

# PENGENDALIAN TANGAN ROBOT PERAKIT PADA DISAIN COMPUTER INTEGRATED MANUFACTURING BERBIAYA RENDAH (CIM) MENGGUNAKAN VISUAL BASIC DAN ARDUINO

Donny Montreano<sup>1</sup>, Sigit Pradana<sup>2</sup>

Program Studi Teknik Industri, Universitas Pembangunan Nasional "Veteran" Jakarta, Jakarta Selatan<sup>1</sup>

Program Studi Teknik Mesin, Universitas Pembangunan Nasional "Veteran" Jakarta, Jakarta Selatan<sup>2</sup>

email<sup>1</sup> : montreano@gmail.ac.id

---

## Abstract

*Computer Integrated Manufacturing (CIM) is a joint fabrication technology whose process units are controlled by one computer. Intelitek is well-known CIM brand and even though it is very expensive, it is still a reference for learning in manufacturing, especially industrial engineering because of the sophistication of simulation software and website-based lab modules. The long-term goal of this research is to build a low-cost alternative CIM, without including simulation software, using the 32-bit Arduino Due microcontroller to create competing products which, although very far cheaper than the Intelitek brand, can still deliver the essence of CIM learning at the basic level. This study intends to build 4 Robotic Arm for assembly stations which were traditionally only for material handling needs or with the term loading unloading. The objects assembled in this study are one Tetris 4 stack beam. The results showed a 50% failure due to the construction of the Robotic Arm assembly that matched between the components was too loose even though it had been tightened.*

*Keywords: CIM, Material Handling, Robot, Automation, design*

---

## PENDAHULUAN

Program studi teknik industri (progdi TI) UPNVJ berdiri tahun 2000. Sejak saat itu sampai sekarang mahasiswa progdi TI tidak pernah merasakan pengalaman merancang sistem manufaktur berbasis *automation*. Namun mereka mendapatkan pengetahuannya hanya di dalam perkuliahan.

Pengetahuan *automation* di perkuliahan biasa, belumlah cukup lengkap jika tidak diwujudkan dalam pengalaman nyata, karena di dalam perkuliahan hanya terdapat gambar dan simbol-simbol *automation*. Perkuliahan yang penuh dengan bahasan-bahasan *hardskill* seperti las, menggambar, pemesinan, perakitan dan sebagainya, sangat disarankan untuk menyajikan materi belajar secara fisik agar mahasiswa terbantu dalam hal imajinasi, pemahaman, dan analisisnya, meskipun hanya 4 atau 5 pertemuan dari 14 pertemuan.



Gambar 1 FMS Intelitek, USA

Menyajikan materi industri *automation* (otomasi), tentunya memiliki beberapa kendala yang harus diselesaikan, diantaranya adalah ruang dan waktu. Menyajikan materi tersebut dalam 4 atau 5 pertemuan biasanya hanya menyentuh kulit ilmu saja. Kemudian ruangan kelas perkuliahan yang ada, tidak memungkinkan untuk dapat mencapai mahasiswa yang duduk di belakang, kecuali materi fisik yang disajikan tersebut relatif kecil sehingga bisa dibawa-bawa oleh instruktornya sampai belakang.

Gambar 1 menjelaskan tentang industri otomasi, merek intelitek, dalam skala tertentu yang membutuhkan ruang relatif besar agar dapat memuat sebanyak-banyaknya mahasiswa tapi tetap dapat ikut mengalami semua proses tersebut. Menghadirkan industri otomasi skala besar kedalam praktikum tentu membutuhkan biaya yang sangat besar. Pada gambar tersebut hanya dapat melibatkan 10 mahasiswa dengan harga barang sebesar Rp 12 Milyar. Meskipun pembeliannya dibiayai negara, namun dengan adanya rancangan sendiri maka negara dapat menghemat devisa nyaris mencapai \$ 1 juta.

## Rumusan Masalah

Kapasitas 10 mahasiswa dalam praktikum ini dari 60 mahasiswa seangkatan, akan menjadi 10 kelompok yang tiap-tiapnya menghabiskan waktu 2.5 jam sehingga total 25 jam praktikum dalam 1 minggu. Hal ini akan bertentangan dengan beban kerja normal seorang dosen yaitu 18 jam mengajar per minggu. Kelebihan jam masih belum bisa dibebankan kepada dosen lain karena fakultas teknik kekurangan tenaga pendidik yang kompeten dibidang ini.

Melihat gambar 1, kapasitas dapat ditambah menjadi 20 mahasiswa, namun menjadi sulit ketika harus bongkar pasang. Untuk itu harus mengundang teknisi dari Amerika yang tentunya memakan biaya puluhan juta rupiah. Dan lagi import barang untuk perluasan juga menambah biaya yang tidak sedikit.

Kendala ruang dan waktu inilah yang akan dipecahkan oleh peralatan, yang dirancang sendiri, yang mengadopsi sistem *Automated Guided Vehicle (AGV)* oleh *Indian Institute of Technology Kharagpur* seperti gambar 2 di bawah. Meskipun hanya 3 stasiun, namun berkat keluwesan AGV maka FMS itu dapat ditambahi 10 stasiun atau lebih dan jalur AGV dapat diperluas dengan bentuk yang berkelok-kelok tanpa harus bongkar pasang seperti telah dijelaskan di atas.



Gambar 2 CIM dengan *Automated Guided Vehicle (AGV)*

## Tujuan Penelitian

Membuat Robotic Arm dan program pengendaliannya untuk membangun peralatan CIM

## TINJAUAN PUSTAKA

### Computer Integrated Manufacturing

Computer integrated manufacturing (CIM) adalah pendekatan manufaktur dengan menggunakan komputer untuk mengendalikan seluruh proses produksi (Kalpakjian, 2006) (Laplante, 2005). Integrasi ini memungkinkan proses tunggal untuk bertukar informasi dengan lainnya dan mengawasi tindakan. Juga manufaktur dapat lebih cepat dan sedikit kesalahan oleh tertintegrasinya komputer-komputer, keuntungan utamanya adalah kemampuan untuk membuat proses manufaktur menjadi otomatis. CIM pada umumnya,

mengandalkan pada proses *closed-loop*, berdasar pada masukan data real-time pada sensor-sensor. CIM juga diketahui sebagai flexible design dan manufacturing (Grover, 1980)

### Flexible Manufacturing System

Sistem manufaktur fleksibel atau FMS (Flexible Manufacturing System) adalah sistem manufaktur yang dapat bereaksi secara fleksibel terhadap perubahan-perubahan. Dua macam perubahan sistem itu dapat berupa perubahan tipe produk yang akan dihasilkan (*machine flexibility*), maupun perubahan urutan proses dalam pembuatan produk tersebut (*routing flexibility*). Keuntungan dari penggunaan FMS dalam suatu sistem produksi massal (*mass production*) adalah kemampuan fleksibilitasnya yang tinggi baik dalam mengalokasikan waktu dan usaha, sehingga dapat menaikkan produktifitas dan mutu produk serta menurunkan biaya produksi.

Alasan mengapa FMS disebut flexible adalah bahwa kemampuannya untuk memproduksi bermacam komponen dan tampilan komponen secara simultan pada workstation yang berbeda, dan jumlah produksi dapat diatur sesuai pola permintaan yang ada. (Groover, 1987)

Kebanyakan sistem FMS terdiri dari 3 bagian, yaitu sebuah sistem mesin CNC yang ter-otomasi, satu grup mesin produksi (*material handling system*) dan robot, serta satu set komputer sentral (termasuk di dalamnya alat-alat elektronik instrumentasi industri/pabrik, alat pengukuran, dan sensor). Melalui jaringan komputer pabrik yang mempunyai ciri tersendiri daripada kebanyakan jaringan komputer perkantoran, semua peralatan dalam FMS ini dapat dikendalikan dan dapat saling berkomunikasi satu sama lain.



Gambar 3 FMS dengan sebuah robot, mesin CNC-Mill dan CNC-Lathe

Ciri khas dari jaringan komputer pabrik adalah tingginya gangguan (*noise*) akibat panas, adanya debu dan kelembaban yang tinggi, yang menyebabkan jaringan komputer sering gagal. Selain itu, beberapa kegiatan mesin dan robot dapat mengakibatkan keadaan yang berbahaya dan perlu penanganan dengan cepat dan darurat. Oleh karena itu jenis jaringan komputer lokal (LAN), seperti

CSMA/CD dan Token Bus standar tidak bisa dipakai. Untuk dapat menangani paket data yang urgen dan bersifat segera, jaringan komputer itu harus dapat memberikan prioritas pengiriman berita. Dalam hal ini, berita urgen dapat meng-interrupt pengiriman data biasa, seperti yang dimiliki oleh jaringan Token ring dan Token Bus termodifikasi.

## Arduino

Arduino merupakan suatu istilah yang banyak mengandung makna di antaranya: sebuah perangkat keras komputer *open-source* dan *software* namun sekaligus juga mewakili perusahaan, proyek dan komunitas pengguna. Komunitas ini yang mendesain dan memproduksi kit berbasis mikrokontroler untuk membangun perangkat digital dan objek interaktif yang dapat merasakan dan mengontrol dunia fisik. (arduino.cc)

Arduino adalah suatu alat elektronik yang memproses masukan menjadi suatu reaksi tertentu seperti menghidupkan lampu, membuat teks berjalan, menggerakkan motor listrik dan sebagainya.



**Gambar 4** Contoh Perangkat Keras Arduino Uno

Masukan yang dibutuhkan dalam percobaan ini hanya ada satu jenis yaitu jarak yang diperoleh dari sensor parallax yaitu jarak antara alat ini dengan penghalang di depannya. Pada jarak tertentu AGV ini diprogram untuk mengikuti garis berbentuk sirkuit. Untuk pemrograman mikrokontroler, platform Arduino menyediakan lingkungan pengembangan terpadu (IDE) berdasarkan proyek Pengolahan, yang meliputi dukungan untuk pemrograman C, C++ dan Java bahasa.

## Bahasa C++

Program Arduino ditulis dalam C atau C++, Arduino IDE datang dengan kumpulan kode perangkat lunak yang disebut "Wiring" dari proyek Wiring asli, yang membuat banyak operasi umum input / output jauh lebih mudah. Para pengguna hanya perlu mendefinisikan dua fungsi untuk membuat program eksekutif siklik dieksekusi: `setup ()`: fungsi yang berjalan sekali pada awal program dan yang dapat menginisialisasi pengaturan.

`loop ()`: fungsi yang dilaksanakan berulang-ulang sampai papan itu tidak menyala lagi.

Sebuah program pertama yang khas mikrokontroler adalah hanya untuk membuat LED berkedip dan mati. Dalam lingkungan Arduino, pengguna mungkin menulis sebuah program seperti ini:

```
#define LED_PIN 13

void setup() {
    pinMode(LED_PIN, OUTPUT);
    // Enable pin 13 for digital
    output
}
```

**Gambar 5** Contoh Sederhana Pemrograman Arduino

Pemrograman di atas dapat diterjemahkan sebagai berikut:

Menugaskan pin ke 13 untuk keluaran digital dan memberi nama pin ke 13 dengan nama LED\_PIN. Ketika program sudah berjalan, LED\_PIN diperintahkan untuk mengalirkan listrik yang menghasilkan lampu bersinar selama 1000 mili detik. Setelah bersinar selama itu, LED\_PIN diperintahkan untuk memutuskan aliran listrik selama 1000 milidetik, akibatnya lampu mati selama itu dan akan hidup kembali dengan selang yang sama di awal dan seterusnya.

## Penelitian Terdahulu

Arduino merupakan mikrokontroler yang murah namun memiliki kemampuan yang memadai untuk mengendalikan banyak aktuator. Penelitian yang menggunakan arduino sebagai pusat kendali machining atau kendali tangan-tangan robot untuk industri, sangat jarang ditemui. Berikut ini adalah penelitian tentang FMS yang berhubungan dengan arduino.

Vasile Alecsandri, 2015, menggunakan arduino pada sistem FMS hanya untuk identifikasi komponen, penentuan posisi, mengarahkan pallet dan menjamin kelancaran aliran komponen apabila terjadi keadaan seperti kegagalan machining center. Bukan menggunakan arduino sebagai pusat machining dan material handling.

V. Barenji dan C. Degirmenci, 2015, melakukan penelitian tentang tangan robot yang dikendalikan jarak jauh melewati internet dan teknologi RFID untuk membedakan posisi benda-benda yang dipindahkan tangan robot. Tangan robot bertugas untuk memindahkan barang ke suatu tempat yang diinginkan. Namun untuk menjamin bahwa barang tersebut berada pada tujuan yang tepat maka RFID perlu disematkan pada tiap barang dan pemasangan receiver pada penadah barang. Penelitian tersebut menggunakan hanya satu tangan

robot dan menggunakan Arduino UNO untuk mengendalikannya.

## METODE PELAKSANAAN

### Enam Fase

Secara keseluruhan tahap pengembangan produk ini terdiri atas 6 fase. Berikut ini penjelasan fase secara berurut:

#### 1. Perencanaan

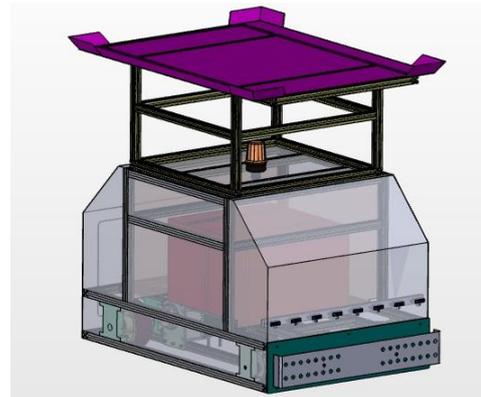
Awal pelaksanaan kegiatan ini dimulai mengumpulkan segala informasi mengenai FMS, referensi desain peralatan FMS dan studi literatur. Studi literatur dapat berupa buku teks, jurnal, paper, dokumentasi manufaktur dan modul praktikum. Dari seluruh sumber literatur dipelajari sehingga penelitian ini dapat berjalan sesuai dengan yang direncanakan. Peralatan dikerjakan secara parallel dengan pemrograman komputer.

#### 2. Desain Platform dengan menggunakan *software*

Desain alat FMS menggunakan *software Computer Aided Design (CAD) 3 Dimensi*. Dengan adanya tahap ini, fabrikasi dapat dipermudah dengan modifikasi part yang mudah dibentuk. Beberapa komponen yang perlu diperhatikan adalah:

- Motor Penggerak
- Peletakan komponen elektronik
- Perencanaan Mur-Baut
- Ukuran akhir peralatan pembersih lantai otomatis

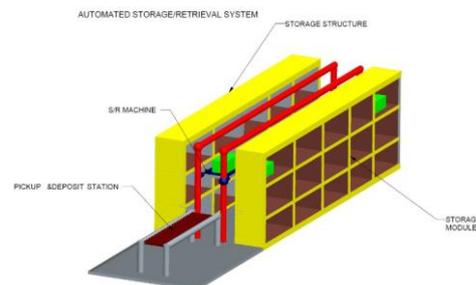
Gambar – gambar di bawah ini adalah desain awal peralatan AGV, *Robotic Arms (RbA)*, *Automated Storage and Retrieval System (ASRS)*. Desain tersebut akan selalu direvisi karena menyesuaikan dengan kebutuhan, misalnya perubahan ukuran komponen, perubahan bentuk komponen untuk menambah kekokohan alat, penambahan komponen elektronik dan lain sebagainya. AGV, RbA dan ASRS, sebelum beroperasi hanya di *setup* oleh beberapa *engineer* selama beberapa menit untuk beroperasi selama waktu yang telah ditentukan. Selibhnya selama proses produksi akan berjalan sendiri tanpa kendali oleh operator. Baik pemasukkan bahan mentah sampai bahan jadi akan disimpan ke gudang oleh ASRS tanpa kendali operator. Hanya AGV yang ditenagai baterai.



**Gambar 6** Desain AGV



**Gambar 7** Desain *Robotic Arms*



**Gambar 8** Desain Automated Storage and Retrieval

#### 3. Produksi

Tahap produksi meliputi pembelian komponen dan bahan mentah, fabrikasi dan perakitan. Peralatan produksi yang digunakan untuk aktifitas *drill*, *cut* dan *milling* menggunakan tiga CNC kecil yang dimiliki fakultas teknik.

#### 4. Perakitan dan Uji Fungsi

Setelah tiga peralatan selesai diproduksi, selanjutnya dirakit dan di uji fungsinya. Peralatan yang pertama kali di uji fungsi adalah AGV yaitu meliputi, uji komunikasi *wireless*, *line follower*, angkat beban, penyelesaian tugas (*rally point*), tombol darurat berhenti, dan lain lain. Kecuali ASRS dan RbA tidak mengalami uji *line follower*.

#### 5. Pengambilan Data

Pada tahap ini ada 3 data yang perlu diambil. Pertama, yaitu kecepatan semua peralatan dalam penyelesaian tugasnya. Kedua, data konsumsi Watt yang akan diambil ketika sedang menggunakan AGV, RbA dan ASRS. Dan yang terakhir adalah data termal operasional, tentang suhu maksimal yang terjadi ketika waktu produksi di percepat.

#### 6. Analisis hasil kerja

Pada tahap ini akan menjelaskan karakteristik masing-masing peralatan dalam hal waktu, energi dan suhu serta saran pemakaian.

#### Benda Kerja

Agar langkah pengambilan data dan analisis hasil kerja dapat dilakukan maka, diperlukan suatu objek benda kerja sebagai masukkan bahan mentah kepada sistem FMS ini sebagai percobaan produksi yang utuh.

Percobaan produksi dilakukan dalam urutan penuh yang tidak terputus agar dapat menyerupai proses produksi yang sebenarnya di industri-industri serupa di luar. Target uji coba tersebut bukanlah kecepatan produksi melainkan keberhasilan membuat 1 produk jadi tanpa terputus.

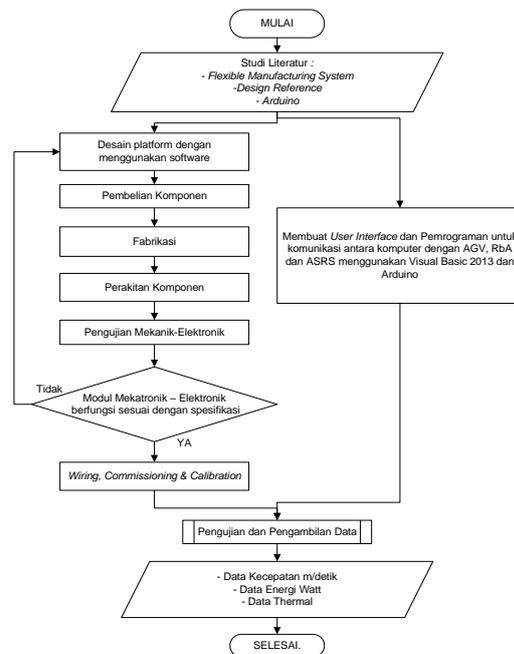
Kecepatan produksi dapat di atur pada sesi penelitian lanjutan dan dalam praktikum sistem produksi. Misalkan pada kondisi tertentu terdapat kendala bottleneck, sehingga kecepatan produksi di lini tertentu harus di perlambat agar tidak terjadi penumpukan.

Berikut ini adalah produk jadi yang dimaksud dalam uji coba produksi:



Gambar 9 Produk Jadi

Berikut ini adalah metode pelaksanaan yang digambarkan dalam diagram alir :



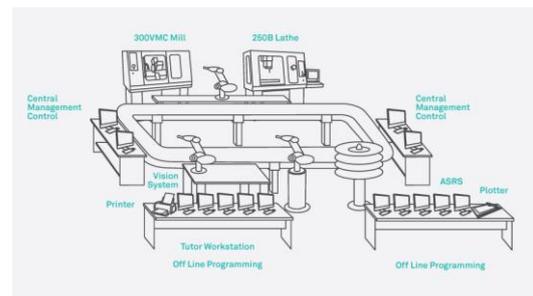
Gambar 10 Diagram Alir Penelitian

## HASIL DAN PEMBAHASAN

### Disain

Disain CIM pada umumnya terdiri dari 4 fasilitas yaitu fasilitas proses manufaktur contohnya mesin bubut, milling, dan perakitan, lalu fasilitas *material handling* contohnya konveyor ataupun semacam forklift, kemudian fasilitas penyimpanan contohnya gudang bahan baku sampai gudang barang jadi.

Fasilitas yang terakhir adalah pengendali berupa komputer PC yang dioperasikan oleh manusia. Berikut ini adalah skema CIM pada umumnya yang di ambil dari boxford.co.uk :

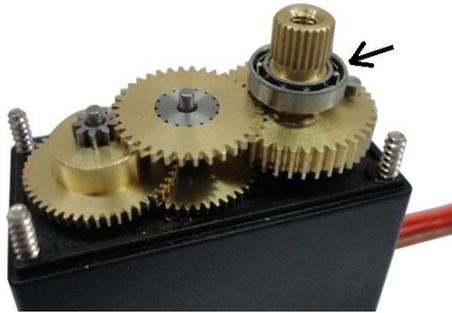


Gambar 11 Skema CIM pada umumnya

### Fabrikasi dan Perakitan

#### Fabrikasi

Secara umum fabrikasi sedikit dilakukan pada penelitian ini. Sebagian besar waktu digunakan untuk merakit tangan robot dan merakit fasilitas penyimpanan. Hanya ada 2 fabrikasi yang dilakukan yaitu, pemotongan alumunium profil untuk konstruksi penyimpanan, kemudian fabrikasi bearing seperti gambar di halaman berikut ini :



**Gambar 12** Komponen bearing pada motor servo

Gambar panah hitam di atas menunjuk pada bearing yang berwarna perak. Bearing tersebut berfungsi sebagai penstabil gerakan agar tidak miring atau tetap tegak lurus. Namun pada kenyataannya, servo yang digunakan berkualitas rendah, sehingga poros berputar miring apabila diberi beban.

Perputaran yang miring menyebabkan akurasi gerakan lengan robot menjadi sangat berkurang apalagi jika lengan robot memiliki panjang yang signifikan. Semakin panjang lengan robot maka semakin besar deviasi posisi tangan. Hal ini akan berpengaruh pada fasilitas *material handling* khususnya *loading / unloading* menjadi selalu meleset, maka benda bisa jatuh atau mesin bubut menjadi macet.

Deviasi posisi tangan terjadi karena bearing tersebut tidak memenuhi ruangan yang tersedia di dalam *casing* servo. Untuk itu perlu dibuatkan bearing yang memiliki akurasi tinggi. Yaitu diameter luar bearing diperbesar agar memenuhi ruang yang telah disediakan di dalam *casing* servo. Bearing tersebut berbahan akrilik agar mudah difabrikasi di dalam laboratorium sistem produksi.

### Perakitan

Perakitan dilakukan pada lengan robot dan fasilitas penyimpanan. Di halaman berikut ini gambar sebelum dan sesudah perakitan tangan robot.

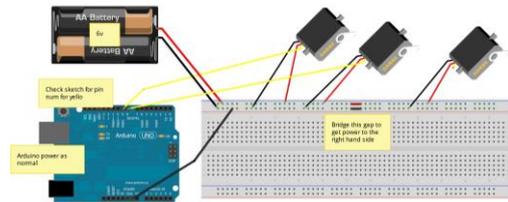


**Gambar 13** Komponen Tangan Robot

### Penyambungan Kabel

Secara umum kabel servo terdiri dari 3

warna yaitu hitam, merah dan kuning atau *orange*. Hitam (atau coklat) dan merah disambung ke negatif dan positif. Kabel kuning di sambung ke salah satu pin Arduino sesuai dengan kode di bab V.4. Pada dasarnya cara penyambungannya seperti gambar di bawah ini :



**Gambar 14** Wiring Diagram



**Gambar 15** Perakitan 100%

Apabila ada penambahan servo, maka kabel hitam dan merah hanya disambungkan secara paralel disebelah kanan atau sebelah kiri servo yang sudah ada sebelumnya. Kabel kuning disambung ke pin digital arduino yang menganggur.

Sumber tenaga pada gambar di atas, di ganti dengan aki 6V 8A karena menyesuaikan spesifikasi servo yang membutuhkan tegangan kerja sekitar 6-7.2V dan membutuhkan arus 0.8A ketika servo dibebani suatu benda misalkan tangan itu sendiri yang menjadi beban. Aki perlu memiliki ampere yang besar karena satu tangan terdiri dari 6 servo yang artinya membutuhkan total  $0.8A \times 6 = 4.8A$ . Jika menggunakan aki yang memiliki ampere kecil maka, tangan robot tidak mampu mengangkat dirinya sendiri.

### Pemrograman Arduino

Tangan robot menggunakan 6 DoF servo sehingga memiliki gerakan di 3 sumbu sekaligus. Tangan tersebut dapat berfungsi sebagai *material handling*, biasanya untuk *loading / unloading* benda kerja pada mesin bubut ataupun mesin milling seperti pada gambar 11. Juga dapat bekerja

sebagai perakitan dan proses cat pada pabrik mobil. Di bawah ini adalah pemrograman arduino untuk 5 DoF dari 6 DoF

```
#include <Servo.h>
#include <Scheduler.h>
Servo myservo1, myservo2, myservo3, myservo4,
myservo5;

String inputString = "";
String DOF01 = "";
String DOF02 = "";
String DOF03 = "";
String DOF04 = "";
String DOF05 = "";
bool stringComplete1, stringComplete2,
stringComplete3, stringComplete4, stringComplete5,
stringComplete10 = false;

void setup() {
myservo1.attach(2);
myservo1.write(90);
myservo2.attach(3);
myservo2.write(90);
myservo3.attach(4);
myservo3.write(110);
myservo4.attach(5);
myservo4.write(90);
myservo5.attach(6);
myservo5.write(90);
inputString.reserve(20);
DOF01.reserve(10);
DOF02.reserve(10);
DOF03.reserve(10);
DOF04.reserve(10);
DOF05.reserve(10);

Scheduler.startLoop(loopDof2);
Scheduler.startLoop(loopDof3);
Scheduler.startLoop(loopDof4);
Scheduler.startLoop(loopDof5);
Serial.begin(115200);
}

void loop() {
if (stringComplete1) {
Serial.println(inputString);
char a1 = inputString.charAt(0);
char a2 = inputString.charAt(1);
char a3 = inputString.charAt(2);

DOF01 = DOF01 + a1;
DOF01 = DOF01 + a2;
DOF01 = DOF01 + a3;

myservo1.write(DOF01.toInt());
Serial.println("ini adalah nilai DOF01= " + DOF01);
DOF01 = "";
stringComplete1 = false;
yield();
}
}
```

```
}

void loopDof2() {
if (stringComplete2) {

char b1 = inputString.charAt(3);
char b2 = inputString.charAt(4);
char b3 = inputString.charAt(5);

DOF02 = DOF02 + b1;
DOF02 = DOF02 + b2;
DOF02 = DOF02 + b3;

myservo2.write(DOF02.toInt());
Serial.println("ini adalah nilai DOF02= " + DOF02);
DOF02 = "";
stringComplete2 = false;
inputString = "";
yield();
}
}

void loopDof3() {
if (stringComplete3) {
char c1 = inputString.charAt(6);
char c2 = inputString.charAt(7);
char c3 = inputString.charAt(8);
DOF03 = DOF03 + c1;
DOF03 = DOF03 + c2;
DOF03 = DOF03 + c3;
myservo3.write(DOF03.toInt());
Serial.println("ini adalah nilai DOF03= " + DOF03);
DOF03 = "";
stringComplete3 = false;
yield();
}
}

void loopDof4() {
if (stringComplete4) {

char d1 = inputString.charAt(9);
char d2 = inputString.charAt(10);
char d3 = inputString.charAt(11);

DOF04 = DOF04 + d1;
DOF04 = DOF04 + d2;
DOF04 = DOF04 + d3;

myservo4.write(DOF04.toInt());
Serial.println("ini adalah nilai DOF04= " + DOF04);
DOF04 = "";
stringComplete4 = false;
yield();
}
}

void loopDof5() {
if (stringComplete5) {
```

```
char e1 = inputString.charAt(12);
char e2 = inputString.charAt(13);
char e3 = inputString.charAt(14);
```

```
DOF05 = DOF05 + e1;
DOF05 = DOF05 + e2;
DOF05 = DOF05 + e3;
```

```
myservo5.write(DOF05.toInt());
Serial.println("ini adalah nilai DOF05 = " + DOF05);
DOF05 = "";
stringComplete5 = false;
yield();
```

```
}
}

void serialEvent() {
while (Serial.available()){
char inChar = (char)Serial.read();
inputString = inputString + inChar;
if (inChar == '_') {stringComplete1 = true;
stringComplete2 = true; stringComplete3 = true;
stringComplete4 = true; stringComplete5 = true;
stringComplete10 = true;}
}
}
```

Penjelasan:

1. serialEvent adalah fungsi yang memungkinkan fungsi berjalan bersamaan dengan fungsi void Loop.
2. serialEvent memungkinkan manusia mengendalikan perintah yang ada di dalam fungsi Loop dengan cara ketik segala character sampai ada character \_ (underscore) maka stringComplete1 bernilai true, dan mengakibatkan fungsi void loop bekerja

Setelah membuat kode di Arduino IDE, selanjutnya *upload* kode tersebut ke *MotherBoard* Arduino dengan tombol *upload*. Lalu coba menggerakkan servo dengan memasukkan angka 3 digit di Serial Monitor, misalkan 45 derajat, dengan penulisan seperti ini: 045\_

Untuk derajat di atas 99, maka tidak perlu menambahkan angka 0 di depan, misalkan 180 derajat, tulis seperti berikut: 180\_

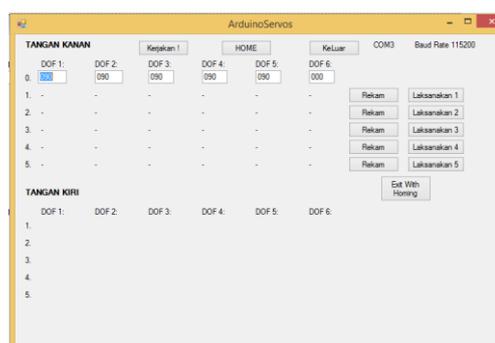
Menurut *manual instruction* yang beredar di internet, Servo tersebut berputar mulai dari 0 sampai 120, namun ketika dicoba angka 185, 184, 183... 180, tidak bergerak. Servo berputar ketika dimasukkan angka 179. Artinya servo memiliki batas atas 180 bukan 120.

Tahap selanjutnya kita menyiapkan UI dengan visual basic .Net

## Pemrograman Visual Studio

Pada bab ini kita membahas tentang program yang mencoba untuk menghubungi *MotherBoard* Arduino melalui jalur *Serial*. Mengapa kita coba menghubungi Arduino? Penelitian sebelumnya kita mengendalikan Servo dengan aplikasi yang sudah tersedia di Arduino IDE, sekarang kita coba dengan aplikasi Visual Basic .Net.

Mengapa Visual Basic Net? pada dasarnya bahasa apapun dapat saja mengendalikan Servo, semisal Java™, C#, C++, Phyton, Linux dan lain lain. Yang terpenting bagaimana cara mengendalikan semua fasilitas



Gambar 16 Disain Tampilan Aplikasi

(yang telah disebut dalam definisi CIM) dengan aplikasi berbasis PC. Visual Basic .Net merupakan aplikasi berbasis PC yang telah teruji kompatibilitasnya. Gambar 16 adalah desain tampilan untuk mengendalikan 6 Servo.

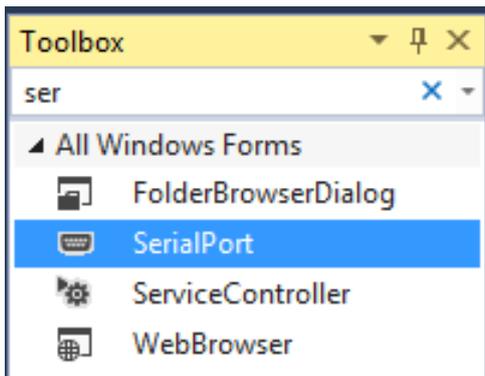
Textbox berfungsi untuk menampung masukkan angka dari operator CIM. Pengembangan selanjutnya akan ada 6 textbox untuk Tangan Kiri karena CIM akan memiliki 6 tangan robot diantaranya untuk *material handling* dan perakitan.

Disisi kiri form terdapat 6 baris dimulai dari 0 sampai 5. Artinya, servo akan bergerak 6 kali berturut-turut mengikuti nilai 0 sampai 5. Pergerakan per baris servo tergantung dari tombol Laksanakan di sisi kanannya, tentu setelah menekan tombol Rekam. Apabila ingin bergerak 3 kali berturut-turut maka operator harus menekan tombol Laksanakan di masing-masing baris, satu per satu. Dalam pengembangan jangka panjangnya, form akan menyediakan variasi gerakan sampai 20 baris atau lebih menggunakan *Data Sheet*.

Ketika tombol Laksanakan ditekan maka komputer mengirim nilai yang ada di TextBox1 ke Arduino melalui sinyal SERIAL. Untuk itu kita perlu menulis program di tombol Laksanakan sebagai berikut:

```
Private Sub Button1_Click(sender As Object, e As EventArgs) Handles Button1.Click
SerialPort1.Write(TextBox1.Text)
```

```
SerialPort1.Write("_")
End Sub
```



**Gambar 17** SerialPort

Namun tentunya Visual Basic .Net tidak mengenali *SerialPort1*, karena kita belum menyiapkan lewat *ToolBox*. Untuk itu klik 2x *SerialPort* seperti gambar 16 di atas ini:

Kemudian tuliskan kode untuk menyiapkan segala parameter *SerialPort1* seperti dibawah ini dengan sebelumnya klik 2x *Form* tersebut:

```
Private Sub Form1_Load(sender As Object, e As EventArgs) Handles MyBase.Load
```

```
Windows.Forms.Control.CheckForIllegalCrossThreadCalls = False
    SerialPort1.Close()
    SerialPort1.PortName = "COM3"
    SerialPort1.BaudRate = 115200
    SerialPort1.DataBits = 8
    SerialPort1.Parity = Parity.None
    SerialPort1.StopBits =
StopBits.One
    SerialPort1.Handshake =
Handshake.None
    SerialPort1.Encoding =
System.Text.Encoding.Default
    SerialPort1.Open()
End Sub
```

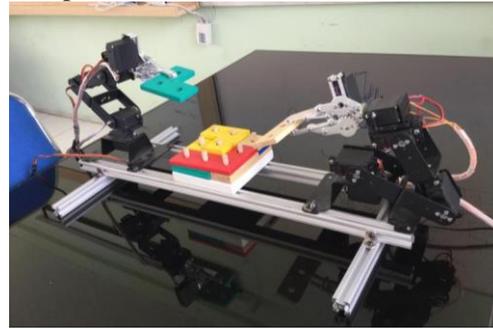
Yang perlu diperhatikan :

1. Beri saja nama aplikasi ini *ArduinoServo*
2. *SerialPort1.PortName = "COM3"*, nama COM3 dapat di ubah tergantung dari pendeteksian komputer terhadap *MotherBoard* *Arduino*
3. Lihatlah di *Device Manager* untuk melihat COM berapa *MotherBoard* *Arduino* dapat jatah.
4. Beda *MotherBoard* maka beda COM-nya meskipun sama sama *Arduino Due*

### Hasil Perakitan

Berikut ini adalah gambar hasil sementara

dari penelitian ini :



**Gambar 17** Robot Perakit

Secara umum gambar di atas menunjukkan fasilitas perakitan atau *assemblying*. Masih ada 1 fasilitas lagi yang perlu dibangun yaitu penyimpanan. Mengenai fasilitas *Control* (untuk operator CIM) tidak diperlukan karena sistem CIM ini akan dirancang jarak jauh (Revolusi Industri 4.0) sehingga bukan keharusan untuk menyediakan PC di lokasi.

### KESIMPULAN DAN SARAN

Pemrograman telah sukses mengendalikan fasilitas perakitan, namun dari 20 kali percobaan merakit menunjukkan terjadi 50% kegagalan rakit yang diakibatkan oleh konstruksi *Robotic Arm* dimana suaian antar komponennya terlalu longgar meski sudah dikencangkan. Berikut ini adalah link untuk demonstrasi perakitan <https://www.youtube.com/watch?v=o3Pq7JP4q7Q>

### DAFTAR PUSTAKA

- "[Arduino](#) - [Introduction](#)", <http://arduino.cc/en/guide/introduction>, diakses tanggal 1 Maret 2016.
- A. V. Barenji and C. Degirmenci, (2013) "*Robot Control System based on Web Application and RFID Technology*", Eastern Mediterranean University (EMU)-Doğu Akdeniz Üniversitesi (DAÜ).
- A. de Toni and S. Tonchia (1998) "*Manufacturing Flexibility: a literature review*", *International Journal of Production Research*, vol. 36, no. 6, 1587-617.
- Alecsandri V., (2015) "*Modeling and Analysis of Processing Flow in a Flexible Manufacturing System using Synchronized Petri Nets*", *Trans Tech Publications*, Switzerland, pp 1474-1479
- G. Chrysolouris (2005), "*Manufacturing Systems – Theory and Practice*", Springer Verlag, 2nd edition, New York, NY.
- T. Tolio, (2009), "*Design of Flexible Production Systems – Methodologies and Tools*", Springer, [ISBN 978-3-540-85413-5](#), Berlin

- Groover, Mikell P (2000),” *Automation, Production Systems, and Computer-Integrated Manufacturing*”, 2nd Edition, Prentice Hall
- Johnson, C. D. (2014). *Process Control Instrumentation Technology* Eight edition. Pearson: Edinburgh.
- Kalpakjian, Serope; Schmid, Steven (2006), “*Manufacturing Engineering and Technology*”, 5th Edition, Prentice Hall, p. 1192, ISBN 978-7-302-12535-8.
- Ortiz, A., and Etter, G. (1990), “*Simulation Modeling vs. Queuing Theory*” in Proceedings of the 1990 Annual Healthcare Information and Management Systems Conference, AHA, pp. 349–357.
- Prassler E., Ritter A., Schaeffer C., & Fiorini P, (2000), "Autonomous Robots" Volume 9, Issue 3, pp 211-226
- Xueshan G., Yan W., Dawei Z., & Koki K., (2009) "Floor-Cleaning Robot Using Omni-directional Wheels", *Industrial Robot: An International Journal*, Vol. 36 Iss: 2, pp.157 – 164
- Y. Korem, (1983) ”*Computer Control of Manufacturing Systems*”, McGraw Hill, Inc., 287 pp, [ISBN 0-07-035341-7](#)