**UTS**

**Grafika Komputer**



Disusun oleh :

Figo Gymnastiar Farhaan Pratama (20051397015)

**PROGRAM STUDI MANAJEMEN INFORMATIKA**

**FAKULTAS VOKASI**

**UNIVERSITAS NEGERI SURABAYA**

**2021**

**Bab I: Pendahuluan**

Algoritma bresenham adalah algoritma membentuk garis digital dengan menentukan koordinat pixel awal (x0, y0) dan koordinat pixel akhir (x1, y1).

Cara yang dilakukan yaitu dengan melakukan penambahan atau pengurangan pada persamaan yang telah dibentuk, sehingga akan ditemukan titik koordinat pixel diantara koordinat awal dan akhir, lalu setiap titik koordinat tersebut akan ditarik garis sehingga koordinat awal (x0, y0) dan koordinat akhir (x1, y1) dapat terhubung.

**Bab II: Contoh Kasus**

Berikut adalah contoh kasus pembentukan garis menggunakan algoritma bresenham dengan gradien (0 ≤ m ≤ 1) . Seluruh program dibuat menggunakan bahasa python pada source code spyder.

* Langkah 1, Memasukkan nilai koordinat awal (x0, y0) dan koordinat akhir (x1, y1). Contoh programnya seperti berikut,

print("Masukkan nilai (x1, y1) dan (x2, y2)")

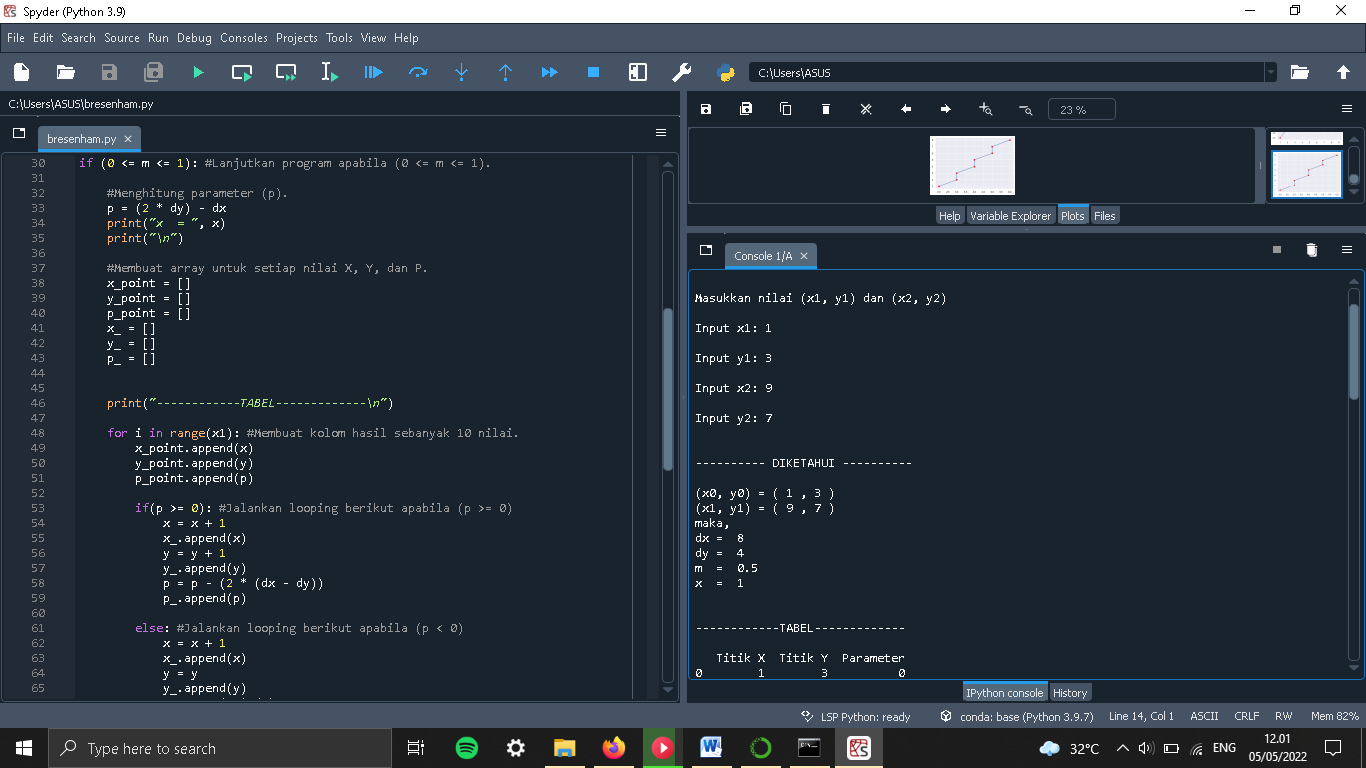
x0 = int(input("Input x1: "))

y0 = int(input("Input y1: "))

x1 = int(input("Input x2: "))

y1 = int(input("Input y2: "))

akan muncul output seperti berikut apabila input koordinat sudah dilakukan. Dalam contoh kasus ini koordinat awal (x0, y0) adalah (1, 3) dan koordinat akhir (x1, y1) adalah (9, 7)



* Langkah 2, mendeklarasikan variabel. Pada langkah ini, program akan menghitung nilai dx, dy, gradien (m) dan parameter (p) sebelum pembuatan tabel. Contoh programnya seperti berikut,

print("\n")

print("---------- DIKETAHUI ----------\n")

#Mendeklarasi (x0, y0) dan (x1, y1).

x = x0

y = y0

print("(x0, y0) = (", x0, ",", y0, ")")

print("(x1, y1) = (", x1, ",", y1, ")")

#Menghitung dX dan dY.

dx = x1 - x0

dy = y1 - y0

print("maka,")

print("dx = ", dx)

print("dy = ", dy)

#Menghitung gradien garis (m).

m = dy/dx

print("m = ", m)

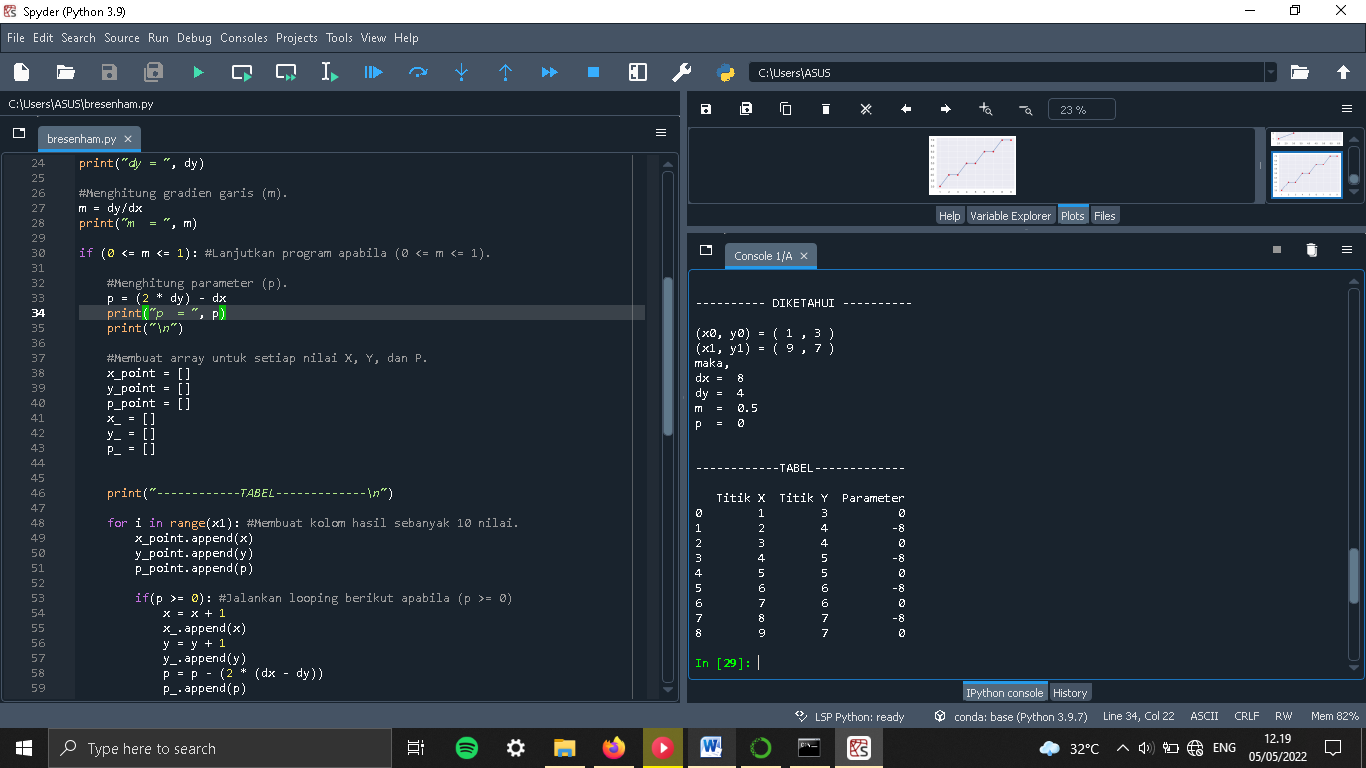
#Menghitung parameter (p).

p = (2 \* dy) - dx

print("p = ", p)

print("\n")

jika deklarasi sudah dilakukan, maka output akan terlihat seperti berikut,



* Langkah 3, membuat array untuk menyimpan hasil looping.

x\_point = []

y\_point = []

p\_point = []

x\_ = []

y\_ = []

p\_ = []

* Langkah 4, membuat tabel dan titiknya. Disini akan dilakukan looping terhadap titik x, y, dan parameter (p) dan looping akan berhenti saat x = x1.

for i in range(x1): #Membuat kolom hasil sampai titik X = X1

Lalu, apabila p >= 0, maka p akan dikurang oleh d2, sehingga

p = p – d2

p = p – ( 2 × (dx – dy) )

Sedangkan titik x dan y akan ditambah menjadi x + 1 dan y + 1. Seperti pada program berikut,

if(p >= 0): #Jalankan looping berikut apabila (p >= 0)

x = x + 1

x\_.append(x)

y = y + 1

y\_.append(y)

p = p - (2 \* (dx - dy))

p\_.append(p)

Namun apabila p < 0, maka p akan dijumlah oleh d1, sehingga

p = p – d2

p = p + ( 2 × dy)

Sedangkan hanya titik x akan ditambah menjadi x + 1 dan titik y tetap.

else: #Jalankan looping berikut apabila (p < 0)

x = x + 1

x\_.append(x)

y = y

y\_.append(y)

p = p + (2 \* dy)

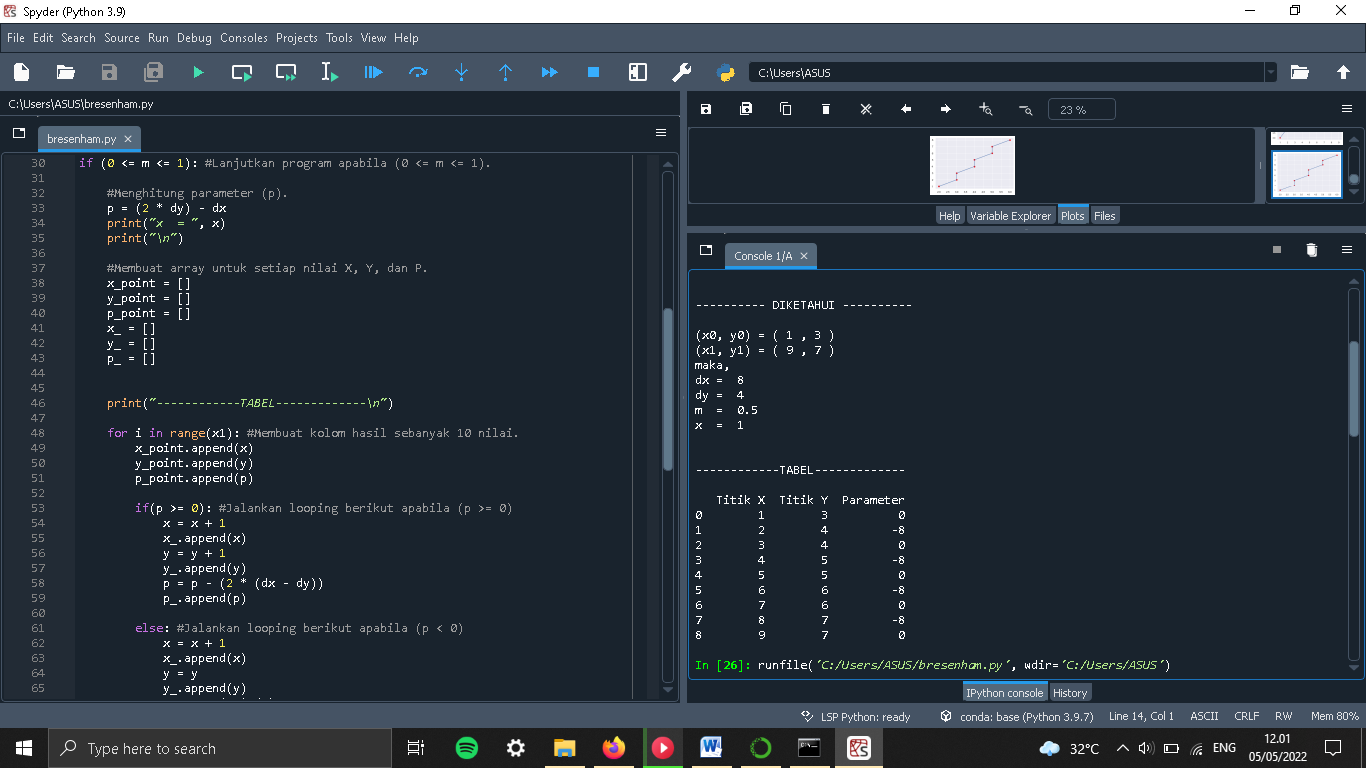
p\_.append(p)

selama looping dilakukan, hasil akan tersimpan pada array yang sudah dibuat pada langkah 3. Selanjutnya akan ditampilkan pada tabel dengan menggunakan program berikut,

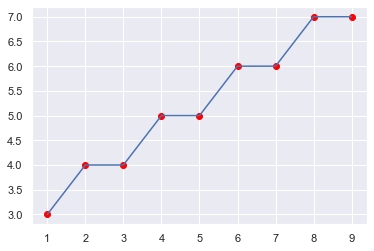
print(pd.DataFrame(

{"Titik X":x\_point,"Titik Y":y\_point,"Parameter":p\_point}))

sehingga output akan terlihat seperti berikut,



* Langkah 5, output garis berupa gambar. Setelah tabel dibuat dan seluruh koordinat sudah diketahui, maka akan ditarik garis berdasarkan koordinat yang sudah diketahui. Berikut adalah hasil dari perhitungan garis dengan titik koordinat awal (1, 3) dan koordinat titik akhir (9, 7) menggunakan algoritma bresenham.



Selanjutnya adalah contoh kasus pembentukan garis menggunakan algoritma bresenham dengan gradien (m > 1)

* Langkah 1, Memasukkan nilai koordinat awal (x0, y0) dan koordinat akhir (x1, y1). Contoh programnya seperti berikut,

print("Masukkan nilai (x1, y1) dan (x2, y2)")

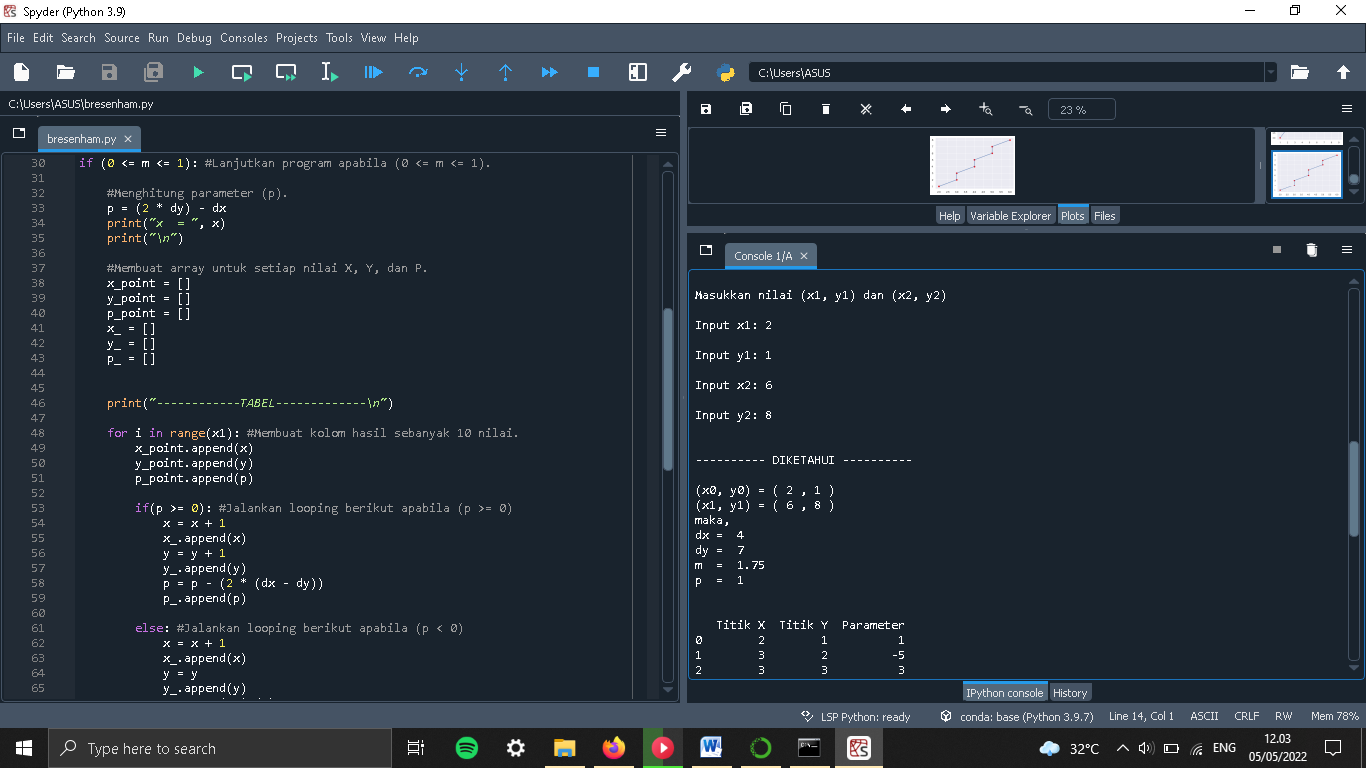
x0 = int(input("Input x1: "))

y0 = int(input("Input y1: "))

x1 = int(input("Input x2: "))

y1 = int(input("Input y2: "))

akan muncul output seperti berikut apabila input koordinat sudah dilakukan. Dalam contoh kasus ini koordinat awal (x0, y0) adalah (2, 1) dan koordinat akhir (x1, y1) adalah (6, 8)



* Langkah 2, mendeklarasikan variabel. Pada langkah ini, program akan menghitung nilai dx, dy, gradien (m) dan parameter (p) sebelum pembuatan tabel. Contoh programnya seperti berikut,

print("\n")

print("---------- DIKETAHUI ----------\n")

#Mendeklarasi (x0, y0) dan (x1, y1).

x = x0

y = y0

print("(x0, y0) = (", x0, ",", y0, ")")

print("(x1, y1) = (", x1, ",", y1, ")")

#Menghitung dX dan dY.

dx = x1 - x0

dy = y1 - y0

print("maka,")

print("dx = ", dx)

print("dy = ", dy)

#Menghitung gradien garis (m).

m = dy/dx

print("m = ", m)

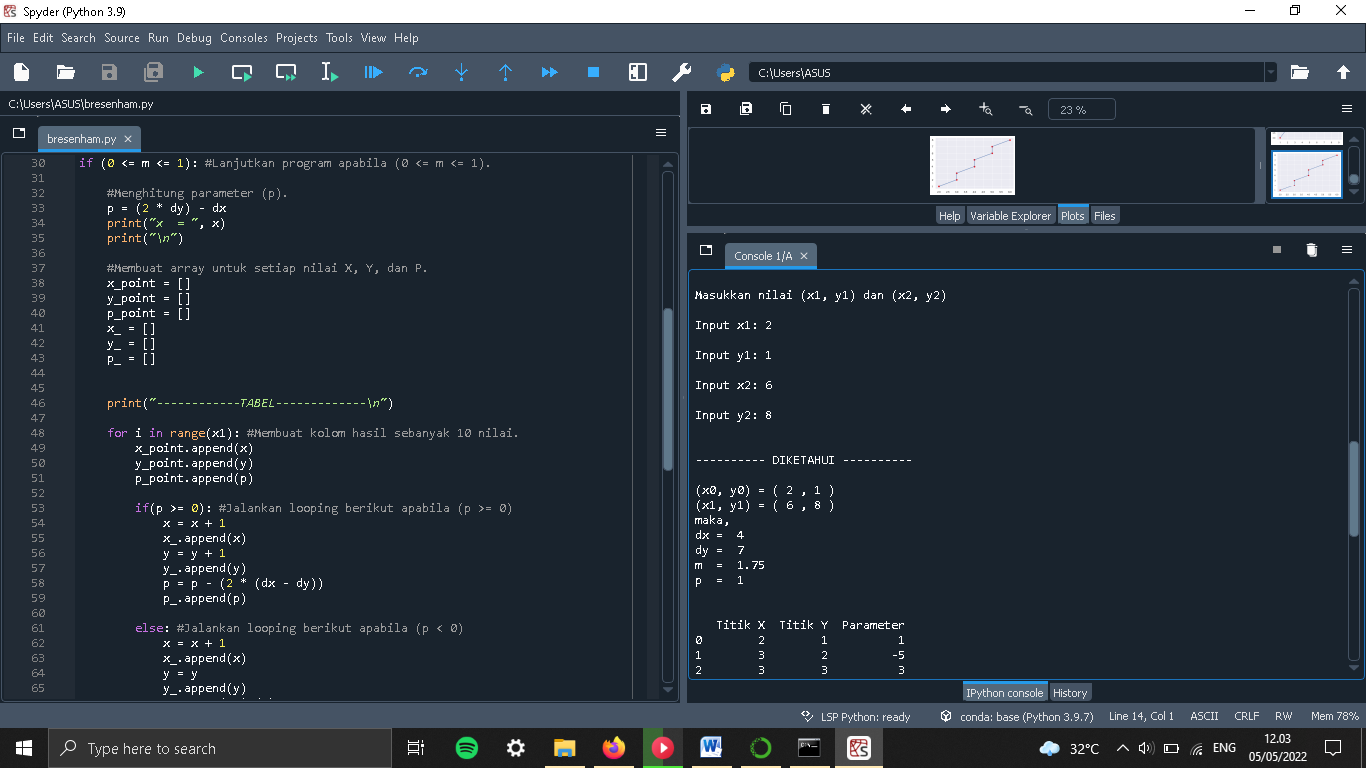
#Menghitung parameter (p).

p = (2 \* dx) - dy

print("p = ", p)

print("\n")

jika deklarasi sudah dilakukan, maka output akan terlihat seperti berikut,



* Langkah 3, membuat array untuk menyimpan hasil looping.

x\_point = []

y\_point = []

p\_point = []

x\_ = []

y\_ = []

p\_ = []

* Langkah 4, membuat tabel dan titiknya. Disini akan dilakukan looping terhadap titik x, y, dan parameter (p) dan looping akan berhenti saat x = x1.

for i in range(x1): #Membuat kolom hasil sampai titik X = X1

Lalu, apabila p >= 0, maka p akan dijumlah oleh d2, sehingga

p = p + d2

p = p + ( 2 × (dx – dy) )

Sedangkan titik x dan y akan ditambah menjadi x + 1 dan y + 1. Seperti pada program berikut,

if(p >= 0): #Jalankan looping berikut apabila (p >= 0)

x = x + 1

x\_.append(x)

y = y + 1

y\_.append(y)

p = p + (2 \* (dx - dy))

p\_.append(p)

Namun apabila p < 0, maka p akan dijumlah oleh d1, sehingga

p = p – d2

p = p + ( 2 × dy)

Sedangkan hanya titik y akan ditambah menjadi y + 1 dan titik x tetap.

else: #Jalankan looping berikut apabila (p < 0)

x = x

x\_.append(x)

y = y + 1

y\_.append(y)

p = p + (2 \* dx)

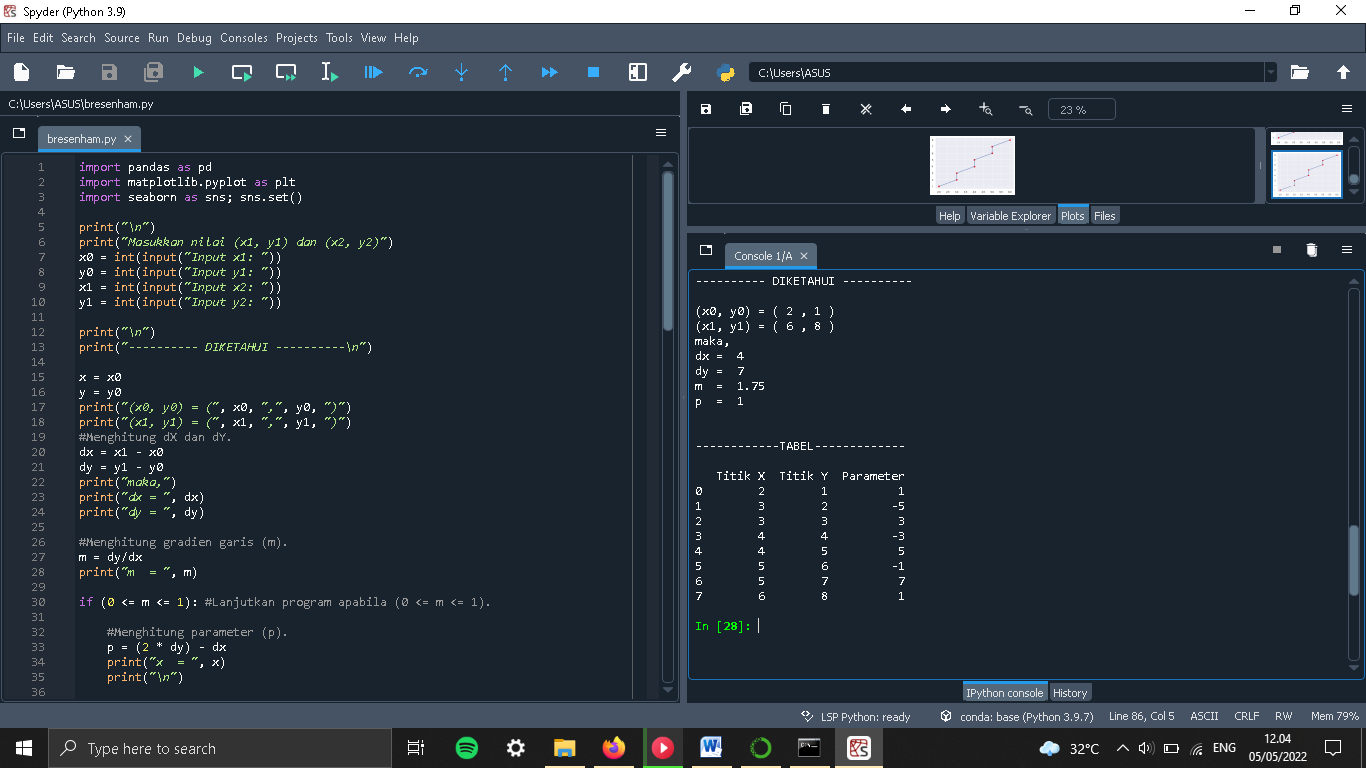
p\_.append(p)

selama looping dilakukan, hasil akan tersimpan pada array yang sudah dibuat pada langkah 3. Selanjutnya akan ditampilkan pada tabel dengan menggunakan program berikut,

print(pd.DataFrame(

{"Titik X":x\_point,"Titik Y":y\_point,"Parameter":p\_point}))

sehingga output akan terlihat seperti berikut,



* Langkah 5, output garis berupa gambar. Setelah tabel dibuat dan seluruh koordinat sudah diketahui, maka akan ditarik garis berdasarkan koordinat yang sudah diketahui. Berikut adalah hasil dari perhitungan garis dengan titik koordinat awal (2, 1) dan koordinat titik akhir (6, 8) menggunakan algoritma bresenham.

