

Penjelasan Kode Inverse Kinematics 4 Omniwheel

Fazle Mawla Wahyuhanda / 5054241020

Langkah-1

Program diawali dengan `#include <iostream>`, `#include <math.h>`, dan `#include <fstream>`. `math.h` digunakan untuk mendefinisikan berbagai function matematika, seperti sin, cos, tan, dan lain-lain. `fstream` digunakan untuk membaca atau menulis ke sebuah file. Selanjutnya `using namespace std;` supaya memudahkan penulisan kode.

```
1. #include <iostream>
2. #include <math.h>
3. #include <fstream>
4.
5. using namespace std;
```

Langkah-2

Setelah itu kita mendefine macro PI dan rad yang akan digunakan di perhitungan nanti.

```
1. #define PI 3.14159265359
2. #define rad PI / 180
```

Langkah-3

Selanjutnya kita membuat class **Omniwheel** dengan awalnya membuat atribut.

```
1. class Omniwheel
2. {
3. public:
4.     float invkin[4][3];
5.     int speed[4];
```

Pertama kita membuat atribut matriks invkin yang akan kita isi dengan operasi perhitungan untuk mendapat hasil dari invers kinematic. Setelah itu ada atribut speed yang nantinya akan kita isi setelah dilakukan perhitungan

```
1. Omniwheel()
2. {
3.     invkin[0][0] = cos(45.0 * rad);
4.     invkin[0][1] = sin(45.0 * rad);
5.     invkin[0][2] = 1;
6.     invkin[1][0] = cos(135.0 * rad);
7.     invkin[1][1] = sin(135.0 * rad);
8.     invkin[1][2] = 1;
9.     invkin[2][0] = cos(225.0 * rad);
10.    invkin[2][1] = sin(225.0 * rad);
11.    invkin[2][2] = 1;
12.    invkin[3][0] = cos(315.0 * rad);
13.    invkin[3][1] = sin(315.0 * rad);
14.    invkin[3][2] = 1;
15. }
```

Kedua kita membuat constructor yang berisi nilai invers kinematic untuk 4 roda dengan sudut masing-masing 45° , 135° , 225° , dan 315° dengan menggunakan fungsi sin, cos, dan makro **rad**.

```
1. void ngitung(int translasi[3])
2. {
3.     for (int i = 0; i < 4; i++)
4.     {
5.         speed[i] = 0;
6.         for (int j = 0; j < 3; j++)
7.         {
8.             speed[i] += invkin[i][j] * translasi[j];
9.         }
10.    }
11. }
```

Ketiga kita membuat method **ngitung()** yang digunakan untuk menghitung kecepatan masing masing roda berdasarkan translasi x, y dan rotasi z dengan menggunakan rumus invers kinematic dan hasilnya disimpan di array **speed**.

```
1. void nampilin()
2. {
3.     for (int i = 0; i < 4; i++)
4.     {
5.         cout << "kecepatan roda " << i + 1 << " : " << speed[i] <<
6.         endl;
7.     }
```

Keempat kita membuat method **nampilin()** yang digunakan untuk menampilkan hasil perhitungan rumus invers kinematic yang disimpan di array **speed**.

```
1. void nyimpen(const string &filename)
2. {
3.     ofstream file(filename);
4.     for (int i = 0; i < 4; i++)
5.     {
6.         file << "kecepatan roda " << i + 1 << " : " << speed[i]
7.         << endl;
8.     }
9.     file.close();
10. }
```

Kelima kita membuat method **nyimpen()** yang digunakan untuk menyimpan hasil dari perhitungan tadi yang berupa kecepatan roda ke dalam file teks dengan nama yang diberikan di parameter.

Langkah-4

Terakhir kita membuat fungsi **main()** dan membuat objek **omni** dari kelas **Omniwheel**. Selanjutnya mengambil input berupa translasi x, y, dan rotasi z dan memanggil metode **ngitung**, **nampilin**, dan **nyimpen** untuk melakukan perhitungan, menampilkan hasil, dan menyimpan hasil ke dalam file.

```
1. int main()
2. {
3.     Omniwheel omni;
4.
5.     int input[3];
6.     cout << "Masukkan translasi x, y, dan rotasi z : ";
7.     cin >> input[0] >> input[1] >> input[2];
8.
9.     omni.ngitung(input);
10.    omni.nampilin();
11.    omni.nyimpen("hasil.txt");
12.
13.    return 0;
14. }
```