

# Knowledge Test Robotics Software Engineer Widya

## Robotics: PART A

Alif Wicaksana Ramadhan

September 10, 2024

**1. Uraikan apa yang dimaksud dengan rigid body transformation! Tuliskan matriks rotasi yang dapat memutar suatu rigid body pada sumbu z dengan sudut putar 30 derajat! Pada sistem ROS, pernahkah anda berhubungan dengan sistem robot yang melibatkan banyak link dan joint? Bagaimana strategi Anda untuk mengatasi permasalahan transformasi dalam sistem robot tersebut?**

Rigid body transformation adalah suatu upaya untuk mengubah posisi atau pose dari suatu objek rigid. Salah satu cara melakukan ini adalah dengan menggunakan operasi matriks, atau bisa disebut dengan matriks rotasi. Sebagai contoh, bentuk umum dari matriks rotasi yang memutar objek pada sumbu z dapat dilihat pada Persamaan 1.

$$R_{z,\gamma} = \begin{bmatrix} \cos \gamma & -\sin \gamma & 0 \\ \sin \gamma & \cos \gamma & 0 \\ 0 & 0 & 1 \end{bmatrix} \quad (1)$$

Untuk memutar objek sebanyak 30 derajat, bisa melakukan substitusi nilai 30 derajat pada  $\gamma$ , sehingga didapatkan matriks berikut:

$$R_{z,30} = \begin{bmatrix} \cos 30 & -\sin 30 & 0 \\ \sin 30 & \cos 30 & 0 \\ 0 & 0 & 1 \end{bmatrix} \quad (2)$$

yang jika disederhanakan, akan didapatkan matriks berikut ini sebagai bentuk akhir dari matriks transformasi 30 derajat pada sumbu z.

$$R_{z,30} = \begin{bmatrix} \frac{\sqrt{3}}{2} & -\frac{1}{2} & 0 \\ \frac{1}{2} & \frac{\sqrt{3}}{2} & 0 \\ 0 & 0 & 1 \end{bmatrix} \quad (3)$$

Karena saya sering bermain dengan mobile robot, pengalaman saya dalam menggunakan sistem ROS untuk robot yang melibatkan banyak link dan joint terbatas hanya pada saat kuliah. Pada saat itu saya menggunakan perhitungan manual, memanfaatkan DH-Parameters. Namun, ketika mempelajari lagi, saya menemukan terdapat beberapa package yang bisa digunakan untuk memudahkan hal tersebut. Package tersebut sudah disediakan oleh ROS. Salah satunya adalah dengan package *tf2\_ros* dan dengan mendefinisikan robot dalam file urdf. Namun, saya belum pernah mengerjakan proyek dengan aplikasi tersebut. Meski begitu, saya yakin saya bisa mempelajari dengan cepat.

2. Pada ruang 3 dimensi, diberikan dua vektor  $a = (a_1, a_2, a_3)$  dan  $b = (b_1, b_2, b_3)$ . Uraikan cara anda untuk menghitung sudut yang dibentuk oleh kedua vektor tersebut?

Untuk menghitung sudut yang terbentuk antara vektor  $a$  dan vektor  $b$ , bisa memanfaatkan perkalian dot matriks. Rumus yang bisa digunakan adalah sebagai berikut:

$$\cos \theta = \frac{\vec{a} \cdot \vec{b}}{|\vec{a}| |\vec{b}|} \quad (4)$$

Dengan  $\vec{a}$  adalah vektor  $a$ , dan  $\vec{b}$  adalah vektor  $b$ .  $|\vec{a}|$  dan  $|\vec{b}|$  adalah panjang dari masing-masing vektor tersebut. Terakhir,  $\theta$  adalah sudut yang dibentuk oleh kedua vektor tersebut.  $\vec{a} \cdot \vec{b}$  adalah perkalian dot matriks antara vektor  $a$  dan vektor  $b$ . Untuk mendapatkan nilai  $\theta$ , hanya perlu memindahkan operasi "cos" ke ruas kanan. Sehingga akan didapatkan persamaan sebagai berikut:

$$\theta = \arccos \left( \frac{\vec{a} \cdot \vec{b}}{|\vec{a}| |\vec{b}|} \right) \quad (5)$$

Untuk menggambarkan implementasinya dalam program/aplikasi, berikut pseudocode dari perhitungan sudut yang dibentuk oleh kedua vektor  $a$  dan  $b$ :

---

**Algorithm 1** Perhitungan sudut antara dua vektor

---

```

1: Input: Vec  $\vec{a}$ , Vec  $\vec{b}$ 
2: Output: Theta
3: if  $\text{len}(\vec{a}) \neq \text{len}(\vec{b})$  then
4:   return -1
5: end if
6:
7: dot = 0
8: A = 0
9: B = 0
10: for  $i$  in range  $\text{len}(\vec{a})$  do
11:   dot = dot +  $(\vec{a}_i * \vec{b}_i)$ 
12:   A = A +  $\vec{a}_i^2$ 
13:   B = B +  $\vec{b}_i^2$ 
14: end for
15:
16: A =  $\sqrt{\text{A}}$ 
17: B =  $\sqrt{\text{B}}$ 
18: theta =  $\arccos \left( \frac{\text{dot}}{\text{A} \times \text{B}} \right)$ 
19: return theta

```

---