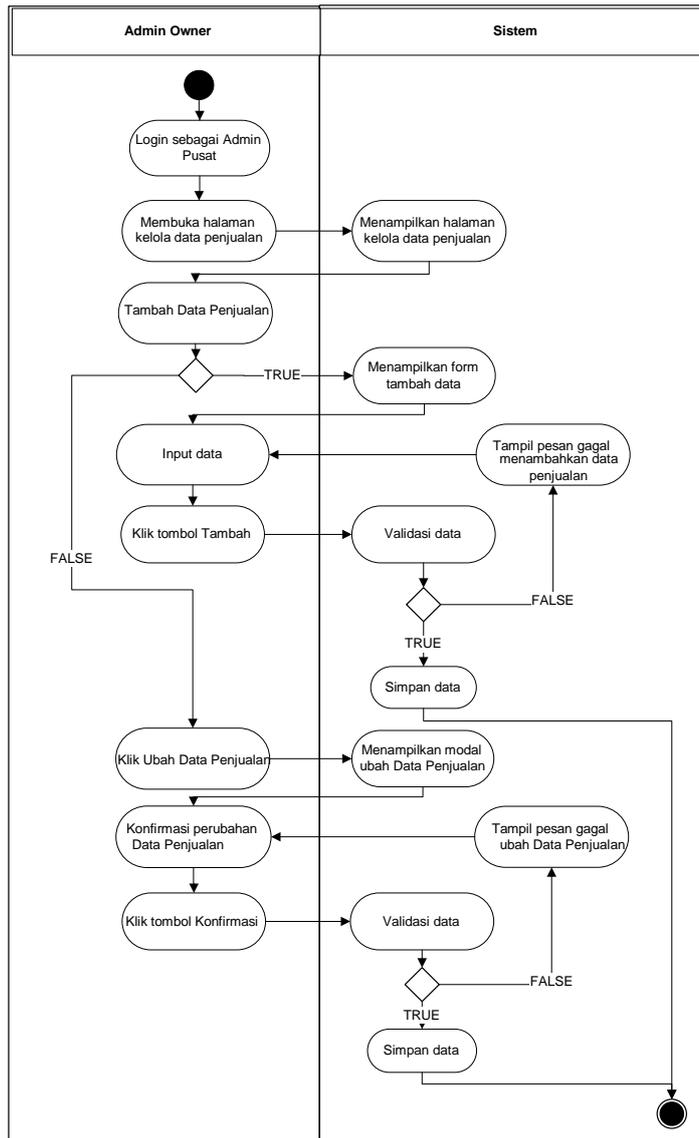
Pada Gambar 3. 8 menggambarkan fungsionalitas menu Kelola Pengguna. Pengguna yang ditampilkan pada menu ini merupakan pengguna dengan jabatan Penjaga Toko. Menu kelola pengguna terdiri dari 3 bagian yaitu mengubah data pengguna, mengubah status pengguna, dan mengubah jabatan pengguna.

Mengubah data pengguna berarti mengubah data diri atau atribut yang terdapat dalam pengguna. Mengubah status pengguna terdiri dari mengaktifkan pengguna yang tidak aktif, menonaktifkan pengguna aktif, menghapus pengguna. Sedangkan mengubah jabatan pengguna dilakukan untuk menaikkan jabatan dari Penjaga Showroom ke Manajer Proyek dan sebaliknya.

##### 5. *Activity Diagram* Kelola Data Penjualan (Admin Owner)

Pada aktivitas mengelola data penjualan hanya tersedia bagi pengguna yang login dengan jabatan Admin Owner. Pada bagian ini pengguna dengan jabatan Admin Owner dapat melakukan manajemen terhadap Data Penjualan. Menu kelola Data Penjualan bertujuan untuk mengumpulkan data *master* trend yang terdiri dari berbagai tipe kendaraan dan yang paling banyak dicari.

Admin Owner dapat melakukan CRUD pada menu ini agar master trend mempunyai pencatatan yang jelas baik di dalam sistem maupun sebagai disimpan sebagai dokumen pegangan. Penjelasan selengkapnya dapat dilihat pada gambar berikut.

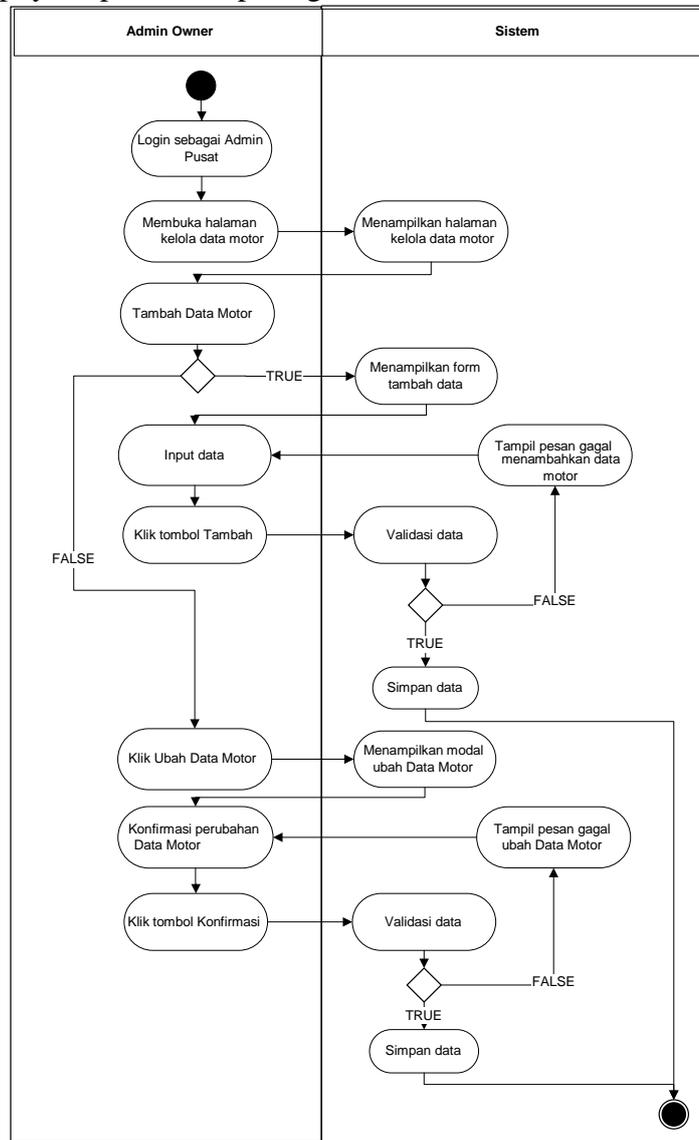


Gambar 3. 9 Activity Diagram Kelola Data Penjualan

Pada Gambar 3. 9 memaparkan fungsionalitas menu Kelola Data Penjualan. Pada sistem juga terdapat fitur memantau data yang di input oleh Penjaga Showroom. Kondisi ini dapat terjadi apabila data trend yang dimasukkan Penjaga Showroom mungkin kurang benar dalam memilih kondisi sehingga memerlukan pemantauan dari Admin Owner.

## 6. Activity Diagram Kelola data motor (Admin Owner / Admin Pusat)

Pada aktivitas mengelola kategori spesifikasi hanya tersedia bagi pengguna yang login dengan jabatan Admin Owner. Penjelasan selengkapnya dapat dilihat pada gambar berikut.

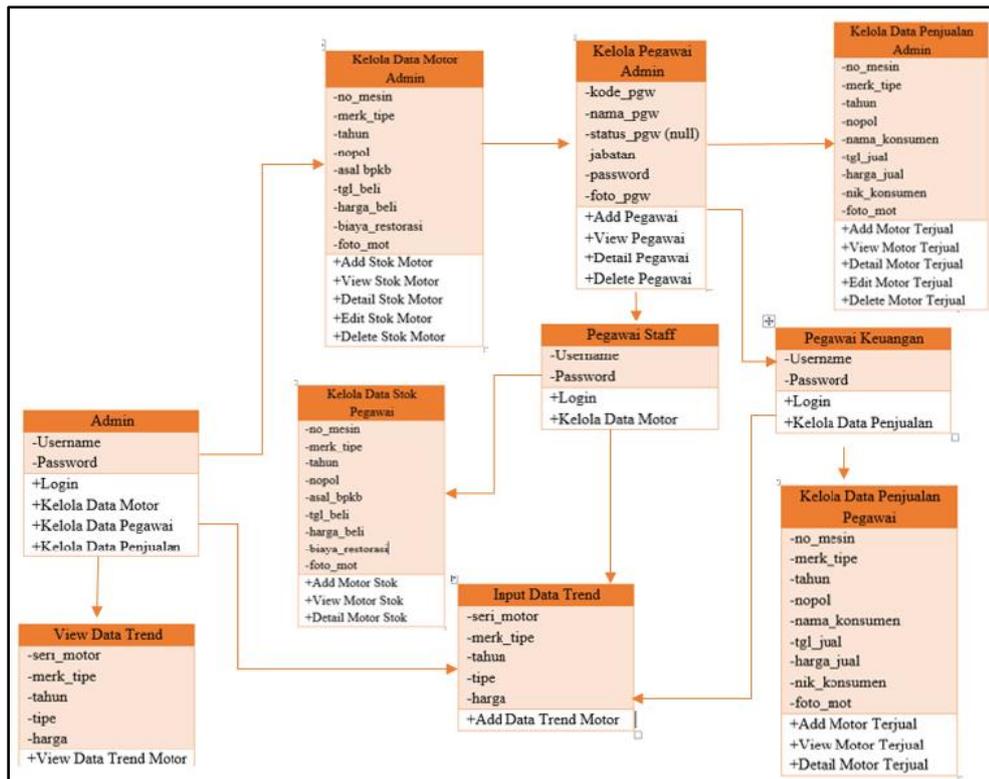


Gambar 3. 10 Activity Diagram Kelola data motor

Gambar 3. 10 menunjukkan menu pengelolaan proyek oleh Admin Owner. Pada menu ini Admin Owner dapat menambah Data Motor dengan menginput data yang diperlukan seperti Merek, Tipe, tahun, dll. Lalu terdapat fitur untuk mengubah data motor yang sudah

ada apabila sewaktu-waktu data motor mengalami perubahan identitas seperti Nopol ketika terjadi perpanjangan pajak 5 tahunan.

### 3. 4. 4 Perancangan *Class Diagram*



Gambar 3. 11 *Class Diagram* pada Sistem

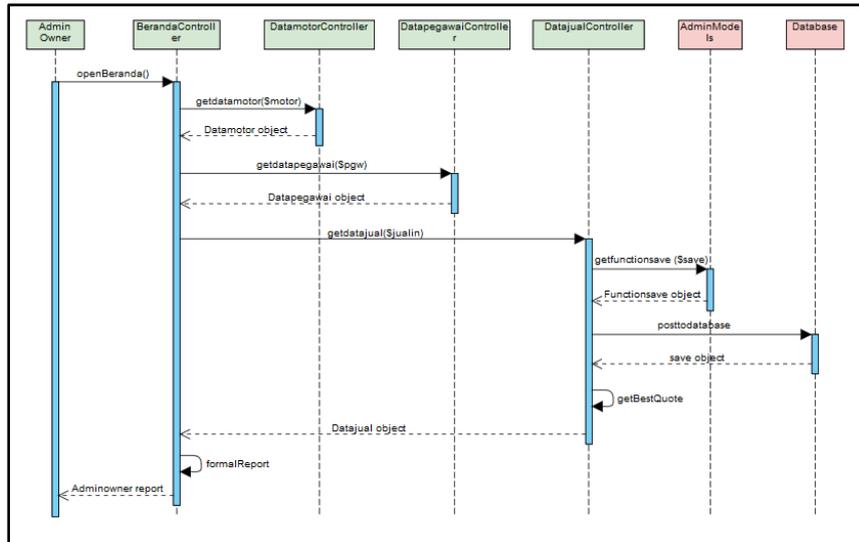
Gambar 3. 11 menjelaskan *Class Diagram* Sistem yang terdiri dari 3 jenis pengguna, yaitu Admin Owner, Pegawai Staff, dan Pegawai Keuangan. Ketika pengguna ini dapat melakukan hal yang sama pada dasarnya, namun untuk pengguna Pegawai Keuangan dan Pegawai Staff tidak dapat melakukan edit dan delete pada data yang mereka inputkan untuk menghindari kemungkinan kecurangan manipulasi data harga yang dapat merugikan pemilik perusahaan.

Dijelaskan pula pada Gambar 3.11 diatas jika Pegawai Staff dan Pegawai Keuangan dapat turut memasukkan data trend motor yang dicari oleh pengunjung. Namun untuk view hasil akhir data trend motor hanya akan ditampilkan pada pengguna Admin selaku pemilik showroom.

Untuk Class Kelola Data Motor baik untuk Admin dan Pegawai Staff sama-sama memiliki kesamaan pada jenis *data entry* yang dimasukkan. Diantaranya adalah memasukkan data no\_mesin, merk\_type, tahun, nopol, asal\_bpkb, tgl\_beli, harga\_beli\_biaya\_restorasi, dan foto unit. Sedangkan untuk Class Kelola Data Penjualan jenis data entry nya hampir mirip dengan Kelola Data Motor, hanya saja data entry asal\_bpkb, tgl\_beli, harga\_beli\_biaya\_restorasi, dan foto unit pada Class Kelola Data Motor diganti menjadi nama\_konsumen, tgl\_jual, harga\_jual, nik\_konsumen, dan foto\_mot pada Class Kelola Data Penjualan.

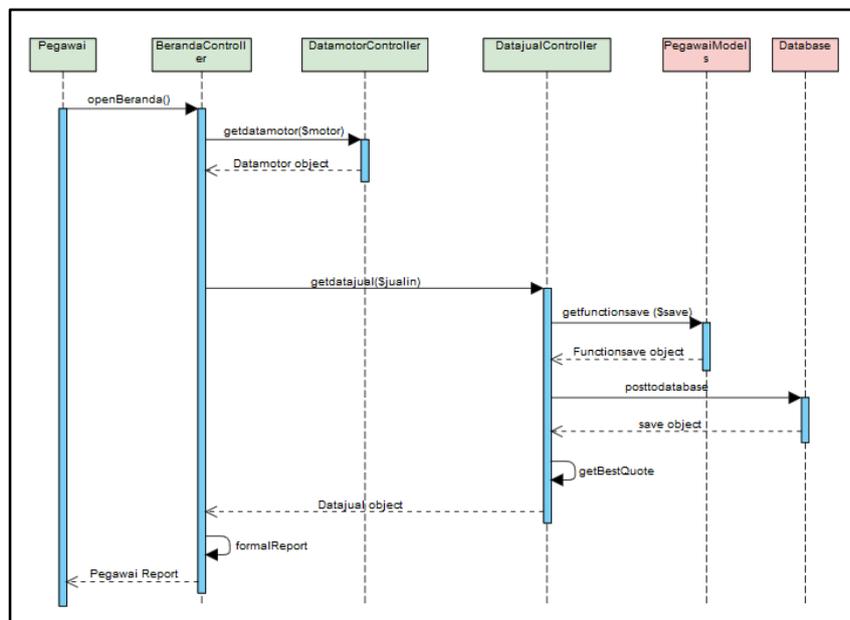
### **3. 4. 5 Perancangan *Sequence Diagram***

*Sequence Diagram* digunakan untuk menggambarkan skenario atau rangkaian langkah-langkah yang dilakukan sebagai respon suatu kejadian untuk menghasilkan output tertentu. *Sequence Diagram* diawali dari apa yang metrigger aktivitas tersebut, proses dan perubahan apa saja yang terjadi secara internal dan output apa yang dihasilkan (Nanthaamornphong and Leatongkam, 2019). *Sequence Diagram* dibuat menggunakan *tool* online berbasis web yaitu [online.visual-paradigm](http://online.visual-paradigm.com).



Gambar 3. 12 Sequence Diagram Admin

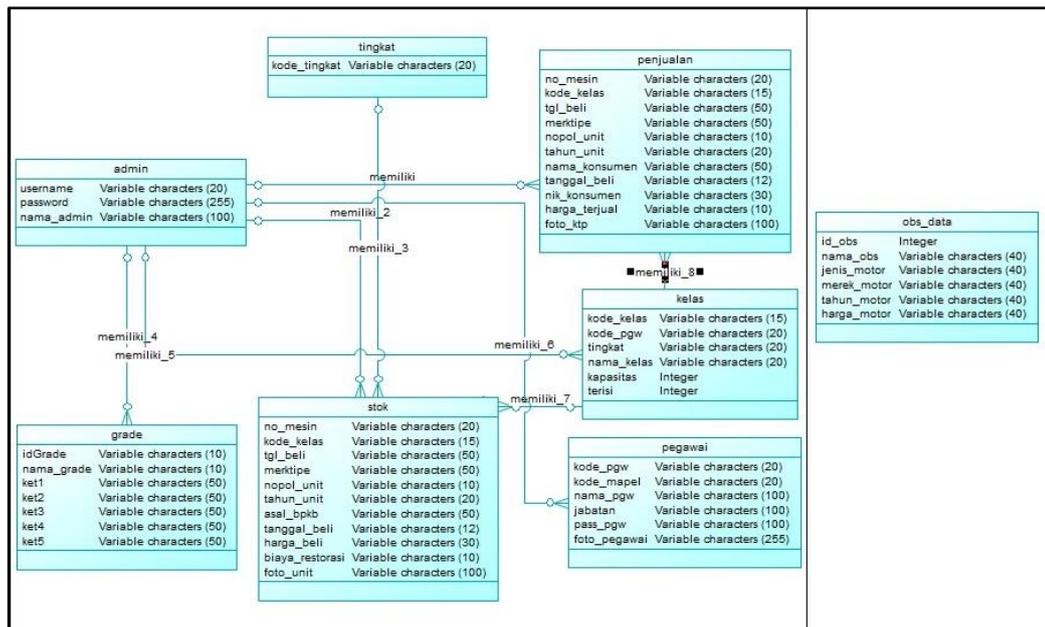
Pada Gambar 3.12 merupakan *Sequence Diagram* Admin. Pertama admin akan diperlihatkan langsung menu yang pertama yaitu beranda, lalu admin dapat menambahkan data stok motor, data pegawai, dan data penjualan. Kemudian data tersebut akan tersimpan ke *database* dan menampilkan pesan “data disimpan”



Gambar 3. 13 Sequence Diagram Pegawai

Pada Gambar 3.13 merupakan *Sequence Diagram* Pegawai. Pertama pegawai akan diperlihatkan langsung menu yang pertama yaitu beranda, lalu pegawai dapat menambahkan data stok motor dan data penjualan. Kemudian data tersebut akan tersimpan ke *database* dan menampilkan pesan “data disimpan”

### 3. 4. 6 Perancangan CDM (*Conceptual Data Model*)



Gambar 3. 14 CDM rancang bangun aplikasi

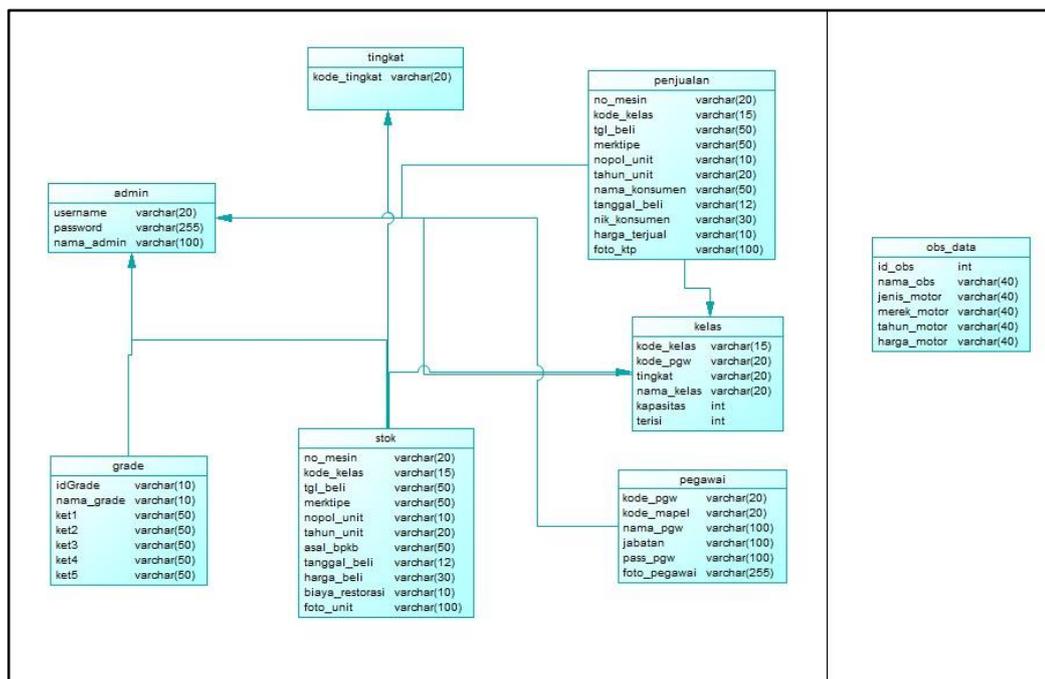
Pada Gambar 3. 14 menunjukkan representasi seluruh muatan informasi yang terkandung dalam *database*. Relasi yang dipakai dalam keseluruhan entitas adalah *dependency one to many*.

Pada Gambar 3. 14 akan dijelaskan mengenai entitas yang berperan penting dalam mengelola data motor yaitu entitas Stok karena memiliki keterkaitan dengan banyak entitas. Entitas Stok dan Penjualan memiliki atribut *no\_mesin* sebagai *primary key*. Sedangkan entitas Pegawai memiliki atribut *kode\_pgw* sebagai *primary key*. Kemudian entitas Kelas yang digunakan untuk grade motor memiliki atribut *kode\_kelas* sebagai *primary key*. Untuk Entitas

Obs\_data memiliki atribut id\_obs sebagai *primary key* dengan tipe data *integer*. Dan yang terakhir entitas Grade yang digunakan sebagai data acuan grade memiliki atribut idGrade sebagai *primary key*.

Pada Gambar 3.14 diatas untuk entitas Obs\_data terpisah dengan entitas lainnya dikarenakan pada data itulah nanti akan disimpan data-data yang diolah menjadi data trend motor dan hanya ditampilkan saja pada beranda Admin. Sehingga entitas tersebut tidak berelasi secara langsung dengan entitas lainnya.

### 3. 4. 7 Perancangan PDM (*Physical Data Model*)



Gambar 3. 15 PDM rancang bangun aplikasi

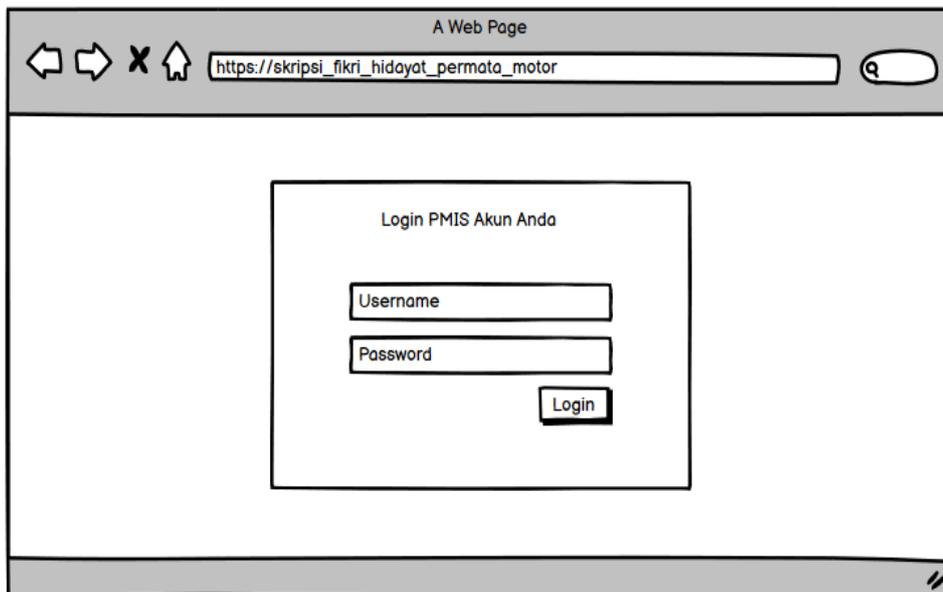
Gambar 3. 15 menunjukkan implementasi basis data yang akan digunakan. Dalam membuat PDM peneliti menggunakan Power Designer versi 16.5. PDM dibuat dengan cara membuat CDM terlebih dahulu sesuai Gambar 3.14 kemudian klik

menu Tools dan pilih Generate Physical Data Model dengan MySQL sebagai konfigurasi *databasenya*.

### 3. 4. 8 Desain Antarmuka

Desain antarmuka merupakan seperangkat alat/elemen yang digunakan untuk memanipulasi objek digital (Wandah and Rahina, 2018). Pembuatan desain antarmuka dalam penelitian ini menggunakan perangkat lunak berbasis desktop dari pihak ketiga yaitu Balsamiq Mockup 3. Desain antarmuka pada penelitian ini adalah sebagai berikut.

1. Desain Antarmuka halaman Login

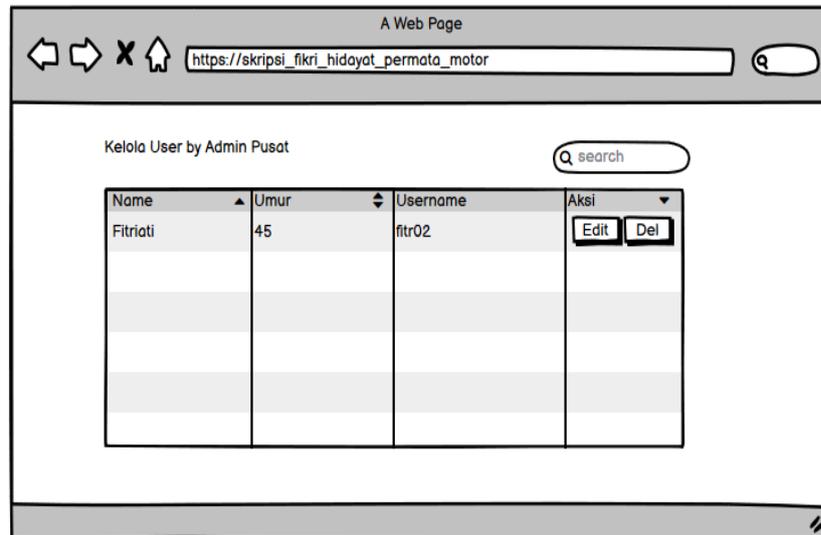


Gambar 3. 16 Desain antarmuka halaman login

Gambar 3. 16 menampilkan tampilan ketika halaman awal dibuka. Pengguna diminta untuk memasukkan autentikasi dalam bentuk username dan password. Apabila pengguna lupa password dapat mengklik link “Forgot password?” maka sistem akan menampilkan modal lupa password. Apabila pengguna belum terdaftar dalam sistem,

pengguna dapat mengklik link “Don’t have an account?” maka sistem akan menampilkan modal daftar.

## 2. Desain Antarmuka halaman Kelola Pengguna (Admin Owner)

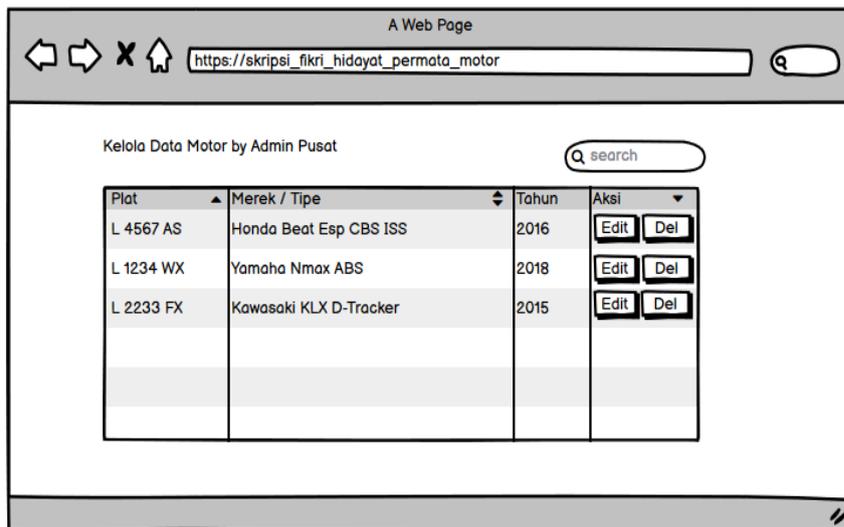


Gambar 3. 17 Desain antarmuka halaman kelola pengguna

Gambar 3. 17 menampilkan tampilan ketika halaman kelola pengguna dibuka. Halaman ini hanya bisa diakses oleh pengguna dengan jabatan Admin Owner dalam sistem.. Ketika tombol “Edit” “Hapus” di klik maka sistem akan menampilkan modal konfirmasi “Apakah Anda yakin?” dimana ketika Admin Owner mengklik “Ya” maka sistem akan menjalankan perintah masing-masing, apabila “Tidak” maka sistem akan membatalkan perintah. Sebelum dilakukan penambahan atau pengeluaran anggota sistem akan menampilkan modal konfirmasi apakah Admin Owner sudah yakin dengan keputusannya. Apabila Admin Owner melakukan penambahan Pegawai maka sistem akan menginformasikan kepada Admin Owner berupa form tambah

pegawai yang akan berisi username dan password pegawai yang akan terdaftar pada sistem

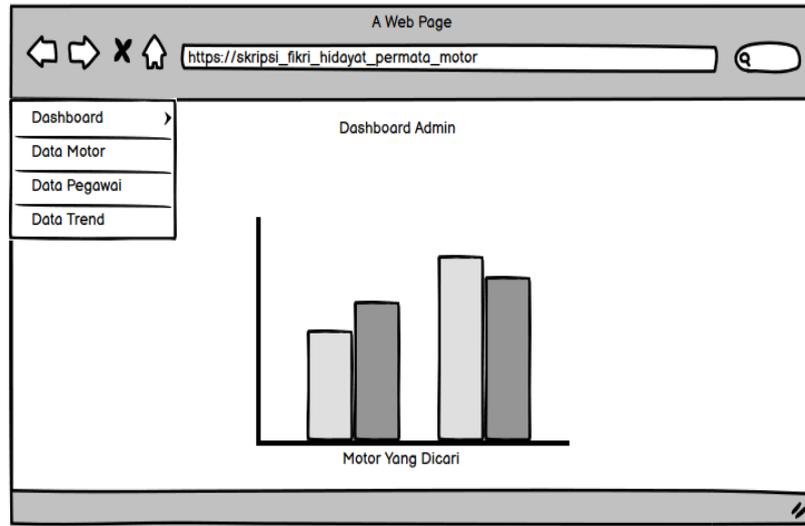
3. Desain Antarmuka halaman Kelola data motor (Admin Owner)



Gambar 3. 18 Desain antarmuka halaman kelola data motor

Gambar 3. 18 menampilkan tampilan ketika halaman kelola data motor dibuka. Pada halaman ini Admin Owner dapat menambah data motor dan mengubah data motor apabila ada perubahan. Apabila Admin Owner mengklik tombol “Tambah” maka sistem akan menampilkan modal berisi form untuk menambah data motor. Disediakan pula tombol edit dan delete pada bagian paling kanan tabel untuk digunakan jika ingin merubah atau menghapus data.

4. Desain Antarmuka halaman Dashboard (Admin Owner)



Gambar 3. 19 Desain antarmuka halaman dashboard Admin Owner

Gambar 3. 19 menampilkan tampilan ketika halaman dashboard pada Admin Owner dibuka. Pada halaman ini Admin Owner dapat memantau poin yang telah dikumpulkan dari proyek berdasarkan filter tahun. Apabila salah satu bar di klik maka akan menampilkan tabel berisi detail poin yang dikumpulkan.

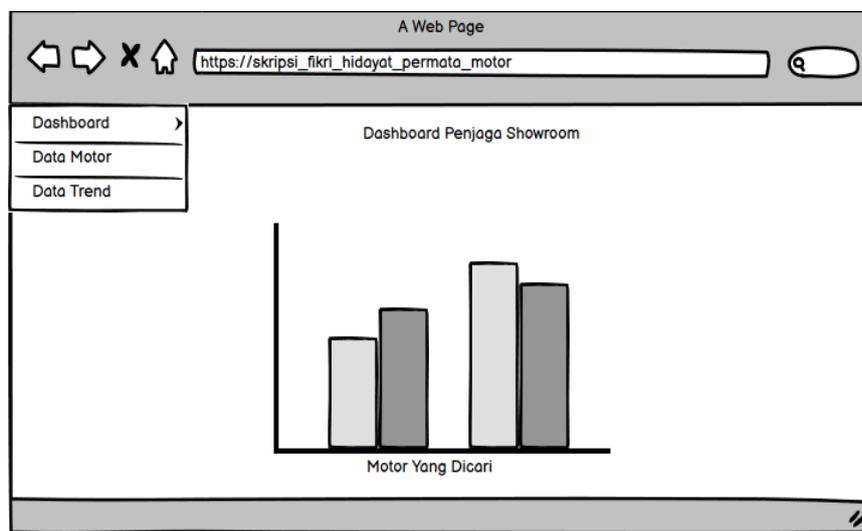
5. Desain Antarmuka halaman Trend Motor (Admin Owner)

Harga	Merek	Tahun	Jumlah
Dibawah 7 juta	Honda	2005-2009	5
Antara 7-12 juta	Yamaha	2009-2012	4

Gambar 3. 20 Desain antarmuka halaman Trend Motor

Gambar 3. 20 menampilkan tampilan ketika halaman Trend Motor dibuka. Pada halaman ini Admin Owner dapat melihat hasil output kinerja Algoritma Simple Additive Weighting (SAW) yang menghasilkan daftar data trend motor paling dicari calon konsumen.

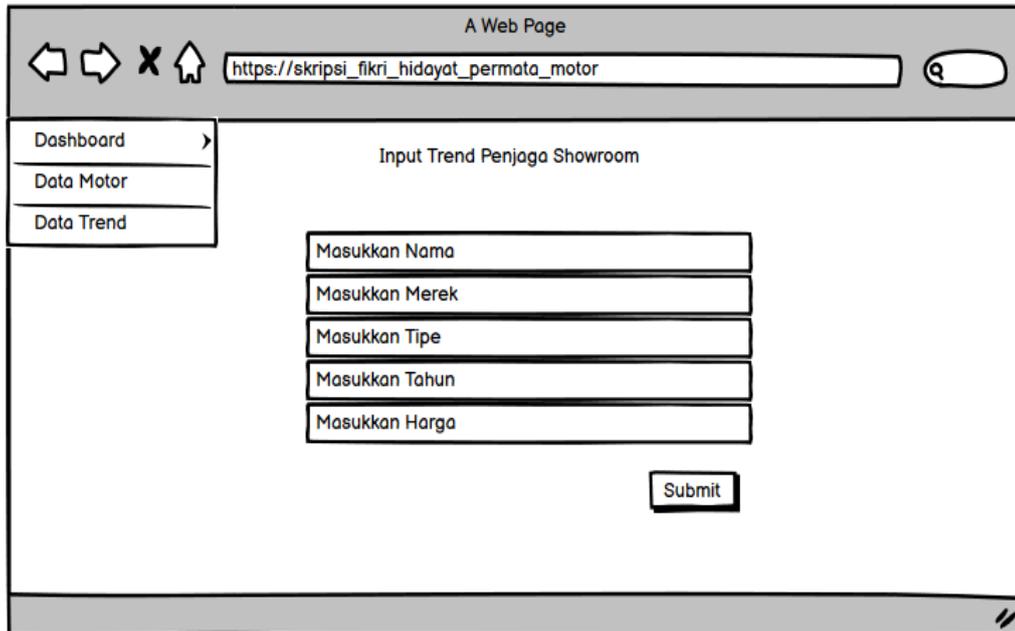
6. Desain Antarmuka halaman Dashboard (Penjaga Showroom dan Pegawai Lain)



Gambar 3. 21 Desain antarmuka halaman dashboard

Gambar 3. 21 menampilkan tampilan ketika halaman dashboard dibuka. Pada halaman ini Penjaga Showroom dapat melihat form dimana dia diwajibkan untuk memasukkan data Data Penjualan yang diperoleh dari pengunjung walk-in showroom.

## 7. Desain Antarmuka halaman Menginput Data Trend Motor (Penjaga Showroom)



The image shows a web browser window with the title 'A Web Page'. The address bar contains the URL 'https://skripsi\_fikri\_hidayat\_permata\_motor'. On the left side, there is a sidebar menu with three items: 'Dashboard', 'Data Motor', and 'Data Trend'. The main content area is titled 'Input Trend Penjaga Showroom' and contains five text input fields stacked vertically, each with a label: 'Masukkan Nama', 'Masukkan Merek', 'Masukkan Tipe', 'Masukkan Tahun', and 'Masukkan Harga'. Below these fields is a 'Submit' button.

Gambar 3. 22 Desain halaman Menginput Data Data Penjualan

Gambar 3. 22 menampilkan tampilan ketika halaman Menginput Data Data Penjualan dibuka. Pada halaman ini Penjaga Showroom dapat menambahkan Trend Motor yang berisi nama, no telp, Merek, Tipe, Tahun, dan Harga.

### 3.5 Skenario Pengujian Sistem

Pada penelitian ini sistem yang akan dibuat diuji menggunakan pengujian *blackbox* sesuai yang telah dijelaskan pada Tinjauan Pustaka. Jenis pengujian *blackbox* yang akan dilakukan berdasarkan fungsionalitas web aplikasi menggunakan *system testing* (Ali, Xiaoling and District, 2019). Pengujian *system testing* didasarkan pada *use case diagram* dan tabel deskripsi yang telah dibuat sebagai acuannya. Selain itu metode *Black Box* testing yang

digunakan ialah *User Acceptance Testing* dikarenakan sistem ini dibuat berdasarkan kebutuhan pengguna Permata Motor Semolowaru.

Pengujian akan dilakukan oleh 28 responden yang terdiri dari 8 responden dari perusahaan Permata Motor, 14 responden dari dealer motor bekas rekan bisnis dari perusahaan Permata Motor dan 6 responden dari mahasiswa Fakultas Ilmu Komputer Universitas Pembangunan Nasional “Veteran” Jawa Timur melalui Form Kuisisioner yang dibuat dan dicetak oleh peneliti. Berikut ini adalah daftar jenis pengujian yang akan diuji.

Tabel 3. 11 Skenario pengujian sistem

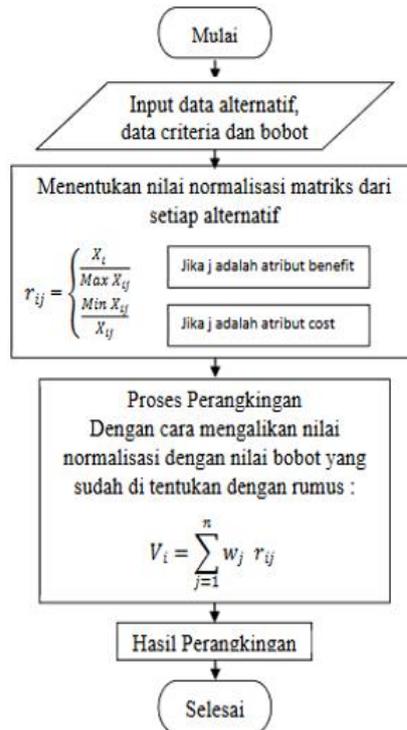
No	Jenis Pengujian	Aksi	Harapan
1	Login	Memasukkan username terdaftar dan password salah	Sistem menolak permintaan
		Memasukkan username yang tidak terdaftar	Sistem menolak permintaan
		Memasukkan username dan password dengan benar	Sistem memvalidasi input dan membuka halaman dasbor
2	Reset Password	Memasukkan username yang tidak terdaftar	Sistem menolak permintaan
		Memasukkan username terdaftar	Sistem memvalidasi input dan mengirim password baru melalui username
3	Logout	Mengklik tombol logout	Sistem menampilkan modal konfirmasi apakah pengguna yakin akan keluar dari sistem
4	Mengelola Profil	Belum login dan membuka url mengelola profil	Sistem menolak permintaan dan mengarahkan pengguna ke halaman login
		Sudah login dan membuka url kelola profil	Sistem membuka menu kelola profil
5	Mengelola Pengguna	Belum login dan membuka url mengelola pengguna	Sistem menolak permintaan dan mengarahkan pengguna ke halaman login

		Sudah login sebagai pengguna selain Admin Owner dan membuka url kelola pengguna	Sistem menolak permintaan dan mengarahkan pengguna ke halaman login
		Sudah login sebagai Admin Owner dan membuka url kelola pengguna	Sistem membuka menu kelola pengguna
6	Mengelola Data Penjualan	Belum login dan membuka url mengelola Data Penjualan	Sistem menolak permintaan dan mengarahkan pengguna ke halaman login
		Sudah login sebagai pengguna selain Admin Owner dan membuka url kelola Data Penjualan	Sistem menolak permintaan dan mengarahkan pengguna ke halaman login
		Sudah login sebagai Admin Owner dan membuka url kelola Data Penjualan	Sistem membuka menu kelola spesifikasi
7	Mengelola Data Motor	Belum login dan membuka url mengelola Data Motor	Sistem menolak permintaan dan mengarahkan pengguna ke halaman login
		Sudah login sebagai pengguna selain Admin Owner dan membuka url kelola Data Motor	Sistem menolak permintaan dan mengarahkan pengguna ke halaman login
		Sudah login sebagai Admin Owner dan membuka url kelola Data Motor	Sistem membuka menu kelola Data Motor
9	Mengelola Pegawai	Belum login dan membuka url mengelola Pegawai	Sistem menolak permintaan dan mengarahkan pengguna ke halaman login
		Sudah login dan membuka url mengelola pegawai	Sistem membuka fitur kelola pegawai

Tabel 3.23 menjelaskan skenario pengujian sistem pada penelitian. Pengujian dilakukan berupa fungsionalitas yang terdiri dari 11 menu. Batasan pengujian yang diuji berhubungan dengan hak akses setiap jabatan dalam sistem yaitu Admin Owner, Penjaga Showroom, dan Pegawai biasa. Apabila pengguna

yang masuk ke dalam sistem melakukan akses terhadap fitur yang tidak ditujukan kepadanya maka sistem akan menolak permintaan dalam bentuk pesan.

### 3.6 Rancangan Algoritma Simple Additive Weighting



Gambar 3. 23 Flowchart Metode SAW

Pada flowchart diatas dapat kita pahami bahwa proses pertama ketika menggunakan metode SAW adalah menginput data kriteria dan bobotnya, kemudian dilakukan proses normalisasi nilai matriks dari setiap alternatif yang muncul. Kemudian dilakukan proses perangkingan dengan mengalikan nilai normalisasi dengan nilai bobot yang telah ditentukan. Terakhir akan muncul hasil perangkingan (Purwanto & Arfyanti, 2012)(Hariyanto & Khotimah, 2018). Dan data ranking ini dapat digunakan oleh pemilik showroom sebagai bahan pertimbangan rekomendasi

Tabel 3. 12 Kriteria dan Subkriteria

Kriteria	Subkriteria
Jenis Motor	<ul style="list-style-type: none"> <li>- Matic</li> <li>- Sport Naked</li> <li>- Sport Fairing</li> <li>- Bebek</li> <li>- Sport Trail</li> </ul>
Merek Motor	<ul style="list-style-type: none"> <li>- Honda</li> <li>- Yamaha</li> <li>- Kawasaki</li> <li>- Suzuki</li> <li>- Merek Lain</li> </ul>
Tahun Motor	<ul style="list-style-type: none"> <li>- 2017 s/d 2020</li> <li>- 2013 s/d 2016</li> <li>- 2009 s/d 2012</li> <li>- 2005 s/d 2008</li> <li>- 2001 s/d 2004</li> </ul>
Harga Motor	<ul style="list-style-type: none"> <li>- Dibawah Rp7.000.000</li> <li>- Antara Rp7.000.000 – Rp12.000.000</li> <li>- Antara Rp12.000.000 – Rp17.000.000</li> <li>- Antara Rp 17.000.000 – Rp22.000.000</li> <li>- Diatas Rp22.000.000</li> </ul>

Semua kriteria dibagi dalam 5 subkriteria, salah satunya pada kriteria Merek Motor memiliki 5 subkriteria yaitu Honda, Yamaha, Kawasaki, Suzuki, dan Merek Lain. Dari ke-empat kriteria ini, untuk kriteria Harga Motor merupakan Kriteria COST, sedangkan ketiga kriteria lain yaitu Jenis Motor, Merek Motor, dan Tahun Motor merupakan kriteria BENEFIT.

### 3.6.1 Analisis Perhitungan Kebutuhan SAW

#### a. Bobot Kriteria dan Subkriteria

Untuk menentukan produk motor terbaik dengan metode *Simple Additive Weighting* di perlukan beberapa kriteria yang digunakan sebagai acuan dalam

pengambilan keputusan. Untuk itu dibuatlah tabel kriteria dan subkriteria dibawah ini yang menjelaskan bobot – bobot dari masing masing kriteria (Masykur & Mahmudi, 2016)(Nalsa C. R., 2017)(Irawan S., 2019), yaitu sebagai berikut:

Tabel 3. 13 Kode dan Ketentuan Kriteria

NO	Kode Kriteria	Ketentuan Kriteria	Bobot Kriteria	Jenis Atribut
1	K1	Jenis Motor	35	Benefit
2	K2	Merek Motor	25	Benefit
3	K3	Tahun Motor	25	Benefit
4	K4	Harga Motor	15	Cost

Tabel 3. 14 Subkriteria Jenis Motor

Jenis Motor	Bobot
Matic	5
Sport Naked	4
Sport Fairing	3
Bebek	2
Sport Trail	1

Tabel 3. 15 Subkriteria Merek Motor

Merek Motor	Bobot
Honda	5
Yamaha	4
Kawasaki	3
Suzuki	2
Merek Lain	1

Tabel 3. 16 Subkriteria Tahun Motor

Tahun Motor	Bobot
2017 – 2020	5
2013 – 2016	4
2009 – 2012	3
2005 – 2008	2
2001 – 2004	1

Tabel 3. 17 Subkriteria Harga Motor

Harga Motor	Bobot
Dibawah Rp7.000.000	5
Antara Rp7.000.000 – Rp12.000.000	4
Antara Rp12.000.000 – Rp17.000.000	3
Antara Rp 17.000.000 – Rp22.000.000	2
Diatas Rp22.000.000	1

b. Nilai Preferensi

Untuk menentukan produk motor terbaik, terdapat lima nilai kelayakan yang terdiri dari 1 – 5 yang digunakan untuk pemberian bobot pada masing-masing kriteria (Pradipta & Diana, 2017).

Tabel 3. 18 Nilai Preferensi

NO	Keterangan Nilai	Jumlah Nilai
1	Sangat Tinggi	5
2	Tinggi	4
3	Cukup	3
4	Rendah	2
5	Sangat Rendah	1

Tabel 3. 19 Data Alternatif

NO	Alternatif	Nama Alternatif
1	A1	Beat ESP
2	A2	Vixion FI
3	A3	Nmax
4	A4	KLX-L
5	A5	CBR 150R
6	A6	Scoopy Karburator

### 3.6.2 Hasil Rancangan Algoritma SAW

$$r_{ij} = \begin{cases} \frac{x_{ij}}{\text{Max}_i x_{ij}} & \text{jika } j \text{ ialah atribut keuntungan (benefit)} \\ \frac{\text{Min}_i x_{ij}}{x_{ij}} & \text{jika } j \text{ ialah atribut biaya (cost)} \end{cases} \quad V_i = \sum_{j=1}^n w_j r_{ij}$$

Gambar 3. 24 Rumus Normalisasi & Perangkingan SAW

Pada penelitian ini, rumus normalisasi matriks dan perangkingan yang digunakan pada Metode Simple Additive Weighting (SAW) dapat dilihat pada Gambar 3.24 diatas. Kemudian dilakukan analisa dari rumus tersebut diatas akan dirubah mejadi barisan kode PHP dengan CodeIgniter. Sehingga menjadi sistem pendukung keputusan yang siap digunakan oleh user.

+ Options							
	id_obs	nama_obs	jenis_motor	merek_motor	tahun_motor	harga_motor	
<input type="checkbox"/>	5	Beat ESP	5	5	4	4	Edit Copy Delete
<input type="checkbox"/>	6	Vixion FI	4	4	3	4	Edit Copy Delete
<input type="checkbox"/>	7	Nmax	5	4	4	2	Edit Copy Delete
<input type="checkbox"/>	8	KLX-L	1	3	4	2	Edit Copy Delete
<input type="checkbox"/>	9	CBR 150R	3	5	4	2	Edit Copy Delete
<input type="checkbox"/>	10	Scoopy Karburator	5	5	3	5	Edit Copy Delete

↑  Check all With selected: Edit Copy Delete Export

Gambar 3. 25 Atribut Database Sistem (obs\_data)

Pada Gambar 3.25 diatas merupakan database sistem pendukung keputusan dengan nama tabel “obs\_data” memiliki atribut id\_obs (auto increment), nama\_obs, jenis\_motor, merek\_motor, tahun\_motor, dan harga\_motor. Pada database inilah akan menjadi tempat data alternatif pada Tabel 8 disimpan.

Kemudian pada file M\_login.php yang berisi function getdataobs dan function maxi untuk melakukan penarikan data dari database dan untuk proses perangkingan. File ini merupakan Model yang digunakan untuk melakukan proses pengolahan data Trend Motor

```
public function getdataobs()
{
    return $this->db->get('obs_data')->result();
}

public function maxi()
{
    $this->db->select_max('jenis_motor', 'maxK1');
    $this->db->select_max('merek_motor', 'maxK2');
    $this->db->select_max('tahun_motor', 'maxK3');
    $this->db->select_max('harga_motor', 'maxK4');
    $query = $this->db->get('obs_data');
    return $query->row();
}
```

Kode 3.1 Function Pada M\_login.php

Pada kode 3.1 diatas memiliki 2 fungsi. Pertama function getdataobs digunakan untuk menarik data hasil inputan yang ada di database dengan nama tabel “obs\_data”. Kedua function maxi digunakan untuk normalisasi perangkingan proses metode SAW dengan mencari nilai maksimal dari masing-masing kriteria

```
public function algosaw()
{
    // return $request->all();
    // return $kJenis + $kMerek + $kTahun + $kHarga;

    $bobot = array(0.35,0.25,0.25,0.15);

    //Normalisasi
    $this->load->model('M_login');
    $maxi=$this->M_login->maxi();
    // echo $maxi['maxK1'];
    $sql3 = $this->M_login->getdataobs();

    foreach ($sql3 as $key) {
        $kriteria_1 = $key->jenis_motor;
        $kriteria_2 = $key->merek_motor;
        $kriteria_3 = $key->tahun_motor;
        $kriteria_4 = $key->harga_motor;
    }
}
```

```

        $jumlah=
($kriteria_1)+($kriteria_2)+($kriteria_3)+($kriteria_4);
        $poin= round(
        (($kriteria_1/$maxi->maxK1)*$bobot[0])+
        (($kriteria_2/$maxi->maxK2)*$bobot[1])+
        (($kriteria_3/$maxi->maxK3)*$bobot[2])+
        (($kriteria_4/$maxi->maxK4)*$bobot[3]),3);

        //Perangkingan
        $obs_data[]=array('nama'=>$key->nama_obs,
        'jumlah'=>$jumlah,
        'poin'=>$poin);}
        $data['obs_data'] = $obs_data;
        $this->load->view("admin/beranda", $data);}

```

### Kode 3.2 Function algosaw pada Beranda.php

Pada kode 3.2 diatas hanya memiliki satu fungsi yaitu *Function algosaw*, dimana fungsi ini bekerja dengan pertamakali mendeklarasikan nilai bobot dari keempat kriteria yang dimiliki. Kemudian melakukan proses normalisasi dengan memanggil *Function maxi* pada Kode 3.1 sehingga kemudian dapat dilakukan proses penjumlahan dari normalisasi. Setelah hasil angka penjumlahan keluar maka dilakukan proses terakhir, yaitu proses perangkingan.