DAFTAR PUSTAKA

Anton, H., & Rorres, C. (2014). *Elementary Linear Algebra.* New York: John Wiley & Sons, Inc.

Bateman, D. N., Carroll, R., Pettie, J., Yamamoto, T., Elamin, M. E., Peart, L., Dow, M., Coyle, J., Cranfield, K. R., Hook, C., Sandilands, E. A., Veiraiah, A., Webb, D., Gray, A., Dargan, P. I., Wood, D. M., Thomas, S. H., Dear, J. W., & Eddleston, M. (2014). Effect of the UK's revised paracetamol poisoning management guidelines on admissions, adverse reactions and costs of treatment. *British journal of clinical pharmacology,* 78(3), 610–618. https://doi.org/10.1111/bcp.12362

Ben-Shachar, R., Chen, Y., Luo, S., Hartman, C., Reed, M., & Nijhout, H. F. (2012). The biochemistry of asetaminofen hepatotoxicity and rescue: a mathematical model. *Theoretical biology & medical modelling*, 9, 55. https://doi.org/10.1186/1742-4682-9-55.

Cairns, R., Brown, J. A., Wylie, C. E., Dawson, A. H., Isbister, G. K., & Buckley, N. A. (2019). Paracetamol Poisoning-Related Hospital Admissions and Deaths in Australia, 2004–2017. *Medical Journal of Australia*, 211(5), 218-223. https://doi.org/10.5694/mja2.50296

Campbell, & Reece. (2008). *Biology ed VIII.* Jakarta: Erlangga.

Craig, D. G. N., Bates, C. M., Davidson, J. S., Martin, K. G., Hayes, P. C., & Simpson, K. J. (2012). Staggered Overdose Pattern and Delay to Hospital Presentation are Associated with Adverse Outcomes Following Paracetamol- Induced Hepatotoxicity. *British Journal of Clinical Pharmacology*, 73(2), 285–294. https://doi.org/10.1111/j.1365-2125.2011.04067

Hinson, J. A., Roberts, D. W., & James, L. P. (2010). Mechanisms of asetaminofen-induced liver necrosis. *Handbook of experimental pharmacology*, (196), 369–405. https://doi.org/10.1007/978-3-642-00663-0_12

Ingalls, B. P. (2012). *Mathematical modeling in systems biology: an introduction*. MIT press.

Gilman, G. (2006). *In The Pharmacological Basic of Therapeutic.* Unites States of America: Mc Graw-Hill.

Hofmeyr, J., Hucka, M., Morohashi, M., & Kitano, H. (2001). Metabolic control analysis in a nutshell.

Kasbawati, Gunawan, A. Y., Hertadi, R., & Sidarto, K. A. (2015, March). Metabolic regulation and maximal reaction optimization in the central metabolism of a yeast cell. *In Symposium on Biomathematics (SYMOMATH 2014)* (1651) 1, 75-85. American Institute of Physics. https://doi.org/10.1063/1.4914436.

Katzung dkk. (2012). *Basic & Clinical Pharmacology.* San Francisco: MC Graw Hill.

Lauterburg, B.H., Adams, J.D. and Mitchell, J.R. (1984). Hepatic Glutathione Homeostasis in the Rat: Efflux Accounts for Glutathione Turnover. Hepatology, 4: 586-590. https://doi.org/10.1002/hep.1840040402.

Lenhart, S., & Workman, J. (2007). *Optimal Control Applied to Biological Models.* Francis: CRC Press.

Nagar, S., Walther, S.E., Blanchard, R.L. (2006). Sulfotransferase (SULT) 1A1 Polymorphic Variants *1, *2, and *3 Are Associated with Altered Enzymatic Activity, Cellular Phenotype, and Protein Degradation. Molecular Pharmacology, 69, 2084 - 2092.

Nayfeh, A. H. (2008). *Nonlinear Oscillations*. John Wiley & Sons.

Marlina, D. (2012). Pengaruh Pemberian Ekstrak Tempe Terhadap Kadar Ureum dan Kreatinin Ginjal Tikus Putih Jantan Galur Wistar (Rattus Norvegicus) Dengan Pemberian Paracetamol Dosis Toksik. *JPP (Jurnal Kesehatan Poltekkes Palembang), 11*(1), 115-123.

Mazaleuskaya, L. L., Sangkuhl, K., Thorn, C. F., FitzGerald, G. A., Altman, R. B., & Klein, T. E. (2015). PharmGKB summary: pathways of asetaminofen metabolism at the therapeutic versus toxic doses. *Pharmacogenetics and genomics*, *25*(8), 416–426.https://doi.org/10.1097/FPC.0000000000000150

Meiss, J. D. (2007). *Differential Dynamical Systems.* USA: SIAM.

Murray, J. (2001). *Mathematical Biology : I. An Introduction.* New York: Springer-Verlag.

Mutlib, A. E., Goosen, T. C., Bauman, J. N., Williams, J. A., Kulkarni, S., Kostrubsky, S. (2006). Kinetics of asetaminofen glucuronidation by UDP glucuronosyltransferases 1A1, 1A6, 1A9 and 2B15. Potential implications in asetaminofen-induced hepatotoxicity. Chemical research in toxicology, 19(5), 701–709. https://doi.org/10.1021/tx050317i.

Olsder, & Woude. (2003). *Mathematical System Theory. 2nd Edition.* Netherlands: Delft University of Technology.

Patten, C. J., Thomas, P. E., Guy, R. L., Lee, M., Gonzalez, F. J., Guengerich, F.P., Yang, C. S. (1993). Cytochrome P450 enzymes involved in asetaminofen activation by rat and human liver microsomes and their kinetics. Chemical research in toxicology, 6(4), 511–518. https://doi.org/10.1021/tx00034a019.

Pezzola, S., Antonini, G., Geroni, C., Beria, I., Colombo, M., Broggini, M., Marchini, S., Mongelli, N., Leboffe, L., MacArthur,R., Mozzi, A. F., Federici, G., Caccuri, A. M. (2010). Role of glutathione transferases in the mechanism of brostallicin activation. Biochemistry, 49(1), 226–235. https://doi.org/10.1021/bi901689s.

Palsson, B. (2006). *Systems Biology: Properties of Reconstructed Network.* Cambridge: Cambridge University Press.

Ramachandran, A., & Jaeschke, H. (2019). Asetaminofen Hepatotoxicity. *Seminars in liver disease*, *39*(2), 221–234. https://doi.org/10.1055/s-0039-1679919.

Reddyhoff, D., Ward, J., Williams, D., Regan, S., & Webb, S. (2015). Timescale analysis of a mathematical model of asetaminofen metabolism and toxicity. *Journal of theoretical biology*, *386*, 132–146. https://doi.org/10.1016/j.jtbi.2015.08.021.

Reith, D., Medlicott, N. J., Silva, R. K. D., Yang, L., Hickling, J., & Zacharias, M. (2009). Simultaneous Modelling of the Michaelis-Menten Kinetics of Paracetamol Sulphation and Glucuronidation. *Clinical and Experimental Pharmacology and Physiology*, 36, 35-42. https://doi.org/10.1111/j.1440-1681.2008.05029.

Reed, M. C., Thomas, R. L., Pavisic, J., James, S. J., Ulrich, C. M., & Nijhout, H. F. (2008). A mathematical model of glutation metabolism. *Theoretical biology & medical modelling*, *5*, 8. https://doi.org/10.1186/1742-4682-5-8

Remien, C. H., Adler, F. R., Waddoups, L., Box, T. D., & Sussman, N. L. (2012). Mathematical modeling of liver injury and dysfunction after asetaminofen overdose: early discrimination between survival and death. *Hepatology (Baltimore, Md.)*, *56*(2), 727–734. https://doi.org/10.1002/hep.25656.

Ross, S. L. (2004). *Differential Equations (III)*. John Wiley & Sons, Inc.

Subiono. (2013). *Sistem Linear dan Kontrol Optimal.* Surabaya: Institut Teknologi Sepuluh Nopember.

Sulfayanti dkk. (2014). Aplikasi Kontrol Optimal Pada Perubahan Perilaku Manusia. *Jurnal Matematika, Statistika, dan Komputasi*, 17-27.

Syahrizal dkk. (2020). *Metabolisme & Bioenergetika.* Banda Aceh: Syiah Kuala University Press.

Trinh, C. T., Wlaschin, A., & Srienc, F. (2009). Elementary mode analysis: a useful metabolic pathway analysis tool for characterizing cellular metabolism. *Applied microbiology and biotechnology*, *81*(5), 813–826. https://doi.org/10.1007/s00253-008-1770-1

Wahjuni, S. (2013). *Metabolisme Biokimia.* Bali: Udayana University Press.

Wen, X., Wang, J. S., Neuvonen, P. J., & Backman, J. T. (2002). Isoniazid is a mechanism-based inhibitor of cytochrome P450 1A2, 2A6, 2C19 and 3A4 isoforms in human liver microsomes. *European journal of clinical pharmacology*, *57*(11), 799–804. https://doi.org/10.1007/s00228-001-0396-3

Wilmana, & Gan. (2008). *Analgesik Antipiretik Analgesik Anti-Inflamasi Nonsteroid dan Obat Gangguan Sendi Lainnya.* Jakarta: Balai Penerbit FKUI.

LAMPIRAN

Lampiran 1. Solusi Numerik Titik Kesetimbangan

```
import numpy as np
from scipy.optimize import fsolve
from matplotlib import pyplot as plt
import scipy.linalg as linalg
from numpy.linalg import eig, inv
from scipy.integrate import odeint
```

```
#Model Metabolisme Asetaminopen
def ModelRotem(s,p):
    #Defines the differential equations for metabolism
system.

    #Arguments:
    #s : vector of the state variables:
    #s = [ s1, s2, s3, s4, s5, s6, s7, s8, s9]
    #t : time
    #p : vector of the parameters:
    #p = [k1, k2, k3, k4, k5, k6, k7, k8, k9, k10, Km1, Km2,
Km3, Km4, Km5, Km6, Km7, Km8, Km9, Km10, Km11, Ki7, Ki9, n,
m, eta, hmax, r, kal, kgl, ksl, kgsh, knqgl, bg, dg, P, d]
    s1, s2, s3, s4, s5, s6, s7, s8, s9= s
    k1, k2, k3, k4, k5, k6, k7, k8, k9, Km1, Km2, Km3, Km4,
Km5, Km6, Km7, Km8, Km9, Km10, Km11, Ki7, Ki9, n, m, eta,
hmax, r, kal, ksl, kgl, kslt, knl, kgsh, knqgl, bg, dg, P, d,
bs, ds, delta = p

    # Create f = (s1', s2', s3', s4', s5', s6', s7', s8',
s9'):
    ds1=-(kal*s1)
    ds2=(kal*s1)-
((((k1*s2)/(Km1+s2))+((k2*s2)/(Km2+s2))+((k3*s2)/(Km3+s2)))*(
1+(P*(s2**n)/(d**n+s2**n))))-
((k4*s2*s3)/(kslt+(Km4*s2)+(Km5*s3)+(s2*s3)))-
(((k5*(s2**m))/((Km6**m)+(s2**m)))+((k6*s2)/(Km7+(s2)*(1+s2/K
i7)))+((k7*s2)/(Km8+s2))+((k8*s2)/(Km9+(s2)*(1+s2/Ki9))))
    ds3=-((k4*s2*s3)/(kslt+(Km4*s2)+(Km5*s3)+(s2*s3)))-
(ds*s3)
    ds4=((k4*s2*s3)/(kslt+(Km4*s2)+(Km5*s3)+(s2*s3)))+bs-
(ksl*s4)
    ds5=(((k5*(s2**m))/((Km6**m)+(s2**m)))+((k6*s2)/(Km7+(s2)
*(1+s2/Ki7)))+((k7*s2)/(Km8+s2))+((k8*s2)/(Km9+(s2)*(1+s2/Ki9
))))-(kgl*s5)
    ds6=((((k1*s2)/(Km1+s2))+((k2*s2)/(Km2+s2))+((k3*s2)/(Km3
+s2)))*(1+(P*(s2**n)/(d**n+s2**n))))-
((k9*s6*s7)/(kgsh+(Km10*s6)+(Km11*s7)+(s6*s7)))-(eta*s6*s9)
    ds7=-((k9*s6*s7)/(kgsh+(Km10*s6)+(Km11*s7)+(s6*s7)))+bg-
(dg*s7)
```

```
    ds8=((k9*s6*s7)/(kgsh+(Km10*s6)+(Km11*s7)+(s6*s7)))-
(knqgl*s8)
    ds9=((r*s9)*(1-(s9)/hmax))-(eta*s6*s9)-(delta*s9)
    f = [ds1,ds2,ds3,ds4,ds5,ds6,ds7,ds8,ds9]
    return f


if __name__ == "__main__":
    k1= 0.55
    k2= 345
    k3= 0.99
    k4= 1785
    k5= 6370
    k6= 490
    k7= 4900
    k8= 8820
    k9= 72000
    Km1= 3430
    Km2= 677
    Km3= 276
    Km4= 97
    Km5= 0.0033
    Km6= 5500
    Km7= 4000
    Km8= 9200
    Km9= 23000
    Km10= 15
    Km11= 4600
    Ki7= 23000
    Ki9= 5300
    kal= 4
    ksl= 0.24
    kgl= 0.81
    kgsh= 0.001875
    knqgl= 0.3
    bg= 0.071
    dg= 0.083
    eta= 0.213*10**-4
    delta= 0.0833
    d= 18000
    P= 20
    hmax= (1.6)*10**11
    r= 0.0417
    n= 2
    m= 3
    bs= 0.138
    ds= 0.083
    knl= 0.001
    kslt= 13.27
```

```
    # Pack up the parameters and initial conditions:
    p = [k1, k2, k3, k4, k5, k6, k7, k8, k9, Km1, Km2, Km3,
Km4, Km5,Km6, Km7,
         Km8, Km9, Km10, Km11, Ki7, Ki9, n, m, eta, hmax, r,
kal, kgl, ksl, kgsh, knqgl, knl, kslt, bg, dg, P, d, ds, bs,
delta ]

    s0 = [7, 5, 1, 1, 0, 0, 0.5, 0, 0.2]

    # Find the equilibrium
    eq = fsolve(ModelRotem, s0, args=(p))
    print(eq)
```

Lampiran 2. Solusi Numerik Kestabilan Titik Kesetimbangan

```
import numpy as np
from scipy.optimize import fsolve
from matplotlib import pyplot as plt
import scipy.linalg as linalg
from numpy.linalg import eig, inv
from scipy.integrate import odeint
```

```
if __name__ == "__main__":
    k1= 0.55
    k2= 345
    k3= 0.99
    k4= 1785
    k5= 6370
    k6= 490
    k7= 4900
    k8= 8820
    k9= 72000
    Km1= 3430
    Km2= 677
    Km3= 276
    Km4= 97
    Km5= 0.0033
    Km6= 5500
    Km7= 4000
    Km8= 9200
    Km9= 23000
    Km10= 15
    Km11= 4600
    Ki7= 23000
    Ki9= 5300
    kal= 4
    ksl= 0.24
    kgl= 0.81
```

```
    kgsh= 0.001875
    knqgl= 0.3
    bg= 0.071
    dg= 0.083
    eta= 0.213*10**-4
    delta= 0.0833
    d= 18000
    P= 20
    hmax= (1.6)*10**11
    r= 0.0417
    n= 2
    m= 3
    bs= 0.138
    ds= 0.083
    knl= 0.001
    kslt= 13.27

    s1E= 5.09298567e-30
    s2E= 1.62814114e-17
    s3E= 1.83099313e-16
    s4E= 1.02469136e-01
    s5E= 7.04569259e-17
    s6E= 5.33986213e-19
    s7E= 8.55421687e-01
    s8E= 6.29844985e-19
    s9E= 4.40647615e-28

    a11= -kal
    a21= kal
    a22= (-(((k1/(Km1+s2E))-
((k1*s2E)/(Km1+s2E)**2))+((k2/(Km2+s2E))-
((k2*s2E)/(Km2+s2E)**2))+((k3/(Km3+s2E))-
((k3*s2E)/(Km3+s2E)**2)))*(1+(P*(s2E**n))/((d**n)+(s2E**n))))
-
((((k1*s2E)/(Km1+s2E))+((k2*s2E)/(Km2+s2E))+((k3*s2E)/(Km3+s2
E)))*(((P*(s2E**n)*n)/(s2E*((d**n)+(s2E**n))))-
((P*((s2E**n)**2)*n)/(((d**n)+(s2E**n))**2)*(s2E))))-
((k4*s3E)/((Km4*s2E)+(Km5*s3E)+(s2E*s3E)+kslt))+(((k4*s2E*s3E
)*(Km4+s3E))/((Km4*s2E)+(Km5*s3E)+(s2E*s3E)+kslt)**2)-
((k5*(s2E**m)*m)/(s2E*(Km6**m)+(s2E**m)))+((k5*((s2E**m)**2)*
m)/((((Km6**m)+(s2E**m))**2)*s2E))-
(k6/(Km7+((s2E)*(1+(s2E/Ki7)))))+((k6*(s2E)*(1+((2*s2E)/Ki7))
)/(Km7+((s2E)*(1+s2E/Ki7))**2))-
(k7/(Km8+s2E))+((k7*s2E)/((Km8+s2E)**2))-
(k8/(Km9+((s2E)*(1+(s2E/Ki9)))))+((k8*((s2E)*(1+(2*s2E/Ki9)))
)/(Km9+((s2E)*(1+(s2E/Ki9)))**2))
    a23= -
((k4*s2E)/((Km4*s2E)+(Km5*s3E)+(s2E*s3E)+kslt))+(((k4*s2E*s3E
)*(Km5+s2E))/((Km4*s2E)+(Km5*s3E)+(s2E*s3E)+kslt)**2)
    a32= -
((k4*s3E)/((Km4*s2E)+(Km5*s3E)+(s2E*s3E)+kslt))+(((k4*s2E*s3E
```

```
)*(Km4+s3E))/((Km4*s2E)+(Km5*s3E)+(s2E*s3E)+kslt)**2)
    a33= -
((k4*s2E)/((Km4*s2E)+(Km5*s3E)+(s2E*s3E)+kslt))+(((k4*s2E*s3E
)*(Km5+s2E))/((Km4*s2E)+(Km5*s3E)+(s2E*s3E)+kslt)**2)-ds
    a42= (((k4*s3E)/((Km4*s2E)+(Km5*s3E)+(s2E*s3E)+kslt))-
(((k4*s2E*s3E)*(Km4+s3E))/((Km4*s2E)+(Km5*s3E)+(s2E*s3E)+kslt
)**2))
    a43= (((k4*s2E)/((Km4*s2E)+(Km5*s3E)+(s2E*s3E)+kslt))-
(((k4*s2E*s3E)*(Km5+s2E))/((Km4*s2E)+(Km5*s3E)+(s2E*s3E)+kslt
)**2))
    a44= -ksl
    a52= ((k5*(s2E**m)*m)/(s2E*(Km6**m)+(s2E**m)))-
((k5*m*((s2E**m)**2))/((((Km6**m)+(s2E**m))**2)*s2E))-
(k6/(Km7+((s2E)*(1+(s2E/Ki7)))))+((k6*(s2E)*(1+((2*s2E)/Ki7))
)/(Km7+((s2E)*(1+s2E/Ki7))**2))-
(k7/(Km8+s2E))+((k7*s2E)/((Km8+s2E)**2))-
(k8/(Km9+((s2E)*(1+(s2E/Ki9)))))+((k8*(s2E)*(1+(2*s2E/Ki9)))/
(Km9+((s2E)*(1+(s2E/Ki9)))**2))
    a55= -kgl
    a62= (((k1/(Km1+s2E))-
((k1*s2E)/(Km1+s2E)**2))+((k2/(Km2+s2E))-
((k2*s2E)/(Km2+s2E)**2))+((k3/(Km3+s2E))-
((k3*s2E)/(Km3+s2E)**2))*(1+((P*(s2E**n))/((d**n)+(s2E**n))))
)+(((k1*s2E)/(Km1+s2E))+((k2*s2E)/(Km2+s2E))+((k3*s2E)/(Km3+s
2E)))*(((P*(s2E**n)*n)/(s2E*((d**n)+(s2E**n))))-
((P*n*((s2E**n)**2))/((((d**n)+(s2E**n))**2)*(s2E)))
    a66= -(((k9*s7E)/((Km10*s6E)+(Km11*s7E)+(s6E*s7E)+kgsh))-
(((k9*s6E*s7E)*(Km10+s7E))/((Km10*s6E)+(Km11*s7E)+(s6E*s7E)+k
gsh)**2))-eta*s9E
    a67= -(((k9*s6E)/((Km10*s6E)+(Km11*s7E)+(s6E*s7E)+kgsh))-
(((k9*s6E*s7E)*(Km11+s6E))/((Km10*s6E)+(Km11*s7E)+(s6E*s7E)+k
gsh)**2))
    a69= -eta*s6E
    a76= -(((k9*s7E)/((Km10*s6E)+(Km11*s7E)+(s6E*s7E)+kgsh))-
(((k9*s6E*s7E)*(Km10+s7E))/((Km10*s6E)+(Km11*s7E)+(s6E*s7E)+k
gsh)**2))
    a77= -(((k9*s6E)/((Km10*s6E)+(Km11*s7E)+(s6E*s7E)+kgsh))-
(((k9*s6E*s7E)*(Km11+s6E))/((Km10*s6E)+(Km11*s7E)+(s6E*s7E)+k
gsh)**2))-dg
    a86= (((k9*s7E)/((Km10*s6E)+(Km11*s7E)+(s6E*s7E)+kgsh))-
(((k9*s6E*s7E)*(Km10+s7E))/((Km10*s6E)+(Km11*s7E)+(s6E*s7E)+k
gsh)**2))
    a87= (((k9*s6E)/((Km10*s6E)+(Km11*s7E)+(s6E*s7E)+kgsh))-
(((k9*s6E*s7E)*(Km11+s6E))/((Km10*s6E)+(Km11*s7E)+(s6E*s7E)+k
gsh)**2))
    a88= -knqgl
    a96= -eta*s9E
    a99= (r*(1-((s9E)/hmax)))-((r*s9E)/hmax)-(eta*s6E)-
(delta)
    i1= -(a99+a77+a66+a33+a22)
    i2= -((-a22*a33)-(a22*a66)-(a22*a77)-(a22*a99)+(a23*a32)-
```

```
(a33*a66)-(a33*a77)-(a33*a99)-(a66*a77)-
(a66*a99)+((a67)**2)+(a69*a96)-(a77*a99))
    i3= -
((a22*a33*a66)+(a22*a33*a77)+(a22*a33*a99)+(a22*a66*a77)+(a22
*a66*a99)-(a22*((a67)**2))-(a22*a69*a96)+(a22*a77*a99)-
(a23*a32*a66)-(a23*a32*a77)-
(a23*a32*a99)+(a33*a66*a77)+(a33*a66*a99)-(a33*((a67)**2))-
(a33*a69*a96)+(a33*a77*a99)+(a66*a77*a99)-(((a67)**2)*a99)-
(a69*a77*a96))
    i4= -((-a22*a33*a66*a77)-
(a22*a33*a66*a99)+(a22*a33*((a67)**2))+(a22*a33*a69*a96)-
(a22*a33*a77*a99)-
(a22*a66*a77*a99)+(a22*((a67)**2)*a99)+(a22*a69*a77*a96)+(a23
*a32*a66*a77)+(a23*a32*a66*a99)-(a23*a32*((a67)**2))-
(a23*a32*a69*a96)+(a23*a32*a77*a99)-
(a33*a66*a77*a99)+(a33*((a67)**2)*a99)+(a33*a69*a77*a96))
    i5= -
(a22*a33*a66*a77*a99)+(a22*a33*((a67)**2)*a99)+(a22*a33*a69*a
77*a96)+(a23*a32*a66*a77*a99)-(a23*a32*((a67)**2)*a99)-
(a23*a32*a69*a77*a96)

Q1 = i1
Q2 = i1*i2-i3
Q3 = i1*i2*i3-i3**2+(i1**2)*i4
Q4 = i1*i2*i3*i4-(i1*i4)**2-(i3**2)*i4
Q5 = (i1*i2*i3*i4*i5)+(2*i1*i4*(i5)**2)+(i2*i3*(i5)**2)-
(((i1)**2)*((i4)**2)*i5)-(((i2)**2)*((i5)**2)*i1)-
(((i3)**2)*(i4)*i5)-i5**3

    J = np.array([[a11, 0, 0, 0, 0, 0, 0, 0, 0],
                  [a21, a22, a23, 0, 0, 0, 0, 0, 0],
                  [0, a32, a33, 0, 0, 0, 0, 0, 0],
                  [0, a42, a43, a44, 0, 0, 0, 0, 0],
                  [0, a52, 0, 0, a55, 0, 0, 0, 0],
                  [0, a62, 0, 0, 0, a66, a67, 0, a69],
                  [0, 0, 0, 0, 0, a76, a77, 0, 0],
                  [0, 0, 0, 0, 0, a86, a87, a88, 0],
                  [0, 0, 0, 0, 0, a96, 0, 0, a99]])
    print(J)
    print("eigenvalues : ", np.linalg.eigvals(J))
    print(Q1,Q2,Q3,Q4,Q5)
```

Lampiran 3. Solusi Numerik *Metabolic Control Analysis*

```
import numpy as np
from scipy.optimize import fsolve
from matplotlib import pyplot as plt
import scipy.linalg as linalg
from numpy.linalg import eig, inv
from scipy.integrate import odeint
```

```
if __name__ == "__main__":
    k1= 0.55
    k2= 345
    k3= 0.99
    k4= 1785
    k5= 6370
    k6= 490
    k7= 4900
    k8= 8820
    k9= 72000
    Km1= 3430
    Km2= 677
    Km3= 276
    Km4= 97
    Km5= 0.0033
    Km6= 5500
    Km7= 4000
    Km8= 9200
    Km9= 23000
    Km10= 15
    Km11= 4600
    Ki7= 23000
    Ki9= 5300
    kal= 4
    ksl= 0.24
    kgl= 0.81
    kgsh= 0.001875
    knqgl= 0.3
    bg= 0.071
    dg= 0.083
    eta= 0.213*10**-4
    delta= 0.0833
    d= 18000
    P= 20
    hmax= (1.6)*10**11
    r= 0.0417
    n= 2
    m= 3
    bs= 0.138
    ds= 0.083
    knl= 0.001
    kslt= 13.27

    s1E= 5.09298567e-30
    s2E= 1.62814114e-17
    s3E= 1.83099313e-16
    s4E= 1.02469136e-01
    s5E= 7.04569259e-17
    s6E= 5.33986213e-19
    s7E= 8.55421687e-01
    s8E= 6.29844985e-19
```

```
    s9E= 4.40647615e-28

    j1=
((((k1*s2E)/(Km1+s2E))+((k2*s2E)/(Km2+s2E))+((k3*s2E)/(Km3+s2
E)))*(1+(P*(s2E**n)/(d**n+s2E**n))))
    j2= ((k4*s2E*s3E)/(kslt+(Km4*s2E)+(Km5*s3E)+(s2E*s3E)))
    j3=
(((k5*(s2E**m))/((Km6**m)+(s2E**m)))+((k6*s2E)/(Km7+(s2E)*(1+
s2E/Ki7)))+((k7*s2E)/(Km8+s2E))+((k8*s2E)/(Km9+(s2E)*(1+s2E/K
i9))))
    j4= ((k9*s6E*s7E)/(kgsh+(Km10*s6E)+(Km11*s7E)+(s6E*s7E)))
    j5= kal*s1E
    j6= ds*s3E
    j7= bs
    j8= ksl*s4E
    j9= kgl*s5E
    j10= eta*s6E*s9E
    #j11= knl*s10E
    j11= dg*s7E
    j12= bg
    j13= knqgl*s8E

    a1= (s2E*(((k1)/(Km1+s2E))-
(k1*s2E)/((Km1+s2E)**2)+(k2)/(Km2+s2E)-
(k2*s2E)/((Km2+s2E)**2)+(k3)/(Km3+s2E)-
(k3*s2E)/((Km3+s2E)**2))*(1+(P*(s2E**n))/((d**n)+(s2E**n)))+(
(k1*s2E)/(Km1+s2E)+(k2*s2E)/(Km2+s2E)+(k3*s2E)/(Km3+s2E))*((P
*(s2E**n)* n)/(s2E*((d**n)+(s2E**n)))-
(P*((s2E**n)**2)*n)/((((d**n)+((s2E**n))**2)*s2E)))/(((k1*s2E)
/(Km1+s2E)+(k2*s2E)/(Km2+s2E)+(k3*s2E)/(Km3+s2E))*(1+(P*(s2E*
*n))/((d**n)+(s2E**n))))
    a2=
((Km4*s2E+Km5*s3E+s2E*s3E+kslt)*((k4*s3E)/(Km4*s2E+Km5*s3E+s2
E*s3E+kslt)-
(k4*s2E*s3E*(Km4+s3E))/((Km4*s2E+Km5*s3E+s2E*s3E+kslt)**2)))/
(k4*s3E)
    a3=
((Km4*s2E+Km5*s3E+s2E*s3E+kslt)*((k4*s2E)/(Km4*s2E+Km5*s3E+s2
E*s3E+kslt)-
(k4*s2E*s3E*(Km5+s2E))/((Km4*s2E+Km5*s3E+s2E*s3E+kslt)**2)))/
(k4*s2E)
    a4= (s2E*((k5*(s2E**n)*n)/(s2E*((Km6**n)+(s2E**n)))-
(k5*((s2E**n)**2)*n)/((((Km6**n)+((s2E**n))**2)*s2E)+(k6)/(Km7
+s2E*(1+(s2E)/(Ki7)))-
(k6*s2E*(1+(2*s2E)/(Ki7)))/((Km7+s2E*(1+(s2E)/(Ki7)))**2)+(k7
)/(Km8+s2E)-
(k7*s2E)/((Km8+s2E)**2)+(k8)/(Km9+s2E*(1+(s2E)/(Ki9)))-
(k8*s2E*(1+(2*s2E)/(Ki9)))/((Km9+s2E*(1+(s2E)/(Ki9)))**2)))/(
(k5*(s2E**n))/((Km6**n)+(s2E**n))+(k6*s2E)/(Km7+s2E*(1+(s2E)/
(Ki7)))+(k7*s2E)/(Km8+s2E)+(k8*s2E)/(Km9+s2E*(1+(s2E)/(Ki9)))
)
```

```
    a5=
((Km10*s6E+Km11*s7E+s6E*s7E+kgsh)*((k9*s7E)/(Km10*s6E+Km11*s7
E+s6E*s7E+kgsh)-
(k9*s6E*s7E*(Km10+s7E))/((Km10*s6E+Km11*s7E+s6E*s7E+kgsh)**2)
))/(k9*s7E)
    a6=
((Km10*s6E+Km11*s7E+s6E*s7E+kgsh)*((k9*s6E)/(Km10*s6E+Km11*s7
E+s6E*s7E+kgsh)-
(k9*s6E*s7E*(Km11+s6E))/((Km10*s6E+Km11*s7E+s6E*s7E+kgsh)**2)
))/(k9*s6E)
K_cantik = np.array([[1, 0, 0, 0, 0, 0],
                     [0, 0, 0, j4/j2, 0,j6/j2],
                     [j1/j3, -j2/j3, 0, 0, 0, 0],
                     [0, 1, 0, 0, 0, 0],
                     [0, 0, j3/j5, 0, 0, 0],
                     [0, 0, 0, 0, 0, j6/j6],
                     [j1/j7, -j2/j7, j3/j7, j4/j7, 0, j6/j7],
                     [0, 0, 1, 0, 0, 0],
                     [0, 0, 0, 1, 0, 0],
                     [0, 0, 0, 0, 1, 0],
                     [0, 0, 0, 0, 0, 1],
                     [j1/j12, 0, 0, 0, 0, 0],
                     [0, 0, 0, 0, j5/j13, j6/j13]])

Mepsilon = np.array([[a1, 0, 0, 0, 0, 0, 0],
                     [a2, a3, 0, 0, 0, 0, 0],
                     [a4, 0, 0, 0, 0, 0, 0],
                     [0, 0, 0, 0, a5, a6, 0],
                     [0, 0, 0, 0, 0, 0, 0],
                     [0, 1, 0, 0, 0, 0, 0],
                     [0, 0, 0, 0, 0, 0, 0],
                     [0, 0, 1, 0, 0, 0, 0],
                     [0, 0, 0, 1, 0, 0, 0],
                     [0, 0, 0, 0, 1, 0, 0],
                     [0, 0, 0, 0, 0, 1, 0],
                     [0, 0, 0, 0, 0, 0, 0],
                     [0, 0, 0, 0, 0, 0, 1]])
print(Mepsilon)
Neg_Mepsilon = -1*Mepsilon
matriks_E = np.hstack((K_cantik, Neg_Mepsilon))
Invers_E = np.linalg.inv(matriks_E)
Nol_1 = np.zeros((13,7))
Gab_1 = np.hstack((K_cantik,Nol_1))
identitas = np.identity(7)
Nol_2 = np.zeros((7,6))
Gab_2 = np.hstack((Nol_2,identitas))
I = np.vstack((Gab_1,Gab_2))

Hasil = np.dot(I,Invers_E)
print(Hasil)
```

Lampiran 4. Simulasi Numerik Model Metabolisme Asetaminofen Tanpa Kontrol

```
function[S1,S2,S3,S4,S5,S6,S7,S8,S9,tsol]=model1Normal()
k1= 0.55;
k2= 345;
k3= 0.99;
k4= 1785;
k5= 6370;
k6= 490;
k7= 4900;
k8= 8820;
k9= 72000;
Km1= 3430;
Km2= 677;
Km3= 276;
Km4= 97;
Km5= 0.0033;
Km6= 5500;
Km7= 4000;
Km8= 9200;
Km9= 23000;
Km10= 15;
Km11= 4600;
Ki7= 23000;
Ki9= 5300;
kal= 4;
ksl= 0.24;
kgl= 0.81;
kgsh= 0.001875;
knqgl= 0.3;
bg= 0.071;
dg= 0.083;
eta= 0.213*10^-4;
delta= 0.0833;
d= 1800;
P= 20;
hmax= (1.6)*10^11;
r= 0.0417;
n= 2;
m= 3;
bs= 0.138;
ds= 0.083;
kslt= 13.27;

f=@(t,x)[
    -(kal*x(1));
    (kal*x(1))-
((((k1*x(2))/(Km1+x(2)))+((k2*x(2))/(Km2+x(2)))+((k3*x(2))/(Km3+x(2
))))*(1+(P*(x(2)^n)/(d^n+x(2)^n))))-
```

```
((k4*x(2)*x(3))/(kslt+(Km4*x(2))+(Km5*x(3))+(x(2)*x(3))))-
(((k5*(x(2)^m))/((Km6^m)+(x(2)^m)))+((k6*x(2))/(Km7+(x(2))*(1+x(2)/
Ki7)))+((k7*x(2))/(Km8+x(2)))+((k8*x(2))/(Km9+(x(2))*(1+x(2)/Ki9)))
);
    -((k4*x(2)*x(3))/(kslt+(Km4*x(2))+(Km5*x(3))+(x(2)*x(3))))-
(ds*x(3));
    ((k4*x(2)*x(3))/(kslt+(Km4*x(2)+(Km5*x(3))+(x(2)*x(3)))))+bs-
(ksl*x(4));

(((k5*(x(2)^m))/((Km6^m)+(x(2)^m)))+((k6*x(2))/(Km7+(x(2))*(1+x(2)/
Ki7)))+((k7*x(2))/(Km8+x(2)))+((k8*x(2))/(Km9+(x(2))*(1+x(2)/Ki9)))
)-(kgl*x(5));

((((k1*x(2))/(Km1+x(2)))+((k2*x(2))/(Km2+x(2)))+((k3*x(2))/(Km3+x(2
)))*(1+(P*(x(2)^n)/(d^n+x(2^n)))))-
((k9*x(6)*x(7))/(kgsh+(Km10*x(6))+(Km11*x(7))+(x(6)*x(7)))))-
(eta*x(6)*x(9));
    -
((k9*x(6)*x(7))/(kgsh+(Km10*x(6))+(Km11*x(7))+(x(6)*x(7))))+bg-
(dg*x(7));
    ((k9*x(6)*x(7))/(kgsh+(Km10*x(6))+(Km11*x(7))+(x(6)*x(7))))-
(knqgl*x(8));
    ((r*(x(9)))*(1-(x(9))/hmax))-(eta*x(6)*x(9))-(delta*x(9));
    ];
 [tsol,xsol]=ode113(f,[0 0.3],[7;5;1;1;0;0;0.5;0;0.2]);
S1=xsol(:,1);
S2=xsol(:,2);
S3=xsol(:,3);
S4=xsol(:,4);
S5=xsol(:,5);
S6=xsol(:,6);
S7=xsol(:,7);
S8=xsol(:,8);
S9=xsol(:,9);
end
```

Plot

```
clear;
clc;
[S1,S2,S3,S4,S5,S6,S7,S8,S9,t]=model1Normal();

figure(1)
plot(t,S1,'b','LineWidth',0.5);
xlabel('Waktu (hari)');
ylabel ('APAP di Usus (S_1)');
figure(2)
plot(t,S2,'b','LineWidth',0.5);
xlabel('Waktu (hari)');
ylabel ('APAP di Hati (S_2)');
```

```
figure(3)
plot(t,S3,'b','LineWidth',0.5);
xlabel('Waktu (hari)');
ylabel ('PAPS di Hati (S_3)');
figure(4)
plot(t,S4,'b','LineWidth',0.5);
xlabel('Waktu (hari)');
ylabel ('APAP-S di Hati (S_4)');
figure(5)
plot(t,S5,'b','LineWidth',0.5);
xlabel('Waktu (hari)');
ylabel ('APAP-G di Hati (S_5)');
figure(6)
plot(t,S6,'b','LineWidth',0.5);
xlabel('Waktu (hari)');
ylabel ('NAPQI di Hati (S_6)');
figure(7)
plot(t,S7,'b','LineWidth',0.5);
xlabel('Waktu (hari)');
ylabel ('Antioksidan GSH di Hati (S_7)');
figure(8)
plot(t,S8,'b','LineWidth',0.5);
xlabel('Waktu (hari)');
ylabel ('NAPQI-GSH di Hati (S_8)');
figure(9)
plot(t,S9,'b','LineWidth',0.5);
xlabel('Waktu (hari)');
ylabel ('Sel hidup di Hati (S_9)');
```

Lampiran 5. Simulasi Numerik Model Metabolisme Asetaminofen Tanpa Kontrol dan dengan kontrol

State

```
function dx=naura_state2(t, x, u, Tu) %Tu adalah inputan %t,x,u
adalah variabel
global k1 k2 k3 k4 k5 k6 k7 k8 k9 Km1 Km2 Km3 Km4 Km5 Km6 Km7 Km8
Km9 Km10 Km11 Ki7 Ki9 kal ksl kgl kgsh knqgl bg dg eta delta d b
hmax r n m bs ds knl kslt
dx=zeros(9,1);
u1=u(1,:);
u1=interp1(Tu,u1',t);
u2=u(2,:);
u2=interp1(Tu,u2',t);
dx(1)=-(kal.*x(1));
dx(2)=(kal.*x(1))-((((k1.*x(2))./(Km1+x(2)))+(((1-
u1).*k2.*x(2))./(Km2+x(2)))+((k3.*x(2))./(Km3+x(2)))).*(1+(b.*(x(2)
.^n)./(d.^n+x(2).^n))))-
((k4.*x(2).*x(3))./(kslt+(Km4.*x(2))+(Km5.*x(3))+(x(2).*x(3))))-
```

```
(((k5.*(x(2).^m))./((Km6.^m)+(x(2).^m)))+((k6.*x(2))./(Km7+(x(2)).*
(1+x(2)./Ki7)))+((k7.*x(2))./(Km8+x(2)))+(((1-
u2).*k8.*x(2))./(Km9+(x(2)).*(1+x(2)./Ki9))));
dx(3)=-
((k4.*x(2).*x(3))./(kslt+(Km4.*x(2))+(Km5.*x(3))+(x(2).*x(3))))-
(ds.*x(3));
dx(4)=((k4.*x(2).*x(3))./(kslt+(Km4.*x(2)+(Km5.*x(3))+(x(2).*x(3)))
))+bs-(ksl.*x(4));
dx(5)=(((k5.*(x(2).^m))./((Km6.^m)+(x(2).^m)))+((k6.*x(2))./(Km7+(x
(2)).*(1+x(2)./Ki7)))+((k7.*x(2))./(Km8+x(2)))+(((1-
u2).*k8.*x(2))./(Km9+(x(2)).*(1+x(2)./Ki9))))-(kgl.*x(5));
dx(6)=((((k1.*x(2))./(Km1+x(2)))+(((1-
u1).*k2.*x(2))./(Km2+x(2)))+((k3.*x(2))./(Km3+x(2))).*(1+(b.*(x(2).
^n)./(d.^n+x(2.^n)))))-
((k9.*x(6).*x(7))./(kgsh+(Km10.*x(6))+(Km11.*x(7))+(x(6).*x(7)))))-
(eta.*x(6).*x(9));
dx(7)=-
((k9.*x(6).*x(7))./(kgsh+(Km10.*x(6))+(Km11.*x(7))+(x(6).*x(7))))+b
g-(dg.*x(7));
dx(8)=((k9.*x(6).*x(7))./(kgsh+(Km10.*x(6))+(Km11.*x(7))+(x(6).*x(7
))))-(knqgl.*x(8));
dx(9)=((r.*(x(9))).*(1-(x(9))./hmax))-(eta.*x(6).*x(9))-
(delta.*x(9));
end
```

Costate

```
function dp=naura_costate2(t, p, u, Tu, X)
global w1 k1 k2 k3 k4 k5 k6 k7 k8 k9 Km1 Km2 Km3 Km4 Km5 Km6 Km7
Km8 Km9 Km10 Km11 Ki7 Ki9 kal ksl kgl kgsh knqgl  dg eta delta d b
hmax r n m  ds knl kslt
% Backward
x2 = X(:, 2);
x3 = X(:, 3);
x6 = X(:, 6);
x7 = X(:, 7);
x9 = X(:, 9);
% Interpolasi variabel state
x2 = interp1(Tu, x2, t);
x3 = interp1(Tu, x3, t);
x6 = interp1(Tu, x6, t);
x7 = interp1(Tu, x7, t);
x9 = interp1(Tu, x9, t);
u1 = u(1,:);
u2 = u(2,:);
u1 = interp1(Tu,u1',t);
u2 = interp1(Tu,u2',t);
dp=zeros(9,1);
dp(1)=p(1)*kal+p(2)*kal;
```

```
dp(2)=-p(2)*((((-(k1/(Km1+x2))-((k1*x2)/(Km1+x2)^2)+(1-
u1)*k2/(Km2+x2)-((1-u1)*k2*x2)/(Km2+x2)^2+k3/(Km3+x2)-
(k3*x2)/(Km3+x2)^2)*(1+b*(x2^n)/((d^n)+(x2^n)))-
((k1*x2)/(Km1+x2))+(((1-
u1)*k2*x2)/(Km2+x2))+((k3*x2)/(Km3+x2)))*((b*x2^n*n/(x2*((d^n)+(x2^
n))))-(b*(x2^n)^2*n/(((d^n)+(x2^n))^2*x2)))-
(k4*x3/((Km4*x2)+(Km5*x3)+(x2*x3)+kslt))+(k4*x2*x3*(Km4+x3)/((Km4*x
2)+(Km5*x3)+(x2*x3)+kslt)^2))-((-
k5*x2^m*m/(x2*((Km6^m)+(x2^m))))+(k5*(x2^m)^2*m/(((Km6^m)+(x2^m))^2
*x2))-
(k6/(Km7+x2*(1+x2/Ki7)))+(k6*x2*(1+2*x2/Ki7)/(Km7+x2*((1+x2)/Ki7))^
2)-(k6/(Km8+x2))+((k6*x2)/(Km8+x2)^2)-((1-
u2).*k8/(Km9+x2*(1+x2/Ki9)))+(((1-
u2).*k8*x2)*(1+2*x2/Ki9)/(Km9+x2*(1+x2/Ki9))^2)))-p(3)*(-
(k4*x3/((Km4*x2)+(Km5*x3)+(x2*x3)+kslt))+(k4*x2*x3*(Km4+x3)/((Km4*x
2)+(Km5*x3)+(x2*x3)+kslt)^2))-
p(4)*((k4*x3/((Km4*x2)+(Km5*x3)+(x2*x3)+kslt))-
(k4*x2*x3*(Km4+x3)/((Km4*x2)+(Km5*x3)+(x2*x3)+kslt)^2))-p(5)*((-
k5*x2^m*m/(x2*((Km6^m)+(x2^m))))+(k5*(x2^m)^2*m/(((Km6^m)+(x2^m))^2
*x2))-
(k6/(Km7+x2*(1+x2/Ki7)))+(k6*x2*(1+2*x2/Ki7)/(Km7+x2*((1+x2)/Ki7))^
2)-(k7/(Km8+x2))+((k7*x2)/(Km8+x2)^2)-((1-
u2).*k8/(Km9+x2*(1+x2/Ki9)))+(((1-
u2).*k8*x2)*(1+2*x2/Ki9)/(Km9+x2*(1+x2/Ki9))^2))-
p(6)*((k1/(Km1+x2)-(k1*x2)/(Km1+x2)^2+(1-u1)*k2/(Km2+x2)-((1-
u1)*k2*x2)/(Km2+x2)^2+k3/(Km3+x2)-
(k3*x2)/(Km3+x2)^2)*(1+b*x2^n/((d^n)+(x2^n)))+((k1*x2)/(Km1+x2)+((1
-u1)*k2*x2)/(Km2+x2)+(k3*x2)/(Km3+x2))*(b*(x2^n)*n/(x2*(d^n+x2^n))-
b*(x2^n)^2*n/(((d^n)+(x2^n))^2*x2)));
dp(3)=-p(2)*((-
k4*x2/((Km4*x2)+(Km5*x3)+(x2*x3)+kslt))+(k4*x2*x3*(Km5+x2)/((Km4*x2
)+(Km5*x3)+(x2*x3)+kslt)^2))-p(3)*((-
k4*x2/((Km4*x2)+(Km5*x3)+(x2*x3)+kslt))+(k4*x2*x3*(Km5+x2)/((Km4*x2
)+(Km5*x3)+(x2*x3)+kslt)^2)-ds)-
p(4)*((k4*x2/((Km4*x2)+(Km5*x3)+(x2*x3)+kslt))-
(k4*x2*x3*(Km5+x2)/((Km4*x2)+(Km5*x3)+(x2*x3)+kslt)^2));
dp(4)=p(4)*ksl;
dp(5)=p(5)*kgl;
dp(6)=-p(6)*((-
k9*x7/((Km10*x6)+(Km11*x7)+(x6*x7)+kgsh))+(k9*x6*x7*(Km10+x7)/((Km1
0*x6)+(Km11*x7)+(x6*x7)+kgsh)^2)-eta*x9)-p(7)*((-
k9*x7/((Km10*x6)+(Km11*x7)+(x6*x7)+kgsh))+(k9*x6*x7*(Km10+x7)/((Km1
0*x6)+(Km11*x7)+(x6*x7)+kgsh)^2))-
p(8)*((k9*x7/((Km10*x6)+(Km11*x7)+(x6*x7)+kgsh))-
(k9*x6*x7*(Km10+x7)/((Km10*x6)+(Km11*x7)+(x6*x7)+kgsh)^2))+(p(9)*et
a*x6);
dp(7)=w1-p(6)*((-
k9*x6/((Km10*x6)+(Km11*x7)+(x6*x7)+kgsh))+(k9*x6*x7*(Km11+x6)/((Km1
0*x6)+(Km11*x7)+(x6*x7)+kgsh)^2))-p(7)*((-
```

```
k9*x6/((Km10*x6)+(Km11*x7)+(x6*x7)+kgsh))+(k9*x6*x7*(Km11+x6)/((Km1
0*x6)+(Km11*x7)+(x6*x7)+kgsh)^2-dg))-
p(8)*((k9*x6/((Km10*x6)+(Km11*x7)+(x6*x7)+kgsh))-
(k9*x6*x7*(Km11+x6)/((Km10*x6)+(Km11*x7)+(x6*x7)+kgsh)^2));
dp(8)=p(8)*knqgl;
dp(9)=-p(6)*(eta*x9)-p(9)*(r*((1-x9)/hmax)-((r*x9)/hmax)-(eta*x6)-
delta);
end
```

Fungsi Objektif

```
function J=naura_objektif2(X,u,Tu)
global w1 w2 w3
x7=X(:,7);
u1=u(1,:);
u2=u(2,:);
% J=w1.*x7;
% J=trapz(Tu,J);

% J=(w2/2)*u1'.^2;
% J=trapz(Tu,J);

% J=w3/2*u2'.^2;
% J=trapz(Tu,J);

J= -w1.*x7+(w2)*u1'-(w3)*u2';
J=trapz(Tu,J);
```

Kontrol

```
function u = naura_kontrol2(X,P,u)
global w2 w3 k2 k8 Km2 Km9 Ki9 kgsh n b d
u1=u(1,:);
u2=u(2,:);
x2 = X(:, 2)';
x6 = X(:, 6)';
x7 = X(:, 7)';
p2 = P(:, 2)';
p5 = P(:, 5)';
p6 = P(:, 6)';
p7 = P(:, 7)';
p8 = P(:, 8)';
dH1= w2.*u1+p2.*k2.*x2.*(1+b.*(x2.^n)./((d.^n)+(x2.^n)))./(Km2+x2)-
p6.*k2.*x2.*(1+b.*(x2.^n)./((d.^n)+(x2.^n)))./(Km2+x2);
dH2= -w3.*u2-p2.*k8.*x2./(Km9+x2.*(1+x2./Ki9))
p5.*k8.*x2./(Km9+x2.*(1+x2./Ki9));
u=[dH1;dH2];
end
```

Batas

```
function s=naura_f_simplebounds2(s,Lb,Ub)
 % untuk batas bawah
```

```
 ns_tmp=s;
 Z=ns_tmp<Lb;
 ns_tmp(Z)=Lb(Z);
 % untuk batas atas
 J=ns_tmp>Ub;
 ns_tmp(J)=Ub(J);
 % Update u
 s=ns_tmp;
end
```

Plot

```
clear all;
close all;
clc;
format long;
global w1 w2 w3 k1 k2 k3 k4 k5 k6 k7 k8 k9 Km1 Km2 Km3 Km4 Km5 Km6
Km7 Km8 Km9 Km10 Km11 Ki7 Ki9 kal ksl kgl kgsh knqgl bg dg eta
delta d b hmax r n m bs ds knl kslt
ptf =[0; 0; 0; 0; 0; 0; 0; 0; 0];
x0 = [7; 5; 1; 1; 0; 0; 0.5; 0; 0.2];

%Nilai parameter model
k1= 0.55;k2= 345;k3= 0.99;k4= 1785;k5= 6370;k6= 490;k7= 4900;k8=
8820;k9= 72000;Km1= 3430;Km2= 677;Km3= 276;Km4= 97;Km5= 0.0033;Km6=
5500;Km7= 4000;Km8= 9200;Km9= 23000;
Km10= 15;Km11= 4600;Ki7= 23000;Ki9= 5300;kal= 4;ksl= 0.24;kgl=
0.81;kgsh= 0.001875;knqgl= 0.3;bg= 0.071;dg= 0.083;
eta= 0.213*10^-4;delta= 0.0833;d= 1800;b= 20;hmax= (1.6)*10^11;r=
0.0417;n= 2;m= 3;bs= 0.138;ds= 0.083;knl= 0.001;kslt= 13.27;
w1 = 0.4; %Bobot Kontrol U
w2 = 0.3; %Bobot Kontrol w1
w3 = 0.3; %Bobot Kontrol w2
% time span:
Nt=1000;
tf =0.3; % Time akhir(Time proporsional)
Tu=linspace(0,tf,Nt);
%batas kontrol: v
M1=0.2;
M2=0.8;
nv=2; % jumlah kontrol
Lb=M1.*ones(nv,Nt); % matrix batas bawah
Ub=M2.*ones(nv,Nt); % matrix batas atas
% parameter SD:
eps=0.6;
kmax=5;
ki=1;
% ----------------------------------------------------------
u=zeros(nv,Nt);
%options = odeset('AbsTol',1e-2,'RelTol',1e-2);
```

```
[Tx, X] = ode113(@(t,x) naura_state2(t, x, u, Tu), Tu, x0);
% ----------------------------------------------------------
H_norm_lama=inf;
langkah(1)=10000;
for it = 1:10
 % Forward
 %options = odeset('AbsTol',1e-2,'RelTol',1e-2);
 [Tx1, X1] = ode113(@(t,x) naura_state2(t, x, u, Tu), Tu, x0);
 %options = odeset('AbsTol',1e-2,'RelTol',1e-2);
 [Tp, P] = ode113(@(t, p) naura_costate2(t, p, u, Tu, X1),
fliplr(Tu), ptf);

 % Menghitung gradien H
 dH = naura_kontrol2(X1, P, u);
 H_norm=norm(dH,2);
 % Menghitung fungsi objektif
 J = naura_objektif2(X1, u, Tu);

 if H_norm < eps
 % Nilai fungsi objektif
 Jopt=J;
 Uopt=u;
 break;
 else
 grad=dH/norm(dH,2);
 langkah(it)=15000;
 newu=u-langkah(ki)*grad;
 newu=naura_f_simplebounds2(newu,Lb,Ub);
 [Tx1, X1] = ode113(@(t,x) naura_state2(t, x, newu, Tu), Tu, x0);

 Jbaru= naura_objektif2(X1, newu, Tu);
 eror(ki)=abs(Jbaru-J);

 j=1;
 while eror(ki)>eps
 j=j+1;
 langkah(ki)=langkah(ki)*(0.5);
 newu=u-langkah(ki)*dH;
 newu=naura_f_simplebounds2(newu,Lb,Ub);
 [Tx1, X1] = ode113(@(t,x) naura_state2(t, x, newu, Tu), Tu, x0);
 Jbaru= naura_objektif2(X1, newu, Tu);
 eror(ki)=abs(Jbaru-J);
 disp(['eror: ',num2str(eror(ki))])

 if j>30
 break
 end
 end
 u=newu;
```

```
 u=naura_f_simplebounds2(u,Lb,Ub);
 disp(['norm(dH,2): ',num2str(norm(dH,2))])
 end
 eror(ki)=abs(Jbaru-J);
end
disp(J);
 [Tx1, X1] = ode113(@(t,x) naura_state2(t, x, u, Tu), Tu, x0);
figure(1)
plot(Tx, X(:, 1),'-r','LineWidth',0.5);
hold on
plot(Tx1, X1(:, 1),'--b','LineWidth',0.5);
xlabel('Waktu (hari)') ;
ylabel ('APAP di Usus (S_1)');
legend('Tanpa kontrol','Dengan kontrol');
set (gcf,'color','w')

figure(2)
plot(Tx, X(:, 2),'-r', 'LineWidth',0.5);
hold on
plot(Tx1, X1(:, 2),'--b', 'LineWidth',0.5);
xlabel('Waktu (hari)') ;
ylabel ('APAP di Hati (S_2)');
legend('Tanpa kontrol','Dengan kontrol');
set (gcf,'color','w')

figure(3)
plot(Tx, X(:, 3),'-r', 'LineWidth',0.5);
hold on
plot(Tx1, X1(:, 3),'--b', 'LineWidth',0.5);
xlabel('Waktu (hari)') ;
ylabel ('PAPS di Hati (S_3)');
legend('Tanpa kontrol','Dengan kontrol');
set (gcf,'color','w')

figure(4);
plot(Tx, X(:,4),'-r', 'LineWidth', 0.5);
hold on
plot(Tx1, X1(:,4),'--b', 'LineWidth',0.5);
xlabel('Waktu (hari)') ;
ylabel ('APAP-S di Hati (S_4)');
legend('Tanpa kontrol','Dengan kontrol');
set (gcf,'color','w')

figure(5);
plot(Tx, X(:,5),'-r', 'LineWidth', 0.5);
hold on
plot(Tx1, X1(:,5),'--b', 'LineWidth',0.5);
xlabel('Waktu (hari)') ;
ylabel ('APAP-G di Hati (S_5)');
```

```
legend('Tanpa kontrol','Dengan kontrol');
set (gcf,'color','w')

figure(6);
plot(Tx, X(:,6),'-r', 'LineWidth', 0.5);
hold on
plot(Tx1, X1(:,6),'--b', 'LineWidth',0.5);
xlabel('Waktu (hari)') ;
ylabel ('NAPQI di Hati (S_6)');
legend('Tanpa kontrol','Dengan kontrol');
set (gcf,'color','w')

figure(7);
plot(Tx, X(:,7),'-r', 'LineWidth', 0.5);
hold on
plot(Tx1, X1(:,7),'--b', 'LineWidth',0.5);
xlabel('Waktu (hari)') ;
ylabel ('Antioksidan GSH di Hati (S_7)');
legend('Tanpa kontrol','Dengan kontrol');
set (gcf,'color','w')

figure(8);
plot(Tx, X(:,8),'-r', 'LineWidth', 0.5);
hold on
plot(Tx1, X1(:,8),'--b', 'LineWidth',0.5);
xlabel('Waktu (hari)') ;
ylabel ('NAPQI-GSH di Hati (S_8)');
legend('Tanpa kontrol','Dengan kontrol');
set (gcf,'color','w')

figure(9);
plot(Tx, X(:,9),'-r', 'LineWidth', 0.5);
hold on
plot(Tx1, X1(:,9),'--b', 'LineWidth',0.5);
xlabel('Waktu (hari)') ;
ylabel ('Sel hidup di Hati (S_9)');
legend('Tanpa kontrol','Dengan kontrol');
set (gcf,'color','w')

figure(10)
plot(Tu, u(1,:),'-r', 'LineWidth',0.5);
hold on
plot(Tu, u(2,:),'--b', 'LineWidth',0.5);
xlabel('Waktu (hari)') ;
ylabel('u_i^*(t)');
%ylim([0 1]);
legend('u_1^*(t)', 'u_2^*(t)', 'Orientation', 'horizontal');
set (gcf,'color','w')
```