DAFTAR PUSTAKA

Carmona, R., & Carmona, R. A., 2004. Statistical analysis of financial data in SPlus. Springer.

Dowd, Kevin. 2005. Measuring Market Risk. https://doi.org/10.1002/9781118673485

Franke, I., Härdle, W.K., Hafner, C.M., 2015. Statistics of Financial Markets: An Introduction, Universitext. Springer Berlin Heidelberg, Berlin, Heidelberg. https://doi.org/10.1007/978-3-642-54539-9

Ghazali, P. L., Riaman, R., & Ulfatmi, R., 2020. Calculation of Value-at-Risk Variance-Covariance with the Approach of Simple Cash Portfolio, Factor Models and Cash Flow. In Operations Research: International Conference Series, 1(1), 19–24. https://doi.org/10.47194/orics.v1i1.20

Hardle, W., Simar, L. 2007. Applied Multivariate Statistical Analysis. Second Edition. New York: Springer Berlin Heidelberg.

Hardy, Charles O., 1923. Risk and Risk-Bearing, Chicago: University of Chicago Press.

Hicks, I. R., 1989. A suggestion for simplifying the theory of money. In General Equilibrium Models of Monetary Economies (pp. 7–23).

Iorion, Philippe., 2003. Financial Risk Manager Second Edition. United State Of America : GARP

Iorion, Philippe., 2007. VALUE AT RISK The New Benchmark for Managing Financial Risk THIRD EDITION. New York : The McGraw-Hill Companies

Konno, H., & Yamazaki, H., 1991. Mean-absolute deviation portfolio optimization model and its applications to Tokyo stock market. Management Science, 37(5), 519–531. INFORMS. https://doi.org/10.1287/mnsc.37.5.519

Kurbatsky, Victor G, Sidorov, Denis N, Spiryaev, Vadim A, Tomin, Nikita V. Forecasting nonstationary time series based on Hilbert-Huang transform and machine learning. Automation and Remote Control, 75, 922–934, 2014, Springer. DOI:10.1134/S0005117914050105

Leavens, D. H., 1945. Diversification of investments. Trusts and Estates, 80(5), 469–473.

Lintner, I., 1965. The valuation of risk assets and the selection of risky investments in stock portfolios and capital budgets. In Stochastic Optimization Models in Finance (pp. 131–155). http://www.istor.org/stable/1924119

Markowitz, Harry, M., (1959). Portfolio Selection: Efficient Diversification of Investments, New York: Iohn Wiley & Sons. https://www.istor.org/stable/i.ctt1bh4c8h

Mossin, I, 1966. Equilibrium in a capital asset market. Econometrica: Iournal of the Econometric Society, 768–783. ISTOR. https://doi.org/10.2307/1910098

Olson, David L, Wu, Desheng. The impact of distribution on value-at-risk measures. Mathematical and Computer Modelling, 58(9-10), 1670–1676, 2013, Elsevier. https://doi.org/10.1016/i.mcm.2011.06.053

Roy, A. D., 1952. Safety first and the holding of assets. Econometrica: Iournal of the Econometric Society, 431–449. ISTOR. https://www.istor.org/stable/i332601

Selection, P., 1959. Efficient Diversification of Investments. New York.

Sharpe, W. F., 1964. Capital asset prices: A theory of market equilibrium under conditions of risk. The Iournal of Finance, 19(3), 425–442. Wiley Online Library. https://doi.org/10.1111/i.1540-6261.1964.tb02865.x

Tobin, I., 1958. Liquidity preference as behavior towards risk. The Review of Economic Studies, 25(2), 65–86. Wiley-Blackwell.

Treynor, I., 1961. Toward a Theory of the Market Value of Risk Assets

Uchiyama, Y., Kadoya, T., & Nakagawa, K., 2019. Complex valued risk diversification. Entropy, 21(2), 119. MDPI. https://doi.org/10.3390/e21020119

LAMPIRAN

Lampiran 1 Data *return* setiap aset

No.	Tanggal	NG	COP	MFST	MAPI	IPY	CNY
1	2/1/2013	0.044025	-0.05134	0.01275	0.21875	-0.02656	-0.0375
2	3/1/2013	0.154332	-0.03805	0.028777	0.166667	-0.00198	-0.02068
3	4/1/2013	0.079274	-0.06593	0.157343	-0.09341	0.062288	0.019559
4	5/1/2013	-0.08266	0.035294	0.054381	0.078788	0.017934	-0.01782
5	6/1/2013	-0.10517	-0.07348	-0.01032	-0.21348	-0.01233	0.001468
6	7/1/2013	-0.03338	0.022077	-0.07817	-0.17143	0.009384	0.020956
7	8/1/2013	0.039176	0.03376	0.048995	-0.18103	-0.00292	-0.00735
8	9/1/2013	-0.00586	0.028633	-0.00359	0.305263	0.023503	0.022964
9	10/1/2013	0.005899	-0.00557	0.063702	-0.12903	0.005722	7.25E-05
10	11/1/2013	0.104161	-0.02693	0.077119	-0.0963	0.041773	0.000169
11	12/1/2013	0.069803	0.05629	-0.01888	0.127049	0.040026	0.005085
12	1/1/2014	0.168558	-0.05785	0.011494	-0.00909	-0.04933	-0.01789
13	2/1/2014	-0.06757	-0.00391	0.012421	0.266055	0.02115	0.037837
14	3/1/2014	-0.05164	-0.05114	0.069956	-0.0942	0.011317	0.009303
15	4/1/2014	0.101579	0.000992	-0.01439	0.0208	-0.00225	0.01389
16	5/1/2014	-0.0567	0.030718	0.013366	-0.18809	-2.14E-02	-0.01891
17	6/1/2014	-0.01783	0.026598	0.018564	-0.07722	-0.00029	-0.00253
18	7/1/2014	-0.13898	0.010145	0.035012	0.087866	-0.00772	-0.02684
19	8/1/2014	0.058318	-0.02565	0.052595	0.159615	-0.00712	-0.02399
20	9/1/2014	0.013776	-0.04615	0.020471	-0.08789	0.013539	-0.03894
21	10/1/2014	-0.06018	0.013134	0.012726	-0.03636	0.015525	-0.0126
22	11/1/2014	0.055513	-0.06268	0.018317	0.056604	0.049986	-0.00095
23	12/1/2014	-0.2933	-0.01068	-0.02845	-0.09286	-0.01957	-0.01859
24	1/1/2015	-0.06854	-0.11679	-0.13025	0.15748	-0.08454	-0.06029
25	2/1/2015	0.015979	0.078541	0.085396	-0.06803	0.00928	-0.00506
26	3/1/2015	-0.03438	0.018391	-0.07298	-0.04197	-0.03648	-0.05216
27	4/1/2015	0.042045	0.053083	0.196556	0.062857	0.039026	0.046335
28	5/1/2015	-0.03962	-0.05318	-0.0366	0.039427	0.01822	-0.02155
29	6/1/2015	0.071915	-0.04318	-0.05783	-0.04655	0.00022	0.013804
30	7/1/2015	-0.04096	-0.09407	0.057758	-0.15913	-0.00176	-0.01175

31	8/1/2015	-0.00994	-0.01287	-0.06809	-0.13978	-0.00191	0.047709
32	9/1/2015	-0.06136	0.001069	0.017004	-0.25	-0.01442	-0.00631
33	10/1/2015	-0.08043	-0.00748	0.189336	0.15	-0.00896	-0.02112
34	11/1/2015	-0.03705	-0.11943	0.032485	0.188406	-0.02049	-0.028
35	12/1/2015	0.045638	0.0435	0.020791	-0.07317	0.004923	0.043221
36	1/1/2016	-0.01669	-0.03021	-0.00703	-0.01316	0.003597	0.010382
37	2/1/2016	-0.25544	0.029944	-0.07642	0.133333	-0.06597	-5.62E-05
38	3/1/2016	0.144944	0.026026	0.085495	0.129412	0.045725	0.029944
39	4/1/2016	0.111792	0.043647	-0.09705	-0.125	-0.04888	0.010714
40	5/1/2016	0.050505	-0.08036	0.062763	-0.1	0.011247	-0.01246
41	6/1/2016	0.277972	0.045476	-0.03453	0.097884	-0.06925	0.00777
42	7/1/2016	-0.01642	0.013437	0.10768	0.144578	-0.00541	0.004878
43	8/1/2016	0.003825	-0.06742	0.013761	-0.03158	0.011839	0.004571
44	9/1/2016	0.006581	0.065301	0.002436	0	-1.31E-02	0.005987
45	10/1/2016	0.041294	-0.00113	0.040278	0.195652	0.010538	-0.00767
46	11/1/2016	0.107733	0.190897	0.005674	-0.04	0.052837	-0.0201
47	12/1/2016	0.110979	-0.04716	0.031198	0.022727	0.014197	0.00166
48	1/1/2017	-0.163	0.091	0.040393	0.024074	-0.00895	0.017125
49	2/1/2017	-0.11004	-0.00713	-0.01036	-0.02712	-0.02086	-0.02194
50	3/1/2017	0.149964	-0.0199	0.029384	0.111524	-0.00503	0.00964
51	4/1/2017	0.026959	-0.01974	0.039478	0.058528	0.024277	0.023925
52	5/1/2017	-0.06258	-0.00805	0.020158	-0.0079	0.024689	0.019742
53	6/1/2017	-0.01172	0.048135	-0.01303	0.082803	0.030761	0.01164
54	7/1/2017	-0.07941	0.069716	0.054693	-0.07353	0.016986	0.028048
55	8/1/2017	0.088046	0.066897	0.028473	0.111111	0.003218	-0.01448
56	9/1/2017	-0.01086	-0.04493	-0.00374	0	0.014587	0.001657
57	10/1/2017	-0.03691	0.051777	0.11666	-0.06429	-0.00414	-0.01727
58	11/1/2017	0.044544	-0.01689	0.011902	0.058015	0.012245	0.01849
59	12/1/2017	-0.0238	0.080183	0.016277	-0.10534	0.009259	-0.00775
60	1/1/2018	0.014223	-0.02848	0.110708	0.177419	0.003329	-1.28E-05
61	2/1/2018	-0.10952	-0.02308	-0.01305	0.041096	-0.041	-0.01151
62	3/1/2018	0.024747	-0.03049	-0.02666	0.023684	0.00669	0.001737
63	4/1/2018	0.010977	0.012183	0.024652	0.060411	0.008555	-0.01044
64	5/1/2018	0.068404	0.000325	0.056886	0.021818	-0.03666	-0.02027
65	6/1/2018	-0.00949	-0.03545	-0.00233	0.067616	0.016352	0.031764
66	7/1/2018	-0.04856	-0.04231	0.075753	-0.01667	0.011603	0.02891
67	8/1/2018	0.048167	-0.06196	0.058918	-0.0565	-0.01499	-0.00387
68	9/1/2018	0.03155	0.05273	0.018161	-0.01198	0.024375	0.006172

69	10/1/2018	0.084109	-0.0492	-0.0661	-0.0303	-0.03213	-0.01067
70	11/1/2018	0.41429	0.044807	0.038199	0.09375	0.005168	-0.00213
71	12/1/2018	-0.36253	-0.0558	-0.08405	-0.08	-0.02126	0.001906
72	1/1/2019	-0.04286	0.060053	0.028158	0.248447	-0.00828	-0.02809
73	2/1/2019	-0.00071	0.05701	0.072776	0.054726	0.016211	-0.00732
74	3/1/2019	-0.05334	-0.00288	0.052754	-0.08491	-0.01816	-0.0105
75	4/1/2019	-0.03268	-0.01208	0.107343	0.025773	0.005067	0.003228
76	5/1/2019	-0.04699	-0.09074	-0.05299	-0.11558	-0.03129	0.021927
77	6/1/2019	-0.05949	0.027646	0.083118	0.022727	0.013135	0.011169
78	7/1/2019	-0.0325	-0.01548	0.017244	0.05	-0.0181	-0.02326
79	8/1/2019	0.023287	-0.0466	0.011668	0.047619	-0.03006	0.031393
80	9/1/2019	0.019694	0.012367	0.008487	0.040404	0.008476	-0.0092
81	10/1/2019	0.130043	0.024821	0.031216	-0.02913	0.022498	0.007407
82	11/1/2019	-0.13369	0.005109	0.055869	0.015	0.001578	-0.01311
83	12/1/2019	-0.04033	0.053087	0.041749	0.039409	0.009285	0.007722
84	1/1/2020	-0.15898	-0.09903	0.079455	-0.09005	-0.01248	-0.01406
85	2/1/2020	-0.08528	0.007937	-0.04829	-0.16667	-0.00898	0.001638
86	3/1/2020	-0.02613	-0.12126	-0.02654	-0.41	-0.0047	0.013273
87	4/1/2020	0.188415	0.050179	0.136326	0.387712	-0.01003	-0.00963
88	5/1/2020	-0.05131	0.038609	0.022543	0.10687	0.01908	0.02379
89	6/1/2020	-0.053	0.120764	0.110559	0.075862	0.013039	0.002096
90	7/1/2020	0.027413	0.054426	0.007371	-0.12179	0.028548	0.035162
91	8/1/2020	0.461923	0.062044	0.100093	-0.0073	0.013878	-0.00496
92	9/1/2020	-0.03916	-0.00753	-0.0674	-0.16176	-0.02271	-0.02655
93	10/1/2020	0.327266	0.006265	-0.03737	0.149123	-0.01336	-0.02036
94	11/1/2020	-0.14073	0.125184	0.057292	0.229008	0.020514	0.006389
95	12/1/2020	-0.11901	0.024902	0.039006	-0.01863	0.01383	0.015949
96	1/1/2021	0.009846	0.011083	0.042892	-0.02532	0.007534	-0.02153
97	2/1/2021	0.080733	0.150225	0.001811	0.045455	0.012909	0.002321
98	3/1/2021	-0.05882	-0.0226	0.014588	-0.08075	0.008937	-0.01683
99	4/1/2021	0.12385	0.117	0.069602	0.081081	0.011477	0.012402
100	5/1/2021	0.018765	0.047672	-0.00991	-0.125	0.020027	0.000848
101	6/1/2021	0.222371	-0.08374	0.084989	-0.10714	-0.01672	-0.0169
102	7/1/2021	0.072329	0.041152	0.051717	-0.008	-0.01139	0.001934
103	8/1/2021	0.118293	-0.02139	0.059563	0.193548	-0.00238	-0.00539
104	9/1/2021	0.340416	-0.0643	-0.06612	0.054054	-0.00801	-0.02145
105	10/1/2021	-0.07517	0.066031	0.176291	0.128205	0.022815	-0.00796
106	11/1/2021	-0.15831	-0.01801	-0.00311	-0.15909	-0.02701	-0.02574

107	12/1/2021	-0.18327	0.042752	0.017333	-0.04054	0.020119	0.000956
108	1/1/2022	0.306702	-0.03069	-0.07534	0.098592	-0.0117	-0.01058
109	2/1/2022	-0.09684	0.029585	-0.0392	0.00641	-0.00217	-0.00942
110	3/1/2022	0.28169	0.066786	0.031862	0.121019	0.043718	-0.0089
111	4/1/2022	0.283942	-0.07229	-0.09987	0.022727	0.016339	-0.00697
112	5/1/2022	0.124379	-0.02484	-0.02036	0	0.009207	0.027995
113	6/1/2022	-0.33407	-0.13701	-0.05532	0.088889	0.030121	-0.0195
114	7/1/2022	0.517146	-0.03639	0.093097	-0.09184	-0.04344	-0.0186
115	8/1/2022	0.109126	-0.01566	-0.06864	0.134831	0.026894	0.005515
116	9/1/2022	-0.25868	-0.03026	-0.10927	0.049505	0.014955	0.006091
117	10/1/2022	-0.06074	-0.01099	-0.00331	0.136792	0.036168	0.035234
118	11/1/2022	0.09048	0.107556	0.099125	0.20332	-0.02279	0.02202
119	12/1/2022	-0.35426	0.019395	-0.06005	-0.00345	-0.02305	0.000868
120	1/1/2023	-0.40022	0.109041	0.033317	-0.10035	0.007056	-0.00612
121	2/1/2023	0.023472	-0.03597	0.006497	0.161538	0.019462	-0.0006
122	3/1/2023	-0.1933	0.005032	0.155882	0	-0.00083	0.015276
123	4/1/2023	0.087545	-0.05209	0.065765	-0.09272	0.043424	0.02312
124	5/1/2023	-0.05975	-0.06292	0.068769	0.29562	-0.00832	-0.00207
125	6/1/2023	0.234775	0.033682	0.036999	-0.04789	0.057208	0.040934
126	7/1/2023	-0.05861	0.066099	-0.01357	0.171598	-0.00635	-0.00723
127	8/1/2023	0.050873	-0.04641	-0.02429	-0.02273	0.008437	0.001859
128	9/1/2023	0.058165	-0.02211	-0.03664	-0.05943	0.00057	-0.01924
129	10/1/2023	0.149198	-0.02001	0.044561	0.076923	0.016154	0.002592

Lampiran 2 Program Software Python

```
#memanggil paket
import math as mt
import numpy as np
import pandas as pd

#load the data
from google.colab import files
files.upload()

#memanggil data
data=pd.read_csv("SAHAM1.csv")
print(data)
           NG       COP       MFST       MAPI       JPY       CNY
```

```
0    0.044025 -0.051344  0.012750  0.218750 -0.026555 -0.037499
1    0.154332 -0.038055  0.028777  0.166667 -0.001984 -0.020680
2    0.079274 -0.065934  0.157343 -0.093407  0.062288  0.019559
3   -0.082662  0.035294  0.054381  0.078788  0.017934 -0.017816
4   -0.105171 -0.073485 -0.010315 -0.213483 -0.012332  0.001468
..        ...       ...       ...       ...       ...       ...
124  0.234775  0.033682  0.036999 -0.047887  0.057208  0.040934
125 -0.058613  0.066099 -0.013567  0.171598 -0.006351 -0.007230
126  0.050873 -0.046407 -0.024291 -0.022727  0.008437  0.001859
127  0.058165 -0.022109 -0.036643 -0.059432  0.000570 -0.019241
128  0.149198 -0.020013  0.044561  0.076923  0.016154  0.002592

[129 rows x 6 columns]
```

```
#pendefinisian data
x1=data["NG"]
x2=data["COP"]
x3=data["MFST"]
x4=data["MAPI"]
x5=data["JPY"]
x6=data["CNY"]
i=1j
N=len(data)

#menghitung transformasi hilbert
exp = [np.cos(2*np.pi*(k/N)) + 1j*np.sin(2*np.pi*(k/N)) for k in
range(N)]


sgn1=-i*np.sign(1-N/2)
rk1=x1*exp
sig1=sum(rk1)
hdr1=sgn1*sig1
z1t=x1+i*hdr1

sgn2=-i*np.sign(2-N/2)
rk2=x2*exp
sig2=sum(rk2)
hdr2=sgn2*sig2
z2t=x2+i*hdr2

sgn3=-i*np.sign(3-N/2)
rk3=x3*exp
sig3=sum(rk3)
```

```
hdr3=sgn3*sig3
z3t=x3+i*hdr3

sgn4=-i*np.sign(4-N/2)
rk4=x4*exp
sig4=sum(rk4)
hdr4=sgn4*sig4
z4t=x4+i*hdr4

sgn5=-i*np.sign(5-N/2)
rk5=x5*exp
sig5=sum(rk5)
hdr5=sgn5*sig5
z5t=x5+i*hdr5

sgn6=-i*np.sign(6-N/2)
rk6=x6*exp
sig6=sum(rk6)
hdr6=sgn6*sig6
z6t=x6+i*hdr6

zt=(z1t,z2t,z3t,z4t,z5t,z6t)
print(zt)
```

```
(0     -0.096206+1.231498j
1      0.014100+1.231498j
2     -0.060957+1.231498j
3     -0.222893+1.231498j
4     -0.245402+1.231498j
              ...
124    0.094544+1.231498j
125   -0.198845+1.231498j
126   -0.089358+1.231498j
127   -0.082066+1.231498j
128    0.008966+1.231498j
Name: NG, Length: 129, dtype: complex128, 0      0.515205+0.509814j
1      0.528494+0.509814j
2      0.500615+0.509814j
3      0.601843+0.509814j
4      0.493064+0.509814j
              ...
124    0.600230+0.509814j
125    0.632648+0.509814j
126    0.520141+0.509814j
127    0.544440+0.509814j
128    0.546535+0.509814j
```

```
Name: COP, Length: 129, dtype: complex128, 0      0.415097+0.102520j
1      0.431124+0.102520j
2      0.559690+0.102520j
3      0.456728+0.102520j
4      0.392032+0.102520j
              ...
124    0.439346+0.102520j
125    0.388780+0.102520j
126    0.378055+0.102520j
127    0.365704+0.102520j
128    0.446907+0.102520j
Name: MFST, Length: 129, dtype: complex128, 0     -
0.018135+0.412183j
1     -0.070218+0.412183j
2     -0.330291+0.412183j
3     -0.158097+0.412183j
4     -0.450368+0.412183j
              ...
124   -0.284772+0.412183j
125   -0.065287+0.412183j
126   -0.259612+0.412183j
127   -0.296316+0.412183j
128   -0.159961+0.412183j
Name: MAPI, Length: 129, dtype: complex128, 0     -
0.309197+0.198615j
1     -0.284625+0.198615j
2     -0.220354+0.198615j
3     -0.264708+0.198615j
4     -0.294974+0.198615j
              ...
124   -0.225433+0.198615j
125   -0.288993+0.198615j
126   -0.274204+0.198615j
127   -0.282071+0.198615j
128   -0.266488+0.198615j
Name: JPY, Length: 129, dtype: complex128, 0      0.077965-0.003427j
1      0.094785-0.003427j
2      0.135023-0.003427j
3      0.097648-0.003427j
4      0.116932-0.003427j
              ...
124    0.156398-0.003427j
125    0.108234-0.003427j
126    0.117323-0.003427j
127    0.096223-0.003427j
128    0.118056-0.003427j
Name: CNY, Length: 129, dtype: complex128)

# membentuk matriks diagonal E[z1t-E(z1t)]*[(z1t-E(z1t))*]
e1=sum(z1t)/N
ze1=z1t-e1
```

```
z1tk=np.conj(z1t)
ze1k=np.conj(ze1)
z1tot=ze1*ze1k
ez1t=sum(z1tot)/N
z1=complex(round(ez1t.real,18),round(ez1t.imag,18))

e2=sum(z2t)/N
ze2=z2t-e2
z2tk=np.conj(z2t)
ze2k=np.conj(ze2)
z2tot=ze2*ze2k
ez2t=sum(z2tot)/N
z2=complex(round(ez2t.real,18),round(ez2t.imag,18))

e3=sum(z3t)/N
ze3=z3t-e3
z3tk=np.conj(z3t)
ze3k=np.conj(ze3)
z3tot=ze3*ze3k
ez3t=sum(z3tot)/N
z3=complex(round(ez3t.real,18),round(ez3t.imag,18))

e4=sum(z4t)/N
ze4=z4t-e4
z4tk=np.conj(z4t)
ze4k=np.conj(ze4)
z4tot=ze4*ze4k
ez4t=sum(z4tot)/N
z4=complex(round(ez4t.real,18),round(ez4t.imag,18))

e5=sum(z5t)/N
ze5=z5t-e5
z5tk=np.conj(z5t)
ze5k=np.conj(ze5)
z5tot=ze5*ze5k
ez5t=sum(z5tot)/N
z5=complex(round(ez5t.real,18),round(ez5t.imag,18))

e6=sum(z6t)/N
ze6=z6t-e6
z6tk=np.conj(z6t)
```

```
ze6k=np.conj(ze6)
z6tot=ze6*ze6k
ez6t=sum(z6tot)/N
z6=complex(round(ez6t.real,18),round(ez6t.imag,18))

#membentuk matriks non diagonal
z7tot=ze1*ze2
ez7t=sum(z7tot)/N
z7=complex(round(ez7t.real,18),round(ez7t.imag,18))

z12tot=ze2*ze1
ez12t=sum(z12tot)/N
z12=complex(round(ez12t.real,18),round(ez12t.imag,18))

z8tot=ze1*ze3
ez8t=sum(z8tot)/N
z8=complex(round(ez8t.real,18),round(ez8t.imag,18))

z17tot=ze3*ze1
ez17t=sum(z17tot)/N
z17=complex(round(ez17t.real,18),round(ez17t.imag,18))

z9tot=ze1*ze4
ez9t=sum(z9tot)/N
z9=complex(round(ez9t.real,18),round(ez9t.imag,18))

z22tot=ze4*ze1
ez22t=sum(z22tot)/N
z22=complex(round(ez22t.real,18),round(ez22t.imag,18))

z10tot=ze1*ze5
ez10t=sum(z10tot)/N
z10=complex(round(ez10t.real,18),round(ez10t.imag,18))

z27tot=ze5*ze1
ez27t=sum(z27tot)/N
z27=complex(round(ez27t.real,18),round(ez27t.imag,18))

z11tot=ze1*ze6
ez11t=sum(z11tot)/N
z11=complex(round(ez11t.real,18),round(ez11t.imag,18))
```

```
z32tot=ze6*ze1
ez32t=sum(z32tot)/N
z32=complex(round(ez32t.real,18),round(ez32t.imag,18))

z13tot=ze2*ze3
ez13t=sum(z13tot)/N
z13=complex(round(ez13t.real,18),round(ez13t.imag,18))

z18tot=ze3*ze2
ez18t=sum(z18tot)/N
z18=complex(round(ez18t.real,18),round(ez18t.imag,18))

z14tot=ze2*ze4
ez14t=sum(z14tot)/N
z14=complex(round(ez14t.real,18),round(ez14t.imag,18))

z23tot=ze4*ze2
ez23t=sum(z23tot)/N
z23=complex(round(ez23t.real,18),round(ez23t.imag,18))

z15tot=ze2*ze5
ez15t=sum(z15tot)/N
z15=complex(round(ez15t.real,18),round(ez15t.imag,18))

z28tot=ze5*ze2
ez28t=sum(z28tot)/N
z28=complex(round(ez28t.real,18),round(ez28t.imag,18))

z16tot=ze2*ze6
ez16t=sum(z16tot)/N
z16=complex(round(ez16t.real,18),round(ez16t.imag,18))

z33tot=ze6*ze2
ez33t=sum(z33tot)/N
z33=complex(round(ez33t.real,18),round(ez33t.imag,18))

z19tot=ze3*ze4
ez19t=sum(z19tot)/N
z19=complex(round(ez19t.real,18),round(ez19t.imag,18))
```

```
z24tot=ze4*ze3
ez24t=sum(z24tot)/N
z24=complex(round(ez24t.real,18),round(ez24t.imag,18))

z20tot=ze3*ze5
ez20t=sum(z20tot)/N
z20=complex(round(ez20t.real,18),round(ez20t.imag,18))

z29tot=ze5*ze3
ez29t=sum(z29tot)/N
z29=complex(round(ez29t.real,18),round(ez29t.imag,18))

z21tot=ze3*ze6
ez21t=sum(z21tot)/N
z21=complex(round(ez21t.real,18),round(ez21t.imag,18))

z34tot=ze6*ze3
ez34t=sum(z34tot)/N
z34=complex(round(ez34t.real,18),round(ez34t.imag,18))

z25tot=ze4*ze5
ez25t=sum(z25tot)/N
z25=complex(round(ez25t.real,18),round(ez25t.imag,18))

z30tot=ze5*ze4
ez30t=sum(z30tot)/N
z30=complex(round(ez30t.real,18),round(ez30t.imag,18))

z26tot=ze4*ze6
ez26t=sum(z26tot)/N
z26=complex(round(ez26t.real,18),round(ez26t.imag,18))

z35tot=ze6*ze4
ez35t=sum(z35tot)/N
z35=complex(round(ez35t.real,18),round(ez35t.imag,18))

z31tot=ze5*ze6
ez31t=sum(z31tot)/N
z31=complex(round(ez31t.real,18),round(ez31t.imag,18))

z36tot=ze6*ze5
```

```
ez36t=sum(z36tot)/N
z36=complex(round(ez36t.real,18),round(ez36t.imag,18))

#matriks varians-kovarians dan menghitung invers
matrix=np.matrix([[z1,z7,z8,z9,z10,z11],[z12,z2,z13,z14,z15,z16],
[z17,z18,z3,z19,z20,z21],[z22,z23,z24,z4,z25,z26],[z27,z28,z29,z3
0,z5,z31],[z32,z33,z34,z35,z36,z6]])
invers=np.linalg.inv(matrix)
print(invers)

[[ 4.74422579e+01-0.j  3.33809421e+00-0.j -9.03247736e+00+0.j
  -6.93644599e+00+0.j -1.41956806e+01+0.j  3.97167973e-01-0.j]
 [ 3.33809421e+00-0.j  3.37762332e+02-0.j -5.04452146e+01+0.j
  -2.08398347e+01+0.j -8.60120276e+01+0.j -1.08539552e+02+0.j]
 [-9.03247736e+00+0.j -5.04452146e+01+0.j  3.12743854e+02-0.j
  -1.78269019e+01+0.j -2.05722082e+02+0.j  1.36975132e+01-0.j]
 [-6.93644599e+00+0.j -2.08398347e+01+0.j -1.78269019e+01+0.j
   7.06718214e+01-0.j -2.57891406e+00+0.j  4.12641916e+01-0.j]
 [-1.41956806e+01+0.j -8.60120276e+01+0.j -2.05722082e+02+0.j
  -2.57891406e+00+0.j  2.22899887e+03+0.j -1.07783036e+03+0.j]
 [ 3.97167973e-01+0.j -1.08539552e+02+0.j  1.36975132e+01+0.j
   4.12641916e+01+0.j -1.07783036e+03+0.j  3.30819923e+03+0.j]]

#membuat vector
vector = np.array([1, 1, 1, 1, 1, 1]).reshape(-1, 1)
vt=np.transpose(vector)

#membuat vektor Sigma^2
sig=np.array([z1,z2,z3,z4,z5,z6]).reshape(-1, 1)

# menghitung bobot mula mula lamda=0
iv = invers @ vector
vi = vt @ invers
viv = vi @ vector
imu = invers @ sig
vim = vi @ sig
ipv = vim / viv
bot = imu - (iv * ipv)
bobot1 = iv / viv
print("bobot minimum", bobot1)
bobot minimum [[0.00651908+0.j]
 [0.02334997+0.j]
 [0.01346905+0.j]
 [0.01977912+0.j]
 [0.26142821+0.j]
```

```
 [0.67545457+0.j]]

# Menghitung Bobot Complex
# Inisialisasi bobot awal
bobot = bobot1

# Menghitung dan mencetak total sum untuk bobot awal
total_sum = np.sum(bobot)
print(f"Inisial bobot: {bobot}")
print(f"Total sum: {total_sum}")
print("\n")

# Inisialisasi variabel untuk menyimpan bobot terakhir yang valid
bobot_optimal = None

# Iterasi untuk nilai t dari 0.01 hingga 0.09
for t in np.arange(0.01, 0.1, 0.01):
    # Menghitung bobot untuk nilai t saat ini
    bobot = bobot + (t * bot)


    # Pengecekan apakah ada elemen negatif dalam bobot
    if any((i < 0).any() for i in bobot):
        print("Ditemukan bobot minus. Iterasi dihentikan.")
        break

    # Menyimpan bobot saat ini sebagai bobot terakhir yang valid
    bobot_optimal = bobot

    # Menghitung dan mencetak total sum untuk bobot saat ini
    total_sum = np.sum(bobot)
    print(f"Bobot untuk t={t}: {bobot}")
    print(f"Total sum untuk t={t}: {total_sum}")
    print("\n")

# Menggunakan bobot terakhir yang valid sebagai hasil
hasil_akhir = bobot_optimal
print(f"Bobot akhir yang tidak negatif: {hasil_akhir}")

Inisial bobot: [[0.00651908+0.j]
 [0.02334997+0.j]
 [0.01346905+0.j]
```

```
 [0.01977912+0.j]
 [0.26142821+0.j]
 [0.67545457+0.j]]
Total sum: (1+0j)


Bobot untuk t=0.01: [[0.01524315+0.j]
 [0.02846571+0.j]
 [0.01715644+0.j]
 [0.02710394+0.j]
 [0.24863747+0.j]
 [0.66339329+0.j]]
Total sum untuk t=0.01: (1+0j)


Bobot untuk t=0.02: [[0.03269129+0.j]
 [0.03869719+0.j]
 [0.02453122+0.j]
 [0.04175357+0.j]
 [0.22305598+0.j]
 [0.63927074+0.j]]
Total sum untuk t=0.02: (1+0j)


Bobot untuk t=0.03: [[0.0588635 +0.j]
 [0.05404441+0.j]
 [0.0355934 +0.j]
 [0.06372802+0.j]
 [0.18468376+0.j]
 [0.60308691+0.j]]
Total sum untuk t=0.03: (1+0j)


Bobot untuk t=0.04: [[0.09375978+0.j]
 [0.07450737+0.j]
 [0.05034297+0.j]
 [0.09302728+0.j]
 [0.13352079+0.j]
 [0.5548418 +0.j]]
Total sum untuk t=0.04: (1+0j)


Bobot untuk t=0.05: [[0.13738014+0.j]
 [0.10008607+0.j]
 [0.06877993+0.j]
 [0.12965136+0.j]
 [0.06956708+0.j]
 [0.49453542+0.j]]
Total sum untuk t=0.05: (1+0j)


Ditemukan bobot minus. Iterasi dihentikan.
```

```
Bobot akhir yang tidak negatif: [[0.13738014+0.j]
 [0.10008607+0.j]
 [0.06877993+0.j]
 [0.12965136+0.j]
 [0.06956708+0.j]
 [0.49453542+0.j]]
```

```
# menghitung portofolio
z1w = x1 * hasil_akhir[0, 0]
z2w = x2 * hasil_akhir[1, 0]
z3w = x3 * hasil_akhir[2, 0]
z4w = x4 * hasil_akhir[3, 0]
z5w = x5 * hasil_akhir[4, 0]
z6w = x6 * hasil_akhir[5, 0]
zw=z1w+z2w+z3w+z4w+z5w+z6w
print(zw)
```

```
0      0.009756+0.000000j
1      0.030616+0.000000j
2      0.017009+0.000000j
3     -0.001431+0.000000j
4     -0.050323+0.000000j
              ...
124    0.056184+0.000000j
125    0.015861+0.000000j
126   -0.000767+0.000000j
127   -0.013923+0.000000j
128    0.033937+0.000000j
Length: 129, dtype: complex128
```

```
# menghitung standar deviasi
std_dev_real1 = np.std(np.real(zw))
std_dev_imag1 = np.std(np.imag(zw))

print("Standar Deviasi pada Bagian Real:", std_dev_real1)
print("Standar Deviasi pada Bagian Imajiner:", std_dev_imag1)

# menghitung Value at Risk
VaR95=std_dev_real1*(1.65)*(1000000)
print("Nilai VaR 95%:", VaR95)
```

```
Standar Deviasi pada Bagian Real: 0.03201907146862666
Standar Deviasi pada Bagian Imajiner: 0.0
Nilai VaR 95%: 52831.46792323399
```

```
#Transformasi Fourier
#memanggil paket
import math as mt
import numpy as np
import pandas as pd

#load the data
from google.colab import files
files.upload()

pip install numpy matplotlib

import numpy as np
import matplotlib.pyplot as plt

#memanggil data
data=pd.read_csv("SAHAM1.csv")
print(data)
#pendefinisian data
x1=data["NG"]
x2=data["COP"]
x3=data["MFST"]
x4=data["MAPI"]
x5=data["JPY"]
x6=data["CNY"]
i=1j
N=len(data)
```

```
           NG       COP      MFST      MAPI       JPY       CNY
0    0.044025 -0.051344  0.012750  0.218750 -0.026555 -0.037499
1    0.154332 -0.038055  0.028777  0.166667 -0.001984 -0.020680
2    0.079274 -0.065934  0.157343 -0.093407  0.062288  0.019559
3   -0.082662  0.035294  0.054381  0.078788  0.017934 -0.017816
4   -0.105171 -0.073485 -0.010315 -0.213483 -0.012332  0.001468
..        ...       ...       ...       ...       ...       ...
124  0.234775  0.033682  0.036999 -0.047887  0.057208  0.040934
125 -0.058613  0.066099 -0.013567  0.171598 -0.006351 -0.007230
126  0.050873 -0.046407 -0.024291 -0.022727  0.008437  0.001859
127  0.058165 -0.022109 -0.036643 -0.059432  0.000570 -0.019241
128  0.149198 -0.020013  0.044561  0.076923  0.016154  0.002592

[129 rows x 6 columns]
```

```
# Hitung Discrete Fourier Transform (DFT)
```

```
X1 = np.fft.fft(x1)
print(X1)
[ 1.40296433+0.j          0.14023122+1.23149843j -
0.94385694+0.8582028j
  0.40489706-1.71998427j  1.15810892-1.84642802j -0.25617617-
0.4842747j
  1.91095456-0.37632453j  1.11100606+0.77398161j
0.27016211+1.59707996j
  0.75844296+2.07833038j  1.1635019 +1.28250061j -
2.17036579+1.64639071j
 -1.42631123-1.17017345j  1.70941734-0.07629278j  0.86115197-
0.09012476j
  0.15475523-0.24571678j -0.29323304-0.87283336j  0.41260675-
1.34032943j
  0.11152125-0.51846305j  1.25531554+0.20163574j
1.51854822+0.32020469j
  0.8040502 +0.30886791j -2.59656156+0.6925899j   1.2713339 -
1.75330969j
  0.58378065+0.21935569j -0.37707725+0.67449046j
1.27959225+0.86824619j
  0.5395101 -0.94751431j -0.98095983+0.43089821j -0.73518395-
2.12708869j
 -0.16006463-0.25064823j  0.02904336+0.1496485j   0.11746793-
1.02659199j
 -0.277192  +0.51237363j  1.66402311+0.85479032j -
0.40205243+1.65078683j
 -0.85508867+1.8916725j  -2.71222935+1.60349679j -1.87519474-
2.95038732j
  0.24877976-2.14239756j -0.23338124-1.79405488j
0.89614744+0.69238234j
  1.60153379+0.50866998j  0.25745398+1.53001618j -
2.19038496+1.00425247j
  1.50446701-0.42740037j -1.41973639-0.52502011j -
0.59070131+0.38430275j
 -0.32659045+0.03741105j  1.53687338-1.71856514j  0.2980138
+0.93867268j
 -1.23964902-1.08806694j -0.30812174-0.37070213j -0.40224056-
0.96218594j
 -0.42967   +2.23491703j  0.96025909-0.11841377j  1.49145354-
0.09749745j
  0.00765064-0.30703409j  1.09584525+0.45209511j
0.55618285+1.38622575j
  0.05016702+1.84247678j -3.0507722 +2.00491335j -1.39570867-
2.50386355j
  0.47152546+0.02496281j -0.41913105+0.22215973j -0.41913105-
0.22215973j
  0.47152546-0.02496281j -1.39570867+2.50386355j -3.0507722 -
2.00491335j
  0.05016702-1.84247678j  0.55618285-1.38622575j  1.09584525-
0.45209511j
```

```
  0.00765064+0.30703409j  1.49145354+0.09749745j
0.96025909+0.11841377j
 -0.42967   -2.23491703j -0.40224056+0.96218594j -
0.30812174+0.37070213j
 -1.23964902+1.08806694j  0.2980138 -0.93867268j
1.53687338+1.71856514j
 -0.32659045-0.03741105j -0.59070131-0.38430275j -
1.41973639+0.52502011j
  1.50446701+0.42740037j -2.19038496-1.00425247j  0.25745398-
1.53001618j
  1.60153379-0.50866998j  0.89614744-0.69238234j -
0.23338124+1.79405488j
  0.24877976+2.14239756j -1.87519474+2.95038732j -2.71222935-
1.60349679j
 -0.85508867-1.8916725j  -0.40205243-1.65078683j  1.66402311-
0.85479032j
 -0.277192  -0.51237363j  0.11746793+1.02659199j  0.02904336-
0.1496485j
 -0.16006463+0.25064823j -0.73518395+2.12708869j -0.98095983-
0.43089821j
  0.5395101 +0.94751431j  1.27959225-0.86824619j -0.37707725-
0.67449046j
  0.58378065-0.21935569j  1.2713339 +1.75330969j -2.59656156-
0.6925899j
  0.8040502 -0.30886791j  1.51854822-0.32020469j  1.25531554-
0.20163574j
  0.11152125+0.51846305j  0.41260675+1.34032943j -
0.29323304+0.87283336j
  0.15475523+0.24571678j  0.86115197+0.09012476j
1.70941734+0.07629278j
 -1.42631123+1.17017345j -2.17036579-1.64639071j  1.1635019 -
1.28250061j
  0.75844296-2.07833038j  0.27016211-1.59707996j  1.11100606-
0.77398161j
  1.91095456+0.37632453j -0.25617617+0.4842747j
1.15810892+1.84642802j
  0.40489706+1.71998427j -0.94385694-0.8582028j   0.14023122-
1.23149843j]
```

```
# Hitung Discrete Fourier Transform (DFT)
X2 = np.fft.fft(x2)
print(X2)
```

```
[ 0.18885736+0.j         -0.56654857+0.50981355j -
0.53631672+0.34240652j
  0.69227595-0.94275746j -0.06336401+0.24425756j -0.4546964
+0.71629675j
 -0.5106128 -0.28376107j  0.00997073-0.04838238j -0.73646669-
0.00571736j
 -0.14772307-0.17642103j -0.12881946-0.51224326j  0.51027763-
0.07863593j
```

```
  0.10599457-0.24451072j -0.16709523+0.91636414j -
0.41155864+1.47878148j
 -0.17450333-0.44615481j  0.05839672+0.44471207j -
0.26824408+0.03584954j
 -0.41168497-0.82994773j -0.20541505+0.26118813j  0.0623805
+0.06355168j
  0.00373918-0.45789963j  0.12124554-0.08932527j -0.53124935-
0.14610984j
 -0.09713164-0.0590984j   1.15377342-0.07957106j  0.16670329-
0.02658339j
  0.26996948-0.54258431j -0.60593119+0.27233102j
0.29516679+0.04422866j
 -0.45380798+0.0267032j  -0.01870096-0.14752537j
0.48888632+0.002475j
 -0.03388177+0.19657844j  0.06087613+0.48000273j -
0.09652916+0.28682952j
 -0.18203181+0.18961092j  0.08718481+0.65782953j
0.32812451+0.07707155j
 -0.04036846-0.06291835j -0.49119151+0.55173579j
0.08459373+0.01657096j
  0.41531981-0.40044269j  0.14360329+0.18088625j -
0.29318681+0.45939656j
 -0.49971793-0.17141339j  0.31155221-0.52042314j -
0.12791421+0.22353625j
 -0.6248341 +0.39111207j  0.06093419-1.09030997j  0.00749117-
0.48234133j
  0.91284358+0.03952101j  0.53586129+0.67810218j -0.5520866 -
0.1897934j
  0.75496435-0.4145361j  -0.07311268-0.53569705j -0.6068565 -
0.45846387j
  0.05800284+0.49008711j  0.23070855-0.471375j   -
1.40479137+0.40689419j
  0.64114198-0.31701617j -0.79049948-0.63416465j -0.71321629-
0.56891585j
  0.60655547+0.46748089j  0.43544938+0.56599255j  0.43544938-
0.56599255j
  0.60655547-0.46748089j -0.71321629+0.56891585j -
0.79049948+0.63416465j
  0.64114198+0.31701617j -1.40479137-0.40689419j
0.23070855+0.471375j
  0.05800284-0.49008711j -0.6068565 +0.45846387j -
0.07311268+0.53569705j
  0.75496435+0.4145361j  -0.5520866 +0.1897934j   0.53586129-
0.67810218j
  0.91284358-0.03952101j  0.00749117+0.48234133j
0.06093419+1.09030997j
 -0.6248341 -0.39111207j -0.12791421-0.22353625j
0.31155221+0.52042314j
 -0.49971793+0.17141339j -0.29318681-0.45939656j  0.14360329-
0.18088625j
  0.41531981+0.40044269j  0.08459373-0.01657096j -0.49119151-
0.55173579j
```

```
 -0.04036846+0.06291835j  0.32812451-0.07707155j  0.08718481-
0.65782953j
 -0.18203181-0.18961092j -0.09652916-0.28682952j  0.06087613-
0.48000273j
 -0.03388177-0.19657844j  0.48888632-0.002475j   -
0.01870096+0.14752537j
 -0.45380798-0.0267032j   0.29516679-0.04422866j -0.60593119-
0.27233102j
  0.26996948+0.54258431j  0.16670329+0.02658339j
1.15377342+0.07957106j
 -0.09713164+0.0590984j  -0.53124935+0.14610984j
0.12124554+0.08932527j
  0.00373918+0.45789963j  0.0623805 -0.06355168j -0.20541505-
0.26118813j
 -0.41168497+0.82994773j -0.26824408-0.03584954j  0.05839672-
0.44471207j
 -0.17450333+0.44615481j -0.41155864-1.47878148j -0.16709523-
0.91636414j
  0.10599457+0.24451072j  0.51027763+0.07863593j -
0.12881946+0.51224326j
 -0.14772307+0.17642103j -0.73646669+0.00571736j
0.00997073+0.04838238j
 -0.5106128 +0.28376107j -0.4546964 -0.71629675j -0.06336401-
0.24425756j
  0.69227595+0.94275746j -0.53631672-0.34240652j -0.56654857-
0.50981355j]
```

```
# Hitung Discrete Fourier Transform (DFT)
X3 = np.fft.fft(x3)
print(X3)
```

```
[ 2.74350998+0.j         -0.40234691+0.10251993j  0.22442563-
0.37292854j
  0.64474035-0.37084578j  0.41812605+0.23468222j
0.27048524+0.21869354j
 -0.45063199+0.6453621j  -0.56211698+0.23378354j -0.29636679-
0.10558235j
  0.36890854-0.69071684j -0.12679964-0.21325669j -0.43505679-
0.16119828j
  0.38746463-0.28029435j -0.13567047-0.44669198j  0.39274447-
0.40784161j
  0.41361353-0.26197772j  1.33565984-0.61877959j -0.60338256-
0.33490204j
 -0.10993834-0.4747736j   0.35636489+0.50022012j  0.43773278-
0.37692783j
  0.13287684-0.90813988j  0.061132  -0.37765355j -
0.47692637+0.52942611j
 -0.31509526+0.03378537j  0.11429021-0.09916212j -0.51461367-
0.00332457j
 -0.55088959+0.26594257j -0.49828755+0.48229839j -0.47456363-
0.59382704j
```

```
  0.07072209-0.10329931j  0.48290372-0.09713752j  0.66395267-
0.40891592j
  0.23681294+0.47512135j -0.71485016+0.92471399j -
0.50470144+0.39692716j
 -0.1822969 -0.22066812j  0.01515471+0.410188j   -
0.96973612+0.07575831j
  0.35968294+0.24706741j -0.15491924+0.02483643j
0.47027342+0.18037459j
  0.1064606 -0.21247382j -0.18786645+1.24534932j  0.37199916-
0.14287349j
  0.13017056-0.25721586j -0.26850432-0.4464609j  -
0.40938846+0.25717305j
 -0.07847799+0.79497103j  0.66692064-0.53187394j  0.40599536-
0.73745271j
 -0.29908479-0.34741819j -0.47744649+1.3295624j
0.54567071+0.0786155j
 -0.13779104+0.55207729j -1.17925277+0.09347281j  0.41904078-
0.5981466j
 -0.07372395+0.68853932j -0.24155126+0.91862642j
0.17335328+1.0355726j
  0.17811539-0.90291906j -0.66611154-0.74669091j -0.09055579-
0.4691533j
  1.0541381 +0.39443993j  0.12966252+0.12285074j  0.12966252-
0.12285074j
  1.0541381 -0.39443993j -0.09055579+0.4691533j  -
0.66611154+0.74669091j
  0.17811539+0.90291906j  0.17335328-1.0355726j  -0.24155126-
0.91862642j
 -0.07372395-0.68853932j  0.41904078+0.5981466j  -1.17925277-
0.09347281j
 -0.13779104-0.55207729j  0.54567071-0.0786155j  -0.47744649-
1.3295624j
 -0.29908479+0.34741819j  0.40599536+0.73745271j
0.66692064+0.53187394j
 -0.07847799-0.79497103j -0.40938846-0.25717305j -
0.26850432+0.4464609j
  0.13017056+0.25721586j  0.37199916+0.14287349j -0.18786645-
1.24534932j
  0.1064606 +0.21247382j  0.47027342-0.18037459j -0.15491924-
0.02483643j
  0.35968294-0.24706741j -0.96973612-0.07575831j  0.01515471-
0.410188j
 -0.1822969 +0.22066812j -0.50470144-0.39692716j -0.71485016-
0.92471399j
  0.23681294-0.47512135j  0.66395267+0.40891592j
0.48290372+0.09713752j
  0.07072209+0.10329931j -0.47456363+0.59382704j -0.49828755-
0.48229839j
 -0.55088959-0.26594257j -0.51461367+0.00332457j
0.11429021+0.09916212j
 -0.31509526-0.03378537j -0.47692637-0.52942611j  0.061132
+0.37765355j
```

```
  0.13287684+0.90813988j  0.43773278+0.37692783j  0.35636489-
0.50022012j
 -0.10993834+0.4747736j  -0.60338256+0.33490204j
1.33565984+0.61877959j
  0.41361353+0.26197772j  0.39274447+0.40784161j -
0.13567047+0.44669198j
  0.38746463+0.28029435j -0.43505679+0.16119828j -
0.12679964+0.21325669j
  0.36890854+0.69071684j -0.29636679+0.10558235j -0.56211698-
0.23378354j
 -0.45063199-0.6453621j   0.27048524-0.21869354j  0.41812605-
0.23468222j
  0.64474035+0.37084578j  0.22442563+0.37292854j -0.40234691-
0.10251993j]

# Hitung Discrete Fourier Transform (DFT)
X4 = np.fft.fft(x4)
print(X4)
[ 2.10141504+0.00000000e+00j  0.2368845 +4.12183351e-01j
  0.70723192+1.62133244e+00j  0.10536078+2.97629183e-01j
 -0.28565927+1.41041215e-01j -0.71817049+9.64550149e-01j
  0.1651417 +2.82516236e-01j  0.83149602+1.55144426e-01j
 -0.21134427+5.75248503e-01j -0.23111325-1.69789108e-03j
  0.93364554+1.14273125e+00j  1.4502856 +8.88133882e-01j
  0.37133839+8.76977245e-02j  0.54262   -4.36385849e-01j
 -0.31864447+5.43694019e-01j  2.06612071-9.42402098e-01j
 -0.02415628-1.15037785e-01j -1.88195028-1.76174914e+00j
  1.00956915-9.28305370e-01j  1.38185633+4.30468999e-01j
  0.84757574-9.04367064e-02j  1.15767931-4.19833026e-01j
  1.04457244+4.10140955e-01j -0.64002249-4.74617962e-01j
  1.69222275-2.79209942e+00j  0.73545411-9.97281197e-02j
 -2.32710969+5.61789128e-03j -1.19910772-7.07531761e-01j
  1.15226075+1.13086510e+00j  0.60999721-8.37169823e-01j
 -0.20357034+2.31369166e-01j  0.28605068+1.68986166e+00j
  0.4979209 +4.01580832e-01j  0.26680585-5.70247969e-01j
 -0.58785581+2.21243712e+00j  0.71370891+5.58021972e-01j
  1.54666282-2.63997739e-01j -0.59713172+4.61647024e-01j
  0.42554365-1.69110122e-01j  1.73512489-1.22106461e+00j
  0.47864114-8.40591387e-01j -0.50853583-1.02549888e+00j
  1.36944062-1.12881323e+00j  1.09132549+1.55774476e+00j
  0.48025451-5.08204345e-01j  1.17738019-1.07934694e+00j
 -0.21621176+1.12784683e-01j -0.28826782+1.82234629e+00j
 -1.39909208+4.41859291e-01j -1.63920261+1.12045923e+00j
 -0.28026611+5.48368737e-01j  0.02893588-7.36031164e-01j
  0.66959646+1.50366255e+00j -0.31378493-1.68108045e+00j
  0.63348607+1.30356912e+00j -0.4246067 -1.52642386e-01j
  0.44746198-1.98482596e+00j  0.69057362-2.57604973e+00j
 -0.84215003-1.19442027e+00j -0.88737301-1.29464608e+00j
  0.01254817-9.10351558e-01j -0.77652979+5.85311364e-01j
  0.05904715-1.30851376e+00j  0.99271504+9.26114339e-01j
 -0.78401273+1.00340150e+00j -0.78401273-1.00340150e+00j
```

```
  0.99271504-9.26114339e-01j  0.05904715+1.30851376e+00j
 -0.77652979-5.85311364e-01j  0.01254817+9.10351558e-01j
 -0.88737301+1.29464608e+00j -0.84215003+1.19442027e+00j
  0.69057362+2.57604973e+00j  0.44746198+1.98482596e+00j
 -0.4246067 +1.52642386e-01j  0.63348607-1.30356912e+00j
 -0.31378493+1.68108045e+00j  0.66959646-1.50366255e+00j
  0.02893588+7.36031164e-01j -0.28026611-5.48368737e-01j
 -1.63920261-1.12045923e+00j -1.39909208-4.41859291e-01j
 -0.28826782-1.82234629e+00j -0.21621176-1.12784683e-01j
  1.17738019+1.07934694e+00j  0.48025451+5.08204345e-01j
  1.09132549-1.55774476e+00j  1.36944062+1.12881323e+00j
 -0.50853583+1.02549888e+00j  0.47864114+8.40591387e-01j
  1.73512489+1.22106461e+00j  0.42554365+1.69110122e-01j
 -0.59713172-4.61647024e-01j  1.54666282+2.63997739e-01j
  0.71370891-5.58021972e-01j -0.58785581-2.21243712e+00j
  0.26680585+5.70247969e-01j  0.4979209 -4.01580832e-01j
  0.28605068-1.68986166e+00j -0.20357034-2.31369166e-01j
  0.60999721+8.37169823e-01j  1.15226075-1.13086510e+00j
 -1.19910772+7.07531761e-01j -2.32710969-5.61789128e-03j
  0.73545411+9.97281197e-02j  1.69222275+2.79209942e+00j
 -0.64002249+4.74617962e-01j  1.04457244-4.10140955e-01j
  1.15767931+4.19833026e-01j  0.84757574+9.04367064e-02j
  1.38185633-4.30468999e-01j  1.00956915+9.28305370e-01j
 -1.88195028+1.76174914e+00j -0.02415628+1.15037785e-01j
  2.06612071+9.42402098e-01j -0.31864447-5.43694019e-01j
  0.54262   +4.36385849e-01j  0.37133839-8.76977245e-02j
  1.4502856 -8.88133882e-01j  0.93364554-1.14273125e+00j
 -0.23111325+1.69789108e-03j -0.21134427-5.75248503e-01j
  0.83149602-1.55144426e-01j  0.1651417 -2.82516236e-01j
 -0.71817049-9.64550149e-01j -0.28565927-1.41041215e-01j
  0.10536078-2.97629183e-01j  0.70723192-1.62133244e+00j
  0.2368845 -4.12183351e-01j]
```

```
# Hitung Discrete Fourier Transform (DFT)
X5 = np.fft.fft(x5)
print(X5)
```

```
[ 0.29457149+0.00000000e+00j  0.28264143+1.98614747e-01j
  0.20654234+1.87186916e-01j  0.17379952-3.10957676e-01j
 -0.12966735+1.81103471e-01j  0.11325505-2.99068094e-02j
 -0.04727311-2.61390180e-01j -0.06430084+1.00189299e-01j
  0.064168  +1.01849833e-01j -0.10076521+1.10798648e-01j
 -0.23147158-9.87353175e-02j -0.24020404+9.01863160e-02j
  0.07624522-3.34850561e-02j  0.01001794+3.82452662e-02j
  0.14946062+7.10194703e-02j  0.06629879-5.10215648e-01j
 -0.11039441-2.50751971e-01j -0.25616817+2.14119802e-01j
  0.14806122-2.90943240e-01j  0.16273735+1.83555892e-02j
  0.26795915-9.58705259e-02j -0.1144562 -3.37718693e-01j
 -0.31999757+1.39840565e-01j -0.30398676+1.54691795e-01j
 -0.28284323-3.30594887e-02j  0.07869374-8.74272062e-02j
  0.04239651-1.79988487e-01j -0.06678243+6.17458525e-02j
 -0.08925574-5.48649599e-02j  0.12933952+9.96471810e-02j
  0.00661354+3.52535604e-01j  0.11265969-1.80971957e-02j
```

```
-0.1599287 -2.27736201e-01j -0.38769607+1.14124340e-01j
-0.10232764-4.30632098e-02j -0.39447537+1.46668909e-04j
-0.22221782+8.27768596e-02j -0.02411966+3.49827976e-01j
 0.08608773-9.16983855e-02j -0.02558706+1.25603819e-01j
 0.06214995+2.83303349e-01j -0.20566211+1.13207164e-01j
 0.07729781+1.23228488e-01j -0.06493018-1.40518053e-01j
 0.04511156-1.11400971e-01j -0.27875242-1.48323688e-01j
-0.01911974+4.44345826e-02j -0.26768588+1.70931990e-01j
-0.20050614+4.30486649e-01j  0.13266482-2.46063177e-01j
 0.07867587-2.46865103e-01j -0.17925441-1.16466129e-01j
 0.52568834+5.00562385e-01j  0.1636913 +1.32835152e-01j
 0.20724846+1.41446835e-01j -0.00134543+7.40055901e-03j
 0.31853584-1.15272368e-01j -0.20722022-1.54870321e-01j
-0.17757481+1.53398407e-01j  0.15150905+3.31682526e-01j
 0.08362204-2.74233263e-01j -0.14845667+9.28588574e-02j
-0.2120911 +4.37132571e-01j  0.05855441+2.97480673e-01j
-0.30529726-4.85617102e-02j -0.30529726+4.85617102e-02j
 0.05855441-2.97480673e-01j -0.2120911 -4.37132571e-01j
-0.14845667-9.28588574e-02j  0.08362204+2.74233263e-01j
 0.15150905-3.31682526e-01j -0.17757481-1.53398407e-01j
-0.20722022+1.54870321e-01j  0.31853584+1.15272368e-01j
-0.00134543-7.40055901e-03j  0.20724846-1.41446835e-01j
 0.1636913 -1.32835152e-01j  0.52568834-5.00562385e-01j
-0.17925441+1.16466129e-01j  0.07867587+2.46865103e-01j
 0.13266482+2.46063177e-01j -0.20050614-4.30486649e-01j
-0.26768588-1.70931990e-01j -0.01911974-4.44345826e-02j
-0.27875242+1.48323688e-01j  0.04511156+1.11400971e-01j
-0.06493018+1.40518053e-01j  0.07729781-1.23228488e-01j
-0.20566211-1.13207164e-01j  0.06214995-2.83303349e-01j
-0.02558706-1.25603819e-01j  0.08608773+9.16983855e-02j
-0.02411966-3.49827976e-01j -0.22221782-8.27768596e-02j
-0.39447537-1.46668909e-04j -0.10232764+4.30632098e-02j
-0.38769607-1.14124340e-01j -0.1599287 +2.27736201e-01j
 0.11265969+1.80971957e-02j  0.00661354-3.52535604e-01j
 0.12933952-9.96471810e-02j -0.08925574+5.48649599e-02j
-0.06678243-6.17458525e-02j  0.04239651+1.79988487e-01j
 0.07869374+8.74272062e-02j -0.28284323+3.30594887e-02j
-0.30398676-1.54691795e-01j -0.31999757-1.39840565e-01j
-0.1144562 +3.37718693e-01j  0.26795915+9.58705259e-02j
 0.16273735-1.83555892e-02j  0.14806122+2.90943240e-01j
-0.25616817-2.14119802e-01j -0.11039441+2.50751971e-01j
 0.06629879+5.10215648e-01j  0.14946062-7.10194703e-02j
 0.01001794-3.82452662e-02j  0.07624522+3.34850561e-02j
-0.24020404-9.01863160e-02j -0.23147158+9.87353175e-02j
-0.10076521-1.10798648e-01j  0.064168  -1.01849833e-01j
-0.06430084-1.00189299e-01j -0.04727311+2.61390180e-01j
 0.11325505+2.99068094e-02j -0.12966735-1.81103471e-01j
 0.17379952+3.10957676e-01j  0.20654234-1.87186916e-01j
 0.28264143-1.98614747e-01j]

# Hitung Discrete Fourier Transform (DFT)
```

```
X6 = np.fft.fft(x6)
print(X6)
```

```
[-6.33888440e-02+0.00000000e+00j -1.15464241e-01-3.42669365e-03j
  1.52279786e-01+2.02599327e-01j  3.32434753e-01+1.39925376e-01j
 -1.01411595e-02+2.63307804e-04j -2.54975343e-01-2.04046756e-01j
 -3.79567481e-01-4.82800404e-02j -2.34466466e-01+1.51384077e-01j
  1.21050991e-03-2.51912827e-03j -1.05982400e-01+8.30507444e-02j
  1.88368909e-01+6.03744028e-02j -3.06051292e-01-1.63823178e-01j
  9.57640493e-03-6.06728853e-02j -8.86537409e-02+1.44977558e-01j
 -1.34357608e-01-9.64560214e-02j -1.53569933e-01-5.86267746e-02j
 -2.39193493e-01-1.01443160e-01j -2.00807342e-01+1.80827680e-01j
  2.25377689e-01-1.07147741e-01j  1.58929173e-01-7.77478338e-02j
  9.33369606e-02-4.34122988e-02j -6.79798931e-05-8.73835521e-02j
  1.97255819e-01+6.01053961e-02j -1.57453829e-01+4.99440166e-02j
 -1.44308074e-01+1.69178318e-01j -3.10742612e-02-7.96668248e-02j
  6.54375684e-02-1.17537799e-01j -2.88807326e-01-6.42264799e-02j
 -3.32394502e-02-1.86648552e-01j  6.02563387e-02+2.70130919e-01j
  2.32540370e-01+1.99412604e-01j  2.47646326e-03-8.36031629e-02j
 -1.05178031e-01-2.15571292e-01j -7.02510310e-02+1.62070884e-01j
  7.79206396e-02+3.99810364e-02j -1.48367541e-01-1.27070359e-01j
 -7.33964811e-04+8.91720807e-02j -2.62227639e-01+1.55465602e-01j
  7.69287043e-02+1.54877574e-01j -8.03138402e-02-9.05141969e-02j
 -2.80987261e-01-6.78780707e-02j -6.00047658e-02+1.42499679e-01j
  4.88122766e-02-6.14372798e-02j -6.18234101e-02-4.25399755e-02j
 -2.82766207e-02+2.36660606e-01j -1.65338322e-01-1.84573279e-01j
 -6.13838843e-02+4.39458401e-02j -2.48012637e-01+5.22865060e-02j
 -2.19188974e-02+1.57500537e-01j -9.82446683e-03+2.29509732e-02j
  2.00275537e-01-1.02356409e-01j  4.82459475e-03+1.33700736e-01j
  1.69408717e-01+2.36527676e-01j -5.45992719e-02+2.96967963e-01j
  4.99342625e-02+9.30243585e-02j  2.49878625e-01-8.78128341e-02j
  1.49100327e-01+6.09480129e-02j -2.08917569e-01-1.62339606e-01j
  5.96682262e-02+1.02746513e-01j  2.10706707e-02+2.70874748e-01j
 -1.17936587e-01-2.89156846e-02j  2.80345606e-02-1.60911861e-01j
 -4.86420573e-01+5.64631262e-02j -1.86391165e-02+4.00061537e-01j
  1.26991113e-01+1.93702941e-01j  1.26991113e-01-1.93702941e-01j
 -1.86391165e-02-4.00061537e-01j -4.86420573e-01-5.64631262e-02j
  2.80345606e-02+1.60911861e-01j -1.17936587e-01+2.89156846e-02j
  2.10706707e-02-2.70874748e-01j  5.96682262e-02-1.02746513e-01j
 -2.08917569e-01+1.62339606e-01j  1.49100327e-01-6.09480129e-02j
  2.49878625e-01+8.78128341e-02j  4.99342625e-02-9.30243585e-02j
 -5.45992719e-02-2.96967963e-01j  1.69408717e-01-2.36527676e-01j
  4.82459475e-03-1.33700736e-01j  2.00275537e-01+1.02356409e-01j
 -9.82446683e-03-2.29509732e-02j -2.19188974e-02-1.57500537e-01j
 -2.48012637e-01-5.22865060e-02j -6.13838843e-02-4.39458401e-02j
 -1.65338322e-01+1.84573279e-01j -2.82766207e-02-2.36660606e-01j
 -6.18234101e-02+4.25399755e-02j  4.88122766e-02+6.14372798e-02j
 -6.00047658e-02-1.42499679e-01j -2.80987261e-01+6.78780707e-02j
 -8.03138402e-02+9.05141969e-02j  7.69287043e-02-1.54877574e-01j
 -2.62227639e-01-1.55465602e-01j -7.33964811e-04-8.91720807e-02j
 -1.48367541e-01+1.27070359e-01j  7.79206396e-02-3.99810364e-02j
 -7.02510310e-02-1.62070884e-01j -1.05178031e-01+2.15571292e-01j
```

```
  2.47646326e-03+8.36031629e-02j  2.32540370e-01-1.99412604e-01j
  6.02563387e-02-2.70130919e-01j -3.32394502e-02+1.86648552e-01j
 -2.88807326e-01+6.42264799e-02j  6.54375684e-02+1.17537799e-01j
 -3.10742612e-02+7.96668248e-02j -1.44308074e-01-1.69178318e-01j
 -1.57453829e-01-4.99440166e-02j  1.97255819e-01-6.01053961e-02j
 -6.79798931e-05+8.73835521e-02j  9.33369606e-02+4.34122988e-02j
  1.58929173e-01+7.77478338e-02j  2.25377689e-01+1.07147741e-01j
 -2.00807342e-01-1.80827680e-01j -2.39193493e-01+1.01443160e-01j
 -1.53569933e-01+5.86267746e-02j -1.34357608e-01+9.64560214e-02j
 -8.86537409e-02-1.44977558e-01j  9.57640493e-03+6.06728853e-02j
 -3.06051292e-01+1.63823178e-01j  1.88368909e-01-6.03744028e-02j
 -1.05982400e-01-8.30507444e-02j  1.21050991e-03+2.51912827e-03j
 -2.34466466e-01-1.51384077e-01j -3.79567481e-01+4.82800404e-02j
 -2.54975343e-01+2.04046756e-01j -1.01411595e-02-2.63307804e-04j
  3.32434753e-01-1.39925376e-01j  1.52279786e-01-2.02599327e-01j
 -1.15464241e-01+3.42669365e-03j]
e1=sum(X1)/N
ze1=X1-e1
X1k=np.conj(X1)
ze1k=np.conj(ze1)
X1ot=ze1*ze1k
eX1=sum(X1ot)/N
z1=complex(round(eX1.real,12),round(eX1.imag,12))

e2=sum(X2)/N
ze2=X2-e2
X2k=np.conj(X2)
ze2k=np.conj(ze2)
X2ot=ze2*ze2k
eX2=sum(X2ot)/N
z2=complex(round(eX2.real,12),round(eX2.imag,12))

e3=sum(X3)/N
ze3=X3-e3
X3k=np.conj(X3)
ze3k=np.conj(ze3)
X3ot=ze3*ze3k
eX3=sum(X3ot)/N
z3=complex(round(eX3.real,12),round(eX3.imag,12))

e4=sum(X4)/N
ze4=X4-e4
X4k=np.conj(X4)
ze4k=np.conj(ze4)
```

```
X4ot=ze4*ze4k
eX4=sum(X4ot)/N
z4=complex(round(eX4.real,12),round(eX4.imag,12))

e5=sum(X5)/N
ze5=X5-e5
X5k=np.conj(X5)
ze5k=np.conj(ze5)
X5ot=ze5*ze5k
eX5=sum(X5ot)/N
z5=complex(round(eX5.real,12),round(eX5.imag,12))

e6=sum(X6)/N
ze6=X6-e6
X6k=np.conj(X6)
ze6k=np.conj(ze6)
X6ot=ze6*ze6k
eX6=sum(X6ot)/N
z6=complex(round(eX6.real,12),round(eX6.imag,12))

#membentuk matriks non diagonal
z7tot=ze1*ze2
ez7t=sum(z7tot)/N
z7=complex(round(ez7t.real,12),round(ez7t.imag,12))

z12tot=ze2*ze1
ez12t=sum(z12tot)/N
z12=complex(round(ez12t.real,12),round(ez12t.imag,12))

z8tot=ze1*ze3
ez8t=sum(z8tot)/N
z8=complex(round(ez8t.real,12),round(ez8t.imag,12))

z17tot=ze3*ze1
ez17t=sum(z17tot)/N
z17=complex(round(ez17t.real,12),round(ez17t.imag,12))

z9tot=ze1*ze4
ez9t=sum(z9tot)/N
z9=complex(round(ez9t.real,12),round(ez9t.imag,12))
```

```
z22tot=ze4*ze1
ez22t=sum(z22tot)/N
z22=complex(round(ez22t.real,12),round(ez22t.imag,12))

z10tot=ze1*ze5
ez10t=sum(z10tot)/N
z10=complex(round(ez10t.real,12),round(ez10t.imag,12))

z27tot=ze5*ze1
ez27t=sum(z27tot)/N
z27=complex(round(ez27t.real,12),round(ez27t.imag,12))

z11tot=ze1*ze6
ez11t=sum(z11tot)/N
z11=complex(round(ez11t.real,12),round(ez11t.imag,12))

z32tot=ze6*ze1
ez32t=sum(z32tot)/N
z32=complex(round(ez32t.real,12),round(ez32t.imag,12))

z13tot=ze2*ze3
ez13t=sum(z13tot)/N
z13=complex(round(ez13t.real,12),round(ez13t.imag,12))

z18tot=ze3*ze2
ez18t=sum(z18tot)/N
z18=complex(round(ez18t.real,12),round(ez18t.imag,12))

z14tot=ze2*ze4
ez14t=sum(z14tot)/N
z14=complex(round(ez14t.real,12),round(ez14t.imag,12))

z23tot=ze4*ze2
ez23t=sum(z23tot)/N
z23=complex(round(ez23t.real,12),round(ez23t.imag,12))

z15tot=ze2*ze5
ez15t=sum(z15tot)/N
z15=complex(round(ez15t.real,12),round(ez15t.imag,12))

z28tot=ze5*ze2
```

```
ez28t=sum(z28tot)/N
z28=complex(round(ez28t.real,12),round(ez28t.imag,12))

z16tot=ze2*ze6
ez16t=sum(z16tot)/N
z16=complex(round(ez16t.real,12),round(ez16t.imag,12))

z33tot=ze6*ze2
ez33t=sum(z33tot)/N
z33=complex(round(ez33t.real,12),round(ez33t.imag,12))

z19tot=ze3*ze4
ez19t=sum(z19tot)/N
z19=complex(round(ez19t.real,12),round(ez19t.imag,12))

z24tot=ze4*ze3
ez24t=sum(z24tot)/N
z24=complex(round(ez24t.real,12),round(ez24t.imag,12))

z20tot=ze3*ze5
ez20t=sum(z20tot)/N
z20=complex(round(ez20t.real,12),round(ez20t.imag,12))

z29tot=ze5*ze3
ez29t=sum(z29tot)/N
z29=complex(round(ez29t.real,12),round(ez29t.imag,12))

z21tot=ze3*ze6
ez21t=sum(z21tot)/N
z21=complex(round(ez21t.real,12),round(ez21t.imag,12))

z34tot=ze6*ze3
ez34t=sum(z34tot)/N
z34=complex(round(ez34t.real,12),round(ez34t.imag,12))

z25tot=ze4*ze5
ez25t=sum(z25tot)/N
z25=complex(round(ez25t.real,12),round(ez25t.imag,12))

z30tot=ze5*ze4
ez30t=sum(z30tot)/N
```

```
z30=complex(round(ez30t.real,12),round(ez30t.imag,12))

z26tot=ze4*ze6
ez26t=sum(z26tot)/N
z26=complex(round(ez26t.real,12),round(ez26t.imag,12))

z35tot=ze6*ze4
ez35t=sum(z35tot)/N
z35=complex(round(ez35t.real,12),round(ez35t.imag,12))

z31tot=ze5*ze6
ez31t=sum(z31tot)/N
z31=complex(round(ez31t.real,12),round(ez31t.imag,12))

z36tot=ze6*ze5
ez36t=sum(z36tot)/N
z36=complex(round(ez36t.real,12),round(ez36t.imag,12))

#matriks varians-kovarians dan menghitung invers
matrix=np.matrix([[z1,z7,z8,z9,z10,z11],[z12,z2,z13,z14,z15,z16],
[z17,z18,z3,z19,z20,z21],[z22,z23,z24,z4,z25,z26],[z27,z28,z29,z3
0,z5,z31],[z32,z33,z34,z35,z36,z6]])
invers=np.linalg.inv(matrix)
print(invers)
[[ 3.76296663e-01-0.j  8.65350638e-02+0.j -1.81472368e-02+0.j
   9.02864199e-02-0.j -9.80341992e-02+0.j -2.29069389e-01+0.j]
 [ 8.65350638e-02+0.j  2.54199284e+00-0.j  3.36309105e-01-0.j
   5.26005780e-02+0.j  1.11559249e+00+0.j  1.38273304e-02+0.j]
 [-1.81472368e-02+0.j  3.36309105e-01-0.j  1.93256242e+00-0.j
   4.62829824e-03+0.j  5.91267384e-01+0.j  4.39979489e-01+0.j]
 [ 9.02864199e-02-0.j  5.26005780e-02-0.j  4.62829824e-03+0.j
   5.41982830e-01-0.j -1.20579524e-02+0.j -2.65675411e-01+0.j]
 [-9.80341992e-02+0.j  1.11559249e+00+0.j  5.91267384e-01+0.j
  -1.20579524e-02+0.j  1.36013039e+01+0.j  1.19723609e+00+0.j]
 [-2.29069389e-01+0.j  1.38273304e-02+0.j  4.39979489e-01+0.j
  -2.65675411e-01+0.j  1.19723609e+00+0.j  2.16817676e+01+0.j]]
#membuat vector
vector = np.array([1, 1, 1, 1, 1, 1]).reshape(-1, 1)
vt=np.transpose(vector)
#membuat vektor Sigma^2
sig=np.array([z1,z2,z3,z4,z5,z6]).reshape(-1, 1)
# menghitung bobot mula mula lamda=0
iv = invers @ vector
vi = vt @ invers
```

```
viv = vi @ vector
imu = invers @ sig
vim = vi @ sig
ipv = vim / viv
bot = imu - (iv * ipv)
bobot1 = iv / viv
print("bobot minimum", bobot1)
bobot minimum [[0.00439592+0.j]
 [0.0876965 +0.j]
 [0.06950402+0.j]
 [0.00870788+0.j]
 [0.34672308+0.j]
 [0.4829726 +0.j]]
# Menghitung Bobot Complex
# Inisialisasi bobot awal
bobot = bobot1

# Menghitung dan mencetak total sum untuk bobot awal
total_sum = np.sum(bobot)
print(f"Inisial bobot: {bobot}")
print(f"Total sum: {total_sum}")
print("\n")

# Inisialisasi variabel untuk menyimpan bobot terakhir yang valid
bobot_optimal = None

# Iterasi untuk nilai t dari 0.01 hingga 0.09
for t in np.arange(0.01, 0.1, 0.01):
    # Menghitung bobot untuk nilai t saat ini
    bobot = bobot + (t * bot)


    # Pengecekan apakah ada elemen negatif dalam bobot
    if any((i < 0).any() for i in bobot):
        print("Ditemukan bobot minus. Iterasi dihentikan.")
        break

    # Menyimpan bobot saat ini sebagai bobot terakhir yang valid
    bobot_optimal = bobot

    # Menghitung dan mencetak total sum untuk bobot saat ini
    total_sum = np.sum(bobot)
    print(f"Bobot untuk t={t}: {bobot}")
```

```
    print(f"Total sum untuk t={t}: {total_sum}")
    print("\n")

# Menggunakan bobot terakhir yang valid sebagai hasil
hasil_akhir = bobot_optimal
print(f"Bobot akhir yang tidak negatif: {hasil_akhir}")
```

```
Inisial bobot: [[0.00439592+0.j]
 [0.0876965 +0.j]
 [0.06950402+0.j]
 [0.00870788+0.j]
 [0.34672308+0.j]
 [0.4829726 +0.j]]
Total sum: (0.9999999999999999+0j)


Bobot untuk t=0.01: [[0.01647736+0.j]
 [0.09815978+0.j]
 [0.07650369+0.j]
 [0.02119408+0.j]
 [0.33763054+0.j]
 [0.45003455+0.j]]
Total sum untuk t=0.01: (1+0j)


Bobot untuk t=0.02: [[0.04064025+0.j]
 [0.11908633+0.j]
 [0.09050301+0.j]
 [0.0461665 +0.j]
 [0.31944547+0.j]
 [0.38415844+0.j]]
Total sum untuk t=0.02: (1+0j)


Bobot untuk t=0.03: [[0.07688458+0.j]
 [0.15047616+0.j]
 [0.11150199+0.j]
 [0.08362512+0.j]
 [0.29216787+0.j]
 [0.28534428+0.j]]
Total sum untuk t=0.03: (1+0j)


Bobot untuk t=0.04: [[0.12521036+0.j]
 [0.19232927+0.j]
 [0.13950063+0.j]
 [0.13356994+0.j]
 [0.25579773+0.j]
 [0.15359206+0.j]]
Total sum untuk t=0.04: (0.9999999999999999+0j)
```

```
Ditemukan bobot minus. Iterasi dihentikan.
Bobot akhir yang tidak negatif: [[0.12521036+0.j]
 [0.19232927+0.j]
 [0.13950063+0.j]
 [0.13356994+0.j]
 [0.25579773+0.j]
 [0.15359206+0.j]]
# menghitung portofolio
z1w = x1 * hasil_akhir[0, 0]
z2w = x2 * hasil_akhir[1, 0]
z3w = x3 * hasil_akhir[2, 0]
z4w = x4 * hasil_akhir[3, 0]
z5w = x5 * hasil_akhir[4, 0]
z6w = x6 * hasil_akhir[5, 0]
zw=z1w+z2w+z3w+z4w+z5w+z6w
print(zw)
0      0.014082+0.000000j
1      0.034597+0.000000j
2      0.025655+0.000000j
3      0.016399+0.000000j
4     -0.060185+0.000000j
              ...
124    0.055560+0.000000j
125    0.023666+0.000000j
126   -0.006536+0.000000j
127   -0.012829+0.000000j
128    0.035853+0.000000j
Length: 129, dtype: complex128
# menghitung standar deviasi
std_dev_real1 = np.std(np.real(zw))
std_dev_imag1 = np.std(np.imag(zw))

print("Standar Deviasi pada Bagian Real:", std_dev_real1)
print("Standar Deviasi pada Bagian Imajiner:", std_dev_imag1)

# menghitung Value at Risk
VaR95=std_dev_real1*(1.65)*(1000000)
print("Nilai VaR 95%:", VaR95)
Standar Deviasi pada Bagian Real: 0.03506185096282436
Standar Deviasi pada Bagian Imajiner: 0.0
Nilai VaR 95%: 57852.05408866019
```