NOV. 20, 2019 • JAKARTA

BCA STOCK Analysis with Makrov Chain

Indira Insiyah Nastiti / 1706986006
Nadhila Fakhira / 1706043033
Amanda N. / 1706985893

Jakarta • Nov. 22, 2019

Latar Belakang

Seseorang harus dapat memperkirakan investasi yang akan dilakukan menguntungkan atau tidak dengan tepat. Hal tersebut dikarenakan dalam penanaman modal atau pembelian saham sebuah perusahaan tidak mudah karena bergantung dari kecenderungan pergerakan nilai saham yang akan dibeli. Apabila nilai saham mengalami kenaikan maka akan menguntungkan begitu juga sebaliknya.

Oleh karena itu untuk mengatasi permasalahan yang sering terjadi yaitu ketika seorang calon investor salah dalam mengambil keputusan karena sulitnya memprediksi pergerakan nilai saham, maka digunakan model stokastik untuk memprediksi pergerakan nilai saham.

Jakarta • Nov. 22, 2019

FORECASTING WITH MAKROV

Jakarta • Nov. 22, 2019

FORECASTING WITH MAKROV

Kasus dan Sumber

Presentasi ini melakukan penggambaran dalam memprediksi pergerakan nilai saham untuk mendapatkan gambaran umum kenaikan maupun penurunan nilai saham untuk membantu seorang investor dalam mengambil keputusan apakah investasi yang akan dilakukan menguntungkan atau merugikan.Analisis model pergerakan saham Bank Central Asia dan simulasinya dengan metode Markov Chain dengan data saham Bank Central Asia berdasarkan sample data dari 5 Januari 2012 – 4 Januari 2013 (1 Tahun) dengan periode perhari diambil pada tanggal 4 Januari 2013. Sample data tersebut dibagi menjadi 4 kelas yaitu Open, High, Low, dan Close.

FORECASTING WITH MAKROV

Date	Open	High	Low	Close	Volume	Adj Close
1/4/2013	9150	9150	9150	9150	0	9150
1/3/2013	9150	9250	9050	9150	8406000	9150
1/2/2013	9100	9250	9000	9100	3454500	9100
1/1/2013	9200	9200	9200	9200	0	9200
12/31/2012	9200	9200	9200	9200	0	9200
12/28/2012	9200	9200	9200	9200	0	9200
12/25/2012	8900	8900	8900	8900	0	8900
12/24/2012	8900	8900	8900	8900	0	8900
12/21/2012	8900	9400	8800	8900	15174500	8900
12/20/2012	9200	9200	8900	9200	12844000	9200
12/19/2012	9050	9250	9000	9050	11117500	9050
12/18/2012	9150	9350	9050	9150	9939500	9150
12/17/2012	9200	9300	9050	9200	9498500	9200
.....	.....	.....	.....	.....	.....	.....
1/13/2012	8150	8150	8150	8150	0	8049.76
1/12/2012	8150	8200	8050	8150	12628500	8049.76
1/11/2012	8100	8250	8050	8100	14614000	8000.38
1/10/2012	8150	8200	8100	8150	9975000	8049.76
1/9/2012	8150	8150	7900	8150	9302000	8049.76
1/6/2012	8000	8050	7950	8000	5438000	7901.61
1/5/2012	8050	8150	8050	8050	11439500	7950.99

Metode Penelitian

Metode yang digunakan dalam paper adalah Rantai Markov. Data diperoleh dari saham BCA tahun 2012 hingga 2013.

Rantai Markov merupakan sebuah proses Markov dengan populasi yang diskrit yang berada pada suatu ruang keadaan yang diskrit. Analisis Rantai Markov merupakan Teknik probabilitas yang menganalisis pergerakan probabilitas dari satu kondisi ke kondisi lainnya sehingga memberikan informasi probabilitas mengenai situasi keputusan yang dapat membantu pengambilan keputusan. Dengan demikian, menghasilkan informasi probabilitas di masa mendatang.

Close	Moving Average j=1	Difference of Price	State
9150	9150	0	Tetap
9150	9100	150	Naik
9100	9200	-100	Turun
9200	9200	0	Tetap
9200	9200	0	Tetap
9200	8900	300	Naik Drastis
8900	8900	0	Tetap
8900	8900	0	Tetap
8900	9200	-300	Turun Drastis
9200	9050	150	Naik
....	....	....	...
8150	8150	0	Tetap
8150	8100	50	Naik
8100	8150	-50	Turun
8150	8150	0	Tetap
8150	8000	150	Naik
8000	8050	-50	Turun
8050	0	0	Tetap

Penyelesaian

Menentukkan Moving Avarage dan persebaran State.

PENYELESAIAN

Matriks Transisi						
	Naik Drastis	Naik	Tetap	Turun	Turun Drastis	Jumlah
Naik Drastis	0	2	3	1	1	7
Naik	2	26	30	31	4	93
Tetap	2	23	19	20	4	68
Turun	2	36	15	25	0	78
Turun Drastis	1	6	2	0	0	9

Matrix Transisi

Matrix Probabilitas

Matriks Peluang Transisi /P^1					
	Naik Drastis	Naik	Tetap	Turun	Turun Drastis
Naik Drastis	0	0.285714286	0.428571429	0.142857143	0.142857143
Naik	0.021505376	0.279569892	0.322580645	0.333333333	0.043010753
Tetap	0.029411765	0.338235294	0.279411765	0.294117647	0.058823529
Turun	0.025641026	0.461538462	0.192307692	0.320512821	0
Turun Drastis	PHI	0.666666667	0.222222222	0	0

PENYELESAIAN

P^2					
	Naik Drastis	Naik	Tetap	Turun	Turun Drastis
Naik Drastis	0.038285455	0.386007257	0.271132357	0.267076061	0.037498871
Naik	0.028825903	0.375931866	0.263193763	0.297976428	0.03407204
Tetap	0.029569296	0.372432979	0.269417117	0.293395186	0.035185421
Turun	0.023799944	0.349332508	0.275242498	0.316798712	0.034826337
Turun Drastis	0.020872865	0.293289359	0.324764314	0.303454715	0.057618746

P^3					
	Naik Drastis	Naik	Tetap	Turun	Turun Drastis
Naik Drastis	0.027290358	0.358826362	0.276377956	0.299484549	0.038020776
Naik	0.027251753	0.362598899	0.27203646	0.302343804	0.035769085
Tetap	0.027365789	0.362565907	0.272331672	0.301645759	0.036090874
Turun	0.027600532	0.366991912	0.268455889	0.302335886	0.034615781
Turun Drastis	0.030042163	0.376273824	0.265458796	0.293525	0.034700217

P^4					
	Naik Drastis	Naik	Tetap	Turun	Turun Drastis
Naik Drastis	0.027749081	0.365165893	0.270711803	0.300783681	0.035589541
Naik	0.027525646	0.364559633	0.270748621	0.301675196	0.035490904
Tetap	0.027551475	0.364575193	0.270806606	0.301543584	0.035523141
Turun	0.027386461	0.363903839	0.271056933	0.302133712	0.035519054
Turun Drastis	0.027281384	0.362172397	0.272584451	0.301870988	0.036090781

P^5					
	Naik Drastis	Naik	Tetap	Turun	Turun Drastis
Naik Drastis	0.027481937	0.364131583	0.271079778	0.301712263	0.035594438
Naik	0.027481882	0.364256416	0.270948265	0.301774828	0.035538609
Tetap	0.027484129	0.364248506	0.27096241	0.301758576	0.035546379
Turun	0.027491732	0.36436798	0.270857646	0.301773987	0.035508655
Turun Drastis	0.027556212	0.364630453	0.270757416	0.301546892	0.035509027

Matrix Prediksi Periode

Akurasi

PREDIKSI

Jakarta • Nov. 22, 2019

∏0 = [0.027488106 0.364297797 0.270921586 0.301760352 0.035532159]

Akurasi Prediksi Saham BCA 1 Hari Kemudian

Distribusi Stasioner = ∏0 * P^2

Didapat

Naik Drastis	Naik	Tetap	Turun	Turun Drastis
2.75%	36.43%	27.09%	30.18%	3.55%

Akurasi Prediksi Saham BCA 2 Hari Kemudian

Distribusi Stasioner = ∏0 * P^2

Didapat

Naik Drastis	Naik	Tetap	Turun	Turun Drastis
2.75%	36.43%	27.09%	30.18%	3.55%

Akurasi Prediksi Saham BCA 3 Hari Kemudian

Distribusi Stasioner = ∏0 * P^3

Didapat

Naik Drastis	Naik	Tetap	Turun	Turun Drastis
2.75%	36.43%	27.09%	30.18%	3.55%

Akurasi

PREDIKSI

Jakarta • Nov. 22, 2019

Akurasi Prediksi Saham BCA 4 Hari Kemudian

Distribusi Stasioner = ∏0 * P^4

Didapat

Naik Drastis	Naik	Tetap	Turun	Turun Drastis
2.75%	36.43%	27.09%	30.18%	3.55%

Akurasi Prediksi Saham BCA 5 Hari Kemudian

Distribusi Stasioner = ∏0 * P^5

Didapat

Naik Drastis	Naik	Tetap	Turun	Turun Drastis
2.75%	36.43%	27.09%	30.18%	3.55%

???

KESIMPULAN

Analisis rantai markov merupakan suatu teknik probabilitas yang menganalisis pergerakan probabilitas dari satu kondisi ke kondisi lainnya dan menghasilkan suatu probabilitas mengenai situasi keputusan yang dapat membantu pengambilan keputusan. Pengaplikasian analisis rantai markov pada pergerakan harga saham akan membantu para investor dalam mengambil keputusan yang terbaik dalam hal jual beli saham. Pada
kasus ini prediksi pergerakan harga saham dikelompokkan menjadi 5 kondisi; naik, naik drastis, tetap, turun dan turun drastis.

Moving Average	Π0	Nilai P^n				
		Hari Ke 1 (P^1)	Hari Ke 2 (P^2)	Hari Ke 3 (P^3)	Hari Ke 4 (P^4)	Hari Ke 5 (P^5)
Interval 1	[0.027488106 0.364297797 0.270921586 0.301760352 0.035532159]	Akurasi	Akurasi	Akurasi	Akurasi	Akurasi

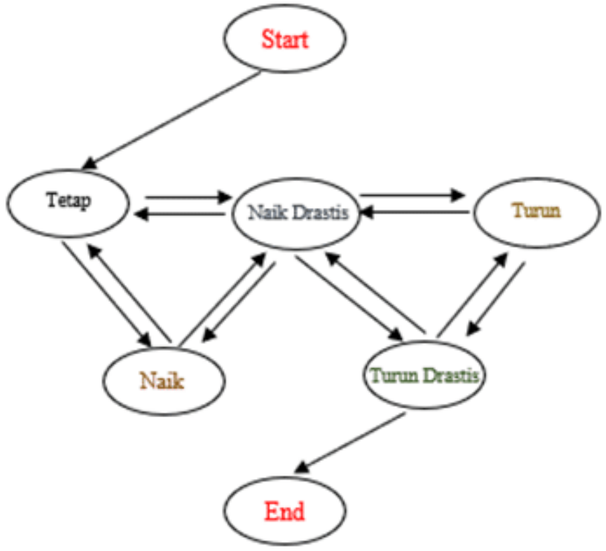
BCA 2012-2013

BCA STOCK ANALYSIS WITH
MAKROV CHAIN

Implementasi BCA Stock 2018-2019

Jakarta • Nov. 22, 2019

Keys		Frequencies
Naik Drastis	51	
Naik	54	
Tetap	34	
Turun	67	
Turun Drastis	55	



Relational Dataset

BCA STOCK

Jakarta • Nov. 22, 2019

```
import pandas as pd
import numpy as np

bbca = pd.read_csv (r'https://raw.githubusercontent.com/IndiraInsiyah/probstok/ma
bbca = bbca.fillna(0)
print (bbca)
```

```
           Date     Open     High      Low    Close     Adj Close      Volume
0    2018-11-22  24700.0  25200.0  24700.0  25100.0  24791.470703  17454000.0
1    2018-11-23  25300.0  25450.0  24875.0  25100.0  24791.470703  11492000.0
2    2018-11-26  24800.0  25400.0  24800.0  25225.0  24914.933594  12867300.0
3    2018-11-27  25200.0  25500.0  25000.0  25500.0  25186.554688  14797800.0
4    2018-11-28  25625.0  25625.0  25400.0  25450.0  25137.169922  18137500.0
..          ...      ...      ...      ...      ...           ...         ...
256  2019-11-15  31400.0  31450.0  31275.0  31375.0  31375.000000   9427600.0
257  2019-11-18      0.0      0.0      0.0      0.0      0.000000         0.0
258  2019-11-19  31650.0  31750.0  31500.0  31575.0  31575.000000  12023000.0
259  2019-11-20  31275.0  31750.0  31275.0  31750.0  31750.000000   8645900.0
260  2019-11-21  31750.0  31750.0  31500.0  31500.0  31500.000000  11504700.0

[261 rows x 7 columns]
```

How to Combat These Problems

Import file cvs ke platform online pengolahan pemrograman Phyton.

```
[3]  bbcaN = bbca.set_index('Date').diff()
     bbcaN = bbcaN.fillna(0)
     print (bbcaN)
```

```
                Open     High      Low    Close     Adj Close      Volume
Date
2018-11-22       0.0      0.0      0.0      0.0      0.000000         0.0
2018-11-23     600.0    250.0    175.0      0.0      0.000000  -5962000.0
2018-11-26    -500.0    -50.0    -75.0    125.0    123.462891   1375300.0
2018-11-27     400.0    100.0    200.0    275.0    271.621094   1930500.0
2018-11-28     425.0    125.0    400.0    -50.0    -49.384766   3339700.0
...              ...      ...      ...      ...           ...         ...
2019-11-15      75.0     50.0    225.0     25.0     25.000000   1855200.0
2019-11-18  -31400.0 -31450.0 -31275.0 -31375.0 -31375.000000  -9427600.0
2019-11-19   31650.0  31750.0  31500.0  31575.0  31575.000000  12023000.0
2019-11-20    -375.0      0.0   -225.0    175.0    175.000000  -3377100.0
2019-11-21     475.0      0.0    225.0   -250.0   -250.000000   2858800.0

[261 rows x 6 columns]
```

How to Combat These Problems

Melakukan pendefinisian data Close yang akan diolah, dengan melakukan pengurangan nilai saat ini dengan nilai sebelumnya - Moving Average

```
bbcaC = bbcaN['Close']
naik=0
naik_d=0
tetap=0
turun=0
turun_d=0

state = list(range(0,len(bbcaC)))
for i in range(1,len (bbcaC)) :
            if bbcaC[i] > 200:
              state[i]="Naik Drastis"
            elif 0< bbcaC[i] and bbcaC[i]<200 :
              state[i]="Naik"
            elif bbcaC[i] == 0 :
              state[i]="Tetap"
            elif -200<bbcaC[i] and bbcaC[i]<0 :
              state[i]="Turun"
            else :
              state[i]="Turun Drastis"

print("STATE FREQUENTION'S")
print("\nBanyak Nilai Drastis :", state.count ("Naik Drastis"))
print("Banyak Nilai :", state.count ("Naik"))
print("Banyak Tetap :", state.count ("Tetap"))
print("Banyak Turun :", state.count ("Turun"))
print("Banyak Turun Drastis :", state.count ("Turun Drastis"))
```

```
STATE FREQUENTION'S

Banyak Nilai Drastis : 51
Banyak Nilai : 54
Banyak Tetap : 33
Banyak Turun : 67
Banyak Turun Drastis : 55
```

How to Combat These Problems

How to Combat These Problems

```
transition = list(range(0,25))

for i in range(1, len(state)):
  if state[i-1] != state[i]:
    if state[i-1] == 'Naik Drastis':
      if state[i] == 'Naik':
        transition[1] += 1
      elif state[i] == 'Tetap':
        transition[2] += 1
      elif state[i] == 'Turun':
        transition[3] += 1
      elif state[i] == 'Turun Drastis':
        transition[4] += 1
    elif state[i-1] == 'Naik':
      if state[i] == 'Naik Drastis':
        transition[5] += 1
      elif state[i] == 'Tetap':
        transition[7] += 1
      elif state[i] == 'Turun':
        transition[8] += 1
      elif state[i] == 'Turun Drastis':
        transition[9] += 1
    elif state[i-1] == 'Tetap':
      if state[i] == 'Naik Drastis':
        transition[10] += 1
      elif state[i] == 'Naik':
        transition[11] += 1
      elif state[i] == 'Turun':
        transition[13] += 1
      elif state[i] == 'Turun Drastis':
        transition[14] += 1
    elif state[i-1] == 'Turun':
      if state[i] == 'Naik Drastis':
        transition[15] += 1
      elif state[i] == 'Naik':
        transition[16] += 1
      elif state[i] == 'Tetap':
        transition[17] += 1
      elif state[i] == 'Turun Drastis':
        transition[19] += 1
    elif state[i-1] == 'Turun Drastis':
      if state[i] == 'Naik Drastis':
        transition[20] += 1
      elif state[i] == 'Naik':
        transition[21] += 1
      elif state[i] == 'Tetap':
        transition[22] += 1
      elif state[i] == 'Turun':
        transition[23] += 1
  else:
    if state [i-1] == 'Naik Drastis':
      if state[i] == 'Naik Drastis':
        transition[0] += 1
    elif state [i-1] == 'Naik':
      if state[i] == 'Naik':
        transition[6] += 1
    elif state [i-1] == 'Tetap':
      if state[i] == 'Tetap':
        transition[12] += 1
    elif state [i-1] == 'Turun':
      if state[i] == 'Turun':
        transition[18] += 1
    elif state [i-1] == 'Turun Drastis':
      if state[i] == 'Turun Drastis':
        transition[24] += 1
```

[redacted]

[redacted]

[redacted]

Jakarta • Nov. 22, 2019

```
sum_ND = transition[0]+transition[1]+transition[2]+transition[3]+transition[4]
sum_N = transition[5]+transition[6]+transition[7]+transition[8]+transition[9]
sum_TE = transition[10]+transition[11]+transition[12]+transition[13]+transition[14]
sum_T = transition[15]+transition[16]+transition[17]+transition[18]+transition[19]
sum_TD = transition[20]+transition[21]+transition[22]+transition[23]+transition[24]
ND_ND = transition[0]
ND_N = transition[1]
ND_TE = transition[2]
ND_T = transition[3]
ND_TD = transition[4]

mattrans = [
            [transition[0], transition[1], transition[2], transition[3], transition[4], sum_ND],
            [transition[5], transition[6], transition[7], transition[8], transition[9], sum_N],
            [transition[10], transition[11], transition[12], transition[13], transition[14], sum_TE],
            [transition[15], transition[16], transition[17], transition[18], transition[19], sum_T],
            [transition[20], transition[21], transition[22], transition[23], transition[24], sum_TD]
          ]

print("COUNTING THE TRANSITIONS\n")
for i in mattrans :
    print (i)
```

```
COUNTING THE TRANSITIONS

[10, 11, 9, 19, 12, 61]
[13, 15, 15, 25, 21, 89]
[16, 15, 22, 18, 22, 93]
[26, 39, 23, 31, 33, 152]
[36, 29, 23, 39, 37, 164]
```

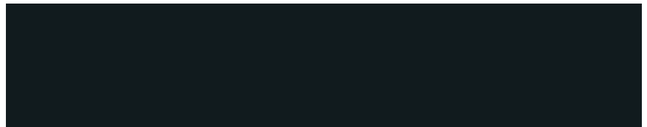
How to Combat These Problems

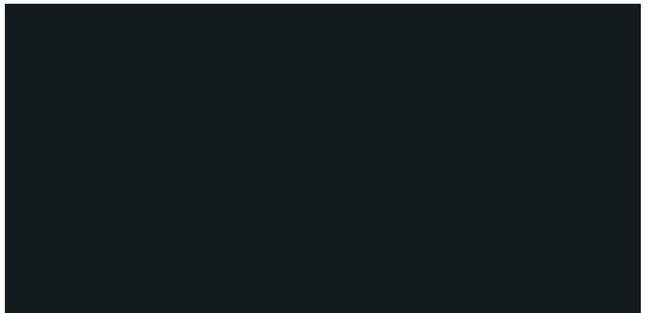
How to Combat These Problems

```
NDND=('%.3f'%(transition[0]/sum_ND))
NDN = ('%.3f'%(transition[1]/sum_ND))
NDTE = ('%.3f'%(transition[2]/sum_ND))
NDT = ('%.3f'%(transition[3]/sum_ND))
NDTD = ('%.3f'%(transition[4]/sum_ND))

NND=('%.3f'%(transition[5]/sum_ND))
NN = ('%.3f'%(transition[6]/sum_ND))
NTE = ('%.3f'%(transition[7]/sum_ND))
NT = ('%.3f'%(transition[8]/sum_ND))
NTD = ('%.3f'%(transition[9]/sum_ND))

TEND=('%.3f'%(transition[10]/sum_ND))
TEN = ('%.3f'%(transition[11]/sum_ND))
TETE = ('%.3f'%(transition[12]/sum_ND))
TET = ('%.3f'%(transition[13]/sum_ND))
TETD = ('%.3f'%(transition[14]/sum_ND))
```

```
TND=('%.3f'%(transition[15]/sum_ND))
TN = ('%.3f'%(transition[16]/sum_ND))
TTE = ('%.3f'%(transition[17]/sum_ND))
TT = ('%.3f'%(transition[18]/sum_ND))
TTD = ('%.3f'%(transition[19]/sum_ND))

TDND=('%.3f'%(transition[20]/sum_ND))
TDN = ('%.3f'%(transition[21]/sum_ND))
TDTE = ('%.3f'%(transition[22]/sum_ND))
TDT = ('%.3f'%(transition[23]/sum_ND))
TDTD = ('%.3f'%(transition[24]/sum_ND))



matpro = [
    [NDND, NDN, NDTE, NDT, NDTD],
    [NND, NN, NTE, NT, NTD],
    [TEND, TEN, TETE, TET, TETD],
    [TND, TN, TTE, TT, TTD],
    [TDND, TDN, TDTE, TDT, TDTD]
]

print("PROBABILITY TRANSITIONS MATRIX\n")
for row in matpro:
 print (' '.join(row))
```

```
PROBABILITY TRANSITIONS MATRIX

0.164 0.180 0.148 0.311 0.197
0.213 0.246 0.246 0.410 0.344
0.262 0.246 0.361 0.295 0.361
0.426 0.639 0.377 0.508 0.541
0.590 0.475 0.377 0.639 0.607
```





Jakarta • Nov. 22, 2019

How to Combat These Problems

Jakarta • Nov. 22, 2019

```
mat1 = np.array(([
    [0.164, 0.180, 0.148, 0.311, 0.197],
    [0.213, 0.246, 0.246, 0.410, 0.344],
    [0.262, 0.246, 0.361, 0.295, 0.361],
    [0.426, 0.639, 0.377, 0.508, 0.541],
    [0.590, 0.475, 0.377, 0.639, 0.6071],
]))

p2 = np.dot(mat1,mat1)
print("PROBABILITAS PERIODE\n\n")
print("matrix p2:\n")
print(p2, "\n\n")

p3 = np.dot(p2,p2)
print("matrix p3:\n")
print(p3, "\n\n")

p4 = np.dot(p3,p3)
print("matrix p4:\n")
print(p4, "\n\n")

p5 = np.dot(p4,p4)
print("matrix p5:\n")
print(p5, "\n\n")
```

```
PROBABILITAS PERIODE


matrix p2:

[[0.352728   0.402512   0.313496   0.452335   0.4355057 ]
 [0.529402   0.584762   0.465104   0.667769   0.6460434 ]
 [0.528608   0.556462   0.476925   0.669376   0.6453171 ]
 [0.840343   0.908203   0.751812   1.109454   1.0431041 ]
 [0.927112   1.0124855  0.8100467  1.2020039  1.12999641]]


matrix p3:

[[1.2871033  1.4035548  1.14015367 1.66350738 1.57991198]
 [1.90227637 2.07442934 1.68512235 2.45868812 2.33505809]
 [1.89393997 2.06486303 1.67796855 2.44825367 2.32491311]
 [3.07402706 3.35142278 2.72347482 3.97453464 3.77384976]
 [3.34895617 3.65176549 2.9669193  4.32952558 4.11131621]]


matrix p4:

[[16.89068932 18.41694099 14.96379092 21.8353318  20.73499477]
 [24.9641657  27.21994097 22.11624104 32.27226738 30.64598702]
 [24.85563841 27.10160682 22.02009474 32.13196983 30.51275916]
 [40.34630507 43.99201792 35.7435785  52.15743204 49.52908768]
 [43.95398338 47.92568824 38.93969091 56.82123491 53.95787013]]


matrix p5:

[[2909.35410219 3172.24485593 2577.45352691 3761.04917086 3571.52045601]
 [4299.97832292 4688.52660638 3809.43463901 5558.76986372 5278.64948758]
 [4281.28494848 4668.14408888 3792.87381411 5534.60412179 5255.70151809]
 [6949.4907531  7577.45036133 6156.68936219 8983.91035151 8531.18854281]
 [7570.89898214 8255.00936197 6707.20702876 9787.23192134 9294.02872093]]
```

Jakarta • Nov. 22, 2019

WHAT WE'VE DONE

```
NDtratition = np.array([0.164,0.213,0.262,0.426,0.590])
Ntrantition = np.array([0.180,0.246,0.246,0.639,0.475])
TEtrantition = np.array([0.148,0.246,0.361,0.377,0.377])
Ttrantition = np.array([0.311,0.410,0.295,0.508,0.639])
TDtrantition = np.array([0.197,0.344,0.361,0.541,0.6071])

hasil = NDtratition + Ntrantition + TEtrantition + Ttrantition + TDtrantition
phi = hasil / 5
print("AKURASI PROBABILITAS PERIODE\n\n")
print("PHI AS AN AVERAGE PROBABILITY TRANSITIONS MATRIX\n")
print(phi, "\n\n")

print("AKURASI DAY 2 FORECASTING\n")
akurasi2 = np.dot(phi,p2)
print(akurasi2, "\n\n")

print("AKURASI DAY 3 FORECASTING\n")
akurasi3 = np.dot(phi,p3)
print(akurasi3, "\n\n")

print("AKURASI DAY 4 FORECASTING\n")
akurasi4 = np.dot(phi,p4)
print(akurasi4, "\n\n")

print("AKURASI DAY 5 FORECASTING\n")
akurasi5 = np.dot(phi,p5)
print(akurasi5, "\n\n")
```

Forcast Report

```
AKURASI PROBABILITAS PERIODE


PHI AS AN AVERAGE PROBABILITY TRANSITIONS MATRIX

[0.2     0.2918  0.305   0.4982  0.53762]


AKURASI DAY 2 FORECASTING

[1.30334338 1.41765605 1.15392872 1.68843299 1.59962145]


AKURASI DAY 3 FORECASTING

[4.72210269 5.14875366 4.18344015 6.10461674 5.79690862]


AKURASI DAY 4 FORECASTING

[61.97472086 67.57478889 54.90461365 80.11742973 76.08011113]


AKURASI DAY 5 FORECASTING

[10674.89940833 11639.48888524  9457.10152939 13799.90890023
 13104.49682767]
```

http://finance.yahoo.com/q/hp
s=BBCA.JK&a=00&b=5&c=2012&
d=00&e=4&f=2013&g=d

https://youtu.be/Lc
HcZcCR8cA

https://github.com/
kelasterbuka

Most Used Websites

OUR REFERENCES

Jakarta • Nov. 22, 2019

Jakarta • Nov. 22, 2019

SOMETHING TO THINK ABOUT

Those who do not move, do not notice their chains.

ROSA LUXEMBURG