



**RANCANG BANGUN SISTEM DETEKSI KERUSAKAN  
POLA BATIK MENGGUNAKAN METODE *FIRST ORDER***



INSTITUT BISNIS  
DAN INFORMATIKA

stikom  
SURABAYA

Oleh:

**REVIDHO YUSUF DEMANIQGA**

**15410200051**

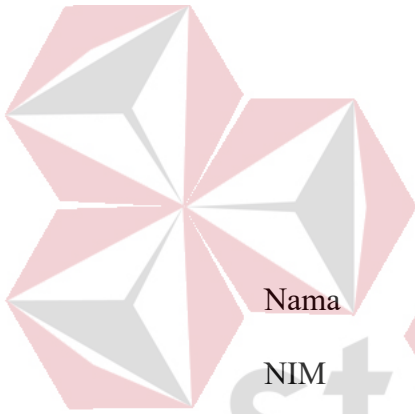
---

**FAKULTAS TEKNOLOGI DAN INFORMATIKA  
INSTITUT BISNIS DAN INFORMATIKA STIKOM SURABAYA  
2019**

**RANCANG BANGUN SISTEM DETEKSI KERUSAKAN POLA BATIK  
MENGUNAKAN METODE *FIRST ORDER***

**TUGAS AKHIR**

Diajukan sebagai salah satu syarat untuk menyelesaikan  
Program Sarjana Teknik



**Disusun Oleh:**

Nama : Revidho Yusuf Demaniqga

NIM : 15410200051

Program : S1 (Strata Satu)

Jurusan : Teknik Komputer

Fakultas : Teknologi dan Informatika

**FAKULTAS TEKNOLOGI DAN INFORMATIKA  
INSTITUT BISNIS DAN INFORMATIKA STIKOM SURABAYA**

**2019**



**"Ubah Pikiranmu dan Kau Dapat Mengubah  
Duniamu"**

INSTITUT BISNIS  
DAN INFORMATIKA

**stikom**

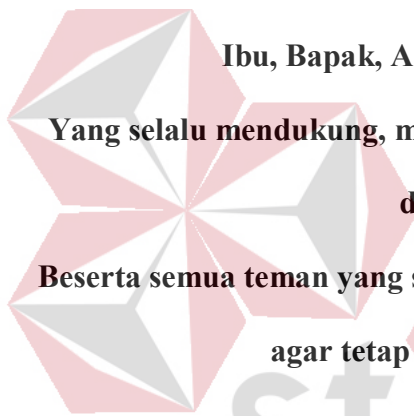
SURABAYA

**Kupersembahkan Kepada ALLAH SWT**

**Ibu, Bapak, Adik dan semua keluarga tercinta,**

**Yang selalu mendukung, memotivasi dan menyisipkan nama saya dalam  
doa-doa terbaiknya.**

**Beserta semua teman yang selalu membantu, mendukung dan memotivasi  
agar tetap berusaha menjadi lebih baik.**



**stikom**  
SURABAYA

## TUGAS AKHIR

### RANCANG BANGUN SISTEM DETEKSI KERUSAKAN POLA BATIK MENGUNAKAN METODE *FIRST ORDER*

Dipersiapkan dan disusun oleh  
**REVIDHO YUSUF DEMANIQGA**  
NIM : 15410200051

Telah diperiksa, diuji dan disetujui oleh Dewan Pembahas

Pada : Juli 2019

#### Susunan Dewan Pembimbing dan Penguji

##### **Pembimbing**

I. Heri Pratikno, M.T., MTCNA., MTCRE.

NIDN. 0716117302

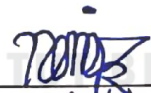
II. Musayyanah, S.ST., M.T.

NIDN. 0730069102

##### **Pembahas**

I. Harianto, S.Kom., M.Eng.

NIDN. 0722087701

  
5/0 2015



Tugas Akhir ini telah diterima sebagai salah satu persyaratan

Untuk memperoleh gelar Sarjana

 FAKULTAS TEKNOLOGI  
DAN INFORMATIKA  
 7/8 19  
**Dr. Jusak**

**Dekan Fakultas Teknologi dan Informatika**

INSTITUT BISNIS DAN INFORMATIKA STIKOM SURABAYA

## SURAT PERNYATAAN

### PERSETUJUAN PUBLIKASI DAN KEASLIAN KARYA ILMIAH

Sebagai mahasiswa Institut Bisnis dan Informatika Stikom Surabaya, saya :

Nama : Revidho Yusuf Demaniqga

NIM : 15410200051

Program Studi : S1 Teknik Komputer

Fakultas : Fakultas Teknologi dan Informatika

Jenis Karya : Tugas Akhir

Judul Karya : **RANCANG BANGUN SISTEM DETEKSI KERUSAKAN  
POLA BATIK MENGGUNAKAN METODE *FIRST  
ORDER***

Menyatakan dengan sesungguhnya bahwa:

1. Demi pengembangan Ilmu Pengetahuan, Teknologi dan Seni, saya menyetujui memberikan kepada Institut Bisnis dan Informatika Stikom Surabaya Hak Bebas Royalti Non-Eksklusif (*Non-Exclusive Royalti Free Right*) atas seluruh isi/ sebagian karya ilmiah saya tersebut di atas untuk disimpan, dialihmediakan dan dikelola dalam bentuk pangkalan data (*database*) untuk selanjutnya didistribusikan atau dipublikasikan demi kepentingan akademis dengan tetap mencantumkan nama saya sebagai penulis atau pencipta dan sebagai pemilik Hak Cipta
2. Karya tersebut di atas adalah karya asli saya, bukan plagiat baik sebagian maupun keseluruhan. Kutipan, karya atau pendapat orang lain yang ada dalam karya ilmiah ini adalah semata hanya rujukan yang dicantumkan dalam Daftar Pustaka saya
3. Apabila dikemudian hari ditemukan dan terbukti terdapat tindakan plagiat pada karya ilmiah ini, maka saya bersedia untuk menerima pencabutan terhadap gelar kesarjanaan yang telah diberikan kepada saya.

Demikian surat pernyataan ini saya buat dengan sebenarnya.

Surabaya, Agustus 2019

Yang menyatakan



Revidho Yusuf Demaniqga  
Nim : 15410200051

## ABSTRAK

Batik merupakan sebagian kecil warisan budaya Indonesia yang memiliki nilai ekspor yang cukup tinggi. Dengan begitu produk batik perlu perhatian yang lebih dalam hal kuantitas serta kualitas pada proses produksinya. Dalam memperbaiki kualitas bahan dan meminimalisir terjadinya kesalahan, sejauh ini pemantauan kualitas dari kain batik hanya dengan pemantauan manual oleh manusia. Maka untuk meminimalisir terjadinya kesalahan produksi dan mempercepat proses *Quality Control*, oleh karena itu pada tugas akhir ini dibuat sebuah sistem pemantauan pada pola kain batik otomatis untuk mendeteksi kesalahan pola menggunakan Image Processing.

Metode yang digunakan pada *Image Processing* adalah metode *First Order* yaitu metode ekstraksi fitur berdasarkan pada karakteristik histogram dari citra. Penggunaan metode *First Order* karena dinilai dapat mendeteksi intensitas warna pada suatu citra. Parameter *First Order* yang digunakan yaitu *Mean* dan *Standar Deviasi* karena dinilai dapat membedakan pola batik dari dispersi warna pada citra.

Hasil pengujian dari tugas akhir ini yaitu berupa keberhasilan mendeteksi kerusakan pola batik dengan nilai PWM ideal 50 dan tingkat akurasi sebesar 90%. Sehingga *prototype* conveyor ini dapat diterapkan pada industri batik.

**Kata Kunci :** Batik, Conveyor, *First Order*, *Image Processing*.

## KATA PENGANTAR

Segala puji syukur kehadiran Allah SWT, karena dengan rahmatnya dan hidayah-Nya penulis dapat menyelesaikan penyusunan Laporan Tugas Akhir yang berjudul “RANCANG BANGUN SISTEM DETEKSI KERUSAKAN POLA BATIK MENGGUNAKAN METODE *FIRST ORDER*”. Laporan Tugas Akhir ini disusun dalam rangka penulisan laporan untuk memperoleh gelar Sarjana Komputer pada program studi S1 Sistem Informasi Stikom Surabaya.

Pada kesempatan ini, penulis ingin mengucapkan rasa terima kasih kepada pihak-pihak yang memberi dukungan dan masukan dalam menyelesaikan laporan Tugas Akhir ini. Oleh karena itu penulis ingin mengucapkan kepada:

1. Orang Tua dan Saudara-saudara saya tercinta yang telah memberikan dorongan dan bantuan baik moral maupun materi sehingga penulis dapat menempuh dan menyelesaikan Tugas Akhir ini.
2. Bapak Dr. Jusak selaku Dekan Fakultas Teknologi dan Informatika (FTI) Institut Bisnis dan Informatika Stikom Surabaya telah membantu proses penyelesaian Tugas Akhir yang dibuat oleh penulis dengan Baik.
3. Bapak Pauladie Susanto, S.Kom., M.T., selaku Ketua Program Studi S1 Sistem Komputer Stikom Surabaya, dan selaku Dosen Penguji atas ijin dan masukkan dalam menyusun Tugas Akhir ini.
4. Bapak Heri Pratikno, M.T., MTCNA., MTCRE. dan Ibu Musayyanah, S.ST., M.T. selaku Dosen Pembimbing yang selalu memberi arahan dan bimbingan dalam menyelesaikan Tugas Akhir beserta laporan ini.
5. Semua staf dosen yang telah mengajar dan memberikan ilmunya.



6. Teman- teman seperjuangan SK angkatan 2015 dan semua pihak yang terlibat namun tidak dapat penulis sebutkan satu persatu atas bantuan dan dukungannya.
7. Serta semua pihak lain yang tidak dapat disebutkan secara satu per satu, yang telah membantu dalam menyelesaikan Tugas Akhir ini baik secara langsung maupun tidak langsung,

Penulis menyadari bahwa Laporan Tugas Akhir ini jauh dari kata sempurna, masih banyak kekurangan dalam menyusun laporan ini. Oleh karena itu dalam kesempatan ini, penulis meminta maaf apabila dalam Laporan Tugas Akhir ini masih banyak kesalahan baik dalam penulisan maupun Bahasa yang digunakan. Penulis juga memerlukan kritik dan saran dari para pembaca yang sifatnya membangun untuk kesempurnaan laporan yang telah penulis susun.

Surabaya, Agustus 2019

Penulis

## DAFTAR ISI

	Halaman
HALAMAN SYARAT .....	ii
MOTTO .....	iii
HALAMAN PERSEMBAHAN .....	iv
HALAMAN PENGESAHAN .....	v
ABSTRAK.....	vii
KATA PENGANTAR .....	viii
DAFTAR ISI.....	x
DAFTAR GAMBAR .....	xiii
DAFTAR TABEL.....	xv
DAFTAR LAMPIRAN.....	xvi
BAB I PENDAHULUAN.....	1
1.1 Latar Belakang.....	1
1.2 Perumusan Masalah.....	3
1.3 Batasan Masalah.....	3
1.4 Tujuan.....	3
1.5 Sistematika Penulisan.....	4
BAB II LANDASAN TEORI.....	5
2.1 Batik.....	5
2.2 <i>Computer Vision</i> .....	7
2.3 Jenis Citra .....	8
2.4 <i>First Order Extraction</i> .....	10
2.5 Microsoft Visual Studio.....	11
2.6 <i>Conveyor</i> .....	12

2.6.1	<i>Roller Conveyor</i> .....	13
2.6.2	<i>Belt Conveyor</i> .....	13
2.7	Arduino Uno .....	14
2.8	Motor DC.....	17
2.9	<i>Webcam</i> .....	17
2.10	<i>Motor Driver</i> BTS9760 .....	20
2.11	UBEC ( <i>Step Down</i> ) .....	21
2.12	PWM .....	23
2.13	Arduino IDE .....	24
2.14	<i>Buzzer</i> .....	27
2.15	Catu Daya ( <i>Power Supply</i> ) .....	28
BAB III ANALISIS DAN PERANCANGAN SISTEM .....		34
3.1	Perancangan Perangkat Keras Sistem.....	36
3.2	Perancangan Perangkat Lunak.....	41
3.2.1	Perancangan Program Pengolahan Citra .....	45
3.2.2	Perancangan Program Arduino .....	51
BAB IV PENGUJIAN .....		53
4.1	Uji Kecepatan Deteksi Kamera .....	53
4.1.1	Tujuan Uji Kecepatan Deteksi Kamera.....	53
4.1.2	Alat Yang Digunakan Pada Uji Kecepatan Deteksi Kamera ..53	
4.1.3	Prosedur Pengujian Pada Uji Kecepatan Deteksi Kamera .....	54
4.1.4	Hasil Pengujian Pada Uji Kecepatan Deteksi Kamera .....	54
4.2	Uji Komunikasi Serial .....	56
4.2.1	Tujuan Uji Komunikasi Serial.....	56
4.2.2	Alat Yang Digunakan Pada Uji Komunikasi Serial .....	56
4.2.3	Prosedur Pengujian Pada Uji Komunikasi Serial .....	56
4.2.4	Hasil Pengujian Pada Uji Komunikasi Serial.....	56

4.3	Uji Akurasi Deteksi Kerusakan Pola Batik .....	58
4.3.1	Tujuan Uji Akurasi Deteksi Kerusakan Pola Batik.....	58
4.3.2	Alat Yang Digunakan Pada Uji Deteksi Kerusakan Pola Batik	58
4.3.3	Prosedur Pengujian Pada Uji Deteksi Kerusakan Pola Batik..	58
4.3.4	Hasil Pengujian Pada Uji Deteksi Kerusakan Pola Batik.....	58
BAB V PENUTUP .....		65
5.1	Kesimpulan.....	65
5.2	Saran .....	66
DAFTAR PUSTAKA .....		67
BIODATA PENULIS .....		73



## DAFTAR GAMBAR

	Halaman
Gambar 2.1 Batik Surya Majapahit.....	6
Gambar 2.2 Cap Pola Batik Surya Majapahit.....	6
Gambar 2.3 Batik Surya Majapahit dengan Isen-isen .....	7
Gambar 2.7 <i>Roller Conveyor</i> .....	13
Gambar 2.8 <i>Belt Conveyor</i> .....	14
Gambar 2.9 Arduino Uno .....	15
Gambar 2.10 Pin Atmega 328.....	16
Gambar 2.11 Motor DC .....	17
Gambar 2.12 <i>Webcam</i> .....	19
Gambar 2.13 Motor Driver BTS9760 .....	20
Gambar 2.14 UBEC ( <i>Step Down</i> ).....	22
Gambar 2.15 PWM (Firman, 2017).....	23
Gambar 2.16 Arduino IDE <i>Startup</i> .....	25
Gambar 2.17 Tampilan Lembar Kerja Arduino IDE .....	26
Gambar 2.18 <i>Buzzer</i> .....	28
Gambar 2.19 Blok Diagram <i>Power Supply</i> .....	30
Gambar 2.20 Transformator / Trafo <i>Step Down</i> .....	30
Gambar 2.21 Rangkaian Penyearah DC <i>Power Supply</i> .....	31
Gambar 2.22 Penyaring (Filter) DC <i>Power Supply</i> .....	32
Gambar 2.23 Rangkaian Dasar IC <i>Voltage Regulator</i> .....	33
Gambar 3.1 Blok Diagram Keseluruhan.....	35

Gambar 3.2 Perancangan Elektronik <i>Conveyor</i> .....	36
Gambar 3.3 Perancangan Aktuator Elektronika .....	38
Gambar 3.4 Perancangan Kontroller Elektronika.....	38
Gambar 3.5 Perancangan Mekanik.....	39
Gambar 3.6 Penempatan Kamera pada <i>Conveyor</i> .....	40
Gambar 3.7 Flowchat Perancangan Keseluruhan Program .....	41
Gambar 3.8 (a) Flowchart pada Visual Studio (b) Flowchart Program Arduino...42	
Gambar 3.9 Citra RGB .....	43
Gambar 3.10 Citra <i>Grayscale</i> .....	43
Gambar 3.11 <i>Flowchart</i> Metode <i>First Order</i> .....	44
Gambar 3.12 Klik <i>File</i> .....	46
Gambar 3.13 Menyimpan <i>Project</i> .....	46
Gambar 3.14 Memilih <i>Application Type</i> dan <i>Additional Option</i> .....	47
Gambar 3.15 Membuat <i>Source File</i> .....	47
Gambar 3.16 Menyimpan <i>Source File</i> .....	48
Gambar 3.17 Properties <i>Source File</i> .....	48
Gambar 3.18 Memasukkan <i>Library Tab C/C++</i> .....	49
Gambar 3.20 Memasukkan <i>Library</i> Pada <i>Input Linker</i> .....	50
Gambar 4.1 Pola <i>Blur</i> .....	55
Gambar 4.2 Pola Tidak <i>Blur</i> .....	55
Gambar 4.3 Monitoring serial dengan <i>Command Prompt</i> .....	57
Gambar 4.4 Monitoring Serial dengan Serial Port.....	57
Gambar 4.6 Pola yang Terdeteksi Salah.....	64

## DAFTAR TABEL

Halaman

Tabel 2.1 Spesifikasi Arduino Uno.....	15
Tabel 2.2 <i>Interface Header Motor Driver</i> .....	21
Tabel 4.1 Pengujian dengan 30 PWM .....	59
Tabel 4.2 Pengujian dengan 50 PWM .....	60
Tabel 4.3 Pengujian dengan 80 PWM .....	60
Tabel 4.4 Pengujian Kain Batik A .....	61
Tabel 4.5 Pengujian Kain Batik B .....	62
Tabel 4.6 Pengujian Kain Batik C .....	62
Tabel 4.7 Pengujian Kain Batik D .....	63

## DAFTAR LAMPIRAN

	Halaman
Lampiran 1. Program Visual Studio .....	69
Lampiran 2. Program Arduino .....	72





# **BAB I**

## **PENDAHULUAN**

### **1.1 Latar Belakang**

Kekayaan budaya Indonesia sangat beragam jenisnya, mulai dari karya musik hingga karya seni yang sudah ada sejak jaman dahulu. Salah satu jenis budaya yang hingga kini masih terjaga adalah batik. Batik adalah salah satu jenis produk sandang yang telah berkembang pesat di Jawa sejak beberapa ratus tahun yang lalu dan merupakan warisan budaya Indonesia telah ditetapkan oleh UNESCO pada tanggal 2 Oktober 2009 sebagai hak kebudayaan intelektual bangsa Indonesia (Arisandi, Suciati, & Yudhi Wijaya, 2011). Pengembangan batik banyak dilakukan di seluruh daerah di Indonesia, hingga tercipta motif – motif baru yang sesuai dengan kebudayaan daerah masing – masing.

Industri batik jika dilihat dari sisi ketenagakerjaan, industri batik di dominasi oleh sektor IKM yang tersebar di 101 sentra seluruh wilayah Indonesia. Dari jumlah tersebut, total penyerapan tenaga kerjanya mencapai 15 ribu orang. Potensi ini terus dikembakan oleh Kementerian Perindustrian untuk mendorong industri padat karya berorientasi ekspor kedepannya (Kementerian Perindustrian, 2018).

Kemenperin mencatat nilai ekspor batik dan produk batik pada 2017 mencapai USD 58,46 juta atau setara Rp. 820,4 miliar (kurs Rp14.000/USD). Pada pertengahan tahun 2018, Kemenperin menargetkan ekspor batik sebesar US\$64,4 juta atau naik 10% dari tahun 2017 dengan pasar utama Amerika Serikat, Jepang, dan Eropa. Perdagangan produk pakaian jadi dunia yang mencapai 442 miliar dolar

AS menjadi peluang besar bagi industri batik untuk meningkatkan pangsa pasarnya, mengingat batik sebagai salah satu bahan baku produk pakaian jadi (Kementrian Perindustrian, 2018).

Berkaca pada presentase nilai ekspor industri batik maka dapat disimpulkan bahwa komoditi industri batik dapat memberikan devisa negara yang cukup banyak. Dengan demikian komoditi batik perlu diperhatikan dalam hal kualitas maupun kuantitas pada sektor produksi. Salah satunya adalah memperbaiki kualitas bahan dan mengurangi kesalahan pada proses produksinya yaitu adanya kerusakan pada kain dan pola batik. Jenis kerusakan batik yang paling utama adalah tinta yang meluber dan kesalahan pada pengulangan tekstur.

Dalam memperbaiki kualitas bahan dan meminimalisir terjadinya kesalahan, sejauh ini pemantauan kualitas dari kain batik hanya dengan pemantauan manual oleh manusia. Maka untuk meminimalisir terjadinya kesalahan produksi dan mempercepat proses *Quality Control*. Oleh karena itu, oleh karena itu pada tugas akhir ini dibuat sebuah sistem pemantauan pada pola kain batik otomatis untuk mendeteksi kesalahan pola menggunakan Image Processing.

Hingga saat ini sudah banyak penelitian yang mengarah pada permasalahan ini, namun kebanyakan penelitian yang telah ada yaitu pengenalan pola batik yang digunakan untuk membedakan dengan pola batik lainnya (Yodha & Kurniawan, 2014). Ada juga penelitian tentang kerusakan tekstil namun hanya kain polos (Dananjaya, 2017). Berdasarkan permasalahan diatas, maka penelitian ini akan dibuat sistem deteksi kerusakan pola batik menggunakan metode *First Order*.

## 1.2 Perumusan Masalah

Berdasarkan latar belakang diatas maka dirumuskan permasalahan :

1. Bagaimana merancang dan membangun sistem deteksi kerusakan pola batik dengan menggunakan kamera?
2. Bagaimana merancang dan membangun sistem deteksi kerusakan pola batik menggunakan metode *First Order*?

## 1.3 Batasan Masalah

Untuk menghindari pembahasan yang lebih luas tesrkait dengan perancangan deteksi kerusakan pola batik menggunakan kamera. Maka penelitian ini ditentukan pada ruang lingkup tertentu antara lain

1. Perekaman menggunakan kamera *webcam* dengan ukuran 5MP.
2. Sistem hanya mendeteksi kerusakan dan memberi peringatan dengan membunyikan *buzzer*.
3. Sistem hanya dapat mendeteksi dengan pencahayaan ruangan yang ideal.
4. Objek yang dideteksi adalah pola kain batik surya majapahit.

## 1.4 Tujuan

Berdasarkan rumusan masalah yang diuraikan diatas, maka tujuan penelitian ini adalah sebagai berikut :

1. Merancang dan membangun sistem deteksi kerusakan pola batik menggunakan kamera.
2. Melakukan implementasi program pengolahan citra digital menggunakan metode *First Order* untuk mendeteksi kerusakan pola batik.

## **1.5 Sistematika Penulisan**

Untuk memudahkan pembaca dalam memahami persoalan dan pembahasannya, maka penulisan laporan tugas akhir ini dibuat dengan sistematika sebagai berikut.

### **BAB I PENDAHULUAN**

Pada bab ini membahas tentang latar belakang masalah dan penjelasan permasalahan secara umum, perumusan masalah serta batasan masalah yang dibuat, tujuan dari pembuatan tugas akhir dan sistematika penulisan buku.

### **BAB II LANDASAN TEORI**

Pada bab ini membahas teori – teori yang berhubungan dan mendukung dalam pembuatan tugas akhir seperti batik, arduino, motor dc, motor driver dan literatur yang menunjang dalam pembuatan Tugas Akhir ini.

### **BAB III ANALISIS DAN PERANCANGAN SISTEM**

Pada bab ini membahas tentang perancangan sistem baik pada bagian perangkat keras, maupun perangkat lunak pada penerapan sistem kinematika ini.

### **BAB IV PENGUJIAN**

Pada bab ini menjelaskan tentang hasil pengujian robot. Pengujian yang dilakukan yaitu uji tingkat akurasi deteksi kerusakan pola batik dan uji kecepatan putar motor *conveyor*.

### **BAB V PENUTUP**

Pada bab ini menjelaskan tentang kesimpulan dan saran. Kesimpulan akan dijelaskan berdasarkan dari hasil pengujian alat tugas akhir ini, serta saran-saran untuk pengembangan.

## **BAB II**

### **LANDASAN TEORI**

#### **2.1 Batik**

Batik merupakan salah satu warisan budaya Indonesia yang sudah terkenal diseluruh dunia. Batik telah ditetapkan oleh UNESCO pada tanggal 2 Oktober 2009 sebagai hak kebudayaan intelektual bangsa Indonesia (Arisandi et al., 2011). Kata batik berasal dari gabungan dua kata Bahasa Jawa yaitu amba dan titik. Amba memiliki arti kain dan titik adalah cara memberi motif pada kain dengan cara di titik-titik (Utami, 2014).

Batik dipercaya sudah ada semenjak zaman Majapahit, dan menjadi sangat populer pada akhir XVIII atau awal abad XIX. Batik yang dihasilkan sampai awal abad XX semuanya adalah batik tulis dan batik cap baru dikenal setelah Perang Dunia I atau sekitar tahun 1920-an. Kata “batik” sendiri berasal dari bahasa Jawa, dan kehadiran batik di Jawa sendiri tidaklah tercatat. G.P. Rouffaer berpendapat bahwa teknik batik ini kemungkinan diperkenalkan dari India atau Srilangka pada abad ke-6 atau ke-7 (Restianti, 2010).

##### **2.1.1 Batik Surya Majapahit**

Batik Surya Majapahit merupakan batik khas dari Kota Mojokerto yang memiliki keunikan yaitu memiliki motif yang digali dari tradisi kebudayaan Kerajaan Majapahit, mengadaptasi elemen – elemen yang ada dalam Kerajaan Majapahit. Batik Surya Majapahit ini juga pernah dipamerkan di Australia pada tahun 2007 dan mulai berkembang dengan sederet nama motif unik dan khas seperti

Mrico Bolong, Sisik Gringsing, Pring Sedapur, dan Masih banyak lagi (Santoso et al., 2014).

Pemberian nama Surya Majapahit karena motif utamanya berupa Surya Majapahit yang merupakan lambang dari Kerajaan Majapahit yang sering dijumpai pada candi – candi peninggalan Kerajaan Majapahit. Surya Majapahit berbentuk cakra segi delapan ini merupakan gambaran dari 9 dewa yang dipuja oleh penduduk Majapahit. Untuk motif pelengkapanya berupa buah maja. Latar dari motif ini berwarna hitam polos tanpa adanya isen – isen, namun seiring berkembangnya jaman banyak juga pembuat batik yang menambahkan aksan kembang maja ataupun aksan lainnya (Santoso et al., 2014).



Gambar 2.1 Batik Surya Majapahit



Gambar 2.2 Cap Pola Batik Surya Majapahit



Gambar 2.3 Batik Surya Majapahit dengan Isen-isen

## 2.2 *Computer Vision*

*Computer vision* adalah transformasi data dari kamera menjadi keputusan atau representasi baru. Semua transformasi tersebut dilakukan untuk mencapai beberapa tujuan tertentu. *Computer vision* merupakan gabungan dari beberapa ilmu bidang seperti ilmu komputer, teknik elektro, matematika, fisiologi, dan ilmu kognitif serta kombinasi perangkat lunak dan perangkat keras pada komputer sehingga komputer dapat meniru penglihatan manusia (Prasojo, 2015).

*Computer vision* memiliki banyak fungsi pendukung untuk mendeteksi kemampuan manusia dalam menangkap informasi dan data. Fungsi – fungsi pendukung tersebut antara lain:

1. Proses penangkapan citra (*image acquisition*)
2. Proses pengolahan citra (*image processing*)
3. Analisa data citra (*image analysis*)
4. Proses pemahaman data citra (*image understanding*)



## 2.3 Jenis Citra

Nilai suatu pixel memiliki nilai dalam rentang tertentu, dari nilai minimum sampai nilai maksimum. Jangkauan yang digunakan berbeda-beda tergantung dari jenis warnanya. Namun secara umum jangkauannya adalah 0 – 255. Citra dengan penggambaran seperti ini digolongkan ke dalam citra *integer*.

Berikut adalah jenis-jenis citra berdasarkan nilai pikselnya :

### 2.3.1. Citra Biner

Citra biner adalah citra digital yang hanya memiliki dua kemungkinan nilai piksel yaitu hitam dan putih. Citra biner juga disebut sebagai citra B&W ( *black and white*) atau citra monokrom. Hanya dibutuhkan 1 bit untuk mewakili nilai setiap piksel dari citra biner. Citra biner sering kali muncul sebagai hasil dari proses pengolahan seperti pengambangan (*thresholding*).

Operasi pengambangan mengelompokkan nilai derajat keabuan setiap piksel ke dalam 2 kelas, hitam dan putih. Dua pendekatan yang digunakan dalam operasi pengambangan adalah pengambangan secara global dan pengambangan secara lokal.



Gambar 2.4 Citra Biner

### 2.3.2. Citra *Grayscale*

Citra grayscale merupakan citra digital yang hanya memiliki satu nilai kanal pada setiap pikselnya, dengan kata lain nilai bagian *RED = GREEN = BLUE*. Nilai



tersebut digunakan untuk menunjukkan tingkat intensitas. Warna yang dimiliki adalah warna dari hitam, keabuan, dan putih. Tingkatan keabuan di sini merupakan warna abu dengan berbagai tingkatan dari hitam hingga mendekati putih. Citra *grayscale* berikut memiliki kedalaman warna 8 bit (256 kombinasi warna keabuan).

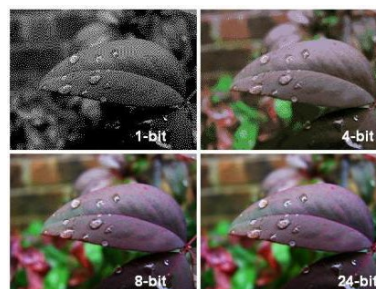


Gambar 2.5 Citra Grayscale

### 2.3.3. Citra Warna (24 bit)

Setiap piksel dari citra warna 24 bit diwakili dengan 24 bit sehingga total 16.777.216 variasi warna. Variasi ini sudah lebih dari cukup untuk memvisualisasikan seluruh warna yang dapat dilihat penglihatan manusia.

Penglihatan manusia dipercaya hanya dapat membedakan hingga 10 juta warna saja. Setiap poin informasi piksel (RGB) disimpan ke dalam 1 byte data. 8 bit pertama menyimpan nilai biru, kemudian diikuti dengan nilai Hijau pada 8 bit kedua dan pada 8 bit terakhir merupakan warna merah.



Gambar 2.6 Citra Warna

## 2.4 First Order Extraction

Ekstraksi ciri orde pertama dilakukan melalui histogram citra yang bersangkutan (histogram menunjukkan nilai derajat keabuan/intensitas piksel dari suatu citra) (Muzami et al., 2016).

*First Order extraction* merupakan metode ekstraksi fitur berdasarkan pada karakteristik histogram dari citra. Nilai-nilai pada histogram merupakan probabilitas kemunculan nilai derajat intensitas dari citra. Bila  $i$  menyatakan tingkat keabuan pada suatu citra maka probabilitas dari  $i$  dinyatakan seperti persamaan berikut :

$$h_i = \frac{n_i}{n} \quad (1)$$

$h_i$  = nilai histogram,  $n_i$  = jumlah piksel yang muncul pada intensitas keabuan ke-  $i$ ,  $n$  = jumlah keseluruhan piksel pada area

Dengan nilai-nilai ini dapat dihitung parameter-parameter ekstraksi fitur orde pertama, diantaranya: *mean*, standard deviasi, *smoothness*, *third moment*, *uniformity* dan *entropy* (Dananjaya, 2017).

*Mean* merupakan ukuran penyebaran (dispersi) nilai intensitas dari suatu citra. Untuk mencari nilai mean dapat dihitung dengan persamaan:

$$m = \sum_{i=0}^{L-1} z_i p(z_i) \quad (2)$$

Dengan  $z_i$  adalah nilai intensitas warna dari citra, dan  $p(z_i)$  merupakan nilai probabilitas kemunculan intensitas tersebut pada citra (histogram). Sedangkan  $L$  adalah nilai tertinggi dari intensitas. Standard deviasi merupakan variasi elemen pada histogram dari suatu citra, yang didefinisikan sebagai:

$$\sigma = \sqrt{\sum_{i=0}^{L-1} (z_i - m)^2 p(z_i)} \quad (3)$$

*Smoothness* menunjukkan tingkat kehalusan relatif dari intensitas suatu citra:

$$R = 1 - \frac{1}{1+\sigma^2} \quad (4)$$

*Third moment* menunjukkan tingkat kemiringan relatif dari histogram suatu citra:

$$\mu_3 = \sum_{i=0}^{L-1} (z_i - m)^3 p(z_i) \quad (5)$$

*Uniformity* merupakan nilai tingkat keseragaman intensitas dari suatu citra:

$$U = \sum_{i=0}^{L-1} p^2(z_i) \quad (6)$$

*Entropy* merupakan nilai ukuran ketidakteraturan bentuk suatu citra.

$$e = - \sum_{i=0}^{L-1} p(z_i) \log_2 p(z_i) \quad (7)$$

## 2.5 Microsoft Visual Studio

Microsoft Visual Studio merupakan sebuah perangkat lunak lengkap (*suite*) yang dapat digunakan untuk melakukan pengembangan aplikasi. Baik itu aplikasi bisnis, aplikasi personal, ataupun komponen aplikasinya dalam bentuk aplikasi *console*, aplikasi Windows, ataupun aplikasi *Web*. Visual Studio mencakup kompiler, SDK, *Integrated Development Environment* (IDE), dan dokumentasi (umumnya berupa *MSDN Library*). Kompiler yang dimasukkan ke dalam paket Visual Studio antara lain Visual C++, Visual C#, Visual Basic, Visual Basic .NET, Visual InterDev, Visual J++, Visual J#, Visual FoxPro, dan Visual SourceSafe (Hemera, 2018).

Microsoft Visual Studio dapat digunakan untuk mengembangkan aplikasi dalam *native code* (dalam bentuk bahasa mesin yang berjalan di Windows) ataupun *managed code* (dalam bentuk Microsoft *Intermediate Language* di atas .NET Framework). Selain itu, Visual Studio juga dapat digunakan untuk mengembangkan aplikasi *Silverlight*, aplikasi *Windows Mobile* (yang berjalan di atas .NET *Compact*

*Framework*).

Konsep dasar pemrograman Visual Studio adalah pembuatan form dengan mengikuti aturan pemrograman *Property*, Metode dan *Event*. Hal ini berarti:

- a. *Property*: Setiap komponen di dalam pemrograman Visual Studio dapat diatur propertinya sesuai dengan kebutuhan aplikasi.
- b. *Metode*: Bahwa jalannya program dapat diatur sesuai aplikasi dengan menggunakan metode pemrograman yang diatur sebagai aksi dari setiap komponen. Metode merupakan tempat untuk mengekspresikan logika pemrograman dari pembuatan suatu program aplikasi.
- c. *Event*: Setiap komponen dapat beraksi melalui *event*, seperti *event click* pada *command button* yang tertulis dalam layar *script Command1\_Click*.

## 2.6 Conveyor

*Conveyor* atau mesin kompayer merupakan peralatan sederhana yang dapat bergerak dari satu tempat ke tempat lain sebagai alat angkut suatu barang tertentu untuk kapasitas kecil sampai besar. *Conveyor* dijadikan sebagai alat transportasi yang cepat dan efisien.

Dalam sebuah industri kadang kala terdapat bahan-bahan yang berat dan juga berbahaya bahkan tidak bisa jika dibawa atau diangkut oleh manusia. sehingga diperlukan alat bantu angkut untuk mengatasi keterbatasan manusia tersebut dalam tenaga untuk menjaga keselamatan dan keamanan para pekerja industri. Untuk itu mesin kompayer banyak dipilih sebagai alat angkut bahan-bahan industri yang padat.

### 2.6.1 *Roller Conveyor*

Merupakan spesifikasi dari *conveyor* yang menggunakan roller untuk mengangkut barang. Dalam perpindahannya, *roller conveyor* memanfaatkan gaya gravitasi bumi. Namun, ada juga yang ditarik atau didorong.

Sedikit berbeda dengan jenis *conveyor* yang lain, sistem roller didesain khusus sehingga dapat sesuai dengan barang yang akan diangkut misalnya berbahan logam, karet, dan lainnya.



Gambar 2.7 *Roller Conveyor*

### 2.6.2 *Belt Conveyor*

Pada dasarnya *belt conveyor* memiliki bentuk yang sederhana. Seperti namanya *conveyor belt* dilengkapi dengan adanya sabuk yang dapat menahan benda-benda padat saat diangkut. *Belt* atau sabuk terbuat dari dari berbagai macam jenis tergantung dari sifat benda yang akan diangkut. Misalnya untuk mengangkut bahan-bahan yang panas, maka diperlukan *belt* yang terbuat dari logam sehingga dapat tahan terhadap panas.



Gambar 2.8 *Belt Conveyor*

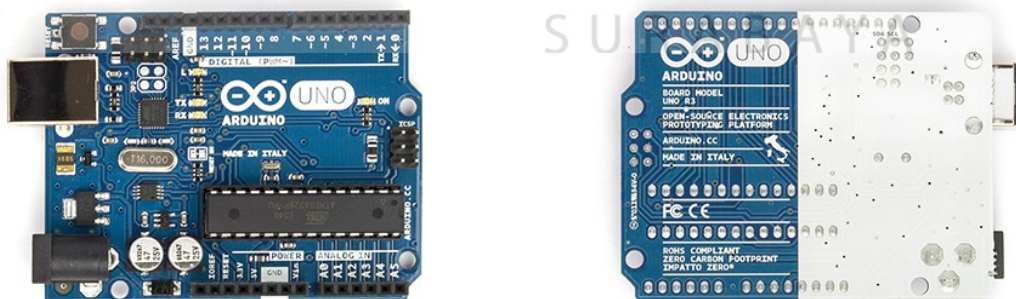
## 2.7 Arduino Uno

Arduino adalah pengendali mikro *single-board* yang bersifat *open-source*, diturunkan dari *Wiring platform*, dirancang untuk memudahkan penggunaan elektronik dalam berbagai bidang. *Hardware* arduino memiliki prosesor Atmel AVR dan *software* Arduino memiliki bahasa pemrograman C. Memori yang dimiliki oleh Arduino Uno sebagai berikut : *Flash Memory* sebesar 32KB, SRAM sebesar 2KB, dan EEPROM sebesar 1KB. *Clock* pada *board* Uno menggunakan XTAL dengan frekuensi 16 Mhz. Dari segi daya, Arduino Uno membutuhkan tegangan aktif kisaran 5 volt, sehingga Uno dapat diaktifkan melalui koneksi USB. Arduino Uno memiliki 28 kaki yang sering digunakan. Untuk Digital I/O terdiri dari 14 kaki, kaki 0 sampai kaki 13, dengan 6 kaki mampu memberikan *output* PWM (kaki 3,5,6,9,10,dan 11). Masing-masing dari 14 kaki digital di Uno beroperasi dengan tegangan maksimum 5 Volt dan dapat memberikan atau menerima maksimum 40mA (Djuandi, 2011).

Untuk *Analog Input* terdiri dari 6 kaki, yaitu kaki A0 sampai kaki A5. Kaki Vin merupakan tempat input tegangan saat menggunakan sumber daya eksternal selain USB dan adaptor. Spesifikasi Arduino Uno R3 dapat dilihat pada Tabel 2.1, gambar Arduino Uno R3 dapat dilihat pada Gambar 2.3 dan pin Atmega 320 dapat dilihat pada Gambar 2.4.

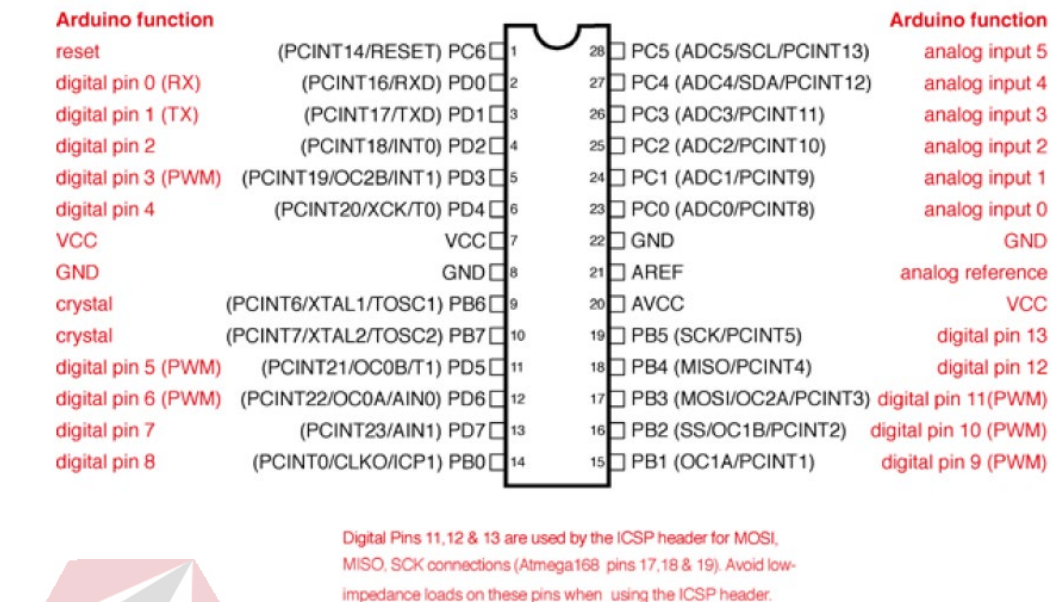
Tabel 2.1 Spesifikasi Arduino Uno

Mikrokontroler	Atmega328
Operasi Tegangan	5 Volt
Input Tegangan	7-12 Volt
Pin I/O Digital	14
Pin Analog	6
Memori Flash	32KB
EEPROM	1 KB
Kecepatan Clock	16 MHz



Gambar 2.9 Arduino Uno





Gambar 2.10 Pin Atmega 328

(<https://doi.org/10.1093/fampra/emi112>)

ATMega328 merupakan mikrokontroler keluarga AVR 8 bit. Beberapa tipe mikrokontroler yang sama dengan ATMega8 ini antara lain ATMega8535, ATMega16, ATMega32, ATMega328, yang membedakan antara mikrokontroler antara lain adalah, ukuran memori, banyaknya GPIO (pin input/output), peripheral (USART, timer, counter, dll). Dari segi ukuran fisik, ATMega328 memiliki ukuran fisik lebih kecil dibandingkan dengan beberapa mikrokontroler diatas. Namun untuk segi memori dan periperial lainnya ATMega328 tidak kalah dengan yang lainnya karena ukuran memori dan periperialnya relatif sama dengan ATMega8535, ATMega32, hanya saja jumlah GPIO lebih sedikit dibandingkan mikrokontroler diatas (Djuandi, 2011).



## 2.8 Motor DC

Motor dc yang digunakan adalah *Planetary Gear 28* dapat dilihat pada gambar 2.5, dengan spesifikasi torsi 15kgfcm, kecepatan 200Rpm, memiliki *rotary* internal dengan *pulse* 7ppr, dan di suplai dengan tegangan 12-24Vdc.



Gambar 2.11 Motor DC

*Rotary* internal yang sudah termasuk pada motor sebenarnya tergolong *rotary* dengan *pulse* rendah namun *rotary* tersebut terpasang langsung pada putaran inti dinamo dan dengan adanya *gearbox* untuk *output* putaran sehingga *rotary* dapat lebih detil dalam melakukan pembacaan putaran.

Motor DC yang digunakan sebagai aktuator robot adalah motor *Planetary Gear-28* dengan perbandingan *gear down* sebesar 1:52 sehingga mempunyai torsi yang cukup besar untuk penggerak robot pada Tugas Akhir ini, sebaliknya motor ini mempunyai kecepatan yang cukup pelan untuk melakukan pergerakan.

## 2.9 Webcam

*Webcam* atau *web camera* adalah sebuah kamera video digital kecil yang dihubungkan dengan komputer melalui port USB atau *port* COM istilah *web camera* merujuk pada teknologi secara umumnya, sehingga kata *web* terkadang diganti dengan kata lain yang mendeskripsikan pemandangan yang ditampilkan di kamera.

Beberapa contoh pemakaian *webcam* yaitu *TruffiCam* yang digunakan untuk memonitor keadaan jalan raya, cuaca dengan *WeatherCam* bahkan dalam memonitoring gunung berapi digunakan *VolcanoCam*. Sebuah web camera yang sederhana terdiri dari sebuah lensa standar, dipasang disebuah papan sirkuit untuk menangkap sinyal gambar, *casing* (penutup), termasuk *casing* depan dan *casing* samping untuk menutupi lensa standar dan memiliki sebuah lubang lensa di-*casing* depan yang berguna untuk memasukkan gambar, kabel *support*, yang dibuat dari bahan yang fleksibel, salah satu ujungnya dihubungkan dengan papan sirkuit dan ujung satu lagi memiliki *connector*, kabel ini dikontrol untuk menyesuaikan ketinggian, arah dan sudut pandang web camera (Rinaldy, Christianti, & Supriyadi, 2016).

Sebuah *web camera* biasanya dilengkapi dengan *software*, *software* ini mengambil gambar-gambar dari camera digital secara *real-time* ataupun dalam interval waktu tertentu dan menyiarkannya melalui koneksi internet. Ada beberapa metode penyiaran, metode yang paling umum digunakan adalah aplikasi yang merubah gambar ke dalam bentuk file JPEG dan mengunggahnya ke *webserver* menggunakan *File Transfer Protocol* (FTP). *Frame rate* mengindikasikan jumlah gambar sebuah *software* dapat diambil dan transfer dalam satu detik. Sebuah *web camera* tidak harus selalu terhubung dengan komputer, ada *web camera* yang memiliki *software web camera* dan *web server built-in*, sehingga yang diperlukan hanyalah koneksi internet. *Web camera* seperti ini dinamakan “*network camera*”.

Sekarang ini web camera yang ada di pasaran pada umurnya terbagi ke dalam dua tipe yang penama web camera permanen (*fixed*) dan *revolving web camera*. Pada *web camera* permanen terdapat pengapit untuk mengapit lensa

standar diposisi yang diinginkan untuk menangkap gambar pengguna. Sedangkan pada *revolving web camera* terdapat landasan dan lensa standar dipasang dilandasan tersebut sehingga dapat disesuaikan ke sudut pandang yang terbaik untuk menangkap gambar pengguna. *Web camera* memiliki fitur-fitur dan pengaturan yang bermacam- macam, diantaranya adalah :

1. *Motion Sensing*, *Web camera* akan mengambil gambar ketika kamera mendeteksi gerakan.
2. *Image archiving*, Penggunaan dapat membuat sebuah archiving yang menyimpan semua gambar dari *web camera* atau hanya gambar-gambar tertentu saat interval pre-set.
3. *Video messaging*, Beberapa program messaging mendukung fitur ini, kabel maupun nirkabel.
4. *Antomotion*, *Camera robotic* yang memungkinkan pengambilan gambar sesuai setting program pengambilan *frame* berdasarkan posisi kamera.



Gambar 2.12 Webcam

## 2.10 Motor Driver BTS9760



Gambar 2.13 Motor Driver BTS9760

Pada *driver motor* DC ini dapat mengeluarkan arus hingga 43A, dengan memiliki fungsi PWM. Tegangan sumber DC yang dapat diberikan antara 5.5V-27V<sub>DC</sub>, sedangkan tegangan *input* level antara 3.3V-5V<sub>DC</sub>, *driver* motor ini menggunakan rangkaian *full H-bridge* dengan IC BTS7960 dengan perlindungan saat terjadi panas dan arus berlebihan. Spesifikasi:

- Double BTS7960 high current (43A) H-bridge drivers
- Input voltage: 6V-27V
- Model: IBT-2
- Maximum current: 43A
- Input level: 3.3-5V
- Control mode: PWM or level

Berikut adalah table I/O pada *BTS7960*.

Tabel 2.2 *Interface Header Motor Driver*

NAMA	I/O	FUNGSI
RPWM	I	Input PWM Forward Level ,Aktif High
LPWM	I	Input PWM Reverse Level ,Aktif High
R_EN	I	Input Enable Forward Driver, Aktif High
L_EN	I	Input Enable Reverse Driver, Aktif High
R_IS	I	Forward Drive ,Side current alarm output
L_IS	I	Reverse Drive ,Side current alarm output
VCC	I	+5 V Power Supply Mikrokontroler
GND	I	Gnd Power Supply Mikrokontroler
M-	O	Di hubungkan ke Motor DC (V-)
M+	O	Di hubungkan ke Motor DC (V+)
B+	O	Tegangan Input V+ Motor
B-	O	Tegangan Input V- Motor

### 2.11 UBEC (*Step Down*)

Mengubah tegangan, tinggi ke rendah atau sebaliknya, memerlukan rangkaian yang tepat, agar daya dapat di-*deliver* dengan tingkat efisiensi setinggi mungkin. Menurunkan tegangan dengan menggunakan IC regulator seperti 7805, sangat umum digunakan. Regulator ini memiliki kemampuan menangani arus hingga 1A, dengan  $V_{in}$  minimal sama dengan 7V, untuk menghasilkan *output* 5V. Dengan perhitungan sederhana, bila  $V_{in} = 9V$ , maka disipasi daya  $\sim 4$  Watt, satu nilai yang cukup besar (panas), atau menggunakan regulator linier tipe LDO, seperti 2940, yang juga memiliki kemampuan menangani arus hingga 1A, dengan  $V_{in}$  minimal sama dengan 5.5V, untuk menghasilkan *output* 5V.

Pilihan lain adalah regulator switching. Untuk kebutuhan mencatu motor servo atau rangkaian lain yang bekerja pada tingkat tegangan 5V – 6V, dapat

menggunakan UBEC. UBEC – *Universal Battery Elimination Circuit* adalah rangkaian elektronik yang mengambil daya dari battery pack atau sumber DC lainnya, dan menurunkannya ke level tegangan 5V atau 6V. Tegangan *input* maksimum tergantung pada spesifikasi UBEC.



Gambar 2.14 UBEC (*Step Down*)

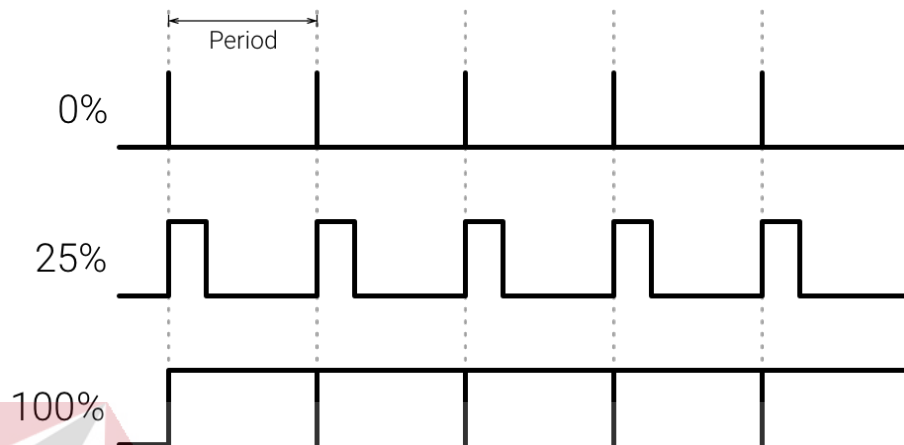
(Tjahyadi, 2018)

UBEC biasanya digunakan pada aplikasi yang memerlukan arus lebih tinggi, dan perangkat ini mampu mengantarkan daya dengan efisiensi hingga 92%. Ketika memilih UBEC, pastikan model UBEC yang dipilih memiliki rating arus yang sesuai dengan kebutuhan (beban). Bentuk fisik dari ubec ditampilkan pada gambar 2.11. Rangkaian lain yang juga sering dibutuhkan adalah DC-DC *Booster*. Sebagai contoh, satu produk DC-DC, mampu menghasilkan *output* 3.7V – 34V dengan *input* 3.7V – 34V. Artinya, dengan tegangan *input* minimum 3.7V dapat dihasilkan *output* maksimum 34V, dengan arus *input* maksimum 3A, serta mampu men-deliver daya dengan tingkat efisiensi hingga 90%.

Tegangan *input* tidak boleh lebih besar dari *output* yang dihasilkan. Dalam banyak aplikasi, khususnya aplikasi robotik, seringkali dibutuhkan kombinasi

keduanya, agar dapat menggunakan satu catu battery pack (Tjahyadi, 2018).

## 2.12 PWM



Gambar 2.15 PWM (Firman, 2017)

PWM ( *Pulse Width Modulation*) adalah salah satu teknik modulasi dengan mengubah lebar pulsa (*duty cycle*) dengan nilai amplitudo dan frekuensi yang tetap. Pada gambar 2.12 ditampilkan perbandingan *duty cycle* pada PWM. Satu siklus pulsa merupakan kondisi high kemudian berada di zona transisi ke kondisi low. Lebar pulsa PWM berbanding lurus dengan amplitudo sinyal asli yang belum termodulasi. *Duty Cycle* merupakan representasi dari kondisi logika high dalam suatu periode sinyal dan di nyatakan dalam bentuk (%) dengan *range* 0% sampai 100%, sebagai contoh jika sinyal berada dalam kondisi high terus menerus artinya memiliki *duty cycle* sebesar 100%. Jika waktu sinyal keadaan high sama dengan keadaan low maka sinyal mempunyai *duty cycle* sebesar 50%.

Aplikasi penggunaan PWM biasanya ditemui untuk pengaturan kecepatan motor dc, pengaturan cerah/redup LED, dan pengendalian sudut pada motor servo. Contoh penggunaan PWM pada pengaturan kecepatan motor dc semakin besar nilai

*duty cycle* yang diberikan maka akan berpengaruh terhadap cepatnya putaran motor. Apabila nilai *duty cycle*-nya kecil maka motor akan bergerak lambat (Firman, 2017).

Untuk membandingkannya terhadap tegangan DC, PWM memiliki 3 mode operasi yaitu:

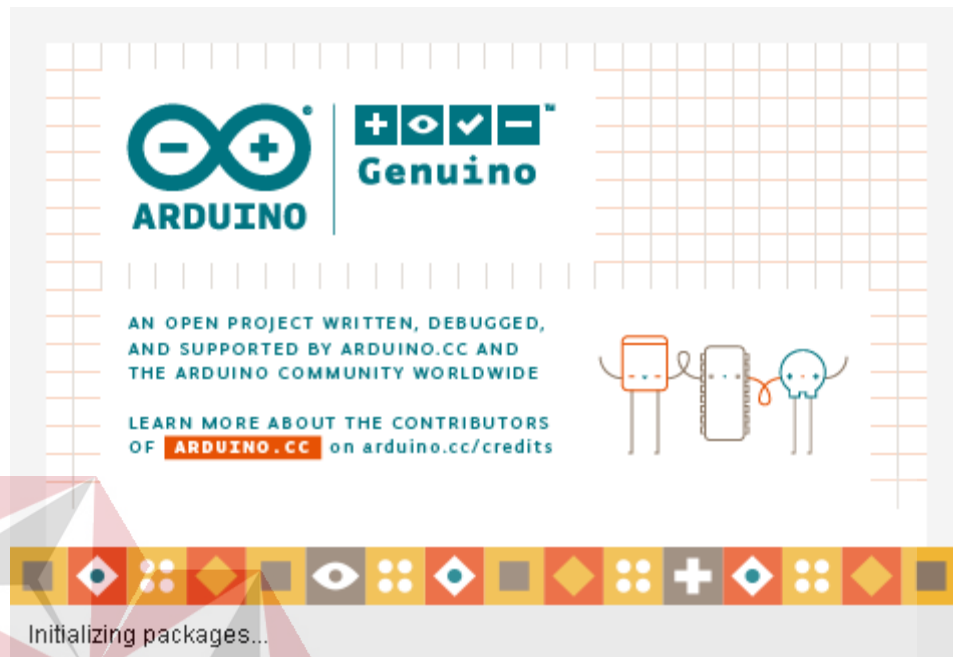
1. *Inverted Mode*. Pada mode *inverted* ini jika nilai sinyal lebih besar dari pada titik pembanding (*compare level*) maka *output* akan di set high (5v) dan sebaliknya jika nilai sinyal lebih kecil maka *output* akan di set low (0v) seperti pada gelombang A pada gambar di atas.
2. *Non Inverted Mode*. Pada mode *non inverted* ini *output* akan bernilai high (5v) jika titik pembanding (*compare level*) lebih besar dari pada nilai sinyal dan sebaliknya jika bernilai low (0v) pada saat titik pembanding lebih kecil dari nilai sinyal seperti pada gelombang B pada gambar di atas.
3. *Toggle Mode*. Pada mode *toggle* *output* akan beralih dari nilai high (5v) ke nilai low (0v) jika titik pembanding sesuai dan sebaliknya beralih dari nilai low ke high.

## 2.13 Arduino IDE

Arduino IDE (*Integrated Development Environment*) adalah software yang di gunakan untuk memprogram di arduino, dengan kata lain Arduino IDE sebagai media untuk memprogram *board* arduino. Arduino IDE ini berguna sebagai *text editor* untuk membuat, mengedit, dan juga mevalidasi kode program. Bisa juga digunakan untuk meng-upload ke board Arduino. Kode program yang digunakan pada Arduino disebut dengan istilah Arduino “sketch” atau disebut juga *source*



*code* arduino, dengan ekstensi file *source code* .ino. Gambar 2.13 menampilkan tampilan awal Arduino IDE saat pertama kali dibuka.



Gambar 2.16 Arduino IDE *Startup*

(Sinaryuda, 2017)

Editor *Programming* pada umumnya memiliki fitur untuk *cut / paste* dan untuk *find / replace* teks, demikian juga pada Arduino IDE. Gambar 2.14 menampilkan tampilan lembar kerja pada Arduino IDE. Pada bagian keterangan aplikasi memberikan pesan balik saat menyimpan dan mengekspor serta sebagai tempat menampilkan kesalahan. Konsol log menampilkan teks log dari aktifitas Arduino IDE, termasuk pesan kesalahan yang lengkap dan informasi lainnya. Pojok kanan bawah menampilkan port serial yang di gunakan. Tombol *toolbar* terdapat ikon tombol pintas untuk memverifikasi dan meng-upload program, membuat, membuka, dan menyimpan *sketch*, dan membuka monitor serial (Sinaryuda, 2017).



Gambar 2.17 Tampilan Lembar Kerja Arduino IDE

(Sinaryuda, 2017)

*Verify* pada versi sebelumnya dikenal dengan istilah *Compile*. Sebelum aplikasi di-upload ke board Arduino, biasanya untuk memverifikasi terlebih dahulu *sketch* yang dibuat. Jika ada kesalahan pada *sketch*, nanti akan muncul error. Proses *Verify* / *Compile* mengubah *sketch* ke binary code untuk di-upload ke mikrokontroler.

*Upload* tombol ini berfungsi untuk mengupload sketch ke *board* Arduino. Walaupun kita tidak mengklik tombol *verify*, maka sketch akan di-*compile*, kemudian langsung di-*upload* ke *board*. Berbeda dengan tombol *verify* yang hanya

berfungsi untuk memverifikasi *source code* saja.

- *New Sketch* Membuka window dan membuat *sketch* baru.
- *Open Sketch* Membuka *sketch* yang sudah pernah dibuat. *Sketch* yang dibuat dengan IDE Arduino akan disimpan dengan ekstensi file *.ino*
- *Save Sketch* menyimpan *sketch*, tapi tidak disertai dengan mengkompile.
- *Serial Monitor* Membuka *interface* untuk komunikasi serial, nanti akan kita diskusikan lebih lanjut pada bagian selanjutnya.
- Keterangan Aplikasi pesan-pesan yang dilakukan aplikasi akan muncul di sini, misal *Compiling* dan *Done Uploading* ketika kita meng-*compile* dan meng-*upload sketch* ke *board* Arduino
- Konsol log Pesan-pesan yang dikerjakan aplikasi dan pesan-pesan tentang *sketch* akan muncul pada bagian ini. Misal, ketika aplikasi meng-*compile* atau ketika ada kesalahan pada *sketch* yang kita buat, maka informasi *error* dan baris akan diinformasikan di bagian ini.
- Baris *Sketch* bagian ini akan menunjukkan posisi baris kursor yang sedang aktif pada *sketch*.

## 2.14 Buzzer

*Buzzer* adalah sebuah komponen elektronika yang berfungsi untuk mengubah getaran listrik menjadi getaran suara getaran listrik menjadi getaran suara. Pada dasarnya prinsip kerja *buzzer* hampir sama dengan loudspeaker, jadi *buzzer* juga terdiri dari kumparan yang terpasang pada diafragma dan kemudian kumparan tersebut dialiri arus sehingga menjadi elektromagnet, kumparan tadi akan tertarik ke dalam atau keluar, tergantung dari arah arus dan polaritas magnetnya,

karena kumparan dipasang pada diafragma maka setiap gerakan kumparan akan menggerakkan diafragma secara bolak-balik sehingga membuat udara bergetar yang akan menghasilkan suara. *Buzzer* biasa digunakan sebagai indikator bahwa proses telah selesai atau terjadi suatu kesalahan pada sebuah alat (alarm) .



Gambar 2.18 *Buzzer*

### 2.15 Catu Daya (*Power Supply*)

Catu daya atau sering disebut dengan Power Supply adalah perangkat elektronika yang berguna sebagai sumber daya untuk perangkat lain. Secara umum istilah catu daya berarti suatu sistem penyearah-filter yang mengubah ac menjadi dc murni. Sumber DC seringkali dapat menjalankan peralatan-peralatan elektronika secara langsung, meskipun mungkin diperlukan beberapa cara untuk meregulasi dan menjaga suatu ggl agar tetap meskipun beban berubah-ubah. Energi yang paling mudah tersedia adalah arus bolak-balik, harus diubah atau disearahkan menjadi dc berpulsa (pulsating dc), yang selanjutnya harus diratakan atau disaring menjadi tegangan yang tidak berubah-ubah. Tegangan DC juga memerlukan regulasi tegangan agar dapat menjalankan rangkaian dengan sebaiknya.

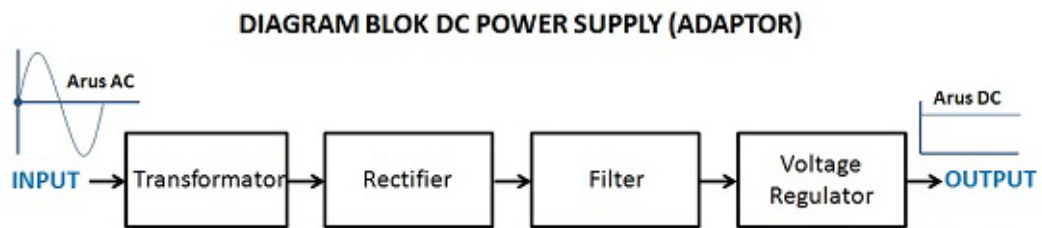
### 2.15.1 Prinsip Kerja DC *Power Supply*

Arus Listrik yang kita gunakan di rumah, kantor dan pabrik pada umumnya adalah dibangkitkan, dikirim dan didistribusikan ke tempat masing-masing dalam bentuk Arus Bolak-balik atau arus AC (Alternating Current). Hal ini dikarenakan pembangkitan dan pendistribusian arus Listrik melalui bentuk arus bolak-balik (AC) merupakan cara yang paling ekonomis dibandingkan dalam bentuk arus searah atau arus DC (Direct Current).

Akan tetapi, peralatan elektronika yang kita gunakan sekarang ini sebagian besar membutuhkan arus DC dengan tegangan yang lebih rendah untuk pengoperasiannya. Oleh karena itu, hampir setiap peralatan Elektronika memiliki sebuah rangkaian yang berfungsi untuk melakukan konversi arus listrik dari arus AC menjadi arus DC dan juga untuk menyediakan tegangan yang sesuai dengan rangkaian Elektronika-nya. Rangkaian yang mengubah arus listrik AC menjadi DC ini disebut dengan DC Power Supply atau dalam bahasa Indonesia disebut dengan Catu daya DC. DC Power Supply atau Catu Daya ini juga sering dikenal dengan nama “Adaptor”.

Sebuah DC Power Supply atau Adaptor pada dasarnya memiliki 4 bagian utama agar dapat menghasilkan arus DC yang stabil. Keempat bagian utama tersebut diantaranya adalah Transformer, Rectifier, Filter dan Voltage Regulator.

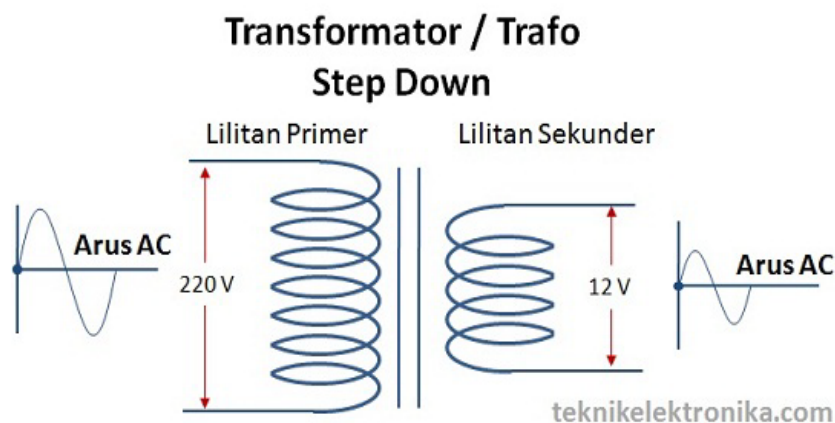
Sebelum kita membahas lebih lanjut mengenai Prinsip Kerja DC Power Supply, sebaiknya kita mengetahui Blok-blok dasar yang membentuk sebuah DC Power Supply atau Pencatu daya ini. Dibawah ini adalah Diagram Blok DC Power Supply (Adaptor) pada umumnya. Berikut adalah diagram blok diagram *power supply*.

Gambar 2.19 Blok Diagram *Power Supply*

(Di, 2017) (Dickson Kho, 2014)

## a. Transformator (Transformer / Trafo)

Transformator (Transformer) atau disingkat dengan Trafo yang digunakan untuk DC Power supply adalah Transformer jenis Step-down yang berfungsi untuk menurunkan tegangan listrik sesuai dengan kebutuhan komponen Elektronika yang terdapat pada rangkaian adaptor (DC Power Supply). Transformator bekerja berdasarkan prinsip Induksi elektromagnetik yang terdiri dari 2 bagian utama yang berbentuk lilitan yaitu lilitan Primer dan lilitan Sekunder. Lilitan Primer merupakan Input dari pada Transformator sedangkan Output-nya adalah pada lilitan sekunder. Meskipun tegangan telah diturunkan, Output dari Transformator masih berbentuk arus bolak-balik (arus AC) yang harus diproses selanjutnya.

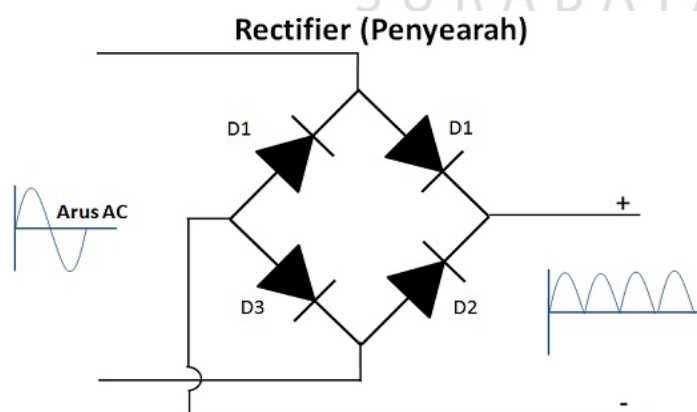
Gambar 2.20 Transformator / Trafo *Step Down*

b. Penyearah Gelombang (*Rectifier*)

Rectifier atau penyearah gelombang adalah rangkaian Elektronika dalam Power Supply (catu daya) yang berfungsi untuk mengubah gelombang AC menjadi gelombang DC setelah tegangannya diturunkan oleh Transformator Step down. Rangkaian Rectifier biasanya terdiri dari komponen Dioda. Terdapat 2 jenis rangkaian Rectifier dalam Power Supply yaitu “Half Wave Rectifier” yang hanya terdiri dari 1 komponen Dioda dan “Full Wave Rectifier” yang terdiri dari 2 atau 4 komponen dioda.

Prinsip penyearah (rectifier) yang paling sederhana ditunjukkan pada gambar 2.14 berikut ini. Transformator diperlukan untuk menurunkan tegangan AC dari jala-jala listrik pada kumparan primernya menjadi tegangan AC yang lebih kecil pada kumparan sekundernya.

Pada rangkaian ini, dioda berperan untuk hanya meneruskan tegangan positif ke beban RL. Ini yang disebut dengan penyearah setengah gelombang (half wave). Untuk mendapatkan penyearah gelombang penuh (full wave) diperlukan transformator dengan center tap (CT) seperti pada gambar 2.2.

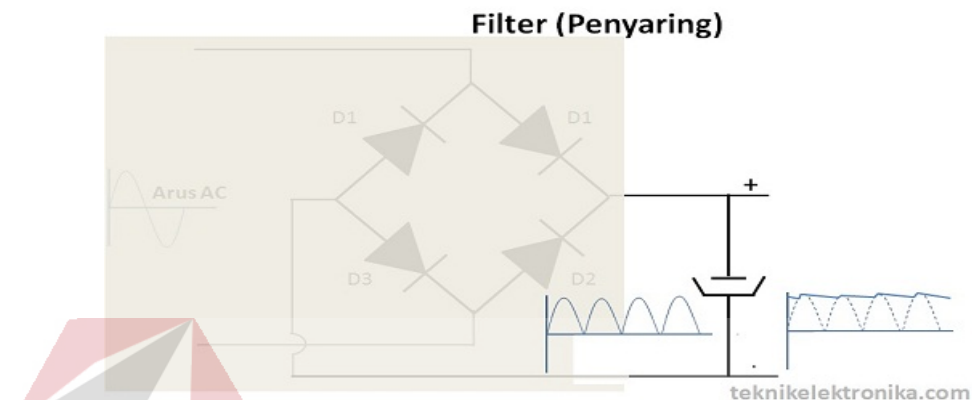


Gambar 2.21 Rangkaian Penyearah DC *Power Supply*

(Dickson Kho, 2014)

c. Penyaring (*Filter*)

Dalam rangkaian DC Power supply, filter digunakan untuk meratakan sinyal arus yang keluar dari Rectifier. Filter ini biasanya terdiri dari komponen Kapasitor (Kondensator) yang berjenis Elektrolit atau ELCO (*Electrolyte Capacitor*).



Gambar 2.22 Penyaring (Filter) DC Power Supply

(Dickson Kho, 2014)

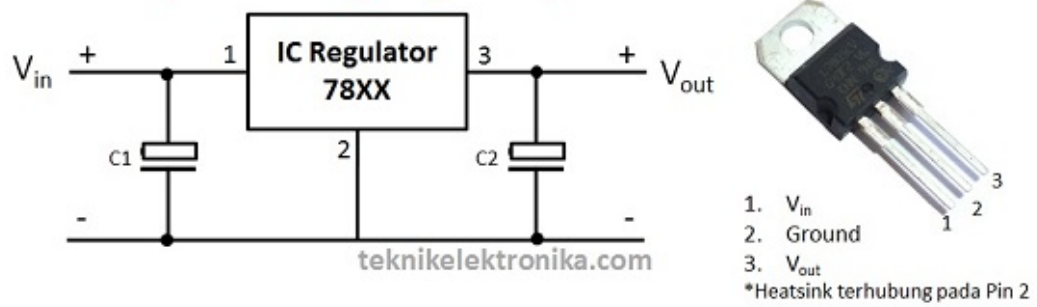
d. Pengatur Tegangan (*Voltage Regulator*)

Untuk menghasilkan Tegangan dan Arus DC (arus searah) yang tetap dan stabil, diperlukan *Voltage Regulator* yang berfungsi untuk mengatur tegangan sehingga tegangan Output tidak dipengaruhi oleh suhu, arus beban dan juga tegangan input yang berasal Output Filter. *Voltage Regulator* pada umumnya terdiri dari Dioda Zener, Transistor atau IC (Integrated Circuit).

Pada DC Power Supply yang canggih, biasanya *Voltage Regulator* juga dilengkapi dengan *Short Circuit Protection* (perlindungan atas hubung singkat), *Current Limiting* (Pembatas Arus) ataupun *Over Voltage Protection* (perlindungan atas kelebihan tegangan) (Dickson Kho, 2014).



### Rangkaian Dasar IC Voltage Regulator (Fixed Voltage Regulator)



Gambar 2.23 Rangkaian Dasar IC *Voltage Regulator*

(Dickson Kho, 2014)



### BAB III

#### ANALISIS DAN PERANCANGAN SISTEM

Pada bagian bab ini akan dijelaskan mengenai perancangan keseluruhan sistem. Perancangan perangkat lunak antara lain aplikasi pemantauan terhadap pola batik dengan bantuan kamera *webcam*, pemrosesan gambar dari kamera sebagai tolak ukur apabila terdapat kesalahan pada pola batik yang kemudian akan membunyikan *buzzer* sebagai isyarat adanya kesalahan atau kerusakan pada pola batik. Perancangan perangkat keras meliputi pengkabelan seluruh perangkat elektronika pada *conveyor* dan perancangan tata letak motor. Dibagian bab ini juga dijelaskan mengenai blok diagram keseluruhan dari sistem.

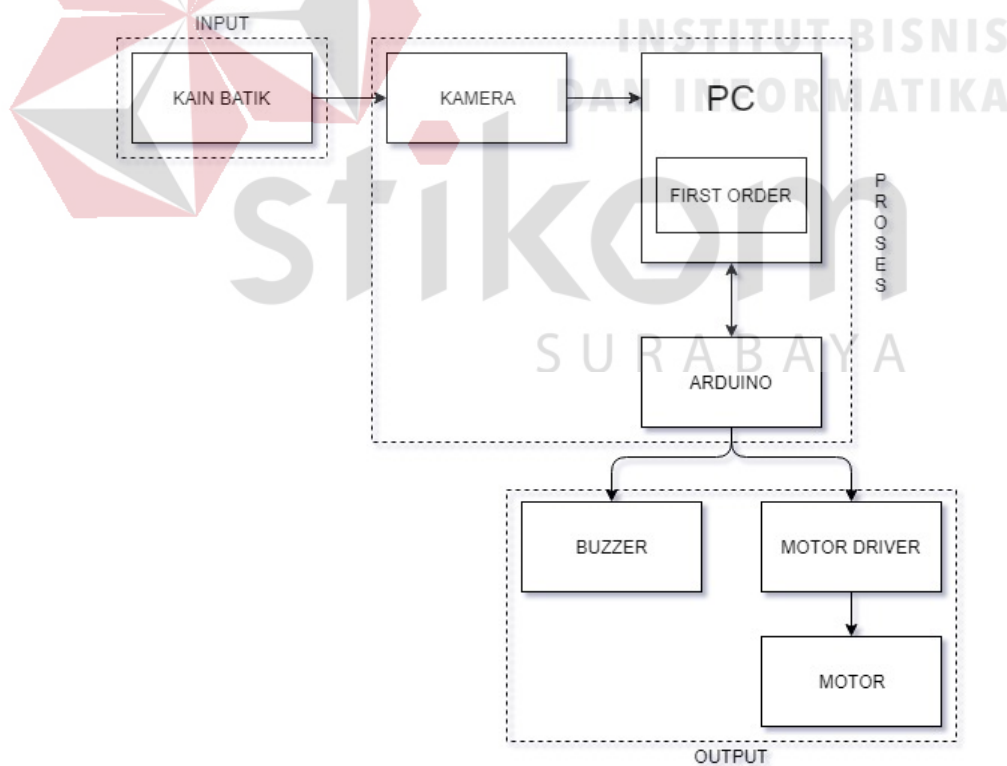
Pada Tugas Akhir ini digunakan perhitungan nilai keluaran dari metode *First Order* untuk menentukan apakah kain batik yang dideteksi layak atau tidak, sehingga apabila terdeteksi tidak layak akan memerintahkan arduino untuk membunyikan *buzzer*.

Aplikasi ini digunakan untuk melakukan pemrosesan gambar yang ditangkap kamera dan memudahkan pengguna dalam melakukan kontrol maupun *monitoring* terhadap kesalahan pada pola batik.

Pada blok diagram keseluruhan sistem deteksi kerusakan pola batik, menjelaskan tentang sistem secara garis besar :

1. Dari *conveyor* berputar, kamera mulai merekam pola batik secara *realtime*.
2. Hasil tangkapan kamera kemudian diproses menggunakan Metode *First Order*.

3. Nilai data yang diperoleh dari keluaran Metode *First Order* digunakan sebagai acuan untuk mendeteksi kerusakan pada pola batik.
4. Apabila nilai dari hasil *First Order* tidak sama dengan *range* nilai yang telah ditentukan sebagai *threshold*, maka program menghitung berapa banyak kesalahan yang terdeteksi pada satu potong kain batik.
5. Ketika satu potong kain telah dideteksi, dilakukan penekanan tombol “b” pada laptop, yang menjadi isyarat bahwa satu potong kain telah terdeteksi dan mengirim karakter “b” ke Arduino.
6. Kemudian sistem akan memberi keputusan apakah kain tersebut layak atau tidak dengan bunyi dari *buzzer*.
7. Apabila *buzzer* berbunyi maka kain tersebut dinyatakan tidak layak.

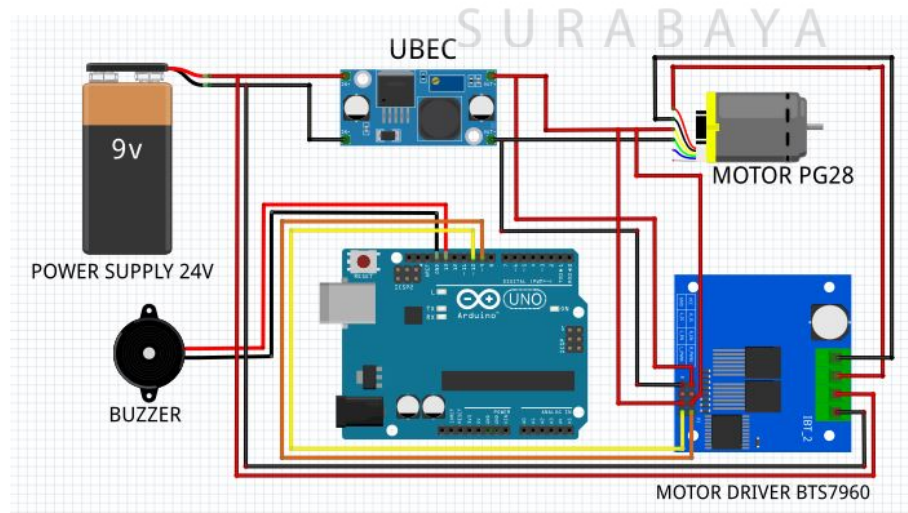


Gambar 3.1 Blok Diagram Keseluruhan

### 3.1 Perancangan Perangkat Keras Sistem

Pada sub bab ini memaparkan mengenai seluruh perancangan perangkat keras (hardware) yang digunakan pada Tugas Akhir ini. Secara garis besar perangkat keras sistem ini berupa konveyor yang digerakkan oleh motor sebagai aktuator, kamera sebagai inputan gambar pola batik yang akan diproses menggunakan metode *First Order* yang kemudian dijadikan sebagai input untuk arduino membunyikan *buzzer*.

*Conveyor* pada Tugas Akhir ini menggunakan kontroler Arduino Uno yang sudah berbasis pada CPU ATmega328P, sehingga dinilai cukup menjalankan program yang digunakan. Arduino Uno digunakan pada Tugas Akhir ini dikarenakan untuk menekan biaya produksi. Penggunaan kontroler Arduino Uno ini juga dikarenakan tidak membutuhkan port yang banyak dimana hanya membutuhkan 2 port untuk motor. Pada tugas akhir ini seluruh sistem pengendalian di kontrol melalui satu kontroler, atau tidak perlu dilakukan penambahan kontroler dengan mode *master/slave*.

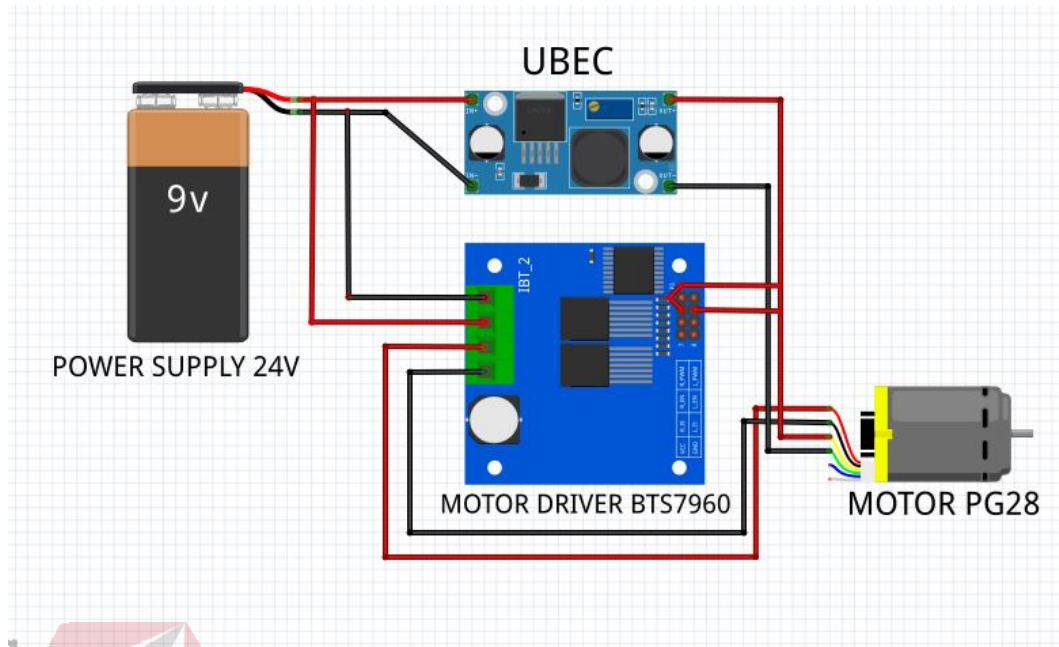


Gambar 3.2 Perancangan Elektronik *Conveyor*

Dapat dilihat pada gambar 3.2, rangkaian elektronika pada Tugas Akhir ini menggunakan beberapa komponen antara lain, *power supply* 24V, Ubec, Arduino Uno, Motor Driver BTS7960, Motor PG28, dan *Buzzer*.

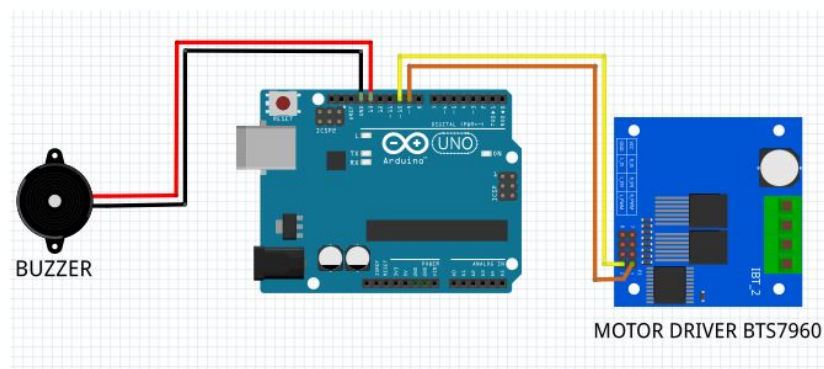
Sistem pada *conveyor* ini mempunyai dua sumber tegangan yaitu sebesar 24V dan 5V, dimana sumber tegangan 24V digunakan sebagai sumber tegangan *input* motor driver untuk menggerakkan aktuator berupa motor, dan untuk sumber tegangan 5V digunakan sebagai sumber tegangan untuk mengaktifkan motor PG28 dan motor *driver*.

Motor *driver* yang digunakan adalah motor *driver* dengan kapasitas ampere yang cukup besar yaitu sebesar 30 ampere sesuai dengan tipenya BTS7960, pada Tugas Akhir ini motor *driver* mengendalikan motor DC dengan sumber tegangan 24V yang berasal dari baterai *power supply* 24V. Untuk mengendalikan motor driver ini diperlukan sebanyak 3 *masukkan* yaitu direksi 1, direksi 2, dan kontrol PWM. Direksi 1 dan direksi 2 digunakan sebagai menentukan arah putaran motor, saat diberi dengan kondisi (*high*) untuk direksi 1 dan (*low*) untuk direksi 2 maka motor akan berputar searah jarum jam dan berlaku juga sebaliknya, namun arah putaran motor juga dapat dipengaruhi dari *input* sumber tegangan motor yang dibalik untuk port M+ atau M- pada motor driver. Untuk kontrol PWM dilakukan dengan memberikan *input* berupa pulsa dengan perbandingan *duty cycle* sesuai kebutuhan guna menentukan kecepatan putar motor DC.



Gambar 3.3 Perancangan Aktuator Elektronika

Pada Gambar 3.3 merupakan rangkaian elektronika dari aktuator yang digunakan untuk memutar *conveyor*. Dimana rangkaian ini bersumber dari *power supply* 24V yang kemudian disambungkan ke UBEC yang keluarannya sebagai sumber tegangan 5V. Daya dari *power supply* juga berguna untuk menggerakkan motor yang spesifikasinya sebesar 24V melalui Motor Driver terlebih dahulu. Sedangkan daya 5V yang keluar dari UBEC berguna untuk mengaktifkan *port enable* pada Motor Driver dan juga Motor PG28.



Gambar 3.4 Perancangan Kontroller Elektronika



Sedangkan pada Gambar 3.4 rangkaian kontroller yang digunakan di sistem ini menggunakan Arduino Uno yang dihubungkan ke Motor Driver BTS7960 dan *Buzzer*. Pin 9 dan 10 pada Arduino Uno yang merupakan pin untuk PWM dihubungkan ke pin L\_PWM dan R\_PWM pada Motor Driver BTS7960 sebagai kontrol PWM atau kecepatan dan arah putar motor. Kemudian *buzzer* dihubungkan ke pin 13 yang merupakan pin *output* dan pin 14 yang merupakan pin *ground*

### 3.1.1 Perancangan Mekanik

Seperti yang ada pada industri tekstil kebanyakan, mekanik yang digunakan adalah berupa penggulung kain yang melewati *conveyor*. Begitu juga dengan mekanik pada tugas akhir ini berupa *conveyor* yang secara dimensi berukuran panjang 100 cm, lebar 50 cm, dan tinggi 100 cm. Panjang *conveyor* dari *roller* ujung ke ujung sepanjang 83cm dengan lebar alas *conveyor* 40cm.



Gambar 3.5 Perancangan Mekanik

*Conveyor* pada gambar 3.5 tersebut terbuat dari kerangka besi dengan ketebalan 3mm dan tiang penyangga kamera terbuat dari *stainless steel*. Pada ujung dari *conveyor* ini terdapat skrup yang berfungsi untuk mengencangkan atau mengendorkan *belt conveyor*. Penggunaan bahan kain kanvas pada *belt conveyor* bertujuan untuk tidak membebani kinerja motor dan juga agar pada sambungan *belt* tidak terdapat tumpukan yang tebal, sehingga kain yang dideteksi menggunakan kamera tidak bergelombang.



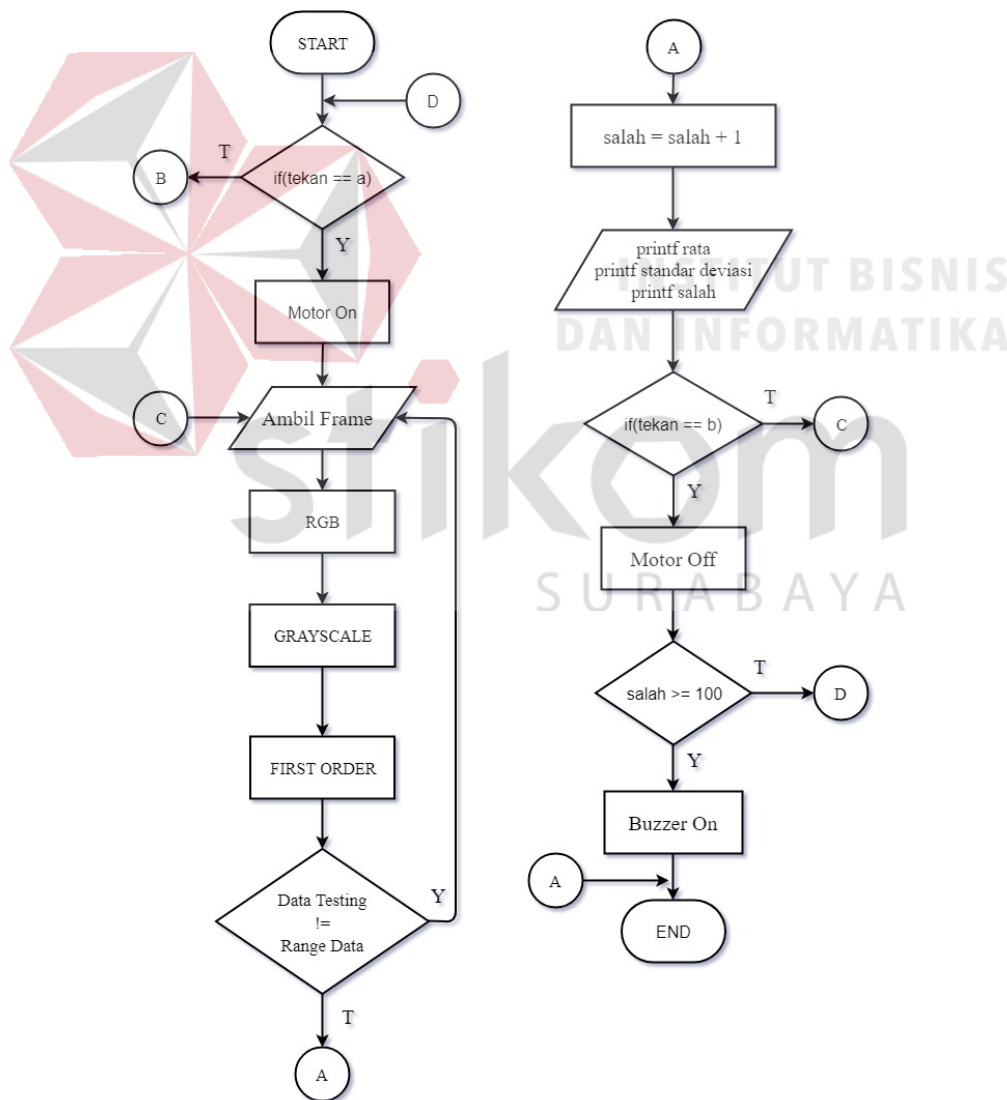
Gambar 3.6 Penempatan Kamera pada *Conveyor*

Pada Gambar 3.6 adalah penempatan kamera pada *conveyor*, bahan yang digunakan tiang penyangga kamera ini menggunakan *stainless steel* dengan tinggi dan lebar yang dapat diatur. Tinggi kamera yang dapat menangkap seluruh lebar dari kain batik yaitu 42 cm dari kain batik dan posisi ideal kamera berada pada tengah dari lebar kain batik. Dengan penentuan tinggi kamera tersebut, kamera dapat menangkap keseluruhan dari lebar kain dan dapat menangkap keseluruhan lebar dari kain batik.



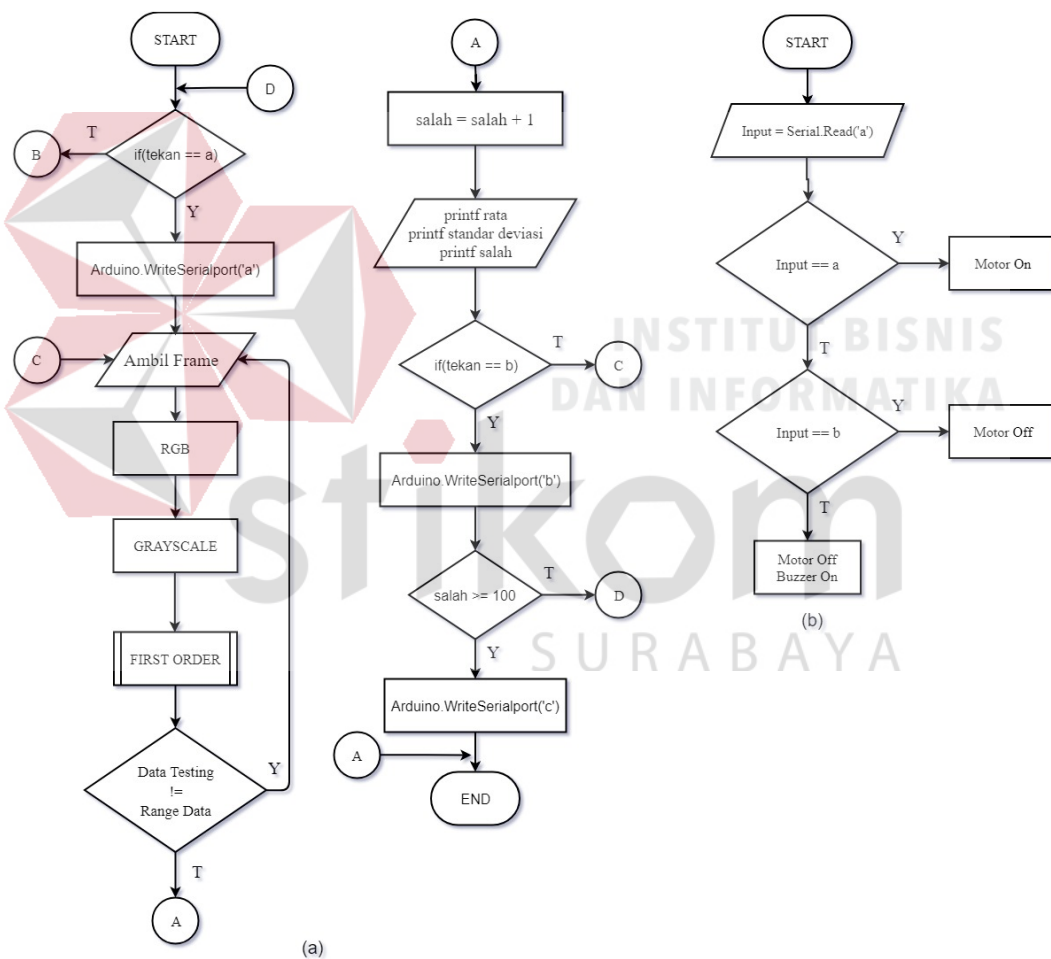
### 3.2 Perancangan Perangkat Lunak

Pada sistem Tugas Akhir ini tidak hanya terdapat perangkat keras tetapi juga perangkat lunak sebagai satu kesatuan yang saling berhubungan satu sama lain. Terdapat perancangan program pengolah citra digital untuk memproses gambar dari pola batik, ada pula perancangan kontrol perangkat keras yaitu *conveyor* sebagai aktuator penggerak kain batik yang akan diuji, dan perancangan komunikasi serial untuk meghubungkan proses pengolahan citra digital dan proses kontrol pada *conveyor*.



Gambar 3.7 Flowchat Perancangan Keseluruhan Program

Saat pertama kali sistem dijalankan sesuai *flowchart* pada gambar 3.7, Arduino menunggu perintah dari PC sebagai pemroses citra digital, jika PC sudah mengirim data berupa karakter "a" maka arduino akan menggerakkan motor DC untuk menjalankan conveyor. Proses ini berfungsi untuk menjalankan kain batik sehingga pola dari kain batik dapat ditangkap oleh kamera sebagai awal dari proses pengolahan citra digital yang akan dilakukan oleh PC secara terus menerus atau real time.



Gambar 3.8 (a) Flowchart pada Visual Studio (b) Flowchart Program Arduino

Pada gambar 3.8, Proses pertama pada tahap pengolahan citra digital yaitu mengambil gambar (*frame*) pada pola kain batik yang bertipe RGB (*red, green, blue*) sebagai gambar uji. Selanjutnya *frame* tersebut akan diubah menjadi citra keabu-abuan (*grayscale*) untuk kemudian diproses oleh metode *First Order*. Untuk merubah menjadi citra keabuan (*grayscale*) digunakan rumus :  $Grayscale = \frac{R+G+B}{3}$ , Hasil dari konversi RGB ke *grayscale* seperti gambar berikut :

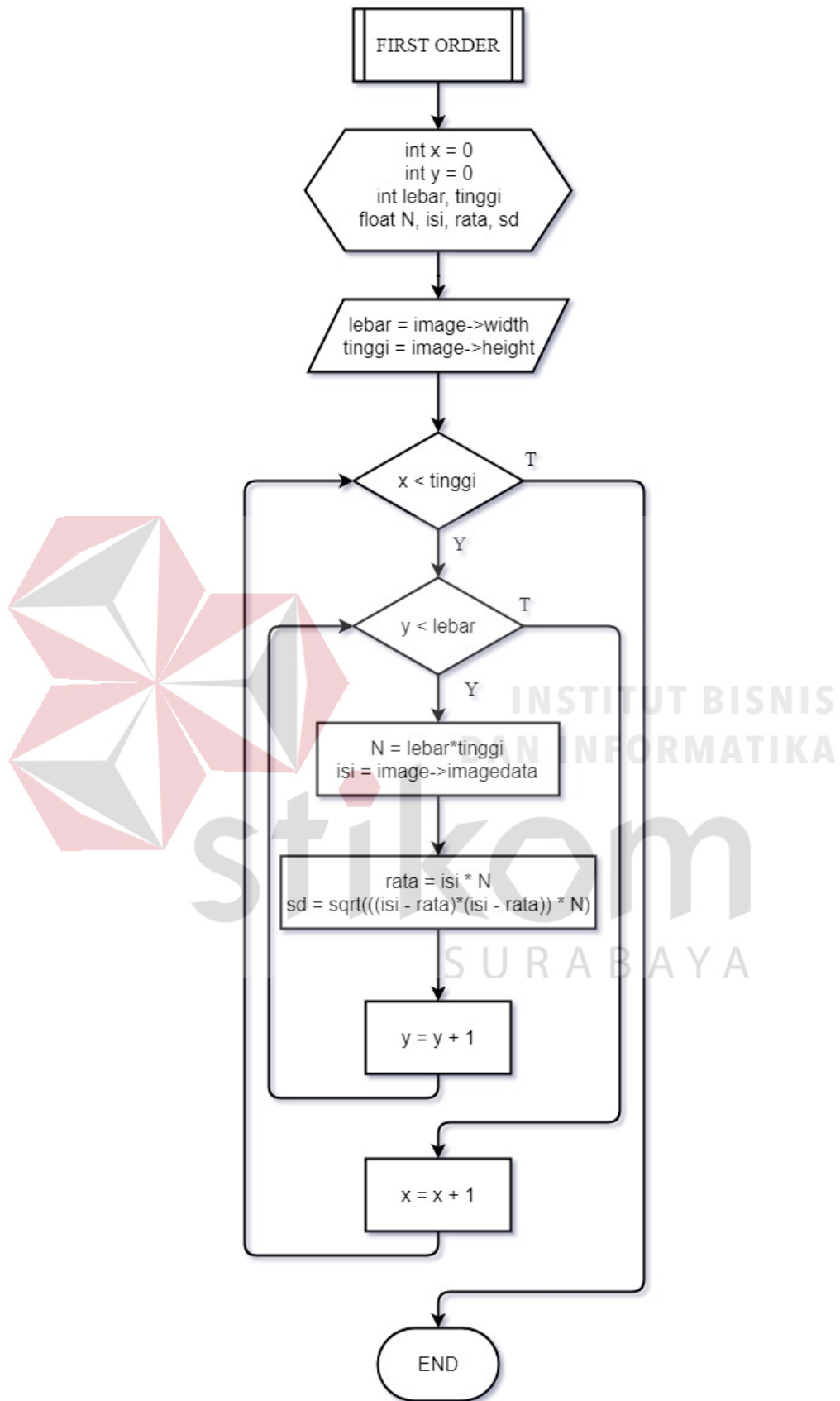


Gambar 3.9 Citra RGB



Gambar 3.10 Citra *Grayscale*

Setelah melalui proses merubah RGB menjadi *grayscale* seperti pada gambar 3.9 dan 3.10, dilakukan proses selanjutnya yaitu pemrosesan gambar menggunakan Metode *First Order*. Pada metode ini parameter yang digunakan yaitu *Mean* dan Standar Deviasi. Proses perhitungan pada Metode *First Order* seperti pada *Flowchart* berikut ini.



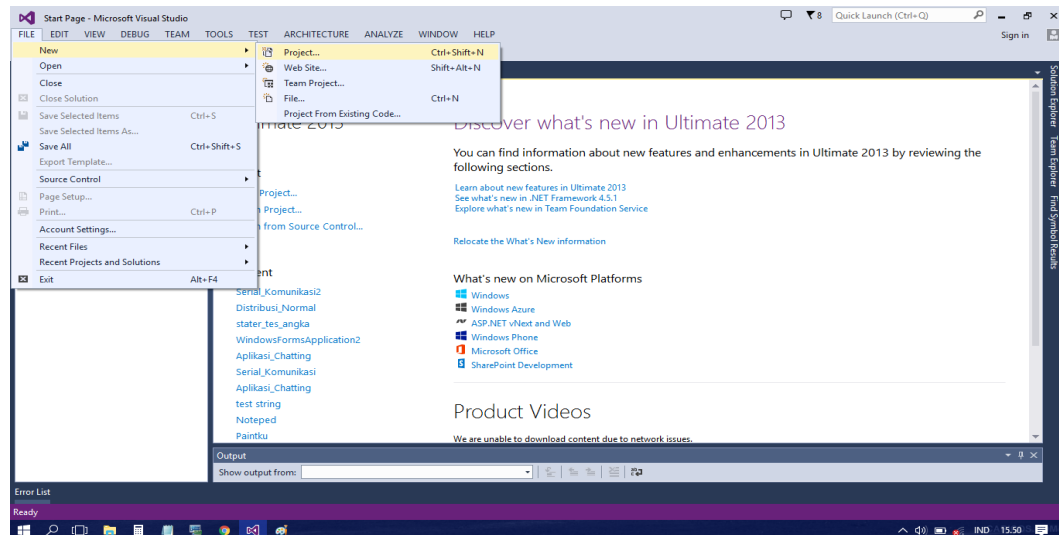
Gambar 3.11 Flowchart Metode First Order

Pada gambar 3.11, Setelah didapatkan hasil dari proses *First Order*, dilakukan pengujian untuk menentukan nilai *range threshold* dari gambar uji. kemudian, dilakukan perbandingan data antara data yang sedang dideteksi dengan *range threshold*. Jika hasil perbandingan ada didalam dari *range threshold*, gambar uji dianggap memiliki kecocokan yang tinggi sehingga pola tersebut benar dan proses akan berjalan secara terus-menerus. Jika gambar uji memiliki nilai tidak dalam *range threshold*, komputer sebagai *device* yang menjalankan proses pengolahan citra digital akan menghitung jumlah kesalahan pada satu potong kain batik. Setelah satu potong kain telah selesai di deteksi, user melakukan penekanan tombol stop dan PC mengirim karakter “b” kepada arduino. Lalu, apabila dalam satu kain terdapat kesalahan yang melebihi batas kesalahan maksimum, maka PC mengirim karakter “c” kepada Arduino melalui komunikasi serial dan membunyikan *buzzer* sebagai isyarat bahwa sistem telah mendeteksi kain batik tersebut tidak layak.

### 3.2.1 Perancangan Program Pengolahan Citra

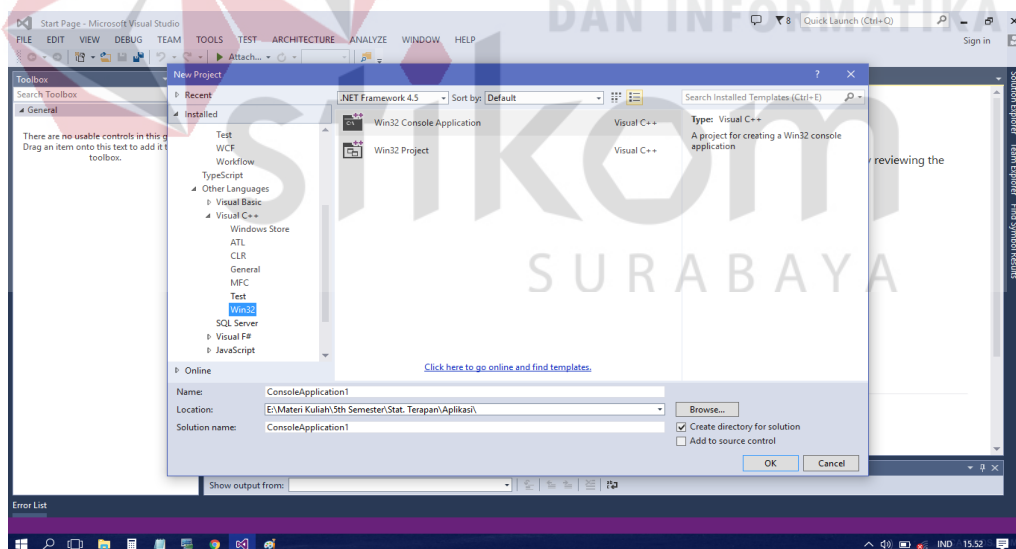
Bahasa pemrograman pengolahan citra digital yang digunakan dalam sistem ini adalah C++ menggunakan aplikasi Visual Studio 2013 dan OpenCV 2.1 sebagai *library* yang umum digunakan dalam proses pengolahan citra digital. Keduanya harus dikoneksikan supaya *library* yang akan dipakai dapat berjalan sesuai yang diinginkan. Berikut ini adalah tata cara membuat dan mengkoneksikan Visual Studio 2013 dan OpenCV 2.1.

1. Mengklik *Toolbar File > New > Project*.



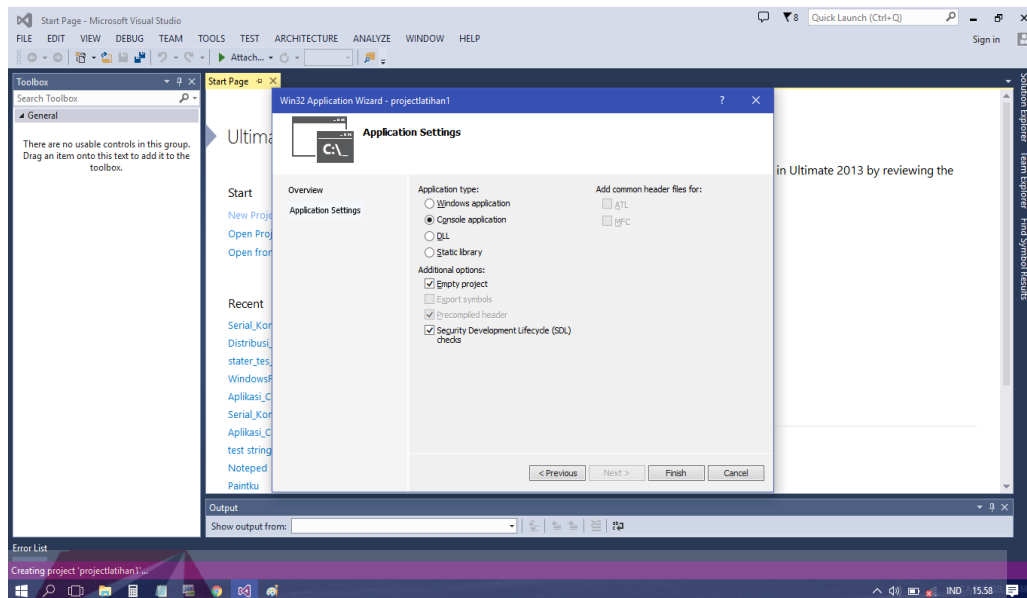
Gambar 3.12 Klik *File*

2. Memilih *Win32 Console Application* kemudian memberi nama dan memilih tempat penyimpanan *Project* tersebut selanjutnya mengklik OK.



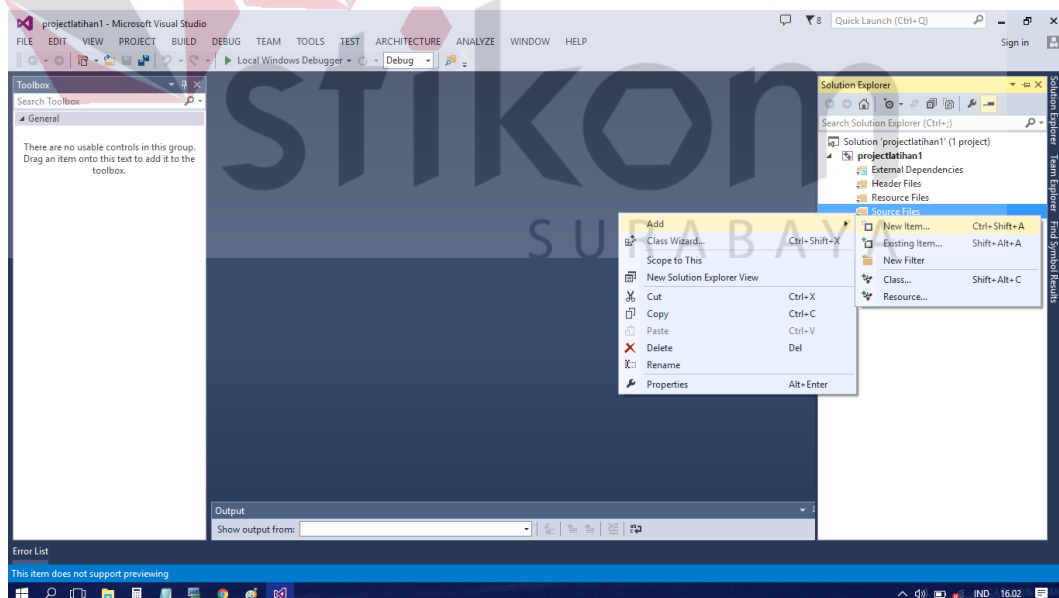
Gambar 3.13 Menyimpan *Project*

3. Memilih *Application Type* dan *Additional Option* sesuai gambar 3. 10.



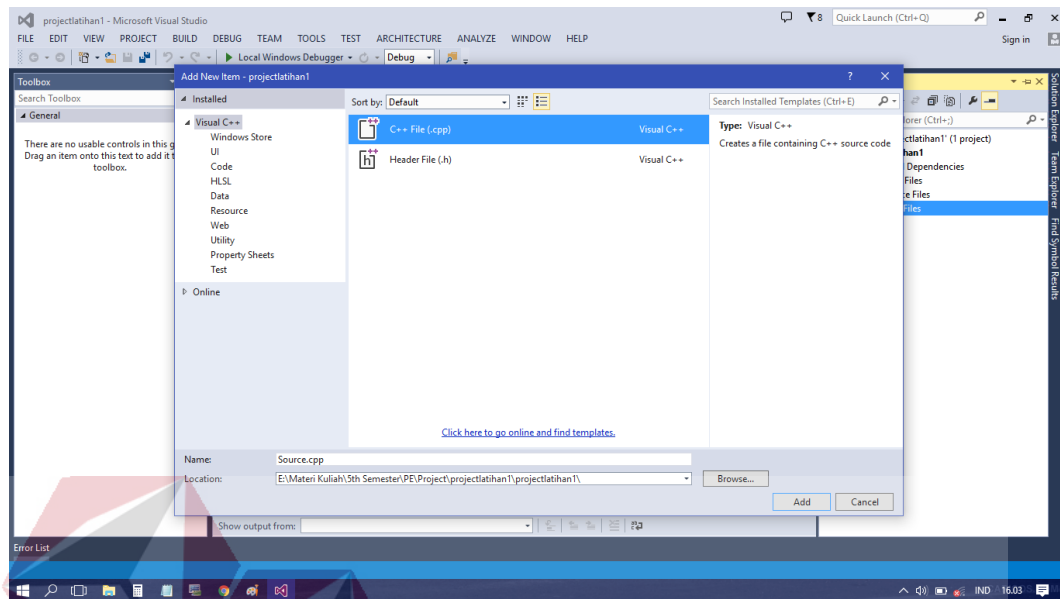
Gambar 3.14 Memilih *Application Type* dan *Additional Option*

4. Membuat lembar kerja dengan cara klik kanan *Source File* > *Add* > *New Item*.



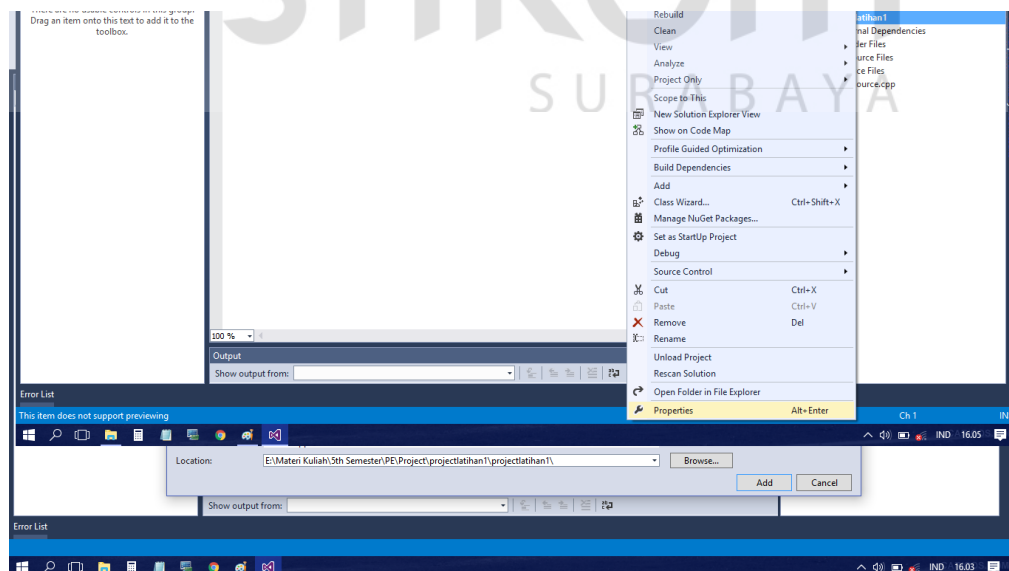
Gambar 3.15 Membuat *Source File*

5. Memilih *C++ File* kemudian memberi nama dan tempat penyimpanan *Source file* tersebut.



Gambar 3.16 Menyimpan *Source File*

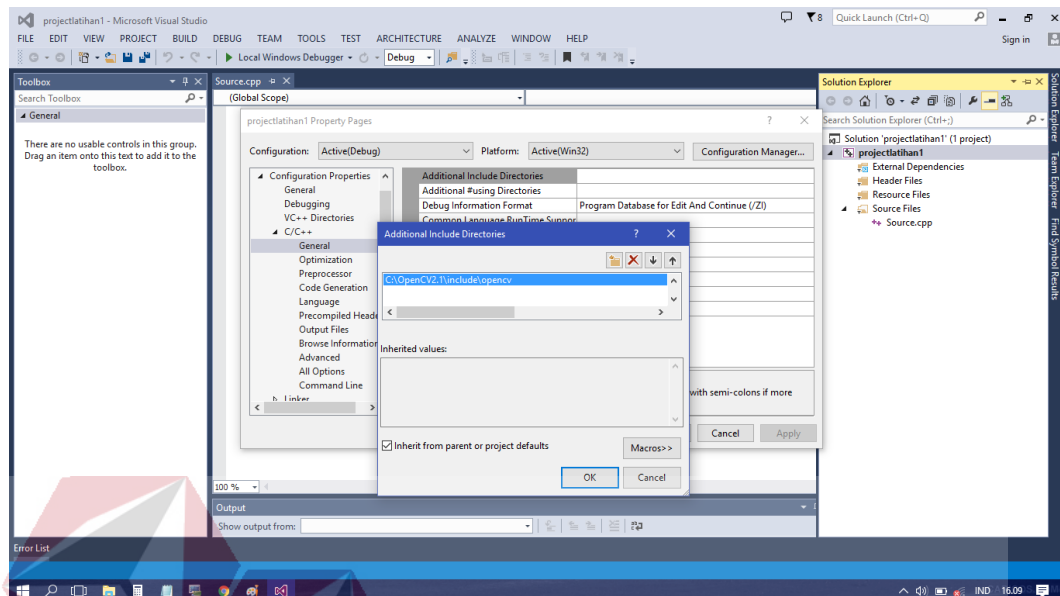
6. Menghubungkan *library* OpenCV 2.1 dan Visual Studio 2013 dengan mengeklik kanan *Source File* dan pilih *Properties*.



Gambar 3.17 Properties *Source File*

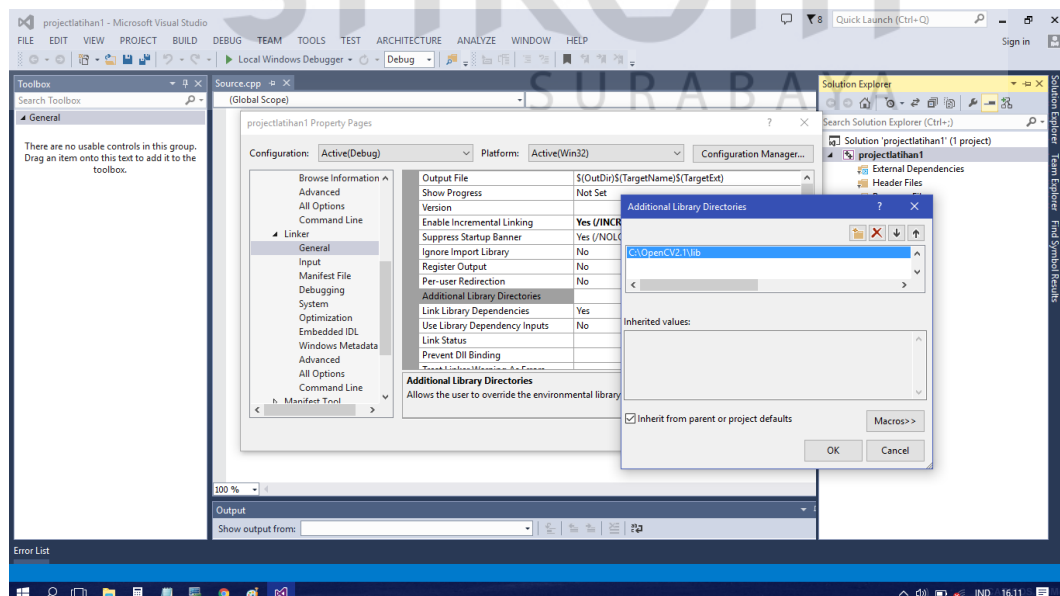


7. Memilih Tab *C/C++ > General > Additional Include Directories > Browse*  
> dan memasukkan alamat *C:\OpenCV2.1\include\opencv > OK*.



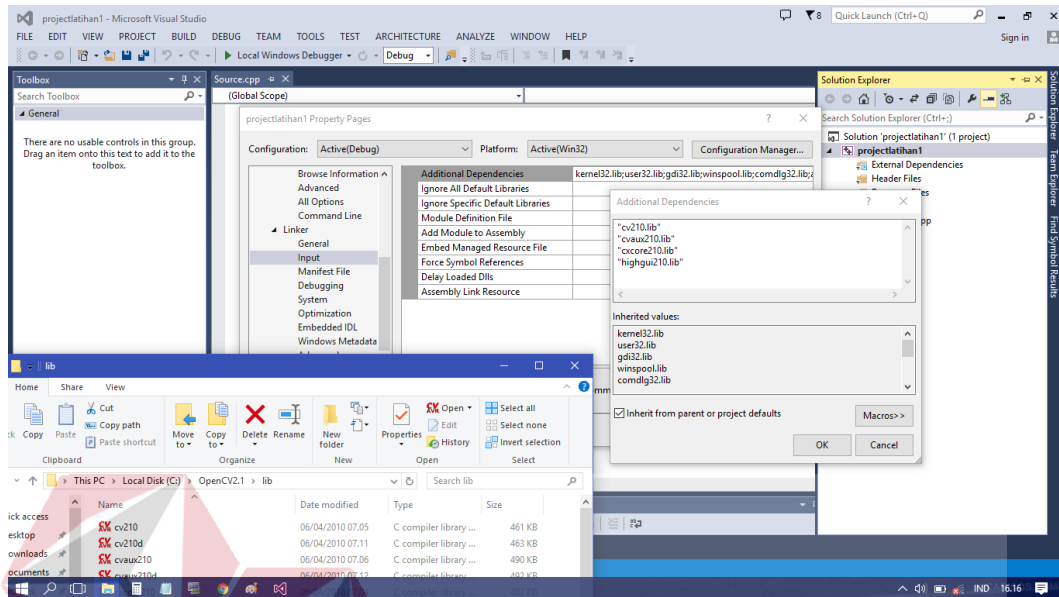
Gambar 3.18 Memasukkan *Library Tab C/C++*

8. Memilih Tab *Linker > General > Additional Include Directories > Browse*  
> dan memasukkan alamat *C:\OpenCV2.1\lib > OK*.



Gambar 3.19 Memasukkan *Library Pada General Linker*

9. Memilih Tab *Linker* > *Input* > *Additional Include Directories* > menuliskan library yang ada pada C:\OpenCV2.1\lib > OK.



Gambar 3.20 Memasukkan *Library* Pada *Input Linker*

Setelah melakukan sinkronisasi *library*, pengolahan citra digital dimulai dengan mengambil gambar *frame* dari kamera dan ditampilkan dalam sebuah *window*. Berikut adalah *listing* Program dari proses pengambilan *frame* dari kamera.

```
IplImage* kamera;
CvCapture* capture = cvCaptureFromCAM(0);
cvNamedWindow("asli", CV_WINDOW_AUTOSIZE);
kamera = cvQueryFrame(capture);
cvShowImage("asli", kamera);
```

Selanjutnya gambar dari kamera akan diubah kedalam citra *grayscale*. Berikut adalah *listing* Program dari proses mengubah gambar dari kamera menjadi citra *grayscale*.

```
IplImage* gray = cvCreateImage(cvGetSize(kamera),
IPL_DEPTH_8U, 1);
cvCvtColor(kamera, gray, CV_RGB2GRAY);
```

Proses inti dari sistem ini adalah perhitungan dari gambar *training* dengan gambar uji menggunakan metode *First Order* untuk mengetahui kecocokan dari pola batik. Parameter *First Order* yang digunakan yaitu *Mean* dan *Standar Deviasi* karena dinilai dapat membedakan pola batik dari dispersi warna pada citra. Berikut adalah *listing* Program dari metode *First Order*.

```
for (int r = 0; r<gray->height; r++)
{
    for (int c = 0; c<gray->width; c++)
    {
        isi2 = gray->imageData[gray->widthStep*r + c];
        sumx2 = sumx2 + (isi2 * n);
        sd = sqrt((isi2 - sumx2)*(isi2 - sumx2)*n)/100;
    }
}
```

Terdapat pula komunikasi serial pada program pengolahan citra digital untuk mengirimkan data ke Arduino jika sistem mendeteksi kesalahan pada pola batik. Berikut adalah *listing* Program dari komunikasi serial dari PC ke Arduino.

```
arduino.writeSerialPort(input, MAX_DATA_LENGTH);
```

*Listing* diatas berfungsi untuk mengirim data dari variabel *input* ke Arduino melalui komunikasi serial.

### 3.2.2 Perancangan Program Arduino

Tidak hanya dari sisi pengolahan citra, Arduino juga salah satu unsur penting dari sistem ini dalam sisi perangkat keras (*hardware*). Proses pertama yang dilakukan ketika sistem baru dimulai adalah menghidupkan motor sebagai

penggerak *conveyor*. Berikut adalah *listing* Program dari pengoperasian motor.

```
pinMode(pwm1, OUTPUT);
pinMode(pwm2, OUTPUT);
analogWrite(pwm1, 250);
analogWrite(pwm2, 0);
delay(200);
```

Proses komunikasi serial juga terdapat pada Arduino untuk menerima data yang dikirim oleh komputer jika sistem mendeteksi ketidakcocokan pada pola batik.

Berikut ini adalah *listing* program pada arduino dari pengoperasian motor.

```
input = Serial.read();
if(input=='a')
{
    analogWrite(pwm1, 50);
    analogWrite(pwm2, LOW);
}
if(input=='b')
{
    analogWrite(pwm1, LOW);
    analogWrite(pwm2, LOW);
}
else
{
    analogWrite(pwm1, LOW);
    analogWrite(pwm2, LOW);
    digitalWrite(11,HIGH);
}
```

Penjelasan dari program diatas Arduino menunggu data yang dikirim oleh PC melalui komunikasi serial dan data tersebut dimasukkan kedalam variabel *input*. Kemudian membaca data dari variabel *input*, jika variabel *input* sama dengan karakter “a” maka Arduino akan mengontrol motor untuk berjalan dengan nilai direksi PWM sebesar 50. *Conveyor* akan beroperasi secara terus menerus sampai variabel *input* diisi oleh karakter “b” yang berfungsi untuk memberhentikan motor. Kemudian apabila variabel *input* tidak sama dengan karakter “a” ataupun “b”, sistem akan memberhentikan motor dan membunyikan *buzzer*.

## BAB IV

### PENGUJIAN

Dalam bab ini akan dibahas tentang beberapa pengujian untuk sistem yang telah dirancang pada Tugas Akhir ini. Tujuan dari bab ini adalah untuk mengetahui tingkat keberhasilan terhadap perancangan sistem yang telah diajukan dan dikerjakan. Pada bab ini terdapat beberapa pengujian yaitu uji kecepatan putar motor, dan uji akurasi deteksi kesalahan pola batik.

#### 4.1 Uji Kecepatan Deteksi Kamera

##### 4.1.1 Tujuan Uji Kecepatan Deteksi Kamera

Pengujian ini digunakan untuk mengetahui perbedaan gambar yang dapat ditangkap oleh *webcam* dengan cara memberikan *input* kecepatan berupa PWM pada motor dan mengamati respon *output* gambar yang ditangkap oleh *webcam*. Dengan diketahuinya perbedaan hasil tangkapan gambar, nilai tersebut dapat ditentukan sebagai nilai ideal untuk *webcam* menangkap gambar.

##### 4.1.2 Alat Yang Digunakan Pada Uji Kecepatan Deteksi Kamera

Peralatan yang dibutuhkan untuk pengujian ini adalah sebagai berikut:

1. *Conveyor*
2. Laptop dan Arduino IDE
3. Program Arduino untuk menampilkan nilai PWM
4. *Webcam*

#### 4.1.3 Prosedur Pengujian Pada Uji Kecepatan Deteksi Kamera

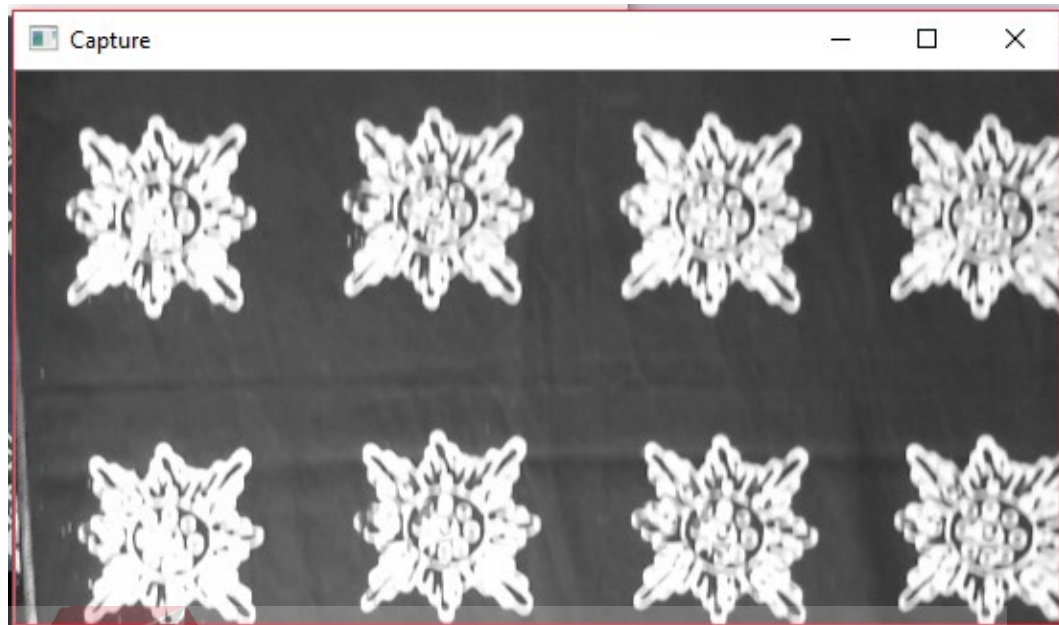
Langkah-langkah yang dilakukan dalam melakukan pengujian ini adalah sebagai berikut:

1. Mempersiapkan *Conveyor*.
2. Melakukan transfer program dari laptop ke Arduino.
3. Menjalankan *conveyor*.
4. Amati gambar yang ditangkap oleh *webcam*.

#### 4.1.4 Hasil Pengujian Pada Uji Kecepatan Deteksi Kamera

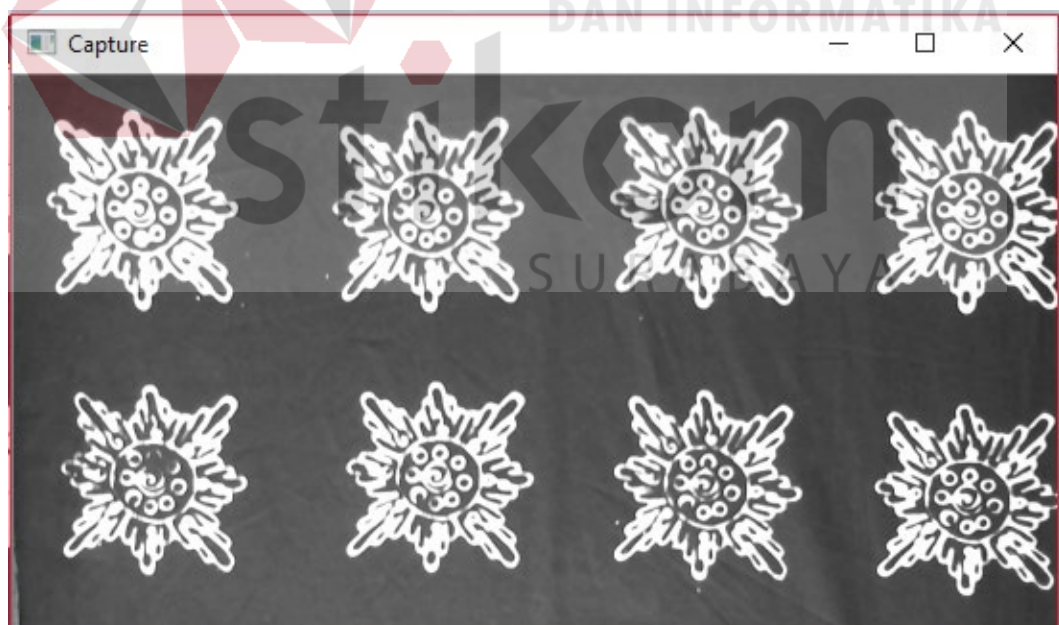
Hasil dari proses pengujian kecepatan masing-masing motor menunjukkan bahwa ketika motor diberi nilai PWM 20, motor tidak dapat memutar *conveyor*. Dengan nilai PWM 30 motor dapat memutar *conveyor*, nilai PWM sebesar 30 ini menjadi batas bawah nilai agar motor dapat memutar *conveyor* dengan baik.

Ketika nilai PWM yang diberikan sebesar 100, maka motor dapat memutar *conveyor* dengan baik, namun gambar hasil tangkapan *webcam* tidak dapat terlihat dengan jelas atau bisa disebut dengan *blur*.



Gambar 4.1 Pola *Blur*

Batas atas nilai PWM agar motor dapat memutar *conveyor* dengan baik dan *webcam* dapat menangkap gambar dengan baik atau tidak *blur* yaitu 80.



Gambar 4.2 Pola Tidak *Blur*

## 4.2 Uji Komunikasi Serial

### 4.2.1 Tujuan Uji Komunikasi Serial

Pada pengujian komunikasi serial ini digunakan untuk mengetahui apakah Arduino uno dapat mengirimkan data kepada Visual Studio atau sebaliknya. Pengujian komunikasi serial ini dilakukan dengan cara melakukan pengiriman data dari Visual Studio ke Arduino uno dan sebaliknya.

### 4.2.2 Alat Yang Digunakan Pada Uji Komunikasi Serial

Peralatan yang dibutuhkan untuk pengujian ini adalah sebagai berikut:

1. Arduino Uno
2. Laptop dan Visual Studio
3. Kabel serial (Downloader Arduino)

### 4.2.3 Prosedur Pengujian Pada Uji Komunikasi Serial

Langkah-langkah yang dilakukan dalam melakukan pengujian ini adalah sebagai berikut:

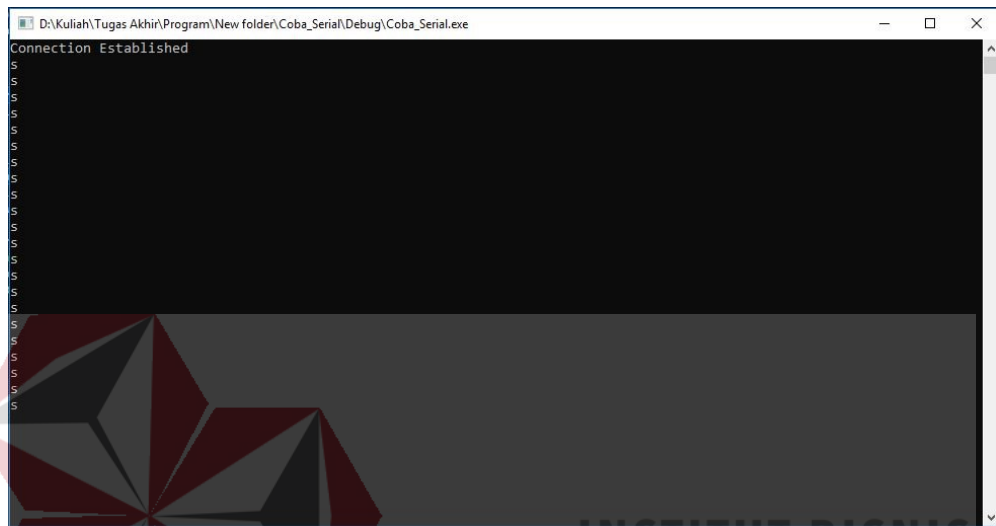
1. Mempersiapkan Arduino Uno yang telah di *load* program
2. Mempersiapkan program Visual Studio di laptop
3. Menghubungkan koneksi antara Arduino dengan Laptop
4. Menjalankan program untuk mengirim data ke Arduino dan sebaliknya

### 4.2.4 Hasil Pengujian Pada Uji Komunikasi Serial

Pada pengujian komunikasi serial ini Laptop dan Arduino Uno dihubungkan dengan kabel *downloader* yang bisa juga digunakan sebagai komunikasi serial.

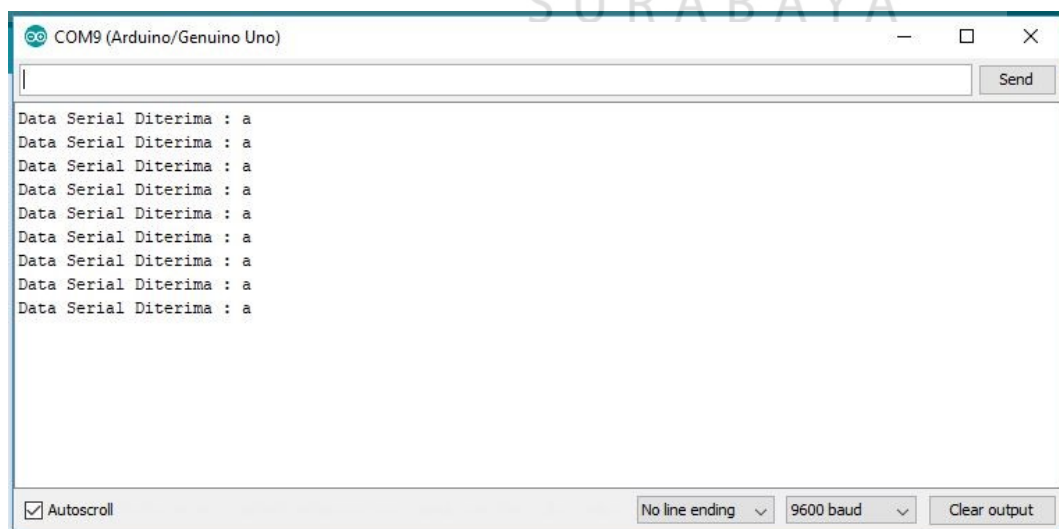


Ketika program di Visual Studio di *run*, maka akan memunculkan window baru yang berupa cmd. Apabila koneksi serial antara Visual Studio dan Arduino telah berhasil terhubung, maka di cmd akan muncul “*connection established*” data karakter huruf “s”.



Gambar 4.3 Monitoring serial dengan *Command Prompt*

Sedangkan pada *serial monitor* pada arduino juga dapat menampilkan data yang dikirimkan oleh Laptop berupa karakter “a”.



Gambar 4.4 Monitoring Serial dengan Serial Port

### 4.3 Uji Akurasi Deteksi Kerusakan Pola Batik

#### 4.3.1 Tujuan Uji Akurasi Deteksi Kerusakan Pola Batik

Tujuan dari pengujian akurasi deteksi kerusakan pola batik untuk mengetahui keakuratan dari metode *First Order* dengan menggabungkan hasil dari pengujian yang telah dilakukan.

#### 4.3.2 Alat Yang Digunakan Pada Uji Deteksi Kerusakan Pola Batik

Peralatan yang dibutuhkan untuk pengujian ini adalah sebagai berikut:

1. *Conveyor*
2. Laptop dan Visual Studio
3. Objek Kain Batik Surya Majapahit

#### 4.3.3 Prosedur Pengujian Pada Uji Deteksi Kerusakan Pola Batik

Langkah-langkah yang dilakukan dalam melakukan pengujian ini adalah sebagai berikut:

1. Mempersiapkan Arduino Uno yang telah di *load* program.
2. Mempersiapkan program Visual Studio di laptop.
3. Menghubungkan koneksi antara Arduino dengan Laptop dan *webcam* dengan laptop.
5. Menjalankan program Visual Studio dan Arduino IDE.
6. Amati *output* cmd pada laptop.

#### 4.3.4 Hasil Pengujian Pada Uji Deteksi Kerusakan Pola Batik

Hasil dari pengujian ini adalah untuk mengetahui kecepatan maksimal dan minimal metode ini dapat melakukan deteksi kerusakan pola batik serta efektivitas metode ini dalam hal melakukan deteksi terhadap kecepatan yang berbeda-beda.

Pengujian yang pertama akan dilakukan dengan cara mendeteksi pola kain batik yang digunakan sebagai data *training*, pengujian akan mengambil sampel sebanyak 10 kali pada kain dengan pola yang dianggap benar.

Dari percobaan diatas, didapatkan data *training* digunakan sebagai acuan nilai rata – rata dan standar deviasi untuk pola yang benar. Didapatkan range nilai rata – rata 350 sampai dengan 450, sedangkan range nilai dari standar deviasi adalah 139 sampai 178. Pada satu potong kain batik dengan panjang 2 meter yang digunakan *sample* memiliki kesalahan saat pendeteksian kurang dari 100 kesalahan, maka kain tersebut dapat dinyatakan layak. Selanjutnya dilakukan dengan cara mendeteksi pola batik dalam berbagai kecepatan motor dan kondisi kain yang berbeda.

Tabel 4.1 Pengujian dengan 30 PWM

Objek	Nilai Minimal dan Maksimal	Nilai Rata – Rata	Nilai Standar Deviasi	Jumlah Kesalahan
Kain 1	Maksimal	480,52	203,77	64
	Minimal	354,11	140,16	
Kain 2	Maksimal	491,97	208,62	76
	Minimal	386,52	152,99	
Kain 3	Maksimal	501,79	203,08	127
	Minimal	415,95	164,64	
Kain 4	Maksimal	522,30	206,73	153
	Minimal	437,66	173,23	

Dengan PWM 30, sistem dapat mendeteksi dengan baik dan mampu membedakan kain layak dan tidak layak.

Tabel 4.2 Pengujian dengan 50 PWM

Objek	Nilai Minimal dan Maksimal	Nilai Rata – Rata	Nilai Standar Deviasi	Jumlah Kesalahan
Kain 1	Maksimal	479,97	203,31	59
	Minimal	360,95	142,87	
Kain 2	Maksimal	493,01	209,07	74
	Minimal	362,67	143,55	
Kain 3	Maksimal	503,17	213,37	131
	Minimal	421,08	166,67	
Kain 4	Maksimal	518,60	205,27	156
	Minimal	446,66	176,80	

Pada nilai PWM 50, sistem juga dapat mendeteksi dengan baik dan sesuai dengan target yang diinginkan, yaitu 2 kain layak dan 2 kain tidak layak. Maka dapat disimpulkan dengan PWM 50 lebih baik daripada nilai PWM 30, dikarenakan kecepatan lebih tinggi dan dengan ketelitian sama dengan nilai PWM 30.

Tabel 4.3 Pengujian dengan 80 PWM

Objek	Nilai Minimal dan Maksimal	Nilai Rata – Rata	Nilai Standar Deviasi	Jumlah Kesalahan
Kain 1	Maksimal	479,97	203,31	87
	Minimal	360,95	142,87	

Kain 2	Maksimal	493,01	209,07	102
	Minimal	362,67	143,55	
Kain 3	Maksimal	503,17	213,37	151
	Minimal	421,08	166,67	
Kain 4	Maksimal	518,60	205,27	179
	Minimal	446,66	176,80	

Nilai PWM 80, tidak dapat mendeteksi dengan baik dikarenakan kain yang seharusnya dianggap layak, namun pada sistem terdeteksi kesalahan lebih dari 100 kali, yang menyebabkan sistem menganggap kain tersebut tidak layak. Dari 2 kain layak dan 2 kain tidak layak, sistem mendeteksi 1 kain layak dan 3 kain tidak layak.

Kemudian dilakukan pengujian untuk menentukan kelayakan sebuah kain, dilakukan 10 percobaan pada 4 kain dengan kondisi yang berbeda tingkat kelayakannya. Dari 10 percobaan pada setiap kain bisa didapatkan presentase *error* dari sistem yang nantinya digunakan sebagai hasil akhir dari pengujian ini. Nilai PWM yang digunakan pada pengujian ini adalah 50.

Tabel 4.4 Pengujian Kain Batik A

Pengujian ke	Jumlah Salah	Hasil Deteksi	Target
1	71	Layak	Layak
2	74	Layak	
3	69	Layak	
4	77	Layak	
5	73	Layak	
6	81	Layak	

7	79	Layak	Layak
8	78	Layak	
9	86	Layak	
10	74	Layak	

Tabel 4.5 Pengujian Kain Batik B

Pengujian ke	Jumlah Salah	Hasil Deteksi	Target
1	89	Layak	Layak
2	92	Layak	
3	86	Layak	
4	97	Layak	
5	103	Tidak Layak	
6	96	Layak	
7	95	Layak	
8	100	Tidak Layak	
9	86	Layak	
10	93	Layak	

Tabel 4.6 Pengujian Kain Batik C

Pengujian ke	Jumlah Salah	Hasil Deteksi	Target
1	123	Tidak Layak	Tidak Layak
2	113	Tidak Layak	
3	97	Layak	
4	118	Tidak Layak	

5	121	Tidak Layak	Tidak Layak
6	119	Tidak Layak	
7	109	Tidak Layak	
8	112	Tidak Layak	
9	99	Layak	
10	107	Tidak Layak	

Tabel 4.7 Pengujian Kain Batik D

Pengujian ke	Jumlah Salah	Hasil Deteksi	Target
1	128	Tidak Layak	Tidak Layak
2	133	Tidak Layak	
3	137	Tidak Layak	
4	138	Tidak Layak	
5	141	Tidak Layak	
6	149	Tidak Layak	
7	139	Tidak Layak	
8	142	Tidak Layak	
9	135	Tidak Layak	
10	137	Tidak Layak	

Setelah dilakukan 40 kali pengujian pada semua kain atau 10 kali pengujian pada setiap kain didapatkan hasil bahwa terdapat *error* dari pengujian ini sebanyak 4 kali, 36 kali sistem mendeteksi dengan benar, dan 4 kali sistem mendeteksi salah. Nilai pendeteksian yang salah oleh sistem tidak jauh beda dari



batas kesalahan maksimal untuk bisa dinyatakan layak yaitu 100 kesalahan. Dari hasil tersebut, didapatkan presentase keakuratan deteksi kerusakan pola batik sebesar 90%.



Gambar 4.5 Contoh Pola Terdeteksi Benar



Gambar 4.6 Pola yang Terdeteksi Salah



## BAB V

### PENUTUP

Dalam bab ini berisi tentang kesimpulan dan saran yang berdasar pada hasil dari pengujian yang telah dilakukan pada Tugas Akhir ini, maka dapat diperoleh beberapa kesimpulan dan saran untuk pengembangan sistem berikutnya.

#### 5.1 Kesimpulan

Berdasar hasil pengujian pada Tugas Akhir ini dapat disimpulkan beberapa kesimpulan sebagai berikut :

1. Hasil dari proses pengujian kecepatan deteksi kerusakan pola batik menunjukkan bahwa kecepatan minimum untuk conveyor dapat berputar dengan baik dan kamera *webcam* dapat menangkap dengan baik bernilai 30 PWM hingga 80 PWM.
2. Dari pengujian keakuratan metode *First Order* dengan kecepatan yang berbeda, didapatkan nilai PWM paling efektif adalah 50.
3. Dapat disimpulkan hasil dari pengujian keakuratan metode *First Order* untuk mendeteksi kesalahan pada pola batik tingkat keberhasilan dengan nilai PWM 50 yaitu sebesar 90%.

## 5.2 Saran

Saran yang diberikan oleh penulis pada pengembangan Tugas Akhir ini selanjutnya adalah sebagai berikut:

1. Untuk mendapat hasil akurasi yang lebih baik dapat menggunakan metode orde kedua atau metode yang berdasarkan pengenalan tekstur.
2. Karena Tugas Akhir ini hanya berupa *prototype* sebaiknya dibuat dengan skala yang sesuai dengan kondisi industri pasar.



## DAFTAR PUSTAKA

- Arisandi, B., Suciati, N., & Yudhi Wijaya, A. (2011). Pengenalan Motif Batik Menggunakan Rotated Wavelet Filte dan Neural Network. *JUTI, Jurnal Ilmiah Teknologi Informasi*, 9(2), 13–19.
- Dananjaya, I. P. A. V. (2017). *Analisis Kerusakan Tekstil Berbasis Tekstur First Order*. Institut Bisnis dan Informatika Stikom Surabaya.
- Djuandi, F. (2011). Pengenalan Arduino. *E-Book. Tobuku*, 1–24.  
<https://doi.org/10.1093/fampra/cm112>
- Hemera, A. (2018). Fungsi Microsoft Visual Studio. Retrieved from <https://itlearningcenter.id/fungsi-microsoft-visual-studio/>
- Muzami, A., Nurhayati, O. D., Martono, K. T., Studi, P., Komputer, S., Teknik, F., & Diponegoro, U. (2016). Aplikasi Identifikasi Citra Telur Ayam Omega-3 Dengan Metode Segmentasi Region Of Interest Berbasis Android. *Universitas Diponegoro*, 4(2), 380–388.
- Prasojo, P. (2015). Computer Vision. Retrieved from <https://praptoprasojo.wordpress.com/2015/11/13/computer-vision/>
- Rinaldy, R., Christiani, R. F., & Supriyadi, D. (2016). Pengendalian Motor Servo Yang Terintegrasi Dengan Webcam Berbasis Internet Dan Arduino. *JURNAL INFOTEL - Informatika Telekomunikasi Elektronika*, 5(2), 17.  
<https://doi.org/10.20895/infotel.v5i2.4>
- Santoso, F. L., Sutanto, R. P., Studi, P., Komunikasi, D., Seni, F., & Petra, U. K. (2014). Perancangan Buku Tentang Batik Mojokerto. *Universitas Kristen Petra*, 1–12.

Yodha, J. W., & Kurniawan, A. W. (2014). Pengenalan Motif Batik Menggunakan Deteksi Tepi Canny Dan K-Nearest Neighbor. *Jurnal Techno.COM, Vol. 13, No. 4, November 2014: 251-262, 13(4), 251–262.* [https://doi.org/DOI:10.1016/0169-5347\(91\)90210-O](https://doi.org/DOI:10.1016/0169-5347(91)90210-O)

