

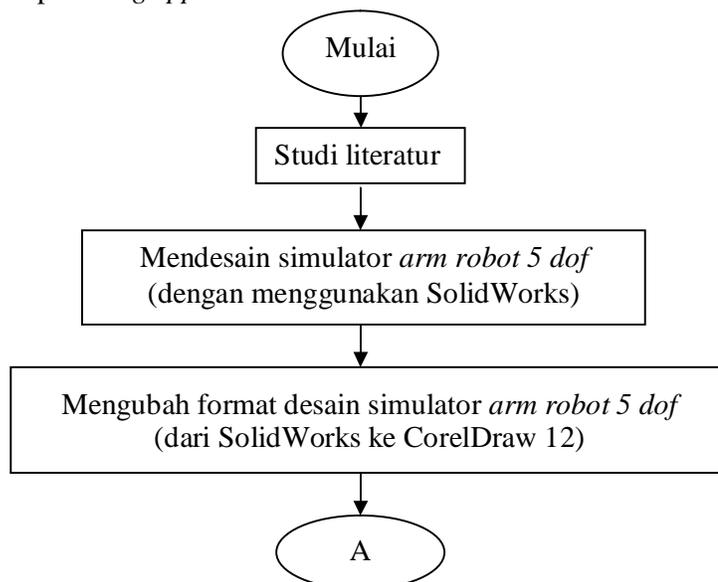
## BAB 3

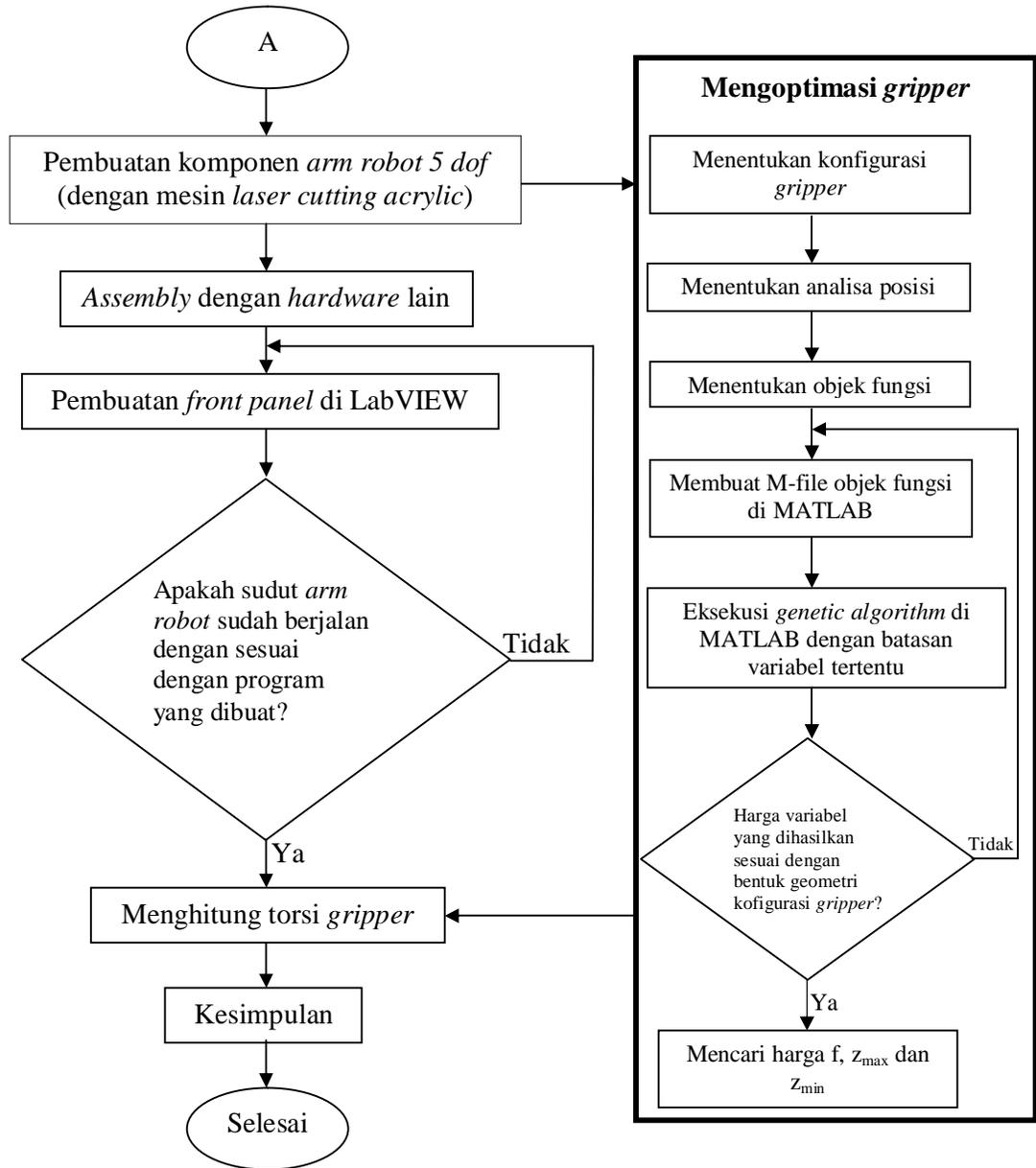
### DESAIN PEMBUATAN SIMULATOR *ARM ROBOT 5 DOF*

Dalam bab ini dijelaskan desain dan pembuatan simulator *arm robot 5 dof*, yang meliputi diagram alir pembuatan, desain simulator dengan menggunakan SolidWorks, desain *hardware* dan *software* serta pembuatan simulator *arm robot 5 dof*. Desain *hardware* yang digunakan diantaranya adalah mikrokontroler, *servomotor*, *power supply* (catu daya) dan regulator. Sedangkan desain *software* yang digunakan adalah pembuatan program *front panel* pada LabVIEW yang menggunakan Firmata Arduino. Untuk pembuatannya dibuat dari bahan akrilik yang kemudian dipotong dengan menggunakan *acrylic cutting laser machine*.

#### 3.1 Diagram Alir Pembuatan Simulator *Arm Robot 5 DOF*

Fokus penelitian ini salah satunya adalah pembuatan simulator *arm robot 5 dof* yang dilengkapi dengan *gripper*. Dengan adanya *gripper* yang didesain berdasarkan salah satu teknik optimasi yaitu *genetic algorithm*, maka *gripper* diharapkan dapat bekerja dengan efisien ketika berinteraksi dengan benda kerja atau lingkungannya. Selanjutnya gambar 3.1 dibawah ini merupakan diagram alir desain pembuatan simulator dan optimasi *gripper*.



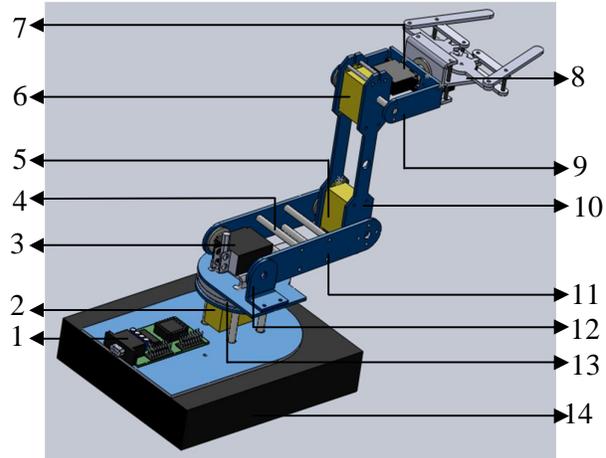


Gambar 3.1 Diagram alir desain pembuatan simulator dan optimasi gripper

### 3.2 Desain Simulator Arm Robot 5 DOF

Dalam pembuatan simulator *arm robot 5 dof* penulis melakukan pendesainan *link*, *joint* serta *gripper*. Dalam membuat desain simulator *arm robot 5 dof* beserta *gripper* ini, penulis merancangnya dahulu dengan menggunakan *software* SolidWorks. Desain pada SolidWorks dibuat dengan sudut pandang yang jelas yaitu sudut pandang *isometric*. Hal ini dilakukan agar dalam melakukan pembuatan simulator tersebut lebih

jelas dan dapat mengurangi kesalahan pada proses pembuatannya. Gambar 3.2 di bawah ini adalah skema pembuatan simulator *arm robot 5 dof* dengan sudut pandang *isometric*.



Gambar 3.2 Skema pembuatan simulator *arm robot 5 dof*

Keterangan Gambar 3.2 :

1 : Mikrokontroler	8 : <i>Gripper</i>
2 : <i>Joint 1</i>	9 : <i>Link 3</i>
3 : <i>Joint 2</i>	10 : <i>Link 2</i>
4 : Baut penghubung	11 : <i>Link 1</i>
5 : <i>Joint 3</i>	12 : Penyangga <i>link 1</i>
6 : <i>Joint 4</i>	13 : <i>Bearing</i>
7 : <i>Joint 5</i>	14 : <i>Base Utama</i>

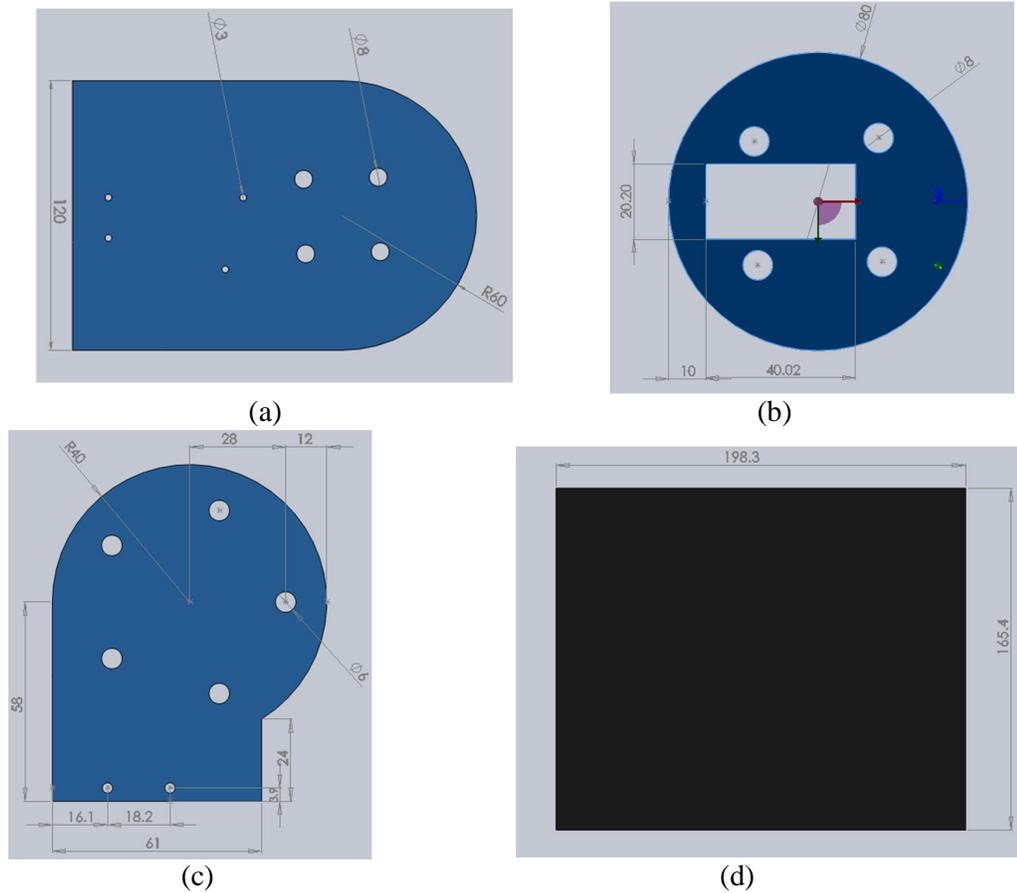
Dari Gambar 3.2 di atas, desain simulator *arm robot* dibagi dalam 5 komponen, diantaranya yaitu :

a. Komponen 1

Komponen 1 ini terdiri dari *base* dan *joint 1*. *Joint 1* terdiri dari 4 bagian, masing masing yaitu dudukan 1, *servomotor*, dudukan 2 dan dudukan 3. Gambar 3.3 di bawah ini adalah skema pada *joint 1*, semua dimensi dalam milimeter.



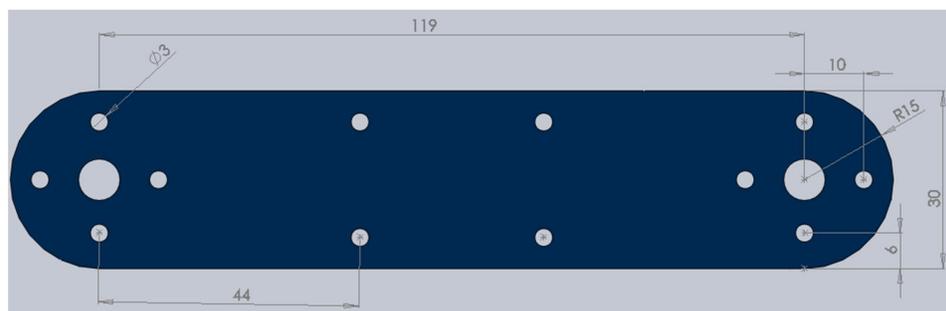
Gambar 3.3 Skema pada *joint 1*



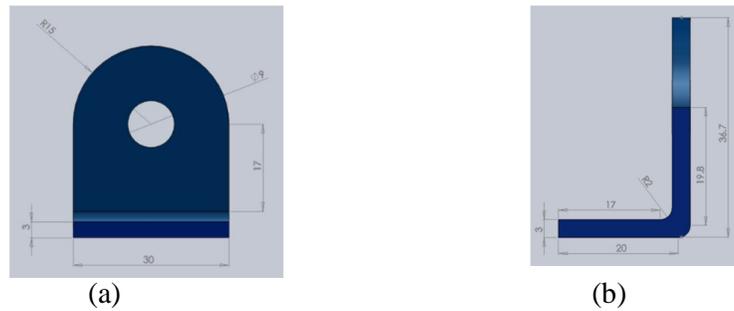
Gambar 3.4 Desain komponen 1: (a) dudukan 1, (b) dudukan 2, (c) dudukan 3, (d) *base* utama

b. Komponen 2

Komponen 2 ini terdiri dari *joint* 2, sebuah penyangga *link* 1 dan 2 buah *link* 1. *Link* ini sebagai penghubung *joint* 3. Bentuk beserta dimensi dari *link* 1 dapat dilihat pada gambar 3.5 berikut. Untuk penyangga *link* 1 dapat dilihat pada gambar 3.6.



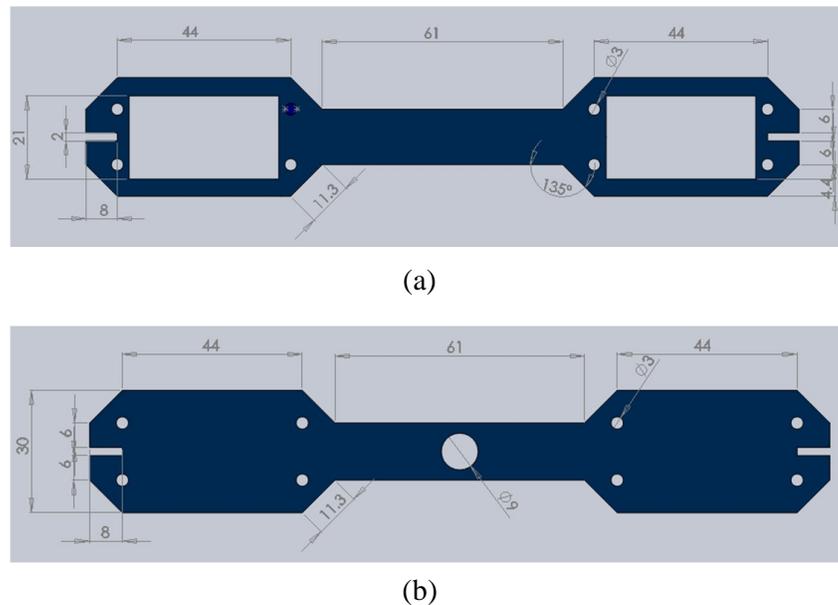
Gambar 3.5 *Link* 1



Gambar 3.6 Penyangga *link* 1: (a) tampak depan, (b) tampak samping

c. Komponen 3

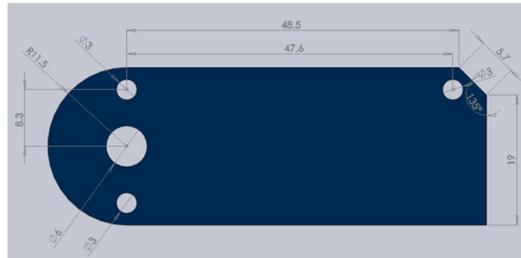
Komponen ini terdiri dari *joint* 3 dan menggunakan 2 buah *link* 2 untuk menghubungkan *joint* 2 ke *joint* 3. Pada gambar 3.7 sekilas dua buah *link* tampak sama. Yang membedakan dari keduanya hanyalah lubang lubang yang dibuat. Lubang ini untuk jalur kabel *servomotor*. Bentuk beserta dimensi dari *link* 2 dapat dilihat pada gambar 3.6 dibawah ini.



Gambar 3.7 *Link* 2 :(a) *Link* tanpa lubang, (b) *Link* dengan lubang

d. Komponen 4

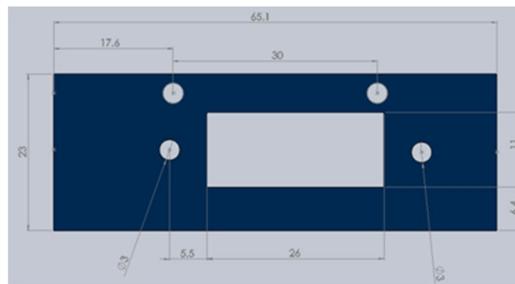
Komponen ini terdiri dari *joint* 4 dan menggunakan 2 buah *link* 3. *Link* 3 untuk menghubungkan ke *joint* 3 ke *joint* 4. Bentuk beserta dimensi dari *link* 3 dapat dilihat pada gambar 3.8.



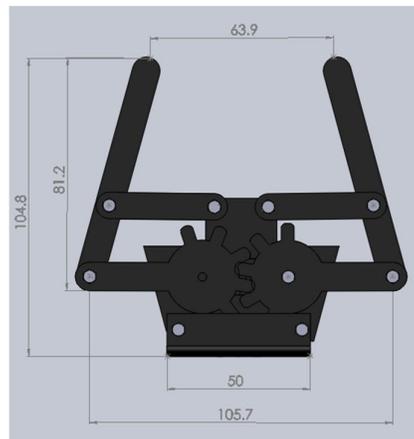
Gambar 3.8 Link 3

e. Komponen 5

Komponen 5 ini terdiri dari penghubung *link 3* dan *end effector*. *End effector* pada *arm robot 5 dof* ini adalah berupa *gripper* yang memiliki dua jari penjepit. *Gripper* dengan 2 jari penjepit ini banyak digunakan pada simulator *arm robot 5 dof*. Desain *gripper* yang dibuat dalam SolidWorks ini adalah desain yang belum teroptimasi. Untuk optimasi *gripper* dapat dilihat pada bab 4. Gambar 3.9 adalah penghubung *link 3* dan gambar 3.10 menunjukkan *gripper* dengan 2 jari penjepit, untuk dimensi dan bagian bagian lebih lengkapnya bisa dilihat pada lampiran.



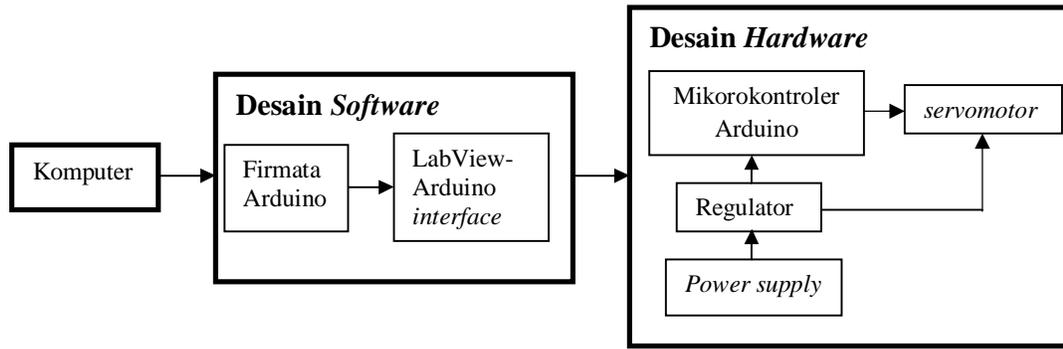
Gambar 3.9 Penghubung link 3



Gambar 3.10 Gripper dengan 2 jari penjepit

### 3.3 Desain *Hardware* dan *Software* Simulator *Arm Robot 5 dof*

Simulator *arm robot 5 dof* ini terdiri dari desain *hardware* dan *software*. Diagram desainnya bisa dilihat pada Gambar 3.11 berikut ini.



Gambar 3.11 Diagram desain *hardware* dan *software* simulator *arm robot 5 dof*

#### 3.3.1 *Hardware*

*Hardware* pada simulator *arm robot 5 dof* ini terdiri dari mikrokontroler, *servomotor*, *power supply* (catu daya) dan regulator.

##### 3.3.1.1 Mikrokontroler

Dalam sebuah sistem kontrol tertutup maka sangat diperlukan sebuah alat untuk mengkomputasi. Alat yang digunakan untuk mengkomputasi adalah mikrokontroler. Penulis menggunakan Arduino Uno SMD sebagai mikrokontroler. Ini menjadi pilihan penulis karena mikrokontroler ini mempunyai pin yang cukup banyak dan punya memori penyimpanan yang besar. Lalu untuk mengkompile program dari komputer ke mikrokontroler penulis menggunakan komunikasi data serial yang bisa dihubungkan dengan USB (*Universal Serial Bus*). Gambar 3.11 menunjukkan *hardware* Arduino Uno SMD dan Tabel 3.12 spesifikasi Arduino Uno SMD.



Gambar 3.12 Arduino Uno SMD

Tabel 3.1 Spesifikasi Arduino Uno SMD

No	Hal	Spesifikasi
1	Mikrokontroler	ATMEGA328
2	Tegangan operasi	5 Volt
3	Tegangan masukan yang disarankan	7-12 Volt
4	Batasan tegangan masukan	6-20 Volt
5	<i>Digital I/O pins</i>	14 (dimana 6 diantaranya sebagai keluaran PWM)
6	<i>Analog input pins</i>	6
7	Arus per I/O pin	40 mA
8	Arus untuk 3,3 volt pin	50 mA
9	<i>Flash memory</i>	32 kB
10	SRAM	2 kB
11	EEPROM	1 kB
12	<i>Clock speed</i>	16 Mhz

Arduino bisa mengambil sumber daya dari koneksi USB maupun dari sumber lainnya. Untuk sumber daya yang bukan diambil dari koneksi USB, bisa digunakan baterai ataupun adaptor yang memiliki koneksi *plug in center positive* sebesar 2,1 mm. Apabila menggunakan baterai, maka sumbu positif (+) baterai dihubungkan dengan pin Vin sedangkan sumbu negatif (-) baterai dihubungkan dengan pin GND pada mikrokontroler Arduino. Pin Vin yaitu tegangan masukan pada mikrokontroler Arduino sedangkan pin GND adalah *ground* pin. Arduino dapat dijalankan dengan sumber daya dari luar dengan rentang tegangan antara 6-20 volt, apabila kurang dari 7 volt maka pin 5v akan menghasilkan tegangan kurang dari 5 volt dan menyebabkan mikrokontroler Arduino menjadi tidak stabil. Dan apabila diberi tegangan masukan lebih dari 12 volt, maka *voltage regulator* akan menjadi cepat panas. Direkomendasikan rentang tegangan yang dipakai yaitu 7-12 volt. Beberapa pin yang ada pada mikrokontroler Arduino diantaranya adalah pin 5V yaitu pin yang menghasilkan tegangan sebesar 5 volt dan pin 3,3V adalah pin yang menghasilkan tegangan 3,3 volt dengan arus sebesar 50 mA.

### 3.3.1.2 Servomotor

*Servomotor* digunakan untuk menggerakkan *link* pada *arm robot*. Pada *joint 1*, *joint 3* dan *joint 4* menggunakan *servomotor* dengan merek Hobbyking HK-15288A. sedangkan pada *joint 2* menggunakan *servomotor* dengan merek Turnigy. Untuk *joint 5*

dan *gripper* menggunakan *servomotor* merek Hitec. Tabel 3.2 berikut ini adalah spesifikasi *servomotor* untuk masing masing *joint*.

Tabel 3.2 Spesifikasi *servomotor* untuk masing-masing *joint*

No	Joint	Servomotor	Spesifikasi <i>servomotor</i>
1	1, 3 dan 4	Hobbyking HK-15288A	Tegangan : 4,8v-6v Torsi : 8kg.cm @4,8v 9kg.cm @6v Berat : 51 gram Kecepatan : 0,21 s / 60deg @ 4,8v 0,20 s / 60 deg @ 6v
2	2	Turnigy Metal Gear	Tegangan : 4,8v-6v Torsi : 15,5kg.cm @4,8v 17kg.cm @6v Berat : 60 gram Kecepatan : 0,16 s/ 60deg @ 4,8v 0,14 s/ 60 deg @ 6v
3	5 dan <i>gripper</i>	Hitec HS-81	Tegangan : 4,8v-6v Torsi : 2.6kg.cm @4,8v 3kg.cm @6v Berat : 51 gram Kecepatan : 0,11 s/ 60deg @ 4,8v 0,09 s / 60 deg @ 6v



(a)



(b)



(c)

Gambar 3.13 Berbagai jenis *servomotor*:

(a) Hobbyking HK-15288A, (b) Turnigy Metal Gear, (c) Hitec HS-81

### 3.3.1.3 Power Supply (Catu Daya)

Dalam menggerakkan *servomotor* diperlukan *power supply* (catu daya). Penulis disini menggunakan baterai DC (*Direct Current*) dengan merek Turnigy yang memiliki kapasitas 1.300 mAh, tegangan *output* sebesar 11,1 volt dan memiliki 3 *cell* di dalamnya. Gambar 3.14 menunjukkan baterai Turnigy yang digunakan.

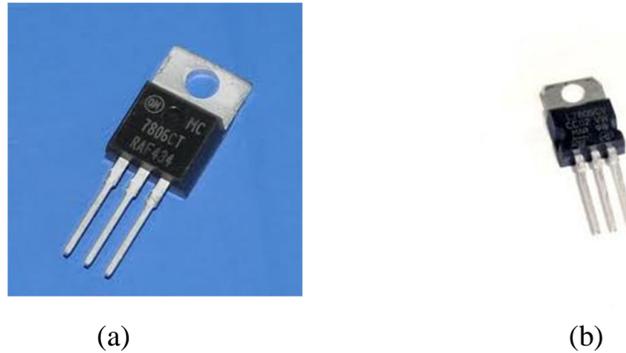


Gambar 3.14 Baterai Turnigy

### 3.3.1.4 Regulator

Regulator adalah rangkaian regulasi atau pengatur tegangan keluaran dari sebuah catu daya agar efek dari naik atau turunnya tegangan tidak mempengaruhi tegangan catu daya sehingga menjadi stabil. Seperti sudah dijelaskan sebelumnya bahwa penulis menggunakan baterai dengan merek Turnigy yang memiliki kapasitas 1300 mAh, tegangan 11,1 volt. Sehingga diperlukan penurunan tegangan sebesar 6 volt untuk *servomotor* HK-15288A dan *servomotor* merek Turnigy, 5 volt untuk *servomotor* HS-81.

Untuk menurunkan tegangan ke 6 volt diperlukan perangkat elektronik IC-7806 sedangkan untuk menurunkan tegangan ke 5 volt diperlukan IC-7805. Gambar 3.15 berikut ini menunjukkan IC-7806 dan IC-7805.

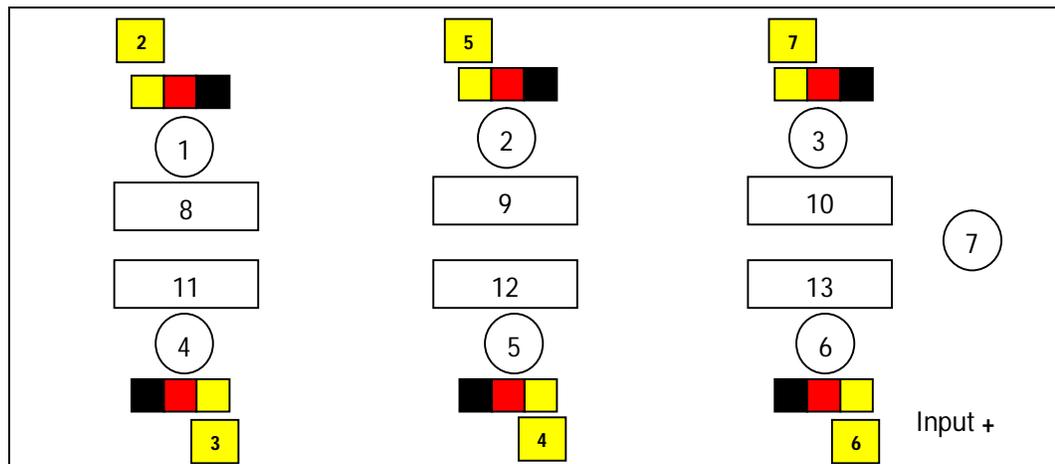


Gambar 3.15 Beberapa tipe IC: (a) IC-7806, (b) IC-7805

Tiap-tiap *servomotor* memerlukan 1 buah IC, sehingga total diperlukan IC-7806 sebanyak 4 buah dan IC-7805 sebanyak 2 buah. Tiap IC diperlukan kapasitor pada kaki *input* dan *output*. Kapasitor adalah komponen elektronika yang dapat menyimpan muatan listrik. Kapasitor yang digunakan dalam rangkaian regulator ini adalah jenis kapasitor elektrolit. Kapasitor ini biasanya berbentuk tabung yang mempunyai dua kutub kaki yaitu berpolaritas positif dan negatif. Kaki yang berpolaritas positif ditandai dengan kaki yang panjang sedangkan yang berpolaritas negatif ditandai dengan kaki yang pendek. Nilai kapasitansnya dari 0,47  $\mu\text{F}$  (mikroFarad) sampai ribuan mikroFarad dengan voltase kerja dari beberapa volt hingga ribuan volt. Penulis menggunakan kapasitor dengan kapasitas 100  $\mu\text{F}$  dengan tegangan 16 volt untuk IC 7805 maupun IC 7806 dan 1000  $\mu\text{F}$  dengan tegangan 16 volt untuk seluruh rangkaian regulator. Gambar 3.16 berikut ini adalah salah satu jenis kapasitor elektrolit dan gambar 3.17 menunjukkan *mapping* dari regulator.



Gambar 3.16 Kapasitor elektrolit

Gambar 3.17 Regulator *mapping*

Keterangan Gambar 3.17 :

1-7 : Kapasitor

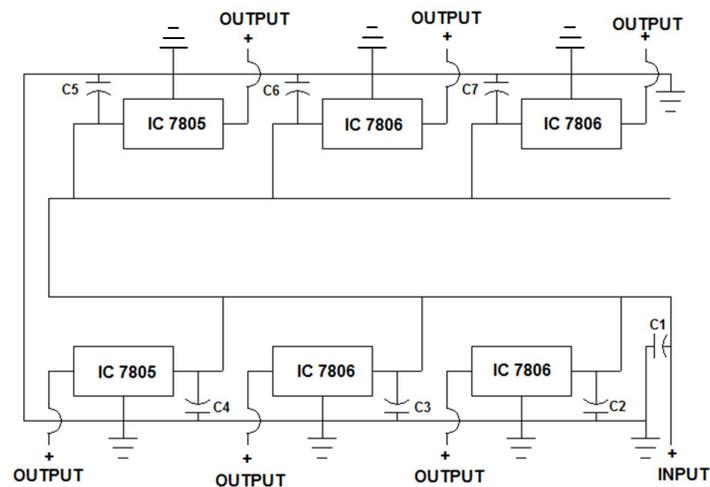
8 dan 11 : IC 7805

9,10, 12, 13 : IC 7806

 : Pin pada *servomotor*, terdiri dari 3 yaitu *ground* (warna hitam), *positive* (warna merah) dan *signal* (warna kuning).

 : Pin signal *servomotor* yang terhubung pada pin Arduino. Angka yang tertera di dalamnya merupakan pin Arduino.

Untuk rangkaian regulator bisa dilihat pada Gambar 3.18. Sedangkan Gambar 3.19 menunjukkan regulator yang digunakan.

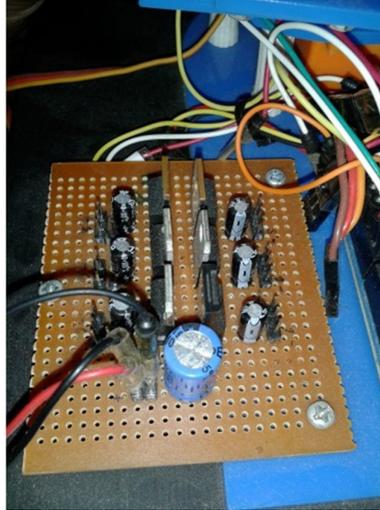


Gambar 3.18 Rangkaian regulator

Keterangan pada gambar 3.18 :

C1 : Kapasitor 1000  $\mu$ F, 16 volt

C2-C7 : Kapasitor 100  $\mu$ F , 16 volt



Gambar 3.19 Regulator yang digunakan

### 3.3.2 Software

#### 3.3.2.1 Firmata Arduino

Untuk Firmata pada Arduino-LabView *interface* terdiri dari 7 *file* yang masing masing adalah kode LVIFA\_Base, AFMotor.cpp, AFmotor.h, AccelStepper.cpp, AccelStepper.h, LabViewInterface.h, dan LabViewinterface. Bahasa pemrograman dari kesemua *file* tersebut bisa dilihat pada lampiran. Kemudian pada LVIFA\_Base di *upload* ke mikrokontroler Arduino.

#### 3.3.2.2 Pengaturan Blok Diagram pada LabView

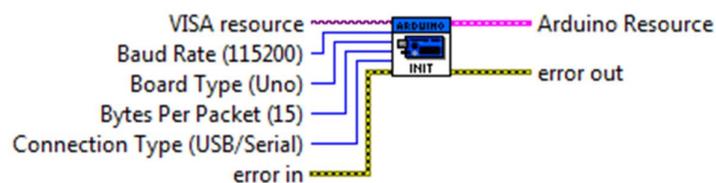
Setelah LVIFA\_Base di *upload* ke Arduino, kemudian membuat program di LabVIEW. Langkahnya adalah sebagai berikut ini :

1. Buka program LabVIEW, kemudian pilih *Blank VI*. Agar lebih jelasnya bisa dilihat lingkaran merah pada gambar 3.20 berikut ini.



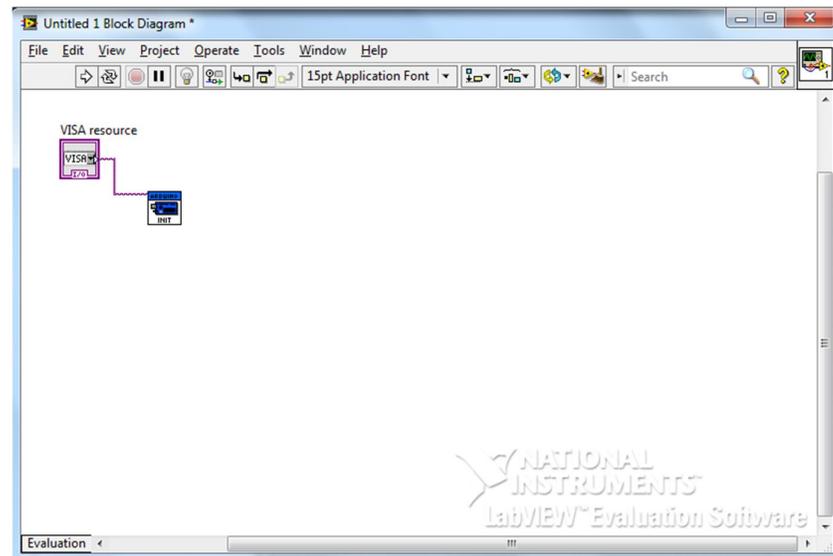
Gambar 3.20 Pemilihan Blank VI pada halaman utama LabVIEW

2. Pilih jendela *block diagram* dan klik kanan pada jendela *block diagram* kemudian pilih “Arduino è init” kemudian letakkan di jendela *block diagram*. LIFA (*LabVIEW Interface Arduino*) Init, untuk mengatur setting komunikasi dengan Arduino, termasuk *port* yang digunakan (*visa resource*), tipe Arduino (*board type*), Pada gambar 3.21 dibawah ini adalah LIFA Init.



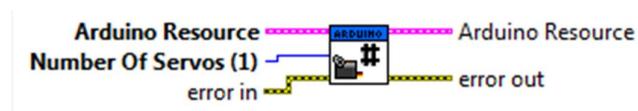
Gambar 3.21 LIFA Init<sup>[11]</sup>

Kemudian klik kanan *visa resource* pada LIFA init, pilih “create è control”. Control disini fungsinya agar masukan dari Arduino melalui *port* yang dapat kita pilih di jendela *front panel* nantinya. *Visa resource* sebagai *port* masukan dari hardware, bisa melalui USB, *bluetooth* atau yang lainnya. Gambar 3.22 berikut menunjukkan LIFA init dan pada *block diagram*.



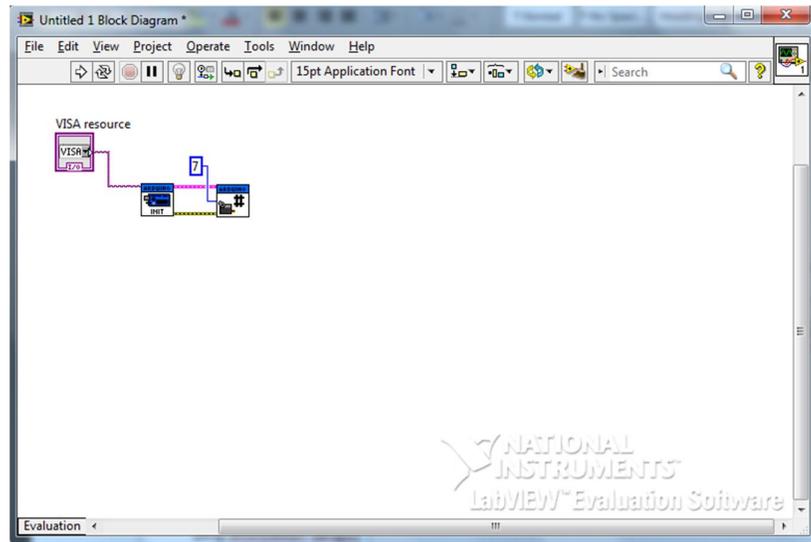
Gambar 3.22 LIFA init dan *visa resource* pada *block diagram*

3. Selanjutnya klik kanan pada jendela *block diagram*, pilih “Arduinoè *sensorsè servoè set number of servos*”. LIFA *set number of servos*, untuk menentukan berapa banyak pin yang akan dikendalikan. Dimulai dari pin 0 pada papan Arduino. Pada pembuatan *arm robot 5 dof* ini *set number of servos* di buat *constant* pada nilai 7, dengan cara klik kanan pada *number of servos* kemudian pilih “*createè constant*”. Gambar 3.23 berikut in adalah LIFA *set number of servos*.



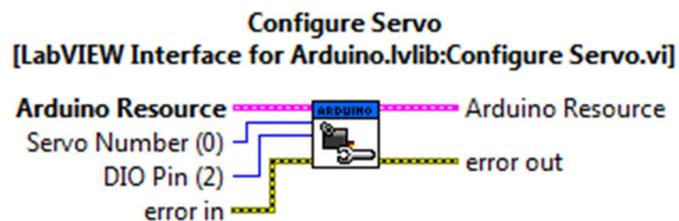
Gambar 3.23 LIFA *set number of servos*<sup>[11]</sup>

Kemudian diletakkan di sebelah kanan LIFA init tadi dan dihubungkan Arduino resource LIFA init ke *Arduino resource set number of servos* yang ditunjukkan dengan *wire* warna merah muda, begitu juga dengan *error out* LIFA init dihubungkan dengan *error in set number of servos* yang ditunjukkan dengan *wire* berwarna coklat. Untuk lebih jelasnya dapat dilihat pada Gambar 3.24 di bawah ini.

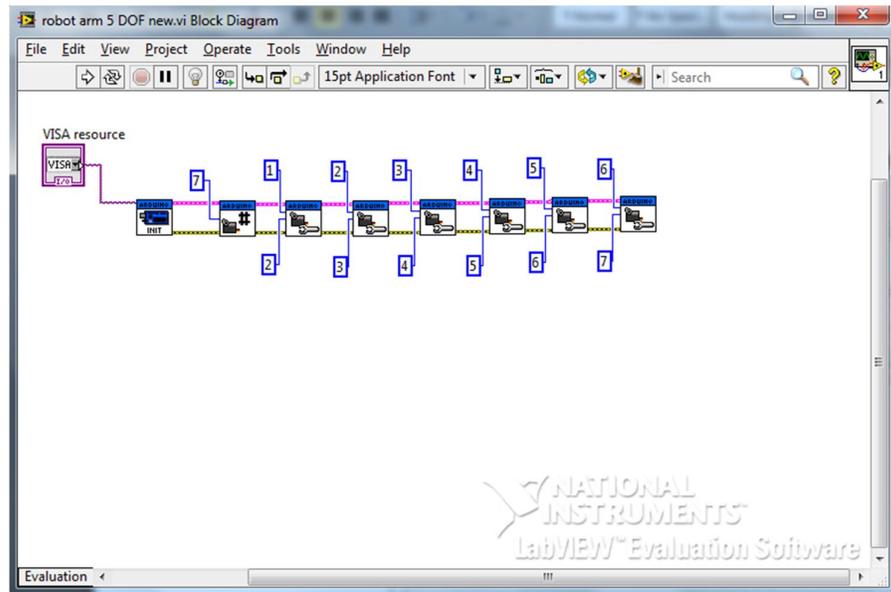


Gambar 3.24 Peletakan *set number of servos* pada *block diagram*

4. Selanjutnya klik kanan pada jendela *block diagram*, pilih “Arduino sensors servo configure servos. LIFA configure servo ini untuk penugasan kaki digital mana yang akan digunakan untuk mengendalikan *servomotor* yang ditunjukkan dengan DIO Pin dan nomor *servomotor* yang ditunjukkan dengan *servo number*. Untuk masing-masing harganya diberikan *constant* (seperti pada langkah 3 sebelumnya) yaitu 1 dan 2. Karena pada *arm robot 5 dof* terdiri dari 6 buah *servomotor* maka dapat dibuat 6 buah LIFA *configure servo* yang masing-masing harga *constant*-nya dibuat berurutan dari harga awal yang telah diberikan tadi. Untuk lebih jelasnya bisa dilihat pada gambar 3.25. dan gambar 3.26 berikut ini adalah LIFA *configure servo*.



Gambar 3.25 LIFA *configure servo*<sup>[11]</sup>

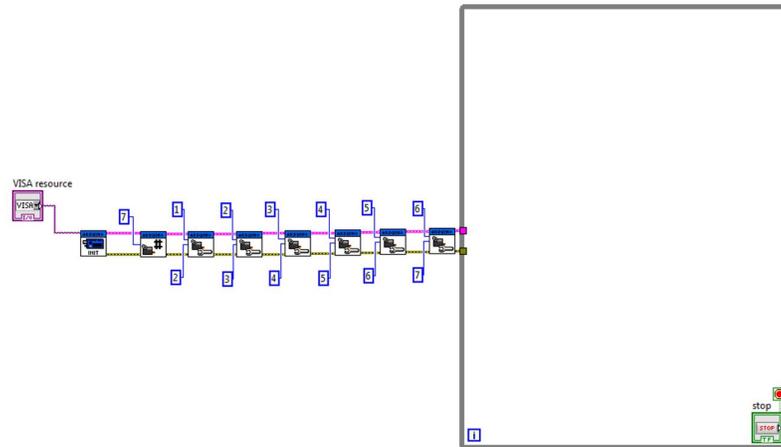


Gambar 3.26 Peletakan *configure servo* pada *block diagram*

5. Membuat *while loop*. Struktur ini terus mengesekusi ikon-ikon yang ada di dalamnya berulang kali hingga terminal *input conditional (stop if true)* mendapat nilai *false*, sebaliknya perulangan akan berhenti ketika mendapat nilai *true*. Struktur *while loop* memiliki 3 komponen utama, yaitu sebuah blok yang dapat diatur luasannya, sebuah terminal *input conditional (stop if true)* dan sebuah terminal *output counter (i)*. Gambar 3.27 di bawah ini adalah struktur *while loop*. Dan pada gambar 3.28 menunjukkan peletakan struktur *while loop* pada jendela *block diagram*.



Gambar 3.27 *While loop*<sup>[11]</sup>



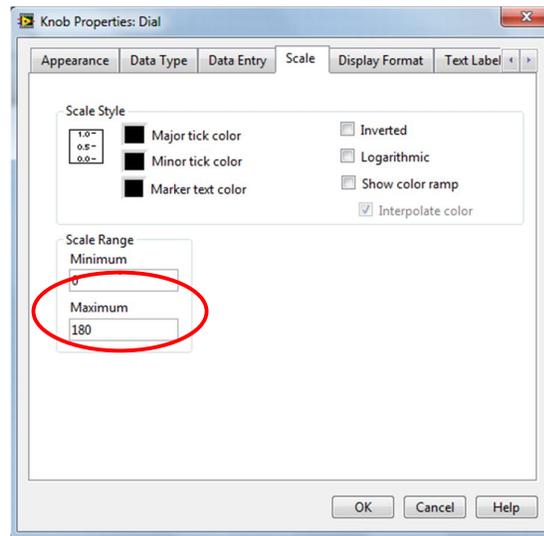
Gambar 3.28 Peletakkan struktur *while loop* pada jendela *block diagram*

6. Selanjutnya membuat *servo write angle*. LIFA *servo write angle* untuk menentukan berapa besar sudut putar *servomotor*. Harga *constant* pada *servo number* menyesuaikan dengan harga *servo number* pada *configure servos*. Gambar 3.29 dibawah ini adalah LIFA *servo write angle*.



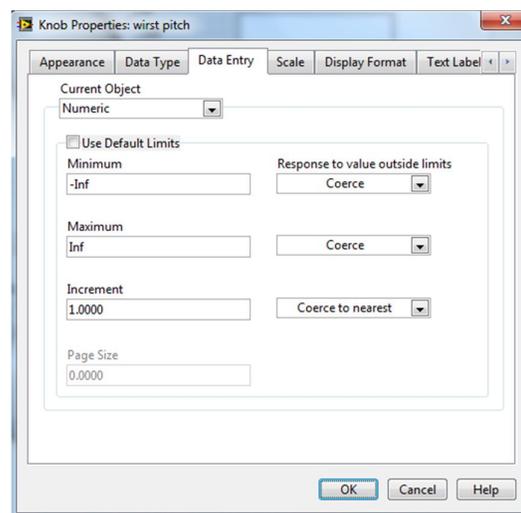
Gambar 3.29 LIFA *servo write angle*<sup>[11]</sup>

Untuk Gambar 3.29 diatas, pada *angle (degrees)* dibuat *dial*. *Dial* berfungsi untuk mengubah sudut *servomotor* secara *manual* pada jendela *front panel*. Untuk membuat *dial*-nya, buka jendela *front panel* kemudian klik kanan pilih “*num ctrl*è *dial*”, klik kanan *icon dial* pilih *properties*, akan muncul jendela *properties* seperti pada gambar dibawah ini, masukkan *scale range* maksimum  $180^0$  (yang ditunjukkan oleh lingkaran merah pada gambar 3.30), dimasukkan harga  $180^0$  karena *servomotor* maksimal berputar  $180^0$ . Untuk beberapa kasus harga minimum dan maksimum bisa disesuaikan dengan besar gerakan dari *arm robot 5 dof*. Jendela *properties dial* bisa dilihat pada Gambar 3.30 di bawah ini.



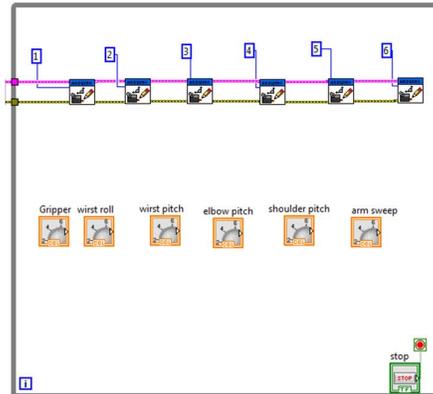
Gambar 3.30 Tampilan *scale* pada jendela *properties dial*

Apabila menginginkan setiap perpindahan angka *dial* hanya berpindah tiap satu derajat saja maka langkahnya adalah dengan menampilkan jendela *properties dial* kemudian pilih *data entry*. Hilangkan centang pada kotak “*Use Default Limits*” setelah itu minimum dan maksimum isikan masing masing dengan  $-\text{Inf}$  dan  $\text{Inf}$  yang artinya sesuai dengan *scale* yang telah dimasukkan sebelumnya. Untuk *response to value outside limits*, pilih *coerce* untuk minimum dan maksimum, dan pilih *coerce to nearest* untuk *increment*, masukkan *increment* dengan nilai 1 yang artinya hanya perpindahan tiap satu derajat saja yang bisa dilakukan pada *dial*.



Gambar 3.31 Tampilan *data entry* pada jendela *properties dial*

7. Buka jendela *block diagram* dan letakkan *dial*-nya di dalam struktur *while loop*. Untuk lebih jelasnya dapat dilihat pada gambar 3.32. Dan masing-masing *dial* bisa diberi nama sesuai dengan konfigurasiinya.

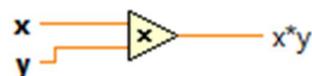


Gambar 3.32 *Dial* yang diletakkan ke dalam struktur *while loop*

8. Sudut yang dimasukkan pada LabVIEW tidaklah sama dengan posisi sudut pada *servomotor*. Dari percobaan pengukuran sudut pada *servomotor* didapatkan bahwa apabila memasukkan sudut  $166^0$  pada LabVIEW maka didapat sudut  $180^0$  pada *servomotor*. Dengan demikian diperlukan suatu koreksi sudut pada LabVIEW. Untuk mengoreksi sudutnya dapat dilakukan dengan menggunakan *numerical divided* dan *numerical multiply*. *Numerical divided* berfungsi untuk membagi input nilai  $x$  dan  $y$  yang dimasukkan. Sedangkan *numerical multiply* berfungsi untuk mengkalikan input nilai  $x$  dan  $y$  yang dimasukkan. Untuk memunculkan ikon *numerical divided* atau *numerical multiply* adalah dengan mengklik kanan pada jendela *block diagram* kemudian pilih “*mathematics è numeric è divide/multiply*”. Gambar 3.33 berikut ini adalah *numerical divided* dan gambar 3.34 adalah *numerical multiply*.



Gambar 3.33 *Numerical divided*

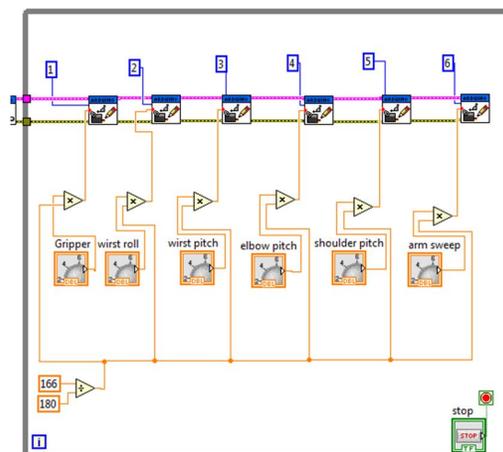


Gambar 3.34 *Numerical multiply*

9. Memberikan nilai x sebesar 166 dan nilai y sebesar 180 pada ikon *numerical divided*. Diberikan nilai tersebut untuk mencari kelebihan interval sudut yang terjadi antara sudut pada LabVIEW dengan sudut pada *servomotor*. Untuk memasukkan nilainya adalah dengan mengklik kanan kaki nilai x kemudian pilih “*create è constant*” dan masukkan nilai 166. Untuk *numerical multiply*, dengan cara yang sama dapat dilakukan untuk memasukkan nilai 180 pada kaki y sedangkan untuk nilai x dimasukkan output dari *numerical divided*. Diberikan nilai tersebut untuk koreksi sudut yang dipakai oleh *dial* dimana *input*-nya melalui hasil dari *numerical divided*. Setelah dilakukan percobaan kembali di pada saat di *input* 180<sup>0</sup>, pada *servomotor* dihasilkan sudut 183<sup>0</sup>, kemudian dapat dituliskan terdapat ketidaksesuaian sebesar 1,67%. Untuk lebih pemberian nilai x dan y pada *numerical divided* dapat dilihat pada gambar 3.35 berikut ini. Sedangkan pada gambar 3.36 adalah tampilan penghubung *input-output servo write angle*, *numerical divided*, *numerical multiply* dan *dial*.

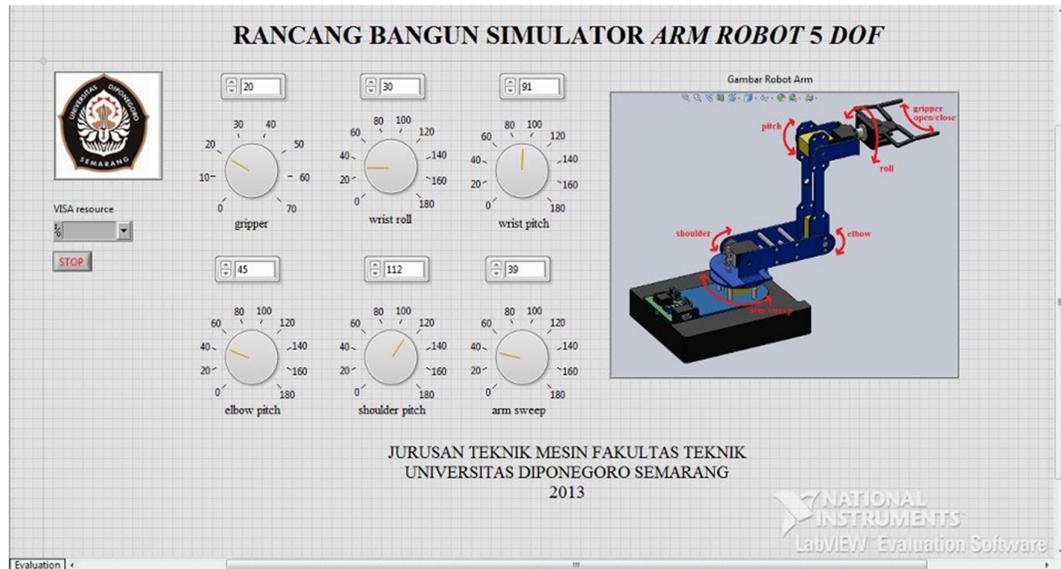


Gambar 3.35 Nilai kaki x dan y pada *numerical divided*



Gambar 3.36 Tampilan penghubung *input-output servo write angle*, *numerical divided*, *numerical multiply* dan *dial*



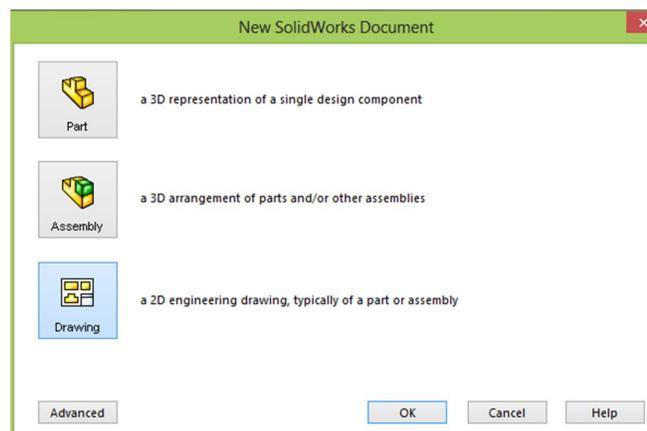


Gambar 3.39 Tampilan jendela *front panel*

### 3.4 Pembuatan Simulator *Arm Robot 5 DOF*

Setelah melakukan desain simulator, *hardware* dan *software* maka penulis melakukan merealisasikan hasil pendesainan tersebut. Hasil desain simulator *arm robot* yang berupa format SolidWorks harus diubah ke format CorelDraw 12. Tujuan perubahan format tersebut adalah agar hasil desain yang kita buat bisa terbaca pada mesin *laser acrylic cutting*. Berikut ini adalah langkah untuk mengubah format dari SolidWorks ke Coreldraw 12 :

1. Buka program SolidWorks, kemudian pilih “*new è drawing*”. Untuk lebih jelasnya bisa dilihat pada Gambar 3.40.



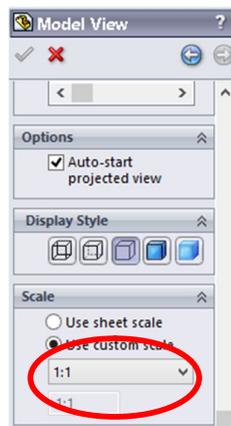
Gambar 3.40 Tampilan jendela *new* pada SolidWorks

2. Setelah memilih *drawing*, kemudian akan muncul jendela *sheet format/size*. Pilih A4-ISO *landscapes* untuk memilih lembar kerja. Setelah itu akan muncul *toolbox* disebelah kiri dengan nama *model view*. Pada pilihan *part/assembly to insert* (yang ditandai dengan lingkaran warna merah pada gambar 3.41) yaitu membuka *file* yang akan dipakai untuk dibuka. Sebagai contoh akan dibuka bagian *link 1*. Untuk lebih jelasnya dapat dilihat pada Gambar 3.41 berikut ini.

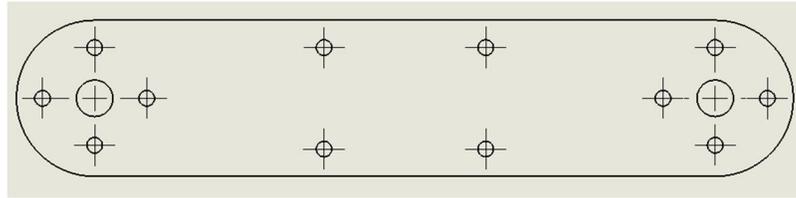


Gambar 3.41 *Toolbox model view*

3. Setelah memilih *link 1*, maka pada sebelah kiri ada terdapat *toolbox* dengan nama *model view*. Ubah skala menjadi 1:1 (yang ditunjukkan dengan lingkaran warna merah) pada gambar 3.42 dan pada gambar 3.38 adalah gambar *link 1* pada lembar kerja yang telah dipilih sebelumnya.

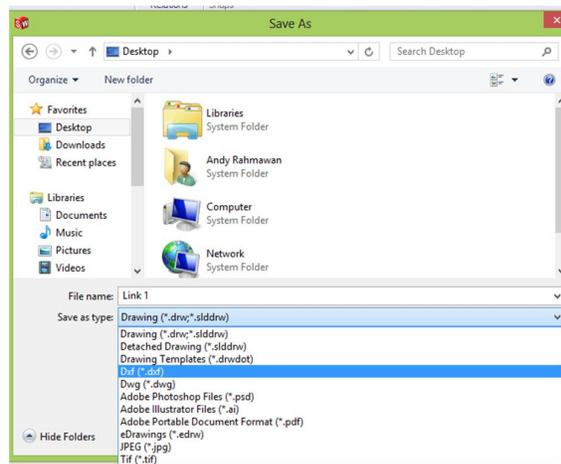


Gambar 3.42 Tampilan *model view* untuk memilih skala



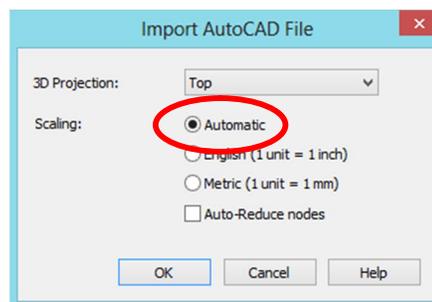
Gambar 3.43 *Link 1* pada lembar kerja

4. Setelah itu disimpan dengan format DXF. Pilih “*file è save as*”. Untuk lebih jelasnya dapat dilihat pada gambar 3.39 berikut ini.



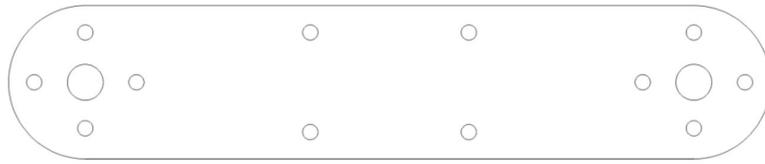
Gambar 3.44 *Save as* dengan format DXF

5. Kemudian membuka program CorelDraw 12, buka *file* dengan format DXF yang telah disimpan tadi. Kemudian akan muncul jendela *import AutoCAD file*, pilih *scaling automatic*. Untuk lebih jelasnya dapat dilihat lingkaran berwarna merah pada gambar 3.45 berikut ini.



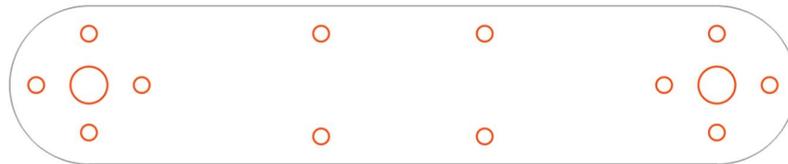
Gambar 3.45 Tampilan jendela *Import AutoCAD File*

6. Hilangkan bagian-bagian yang tidak perlu, seperti garis yang berada di dalam lingkaran. Kemudian hasilnya dapat dilihat pada gambar 3.41 berikut ini.

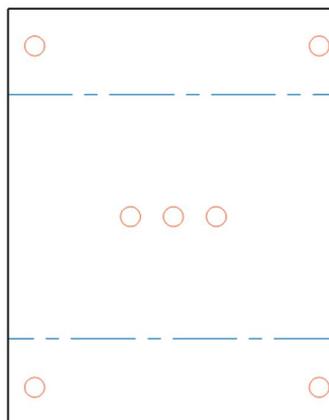


Gambar 3.46 Gambar desain *link 1* pada CorelDraw 12

7. Aturan untuk membuat lubang pada *link 1* yaitu garisnya harus berwarna merah, sedangkan untuk membuat *bending* menggunakan garis berwarna biru, dengan keseluruhan bentuknya harus dalam satu kurva dan keseluruhan garis atau lingkaran harus dalam satu grup. Untuk membentuk satu kurva yaitu dengan cara memblok semua garisnya lalu klik kanan pilih *combine*. Gambar 3.47 adalah hasil akhir desain *link 1* dan gambar 3.48 adalah *bracket* pada *gripper* yang menggunakan *bending*. Hasil akhir tersebut siap untuk dikirimkan ke jasa pemotongan akrilik yang menggunakan mesin *laser acrylic cutting*.

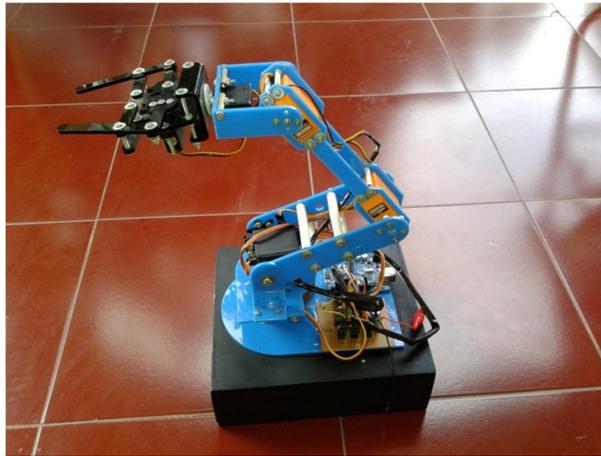


Gambar 3.47 Hasil akhir desain *link 1* pada CorelDraw 12



Gambar 3.48 Hasil akhir desain *bracket gripper* pada CorelDraw 12

Simulator *link* dan penyangga *arm robot 5 dof* dibuat menggunakan akrilik berwarna biru muda yang memiliki ketebalan 3 mm. Untuk *gripper* menggunakan akrilik berwarna hitam dengan ketebalan 3 mm. Gambar 3.49 berikut menunjukkan hasil desain dari *frame arm robot 5 dof* yang sudah dirakit (*assembly*) dengan *hardware* lain.



Gambar 3.49 Simulator *arm robot 5 dof*



Gambar 3.50 Simulator *arm robot 5 dof* menjepit kardus

Material akrilik dipilih karena memiliki kelebihan. Akrilik adalah semacam plastik yang menyerupai kaca, namun memiliki sifat yang membuatnya lebih unggul daripada kaca sehingga lebih tahan dan tidak pecah sehingga lebih aman. Keuntungan lain darinya adalah akrilik jauh lebih ringan dari pada kaca. Hal ini membuat bekerja dengan akrilik lebih mudah sehingga digunakan untuk pembuatan simulator *arm robot 5 dof*.

Untuk pembuatan *base* utama terbuat dari kayu sengon dengan tebal 2 cm. Kayu sengon dipilih karena memiliki beberapa kelebihan diantaranya adalah tidak mudah bengkok, tidak berbekas cabang, tidak mudah lapuk, tidak mudah busuk, dan juga tidak mudah pecah. Kayu tersebut dipotong sesuai dengan desainnya dan kemudian dicat dengan warna hitam.