MODUL

Nyquist dan Efek Aliasing, dan Transformasi Fourier Diskrit

PENDAHULUAN

Pada awalnya kita hanya mengenal sinyal atau isyarat **analog** dan **kontinyu** (terus menerus tanpa ada jeda sedikitpun, misalnya antara data untuk t=0 detik hinga t=1 detik, kita memiliki semua data secara lengkap, tidak hanya pada t=0 detik dan t=1 detik saja).

Dengan adanya teknologi komputer, pemrosesan sinyal mengalami kemajuan karena data-data sinyal tersebut dapat tersimpan dan diproses menggunakan komputer, caranya? Yaitu dengan melakukan **pencuplikan** (bisa dibayangkan berapa banyak data yang tersimpan jika masih bersifat kontinu? Karena antara t=0 detik hingga t=1 detik bisa berjumlah tak-hingga) menjadi **data-data diskrit**, hanya untuk saat t tertentu saja, misalnya dengan **periode pencuplikan** T=0.5 detik, akan diperoleh **frekuensi pencuplikan** fs=2 Hz atau 2 data tiap detik, sehingga untuk 1 menit = 60 x 2 daa = 120 data/menit.

Tidak hanya proses pencuplikan, juga dilakukan **proses kuantisasi**, yaitu merubah angka analog menjadi digital selebar n-bit, artinya jika hanya menggunakan 3-bit maka hanya ada 2^3 = 8 tingkat data, demikian seterusnya, semakin lebar bit-nya semakin akurat dan otomatis semakin membutuhkan banyak ruang penyimpan.

Data digital tersebut perlu dianalisa lebih lanjut, karena masih dalam **ranah waktu** (***time domain***), informasi yang diperoleh hampir tidak ada, sehingga seringkali dibutuhkan informasi, misalnya, spektrum atau kandungan frekuensi dari sinyal yang bersangkutan (**ranah frekuensi** atau ***frequency domain***). Sehingga perlu mempelajari **DFT** atau **Discrete Fourier Transform**.

PERCOBAAN

1. Nyquist

Dalam dunia Pemrosesan Sinyal Digital, ada suatu proses untuk mendapatkan data digital melalui proses **pencuplikan**, artinya sinyal analog dicuplik (diambil) secara diskrit dengan periode Ts atau frekuensi cuplik Fs. agar tidak terjadi kesalahan (yang kemudian diberi nama ***aliasing***), Mr. Nyquist memberikan aturan bahwa frekuensi cuplik **minimal harus 2 (dua) kali** lipat frekuensi maksimum yang dikandung sinyal yang bersangkutan.

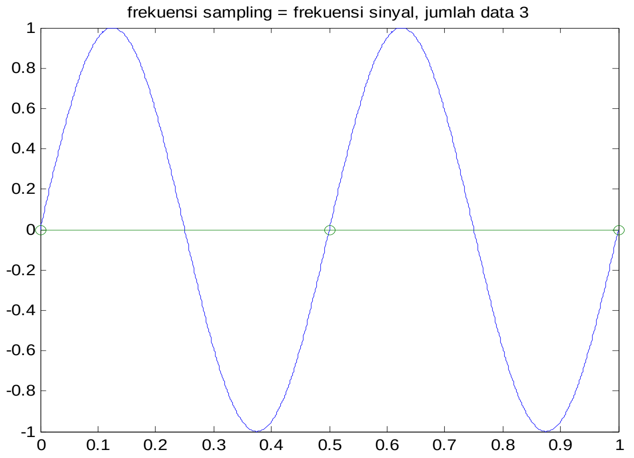
Untuk memahami hal tersebut, mari kita persiapkan dulu sinyal sinusoidal dengan frekuensi 2 Hz. Kita gunakan frekuensi cuplik 1000 Hz atau periode 0.001 detik (supaya gambarnya jauh lebih 'smooth' dibandingkan dengan eksperimen-eksperimen yang akan kita lakukan)

```
>>t=0:0.001:1;
>>f=2;
>>y=sin(2*pi*f*t);
```

kemudian diteruskan dengan menyiapkan variabel waktu t1 dengan frekuensi sampling = frekuensi sinyal (atau periodenya 1/f detik), kita hitung y1-nya kemudian kita gambarkan sinyal asli y dan sinyal hasil pencuplikan y1, menggunakan perintah-perintah berikut:

```
>>t1=0:1/f:1;
>>y1=sin(2*pi*f*t1);
>>plot(t,y,t1,y1,'-o');
>>title(sprintf('frekuensi sampling = frekuensi sinyal, jumlah data
%d',length(y1)));
```

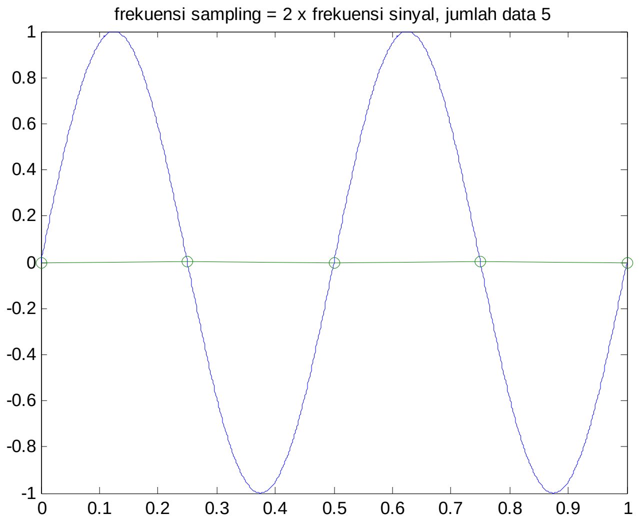
Maka akan didapatkan hasil sebagai berikut (yang asli warna biru, yang cuplikan warna hijau)



Hmmm…kok cuman garis lurus, berarti salah ya……
Bukan salah… tapi di Program yang Anda buat frekuensi pencuplikan = frekuensi sinyal, harusnya kan 2 kali lipat, coba kita ubah lagi… (gunakan 1/(2*f))

```
>>figure;
>>t1=0:1/(2*f):1;
>>y1=sin(2*pi*f*t2);
>>plot(t,y,t1,y1,'-o');
>>title(sprintf('frekuensi sampling = 2 x frekuensi sinyal, jumlah data
%d',length(y1)));
```

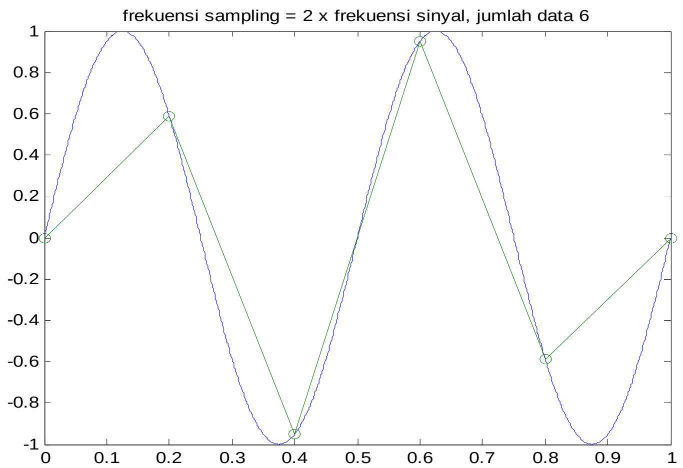
Hasilnya



Lho…sama saja… ya garis lurus lagi (walaupn jumlah datanya bertambah, perhatikan lingkaran-lingkaran hijau, bandingkan dengan gambar sebelumnya)! Mr. Nyquist sudah bilang kalo itu 2 kali lipat adalah minimal, ya mestinya pake yang lebih tinggi lagi, coba sekarang pake 2,5 kali lipat.

```
>>figure;
>>t1=0:1/(2.5*f):1;
>>y1=sin(2*pi*f*t1);
>>plot(t,y,t1,y1,'-o');
>>title(sprintf('frekuensi sampling = 2,5 x frekuensi sinyal, jumlah data
%d',length(y1)));
```

Hasilnya…



Wah ini agak lumayan (maksudnya dibandingkan hasil-hasil sebelumnya), bukan garis lurus, tapi… belum berbentuk sinusoidal ya…?? Okey sekarang coba anda teruskan sendiri 5, 10, 30, 40, 70 kali lipat. Apa yang dapat Anda simpulkan?

2. Efek-Alliasing

Efek Aliasing (yang nanti akan lihat bahwa frekuensi alias = frekuensi pencuplikan - frekuensi sinyal), yaitu suatu efek yang akan terjadi jika kita melakukan pencuplikan dengan frekuensi pencuplikan dibawah dari ketentuan Nyquist.

Pertama kita gunakan frekuensi pencuplikan sebesar 20 Hz dan rentang waktu selama 1 detik.

```
fs=20;
t=0:1/fs:1;
```

Kita awali dengan frekuensi 1Hz, kemudian dilanjutkan dengan frekuensi 2Hz, 5Hz, 10Hz (tepat setengah dari frekuensi pencuplikan), 15Hz, 18Hz dan 19Hz.

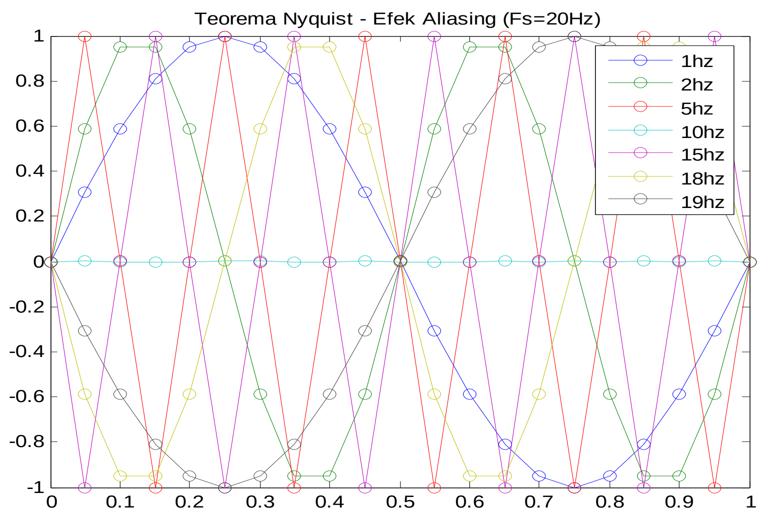
```
f1=1;
x1=sin(2*pi*f1*t);
f2=2;
x2=sin(2*pi*f2*t);
f3=5;
x3=sin(2*pi*f3*t);
f4=10;
x4=sin(2*pi*f4*t);
f5=15;
x5=sin(2*pi*f5*t);
f6=18;
```

```
x6=sin(2*pi*f6*t);
f7=19;
x7=sin(2*pi*f7*t);
```

Kemudian kita gambarkan semua hasil dalam satu plot agar kita bisa membandingkan 7 frekuensi tersebut (dengan frekuensi pencuplikan yang sama, 20Hz)…

```
plot(t,x1,'-o',t,x2,'-o',t,x3,'-o',t,x4,'-o',t,x5,'-o',t,x6,'-o',t,x7,'-o');
legend('1hz','2hz','5hz','10hz','15hz','18hz','19hz');
title('Teorema Nyquist - Efek Aliasing (Fs=20Hz)');
```

Hasilnya ditunjukkan pada gambar berikut.



Anda perhatikan bahwa untuk frekuensi 10Hz hasilnya berupa garis lurus (masih ingat dengan materi Teorema Nyquist), jelas karena frekuensi pencuplikan = 2 kali frekuensi 10Hz. Bagaimana dengan frekuensi 1Hz, 2Hz dan 5Hz? baik2 aja kan?

Bagaimana dengan frekuensi 15Hz, 18Hz dan 19Hz? Nah gambarnya memang terbalik dari frekuensi, masing-masing, 5Hz, 2Hz dan 1Hz tetapi frekuensi-nya khan sama, iya khan? Ini dia yang dinamakan **Efek Aliasing**. Misalnya untuk 15Hz menghasilkan…

frekuensi aliasing = frekuensi pencuplikan - frekuensi 15Hz = 20Hz - 15Hz =5Hz

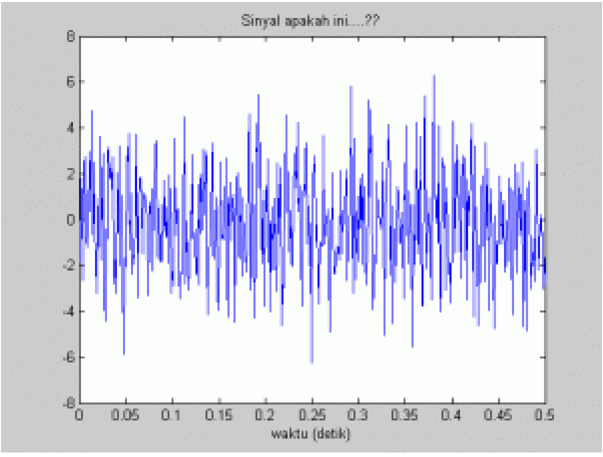
Jadi frekuensi 15Hz merupakan kembaran (alias) dari 5Hz. Demikian seterusnya untuk contoh-contoh frekuensi aliasing lainnya (18Hz dan 19Hz yang masing-masing menghasilkan 2Hz dan 1Hz). Apa yang dapat Anda Simpulkan tentang Alliasing?

3. Transformasi Fourier

Transformasi Fourier adalah ‘alat’ yang bisa kita gunakan untuk melihat sinyal dengan kacamata yang lain. Jika selama ini kita hanya melihat sinyal melalui osiloskop atau alat sejenis lainnya, itu adalah visualisasi sinyal dalam ranah waktu (*time domain*), sumbu horisontal-nya waktu (t) dan sumbu vertikal-nya adalah amplitudo (A).

Kita tidak bisa tahu secara langsung informasi penting di ranah waktu kecuali amplitudo dan kapan terjadinya. Bisakah menghitung frekuensinya? itu mungkin jawaban Anda, karena bisa jadi Anda hanya membayangkan sebuah gelombang sinusoidal, Anda tinggal hitung berapa gelombang dalam satu detiknya, Langsung ketemu sekian gelombang/detik atau pake satuan cycle/sec atau Hz.

Bagaiman kalo sinyalnya seperti ini



Bagaimana kita bisa ngitungnya kalo bentuknya gak karuan, terkesan acak, jangan-jangan memang tidak periodik... ☺

Baik sebelum kita cari informasi frekuensi dari gelombang tersebut, perhatikan penjelasan berikut:

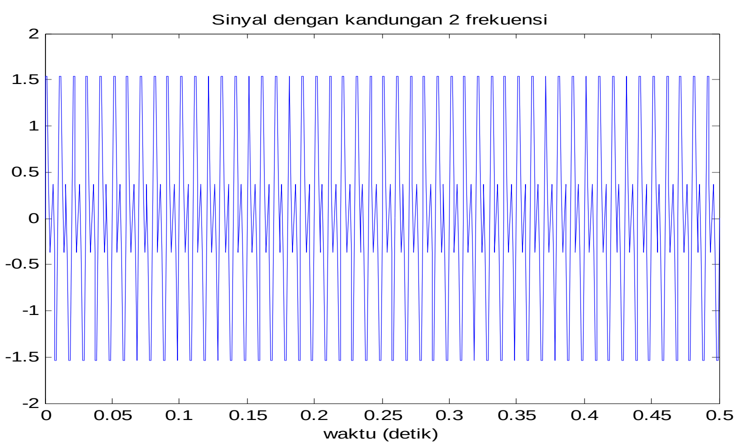
Masih pake bantuan **Matlab**, kita awali dengan mendefinisikan sinyal yang akan digunakan sebagai percobaan. Frekuensi cuplikan 1000 Hz (=fs), kita gambar hanya 0,5 detik pertama (=t) untuk sebuah sinyal dengan 2 kandungan frekuensi 100 Hz dan 200 Hz (=y):

```
fs = 1000;
t = 0:1/fs:0.5;
y = sin(2*pi*100*t)+sin(2*pi*200*t);
```

kemudian kita gambar …

```
plot(t,y);
title('Sinyal dengan kandungan 2 frekuensi')
xlabel('waktu (detik)');
```

hasilnya…



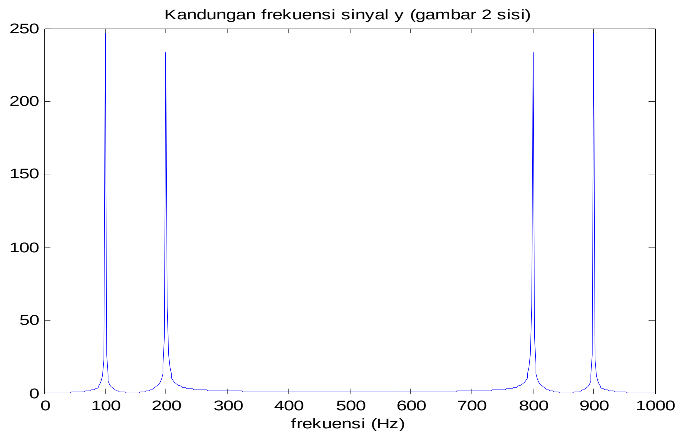
Kemudian kita lakukan **FFT** (***Fast Fourier Transform***), pake perintah fft-nya Matlab:

```
Y = fft(y);
```

Kemudian kita gambar hasilnya apa adanya, ingat bahwa hasil dari FFT selalu ada 2 (karena efek simetris):

```
f = fs*(0:length(Y)-1)/length(Y);
figure;
plot(f,abs(Y));
title('Kandungan frekuensi sinyal y (gambar 2 sisi)')
xlabel('frekuensi (Hz)');
```

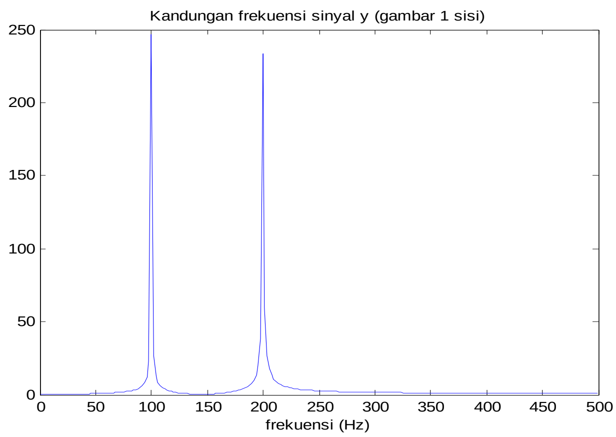
hasilnya



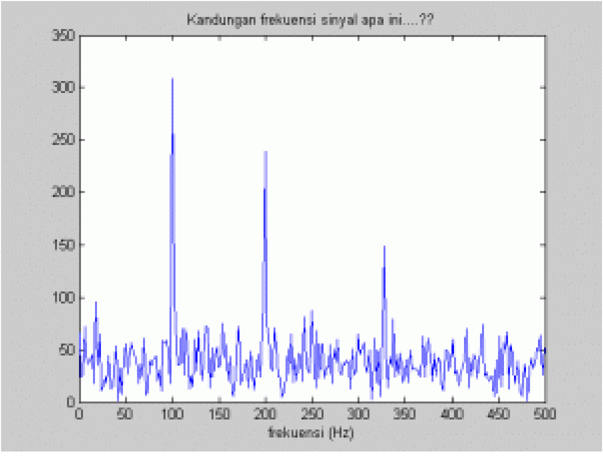
Baik sekarang kita gambar 1 sisi saja

```
f = fs*(0:(length(Y)-1)/2)/length(Y);
figure
plot(f,abs(Y(1:(length(Y)+1)/2)));
title('Kandungan frekuensi sinyal y (gambar 1 sisi)')
xlabel('frekuensi (Hz)');
```

hasilnya…



Langsung ketahuan frekuensinya… 100Hz dan 200Hz…! Terus bagaimana dengan gambar pertama tadi? Ini dia hasil FFT-nya…



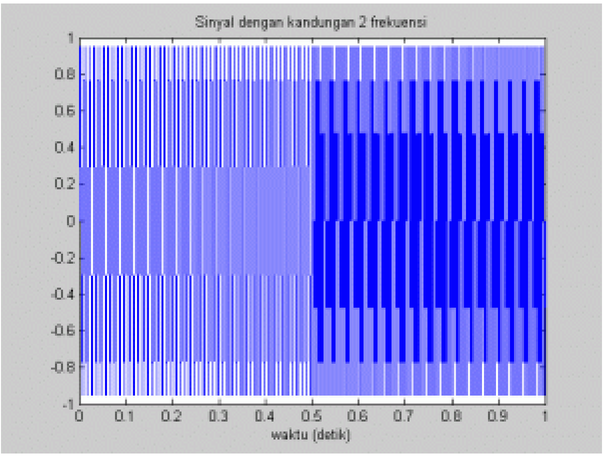
bagaimana? kandungan-nya sama seperti sinyal awal, 100 Hz dan 200Hz, tapi kok ada tambahan gambar 'rumput'-nya ya? (coba cari jawaban sendiri ☺)

Kapan Muncul Frekuensinya

Kita buat sebuah sinyal sebagai berikut, masih sama seperti sebelumnya, namun kali ini kedua frekuensi, yaitu 100 Hz dan 200Hz tidak muncul bersamaan tetapi bergantian, apakah Transformasi Fourier mampu melihat kedua frekuensi ini?

```
fs = 1000;
t = 0:1/fs:0.5;
tx = [t  t+t(length(t))];
y1 = sin(2*pi*100*t);
y2 = sin(2*pi*200*t);
y = [y1 y2];
plot(tx,y);
title('Sinyal dengan kandungan 2 frekuensi')
xlabel('waktu (detik)');
```

Hasilnya, gambaran dari sinyal yang saya tanyakan adalah sebagai berikut…



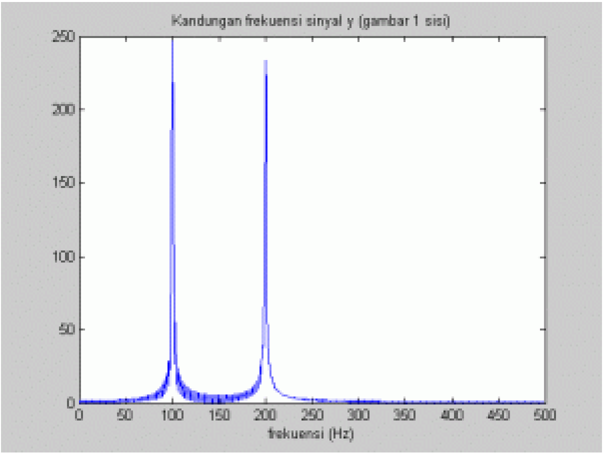
Kemudian kita lakukan FFT…

```
Y = fft(y);
```

Kita gambarkan hasilnya dengan perintah-perintah berikut…

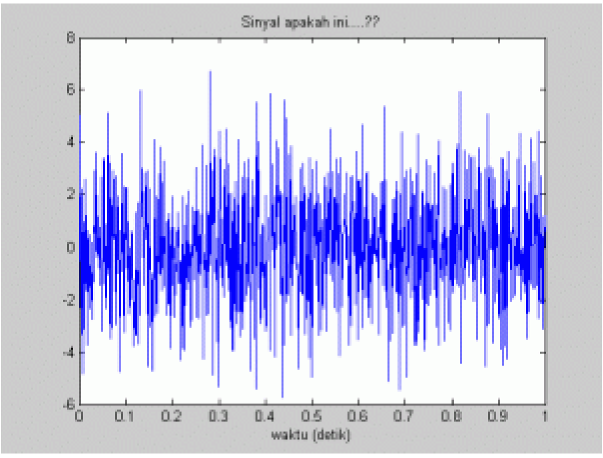
```
f = fs*(0:(length(Y)-1)/2)/length(Y);
figure;
plot(f,abs(Y(1:(length(Y)+1)/2)));
title('Kandungan frekuensi sinyal y (gambar 1 sisi)');
xlabel('frekuensi (Hz)');
```

hasilnya sebagai berikut…



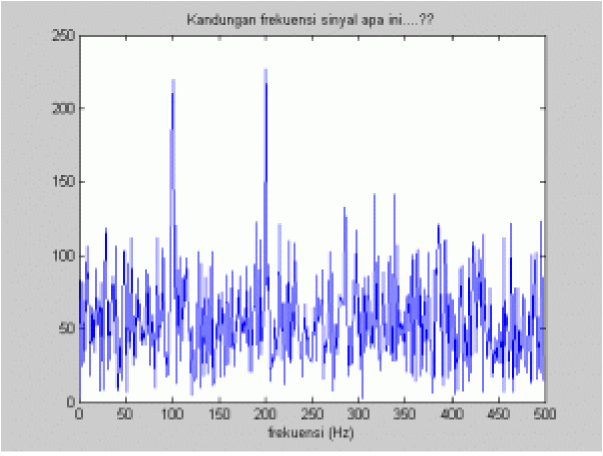
Bagaimana? Hasilnya sama… bisa diperoleh 2 frekuensi sesuai dengan dugaan kita, bagaimana jika ditambahkan derau kemudian di-FFT…

```
ya= y + 2*randn(size(tx));
figure;
plot(tx,ya);
title('Sinyal apakah ini….??');
xlabel('waktu (detik)');
```




```
YA = fft(ya);
f = fs*(0:(length(YA)-1)/2)/length(YA);
figure;
plot(f,abs(YA(1:(length(YA)+1)/2)));
title('Kandungan frekuensi sinyal apa ini….??');
xlabel('frekuensi (Hz)');
```

Hasilnya



Sama seperti pembahasan sebelumnya khan? Luar biasa Transformasi Fourier ini…

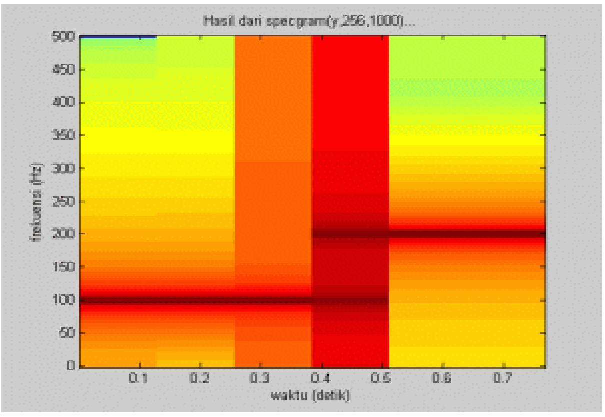
Bisakah kita tahu bahwasanya ke-2 frekuensi tidak bersamaan munculnya? Ya tidak bisa-lah… hanya kandungan frekuensi saja, sedangkan kapan dan lama waktu muncul masing-masing freuensi itu kita tidak tahu… kalo mo tahu? pake lainnya donk… Pake STFT (Short Time Fourier Trasnform)

Jika TF bekerja untuk seluruh sinyal, tapi STFT hanya bekerja pada sebuah jendela yang kecil yang kemudian digeser-geser mulai dari awal hingga akhir untuk mendapatkan interpretasi data keseluruhan secara waktu dan frekuensi atau istilahnya time-frequency domain, di Matlab pake perintah **specgram()**.

```
figure;
specgram(y,256,1000);
```

Keterangan: 256 sebagai panjang jendela, sedangkan 1000 merupakan fs-nya

Hasilnya

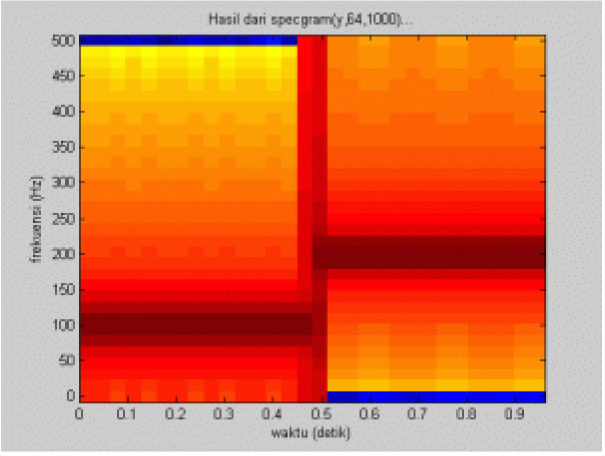


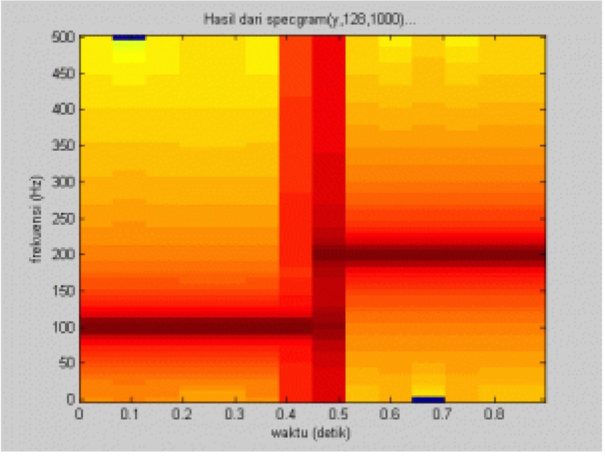
Terlihat bahwa kedua frekuensi muncul secara tidak bersamaan, lebih tepat berturutan, hanya saja tidak terlalu jelas dimana tepatnya frekuensi mulai bergantian… Baik sekarang Anda perhatikan masing-masing perintah dan hasil gambarnya sebagai berikut:

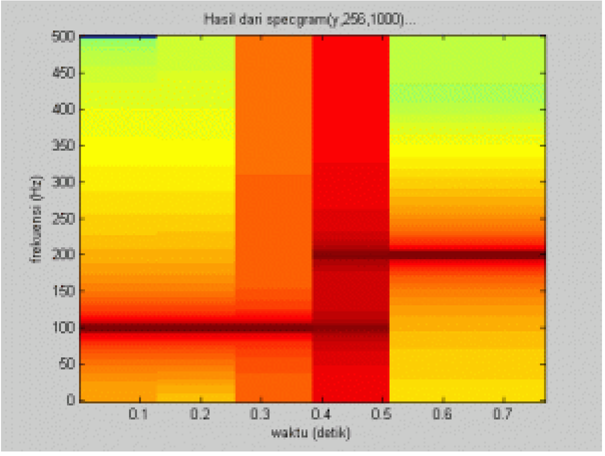
```
figure;
specgram(y,64,1000);
figure;
```

specgram(y,128,1000);
figure;
specgram(y,256,1000);

Hasilnya secara berturutan…







Bagaimana kesimpulannya? Ini-lah yang dimaksudkan dengan Ketidak-pastian Heisenberg… ada semacam trade-off antara resolusi waktu dan frekuensi, tapi minimal sudah kita peroleh ranah waktu-frekuensi.

Buat Laporan resmi dari modul Praktikum ini! (kumpulkan minggu depan)

Referensi : Agfianto Eko Putra, ‘Nyquist –Efek Alliasing’, Catatn Kuliah