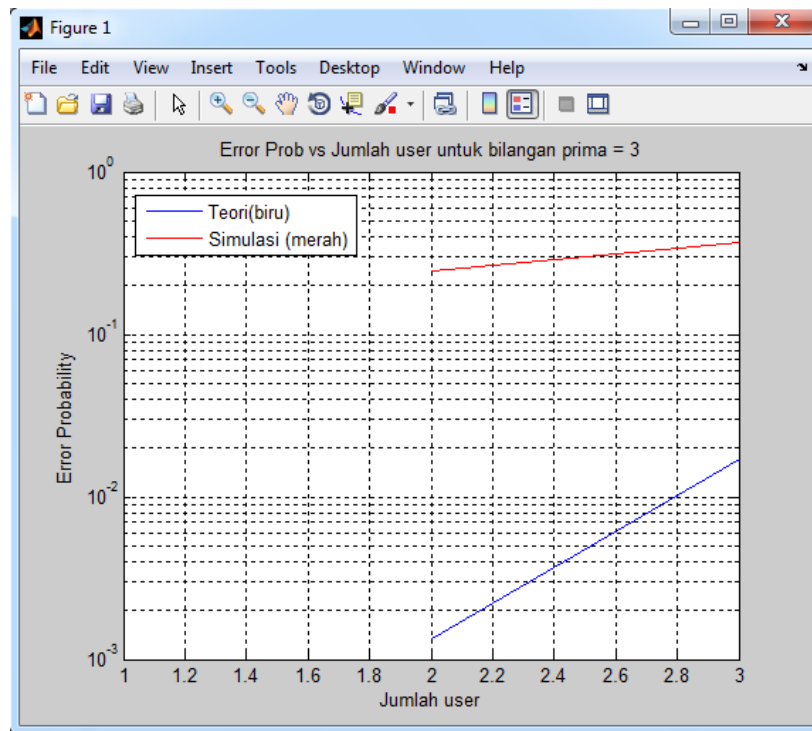


Lampiran Keterangan Gambar Data Pengamatan Pada Bab 4.



Pada Gambar terdapat garis biru mewakili teori, dan garis merah mewakili simulasi. Pada tugas akhir ini, Teori(biru) yang dimaksud adalah nilai kinerja probability of error dari Modified Prime Code (MPC) tanpa adanya Multiple Access Interference (MAI) yang disertakan pada proses pengiriman data. Sedangkan pada Simulasi(merah) yang dimaksud adalah nilai kinerja probability of error dari Modified Prime Code (MPC) dengan adanya Multiple Access Interference (MAI) yang muncul karena adanya pemakaian kanal yang sama dalam proses pengiriman data.

Program Modified Prime Code (MPC)

```
function[code]=mprime(P1)

%

% Function ini untuk membentuk Modified Prime Code (MPC)

%

% Variabel masukan : P1 : banyaknya bit "1"

%                               yang diinginkan pada kode

% Variabel keluaran : code MPC

%

N=P1^3-P1^2+P1;

%Inisialisasi variabel x, i, dan t

for x=0:P1-1,

    for i=0:P1-1,

        for t=0:P1-1,

            %Pembangkitan barisan prima

            s(x+1,i+1,t+1)=rem((x*i*t),P1);

        end;

    end;

end;
```

```
for x=0:P1-1,
    count=0;
    j=0;
    for i=0:N-1,
        count=count+1;
        if j<P1
            code(x+1,i+1,t+1)=0;
        end;
        if (rem(count,P1)==0)
            j=j+1;
        end;
    end;
end;
end;
```

Kode Prima

```
function keluar=kode_prima(masuk)

%

% Function ini untuk menghasilkan kode prima (prime code)

%

% Catatan : masukan berupa barisan prima

%

% Variabel masukan : masuk

% Variabel keluaran : keluar

%

keluar=zeros(size(masuk,1),size(masuk,1).*size(masuk,2));

for m=1:size(masuk,1)

    temp=masuk(m,:);

    for n=1:length(temp)

        end;

    end;
```

Baris Prima

```
function keluar=baris_prima(masuk)

%

% Function ini untuk menghasilkan barisan prima (prime sequence)

%

% Catatan : masukan harus berupa bilangan prima

%

% Variabel masukan : masuk = bilangan prima

%

% Variabel keluaran : keluar = barisan prima

%

cek=isprime(masuk);

p = masuk; % bilangan prima yang diinputkan

for m = 1 : p

    for n=1:p

        end;

    end;

    end;

    keluar=S;
```

Kali

```
function keluar = kali(masuk1,masuk2)

%
% Function ini untuk operasi perkalian
%
% Variabel masukan : masuk1 dan masuk2
% Variabel keluaran : keluar
%

if length(masuk1) ~= length(masuk2)

    error('Dimensi kedua sinyal masukan berbeda');

end;

for k=1:length(masuk1)

    if masuk1(k)==0 | masuk2(k)==0

        else

            keluar(k)=1;

        end;

    end;

end;
```

Modif Kode Prima Untuk Bilangan Prima 3

```
clear;
```

```
close all;
```

```
clc;
```

```
% Program Simulasi OCDMA dengan Modified Prime (MP) Code Sequences
```

```
% Untuk P=3
```

```
%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%
```

```
% P-1 sequences untuk membangun kode
```

```
s0=[1 0 0];
```

```
s1=[0 1 0];
```

```
s2=[0 0 1];
```

```
% Code sequences untuk group 0
```

```
c00=[s0 s0 s0];
```

```
c01=[s2 s2 s2];
```

```
c02=[s1 s1 s1];
```

```
% Code sequences untuk group 1
```

```
c10=[s0 s1 s2];
```

```
c11=[s1 s2 s0];
```

```
c12=[s2 s0 s1];
```

```
% Code sequences untuk group 2
```

```
c20=[s0 s2 s1];
```

```
c21=[s2 s1 s0];
```

```
c22=[s1 s0 s2];
```

```
% Penggabungan group
```

```
g0=[c00 c01 c02];
```

```
g1=[c10 c11 c12];
```

```
g2=[c20 c21 c22];
```

```
%%%%%%%%%
```

```
mod_pseq=[g0;g1;g2];
```

```
%-----
```

```
%
```

```
N=3; % Jumlah user (sama dengan bilangan prima yang dipilih)
```

```
ML=10000; % Maximum panjang data
```



```

% Proses pembentukan kode MP

Lc=length(mod_pseq(1,:));

% Jumlah user transmit serentak

for n=1:N

    err_count=0;

    % Pesan untuk user tertentu yang sdg transmit

    msg=zeros(n,ML);

% Proses pembentukan data unipolar

for co=1:n

    msg(co,:)=randuni(ML);

end;

for msgnum=1:ML

% Inialisasi sinyal yang ditransmisikan untuk user

trn_sig=zeros(N,Lc);

rpdmsg=zeros(N,Lc);

for co=1:n

    % Proses repetisi pesan sebanyak panjang kode (Lc)

```

```

for j=1:Lc

    rpdmsg(co,j)=msg(co,msgnum);

end;

% Modulasi dengan spreading sequences yang mirip

% ke optical encoder

trn_sig(co,:)=rpdmsg(co,:).*mod_pseq(co,:);

end;

%Inisialisasi sinyal osilator lokal

locsig=zeros(size(1:Lc));

%Interferensi dan proses korelasi

for co=1:n

    locsig=trn_sig(co,:)+locsig;

end;

%Encoding/decoding sinyal

temp=locsig.*mod_pseq(1,:);

decide=sum(temp);

```

```

% Keputusan benar / salah (threshold 3db)

if(decide<(N/2))

    decision=0;

else

    decision=1;

end;

if decision ~=msg(1,msgnum)

    err_count=err_count+1;

end;

end;

% Simulasi Error Probability

P(n)=err_count/ML;

% Kalkulasi error probability (secara teori)

clear k;

P1=3;

k=1:P1;

Pb=qfunct(P1./sqrt(k-1));

end;

P

```

```

Pb

nc=1:n;

clf;

semilogy(k,Pb,'b');hold on;

semilogy(nc,P,'r');grid on;

legend('Teori(biru),Simulasi (merah)');

xlabel('Jumlah user');

ylabel('Error Probability');

title(['Error Prob vs Jumlah user untuk bilangan prima = ',num2str(N)]);

kode_used1=kode_mod(1,:);

pp=2;    % variabel pp diganti selama diulang --> nilainya 2,3,4,...,p

kode_used2=kode_mod(pp,:);

%Pembangkitan data

% data=randuni(10); % Angka 10 bisa diganti integer lain --> usahakan tidak terlalu besar agar
bisa ditampilkan di layar

data=[1 0 1 1 1];

%Pengulangan data agar sama panjang dengan kode prima yang digunakan

data_ulang=ulang(p,data);

```

```
% Proses modulasi (data dikalikan dengan kode penebar yang digunakan
```

```
for k=1:length(data)
```

```
    data_termod_kali1((k-1)*p.^2+1:k*p.^2)=...
```

```
        kali(data_ulang((k-1)*p.^2+1:k*p.^2),kode_used1);
```

```
% Proses perhitungan fungsi korelasi silang antar 2 kode yang berbeda
```

```
for m=1:length(data)
```

```
    bantu1(1:p.^2)=data_termod_kali1((m-1)*p.^2+1:m*p.^2);
```

```
    bantu2(1:p.^2)=data_termod_kali2((m-1)*p.^2+1:m*p.^2);
```

```
    for k=1:p.^2
```

```
        kor(k)=hitung_korelasi(bantu1,circshift(bantu2,[0 k-1]));
```

```
    end;
```

```
    korelasi((m-1)*p.^2+1:m*p.^2)=kor;
```

```
    clear kor;
```

```
end;
```

```
% Tampilan grafik fungsi korelasi silang
```

```
figure;stem(1:length(korelasi),korelasi);grid;
```

```
xlabel('Indeks waktu');ylabel('Nilai fungsi korelasi silang');
```

Modif Kode Prima Untuk Bilangan Prima 5

```
clear;
```

```
close all;
```

```
clc;
```

```
% Program Simulasi OCDMA dengan Modified Prime (MP) Code Sequences
```

```
% Untuk P=5
```

```
%%%%%%%%%
```

```
% P-1 sequences untuk membangun kode
```

```
s0=[1 0 0 0 0];
```

```
s1=[0 1 0 0 0];
```

```
s2=[0 0 1 0 0];
```

```
s3=[0 0 0 1 0];
```

```
s4=[0 0 0 0 1];
```

```
% Code sequences untuk group 0
```

```
c00=[s0 s0 s0 s0 s0];
```

```
c01=[s4 s4 s4 s4 s4];
```

```
c02=[s3 s3 s3 s3 s3];
```

```
c03=[s2 s2 s2 s2 s2];
```

```
c04=[s1 s1 s1 s1 s1];
```

```
% Code sequences untuk group 1
```

```
c10=[s0 s1 s2 s3 s4];
```

```
c11=[s1 s2 s3 s4 s0];
```

```
c12=[s2 s3 s4 s0 s1];
```

```
c13=[s3 s4 s0 s1 s2];
```

```
c14=[s4 s0 s1 s2 s3];
```

```
% Code sequences untuk group 2
```

```
c20=[s0 s2 s4 s1 s3];
```

```
c21=[s2 s4 s1 s3 s0];
```

```
c22=[s4 s1 s3 s0 s2];
```

```
c23=[s1 s3 s0 s2 s4];
```

```
c24=[s3 s0 s2 s4 s1];
```

```
% Code sequences untuk group 3
```

```
c30=[s0 s3 s2 s1 s4];
```

```
c31=[s3 s2 s1 s4 s0];
```

```
c32=[s2 s1 s4 s0 s3];
```

```
c33=[s1 s4 s0 s3 s2];
```

```
c34=[s4 s0 s3 s2 s1];
```

```
% Code sequences untuk group 4
```

```
c40=[s0 s4 s1 s2 s3];
```

```
c41=[s4 s1 s2 s3 s0];
```

```
c42=[s1 s2 s3 s0 s4];
```

```
c43=[s2 s3 s0 s4 s1];
```

```
c44=[s3 s0 s4 s1 s2];
```

```
% Penggabungan group
```

```
g0=[c00 c01 c02 c03 c04];
```

```
g1=[c10 c11 c12 c13 c14];
```

```
g2=[c20 c21 c22 c23 c24];
```

```
g3=[c30 c31 c32 c33 c34];
```

```
g4=[c40 c41 c42 c43 c44];
```

```
%%%%%%%%%
```

```
mod_pseq=[g0;g1;g2;g3;g4];
```

```
%-----
```

```
%
```



```
N=5; % Jumlah user (sama dengan bilangan prima yang dipilih)
```

```
ML=10000; % Maximum panjang kode
```

```
% Proses pembentukan kode MP
```

```
Lc=length(mod_pseq(1,:));
```

```
% Jumlah user transmit serentak
```

```
for n=1:N
```

```
    err_count=0;
```

```
    % Pesan untuk user tertentu yang sdg transmit
```

```
    msg=zeros(n,ML);
```

```
    % Proses pembentukan data unipolar
```

```
    for co=1:n
```

```
        msg(co,:)=randuni(ML);
```

```
    end;
```

```
for msgnum=1:ML
```

```
    % Inisialisasi sinyal yang ditransmisikan untuk user
```

```
    trn_sig=zeros(N,Lc);
```

```

rpdmsg=zeros(N,Lc);

for co=1:n

    % Proses repetisi pesan sebanyak panjang kode (Lc)

    for j=1:Lc

        rpdmsg(co,j)=msg(co,msgnum);

    end;

    % Modulasi dengan spreading sequences yang mirip

    % ke optical encoder

    trn_sig(co,:)=rpdmsg(co,:).*mod_pseq(co,:);

end;

%Inisialisasi sinyal osilator lokal

locsig=zeros(size(1:Lc));

%Interferensi dan proses korelasi

for co=1:n

    locsig=trn_sig(co,:)+locsig;

end;

%Encoding/decoding sinyal

```

```

temp=locsig.*mod_pseq(1,:);

decide=sum(temp);

% Keputusan benar / salah (threshold 3db)

if(decide<(N/2))

    decision=0;

else

    decision=1;

end;

if decision ~=msg(1,msgnum)

    err_count=err_count+1;

end;

end;

% Simulasi Error Probability

P(n)=err_count/ML;

% Kalkulasi error probability (secara teori)

clear k;

P1=5;

k=1:P1;

```

```

    Pb=qfunct(P1./sqrt(k-1));

end;

P

Pb

nc=1:n;

clf;

semilogy(k,Pb,'b');hold on;

semilogy(nc,P,'r');grid on;

legend('Teori(biru),Simulasi (merah)');

xlabel('Jumlah user');

ylabel('Error Probability');

title(['Error Prob vs Jumlah user untuk bilangan prima = ',num2str(N)]);

```

% Pengujian perhitungan fungsi korelasi silang untuk bilangan prima (p) = 5

```
p=5;
```

```
[kode kode_mod]=modif_kode_prima(p);
```

```
kode_used1=kode_mod(1,:);
```

```
pp=4; % variabel pp diganti selama diulang --> nilainya 2,3,4,...,p
```

```

kode_used2=kode_mod(pp,:);

%Pembangkitan data

% data=randuni(10); % Angka 10 bisa diganti integer lain --> usahakan tidak terlalu besar agar
bisa ditampilkan di layar

data=[1 0 1 1 1];

%Pengulangan data agar sama panjang dengan kode prima yang digunakan

data_ulang=ulang(p,data);

% Proses modulasi (data dikalikan dengan kode penebar yang digunakan

for k=1:length(data)

    data_termod_kali1((k-1)*p.^2+1:k*p.^2)=...

        kali(data_ulang((k-1)*p.^2+1:k*p.^2),kode_used2);

end;

% Proses perhitungan fungsi korelasi silang antar 2 kode yang berbeda

for m=1:length(data)

    bantu1(1:p.^2)=data_termod_kali1((m-1)*p.^2+1:m*p.^2);

    bantu2(1:p.^2)=data_termod_kali2((m-1)*p.^2+1:m*p.^2);

    for k=1:p.^2

        kor(k)=hitung_korelasi(bantu1,circshift(bantu2,[0 k-1]));

```

```

end;

korelasi((m-1)*p.^2+1:m*p.^2)=kor;

clear kor;

end;

% Tampilan grafik fungsi korelasi silang

figure;stem(1:length(korelasi),korelasi);grid;

title(['Perhitungan Korelasi Silang untuk C1 dan C',num2str(pp),' pada GF(',num2str(p),') untuk
data [',...

num2str(data),']]');

xlabel('Indeks waktu');ylabel('Nilai fungsi korelasi silang');

```

Modif Kode Prima Untuk Bilangan Prima 7

```
clear;
```

```
close all;
```

```
clc;
```

```
% Program Simulasi OCDMA dengan Modified Prime (MP) Code Sequences
```

```
% Untuk P=7
```

```
%%%%%%%%%
```

```
% P-1 sequences untuk membangun kode
```

```
s0=[1 0 0 0 0 0];
```

```
s1=[0 1 0 0 0 0];
```

```
s2=[0 0 1 0 0 0];
```

```
s3=[0 0 0 1 0 0];
```

```
s4=[0 0 0 0 1 0];
```

```
s5=[0 0 0 0 0 1];
```

```
s6=[0 0 0 0 0 1];
```

```
% Code sequences untuk group 0
```

```
c00=[s0 s0 s0 s0 s0 s0];
```

c01=[s6 s6 s6 s6 s6 s6 s6];

c02=[s5 s5 s5 s5 s5 s5 s5];

c03=[s4 s4 s4 s4 s4 s4 s4];

c04=[s3 s3 s3 s3 s3 s3 s3];

c05=[s2 s2 s2 s2 s2 s2 s2];

c06=[s1 s1 s1 s1 s1 s1 s1];

% Code sequences untuk group 1

c10=[s0 s1 s2 s3 s4 s5 s6];

c11=[s1 s2 s3 s4 s5 s6 s0];

c12=[s2 s3 s4 s5 s6 s0 s1];

c13=[s3 s4 s5 s6 s0 s1 s2];

c14=[s4 s5 s6 s0 s1 s2 s3];

c15=[s5 s6 s0 s1 s2 s3 s4];

c16=[s6 s0 s1 s2 s3 s4 s5];

% Code sequences untuk group 2

c20=[s0 s2 s4 s6 s1 s3 s5];

c21=[s2 s4 s6 s1 s3 s5 s0];

c22=[s4 s6 s1 s3 s5 s0 s2];

c23=[s6 s1 s3 s5 s0 s2 s4];

c24=[s1 s3 s5 s0 s2 s4 s6];

c25=[s3 s5 s0 s2 s4 s6 s1];

c26=[s5 s0 s2 s4 s6 s1 s3];

% Code sequences untuk group 3

c30=[s0 s3 s6 s2 s5 s1 s4];

c31=[s3 s6 s2 s5 s1 s4 s0];

c32=[s6 s2 s5 s1 s4 s0 s3];

c33=[s2 s5 s1 s4 s0 s3 s6];

c34=[s5 s1 s4 s0 s3 s6 s2];

c35=[s1 s4 s0 s3 s6 s2 s5];

c36=[s4 s0 s3 s6 s2 s5 s1];

% Code sequences untuk group 4

c40=[s0 s4 s1 s5 s2 s6 s3];

c41=[s4 s1 s5 s2 s6 s3 s0];

c42=[s1 s5 s2 s6 s3 s0 s4];

c43=[s5 s2 s6 s3 s0 s4 s1];

c44=[s2 s6 s3 s0 s4 s1 s5];

c45=[s6 s3 s0 s4 s1 s5 s2];

c46=[s3 s0 s4 s1 s5 s2 s6];

```
% Code sequences untuk group 5
```

```
c50=[s0 s5 s3 s1 s6 s4 s2];
```

```
c51=[s5 s3 s1 s6 s4 s2 s0];
```

```
c52=[s3 s1 s6 s4 s2 s0 s5];
```

```
c53=[s1 s6 s4 s2 s0 s5 s3];
```

```
c54=[s6 s4 s2 s0 s5 s3 s1];
```

```
c55=[s4 s2 s0 s5 s3 s1 s6];
```

```
c56=[s2 s0 s5 s3 s1 s6 s4];
```

```
% Code sequences untuk group 6
```

```
c60=[s0 s6 s5 s4 s3 s2 s1];
```

```
c61=[s6 s5 s4 s3 s2 s1 s0];
```

```
c62=[s5 s4 s3 s2 s1 s0 s6];
```

```
c63=[s4 s3 s2 s1 s0 s6 s5];
```

```
c64=[s3 s2 s1 s0 s6 s5 s4];
```

```
c65=[s2 s1 s0 s6 s5 s4 s3];
```

```
c66=[s1 s0 s6 s5 s4 s3 s2];
```

```
% Penggabungan group
```

```
g0=[c00 c01 c02 c03 c04 c05 c06];
```

```

g1=[c10 c11 c12 c13 c14 c15 c16];
g2=[c20 c21 c22 c23 c24 c25 c26];
g3=[c30 c31 c32 c33 c34 c35 c36];
g4=[c40 c41 c42 c43 c44 c45 c46];
g5=[c50 c51 c52 c53 c54 c55 c56];
g6=[c60 c61 c62 c63 c64 c65 c66];

%%%%%%%%%%

mod_pseq=[g0;g1;g2;g3;g4;g5;g6];

%-----

N=7; % Jumlah user (sama dengan bilangan prima yang dipilih)

ML=10000; % Maximum panjang data

% Proses pembentukan kode MP

Lc=length(mod_pseq(1,:));

% Jumlah user transmit serentak

for n=1:N

    err_count=0;

    % Pesan untuk user tertentu yang sdg transmit

    msg=zeros(n,ML);

```

```

% Proses pembentukan data unipolar

for co=1:n

    msg(co,:)=randuni(ML);

end;

for msgnum=1:ML

    % Inisialisasi sinyal yang ditransmisikan untuk user

    trn_sig=zeros(N,Lc);

    rpdmsg=zeros(N,Lc);

    for co=1:n

        % Proses repetisi pesan sebanyak panjang kode (Lc)

        for j=1:Lc

            rpdmsg(co,j)=msg(co,msgnum);

        end;

    % Modulasi dengan spreading sequences yang mirip

    % ke optical encoder

    trn_sig(co,:)=rpdmsg(co,:).*mod_pseq(co,:);

end;

```

```

%Inisialisasi sinyal osilator lokal

locsig=zeros(size(1:Lc));

%Interferensi dan proses korelasi

for co=1:n

    locsig=trn_sig(co,)+locsig;

end;

%Encoding/decoding sinyal

temp=locsig.*mod_pseq(1,:);

decide=sum(temp);

% Keputusan benar / salah (threshold 3db)

if(decide<(N/2))

    decision=0;

else

    decision=1;

end;

if decision ~=msg(1,msgnum)

    err_count=err_count+1;

```

```

    end;

end;

% Simulasi Error Probability

P(n)=err_count/ML;

% Kalkulasi error probability (secara teori)

clear k;

P1=7;

k=1:P1;

Pb=qfunct(P1./sqrt(k-1));

end;

P

Pb

nc=1:n;

clf;

semilogy(k,Pb,'b');hold on;

semilogy(nc,P,'r');grid on;

legend('Teori(biru),Simulasi (merah)');

xlabel('Jumlah user');

ylabel('Error Probability');

```

```

title(['Error Prob vs Jumlah user untuk bilangan prima = ',num2str(N)]);

% Pengujian perhitungan fungsi korelasi silang untuk bilangan prima (p) = 7
p=7;

[kode kode_mod]=modif_kode_prima(p);

kode_used1=kode_mod(1,:);

pp=5; % variabel pp diganti selama diulang --> nilainya 2,3,4,...,p
kode_used2=kode_mod(pp,:);

%Pembangkitan data

% data=randuni(10); % Angka 10 bisa diganti integer lain --> usahakan tidak terlalu besar agar
bisa ditampilkan di layar

data=[1 0 1 1 1];

%Pengulangan data agar sama panjang dengan kode prima yang digunakan

data_ulang=ulang(p,data);

% Proses modulasi (data dikalikan dengan kode penebar yang digunakan

for k=1:length(data)

    data_termod_kali1((k-1)*p.^2+1:k*p.^2)=...

        kali(data_ulang((k-1)*p.^2+1:k*p.^2),kode_used2);

```

```

end;

% Proses perhitungan fungsi korelasi silang antar 2 kode yang berbeda

for m=1:length(data)

    bantu2(1:p.^2)=data_termod_kali2((m-1)*p.^2+1:m*p.^2);

    for k=1:p.^2

        kor(k)=hitung_korelasi(bantu1,circshift(bantu2,[0 k-1]));

    end;

    korelasi((m-1)*p.^2+1:m*p.^2)=kor;

    clear kor;

end;

% Tampilan grafik fungsi korelasi silang

figure;stem(1:length(korelasi),korelasi);grid;

title(['Perhitungan Korelasi Silang untuk C1 dan C',num2str(pp),' pada GF(',num2str(p),') untuk
data [',...

    num2str(data),']]');

xlabel('Indeks waktu');ylabel('Nilai fungsi korelasi silang');

```


Modif Kode Prima Untuk Bilangan Prima 11

```
clear;
```

```
close all;
```

```
clc;
```

```
% Program Simulasi OCDMA dengan Modified Prime (MP) Code Sequences
```

```
% Untuk P=11
```

```
%%%%%%%%%
```

```
% P-1 sequences untuk membangun kode
```

```
s0=[1 0 0 0 0 0 0 0 0 0];
```

```
s1=[0 1 0 0 0 0 0 0 0 0];
```

```
s2=[0 0 1 0 0 0 0 0 0 0];
```

```
s3=[0 0 0 1 0 0 0 0 0 0];
```

```
s4=[0 0 0 0 1 0 0 0 0 0];
```

```
s5=[0 0 0 0 0 1 0 0 0 0];
```

```
s6=[0 0 0 0 0 0 1 0 0 0];
```

```
s7=[0 0 0 0 0 0 0 1 0 0];
```

```
s8=[0 0 0 0 0 0 0 0 1 0];
```

```
s9=[0 0 0 0 0 0 0 0 0 1];
```

```
s10=[0 0 0 0 0 0 0 0 0 0 1];
```

```
% Code sequences untuk group 0
```

```
c00=[s0 s0 s0 s0 s0 s0 s0 s0 s0 s0 s0];
```

```
c01=[s10 s10 s10 s10 s10 s10 s10 s10 s10 s10 s10];
```

```
c02=[s9 s9 s9 s9 s9 s9 s9 s9 s9 s9 s9];
```

```
c03=[s8 s8 s8 s8 s8 s8 s8 s8 s8 s8 s8];
```

```
c04=[s7 s7 s7 s7 s7 s7 s7 s7 s7 s7 s7];
```

```
c05=[s6 s6 s6 s6 s6 s6 s6 s6 s6 s6 s6];
```

```
c06=[s5 s5 s5 s5 s5 s5 s5 s5 s5 s5 s5];
```

```
c07=[s4 s4 s4 s4 s4 s4 s4 s4 s4 s4 s4];
```

```
c08=[s3 s3 s3 s3 s3 s3 s3 s3 s3 s3 s3];
```

```
c09=[s2 s2 s2 s2 s2 s2 s2 s2 s2 s2 s2];
```

```
c010=[s1 s1 s1 s1 s1 s1 s1 s1 s1 s1 s1];
```

```
% Code sequences untuk group 1
```

```
c10=[s0 s1 s2 s3 s4 s5 s6 s7 s8 s9 s10];
```

```
c11=[s1 s2 s3 s4 s5 s6 s7 s8 s9 s10 s0];
```

```
c12=[s2 s3 s4 s5 s6 s7 s8 s9 s10 s0 s1];
```

```
c13=[s3 s4 s5 s6 s7 s8 s9 s10 s0 s1 s2];
```

```
c14=[s4 s5 s6 s7 s8 s9 s10 s0 s1 s2 s3];
```

```
c15=[s5 s6 s7 s8 s9 s10 s0 s1 s2 s3 s4];  
c16=[s6 s7 s8 s9 s10 s0 s1 s2 s3 s4 s5];  
c17=[s7 s8 s9 s10 s0 s1 s2 s3 s4 s5 s6];  
c18=[s8 s9 s10 s0 s1 s2 s3 s4 s5 s6 s7];  
c19=[s9 s10 s0 s1 s2 s3 s4 s5 s6 s7 s8];  
c110=[s10 s0 s1 s2 s3 s4 s5 s6 s7 s8 s9];
```

```
% Code sequences untuk group 2
```

```
c20=[s0 s2 s4 s6 s8 s10 s1 s3 s5 s7 s9];  
c21=[s2 s4 s6 s8 s10 s1 s3 s5 s7 s9 s0];  
c22=[s4 s6 s8 s10 s1 s3 s5 s7 s9 s0 s2];  
c23=[s6 s8 s10 s1 s3 s5 s7 s9 s0 s2 s4];  
c24=[s8 s10 s1 s3 s5 s7 s9 s0 s2 s4 s6];  
c25=[s10 s1 s3 s5 s7 s9 s0 s2 s4 s6 s8];  
c26=[s1 s3 s5 s7 s9 s0 s2 s4 s6 s8 s10];  
c27=[s3 s5 s7 s9 s0 s2 s4 s6 s8 s10 s1];  
c28=[s5 s7 s9 s0 s2 s4 s6 s8 s10 s1 s3];  
c29=[s7 s9 s0 s2 s4 s6 s8 s10 s1 s3 s5];  
c210=[s9 s0 s2 s4 s6 s8 s10 s1 s3 s5 s7];
```

```
% Code sequences untuk group 3
```

c30=[s0 s3 s6 s9 s2 s5 s8 s1 s4 s7 s10];
c31=[s3 s6 s9 s2 s5 s8 s1 s4 s7 s10 s0];
c32=[s6 s9 s2 s5 s8 s1 s4 s7 s10 s0 s3];
c33=[s9 s2 s5 s8 s1 s4 s7 s10 s0 s3 s6];
c34=[s2 s5 s8 s1 s4 s7 s10 s0 s3 s6 s9];
c35=[s5 s8 s1 s4 s7 s10 s0 s3 s6 s9 s2];
c36=[s8 s1 s4 s7 s10 s0 s3 s6 s9 s2 s5];
c37=[s1 s4 s7 s10 s0 s3 s6 s9 s2 s5 s8];
c38=[s4 s7 s10 s0 s3 s6 s9 s2 s5 s8 s1];
c39=[s7 s10 s0 s3 s6 s9 s2 s5 s8 s1 s4];
c310=[s10 s0 s3 s6 s9 s2 s5 s8 s1 s4 s7];

% Code sequences untuk group 4

c40=[s0 s4 s8 s1 s5 s9 s2 s6 s10 s3 s7];
c41=[s4 s8 s1 s5 s9 s2 s6 s10 s3 s7 s0];
c42=[s8 s1 s5 s9 s2 s6 s10 s3 s7 s0 s4];
c43=[s1 s5 s9 s2 s6 s10 s3 s7 s0 s4 s8];
c44=[s5 s9 s2 s6 s10 s3 s7 s0 s4 s8 s1];
c45=[s9 s2 s6 s10 s3 s7 s0 s4 s8 s1 s5];
c46=[s2 s6 s10 s3 s7 s0 s4 s8 s1 s5 s9];

```
c47=[s6 s10 s3 s7 s0 s4 s8 s1 s5 s9 s2];  
c48=[s10 s3 s7 s0 s4 s8 s1 s5 s9 s2 s6];  
c49=[s3 s7 s0 s4 s8 s1 s5 s9 s2 s6 s10];  
c410=[s7 s0 s4 s8 s1 s5 s9 s2 s6 s10 s3];
```

```
% Code sequences untuk group 5
```

```
c50=[s0 s5 s10 s3 s8 s1 s6 s4 s9 s2 s7];  
c51=[s5 s10 s3 s8 s1 s6 s4 s9 s2 s7 s0];  
c52=[s10 s3 s8 s1 s6 s4 s9 s2 s7 s0 s5];  
c53=[s3 s8 s1 s6 s4 s9 s2 s7 s0 s5 s10];  
c54=[s8 s1 s6 s4 s9 s2 s7 s0 s5 s10 s3];  
c55=[s1 s6 s4 s9 s2 s7 s0 s5 s10 s3 s8];  
c56=[s6 s4 s9 s2 s7 s0 s5 s10 s3 s8 s1];  
c57=[s4 s9 s2 s7 s0 s5 s10 s3 s8 s1 s6];  
c58=[s9 s2 s7 s0 s5 s10 s3 s8 s1 s6 s4];  
c59=[s2 s7 s0 s5 s10 s3 s8 s1 s6 s4 s9];  
c510=[s7 s0 s5 s10 s3 s8 s1 s6 s4 s9 s2];
```

```
% Code sequences untuk group 6
```

```
c60=[s0 s6 s5 s4 s10 s3 s9 s2 s8 s1 s7];  
c61=[s6 s5 s4 s10 s3 s9 s2 s8 s1 s7 s0];
```

```
c62=[s5 s4 s10 s3 s9 s2 s8 s1 s7 s0 s6];
c63=[s4 s10 s3 s9 s2 s8 s1 s7 s0 s6 s5];
c64=[s10 s3 s9 s2 s8 s1 s7 s0 s6 s5 s4];
c65=[s3 s9 s2 s8 s1 s7 s0 s6 s5 s4 s10];
c66=[s9 s2 s8 s1 s7 s0 s6 s5 s4 s10 s3];
c67=[s2 s8 s1 s7 s0 s6 s5 s4 s10 s3 s9];
c68=[s8 s1 s7 s0 s6 s5 s4 s10 s3 s9 s2];
c69=[s1 s7 s0 s6 s5 s4 s10 s3 s9 s2 s8];
c610=[s7 s0 s6 s5 s4 s10 s3 s9 s2 s8 s1];
```

```
% Code sequences untuk group 7
```

```
c70=[s0 s7 s1 s8 s2 s9 s3 s10 s4 s5 s6];
c71=[s7 s1 s8 s2 s9 s3 s10 s4 s5 s6 s0];
c72=[s1 s8 s2 s9 s3 s10 s4 s5 s6 s0 s7];
c73=[s8 s2 s9 s3 s10 s4 s5 s6 s0 s7 s1];
c74=[s2 s9 s3 s10 s4 s5 s6 s0 s7 s1 s8];
c75=[s9 s3 s10 s4 s5 s6 s0 s7 s1 s8 s2];
c76=[s3 s10 s4 s5 s6 s0 s7 s1 s8 s2 s9];
c77=[s10 s4 s5 s6 s0 s7 s1 s8 s2 s9 s3];
c78=[s4 s5 s6 s0 s7 s1 s8 s2 s9 s3 s10];
c79=[s5 s6 s0 s7 s1 s8 s2 s9 s3 s10 s4];
```

c710=[s6 s0 s7 s1 s8 s2 s9 s3 s10 s4 s5];

% Code sequences untuk group 8

c80=[s0 s8 s2 s10 s1 s9 s3 s4 s5 s6 s7];

c81=[s8 s2 s10 s1 s9 s3 s4 s5 s6 s7 s0];

c82=[s2 s10 s1 s9 s3 s4 s5 s6 s7 s0 s8];

c83=[s10 s1 s9 s3 s4 s5 s6 s7 s0 s8 s2];

c84=[s1 s9 s3 s4 s5 s6 s7 s0 s8 s2 s10];

c85=[s9 s3 s4 s5 s6 s7 s0 s8 s2 s10 s1];

c86=[s3 s4 s5 s6 s7 s0 s8 s2 s10 s1 s9];

c87=[s4 s5 s6 s7 s0 s8 s2 s10 s1 s9 s3];

c88=[s5 s6 s7 s0 s8 s2 s10 s1 s9 s3 s4];

c89=[s6 s7 s0 s8 s2 s10 s1 s9 s3 s4 s5];

c810=[s7 s0 s8 s2 s10 s1 s9 s3 s4 s5 s6];

% Code sequences untuk group 9

c90=[s0 s9 s1 s10 s8 s7 s6 s5 s4 s3 s2];

c91=[s9 s1 s10 s8 s7 s6 s5 s4 s3 s2 s0];

c92=[s1 s10 s8 s7 s6 s5 s4 s3 s2 s0 s9];

c93=[s10 s8 s7 s6 s5 s4 s3 s2 s0 s9 s1];

c94=[s8 s7 s6 s5 s4 s3 s2 s0 s9 s1 s10];

```
c95=[s7 s6 s5 s4 s3 s2 s0 s9 s1 s10 s8];  
c96=[s6 s5 s4 s3 s2 s0 s9 s1 s10 s8 s7];  
c97=[s5 s4 s3 s2 s0 s9 s1 s10 s8 s7 s6];  
c98=[s4 s3 s2 s0 s9 s1 s10 s8 s7 s6 s5];  
c99=[s3 s2 s0 s9 s1 s10 s8 s7 s6 s5 s4];  
c910=[s2 s0 s9 s1 s10 s8 s7 s6 s5 s4 s3];
```

```
% Code sequences untuk group 10
```

```
c100=[s0 s10 s9 s8 s7 s6 s5 s4 s3 s2 s1];  
c101=[s10 s9 s8 s7 s6 s5 s4 s3 s2 s1 s0];  
c102=[s9 s8 s7 s6 s5 s4 s3 s2 s1 s0 s10];  
c103=[s8 s7 s6 s5 s4 s3 s2 s1 s0 s10 s9];  
c104=[s7 s6 s5 s4 s3 s2 s1 s0 s10 s9 s8];  
c105=[s6 s5 s4 s3 s2 s1 s0 s10 s9 s8 s7];  
c106=[s5 s4 s3 s2 s1 s0 s10 s9 s8 s7 s6];  
c107=[s4 s3 s2 s1 s0 s10 s9 s8 s7 s6 s5];  
c108=[s3 s2 s1 s0 s10 s9 s8 s7 s6 s5 s4];  
c109=[s2 s1 s0 s10 s9 s8 s7 s6 s5 s4 s3];  
c1010=[s1 s0 s10 s9 s8 s7 s6 s5 s4 s3 s2];
```

```
% Penggabungan group
```



```

g0=[c00 c01 c02 c03 c04 c05 c06 c07 c08 c09 c10];
g1=[c10 c11 c12 c13 c14 c15 c16 c17 c18 c19 c110];
g2=[c20 c21 c22 c23 c24 c25 c26 c27 c28 c29 c210];
g3=[c30 c31 c32 c33 c34 c35 c36 c37 c38 c39 c310];
g4=[c40 c41 c42 c43 c44 c45 c46 c47 c48 c49 c410];
g5=[c50 c51 c52 c53 c54 c55 c56 c57 c58 c59 c510];
g6=[c60 c61 c62 c63 c64 c65 c66 c67 c68 c69 c610];
g7=[c70 c71 c72 c73 c74 c75 c76 c77 c78 c79 c710];
g8=[c80 c81 c82 c83 c84 c85 c86 c87 c88 c89 c810];
g9=[c90 c91 c92 c93 c94 c95 c96 c97 c98 c99 c910];
g10=[c100 c101 c102 c103 c104 c105 c106 c107 c108 c109 c1010];

```

```

%%%%%%%%%%

```

```

mod_pseq=[g0;g1;g2;g3;g4;g5;g6;g7;g8;g9;g10];

```

```

%-----

```

```

%

```

```

N=11; % Jumlah user (sama dengan bilangan prima yang dipilih)

```

```

ML=10000; % Maximum panjang data

```

```

% Proses pembentukan kode MP

```

```

Lc=length(mod_pseq(1,:));

```

```

% Jumlah user transmit serentak

for n=1:N

    err_count=0;

    % Pesan untuk user tertentu yang sdg transmit

    msg=zeros(n,ML);

    % Proses pembentukan data unipolar

    for co=1:n

        msg(co,:)=randuni(ML);

    end;

    for msgnum=1:ML

        % Inisialisasi sinyal yang ditransmisikan untuk user

        trn_sig=zeros(N,Lc);

        rpdmsg=zeros(N,Lc);

        for co=1:n

            % Proses repetisi pesan sebanyak panjang kode (Lc)

            for j=1:Lc

                rpdmsg(co,j)=msg(co,msgnum);

```

```

end;

% Modulasi dengan spreading sequences yang mirip
% ke optical encoder
trn_sig(co,:)=rpdmsg(co,:).*mod_pseq(co,:);

end;

%Inisialisasi sinyal osilator lokal
locsig=zeros(size(1:Lc));

%Interferensi dan proses korelasi
for co=1:n
    locsig=trn_sig(co,:)+locsig;
end;

%Encoding/decoding sinyal
temp=locsig.*mod_pseq(1,:);
decide=sum(temp);

% Keputusan benar / salah (threshold 3db)
if(decide<(N/2))

```

```

        decision=0;

    else

        decision=1;

    end;

    if decision ~=msg(1,msgnum)

        err_count=err_count+1;

    end;

end;

% Simulasi Error Probability

P(n)=err_count/ML;

% Kalkulasi error probability (secara teori)

clear k;

P1=11;

k=1:12;

Pb=qfunct(P1./sqrt(k-1));

end;

P

Pb

```

```

nc=1:n;

clf;

semilogy(k,Pb,'b');hold on;

semilogy(nc,P,'r');grid on;

legend('Teori(biru),Simulasi (merah)');

xlabel('Jumlah user');

ylabel('Error Probability');

title(['Error Prob vs Jumlah user untuk bilangan prima = ',num2str(N)]);

```

% Pengujian perhitungan fungsi korelasi silang untuk bilangan prima (p)=11

```
p=11;
```

```
[kode kode_mod]=modif_kode_prima(p);
```

```
kode_used1=kode_mod(1,:);
```

```
pp=7; % variabel pp diganti selama diulang --> nilainya 2,3,4,...,p
```

```
kode_used2=kode_mod(pp,:);
```

```
%Pembangkitan data
```

```
% data=randuni(10); % Angka 10 bisa diganti integer lain --> usahakan tidak terlalu besar agar  
bisa ditampilkan di layar
```

```
data=[1 0 1 1 1];
```

```
%Pengulangan data agar sama panjang dengan kode prima yang digunakan
```

```
data_ulang=ulang(p,data);
```

```
% Proses modulasi (data dikalikan dengan kode penebar yang digunakan
```

```
for k=1:length(data)
```

```
    data_termod_kali1((k-1)*p.^2+1:k*p.^2)=...
```

```
        kali(data_ulang((k-1)*p.^2+1:k*p.^2),kode_used2);
```

```
end;
```

```
% Proses perhitungan fungsi korelasi silang antar 2 kode yang berbeda
```

```
for m=1:length(data)
```

```
    bantu1(1:p.^2)=data_termod_kali1((m-1)*p.^2+1:m*p.^2);
```

```
    bantu2(1:p.^2)=data_termod_kali2((m-1)*p.^2+1:m*p.^2);
```

```
    for k=1:p.^2
```

```
        kor(k)=hitung_korelasi(bantu1,circshift(bantu2,[0 k-1]));
```

```
    end;
```

```
    korelasi((m-1)*p.^2+1:m*p.^2)=kor;
```

```
clear kor;
```

```
end;
```

```
% Tampilan grafik fungsi korelasi silang
```

```
figure;stem(1:length(korelasi),korelasi);grid;
```

```
title(['Perhitungan Korelasi Silang untuk C1 dan C',num2str(pp),' pada GF(',num2str(p),') untuk  
data [',...
```

```
num2str(data),']]');
```

```
xlabel('Indeks waktu');ylabel('Nilai fungsi korelasi silang');
```

Qfunct

```
function[y]=qfunct(x)
```

```
%qfunct evaluates the Q-Function
```

```
y=(1/2)*erfc(x/sqrt(2));
```


Fungsi membangkitkan data

```
function[p]=randuni(N);  
  
%  
  
% Fungsi ini untuk membangkitkan data unipolar  
  
%  
  
% Variabel masukan : N = banyaknya data yang ingin dikirim  
  
%  
  
% Variabel keluaran : p  
  
for i=1:N,  
  
    temp=rand;  
  
    data(1,i)=0;  
  
    else  
  
        data(1,i)=1;  
  
    end;  
  
end;  
  
p=data(1,:);
```

Fungsi untuk mengulang data

```
function keluar=ulang(p,masuk)

%

% Function ini untuk mengulang data user

% sesuai dengan periode chip yang digunakan(p^2)

%

% Variabel masukan : p =bilangan prima

%           masuk = data user

%

% Variabel keluaran : keluar = data user yang

%           sudah di-repetisi (diulang)

%           sehingga panjangnya sama

%           dengan banyak data * p^2

%

for m=1:length(masuk)

    keluar((m-1)*p^2+1:m*p^2)=masuk(m);

end;
```

Hitung Korelasi

```
function [keluar]=hitung_korelasi(masuk1,masuk2)

%

% Function untuk menghitung nilai korelasi antara

% dua buah kode (korelasi sendiri atau korelasi silang)

%

% Variabel masukan : masuk1=kode pertama

%           masuk2=kode kedua

% Variabel keluaran : keluar = nilai korelasi

%

if (length(masuk1) ~= length (masuk2))

    error('Panjang kode tidak sama');

end;

keluar=0;

for k=1:length(masuk1)

    if (masuk1(k)==1 & masuk2(k)==1)

        end;

end;
```

Hitung Korelasi Silang

```
clear;
```

```
close all;
```

```
clc;
```

```
p=5;
```

```
[kode kode_mod]=modif_kode_prima(p);
```

```
kode_used1=kode_mod(1,:);
```

```
pp=3; % variabel pp diganti selama diulang --> nilainya 2,3,4,....,p
```

```
kode_used2=kode_mod(pp,:);
```

```
%Pembangkitan data
```

```
% data=randuni(10); % Angka 10 bisa diganti integer lain --> usahakan tidak terlalu besar agar  
bisa ditampilkan di layar
```

```
%Pengulangan data agar sama panjang dengan kode prima yang digunakan
```

```
data_ulang=ulang(p,data);
```

```
% Proses modulasi (data dikalikan dengan kode penebar yang digunakan
```

```
for k=1:length(data)
```

```

data_termod_kali1((k-1)*p.^2+1:k*p.^2)=...

kali(data_ulang((k-1)*p.^2+1:k*p.^2),kode_used1);

end;

% Proses perhitungan fungsi korelasi silang antar 2 kode yang berbeda

for m=1:length(data)

    bantu1(1:p.^2)=data_termod_kali1((m-1)*p.^2+1:m*p.^2);

    bantu2(1:p.^2)=data_termod_kali2((m-1)*p.^2+1:m*p.^2);

    for k=1:p.^2

        kor(k)=hitung_korelasi(bantu1,circshift(bantu2,[0 k-1]));

    end;

    korelasi((m-1)*p.^2+1:m*p.^2)=kor;

    clear kor;

end;

% Tampilan grafik fungsi korelasi silang

figure;stem(1:length(korelasi),korelasi);grid;

title(['Perhitungan Korelasi Silang untuk C1 dan C',num2str(pp),' pada GF(',num2str(p),') untuk
data [',...

    num2str(data),']]);

xlabel('Indeks waktu');ylabel('Nilai fungsi korelasi silang');

```