Belajar Matematika dengan Python + Sympy

19 Juni 2021

I Wayan Sudiarta, Ph.D.

Tutorial Python+SymPy ini tidak mengajarkan dasar pemrograman Python. Hanya memberikan bagian yang dibutuhkan untuk penggunaan operasi matematis dengan modul Sympy.

Catatan Penting:

1. Bagian yang diawali dengan **tanda pagar (#)** merupakan catatan atau komentar, tidak dievaluasi/dijalankan
2. Menjalankan, run cell dengan cara tekan **Shift+Enter**
3. Penulisan dengan huruf kecil dan huruf besar dibedakan, contoh abs berbeda dengan Abs.

In [1]:
```
# Mari mulai dengan import modul SymPy
from sympy import *
init_printing()
```

Catatan: Ingat! Baris pertama **# Mari Mulai...** merupakan catatan

Baris kedua import modul SymPy *, tanda bintang ini artinya semua.

Baris ketiga **init_printing()** menginisiasi setting printing dengan tampilan terbaik pada sistem Anda.

Sebagai Kalkulator

Penggunaan Python paling sederhana adalah sebagai kalkulator canggih. Python menggunakan simbol-simbol berikut ini untuk melakukan operasi matematis:

1. + - * / untuk operasi penjumlahan, pengurangan, perkalian dan pembagian
2. ** untuk operasi pangkat
3. // untuk pembagian bilangan bulat

Variabel dideklarasikan dengan cara memberi nilai pada variabel, contoh a = 1.234

In [2]:
```
a = 1.234
b = 4.321
```

In [3]:
```
a + b
```
Out[3]: 5.555

In [4]:
```
b - a
```
Out[4]: 3.087

In [5]:
```
a*b
```
Out[5]: 5.332114

In [6]:
```
a/b
```
Out[6]: 0.285582041194168

In [7]:
```
a**2
```
Out[7]: 1.522756

In [8]:
```
5//2
```
Out[8]: 2

Simbol untuk konstanta Matematis

Sympy menggunakan variabel-variabel berikut ini untuk konstanta matematis:

1. **pi** untuk bilangan π
2. **E** untuk bilangan e
3. **oo** untuk tak hingga
4. **I** untuk bilangan imajiner

In [9]:
```
# Luas lingkaran
r = 3
luas = pi*r**2
luas
```
Out[9]: 9π

Operasi dengan modul SymPy dilakukan secara simbolik, untuk mendapatkan nilai numerik dilakukan dengan fungsi **N(...)**, perhatikan contoh berikut. N(pi, 10) mendapatkan 10 digit angka dibelakang koma (titik).

In [10]:
```
N(luas)
```
Out[10]: 28.2743338823081

In [11]:
```
N(pi, 10)
```
Out[11]: 3.141592654

In [12]:
```
N(pi,100)
```
Out[12]: $3.14159265358979323846264338327950288419716939937510582097494459230781640628620899862$

In [13]:
```
z1 = 1 + 2*I
z2 = 5 - I
```
In [14]:
```
z = z1 + z2
```
In [15]:
```
z
```
Out[15]: $6 + i$

Fungsi Matematis

Operasi dengan fungsi-fungsi matematis berikut ini. x untuk variabel input.

1. Trigonometri : sin(x), cos(x), tan(x), cot(x), asin(x), acos(x), atan(x), acot(x), sec(x), csc(x)
2. Eksponensial : exp(x)
3. Akar pangkat 2 : sqrt(x), akar pangkat 3: cbrt(x)
4. Logaritma natural : log(x)
5. Logaritma $log_b a$ basis b : log(x, b)
6. Hiperbolik : sinh(x), cosh(x), tanh(x), coth(x), asinh(x), acosh(x), atanh(x), acoth(x)
7. Gamma ($\Gamma(x)$): gamma(x)

In [16]:
```
x = pi/4
```
In [17]:
```
sin(x)
```
Out[17]: $\frac{\sqrt{2}}{2}$

In [18]:
```
sqrt(x)
```
Out[18]: $\frac{\sqrt{\pi}}{2}$

In [19]:
```
log(x)
```
Out[19]: $\log\left(\frac{\pi}{4}\right)$

Operasi Bilangan Kompleks

z = x + I*y

In [20]:
```
# Bilangan kompleks
z = 7 - 2*I
```

In [21]:
```
# Bagian Riil
re(z)
```

Out[21]: 7

In [22]:
```
# Bagian Imajiner
im(z)
```

Out[22]: -2

In [23]:
```
# Modulus
Abs(z)
```

Out[23]: $\sqrt{53}$

In [24]:
```
# Konjugat
conjugate(z)
```

Out[24]: $7 + 2i$

In [25]:
```
# Argument/theta
arg(z)
```

Out[25]: $-\operatorname{atan}\left(\frac{2}{7}\right)$

Operasi Matematis Simbolik

Sebelum melakukan operasi matematis secara simbolik, deklarasi objek berupa simbol perlu dilakukan dengan menggunakan fungsi **symbols(...)**

In [26]:
```
x = symbols('x')
```

In [27]:
```
y = (x+2)**2
y
```

Out[27]: $(x + 2)^2$

In [28]:
```
y2 = expand(y)
y2
```

Out[28]: $x^2 + 4x + 4$

In [29]:
```
factor(y2)
```

Out[29]: $(x+2)^2$

In [30]:
```
# deklarasi lebih dari satu simbol
theta1, theta2 = symbols('theta_1 theta_2')
```

In [31]:
```
y = sin(theta1 + theta2)
```

In [32]:
```
y
```

Out[32]: $\sin(\theta_1 + \theta_2)$

In [33]:
```
expand_trig(y)
```

Out[33]: $\sin(\theta_1)\cos(\theta_2) + \sin(\theta_2)\cos(\theta_1)$

In [34]:
```
# deklarasi varial theta dgn asumsi riil
theta = symbols('theta', real = True)
```

In [35]:
```
z = exp(I*theta)
z
```

Out[35]: $e^{i\theta}$

In [36]:
```
re(z)
```

Out[36]: $\cos(\theta)$

In [37]:
```
im(z)
```

Out[37]: $\sin(\theta)$

Kalkulus

In [38]:
```
x, y = symbols('x y')
```

In [39]:
```
f = x**3 + 2*x**2 - 5
```

In [40]:
```
# Turunan pertama
fd = diff(f, x)
fd
```

Out[40]: $3x^2 + 4x$

In [41]:
```
# Turunan kedua
fd2 = diff(f, x, 2)
fd2
```

Out[41]: $2(3x + 2)$

In [42]:
```
# atau dengan penulisan berikut
diff(f,x,x)
```

Out[42]: $2(3x+2)$

In [43]:
```
# Substitusi x = 2
f.subs(x, 2)
```

Out[43]: 11

In [44]:
```
# fungsi multivariable
g = x*y**2
g
```

Out[44]: xy^2

In [45]:
```
# turunan gx
diff(g, x)
```

Out[45]: y^2

In [46]:
```
# Turunan gxy
diff(g, x, y)
```

Out[46]: $2y$

In [47]:
```
# Integral tak tentu
integrate(f, x)
```

Out[47]: $\frac{x^4}{4} + \frac{2x^3}{3} - 5x$

Menghitung integral tentu $\int_0^3 f(x)dx$

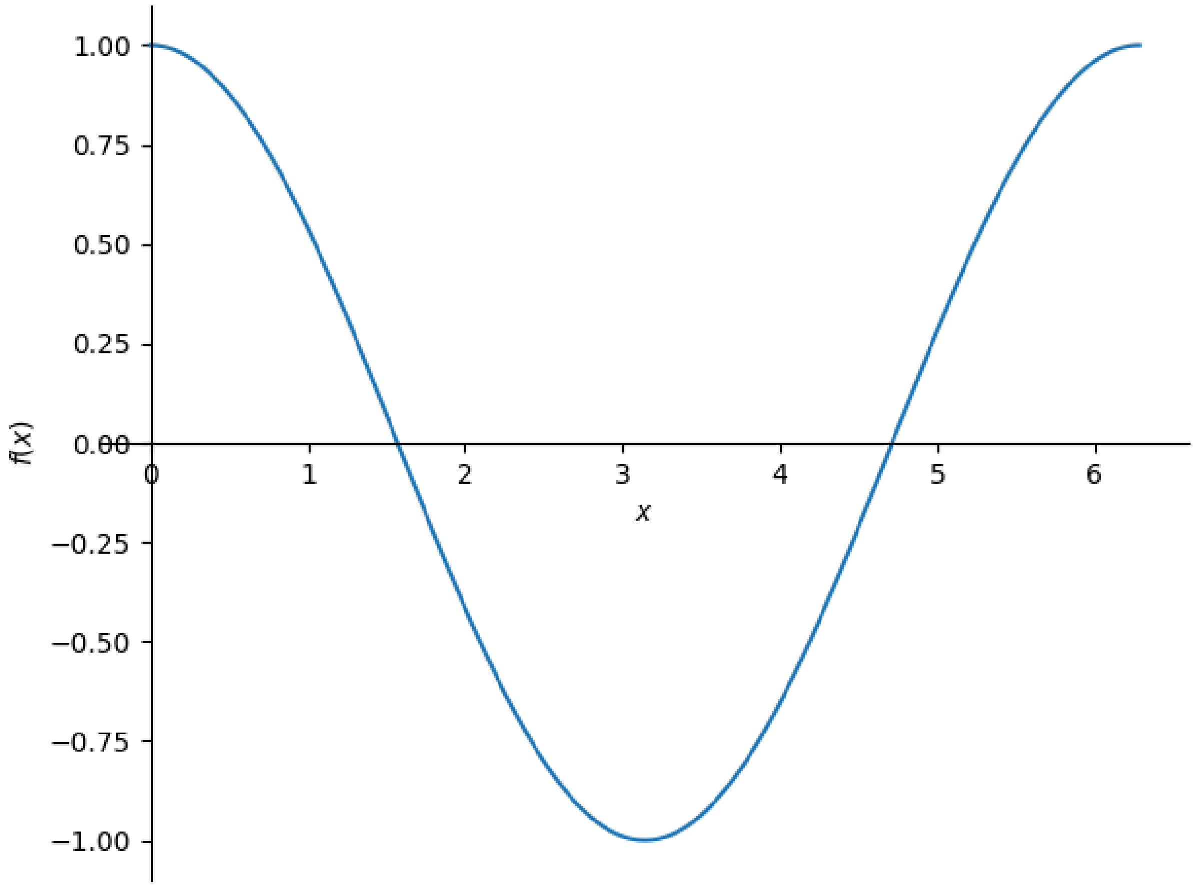
In [48]:
```
integrate(f, (x, 0, 3))
```

Out[48]: $\frac{93}{4}$

Membuat Grafik dengan plot

In [49]:
```
x = symbols('x')
```

In [50]:
```
plot(cos(x), (x,0,2*pi))
```



Out[50]:

```
<sympy.plotting.backends.matplotlibbackend.matplotlib.MatplotlibBackend at 0x21e5bf2
6b10>
```

Dengan bantuan modul Matplotlib dan NumPy

In [51]:

```
import matplotlib.pyplot as plt
import numpy as np
```

In [52]:

```
f = sin(x)
```

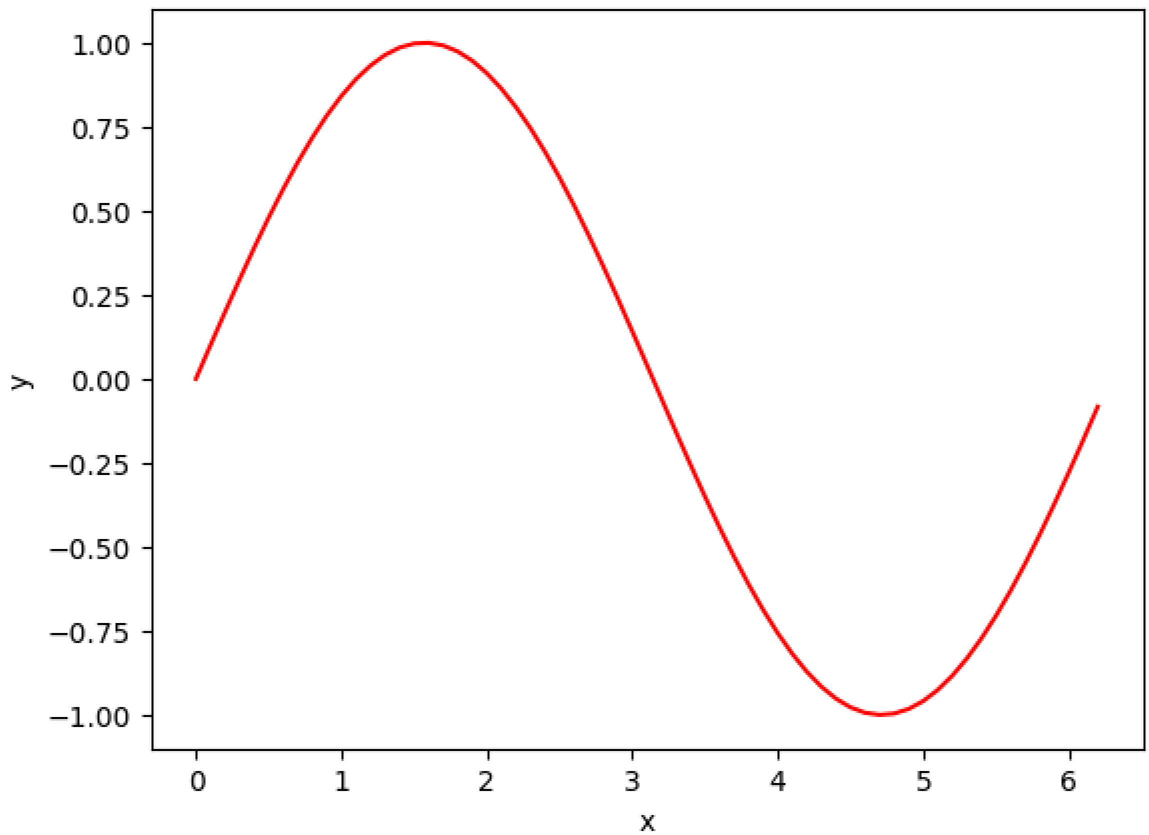
In [53]:

```
# Gunakan lambdify untuk ke objek numpy
ff = lambdify(x, f, 'numpy')
ff(0)
```

Out[53]: 0.0

In [54]:

```
# buat data untuk plot
xx = np.arange(0.0,2*np.pi,0.1)
yy = ff(xx)
plt.plot(xx,yy, '-r')
plt.xlabel('x')
plt.ylabel('y')
plt.show()
```



Deret Taylor dan Fourier

In [55]:
```
x = symbols('x')
```

Deret Taylor

$$f(x) = f(a) + f'(a)(x-a) + \frac{f''(a)}{2!}(x-a)^2 + \dots$$

In [56]:
```
f = cos(x)
```

In [57]:
```
g = series(f)
g
```

Out[57]:

$$1 - \frac{x^2}{2} + \frac{x^4}{24} + O\left(x^6\right)$$

In [58]:
```
# Hilangkan Orde kesalahan
g = series(f).removeO()
g
```

Out[58]:

$$\frac{x^4}{24} - \frac{x^2}{2} + 1$$

In [59]:
```
# sampai orde 10
g = series(f, n = 10)
```

```
g
```

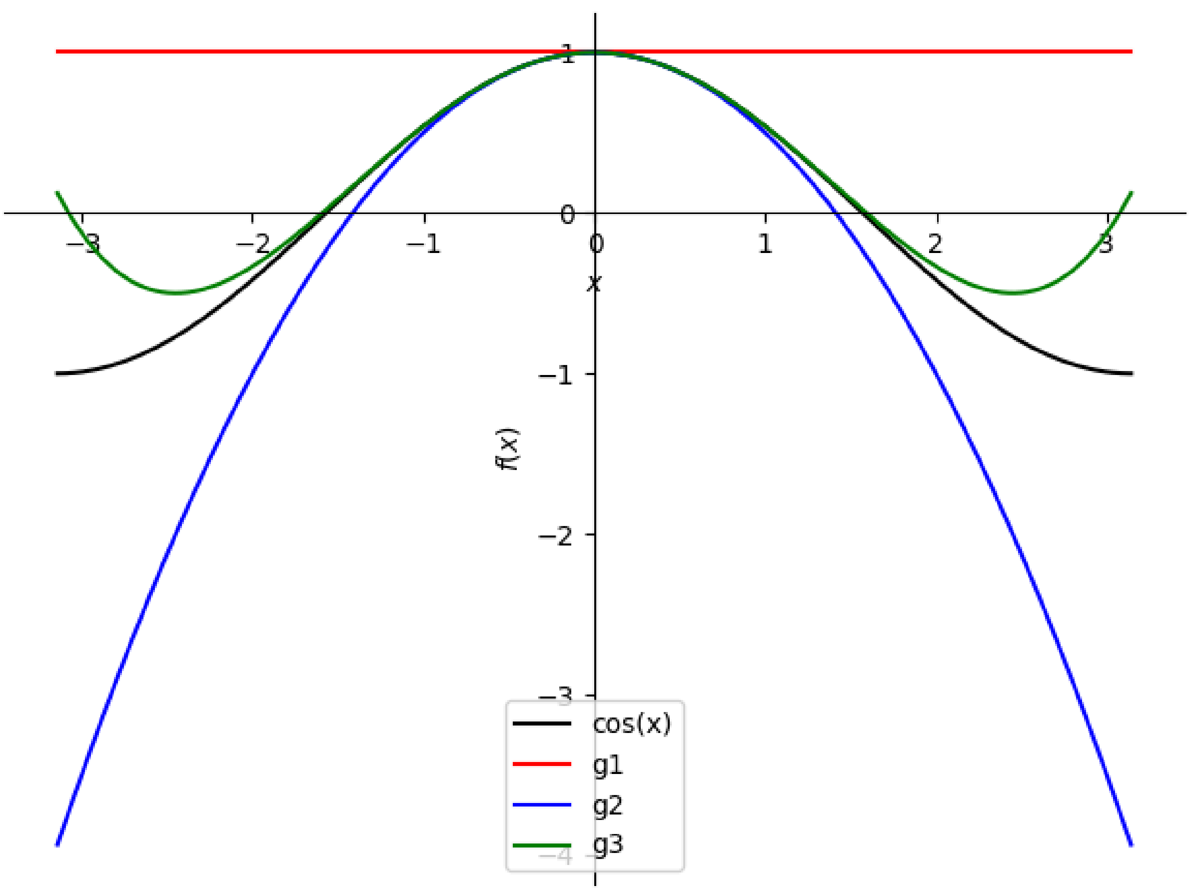
Out[59]: $1 - \frac{x^2}{2} + \frac{x^4}{24} - \frac{x^6}{720} + \frac{x^8}{40320} + O\left(x^{10}\right)$

In [60]:
```
# mulai dari a = x0
g = series(f, x0 = pi, n = 5)
g
```

Out[60]: $-1 + \frac{(x-\pi)^2}{2} - \frac{(x-\pi)^4}{24} + O\left((x-\pi)^5; x \to \pi\right)$

In [61]:
```
g1 = series(f, n = 1).removeO()
g2 = series(f, n = 3).removeO()
g3 = series(f, n = 5).removeO()

# Grafik
p = plot(f, g1, g2, g3, (x, -pi, pi), show=False, legend=True)
# ubah label
p[0].line_color = 'k' # black
p[0].label = 'cos(x)'
p[1].line_color = 'r' # red
p[1].label = 'g1'
p[2].line_color = 'b'
p[2].label = 'g2'
p[3].line_color = 'g'
p[3].label = 'g3'
p.show()
```



Deret Fourier

$$f(x) = \frac{a_0}{2} + \sum_{n=1}^{\infty}(a_n \cos(\frac{2n\pi x}{L}) + b_n \sin(\frac{2n\pi x}{L}))$$

Mengikuti contoh di https://docs.sympy.org/latest/modules/series/fourier.html

In [62]:
```
f = x
```

In [63]:
```
s = fourier_series(f, (x, -pi, pi))
```

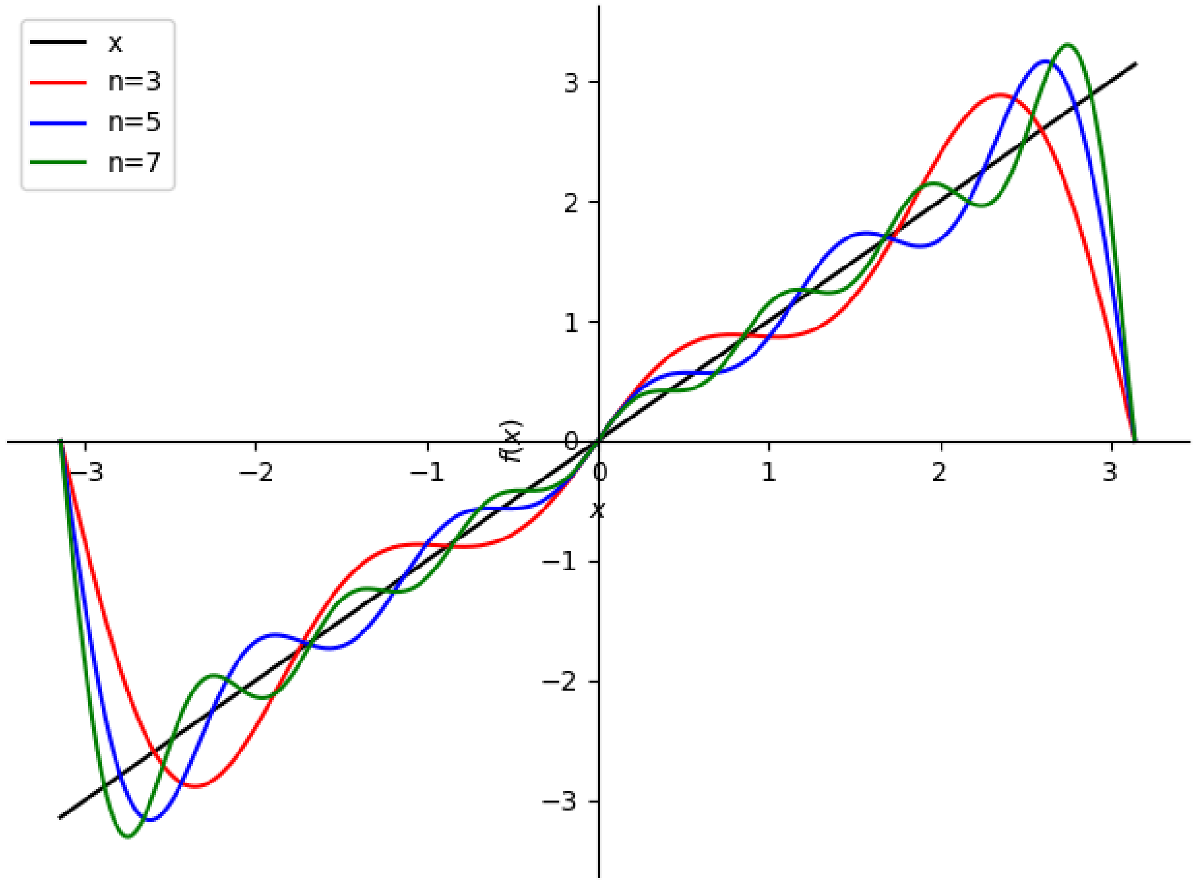
In [64]:
```
s
```

Out[64]:

$$2\sin(x) - \sin(2x) + \frac{2\sin(3x)}{3} + \ldots$$

In [65]:
```
# Pemotongan dengan truncate
s1 = s.truncate(n = 3)
s2 = s.truncate(n = 5)
s3 = s.truncate(n = 7)
```

In [66]:
```
# Grafik
p = plot(f, s1, s2, s3, (x, -pi, pi), show=False, legend=True)
# ubah label
p[0].line_color = 'k'
p[0].label = 'x'
p[1].line_color = 'r'
p[1].label = 'n=3'
p[2].line_color = 'b'
p[2].label = 'n=5'
p[3].line_color = 'g'
p[3].label = 'n=7'
p.show()
```



Penjumlahan dan Perkalian

$$S = \sum_{n=1}^{M} U_n$$

$$P = \prod_{n=1}^{M} U_n$$

In [67]:
```
n = symbols('n', integer=True)
```

$$s = \sum_{n=1}^{5} n^2$$

In [68]:
```
s = summation(n**2, (n, 1, 5))
s
```

Out[68]: 55

In [69]:
```
x = symbols('x')
f = summation(x**n/factorial(n), (n, 0, 5))
f
```

Out[69]: $\frac{x^5}{120} + \frac{x^4}{24} + \frac{x^3}{6} + \frac{x^2}{2} + x + 1$

$$p = \prod_{n=1}^{6} \frac{1}{n}$$

In [70]:
```
p = product(1/n, (n,1,6))
p
```

Out[70]: $\frac{1}{720}$

Menyelesaikan Persamaan

Untuk memudahkan menyelesaikan persamaan dengan SymPy, semua persamaan dibentuk menjadi $f(x) = 0$. Jadi nol di sisi kanan persamaan.

Contoh $x^2 = 4$, diubah menjadi persamaan $x^2 - 4 = 0$.

In [71]:
```
x, y = symbols('x y')
```

In [72]:
```
eq = 2*x**2 - 3*x + 5
eq
```

Out[72]: $2x^2 - 3x + 5$

In [73]:
```
# solusi persamaan kuadrat
solve(eq, x)
```

Out[73]: $\left[\frac{3}{4} - \frac{\sqrt{31}i}{4}, \frac{3}{4} + \frac{\sqrt{31}i}{4}\right]$

In [74]:
```
a, b, c = symbols('a b c')
```

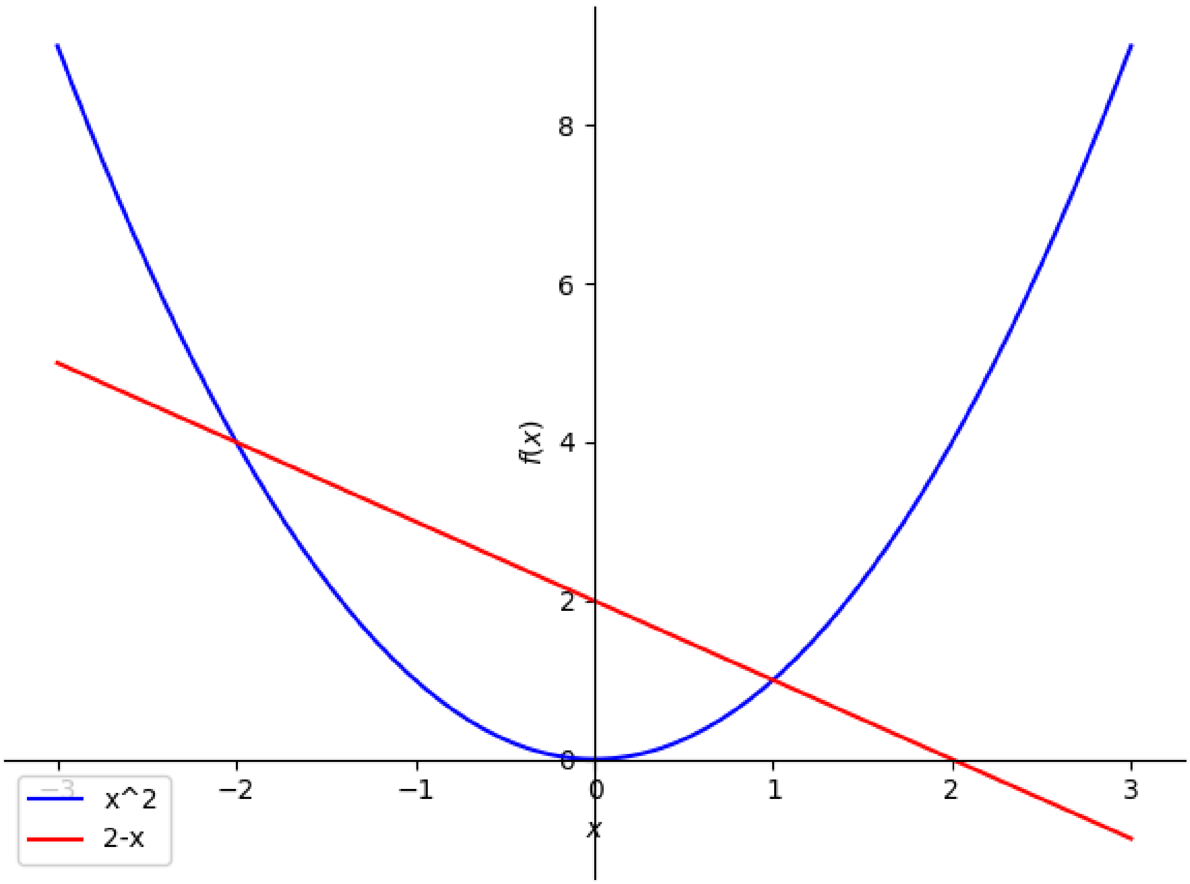
In [75]:
```
# rumus abc
solve (a*x**2 + b*x + c, x)
```

Out[75]: $\left[\frac{-b - \sqrt{-4ac + b^2}}{2a}, \frac{-b + \sqrt{-4ac + b^2}}{2a}\right]$

In [76]:
```
# Mencari titik potong 2 kurva
# y = x^2 dan y = 2 - x
# diubah menjadi dua persamaan:
eq1 = y - x**2
eq2 = y + x - 2
solve((eq1, eq2), x, y)
```

Out[76]: $[(-2,\ 4),\ (1,\ 1)]$

In [77]:
```
p = plot(x**2, 2-x, (x, -3, 3), show=False, legend=True)
# ubah label
p[0].line_color = 'b'
p[0].label = 'x^2'
p[1].line_color = 'r'
p[1].label = '2-x'
p.show()
```



Aljabar Linier

Persamaan linier berbentuk:

$$
\begin{align}
x + y &= 3 \tag{1} \\
2x - y &= 10 \tag{2}
\end{align}
$$

dapat diubah ke bentuk persamaan matriks:

$$Mv = b$$

$$M = \begin{bmatrix} 1 & 1 \\ 2 & -1 \end{bmatrix}$$

$$v = \begin{bmatrix} x \\ y \end{bmatrix}$$

$$b = \begin{bmatrix} 3 \\ 10 \end{bmatrix}$$

Membentuk Augmented Matrix (M|b):

$$A = \begin{bmatrix} 1 & 1 & 3 \\ 2 & -1 & 10 \end{bmatrix}$$

In [78]:
```
M = Matrix([[1, 1],
            [2, -1]])
b = Matrix([[3],
            [10]])
M, b
```

Out[78]: $\left(\begin{bmatrix} 1 & 1 \\ 2 & -1 \end{bmatrix}, \begin{bmatrix} 3 \\ 10 \end{bmatrix} \right)$

In [79]:
```
# matriks augmented
A = Matrix([[1, 1, 3],
           [2, -1, 10]])
```

In [80]:
```
# solusi
solve_linear_system(A, x, y)
```

Out[80]: $\left\{ x : \frac{13}{3}, \ y : -\frac{4}{3} \right\}$

In [81]:
```
# solusi dengan cara invers
Minv = M.inv()
Minv
```

Out[81]: $\begin{bmatrix} \frac{1}{3} & \frac{1}{3} \\ \frac{2}{3} & -\frac{1}{3} \end{bmatrix}$

In [82]:
```
# solusi
v = Minv*b
v
```

Out[82]: $\begin{bmatrix} \frac{13}{3} \\ -\frac{4}{3} \end{bmatrix}$

In [83]:
```
# determinan matriks M
M.det()
```

Out[83]: -3

In [84]:
```
det(M)
```

Out[84]: -3

Persamaan Diferensial Biasa (PDB)

Contoh persamaan PDB:

$$y'' + y' + 16y = 0$$

perlu di ubah ke bentuk pers = 0 (seperti sebelumbnya)

Gunakan:

1. Function() untuk membuat fungsi
2. y(x).diff(x) untuk turunan pertama y, y'(x)
3. y(x).diff(x, x) untuk turunan kedua y, y''(x)

In [85]:
```
x = symbols('x')
y = Function('y')
```

In [86]:
```
# persamaan
pdb = y(x).diff(x,x) + y(x).diff(x) + 16*y(x)
```

In [87]:
```
# solusi
sol = dsolve(pdb, y(x))
sol
```

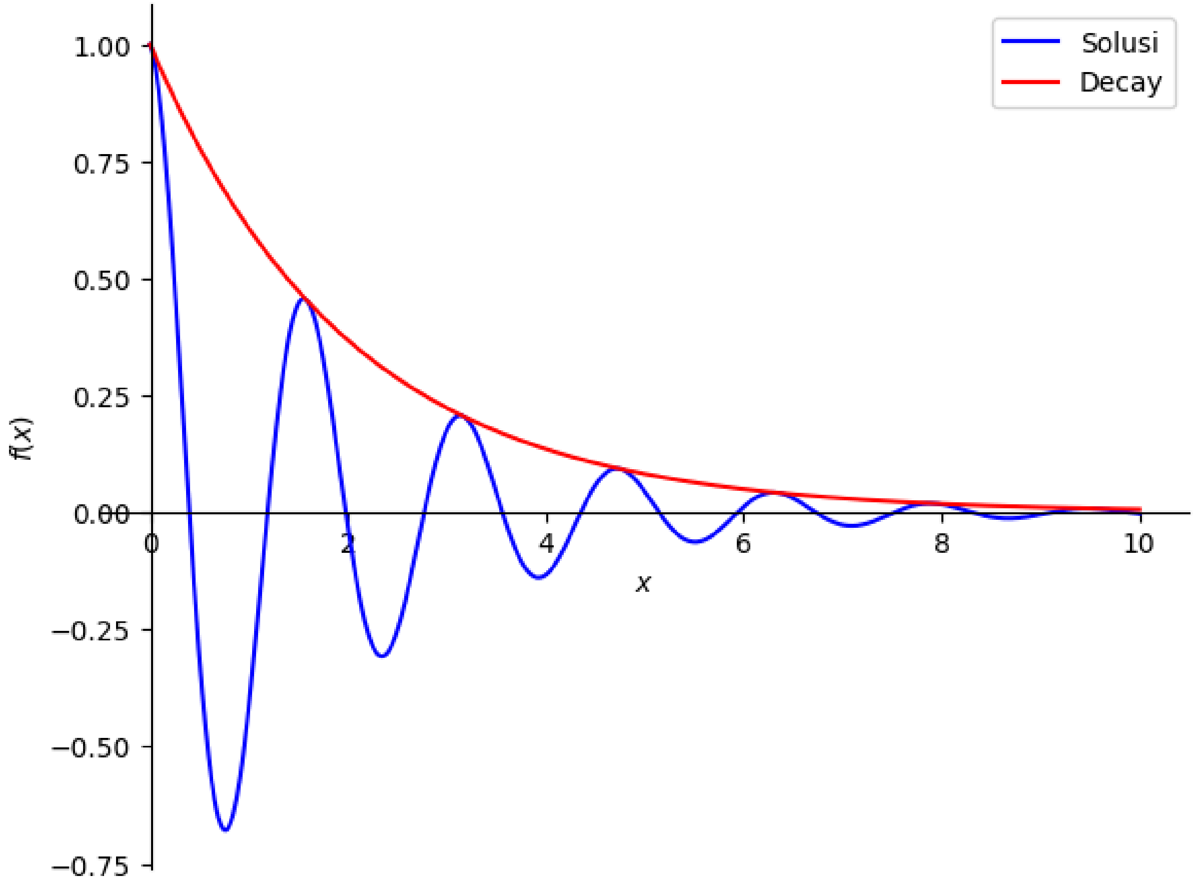
Out[87]:

$$y(x) = \left(C_1 \sin\left(\frac{3\sqrt{7}x}{2}\right) + C_2 \cos\left(\frac{3\sqrt{7}x}{2}\right)\right) e^{-\frac{x}{2}}$$

In [88]:
```
C1, C2 = symbols('C1 C2')
# umpama nilai awal
# y(0) = 1 -> C2 = 1
# y'(0) = 0 -> C1 = 0
ys = sol.rhs # dapatkan hasil sebelah kanan (rhs)
```

In [89]:
```
ys = ys.subs({C2: 1, C1:0})
```

In [90]:
```
p = plot(ys, exp(-x/2), (x,0,10), show=False, legend=True)
p[0].line_color = 'b'
p[0].label = 'Solusi'
p[1].line_color = 'r'
p[1].label = 'Decay'
p.show()
```



Grafik 3D

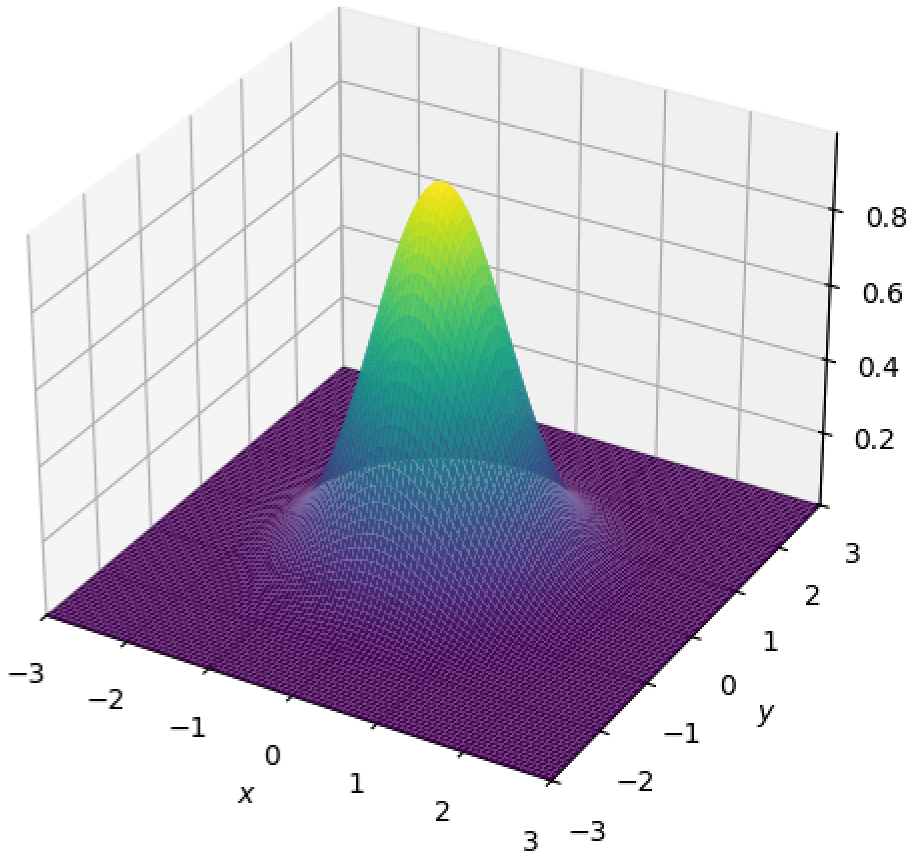
In [91]:
```
from sympy import symbols
from sympy.plotting import plot3d
```

In [92]:
```
x, y = symbols('x y')
```

Visualisasi $f(x, y)$

In [93]:
```
f = exp(-x**2 - y**2)
```

In [94]:
```
plot3d(f, (x, -3, 3), (y, -3, 3))
```



Out[94]: <sympy.plotting.backends.matplotlibbackend.matplotlib.MatplotlibBackend at 0x21e7baa9280>

Visualisasi Kurva/Lintasan 3D

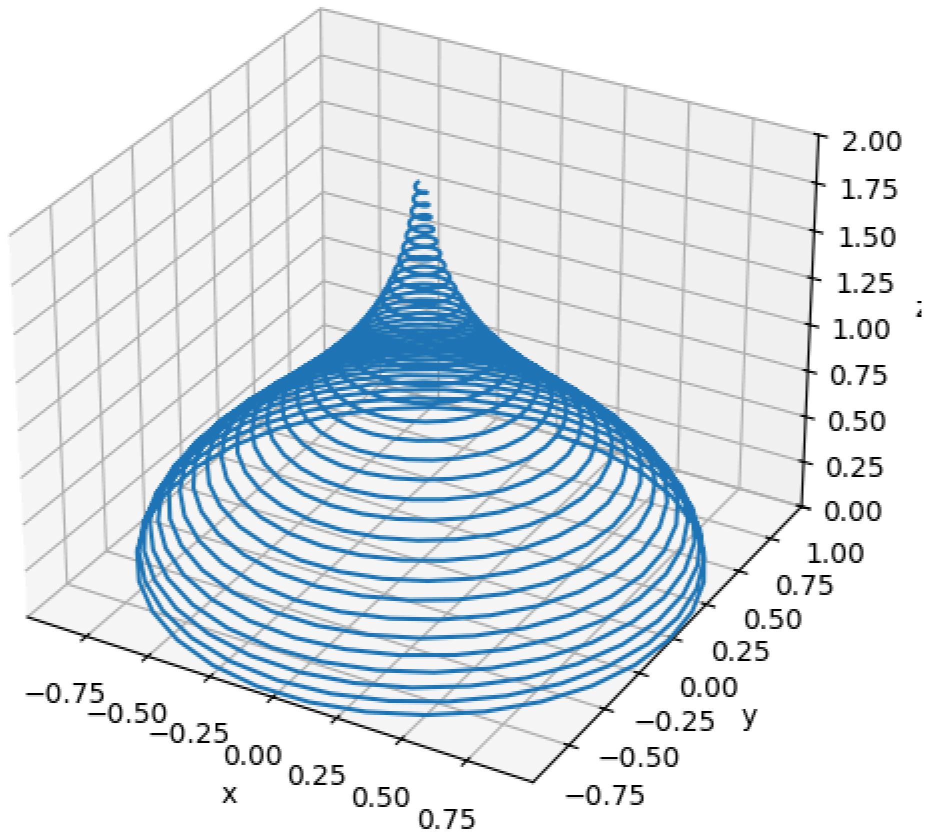
Kurva atau lintasan 3D memiliki persamaan:

$(x(t), y(t), z(t))$ dengan parameter t dapat berupa waktu.

In [95]:
```
from sympy.plotting import plot3d_parametric_line
```

In [96]:
```
t = symbols('t')
x = exp(-t**2)*sin(100*t)
y = exp(-t**2)*cos(100*t)
z = t
```

In [97]:
```
t = symbols('t')
plot3d_parametric_line(x, y , z, (t, 0, 2))
```



Out[97]: <sympy.plotting.backends.matplotlibbackend.matplotlib.MatplotlibBackend at 0x21e7ced8740>

Visualisasi Permukaan

Persamaan permukaan diberikan oleh:

$$x(u, v), y(u, v), z(u, v)$$

Sebagai contoh untuk permukaan bola:

$$x(u, v) = sin(u)cos(v)$$

$$y(u, v) = sin(u)sin(v)$$

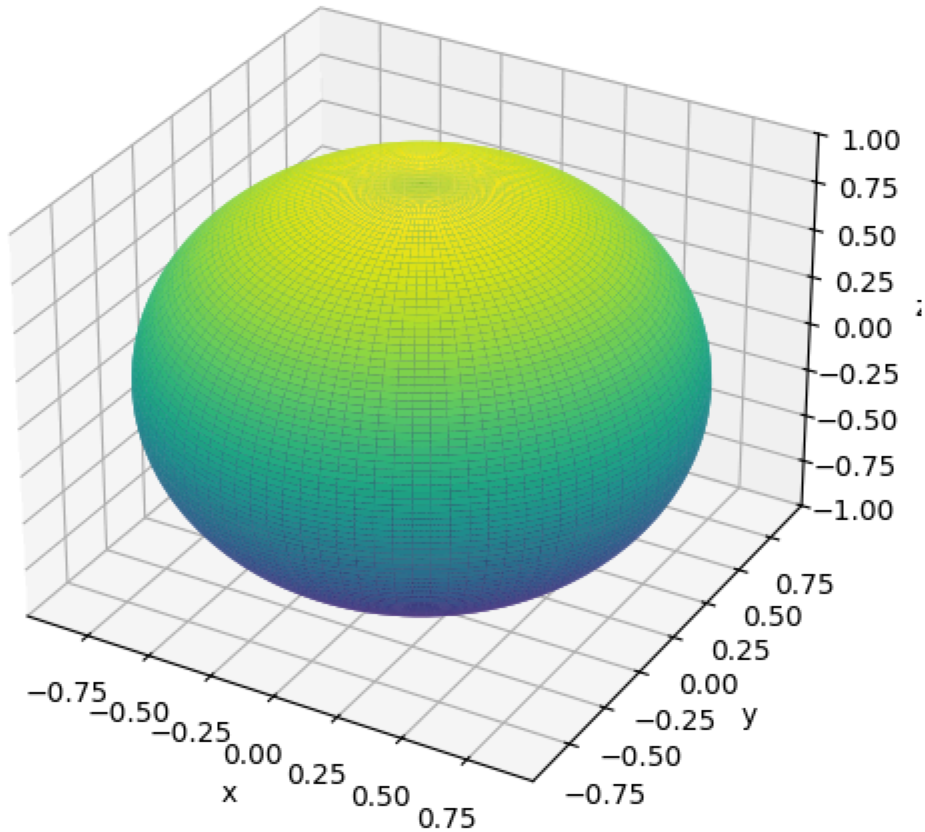
$$z(u, v) = cos(u)$$

In [98]:
```
from sympy.plotting import plot3d_parametric_surface
```

In [99]:
```
u, v = symbols('u v')

# Permukaan Bola
x = sin(u)*cos(v)
y = sin(u)*sin(v)
z = cos(u)
```

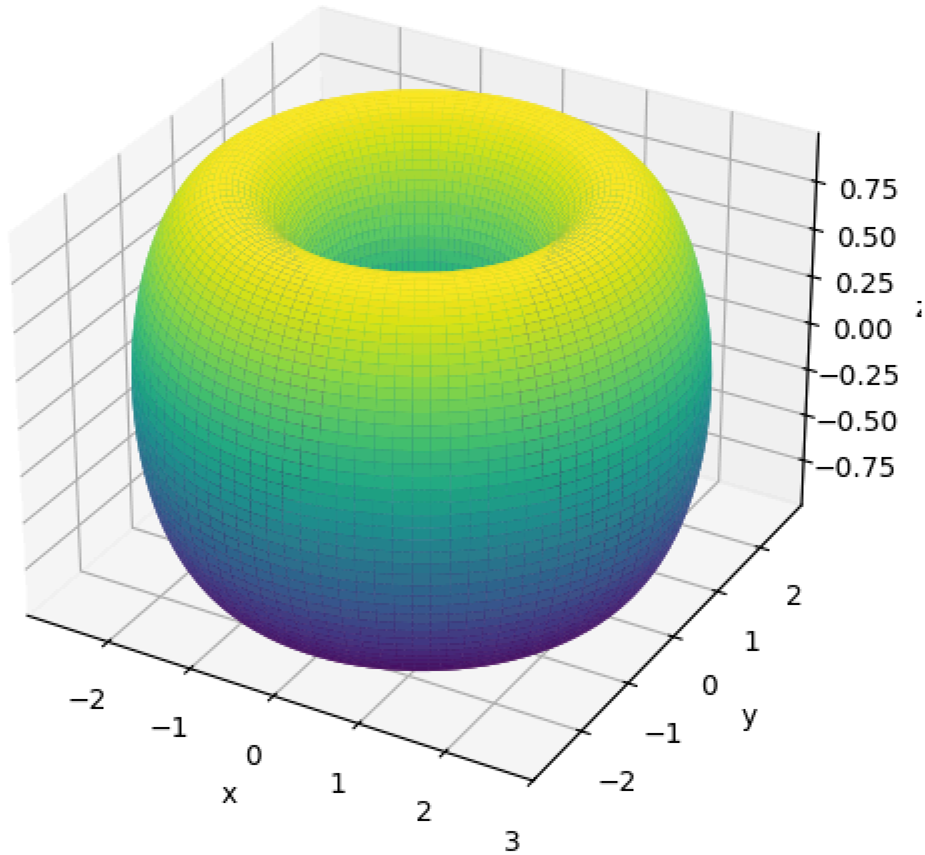
```
plot3d_parametric_surface(x, y, z, (u, 0, pi), (v, 0, 2*pi))
```



Out[99]: <sympy.plotting.backends.matplotlibbackend.matplotlib.MatplotlibBackend at 0x21e7cea6a20>

In [100]:
```
# Permukaan Torus
R = 2
r = 1
x = (R + r*cos(u))*cos(v)
y = (R + r*cos(u))*sin(v)
z = r*sin(u)

plot3d_parametric_surface(x, y, z, (u, 0, 2*pi), (v, 0, 2*pi))
```




```
Out[100]: <sympy.plotting.backends.matplotlibbackend.matplotlib.MatplotlibBackend at 0x21e7cfe
ca40>
```

```
In [ ]:
```