EXPLORATORY DATA ANALYSIS : PREPROCESSING DATA

LAPORAN

Disusun oleh :
Nama :
NPM :

Dosen :
JOKO TRILOKA, S.Kom., MT., PH.D

PROGRAM STUDI MAGISTER TEKNIK INFORMATIKA
FAKULTAS ILMU KOMPUTER

INSTITUT INFORMATIKA DAN BISNIS DARMAJAYA

2024

1. CLEAN MISSING DATA

Langkah pertama yang dilakukan adalah membuat dan memplot vektor data yang masih acak, yang terdiri dari empat nilai NaN dan lima nilai pencilan.

```
x = 1:100;
data = cos(2*pi*0.05*x+2*pi*rand) + 0.5*randn(1,100);
data(20:20:80) = NaN;
data(10:20:90) = [-50 40 30 -45 35];
```

x = 1:100;

- Baris ini mendefinisikan vektor x yang terdiri dari angka dari 1 hingga 100.
- 1:100 adalah ekspresi MATLAB yang menghasilkan vektor baris dengan elemen mulai dari 1 hingga 100 dengan selisih 1.

data = cos(2*pi*0.05*x + 2*pi*rand) + 0.5*randn(1,100);

- Bagian ini mendefinisikan vektor data berisi 100 elemen yang dibangkitkan menggunakan fungsi trigonometri dan tambahan komponen acak.
- Cos (2*pi*0.05*x + 2*pi*rand) menghasilkan nilai sinusoidal berdasarkan vektor x, dimana frekuensi sudutnya 0.05 (memberikan variasi periodik lambat). Fungsi rand digunakan untuk menambahkan fase acak ke gelombang sinusoidal tersebut.
- 0.5*randn (1,100) menambahkan *noise* Gaussian ke data sinusoidal dengan distribusi normal acak (mean 0, standar deviasi 0.5). randn (1,100) menghasilkan array 1x100 nilai acak dari distribusi normal.
- Hasil dari kedua komponen tersebut digabungkan untuk menghasilkan vektor data yang memiliki karakteristik berisik dan berosilasi.

data(20:20:80) = NaN;

- Baris ini menggantikan beberapa elemen dalam vektor data dengan nilai NaN (Not a Number), yang digunakan untuk merepresentasikan data yang hilang atau tidak valid.
- 20:20:80 adalah indeks yang dipilih dalam vektor data, yang berarti elemen ke-20, ke-40, ke-60, dan ke-80 dari vektor data akan diset menjadi NaN.

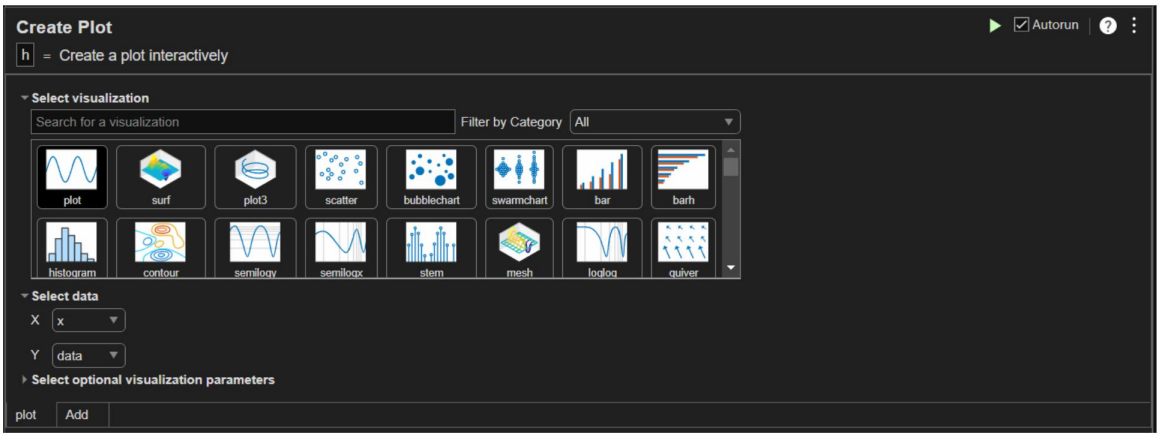
data(10:20:90) = [-50 40 30 -45 35];

- Baris ini menggantikan beberapa elemen pada posisi tertentu dalam vektor data dengan *outliers* atau nilai pencilan.
- 10:20:90 berarti elemen ke-10, ke-30, ke-50, ke-70, dan ke-90 dari vektor data akan diubah.
- Nilai-nilai [-50, 40, 30, -45, 35] akan dimasukkan ke elemen-elemen tersebut, di mana nilai ini dianggap jauh lebih besar atau lebih kecil dibandingkan dengan data lainnya, sehingga dianggap sebagai pencilan.

Maka:

- **Vektor x**: Berisi nilai dari 1 hingga 100.
- **Vektor data**: Berisi kombinasi nilai sinusoidal yang ditambah dengan *noise*, beberapa elemen dengan nilai *NaN*, dan beberapa elemen yang merupakan pencilan (*outliers*).
- Fungsi ini digunakan untuk membuat data yang berisik dan mengandung ketidaksesuaian (seperti data hilang atau pencilan), yang selanjutnya bisa dibersihkan atau dianalisis lebih lanjut dalam MATLAB.

Setting untuk plotting dapat dilihat pada Gambar 1.



Gambar 1 Plotting data

Gambar 1 ini menunjukkan antarmuka MATLAB *Live Editor* untuk tugas **Create Plot**. Tugas ini memungkinkan pengguna membuat plot secara interaktif tanpa harus menulis kode secara manual. Berikut adalah penjelasan detail mengenai elemen-elemen yang ditampilkan dalam gambar:

1. Create Plot Interactively

- Bagian ini memberikan opsi untuk membuat plot secara interaktif melalui *Live Editor*.
- Pengguna bisa memilih jenis visualisasi, memasukkan data, dan mengatur parameter visualisasi tambahan.

2. Select Visualization (Pilih Visualisasi)

- **Search for a Visualization**: Ini adalah kolom pencarian yang memungkinkan pengguna untuk mencari jenis visualisasi tertentu dengan mengetikkan kata kunci seperti "plot", "scatter", "histogram", dan sebagainya.
- **Filter by Category**: Opsi ini menyediakan filter untuk memudahkan pencarian dengan kategori yang lebih spesifik. Secara default, pengaturan ini diatur pada "All", yang menampilkan semua jenis visualisasi yang tersedia.
- **Jenis-jenis Visualisasi yang Tersedia**:
    - **Plot**: Plot garis sederhana untuk dua variabel (x dan y).
    - **Surf (Surface Plot)**: Menampilkan plot permukaan 3D.
    - **Plot3**: Plot 3D untuk tiga variabel (x, y, dan z).

- **Scatter**: Menampilkan plot sebar (scatter plot), biasanya digunakan untuk melihat hubungan antara dua variabel.
- **Bubblechart**: Plot sebar yang menggunakan ukuran gelembung untuk mewakili dimensi ketiga.
- **Swarmchart**: Sebuah jenis plot distribusi data.
- **Bar**: Grafik batang.
- **Histogram**: Menampilkan distribusi frekuensi data.
- **Contour**: Plot kontur untuk menggambarkan fungsi 2D.
- **Semilog**: Plot dengan salah satu sumbu dalam skala logaritmik.
- **Stem**: Plot batang vertikal dari titik-titik data.
- **Loglog**: Plot dengan kedua sumbu dalam skala logaritmik.
- **Quiver**: Menampilkan panah untuk menggambarkan vektor.
- **Contourf:** Visualisasi kontur *filled* (diisi warna) yang menggambarkan grafik 3D dalam bentuk kontur 2D.
- **Errorbar:** Menampilkan nilai rata-rata serta margin kesalahan atau variabilitas data dalam bentuk batang kesalahan (*error bars*)
- **Scatter3:** *Scatter plot* dalam 3D yang menggambarkan hubungan antara tiga variabel dengan titik-titik dalam ruang tiga dimensi
- **Bubblechart3:** Variasi dari *scatter plot* 3D di mana ukuran titik bervariasi untuk mewakili dimensi tambahan, selain dari koordinat X, Y, dan Z.
- **Swarmchart3:** digunakan untuk menggambarkan distribusi data yang sangat padat
- **Area:** Grafik area yang digunakan untuk menggambarkan perubahan nilai kumulatif dari data.
- **Pcolor:** Menampilkan grafik grid yang diisi dengan warna yang mewakili intensitas atau nilai data di titik-titik grid tertentu.
- **Stairs:** Membuat plot tangga atau tangga blok (step plot) yang menampilkan perubahan diskrit dalam data.
- **Stackedplot:** Plot bertumpuk yang memungkinkan beberapa variabel dipetakan secara vertikal pada satu grafik,
- **Quiver3:** Grafik vektor 3D yang menggambarkan arah dan magnitudo dari vektor di ruang tiga dimensi.
- **Polarplot:** Plot polar digunakan untuk menggambarkan data dalam koordinat polar, di mana sudut dan radius menentukan posisi titik.

- **Surfc:** Plot permukaan 3D dengan kontur di dasar grafik. Ini menggabungkan grafik permukaan dan kontur untuk memberikan pemahaman yang lebih baik tentang variasi data.
- **Animatedline:** Plot garis yang bisa dianimasikan, cocok untuk menggambarkan data yang berubah dari waktu ke waktu. Ini berguna untuk visualisasi waktu nyata atau simulasi dinamis.
- **Contour3:** Plot kontur 3D yang digunakan untuk menggambarkan garis kontur dalam grafik tiga dimensi, memberikan tampilan berbagai level dari perspektif 3D.
- **Heatmap:** Representasi visual dari data matriks di mana nilai diwakili oleh warna.
- **Parallelplot:** Digunakan untuk menggambarkan hubungan antara beberapa variabel dalam satu grafik.
- **Bar3:** Grafik batang 3D untuk menampilkan data dalam bentuk batang dalam tiga dimensi.
- **Bar3h:** Grafik batang horizontal 3D, mirip dengan bar3 namun batang diplot secara horizontal.
- **Comet:** Plot yang menampilkan data sebagai garis dengan "ekor" yang bergerak, sehingga menghasilkan efek visual mirip komet yang bergerak di sepanjang jalur data.
- **Meshc:** Grafik mesh (jaring) 3D dengan kontur, yang menggabungkan tampilan permukaan 3D dan kontur di dasar grafik,
- **Surfl:** Mirip dengan surf, tetapi dengan tambahan efek pencahayaan untuk memberikan kedalaman pada plot permukaan 3D. Ini membantu untuk menonjolkan fitur topografis atau variasi ketinggian.
- **Plotmatrix:** Membuat matriks plot yang memungkinkan banyak grafik scatter antara beberapa variabel. Sangat berguna untuk melihat korelasi antar banyak variabel dalam satu tampilan.
- **Compass:** Menampilkan panah yang digunakan untuk menggambarkan vektor 2D.
- **Waterfall:** Plot garis 3D yang disusun secara bertahap atau berurutan, sering digunakan untuk menggambarkan perubahan fungsi dalam beberapa interval.
- **Stem3:** Versi 3D dari plot batang (stem plot), yang menunjukkan nilai diskrit dalam ruang 3D dengan garis vertikal dari titik data ke sumbu dasar.
- **Comet3:** Versi 3D dari grafik komet, di mana ekor komet bergerak di sepanjang jalur dalam tiga dimensi.
- **Histogram2:** Histogram 2D, menampilkan distribusi data dalam dua dimensi.

- **Meshz:** Grafik mesh 3D dengan sumbu Z yang diperpanjang sampai ke dasar grafik, menghasilkan efek "tiang" untuk lebih jelas menonjolkan nilai Z.
- **Geoplot:** Digunakan untuk memvisualisasikan data geografis pada peta.
- **Feather:** Grafik vektor yang menampilkan panah pada titik data tertentu, menggambarkan besaran dan arah vektor.
- **Geobubble**: Versi peta gelembung yang memetakan data geografis dengan menggunakan gelembung yang ukurannya mewakili variabel tambahan pada peta geografis.
- **Ribbon:** Grafik pita (ribbon plot) 3D yang menunjukkan data sebagai pita yang mengikuti jalur tertentu, sangat cocok untuk menggambarkan perubahan nilai yang terus-menerus dalam 3D.
- **Geoscatter:** Plot penyebaran data geografis.
- **Scatterhistogram**: Kombinasi scatter plot dan histogram.
- **Polarhistogram**: Histogram dalam koordinat polar.
- **Polarscatter**: Plot penyebaran dalam koordinat polar.
- **Polarbubblechart**: Scatter plot polar dengan ukuran data variabel.
- **Contourslice**: Potongan kontur 3D data.
- **Geodensityplot**: Plot kepadatan geografis.
- **Binscatter**: Scatter plot yang dibagi dalam bin (grup).
- **Boxchart**: Diagram kotak (box plot) untuk menunjukkan distribusi data.
- **Imagesc**: Representasi gambar dalam skala warna.
- **Image**: Plot gambar dua dimensi.
- **Fplot**: Plot fungsi matematis.
- **Pie**: Diagram lingkaran.
- **Fsurf**: Permukaan fungsi 3D.
- **Fimplicit**: Plot fungsi implisit.
- **Fcontour**: Plot kontur fungsi dua dimensi.
- **Fplot3**: Plot fungsi 3D.
- **Wordcloud**: Visualisasi kata-kata dalam bentuk awan, berdasarkan frekuensi atau kepentingan.
- **Pie3**: Diagram lingkaran 3D.
- **Pareto**: Diagram pareto, menampilkan distribusi kumulatif.
- **Fimplicit3**: Plot fungsi implisit dalam tiga dimensi.

- **Fmesh**: Mesh plot tiga dimensi untuk visualisasi permukaan.
- **Ezpolar**: Plot data dalam koordinat polar secara sederhana.
- **Bodeplot**: Plot Bode untuk sistem kontrol, menunjukkan respons frekuensi.
- **Nicholsplot**: Plot Nichols, digunakan dalam analisis kontrol.
- **Nyquistplot**: Plot Nyquist untuk analisis stabilitas sistem kontrol.
- **Andrewsplot**: Visualisasi multidimensi menggunakan kurva Andrews.
- **Freqz**: Respons frekuensi dari filter digital.
- **Glyphplot**: Plot simbol grafis untuk representasi visual data multidimensi.
- **Grpdelay**: Plot delay grup dari suatu sistem.
- **Hsvplot**: Plot HSV untuk representasi warna.
- **Impulseplot:** Plot respons impuls dari suatu sistem.
- **Impz**: Plot respons impuls dari sistem digital.
- **Iopzplot**: Plot zeros dan poles dari sistem kontrol.
- **Periodogram**: Menampilkan spektrum kekuatan sinyal berdasarkan periodogram.
- **Pspectrum**: Plot spektrum kekuatan sinyal.
- **Pwelch**: Estimasi spektrum kekuatan menggunakan metode Welch.
- **Pzplot**: Plot poles dan zeros untuk sistem kontrol.
- **Rlocusplot**: Plot locus akar untuk analisis stabilitas sistem kontrol.
- **Sigmaplot**: Menampilkan singular value plot.
- **Spectrogram**: Visualisasi frekuensi sinyal seiring waktu.
- **Stepplot**: Plot respons tangga (step response) dari suatu sistem.
- **Zplane**: Plot poles dan zeros dari sistem diskrit dalam bidang Z.

3. Select Data (Pilih Data)

- Pada bagian ini, pengguna memilih data yang akan dipetakan pada sumbu x dan sumbu y.
- Dalam contoh gambar, ada dua dropdown:
  - **X**: Memilih data yang akan dipetakan di sumbu X.
  - **Y**: Memilih data yang akan dipetakan di sumbu Y.
- Pada gambar, x telah dipilih sebagai sumbu X, dan data sebagai sumbu Y. x adalah vektor dari 1 hingga 100 yang didefinisikan sebelumnya, dan data adalah vektor data dengan *noise* dan pencilan yang telah dibuat.

4. Optional Visualization Parameters (Parameter Visualisasi Tambahan)

- Bagian ini menyediakan opsi tambahan untuk menyesuaikan tampilan plot, seperti mengganti warna, mengatur jenis garis, atau menambahkan label sumbu. Namun, pada gambar ini, tidak ada parameter tambahan yang dipilih atau ditampilkan.

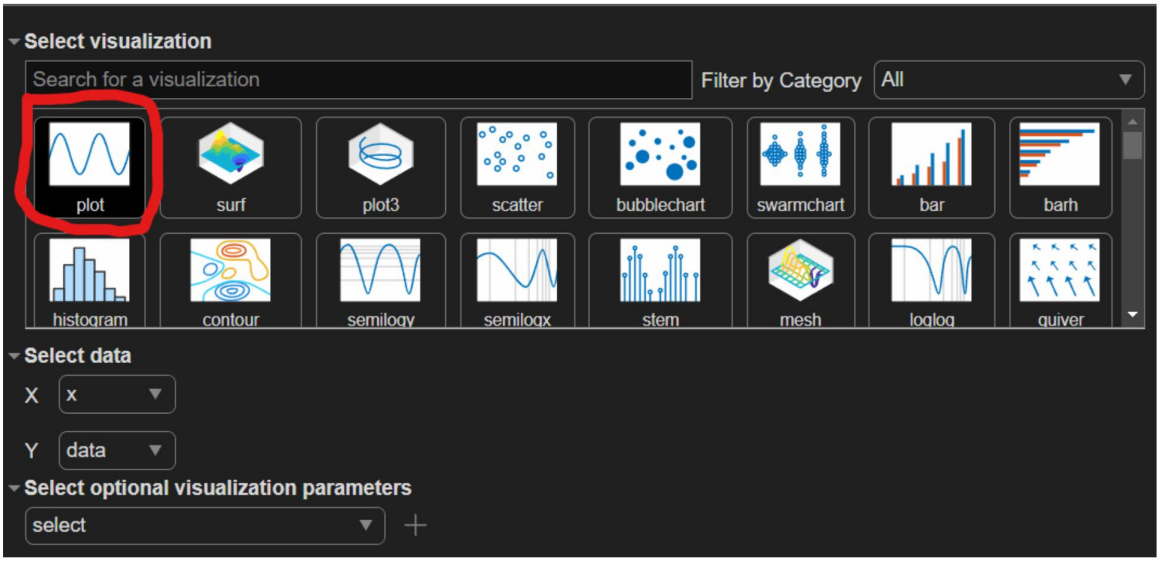
5. Plot and Add Button

- **Plot**: Setelah pengguna selesai memilih visualisasi dan data yang diinginkan, mereka dapat mengklik tombol ini untuk memvisualisasikan data dalam bentuk grafik.
- **Add**: Jika pengguna ingin menambahkan lebih banyak elemen ke dalam visualisasi, mereka bisa menggunakan tombol ini untuk melanjutkan.

6. Run and Autorun (di Pojok Kanan Atas)

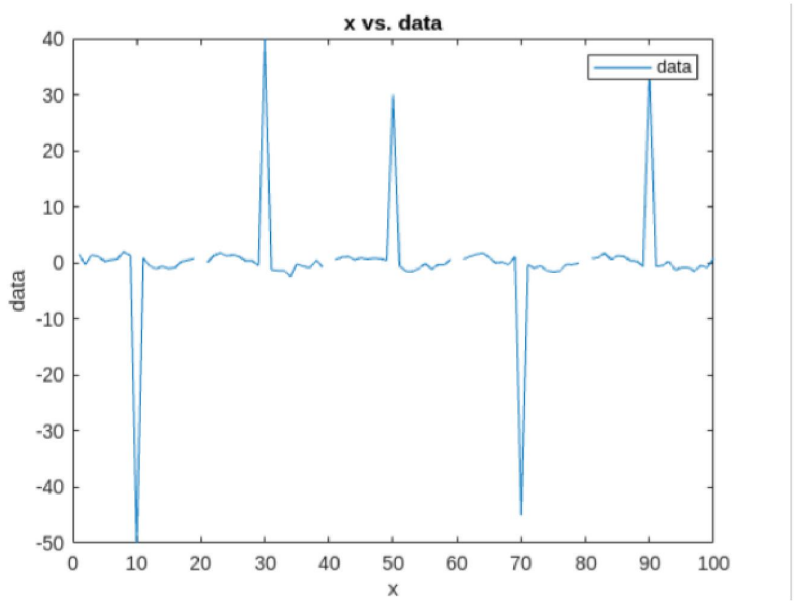
- **Run (Tombol Play)**: Tombol ini digunakan untuk menjalankan skrip dan menampilkan hasil plot.
- **Autorun**: Jika diaktifkan, MATLAB akan secara otomatis menjalankan dan memperbarui plot setiap kali ada perubahan pada skrip atau parameter yang dipilih.

Setting untuk plotting (plot) dapat dilihat pada Gambar 2.



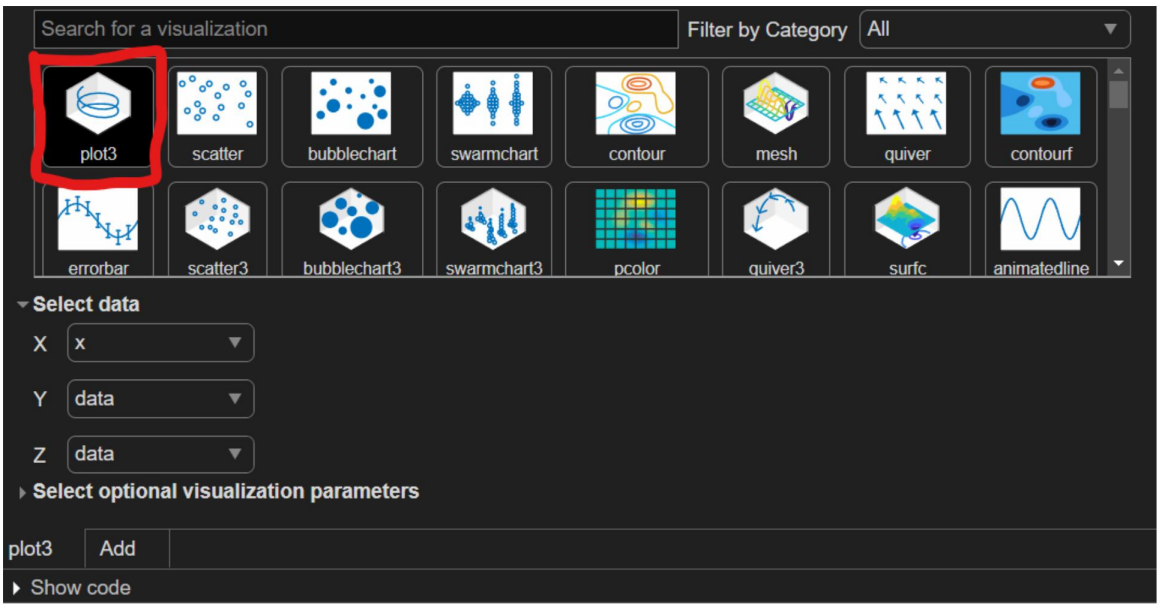
Gambar 2 Plotting data plot

Berikut ini adalah hasil dari plotting data jenis (plot) nilai x dan data dapat dilihat pada Gambar 3.



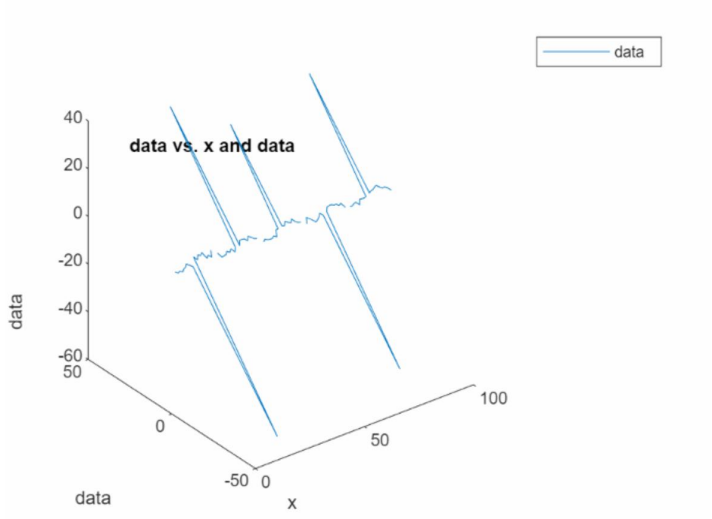
Gambar 3. Hasil plotting nilai x dan data

Setting untuk plotting (plot3) dapat dilihat pada Gambar 4.



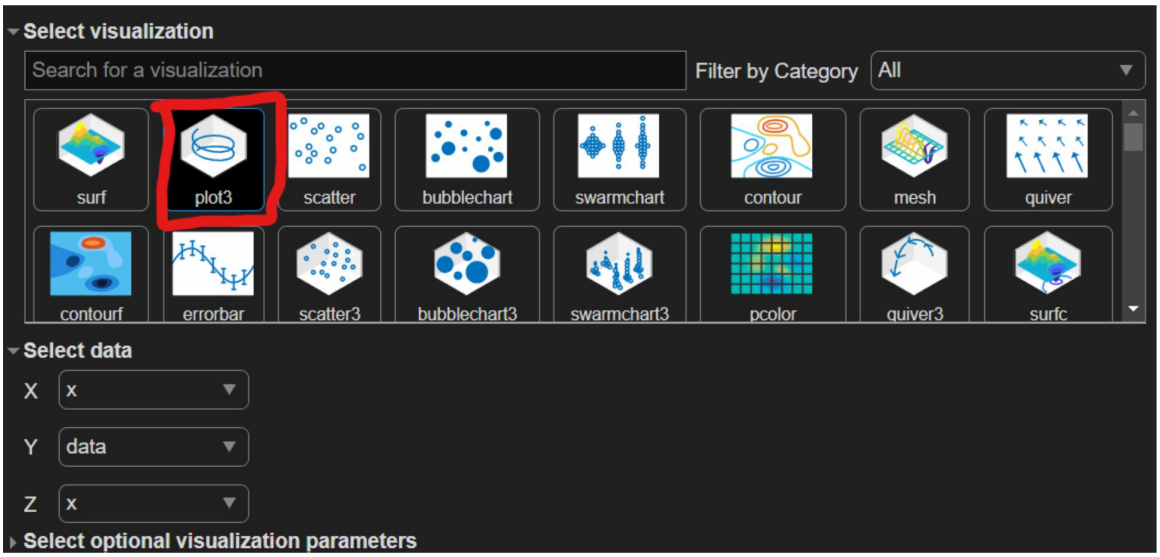
Gambar 4 Plotting data plot3

Berikut ini adalah hasil dari plotting data jenis (plot3) data vs, x dan data dapat dilihat pada Gambar 5.



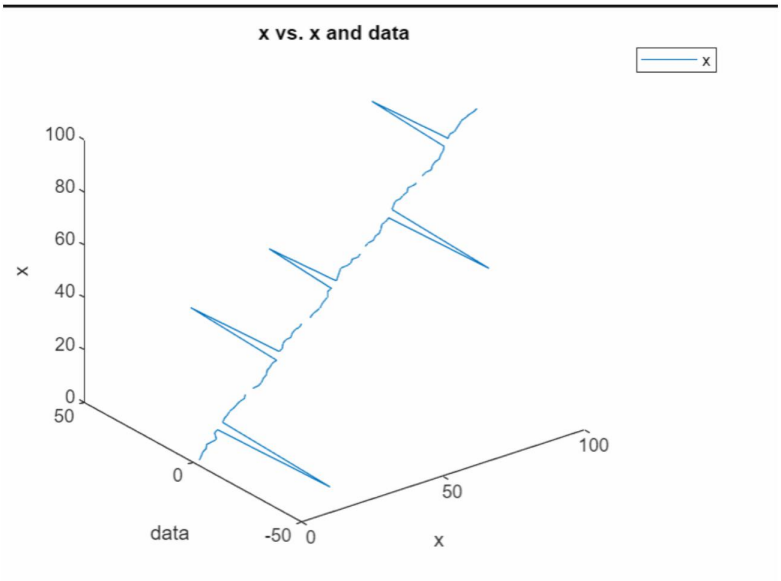
Gambar 5. Hasil plotting data vs, x dan data

Setting untuk plotting (plot3) dapat dilihat pada Gambar 6.



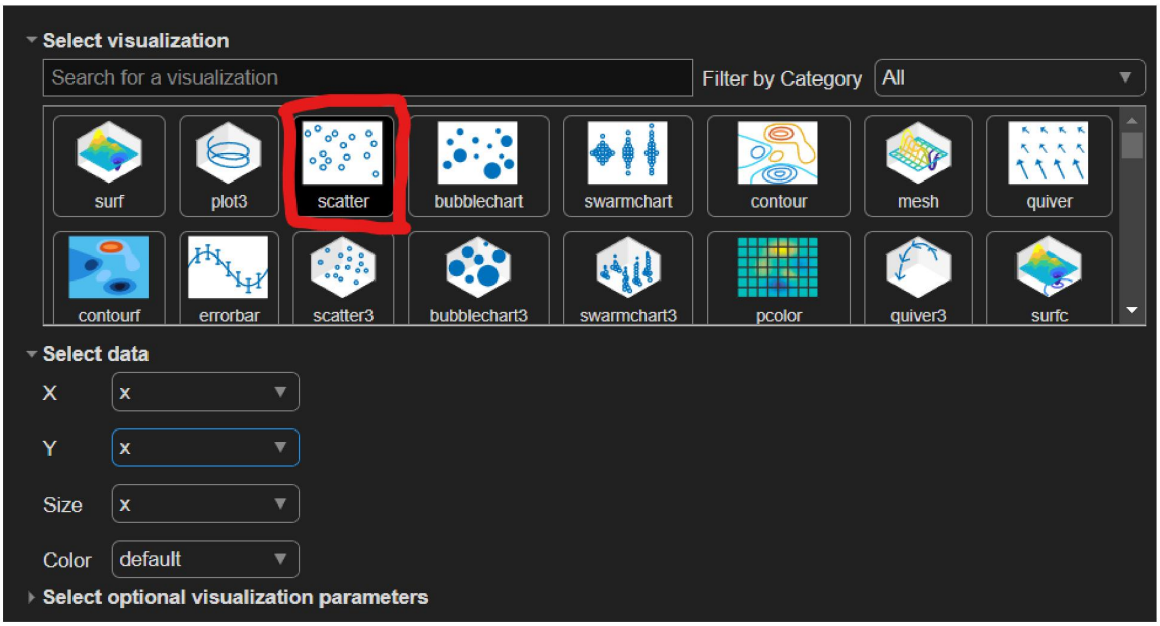
Gambar 6 Plotting data plot3

Berikut ini adalah hasil dari plotting data jenis (plot3) x vs, x dan data dapat dilihat pada Gambar 7.



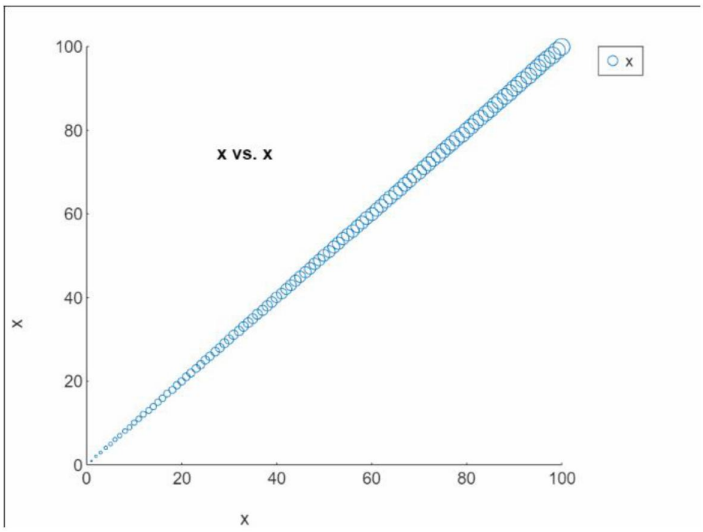
Gambar 7. Hasil plotting data vs, x dan x

Setting untuk plotting (Scatter) dapat dilihat pada Gambar 8.



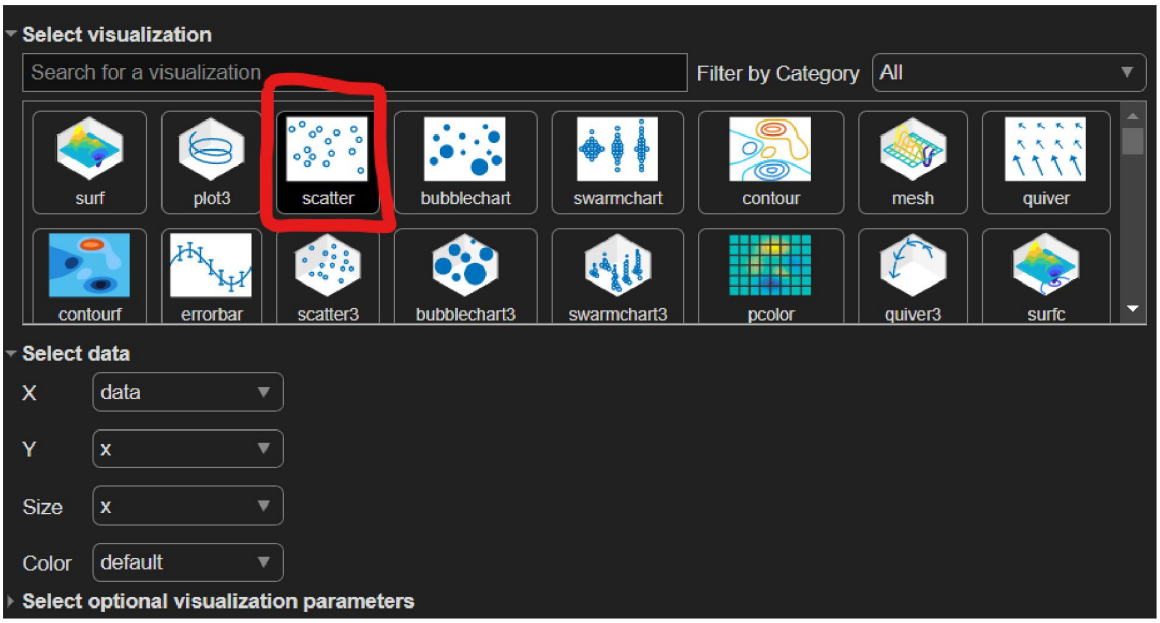
Gambar 8 Plotting data scatter

Berikut ini adalah hasil dari plotting data jenis (Scatter) x vs, x dapat dilihat pada Gambar 9.



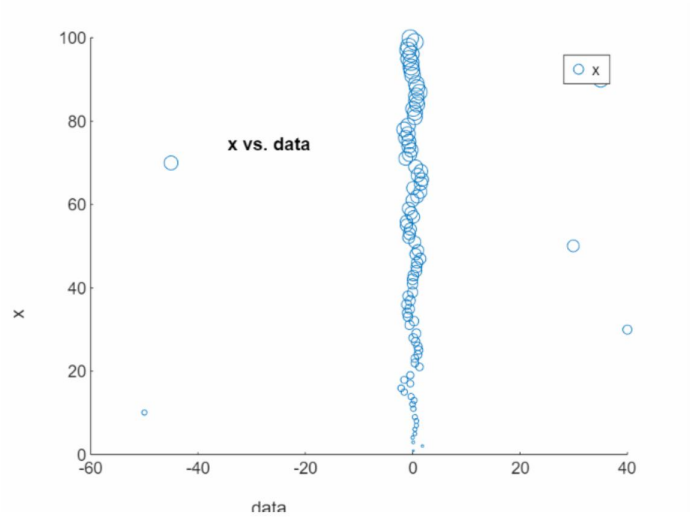
Gambar 9. Hasil plotting x vs, x

Setting untuk plotting (Scatter) dapat dilihat pada Gambar 10.



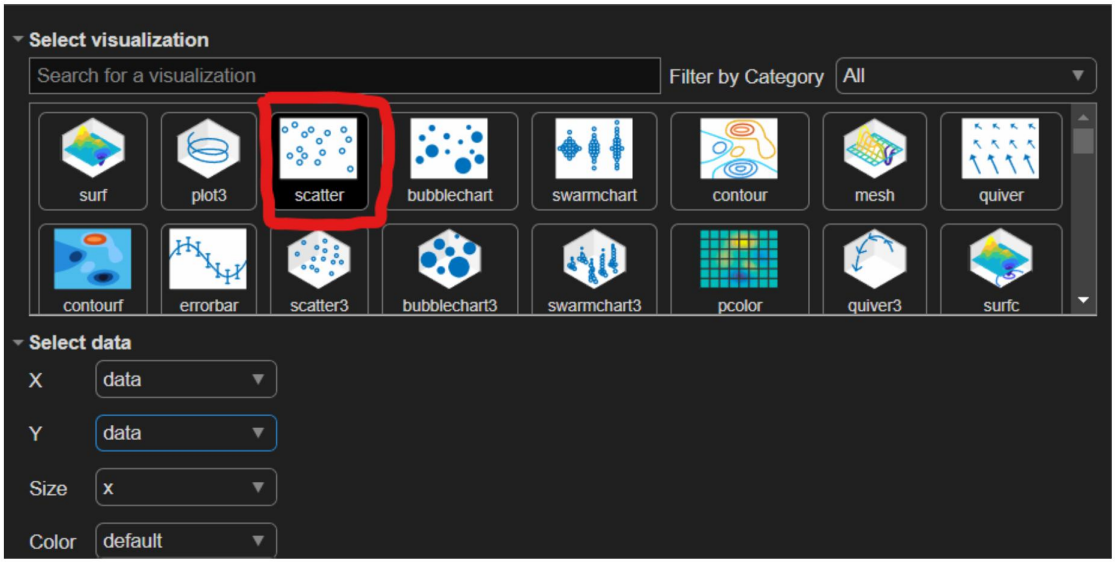
Gambar 10 Plotting data scatter

Berikut ini adalah hasil dari plotting data jenis (Scatter) x vs, data dapat dilihat pada Gambar 11.



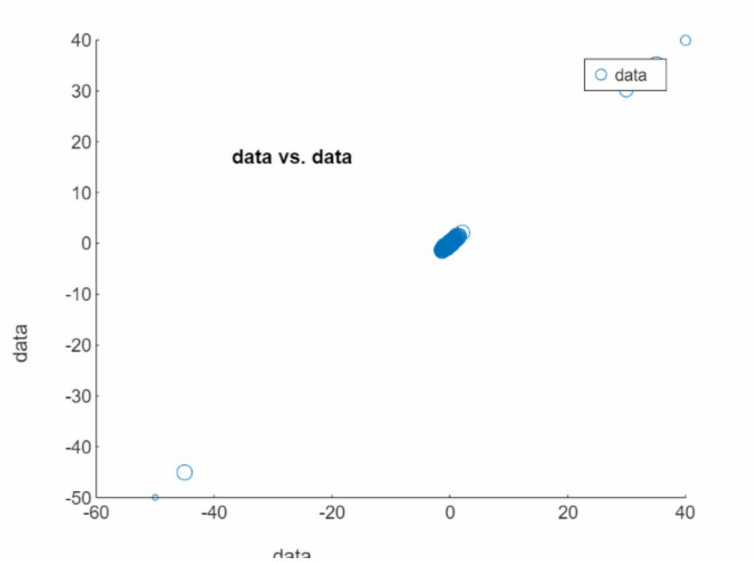
Gambar 11. Hasil plotting x vs, data

Setting untuk plotting (Scatter) dapat dilihat pada Gambar 12.



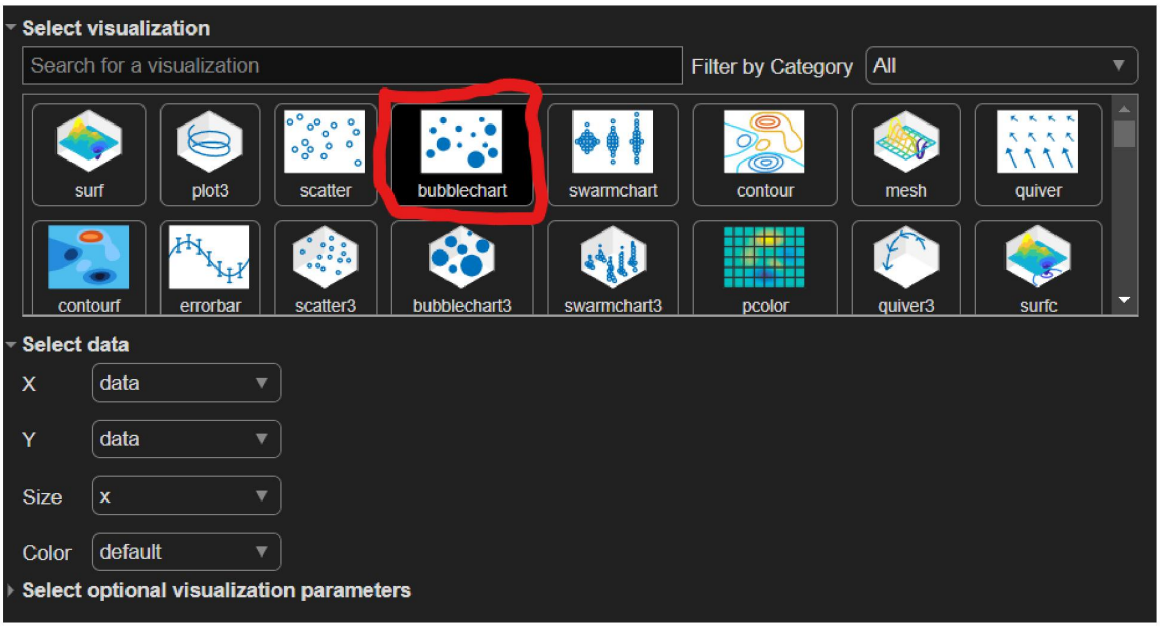
Gambar 12 Plotting data scatter

Berikut ini adalah hasil dari plotting data jenis (Scatter) data vs, data dapat dilihat pada Gambar 13.



Gambar 13. Hasil plotting data vs, data

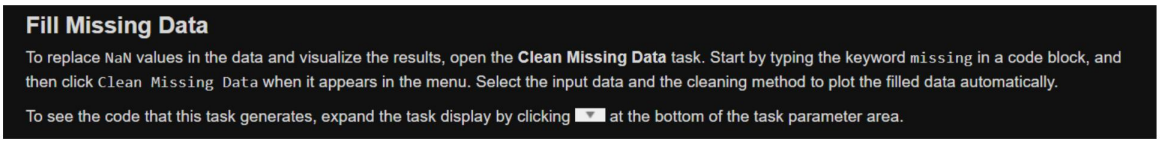
Setting untuk plotting (bubblechart) dapat dilihat pada Gambar 14.



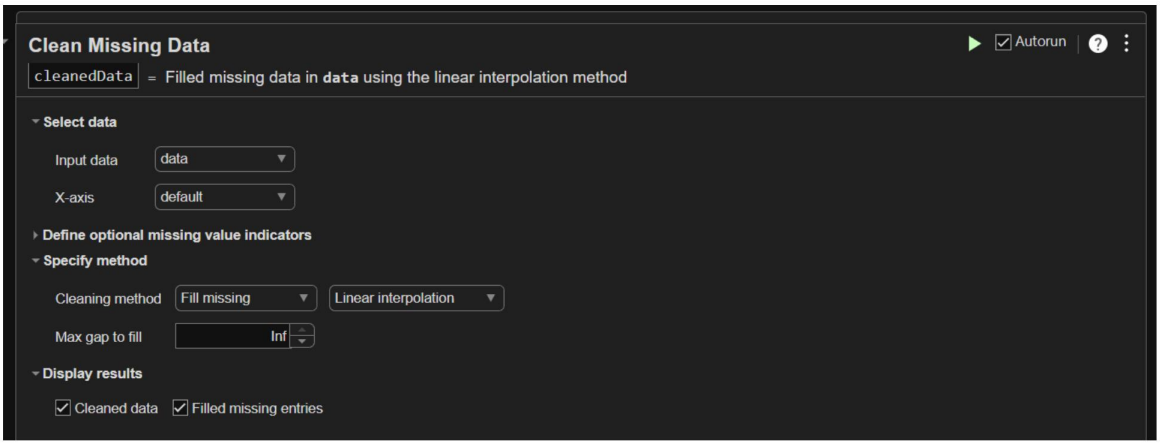
Gambar 14 Plotting data bubblechart

Berikut ini adalah hasil dari plotting data jenis (bubblechart) data vs, data dapat dilihat pada Gambar 15.

2. FILL MISSING DATA



Ini berisikan tentang cara menggunakan task **Clean Missing Data** untuk mengisi data yang hilang secara otomatis di MATLAB, serta cara melihat kode yang dihasilkan oleh MATLAB untuk melakukan pembersihan tersebut.

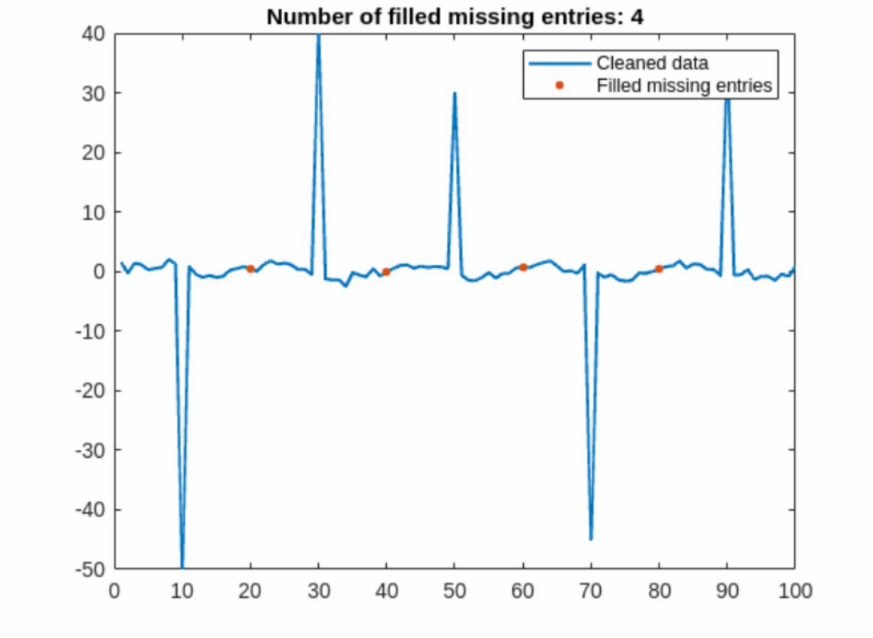


Gambar 15. Clean missing data

Berikut penjelasan setiap bagian:

1. **Input data**:
    - Pilihan ini memungkinkan Anda untuk memilih data masukan yang ingin dibersihkan dari missing data (nilai yang hilang). Anda dapat memilih data dari workspace MATLAB.
2. **X-axis**:
    - Bagian ini memungkinkan Anda menentukan variabel sumbu X jika data Anda berbentuk dua dimensi atau lebih. Dalam hal ini, X-axis ditentukan sebagai default.
3. **Define optional missing value indicators**:
    - Bagian ini biasanya digunakan untuk menentukan indikator lain yang mungkin digunakan untuk menandakan data hilang selain NaN (Not a Number).
4. **Specify method**:
    - **Cleaning method**: Diatur ke "Fill missing", yang artinya Anda memilih untuk mengisi data yang hilang.
    - **Linear interpolation**: Ini adalah metode interpolasi yang digunakan untuk mengisi data yang hilang. Metode ini menghitung nilai di antara dua titik data yang ada menggunakan pendekatan linier.
    - **Max gap to fill**: Ini menentukan berapa banyak celah data yang boleh diisi. Jika ada celah yang terlalu besar, MATLAB akan melewatkan pengisian. Di sini diatur ke 'Inf' yang artinya tidak ada batasan untuk jarak celah yang dapat diisi.
5. **Display results**:

- Terdapat dua kotak pilihan:
  - **Cleaned data**: Jika dicentang, hasil dari data yang telah dibersihkan akan ditampilkan.
  - **Filled missing entries**: Jika dicentang, ini menunjukkan bahwa entri yang hilang telah diisi.

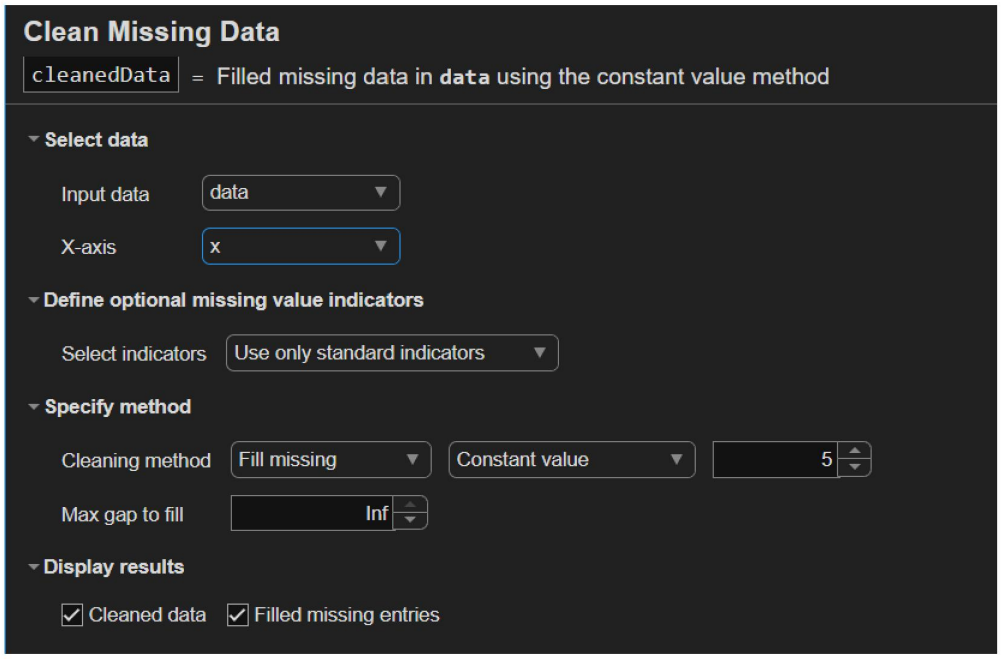


Gambar 16. Hasil Clean missing data

Ini adalah hasil dari proses **Clean Missing Data**, yang digunakan untuk mengisi data yang hilang dalam dataset. Berikut adalah penjelasan rincinya:

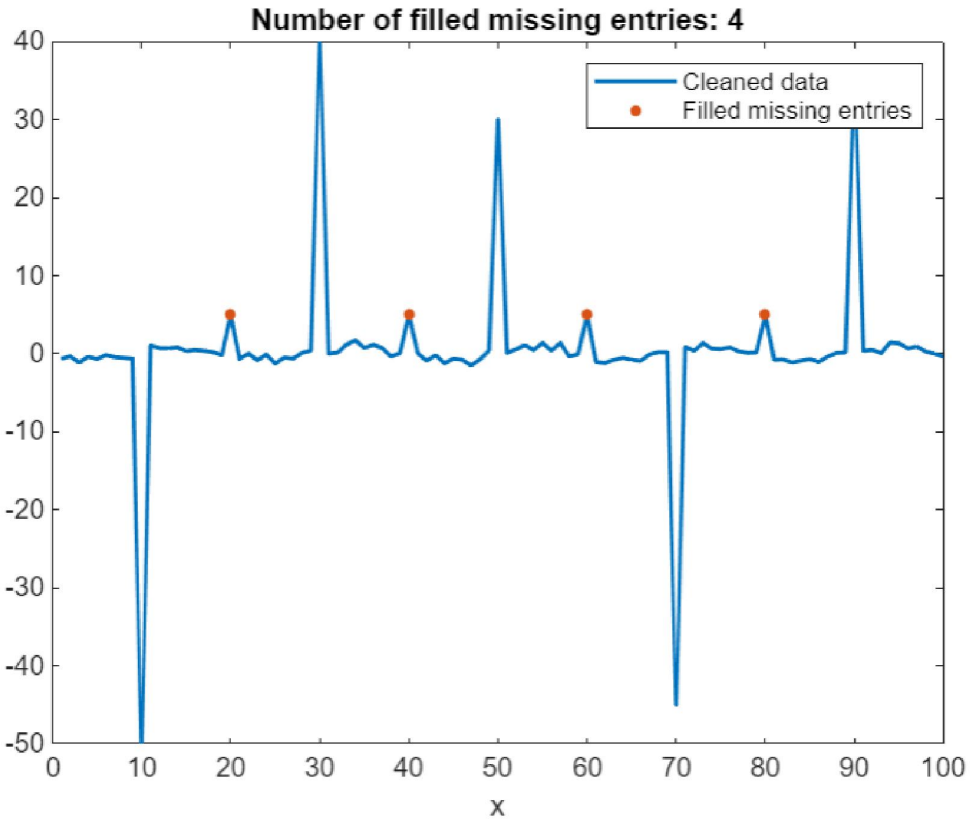
1. **Judul Grafik**:
   - Judul grafik menunjukkan bahwa ada **4 missing entries** yang telah diisi atau diperbaiki dalam data, ditampilkan sebagai "Number of filled missing entries: 4".
2. **Sumbu X dan Y**:
   - **Sumbu X** mewakili indeks atau posisi data dalam dataset (dari 0 hingga 100).
   - **Sumbu Y** mewakili nilai data pada titik tertentu dalam dataset, yang berkisar dari -50 hingga 40.
3. **Garis Biru (Cleaned Data)**:
   - Garis biru di grafik ini menunjukkan data asli yang sudah dibersihkan, setelah nilai yang hilang diisi. Kita bisa melihat fluktuasi data pada beberapa titik.
4. **Titik Oranye (Filled Missing Entries)**:
   - Titik-titik oranye menunjukkan lokasi di mana nilai-nilai yang hilang (missing data) telah diisi. Pada grafik ini, Kita bisa melihat bahwa ada 4 titik oranye di sepanjang garis biru, menunjukkan 4 nilai yang diisi di posisi tersebut.

5. **Keterangan (Legend)**:
    - Keterangan di bagian atas kanan grafik menunjukkan dua elemen:
        - **Cleaned data**: Menunjukkan data yang sudah dibersihkan, direpresentasikan oleh garis biru.
        - **Filled missing entries**: Titik-titik oranye yang menandai lokasi-lokasi di mana data yang hilang telah diisi.



Gambar 17. Clean missing data Constant value

1. **Constant Value**:
    - Opsi **Constant value** dipilih sebagai metode untuk mengisi data yang hilang. Ini berarti bahwa setiap data yang hilang dalam dataset akan diisi menggunakan nilai yang sama, yaitu nilai konstan yang ditentukan oleh pengguna. Di sini, nilai konstan yang dipilih adalah angka **5**.
2. **Angka 5**:
    - Angka **5** yang dipilih menunjukkan bahwa semua nilai yang hilang dalam dataset akan digantikan atau diisi dengan angka **5**. Artinya, setiap kali MATLAB mendeteksi data yang hilang (misalnya nilai NaN), program akan mengganti nilai tersebut dengan angka **5**.

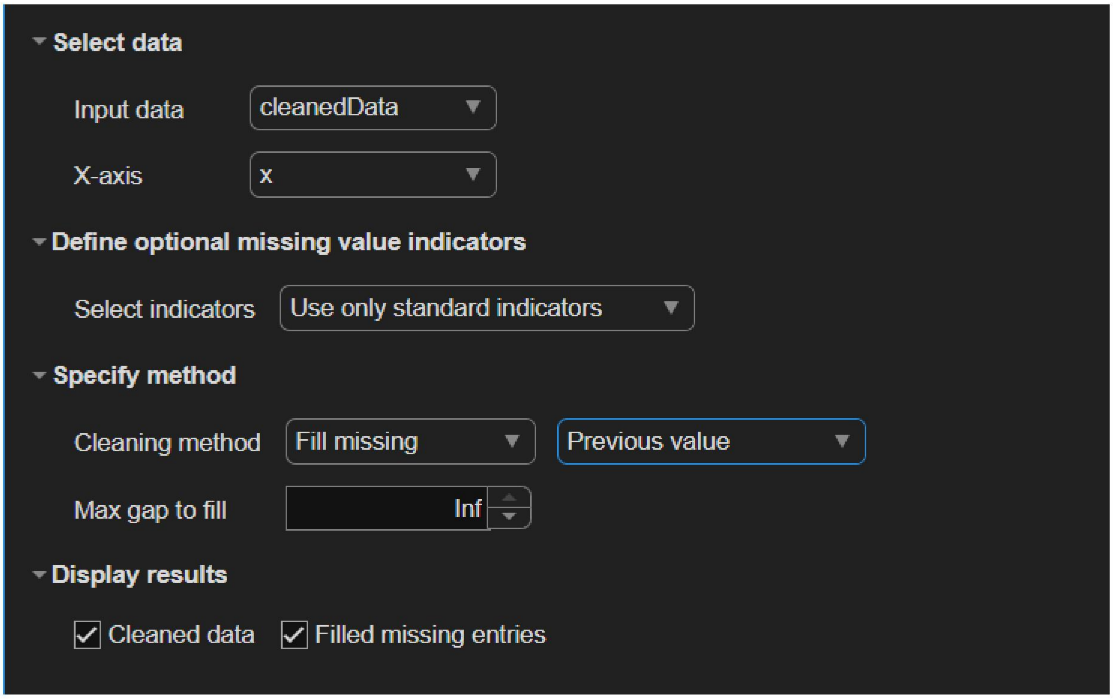


Gambar 18. Hasil Clean missing data Constant value

1. **Judul Grafik**:
   - Judul di bagian atas grafik menyatakan **"Number of filled missing entries: 4"**. Ini berarti ada **4 nilai yang hilang** dalam dataset awal yang telah diisi (dalam contoh ini menggunakan metode pengisian dengan nilai konstan).
2. **Sumbu X**:
   - Sumbu horizontal (X-axis) diberi label **"x"**, yang menunjukkan nilai data pada sumbu X, yang biasanya digunakan sebagai penanda posisi data.
3. **Sumbu Y**:
   - Sumbu vertikal (Y-axis) menunjukkan nilai dari data asli serta data yang telah diisi setelah pembersihan. Nilai-nilai pada sumbu ini mencerminkan ukuran data atau perubahan nilai dalam dataset.
4. **Garis Biru – Cleaned Data**:
   - Garis biru pada grafik merepresentasikan **data yang telah dibersihkan**. Artinya, setelah proses pembersihan data dijalankan, data yang awalnya hilang atau tidak ada (misalnya NaN) telah diisi. Ini adalah visualisasi keseluruhan dataset setelah nilai yang hilang diisi.

- Anda dapat melihat beberapa titik yang sangat rendah (turun drastis) dan tinggi yang kemungkinan besar adalah hasil dari *spike* atau *drop* data asli yang hilang.

5. **Titik Merah – Filled Missing Entries**:
   - Titik-titik merah pada grafik menandai **posisi dari data yang hilang** dan yang telah diisi. Ada 4 titik merah yang sesuai dengan 4 entri yang hilang di dataset asli, yang kemudian diisi dengan nilai konstan (misalnya, angka 5 sesuai dengan gambar sebelumnya).
6. **Interpretasi**:
   - Dataset yang ditampilkan awalnya memiliki 4 nilai hilang. Setelah menjalankan proses pembersihan dengan metode *fill missing* (mengisi nilai yang hilang dengan nilai konstan), keempat titik yang hilang tersebut diisi, yang ditunjukkan dengan titik merah.
   - Posisi dari titik merah menunjukkan di mana data hilang di sepanjang sumbu X.
   - Garis biru yang menghubungkan titik-titik ini mewakili seluruh dataset setelah nilai yang hilang diisi.



Gambar 19. Clean missing data Previous Value

Cleaned Data:

1. **Cleaned Data**:
   - Pada bagian *Select Data*, input data yang dipilih adalah **cleanedData**, yang berarti data ini merupakan hasil dari proses pembersihan sebelumnya. Ini adalah data yang sudah diproses dengan metode tertentu untuk mengatasi nilai-nilai yang hilang.

- **CleanedData** mengacu pada variabel yang sudah dibersihkan dari nilai hilang. Biasanya, dataset ini sudah mengalami pengisian nilai hilang berdasarkan metode yang diterapkan dalam langkah sebelumnya (misalnya, konstan atau interpolasi).

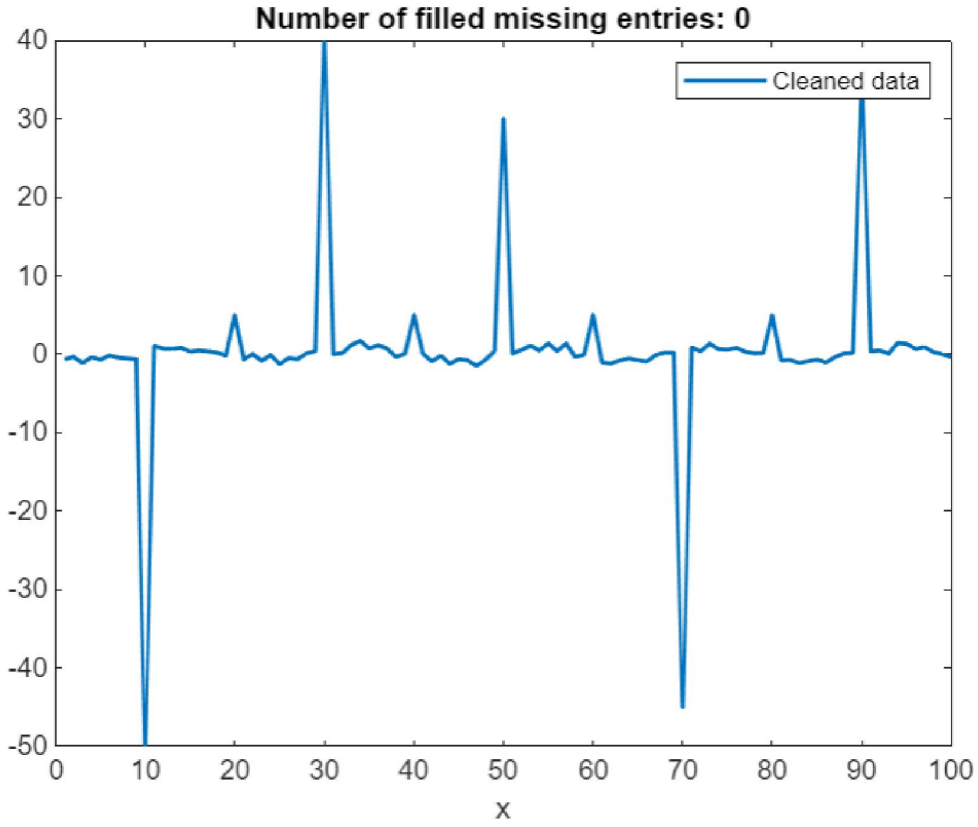
Jadi, ketika "cleanedData" dipilih sebagai input, MATLAB akan melakukan operasi pembersihan lebih lanjut pada dataset yang sudah bersih sebelumnya. Dengan kata lain, pengguna mungkin ingin menerapkan pembersihan data lagi atau metode pengisian lain pada data tersebut.

Previous Value:

2. **Previous Value**:
    - Pada bagian *Specify method*, metode pembersihan yang dipilih adalah **Fill missing** dengan pilihan metode pengisian **Previous value**.
    - **Previous value** adalah salah satu metode untuk mengisi data yang hilang dengan menggunakan nilai sebelumnya yang ada di dataset. Artinya, jika MATLAB menemukan nilai yang hilang di dataset, nilai tersebut akan digantikan dengan nilai yang ada tepat sebelum titik data yang hilang tersebut.

Contoh:

- Misalkan dataset berisi data: [1, 2, NaN, 4]
- Dengan menggunakan metode **Previous value**, nilai yang hilang (NaN) akan diisi dengan nilai sebelumnya, yaitu 2, sehingga hasilnya menjadi: [1, 2, 2, 4].



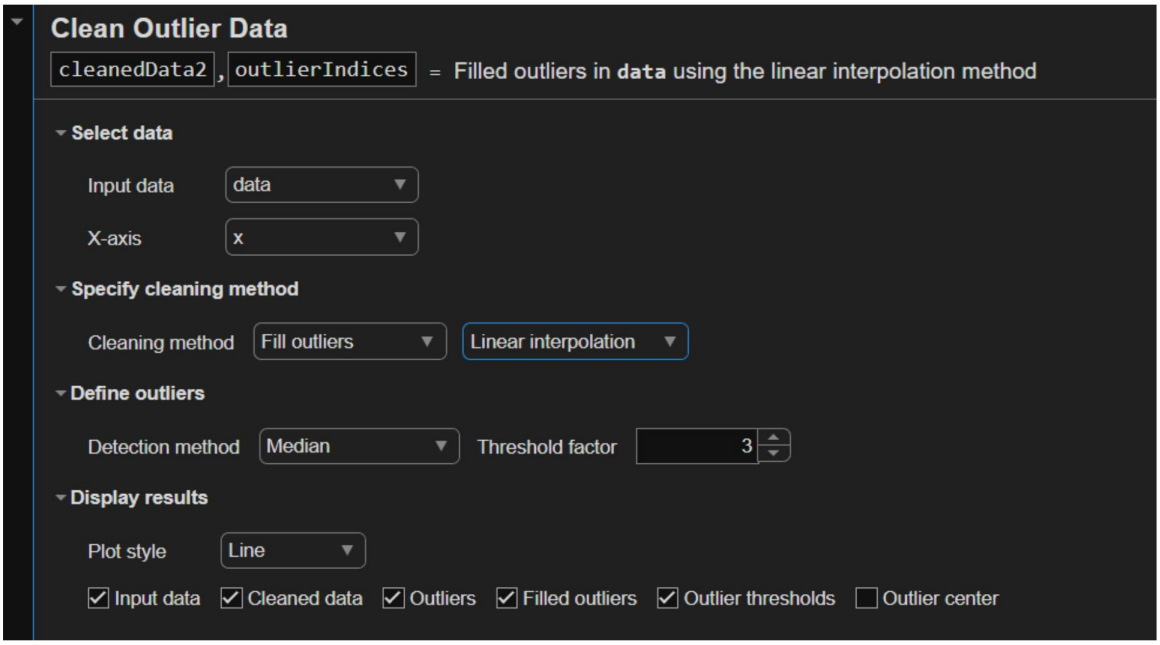
Gambar 20. Hasil Clean missing data Previous Value

Grafik ini menunjukkan data yang telah dibersihkan setelah menggunakan metode pembersihan data pada dataset. Berikut adalah beberapa poin penting mengenai grafik ini:

1. **Judul**:
    - Judul grafik menyatakan **"Number of filled missing entries: 0"**, yang berarti tidak ada data yang hilang yang diisi dalam proses pembersihan ini. Artinya, mungkin dataset sudah lengkap dan tidak ada nilai yang hilang untuk diisi, atau nilai yang hilang tidak terdeteksi.
2. **Sumbu X**:
    - Sumbu X (horizontal) menampilkan nilai dari variabel x, yang mungkin mewakili urutan atau posisi data dalam dataset. Nilai x berada dalam rentang 0 hingga 100.
3. **Sumbu Y**:
    - Sumbu Y (vertikal) menunjukkan nilai dari data yang dibersihkan. Nilai pada sumbu ini bervariasi dari sekitar -50 hingga 40, menunjukkan bahwa dataset ini memiliki variasi nilai yang cukup signifikan, termasuk beberapa *spike* (lonjakan) dan *drop* (penurunan drastis) pada titik-titik tertentu.
4. **Garis Biru – Cleaned Data**:
    - Garis biru dalam grafik ini mewakili **cleaned data**, yaitu data yang sudah melewati proses pembersihan. Karena tidak ada nilai hilang yang diisi (ditunjukkan oleh jumlah 0 untuk *filled missing entries*), ini kemungkinan besar merupakan visualisasi dari data asli atau data yang sudah bersih tanpa nilai yang hilang.
    - Beberapa titik dalam grafik menunjukkan lonjakan besar atau penurunan tajam pada data, tetapi karena tidak ada entri yang hilang, ini adalah karakteristik asli dari dataset, bukan hasil dari proses pembersihan.

3. FILL OUTLIERS

Instruksi ini menjelaskan bagaimana pengguna MATLAB dapat mendeteksi dan menghapus *outliers* dari dataset yang sudah dibersihkan. Selain itu, pengguna juga diberikan opsi untuk melihat kode yang dihasilkan secara otomatis oleh MATLAB, sehingga mereka dapat memahami dan memodifikasi kode sesuai kebutuhan.



Gambar 21. Clean Outliers Data Linear Interpolation

A. **cleanedData2, outlierIndices = ...**:

- Dua variabel yang dihasilkan dari tugas ini:
    - **cleanedData2**: Dataset yang sudah dibersihkan dengan *outliers* yang telah diisi atau ditangani.
    - **outlierIndices**: Indeks atau lokasi dari *outliers* yang terdeteksi di dalam dataset asli.

B. **Select data (Pemilihan data)**:

- **Input data**: Dataset yang akan diproses disebut data.
- **X-axis**: Variabel x digunakan sebagai sumbu X.

C. **Specify cleaning method (Metode pembersihan)**:

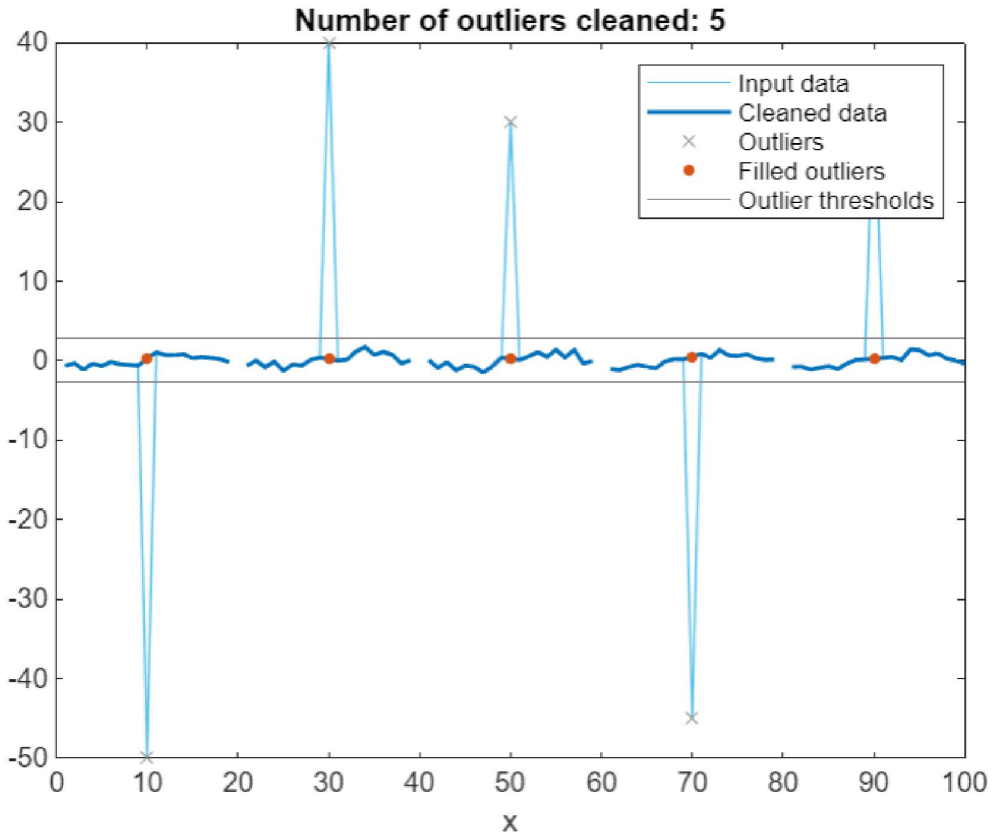
- **Cleaning method**: Metode yang dipilih untuk menangani *outliers* adalah **Fill outliers** (mengisi pencilan) menggunakan **Linear interpolation**. Ini berarti nilai pencilan akan diisi dengan nilai yang dihitung menggunakan interpolasi linier, yakni nilai yang terletak di antara dua titik yang valid di sekitarnya.

D. **Define outliers (Mendefinisikan pencilan)**:

- **Detection method**: *Outliers* dideteksi menggunakan **Median**. Artinya, pencilan dideteksi berdasarkan median data.
- **Threshold factor**: Faktor ambang batas yang digunakan untuk mendeteksi *outliers* diatur ke **3**. Ini berarti data yang lebih dari 3 kali jarak interkuartil dari median dianggap sebagai *outliers*.

E. **Display results (Menampilkan hasil)**:

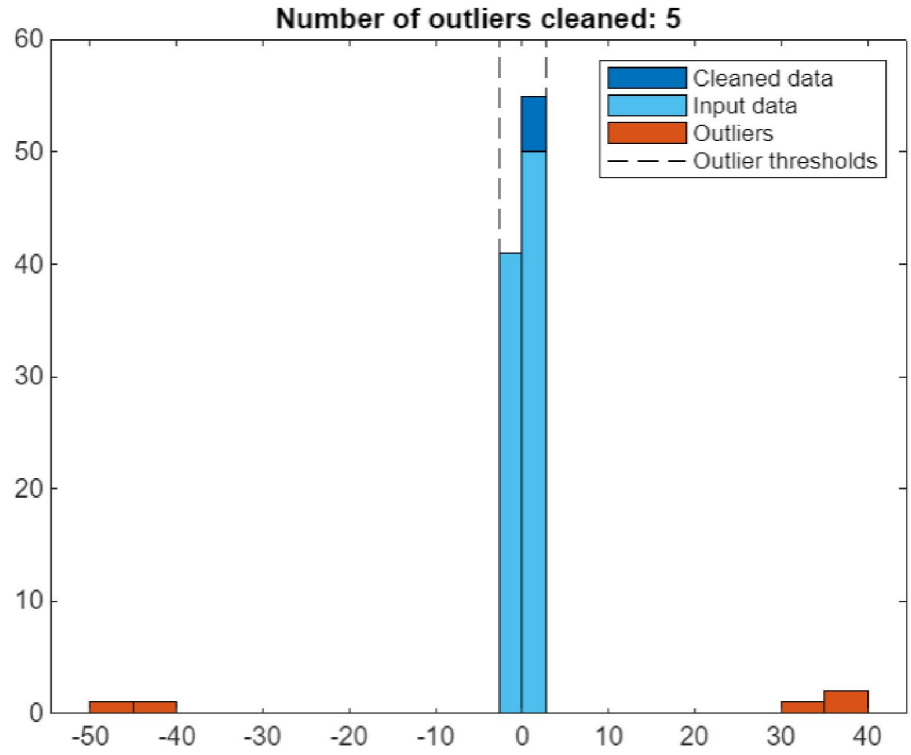
- **Plot style**: Grafik yang dihasilkan akan ditampilkan dalam bentuk **Line** (garis).
- **Input data, Cleaned data, Outliers, Filled outliers, Outlier thresholds**: Semua opsi ini dicentang, yang berarti grafik akan menampilkan data asli (input), data yang sudah dibersihkan, *outliers* yang terdeteksi, *outliers* yang telah diisi, dan ambang batas pencilan.



Gambar 22. Hasil Clean Outliers Data Linear Interpolation

Grafik ini menggambarkan bahwa setelah proses deteksi outliers, ada 5 outliers yang ditemukan di berbagai posisi sepanjang sumbu X (10, 30, 50, 70, 90). Nilai-nilai ini diisi atau diinterpolasi untuk menghasilkan data yang lebih halus, tanpa adanya lonjakan atau penurunan tajam. Outliers asli ditandai dengan tanda silang abu-abu, sementara hasil isian outliers ditandai dengan titik merah pada garis data yang sudah dibersihkan.

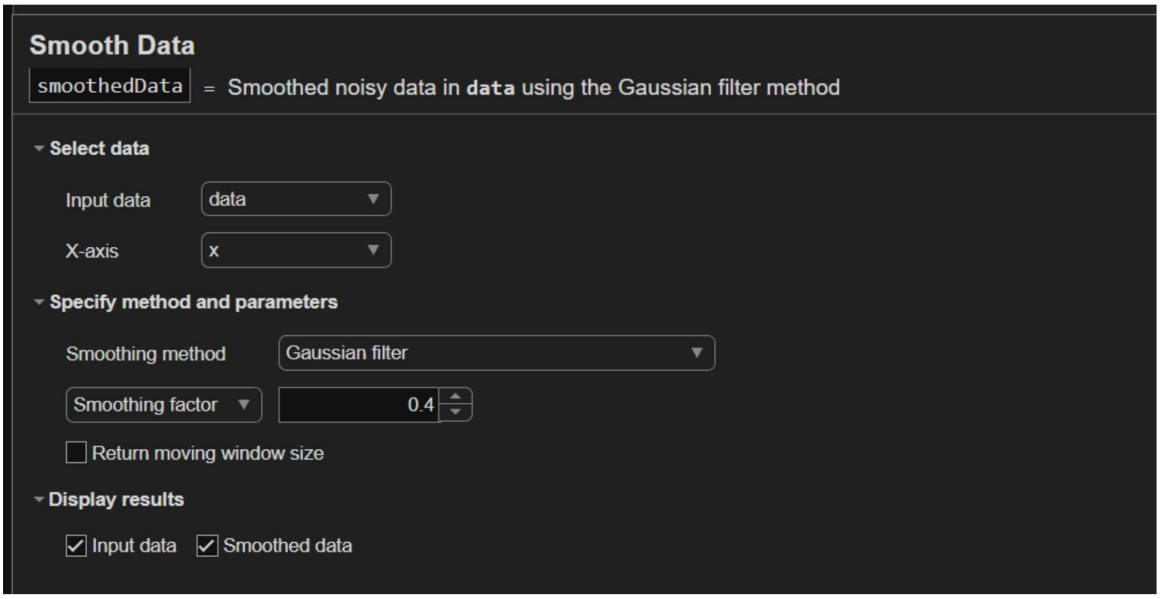
Berikut ini adalah tampilan dalam bentuk histogram:



Gambar 23. Clean Outliers Data Linear Interpolation

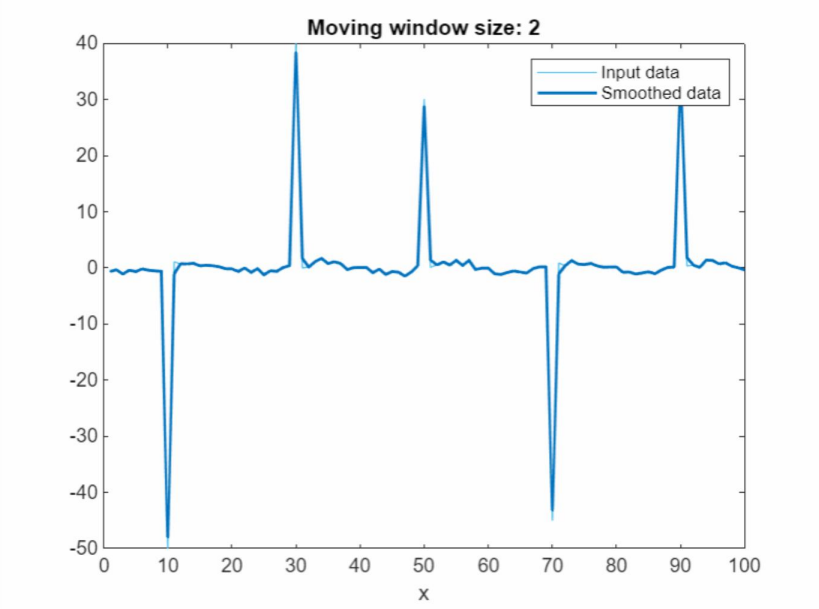
4. SMOOTH DATA

Instruksi ini menjelaskan cara melanjutkan proses pembersihan data dengan menghaluskan data yang sudah dibersihkan dari *outliers*. Proses ini menggunakan tugas **Smooth Data**, dan pengguna dapat menyesuaikan faktor penghalusan untuk mendapatkan hasil yang optimal.



Gambar 24. Smooth Data Gaussian filter

1. **Select data (Pemilihan data)**:
   - **Input data**: Input yang dipilih adalah variabel **data**. Ini adalah dataset yang akan dihaluskan.
   - **X-axis**: Variabel **x** dipilih sebagai sumbu X, yang menentukan bagaimana data diatur berdasarkan sumbu horizontal.
2. **Specify method and parameters (Menentukan metode dan parameter)**:
   - **Smoothing method**: Metode penghalusan yang digunakan adalah **Gaussian filter**. Filter ini menghaluskan data dengan cara menerapkan fungsi Gaussian, yang sering digunakan untuk meredam noise sambil tetap mempertahankan tren utama dalam data.
   - **Smoothing factor**: Faktor penghalusan yang dipilih adalah **0.4**. Nilai ini menentukan seberapa banyak penghalusan yang diterapkan pada data. Nilai yang lebih kecil cenderung mempertahankan detail data asli, sementara nilai yang lebih besar akan lebih banyak menghaluskan data.
   - **Return moving window size**: Opsi ini tidak dicentang, yang berarti ukuran jendela penghalusan tidak akan dikembalikan atau ditampilkan.
3. **Display results (Menampilkan hasil)**:
   - **Input data**: Dicentang, yang berarti data asli (input) akan ditampilkan di hasil visualisasi.
   - **Smoothed data**: Dicentang, yang berarti data yang telah dihaluskan juga akan ditampilkan bersama data asli untuk perbandingan.



Gambar 25. Hasil Smooth Data Gaussian filter

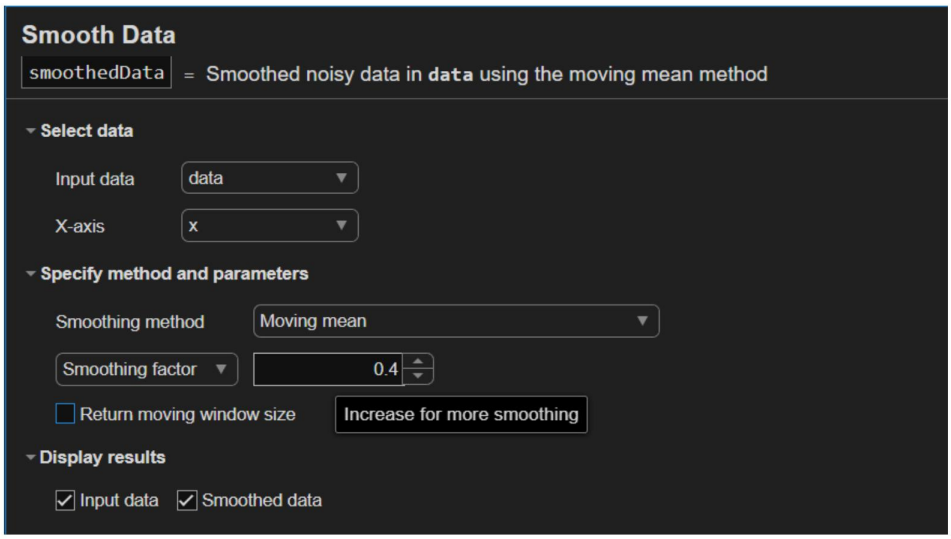
- **Moving window size: 2**:
    - Ini menunjukkan bahwa proses penghalusan dilakukan dengan ukuran jendela penghalusan sebesar 2, yang berarti data dihaluskan dengan mempertimbangkan dua titik tetangga terdekat di sekitar setiap titik data.

Elemen pada Grafik:

1. **Input data (Garis Biru Terang)**:
    - Garis biru terang menunjukkan data asli sebelum proses *smoothing*. Garis ini memperlihatkan beberapa lonjakan besar (spike) dan penurunan tajam yang menandakan adanya noise atau fluktuasi yang cukup signifikan dalam data.
2. **Smoothed data (Garis Biru Gelap)**:
    - Garis biru gelap menunjukkan data yang telah dihaluskan menggunakan metode *Gaussian filter*. Garis ini lebih halus dibandingkan data asli, terutama di sekitar area yang memiliki lonjakan atau penurunan tajam. Penghalusan ini mengurangi fluktuasi kecil dalam data tanpa menghilangkan tren utama.

Analisis:

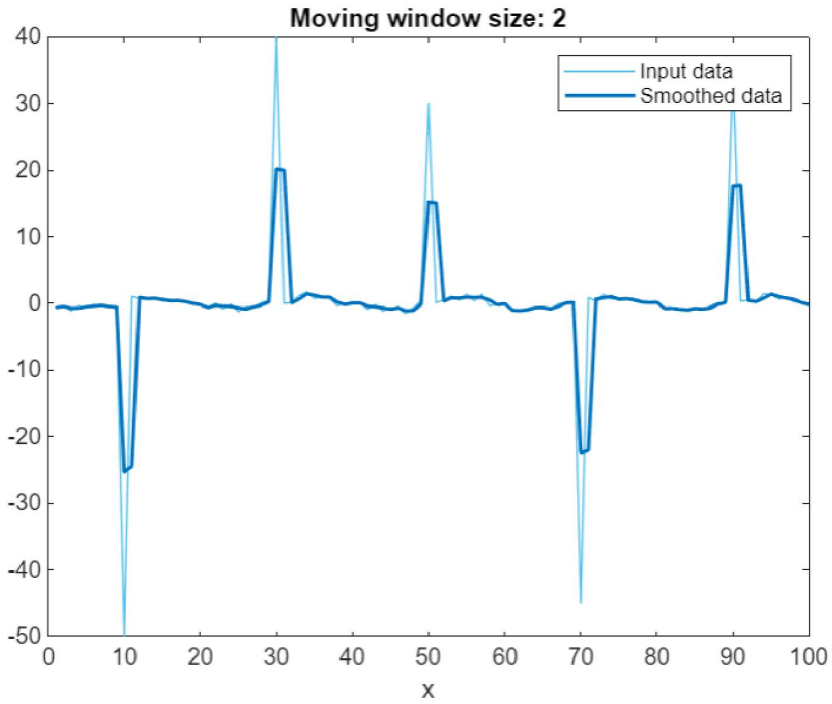
- **Smoothing**: Proses penghalusan berhasil meredam beberapa fluktuasi tajam dalam data asli. Meskipun lonjakan dan penurunan tajam masih terlihat (terutama di sekitar titik 10, 30, 50, 70, dan 90 pada sumbu X), penghalusan berhasil membuat data menjadi lebih mulus. Hal ini menunjukkan bahwa ukuran jendela penghalusan yang kecil (2) menjaga beberapa detail penting dari data asli.
- **Perbandingan Input dan Smoothed Data**: Anda dapat melihat bahwa data yang dihaluskan (garis biru gelap) tetap mengikuti pola umum dari data asli, namun variasi kecil atau fluktuasi yang cepat telah direduksi. Ini membantu mempertahankan informasi utama dari data asli sembari mengurangi efek noise atau fluktuasi yang tidak diinginkan.



Gambar 26. Smooth Data Moving mean

Gambar di atas menunjukkan metode penghalusan (*smoothing method*) yang dipilih adalah **Moving mean. Moving mean** adalah metode penghalusan data yang bekerja dengan menghitung rata-rata dari sejumlah titik data yang berdekatan dalam suatu jendela bergerak (moving window). Setiap kali jendela bergerak melintasi dataset, rata-rata dihitung ulang untuk mencakup titik-titik baru, dan hasilnya digunakan untuk menggantikan nilai asli.

Hasil grafiknya sebagai berikut:



Gambar 27. Hasil Smooth Data Moving Mean

Grafik ini menunjukkan perbandingan antara **data asli** dan **data yang telah dihaluskan** menggunakan metode *Moving Mean* dengan ukuran jendela penghalusan sebesar 2, seperti yang terlihat dari judul grafik **"Moving window size: 2"**. Berikut adalah penjelasan singkat dari grafik ini:

Elemen dalam Grafik:

1. **Input data (Garis Biru Terang)**:
    - Garis biru terang menggambarkan **data asli** yang memiliki lonjakan (spikes) dan penurunan tajam di beberapa titik. Lonjakan ini menunjukkan adanya noise atau variasi besar pada dataset.
2. **Smoothed data (Garis Biru Gelap)**:
    - Garis biru gelap menunjukkan **data yang telah dihaluskan**. Setelah proses *Moving Mean*, data menjadi lebih mulus, terutama di area yang sebelumnya mengalami lonjakan tajam.
    - Proses penghalusan ini meredam lonjakan dan fluktuasi kecil, menghasilkan data yang lebih stabil dan mudah dianalisis.

Analisis:

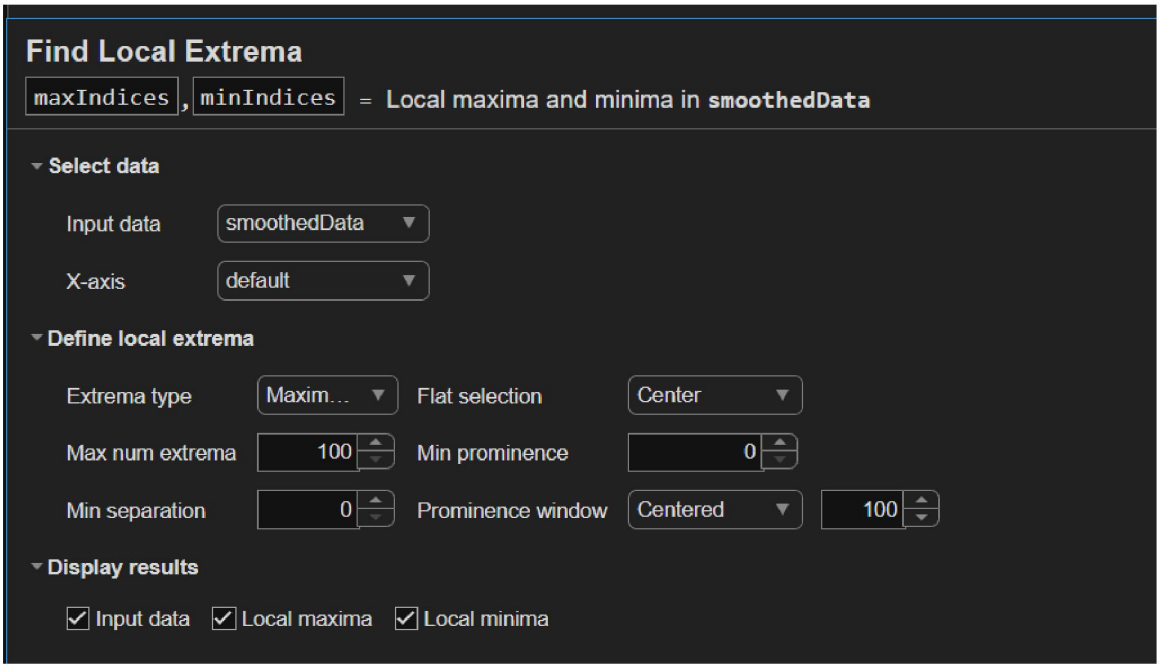
- Dengan **Moving window size = 2**, penghalusan dilakukan dengan rata-rata dari dua titik data di sekitar setiap titik. Hal ini menyebabkan lonjakan tajam di data asli sedikit berkurang pada data yang telah dihaluskan.
- Meskipun lonjakan tidak sepenuhnya hilang, data yang dihaluskan lebih halus dan lebih mudah untuk diinterpretasikan, karena metode ini meredam noise tanpa menghilangkan pola utama dari data.

Kesimpulan:

Grafik ini menunjukkan bahwa metode *Moving Mean* berhasil menghaluskan data asli, mengurangi variasi kecil dan lonjakan tajam, sehingga data menjadi lebih halus dan lebih mudah dianalisis, terutama untuk tujuan visualisasi atau deteksi tren dalam dataset.

5. LOCAL EXTREMA

Instruksi ini memberikan panduan untuk menggunakan tugas **Find Local Extrema** di MATLAB. Anda bisa menggunakannya untuk menemukan puncak (maxima) dan lembah (minima) dari data yang sudah dihaluskan. Parameter untuk deteksi *extrema* juga dapat disesuaikan untuk menangkap lebih banyak atau lebih sedikit titik penting di dalam dataset.



Gambar 28. Find Local Extrema

1. maxIndices, minIndices = Local maxima and minima in smoothedData:

- **maxIndices**: Indeks dari puncak lokal (local maxima) yang ditemukan dalam data.
- **minIndices**: Indeks dari lembah lokal (local minima) yang ditemukan dalam data.
- Hasil tugas ini adalah daftar indeks puncak dan lembah dalam variabel smoothedData (data yang telah dihaluskan).

2. Select Data (Pemilihan Data):

- **Input data**: Data input yang dipilih adalah **smoothedData**, yang merupakan data hasil penghalusan sebelumnya.
- **X-axis**: Pilihan default untuk sumbu X digunakan, yang berarti data akan diplot menggunakan urutan default tanpa pengubahan sumbu X.

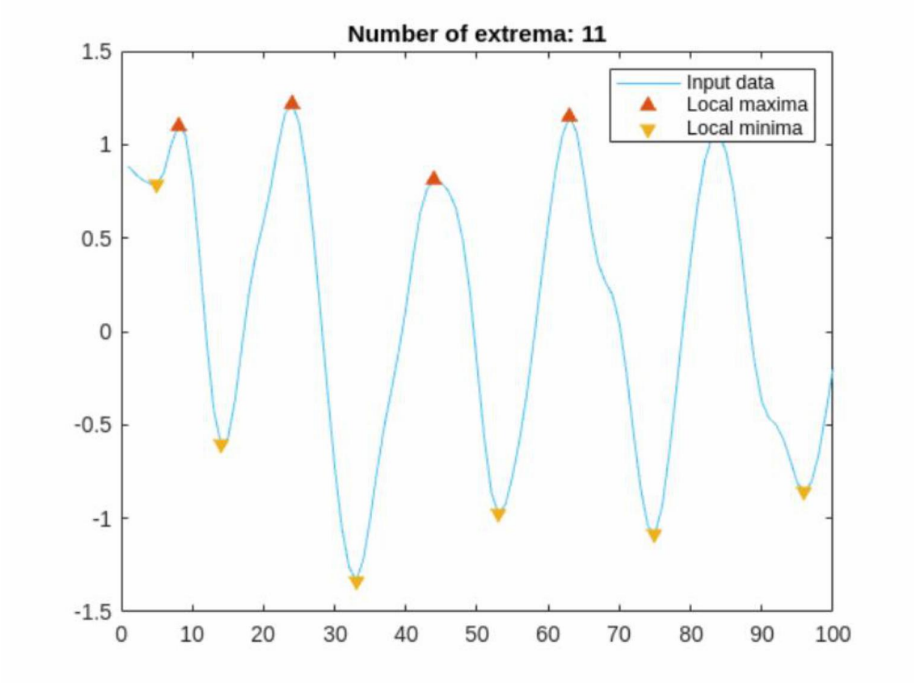
3. Define Local Extrema (Mendefinisikan Ekstrema Lokal):

- **Extrema type**: Tipe yang dipilih adalah **Maxima...**, yang berarti tugas ini akan mencari *local maxima* (puncak lokal). Jika opsi ini diklik, pengguna dapat beralih untuk mencari *minima* atau *both* (puncak dan lembah).
- **Flat selection**: Opsi **Center** dipilih, yang mungkin mengacu pada bagaimana puncak/lembah datar ditangani, memilih pusat datar untuk dianggap sebagai ekstrema.
- **Max num extrema**: Diatur ke **100**, artinya maksimal 100 puncak/lembah lokal akan ditemukan.

- **Min prominence**: Diatur ke **0**, yang berarti tidak ada batas minimum untuk seberapa menonjol puncak/lembah tersebut. Ekstrema dengan tingkat penonjolan yang sangat kecil tetap akan terdeteksi.
- **Min separation**: Diatur ke **0**, yang berarti tidak ada jarak minimum yang diperlukan antara dua ekstrema. Ekstrema yang sangat dekat satu sama lain masih akan terdeteksi.
- **Prominence window**: Opsi **Centered** dengan nilai **100** dipilih, yang mengatur bagaimana rentang jendela dihitung untuk mengidentifikasi ekstrema yang menonjol.

4. Display Results (Menampilkan Hasil):

- **Input data**: Dicentang, artinya data asli (input) akan ditampilkan dalam grafik.
- **Local maxima**: Dicentang, artinya puncak lokal yang terdeteksi akan ditampilkan dalam grafik.
- **Local minima**: Dicentang, artinya lembah lokal yang terdeteksi juga akan ditampilkan dalam grafik.

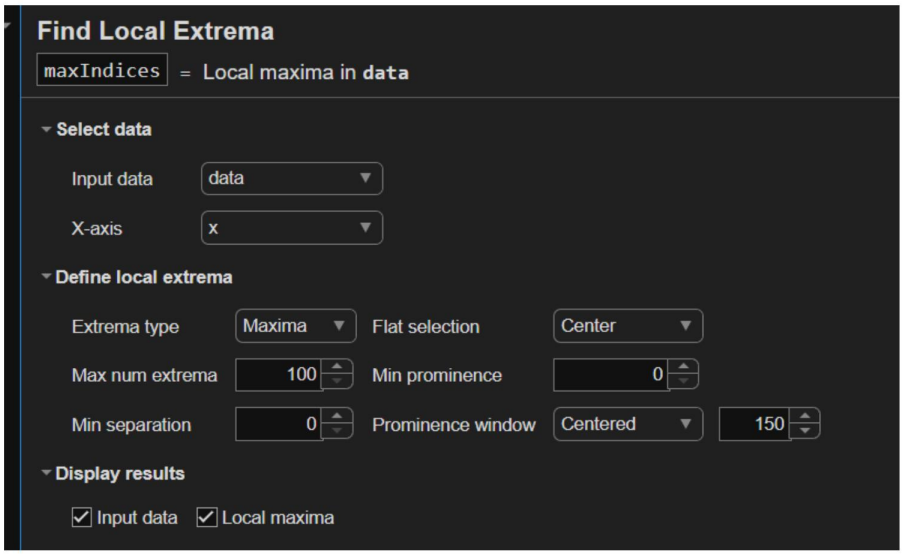


Gambar 29. Hasil Find Local Extrema

- Data ini memiliki pola yang berosilasi secara reguler, dengan puncak dan lembah yang jelas terlihat. Deteksi *local extrema* membantu mengidentifikasi titik-titik tertinggi (puncak) dan terendah (lembah) pada dataset ini.
- Total ada 11 titik ekstrim yang ditemukan, yang terdiri dari puncak dan lembah, sesuai dengan perubahan yang konsisten pada pola data.

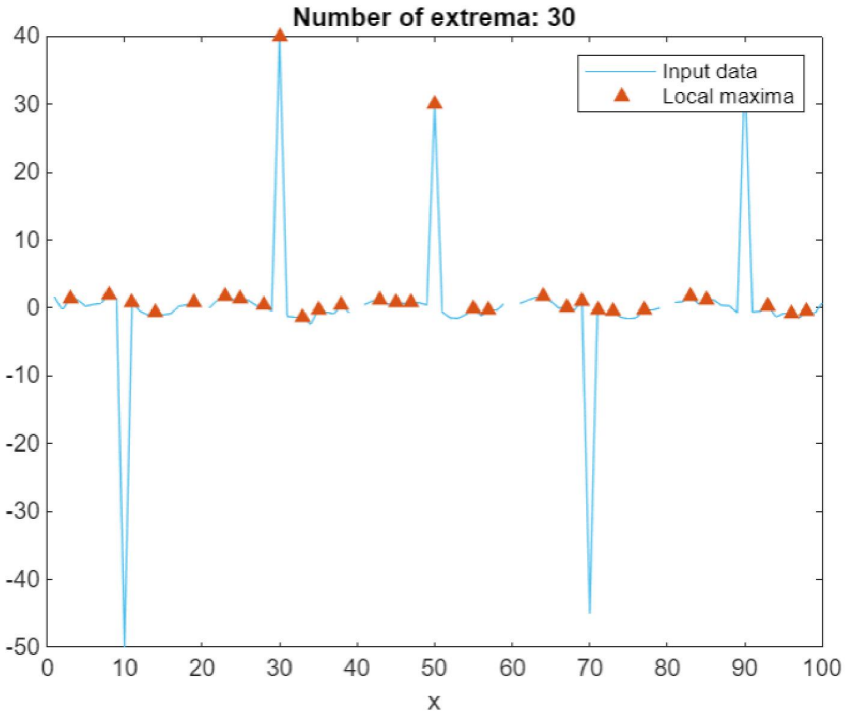
Grafik ini menunjukkan hasil deteksi dari puncak lokal dan lembah lokal dalam data. Metode deteksi ini membantu memvisualisasikan titik-titik penting dalam data yang mewakili perubahan ekstrem, baik naik (puncak) maupun turun (lembah). Hasil ini sangat

berguna dalam analisis data, terutama ketika Anda ingin memahami tren utama atau fluktuasi dalam data.



Gambar 30. Find Local Extrema

Ketika Input data di setting “data” dan x axis di setting “x” maka hasil grafiknya sebagai berikut:



Gambar 31. Hasil Find Local Extrema

- **Lonjakan Data**: Terdapat beberapa lonjakan yang cukup signifikan dalam dataset, yang ditandai dengan puncak yang terdeteksi di sekitar titik-titik seperti 10, 30, 50, dan 90. Lonjakan ini diidentifikasi sebagai puncak lokal utama (local maxima).
- **Puncak Kecil**: Selain puncak besar, terdapat juga puncak kecil yang terdeteksi di antara fluktuasi data yang lebih kecil. Hal ini menunjukkan bahwa data memiliki variasi yang cukup tinggi, dan puncak-puncak ini mungkin dianggap sebagai noise atau variasi kecil.

Grafik ini menunjukkan bahwa terdapat 30 **local maxima** (puncak lokal) dalam dataset. Sebagian besar puncak yang terdeteksi berhubungan dengan lonjakan yang signifikan dalam data, tetapi beberapa juga menunjukkan fluktuasi kecil yang dianggap sebagai puncak lokal. Visualisasi ini memberikan wawasan tentang bagaimana data berfluktuasi, dengan informasi yang detail mengenai titik-titik tertinggi dalam dataset.