

## **LAMPIRAN A**

---

**PROGRAM SIMULASI PERGERAKAN ROBOT  
MOBIL BERODA DARI KOORDINAT KE  
KOORDINAT LAIN TANPA GANGGUAN**

```
em = input('Masukkan tegangan motor (tegangan motor kanan=tegangan
motor kiri)= ');
emrr=em;
er(1)=0;
emll=em;
el(1)=0;

x = input('Masukkan titik awal sumbu x = ');
x1(1)=x;
xaw=x;
y = input('Masukkan titik awal sumbu y = ');
y1(1)=y;
yaw=y;
xrr = input('Masukkan titik akhir sumbu x = ');
yrr = input('Masukkan titik akhir sumbu y = ');

phi = (pi/2);
tetha(1)=phi;

RPM=263;
Vmin=3;
Vmaks=8;

if em<=Vmin
    ' Tegangan diberikan pada motor terlalu kecil '
    Vmin
    www5;
elseif em>Vmaks
    ' Tegangan diberikan pada motor terlalu besar '
    Vmaks
    www5;
elseif em>Vmin && em<=Vmaks

selisih=Vmaks-Vmin;
kec=RPM/selisih;
kenaikan=emrr-Vmin;

Ks=kec*kenaikan;
r2=25e-3;
r1=1e-3;
Ki=3e-1;
R=10000;
Lk=0.63;
Mhub=0.025;
Mwheel=0.05;
Mrobot=3;
m=Mrobot;
rin=0.03;
rout=0.05;
Jmr=0.56e7;
Jml=0.56e7;
b=0.05;
L=0.1;

format long
```

```

xtp=xaw-(L*cos(phi));
ytp=yaw-(L*sin(phi));
deltax=xrr-xtp;
deltay=yrr-ytp;
p=atan2((deltay),(deltax))

xharus=xtp+L*cos(p);
yharus=ytp+L*sin(p);
deltaxs=xrr-xharus;
deltays=yrr-yharus;
smust=sqrt((deltaxs)^2+(deltays)^2)

sampling=0.1;
vcmot=Ks*(2*pi*r1)/60;
wcmot=vcmot/r2;
vc=wcmot*rout

%% Initalisasi
while vt<=vc
    ttti(iii+1)=ti;
    emr=emrr;
    eml=emll;

kg=r2/r1;
ir1=(emr/R)-(emr/R)*exp((-R/Lk)*ti);
il1=(eml/R)-(eml/R)*exp((-R/Lk)*ti);
ir=ir1;
il=il1;
Eemfr1=-emr*exp((-R/Lk)*ti);
Eemfl1=-eml*exp((-R/Lk)*ti);
Eemfr=Eemfr1;
Eemfl=Eemfl1;

wmr=Eemfr/Ks;
wml=Eemfl/Ks;
Tmr=ir*Ki;
Tml=il*Ki;
TLr=Tmr*kg;
TLl=Tml*kg;
wLr=(1/kg)*wmr;
wLl=(1/kg)*wml;
Jhub=Mhub*rin^2/2;
Jwheel=Mwheel*(rin^2+rout^2)/2;
Jrobot=Mrobot*rout^2/2;
J=Jrobot;
JL=Jhub+Jwheel+Jrobot;
Jr=JL+kg^2*Jmr;
Jl=JL+kg^2*Jml;
wLrdot=TLr/Jr;
wLldot=TLl/Jl;
fr=(Jr*wLrdot)/rout;
fl=(Jl*wLldot)/rout;
F=(fr+fl)/2;
vdoot=((1/(2*m))*fr) + ((1/(2*m))*fl);

```

```
        vt=v0+vdoot*ti;
        s=s0+vt*ti+0.5*vdoot*(ti^2);

vaw(iii+1)=vt;
ssaw(iii+1)=s;
iii=1+iii;
ti=ti+sampling;
if s<smust
    tmust=ti;
end
end
tper=ti;
tpercepatan=ti/60
spercepatan=s
vcc=vt

teta=phi-p
tetax=teta;

if teta<0 || teta >=pi
    tetal=abs(teta);
    if tetal>=pi
        teta=2*pi-tetal
    else
        teta=tetal
    end
end

kel=2*pi*(2*b);
fi=(vcc/kel)*(2*pi);
rad1=fi*sampling;
radi=rad1*2;
n=teta/radi;
mut=abs(teta);
waktu=(mut/radi)*sampling;

sa=spercepatan;
tsam=tpercepatan;
scm=smust-(2*sa);
tcom=(scm/vcc)/60;
t1m=tsam+tcom;
totm=(tsam*2)+tcom;
tcm=tsam*60+waktu;
tbm=t1m*60+waktu;
tam=tbm+tcm-waktu;
if tbm<=(tsam*60)
    scm=0;
    tcom=0;
    t1m=(tmust/60)/2;
    totm=tmust/60;
    tcm=t1m*60+waktu;
    tbm =t1m*60+waktu;
    tam=tbm+tcm-waktu;
end
totalwaktu=tam+waktu;
```

```
%% MULAI PROGRAM
while t<=totalwaktu
    ttt(iii+1)=t;
    tbay=tbay+sampling;
    tbayang(ibay+1)=tbay;
    if t<waktu
        if (deltax>=0 & deltay>0) || (deltax>0 & deltay<=0)
            emr=-emrr;
            eml=epll;
            er(iiii+1)=emr;
            el(iiii+1)=eml;
            ir1=(emr/R) - (emr/R)*exp((-R/Lk)*t);
            il1=(eml/R) - (eml/R)*exp((-R/Lk)*t);
            ir=ir1;
            il=il1;
            Eemfr1=-emr*exp((-R/Lk)*t);
            Eemfl1=-eml*exp((-R/Lk)*t);
            Eemfr=Eemfr1;
            Eemfl=Eemfl1;
        end
        if (deltax<0 & deltay>0) || (deltax<=0 & deltay<=0)
            emr=emrr;
            eml=-epll;
            er(iiii+1)=emr;
            el(iiii+1)=eml;
            ir1=(emr/R) - (emr/R)*exp((-R/Lk)*t);
            il1=(eml/R) - (eml/R)*exp((-R/Lk)*t);
            ir=ir1;
            il=il1;
            Eemfr1=-emr*exp((-R/Lk)*t);
            Eemfl1=-eml*exp((-R/Lk)*t);
            Eemfr=Eemfr1;
            Eemfl=Eemfl1;
        end
    end
    if p==phi
        waktu=0;
        teth=phi;
        sdt=teth;
    end

if t>=waktu
    deltaxs=xrr-xharus;
    deltays=yrr-yharus;
    s=sqrt(((deltaxs)^2)+((deltays)^2));
    if s>(2*sa)
        sc=s-(2*sa);
        tco=(sc/vcc)/60;
        t1=tsam+tco;
        tot=(tsam*2)+tco;
        tc=tsam*60+waktu;
        tb=t1*60+waktu;
        ta=tb+tc-waktu;
    end
    if s==(2*sa)
        sc=0;
    end
end
```

```

        tco=0;
        t1=tsam/2;
        tot=tsam;
        tc=t1*60+waktu;
        tb=tsam/2+waktu;
        ta=tb+tc-waktu;
    end
    if s<(2*sa)
        sc=0;
        tco=0;
        t1=(tmust/60)/2;
        tot=tmust/60;
        tc=t1*60+waktu;
        tb =t1*60+waktu;
        ta=tb+tc-waktu;
    end

    if t<tc
        emr=emrr;
        eml=eml1;
        er(iiii+1)=emr;
        el(iiii+1)=eml;
        ir1=(emr/R) - (emr/R)*exp((-R/Lk)*ts);
        il1=(eml/R) - (eml/R)*exp((-R/Lk)*ts);
        ir=ir1;
        il=il1;
        Eemfr1=-emr*exp((-R/Lk)*ts);
        Eemfl1=-eml*exp((-R/Lk)*ts);
        Eemfr=Eemfr1;
        Eemfl=Eemfl1;
        ts=ts+sampling;
    end

    if s>(2*sa)
        if t>=tc && t<tb
            emr=emrr;
            eml=eml1;
            er(iiii+1)=emr;
            el(iiii+1)=eml;
            ir1=(emr/R) - (emr/R)*exp((-R/Lk)*ts);
            il1=(eml/R) - (eml/R)*exp((-R/Lk)*ts);
            ir2=- (emr/R) + (emr/R)*exp((-R/Lk)*t0);
            il2=- (eml/R) + (eml/R)*exp((-R/Lk)*t0);

            ir=ir1+ir2;
            il=il1+il2;

            Eemfr1=-emr*exp((-R/Lk)*ts);
            Eemfl1=-eml*exp((-R/Lk)*ts);
            Eemfr2=emr*exp((-R/Lk)*t0);
            Eemfl2=eml*exp((-R/Lk)*t0);

            Eemfr=Eemfr1+Eemfr2;
            Eemfl=Eemfl1+Eemfl2;
            ts=ts+sampling;
        end
    end

```

```

        t0=t0+sampling;
end

if t>=tb && t<(tb+tc-waktu)
    emr=emrr;
    eml=emll;
    er(iiii+1)=0;
    el(iiii+1)=0;
    ir1=(emr/R) - (emr/R)*exp((-R/Lk)*ts);
    il1=(eml/R) - (eml/R)*exp((-R/Lk)*ts);
    ir2=- (emr/R) + (emr/R)*exp((-R/Lk)*t0);
    il2=- (eml/R) + (eml/R)*exp((-R/Lk)*t0);
    ir1off=- ((emr/R) - (emr/R)*exp((-R/Lk)*t00));
    il1off=- ((eml/R) - (eml/R)*exp((-R/Lk)*t00));
    ir=ir1+ir2+ir1off;
    il=il1+il2+il1off;

    Eemfr1=-emr*exp((-R/Lk)*ts);
    Eemfl1=-eml*exp((-R/Lk)*ts);
    Eemfr2=emr*exp((-R/Lk)*t0);
    Eemfl2=eml*exp((-R/Lk)*t0);
    Eemfr1off=emr*exp((-R/Lk)*t00);
    Eemfl1off=eml*exp((-R/Lk)*t00);

    Eemfr=Eemfr1+Eemfr2+Eemfr1off;
    Eemfl=Eemfl1+Eemfl2+Eemfl1off;
    t0=t0+sampling;
    t00=t00+sampling;
    ts=ts+sampling;
end

if t>=(tb+tc-waktu)
    emr=emrr;
    eml=emll;
    er(iiii+1)=0;
    el(iiii+1)=0;
    ir1=(emr/R) - (emr/R)*exp((-R/Lk)*ts);
    il1=(eml/R) - (eml/R)*exp((-R/Lk)*ts);
    ir2=- (emr/R) + (emr/R)*exp((-R/Lk)*t0);
    il2=- (eml/R) + (eml/R)*exp((-R/Lk)*t0);
    ir1off=- ((emr/R) - (emr/R)*exp((-R/Lk)*t00));
    il1off=- ((eml/R) - (eml/R)*exp((-R/Lk)*t00));
    ir2off=(emr/R) - (emr/R)*exp((-R/Lk)*t000);
    il2off=(eml/R) - (eml/R)*exp((-R/Lk)*t000);

    ir=ir1+ir2+ir1off+ir2off;
    il=il1+il2+il1off+il2off;

    Eemfr1=-emr*exp((-R/Lk)*ts);
    Eemfl1=-eml*exp((-R/Lk)*ts);
    Eemfr2=emr*exp((-R/Lk)*t0);
    Eemfl2=eml*exp((-R/Lk)*t0);
    Eemfr1off=emr*exp((-R/Lk)*t00);
    Eemfl1off=eml*exp((-R/Lk)*t00);
    Eemfr2off=-emr*exp((-R/Lk)*t000);

```

```
Eemfl2off=-eml*exp((-R/Lk)*t000);
Eemfr=Eemfr1+Eemfr2+Eemfr1off+Eemfr2off;
Eemfl=Eemfl1+Eemfl2+Eemfl1off+Eemfl2off;
ts=ts+sampling;
t0=t0+sampling;
t00=t00+sampling;
t000=t000+sampling;
end
end

if s<(2*sa)
    if t>=tb && t<(tb+tc-waktu)
        emr=emrr;
        eml=emll;
        er(iiii+1)=0;
        el(iiii+1)=0;
        ir1=(emr/R)-(emr/R)*exp((-R/Lk)*ts);
        il1=(eml/R)-(eml/R)*exp((-R/Lk)*ts);
        ir2=- (emr/R)+(emr/R)*exp((-R/Lk)*t0);
        il2=- (eml/R)+(eml/R)*exp((-R/Lk)*t0);
        ir1off=- ((emr/R)-(emr/R)*exp((-R/Lk)*t00));
        il1off=- ((eml/R)-(eml/R)*exp((-R/Lk)*t00));
        ir=ir1+ir2+ir1off;
        il=il1+il2+il1off;

        Eemfr1=-emr*exp((-R/Lk)*ts);
        Eemfl1=-eml*exp((-R/Lk)*ts);
        Eemfr2=emr*exp((-R/Lk)*t0);
        Eemfl2=eml*exp((-R/Lk)*t0);
        Eemfr1off=emr*exp((-R/Lk)*t00);
        Eemfl1off=eml*exp((-R/Lk)*t00);

        Eemfr=Eemfr1+Eemfr2+Eemfr1off;
        Eemfl=Eemfl1+Eemfl2+Eemfl1off;
        ts=ts+sampling;
        t0=t0+sampling;
        t00=t00+sampling;
    end

    if t>=(tb+tc-waktu)
        emr=0;
        eml=0;
        er(iiii+1)=0;
        el(iiii+1)=0;
        ir1=(emr/R)-(emr/R)*exp((-R/Lk)*ts);
        il1=(eml/R)-(eml/R)*exp((-R/Lk)*ts);
        ir2=- (emr/R)+(emr/R)*exp((-R/Lk)*t0);
        il2=- (eml/R)+(eml/R)*exp((-R/Lk)*t0);
        ir1off=- ((emr/R)-(emr/R)*exp((-R/Lk)*t00));
        il1off=- ((eml/R)-(eml/R)*exp((-R/Lk)*t00));
        ir2off=(emr/R)-(emr/R)*exp((-R/Lk)*t000);
        il2off=(eml/R)-(eml/R)*exp((-R/Lk)*t000);
        ir=ir1+ir2+ir1off+ir2off;
        il=il1+il2+il1off+il2off;
```



```

Eemfr1=-emr*exp((-R/Lk)*ts);
Eemfl1=-eml*exp((-R/Lk)*ts);
Eemfr2=emr*exp((-R/Lk)*t0);
Eemfl2=eml*exp((-R/Lk)*t0);
Eemfr1off=emr*exp((-R/Lk)*t00);
Eemfl1off=eml*exp((-R/Lk)*t00);
Eemfr2off=-emr*exp((-R/Lk)*t000);
Eemfl2off=-eml*exp((-R/Lk)*t000);
Eemfr=Eemfr1+Eemfr2+Eemfr1off+Eemfr2off;
Eemfl=Eemfl1+Eemfl2+Eemfl1off+Eemfl2off;
ts=ts+sampling;
t0=t0+sampling;
t00=t00+sampling;
t000=t000+sampling;
end
end
end

kg=r2/r1;
wmr=Eemfr/Ks;
wml=Eemfl/Ks;
Tmr=ir*Ki;
Tml=il*Ki;
TLr=Tmr*kg;
TLl=Tml*kg;
wLr=(1/kg)*wmr;
wLl=(1/kg)*wml;
Jhub=Mhub*rin^2/2;
Jwheel=Mwheel*(rin^2+rout^2)/2;
Jrobot=Mrobot*rout^2/2;
J=Jrobot;
JL=Jhub+Jwheel+Jrobot;
Jr=JL+kg^2*Jmr;
Jl=JL+kg^2*Jml;
wLrdot=TLr/Jr;
wLldot=TLl/Jl;
fr=(Jr*wLrdot)/rout;
fl=(Jl*wLldot)/rout;
F=(fr+fl)/2;
vdott=((1/(2*m))*fr) + ((1/(2*m))*fl);
vdotrr(iii+1)=vdott;
wdoot=((1/(2*b*J))*fr - (1/(2*b*J))*fl);
wdotrr(iii+1)=wdoot;
wt=wt+wdoot*t;

if t<waktu
vvt=v0+vdott*t;
sss=s0+vvt*t+0.5*vdott*(t^2);
vvtw1=vvt;
s0=sss;
if (phi>=0 && phi<=(pi/2)) || (phi>(pi/2) && phi<pi)
if (deltax>0 && deltay>0)
te=te+radi;
teth=phi-te;
sdt=teth;

```

```
end
if (deltax<0 && deltay>=0)
    te=te+radi;
    teth=phi+te;
    sdt=teth;
end
if (deltax<=0 && deltay<=0)
    te=te+radi;
    teth=phi-te;
    sdt=teth;
    if phi>=(pi/2) && phi<(pi)
        teth=phi+te;
        sdt=teth;
    end
end
if (deltax>0 && deltay<0)
    te=te+radi;
    teth=phi-te;
    sdt=teth;
end
end
if (phi<0 && phi>=(-pi/2)) || (phi<(-pi/2) && phi>(-pi))
    if (deltax>0 && deltay>0)
        te=te+radi;
        teth=phi+te;
        sdt=teth;
    end
    if (deltax<0 && deltay>=0)
        te=te+radi;
        teth=phi+te;
        sdt=teth;
        if (phi<(-pi/2) && phi>(-pi))
            teth=phi-te;
            sdt=teth;
        end
    end
    if (deltax<=0 && deltay<=0)
        te=te+radi;
        teth=phi-te;
        sdt=teth;
    end
    if (deltax>0 && deltay<0)
        te=te+radi;
        teth=phi-te;
        sdt=teth;
        if (phi<(-pi/2) && phi>(-pi))
            teth=te+phi;
            sdt=teth;
        end
    end
end
end
end
xharus=xtp+L*cos(teth);
yharus=ytp+L*sin(teth);

if t>=waktu
```

```

        if t<tc
            vvt=v0+vdott*tss;
            sss=s0+vvt*tss+0.5*vdott*(tss^2);
            vvt1=vvt;
            ss1=sss;
            tss=tss+sampling;
        end
    if s>(2*sa)
        if t>=tc && t<tb
            vvt=vvt1+vdott*tv;
            sss=ss1+vvt*tv+0.5*vdott*(tv^2);
            vvt2=vvt;
            ss2=sss;
            tv=tv+sampling;
        end
        if t>=tb && t<(tb+tc-waktu)
            vvt=vvt2+vdott*tvv;
            sss=ss2+vvt*tvv-0.5*vdott*(tvv^2);
            vvt3=vvt;
            ss3=sss;
            tvv=tvv+sampling;
        end
        if t>=(tb+tc-waktu)
            vt0=v0;
            vvt=vt0+vdott*tvvv;
            sss=ss3+vvt*tvvv-0.5*vdott*(tvvv^2);
            tvvv=tvvv+sampling;
        end
    end
end

if s<=(2*sa)
    if t>=tb && t<(tb+tc-waktu)
        vvt=vvt1+vdott*tvvi;
        sss=ss1+vvt*tvvi-0.5*vdott*(tvvi^2);
        vvt2=vvt;
        ss3=sss;
        tvvi=tvvi+sampling;
    end
    if t>=(tb+tc-waktu)
        vt0=v0;
        vvt=vt0+vdott*tvvvi;
        sss=ss3+vvt*tvvvi-0.5*vdott*(tvvvi^2);
        tvvvi=tvvvi+sampling;
    end
end
end

waw(iii+1)=wt;
tetha(iiii+1)=teth;
tettha(iii+1)=te;

vv(iii+1)=vvt;
ssss(iii+1)=sss;
iii=iii+1;
t=t+sampling;

```

```
if (deltaxs>=0 && deltays>0) || (deltaxs>0 && deltays<=0)
    xakhir=xtp+L*cos(teth)+sss*cos(sdt);
    yakhir=ytp+L*sin(teth)+sss*sin(sdt);
end
if (deltaxs<0 && deltays>0) || (deltaxs<=0 && deltays<=0)
    xakhir=xtp+L*cos(teth)+sss*cos(sdt);
    yakhir=ytp+L*sin(teth)+sss*sin(sdt);
end

x1(iiii+1)=xakhir;
y1(iiii+1)=yakhir;
iiii=iiii+1;
ibay=ibay+1;
end
sudutakhir=teth
xterakhir=xakhir
yterakhir=yakhir
waktu=tam

%% Output

figure(1);
subplot(4,1,1);
plot(tbayang,er);
grid;
xlabel('Waktu (s)'), ylabel('Tegangan (V)');
title('Hubungan tegangan motor kanan terhadap waktu')

subplot(4,1,2);
plot(tbayang,el);
grid;
xlabel('Waktu (s)'), ylabel('Tegangan (V)');
title('Hubungan tegangan motor kiri terhadap waktu')

subplot(4,1,3);
plot(ttt,vv);
grid;
xlabel('Waktu (s)'), ylabel('Kecepatan(m/s)');
title('Hubungan kecepatan robot terhadap waktu')

subplot(4,1,4);
plot(ttt,ssss);
grid;
xlabel('Waktu (s)'), ylabel('jarak');
title('Hubungan jarak terhadap waktu')

figure(2);
subplot(2,2,1);
plot(ttt,vdotrr);
grid;
xlabel('Waktu (s)'), ylabel('Percepatan');
title('Hubungan percepatan robot terhadap waktu')
subplot(2,2,2);
```

```
plot(ttt,vv);
grid;
xlabel('Waktu (s)'), ylabel('Kecepatan (m/s)');
title('Hubungan kecepatan robot terhadap waktu')

subplot(2,2,3);
plot(ttt,ssss);
grid;
xlabel('Waktu (s)'), ylabel('jarak');
title('Hubungan jarak terhadap waktu')

subplot(2,2,4);
plot(x1,y1);
grid;
xlabel('X'), ylabel('Y');
title('Koordinat posisi robot')

figure(3);
plot(tbayang,tetha);
grid;
xlabel('Waktu (s)'), ylabel('sudut');
title('Hubungan sudut terhadap waktu')
end
```

## **LAMPIRAN B**

---

**PROGRAM SIMULASI PERGERAKAN ROBOT  
MOBIL BERODA DARI KOORDINAT KE  
KOORDINAT LAIN DENGAN GANGGUAN**

```
em = input('Masukkan tegangan motor (tegangan motor kanan=tegangan
motor kiri)= ');
emrr=em;
er(1)=0;
emll=em;
el(1)=0;

x = input('masukkan titik awal sumbu x = ');
x1(1)=x;
xaw=x;
y = input('masukkan titik awal sumbu y = ');
y1(1)=y;
yaw=y;
xrr = input('masukkan titik akhir sumbu x = ');
yrr = input('masukkan titik akhir sumbu y = ');

phi = pi/2;
tetha(1)=phi;

RPM=263;
Vmin=3;
Vmaks=8;

if em<=Vmin
    ' Tegangan diberikan pada motor terlalu kecil '
    Vmin
    noise7;
elseif em>Vmaks
    ' Tegangan diberikan pada motor terlalu besar '
    Vmaks
    noise7;
elseif em>Vmin && em<=Vmaks

selisih=Vmaks-Vmin;
kec=RPM/selisih;
kenaikan=emrr-Vmin;

Ks=kec*kenaikan;
r2=25e-3;
r1=1e-3;
Ki=3e-1;
R=10000;
Lk=0.63;
Mhub=0.025;
Mwheel=0.05;
Mrobot=3;
m=Mrobot;
rin=0.03;
rout=0.05;
Jmr=0.56e7;
Jml=0.56e7;
b=0.05;
L=0.1;

format long
```

```

xtp=xaw-(L*cos(phi));
ytp=yaw-(L*sin(phi));
deltax=xrr-xtp;
deltay=yrr-ytp;
p=atan2((deltay),(deltax))

xharus=xtp+L*cos(p);
yharus=ytp+L*sin(p);
deltaxs=xrr-xharus;
deltays=yrr-yharus;
smust=sqrt((deltaxs)^2+(deltays)^2)

sampling=0.01;
vcmot=Ks*(2*pi*r1)/60;
wcmot=vcmot/r2;
vc=wcmot*rout;

while vt<=vc
    ttti(iii+1)=ti;
    emr=emrr;
    eml=emll;

kg=r2/r1;
ir1=(emr/R)-(emr/R)*exp((-R/Lk)*ti);
il1=(eml/R)-(eml/R)*exp((-R/Lk)*ti);
ir=ir1;
il=il1;
Eemfr1=-emr*exp((-R/Lk)*ti);
Eemfl1=-eml*exp((-R/Lk)*ti);
Eemfr=Eemfr1;
Eemfl=Eemfl1;

wmr=Eemfr/Ks;
wml=Eemfl/Ks;
Tmr=ir*Ki;
Tml=il*Ki;
TLr=Tmr*kg;
TLl=Tml*kg;
wLr=(1/kg)*wmr;
wLl=(1/kg)*wml;
Jhub=Mhub*rin^2/2;
Jwheel=Mwheel*(rin^2+rout^2)/2;
Jrobot=Mrobot*rout^2/2;
J=Jrobot;
JL=Jhub+Jwheel+Jrobot;
Jr=JL+kg^2*Jmr;
Jl=JL+kg^2*Jml;
wLrdot=TLr/Jr;
wLldot=TLl/Jl;
fr=(Jr*wLrdot)/rout;
fl=(Jl*wLldot)/rout;
F=(fr+fl)/2;
vdoot=((1/(2*m))*fr) + ((1/(2*m))*fl);

vt=v0+vdoot*ti;

```



```
s=s0+vt*ti+0.5*vdoot*(ti^2);

vaw(iii+1)=vt;
ssaw(iii+1)=s;
iii=1+iii;
ti=ti+sampling;
if s<smust
    tmust=ti;
end
end
tper=ti;
tpercepatan=ti/60
spercepatan=s
vcc=vt

teta=phi-p
if teta<0
    teta=abs(teta)
end
if teta>=pi
    teta=2*pi-teta
end

kel=2*pi*(2*b);
fi=(vcc/kel)*(2*pi);
radl=fi*sampling;
radi=radl*2;
mut=abs(teta);
waktu=(mut/radi)*sampling;

sa=spercepatan;
tsam=tpercepatan;
scm=smust-(2*sa);
tcom=(scm/vcc)/60;
t1m=tsam+tcom;
totm=(tsam*2)+tcom;
tcm=tsam*60+waktu;
tbm=t1m*60+waktu;
tam=tbm+tcm-waktu;

if tbm<=(tsam*60)
    scm=0;
    tcom=0;
    t1m=(tmust/60)/2;
    totm=tmust/60;
    tcm=t1m*60+waktu;
    tbm=t1m*60+waktu;
    tam=tbm+tcm-waktu;
end
totalwaktu=tam;

while tnn<(totm-sampling)
    tnn=tnn+sampling;
    tnnn(nn+1)=tnn;
    nn=nn+1;
```

```

end
noises=randperm(nn);
noises1=noises(1);
tn=tnnn(noises1);
tno=(tn*60)
sudut=0;
sud=0;
while sudut<(2*pi)
    sudut=sudut+0.01
    suduts(sud+1)=sudut
    sud=sud+1;
end
sudutt=randperm(sud);
sudutt1=sudutt(1);
suduttt=suduts(sudutt1);

%% MULAI PROGRAM
tcl=tsam*60;
while t<=tno
    tt(iii+1)=t;
    tbay=tbay+sampling;
    tbayang(ibay+1)=tbay;
    if t<waktu
        if (deltaxs>=0 & deltaxs>0) || (deltaxs>0 & deltaxs<=0)
            emr=-emrr;
            eml=eml1;
            er(iiii+1)=emr;
            el(iiii+1)=eml;
            ir1=(emr/R) - (emr/R)*exp((-R/Lk)*t);
            il1=(eml/R) - (eml/R)*exp((-R/Lk)*t);
            ir=ir1;
            il=il1;
            Eemfr1=-emr*exp((-R/Lk)*t);
            Eemfl1=-eml*exp((-R/Lk)*t);
            Eemfr=Eemfr1;
            Eemfl=Eemfl1;

        end
        if (deltaxs<0 & deltaxs>0) || (deltaxs<=0 & deltaxs<=0)
            emr=emrr;
            eml=-eml1;
            er(iiii+1)=emr;
            el(iiii+1)=eml;
            ir1=(emr/R) - (emr/R)*exp((-R/Lk)*t);
            il1=(eml/R) - (eml/R)*exp((-R/Lk)*t);
            ir=ir1;
            il=il1;
            Eemfr1=-emr*exp((-R/Lk)*t);
            Eemfl1=-eml*exp((-R/Lk)*t);
            Eemfr=Eemfr1;
            Eemfl=Eemfl1;

        end
    end
    if p==phi
        waktu=0;
        teth=phi;
    end
end

```

```

        sdt=teth;
    end

    if t>=waktu
        deltaxs=xrr-xharus;
        deltays=yrr-yharus;
        s=sqrt(((deltaxs)^2)+((deltays)^2));
        if s>(2*sa)
            sc=s-(2*sa);
            tco=(sc/vcc)/60;
            t1=tsam+tco;
            tot=(tsam*2)+tco;
            tc=tsam*60+waktu;
            tb=t1*60+waktu;
            ta=tb+tc-waktu;
        end
        if s==(2*sa)
            sc=0;
            tco=0;
            t1=tsam/2;
            tot=tsam;
            tc=t1*60+waktu;
            tb=tsam/2+waktu;
            ta=tb+tc-waktu;
        end
        if s<(2*sa)
            sc=0;
            tco=0;
            t1=(tmust/60)/2;
            tot=tmust/60;
            tc=t1*60+waktu;
            tb =t1*60+waktu;
            ta=tb+tc-waktu;
        end

        if t<tc
            emr=emrr;
            eml=emll;
            er(iiii+1)=emr;
            el(iiii+1)=eml;
            ir1=(emr/R)-(emr/R)*exp((-R/Lk)*ts);
            il1=(eml/R)-(eml/R)*exp((-R/Lk)*ts);
            ir=ir1;
            il=il1;
            ia(iii+1)=ir;
            ib(iii+1)=il;
            il=il1;
            Eemfr1=-emr*exp((-R/Lk)*ts);
            Eemfl1=-eml*exp((-R/Lk)*ts);
            Eemfr=Eemfr1;
            Eemfl=Eemfl1;
            ts=ts+sampling;
        end

        if s>(2*sa)
            if t>=tc && t<tb

```

```
emr=emrr;
eml=emll;
er(iiii+1)=emr;
el(iiii+1)=eml;
ir1=(emr/R)-(emr/R)*exp((-R/Lk)*ts);
il1=(eml/R)-(eml/R)*exp((-R/Lk)*ts);
ir2=- (emr/R)+(emr/R)*exp((-R/Lk)*t0);
il2=- (eml/R)+(eml/R)*exp((-R/Lk)*t0);

ir=ir1+ir2;
il=il1+il2;

Eemfr1=-emr*exp((-R/Lk)*ts);
Eemfl1=-eml*exp((-R/Lk)*ts);
Eemfr2=emr*exp((-R/Lk)*t0);
Eemfl2=eml*exp((-R/Lk)*t0);

Eemfr=Eemfr1+Eemfr2;
Eemfl=Eemfl1+Eemfl2;
ts=ts+sampling;
t0=t0+sampling;
end

if t>=tb && t<(tb+tc-waktu)
emr=emrr;
eml=emll;
er(iiii+1)=0;
el(iiii+1)=0;
ir1=(emr/R)-(emr/R)*exp((-R/Lk)*ts);
il1=(eml/R)-(eml/R)*exp((-R/Lk)*ts);
ir2=- (emr/R)+(emr/R)*exp((-R/Lk)*t0);
il2=- (eml/R)+(eml/R)*exp((-R/Lk)*t0);
ir1off=- ((emr/R)-(emr/R)*exp((-R/Lk)*t00));
il1off=- ((eml/R)-(eml/R)*exp((-R/Lk)*t00));
ir=ir1+ir2+ir1off;
il=il1+il2+il1off;

Eemfr1=-emr*exp((-R/Lk)*ts);
Eemfl1=-eml*exp((-R/Lk)*ts);
Eemfr2=emr*exp((-R/Lk)*t0);
Eemfl2=eml*exp((-R/Lk)*t0);
Eemfr1off=emr*exp((-R/Lk)*t00);
Eemfl1off=eml*exp((-R/Lk)*t00);

Eemfr=Eemfr1+Eemfr2+Eemfr1off;
Eemfl=Eemfl1+Eemfl2+Eemfl1off;
t0=t0+sampling;
t00=t00+sampling;
ts=ts+sampling;
end

if t>=(tb+tc-waktu)
emr=emrr;
eml=emll;
er(iiii+1)=0;
```

```

el(iiii+1)=0;
ir1=(emr/R)-(emr/R)*exp((-R/Lk)*ts);
il1=(eml/R)-(eml/R)*exp((-R/Lk)*ts);
ir2=- (emr/R)+(emr/R)*exp((-R/Lk)*t0);
il2=- (eml/R)+(eml/R)*exp((-R/Lk)*t0);
ir1off=- ((emr/R)-(emr/R)*exp((-R/Lk)*t00));
il1off=- ((eml/R)-(eml/R)*exp((-R/Lk)*t00));
ir2off=(emr/R)-(emr/R)*exp((-R/Lk)*t000);
il2off=(eml/R)-(eml/R)*exp((-R/Lk)*t000);

ir=ir1+ir2+ir1off+ir2off;
il=il1+il2+il1off+il2off;

Eemfr1=-emr*exp((-R/Lk)*ts);
Eemfl1=-eml*exp((-R/Lk)*ts);
Eemfr2=emr*exp((-R/Lk)*t0);
Eemfl2=eml*exp((-R/Lk)*t0);
Eemfr1off=emr*exp((-R/Lk)*t00);
Eemfl1off=eml*exp((-R/Lk)*t00);
Eemfr2off=-emr*exp((-R/Lk)*t000);
Eemfl2off=-eml*exp((-R/Lk)*t000);
Eemfr=Eemfr1+Eemfr2+Eemfr1off+Eemfr2off;
Eemfl=Eemfl1+Eemfl2+Eemfl1off+Eemfl2off;
ts=ts+sampling;
t0=t0+sampling;
t00=t00+sampling;
t000=t000+sampling;
end
end

if s<(2*sa)
    if t>=tb && t<(tb+tc-waktu)
        emr=emrr;
        eml=emll;
        er(iiii+1)=0;
        el(iiii+1)=0;
        ir1=(emr/R)-(emr/R)*exp((-R/Lk)*ts);
        il1=(eml/R)-(eml/R)*exp((-R/Lk)*ts);
        ir2=- (emr/R)+(emr/R)*exp((-R/Lk)*t0);
        il2=- (eml/R)+(eml/R)*exp((-R/Lk)*t0);
        ir1off=- ((emr/R)-(emr/R)*exp((-R/Lk)*t00));
        il1off=- ((eml/R)-(eml/R)*exp((-R/Lk)*t00));
        ir=ir1+ir2+ir1off;
        il=il1+il2+il1off;
        Eemfr1=-emr*exp((-R/Lk)*ts);
        Eemfl1=-eml*exp((-R/Lk)*ts);
        Eemfr2=emr*exp((-R/Lk)*t0);
        Eemfl2=eml*exp((-R/Lk)*t0);
        Eemfr1off=emr*exp((-R/Lk)*t00);
        Eemfl1off=eml*exp((-R/Lk)*t00);

        Eemfr=Eemfr1+Eemfr2+Eemfr1off;
        Eemfl=Eemfl1+Eemfl2+Eemfl1off;
        ts=ts+sampling;
        t0=t0+sampling;
    
```

```

        t00=t00+sampling;
    end

    if t>=(tb+tc-waktu)
        emr=0;
        eml=0;
        er(iiii+1)=0;
        el(iiii+1)=0;
        ir1=(emr/R) - (emr/R)*exp((-R/Lk)*ts);
        il1=(eml/R) - (eml/R)*exp((-R/Lk)*ts);
        ir2=- (emr/R) + (emr/R)*exp((-R/Lk)*t0);
        il2=- (eml/R) + (eml/R)*exp((-R/Lk)*t0);
        ir1off=- ((emr/R) - (emr/R)*exp((-R/Lk)*t00));
        il1off=- ((eml/R) - (eml/R)*exp((-R/Lk)*t00));
        ir2off=(emr/R) - (emr/R)*exp((-R/Lk)*t000);
        il2off=(eml/R) - (eml/R)*exp((-R/Lk)*t000);

        ir=ir1+ir2+ir1off+ir2off;
        il=il1+il2+il1off+il2off;

        Eemfr1=-emr*exp((-R/Lk)*ts);
        Eemfl1=-eml*exp((-R/Lk)*ts);
        Eemfr2=emr*exp((-R/Lk)*t0);
        Eemfl2=eml*exp((-R/Lk)*t0);
        Eemfr1off=emr*exp((-R/Lk)*t00);
        Eemfl1off=eml*exp((-R/Lk)*t00);
        Eemfr2off=-emr*exp((-R/Lk)*t000);
        Eemfl2off=-eml*exp((-R/Lk)*t000);
        Eemfr=Eemfr1+Eemfr2+Eemfr1off+Eemfr2off;
        Eemfl=Eemfl1+Eemfl2+Eemfl1off+Eemfl2off;
        ts=ts+sampling;
        t0=t0+sampling;
        t00=t00+sampling;
        t000=t000+sampling;
    end
end
end

kg=r2/r1;
wmr=Eemfr/Ks;
wml=Eemfl/Ks;
Tmr=ir*Ki;
Tml=il*Ki;
TLr=Tmr*kg;
TLl=Tml*kg;
wLr=(1/kg)*wmr;
wLl=(1/kg)*wml;
Jhub=Mhub*rin^2/2;
Jwheel=Mwheel*(rin^2+rout^2)/2;
Jrobot=Mrobot*rout^2/2;
J=Jrobot;
JL=Jhub+Jwheel+Jrobot;
Jr=JL+kg^2*Jmr;
Jl=JL+kg^2*Jml;
wLrdot=TLr/Jr;

```

```
wLldot=TL1/J1;
fr=(Jr*wLrdot)/rout;
fl=(Jl*wLldot)/rout;
F=(fr+fl)/2;
vdott=((1/(2*m))*fr) + ((1/(2*m))*fl);
vdotr(iii+1)=vdott;

    if t<waktu
        vvt=v0+vdott*t;
        sss=s0+vvt*t+0.5*vdott*(t^2);
        vvtw1=vvt;
        s0=sss;
        if (deltax>0 && deltay>0)
            te=te+radi;
            teth=phi-te;
            sdt=teth;
        end
        if (deltax<0 && deltay>=0)
            te=te+radi;
            teth=phi+te;
            sdt=(pi/2)-teth;
        end
        if (deltax<=0 && deltay<=0)
            te=te+radi;
            teth=phi+te;
            sdt=teth-pi;
        end
        if (deltax>0 && deltay<0)
            te=te+radi;
            teth=phi-te;
            sdt=teth;
        end
        if phi<0
            teth=pi-teth;
            sdt=pi-sdt;
            if (deltaxs<0 && deltaxs>=0)
                sdt=(pi/2)-sdt;
            end
        end
    end
    xharus=xtp+L*cos(teth);
    yharus=ytp+L*sin(teth);

    if t>=waktu
        if t<tc
            vvt=v0+vdott*tss;
            sss=s0+vvt*tss+0.5*vdott*(tss^2);
            vvt1=vvt;
            ss1=sss;
            tss=tss+sampling;
        end
        if s>(2*sa)
            if t>=tc && t<tb
                vvt=vvt1+vdott*tv;
                sss=ss1+vvt*tv+0.5*vdott*(tv^2);
                vvt2=vvt;
```

```

        ss2=sss;
        tv=tv+sampling;
    end
    if t>=tb && t<(tb+tc-waktu)
        vvt=vvt2+vdott*tvv;
        sss=ss2+vvt*tvv-0.5*vdott*(tvv^2);
        vvt3=vvt;
        ss3=sss;
        tvv=tvv+sampling;
    end
    if t>=(tb+tc-waktu)
        vt0=v0;
        vvt=vt0+vdott*tvvv;
        sss=ss3+vvt*tvvv-0.5*vdott*(tvvv^2);
        tvvv=tvvv+sampling;
    end
end

if s<=(2*sa)
    if t>=tb && t<(tb+tc-waktu)
        vvt=vvt1+vdott*tvvi;
        sss=ss1+vvt*tvvi-0.5*vdott*(tvvi^2);
        vvt2=vvt;
        ss3=sss;
        tvvi=tvvi+sampling;
    end
    if t>=(tb+tc-waktu)
        vt0=v0;
        vvt=vt0+vdott*tvvvi;
        sss=ss3+vvt*tvvvi-0.5*vdott*(tvvvi^2);
        tvvvi=tvvvi+sampling;
    end
end
end

waw(iii+1)=wt;
tetha(iiii+1)=teth;
tettha(iiii+1)=te;

v(iii+1)=vvt;
ss(iii+1)=sss;
iii=iii+1;
t=t+sampling;

if (deltax>=0 && deltax>0) || (deltax>0 && deltax<=0)
    xakhir=xtp+L*cos(teth)+sss*cos(sdt);
    yakhir=ytp+L*sin(teth)+sss*sin(sdt);
end
if (deltax<0 && deltax>0) || (deltax<=0 && deltax<=0)
    xakhir=xtp+L*cos(teth)-sss*cos(sdt);
    yakhir=ytp+L*sin(teth)-sss*sin(sdt);
end
x1(iiii+1)=xakhir;
y1(iiii+1)=yakhir;
iiii=iiii+1;

```



```
    ibay=ibay+1;
end

s4=sss
if t>=tno;
    xn=random('Normal',0:0,4);
    x=abs(xn);
    xxn=x
    yn=random('Normal',0:0,4);
    y=abs(yn);
    yyn=y
    x1(iiii+1)=xxn;
    y1(iiii+1)=yyn;
    pn=suduttt;
    tetha(iiii+1)=pn;
    tettha(iiii+1)=0;
    iiii=iiii+1;
    tbay=tbay+sampling;
    tbayang(ibay+1)=tbay;
    ibay=ibay+1;
    mutpn=abs(pn);
    pnx= pn

    if pnx<0
        pnx=(2*pi)-mutpn
    end

        xtpn= xxn-L*cos(abs(pnx));
        ytpn= yyn-L*sin(abs(pnx));
    deltaax=xrr-xtpn
    deltaay=yrr-ytpn
    pnn= atan2(deltaay,deltaax)
    mutpnn=abs(pnn);

    pnnx=pnn
    if pnn<0
        pnnx=(2*pi)+pnn
    end

    if deltaax>=0 && deltaay>=0
        xnoise=xtpn+L*cos(pnn);
        ynoise=ytpn+L*sin(pnn);
    end
    if deltaax>0 && deltaay<0
        xnoise= xtpn+L*cos((2*pi)-pnn);
        ynoise= ytpn-L*sin((2*pi)-pnn);
    end
    if deltaax<0 && deltaay>0
        xnoise= xtpn-L*cos(abs(pn));
        ynoise= ytpn+L*sin(abs(pn));
    end
    if deltaax<0 && deltaay<0
        xnoise= xtpn-L*cos(abs(pi-mutpnn));
        ynoise= ytpn-L*sin(abs(pi-mutpnn));
    end
end
```

```
deltaaxs=xrr-xnoise;
deltaays=yrr-ynoise;

snoise=sqrt((deltaaxs)^2+(deltaays)^2)

tetaa=pn-pnn
tetaax=tetaa;
if tetaa<0 || tetaa>=pi
    tetaa1=abs(tetaa);
    if tetaa1>=pi
        tetaa=2*pi-tetaa1
    else
        tetaa=tetaa1
    end
end

mutn=abs(tetaa);
waktun=((mutn/radi)*sampling)+tno;

sa=spercepatan;
tsam=tpercepatan;

scmn=snoise-(2*sa);
tcomn=(scmn/vcc)/60;
t1mn=tsam+tcomn;
totmn=(tsam*2)+tcomn;
tcmn=tsam*60+waktun;
tbmn=t1mn*60+waktun;
tamn=tbmn+tcmn-waktun;
if tbmn<=(tsam*60)
    scmn=0;
    tcomn=0;
    t1mn=((tmust/60)/2);
    totm=tmust/60;
    tcmn=t1m*60+waktun;
% waktu stop
    tbmn =t1m*60+waktun;
    tamn=tbmn+tcmn-waktun;
end
totalwaktun=tamn+waktun
end

while t>=tno && t<totalwaktun
    tt(iii+1)=t;

    if t<waktun
        if (pn>=0 && pn<=(pi/2)) || (pn>(pi/2) && pn<pi)
            if (deltaax>0 && deltaay>0)
                if pnx==pnnx
                    emr=emrr;
                    eml=emll;
                end
                if pnnx>pnx && pnnx<=(pi+pnx)
                    emr=emrr;
                    eml=-emll;
                end
            end
        end
    end
end
```

```
else
    emr=-emrr;
    eml=emll;
end

if pn>(pi/2) && pn<pi;
    emr=-emrr;
    eml=emll;
end
end

if (deltaax<0 && deltaay>=0)
    if pn>(pi/2) && pn<pi;
        if pn==pnnx
            emr=emrr;
            eml=emll;
        end
        if pnnx>pnx && pnnx<=(pi+pnx)
            emr=emrr;
            eml=-emll;
        else
            % 'kanan'
            emr=-emrr;
            eml=emll;
        end
    else
        emr=emrr;
        eml=-emll;
    end
end

if (deltaax<=0 && deltaay<=0)
    if pnnx>pnx && pnnx<=(pi+pnx)
        emr=emrr;
        eml=-emll;
    else
        emr=-emrr;
        eml=emll;
    end
    if pn>(pi/2) && pn<pi;
        emr=emrr;
        eml=-emll;
    end
end

if (deltaax>0 && deltaay<0)
    if pn>(pi/2) && pn<pi;
        if pnnx>pnx && pnnx<(pi+pnx)
            emr=emrr;
            eml=-emll;
        else
            emr=-emrr;
            eml=emll;
        end
    else
```

```

        emr=-emrr;
        eml=epll;
    end
end
er(iiii+1)=emr;
el(iiii+1)=eml;
ir1=(emr/R)-(emr/R)*exp((-R/Lk)*thh);
il1=(eml/R)-(eml/R)*exp((-R/Lk)*thh);
ir=ir1;
il=il1;
ia(iii+1)=ir;
ib(iii+1)=il;
il=il1;
Eemfr1=-emr*exp((-R/Lk)*thh);
Eemfl1=-eml*exp((-R/Lk)*thh);
Eemfr=Eemfr1;
Eemfl=Eemfl1;
end

    if (pn<0 && pn>=(-pi/2)) || (pn<(-pi/2) && pn>(-pi))
|| (pn>(pi) && pn<(3*pi/2)) || (pn>=(3*pi/2) && pn<(2*pi))
    if (deltaax>0 && deltaay>0)
        if (pn<(-pi/2) && pn>(-pi)) || (pn>(pi) &&
pn<(3*pi/2))
            if pnnx<pnx && pnnx>(pnx-pi)
                emr=-emrr;
                eml=epll;
            else
                emr=emrr;
                eml=-epll;
            end
        else
            emr=emrr;
            eml=-epll;
        end
    end
end

    if (deltaax<0 && deltaay>=0)
    if pnnx<pnx && pnnx>(pnx-pi)
        emr=-emrr;
        eml=epll;
    else
        emr=emrr;
        eml=-epll;
    end
    if (pn<(-pi/2) && pn>(-pi)) || (pn>(pi) &&
pn<(3*pi/2))
        emr=-emrr;
        eml=epll;
    end
end

    if (deltaax<=0 && deltaay<=0)
    if (pn<(-pi/2) && pn>(-pi)) || (pn>(pi) &&
pn<(3*pi/2))
        if pnx==pnnx

```

```

        emr=emrr;
        eml=emll;
    end
    if pnnx<pnx && pnnx>(pnx-pi)
        emr=-emrr;
        eml=emll;
    else
        emr=emrr;
        eml=-emll;
    end
else
    emr=-emrr;
    eml=emll;
end
end

pn<(3*pi/2))
    if (deltaax>0 && deltaay<0)
        if (pn<(-pi/2) && pn>(-pi)) || (pn>(pi) &&
            emr=emrr;
            eml=-emll;
        else
            if pnx==pnnx
                emr=emrr;
                eml=emll;
            end
            if pnnx<pnx && pnnx>(pnx-pi)
                emr=-emrr;
                eml=emll;
            else
                emr=emrr;
                eml=-emll;
            end
        end
    end
    er(iiii+1)=emr;
    el(iiii+1)=eml;
    ir1=(emr/R)-(emr/R)*exp((-R/Lk)*thh);
    il1=(eml/R)-(eml/R)*exp((-R/Lk)*thh);
    ir=ir1;
    il=il1;
    Eemfr1=-emr*exp((-R/Lk)*thh);
    Eemfl1=-eml*exp((-R/Lk)*thh);
    Eemfr=Eemfr1;
    Eemfl=Eemfl1;
end
end

if t>=waktun
    deltaaxs=xrr-xnoise;
    deltaays=yrr-ynoise;
    snoise=sqrt(((deltaaxs)^2)+((deltaays)^2));
    if snoise>(2*sa)
        sco=snoise-(2*sa);
        tco=((sco/vc))/60;
        t1=tsam+tco;

```

```
total=((2*tsam)+tco)*60;
tcc=(tsam*60)+waktun;
tbb=(t1*60)+waktun;
taa=tbb+tcc-waktun;
end
if snoise==(2*sa)
sco=0;
tco=0;
t1=tsam/2;
total=tsam
tcc=t1*60+waktun;
tbb=t1*60+waktun;
taa=tbb+tcc-waktun;
end
if snoise<(2*sa)
sco=0;
tco=0;
t1=(tmust/60)/2;
total=tmust/60
tcc=t1*60+waktun;
tbb=t1*60+waktun;
taa=tbb+tcc-waktun;
end

if t<tcc
emr=emrr;
eml=emll;
er(iiii+1)=emr;
el(iiii+1)=eml;
ir1=(emr/R)-(emr/R)*exp((-R/Lk)*th);
il1=(eml/R)-(eml/R)*exp((-R/Lk)*th);
ir=ir1;
il=il1;

Eemfr1=-emr*exp((-R/Lk)*th);
Eemfl1=-eml*exp((-R/Lk)*th);
Eemfr=Eemfr1;
Eemfl=Eemfl1;
th=th+sampling;
end

if snoise>(2*sa)
if t>=tcc && t<tbb
emr=emrr;
eml=emll;
er(iiii+1)=emr;
el(iiii+1)=eml;
ir1=(emr/R)-(emr/R)*exp((-R/Lk)*th);
il1=(eml/R)-(eml/R)*exp((-R/Lk)*th);
ir2=- (emr/R)+(emr/R)*exp((-R/Lk)*t0);
il2=- (eml/R)+(eml/R)*exp((-R/Lk)*t0);

ir=ir1+ir2;
il=il1+il2;
```

```

Eemfr1=-emr*exp((-R/Lk)*th);
Eemfl1=-eml*exp((-R/Lk)*th);
Eemfr2=emr*exp((-R/Lk)*t0);
Eemfl2=eml*exp((-R/Lk)*t0);

Eemfr=Eemfr1+Eemfr2;
Eemfl=Eemfl1+Eemfl2;
th=th+sampling;
t0=t0+sampling;
end

if t>=tbb && t<(tbb+tc1)
    emr=emrr;
    eml=emll;
    er(iiii+1)=0;
    el(iiii+1)=0;
    ir1=(emr/R)-(emr/R)*exp((-R/Lk)*th);
    il1=(eml/R)-(eml/R)*exp((-R/Lk)*th);
    ir2=- (emr/R)+(emr/R)*exp((-R/Lk)*t0);
    il2=- (eml/R)+(eml/R)*exp((-R/Lk)*t0);
    ir1off=- ((emr/R)-(emr/R)*exp((-R/Lk)*t00));
    il1off=- ((eml/R)-(eml/R)*exp((-R/Lk)*t00));
    ir=ir1+ir2+ir1off;
    il=il1+il2+il1off;

    Eemfr1=-emr*exp((-R/Lk)*th);
    Eemfl1=-eml*exp((-R/Lk)*th);
    Eemfr2=emr*exp((-R/Lk)*t0);
    Eemfl2=eml*exp((-R/Lk)*t0);
    Eemfr1off=emr*exp((-R/Lk)*t00);
    Eemfl1off=eml*exp((-R/Lk)*t00);

    Eemfr=Eemfr1+Eemfr2+Eemfr1off;
    Eemfl=Eemfl1+Eemfl2+Eemfl1off;
    th=th+sampling;
    t0=t0+sampling;
    t00=t00+sampling;
end

if t>=(tbb+tc1)
    emr=emrr;
    eml=emll;
    er(iiii+1)=0;
    el(iiii+1)=0;
    ir1=(emr/R)-(emr/R)*exp((-R/Lk)*th);
    il1=(eml/R)-(eml/R)*exp((-R/Lk)*th);
    ir2=- (emr/R)+(emr/R)*exp((-R/Lk)*t0);
    il2=- (eml/R)+(eml/R)*exp((-R/Lk)*t0);
    ir1off=- ((emr/R)-(emr/R)*exp((-R/Lk)*t00));
    il1off=- ((eml/R)-(eml/R)*exp((-R/Lk)*t00));
    ir2off=(emr/R)-(emr/R)*exp((-R/Lk)*t000);
    il2off=(eml/R)-(eml/R)*exp((-R/Lk)*t000);

    ir=ir1+ir2+ir1off+ir2off;
    il=il1+il2+il1off+il2off;

```

```

Eemfr1=-emr*exp((-R/Lk)*th);
Eemfl1=-eml*exp((-R/Lk)*th);
Eemfr2=emr*exp((-R/Lk)*t0);
Eemfl2=eml*exp((-R/Lk)*t0);
Eemfr1off=emr*exp((-R/Lk)*t00);
Eemfl1off=eml*exp((-R/Lk)*t00);
Eemfr2off=-emr*exp((-R/Lk)*t000);
Eemfl2off=-eml*exp((-R/Lk)*t000);
Eemfr=Eemfr1+Eemfr2+Eemfr1off+Eemfr2off;
Eemfl=Eemfl1+Eemfl2+Eemfl1off+Eemfl2off;
th=th+sampling;
t0=t0+sampling;
t00=t00+sampling;
t000=t000+sampling;
end
end

if snoise<(2*sa)
  if t>=tbb && t<(tbb+tc1)
    emr=emrr;
    eml=emll;
    er(iiii+1)=0;
    el(iiii+1)=0;
    ir1=(emr/R)-(emr/R)*exp((-R/Lk)*th);
    il1=(eml/R)-(eml/R)*exp((-R/Lk)*th);
    ir2=-(emr/R)+(emr/R)*exp((-R/Lk)*t0);
    il2=-(eml/R)+(eml/R)*exp((-R/Lk)*t0);
    ir1off=-((emr/R)-(emr/R)*exp((-R/Lk)*t00));
    il1off=-((eml/R)-(eml/R)*exp((-R/Lk)*t00));
    ir=ir1+ir2+ir1off;
    il=il1+il2+il1off;

    Eemfr1=-emr*exp((-R/Lk)*th);
    Eemfl1=-eml*exp((-R/Lk)*th);
    Eemfr2=emr*exp((-R/Lk)*t0);
    Eemfl2=eml*exp((-R/Lk)*t0);
    Eemfr1off=emr*exp((-R/Lk)*t00);
    Eemfl1off=eml*exp((-R/Lk)*t00);

    Eemfr=Eemfr1+Eemfr2+Eemfr1off;
    Eemfl=Eemfl1+Eemfl2+Eemfl1off;
    th=th+sampling;
    t0=t0+sampling;
    t00=t00+sampling;
  end

  if t>=(tbb+tc1)
    emr=emrr;
    eml=emll;
    er(iiii+1)=0;
    el(iiii+1)=0;
    ir1=(emr/R)-(emr/R)*exp((-R/Lk)*th);
    il1=(eml/R)-(eml/R)*exp((-R/Lk)*th);
    ir2=-(emr/R)+(emr/R)*exp((-R/Lk)*t0);

```



```

il2=- (eml/R) + (eml/R) *exp((-R/Lk) *t0);
ir1off=- ((emr/R) - (emr/R) *exp((-R/Lk) *t00));
il1off=- ((eml/R) - (eml/R) *exp((-R/Lk) *t00));
ir2off=(emr/R) - (emr/R) *exp((-R/Lk) *t000);
il2off=(eml/R) - (eml/R) *exp((-R/Lk) *t000);

ir=ir1+ir2+ir1off+ir2off;
il=il1+il2+il1off+il2off;

Eemfr1=-emr*exp((-R/Lk) *th);
Eemfl1=-eml*exp((-R/Lk) *th);
Eemfr2=emr*exp((-R/Lk) *t0);
Eemfl2=eml*exp((-R/Lk) *t0);
Eemfr1off=emr*exp((-R/Lk) *t00);
Eemfl1off=eml*exp((-R/Lk) *t00);
Eemfr2off=-emr*exp((-R/Lk) *t000);
Eemfl2off=-eml*exp((-R/Lk) *t000);
Eemfr=Eemfr1+Eemfr2+Eemfr1off+Eemfr2off;
Eemfl=Eemfl1+Eemfl2+Eemfl1off+Eemfl2off;
th=th+sampling;
t0=t0+sampling;
t00=t00+sampling;
t000=t000+sampling;
end
end
end

kg=r2/r1;
wmr=Eemfr/Ks;
wml=Eemfl/Ks;
Tmr=ir*Ki;
Tml=il*Ki;
TLr=Tmr*kg;
TLl=Tml*kg;
wLr=(1/kg) *wmr;
wLl=(1/kg) *wml;
Jhub=Mhub*rin^2/2;
Jwheel=Mwheel*(rin^2+rout^2)/2;
Jrobot=Mrobot*rout^2/2;
J=Jrobot;
JL=Jhub+Jwheel+Jrobot;
Jr=JL+kg^2*Jmr;
Jl=JL+kg^2*Jml;
wLrdot=TLr/Jr;
wLldot=TLl/Jl;
fr=(Jr*wLrdot)/rout;
fl=(Jl*wLldot)/rout;
F=(fr+fl)/2;
vdott=((1/(2*m))*fr) + ((1/(2*m))*fl);
vdotr(iii+1)=vdott;
wdot=((1/(2*b*J))*fr - (1/(2*b*J))*fl);
wdotr(iii+1)=wdot;
wt=wt+wdot*t;
waw(iii+1)=wt;
if t<waktun
vt=v0+vdott*thh;

```

```
sss=s4+vt*thh+0.5*vdott*(thh^2);
sbay=sbay0+vt*thh+0.5*vdott*(thh^2);
sbay0=sbay;
vvtw1=vt;
s0=sss;

if (pn>=0 && pn<=(pi/2)) || (pn>(pi/2) && pn<pi)
  if (deltaax>0 && deltaay>0)
    ten=ten+radi;
    if pnx==pnnx
      teth=pn; 'lurus'
      sdt=pn;
    end
    if pnnx>pnx && pnnx<=(pi+pnx)
      teth=pn+ten; 'kiri'
      sdt=teth;
    else
      teth=pn-ten; 'kanan'
      sdt=teth;
    end
  end
  if pn>(pi/2) && pn<pi;
    teth=pn-ten; 'kanan'
    sdt=teth;
  end
end

if (deltaax<0 && deltaay>=0)
  ten=ten+radi;
  teth=pn+ten; 'kiri'
  sdt=pi+teth;

  if pn>(pi/2) && pn<pi;
    if pnx==pnnx
      teth=pn; 'lurus'
      sdt=pn;
    end
    if pnnx>pnx && pnnx<=(pi+pnx)
      teth=pn+ten; 'kiri'
      sdt=pi+teth;
    else
      teth=pn-ten; 'kanan'
      sdt=pi+teth;
    end
  end
end

if (deltaax<=0 && deltaay<=0)
  ten=ten+radi;
  if pnnx>pnx && pnnx<=(pi+pnx)
    teth=pn+ten; 'kiri'
    sdt=teth-pi;
  else
    teth=pn-ten; 'kanan'
    sdt=teth-pi;
  end
end
```

```

        end

        if pn>(pi/2) && pn<pi;
            teth=pn+ten;
            sdt=teth-pi;
        end
    end

    if (deltaax>0 && deltaay<0)
        ten=ten+radi;
        teth=pn-ten;
        sdt=teth;

        if pn>(pi/2) && pn<pi;
            if pnnx>pnx && pnnx<(pi+pnx)
                teth=pn+ten;
                sdt=teth;
            else
                teth=pn-ten;
                sdt=teth;
            end
        end
    end
end

if (pn<0 && pn>=(-pi/2)) || (pn<(-pi/2) && pn>(-pi))
|| (pn>(pi) && pn<(3*pi/2)) || (pn>=(3*pi/2) && pn<(2*pi))
    if (deltaax>0 && deltaay>0)
        ten=ten+radi;
        teth=pn+ten;
        sdt=teth;

        if (pn<(-pi/2) && pn>(-pi)) || (pn>(pi) &&
pn<(3*pi/2))
            if pnnx<pnx && pnnx>(pnx-pi)
                teth=pn-ten;
                sdt=teth;
            else
                teth=pn+ten;
                sdt=teth;
            end
        end
    end

    if (deltaax<0 && deltaay>=0)
        ten=ten+radi;
        if pnnx<pnx && pnnx>(pnx-pi)
            teth=pn-ten;
            sdt=pi+teth;
        else
            teth=pn+ten;
            sdt=pi+teth;
        end
    end
end

```

```
pn<(3*pi/2))          if (pn<(-pi/2) && pn>(-pi)) || (pn>(pi) &&
                      teth=pn-ten;
                      sdt=pi+teth;
                      end
                      end

                      if (deltaax<=0 && deltaay<=0)
                        ten=ten+radi;
                        teth=pn-ten;
                        sdt=pi+teth;

pn<(3*pi/2))          if (pn<(-pi/2) && pn>(-pi)) || (pn>(pi) &&
                      if pnx==pnnx
                        teth=pn;
                        sdt=pn;
                      end
                      if pnnx<pnx && pnnx>(pnx-pi)
                        teth=pn-ten;
                        sdt=pi+teth;
                      else
                        teth=pn+ten;
                        sdt=pi+teth;
                      end
                      end
                      end

                      if (deltaax>0 && deltaay<0)
                        ten=ten+radi;
                        if pnx==pnnx
                          teth=pn;
                          sdt=pn;
                        end
                        if pnnx<pnx && pnnx>(pnx-pi)
                          teth=pn-ten;
                          sdt=teth;
                        else
                          teth=pn+ten;
                          sdt=teth;
                        end

pn<(3*pi/2))          if (pn<(-pi/2) && pn>(-pi)) || (pn>(pi) &&
                      teth=pn+ten;
                      sdt=teth;
                      end
                      end
                      end
                      end
                      mutteth=abs(teth);
                      if deltaax>=0 && deltaay>=0
                        xnoise=xtpn+L*cos(teth);
                        ynoise=ytpn+L*sin(teth);
                      end
```

```
if deltaax>0 && deltaay<0
    xnoise= xtpn+L*cos((2*pi)-teth);
    ynoise= ytpn-L*sin((2*pi)-teth);
end
if deltaax<0 && deltaay>0
    xnoise= xtpn-L*cos(abs(teth));
    ynoise= ytpn+L*sin(abs(teth));
end
if deltaax<0 && deltaay<0
    xnoise= xtpn-L*cos(abs(pi-mutteth));
    ynoise= ytpn-L*sin(abs(pi-mutteth));
end
xnoises(iii+1)=xnoise;
y noises(iii+1)=y noise;

if t>=waktun
    if t<tcc
        vt=v0+vdott*th;
        sss=s0+vt*th+0.5*vdott*(th^2);
        sbay=sbay0+vt*th+0.5*vdott*(th^2);
        sbay1=sbay;
        vt1=vt;
        s5=sss;
    end
    if snoise>(2*sa)
        if t>=tcc && t<tbb
            vt=vt1+vdott*tv;
            sss=s5+vt*tv+0.5*vdott*(tv^2);
            sbay=sbay1+vt*tv+0.5*vdott*(tv^2);
            sbay2=sbay;
            vt2=vt;
            s6=sss;
            tv=tv+sampling;
        end
        if t>=tbb && t<(tbb+tc1)
            vt=vt2+vdott*tvv;
            sss=s6+vt*tvv-0.5*vdott*(tvv^2);
            sbay=sbay2+vt*tvv-0.5*vdott*(tvv^2);
            sbay3=sbay;
            vt3=vt;
            s7=sss;
            tvv=tvv+sampling;
        end
        if t>=(tbb+tc1)
            vt0=v0;
            vt=vt0+vdott*tvvv;
            sss=s7+vt*tvvv-0.5*vdott*(tvvv^2);
            sbay=sbay3+vt*tvvv-0.5*vdott*(tvvv^2);
            sbay4=sbay;
            tvvv=tvvv+sampling;
        end
    end
end

if snoise<(2*sa)
    if t>=tbb && t<(tbb+tc1)
        vt=vt1+vdott*tvvii;
```

```

        sss=s5+vt*tvvii-0.5*vdott*(tvvii^2);
        sbay=sbay1+vt*tvvii-0.5*vdott*(tvvii^2);
        sbay5=sbay;
        vt5=vt;
        s8=sss;
        tvvii=tvvii+sampling;
    end
    if t>=(tbb+tc1)
        vt0=v0;
        vt=vt0+vdott*tvvvii;
        sss=s8+vt*tvvvii-0.5*vdott*(tvvvii^2);
        sbay=sbay5+vt*tvvvii-0.5*vdott*(tvvvii^2);
        sbay6=sbay;
        tvvvii=tvvvii+sampling;
    end
end
end
v(iii+1)=vt;
ss(iii+1)=sss;
iii=1+iii;
t=t+sampling;
thh=thh+sampling;
ssbay(bay+1)=sbay;
    tetha(ibay+1)=teth;
    tettha(ibay+1)=ten;

if (deltaax>=0 && deltaay>0) || (deltaax>0 && deltaay<=0)
    xakhir=xtpn+L*cos(teth)+sbay*cos(sdt);
    yakhir=ytpn+L*sin(teth)+sbay*sin(sdt);
end
if (deltaax<0 && deltaay>0) || (deltaax<=0 && deltaay<=0)
    xakhir=xtpn+L*cos(teth)-sbay*cos(sdt);
    yakhir=ytpn+L*sin(teth)-sbay*sin(sdt);
end

x1(iiii+1)=xakhir;
y1(iiii+1)=yakhir;
tbayang(ibay+1)=tbay;
tbay=tbay+sampling;
ibay=ibay+1;
iiii=iiii+1;
end
xgangguan=xxn
ygangguan=yyn

pos=suduttt
posisigangguan=((suduttt/(2*pi))*360)
xterakhir=xakhir
yterakhir=yakhir

%% Output

figure(1);
subplot(4,1,1);
plot(tbayang,er);

```

```
grid;
xlabel('Waktu (s)'), ylabel('Tegangan (V)');
title('Hubungan tegangan motor kanan terhadap waktu')

subplot(4,1,2);
plot(tbayang,el);
grid;
xlabel('Waktu (s)'), ylabel('Tegangan (V)');
title('Hubungan tegangan motor kiri terhadap waktu')

subplot(4,1,3);
plot(tt,v);
grid;
xlabel('Waktu (s)'), ylabel('Kecepatan(m/s)');
title('Hubungan kecepatan robot terhadap waktu')

subplot(4,1,4);
plot(tt,ss);
grid;
xlabel('Waktu (s)'), ylabel('jarak');
title('Hubungan jarak terhadap waktu')

figure(2)
subplot(2,2,1);
plot(tt,vdotr);
grid;
xlabel('Waktu (s)'), ylabel('Percepatan');
title('Hubungan percepatan robot terhadap waktu')

subplot(2,2,2);
plot(tt,v);
grid;
xlabel('Waktu (s)'), ylabel('Kecepatan (m/s)');
title('Hubungan kecepatan robot terhadap waktu')

subplot(2,2,3);
plot(tt,ss);
grid;
xlabel('Waktu (s)'), ylabel('jarak');
title('Hubungan jarak terhadap waktu')

subplot(2,2,4);
plot(x1,y1);
grid;
xlabel('X'), ylabel('Y');
title('Koordinat posisi robot')

end
```

## **LAMPIRAN C**

---

**PROGRAM SIMULASI PERGERAKAN ROBOT  
MOBIL BERODA MELALUI TIGA KOORDINAT  
LAIN**



```
em = input('Masukkan tegangan motor (tegangan motor kanan=tegangan
motor kiri)= ');
emrr=em;
er(1)=0;
emll=em;
el(1)=0;

x = input('masukkan titik awal sumbu x = ');
x1(1)=x;
xaw=x;
y = input('masukkan titik awal sumbu y = ');
y1(1)=y;
yaw=y;
xrr = input('masukkan sumbu x selanjutnya= ');
yrr = input('masukkan sumbu y selanjutnya= ');
xrrr = input('masukkan sumbu x selanjutnya = ');
yrrr = input('masukkan sumbu y selanjutnya= ');

phi = (pi/2);
tetha(1)=phi;

RPM=263;
Vmin=3;
Vmaks=8;

if em<=Vmin
    'Tegangan diberikan pada motor terlalu kecil '
    Vmin
    new3;
elseif em>Vmaks
    'Tegangan diberikan pada motor terlalu besar '
    Vmaks
    new3;
elseif em>Vmin && em<=Vmaks

selisih=Vmaks-Vmin;
kec=RPM/selisih;
kenaikan=emrr-Vmin;

Ks=kec*kenaikan;
r2=25e-3;
r1=1e-3;
Ki=3e-1;
R=10000;
Lk=0.63;
Mhub=0.025;
Mwheel=0.05;
Mrobot=3;
m=Mrobot;
rin=0.03;
rout=0.05;
Jmr=0.56e7;
Jml=0.56e7;
b=0.05;
L=0.1;
```

```
format long
xtp=xaw-(L*cos(phi));
ytp=yaw-(L*sin(phi));
deltax=xrr-xtp;
deltay=yrr-ytp;
p=atan2((deltay),(deltax))

xharus=xtp+L*cos(p);
yharus=ytp+L*sin(p);
deltaxs=xrr-xharus;
deltays=yrr-yharus;
smust=sqrt((deltaxs)^2+(deltays)^2)

sampling=0.01;
vcmot=Ks*(2*pi*r1)/60;
wcmot=vcmot/r2;
vc=wcmot*rout

while vt<=vc
    ttti(iii+1)=ti;
    emr=emrr;
    eml=emll;

kg=r2/r1;
ir1=(emr/R)-(emr/R)*exp((-R/Lk)*ti);
il1=(eml/R)-(eml/R)*exp((-R/Lk)*ti);
ir=ir1;
il=il1;
ia(iii+1)=ir;
ib(iii+1)=il;
il=il1;
Eemfr1=-emr*exp((-R/Lk)*ti);
Eemfl1=-eml*exp((-R/Lk)*ti);
Eemfr=Eemfr1;
Eemfl=Eemfl1;
wmr=Eemfr/Ks;
wml=Eemfl/Ks;
Tmr=ir*Ki;
Tml=il*Ki;
TLr=Tmr*kg;
TLl=Tml*kg;
wLr=(1/kg)*wmr;
wLl=(1/kg)*wml;
Jhub=Mhub*rin^2/2;
Jwheel=Mwheel*(rin^2+rout^2)/2;
Jrobot=Mrobot*rout^2/2;
J=Jrobot;
JL=Jhub+Jwheel+Jrobot;
Jr=JL+kg^2*Jmr;
Jl=JL+kg^2*Jml;
wLrdot=TLr/Jr;
wLldot=TLl/Jl;
fr=(Jr*wLrdot)/rout;
fl=(Jl*wLldot)/rout;
F=(fr+fl)/2;
```

```
vdoot=((1/(2*m))*fr) + ((1/(2*m))*fl);

vt=v0+vdoot*ti;
s=s0+vt*ti+0.5*vdoot*(ti^2);

vaw(iii+1)=vt;
ssaw(iii+1)=s;
iii=1+iii;
ti=ti+sampling;
if s<smust
    tmust=ti;
end
end
tper=ti;
tpercepatan=ti/60
spercepatan=s
vcc=vt
teta=phi-p
tetax=teta;
if teta<0 || teta >=pi
    tetal=abs(teta);
    if tetal>=pi
        teta=2*pi-tetal
    else
        teta=tetal
    end
end

kel=2*pi*(2*b);
fi=(vcc/kel)*(2*pi);
rad1=fi*sampling;
radi=rad1*2;
n=teta/radi;
mut=abs(teta);
waktu=(mut/radi)*sampling;

sa=spercepatan;
tsam=tpercepatan;
scm=smust-(2*sa);
tcom=(scm/vcc)/60;
t1m=tsam+tcom;
totm=(tsam*2)+tcom;
tcm=tsam*60+waktu;
tbm=t1m*60+waktu;
tam=tbm+tcm-waktu;
if tbm<=(tsam*60)
    scm=0;
    tcom=0;
    t1m=(tmust/60)/2;
    totm=tmust/60;
    tcm=t1m*60+waktu;
    tbm =t1m*60+waktu;
    tam=tbm+tcm-waktu;
end
totalwaktu=tam;
```

```

%% MULAI PROGRAM
while t<=totalwaktu
    tt(iii+1)=t;
    tbay=tbay+sampling;
    tbayang(ibay+1)=tbay;
    if t<waktu
        if (deltax>=0 & deltay>0) || (deltax>0 & deltay<=0)
'kanan'
            emr=-emrr;
            eml=epll;
            er(iiii+1)=emr;
            el(iiii+1)=eml;
            ir1=(emr/R) - (emr/R)*exp((-R/Lk)*t);
            il1=(eml/R) - (eml/R)*exp((-R/Lk)*t);
            ir=ir1;
            il=il1;
            ia(iii+1)=ir;
            ib(iii+1)=il;
            il=il1;
            Eemfr1=-emr*exp((-R/Lk)*t);
            Eemfl1=-eml*exp((-R/Lk)*t);
            Eemfr=Eemfr1;
            Eemfl=Eemfl1;

        end
        if (deltax<0 & deltay>0) || (deltax<=0 & deltay<=0)
'kiri'
            emr=emrr;
            eml=-epll;
            er(iiii+1)=emr;
            el(iiii+1)=eml;
            ir1=(emr/R) - (emr/R)*exp((-R/Lk)*t);
            il1=(eml/R) - (eml/R)*exp((-R/Lk)*t);
            ir=ir1;
            il=il1;
            Eemfr1=-emr*exp((-R/Lk)*t);
            Eemfl1=-eml*exp((-R/Lk)*t);
            Eemfr=Eemfr1;
            Eemfl=Eemfl1;

        end
    end
    if p==phi
        waktu=0;
        teth=phi;
        sdt=teth;
    end

    if t>=waktu
        deltaxs=xrr-xharus;
        deltays=yrr-yharus;
        s=sqrt(((deltaxs)^2)+((deltays)^2));
        if s>(2*sa)
            sc=s-(2*sa);
            tco=(sc/vcc)/60;
            t1=tsam+tco;
            tot=(tsam*2)+tco;
        end
    end
end

```

```
        tc=tsam*60+waktu;
        tb=t1*60+waktu;
        ta=tb+tc-waktu;
end
if s==(2*sa)
    sc=0;
    tco=0;
    t1=tsam/2;
    tot=tsam;
    tc=t1*60+waktu;
    tb=tsam/2+waktu;
    ta=tb+tc-waktu;
end
if s<(2*sa)
    sc=0;
    tco=0;
    t1=(tmust/60)/2;
    tot=tmust/60;
    tc=t1*60+waktu;
    tb =t1*60+waktu;
    ta=tb+tc-waktu;
end

    if t<tc
        emr=emrr;
        eml=emll;
        er(iiii+1)=emr;
        el(iiii+1)=eml;
        ir1=(emr/R) - (emr/R)*exp((-R/Lk)*ts);
        il1=(eml/R) - (eml/R)*exp((-R/Lk)*ts);
        ir=ir1;
        il=il1;
        Eemfr1=-emr*exp((-R/Lk)*ts);
        Eemfl1=-eml*exp((-R/Lk)*ts);
        Eemfr=Eemfr1;
        Eemfl=Eemfl1;
        ts=ts+sampling;
end

if s>(2*sa)
    if t>=tc && t<tb
        emr=emrr;
        eml=emll;
        er(iiii+1)=emr;
        el(iiii+1)=eml;
        ir1=(emr/R) - (emr/R)*exp((-R/Lk)*ts);
        il1=(eml/R) - (eml/R)*exp((-R/Lk)*ts);
        ir2=- (emr/R) + (emr/R)*exp((-R/Lk)*t0);
        il2=- (eml/R) + (eml/R)*exp((-R/Lk)*t0);
        ir=ir1+ir2;
        il=il1+il2;
        Eemfr1=-emr*exp((-R/Lk)*ts);
        Eemfl1=-eml*exp((-R/Lk)*ts);
        Eemfr2=emr*exp((-R/Lk)*t0);
        Eemfl2=eml*exp((-R/Lk)*t0);
```

```

Eemfr=Eemfr1+Eemfr2;
Eemfl=Eemfl1+Eemfl2;
ts=ts+sampling;
t0=t0+sampling;
end

if t>=tb && t<(tb+tc-waktu)
emr=emrr;
eml=emll;
er(iiii+1)=0;
el(iiii+1)=0;
ir1=(emr/R)-(emr/R)*exp((-R/Lk)*ts);
il1=(eml/R)-(eml/R)*exp((-R/Lk)*ts);
ir2=- (emr/R)+(emr/R)*exp((-R/Lk)*t0);
il2=- (eml/R)+(eml/R)*exp((-R/Lk)*t0);
ir1off=- ((emr/R)-(emr/R)*exp((-R/Lk)*t00));
il1off=- ((eml/R)-(eml/R)*exp((-R/Lk)*t00));
ir=ir1+ir2+ir1off;
il=il1+il2+il1off;
Eemfr1=-emr*exp((-R/Lk)*ts);
Eemfl1=-eml*exp((-R/Lk)*ts);
Eemfr2=emr*exp((-R/Lk)*t0);
Eemfl2=eml*exp((-R/Lk)*t0);
Eemfr1off=emr*exp((-R/Lk)*t00);
Eemfl1off=eml*exp((-R/Lk)*t00);

Eemfr=Eemfr1+Eemfr2+Eemfr1off;
Eemfl=Eemfl1+Eemfl2+Eemfl1off;
t0=t0+sampling;
t00=t00+sampling;
ts=ts+sampling;
end

if t>=(tb+tc-waktu)
emr=emrr;
eml=emll;
er(iiii+1)=0;
el(iiii+1)=0;
ir1=(emr/R)-(emr/R)*exp((-R/Lk)*ts);
il1=(eml/R)-(eml/R)*exp((-R/Lk)*ts);
ir2=- (emr/R)+(emr/R)*exp((-R/Lk)*t0);
il2=- (eml/R)+(eml/R)*exp((-R/Lk)*t0);
ir1off=- ((emr/R)-(emr/R)*exp((-R/Lk)*t00));
il1off=- ((eml/R)-(eml/R)*exp((-R/Lk)*t00));
ir2off=(emr/R)-(emr/R)*exp((-R/Lk)*t000);
il2off=(eml/R)-(eml/R)*exp((-R/Lk)*t000);

ir=ir1+ir2+ir1off+ir2off;
il=il1+il2+il1off+il2off;

Eemfr1=-emr*exp((-R/Lk)*ts);
Eemfl1=-eml*exp((-R/Lk)*ts);
Eemfr2=emr*exp((-R/Lk)*t0);
Eemfl2=eml*exp((-R/Lk)*t0);
Eemfr1off=emr*exp((-R/Lk)*t00);

```

```

Eemfl1off=eml*exp((-R/Lk)*t00);
Eemfr2off=-emr*exp((-R/Lk)*t000);
Eemfl2off=-eml*exp((-R/Lk)*t000);
Eemfr=Eemfr1+Eemfr2+Eemfr1off+Eemfr2off;
Eemfl=Eemfl1+Eemfl2+Eemfl1off+Eemfl2off;
ts=ts+sampling;
t0=t0+sampling;
t00=t00+sampling;
t000=t000+sampling;
end
end

if s<(2*sa)
    if t>=tb && t<(tb+tc-waktu)
        emr=emrr;
        eml=eml1;
        er(iiii+1)=0;
        el(iiii+1)=0;
        ir1=(emr/R)-(emr/R)*exp((-R/Lk)*ts);
        il1=(eml/R)-(eml/R)*exp((-R/Lk)*ts);
        ir2=-(emr/R)+(emr/R)*exp((-R/Lk)*t0);
        il2=-(eml/R)+(eml/R)*exp((-R/Lk)*t0);
        ir1off=-((emr/R)-(emr/R)*exp((-R/Lk)*t00));
        il1off=-((eml/R)-(eml/R)*exp((-R/Lk)*t00));
        ir=ir1+ir2+ir1off;
        il=il1+il2+il1off;

        Eemfr1=-emr*exp((-R/Lk)*ts);
        Eemfl1=-eml*exp((-R/Lk)*ts);
        Eemfr2=emr*exp((-R/Lk)*t0);
        Eemfl2=eml*exp((-R/Lk)*t0);
        Eemfr1off=emr*exp((-R/Lk)*t00);
        Eemfl1off=eml*exp((-R/Lk)*t00);

        Eemfr=Eemfr1+Eemfr2+Eemfr1off;
        Eemfl=Eemfl1+Eemfl2+Eemfl1off;
        ts=ts+sampling;
        t0=t0+sampling;
        t00=t00+sampling;
    end

    if t>=(tb+tc-waktu)
        emr=0;
        eml=0;
        er(iiii+1)=0;
        el(iiii+1)=0;
        ir1=(emr/R)-(emr/R)*exp((-R/Lk)*ts);
        il1=(eml/R)-(eml/R)*exp((-R/Lk)*ts);
        ir2=-(emr/R)+(emr/R)*exp((-R/Lk)*t0);
        il2=-(eml/R)+(eml/R)*exp((-R/Lk)*t0);
        ir1off=-((emr/R)-(emr/R)*exp((-R/Lk)*t00));
        il1off=-((eml/R)-(eml/R)*exp((-R/Lk)*t00));
        ir2off=(emr/R)-(emr/R)*exp((-R/Lk)*t000);
        il2off=(eml/R)-(eml/R)*exp((-R/Lk)*t000);
    end
end

```

```

        ir=ir1+ir2+ir1off+ir2off;
        il=il1+il2+il1off+il2off;

        Eemfr1=-emr*exp((-R/Lk)*ts);
        Eemfl1=-eml*exp((-R/Lk)*ts);
        Eemfr2=emr*exp((-R/Lk)*t0);
        Eemfl2=eml*exp((-R/Lk)*t0);
        Eemfr1off=emr*exp((-R/Lk)*t00);
        Eemfl1off=eml*exp((-R/Lk)*t00);
        Eemfr2off=-emr*exp((-R/Lk)*t000);
        Eemfl2off=-eml*exp((-R/Lk)*t000);
        Eemfr=Eemfr1+Eemfr2+Eemfr1off+Eemfr2off;
        Eemfl=Eemfl1+Eemfl2+Eemfl1off+Eemfl2off;
        ts=ts+sampling;
        t0=t0+sampling;
        t00=t00+sampling;
        t000=t000+sampling;
    end
end
end

kg=r2/r1;
wmr=Eemfr/Ks;
wml=Eemfl/Ks;
Tmr=ir*Ki;
Tml=il*Ki;
TLr=Tmr*kg;
TLl=Tml*kg;
wLr=(1/kg)*wmr;
wLl=(1/kg)*wml;
Jhub=Mhub*rin^2/2;
Jwheel=Mwheel*(rin^2+rout^2)/2;
Jrobot=Mrobot*rout^2/2;
J=Jrobot;
JL=Jhub+Jwheel+Jrobot;
Jr=JL+kg^2*Jmr;
Jl=JL+kg^2*Jml;
wLrdot=TLr/Jr;
wLldot=TLl/Jl;
fr=(Jr*wLrdot)/rout;
fl=(Jl*wLldot)/rout;
F=(fr+fl)/2;
vdott=((1/(2*m))*fr) + ((1/(2*m))*fl);
vdotrr(iii+1)=vdott;
wdoot=((1/(2*b*J))*fr - (1/(2*b*J))*fl);
wdotrr(iii+1)=wdoot;
wt=wt+wdoot*t;

if t<waktu
    vvt=v0+vdott*t;
    sss=s0+vvt*t+0.5*vdott*(t^2);
    vvtw1=vvt;
    s0=sss;
    if (phi>=0 && phi<=(pi/2)) || (phi>(pi/2) && phi<pi)
        if (deltax>0 && deltay>0)
            te=te+radi;

```



```

        teth=phi-te;
        sdt=teth;
    end
    if (deltax<0 && deltay>=0)
        te=te+radi;
        teth=phi+te;
        sdt=teth;
    end
    if (deltax<=0 && deltay<=0)
        te=te+radi;
        teth=phi-te;
        sdt=teth;
        if phi>=(pi/2) && phi<(pi)
            teth=phi+te;
            sdt=teth;
        end
    end
    if (deltax>0 && deltay<0)
        te=te+radi;
        teth=phi-te;
        sdt=teth;
    end
end
pi))
if (phi<0 && phi>=(-pi/2)) || (phi<(-pi/2) && phi>(-
    if (deltax>0 && deltay>0)
        te=te+radi;
        teth=phi+te;
        sdt=teth;
    end
    if (deltax<0 && deltay>=0)
        te=te+radi;
        teth=phi+te;
        sdt=teth;
        if (phi<(-pi/2) && phi>(-pi))
            teth=phi-te;
            sdt=teth;
        end
    end
    if (deltax<=0 && deltay<=0)
        te=te+radi;
        teth=phi-te;
        sdt=teth;
    end
    if (deltax>0 && deltay<0)
        te=te+radi;
        teth=phi-te;
        sdt=teth;
        if (phi<(-pi/2) && phi>(-pi))
            teth=te+phi;
            sdt=teth;
        end
    end
end
end
xharus=xtp+L*cos(teth);

```

```

        yharus=ytp+L*sin(teth);

    if t>=waktu
        if t<tc
            vvt=v0+vdott*tss;
            sss=s0+vvt*tss+0.5*vdott*(tss^2);
            vvt1=vvt;
            ss1=sss;
            tss=tss+sampling;
        end
        if s>(2*sa)
            if t>=tc && t<tb
                vvt=vvt1+vdott*tv;
                sss=ss1+vvt*tv+0.5*vdott*(tv^2);
                vvt2=vvt;
                ss2=sss;
                tv=tv+sampling;
            end
            if t>=tb && t<(tb+tc-waktu)
                vvt=vvt2+vdott*tvv;
                sss=ss2+vvt*tvv-0.5*vdott*(tvv^2);
                vvt3=vvt;
                ss3=sss;
                tvv=tvv+sampling;
            end
            if t>=(tb+tc-waktu)
                vt0=v0;
                vvt=vt0+vdott*tvvv;
                sss=ss3+vvt*tvvv-0.5*vdott*(tvvv^2);
                tvvv=tvvv+sampling;
            end
        end
    end

    if s<=(2*sa)
        if t>=tb && t<(tb+tc-waktu)
            vvt=vvt1+vdott*tvvi;
            sss=ss1+vvt*tvvi-0.5*vdott*(tvvi^2);
            vvt2=vvt;
            ss3=sss;
            tvvi=tvvi+sampling;
        end
        if t>=(tb+tc-waktu)
            vt0=v0;
            vvt=vt0+vdott*tvvvi;
            sss=ss3+vvt*tvvvi-0.5*vdott*(tvvvi^2);
            tvvvi=tvvvi+sampling;
        end
    end
end

waw(iii+1)=wt;
tetha(iii+1)=teth;
tettha(iii+1)=te;

v(iii+1)=vvt;

```

```

ss(iii+1)=sss;
iii=iii+1;
t=t+sampling;

    if (deltaxs>=0 && deltays>0) || (deltaxs>0 && deltays<=0)
        xakhir=xtp+L*cos(teth)+sss*cos(sdt);
        yakhir=ytp+L*sin(teth)+sss*sin(sdt);
    end
    if (deltaxs<0 && deltays>0) || (deltaxs<=0 && deltays<=0)
        xakhir=xtp+L*cos(teth)+sss*cos(sdt);
        yakhir=ytp+L*sin(teth)+sss*sin(sdt);
    end

x1(iiii+1)=xakhir;
y1(iiii+1)=yakhir;
iiii=iiii+1;
ibay=ibay+1;
end

s4=sss
if t>=totalwaktu;
    xn=xakhir
    xxn=xn;
    yn=yakhir
    yyn=yn;

    pn= sdt
    tetha(ibay+1)=pn;
    tettha(ibay+1)=0;
    mutpn=abs(pn);
    pnx= pn
    if pnx<0
        pnx=(2*pi)-mutpn
    end

    if deltaxs>=0 && deltays>=0
        xtpn= xxn-L*cos(pn);
        ytpn= yyn-L*sin(pn);
    end
    if deltaxs>0 && deltays<0
        xtpn= xxn-L*cos(abs((2*pi)-pn));
        ytpn= yyn+L*sin(abs((2*pi)-pn));
    end
    if deltaxs<0 && deltays>0
        xtpn= xxn-L*cos(abs(pn));
        ytpn= yyn-L*sin(abs(pn));
    end
    if deltaxs<0 && deltays<0
        xtpn= xxn+L*cos(abs(pi-mutpn));
        ytpn= yyn+L*sin(abs(pi-mutpn));
    end
    end
deltaax=xrrr-xtpn
deltaay=yrrr-ytpn
pnn= atan2(deltaay,deltaax)
mutpnn=abs(pnn);

```

```
pnnx=pnn
if pnn<0
    pnnx=(2*pi)+pnn
end

if deltaax>=0 && deltaay>=0
    x2=xtpn+L*cos(pnn);
    y2=ytpn+L*sin(pnn);
end
if deltaax>0 && deltaay<0
    x2= xtpn+L*cos(abs((2*pi)-pnn));
    y2= ytpn-L*sin(abs((2*pi)-pnn));
end
if deltaax<0 && deltaay>0
    x2= xtpn+L*cos(abs(pn));
    y2= ytpn+L*sin(abs(pn));
end
if deltaax<0 && deltaay<0
    x2= xtpn-L*cos(abs(pi-mutpnn));
    y2= ytpn-L*sin(abs(pi-mutpnn));
end
deltaaxs=xrrr-x2;
deltaaays=yrrr-y2;

snoise=sqrt((deltaaxs)^2+(deltaaays)^2)

tetaa=pn-pnn
tetaax=tetaa;
if tetaa<0 || tetaa>=pi
    tetaa1=abs(tetaa);
    if tetaa1>=pi
        tetaa=2*pi-tetaa1
    else
        tetaa=tetaa1
    end
end
mutn=abs(tetaa);
waktun=((mutn/radi)*sampling)+totalwaktu;

sa=spercepatan;
tsam=tpercepatan;

scmn=snoise-(2*sa);
tcomn=(scmn/vcc)/60;
t1mn=tsam+tcomn;
totmn=(tsam*2)+tcomn;
tcmn=tsam*60+waktun;
tbmn=t1mn*60+waktun;
tamn=tbmn+tcmn-waktun;

if tbmn<=(tsam*60)
    scmn=0;
    tcomn=0;
    t1mn=((tmust/60)/2);
    totm=tmust/60;
```

```
        tcmn=t1m*60+waktun;
        tbmn =t1m*60+waktun;
        tamn=tbmn+tcmn-waktun;
    end
    totalwaktun=tamn
end
    tc1=tsam*60;

while t>=totalwaktu && t<totalwaktun
    tt(iii+1)=t;
    ibay=ibay+1;
    tbayang(ibay+1)=tbay;
    if t<waktun
        if (pn>=0 && pn<=(pi/2)) || (pn>(pi/2) && pn<pi)
            if (deltaax>0 && deltaay>0)
                if pnx==pnnx
                    emr=emrr;
                    eml=emll;
                end
                if pnnx>pnx && pnnx<=(pi+pnx)
                    emr=emrr;
                    eml=-emll;
                else
                    emr=-emrr
                    eml=emll
                end
            if pn>(pi/2) && pn<pi;
                emr=-emrr
                eml=emll
            end
        end
    end

    if (deltaax<0 && deltaay>=0)
        if pn>(pi/2) && pn<pi;
            if pn==pnnx
                emr=emrr;
                eml=emll;
            end
            if pnnx>pnx && pnnx<=(pi+pnx)
                emr=emrr;
                eml=-emll;
            else
                emr=-emrr
                eml=emll
            end
        else
            emr=emrr;
            eml=-emll;
        end
    end
end

    if (deltaax<=0 && deltaay<=0)
        if pnnx>pnx && pnnx<=(pi+pnx)
            emr=emrr;
```

```

        eml=-emll;
    else
        emr=-emrr;
        eml=emll;
    end
    if pn>(pi/2) && pn<pi;
        emr=emrr;
        eml=-emll;
    end
end

if (deltaax>0 && deltaay<0)
    if pn>(pi/2) && pn<pi;
        if pnnx>pnx && pnnx<(pi+pnx)
            emr=emrr;
            eml=-emll;
        else
            emr=-emrr
            eml=emll
        end
    else
        emr=-emrr
        eml=emll
    end
end
er(iiii+1)=emr;
el(iiii+1)=eml;
ir1=(emr/R)-(emr/R)*exp((-R/Lk)*thh);
il1=(eml/R)-(eml/R)*exp((-R/Lk)*thh);
ir=ir1;
il=il1;
Eemfr1=-emr*exp((-R/Lk)*thh);
Eemfl1=-eml*exp((-R/Lk)*thh);
Eemfr=Eemfr1;
Eemfl=Eemfl1;
end

if (pn<0 && pn>=(-pi/2)) || (pn<(-pi/2) && pn>(-pi))
|| (pn>(pi) && pn<(3*pi/2)) || (pn>=(3*pi/2) && pn<(2*pi))
    if (deltaax>0 && deltaay>0)
        if (pn<(-pi/2) && pn>(-pi)) || (pn>(pi) &&
pn<(3*pi/2))
            if pnnx<pnx && pnnx>(pnx-pi)
                emr=-emrr
                eml=emll
            else
                emr=emrr;
                eml=-emll;
            end
        else
            emr=emrr;
            eml=-emll;
        end
    end
end

if (deltaax<0 && deltaay>=0)

```

```

        if pnnx<pnx && pnnx>(pnx-pi)
            emr=-emrr;
            eml=emll;
        else
            emr=emrr;
            eml=-emll;
        end
        if (pn<(-pi/2) && pn>(-pi)) || (pn>(pi) &&
pn<(3*pi/2))
            emr=-emrr
            eml=emll
        end
    end

    if (deltaax<=0 && deltaay<=0)
        if (pn<(-pi/2) && pn>(-pi)) || (pn>(pi) &&
pn<(3*pi/2))
            if pnx==pnnx
                emr=emrr;
                eml=emll;
            end
            if pnnx<pnx && pnnx>(pnx-pi)
                emr=-emrr
                eml=emll
            else
                emr=emrr;
                eml=-emll;
            end
        else
            emr=-emrr
            eml=emll
        end
    end

    if (deltaax>0 && deltaay<0)
        if (pn<(-pi/2) && pn>(-pi)) || (pn>(pi) &&
pn<(3*pi/2))
            emr=emrr;
            eml=-emll;
        else
            if pnx==pnnx
                emr=emrr;
                eml=emll;
            end
            if pnnx<pnx && pnnx>(pnx-pi)
                emr=-emrr;
                eml=emll;
            else
                emr=emrr;
                eml=-emll;
            end
        end
    end
end
er(iiii+1)=emr;
el(iiii+1)=eml;
ir1=(emr/R) - (emr/R)*exp((-R/Lk)*thh);

```

```

        i11=(eml/R)-(eml/R)*exp((-R/Lk)*thh);
        ir=ir1;
        il=i11;
        ia(iii+1)=ir;
        ib(iii+1)=il;
        il=i11;
        Eemfr1=-emr*exp((-R/Lk)*thh);
        Eemfl1=-eml*exp((-R/Lk)*thh);
        Eemfr=Eemfr1;
        Eemfl=Eemfl1;
    end
end

if t>=waktun
    deltaaxs=xrrr-xnoise;
    deltaays=yrrr-ynoise;
    snoise=sqrt(((deltaaxs)^2)+((deltaays)^2));
    if snoise>(2*sa)
        sco=snoise-(2*sa);
        tco=(sco/vcc)/60;
        t1=tsam+tco;
        total=(2*tsam)+tco*60;
        tcc=(tsam*60)+waktun;
        tbb=(t1*60)+waktun;
        taa=tbb+tcc-waktun;
    end
    if snoise==(2*sa)
        sco=0;
        tco=0;
        t1=tsam/2;
        total=tsam
        tcc=t1*60+waktun;
        tbb=t1*60+waktun;
        taa=tbb+tcc-waktun;
    end
    if snoise<(2*sa)
        sco=0;
        tco=0;
        t1=(tmust/60)/2;
        total=tmust/60
        tcc=t1*60+waktun;
        tbb=t1*60+waktun;
        taa=tbb+tcc-waktun;
    end
end

if t<tcc
    emr=emrr;
    eml=emll;
    er(iii+1)=emr;
    el(iii+1)=eml;
    ir1=(emr/R)-(emr/R)*exp((-R/Lk)*th);
    i11=(eml/R)-(eml/R)*exp((-R/Lk)*th);
    ir=ir1;
    il=i11;

    Eemfr1=-emr*exp((-R/Lk)*th);

```



```

        Eemfl1=-eml*exp((-R/Lk)*th);
        Eemfr=Eemfr1;
        Eemfl=Eemfl1;
        th=th+sampling;
    end

if snoise>(2*sa)
    if t>=tcc && t<tbb
        emr=emrr;
        eml=emll;
        er(iii+1)=emr;
        el(iii+1)=eml;
        ir1=(emr/R)-(emr/R)*exp((-R/Lk)*th);
        il1=(eml/R)-(eml/R)*exp((-R/Lk)*th);
        ir2=- (emr/R)+(emr/R)*exp((-R/Lk)*t0);
        il2=- (eml/R)+(eml/R)*exp((-R/Lk)*t0);

        ir=ir1+ir2;
        il=il1+il2;

        Eemfr1=-emr*exp((-R/Lk)*th);
        Eemfl1=-eml*exp((-R/Lk)*th);
        Eemfr2=emr*exp((-R/Lk)*t0);
        Eemfl2=eml*exp((-R/Lk)*t0);

        Eemfr=Eemfr1+Eemfr2;
        Eemfl=Eemfl1+Eemfl2;
        th=th+sampling;
        t0=t0+sampling;
    end

    if t>=tbb && t<(tbb+tc1)
        emr=emrr;
        eml=emll;
        er(iii+1)=0;
        el(iii+1)=0;
        ir1=(emr/R)-(emr/R)*exp((-R/Lk)*th);
        il1=(eml/R)-(eml/R)*exp((-R/Lk)*th);
        ir2=- (emr/R)+(emr/R)*exp((-R/Lk)*t0);
        il2=- (eml/R)+(eml/R)*exp((-R/Lk)*t0);
        ir1off=- ((emr/R)-(emr/R)*exp((-R/Lk)*t00));
        il1off=- ((eml/R)-(eml/R)*exp((-R/Lk)*t00));
        ir=ir1+ir2+ir1off;
        il=il1+il2+il1off;

        Eemfr1=-emr*exp((-R/Lk)*th);
        Eemfl1=-eml*exp((-R/Lk)*th);
        Eemfr2=emr*exp((-R/Lk)*t0);
        Eemfl2=eml*exp((-R/Lk)*t0);
        Eemfr1off=emr*exp((-R/Lk)*t00);
        Eemfl1off=eml*exp((-R/Lk)*t00);

        Eemfr=Eemfr1+Eemfr2+Eemfr1off;
        Eemfl=Eemfl1+Eemfl2+Eemfl1off;
        th=th+sampling;
    end

```

```

        t0=t0+sampling;
        t00=t00+sampling;
    end

    if t>=(tbb+tc1)
        emr=emrr;
        eml=emll;
        er(iii+1)=0;
        el(iii+1)=0;
        ir1=(emr/R)-(emr/R)*exp((-R/Lk)*th);
        il1=(eml/R)-(eml/R)*exp((-R/Lk)*th);
        ir2=- (emr/R)+(emr/R)*exp((-R/Lk)*t0);
        il2=- (eml/R)+(eml/R)*exp((-R/Lk)*t0);
        ir1off=- ((emr/R)-(emr/R)*exp((-R/Lk)*t00));
        il1off=- ((eml/R)-(eml/R)*exp((-R/Lk)*t00));
        ir2off=(emr/R)-(emr/R)*exp((-R/Lk)*t000);
        il2off=(eml/R)-(eml/R)*exp((-R/Lk)*t000);

        ir=ir1+ir2+ir1off+ir2off;
        il=il1+il2+il1off+il2off;

        Eemfr1=-emr*exp((-R/Lk)*th);
        Eemfl1=-eml*exp((-R/Lk)*th);
        Eemfr2=emr*exp((-R/Lk)*t0);
        Eemfl2=eml*exp((-R/Lk)*t0);
        Eemfr1off=emr*exp((-R/Lk)*t00);
        Eemfl1off=eml*exp((-R/Lk)*t00);
        Eemfr2off=-emr*exp((-R/Lk)*t000);
        Eemfl2off=-eml*exp((-R/Lk)*t000);
        Eemfr=Eemfr1+Eemfr2+Eemfr1off+Eemfr2off;
        Eemfl=Eemfl1+Eemfl2+Eemfl1off+Eemfl2off;
        th=th+sampling;
        t0=t0+sampling;
        t00=t00+sampling;
        t000=t000+sampling;
    end
end

if snoise<(2*sa)
    if t>=tbb && t<(tbb+tc1)
        emr=emrr;
        eml=emll;
        er(iii+1)=0;
        el(iii+1)=0;
        ir1=(emr/R)-(emr/R)*exp((-R/Lk)*th);
        il1=(eml/R)-(eml/R)*exp((-R/Lk)*th);
        ir2=- (emr/R)+(emr/R)*exp((-R/Lk)*t0);
        il2=- (eml/R)+(eml/R)*exp((-R/Lk)*t0);
        ir1off=- ((emr/R)-(emr/R)*exp((-R/Lk)*t00));
        il1off=- ((eml/R)-(eml/R)*exp((-R/Lk)*t00));

        ir=ir1+ir2+ir1off;
        il=il1+il2+il1off;

        Eemfr1=-emr*exp((-R/Lk)*th);

```

```

Eemfl1=-eml*exp((-R/Lk)*th);
Eemfr2=emr*exp((-R/Lk)*t0);
Eemfl2=eml*exp((-R/Lk)*t0);
Eemfr1off=emr*exp((-R/Lk)*t00);
Eemfl1off=eml*exp((-R/Lk)*t00);
Eemfr=Eemfr1+Eemfr2+Eemfr1off;
Eemfl=Eemfl1+Eemfl2+Eemfl1off;
th=th+sampling;
t0=t0+sampling;
t00=t00+sampling;
end

if t>=(tbb+tc1)
    emr=emrr;
    eml=emll;
    er(iii+1)=0;
    el(iii+1)=0;
    ir1=(emr/R)-(emr/R)*exp((-R/Lk)*th);
    il1=(eml/R)-(eml/R)*exp((-R/Lk)*th);
    ir2=-emr/R+(emr/R)*exp((-R/Lk)*t0);
    il2=-eml/R+(eml/R)*exp((-R/Lk)*t0);
    ir1off=-emr/R+(emr/R)*exp((-R/Lk)*t00);
    il1off=-eml/R+(eml/R)*exp((-R/Lk)*t00);
    ir2off=emr/R-(emr/R)*exp((-R/Lk)*t000);
    il2off=eml/R-(eml/R)*exp((-R/Lk)*t000);
    ir=ir1+ir2+ir1off+ir2off;
    il=il1+il2+il1off+il2off;
    Eemfr1=-emr*exp((-R/Lk)*th);
    Eemfl1=-eml*exp((-R/Lk)*th);
    Eemfr2=emr*exp((-R/Lk)*t0);
    Eemfl2=eml*exp((-R/Lk)*t0);
    Eemfr1off=emr*exp((-R/Lk)*t00);
    Eemfl1off=eml*exp((-R/Lk)*t00);
    Eemfr2off=-emr*exp((-R/Lk)*t000);
    Eemfl2off=-eml*exp((-R/Lk)*t000);
    Eemfr=Eemfr1+Eemfr2+Eemfr1off+Eemfr2off;
    Eemfl=Eemfl1+Eemfl2+Eemfl1off+Eemfl2off;
    th=th+sampling;
    t0=t0+sampling;
    t00=t00+sampling;
    t000=t000+sampling;
end
end

kg=r2/r1;
wmr=Eemfr/Ks;
wml=Eemfl/Ks;
Tmr=ir*Ki;
Tml=il*Ki;
TLr=Tmr*kg;
TLl=Tml*kg;
wLr=(1/kg)*wmr;
wLl=(1/kg)*wml;
Jhub=Mhub*rin^2/2;
Jwheel=Mwheel*(rin^2+rout^2)/2;

```

```

Jrobot=Mrobot*rout^2/2;
J=Jrobot;
JL=Jhub+Jwheel+Jrobot;
Jr=JL+kg^2*Jmr;
Jl=JL+kg^2*Jml;
wLrldot=TLr/Jr;
wLldot=TLl/Jl;
fr=(Jr*wLrldot)/rout;
fl=(Jl*wLldot)/rout;
F=(fr+fl)/2;
vdott=((1/(2*m))*fr) + ((1/(2*m))*fl);
vdotrr(iii+1)=vdott;
wdot=((1/(2*b*J))*fr - (1/(2*b*J))*fl);
wdotrr(iii+1)=wdot;
wt=wt+wdot*t;
waw(iii+1)=wt;

    if t<waktun
        vt=v0+vdott*thh;
        sss=s4+vt*thh+0.5*vdott*(thh^2);
        sbay=sbay0+vt*thh+0.5*vdott*(thh^2);
        sbay0=sbay;
        vvtw1=vt;
        s0=sss;

        if (pn>=0 && pn<=(pi/2)) || (pn>(pi/2) && pn<pi)
            if (deltaax>0 && deltaay>0)
                ten=ten+radi;
                if pnx==pnnx
                    teth=pn; 'lurus'
                    sdt=pn;
                end
                if pnnx>pnx && pnnx<=(pi+pnx)
                    teth=pn+ten; 'kiri'
                    sdt=teth;
                else
                    teth=pn-ten; 'kanan'
                    sdt=teth;
                end
            end
            if pn>(pi/2) && pn<pi;
                teth=pn-ten; 'kanan'
                sdt=teth;
            end
        end

        if (deltaax<0 && deltaay>=0)
            ten=ten+radi;
            teth=pn+ten; 'kiri'
            sdt=pi+teth;
            if pn>(pi/2) && pn<pi;
                if pnx==pnnx
                    teth=pn; 'lurus'
                    sdt=pn;
                end
            end
        end
    end

```

```

        if pnnx>pnx && pnnx<=(pi+pnx)
            teth=pn+ten; 'kiri'
            sdt=pi+teth;
        else
            teth=pn-ten; 'kanan'
            sdt=pi+teth;
        end
    end
end
if (deltaax<=0 && deltaay<=0)
    ten=ten+radi;
    if pnnx>pnx && pnnx<=(pi+pnx)
        teth=pn+ten; 'kiri'
        sdt=teth-pi;
    else
        teth=pn-ten; 'kanan'
        sdt=teth-pi;
    end
end

    if pn>(pi/2) && pn<pi;
        teth=pn+ten; 'kiri'
        sdt=teth-pi;
    end
end

    if (deltaax>0 && deltaay<0)
        ten=ten+radi;
        teth=pn-ten; 'kanan'
        sdt=teth;

        if pn>(pi/2) && pn<pi;
            if pnnx>pnx && pnnx<(pi+pnx)
                teth=pn+ten; 'kiri'
                sdt=teth;
            else
                teth=pn-ten; 'kanan'
                sdt=teth;
            end
        end
    end
end

    if (pn<0 && pn>=(-pi/2)) || (pn<(-pi/2) && pn>(-pi))
|| (pn>(pi) && pn<(3*pi/2)) || (pn>=(3*pi/2) && pn<(2*pi))
        if (deltaax>0 && deltaay>0)
            ten=ten+radi; 'kiri'
            teth=pn+ten;
            sdt=teth;

            if (pn<(-pi/2) && pn>(-pi)) || (pn>(pi) &&
pn<(3*pi/2))
                if pnnx<pnx && pnnx>(pnx-pi)
                    teth=pn-ten; 'kanan'
                    sdt=teth;
                else

```

```

        teth=pn+ten; 'kiri'
        sdt=teth;
    end
end
end

if (deltaax<0 && deltaay>=0)
    ten=ten+radi;
    if pnnx<pnx && pnnx>(pnx-pi)
        teth=pn-ten; 'kanan'
        sdt=pi+teth;
    else
        teth=pn+ten; 'kiri'
        sdt=pi+teth;
    end

    if (pn<(-pi/2) && pn>(-pi)) || (pn>(pi) &&
pn<(3*pi/2))
        teth=pn-ten; 'kanan'
        sdt=pi+teth;
    end
end

if (deltaax<=0 && deltaay<=0)
    ten=ten+radi;
    teth=pn-ten; 'kanan'
    sdt=pi+teth;

    if (pn<(-pi/2) && pn>(-pi)) || (pn>(pi) &&
pn<(3*pi/2))
        if pnx==pnnx
            teth=pn; 'lurus'
            sdt=pn;
        end
        if pnnx<pnx && pnnx>(pnx-pi)
            teth=pn-ten; 'kanan'
            sdt=pi+teth;
        else
            teth=pn+ten; 'kiri'
            sdt=pi+teth;
        end
    end
end

if (deltaax>0 && deltaay<0)
    ten=ten+radi;
    if pnx==pnnx
        teth=pn; 'lurus'
        sdt=pn;
    end
    if pnnx<pnx && pnnx>(pnx-pi)
        teth=pn-ten; 'kanan'
        sdt=teth;
    else
        teth=pn+ten; 'kiri'
    end
end

```

```

                                sdt=teth;
                                end
                                if (pn<(-pi/2) && pn>(-pi)) || (pn>(pi) &&
pn<(3*pi/2))
                                teth=pn+ten; 'kiri'
                                sdt=teth;
                                end
                                end
                                end
                                end

                                mutteth=abs(teth);
                                if deltaax>=0 && deltaay>=0
                                    xnoise=xtpn+L*cos(teth);
                                    ynoise=ytpn+L*sin(teth);
                                end
                                if deltaax>0 && deltaay<0
                                    xnoise= xtpn+L*cos(abs((2*pi)-mutteth));
                                    ynoise= ytpn-L*sin(abs((2*pi)-mutteth));
                                end
                                if deltaax<0 && deltaay>0
                                    xnoise= xtpn+L*cos(abs(teth));
                                    ynoise= ytpn+L*sin(abs(teth));
                                end
                                if deltaax<0 && deltaay<0
                                    xnoise= xtpn-L*cos(abs(pi-mutteth));
                                    ynoise= ytpn-L*sin(abs(pi-mutteth));
                                end
                                xnoises(iii+1)=xnoise;
                                ynoises(iii+1)=ynoise;

                                if t>=waktun
                                    if t<tcc
                                        vt=v0+vdott*th;
                                        sss=s0+vt*th+0.5*vdott*(th^2);
                                        sbay=sbay0+vt*th+0.5*vdott*(th^2);
                                        sbay1=sbay;
                                        vt1=vt;
                                        s5=sss;
                                    end
                                    if snoise>(2*sa)
                                        if t>=tcc && t<tbb
                                            vt=vt1+vdott*tv;
                                            sss=s5+vt*tv+0.5*vdott*(tv^2);
                                            sbay=sbay1+vt*tv+0.5*vdott*(tv^2);
                                            sbay2=sbay;
                                            vt2=vt;
                                            s6=sss;
                                            tv=tv+sampling;
                                        end
                                        if t>=tbb && t<(tbb+tc1)
                                            vt=vt2+vdott*tvv;
                                            sss=s6+vt*tvv-0.5*vdott*(tvv^2);
                                            sbay=sbay2+vt*tvv-0.5*vdott*(tvv^2);
                                            sbay3=sbay;
                                        end
                                    end
                                end

```

```

        vt3=vt;
        s7=sss;
        tvv=tvv+sampling;
    end
    if t>=(tbb+tc1)
        vt0=v0;
        vt=vt0+vdott*tvvv;
        sss=s7+vt*tvvv-0.5*vdott*(tvvv^2);
        sbay=sbay3+vt*tvvv-0.5*vdott*(tvvv^2);
        sbay4=sbay;
        tvvv=tvvv+sampling;
    end
end
end
if snoise<(2*sa)
    if t>=tbb && t<(tbb+tc1)
        vt=vt1+vdott*tvvii;
        sss=s5+vt*tvvii-0.5*vdott*(tvvii^2);
        sbay=sbay1+vt*tvvii-0.5*vdott*(tvvii^2);
        sbay5=sbay;
        vt5=vt;
        s8=sss;
        tvvii=tvvii+sampling;
    end
    if t>=(tbb+tc1)
        vt0=v0;
        vt=vt0+vdott*tvvvii;
        sss=s8+vt*tvvvii-0.5*vdott*(tvvvii^2);
        sbay=sbay5+vt*tvvvii-0.5*vdott*(tvvvii^2);
        sbay6=sbay;
        tvvvii=tvvvii+sampling;
    end
end
end
v(iii+1)=vt;
ss(iii+1)=sss;
iii=1+iii;
t=t+sampling;
thh=thh+sampling;
ssbay(bay+1)=sbay;
    tetha(ibay+1)=teth;
    tettha(ibay+1)=ten;
if (deltaax>=0 && deltaay>0) || (deltaax>0 && deltaay<=0)
    xakhir=xtpn+L*cos(teth)+sbay*cos(sdt);
    yakhir=ytpn+L*sin(teth)+sbay*sin(sdt);
end
if (deltaax<0 && deltaay>0) || (deltaax<=0 && deltaay<=0)
    xakhir=xtpn+L*cos(teth)-sbay*cos(sdt);
    yakhir=ytpn+L*sin(teth)-sbay*sin(sdt);
end

x1(iiii+1)=xakhir;
y1(iiii+1)=yakhir;
tbay=tbay+sampling;
iiii=iiii+1;
end
xterakhir=xakhir

```



```
yterakhir=yakhir
waktu=tamn

%% Output
figure(1)
subplot(4,1,1);
plot(tt,er);
grid;
xlabel('Waktu (s)'), ylabel('Tegangan (V)');
title('Hubungan tegangan motor kanan terhadap waktu')

subplot(4,1,2);
plot(tt,el);
grid;
xlabel('Waktu (s)'), ylabel('Tegangan (V)');
title('Hubungan tegangan motor kiri terhadap waktu')

subplot(4,1,3);
plot(tt,v);
grid;
xlabel('Waktu (s)'), ylabel('Kecepatan(m/s)');
title('Hubungan kecepatan robot terhadap waktu')

subplot(4,1,4);
plot(tt,ss);
grid;
xlabel('Waktu (s)'), ylabel('jarak');
title('Hubungan jarak terhadap waktu')

figure(2);
subplot(2,2,1);
plot(tt,vdotrr);
grid;
xlabel('Waktu (s)'), ylabel('Percepatan');
title('Hubungan percepatan robot terhadap waktu')

subplot(2,2,2);
plot(tt,v);
grid;
xlabel('Waktu (s)'), ylabel('Kecepatan (m/s)');
title('Hubungan kecepatan robot terhadap waktu')

subplot(2,2,3);
plot(tt,ss);
grid;
xlabel('Waktu (s)'), ylabel('jarak');
title('Hubungan jarak terhadap waktu')

subplot(2,2,4);
plot(x1,y1);
grid;
xlabel('X'), ylabel('Y');
title('Koordinat posisi robot')

end
```

## **LAMPIRAN D**

---

**PROGRAM SIMULASI PERGERAKAN ROBOT  
MOBIL BERODA MELALUI TRAYEKTORI TANPA  
GANGGUAN**

```
em = input('Masukkan tegangan motor (tegangan motor kanan=tegangan
motor kiri)= ');
emrr=em;
er(1)=0;
emll=em;
el(1)=0;

pangkat      = input ('Masukkan pangkat untuk persamaan trayektori
: ');
xref         = input ('Masukkan harga x pertama robot mobil
bergerak : ');
xpterakhir  = input ('Masukkan harga x terakhir robot berhenti
: ');
yref=xref;

x1(1)=xref;
xaw=xref;
yaw=yref;
y1(1)=yaw;
sampling=0.01;
samplingx=0.01;
phi = (pi/2);

xtray=xref:samplingx:xpterakhir;
ytray=xtray.^pangkat;

RPM=263;
Vmin=3;
Vmaks=8;

if em<=Vmin
    ' Tegangan diberikan pada motor terlalu kecil '
    Vmin
    trayek1;
elseif em>Vmaks
    ' Tegangan diberikan pada motor terlalu besar '
    Vmaks
    trayek1;
elseif em>Vmin && em<=Vmaks

selisih=Vmaks-Vmin;
kec=RPM/selisih;
kenaikan=emrr-Vmin;

Ks=kec*kenaikan;
r2=25e-3;
r1=1e-3;
Ki=3e-1;
R=10000;
Lk=0.63;
Mhub=0.025;
Mwheel=0.05;
Mrobot=3;
m=Mrobot;
rin=0.03;
```

```
rout=0.05;
Jmr=0.56e7;
Jml=0.56e7;
b=0.05;
L=0.1;

format long
xtp=xaw-(L*cos(phi));
ytp=yaw-(L*sin(phi));

    mutphi=abs(phi)
    phix=phi
    if phix<0
        phix=(2*pi)-mutphi
    end

vcmot=Ks*(2*pi*r1)/60;
wcmot=vcmot/r2;
vc=wcmot*rout;
iiii=1;
ulang=0

tbay=0;
ibay=1;
tbayang(1)=0;

while xref<xpterakhir
    xr=xref
    xrr=xr+samplingx
    yrr=xrr^pangkat

    deltax=xrr-xtp;
    deltay=yrr-ytp;

    p= atan2(deltay,deltax)
    mutp=abs(p);
    px=p
    if p<0
        px=(2*pi)+p
    end

    if deltax>=0 && deltay>=0
        xharus=xtp+L*cos(p);
        yharus=ytp+L*sin(p);
    end
    if deltax>0 && deltay<0
        if p>(3*pi/2)
            xharus= xtp+L*cos((2*pi)-p);
            yharus= ytp-L*sin((2*pi)-p);
        end
        if p<0
            xharus= xtp+L*cos(mutp);
            yharus= ytp-L*sin(mutp);
        end
    end
end
```

```

    if deltax<0 && deltax>0
        xharus= xtp-L*cos(phi-(pi/2));
        yharus= ytp+L*sin(phi-(pi/2));
    end
    if deltax<0 && deltax<0
        xharus= xtp-L*cos(abs(pi-mutp));
        yharus= ytp-L*sin(abs(pi-mutp));
    end
    deltaax=xrr-xharus;
    deltaay=yrr-yharus;
    s=sqrt((deltaax)^2+(deltaay)^2)
    smust=s;

while vt<=vc
    ttti(iii+1)=ti;
    emr=emrr;
    eml=eml1;

kg=r2/r1;
ir1=(emr/R)-(emr/R)*exp((-R/Lk)*ti);
il1=(eml/R)-(eml/R)*exp((-R/Lk)*ti);
ir=ir1;
il=il1;
ia(iii+1)=ir;
ib(iii+1)=il;
il=il1;
Eemfr1=-emr*exp((-R/Lk)*ti);
Eemfl1=-eml*exp((-R/Lk)*ti);
Eemfr=Eemfr1;
Eemfl=Eemfl1;

wmr=Eemfr/Ks;
wml=Eemfl/Ks;
Tmr=ir*Ki;
Tml=il*Ki;
TLr=Tmr*kg;
TLl=Tml*kg;
wLr=(1/kg)*wmr;
wLl=(1/kg)*wml;
Jhub=Mhub*rin^2/2;
Jwheel=Mwheel*(rin^2+rout^2)/2;
Jrobot=Mrobot*rout^2/2;
J=Jrobot;
JL=Jhub+Jwheel+Jrobot;
Jr=JL+kg^2*Jmr;
Jl=JL+kg^2*Jml;
wLrdot=TLr/Jr;
wLldot=TLl/Jl;
fr=(Jr*wLrdot)/rout;
fl=(Jl*wLldot)/rout;
F=(fr+fl)/2;
vdoot=((1/(2*m))*fr) + ((1/(2*m))*fl);

vt=v0+vdoot*ti;
s=s0+vt*ti+0.5*vdoot*(ti^2);

```

```
vaw(iii+1)=vt;
ssaw(iii+1)=s;
iii=1+iii;
ti=ti+sampling;
if s<smust
    tmust=ti;
end
end
tper=ti;
tpercepatan=ti/60
spercepatan=s
vcc=vt

teta=phi-p
tetax=teta;
if teta<0 || teta >=pi
    tetal=abs(teta);
    if tetal>=pi
        teta=2*pi-tetal
    else
        teta=tetal
    end
end

kel=2*pi*(2*b);
fi=(vcc/kel)*(2*pi);
radl=fi*sampling;
radi=radl*2;
n=teta/radi;
mut=abs(teta);
waktu=(mut/radi)*sampling;

sa=spercepatan;
tsam=tpercepatan;
scm=smust-(2*sa);
tcom=(scm/vcc)/60;
t1m=tsam+tcom;
totm=(tsam*2)+tcom;
tcm=tsam*60+waktu;
tbm=t1m*60+waktu;
tam=tbm+tcm-waktu;
if tbm<=(tsam*60)
    scm=0;
    tcom=0;
    t1m=(tmust/60)/2;
    totm=tmust/60;
    tcm=t1m*60+waktu;
    tbm =t1m*60+waktu;
    tam=tbm+tcm-waktu;
end
totalwaktu=tam+waktu;
```

```
%% MULAI PROGRAM
while t<=totalwaktu
    ttt(iii+1)=t;
    tbay=tbay+sampling;
    tbayang(ibay+1)=tbay;
    if t<waktu
        if (phi>=0 && phi<=(pi/2)) || (phi>(pi/2) && phi<pi)
            if (deltaax>0 && deltaay>0)
                if phix==px
                    emr=emrr;
                    eml=emll;
                end

                if px>phix && px<=(pi+phix)
                    emr=emrr;
                    eml=-emll;
                else
                    emr=-emrr
                    eml=emll
                end

                if phi>(pi/2) && phi<pi;
                    emr=-emrr
                    eml=emll
                end
            end
        end

        if (deltaax<0 && deltaay>=0)
            if phi>(pi/2) && phi<pi;
                if phix==px
                    emr=emrr;
                    eml=emll;
                end
                if px>phix && px<=(pi+phix)
                    emr=emrr;
                    eml=-emll;
                else
                    emr=-emrr
                    eml=emll
                end
            end
        else
            emr=emrr;
            eml=-emll;
        end
    end

    if (deltaax<=0 && deltaay<=0)
        if px>phix && px<=(pi+phix)
            emr=emrr;
            eml=-emll;
        else
            emr=-emrr;
            eml=emll;
        end
    end
    if phi>(pi/2) && phi<pi;
```

```

        emr=emrr;
        eml=-emll;
    end
end

if (deltaax>0 && deltaay<0)
    if phi>(pi/2) && phi<pi;
        if px>phix && px<(pi+phix)
            emr=emrr;
            eml=-emll;
        else
            emr=-emrr
            eml=emll
        end
    else
        emr=-emrr
        eml=emll
    end
end

er(iiii+1)=emr;
el(iiii+1)=eml;
ir1=(emr/R)-(emr/R)*exp((-R/Lk)*t);
il1=(eml/R)-(eml/R)*exp((-R/Lk)*t);
ir=ir1;
il=il1;
Eemfr1=-emr*exp((-R/Lk)*t);
Eemfl1=-eml*exp((-R/Lk)*t);
Eemfr=Eemfr1;
Eemfl=Eemfl1;
end

    if (phi<0 && phi>=(-pi/2)) || (phi<(-pi/2) && phi>(-
pi)) || (phi>(pi) && phi<(3*pi/2)) || (phi>=(3*pi/2) &&
phi<(2*pi))
        if (deltaax>0 && deltaay>0)
            if (phi<(-pi/2) && phi>(-pi)) || (phi>(pi) &&
phi<(3*pi/2))
                if px<phix && px>(phix-pi)
                    emr=-emrr
                    eml=emll
                else
                    emr=emrr;
                    eml=-emll;
                end
            else
                emr=emrr;
                eml=-emll;
            end
        end
    end

if (deltaax<0 && deltaay>=0)
    if px<phix && px>(phix-pi)
        emr=-emrr;
        eml=emll;
    else
        emr=emrr;
    end
end

```



```

        eml=-emll;
    end
    if (phi<(-pi/2) && phi>(-pi)) || (phi>(pi) &&
phi<(3*pi/2))
        emr=-emrr
        eml=epll
    end
end

if (deltaax<=0 && deltaay<=0)
    if (phi<(-pi/2) && phi>(-pi)) || (phi>(pi) &&
phi<(3*pi/2))
        if phix==px
            emr=emrr;
            eml=epll;
        end
        if px<phix && px>(phix-pi)
            emr=-emrr
            eml=epll
        else
            emr=emrr;
            eml=-emll;
        end
    else
        emr=-emrr
        eml=epll
    end
end

if (deltaax>0 && deltaay<0)
    if (phi<(-pi/2) && phi>(-pi)) || (phi>(pi) &&
phi<(3*pi/2))
        emr=emrr;
        eml=-emll;
    else
        if phix==px
            emr=emrr;
            eml=epll;
        end
        if px<phix && px>(phix-pi)
            emr=-emrr;
            eml=epll;
        else
            emr=emrr;
            eml=-emll;
        end
    end
end
er(iiii+1)=emr;
el(iiii+1)=eml;
ir1=(emr/R) - (emr/R)*exp((-R/Lk)*t);
il1=(eml/R) - (eml/R)*exp((-R/Lk)*t);
ir=ir1;
il=il1;
il=il1;
Eemfr1=-emr*exp((-R/Lk)*t);

```

```

        Eemf11=-eml*exp((-R/Lk)*t);
        Eemfr=Eemfr1;
        Eemfl=Eemf11;
    end
end
if p==phi
    waktu=0;
    teth=phi;
    sdt=teth;
end

if t>=waktu
    deltaxs=xrr-xharus;
    deltays=yrr-yharus;
    s=sqrt(((deltaxs)^2)+((deltays)^2));
    if s>(2*sa)
        sc=s-(2*sa);
        tco=(sc/vcc)/60;
        t1=tsam+tco;
        tot=(tsam*2)+tco;
        tc=tsam*60+waktu;
        tb=t1*60+waktu;
        ta=tb+tc-waktu;
    end
    if s==(2*sa)
        sc=0;
        tco=0;
        t1=tsam/2;
        tot=tsam;
        tc=t1*60+waktu;
        tb=tsam/2+waktu;
        ta=tb+tc-waktu;
    end
    if s<(2*sa)
        sc=0;
        tco=0;
        t1=(tmust/60)/2;
        tot=tmust/60;
        tc=t1*60+waktu;
        tb =t1*60+waktu;
        ta=tb+tc-waktu;
    end

    if t<tc
        emr=emrr;
        eml=emll;
        er(iiii+1)=emr;
        el(iiii+1)=eml;
        ir1=(emr/R)-(emr/R)*exp((-R/Lk)*ts);
        il1=(eml/R)-(eml/R)*exp((-R/Lk)*ts);
        ir=ir1;
        il=il1;
        ia(iii+1)=ir;
        ib(iii+1)=il;
        il=il1;
        Eemfr1=-emr*exp((-R/Lk)*ts);

```

```

Eemfl1=-eml*exp((-R/Lk)*ts);
Eemfr=Eemfr1;
Eemfl=Eemfl1;
ts=ts+sampling;
end

if s>(2*sa)
  if t>=tc && t<tb
    emr=emrr;
    eml=epll;
    er(iiii+1)=emr;
    el(iiii+1)=eml;
    ir1=(emr/R)-(emr/R)*exp((-R/Lk)*ts);
    il1=(eml/R)-(eml/R)*exp((-R/Lk)*ts);
    ir2=- (emr/R)+(emr/R)*exp((-R/Lk)*t0);
    il2=- (eml/R)+(eml/R)*exp((-R/Lk)*t0);

    ir=ir1+ir2;
    il=il1+il2;
    ia(iii+1)=ir;
    ib(iii+1)=il;

    Eemfr1=-emr*exp((-R/Lk)*ts);
    Eemfl1=-eml*exp((-R/Lk)*ts);
    Eemfr2=emr*exp((-R/Lk)*t0);
    Eemfl2=eml*exp((-R/Lk)*t0);

    Eemfr=Eemfr1+Eemfr2;
    Eemfl=Eemfl1+Eemfl2;
    ts=ts+sampling;
    t0=t0+sampling;
  end

  if t>=tb && t<(tb+tc-waktu)
    emr=emrr;
    eml=epll;
    er(iiii+1)=0;
    el(iiii+1)=0;
    ir1=(emr/R)-(emr/R)*exp((-R/Lk)*ts);
    il1=(eml/R)-(eml/R)*exp((-R/Lk)*ts);
    ir2=- (emr/R)+(emr/R)*exp((-R/Lk)*t0);
    il2=- (eml/R)+(eml/R)*exp((-R/Lk)*t0);
    irloff=- ((emr/R)-(emr/R)*exp((-R/Lk)*t00));
    illoff=- ((eml/R)-(eml/R)*exp((-R/Lk)*t00));
    ir=ir1+ir2+irloff;
    il=il1+il2+illoff;
    ia(iii+1)=ir;
    ib(iii+1)=il;

    Eemfr1=-emr*exp((-R/Lk)*ts);
    Eemfl1=-eml*exp((-R/Lk)*ts);
    Eemfr2=emr*exp((-R/Lk)*t0);
    Eemfl2=eml*exp((-R/Lk)*t0);
    Eemfrloff=emr*exp((-R/Lk)*t00);
    Eemflloff=eml*exp((-R/Lk)*t00);
  end
end

```

```

        Eemfr=Eemfr1+Eemfr2+Eemfrloff;
        Eemfl=Eemfl1+Eemfl2+Eemflloff;
        t0=t0+sampling;
        t00=t00+sampling;
        ts=ts+sampling;
    end

    if t>=(tb+tc-waktu)
        emr=emrr;
        eml=emll;
        er(iiii+1)=0;
        el(iiii+1)=0;
        ir1=(emr/R)-(emr/R)*exp((-R/Lk)*ts);
        il1=(eml/R)-(eml/R)*exp((-R/Lk)*ts);
        ir2=- (emr/R)+(emr/R)*exp((-R/Lk)*t0);
        il2=- (eml/R)+(eml/R)*exp((-R/Lk)*t0);
        irloff=- ((emr/R)-(emr/R)*exp((-R/Lk)*t00));
        illoff=- ((eml/R)-(eml/R)*exp((-R/Lk)*t00));
        ir2off=(emr/R)-(emr/R)*exp((-R/Lk)*t000);
        il2off=(eml/R)-(eml/R)*exp((-R/Lk)*t000);

        ir=ir1+ir2+irloff+ir2off;
        il=il1+il2+illoff+il2off;

        ia(iii+1)=ir;
        ib(iii+1)=il;

        Eemfr1=-emr*exp((-R/Lk)*ts);
        Eemfl1=-eml*exp((-R/Lk)*ts);
        Eemfr2=emr*exp((-R/Lk)*t0);
        Eemfl2=eml*exp((-R/Lk)*t0);
        Eemfrloff=emr*exp((-R/Lk)*t00);
        Eemflloff=eml*exp((-R/Lk)*t00);
        Eemfr2off=-emr*exp((-R/Lk)*t000);
        Eemfl2off=-eml*exp((-R/Lk)*t000);
        Eemfr=Eemfr1+Eemfr2+Eemfrloff+Eemfr2off;
        Eemfl=Eemfl1+Eemfl2+Eemflloff+Eemfl2off;
        ts=ts+sampling;
        t0=t0+sampling;
        t00=t00+sampling;
        t000=t000+sampling;
    end
end

if s<(2*sa)
    if t>=tb && t<(tb+tc-waktu)
        emr=emrr;
        eml=emll;
        er(iiii+1)=0;
        el(iiii+1)=0;
        ir1=(emr/R)-(emr/R)*exp((-R/Lk)*ts);
        il1=(eml/R)-(eml/R)*exp((-R/Lk)*ts);
        ir2=- (emr/R)+(emr/R)*exp((-R/Lk)*t0);
        il2=- (eml/R)+(eml/R)*exp((-R/Lk)*t0);

```

```

        ir1off=-((emr/R)-(emr/R)*exp((-R/Lk)*t00));
        il1off=-((eml/R)-(eml/R)*exp((-R/Lk)*t00));

        ir=ir1+ir2+ir1off;
        il=il1+il2+il1off;

        ia(iii+1)=ir;
        ib(iii+1)=il;

        Eemfr1=-emr*exp((-R/Lk)*ts);
        Eemfl1=-eml*exp((-R/Lk)*ts);
        Eemfr2=emr*exp((-R/Lk)*t0);
        Eemfl2=eml*exp((-R/Lk)*t0);
        Eemfr1off=emr*exp((-R/Lk)*t00);
        Eemfl1off=eml*exp((-R/Lk)*t00);
        Eemfr=Eemfr1+Eemfr2+Eemfr1off;
        Eemfl=Eemfl1+Eemfl2+Eemfl1off;
        ts=ts+sampling;
        t0=t0+sampling;
        t00=t00+sampling;
    end

    if t>=(tb+tc-waktu)
        emr=0;
        eml=0;
        er(iiii+1)=0;
        el(iiii+1)=0;
        ir1=(emr/R)-(emr/R)*exp((-R/Lk)*ts);
        il1=(eml/R)-(eml/R)*exp((-R/Lk)*ts);
        ir2=-emr/R+emr/R*exp((-R/Lk)*t0);
        il2=-eml/R+eml/R*exp((-R/Lk)*t0);
        ir1off=-((emr/R)-(emr/R)*exp((-R/Lk)*t00));
        il1off=-((eml/R)-(eml/R)*exp((-R/Lk)*t00));
        ir2off=(emr/R)-(emr/R)*exp((-R/Lk)*t000);
        il2off=(eml/R)-(eml/R)*exp((-R/Lk)*t000);
        ir=ir1+ir2+ir1off+ir2off;
        il=il1+il2+il1off+il2off;

        Eemfr1=-emr*exp((-R/Lk)*ts);
        Eemfl1=-eml*exp((-R/Lk)*ts);
        Eemfr2=emr*exp((-R/Lk)*t0);
        Eemfl2=eml*exp((-R/Lk)*t0);
        Eemfr1off=emr*exp((-R/Lk)*t00);
        Eemfl1off=eml*exp((-R/Lk)*t00);
        Eemfr2off=-emr*exp((-R/Lk)*t000);
        Eemfl2off=-eml*exp((-R/Lk)*t000);
        Eemfr=Eemfr1+Eemfr2+Eemfr1off+Eemfr2off;
        Eemfl=Eemfl1+Eemfl2+Eemfl1off+Eemfl2off;
        ts=ts+sampling;
        t0=t0+sampling;
        t00=t00+sampling;
        t000=t000+sampling;
    end
end
end
end

```

```
kg=r2/r1;
wmr=Eemfr/Ks;
wml=Eemfl/Ks;
Tmr=ir*Ki;
Tml=il*Ki;
TLr=Tmr*kg;
TLl=Tml*kg;
wLr=(1/kg)*wmr;
wLl=(1/kg)*wml;
Jhub=Mhub*rin^2/2;
Jwheel=Mwheel*(rin^2+rout^2)/2;
Jrobot=Mrobot*rout^2/2;
J=Jrobot;
JL=Jhub+Jwheel+Jrobot;
Jr=JL+kg^2*Jmr;
Jl=JL+kg^2*Jml;
wLrdot=TLr/Jr;
wLldot=TLl/Jl;
fr=(Jr*wLrdot)/rout;
fl=(Jl*wLldot)/rout;
F=(fr+fl)/2;
vdott=((1/(2*m))*fr) + ((1/(2*m))*fl);
vdotrr(iii+1)=vdott;
wdoot=((1/(2*b*J))*fr - (1/(2*b*J))*fl);
vdotrr(iii+1)=wdoot;
wt=wt+wdoot*t;

    if t<waktu
        vvt=v0+vdott*t;
        sss=s0+vvt*t+0.5*vdott*(t^2);
        vvtw1=vvt;
        s0=sss;

        if (phi>=0 && phi<=(pi/2)) || (phi>(pi/2) && phi<pi)
            if (deltaax>0 && deltaay>0)
                te=te+radi;
                if phix==px
                    teth=phi; 'lurus'
                    sdt=phi;
                end

                if px>phix && px<=(pi+phix)
                    teth=phi+te; 'kiri'
                    sdt=teth;
                else
                    teth=phi-te; 'kanan'
                    sdt=teth;
                end

                if phi>(pi/2) && phi<pi;
                    teth=phi-te; 'kanan'
                    sdt=teth;
                end
            end
        end
    end
```

```
if (deltaax<0 && deltaay>=0)
    te=te+radi;
    teth=phi+te; 'kiri'
    sdt=pi+teth;

    if phi>(pi/2) && phi<pi;
        if phix==px
            teth=phi; 'lurus'
            sdt=phi;
        end
        if px>phix && px<=(pi+phix)
            teth=phi+te; 'kiri'
            sdt=pi+teth;
        else
            teth=phi-te; 'kanan'
            sdt=pi+teth;
        end
    end
end

if (deltaax<=0 && deltaay<=0)
    te=te+radi;
    if px>phix && px<=(pi+phix)
        teth=phi+te; 'kiri'
        sdt=teth-pi;
    else
        teth=phi-te; 'kanan'
        sdt=teth-pi;
    end

    if phi>(pi/2) && phi<pi;
        teth=phi+te; 'kiri'
        sdt=teth-pi;
    end
end
if (deltaax>0 && deltaay<0)
    te=te+radi;
    teth=phi-te; 'kanan'
    sdt=teth;

    if phi>(pi/2) && phi<pi;
        if px>phix && px<(pi+phix)
            teth=phi+te; 'kiri'
            sdt=teth;
        else
            teth=phi-te; 'kanan'
            sdt=teth;
        end
    end
end
end
end
```

```

        if (phi<0 && phi>=(-pi/2)) || (phi<(-pi/2) && phi>(-
pi)) || (phi>(pi) && phi<(3*pi/2)) || (phi>=(3*pi/2) &&
phi<(2*pi))
            if (deltaax>0 && deltaay>0)
                te=te+radi; 'kiri'
                teth=phi+te;
                sdt=teth;

                if (phi<(-pi/2) && phi>(-pi)) || (phi>(pi) &&
phi<(3*pi/2))
                    if px<phix && px>(phix-pi)
                        teth=phi-te; 'kanan'
                        sdt=teth;
                    else
                        teth=phi+te; 'kiri'
                        sdt=teth;
                    end
                end
            end
        end

        if (deltaax<0 && deltaay>=0)
            te=te+radi;
            if px<phix && px>(phix-pi)
                teth=phi-te; 'kanan'
                sdt=pi+teth;
            else
                teth=phi+te; 'kiri'
                sdt=pi+teth;
            end

            if (phi<(-pi/2) && phi>(-pi)) || (phi>(pi) &&
phi<(3*pi/2))
                teth=phi-te; 'kanan'
                sdt=pi+teth;
            end
        end

        if (deltaax<=0 && deltaay<=0)
            te=te+radi;
            teth=phi-te; 'kanan'
            sdt=pi+teth;
            if (phi<(-pi/2) && phi>(-pi)) || (phi>(pi) &&
phi<(3*pi/2))
                if phix==px
                    teth=phi; 'lurus'
                    sdt=phi;
                end
                if px<phix && px>(phix-pi)
                    teth=phi-te; 'kanan'
                    sdt=pi+teth;
                else
                    teth=phi+te; 'kiri'
                    sdt=pi+teth;
                end
            end
        end
    end

```



```

end

if (deltaax>0 && deltaay<0)
    te=te+radi;
    if phix==px
        teth=phi; 'lurus'
        sdt=phi;
    end
    if px<phix && px>(phix-pi)
        teth=phi-te; 'kanan'
        sdt=teth;
    else
        teth=phi+te; 'kiri'
        sdt=teth;
    end

    if (phi<(-pi/2) && phi>(-pi)) || (phi>(pi) &&
phi<(3*pi/2))
        teth=phi+te; 'kiri'
        sdt=teth;
    end
end
end
end
if teth<0
    teth=(2*pi)+teth;
end
xharus=xtp+L*cos(teth);
yharus=ytp+L*sin(teth);

if t>=waktu
    if t<tc
        vvt=v0+vdott*tss;
        sss=s0+vvt*tss+0.5*vdott*(tss^2);
        vvt1=vvt;
        ss1=sss;
        tss=tss+sampling;
    end
    if s>(2*sa)
        if t>=tc && t<tb
            vvt=vvt1+vdott*tv;
            sss=ss1+vvt*tv+0.5*vdott*(tv^2);
            vvt2=vvt;
            ss2=sss;
            tv=tv+sampling;
        end
        if t>=tb && t<(tb+tc-waktu)
            vvt=vvt2+vdott*tvv;
            sss=ss2+vvt*tvv-0.5*vdott*(tvv^2);
            vvt3=vvt;
            ss3=sss;
            tvv=tvv+sampling;
        end
        if t>=(tb+tc-waktu)
            vt0=v0;
            vvt=vt0+vdott*tvvv;

```

```

        sss=ss3+vvt*tvvv-0.5*vdott*(tvvv^2);
        tvvv=tvvv+sampling;
    end
end

    if s<=(2*sa)
        if t>=tb && t<(tb+tc-waktu)
            vvt=vvt1+vdott*tvvi;
            sss=ss1+vvt*tvvi-0.5*vdott*(tvvi^2);
            vvt2=vvt;
            ss3=sss;
            tvvi=tvvi+sampling;
        end
        if t>=(tb+tc-waktu)
            vt0=v0;
            vvt=vt0+vdott*tvvvi;
            sss=ss3+vvt*tvvvi-0.5*vdott*(tvvvi^2);
            tvvvi=tvvvi+sampling;
        end
    end
end

    waw(iii+1)=wt;
    tetha(iii+1)=teth;
    tettha(iii+1)=te;

vv(iii+1)=vvt;
ssss(iii+1)=sss;
iii=iii+1;
t=t+sampling;

    if (deltax>=0 && deltax>0 && deltax<=0) || (deltax>0 && deltax<=0)
        xakhir=xtp+L*cos(teth)+sss*cos(sdt);
        yakhir=ytp+L*sin(teth)+sss*sin(sdt);
    end
    if (deltax<0 && deltax>0) || (deltax<=0 && deltax<=0)
        xakhir=xtp+L*cos(teth)-sss*cos(sdt);
        yakhir=ytp+L*sin(teth)-sss*sin(sdt);
    end

    x1(iiii+1)=xakhir;
    y1(iiii+1)=yakhir;
    iiii=iiii+1;
    ibay=ibay+1;
end

xref=xref+samplingx;
xreff(iiii+1)=xref;
x1(iiii)=xakhir;
y1(iiii)=yakhir;
tbayang(ibay)=tbay;
phi=sdt

    mutphi=abs(phi)
    phix=phi

```

```
if phix<0
    phix=(2*pi)-mutphi
end

xdasar=xakhir
ydasar=yakhir
    if deltax>=0 && deltax>=0
        xtp= xakhir-L*cos(phi)
        ytp= yakhir-L*sin(phi)
    end
    if deltax>0 && deltax<0
        if p>(3*pi/2)
            xtp= xakhir-L*cos((2*pi)-phi)
            ytp= yakhir+L*sin((2*pi)-phi)
        end
        if p<0
            xtp= xakhir-L*cos(mutphi)
            ytp= yakhir+L*sin(mutphi)
        end
    end
    if deltax<0 && deltax>0
        xtp= xakhir+L*cos(phi-(pi/2))
        ytp= yakhir-L*sin(phi-(pi/2))
    end
    if deltax<0 && deltax<0
        xtp= xakhir+L*cos(abs(pi-mutphi))
        ytp= yakhir+L*sin(abs(pi-mutphi))
    end
end
xterakhir=xakhir
yterakhir=yakhir

%% Output
figure(1);
subplot(2,1,1);
plot(tbayang,er);
grid;
xlabel('Waktu (s)'), ylabel('Tegangan(V)');
title('Hubungan tegangan motor kanan terhadap waktu')

subplot(2,1,2);
plot(tbayang,el);
grid;
xlabel('Waktu (s)'), ylabel('Tegangan(V)');
title('Hubungan tegangan motor kiri terhadap waktu')

figure(2);
plot(xtray,ytray,'r');
hold
plot(x1,y1);
grid;
xlabel('X'), ylabel('Y');
title('Koordinat posisi robot')

end
```

## **LAMPIRAN E**

---

**PROGRAM SIMULASI PERGERAKAN ROBOT  
MOBIL BERODA MELALUI TRAYEKTORI DENGAN  
GANGGUAN**

```
em = input('Masukkan tegangan motor (tegangan motor kanan=tegangan
motor kiri)= ');
emrr=em;
er(1)=0;
emll=em;
el(1)=0;

pangkat      = input('Masukkan pangkat untuk persamaan trayektori
:');
xref         = input('Masukkan harga x pertama robot mobil bergerak
:');
xpterakhir  = input('Masukkan harga x terakhir robot berhenti
:');
yref=xref;

x1(1)=xref;
xaw=xref;
yaw=yref;
y1(1)=yaw;
sampling=0.1;
samplingx=0.1;
phi = (pi/2);

RPM=263;
Vmin=3;
Vmaks=8;

if em<=Vmin
    'Tegangan diberikan pada motor terlalu kecil '
    Vmin
    tnoise;
elseif em>Vmaks
    'Tegangan diberikan pada motor terlalu besar '
    Vmaks
    tnoise;
elseif em>Vmin && em<=Vmaks

selisih=Vmaks-Vmin;
kec=RPM/selisih;
kenaikan=emrr-Vmin;

Ks=kec*kenaikan;
r2=25e-3;
r1=1e-3;
Ki=3e-1;
R=10000;
Lk=0.63;
Mhub=0.025;
Mwheel=0.05;
Mrobot=3;
m=Mrobot;
rin=0.03;
rout=0.05;
Jmr=0.56e7;
Jml=0.56e7;
```

```
b=0.05;
L=0.1;

xtray=xref:samplingx:xpterakhir;
ytray=xtray.^pangkat;

    xnois=xref;
    nois=0;
    sudut=0;
    sud=0;
    while sudut<(2*pi)
        sudut=sudut+0.01
        suduts(sud+1)=sudut;
        sud=sud+1;
    end
    sudutt=randperm(sud);
    sudutt1=sudutt(1);
    suduttt=suduts(sudutt1);
    while xnois<xpterakhir
        xnois=xnois+samplingx;
        xnoiss(nois+1)=xnois
        nois=nois+1;
    end
    noiss=randperm(nois);
    noiss1=noiss(1);
    noisss=xnoiss(noiss1);

format long
xtp=xaw-(L*cos(abs(phi)));
ytp=yaw-(L*sin(abs(phi)));

    mutphi=abs(phi)
    phix=phi
    if phix<0
        phix=(2*pi)-mutphi
    end
vcmot=Ks*(2*pi*r1)/60;
wcmot=vcmot/r2;
vc=wcmot*rout
iiii=1;

tbay=0;
ibay=1;
tbayang(1)=0;

while xref<xpterakhir
    xr=xref

    xrr=xr+samplingx

    yrr=xrr^pangkat
    deltax=xrr-xtp
    deltay=yrr-ytp
```

```
p= atan2(deltay,deltax)
mutp=abs(p);
px=p
if p<0
    px=(2*pi)+p
end
if deltax>=0 && deltax>=0
    xharus=xtp+L*cos(px);
    yharus=ytp+L*sin(px);
end
if deltax>0 && deltax<0
    xharus= xtp+L*cos((2*pi)-p);
    yharus= ytp-L*sin((2*pi)-p);
end
if deltax<0 && deltax>0
    xharus= xtp-L*cos(abs(phi));
    yharus= ytp+L*sin(abs(phi));
end
if deltax<0 && deltax<0
    xharus= xtp-L*cos(abs(pi-mutp));
    yharus= ytp-L*sin(abs(pi-mutp));
end
deltaax=xrr-xharus;
deltaay=yrr-yharus;
s=sqrt((deltaax)^2+(deltaay)^2)
smust=s;

while vt<=vc
    ttti(iii+1)=ti;
    emr=emrr;
    eml=emll;

kg=r2/r1;
ir1=(emr/R)-(emr/R)*exp((-R/Lk)*ti);
il1=(eml/R)-(eml/R)*exp((-R/Lk)*ti);
ir=ir1;
il=il1;
Eemfr1=-emr*exp((-R/Lk)*ti);
Eemfl1=-eml*exp((-R/Lk)*ti);
Eemfr=Eemfr1;
Eemfl=Eemfl1;

wmr=Eemfr/Ks;
wml=Eemfl/Ks;
Tmr=ir*Ki;
Tml=il*Ki;
TLr=Tmr*kg;
TLl=Tml*kg;
wLr=(1/kg)*wmr;
wLl=(1/kg)*wml;
Jhub=Mhub*rin^2/2;
Jwheel=Mwheel*(rin^2+rout^2)/2;
Jrobot=Mrobot*rout^2/2;
J=Jrobot;
```

```
JL=Jhub+Jwheel+Jrobot;
Jr=JL+kg^2*Jmr;
Jl=JL+kg^2*Jml;
wLrdot=TLr/Jr;
wLldot=TLl/Jl;
fr=(Jr*wLrdot)/rout;
fl=(Jl*wLldot)/rout;
F=(fr+fl)/2;
vdoot=((1/(2*m))*fr) + ((1/(2*m))*fl);

vt=v0+vdoot*ti;
s=s0+vt*ti+0.5*vdoot*(ti^2);

vaw(iii+1)=vt;
ssaw(iii+1)=s;
iii=1+iii;
ti=ti+sampling;
if s<smust
    tmust=ti;
end
end
tper=ti;
tpercepatan=ti/60
spercepatan=s
vcc=vt

teta=phi-p
tetax=teta;
if teta<0 || teta >=pi
    tetal=abs(teta);
    if tetal>=pi && tetal<=(2*pi)
        teta=2*pi-tetal
    elseif tetal>=2*pi
        teta=tetal-(2*pi)
    else
        teta=tetal
    end
end

kel=2*pi*(2*b);
fi=(vcc/kel)*(2*pi);
radl=fi*sampling;
radi=radl*2;
n=teta/radi;
mut=abs(teta);
waktu=(mut/radi)*sampling;

sa=spercepatan;
tsam=tpercepatan;
scm=smust-(2*sa);
tcom=(scm/vcc)/60;
t1m=tsam+tcom;
totm=(tsam*2)+tcom;
tcm=tsam*60+waktu;
tbm=t1m*60+waktu;
```



```
    tam=tbm+tcn-waktu;
if tbn<=(tsam*60)
    scm=0;
    tcom=0;
    t1m=((tmust/60)/2);
    totm=tmust/60;
    tcm=t1m*60+waktu;
    tbn =t1m*60+waktu;
    tam=tbn+tcn-waktu;
end
    totalwaktu=tam+waktu;

while t<=totalwaktu
    ttt(iii+1)=t;
    tbay=tbay+sampling;
    tbayang(ibay+1)=tbay;
    if t<waktu
        if (phi>=0 && phi<=(pi/2)) || (phi>(pi/2) && phi<pi)
            if (deltaax>0 && deltaay>0)
                if phix==px
                    emr=emrr;
                    eml=emll;
                end

                if px>phix && px<=(pi+phix)
                    emr=emrr;
                    eml=-emll;
                else
                    emr=-emrr
                    eml=emll
                end

                if phi>(pi/2) && phi<pi;
                    emr=-emrr
                    eml=emll
                end
            end

            if (deltaax<0 && deltaay>=0)
                if phi>(pi/2) && phi<pi;
                    if phix==px
                        emr=emrr;
                        eml=emll;
                    end
                    if px>phix && px<=(pi+phix)
                        emr=emrr;
                        eml=-emll;
                    else
                        emr=-emrr
                        eml=emll
                    end
                else
                    emr=emrr;
                    eml=-emll;
                end
            end
        end
    end
end
```

```

end

if (deltaax<=0 && deltaay<=0)
  if px>phix && px<=(pi+phix)
    emr=emrr;
    eml=-emll;
  else
    emr=-emrr;
    eml=emll;
  end
  if phi>(pi/2) && phi<pi;
    emr=emrr;
    eml=-emll;
  end
end

if (deltaax>0 && deltaay<0)
  if phi>(pi/2) && phi<pi;
    if px>phix && px<(pi+phix)
      emr=emrr;
      eml=-emll;
    else
      emr=-emrr
      eml=emll
    end
  else
    emr=-emrr
    eml=emll
  end
end
er(iiii+1)=emr;
el(iiii+1)=eml;
ir1=(emr/R)-(emr/R)*exp((-R/Lk)*t);
il1=(eml/R)-(eml/R)*exp((-R/Lk)*t);
ir=ir1;
il=il1;
Eemfr1=-emr*exp((-R/Lk)*t);
Eemfl1=-eml*exp((-R/Lk)*t);
Eemfr=Eemfr1;
Eemfl=Eemfl1;
end

if (phi<0 && phi>=(-pi/2)) || (phi<(-pi/2) && phi>(-
pi)) || (phi>(pi) && phi<(3*pi/2)) || (phi>=(3*pi/2) &&
phi<(2*pi))
  if (deltaax>0 && deltaay>0)
    if (phi<(-pi/2) && phi>(-pi)) || (phi>(pi) &&
phi<(3*pi/2))
      if px<phix && px>(phix-pi)
        emr=-emrr
        eml=emll
      else
        emr=emrr;
        eml=-emll;
      end
    else

```

```
        emr=emrr;
        eml=-emll;
    end
end

if (deltaax<0 && deltaay>=0)
    if px<phix && px>(phix-pi)
        emr=-emrr;
        eml=emll;
    else
        emr=emrr;
        eml=-emll;
    end
    if (phi<(-pi/2) && phi>(-pi)) || (phi>(pi) &&
phi<(3*pi/2))
        emr=-emrr
        eml=emll
    end
end

if (deltaax<=0 && deltaay<=0)
    if (phi<(-pi/2) && phi>(-pi)) || (phi>(pi) &&
phi<(3*pi/2))
        if phix==px
            emr=emrr;
            eml=emll;
        end
        if px<phix && px>(phix-pi)
            emr=-emrr
            eml=emll
        else
            emr=emrr;
            eml=-emll;
        end
    else
        emr=-emrr
        eml=emll
    end
end

if (deltaax>0 && deltaay<0)
    if (phi<(-pi/2) && phi>(-pi)) || (phi>(pi) &&
phi<(3*pi/2))
        emr=emrr;
        eml=-emll;
    else
        if phix==px
            emr=emrr;
            eml=emll;
        end
        if px<phix && px>(phix-pi)
            emr=-emrr;
            eml=emll;
        else
            emr=emrr;
            eml=-emll;
        end
    end
end
```

```

        end
    end
end
er(iiii+1)=emr;
el(iiii+1)=eml;
ir1=(emr/R) - (emr/R)*exp((-R/Lk)*t);
il1=(eml/R) - (eml/R)*exp((-R/Lk)*t);
ir=ir1;
il=il1;
Eemfr1=-emr*exp((-R/Lk)*t);
Eemfl1=-eml*exp((-R/Lk)*t);
Eemfr=Eemfr1;
Eemfl=Eemfl1;
end
end
if p==phi
    waktu=0;
    teth=phi;
    sdt=teth;
end

if t>=waktu
    deltaxs=xrr-xharus;
    deltays=yrr-yharus;
    s=sqrt(((deltaxs)^2)+((deltays)^2));
    if s>(2*sa)
        sc=s-(2*sa);
        tco=(sc/vcc)/60;
        t1=tsam+tco;
        tot=(tsam*2)+tco;
        tc=tsam*60+waktu;
        tb=t1*60+waktu;
        ta=tb+tc-waktu;
    end
    if s==(2*sa)
        sc=0;
        tco=0;
        t1=tsam/2;
        tot=tsam;
        tc=t1*60+waktu;
        tb=tsam/2+waktu;
        ta=tb+tc-waktu;
    end
    if s<(2*sa)
        sc=0;
        tco=0;
        t1=(tmust/60)/2;
        tot=tmust/60;
        tc=t1*60+waktu;
        tb=t1*60+waktu;
        ta=tb+tc-waktu;
    end

    if t<tc
        emr=emrr;
        eml=emll;
    end
end

```

```

er(iiii+1)=emr;
el(iiii+1)=eml;
ir1=(emr/R) - (emr/R)*exp((-R/Lk)*ts);
il1=(eml/R) - (eml/R)*exp((-R/Lk)*ts);
ir=ir1;
il=il1;
Eemfr1=-emr*exp((-R/Lk)*ts);
Eemfl1=-eml*exp((-R/Lk)*ts);
Eemfr=Eemfr1;
Eemfl=Eemfl1;
ts=ts+sampling;
end

if s>(2*sa)
  if t>=tc && t<tb
    emr=emrr;
    eml=emll;
    er(iiii+1)=emr;
    el(iiii+1)=eml;
    ir1=(emr/R) - (emr/R)*exp((-R/Lk)*ts);
    il1=(eml/R) - (eml/R)*exp((-R/Lk)*ts);
    ir2=- (emr/R) + (emr/R)*exp((-R/Lk)*t0);
    il2=- (eml/R) + (eml/R)*exp((-R/Lk)*t0);
    ir=ir1+ir2;
    il=il1+il2;
    Eemfr1=-emr*exp((-R/Lk)*ts);
    Eemfl1=-eml*exp((-R/Lk)*ts);
    Eemfr2=emr*exp((-R/Lk)*t0);
    Eemfl2=eml*exp((-R/Lk)*t0);

    Eemfr=Eemfr1+Eemfr2;
    Eemfl=Eemfl1+Eemfl2;
    ts=ts+sampling;
    t0=t0+sampling;
  end

  if t>=tb && t<(tb+tc-waktu)
    emr=emrr;
    eml=emll;
    er(iiii+1)=0;
    el(iiii+1)=0;
    ir1=(emr/R) - (emr/R)*exp((-R/Lk)*ts);
    il1=(eml/R) - (eml/R)*exp((-R/Lk)*ts);
    ir2=- (emr/R) + (emr/R)*exp((-R/Lk)*t0);
    il2=- (eml/R) + (eml/R)*exp((-R/Lk)*t0);
    ir1off=- ((emr/R) - (emr/R)*exp((-R/Lk)*t00));
    il1off=- ((eml/R) - (eml/R)*exp((-R/Lk)*t00));
    ir=ir1+ir2+ir1off;
    il=il1+il2+il1off;
    Eemfr1=-emr*exp((-R/Lk)*ts);
    Eemfl1=-eml*exp((-R/Lk)*ts);
    Eemfr2=emr*exp((-R/Lk)*t0);
    Eemfl2=eml*exp((-R/Lk)*t0);
    Eemfr1off=emr*exp((-R/Lk)*t00);
    Eemfl1off=eml*exp((-R/Lk)*t00);
  end
end

```

```

Eemfr=Eemfr1+Eemfr2+Eemfrloff;
Eemfl=Eemfl1+Eemfl2+Eemflloff;
t0=t0+sampling;
t00=t00+sampling;
ts=ts+sampling;
end

if t>=(tb+tc-waktu)
    emr=emrr;
    eml=eml1;
    er(iiii+1)=0;
    el(iiii+1)=0;
    ir1=(emr/R)-(emr/R)*exp((-R/Lk)*ts);
    il1=(eml/R)-(eml/R)*exp((-R/Lk)*ts);
    ir2=- (emr/R)+(emr/R)*exp((-R/Lk)*t0);
    il2=- (eml/R)+(eml/R)*exp((-R/Lk)*t0);
    irloff=- ((emr/R)-(emr/R)*exp((-R/Lk)*t00));
    illoff=- ((eml/R)-(eml/R)*exp((-R/Lk)*t00));
    ir2off=(emr/R)-(emr/R)*exp((-R/Lk)*t000);
    il2off=(eml/R)-(eml/R)*exp((-R/Lk)*t000);
    ir=ir1+ir2+irloff+ir2off;
    il=il1+il2+illoff+il2off;
    Eemfr1=-emr*exp((-R/Lk)*ts);
    Eemfl1=-eml*exp((-R/Lk)*ts);
    Eemfr2=emr*exp((-R/Lk)*t0);
    Eemfl2=eml*exp((-R/Lk)*t0);
    Eemfrloff=emr*exp((-R/Lk)*t00);
    Eemflloff=eml*exp((-R/Lk)*t00);
    Eemfr2off=-emr*exp((-R/Lk)*t000);
    Eemfl2off=-eml*exp((-R/Lk)*t000);
    Eemfr=Eemfr1+Eemfr2+Eemfrloff+Eemfr2off;
    Eemfl=Eemfl1+Eemfl2+Eemflloff+Eemfl2off;
    ts=ts+sampling;
    t0=t0+sampling;
    t00=t00+sampling;
    t000=t000+sampling;
end

end

if s<(2*sa)
    if t>=tb && t<(tb+tc-waktu)
        emr=emrr;
        eml=eml1;
        er(iiii+1)=0;
        el(iiii+1)=0;
        ir1=(emr/R)-(emr/R)*exp((-R/Lk)*ts);
        il1=(eml/R)-(eml/R)*exp((-R/Lk)*ts);
        ir2=- (emr/R)+(emr/R)*exp((-R/Lk)*t0);
        il2=- (eml/R)+(eml/R)*exp((-R/Lk)*t0);
        irloff=- ((emr/R)-(emr/R)*exp((-R/Lk)*t00));
        illoff=- ((eml/R)-(eml/R)*exp((-R/Lk)*t00));
        ir=ir1+ir2+irloff;
        il=il1+il2+illoff;
        ia(iii+1)=ir;
        ib(iii+1)=il;
    end
end

```

```

Eemfr1=-emr*exp((-R/Lk)*ts);
Eemfl1=-eml*exp((-R/Lk)*ts);
Eemfr2=emr*exp((-R/Lk)*t0);
Eemfl2=eml*exp((-R/Lk)*t0);
Eemfr1off=emr*exp((-R/Lk)*t00);
Eemfl1off=eml*exp((-R/Lk)*t00);

Eemfr=Eemfr1+Eemfr2+Eemfr1off;
Eemfl=Eemfl1+Eemfl2+Eemfl1off;
ts=ts+sampling;
t0=t0+sampling;
t00=t00+sampling;
end

if t>=(tb+tc-waktu)
    emr=0;
    eml=0;
    er(iiii+1)=0;
    el(iiii+1)=0;
    ir1=(emr/R)-(emr/R)*exp((-R/Lk)*ts);
    il1=(eml/R)-(eml/R)*exp((-R/Lk)*ts);
    ir2=-emr/R+(emr/R)*exp((-R/Lk)*t0);
    il2=-eml/R+(eml/R)*exp((-R/Lk)*t0);
    ir1off=-((emr/R)-(emr/R)*exp((-R/Lk)*t00));
    il1off=-((eml/R)-(eml/R)*exp((-R/Lk)*t00));
    ir2off=(emr/R)-(emr/R)*exp((-R/Lk)*t000);
    il2off=(eml/R)-(eml/R)*exp((-R/Lk)*t000);
    ir=ir1+ir2+ir1off+ir2off;
    il=il1+il2+il1off+il2off;
    Eemfr1=-emr*exp((-R/Lk)*ts);
    Eemfl1=-eml*exp((-R/Lk)*ts);
    Eemfr2=emr*exp((-R/Lk)*t0);
    Eemfl2=eml*exp((-R/Lk)*t0);
    Eemfr1off=emr*exp((-R/Lk)*t00);
    Eemfl1off=eml*exp((-R/Lk)*t00);
    Eemfr2off=-emr*exp((-R/Lk)*t000);
    Eemfl2off=-eml*exp((-R/Lk)*t000);
    Eemfr=Eemfr1+Eemfr2+Eemfr1off+Eemfr2off;
    Eemfl=Eemfl1+Eemfl2+Eemfl1off+Eemfl2off;
    ts=ts+sampling;
    t0=t0+sampling;
    t00=t00+sampling;
    t000=t000+sampling;
end
end
end

kg=r2/r1;
wmr=Eemfr/Ks;
wml=Eemfl/Ks;
Tmr=ir*Ki;
Tml=il*Ki;
TLr=Tmr*kg;
TLl=Tml*kg;

```

```
wLr=(1/kg)*wmr;
wLl=(1/kg)*wml;
Jhub=Mhub*rin^2/2;
Jwheel=Mwheel*(rin^2+rout^2)/2;
Jrobot=Mrobot*rout^2/2;
J=Jrobot;
JL=Jhub+Jwheel+Jrobot;
Jr=JL+kg^2*Jmr;
Jl=JL+kg^2*Jml;
wLrdot=TLr/Jr;
wLldot=TLl/Jl;
fr=(Jr*wLrdot)/rout;
fl=(Jl*wLldot)/rout;
F=(fr+fl)/2;
vdott=((1/(2*m))*fr) + ((1/(2*m))*fl);
vdotrr(iii+1)=vdott;
wdoot=((1/(2*b*J))*fr - (1/(2*b*J))*fl);
wdotrr(iii+1)=wdoot;
wt=wt+wdoot*t;

    if t<waktu
        vvt=v0+vdott*t;
        sss=s0+vvt*t+0.5*vdott*(t^2);
        vvtw1=vvt;
        s0=sss;

        if (phi>=0 && phi<=(pi/2)) || (phi>(pi/2) && phi<pi)
            if (deltaax>0 && deltaay>0)
                te=te+radi;
                if phix==px
                    teth=phi; 'lurus'
                    sdt=phi;
                end
                if px>phix && px<=(pi+phix)
                    teth=phi+te; 'kiri'
                    sdt=teth;
                else
                    teth=phi-te; 'kanan'
                    sdt=teth;
                end
                if phi>(pi/2) && phi<pi;
                    teth=phi-te; 'kanan'
                    sdt=teth;
                end
            end
        end

        if (deltaax<0 && deltaay>=0)
            te=te+radi;
            teth=phi+te; 'kiri'
            sdt=pi+teth;

            if phi>(pi/2) && phi<pi;
                if phix==px
                    teth=phi; 'lurus'
                    sdt=phi;
                end
            end
        end
    end
```



```

        end
        if px>phix && px<=(pi+phix)
            teth=phi+te; 'kiri'
            sdt=pi+teth;
        else
            teth=phi-te; 'kanan'
            sdt=pi+teth;
        end
    end
end
if (deltaax<=0 && deltaay<=0)
    te=te+radi;
    if px>phix && px<=(pi+phix)
        teth=phi+te; 'kiri'
        sdt=teth-pi;
    else
        teth=phi-te; 'kanan'
        sdt=teth-pi;
    end
end

    if phi>(pi/2) && phi<pi;
        teth=phi+te; 'kiri'
        sdt=teth-pi;
    end
end
if (deltaax>0 && deltaay<0)
    te=te+radi;
    teth=phi-te; 'kanan'
    sdt=teth;
    if phi>(pi/2) && phi<pi;
        if px>phix && px<(pi+phix)
            teth=phi+te; 'kiri'
            sdt=teth;
        else
            teth=phi-te; 'kanan'
            sdt=teth;
        end
    end
end
end
end

    if (phi<0 && phi>=(-pi/2)) || (phi<(-pi/2) && phi>(-
pi)) || (phi>(pi) && phi<(3*pi/2)) || (phi>=(3*pi/2) &&
phi<(2*pi))
        if (deltaax>0 && deltaay>0)
            te=te+radi; 'kiri'
            teth=phi+te;
            sdt=teth;

            if (phi<(-pi/2) && phi>(-pi)) || (phi>(pi) &&
phi<(3*pi/2))
                if px<phix && px>(phix-pi)
                    teth=phi-te; 'kanan'
                    sdt=teth;
                else
                    teth=phi+te; 'kiri'
                end
            end
        end
    end
end

```

```

        sdt=teth;
    end
end
end

if (deltaax<0 && deltaay>=0)
    te=te+radi;
    if px<phix && px>(phix-pi)
        teth=phi-te; 'kanan'
        sdt=pi+teth;
    else
        teth=phi+te; 'kiri'
        sdt=pi+teth;
    end
    if (phi<(-pi/2) && phi>(-pi)) || (phi>(pi) &&
phi<(3*pi/2))
        teth=phi-te; 'kanan'
        sdt=pi+teth;
    end
end

if (deltaax<=0 && deltaay<=0)
    te=te+radi;
    teth=phi-te; 'kanan'
    sdt=pi+teth;
    if (phi<(-pi/2) && phi>(-pi)) || (phi>(pi) &&
phi<(3*pi/2))
        if phix==px
            teth=phi; 'lurus'
            sdt=phi;
        end
        if px<phix && px>(phix-pi)
            teth=phi-te; 'kanan'
            sdt=pi+teth;
        else
            teth=phi+te; 'kiri'
            sdt=pi+teth;
        end
    end
end

if (deltaax>0 && deltaay<0)
    te=te+radi;
    if phix==px
        teth=phi; 'lurus'
        sdt=phi;
    end
    if px<phix && px>(phix-pi)
        teth=phi-te; 'kanan'
        sdt=teth;
    else
        teth=phi+te; 'kiri'
        sdt=teth;
    end

    if (phi<(-pi/2) && phi>(-pi)) || (phi>(pi) &&
phi<(3*pi/2))

```

```
teth=phi+te; 'kiri'
sdt=teth;
end
end
end
if teth<0
teth=(2*pi)+teth;
end
xharus=xtp+L*cos(teth);
yharus=ytp+L*sin(teth);
if t>=waktu
if t<tc
vvt=v0+vdott*tss;
sss=s0+vvt*tss+0.5*vdott*(tss^2);
vvt1=vvt;
ss1=sss;
tss=tss+sampling;
end
if s>(2*sa)
if t>=tc && t<tb
vvt=vvt1+vdott*tv;
sss=ss1+vvt*tv+0.5*vdott*(tv^2);
vvt2=vvt;
ss2=sss;
tv=tv+sampling;
end
if t>=tb && t<(tb+tc-waktu)
vvt=vvt2+vdott*tvv;
sss=ss2+vvt*tvv-0.5*vdott*(tvv^2);
vvt3=vvt;
ss3=sss;
tvv=tvv+sampling;
end
if t>=(tb+tc-waktu)
vt0=v0;
vvt=vt0+vdott*tvvv;
sss=ss3+vvt*tvvv-0.5*vdott*(tvvv^2);
tvvv=tvvv+sampling;
end
end
if s<=(2*sa)
if t>=tb && t<(tb+tc-waktu)
vvt=vvt1+vdott*tvvi;
sss=ss1+vvt*tvvi-0.5*vdott*(tvvi^2);
vvt2=vvt;
ss3=sss;
tvvi=tvvi+sampling;
end
if t>=(tb+tc-waktu)
vt0=v0;
vvt=vt0+vdott*tvvvi;
sss=ss3+vvt*tvvvi-0.5*vdott*(tvvvi^2);
tvvvi=tvvvi+sampling;
end
end
end
```

```

        end
        waw(iii+1)=wt;
        tetha(iii+1)=teth;
        tettha(iii+1)=te;
vv(iii+1)=vvt;
ssss(iii+1)=sss;
iii=iii+1;
t=t+sampling;
    if (deltax>=0 && deltay>0) || (deltax>0 && deltay<=0)
        xakhir=xtp+L*cos(teth)+sss*cos(sdt);
        yakhir=ytp+L*sin(teth)+sss*sin(sdt);
    end
    if (deltax<0 && deltay>0) || (deltax<=0 && deltay<=0)
        xakhir=xtp+L*cos(teth)-sss*cos(sdt);
        yakhir=ytp+L*sin(teth)-sss*sin(sdt);
    end
    xakhir
    yakhir
    x1(iiii+1)=xakhir;
    y1(iiii+1)=yakhir;
    iiii=iiii+1;
    ibay=ibay+1;
end
xref=xref+samplingx;
xreff(iiii+1)=xref;
x1(iiii)=xakhir;
y1(iiii)=yakhir;
tbayang(ibay)=tbay;
phi=sdt
mutphi=abs(phi)
    phix=phi
    if phix<0
        phix=(2*pi)-mutphi
    end
    if deltax>=0 && deltay>=0
        xtp= xakhir-L*cos(phi)
        ytp= yakhir-L*sin(phi)
    end
    if deltax>0 && deltay<0
        xtp= xakhir-L*cos((2*pi)-phi)
        ytp= yakhir+L*sin((2*pi)-phi)
    end
    if deltax<0 && deltay>0
        xtp= xakhir+L*cos(abs(phi))
        ytp= yakhir-L*sin(abs(phi))
    end
    if deltax<0 && deltay<0
        xtp= xakhir+L*cos(abs(pi-mutphi))
        ytp= yakhir+L*sin(abs(pi-mutphi))
    end
if xref==noisss && xref<=xpterakhir
    xn=random('Normal',0:0,2);
    xakhir=abs(xn);
    xxn=xakhir;
    yn=random('Normal',0:0,2);
    yakhir=abs(yn);

```

```
        yyn=yakhir;
        x1(iiii+1)=xxn;
        y1(iiii+1)=yyn;
        iiii=iiii+1;
        tbayang(ibay+1)=tbay;
        ibay=ibay+1;
        phi=suduttt;
        phix=phi
        if phix<0
            phix=(2*pi)-mutphi
        end
        xtp= xakhir-L*cos(abs(phix))
        ytp= yakhir-L*sin(abs(phix))
        xref=xpterakhir-samplingx;
    end

end

xgangguan=xxn
ygangguan=yyn
pos=suduttt
posisigangguan=((suduttt/(2*pi))*360)
xterakhir=xakhir
yterakhir=yakhir

%% Output
figure(1);
subplot(2,1,1);
plot(tbayang,er);
grid;
xlabel('Waktu (s)'), ylabel('Tegangan(V)');
title('Hubungan tegangan motor kanan terhadap waktu')

subplot(2,1,2);
plot(tbayang,el);
grid;
xlabel('Waktu (s)'), ylabel('Tegangan(V)');
title('Hubungan tegangan motor kiri terhadap waktu')

figure(2);
plot(xtray,ytray,'r');
hold;
grid;

plot(x1,y1);
grid;
xlabel('X'), ylabel('Y');
title('Koordinat posisi robot')

end
```