

# PERCOBAAN 1

## ENCODER DAN DECODER PCM

### (Simulasi Menggunakan Software Matlab)

#### 1. Tujuan Percobaan:

Setelah melakukan percobaan ini, diharapkan mahasiswa dapat menjelaskan secara praktis proses konversi sinyal analog menjadi sinyal PCM melalui proses PCM encoding dan kembali menjadi sinyal analog melalui proses PCM decoding.

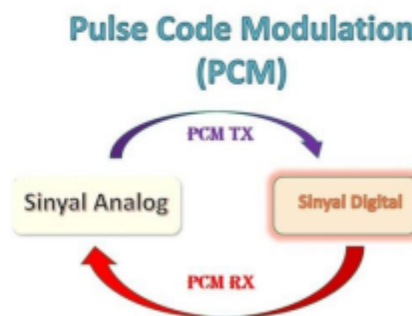
#### 2. Teori Penunjang :

PCM : Pulse Code Modulation, percobaan ini akan memperkenalkan tentang PCM encoder. Percobaan ini akan menghasilkan sinyal output berupa sinyal PCM dari input berupa sinyal analog.

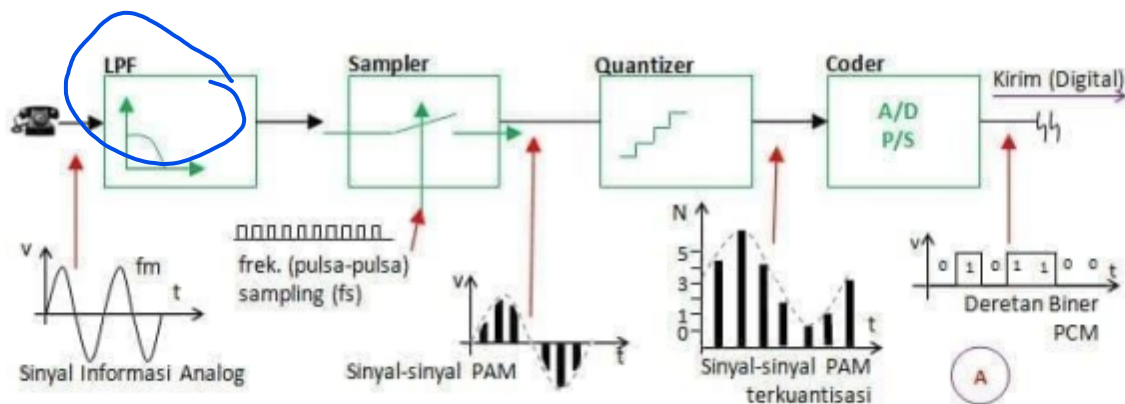
PCM Encoding : input PCM encoder adalah sinyal analog, sedangkan output encoding berupa sinyal sample and hold. Sinyal input harus dibatasi untuk bandwidth dan range amplitud tertentu. Bandwidth maximum yang di ijinan tergantung dari kecepatan sampling yang digunakan berdasarkan criteria Nyquist. Sedangkan range amplitud disesuaikan dengan tampilan yang memungkinkan.

PCM Decoding : input PCM decoding berupa sinyal PCM, sedangkan outputnya akan serupa dengan sinyal input dari sumber. Kalau toh terjadi distorsi, hal ini disebabkan karena proses kuantisasi dari encodernya

#### Proses PCM :



Gambar 1.1 Encoder dan Decoder PCM

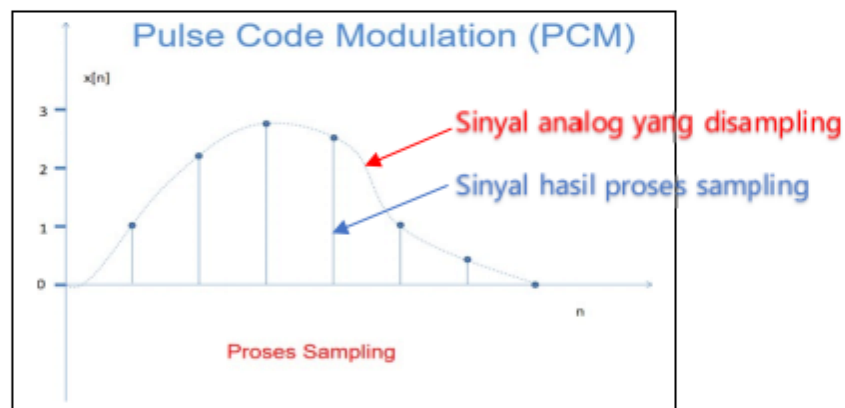


Gambar 1.2 Proses PCM

Berdasarkan blok diagram proses konversi sinyal analog menjadi sinyal PCM pada gambar 1.2, maka dapat diuraikan seperti proses berikut ini :

## 1. Proses Sampling

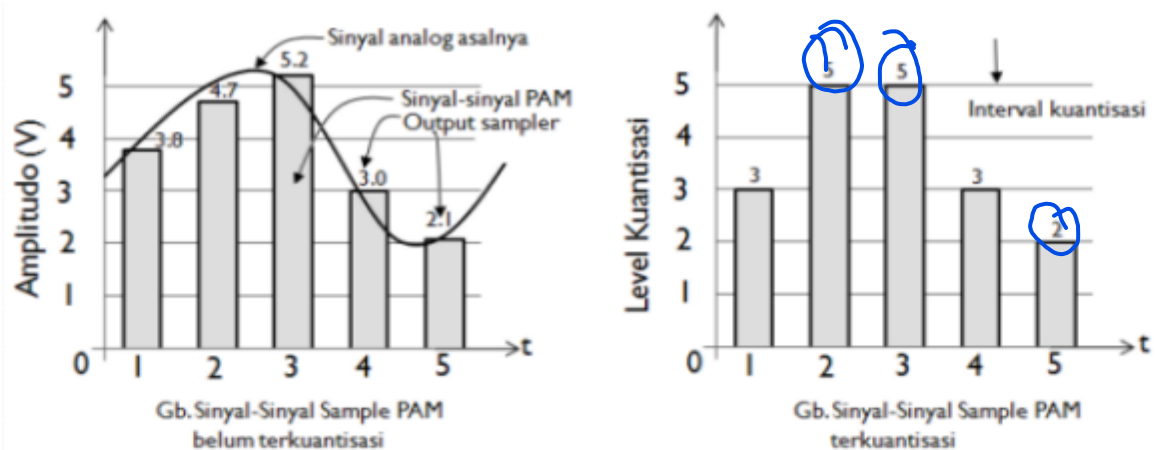
Proses sampling adalah proses yang mengubah sinyal analog menjadi sinyal diskrit agar sesuai dengan format digital. Proses pengambilan sample atau contoh besaran sinyal analog pada titik tertentu secara teratur dan berurutan. Frekuensi sampling harus lebih besar dari 2x frekuensi yang disampling atau sekurang-kurangnya memperoleh puncak dan lembah berdasarkan Theorema Nyquist. Hasil proses sampling berupa sinyal PAM (Pulse Amplitude Modulation), seperti terlihat pada gambar 1.3.



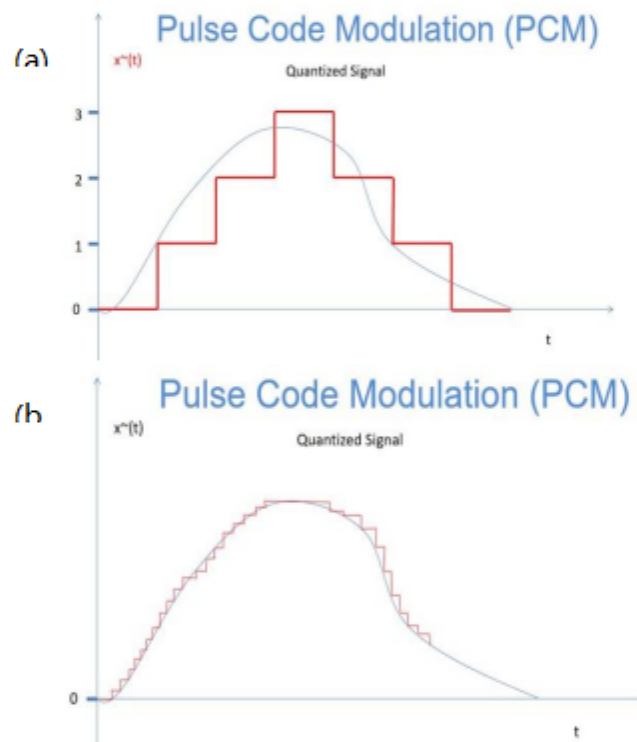
Gambar 1.3 Proses Sampling

## 2. Proses Kuantisasi

Kuantisasi Kuantisasi adalah suatu tahap pemetaan dari sinyal yang telah disampling menjadi sinyal dengan level tertentu berdasarkan bit ADC yang digunakan. Amplitudo dari masing masing sample dinyatakan dengan nilai integer dari level kuantisasi yang terdekat sesuai pada gambar 1.4.



Gambar 1.4 Proses Kuantisasi

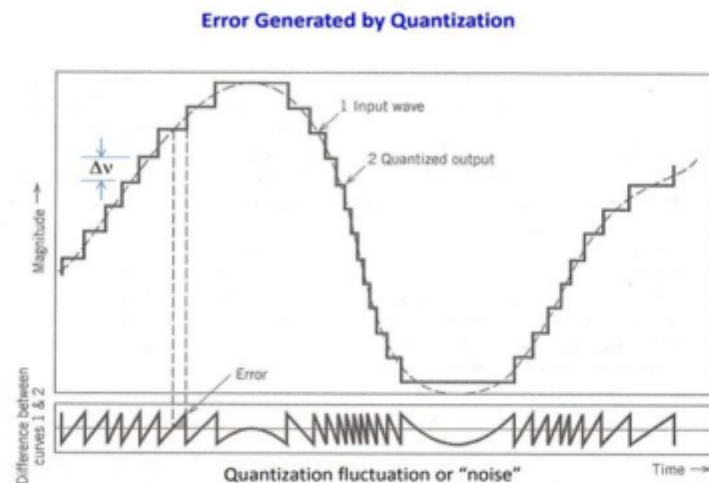


Gambar 1.5 Sinyal hasil kuantisasi

Pada gambar 1.5 merupakan sinyal hasil kuantisasi yang tampak berbeda, perbedaan tersebut dikarenakan pada proses kuantisasi menggunakan level kuantisasi yang berbeda. Dimana level kuantisasi akan menyebabkan jumlah bit kode dari sinyal yang di hasilkan menjadi berbeda.  $N$  adalah jumlah level sinyal (level kuantisasi) yang di ambil dan  $n$  adalah jumlah bit yang digunakan untuk mengkodekan satu sinyal terkuantisasi, maka  $N = 2^n$ . Misalkan sinyal-sinyal terkuantisasi tersebut akan dikodekan menjadi 4 bit maka jumlah level kuantisasi yang akan diperoleh sebesar  $N = 2^4 = 16$ .

Yang perlu diperhatikan pada saat pemilihan level kuantisasi harus disesuaikan dengan bentuk sinyal yang akan di kuantisasi karena setiap level kuantisasi yang ditentukan pada proses kuantisasi akan menghasilkan error kuantisasi. Pilihlah level kuantisasi yang

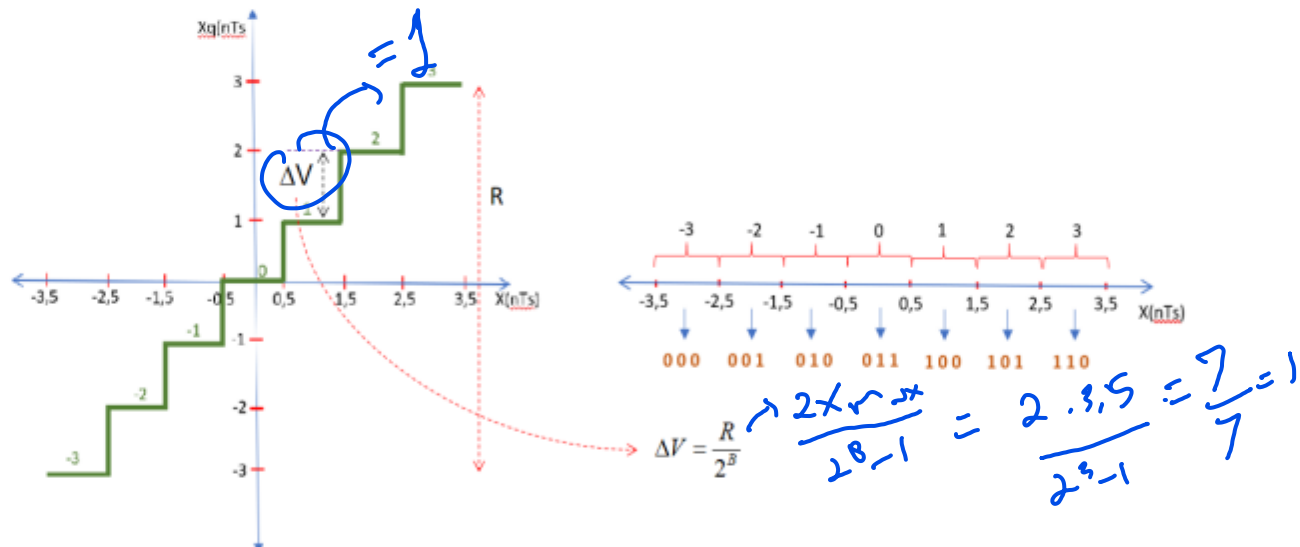
menyebabkan error kuantisasi sekecil mungkin. Munculnya error pada proses kuantisasi dapat dilihat seperti gambar 1.6.



Gambar 1.6 Tampilan error kuantisasi

### Metode Kuantisasi :

Banyak metode yang dapat digunakan untuk proses kuantisasi, yang mana masing-masing metode mempunyai kelebihan dan kekurangan yang berbeda beda. Sebagai contoh metode kuantisasi yang banyak digunakan adalah metode "Mid Tread Quantization". Metode inilah yang akan digunakan untuk proses kuantisasi pada simulasi menggunakan matlab.

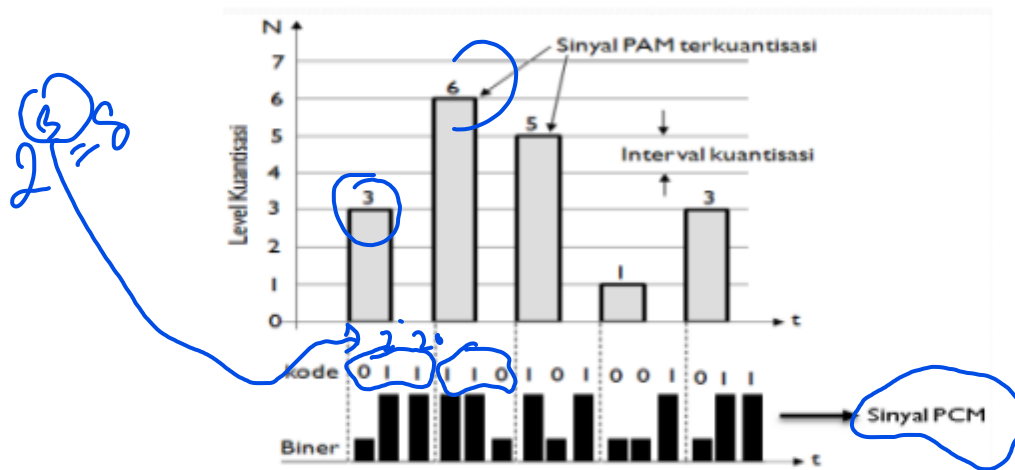


Gambar 1.7 Ilustrasi Mid Tread Quantization

Kuantisasi pada gambar 1.7 untuk mendapatkan sinyal PCM sebanyak 3 bit per sample, jadi jumlah level kuantisasi yang digunakan adalah 8.

### 3. Encode (pengkodean)

Pengkodean merupakan proses mengubah suatu besaran tertentu kedalam bentuk lain yang dikenali berdasarkan ketentuan yang teratur. Pengkodean NRZ adalah suatu pengkodean dimana sinyal tidak kembali ke 0 volt di tengah-tengah bit. Dalam pengkodean NRZ, bit 0 direpresentasikan oleh sinyal dengan tegangan 0 volt, sedangkan bit 1 direpresentasikan oleh sinyal dengan tegangan +V volt. Seperti dapat dilihat pada gambar 1.7.



Gambar 1.7 Proses Pengkodean

### 4. Decode (pengkodean kembali)

Proses decoding berada pada sisi terima, dimana proses ini akan merubah sinyal PCM menjadi sinyal analog kembali. Pada prakteknya sebelum mendapatkan sinyal analog, maka diperlukan proses filtering yaitu untuk mendapatkan sinyal dengan frekuensi yang sama dengan sinyal analog yang di inputkan. Sedangkan filter yang digunakan bisa beragam sesuai dengan frekuensi sinyal yang di kehendaki (LPF, HPF, BPF dsb).

### PROSEDUR PERCOBAAN :

Buatlah program menggunakan software Matlab pada m-file, kemudian ikuti Langkah-langkah simulai sebagai berikut :

Pada PCM TX

1. Membangkitkan sinyal sinusoida( $x(t)$ )

$$x(t) = A \sin(2\pi f t)$$

Amplitudo      Frekuensi      Waktu

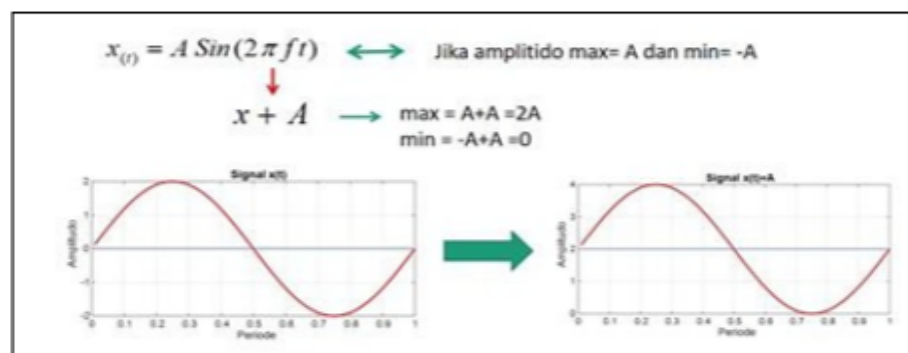
$A$  max  
 $-A$  min

2. Sampling sinyal sinusoida ( $x(t)$ ) dengan menggunakan teori sampling Nyquist

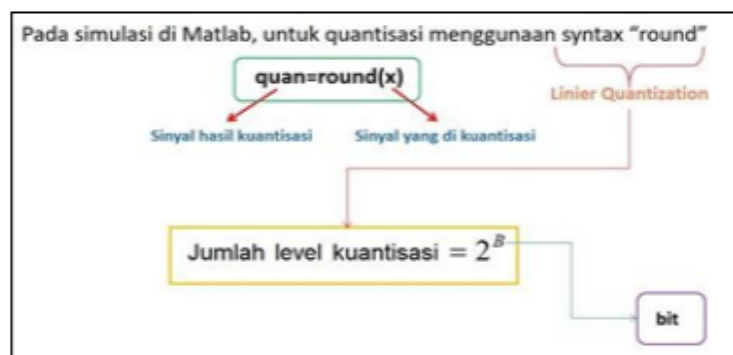
$$f_s \geq 2f_m$$
$$f_s = 20 \times f_m \rightarrow T_s = \frac{1}{f_s}$$

Syarat Nyquist

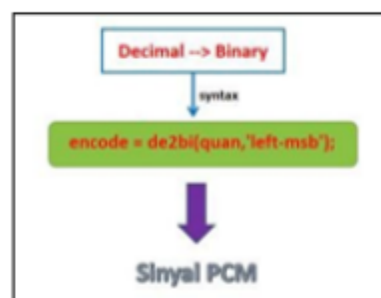
3. Level Shifting (menggeser amplitudo)



4. Proses Kuantisasi

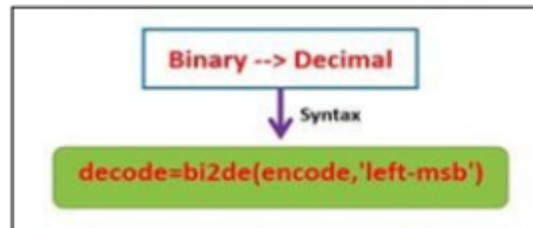


5. Encoding (Proses Pengkodean)

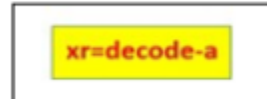


Pada PCM RX

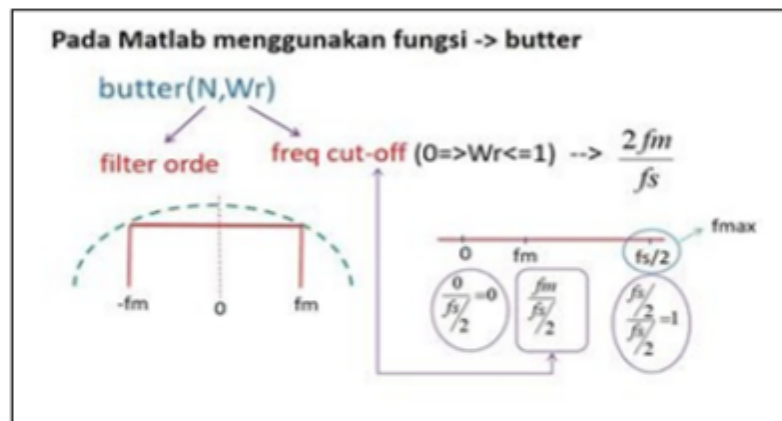
1. Decoding (Proses Pengkodean Kmebali)



2. Shift Amplitudo



3. Smoothing Menggunakan Filter



4. Tampilkan semua gambar hasil running program sebagai laporan sementara dan beri penjelasan semua hasil yang diperoleh dari setiap langkah percobaan.

**TUGAS (DIKERJAKAN DAN HASILNYA DITAMBAHKAN SEBAGAI LAPORAN SEMENTARA)**

1. Buat program proses qantisasi menggunakan methode "Mid-Tread Quantization" dengan jumlah level quantisasi  $\pm 2^3$  (seperti gambar 1.7 ).
2. Buat program pengembangan (pada mfile yg sama pada tugas point 1) dengan merubah jumlah quantisasi sebesar  $\pm 2^4$  (kembangkan metode MTQ nya untuk  $n=4$ )



**Contoh listing program matlab dengan proses quantisasi menggunakan fungsi “round”**

**%PCM Transmitter (Tx)**

```
%Sinyal Analog (sinyal informasi)
f=2; %Frek sinyal informasi
fs=40*f; %Frekuensi sampling t=0:1/fs:1; %Time
a=2; %Amplitudo sinyal
x=a*sin(2*pi*f*t); %Sinyal informasi
%Shifting Amplitudo (Level shifting
x1=x+a;
%Proses Quantisasi
quan=round(x1);
quan1=quan-a;
encode=de2bi(quan,'left-msb');
xe=encode;
```

**%PCM Receiver**

```
decode=bi2de(encode,'left-msb');
%Shifting Level sinyal ke level sinyal asli
xr=decode-a;
%Low Pass Filtering
[num den]=butter(5,4*f/fs); % Butterworth LPF
rec_op=filter(num,den,xr); % Smoothing
```

**%Plotting**

```
figure(1)
subplot(211)
plot(t,x,'r','linewidth',2)
xlabel('Time')
ylabel('Amplitudo')
title('Sinyal Sinus')
grid on
subplot(212)
stem(t,x,'g','linewidth',2)
xlabel('Time')
ylabel('Amplitudo')
title('Sinyal Sampling')
grid on

figure(2)
plot(t,x,'r',t,quan1,'b','linewidth',2)
xlabel('Time')
ylabel('Amplitudo')
title('Sinyal Hasil Quantisasi')
legend('Sinyal Asli','Sinyal Quantisasi')
grid on
```



### **% Plotting (continued)**

```
figure(3)
subplot(211)
plot(t,x,'r',t,quan1,'b','linewidth',2)
xlabel('Time')
ylabel('Amplitudo')
title('Sinyal Hasil Quantisasi')
legend('Sinyal Asli','Sinyal Quantisasi')
grid on
subplot(212)
plot(t,x'-quan1,'m')
xlabel('Time')
ylabel('Amplitudo')
title('Error Quantization')
grid on

figure(5)
plot(t,x,'r',t,rec_op,'g','linewidth',2)
xlabel('Time')
ylabel('Amplitudo')
title('Sinyal Hasil Rekonstruksi')
legend('Sinyal Asli','Sinyal Rekonstruksi')
grid on
```

```
figure(4)
l=encode(:,1);
m=encode(:,2);
n=encode(:,3);
subplot(4,1,1)
stairs(t,xl,'linewidth',2)
title('Sinyal PCM 3bit')
grid on
subplot(4,1,2)
stairs(t,l,'r','linewidth',2);
title('Sinyal PCM bit 1')
grid on
subplot(4,1,3)
stairs(t,m,'b','linewidth',2);
title('Sinyal PCM bit 2')
grid on
subplot(4,1,4)
stairs(t,n,'m','linewidth',2);
title('Sinyal PCM bit 3');
xlabel('Time')
grid on
```

**---oo0 Selesai 0oo---**