

**LAMPIRAN A:**  
**LIST PROGRAM**

## **%Program Utama**

```
clear;
```

```
close all;
```

```
clc;
```

```
% Program ini adalah untuk membentuk 2D PC/PC
```

```
% (kombinasi TS dan WH)
```

```
% Inisialisasi
```

```
p1=3; % p1 : bilangan prima untuk melakukan proses time-spreading
```

```
p2=p1; % p2 : bilangan prima untuk melakukan proses wavelength-hop
```

```
w=p2; % w : banyaknya bit "1" pada kode yang valid
```

```
jum_data=5;
```

```
data_user{1}=randuni(jum_data);
```

```
for m=2:(p1*(p2-1))
```

```
    data_user{m}=data_user{1}; % Misal data semua user sama
```

```

end;

data_tamp=data_user{1}; % data tiap user sama

data_user_ulang=[];

for m=1:(p1*(p2-1))

    simp=data_user{m};

    sig_ulang=ulang(p1,simp);

    data_user_ulang=[data_user_ulang;sig_ulang];

end;

% Proses pembentukan baris prima untuk time-spreading

brs_prima_time=baris_prima(p1);

% Proses pembentukan kode prima untuk time-spreading

kode_prima_time=kode_prima(brs_prima_time);

% Proses pembentukan baris prima untuk wavelength-hop

brs_prima_wavelength=baris_prima(p2);

```

```

brs_prima_wavelength=brs_prima_wavelength(2:end,:)+1;

% Untuk mengacak urutan kode penebar

f=1:p1;

urut_kode=acak_urut(f);

for h=1:p1

    kode_prima_time_acak(h,:)=kode_prima_time(urut_kode(h),:);

end;

% Menyisipkan frekuensi

x1=[];

y=brs_prima_wavelength;

x=kode_prima_time_acak;

for gg=1:size(y,1)

    y1=y(gg,:);

    for m=1:size(x,1)

        x_temp=x(m,:);

```

```

n=1;

for p=1:length(x_temp)

    if x_temp(p)==1

        x_temp(p)=y1(n);

        n=n+1;

    end;

end;

x1=[x1; x_temp];

end;

end;

kode_user_final=x1;

% Ubah urutan kode_user_final

var_indeks=1:p1*(p2-1);

var_modif=acak_urut(var_indeks);

for m=1:size(kode_user_final,1)

```

```

    kode_user_fin_modif(m,:)=kode_user_final(var_modif(m),:);

end;

% Modulasi semua data user dengan kode masing2x

% (dalam hal ini dikalikan)

data_tx=[];

for m=1:size(kode_user_final,1)

    simp=kode_user_final(m,:);

    data_rep=data_user_ulang(m,:);

    for k=1:jum_data

        data_termod_kali((k-1)*p1.^2+1:k*p1.^2)=kali(data_rep((k-
1)*p1.^2+1:k*p1.^2),simp);

    end;

    data_tx=[data_tx;data_termod_kali];

end;

sig_out=zeros(p1*(p2-1),jum_data*p1.^2*141);

for n=1:size(data_tx,1)

```

```

temp=[];

simpan=data_tx(n,:);

for k = 1: length(simpan)

    if simpan(k)==0

        temp=[temp sinyal_nol(simpan(k))];

    elseif simpan(k)~=0

        temp=[temp sinyal_taknol(simpan(k))];

    end;

end;

sig_out(n,:)=temp;

end;

sig_gbr_1=sig_out(1:3,:);

figure;

for m=1:3

    subplot(3,1,m);plot(sig_gbr_1(m,:));grid;

    xlabel('Waktu');ylabel('Amplitudo sinyal');

    title(['Data user ke -',num2str(m),'', yaitu [' ',num2str(data_tamp),'']);

```

```

end;

sig_gbr_2=sig_out(4:6,:);

figure;

for m=1:3

    subplot(3,1,m);plot(sig_gbr_2(m,:));grid;

    xlabel('Waktu');ylabel('Amplitudo sinyal');

    title(['Data user ke -',num2str(1*3+m),'', yaitu [' ',num2str(data_tamp),'']);

end;

% % Jika p=5;

% sig_gbr_3=sig_out(7:9,:);

% figure;

% for m=1:3

% subplot(3,1,m);plot(sig_gbr_3(m,:));grid;

% xlabel('Waktu');ylabel('Amplitudo sinyal');

% title(['Data user ke -',num2str(2*3+m),'', yaitu [' ',num2str(data_tamp),'']));

% end;

```



```

%

% sig_gbr_4=sig_out(10:12,:);

% figure;

% for m=1:3

% subplot(3,1,m);plot(sig_gbr_4(m,:));grid;

% xlabel('Waktu');ylabel('Amplitudo sinyal');

% title(['Data user ke -',num2str(3*3+m),'', yaitu [' ',num2str(data_tamp),'']);

% end;

%

% sig_gbr_5=sig_out(13:15,:);

% figure;

% for m=1:3

% subplot(3,1,m);plot(sig_gbr_5(m,:));grid;

% xlabel('Waktu');ylabel('Amplitudo sinyal');

% title(['Data user ke -',num2str(4*3+m),'', yaitu [' ',num2str(data_tamp),'']));

% end;

%

% sig_gbr_6=sig_out(16:18,:);

```

```

% figure;

% for m=1:3

% subplot(3,1,m);plot(sig_gbr_6(m,:));grid;

% xlabel('Waktu');ylabel('Amplitudo sinyal');

% title(['Data user ke -',num2str(5*3+m),'', yaitu [' ',num2str(data_tamp),'']);

% end;

%

% sig_gbr_7=sig_out(19:20,:);

% figure;

% for m=1:2

% subplot(2,1,m);plot(sig_gbr_7(m,:));grid;

% xlabel('Waktu');ylabel('Amplitudo sinyal');

% title(['Data user ke -',num2str(6*3+m),'', yaitu [' ',num2str(data_tamp),'']));

% end;

% % Batas jika p1=5

% % jika p=7

% sig_gbr_7=sig_out(19:21,:);

% figure;

```

```

% for m=1:3

% subplot(3,1,m);plot(sig_gbr_7(m,:));grid;

% xlabel('Waktu');ylabel('Amplitudo sinyal');

% title(['Data user ke -',num2str(6*3+m),'', yaitu [' ',num2str(data_tamp),'']);

% end;

%

% sig_gbr_8=sig_out(22:24,:);

% figure;

% for m=1:3

% subplot(3,1,m);plot(sig_gbr_8(m,:));grid;

% xlabel('Waktu');ylabel('Amplitudo sinyal');

% title(['Data user ke -',num2str(7*3+m),'', yaitu [' ',num2str(data_tamp),'']));

% end;

%

% sig_gbr_9=sig_out(25:27,:);

% figure;

% for m=1:3

% subplot(3,1,m);plot(sig_gbr_9(m,:));grid;

```

```

% xlabel('Waktu');ylabel('Amplitudo sinyal');

% title(['Data user ke -',num2str(8*3+m),' , yaitu [' ,num2str(data_tamp),'']]);

% end;

% sig_gbr_10=sig_out(28:30,:);

% figure;

% for m=1:3

% subplot(3,1,m);plot(sig_gbr_10(m,:));grid;

% xlabel('Waktu');ylabel('Amplitudo sinyal');

% title(['Data user ke -',num2str(9*3+m),' , yaitu [' ,num2str(data_tamp),'']]);

% end;

% sig_gbr_11=sig_out(31:33,:);

% figure;

% for m=1:3

% subplot(3,1,m);plot(sig_gbr_11(m,:));grid;

% xlabel('Waktu');ylabel('Amplitudo sinyal');

% title(['Data user ke -',num2str(10*3+m),' , yaitu [' ,num2str(data_tamp),'']]);

```

```

% end;

% sig_gbr_12=sig_out(34:36,:);

% figure;

% for m=1:3

%   subplot(3,1,m);plot(sig_gbr_12(m,:));grid;

%   xlabel('Waktu');ylabel('Amplitudo sinyal');

%   title(['Data user ke -',num2str(11*3+m),' , yaitu [' ,num2str(data_tamp),'']);

% end;

% sig_gbr_13=sig_out(37:39,:);

% figure;

% for m=1:3

%   subplot(3,1,m);plot(sig_gbr_13(m,:));grid;

%   xlabel('Waktu');ylabel('Amplitudo sinyal');

%   title(['Data user ke -',num2str(12*3+m),' , yaitu [' ,num2str(data_tamp),'']);

% end;

```

```

% sig_gbr_14=sig_out(40:42,:);

% figure;

% for m=1:3

% subplot(3,1,m);plot(sig_gbr_14(m,:));grid;

% xlabel('Waktu');ylabel('Amplitudo sinyal');

% title(['Data user ke -',num2str(13*3+m),'', yaitu [' ',num2str(data_tamp),'']);

% end;

% % batas jika p=7

% mencari nilai korelasi silang

indeks_korelasi=1;

for m=1:size(kode_user_final,1)

    for n=m+1:size(kode_user_final,1)

        korelasi(indeks_korelasi)=0;

        for kk=1:p1^2

            if ((kode_user_final(m,kk)~=0)&&(kode_user_final(n,kk)~=0))

                if (kode_user_final(m,kk)==kode_user_final(n,kk))

                    korelasi(indeks_korelasi)=korelasi(indeks_korelasi)+1;

```

```
    else
        korelasi(indeks_korelasi)=korelasi(indeks_korelasi);
    end;
end;
end;
end;
indeks_korelasi=indeks_korelasi+1;
end;
end;
figure;
stem(korelasi);grid;title(['Korelasi silang',', yaitu [' ,num2str(data_tamp),']]');
xlabel('Indeks kode');ylabel('Nilai korelasi silang');
```

## Function 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);

if cek == 0

    error('Bilangan yang dimasukkan BUKAN bilangan prima');

end;

p = masuk; % bilangan prima yang diinputkan

for m = 1 : p
```



```
for n=1:p
    S(m,n) = mod ((m-1).*(n-1),p);
end;
end;
keluar=S;
```

## Function 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)

        keluar(m,(n-1).*size(masuk,2)+temp(n)+1)=1;

    end;

end;
```

## Function Acak Urut

```
function [keluar]=acak_urut(masuk)

% Function ini untuk mengacak urutan yang masuk

keluar=[];

m=length(masuk);

while m~=0

    masuk_baru=[];

    cek=randperm(length(masuk));

    cek=cek(1);

    keluar=[keluar masuk(cek)];

    for k=1:length(masuk)

        if masuk(k)~=masuk(cek)

            masuk_baru=[masuk_baru masuk(k)];

        end;

    end;

    masuk=masuk_baru;

    m=m-1;

end;
```

## Function Randuni

```
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;

    if (temp<0.5),

        data(1,i)=0;

    else

        data(1,i)=1;

    end;

end;

p=data(1,:);
```

## Function Sinyal Nol

```
function keluar=sinyal_nol(masuk)

%

% Function ini untuk modulasi ASK untuk bit "0"

%

% Variabel masukan : masuk,f(frek)

% Variabel keluaran : keluar

%

t=0.05:.00675:1;

A=1;

if masuk==0

% if masuk~=1

    keluar=zeros(1,length(t));

end;
```

## Function Sinyal tak nol

```
function keluar=sinyal_taknol(masuk)

%

% Function ini untuk modulasi ASK untuk bit"1"

%

% Variabel masukan : masuk,f(frek)

% Variabel keluaran : keluar

%

t=0.05:.00675:1;

A=1;

f=masuk;

if masuk~=0

    keluar=A.*sin(2.*pi.*f.*t);

end;
```

## Function Sinyal satu

```
function keluar=sinyal_satu(masuk,f)

%

% Function ini untuk modulasi ASK untuk bit"1"

%

% Variabel masukan : masuk,f(frek)

% Variabel keluaran : keluar

%

t=0.05:.00675:1;

A=1;

if masuk~=0

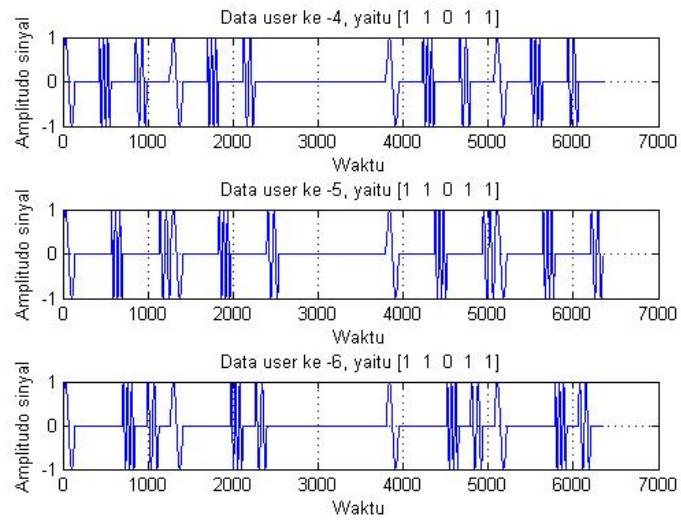
    keluar=A.*sin(2.*pi.*f.*t);

end;
```

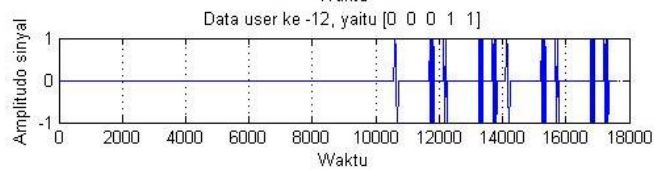
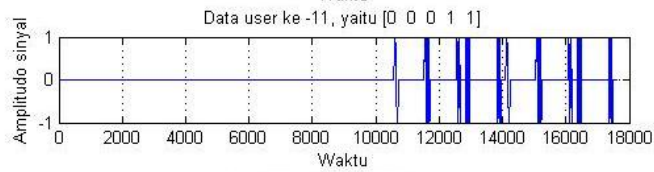
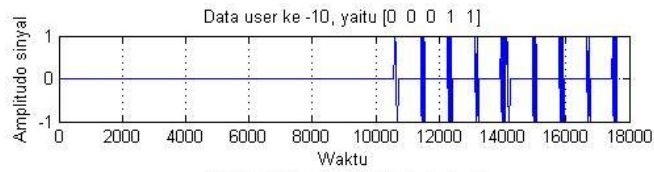
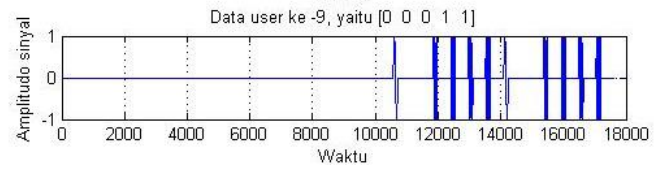
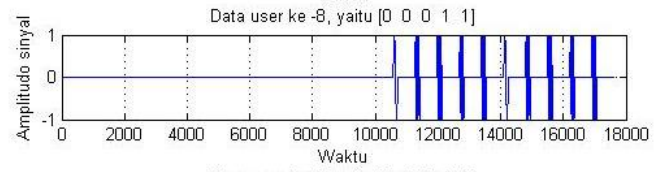
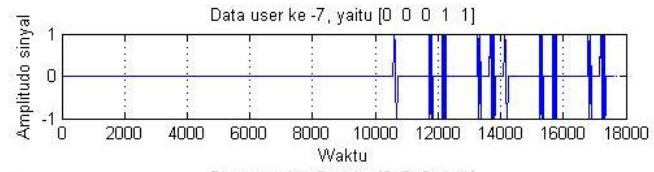
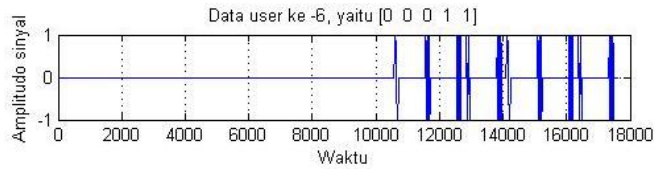
**LAMPIRAN B:**  
**GAMBAR**

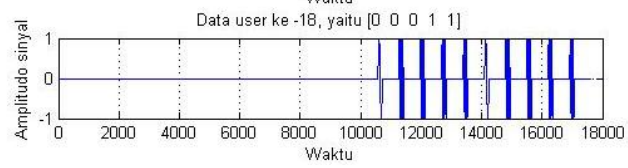
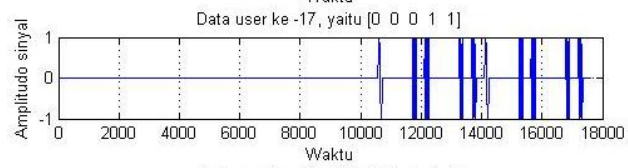
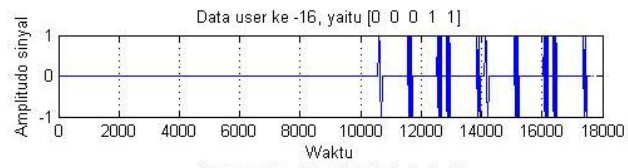
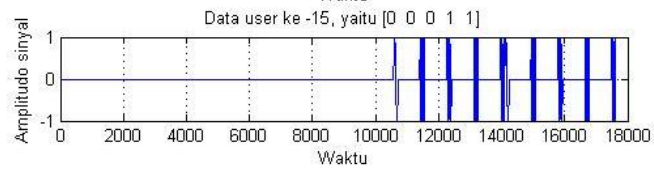
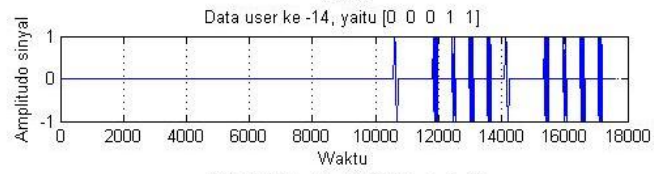
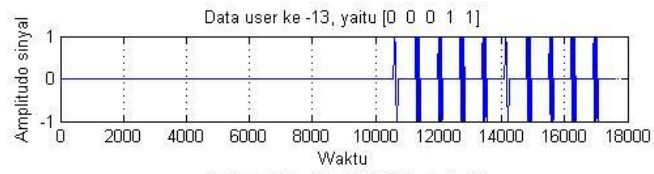


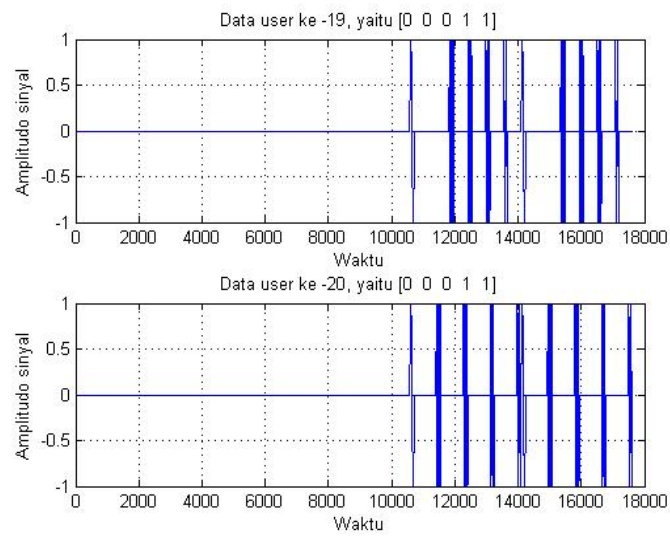
## GAMBAR



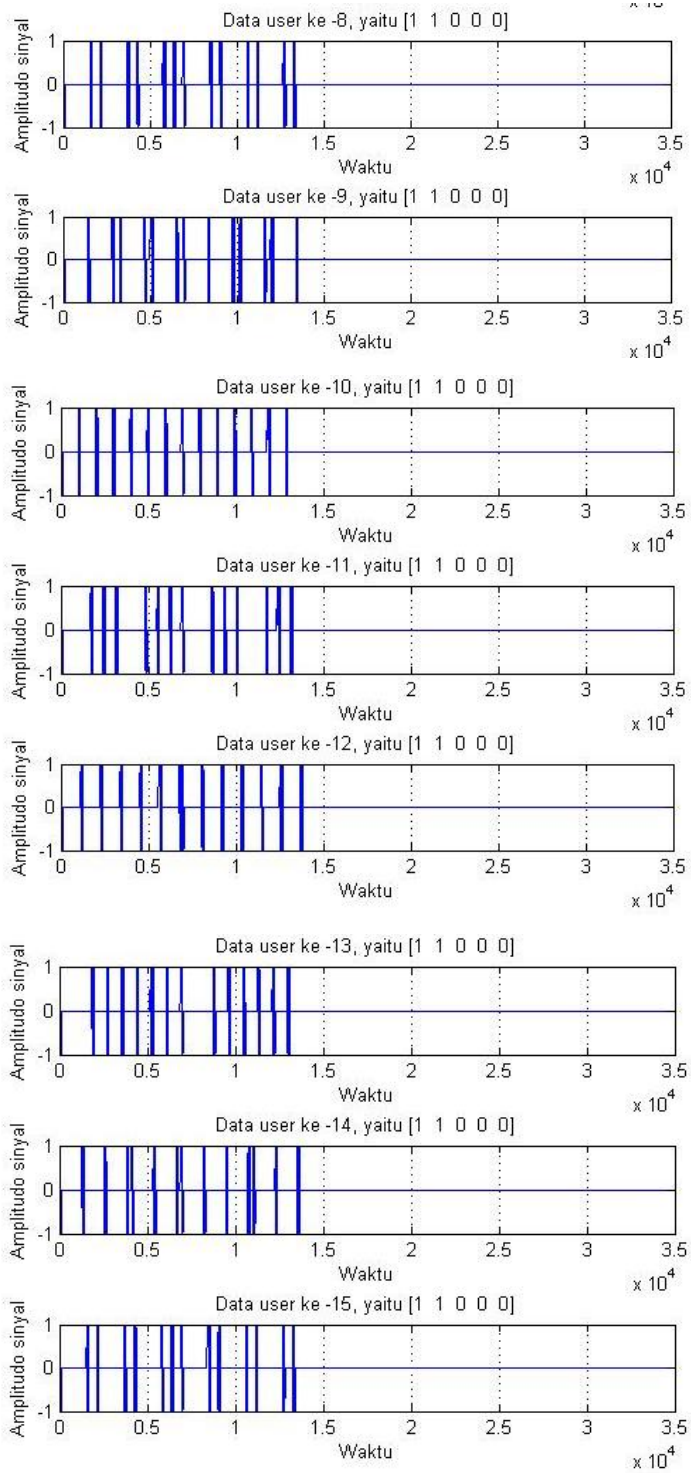
**Gambar B.1** Pengiriman Data(1 1 0 1 1) P=3 untuk user 4 sampai user 6

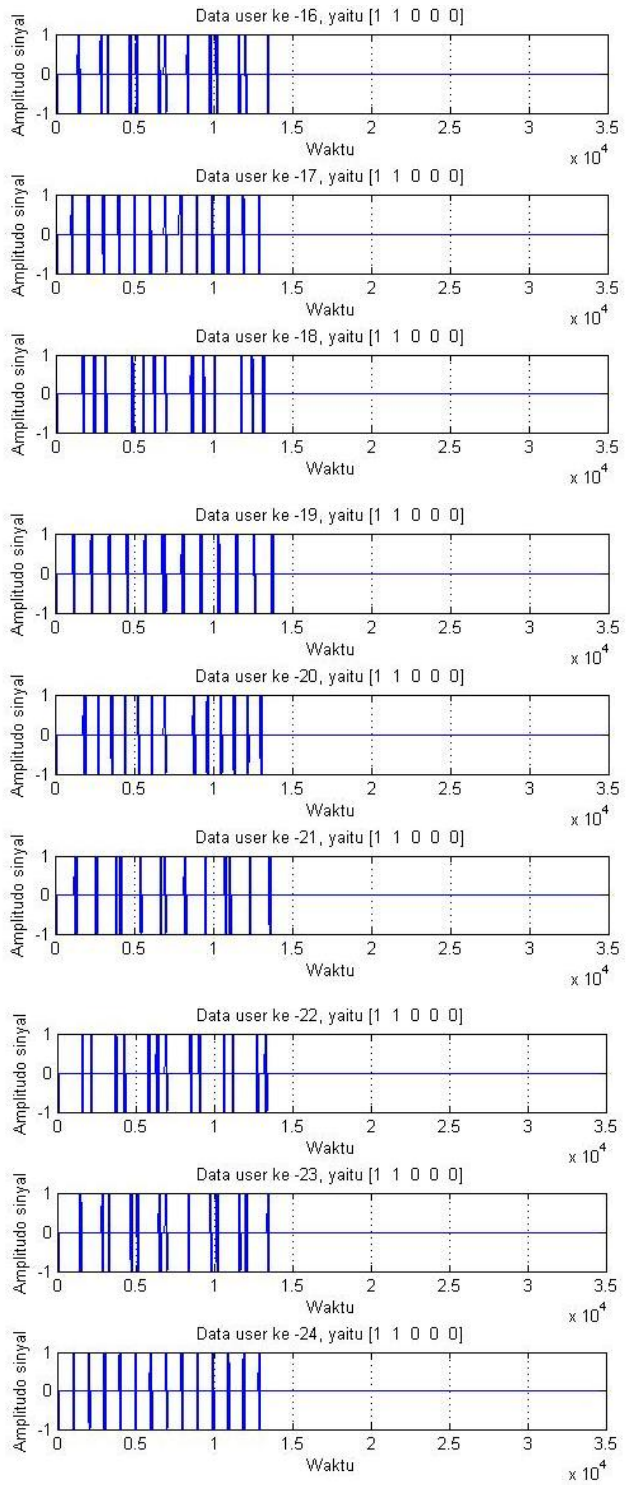


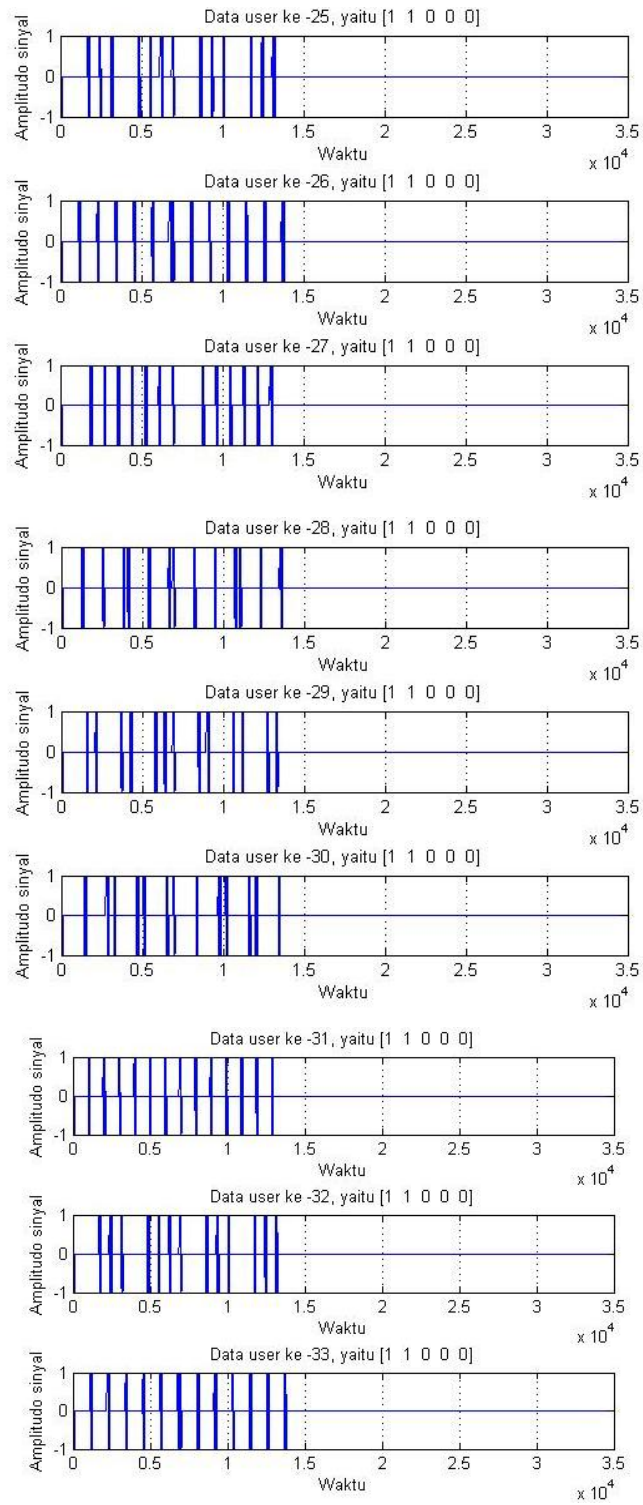


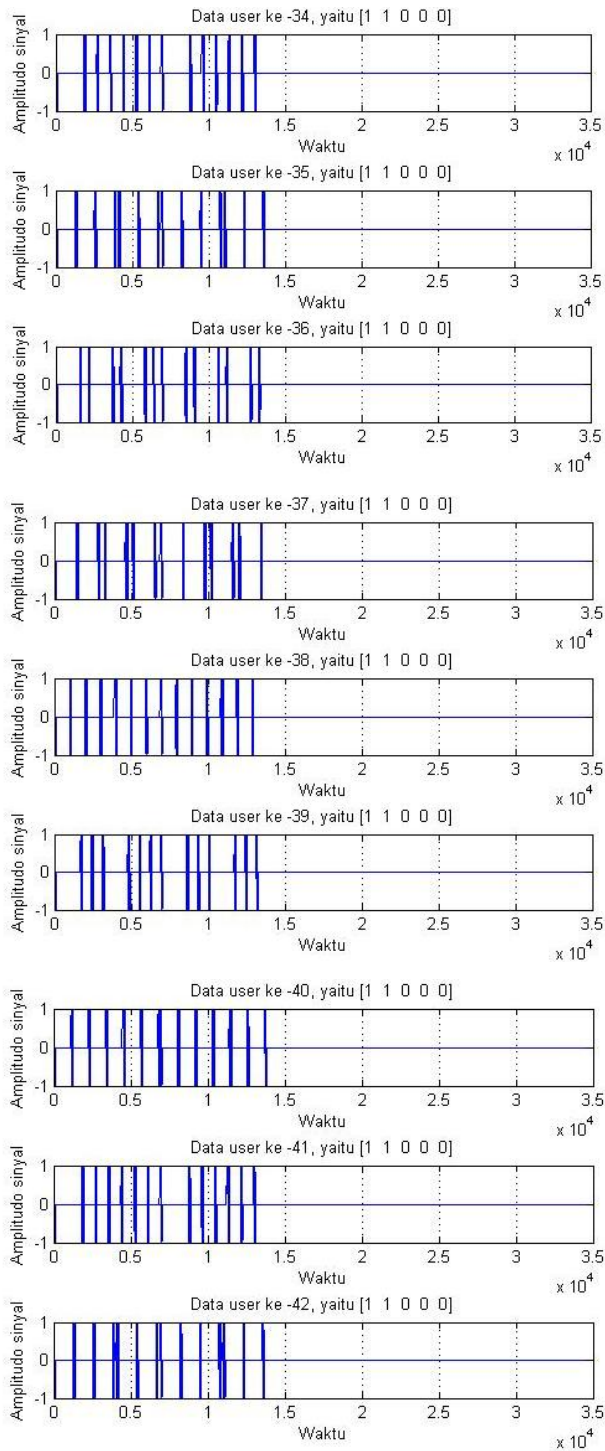


**Gambar B.2** Pengiriman Data(0 0 0 1 1) P=5 untuk user 6 sampai user 20









**Gambar B.3** Pengiriman Data(1 1 0 0 0) P=7 untuk user 8 sampai user 42