

MODUL 7

TRANSFORMASI FOURIER DISKRIT

I. TUJUAN

- Siswa mampu memahami konsep dasar transformasi sinyal awaktu diskrit dan mampu menyusun program simulasinya.

II. TEORI DASAR

Sebelum kita berbicara tentang transformasi Foureir Diskrit atau dalam bahasa aslinya disebut sebagai discrete Fourier transform (DFT), marilah kita kembali sejenak tentang sesuatu yang sudah populer di telinga kita yaitu *Fourier transform* (FT). Transformasi Fourier untuk sebuah sinyal waktu kontinyu $x(t)$ secara matematis dituliskan sebagai

$$X(\omega) = \int_{-\infty}^{\infty} x(t)e^{-j\omega t} dt \quad \text{dimana } \omega \in (-\infty, \infty) \quad (1)$$

Sementara DFT dibentuk dengan menggantikan integral berhingga dengan sederetan jumlahan pada suatu nilai berhingga:

$$X(\omega_k) \triangleq \sum_{n=0}^{N-1} x(t_n)e^{-j\omega_k t_n} \quad k = 0, 1, 2, \dots, N-1 \quad (2)$$

Simbol \triangleq memiliki *arti equal by definition* atau dalam bahasa yang sudah kita ketahui adalah bahwa sisi kiri secara definisi akan senilai dengan sisi kanan. Sementara $x(t_n)$ selanjutnya akan kita kenal juga sebagai $x(n)$, yang merupakan notasi sample ke- n pada sinyal input. $X(\omega_k)$ juga dapat dijumpai sebagai $X(k)$ yang merupakan spectral sample ke- k .

Parameter lain yaitu:

- $j\triangleq\sqrt{-1}$ = merupakan dasar dari bilangan kompleks.
- $e\triangleq\lim_{n \rightarrow \infty} \left(1 + \frac{1}{n}\right)^n = 2,718281828\dots$
- $\omega_k = k\Omega$ = merupakan sample frekuensi ke- k . Sedangkan Ω merupakan interval sampling dalam radian dan memiliki nilai $\Omega = 2\pi/NT$.
- N = merupakan sample frekuensi yang digunakan.
- $T = 1/f_s = 1/(\text{sampling rate})$.

Dengan melihat persamaan (2) jelas bagi kita bahwa DFT memiliki basis sinyal sinusoda dan merupakan bentuk kompleks. Sehingga representasi domain frekuensi yang dihasilkan juga akan memiliki bentuk kompleks. Dengan demikian anda akan melihat adanya bagian real dan imajiner, dan bisa juga hasil transformasi direpresentasikan dalam bentuk nilai absolute yang juga dikenal sebagai magnitudo respon frekuensinya dan magnitudo respon fase.

Selanjutnya untuk proses pengolahan sinyal digital, kita DFT mutlak diperlukan karena kita akan berhubungan dengan sinyal waktu diskrit, yang merupakan bentuk tersampel dari sinyal waktu kontinyu. Dan dalam praktikum ini kita akan memanfaatkan bentuk dasar library fft yang merupakan pengembangan dari algoritma dasar DFT. Mengapa kita menggunakan fft? Hal ini bisa dijawab dengan anda masuk ke Matlab command like dan ketikkan *help fft* Akan muncul keterangan:

FFT Discrete Fourier transform.

FFT(X) is the discrete Fourier transform (DFT) of vector X. For matrices, the FFT operation is applied to each column. For N-D arrays, the FFT operation operates on the first non-singleton dimension.

FFT(X,N) is the N-point FFT, padded with zeros if X has less than N points and truncated if it has more.

Cukup jelas bagi kita mengapa kita bisa memanfaatkan library fft dalam praktikum kali ini.

III. PERALATAN

- PC multimedia yang sudah dilengkapi dengan OS Windows
- Perangkat Lunak Matlab yang dilengkapi dengan Tool Box DSP

IV. LANGKAH PERCOBAAN

Sebelum memasuki bentuk DFT yang benar-benar representatif dalam pengolahan ke domain frekuensi yang sebenarnya, kita akan memulai dengan langkah yang paling dasar dengan tujuan anda akan merasa lebih mudah memahaminya bagaimana sebenarnya konsep DFT bekerja.

1. Dasar Pembentukan DFT

Disini kita mulai dengan mencoba melihat bentuk transformasi Fourier dari sinyal cosinus yang memiliki periode eksak didalam window yang terdapat pada sampel. Langkahnya adalah sebagai berikut:

1. Bangkitkan sinyal sinus $x(t) = 3\cos(2\pi t)$, pada $t = nT$. Untuk suatu $n = 0 \sim 99$, dan $T=0,01$.

```
%File Name: dft_1.m
n=0:199;
T=0.01;
x_t=3*cos(2*pi*n*T);
plot(n,x_t)
grid;
```

2. Untuk sementara anda jangan memperhatikan apakah sinyal yang muncul sesuai dengan nilai sebenarnya. Biarkan axis dan ordinatnya masih dalam angka seadanya. Anda ganti bagian perintah $plot(n,x_t)$ dengan $stem(n,x_t)$. Coba perhatikan apa yang anda dapatkan.
3. Untuk memulai langkah program DFT, kita mulai dengan membuat program baru, yang mengacu pada bentuk persamaan berikut ini.

$$X(k) = \sum_{n=0}^{N-1} x(n)e^{-jk\omega_0 n} \quad 0 \leq k \leq N-1$$

Atau dalam bentuk real dan imajiner:

$$X(k) = \sum_{n=0}^{N-1} (3\cos(0,02\pi n))(\cos(k\omega_0 n) - j \sin(k\omega_0 n))$$

```
%File Name: dft_2.m
clear all;
N=200;
nn=N-1;
for k=1:200;
    x_n=0.0;
    for n=1:nn
        x_n = (3*cos(0.02*pi*n)).*(exp(-j*k*2*pi*n/200)) + x_n;
    end
    yR(k)=real(x_n);
    yI(k)=imag(x_n);
```

```

    magni_k(k)=sqrt(real(x_n).*real(x_n) +imag(x_n).*imag(x_n));
end

```

```

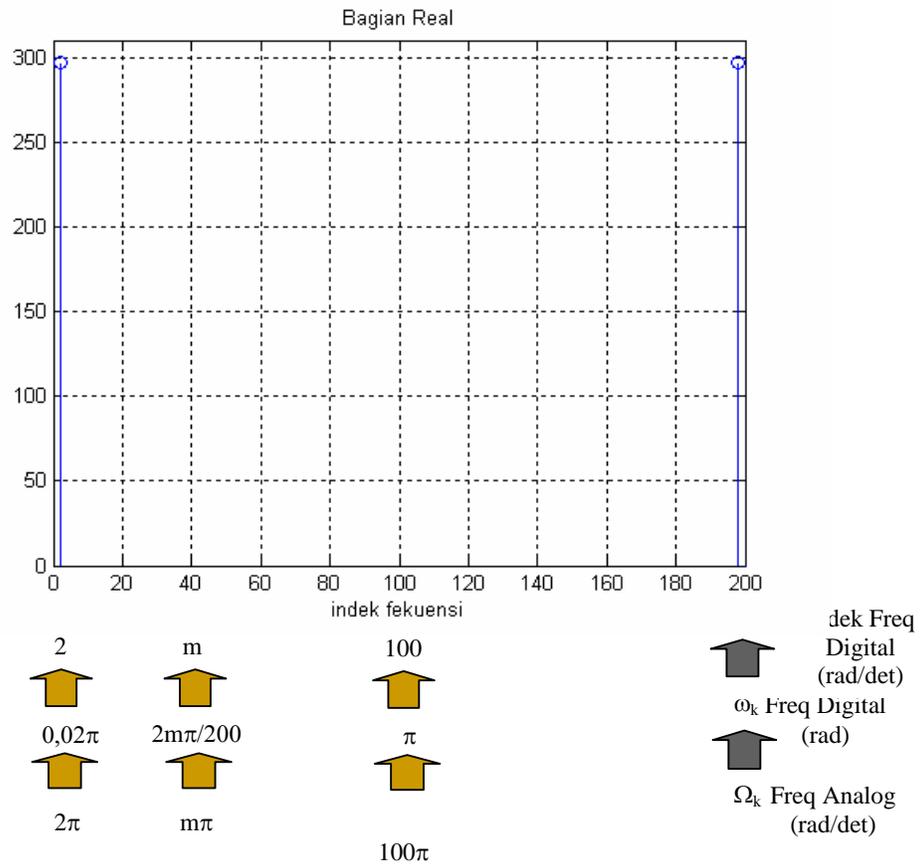
figure(1)
stem(yR)
axis([0 200 0 800])
xlabel('indek fekuensi')
title('Bagian Real')
grid;

```

```

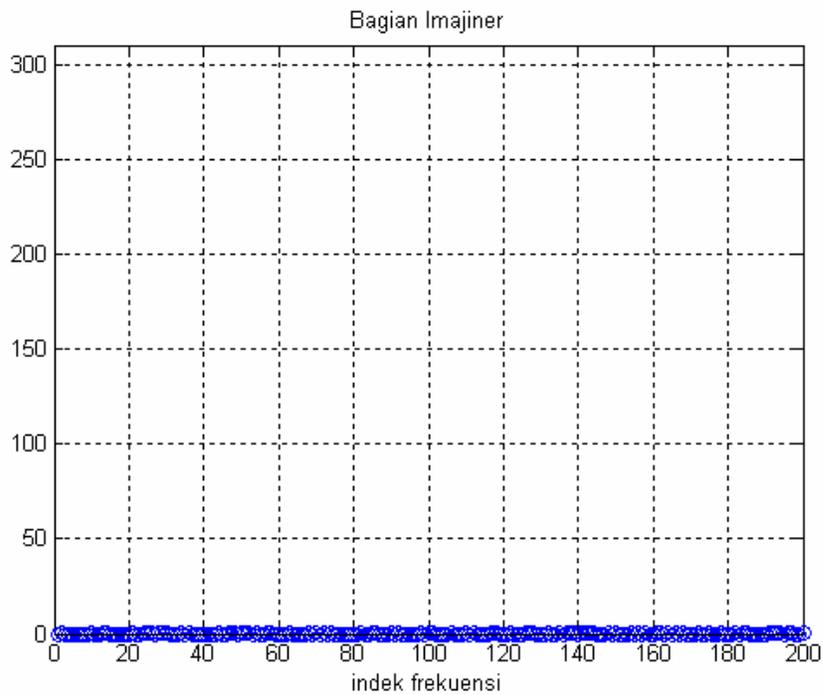
figure(2)
stem(yI)
axis([0 200 0 800])
xlabel('indek frekuensi')
title('Bagian Imajiner')
grid;

```



Gambar 1. Bagian real pada domain frekuensi

Anda perhatikan ada dua nilai non-zero dalam domain frekuensi indek, tepatnya pada $n=2$ dan $n=N-2$ atau 198, masing-masing bernilai 300. Nilai ini merepresentasikan $AN/2$, dimana $A=3$ yang merupakan amplitudo sinyal cosinus dan $N = 200$ merupakan jumlah sample yang digunakan. Sementara bagian imajiner bernilai nol semua, mengapa?



Gambar 2. Bagian imajiner pada domain frekuensi

4. Coba ulangi langkah 1-3 dengan merubah dari sinyal cosinus menjadi sinyal sinus. Untuk langkah k-1 anda rubah

$$x_t = 3 \cdot \cos(2 \cdot \pi \cdot n \cdot T); \rightarrow \text{menjadi} \rightarrow x_t = 3 \cdot \sin(2 \cdot \pi \cdot n \cdot T);$$

Demikian juga pada untuk langkah ke-3 bentuk

$$x_n = (3 \cdot \cos(0.02 \cdot \pi \cdot n)) \cdot (\exp(-j \cdot k \cdot 2 \cdot \pi \cdot n / 200)) + x_n;$$

menjadi

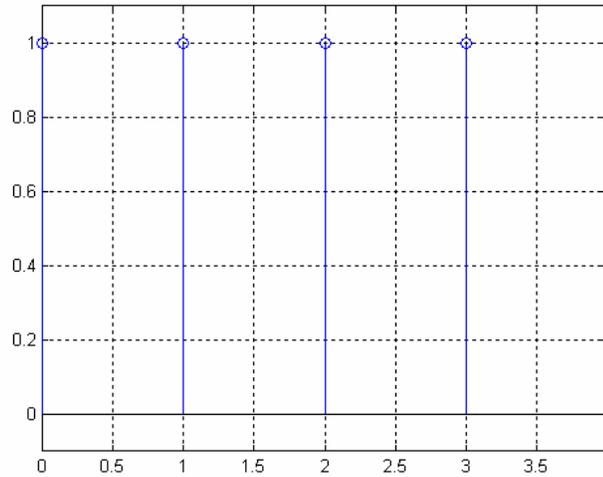
$$\rightarrow x_n = (3 \cdot \sin(0.02 \cdot \pi \cdot n)) \cdot (\exp(-j \cdot k \cdot 2 \cdot \pi \cdot n / 200)) + x_n;$$

Apa yang anda dapatkan?

5. Ulangi langkah 1-3 dengan merubah nilai sample $N=200$, menjadi $N=1000$. Apa yang anda dapatkan?

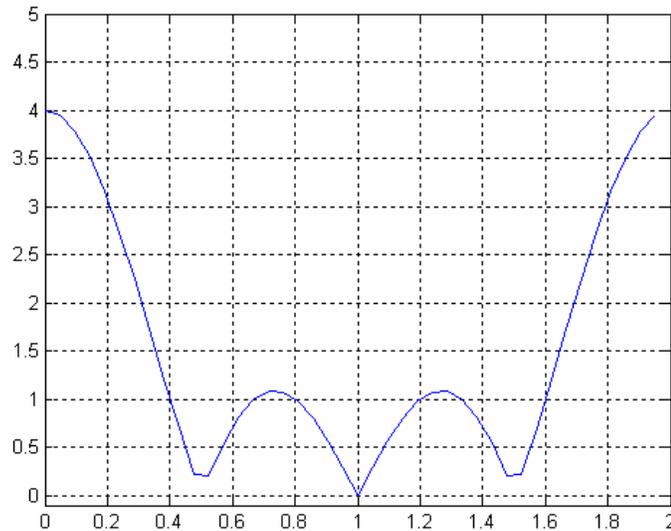
2. Zero Padding

Kita mulai dengan sebuah sinyal waktu diskrit berupa sekuen unit step.



Gambar 3. Sekuen unit step

Apabila kita menggunakan transformasi Fourier pada sinyal ini, akan diperoleh bentuk seperti berikut:

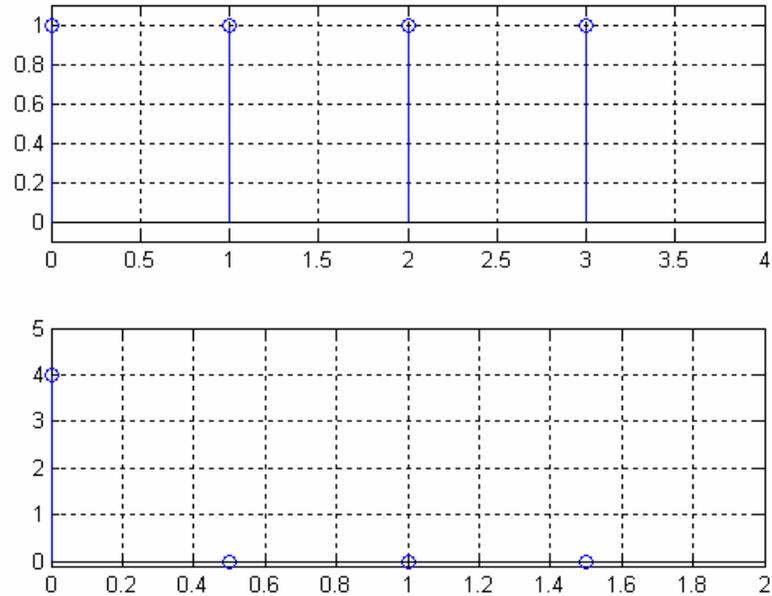


Gambar 4. Transformasi fourier sekuen unit

Untuk memahami konsep zero padding pada DFT, anda ikuti langkah-langkah percobaan berikut ini.

1. Buat program baru untuk pembangkitan sekuen unit step dan gunakan juga fft untuk memperoleh nilai DFT.
2. Modifikasi program anda dengan menambahkan nilai nol sebanyak 4 angka di belakang sekuen bernilai satu tersebut.
3. Modifikasi program anda sehingga nilai nol dibelakang sekuen unit step menjadi 12, catat apa yang terjadi.

4. Lanjutkan penambahan nilai nol menjadi 16, dan catat apa yang terjadi.



Gambar 5. Sekuen unit step dan hasil DFT

Jelaskan konsep zero padding yang telah anda buat simulasinya.....(smile)

3. Representasi Dalam Domain Frekuensi

Cara yang paling mudah dalam menguji program transformasi ke domain frekuensi adalah dengan menggunakan sinyal bernada tunggal, yaitu sinyal dengan fungsi dasar sinusoida. Untuk itu coba anda perhatikan dengan yang telah anda lakukan pada percobaan ke-1, yaitu pada pemahaman dasar DFT. Disitu sinyal cosinus yang ditransformasikan menghasilkan bentuk dalam tampilan indek frekuensi. Dengan mengkombinasikan percobaan ke-1 dan percobaan ke-2 kita akan mampu menyusun sebuah program DFT yang mampu digunakan untuk pengamatan sinyal waktu diskrit dan melihat tampilannya dalam domain frekuensi. Untuk itu ikuti langkah berikut.

1. Susun sebuah program baru dengan algoritma yang merupakan kombinasi dari percobaan ke-1 dan percobaan ke-2.

```
%prak_SS_7_2.m
% zero-padded data:
clear all
T = 128;           % sampling rate
zpf = 2;          % zero-padding factor
```

```

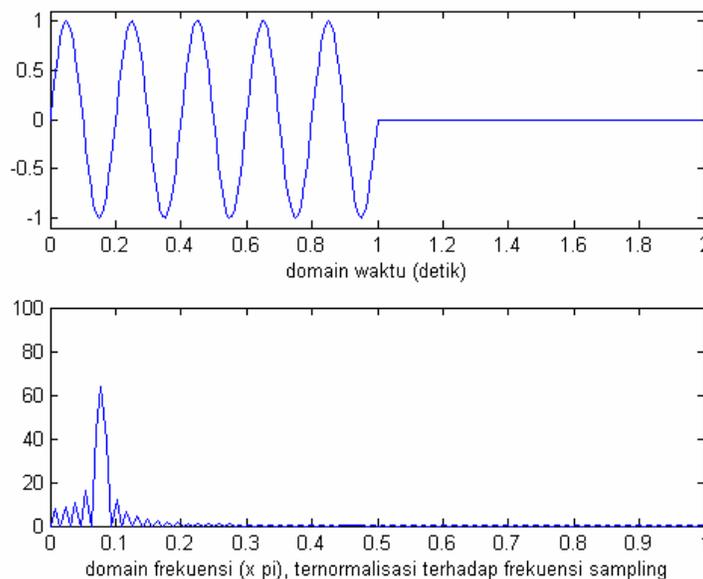
n = 0:1/T:(T-1)/T;      % discrete time axis
fi = 5;                 % frequency
xw = [sin(2*pi*n*fi),zeros(1,(zpf-1)*T)];
nn=length(xw);
k=0:nn-1;

% Plot time data:
subplot(2,1,1);
plot(zpf*k/nn,xw);%normalisasi absis domain waktu
axis([0 zpf -1.1 1.1])
xlabel('domain waktu (detik)')

% Smoothed, interpolated spectrum:
X = fft(xw);
spec = abs(X);
f_X=length(X)
f=0:f_X-1;

% Plot spectral magnitude:
subplot(2,1,2);
plot(f/T,spec);
axis([0 T/T 0 100])
xlabel('domain frekuensi (x pi), ternormalisasi terhadap frekuensi sampling')

```



Gambar 6. Sinyal sinus dalam domain waktu dan hasil DFT

- Lakukan beberapa modifikasi, sehingga tampilannya nilai frekuensi dalam Hz.

```
% Plot spectral magnitude:
subplot(2,1,2);
plot(f/2,spec);
axis([0 T/2 0 100])
xlabel('domain frekuensi')
```

Amati dan catat hasilnya.

- Lakukan modifikasi kembali untuk mendapatkan nilai magnitudo dalam besaran dB

```
% Plot spectral magnitude:
subplot(2,1,2);
plot(f/2,spec);
axis([0 T/2 0 40])
xlabel('domain frekuensi dalam dB')
grid
```

Amati dan catat hasilnya

- Sekarang coba bangkitkan sebuah sinyal sinus dan dapatkan nilai frekuensinya dengan memanfaatkan DFT. Dimana sinyal sinus ini memiliki bentuk dasar sebagai berikut.

$$x(n) = (1/64) * (\sin(2 * \pi * n / 64) + (1/3) * \sin(2 * \pi * 15 * n / 64))$$

V. ANALISA DATA DAN TUGAS

Dari apa yang telah anda lakukan anda catat hasilnya, dan jawab beberapa pertanyaan berikut:

1. Apa yang dimaksud dengan zero padding?
2. Apa pengaruh perbedaan nilai zero padding pada tampilan sinyal dalam domain frekuensi?
3. Berapa sample yang dipersyaratkan dalam operasi DFT?
4. Apa perbedaan tampilan nilai frekuensi dalam radiant dan tampilan frekuensi dalam Hz?
5. Apa yang dimaksud tampilan nilai magnitudo dalam dB?