LAMPIRAN A

MATLAB

```
%-----Simulasi Error Correcting Code dengan algoritma------%
%--- Berlekamp-Massey atau Peterson-Gorenstein-Zierler ----%
%---------------- Pada kode Reed-Solomon -------------------%

%*** Parameter Kode RS ***
m = 4          %Jumlah bit dalam 1 simbol, GF(2^m)
n = 2^m - 1    %Panjang kode RS
k = 3          %Jumlah data dalam satu kode RS
h = n-k        %Panjang parity tiap kode RS
t = h/2        %Jumlah maksimum error yang dapat dikoreksi


%****Membangkitkan Galois Field dan Generator polynomial***%

% Membentuk Galois Field
field = gftuple([-1:2^m-2]', m, 2);

%Membangkitkan polinomial generator
c = [1 0];
p(1) = c(1);

for i = 1:h-1
    p(1) = gfmul(p(1),1,field);
    p(2) = 0;
    c = gfconv(c,p,field);
end
g = c;


%**************** Pengkodean Reed-Solomon *****************%

%Membangkitkan data acak integer
DATA_IN = randint(1,k,[0 n-1]);

%Membentuk kode RS
parity = RS_ENC4(DATA_IN,n,k,g,field);
RS_CODE = [parity DATA_IN];

%************************ Channel ************************%
RECEIVED = RS_CODE

%************Menambahkan error secara manual**************%
%*********dengan nilai acak di sembarang posisi***********%
RECEIVED(3) = gfadd(RECEIVED(3),randint(1,1,[-1 n-1]),field);
RECEIVED(5) = gfadd(RECEIVED(3),randint(1,1,[-1 n-1]),field);

%****************

%***** Pendekodean dengan algoritma Berlekamp-Massey ******%
DECODED = RS_E_DEC(RECEIVED,n,k,t,h,g,field);

%*Pendekodean dengan algoritma Peterson-Gorenstein-Zierler*%
DECODED = RS_E_DEC_PGZ(RECEIVED,n,k,t,h,g,field);
```

```
%****************
if all(DECODED == RS_CODE)
    disp('Decoding Success')
else
    disp('Decoding Failure')
end



%----------Subroutine pembentukan kode Reed-Solomon--------%
function R = RS_ENC(code,n,k,g,field)
for ii = 1:n-k
  shiftpol(ii)= -inf;
end
shiftpol(n-k+1)=0;
shiftcode=gfconv(code,shiftpol,field);

[Q, R]=gfdeconv(shiftcode,g,field);
while length(R)< n-k
    R=[R -inf];
end


%--Subroutine pendekodean dgn algoritma Berlekamp-Massey---%
function DECODED = RS_E_DEC(received,n,k,t,h,g,field);

%******************Menghitung nilai sindrom**********************%
S = [];
%Substitusi nilai alpha^i dari tiap polinomial yg diterima
for ii = 1:2*t
    S(ii)= -Inf;
    for cc = 1:n
      S(ii) = gfadd(S(ii),gfmul(received(cc),gfpow(ii,cc-
      1,n),field) ,field); %Sum all the terms
    end
end
%Periksa apakah nilai semua sindrom = 0,
%Jika semua nilai sindrom 0, artinya tidak terjadi error
for i = 1:2*t
    pol_tes(i) = -Inf;
end

if all (S == pol_tes)
    message = received;
else

%*Hitung Error Locator Polynomial dgn algoritma Berlekamp-Massey*%
        sigma = BMA(n,k,t,S,field);

%*Mencari nilai akar dr error locator polynomial dg Chien search*%
    akar_polinomial = [];
    kk = 0;
      for ii = 0:n-1
      error_r = -Inf;
```

```
  for cc = 1:length(sigma)
  error_r = gfadd(error_r,gfmul(sigma(cc),gfpow(ii,cc-1,n),field)
,field); %Sum all the terms
  end
            if error_r == -Inf
                kk = kk + 1;
                akar_polinomial(kk) = ii;
            end
        end

 % Mengecek nilai akar apakah bernilai real
 % Membentuk kembali error locator polynomial dgn mengalikan tiap
 % akar yg didapat
 % Lalu bagi error locator polynomial dgn polinomial_tes
 % Jika panjang sisa hasil bagi lbh dr 1 maka akar tidak real
        pol_tes = 0;
        for ii = 1:length(akar_polinomial)
            pol_tes = gfconv(pol_tes,[akar_polinomial(ii)
0],field);
        end

        [QQ,RR] = gfdeconv(sigma,pol_tes,field);
        if length(QQ) > 1
            DECODED = received;
            return
        end

 %******************** Mencari posisi error *********************%
        lokasi_error = [];
          for ii = 1:length(akar_polinomial)
            lokasi_error(ii) = gfdiv(0,akar_polinomial(ii),field);
          end

 %********Menghitung Error Magnitude dgn algoritma Forney********%

    %Hitung polinomial error magnitude:
    %1.  Membentuk fungsi [1 + S(x)]
    SS(1) = 0;
    for ii = 1: 2*t
        SS(ii+1) = S(ii);
    end

    %2. Membentuk persamaan (x)= ?(x) [1 + S(x)]
    OMEGA = gfconv(sigma,gfadd(0,SS,field),field);

    %3. OMEGA = (SS * sigma)mod(x^(2t+1))
    %3.1. Bentuk fungsi := x^(2t+1)
    for ii = 1: (2*t)
        DIV(ii)= -Inf;
    end
    DIV(2*t+1) = 0;

    %3.2.  OMEGA = (SS * sigma)mod(x^(2t+1))
    [DUMMY, OMEGA] = gfdeconv(OMEGA,DIV,field);
```

```
    %OMEGA

    %4. Differentiate the key equation with respect to x
    sigma_diff = gfdiff(sigma);

    %Menghitung error magnitude
    %Substitusi nilai invers ke sigma_diff
    for ii = 1:length(lokasi_error)
ERR_DEN= gfsubstitute(sigma_diff,gfdiv(0,lokasi_error(ii),field),
length(sigma_diff),n,field);
ERR_NUM= gfsubstitute(OMEGA,gfdiv(0,lokasi_error(ii),field),length
(OMEGA),n,field);
ERR_NUM= gfmul(ERR_NUM,lokasi_error(ii),field);
ERR(ii) = gfmul(ERR_NUM,gfdiv(0,ERR_DEN,field),field);
    end

    %Menyusun polinomial error sesuai lokasi dan error magnitude
    for ii = 1:n
        ERR_p(ii) = -Inf;
    end

    for ii = 1:length(lokasi_error)
        pp = lokasi_error(ii);
        ERR_p(pp+1) = ERR(ii);
    end

  % kode RS yang diterima lalu dijumlahkan dengan polinomial error
    message = gfadd(received,ERR_p,field);
end

DECODED = message;


%---Subroutine mencari error locator polynomial dgn BMA----%
function sigma = BMA(n,k,t,S,field)

%Step 2: Inisialisasi variabel
kk = 0;
for i = 1:n
    Kappa(1,i) = -Inf;
end
Kappa(1,1) = 0;
%Kappa=>?(0)(x)=1

%shift register
L = 0;
%C(x)= x
C = [-inf 0];

done = 0;

%Step 3:
while (done ~= 1)
    kk = kk + 1;
```

```
    sum = -Inf;
    for i = 1:L    %Kappa(kk,i+1)      %S(kk-i)
        sum = gfadd(sum,gfmul(Kappa(kk,i+1),S(kk-i),field),field);
    end

    %discrepancy ->pers.2.14
    discrepancy(kk) = gfadd(S(kk),sum,field);

    %Step 4:
    if (discrepancy(kk) == -Inf)
        for i = 1:n
            Kappa(kk+1,i) = Kappa(kk,i);
        end
    end

    if (discrepancy(kk) ~= -Inf)
        for i = 1:n
            Kappa_i(i) = Kappa(kk-1+1,i);
        end

   Kappa_k = gfadd(Kappa_i,gfconv(discrepancy(kk),C,field),field);

        while length(Kappa_k) < n
            Kappa_k = [Kappa_k -Inf];
        end

        for i = 1:length(Kappa_k)
            Kappa(kk+1,i) = Kappa_k(i);
        end
        %Step 7:
        if (2*L < kk)
            L = kk - L;
            for i = 1:n
                Kappa_k(i) = Kappa(kk+1-1,i);
            end
         C = gfconv(Kappa_k,gfdiv(0,discrepancy(kk),field),field);
        end
    end

    %Step 8:
    C = gfconv([-Inf 0],C,field);

    %step 9:
    if kk >= 2*t
        done = 1;
    end
end

for i = 1:n
    sigma(i) = Kappa(kk+1,i);
end


%----------Subroutine pendekodean dgn algoritma PGZ--------%
function DECODED = RS_E_DEC_PGZ(received,n,k,t,h,g,field);
```

```
%*****************Menghitung nilai sindrom**********************%
S = [];
%Substitusi nilai alpha^i dari tiap polinomial yg diterima
for ii = 1:2*t
    S(ii)= -Inf;
    for cc = 1:n
    S(ii) = gfadd(S(ii),gfmul(received(cc),gfpow(ii,cc-1,n),field)
,field); %Sum all the terms
    end
end

%Periksa apakah nilai semua sindrom = 0,Jika semua nilai sindrom
0, artinya tidak terjadi error
for i = 1:2*t
    pol_tes(i) = -Inf;
end

if all (S == pol_tes)
    message = received;
else

%******Hitung Error Locator Polynomial dengan algoritma PGZ******%
   sigma= PGZ(n,k,t,S,field);

%*Mencari nilai akar dr error locator polynomial dgn Chien search%
        akar_polinomial = [];
        kk = 0;

for ii = 0:n-1
   error_r = -Inf;
   for cc = 1:length(sigma)
error_r = gfadd(error_r,gfmul(sigma(cc),gfpow(ii,cc-1,n),field),
field); %Sum all the terms
   end
       if error_r == -Inf
          kk = kk + 1;
          akar_polinomial(kk) = ii;
       end
end

% Mengecek nilai akar adalah real dengan Membentuk kembali error
% locator polynomial dgn mengalikan tiap akar yg didapat
% Lalu bagi error locator polynomial dgn polinomial_tes
% Jika panjang sisa hasil bagi lbh dr 1 maka akar tidak real
        pol_tes = 0;
        for ii = 1:length(akar_polinomial)
            pol_tes = gfconv(pol_tes,[akar_polinomial(ii)
0],field);
        end

        [QQ,RR] = gfdeconv(sigma,pol_tes,field);
        if length(QQ) > 1
            DECODED = received;
            return
```

```
        end

        %Mencari posisi error
        lokasi_error = [];

        for ii = 1:length(akar_polinomial)
            lokasi_error(ii) = gfdiv(0,akar_polinomial(ii),field);
        end

        %disusun dari posisi terendah
        lokasi_error=sort(lokasi_error);

%*******Menghitung Error Magnitude dgn menyusun Matriks Be*******%
        for ii=1:length(akar_polinomial)
            for jj=1:length(akar_polinomial)
                Be(ii,jj)=mod(lokasi_error(jj)*ii,n);
            end
        end


%Mencari besar error jika hanya terjadi 1 error
      if(length(akar_polinomial)==1)
        besar_error(1)=gfdiv(S(1),sigma(2),field);

      else
        determinan=deter(Be,field);
        kofaktor=kofak(Be,field);
        Adj=kofaktor';
        mydets=abs(determinan);

        %Nilai invers matriks
        [q,z]=size(Be);
           for a=1:q
            for d=1:z
                Adj(a,d)=gfdeconv(Adj(a,d),mydets,field);
            end
          end

%Mencari besar error dari brp banyak error yg terjadi
     if(length(akar_polinomial)==2)
besar_error(1)=gfadd(gfconv(Adj(1,1),S(1),field),gfconv(Adj(1,2),S
(2),field),field);
besar_error(2)=gfadd(gfconv(Adj(2,1),S(1),field),gfconv(Adj(2,2),S
(2),field),field);

    else if(length(akar_polinomial)==3)
besar_error(1)=gfadd(gfadd(gfconv(Adj(1,1),S(1),field),gfconv(Adj(
1,2),S(2),field),field),gfconv(Adj(1,3),S(3),field),field);
besar_error(2)=gfadd(gfadd(gfconv(Adj(2,1),S(1),field),gfconv(Adj(
2,2),S(2),field),field),gfconv(Adj(2,3),S(3),field),field);
besar_error(3)=gfadd(gfadd(gfconv(Adj(3,1),S(1),field),gfconv(Adj(
3,2),S(2),field),field),gfconv(Adj(3,3),S(3),field),field);
```

```
    else if(length(akar_polinomial)==4)
besar_error(1)=gfadd(gfadd(gfadd(gfconv(Adj(1,1),S(1),field),gfcon
v(Adj(1,2),S(2),field),field),gfconv(Adj(1,3),S(3),field),field),g
fconv(Adj(1,4),S(4),field),field);
besar_error(2)=gfadd(gfadd(gfadd(gfconv(Adj(2,1),S(1),field),gfcon
v(Adj(2,2),S(2),field),field),gfconv(Adj(2,3),S(3),field),field),g
fconv(Adj(2,4),S(4),field),field);
besar_error(3)=gfadd(gfadd(gfadd(gfconv(Adj(3,1),S(1),field),gfcon
v(Adj(3,2),S(2),field),field),gfconv(Adj(3,3),S(3),field),field),g
fconv(Adj(3,4),S(4),field),field);
besar_error(4)=gfadd(gfadd(gfadd(gfconv(Adj(4,1),S(1),field),gfcon
v(Adj(4,2),S(2),field),field),gfconv(Adj(4,3),S(3),field),field),g
fconv(Adj(4,4),S(4),field),field);

    else if(length(akar_polinomial)==5)
besar_error(1)=gfadd(gfadd(gfadd(gfadd(gfconv(Adj(1,1),S(1),field)
,gfconv(Adj(1,2),S(2),field),field),gfconv(Adj(1,3),S(3),field),fi
eld),gfconv(Adj(1,4),S(4),field),field),gfconv(Adj(1,5),S(5),field
),field);
besar_error(2)=gfadd(gfadd(gfadd(gfadd(gfconv(Adj(2,1),S(1),field)
,gfconv(Adj(2,2),S(2),field),field),gfconv(Adj(2,3),S(3),field),fi
eld),gfconv(Adj(2,4),S(4),field),field),gfconv(Adj(2,5),S(5),field
),field);
besar_error(3)=gfadd(gfadd(gfadd(gfadd(gfconv(Adj(3,1),S(1),field)
,gfconv(Adj(3,2),S(2),field),field),gfconv(Adj(3,3),S(3),field),fi
eld),gfconv(Adj(3,4),S(4),field),field),gfconv(Adj(3,5),S(5),field
),field);
besar_error(4)=gfadd(gfadd(gfadd(gfadd(gfconv(Adj(4,1),S(1),field)
,gfconv(Adj(4,2),S(2),field),field),gfconv(Adj(4,3),S(3),field),fi
eld),gfconv(Adj(4,4),S(4),field),field),gfconv(Adj(4,5),S(5),field
),field);
besar_error(5)=gfadd(gfadd(gfadd(gfadd(gfconv(Adj(5,1),S(1),field)
,gfconv(Adj(5,2),S(2),field),field),gfconv(Adj(5,3),S(3),field),fi
eld),gfconv(Adj(5,4),S(4),field),field),gfconv(Adj(5,5),S(5),field
),field);

    else if(length(akar_polinomial)==6)
besar_error(1)=gfadd(gfadd(gfadd(gfadd(gfadd(gfconv(Adj(1,1),S(1),
field),gfconv(Adj(1,2),S(2),field),field),gfconv(Adj(1,3),S(3),fie
ld),field),gfconv(Adj(1,4),S(4),field),field),gfconv(Adj(1,5),S(5)
,field),field),gfconv(Adj(1,6),S(6),field),field);
besar_error(2)=gfadd(gfadd(gfadd(gfadd(gfadd(gfconv(Adj(2,1),S(1),
field),gfconv(Adj(2,2),S(2),field),field),gfconv(Adj(2,3),S(3),fie
ld),field),gfconv(Adj(2,4),S(4),field),field),gfconv(Adj(2,5),S(5)
,field),field),gfconv(Adj(2,6),S(6),field),field);
besar_error(3)=gfadd(gfadd(gfadd(gfadd(gfadd(gfconv(Adj(3,1),S(1),
field),gfconv(Adj(3,2),S(2),field),field),gfconv(Adj(3,3),S(3),fie
ld),field),gfconv(Adj(3,4),S(4),field),field),gfconv(Adj(3,5),S(5)
,field),field),gfconv(Adj(3,6),S(6),field),field);
```

```
besar_error(4)=gfadd(gfadd(gfadd(gfadd(gfadd(gfconv(Adj(4,1),S(1),
field),gfconv(Adj(4,2),S(2),field),field),gfconv(Adj(4,3),S(3),fie
ld),field),gfconv(Adj(4,4),S(4),field),field),gfconv(Adj(4,5),S(5)
,field),field),gfconv(Adj(4,6),S(6),field),field);
besar_error(5)=gfadd(gfadd(gfadd(gfadd(gfadd(gfconv(Adj(5,1),S(1),
field),gfconv(Adj(5,2),S(2),field),field),gfconv(Adj(5,3),S(3),fie
ld),field),gfconv(Adj(5,4),S(4),field),field),gfconv(Adj(5,5),S(5)
,field),field),gfconv(Adj(5,6),S(6),field),field);
besar_error(6)=gfadd(gfadd(gfadd(gfadd(gfadd(gfconv(Adj(6,1),S(1),
field),gfconv(Adj(6,2),S(2),field),field),gfconv(Adj(6,3),S(3),fie
ld),field),gfconv(Adj(6,4),S(4),field),field),gfconv(Adj(6,5),S(5)
,field),field),gfconv(Adj(6,6),S(6),field),field);

    else if(length(akar_polinomial)==7)
besar_error(1)=gfadd(gfadd(gfadd(gfadd(gfadd(gfadd(gfconv(Adj(1,1)
,S(1),field),gfconv(Adj(1,2),S(2),field),field),gfconv(Adj(1,3),S(
3),field),field),gfconv(Adj(1,4),S(4),field),field),gfconv(Adj(1,5
),S(5),field),field),gfconv(Adj(1,6),S(6),field),field),gfconv(Adj
(1,7),S(7),field),field);

besar_error(2)=gfadd(gfadd(gfadd(gfadd(gfadd(gfadd(gfconv(Adj(2,1)
,S(1),field),gfconv(Adj(2,2),S(2),field),field),gfconv(Adj(2,3),S(
3),field),field),gfconv(Adj(2,4),S(4),field),field),gfconv(Adj(2,5
),S(5),field),field),gfconv(Adj(2,6),S(6),field),field),gfconv(Adj
(2,7),S(7),field),field);

besar_error(3)=gfadd(gfadd(gfadd(gfadd(gfadd(gfadd(gfconv(Adj(3,1)
,S(1),field),gfconv(Adj(3,2),S(2),field),field),gfconv(Adj(3,3),S(
3),field),field),gfconv(Adj(3,4),S(4),field),field),gfconv(Adj(3,5
),S(5),field),field),gfconv(Adj(3,6),S(6),field),field),gfconv(Adj
(3,7),S(7),field),field);

besar_error(4)=gfadd(gfadd(gfadd(gfadd(gfadd(gfadd(gfconv(Adj(4,1)
,S(1),field),gfconv(Adj(4,2),S(2),field),field),gfconv(Adj(4,3),S(
3),field),field),gfconv(Adj(4,4),S(4),field),field),gfconv(Adj(4,5
),S(5),field),field),gfconv(Adj(4,6),S(6),field),field),gfconv(Adj
(4,7),S(7),field),field);

besar_error(5)=gfadd(gfadd(gfadd(gfadd(gfadd(gfadd(gfconv(Adj(5,1)
,S(1),field),gfconv(Adj(5,2),S(2),field),field),gfconv(Adj(5,3),S(
3),field),field),gfconv(Adj(5,4),S(4),field),field),gfconv(Adj(5,5
),S(5),field),field),gfconv(Adj(5,6),S(6),field),field),gfconv(Adj
(5,7),S(7),field),field);

besar_error(6)=gfadd(gfadd(gfadd(gfadd(gfadd(gfadd(gfconv(Adj(6,1)
,S(1),field),gfconv(Adj(6,2),S(2),field),field),gfconv(Adj(6,3),S(
3),field),field),gfconv(Adj(6,4),S(4),field),field),gfconv(Adj(6,5
),S(5),field),field),gfconv(Adj(6,6),S(6),field),field),gfconv(Adj
(6,7),S(7),field),field);

besar_error(7)=gfadd(gfadd(gfadd(gfadd(gfadd(gfadd(gfconv(Adj(7,1)
,S(1),field),gfconv(Adj(7,2),S(2),field),field),gfconv(Adj(7,3),S(
3),field),field),gfconv(Adj(7,4),S(4),field),field),gfconv(Adj(7,5
```

```
),S(5),field),field),gfconv(Adj(7,6),S(6),field),field),gfconv(Adj
(7,7),S(7),field),field);

    else if(length(akar_polinomial)==8)
besar_error(1)=gfadd(gfadd(gfadd(gfadd(gfadd(gfadd(gfadd(gfconv(Ad
j(1,1),S(1),field),gfconv(Adj(1,2),S(2),field),field),gfconv(Adj(1
,3),S(3),field),field),gfconv(Adj(1,4),S(4),field),field),gfconv(A
dj(1,5),S(5),field),field),gfconv(Adj(1,6),S(6),field),field),gfco
nv(Adj(1,7),S(7),field),field),gfconv(Adj(1,8),S(8),field),field);

besar_error(2)=gfadd(gfadd(gfadd(gfadd(gfadd(gfadd(gfadd(gfconv(Ad
j(2,1),S(1),field),gfconv(Adj(2,2),S(2),field),field),gfconv(Adj(2
,3),S(3),field),field),gfconv(Adj(2,4),S(4),field),field),gfconv(A
dj(2,5),S(5),field),field),gfconv(Adj(2,6),S(6),field),field),gfco
nv(Adj(2,7),S(7),field),field),gfconv(Adj(2,8),S(8),field),field);

besar_error(3)=gfadd(gfadd(gfadd(gfadd(gfadd(gfadd(gfadd(gfconv(Ad
j(3,1),S(1),field),gfconv(Adj(3,2),S(2),field),field),gfconv(Adj(3
,3),S(3),field),field),gfconv(Adj(3,4),S(4),field),field),gfconv(A
dj(3,5),S(5),field),field),gfconv(Adj(3,6),S(6),field),field),gfco
nv(Adj(3,7),S(7),field),field),gfconv(Adj(3,8),S(8),field),field);

besar_error(4)=gfadd(gfadd(gfadd(gfadd(gfadd(gfadd(gfadd(gfconv(Ad
j(4,1),S(1),field),gfconv(Adj(4,2),S(2),field),field),gfconv(Adj(4
,3),S(3),field),field),gfconv(Adj(4,4),S(4),field),field),gfconv(A
dj(4,5),S(5),field),field),gfconv(Adj(4,6),S(6),field),field),gfco
nv(Adj(4,7),S(7),field),field),gfconv(Adj(4,8),S(8),field),field);

besar_error(5)=gfadd(gfadd(gfadd(gfadd(gfadd(gfadd(gfadd(gfconv(Ad
j(5,1),S(1),field),gfconv(Adj(5,2),S(2),field),field),gfconv(Adj(5
,3),S(3),field),field),gfconv(Adj(5,4),S(4),field),field),gfconv(A
dj(5,5),S(5),field),field),gfconv(Adj(5,6),S(6),field),field),gfco
nv(Adj(5,7),S(7),field),field),gfconv(Adj(5,8),S(8),field),field);

besar_error(6)=gfadd(gfadd(gfadd(gfadd(gfadd(gfadd(gfadd(gfconv(Ad
j(6,1),S(1),field),gfconv(Adj(6,2),S(2),field),field),gfconv(Adj(6
,3),S(3),field),field),gfconv(Adj(6,4),S(4),field),field),gfconv(A
dj(6,5),S(5),field),field),gfconv(Adj(6,6),S(6),field),field),gfco
nv(Adj(6,7),S(7),field),field),gfconv(Adj(6,8),S(8),field),field);

besar_error(7)=gfadd(gfadd(gfadd(gfadd(gfadd(gfadd(gfadd(gfconv(Ad
j(7,1),S(1),field),gfconv(Adj(7,2),S(2),field),field),gfconv(Adj(7
,3),S(3),field),field),gfconv(Adj(7,4),S(4),field),field),gfconv(A
dj(7,5),S(5),field),field),gfconv(Adj(7,6),S(6),field),field),gfco
nv(Adj(7,7),S(7),field),field),gfconv(Adj(7,8),S(8),field),field);

besar_error(8)=gfadd(gfadd(gfadd(gfadd(gfadd(gfadd(gfadd(gfconv(Ad
j(8,1),S(1),field),gfconv(Adj(8,2),S(2),field),field),gfconv(Adj(8
,3),S(3),field),field),gfconv(Adj(8,4),S(4),field),field),gfconv(A
dj(8,5),S(5),field),field),gfconv(Adj(8,6),S(6),field),field),gfco
nv(Adj(8,7),S(7),field),field),gfconv(Adj(8,8),S(8),field),field);
        end
       end
      end
     end
    end
```

```
   end
  end
 end
%Menyusun polinomial error sesuai lokasi dan error magnitude
    for ii = 1:n
        ERR_p(ii) = -Inf;
    end

    for ii = 1:length(lokasi_error)
        pp = lokasi_error(ii);
        ERR_p(pp+1) = besar_error(ii);
    end

    % kode RS yang diterima lalu dijumlahkan dengan polinomial
error
    % untuk mendapatkan kode RS yang sama dengan yg dikirim
    message = gfadd(received,ERR_p,field);
end
DECODED = message;


%*Subroutine mencari error locator polynomial dgn algoritma PGZ**%
function sigma= PGZ(n,k,t,S,field)

%Menyusun matriks sindrom
c=0;
for i = 1:t
    for j = 1:t
        M(i,j)=S(j+c);
    end
   c=c+1;
end

    for i = 1:n
        Kappa(1,i) = -Inf;
    end
    Kappa(1,1)=0;

%******************************
if (t==1)
 Sigma(1) = gfdeconv(S(2),S(1),field);
 Kappa(1,2)=Sigma(1);

elseif(t==2)
   %Hitung nilai determinan matriks 2x2
   determinan = deter(M,field);
      if (determinan==-inf)
      Sigma(1)=gfdeconv(S(2),S(1),field);
        Kappa(1,2)=Sigma(1);
      else
        %****mencari nilai invers matriks sindrom****%
        kofaktor=kof(M,S,field);
        Adj=kofaktor'; %adjoin adalah tranpose kofaktornya
        mydets=abs(determinan);
        %Nilai invers matriks
```

```
        Adj(1,1)=gfdeconv(Adj(1,1),mydets,field);
        Adj(2,2)=gfdeconv(Adj(2,2),mydets,field);
        Adj(1,2)=gfdeconv(Adj(1,2),mydets,field);
        Adj(2,1)=gfdeconv(Adj(2,1),mydets,field);
Sigma(2)=gfadd(gfconv(Adj(1,1),S(3),field),gfconv(Adj(1,2),S(4),fi
eld),field);
Sigma(1)=gfadd(gfconv(Adj(2,1),S(3),field),gfconv(Adj(2,2),S(4),fi
eld),field);
        Kappa(1,2)=Sigma(1);
        Kappa(1,3)=Sigma(2);
      end

elseif(t==3)
     %hitung nilai determinan 3x3
      determinan = deter(M,field);
      if(determinan==-inf)
         [n,m]=size(M);
          n1=n-1;
          m1=m-1;
          B=zeros(2,2);
          for i=1:n1
              for j=1:m1
                  B(i,j)=M(i,j);
              end
          end
          M=B;
      %Matriks 2x2
      determinan=deter(M,field);
      if(determinan==-inf)
         Sigma(1)=gfdeconv(S(2),S(1),field);
         Kappa(1,2)=Sigma(1);
          else
            kofaktor=kof(M,S,field);
            Adj=kofaktor';
            mydets=abs(determinan);
            Adj(1,1)=gfdeconv(Adj(1,1),mydets,field);
            Adj(2,2)=gfdeconv(Adj(2,2),mydets,field);
            Adj(1,2)=gfdeconv(Adj(1,2),mydets,field);
            Adj(2,1)=gfdeconv(Adj(2,1),mydets,field);
Sigma(2)=gfadd(gfconv(Adj(1,1),S(3),field),gfconv(Adj(1,2),S(4),fi
eld),field);
Sigma(1)=gfadd(gfconv(Adj(2,1),S(3),field),gfconv(Adj(2,2),S(4),fi
eld),field);
          Kappa(1,2)=Sigma(1);
          Kappa(1,3)=Sigma(2);
          end
      else
        kofaktor=kof(M,S,field);
        Adj=kofaktor';
        mydets=abs(determinan);
        for a=1:3
            for d=1:3
            Adj(a,d)=gfdeconv(Adj(a,d),mydets,field);
            end
```

```
        end
Sigma(3)=gfadd(gfadd(gfconv(Adj(1,1),S(4),field),gfconv(Adj(1,2),S
(5),field),field),gfconv(Adj(1,3),S(6),field),field);

Sigma(2)=gfadd(gfadd(gfconv(Adj(2,1),S(4),field),gfconv(Adj(2,2),S
(5),field),field),gfconv(Adj(2,3),S(6),field),field);

Sigma(1)=gfadd(gfadd(gfconv(Adj(3,1),S(4),field),gfconv(Adj(3,2),S
(5),field),field),gfconv(Adj(3,3),S(6),field),field);
    Kappa(1,2)=Sigma(1);
    Kappa(1,3)=Sigma(2);
    Kappa(1,4)=Sigma(3);
      end

elseif(t==4)
      %hitung nilai determinan 4x4
      determinan = deter(M,field);
      if(determinan==-inf)
         [n,m]=size(M);
          n1=n-1;
          m1=m-1;
          B=zeros(3,3);
          for i=1:n1
              for j=1:m1
                  B(i,j)=M(i,j);
              end
          end
          M=B;
          determinan=deter(M,field);
          if(determinan==-inf)
            [n,m]=size(M);
            n1=n-1;
            m1=m-1;
            B=zeros(2,2);
            for i=1:n1
                for j=1:m1
                  B(i,j)=M(i,j);
                  end
            end
            M=B;
          %Matrik 2x2
          determinan=deter(M,field);
          if(determinan==-inf)
            Sigma(1)=gfdeconv(S(2),S(1),field);
            Kappa(1,2)=Sigma(1);
          else
            kofaktor=kof(M,S,field);
            Adj=kofaktor';
            mydets=abs(determinan);
            Adj(1,1)=gfdeconv(Adj(1,1),mydets,field);
            Adj(2,2)=gfdeconv(Adj(2,2),mydets,field);
            Adj(1,2)=gfdeconv(Adj(1,2),mydets,field);
            Adj(2,1)=gfdeconv(Adj(2,1),mydets,field);
Sigma(2)=gfadd(gfconv(Adj(1,1),S(3),field),gfconv(Adj(1,2),S(4),fi
eld),field);
```

```
Sigma(1)=gfadd(gfconv(Adj(2,1),S(3),field),gfconv(Adj(2,2),S(4),fi
eld),field);
            Kappa(1,2)=Sigma(1);
            Kappa(1,3)=Sigma(2);
          end
      else
        kofaktor=kof(M,S,field);
        Adj=kofaktor';
        mydets=abs(determinan);
        for a=1:3
            for d=1:3
                Adj(a,d)=gfdeconv(Adj(a,d),mydets,field);
            end
        end
Sigma(3)=gfadd(gfadd(gfconv(Adj(1,1),S(4),field),gfconv(Adj(1,2),S
(5),field),field),gfconv(Adj(1,3),S(6),field),field);

Sigma(2)=gfadd(gfadd(gfconv(Adj(2,1),S(4),field),gfconv(Adj(2,2),S
(5),field),field),gfconv(Adj(2,3),S(6),field),field);

Sigma(1)=gfadd(gfadd(gfconv(Adj(3,1),S(4),field),gfconv(Adj(3,2),S
(5),field),field),gfconv(Adj(3,3),S(6),field),field);
    Kappa(1,2)=Sigma(1);
    Kappa(1,3)=Sigma(2);
    Kappa(1,4)=Sigma(3);
          end
    else
        kofaktor=kof(M,S,field);
        Adj=kofaktor';
        mydets=abs(determinan);
        for a=1:4
            for d=1:4
                Adj(a,d)=gfdeconv(Adj(a,d),mydets,field);
            end
        end
Sigma(4)=gfadd(gfadd(gfadd(gfconv(Adj(1,1),S(5),field),gfconv(Adj(
1,2),S(6),field),field),gfconv(Adj(1,3),S(7),field),field),gfconv(
Adj(1,4),S(8),field),field);

Sigma(3)=gfadd(gfadd(gfadd(gfconv(Adj(2,1),S(5),field),gfconv(Adj(
2,2),S(6),field),field),gfconv(Adj(2,3),S(7),field),field),gfconv(
Adj(2,4),S(8),field),field);

Sigma(2)=gfadd(gfadd(gfadd(gfconv(Adj(3,1),S(5),field),gfconv(Adj(
3,2),S(6),field),field),gfconv(Adj(3,3),S(7),field),field),gfconv(
Adj(3,4),S(8),field),field);

Sigma(1)=gfadd(gfadd(gfadd(gfconv(Adj(4,1),S(5),field),gfconv(Adj(
4,2),S(6),field),field),gfconv(Adj(4,3),S(7),field),field),gfconv(
Adj(4,4),S(8),field),field);
    Kappa(1,2)=Sigma(1);
    Kappa(1,3)=Sigma(2);
    Kappa(1,4)=Sigma(3);
    Kappa(1,5)=Sigma(4);
      end
```

```
elseif(t==5)
   %hitung nilai determinan 5x5
      determinan = deter(M,field);
      if(determinan==-inf) %Ubah jadi matriks 4x4
          [n,m]=size(M);
          n1=n-1;
          m1=m-1;
          B=zeros(4,4);
          for i=1:n1
              for j=1:m1
                  B(i,j)=M(i,j);
              end
          end
          M=B;
          determinan=deter(M,field);
          if(determinan==-inf)
          [n,m]=size(M);
          n1=n-1;
          m1=m-1;
          B=zeros(3,3);
          for i=1:n1
              for j=1:m1
                  B(i,j)=M(i,j);
              end
          end
          M=B;
          %Matrik 2x2
          determinan=deter(M,field);
          if(determinan==-inf)
          [n,m]=size(M);
          n1=n-1;
          m1=m-1;
          B=zeros(2,2);
          for i=1:n1
              for j=1:m1
                  B(i,j)=M(i,j);
              end
          end
          M=B;
          determinan=deter(M,field);
          if(determinan==-inf)
            Sigma(1)=gfdeconv(S(2),S(1),field);
            Kappa(1,2)=Sigma(1);
       else %terdapat 2 error
            kofaktor=kof(M,S,field);
            Adj=kofaktor';
            mydets=abs(determinan);
            Adj(1,1)=gfdeconv(Adj(1,1),mydets,field);
            Adj(2,2)=gfdeconv(Adj(2,2),mydets,field);
            Adj(1,2)=gfdeconv(Adj(1,2),mydets,field);
            Adj(2,1)=gfdeconv(Adj(2,1),mydets,field);
Sigma(2)=gfadd(gfconv(Adj(1,1),S(3),field),gfconv(Adj(1,2),S(4),fi
eld),field);
Sigma(1)=gfadd(gfconv(Adj(2,1),S(3),field),gfconv(Adj(2,2),S(4),fi
eld),field);
            Kappa(1,2)=Sigma(1);
```

```
            Kappa(1,3)=Sigma(2);
          end
   else %terdapat 3 error
        kofaktor=kof(M,S,field);
        Adj=kofaktor';
        mydets=abs(determinan);
        for a=1:3
            for d=1:3
            Adj(a,d)=gfdeconv(Adj(a,d),mydets,field);
            end
        end
Sigma(3)=gfadd(gfadd(gfconv(Adj(1,1),S(4),field),gfconv(Adj(1,2),S
(5),field),field),gfconv(Adj(1,3),S(6),field),field);

Sigma(2)=gfadd(gfadd(gfconv(Adj(2,1),S(4),field),gfconv(Adj(2,2),S
(5),field),field),gfconv(Adj(2,3),S(6),field),field);

Sigma(1)=gfadd(gfadd(gfconv(Adj(3,1),S(4),field),gfconv(Adj(3,2),S
(5),field),field),gfconv(Adj(3,3),S(6),field),field);
    Kappa(1,2)=Sigma(1);
    Kappa(1,3)=Sigma(2);
    Kappa(1,4)=Sigma(3);
          end
   else %terdapat 4 error
        kofaktor=kof(M,S,field);
        Adj=kofaktor';
        mydets=abs(determinan);
        for a=1:4
            for d=1:4
                Adj(a,d)=gfdeconv(Adj(a,d),mydets,field);
            end
        end
Sigma(4)=gfadd(gfadd(gfadd(gfconv(Adj(1,1),S(5),field),gfconv(Adj(
1,2),S(6),field),field),gfconv(Adj(1,3),S(7),field),field),gfconv(
Adj(1,4),S(8),field),field);

Sigma(3)=gfadd(gfadd(gfadd(gfconv(Adj(2,1),S(5),field),gfconv(Adj(
2,2),S(6),field),field),gfconv(Adj(2,3),S(7),field),field),gfconv(
Adj(2,4),S(8),field),field);

Sigma(2)=gfadd(gfadd(gfadd(gfconv(Adj(3,1),S(5),field),gfconv(Adj(
3,2),S(6),field),field),gfconv(Adj(3,3),S(7),field),field),gfconv(
Adj(3,4),S(8),field),field);

Sigma(1)=gfadd(gfadd(gfadd(gfconv(Adj(4,1),S(5),field),gfconv(Adj(
4,2),S(6),field),field),gfconv(Adj(4,3),S(7),field),field),gfconv(
Adj(4,4),S(8),field),field);
    Kappa(1,2)=Sigma(1);
    Kappa(1,3)=Sigma(2);
    Kappa(1,4)=Sigma(3);
    Kappa(1,5)=Sigma(4);
    end
        else %terdapat 5 error
        kofaktor=kof(M,S,field);
        Adj=kofaktor';
        mydets=abs(determinan);
```

```
        for a=1:5
            for d=1:5
                Adj(a,d)=gfdeconv(Adj(a,d),mydets,field);
            end
        end
Sigma(5)=gfadd(gfadd(gfadd(gfadd(gfconv(Adj(1,1),S(6),field),gfcon
v(Adj(1,2),S(7),field),field),gfconv(Adj(1,3),S(8),field),field),g
fconv(Adj(1,4),S(9),field),field),gfconv(Adj(1,5),S(10),field),fie
ld);

Sigma(4)=gfadd(gfadd(gfadd(gfadd(gfconv(Adj(2,1),S(6),field),gfcon
v(Adj(2,2),S(7),field),field),gfconv(Adj(2,3),S(8),field),field),g
fconv(Adj(2,4),S(9),field),field),gfconv(Adj(2,5),S(10),field),fie
ld);

Sigma(3)=gfadd(gfadd(gfadd(gfadd(gfconv(Adj(3,1),S(6),field),gfcon
v(Adj(3,2),S(7),field),field),gfconv(Adj(3,3),S(8),field),field),g
fconv(Adj(3,4),S(9),field),field),gfconv(Adj(3,5),S(10),field),fie
ld);

Sigma(2)=gfadd(gfadd(gfadd(gfadd(gfconv(Adj(4,1),S(6),field),gfcon
v(Adj(4,2),S(7),field),field),gfconv(Adj(4,3),S(8),field),field),g
fconv(Adj(4,4),S(9),field),field),gfconv(Adj(4,5),S(10),field),fie
ld);

Sigma(1)=gfadd(gfadd(gfadd(gfadd(gfconv(Adj(5,1),S(6),field),gfcon
v(Adj(5,2),S(7),field),field),gfconv(Adj(5,3),S(8),field),field),g
fconv(Adj(5,4),S(9),field),field),gfconv(Adj(5,5),S(10),field),fie
ld);
    Kappa(1,2)=Sigma(1);
    Kappa(1,3)=Sigma(2);
    Kappa(1,4)=Sigma(3);
    Kappa(1,5)=Sigma(4);
    Kappa(1,6)=Sigma(5);
    end

elseif(t==6)
      determinan = deter(M,field);
      if(determinan==-inf) %Ubah jadi matriks 5x5
          [n,m]=size(M);
          n1=n-1;
          m1=m-1;
          B=zeros(5,5);
          for i=1:n1
              for j=1:m1
                  B(i,j)=M(i,j);
              end
          end
          M=B;
          determinan=deter(M,field);
          if(determinan==-inf)
          [n,m]=size(M);
          n1=n-1;
          m1=m-1;
          B=zeros(4,4);
          for i=1:n1
```

```
            for j=1:m1
                B(i,j)=M(i,j);
            end
        end
        M=B;
        %Matrik 3x3
        determinan=deter(M,field);
        if(determinan==-inf)
        [n,m]=size(M);
        n1=n-1;
        m1=m-1;
        B=zeros(3,3);
        for i=1:n1
            for j=1:m1
                B(i,j)=M(i,j);
            end
        end

        if(determinan==-inf)
        [n,m]=size(M);
        n1=n-1;
        m1=m-1;
        B=zeros(2,2);
        for i=1:n1
            for j=1:m1
                B(i,j)=M(i,j);
            end
        end
        M=B;
        determinan=deter(M,field);
        if(determinan==-inf) %hanya terdapat 1 error
        Sigma(1)=gfdeconv(S(2),S(1),field);
        Kappa(1,2)=Sigma(1);\
    else %terdapat 2 error
        kofaktor=kof(M,S,field);
        Adj=kofaktor';
        mydets=abs(determinan);
          Adj(1,1)=gfdeconv(Adj(1,1),mydets,field);
          Adj(2,2)=gfdeconv(Adj(2,2),mydets,field);
          Adj(1,2)=gfdeconv(Adj(1,2),mydets,field);
          Adj(2,1)=gfdeconv(Adj(2,1),mydets,field);

Sigma(2)=gfadd(gfconv(Adj(1,1),S(3),field),gfconv(Adj(1,2),S(4),fi
eld),field);
Sigma(1)=gfadd(gfconv(Adj(2,1),S(3),field),gfconv(Adj(2,2),S(4),fi
eld),field);
          Kappa(1,2)=Sigma(1);
          Kappa(1,3)=Sigma(2);
        end
   else %terdapat 3 error
       kofaktor=kof(M,S,field);
       Adj=kofaktor';
       mydets=abs(determinan);
       for a=1:3
           for d=1:3
               Adj(a,d)=gfdeconv(Adj(a,d),mydets,field);
```

```
            end
        end
Sigma(3)=gfadd(gfadd(gfconv(Adj(1,1),S(4),field),gfconv(Adj(1,2),S
(5),field),field),gfconv(Adj(1,3),S(6),field),field);

Sigma(2)=gfadd(gfadd(gfconv(Adj(2,1),S(4),field),gfconv(Adj(2,2),S
(5),field),field),gfconv(Adj(2,3),S(6),field),field);

Sigma(1)=gfadd(gfadd(gfconv(Adj(3,1),S(4),field),gfconv(Adj(3,2),S
(5),field),field),gfconv(Adj(3,3),S(6),field),field);
        Kappa(1,2)=Sigma(1);
        Kappa(1,3)=Sigma(2);
        Kappa(1,4)=Sigma(3);
          end
   else %terdapat 4 error
        kofaktor=kof(M,S,field);
        Adj=kofaktor';
        mydets=abs(determinan);
        for a=1:4
            for d=1:4
                Adj(a,d)=gfdeconv(Adj(a,d),mydets,field);
            end
        end
Sigma(4)=gfadd(gfadd(gfadd(gfconv(Adj(1,1),S(5),field),gfconv(Adj(
1,2),S(6),field),field),gfconv(Adj(1,3),S(7),field),field),gfconv(
Adj(1,4),S(8),field),field);

Sigma(3)=gfadd(gfadd(gfadd(gfconv(Adj(2,1),S(5),field),gfconv(Adj(
2,2),S(6),field),field),gfconv(Adj(2,3),S(7),field),field),gfconv(
Adj(2,4),S(8),field),field);

Sigma(2)=gfadd(gfadd(gfadd(gfconv(Adj(3,1),S(5),field),gfconv(Adj(
3,2),S(6),field),field),gfconv(Adj(3,3),S(7),field),field),gfconv(
Adj(3,4),S(8),field),field);

Sigma(1)=gfadd(gfadd(gfadd(gfconv(Adj(4,1),S(5),field),gfconv(Adj(
4,2),S(6),field),field),gfconv(Adj(4,3),S(7),field),field),gfconv(
Adj(4,4),S(8),field),field);
        Kappa(1,2)=Sigma(1);
        Kappa(1,3)=Sigma(2);
        Kappa(1,4)=Sigma(3);
        Kappa(1,5)=Sigma(4);
          end
   else %terdapat 5 error
        kofaktor=kof(M,S,field);
        Adj=kofaktor';
        mydets=abs(determinan);
        for a=1:5
            for d=1:5
                Adj(a,d)=gfdeconv(Adj(a,d),mydets,field);
            end
        end
Sigma(5)=gfadd(gfadd(gfadd(gfadd(gfconv(Adj(1,1),S(6),field),gfcon
v(Adj(1,2),S(7),field),field),gfconv(Adj(1,3),S(8),field),field),g
fconv(Adj(1,4),S(9),field),field),gfconv(Adj(1,5),S(10),field),fie
ld);
```

```
Sigma(4)=gfadd(gfadd(gfadd(gfadd(gfconv(Adj(2,1),S(6),field),gfcon
v(Adj(2,2),S(7),field),field),gfconv(Adj(2,3),S(8),field),field),g
fconv(Adj(2,4),S(9),field),field),gfconv(Adj(2,5),S(10),field),fie
ld);

Sigma(3)=gfadd(gfadd(gfadd(gfadd(gfconv(Adj(3,1),S(6),field),gfcon
v(Adj(3,2),S(7),field),field),gfconv(Adj(3,3),S(8),field),field),g
fconv(Adj(3,4),S(9),field),field),gfconv(Adj(3,5),S(10),field),fie
ld);

Sigma(2)=gfadd(gfadd(gfadd(gfadd(gfconv(Adj(4,1),S(6),field),gfcon
v(Adj(4,2),S(7),field),field),gfconv(Adj(4,3),S(8),field),field),g
fconv(Adj(4,4),S(9),field),field),gfconv(Adj(4,5),S(10),field),fie
ld);

Sigma(1)=gfadd(gfadd(gfadd(gfadd(gfconv(Adj(5,1),S(6),field),gfcon
v(Adj(5,2),S(7),field),field),gfconv(Adj(5,3),S(8),field),field),g
fconv(Adj(5,4),S(9),field),field),gfconv(Adj(5,5),S(10),field),fie
ld);
        Kappa(1,2)=Sigma(1);
        Kappa(1,3)=Sigma(2);
        Kappa(1,4)=Sigma(3);
        Kappa(1,5)=Sigma(4);
        Kappa(1,6)=Sigma(5);
          end
 else %terdapat 6 error
      kofaktor=kof(M,S,field);
      Adj=kofaktor';
      mydets=abs(determinan);
        for a=1:6
            for d=1:6
                Adj(a,d)=gfdeconv(Adj(a,d),mydets,field);
            end
        end
Sigma(6)=gfadd(gfadd(gfadd(gfadd(gfadd(gfconv(Adj(1,1),S(7),field)
,gfconv(Adj(1,2),S(8),field),field),gfconv(Adj(1,3),S(9),field),fi
eld),gfconv(Adj(1,4),S(10),field),field),gfconv(Adj(1,5),S(11),fie
ld),field),gfconv(Adj(1,6),S(12),field),field);

Sigma(5)=gfadd(gfadd(gfadd(gfadd(gfadd(gfconv(Adj(2,1),S(7),field)
,gfconv(Adj(2,2),S(8),field),field),gfconv(Adj(2,3),S(9),field),fi
eld),gfconv(Adj(2,4),S(10),field),field),gfconv(Adj(2,5),S(11),fie
ld),field),gfconv(Adj(2,6),S(12),field),field);

Sigma(4)=gfadd(gfadd(gfadd(gfadd(gfadd(gfconv(Adj(3,1),S(7),field)
,gfconv(Adj(3,2),S(8),field),field),gfconv(Adj(3,3),S(9),field),fi
eld),gfconv(Adj(3,4),S(10),field),field),gfconv(Adj(3,5),S(11),fie
ld),field),gfconv(Adj(3,6),S(12),field),field);

Sigma(3)=gfadd(gfadd(gfadd(gfadd(gfadd(gfconv(Adj(4,1),S(7),field)
,gfconv(Adj(4,2),S(8),field),field),gfconv(Adj(4,3),S(9),field),fi
eld),gfconv(Adj(4,4),S(10),field),field),gfconv(Adj(4,5),S(11),fie
ld),field),gfconv(Adj(4,6),S(12),field),field);

Sigma(2)=gfadd(gfadd(gfadd(gfadd(gfadd(gfconv(Adj(5,1),S(7),field)
```

```
,gfconv(Adj(5,2),S(8),field),field),gfconv(Adj(5,3),S(9),field),fi
eld),gfconv(Adj(5,4),S(10),field),field),gfconv(Adj(5,5),S(11),fie
ld),field),gfconv(Adj(5,6),S(12),field),field);

Sigma(1)=gfadd(gfadd(gfadd(gfadd(gfadd(gfconv(Adj(6,1),S(7),field)
,gfconv(Adj(6,2),S(8),field),field),gfconv(Adj(6,3),S(9),field),fi
eld),gfconv(Adj(6,4),S(10),field),field),gfconv(Adj(6,5),S(11),fie
ld),field),gfconv(Adj(6,6),S(12),field),field);
        Kappa(1,2)=Sigma(1);
        Kappa(1,3)=Sigma(2);
        Kappa(1,4)=Sigma(3);
        Kappa(1,5)=Sigma(4);
        Kappa(1,6)=Sigma(5);
        Kappa(1,7)=Sigma(6);
      end

elseif(t==7)
      determinan = deter(M,field);
      if(determinan==-inf) %Ubah jadi matriks 6x6
          [n,m]=size(M);
          n1=n-1;
          m1=m-1;
          B=zeros(6,6);
          for i=1:n1
              for j=1:m1
                  B(i,j)=M(i,j);
              end
          end
          M=B;
          determinan=deter(M,field);
          if(determinan==-inf)
          [n,m]=size(M);
          n1=n-1;
          m1=m-1;
          B=zeros(5,5);
          for i=1:n1
              for j=1:m1
                  B(i,j)=M(i,j);
              end
          end
          M=B;
          %Matrik 4x4
          determinan=deter(M,field);
          if(determinan==-inf)
          [n,m]=size(M);
          n1=n-1;
          m1=m-1;
          B=zeros(4,4);
          for i=1:n1
              for j=1:m1
                  B(i,j)=M(i,j);
              end
          end
          M=B;
          determinan=deter(M,field);
          if(determinan==-inf) %ubah jadi matrik 3x3
```

```
            [n,m]=size(M);
            n1=n-1;
            m1=m-1;
            B=zeros(3,3);
            for i=1:n1
                for j=1:m1
                    B(i,j)=M(i,j);
                end
            end
            M=B;
            determinan=deter(M,field);
            if(determinan==-inf) %ubah jadi matrik 2x2
            [n,m]=size(M);
            n1=n-1;
            m1=m-1;
            B=zeros(2,2);
            for i=1:n1
                for j=1:m1
                    B(i,j)=M(i,j);
                end
            end
          M=B;
          determinan=deter(M,field);
       if(determinan==-inf) %hanya terdapat 1 error
          Sigma(1)=gfdeconv(S(2),S(1),field);%Hanya 1 error yg
terjadi
          Kappa(1,2)=Sigma(1);
       else %terdapat 2 error
            kofaktor=kof(M,S,field);
            Adj=kofaktor';
            mydets=abs(determinan);
              Adj(1,1)=gfdeconv(Adj(1,1),mydets,field);
              Adj(2,2)=gfdeconv(Adj(2,2),mydets,field);
              Adj(1,2)=gfdeconv(Adj(1,2),mydets,field);
              Adj(2,1)=gfdeconv(Adj(2,1),mydets,field);
Sigma(2)=gfadd(gfconv(Adj(1,1),S(3),field),gfconv(Adj(1,2),S(4),fi
eld),field);
Sigma(1)=gfadd(gfconv(Adj(2,1),S(3),field),gfconv(Adj(2,2),S(4),fi
eld),field);
            Kappa(1,2)=Sigma(1);
            Kappa(1,3)=Sigma(2);
            end
     else %terdapat 3 error
          kofaktor=kof(M,S,field);
          Adj=kofaktor';
          mydets=abs(determinan);
          for a=1:3
              for d=1:3
                  Adj(a,d)=gfdeconv(Adj(a,d),mydets,field);
              end
          end
Sigma(3)=gfadd(gfadd(gfconv(Adj(1,1),S(4),field),gfconv(Adj(1,2),S
(5),field),field),gfconv(Adj(1,3),S(6),field),field);

Sigma(2)=gfadd(gfadd(gfconv(Adj(2,1),S(4),field),gfconv(Adj(2,2),S
(5),field),field),gfconv(Adj(2,3),S(6),field),field);
```

```
Sigma(1)=gfadd(gfadd(gfconv(Adj(3,1),S(4),field),gfconv(Adj(3,2),S
(5),field),field),gfconv(Adj(3,3),S(6),field),field);
        Kappa(1,2)=Sigma(1);
        Kappa(1,3)=Sigma(2);
        Kappa(1,4)=Sigma(3);
          end
   else %terdapat 4 error
        kofaktor=kof(M,S,field);
        Adj=kofaktor';
        mydets=abs(determinan);
        for a=1:4
            for d=1:4
                Adj(a,d)=gfdeconv(Adj(a,d),mydets,field);
            end
        end
Sigma(4)=gfadd(gfadd(gfadd(gfconv(Adj(1,1),S(5),field),gfconv(Adj(
1,2),S(6),field),field),gfconv(Adj(1,3),S(7),field),field),gfconv(
Adj(1,4),S(8),field),field);

Sigma(3)=gfadd(gfadd(gfadd(gfconv(Adj(2,1),S(5),field),gfconv(Adj(
2,2),S(6),field),field),gfconv(Adj(2,3),S(7),field),field),gfconv(
Adj(2,4),S(8),field),field);

Sigma(2)=gfadd(gfadd(gfadd(gfconv(Adj(3,1),S(5),field),gfconv(Adj(
3,2),S(6),field),field),gfconv(Adj(3,3),S(7),field),field),gfconv(
Adj(3,4),S(8),field),field);

Sigma(1)=gfadd(gfadd(gfadd(gfconv(Adj(4,1),S(5),field),gfconv(Adj(
4,2),S(6),field),field),gfconv(Adj(4,3),S(7),field),field),gfconv(
Adj(4,4),S(8),field),field);
        Kappa(1,2)=Sigma(1);
        Kappa(1,3)=Sigma(2);
        Kappa(1,4)=Sigma(3);
        Kappa(1,5)=Sigma(4);
          end
   else %terdapat 5 error
        kofaktor=kof(M,S,field);
        Adj=kofaktor';
        mydets=abs(determinan);
        for a=1:5
            for d=1:5
                Adj(a,d)=gfdeconv(Adj(a,d),mydets,field);
            end
        end
Sigma(5)=gfadd(gfadd(gfadd(gfadd(gfconv(Adj(1,1),S(6),field),gfcon
v(Adj(1,2),S(7),field),field),gfconv(Adj(1,3),S(8),field),field),g
fconv(Adj(1,4),S(9),field),field),gfconv(Adj(1,5),S(10),field),fie
ld);

Sigma(4)=gfadd(gfadd(gfadd(gfadd(gfconv(Adj(2,1),S(6),field),gfcon
v(Adj(2,2),S(7),field),field),gfconv(Adj(2,3),S(8),field),field),g
fconv(Adj(2,4),S(9),field),field),gfconv(Adj(2,5),S(10),field),fie
ld);

Sigma(3)=gfadd(gfadd(gfadd(gfadd(gfconv(Adj(3,1),S(6),field),gfcon
```

```
v(Adj(3,2),S(7),field),field),gfconv(Adj(3,3),S(8),field),field),g
fconv(Adj(3,4),S(9),field),field),gfconv(Adj(3,5),S(10),field),fie
ld);

Sigma(2)=gfadd(gfadd(gfadd(gfadd(gfconv(Adj(4,1),S(6),field),gfcon
v(Adj(4,2),S(7),field),field),gfconv(Adj(4,3),S(8),field),field),g
fconv(Adj(4,4),S(9),field),field),gfconv(Adj(4,5),S(10),field),fie
ld);

Sigma(1)=gfadd(gfadd(gfadd(gfadd(gfconv(Adj(5,1),S(6),field),gfcon
v(Adj(5,2),S(7),field),field),gfconv(Adj(5,3),S(8),field),field),g
fconv(Adj(5,4),S(9),field),field),gfconv(Adj(5,5),S(10),field),fie
ld);
        Kappa(1,2)=Sigma(1);
        Kappa(1,3)=Sigma(2);
        Kappa(1,4)=Sigma(3);
        Kappa(1,5)=Sigma(4);
        Kappa(1,6)=Sigma(5);
          end

    else %terdapat 6 error
        kofaktor=kof(M,S,field);
        Adj=kofaktor';
        mydets=abs(determinan);
        for a=1:6
            for d=1:6
                Adj(a,d)=gfdeconv(Adj(a,d),mydets,field);
            end
        end
Sigma(6)=gfadd(gfadd(gfadd(gfadd(gfadd(gfconv(Adj(1,1),S(7),field)
,gfconv(Adj(1,2),S(8),field),field),gfconv(Adj(1,3),S(9),field),fi
eld),gfconv(Adj(1,4),S(10),field),field),gfconv(Adj(1,5),S(11),fie
ld),field),gfconv(Adj(1,6),S(12),field),field);

Sigma(5)=gfadd(gfadd(gfadd(gfadd(gfadd(gfconv(Adj(2,1),S(7),field)
,gfconv(Adj(2,2),S(8),field),field),gfconv(Adj(2,3),S(9),field),fi
eld),gfconv(Adj(2,4),S(10),field),field),gfconv(Adj(2,5),S(11),fie
ld),field),gfconv(Adj(2,6),S(12),field),field);

Sigma(4)=gfadd(gfadd(gfadd(gfadd(gfadd(gfconv(Adj(3,1),S(7),field)
,gfconv(Adj(3,2),S(8),field),field),gfconv(Adj(3,3),S(9),field),fi
eld),gfconv(Adj(3,4),S(10),field),field),gfconv(Adj(3,5),S(11),fie
ld),field),gfconv(Adj(3,6),S(12),field),field);

Sigma(3)=gfadd(gfadd(gfadd(gfadd(gfadd(gfconv(Adj(4,1),S(7),field)
,gfconv(Adj(4,2),S(8),field),field),gfconv(Adj(4,3),S(9),field),fi
eld),gfconv(Adj(4,4),S(10),field),field),gfconv(Adj(4,5),S(11),fie
ld),field),gfconv(Adj(4,6),S(12),field),field);

Sigma(2)=gfadd(gfadd(gfadd(gfadd(gfadd(gfconv(Adj(5,1),S(7),field)
,gfconv(Adj(5,2),S(8),field),field),gfconv(Adj(5,3),S(9),field),fi
eld),gfconv(Adj(5,4),S(10),field),field),gfconv(Adj(5,5),S(11),fie
ld),field),gfconv(Adj(5,6),S(12),field),field);

Sigma(1)=gfadd(gfadd(gfadd(gfadd(gfadd(gfconv(Adj(6,1),S(7),field)
,gfconv(Adj(6,2),S(8),field),field),gfconv(Adj(6,3),S(9),field),fi
```

```
eld),gfconv(Adj(6,4),S(10),field),field),gfconv(Adj(6,5),S(11),fie
ld),field),gfconv(Adj(6,6),S(12),field),field);
        Kappa(1,2)=Sigma(1);
        Kappa(1,3)=Sigma(2);
        Kappa(1,4)=Sigma(3);
        Kappa(1,5)=Sigma(4);
        Kappa(1,6)=Sigma(5);
        Kappa(1,7)=Sigma(6);
          end
   else %terdapat 7 error
        kofaktor=kof(M,S,field);
        Adj=kofaktor';
        mydets=abs(determinan);
        for a=1:7
            for d=1:7
                Adj(a,d)=gfdeconv(Adj(a,d),mydets,field);
            end
        end
Sigma(7)=gfadd(gfadd(gfadd(gfadd(gfadd(gfadd(gfconv(Adj(1,1),S(8),
field),gfconv(Adj(1,2),S(9),field),field),gfconv(Adj(1,3),S(10),fi
eld),field),gfconv(Adj(1,4),S(11),field),field),gfconv(Adj(1,5),S(
12),field),field),gfconv(Adj(1,6),S(13),field),field),gfconv(Adj(1
,7),S(14),field),field);

Sigma(6)=gfadd(gfadd(gfadd(gfadd(gfadd(gfadd(gfconv(Adj(2,1),S(8),
field),gfconv(Adj(2,2),S(9),field),field),gfconv(Adj(2,3),S(10),fi
eld),field),gfconv(Adj(2,4),S(11),field),field),gfconv(Adj(2,5),S(
12),field),field),gfconv(Adj(2,6),S(13),field),field),gfconv(Adj(2
,7),S(14),field),field);

Sigma(5)=gfadd(gfadd(gfadd(gfadd(gfadd(gfadd(gfconv(Adj(3,1),S(8),
field),gfconv(Adj(3,2),S(9),field),field),gfconv(Adj(3,3),S(10),fi
eld),field),gfconv(Adj(3,4),S(11),field),field),gfconv(Adj(3,5),S(
12),field),field),gfconv(Adj(3,6),S(13),field),field),gfconv(Adj(3
,7),S(14),field),field);

Sigma(4)=gfadd(gfadd(gfadd(gfadd(gfadd(gfadd(gfconv(Adj(4,1),S(8),
field),gfconv(Adj(4,2),S(9),field),field),gfconv(Adj(4,3),S(10),fi
eld),field),gfconv(Adj(4,4),S(11),field),field),gfconv(Adj(4,5),S(
12),field),field),gfconv(Adj(4,6),S(13),field),field),gfconv(Adj(4
,7),S(14),field),field);

Sigma(3)=gfadd(gfadd(gfadd(gfadd(gfadd(gfadd(gfconv(Adj(5,1),S(8),
field),gfconv(Adj(5,2),S(9),field),field),gfconv(Adj(5,3),S(10),fi
eld),field),gfconv(Adj(5,4),S(11),field),field),gfconv(Adj(5,5),S(
12),field),field),gfconv(Adj(5,6),S(13),field),field),gfconv(Adj(5
,7),S(14),field),field);

Sigma(2)=gfadd(gfadd(gfadd(gfadd(gfadd(gfadd(gfconv(Adj(6,1),S(8),
field),gfconv(Adj(6,2),S(9),field),field),gfconv(Adj(6,3),S(10),fi
eld),field),gfconv(Adj(6,4),S(11),field),field),gfconv(Adj(6,5),S(
12),field),field),gfconv(Adj(6,6),S(13),field),field),gfconv(Adj(6
,7),S(14),field),field);
Sigma(1)=gfadd(gfadd(gfadd(gfadd(gfadd(gfadd(gfconv(Adj(7,1),S(8),
field),gfconv(Adj(7,2),S(9),field),field),gfconv(Adj(7,3),S(10),fi
eld),field),gfconv(Adj(7,4),S(11),field),field),gfconv(Adj(7,5),S(
```

```
12),field),field),gfconv(Adj(7,6),S(13),field),field),gfconv(Adj(7
,7),S(14),field),field);
        Kappa(1,2)=Sigma(1);
        Kappa(1,3)=Sigma(2);
        Kappa(1,4)=Sigma(3);
        Kappa(1,5)=Sigma(4);
        Kappa(1,6)=Sigma(5);
        Kappa(1,7)=Sigma(6);
        Kappa(1,8)=Sigma(7);
      end

elseif(t==8)
    %hitung nilai determinan 8x8
      determinan = deter(M,field);
      if(determinan==-inf) %Ubah jadi matriks 7x7
          [n,m]=size(M);
          n1=n-1;
          m1=m-1;
          B=zeros(7,7);
          for i=1:n1
              for j=1:m1
                  B(i,j)=M(i,j);
              end
          end
          M=B;
          determinan=deter(M,field);
          if(determinan==-inf)
          [n,m]=size(M);
          n1=n-1;
          m1=m-1;
          B=zeros(6,6);
          for i=1:n1
              for j=1:m1
                  B(i,j)=M(i,j);
              end
          end
          M=B;
          %Matrik 5x5
          determinan=deter(M,field);
          if(determinan==-inf)
          [n,m]=size(M);
          n1=n-1;
          m1=m-1;
          B=zeros(5,5);
          for i=1:n1
              for j=1:m1
                  B(i,j)=M(i,j);
              end
          end
          M=B;
          determinan=deter(M,field);
          if(determinan==-inf) %ubah jadi matrik 4x4
          [n,m]=size(M);
          n1=n-1;
          m1=m-1;
          B=zeros(4,4);
```

```
            for i=1:n1
                for j=1:m1
                    B(i,j)=M(i,j);
                end
            end
            M=B;
            determinan=deter(M,field);
            if(determinan==-inf) %ubah jadi matrik 3x3
            [n,m]=size(M);
            n1=n-1;
            m1=m-1;
            B=zeros(3,3);
            for i=1:n1
                for j=1:m1
                    B(i,j)=M(i,j);
                end
            end
            M=B;
            determinan=deter(M,field);
            if(determinan==-inf) %ubah jadi matrik 2x2
            [n,m]=size(M);
            n1=n-1;
            m1=m-1;
            B=zeros(2,2);
            for i=1:n1
                for j=1:m1
                    B(i,j)=M(i,j);
                end
            end
            M=B;
            determinan=deter(M,field);
     if(determinan==-inf) %hanya terdapat 1 error
        Sigma(1)=gfdeconv(S(2),S(1),field);%Hanya 1 error yg
terjadi
        Kappa(1,2)=Sigma(1);

     else %terdapat 2 error
          kofaktor=kof(M,S,field);
          Adj=kofaktor';
          mydets=abs(determinan);
          Adj(1,1)=gfdeconv(Adj(1,1),mydets,field);
          Adj(2,2)=gfdeconv(Adj(2,2),mydets,field);
          Adj(1,2)=gfdeconv(Adj(1,2),mydets,field);
          Adj(2,1)=gfdeconv(Adj(2,1),mydets,field);
Sigma(2)=gfadd(gfconv(Adj(1,1),S(3),field),gfconv(Adj(1,2),S(4),fi
eld),field);
Sigma(1)=gfadd(gfconv(Adj(2,1),S(3),field),gfconv(Adj(2,2),S(4),fi
eld),field);
          Kappa(1,2)=Sigma(1);
          Kappa(1,3)=Sigma(2);
          end
   else %terdapat 3 error
        kofaktor=kof(M,S,field);
        Adj=kofaktor';
        mydets=abs(determinan);
        for a=1:3
```

```
            for d=1:3
                Adj(a,d)=gfdeconv(Adj(a,d),mydets,field);
            end
        end
Sigma(3)=gfadd(gfadd(gfconv(Adj(1,1),S(4),field),gfconv(Adj(1,2),S
(5),field),field),gfconv(Adj(1,3),S(6),field),field);

Sigma(2)=gfadd(gfadd(gfconv(Adj(2,1),S(4),field),gfconv(Adj(2,2),S
(5),field),field),gfconv(Adj(2,3),S(6),field),field);

Sigma(1)=gfadd(gfadd(gfconv(Adj(3,1),S(4),field),gfconv(Adj(3,2),S
(5),field),field),gfconv(Adj(3,3),S(6),field),field);
        Kappa(1,2)=Sigma(1);
        Kappa(1,3)=Sigma(2);
        Kappa(1,4)=Sigma(3);
        end

   else %terdapat 4 error
        kofaktor=kof(M,S,field);
        Adj=kofaktor';
        mydets=abs(determinan);
        for a=1:4
            for d=1:4
                Adj(a,d)=gfdeconv(Adj(a,d),mydets,field);
            end
        end
Sigma(4)=gfadd(gfadd(gfadd(gfconv(Adj(1,1),S(5),field),gfconv(Adj(
1,2),S(6),field),field),gfconv(Adj(1,3),S(7),field),field),gfconv(
Adj(1,4),S(8),field),field);

Sigma(3)=gfadd(gfadd(gfadd(gfconv(Adj(2,1),S(5),field),gfconv(Adj(
2,2),S(6),field),field),gfconv(Adj(2,3),S(7),field),field),gfconv(
Adj(2,4),S(8),field),field);

Sigma(2)=gfadd(gfadd(gfadd(gfconv(Adj(3,1),S(5),field),gfconv(Adj(
3,2),S(6),field),field),gfconv(Adj(3,3),S(7),field),field),gfconv(
Adj(3,4),S(8),field),field);

Sigma(1)=gfadd(gfadd(gfadd(gfconv(Adj(4,1),S(5),field),gfconv(Adj(
4,2),S(6),field),field),gfconv(Adj(4,3),S(7),field),field),gfconv(
Adj(4,4),S(8),field),field);
        Kappa(1,2)=Sigma(1);
        Kappa(1,3)=Sigma(2);
        Kappa(1,4)=Sigma(3);
        Kappa(1,5)=Sigma(4);
          end
   else %terdapat 5 error
        kofaktor=kof(M,S,field);
        Adj=kofaktor';
        mydets=abs(determinan);
        for a=1:5
            for d=1:5
                Adj(a,d)=gfdeconv(Adj(a,d),mydets,field);
            end
        end
Sigma(5)=gfadd(gfadd(gfadd(gfadd(gfconv(Adj(1,1),S(6),field),gfcon
```

```
v(Adj(1,2),S(7),field),field),gfconv(Adj(1,3),S(8),field),field),g
fconv(Adj(1,4),S(9),field),field),gfconv(Adj(1,5),S(10),field),fie
ld);

Sigma(4)=gfadd(gfadd(gfadd(gfadd(gfconv(Adj(2,1),S(6),field),gfcon
v(Adj(2,2),S(7),field),field),gfconv(Adj(2,3),S(8),field),field),g
fconv(Adj(2,4),S(9),field),field),gfconv(Adj(2,5),S(10),field),fie
ld);

Sigma(3)=gfadd(gfadd(gfadd(gfadd(gfconv(Adj(3,1),S(6),field),gfcon
v(Adj(3,2),S(7),field),field),gfconv(Adj(3,3),S(8),field),field),g
fconv(Adj(3,4),S(9),field),field),gfconv(Adj(3,5),S(10),field),fie
ld);

Sigma(2)=gfadd(gfadd(gfadd(gfadd(gfconv(Adj(4,1),S(6),field),gfcon
v(Adj(4,2),S(7),field),field),gfconv(Adj(4,3),S(8),field),field),g
fconv(Adj(4,4),S(9),field),field),gfconv(Adj(4,5),S(10),field),fie
ld);

Sigma(1)=gfadd(gfadd(gfadd(gfadd(gfconv(Adj(5,1),S(6),field),gfcon
v(Adj(5,2),S(7),field),field),gfconv(Adj(5,3),S(8),field),field),g
fconv(Adj(5,4),S(9),field),field),gfconv(Adj(5,5),S(10),field),fie
ld);
        Kappa(1,2)=Sigma(1);
        Kappa(1,3)=Sigma(2);
        Kappa(1,4)=Sigma(3);
        Kappa(1,5)=Sigma(4);
        Kappa(1,6)=Sigma(5);
          end
 else %terdapat 6 error
      kofaktor=kof(M,S,field);
        Adj=kofaktor';
        mydets=abs(determinan);
        for a=1:6
            for d=1:6
            Adj(a,d)=gfdeconv(Adj(a,d),mydets,field);
            end
        end
Sigma(6)=gfadd(gfadd(gfadd(gfadd(gfadd(gfconv(Adj(1,1),S(7),field)
,gfconv(Adj(1,2),S(8),field),field),gfconv(Adj(1,3),S(9),field),fi
eld),gfconv(Adj(1,4),S(10),field),field),gfconv(Adj(1,5),S(11),fie
ld),field),gfconv(Adj(1,6),S(12),field),field);

Sigma(5)=gfadd(gfadd(gfadd(gfadd(gfadd(gfconv(Adj(2,1),S(7),field)
,gfconv(Adj(2,2),S(8),field),field),gfconv(Adj(2,3),S(9),field),fi
eld),gfconv(Adj(2,4),S(10),field),field),gfconv(Adj(2,5),S(11),fie
ld),field),gfconv(Adj(2,6),S(12),field),field);

Sigma(4)=gfadd(gfadd(gfadd(gfadd(gfadd(gfconv(Adj(3,1),S(7),field)
,gfconv(Adj(3,2),S(8),field),field),gfconv(Adj(3,3),S(9),field),fi
eld),gfconv(Adj(3,4),S(10),field),field),gfconv(Adj(3,5),S(11),fie
ld),field),gfconv(Adj(3,6),S(12),field),field);

Sigma(3)=gfadd(gfadd(gfadd(gfadd(gfadd(gfconv(Adj(4,1),S(7),field)
,gfconv(Adj(4,2),S(8),field),field),gfconv(Adj(4,3),S(9),field),fi
```

```
eld),gfconv(Adj(4,4),S(10),field),field),gfconv(Adj(4,5),S(11),fie
ld),field),gfconv(Adj(4,6),S(12),field),field);

Sigma(2)=gfadd(gfadd(gfadd(gfadd(gfadd(gfconv(Adj(5,1),S(7),field)
,gfconv(Adj(5,2),S(8),field),field),gfconv(Adj(5,3),S(9),field),fi
eld),gfconv(Adj(5,4),S(10),field),field),gfconv(Adj(5,5),S(11),fie
ld),field),gfconv(Adj(5,6),S(12),field),field);

Sigma(1)=gfadd(gfadd(gfadd(gfadd(gfadd(gfconv(Adj(6,1),S(7),field)
,gfconv(Adj(6,2),S(8),field),field),gfconv(Adj(6,3),S(9),field),fi
eld),gfconv(Adj(6,4),S(10),field),field),gfconv(Adj(6,5),S(11),fie
ld),field),gfconv(Adj(6,6),S(12),field),field);
        Kappa(1,2)=Sigma(1);
        Kappa(1,3)=Sigma(2);
        Kappa(1,4)=Sigma(3);
        Kappa(1,5)=Sigma(4);
        Kappa(1,6)=Sigma(5);
        Kappa(1,7)=Sigma(6);
          end

   else %terdapat 7 error
        kofaktor=kof(M,S,field);
        Adj=kofaktor';
        mydets=abs(determinan);
        for a=1:7
            for d=1:7
            Adj(a,d)=gfdeconv(Adj(a,d),mydets,field);
            end
        end
Sigma(7)=gfadd(gfadd(gfadd(gfadd(gfadd(gfadd(gfconv(Adj(1,1),S(8),
field),gfconv(Adj(1,2),S(9),field),field),gfconv(Adj(1,3),S(10),fi
eld),field),gfconv(Adj(1,4),S(11),field),field),gfconv(Adj(1,5),S(
12),field),field),gfconv(Adj(1,6),S(13),field),field),gfconv(Adj(1
,7),S(14),field),field);

Sigma(6)=gfadd(gfadd(gfadd(gfadd(gfadd(gfadd(gfconv(Adj(2,1),S(8),
field),gfconv(Adj(2,2),S(9),field),field),gfconv(Adj(2,3),S(10),fi
eld),field),gfconv(Adj(2,4),S(11),field),field),gfconv(Adj(2,5),S(
12),field),field),gfconv(Adj(2,6),S(13),field),field),gfconv(Adj(2
,7),S(14),field),field);

Sigma(5)=gfadd(gfadd(gfadd(gfadd(gfadd(gfadd(gfconv(Adj(3,1),S(8),
field),gfconv(Adj(3,2),S(9),field),field),gfconv(Adj(3,3),S(10),fi
eld),field),gfconv(Adj(3,4),S(11),field),field),gfconv(Adj(3,5),S(
12),field),field),gfconv(Adj(3,6),S(13),field),field),gfconv(Adj(3
,7),S(14),field),field);

Sigma(4)=gfadd(gfadd(gfadd(gfadd(gfadd(gfadd(gfconv(Adj(4,1),S(8),
field),gfconv(Adj(4,2),S(9),field),field),gfconv(Adj(4,3),S(10),fi
eld),field),gfconv(Adj(4,4),S(11),field),field),gfconv(Adj(4,5),S(
12),field),field),gfconv(Adj(4,6),S(13),field),field),gfconv(Adj(4
,7),S(14),field),field);

Sigma(3)=gfadd(gfadd(gfadd(gfadd(gfadd(gfadd(gfconv(Adj(5,1),S(8),
field),gfconv(Adj(5,2),S(9),field),field),gfconv(Adj(5,3),S(10),fi
eld),field),gfconv(Adj(5,4),S(11),field),field),gfconv(Adj(5,5),S(
```

```
12),field),field),gfconv(Adj(5,6),S(13),field),field),gfconv(Adj(5
,7),S(14),field),field);

Sigma(2)=gfadd(gfadd(gfadd(gfadd(gfadd(gfadd(gfconv(Adj(6,1),S(8),
field),gfconv(Adj(6,2),S(9),field),field),gfconv(Adj(6,3),S(10),fi
eld),field),gfconv(Adj(6,4),S(11),field),field),gfconv(Adj(6,5),S(
12),field),field),gfconv(Adj(6,6),S(13),field),field),gfconv(Adj(6
,7),S(14),field),field);

Sigma(1)=gfadd(gfadd(gfadd(gfadd(gfadd(gfadd(gfconv(Adj(7,1),S(8),
field),gfconv(Adj(7,2),S(9),field),field),gfconv(Adj(7,3),S(10),fi
eld),field),gfconv(Adj(7,4),S(11),field),field),gfconv(Adj(7,5),S(
12),field),field),gfconv(Adj(7,6),S(13),field),field),gfconv(Adj(7
,7),S(14),field),field);
        Kappa(1,2)=Sigma(1);
        Kappa(1,3)=Sigma(2);
        Kappa(1,4)=Sigma(3);
        Kappa(1,5)=Sigma(4);
        Kappa(1,6)=Sigma(5);
        Kappa(1,7)=Sigma(6);
        Kappa(1,8)=Sigma(7);
          end
  else %terdapat 8 error
      kofaktor=kof(M,S,field);
      Adj=kofaktor';
      mydets=abs(determinan);
        for a=1:8
            for d=1:8
                Adj(a,d)=gfdeconv(Adj(a,d),mydets,field);
            end
        end
Sigma(8)=gfadd(gfadd(gfadd(gfadd(gfadd(gfadd(gfadd(gfconv(Adj(1,1)
,S(9),field),gfconv(Adj(1,2),S(10),field),field),gfconv(Adj(1,3),S
(11),field),field),gfconv(Adj(1,4),S(12),field),field),gfconv(Adj(
1,5),S(13),field),field),gfconv(Adj(1,6),S(14),field),field),gfcon
v(Adj(1,7),S(15),field),field),gfconv(Adj(1,8),S(16),field),field)
;

Sigma(7)=gfadd(gfadd(gfadd(gfadd(gfadd(gfadd(gfadd(gfconv(Adj(2,1)
,S(9),field),gfconv(Adj(2,2),S(10),field),field),gfconv(Adj(2,3),S
(11),field),field),gfconv(Adj(2,4),S(12),field),field),gfconv(Adj(
2,5),S(13),field),field),gfconv(Adj(2,6),S(14),field),field),gfcon
v(Adj(2,7),S(15),field),field),gfconv(Adj(2,8),S(16),field),field)
;

Sigma(6)=gfadd(gfadd(gfadd(gfadd(gfadd(gfadd(gfadd(gfconv(Adj(3,1)
,S(9),field),gfconv(Adj(3,2),S(10),field),field),gfconv(Adj(3,3),S
(11),field),field),gfconv(Adj(3,4),S(12),field),field),gfconv(Adj(
3,5),S(13),field),field),gfconv(Adj(3,6),S(14),field),field),gfcon
v(Adj(3,7),S(15),field),field),gfconv(Adj(3,8),S(16),field),field)
;

Sigma(5)=gfadd(gfadd(gfadd(gfadd(gfadd(gfadd(gfadd(gfconv(Adj(4,1)
,S(9),field),gfconv(Adj(4,2),S(10),field),field),gfconv(Adj(4,3),S
(11),field),field),gfconv(Adj(4,4),S(12),field),field),gfconv(Adj(
4,5),S(13),field),field),gfconv(Adj(4,6),S(14),field),field),gfcon
```

```
v(Adj(4,7),S(15),field),field),gfconv(Adj(4,8),S(16),field),field)
;

Sigma(4)=gfadd(gfadd(gfadd(gfadd(gfadd(gfadd(gfadd(gfconv(Adj(5,1)
,S(9),field),gfconv(Adj(5,2),S(10),field),field),gfconv(Adj(5,3),S
(11),field),field),gfconv(Adj(5,4),S(12),field),field),gfconv(Adj(
5,5),S(13),field),field),gfconv(Adj(5,6),S(14),field),field),gfcon
v(Adj(5,7),S(15),field),field),gfconv(Adj(5,8),S(16),field),field)
;

Sigma(3)=gfadd(gfadd(gfadd(gfadd(gfadd(gfadd(gfadd(gfconv(Adj(6,1)
,S(9),field),gfconv(Adj(6,2),S(10),field),field),gfconv(Adj(6,3),S
(11),field),field),gfconv(Adj(6,4),S(12),field),field),gfconv(Adj(
6,5),S(13),field),field),gfconv(Adj(6,6),S(14),field),field),gfcon
v(Adj(6,7),S(15),field),field),gfconv(Adj(6,8),S(16),field),field)
;

Sigma(2)=gfadd(gfadd(gfadd(gfadd(gfadd(gfadd(gfadd(gfconv(Adj(7,1)
,S(9),field),gfconv(Adj(7,2),S(10),field),field),gfconv(Adj(7,3),S
(11),field),field),gfconv(Adj(7,4),S(12),field),field),gfconv(Adj(
7,5),S(13),field),field),gfconv(Adj(7,6),S(14),field),field),gfcon
v(Adj(7,7),S(15),field),field),gfconv(Adj(7,8),S(16),field),field)
;

Sigma(1)=gfadd(gfadd(gfadd(gfadd(gfadd(gfadd(gfadd(gfconv(Adj(8,1)
,S(9),field),gfconv(Adj(8,2),S(10),field),field),gfconv(Adj(8,3),S
(11),field),field),gfconv(Adj(8,4),S(12),field),field),gfconv(Adj(
8,5),S(13),field),field),gfconv(Adj(8,6),S(14),field),field),gfcon
v(Adj(8,7),S(15),field),field),gfconv(Adj(8,8),S(16),field),field)
;
        Kappa(1,2)=Sigma(1);
        Kappa(1,3)=Sigma(2);
        Kappa(1,4)=Sigma(3);
        Kappa(1,5)=Sigma(4);
        Kappa(1,6)=Sigma(5);
        Kappa(1,7)=Sigma(6);
        Kappa(1,8)=Sigma(7);
        Kappa(1,9)=Sigma(8);
         end
end
%Menyusun error locator polynomial
    for i = 1:n
        sigma(i) = Kappa(1,i);
    end
```