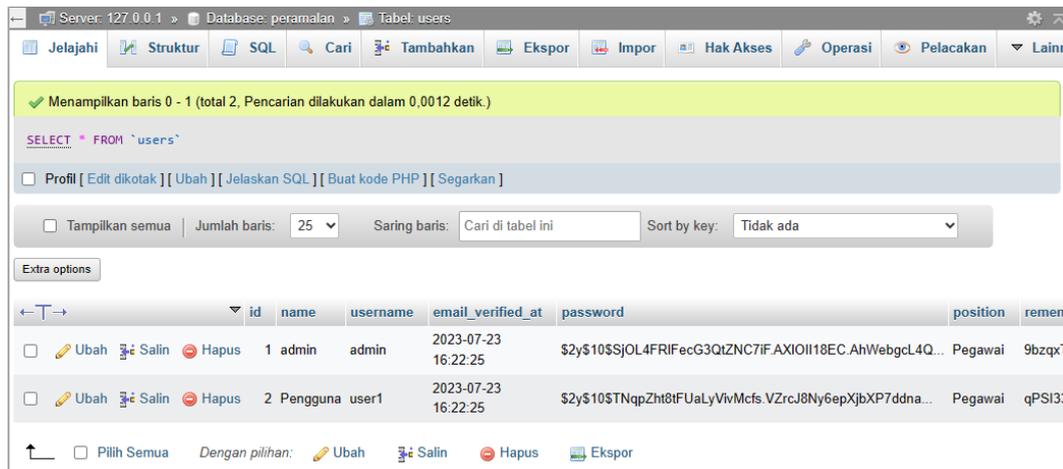


BAB V. IMPLEMENTASI DAN PENGUJIAN

5.1 Implementasi Basis Data

Basis data atau Database adalah Kumpulan informasi yang disimpan di dalam komputer secara sistematis sehingga dapat diperiksa menggunakan suatu program komputer untuk memperoleh informasi dari basis data tersebut (Andaru, 2018). Dalam pembuatan dan pengolahan database untuk Sistem Peramalan Penjualan Pakan Ternak, peneliti menggunakan MySQL. Berikut merupakan hasil implementasi desain database dari sistem peramalan sebagai berikut:

5.1.1 Tabel Users



Menampilkan baris 0 - 1 (total 2, Pencarian dilakukan dalam 0,0012 detik.)

```
SELECT * FROM `users`
```

Profil [Edit dikotak] [Ubah] [Jelaskan SQL] [Buat kode PHP] [Segarkan]

Tampilkan semua | Jumlah baris: 25 | Saring baris: Cari di tabel ini | Sort by key: Tidak ada

Extra options

	id	name	username	email_verified_at	password	position	remember_token
<input type="checkbox"/>	1	admin	admin	2023-07-23 16:22:25	\$2y\$10\$SjOL4FRIFecG3QIZNC7IFAXIOI118EC.AhWebgcL4Q...	Pegawai	9bzqxT
<input type="checkbox"/>	2	Pengguna	user1	2023-07-23 16:22:25	\$2y\$10\$TNqpZht8tFUaLyVivMcfS.VZrcJ8Ny6epXjbXP7ddna...	Pegawai	qPSi33

Pilih Semua Dengan pilihan: Ubah Salin Hapus Ekspor

Gambar 5. 1 Tabel User

Tabel User pada gambar 5.1 digunakan untuk menyimpan data user yang sudah terdaftar. Di dalam tabel ini berisikan kolom id, name, username, email_verified_at, password, position, remember token, created_at, updated_at.

5.1.2 Tabel Role

Profil [[Edit dikotak](#)] [[Ubah](#)] [[Jelaskan SQL](#)] [[Buat kode PHP](#)] [[Segarkan](#)]

Tampilkan semua | Jumlah baris: 25 | Saring baris: Cari di tabel ini | Sort by key:

Extra options

	id	name	guard_name	created_at	updated_at
<input type="checkbox"/> Ubah Salin Hapus	1	Super Admin	web	2023-07-23 16:22:24	2023-07-23 16:22:24
<input type="checkbox"/> Ubah Salin Hapus	2	User	web	2023-07-23 16:22:24	2023-07-23 16:22:24

Pilih Semua Dengan pilihan: [Ubah](#) [Salin](#) [Hapus](#) [Ekspor](#)

Gambar 5. 2 Tabel Role

Tabel Role pada gambar 5.2 digunakan untuk memberikan role pada user. Di dalam table Role ini berisikan kolom id, name, guard_name, created_at, update_at.

5.1.3 Tabel Data Aktual

	id	price	price_date	created_at	updated_at
<input type="checkbox"/> Ubah Salin Hapus	1	5415.16	2018-01-01	2023-07-23 16:22:26	2023-07-23 16:22:26
<input type="checkbox"/> Ubah Salin Hapus	2	5207.47	2018-02-01	2023-07-23 16:22:26	2023-07-23 16:22:26
<input type="checkbox"/> Ubah Salin Hapus	3	4757.03	2018-03-01	2023-07-23 16:22:26	2023-07-23 16:22:26
<input type="checkbox"/> Ubah Salin Hapus	4	4556.33	2018-04-01	2023-07-23 16:22:26	2023-07-23 16:22:26
<input type="checkbox"/> Ubah Salin Hapus	5	4554.47	2018-05-01	2023-07-23 16:22:26	2023-07-23 16:22:26
<input type="checkbox"/> Ubah Salin Hapus	6	4650.23	2018-06-01	2023-07-23 16:22:26	2023-07-23 16:22:26
<input type="checkbox"/> Ubah Salin Hapus	7	4632.64	2018-07-01	2023-07-23 16:22:26	2023-07-23 16:22:26
<input type="checkbox"/> Ubah Salin Hapus	8	4774.09	2018-08-01	2023-07-23 16:22:26	2023-07-23 16:22:26
<input type="checkbox"/> Ubah Salin Hapus	9	4888.87	2018-09-01	2023-07-23 16:22:26	2023-07-23 16:22:26
<input type="checkbox"/> Ubah Salin Hapus	10	4936.7	2018-10-01	2023-07-23 16:22:26	2023-07-23 16:22:26
<input type="checkbox"/> Ubah Salin Hapus	11	5116.23	2018-11-01	2023-07-23 16:22:26	2023-07-23 16:22:26
<input type="checkbox"/> Ubah Salin Hapus	12	5236.63	2018-12-01	2023-07-23 16:22:26	2023-07-23 16:22:26
<input type="checkbox"/> Ubah Salin Hapus	13	5352.85	2019-01-01	2023-07-23 16:22:26	2023-07-23 16:22:26

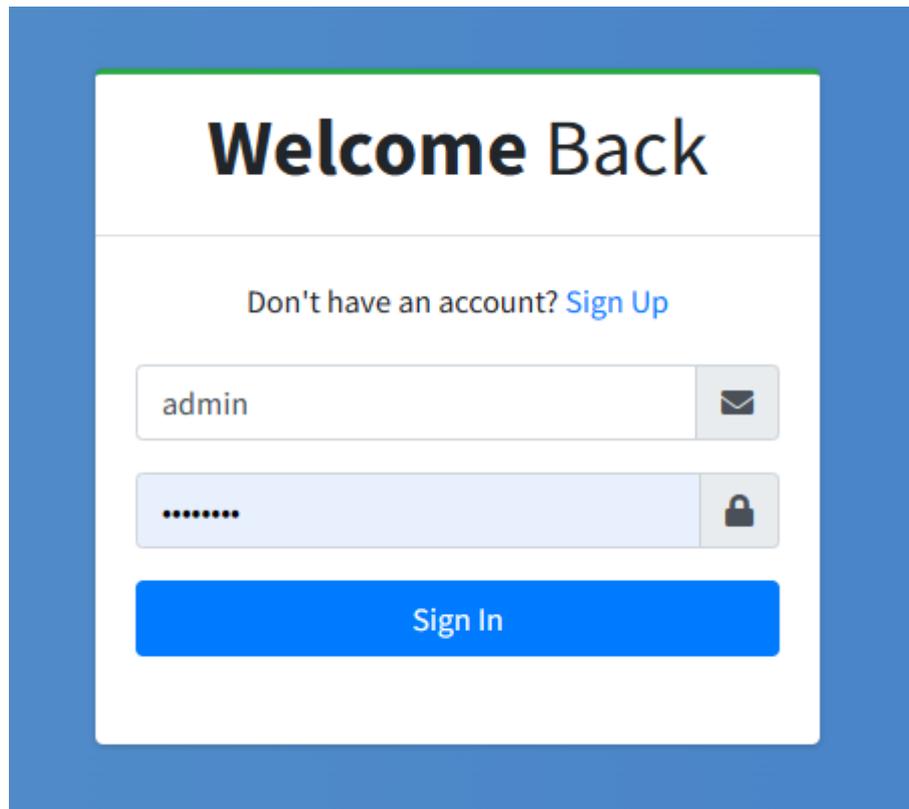
Gambar 5. 3 Tabel Data Aktual

Tabel data aktual pada gambar 5.3 digunakan untuk mengelola data aktual. Di dalam tabel data aktual ini berisikan kolom id, price, price_date, created_at, update_at.

5.2 Implementasi Antarmuka

Implementasi antarmuka pada sistem ini dibuat berdasarkan prinsip mockup yang sudah dibuat sebelumnya. Pada sistem ini dibagi menjadi 2 pengguna, yaitu tampilan pada admin dan pegawai. Berikut ini adalah tampilan antarmuka:

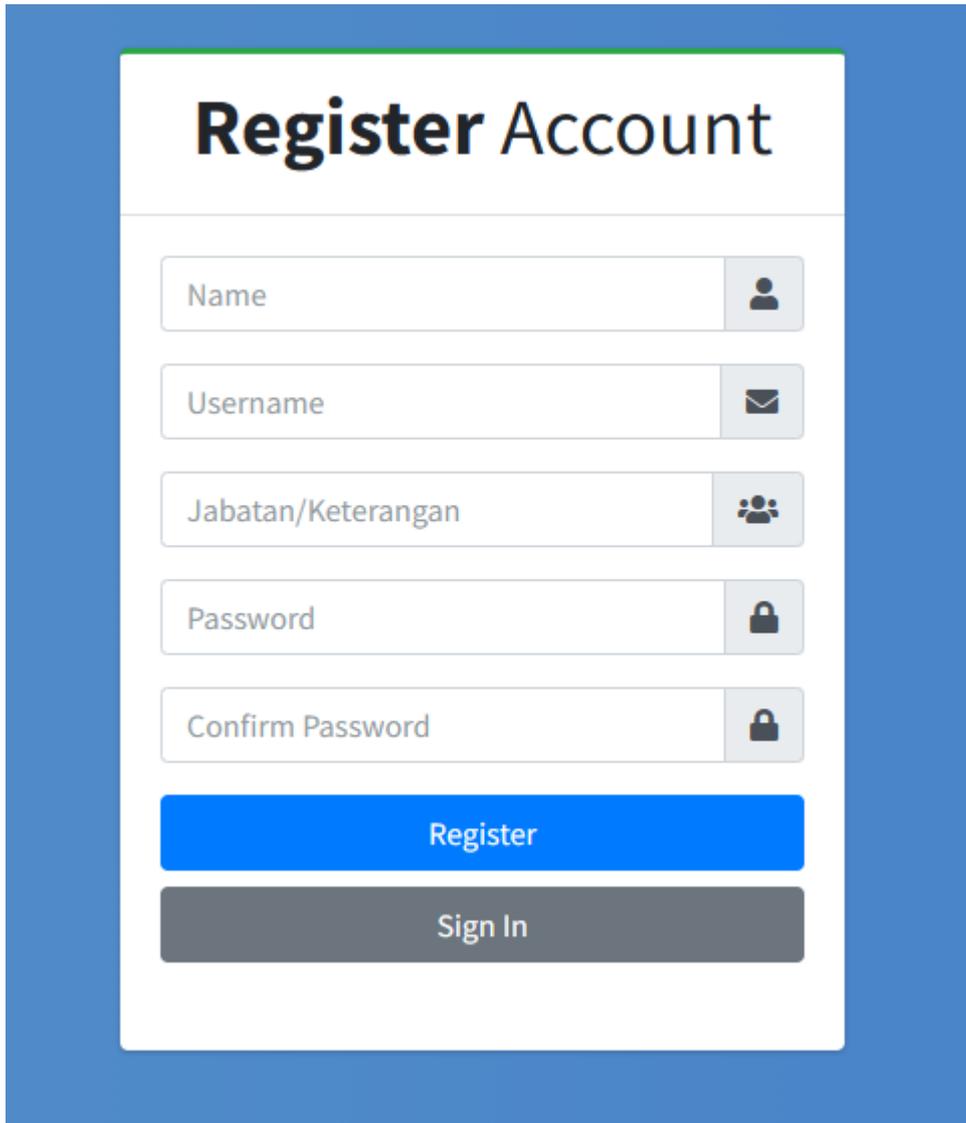
5.2.1 Halaman Login



Gambar 5. 4 Halaman Login

Halaman login adalah halaman yang pertama kali diakses oleh user. Pada halaman login, user akan memasukkan *username* dan *password* lalu klik sign in untuk masuk ke halaman Dashboard.

5.2.2 Halaman Sign Up

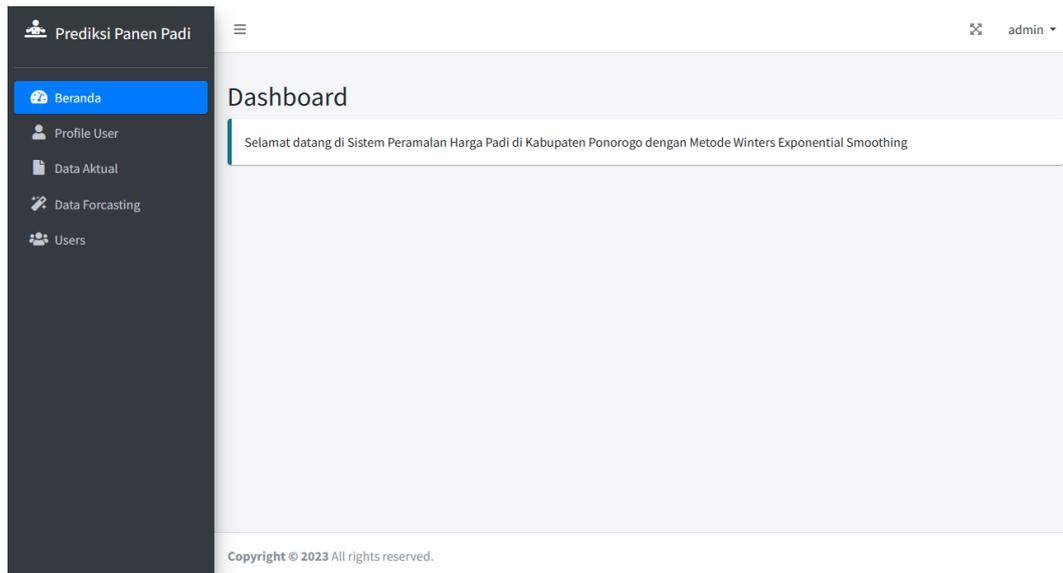


The image shows a 'Register Account' form with a white background and a blue border. The title 'Register Account' is at the top in bold black text. Below the title are five input fields, each with a label and an icon: 'Name' (person icon), 'Username' (envelope icon), 'Jabatan/Keterangan' (group of people icon), 'Password' (lock icon), and 'Confirm Password' (lock icon). At the bottom are two buttons: a blue 'Register' button and a grey 'Sign In' button.

Gambar 5. 5 Halaman sign up

Halaman sign up merupakan halaman untuk menambahkan user baru, di halaman ini terdapat beberapa form seperti form nama, username, jabatan, password, confirm password dan terdapat 2 button register dan sign in

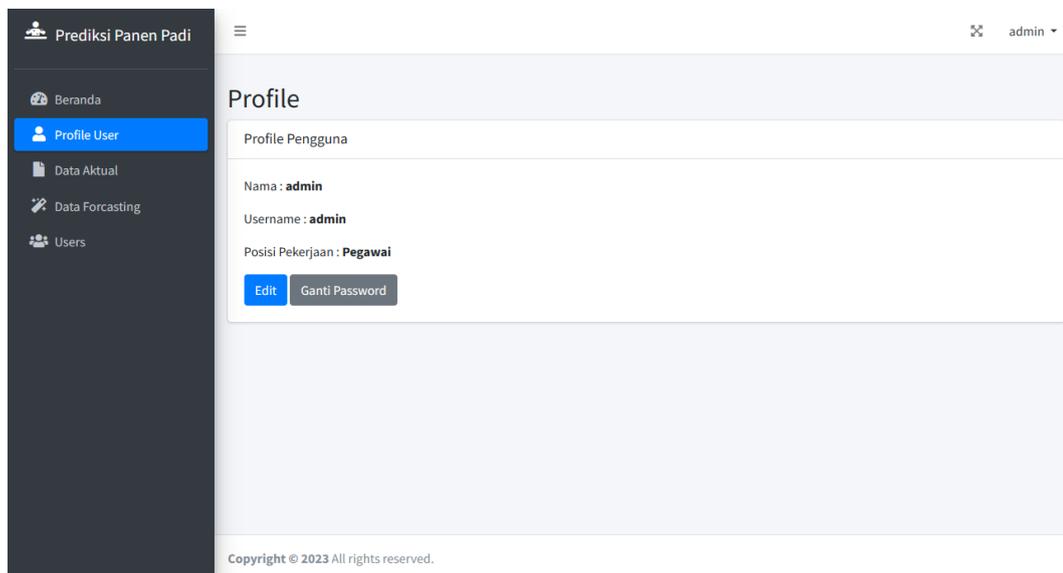
5.2.3 Halaman Dashboard



Gambar 5. 6 Halaman Dashboard

Halaman dashboard adalah halaman pertama yang akan diakses oleh user setelah login dimana terdapat beberapa menu seperti profile user, data aktual, data forecasting, dan user lalu terdapat juga dropdown di pojok kanan diatas untuk logout

5.2.4 Halaman Profile



Gambar 5. 7 Halaman Profile

Halaman profile disini user bisa mengetahui profile akun yang digunakan dan bisa melakukan edit akun dan mengganti password akun tersebut.

5.2.5 Halaman Data Aktual

The screenshot displays the 'Data Aktual' page. The sidebar on the left is dark grey with white text and icons for navigation: Beranda, Profile User, Data Aktual (highlighted in blue), Data Forecasting, and Users. The main content area has a light grey header with the title 'Data Aktual' and a sub-header 'Daftar Data Aktual'. A blue button '+ Tambah data Data Aktual' is in the top right. Below the header, there is a 'Show 12 entries' and a search box. The main part of the page is a table with 7 rows and 7 columns: No, Tahun, Bulan, Harga, Updated At, and Aksi. Each row contains data for the year 2023, with months from June to December. The 'Aksi' column contains a blue 'Action' button with a dropdown arrow.

No	Tahun	Bulan	Harga	Updated At	Aksi
1	2023	December	Rp. 5.624,02	22/7/2023, 02.14.33	Action
2	2023	November	Rp. 5.397,38	22/7/2023, 02.14.33	Action
3	2023	October	Rp. 5.353,97	22/7/2023, 02.14.33	Action
4	2023	September	Rp. 5.141,62	22/7/2023, 02.14.33	Action
5	2023	August	Rp. 4.865,34	22/7/2023, 02.14.33	Action
6	2023	July	Rp. 4.568,84	22/7/2023, 02.14.33	Action
7	2023	June	Rp. 4.538,15	22/7/2023, 02.14.33	Action

Gambar 5. 8 Halaman Data Aktual

Pada halaman ini menampilkan data aktual. Admin dan user dapat melakukan tambah data, edit data, dan hapus data, selain itu terdapat tabel yang berisikan no, tahun, bulan, harga, update at, dan aksi yang di dalam aksi tersebut terdapat 2 pilihan yaitu hapus data dan edit data

5.2.6 Halaman Peramalan

The screenshot displays the 'Forecasting' page. The sidebar on the left is dark grey with white text and icons for navigation: Beranda, Profile User, Data Aktual, Data Forecasting (highlighted in blue), and Users. The main content area has a light grey header with the title 'Forecasting' and a sub-header 'Forecasting (Peramalan)'. Below the header, there is a form titled 'Prediksi Padi Kabupaten'. The form has a 'Tahun' label and a dropdown menu with the text 'Select Option Tahun Value'. A blue 'Submit' button is at the bottom of the form. The footer of the page contains the text 'Copyright © 2023 All rights reserved.'

Gambar 5. 9 Halaman Peramalan

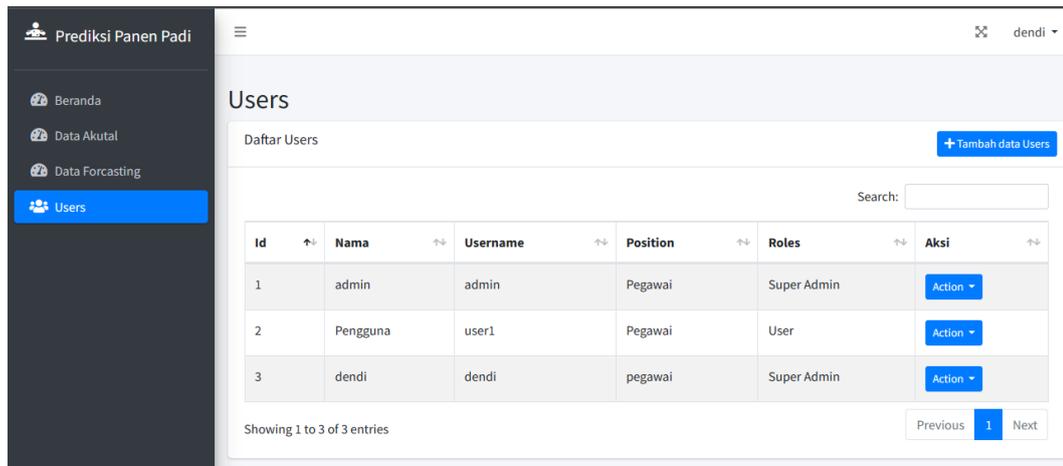
Ramalan (Forcast) Harga Padi Data Tahun 2023

Bulan	Harga	Level	Trend	Seasonal	Forcast	Mape
Februari	5009.67	4527.08	3.93	1.08	4722.51	5.73
Maret	4849.13	4612.86	28.48	1.02	4447.43	8.28
April	4569.86	4684.43	41.41	0.96	4367.16	4.44
Mei	4368.75	4698.13	33.10	0.94	4500.72	3.02
Juni	4461.34	4697.57	23.00	0.96	4625.84	3.69
Juli	4538.15	4724.24	24.10	0.96	4520.60	0.39
Agustus	4568.84	4729.44	18.43	0.97	4661.60	2.03
September	4865.34	4767.28	24.25	1.01	4767.91	2.00
Oktober	5141.62	4848.73	41.41	1.04	4852.06	5.63
November	5353.97	4960.55	62.53	1.05	4994.41	6.72
Desember	5397.38	5048.81	70.25	1.06	5262.62	2.50
Januari	5624.02	5167.31	84.73	1.07	5370.91	4.50

Gambar 5. 10 Hasil Peramalan

Pada halaman ini digunakan untuk melakukan peramalan harga. Dimana user akan memasukkan tahun yang akan diramalkan dan menekan tombol submit maka akan keluar tabel hasil peramalan yang berisikan bulan, harga, level, trend, seasonal, forecast, dan mape.

5.2.7 Halaman User



Gambar 5. 11 Halaman User

Pada halaman ini akan menampilkan data user. Admin dapat melakukan tambah data, edit data, short, dan hapus data pada user, lalu terdapat tabel yang berisikan data user meliputi id, nama, username, posisi, role, dan aksi yang didalam aksi tersebut terdapat tombol edit data dan hapus data

5.3 Implementasi Pengkodean Sistem

5.3.1 User Login



Gambar 5. 12 User Login

Pada gambar 5.12 adalah implementasi kode untuk login, dimana di dalam kode tersebut dijelaskan bahwa jika request data sesuai maka akan di arahkan ke

file manajemen.index dan jika gagal akan menampilkan pesan username/password yang anda masukkan salah.

5.3.2 Kelola Data User

```

1 public function edit(User $user)
2     {
3         $roles = Role::all();
4         return view($this->guard.'.'.$this->prefix.'.edit', [
5             'title' => $this->title,
6             'prefix' => $this->prefix,
7             'guard' => $this->guard,
8             'roles' => $roles,
9             'data' => $user
10        ]);

```

```

1 public function update(Request $request, User $user)
2     {
3         $validated = $request->validate([
4             'username' => "required|string|alpha_dash|unique:users,username, ".$user->id."|min:5",
5             'name' => "required|string|min:5",
6             'position' => 'required|string',
7             'role' => 'required'
8         ]);
9         $user->update($request->except('role'));
10        if(!$user->hasRole(Role::findById($request->get('role')))){
11            $user->assignRole(Role::findById($request->get("role")));
12        }
13
14        $toast = array(
15            'typeToast' => 'success',
16            'messageToast' => 'Success update data '.$this->title,
17        );
18        return redirect()->route($this->guard.'.'.$this->prefix.'.index')->with($toast);
19    }
20
21    /**
22     * Remove the specified resource from storage.
23     *
24     * @param int $id
25     * @return \Illuminate\Http\Response
26     */
27    public function destroy(User $user)
28    {
29        $user->delete();
30        $toast = array(
31            'typeToast' => 'success',
32            'messageToast' => 'Success delete data '.$this->title,
33        );
34        return redirect()->route($this->guard.'.'.$this->prefix.'.index')->with($toast);
35    }

```

Gambar 5. 13 Kelola Data User

Pada gambar 5.13 adalah implementasi kode untuk Kelola Data User, di dalam kode tersebut dijelaskan untuk edit data, delete data, dan update data. Untuk edit data akan merubah pada title, prefix, guard role, dan data. Lalu untuk update data yang di update adalah username, nama, posisi, role jika berhasil update data akan muncul pesan success update data dan jika gagal akan diarahkan Kembali ke .index. dan yang terakhir

5.3.3 Kelola Data Aktual

```

1 public function destroy(ActualData $actualdatum)
2     {
3         $actualdatum->delete();
4         $toast = array(
5             'typeToast' => 'success',
6             'messageToast' => 'Success delete data '.$this->title,
7         );
8         return redirect()->route($this->guard.'.'.$this->prefix.'.index')->with($toast);
9     }

```

```

1 public function store(Request $req)
2     {
3         $req->validate([
4             'price_date' => 'required|date',
5             'price' => 'required',
6         ]);
7         $this->model->create([
8             "price" => $req->get('price'),
9             "price_date" => date_format(date_create($req->get('price_date')), 'Y/m/d')
10        ]);
11        $toast = array(
12            'typeToast' => 'success',
13            'messageToast' => 'Success created data '.$this->title,
14        );
15        return redirect()->route($this->guard.'.'.$this->prefix.'.index')->with($toast);
16    }
17

```

```

1 public function edit(ActualData $actualdatum)
2     {
3         return view($this->guard.'.'.$this->prefix.'.edit', [
4             'title' => $this->title,
5             'prefix' => $this->prefix,
6             'guard' => $this->guard,
7             'data' => $actualdatum
8         ]);
9     }

```

Gambar 5. 14 Kelola Data Aktual

Pada gambar 5.14 adalah implementasi kode untuk data aktual. Pada destroy data dijelaskan bahwa deklarasi data aktual lalu memberikan perintah

delete jika berhasil maka akan muncul pesan success delete data jika tidak akan Kembali ke halaman sebelumnya. Lalu untuk store data yang pertama validasi data berupa price date dan price, lalu membuat model price dan price date, dan jika data berhasil ditambahkan maka akan muncul pesan success create data jika tidak akan Kembali ke halaman sebelumnya. Dan untuk edit terdapat function edit yang berisi title, prefix, guard, dan data

5.3.4 Perhitungan Peramalan

```

1 $datas = array_slice($data, 12, count($data));
2 $monthIndex = 0;
3
4 foreach($datas as $skx => $skv){
5     $level = ($this->alpha * ($skv / $initialSeason[$skx]) + (1-$this->alpha) * ($initialLevel + $initialTrend)); //forecast
6     $trend = ($this->beta * ($level - $initialLevel) + (1-$this->beta) * $initialTrend); //forecast
7     $season = ($this->gamma * ($skv / $level) + (1-$this->gamma) * $initialSeason[$skx]); //forecast
8     $forecast = (($initialLevel + $initialTrend) + $initialSeason[$skx]); //forecast
9     $mape = (abs($forecast-$skv)/$skv) * 100; //hitung MAPE
10    $castData[] = ['month' => $this->getMonthName($monthIndex), 'price' => $skv, 'level' => $level, 'trend' => $trend, 'season' => $season, 'forecast' => $forecast, 'mape' => $mape];
11    $initialLevel = $level;
12    $initialTrend = $trend;
13    $initialSeason[] = $season;
14    if(($skx+1) % 12 == 0){ //untuk generate nama bulan
15        $monthIndex = 0;
16    }else{
17        $monthIndex = $monthIndex + 1;
18    }
19 }

```

```

1 public function forecastCore($data, $isPredictNextYear = false){ //core forecast logic
2     $castData = []; //semua data ramalan disimpan disini
3     // Initial level and initial Seasonal and initial trend
4     $initialLevel = 0;
5     $initialSeason = [];
6     $initialTrend = 0;
7     for($i = 0; $i < 12; $i++){
8         $firstYears = array_slice($data, 0, 12);
9         $secondYears = array_slice($data, 12, 24);
10        $initialLevel = (array_sum($firstYears) / 12);
11        $initialSeason[$i] = ($firstYears[$i]/$initialLevel);
12        $initialTrend += ($secondYears[$i] - $firstYears[$i])/12;
13    }
14    $initialTrend = ($initialTrend / 12);
15    $castData[] = ['price' => $initialLevel, 'level' => $initialLevel, 'trend' => $initialTrend, 'season' => $initialSeason[$i], 'forecast' => 0, 'mape' => 0];
16 }else{
17     $castData[] = ['price' => $initialLevel, 'level' => 0, 'trend' => 0, 'season' => $initialSeason[$i], 'forecast' => 0, 'mape' => 0];
18 }
19 }

```

```

1 $lastIndexCalc = count($data) - 12; //get index tahun terakhir
2 $lastIndexCastData = $castData[$lastIndexCalc - 1]; //get data terakhir
3 $dataSetTest = array_slice($castData, $lastIndexCalc, count($data)); //data ramalan tahun yang dicari
4 if($isPredictNextYear){
5     return $dataSetTest;
6 }
7 $forecastPrediction = [];
8 for($i = 0; $i < 12; $i++){
9     $forecast = ($lastIndexCastData['level'] + $i * $lastIndexCastData['trend']) * $dataSetTest[$i]['season']; // get data terbaru
10    $forecastPrediction[] = ['price' => 0, 'month' => $this->getMonthName($i), 'level' => 0, 'trend' => 0, 'season' => 0, 'forecast' => $forecast, 'mape' => 0];
11 }
12 return $forecastPrediction;
13 }

```

Gambar 5. 15 Peramalan

Pada gambar 5.15 adalah implementasi kode untuk peramalan. Untuk yang pertama function cast data untuk memanggil semua data yang disimpan lalu menentukan nilai initial level, initial, seasonal dan initial trend, lalu melakukan perulangan dengan nilai kurang dari 12 dengan parameter sesuai metode yaitu

firstyear, secondyear, initialtrend, initiallevel, initialseasonal dan yang terakhir adalah forecast.

5.3.5 Register

```
1 class RegisterRequest extends FormRequest
2 {
3     /**
4      * Determine if the user is authorized to make this request.
5      *
6      * @return bool
7      */
8     public function authorize()
9     {
10         return true;
11     }
12
13     /**
14      * Get the validation rules that apply to the request.
15      *
16      * @return array
17      */
18     public function rules()
19     {
20         return [
21             'name' => "required|string|min:5|alpha_dash",
22             'username' => "required|string|unique:users,username|min:5",
23             'position' => 'required|string',
24             'password' => 'required|string|min:8|confirmed',
25         ];
26     }
27 }
```

Gambar 5. 16 Register

Pada gambar 5.23 adalah implementasi kode untuk register. Disini di jelaskan yang pertama membuat function berupa authorize dan fuction rules yang digunakan menyimpan data berupa name, username, position dan password

5.3.6 Profile

```

1 public function index(Request $request){
2     return view($this->guard.'. '.$this->prefix.'.index', [
3         'title' => $this->title,
4         'prefix' => $this->prefix,
5         'guard' => $this->guard,
6         'data' => auth()->user()
7     ]);
8 }
9
10 public function edit(Request $request){
11     return view($this->guard.'. '.$this->prefix.'.edit', [
12         'title' => $this->title,
13         'prefix' => $this->prefix,
14         'guard' => $this->guard,
15         'data' => auth()->user()
16     ]);
17 }
18
19 public function update(Request $request){
20     $validated = $request->validate([
21         'username' => "required|string|alpha_dash|unique:users,username, ".auth()->user()->id."|min:5",
22         'name' => "required|string|min:5",
23         'position' => 'required|string',
24     ]);
25     User::where('id', auth()->user()->id)->update($request->only(['username', 'name', 'position']));
26     $toast = array(
27         'typeToast' => 'success',
28         'messageToast' => 'Success update data '.$this->title,
29     );
30     return redirect()->route($this->guard.'. '.$this->prefix.'.index')->with($toast);
31 }
32 }

```

```

1 public function changepassword(Request $request){
2     return view($this->guard.'. '.$this->prefix.'.changepassword', [
3         'title' => $this->title,
4         'prefix' => $this->prefix,
5         'guard' => $this->guard,
6         'data' => auth()->user()
7     ]);
8 }
9
10 public function changepasswordupdate(Request $request){
11     $validated = $request->validate([
12         'oldPassword' => "required",
13         'password' => "required|confirmed|min:7",
14         'password_confirmation' => 'required|min:7'
15     ]);
16
17     if(Hash::check($request->get('oldPassword'), auth()->user()->getAuthPassword())){
18         User::where('id', auth()->user()->id)->update(['password' => Hash::make($request->get('password'))]);
19         $toast = array(
20             'typeToast' => 'success',
21             'messageToast' => 'Success update data '.$this->title,
22         );
23         return redirect()->route($this->guard.'. '.$this->prefix.'.index')->with($toast);
24     }
25
26     return back()->withErrors(['oldPassword' => "Password yang anda masukkan salah."]);
27 }
28 }

```

Gambar 5. 17 Profile

Pada gambar 5.17 adalah implementasi kode untuk profile. Yang pertama function index dimana fungsi ini untuk menambahkan data berupa title, prefix, guard dan data. Selanjutnya function update dimana fungsi ini untuk

menambahkan data berupa username, name, position dan jika data berhasil diubah maka akan muncul pesan success update data. Dan yang terakhir function `changepasswordupdate`, function ini berfungsi untuk mengubah password yang lama ke yang baru dengan parameter `oldPassword`, `password`, `password_confirmation` dan jika password berhasil diubah maka akan muncul pesan success update data dan jika gagal maka pesan yang akan muncul Password yang anda masukkan salah!

5.4 Pengujian

Pada tahapan pengujian ini dilakukan dengan pengujian sistem untuk mengetahui apakah sistem yang dibuat sudah berjalan sesuai dengan yang dirancang. Tahap pengujian ini dapat berupa pengujian fungsionalitas sistem (black box), pengujian menggunakan excel, pengujian akurasi menggunakan MAPE, dan pengujian UAT.

5.4.1 Pengujian menggunakan MAPE

MAPE dapat digunakan oleh masyarakat luas karena MAPE mudah dipahami dan diterapkan dalam memprediksi akurasi peramalan khususnya menggunakan metode Holt-Winters Exponential Smoothing oleh karena itu MAPE dipilih untuk pengujian akurasi karena dapat memberikan hasil yang relatif akurat, dari sebuah pengujian dengan cara menghitung kesalahan mutlak dari setiap periode yang dibagi dengan nilai-nilai yang diteliti untuk periode tersebut.

$$\text{MAPE} = \frac{1}{n} \sum_{t=1}^t \left[\frac{y^2_i - y_i}{y^2} \right] \times 100$$

dengan,

X_t = data riil periode t

Y_t = data peramalan periode sekarang

n = jumlah periode

5.4.2 Pengujian menggunakan Black Box

Pengujian sistem (black box testing) digunakan untuk mengetahui apakah fitur pada sistem berjalan sesuai dengan fungsinya, Adapun fungsi yang di jalankan seperti login, register, edit profile, tambah data aktual, edit data akutal, hapus data aktual, forecast, tambah user, edit user dan delete user. Berikut adalah pengujian menggunakan black box testing :

Tabel 5. 1 Pengujian Fungsionalitas Sistem

No	Fungsionalitas	Skenario Pengujian	Hasil yang Diharapkan	Hasil yang Didapat

1	Login	Melakukan input username dan password kemudian klik tombol login.	Sistem menampilkan halaman Dashboard	Sesuai
2	Register	Melakukan pendaftaran akun kemudian klik tombol simpan	Sistem menyimpan data user	Sesuai
3	Edit Profile	Melakukan edit username, nama, dan posisi	Sistem akan menyimpan data yang telah diedit ke dalam database. Kemudian Sistem Menampilkan profil.	Sesuai
4	Tambah Data Aktual	Melakukan input bulan, tahun, dan harga	Sistem akan menyimpan data yang telah ditambahkan ke dalam database. Kemudian Sistem Menampilkan data aktual.	Sesuai
5	Edit Data Aktual	Melakukan edit bulan, tahun dan harga	Sistem akan menyimpan data yang	Sesuai

			telah diedit ke dalam database. Kemudian Sistem Menampilkan data aktual.	
6	Hapus Data Aktual	Memilih tombol hapus data aktual	Sistem akan menghapus data aktual	Sesuai
7	Forecast	Melakukan perhitungan dengan menginputkan tahun	Sistem menampilkan hasil peramalan	Sesuai
8	Tambah User	Melakukan input nama, username, posisi, password, password konfirmasi, dan role	Sistem akan menyimpan data yang telah ditambahkan ke dalam database. Kemudian Sistem Menampilkan data user.	Sesuai
9	Edit User	Melakukan edit nama, username, posisi, dan role	Sistem akan menyimpan data yang telah diedit ke dalam database. Kemudian	Sesuai

			Sistem Menampilkan data user.	
10	Delete User	Memilih tombol hapus user	Sistem akan menghapus data user	Sesuai

5.4.3 Pengujian UAT

Pengujian User Acceptance Testing (UAT) bertujuan untuk mengetahui apakah sistem yang telah dibuat sesuai dengan kebutuhan pengguna. Pada tahap ini sistem akan diuji untuk memastikan apakah fungsi dan tugasnya sudah sesuai dengan kebutuhan pengguna di dalam tabel tersebut terdapat no, pertanyaan, SS(sangat setuju), S(setuju), N(netral), KS(kurang setuju), dan TS(tidak setuju). Berikut adalah tabel 5.4 pengujian menggunakan User Acceptance Testing (UAT):

Tabel 5. 2 Pengujian User Acceptance Testing (UAT)

No	Pertanyaan	SS	S	N	KS	TS
1	Apakah tulisan pada aplikasi ini mudah dibaca dan di pahami oleh pengguna?					
2	Apakah tampilan aplikasi ini nyaman digunakan oleh pengguna?					
3	Menurut anda, apakah fitur pada sistem mudah untuk dipahami?					
4	Menurut anda, apakah fitur pada sistem ini sudah berjalan dengan semestinya?					
5	Menurut anda, apakah fitur pada sistem mudah digunakan?					

6	Menurut anda, apakah sistem ini membantu dalam meramalkan harga Gabah?					
7	Menurut anda, apakah sistem ini sudah sesuai dengan kebutuhan?					