



*Institut Teknologi Bandung*

*Astronomi*

---

***Date-Compensated Discrete Fourier  
Transform Kurva Cahaya DT Vir dan  
Kurva Kecepatan Radial Pr0211***

---

*Penulis:*

Muhammad Ali Syaifudin  
(10316012)

*Dosen:*

Dr. Hakim L. Malasan

*AS4103: Astrofisika Pengamatan*

11 November 2019

# 1 Pendahuluan

Data pengamatan astronomi seringkali memiliki jarak antar pengamatan yang tidak sama dan total data yang diperoleh juga sedikit. Hal ini membuat analisis deret waktu tidak bisa menggunakan cara konvensional. Ferraz-Mello (1981) [1] mengembangkan metode *Date-Compensated Discrete Fourier Transform* (DCDFT) untuk menghadapi permasalahan ini. Metode DCDFT baik digunakan untuk data pengamatan dengan jarak antar pengamatan yang tidak sama dan total data yang sedikit.

Dalam tugas ini, saya akan membuat program yang menggunakan DCDFT untuk data kurva cahaya DT Vir sesuai dengan petunjuk dari buku *Excercises in Astronomy* bab SEARCH AND DETERMINATION OF PERIODICITY [2]. Setelah berhasil mendapatkan hasil DCDFT yang sesuai dengan petunjuk buku ini, program yang saya bangun saya terapkan pada data kurva kecepatan radial Pr0211. Pr0211 merupakan salah satu sistem ekstrasolar di Beehive Cluster dengan periode orbit  $2.1451 \pm 0.0012$  hari [3].

## 2 Prosedur DCDFT

1. Simpan data dengan format kolom pertama sebagai waktu (contoh JD) dan kolom kedua sebagai magnitudo (sebagai tambahan, kolom ketiga bisa diisi error magnitudo)
2. Cari nilai rata-rata magnitudo, lalu hitung selisih magnitudo terhadap rata-rata magnitudo. Kita sebut nilai ini  $f$ . Waktu disimbolkan  $t$
3. Tentukan jumlah data ( $N$ ), rentang seluruh pengamatan ( $\Delta t$ ), dan rentang antar pengamatan paling kecil ( $T$ ). Periode Nyquist adalah dua kali periode  $T$ ,  $T_{Ny} = 2T$ . Frekuensi Nyquist adalah  $f_{Ny} = 1/T_{Ny}$
4. Buat cacahan frekuensi yang akan ditinjau  $\omega = \{\omega_0, \dots, \omega_{max}\}$  dengan rentang  $\omega$  adalah  $\Delta\omega = \omega_{max} - \omega_0$ ,  $\omega_0 > 0$  dan  $\omega_{max} \leq f_{Ny}$
5. Definisikan  $x = 2\pi\omega t$ . Hitung nilai-nilai berikut untuk masing-masing  $\omega$

$$a_0^{-2} = N$$

$$M = \sum \cos x_i \sin x_i - a_0^2 \left( \sum \sin x_i \right) \left( \sum \cos x_i \right)$$

$$a_1^{-2} = \sum \cos^2 x_i - a_0^2 \left( \sum \cos x_i \right)^2$$

$$a_2^{-2} = \sum \sin^2 x_i - a_0^2 \left( \sum \sin x_i \right)^2 - a_1^2 M^2$$

$$c_1 = a_1 \sum f_i \cos x_i$$

$$c_2 = a_2 \sum f_i \sin x_i - a_1 a_2 c_1 M$$

$$S = \frac{c_1^2 + c_2^2}{\sum f_i^2}$$

6. Hitung nilai berikut

$$G = -\frac{N-3}{2} \ln(1-S)$$

$$H = \frac{N-4}{N-3} \left( G + e^{-G} - 1 \right)$$

$$\alpha = \frac{2(N-3)\Delta t \Delta \omega}{3(N-4)}$$

$$C = \left( 1 - e^{-H} \right)^\alpha$$

Nilai C menyatakan *Confidence Level* dari nilai  $\omega$

### 3 Kode program

Saya menggunakan bahasa pemrograman Python untuk mengaplikasikan DCDFIT. Berikut kode programnya:

```
import numpy as np
import matplotlib.pyplot as plt

##### Inisiasi #####
file = 'Pr211'
date = np.loadtxt(file+'.txt', usecols=0)
vr = np.loadtxt(file+'.txt', usecols=1)
err = np.loadtxt(file+'.txt', usecols=2)
t0 = date[0]
##### Pre proc #####
N = len(date)      # Jumlah observasi
print('sebelum = ', N)
t = []
v = []
er = []
cek = np.full(N, True)
for i in range(N-1):
    if cek[i]:
        j = i+1
        count = 1
        Vr = vr[i]
        while date[i] == date[j]:
            Vr += vr[j]
            cek[j] = False
            j += 1
            count += 1
        t.append(date[i])
        v.append(Vr/count)
        er.append(err[i])
t = np.array(t, dtype='float')
v = np.array(v, dtype='float')
er = np.array(er, dtype='float')
N = len(t)
print('setelah = ', N)
#m = np.delete(m, np.argmax(m))
#t = np.delete(t, np.argmax(m))

##### Plot #####

plt.figure(0, figsize=(15,5))
plt.errorbar(t, v, yerr=er, fmt='k')
plt.xlabel('JD (%d+)'%date[0])
plt.ylabel('v (m/s)')
plt.title('Kurva Kecepatan Radial %s'%file)
plt.grid(True)
plt.show()

##### Inisiasi 2 #####
mulai = float(input('Mulai dari tanggal (%d+)= '%t0))
tf = t[len(t)-1]-t0
selesai = float(input('Sampai (max = %f) = '%tf))
t2 = []
v2 = []
er2 = []
count = 0
while mulai > t[count]:
    count += 1
while selesai >= t[count]:
    t2.append(t[count])
    v2.append(v[count])
    er2.append(er[count])
```

```

        if count == N-1:
            selesai = -1
        else:
            count += 1
    t = np.array(t2)
    v = np.array(v2)
    er = np.array(er2)
    ##### Plot #####

    plt.figure(0,figsize=(15,5))
    plt.errorbar(t, v, yerr=er, fmt='k')
    plt.xlabel('JD (%d+)'%date0)
    plt.ylabel('v (km/s)')
    plt.title('Kurva Kecepatan radial %s'%file)
    plt.grid(True)
    plt.show()

    ##### Perhitungan #####

    mr = np.mean(v) # Mag rata-rata
    f = v - mr      # simpangan dari rerata
    Difdate = np.diff(t) # Mencari selisih antar pengamatan
    NP = 2*min(Difdate) # Periode Nyquist, 2xselisih minimum

    if NP == 0:
        fMax = float(input('Masukkan frekuensi maksimum (frekuensi Nyquist = tidak terdefinisi) = '))
    else:
        fN = 1/NP #1/Day # Frekuensi Nyquist
        fMax = float(input('Masukkan frekuensi maksimum (frekuensi Nyquist = %f) = '%fN))

    Nfreq = int(input('Masukkan jumlah partisi = '))
    Df = fMax/Nfreq #Selang frekuensi
    fMin = float(input('Masukkan frekuensi minimum (>%f) = '%Df))
    DT = date[N-1]-date[0] #Lama observasi
    Nfreq = int(fMax/Df)
    # ##### Perhitungan konstanta #####

    omega = np.linspace(fMin, fMax, Nfreq) #bikin array omega

    x = 2*np.pi*omega
    a0 = np.sqrt(1/N)

    S = np.zeros(Nfreq)

    for i in range(Nfreq):
        cos2x = np.sum(np.cos(x[i]*t)**2)
        cosx2 = np.sum(np.cos(x[i]*t))**2
        sin2x = np.sum(np.sin(x[i]*t)**2)
        sinx2 = np.sum(np.sin(x[i]*t))**2
        M = np.sum(np.cos(x[i]*t)*np.sin(x[i]*t)) - a0**2*np.sum(np.sin(x[i]*t))*np.sum(np.cos(x[i]*t))

        a1 = np.sqrt(1./(cos2x - a0**2*cosx2))
        a2 = np.sqrt(1./(sin2x - a0**2*sinx2 - a1**2*M**2))

        c1 = a1*np.sum(f*np.cos(x[i]*t))
        c2 = a2*np.sum(f*np.sin(x[i]*t)) - a1*a2*c1*M

        S[i] = (c1**2+c2**2)/np.sum(f**2)

    G = -(N-3)/2*np.log(1-S)
    H = (N-4)/(N-3)*(G+np.exp(-G)-1)
    alpha = 2*(N-3)*DT*(fMax-fMin)/(3*(N-4))
    C = 100*(1-np.exp(-H))**alpha
    Period = 1/omega[np.argmax(H)]

    ##### Plot DCDF #####

    fig, ax1 = plt.subplots(figsize=(15,5))

    ax1.plot(omega, H, 'b-', label = 'H')
    ax1.plot(omega, S, 'g-', label = 'S')
    ax2 = ax1.twinx()
    ax2.plot(omega, C, 'r-.', label = 'Conf Lvl')
    fig.tight_layout()
    ax1.legend(loc='upper right')
    ax2.legend(loc='upper left')
    ax1.set_xlabel('Frekuensi (1/D)')
    ax1.set_ylabel('Power')
    ax2.set_ylabel('Confidence Level (%)')
    ax1.set_title('DCDF Kurva cahaya %s'%file)
    ax1.set_xlim(fMin,fMax)
    ax2.grid(color='k', linestyle='--', linewidth=.5, which='both', axis='y')
    ax2.set_ylim(0,100)
    ax1.set_ylim(0,max(H)*1.2)
    plt.show()
    ##### Menentukan fase dan plot kurva cahayanya #####

    print('Periode = %f hari'%Period)
    print('Frekuensi = %f /hari'%omega[np.argmax(H)])

    print('Confidence Interval = %f%%'%max(C))
    Ya = input('Gunakan periode? (y/n) = ')
    if Ya == 'n':
        Period = float(input('Masukkan periode = '))
    fase = (t-t[0])/Period - (t-t[0])/Period
    plt.figure(0,figsize=(15,5))
    plt.errorbar(fase, v, yerr=er, fmt='k')

```

```
plt.ylim(mr+min(f)*1.1,mr+max(f)*1.1)
plt.xlabel('Fase')
plt.ylabel('v (m/s)')
plt.xlim(0,1)
plt.title('Kurva kecepatan radial %s'%file)
plt.grid(True)
plt.show()
```

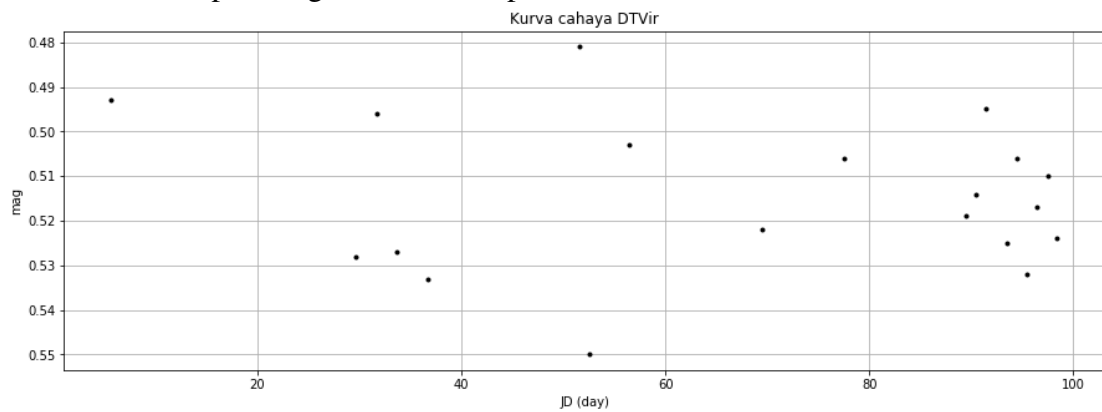
## 4 Data dan Hasil

### 4.1 Kurva Cahaya DT Vir

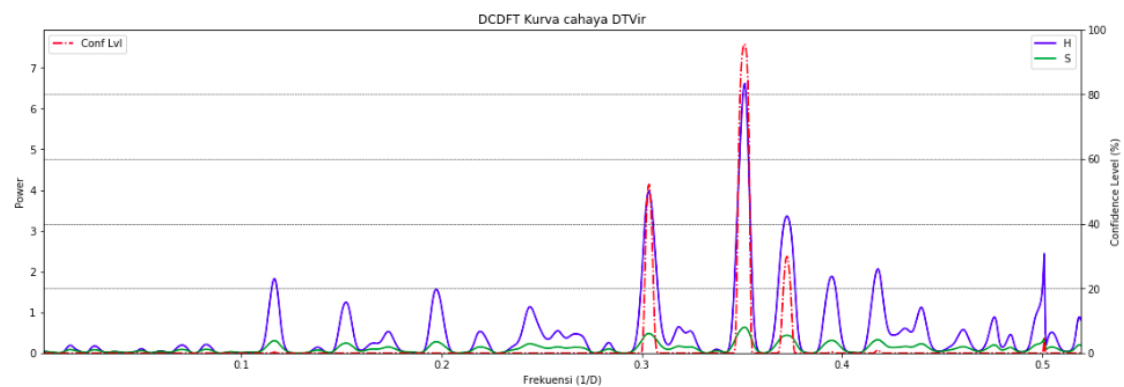
Data kurva Cahaya DT Vir, dengan kolom pertama adalah JD2441400+ dan kolom kedua adalah magnitudo relatif B

5.682	0.493
29.630	0.528
31.664	0.496
33.651	0.527
36.671	0.533
51.596	0.481
52.583	0.550
56.511	0.503
69.518	0.522
77.499	0.506
89.477	0.519
90.458	0.514
91.503	0.495
93.468	0.525
94.464	0.506
95.443	0.532
96.461	0.517
97.479	0.510
98.442	0.524
101.469	0.532

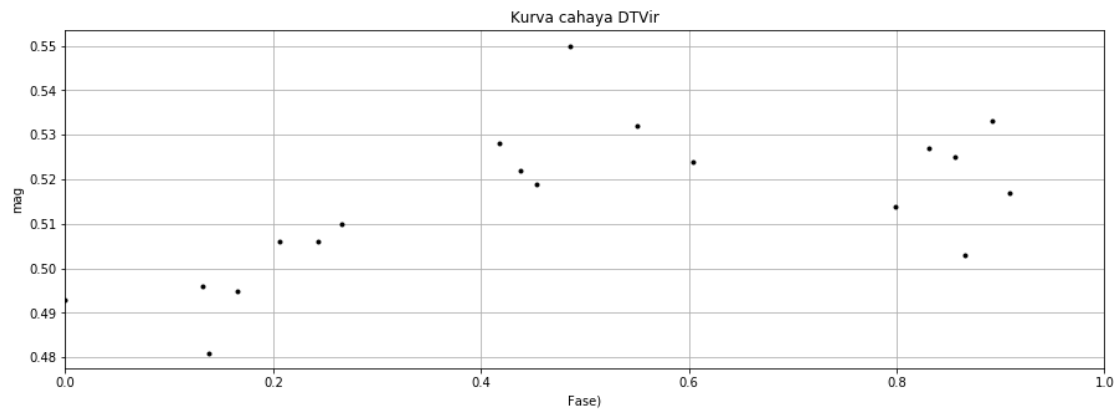
dari data di atas, plot magnitudo terhadap waktu



didapatkan plot hasil DCDF



Dari grafik di atas, dapat ditentukan frekuensi sistem ini adalah 0.351493 /hari. Sehingga hasil kurva cahaya dengan pemilihan periode = 1/frekuensi = 2.845004 hari menjadi

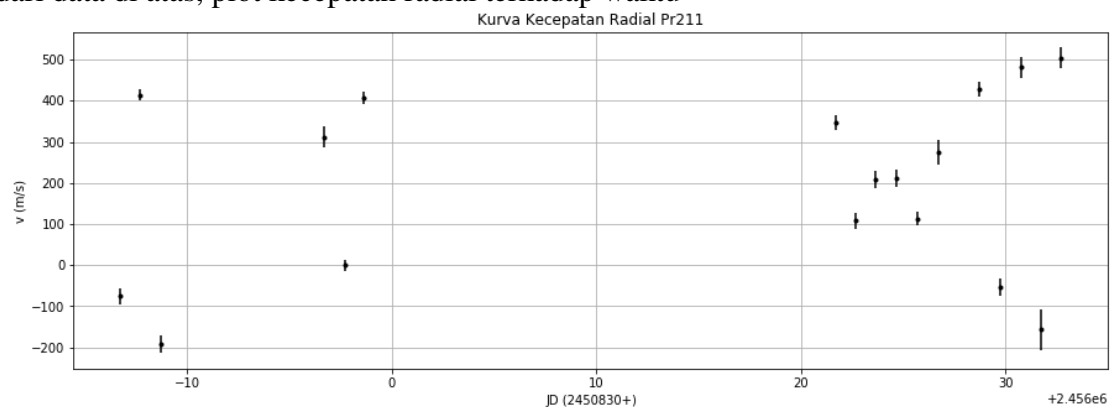


## 4.2 Kurva Kecepatan Radial Pr0211

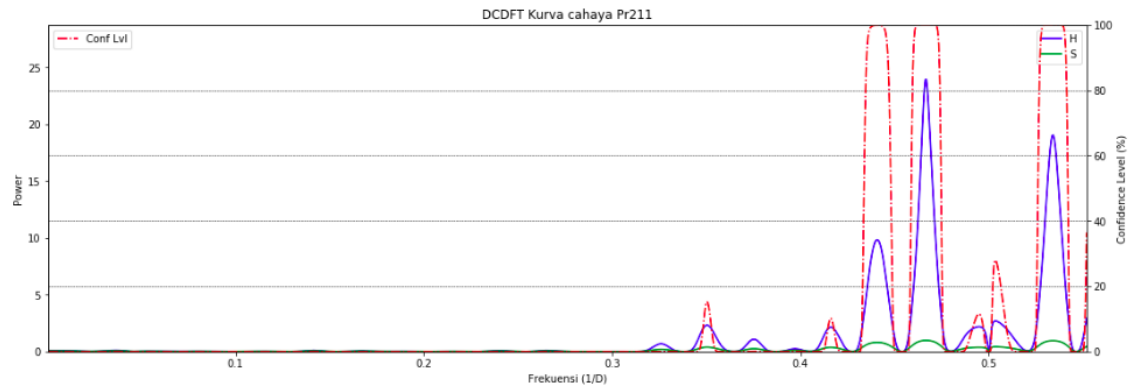
Data kecepatan radial Pr0211 dapat diperoleh melalui Nasa exoplanet archive [4] yang bersumber dari [3]. Di website tersebut terdapat 1072 data kurva kecepatan radial eksoplanet. Berikut data kecepatan radial Pr0211. kolom pertama adalah JD, kolom kedua adalah kecepatan radial (m/s), dan kolom ketiga adalah error kecepatan radial

2455986.772462	-75.70	19.10
2455987.720167	413.80	13.80
2455988.721641	-193.00	21.30
2455996.712491	311.80	24.90
2455997.756832	0.00	13.80
2455998.662801	405.80	14.70
2456021.718407	346.40	18.90
2456022.663113	108.00	20.30
2456023.652317	208.10	21.80
2456024.663763	212.00	21.40
2456025.714331	113.10	16.90
2456026.737245	275.70	30.00
2456028.698449	428.70	18.20
2456029.716113	-52.50	21.00
2456030.741942	480.60	26.50
2456031.720488	-156.90	50.50
2456032.715068	504.40	24.70
2456033.654723	-98.10	19.90

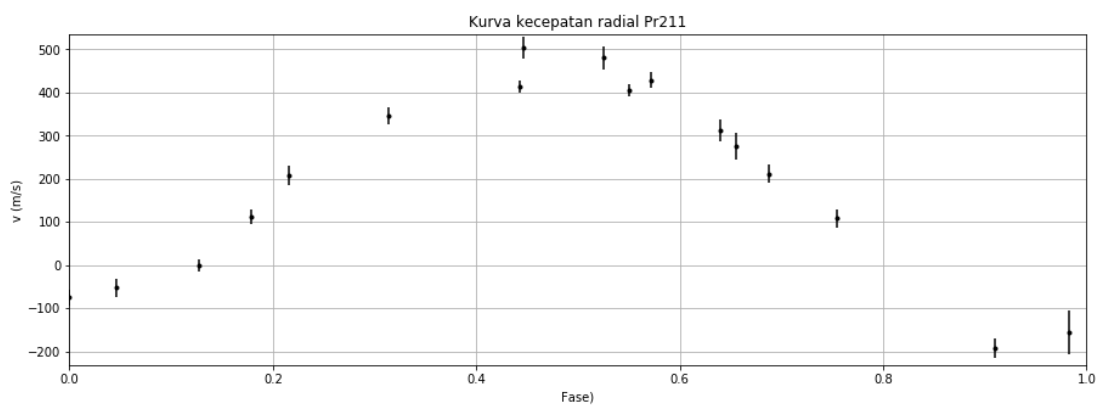
dari data di atas, plot kecepatan radial terhadap waktu



didapatkan plot hasil DCDFIT



Dari grafik di atas, dapat ditentukan frekuensi sistem ini adalah 0.466810 /hari. Sehingga didapat periode =  $1/\text{frekuensi} = 2.142199$  hari, cocok dengan hasil referensi  $2.1451 \pm 0.0012$  [3]. Plot kurva kecepatan radial dengan pemilihan periode tersebut



## Referensi

- [1] S Ferraz-Mello. "Estimation of periods from unequally spaced observations". In: *The Astronomical Journal* 86 (1981), p. 619.
- [2] Josip Kleczek. *Exercises in Astronomy: Revised and Extended Edition of "Practical Work in Elementary Astronomy" by MGJ Minnaert*. Springer Science & Business Media, 2012, p. 231.
- [3] Samuel N Quinn et al. "Two "b" s in the beehive: the discovery of the first hot Jupiters in an open cluster". In: *The Astrophysical Journal Letters* 756.2 (2012), p. L33.
- [4] *Radial Velocity Planet Resources in the Exoplanet Archive*. URL: <https://exoplanetarchive.ipac.caltech.edu/docs/rv.html>.