

SKRIPSI

IMPLEMENTASI METODE SINGLE EXPONENTIAL  
SMOOTHING UNTUK PERAMALAN PRE-ORDER DAN  
ANGGARAN TOKO SEBATNSANS



DANANG SETYA NUGRAHA

NPM. 18.0504.0046

PROGRAM STUDI TEKNIK INFORMATIKA S1  
FAKULTAS TEKNIK  
UNIVERSITAS MUHAMMADIYAH MAGELANG  
JANUARI, 2025

## BAB 4

### HASIL DAN PEMBAHASAN

#### 4.1 Hasil

##### 4.1.1 Implementasi kode pada Ramalan Penjualan

Implementasi metode *Single Exponential Smoothing* pada penelitian ini menggunakan pemrograman PHP untuk mendukung proses peramalan. Metode ini diintegrasikan pada halaman *dashboard* guna mempermudah visualisasi hasil peramalan sehingga pengguna dapat dengan mudah memahami tren dan pola data yang ditampilkan. Tahapan implementasi dimulai dengan menyusun *query* untuk mengambil data historis yang tersimpan di basis data, yang menjadi dasar perhitungan metode ini. Data yang diambil kemudian diolah melalui algoritma *Single Exponential Smoothing* untuk menghasilkan prediksi. Subbab ini fokus pembahasan pada bagaimana algoritma diimplementasikan secara teknis dan struktur data yang digunakan pada ramalan penjualan.

```
$brand = $this->filters['brand_id'] ?? null;  
$series = $this->filters['series_id'] ?? null;  
$numberOfLimit = $this->filters['limit'] ?? 6;
```

Gambar 4. 1 Inisialisasi dan pemrosesan data filter

Gambar 4. 1 merupakan implementasi kode yang bertujuan untuk menghasilkan grafik penjualan aktual dan peramalan menggunakan metode *Single Exponential Smoothing* (SES). Langkah pertama adalah Inisialisasi filter yang diterima dari antarmuka pengguna. Filter tersebut mencakup `brand_id` untuk menentukan merek, `series_id` untuk memilih seri dari merek tersebut, dan `limit` yang membatasi jumlah data yang akan ditampilkan pada grafik. Jika filter tidak disediakan, nilai bawaan limit diatur menjadi 6.

```

if ($brand && $series) {
    $sales = \App\Models\Sale::whereHas('series', function ($query) use ($brand) {
        $query->where('brand_id', $brand);
    })
        ->where('series_id', $series)
        ->latest()
        ->get()
        ->reverse()
        ->values();
} else {
    $sales = \App\Models\Sale::whereHas('series', function ($query) use ($brand) {
        $query->where('brand_id', $brand);
    })
        ->selectRaw('month, year, SUM(total) as total')
        ->groupBy('month', 'year')
        ->orderBy('year')
        ->orderBy('month')
        ->get();
}

```

Gambar 4. 2 Pengambilan data penjualan berdasarkan filter

Gambar 4. 2 merupakan *Eloquent Model* data penjualan diambil dari basis data berdasarkan filter yang diberikan. Jika filter `brand_id` dan `series_id` tersedia, data penjualan diambil secara spesifik untuk seri tertentu dalam merek yang dipilih. Jika hanya `brand_id` yang tersedia, data diambil dengan mengelompokkan penjualan berdasarkan bulan dan tahun, lalu menjumlahkan total penjualan dalam kelompok tersebut. Data yang diperoleh diurutkan berdasarkan waktu untuk memastikan urutan data historis.

```

$labels = $sales->map(function ($sale) {
    $monthName = date('F', mktime(0, 0, 0, $sale->month, 1));
    return "{$monthName} {$sale->year}";
})->toArray();
$lastLabel = end($labels);

```

Gambar 4. 3 Pembuatan label data untuk grafik

Gambar 4.3 merupakan kode untuk keperluan grafik. Label merepresentasikan bulan dan tahun dari data penjualan. Nama bulan diambil menggunakan fungsi `date()` dan `mktime()` untuk mengonversi nomor bulan menjadi format yang mudah dipahami, seperti "Januari" atau "Februari". Label digunakan untuk mempermudah pembaca dalam memahami periode data pada grafik.

```

$lastLabel = end($labels);
if (strpos($lastLabel, ' ') !== false) {
    $lastMonthYear = explode(' ', $lastLabel);
    $lastMonth = $lastMonthYear[0];
    $lastYear = (int)$lastMonthYear[1];
} else {
    $lastMonth = null;
    $lastYear = null;
}

$nextMonth = (int)date('m', strtotime("first day of +1 month", mktime(0, 0, 0, array_search(
    $lastMonth,
    array(
        'January',
        'February',
        'March',
        'April',
        'May',
        'June',
        'July',
        'August',
        'September',
        'October',
        'November',
        'December'
    )
) + 1, 1));

$nextMonthName = date('F', mktime(0, 0, 0, $nextMonth, 1));
$nextYear = $nextMonth === 1 ? $lastYear + 1 : $lastYear;
array_push($labels, "{$nextMonthName} {$nextYear}");

```

Gambar 4. 4 Kode menentukan bulan berikutnya

Gambar 4.4 merupakan kode untuk menghitung bulan dan tahun berikutnya berdasarkan data terakhir. Hal ini dilakukan dengan memecah label terakhir menjadi nama bulan dan tahun kemudian menghitung bulan berikutnya menggunakan fungsi `strtotime()`. Jika bulan berikutnya adalah Januari, maka tahun otomatis ditingkatkan. Bulan dan tahun berikutnya ini ditambahkan ke *array* label untuk digunakan pada peramalan.

Metode SES diterapkan untuk membuat prediksi penjualan di masa depan. Faktor perataan ditetapkan ( $a$ ) sebesar 0.2 yang menentukan sensitivitas ramalan terhadap data terbaru. Nilai ramalan dihitung menggunakan formula SES di mana nilai ramalan saat ini merupakan kombinasi bobot dari data aktual saat ini dan ramalan sebelumnya. Hasil ramalan disimpan dalam *array* terpisah, yang kemudian digunakan untuk

grafik. Kode pada gambar 4.5 berikut merupakan kode untuk membuat metode *Single Exponential Smoothing*.

```

/* ----- SES ----- */
$data = $sales->pluck('total')->toArray();
$forecast = [];
$previousForecast = null;
$alpha = 0.2;

foreach ($data as $key => $value) {
    if (is_null($previousForecast)) {
        $previousForecast = $value;
    } else {
        $previousForecast = $alpha * $value + (1 - $alpha) * $previousForecast;
    }
    $forecast[$key] = round($previousForecast, 0);
}
array_unshift($forecast, $forecast[0] ?? null);

```

Gambar 4. 5 Penerapan metode *Single Exponential Smoothing* (SES)

#### 4.1.2 Implementasi kode pada Ramalan Anggaran

Penelitian ini mengimplementasikan metode *Single Exponential Smoothing* menggunakan pemrograman PHP untuk mendukung proses peramalan anggaran. Metode ini diterapkan pada halaman *dashboard* untuk memudahkan visualisasi hasil peramalan sehingga pengguna dapat memahami tren dan pola data anggaran yang ditampilkan secara lebih intuitif. Tahapan implementasi diawali dengan pembuatan *query* untuk mengambil data historis dari basis data yang menjadi dasar dalam perhitungan metode ini. Data yang diperoleh kemudian diproses menggunakan algoritma *Single Exponential Smoothing* untuk menghasilkan prediksi anggaran. Subbab ini berfokus pada pembahasan teknis implementasi algoritma serta struktur data yang digunakan dalam proses peramalan anggaran.

```

$brand = $this->filters['brand_id'] ?? null;
$series = $this->filters['series_id'] ?? null;
$numberOfLimit = $this->filters['limit'] ?? 6;
if ($brand && $series) {
    $sales = \App\Models\Sale::whereHas('series', function ($query) use ($brand) {
        $query->where('brand_id', $brand);
    })
        ->where('series_id', $series)
        ->latest()
        ->get()
        ->reverse()
        ->values();
    $data = $sales->map(function ($sale) {
        return $sale->total * $sale->price;
    })->toArray();
} else {
    $sales = \App\Models\Sale::with('series')
        ->whereHas('series', function ($query) use ($brand) {
            $query->where('brand_id', $brand);
        })
        ->select('series_id', 'price', 'total', 'month', 'year')
        ->get();
    $data = $sales->groupBy(function ($sale) {
        return $sale->series->brand_id . '-' . $sale->month;
    })->map(function ($group) {
        return $group->sum(function ($sale) {
            return $sale->total * $sale->price;
        });
    })->values()->toArray();
}

```

Gambar 4. 6 Filter dan data awal

Gambar 4.6 merupakan kode yang berfungsi untuk mendapatkan data anggaran berdasarkan filter tertentu seperti brand\_id, series\_id, dan batas jumlah data (limit). Jika filter brand\_id dan series\_id di isi, data akan difilter untuk kombinasi brand dan series tertentu. Jika tidak, data anggaran akan dikelompokkan berdasarkan brand\_id dan bulan lalu dihitung total penjualannya. Hasil dari pengolahan data ini akan disimpan dalam variabel \$data.

```

$labels = $sales->map(function ($sale) {
    $monthName = date('F', mktime(0, 0, 0, $sale->month, 1));
    return "{$monthName} {$sale->year}";
})->unique()->toArray();
$lastLabel = end($labels);

if (strpos($lastLabel, ' ') !== false) {
    $lastMonthYear = explode(' ', $lastLabel);
    $lastMonth = $lastMonthYear[0];
    $lastYear = (int)$lastMonthYear[1];
} else {
    $lastMonth = null;
    $lastYear = null;
}

```

Gambar 4. 7 Label untuk data anggaran

Gambar 4.7 merupakan kode untuk membuat label untuk data anggaran berdasarkan bulan dan tahun dari setiap penjualan. Label ini digunakan untuk menunjukkan periode waktu dari data yang ditampilkan. Selain itu bulan dan tahun terakhir dari data juga diidentifikasi untuk kebutuhan prediksi bulan berikutnya.

```

$nextMonth = (int)date('m', strtotime("first day of +1 month", mktime(0, 0, 0, array_search(
    $lastMonth,
    array(
        'January',
        'February',
        'March',
        'April',
        'May',
        'June',
        'July',
        'August',
        'September',
        'October',
        'November',
        'December'
    )
    + 1, 1)));

$nextMonthName = date('F', mktime(0, 0, 0, $nextMonth, 1));
$nextYear = $nextMonth == 1 ? $lastYear + 1 : $lastYear;
array_push($labels, "{$nextMonthName} {$nextYear}");

```

Gambar 4. 8 Kode menentukan bulan dan tahun berikutnya.

Gambar 4.8 merupakan kode untuk menghitung bulan dan tahun berikutnya berdasarkan bulan terakhir dari data. Jika bulan berikutnya adalah Januari, maka tahun akan bertambah satu. Label untuk bulan dan tahun berikutnya kemudian ditambahkan ke *array* \$labels.

```

$forecast = [];
$previousForecast = null;
$alpha = 0.2;

foreach ($data as $key => $value) {
    if (is_null($previousForecast)) {
        $previousForecast = $value;
    } else {
        $previousForecast = $alpha * $value + (1 - $alpha) * $previousForecast;
    }
    $forecast[$key] = round($previousForecast, 0);
}
array_unshift($forecast, $forecast[0] ?? null);

```

Gambar 4. 9 Kode metode *Single Exponential Smoothing*

Gambar 4.9 merupakan kode untuk menghitung prediksi anggaran menggunakan metode *Single Exponential Smoothing* dengan parameter *alpha* sebesar 0.2. Prediksi dilakukan secara iteratif untuk setiap data anggaran. Nilai prediksi awal ditentukan oleh data anggaran pertama dan prediksi berikutnya dihitung berdasarkan formula *Single Exponential Smoothing*.

### 4.1.3 Hasil Implementasi Antarmuka Sistem

#### 4.1.4.1 Halaman *Login*

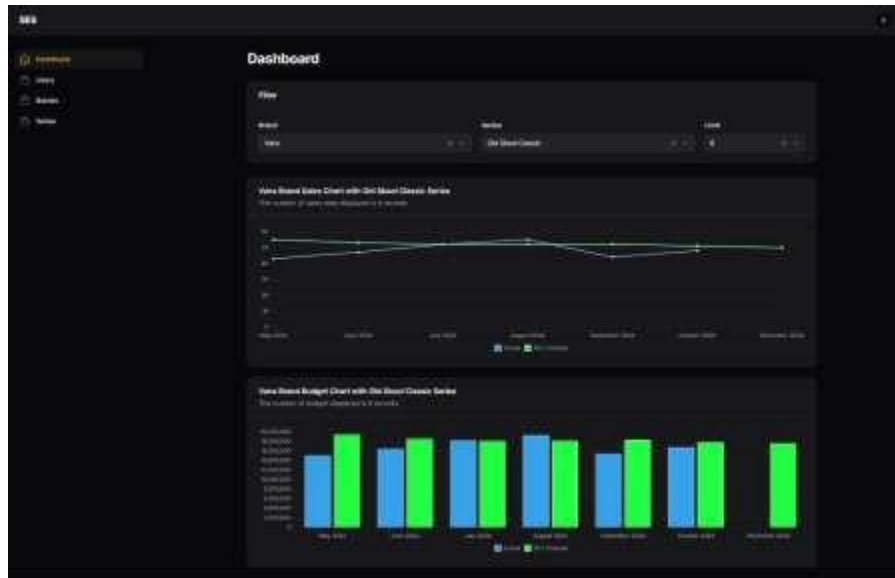


Gambar 4. 10 Halaman *login*



Gambar 4.10 menunjukkan halaman *login* aplikasi pada sistem peramalan penjualan dan anggaran sepatu Toko Sebatsans. Halaman ini dilengkapi fitur *remember me* untuk memudahkan *login* ulang. Tombol *Sign in* berwarna kuning memberikan penekanan tindakan utama untuk mendukung *user experience* yang nyaman dan aman melalui validasi data.

#### 4.1.4.2 Halaman *Dashboard*

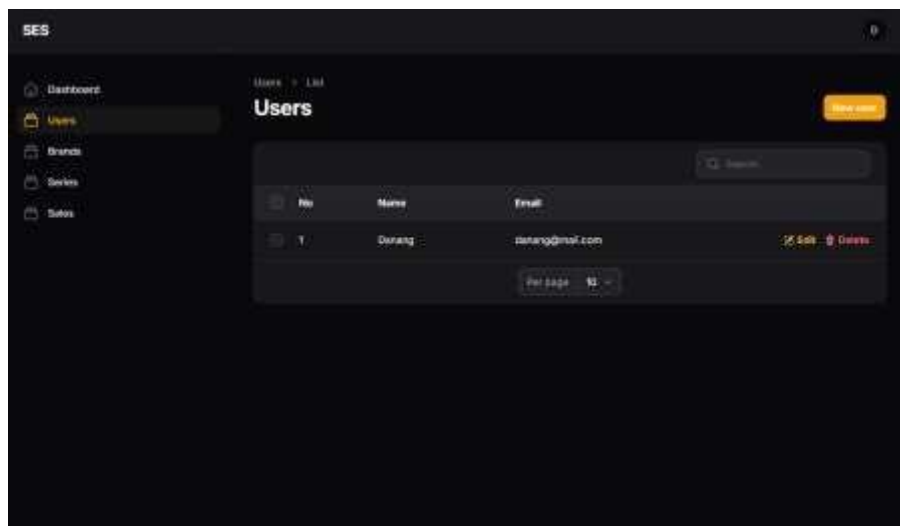


Gambar 4. 11 Halaman *dashboard*

Gambar 4.11 di atas merupakan halaman *dashboard* yang berfungsi untuk menampilkan informasi terperinci terkait data penjualan dan anggaran berdasarkan merek dan seri produk. Terdapat fitur filter yang memungkinkan pengguna untuk memilih merek, seri, dan batas data yang ingin ditampilkan. Halaman *dashboard* menyajikan dua grafik yaitu grafik penjualan dan grafik anggaran di mana masing-masing grafik memperlihatkan data aktual dan hasil peramalan menggunakan metode *Single Exponential Smoothing*. Grafik penjualan menggambarkan jumlah unit yang terjual selama periode tertentu, sedangkan grafik anggaran menunjukkan perbandingan anggaran aktual dengan hasil prediksi. Desain antarmuka yang sederhana dan terstruktur ini mempermudah pengguna untuk menganalisis ramalan penjualan dan anggaran untuk mengambil keputusan berbasis data secara cepat.

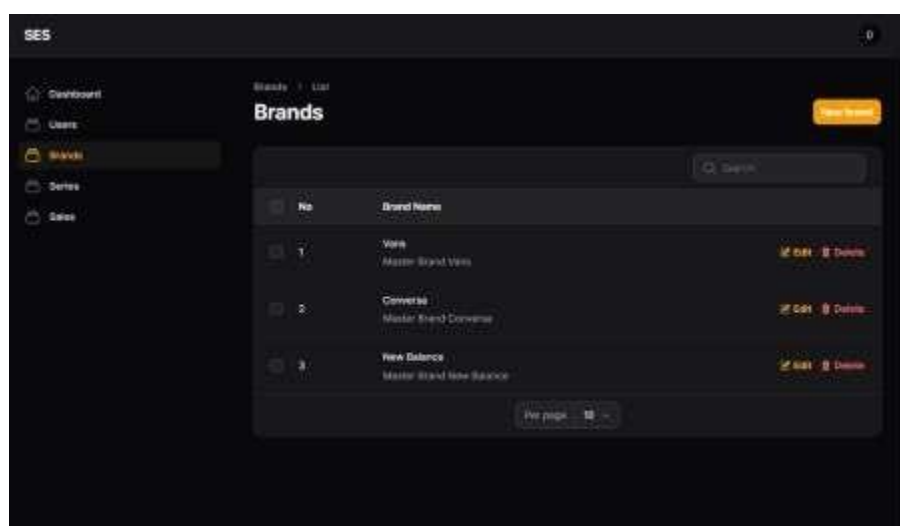
#### 4.1.4.3 Halaman Pengguna

Gambar 4.12 di bawah ini merupakan halaman pengguna pada aplikasi ini menampilkan daftar pengguna yang terdaftar di sistem. Pada bagian kanan tersedia tombol *New User* untuk menambahkan pengguna baru. Setiap baris pada tabel memiliki opsi *Edit* untuk mengubah informasi pengguna dan *Delete* untuk menghapus pengguna dari sistem. Desain halaman ini dirancang untuk mendukung pengelolaan data pengguna dengan efisien dan mudah digunakan.



Gambar 4. 12 Halaman pengguna

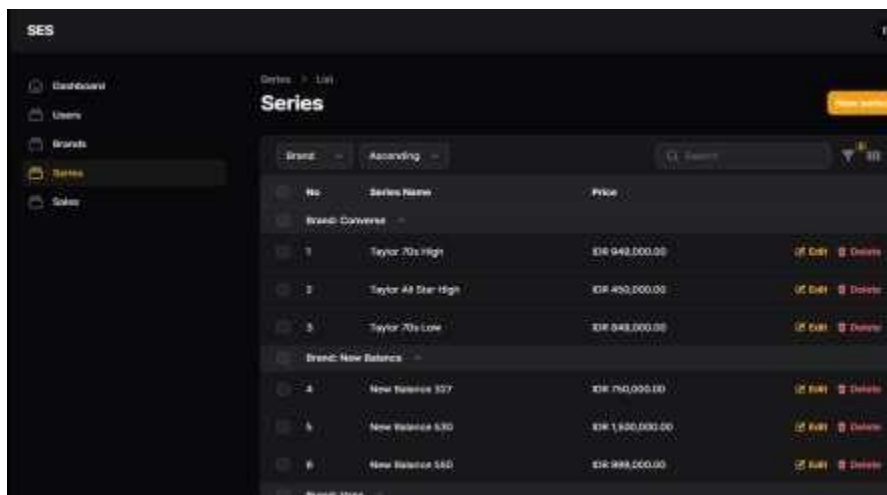
#### 4.1.4.4 Halaman Merek



Gambar 4. 13 Halaman merek

Gambar 4.13 di atas merupakan halaman *Brands* atau merek pada aplikasi ini berfungsi untuk menampilkan dan mengelola daftar master merek yang tersedia di sistem.. Tombol *New Brand* di sisi kanan atas memungkinkan pengguna menambahkan merek baru ke dalam sistem. Setiap baris pada tabel menyediakan opsi Edit untuk memperbarui informasi merek dan *Delete* untuk menghapus merek dari daftar. Di bagian atas kanan tabel terdapat fitur pencarian untuk mempermudah pengguna menemukan merek tertentu serta pengaturan jumlah data per halaman yang dapat disesuaikan.

#### 4.1.4.5 Halaman Seri



No.	Series Name	Price		
Brand: Converse				
1	Taylor 70s High	₹18.948.000,00	[Edit]	[Delete]
2	Taylor All Star High	₹18.400.000,00	[Edit]	[Delete]
3	Taylor 70s Low	₹18.048.000,00	[Edit]	[Delete]
Brand: New Balance				
4	New Balance 327	₹18.710.000,00	[Edit]	[Delete]
5	New Balance 330	₹18.1.800.000,00	[Edit]	[Delete]
6	New Balance 360	₹18.998.000,00	[Edit]	[Delete]
Brand: Nike				

Gambar 4. 14 Halaman Seri

Gambar 4.14 di atas merupakan halaman *Series* atau seri sepatu pada aplikasi ini berfungsi untuk menampilkan dan mengelola daftar seri produk yang tersedia di sistem. Tabel utama berisi kolom nomor, seri, dan harga masing-masing produk. Data dalam tabel dikelompokkan berdasarkan merek sehingga memudahkan pengguna dalam navigasi. Tombol *New Series* di sisi kanan atas memungkinkan pengguna menambahkan seri baru ke dalam sistem. Setiap baris pada tabel menyediakan opsi Edit untuk memperbarui informasi seri dan *Delete* untuk menghapus seri dari daftar. Di bagian atas tabel, terdapat fitur pencarian untuk mempermudah pengguna menemukan seri tertentu, serta filter berdasarkan merek dan opsi pengurutan.

#### 4.1.4.6 Halaman Rekap Penjualan

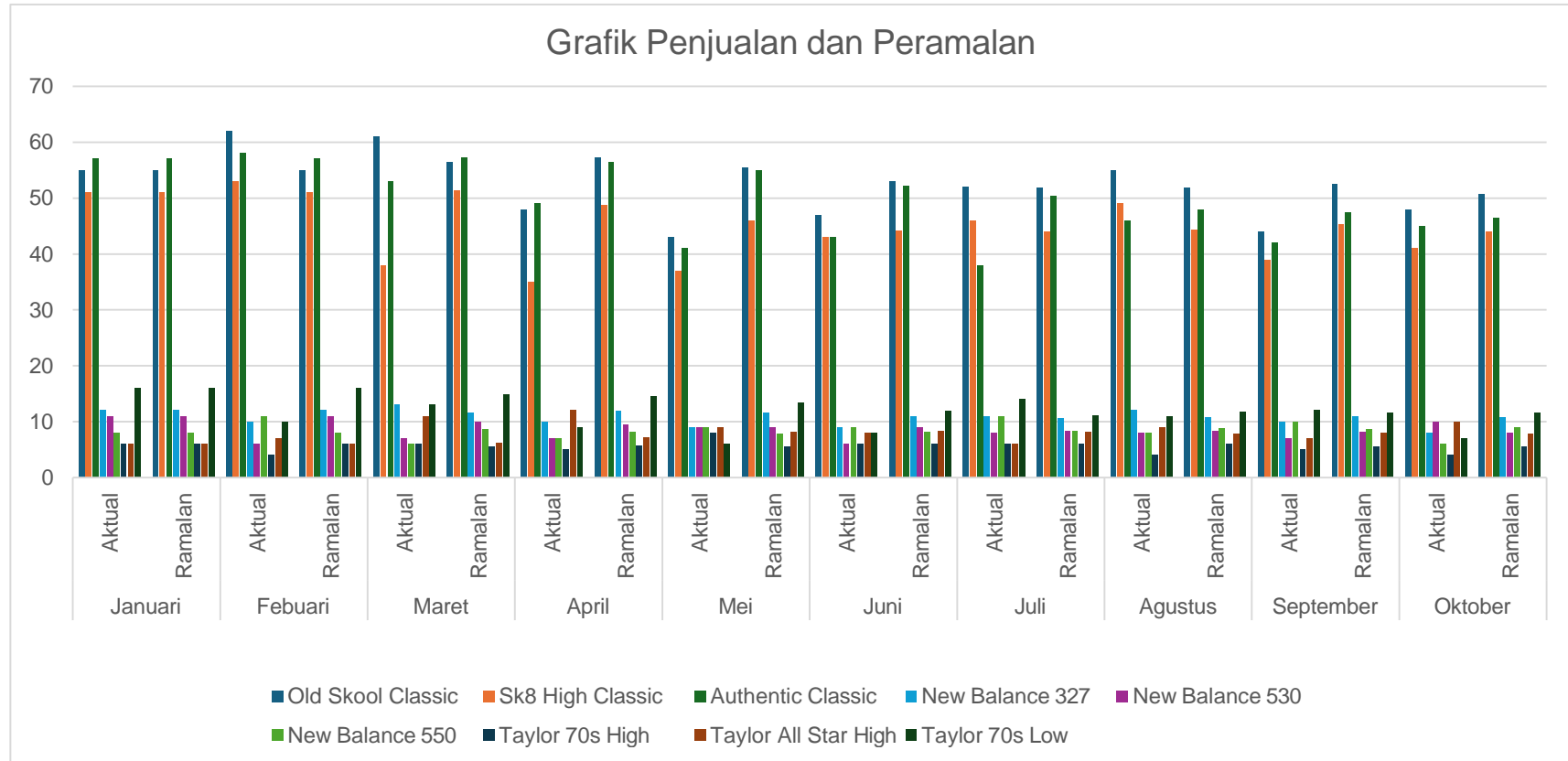
Gambar 4.15 di bawah ini merupakan halaman *Sales* atau penjualan pada aplikasi yang berfungsi untuk merekap dan mengelola data penjualan sepatu berdasarkan merek dan seri yang terjual setiap bulannya. Data ditampilkan dalam bentuk tabel dengan kolom yang mencakup informasi nomor, merek, seri, bulan penjualan, bulan penjualan, harga, total penjualan, dan total anggaran sehingga memberikan gambaran yang jelas mengenai jumlah penjualan dan anggaran.

Pengguna dapat memanfaatkan tombol *New Sale* di kanan atas untuk menambahkan data penjualan baru. Setiap baris dalam tabel dilengkapi dengan opsi *Edit* untuk memperbarui informasi dan *Delete* untuk menghapus data yang tidak relevan. Di bagian atas tabel, terdapat fitur pencarian dan filter yang mempermudah pengguna dalam menemukan data tertentu. Selain itu data dapat diurutkan berdasarkan bulan, tahun, atau harga secara *Ascending* atau *Descending*. Bagian bawah tabel menyajikan *Summary* yang mencantumkan total penjualan dalam pasangan sepatu serta total anggaran untuk bulan tersebut. Dengan fitur-fitur ini halaman penjualan dirancang untuk mempermudah pengguna dalam memantau kinerja penjualan, melakukan analisis data, dan mengelola pencatatan penjualan.

No	Brand Name	Series Name	Month	Year	Price (Rp)	Total Sales (pairs)	Total Pk
1	New Balance	New Balance 327	January	2024	Rp 750,000.00	12	Rp 9,000
2	Vans	Old Skool Classic	January	2024	Rp 250,000.00	59	Rp 14,750
3	Vans	Authentic Classic	January	2024	Rp 350,000.00	57	Rp 19,950
4	Vans	Old High Classic	January	2024	Rp 400,000.00	31	Rp 12,400
January summary						Total Sales (pairs)	Total Pk
						170	Rp 713,000,000

Gambar 4. 15 Halaman penjualan

#### 4.1.4 Hasil Implementasi Metode *Single Exponential Smoothing*



Gambar 4. 16 Grafik penjualan dan peramalan pada sistem

Gambar 4.16 merupakan grafik representasi visual dari hasil peramalan yang diperoleh melalui penerapan metode *Single Exponential Smoothing* (SES) pada aplikasi peramalan yang telah dikembangkan. Hasil yang tergambar dalam grafik tersebut konsisten dengan perhitungan yang tercatat dalam Tabel 3.6 yang mencakup peramalan untuk masing-masing kategori produk yaitu Vans, New Balance, dan Converse dengan masing-masing serinya. Dengan demikian hasil peramalan yang disajikan dalam grafik ini mencerminkan perhitungan yang valid dan sesuai dengan data penjualan aktual yang tersedia. Langkah berikutnya adalah mengevaluasi akurasi hasil peramalan. Evaluasi dilakukan dengan cara melakukan perhitungan terhadap tingkat kesalahan menggunakan metrik *Mean Absolute Percentage Error* (MAPE). Perhitungan MAPE ini bertujuan untuk mengukur sejauh mana hasil peramalan mendekati nilai aktual dengan memberikan gambaran yang lebih jelas tentang efektivitas metode SES dalam memprediksi volume penjualan produk yang dianalisis.

Tabel 4. 1 Perhitungan Akurasi

T	D	Vans			New Balance			Converse		
		V1	V2	V3	N1	N2	N3	C1	C2	
A	1	55	51	57	12	11	8	6	6	
F		55	51	57	12	11	8	6	6	
MAPE		0%	0%	0%	0%	0%	0%	0%	0%	
A	2	62	53	58	10	6	11	4	7	
F		55	51	57	12	11	8	6	6	16
MAPE		11%	4%	2%	20%	83%	27%	50%	14%	
A	3	61	38	53	13	7	6	6	11	
F		56	51	57	12	10	9	6	6	
MAPE		8%	35%	8%	11%	43%	43%	7%	44%	
A	4	48	35	49	10	7	7	5	12	
F		57	49	56	12	9	8	6	7	14
MAPE		19%	39%	15%	19%	34%	15%	14%	40%	
A	5	43	37	41	9	9	9	8	9	
F		55	46	55	12	9	8	6	8	13
MAPE		29%	24%	34%	28%	1%	13%	31%	10%	
A	6	47	43	43	9	6	9	6	8	
F		53	44	52	11	9	8	6	8	
MAPE		13%	3%	21%	22%	49%	10%	1%	4%	

T	D	Vans			New Balance			Converse		
		V1	V2	V3	N1	N2	N3	C1	C2	
A	7	52	46	38	11	8	11	6	6	
F		52	44	50	11	8	8	6	8	
MAPE		0%	4%	32%	4%	4%	25%	0%	37%	
A	8	55	49	46	12	8	8	4	9	11
F		52	44	48	11	8	9	6	8	
MAPE		6%	9%	4%	11%	3%	10%	51%	13%	
A	9	44	39	42	10	7	10	5	7	
F		52	45	47	11	8	9	6	8	
MAPE		19%	16%	13%	9%	17%	13%	12%	15%	
A	10	48	41	45	8	10	6	4	10	7
F		51	44	46	11	8	9	5	8	12
MAPE		6%	7%	3%	34%	20%	49%	37%	22%	66%
S-Avg		11%	14%	13%	16%	26%	21%	20%	20%	40%
B-Avg		13%			21%			27%		
A-Avg		20%								

Keterangan :

- T : Jenis data  
D : Nomor data  
A : Actual  
F : Forecasting  
V1 : Vans Old Skool Classic  
V2 : Vans Sk8 High Classic  
V3 : Vans Authentic Classic  
N1 : New Balance 327  
N2 : New Balance 530  
N3 : New Balance 550  
C1 : Taylor 70s High  
C2 : Taylor All Star High  
C3 : Taylor 70s Low  
S-Avg : Rata-rata berdasarkan seri  
B-Avg : Rata-rata berdasarkan merek  
A-Avg : Rata-rata seluruh

Tabel 4.1 di atas merupakan perhitungan nilai akurasi berdasarkan hasil perhitungan MAPE. Dari perhitungan tersebut diperoleh nilai *error* untuk masing-masing merek dan seri serta rata-rata nilai *error* keseluruhan peramalan menggunakan metode *Single Exponential Smoothing*. Analisis ini mengindikasikan tingkat akurasi peramalan untuk setiap kategori produk yang dihitung sehingga memungkinkan evaluasi yang lebih mendetail terkait performa model peramalan yang diterapkan.

#### 4.2 Pembahasan

Pada bab ini akan dibahas hasil evaluasi dari metode peramalan yang diterapkan yaitu *Single Exponential Smoothing* (SES). Pembahasan ini mencakup analisis kesalahan peramalan yang diperoleh serta perbandingan dengan referensi yang ada untuk menilai akurasi dan efektivitas metode yang digunakan. Berdasarkan perhitungan kesalahan (MAPE) yang telah dilakukan dengan hasil pada tabel 4.1. Pembahasan akan mengaitkan hasil yang diperoleh dengan literatur berdasarkan tabel 2.1 untuk memberikan gambaran yang lebih mendalam mengenai kemampuan metode *Single Exponential Smoothing* pada penelitian ini.



Gambar 4. 17 Grafik perbandingan akurasi

Gambar 4.17 di atas merupakan hasil rata-rata keseluruhan nilai *error* yang tercantum dalam Tabel 4.1 yang kemudian dibandingkan dengan referensi pada Tabel 2.1. Berdasarkan referensi tersebut metode peramalan yang digunakan yaitu *Single Exponential Smoothing* (SES) termasuk dalam kategori "**Good**



*Forecasting Ability*" dengan nilai akurasi mencapai 80% sebagaimana dapat dilihat pada Gambar 4.17. Data yang diperoleh menunjukkan bahwa metode SES mampu memberikan hasil peramalan yang cukup memadai dalam memprediksi data penjualan yang tercermin pada nilai MAPE yang terhitung.

Nilai akurasi yang tercatat sebesar 80% mengindikasikan bahwa metode peramalan ini dapat menghasilkan prediksi dengan kesalahan yang relatif rendah. Hal ini menandakan adanya konsistensi dalam model dalam mengurangi tingkat kesalahan peramalan. Dengan demikian, metode *Single Exponential Smoothing* dapat dianggap efektif dalam memberikan prediksi yang cukup baik serta memiliki potensi untuk dijadikan sebagai alat bantu dalam perencanaan atau pengambilan keputusan yang berbasis pada data historis dan tren yang sedang dianalisis. Perbandingan ini menegaskan bahwa meskipun terdapat variasi dalam hasil peramalan untuk setiap produk, model ini tetap menunjukkan kemampuan yang stabil dan dapat dipertimbangkan untuk aplikasi peramalan dalam konteks yang lebih luas terutama dalam meramalkan penjualan produk secara lebih akurat.