

**LAPORAN TUGAS ALGORITMA DAN PEMROGRAMAN**  
Laporan ini disusun untuk memenuhi penilaian mata kuliah Algoritma dan  
Pemrograman  
Program Studi S1 Teknik Fisika Universitas Telkom

**PUSAT MASSA**



**Disusun oleh :**  
**Kelompok 5**  
**TF 47-03**

- ANGGOTA :**
- 1. Ananda Farrel Rinaldi (101042300097)**
  - 2. Ardelia Indriana Elvandari (1104223095)**
  - 3. Khairunnisa Anaqoh (101042300022)**
  - 4. Kristhopia Nakesha Dapel (101042330053)**

# **Pusat Massa: Konsep dan Aplikasi dalam Fisika**

## **Pendahuluan**

Salah satu prinsip utama dalam fisika adalah konsep pusat massa, yang dapat digunakan dalam berbagai bidang, mulai dari mekanika klasik hingga teknologi kontemporer. Dalam analisis gerak benda, pusat massa, yang digambarkan sebagai titik rata-rata posisi massa dalam suatu sistem, sangat penting. Ide ini sangat penting untuk memahami gerak translasi dan rotasi benda, serta untuk menentukan stabilitas struktur dan respons dinamis terhadap gaya eksternal.

Laporan ini akan membahas teori pusat massa, derivasi matematisnya, dan aplikasinya dalam berbagai konteks fisika. Memahami pusat massa memungkinkan kita untuk memprediksi perilaku sistem partikel atau benda padat baik dalam kondisi statis maupun dinamis. Memahami pusat massa memungkinkan kita untuk membuat kemajuan di masa depan.

Tujuan dari laporan ini adalah untuk meningkatkan pemahaman kita tentang bagaimana pusat massa dapat membantu memecahkan masalah dunia nyata dan kemajuan sains dan teknologi. Mari kita mulai dengan mempelajari konsep dasar tentang pusat massa dan bagaimana konsep ini diterapkan dalam fisika.

## **Metode**

Metode yang digunakan adalah Metode Monte Carlo merupakan teknik komputasi yang digunakan untuk mensimulasikan perilaku sistem fisika dan matematika, termasuk perhitungan pusat massa. Metode ini sangat berguna dalam menyelesaikan masalah yang kompleks atau memiliki banyak variabel yang sulit dipecahkan dengan metode analitis atau numerik lainnya. Dalam konteks pusat massa, Metode Monte Carlo digunakan untuk mengestimasi posisi pusat massa suatu objek atau sistem partikel dengan cara berikut:

1. **Pembangkitan Sampel Acak:** Metode ini memulai dengan pembangkitan sejumlah besar titik atau sampel acak dalam ruang yang didefinisikan oleh objek atau sistem yang sedang diteliti.
2. **Penilaian Kondisi:** Setiap sampel yang dihasilkan kemudian dievaluasi untuk menentukan apakah sampel tersebut berada dalam objek atau memenuhi kondisi tertentu yang berkaitan dengan sistem.

3. Perhitungan Statistik: Dari sampel yang valid, dilakukan perhitungan statistik untuk mengestimasi pusat massa. Ini bisa melibatkan menghitung rata-rata posisi sampel atau menggunakan metode lain untuk menentukan titik pusat massa.
4. Estimasi dan Ketidakpastian: Hasil akhir dari simulasi Monte Carlo adalah estimasi pusat massa beserta ketidakpastian yang terkait, yang memberikan gambaran tentang seberapa akurat estimasi tersebut.

Metode Monte Carlo memiliki dua keuntungan besar, yang pertama adalah kemampuan untuk menangani sistem dengan bentuk atau batasan yang rumit. dan yang kedua adalah kemampuan untuk memberikan estimasi yang baik meskipun data mengandung ketidakpastian atau variabilitas. Selain itu, metode Monte Carlo sangat fleksibel dan dapat diterapkan untuk berbagai jenis masalah fisika dan matematika. Metode Monte Carlo telah digunakan dalam banyak bidang, termasuk fisika komputasi, perancangan aerodinamika, dan banyak lagi. Metode ini semakin populer dan dapat digunakan dengan lebih efisien berkat kemajuan teknologi komputer.

### **Hasil dan Pembahasan**

Dalam menganalisis data, identifikasi pola dan tren yang berkaitan dengan teori pusat massa. Beberapa operasi kodingan yang sudah didapatkan mungkin memiliki beberapa hasil. Dalam program ini untuk memecahkan persoalan masalah Pusat Massa ini menggunakan Metode Monte Carlo yang diimplementasikan dalam Pemrograman.

Adapun penjabaran dalam solusi yang digunakan pertama, melakukan perhitungan pusat massa dengan koordinat pusat lingkaran sebesar (0,0) serta jari-jari lingkaran yang digunakan bernilai 2, dari titik koordinat yang telah ditentukan. Pada program yang telah dijabarkan, terdapat 'double randfunc()', yang mana fungsi ini bertujuan untuk mengembalikan angka acak ataupun menghasilkan nilai acak selama program tersebut berjalan. Pada program ini angka acak yang digunakan untuk memberikan koordinat x yakni antara rentang -2 dan +2 dan pada koordinat y antara rentang -2 dan +2. Kemudian, menggunakan rumus standar untuk lingkaran yang berpusat pada titik asal radius 2 satuan.

$$x^2 + y^2 = 22$$

Kita dapat menggunakan rumus ini untuk mendemonstrasikan secara manual cara kerja teknik Monte Carlo bekerja. Melihat diagram sebelumnya, kita melihat bahwa lingkaran kita terdapat di dalam persegi. Karena bilangan acak kita berada di antara -2 dan +2 untuk x dan y, maka semuanya berada di dalam kotak itu. Saat kita menghasilkan angka, kita dapat menguji apakah angka tersebut ada di dalam lingkaran. Kita melakukan ini dengan mensubstitusi angka-angka yang dihasilkan x dan y ke dalam rumus sebelumnya.

Dalam tabel yang digunakan sebagai referensi perhitungan pusat massa ini, setiap pasangan (x, y) mewakili nilai yang kita hasilkan dan juga mewakili sebuah titik digrafik. Dalam program kami, kami menggunakan rumus  $x^2 + y^2 = 4$  diatur ulang menjadi y subjek rumus. Jadi kita mengerti  $y = \pm\sqrt{4 - x^2}$ . Kita menambahkan nilai x dan y ini ke dalam kolom xcofm dan ycofm yang mengakumulasi nilai positif dan negatif. Kami membaginya dengan jumlah total titik untuk memberikan koordinat pusat massa kita.

Kita juga dapat mengakumulasi nilai x dan y ke dalam xout dan yout dan menuliskannya ke dalam file kita untuk dicetak menggunakan paket Graph. Program juga dapat menambahkan nilai cetak kita pusat massa ke grafik ini menggunakan "Sisipkan Deret Titik".

### **Kesimpulan**

Kesimpulan dari laporan ini adalah bahwa Pusat Massa adalah suatu titik di mana massa suatu benda dapat dianggap terkonsentrasi sehingga perilakunya serupa dengan benda tersebut. Konsep ini memiliki aplikasi penting dalam pemahaman dan perhitungan gerak benda, terutama dalam konteks fisika.

Dalam laporan ini, terdapat pembahasan khusus mengenai pusat massa pada plat melingkar dan bentuk lainnya. Plat melingkar adalah contoh kasus di mana perhitungan pusat massa menjadi lebih kompleks, tetapi dengan menggunakan prinsip-prinsip fisika yang relevan, kita dapat menentukan lokasi pusat massa dengan tepat.

Selain itu, laporan ini mencantumkan implementasi konsep pusat massa dalam bahasa pemrograman C. Program tersebut dirancang untuk menghitung pusat massa dari suatu sistem benda, memberikan pemahaman praktis tentang bagaimana konsep teoritis dapat diaplikasikan dalam konteks pemrograman.

Secara keseluruhan, laporan ini menyoroti pentingnya konsep pusat massa dalam memahami dan memodelkan perilaku benda-benda dengan massa yang berbeda-beda. Implementasi dalam bahasa C menunjukkan bahwa konsep fisika dapat diaplikasikan dalam pengembangan program komputer, memperluas pemahaman kita tentang hubungan antara fisika dan dunia digital.

## Lampiran

x	y	$x^2 + y^2$	Is $x^2 + y^2 < 4$
0	0	$0+0=0$	yes
1	1	$1+1=2$	yes
1.9	1.9	$3.61+3.61 = 7.22$	no
1.8	1.8	$3.24+3.24 = 6.48$	no
1.7	1.7	$2.89+2.89 = 5.78$	no
1.6	1.6	$2.56+2.56 = 5.12$	no
1.5	1.5	$2.25+2.25 = 4.5$	no
1.4	1.4	$1.96+1.96 = 3.92$	yes
1	1.7	$1+2.89 = 3.89$	yes

### 1.Codingan

```
#define _CRT_SECURE_NO_WARNINGS

#include <stdlib.h>

#include <stdio.h>

#include <time.h>

#include <math.h>

double randfunc(); /* Function to return random number */

void main(){

    int l,outcount;

    float area,total,count;

    FILE *fptr;

    time_t t;

    /* Local arrays */

    double x, y,xout[3500],yout[3500],xcofm,ycofm;

    fptr=fopen("cofmc.dos","w");

    /* Intializes random number generator */

    srand((unsigned) time(&t));
```

```

/* clears arrays to zero */
for( l = 0; l<3500;l++)
{
xout[l] = 0.0;
yout[l] = 0.0;
}

/* Set x and y cofm accumulators to zero */
xcofm=0.0;
ycofm=0.0;
total = 0.0;
count = 0.0;
outcount = 0;
for( l = 1;l<= 3500;l++)
{
/* Call random number function */

/* Get x values between -2 and +2 */

/* Get y values between -2 and +2 */

x = randfunc()*4.0-2.0;
y = randfunc()*4.0-2.0;

/* If the generated x and y values show y is greater than */
/* -  $\sqrt{(4-x^2)}$  and less than +  $\sqrt{(4-x^2)}$ , then add 1 to count */
/* and update the x and y cofm values */
if(y>-sqrt(4-pow(x,2)) && y<sqrt(4-pow(x,2)))
{
xcofm=xcofm+x;

```

```

ycofm=ycofm+y;

total = total+1;

outcount = outcount +1;

xout[outcount] = x;

yout[outcount] = y;

}

count = count+1;

}

area=(total/count)16; / Area is part of the square which is 4x4 or
16 sq units */

printf("total is %f count is %f\n",total,count);

xcofm=xcofm/total;

ycofm=ycofm/total;

printf("area is %lf\n",area);

printf("cofm is %lf,%lf",xcofm,ycofm);

/* Plot the data */

if(outcount >= 2700)

outcount = 2700;

for(l = 1; l<=outcount-1;l++)

fprintf(fptr,"%lf %lf\n",xout[l],yout[l]);

fclose(fptr);

}

double randfunc()

{

/* Get a random number 0 to 1 */

```

```

double ans;
ans=rand()%1000;
ans=ans/1000;
return ans;
}

```

## 2.Data hasil coding

```

2667 0.660000 1.180000
2668 -0.420000 -0.004000
2669 0.528000 -0.404000
2670 -0.960000 0.012000
2671 -1.384000 0.496000
2672 -1.384000 1.072000
2673 0.736000 -1.116000
2674 0.968000 0.904000
2675 0.288000 -1.712000
2676 -0.072000 -0.748000
2677 -0.160000 1.696000
2678 -0.588000 -0.284000
2679 0.000000 0.224000
2680 -1.664000 0.888000
2681 0.516000 -0.624000
2682 -1.176000 -1.460000
2683 -0.720000 0.848000
2684 1.616000 -0.576000
2685 -0.264000 -0.336000
2686 -0.216000 -1.976000
2687 -0.724000 0.928000
2688 0.624000 -0.280000
2689 -1.164000 0.816000
2690 0.124000 -0.040000
2691 -0.408000 -1.312000
2692 -1.656000 -0.484000
2693 1.472000 0.576000
2694 0.060000 0.160000
2695 -0.576000 1.084000

```

## 3. Plotting

# Import the necessary packages and modules

```

import matplotlib.pyplot as plt
plt.style.use('style/sci.mplstyle')
import numpy as np

```

# Open and read the file gr.dos

```

a, b = np.loadtxt('cofmc.dat', unpack=True)

```

# Create figure object

```

plt.figure()

```

# Plot the DOS, in which the Fermi energy shifts to zero

```

plt.plot(a, b, color='red')

```

# Add the x and y-axis labels

```

plt.xlabel('a')

```

```

plt.ylabel('b')

```

# Set the axis limits

```

plt.xlim(-3, 3)

```

```

plt.ylim(-3, 3)

```

# Save the figure to the pdf file

```

plt.savefig('plot-dos.pdf', bbox_inches='tight')

```



#### 4. Hasil Output Plotting

