

BAB 4. PEMBAHASAN

4.1 Studi Pendahuluan

Studi Pendahuluan merupakan tahap awal yang dilakukan dalam membuat penelitian ini. Studi pendahuluan dilakukan dengan melakukan studi pustaka dan studi lapang. Dalam penelitian ini studi pustaka yang dilakukan untuk mendapatkan dasar pustaka, pengungkapan teori sesuai permasalahan yang akan digunakan untuk menyelesaikan masalah dan mencapai tujuan penelitian adalah topik yang terkait yaitu perhitungan peramalan, analisis perhitungan kesalahan peramalan, stok produk, website, metode Triple Exponential Smoothing yang didapatkan melalui jurnal, buku dan artikel yang berkaitan dengan topik penelitian yang dituliskan pada bab 2. Untuk studi lapang, peneliti menemui langsung distributor Personal Beauty Jember dan menanyakan masalah-masalah yang dihadapi oleh Distributor. Masalah yang berkaitan dengan alur bisnis yang sedang di jalankan meliputi pengambilan produk, transaksi antar pelanggan dan harga minimal pengambilan produk.

4.2 Identifikasi Masalah

Permasalahan yang terjadi di bagian penyetokan produk kecantikan di bagian Distributor Personal Beauty Jember. Berdasarkan hasil wawancara dengan salah satu Distributor di Klinik Personal Beauty Jember, permasalahan yang terjadi yaitu penyetokan produk yang dilakukan setiap bulannya terkadang mengalami kelebihan (over stock) maupun kekurangan obat (stock out). Hal ini terjadi karena penyetokan yang dilakukan hanya berdasarkan asumsi, selain itu permintaan akan kebutuhan produk yang tidak dapat di prediksi sehingga penyediaan stok yang dilakukan menjadi tidak optimal sehingga membuat produk tidak terjual dan berdampak terhadap nilai investasi ataupun pengelolaan uang yang diputar.

Dari permasalahan tersebut maka peneliti mencoba membangun sebuah sistem peramalan yang dapat memprediksi ketersediaan produk dimasa yang akan datang, sehingga dapat menjadi acuan dalam melakukan proses pengambilan produk di pusat klinik dan dapat memberi kemudahan bagi Distributor Personal Beauty Jember dalam melakukan proses penyetokan. Dengan adanya sistem peramalan ini, diharapkan dapat menjadi solusi terbaik bagi Distributor di Klinik Personal Beauty Jember sehingga dapat mengurangi terjadinya kelebihan produk (*over stock*) maupun kekurangan produk (*stock out*) serta dapat membuat nilai investasi menjadi lebih optimal.

4.3 Perumusan Masalah

Berdasarkan identifikasi masalah yang telah dilakukan pada proses sebelumnya didapatkan rumusan masalah dalam penelitian yang dilakukan yaitu merancang sebuah sistem peramalan yang dapat memprediksi kebutuhan produk *skincare* di masa yang akan datang sehingga dapat dijadikan rekomendasi dalam proses penentuan stok produk *skincare*. Kedua, cara membangun sebuah sistem peramalan yang dapat meramalkan kebutuhan produk *skincare* secara akurat sesuai dengan pola data yang ada dengan menggunakan metode *triple exponential smoothing*.

4.4 Tujuan Penelitian

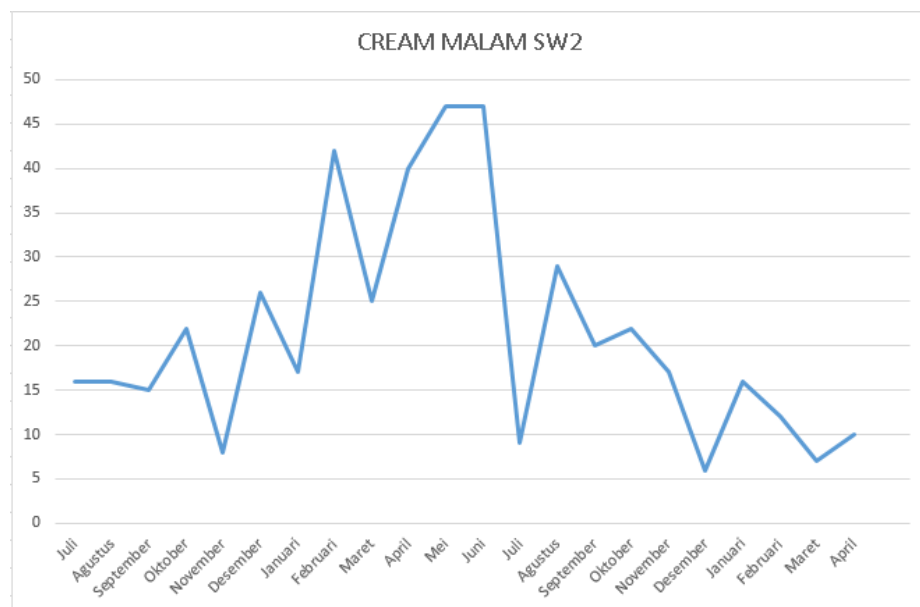
Tujuan penelitian yang dilakukan yakni mengetahui hasil berdasarkan rumusan masalah yang sudah dijelaskan sebelumnya. Tujuan pada penelitian yang dilakukan yaitu membangun sebuah sistem peramalan yang dapat membantu memberikan keputusan dalam penyetokan produk *skincare* pada Distributor Personal Beauty Jember sehingga dapat meminimalisir terjadinya produk *over stock* maupun *stock out*. Untuk mencapai tujuan penelitian, dibutuhkan data transaksi penjualan produk, lalu data tersebut dihitung menggunakan metode *triple exponential smoothing*, setelah itu hasil dari perhitungan metode *triple exponential smoothing* dihitung kembali menggunakan rumus kesalahan peramalan.

4.5 Pengumpulan Data

Pengumpululan data dilakukan dengan observasi, mendatangi langsung Klinik Personal Beauty Jember dan melihat proses bisnis yang ada, terutama di bagian salah satu distributor tentang pengambilan produk. Setelah melakukan observasi, peneliti kemudian melakukan *collecting* data untuk dijadikan *sample* dalam penelitian dimana data yang dikumpulkan terdiri dari 284 data produk yang terdiri atas paket dan satuan jenis serta data transaksi produk mulai dari Juli 2018 sampai dengan April 2020.

4.6 Pengelolaan Data

Setelah *sample* data didapatkan, selanjutnya peneliti melakukan proses pengolahan data. Pengolahan data dilakukan dengan cara memploting data kedalam sebuah grafik yang kemudian dianalisis lebih lanjut untuk membaca pola. Berikut contoh hasil dari pengolahan data berupa grafik transaksi produk dari Juli 2018 sampai dengan April 2020 :



Gambar 4.1 Grafik Transaksi produk cream malam sw2

4.7 Penerapan Metode *Triple Exponential Smoothing*

Setelah memploting data yang didapatkan ke dalam sebuah grafik dan pola dari data sudah dapat dibaca, maka selanjutnya yaitu menghitung berdasarkan metode yang akan digunakan untuk melakukan peramalan. Metode yang digunakan yaitu metode peramalan *triple exponential smoothing*. Metode *triple exponential smoothing* digunakan untuk menangani produk dengan pola data penjualannya memiliki kecenderungan terhadap trend, baik trend naik maupun trend turun serta memiliki faktor musiman atau hanya naik maupun turun pada bulan-bulan tertentu.

Di dalam metode *triple exponential smoothing*, data yang nantinya diramalkan akan dimuluskan terlebih dahulu sebanyak tiga kali pemulusan dan kemudian akan dihitung nilai peramalannya, selain itu hasil peramalan dari *triple exponential smoothing* juga bergantung kepada satu buah variabel yaitu variabel α yang dapat bernilai 0-1. Dalam perhitungan metode dibawah ini, variabel α yang digunakan bernilai 0,1. Berikut contoh perhitungan peramalan metode *triple exponential smoothing* dengan menggunakan data penjualan produk *cream* malam SW2 sebagai perhitungannya :

Tabel 4.1 Transaksi penjualan cream malam sw2

Periode	Jumlah
Juli 2018	16
Agustus 2018	16
September 2018	15
Oktober 2018	22
November 2018	8
Desember 2018	26
Januari 2019	17
Februari 2019	42
Maret 2019	25
April 2019	40

Mei 2019	47
Juni 2019	47
Juli 2019	9
Agustus 2019	29
September 2019	20
Oktober 2019	22
November 2019	17
Desember 2019	6
Januari 2020	16
Februari 2020	12
Maret 2020	7
April 2020	10

Perhitungan Peramalan Bulan Agustus 2017 :

$$\begin{aligned}
 S'_1 &= 0.1 * 16 + (1 - 0.1) * 16 \\
 &= 1.6 + 0.9 * 16 \\
 &= 16
 \end{aligned}$$

$$\begin{aligned}
 S''_1 &= 0.1 * 16 + (1 - 0.1) * 16 \\
 &= 1.6 + 0.9 * 16 \\
 &= 16
 \end{aligned}$$

$$\begin{aligned}
 S'''_1 &= 0.1 * 16 + (1 - 0.1) * 16 \\
 &= 1.6 + 0.9 * 16 \\
 &= 16
 \end{aligned}$$

$$at_1 = 3 * 16 - 3 * 16 + 16 = 16$$

$$\begin{aligned}
 bt_1 &= \frac{0.1}{2(1-0.1)^2} [6 - 5 * 0.1) * 16 - (10 - 8 * 0.1) * 16 + (4 - 3 * 0.1) * 16] \\
 &= 0
 \end{aligned}$$

$$\begin{aligned}
 ct_1 &= \frac{0.1^2}{(1-0.1)^2} (16 - 2 * 16 + 16) \\
 &= 0
 \end{aligned}$$

$$F_{t+m} = 16 + 0 * 1 + \frac{1}{2} 0 * 1^2$$

$$= 16$$

Perhitungan Peramalan Bulan September 2017 :

$$S'_2 = 0.1 * 16 + (1 - 0.1) * 16$$

$$= 1.6 + 0.9 * 16$$

$$= 16$$

$$S''_2 = 0.1 * 16 + (1 - 0.1) * 16$$

$$= 1.6 + 0.9 * 16$$

$$= 16$$

$$S'''_2 = 0.1 * 16 + (1 - 0.1) * 16$$

$$= 1.6 + 0.9 * 16$$

$$= 16$$

$$at_2 = 3 * 16 - 3 * 16 + 16 = 16$$

$$bt_2 = \frac{0.1}{2(1-0.1)^2} [6 - 5 * 0.1) * 16 - (10 - 8 * 0.1) * 16 + (4 - 3 * 0.1) * 16]$$

$$= 0$$

$$ct_2 = \frac{0.1^2}{(1-0.1)^2} (16 - 2 * 16 + 16)$$

$$= 0$$

$$F_{t+m} = 16 + 0 * 1 + \frac{1}{2} 0 * 1^2$$

$$= 16$$

Perhitungan Peramalan Bulan Oktober 2017 :

$$S'_3 = 0.1 * 15 + (1 - 0.1) * 16$$

$$= 1.5 + 0.9 * 16$$

$$= 15.9$$

$$S''_3 = 0.1 * 15.9 + (1 - 0.1) * 16$$

$$= 1.59 + 0.9 * 16$$

$$= 15.99$$

$$S'''_3 = 0.1 * 15.99 + (1 - 0.1) * 16$$

$$= 1.599 + 0.9 * 16$$

$$= 16$$

$$at_3 = 3 * 15.9 - 3 * 15.99 + 16 = 15.73$$

$$bt_3 = \frac{0.1}{2(1-0.1)^2} [6 - 5 * 0.1) * 15.9 - (10 - 8 * 0.1) * 15.99 + (4 - 3 * 0.1) *$$

$$16]$$

$$= -0.02$$

$$ct_3 = \frac{0.1^2}{(1-0.1)^2} (15.9 - 2 * 15.99 + 16)$$

$$= 0$$

$$F_{t+m} = 15.73 + (-0.02) * 1 + \frac{1}{2} 0 * 1^2$$

$$= 16$$

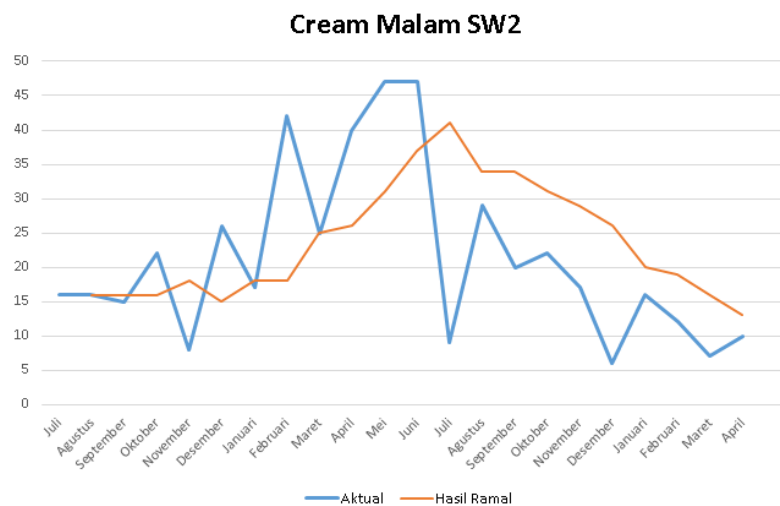
Untuk perhitungan di periode selanjutnya mengikuti rumus yang telah ada. Berikut hasil perhitungan dari peramalan setiap periode sampai dengan periode ke-22 atau bulan April 2020.

Tabel 4.2 Hasil Peramalan cream malam sw2.

Periode	Jumlah Penjualan	Hasil Peramalan
Juli 2018	16	-
Agustus 2018	16	16
September 2018	15	16
Oktober 2018	22	16
November 2018	8	18
Desember 2018	26	15
Januari 2019	17	18
Februari 2019	42	18
Maret 2019	25	25
April 2019	40	26
Mei 2019	47	31

Juni 2019	47	37
Juli 2019	9	41
Agustus 2019	29	34
September 2019	20	34
Oktober 2019	22	31
November 2019	17	29
Desember 2019	6	26
Januari 2020	16	20
Februari 2020	12	19
Maret 2020	7	16
April 2020	10	13
Mei 2020	-	11

Berikut hasil grafik setelah dilakukan peramalan dengan metode *Triple Exponential Smoothing*.



Gambar 4.2 Hasil aktual dan peramalan transaksi produk

4.7.1 Menghitung Kesalahan Peramalan

Untuk mengukur tingkat akurasi dan error terhadap peramalan yang telah dilakukan, dalam penelitian ini menggunakan tiga metode pengukuran yaitu MAD (Mean Absolute Deviation), MSE (Mean Square Error), dan MAPE (Mean Absolute Percent Error). Dari ketiga metode ini nantinya akan dihitung nilai rata-

rata sehingga akan memperoleh nilai kesalahan peramalan secara total.. Berikut contoh hasil perhitungan nilai kesalahan dari metode *Triple Exponential Smoothing* berdasarkan peramalan yang telah dilakukan terhadap produk Cream Malam SW2.

$$\begin{aligned}
 MAD &= \frac{(16-16)+(15-16)+(22-16)+(8-18)+\dots(10-13)}{21} \\
 &= 10.048 \\
 MSE &= \frac{(16-16)^2+(15-16)^2+(22-16)^2+(8-18)^2+\dots(10-13)^2}{21-1} \\
 &= 171.95 \\
 MAPE &= \frac{\left(\frac{(16-16)}{16} \cdot 100\right) + \left(\frac{(15-16)}{15} \cdot 100\right) + \left(\frac{(22-16)}{22} \cdot 100\right) + \left(\frac{(8-18)}{8} \cdot 100\right) + \dots \left(\frac{(10-13)}{10} \cdot 100\right)}{21} \\
 &= 73.324 \\
 \text{Rata - rata kesalahan} &= \frac{10.048 + 171.95 + 73.324}{3} = 85,107
 \end{aligned}$$

Dari hasil perhitungan nilai kesalahan, metode triple exponential smoothing memiliki nilai kesalahan sebesar 85,107 dengan nilai alpha 0.1. Hasil dari perhitungan diatas hanya menghitung satu produk yaitu cream malam sw2. Jika menghitung produk yang lain, maka hasil dari nilai kesalahan pun mengikuti.

4.8 Pengembangan Sistem

Pengembangan sistem dilakukan dengan menggunakan metode *waterfall*. Metode *waterfall* dimulai dengan melakukan analisis untuk menggali informasi mengenai sistem yang akan dikembangkan dan setelah itu dilanjutkan dengan menganalisis kebutuhan dari sistem, baik kebutuhan fungsional maupun kebutuhan non-fungsional, kemudian dari kebutuhan yang telah dianalisis dibuat desain yang mencakup kerangka sistem antara lain perancangan basis data dan perancangan UML, serta dalam langkah ini dibuatkan desain *interface* untuk mendapatkan gambaran tentang aplikasi yang akan dibuat. Tahap selanjutnya yaitu proses *developing* atau pengkodean sistem sesuai dengan desain sistem yang telah dibuat. Setelah proses *developing* selesai, sistem akan diuji untuk

memastikan bahwa sistem sudah berjalan sesuai dengan yang diharapkan, jika sistem masih belum sesuai maka akan ke tahap selanjutnya yaitu *maintenance*. *Maintenance* yaitu proses analisis, desain sistem, proses pengkodean, proses *testing* akan terus berulang selama proses pengembangan sampai aplikasi berjalan sesuai kebutuhan.

4.8.1 Analisis

Tahap awal yang dilakukan yaitu melakukan analisis mengenai pengembangan dari sistem yang akan dilakukan dan dilanjutkan dengan menentukan kebutuhan dari sistem, kebutuhan dari sistem sendiri dibagi menjadi dua yaitu kebutuhan fungsional dan kebutuhan non-fungsional.

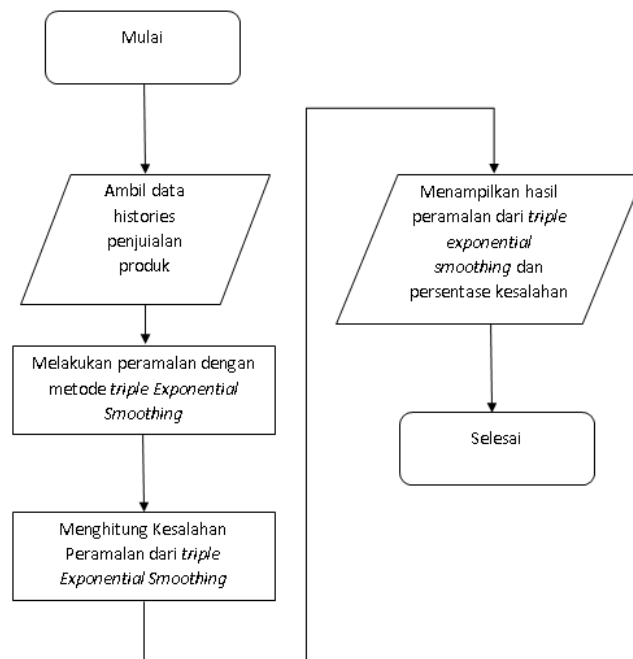
a. Fungsional

- 1) Sistem dapat mengelola data admin.
- 2) Sistem dapat menampilkan semua data produk.
- 3) Sistem dapat menambahkan produk baru.
- 4) Sistem dapat mengedit data produk.
- 5) Sistem dapat menghapus data produk.
- 6) Sistem dapat menambahkan data transaksi produk.
- 7) Sistem dapat menampilkan data transaksi produk yang ditambahkan.
- 8) Sistem dapat menghapus data transaksi produk yang ditambahkan.
- 9) Sistem dapat mengedit data transaksi produk yang ditambahkan.
- 10) Sistem dapat melakukan peramalan untuk setiap produk.
- 11) Sistem dapat melakukan peramalan untuk beberapa produk dan menampilkan hasil dari metode peramalan.

b. Non-fungsional

- 1) Menggunakan OS Windows 10
- 2) Kebutuhan RAM minimal 4 GB
- 3) Processor minimal intel core i3
- 4) Kebutuhan harddisk 100 GB
- 5) Kecepatan internet minimal 1 Mbps

Pada tahap selanjutnya yaitu menggambarkan alur berjalannya sistem yang akan diimplementasikan ke sistem. Fungsinya yaitu untuk mempermudah proses pengembangan sistem, sistem harus digambarkan ke dalam bentuk yang lebih sederhana agar lebih mudah dipahami.



Gambar 4.3 Flowchart Sistem

Alur dari sistem yang akan dikembangkan yaitu dimulai dengan mengambil data historis penjualan produk dari database, kemudian sistem akan melakukan proses peramalan dengan menggunakan metode *Triple Exponential Smoothing*, setelah proses peramalan selesai dilakukan, maka sistem akan menghitung tingkat kesalahan dari peramalan dengan metode MAD (*Mean Absolute Deviation*), MSE (*Mean Absolute Error*), dan MAPE (*Mean Absolute Percent Error*). Selanjutnya sistem akan menampilkan hasil dari peramalan dan tingkat kesalahan peramalan yang kemudian dapat dijadikan sebagai acuan untuk melakukan pentuan stok produk di periode selanjutnya.

4.8.2 Desain

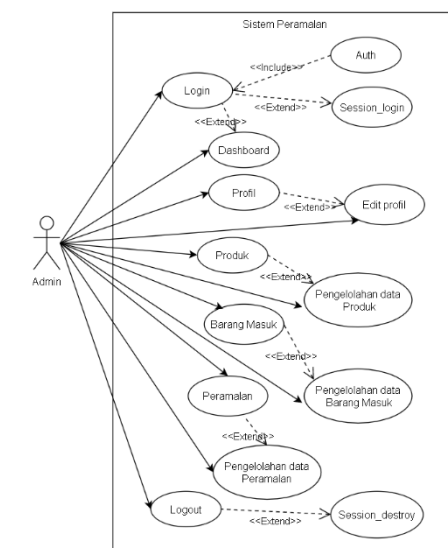
Proses desain berfokus pada perancangan sistem dan pemodelan arsitektur sistem. Pada tahap ini penulis melakukan perancangan system dengan menggunakan *Unified Modelling Language* (UML) dan perancangan basis data.

a. UML (*Unified Modelling Language*)

Penulis menggunakan dua diagram UML, yaitu *Use Case Diagram* dan *Activity Diagram*.

1) *Use Case Diagram*

Use case diagram adalah diagram *use case* yang digunakan untuk menggambarkan secara ringkas siapa yang menggunakan sistem dan apa saja yang bisa dilakukannya. *Use case class* digunakan untuk memodelkan dan menyatakan unit fungsi/layanan yang disediakan oleh sistem ke pemakai. *Use Case diagram* dapat dilihat pada gambar 4.3

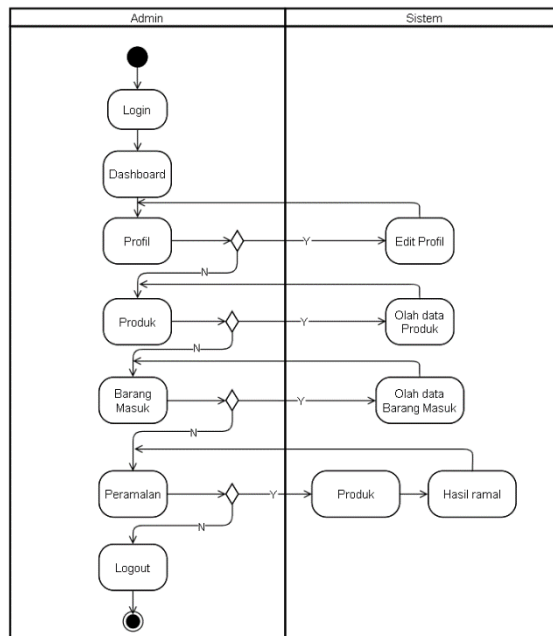


Gambar 4.4 Use Case Diagram

2) *Activity Diagram*

Activity Diagram merupakan rancangan aliran aktivitas atau aliran kerja dalam sebuah sistem yang akan dijalankan. *Activity Diagram* juga

digunakan untuk mendefinisikan atau mengelompokkan aluran tampilan dari sistem tersebut. *Activity diagram* dapat dilihat pada gambar 4.5



Gambar 4.5 Activity Diagram

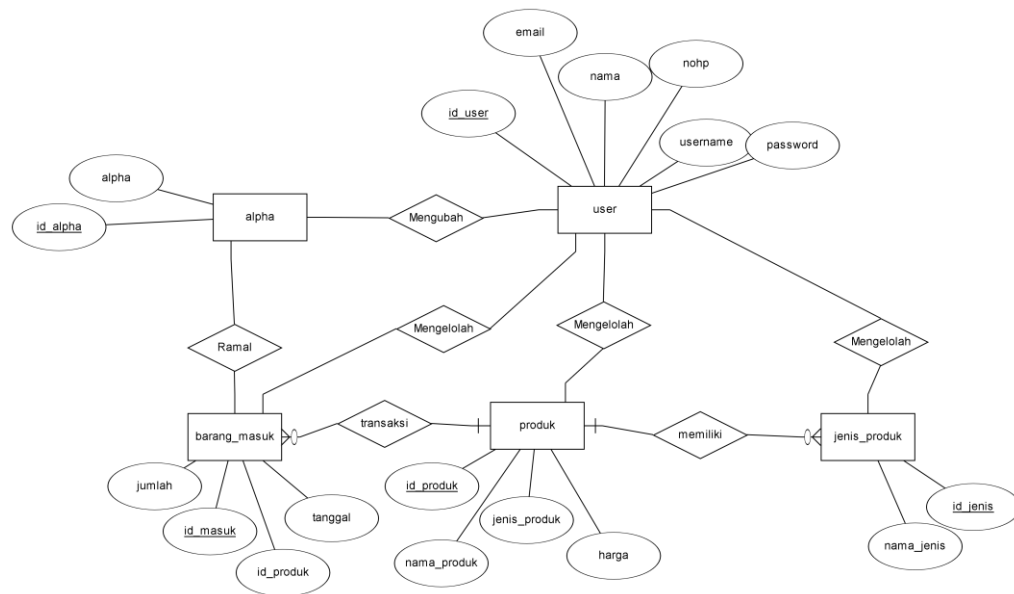
Dijelaskan pada gambar. Alur dalam melakukan peramalan. Yang pertama, admin harus login terlebih dahulu. Setelah itu, admin diarahkan ke *Dashboard*, pada halaman *dashboard* ditampilkan jumlah transaksi selama penjualan, produk yang paling sering dijual dan jumlah produk yang terdaftar. Setelah itu, jika admin ingin mengubah data profil, maka akan diarahkan ke halaman ubah profil. Lalu, jika admin ingin menambah, melihat, mengubah dan menghapus data produk, maka akan di arahkan ke halaman olah data produk. Kemudian, jika admin ingin menambah, melihat, mengubah dan menghapus data barang masuk, makan akan di arahkan ke olah data barang masuk. Setelah itu, jika admin ingin melakukan peramalan, maka admin harus memilih produk yang akan diramal, lalu hasilnya akan terlihat pada halaman hasil ramal. Jika selesai, maka admin harus logout atau keluar.

b. Perancangan Basis Data

Penulis menggunakan 2 jenis rancangan basis data, yaitu *Entity Relationship Diagram* (ERD) dan *Class Diagram*.

1) *Entity Relationship Diagram (ERD)*

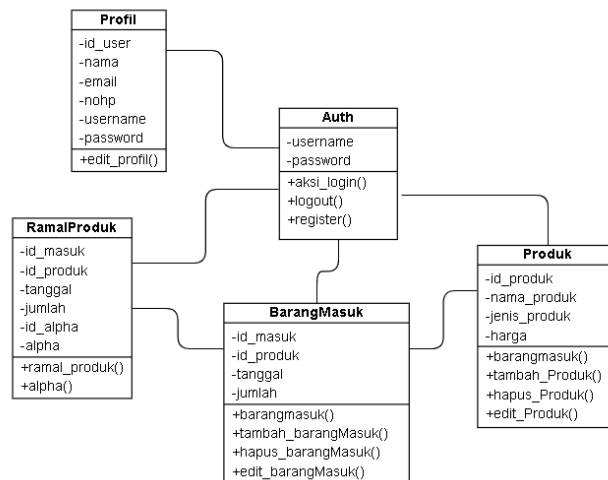
Sebuah model untuk menyusun database agar dapat menggambarkan data yang mempunyai relasi dengan database yang akan didesain. ERD dapat dilihat pada gambar 4.6



Gambar 4.6 ERD

2) *Class Diagram*

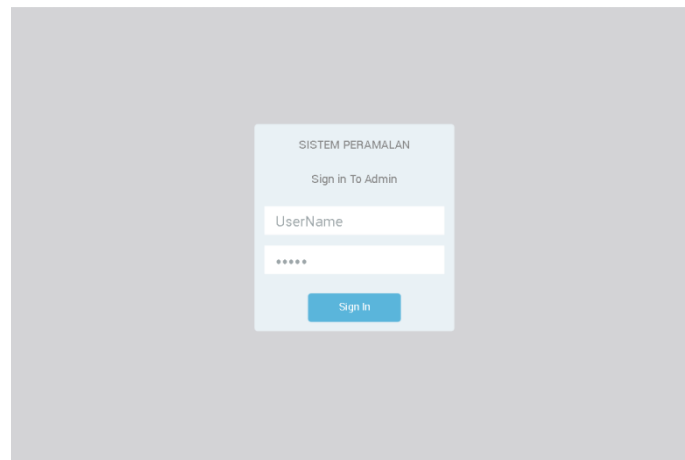
Class diagram adalah visual dari struktur sistem program pada jenis-jenis yang di bentuk. Class diagram merupakan alur jalannya database pada sebuah sistem. *Class diagram* dapat dilihat pada gambar 4.7



Gambar 4.7 Class Diagram

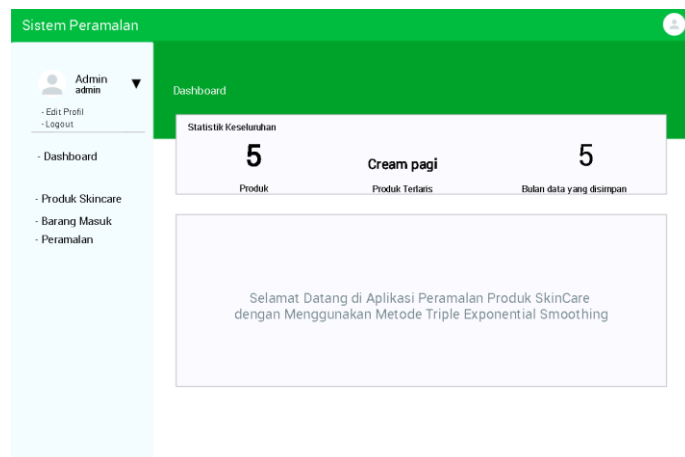
c. *Mock up* Sistem Peramalan.

Tujuan dari pembuatan *mock up* sistem yaitu agar penulis mendapatkan gambaran tentang aplikasi yang akan dibuat seperti alur sistem aplikasi dan desain *interface*. Hasil dari *mock up* sistem mencakup halaman login, halaman profil, halaman produk, halaman barang masuk, dan halaman peramalan. *Mock up* login dapat dilihat pada gambar 4.8, sedangkan *Mock up dashboard* dapat dilihat pada gambar 4.9, lalu *Mock up* ubah profil dapat dilihat pada gambar 4.10, *Mock up* produk dapat dilihat pada gambar 4.11, *Mock up* barang masuk dan sorting barang masuk dapat dilihat pada gambar 4.12 dan gambar 4.13 . untuk *Mock up* peramalan dan hasil ramal dapat dilihat pada gambar 4.14 dan 4.15.



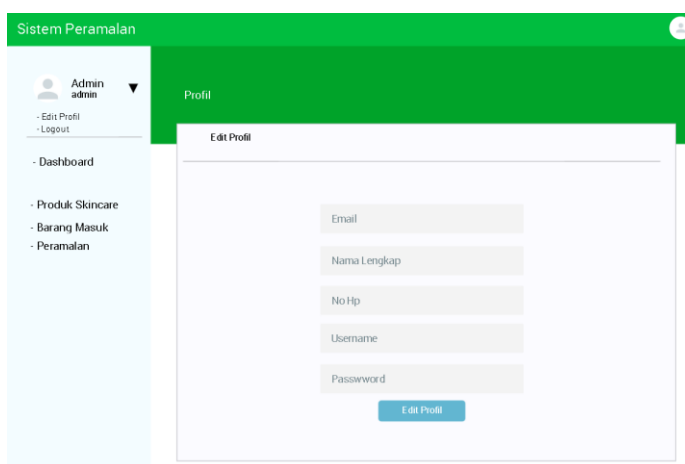
A login form titled "SISTEM PERAMALAN" with the subtitle "Sign in To Admin". It features two input fields: "UserName" and a password field represented by six asterisks. A blue "Sign in" button is positioned below the password field. The form is centered on a light gray background.

Gambar 4.8 Mock Up login



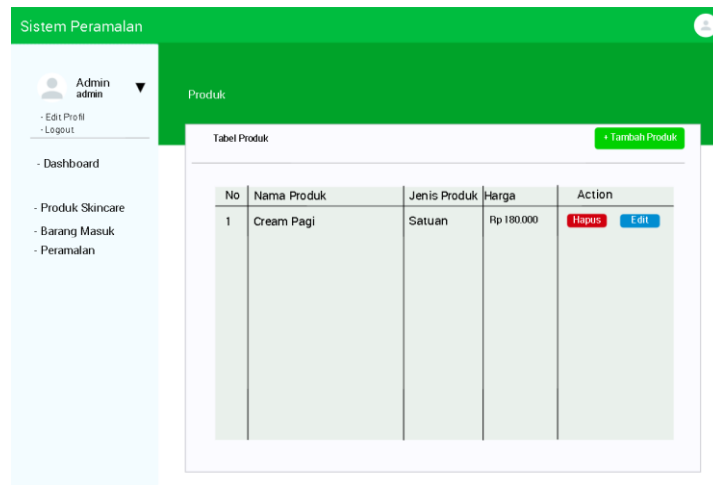
A dashboard interface for "Sistem Peramalan". The left sidebar contains a user profile for "Admin admin" with options to "Edit Profil" and "Logout", and a menu with "Dashboard", "Produk Skincare", "Barang Masuk", and "Peramalan". The main content area has a green header and a "Dashboard" title. It displays "Statistik Keseluruhan" with three metrics: "5 Produk", "Cream pagi", and "5 Produk Tertaris", with a note "Bulan data yang disimpan". Below this is a large box with the text: "Selamat Datang di Aplikasi Peramalan Produk SkinCare dengan Menggunakan Metode Triple Exponential Smoothing".

Gambar 4.9 Mock Up Dashboard

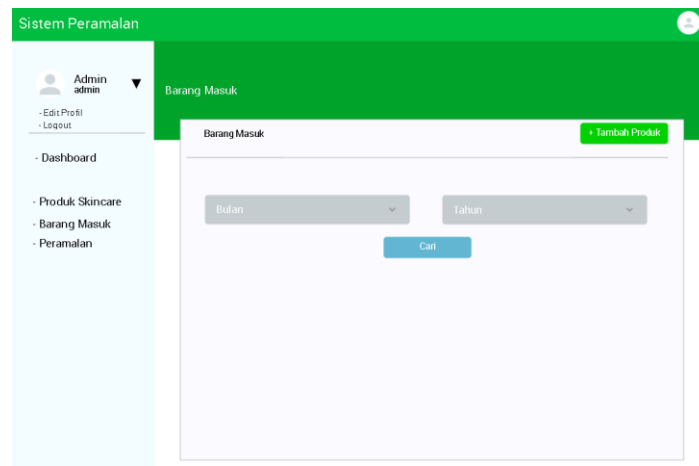


A profile editing form titled "Sistem Peramalan" with the subtitle "Profil". The left sidebar is identical to the dashboard. The main content area has a green header and a "Profil" title. It displays an "Edit Profil" form with input fields for "Email", "Nama Lengkap", "No Hp", "Username", and "Password". A blue "Edit Profil" button is at the bottom right of the form.

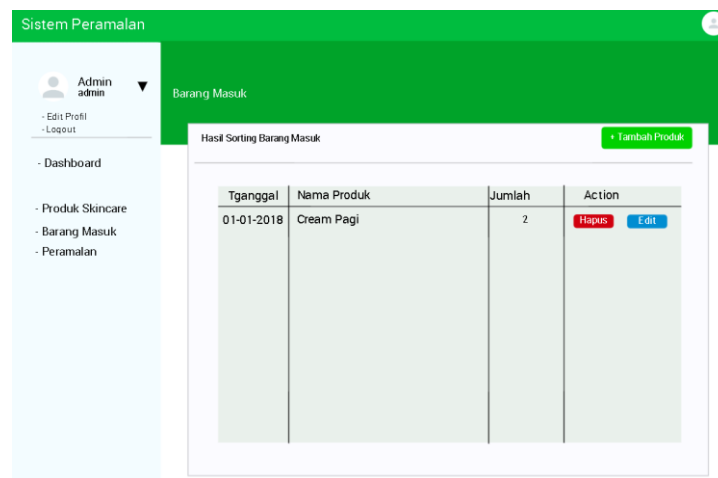
Gambar 4.10 Mock Up Profil



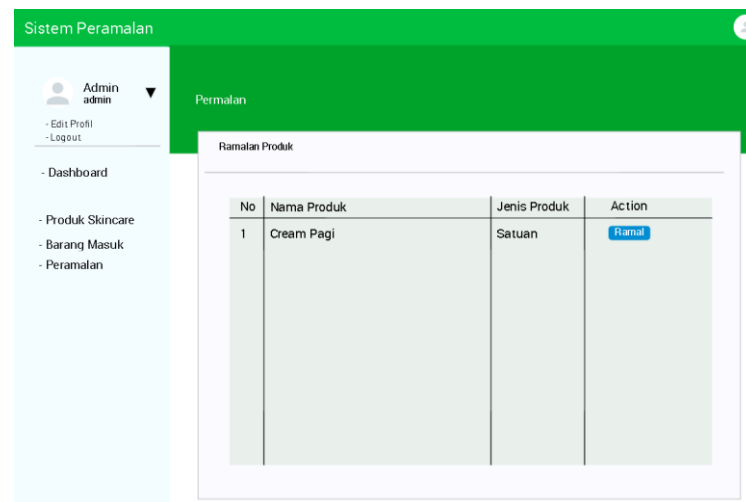
Gambar 4.11 Mock Up Produk



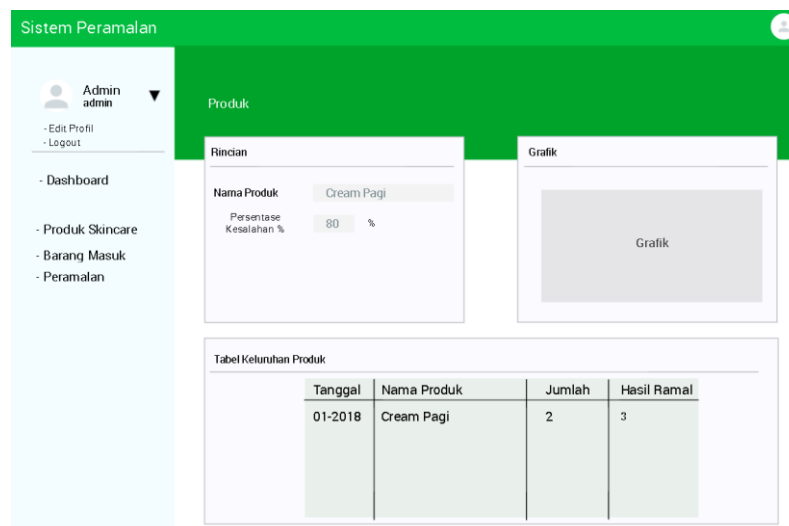
Gambar 4.12 Mock Up Barang masuk



Gambar 4.13 Mock Up Sorting Barang Masuk



Gambar 4.14 Mock Up Ramal Produk



Gambar 4.15 Mock Up hasil ramal

4.8.3 Implementasi

a. Halaman login

Pada halaman login admin harus memasukkan *username* dan *password* yang sudah terdaftar. Jika *username* dan *password* tidak sesuai dengan data yang disimpan di database, maka akan dikembalikan ke halaman login dan menampilkan notifikasi gagal login. Pada tahap login, *username* dan *password* diinput pada baris 24 dan 25, lalu dicek terlebih dahulu melalui database pada baris 27 sampai 30. Setelah itu, jika *username* dan *password* sama dengan database, maka membuat data *session* meliputi *id_user*, *nama*, *email*, *username*,

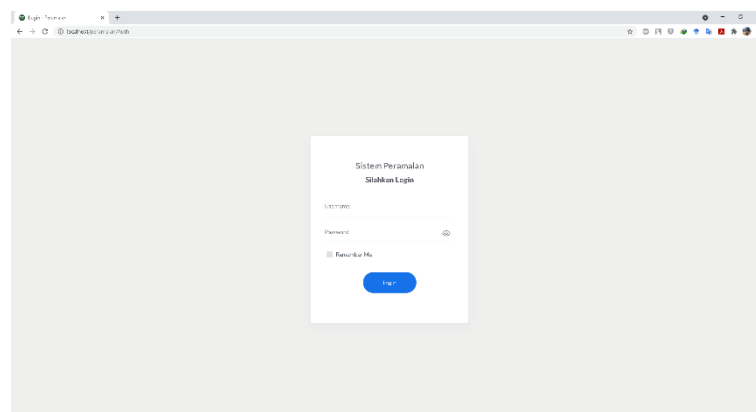
dan status. Source code dapat dilihat pada gambar 4.16 dan halaman login dapat dilihat pada gambar 4.17.

```

23 function aksi_login(){
24     $username = $this->input->post('username');
25     $password = $this->input->post('password');
26     $where = array(
27         'username' => $username,
28         'password' => md5($password)
29     );
30     $data= $this->M_login->cek_login("user",$where)->row();
31     // var_dump($password);
32     // var_dump($data->password);
33
34     $cek = $this->M_login->cek_login("user",$where)->num_rows();
35     if($cek > 0){
36
37         $data_session = array(
38             'id_user'=> $data->id_user,
39             'nama' => $data->nama,
40             'email'=> $data->email,
41             'username' => $username,
42             'status' => "login"
43         );
44
45         $this->session->set_userdata($data_session);
46
47         redirect(base_url('Dashboard'));
48
49     }else{
50         $this->session->set_flashdata('success', 'Password atau Username salah! Silahkan coba lagi!');
51         redirect(base_url('Auth'));
52     }
53 }
54

```

Gambar 4.16 Source code Login



Gambar 4.17 Tampilan Login

b. Halaman Dashboard

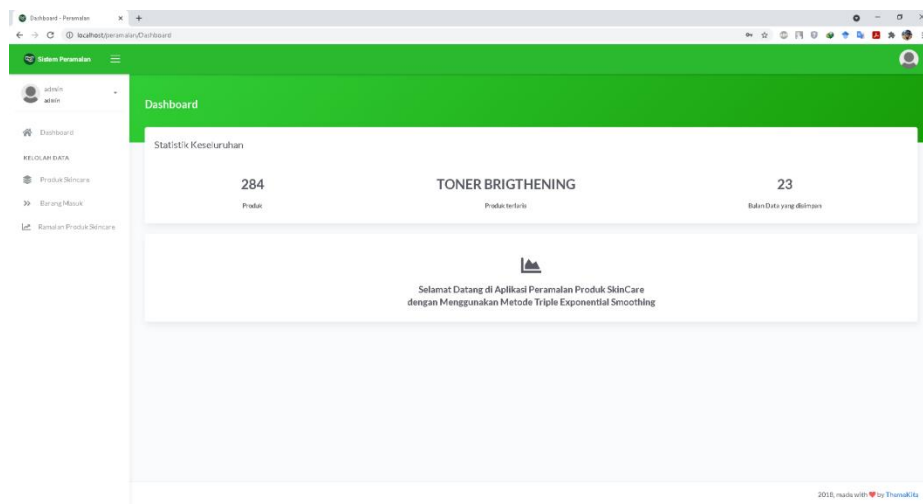
Pada halaman dashboard admin mendapatkan informasi tentang statistik keseluruhan yaitu jumlah produk, produk yang paling sering terjual dan banyaknya bulan yang diramal. Source code dapat dilihat pada gambar 4.18 dan halaman dashboard dapat dilihat pada gambar 4.19

```

public function index()
{
    $detail = $this->db->query("SELECT MONTH(tanggal) as bulan, year(tanggal) as tahun FROM barang_masuk GROUP BY MONTH(tanggal), YEAR(tanggal) ORDER BY tanggal asc")->result();
    $data['jumlah_bulan'] = count($detail);
    $data['jumlah_produk'] = $this->M_produk->jumlah_produk()->row();
    $data['nama_produk'] = $this->M_produk->nama_produk()->row();
    $this->load->view('dashboard', $data);
}

```

Gambar 4.18 Source code Dashboard



Gambar 4.19 Tampilan Dashboard

c. Halaman Profil

Pada halaman profil admin dapat mengubah data diri mereka. Admin dapat mengubah nama, email, nohp, *username* dan *password* sesuai input pada baris 26 sampai 31. Diberis 37, jika input *password* kosong maka password tidak diubah, begitu juga sebaliknya dibari 49. Source code dapat dilihat pada gambar 4.20 dan halaman profil dapat dilihat pada gambar 4.21

```

25 public function edit_profil(){
26     $id_user=$this->input->post('id_user');
27     $nama=$this->input->post('nama');
28     $email=$this->input->post('email');
29     $nohp=$this->input->post('nohp');
30     $username=$this->input->post('username');
31     $password=$this->input->post('password');
32
33     $where = array(
34         'id_user' => $id_user
35     );
36
37     if ($password == "") {
38         $data = array(
39             'nama' => $nama,
40             'email' => $email,
41             'nohp' => $nohp,
42             'username' => $username
43         );
44
45         $this->session->set_flashdata('success', 'Anda Berhasil Mengedit data Profil');
46         $this->M_login->update_data($where,$data,'user');
47         redirect('profil');
48     } else{
49         $data = array(
50             'nama' => $nama,
51             'email' => $email,
52             'nohp' => $nohp,
53             'username' => $username,
54             'password' => md5($password)
55         );
56
57         $this->session->set_flashdata('success', 'Anda Berhasil Mengedit data Profil');
58         $this->M_login->update_data($where,$data,'user');
59         redirect('profil');
60     }
61 }
62 }

```

Gambar 4.20 Source code Profil

Gambar 4.21 Tampilan Profil

d. Halaman Produk

Pada halaman produk, admin dapat mengelolah data produk meliputi melihat keseluruhan produk yang dijual, menambah produk baru, mengubah data produk yang sudah pernah disimpan, dan menghapus data produk. Untuk menambah, mengubah dan menghapus menggunakan modal dengan tujuan agar admin nyaman dengan tampilan *website*. Source code dapat dilihat pada gambar 4.22, 4.23, 4.24, 4.25 dan halaman produk dapat dilihat pada Gambar 4.26

```
//TAMPIL PRODUK
public function index()
{
    $data['produk'] = $this->M_produk->tampil_produk()->result();
    $this->load->view('produk',$data);
}
```

Gambar 4.22 Source code Tampil Produk

```
24 // TAMBAH PRODUK
25 function tambah_produk(){
26
27     $this->form_validation->set_rules('nama_produk', 'Nama Produk', 'required|is_unique[produk.nama_produk]');
28
29     if ($this->form_validation->run()==true) {
30         $nama_produk = $this->input->post('nama_produk');
31         $jenis_produk = $this->input->post('jenis_produk');
32         $harga = $this->input->post('harga');
33
34         $data = array(
35             'nama_produk' => $nama_produk,
36             'jenis_produk' => $jenis_produk,
37             'harga' => $harga
38         );
39
40         $this->session->set_flashdata('success', 'Anda Berhasil Menambah data Produk');
41         $this->M_produk->input_data($data,'produk');
42         redirect('produk');
43     }
44     else
45     {
46         $data['produk'] = $this->M_produk->tampil_produk()->result();
47         $this->load->view('produk',$data);
48     }
49 }
```

Gambar 4.23 Source code Tambah Produk

Pada Gambar 4.23, baris 27 bermaksud jika ada nama produk yang sama maka nama produk tidak diinputkan. Tambah produk meliputi nama produk, jenis produk dan harga produk.

```

58     function edit_produk(){
59         $id_produk = $this->input->post('id_produk');
60         $nama_produk = $this->input->post('nama_produk');
61         $jenis_produk = $this->input->post('jenis_produk');
62         $harga = $this->input->post('harga');
63
64         $data = array(
65             'nama_produk' => $nama_produk,
66             'jenis_produk' => $jenis_produk,
67             'harga' => $harga
68         );
69
70         $where = array(
71             'id_produk' => $id_produk
72         );
73
74         $this->session->set_flashdata('success', 'Anda Berhasil Mengubah data Produk');
75         $this->M_produk->update_data($where,$data,'produk');
76         redirect('produk');
77     }

```

Gambar 4.24 Source code Ubah Produk

Pada Gambar 4.24, baris 74 bermaksud jika ada produk yang telah diubah, maka akan muncul pemberitahuan “Anda berhasil mengubah data produk”. ubah produk meliputi nama produk, jenis produk dan harga produk.

```

79     // HAPUS PRODUK
80     public function hapus_produk($id_produk){
81         $where = array('id_produk' => $id_produk);
82         $this->M_produk->hapus_data($where,'produk');
83         $this->session->set_flashdata('hapus', 'Anda Berhasil Menghapus data Produk');
84         redirect('produk');
85     }

```

Gambar 4.25 Source code Hapus Produk

Pada Gambar 4.25, Untuk menghapus suatu produk, diperlukan id produk. id produk diperoleh berdasarkan memilih produk. Setelah itu, di baris 81 produk dicari dalam database, jika data tersebut ada maka akan secara otomatis terhapus didalam database

Nomor	Nama Produk	Jenis Produk	Harga	Action
1	PAKET SW2 TONER MELAKSA MC AP	Paket	Rp 278,000	[Edit] [Delete]
2	PAKET BODY W5-ITENING COMPLEX 3 IN 1 PH+ CAIR	Paket	Rp 385,000	[Edit] [Delete]
3	PAKET SW2 MK2 AC TONER ACNE CSTR	Paket	Rp 420,000	[Edit] [Delete]
4	PAKET M1 MK2 PEI PS + TONER E-STRONG	Paket	Rp 350,000	[Edit] [Delete]
5	PAKET 2 AC TONER ACNE CSTR	Paket	Rp 420,000	[Edit] [Delete]
6	PAKET 1 SW2 TONER MC AP	Paket	Rp 420,000	[Edit] [Delete]
7	PAKET M1 TONER E-STRONG	Paket	Rp 345,000	[Edit] [Delete]
8	PAKET BAHAN PEMULA HEMAT CAIR	Paket	Rp 270,000	[Edit] [Delete]
9	PAKET BAHAN JUMBO HEMAT CAIR	Paket	Rp 420,000	[Edit] [Delete]
10	PAKET M1 FSTONER E-STRONG	Paket	Rp 265,000	[Edit] [Delete]

Gambar 4.26 Tampilan Produk

e. Halaman Barang Masuk

Pada halaman barang masuk, admin dapat mengelolah data barang meliputi melihat keseluruhan barang yang sudah diambil, menambah transaksi baru, mengubah data transaksi yang sudah pernah disimpan, dan menghapus data transaksi. Untuk menambah, mengubah dan menghapus menggunakan modal dengan tujuan agar admin nyaman dengan tampilan website. Pada awal halaman barang masuk, admin harus terlebih dahulu mencari data berdasarkan bulan dan tahun, tujuannya agar sorting data transaksi mudah untuk dibaca. Source code dapat dilihat pada gambar 4.27, 4.28, 4.29, 4.30 dan halaman barang masuk dapat dilihat pada gambar 4.31 sedangkan halaman rinci barang masuk dapat dilihat pada gambar 4.32

```

public function index()
{
    $data['orderBySatuan'] = $this->M_produk->orderBySatuan()->result();
    $data['orderByPaket'] = $this->M_produk->orderByPaket()->result();
    $data['tahun'] = $this->M_barangMasuk->tahun()->result();
    // $data['barangMasuk'] = $this->M_barangMasuk->tampil_barangMasuk()->result();
    $this->load->view('barangMasuk',$data);
}

public function barang_masuk(){
    $bulan = $this->input->post('bulan');
    $tahun = $this->input->post('tahun');

    // $data['tampilBarangMasuk'] = $this->M_barangMasuk->tampil_barangMasuk()->result();
    $data['produk'] = $this->M_produk->tampil_produk()->row();
    $data['orderBySatuan'] = $this->M_produk->orderBySatuan()->result();
    $data['orderByPaket'] = $this->M_produk->orderByPaket()->result();
    $data['tahun'] = $this->M_barangMasuk->tahun()->result();
    $data['caribarangMasuk'] = $this->M_barangMasuk->rinci_barangMasuk($bulan,$tahun)->result();
    // $data = $this->M_barangMasuk->tampil_barangMasuk()->result();
    $this->load->view('caribarangMasuk',$data);
}

```

Gambar 4.27 Source code Tampil Barang Masuk


```

77 function edit_barangMasuk(){
78     $id_masuk = $this->input->post('id_masuk');
79     $tanggal = $this->input->post('tanggal');
80     $id_produk = $this->input->post('id_produk');
81     $jumlah = $this->input->post('jumlah');
82
83     $data = array(
84         'id_masuk' => $id_masuk,
85         'tanggal' => $tanggal,
86         'id_produk' => $id_produk,
87         'jumlah' => $jumlah
88     );
89
90     $where = array(
91         'id_masuk' => $id_masuk
92     );
93
94     $this->session->set_flashdata('success', 'Anda Berhasil Mengubah data Barang Masuk');
95     $this->M_barangMasuk->update_data($where,$data,'barang_masuk');
96     redirect('barangmasuk');
97
98 }

```

Gambar 4.28 Source code Ubah Barang Masuk

Pada Gambar 4.28, baris 94 bermaksud jika ada produk yang telah diubah, maka akan muncul pemberitahuan “Anda berhasil mengubah data barang masuk”. ubah barang masuk meliputi tanggal, id produk dan jumlah.

```

70 public function hapus_barangMasuk($id_masuk){
71     $where = array('id_masuk' => $id_masuk);
72     $this->M_barangMasuk->hapus_data($where,'barang_masuk');
73     $this->session->set_flashdata('hapus', 'Anda Berhasil Menghapus data Barang Masuk');
74     redirect('barangmasuk');
75 }

```

Gambar 4.29 Source code Hapus Barang Masuk

Pada Gambar 4.29, Untuk menghapus barang masuk, diperlukan id masuk. id produk diperoleh berdasarkan memilih barang yang masuk. Setelah itu, di baris 71 barang masuk dicari dalam database, jika data tersebut ada maka akan secara otomatis terhapus didalam database.

```

53 function tambah_barangMasuk(){
54     $id_produk = $this->input->post('id_produk');
55     $tanggal = $this->input->post('tanggal');
56     $jumlah = $this->input->post('jumlah');
57
58     $data = array(
59         'id_produk' => $id_produk,
60         'tanggal' => $tanggal,
61         'jumlah' => $jumlah
62     );
63
64     $this->session->set_flashdata('success', 'Anda Berhasil Menambah data Barang Masuk');
65     $this->M_barangMasuk->input_data($data,'barang_masuk');
66     redirect('Barangmasuk');
67 }

```

Gambar 4.30 Source code Tambah Barang Masuk

Pada Gambar 4.30, baris 64 bermaksud jika ada barang masuk yang telah diubah, maka akan muncul pemberitahuan “Anda berhasil menambah data barang masuk”. tambah barang masuk meliputi tanggal, id produk dan jumlah.

Gambar 4.31 Tampilan Barang Masuk

Tanggal	Nama Produk	Jumlah	Aksi
01/07/2019	AP1	2	Edit Hapus
01/07/2019	AP1	2	Edit Hapus
04/07/2019	AP2	1	Edit Hapus
05/07/2019	ANIR BIRU	1	Edit Hapus
01/07/2019	CREAM MALAYSIK GOLD	2	Edit Hapus
04/07/2019	CREAM MALAYSIK	1	Edit Hapus
05/07/2019	MEDICEXCELONE DAY CREAM	1	Edit Hapus
01/07/2019	MEDICEXCELONE DAY CREAM	1	Edit Hapus
04/07/2019	STITCH PIGE WHITE LING BCEL	2	Edit Hapus

Gambar 4.32 Tampilan sorting barang masuk

f. Halaman Peramalan

Pada halaman Peramalan, admin dapat memilih produk yang akan diramal. Pada awal halaman peramalan, admin dapat mencari data berdasarkan nama produk, setelah itu halaman hasil peramalan ditampilkan sesuai dengan produk yang dipilih. Hasilnya mencakup nama produk, kesalahan persentase dan hasil ramalan untuk bulan berikutnya. halaman hasil peramalan dapat dilihat pada gambar 4.45

```

463 //SMOOTHING 1
464 $s1[] = $nilaiAwalInt[0];
465 $s1Int = array_map('intval', $s1);
466
467 for($i = 1; $i < count($aktualInt)+1; $i++)
468 {
469     $s1Int[$i] = round($alpha[$a] * $aktualInt[$i-1] + (1-$alpha[$a]) * $s1Int[$i-1], 2);
470 }

```

Gambar 4.33 Source code Hitung S'

Pada Gambar 4.33, Pemulusan pertama dihitung pada baris 469. Setelah melakukan pengulangan, semua hasil disimpan pada array S1Int.

```

472 //SMOOTHING 2
473 $s2[] = $nilaiAwalInt[0];
474 $s2Int = array_map('intval', $s2);
475
476 for($i = 1; $i < count($s1Int); $i++)
477 {
478     $s2Int[$i] = round($alpha[$a] * $s1Int[$i] + (1-$alpha[$a]) * $s2Int[$i-1], 2);
479 }

```

Gambar 4.34 Source code Hitung S''

Pada Gambar 4.34, Pemulusan kedua dihitung pada baris 478. Setelah melakukan pengulangan, semua hasil disimpan pada array S2Int.

```

481 //SMOOTHING 3
482 $s3[] = $nilaiAwalInt[0];
483 $s3Int = array_map('intval', $s2);
484
485 for($i = 1; $i < count($s2Int); $i++)
486 {
487     $s3Int[$i] = round($alpha[$a] * $s2Int[$i] + (1-$alpha[$a]) * $s3Int[$i-1], 2);
488 }

```

Gambar 4.35 Source code Hitung S'''

Pada Gambar 4.35, Pemulusan ketiga dihitung pada baris 487. Setelah melakukan pengulangan, semua hasil disimpan pada array S3Int.

```

490          //AT
491          $at = array();
492          $atInt = array_map('intval',$at);
493
494          for($i = 1;$i < count($s1Int);$i++)
495          {
496              $atInt[$i]= round(3 * $s1Int[$i] - 3 * $s2Int[$i] + $s3Int[$i],2);
497          }

```

Gambar 4.36 Source code Hitung at

Pada Gambar 4.36, setelah semua pemulusan dihitung, selanjutnya menghitung at pada baris 496. Setelah melakukan pengulangan, semua hasil disimpan pada array atInt.

```

499          //BT
500          $bt = array();
501          $btInt = array_map('intval',$bt);
502
503          for($i = 1;$i < count($s1Int);$i++)
504          {
505              $btInt[$i]= round($alpha[$a]/2*pow(1-$alpha[$a],2)*((6-5*$alpha[$a])*$s1Int[$i]-(10-8*$alpha[$a])*$s2Int[$i]+(4-3*$alpha[$a])*$s3Int[$i]),2);
506          }

```

Gambar 4.37 Source code Hitung bt

Pada Gambar 4.37, selanjutnya menghitung bt pada baris 505. Setelah melakukan pengulangan, semua hasil disimpan pada array btInt.

```

508          //CT
509          $ct = array();
510          $ctInt = array_map('intval',$ct);
511
512          for($i = 1;$i < count($s1Int);$i++)
513          {
514              $ctInt[$i]= round(pow($alpha[$a],2)/pow(1-$alpha[$a],2)*($s1Int[$i]-2*$s2Int[$i]+$s3Int[$i]),2);
515          }

```

Gambar 4.38 Source code Hitung ct

Pada Gambar 4.38, selanjutnya menghitung ct pada baris 514. Setelah melakukan pengulangan, semua hasil disimpan pada array ctInt.

```

517          //FT+M
518          $ft = array();
519          $ftInt = array_map('intval',$ft);
520
521          for($i = 1;$i < count($atInt)+1;$i++)
522          {
523              $ftInt[$i]= round($atInt[$i]+$btInt[$i]*1+1/2*$ctInt[$i]*pow(1,2),0);
524          }

```

Gambar 4.39 Source code Hitung ft+m

Pada Gambar 4.39, setelah mendapatkan nilai at, bt dan ct, Selanjutnya menghitung nilai Ft+m pada baris 523. Setelah melakukan pengulangan, semua hasil disimpan pada array ftInt.

```

526         //at-ft
527         $selisih = array();
528         $selisihInt = array_map('intval',$selisih);
529
530         for($i = 1;$i < count($aktualInt);$i++)
531         {
532             $selisihInt[$i] = $aktualInt[$i] - $ftInt[$i];
533         }

```

Gambar 4.40 Source code Hitung at-ft

Pada Gambar 4.40, setelah menghitung nilai $ft+m$, Selanjutnya menghitung nilai $at-ft$ pada baris 532. Setelah melakukan pengulangan, semua hasil disimpan pada array `selisihInt`.

```

535         //(at-ft)2
536         $selisihPangkat = array();
537         $selisihPangkatInt = array_map('intval',$selisihPangkat);
538
539         for($i = 1;$i < count($selisihInt)+1;$i++)
540         {
541             $selisihPangkatInt[$i] = pow($selisihInt[$i],2);
542         }

```

Gambar 4.41 Source code Hitung $(at-ft)^2$

Pada Gambar 4.41, setelah menghitung nilai $at-ft$, Selanjutnya menghitung nilai $(at-ft)^2$ pada baris 541. Setelah melakukan pengulangan, semua hasil disimpan pada array `selisihPangkatInt`.

```

544         //abs((at-ft)/at)*100
545         $selisihSeratus = array();
546         $selisihSeratusInt = array_map('intval',$selisihSeratus);
547
548         for($i = 1;$i < count($selisihInt)+1;$i++)
549         {
550             $selisihSeratusInt[$i] = abs($selisihInt[$i]/$aktualInt[$i]);
551         }

```

Gambar 4.42 Source code Hitung $(at-ft)/at$

Pada Gambar 4.42, setelah menghitung nilai $(at-ft)^2$, Selanjutnya menghitung nilai $(at-ft)/at$ pada baris 550. Setelah melakukan pengulangan, semua hasil disimpan pada array `selisihSeratusInt`.

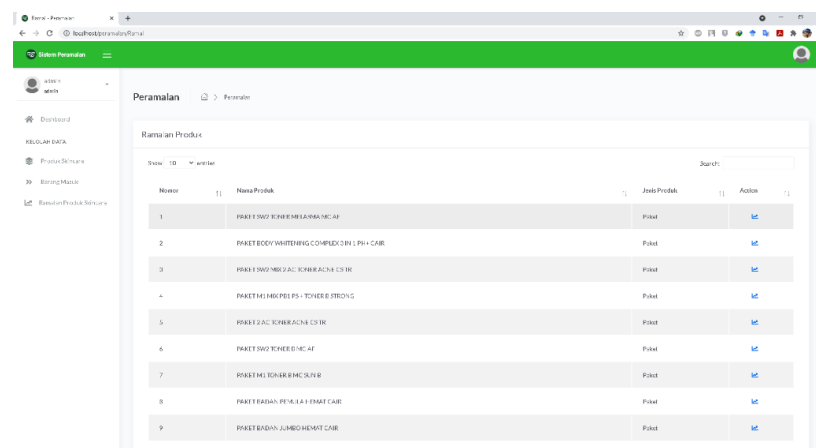
```

553     $jumlah= count(array_slice($aktualInt,1));
554     $MAD = array_sum(array_map("abs",$selisihInt));
555     $MSE = array_sum($selisihPangkatInt);
556     $MAPE = round(array_sum($selisihSeratusInt),3);
557
558     $hasilFtInt[$a] = end($ftInt);
559     $hasilMAD[$a] = round($MAD / $jumlah,3);
560     $hasilMSE[$a] = round($MSE / ($jumlah-1),3);
561     $hasilMAPE[$a]= round($MAPE / $jumlah * 100,3);
562     $hasilKeseluruhan = $hasilMAD[$a] + $hasilMSE[$a] + $hasilMAPE[$a];
563     $rataRataKesalahan[$a] = round($hasilKeseluruhan/3,3);
564 }

```

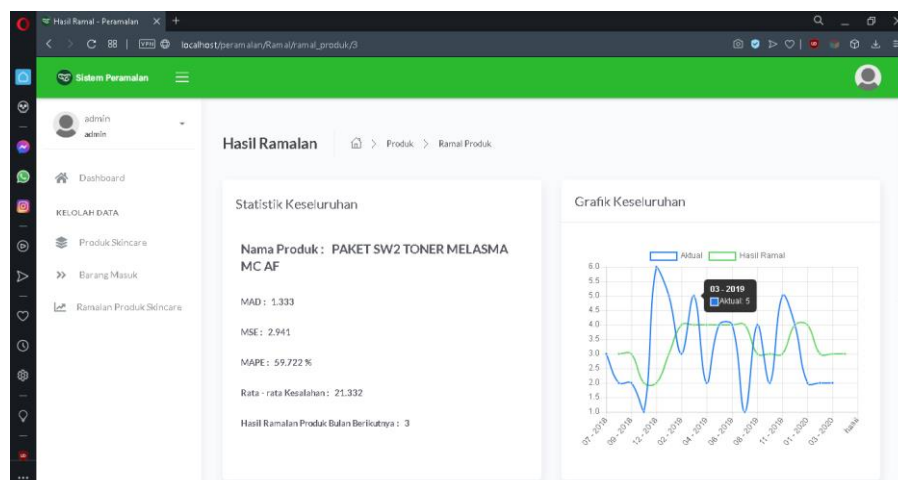
Gambar 4.43 Source code Hitung Kesalahan Peramalan

Pada Gambar 4.43, setelah menghitung peramalan, Selanjutnya menghitung hasil kesalahan peramalan, kesalahan yang dihitung antara lain menggunakan rumus MAD, MSE dan MAPE. Setelah melakukan pengulangan, semua hasil disimpan pada array nama yang sudah tertera pada baris 558 sampai 563.



No	Nama Produk	Jenis Produk	Aksi
1	PAKET SW2 TONER MELASMA MC AF	Paket	Detail
2	PAKET BODY WHITENING COMPLEXION LIFT CARE	Paket	Detail
3	PAKET SW2 TONER MELASMA CS TR	Paket	Detail
4	PAKET MELASMA PPS TONER BROW	Paket	Detail
5	PAKET TONER MELASMA CS TR	Paket	Detail
6	PAKET SW2 TONER MC AF	Paket	Detail
7	PAKET MELASMA BROW	Paket	Detail
8	PAKET BROW PPS LIFT CARE	Paket	Detail
9	PAKET BROW JUNGLE HONEY CARE	Paket	Detail

Gambar 4.44 Tampilan Peramalan



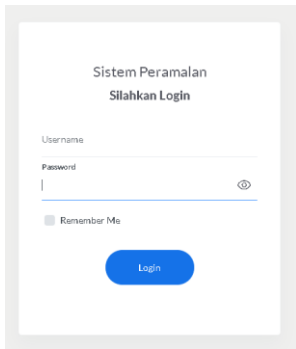
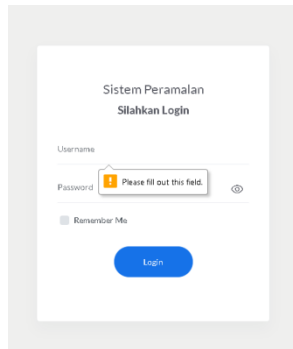


Gambar 4.45 Tampilan Hasil ramal

4.8.4 Pengujian

a. *BlackBox Testing*

Blackbox Testing adalah pengujian yang dilakukan dengan menjalankan modul yang telah dibuat apakah sudah sesuai dengan permintaan atau kondisi sebenarnya, dan melakukan input data sehingga dapat mendapatkan hasil yang sesuai dengan permasalahan.

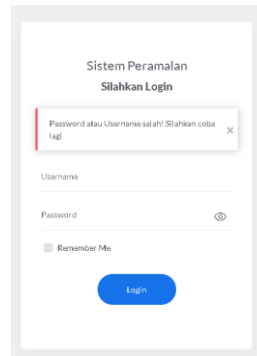
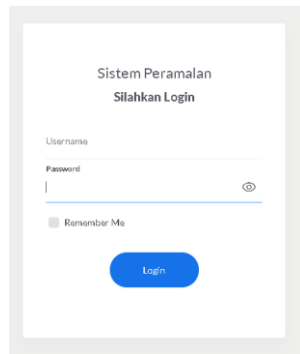
Tabel 4.3 Tabel Pengujian BlackBox Testing pada Login

NO	Skenario Pengujian	Hasil Yang Diharapkan	Kesimpulan
1.	Mengosongkan username dan password, kemudian langsung klik tombol login.	Sistem akan menolak dan meminta untuk memasukkan username dan password.	Valid
Test case :		Hasil :	
			
2	Melihat password dengan klik tombol mata.	Sistem akan menampilkan password yang telah diinputkan.	Valid
Tes case :		Hasil :	
			
3	Memasukkan username dan password yang tidak sesuai.	Sistem akan menolak dan menampilkan pesan "username atau password".	Valid

salah”.

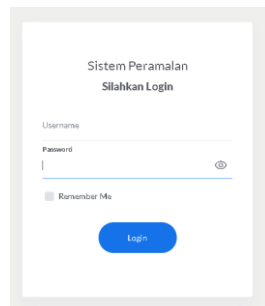
Hasil :

Test case :

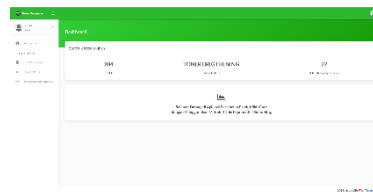


- | | | | |
|---|--|---|-------|
| 4 | Memasukkan username yang terdaftar di sistem dan password yang sesuai. | Sistem menerima akses login dan mengarahkan ke halaman dashboard. | Valid |
|---|--|---|-------|

Tes case :



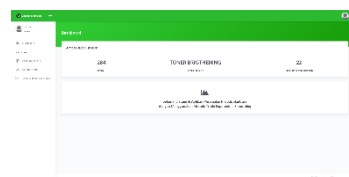
Hasil :



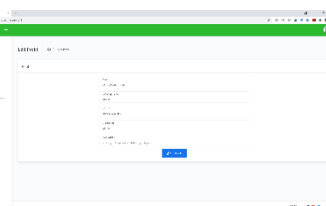
Tabel 4.4 Pengujian BlackBox Testing pada Dashboard

NO	Skenario Pengujian	Hasil Yang Diharapkan	Kesimpulan
1.	Memilih menu edit profil.	Sistem akan mengarahkan ke halaman edit profil.	Valid

Test case :

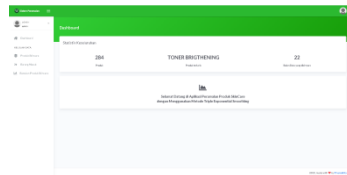


Hasil :

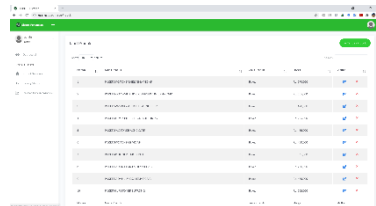


-
- 2 Memilih menu produk. Sistem akan mengarahkan ke Valid halaman produk.

Tes case :

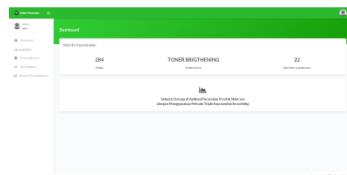


Hasil :

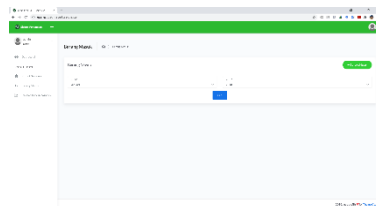


-
- 3 Memilih menu barang masuk. Sistem akan mengarahkan ke Valid halaman barang masuk.

Test case :

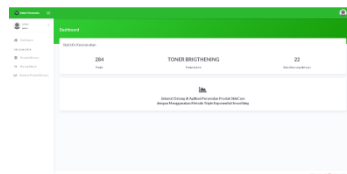


Hasil :

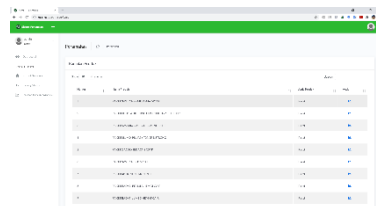


-
- 4 Memilih menu peramalan. Sistem akan mengarahkan ke Valid halaman peramalan.

Tes case :



Hasil :

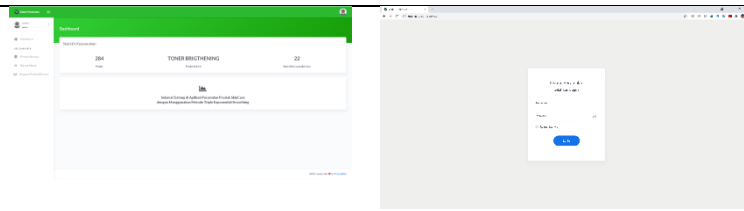


-
- 5 Memilih menu logout. Sistem akan mengarahkan ke Valid halaman login.

Tes case :

Hasil :



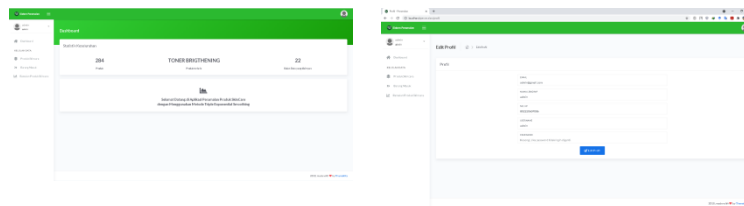


Tabel 4.5 Pengujian BlackBox Testing pada profil

NO	Skenario Pengujian	Hasil Yang Diharapkan	Kesimpulan
1.	Mengubah data profil dengan password.	Sistem akan mengubah password yang disimpan di database.	Valid

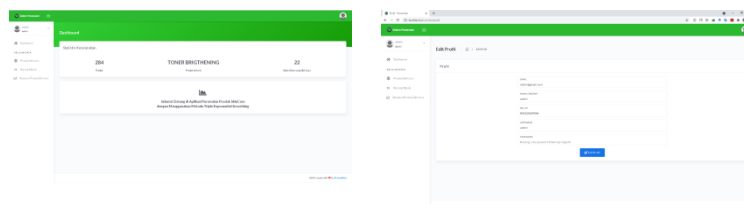
Test case :

Hasil :



Tes case :

Hasil :



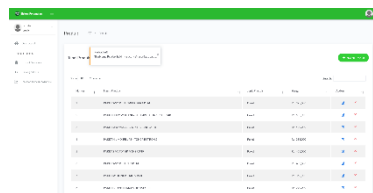
Tabel 4.6 Pengujian BlackBox Testing pada Produk

NO	Skenario Pengujian	Hasil Yang Diharapkan	Kesimpulan
1.	Menambahkan produk dengan nama produk yang sama.	Sistem akan menolak dan menambahkan data.	Valid

Test case :



Hasil :



2	Menambahkan produk dengan nama produk yang berbeda.	Sistem akan menambahkan data produk dan menampilkan notifikasi.	Valid
---	---	---	-------

Test case :



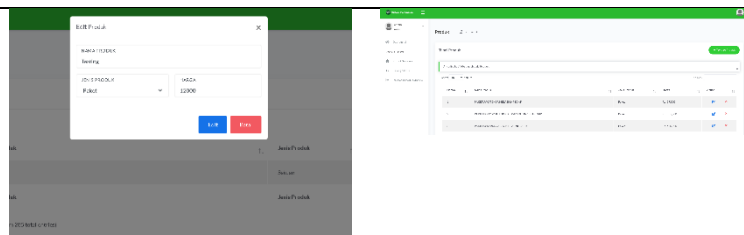
Hasil :



3	Mengubah data produk	Sistem akan mengubah data produk dan menampilkan notifikasi.	Valid
---	----------------------	--	-------

Test case :

Hasil :



- 4 Menghapus data produk Sistem akan menghapus data produk dan menampilkan notifikasi.

Test case :



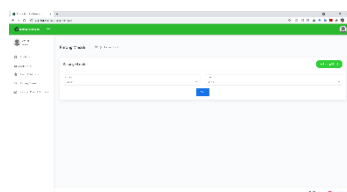
Hasil :



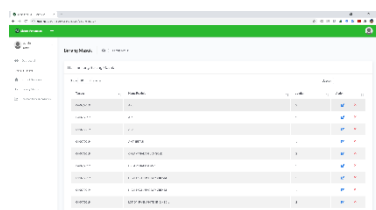
Tabel 4.7 Pengujian BlackBox Testing pada barang Masuk

NO	Skenario Pengujian	Hasil Yang Diharapkan	Kesimpulan
1.	Menampilkan data barang masuk berdasarkan bulan dan tahun.	Sistem akan menampilkan semua transaksi berdasarkan bulan dan tahun yang dipilih.	Valid

Test case :



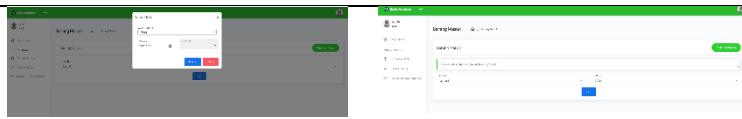
Hasil :



- 2 Menambahkan barang masuk. Sistem akan menambahkan data barang masuk dan menampilkan notifikasi.

Test case :

Hasil :

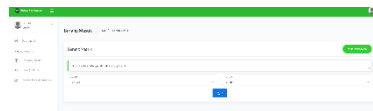


- | | | | |
|---|-----------------------------|--|-------|
| 3 | Mengubah data barang masuk. | Sistem akan mengubah data barang masuk dan menampilkan notifikasi. | Valid |
|---|-----------------------------|--|-------|

Test case :



Hasil :

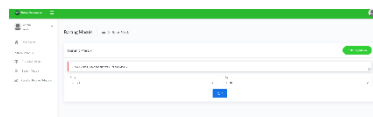


- | | | | |
|---|-----------------------------|---|-------|
| 4 | Menghapus data barang masuk | Sistem akan menghapus data barang masuk dan menampilkan notifikasi. | Valid |
|---|-----------------------------|---|-------|

Test case :

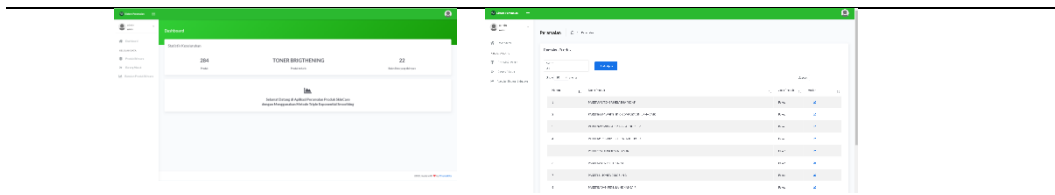


Hasil :



Tabel 4.8 Pengujian BlackBox Testing pada peramalan

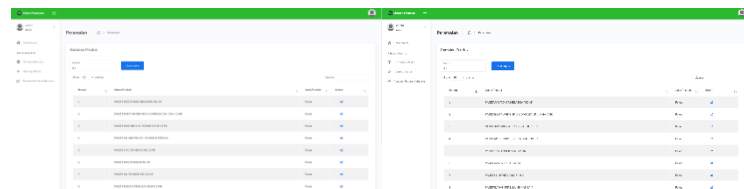
NO	Skenario Pengujian	Hasil Yang Diharapkan	Kesimpulan
1.	Menampilkan produk yang ingin diramalkan	Sistem akan menampilkan semua produk didalam tabel.	Valid
Test case :		Hasil :	



2. Memilih produk yang ingin diramalkan. Sistem akan menampilkan Valid sesuai produk yang ingin diramal meliputi kesalahan permalanan, tabel hasil dan grafik.

Test case :

Hasil :



b. *Usability testing*

Merupakan proses yang paling berkaitan dengan pengguna. Pada pengujian ini pengguna diminta untuk mencoba secara langsung sistem peramalan yang sudah dibuat. Tujuannya adalah untuk mengetahui tingkat kemudahan serta tingkat pemahaman terhadap alur dari sistem tersebut. Pengujian dilakukan langsung oleh distributor Personal Beauty Jember. Hasil *Usability Testing* pada Lampiran 5.

4.9 Hasil dan Pembahasan

Berbeda dengan pengujian sebelumnya, pengujian kali ini berfokus kepada hasil dari peramalan yang dilakukan oleh sistem, pengujian ini dilakukan untuk mengukur performa dari sistem dalam melakukan peramalan. Pengujian dilakukan dengan mengambil sejumlah sampel dari daftar produk untuk diujikan.

Pengujian dilakukan dengan menggunakan indikator kesalahan MAPE, karena dianggap paling relevan dalam mengukur tingkat kesalahan dari sebuah peramalan. Dari hasil pengujian pada produk Cream Malam SW2 dengan alpha yang digunakan 0.1 sampai 0.9, rata-rata kesalahan MAPE yang paling terkecil dari metode triple exponential smoothing adalah sebesar 57.771%, artinya rata-rata akurasi yang diperoleh dari metode triple exponential smoothing mencapai 42.229% dengan nilai alpha 0.3. Setiap Produk memiliki nilai alpha yang berbeda-beda sesuai dengan pola datanya.