Tugas 4 Sistem Robot Otonom: Simulasi Transformasi Koordinat Menggunakan Bahasa Pemrograman Python dan Pustaka NumPy

Nama: Falah Amru Dikasmara
NRP: 5022211041
Mata Kuliah: Sistem Robot Otonom

1 Pendahuluan

1.1 Latar Belakang

Proyek ini adalah simulasi transformasi koordinat menggunakan bahasa pemrograman python dan pustaka NumPy. Dalam proyek ini, sebuah titik awal pada kerangka acuan U direpresentasikan dalam bentuk koordinat homogen, kemudian ditransformasikan ke kerangka acuan lain B menggunakan kombinasi rotasi (*roll*, *pitch*, *yaw*) dan translasi, yang semuanya dibangun sebagai perkalian matriks homogen berukuran 4×4. Struktur kode dibuat modular dengan fungsi-fungsi khusus untuk menghasilkan matriks rotasi dan translasi, serta utilitas untuk konversi koordinat homogen, sehingga mudah dipahami sesuai teori transformasi koordinat. Dengan mengubah variabel konfigurasi di bagian awal program, pengguna dapat dengan cepat menguji berbagai kasus transformasi tanpa harus mengubah logika utama. Proyek ini bertujuan sebagai sarana edukasi dalam memahami bagaimana teori transformasi koordinat diwujudkan secara praktis dalam pemrograman.

1.2 Tujuan

Adapun tujuan dari proyek ini adalah :

1. Mempelajari konsep transformasi koordinat homogen dalam memodelkan perubahan posisi dan orientasi titik di ruang tiga dimensi.
2. Memahami cara membangun matriks rotasi dan translasi dalam bentuk matriks homogen berukuran 4×4.
3. Mengimplementasikan transformasi koordinat homogen menggunakan bahasa pemrograman Python dan pustaka *NumPy*.
4. Merancang program Python yang dapat menghitung posisi titik pada kerangka acuan baru setelah diberikan rotasi dan translasi tertentu.
5. Menyediakan program yang bersifat modular dan mudah digunakan sebagai sarana pembelajaran praktis mengenai teori transformasi koordinat.
6. Membuat laporan sebagai bahan pembelajaran mahasiswa

1.3 Fitur Proyek

Proyek ini memiliki beberapa fitur utama yang dirancang untuk memudahkan pemahaman dan penerapan teori transformasi koordinat homogen. Fitur pertama adalah adanya konfigurasi sederhana di bagian awal program, sehingga pengguna hanya perlu mengubah nilai titik, sudut rotasi, atau translasi untuk mencoba berbagai kasus tanpa harus memodifikasi logika inti. Fitur kedua adalah penggunaan representasi matriks homogen 4×4 yang menyatukan rotasi dan translasi ke dalam satu operasi perkalian matriks, sesuai dengan teori yang diajarkan. Selain itu, program dilengkapi dengan fungsi-fungsi modular yang terpisah untuk menghasilkan matriks rotasi di sekitar sumbu X, Y, dan Z, serta matriks translasi, sehingga struktur kodenya mudah dipelajari. Program ini juga memiliki opsi untuk menampilkan matriks transformasi antara (rotasi dan translasi) sebelum digabungkan, yang membantu pengguna dalam memverifikasi langkah perhitungan. Keseluruhan fitur ini membuat proyek tidak hanya berfungsi sebagai alat komputasi, tetapi juga sebagai media pembelajaran praktis untuk memahami konsep transformasi koordinat secara lebih intuitif.

1.4 Software yang Dibutuhkan dan Fungsinya

Adapun beberapa software yang dibutuhkan untuk membuat proyek ini antara lain :

1. Python : Sebagai bahasa pemrograman dan interpreter utama untuk menuliskan serta menjalankan program transformasi koordinat.
2. NumPy : Sebagai pustaka Python yang digunakan untuk melakukan operasi vektor dan matriks, khususnya dalam membentuk matriks homogen 4×4 dan melakukan perkalian matriks.
3. Editor Teks / IDE (misalnya VS Code atau PyCharm) : Untuk menulis, mengedit, dan menjalankan program Python dengan lebih terorganisir.
4. Command Prompt / Terminal : Sebagai sarana untuk mengeksekusi program Python dan menampilkan hasil perhitungan transformasi koordinat.

1.5 Tantangan

Dalam pengerjaan proyek ini, terdapat beberapa tantangan yang dihadapi, antara lain :

1. Memahami konsep dasar transformasi koordinat homogen agar dapat diimplementasikan dalam bentuk kode program.
2. Menentukan urutan rotasi yang sesuai (konvensi Euler ZYX) sehingga hasil transformasi sesuai dengan teori.
3. Mengimplementasikan operasi matriks menggunakan pustaka *NumPy* dengan sintaks yang tepat, khususnya perbedaan antara perkalian elemen per elemen (`*`) dan perkalian matriks (`@`).

1.6 Kemampuan yang Dipelajari

1. Pemrograman dalam bahasa Python untuk mengimplementasikan konsep transformasi koordinat homogen.
2. Penggunaan pustaka *NumPy* untuk membangun matriks rotasi dan translasi serta melakukan operasi perkalian matriks.
3. Pemahaman konsep koordinat homogen, matriks rotasi, dan translasi dalam ruang tiga dimensi.
4. Penyusunan program yang modular dan mudah digunakan sebagai sarana pembelajaran teori transformasi koordinat.

2 Dasar Teori

2.1 Python

Python merupakan bahasa pemrograman yang diciptakan Guido van Rossum dan pertama kali dirilis pada tahun 1991. Python digolongkan sebagai bahasa pemrograman tingkat tinggi karena cara penulisannya lebih dekat dengan bahasa manusia dan lebih jauh dari bahasa mesin, sehingga memberikan beberapa kelebihan. Pengguna Python tidak perlu mengatur detail teknis yang berhubungan dengan perangkat keras seperti alamat memori, register CPU, atau instruksi mesin, sehingga membuat baris perintah Python lebih ringkas daripada bahasa pemrograman lain untuk tugas yang sama. Sintaks pada Python juga mudah dipahami karena mirip dengan bahasa Inggris. Python dapat dijalankan pada berbagai sistem operasi karena memiliki interpreter yang dapat menerjemahkan kode ke instruksi yang sesuai dengan sistem yang dipakai. Berbagai kelebihan tersebut membuat penggunaan Python sangat populer bagi banyak orang dan mendorong dikembangkannya berbagai pustaka dan modul siap pakai untuk berbagai tugas, sehingga memungkinkan programmer fokus ke logika aplikasi daripada detail implementasi rendah.

2.2 NumPy

NumPy adalah pustaka Python untuk komputasi numerik. Inti NumPy adalah tipe data `ndarray`, yaitu larik (array) berdimensi-n dengan operasi vektor/matriks yang efisien. Beberapa hal penting:

- **Array** dibuat dengan `np.array([...], dtype=float)`.
- **Matriks identitas** dengan `np.eye(n)`.

- **Perkalian matriks** memakai operator `@` (bukan `*`).
- **Transpos** dengan `A.T`, dan blok-subarray dengan `A[:3, :3]`.

NumPy memudahkan kita membangun matriks rotasi, translasi, menggabungkannya menjadi matriks transformasi homogen 4×4, lalu mengalikan dengan vektor titik.

2.3 Vektor dan Matriks

Posisi titik di ruang tiga dimensi dinyatakan sebagai vektor kolom

$$\mathbf{p} = \begin{bmatrix} x \\ y \\ z \end{bmatrix}.$$

Matriks adalah susunan bilangan berbentuk persegi panjang yang dapat mewakili transformasi linier. Jika A adalah matriks yang sesuai dimensinya, maka hasil transformasi adalah $A\,\mathbf{p}$.

2.4 Kerangka Acuan, Orientasi, dan Pose

Kerangka acuan (frame) adalah sistem koordinat ortonormal (sumbu x, y, z saling tegak lurus) dengan sebuah titik asal. **Pose** suatu frame U relatif terhadap frame B terdiri dari:

1. **Orientasi**—dinyatakan oleh matriks rotasi 3×3 (kolom-kolomnya ortonormal dan determinannya $+1$).
2. **Posisi**—dinyatakan oleh vektor translasi $\mathbf{t} = \begin{bmatrix} t_x & t_y & t_z \end{bmatrix}^\top$.

Jika $\mathbf{p}$ adalah koordinat titik pada frame U, maka koordinat titik yang sama pada frame B adalah

$$\mathbf{p}_B = R\,\mathbf{p}_U + \mathbf{t},$$

dengan R matriks rotasi (orientasi U relatif terhadap B) dan $\mathbf{t}$ vektor translasi asal U relatif terhadap B.

2.5 Translasi

Translasi memindahkan setiap titik sejauh vektor tertentu. Jika $\mathbf{t} = \begin{bmatrix} t_x & t_y & t_z \end{bmatrix}^\top$, maka

$$\mathbf{p}' = \mathbf{p} + \mathbf{t}.$$

2.6 Matriks Rotasi (3D)

Rotasi mengubah *orientasi* vektor/titik terhadap sumbu tertentu. Tiga matriks rotasi elementer (konvensi tangan kanan) adalah:

$$R_x(\phi) = \begin{bmatrix} 1 & 0 & 0 \\ 0 & \cos\phi & -\sin\phi \\ 0 & \sin\phi & \cos\phi \end{bmatrix}, \quad R_y(\theta) = \begin{bmatrix} \cos\theta & 0 & \sin\theta \\ 0 & 1 & 0 \\ -\sin\theta & 0 & \cos\theta \end{bmatrix}, \quad R_z(\psi) = \begin{bmatrix} \cos\psi & -\sin\psi & 0 \\ \sin\psi & \cos\psi & 0 \\ 0 & 0 & 1 \end{bmatrix}.$$

2.7 Koordinat Homogen

Agar *rotasi dan translasi* dapat digabungkan dalam **satu** operasi, kita menambahkan komponen keempat w:

$$\tilde{\mathbf{p}} = \begin{bmatrix} x \\ y \\ z \\ 1 \end{bmatrix}.$$

Dengan bentuk ini, baik rotasi maupun translasi bisa diterapkan sebagai satu perkalian matriks 4×4 terhadap vektor homogen $\tilde{\mathbf{p}}$.

2.8 Matriks Transformasi Homogen

Gabungan rotasi R dan translasi $\mathbf{t}$ ditulis sebagai:

$$T = \begin{bmatrix} R & \mathbf{t} \\ \mathbf{0}^\top & 1 \end{bmatrix} \quad \text{(ukuran } 4 \times 4\text{)}.$$

Jika titik ditulis homogen $\tilde{\mathbf{p}} = \begin{bmatrix} x & y & z & 1 \end{bmatrix}^\top$, maka hasil transformasinya

$$\tilde{\mathbf{p}}' = T\,\tilde{\mathbf{p}} \quad \Longrightarrow \quad \mathbf{p}' = R\,\mathbf{p} + \mathbf{t}.$$

2.9 Urutan Rotasi (Yaw–Pitch–Roll, ZYX)

Orientasi umum sering dinyatakan sebagai urutan tiga rotasi elementer. Dalam proyek ini dipakai konvensi **ZYX** (yaw–pitch–roll):

$$R(\phi, \theta, \psi) = R_z(\psi)\, R_y(\theta)\, R_x(\phi).$$

Urutan sangat penting karena rotasi tidak komutatif (misal $R_z R_y \neq R_y R_z$). Dengan vektor kolom, *matriks paling kanan* bekerja *lebih dahulu*. Artinya, jika $T = \mathrm{Trans}(\mathbf{t})\, R_z R_y R_x$, maka pada vektor titik, urutannya adalah R_x lalu R_y, lalu R_z, baru translasi.

2.10 Membangun T dari Yaw–Pitch–Roll dan Translasi

Dengan konvensi di atas, transform penuh yang kita pakai dalam program adalah

$$T \;=\; \underbrace{\begin{bmatrix} I & \mathbf{t} \\ \mathbf{0}^\top & 1 \end{bmatrix}}_{\mathrm{Trans}(\mathbf{t})} \underbrace{R_z(\psi)}_{\text{yaw}} \underbrace{R_y(\theta)}_{\text{pitch}} \underbrace{R_x(\phi)}_{\text{roll}}.$$

Penerapannya ke titik homogen:

$$\tilde{\mathbf{p}}_B \;=\; T\, \tilde{\mathbf{p}}_U.$$

Pada implementasi, sudut biasanya diberikan dalam *derajat*, lalu dikonversi menjadi *radian* sebelum dipakai oleh fungsi trigonometri.

2.11 Komposisi Transformasi

Jika ada dua transformasi homogen T_1 lalu T_2, hasil gabungannya adalah

$$T = T_2\, T_1.$$

Matriks paling kanan bekerja terlebih dahulu pada vektor kolom. Prinsip ini persis dengan cara kita menyusun $R_z R_y R_x$ dan menambahkan translasi.

2.12 Transformasi Invers (Membalik Arah Transform)

Jika

$$T = \begin{bmatrix} R & \mathbf{t} \\ \mathbf{0}^\top & 1 \end{bmatrix}, \quad \text{maka} \quad T^{-1} = \begin{bmatrix} R^\top & -R^\top \mathbf{t} \\ \mathbf{0}^\top & 1 \end{bmatrix}.$$

Invers ini berguna bila kita perlu mengubah koordinat dari frame tujuan kembali ke frame asal.

2.13 Matriks Transformasi Untuk Bidang 2D

Pada bidang 2D (rotasi hanya sekitar z), bentuk homogen turun derajat menjadi 3×3:

$$R_\theta = \begin{bmatrix} \cos\theta & -\sin\theta & 0 \\ \sin\theta & \cos\theta & 0 \\ 0 & 0 & 1 \end{bmatrix}, \quad \mathrm{Trans}_{2D}(t_x, t_y) = \begin{bmatrix} 1 & 0 & t_x \\ 0 & 1 & t_y \\ 0 & 0 & 1 \end{bmatrix}.$$

3 Alur Kerja Program Simulasi Transformasi Koordinat

Program transformasi koordinat ini memiliki alur kerja sebagai berikut:

1. **Inisialisasi Variabel Konfigurasi**
   Pada bagian awal, pengguna menentukan nilai parameter transformasi:
   - Sudut rotasi *roll* (ϕ), *pitch* (θ), dan *yaw* (ψ) dalam derajat.
   - Vektor translasi $\mathbf{t} = [t_x, t_y, t_z]^\top$.
   - Titik yang akan ditransformasikan, ditulis sebagai vektor homogen $\tilde{\mathbf{p}} = [x, y, z, 1]^\top$.

   Semua variabel ini mudah diubah karena didefinisikan di awal program.

2. **Membangun Matriks Rotasi Homogen**
   Program membentuk tiga buah matriks rotasi homogen $R_x(\phi)$, $R_y(\theta)$, dan $R_z(\psi)$ sesuai sumbu x, y, dan z. Masing-masing matriks berukuran 4×4 dengan blok kiri-atas berisi rotasi 3×3 dan blok lainnya identitas homogen.

3. **Membangun Matriks Translasi Homogen**
   Program membentuk matriks translasi homogen $T(\mathbf{t})$ dengan bentuk:
   $$T(\mathbf{t}) = \begin{bmatrix} I_{3\times3} & \mathbf{t} \\ \mathbf{0}^\top & 1 \end{bmatrix}.$$

4. **Menyusun Matriks Transformasi Total**
   Matriks transformasi penuh dibentuk dengan mengalikan translasi dan rotasi sesuai urutan konvensi ZYX (yaw–pitch–roll):
   $$T = T(\mathbf{t}) \cdot R_z(\psi) \cdot R_y(\theta) \cdot R_x(\phi).$$
   Urutan ini penting karena operasi rotasi tidak bersifat komutatif.

5. **Menerapkan Transformasi ke Titik**
   Titik masukan $\tilde{\mathbf{p}}$ dalam homogen dikalikan dengan matriks transformasi:
   $$\tilde{\mathbf{p}}' = T \cdot \tilde{\mathbf{p}}.$$
   Hasil $\tilde{\mathbf{p}}'$ kemudian dikonversi kembali menjadi koordinat Kartesius (x', y', z') dengan cara mengambil tiga komponen pertama.

6. **Menampilkan Hasil Transformasi**
   Program menampilkan:
   - Matriks transformasi homogen T.
   - Titik sebelum transformasi (x, y, z).
   - Titik setelah transformasi (x', y', z').

   Dengan demikian, pengguna dapat langsung melihat efek rotasi dan translasi yang diterapkan pada titik.

4 Tutorial Penggunaan Program

Agar program dapat dijalankan dengan baik, ikuti langkah-langkah berikut:

1. **Instalasi Python dan NumPy**
   Pastikan Python telah terpasang pada sistem. Versi Python yang direkomendasikan adalah minimal 3.8. Selanjutnya, pasang pustaka NumPy dengan perintah:
   ```
   pip install numpy
   ```

2. **Persiapan Folder Kerja**
   Simpan file program `.py` beserta file pendukung (jika ada) di dalam satu folder kerja agar mudah dijalankan dan dikelola.

3. **Edit Bagian Konfigurasi Program**
   Buka file program menggunakan editor teks (misalnya VS Code, PyCharm, atau Notepad++). Pada bagian awal program, ubah nilai variabel konfigurasi sesuai kebutuhan:
   - Sudut rotasi: `roll`, `pitch`, dan `yaw` (dalam derajat).
   - Vektor translasi: `tx`, `ty`, `tz`.
   - Titik awal yang akan ditransformasi: `px`, `py`, `pz`.

4. **Menjalankan Program**
   Jalankan program dengan perintah:
   ```
   python nama_file.py
   ```
   Gantilah `nama_file.py` dengan nama file program Anda.

5. **Melihat Hasil**
   Setelah dijalankan, program akan menampilkan:
   - Matriks transformasi homogen 4×4 yang terbentuk.
   - Koordinat titik sebelum transformasi.
   - Koordinat titik setelah transformasi.

   Hasil tersebut ditampilkan pada terminal atau command prompt.

5 Perbandingan Hasil Program Terhadap Perhitungan Teoritis

Bentuk Umum

Rotasi terhadap sumbu-z sebesar γ dan translasi $\mathbf{t} = \mathbf{0}$ dituliskan sebagai:

$$T(\gamma) = \begin{bmatrix} R_z(\gamma) & \mathbf{0} \\ \mathbf{0}^\top & 1 \end{bmatrix} = \begin{bmatrix} \cos\gamma & -\sin\gamma & 0 & 0 \\ \sin\gamma & \cos\gamma & 0 & 0 \\ 0 & 0 & 1 & 0 \\ 0 & 0 & 0 & 1 \end{bmatrix}.$$

Titik homogen $\tilde{\mathbf{p}}_U = [x_U \; y_U \; z_U \; 1]^\top$ ditransformasikan menjadi

$$\tilde{\mathbf{p}}_B \;=\; T(\gamma)\,\tilde{\mathbf{p}}_U \quad \Rightarrow \quad \begin{bmatrix} x_B \\ y_B \\ z_B \\ 1 \end{bmatrix}.$$

Case 1: $(x_U, y_U, z_U) = (2, 0, 0),\ \gamma = 90°$

$$T(90°) = \begin{bmatrix} 0 & -1 & 0 & 0 \\ 1 & 0 & 0 & 0 \\ 0 & 0 & 1 & 0 \\ 0 & 0 & 0 & 1 \end{bmatrix}, \qquad \tilde{\mathbf{p}}_U = \begin{bmatrix} 2 \\ 0 \\ 0 \\ 1 \end{bmatrix}$$

$$\tilde{\mathbf{p}}_B = \begin{bmatrix} 0 & -1 & 0 & 0 \\ 1 & 0 & 0 & 0 \\ 0 & 0 & 1 & 0 \\ 0 & 0 & 0 & 1 \end{bmatrix} \begin{bmatrix} 2 \\ 0 \\ 0 \\ 1 \end{bmatrix} = \begin{bmatrix} 0 \\ 2 \\ 0 \\ 1 \end{bmatrix} \Rightarrow (x_B, y_B, z_B) = (0, 2, 0).$$

Sedangkan program mendapatkan hasil seperti berikut :

```
T (Trans @ Rz @ Ry @ Rx) =
[[ 0. -1.  0.  0.]
 [ 1.  0.  0.  0.]
 [ 0.  0.  1.  0.]
 [ 0.  0.  0.  1.]]

u_P = (2.0, 0.0, 0.0)
b_P = [0. 2. 0.]
```

Figure 1: Case 1

Case 2: $(x_U, y_U, z_U) = (0, 2, 0),\ \gamma = 90°$

$$\tilde{\mathbf{p}}_U = \begin{bmatrix} 0 \\ 2 \\ 0 \\ 1 \end{bmatrix}, \qquad \tilde{\mathbf{p}}_B = \begin{bmatrix} 0 & -1 & 0 & 0 \\ 1 & 0 & 0 & 0 \\ 0 & 0 & 1 & 0 \\ 0 & 0 & 0 & 1 \end{bmatrix} \begin{bmatrix} 0 \\ 2 \\ 0 \\ 1 \end{bmatrix} = \begin{bmatrix} -2 \\ 0 \\ 0 \\ 1 \end{bmatrix} \Rightarrow (x_B, y_B, z_B) = (-2, 0, 0).$$

Sedangkan program mendapatkan hasil seperti berikut :

Case 3: $(x_U, y_U, z_U) = (2, 0, 2),\ \gamma = 90°$

$$\tilde{\mathbf{p}}_U = \begin{bmatrix} 2 \\ 0 \\ 2 \\ 1 \end{bmatrix}, \qquad \tilde{\mathbf{p}}_B = \begin{bmatrix} 0 & -1 & 0 & 0 \\ 1 & 0 & 0 & 0 \\ 0 & 0 & 1 & 0 \\ 0 & 0 & 0 & 1 \end{bmatrix} \begin{bmatrix} 2 \\ 0 \\ 2 \\ 1 \end{bmatrix} = \begin{bmatrix} 0 \\ 2 \\ 2 \\ 1 \end{bmatrix} \Rightarrow (x_B, y_B, z_B) = (0, 2, 2).$$

Sedangkan program mendapatkan hasil seperti berikut :

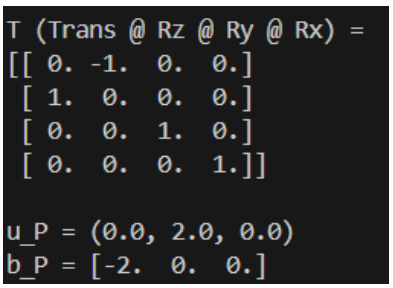

Figure 2: Case 2

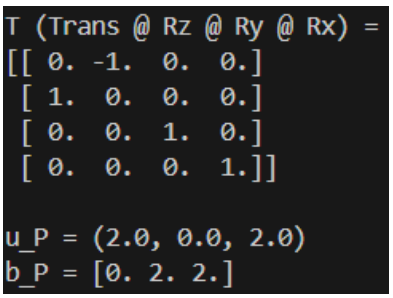

Figure 3: Case 3

Case 4: $(x_U, y_U, z_U) = (2, 0, 2),\ \gamma = -180°$

Karena $\cos(-180°) = -1$ dan $\sin(-180°) = 0$,

$$T(-180°) = \begin{bmatrix} -1 & 0 & 0 & 0 \\ 0 & -1 & 0 & 0 \\ 0 & 0 & 1 & 0 \\ 0 & 0 & 0 & 1 \end{bmatrix}, \qquad \tilde{\mathbf{p}}_U = \begin{bmatrix} 2 \\ 0 \\ 2 \\ 1 \end{bmatrix}$$

$$\tilde{\mathbf{p}}_B = \begin{bmatrix} -1 & 0 & 0 & 0 \\ 0 & -1 & 0 & 0 \\ 0 & 0 & 1 & 0 \\ 0 & 0 & 0 & 1 \end{bmatrix} \begin{bmatrix} 2 \\ 0 \\ 2 \\ 1 \end{bmatrix} = \begin{bmatrix} -2 \\ 0 \\ 2 \\ 1 \end{bmatrix} \Rightarrow (x_B, y_B, z_B) = (-2, 0, 2).$$

Sedangkan program mendapatkan hasil seperti berikut :

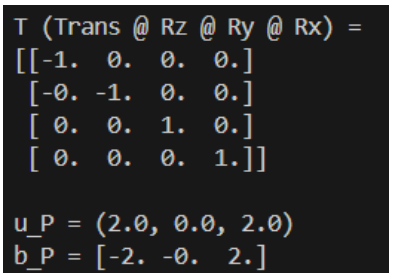

Figure 4: Case 4

Ringkasan

Case	$(x_U, y_U, z_U),\ \gamma$	(x_B, y_B, z_B)
1	$(2, 0, 0),\ 90°$	$(0, 2, 0)$
2	$(0, 2, 0),\ 90°$	$(-2, 0, 0)$
3	$(2, 0, 2),\ 90°$	$(0, 2, 2)$
4	$(2, 0, 2),\ -180°$	$(-2, 0, 2)$

6 Kesimpulan

Berdasarkan implementasi dan pengujian yang telah dilakukan, dapat disimpulkan bahwa program berhasil menjalankan tugasnya dalam melakukan transformasi koordinat tiga dimensi dengan menggunakan pendekatan matriks homogen. Program mampu:

1. Menerapkan rotasi dengan urutan yaw–pitch–roll sesuai teori matriks rotasi.
2. Menggabungkan rotasi dan translasi ke dalam satu operasi matriks transformasi homogen berukuran 4×4.
3. Mentransformasikan titik masukan menjadi titik keluaran dengan hasil yang konsisten dengan perhitungan teoretis.
4. Menampilkan hasil transformasi dengan jelas, baik dalam bentuk matriks transformasi maupun koordinat titik sebelum dan sesudah transformasi.

- Link Github :

  https://github.com/amrufal/transformasi_koordinat

- Linkedin :

  https://www.linkedin.com/posts/falah-amru-9a192a385_proyek-ini-membahas-implementasi-transformasi-
  utm_source=share&utm_medium=member_desktop&rcm=ACoAAF7_5wkBHEm5eAEQ0al5sPz8ty6cgqgTe08