

## 2. DASAR TEORI

### 2.1 Robot Industri

Robot industri adalah sistem robot yang digunakan untuk manufaktur. Mereka adalah otomatis, dapat diprogram, dan mampu bergerak pada tiga atau lebih sumbu. Aplikasi khas robot industri meliputi pengelasan, pengecatan, perakitan, pemilihan dan penempatan, pemeriksaan produk, dan pengujian, semuanya dilakukan dengan daya tahan tinggi, kecepatan, dan ketepatan. Robot industri dapat membantu dalam penanganan material dan memiliki berbagai kategori, termasuk robot untuk perakitan, pengelasan, pengecatan, dan dukungan operator. Mereka juga dapat dibagi menjadi beberapa tingkat tergantung pada jenis pemrosesan yang akan diimplementasikan dan tingkat otonomi (Wikipedia).

### 2.2 Robot Manipulator

Robot manipulator adalah robot yang memiliki bentuk seperti halnya lengan manusia yang berfungsi untuk membantu manusia. Robot ini biasanya digunakan untuk memindahkan dan mengangkat barang berat dan berskala besar serta harus dengan kecepatan dan ketepatan yang akurat.

Robot manipulator dapat terkendalikan dengan otomatis dan juga manual. Otomatis robot merupakan robot yang bergerak sesuai perintah dari sistem tanpa campur tangan manusia. Manual robot merupakan robot yang bergerak menggunakan sistem yang dikontrol oleh seorang operator.

Lengan robot biasanya terdiri atas bahu, persendian, dan tangan. Bagian tangan robot biasanya disebut lengan manipulator, yaitu sistem gerak yang berfungsi untuk manipulasi (memegang, mengambil, mengangkat, memindahkan, mengolah) objek. Untuk mengambil barang biasanya tangan robot dilengkapi dengan *gripper* (pemegang) yang berupa jari-jari.

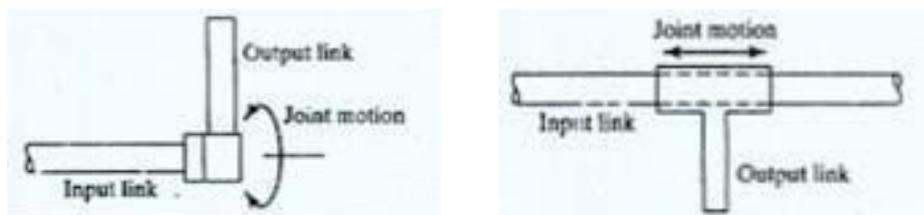
Lengan robot biasanya didesain sesuai gerakan yang dibutuhkan. Pengontrol dibuat dengan *potensiometer* untuk persendian lengan dan *Flex Sensor* yang diletakkan pada

jari-jari manusia dengan cara membuat pengendali yang sesuai dengan bentuk lengan dan jari-jari manusia agar dapat digunakan sebagai penggerak sendi-sendi pada lengan robot.

### 2.2.1 Robot Manipulator Serial

Robot manipulator serial adalah jenis robot manipulator yang umum digunakan di industri. Robot ini dirancang sebagai serangkaian link yang terhubung oleh sendi-sendi yang diaktifkan oleh motor, yang membentang dari sebuah basis hingga end-effector. Biasanya, robot ini memiliki beberapa keuntungan dan kelemahan. Keuntungan utamanya adalah ruang kerja yang besar dibandingkan dengan ukuran robot dan ruang lantai yang ditematinya. Namun, kelemahannya termasuk kekakuan rendah, akumulasi kesalahan dari link ke link, dan beban yang harus diangkat dan dipindahkan oleh sebagian besar aktuator. Robot manipulator serial umumnya digunakan untuk berbagai aplikasi industri seperti pengecatan, pengangkutan barang, pengelasan, dan pengambilan. (Teknovidia)

Mobilitas dari sebuah robot sangat ditentukan dengan kehadiran sendi dari robot tersebut. Secara umum, sendi dari robot dibagi menjadi dua macam, yaitu sendi prismatic dan sendi putar. Sendi prismatic akan membuat dimensi yang panjang dari bodi kaku robot bisa berubah, namun orientasi nya tidak berubah, sedangkan sendi putar akan membuat orientasi dari sebuah bodi kaku bisa berubah tetapi dimensi panjang dari bodi kaku robot tidak bisa berubah. (Suciliano et al., 2009) Namun perlu diketahui bahwa sendi pada robot manipulator tidak hanya dua jenis tersebut saja, dan masih banyak model sendi yang lainnya, seperti sendi peluru (spherical)

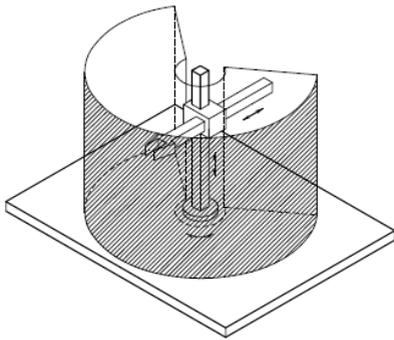


Gambar 2. 1 Jenis-Jenis Sendi pada Robot Manipulator (Sendi Putar Sebelah Kiri dan Sendi Prismatic Sebelah Kanan)

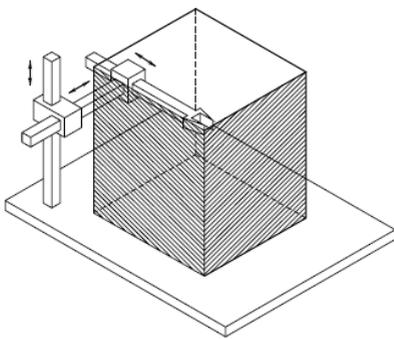
Sumber : Elinsmkamga et al., (2017)

Dengan kombinasi sendi putar dan sendi prismatic maka didapatkan secara umum terdapat 5 jenis manipulator. Kelima jenis manipulator tersebut adalah sebagai berikut:

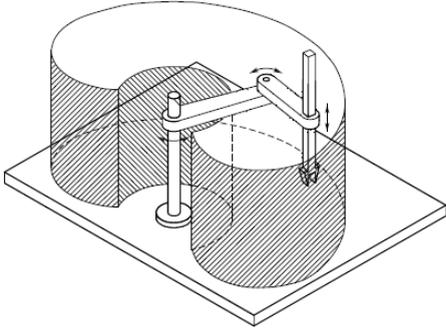
1. Manipulator *Silindris*
2. Manipulator *Kartesian*
3. Manipulator SCARA
4. Manipulator *Spherical*
5. Manipulator *Anthropomorphic*



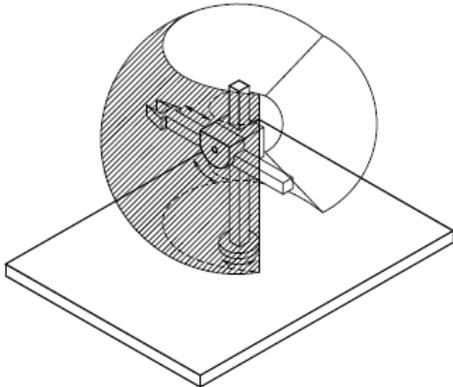
Gambar 2. 2 Manipulator Silindris dan Ruang Kerjanya)  
Sumber: Siciliano et al., (2009)



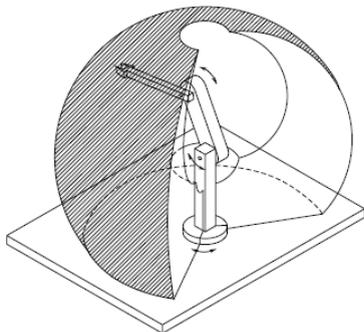
Gambar 2. 3 Manipulator Kartesian dan Ruang  
Sumber: Siciliano et al., (2009)



Gambar 2. 4 Manipulator SCARA dan Ruang  
Sumber: Siciliano et al., (2009)



Gambar 2. 5 Manipulator Spherical dan Ruang Kerjanya  
Sumber: Siciliano et al., (2009)



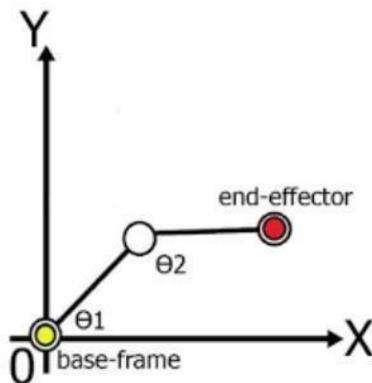
Gambar 2. 6 Manipulator Anthropomorphic dan Ruang Kerjanya  
Sumber: Siciliano et al., (2009)

Struktur silinder menyediakan kekakuan mekanis yang optimal. Ketepatan posisi pergelangan tangan cenderung menurun seiring peningkatan goresan horizontal. Ruang kerja sebagian besar terletak di dalam silinder berongga (lihat Gambar 2.1). Geometri Cartesian dihasilkan melalui tiga sambungan prismatic, di mana sumbu-sumbunya saling bersilangan secara ortogonal (lihat Gambar 2.2). Dengan mempertimbangkan sederhananya geometri ini, setiap Derajat Kebebasan (DOF) terkait dengan variabel ruang Cartesian, sehingga sangat beralasan untuk menjalankan pergerakan lurus dalam ruang. Geometri khusus merujuk pada geometri SCARA yang dapat direalisasikan dengan menghilangkan dua sambungan berputar dan satu sambungan prismatic, sehingga semua sumbu geraknya menjadi sejajar (lihat Gambar 2.3). Geometri bola dapat dibedakan dari geometri silinder karena pada geometri bola, dua sambungan prismatic digantikan dengan sambungan berputar (lihat Gambar 2.4). Geometri antropomorfik diwujudkan melalui tiga sambungan berputar; sumbu revolusi pada sambungan pertama bersilangan ortogonal dengan sumbu dua sambungan lainnya yang berjalan sejajar (lihat Gambar 2.5). Mirip dengan struktur lengan manusia, sendi kedua disebut sendi bahu, sementara sendi ketiga disebut sendi siku, menghubungkan "lengan" dengan "lengan bawah." Struktur antropomorfik ini sangat fleksibel karena semua sendinya berputar. (Siciliano et al., 2009)

### 2.3 Kinematika Robot

Kinematika mempelajari gerak dan pengembangan persamaan untuk menggambarkan gerakan benda-benda tanpa mempertimbangan gaya (*forces*), penyebab gerakan atau interaksi yang mungkin terjadi selama gerakan. Rantai Kinematika (*kinematic chain*) adalah manipulator buatan yang berinteraksi dengan lingkungan dan biasanya digambarkan sebagai rakitan *frame* robot yang bersambungan dengan sendi putar. Sendi berputar dan memberikan kontrol posisi sudut relatif dari manipulator. Tidak semua kombinasi posisi sendi dalam suatu rantai kinematika valid, karena beberapa kombinasi menyebabkan tabrakan antar *frame* atau dengan lingkungan yang bersifat tetap seperti lantai atau dinding. Semua kombinasi valid dari rantai kinematika membentuk ruang sendi (*joint space*). Sedangkan banyaknya sendi dalam rantai kinematika disebut *Degree of Freedom* (DOF) atau derajat kebebasan. Semakin banyak jumlah DOF yang ada membuat fleksibilitas gerakan manipulator menjadi lebih banyak. (repository.uksw et al., 2020)

Kinematika robot adalah aplikasi geometri untuk mempelajari rantai kinematika dengan beberapa DOF. Kinematika robot memberikan transformasi dari *joint space* yang didefinisikan ke ruang Kartesian dimana robot manipulator bergerak. Kinematika robot cukup berguna, karena dapat digunakan untuk perencanaan dan pelaksanaan gerakan. (repository.uksw et al., 2020)



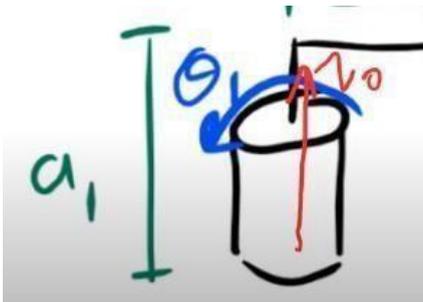
Gambar 2. 7 Rantai Kinematika

### 2.3.1 Diagram Kinematika

Secara umum, diagram kinematika merujuk pada representasi visual yang mengilustrasikan koneksi antara bodi kaku robot dan sendi ketika nilai semua sendi adalah nol. Pada praktiknya, bodi kaku robot mencakup semua komponen dalam robot yang menentukan jarak antara alas dan sendi pertama, antara sendi-sendi, serta antara sendi dan end-effector. Representasi bodi kaku robot dilambangkan dengan garis lurus, dan panjang garis tersebut diwakili oleh simbol "a". Penting untuk mencatat bahwa setiap bodi kaku robot harus diberi indeks yang berbeda, dimulai dari indeks 1 (Sodemann et al., 2017).

Sebagaimana telah dijelaskan pada bagian 2.2.1, sendi pada manipulator robot dibagi menjadi dua jenis, yakni sendi putar dan sendi prismatic. Pada diagram kinematika, sendi putar direpresentasikan dengan bentuk silindris, sementara sendi prismatic diilustrasikan dengan sebuah kubus yang memiliki persegi menunjukkan arah positif pergerakan sendi prismatic. Dalam penggambaran ini, arah putaran sendi harus mengikuti arah positif putaran, sesuai

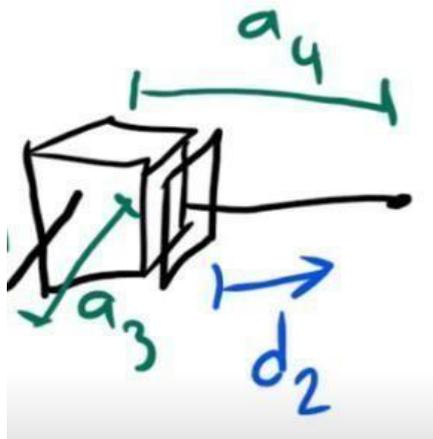
dengan kaidah tangan kanan pada sumbu putar. Sudut putaran pada sendi putar disimbolkan dengan variabel  $\theta$ . Untuk kemudahan penamaan, simbol  $\theta$  akan digantikan dengan simbol  $q$  mulai dari bab 3 dan seterusnya. Arah pergerakan sendi prismatic digambarkan dengan anak panah yang keluar dari persegi. Jarak pergerakan sendi prismatic diukur dengan variabel disimbolkan dengan huruf “d”. Penomoran indeks untuk sendi disesuaikan dengan urutan posisi sendi tersebut, dimulai dari indeks 1 untuk sendi yang terletak paling dekat dengan landasan. Ilustrasi dari sendi putar dan sendi prismatic dapat ditemukan pada gambar 2.9. (Sodemann et al., 2017)



Gambar 2. 8 Sendi Putar

Sumber: Sodemann et al., (2017, Juny 7). *Robotics 1 U1 (Kinematics) S2 (Kinematics Diagram)*. [Video]. Youtube.

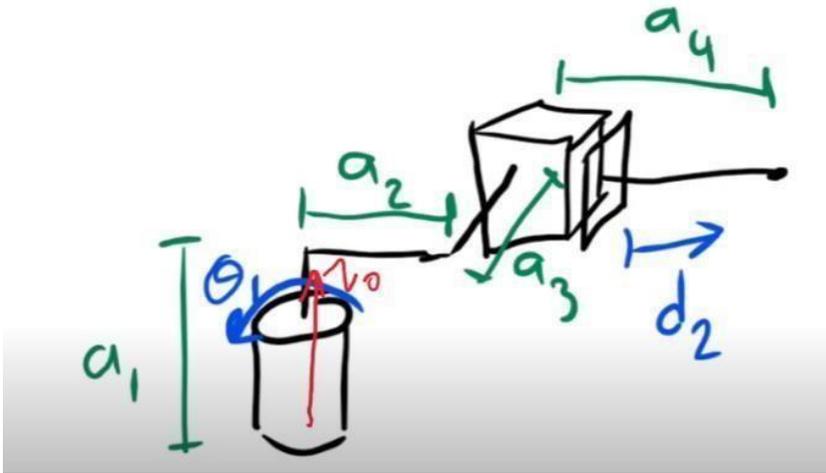
<https://www.youtube.com/watch?v=ut4uZ6Yzv6o>



Gambar 2. 9 Sendi Prismatic

Sumber: Sodemann et al., (2017, Juny 7). *Robotics 1 U1 (Kinematics) S2 (Kinematics Diagram)*. [Video]. Youtube.

<https://www.youtube.com/watch?v=ut4uZ6Yzv6o>



Gambar 2. 10 Robot Manipulator dengan 2 Sendi

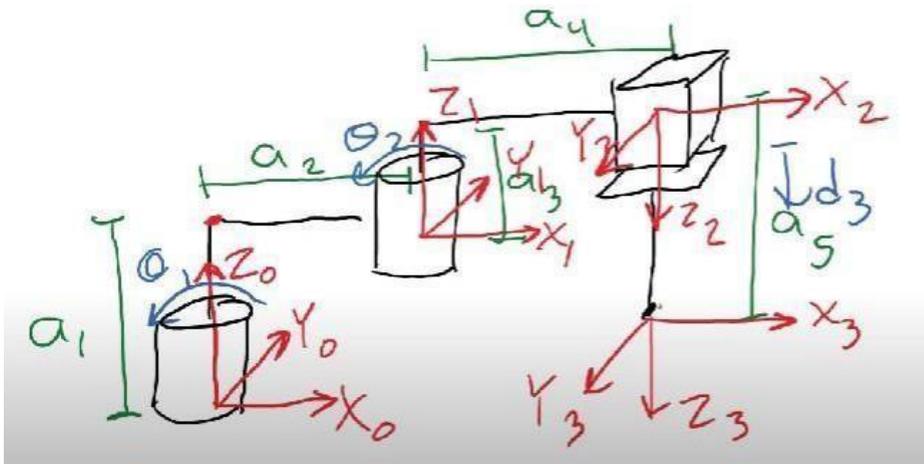
Sumber: Sodemann et al., (2017, Juny 7). *Robotics 1 U1 (Kinematics) S2 (Kinematics Diagram)*. [Video]. Youtube.

<https://www.youtube.com/watch?v=ut4uZ6Yzv6o>

Pada gambar 2.8 terlihat bahwa pada sambungan putar terdapat suatu jarak  $a_1$  dengan sudut putar variabel  $\theta_1$ , dan perputarannya berlawanan arah jarum jam. Sementara itu, gambar 2.9 menunjukkan bahwa sambungan prismatic memiliki jarak  $d_1$  sebagai variabelnya. Gambar 2.10 memberikan contoh penamaan variabel untuk setiap sambungan dan dimensi tubuh kaku dari manipulator serial robot. (Sodemann, 2017a)

Setelah mendapatkan posisi masing-masing sendi dan bodi kaku, langkah berikutnya adalah melakukan penggambaran kerangka sumbu x, sumbu y, dan sumbu z. Penggambaran dimulai dari kerangka dasar tempat robot ditempatkan, dan setiap sendi harus memiliki satu kerangka. Kerangka terakhir berada pada end-effector. Penetapan sumbu pada kerangka ini mengikuti aturan Denavit-Hartenberg, yang terdiri dari empat aturan sebagai berikut (Sodemann, 2017a):

- a. Arah sumbu z harus sejajar dengan sumbu rotasi atau arah pergerakan linier dari sendi prismatic.
- b. Sumbu x harus tegak lurus terhadap sumbu z dari kerangka itu sendiri dan kerangka sebelumnya.
- c. Semua kerangka harus mengikuti aturan tangan kanan untuk menentukan sumbu y masing-masing kerangka.
- d. Setiap perpanjangan sumbu x harus berpotongan dengan perpanjangan sumbu z pada kerangka sebelumnya.



Gambar 2. 11 Kerangka pada Diagram Kinematika Robot Manipulator

Sumber: Sodemann, (2017, Juny 7a). *Robotics 1 U1 (Kinematics) S2 (Kinematics Diagram)*. [Video]. Youtube.

<https://www.youtube.com/watch?v=ut4uZ6Yzv6o>

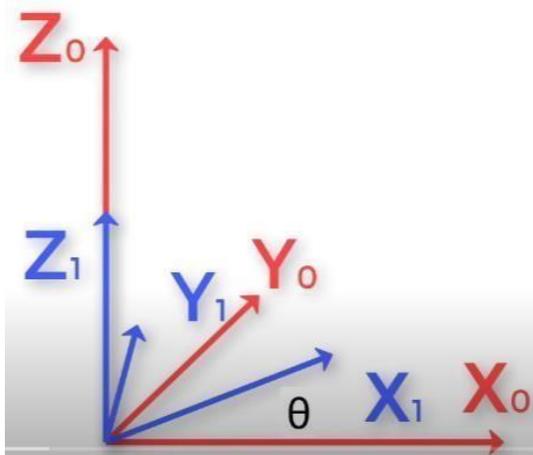
### 2.3.2 Forward Kinematic

*Forward Kinematic*, seperti yang dijelaskan pada gambar 2.2 merupakan metode untuk bisa menemukan posisi dan orientasi dari kerangka *end-effector* terhadap kerangka landasan jika terjadi pada pergerakan pada masing-masing sendi pada robot manipulator. Salah satu faktor permasalahan utama dari robot lengan ini adalah dibutuhkan ketelitian dalam menentukan posisi dari objek yang akan dipindahkan. Salah satu perhitungan yang dapat digunakan adalah perhitungan *forward kinematics*. Perhitungan *forward kinematics* merupakan suatu perhitungan yang mana sudut gerak dari lengan robot sudah ditentukan kemudian dapat dicari posisi akhir lengan robot. Dalam proses, ada dua metode yang dapat digunakan untuk bisa membuat sebuah persamaan yang bisa menggambarkan posisi dan orientasi dari kerangka *end-effector* terhadap kerangka landasan. Metode pertama yang dipakai adalah metode parameter Denavit-Hartenberg, Keuntungan dari metode ini adalah memerlukan lebih sedikit angka yang menunjukkan parameter posisi dan orientasi pada *end-effector*. Metode kedua adalah dengan rumus *product of exponential (PoE)*, keuntungan metode ini adalah tidak perlu adanya sebuah kerangka penghubung untuk bisa menyelesaikan persoalan dari *forward kinematics* pada sebuah robot manipulator. Tugas akhir ini menggunakan metode Denavit-Hartenberg. (Lynch & Park, 2017)

### 2.3.2.1 Matriks Rotasi

Matriks rotasi adalah suatu matriks yang menggambarkan orientasi dan posisi *end-effector* terhadap kerangka acuan. Untuk memahami konsep ini, kita perlu memahami proyeksi terlebih dahulu. Misalnya, jika terdapat vektor  $a$  pada bidang  $xy$  dengan sudut  $45^\circ$  dan panjang 1 satuan, proyeksi vektor tersebut pada sumbu  $x$  adalah  $1\cos(45^\circ)$ , sementara proyeksi vektor  $a$  pada sumbu  $y$  adalah  $1\sin(45^\circ)$ . Ide proyeksi ini kemudian diperluas ke ruang tiga dimensi, di mana dalam konteks robotika, proyeksi melibatkan kerangka  $x$ ,  $y$ , dan  $z$  yang berbeda (Sodemann, 2017).

Sebagai contoh, sebuah kerangka  $x_1, y_1, z_1$  akan diproyeksikan ke dalam kerangka  $x_0, y_0, z_0$  seperti pada gambar 2.12. Untuk menentukan penulisan proyeksinya akan lebih mudah jika proyeksinya dituliskan dalam bentuk matriks. Matriks proyeksi inilah yang disebut dengan matriks rotasi yang akan membantu dalam menunjukkan orientasi dari kerangka *end-effector* sebuah robot manipulator terhadap kerangka acuan atau landasan. (Sodemann, 2017)



Gambar 2. 12 Kerangka  $x_1, y_1, z_1$  dan Kerangka  $x_0, y_0, z_0$

Sumber: Sodemann, (2017, Juny 11). *Robotics 1 U1 (Kinematics) S3 (Rotation Matrices) P1(Rotation Matrices)*. [Video]. Youtube.

<https://www.youtube.com/watch?v=IVjFhNv2N8o>

Penulisan matriks orientasi kerangka-kerangka  $x_1, y_1, z_1$  terhadap Kerangka  $x_0, y_0, z_0$  adalah sebagai berikut :

$$R_1^0 = \begin{matrix} x_0 & x_1 & y_1 & z_1 \\ y_0 & \begin{bmatrix} \cos\theta & -\sin\theta & 0 \\ \sin\theta & \cos\theta & 0 \\ 0 & 0 & 1 \end{bmatrix} \\ z_0 \end{matrix} \quad (2.1)$$

Dari rumus 2.1 diatas dapat dilihat bahwa penulisan  $R_1^0$  memiliki arti proyeksi kerangka 1 terhadap kerangka 0. Selanjutnya untuk masing-masing baris dan kolom memiliki cara membaca yang sama. Sebagai contoh, untuk kolom pertama baris pertama memiliki komponen yang akan menggambarkan proyeksi kerangka  $x_1$  terhadap  $x_0$ , Proyeksi kerangka  $x_1$  terhadap  $X_0$  adalah sebesar  $X_1 \cos\theta$ . Perlu diingat bahwa untuk proyeksi kerangka memiliki asumsi bahwapanjang masing-masing vector sumbu adalah 1 satuan. Matriks dari persamaan 2.1 disamping menunjukkan matriks rotasi kerangka 1 terhadap kerangka 0 juga memiliki arti matriks rotasi terhadap sumbu z (kerangka 1 diputar searah dengan sumbu z pada kerangka 0). Matriks rotasi terhadap sumbu y dan matriks rotasi terhadap sumbu x dapat dilihat pada persamaan 2.2 dan persamaan 2.3. (Sodemann, 2017).

$$R_X = \begin{bmatrix} 1 & 0 & 0 \\ 0 & \cos\theta & -\sin\theta \\ 0 & \sin\theta & \cos\theta \end{bmatrix} \quad (2.2)$$

$$R_Y = \begin{bmatrix} \cos\theta & 0 & \sin\theta \\ 0 & 1 & 0 \\ -\sin\theta & 0 & \cos\theta \end{bmatrix} \quad (2.3)$$

Untuk dapat menyelesaikan sebuah matriks rotasi dari sebuah robot manipulator yang dapat menunjukkan orientasi dari kerangka *end-effector* terhadap kerangka landasan maka kita perlu untuk mengetahui matriks orientasi kerangka n terhadap kerangka n-1 untuk semua kerangka yang ada. Selanjutnya maka kita perlu untuk mengahlikan semua matriks kerangka yang ada sehingga didapatkanlah matriks rotasi kerangka *end-effector* terhadap kerangka acuan dan landasan. Sebagai contoh kasus pada gambar 2.11, maka progress pengerjaanya dimulai dari mencari semua matriks orientasi kerangka n terhadap kerangka n-1. (Sodemann, 2017)

$$R_1^0 = \begin{bmatrix} \cos\theta_1 & \sin\theta_1 & 0 & 0 & 0 & 1 \\ \sin\theta_1 & \cos\theta_1 & 0 & 0 & 0 & 0 \\ 0 & 0 & 1 & 0 & -1 & 0 \end{bmatrix} \quad (2.4)$$

$$R_2^1 = \begin{bmatrix} 1 & 0 & 0 & 1 & 0 & 0 \\ 0 & 1 & 0 & 0 & 1 & 0 \\ 0 & 0 & 1 & 0 & 0 & 1 \end{bmatrix} \quad (2.5)$$

Matriks dari persamaan 2.4 menunjukkan matriks rotasi kerangka 1 terhadap kerangka 0. Dalam hal ini, kerangka 1 diputar terhadap sumbu z dari kerangka 0, untuk itu maka perlu untuk menuliskan matriks rotasi terhadap sumbu z. Selanjutnya untuk dapat menggambarkan secara tepat matriks rotasi kerangka 1 terhadap kerangka 0 maka matriks rotasi terhadap sumbu z perlu dikalikan dengan matriks yang menggambarkan proyeksi matriks 1 terhadap matriks 0 pada saat belum diputar atau pada  $0^\circ$ . (Sodemann, 2017)

Matriks dari persamaan 2.5 menunjukkan matriks rotasi kerangka 2 terhadap kerangka 1. Kerangka 2 hanya mengalami pergeseran searah dengan sumbu z dan tidak mengalami perubahan orientasi karena sendi yang ada merupakan sendi prismatic., maka dari itu matriks pertama adalah matriks identitas, hal ini dikarenakan matriks 2 dan matriks 1 memiliki orientasi yang sama. Langkah selanjutnya untuk mendapatkan matriks orientasi kerangka 2 terhadap kerangka 0 adalah dengan mengalihkan matriks orientasi kerangka 1 terhadap kerangka 0 dan matriks kerangka 2 terhadap kerangka 1. Hasil dari perkalian kedua matriks tersebut adalah:

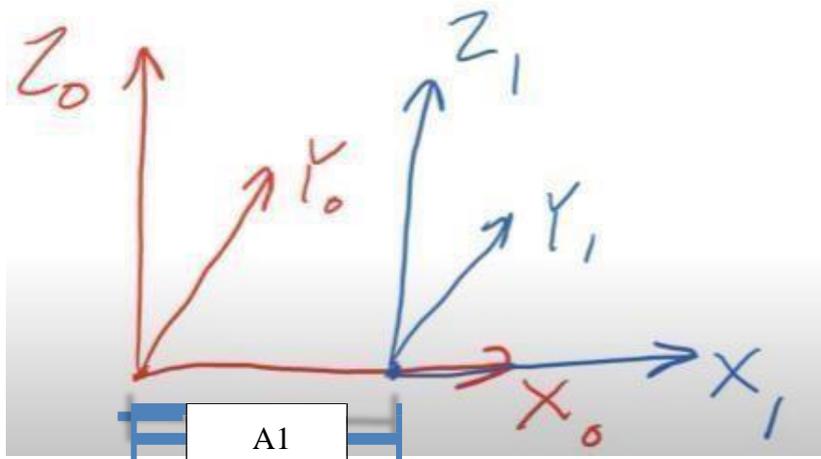
$$R_2^0 = \begin{bmatrix} \sin\theta_1 & 0 & \cos\theta_1 \\ -\cos\theta_1 & 0 & \sin\theta_1 \\ 0 & -1 & 0 \end{bmatrix} \quad (2.6)$$

Matriks dari persamaan 2.6 menunjukkan matriks orientasi kerangka 2 terhadap kerangka 0. Matriks dari persamaan ini menunjukkan bagaimana orientasi dari kerangka *end-effector* terhadap kerangka acuan atau landasan. Untuk menguji validasi dari persamaan ini maka bisa menggunakan nilai teta tertentu. (Sodemann, 2017)

### 2.3.2.2 Vektor Perpindahan

Agar dapat secara akurat menunjukkan posisi end-effector relatif terhadap referensi atau kerangka dasar, dua aspek krusial diperlukan. Pertama, orientasi end-effector terhadap referensi atau kerangka dasar, dan kedua, posisi end-effector terhadap referensi atau kerangka dasar. Proses untuk mendapatkan orientasi end-effector sudah dibahas pada poin 2.3.2.1, dan sekarang fokus kita adalah menjelaskan cara menentukan posisi *end-effector* terhadap referensi atau kerangka dasar. (Sodemann, 2017).

Untuk memudahkan perhitungan maka vektor perpindahan juga menggunakan matriks. Perbedaannya adalah matriks pada vektor perpindahan hanya berisi 3 baris dan 1 kolom. Baris pertama menunjukkan bagaimana perpindahan kerangka n terhadap kerangka n-1 searah dengan sumbu  $X_0$ . Baris kedua menunjukkan bagaimana perpindahan kerangka n terhadap kerangka n-1 searah dengan sumbu  $Y_0$ . Baris pertama menunjukkan bagaimana perpindahan kerangka n terhadap kerangka n-1 searah dengan sumbu  $Z_0$ . Sebagai contoh, dapat dilihat kerangka pada gambar 2.13 dan matriks vektor perpindahannya pada persamaan 2.8 (Sodemann, 2017)



Gambar 2. 13 Kerangka 0 dan Kerangka 1

$$d_1^0 = \begin{bmatrix} a_1 \\ 0 \\ 0 \end{bmatrix} \begin{matrix} x_1^0 \\ y_1^0 \\ z_1^0 \end{matrix} \quad (2.7)$$

Dari gambar 2.13 dapat dilihat bahwa kerangka 1 digeser sejauh  $a_1$  ke kanan searah dengan sumbu  $x_0$ . Dengan mengetahui perpindahan kerangka 1 terhadap kerangka 0 maka matriks vektor perpindahannya seperti yang tertulis pada persamaan 2.7. Untuk mendapatkan persamaan lengkap perpindahan dari kerangka *end-effector* terhadap kerangka acuan atau landasan memerlukan sebuah matriks bantuan yang disebut dengan *homogenous transformation matrix*. Pada bagian ini hanya bisa didapatkan bagaimana matriks perpindahan kerangka n terhadap kerangka n-1 untuk masing-masing kerangka. Sebagai contoh, matriks vektor perpindahan kerangka 1 terhadap kerangka 0 pada gambar 2.11 dan matriks vektor perpindahannya pada kerangka 2 terhadap kerangka 1 pada gambar 2.11 dituliskan pada persamaan 2.8 dan persamaan 2.9 (Sodemann, 2017)

$$d_1^0 = \begin{bmatrix} 0 \\ 0 \\ a_1 + a_2 \end{bmatrix} \quad (2.8)$$

$$d_2^1 = \begin{bmatrix} 0 \\ 0 \\ a_3 + d_2 \end{bmatrix} \quad (2.9)$$

### 2.3.2.3 Homogenous Transformation Matrix

Setelah memperoleh matriks rotasi dan matriks vektor perpindahan, langkah berikutnya adalah menggabungkan kedua matriks tersebut untuk mendapatkan matriks posisi dan orientasi dari kerangka *end-effector* relatif terhadap kerangka acuan atau landasan. Matriks gabungan ini dikenal sebagai *Homogenous Transformation Matrix*. Matriks ini memiliki dimensi 4 baris dan 4 kolom. Baris 1 hingga 3 diisi dengan matriks rotasi yang berdekatan dengan matriks vektor perpindahan, sementara baris terakhir berisi kolom dan matriks identitas 4x4. Pendekatan ini bertujuan untuk memungkinkan perkalian matriks yang konsisten dan sesuai aturan, tanpa mengubah makna yang terkandung di dalamnya. *Homogenous Transformation Matrix* disimbolkan dengan "H" (Sodemann, 2017).

Untuk mendapatkan *Homogenous Transformation Matrix* dari kerangka *end-effector* terhadap kerangka landasan atau acuan maka perlu untuk mengalikan setiap *Homogenous Transformation Matrix* dari masing-masing *Homogenous Transformation Matrix* matriks n terhadap *Homogenous Transformation Matrix* matriks n-1 yang ada untuk robot manipulator tersebut. Sebagai contoh dengan menggunakan model robot manipulator pada gambar 2.11, *Homogenous Transformation Matrix* kerangka 1 terhadap *Homogenous Transformation Matrix* kerangka 0 dan *Homogenous Transformation Matrix* kerangka 2 terhadap *Homogenous Transformation Matrix* kerangka 1 dapat dilihat pada persamaan 2.10 dan 2.11, sedangkan untuk *Homogenous Transformation Matrix* kerangka 2 terhadap *Homogenous Transformation Matrix* kerangka 0 dapat dilihat pada persamaan 2.12. (Sodemann, 2017)

$$H_1^0 = \begin{bmatrix} \sin\theta_1 & 0 & \cos\theta_1 & 0 \\ -\cos\theta_1 & 0 & \sin\theta_1 & 0 \\ 0 & -1 & 0 & a_1 + a_2 \\ 0 & 0 & 0 & 1 \end{bmatrix} \quad (2.10)$$

$$H_2^1 = \begin{bmatrix} 1 & 0 & 0 & 0 \\ 0 & 1 & 0 & 0 \\ 0 & 0 & 1 & a_3 + d_2 \\ 0 & 0 & 0 & 1 \end{bmatrix} \quad (2.11)$$

$$H_2^0 = \begin{bmatrix} \sin\theta_1 & 0 & \cos\theta_1 & \cos\theta_1(a_3 + d_2) \\ -\cos\theta_1 & 0 & \sin\theta_1 & \sin\theta_1(a_3 + d_2) \\ 0 & -1 & 0 & a_1 + a_2 \\ 0 & 0 & 0 & 1 \end{bmatrix} \quad (2.12)$$

Untuk menguji validasi dari persamaan 2.12 maka kita dapat mencoba memasukkan sebuah nilai untuk  $\theta$ , nilai untuk  $a_1$ , nilai untuk  $a_2$ , nilai untuk  $a_3$ , dan nilai untuk  $d_2$ .

#### 2.3.2.4 Homogenous Transformation Matrix Dengan Metode Denavit-Hartenberg

Metode alternatif untuk variasi *Homogeneous Transformation Matrix* adalah melalui pendekatan *Denavit-Hartenberg*. Tahap-tahap yang diperlukan untuk menyelesaikannya menggunakan metode ini melibatkan langkah awal, yaitu mendeskripsikan diagram kinematika sesuai dengan aturan *Denavit-Hartenberg* (D.H.). Tindakan selanjutnya adalah mengisi parameter pada tabel parameter D.H. Tabel parameter D.H. terdiri dari 5 kolom dengan jumlah baris sebanyak  $n-1$ , di mana  $n$  adalah jumlah kerangka yang ada dalam desain robot manipulator. Parameter yang diperlukan untuk diisikan disesuaikan dengan desain robot manipulator dan mencakup parameter  $\alpha$  dan  $\theta$  untuk merepresentasikan rotasi, serta parameter "r" dan "d" untuk merepresentasikan perpindahan linear (Sodemann, 2017).

Parameter  $\theta$  melambangkan seberapa besar sudut putar kerangka  $n-1$  terhadap sumbu  $z_{n-1}$  agar sumbu  $x_{n-1}$  memiliki orientasi yang sama dengan sumbu  $x_n$  ditambah dengan variabel perubahan sudut yang sudah ada. Parameter  $\alpha$  melambangkan seberapa besar sudut putar terhadap kerangka  $n-1$  terhadap sumbu  $x_n$  agar sumbu  $z_n$  memiliki orientasi yang sama terhadap sumbu  $z_{n-1}$ . Parameter "r" dilambangkan sebagai seberapa jauh perpindahan kerangka  $n-1$  pada kerangka  $n$  yang diukur hanya pada arah  $x_n$ . Parameter "d" dilambangkan sebagai seberapa jauh

perpindahan kerangka n-1 pada kerangka n yang diukur hanya pada arah  $z_{n-1}$ . Untuk penjelasan dapat dilihat pada tabel 2.1 yang merupakan contoh dari permodelan robot manipulator pada gambar 2.11.(Sodemann, 2017)

Tabel 2. 1  
Parameter D.H Untuk Model Robot Manipulator

n	$\theta$	$\alpha$	r	d
1	$\theta_1 - 90^\circ$	$-90^\circ$	0	$a_1+a_2$
2	0	0	0	$a_3+d_2$

Setelah mendapatkan tabel parameter seperti pada tabel 2.1 selanjutnya adalah membuat *Homogeneous Transformation Matrix*. Rumus *Homogeneous Transformation Matrix* dengan menggunakan metode *Denavit-Hartenberg* dapat dilihat pada persamaan 2.13

$$H_n^{n-1} = \begin{bmatrix} \cos\theta_n & -\sin\theta_n\cos\alpha_n & \sin\theta_n\sin\alpha_n & r_n\cos\theta_n \\ \sin\theta_n & \cos\theta_n\cos\alpha_n & -\cos\theta_n\sin\alpha_n & r_n\sin\theta_n \\ 0 & \sin\alpha_n & \cos\alpha_n & d_n \\ 0 & 0 & 0 & 1 \end{bmatrix} \quad (2.13)$$

Dari persamaan di atas maka didapatkan *Homogeneous Transformation Matrix* untuk model robot manipulator pada gambar 2.10 sebagai berikut :

$$H_1^0 = \begin{bmatrix} \cos(\theta_1 - 90^\circ) & -\sin(\theta_1 - 90^\circ)\cos(-90^\circ) & \sin(\theta_1 - 90^\circ)\sin(-90^\circ) & 0\cos\theta_n \\ \sin(\theta_1 - 90^\circ) & \cos(\theta_1 - 90^\circ)\cos(-90^\circ) & -\cos(\theta_1 - 90^\circ)\sin(-90^\circ) & 0\sin\theta_n \\ 0 & \sin(-90^\circ) & \cos(-90^\circ) & a_1 + a_2 \\ 0 & 0 & 0 & 1 \end{bmatrix} \quad (2.14)$$

$$H_2^1 = \begin{bmatrix} \cos 0 & -\sin 0\cos 0 & \sin 0\sin 0 & 0\cos 0 \\ \sin 0 & \cos 0\cos 0 & -\cos 0\sin 0 & 0\sin 0 \\ 0 & \sin 0 & \cos 0 & a_3 + d_2 \\ 0 & 0 & 0 & 1 \end{bmatrix} \quad (2.15)$$

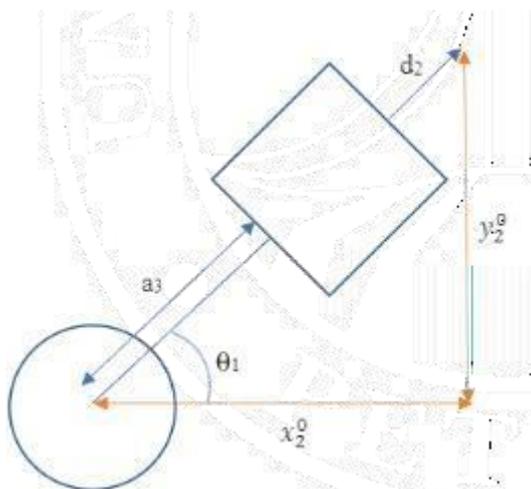
Matriks yang didapat dari persamaan 2.14 dan 2.15 jika disederhanakan maka akan didapatkan hasil yang sama seperti matriks pada persamaan 2.10 dan persamaan 2.11 langkah selanjutnya untuk mendapatkan orientasi dan posisi dari end-effector terhadap landasan dengan metode D.H adalah dengan mengalikan setiap matriks kerangka n terhadap kerangka n-1 (Sodemann, 2017).

### 2.3.3. Invers Kinematics

Salah satu metode untuk menentukan seberapa besar masing masing sendi digerakkan untuk bisa mendapatkan orientasi dan posisi dari *end-effector* adalah dengan metode *Inverse kinematics*. *Inverse kinematics* memiliki permasalahan yaitu penyelesaian yang jauh lebih rumit dibandingkan dengan *forward kinematics*. Seringkali permasalahan *Invers kinematics* memerlukan kemampuan numerik untuk bisa menyelesaikannya. Disamping itu, untuk derajat kebebasan yang lebih tinggi solusi yang bisa didapatkan tidak terbatas pada satu solusi. (Sodemann, 2017)

Metode pertama yang dilakukan untuk menyelesaikan persoalan *Invers Kinematics* adalah dengan metode grafis. Metode ini pada prinsipnya adalah menggunakan konsep-konsep yang terdapat pada trigonometri untuk dapat menyelesaikan persamaan *Invers Kinematics*. Untuk memudahkan perhitungan maka biasanya dibutuhkan bantuan berupa gambar tampak depan dan tampak atas dari desain robot manipulator. Beberapa persamaan yang perlu di ingat kembali adalah persamaan *Pythagoras*, persamaan *sohcahtoa* (hubungan *Sinus cosinus* dan tangen pada sebuah Segitiga Siku Siku), dan hukum *cosinus* pada sebuah Segitiga (Sodemann, 2017)

Sebagai contoh, pada gambar 2.11 maka input yang diberikan kepada sistem berupa posisi  $x_2^0$  dan posisi  $y_2^0$ . Dari kedua posisi ini maka dengan melakukan *Invers Kinematics* akan didapatkan nilai dari  $\theta_1$  dan nilai dari  $d_2$ . Gambar tampak atas dari gambar 2.11 dapat dilihat pada gambar 2.14 sebagai berikut :



Gambar 2. 14 Tampak Atas Gambar 2.11

Dari gambar diatas maka bisa didapatkan persamaan 2.16 dan persamaan 2.17 untuk menentukan nilai dari  $d_2$  dan nilai dari  $\theta_1$ , sebagai berikut:

$$d_2 = \sqrt{(x_2^0)^2 + (y_2^0)^2} - a_3 \quad (2.16)$$

$$\theta_1 = \tan^{-1} \left( \frac{y_2^0}{x_2^0} \right) \quad (2.17)$$

Dengan menemukan nilai dari  $d_2$  dan nilai dari  $\theta_1$ , maka penyelesaian *Inverse Kinematics* telah ditemukan. (Sodemann, 2017).

### 2.3.3.1. Invers Kinematics 6 DOF

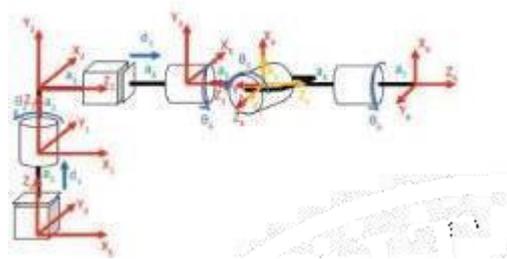
Dalam menangani permasalahan sistem robot manipulator yang memiliki lebih dari 3 derajat kebebasan, penyelesaian *Invers Kinematics* dengan metode yang telah dijelaskan pada sub-bab sebelumnya akan menjadi sangat sulit. Oleh karena itu, diperlukan modifikasi terhadap metode yang telah diperoleh dari sub-bab sebelumnya untuk mengatasi kendala tersebut.

Selain itu, untuk menyederhanakan perhitungannya, diperlukan penerapan asumsi bahwa tiga sendi pertama sepenuhnya menentukan posisi dari *end-effector*, sementara sendi tambahan hanya mempengaruhi orientasi dari *end-effector*. Sendi tambahan yang memengaruhi orientasi ini sering kali merupakan bagian dari model robot manipulator tipe *spherical*. (Sodemann, 2018)

Langkah-langkah dalam menyelesaikan *Invers Kinematics* untuk 6 derajat kebebasan adalah sebagai berikut :

1. Menggambarkan diagram kinematika dari 3 sendi pertama dan lakukan *Invers Kinematics* untuk posisi
2. Lakukan *Forward Kinematics* pada 3 sendi pertama untuk mendapatkan matriks rotasi  $R_3^0$ .
3. Temukan invers dari matriks rotasi  $R_3^0$
4. Lakukan *Forward Kinematics* pada 3 sendi lainnya dan dapatkan matriks rotasi  $R_3^0$ .
5. Temukan asumsi untuk matriks rotasi  $R_3^0$

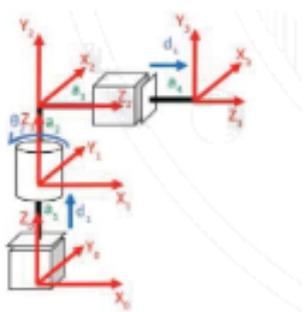
6. Dengan mengasumsikan posisi dari *end-effector* yang telah ditentukan maka selesaikan persamaan variabel sendi dengan menggunakan persamaan *Invers Kinematics* yang telah didapatkan dari persamaan pertama.
7. Memasukkan variabel-variabel tersebut dan gunakan matriks rotasi yang telah ditemukan untuk menyelesaikan variabel pada 3 sendi sisanya.



Gambar 2. 15 Model Robot Manipulator 6 DoF dan Diagram Kinematikanya

Sumber : Sodemann, (2018, January 24). *Robotics 2 u1 (Kinematics) s5 (Inverse Kinematics) p2 (procedure and programming)*. [Video]. Youtube.  
<https://www.youtube.com/watch?v=ZM9GOENJcuo>

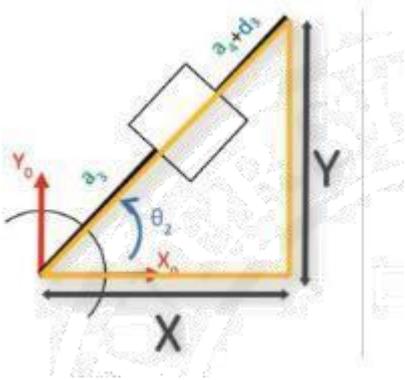
Sebuah rancangan robot manipulator terlihat seperti yang tergambar pada Gambar 2.15, di mana Diagram Kinematika disediakan dan sudah mematuhi aturan D.H. Langkah awal yang perlu diambil adalah menggambar secara terpisah Diagram Kinematika untuk tiga sendi pertama dari model robot manipulator 6 *DoF* tersebut. Tahap awal ini telah diselesaikan, seperti yang dapat dilihat pada Gambar 2.16 (Sodemann, 2018).



Gambar 2. 17 Diagram Kinematika 3 Sendi Pertama Robot Manipulator 6 DoF  
 Sumber : Sodemann., (2018, January 24) *Robotics 2 u1 (Kinematics) s5 (Inverse Kinematics) p2 (procedure and programming)*. [Video]. Youtube.  
<https://www.youtube.com/watch?v=ZM9GOENJcuo>

Harap dicatat pada Gambar 2.16, kerangka *end-effector* sejajar dengan sendi putar kedua pada model robot manipulator 6 *DoF*. Selanjutnya, kita akan melakukan *forward kinematics* sesuai dengan metode yang dijelaskan pada sub-bab sebelumnya. Oleh karena itu, disarankan untuk menambahkan anotasi di bagian atas Gambar 2.16

agar mempermudah perhitungan.



Gambar 2. 19 Tampak Atas Gambar 2.16

Sumber : Sodemann., ( 2018, January 24). *Robotics 2 u1 (Kinematics) s5 (Inverse Kinematics) p2 (procedure and programming)*. [Video]. Youtube.

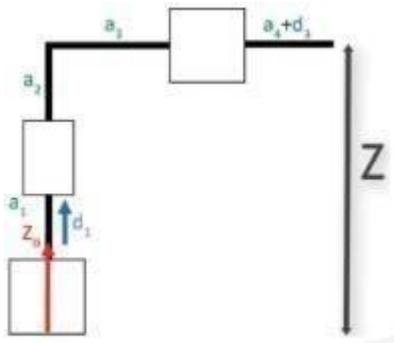
<https://www.youtube.com/watch?v=ZM9GOENJcuo>

Dari gambar 2.17 maka didapatkan persamaan 2.18 dan 2.19 dimana persamaan tersebut digunakan untuk menemukan nilai  $d_3$  dan persamaan untuk menemukan nilai  $\theta_1$  (Sodemann, 2018)

$$d_3 = \sqrt{(x)^2 + (y)^2} - a_3 - a_4 \quad (2.18)$$

$$\theta_1 = \tan^{-1} \left( \frac{y}{x} \right) \quad (2.19)$$

Selanjutnya untuk mendapatkan persamaan yang bisa menentukan nilai  $d_1$  maka dibutuhkan tampak samping dari gambar 2.17 yang dapat dilihat pada gambar 2.18 dibawah ini :



Gambar 2. 21 Tampak Samping dari Gambar

Sumber : : Sodemann., ( 2018, January 24). *Robotics 2 u1 (Kinematics) s5 (Inverse Kinematics) p2 (procedure and programming)*. [Video]. Youtube.

<https://www.youtube.com/watch?v=ZM9GOENJcuo>

Dari gambar diatas maka persamaan agar dapat rumus yang bisa menyatakan nilai dari variabel  $d_1$  adalah sebagai persamaan dibawah ini:

$$d_1 = z - a_2 - a_1 \quad (2.20)$$

Langkah kedua dari penyelesaian *Inverse Kinematics* pada robot manipulator 6 *DoF* adalah dengan melakukan *Forward Kinematics* untuk 3 sendi pertama dan menyelesaikan rotasi  $R_3^0$ . Matriks rotasi  $R_3^0$  ini dapat dilihat pada persamaan dibawah ini

$$R_3^0 = \begin{bmatrix} -\sin\theta_2 & 0 & \cos\theta_2 \\ -\cos\theta_2 & 0 & \sin\theta_2 \\ 0 & 1 & 0 \end{bmatrix} \quad (2.21)$$

Langkah ketiga dapat dilakukan dengan menggunakan metode numerik, sehingga langkah ini dapat ditinggalkan untuk sementara waktu. Langkah selanjutnya adalah untuk menemukan persamaan matriks  $R_6^3$ . Matriks rotasi  $R_6^3$  bisa didapatkan cara pada sub bab 2.3.2.1. Persamaan matriks rotasi  $R_6^3$  dapat dilihat pada persamaan 2.22 yang ada dibawah ini. Untuk mempersingkat persamaan maka *sin* e diinisialkan dengan huruf "s" dan untuk *cosin* e diinisialkan dengan huruf "c". (Sodemann, 2018)

$$R_6^3 = \begin{pmatrix} -s\theta_4 c\theta_5 c\theta_6 - c\theta_4 s\theta_6 & s\theta_4 c\theta_5 s\theta_6 - c\theta_4 c\theta_6 & -s\theta_4 s\theta_5 \\ c\theta_4 c\theta_5 c\theta_6 - s\theta_4 s\theta_6 & -c\theta_4 c\theta_5 s\theta_6 - s\theta_4 c\theta_6 & c\theta_4 s\theta_5 \\ -s\theta_5 c\theta_6 & s\theta_5 s\theta_6 + c\theta_5 & c\theta_5 \end{pmatrix} \quad (2.22)$$

Langkah kelima adalah dengan menentukan asumsi matriks rotasi  $R_6^0$ . Dalam menentukan asumsi matriks rotasi  $R_6^0$  perlu diingat bahwa matriks tersebut harus memenuhi syarat sebagai matriks rotasi. Syarat yang pertama adalah setiap baris dan kolom harus memiliki panjang vektor yang sama dengan 1. Maka dari itu akar dari jumlah kuadrat masing-masing suku pada baris yang sama harus sama dengan 1. Syarat kedua adalah matriks rotasi harus mengikuti aturan tangan kanan. Pada persoalan ini matriks rotasi  $R_6^0$  diasumsikan pada persamaaan yang ada dibawah ini. (Sodemann, 2018)

$$R_6^0 = \begin{bmatrix} -1 & 0 & 0 \\ 0 & -1 & 0 \\ 0 & 0 & 1 \end{bmatrix} \quad (2.23)$$

Dengan mengalikan matriks rotasi  $R_3^0$  dengan matriks rotasi  $R_6^3$  akan didapatkan matriks rotasi  $R_6^0$ . Untuk menemukan persamaan yang dapat menunjukkan nilai dari  $\theta_4$ ,  $\theta_5$ ,  $\theta_6$  diperlukan perpindahan ruas dari kiri ke kanan untuk

matriks rotasi  $R_3^0$ . Persamaan akhir matriks rotasi yang didapatkan adalah matriks rotasi  $R_6^3$  sama dengan invers matriks rotasi  $R_3^0$  dikalikan dengan matriks rotasi  $R_6^0$ . Untuk mendapatkan persamaan yang dapat menunjukkan nilai dari  $\theta_4, \theta_5, \theta_6$  adalah dengan menyamakan setiap suku dari matriks rotasi  $R_6^3$  dengan matriks yang didapatkan dari hasil perkalian *invers matriks* rotasi  $R_3^0$  dengan matriks  $R_6^0$  (Sodemann, 2018)

Hingga tahap ini, perlu diperhatikan bahwa penyelesaian yang didapatkan seringkali lebih dari 1 penyelesaian sehingga perlu melakukan pengecekan ulang dengan memasukkan nilai dari  $\theta_4, \theta_5, \theta_6$  yang didapatkan dari persamaan-persamaan yang diperoleh sebelumnya ke matriks rotasi  $R_6^3$ . Jika hasil telah diperoleh dengan tepat maka persamaan penyelesaian untuk  $\theta_4, \theta_5, \theta_6$  telah tepat dan bisa digunakan untuk seterusnya pada desain robot manipulator 6 *DoF* ini. Dengan ini maka proses untuk mencari masing-masing nilai untuk masing-masing sendi pada robot manipulator 6 *DoF* yang ada pada gambar 2.15 sudah didapatkan. (Sodemann, 2018)

## 2.4 Kontrol Robot Manipulator

Kontrol robot manipulator merujuk pada upaya untuk mengatur dan mengendalikan pergerakan dan perilaku robot manipulator, yang merupakan jenis robot yang digunakan untuk melakukan tugas-tugas manipulasi, seperti mengangkat, memindahkan, atau memanipulasi objek di lingkungannya. Kontrol ini melibatkan penggunaan algoritma dan strategi kontrol untuk memastikan bahwa robot dapat menjalankan tugas-tugasnya dengan akurat, efisien, dan aman.

Berikut adalah beberapa konsep umum yang terlibat dalam kontrol robot manipulator:

- **Posisi dan Orientasi *End-Effector*:** Kontrol ini mencakup penentuan posisi dan orientasi akhir dari alat penggerak robot, yang sering disebut sebagai *end-effector*. Tujuannya adalah untuk memastikan bahwa *end-effector* berada pada posisi yang diinginkan untuk menyelesaikan tugas tertentu.
- **Perencanaan Jalur (*Path Planning*):** Ini melibatkan perencanaan jalur optimal yang harus diikuti oleh robot untuk mencapai tujuan tertentu. *Path planning* melibatkan menghindari rintangan dan memilih jalur yang efisien dalam ruang tugas.
- **Kontrol Kecepatan dan Percepatan:** Kontrol ini berfokus pada mengatur kecepatan dan percepatan sendi robot. Ini penting untuk menghindari gerakan yang terlalu cepat atau lambat, yang dapat mempengaruhi akurasi dan keamanan.
- **Kontrol Torsi atau Gaya Hubungan Dinamika:** Kontrol ini mempertimbangkan dinamika robot untuk menghasilkan gaya atau torsi yang diperlukan pada setiap sendi agar dapat menjalankan tugas dengan efisien. Ini termasuk perhitungan berat, gesekan, dan inersia dalam sistem.
- **Kontrol Visual:** Kontrol visual melibatkan penggunaan sensor dan pengolahan citra untuk memantau lingkungan sekitar dan memperbaiki posisi atau jalur robot berdasarkan informasi visual.
- **Kontrol Keamanan :** Kontrol ini melibatkan implementasi fitur keamanan untuk mencegah tabrakan, mendeteksi gangguan, dan menghentikan operasi robot jika diperlukan untuk menjaga keamanan.
- **Sistem Kontrol Terdistribusi atau Terpusat:** Robot manipulator dapat

dikendalikan dengan sistem kontrol yang terdistribusi di setiap sendi atau dengan sistem kontrol terpusat yang mengkoordinasikan seluruh robot.

Penerapan kontrol robot manipulator melibatkan penggunaan berbagai teknik dan algoritma, termasuk kontrol PID (Proporsional, Integral, Derivatif), kontrol adaptif, dan kontrol berbasis visi. Pengembangan teknologi kontrol ini terus berkembang untuk meningkatkan kinerja, akurasi, dan fleksibilitas robot manipulator dalam berbagai aplikasi industri dan penelitian.

## 2.5 Komponen Elektronika Robot 6 Derajat Kebebasan

Secara kontrol yang telah dibuat dalam software perlu untuk diteruskan menuju ke aktuator, dalam hal ini adalah motor *stepper*. Disamping itu, perlu perangkat-perangkat yang bisa mendeteksi kondisi robot tersebut dan perangkat-perangkat untuk memberikan daya ke dalam robot. Untuk itu, sebuah sistem pada robot memerlukan komponen-komponen elektronika.

### 2.5.1 *Teensy4.1*

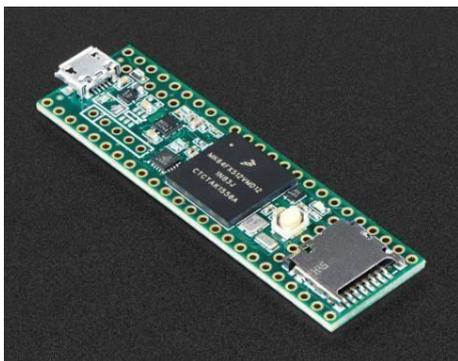
*Teensy 4.1* adalah salah satu dari serangkaian papan pengembangan mikrokontroler yang dikembangkan oleh PJRC (*Paul Stoffregen*). *Teensy* dikenal sebagai platform mikrokontroler yang kuat, kecil, dan dapat diakses dengan harga yang terjangkau. *Teensy 4.1* adalah salah satu varian dari keluarga *Teensy* yang memiliki beberapa fitur khusus.

Berikut beberapa fitur utama *Teensy 4.1*

- Mikrokontroler:  
*Teensy 4.1* dilengkapi dengan mikrokontroler *NXP/Freescale MK64FX512VMD12*, yang merupakan bagian dari keluarga mikrokontroler *ARM Cortex-M4 32-bit*. Kecepatan *clock* dapat mencapai *120 MHz*.
- Memori:  
*Teensy 4.1* memiliki memori *Flash* sebesar *512 KB*, *RAM* sebesar *192 KB*, dan *EEPROM* sebesar *2 KB*.
- *Input/Output (I/O)*:  
*Teensy 4.1* dilengkapi dengan sejumlah pin *I/O digital* dan *analog*. Pin *analog* dapat digunakan sebagai pin *PWM*, dan beberapa di antaranya mendukung fungsi khusus seperti *I2C*, *SPI*, *UART*, dan sebagainya.

- USB:  
*Teensy 4.1* memiliki dukungan USB penuh, termasuk kemampuan untuk diatur sebagai perangkat *HID (Human Interface Device)*, *MIDI*, Audio, dan sebagainya. Ini membuatnya cocok untuk proyek-proyek yang memerlukan komunikasi USB.
- Kecepatan:  
*Teensy 4.1* dirancang untuk memberikan kinerja tinggi dengan prosesor *ARM Cortex-M4* dan kecepatan *clock* yang tinggi.
- Sensor:  
*Teensy 4.1* mendukung sensor akselerometer dan giroskop *built-in*, yang dapat berguna untuk aplikasi yang melibatkan deteksi gerakan atau orientasi.
- Pengembangan:  
*Teensy* dapat diprogram menggunakan *Arduino IDE* dengan *Teensyduino add-on*. *Teensyduino* menyediakan dukungan penuh untuk fungsi-fungsi *Teensy* dan mempermudah pengembangan perangkat lunak.
- Ukuran:  
Meskipun memiliki banyak fitur, *Teensy 4.1* tetap relatif kecil, memudahkannya untuk diintegrasikan ke dalam proyek-proyek yang memerlukan ukuran yang kompak.

*Teensy 4.1* pada robot manipulator seral kamera 6 *DoF* ini digunakan untuk komunikasi antara *software* dengan *driver* dari motor *stepper*, *encoder*, dan *limit switch*. Disamping itu, *Teensy 4.1* ini digunakan juga sebagai daya bertegangan 5 volt DC untuk *encoder*. (Makers.id)

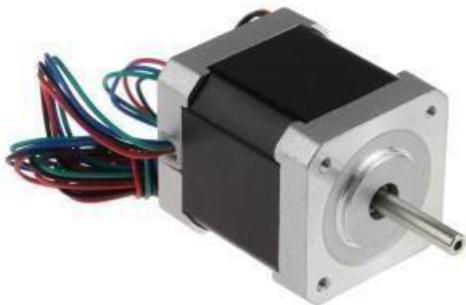


Gambar 2. 22 Teensy 4.1

Sumber : Teensy 4.1 DEV-16771-Sparkfun. (2020, May 19). Mouser Electronic  
<https://www.mouser.co.id/new/sparkfun/sparkfun-teensy-4-1-dev-16771/>

### 2.5.2 Motor Stepper

LMotor *stepper* adalah jenis motor listrik yang berputar dalam serangkaian langkah sudut kecil, bukan secara kontinu seperti motor listrik lainnya. Motor ini merupakan jenis aktuator *digital* yang mengubah energi elektromagnetik menjadi energi mekanik untuk melakukan pekerjaan mekanik. Sebuah motor *stepper* membagi satu rotasi penuh menjadi sejumlah langkah yang sama. Posisi motor dapat diperintahkan untuk bergerak dan bertahan pada salah satu langkah tersebut tanpa sensor posisi untuk umpan balik, asalkan motor tersebut memiliki ukuran yang sesuai untuk aplikasi yang berkaitan dengan torsi dan kecepatan. Motor *stepper* memiliki beberapa jenis, antara lain berdasarkan struktur rotor-stator dan lilitan stator. Jenis motor *stepper* juga dapat diklasifikasikan sebagai unipolar dan bipolar berdasarkan jenis belitan stator. Motor *stepper* memiliki kelebihan, seperti posisi yang tepat, respons yang baik terhadap mulai/berhenti/mundur, keandalan tinggi karena tidak memiliki sikat kontak, serta kontrol *loop* terbuka yang membuatnya lebih sederhana dan mudah untuk dikendalikan. Motor *stepper* banyak digunakan dalam berbagai aplikasi, seperti printer 3D, mesin *CNC engraving*, lengan robot, dan sebagainya (Wikipedia)



Gambar 2. 23 Motor Stepper

Sumber : Prasetyo.E.A. (2022, August 30). Penjelasan tentang Motor Stepper.

<https://www.arduinoindonesia.id/2022/08/penjelasan-tentang-motor-stepper.html>

### 2.5.3 Power Supply 250W 24V 30A 115/230V

250W 24V 30A 115/230V *Switching Power Supply* adalah sebuah perangkat yang dapat digunakan dalam berbagai aplikasi seperti otomasi industri dan sistem *CNC stepper/servo*. Perangkat ini memiliki *output* 24V DC 30A dan dapat menerima input AC 115V atau 230V yang dapat dipilih melalui *switch*. Perangkat ini juga dilengkapi dengan pendingin udara yang dipaksa oleh kipas DC bawaan. Perangkat ini dapat digunakan untuk menggerakkan motor *stepper* atau *servo* pada sistem *CNC* atau 3D printer.

Dalam desain robot manipulator serial kamera 6 *DoF* ini, tipe *power supply* yang digunakan adalah 250W 24V 30A 115/230V *Power Supply*, dimana *power supply* ini memiliki spesifikasi sebagai berikut : (Stepperonline, 2020)

- Keluaran tegangan sebesar 24V DC
- Keluaran arus sebesar 30A
- Berat 1.1 kg
- Efisiensi 80%



Gambar 2. 24 Power Supply 250W 24V 30A 115/230V

## 2.6 Penelitian Terdahulu

### 2.6.1 Desain dan Pemodelan Mekanisme Sambungan Kekakuan Variabel *Hibrid Seri Paralel Berpenggerak Kabel untuk Robotika*

Jurnal ini menjelaskan tentang Merancang lengan robotik yang signifikan berbeda dengan merancang manipulator robot untuk beroperasi dengan aman dan nyaman dan konsep desain nya adalah dengan mekanisme yang diusulkan meliputi pelat bawah dan pelat atas yang digunakan untuk menahan pegas kompresi pada posisi konsentris. Struktur digerakkan oleh tiga kabel yang ditarik atau dilepaskan oleh tiga motor servo yang terletak di bawah mekanisme CDPS. Desain yang dikonsepskan adalah dengan menyambungkan kabel atau *slink* pada *upper plate* dan poros yang tersambung pada motor yang disebutkan dengan *capstan* letak *capstan* ini tersambung pada motor servo. (Yigit & P.Boyraz., 2017)

### **2.6.2 Perancangan Lengan Robot Lima *Axis* Untuk Material *Handling* Ringan**

Skripsi ini mencakup rancangan robot lima sumbu yang ditujukan untuk penanganan material. Pada dasarnya, perancangan robot lima sumbu ini dirancang untuk mengangkat benda dengan bobot ringan di lingkungan kerja yang dianggap berbahaya bagi manusia. Kelemahan desain robot dalam skripsi ini terletak pada kelancaran gerak yang masih terhambat, sehingga diperlukan mekanisme yang lebih baik agar robot yang dihasilkan dapat bergerak dengan lebih lancar. (Prayoga, 2005)