MODUL
PEMBANGKITAN SINYAL

I. Tujuan

1. Peserta mengerti cara membangkitkan sinyal menggunakan Matlab
2. Peserta dapat membangkitkan nada dasar menggunakan Matlab

II. Perangkat Yang Diperlukan

- 1 set PC yang dilengkapi dengan software Matlab

III. Langkah-Langkah Percobaan

3.1 Pembangkitan Sinyal Waktu Kontinu

a) Pembangkitan sinyal sinus (sinyal periodic)

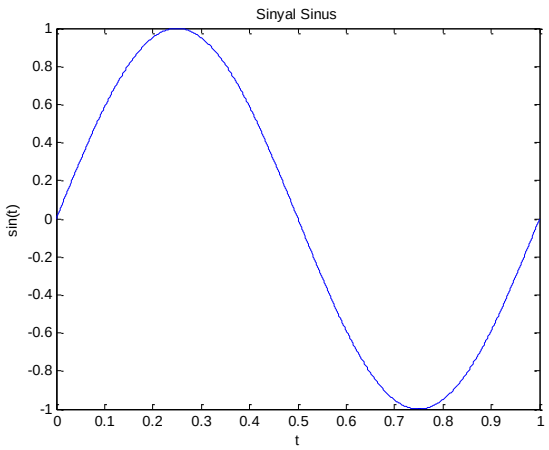
Sinyal sinus memiliki bentuk umum:

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

dimana A= Amplitudo, ω= perode, f=frekuensi, dan θ adalah sudut phase.
Untuk membangkitkan sinyal sinus, buat program berikut

```
fs=1000;      % frekuensi sampling 1kHz
t=0:1/fs:1;   % untuk 1 detik
A=1;          % amplitudo 1
f=1;          % frekuensi 1Hz
phase=0;      % sudut fase 0
x=A*sin(2*pi*f*t + phase);
plot(t,x)
xlabel('t')
ylabel('sin(t)')
title('Sinyal Sinus')
```

simpan dan jalankan, maka akan tampil grafik berikut:



Sinyal yang terbangkit adalah sebuah sinus dengan amplitudo A= 1, frekuensi f = 1 Hz dan fase awal θ = 0. Diharapkan anda sudah memahami tiga parameter dasar pada sinyal sinus ini.

Untuk lebih memahami pengaruh perubahan tiga parameter tersebut, coba lanjutkan program Saudara dengan mengubah-ubah ketiga parameter tersebut untuk beberapa nilai (minimal 3 nilai). Kemudian perhatikan apa yang terjadi dan beri komentar Anda terhadap perubahan yang terjadi. (Note: untuk melihat pengaruh perubahan suatu parameter, misal parameter Amplitudo, anda dapat mengubah-ubah niali Amplitudo tersebut pada beberapa nilai, sedangakan parameter yang lain tetap/tidak diubah)

Akan lebih efektif jika program dibuat dalam bentuk function (pada modul ini akan diberi contoh membuat function untuk membangkitkan fungsi sinus, selanjutnya diharapkan Saudara bisa membuat function sendiri).

```
function []=sinus(A, f, phase)
fs=1000;     % frekuensi sampling 1kHz
t=0:1/fs:1; % untuk 1 detik
x=A*sin(2*pi*f*t + phase);
plot(t,x);
xlabel('t')
ylabel('sin(t)')
title('Sinyal Sinus')
```

untuk memanggil/menggunakan fungsi tersebut, cukup ketikkan:
>> sinus(2, 2, 0) %membangkitkan fungsi sinus dengan A=2, f=2, dan sudut phase=0

3.2 Pembangkitan Sinyal Waktu Diskrit

a) Pembangkitan sinyal/barisan cuplik satuan atau sample unit

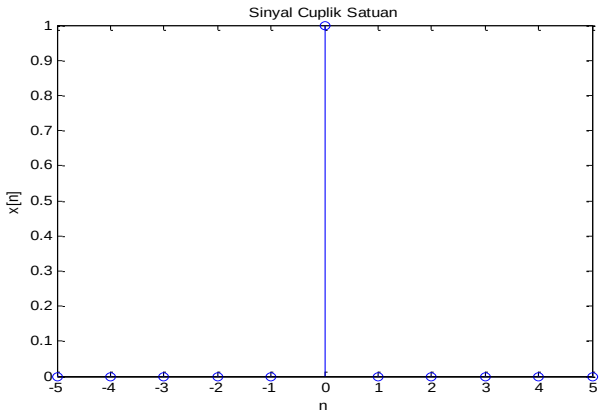
Barisan cuplik satuan memiliki bentuk umum:

$$\delta(n) = \begin{cases} 1, n = 0 \\ 0, n \neq 0 \end{cases} \text{ atau } \quad \delta(n-k) = \begin{cases} 1, n = k \\ 0, n \neq k \end{cases}$$

untuk membangkitkan barisan cuplik satuan, ketikkan program berikut:

```
k=0;
n1=-5;
n2=5;
n =[n1:n2];
x =[(n-k) == 0];  % akan bernilai 1 saat(n-k)=0, selain itu berniali 0
stem(n,x);
title('Sinyal Cuplik Satuan')
xlabel('n')
ylabel('x[n]')
```

hasil plotnya adalah



Coba anda ulangi langkah-langkah program di atas, masukkan 3 nilai k yang berbeda-beda. Anda juga bisa mengubah batas-batas n1 dan n2. Catat apa yang terjadi.

b) Pembangkitan sinyal Langkah Satuan (Unit Step)

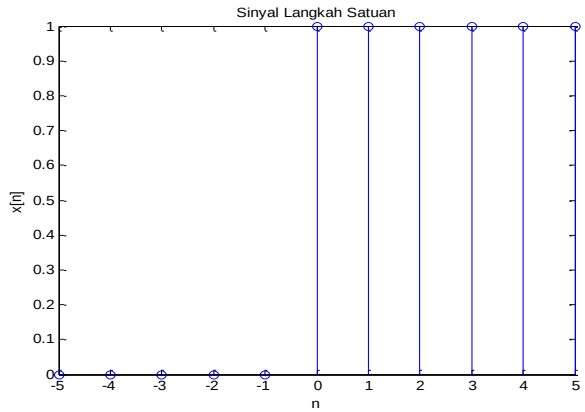
Barisan langkah satuan memiliki bentuk umum:

$$u(n) = \begin{cases} 1, n \geq 0 \\ 0, n < 0 \end{cases} \text{ atau } \quad u(n-k) = \begin{cases} 1, n \geq k \\ 0, n < k \end{cases}$$

untuk membangkitkan barisan langkah satuan, ketikkan program berikut:

```
k=0;
n1=-5;
n2=5;
n =[n1:n2];
x =[(n-k) >= 0];
stem(n,x);
title('Sinyal Langkah Satuan')
xlabel('n')
ylabel('x[n]')
```

hasil running program diatas adalah:



Coba anda ulangi langkah-langkah program di atas, masukkan tiga nilai k yang berbeda-beda. Anda juga bisa mengubah batas-batas n1 dan n2. Catat apa yang terjadi.

c) Pembangkitan sinyal eksponensial

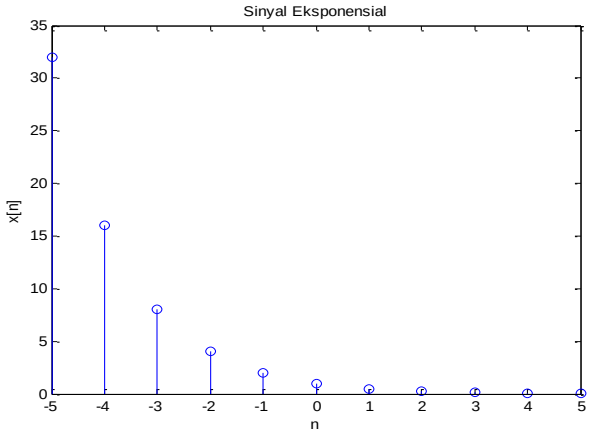
Sinyal eksponensial didefinisikan sebagai berikut:

$$x(n) = a^n$$

untuk membangkitkan barisan eksponensial, ketikkan program berikut:

```
a=0.5
n1=-5;
n2=5;
n =[n1:n2]
x=a.^n
stem(n,x);
title('Sinyal Eksponensial')
xlabel('n')
ylabel('x[n]')
```

hasil running program diatas adalah:



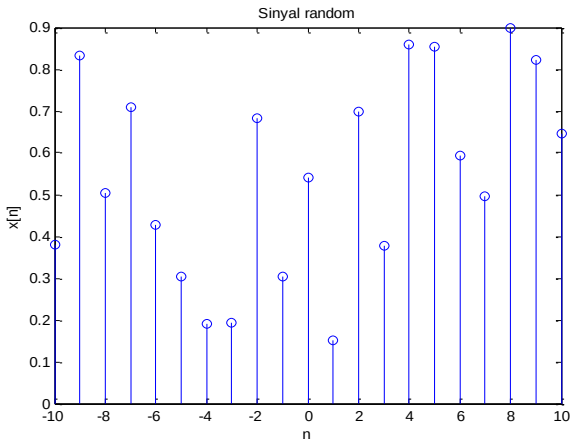
Coba anda ulangi langkah-langkah program di atas, masukkan nilai a yang berbeda-beda, yaitu tiga nilai a untuk a>1, a<1 dan diantaranya. Anda juga bisa mengubah batas-batas n1 dan n2. Catat apa yang terjadi.

d) Pembangkitan sinyal acak

Untuk membangkitkan sinyal acak, ketikkan program berikut:

```
n1=-10;
n2=10;
n =[n1:n2];
p=length(n);
x=rand(1,p);
stem(n,x);
title('Sinyal random')
xlabel('n')
ylabel('x[n]')
```

hasil running program diatas adalah:



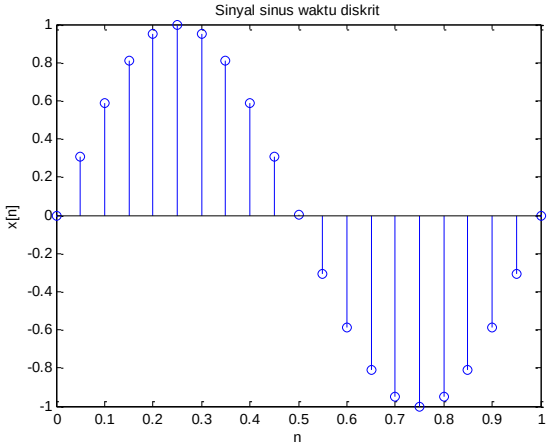
Coba anda ulangi langkah-langkah program di atas, Lakukan perubahan pada nilai Fs, sehingga bernilai 15, 12, 10, 60, 70, dan 80. Catat apa yang terjadi ?

e) Pembangkitan sinyal sinus waktu diskrit

Sinyal sinus merupakan sinyal periodic. Secara umum sifat dasar dari sinyal waktu diskrit memiliki kemiripan dengan sinus waktu kontinyu. Untuk membangkitkan sinyal sinus waktu diskrit.

```
Fs=20;%frekuensi sampling
t=0:1/Fs:1;%proses normalisasi
f=1;
A=1;
x=A*sin(2*pi*f*t);
stem(t,x)
title('Sinyal sinus waktu diskrit')
xlabel('n')
ylabel('x[n]')
```

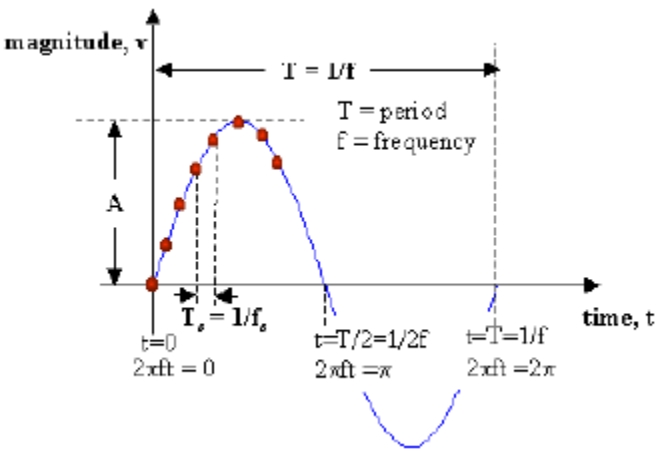
hasil running program diatas adalah:



Coba anda ulangi langkah-langkah program di atas, gantilah nilai Fs, f, dan A untuk beberapa nilai. Beri komentar Anda!

3.3 Pembangkitan Sinyal Suara

Sebelum memulai membuat suara(music), mari kita tinjau kembali sedikit dasar dari sinyal AC. Anggaplah sinyal sinus seperti ditunjukkan oleh gambar.



Sinyal sinus tersebut mempunyai persamaan matematis berikut:

$$v(t) = A * \sin(2*pi*f*t)$$

Dimana: A = amplitudo
f = frekuensi (Hertz)
t = waktu (detik)
T merupakan periode sinyal (dalam detik) dan T=1/f.

Suara (*sound*) terbentuk ketika gelombang diatas digunakan untuk menggetarkan molekul bahan perantara pada frekuensi yang dapat didengar manusia (frekuensi audio) yaitu dalam range 300Hz sampai 3kHz.

Bila ingin membuat gelombang / sinyal seperti pada gambar 4, perlu mengevaluasi persamaan v(t)=A.sin (2*pi*f*t), pada waktu diskrit (t pada gambar 4 ditunjukkan sebagai waktu sesaat dengan titik merah), biasanya pada selang waktu tertentu (jarak antar t).

Selang waktu antar t tersebut, untuk selanjutnya disebut interval waktu, haruslah cukup (seberapa cukup? Ingat kriteria Nyquist) untuk mendapatkan sinyal yang halus. Interval waktu ini disebut sebagai interval sampling atau sampling-rate, dan merupakan proses pencuplikan sinyal kontinyu menjadi sinyal diskrit pada titik-titik yang disebut *sampling.*

Sampling rate, pada gambar diatas adalah Ts, yang merupakan frekuensi sampling, dengan fs =1/Ts. Sebagai contoh, perintah pada Matlab untuk menghasilkan sinyal sinus dengan amplitudo 1, pada frekuensi 466,16Hz (adalah nada A pada tabel oktaf) adalah

```
>>fs=8000;
>>t=0:1/fs:1;
>>v=sin(2*pi*466.16*t);
```

sekarang vektor v berisi sampel dari sinyal sinus pada t=0 sampai t=1, dengan jarak antar sampel adalah 1/8000 detik, merupakan kualitas standar suara audio. Sekarang sinyal sinus tersebut dapat digambarkan ataupun didengarkan.

Untuk menggambarkan sinyal tersebut digunakan perintah:

```
>>plot(v);
```

Untuk mendengarkan vektor v tersebut pada frekuensi sampling 8KHz (default), tentu saja komputer anda harus dilengkapi dengan Sound Card dan Speaker, digunakan perintah:

```
>>sound(v);
```

Untuk menyimpan vektor v dalam format wav kedalam harddisk komputer digunakan perintah:

```
>>wavwrite(v,'namafile');
```

Tabel 1. Standar frekuensi nada-nada pada oktaf dasar

Nada	C	D	E	F	G	A	B	C
Frek(Hz)	262	294	330	349	392	440	494	523

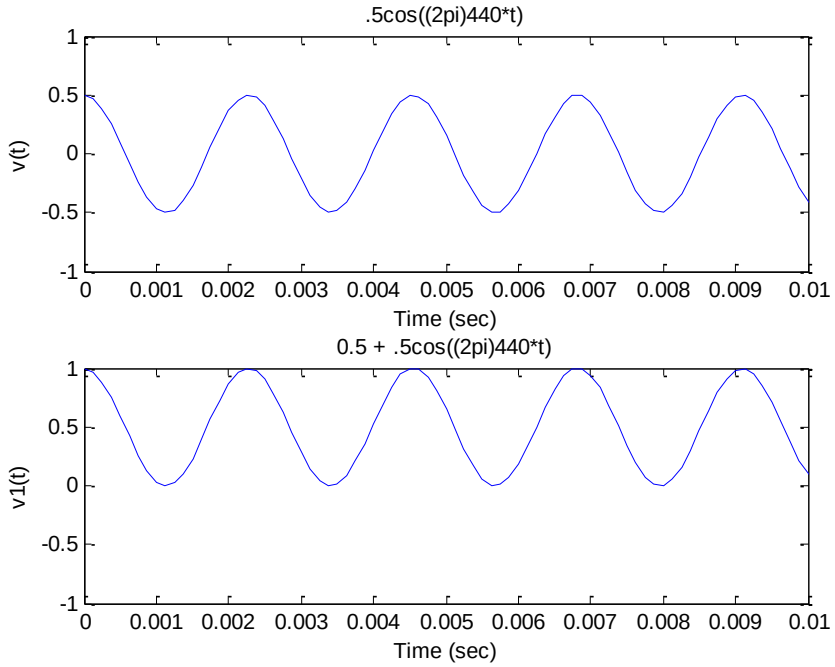
Lakukakan langkah-langkah percobaan sebagai berikut:

1. Buatlah file baru pada Matlab Editor.
2. Membuat nada A pada frekuensi 440Hz. Ketikkan perintah berikut.

```
fs = 8000;                  % frekuensi sampling pada 8KHz
t = 0:1/fs:1;               % panjang tiap nada 1 detik
v = .5*cos(2*pi*440*t);     % nada A
subplot(2,1,1);
plot(t,v);                  % gambarkan sinyal v
axis([0 .01 -1 1])          % atur skala
sound(v,fs);                % bunyikan pada fs 8kHz!
title('.5cos((2pi)440*t)')
xlabel('Time (sec)')
ylabel('v(t)')
```

3. Tambahkan offset DC pada sinyal nada A sebelumnya. Ketiklah perintah tambahan berikut dibawah listing nomor 2. Dan jalankan kembali.

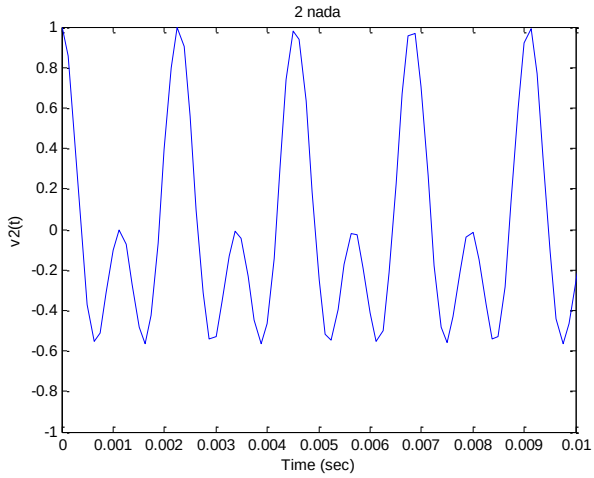
```
pause
v1 = .5+v;      % apa yang terjadi bila ditambah komponen DC
subplot(2,1,2);
plot(t,v1);
axis([0 .01 -1 1])
sound(v1,fs);
title('0.5 + .5cos((2pi)440*t)')
xlabel('Time (sec)')
ylabel('v1(t)')
```



Maka seharusnya akan terdengar nada yang sama karena telinga manusia tidak merespon bias DC. Bila tidak terdengar sama kemungkinan speaker telah saturasi yang menyebabkan sinyal terpotong.

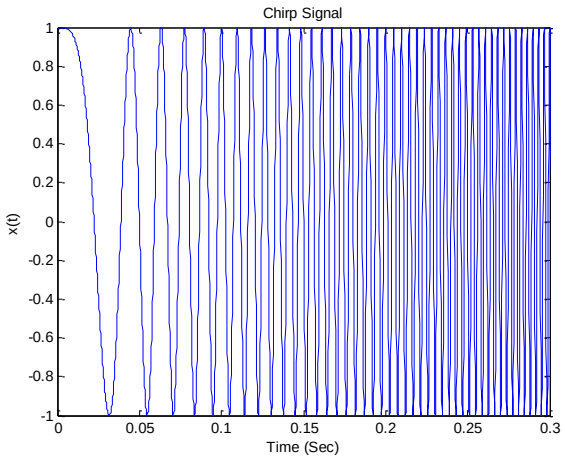
4. Bagaimana bila dua buah nada digabung. Buatlah file baru. Ketikkan perintah baru berikut.

```
fs = 8000;                          % frekuensi sampling pada 8KHz
t = 0:1/fs:1;                       % panjang tiap nada
v = .5*cos(2*pi*440*t);             % nada A
v2 = v+.5*cos(2*pi*880*t);          % ditambahkan nada lain
sound(v2,fs);                       % bunyikan pada fs 8kHz
plot(t,v2);
title('2 nada')
axis([0 .01 -1 1])
xlabel('Time (sec)')
ylabel('v2(t)')
```



5. Membuat suara dengan frekuensi yang semakin naik. Ketikkan perintah baru berikut.

```
fs=8000;                % frekuensi sampling pada 8KHz
t=0:1/fs:2;             % selama 2 detik
f = 500*t;              % frekuensi maksimum 500*2 = 1000Hz
x=cos(2*pi*(f.*t));     % sinyal cos dengan frekuensi makin naik
plot(t,x)
title('Chirp Signal')
xlabel('Time (Sec)')
ylabel('x(t)')
axis([0 .3 -1 1])
sound(x,fs)             % bunyikan pada fs 8kHz
```



6. Buatlah tangga nada dasar dengan mengetikkan perintah baru berikut. Jalankan dan perhatikan hasilnya.

```
fs=8000;              % frekuensi sampling 8kHz
t=0:1/fs:0.25;        % panjang not ¼ detik
c=sin(2*pi*262*t);    % nada c
d=sin(2*pi*294*t);    % nada d
e=sin(2*pi*330*t);    % nada e
f=sin(2*pi*349*t);    % nada f
g=sin(2*pi*392*t);    % nada g
a=sin(2*pi*440*t);    % nada a
b=sin(2*pi*494*t);    % nada b
c1=sin(2*pi*523*t);   % nada C
nol=zeros(size(t));   % spasi kosong
unyil=[b,nol,a,nol,g,nol,a,nol,b,nol,a,nol,a,nol,...
nol,b,nol,a,nol,g,nol,e,nol,g,nol,e,nol,e];
sound(unyil,fs); % mainkan pada fs 8kHz
wavwrite(unyil,'d:\unyil.wav'); % simpan ke file WAV
```

Selamat menikmati ^_^

Buatlah *ringtone* sederhana seperti pada bagian 3.3 nomor 6.

Referensi:

[1] Tri Budi, Miftahul Huda, Modul Praktikum DSP, PENS-ITS, 2001

[2] Hary Oktavianto, Modul Praktikum DSP, PENS-ITS, 2001