# BAB I PENDAHULUAN

## Tujuan

1. Dapat menentukan penyelesaian model regresi secara Numerik.
2. Mencari besarnya kesalahan dari suatu perhitungan solusi model regresi secara numerik.

## Rumusan Permasalahan

Cocokkan polinomial orde kedua ke data berikut



Persamaan model polinomial orde dua

Misalkan **a**

Tentukan solusi model regresi polinomial orde dua tersebut!

# BAB II DASAR TEORI

## 2.1 Pendahuluan Teori

Dalam penerapan model regresi, satu tujuan adalah untuk mendapatkan persamaan y=f(x) yang paling menggambarkan n titik-titik data respon (x1 ,y1), (x2 ,y2), ....., (xn , yn). Akibatnya, kita dihadapkan dengan jawaban dua pertanyaan dasar.

1. Apakah model y = f (x) menggambarkan data secara memadai, yaitu apakah ada kecocokan yang memadai?

2. Seberapa baik model memprediksi variabel respons (prediktabilitas)?

Lihatlah dua persamaan yang diberikan di bawah ini.

dimana Sr adalah jumlahan kuadrat dari residu (residu adalah perbedaan antara nilai observasi, yi dan nilai prediksi, ), dan St adalah jumlahan kuadrat dari perbedaan antara nilai obervasi dan nilai rata-rata.

## 2.2 Metode Regresi

Diberikan N titik data, , kemudian mencari persamaan polinomial, sebagai

Atau dapat ditulis sebagai

Algoritma Regresi :

1. Menghitung sum(x), sum(, sum(, sum(
2. Membuat matrix
3. Menghitung sum(y), sum(xy), sum(
4. Membuat matrix
5. Invers matrix A
6. Kalikan invers matrix A dengan matrix B
7. Hasilnya

## 2.3 Metode Eliminasi Gauss

Diberikan N titik data, , kemudian mencari persamaan polinomial, sebagai

Atau dapat ditulis sebagai

Algoritma Eliminasi Gauss:

1. Menghitung sum(x), sum(, sum(, sum(, sum(y), sum(xy), sum(
2. Membuat matrix
3. Untuk m=0 sampai n:
   1. Untuk r=m sampai n

3.1.1 lead = A[r][m]

3.1.2 untuk k=0 sampai n+1, A[r][k] = A[r][k]/lead

* 1. Untuk r=n-1 sampai 0:

3.2.1 untuk k=0 sampai n+1, A[r][k] = A[r][k] - A[r-1][k]

1. Untuk i=n-1 sampai 0:
   1. x[i] = A[i][n]/A[i][i]
   2. untuk k=i-1 sampai 0, A[k][n] = A[k][n] - A[k][i]\*x[i]
2. Kembalikan nilai x

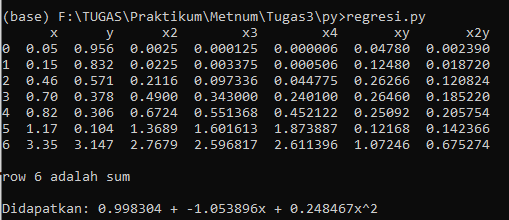
# BAB III PEMBAHASAN

## 3.1 Penyelesaian dengan Metode Regresi

**3.1.1 Source Code**

|  |
| --- |
| # Metode Regresi  # 24060119130045 - Nashirudin Baqiy  import numpy as np  import pandas as pd  #data pada tabel  x = [0.05, 0.15, 0.46, 0.70, 0.82, 1.17]  y = [0.956, 0.832, 0.571, 0.378, 0.306, 0.104]  def x2(x):  return x\*x  def x3(x):  return x\*x\*x  def x4(x):  return x\*x\*x\*x  def regresi(x, y):  #Panjang/jumlah data point  n = len(x)  #Inisiasi tabel  tbl = [[None for x in range(7)] for x in range(n+1)]  #Sum x  sx = 0  for i in range(n):  sx = sx + x[i]  tbl[i][0] = x[i]  tbl[n][0] = sx  #Sum y  sy = 0  for i in range(n):  sy = sy + y[i]  tbl[i][1] = y[i]  tbl[n][1] = sy  #Sum x2  sx2 = 0  for i in range(n):  sx2 = sx2 + x2(x[i])  tbl[i][2] = x2(x[i])  tbl[n][2] = sx2  #Sum x3  sx3 = 0  for i in range(n):  sx3 = sx3 + x3(x[i])  tbl[i][3] = x3(x[i])  tbl[n][3] = sx3  #Sum x4  sx4 = 0  for i in range(n):  sx4 = sx4 + x4(x[i])  tbl[i][4] = x4(x[i])  tbl[n][4] = sx4  #Sum x.y  sxy = 0  for i in range(n):  sxy = sxy + x[i]\*y[i]  tbl[i][5] = x[i]\*y[i]  tbl[n][5] = sxy  #Sum x2y  sx2y = 0  for i in range(n):  sx2y = sx2y + x2(x[i])\*y[i]  tbl[i][6] = x2(x[i])\*y[i]  tbl[n][6] = sx2y  tbl\_table = pd.DataFrame(tbl, columns=['x','y','x2','x3','x4','xy','x2y'])  print(tbl\_table)  print()  print('row 6 adalah sum')  a = np.linalg.inv([ [n, sx, sx2], [sx, sx2, sx3], [sx2, sx3, sx4] ])  b = [sy, sxy, sx2y]  hasil = np.matmul(a, b)  print()  print('Didapatkan: %f + %fx + %fx^2' % (hasil[0], hasil[1], hasil[2]))  regresi(x,y) |

**3.1.2 Hasil Command Prompt**



**3.1.3 Excel**



**3.1.4 Penjelasan**

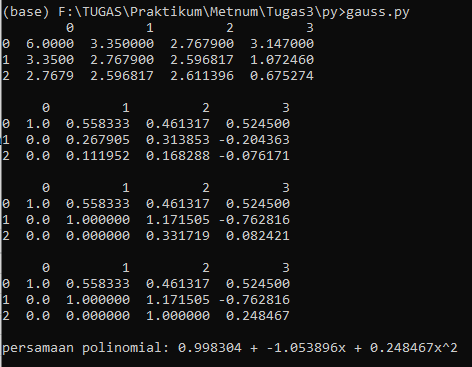
Program untuk metode regresi numerik di atas baik menggunakan python maupun excel sama-sama melakukan algoritma yang sama seperti pada dasar teori. Untuk mencari menggunakan matriks A mencari terlebih dahulu Sum-sum yang diperlukan yaitu didapatkan nilainya 3.35, 2.7679, 2.597, 2.611. Kemudian mencari nilai matriks B menghitung sum-sum untuk didapatkan 3.147, 1.072, 0.675. Kemudian invers matriks A dan kalikan dengan matriks B. Sehingga didapatkan nilai matriks untuk .

## 3.2 Penyelesaian dengan Metode Eliminasi Gauss

**3.1.1 Source Code**

|  |
| --- |
| # Metode Regresi  # 24060119130045 - Nashirudin Baqiy  import numpy as np  import pandas as pd  #data pada tabel  x = [0.05, 0.15, 0.46, 0.70, 0.82, 1.17]  y = [0.956, 0.832, 0.571, 0.378, 0.306, 0.104]  n = len(x)  def xn(x,n):  s = x  for i in range(n-1):  s = s\*x  return s  def mat(x, y):  #Panjang/jumlah data point  n = len(x)  #Inisiasi tabel  A = [[None for x in range(4)] for x in range(3)]  A[0][0] = n  #Sum x  sx = 0  for i in range(n):  sx = sx + x[i]  A[1][0] = sx  A[0][1] = sx  #Sum x2  sx2 = 0  for i in range(n):  sx2 = sx2 + xn(x[i],2)  A[2][0] = sx2  A[1][1] = sx2  A[0][2] = sx2  #Sum x3  sx3 = 0  for i in range(n):  sx3 = sx3 + xn(x[i],3)  A[2][1] = sx3  A[1][2] = sx3  #Sum x4  sx4 = 0  for i in range(n):  sx4 = sx4 + xn(x[i],4)  A[2][2] = sx4  #Sum y  sy = 0  for i in range(n):  sy = sy + y[i]  A[0][3] = sy  #Sum x.y  sxy = 0  for i in range(n):  sxy = sxy + x[i]\*y[i]  A[1][3] = sxy  #Sum x2y  sx2y = 0  for i in range(n):  sx2y = sx2y + xn(x[i],2)\*y[i]  A[2][3] = sx2y    return A  def gauss(A):  n = len(A)  tp = [[0 for i in range(n+1)] for i in range(n)]  for m in range(0, n):  for r in range(m, n):  lead = A[r][m]  for k in range(0, n+1):  A[r][k] = A[r][k]/lead  for r in range(n-1, -1, -1):  for k in range(0, n+1):  if r > m:  A[r][k] = A[r][k] - A[r-1][k]  G\_table = pd.DataFrame(A)  print(G\_table)  print()    x = [0 for i in range(n)]  for i in range(n - 1, -1, -1):  x[i] = A[i][n] / A[i][i]  for k in range(i - 1, -1, -1):  A[k][n] -= A[k][i] \* x[i]    return x  A = mat(x,y)  A\_table = pd.DataFrame(A)  print(A\_table)  print()  G = gauss(A)  print('persamaan polinomial: %f + %fx + %fx^2' % (G[0],G[1],G[2])) |

**3.1.2 Hasil Command Prompt**



**3.1.3 Excel**



**3.1.4 Penjelasan**

Program untuk metode eliminasi gauss di atas baik menggunakan python maupun excel sama-sama melakukan algoritma yang sama seperti pada dasar teori. Untuk mencari menggunakan matriks A mencari terlebih dahulu Sum-sum yang diperlukan yaitu didapatkan 3.35, 2.7679, 2.597, 2.611, 3.147, 1.072, 0.675. Kemudian perulangan (M) agar lead 1 diagonal yaitu membagi baris yang akan menjadi lead 1 dengan bilangan itu sendiri lalu dikurangi dari bawah antara baris bawah dengan baris atasnya tetapi ada syarat untuk pengurangan yaitu row>M. Terjadi perulangan sebanyak n = 3, sehingga dapat mencari untuk

# BAB IV PENUTUPAN

## Kesimpulan

Solusi model regresi polinomial dicari dengan Metode Regresi dan Metode Eliminasi Gauss. Langkah-langkah yang tersedia dapat dibuat dengan menggunakan Bahasa *Python* dan menggunakan Spreadsheet seperti excel sehingga tidak perlu melakukan iterasi secara sistem manual seperti yang ada pada *proses* di atas. Hasil kerja dari sistem program tersebut adalah sebagai berikut.

1. Pada soal tentang implementasi Metode Regresi, dicari solusi model regresi polinomial. Hasil yang diperoleh menggunakan Metode Regresi yang telah dibuat dalam Bahasa *Python* dan Excel adalah .
2. Pada soal tentang implementasi Metode Eliminasi Gauss, dicari solusi model regresi polinomial. Hasil yang diperoleh menggunakan Metode Regresi yang telah dibuat dalam Bahasa *Python* dan Excel adalah .