PengenalanSymPy

March 16, 2019

Pengenalan SymPy
Fadjar Fathurrahman

1 Pengenalan

SymPy adalah pustaka Python yang dapat digunakan untuk melakukan perhitungan matematika simbolik seperti aljabar dan teori bilangan. SymPy dapat digunakan sebagai alternatif dari sistem aljabar komputer (*computer algebra system*) komersial seperti Mathematica dan Maple.

```
In [1]: import sympy
        from sympy import *
```

```
In [2]: print(sympy.__version__)
```

```
1.3
```

```
In [3]: init_printing(use_latex=True)
```

2 Operasi dasar

Fungsi `symbols` dapat digunakan untuk membuat objek simbolik. Sebagai contoh, kita akan membuat suatu tiga objek simbolik `x`, `y`, dan `z`:

```
In [4]: x, y, z = symbols("x y z")
```

Jika kita menggunakan `init_print(use_latex=True)`, maka tampilan dari objek tersebut merupakan x, y, dan z:

```
In [12]: x, y, z
```

Out[12]:

$$(x, \quad y, \quad z)$$

Tipe dari `x`, `y`, dan `z` adalah `Symbol` (lebih lengkapnya adalah `sympy.symbol.Symbol`).

```
In [16]: type(x), type(y), type(z)
```

Out[16]: (sympy.core.symbol.Symbol, sympy.core.symbol.Symbol, sympy.core.symbol.Symbol)

Mari kita mulai dengan perhitungan sederhana. Misalkan kita ingin menghitung hasil dari

$$2x + \frac{3x}{5x^2} - 7x$$

In [5]:
```
expr = 2*x + 3*x/(5*x**2) - 7*x
expr
```

Out[5]:

$$-5x + \frac{3}{5x}$$

In [13]:
```
factor(expr)
```

Out[13]:

$$-\frac{25x^2 - 3}{5x}$$

2.1 Substitusi

Salah satu operasi yang sering dilakukan adalah substitusi. Misalkan kita ingin melakukan substitusi $x \rightarrow 3y$, maka kita dapat menggunakan metode `subs` dari suatu objek simbolik (`Symbol`).

In [15]:
```
expr.subs(x, 3*y)
```

Out[15]:

$$-15y + \frac{1}{5y}$$

In [17]:
```
expr.subs(x, 1)
```

Out[17]:

$$-\frac{22}{5}$$

In [25]:
```
expr.subs(x, 2/3)
```

Out[25]:

$$-2.43333333333333$$

In [26]:
```
expr.subs(x, Integer(2)/Integer(3))
```

Out[26]:

$$-\frac{73}{30}$$

```
In [30]: type(Integer(2)/Integer(3)), type(2/3)
```

```
Out[30]: (sympy.core.numbers.Rational, float)
```

```
In [32]: Rational(2,3)
```

Out[32]:

$$\frac{2}{3}$$

```
In [33]: expr.subs(x, Rational(2,3))
```

Out[33]:

$$-\frac{73}{30}$$

Untuk melakukan beberapa substitusi sekaligus, kita dapat memberikan list `(old, new)` ke metode `subs`. Sebagai contoh:

```
In [40]: expr = x**3 + 6*x*y - z
         expr, expr.subs( [(x, 2*x), (y, 4), (z, 0)] )
```

Out[40]:

$$\left(x^3 + 6xy - z, \quad 8x^3 + 48x\right)$$

2.2 Mengubah string menjadi ekspresi SymPy

Metode `sympify` dapat digunakan untuk mengubah suatu string menjadi ekspresi SymPy.

```
In [42]: str_expr = "x**2 + 2*y + 4/5"
         expr = sympify(str_expr)
         expr
```

Out[42]:

$$x^2 + 2y + \frac{4}{5}$$

```
In [43]: expr.subs(x, Rational(4,3))
```

Out[43]:

$$2y + \frac{116}{45}$$

2.3 Mengevaluasi ekspresi menjadi nilai numerik float

Metode `evalf` dapat digunakan untuk mengevaluasi suatu ekspresi menjadi nilai numerik float.

```
In [45]: expr = sqrt(Rational(2,3))
         expr, expr.evalf()
```

Out[45]:

$$\left(\frac{\sqrt{6}}{3}, \quad 0.816496580927726\right)$$

Karena kita menggunakan `from sympy import *` maka beberapa konstanta dan fungsi dari modul standard Python `math` akan digantikan dengan konstanta dan fungsi dari SymPy yang lebih cocok untuk perhitungan simbolik. Salah satu konstanta tersebut adalah `pi`. Perhatikan bahwa `pi` di sini adalah `pi` dari modul `sympy` bukan dari `math`.

```
In [52]: import math
         pi, math.pi
```

Out[52]:

$$\left(\pi, \quad 3.141592653589793\right)$$

Evaluasi nilai numerik dari π dalam 40 digit:

```
In [49]: pi.evalf(40)
```

Out[49]:

$$3.141592653589793238462643383279502884197$$

Evaluasi nilai π dalam 1000 digit. Kita menggunakan fungsi `print` agar lebih mudah ditampilkan (tidak menggunakan LaTeX).

```
In [51]: print(pi.evalf(1000))
```

```
3.14159265358979323846264338327950288419716939937510582097494459230781640628620899862803482534
```

Contoh lain:

```
In [56]: expr = 2*x/y**2 - 1
         expr
```

Out[56]:

$$\frac{2x}{y^{2}} - 1$$

```
In [59]: expr1 = expr.subs([ (x,1), (y,3)])
         expr1, expr1.evalf()
```

Out[59]:

$$\left(-\frac{7}{9}, \quad -0.777777777777778\right)$$

```
In [60]: expr.evalf(subs={x: 1, y: 3})
```

Out[60]:

$$-0.777777777777778$$

3 Kalkulus

Turunan dapat dihitung dengan menggunakan perintah `diff`:

```
In [9]: eksp = cos(x**3)*cos(y**4)*cos(z**2)
        eksp
```

Out[9]:

$$\cos\left(x^{3}\right)\cos\left(y^{4}\right)\cos\left(z^{2}\right)$$

Turunan pertama terhadap `x`. Secara default `diff` akan menghitung turunan pertama.

```
In [10]: diff(eksp, x)
```

Out[10]:

$$-3x^{2}\sin\left(x^{3}\right)\cos\left(y^{4}\right)\cos\left(z^{2}\right)$$

Turunan kedua terhadap `y`

```
In [11]: diff(eksp, y, 2)
```

Out[11]:

$$-4y^{2}\left(4y^{4}\cos\left(y^{4}\right)+3\sin\left(y^{4}\right)\right)\cos\left(x^{3}\right)\cos\left(z^{2}\right)$$

Turunan keempat terhadap `z`

```
In [13]: diff(eksp, z, 4)
```

Out[13]:

$$4\left(4z^{4}\cos\left(z^{2}\right)+12z^{2}\sin\left(z^{2}\right)-3\cos\left(z^{2}\right)\right)\cos\left(x^{3}\right)\cos\left(y^{4}\right)$$

Integral dapat dihitung dengan menggunakan fungsi `integrate`.

```
In [17]: eksp = 2*x/(3*y) + z
         eksp
```

Out[17]:

$$\frac{2x}{3y} + z$$

```
In [18]: integrate(eksp, x)
```

Out[18]:

$$\frac{x^2}{3y} + xz$$

```
In [19]: integrate(eksp, y)
```

Out[19]:

$$\frac{2x}{3}\log(y) + yz$$

Ingat: Secara default fungsi `log` dihitung terhadap bilangan Euler e (logaritma natural)

```
In [22]: exp(1), log(exp(1))
```

Out[22]:

$$(e, \quad 1)$$

```
In [23]: integrate(eksp, z)
```

Out[23]:

$$\frac{2xz}{3y} + \frac{z^2}{2}$$

Contoh integral definit (dengan batas atas dan bawah).

```
In [24]: integrate(eksp, (z,0,1))
```

Out[24]:

$$\frac{2x}{3y} + \frac{1}{2}$$

Fungsi `Integral` dapat digunakan untuk membuat sebuah objek integral yang belum dievaluasi (berguna untuk menampilkan integral).

```
In [27]: myinteg = Integral(eksp, (z,0,1))
         myinteg
```

Out[27]:

$$\int_0^1 \left(\frac{2x}{3y} + z\right) dz$$

Untuk mengevaluasi integral tersebut, kita dapat menggunakan metode `doit()`.

In [28]:
```
myinteg.doit()
```

Out[28]:

$$\frac{2x}{3y} + \frac{1}{2}$$

Contoh lain, fungsi Gaussian:

In [32]:
```
eksp = exp(-x**2)
eksp
```

Out[32]:

$$e^{-x^2}$$

Tanda oo dapat digunakan untuk merepresentasikan tak hingga (∞)

In [33]:
```
myinteg = Integral(eksp, (x,-oo,oo))
myinteg
```

Out[33]:

$$\int_{-\infty}^{\infty} e^{-x^2}\, dx$$

In [34]:
```
myinteg.doit()
```

Out[34]:

$$\sqrt{\pi}$$

Fungsi Gaussian dikalikan dengan x^2

In [37]:
```
eksp = x**2*exp(-x**2)
eksp
```

Out[37]:

$$x^2 e^{-x^2}$$

In [38]:
```
myinteg = Integral(eksp, (x,-oo,oo))
myinteg
```

Out[38]:

$$\int_{-\infty}^{\infty} x^2 e^{-x^2}\, dx$$

In [39]:
```
myinteg.doit()
```

Out[39]:

$$\frac{\sqrt{\pi}}{2}$$

Perhitungan integral secara simbolik sangat sulit. Tidak semua integral dapat diselesaikan secara analitik oleh SymPy.

4 Matriks

Tipe `Matrix` dapat digunakan untuk merepresentasikan suatu matriks dan juga vektor (kolom atau baris).

In [62]:
```
mat = Matrix([[1,2,3], [4,5,6], [7,8,9]] )
mat
```

Out[62]:

$$\begin{bmatrix} 1 & 2 & 3 \\ 4 & 5 & 6 \\ 7 & 8 & 9 \end{bmatrix}$$

In [63]:
```
mat.row(0)
```

Out[63]:

$$\begin{bmatrix} 1 & 2 & 3 \end{bmatrix}$$

In [67]:
```
mat.col(1)
```

Out[67]:

$$\begin{bmatrix} 2 \\ 5 \\ 8 \end{bmatrix}$$

In [71]:
```
3*ones(3,1) - mat.col(2)
```

Out[71]:

$$\begin{bmatrix} 0 \\ -3 \\ -6 \end{bmatrix}$$

In [72]:
```
mat
```

Out[72]:

$$\begin{bmatrix} 1 & 2 & 3 \\ 4 & 5 & 6 \\ 7 & 8 & 9 \end{bmatrix}$$

In [77]:
```
mat = mat*2
mat
```

Out[77]:

$$\begin{bmatrix} 4 & 8 & 12 \\ 16 & 20 & 24 \\ 28 & 32 & 36 \end{bmatrix}$$

```
In [78]: mat1 = mat

In [79]: mat1
```

Out[79]:

$$\begin{bmatrix} 4 & 8 & 12 \\ 16 & 20 & 24 \\ 28 & 32 & 36 \end{bmatrix}$$

```
In [80]: mat1 = mat1*3
         mat1, mat
```

Out[80]:

$$\left(\begin{bmatrix} 12 & 24 & 36 \\ 48 & 60 & 72 \\ 84 & 96 & 108 \end{bmatrix}, \quad \begin{bmatrix} 4 & 8 & 12 \\ 16 & 20 & 24 \\ 28 & 32 & 36 \end{bmatrix} \right)$$

```
In [82]: mat1[0] = 2

In [88]: mat1[0:2,2:3] = Matrix([[0],[0]])

In [89]: mat1
```

Out[89]:

$$\begin{bmatrix} 2 & 24 & 0 \\ 48 & 60 & 0 \\ 84 & 96 & 108 \end{bmatrix}$$

```
In [91]: mat1[:,2] = Matrix([ [100], [33], [44]])
         mat1
```

Out[91]:

$$\begin{bmatrix} 2 & 24 & 100 \\ 48 & 60 & 33 \\ 84 & 96 & 44 \end{bmatrix}$$

```
In [94]: mat1[0,:] = mat1[0,:]/2
         mat1
```

Out[94]:

$$\begin{bmatrix} 1 & 12 & 50 \\ 48 & 60 & 33 \\ 84 & 96 & 44 \end{bmatrix}$$

```
In [95]: mat1[1,:] = mat1[1,:] - 48*mat1[0,:]
         mat1
```

Out[95]:

$$\begin{bmatrix}1 & 12 & 50\\0 & -516 & -2367\\84 & 96 & 44\end{bmatrix}$$

```
In [96]: mat1[1,:] = mat1[1,:]/(-516)
         mat1
```

Out[96]:

$$\begin{bmatrix}1 & 12 & 50\\0 & 1 & \frac{789}{172}\\84 & 96 & 44\end{bmatrix}$$

```
In [97]: mat1[2,:] = mat1[2,:] - 84*mat1[0,:]
         mat1
```

Out[97]:

$$\begin{bmatrix}1 & 12 & 50\\0 & 1 & \frac{789}{172}\\0 & -912 & -4156\end{bmatrix}$$

```
In [98]: mat1[2,:] = mat1[2,:] - mat1[1,:]*-912
         mat1
```

Out[98]:

$$\begin{bmatrix}1 & 12 & 50\\0 & 1 & \frac{789}{172}\\0 & 0 & \frac{1184}{43}\end{bmatrix}$$