

MODUL 4

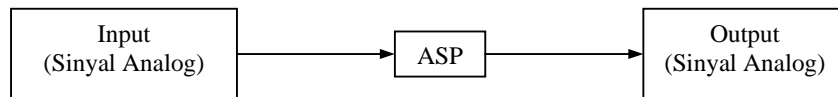
SAMPLING DAN ALIASING

I. TUJUAN

- Siswa memahami pengaruh pemilihan jumlah sample dan pengaruhnya pada proses recovery sinyal

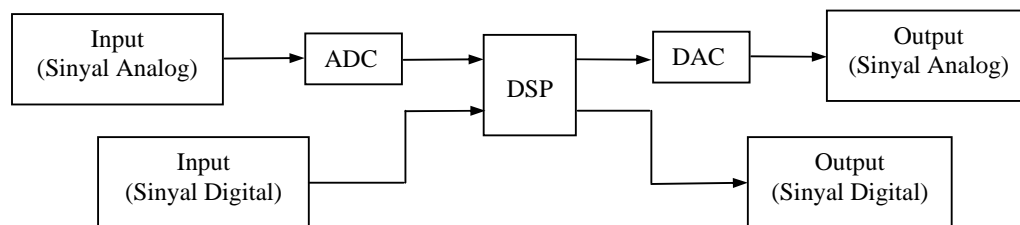
II. DASAR TEORI

Dalam proses pengolahan sinyal analog, sinyal input masuk ke Analog Signal Processing (ASP), diberi berbagai perlakuan (misalnya pemfilteran, penguatan, dsb.) dan outputnya berupa sinyal analog



Gambar 1. Sistem Pengolahan Sinyal Analog

Proses pengolahan sinyal secara digital memiliki bentuk sedikit berbeda. Komponen utama system ini berupa sebuah processor digital yang mampu bekerja apabila inputnya berupa sinyal digital. Untuk sebuah input berupa sinyal analog perlu proses awal yang bernama digitalisasi melalui perangkat yang bernama *analog-to-digital conversion* (ADC), dimana sinyal analog harus melalui proses sampling, quantizing dan coding. Demikian juga output dari processor digital harus melalui perangkat *digital-to-analog conversion* (DAC) agar outputnya kembali menjadi bentuk analog. Ini bisa kita amati pada perangkat seperti PC, digital sound system, dsb. Secara sederhana bentuk diagram bloknya adalah seperti berikut ini.



Gambar 2. Sistem Pengolahan Sinyal Digital

2.1. Sinyal Waktu Diskrit

Berdasarkan pada penjelasan diatas kita tahu betapa pentingnya satu proses yang bernama sampling. Setelah sinyal waktu kontinyu atau yang juga popoler kita kenal sebagai sinyal analog disampel, akan didapatkan bentuk sinyal waktu diskrit. Untun mendapatkan sinyal waktu diskrit yang mampu mewakili sifat sinyal aslinya, proses sampling harus memenuhi syarat Nyquist:

$$f_s \geq 2 f_i \quad (1)$$

dimana:

f_s = frekuensi sinyal sampling

f_i = frekuensi sinyal informasi yang kan disampel

Fenomena aliasing proses sampling akan muncul pada sinyal hasil sampling apabila proses frekuensi sinyal sampling tidak memenuhi criteria diatas.

Perhatikan sebuah sinyal sinusoida waktu diskrit yang memiliki bentuk persamaan matematika seperti berikut:

$$x(n) = A \sin(\omega n + \theta) \quad (2)$$

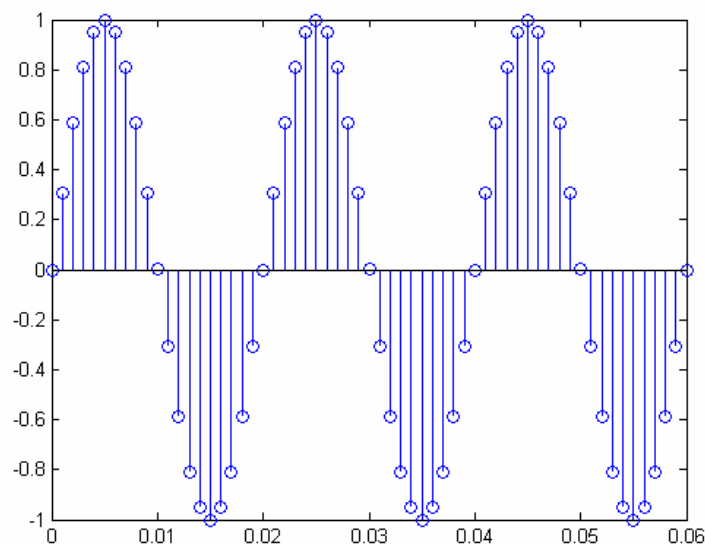
dimana:

A = amplitudo sinyal

ω = frekuensi sudut

θ = fase awal sinyal

Frekuensi dalam sinyal waktu diskrit memiliki satuan radian per indek sample, dan memiliki ekuivalensi dengan $2\pi f$.



Gambar 3. Sinyal sinus diskrit

Sinyal sinus pada Gambar 3 tersusun dari 61 sampel, sinyal ini memiliki frekuensi $f = 50$ dan disampel dengan $F_s = 1000$. Sehingga untuk satu siklus sinyal sinus memiliki sample sebanyak $F_s/f = 1000/50 = 20$ sampel.

Berbeda dengan sinyal waktu kontinyu (C-T), sifat frekuensi pada sinyal waktu diskrit (D-T) adalah:

1. Sinyal hanya periodik jika f rasional. Sinyal periodic dengan periode N apabila berlaku untuk semua n bahwa $x(n+N) = x(n)$. Periode fundamental N_F adalah nilai N yang terkecil.

Sebagai contoh: agar suatu sinyal periodic maka

$$\cos(2\pi(N+n)\theta) = \cos(2\pi n\theta) = \cos(2\pi n\theta + 2\pi k)$$

$$\Leftrightarrow 2\pi fN = 2\pi k \Leftrightarrow f = \frac{k}{N} \Leftrightarrow f \text{ harus rasional}$$

2. Sinyal dengan frekuensi beda sejauh $k2\pi$ (dengan k bernilai integer) adalah identik. Jadi berbeda dengan kasus pada C-T, pada kasus D-T ini sinyal yang memiliki suatu frekuensi unik tidak berarti sinyal nya bersifat unik.

Sebagai contoh:

$$\cos[(\omega_0 + 2\pi)n + \theta] = \cos(\omega_0 n + \theta)$$

karena $\cos(\omega_0 + 2\pi) = \cos(\omega_0)$. Jadi bila $x_k(n) = \cos(\omega_0 n + 2\pi k)$, $k = 0, 1, \dots$. Dimana

$\omega_k = \omega_0 + 2\pi k$, maka $x_k(n)$ tidak bisa dibedakan satu sama lain.

Artinya $x_1(n) = x_2(n) = x_3(n) = \dots = x_k(n)$. Sehingga suatu sinyal dengan frekuensi berbeda akan berbeda jika frekuensinya dibatasi pada daerah $-\pi < \omega < \pi$ atau $-1/2 < f < 1/2$. Diluar itu akan terjadi fenomena aliasing.

III. PERALATAN

- PC yang dilengkapi dengan perangkat multimedia (sound card, Microphone, Speaker active, atau headset)
- Sistem Operasi Windows dan Perangkat Lunak Matlab yang dilengkapi dengan tool box DSP

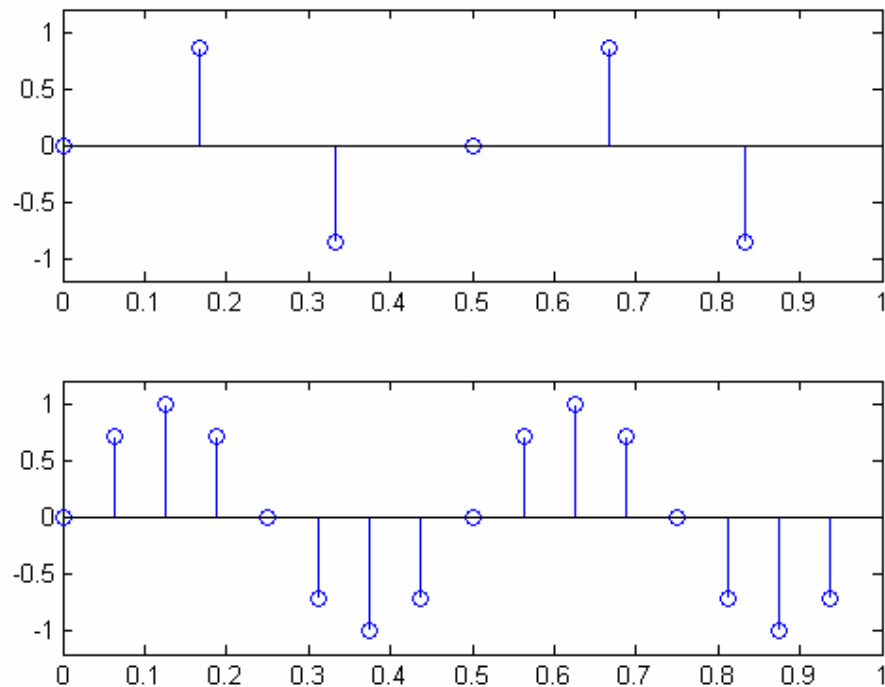
IV. LANGKAH PERCOBAAN

4.1 Pengamatan Pengaruh Pemilihan Frekuensi Sampling Secara Visual

Prosedur yang akan anda lakukan mirip dengan yang ada di percobaan 2, tetapi disini lebih ditekankan pada konsep pemahaman fenomena sampling. Untuk itu anda mulai dengan membuat program baru dengan perintah seperti berikut.

```
%sin_dikrit1.m
Fs=8;%frekuensi sampling
t=(0:Fs-1)/Fs;%proses normalisasi
s1=sin(2*pi*t*2);
subplot(211)
stem(t,s1)
axis([0 1 -1.2 1.2])
```

```
Fs=16;%frekuensi sampling
t=(0:Fs-1)/Fs;%proses normalisasi
s2=sin(2*pi*t*2);
subplot(212)
stem(t,s2)
axis([0 1 -1.2 1.2])
```



Gambar 4. Pengaruh jumlah sample berbeda terhadap satu periode sinyal terbangkit

2. Lakukan perubahan pada nilai F_s , pada sinyal s_1 sehingga bernilai 10, 12, 14, 16, 20, dan 30. Catat apa yang terjadi ? Apa pengaruh jumlah sample berbeda untuk satu periode sinyal terbangkit?

4.2 Pengamatan Pengaruh Pemilihan Frekuensi Sampling pada Efek Audio

Disini kita akan mendengarkan bagaimana pengaruh frekuensi sampling melalui sinyal audio. Untuk itu anda harus mempersiapkan PC anda dengan speaker aktif yang sudah terkoneksi dengan sound card. Selanjutnya anda ikuti langkah berikut.

1. Buat program bari `sampling_2.m` dengan perintah seperti berikut ini.

```
%sampling_2.m
clear all;
Fs=1000;
t=0:1/Fs:0.25;
f=100;
x=sin(2*pi*f*t);
sound(x,Fs)
```

2. Setelah anda menjalankan program tersebut apa yang anda dapatkan? Selanjutnya coba anda rubah nilai $f = 200, 300, 400, 500, 600, 700, 800$, dan 900 . Apa yang anda dapatkan? Bentuk suara yang sama dengan frekuensi pembangkitan berbeda itulah yang seringkali disebut orang sebagai efek aliasing. Coba anda catat frekuensi 200 memiliki bunyi yang sama dengan frekuensi berapa ? Sehingga frekuensi 200 adalah alias dari frekuensi tsb.

4.3 Pengamatan Efek Aliasing pada Audio 1

Tentunya anda bosan dengan sesuatu yang selalu serius, marilah kita sedikit bernafas melepaskan ketegangan tanpa harus meninggalkan laboratorium tempak praktikum. Caranya?

1. Anda susun sebuah lagu sederhana dengan cara membuat program baru berikut ini.

```
%gundul.m
clc
Fs=16000;
t=0:1/Fs:0.25;
c=sin(2*pi*262*t);
d=sin(2*pi*294*t);
e=sin(2*pi*330*t);
f=sin(2*pi*249*t);
g=sin(2*pi*392*t);
```



```

a=sin(2*pi*440*t);
b=sin(2*pi*494*t);
c1=sin(2*pi*523*t);
nol = [zeros(size(t))];
nada1 = [c,e,c,e,f,g,g,nol,b,c1,b,c1,b,g,nol,nol];
nada2 = [c,e,c,e,f,g,g,nol,b,c1,b,c1,b,g,nol];
nada3 = [c,nol,e,nol,g,nol,f,f,g,f,e,c,f,e,c,nol];
nada4 = [c,nol,e,nol,g,nol,f,f,g,f,e,c,f,e,c];
lagu=[nada1,nada2,nada3,nada4];
sound(lagu,Fs)

```

2. Pada bagian akhir program anda tambahkan perintah berikut
`wavwrite(lagu,'gundul.wav')`
3. Coba anda minimize Matlab anda, cobalah gunakan Windows Explorer untuk melihat dimana file gundul.wav berada. Kalau sudah terlihat coba click kanan pada gundul.wav dan bunyikan.
4. Coba anda edit program anda diatas, dan anda lakukan perubahan pada nilai frekuensi sampling $F_s=16000$, menjadi $F_s=10000, 8000, 2000, 1000, 900, 800, 700, 600$, dan 500. Apa yang anda dapatkan?

4.4 Pengamatan Efek Aliasing pada Audio 2

Disini kita akan bermain dengan sebuah lagu yang diambil dari sebuah file *.wav. Untuk itu mulailah dengan langkah

1. Anda buat program baru seperti berikut ini.

```

%sampling_3.m
%bersama: Tri Budi 212
clear all;
[Y,Fs]=wavread('lagu_1_potong.wav');
Fs=16000;%nilai default Fs=16000
%Pilihan untuk memainkan lainnya Fs=8000, 11025, 22050,44100
sound(Y,Fs)

```

Apakah anda sudah menikmati musiknya?

3. Lanjutkan langkah anda dengan merubah nilai $F_s = 8000$. Jalankan program anda, dan dengarkan yang terjadi.

4. Ulangi lagi dengan merubah nilai $F_s = 11025, 22050, \text{ dan } 44100$. Kalau anda belum puas coba ganti F_s sesuka hati anda. Jangan lupa catat dan buat analisa tentang fenomena yang terjadi dengan percobaan anda.

V. ANALISA DATA

Setelah anda puas bermain dengan teorema sampling, sekarang saatnya anda melakukan hal yang lebih bermanfaat. Apa yang telah anda lakukan dan dicatat tentunya (smile), buat laporan dan analisa mengapa muncul fenomena seperti diatas? Fenomena itu lebih dikenal dengan nama apa? Apa yang menyebabkannya?