

# MEMBANGKITKAN SUATU MATRIKS UNIMODULAR DENGAN PYTHON

Samsul Arifin<sup>1</sup>, Indra Bayu Muktyas<sup>2</sup>

[samsul.arifin@stkipsurya.ac.id](mailto:samsul.arifin@stkipsurya.ac.id)

[indrabayu.muktyas@stkipsurya.ac.id](mailto:indrabayu.muktyas@stkipsurya.ac.id)

<sup>1,2</sup> Program Studi Pendidikan Matematika - STKIP Surya

---

## Abstract

An SPL can be represented as a multiplication of the coefficient matrix and solution vector of the SPL. Determining the solution of an SPL can use the inverse matrix method and Cramer's rule, where both can use the concept of the determinant of a matrix. If the coefficient matrix is a unimodular matrix, then all solutions of an SPL are integers. In this paper we will present a method of generating a unimodular matrix using Python so that it can be utilized on an SPL.

**Keywords:** SPL, Unimodular Matrix, Python

---

## 1. Pendahuluan

Ilmu aljabar adalah suatu cabang dalam ilmu matematika yang salah satu diantaranya mempelajari mengenai Sistem Persamaan Linier (SPL) dan matriks. Suatu persamaan linear dalam  $n$  variabel dengan bentuk  $x_1, \dots, x_n$  merupakan suatu persamaan dalam bentuk  $a_1x_1 + \dots + a_nx_n = b_1$  dengan  $a_1, \dots, a_n, b_1 \in \mathbb{R}$ . Dalam (Anton: 2004), koleksi sebanyak berhingga persamaan-persamaan linier dalam variabel  $x_1, \dots, x_n$  dinamakan SPL.

Terdapat beberapa cara untuk menentukan solusi suatu SPL, seperti dengan menggunakan metode invers matriks dan aturan Cramer. Namun kedua cara tersebut hanya bisa digunakan untuk suatu matriks bujur sangkar dan non singular. Di lain pihak, seringkali terdapat masalah mengenai dapat atau tidaknya suatu matriks diinverskan, yaitu didasarkan pada konsep determinan suatu matriks. Salah satu teorema penting yang bersangkutan menyatakan bahwa matriks  $A$  mempunyai invers jika hanya jika  $\det(A) \neq 0$  (Anton: 2004). Masalah lain muncul jika entri-entri dari invers matriks bukanlah bilangan bulat. Untuk itu diperlukan suatu matriks yang memiliki nilai determinan 1 atau -1 sehingga menghasilkan suatu matriks invers dengan semua entri adalah bilangan bulat. Matriks yang memiliki nilai determinan 1 atau -1 dinamakan matriks unimodular. Langkah-langkah dalam membangkitkan suatu matriks unimodular dapat dipelajari di (Harson: 1982).

Python adalah suatu bahasa pemrograman yang multiguna dan mudah untuk dipelajari. Python juga dapat berjalan di berbagai platform system operasi, seperti Windows (python),

Linux, Mac OS, Android (Google Play), dan lain lain. Dengan menggunakan program Python, dapat dibangkitkan suatu matriks unimodular sehingga dapat menghasilkan suatu SPL dengan solusi tunggal semua bilangan bulat. Dalam tulisan ini akan dikaji mengenai suatu matriks unimodular yang dapat dijadikan sebagai matriks koefisien dalam suatu SPL, dan diberikan metode untuk membangkitkannya dengan menggunakan program Python.

## 2. Sistem Persamaan Linier

Dalam sesi ini akan dikaji mengenai pengertian Sistem Persamaan Linier (SPL). Untuk ke depannya, istilah sistem persamaan linier cukup ditulis dengan SPL. Perhatikan bahwa bentuk umum suatu SPL dengan  $n$  persamaan dan  $n$  variabel adalah sebagai berikut:

$$\begin{cases} a_{11}x_1 + \cdots + a_{1n}x_n = b_1 \\ \vdots \quad \ddots \quad \vdots \quad \vdots \\ a_{n1}x_1 + \cdots + a_{nn}x_n = b_n \end{cases}$$

Penyelesaian suatu SPL dengan  $n$  variabel adalah bilangan-bilangan real  $x_1, \dots, x_n$  yang memenuhi semua persamaan linier dalam sistem persamaan linier tersebut. Suatu SPL dengan  $x_1, \dots, x_n$  adalah variabel dan  $a, b$  adalah konstanta riil dapat ditulis sebagai:

$$\begin{bmatrix} a_{11} & \cdots & a_{1n} \\ \vdots & \ddots & \vdots \\ a_{n1} & \cdots & a_{nn} \end{bmatrix} \begin{bmatrix} x_1 \\ \vdots \\ x_n \end{bmatrix} = \begin{bmatrix} b_1 \\ \vdots \\ b_n \end{bmatrix}$$

Mencari solusi SPL dengan  $n$  persamaan dan  $n$  variabel dapat menggunakan metode invers matriks dan aturan Cramer, di mana keduanya melibatkan konsep determinan suatu matriks. Di lain pihak, berdasarkan perbandingan nilai  $n$  (banyak persamaan) dan  $n$  (banyak variabel), salah satu jenis SPL adalah SPL *Determined*, yaitu SPL yang variabel dan persamaannya sama banyak. Dalam tulisan ini, diasumsikan bahwa semua SPL yang diberikan adalah suatu SPL *Determined*.

Kaitan antara invers suatu matriks dan nilai determinannya adalah sebagai berikut.

**Teorema 2.1. Anton** [Anton: 2004].

*Matriks  $A$  adalah suatu matriks invertible jika dan hanya jika  $\det(A) \neq 0$ .*

Menentukan invers suatu matriks dengan menggunakan adjoin dari matriks tersebut, tertuang dalam sifat berikut.

**Teorema 2.2. Anton [Anton: 2004].**

Jika  $A$  suatu matriks invertible, maka  $A^{-1} = \frac{1}{\det(A)} \text{adj}(A)$

Menentukan solusi suatu SPL dapat menggunakan invers dari matriks koefisiennya, yang dijamin oleh sifat berikut.

**Teorema 2.3. Anton [Anton: 2004].**

Suatu SPL  $n$  persamaan dan  $n$  variabel dituliskan dalam bentuk

$$A_{mn} X_{n1} = B_{n1}$$

memiliki solusi tunggal jika dan hanya jika matriks  $A$  memiliki invers. Jika  $A^{-1}$  ada maka solusi dari SPL tersebut adalah  $X_{n1} = A^{-1}_{mn} B_{n1}$

Berikut adalah contoh suatu SPL dengan empat variabel yang solusinya dapat dicari dengan menggunakan metode matriks invers.

**Contoh 2.4.**

Misalkan kita memiliki SPL empat variabel:

$$\begin{aligned} 4x_1 + 5x_2 + 43x_3 + 9x_4 &= -1 \\ x_1 + x_2 + 9x_3 &= 0 \\ 2x_1 + 7x_2 + 54x_3 + 48x_4 &= 1 \\ 2x_1 + 6x_2 + 50x_3 + 49x_4 &= 0 \end{aligned}$$

SPL dapat ditulis sebagai:

$$\begin{pmatrix} 4 & 5 & 43 & 9 \\ 1 & 1 & 9 & 0 \\ 2 & 7 & 54 & 48 \\ 2 & 6 & 50 & 49 \end{pmatrix} \begin{pmatrix} x_1 \\ x_2 \\ x_3 \\ x_4 \end{pmatrix} = \begin{pmatrix} -1 \\ 0 \\ 1 \\ 0 \end{pmatrix}$$

Dalam hal ini matriks koefisien adalah

$$A = \begin{pmatrix} 4 & 5 & 43 & 9 \\ 1 & 1 & 9 & 0 \\ 2 & 7 & 54 & 48 \\ 2 & 6 & 50 & 49 \end{pmatrix}$$

Kita memiliki

$$A^{-1} = \begin{pmatrix} 249 & -901 & -62 & 15 \\ 228 & -826 & -55 & 12 \\ -53 & 192 & 13 & -3 \\ 16 & -58 & -4 & 1 \end{pmatrix}$$

Akibatnya:

$$\begin{pmatrix} x_1 \\ x_2 \\ x_3 \\ x_4 \end{pmatrix} = \begin{pmatrix} 4 & 5 & 43 & 9 \\ 1 & 1 & 9 & 0 \\ 2 & 7 & 54 & 48 \\ 2 & 6 & 50 & 49 \end{pmatrix}^{-1} \begin{pmatrix} -1 \\ 0 \\ 1 \\ 0 \end{pmatrix} = \begin{pmatrix} 249 & -901 & -62 & 15 \\ 228 & -826 & -55 & 12 \\ -53 & 192 & 13 & -3 \\ 16 & -58 & -4 & 1 \end{pmatrix} \begin{pmatrix} -1 \\ 0 \\ 1 \\ 0 \end{pmatrix} = \begin{pmatrix} -311 \\ -283 \\ 66 \\ -20 \end{pmatrix}$$

Di lain pihak, menentukan solusi suatu SPL juga dapat dilakukan dengan menggunakan Aturan Cramer, yang tertuang dalam sifat di bawah ini.

**Teorema 2.5. Anton** [Anton: 2004].

Jika  $AX = B$  adalah suatu SPL  $n$  variabel sedemikian hingga

$$\begin{bmatrix} a_{11} & \cdots & a_{1n} \\ \vdots & \ddots & \vdots \\ a_{n1} & \cdots & a_{nn} \end{bmatrix} \begin{bmatrix} x_1 \\ \vdots \\ x_n \end{bmatrix} = \begin{bmatrix} b_1 \\ \vdots \\ b_n \end{bmatrix}$$

dan  $\det(A) \neq 0$  maka SPL tersebut memiliki suatu solusi tunggal, yaitu:

$$x_1 = \frac{\det(A_1)}{\det(A)}, x_2 = \frac{\det(A_2)}{\det(A)}, \dots, x_n = \frac{\det(A_n)}{\det(A)}$$

di mana  $A_i$  adalah matriks yang diperoleh dengan mengganti kolom ke- $i$  dari matrik  $A$  oleh

matriks kolom  $B = \begin{pmatrix} b_1 \\ \vdots \\ b_n \end{pmatrix}$

Berikut adalah contoh suatu SPL dengan empat variabel yang solusinya dapat dicari dengan menggunakan aturan Cramer. Contoh 2.6. di bawah ini akan menutup sesi ini.

**Contoh 2.6.**

Kita akan memakai aturan Cramer untuk mencari solusi SPL lima variabel berikut:

$$\begin{aligned} 2x_1 + 4x_2 + 26x_3 + 41x_4 + 38x_5 &= 1 \\ 2x_1 + 3x_2 + 17x_3 + 17x_4 + 20x_5 &= -1 \\ 3x_1 + 5x_2 + 30x_3 + 37x_4 + 39x_5 &= -1 \\ 1x_1 + 1x_2 + 5x_3 + 4x_4 + 9x_5 &= 1 \\ 4x_1 + 6x_2 + 39x_3 + 71x_4 + 81x_5 &= -1 \end{aligned}$$

SPL dapat ditulis sebagai:

$$\begin{pmatrix} 2 & 4 & 26 & 41 & 38 \\ 2 & 3 & 17 & 17 & 20 \\ 3 & 5 & 30 & 37 & 39 \\ 1 & 1 & 5 & 4 & 9 \\ 4 & 6 & 39 & 71 & 81 \end{pmatrix} \begin{pmatrix} x_1 \\ x_2 \\ x_3 \\ x_4 \\ x_5 \end{pmatrix} = \begin{pmatrix} 1 \\ -1 \\ -1 \\ 1 \\ -1 \end{pmatrix}$$

dengan

$$A_1 = \begin{pmatrix} 1 & 4 & 26 & 41 & 38 \\ -1 & 3 & 17 & 17 & 20 \\ -1 & 5 & 30 & 37 & 39 \\ 1 & 1 & 5 & 4 & 9 \\ -1 & 6 & 39 & 71 & 81 \end{pmatrix}, A_2 = \begin{pmatrix} 2 & 1 & 26 & 41 & 38 \\ 2 & -1 & 17 & 17 & 20 \\ 3 & -1 & 30 & 37 & 39 \\ 1 & 1 & 5 & 4 & 9 \\ 4 & -1 & 39 & 71 & 81 \end{pmatrix}, A_3 = \begin{pmatrix} 2 & 4 & 1 & 41 & 38 \\ 2 & 3 & -1 & 17 & 20 \\ 3 & 5 & -1 & 37 & 39 \\ 1 & 1 & 1 & 4 & 9 \\ 4 & 6 & -1 & 71 & 81 \end{pmatrix},$$

$$A_4 = \begin{pmatrix} 2 & 4 & 26 & 1 & 38 \\ 2 & 3 & 17 & -1 & 20 \\ 3 & 5 & 30 & -1 & 39 \\ 1 & 1 & 5 & 1 & 9 \\ 4 & 6 & 39 & -1 & 81 \end{pmatrix}, A_5 = \begin{pmatrix} 2 & 4 & 26 & 41 & 1 \\ 2 & 3 & 17 & 17 & -1 \\ 3 & 5 & 30 & 37 & -1 \\ 1 & 1 & 5 & 4 & 1 \\ 4 & 6 & 39 & 71 & -1 \end{pmatrix},$$

Solusi dari SPL dapat dicari sebagai berikut:

$$x_1 = \frac{\det(A_1)}{\det(A)} = \frac{-252}{-1} = 252 \qquad x_4 = \frac{\det(A_4)}{\det(A)} = \frac{-1}{-1} = 1$$

$$x_2 = \frac{\det(A_2)}{\det(A)} = \frac{571}{-1} = -571 \qquad x_5 = \frac{\det(A_5)}{\det(A)} = \frac{11}{-1} = -11$$

$$x_3 = \frac{\det(A_3)}{\det(A)} = \frac{-83}{-1} = 83$$

### 3. Matrik Unimodular

Dalam sesi ini akan dikaji mengenai pengertian matriks unimodular dan bagaimana cara membangkitkannya. Misalkan diberikan matriks sebagai berikut:

$$A_{n \times n} = \begin{pmatrix} a_{11} & \cdots & a_{1n} \\ \vdots & \ddots & \vdots \\ a_{n1} & \cdots & a_{nn} \end{pmatrix}$$

Dalam Harson: 1982, disebutkan bahwa sebuah matriks  $A_{n \times n}$  dengan setiap entrinya berupa bilangan bulat disebut unimodular jika  $\det(A) = -1$  atau  $\det(A) = 1$ . Contoh dari matriks unimodular adalah matriks identitas, matriks segitiga atas atau matriks segitiga bawah yang entri-entri diagonalnya 1 atau -1. Hal ini diperkuat oleh teorema berikut ini.

**Teorema 3.1. Anton** [Anton: 2004].

Jika  $A_{n \times n}$  adalah matriks segitiga maka  $\det(A) = a_{11}a_{22} \dots a_{nn}$ .

Lema berikut menjelaskan langkah-langkah dalam membangkitkan suatu matriks unimodular. Perhatikan bahwa Lema 3.2. di bawah ini yang akan digunakan sebagai rujukan dalam membuat program untuk membangkitkan suatu matriks unimodular dengan menggunakan Python.

**Lema 3.2. Hanson** [Hanson: 1982]

Matriks unimodular  $A_{n \times n}$  dapat dibangun dengan cara berikut ini:

1. Pertama buat matriks diagonal dengan entri diagonalnya  $a_{ii} = 1$  atau  $a_{ii} = -1$ .
2. Kedua, isikan bilangan bulat sembarang pada setiap entri  $a_{ij}$  dengan  $i < j$ . Dari sini telah terbentuk suatu matriks segitiga atas yang determinannya 1 atau -1. Ini merupakan matriks unimodular.
3. Ketiga, agar menjadi matriks yang utuh, gunakan operasi baris elementer atau operasi kolom elementer secara menurun dari baris/kolom terakhir ke baris/kolom awal.

Berikut adalah contoh-contoh dari matriks unimodular.

**Contoh 3.3.**

Matriks-matriks berikut adalah matriks unimodular.

$$A = \begin{pmatrix} 1 & -2 & 3 \\ 0 & -1 & 4 \\ 0 & 0 & 1 \end{pmatrix}, B = \begin{pmatrix} 1 & -2 & 3 \\ 2 & -5 & 10 \\ 0 & 0 & 1 \end{pmatrix}, C = \begin{pmatrix} 1 & 0 & 0 & 0 \\ 3 & -1 & 0 & 0 \\ 0 & 1 & -1 & 0 \\ 2 & 4 & 3 & 1 \end{pmatrix}, E = \begin{pmatrix} 5 & 8 & 6 & 2 \\ 3 & -1 & 0 & 0 \\ 0 & 1 & -1 & 0 \\ 2 & 4 & 3 & 1 \end{pmatrix}$$

#### 4. Python

Dalam sesi ini akan dikaji mengenai pemrograman dalam membangkitkan suatu matriks unimodular berukuran nxn dengan menggunakan Python 2.7.14, yang merupakan hasil utama dari tulisan ini. Hal-hal yang menjadi dasar dalam pembuatan program adalah Lemma 3.2. di atas. Namun sebelumnya perlu diperhatikan bahwa program Python yang dipakai harus dilengkapi dengan plugin “numpy”. Langkah-langkah untuk menginstall numpy dapat dipelajari di (Stackoverflow). Berikut adalah tampilan pemrograman yang digunakan dalam tulisan ini.

```
print "====="
print "Membangkitkan Matriks Unimodular Untuk Suatu SPL Determined"
print "====="
import numpy as np
import random

def r_ij(m, baris_i, baris_j, r):
```

```
        return m[baris_i] + r*m[baris_j]

def tukar(m, baris_i, baris_j):
    m[baris_i] = m[baris_i] + m[baris_j]
    m[baris_j] = m[baris_i] - m[baris_j]
    m[baris_i] = m[baris_i] - m[baris_j]

n = input('Masukkan nilai n untuk menentukan ukuran matriks (n x n):')
a = np.eye(n)

for i in range(n):
    for j in range(i+1,n):
        a[i,j] = random.randint(0,9)

print "Membangkitkan matriks segitiga asal A berukuran ",n,"x",n,":"
print a

for j in range(n-1,-1,-1):
    for i in range(j+1,n):
        a[i] = r_ij(a, i, j, random.randint(2,5))

tukar(a,0,random.randint(1,n-1))
print "-----"
print "Matriks unimodular U berukuran ",n,"x",n,"yang dihasilkan:"
print a
print "dengan det(U):", np.linalg.det(a)
print "-----"
```

Berikut adalah tampilan program menggunakan Python dari program di atas untuk versi OS Windows.

## Membangkitkan Suatu Matriks Unimodular Dengan Python

Samsul Arifin, Indra Bayu Muktyas.

```
print "-----"
print "Membangkitkan Matriks Unimodular Untuk Suatu SPL Determined"
print "-----"
import numpy as np
import random

def r_ij(m, baris_i, baris_j, r):
    return m[baris_i] + r*m[baris_j]

def tukar(m, baris_i, baris_j):
    m[baris_i] = m[baris_i] + m[baris_j]
    m[baris_j] = m[baris_i] - m[baris_j]
    m[baris_i] = m[baris_i] - m[baris_j]

n = input('Masukkan nilai n untuk menentukan ukuran matriks (n x n):')
a = np.eye(n)

for i in range(n):
    for j in range(i+1,n):
        a[i,j] = random.randint(0,9)

print "Membangkitkan matriks segitiga asal A berukuran ",n,"x",n,": "
print a

for j in range(n-1,-1,-1):
    for i in range(j+1,n):
        a[i] = r_ij(a, i, j, random.randint(2,5))

tukar(a,0,random.randint(1,n-1))
print "-----"
print "Matriks unimodular U berukuran ",n,"x",n,"yang dihasilkan:"
print a
print "dengan det(U):", np.linalg.det(a)
print "-----"
```

### 5. Hasil dan Pembahasan

Dalam sesi ini akan dikaji mengenai output dari program Python yang telah dibuat di atas. Beberapa output dari kode program yang digunakan adalah sebagai berikut, yaitu matriks-matriks unimodular dengan beberapa ukuran.

a) Berikut adalah contoh matriks unimodular berukuran 4x4.

```
=====
Membangkitkan Matriks Unimodular Untuk Suatu SPL Determined
=====
Masukkan nilai n untuk menentukan ukuran matriks (n x n):4
Membangkitkan matriks segitiga asal A berukuran 4 x 4 :
[[1. 2. 2. 5.]
 [0. 1. 8. 9.]
 [0. 0. 1. 8.]
 [0. 0. 0. 1.]]
-----
Matriks unimodular U berukuran 4 x 4 yang dihasilkan:
[[ 2.  6. 21. 36.]
 [ 3.  7. 14. 24.]
 [ 1.  2.  2.  5.]
 [ 2.  6. 23. 53.]]
dengan det(U): -0.9999999999998446
-----
>>>
```

b) Berikut adalah contoh matriks unimodular berukuran 5x5.

```
=====
Membangkitkan Matriks Unimodular Untuk Suatu SPL Determined
=====
Masukkan nilai n untuk menentukan ukuran matriks (n x n):5
Membangkitkan matriks segitiga asal A berukuran 5 x 5 :
[[1. 2. 7. 8. 3.]
 [0. 1. 7. 5. 1.]
 [0. 0. 1. 6. 6.]
 [0. 0. 0. 1. 3.]
 [0. 0. 0. 0. 1.]]
-----
Matriks unimodular U berukuran 5 x 5 yang dihasilkan:
[[ 3. 11. 61. 83. 57.]
 [ 5. 11. 42. 45. 16.]
 [ 3. 9. 43. 45. 18.]
 [ 4. 12. 59. 71. 37.]
 [ 1. 2. 7. 8. 3.]]
dengan det(U): -1.0000000000010831
-----
>>>
```

c) Berikut adalah contoh matriks unimodular berukuran 9x9.

```
=====
Membangkitkan Matriks Unimodular Untuk Suatu SPL Determined
=====
Masukkan nilai n untuk menentukan ukuran matriks (n x n):9
Membangkitkan matriks segitiga asal A berukuran 9 x 9 :
[[1. 7. 9. 8. 0. 3. 6. 7. 6.]
 [0. 1. 4. 0. 6. 4. 5. 7. 4.]
 [0. 0. 1. 0. 8. 3. 6. 8. 4.]
 [0. 0. 0. 1. 5. 9. 9. 5. 8.]
 [0. 0. 0. 0. 1. 9. 6. 7. 1.]
 [0. 0. 0. 0. 0. 1. 3. 2. 7.]
 [0. 0. 0. 0. 0. 0. 1. 0. 6.]
 [0. 0. 0. 0. 0. 0. 0. 1. 9.]
 [0. 0. 0. 0. 0. 0. 0. 0. 1.]]
-----
Matriks unimodular U berukuran 9 x 9 yang dihasilkan:
[[ 4. 30. 48. 37. 71. 96. 118. 115. 97.]
 [ 5. 36. 49. 40. 6. 19. 35. 42. 34.]
 [ 5. 40. 66. 40. 38. 38. 61. 78. 54.]
 [ 2. 19. 41. 17. 59. 44. 64. 78. 52.]
 [ 2. 19. 42. 20. 83. 83. 103. 108. 81.]
 [ 1. 7. 9. 8. 0. 3. 6. 7. 6.]
 [ 5. 40. 68. 44. 77. 110. 137. 141. 124.]
 [ 5. 39. 63. 45. 69. 121. 142. 139. 140.]
 [ 2. 18. 39. 18. 79. 104. 127. 140. 173.]]
dengan det(U): -1.0000000000014806
-----
>>>
```

## 6. Kesimpulan

Kesimpulan dari tulisan ini adalah sebagai berikut:

- a) Jika suatu SPL *Determined* matriks koefisiennya adalah suatu matriks unimodular, maka semua solusi tunggal yang dihasilkan adalah bilangan bulat.

- b) Program Python dapat digunakan untuk membangkitkan suatu matriks unimodular dengan mudah.
- c) Dengan menggunakan suatu matriks unimodular, dapat dibentuk suatu SPL yang pasti memiliki solusi dan mudah dikerjakan karena tidak perlu melibatkan bilangan pecahan.

## **7. Referensi**

Anton, H. dan Rorres, C. 2004. *Aljabar Linear Elementer Versi Aplikasi Edisi Kedelapan*. Jakarta: Penerbit Erlangga

Google, Tersedia di <https://play.google.com/store/apps/details?id=org.qpython.qpy&hl=en> , Terakhir diakses 27 April 2018.

Hanson, R. 1982. "Integer Matrices Whose Inverses Contain Only Integers". *The Two-Year College Mathematics Journal*, Vol. 13, No. 1 (Jan., 1982), pp. 18-21.

Python, Tersedia di <https://www.python.org/>, Terakhir diakses 28 April 2018.

Stackoverflow, Installing numpy for Python 2.7 while also having Python 3.4 installed, Tersedia di <https://stackoverflow.com/questions/25900818/installing-numpy-for-python-2-7-while-also-having-python-3-4-installed/>, Terakhir diakses 29 April 2018.