#Simulasi Model Matematika untuk Mempelajari Dinamika Penangkapan Karbon Pada Tumbuhan dalam Hutan

kondisi awal :

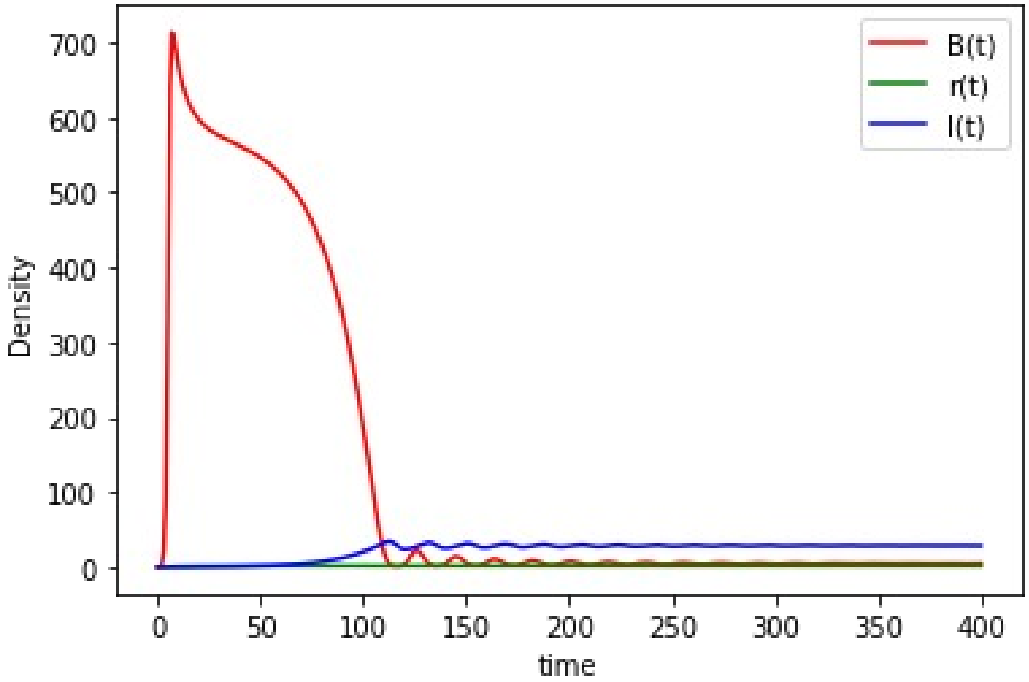
- B(0)=0.4
- r(0)=0.2
- I(0)=0.2

nilai parameter :

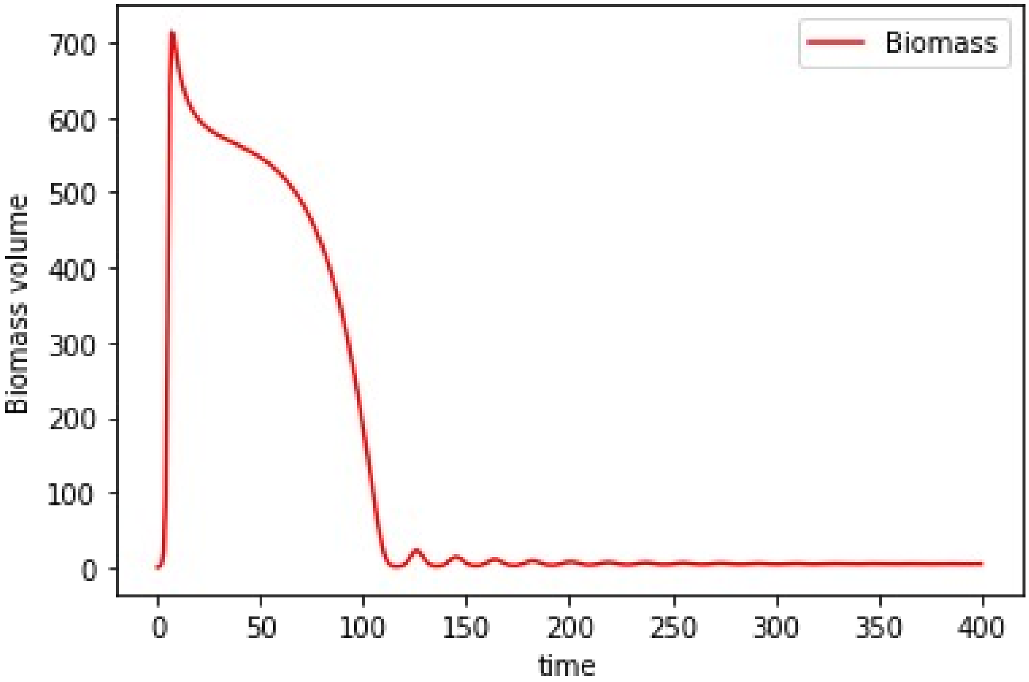
- K=400
- μ1=0.1
- h1=0.9
- r0=0.2
- ρ=0.1
- μ2=0.3
- h2=0.25
- α=0.5
- T=400

```
import numpy as np
from scipy.integrate import odeint
import matplotlib.pyplot as plt

# function that returns dz/dt
def model(z,t):
    K=400
    m1=0.1
    h1=0.9
    r0=0.2
    p=0.1
    m2=0.3
    h2=0.25
    alpha=0.5
    B= z[0]
    r= z[1]
    I= z[2]
    dBdt = (r*B)*(1-(B/K))+h1*B-(m1*I*B)
    drdt = r0-p*r
    dIdt = (m2*I*(B/(1+B)))-h2*I
    dzdt = [dBdt,drdt,dIdt]
    return dzdt

# initial condition
z0 = [0.4,0.2,0.2]

# number of time points
n = 401
```

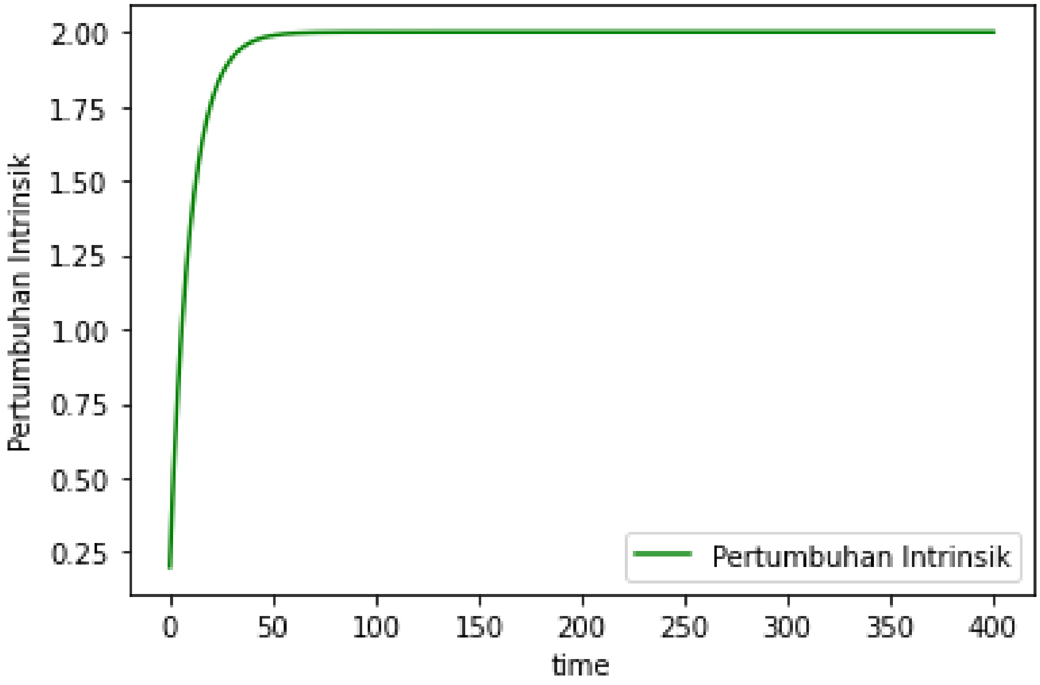
```python
# time points
t = np.linspace(0,400,n)

# store solution
B = np.empty_like(t)
r = np.empty_like(t)
I = np.empty_like(t)

# record initial conditions
B[0] = z0[0]
r[0] = z0[1]
I[0] = z0[2]

# solve ODE
for i in range(1,n):
    # span for next time step
    tspan = [t[i-1],t[i]]
    # solve for next step
    z = odeint(model,z0,tspan)
    # store solution for plotting
    B[i] = z[1][0]
    r[i] = z[1][1]
    I[i] = z[1][2]

    # next initial condition
    z0 = z[1]

# plot results
plt.plot(t,B,'r-',label='B(t)')
plt.plot(t,r,'g-',label='r(t)')
plt.plot(t,I,'b-',label='I(t)')
plt.ylabel('Density')
plt.xlabel('time')
plt.legend(loc='best')
plt.show()
```



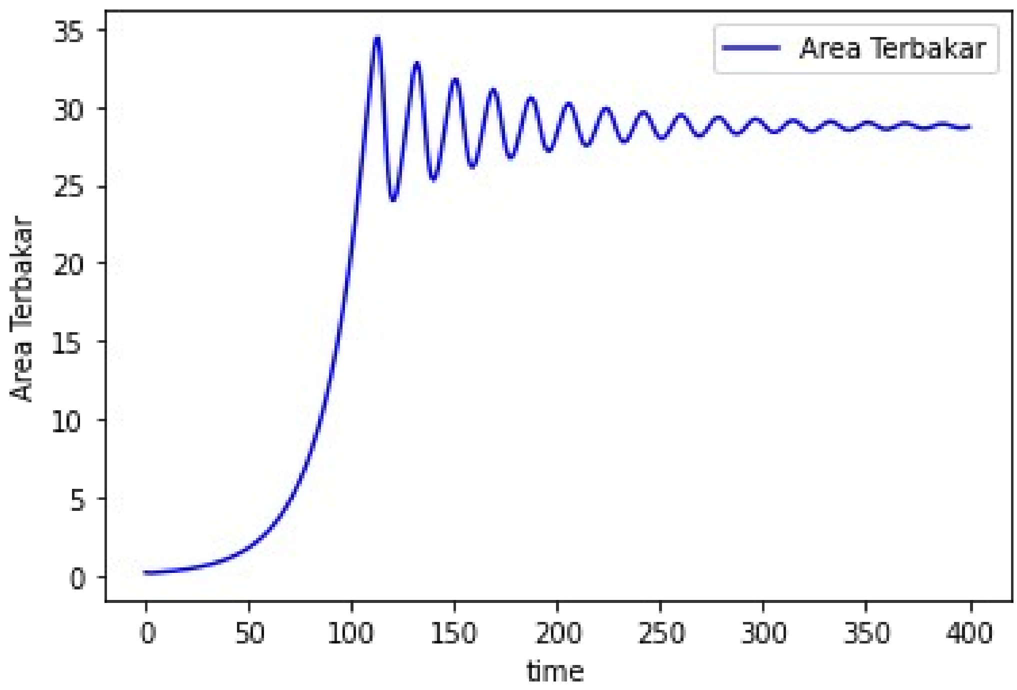
```
# plot results
plt.plot(t,B,'r-',label='Biomass')
plt.ylabel('Biomass volume')
plt.xlabel('time')
plt.legend(loc='best')
plt.show()
```



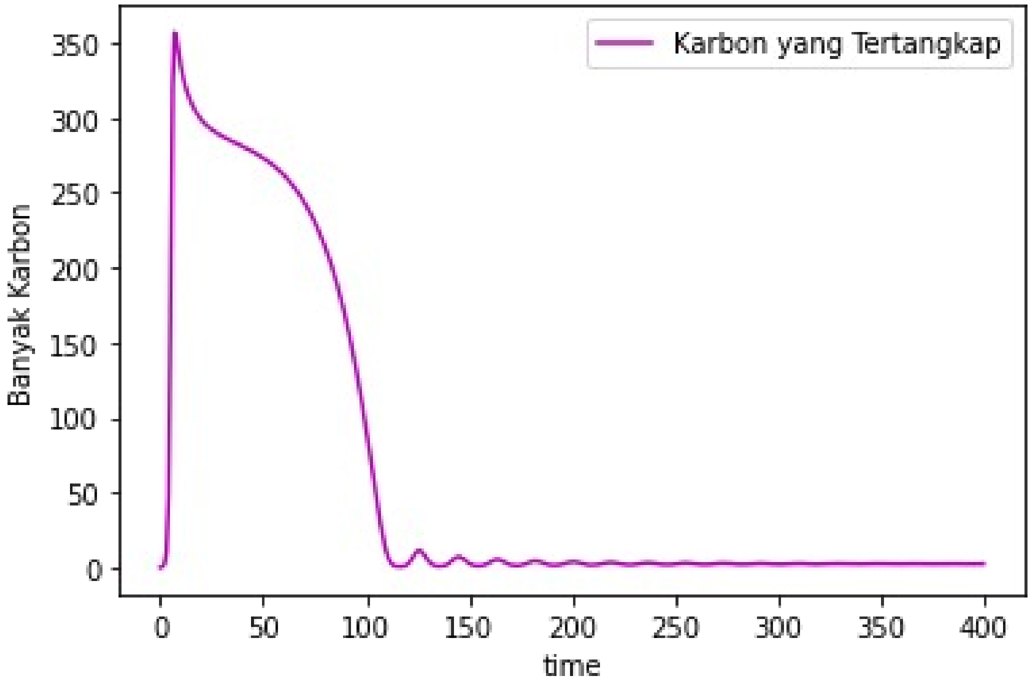
```
# plot results
plt.plot(t,r,'g-',label='Pertumbuhan Intrinsik')
plt.ylabel('Pertumbuhan Intrinsik')
plt.xlabel('time')
plt.legend(loc='best')
plt.show()
```



```
# plot results
plt.plot(t,I,'b-',label='Area Terbakar')
plt.ylabel('Area Terbakar')
plt.xlabel('time')
plt.legend(loc='best')
plt.show()
```



```
# plot results
alpha = 0.5
C = B*alpha
plt.plot(t,C,'m-',label='Karbon yang Tertangkap')
plt.ylabel('Banyak Karbon')
plt.xlabel('time')
plt.legend(loc='best')
plt.show()
```



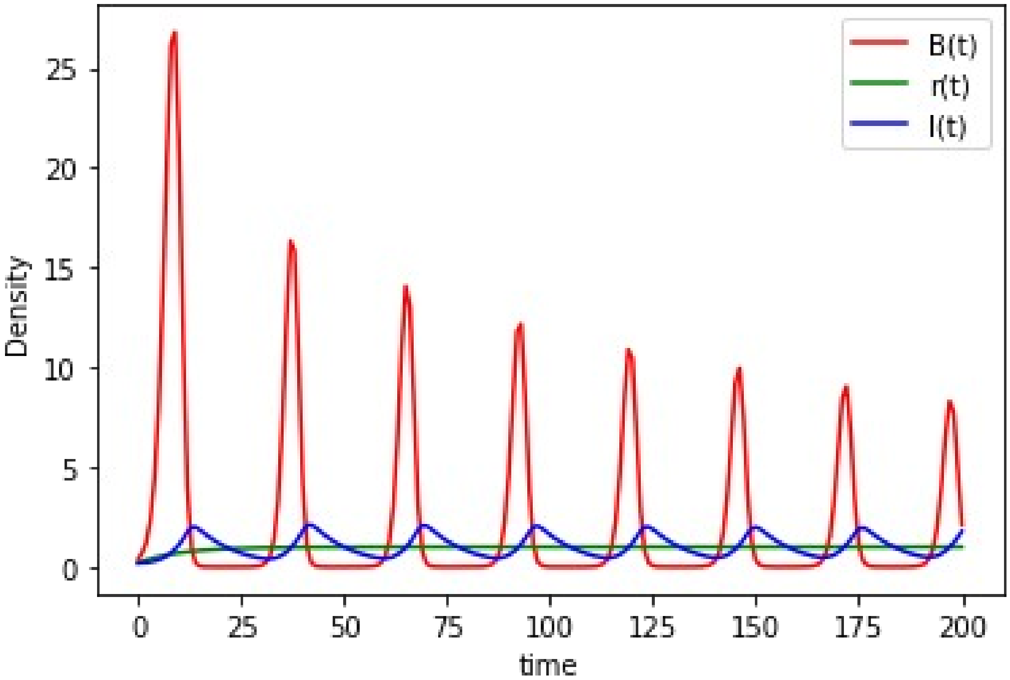
kondisi awal :

- B(0)=0.4
- r(0)=0.2
- I(0)=0.2

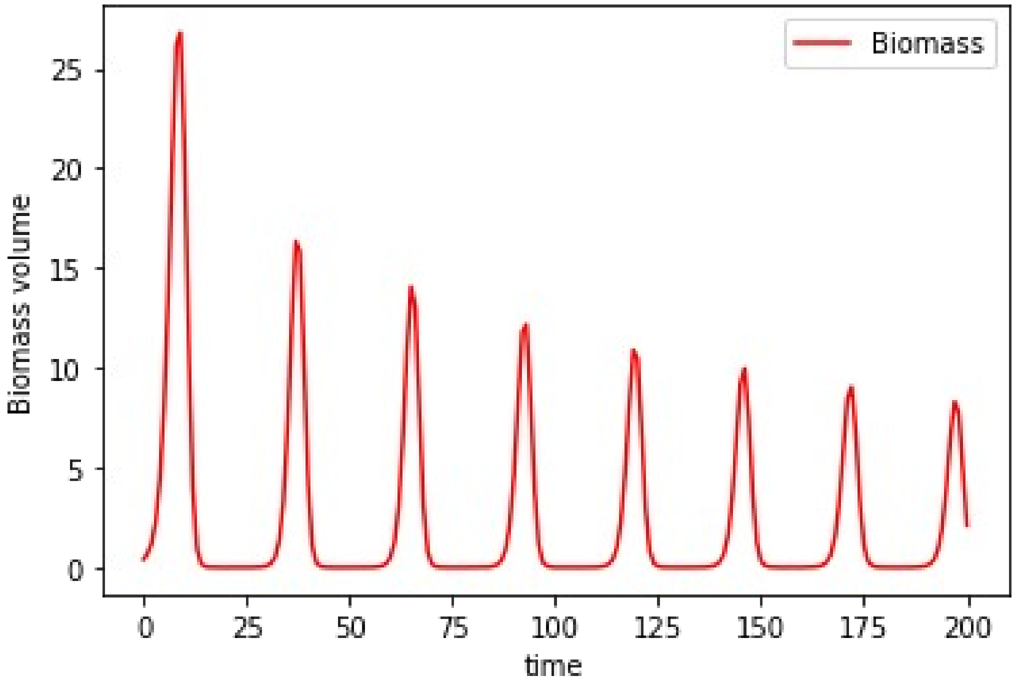
nilai parameter :

- K=400
- µ1=1.5
- h1=0.6
- r0=0.1
- ρ=0.1
- µ2=0.35
- h2=0.1
- α=0.5
- T=200

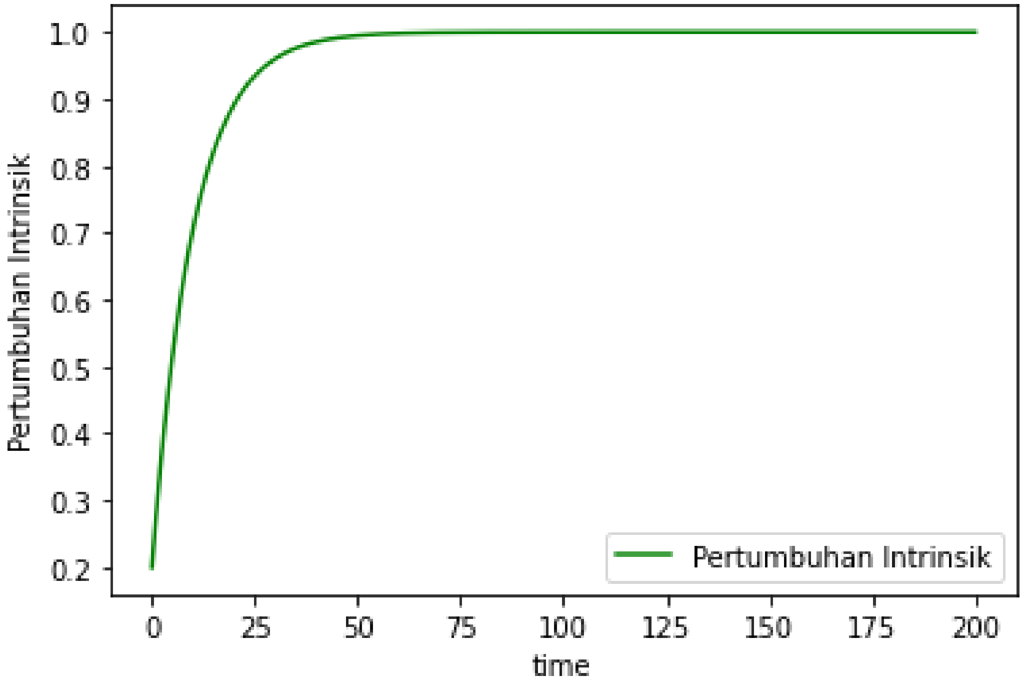
```
# function that returns dz/dt
def model(z,t):
    K=400
    m1=1.5
    h1=0.6
    r0=0.1
    p=0.1
    m2=0.35
    h2=0.1
    alpha=0.5
    B= z[0]
    r= z[1]
    I= z[2]
    dBdt = (r*B)*(1-(B/K))+h1*B-(m1*I*B)
    drdt = r0-p*r
    dIdt = (m2*I*(B/(1+B)))-h2*I
    dzdt = [dBdt,drdt,dIdt]
    return dzdt

# initial condition
z0 = [0.4,0.2,0.2]

# number of time points
n = 201

# time points
t = np.linspace(0,200,n)

# store solution
B = np.empty_like(t)
r = np.empty_like(t)
I = np.empty_like(t)

# record initial conditions
B[0] = z0[0]
r[0] = z0[1]
I[0] = z0[2]

# solve ODE
for i in range(1,n):
    # span for next time step
    tspan = [t[i-1],t[i]]
    # solve for next step
    z = odeint(model,z0,tspan)
    # store solution for plotting
    B[i] = z[1][0]
    r[i] = z[1][1]
    I[i] = z[1][2]

    # next initial condition
    z0 = z[1]

# plot results
plt.plot(t,B,'r-',label='B(t)')
plt.plot(t,r,'g-',label='r(t)')
plt.plot(t,I,'b-',label='I(t)')
plt.ylabel('Density')
plt.xlabel('time')
plt.legend(loc='best')
plt.show()
```



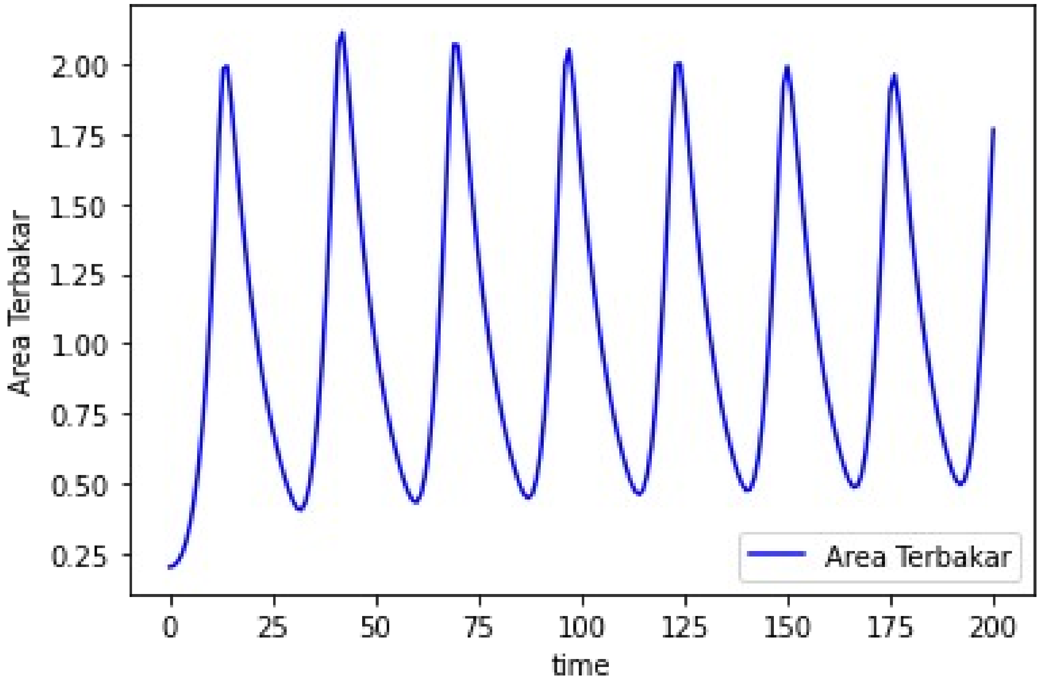
```
# plot results
plt.plot(t,B,'r-',label='Biomass')
plt.ylabel('Biomass volume')
plt.xlabel('time')
plt.legend(loc='best')
plt.show()
```



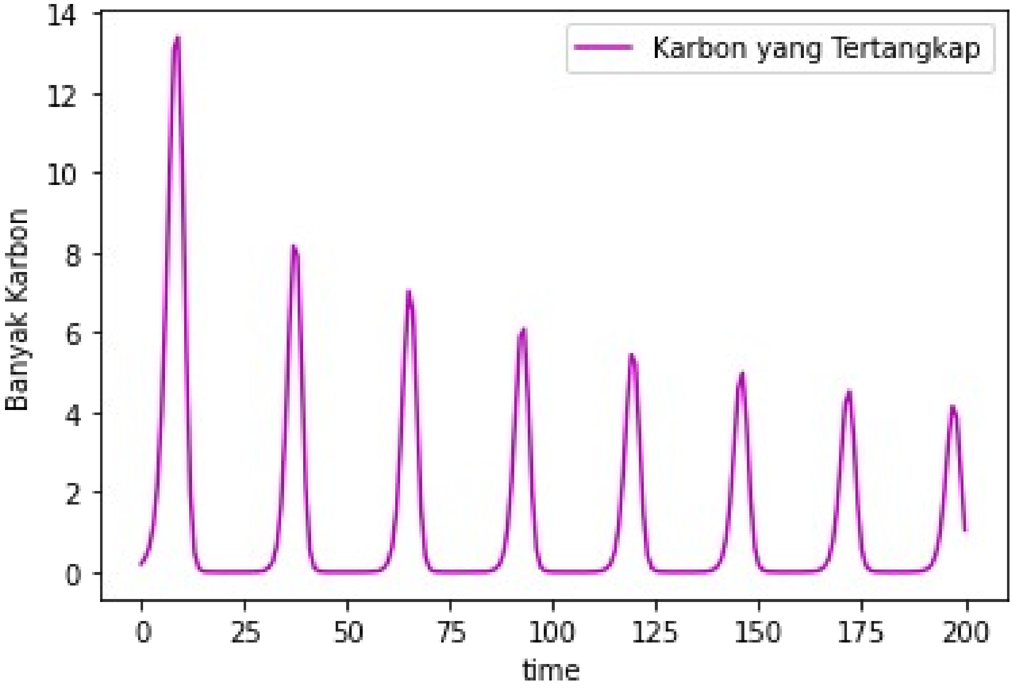
```
# plot results
plt.plot(t,r,'g-',label='Pertumbuhan Intrinsik')
plt.ylabel('Pertumbuhan Intrinsik')
plt.xlabel('time')
plt.legend(loc='best')
plt.show()
```



```
# plot results
plt.plot(t,I,'b-',label='Area Terbakar')
plt.ylabel('Area Terbakar')
plt.xlabel('time')
plt.legend(loc='best')
plt.show()
```



```
# plot results
alpha = 0.5
C = B*alpha
plt.plot(t,C,'m-',label='Karbon yang Tertangkap')
plt.ylabel('Banyak Karbon')
plt.xlabel('time')
plt.legend(loc='best')
plt.show()
```


Karbon yang Tertangkap
Banyak Karbon
0
2
4
6
8
10
12
14
time
0
25
50
75
100
125
150
175
200