

## Materi Menggerakan Kaki Robot

```
bool statusGerak = false;
bool modeGerak = true;
int steps = 0;
float theta = 90;
float tAwal, degAwal;
float tAkhir, degAkhir;
float Increment;
```

```
float xFR, yFR, zFR;
float xFR_Awal, xFR_Akhir;
float actual_xFR, actual_yFR;
float actual_zFR;
int outServo[4][3];
```

```
//KANAN DEPAN (FR)
float xFR0, yFR0, xFR1, yFR1, zFR0, zFRp;
```

```
//Variabel Invers Kinematik
const float cx = 22;
const int fm = 53;
const int tb = 70;
int heightFront = -92, heightBack = -92;
```

```
float z, sdtcoxa, sdtcoxa1, sdtcoxa2, sdtcoxa3, sdtcoxa4, sdtrotate, sdtfemur,
sdttibia, theta2, theta3, angle1, angle2, P, c, alas, alpha, beta;
const int legoffset[6] = { 0, 45, 135, 180, 225, 315 };
//Koordinat Awal (Standby) per Kaki :
const float standFR[3][1] = { { -55 }, { 55 }, { 0 } };
```

```
float offsetCX[4] = { 0 }; //0 = FR, 1 = BR, 2 = BL, 3 = FL
float rightFM = 0, rightTB = 0;
int jmlhStep;
```

```

void setup() {
  //Serial Untuk Mengirim Data sudut ke servo
  Serial2.setTx(PA2);
  Serial2.setRx(PA3);
  Serial2.begin(1000000);

  delay(3000);
  StandbyAwal();
  delay(1000);
}

void loop() {
  GerakDinamis(15, 15, 6, 0, 0);
  TrayektoriSinus();
  KirimIntruksiGerak(0);
  delay(5);
}

```

Pertama kali kita akan melihat fungsi **setup()**, dimana dalam fungsi ini kita akan mengatur persiapan yang kita pakai, mulai dari pin – pin yang digunakan dan kecepatan serial. Di akhir fungsi ini terdapat sebuah fungsi, dalam fungsi itu membuat Kaki robot berada di posisi **Standby()**, yaitu robot berdiri tegak.

```
StandbyAwal();
```

Dalam fungsi ini kita akan memanggil Invers-Kinematik yaitu

```
FR(-55, 55, 20);
```

Kita coba cari nilai yang ada di dalam program invers, untuk kode bisa lihat pada program yang dikirim pada file Invers-kinematik.ino. Tujuan akhir dari kode ini adalah mendapatkan sudut **coxa**, **femur** dan **tibia**. Kita akan mendapatkan hasil seperti ini saat memanggil fungsi diatas :

```

CX :658.13
FM :722.92
TB :733.15

```

Dimana hasilnya akan disimpan dalam sebuah array **outServo** yang nantinya akan menjadi nilai sudut untuk menggerakkan Servo. Setelah mendapatkan nilai sudut setiap servo, kita akan memanggil sebuah fungsi untuk mengirim sudut ke servo dan menggerakkan servo

```
KirimIntruksiGerak(512);
```

Dalam fungsi ini bisa kita temukan baris kode seperti ini

```
position_H = outServo[0][0] >> 8;  
position_L = outServo[0][0] & 0xff;  
id = 1;
```

Dalam kode ini terdapat position\_H untuk Pengiriman cepat, dimana angka sudut yang sudah disimpan ke variable **outServo** tadi akan dikonversi ke 8 bit untuk High dan 16 bit untuk Low ( ini merupakan kode default dari servo yang kita pakai ). Di baris setelahnya terdapat **id** atau nomor unik setiap servo. Karena kita ingin menggunakan nilai outServo[0][0] ini ke servo dengan **id** 1, maka kita set dengan **id** = 1 (diatur dengan alat yang nantinya akan diberitahukan saat progres) maka servo dengan id 1 nantinya akan bergerak sesuai nilai sudut.

Setelah servo bergerak, kita akan memanggil lagi fungsi yang sama dengan parameter berbeda.

```
FR(-55, 55, 0);
```

Jika dijalankan maka akan menghasilkan :

CX :658.13

FM :801.35

TB :821.81

Jika dilihat dari hasil sudut, yang berbeda hanyalah **FM** dan **TB**, semakin kecil sudut FM, maka servo akan semakin naik, begitu sebaliknya, semakin besar sudut, maka servo akan turun, kesimpulan ini didapatkan dari uji coba manual saat menggerakkan servo. Tujuan gerakan ini adalah untuk membuat robot berdiri tegak.

Setelah kita membuat robot berdiri tegak. Kita akan menjalankan fungsi **loop()**, fungsi ini akan dijalankan secara terus menerus atau berulang – ulang. Dalam fungsi ini, pertama ada sebuah fungsi Gerak Dinamis untuk memberikan arah gerakan apakah maju atau mundur.

```
void GerakDinamis(float Lebar, float tinggi, int speeds, float lKiri, float lKanan)
```

Jika maju parameter pertama bernilai positif, jika mundur parameter pertama bernilai negatif, dalam contoh ini robot akan bergerak maju, kita berikan nilai 15 untuk lebar, 15 untuk tinggi dan 6 titik, 0 lebar kanan dan 0 lebar kiri, karena kita belum menggunakan PID.

```
GerakDinamis(15, 15, 6, 0, 0);
```

Saat berada di dalam fungsi gerak dinamis, pertama kita akan dihadapkan dengan sebuah percabangan yaitu :

```
if (!statusGerak) {
```

karena **statusGerak** diawal bernilai False ( lihat Dokumentasi Variabel di awal Dokumentasi ini ).

```
bool statusGerak = false;
```

Jika di Not kan akan bernilai True, seperti  $1! = 0$  atau  $0! = 1$

```
Increment = 180 / speeds;
```

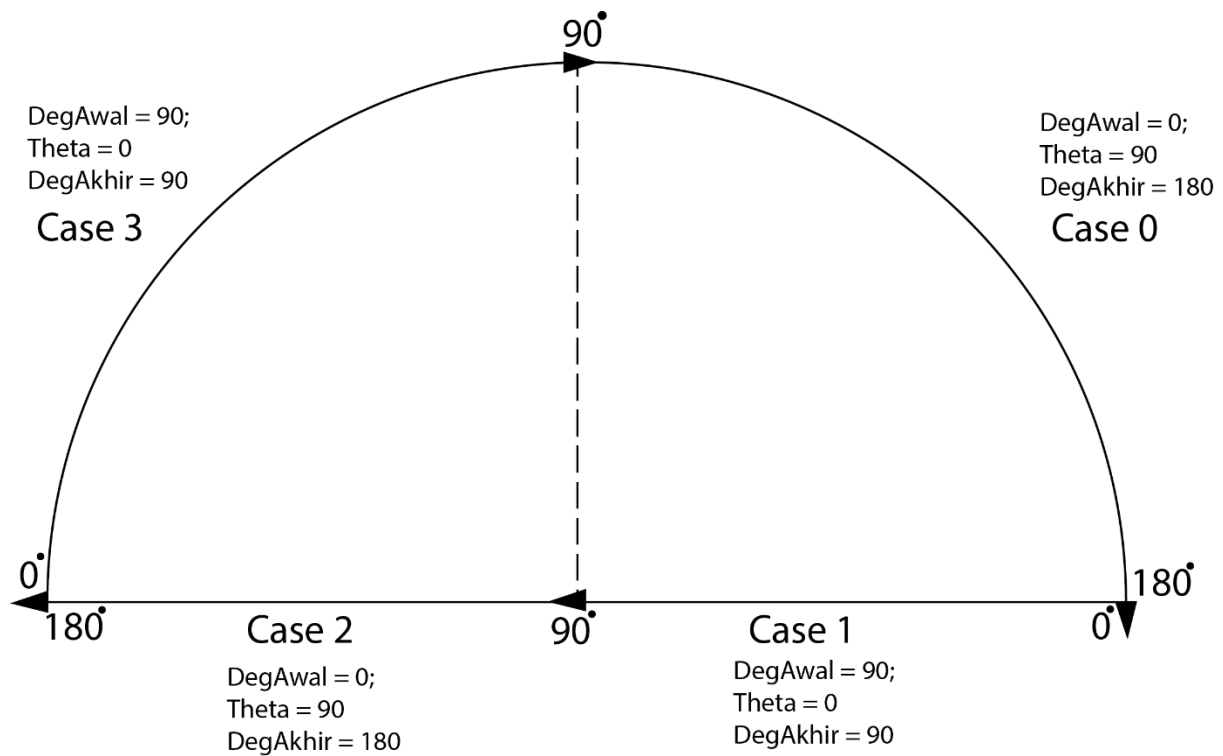
Karena kita memberikan speed = 6, maka  $180 / 6 = 30$ . Jadi increment = 30;

```
switch (steps) {
```

Karena Steps diawal bernilai 0 ( lihat Dokumentasi Variabel di awal Dokumentasi ini ). Maka program akan menjalankan **Case 0** :

```
degAwal = 0 + Increment;  
degAkhir = 180;
```

Disini kita Memberikan degAwal ditambah Increment, jadi degAwal = 30 dan degAkhir atau derajat akhir = 180. Bisa perhatikan gambar dibawah :



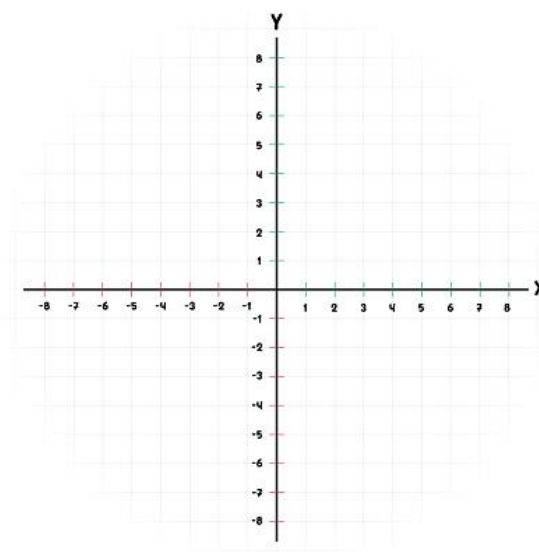
Untuk memahami Degawal dan DegAkhir, bisa diperhatikan gambar yang ada di atas. Pada dasarnya, jika dilihat dengan video slowmotion, kaki robot sebenarnya bergerak seperti setengah lingkaran, dimana dalam gerakan tersebut terdapat 4 bagian ( case 0 – 3 ). Nilai degAwal digunakan untuk menentukan sudut kaki berikutnya ( mengatur nilai theta berikutnya ), theta digunakan untuk menyimpan nilai sudut kaki saat ini, dan degAkhir menyimpan nilai tujuan sudut saat ini.

Saat pertama kali robot dijalankan, semua kaki robot akan berada di posisi tegak ditengah ( standby() ), yaitu nilai theta = 90, oleh karena itu, nilai theta saat pertama kali diberikan nilai 90 ( lihat Dokumentasi Variabel di awal Dokumentasi ini ). Ketika robot dijalankan, kaki robot akan mencari sudut tujuannya dari theta ke degAkhir ( kita anggap berada di case 0, jadi nilai degakhir 180 ) dengan menambahkan nilai theta sesuai dengan nilai increment. Saat nilai theta sudah lebih besar dari degAkhir maka, kaki robot akan beralih dari case 0 ke case 1, dimana sekarang degAwal 90 ( untuk menentukan nilai theta berikutnya ), dan nilai theta sesuai nilai degAwal tadi yaitu 0 dan degAkhir adalah 90, jadi sekarang kaki robot akan bergerak dari derajat ke 0 sampai derajat ke 90, dan begitu seterusnya.

Perlu diingat, sudut 0 dan 180 itu bernilai sama, dimana  $0 == 180$  dan kaki robot harus bergerak searah dengan jarum jam, jadi pada case 0 robot akan bergerak dari 90 derajat ke 180 derajat. Lalu kenapa kita tidak memberi nilai 180 saja, Kenapa kita ganti ke 0?, jadi jika kita berikan nilai degAwal ( pada case 0 ) 180, dan pada case 1 nilai degAkhir atau tujuan kita

berada di 90, maka kaki robot akan kembali ke posisi awal di case 0 yaitu pada theta = 90, sehingga kaki robot akan berlawanan arah jarum jam. Untuk mempermudah memahaminya kita andaikan seperti ini,  $0 > 90 > 180$ , jadi jika dari 180 ke 90 maka kaki robot akan bergerak ke kiri  $90 < 180$ . Oleh karena itu, kita harus mengganti ke 0 supaya kaki robot bergerak searah jarum jam.

Karena gerakDinamis ini digunakan untuk robot melangkah maju dan mundur, jika parameter lebar bernilai positif, robot bergerak maju, jika parameter lebar bernilai negatif maka robot akan bergerak mundur. Jadi kita akan mengatur nilai Y awal dan Y akhir. ( bisa dilihat dalam gambar dibawah ).



Kordinat Y untuk Maju dan mundur, Kordinat X untuk Geser kanan dan Kiri. Sekarang kita akan mengatur nilai titik Y awal dan Y akhir dengan memanggil fungsi :

```
TransformasiGerak(LebarY, lbrKiri, lbrKanan);
```

Untuk nilai parameter Lebar, lkiri dan lkanan mengambil dari fungsi gerakDinamis :

```
GerakDinamis(15, 15, 6, 0, 0);
```

Maka Lebar = 15, lkiri = 0, lkanan = 0, sekarang kita coba akses kode yang ada dalam fungsi TransformasiGerak :

```
yFR_Awal = (standFR[1][0] - lebarY) + lbrKanan, yFR_Akhir = (standFR[1][0] +  
lebarY) - lbrKanan;
```

standFR

Sekarang kita lihat variabel StandFR :

```
const float standFR[3][1] = { { -55 }, { 55 }, { 0 } };
```

StandFR merupakan sebuah array, dimana  $FR[0][0] = -55$ ,  $FR[1][0] = 55$ ,  $FR[2][0] = 0$ .  
Jadi  $yFR\_Awal = 55 - 15$ ,  $yFR\_Akhir = (55 + 15) - 0$ , kita simpan nilai ini dan kembali ke fungsi gerakDinamis.

```
xFR0 = [0][0], yFR0 = yFR_Awal, xFR1 = standFR[0][0], yFR1 = yFR_Akhir,  
zFRp = tinggi;
```

Sekarang kita coba untuk menginput nilai ke setiap variabel

$xFR0 = -55$   $xFR1 = -55$

$yFR0 = 40$   $yFR1 = 70$

$zFR = 15$

Jika dilihat dari nilai variabel, maka kaki robot akan maju dari titik  $y40$  ke  $y70$ , nilai  $XFR0$  dan  $XFR1$  sama bernilai  $-55$  karena robot hanya maju dan tidak bergeser, untuk  $zFR$  bernilai  $15$ , jadi ketika kaki robot naik akan bernilai  $15$ . Setelah mendapatkan titik awal dan akhir, sekarang akan keluar dari case 0 dan mengatur nilai variabel :

```
modeGerak = true;  
statusGerak = true;  
break;
```

`modeGerak` kita gunakan untuk mengatur titik aktual dari  $xFR$  dan  $yFR$  sedangkan `statusGerak` kita gunakan untuk menjaga supaya fungsi gerakDinamis tidak diakses sampai  $\theta >$  dari  $\theta_{Akhir}$ , atau tujuan derajat akhir. (misalkan  $\theta$  sekarang  $90$  derajat akhir  $180$ ). Setelah mengatur nilai beberapa variabel, program akan keluar dari case karena perintah `break` dan menambahkan `steps` :

```
steps++;
```

`steps++` artinya  $steps = steps + 1$ , maka  $steps = 0 + 1 = 1$ , jadi nilai `steps` sekarang adalah 1. Setelah itu kita akan mengecek sebuah percabangan, yaitu :

```
if (steps > 3) steps = 0;
```

Percabangan ini berfungsi mengatur ulang nilai `steps` saat lebih besar dari 3, karena didalam case, kita memiliki `steps` 0 sampai 3, maka saat `steps` melebihi 3 akan diatur lagi ke

awal atau 0. Karena  $\text{steps} > 3$  atau  $1 > 3 = \text{false}$  atau salah, maka sekarang kita tidak akan mereset nilai steps. Setelah selesai mengecek fungsi gerakDinamis, sekarang kita akan ke fungsi trayektoriSinus :

```
TrayektoriSinus();
```

Jika kita lihat, terdapat sebuah percabangan di awal kode :

```
if (modeGerak == true) {
```

Karena modeGerak sudah kita atur ke true pada fungsi gerakDinamis, maka sekarang kita akan mengakses baris kode yang terdapat di dalamnya :

```
actual_xFR = ((xFR1 - xFR0) / 10) * 2;  
actual_yFR = ((yFR1 - yFR0) / 10) * 2;
```

jika kita lihat, terdapat baris kode yang mengatur nilai aktual X dan Y, jika kita inputkan nilai tadi maka hasilnya :

$$\text{actual\_xFR} = ((-55 - (-55) / 10) * 2 = 0$$

$$\text{actual\_yFR} = ((70 - 40) / 10) * 2 = 6$$

setelah kita mendapatkan nilai aktual, modeGerak kembali di atur ke false, supaya kita hanya mengatur nilai actual hanya sekali ketika pertama kali memulai steps di gerakDinamis

```
modeGerak = false;
```

saat kita sudah mendapatkan nilai actual, selanjutnya terdapat sebuah percabangan lagi, dimana percabangan ini akan dijalankan ketika theta kurang dari degAkhir, karena theta di awal kita atur ke 90 dan degAkhir diatur 180, karena kondisinya benar, maka percabangan ini akan dijalankan. Sekarang kita akan lihat kode didalamnya :

```
//KANAN ATAS  
xFR = xFR0 + (((actual_xFR * theta) / 360)) * 10;  
yFR = yFR0 + (((actual_yFR * theta) / 360)) * 10;  
zFR = zFRp * sin(theta * (M_PI / 180));
```

dalam percabangan ini, kita akan mengatur nilai xFR, yFR dan zFR, kita coba dulu cari nilainya :

$$\text{xFR} = -55 + (((0 * 90) / 360)) * 10 = -55$$

$$\text{yFR} = 40 + (((6 * 90) / 360)) * 10 = 55$$

$$\text{zFR} = 15 * \sin(90 * (M\_PI / 180)) = 1.58$$



Setelah kita mendapatkan nilai xFR, yFR dan zFR, maka nilai ini akan dijadikan sebagai parameter ke fungsi invers-kinematik :

```
FR(xFR, yFR, zFR);
```

Untuk proses pada fungsi ini sama seperti yang sudah kita bahas di atas, dimana nantinya akan menghasilkan sudut setiap servo ( servo coxa, servo femur, servo tibia ) yang akan disimpan di variable outServo. Setelah mendapatkan sudut servo, kita akan menambahkan theta :

```
theta += Increment;
```

Dalam kode ini kita akan menambahkan nilai theta, kode ini berarti  $\text{theta} = \text{theta} + \text{increment}$ , dimana theta akan bernilai  $90 + 30 = 120$ , nilai increment sudah kita dapatkan tadi di fungsi gerakDinamis yaitu 30. Jadi sekarang theta bernilai 120. Setelah menambahkan nilai theta, kita akan mengecek sebuah percabangan dibawahnya, yaitu :

```
if (theta > degAkhir) {
```

Jika kita lihat, maka 90 tidak lebih besar dari 180 maka  $90 > 180 = \text{false}$  ( salah ), maka kita tidak akan masuk ke percabangan ini. Setelah itu kita kembali ke fungsi loop dimana terdapat lagi kode yang akan dijalankan yaitu :

```
KirimIntruksiGerak(0);
```

Sama seperti yang sudah kita bahas, ini adalah fungsi untuk mengirim data sudut ke servo dan menggerakkan servo. Setelah kita menjalankan servo, terdapat sebuah fungsi lagi, yaitu :

```
delay(5);
```

Program akan terhenti selama 5 milisecond sebelum kembali ke kode paling atas pada fungsi loop(). Setelah program terhenti selama 5 microsecond, sekarang akan kembali mengakses kode paling atas yaitu :

```
GerakDinamis(15, 15, 6, 0, 0);
```

Kita akan ke gerak dinamis, jika kita ke fungsi gerak dinamis, kita akan menemui lagi percabangan yaitu :

```
if (!statusGerak) {
```

Karena terakhir setelah kita memanggil fungsi ini, kita mengatur nilai statusGerak ke true, maka jika true di not kan atau !true maka hasilnya adalah false, maka fungsi ini tidak akan

dijalankan. Setelah itu program akan menjalankan kode yang ada di bawah fungsi gerakDinamis yaitu :

```
TrayektoriSinus();
```

Di dalam kode ini kita akan menemukan percabangan lagi yaitu :

```
if (modeGerak == true) {
```

Karena terakhir saat kita mengakses kode di dalam percabangan ini, kita telah mengatur modeGerak menjadi false, sehingga kita tidak akan masuk ke percabangan ini. Selanjutnya kita akan menemukan percabangan lagi yaitu :

```
if (theta <= degAkhir) {
```

karena terakhir setelah mengakses kode ini kita sudah mengatur nilai theta = 120 dan degAkhir bernilai 180 maka  $120 \leq 180$ , maka kita akan masuk ke percabangan ini. Di dalam percabangan ini kita akan mengatur nilai xFR, yFR dan zFR :

```
xFR = xFR0 + (((actual_xFR * theta) / 360)) * 10;  
yFR = yFR0 + (((actual_yFR * theta) / 360)) * 10;  
zFR = zFRp * sin(theta * (M_PI / 180));
```

sekarang kita akan mengatur nilainya :

$$xFR = -55 + (((0 * 120) / 360)) * 10 = -55$$

$$yFR = 40 + (((6 * 120) / 360)) * 10 = 60$$

$$zFR = 15 * \sin(120 * (M\_PI / 180)) = 1.58$$

Setelah mendapatkan nilai dari xFR, yFR dan zFR, kemudian kita akan memanggil lagi fungsi invers-kinematik :

```
FR(xFR, yFR, zFR);
```

Setelah memanggil fungsi ini, maka kembali lagi seperti yang sudah dibahas, dimana fungsi ini akan memberikan nilai sudut untuk setiap servo dan dikirim ke variabel outServo yang menyimpan sudut setiap servo. Setelah ini kita kembali lagi ke fungsi trayektoriSinus, dimana akan menambahkan nilai theta

```
theta += Increment;
```

karena tadi theta bernilai 120, kemudian kita tambahkan increment yang bernilai 30 yang tadi didapatkan saat terakhir mengakses fungsi gerak dinamis, maka  $120 + 30 = 150$ , jadi

sekarang nilai theta = 150. Setelah menambahkan nilai theta, sekarang kita akan mengecek sebuah percabangan dibawahnya, yaitu :

```
if (theta > degAkhir) {
```

Karena 150 tidak lebih besar dari 180 maka  $150 > 180 = \text{false}$ , jadi kita tidak masuk ke percabangan ini. Sekarang kita kembali lagi ke fungsi loop dan mengakses kode setelahnya, yaitu :

```
KirimIntruksiGerak(0);
```

Sama seperti sebelumnya, fungsi ini akan menggerakkan servo dengan nilai sudut dari outServo yang sudah kita dapatkan saat memanggil fungsi invers-kinematik tadi. Setelah itu kita kembali ke loop dan mengakses kode setelahnya, yaitu :

```
delay(5);
```

Program akan berhenti selama 5 milisecond dan setelah itu akan kembali ke bagian atas loop yaitu :

```
GerakDinamis(15, 15, 6, 0, 0);
```

Saat masuk ke gerak dinamis, kita dipertemukan lagi dengan percabangan, yaitu :

```
if (!statusGerak) {
```

karena nilai dari statusGerak masih bernilai true, maka !true atau !1 adalah false atau 0, maka kita tidak akan masuk ke kode di dalam percabangan ini, sekarang kita kembali lagi ke fungsi loop, dimana setelah kode gerakDinamis ada kode :

```
TrayektoriSinus();
```

Jadi kita sekarang memanggil fungsi TrayektoriSinus, dimana kita menemukan percabangan yang pertama yaitu :

```
if (modeGerak == true) {
```

Karena modeGerak masih bernilai false, maka kita tidak akan masuk ke percabangan ini. Sekarang kita akan mengakses percabangan setelahnya, yaitu :

```
if (theta <= degAkhir) {
```

Jika kita ingat lagi, nilai theta sudah kita perbarui ke 150 dan nilai degAkhir adalah 180, maka  $150 \leq 180 = \text{true}$ , jadi kita akan masuk ke percabangan ini dan mengatur nilai xFR, yFR dan zFR,

```
xFR = xFR0 + (((actual_xFR * theta) / 360)) * 10;
```

```
yFR = yFR0 + (((actual_yFR * theta) / 360)) * 10;  
zFR = zFRp * sin(theta * (M_PI / 180));
```

Sekarang kita atur nilainya :

$$xFR = -55 + (((0 * 150) / 360)) * 10 = -55$$

$$yFR = 40 + (((6 * 150) / 360)) * 10 = 65$$

$$zFR = 15 * \sin(120 * (M\_PI / 180)) = 1.58$$

Setelah mendapatkan nilai dari xFR, yFR dan zFR, kita akan memanggil fungsi invers-kinematik yaitu :

```
FR(xFR, yFR, zFR);
```

Jadi kembali lagi, fungsi ini akan memberikan nilai sudut ke masing masing servo dan menyimpannya ke variable outServo. Sekarang kita kembali ke TrayektoriSinus dan menambahkan nilai theta yaitu 150 ditambah increment yang sudah didapatkan saat pertama kali mengakses fungsi gerakDinamis yaitu 30, maka  $150 + 30 = 180$ , sekarang nilai theta adalah 180. Setelah mendapatkan nilai theta, kita akan mengecek percabangan dibawahnya, yaitu :

```
if (theta > degAkhir) {
```

Karena  $180 > 180$  adalah false, ( $>$  lebih besar dari, jadi jika nilai sama adalah false,  $>=$  lebih besar atau sama dengan, jika nilai sama adalah true). Jadi karena nilainya false, kita tidak akan masuk ke percabangan ini. Sekarang kita kembali lagi ke fungsi loop, dimana terdapat fungsi yang menggerakkan servo yaitu :

```
KirimIntruksiGerak(0);
```

Jadi fungsi ini akan menggerakkan servo sesuai sudut yang tadi sudah didapatkan di fungsi invers-kinematik. Setelah menggerakkan servo, kita kembali lagi ke fungsi loop untuk melakukan delay, atau menghentikam program:

```
delay(5);
```

setelah program terhenti, sekarang kita kembali lagi ke baris pertama fungsi loop, yaitu :

```
GerakDinamis(15, 15, 6, 0, 0);
```

Dimana di dalam fungsi ini terdapat percabangan statusGerak tadi yang nilainya masih true maka  $!true$  adalah false, jadi kita tidak masuk ke dalam percabangan ini :

```
if (!statusGerak) {
```

karena kita tidak masuk ke percabangan ini, dan fungsi gerak dinamis selesai, maka sekarang kita kembali ke fungsi loop, dan memanggil fungsi setelah gerakDinamis, yaitu :

```
TrayektoriSinus()
```

Kembali kita mengecek fungsi ini, dan kita akan mengecek lagi percabangan pertama yaitu :

```
if (modeGerak == true) {
```

Karena nilai modeGerak masih false, maka kita tidak akan masuk ke percabangan ini, dan kita akan menemukan percabangan lagi yaitu :

```
if (theta <= degAkhir) {
```

Jika kita ingat lagi, nilai theta yang kita atur terakhir adalah 180 dan degAkhir adalah 180, maka  $180 \leq 180$  atau 180 kurang dari atau sama dengan 180, maka hasilnya adalah true, jadi kita masuk lagi ke percabangan ini dan mengatur nilai xFR, yFR dan zFR :

```
xFR = xFR0 + (((actual_xFR * theta) / 360)) * 10;  
yFR = yFR0 + (((actual_yFR * theta) / 360)) * 10;  
zFR = zFRp * sin(theta * (M_PI / 180));
```

Sekarang kita hitung lagi nilainya :

$$xFR = -55 + (((0 * 180) / 360)) * 10 = -55$$

$$yFR = 40 + (((6 * 180) / 360)) * 10 = 70$$

$$zFR = 15 * \sin(120 * (M\_PI / 180)) = 1.58$$

Setelah kita mendapatkan nilai xFR, yFR dan zFR, kita akan memanggil lagi fungsi invers-kinematik untuk mendapatkan nilai sudut setiap servo yang akan disimpan di variabel outServo :

```
FR(xFR, yFR, zFR);
```

Setelah kita mendapatkan nilai sudut setiap servo, sekarang kita akan mengatur nilai theta :

```
theta += Increment;
```

Nilai theta terakhir adalah 180, jadi  $180 + 30$  ( increment ) = 210, setelah itu kita akan mengecek lagi percabangan dibawahnya :

```
if (theta > degAkhir) {
```

Karena nilai theta 210 dan degAkhir adalah 180 maka  $210 > 180$  adalah true, jadi sekarang kita masuk ke dalam percabangan :

```
theta = degAwal;  
statusGerak = false;
```

Sekarang kita akan mereset nilai theta dengan degAwal yang sudah kita inialisasi saat pertama kali kita mengakses fungsi gerakDinamis yaitu 30, jadi  $\theta = \text{degAwal}$  atau  $\theta = 30$  dan statusGerak kita atur ke false, sehingga kita sekarang bisa mengakses gerakDinamis. Setelah kita mengatur beberapa nilai di percabangan ini, sekarang kita kembali ke fungsi loop dan memanggil fungsi untuk menggerakkan servo :

```
KirimIntruksiGerak(0);  
delay(5);
```

setelah program menggerakkan servo, program akan melakukan delay selama 5 milisecond, dan setelah program delay, kita akan kembali lagi ke baris pertama di fungsi loop yaitu gerakDinamis.

```
GerakDinamis(15, 15, 6, 0, 0);
```

Sekarang memanggil fungsi gerakDinamis, dan mengecek percabangan pertama, yaitu :

```
if (!statusGerak) {
```

Karena tadi kita sudah mengatur ulang nilai dari variable statusGerak ke false maka, !false adalah true ( !false = true atau  $!0 = 1$ ). Jadi sekarang kita akan masuk ke dalam percabangan ini, karena nilai steps tadi sudah kita atur ke 1, maka sekarang kita akan mengarah ke case 1.

```
case 1:  
    degAwal = 90 + Increment;  
    degAkhir = 90;  
    TransformasiGerak(Lebar, lKiri, lKanan);  
    //KANAN DEPAN (FR)  
    xFR0 = standFR[0][0], yFR0 = yFR_Akhir, xFR1 = standFR[0][0] , yFR1 = yFR_Awal, zFRp = 0;  
    //KANAN BELAKANG (BR)  
    xBR0 = standBR[0][0], yBR0 = yBR_Akhir, xBR1 = standBR[0][0], yBR1 = yBR_Awal, zBRp = 0;  
    //KIRI DEPAN (FL)  
    xFL0 = standFL[0][0], yFL0 = yFL_Akhir, xFL1 = standFL[0][0], yFL1 = yFL_Awal, zFLp = tinggi;
```

```
//KIRI BELAKANG (BL)
    xBL0 = standBL[0][0], yBL0 = yBL_Akhir, xBL1 = standBL[0][0], yBL1 =
yBL_Awal, zBLp = tinggi;
    modeGerak = true;
    statusGerak = true;
    break;
```

Disini kita ulangi lagi proses yang sudah kita pelajari dari pertama kali kita memanggil fungsi gerakDinamis. Mungkin sampai disini dulu pembahasan tentang kode untuk menggerakan robot. Selanjutnya, kita akan pelajari lagi materi penjadwalan atau RTOS dan HardwareTimer.