

## BAB II LANDASAN TEORI

### 2.1 Tinjauan Pustaka

Dalam penelitian ini peneliti menggunakan tinjauan pustaka pada penelitian sebelumnya yang dapat digunakan sebagai pendukung dalam penelitian, berikut merupakan tinjauan pustaka dalam penelitian ini:

**Tabel 2. 1** Tinjauan Pustaka

Nomor Tinjauan Pustaka	Penulis	Tahun	Judul
Tinjauan Pustaka 1	Ken Masuya dan Tomomichi Sugihara	2015	Dead Reckoning for Biped Robots that Suffers Less from Foot Contact Condition Based on Anchoring Pivot Estimation
Tinjauan Pustaka 2	Michal Reinstein dan Matej Hoffman	2011	Dead Reckoning in a Dynamic Quadrupped: Inertial Navigation System Aided by a Legged Odometer

**Tabel 2.1** Tinjauan Pustaka (Lanjutan)

Nomor Tinjauan Pustaka	Penulis	Tahun	Judul
Tinjauan Pustaka 3	Huang Li, Jing Xiong, Yangzhou Gan, Zeyang Xia	2015	A Humanoid Robot Localization Method for Biped Navigation in Human-Living Environments
Tinjauan Pustaka 4	Andi Adriansyah	2014	Perancangan Robot Localization Menggunakan Metode Dead Reckoning
Tinjauan Pustaka 5	Diah Liani, Ade Silvia, dan Lindawati	2017	Sistem Navigasi Pada Mobile Robot Dalam Penentuan Arah dan Pemetaan Posisi

### 2.1.1 Tinjauan Pustaka 1

Pada penelitian ini membahas tentang teknik perhitungan *dead reckoning* untuk umpan balik pada robot berkaki dua. Estimasi posisi pada robot diperoleh dengan menggabungkan informasi hanya dari sensor *internal* termasuk penyandi sudut sendi, sensor inersia, dan sensor gaya. Hal ini menggabungkan perhitungan kinematika dan integral ganda dengan cara yang saling melengkapi untuk meningkatkan akurasi. Perhitungan kinematika mempertimbangkan pergerakan kaki penyangga, khususnya, rotasi di sekitar titik tetap dan berguling di medan. Bobot pada setiap informasi disesuaikan secara otomatis berdasarkan gaya reaksi dari tanah. Melalui simulasi ditunjukkan bahwa metode ini dapat bekerja dengan baik sekalipun pada situasi kaki pendukung bergerak sehubungan dengan tanah dan bahkan tidak memiliki kontak dengan tanah (Masuya et al., 2015).

### 2.1.2 Tinjauan Pustaka 2

Pada penelitian ini membahas algoritma navigasi dalam mengestimasi postur dan jarak yang ditempuh oleh robot berkaki empat dengan menggunakan kombinasi algoritma mekanisasi Sistem Navigasi Inersia (INS) dan odometri berkaki. *Extended Kalman Filter* (EKF) digunakan untuk estimasi kesalahan dan penggabungan data dari dua sumber informasi yang independen ini. INS memproses data inersia mentah untuk memberikan informasi tentang posisi, kecepatan, dan sikap robot. Disisi lain, odometri berkaki menggunakan kombinasi sinyal sensor sendi dan sensor tekanan untuk memperkirakan kecepatan robot. Sistem navigasi ini memastikan pelacakan yang tepat dari postur robot (*roll* dan *pitch*) dan

pelacakan yang memuaskan dari posisinya dalam *interval* waktu menengah (Reinstein et al., 2011).

### **2.1.3 Tinjauan Pustaka 3**

Pada penelitian ini membahas tentang perencanaan langkah kaki secara *real-time* untuk robot *humanoid*, diperlukan agar mendapatkan lokasi yang akurat dari penempatan kakinya. Dalam penelitian ini menyajikan metode lokalisasi untuk penempatan kaki pada robot *humanoid* yang bernavigasi pada lingkungan manusia. Sensor *Kinect* digunakan pertama kali untuk mendapatkan peta digital dari lingkungan *global* dan posisi serta orientasi kepala robot diekstraksi. Metode yang diusulkan digunakan dalam perencanaan langkah kaki secara *real-time* di lingkungan yang dinamis. Metode yang disajikan telah diuji pada platform robot dalam sebuah adegan eksperimental. Hasil pengujian memverifikasi bahwa metode yang disajikan layak untuk lokalisasi langkah kaki dan perencanaan langkah kaki pada robot berkaki dua (Li et al., 2015).

### **2.1.4 Tinjauan Pustaka 4**

Pada penelitian ini membahas perancangan *mobile localization* menggunakan metode *dead reckoning*. Dalam merancang robot bergerak terdapat sebuah permasalahan, yaitu dimana robot dapat mengetahui posisinya dari waktu ke waktu. Permasalahan ini biasa disebut sebagai sistem lokalisasi robot. *Dead reckoning* adalah teknik lokalisasi robot bergerak yang memanfaatkan hasil dari data sensor bergerak untuk memperkirakan perubahan posisi robot dari waktu ke waktu sepanjang robot bergerak. Berdasarkan hasil pengujian, dapat dikatakan bahwa robot hasil

rancangan dapat bergerak dengan pergerakan yang sesuai dengan program yang dikehendaki, baik pergerakan lurus maupun pergerakan berbelok. Namun, terjadi penyimpangan posisi baik pada sumbu X maupun pada sumbu Y. Hal ini dikarenakan slip pada masing-masing roda yang menyebabkan akumulasi penyimpangan posisi secara keseluruhan (Adriansyah, 2014).

### 2.1.5 Tinjauan Pustaka 5

Pada penelitian ini dijelaskan bahwa *mobile* robot menggunakan sebuah *compass* sebagai penentu arah sehingga robot memiliki kemampuan untuk mengatur arah gerak untuk dapat menuju arah yang dituju. Untuk melakukan gerak kearah yang dituju yaitu menggunakan metode *waypoints*. Hasil dari penelitian yang telah dilakukan peneliti yaitu modul *compass* memiliki selisih rata-rata sebesar  $4^\circ$  setiap *interval* 5 detik antara robot 1, 2, dan 3, dan selisih sebesar  $3^\circ$  pada sensor *compass* untuk dapat mencapai gerak rotasi yang ditargetkan. Pembacaan nilai *compass* juga dipengaruhi dengan benda yang ada di sekitarnya sehingga pergerakan robot tidak sesuai ketika dijalankan di dalam area *indoor* dan akan lebih baik jika dilakukan di area *outdoor* (Liani et al., 2017)

## 2.2 Dead Reckoning

*Dead Reckoning* adalah salah satu metode lokalisasi yang termasuk dalam kategori *relative* atau *local localization*, yang digunakan untuk mengestimasi posisi saat ini dan perubahan posisi dengan menghitung jarak yang telah ditempuh dari titik sebelumnya yang telah diketahui. Metode ini diterapkan secara bertahap untuk memperbarui posisi serta orientasi pada robot sepanjang robot

bergerak (Setiawan et al., 2012) . Hal ini mengacu pada metode penentuan posisi *relative* yang sederhana, mudah diterapkan dan dapat memberikan informasi posisi secara nyata.

Algoritma *dead reckoning* adalah metode navigasi yang dapat memperkirakan posisi tanpa menggunakan sinyal GPS atau sinyal sedang tidak tersedia, biasanya sinyal tidak tersedia ketika berada di dalam ruangan sebuah gedung, di lautan, atau pada saat sedang berada di udara. Metode ini memanfaatkan gabungan data dari dua sensor yaitu *accelerometer* dan sensor *compass*, yang kemudian dilakukan perhitungan sehingga dapat menghasilkan perkiraan posisi baru berdasarkan posisi awal yang telah diketahui. Metode *dead reckoning* sudah banyak diimplementasikan diantaranya adalah untuk navigasi pesawat terbang pada saat mengudara, navigasi pada kapal laut yang sedang menjelajahi lautan, dan navigasi bagi pejalan kaki atau lebih dikenal sebagai pedestarian *dead reckoning*. Algoritma pedestarian *dead reckoning* adalah algoritma yang memanfaatkan kinematika gaya berjalan pada manusia untuk dapat memberikan perkiraan posisi pada pejalan kaki dengan menghitung jarak yang telah ditempuh terhadap posisi awal yang telah diketahui (Ramadhana et al., 2013).

### **2.3 Robot Humanoid**

Robot *humanoid* adalah jenis robot *autonomus* yang memiliki bentuk fisik secara keseluruhan menyerupai manusia dan dirancang memiliki fungsi dasar yang sama dengan milik manusia. Secara umum robot *humanoid* memiliki struktur tubuh manusia dilengkapi dengan sebuah kepala, dua buah lengan dan dua buah kaki. Meskipun demikian tidak semua jenis robot *humanoid* memiliki bentuk seperti layaknya manusia tetapi juga ada beberapa robot *humanoid* yang memiliki bentuk

hanya sebagian tubuh manusia dimulai dari pinggang hingga ke atas atau hanya berbentuk lengan saja. Selain itu robot *humanoid* juga ada yang dilengkapi dengan wajah beserta mata dan mulut layaknya manusia (Erdinal et al., 2021).

Robot *humanoid* dirancang untuk dapat melakukan beberapa tugas fisik dan mental yang sama dengan manusia dalam menjalankan kehidupan sehari-harinya. Tujuan para ilmuwan dalam menciptakan dan mengembangkan robot *humanoid* adalah agar robot dapat memahami kecerdasan akal milik manusia serta dapat bertindak dan mengambil keputusan layaknya manusia sehingga robot *humanoid* memiliki potensi untuk melakukan berbagai tugas di lingkungan manusia. Robot *humanoid* memiliki kemampuan bergerak secara autonomus, dapat mengumpulkan informasi pada lingkungan manusia dengan memanfaatkan sensor-sensor seperti (*Compass* dan GPS) dan berinteraksi dengan informasi-informasi yang diperoleh dari sensor tersebut sehingga robot tidak hanya diam ditempat seperti robot manipulator yang bekerja di tempat terstruktur sehingga robot *humanoid* harus memiliki perencanaan jalur dan penghindaran rintangan untuk dapat bergerak di dalam lingkungan kompleks.

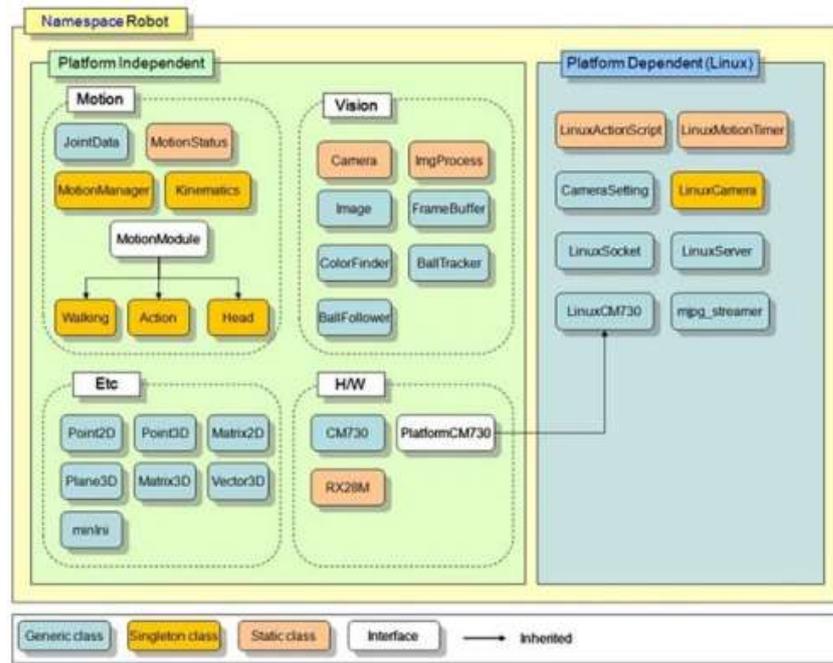
#### **2.4 Platform Robot**

Robot sepak bola *humanoid* milik tim Krakatau FC dibangun dengan mereferensi pada platform robot Darwin-OP. Robot Krakatau di desain dengan rangka *alloy* dan memiliki 20 DOF (*Degree of Freedom*). Pada setiap DOF digerakkan menggunakan sebuah aktuator berupa servo motor. Servo sendiri diibaratkan sebagai sendi pada manusia. Robot Krakatau menggunakan 3 buah jenis servo yaitu servo MX-28T, MX-64T, dan MX-106. Pada bagian kepala terdapat 2 sendi dan setiap lengan terdapat 3 sendi, masing-masing digerakkan menggunakan servo MX-28T

sedangkan pada sendi bagian pinggang sampai dengan kaki digerakkan menggunakan gabungan dari servo MX-64T dan MX-106. Sistem kepala robot dirancang dalam 2 DOF, masing-masing untuk mendukung fungsi angguk (*Tilt*) dan fungsi geleng (*Pan*). Kedua fungsi tersebut digunakan untuk mengarahkan sistem sensor utama, yakni kamera digital PS Eye, untuk mengenali lingkungan sekitar yang dibutuhkan seperti melacak keberadaan bola dan gawang. Robot Krakatau diotaki oleh komputer papan tunggal intel NUC10i3FNH dengan *processor* intel core i3 menggunakan sistem operasi Ubuntu. *Platform* perangkat lunak pada robot Krakatau diadopsi dari *platform* Darwin yang merupakan sebuah *open platform* dan dibangun secara hirarkis dengan mempertimbangkan modularitas dan independensi.

## 2.5 Framework Krakatau

*Framework* Krakatau merupakan *framework* robot sepak bola *humanoid* yang dikembangkan oleh tim robot sepak bola Krakatau FC Universitas Teknokrat Indonesia. *Framework* Krakatau dibuat berdasarkan *framework* Darwin yang merupakan sebuah *open platform*. Robot sepak bola *humanoid* milik tim Krakatau FC secara keseluruhan mengadopsi *platform* Darwin-OP dimulai dari perancangan mekanika, gaya berjalan pada robot hingga perangkat lunak yang digunakan. *Framework* Darwin-OP terdiri dari modul komunikasi perangkat, modul gerak, modul jalan kaki, modul penginderaan, modul perilaku, modul penglihatan, dan modul diagnostik. Kerangka kerja ini telah dikembangkan dengan bahasa pemrograman C++ dimana kodenya beroperasi dengan sistem-independen (Ha et al., 2013). Berikut adalah kerangka kerja perangkat lunak pada *platform* Darwin-OP dapat dilihat pada gambar 2.1.



**Gambar 2.1** Kerangka Kerja Platform Darwin-OP

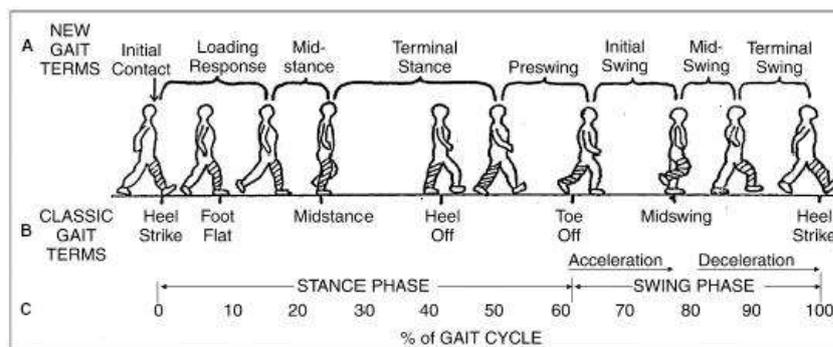
Sumber: (Ha et al., 2013)

*Framework* Krakatau sendiri dikembangkan pertama kali pada tahun 2013. Sejak dikembangkan, *framework* Krakatau telah mengalami berbagai perubahan dibandingkan baseline (*framework* Darwin). Setelah 2 tahun dalam masa pengembangan akhirnya pada tahun 2015 *framework* Krakatau versi 1.0 yang merupakan versi stabil dirilis. Pada versi 1.0 tim pengembang menambahkan fitur deteksi bola berdasarkan 2 parameter yaitu warna lapangan dan warna bola sehingga robot dapat mengeliminasi *noise* yang berada di luar lapangan. Satu tahun setelahnya pada tahun 2016 *framework* Krakatau dikembangkan kembali serta melakukan *update* pada bagian *trajectory*. Pada tahun yang sama dirilis kembali *framework* Krakatau versi 1.5 ini dimana pada versi ini tim pengembang menambahkan *balancing* pada *trajectory* robot sehingga robot dapat merespon ketidakseimbangan dengan cara menggerakkan *offset actuator* tubuh bagian atas serta menambahkan pola gerak dinamis pada robot. Pada tahun 2017 *framework*

Krakatau kembali merilis versi 2.0 dengan *update* yang cukup signifikan diantaranya deteksi bola dan gawang yang sudah menggunakan opencv serta komunikasi *multicast* dengan rekan satu tim.

## 2.6 Konsep Berjalan Pada Robot Humanoid

Robot *humanoid* memiliki mekanisme berjalan *bipedal*. Untuk menciptakan robot yang dapat berjalan dengan dua kaki terdapat berbagai kendala dalam mewujudkannya. Gerakan berjalan pada robot *bipedal* dimodelkan berdasarkan gerakan berjalan manusia yang digambarkan sebagai urutan gerak berulang. Gerakan ini disebut sebagai *gait cycle*. *Gait cycle* terdiri dari dua fase yaitu *stance phase* dan *swing phase*. Fase *stance* meliputi proses *initial contact*, *foot-flat* atau *loading response* yaitu permulaan kaki kontak dengan tanah, dan *toe-off* atau *pre-swing*. Sedangkan pada fase *swing* terdapat proses *acceleration*, *midswing*, dan *deceleration*. Durasi untuk menyelesaikan siklus *gait* dikenal sebagai *gait time*, yang terbagi dalam *stance time* dan *swing time* (Permatasari, 2016).

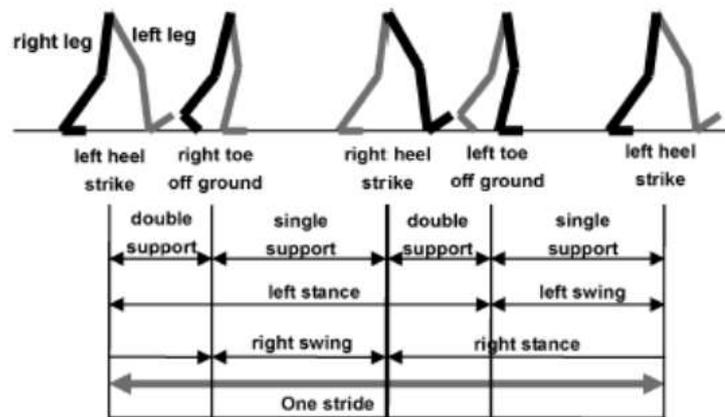


**Gambar 2.2** Gait Cycle Selama Satu Siklus.

Sumber: (Ariano et al., 2019)

Sedangkan pada robot *humanoid* gerakan berjalan robot direpresentasikan sebagai siklus berulang, atau biasa disebut juga dengan gaya berjalan atau *walking gait*. Gaya berjalan ini terdiri dari serangkaian Gerakan yang menyerupai cara berjalan

manusia. Gaya berjalan robot humanoid dapat dibagi menjadi dua fase saat berjalan yaitu fase SSP (*Single Support Phase*) dan DSP (*Double Support Phase*). SSP adalah kondisi pada saat robot berjalan hanya bertumpu pada salah satu telapak kaki sedangkan DSP adalah kondisi pada saat robot berjalan dengan bertumpu pada kedua kaki (Anragani, 2016). Ilustrasi gaya berjalan robot *humanoid* dapat dilihat pada gambar 2.4.



**Gambar 2.3** Ilustrasi Gaya Berjalan Robot

Sumber: (Yilmaz, 2010)

## 2.7 Transformasi Geometri

Transformasi geometri berasal dari kata transformasi yang berarti perubahan dan geometri yang berarti ilmu yang membahas tentang bangun. Transformasi geometri merupakan proses dalam penentuan titik koordinat baru pada sebuah bidang.

Transformasi geometri dibagi menjadi empat jenis yaitu transformasi pergeseran (translasi), pencerminan (refleksi), perputaran (rotasi), dan perbesaran (dilatasi). Pada penelitian ini, transformasi geometri digunakan untuk membuat gerakan robot bermanuver seperti berjalan lurus dan belok (Wibowo, 2017).

### 2.7.1 Translasi (Pergeseran)

Translasi atau pergeseran adalah sebuah proses yang dilakukan untuk menggeser sebuah bangun pada koordinat kartesian sebanyak nilai pergeseran sesuai dengan yang telah ditentukan dalam sumbu x, y, z. Pada transformasi translasi, ukuran dan bentuk serta arah bangun tidak berubah, yang berubah hanya posisinya terhadap posisi awal. Pada Gambar 2. 4 segitiga ABC berwarna hitam mempunyai koordinat A (3,9), B (3,3), C (6,3), apabila ditransformasikan translasi menjadi segitiga ABC warna oranye, hijau dan ungu perhitungannya adalah sebagai berikut :

1. Segitiga ABC oranye

Koordinat segitiga ABC oranye A<sub>2</sub>(-7,9), B<sub>2</sub>(-7,3), C<sub>2</sub>(- 4,3). Maka pergeseran translasi sebanyak x,y (-10,0).

2. Segitiga ABC ungu

Koordinat segitiga ABC ungu A<sub>3</sub>(3,-4), B<sub>3</sub>(3,-10), C<sub>3</sub>(6,-10). Maka pergeseran translasi sebanyak x,y (0,- 13).

3. Segitiga ABC hijau

Koordinat segitiga ABC hijau A<sub>4</sub>(3,-4), B<sub>4</sub>(3,-10), C<sub>4</sub>(6,-10). Maka pergeseran translasi sebanyak x,y (-10,- 13).

Dari perhitungan di atas maka transformasi pergeseran / translasi dapat dirumuskan sebagai berikut :

$$P_{(x,y)} \xrightarrow{T\left(\begin{matrix} a \\ b \end{matrix}\right)} P'_{(x+a, y+b)}$$

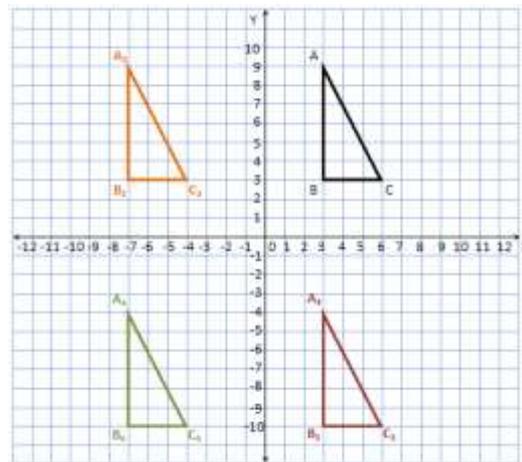
Dimana :

P = koordinat awal

P' = koordinat akhir (setelah di transformasi translasi)

a = pergeseran horisontal (+ ke kanan, - ke kiri)

$b$  = pergeseran vertikal (+ ke atas, - ke bawah)



**Gambar 2.4** Contoh Transformasi Translasi

Sumber: (Wibowo, 2017)

### 2.7.2 Rotasi (Perputaran)

Transformasi rotasi adalah jenis transformasi geometri yang digunakan untuk memutar objek atau titik dalam ruang terhadap suatu titik pusat atau sumbu rotasi. Transformasi ini mengubah orientasi atau posisi objek tanpa mengubah bentuk atau ukurannya. Dalam transformasi rotasi, objek atau titik diputar sejauh sudut tertentu dalam sekitar titik pusat tertentu atau sumbu rotasi. Pada Gambar 2. 5 segitiga ABC berwarna hitam mempunyai koordinat  $A(3,9)$ ,  $B(3,3)$ ,  $C(6,3)$ , apabila ditransformasikan rotasi mejadi segitiga ABC warna oranye, hijau dan ungu perhitungannya adalah sebagai berikut :

#### 1. Segitiga ABC oranye

Koordinat segitiga ABC oranye  $A_2(-9,3)$ ,  $B_2(-3,3)$ ,  $C_2(-3,6)$ . Maka pergeseran rotasi dengan sudut  $+90^\circ$  atau  $-270^\circ$  dengan pusat rotasi pada  $O(0,0)$ .

## 2. Segitiga ABC ungu

Koordinat segitiga ABC ungu A3(9,-3), B3(3,-3), C3(3,- 6). Maka pergeseran rotasi dengan sudut  $+270^\circ$  atau  $-90^\circ$  dengan pusat rotasi pada O(0,0). 16

## 3. Segitiga ABC hijau

Koordinat segitiga ABC hijau A4(3,-9), B4(-3,-3), C4(- 6,-3). Maka pergeseran rotasi dengan sudut  $+180^\circ$  atau  $- 180^\circ$  dengan pusat rotasi pada O(0,0).

Dari perhitungan di atas maka transformasi rotasi / perputaran dapat dirumuskan sebagai berikut :

$$\begin{bmatrix} x' \\ y' \end{bmatrix} = \begin{bmatrix} \cos \theta & -\sin \theta \\ \sin \theta & \cos \theta \end{bmatrix} \begin{bmatrix} x - a \\ y - b \end{bmatrix} + \begin{bmatrix} a \\ b \end{bmatrix}$$

Dimana :

x = koordinat x awal

y = koordinat y awal

x' = koordinat x akhir (setelah ditransformasi rotasi)

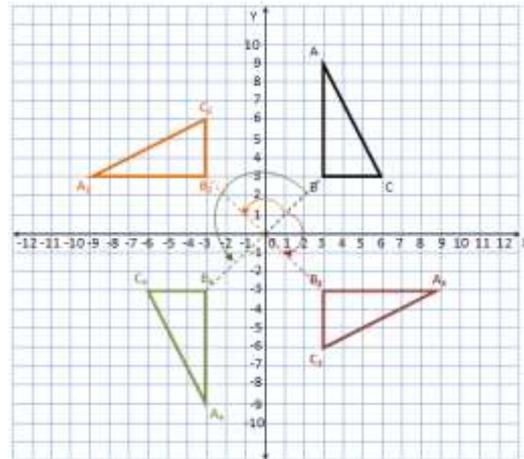
y' = koordinat y akhir (setelah ditransformasi rotasi)

$\theta$  = sudut putar

a = sumbu x pusat putar

b = sumbu y pusat putar

Pada transformasi rotasi, sudut putar ( $\theta$ ) bernilai positif (+) akan menyebabkan bangun berputar berlawanan arah jarum jam (*counter clockwise*) sedangkan apabila bernilai negatif (-) akan menyebabkan bangun berputar searah jarum jam (*clockwise*).

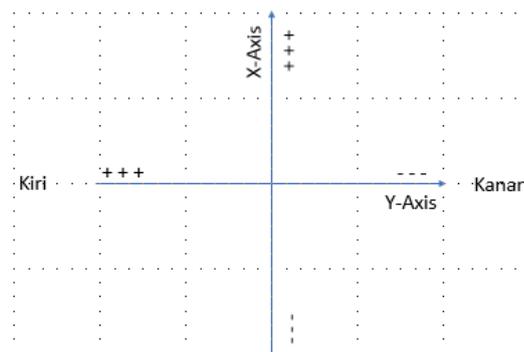


**Gambar 2.5** Contoh Transformasi Rotasi

Sumber: (Wibowo, 2017)

## 2.8 Sistem Koordinat Kartesian

Konstruksi robot terdiri dari tiga sumbu linier (prismatik). Sistem koordinat kartesius digunakan dalam matematika untuk menghitung posisi titik dalam ruang tiga dimensi menggunakan tiga nilai yang dikenal sebagai koordinat-x, koordinat-y, dan koordinat-z. Setiap sumbu memiliki kemampuan untuk bergerak dalam area sumbu x-y-z.



**Gambar 2.6** Koordinat Kartesian Pada Robot