

Nama : Al Fitra Nur Ramadhani

NIM : 202210370311264

Mata Kuliah : Data Modeling & Simulation B

Pemodelan dan Peramalan Perubahan Suhu

1. Pendahuluan

Laporan ini menyajikan analisis detail tentang pemodelan dan peramalan perubahan suhu menggunakan beberapa teknik: Hukum Newton tentang pendinginan/pemanasan, simulasi perubahan suhu berkesinambungan menggunakan data dunia nyata, dan metode peramalan exponential smoothing. Implementasi menggunakan dataset cuaca London yang mencakup beberapa dekade untuk memvalidasi model matematika dan menilai akurasi peramalan.

Link Data Set : [London Weather Data](#)

Link GitHub : [DATA-MODELING-AND-SIMULATION/LondonWeatherSimulation at master · alfitranurr/DATA-MODELING-AND-SIMULATION](#)

2. Implementasi dan Analisis Step-by-Step

2.1 Mengubah Laju Pendinginan dan Mengamati Efeknya pada Penurunan Suhu

Program mengimplementasikan Hukum Newton tentang pendinginan yang menyatakan bahwa laju kehilangan panas sebanding dengan perbedaan suhu antara objek dan sekitarnya:

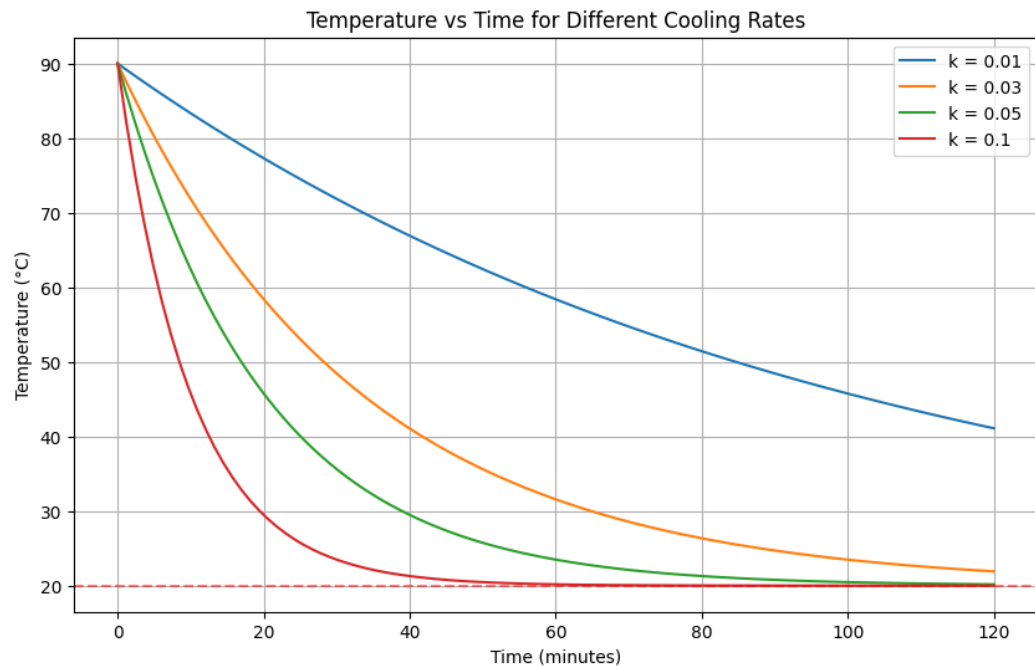
```
def cooling_simulation(t, T, cooling_rate, T_ambient):  
    """Newton's law of cooling:  $dT/dt = -k(T - T_{ambient})$ """  
    return -cooling_rate * (T - T_ambient)
```

Persamaan diferensial ini diselesaikan menggunakan solve_ivp (Initial Value Problem) dari SciPy dengan metode Runge-Kutta.

Analisis Hasil

Dari Gambar, dapat diamati bahwa:

- Simulasi dilakukan dengan empat nilai laju pendinginan berbeda ($k = 0,01, 0,03, 0,05, 0,1$)
- Suhu awal adalah 90°C dan suhu lingkungan adalah 20°C
- Nilai k yang lebih tinggi menyebabkan penurunan suhu lebih cepat:
 - Dengan $k = 0,01$: Suhu turun perlahan, membutuhkan lebih dari 120 menit untuk mendekati suhu lingkungan
 - Dengan $k = 0,1$: Suhu turun dengan cepat, mencapai sekitar 22°C dalam 60 menit
- Semua kurva menunjukkan pola peluruhan eksponensial, sesuai dengan model matematis



2.2 Mensimulasikan Pemanasan Alih-alih Pendinginan

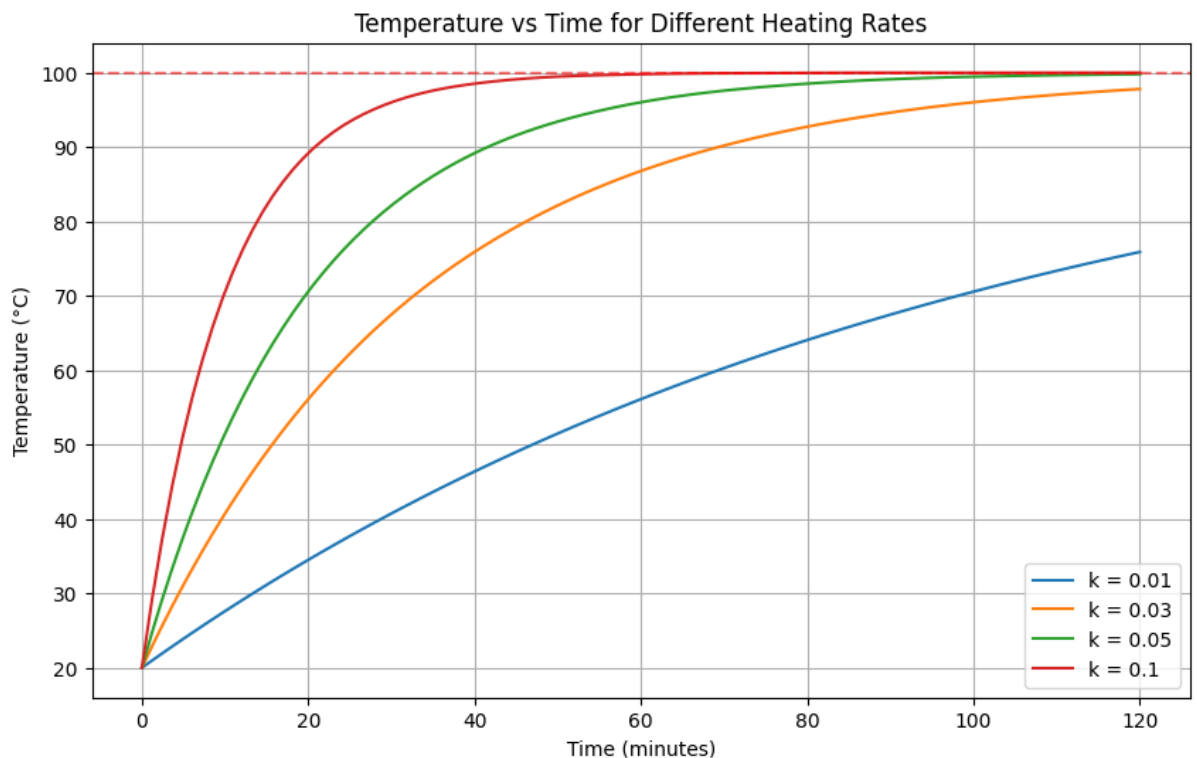
Untuk mensimulasikan pemanasan, persamaan diferensial dimodifikasi menjadi:

```
def heating_simulation(t, T, heating_rate, T_max):
    """Modified equation for heating: dT/dt = k(T_max - T)"""
    return heating_rate * (T_max - T)
```

Analisis Hasil

Dari Gambar, dapat diamati bahwa:

- Simulasi dilakukan dengan empat nilai laju pemanasan berbeda ($k = 0,01, 0,03, 0,05, 0,1$)
- Suhu awal adalah 20°C dan suhu maksimum adalah 100°C
- Nilai k yang lebih tinggi menyebabkan kenaikan suhu lebih cepat:
 - Dengan $k = 0,01$: Suhu naik perlahan, mencapai sekitar 75°C setelah 120 menit
 - Dengan $k = 0,1$: Suhu naik cepat, mendekati 100°C dalam waktu kurang dari 60 menit
- Semua kurva menunjukkan pola pertumbuhan eksponensial yang asimtotik menuju suhu maksimum



2.3 Menggunakan Dataset Dunia Nyata untuk Mensimulasikan Perubahan Suhu

Program mengadaptasi Hukum Newton untuk mensimulasikan perubahan suhu per jam berdasarkan suhu rata-rata harian dari data cuaca London:

```
def temperature_change_model(t, T, k, T_target):
    """Model the change in temperature towards a target temperature"""
    return -k * (T - T_target)
```

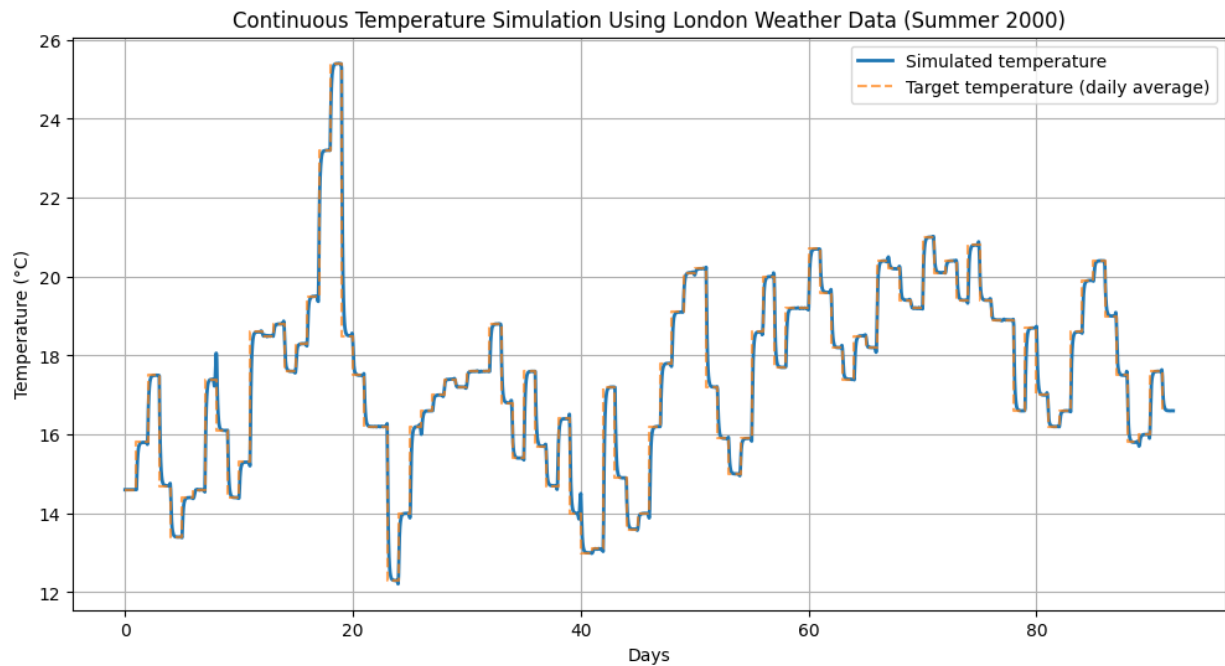
Langkah-langkah implementasi meliputi:

1. Memuat data cuaca London
2. Mendeteksi tahun yang tersedia dan memilih periode musim panas yang sesuai
3. Membuat fungsi suhu target yang berubah setiap hari
4. Mensimulasikan perubahan suhu per jam menggunakan **solve_ivp**

Analisis Hasil

Dari Gambar (simulasi untuk Musim Panas 2000), dapat diamati bahwa:

- Model berhasil mengikuti pola suhu rata-rata harian dengan baik
- Transisi suhu antar hari tampak realistis
- Nilai k sebesar 0,5 memberikan keseimbangan yang baik antara responsivitas dan kehalusan kurva
- Variasi suhu hari ke hari tertangkap dengan baik, termasuk puncak signifikan sekitar hari ke-20
- Simulasi menghasilkan kurva yang mulus antara nilai suhu rata-rata harian



2.4 Membandingkan CES vs. DES untuk Sistem yang Serupa

Kode mengimplementasikan dan membandingkan dua metode peramalan:

1. CES (Continuous/Simple Exponential Smoothing):

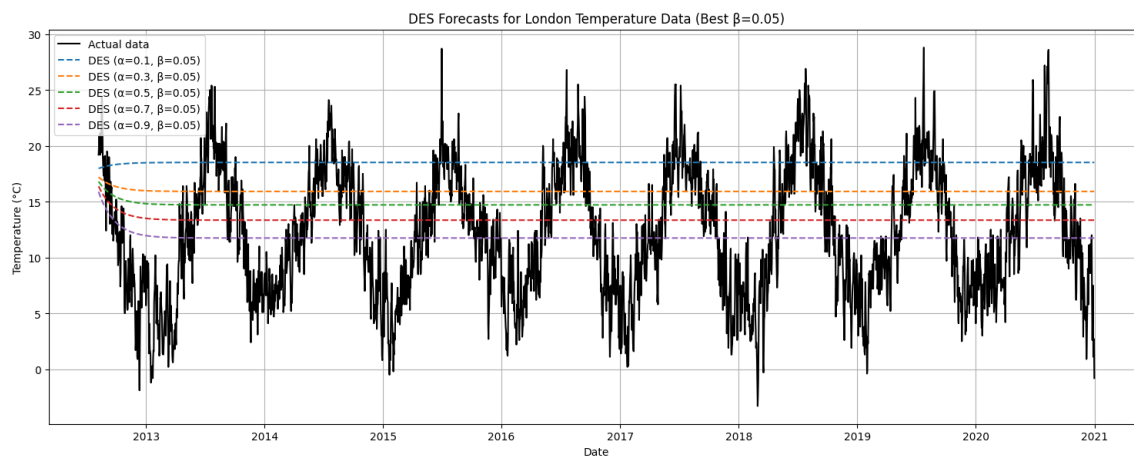
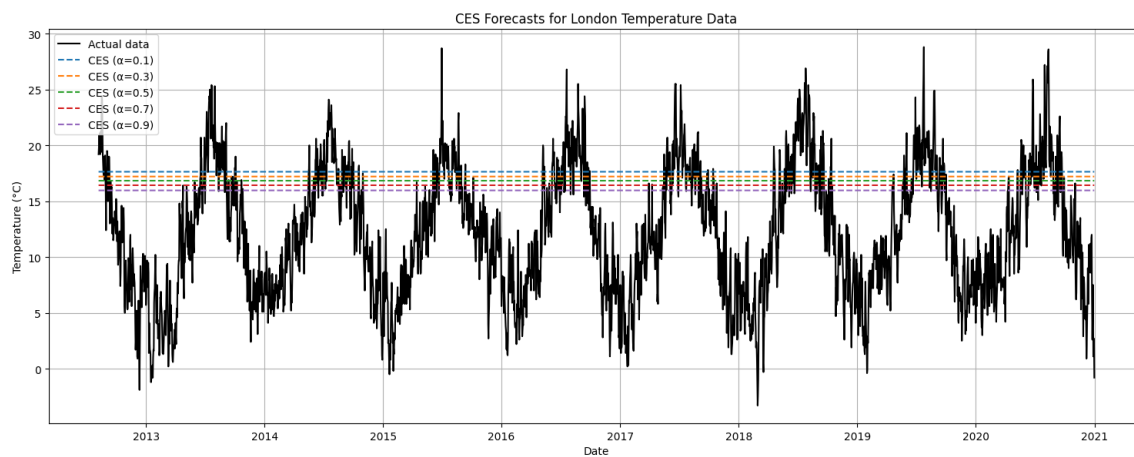
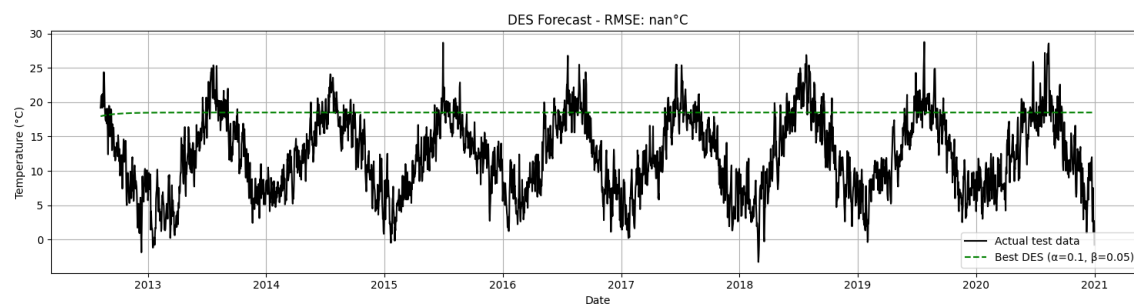
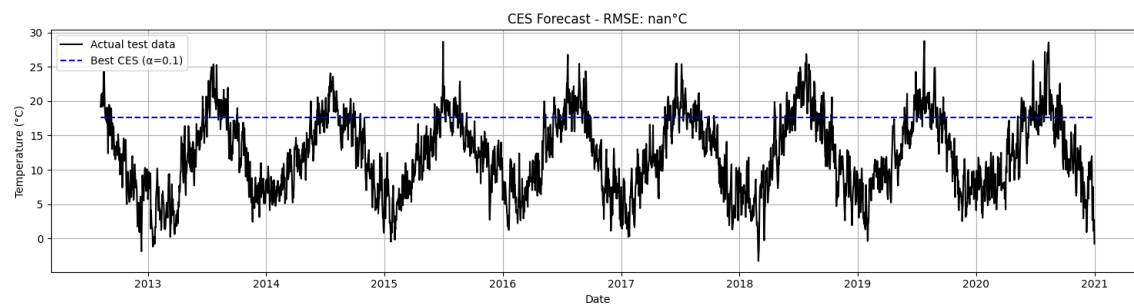
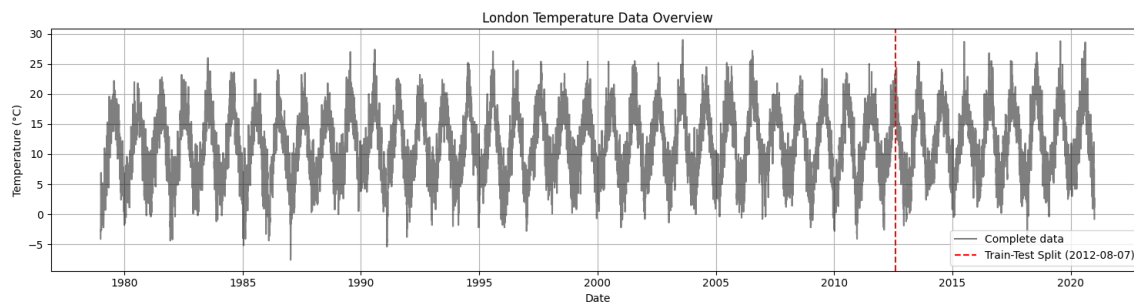
```
def apply_ces(data, alpha=0.3):
    model = SimpleExpSmoothing(data)
    model_fit = model.fit(smoothing_level=alpha, optimized=False)
    return model_fit
```

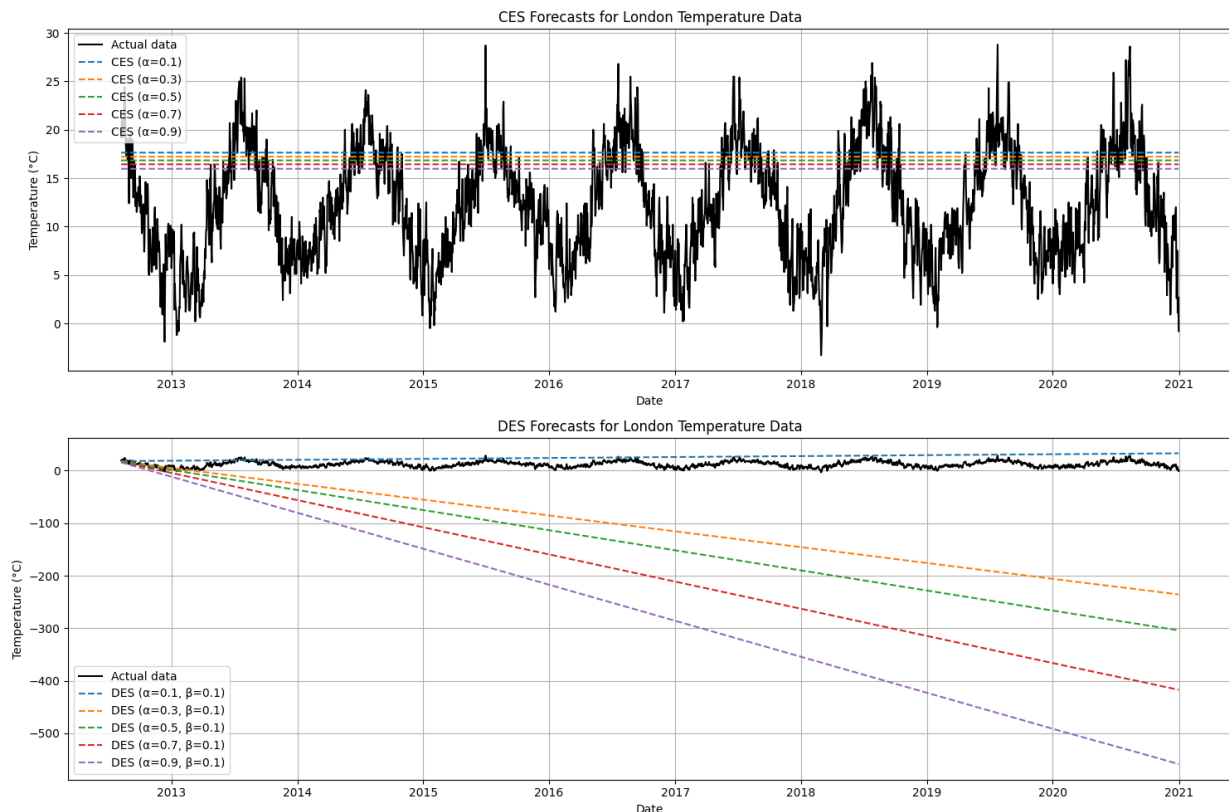
2. DES (Double Exponential Smoothing) :

```
def apply_des(data, alpha=0.3, beta=0.1, damped=True):
    damping_trend = 0.98 if damped else 1.0
    model = ExponentialSmoothing(data, trend='add', seasonal=None, damped=damped)
    model_fit = model.fit(smoothing_level=alpha, smoothing_trend=beta,
                          damping_trend=damping_trend, optimized=False)
    return model_fit
```

Langkah-langkah implementasi:

1. Membagi data menjadi data latih (80%) dan data uji (20%)
2. Menguji kedua metode dengan berbagai parameter ($\alpha = 0,1, 0,3, 0,5, 0,7, 0,9$ dan $\beta = 0,05, 0,1, 0,2$)
3. Menghitung RMSE (Root Mean Square Error) untuk setiap konfigurasi
4. Membandingkan kinerja antara kedua metode





Analisis Hasil

Dari Gambar, dapat diamati bahwa:

1. Kinerja CES:

- Kinerja terbaik dengan $\alpha = 0,1$ (nilai α yang lebih rendah bekerja lebih baik dengan data musiman)
- Hasil peramalan tampak sebagai garis horizontal (CES tidak dapat menangkap tren)
- Semua peramalan konvergen ke nilai konstan yang berbeda berdasarkan nilai α

2. Kinerja DES:

- Pada Gambar 1 dan 2, DES dengan $\alpha = 0,1$, $\beta = 0,05$ tampak sebagai konfigurasi terbaik
- Pada Gambar 3, beberapa model DES menunjukkan tren menurun yang eksplosif, menunjukkan potensi ketidakstabilan
- Faktor redaman (0,98) membantu menstabilkan peramalan tetapi tidak menghilangkan semua ketidakstabilan

3. Temuan Komparatif:

- CES umumnya mengungguli DES untuk dataset ini
- Model DES cenderung tidak stabil dengan nilai α dan β yang lebih tinggi
- Data suhu mengandung musiman yang kuat yang tidak sepenuhnya ditangkap oleh kedua model

3. Kesimpulan

- **Model Hukum Newton:** Model persamaan diferensial secara akurat merepresentasikan dinamika perubahan suhu, dengan parameter yang mengontrol laju perubahan. Model-model ini mengikuti pola eksponensial yang diharapkan untuk proses pendinginan dan pemanasan.
- **Simulasi Data Dunia Nyata:** Adaptasi Hukum Newton untuk simulasi suhu berkesinambungan memberikan model suhu per jam yang realistis berdasarkan rata-rata harian, mendemonstrasikan bagaimana persamaan diferensial dapat memodelkan fenomena alam.
- **Metode Peramalan: Untuk data suhu London:**
 - CES memberikan peramalan yang lebih stabil daripada DES
 - Parameter penghalusan yang lebih rendah ($\alpha = 0,1$) berkinerja lebih baik
 - Kedua metode tidak menangani musiman dengan tepat
 - DES dapat menjadi tidak stabil dengan parameter yang tidak tepat

4. Insight

- **Sensitivitas Parameter:** Baik model persamaan diferensial maupun metode peramalan menunjukkan sensitivitas tinggi terhadap pemilihan parameter. Bahkan perubahan kecil pada laju pendinginan/pemanasan atau parameter penghalusan dapat mengubah hasil secara dramatis.
- **Keterbatasan Model:**
 - Model Hukum Newton mengasumsikan kondisi lingkungan yang konstan
 - CES tidak dapat menangkap tren
 - DES dapat memodelkan tren tetapi mungkin menjadi tidak stabil
 - Kedua metode peramalan tidak mengatasi musiman dalam data suhu
- **Karakteristik Data: Data suhu London menunjukkan:**
 - Pola musiman yang kuat (siklus tahunan)
 - Variasi hari ke hari
 - Stabilitas jangka panjang (tidak ada tren pemanasan/pendinginan signifikan yang terlihat)

5. Rekomendasi

- **Untuk Pemodelan Persamaan Diferensial:**
 - Gunakan parameter laju adaptif yang berubah dengan kondisi (misalnya, laju pendinginan berbeda pada suhu yang lebih tinggi)
 - Masukkan komponen siklus harian untuk memodelkan variasi suhu dalam sehari
 - Pertimbangkan model yang lebih kompleks yang memperhitungkan faktor cuaca selain suhu lingkungan
- **Untuk Aplikasi Peramalan:**
 - Implementasikan Triple Exponential Smoothing (Holt-Winters) untuk menangkap musiman
 - Gunakan nilai α yang lebih rendah (0,1-0,3) untuk peramalan yang lebih stabil dengan data suhu

- Terapkan redaman pada model DES saat meramalkan suhu
- Pertimbangkan metode lanjutan seperti SARIMA atau pendekatan machine learning untuk akurasi yang lebih baik

- **Untuk Penelitian Lebih Lanjut:**

- Jelajahi hubungan antara parameter persamaan diferensial dan parameter exponential smoothing
- Selidiki model hybrid yang menggabungkan persamaan fisika dengan peramalan statistik
- Analisis bagaimana perubahan iklim mempengaruhi stabilitas model peramalan suhu
- Uji model dengan data dari berbagai wilayah iklim untuk menilai generalisasinya